SGOU
Higher Education for All

DATA ANALYTICS AND VISUALIZATION

COURSE CODE: B21CA03DE

Bachelor of Computer Applications
Discipline Specific Elective Course
Self Learning Material



SGOU

SREENARAYANAGURU OPEN UNIVERSITY

SREENARAYANAGURU OPEN UNIVERSITY

The State University for Education, Training and Research in Blended Format, Kerala

SREENARAYANAGURU OPEN UNIVERSITY

SGOU

Vision

To increase access of potential learners of all categories to higher education, research and training, and ensure equity through delivery of high quality processes and outcomes fostering inclusive educational empowerment for social advancement.

Mission

To be benchmarked as a model for conservation and dissemination of knowledge and skill on blended and virtual mode in education, training and research for normal, continuing, and adult learners.

Pathway

Access and Quality define Equity.

Data Analytics and Visualization

Course Code: B21CA03DE

Semester - VI

SGOU

Discipline Specific Elective Course
Undergraduate Programme
Bachelor of Computer Applications
Self Learning Material

(With Model Question Paper Sets)

SREENARAYANAGURU
OPEN UNIVERSITY

SREENARAYANAGURU OPEN UNIVERSITY

The State University for Education, Training and Research in Blended Format, Kerala

SREENARAYANAGURU
OPEN UNIVERSITY

DATA ANALYTICS AND VISUALIZATION

Course Code: B21CA03DE
Semester- VI
Discipline Specific Elective Course
Bachelor of Computer Applications

Academic Committee

Dr. M.V. Judy
Dr. Aji S.
Dr. Vishnukumar S.
Rajesh R.
Dr. Rafidha Rehiman K.A.
P.M. Ameera Mol
Dr. Ajitha R.S.
Dr. Bindu Lal T.S.
Dr. Sreeja S.

Development of the Content

ICT Academy of Kerala

Review

Mahesh C.

Edit

Dr. Rohini S. Nair

Proofreading

Dr. Rohini S. Nair

Scrutiny

Greeshma P.P.
Dr. Kanitha D.K.
Sreerekha V.K.,
Shamin S.
Subi Priya Laxmi S.B.N.
Aswathy V.S.

Cover Design

Jobin J.

Co-ordination

Director (IC), MDDC :
Dr. Sajeevkumar G.
Coordinator, Development:
Dr. Anfal M.
Coordinator, Distribution:
Dr. Abdul Razak Kunnathodi

Scan this QR Code for reading the SLM on a digital device.

Edition
October 2026

Copyright
© Sreenarayanaguru Open University

ISBN 978-81-903304-3-5

9 788190 330435

SGOU

All rights reserved. No part of this work may be reproduced in any form, by mimeograph or any other means, without permission in writing from Sreenarayanaguru Open University. Printed and published on behalf of Sreenarayanaguru Open University by Registrar, SGOU, Kollam.

www.sgou.ac.in

Visit and Subscribe our Social Media Platforms

MESSAGE FROM VICE CHANCELLOR

Dear

With immense joy and excitement, I extend my heartfelt greetings to all of you and warmly welcome you to Sreenarayanaguru Open University.

Established in September 2020 as a state-driven initiative, Sreenarayanaguru Open University is dedicated to advancing higher education through open and distance learning. Our vision is guided by the principle of "access and quality define equity," laying the foundation for a celebration of excellence in education. I am delighted to share that we are steadfast in our commitment to uphold the highest standards and refrain from compromising on the quality of education we offer. The university draws its inspiration from the legacy of Sreenarayana Guru, a revered figure in the Indian renaissance movement. His name serves as a constant reminder for us to prioritize quality in all our academic endeavors.

Sreenarayanaguru Open University operates within the practical framework of the widely recognized "blended format." Acknowledging the constraints faced by distance learners in accessing traditional classroom settings, we have curated a pedagogical approach centered on three main components: Self Learning Material, Classroom Counselling, and Virtual Modes. This comprehensive blend is poised to deliver dynamic learning and teaching experiences, maximizing engagement and effectiveness. Our unwavering commitment to quality ensures excellence across all aspects of our educational initiatives.

The University aims to offer you an engaging and stimulating educational environment that fosters active learning. The SLM is designed to offer a comprehensive and cohesive learning experience, fostering a deep interest in the study of technological advancements in IT. Careful consideration has been given to ensure a logical progression of topics, facilitating a clear understanding of the discipline's evolution. The curriculum is thoughtfully crafted to provide ample opportunities for students to navigate through the current trends in information technology. Furthermore, this course is designed to provide essential insights into computer hardware, software classification, and foundational HTML concepts crucial for web development.

We assure you that the university student support services will closely stay with you for the redressal of your grievances during your studentship. Feel free to write to us about anything that seems relevant regarding the academic programme.

Wish you the best.



Regards,
Dr. Jagathy Raj V.P.

01-10-2026

SGOU

Contents

BLOCK 1 Mathematics for Data Analysts **1**
UNIT 1 Descriptive Statistics 2
UNIT 2 Anomalies and Normalization 19
UNIT 3 Distributions 34
UNIT 4 Concept of Probability 48

BLOCK 2 Introduction to Analytics **61**
UNIT 1 Big Data Technologies and Data Warehousing 62
UNIT 2 Data Analytics Life Cycle 77
UNIT 3 Unstructured Data and ETL 90
UNIT 4 Data Analysis Tools and Platforms 104

BLOCK 3 Advanced Data Analytics **118**
UNIT 1 Basics of Machine Learning Algorithms 119
UNIT 2 Statistical Measures and Error Metrics 132
UNIT 3 Time Series Analysis and Trend Analysis 146
UNIT 4 Text Analytics and Forecasting 159

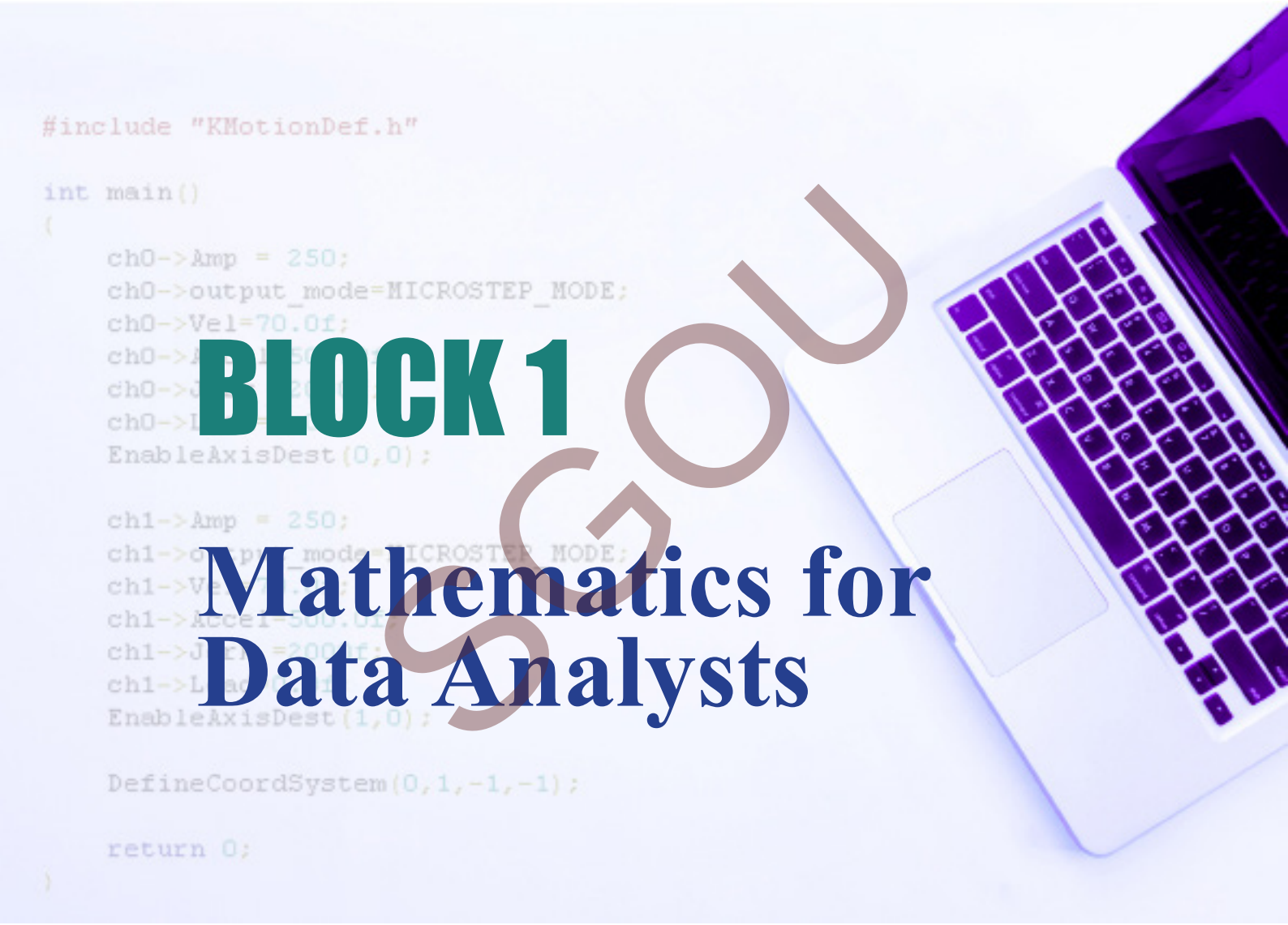
BLOCK 4 Data Visualization **172**
UNIT 1 Data Visualization Basics 173
UNIT 2 Visualization of Geospatial and Multitype Data 186
UNIT 3 Interactive Dashboards 199
UNIT 4 Data Visualization Cycle 212

BLOCK 5 Data Analytics and Visualization using Power BI **225**
UNIT 1 Power BI Basics and Data Preparation 226
UNIT 2 Data Modelling and DAX in Power BI 241
UNIT 3 Building and Publishing Visualizations 256
UNIT 4 Power BI Security and Deployment 269

BLOCK 6 Data Analytics and Visualization using Tableau **283**
UNIT 1 Tableau Basics and Data Preparation 284
UNIT 2 Data Modelling and Calculations 299
UNIT 3 Creating and Publishing Visualizations 313
UNIT 4 Dashboard Development and IoT Implementation 327

Model Questionpaper Sets **342**

SGOU



BLOCK 1

Mathematics for Data Analysts

SGOU



Descriptive Statistics

Learning Outcomes

After completion of this unit, learner will be able to:

- define descriptive statistics and distinguish it from inferential statistics
- differentiate between a population and a sample, and use correct statistical notation for each
- compute and interpret the mean, median and mode of both ungrouped and grouped data
- calculate and interpret measures of dispersion, namely range, variance and standard deviation
- describe skewness, identify the direction of skew in a distribution, and compute a coefficient of skewness
- apply scaling and shifting transformations to numerical data and explain their effect on the mean and standard deviation

SGOU

Prerequisite

Imagine a small bakery owner who has just decided, for the first time, to keep a daily record of how much she sells. After a month, she has a notebook full of numbers and a nagging question: was this actually a good month, or does it just feel that way because of one unusually busy Saturday? To answer that question sensibly, she does not need advanced mathematics -- she needs only to be comfortable adding up a long column of figures, working with simple fractions and percentages, and reading her numbers once they are organized into a table.

As you work through this unit, you will be doing exactly what she eventually learns to do: turning a page of raw numbers into a handful of meaningful figures that actually answer her question. Keeping a calculator or a spreadsheet open alongside the worked examples in this unit -- just as she eventually starts doing herself -- will help you follow each calculation and build the same numeric confidence, before these ideas are applied to much larger, real-world datasets using tools such as Python or R later in the course.

Keywords

Descriptive Statistics, Population, Sample, Mean, Median, Mode, Range, Variance, Standard Deviation, Skewness, Scaling, Shifting.

Discussion

1.1.1 Introduction to Descriptive Statistics

Statistics, as a discipline, is broadly divided into two branches: descriptive statistics and inferential statistics. Descriptive statistics deals with methods of organizing, summarizing, and presenting data in a meaningful way, without drawing conclusions beyond the data at hand. Inferential statistics, on the other hand, uses a sample of data to make generalizations, predictions, or decisions about a larger population.

In the context of data analytics, descriptive statistics is the very first step of any analysis. Before a data analyst can build models, generate visualizations, or draw business insights, the raw data must be understood -- its typical value, its spread, and its overall shape. This process is often referred to as Exploratory Data Analysis (EDA), and descriptive statistics provides the essential toolkit for it.

Consider a retail company that collects daily sales figures from hundreds of stores. Simply looking at thousands of individual numbers tells the analyst very little. However, if the analyst computes the average daily sale, the most frequently occurring sale amount, and how much the sales figures vary from store to store, a clear and actionable picture begins to emerge. This is precisely the role that descriptive statistics plays: it condenses large volumes of raw data into a small number of meaningful summary values.

Descriptive statistics is generally grouped into two broad categories:

- Measures of Central Tendency -- values that describe the centre or typical value of a dataset, such as the mean, median, and mode.
- Measures of Dispersion (or Variability) -- values that describe how spread out the data is around the centre, such as the range, variance, and standard deviation.

This unit introduces both categories in detail, along with the related concepts of skewness (which describes the shape of a distribution) and scaling and shifting (which describe how data values change under simple mathematical transformations). Together, these concepts form the mathematical backbone that supports almost every later stage of the data analytics life cycle, including data cleaning, visualization, and predictive modelling.

1.1.2 Population and Sample

Before any statistical measure can be computed, it is essential to understand whether the data being analyzed represents an entire population or only a sample drawn from it.

SGOU

Population: A population refers to the complete set of items, individuals, or observations that are of interest in a particular study. For example, if a researcher wants to study the average height of all engineering students in a country, then every engineering student in that country forms the population.

Sample: A sample is a subset of the population, selected in such a way that it reasonably represents the characteristics of the whole population. Since it is often impractical, expensive, or time-consuming to collect data from an entire population, analysts typically work with a carefully chosen sample and use it to estimate the properties of the population.

The distinction between population and sample is not merely conceptual -- it directly affects the formulas used to calculate statistical measures. Values calculated from a population are called parameters and are usually denoted using Greek letters, while values calculated from a sample are called statistics and are usually denoted using Roman letters. Table 1.1.1 summarizes this distinction.

Table 1.1.1 Notation for Population Parameters versus Sample Statistics

Measure	Population Parameter	Sample Statistic
Size	N	n
Mean	μ (mu)	$\bar{x}$ (x-bar)
Variance	σ^2 (sigma-squared)	s^2
Standard Deviation	σ (sigma)	s

In real-world data analytics, most datasets that analysts work with -- such as a month of transaction logs, a batch of sensor readings, or a survey of 500 customers -- are samples rather than complete populations. This distinction becomes particularly important later in this unit when computing variance and standard deviation, since the formula used for a sample includes a small correction (known as Bessel's correction) that is not required when the entire population is available.

1.1.3 Measures of Central Tendency: The Mean

The arithmetic mean, commonly referred to simply as the mean or average, is the most widely used measure of central tendency. It represents the sum of all observations divided by the number of observations.

1.1.3.1 Arithmetic Mean of Ungrouped Data

For a set of n ungrouped observations x1, x2, x3, ..., xn, the arithmetic mean is calculated as:

$$\bar{x} = (\Sigma x) / n$$

where Σx denotes the sum of all observed values and n is the total number of observations. In the case of a population, the same formula is used but is denoted by μ = (Σx) / N, where N is the population size.

SGOU

Worked Example 1: The marks obtained by 10 students in a class test (out of 100) are: 45, 52, 52, 60, 63, 70, 72, 75, 80, 88. Calculate the mean.

$$\bar{x} = (45+52+52+60+63+70+72+75+80+88) / 10 = 657 / 10 = 65.7$$

The mean mark of the class is 65.7. This single number gives the analyst an immediate sense of the overall class performance, without needing to examine every individual score.

1.1.3.2 Arithmetic Mean of Grouped Data

When data is organized into a frequency distribution -- that is, values are grouped into class intervals along with the number of observations (frequency) falling in each interval -- the mean is calculated using the midpoint of each class:

$$\bar{x} = (\Sigma f \cdot x) / (\Sigma f)$$

where x is the midpoint of each class interval and f is the corresponding frequency.

Worked Example 2: The table below shows the daily sales (in ₹'000) recorded over 50 days by a small retail outlet.

Table 1.1.2 Frequency Distribution of Daily Sales

Sales Class (₹'000)	Frequency (f)	Midpoint (x)	f × x
10 -- 20	5	15	75
20 -- 30	12	25	300
30 -- 40	20	35	700
40 -- 50	9	45	405
50 -- 60	4	55	220
Total	Σf = 50		Σfx = 1700

$$\bar{x} = 1700 / 50 = 34 \text{ (i.e., ₹34,000)}$$

The average daily sale over the 50-day period is ₹34,000. This example will be revisited in the next two sections to compute the corresponding median and mode of the same dataset.

1.1.3.3 Properties and Limitations of the Mean

The arithmetic mean has several useful mathematical properties that make it convenient for further statistical analysis:

- The sum of the deviations of all observations from the mean is always zero -- that is, $\Sigma(x - \bar{x}) = 0$.

SGOU

- It uses every observation in the dataset, making it a comprehensive measure that reflects the entire distribution.
- It is algebraically well-defined and forms the basis for more advanced statistical measures such as variance and correlation.

However, the mean also has a significant limitation: it is highly sensitive to extreme values (outliers). A single unusually large or small observation can pull the mean away from the true 'typical' value of the dataset. For example, if one student in the class from Worked Example 1 had scored 5 marks instead of 45, the mean would drop noticeably, even though it would not accurately represent the performance of the remaining nine students. This sensitivity to outliers is one of the key reasons analysts also examine the median and mode alongside the mean, as discussed in the following sections.

1.1.4 Median

The median is the middle value of a dataset when all observations are arranged in ascending or descending order. Unlike the mean, the median divides the dataset into two equal halves -- exactly 50% of the observations lie below it and 50% lie above it.

For ungrouped data: The observations are first sorted. If the number of observations (n) is odd, the median is the middle value. If n is even, the median is the average of the two middle values.

Median = value at position (n+1)/2 [if n is odd]

Median = average of values at positions n/2 and (n/2 + 1) [if n is even]

Worked Example 3: Using the same 10 student marks from Worked Example 1, already sorted in ascending order: 45, 52, 52, 60, 63, 70, 72, 75, 80, 88.

Since n = 10 (even), the median is the average of the 5th and 6th values, which are 63 and 70.

Median = (63 + 70) / 2 = 66.5

For grouped data: The median class is the class interval in which the cumulative frequency first reaches or exceeds n/2. The exact median is then interpolated using the formula:

$$\text{Median} = l + [\ (\ (n/2) - cf\ )\ /\ f\ ] \times h$$

where l is the lower boundary of the median class, cf is the cumulative frequency of the class preceding the median class, f is the frequency of the median class, and h is the class width.

Worked Example 4: Continuing with the sales data from Table 1.1.2, the cumulative frequencies are 5, 17, 37, 46, and 50. Since n/2 = 25 falls between the cumulative frequency 17 and 37, the median class is 30 -- 40 (l = 30, cf = 17, f = 20, h = 10).

Median = 30 + [(25 − 17) / 20] × 10 = 30 + 4 = 34 (i.e., ₹34,000)

SGOU

The median daily sale of ₹34,000 is almost identical to the mean of ₹34,000 calculated earlier, which suggests that the sales data is fairly symmetrically distributed with no strong skew, a point that is examined more formally in Section 1.1.10.

The median is particularly useful when a dataset contains outliers or is skewed, since -- unlike the mean -- it is not distorted by a small number of extreme values. For this reason, figures such as median household income are commonly reported in place of the mean when describing income distributions, which tend to be strongly right-skewed.

1.1.5 Mode

The mode is the value that occurs most frequently in a dataset. Unlike the mean and median, the mode can be used with both numerical and categorical data, making it the only measure of central tendency applicable to non-numeric data such as favourite colour or preferred payment method.

A dataset may have one mode (unimodal), two modes (bimodal), more than two modes (multimodal), or no mode at all if every value occurs with equal frequency.

Worked Example 5: In the marks dataset 45, 52, 52, 60, 63, 70, 72, 75, 80, 88, the value 52 occurs twice while every other value occurs only once. Hence, the mode of this dataset is 52.

For grouped data: The modal class is the class interval with the highest frequency. The exact mode is then estimated using the formula:

$$\text{Mode} = l + [\ (f1 - f0) / (2f1 - f0 - f2)\ ] \times h$$

where l is the lower boundary of the modal class, f1 is the frequency of the modal class, f0 is the frequency of the class preceding the modal class, f2 is the frequency of the class following the modal class, and h is the class width.

Worked Example 6: In Table 1.1.2, the highest frequency is 20, occurring in the class 30 -- 40, which is therefore the modal class ($l = 30$, $f1 = 20$, $f0 = 12$, $f2 = 9$, $h = 10$).

$$\text{Mode} = 30 + [\ (20 - 12) / (2\times20 - 12 - 9)\ ] \times 10 = 30 + (8/19)\times10 \approx 34.21$$

(i.e., ≈ ₹34,210)

Once again, the modal daily sale of approximately ₹34,210 is very close to both the mean (₹34,000) and the median (₹34,000), reinforcing the earlier observation that this particular dataset is close to symmetric.

1.1.6 Comparing Mean, Median and Mode

Although the mean, median, and mode all attempt to describe the 'centre' of a dataset, they behave differently depending on the shape of the underlying distribution. Table 1.1.3 summarizes when each measure is most appropriate.

SGOU

Table 1.1.3 Comparison of Measures of Central Tendency

Measure	Best Used When	Key Limitation
Mean	Data is fairly symmetric with no major outliers; further statistical computation (e.g., variance) is required	Highly sensitive to outliers and extreme values
Median	Data is skewed or contains outliers (e.g., income, house prices)	Does not use the magnitude of every observation
Mode	Data is categorical, or the most 'typical' / frequently occurring value is of interest	May not exist, or may not be unique, in some datasets

The relationship between the three measures also provides a quick indication of the shape of a distribution. In a perfectly symmetric distribution, the mean, median, and mode coincide at the same central value. As a distribution becomes skewed, the three measures separate from one another and typically follow a predictable order, as illustrated in Figure 1.1.1.

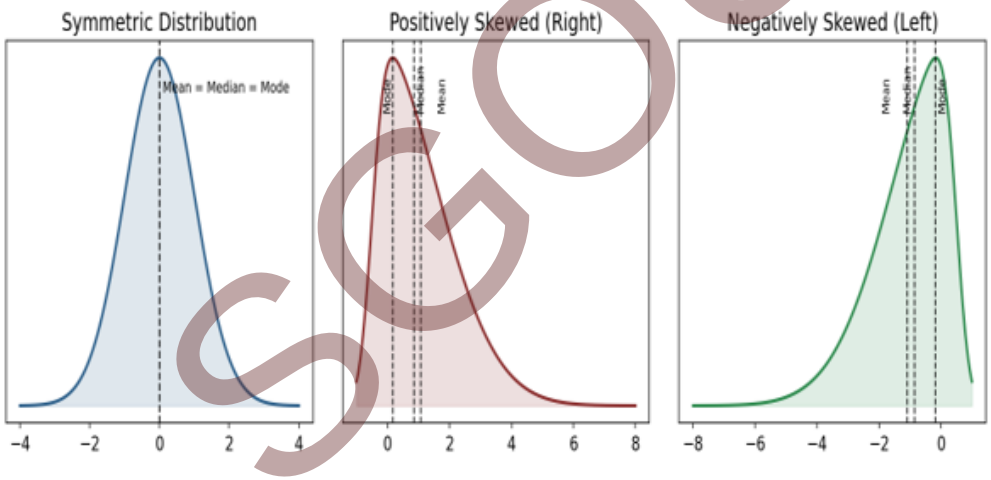


Fig.1.1.1 Position of Mean, Median and Mode in Symmetric and Skewed Distributions

As shown in Figure 1.1.1, in a positively (right) skewed distribution the order is typically Mode < Median < Mean, since the long right tail of large values pulls the mean upward more than the median. In a negatively (left) skewed distribution, the order reverses to Mean < Median < Mode. This relationship is formalized further in Section 1.1.10, which discusses skewness in detail.

An approximate empirical relationship, useful when only two of the three measures are known, is given by:

$$\text{Mean} - \text{Mode} \approx 3 \times (\text{Mean} - \text{Median})$$

SGOU

1.1.7 Measures of Dispersion: Range

While measures of central tendency describe the typical value of a dataset, they say nothing about how spread out or consistent the data is. Two datasets can have identical means but very different degrees of variability -- for example, the daily temperature of a coastal city and a desert city may both average 30°C, yet the desert city may swing between 15°C and 45°C while the coastal city stays between 27°C and 33°C. Measures of dispersion capture this important difference.

The range is the simplest measure of dispersion and is defined as the difference between the maximum and minimum values in a dataset:

$$\text{Range} = \text{Maximum value} - \text{Minimum value}$$

Worked Example 7: In the marks dataset (45, 52, 52, 60, 63, 70, 72, 75, 80, 88), the maximum value is 88 and the minimum value is 45.

$$\text{Range} = 88 - 45 = 43$$

A related measure, the coefficient of range, expresses this spread as a proportion of the total, making it easier to compare the variability of two datasets measured on different scales:

$$\text{Coefficient of Range} = (\text{Max} - \text{Min}) / (\text{Max} + \text{Min})$$

For the example above, the coefficient of range is

$(88 - 45) / (88 + 45) = 43 / 133 \approx 0.323$.

The range is easy to calculate and interpret, but it has a major limitation: it depends only on the two extreme values and completely ignores how the remaining observations are distributed between them. A single unusually high or low value can therefore make the range highly misleading. This limitation motivates the need for more robust measures of dispersion, such as variance and standard deviation, discussed next.

1.1.8 Variance

Variance measures the average squared deviation of each observation from the mean. Unlike the range, variance takes every single observation in the dataset into account, making it a far more comprehensive measure of spread.

Population Variance: When data for the entire population is available, the population variance is calculated as:

$$\sigma^2 = \Sigma (x - \mu)^2 / N$$

Sample Variance: When working with a sample, the sample variance uses a slightly modified formula, dividing by $(n - 1)$ instead of n:

$$s^2 = \Sigma (x - \bar{x})^2 / (n - 1)$$

The use of $(n - 1)$ rather than n in the sample variance formula is known as Bessel's correction. A sample tends to slightly underestimate the true variability of the population

SGOU

from which it is drawn, and dividing by the smaller number (n − 1) compensates for this, producing a statistic that, on average, more accurately estimates the population variance.

Worked Example 8: Using the marks dataset (mean $\bar{x} = 65.7$), the deviations of each mark from the mean are squared and summed as shown in Table 1.1.4.

Table 1.1.4 Computation of Squared Deviations for the Marks Dataset

x	45	52	52	60	63	70
$(x - \bar{x})$	-20.7	-13.7	-13.7	-5.7	-2.7	4.3
$(x - \bar{x})^2$	428.49	187.69	187.69	32.49	7.29	18.49
x	72	75	80	88	$\Sigma x = 657$	$n = 10$
$(x - \bar{x})$	6.3	9.3	14.3	22.3	$\Sigma(x-\bar{x}) = 0$	
$(x - \bar{x})^2$	39.69	86.49	204.49	497.29	$\Sigma (x - \bar{x})^2 = 1690.10$	

Note, as expected from the properties discussed in Section 1.1.3.3, that the deviations from the mean sum to zero. Treating this dataset as the full population (N = 10):

$$\sigma^2 = 1690.10 / 10 = 169.01$$

If instead this dataset is treated as a sample drawn from a larger population of students (n = 10), the sample variance is:

$$s^2 = 1690.10 / 9 \approx 187.79$$

Although variance provides a mathematically sound measure of spread, it is expressed in squared units (for example, 'marks squared'), which makes it difficult to interpret directly alongside the original data. This is the motivation for the standard deviation, discussed in the next section.

1.1.9 Standard Deviation

The standard deviation is simply the square root of the variance. By undoing the squaring operation, the standard deviation is expressed in the same units as the original data, making it far easier to interpret and communicate.

$$\sigma = \sqrt{(\sigma^2)} \quad \text{[Population]} \qquad s = \sqrt{(s^2)} \quad \text{[Sample]}$$

Worked Example 9: Continuing from Worked Example 8:

$$\sigma = \sqrt{169.01} \approx 13.0 \text{ marks} \quad \text{(Population Standard Deviation)}$$

$$s = \sqrt{187.79} \approx 13.7 \text{ marks} \quad \text{(Sample Standard Deviation)}$$

A standard deviation of approximately 13 marks tells the analyst that, on average, individual student scores deviate from the class mean of 65.7 by about 13 marks in either direction -- a much more intuitive statement than '169.01 marks squared'.

SGOU

The standard deviation is closely linked to the shape of a normal (bell-shaped) distribution through the empirical rule, which states that for approximately normally distributed data:

- about 68% of observations fall within one standard deviation of the mean ($\mu \pm 1\sigma$);
- about 95% of observations fall within two standard deviations of the mean ($\mu \pm 2\sigma$);
- about 99.7% of observations fall within three standard deviations of the mean ($\mu \pm 3\sigma$).

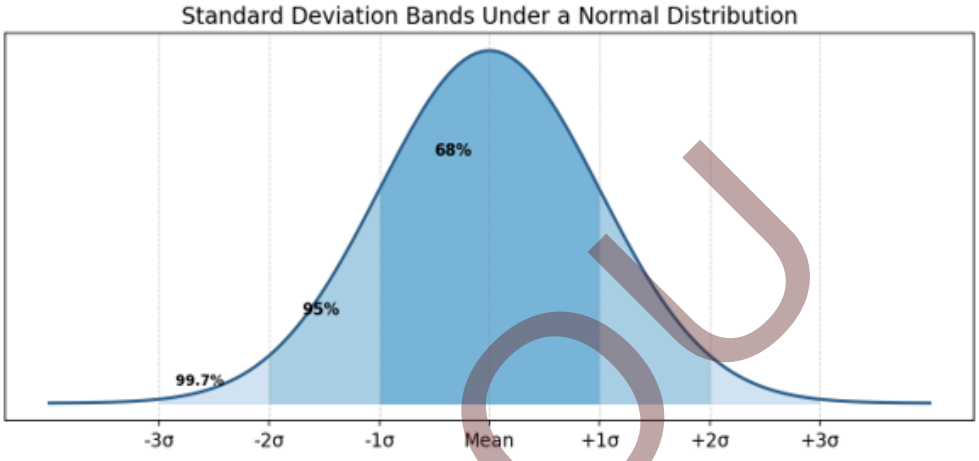


Fig.1.1.2 The Empirical Rule -- Standard Deviation Bands Under a Normal Distribution

As illustrated in Figure 1.1.2, standard deviation gives analysts a way to judge whether a particular observation is typical or unusually extreme. In data analytics, this principle underlies common outlier-detection techniques, where any observation lying beyond roughly three standard deviations from the mean is flagged for further investigation.

A related measure, the coefficient of variation (CV), expresses the standard deviation as a percentage of the mean, allowing the variability of two datasets with different units or scales to be compared directly:

$$CV = (\sigma / \mu) \times 100\%$$

For the marks dataset, CV = (13.0 / 65.7) × 100 ≈ 19.8%, indicating a moderate degree of relative variability in student performance.

1.1.10 Skewness

Skewness measures the asymmetry of a data distribution around its mean. A distribution can be symmetric, positively (right) skewed, or negatively (left) skewed, and identifying this shape is essential for choosing the right descriptive measure and the right statistical technique for further analysis.

SGOU

- Symmetric distribution: the data is evenly balanced on both sides of the centre; Mean = Median = Mode.
- Positive (right) skew: a long tail extends towards higher values; typically Mode < Median < Mean.
- Negative (left) skew: a long tail extends towards lower values; typically Mean < Median < Mode.

One commonly used measure is Pearson's coefficient of skewness, which can be computed in two equivalent forms:

$$\text{Skewness (1st coefficient)} = (\text{Mean} - \text{Mode}) / \text{SD}$$

$$\text{Skewness (2nd coefficient)} = 3 \times (\text{Mean} - \text{Median}) / \text{SD}$$

A positive value indicates right (positive) skew, a negative value indicates left (negative) skew, and a value close to zero indicates an approximately symmetric distribution.

Worked Example 10: Consider the monthly income (in ₹'000) of 9 employees in a small firm: 18, 20, 20, 21, 22, 23, 24, 25, 95. This dataset includes one very high earner, whose income is likely to pull the mean upward.

$$\text{Mean} = (18+20+20+21+22+23+24+25+95)/9 = 268/9 \approx 29.78$$

$$\text{Median (5th value of 9, sorted)} = 22 \qquad \text{Mode} = 20$$

$$\text{Standard Deviation (population)} \approx 23.15$$

Applying the second coefficient of skewness:

$$\text{Skewness} = 3 \times (29.78 - 22) / 23.15 = 3 \times 7.78 / 23.15 \approx 1.01$$

The positive value of approximately 1.01, together with the ordering Mode (20) < Median (22) < Mean (29.78), confirms that this income data is strongly positively (right) skewed -- a single very high income has pulled the mean well above the median and mode, exactly the pattern illustrated for the right-skewed curve in Figure 1.1.1.

Recognizing skewness is critically important in data analytics. For strongly skewed data, such as income or house prices, the median is generally a more representative measure of the 'typical' value than the mean, and many statistical and machine-learning techniques (which assume roughly symmetric, normally distributed data) may require the data to be transformed -- for example using a logarithmic transformation -- before being used reliably.

1.1.11 Scaling and Shifting of Data

Data analysts frequently need to transform raw data before analysis -- for example, converting units, aligning different scales for comparison, or preparing data for machine learning algorithms. Two of the most fundamental transformations are shifting and scaling.

SGOU

Shifting (Adding or Subtracting a Constant)

Shifting refers to adding (or subtracting) the same constant value, c, to every observation in a dataset. If a constant c is added to every value:

- the new mean increases by exactly c (new mean = old mean + c);
- the standard deviation remains completely unchanged, since the relative spacing between observations does not change.

Scaling (Multiplying by a Constant)

Scaling refers to multiplying every observation in a dataset by the same constant value, k. If every value is multiplied by k:

- the new mean is scaled by the same factor k (new mean = k × old mean);
- the new standard deviation is also scaled by the factor |k| (new SD = |k| × old SD), since spread changes proportionally with the values themselves.

Worked Example 11: Consider the temperature readings (in °C) recorded on five consecutive days: 20, 22, 25, 28, 30.

Original Mean = 125/5 = 25°C Original SD ≈ 3.69°C

Shifting: If every reading is increased by 5°C (perhaps to correct a calibration error), the new readings become 25, 27, 30, 33, 35.

New Mean = 25 + 5 = 30°C New SD = 3.69°C (unchanged)

Scaling: If every original reading (in °C) is instead doubled, for example to compare against a scale twice as sensitive, the new readings become 40, 44, 50, 56, 60.

New Mean = 2 × 25 = 50 New SD = 2 × 3.69 ≈ 7.38

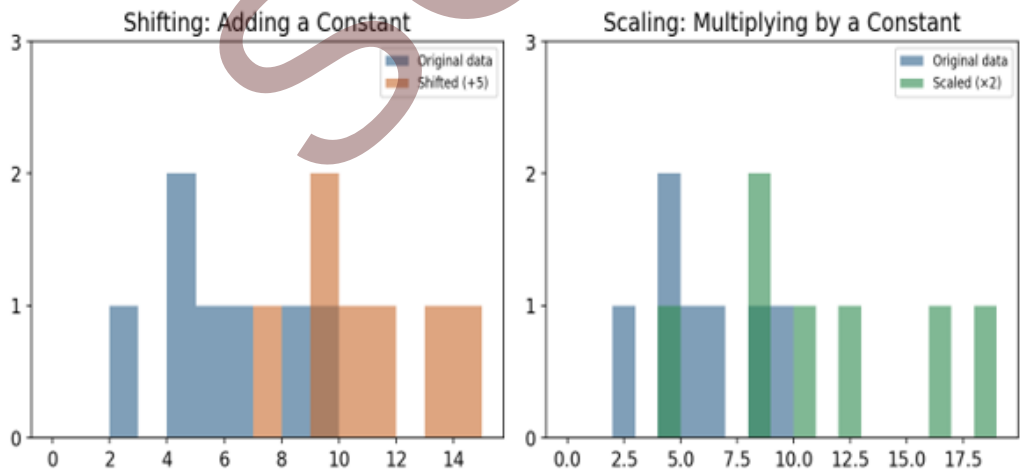


Fig.1.1.3 Effect of Shifting (adding a constant) and Scaling (multiplying by a constant) on a Dataset

SGOU

As shown in Figure 1.1.3, shifting moves the entire distribution left or right without changing its shape or spread, whereas scaling stretches or compresses the distribution, changing both its centre and its spread proportionally.

Standardization (Z-scores): A particularly important combined application of shifting and scaling is standardization, which transforms data to have a mean of 0 and a standard deviation of 1. Each observation is converted into a z-score using the formula:

$$z = (x - \mu) / \sigma$$

For example, the temperature reading of 28°C from Worked Example 11 (original mean 25°C, SD 3.69°C) has a z-score of $z = (28 - 25) / 3.69 \approx 0.81$, meaning this observation lies about 0.81 standard deviations above the mean. Standardization is used extensively in data analytics and machine learning to bring variables measured on different scales -- such as age in years and income in rupees -- onto a common footing before further analysis.

1.1.12 Applications of Descriptive Statistics in Data Analytics

The concepts introduced in this unit are not merely theoretical exercises -- they are applied constantly throughout the real-world data analytics pipeline:

- **Exploratory Data Analysis (EDA):** Analysts routinely begin any project by generating summary statistics (mean, median, mode, standard deviation) for every variable in a dataset -- a step that tools such as Python's pandas library automate through functions like describe().
- **Outlier Detection:** Observations lying beyond a chosen number of standard deviations from the mean, or outside a range defined using the interquartile range, are commonly flagged as potential outliers or data-entry errors requiring further investigation.
- **Feature Scaling for Machine Learning:** Many machine learning algorithms perform better, or converge faster, when input features are standardized (using z-scores) or scaled to a common range, directly applying the scaling and shifting concepts introduced in Section 1.1.11.
- **Business Reporting and Dashboards:** Key performance indicators reported on business dashboards -- average order value, median delivery time, most frequently purchased product -- are direct applications of the measures of central tendency covered in this unit.
- **Quality Control:** In manufacturing and process industries, control charts use the mean and standard deviation of a process to determine whether it is operating within acceptable limits, triggering an alert whenever a measurement drifts too far from the expected value.

As the course progresses into subsequent units and blocks, these foundational descriptive measures will be used repeatedly -- from detecting anomalies and normalizing data in Unit 2, to modelling data using probability distributions in Unit 3, and eventually to building interactive visualizations and dashboards in later blocks of this course.

SGOU

Recap

- Descriptive statistics summarizes and organizes data, while inferential statistics draws conclusions about a population from a sample.
- A population includes every unit of interest, while a sample is a representative subset; population parameters (μ, σ) and sample statistics ($\bar{x}$, s) use different notation and, for variance, slightly different formulas.
- The mean is the arithmetic average and uses every observation, but is sensitive to outliers; the median is the middle value and is robust to outliers; the mode is the most frequent value and is the only measure usable with categorical data.
- The range is the simplest measure of dispersion but considers only the two extreme values.
- Variance measures the average squared deviation from the mean; the sample variance formula divides by $(n - 1)$ rather than n, a correction known as Bessel's correction.
- Standard deviation is the square root of variance, expressed in the same units as the original data, and underlies the empirical rule (68%-95%-99.7%) for normally distributed data.
- Skewness describes the asymmetry of a distribution; positive skew follows the order Mode < Median < Mean, while negative skew follows Mean < Median < Mode.
- Shifting a dataset (adding a constant) changes the mean but leaves the standard deviation unchanged; scaling a dataset (multiplying by a constant) changes both the mean and the standard deviation proportionally.
- Standardization (z-scores) combines shifting and scaling to transform any dataset to a mean of 0 and a standard deviation of 1, and is widely used in feature preparation for machine learning.

SGOU

Objective Type Questions

1. What is the primary difference between descriptive statistics and inferential statistics?
2. What is a population in statistics?
3. Which symbol is used to denote the population mean?
4. How is the sample mean calculated for ungrouped data?
5. What is the median of a dataset?

6. Which measure of central tendency can be used with categorical data?
7. What does the range of a dataset measure?
8. Why is (n − 1) used in the denominator of the sample variance formula instead of n?
9. What is the relationship between variance and standard deviation?
10. According to the empirical rule, what percentage of data lies within one standard deviation of the mean in a normal distribution?
11. What does a positive value of Pearson's coefficient of skewness indicate?
12. In a positively skewed distribution, what is the typical order of the mean, median, and mode?
13. What happens to the standard deviation of a dataset when a constant is added to every value?
14. What happens to the mean and standard deviation when every value in a dataset is multiplied by a constant?
15. What is the formula used to calculate a z-score?

Answers

1. Descriptive statistics summarizes data; inferential statistics draws conclusions about a population from a sample
2. The complete set of items or individuals under study
3. μ (mu)
4. $\bar{x} = (\Sigma x) / n$
5. The middle value of a dataset when arranged in order
6. Mode
7. The difference between the maximum and minimum values
8. To correct for the tendency of a sample to underestimate population variability (Bessel's correction)
9. Standard deviation is the square root of variance
10. Approximately 68%
11. Positive (right) skew
12. Mode < Median < Mean

SGOU

13. It remains unchanged

14. Both are scaled by the same constant

15. $z = (x - \mu) / \sigma$

Assignments

1. A dataset of 12 monthly electricity bills (in ₹) is given: 850, 900, 920, 940, 960, 980, 1000, 1020, 1050, 1100, 1150, 3200. Calculate the mean and median of this dataset, and explain, with reasoning, which measure better represents a 'typical' monthly bill.
2. For the frequency distribution of ages (in years) of 60 members of a fitness club -- 15-25: 8, 25-35: 18, 35-45: 22, 45-55: 9, 55-65: 3 -- calculate the mean, median, and mode using the grouped-data formulas discussed in this unit.
3. Explain, with a suitable real-world example of your own, why the standard deviation is generally preferred over the range as a measure of dispersion.
4. A dataset has a mean of 40 and a standard deviation of 8. If every value in the dataset is first increased by 10 and then multiplied by 2, calculate the new mean and new standard deviation. Show each step separately.
5. Using the income dataset from Worked Example 10 in Section 1.1.10, discuss why the median rather than the mean would be a more appropriate summary measure to report to a policymaker studying typical employee earnings.

Suggested Reading

1. Agarwal, B. L. (2013). Basic Statistics. New Age International.
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.
3. Spiegel, M. R., & Stephens, L. J. (2017). Schaum's Outline of Statistics. McGraw-Hill Education.

SGOU

Reference

1. Agarwal, B. L. (2013). Basic Statistics. New Age International.
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

SGOU



Anomalies and Normalization

Learning Outcomes

After completion of this unit, learner will be able to:

- explain what constitutes an anomaly (outlier) in a dataset and why detecting anomalies matters in data analytics
- distinguish between point, contextual, and collective anomalies
- apply the z-score method and the interquartile range (IQR) method to detect outliers in numeric data
- interpret a box plot and use it to visually identify outliers
- describe common strategies for handling anomalies once detected
- explain the need for data normalization and apply min-max normalization, z-score standardization, decimal scaling, and robust scaling to a dataset
- choose an appropriate normalization technique based on the characteristics of a given dataset

SGOU

Prerequisite

Consider an e-commerce analyst reviewing a spreadsheet of daily order counts when one row stops her scrolling entirely: a single day with almost ten times the usual number of orders. Was it a flash sale, a system glitch, or something else worth investigating? Before she can even ask that question sensibly, she needs the same tools introduced in Unit 1 -- a sense of what the 'typical' day looks like (the mean and median), and how much the numbers usually vary from one day to the next (the standard deviation).

This unit picks up exactly where that instinct leaves off, turning 'that number looks a bit odd' into a precise, repeatable rule for deciding how odd is odd enough to matter. As you follow the worked examples ahead, it will help to already be comfortable computing a mean and a standard deviation, and to be able to arrange a short list of numbers in order -- a skill used repeatedly when locating quartiles and medians.

Keywords

Anomaly, Outlier, Point Anomaly, Contextual Anomaly, Collective Anomaly, Z-score, Interquartile Range (IQR), Box Plot, Normalization, Min-Max Scaling, Standardization, Decimal Scaling, Robust Scaling.

Discussion

1.2.1 Introduction to Data Quality and the Need for Anomaly Detection

Real-world data is rarely perfect. It is collected from sensors that occasionally malfunction, entered manually by people who sometimes make typing errors, and generated by systems that can behave unpredictably during rare events. As a result, almost every dataset a data analyst works with contains at least a few values that do not fit the general pattern of the rest of the data. Such values are referred to as anomalies, or more commonly, outliers.

An anomaly can be formally defined as an observation that deviates so significantly from the rest of the data that it raises suspicion of being generated by a different process altogether -- whether that process is a data-entry error, a sensor malfunction, a fraudulent transaction, or a genuinely rare but real event.

Detecting anomalies is a critical step in the data analytics life cycle for two related but distinct reasons. First, some anomalies represent errors or noise that, if left untreated, can distort summary statistics (as seen in Unit 1, where a single extreme value noticeably shifted the mean) and mislead any model built on top of the data. Second, some anomalies are themselves the object of interest -- for example, in fraud detection, network intrusion monitoring, and medical diagnosis, the anomaly is precisely the pattern the analyst is trying to find. In both cases, the analyst must first be able to identify which observations are unusual before deciding whether to correct, remove, or specifically investigate them.

It is worth emphasizing that anomaly detection sits at the intersection of the descriptive statistics covered in Unit 1 and the data-preparation stage of the analytics life cycle. Every technique introduced later in this unit -- the z-score method and the interquartile range method -- is, in essence, a formal rule for deciding how far is 'too far' from the centre of the data before an observation is considered suspicious. Without such a rule, analysts would be left relying purely on subjective judgement, which does not scale well to the large datasets typically encountered in modern data analytics, where a human being cannot realistically inspect every individual value by eye.

1.2.2 Types of Anomalies

Anomalies are commonly classified into three broad categories, based on how the unusual behaviour manifests in the data.

SGOU

- **Point Anomalies:** A single data point that is far removed from the rest of the dataset. For example, a single transaction of ₹5,00,000 in a dataset where most transactions are between ₹500 and ₹5,000 is a clear point anomaly.

- **Contextual Anomalies:** A value that would be considered normal in one context but abnormal in another. For example, a temperature reading of 35°C is unremarkable in summer but highly unusual in the middle of winter -- the same value is anomalous only relative to its surrounding context (in this case, the season).

- **Collective Anomalies:** A group of related data points that, taken together, deviate from the expected pattern, even though no single point in the group is unusual on its own. For example, a short but sustained sequence of unusually low heart-rate readings recorded one after another may indicate a developing medical issue, even if none of the individual readings alone would be flagged as abnormal.

Most of the techniques introduced in this unit -- the z-score method and the interquartile range method -- are designed primarily to detect point anomalies, since these are the most common type encountered in typical business and analytics datasets. However, recognizing that contextual and collective anomalies exist is important, since these often require more advanced techniques, such as time-series analysis or machine learning-based anomaly detection, which are beyond the scope of this introductory unit but build directly on the concepts introduced here.

Table 1.2.1 The Three Types of Anomalies with Illustrative Examples

Anomaly Type	Description	Typical Example
Point Anomaly	A single observation far removed from the rest of the data	One transaction of ₹5,00,000 among transactions of ₹500--₹5,000
Contextual Anomaly	A value that is normal in one context but abnormal in another	A temperature of 35°C recorded in winter rather than summer
Collective Anomaly	A group of related observations that together deviate from the expected pattern	A short sequence of unusually low heart-rate readings recorded consecutively

1.2.3 Causes of Anomalies in Real-World Data

Understanding why anomalies occur helps an analyst decide how they should be handled. Common causes include:

- **Data Entry Errors:** A missing decimal point, an extra digit, or a misplaced comma can turn a normal value into an extreme one -- for example, entering an age of 250 instead of 25.

SGOU

- **Sensor or Equipment Malfunction:** Faulty hardware, such as a temperature sensor with a loose connection, can occasionally produce readings that are physically impossible.
- **Measurement or Sampling Errors:** Inconsistent measurement procedures, or a sample that accidentally includes a unit that does not belong to the population of interest, can introduce anomalies.
- **Genuine Rare Events:** Not every anomaly is an error. A sudden but real spike in website traffic following a viral social media post, or an unusually large but legitimate transaction, is a true reflection of an unusual real-world event.

This distinction is important: an anomaly caused by a data-entry error should typically be corrected or removed, whereas an anomaly reflecting a genuine rare event may need to be retained and studied rather than discarded. Simply deleting every extreme value without investigation risks throwing away some of the most informative data points in the dataset.

Consider, for example, an e-commerce company reviewing its daily order data and finding a single day on which the number of orders was ten times the usual average. Before treating this as noise, the analyst should find the cause. It could be one of three things. It could be a logging bug that duplicated every order (a data error, best corrected or removed). It could be a flash sale held on that day (a genuine event, best kept and highlighted in reports). It could be a bot attack placing fraudulent orders (a genuine anomaly, to be kept but flagged separately for fraud investigation, instead of being mixed into ordinary sales figures). The correct handling strategy is different in each of these three scenarios, even though the raw anomalous value -- an unusually high order count -- looks identical in every case.

1.2.4 Statistical Methods for Anomaly Detection -- The Z-score Method

The z-score method, introduced as a standardization technique in Unit 1, can also be used directly as an anomaly-detection tool. Recall that the z-score of an observation measures how many standard deviations that observation lies away from the mean:

$$z = (x - \mu) / \sigma$$

A commonly used rule of thumb is that any observation with an absolute z-score greater than 3 (that is, $|z| > 3$) is flagged as a potential outlier, since under a normal distribution fewer than 0.3% of observations are expected to lie beyond three standard deviations from the mean, as established by the empirical rule discussed in Unit 1.

Worked Example 1: A website recorded the following number of daily visitors over 10 days: 220, 235, 210, 225, 240, 215, 230, 225, 980, 238.

$$\text{Mean } (\mu) = 3018 / 10 = 301.8 \qquad \text{Standard Deviation } (\sigma) \approx 226.25$$

Computing the z-score for the ninth day's reading of 980 visitors:

$$z = (980 - 301.8) / 226.25 \approx 3.00$$

SGOU

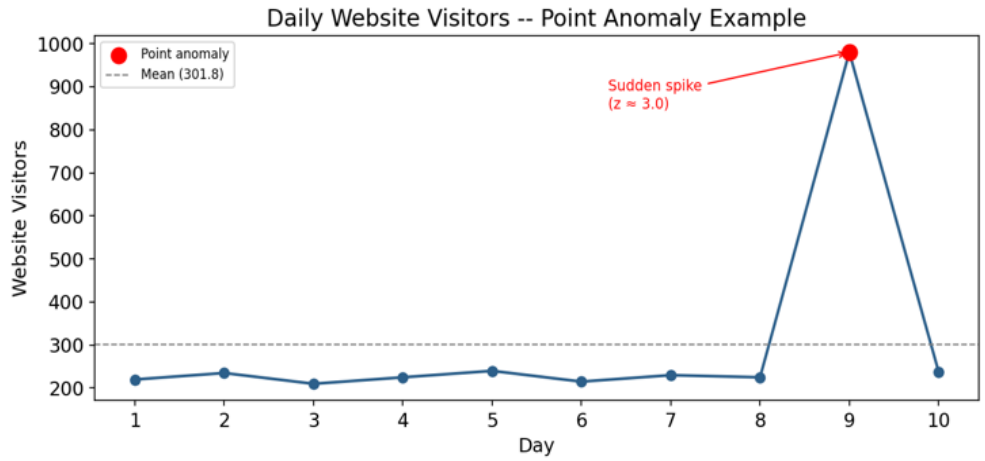


Fig. 1.2.1 Daily Website Visitors -- a Point Anomaly Detected Using the Z-score Method

As shown in Figure 1.2.1, the reading of 980 visitors on day 9 stands out sharply from the otherwise stable pattern of 210 to 240 visitors per day. With a z-score of approximately 3.00, this observation sits right at the common threshold used to flag potential outliers, and would prompt the analyst to investigate further -- perhaps the spike was caused by a promotional campaign, a bot attack, or a data logging error.

The z-score method is simple to compute and works well when the underlying data is approximately normally distributed. However, it has an important limitation: because the mean and standard deviation used in the formula are themselves calculated from the same data that includes the outlier, a very extreme value can inflate the standard deviation and, in some cases, mask its own anomalous nature. This limitation motivates the interquartile range method discussed next, which relies on the median and quartiles instead of the mean and standard deviation, making it more robust to extreme values.

1.2.5 The Interquartile Range (IQR) Method

Quartiles divide an ordered dataset into four equal parts. The first quartile (Q1) is the value below which 25% of the data falls, the second quartile (Q2) is simply the median (50% of the data falls below it), and the third quartile (Q3) is the value below which 75% of the data falls. The interquartile range (IQR) is the difference between the third and first quartiles and represents the spread of the middle 50% of the data:

$$IQR = Q3 - Q1$$

Since Q1 and Q3 are based on the position of values in a sorted dataset rather than on the mean, they are far less sensitive to extreme values than the mean and standard deviation used in the z-score method. Using the IQR, upper and lower 'fences' are constructed to flag outliers:

$$\text{Lower Fence} = Q1 - 1.5 \times IQR \qquad \text{Upper Fence} = Q3 + 1.5 \times IQR$$

Any observation falling below the lower fence or above the upper fence is flagged as a potential outlier.

SGOU

Worked Example 2: The examination scores of 12 students, arranged in ascending order, are: 55, 58, 60, 62, 64, 65, 67, 68, 70, 72, 75, 98.

$$Q1 = 60.5 \qquad Q3 = 71.5 \qquad IQR = 71.5 - 60.5 = 11$$

$$\text{Lower Fence} = 60.5 - (1.5 \times 11) = 60.5 - 16.5 = 44$$

$$\text{Upper Fence} = 71.5 + (1.5 \times 11) = 71.5 + 16.5 = 88$$

Since the highest score, 98, lies above the upper fence of 88, it is flagged as an outlier under the IQR method. All the remaining scores lie comfortably between the fences and are not flagged.

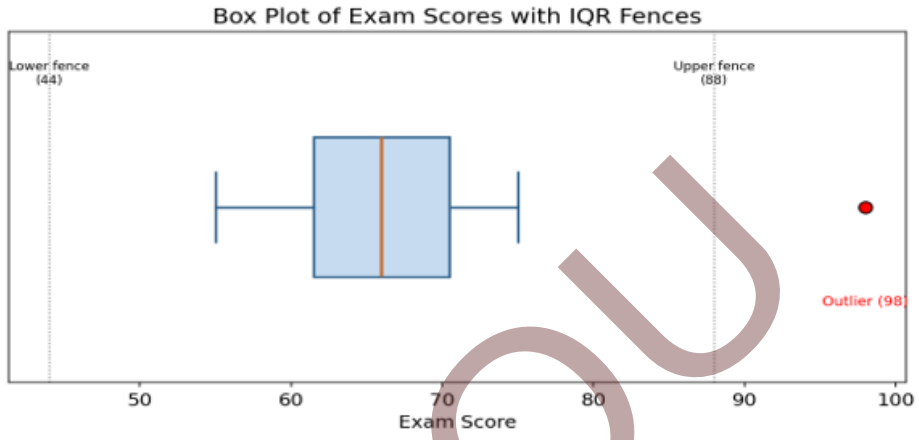


Fig. 1.2.2 Box Plot of Examination Scores Showing Q1, Q3, the IQR Fences, and the Detected Outlier

As illustrated in Figure 1.2.2, the box plot (also called a box-and-whisker plot) is the standard visual representation of the IQR method. The box spans from Q1 to Q3, the line inside the box marks the median, the whiskers extend to the fences (or to the most extreme non-outlying value within the fences), and any points beyond the whiskers are plotted individually as potential outliers -- exactly as seen with the score of 98 in this example.

1.2.6 Visualization-Based Detection

While the z-score and IQR methods provide precise numerical rules, visualizing data is often the fastest way for an analyst to build an initial sense of where anomalies might exist, and to sanity-check the results of a numerical method. Two visualizations are especially useful for this purpose.

- **Box Plots:** As discussed in Section 1.2.5, a box plot immediately reveals the spread of the middle 50% of the data (the box), the median, and any points lying beyond the whiskers, making outliers visually obvious at a glance.
- **Scatter Plots and Line Charts:** Plotting data points against time (as in Figure 1.2.1) or against another variable makes point anomalies stand out as

SGOU

points that are clearly separated from the general cloud or trend of the rest of the data. This is particularly useful for detecting contextual anomalies, since the surrounding pattern -- for example, a seasonal trend in a time series -- provides the context against which a value can be judged unusual.

In practice, data analysts typically combine visual inspection with a statistical method: a quick box plot or scatter plot is used to form an initial hypothesis about which observations look unusual, and the z-score or IQR method is then applied to confirm the finding with an objective, repeatable rule.

1.2.7 Handling Anomalies

Once an anomaly has been identified, the analyst must decide how to handle it. The right choice depends heavily on the cause of the anomaly, as discussed in Section 1.2.3, and on the goal of the analysis. Common strategies include:

- **Removal:** If an anomaly is confirmed to be a data-entry error or an impossible value (for example, a negative age), the safest approach is often to remove that observation entirely from the dataset.
- **Capping / Winsorization:** Rather than removing an extreme value outright, it can be replaced (capped) with the nearest acceptable boundary value -- for example, replacing any value above the upper fence with the upper fence value itself. This retains the observation while limiting its distorting influence on subsequent calculations.
- **Transformation:** Applying a mathematical transformation, such as a logarithmic transformation, can reduce the influence of extreme values by compressing the scale of the data, which is particularly useful for strongly right-skewed data such as income or transaction amounts (as seen in the skewness example in Unit 1).
- **Imputation:** If an anomaly is judged to be a missing or corrupted value rather than a real observation, it can be replaced with an estimated value, such as the mean, median, or a value predicted from other related variables.
- **Retention for Further Study:** If the anomaly is judged to be a genuine and important rare event -- such as a fraudulent transaction or a critical equipment failure -- it should be retained and flagged for specific investigation, rather than treated as noise to be removed.

There is no single correct strategy that applies to every situation; a skilled data analyst always considers the likely cause of an anomaly and the purpose of the analysis before deciding how to treat it, since handling anomalies carelessly can either discard valuable information or leave misleading values in the dataset.

1.2.8 Introduction to Data Normalization

Once a dataset has been cleaned of clear errors and anomalies have been appropriately handled, it is often still not ready for direct analysis or modelling, because different variables (features) in a dataset are frequently measured on very different scales. For example, in a dataset used to predict house prices, the 'area' of a house might range from

SGOU

500 to 5,000 square feet, while the 'number of bedrooms' might range only from 1 to 5, and the 'price' itself might run into lakhs of rupees.

If such data is used directly in many statistical or machine learning techniques, the variable with the largest numerical range (area, in this example) can dominate the analysis simply because of its scale, not because it is actually more important. Normalization refers to the family of techniques used to rescale numeric data onto a common, comparable scale, without distorting the relative relationships between values.

Normalization is especially important for:

- Machine learning algorithms that rely on distance calculations (such as k-nearest neighbours and k-means clustering), where features on larger scales would otherwise dominate the distance metric.
- Gradient-based optimization algorithms (used in training many machine learning models), which typically converge faster and more reliably when input features are on a similar scale.
- Fair visual and statistical comparison of variables that are naturally measured in different units, such as comparing 'income in rupees' with 'years of education'.

The remainder of this unit introduces four widely used normalization techniques: min-max normalization, z-score standardization, decimal scaling, and robust scaling.

1.2.9 Min-Max Normalization

Min-max normalization rescales data so that every value falls within a fixed range, most commonly between 0 and 1. It is calculated as:

$$x' = (x - \min) / (\max - \min)$$

where min and max are the smallest and largest values in the dataset (or feature) being normalized. After transformation, the smallest original value maps to 0 and the largest maps to 1, with every other value falling proportionally in between.

Worked Example 3: The prices (in ₹ lakh) of five houses are: 25, 32, 40, 45, 60. Apply min-max normalization.

$$\min = 25 \qquad \max = 60 \qquad (\max - \min) = 35$$

Table 1.2.2 Min-Max Normalization of House Price Data

Original Price (x)	25	32	40	45	60
Normalized (x')	0.0000	0.2000	0.4286	0.5714	1.0000

Min-max normalization is intuitive and guarantees that every value lies within a known, fixed range, which is convenient for many applications. However, it has a limitation. It depends directly on the minimum and maximum values, so one extreme outlier can squeeze the rest of the normalized data into a very narrow band close to zero.

This is why the anomaly-detection techniques discussed earlier in this unit should be applied before normalization.

1.2.10 Z-score Standardization

Z-score standardization, introduced in Unit 1 and again in Section 1.2.4 as an anomaly-detection tool, is equally important as a normalization technique. It rescales data to have a mean of 0 and a standard deviation of 1, using the same formula as before:

$$x' = (x - \mu) / \sigma$$

Unlike min-max normalization, z-score standardization does not force values into a fixed bounded range; instead, it expresses every value in terms of how many standard deviations it lies from the mean. This makes it more robust to the presence of a small number of outliers than min-max normalization, since a single extreme value affects the mean and standard deviation less severely than it affects the minimum or maximum used in min-max scaling.

Worked Example 4: Using the marks dataset from Unit 1 (mean = 65.7, population standard deviation ≈ 13.0), the standardized value for a student who scored 80 is:

$$z = (80 - 65.7) / 13.0 \approx 1.10$$

This standardized score of 1.10 indicates that the student's mark lies about 1.10 standard deviations above the class average -- a value that can now be directly compared with standardized scores from a completely different test, dataset, or scale, which would not have been possible using the raw marks alone.

1.2.11 Decimal Scaling

Decimal scaling normalizes data by moving the decimal point of each value, based on the maximum absolute value in the dataset. It is defined as:

$$x' = x / 10^{j}$$

where j is the smallest integer such that the maximum absolute normalized value is less than 1.

Worked Example 5: A dataset of transaction amounts (in ₹) is: 700, 850, 990, 320. The maximum absolute value is 990.

Since 990 has 3 digits, the smallest power of 10 that brings every normalized value below 1 in magnitude is $10^3 = 1000$ (j = 3), because 990 / 1000 = 0.99, which is less than 1.

Table 1.2.3 Decimal Scaling of Transaction Amounts (Illustrative, j = 3)

Original (x)	700	850	990
Normalized (x' = x/1000)	0.70	0.85	0.99

SGOU

Decimal scaling is simple to compute and easy to interpret, since it merely shifts the decimal point without altering the relative pattern of the digits in each value. However, it is used less frequently in modern data analytics workflows than min-max normalization or z-score standardization, and is most useful for quick, approximate scaling of large monetary or count-based figures.

1.2.12 Robust Scaling

Robust scaling addresses the main weakness of min-max normalization -- its sensitivity to outliers -- by using the median and the interquartile range (both introduced in Section 1.2.5) instead of the minimum, maximum, mean, or standard deviation:

$$x' = (x - \text{median}) / \text{IQR}$$

Because the median and IQR are based on the position of values in an ordered dataset rather than on their exact magnitude, robust scaling is far less influenced by a small number of extreme values than either min-max normalization or z-score standardization.

Worked Example 6: Using the examination scores from Worked Example 2 (median = 66, IQR = 11), the robust-scaled value for the outlying score of 98 is:

$$x' = (98 - 66) / 11 = 32 / 11 \approx 2.91$$

For comparison, the robust-scaled value for a typical score of 55 is $x' = (55 - 66) / 11 = -1.00$. Note that 98 is a large outlier, but the median and IQR used in robust scaling are barely affected by it. The mean, standard deviation, minimum and maximum used in the other techniques are affected by it. This is why robust scaling is the preferred technique whenever a dataset contains outliers that cannot, or should not, be removed.

1.2.13 Choosing the Right Normalization Technique

With several normalization techniques available, choosing the most suitable one for a given dataset depends on the characteristics of the data and the requirements of the analysis. Table 1.2.4 summarizes the key considerations.

Table 1.2.4 Comparison of Normalization Techniques

Technique	Best Used When	Sensitivity to Outliers
Min-Max Normalization	Data has no significant outliers and a bounded range (e.g., 0 to 1) is specifically required	High
Z-score Standardization	Data is approximately normally distributed; used widely as a default for machine learning features	Moderate
Decimal Scaling	A quick, simple rescaling of large monetary or count values is needed	Moderate
Robust Scaling	Data contains outliers that should not distort the scaling of the remaining values	Low

SGOU

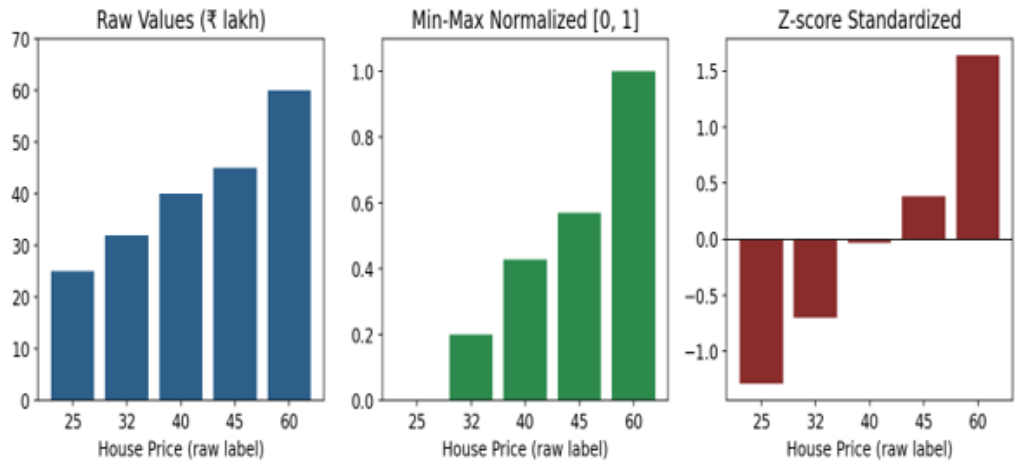


Fig.1.2.3 Raw House Price Data Compared with Min-Max Normalized and Z-score Standardized Versions

As shown in Figure 1.2.3, both min-max normalization and z-score standardization preserve the relative ordering and shape of the original data while placing it on a new, more comparable scale -- min-max compresses every value into the range 0 to 1, while z-score standardization centres the data around 0, with roughly equal spread on either side for symmetric data. The choice between them, and robust scaling where relevant, should always be guided by whether the dataset contains outliers that the analyst wishes to preserve, dampen, or exclude.

1.2.14 Applications of Anomaly Detection and Normalization in Data Analytics

The techniques introduced in this unit are applied throughout the modern data analytics and machine learning pipeline:

- **Fraud Detection:** Banks and payment platforms use anomaly-detection techniques, often extensions of the z-score and IQR methods introduced here, to flag unusual transactions for manual review in real time.
- **Data Cleaning in EDA:** As part of exploratory data analysis, box plots and z-score checks are routinely used to identify and treat data-entry errors before any further analysis is performed.
- **Machine Learning Preprocessing:** Feature scaling using min-max normalization or z-score standardization is a standard preprocessing step before training most machine learning models, particularly those based on distance calculations or gradient-based optimization.
- **Quality Monitoring in Manufacturing:** Sensor readings from production lines are continuously checked for anomalies, allowing early detection of equipment faults before they cause larger failures.

SGOU

- **Healthcare Analytics:** Robust scaling and outlier detection are used when analyzing patient vital signs and laboratory results, where a small number of extreme readings must be handled carefully rather than automatically discarded, since they may indicate a genuine medical emergency.

Having learned how to detect and appropriately handle anomalies and how to normalize data onto comparable scales, the learner is now equipped with a clean, well-prepared dataset -- the essential starting point for the probability distributions introduced in Unit 3, which describe how such prepared data is expected to behave under different real-world processes.

Recap

- An anomaly (or outlier) is an observation that deviates significantly from the rest of the dataset, and may be classified as a point, contextual, or collective anomaly.
- Anomalies can arise from data-entry errors, sensor malfunctions, sampling errors, or genuine rare events, and the appropriate handling strategy depends on the underlying cause.
- The z-score method flags an observation as a potential outlier when its absolute z-score exceeds a threshold, commonly 3, but can be distorted by the very outliers it is trying to detect.
- The interquartile range (IQR) method uses the first and third quartiles to construct lower and upper fences, and is more robust to extreme values than the z-score method.
- Box plots provide a standard visual representation of the IQR method, displaying the median, the interquartile range, and any points beyond the fences as outliers.
- Anomalies may be handled by removal, capping (winsorization), transformation, imputation, or retention for further investigation, depending on their cause and importance.
- Normalization rescales numeric data onto a common scale, which is essential for distance-based and gradient-based machine learning algorithms and for fair comparison of variables measured in different units.
- Min-max normalization rescales data to a fixed range (commonly 0 to 1) but is sensitive to outliers; z-score standardization centres data around a mean of 0; decimal scaling shifts the decimal point; and robust scaling, based on the median and IQR, is the most resistant to outliers.
- The choice of normalization technique should be guided by whether the dataset contains outliers that should be preserved, dampened, or excluded.

SGOU

Objective Type Questions

1. What is an anomaly in a dataset?
2. Which type of anomaly involves a single data point that is far removed from the rest of the dataset?
3. What is a contextual anomaly?
4. Which two statistical values are used to calculate a z-score?
5. What is the commonly used z-score threshold for flagging an outlier?
6. How is the interquartile range (IQR) calculated?
7. What is the formula for the upper fence in the IQR method?
8. Which visualization tool displays the median, interquartile range, and outliers of a dataset?
9. What is winsorization?
10. Why is normalization important before applying many machine learning algorithms?
11. What is the formula for min-max normalization?
12. What is the main limitation of min-max normalization?
13. Which normalization technique centres data around a mean of 0 and a standard deviation of 1?
14. Which two statistical values are used in robust scaling?
15. Which normalization technique is most resistant to the influence of outliers?

SGOU

Answers

1. An observation that deviates significantly from the rest of the dataset
2. Point anomaly
3. A value that is normal in one context but abnormal in another
4. The mean and the standard deviation
5. 3 (an absolute z-score greater than 3)
6. IQR = Q3 − Q1
7. Upper Fence = Q3 + 1.5 × IQR

8. Box plot
9. Capping an extreme value to the nearest acceptable boundary value
10. To bring variables measured on different scales onto a common, comparable scale
11. x' = (x − min) / (max − min)
12. It is highly sensitive to outliers
13. Z-score standardization
14. The median and the interquartile range (IQR)
15. Robust scaling

Assignments

1. A dataset of daily temperatures (in °C) recorded over 9 days is: 28, 29, 27, 30, 29, 28, 31, 30, 4. Compute the mean and standard deviation, calculate the z-score of each observation, and identify any point anomaly using a threshold of $|z| > 2$.
2. For the dataset 12, 15, 14, 18, 20, 21, 22, 25, 60, calculate Q1, Q3, and the IQR, and use the 1.5×IQR rule to determine the lower and upper fences. State which value, if any, is flagged as an outlier.
3. Explain, using an example of your own, the difference between removing an anomaly and winsorizing it. Under what circumstances would you prefer one approach over the other?
4. A feature representing 'annual income (in ₹ lakh)' has values 4, 5, 6, 7, 8, 50. Apply both min-max normalization and robust scaling to this feature, and explain, with reference to your results, why robust scaling is generally preferred when a dataset contains a significant outlier.
5. Describe, in your own words, why data normalization is typically performed only after anomalies have been detected and appropriately handled, rather than before.

SGOU

Suggested Reading

1. Agarwal, B. L. (2013). Basic Statistics. New Age International.
2. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. Morgan Kaufmann.
3. Aggarwal, C. C. (2017). Outlier Analysis (2nd ed.). Springer.

Reference

1. Agarwal, B. L. (2013). Basic Statistics. New Age International.
2. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. Morgan Kaufmann.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

SGOU



Distributions

Learning Outcomes

After completion of this unit, learner will be able to:

- distinguish between a discrete random variable and a continuous random variable
- identify the conditions under which a binomial distribution applies, and calculate binomial probabilities, mean, and variance
- identify the conditions under which a Poisson distribution applies, and calculate Poisson probabilities, mean, and variance
- explain the relationship between the binomial and Poisson distributions
- describe the continuous uniform distribution and calculate probabilities using its probability density function
- describe the normal distribution, use the standard normal distribution to calculate probabilities, and relate the empirical rule to the normal curve
- select an appropriate probability distribution to model a given real-world data analytics scenario

SGOU

Prerequisite

Return to the call-centre manager from earlier: she knows her centre receives about four calls an hour on average, but staffing decisions require more than an average -- she needs to know how likely it is to receive six calls, or eight, or none at all, in any given hour. This unit shows both her and you how to turn a single average figure into a complete, exact set of probabilities for every possible outcome.

To follow her example, you should be comfortable with the mean and standard deviation from Unit 1. You should also know that probabilities are numbers between 0 and 1, and be able to work out a simple counting expression such as 'how many ways can 2 items be chosen from 8'. This unit refreshes that skill in Section 1.3.3, when it is needed.

Keywords

Random Variable, Discrete Distribution, Continuous Distribution, Binomial Distribution, Poisson Distribution, Uniform Distribution, Normal Distribution, Probability Density Function, Standard Normal Distribution.

Discussion

1.3.1 Introduction to Probability Distributions

A probability distribution describes how the probabilities of a random variable are spread across its possible values. The descriptive statistics of Unit 1 described one fixed dataset. A probability distribution is different. It is a mathematical model of how a whole population of possible outcomes is expected to behave. Examples are the number of defective items in a batch, the number of customer calls received in an hour, or the height of an adult chosen at random.

Probability distributions form the theoretical bridge between the descriptive statistics covered in Unit 1 and the inferential statistics that data analysts rely on to make predictions and decisions. Once an analyst identifies the distribution that best describes a real-world process, it becomes possible to calculate the probability of specific outcomes, estimate how unusual a particular observation is, and even simulate new data that behaves similarly to the real process being studied.

This unit introduces four of the most widely used probability distributions in data analytics -- the binomial and Poisson distributions, which describe discrete random variables, and the uniform and normal distributions, which describe continuous random variables.

It is useful to think of each probability distribution as a compact mathematical summary of an entire family of possible datasets that could arise from a particular type of random process. Just as the mean and standard deviation in Unit 1 condensed a specific dataset into two informative numbers, a probability distribution condenses an entire random process into a small number of parameters -- for example, just n and p for the binomial distribution, or just μ and σ for the normal distribution. Once an analyst has identified the right distribution and estimated its parameters from data, many practical questions can be answered without collecting more data. For example, the analyst can estimate the probability of an unusually high number of customer complaints in a week, or the chance that a new product's weight falls outside an acceptable range.

1.3.2 Random Variables: Discrete versus Continuous

A random variable is a variable whose value is determined by the outcome of a random process or experiment. Random variables are classified into two broad types, and identifying which type applies to a given situation is the first step in choosing the correct probability distribution.

SGOU

Discrete Random Variable: A variable that can take only a countable number of distinct values, typically whole numbers arising from a counting process. Examples include the number of defective items in a batch, the number of customer complaints received in a day, or the number of heads obtained in five coin tosses.

Continuous Random Variable: A variable that can take any value within a given range, typically arising from a measurement process. Examples include the exact height of a person, the time taken to complete a task, or the temperature recorded at a weather station.

Table 1.3.1 Discrete versus Continuous Random Variables

Type	Nature of Values	Distributions Covered in this Unit
Discrete	Countable, often whole-number counts	Binomial, Poisson
Continuous	Any value within a range (measured, not counted)	Uniform, Normal

For a discrete random variable, probabilities are described using a probability mass function (PMF), which gives the exact probability of each individual value. For a continuous random variable, probabilities are described using a probability density function (PDF); because a continuous variable can take infinitely many values, the probability of any single exact value is technically zero, and probabilities are instead calculated for a variable falling within a given range or interval.

A helpful way to build intuition for this distinction is to ask whether the variable is the result of counting or measuring. The number of emails received in a day, the number of items sold, or the number of typing errors on a page are all the result of counting, and are therefore discrete. In contrast, the exact time taken to load a webpage, the precise weight of a product, or the temperature at noon are all the result of measuring, and are therefore continuous, even though in practice such measurements are often rounded to a convenient number of decimal places for reporting.

SGOU

1.3.3 Discrete Distribution: The Binomial Distribution

The binomial distribution models the number of successes in a fixed number of independent trials, where each trial has only two possible outcomes (commonly labelled 'success' and 'failure') and the same probability of success on every trial. Such a trial is known as a Bernoulli trial. A random variable X follows a binomial distribution when the following four conditions are satisfied:

- There is a fixed number of trials, n.
- Each trial results in one of exactly two outcomes: success or failure.
- The probability of success, p, remains the same on every trial.

- The trials are independent of one another -- the outcome of one trial does not affect any other.

Under these conditions, the probability of obtaining exactly x successes in n trials is given by the binomial probability mass function:

$$P(X = x) = {}^{n}C_{x} \cdot p^{x} \cdot (1-p)^{n-x}$$

where ${}^{n}C_{x} = n! / [x!(n-x)!]$ is the number of ways of choosing x successes from n trials, p is the probability of success on a single trial, and $(1-p)$ is the probability of failure.

The mean and variance of a binomial distribution are given by simple formulas involving only n and p:

$$\text{Mean } (\mu) = n \cdot p \qquad \text{Variance } (\sigma^2) = n \cdot p \cdot (1-p)$$

Worked Example 1: A quality control inspector examines a random sample of 8 items from a production line known to have a 10% defect rate (p = 0.1). What is the probability that exactly 2 of the 8 items are defective?

$$P(X = 2) = {}^{8}C_{2} \cdot (0.1)^{2} \cdot (0.9)^{6} = 28 \times 0.01 \times 0.531441 \approx 0.1488$$

Mean = 8 × 0.1 = 0.8 defective items Variance = 8 × 0.1 × 0.9 = 0.72

SD ≈ 0.8485

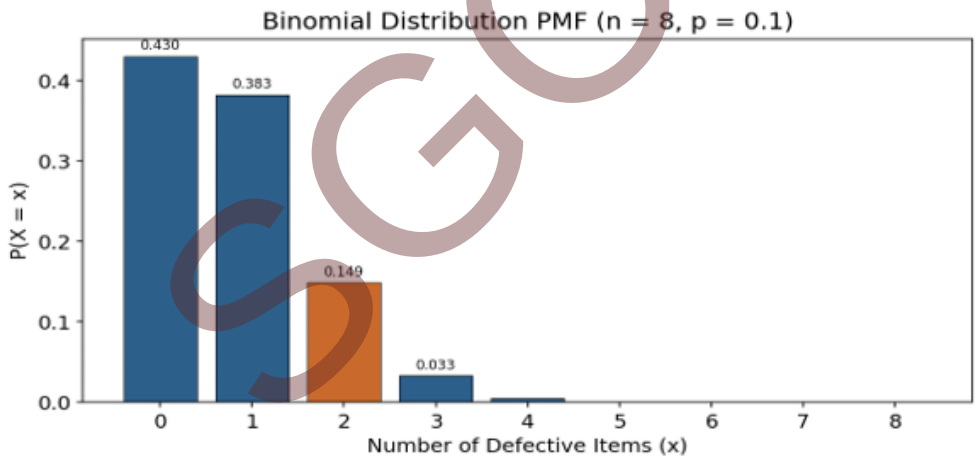


Fig. 1.3.1 Binomial Distribution PMF for n = 8, p = 0.1, with P(X = 2) Highlighted

As Figure 1.3.1 shows, the probability of finding exactly 2 defective items out of 8 is approximately 0.1488, or 14.88%. Note that the distribution is concentrated around small values of x, since with a mean of only 0.8 defective items, finding zero or one defective item is far more likely than finding two or more -- a pattern typical of binomial distributions with a small success probability p.

The binomial distribution is one of the most frequently used distributions in business analytics. It suits situations with a clear 'success or failure' result repeated a fixed number

SGOU

of times. Examples are the number of customers who click an advertisement out of 100 shown, the number of patients who respond to a treatment in a fixed trial group, or the number of correct answers a student gets in a 20-question multiple-choice test.

Worked Example 1 (continued): Using the same inspection scenario, what is the probability that at least one of the 8 items is defective? This is most easily calculated using the complement rule -- rather than adding up the probabilities of 1, 2, 3, ... up to 8 defective items, it is far simpler to subtract the probability of zero defective items from 1.

$$P(X = 0) = {}^{8}C_0 \cdot (0.1)^0 \cdot (0.9)^8 = 1 \times 1 \times 0.4305 \approx 0.4305$$

$$P(X \geq 1) = 1 - P(X = 0) = 1 - 0.4305 = 0.5695 \quad (56.95\%)$$

This tells the inspector that, despite a relatively low 10% individual defect rate, there is still a better-than-even chance (about 57%) that at least one defective item will turn up in any batch of 8 items inspected -- a useful and sometimes counter-intuitive insight that highlights the value of computing exact probabilities rather than relying on intuition alone.

1.3.4 Discrete Distribution: The Poisson Distribution

The Poisson distribution models the number of times an event occurs within a fixed interval of time or space, when such events happen independently of one another and at a constant average rate. Unlike the binomial distribution, the Poisson distribution does not require a fixed number of trials -- instead, it is suited to situations where events can, in principle, occur any number of times within the interval being studied.

A random variable X follows a Poisson distribution when the following conditions are satisfied:

- Events occur independently of one another.
- The average rate at which events occur, denoted λ (lambda), is constant over the interval being studied.
- Two events cannot occur at exactly the same instant.

Under these conditions, the probability of exactly x events occurring in the given interval is given by the Poisson probability mass function:

$$P(X = x) = (e^{-}\lambda \cdot \lambda^{x}) / x!$$

where λ is the average number of events in the interval, and e ≈ 2.71828 is Euler's number. A key property of the Poisson distribution is that its mean and variance are both equal to λ:

$$\text{Mean } (\mu) = \lambda \qquad \text{Variance } (\sigma^2) = \lambda$$

Worked Example 2: A customer support call centre receives an average of 4 calls per hour (λ = 4). What is the probability that exactly 6 calls are received in a given hour?

$$P(X = 6) = (e^{-4} \times 4^6) / 6! = (0.018316 \times 4096) / 720 \approx 0.1042$$

Mean = 4 calls per hour Variance = 4 SD = 2 calls per hour

SGOU

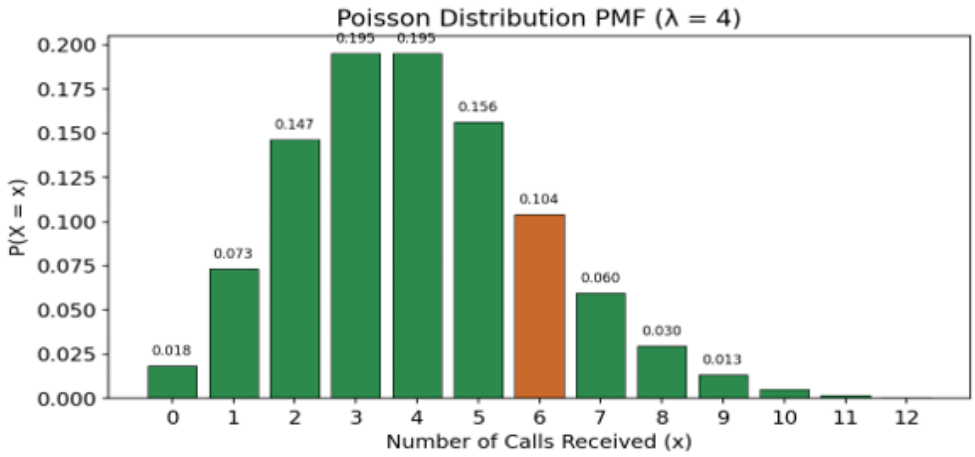


Fig. 1.3.2 Poisson Distribution PMF for λ = 4, with P(X = 6) Highlighted

As shown in Figure 1.3.2, the probability of receiving exactly 6 calls in an hour is approximately 0.1042, or 10.42%, while the most likely outcomes cluster around the mean of 4 calls -- illustrating the characteristic shape of a Poisson distribution, which is right-skewed for small λ but becomes increasingly symmetric and bell-shaped as λ grows larger.

Relationship Between the Binomial and Poisson Distributions

The Poisson distribution can be viewed as a limiting case of the binomial distribution, arising when the number of trials n is very large and the probability of success p is very small, such that their product, $\lambda = n \cdot p$, remains a moderate, fixed value. This makes the Poisson distribution a convenient approximation to the binomial distribution when n is large and p is small. For example, take the number of typing errors on a page of a book. There is a very large number of characters, and each has a tiny chance of being mistyped. A Poisson distribution handles this more easily than the full binomial formula.

In data analytics, the Poisson distribution is widely used to model rare, independent events occurring over time or space -- website server errors per day, machine breakdowns per month, insurance claims per year, or customer arrivals at a service counter -- making it one of the most practically important distributions for operational and reliability analysis.

Worked Example 2 (continued): What is the probability that the call centre receives no calls at all in a given hour?

$$P(X = 0) = (e^{-4} \times 4^0) / 0! = e^{-4} \approx 0.0183 \quad (1.83\%)$$

With an average of 4 calls per hour, there is only about a 1.83% chance of a completely silent hour -- information a call-centre manager could use to justify maintaining at least minimal staffing during every operating hour, since a genuinely quiet hour is a rare occurrence rather than something to be routinely expected.

SGOU

1.3.5 Continuous Distribution: The Uniform Distribution

The continuous uniform distribution describes a random variable that is equally likely to take any value within a fixed interval [a, b], with no value in that interval being more likely than any other. Its probability density function is constant across the interval and zero elsewhere:

$f(x) = 1 / (b - a)$ for $a \leq x \leq b$; $f(x) = 0$ otherwise

The mean and variance of a uniform distribution depend only on the endpoints of the interval:

Mean (μ) = $(a + b) / 2$ Variance (σ^2) = $(b - a)^2 / 12$

Worked Example 3: A city bus is scheduled to arrive at a stop every 20 minutes, and a passenger arrives at the stop at a random time. The passenger's waiting time, X, is uniformly distributed between 0 and 20 minutes (a = 0, b = 20). What is the probability that the passenger waits more than 15 minutes?

Mean = $(0 + 20)/2 = 10$ minutes Variance = $20^2/12 \approx 33.33$ SD ≈ 5.77 minutes

$$P(X > 15) = (20 - 15) / (20 - 0) = 5/20 = 0.25$$

There is a 25% chance that the passenger will have to wait more than 15 minutes for the bus. Notice that, unlike the binomial and Poisson distributions, calculating a uniform probability requires only the length of the relevant interval relative to the total range -- a direct consequence of the flat, constant shape of the uniform density function, shown in the left panel of Figure 1.3.3.

The uniform distribution is commonly used in data analytics to model situations of complete randomness within known bounds, such as random number generation for simulations, modelling rounding errors, or representing a starting assumption of 'no prior information' before more specific data becomes available.

1.3.6 Continuous Distribution: The Normal Distribution

The normal distribution (also called the Gaussian distribution) is the most important continuous probability distribution in statistics and data analytics, owing to how frequently it arises in nature and how central it is to statistical theory. It produces the familiar symmetric, bell-shaped curve first introduced in the discussion of standard deviation in Unit 1. Its probability density function is:

$$f(x) = [\ 1 / (\sigma\sqrt{(2\pi)})\ ] \cdot e^{\wedge}[\ -(x-\mu)^2 / (2\sigma^2)\ ]$$

where μ is the mean and σ is the standard deviation of the distribution. The normal distribution is fully determined by just these two parameters, and it is always symmetric about its mean, with Mean = Median = Mode.

The Standard Normal Distribution: Since the normal distribution's formula depends on μ and σ, calculating probabilities directly for every possible combination of these parameters would be impractical. Instead, any normal distribution can be

SGOU

converted into the standard normal distribution -- a normal distribution with $\mu = 0$ and $\sigma = 1$ -- using the same z-score transformation introduced in Unit 1:

$$z = (x - \mu) / \sigma$$

Once converted to a z-score, probabilities can be read directly from a standard normal (z) table, which lists the cumulative probability, $\Phi(z)$, that a standard normal variable is less than or equal to z. A short extract of such a table is shown in Table 1.3.2.

Table 1.3.2 Selected Cumulative Probabilities, $\Phi(z)$, for the Standard Normal Distribution

z	0.0	0.5	1.0	1.5	2.0	2.5
$\Phi(z)$	0.5000	0.6915	0.8413	0.9332	0.9772	0.9938

Worked Example 4: The scores on a standardized test are normally distributed with a mean of 70 and a standard deviation of 10 ($\mu = 70$, $\sigma = 10$). What is the probability that a randomly selected student scores less than 85?

$$z = (85 - 70) / 10 = 1.5 \qquad P(X < 85) = \Phi(1.5) = 0.9332 \ \ (93.32\%)$$

Worked Example 5: Using the same test, what is the probability that a randomly selected student scores between 60 and 80?

$$z_1 = (60 - 70)/10 = -1 \qquad z_2 = (80 - 70)/10 = 1$$

$$P(60 < X < 80) = \Phi(1) - \Phi(-1) = 0.8413 - 0.1587 = 0.6826 \ \ (68.26\%)$$

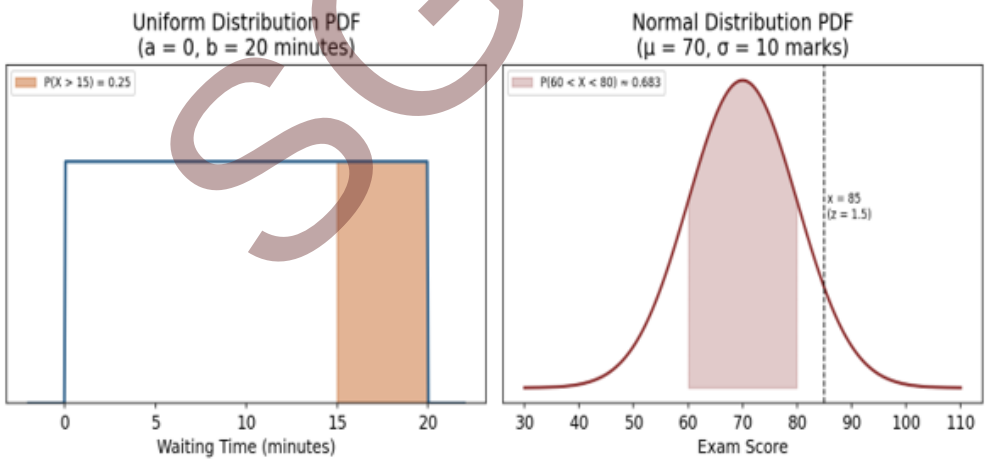


Fig. 1.3.3 The Continuous Uniform Distribution (left) and the Normal Distribution (right), with the Probabilities from Worked Examples 3 to 5 Shaded

The result in Worked Example 5 -- a probability of approximately 68.26% that a score falls within one standard deviation of the mean -- is precisely the empirical rule introduced in Unit 1, now derived formally using the standard normal distribution

SGOU

rather than simply stated as an approximation. This is why the normal distribution is so important in statistics. Many real-world quantities, such as measurement errors, human heights and standardized test scores, closely follow this bell-shaped pattern. In addition, the Central Limit Theorem, a key result in statistics, shows that the average of a large number of independent observations from almost any distribution tends to follow a normal distribution, whatever the shape of the original data.

Worked Example 6: Continuing with the same test ($\mu = 70$, $\sigma = 10$), what proportion of students would be expected to score above 90, a level the institution has set as the cut-off for a merit award?

$z = (90 - 70)/10 = 2.0$ $P(X > 90) = 1 - \Phi(2.0) = 1 - 0.9772 = 0.0228$ (2.28%)

Only about 2.28% of students are expected to score above 90 -- a result the institution can use to estimate, in advance, roughly how many merit awards it should budget for in a class of a given size, without waiting for the actual results to be declared.

1.3.7 Comparing the Four Distributions

Table 1.3.3 summarizes the four probability distributions introduced in this unit, to help the learner quickly identify which distribution is appropriate for a given data analytics scenario.

Table 1.3.3 Summary Comparison of the Binomial, Poisson, Uniform, and Normal Distributions

Distribution	Type	Key Parameter(s)	Mean	Variance
Binomial	Discrete	n (trials), p (success probability)	$n \cdot p$	$n \cdot p \cdot (1-p)$
Poisson	Discrete	λ (average rate of events)	λ	λ
Uniform	Continuous	a, b (interval endpoints)	$(a+b)/2$	$(b-a)^2/12$
Normal	Continuous	μ (mean), σ (standard deviation)	μ	σ^2

A useful practical guideline is to first decide whether the quantity being modelled is discrete (a count) or continuous (a measurement). If discrete, and there is a fixed number of trials with a constant success probability, the binomial distribution applies; if instead events can occur any number of times at a constant average rate, the Poisson distribution applies. If continuous, and every value in a range is equally likely, the uniform distribution applies; if the data instead clusters symmetrically around a central value with progressively fewer observations further away, the normal distribution applies.

SGOU

It is also worth noting how the four distributions relate to the descriptive statistics covered in Unit 1. For these distributions, the mean and variance are not calculated from raw data as they were in Unit 1. They are worked out from the distribution's parameters: n and p for the binomial, λ for the Poisson, a and b for the uniform, and μ and σ for the normal (which are themselves the mean and standard deviation). This is one of the most powerful ideas in this unit. Once an analyst is confident that a distribution fits a process, only a few parameters need to be estimated from data. After that, the mean, the variance and every other probability of interest can be calculated exactly from the distribution's formulas, without observing every possible outcome.

1.3.8 Applications of Probability Distributions in Data Analytics

Probability distributions are not merely abstract mathematical models -- they are applied directly across many areas of data analytics:

- A/B Testing and Conversion Rate Analysis: The binomial distribution is used to model the number of users who convert (make a purchase, click a link) out of a fixed number shown a particular version of a webpage or advertisement, forming the statistical basis of A/B testing.
- Reliability and Operations Analytics: The Poisson distribution is used to model equipment failures, customer arrivals, website traffic spikes, and insurance claims, helping businesses plan staffing levels, inventory, and maintenance schedules.
- Simulation and Random Sampling: The uniform distribution underlies random number generators used in Monte Carlo simulations, random sampling of survey respondents, and randomized controlled experiments.
- Quality Control and Process Monitoring: The normal distribution underlies statistical process control charts, where a manufacturing process is considered to be operating normally as long as measurements stay within a defined number of standard deviations of the target mean.
- Foundations of Statistical Inference: Nearly all hypothesis testing and confidence interval methods used in inferential statistics rely on the normal distribution, either directly or through the Central Limit Theorem, making it the single most important distribution for a data analyst to master.

In practice, an analyst rarely knows in advance which distribution fits a dataset. Part of exploratory data analysis (introduced in Unit 2) is to examine the shape of the data using histograms, summary statistics and, at a more advanced level, formal statistical tests. This helps to judge whether a binomial, Poisson, normal or other distribution fits best. Choosing the right distribution is not merely an academic exercise: a mismatched distributional assumption can lead to badly wrong probability estimates, unreliable forecasts, and misleading business decisions, which is why this unit places equal emphasis on recognizing when each distribution applies and on the mechanics of calculating probabilities from it.

SGOU

Having learned how to model discrete and continuous random variables using appropriate probability distributions, the learner is now well prepared for Unit 4, which formalizes the underlying rules of probability -- including conditional probability and Bayes' theorem -- that govern how these distributions, and the events they describe, combine and interact with one another.

Recap

- A probability distribution describes how the probabilities of a random variable are spread across its possible values, providing a mathematical model of a random process.
- A discrete random variable takes countable values and is described by a probability mass function (PMF); a continuous random variable takes any value within a range and is described by a probability density function (PDF).
- The binomial distribution models the number of successes in a fixed number of independent trials, each with the same probability of success p; its mean is $n \cdot p$ and its variance is $n \cdot p \cdot (1-p)$.
- The Poisson distribution models the number of independent events occurring at a constant average rate λ within a fixed interval; its mean and variance are both equal to λ, and it can approximate the binomial distribution when n is large and p is small.
- The continuous uniform distribution assigns equal likelihood to every value within a fixed interval [a, b]; its probabilities correspond simply to the proportion of the interval covered.
- The normal distribution is a symmetric, bell-shaped continuous distribution defined by its mean μ and standard deviation σ, and can be converted to the standard normal distribution using the z-score transformation to look up probabilities in a z-table.
- The empirical rule (68%-95%-99.7%) introduced in Unit 1 is a direct consequence of the mathematical properties of the normal distribution.
- Choosing the correct distribution for a given scenario begins with identifying whether the underlying variable is discrete or continuous, and then matching the specific pattern of the data to the binomial, Poisson, uniform, or normal distribution as appropriate

SGOU

Objective Type Questions

1. What is a random variable?
2. What is the key difference between a discrete random variable and a continuous random variable?
3. Which function describes the probabilities of a discrete random variable?
4. What are the four conditions required for a binomial distribution to apply?
5. What is the formula for the mean of a binomial distribution?
6. What is the formula for the variance of a Poisson distribution?
7. Under what condition can the Poisson distribution be used to approximate the binomial distribution?
8. What is the probability density function of a continuous uniform distribution on the interval [a, b]?
9. What is the mean of a uniform distribution on the interval [a, b]?
10. What are the two parameters that fully define a normal distribution?
11. What is the mean and standard deviation of the standard normal distribution?
12. What transformation converts a normal random variable into a standard normal random variable?
13. According to the standard normal table, what is $\Phi(1.0)$?
14. Which theorem states that the average of a large number of independent observations tends to follow a normal distribution?
15. Which distribution would be most appropriate for modelling the number of website server errors occurring per day?

SGOU

Answers

1. A variable whose value is determined by the outcome of a random process or experiment
2. A discrete random variable takes countable values; a continuous random variable takes any value within a range
3. The probability mass function (PMF)

4. A fixed number of trials, two possible outcomes per trial, a constant probability of success, and independent trials
5. Mean = $n \times p$
6. Variance = λ
7. When the number of trials n is large and the probability of success p is small
8. $f(x) = 1 / (b - a)$ for $a \leq x \leq b$
9. Mean = $(a + b) / 2$
10. The mean (μ) and the standard deviation (σ)
11. Mean = 0, Standard Deviation = 1
12. The z-score transformation, $z = (x - \mu) / \sigma$
13. 0.8413
14. The Central Limit Theorem
15. The Poisson distribution

Assignments

1. A fair coin is tossed 6 times. Using the binomial distribution, calculate the probability of obtaining exactly 4 heads, and also calculate the mean and variance of the number of heads obtained.
2. A small bakery sells, on average, 5 units of a specialty cake per day ($\lambda = 5$). Using the Poisson distribution, calculate the probability that exactly 3 units are sold on a given day, and the probability that no units at all are sold that day.
3. The time (in minutes) a customer spends waiting in a queue is uniformly distributed between 2 and 12 minutes. Calculate the mean waiting time, the standard deviation, and the probability that a customer waits less than 5 minutes.
4. The weights of packets produced by a machine are normally distributed with a mean of 500 grams and a standard deviation of 15 grams. Calculate the probability that a randomly selected packet weighs more than 520 grams, and the probability that it weighs between 485 and 515 grams.
5. Explain, in your own words, why the binomial distribution would not be an appropriate model for the number of earthquakes occurring in a region over a year, and suggest which distribution introduced in this unit would be more suitable, with reasoning.

SGOU

Suggested Reading

1. Agarwal, B. L. (2013). Basic Statistics. New Age International.
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.
3. Spiegel, M. R., & Stephens, L. J. (2017). Schaum's Outline of Statistics. McGraw-Hill Education.

Reference

1. Agarwal, B. L. (2013). Basic Statistics. New Age International.
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

SGOU



Concept of Probability

Learning Outcomes

After completion of this unit, learner will be able to:

- define basic probability terminology, including experiment, sample space, event, and outcome
- distinguish between mutually exclusive, independent, and exhaustive events
- describe the classical, empirical, and subjective approaches to assigning probability
- apply the addition rule and the complement rule to calculate the probability of combined events
- apply the multiplication rule to calculate the probability of independent and dependent events occurring together
- calculate and interpret conditional probability
- apply Bayes' theorem to update the probability of an event in light of new evidence

SGOU

Prerequisite

Return to the doctor from earlier, holding a patient's positive test result and pausing before she speaks. She knows the test correctly identifies 95% of true cases, yet she also knows how rare the disease is in the wider population -- and something about combining those two facts changes the picture more than it first appears to. This unit exists to work out exactly how much, and why.

To follow along, you do not need anything beyond what you have already been using throughout this block: comfort with fractions, decimals and percentages, and an everyday sense of when two groups of things overlap and when they do not -- for instance, picturing two overlapping circles of customers who like different products. Along the way, you will also be formalizing probability ideas you have already been using informally since Unit 1, such as reading a z-score or calculating a binomial probability, without perhaps stopping to ask what probability really means or how its rules work.

Keywords

Experiment, Sample Space, Event, Mutually Exclusive Events, Independent Events, Exhaustive Events, Classical Probability, Empirical Probability, Addition Rule, Complement Rule, Multiplication Rule, Conditional Probability, Bayes' Theorem.

Discussion

1.4.1 Introduction to Probability

Probability is the branch of mathematics that quantifies uncertainty. It assigns a numerical value, between 0 and 1 (or equivalently, between 0% and 100%), to how likely a particular outcome or event is to occur. A probability of 0 means an event is impossible, a probability of 1 means an event is certain, and values in between reflect varying degrees of likelihood.

Probability provides the theoretical foundation for every distribution introduced in Unit 3 -- the binomial, Poisson, uniform, and normal distributions are all, at their core, precise rules for assigning probabilities to the possible values of a random variable. This unit steps back from specific distributions to formalize the more general rules of probability that apply to any random event, whether or not it follows one of those particular named distributions.

In data analytics, probability underlies nearly every technique used to handle uncertainty and make decisions from data -- from simple questions such as 'what is the chance that a customer will churn this month?' to sophisticated machine learning classifiers, such as the Naive Bayes algorithm, which is built directly on Bayes' theorem, introduced later in this unit.

It is worth pausing to appreciate why probability, rather than certainty, is the natural language of data analytics. Real-world processes -- customer behaviour, equipment failures, disease spread, financial markets -- are influenced by so many small, unobserved factors that predicting their outcomes with absolute certainty is essentially impossible. Probability offers a disciplined, mathematically consistent way to express and reason about this uncertainty, allowing an analyst to make the most informed decision possible given the available information, rather than either pretending to have certainty that does not exist or refusing to make any decision at all in the face of incomplete information.

1.4.2 Basic Terminology

Before probability rules can be introduced, a small set of foundational terms must be clearly defined.

Experiment : A process or action that produces a well-defined outcome, such as rolling a die, tossing a coin, or observing whether a customer completes a purchase.

Sample Space (S) : The set of all possible outcomes of an experiment. For example, when rolling a standard six-sided die, the sample space is S = {1, 2, 3, 4, 5, 6}.

SGOU

Outcome: A single possible result of an experiment, such as rolling a 4.

Event: A subset of the sample space -- that is, one or more outcomes grouped together based on some condition of interest. For example, the event 'rolling an even number' corresponds to the subset A = {2, 4, 6} of the sample space above.

Events can also be related to one another in specific ways that affect how their probabilities are combined:

- **Mutually Exclusive Events:** Two events that cannot occur at the same time -- that is, they share no common outcomes. For example, when rolling a die, 'rolling a 2' and 'rolling a 5' are mutually exclusive, since a single roll cannot produce both outcomes at once.
- **Independent Events:** Two events where the occurrence of one does not affect the probability of the other occurring. For example, the outcome of one coin toss does not influence the outcome of a second coin toss.
- **Exhaustive Events:** A set of events that together cover the entire sample space, so that at least one of them must occur. For example, 'rolling an even number' and 'rolling an odd number' are exhaustive for a single die roll, since every possible outcome falls into one of these two events.
- **Complementary Event:** The complement of an event A, denoted A' (or Ac), consists of all outcomes in the sample space that are not in A. For a die roll, if A = 'rolling an even number' = {2, 4, 6}, then A' = 'rolling an odd number' = {1, 3, 5}.

Illustrative Example : Consider drawing a single card from a standard deck of 52 playing cards. The sample space S consists of all 52 cards. Let event A = 'drawing a heart' (13 outcomes) and event B = 'drawing a face card, i.e., a Jack, Queen, or King' (12 outcomes). Events A and B are not mutually exclusive, since 3 cards -- the Jack, Queen, and King of hearts -- belong to both events. Event A and its complement A' ('not drawing a heart', 39 outcomes) are, by definition, always mutually exclusive and always exhaustive, since together they account for every card in the deck.

1.4.3 Approaches to Assigning Probability

There are three broad approaches used to assign a numerical probability to an event, depending on the nature of the situation and the information available.

Classical (Theoretical) Probability

Classical probability applies when all outcomes in the sample space are equally likely. It is calculated as the ratio of the number of favourable outcomes to the total number of possible outcomes:

P(A) = (Number of outcomes favourable to A) / (Total number of outcomes in S)

For example, the classical probability of rolling an even number on a fair six-sided die is P(A) = 3/6 = 0.5, since 3 of the 6 equally likely outcomes (2, 4, and 6) are favourable to the event.

SGOU

Empirical (Relative Frequency) Probability

Empirical probability is based on observed data from repeated trials or historical records, rather than a theoretical assumption of equally likely outcomes. It is calculated as:

P(A) = (Number of times A occurred) / (Total number of trials observed)

Worked Example 1: A factory inspected 500 items produced last month and found that 20 of them were defective. The empirical probability that a randomly selected item from this production run is defective is:

P(Defective) = 20 / 500 = 0.04 (4%)

Empirical probability is especially important in data analytics, since most real-world probabilities used in business -- such as a customer's likelihood of churning, or a transaction's likelihood of being fraudulent -- are estimated from historical data using exactly this relative-frequency approach, rather than from a theoretical assumption of equally likely outcomes.

Subjective Probability

Subjective probability reflects an individual's personal judgement or degree of belief about the likelihood of an event, based on experience, intuition, or incomplete information, rather than on a formal calculation. For example, a product manager might estimate that there is a 70% chance that a new feature will be well received by users, based on market experience rather than on any specific dataset. While less rigorous than the classical or empirical approaches, subjective probability is often used in business forecasting and risk assessment when historical data is limited or unavailable.

1.4.4 The Addition Rule and the Complement Rule

The addition rule is used to calculate the probability that at least one of two events occurs -- that is, the probability of their union. For any two events A and B, the general addition rule is:

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$

The term $P(A \cap B)$ -- the probability that both A and B occur -- is subtracted because, when simply adding P(A) and P(B), the outcomes common to both events would otherwise be counted twice.

Worked Example 2: A survey of 100 customers found that 40% like Product A, 35% like Product B, and 15% like both products. What is the probability that a randomly selected customer likes Product A or Product B (or both)?

$P(A \cup B) = P(A) + P(B) - P(A \cap B) = 0.40 + 0.35 - 0.15 = 0.60$

SGOU

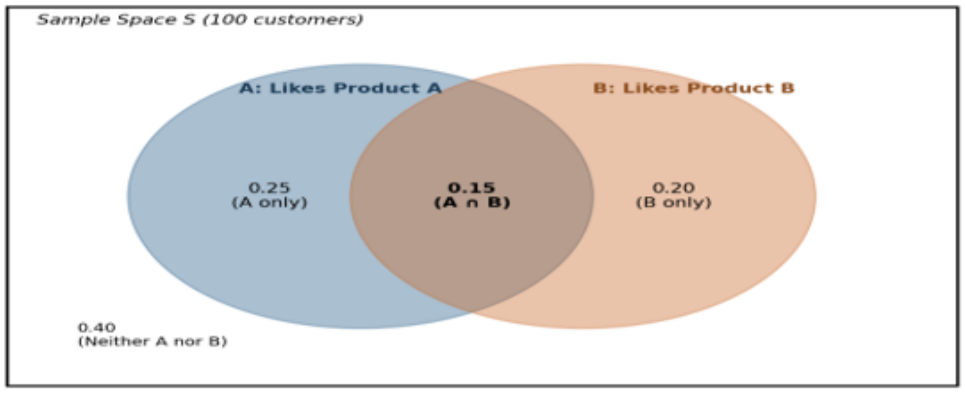


Fig.1.4.1 Venn Diagram Illustrating the Addition Rule for Customers Liking Product A and/or Product B

As shown in Figure 1.4.1, the overlapping region of the two circles represents the 15% of customers who like both products, and is counted only once in the final union probability of 0.60, rather than being added twice.

When two events are mutually exclusive, they share no common outcomes, so $P(A \cap B) = 0$, and the addition rule simplifies to:

$P(A \cup B) = P(A) + P(B)$ [Mutually Exclusive Events Only]

Closely related to the addition rule is the complement rule, which follows from the fact that an event A and its complement A' together account for the entire sample space, and are always mutually exclusive of one another:

$P(A') = 1 - P(A)$

Worked Example 3: Using Worked Example 2, what is the probability that a randomly selected customer likes neither Product A nor Product B?

$P(\text{Neither}) = 1 - P(A \cup B) = 1 - 0.60 = 0.40$ (40%)

The complement rule is often the simplest way to calculate the probability of 'at least one' event occurring, since it is frequently far easier to calculate the probability of the opposite outcome (none of the events occurring) and subtract it from 1, as will be seen again in Worked Example 4 in the next section.

Worked Example (Mutually Exclusive Events): A single card is drawn from a standard deck of 52 cards. What is the probability that the card is either a King or a Queen?

Since a single card cannot simultaneously be both a King and a Queen, these two events are mutually exclusive, and the simplified addition rule applies directly:

$P(\text{King} \cup \text{Queen}) = P(\text{King}) + P(\text{Queen}) = 4/52 + 4/52 = 8/52 \approx 0.1538$ (15.38%)

SGOU

1.4.5 The Multiplication Rule and Independent Events

The multiplication rule is used to calculate the probability that two events both occur -- that is, the probability of their intersection. The general multiplication rule, which applies to any two events, is:

$$P(A \cap B) = P(A) \times P(B \mid A)$$

where P(B | A), read as 'the probability of B given A', is the conditional probability of B occurring given that A has already occurred (conditional probability is formally introduced in Section 1.4.6). When two events are independent -- meaning the occurrence of one does not affect the probability of the other -- this simplifies, since P(B | A) = P(B), to the special case:

$$P(A \cap B) = P(A) \times P(B) \quad \text{[Independent Events Only]}$$

Worked Example 4: Two machines on a production line operate independently of one another. The probability that Machine 1 fails on a given day is 0.02, and the probability that Machine 2 fails on the same day is 0.03. What is the probability that both machines fail on the same day?

$$P(\text{Both Fail}) = P(\text{M1 Fails}) \times P(\text{M2 Fails}) = 0.02 \times 0.03 = 0.0006 \quad (0.06\%)$$

Since the machines operate independently, the probability of both failing simultaneously is very small -- a useful insight for planning redundancy, since relying on two independent machines dramatically reduces the chance of a complete production stoppage compared to relying on either machine alone.

Worked Example 5 (Dependent Events): Two cards are drawn, one after another, without replacement, from a standard deck of 52 playing cards (which contains 4 aces). What is the probability that both cards drawn are aces?

Since the first card is not replaced before the second card is drawn, the two draws are not independent -- the probability of drawing an ace on the second draw depends on the outcome of the first draw.

$$P(\text{1st card is an Ace}) = 4/52 \qquad P(\text{2nd card is an Ace} \mid \text{1st was an Ace}) = 3/51$$

$$P(\text{Both Aces}) = (4/52) \times (3/51) = 12/2652 \approx 0.0045 \quad (0.45\%)$$

This example illustrates why it is essential to check whether two events are independent before applying the simplified multiplication rule: applying $P(A) \times P(B) = (4/52) \times (4/52) \approx 0.0059$ here, ignoring the effect of the first draw on the second, would have given a noticeably different and incorrect answer.

The distinction between independent and dependent events has direct practical consequences in data analytics. Many statistical and machine learning methods explicitly assume that observations are independent of one another -- for example, that one customer's purchasing behaviour does not directly influence another unrelated customer's behaviour. When this assumption is violated, as it often is with time-series data (where today's sales are related to yesterday's sales) or network data (where connected individuals influence one another), standard formulas such as the simplified

SGOU

multiplication rule can no longer be applied directly, and more specialized techniques that explicitly account for the dependence between observations must be used instead.

1.4.6 Conditional Probability

Conditional probability measures the probability of an event occurring given that another event is already known to have occurred. It is formally defined as:

$P(A \mid B) = P(A \cap B) / P(B)$ (for $P(B) > 0$)

Conditional probability is one of the most important ideas in data analytics, since it allows an analyst to update the probability of an event of interest once relevant additional information becomes available -- for example, updating the probability that a customer will churn, given that they have already reduced their usage of a service.

Worked Example :6 Using the customer survey data from Worked Example 2 ($P(A) = 0.40$, $P(B) = 0.35$, $P(A \cap B) = 0.15$), what is the probability that a randomly selected customer likes Product A, given that they are already known to like Product B?

$P(A \mid B) = P(A \cap B) / P(B) = 0.15 / 0.35 \approx 0.4286$ (42.86%)

Notice that $P(A \mid B) \approx 42.86\%$ is somewhat higher than the overall (unconditional) probability $P(A) = 40\%$. This suggests that liking Product B is associated with a slightly higher chance of also liking Product A -- in other words, the two events are not perfectly independent. This can be checked formally: since $P(A) \times P(B) = 0.40 \times 0.35 = 0.14$, which is not equal to the actual value of $P(A \cap B) = 0.15$, events A and B are not independent, and the small difference indicates a mild positive association between liking the two products. In practice, this kind of insight might prompt a marketing analyst to consider cross-promoting Product A to customers who already show interest in Product B.

More generally, conditional probability is the mathematical mechanism behind the idea of 'updating beliefs' as new information arrives -- a theme that recurs throughout data analytics. A recommendation engine, for instance, continuously recalculates the conditional probability that a customer will like a given product, given the growing history of items that customer has already viewed or purchased; an intrusion-detection system recalculates the conditional probability that network activity is malicious, given the specific pattern of requests observed so far. In every case, the unconditional probability calculated in isolation is only a starting point, and the conditional probability, calculated in light of relevant evidence, is what ultimately drives the analytical decision.

1.4.7 Bayes' Theorem

Bayes' theorem provides a formal rule for reversing the direction of a conditional probability -- that is, for calculating $P(A \mid B)$ when only $P(B \mid A)$ and the individual probabilities of A and B are known. It is one of the most powerful and widely applied results in probability theory, and is stated as:

$P(A \mid B) = [\, P(B \mid A) \times P(A) \,] / P(B)$

SGOU

In many practical applications, P(B) is not directly known and must first be calculated using the law of total probability, by summing over all the ways B could have occurred:

$$P(B) = P(B \mid A) \times P(A) + P(B \mid A') \times P(A')$$

Worked Example : 7 (Medical Diagnosis): A disease affects 1% of a population (P(D) = 0.01). A diagnostic test correctly identifies the disease in 95% of people who actually have it (P(Positive | D) = 0.95, the test's sensitivity), but also incorrectly gives a positive result in 5% of people who do not have the disease (P(Positive | No D) = 0.05, the false positive rate). If a randomly selected person tests positive, what is the probability that they actually have the disease?

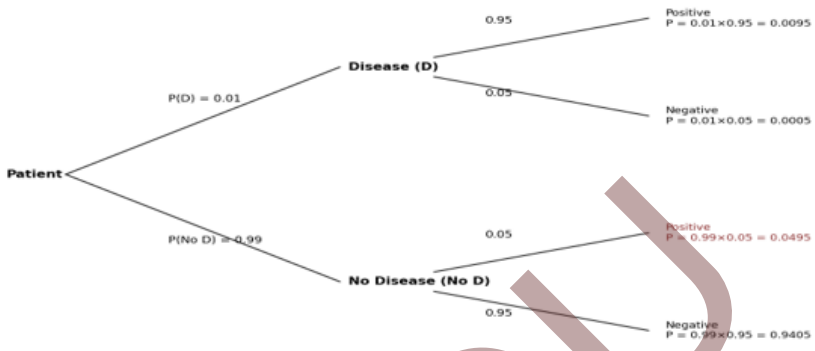


Fig. 1.4.2 Probability Tree Diagram for the Medical Diagnosis Example

Using the law of total probability, based on the branch probabilities shown in Figure 1.4.2:

$$P(\text{Positive}) = (0.95 \times 0.01) + (0.05 \times 0.99) = 0.0095 + 0.0495 = 0.059$$

Applying Bayes' theorem:

$$P(D \mid \text{Positive}) = (0.95 \times 0.01) / 0.059 = 0.0095 / 0.059 \approx 0.161 \quad (16.1\%)$$

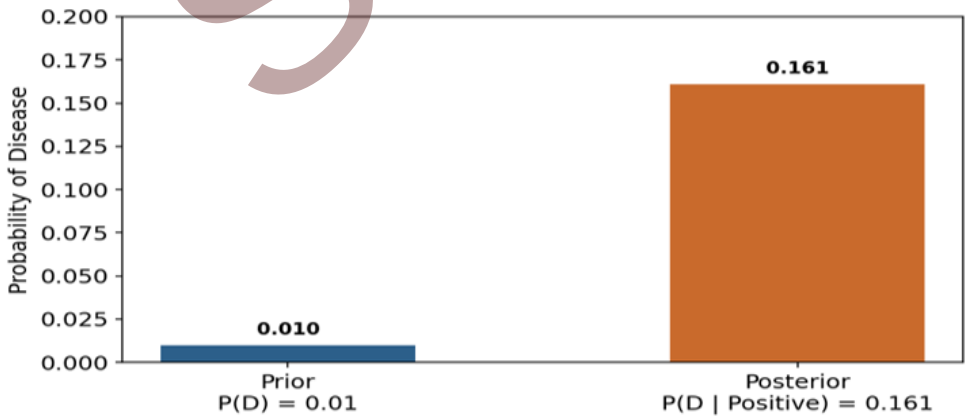


Fig. 1.4.3 Bayesian Updating -- the Prior Probability P(D) Compared with the Posterior Probability P(D | Positive)

SGOU

This result is famously counter-intuitive: even though the test is 95% accurate in both directions, a positive result still only corresponds to a 16.1% chance of actually having the disease. This happens because the disease is rare (only 1% of the population has it), so the much larger group of healthy people -- even at a small 5% false-positive rate -- generates far more false positives in absolute terms than the smaller group of genuinely sick people generates true positives. This is precisely why doctors typically order a second, confirmatory test before concluding that a patient has a rare disease based on a single positive result, and it is a critical lesson for any data analyst interpreting the results of an imperfect classification system.

This same pattern arises constantly in data analytics whenever a classifier or detection system is applied to a rare category -- for example, a fraud-detection model applied to a payment stream in which only a tiny fraction of transactions are actually fraudulent, or a rare-disease screening test applied to an entire population. In every such case, the base rate (prior probability) of the rare category strongly affects how positive flags should be read. An analyst who ignores the base rate and looks only at the accuracy of the test or model risks drawing seriously misleading conclusions. This error is informally known as the 'base rate fallacy'.

Worked Example 8 (Spam Filtering): An email spam filter is being designed. It is known that 30% of all incoming emails are spam (P(Spam) = 0.30). The word 'free' appears in 60% of spam emails (P('free' | Spam) = 0.60) but in only 5% of legitimate emails (P('free' | Not Spam) = 0.05). If an incoming email contains the word 'free', what is the probability that it is spam?

P('free') = (0.60 × 0.30) + (0.05 × 0.70) = 0.18 + 0.035 = 0.215

P(Spam | 'free') = (0.60 × 0.30) / 0.215 = 0.18 / 0.215 ≈ 0.837 (83.7%)

Unlike the rare-disease example, here the base rate of spam (30%) is high enough, and the word 'free' is distinctive enough, that observing it in an email substantially raises the probability of spam, from a prior of 30% to a posterior of about 83.7%. This calculation -- updating the probability of a category based on the presence of specific words -- is precisely the mechanism underlying the Naive Bayes classifier, one of the most widely used and computationally efficient machine learning algorithms for text classification and spam filtering.

1.4.8 Applications of Probability in Data Analytics

The concepts introduced in this unit form the theoretical backbone of many widely used data analytics and machine learning techniques:

- **Naive Bayes Classifiers:** As shown in Worked Example 8, Bayes' theorem is applied directly, and repeatedly, to classify emails, documents, and customer reviews based on the words or features they contain.
- **Medical and Diagnostic Screening:** Bayes' theorem is used to correctly interpret the results of diagnostic tests, accounting for both the accuracy of the test and the underlying prevalence (base rate) of the condition being tested for, exactly as shown in Worked Example 7.

SGOU

- **Fraud Detection:** Conditional probability is used to update the likelihood that a transaction is fraudulent, based on evidence such as an unusual transaction location, amount, or time, building directly on the anomaly-detection ideas introduced in Unit 2.

- **Recommendation Systems:** The addition and multiplication rules underlie the calculation of the probability that a customer will be interested in a product, based on their past purchases and the purchasing patterns of similar customers.

- **A/B Testing and Experimentation:** Understanding independent and mutually exclusive events is essential when designing controlled experiments, to ensure that the probability calculations used to judge whether one version of a product performs better than another are valid.

Across all of these applications, the same handful of ideas introduced in this unit reappear repeatedly: identifying the relevant sample space and events, deciding whether events are independent or dependent, and, whenever new evidence becomes available, using conditional probability and Bayes' theorem to update what is believed about an uncertain outcome. Mastery of these foundational rules is therefore not simply an academic requirement but a practical toolkit that a data analyst will draw upon, explicitly or implicitly, in almost every subsequent stage of this course.

Having completed this unit, the learner has now covered the full mathematical foundation of Block I -- from summarizing data with descriptive statistics, through preparing it by handling anomalies and normalizing scales, to modelling it with probability distributions, and finally to reasoning formally about uncertainty using the rules of probability. These foundations will be applied and built upon extensively as the course progresses into more advanced data analytics and visualization techniques in the subsequent blocks.

Recap

- Probability quantifies uncertainty on a scale from 0 (impossible) to 1 (certain), and is built on the foundational concepts of experiment, sample space, event, and outcome.

- Events may be mutually exclusive (cannot occur together), independent (the occurrence of one does not affect the other), or exhaustive (together cover the entire sample space); every event also has a complement.

- Probability can be assigned using the classical approach (equally likely outcomes), the empirical approach (observed relative frequency), or the subjective approach (personal judgement).

- The addition rule, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$, calculates the probability that at least one of two events occurs; it simplifies to $P(A) + P(B)$ for mutually exclusive events.

SGOU

- The complement rule, $P(A') = 1 - P(A)$, is often the simplest way to calculate the probability of at least one event occurring among several.
- The multiplication rule, $P(A \cap B) = P(A) \times P(B \mid A)$, calculates the probability that two events both occur; it simplifies to $P(A) \times P(B)$ only when the events are independent.
- Conditional probability, $P(A \mid B) = P(A \cap B) / P(B)$, measures how the probability of an event changes once another related event is known to have occurred.
- Bayes' theorem, $P(A \mid B) = [P(B \mid A) \times P(A)] / P(B)$, allows a conditional probability to be reversed, and is the mathematical foundation of Naive Bayes classifiers, diagnostic test interpretation, and spam filtering.
- A low base rate (prior probability) can make even a highly accurate test produce a surprisingly low posterior probability of a positive result being a true positive -- a critical lesson for interpreting any imperfect classification system.

Objective Type Questions

1. What is the sample space of an experiment?
2. What is an event in probability?
3. What are mutually exclusive events?
4. What are independent events?
5. Which approach to probability is based on the ratio of favourable outcomes to total outcomes, assuming all outcomes are equally likely?
6. Which approach to probability is based on observed relative frequency from historical data?
7. What is the general addition rule for two events A and B?
8. What is the complement rule?
9. What is the formula for the probability of two independent events both occurring?
10. What is the formula for conditional probability, $P(A \mid B)$?
11. What does Bayes' theorem allow an analyst to calculate?
12. In the medical diagnosis worked example, what term describes the probability $P(\text{Positive} \mid \text{Disease})$?

SGOU

13. Why was the posterior probability of disease, given a positive test, much lower than the test's stated accuracy?

14. Which machine learning algorithm is directly based on Bayes' theorem?

15. What is the law of total probability used for in Bayes' theorem calculations?

Answers

1. The set of all possible outcomes of an experiment
2. A subset of the sample space representing one or more outcomes of interest
3. Events that cannot occur at the same time (share no common outcomes)
4. Events where the occurrence of one does not affect the probability of the other
5. The classical (theoretical) approach
6. The empirical (relative frequency) approach
7. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
8. $P(A') = 1 - P(A)$
9. $P(A \cap B) = P(A) \times P(B)$
10. $P(A \mid B) = P(A \cap B) / P(B)$
11. The probability of an event given new evidence, by reversing a known conditional probability
12. Sensitivity (the true positive rate)
13. Because the disease has a low base rate (prior probability), so false positives from the much larger healthy population outnumber true positives
14. The Naive Bayes classifier
15. To calculate the overall probability of the evidence, P(B), by summing over all the ways it could occur

SGOU

Assignments

1. A card is drawn at random from a standard deck of 52 playing cards. Define the sample space, and calculate the classical probability of drawing a face card (Jack, Queen, or King).
2. In a survey of 200 employees, 90 reported using public transport, 70 reported using a personal vehicle, and 30 reported using both. Calculate the probability that a randomly selected employee uses public transport or a personal vehicle (or both), and the probability that they use neither.
3. A bag contains 5 red balls and 7 blue balls. Two balls are drawn one after another without replacement. Calculate the probability that both balls drawn are red, showing clearly why the multiplication rule for dependent events must be used.
4. A company finds that 8% of its customers are 'high value' customers. A predictive model flags 80% of actual high-value customers correctly (sensitivity), but also incorrectly flags 10% of non-high-value customers as high-value. Using Bayes' theorem, calculate the probability that a customer flagged by the model is actually a high-value customer, and comment on whether the model's flag alone should be considered reliable.
5. Explain, using an example of your own (different from those used in this unit), the difference between independent events and mutually exclusive events, and why two events that are mutually exclusive (and have non-zero probabilities) can never also be independent.

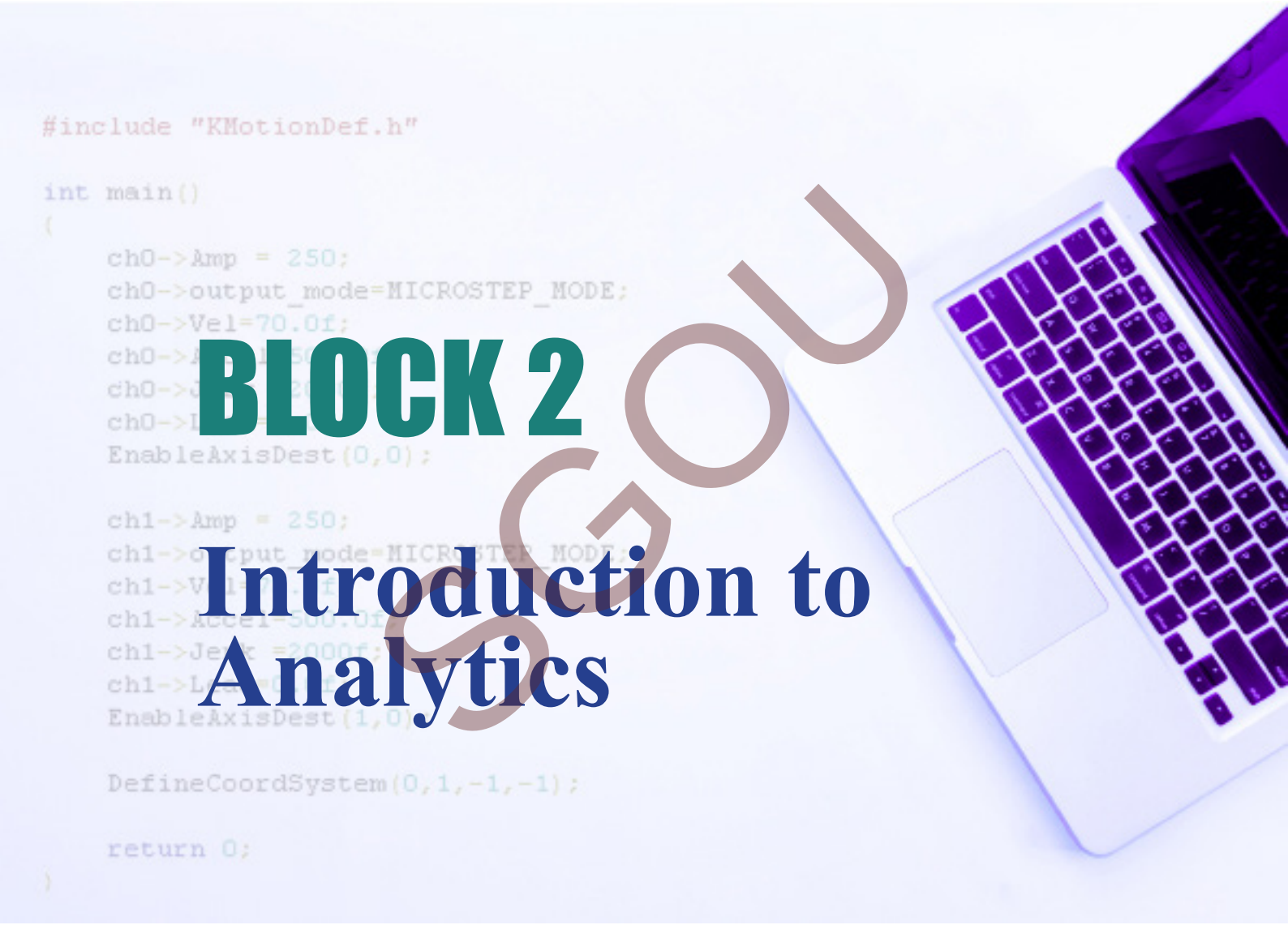
Suggested Reading

1. Agarwal, B. L. (2013). Basic Statistics. New Age International.
2. Ross, S. M. (2014). A First Course in Probability (9th ed.). Pearson.
3. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.

Reference

1. Agarwal, B. L. (2013). Basic Statistics. New Age International.
2. Ross, S. M. (2014). A First Course in Probability (9th ed.). Pearson.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

SGOU



BLOCK 2

Introduction to Analytics

SGOU



Big Data Technologies and Data Warehousing

Learning Outcomes

After completion of this unit, learner will be able to:

- define big data and describe its characteristics
- identify the 5 Vs of big data and the different types of data
- describe the basic components of the Hadoop ecosystem
- explain the functions of HDFS and the MapReduce programming model
- identify the major components of a data warehouse and explain the ETL process
- differentiate between OLTP and OLAP systems, and between star and snowflake schemas

SGOU

Prerequisite

Have you ever wondered how online shopping websites manage millions of orders, how banks monitor thousands of transactions every second, or how video-sharing platforms store and process countless photos and videos uploaded by users worldwide? Every online purchase, social media post, digital payment, and GPS location adds to the growing amount of data generated every day. Managing such enormous amounts of information is a major challenge in today's digital world. How can this data be stored safely, processed efficiently, and converted into useful information?

To explore these questions, it is helpful to begin with what you already know about data and databases. You may have used spreadsheets to organize information, searched for records in a database, or observed how businesses use reports to understand their performance. These familiar experiences provide a useful starting point for understanding the challenges of handling large and varied datasets. In this unit, you will explore the characteristics of big data, discover how Apache Hadoop distributes data storage and processing across multiple computers, and understand how data warehouses organize information for business reporting and decision-making. You will also learn how technologies such as HDFS, MapReduce, and the ETL process support

large-scale data management and analysis. No previous hands-on experience with big data technologies is required. Bring your curiosity and readiness to explore, and get ready to discover how the data-driven world works behind the scenes.

Keywords

Big Data, Structured Data, Unstructured Data, Hadoop, HDFS, MapReduce, YARN, Hive, Pig, HBase, Data Warehouse, OLTP, OLAP, Star Schema, Snowflake Schema, ETL.

Discussion

2.1.1 Introduction to Big Data

Every time a person swipes a debit card, posts a photo, streams a video, or drives a car fitted with GPS tracking, a small piece of data is generated. Individually, each of these events is trivial. Collectively, across billions of people and devices acting every second of every day, they add up to an almost unimaginable volume of information and it is this scale, arriving from this many directions, at this kind of speed, that the term big data is used to describe.

Big data refers to datasets that are so large, so fast-moving, or so varied in structure that they cannot be efficiently captured, stored, managed, or analyzed using traditional data-processing software and a single, standard computer. A traditional spreadsheet application, for example, may struggle to even open a file containing several hundred million rows, let alone analyze it -- yet datasets of this size, and far larger, are now routine in fields such as e-commerce, telecommunications, healthcare, and social media.

It is important to note that 'big data' is not simply 'a lot of data'. It refers to a qualitatively different challenge that requires fundamentally different tools and approaches -- distributing both the storage and the processing of data across many computers working together, rather than relying on a single, more powerful machine. The remainder of this unit introduces exactly the technologies, led by the Hadoop ecosystem, that were built to meet this challenge, before turning to data warehousing, a complementary and longer-established approach to organizing data specifically for business reporting and analysis.

2.1.2 Characteristics of Big Data: The 5 Vs

Big data is most commonly described in terms of five defining characteristics, widely known as the 5 Vs. This framework, which grew out of an original 'three Vs' definition proposed by industry analyst Doug Laney in 2001, helps clarify precisely what distinguishes big data from data that can be comfortably handled with traditional tools.

SGOU

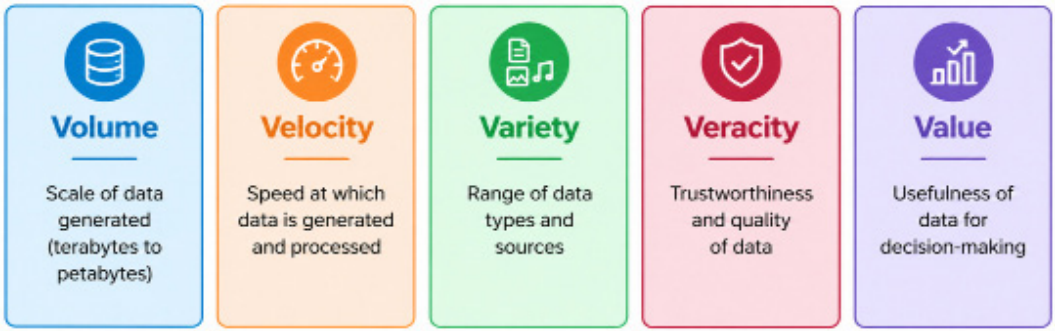


Fig. 2.1.1 The 5 Vs of Big Data

- **Volume:** The sheer quantity of data generated and stored, often measured in terabytes (1,000 gigabytes), petabytes (1,000 terabytes), or beyond. A single company logging user activity on a popular mobile app, for instance, might easily generate 2 terabytes of log data per day which, left uncontrolled, would accumulate to roughly 730 terabytes over the course of a single year (2 TB × 365 days).

- **Velocity:** The speed at which new data is generated and must be processed. Stock market feeds, social media posts, and sensor readings from industrial equipment can all arrive continuously, in real time, requiring systems that can ingest and act on data within seconds rather than waiting for a nightly batch update.

- **Variety:** The range of different formats and structures in which data arrives from neatly organized database tables, to semi-structured formats such as JSON or XML, to entirely unstructured formats such as emails, images, audio recordings, and video.

- **Veracity:** The trustworthiness, accuracy, and overall quality of the data. Big data is frequently noisy, incomplete, or inconsistent -- reinforcing the importance of the anomaly-detection techniques introduced in Block I, Unit 2, which become even more critical to apply correctly at big data scale.

- **Value:** The ultimate usefulness of the data for driving real decisions or insight. Collecting enormous volumes of data has little benefit if that data cannot be turned into meaningful, actionable conclusions -- which is precisely the purpose of the analytics techniques and tools introduced throughout the remainder of this course.

Together, the 5 Vs provide a practical checklist for recognizing when a data problem has moved beyond the reach of traditional tools such as a single spreadsheet or a conventional single-server database, and requires the kind of distributed technologies introduced in the following sections.

2.1.3 Sources and Types of Big Data

Big data is commonly classified into three broad types, based on how rigidly it is organized:

SGOU

- **Structured Data:** Data that is organized into a strict, predefined format, typically rows and columns, such as data stored in a traditional relational database or a spreadsheet. For example, a table of customer names, ages, and purchase amounts.
- **Semi-Structured Data:** Data that does not sit in a rigid table but still contains tags, labels, or other markers that give it some organizational structure, such as JSON or XML files commonly used to exchange data between web applications.
- **Unstructured Data:** Data with no predefined organizational structure at all, such as text documents, emails, social media posts, photographs, audio recordings, and video files. Unstructured data is estimated to make up the large majority of all data generated worldwide, and is the primary focus of Unit 3 of this block.

Table 2.1.1 Types of Big Data with Illustrative Sources and Storage Formats

Data Type	Example Sources	Typical Storage Format
Structured	Sales transaction records, employee databases, sensor readings logged in fixed fields	Relational database tables, spreadsheets
Semi-Structured	Website API responses, configuration files, email headers	JSON, XML, CSV with nested fields
Unstructured	Social media posts, customer service call recordings, product photographs, video content	Text files, images, audio/ video files

2.1.4 Introduction to the Hadoop Ecosystem

Apache Hadoop is an open-source software framework designed specifically to store and process very large datasets across clusters of ordinary, relatively inexpensive computers, working together as a single system. Rather than relying on one extremely powerful (and extremely expensive) machine, Hadoop distributes both the storage and the computation of a large dataset across many machines, known as nodes, that together form a cluster.

This distributed approach directly addresses the Volume and Velocity challenges of big data introduced in Section 2.1.2: as the size of a dataset grows, an organization can simply add more ordinary machines to its Hadoop cluster, rather than needing to purchase an ever more powerful single computer -- an approach known as horizontal scaling.

The Hadoop ecosystem is built around two foundational components, supported by a wider collection of additional tools:

SGOU

- **Hadoop Distributed File System (HDFS):** The storage layer of Hadoop, responsible for splitting large files into smaller blocks and distributing (and replicating) those blocks across the machines in a cluster, discussed in detail in Section 2.1.5.

- **MapReduce:** The original processing model of Hadoop, used to perform computations on data stored in HDFS by breaking a large task into many smaller tasks that can run in parallel across the cluster, discussed in detail in Section 2.1.6.

- **YARN (Yet Another Resource Negotiator):** The resource management layer of Hadoop, responsible for allocating the cluster's computing resources (processing power and memory) to different applications running on it, and for scheduling their tasks efficiently.

Beyond these three foundational components, a rich collection of additional tools has grown up around Hadoop to make it easier to use for specific tasks, several of which are introduced in Section 2.1.7.

2.1.5 Hadoop Distributed File System (HDFS)

HDFS is designed on the principle that, when working with extremely large files across many machines, hardware failure is not a rare exception but a routine, expected occurrence with potentially thousands of machines in a cluster, a hard disk failing on any given day is a statistical certainty rather than a surprise. HDFS is built from the ground up to tolerate such failures without losing data or interrupting processing.

HDFS achieves this through two key design principles: splitting files into blocks, and replicating those blocks across multiple machines.

Block Splitting: Rather than storing an entire large file on a single machine, HDFS splits it into fixed-size chunks called blocks (128 megabytes each, by default) and distributes these blocks across different machines in the cluster.

Worked Example 1: Consider a 300 MB file being stored in HDFS with the default block size of 128 MB. The number of blocks required is calculated by dividing the file size by the block size and rounding up to the next whole block:

Number of Blocks = $\lceil 300 \text{ MB} / 128 \text{ MB} \rceil = \lceil 2.34 \rceil = 3$ blocks (128 MB + 128 MB + 44 MB)

Replication: To guard against the loss of any single machine, each block is copied, or replicated, onto several different machines -- by default, three copies (a replication factor of 3) are kept, ideally on different physical machines and, where possible, in different server racks.

Worked Example 2: If an organization has 500 GB of raw data to store in HDFS with the default replication factor of 3, how much total storage capacity must the cluster provide?

Total Storage Required = Raw Data Size × Replication Factor = 500 GB × 3 = 1,500 GB (1.5 TB)

SGOU

This example highlights an important, often underappreciated trade-off in distributed storage systems: HDFS trades extra storage capacity (three times the raw data size, by default) for a very high degree of fault tolerance and, as a useful side benefit, the ability to read the same block from multiple machines simultaneously, improving overall read performance.

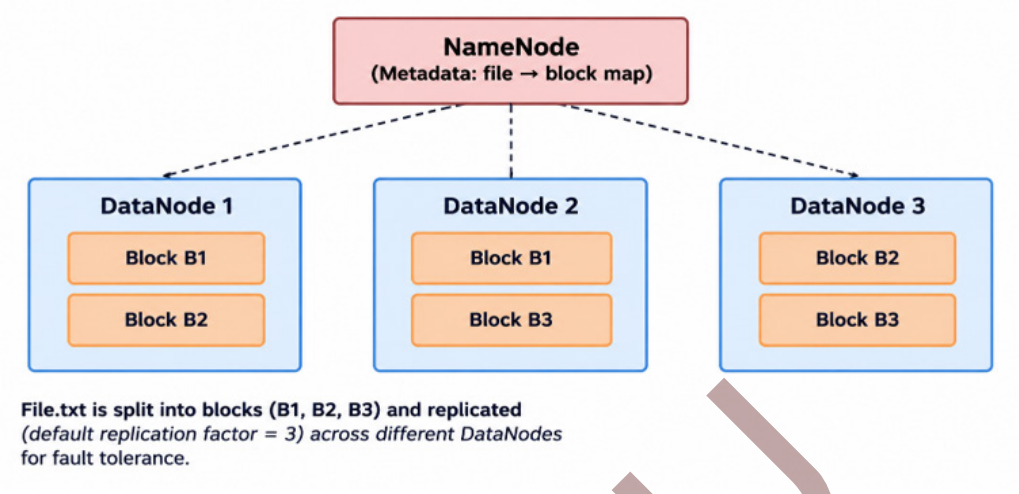


Fig. 2.1.2 Hadoop Distributed File System (HDFS) Architecture, Showing Blocks Replicated Across DataNodes

As shown in Figure 2.1.2, HDFS follows a master-worker architecture. The NameNode acts as the master, keeping track of the overall file structure and precisely which blocks make up each file and where each replica of every block is stored, but it does not store any of the actual file data itself. The DataNodes act as the workers, each storing and serving the actual data blocks assigned to it, and regularly reporting their status back to the NameNode. If a DataNode fails, the NameNode detects the failure and instructs other DataNodes holding replicas of the affected blocks to create fresh copies elsewhere, restoring the desired replication level automatically.

2.1.6 The MapReduce Programming Model

MapReduce is a programming model for processing large datasets in parallel across a Hadoop cluster, by breaking a computation down into two straightforward steps: a Map step and a Reduce step.

Map Step: Each machine in the cluster processes the portion of the data it holds locally, transforming each individual record into one or more intermediate key-value pairs.

Shuffle and Sort: The intermediate key-value pairs produced by every machine are automatically grouped together by key and distributed to the appropriate machine for the next step.

Reduce Step: Each machine combines (or 'reduces') all the values sharing the same key into a final summarized result.

SGOU

Worked Example 3 (Word Count): The classic illustration of MapReduce is counting how often each word appears across a large collection of text documents. Consider a tiny extract of text: 'the cat sat on the mat the cat ran'.

Table 2.1.2 A Simplified Word Count Walkthrough Using the MapReduce Model

Map Output (Key, Value)	After Shuffle and Sort (Key, Values)	Reduce Output (Key, Total)
(the,1), (cat,1), (sat,1), (on,1), (the,1), (mat,1), (the,1), (cat,1), (ran,1)	the: [1,1,1] cat: [1,1] sat: [1] on: [1] mat: [1] ran: [1]	the: 3 cat: 2 sat: 1 on: 1 mat: 1 ran: 1

In a real Hadoop cluster that processes millions of documents, the same Map and Reduce logic runs at the same time on many machines, each handling a small part of the text. Hadoop automatically manages the distribution of data, the shuffling of intermediate results and the recovery from any machine failure. This lets the analyst focus on the simple Map and Reduce logic instead of the complexity of coordinating a distributed computation.

While MapReduce was the original processing engine of Hadoop, more recent frameworks such as Apache Spark (introduced briefly in Section 2.1.7) have become increasingly popular for many workloads, largely because they can perform certain types of computations considerably faster than MapReduce by keeping intermediate data in memory rather than repeatedly writing it back to disk.

2.1.7 Other Components of the Hadoop Ecosystem

Writing raw MapReduce programs directly can be time-consuming and requires programming expertise. A wider ecosystem of tools has therefore grown up around Hadoop's core storage and processing layers, allowing analysts and engineers to work with big data more conveniently, often without needing to write low-level MapReduce code at all. Table 2.1.3 summarizes several of the most widely used tools in this ecosystem.

Table 2.1.3 Selected Tools in the Hadoop Ecosystem and Their Typical Purpose

Tool	Purpose	Typical User
Hive	Provides a SQL-like query language (HiveQL) for querying and analyzing data stored in HDFS, without writing MapReduce code directly	Data Analysts
Pig	Provides a scripting language (Pig Latin) for expressing data transformation and analysis tasks as a sequence of simple steps	Data Engineers

SGOU

HBase	A distributed, column-oriented database built on top of HDFS, designed for very fast random read/write access to large tables	Application Developers
Sqoop	Transfers bulk data efficiently between Hadoop and traditional relational databases	Data Engineers
Zookeeper	Coordinates and synchronizes activity across the many machines in a distributed cluster	System Administrators
Apache Spark	A fast, general-purpose distributed processing engine that can run on top of Hadoop, well suited to iterative and in-memory computations such as machine learning	Data Scientists

It helps to think of this ecosystem as a set of increasingly convenient layers built on top of HDFS and MapReduce. It is still possible to write raw MapReduce programs. However, tools such as Hive and Pig let an analyst express the same computation in more familiar, higher-level languages. Spark offers another processing engine that is often faster for many of the same tasks.

2.1.8 Introduction to Data Warehousing

While Hadoop and its ecosystem were built to handle the scale and variety of big data, a complementary, longer-established technology called the data warehouse remains central to how most organizations organize data specifically for business reporting and decision-making. A data warehouse is a large, centralized repository that stores integrated, historical data gathered from multiple operational source systems (such as a sales system, an inventory system, and a customer relationship management system), organized specifically to support fast, reliable querying and reporting.

To understand why data warehouses are designed the way they are, it helps to first distinguish between two very different categories of database system:

Table 2.1.4 Comparing OLTP and OLAP Systems

Aspect	**OLTP (Online Transaction Processing)**	**OLAP (Online Analytical Processing)**
Purpose	Records and manages day-to-day operational transactions	Supports complex analysis, reporting, and business intelligence

SGOU

Typical Operation	Insert, update, and delete single records quickly (e.g., processing one sale)	Read and aggregate large volumes of historical data (e.g., total sales by region, by quarter)
Example System	The checkout system at a retail store, processing each sale as it happens	A quarterly sales dashboard summarizing performance across all stores
Data Warehouse Role	Acts as a source system feeding data into the warehouse	Describes how the data warehouse itself is typically used

A data warehouse is built specifically to support the OLAP style of usage -- fast, flexible aggregation and analysis across large volumes of historical data -- which a typical OLTP system, optimized instead for quickly processing individual transactions, is usually poorly suited to handle efficiently at scale.

2.1.9 Data Warehouse Architecture and Schema Design

Data within a warehouse is typically organized using a dimensional model, built around two categories of table:

- **Fact Tables:** Store the measurable, numeric events of interest to the business. For example, individual sales transactions, including quantities and amounts along with references (keys) linking each record to relevant descriptive information.
- **Dimension Tables:** Store descriptive information that provides context for the facts for example, details about customers, products, dates, and store locations.

The most common and straightforward way of organizing fact and dimension tables is the star schema, in which a single, central fact table is directly connected to a set of surrounding dimension tables, resembling a star when diagrammed -- exactly as shown in Figure 2.1.3.

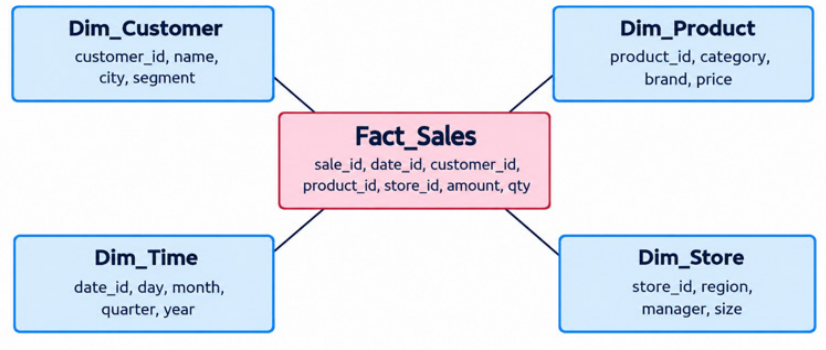


Fig. 2.1.3 A Star Schema for a Retail Sales Data Warehouse

SGOU

As Figure 2.1.3 shows, the central Fact_Sales table records individual sales transactions. The surrounding Dim_Customer, Dim_Product, Dim_Time and Dim_Store tables give the descriptive context for those transactions. By joining the fact table to the relevant dimension table, an analyst can easily calculate total sales by customer segment, product category, quarter or store region.

A related but more normalized design is the snowflake schema, in which dimension tables are further broken down into additional, smaller related tables. For example, splitting Dim_Product into a separate Dim_Product table and a further Dim_Category table -- reducing data redundancy at the cost of requiring more complex queries with additional joins. In practice, the star schema is generally preferred in data warehousing for its simplicity and query performance, while the snowflake schema is used selectively where reducing data redundancy is a higher priority.

2.1.10 The ETL Process

Before data from multiple, often inconsistent source systems can be loaded into a data warehouse's clean, well-structured schema, it must first pass through a process known as ETL -- Extract, Transform, Load.

- **Extract :** Data is pulled from one or more source systems, such as an OLTP sales database, a customer relationship management system, or external files.
- **Transform :** The extracted data is cleaned, validated, and reshaped into the format required by the warehouse -- for example, standardizing date formats, removing duplicate records, handling missing values, and applying the anomaly-detection and normalization techniques introduced in Block I, Unit 2.
- **Load:** The cleaned, transformed data is written into the appropriate fact and dimension tables of the data warehouse, ready for querying and analysis.

ETL is typically run on a regular schedule -- nightly, weekly, or in near real time for more advanced systems -- to keep the data warehouse continuously updated with fresh data from the underlying source systems. Dedicated ETL tools, such as Talend and Microsoft SQL Server Integration Services (SSIS), are commonly used to design, automate, and monitor this process, and are examined in greater depth in Unit 3 of this block.

2.1.11 Comparing Big Data Technologies and Traditional Data Warehousing

With both Hadoop-based big data technologies and traditional data warehousing now introduced, it is useful to compare them directly, since the two approaches are complementary rather than competing, and many modern organizations rely on both, often side by side.

SGOU

Table 2.1.5 Comparing Traditional Data Warehousing with Hadoop-Based Big Data Platforms

Aspect	Traditional Data Warehouse	Hadoop-Based Big Data Platform
Data Structure	Requires structured data, organized into a predefined schema before loading (schema-on-write)	Can store structured, semi-structured, and unstructured data, with structure applied only when the data is read (schema-on-read)
Typical Scale	Gigabytes to a few terabytes	Terabytes to petabytes and beyond
Hardware	Often relies on specialized, higher-cost servers	Runs on clusters of ordinary, relatively inexpensive machines
Best Suited For	Well-defined, recurring business reporting and dashboards	Exploratory analysis of large, varied, and rapidly changing datasets

In practice, many organizations use Hadoop-based platforms to store and first process large amounts of raw, varied data. They then load selected, summarized and cleaned subsets into a traditional data warehouse. The warehouse supports the fast, well-structured business reporting that OLAP systems are designed for. The two technologies therefore work together as parts of one analytics architecture. They are not direct substitutes for each other.

2.1.12 Applications in Data Analytics

The concepts introduced in this unit form the infrastructure layer that underlies almost every large-scale data analytics project:

- E-Commerce and Retail: Hadoop clusters store and process vast clickstream and transaction logs, while data warehouses power the daily and monthly sales dashboards used by store managers and executives.
- Telecommunications: Call detail records, generated in enormous volume and at high velocity, are processed using Hadoop-based tools to detect network issues and usage patterns, while customer billing summaries are maintained in traditional data warehouses.
- Healthcare: Unstructured data such as medical images and clinical notes are increasingly stored and analyzed using big data platforms, while structured patient records and hospital administrative data continue to rely on traditional data warehousing.

SGOU

- Financial Services: Big data platforms process high-velocity transaction streams for real-time fraud detection, while regulatory and financial reporting continues to rely on the well-structured, auditable design of traditional data warehouses.

Having introduced the foundational infrastructure and concepts of big data, the Hadoop ecosystem, and data warehousing, the learner is now well prepared for Unit 2 of this block, which introduces the data analytics life cycle -- the structured, step-by-step process by which raw data, once stored using the technologies introduced in this unit, is turned into actionable analytical insight.

Recap

- Big data refers to datasets so large, fast-moving, or varied that traditional data-processing tools and single-machine systems cannot handle them efficiently.
- The 5 Vs -- Volume, Velocity, Variety, Veracity, and Value -- describe the defining characteristics that distinguish big data from conventional data.
- Data can be structured (rigidly organized into tables), semi-structured (partially organized, such as JSON or XML), or unstructured (no predefined organization, such as text, images, or video).
- Apache Hadoop distributes both storage (via HDFS) and processing (via MapReduce, coordinated by YARN) across clusters of ordinary machines, allowing systems to scale horizontally as data grows.
- HDFS splits large files into blocks and replicates each block (by default, three times) across different machines, trading extra storage capacity for fault tolerance.
- MapReduce processes data in parallel using a Map step (transforming records into key-value pairs), a shuffle-and-sort step, and a Reduce step (combining values sharing the same key).
- Additional Hadoop ecosystem tools, such as Hive, Pig, HBase, Sqoop, and Spark, provide higher-level, more convenient ways to store, query, and process data without always requiring raw MapReduce code.
- A data warehouse is a centralized repository designed for OLAP-style reporting and analysis, in contrast to OLTP systems, which are designed for fast day-to-day transaction processing.
- Data warehouses are typically organized using a star schema (a central fact table surrounded by dimension tables) or, less commonly, a more normalized snowflake schema.

SGOU

- The ETL process (Extract, Transform, Load) moves data from source systems into a data warehouse, cleaning and reshaping it along the way; Hadoop-based platforms and traditional data warehouses are complementary technologies often used together in a single analytics architecture.

Objective Type Questions

1. Which type of system supports complex data analysis and reporting?
2. What are the 5 Vs of big data?
3. Which of the 5 Vs refers to the trustworthiness and quality of data?
4. What term describes the quantity of data generated and stored?
5. What is Apache Hadoop designed to do?
6. What is the default block size used by HDFS?
7. What is the default replication factor used by HDFS?
8. Which Hadoop component acts as the master, storing metadata about file blocks and their locations?
9. What are the two main steps of the MapReduce programming model?
10. Which Hadoop ecosystem tool provides a SQL-like query language for data stored in HDFS?
11. Which type of database system is designed for day-to-day operational transactions?
12. What is a fact table in a data warehouse?
13. Which characteristic of big data refers to the speed at which data is generated and processed?
14. What do the letters E, T, and L stand for in the ETL process?
15. What is the storage layer of Hadoop called?

SGOU

Answers

1. OLAP
2. Volume, Velocity, Variety, Veracity, and Value
3. Veracity
4. Volume
5. To store and process very large datasets across clusters of ordinary computers
6. 128 megabytes
7. 3 (three copies of each block)
8. The NameNode
9. The Map step and the Reduce step
10. Hive
11. OLTP
12. A table storing measurable, numeric business events (such as sales transactions) along with keys linking to dimension tables
13. Velocity
14. Extract, Transform, and Load
15. HDFS

SGOU

Assignments

1. A video-sharing platform generates 5 TB of new video data per day. Calculate the total volume of raw data generated over a 30-day month, and, assuming HDFS is used with the default replication factor of 3, calculate the total storage capacity the platform's Hadoop cluster must provide for that month's data alone.
2. A 700 MB log file is to be stored in HDFS using the default block size of 128 MB. Calculate the number of blocks the file will be split into, and state the size of the final (smallest) block.
3. Using an example of your own (different from the ones used in this unit), classify a real-world data source as structured, semi-structured, or unstructured, and justify your classification.

4. Design a simple star schema for a college library that wants to analyze book borrowing patterns. Identify a suitable fact table and at least three dimension tables, listing a few plausible columns for each.

5. Explain, in your own words, why a company might choose to use both a Hadoop-based big data platform and a traditional data warehouse, rather than relying on just one of the two technologies.

Suggested Reading

1. Kaur, A., & Madan, H. (2020). Introduction to Big Data and Analytics: Understanding of Hadoop and its Ecosystem. Notion Press.

2. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

3. White, T. (2015). Hadoop: The Definitive Guide (4th ed.). O'Reilly Media.

Reference

1. Kaur, A., & Madan, H. (2020). Introduction to Big Data and Analytics: Understanding of Hadoop and its Ecosystem. Notion Press.

2. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

3. White, T. (2015). Hadoop: The Definitive Guide (4th ed.). O'Reilly Media.

SGOU



UNIT 2 Data Analytics Life Cycle

Learning Outcomes

After completion of this unit, learner will be able to:

- explain the purpose and phases of the Data Analytics Life Cycle
- describe the key activities of the Discovery phase
- explain the role of the analytic sandbox in Data Preparation
- identify suitable analytical techniques during Model Planning
- explain the use of training and test sets in Model Building
- explain the iterative nature of the Data Analytics Life Cycle

SGOU

Prerequisite

Imagine that a telecommunications company discovers that many of its long-standing customers are cancelling their contracts. The company wants to understand why this is happening and identify customers who may leave in the future. Would simply collecting customer records be enough to solve this problem? Not really! The company must first understand the problem, gather relevant data, prepare it for analysis, choose suitable analytical methods, and test whether the results are reliable. This is how a data analytics project moves from a business question to a meaningful solution.

Before exploring this process, think about how you usually approach a problem. You identify what needs to be solved, collect relevant information, examine possible solutions, and check whether your approach works. These familiar steps provide a useful foundation for understanding the data analytics life cycle. A basic awareness of data, databases, and simple statistical concepts will also help you connect the ideas introduced in this unit. As you progress, you will discover how the different phases guide an analytics team, why data preparation and model testing are essential, and how an analytical model can ultimately support real-world decisions. No previous experience in building analytical models is required; a willingness to investigate problems and learn through examples is enough to get started.

Keywords

Data Analytics Life Cycle, Discovery, Analytic Sandbox, Data Preparation, Data Conditioning, Model Planning, Model Building, Training Set, Test Set, Iterative Process.

Discussion

2.2.1 Introduction to the Data Analytics Life Cycle

A data analytics project rarely succeeds by simply opening a dataset and immediately building a model. Without a clear, structured process, projects frequently run over budget, produce results that do not actually answer the business question that was originally asked, or fail entirely once the realities of messy, incomplete data are encountered. The data analytics life cycle is a structured framework that guides a project from an initial business question through to a working, validated analytical result, helping teams avoid exactly these pitfalls.

A widely used version of this life cycle, developed by EMC (now part of Dell Technologies) and adopted broadly across the data analytics industry, organizes a project into six phases: (1) Discovery, (2) Data Preparation, (3) Model Planning, (4) Model Building, (5) Communicate Results, and (6) Operationalize. This unit focuses in depth on the first four of these phases, Discovery, Data Preparation, Model Planning, and Model Building, which together take a project from a vague business problem to a working, tested analytical model, with the remaining two phases introduced briefly in Section 2.2.6 for completeness.

Throughout this unit, one running case study makes each phase concrete. A telecom company has noticed that a growing number of long-standing customers are cancelling their contracts (known in the industry as customer churn). The company wants to understand why this is happening. Ideally, it also wants to predict which customers are likely to leave next, before they actually go.

It is worth noting that the data analytics life cycle described in this unit is deliberately similar in spirit to the general scientific method: form a question, gather and examine evidence, propose and test explanations, and refine the conclusion in light of what is learned. What the life cycle adds, specifically for data analytics projects, is attention to the practical realities of working with real organizational data -- where it is stored, who controls access to it, how messy it typically is, and how a technically sound model must still be translated into something a business can actually act upon. Keeping this broader parallel in mind can make each of the four phases introduced in this unit feel less like an arbitrary checklist and more like a natural, disciplined way of answering any data-driven question responsibly.

2.2.2 Phase 1: Discovery

Discovery is the foundational phase of the data analytics life cycle, during which

SGOU

the project team works to thoroughly understand the business problem, the resources available, and the constraints under which the project must operate, before any data is seriously analyzed. Skipping or rushing this phase is one of the most common causes of analytics projects that ultimately fail to deliver useful results, since a technically excellent model built to answer the wrong question is of little value to the business.

Key activities during the Discovery phase typically include:

- **Understanding the Business Domain:** Learning enough about the business context for the telecom case study, this means understanding how contracts, billing cycles, and customer support interactions work to be able to interpret the data meaningfully later on.
- **Identifying Stakeholders and Resources:** Establishing who is sponsoring the project, who will use its results, and what data, personnel, and technical resources (such as the Hadoop cluster and data warehouse introduced in Unit 1) are available.
- **Framing the Business Problem as an Analytics Problem:** Translating a general business concern ('too many customers are leaving') into a specific, answerable analytics question ('can we predict, using data available at least one month in advance, which customers are likely to cancel their contract in the next 30 days?').
- **Formulating Initial Hypotheses:** Drawing on business knowledge and past experience to propose plausible explanations that the data can later help confirm or refute, for example, hypothesizing that customers who contact customer support more than three times in a month are more likely to churn.

Because analytics projects typically involve people with very different backgrounds and priorities, it is useful during Discovery to clearly identify the roles involved, several of which are summarized in Table 2.2.1.

Table 2.2.1 Common Roles in a Data Analytics Project

Role	Typical Responsibility
Business User	Understands the domain and typically uses the final results to make a decision (e.g., the marketing manager who wants to reduce churn)
Project Sponsor	Provides the project's funding and priorities, and defines what a successful outcome looks like
Project Manager	Ensures the project meets its objectives, on time and within budget, and manages coordination between the team members
Business Intelligence Analyst	Provides domain-specific context, historical reporting, and dashboards relevant to the business problem

SGOU

Data Engineer	Provides technical expertise in areas such as the Hadoop ecosystem and data warehousing (Unit 1), and assists with loading and preparing data
Database Administrator (DBA)	Manages access permissions and the configuration of the databases the project needs to draw data from
Data Scientist	Provides expertise in analytical techniques, model building, and interpreting the results in the context of the business problem

Case Study -- Discovery : For the telecom churn project, Discovery would involve three tasks. First, meet the customer retention team (the business users) to understand what 'churn' means for this company, for example a contract that is not renewed within 30 days of expiry. Second, confirm with the data engineering team what customer data already exists in the data warehouse and the Hadoop cluster. Third, propose initial hypotheses, such as 'customers on older contract plans churn more often' or 'a spike in customer support complaints often comes before a cancellation.'

2.2.3 Phase 2: Data Preparation

Once the business problem is well understood and the relevant data sources have been identified, the Data Preparation phase focuses on assembling, exploring, and conditioning the data so that it is ready to support serious analysis. This phase usually takes a large share of the time and effort in an analytics project. It is widely reported in industry that data preparation alone can take 50 to 80 percent of the total time, because real-world data is rarely clean, complete or well organized.

A central concept introduced during this phase is the analytic sandbox: a dedicated workspace, separate from an organization's production systems, where the project team can freely load, explore, transform, and experiment with data without any risk of disrupting the live systems that the rest of the business depends on.

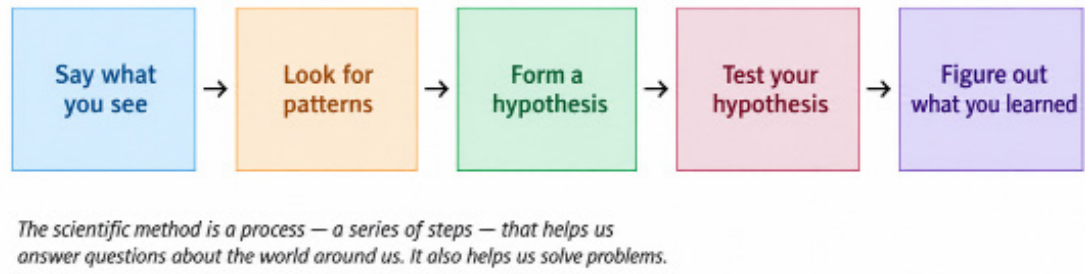


Fig. 2.2.1 Data Flow Through the Analytic Sandbox During Data Preparation

As Figure 2.2.1 shows, data is first extracted from source systems (such as the OLTP billing system and the data warehouse from Unit 1) into the analytic sandbox. There it is transformed and conditioned. This is sometimes called ETLT (Extract, Transform, Load, Transform again), because data is often transformed both before and after loading

SGOU

into the sandbox. The sandbox allows more flexible, exploratory transformations than a production ETL pipeline usually does.

Key activities during Data Preparation include:

- **Data Conditioning:** Cleaning, correcting, and standardizing the data, including the anomaly-detection and normalization techniques introduced in Block I, Unit 2, such as identifying and handling outliers and rescaling variables onto comparable scales.
- **Exploratory Data Analysis (EDA):** Calculating summary statistics (as introduced in Block I, Unit 1) and creating simple visualizations to get an initial feel for the data before deeper modelling begins.
- **Handling Missing and Inconsistent Data:** Deciding how to treat missing values, duplicate records, and inconsistently formatted fields (for example, phone numbers stored in several different formats across different source systems).

Case Study Data Preparation: For the telecom churn project, the team would extract customer billing history, contract details and customer support logs from their source systems into an analytic sandbox. They would then condition the data in three ways. They would standardize contract-type labels that may be recorded differently in older and newer systems. They would check customer tenure and monthly bill amounts for unlikely values using the z-score or IQR methods from Block I, Unit 2. They would also calculate simple summary statistics on support call frequency to get an initial feel for the data before deeper analysis.

2.2.4 Phase 3: Model Planning

With clean, well-understood data in hand, the Model Planning phase focuses on determining the methods, techniques, and workflow to be used in the subsequent Model Building phase. This phase is where the project team explores relationships between variables in greater depth and narrows down a shortlist of candidate analytical techniques suited to the business problem defined during Discovery.

Key activities during Model Planning include:

- **Exploring Variable Relationships:** Investigating which variables in the dataset appear to be associated with the outcome of interest -- for example, checking whether customers with more customer support complaints do, in fact, show a noticeably higher churn rate in the historical data.
- **Selecting Candidate Techniques:** Identifying which analytical or statistical techniques are best suited to the business problem. Since the telecom churn problem is fundamentally about predicting one of two outcomes for each customer (churn or no churn), techniques for classification -- such as logistic regression, decision trees, or the Naive Bayes classifier, all introduced in detail in Block III, Unit 1 of this course -- would be identified here as strong candidates.

SGOU

- **Planning the Analytical Workflow:** Deciding how the data will be divided for the purposes of building and testing a model, what software or platform will be used (several options for which are introduced in Unit 4 of this block), and what the criteria for a 'successful' model will be.

Case Study -- Model Planning : For the telecom churn project, Model Planning would first confirm, through exploratory analysis, that variables such as contract type, monthly bill amount and support call frequency are related to churn in the historical data. The team would then shortlist classification techniques, for example a decision tree and a Naive Bayes classifier. These suit the task of predicting whether each customer 'will churn' or 'will not churn'.

2.2.5 Phase 4: Model Building

The Model Building phase is where the analytical techniques shortlisted during Model Planning are actually implemented, trained, and evaluated using the prepared data. A central practice in this phase is dividing the available data into a training set, used to build (fit) the model, and a separate test set, held back and used only to evaluate how well the finished model performs on data it has not already seen.

Keeping the test set completely separate from the training process is essential: a model that is evaluated only on the same data it was built from will tend to appear far more accurate than it will actually be when applied to genuinely new customers in the future -- a problem known as overfitting.

Worked Example -- Case Study Data Split: Suppose the telecom company's analytic sandbox contains cleaned, prepared data for 10,000 customers, of whom, historically, 15% eventually churned. If the project team decides to use a 70/30 training-test split, the resulting sample sizes are:

$$\text{Training Set} = 70\% \times 10{,}000 = 7{,}000 \text{ customers}$$
$$\text{Test Set} = 30\% \times 10{,}000 = 3{,}000 \text{ customers}$$

If the same 15% churn rate is preserved proportionally in both the training and test sets -- a practice known as stratified sampling, used to ensure that both sets remain representative of the overall population -- the number of churned customers in each set would be:

$$\text{Churned in Training Set} = 15\% \times 7{,}000 = 1{,}050 \text{ customers}$$

$$\text{Churned in Test Set} = 15\% \times 3{,}000 = 450 \text{ customers}$$

As Figure 2.2.2 illustrates, only 15% of customers in this dataset actually churned -- a pattern known as class imbalance, which is extremely common in real-world business problems such as churn prediction, fraud detection, and rare-disease screening (recalling the base-rate discussion in Block I, Unit 4). Class imbalance must be handled carefully during Model Building, since a naive model that simply predicted 'no churn' for every single customer would already be correct 85% of the time, despite being completely useless for identifying the customers the business actually wants to retain.

SGOU

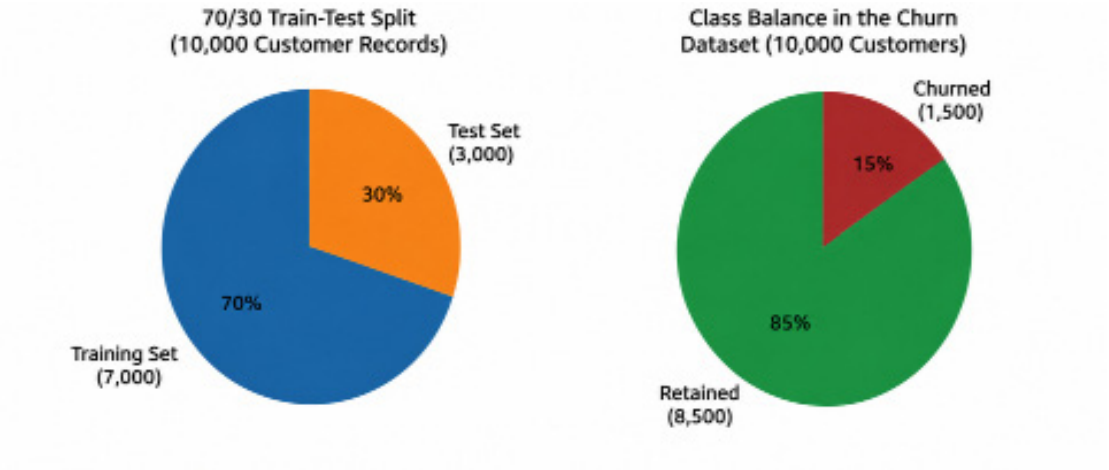


Fig. 2.2.2 Train-Test Split and Class Balance for the Telecom Churn Case Study

Model Building is usually iterative. An initial model is built and then evaluated on the test set using suitable performance measures (some of these, including the correlation coefficient and root mean square error, are introduced in Block III, Unit 2). The model is then refined, perhaps by changing the technique, adding new variables or revisiting decisions from Model Planning, until the result is satisfactory.

It is also common practice, particularly with limited data, to carve out a third portion of the data -- a validation set -- used during the iterative refinement process itself, reserving the test set purely for a final, one-time evaluation once all refinement is complete. This extra discipline further protects against a subtler version of overfitting, in which a model is repeatedly tuned against the same test set until it performs well on that specific set by chance, rather than because it has genuinely learned to generalize to new, unseen customers.

2.2.6 The Iterative and Non-Linear Nature of the Life Cycle

Although Discovery, Data Preparation, Model Planning, and Model Building have been presented here as four sequential phases, in practice the data analytics life cycle is rarely a strict, one-directional sequence. It is far more common for a team to move backward as well as forward between phases as new information emerges, exactly as illustrated by the circular arrows in Figure 2.2.3.

For example, if Exploratory Data Analysis during Data Preparation reveals that a key variable assumed to be available (such as detailed customer support call transcripts) is, in fact, missing or unusable for most customers, the team may need to return to Discovery to revise the scope of the project. Similarly, if Model Building reveals that none of the shortlisted techniques perform adequately, the team may need to return to Model Planning to consider additional candidate techniques, or even to Data Preparation to engineer new variables that better capture the patterns driving churn.

SGOU

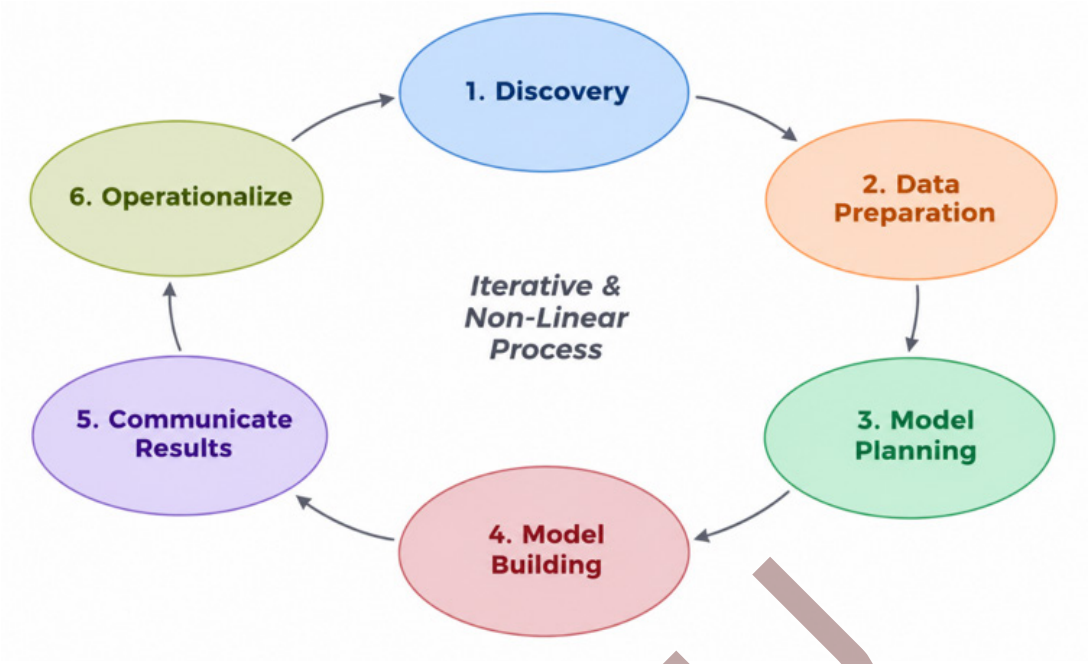


Fig. 2.2.3 The Six Phases of the Data Analytics Life Cycle

For completeness, the two phases of the life cycle beyond the scope of this unit are briefly described below:

- **Phase 5, Communicate Results:** The project team works with stakeholders to decide how best to present the findings. They weigh the strength and importance of each conclusion and turn technical results into clear business recommendations. For example, the finished churn model's predictions can be presented as a prioritized list of at-risk customers for the retention team to act on.
- **Phase 6, Operationalize:** The validated model (or analytical process) is deployed into the organization's live production environment, often integrated into existing business systems, and its performance is monitored on an ongoing basis to ensure it continues to perform well as new data arrives over time.

Understanding this iterative, feedback-driven structure is one of the most important lessons of this unit. A team that treats the life cycle as a rigid checklist, completed once and in order, is more likely to produce a result that is technically complete but not useful. A team that is willing to revisit earlier phases when the data or results demand it does better.

2.2.7 Common Challenges and Best Practices Across the Life Cycle

Certain challenges recur across almost every data analytics project, regardless of the specific business problem being addressed, and the data analytics life cycle helps teams anticipate and manage them systematically.

SGOU

Table 2.2.2 Common Challenges Across the Data Analytics Life Cycle and Recommended Practices

Common Challenge	Life Cycle Phase Most Affected	Recommended Practice
Business problem is too vague to be answered with data	Discovery	Invest time upfront translating the business question into a specific, measurable analytics question
Data is incomplete, inconsistent, or contains many outliers	Data Preparation	Use the analytic sandbox to explore and condition data thoroughly before modelling begins
No single technique is obviously the right choice	Model Planning	Shortlist and later compare multiple candidate techniques rather than committing to only one upfront
Model performs very well on training data but poorly in practice	Model Building	Always hold back a separate test set, and be alert to the risk of overfitting

A common theme in all these practices is the importance of documentation and communication. Teams should record the business question, the data sources used, the techniques considered (and why others were rejected) and the assumptions made at every phase. This keeps the reasoning behind the final model clear and lets people revisit it if circumstances change. This discipline becomes more important as projects grow in size and business impact.

This documentation discipline also pays off well beyond the lifetime of a single project. Suppose that a year later the retention team asks why the churn model flags some customers and not others. Or suppose a new team member needs to understand decisions made long before they arrived. A well-documented record of each phase (what was tried, what worked and what did not) turns a confusing, time-consuming investigation into a simple matter of reading the project's own history.

2.2.8 Applications of the Data Analytics Life Cycle in Practice

The structured, iterative approach introduced in this unit is applied, in some form, across virtually every serious data analytics initiative:

SGOU

- Customer Analytics: Churn prediction, as illustrated throughout this unit's case study, along with customer segmentation and lifetime value estimation, all follow essentially the same four-phase structure introduced here.
- Healthcare Analytics: Predicting patient readmission risk requires the same careful Discovery phase (defining what counts as a 'readmission' and understanding clinical workflows) before any data preparation or modelling can meaningfully begin.
- Financial Services: Credit risk scoring and fraud detection projects rely heavily on a disciplined Data Preparation phase, given the sensitivity and regulatory scrutiny applied to financial data.
- Manufacturing and Operations: Predicting equipment failure (predictive maintenance) follows the same life cycle, typically drawing on the big data technologies introduced in Unit 1 to handle high-velocity sensor data during Data Preparation.

Now that you have learned how to structure an analytics project, from the first business question to a working, tested model, you are ready for Unit 3 of this block. That unit looks more closely at unstructured data, one of the hardest and fastest-growing kinds of data met during the Data Preparation phase, and at the ETL tools used to bring it into a usable form.

Recap

- The Data Analytics Life Cycle consists of six phases: Discovery, Data Preparation, Model Planning, Model Building, Communicate Results, and Operationalize.
- Discovery involves understanding the business problem, identifying stakeholders and resources, and developing initial hypotheses.
- Data Preparation involves collecting, cleaning, exploring, and organizing data for analysis. It often takes the most time and effort.
- Model Planning involves studying relationships between variables and selecting suitable analytical techniques.
- Model Building involves developing, training, and testing analytical models to evaluate their performance and prevent overfitting.
- Class Imbalance occurs when one category has significantly more records than another. It can lead to misleading accuracy and poor model performance.
- The Data Analytics Life Cycle is iterative, meaning that teams may return to earlier phases when changes are needed.
- Communicate Results involves presenting findings and recommendations to support business decisions.

SGOU

- Operationalize involves deploying the validated model for practical use and monitoring its performance.

Objective Type Questions

1. What are the six phases of the data analytics life cycle?
2. Which phase of the life cycle focuses on understanding the business problem and identifying stakeholders?
3. What is an analytic sandbox?
4. Approximately what proportion of total project time is often spent on the Data Preparation phase?
5. What is the dedicated workspace used to explore and transform data called?
6. Which sampling method helps maintain class proportions in training and test sets?
7. What is the purpose of splitting data into a training set and a test set?
8. What is overfitting?
9. What is class imbalance?
10. Which phase involves studying relationships between variables?
11. Which phase involves deploying a validated model into practical use?
12. Which role in a data analytics project is primarily responsible for providing funding and defining project priorities?

SGOU

Answers

1. Discovery, Data Preparation, Model Planning, Model Building, Communicate Results, and Operationalize
2. Discovery
3. A dedicated, isolated workspace for freely exploring, transforming, and modelling data without affecting production systems
4. 50 to 80 percent
5. Sandbox
6. Stratified sampling

7. To evaluate how well a model performs on data it has not already seen, guarding against overfitting
8. When a model appears highly accurate on its training data but performs poorly on new, unseen data
9. When one outcome in the data is much rarer than another, such as churned customers being far fewer than retained customers
10. Planning
11. Operationalize
12. Project Sponsor

Assignments

1. Choose a real-world business problem of your own (for example, predicting which students are at risk of failing a course, or which products are likely to run out of stock). Write a short Discovery-phase summary identifying the business question, at least two stakeholders, and two plausible initial hypotheses.
2. For the problem you chose in Question 1, describe what data preparation activities you might need to perform, referring specifically to at least one technique from Block I, Unit 2 (anomalies and normalization).
3. A retail company's analytic sandbox contains cleaned data for 8,000 customers, of whom 20% historically made a repeat purchase within 30 days. If the team uses an 80/20 training-test split with stratified sampling, calculate the number of customers in each set, and the number of repeat purchasers expected in each set.
4. Explain, in your own words, why building a model without holding back a separate test set can lead to a misleadingly optimistic view of how well the model will perform in practice.
5. Describe a realistic scenario in which a project team, partway through Model Building, would need to go back to an earlier phase of the data analytics life cycle. Identify which phase they would return to and why.

SGOU

Suggested Reading

1. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.
2. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services / Wiley.
3. Kaur, A., & Madan, H. (2020). Introduction to Big Data and Analytics: Understanding of Hadoop and its Ecosystem. Notion Press.

Reference

1. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.
2. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services / Wiley.
3. Kaur, A., & Madan, H. (2020). Introduction to Big Data and Analytics: Understanding of Hadoop and its Ecosystem. Notion Press.

SGOU



Unstructured Data and ETL

Learning Outcomes

After completion of this unit, learner will be able to:

- explain the concept and challenges of unstructured data
- describe the main stages of text preprocessing, including tokenization, lowercasing, stop-word removal, and stemming
- explain the Bag-of-Words model and its use in text analytics
- describe the basic applications of sentiment analysis, image analytics, audio analytics, and video analytics
- explain the role of ETL tools in extracting, transforming, and loading data for analysis
- describe the key features of Talend and SQL Server Integration Services (SSIS)
- compare Talend and SSIS and identify suitable situations for using each tool

SGOU

Prerequisite

Have you ever written a product review, shared your opinion on social media, or contacted customer support to report a problem? Although these messages may look like ordinary text, they often contain valuable information about customer needs, satisfaction, and expectations. Similarly, a photograph of a damaged product or a recording of a customer conversation may reveal details that cannot be easily understood from numbers and tables alone. But how can a computer analyze such information when it does not follow a fixed structure?

To explore this question, it is helpful to recall what you have already learned about structured data, databases, and the ETL process. You may be familiar with organizing information in tables, cleaning data before analysis, or using simple statistical techniques to identify patterns. These concepts provide a useful foundation for understanding how unstructured data can be processed and analyzed. In this unit, you will discover how text preprocessing and the Bag-of-Words model help convert text into a numerical

form, how sentiment analysis identifies opinions, and how image, audio, and video analytics extract useful information from different media. You will also learn how ETL tools such as Talend and SQL Server Integration Services (SSIS) help combine data from different sources. No previous experience with text analytics or ETL tools is required—your curiosity about how computers make sense of everyday information is all you need to begin.

Keywords

Unstructured Data, Text Analytics, Tokenization, Stop Words, Stemming, Bag-of-Words, Sentiment Analysis, Image Analytics, ETL, Talend, SSIS (SQL Server Integration Services).

Discussion

2.3.1 Introduction to Unstructured Data

As introduced in Unit 1, unstructured data is data that has no predefined organizational structure, no fixed rows and columns, no consistent fields to fill in. Customer emails, call-centre transcripts, product reviews, social media posts, photographs and video recordings are all unstructured data. By most industry estimates, unstructured data makes up 80 to 90 percent of all data generated by organizations today. This share keeps growing as customers deal with businesses through text messages, voice calls and social media rather than structured forms.

This creates an important gap. An organization's data warehouse, built around the fact and dimension tables from Unit 1, answers questions like 'what were total sales last quarter?' very well. But it cannot directly answer a question like 'what are customers actually unhappy about, based on what they have written to us?' To answer that kind of question, we need special analytical techniques, several of which are introduced in this unit, together with ETL tools that bring such data into a form ready for analysis.

It is worth being precise about what makes unstructured data harder to work with, rather than simply asserting that it is. A structured billing record has a fixed number of clearly labelled fields like customer_id, bill_amount, due_date, so a computer program always knows exactly where to look for a given piece of information. A paragraph of free-form text has no such guarantee: the same information (say, a reason for dissatisfaction) might appear as the very first word of one complaint and be entirely absent, only implied, in another. Every technique introduced in this unit exists specifically to close that gap to take text, images, audio, or video and extract from it some smaller, more consistent, more structured summary that traditional analytical tools, including the very statistical techniques introduced throughout Block I of this course, can then be applied to.

SGOU

2.3.2 Challenges in Analyzing Unstructured Data

Unstructured data poses several challenges that do not arise when working with the neatly organized structured data introduced in Block I of this course:

- **No Fixed Fields:** There is no predefined column such as 'complaint_reason' to read directly; the reason a customer is unhappy, if mentioned at all, must be inferred from free-form sentences.
- **Ambiguity and Context-Dependence:** The same word or phrase can mean different things in different contexts. For example, the word 'crash' means something very different in a review of a laptop compared to a review of a car.
- **Volume and Variety of Expression:** The same underlying idea ('I am unhappy with my bill') can be expressed in an almost unlimited number of different ways, using different words, sentence structures, spelling, and even languages.
- **Non-Textual Formats:** Images, audio, and video require entirely different techniques from text, since the 'content' of interest an object in a photograph, a spoken word in a call recording is encoded in pixels or sound waves rather than in words at all.

Despite these challenges, unstructured data very often contains exactly the insight a business most needs. For example, the specific reasons behind customer dissatisfaction, which are rarely captured in a structured billing record but are frequently stated plainly, if messily, in a customer's own words. The remainder of this unit introduces foundational techniques for extracting that insight, beginning with text.

It is worth noting that these challenges are not reasons to avoid unstructured data, but rather reasons that specialized techniques rather than the descriptive statistics and structured-data methods introduced in Block I of this course are needed to work with it responsibly. Suppose an analyst tried to compute a 'mean' or a 'standard deviation' of a folder of customer complaint text without first preprocessing it and converting it to numbers. The result would be meaningless. The techniques in the rest of this unit bridge that gap. They convert free-form text into a form to which familiar analytical thinking can be applied again.

2.3.3 Text Analytics: Preprocessing and the Bag-of-Words Model

Before any meaningful analysis of text can take place, raw text is almost always passed through a text preprocessing pipeline, which systematically cleans and standardizes it into a small set of normalized tokens (individual words or word-like units) suitable for further analysis.

As shown in Figure 2.3.1, a typical pipeline consists of several stages, each of which is illustrated below using a real-looking customer complaint: 'I've been charged twice for my last two bills and nobody has responded to my emails!!!'

SGOU

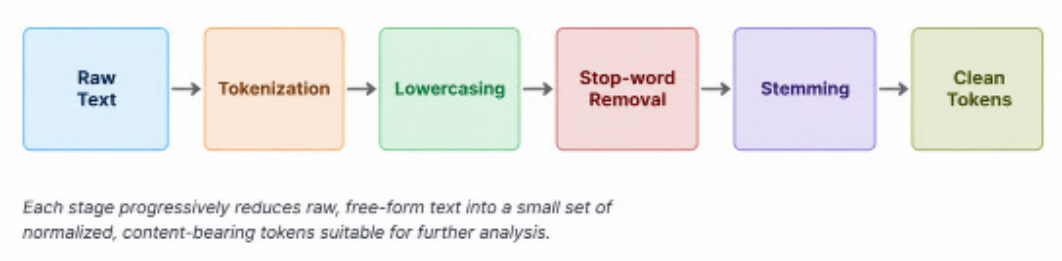


Fig. 2.3.1 A Typical Text Preprocessing Pipeline

- **Tokenization:** Splitting the text into individual units (tokens), typically words, and removing punctuation.
- **Lowercasing:** Converting all text to a single consistent case, so that 'Bill', 'BILL', and 'bill' are all treated as the same word.
- **Stop-word Removal:** Removing extremely common words (such as 'the', 'is', 'and', 'to') that carry little specific meaning on their own and would otherwise dominate any word-frequency analysis.
- **Stemming:** Reducing words to an approximate root form, so that related words such as 'charged', 'charges', and 'charging' are all treated as variants of the same underlying concept.

Table 2.3.1 A Worked Text Preprocessing Example

Stage	Result for the Example Complaint
Raw Text	"I've been charged twice for my last two bills and nobody has responded to my emails!!!"
Tokenization	ive, been, charged, twice, for, my, last, two, bills, and, nobody, has, responded, to, my, emails
Stop-word Removal	charged, twice, last, two, bills, nobody, responded, emails
Stemming	charg, twice, last, two, bill, nobodi, respond, email

Notice, in the final stemming row of Table 2.3.1, that the stemmed forms 'charg' and 'nobodi' are not themselves proper English words. This is an expected and well-known limitation of stemming, which applies simple, mechanical rules to chop word endings rather than genuinely understanding the word's grammar; a more linguistically careful alternative, called lemmatization, reduces words to their proper dictionary root form (for example, 'responded' to 'respond', but 'better' to 'good') at the cost of being more computationally involved. For most large-scale text analytics tasks, the speed and simplicity of stemming is judged to outweigh the slightly imperfect output it produces.

Once a collection of documents has been preprocessed in this way, it must be represented as numbers before most statistical or machine learning techniques can be applied. This includes the classification techniques introduced in Block II, Unit 2 and

SGOU

Block III, Unit 1. One of the simplest and most widely used ways to do this is the bag-of-words model. It represents each document as a count of how often each word appears and ignores word order and grammar.

Worked Example -- Bag-of-Words: Consider five short, preprocessed customer complaints submitted to the telecom company's support portal. (1) 'billing error again this month please fix my bill'. (2) 'the support team never called me back about my billing issue'. (3) 'I am extremely unhappy with the poor customer support I received'. (4) 'my bill was late and support did not respond quickly'. (5) 'great support this time the agent fixed my billing problem fast'. Counting how often each content word (after stemming 'bill' and 'billing' to a common root) appears across all five complaints produces the frequencies shown in Figure 2.3.2.

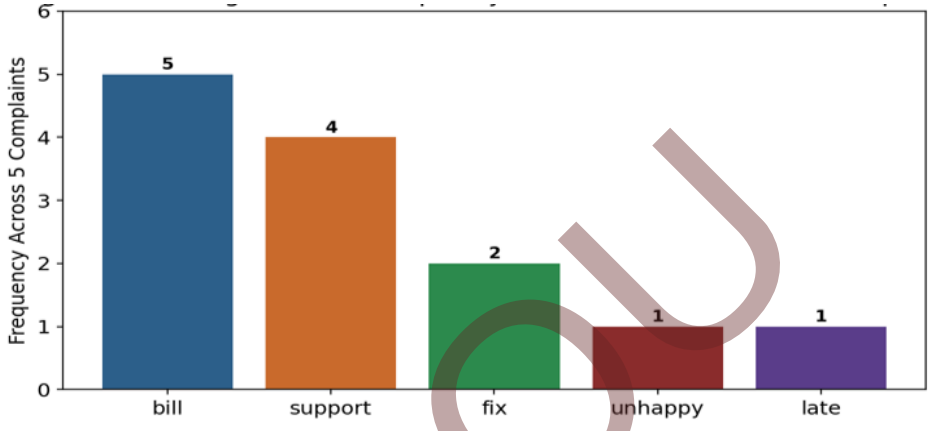


Fig. 2.3.2 Bag-of-Words Frequency Count from 5 Customer Complaints

As Figure 2.3.2 shows, the word 'bill' (combining 'bill' and 'billing') is the most frequently occurring term across these five complaints, appearing 5 times, followed by 'support' at 4 occurrences -- immediately suggesting, even from this tiny sample, that billing issues and support-quality issues are the two dominant themes among these particular complaints. In a real analytics project, this same bag-of-words representation, computed across thousands of complaints, could be fed directly into a classification technique to automatically categorize incoming complaints by theme, or used as an input feature, alongside the structured billing and contract data from Unit 2, in the churn-prediction model itself.

Sentiment Analysis: A closely related text analytics technique, sentiment analysis, attempts to automatically classify a piece of text as expressing a positive, negative, or neutral opinion, often using a lexicon-based approach that assigns a positive or negative score to individual words and combines them into an overall score for the document. A simple example shows why this task is harder than it looks. The sentence 'The service was terrible, but the staff were friendly and helpful' has one clearly negative word ('terrible') and two clearly positive words ('friendly', 'helpful'). A naive word-counting approach would therefore score it as positive overall, even though the sentence begins with a serious complaint about the service. Correctly handling such mixed sentiment, along with negation ('not helpful') and sarcasm, requires more sophisticated techniques than simple word counting, and is revisited in the context of forecasting and text analytics in Block III, Unit 4 of this course.

SGOU

2.3.4 Analytics for Other Unstructured Data Types

While text is one of the most common forms of unstructured data encountered in business analytics, images, audio, and video are increasingly important as well, and each requires its own specialized analytical approach, briefly introduced here at a conceptual level.

- **Image Analytics:** Techniques such as image recognition and computer vision analyze the patterns of pixels in a photograph or video frame to identify objects, faces, text, or defects -- for example, automatically detecting a damaged product on a manufacturing line from a camera image, or identifying a company's logo appearing in a social media photo.
- **Audio Analytics:** Speech-to-text (transcription) technology first converts spoken audio, such as a customer support call recording, into written text, after which the text analytics techniques introduced in Section 2.3.3 can be applied directly to the resulting transcript.
- **Video Analytics:** Combines image analytics (applied frame by frame) with an analysis of motion and change over time, used, for example, in retail stores to analyze customer foot-traffic patterns from security camera footage.

All three of these data types share a common thread. The raw data (pixels or sound waves) is first converted, using specialized techniques, into a more structured or text-based form. Examples are numeric feature values that describe an image, or a written transcript of spoken audio. After that, more familiar analytical techniques, including many introduced elsewhere in this course, can be applied. For the telecom churn project, this means that a folder of raw call-centre audio recordings would typically be run through speech-to-text transcription first, after which the resulting transcripts could be analyzed using exactly the text preprocessing and bag-of-words techniques introduced in Section 2.3.3.

Consider, briefly, how naturally this same pattern extends to the telecom company's other unstructured data sources. Its social media team might want to know whether a photo posted by a frustrated customer shows a damaged router the company shipped them -- a job for image analytics. Its network operations team might want to review hours of recorded customer support calls to identify how often background noise or a dropped call correlates with a subsequent complaint -- a job for audio analytics, and potentially video analytics if support is offered over a video call. In every case, the specific technique differs, but the underlying goal is the same one introduced throughout this unit: turning something unstructured and free-form into a smaller, structured summary that the rest of the analytics life cycle, introduced in Unit 2, can act upon.

SGOU

2.3.5 The Role of ETL in Preparing Unstructured and Structured Data

Recall from Block II, Unit 1 that ETL -- Extract, Transform, Load is the process of pulling data out of source systems, cleaning and reshaping it, and loading it into a data warehouse or analytic sandbox. While ETL was originally developed primarily for structured, tabular data moving between databases, modern ETL tools are equally important for bringing semi-structured and unstructured data such as the call-centre transcripts and social media mentions handed to the intern at the start of this unit -- into a project's analytic sandbox alongside more familiar structured data.

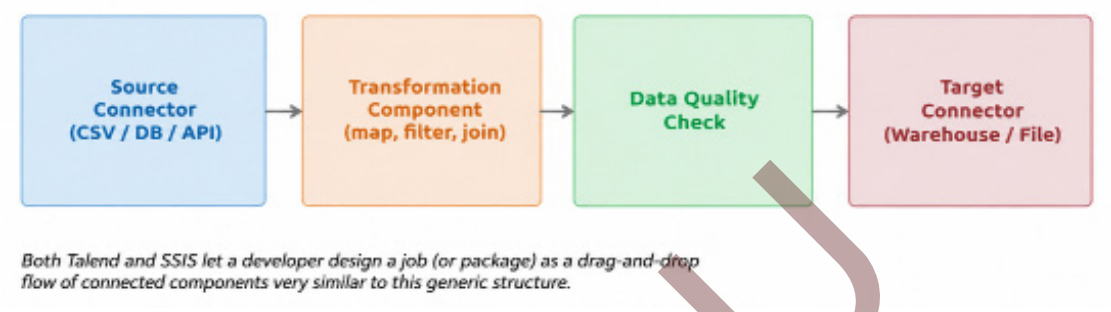


Fig. 2.3.3 A Generic ETL Job Flow, as Designed in Tools such as Talend or SSIS

As Figure 2.3.3 shows, dedicated ETL tools let a developer design a job (called a package in Microsoft's terminology) visually, instead of writing all the logic as raw computer code. The job is a flow of connected, configurable components: a source connector, one or more transformation components and a target connector. This greatly reduces the time and programming skill needed to build and maintain data pipelines. Two of the most widely used ETL tools in industry, Talend and SQL Server Integration Services (SSIS), are introduced in the following two sections.

2.3.6 ETL Tool: Talend

Talend is a widely used data integration platform, built around a visual, drag-and-drop job designer called Talend Open Studio, which is available in a free, open-source edition alongside paid, enterprise-grade editions offering additional features and support.

Key features of Talend include:

- A large library of pre-built connectors for a very wide range of data sources and targets, including relational databases, cloud storage services, Hadoop and Spark (introduced in Unit 1 of this block), and popular business applications such as Salesforce.
- A visual job designer in which a developer drags components (such as a database input, a filter, and a file output) onto a canvas and connects them into a flow, similar in spirit to Figure 2.3.3.

SGOU

- Underlying code generation: unlike some purely visual tools, Talend automatically generates Java code behind the scenes for each designed job, which can offer performance and deployment advantages for technically experienced teams.

- Native support for big data processing, including the ability to generate jobs that run directly on a Hadoop or Spark cluster, connecting the ETL tooling introduced in this unit directly to the big data technologies introduced in Unit 1 of this block.

Talend's combination of a free, open-source starting point and strong big data integration has made it a popular choice for organizations working with large volumes of varied data across many different source systems, including the kind of mixed structured, semi-structured, and unstructured data sources the telecom churn project draws from.

2.3.7 ETL Tool: SQL Server Integration Services (SSIS)

SQL Server Integration Services (SSIS) is Microsoft's ETL platform, bundled with Microsoft SQL Server and designed, using a visual tool called SQL Server Data Tools (SSDT), around two closely linked types of workflow within a single package (saved as a .dtsx file):

- Control Flow: Defines the overall sequence and logic of a package's execution. For example, running a data flow task, then sending an email notification, then archiving a processed file, with conditional branching and looping available where needed.

- Data Flow: Defines the specific movement and transformation of data itself, from a source (such as a SQL Server table or a flat file) through transformation components (such as sorting, aggregating, or looking up reference values) to a destination.

Key features of SSIS include:

- Deep, seamless integration with the Microsoft data platform, including SQL Server, Azure SQL Database, and Azure Data Factory, making it a natural choice for organizations already heavily invested in Microsoft's database and cloud ecosystem.

- A rich set of built-in transformation components covering common data-cleaning and reshaping tasks, many of which map directly onto the data conditioning activities introduced in Block II, Unit 2 of this course.

- Strong support for scheduling and automating packages to run on a regular basis, using SQL Server Agent, so that a data warehouse can be refreshed automatically overnight or at any other required interval.

Because SSIS is tightly bundled with SQL Server licensing, it is most commonly found in organizations that already use Microsoft's database platform as their primary transactional or data warehousing environment, making it a natural, low-friction choice for extending an existing SQL Server-based system with structured ETL capability.

SGOU

2.3.8 Comparing Talend and SSIS

Both Talend and SSIS allow a developer to design an ETL job visually, as a flow of connected components, but they differ in several practical respects that influence which tool an organization is likely to choose, summarized in Table 2.3.2.

Table 2.3.2 Comparing Talend and SSIS as ETL Tools

Aspect	Talend	SSIS
Licensing	Free, open-source edition available (Talend Open Studio), with paid enterprise editions	Bundled with Microsoft SQL Server licensing
Platform Ecosystem	Broad connectivity across many vendors and platforms, including strong Hadoop/Spark support	Deepest integration within the Microsoft ecosystem (SQL Server, Azure)
Underlying Technology	Generates Java code from visually designed jobs	Executes packages using the SQL Server Integration Services runtime engine
Typical Organization	Organizations with varied, multi-vendor, or big-data-heavy environments	Organizations standardized primarily on Microsoft's database and cloud platform

In practice, the choice between Talend, SSIS or another ETL tool depends less on which tool is 'better' in the abstract. It depends more on which tool best fits the organization's existing technology, the variety of source systems it must connect to, and the skills of its data engineering team. For example, an organization that already uses Microsoft SQL Server for everything will probably find SSIS a more natural fit than a separate platform such as Talend. The opposite is likely for an organization with a more varied, big-data-oriented environment.

It is also increasingly common for a single organization to use more than one ETL tool side by side, rather than picking exactly one for every purpose. A company might, for example, rely on SSIS for its established, SQL Server-based financial reporting pipelines, where stability and deep integration with existing systems matter most, while separately using Talend to bring newer, more varied big-data sources -- such as the unstructured call-transcript and social-media data introduced earlier in this unit -- into its Hadoop-based analytic sandbox. Recognizing that ETL tool choice need not be an all-or-nothing decision is itself a useful piece of practical judgement for any aspiring data analyst to carry forward into Unit 4, which surveys the broader landscape of analytics tools and platforms in a similar spirit.

SGOU

2.3.9 Applications of Unstructured Data Analytics and ETL in Practice

The techniques and tools introduced in this unit are applied across a wide range of real-world data analytics scenarios:

- **Customer Experience Analytics:** Text analytics applied to support tickets, call transcripts, and social media mentions -- exactly as in this unit's running churn case study -- helps organizations identify recurring themes in customer dissatisfaction that structured billing data alone would never reveal.
- **Brand and Social Media Monitoring:** Sentiment analysis is widely used to track public opinion about a brand or product in near real time across social media platforms.
- **Quality Control:** Image analytics is used on manufacturing lines to automatically detect visual defects in products far faster and more consistently than manual visual inspection.
- **Enterprise Data Integration:** ETL tools such as Talend and SSIS form the operational backbone connecting an organization's many source systems -- including the Hadoop-based platforms and data warehouses introduced in Unit 1 -- into the analytic sandboxes and reporting systems that the rest of this course builds upon.

Returning one final time to the telecom churn project that has run throughout this block: by the end of this unit, the intern is in a position to bring together three previously separate strands of work. The structured billing and contract data prepared in Unit 2 can now be joined with a numeric, bag-of-words summary of customer support transcripts and social media mentions. All of this is moved into her analytic sandbox using an ETL tool such as Talend or SSIS. The churn model can then use not only what a customer has been billed but also what the customer has said. This gives a much richer base for the Model Planning and Model Building phases from Unit 2.

Now that you have learned how to work with unstructured data and how ETL tools bring varied data sources into a form ready for analysis, you are ready for Unit 4 of this block. That unit surveys the wider range of data analysis tools and platforms, including Python, R, Power BI, Tableau and major cloud analytics services. These are used for the modelling and visualization work that follows data preparation.

SGOU

Recap

- Unstructured Data: Unstructured data includes text, images, audio, and video. It does not follow a predefined structure and can provide useful insights.
- Text Preprocessing: Text preprocessing cleans and prepares text for analysis through steps such as tokenization, lowercasing, stop-word removal, and stemming.
- Stemming and Lemmatization: Stemming reduces words to approximate root forms using simple rules. Lemmatization finds more accurate base forms but requires more processing.
- Bag-of-Words Model: The Bag-of-Words model represents a document using word counts. It ignores grammar and word order and helps in text classification.
- Sentiment Analysis: Sentiment analysis identifies text as positive, negative, or neutral. Negation, mixed opinions, and sarcasm can affect its accuracy.
- Image, Audio, and Video Analytics: These techniques convert images, sounds, and videos into structured or text-based forms for analysis.
- ETL Tools: ETL tools help users design data-processing workflows by connecting data sources, transformation components, and target systems.
- Talend and SSIS: Talend supports big data processing and connections to different platforms. SSIS is Microsoft's ETL tool and works closely with SQL Server and Azure.
- Choosing ETL Tools: The choice between Talend and SSIS depends on an organization's existing technologies and data sources. Neither tool is always the best choice for every organization.

SGOU

Objective Type Questions

1. What type of data includes text, images, audio, and video without a predefined structure?
2. Which type of data is organized into fixed rows and columns?
3. What type of data often contains customer emails, reviews, and social media posts?
4. What is the process of splitting text into individual words or units called?
5. Which preprocessing step converts all text into a consistent case?
6. What are common words such as “the”, “is”, and “and” called in text analytics?

7. Which preprocessing step removes common words that carry little specific meaning?
8. Which technique reduces words to approximate root forms by removing endings?
9. What is the process of converting words into their proper dictionary base forms called?
10. Which text representation uses word counts to represent a document?
11. Which ETL tool is developed by Microsoft?
12. What is the full form of SSIS?
13. Which ETL tool is generally suitable for organizations using varied platforms and big data technologies?

Answers

1. Unstructured
2. Structured
3. Unstructured
4. Tokenization
5. Lowercasing
6. Stopwords
7. Stop-word removal
8. Stemming
9. Lemmatization
10. Bag-of-Words
11. SSIS
12. SQL Server Integration Services
13. Talend

SGOU

Assignments

1. Take a short piece of text of your own choosing (a product review or social media post, at least two sentences long), and manually work through the text preprocessing pipeline introduced in this unit: tokenize it, remove stop words, and apply simple stemming rules of your own. Present your result in a table similar to Table 2.3.1.
2. Using the five preprocessed customer complaints given in the bag-of-words worked example in this unit, identify one additional word (besides 'bill' and 'support') that appears in more than one complaint, and state how many times it occurs in total.
3. Explain, using an example sentence of your own, why sentiment analysis based on simple word counting can produce an incorrect or misleading result.
4. Describe, in your own words, how a folder of raw customer call recordings could eventually be analyzed using the text analytics techniques introduced in this unit, naming every intermediate step required.
5. A retail company currently uses SQL Server for all of its transactional and reporting systems, but is beginning to collect large volumes of website clickstream data that it plans to store on a Hadoop cluster. Based on the comparison in this unit, would you recommend Talend, SSIS, or a combination of both for its ETL needs? Justify your answer.

Suggested Reading

1. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.
2. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services / Wiley.
3. Kaur, A., & Madan, H. (2020). Introduction to Big Data and Analytics: Understanding of Hadoop and its Ecosystem. Notion Press.

SGOU

Reference

1. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.
2. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services / Wiley.
3. Kaur, A., & Madan, H. (2020). Introduction to Big Data and Analytics: Understanding of Hadoop and its Ecosystem. Notion Press.

SGOU



Data Analysis Tools and Platforms

Learning Outcomes

After completion of this unit, learner will be able to:

- classify data analysis tools and platforms into programming languages, business intelligence tools, and cloud analytics platforms
- describe the key features and applications of Python and R for data analysis
- explain the features and uses of Power BI and Tableau for data visualization and reporting
- identify the major components of Microsoft Azure Analytics and AWS Analytics
- apply a weighted decision matrix to select a suitable data analysis tool for a given project

Prerequisite

Imagine that you are part of a team developing a data analytics project for a company. You have collected data, organized it, cleaned it, and prepared it for analysis. Now comes an important question: Which tools should you use to analyze the data, build a predictive model, and present the results in a way that helps people make decisions? Would you choose a programming language, a dashboard tool, or a cloud-based platform? The answer depends on what you want to achieve and the resources available to you.

Before exploring these choices, recall your basic understanding of data, databases, data preparation, and the data analytics life cycle from the previous units. You may have seen how tables organize information, how data is cleaned before analysis, and how analytical results can help solve real-world problems. These ideas provide a foundation for understanding the different tools used by data analysts. In this unit, you will explore programming languages such as Python and R, visualization tools such as Power BI and Tableau, and cloud analytics platforms such as Microsoft Azure and Amazon Web Services (AWS). You will also learn how to compare these tools based on project requirements, team skills, cost, and existing infrastructure. You do not need prior experience with these tools to begin—an interest in discovering how data is transformed into useful insights is a good starting point

SGOU

Keywords

Python, R, Business Intelligence (BI), Power BI, Tableau, Cloud Analytics, Microsoft Azure, Amazon Web Services (AWS), Data Warehousing, Weighted Decision Matrix.

Discussion

2.4.1 Introduction to Data Analysis Tools and Platforms

Having stored data (Unit 1), planned and structured a project (Unit 2), and prepared both structured and unstructured data (Unit 3), a data analyst must finally choose the actual software used to build models, create visualizations, and share results. No single tool is 'best' in every situation; different tools are optimized for different tasks, different levels of programming skill, and different organizational environments.

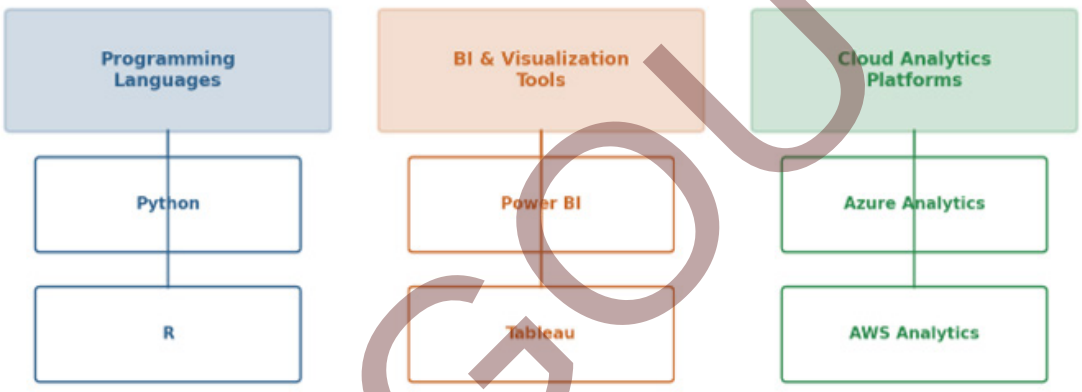


Fig. 2.4.1 The Landscape of Data Analysis Tools and Platforms

As shown in Figure 2.4.1, it is useful to group the wide range of available tools into three broad categories, which structure the remainder of this unit:

- **Programming Languages:** General-purpose or statistically-focused languages, such as Python and R, that offer maximum flexibility for custom analysis, automation, and machine learning, at the cost of requiring coding skill.
- **Business Intelligence (BI) and Visualization Tools:** Tools such as Power BI and Tableau, designed around drag-and-drop, largely code-free interfaces for building interactive dashboards and reports for a wide business audience.
- **Cloud Analytics Platforms:** Comprehensive, managed suites of services, such as Microsoft Azure Analytics and AWS Analytics, that bundle data warehousing, ETL, machine learning, and visualization capabilities together, hosted and scaled by a cloud provider rather than run on an organization's own servers.

These categories are not mutually exclusive in practice -- a single project, such as the telecom churn case study followed throughout this block, might reasonably use a

SGOU

programming language for model building, a BI tool for the final dashboard presented to executives, and a cloud analytics platform to host and scale the whole pipeline. The remainder of this unit examines each category in turn, before returning to the case study to work through exactly this kind of combined decision.

2.4.2 Programming Languages for Data Analysis: Python

Python is a general-purpose, open-source programming language that has become one of the most widely used languages in data analytics and data science, valued for its readable syntax and its vast ecosystem of specialized libraries. Rather than being built specifically for statistics, Python's strength in data analysis comes largely from a small number of powerful, widely adopted libraries built on top of the core language:

- **pandas :** Provides fast, flexible data structures (particularly the DataFrame) for cleaning, reshaping, and analyzing tabular data -- directly supporting many of the data preparation activities introduced in Unit 2.
- **NumPy :** Provides efficient support for numerical computation, particularly with large arrays and matrices of numbers, underpinning most other Python data-analysis libraries.
- **scikit-learn:** Provides a consistent, well-documented interface to a very wide range of machine learning techniques, including the classification techniques (such as decision trees and Naive Bayes) shortlisted in the Model Planning phase of Unit 2's case study.
- **Matplotlib and Seaborn:** Provide flexible charting and visualization capabilities for exploring and presenting data directly within Python code.

Because Python is a full general-purpose programming language, it is particularly well suited to tasks that require automation, custom logic, or integration with other software systems -- for example, automatically retraining the telecom churn model every month as new customer data arrives, or embedding a trained model directly into a company's customer service application. This flexibility does, however, require a genuine investment in learning to write and debug code, which is a less immediate skill than the drag-and-drop interfaces introduced later in this unit.

2.4.3 Programming Languages for Data Analysis: R

R is a programming language and free software environment developed specifically for statistical computing and graphics, and remains especially popular among statisticians, academic researchers, and analysts whose work centres on rigorous statistical modelling. R is most commonly used through RStudio, a dedicated integrated development environment (IDE) that brings together code editing, data viewing, plotting, and documentation in a single, analyst-friendly interface.

SGOU

Key strengths of R include:

- **CRAN (the Comprehensive R Archive Network):** A vast, curated repository of packages covering an exceptionally wide range of statistical methods, many of which are contributed directly by the academic researchers who developed the underlying statistical technique.
- **ggplot2:** A widely admired data visualization package, based on a coherent and expressive 'grammar of graphics', frequently used to produce publication-quality statistical charts.
- **A statistics-first design:** Because R was built from the outset for statistical analysis, many core statistical procedures -- from the descriptive statistics and probability distributions introduced in Block I of this course, through to advanced modelling techniques -- are available as simple, built-in functions requiring very little additional setup.

R is particularly well suited to projects with a strong emphasis on rigorous statistical modelling, exploratory data analysis, and publication-quality visualization, such as academic research or a specialized statistical modelling team, though it is somewhat less commonly used than Python for building large-scale, production machine learning systems or general-purpose software automation.

2.4.4 Comparing Python and R, and a Worked Decision-Matrix Example

Table 2.4.1 summarizes the main practical differences between Python and R, to guide the comparison that follows.

Table 2.4.1 Comparing Python and R for Data Analysis

Aspect	Python	R
Origin and Primary Design Goal	General-purpose programming language, later adopted widely for data analysis	Purpose-built from the outset for statistics and graphics
Ecosystem Strength	Broadest machine learning and general-purpose automation ecosystem	Deepest and most specialized statistical modelling ecosystem
Typical User	Data scientists and software/ ML engineers	Statisticians and academic researchers
Production Deployment	Very well suited to embedding models into live software systems	Less commonly used for large-scale production deployment

SGOU

Rather than declaring one language universally 'better', it is far more useful to apply a simple, transparent weighted decision matrix: list the criteria that matter for a specific project, assign each criterion a weight reflecting its relative importance, score each candidate tool against every criterion, and multiply and sum to reach an overall score.

Worked Example: For the telecom churn project's Model Building phase, suppose the team identifies four relevant criteria: ease of automating the monthly retraining process (weight 0.3), strength of the machine learning ecosystem (weight 0.3), quality of visualization for internal reporting (weight 0.2), and how quickly the existing team can learn the tool (weight 0.2). Suppose the team scores each language out of 10 on each criterion, based on their own research and skills:

Table 2.4.2 Weighted Decision Matrix for Python versus R on the Telecom Churn Project

Criterion (Weight)	Python Score	Weighted	R Score	Weighted
Automation (0.3)	9	2.7	6	1.8
ML Ecosystem (0.3)	9	2.7	7	2.1
Visualization (0.2)	7	1.4	9	1.8
Learning Curve (0.2)	7	1.4	6	1.2
Python Total = 2.7 + 2.7 + 1.4 + 1.4 = 8.2			R Total = 1.8 + 2.1 + 1.8 + 1.2 = 6.9	



Fig.2.4.2 Illustrative Weighted Decision-Matrix Scores for the Telecom Churn Project

For this specific project, with these specific weights, Python scores higher overall (8.2 versus 6.9), primarily due to its stronger fit for automation and machine learning deployment. It is essential to recognize that this result is not a universal verdict on Python versus R -- a different project, with a stronger emphasis on rigorous statistical modelling and a team of trained statisticians, might reasonably assign different weights and arrive at the opposite conclusion. The value of the weighted decision matrix lies precisely in making these assumptions explicit and open to challenge, rather than relying on an unstated, informal preference.

SGOU

2.4.5 Business Intelligence and Visualization Tools: Power BI

Power BI is Microsoft's business intelligence and data visualization platform, designed to let analysts and business users build interactive reports and dashboards largely through a drag-and-drop interface, with limited coding required for most everyday tasks. Key components of Power BI include:

- **Power Query:** A built-in data preparation tool for connecting to, cleaning, and reshaping data before it is loaded into a report -- functioning as a lightweight, user-friendly complement to the dedicated ETL tools (such as SSIS, introduced in Unit 3) used for larger, more complex data integration tasks.
- **Data Modelling and DAX:** Power BI allows relationships to be defined between multiple tables (echoing the star schema concepts from Unit 1) and supports a formula language called DAX (Data Analysis Expressions) for defining custom calculations and metrics.
- **Interactive Report Canvas:** A drag-and-drop canvas for building charts, tables, and filters into an interactive report that business users can explore themselves, without needing to write any code.

Because Power BI is developed by Microsoft, it integrates especially closely with the broader Microsoft data platform -- including SQL Server, Excel, Azure services, and, as seen in Unit 3, SSIS -- making it a particularly natural choice for organizations, like the telecom company followed throughout this block, that are already heavily invested in Microsoft's technology ecosystem.

2.4.6 Business Intelligence and Visualization Tools: Tableau

Tableau is a dedicated data visualization platform, widely recognized for the flexibility, polish, and interactivity of the visual dashboards it can produce. Like Power BI, Tableau is built around a largely drag-and-drop interface, but is often considered to offer somewhat greater flexibility for highly customized, exploratory visual analysis, owing to its underlying VizQL engine, which automatically translates drag-and-drop actions into the appropriate underlying query and visual representation.

Tableau is available in several forms, including Tableau Desktop (for individual report building), Tableau Server and Tableau Cloud (for sharing dashboards across an organization), and Tableau Public (a free version for publishing publicly shareable visualizations, widely used in journalism and education).

Tableau is generally platform-agnostic about its data sources. It connects easily to a very wide range of databases, cloud platforms and file formats, whatever the vendor. This makes it popular with organizations that have a more varied technology environment. It also suits teams whose main need is the richest exploratory visualization experience rather than close integration with one vendor's ecosystem.

SGOU

2.4.7 Comparing Power BI and Tableau

Table 2.4.3 summarizes the main practical differences between Power BI and Tableau.

Table 2.4.3 Comparing Power BI and Tableau

Aspect	Power BI	Tableau
Ecosystem Integration	Deepest integration within the Microsoft platform (Excel, SQL Server, Azure)	Broad, vendor-neutral connectivity across many different platforms
Visualization Flexibility	Strong, continuously improving, standard business charting	Often considered the most flexible for highly customized, exploratory visuals
Typical Cost Positioning	Generally positioned as the more cost-effective option, particularly for organizations already licensing Microsoft products	Generally positioned at a premium price point, reflecting its specialized visualization capabilities
Typical Organization	Organizations standardized on the Microsoft ecosystem	Organizations prioritizing maximum visualization flexibility across varied data sources

Apply the same weighted decision-matrix approach from Section 2.4.4, now with criteria suited to choosing a reporting tool. The criteria are ecosystem integration with the company's existing SQL Server environment (weight 0.4, reflecting the SSIS-based infrastructure already in place from Unit 3), cost-effectiveness (weight 0.3) and visualization flexibility (weight 0.3). For the telecom churn project, this scoring exercise gives Power BI a total of 8.1 and Tableau a total of 6.5. The main reason is Power BI's better fit with the company's existing Microsoft-based infrastructure, as shown in the right-hand panel of Figure 2.4.2. As with the Python versus R comparison, a different organization with a more varied, non-Microsoft technology environment and a stronger emphasis on exploratory visual analysis might reasonably favour Tableau instead.

2.4.8 Cloud Analytics Platforms: Microsoft Azure Analytics

Rather than running analytics infrastructure entirely on an organization's own physical servers, many organizations now rely on cloud analytics platforms, which provide storage, processing power, and analytical services on demand, hosted and maintained by a cloud provider and typically billed according to actual usage. Microsoft Azure offers a broad suite of analytics services, several of the most relevant to this course including:

SGOU

- **Azure Synapse Analytics:** A large-scale, cloud-based data warehousing and analytics service, extending the star-schema and OLAP concepts introduced in Unit 1 to a fully managed, elastically scalable cloud environment.

- **Azure Data Factory:** A cloud-based ETL and data integration service, serving a similar role to the SSIS packages introduced in Unit 3, but designed specifically to orchestrate data movement at cloud scale, including from and to Hadoop-based big data platforms.

- **Azure Machine Learning:** A managed platform for building, training, and deploying machine learning models, including many of the same techniques accessible through Python's scikit-learn library, but with additional tooling for managing the full model life cycle at production scale.

- **Azure Databricks:** A cloud-based platform, built on Apache Spark (introduced in Unit 1), for large-scale data engineering and machine learning workloads requiring the combined processing power of many machines.

For an organization already using Microsoft SQL Server, SSIS, and Power BI, Azure Analytics offers a natural path to scale those same familiar tools and concepts into the cloud, without needing to learn an entirely different vendor's platform from scratch.

2.4.9 Cloud Analytics Platforms: AWS Analytics

Amazon Web Services (AWS), one of the earliest and largest cloud computing providers, offers its own comprehensive suite of analytics services, broadly comparable in purpose to Azure's offerings but built around Amazon's own tools and terminology:

- **Amazon Redshift:** A large-scale, cloud-based data warehousing service, serving the same fundamental purpose as Azure Synapse Analytics -- fast, scalable OLAP-style querying and reporting over large volumes of structured data.

- **AWS Glue:** A cloud-based ETL service for discovering, preparing, and combining data for analytics, serving a role comparable to Azure Data Factory or the Talend and SSIS tools introduced in Unit 3.

- **Amazon SageMaker:** A fully managed platform for building, training, and deploying machine learning models at scale, comparable in purpose to Azure Machine Learning.

- **Amazon QuickSight:** A cloud-based business intelligence and visualization service, allowing dashboards and reports to be built and shared directly within the AWS ecosystem, comparable in purpose to Power BI or Tableau.

SGOU

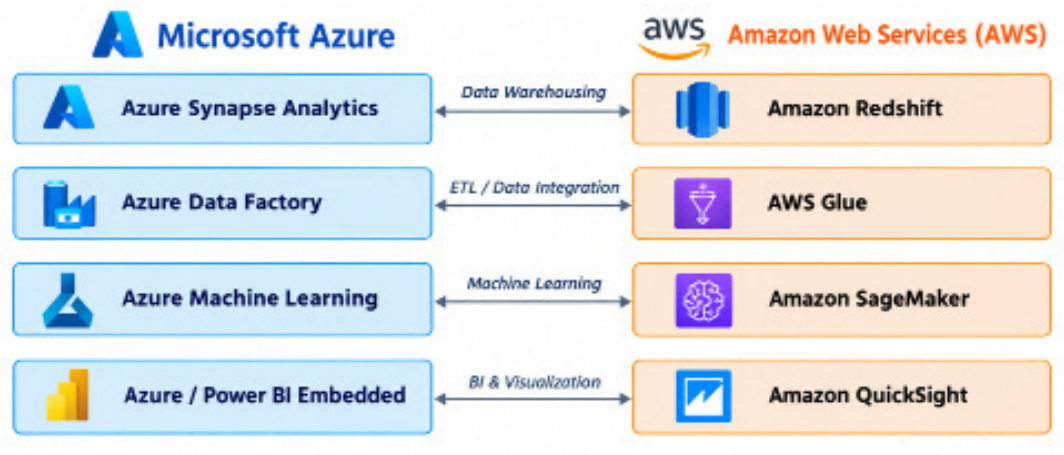


Fig. 2.4.3 Roughly Equivalent Analytics Services on Azure and AWS

As Figure 2.4.3 illustrates, Azure and AWS offer largely parallel sets of analytics services, addressing the same underlying needs -- data warehousing, ETL, machine learning, and visualization -- introduced throughout this block, but packaged under each provider's own branding and integrated most tightly within that provider's own broader cloud ecosystem. An organization's choice between Azure, AWS, or another provider entirely is typically driven by its existing infrastructure investments, team expertise, and vendor relationships, rather than by one platform being fundamentally more capable than the other for the kinds of analytics tasks introduced in this course.

2.4.10 Choosing the Right Tool: A Decision Framework

Drawing together every tool introduced in this unit, three practical questions consistently help narrow down an appropriate choice for a given analytics task:

- What is the nature of the task? Custom, code-based modelling and automation generally favour Python or R; business-facing dashboards and self-service reporting generally favour Power BI or Tableau; large-scale, organization-wide infrastructure generally favours a cloud analytics platform such as Azure or AWS.
- What skills does the team already have? A team comfortable with programming can take fuller advantage of Python or R's flexibility; a team without programming experience may be far more productive, far more quickly, using a drag-and-drop BI tool.
- What infrastructure does the organization already use? An organization already standardized on Microsoft SQL Server and SSIS, as seen throughout this block's case study, is likely to find Power BI and Azure Analytics a more natural extension of its existing investment than introducing entirely new platforms from a different vendor.

These three questions are, in effect, an informal version of the same weighted decision-matrix approach introduced formally in Section 2.4.4 -- naming the criteria that matter most for a specific situation, and using them to guide, rather than dictate, a final tool selection.

SGOU

2.4.11 Conclusion: Resolving the Case Study and Applications in Practice

Returning one final time to the intern and her telecom churn project: applying the decision framework from Section 2.4.10 to a company already standardized on Microsoft SQL Server and SSIS (Unit 3), with a small team new to formal modelling but needing to automate a monthly retraining process, a reasonable resolution emerges. Python, scoring highest in the weighted decision matrix in Section 2.4.4, is selected for Model Building, taking advantage of its strong scikit-learn ecosystem for the classification techniques shortlisted during Model Planning in Unit 2. Power BI, scoring highest in Section 2.4.7's comparison and integrating naturally with the company's existing SQL Server environment, is selected to build the final interactive dashboard presented to the retention team -- closing the loop from the very first Discovery conversation in Unit 2, through data preparation, modelling, and finally to a business-facing result.

This same pattern of tool selection recurs constantly across real-world data analytics practice:

- **Retail and E-Commerce:** Python or R is commonly used to build demand-forecasting and recommendation models, with results delivered to store managers through Power BI or Tableau dashboards.
- **Financial Services:** R remains widely used for rigorous statistical risk modelling, often deployed within a bank's own infrastructure or a cloud analytics platform for regulatory reporting at scale.
- **Healthcare:** Cloud analytics platforms such as Azure or AWS are increasingly used to securely store and process large volumes of patient data, with BI tools used to present de-identified, aggregated insights to hospital administrators.
- **Technology Start-ups:** Smaller organizations without existing infrastructure investments often choose based purely on team skill and cost, frequently favouring Python's open-source ecosystem and free or low-cost BI tools during their early growth.

You have now surveyed the data analysis tools and platforms and tried a clear, criteria-based way of choosing between them. This completes Block II. In this block you moved from where data is stored (Unit 1), through how a project is structured (Unit 2) and how varied data is prepared (Unit 3), to how that data is finally analyzed and presented. Block III builds directly on this foundation, introducing the specific classification and prediction techniques -- including decision trees, the Naive Bayes classifier, and regression-based forecasting -- shortlisted but not yet built in this block's running case study.

SGOU

Recap

- Data Analysis Tools: Data analysis tools are grouped into three categories: programming languages, Business Intelligence (BI) tools, and cloud analytics platforms.
- Python: Python is an open-source programming language used for data analysis, automation, and machine learning. Libraries such as pandas, NumPy, scikit-learn, and Matplotlib support data processing, modelling, and visualization.
- R: R is a programming language designed for statistical computing and graphics. It is widely used in statistical analysis, research, and data visualization.
- Python and R: Python is well suited for automation and machine learning deployment, while R is particularly useful for statistical modelling and research.
- Power BI: Power BI is a Microsoft BI tool used to create interactive reports and dashboards. It supports data preparation through Power Query and calculations through DAX.
- Tableau: Tableau is a data visualization tool used to create interactive dashboards. It supports a wide range of data sources and offers flexibility in exploring and presenting data.
- Power BI and Tableau: Power BI is particularly suitable for organizations using Microsoft technologies, while Tableau is useful for organizations that need flexible visualization across different platforms.
- Clod Analytics Platforms: Cloud analytics platforms provide storage, data processing, machine learning, and visualization services over the internet. They can scale according to an organization's needs.
- Microsoft Azure Analytics: Important services include Azure Synapse Analytics for data warehousing, Azure Data Factory for data integration, Azure Machine Learning for model development, and Azure Databricks for large-scale data processing.
- AWS Analytics: Important services include Amazon Redshift for data warehousing, AWS Glue for data integration, Amazon SageMaker for machine learning, and Amazon QuickSight for visualization.
- Weighted Decision Matrix: A weighted decision matrix helps compare tools using selected criteria. Each criterion is assigned a weight, and each tool is scored to calculate a total weighted score.
- Choosing the Right Tool: Tool selection depends on the project requirements, team skills, cost, existing infrastructure, and the type of analysis required.

SGOU

- Applications of Data Analysis Tools: Python and R support analytical modelling, Power BI and Tableau present results through dashboards, and cloud platforms support large-scale data storage and processing.
- Integrated Tool Selection: A single project may use different tools for different purposes. For example, Python can build a machine learning model, Power BI can present the results, and a cloud platform can provide storage and processing facilities.

Objective Type Questions

1. What are the three broad categories of data analysis tools and platforms introduced in this unit?
2. Which Python library provides fast, flexible data structures for cleaning and analyzing tabular data?
3. Which Python library provides a consistent interface to a wide range of machine learning techniques?
4. What is CRAN?
5. Which R visualization package is based on a 'grammar of graphics'?
6. What is a weighted decision matrix used for?
7. Which company develops Power BI?
8. What does DAX stand for in Power BI?
9. Which underlying engine translates Tableau's drag-and-drop actions into visualizations?
10. What is Azure Synapse Analytics roughly equivalent to on AWS?
11. What is AWS Glue used for?
12. Which AWS service is roughly equivalent to Azure Machine Learning?
13. Which Azure service provides cloud-based data warehousing and analytics?
14. Which data processing framework underlies Azure Databricks?
15. Which AWS service is comparable to Azure Data Factory?

SGOU

Answers

1. Programming languages, business intelligence (BI) and visualization tools, and cloud analytics platforms
2. pandas
3. scikit-learn
4. The Comprehensive R Archive Network, a repository of R packages for statistical methods
5. ggplot2
6. To transparently compare tools against weighted criteria and compute an overall score to guide tool selection
7. Microsoft
8. Data Analysis Expressions
9. VizQL
10. Amazon Redshift
11. ETL and data integration (discovering, preparing, and combining data for analytics)
12. Amazon SageMaker
13. Synapse
14. Spark
15. Glue

SGOU

Assignments

1. Construct your own weighted decision matrix, with at least three criteria and explicit weights, to choose between Python and R for a data analysis project of your own choosing (different from the telecom churn case study). Show your scores and calculate the weighted totals.
2. Explain, in your own words, the difference between a programming language such as Python and a BI tool such as Power BI, in terms of the flexibility they offer and the skills they require.
3. A university wants to build a dashboard showing student enrolment trends, using data already stored in a Microsoft SQL Server database. Using the decision framework introduced in this unit, recommend whether Power BI or Tableau would likely be the more natural choice, and justify your answer.

4. Identify one Azure Analytics service and its roughly equivalent AWS Analytics service (different from the pairs already discussed in this unit, if possible, or explained in your own words), and describe the purpose they both serve.

5. Describe a realistic analytics project of your own choosing that would benefit from using more than one tool together (for example, a programming language and a BI tool, or an ETL tool and a cloud platform), and explain the role each tool would play

Suggested Reading

1. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

2. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services / Wiley.

3. McKinney, W. (2022). Python for Data Analysis (3rd ed.). O'Reilly Media.

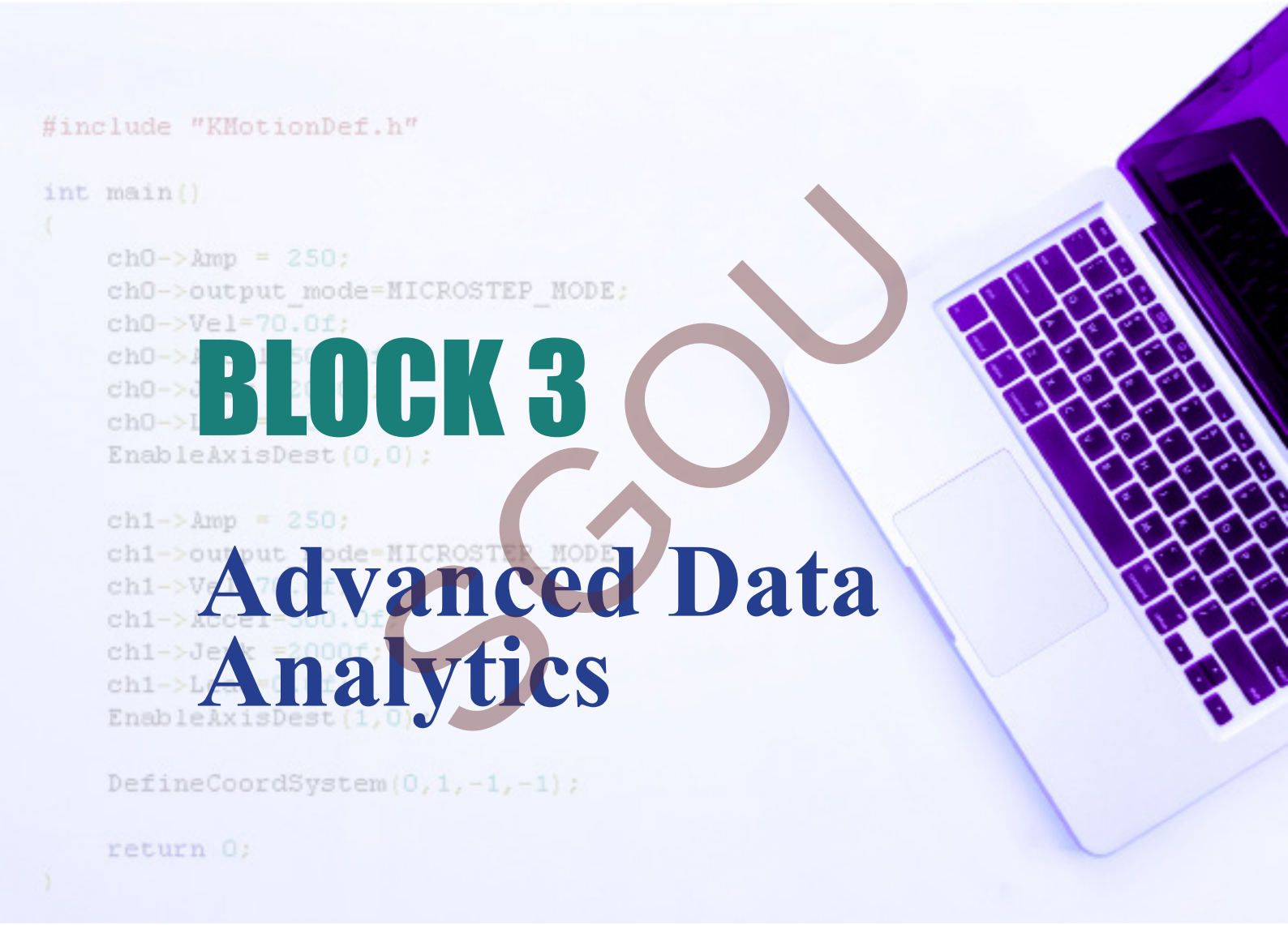
Reference

1. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

2. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services / Wiley.

3. McKinney, W. (2022). Python for Data Analysis (3rd ed.). O'Reilly Media.

SGOU



BLOCK 3

Advanced Data Analytics

SGOU



Basics of Machine Learning Algorithms

Learning Outcomes

After completion of this unit, learner will be able to:

- distinguish supervised learning from unsupervised learning
- explain the goal of clustering and apply the K-Means algorithm to group a small dataset
- explain the goal of regression and fit a simple linear regression line to a small dataset
- explain how a decision tree splits data, and calculate Gini impurity to evaluate a candidate split
- explain the Naive Bayes classifier's independence assumption and apply it to classify a new observation using multiple features
- compare clustering, regression, and classification, and identify which is appropriate for a given analytics problem

Prerequisite

Picture the intern, now with a laptop open and a blank Python notebook in front of her, staring at the two techniques -- decision trees and the Naive Bayes classifier -- that she confidently wrote into her Model Planning notes back in Block II, Unit 2. She quickly realizes that naming a technique and actually understanding what it computes are two very different skills, and that she cannot write a single line of working model-building code until she does.

To follow her into this unit, you should be comfortable computing a mean and a standard deviation (Block I, Unit 1). You should also recall how Bayes' theorem combines a prior probability with new evidence to give an updated probability (Block I, Unit 4). The Naive Bayes classifier introduced later in this unit is a direct, practical extension of that idea. No prior exposure to machine learning, programming, or any specific software tool is assumed.

SGOU

Keywords

Machine Learning, Supervised Learning, Unsupervised Learning, Clustering, K-Means, Regression, Linear Regression, Classification, Decision Tree, Gini Impurity, Naive Bayes Classifier.

Discussion

3.1.1 Introduction to Machine Learning

Machine learning is a branch of data analytics in which a computer program learns patterns directly from data, rather than being explicitly programmed with a fixed set of rules for every situation it might encounter. Instead of a human analyst manually writing a rule such as 'flag a customer as likely to churn if their monthly bill is above ₹2,000 and they have called support more than three times', a machine learning algorithm studies historical examples of customers who did and did not churn. It then works out automatically which patterns in the data are linked to churning.

Every technique in this unit builds on ideas from earlier in this course. Clustering and classification are systematic ways of organizing and labelling data, extending the descriptive statistics of Block I. Regression extends the thinking about relationships between variables that was first touched on in Block I. The Naive Bayes classifier is a practical application of Bayes' theorem, introduced in Block I, Unit 4. Machine learning, in this sense, is not an entirely separate subject, but rather the point where the statistical and probabilistic foundations built across this course are assembled into automated, predictive tools.

It is worth being clear about what machine learning does and does not do, since the term is often used loosely. A machine learning algorithm does not 'understand' a business problem in any human sense; it simply identifies statistical regularities in the historical data it is given, and assumes, often reasonably but not always safely, that those same regularities will continue to hold for new, future data. This is why every phase of the data analytics life cycle from Block II, Unit 2 matters so much, especially Discovery and Data Preparation. A machine learning algorithm applied to a poorly understood business problem, or to poorly prepared data, will confidently learn and repeat whatever patterns exist in that data, including any errors or biases.

3.1.2 Supervised versus Unsupervised Learning

Machine learning algorithms are broadly divided into two categories, based on whether the historical data used to train them includes a known, labelled outcome.

Supervised Learning : The training data includes a known outcome (label) for every example, and the algorithm learns to predict that outcome for new, unseen examples. Supervised learning is further divided into:

SGOU

- **Regression :** Used when the outcome to be predicted is a continuous number. For example, predicting a customer's expected monthly bill amount.
- **Classification :** Used when the outcome to be predicted is a category. For example, predicting whether a customer will churn (Yes/No), using techniques such as decision trees and the Naive Bayes classifier, both introduced later in this unit.

Unsupervised Learning : The training data has no known outcome or label at all; the algorithm's job is instead to find some useful underlying structure or grouping within the data on its own. Clustering, introduced in Section 3.1.3, is the most common unsupervised technique and is used, for example, to automatically group customers into segments based on their behaviour, without any predefined notion of what those segments should be.

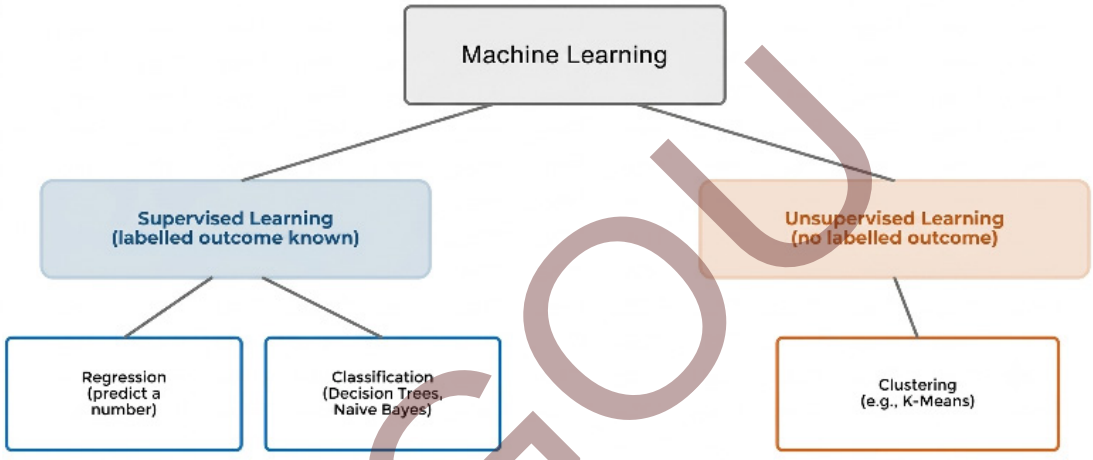


Fig 3.1.1 A Basic Taxonomy of Machine Learning Algorithms

As shown in Figure 3.1.1, this unit introduces one representative technique from each of these three branches -- K-Means for clustering, linear regression for regression, and decision trees together with the Naive Bayes classifier for classification -- providing a foundational toolkit spanning both supervised and unsupervised learning.

3.1.3 Clustering: The K-Means Algorithm

Clustering is an unsupervised technique that groups similar observations together into clusters, based purely on how close they are to one another, without using any predefined labels. K-Means is the most widely used clustering algorithm, and works through the following steps:

Before working through the mechanics, it is worth being clear about what problem clustering actually solves. Unlike every other technique introduced in this unit, clustering is never 'told' the right answer during training -- there is no historical column labelled 'segment' for it to learn from. Instead, it is given only raw, unlabelled measurements and asked to discover, purely from the geometry of the data itself, which observations naturally belong together. This makes clustering particularly valuable during the earlier, more exploratory stages of an analytics project -- for instance, well before a business

SGOU

even knows that 'high-complaint, low-spend customers' is a meaningful group worth studying, clustering can reveal that such a group exists in the data at all.

- Step 1 Choose k: Decide in advance how many clusters, k, to divide the data into.
- Step 2 Initialize Centroids: Select k initial points (often simply k of the actual data points) to act as the starting centre, or centroid, of each cluster.
- Step 3 Assign Points: Assign every data point to its nearest centroid, typically measured using straight-line (Euclidean) distance.
- Step 4 Update Centroids: Recalculate each centroid as the mean position of all the points currently assigned to it.
- Step 5 Repeat: Repeat Steps 3 and 4 until the cluster assignments no longer change (the algorithm has converged).

Worked Example: Consider four customers described by two features: monthly spend (in ₹'00) and number of support complaints filed -- P1 (2, 10), P2 (2, 8), P3 (8, 2), and P4 (8, 4). Suppose k = 2, with initial centroids chosen as C1 = P1 = (2, 10) and C2 = P3 = (8, 2).

Calculating the Euclidean distance, $\sqrt{[(x1-x2)^2 + (y1-y2)^2]}$, from every point to each centroid:

P1 to C1 = 0.00, to C2 = 10.00 → assigned to Cluster 1

P2 to C1 = 2.00, to C2 = 8.49 → assigned to Cluster 1

P3 to C1 = 10.00, to C2 = 0.00 → assigned to Cluster 2

P4 to C1 = 8.49, to C2 = 2.00 → assigned to Cluster 2

Updating each centroid as the mean of its assigned points:

New C1 = mean of P1(2,10), P2(2,8) = (2, 9)

New C2 = mean of P3(8,2), P4(8,4) = (8, 3)

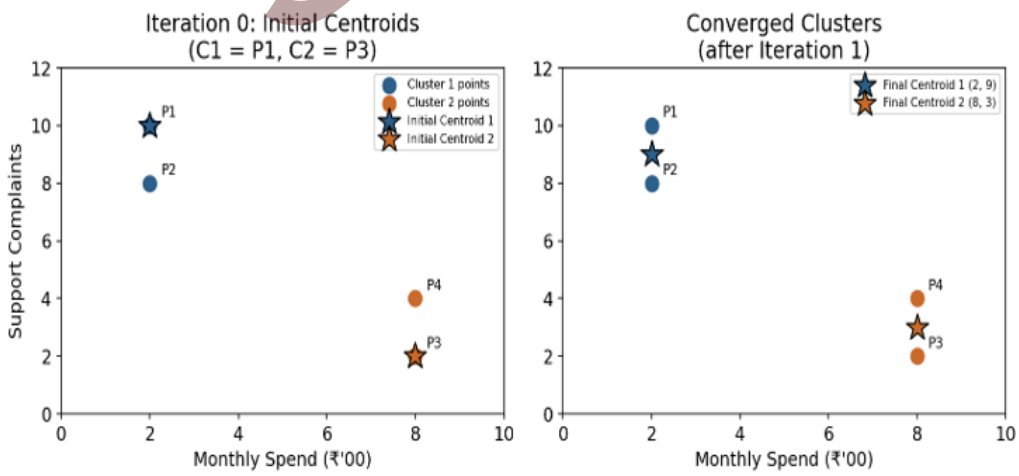


Fig 3.1.2 K-Means Clustering Worked Example (k = 2)

SGOU

Recomputing the distances from every point to these new centroids shows that no point changes its assigned cluster, so the algorithm has converged after just one iteration, with final centroids at (2, 9) and (8, 3), as shown in Figure 3.1.2. Interpreting the result in business terms, Cluster 1 (low spend, many complaints) might represent customers at high risk of churning, while Cluster 2 (higher spend, few complaints) might represent a company's more loyal, satisfied customer base -- exactly the kind of customer segmentation that clustering is commonly used to produce.

3.1.4 Regression: Simple Linear Regression

Regression is a supervised learning technique used to predict a continuous numeric outcome based on one or more input variables, by fitting a mathematical equation that best describes the relationship between them. The simplest form, simple linear regression, fits a straight line through the data of the form:

$$Y = b0 + b1X$$

where Y is the outcome being predicted, X is the input variable, b1 is the slope (how much Y changes for each one-unit increase in X), and b0 is the intercept (the predicted value of Y when X is zero). The values of b0 and b1 that best fit a given dataset -- in the sense of minimizing the total squared vertical distance between the line and the actual data points -- are calculated using the following formulas:

$$b1 = \Sigma(x - \bar{x})(y - \bar{y}) / \Sigma(x - \bar{x})^2 \qquad b0 = \bar{y} - b1 \cdot \bar{x}$$

Worked Example: A regional manager records monthly advertising spend (X, in ₹'000) and units sold (Y, in '00) over 5 months: X = 1, 2, 3, 4, 5 and Y = 3, 4, 4, 5, 6. Fit a simple linear regression line.

Table 3.1.1 Computing the Components of Simple Linear Regression ($\bar{x} = 3$, $\bar{y} = 4.4$)

X	Y	$x - \bar{x}$	$y - \bar{y}$	$(x - \bar{x})(y - \bar{y})$	$(x - \bar{x})^2$
1	3	-2	-1.4	2.8	4
2	4	-1	-0.4	0.4	1
3	4	0	-0.4	0	0
4	5	1	0.6	0.6	1
5	6	2	1.6	3.2	4

$$b1 = (2.8+0.4+0+0.6+3.2) / (4+1+0+1+4) = 7.0 / 10 = 0.7$$

$$b0 = 4.4 - (0.7 \times 3) = 4.4 - 2.1 = 2.3$$

Fitted Regression Equation: $Y = 2.3 + 0.7X$

SGOU

Using this equation, the manager can now predict units sold for a new level of advertising spend -- for example, at X = 6 (₹6,000): Y = 2.3 + (0.7 × 6) = 2.3 + 4.2 = 6.5, predicting sales of approximately 650 units. In the telecom churn project, an identical technique could be used, for instance, to predict a customer's expected monthly bill from their historical data usage, or to predict expected customer lifetime value from tenure -- any situation calling for a continuous numeric prediction rather than a category.

3.1.5 Classification: Decision Trees

A decision tree is a classification technique that predicts a categorical outcome by asking a sequence of simple, automatically learned yes/no (or this-category/that-category) questions about the input variables, arranged in a tree-like structure of nodes and branches, until it arrives at a final predicted category at a leaf node.

The central challenge in building a decision tree is deciding which variable to split on at each node, and where. This is typically done by choosing the split that produces the greatest reduction in impurity -- a measure of how mixed together the different outcome categories are within a node. One widely used impurity measure is the Gini impurity, calculated for a node as:

$$\text{Gini} = 1 - \Sigma(p_i^2)$$

where pi is the proportion of observations in the node belonging to category i. A Gini value of 0 indicates a perfectly 'pure' node (every observation belongs to the same category), while higher values indicate a more evenly mixed node.

Worked Example: Among 10 telecom customers, 4 churned and 6 stayed. The Gini impurity of this entire (root) group is:

$$\text{Gini(root)} = 1 - [(4/10)^2 + (6/10)^2] = 1 - (0.16 + 0.36) = 1 - 0.52 = 0.48$$

Suppose the tree considers splitting on Contract Type, dividing the 10 customers into 6 on month-to-month contracts (4 churned, 2 stayed) and 4 on long-term contracts (0 churned, 4 stayed):

$$\text{Gini(Month-to-Month)} = 1 - [(4/6)^2 + (2/6)^2] = 1 - (0.444 + 0.111) = 0.444$$

$$\text{Gini(Long-Term)} = 1 - [(0/4)^2 + (4/4)^2] = 1 - (0 + 1) = 0.0 \quad \text{(a perfectly pure node)}$$

The overall impurity after this split is the weighted average of the two child nodes' Gini values, weighted by how many customers fall into each:

$$\text{Weighted Gini} = (6/10 \times 0.444) + (4/10 \times 0.0) = 0.267$$

$$\text{Gini Reduction (Gini Gain)} = \text{Gini(root)} - \text{Weighted Gini} = 0.48 - 0.267 = 0.213$$

As shown in Figure 3.1.3, splitting on Contract Type produces a substantial reduction in impurity (0.213), and, notably, produces a perfectly pure long-term-contract node in which every customer stayed -- meaning that, on this small example, contract type alone almost perfectly separates churners from non-churners. A real decision tree algorithm evaluates many candidate variables and split points in this same way at every node, always choosing the split that most reduces impurity, and repeats the process on each

SGOU

resulting child node until a stopping condition (such as a maximum tree depth, or nodes that are already sufficiently pure) is reached.

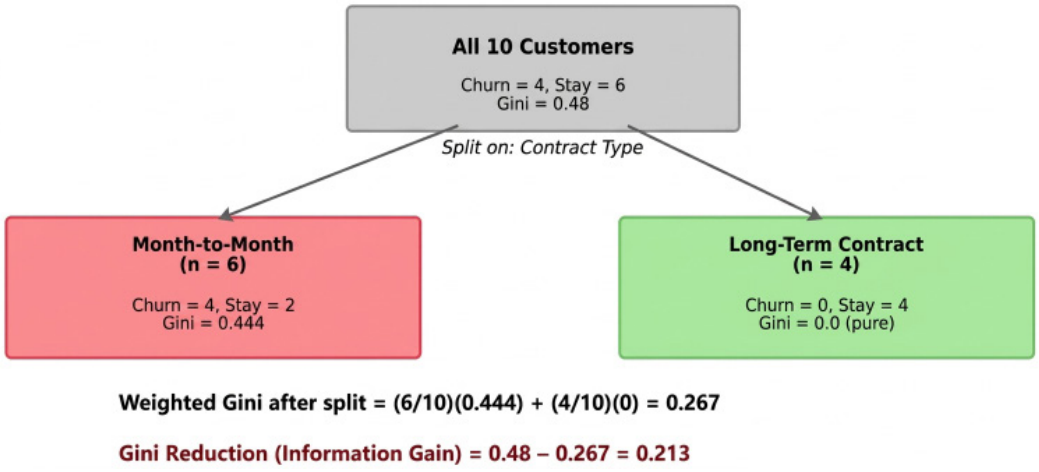


Fig. 3.1.3 A Single Decision Tree Split for the Telecom Churn Example

Note that a decision tree grown with no stopping condition will keep splitting until every leaf node is perfectly pure. This usually means the tree has memorized the training data, including its noise and quirks, instead of learning a pattern general enough for new customers. This is an example of the overfitting problem first introduced in Block II, Unit 2. For this reason, real-world decision tree implementations almost always impose some combination of a maximum depth, a minimum number of observations required in a node before it can be split further, or a separate validation step, to keep the tree general enough to be genuinely useful on new, unseen data.

3.1.6 Classification: The Naive Bayes Classifier

The Naive Bayes classifier is a classification technique built directly on Bayes' theorem, introduced in Block I, Unit 4. Recall that Bayes' theorem allows a conditional probability to be reversed: P(A|B) = [P(B|A) × P(A)] / P(B). The Naive Bayes classifier extends this idea to classification problems involving multiple input features simultaneously, by making a simplifying assumption -- the 'naive' assumption -- that all input features are conditionally independent of one another, given the outcome category. This assumption is rarely exactly true in real data, but it makes the underlying calculation dramatically simpler, and the classifier frequently performs well in practice even when the assumption does not hold perfectly.

Under this assumption, for two features and two possible outcome categories, the Naive Bayes classifier compares:

Score(Class) ∝ P(Class) × P(Feature1 | Class) × P(Feature2 | Class) and predicts whichever class produces the higher score.

Worked Example : Among 10 training customers, 4 churned and 6 did not. Among the 4 who churned, 3 had made many support calls and 3 were not on a long-term contract. Among the 6 who did not churn, only 1 had made many support calls and only

SGOU

1 was not on a long-term contract. A new customer has made many support calls and is not on a long-term contract. Should this customer be classified as likely to churn?

Table 3.1.2 Prior Probabilities and Likelihoods for the Naive Bayes Worked Example

Quantity	Value	Calculation
P(Churn)	0.4	4 churned out of 10 total
P(No Churn)	0.6	6 not churned out of 10 total
P(Many Calls | Churn)	0.75	3 of 4 churned customers
P(No Long-Term Contract | Churn)	0.75	3 of 4 churned customers
P(Many Calls | No Churn)	0.167	1 of 6 non-churned customers
P(No Long-Term Contract | No Churn)	0.167	1 of 6 non-churned customers

$$\text{Score(Churn)} = 0.4 \times 0.75 \times 0.75 = 0.225$$

$$\text{Score(No Churn)} = 0.6 \times 0.167 \times 0.167 \approx 0.0167$$

Since Score(Churn) is considerably larger than Score(No Churn), the new customer is classified as likely to churn. Converting these scores into an actual probability by normalizing them so that they sum to 1:

$$P(\text{Churn} \mid \text{Evidence}) = 0.225 / (0.225 + 0.0167) \approx 0.931 \quad (93.1\%)$$

The model estimates a 93.1% probability that this new customer will churn. It combines the evidence from both features in the same way as the medical-test and spam-filtering examples in Block I, Unit 4, but now with two pieces of independent evidence instead of one. This shows how the Naive Bayes classifier scales Bayes' theorem up to realistic problems with many features.

It is worth revisiting, briefly, why the word 'naive' is used at all. In reality, 'making many support calls' and 'not being on a long-term contract' might well be related to each other -- perhaps customers on short, flexible contracts feel more comfortable calling support frequently, precisely because they know they can leave easily if unsatisfied. The Naive Bayes classifier simply ignores any such relationship between features and treats them as independent, which is a simplification of reality rather than a claim that the features truly are unrelated. Despite this simplification, the technique has remained popular for decades precisely because it is fast to compute, requires relatively little training data to produce reasonable estimates, and, in practice, frequently classifies new observations correctly even when its underlying independence assumption is only approximately true.

SGOU

3.1.7 Comparing Clustering, Regression, and Classification

Table 3.1.3 summarizes the four techniques introduced in this unit, to help the learner quickly identify which is appropriate for a given analytics problem.

Table 3.1.3 Comparing the four machine learning techniques introduced in this unit

Technique	Learning Type	Typical Question Answered	Output Type
K-Means Clustering	Unsupervised	'What natural groups exist in this data?'	Cluster assignment (no predefined labels)
Linear Regression	Supervised	'How much will this numeric value be?'	A continuous number
Decision Tree	Supervised (Classification)	'Which category does this belong to?' (interpretable, rule-based)	A category
Naive Bayes Classifier	Supervised (Classification)	'Which category does this belong to?' (probability-based)	A category, with a probability

A practical first question for any new analytics problem is whether the historical data has a known outcome (label) at all. If not, clustering is the natural starting point. If there is a known outcome and it is a number, regression applies. If there is a known outcome and it is a category, a classification technique such as a decision tree or the Naive Bayes classifier applies. Between the two classification techniques, decision trees offer an easily interpretable, rule-based structure that a business user can follow step by step, while the Naive Bayes classifier offers a direct probability estimate and tends to perform particularly well on text and other high-dimensional data, such as the bag-of-words representations introduced in Block II, Unit 3.

It is also worth noting that these four techniques are not always used in isolation from one another within a single project. For example, an analyst might first use K-Means clustering to split customers into broad behavioural groups with no predefined labels. Suppose one segment shows an unusually high churn rate. The analyst can then build a decision tree or Naive Bayes classifier just for that segment. This explains and predicts churn within the segment more precisely than a single model built on the whole customer base. Recognizing when to combine techniques, rather than treating each as a completely separate tool, is itself a valuable and frequently underappreciated analytical skill.

SGOU

3.1.8 Applications of Machine Learning in Data Analytics

The four techniques introduced in this unit are among the most widely applied in real-world data analytics:

- Customer Segmentation: K-Means clustering is used extensively to group customers by behaviour or spending patterns, informing targeted marketing campaigns.
- Sales and Demand Forecasting: Linear regression (and its extensions) is used to predict sales, demand, or resource requirements from historical and related data.
- Credit Scoring and Medical Diagnosis: Decision trees are widely valued in these fields specifically because their step-by-step, rule-based structure can be explained clearly to a loan applicant or a patient, unlike less transparent techniques.
- Spam Filtering and Document Classification: The Naive Bayes classifier remains a fast, effective, and widely used technique for classifying text documents, building directly on the bag-of-words representations introduced in Block II, Unit 3.

It is worth reflecting, briefly, on just how much of this course's earlier material has fed into this single unit. Several ideas from earlier blocks support this unit. The mean and standard deviation from Block I, Unit 1 underlie how a K-Means centroid is calculated and how a regression line minimizes squared error. The probability and Bayes' theorem ideas from Block I, Unit 4 underlie the Naive Bayes classifier directly. The analytic sandbox and model-planning vocabulary from Block II, Unit 2 describe the stage of a project at which these techniques are applied. The bag-of-words text representation from Block II, Unit 3 gives the kind of numeric input on which a Naive Bayes classifier or decision tree can be trained. Machine learning, in this sense, is best understood not as a new, separate subject to be learned from scratch, but as the point at which nearly everything covered so far in this course is finally put to direct, predictive use.

You have now learned the basic mechanics of clustering, regression, decision trees and the Naive Bayes classifier. The intern can now build a working version of the churn model she first planned in Block II. You are also ready for Unit 2 of this block, which introduces the statistical measures and error metrics, including the correlation coefficient and root mean square error, used to evaluate how well such models perform.

Recap

- Machine learning algorithms learn patterns directly from historical data, and are broadly divided into supervised learning (a known outcome is available) and unsupervised learning (no known outcome is available).
- K-Means clustering is an unsupervised technique that groups data into k clusters by repeatedly assigning points to their nearest centroid and recalculating centroids until convergence.

SGOU

- Simple linear regression fits a straight line, Y = b0 + b1X, to predict a continuous numeric outcome, with b0 and b1 calculated to minimize the total squared distance between the line and the actual data.
- A decision tree classifies observations by repeatedly splitting data on the variable and value that most reduces impurity, commonly measured using Gini impurity.
- The Naive Bayes classifier extends Bayes' theorem to multiple features by assuming those features are conditionally independent given the outcome class, multiplying together the prior probability and each feature's likelihood to score each candidate class.
- Clustering answers 'what natural groups exist?', regression answers 'how much?', and classification (decision trees, Naive Bayes) answers 'which category?' -- providing a practical starting point for selecting the right technique for a given problem.
- Decision trees offer an easily interpretable, rule-based structure, while the Naive Bayes classifier offers a direct probability estimate and performs particularly well on text and other high-dimensional data.

Objective Type Questions

1. What is the main difference between supervised and unsupervised learning?
2. Which of the four techniques introduced in this unit is unsupervised?
3. What does the letter k represent in the K-Means algorithm?
4. How is a new centroid calculated in the K-Means algorithm?
5. What type of outcome does regression predict?
6. In the simple linear regression equation Y = b0 + b1X, what does b1 represent?
7. What does a Gini impurity value of 0 indicate about a node?
8. What is the formula for Gini impurity?
9. What is Information Gain (Gini Reduction) in a decision tree?
10. What is the 'naive' assumption made by the Naive Bayes classifier?
11. What theorem is the Naive Bayes classifier directly based on?
12. What does the Naive Bayes classifier predict when comparing scores for two classes?
13. Which classification technique is generally considered easiest for a non-technical business user to interpret?

SGOU

14. Which classification technique tends to perform particularly well on text and other high-dimensional data?

15. What practical first question helps decide whether to use clustering, regression, or classification for a new problem?

Answers

1. Supervised learning uses data with a known outcome/label; unsupervised learning uses data with no known outcome
2. K-Means clustering
3. The number of clusters to divide the data into
4. As the mean position of all data points currently assigned to that cluster
5. A continuous number
6. The slope, i.e., how much Y changes for each one-unit increase in X
7. The node is perfectly pure (all observations belong to the same category)
8. Gini = $1 - \Sigma(pi^2)$, where pi is the proportion of observations in category i
9. The reduction in Gini impurity achieved by a particular split, compared to the parent node
10. That all input features are conditionally independent of one another, given the outcome class
11. Bayes' theorem
12. It predicts whichever class produces the higher score
13. Decision trees
14. The Naive Bayes classifier
15. Whether a known outcome (label) is available in the historical data, and if so, whether it is a number (regression) or a category (classification)

Assignments

1. Given four points A(1,1), B(2,1), C(8,8), D(9,8), apply one full iteration of K-Means with k = 2, using A and C as the initial centroids. Show the distance calculations, the resulting cluster assignments, and the updated centroids.

2. A shop records advertising spend (X, in ₹'000) and daily footfall (Y, in customers) over 5 days: X = 2, 3, 4, 5, 6 and Y = 20, 25, 27, 30, 33. Fit a simple linear regression line and predict the footfall expected for an advertising spend of ₹7,000.

SGOU

3. Among 20 loan applicants, 8 defaulted and 12 repaid their loan. If splitting on 'Employment Status' produces a group of 10 employed applicants (2 defaults, 8 repayments) and 10 unemployed applicants (6 defaults, 4 repayments), calculate the Gini impurity of the root node, the Gini impurity of each child node, and the resulting Information Gain.

4. Among 12 email messages, 5 are spam and 7 are not. Among the spam messages, 4 contain the word 'free' and 3 contain the word 'winner'. Among the non-spam messages, 1 contains the word 'free' and 0 contain the word 'winner'. Using the Naive Bayes classifier, calculate the score for 'spam' and 'not spam' for a new email containing both the words 'free' and 'winner', and state the predicted classification.

5. Describe a real-world analytics problem of your own choosing, and explain, with justification, which of the four techniques introduced in this unit (K-Means, linear regression, decision tree, or Naive Bayes classifier) would be the most appropriate starting point, and why.

Suggested Reading

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.
2. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.
3. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services / Wiley.

Reference

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.
2. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.
3. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. EMC Education Services / Wiley.

SGOU



Statistical Measures and Error Metrics

Learning Outcomes

After completion of this unit, learner will be able to:

- calculate covariance between two variables as a foundation for correlation
- calculate and interpret Pearson's correlation coefficient
- interpret the strength and direction of a correlation, and recognize its limitations
- explain the difference between correlation and causation, and identify a confounding variable
- calculate the Root Mean Square Error (RMSE) of a regression model
- describe related error metrics, including Mean Absolute Error (MAE) and R-squared
- explain the different purposes served by correlation (assessing relationships before modelling) and error metrics such as RMSE (assessing prediction accuracy after modelling)

SGOU

Prerequisite

Picture the intern again, regression equation in hand, now facing her manager's two questions with no immediate answer. She already knows how to compute a mean, a variance, and a standard deviation from Block I, and she already has a fitted regression line, Y = 2.3 + 0.7X, from Unit 1 of this block. What she is missing is simply a next set of formulas -- built directly on those same familiar building blocks -- for turning 'are these related?' and 'is my model good?' into precise, calculable numbers rather than vague impressions.

No new statistical background beyond Block I, Unit 1 (descriptive statistics) and the regression example in Block III, Unit 1 is assumed in this unit.

Keywords

Covariance, Correlation Coefficient, Pearson's r, Scatter Plot, Confounding Variable, Residual, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), R-squared.

Discussion

3.2.1 Introduction to Statistical Measures and Error Metrics

Two closely related but distinct questions arise constantly throughout data analytics. The first arises before a model is even built: are two variables actually related to one another at all, and if so, how strongly? The second arises after a model has been built: how accurate are that model's predictions, and how much should they be trusted? This unit introduces the two most fundamental tools for answering each question -- the correlation coefficient, which measures the strength and direction of a linear relationship between two variables, and the Root Mean Square Error (RMSE), which measures how far a model's predictions typically deviate from the actual values.

Both tools, perhaps unsurprisingly at this stage of the course, are built directly on the same core idea introduced in Block I, Unit 1: measuring how far individual values deviate from some reference point, and then squaring, summing, and averaging those deviations to arrive at a single, informative number. Where variance measured how far a variable's own values deviate from its own mean, this unit shows how the very same style of calculation can measure how two different variables move together, and how far a model's predictions deviate from reality.

It is worth being clear, from the outset, about why these two questions must be kept separate rather than treated as a single concern. A pair of variables can be strongly correlated and still give a poor model if the wrong type of equation is fitted to them. The opposite can also happen. A carefully built model can show respectable-looking error metrics even when the relationship between its inputs and its outcome is weak, simply because of how a particular sample of data turned out. Only by checking both -- the strength of the relationship first, and the accuracy of the resulting model second -- can an analyst be reasonably confident that a model is both well-founded and well-fitted, rather than merely one or the other.

3.2.2 Covariance: A Building Block for Correlation

Covariance measures the extent to which two variables change together. If two variables tend to increase and decrease together, their covariance is positive; if one tends to increase as the other decreases, their covariance is negative; and if there is no consistent pattern between them, their covariance is close to zero. Covariance is calculated as:

$$\text{Cov}(X, Y) = \Sigma(x - \bar{x})(y - \bar{y}) / n$$

SGOU

Worked Example: Recall the advertising spend and units sold data from Block III, Unit 1's regression worked example: X = 1, 2, 3, 4, 5 (₹'000) and Y = 3, 4, 4, 5, 6 ('00 units), with $\bar{x} = 3$ and $\bar{y} = 4.4$. That earlier worked example already calculated $\Sigma(x - \bar{x})(y - \bar{y}) = 7.0$ across all 5 months.

$$\text{Cov}(X, Y) = 7.0 / 5 = 1.4$$

A positive covariance of 1.4 confirms that advertising spend and units sold tend to rise and fall together. However, covariance has an important limitation. Its value depends on the units in which X and Y are measured. From the number 1.4 alone, it is hard to judge whether the relationship is weak or strong. It is also hard to compare it with the covariance of a completely different pair of variables measured on different scales. The correlation coefficient, introduced next, solves exactly this problem.

To see why this limitation matters in practice, imagine the same advertising spend and units sold data re-measured in different units -- advertising spend in individual rupees rather than thousands of rupees, for instance. The underlying relationship between the two variables has not changed in any meaningful sense, yet the covariance calculated from the rescaled data would come out one thousand times larger, simply as an artifact of the unit change, with no accompanying change in how strongly the variables are actually related. A measure of relationship strength that shifts so dramatically based on an arbitrary choice of units is of limited practical use for comparing relationships across different variables or different datasets.

3.2.3 The Correlation Coefficient (Pearson's r)

Pearson's correlation coefficient, commonly denoted r, rescales covariance by the standard deviations of both variables, producing a value that always falls between −1 and +1, regardless of the original units of measurement. This makes r directly interpretable and comparable across entirely different pairs of variables. It is calculated as:

$$r = \Sigma(x - \bar{x})(y - \bar{y}) \; / \; \sqrt{[\, \Sigma(x - \bar{x})^2 \times \Sigma(y - \bar{y})^2 \,]}$$

Worked Example: Continuing with the advertising spend and units sold data, the earlier regression worked example in Block III, Unit 1 already computed $\Sigma(x - \bar{x})^2 = 10$. Calculating $\Sigma(y - \bar{y})^2$ for the same data:

Table 3.2.1 Computing $\Sigma(y - \bar{y})^2$ for the Advertising Spend and Units Sold Data

Y	3	4	4	5	6
$(y - \bar{y})$	-1.4	-0.4	-0.4	0.6	1.6
$(y - \bar{y})^2$	1.96	0.16	0.16	0.36	2.56

$$\Sigma(y - \bar{y})^2 = 1.96 + 0.16 + 0.16 + 0.36 + 2.56 = 5.2$$

$$r = 7.0 / \sqrt{(10 \times 5.2)} = 7.0 / \sqrt{52} = 7.0 / 7.211 \approx 0.971$$

SGOU

A correlation coefficient of approximately 0.971 indicates a very strong positive linear relationship between advertising spend and units sold -- as advertising spend increases, units sold increase in a highly consistent, near-linear fashion, which explains why the regression line fitted in Block III, Unit 1 was able to describe this data so well.

3.2.4 Interpreting Correlation: Strength, Direction, and Scatter Plots

The sign of r indicates the direction of the relationship: a positive r means the two variables tend to increase together, while a negative r means one tends to increase as the other decreases. The magnitude of r (how close it is to −1 or +1, regardless of sign) indicates the strength of the linear relationship. While exact thresholds vary somewhat by field, Table 3.2.2 provides a commonly used general guideline.

Table 3.2.2 A General Guideline for Interpreting the Strength of |r|

Approximate Range of |r|	Typical Interpretation
0.00 -- 0.29	Weak or negligible linear relationship
0.30 -- 0.69	Moderate linear relationship
0.70 -- 0.89	Strong linear relationship
0.90 -- 1.00	Very strong linear relationship

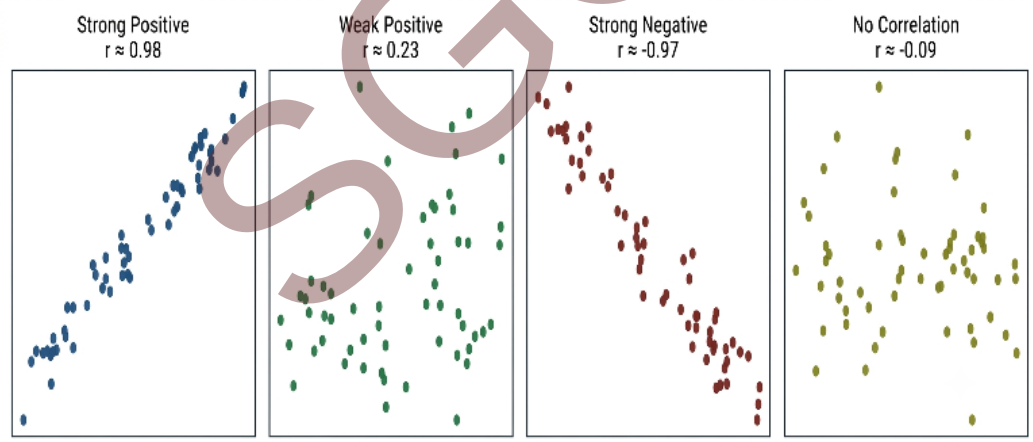


Fig. 3.2.1 Scatter Plots Illustrating Different Correlation Strengths

As Figure 3.2.1 illustrates, a scatter plot provides an immediate visual sense of correlation, well before any formula is calculated: points clustering tightly around an upward- or downward-sloping line indicate a strong relationship, while a shapeless cloud of points indicates little to no linear relationship at all.

An important limitation of the correlation coefficient must be kept in mind: r measures only the strength of a linear relationship. Two variables can have a strong, entirely consistent curved (non-linear) relationship -- for example, a U-shaped pattern -- and

SGOU

still produce a correlation coefficient close to zero, since a straight line poorly describes a curve. For this reason, an analyst should always inspect a scatter plot alongside the calculated correlation coefficient, rather than relying on the single number r in isolation, particularly when preparing to fit a model such as the linear regression introduced in Block III, Unit 1, which itself assumes a linear relationship.

A further practical caution concerns sample size. A correlation coefficient calculated from only a handful of observations, as in the five-month advertising example used throughout this unit, can be considerably less reliable than the same coefficient calculated from hundreds or thousands of observations, since a small sample is more easily influenced by chance patterns that would not persist in a larger dataset. In a real analytics project, an analyst would typically want considerably more than five data points before placing substantial business confidence in a calculated correlation coefficient; the small worked example used throughout this unit is chosen purely for ease of manual calculation, not as a realistic sample size for a genuine business decision.

3.2.5 Correlation versus Causation

Perhaps the single most important caution surrounding correlation is that a strong correlation between two variables does not, by itself, prove that one variable causes the other. Two variables can be strongly correlated because one genuinely causes the other, because the relationship runs in the opposite direction from what is assumed, or because both are independently influenced by some third variable -- known as a confounding variable -- that the analyst has not accounted for.

Classic Example : Monthly data often shows a strong positive correlation between ice cream sales and the number of drowning incidents at public beaches. It would clearly be wrong to conclude that ice cream sales cause drowning, or that drowning causes ice cream sales. Instead, both variables are independently driven by a third, confounding variable: hot weather. Hot weather leads more people to buy ice cream, and, separately, leads more people to go swimming, increasing the opportunity for drowning incidents to occur.

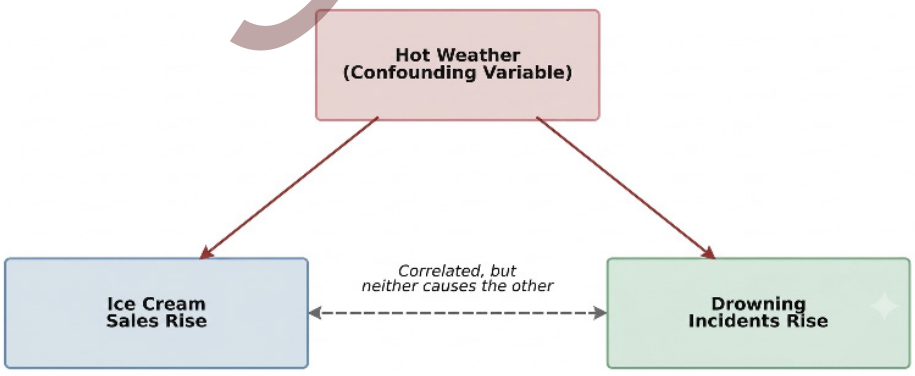


Fig. 3.2.2 Correlation Without Causation -- a Confounding Variable Example

As Figure 3.2.2 illustrates, both ice cream sales and drowning incidents are driven by hot weather, producing a strong correlation between the two even though neither directly causes the other. The same caution applies to the telecom churn project

SGOU

followed throughout this course. A strong correlation between, say, customer support call frequency and churn does not by itself prove that support calls cause churn. Some underlying issue, such as a specific billing system error, may independently cause both more support calls and more churn. Correctly distinguishing correlation from causation typically requires either carefully designed controlled experiments or more advanced causal analysis techniques that go beyond the scope of this introductory unit, but recognizing the distinction at all is an essential habit of mind for any responsible data analyst.

A useful, practical habit for guarding against this trap is to ask, whenever a strong correlation is found, whether there is a plausible third factor that could independently explain both variables, before recommending any business action based on the correlation alone. If the telecom company, for example, discovered a strong correlation between customers switching to a particular payment method and churn, it would be worth asking whether both are actually driven by something else entirely -- perhaps that payment method is disproportionately used by customers on a specific, older contract type that is itself the true driver of churn. Pausing to ask this question costs very little time, but can prevent a business from investing resources in addressing the wrong problem entirely.

3.2.6 Evaluating Regression Models: Introduction to Error Metrics

Correlation, introduced in Sections 3.2.2 to 3.2.5, helps an analyst decide, before building a model, whether two variables appear related enough to be worth modelling together. Once a regression model such as the one fitted in Block III, Unit 1 has actually been built, a different question arises: exactly how accurate are its predictions? Answering this requires examining the residuals -- the difference between each actual observed value and the value the model predicted for it:

$$\text{Residual} = \text{Actual Value } (Y) - \text{Predicted Value } (\hat{Y})$$

A model with small residuals, consistently close to zero, is making accurate predictions; a model with large residuals is not. Since residuals can be positive (the model under-predicted) or negative (the model over-predicted), simply averaging them directly would allow positive and negative errors to cancel out, hiding the true extent of a model's inaccuracy -- exactly the same issue encountered with deviations from the mean when variance was first introduced in Block I, Unit 1. The solution, once again, is to square the residuals before averaging them, leading directly to the Root Mean Square Error introduced in the next section.

3.2.7 Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) summarizes the typical size of a regression model's prediction errors in a single number, expressed in the same units as the outcome variable being predicted. It is calculated by squaring every residual, averaging the squared residuals, and then taking the square root to return to the original units:

SGOU

$$RMSE = \sqrt{[\ \Sigma(Actual - Predicted)^2 / n\ ]}$$

Worked Example: Recall the fitted regression equation from Block III, Unit 1, $\hat{Y} = 2.3 + 0.7X$, applied to the advertising spend and units sold data (X = 1, 2, 3, 4, 5; actual Y = 3, 4, 4, 5, 6).

Table 3.2.3 Computing Residuals and Squared Residuals for the Regression Worked Example

X	Actual (Y)	Predicted ($\hat{Y}$)	Residual ($Y-\hat{Y}$)	Squared Residual
1	3	3.0	0.0	0.00
2	4	3.7	0.3	0.09
3	4	4.4	-0.4	0.16
4	5	5.1	-0.1	0.01
5	6	5.8	0.2	0.04

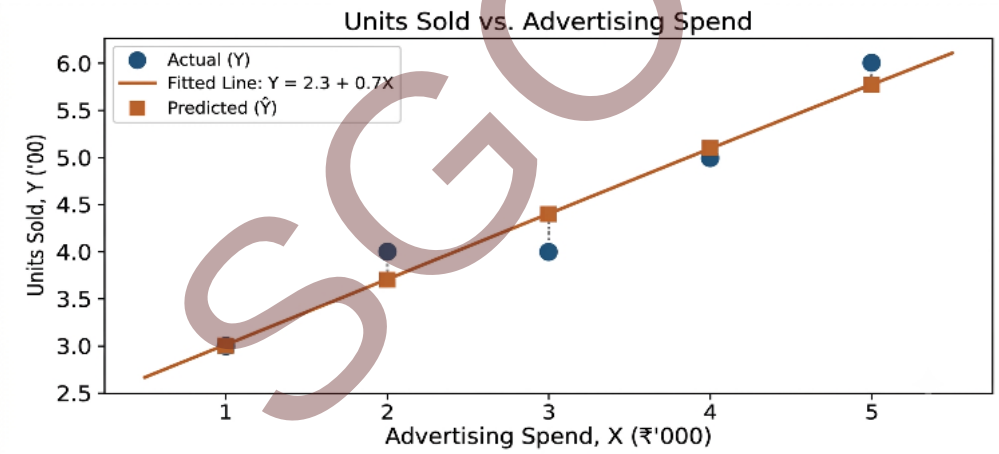


Fig. 3.2.3 Regression Line with Residuals (Dotted Lines = Errors Used to Compute RMSE)

Sum of Squared Residuals = 0.00 + 0.09 + 0.16 + 0.01 + 0.04 = 0.30

Mean Squared Error (MSE) = 0.30 / 5 = 0.06

RMSE = $\sqrt{0.06} \approx 0.245$ ('00 units, i.e., approximately 24.5 units)

An RMSE of about 24.5 units means that, on average, this model's predictions of units sold are off by about 24.5 units. This is a small error compared with the overall scale of sales in this dataset (roughly 300 to 600 units per month). It fits the very strong correlation ($r \approx 0.971$) found earlier in this unit: a strongly correlated pair of variables has produced a regression model with a small typical error.

SGOU

RMSE is particularly useful for two reasons. First, unlike a plain average of residuals, positive and negative errors cannot cancel out and artificially reduce it. Second, squaring the residuals means RMSE penalizes large errors more heavily than small ones. One badly mispredicted observation raises RMSE more than several small errors of the same total size. This is often what a business wants, since a few very wrong predictions can do more harm than many slightly-off ones.

It is worth pausing on why this sensitivity to large errors matters in a concrete business setting. For the telecom churn project, imagine two candidate models with the same average error when measured a simpler way, but one model makes small, evenly-spread mistakes across many customers, while the other makes almost perfect predictions for most customers but badly misjudges a handful of high-value accounts. RMSE would flag the second model as considerably worse, even though a simpler metric might rate the two as similar, precisely because badly misjudging a small number of important cases is usually far more costly to a business than being slightly off on many minor ones. This is one of the main reasons RMSE remains the default error metric reported for regression models across so much of data analytics practice, despite the existence of several reasonable alternatives.

3.2.8 Related Error Metrics: MAE and R-Squared

RMSE is one of several widely used metrics for evaluating a regression model's accuracy. Two other commonly encountered metrics are briefly introduced here for context.

Mean Absolute Error (MAE): Rather than squaring residuals, MAE simply averages the absolute (unsigned) value of every residual:

$$MAE = \Sigma|Actual - Predicted| / n$$

Worked Example: Using the residuals from Table 3.2.3 (0.0, 0.3, -0.4, -0.1, 0.2), taking absolute values gives 0.0, 0.3, 0.4, 0.1, 0.2.

$$MAE = (0.0 + 0.3 + 0.4 + 0.1 + 0.2) / 5 = 1.0 / 5 = 0.2$$

Unlike RMSE, MAE treats every unit of error equally regardless of whether it comes from one large mistake or several small ones, making it somewhat less sensitive to occasional large errors than RMSE, which is usually slightly larger than MAE for the same dataset (0.245 versus 0.2 in this example) precisely because RMSE penalizes larger errors more heavily.

R-squared (R^2): R-squared measures the proportion of the total variation in the outcome variable that is explained by the regression model, and is calculated as:

$$R^2 = 1 - (\text{Sum of Squared Residuals} / \Sigma(y - \bar{y})^2)$$

$$R^2 = 1 - (0.30 / 5.2) = 1 - 0.0577 \approx 0.942$$

An R^2 of approximately 0.942 means that about 94.2% of the variation in units sold across these five months is explained by advertising spend alone, leaving only about 5.8% unexplained by the model. This value is almost identical to r^2 ($0.971^2 \approx 0.943$),

SGOU

and the small difference is due to rounding. This is not a coincidence. For simple linear regression with one input variable, R^2 is mathematically equal to the square of the correlation coefficient r. This links the two halves of this unit: the strength of the original relationship, and the accuracy of the model built on it.

This equivalence is a useful sanity check with real data. If a fitted model's R^2 is very different from the square of the correlation coefficient between its two variables, something may have gone wrong, for example a mistake in the regression calculation itself. With several input variables, the simple equivalence no longer holds exactly. In that case, look more closely at which variables contribute most to the model's explanatory power.

3.2.9 Comparing the Correlation Coefficient and RMSE

Although both the correlation coefficient and RMSE are built on the same underlying idea of squared deviations, they serve genuinely different purposes at different stages of an analytics project, as summarized in Table 3.2.4.

Table 3.2.4 Comparing the Correlation Coefficient and RMSE

Aspect	**Correlation Coefficient (r)**	**Root Mean Square Error (RMSE)**
When It Is Used	Before modelling, to assess whether two variables appear related	After modelling, to assess how accurate a model's predictions are
What It Measures	The strength and direction of a linear relationship between two variables	The typical size of a model's prediction errors
Typical Range	Always between −1 and +1	Zero (a perfect model) upward, in the same units as the outcome variable
Relationship in Simple Regression	R^2 (proportion of variance explained) equals r^2	A smaller RMSE generally corresponds to a stronger underlying correlation

In a well-run analytics project, correlation and error metrics such as RMSE are used together, not as alternatives. Correlation helps decide early on which variables are worth including in a model. RMSE (and related metrics such as MAE and R^2) is used afterwards, and repeatedly, to compare candidate models and to monitor how well a deployed model keeps performing on new, real-world data.

SGOU

3.2.10 Applications in Data Analytics

The measures introduced in this unit are applied constantly throughout data analytics practice:

- Feature Selection: Before building a machine learning model, analysts routinely calculate correlations between candidate input variables and the outcome of interest, to prioritize the most promising variables and avoid including redundant, highly correlated inputs.
- Model Comparison: When several candidate regression models are available for the same problem, RMSE (or MAE) is commonly used to objectively compare their accuracy and select the best-performing one.
- Forecast Evaluation: RMSE is one of the standard metrics used to evaluate the accuracy of time series forecasts, a topic introduced in Unit 3 of this block, by comparing forecasted values against what actually occurred.
- Avoiding Misleading Conclusions: Understanding the difference between correlation and causation helps analysts avoid recommending ineffective or even counterproductive business actions based on a correlation that does not reflect a genuine causal relationship.

Bringing the two halves of this unit together: for the telecom churn project, the intern would use correlation early on to check which candidate variables are worth including when building her decision tree or Naive Bayes classifier from Unit 1. The variables are contract type, support call frequency and monthly bill amount. Once a model is built, she would use RMSE (for a numeric prediction such as expected customer lifetime value), or comparable classification accuracy measures, to judge whether the model is good enough to take back to her manager with confidence. Used together in this way, correlation and error metrics such as RMSE form a complete, evidence-based cycle running from initial variable selection through to final model evaluation.

Now that you have learned how to measure the strength of a relationship between two variables and how to evaluate the accuracy of a fitted model's predictions, you are ready for Unit 3 of this block. That unit applies many of the same ideas to data collected over time. It introduces time series analysis and trend analysis as a further extension of the regression concepts in this block.

Recap

- ♦ Covariance measures whether two variables tend to move together, but its value depends on the units of measurement, limiting direct interpretation.
- ♦ Pearson's correlation coefficient (r) rescales covariance to always fall between −1 and +1, making the strength and direction of a linear relationship directly interpretable and comparable across variables.

SGOU

- The sign of r indicates direction (positive or negative), while its magnitude indicates strength; r captures only linear relationships and should always be interpreted alongside a scatter plot.
- A strong correlation does not prove causation; two variables may be correlated because of a confounding variable that independently influences both, as in the classic ice cream sales and drowning incidents example.
- A residual is the difference between an actual value and a model's predicted value; Root Mean Square Error (RMSE) squares, averages, and takes the square root of residuals to summarize a model's typical prediction error in the original units.
- RMSE penalizes large errors more heavily than small ones, while Mean Absolute Error (MAE) treats all errors proportionally to their size; R-squared measures the proportion of variance in the outcome explained by the model, and equals r^2 in simple linear regression.
- Correlation is used before modelling to assess whether variables appear related, while RMSE and related error metrics are used after modelling to assess and compare how accurate the resulting predictions actually are.

Objective Type Questions

1. What does covariance measure?
2. What is the main limitation of covariance as a measure of relationship strength?
3. What is the range of possible values for Pearson's correlation coefficient, r?
4. What does a negative correlation coefficient indicate?
5. According to the general guideline in this unit, what strength of linear relationship does $|r| = 0.85$ indicate?
6. Why should a scatter plot always be examined alongside a calculated correlation coefficient?
7. What is a confounding variable?
8. In the ice cream sales and drowning incidents example, what is the confounding variable?
9. What is a residual in the context of a regression model?
10. What is the formula for Root Mean Square Error (RMSE)?

SGOU

11. Why are residuals squared before being averaged in RMSE, rather than averaged directly?
12. What is the main difference between RMSE and Mean Absolute Error (MAE)?
13. What does R-squared (R^2) measure?
14. In simple linear regression, what is the mathematical relationship between R^2 and r?
15. At what stage of an analytics project is correlation typically used, compared to RMSE?

Answers

1. The extent to which two variables change together (positively, negatively, or not at all)
2. Its value depends on the units of measurement, making it hard to judge relationship strength directly
3. Between −1 and +1
4. That one variable tends to increase as the other decreases
5. A strong linear relationship
6. Because correlation only captures linear relationships, and a strong non-linear (e.g., curved) relationship can still produce a correlation coefficient close to zero
7. A third variable that independently influences two other variables, creating a correlation between them without either causing the other
8. Hot weather
9. The difference between the actual observed value and the value predicted by the model
10. RMSE = $\sqrt{[\Sigma(\text{Actual} - \text{Predicted})^2 / n]}$
11. To prevent positive and negative errors from cancelling out and hiding the true extent of a model's inaccuracy
12. RMSE penalizes large errors more heavily than small ones, while MAE treats errors proportionally to their size

SGOU

13. The proportion of the total variation in the outcome variable explained by the regression model

14. R^2 equals r^2 (the square of the correlation coefficient)

15. Correlation is typically used before modelling; RMSE is typically used after modelling

Assignments

1. For the data X = 2, 4, 6, 8, 10 and Y = 3, 5, 6, 9, 10, calculate $\bar{x}$, $\bar{y}$, $\Sigma(x-\bar{x})(y-\bar{y})$, $\Sigma(x-\bar{x})^2$, and $\Sigma(y-\bar{y})^2$, and use them to calculate the covariance and the correlation coefficient r.

2. Using the correlation coefficient calculated in Question 1, classify the strength of the relationship using the guideline in Table 3.2.2, and state whether the relationship is positive or negative.

3. Describe a real-world example of your own (different from the ice cream and drowning example used in this unit) where two variables might be strongly correlated without one causing the other, and identify a plausible confounding variable.

4. A regression model predicts the following values against actual observations: Actual = 10, 12, 14, 16; Predicted = 11, 12, 13, 17. Calculate the residuals, the RMSE, and the MAE for this model, and state which metric is larger and why.

5. Explain, in your own words, why a data analyst evaluating two different regression models for the same problem might prefer the model with the lower RMSE, even if both models have a similar R^2 value.

SGOU

Suggested Reading

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.

2. Agarwal, B. L. (2013). Basic Statistics. New Age International.

3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

Reference

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.
2. Agarwal, B. L. (2013). Basic Statistics. New Age International.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

SGOU



Time Series Analysis and Trend Analysis

Learning Outcomes

After completion of this unit, learner will be able to:

- define a time series and distinguish its four main components: trend, seasonality, cyclic variation, and irregular (random) variation
- calculate and interpret a moving average to smooth short-term fluctuations in a time series
- fit a linear trend line to a time series using regression on a time index
- calculate a seasonal index and use it to describe recurring seasonal patterns in data
- explain the concept of stationarity and describe how differencing can help remove a trend
- describe basic time series forecasting approaches, including the naive method, moving-average forecasting, and trend-based forecasting

SGOU

Prerequisite

Picture the intern, opening a fresh spreadsheet of the past 12 months of customer complaints, unsure for the first time what to actually do with it, since nothing in her decision tree or Naive Bayes toolkit from Unit 1 was built with the idea of 'time' in mind at all. What she needs is a way of thinking about a single variable, complaints, not as 12 independent observations but as a connected sequence -- and, conveniently, the very same regression techniques she used to relate advertising spend to sales in Unit 1 turn out to apply almost unchanged once 'time' itself is used as the input variable.

A comfortable working knowledge of the mean (Block I, Unit 1) and simple linear regression (Block III, Unit 1) is all that is required to follow this unit; no prior exposure to time series analysis of any kind is assumed.

Keywords

Time Series, Trend, Seasonality, Cyclic Variation, Irregular Variation, Moving Average, Linear Trend, Seasonal Index, Stationarity, Differencing, Forecasting.

Discussion

3.3.1 Introduction to Time Series Data

A time series is a sequence of observations of the same variable, recorded at successive, typically equally spaced points in time -- for example, a company's monthly sales figures, the daily closing price of a stock, or, as followed throughout this unit, the number of customer complaints a telecom company receives each month. A time series differs from the data in earlier units because the order of the observations carries meaning. Rearranging a time series destroys information that is central to understanding it. A simple list of customer ages, by contrast, can be sorted or shuffled without losing anything important.

Time series analysis is the practice of describing, summarizing and modelling this kind of ordered data. It has two closely related goals. The first is to understand what patterns have driven the variable's past behaviour, which is the focus of most of this unit. The second is to predict its likely future values (forecasting), which is introduced briefly in Section 3.3.7 and developed further in Unit 4 of this block.

It is worth pausing on why time series data needs its own dedicated set of techniques at all, rather than simply being analyzed using the descriptive statistics and regression methods already covered earlier in this course. The mean and standard deviation introduced in Block I, Unit 1 summarize a set of numbers without any regard to the order in which they occurred, which is entirely appropriate for a dataset such as a class's exam scores, where no meaningful 'before' and 'after' exists between one student's score and another's. A time series is fundamentally different: knowing that complaints were low in January and high in December is a materially different, and more useful, piece of information than simply knowing that the average complaint count across the year was such-and-such a number. Time series analysis exists precisely to preserve and make use of that ordering, which a simple average would otherwise discard entirely.

3.3.2 Components of a Time Series

A time series is typically thought of as being made up of four distinct components, each capturing a different kind of pattern in how the variable changes over time.

Separating a time series into these four components serves a practical purpose beyond mere academic tidiness: each component often calls for a different business response. A rising trend in customer complaints, for instance, may call for a structural fix -- retraining support staff, or addressing a recurring product issue -- while a predictable seasonal spike each December may simply call for temporary additional staffing during

SGOU

that period each year, with no deeper structural problem implied at all. Confusing the two can lead a business to spend time and money on a problem that was never there. For example, a normal seasonal spike may be mistaken for a worsening structural trend. This is why splitting a series into its separate components, as described in the rest of this section, is a valuable habit for any data analyst.

- **Trend :** The long-term, general direction in which the data is moving -- steadily increasing, steadily decreasing, or relatively flat -- over an extended period, ignoring short-term fluctuations.
- **Seasonality :** A pattern that repeats at fixed, known intervals related to the calendar -- for example, ice cream sales rising every summer and falling every winter, year after year.
- **Cyclic Variation :** Fluctuations that rise and fall, but not at fixed calendar-based intervals -- for example, broader economic boom-and-bust cycles, which may each last several years and vary in length from one cycle to the next.
- **Irregular (Random) Variation:** Unpredictable, short-term fluctuations remaining after the trend, seasonal, and cyclic components have been accounted for, arising from unforeseeable one-off events or simple random noise.

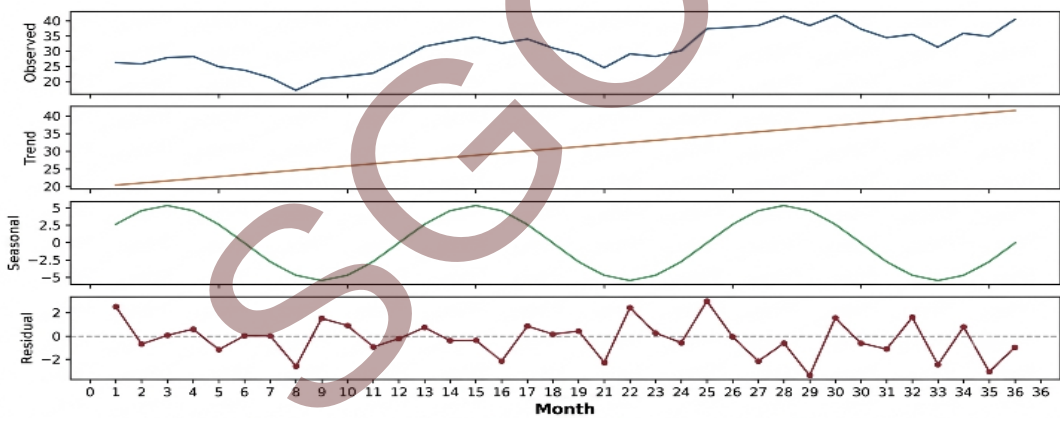


Fig. 3.3.1 Illustrative Decomposition of a Time Series into Trend, Seasonal, and Residual Components

As Figure 3.3.1 illustrates using an artificially generated example, decomposing an observed time series (top panel) reveals a steadily rising trend (second panel) and a repeating seasonal wave (third panel); what remains after removing both (bottom panel) is the irregular, essentially unpredictable residual variation. Real-world time series rarely separate this cleanly, and cyclic variation, in particular, is often difficult to distinguish clearly from a slowly changing trend, but this four-part framework remains a useful starting point for describing almost any time series encountered in practice. The remainder of this unit focuses primarily on trend (Sections 3.3.3 to 3.3.5) and seasonality (Section 3.3.6), the two components most directly relevant to everyday business analytics.

SGOU

Table 3.3.1The Four Components of a Time Series, with Typical Time Scales and Examples

Component	Typical Time Scale	Illustrative Example
Trend	Long-term, spanning the entire series	A steady rise in a company's customer base over several years
Seasonality	Fixed, calendar-based (e.g., quarterly, monthly, weekly)	Ice cream sales peaking every summer, year after year
Cyclic Variation	Longer, irregular in length (often multiple years)	Broader economic expansion and recession cycles
Irregular Variation	Short-term, unpredictable	A one-off spike in website traffic following an unplanned viral social media post

Distinguishing cyclic variation from trend deserves a brief additional comment, since the two are easily confused. A trend is assumed to continue in roughly the same direction over the period being studied, while a cyclic pattern is expected to eventually reverse -- an economic expansion is always eventually followed by a slowdown, even though the exact timing and length of each cycle varies unpredictably. In practice, unless a series is observed over a very long period, a genuine cyclic pattern can be difficult to distinguish from a trend that simply has not yet reversed, and analysts often use the term 'trend-cycle' to acknowledge this practical difficulty rather than force an artificial separation between the two.

3.3.3 Trend Analysis: The Moving Average

A moving average is one of the simplest and most widely used techniques for revealing the underlying trend in a time series, by smoothing out short-term, irregular fluctuations. A k-period moving average replaces each observation with the average of itself and the surrounding k−1 observations, and is recalculated ('moved') one period at a time across the whole series.

$$\text{3-Period Moving Average at time t} = (Y[t-1] + Y[t] + Y[t+1]) / 3$$

Worked Example: A telecom company records the following number of customer complaints over 12 consecutive months: 20, 22, 19, 23, 25, 24, 28, 30, 27, 32, 35, 33.

Table 3.3.2 Monthly Complaints with a 3-Month Moving Average

Month	1	2	3	4	5	6	7	8	9	10	11	12
Complaints	20	22	19	23	25	24	28	30	27	32	35	33
3-Mo. MA	--	20.33	21.33	22.33	24.00	25.67	27.33	28.33	29.67	31.33	33.33	

Notice that the moving average column is noticeably smoother than the raw monthly figures -- the sharp month-to-month bounces (for example, the drop from 22 in month 2 to 19 in month 3) are smoothed away, revealing a clear, steadily rising underlying trend

SGOU

that is harder to see in the raw data alone. This smoothing comes at a cost: a moving average cannot be calculated for the first and last (k−1)/2 periods of the series (month 1 and month 12 in this example), since it requires data on both sides of each point.

The choice of k, the number of periods averaged together, involves a trade-off that an analyst must judge based on the data at hand. A larger k (for example, a 12-month moving average applied to monthly data) smooths out even more short-term noise, and is particularly useful for removing seasonality entirely, but reacts more slowly to a genuine, recent change in the underlying trend, since older observations continue to influence the average for longer. A smaller k (for example, a 3-month moving average, as used in this unit's worked example) responds more quickly to recent changes, but retains more of the short-term noise the moving average is meant to smooth away in the first place. In practice, k is often chosen to match a series' natural cycle length -- a 12-month moving average is a natural choice for monthly data suspected of having yearly seasonality, since averaging across a full year cancels out the seasonal pattern along with the shorter-term noise.

3.3.4 Trend Analysis: Fitting a Linear Trend

A moving average smooths a series but does not give a formula. Trend analysis can also be done by fitting a regression equation, using the simple linear regression technique from Block III, Unit 1. The difference is that the input variable is now a time index (month 1, month 2, month 3 and so on) rather than a business variable such as advertising spend:

$$\hat{Y} = b0 + b1 \cdot t$$

Worked Example: Using the same 12 months of complaints data, with the time index t = 1, 2, ..., 12 and $\bar{t} = 6.5$, $\bar{Y} = 26.5$, the same regression formulas from Block III, Unit 1 give:

$$b1 = \Sigma(t - \bar{t})(Y - \bar{Y}) / \Sigma(t - \bar{t})^2 = 195.0 / 143.0 \approx 1.364$$

$$b0 = \bar{Y} - b1 \cdot \bar{t} = 26.5 - (1.364 \times 6.5) \approx 17.64$$

Fitted Trend Line: $\hat{Y} = 17.64 + 1.364t$

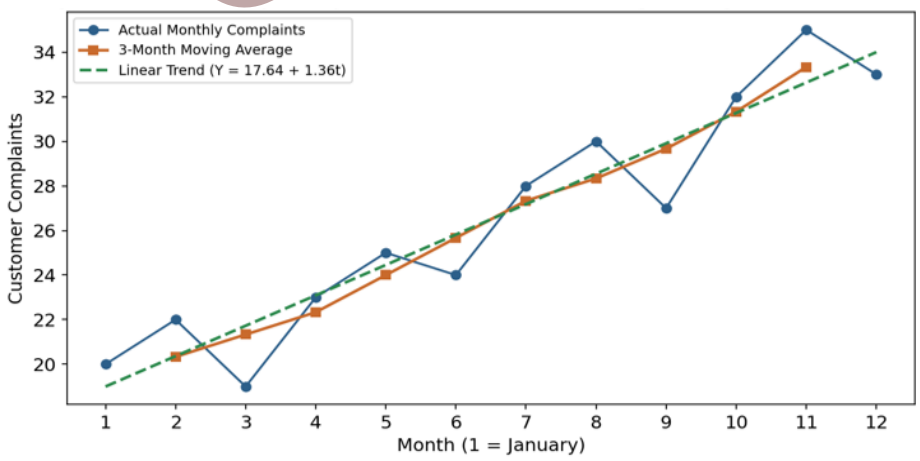


Fig. 3.3.2 Monthly Complaints with 3-Month Moving Average and Linear Trend

SGOU

As shown in Figure 3.3.2, both the moving average and the fitted trend line agree on the same basic story -- customer complaints are rising by roughly 1.36 complaints per month, on average, over this period. The fitted equation has an important further advantage over the moving average: it can be extrapolated beyond the observed data to produce a forecast for a future period, simply by substituting a value of t beyond the range already observed, a technique returned to in Section 3.3.8.

It is worth being careful, however, about how far beyond the observed data a fitted trend line is extrapolated. The regression equation $\hat{Y} = 17.64 + 1.364t$ was estimated from only 12 months of data. There is no guarantee that a trend that held steadily for one year will continue unchanged for many years. A rising trend in customer complaints cannot continue forever if left unaddressed, because it would eventually mean more complaints than the company has customers. Extrapolating a short distance beyond the observed data, as in the month 13 forecast used later in this unit, is generally more defensible than projecting a fitted trend line many years into the future, where the risk of the underlying trend changing, levelling off, or reversing grows considerably.

3.3.5 Detecting Seasonality: The Seasonal Index

While the complaints data used in Sections 3.3.3 and 3.3.4 shows a clear trend but no obvious seasonal pattern over a single year, many real-world time series show both a trend and a clear seasonal pattern together, repeating at the same point in the calendar year after year. A seasonal index quantifies the typical effect of each season (or month, or quarter) relative to the overall average, and is calculated as:

Seasonal Index (for a given period) = Average value for that period across all years / Overall average of all data

Worked Example: An ice cream parlour records the following quarterly sales (in ₹'000) over two years:

Table 3.3.3 Quarterly Ice Cream Sales (₹'000) Over Two Years

Year	Q1 (Jan-Mar)	Q2 (Apr-Jun)	Q3 (Jul-Sep)	Q4 (Oct-Dec)
Year 1	100	180	200	110
Year 2	120	200	220	130
2-Year Average	110	190	210	120

Overall Average (all 8 quarters) = (100+180+200+110+120+200+220+130) / 8= 157.5

Seasonal Index: Q1 = 110/157.5 ≈ 0.698 Q2 = 190/157.5 ≈ 1.206 Q3 = 210/157.5 ≈ 1.333 Q4 = 120/157.5 ≈ 0.762

SGOU

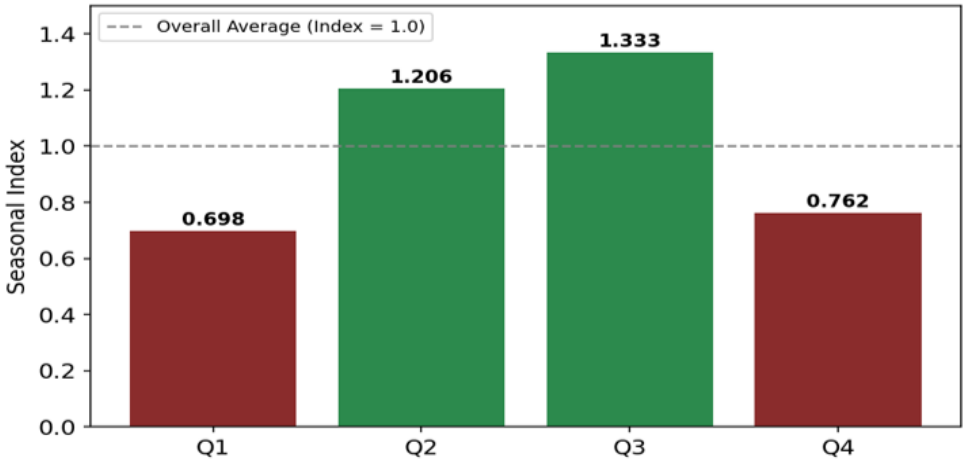


Fig. 3.3.3 Seasonal Index by Quarter -- Ice Cream Sales Example

Source: Self-generated using Python (Matplotlib), created for this Self-Learning Material (SLM).

As Figure 3.3.3 shows, a seasonal index of 1.333 for Q3 means that sales in Q3 (the peak summer months) are typically about 33.3% above the overall quarterly average, while a seasonal index of 0.698 for Q1 means Q1 sales are typically about 30.2% below average. These indices can be used to adjust raw data for seasonality (a process known as seasonal adjustment) or to build a seasonally aware forecast, by combining an estimated trend with the appropriate seasonal index for the period being forecast, a refinement briefly revisited in Section 3.3.8.

Seasonal adjustment is particularly valuable when comparing performance across consecutive periods that are not naturally comparable due to seasonality alone. Suppose the ice cream parlour compared its Q4 sales of ₹120,000 directly with its Q3 sales of ₹210,000. It might wrongly conclude that the business is declining sharply. In fact, the drop is fully explained by the normal seasonal shift from summer to autumn. It says nothing about the underlying health of the business. Dividing each quarter's raw sales figure by its corresponding seasonal index produces a seasonally adjusted figure that strips out this predictable calendar effect, allowing a fairer, apples-to-apples comparison of the business's genuine underlying trend from one quarter to the next.

3.3.6 Stationarity and Differencing

A time series is said to be stationary if its statistical properties -- particularly its mean and variance -- remain roughly constant over time, with no systematic trend or seasonal pattern. Stationarity matters because many more advanced time series forecasting techniques (beyond the scope of this introductory unit) assume, as a mathematical requirement, that the series being modelled is stationary, or can be made stationary through a suitable transformation.

The complaints data used throughout this unit is clearly non-stationary, since its mean is steadily increasing over time (as confirmed by the upward-sloping trend line fitted in Section 3.3.4). A common and simple technique for removing a trend and moving a series closer to stationarity is differencing, which replaces each observation with the difference between it and the previous observation:

SGOU

$$\text{Differenced Value at time } t = Y[t] - Y[t-1]$$

Worked Example: Applying first differencing to the complaints data (20, 22, 19, 23, 25, 24, 28, 30, 27, 32, 35, 33):

Differenced Series: 2, -3, 4, 2, -1, 4, 2, -3, 5, 3, -2

Unlike the original series, which climbed steadily from 20 to 33, the differenced series fluctuates back and forth around a small positive average (approximately 1.18), with no obvious continuing upward drift -- exactly the kind of more stable, trend-free behaviour that stationarity requires. Differencing does not change the underlying information present in a series; it simply re-expresses it in terms of period-to-period change rather than absolute level, a transformation that more advanced forecasting techniques frequently apply as a first step before modelling.

It is worth being clear about what stationarity does and does not imply in practice. A stationary series is not necessarily 'flat' in the sense of every value being identical -- the differenced complaints series above still moves up and down noticeably from month to month. What stationarity requires is that the statistical character of that movement (roughly how far values typically deviate from a stable central value, and how that central value itself behaves) does not systematically change over time. This distinction matters because many advanced forecasting techniques, which are only briefly introduced in this course and covered in more depth in dedicated time series courses, assume that the data is stationary. If such techniques are applied directly to a strongly trending, non-stationary series, such as the raw complaints data, the forecasts can be calculated but still unreliable. This is why checking for non-stationarity, and correcting it where needed, is treated as an essential early step in more advanced time series work.

3.3.7 An Introduction to Time Series Forecasting

Having described the trend and seasonal patterns present in a time series, a natural next question is how to use that description to predict future values -- a process known as forecasting, developed in greater depth in Unit 4 of this block, particularly in combination with the text analytics techniques introduced there. Three simple forecasting approaches follow directly from the tools already introduced in this unit:

- Naive Forecast: Simply predicts that the next period's value will equal the most recently observed value. Despite its simplicity, a naive forecast is a useful baseline against which more sophisticated methods should always be compared -- a genuinely useful forecasting method ought to outperform this trivial baseline by a meaningful margin.
- Moving-Average Forecast: Predicts that the next period's value will equal the average of the most recent k periods, using the same moving-average calculation introduced in Section 3.3.3.
- Trend-Based Forecast: Extrapolates the fitted linear trend equation from Section 3.3.4 beyond the range of observed data, by substituting a future time index into the fitted equation.

SGOU

Worked Example: Using the trend equation fitted in Section 3.3.4, $\hat{Y} = 17.64 + 1.364t$, a trend-based forecast for month 13 (the next, unobserved month) is:

$$\hat{Y}(\text{month } 13) = 17.64 + (1.364 \times 13) \approx 35.36 \text{ complaints}$$

By comparison, a naive forecast for month 13 would simply predict 33 complaints (the actual value observed in month 12), while a moving-average forecast using the final available 3-month average would predict approximately 33.33 complaints (see Table 3.3.2). All three methods point to a broadly similar figure in the mid-thirties, but the trend-based forecast captures the ongoing upward direction of the series more explicitly than the other two methods, which each rely more heavily on only the most recent observations.

In practice, the choice between these and other, more advanced forecasting methods depends on two things. One is how much of a series' recent history can reasonably be expected to continue. The other is how strong and stable the series' trend and seasonal patterns have proved to be. These questions return, together with the accuracy metrics from Unit 2 of this block, when forecasting is developed further in Unit 4.

It is worth explicitly connecting this final point back to Unit 2 of this block: none of these three forecasting methods should simply be assumed to be the best choice for a given series without evidence. A responsible analyst would typically hold back the most recent few periods of a series as a simple test set, generate forecasts for those periods using each candidate method, and compare the resulting errors using RMSE or MAE, exactly as introduced in Unit 2, before committing to one method over another for genuinely unknown future periods. Forecasting, in this sense, is not a separate topic from model evaluation, but rather a direct, practical application of it to data ordered in time.

3.3.8 Applications of Time Series and Trend Analysis in Data Analytics

The techniques introduced in this unit are applied constantly across business and data analytics:

- Sales and Demand Planning: Retailers rely heavily on trend and seasonal analysis to plan inventory and staffing levels -- correctly anticipating a Q3 sales peak, as seen in this unit's ice cream example, well before it occurs.
- Financial Markets: Stock prices, exchange rates, and economic indicators are routinely analyzed for trend and cyclic patterns to inform investment decisions.
- Customer Analytics: Monitoring whether a metric such as monthly churn or customer complaints is trending upward or downward over time, as followed throughout this unit, often provides an earlier and clearer warning sign of a developing business problem than any single month's figure viewed in isolation.

SGOU

- Operations and Capacity Planning: Server load, call-centre volume, and manufacturing output are commonly analyzed for both trend and seasonality to plan infrastructure and staffing capacity efficiently throughout the year.

It is also worth noting how naturally the techniques introduced in this unit build on, and feed into, the rest of this course. The moving average and linear trend line apply averaging (Block I) and regression (Block III, Unit 1) to a new kind of input: time itself. The seasonal index is a set of group averages compared with an overall average, echoing the descriptive statistics of Block I, Unit 1. The forecasting methods in Section 3.3.7 will shortly be evaluated using the RMSE and MAE metrics from Unit 2 of this block. Very little in this unit is conceptually new; what is new is applying already-familiar tools to data where the calendar itself carries meaning.

Returning one final time to the telecom churn project: with the trend and seasonal analysis from this unit, the intern can tell her manager more than which customers are likely to churn. She can also say whether churn as a whole is getting better or worse over time, and whether it follows a predictable seasonal pattern worth planning around. This is a much richer answer than her original models from Unit 1 could give on their own. Having learned to describe and extrapolate patterns in data collected over time, the learner is now well prepared for Unit 4 of this block, which extends forecasting further and introduces text analytics techniques, including sentiment analysis, for working with unstructured text data collected over time.

Recap

- A time series is a sequence of observations recorded at successive points in time, in which the order of observations carries essential information.
- A time series can be described in terms of four components: trend (long-term direction), seasonality (fixed, calendar-based repetition), cyclic variation (longer, non-fixed fluctuations), and irregular variation (unpredictable residual noise).
- A moving average smooths short-term fluctuations by averaging each observation with its neighbours, revealing the underlying trend more clearly, at the cost of being unavailable for the first and last few periods of the series.
- A linear trend line, fitted using the same regression technique as Block III, Unit 1 but with time as the input variable, provides an explicit equation describing a series' overall direction and can be extrapolated to forecast future periods.
- A seasonal index expresses how much a given period (such as a quarter) typically deviates from the overall average, and can be used to describe or adjust for recurring seasonal patterns.
- A time series is stationary if its mean and variance remain roughly constant over time; differencing (subtracting each observation from the previous one) is a simple technique for reducing or removing a trend and moving a series closer to stationarity.

SGOU

- Simple forecasting approaches include the naive method (repeat the last observed value), the moving-average forecast, and the trend-based forecast (extrapolating a fitted trend line), each providing a useful baseline before more advanced forecasting techniques are applied.

Objective Type Questions

1. What distinguishes a time series from other types of data examined earlier in this course?
2. What are the four components of a time series?
3. Which component of a time series repeats at fixed, calendar-based intervals?
4. Which component of a time series remains after trend, seasonal, and cyclic patterns have been removed?
5. What is the purpose of calculating a moving average?
6. Why can a 3-month moving average not be calculated for the first and last month of a 12-month series?
7. What input variable is used when fitting a linear trend line to a time series, in place of an ordinary business variable?
8. How is a seasonal index calculated?
9. In the ice cream sales example, what does a seasonal index of 1.333 for Q3 indicate?
10. What does it mean for a time series to be stationary?
11. What is differencing, and what is it typically used for?
12. What does a naive forecast predict for the next period?
13. What does a moving-average forecast predict for the next period?
14. What does a trend-based forecast predict for the next period?
15. Why is the naive forecast useful even though it is very simple?

SGOU

Answers

1. The order of its observations carries meaning, unlike data that can be freely rearranged
2. Trend, seasonality, cyclic variation, and irregular (random) variation
3. Seasonality
4. Irregular (random) variation
5. To smooth out short-term fluctuations and reveal the underlying trend in a time series
6. Because a moving average requires observations on both sides of each point, which are unavailable at the very start and end of the series
7. A time index (e.g., month 1, month 2, month 3, ...)
8. As the average value for that period across all years, divided by the overall average of all the data
9. That sales in Q3 are typically about 33.3% above the overall quarterly average
10. That its mean and variance remain roughly constant over time, with no systematic trend or seasonal pattern
11. Subtracting each observation from the previous one; it is used to remove a trend and move a series closer to stationarity
12. The most recently observed value
13. The average of the most recent k periods
14. The value obtained by extrapolating the fitted linear trend equation to a future time index
15. Because it serves as a simple baseline that any genuinely useful forecasting method should be able to outperform

SGOU

Assignments

1. A company records the following monthly website visits ('000): 50, 55, 52, 58, 60, 57, 63, 65, 62, 68, 70, 67. Calculate the 3-month moving average for months 2 through 11, and describe, in your own words, what the moving average reveals about the underlying trend.
2. Using the same data from Question 1, fit a linear trend line ($\hat{Y} = b0 + b1t$) using the time index t = 1, ..., 12, and use it to forecast website visits for month 13.
3. A clothing retailer records the following quarterly sales (₹'000) over two years: Year 1 -- Q1: 80, Q2: 60, Q3: 70, Q4: 130; Year 2 -- Q1: 90, Q2: 65, Q3: 75, Q4: 140. Calculate the seasonal index for each quarter and identify which quarter shows the strongest seasonal effect.
4. Apply first differencing to the website visits data from Question 1, and comment on whether the differenced series appears closer to stationary than the original series.
5. Explain, in your own words, why a business might prefer a trend-based forecast over a naive forecast when a time series shows a clear, consistent upward or downward direction, and describe a situation in which the naive forecast might actually be the more sensible choice instead.

Suggested Reading

1. Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts.
2. Agarwal, B. L. (2013). Basic Statistics. New Age International.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

Reference

1. Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts.
2. Agarwal, B. L. (2013). Basic Statistics. New Age International.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

SGOU



Text Analytics and Forecasting

Learning Outcomes

After completion of this unit, learner will be able to:

- distinguish lexicon-based and machine-learning-based approaches to sentiment analysis
- apply a simple sentiment lexicon to score the sentiment of a piece of text
- apply the Naive Bayes classifier to classify text sentiment, and explain and correct the zero-frequency problem using Laplace smoothing
- describe common challenges in sentiment analysis, including negation, sarcasm, and domain-specific language
- apply simple exponential smoothing to forecast a future value of a time series
- evaluate and compare the accuracy of different forecasting methods using RMSE
- describe how text analytics and forecasting can be combined to track and project sentiment trends over time

SGOU

Prerequisite

Picture the intern once more, with two folders open side by side: one containing the hundreds of customer complaint transcripts she first encountered in Block II, Unit 3, and one containing the monthly complaint-count trend she analyzed in Unit 3 of this block. She realizes that everything she needs for this final piece of the project is already sitting in tools she has built or seen before -- the bag-of-words representation, the Naive Bayes classifier, RMSE, and moving averages -- and that this unit is mostly about combining and extending them, rather than learning something entirely new.

A comfortable working knowledge of the bag-of-words model (Block II, Unit 3), the Naive Bayes classifier (Block III, Unit 1), RMSE (Block III, Unit 2), and basic forecasting methods (Block III, Unit 3) is assumed throughout this unit.

Keywords

Sentiment Analysis, Sentiment Lexicon, Polarity Score, Laplace Smoothing, Zero-Frequency Problem, Negation, Sarcasm, Simple Exponential Smoothing, Smoothing Constant, Forecast Accuracy.

Discussion

3.4.1 Introduction to Text Analytics and Forecasting

This final unit of Block III brings together two threads that have been developing separately across this course: text analytics, first introduced through tokenization and the bag-of-words model in Block II, Unit 3, and forecasting, first introduced through the naive, moving-average, and trend-based methods of Unit 3 of this block. Here, text analytics is extended specifically into sentiment analysis -- automatically judging whether a piece of text expresses a positive, negative, or neutral opinion -- and forecasting is extended with a further refinement, simple exponential smoothing, that addresses some of the limitations of the simpler methods already introduced.

Bringing these two threads together is not merely a matter of convenience at the end of a course. In practice, a business rarely wants to know only what customers said in the past. It wants to know whether customer sentiment is improving or getting worse, and where it is likely to go next. Answering this needs two techniques together: a text analytics technique to score sentiment, and a forecasting technique to project the score forward in time. This is the combination introduced in Section 3.4.8 of this unit.

It is fitting that this unit brings both threads together in the final unit of the block on Advanced Data Analytics, since sentiment analysis and forecasting are, in a real sense, the most immediately business-facing techniques introduced across all three blocks of this course so far. Earlier units often produced a number or a classification meant mainly for further analysis. A sentiment score and a forecast value are different, because a non-technical manager can act on them directly. 'Customer sentiment fell this quarter' and 'we expect 35 complaints next month' are sentences a business leader understands at once, without needing to know the statistics behind them.

3.4.2 Sentiment Analysis: Two Broad Approaches

Sentiment analysis, briefly introduced in Block II, Unit 3, can be approached in two broad ways, each building on tools already developed earlier in this course.

- Lexicon-Based Sentiment Analysis: Uses a predefined dictionary (lexicon) that assigns a positive or negative score to individual words, and combines the scores of the words present in a piece of text to produce an overall sentiment score. This approach requires no training data at all, but depends entirely on the quality and coverage of the chosen lexicon.
- Machine-Learning-Based Sentiment Analysis: Trains a classifier, such as the Naive Bayes classifier introduced in Block III, Unit 1, on a labelled dataset

SGOU

of text examples already tagged as positive or negative, allowing the model to learn which words (and combinations of words) are associated with each sentiment directly from data, rather than from a fixed, predefined dictionary.

Sections 3.4.3 and 3.4.4 work through a worked example of each approach in turn, using customer feedback text similar in style to the complaint examples first introduced in Block II, Unit 3.

Neither approach is universally superior to the other, and the choice between them mirrors a distinction already drawn elsewhere in this course. A lexicon-based approach can be used at once, with no training data. This makes it well suited to a new business setting where labelled examples of positive and negative feedback do not yet exist. A machine-learning-based approach needs an upfront effort to label a training set, just as the churn classifier in Block III, Unit 1 needed labelled examples of past churned and retained customers. In return, it can learn subtler, domain-specific patterns of language that a general-purpose lexicon would probably miss.

3.4.3 Lexicon-Based Sentiment Analysis: A Worked Example

A sentiment lexicon assigns a numeric score to individual words, typically ranging from strongly negative (for example, −3) to strongly positive (for example, +3), with 0 reserved for neutral words. To score a piece of text, every word present in the lexicon is looked up, and the resulting scores are summed (or averaged) to produce an overall polarity score for the text.

Table 3.4.1 A Small Illustrative Sentiment Lexicon

Word	Score	Word	Score
amazing	+3	unhelpful	-2
fast	+2	crashed	-2
nice	+2	rude	-2

Worked Example: Consider two pieces of customer feedback, scored using the lexicon in Table 3.4.1.

Sentence A: "The new update is amazing and fast." → Score = (+3) + (+2) = +5

Sentence B: "...crashed constantly and support was unhelpful, though the design is nice." → Score = (−2) + (−2) + (+2) = −2

As Figure 3.4.1 shows, Sentence A receives a clearly positive score of +5, reflecting its two straightforwardly positive words. Sentence B is more interesting: despite ending on the positive word 'nice', its overall score of −2 is negative, because the two negative words ('crashed' and 'unhelpful') outweigh the single positive word. This is exactly the mixed-sentiment challenge first previewed in Block II, Unit 3 -- lexicon-based scoring handles it correctly here simply because it sums every scored word in the sentence rather than reacting only to the most recent or most prominent one, though, as discussed

SGOU

further in Section 3.4.5, this simple word-by-word summing approach has its own limitations.

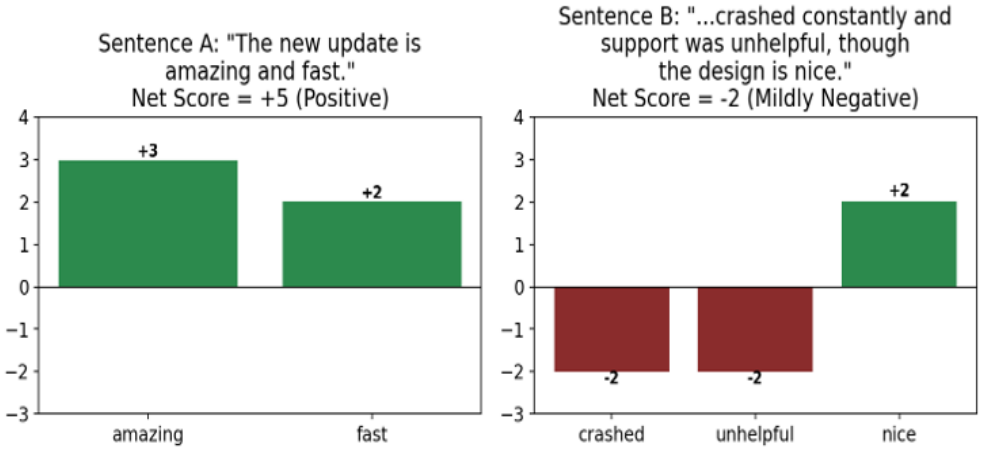


Fig.3.4.1 Word-Level Contributions in Lexicon-Based Sentiment Analysis

As Figure 3.4.1 shows, Sentence A receives a clearly positive score of +5, reflecting its two straightforwardly positive words. Sentence B is more interesting: despite ending on the positive word 'nice', its overall score of −2 is negative, because the two negative words ('crashed' and 'unhelpful') outweigh the single positive word. This is exactly the mixed-sentiment challenge first previewed in Block II, Unit 3 -- lexicon-based scoring handles it correctly here simply because it sums every scored word in the sentence rather than reacting only to the most recent or most prominent one, though, as discussed further in Section 3.4.5, this simple word-by-word summing approach has its own limitations.

3.4.4 Machine-Learning-Based Sentiment Analysis Using Naive Bayes

Rather than relying on a predefined lexicon, a Naive Bayes classifier -- introduced in Block III, Unit 1 for churn prediction -- can instead be trained directly on a labelled set of customer reviews to learn which words are associated with positive and negative sentiment. Consider a training set of 10 customer reviews, 6 labelled Positive and 4 labelled Negative, examined for the presence of two words: 'great' and 'slow'.

Quantity	Positive Reviews (n = 6)	Negative Reviews (n = 4)
Contains 'great'	4 of 6 reviews	0 of 4 reviews
Contains 'slow'	0 of 6 reviews	3 of 4 reviews

Table 3.4.2 Word Presence Counts by Sentiment Class in the Training Data

Worked Example: A new review reads: 'Great service but slow delivery', containing both 'great' and 'slow'. Applying the Naive Bayes formula from Block III, Unit 1 directly, using the raw proportions in Table 3.4.2:

SGOU

P(great | Positive) = 4/6 ≈ 0.667 P(slow | Positive) = 0/6 = 0

P(great | Negative) = 0/4 = 0 P(slow | Negative) = 3/4 = 0.75

Score(Positive) = 0.6 × 0.667 × 0 = 0 Score(Negative) = 0.4 × 0 × 0.75 = 0

The Zero-Frequency Problem: Both scores come out to exactly zero, leaving the classifier completely unable to distinguish between the two classes, even though the evidence is genuinely mixed (one word favouring each class) rather than completely uninformative. This happens because a single unseen word/class combination in the training data (no positive review contained 'slow', and no negative review contained 'great') forces an entire product of probabilities to zero, regardless of how informative the other features might be. This is known as the zero-frequency problem, and is a well-known practical limitation of the basic Naive Bayes formula introduced in Block III, Unit 1.

Laplace (Add-One) Smoothing: The standard fix is to add 1 to every count before calculating a probability, and to add the number of possible categories for that feature (2, for 'present' or 'absent') to the denominator, ensuring no probability is ever exactly zero:

P(great | Positive) = (4+1)/(6+2) = 0.625 **P(slow | Positive) = (0+1)/(6+2) = 0.125**

P(great | Negative) = (0+1)/(4+2) ≈ 0.167 **P(slow | Negative) = (3+1)/(4+2) ≈ 0.667**

Score(Positive) = 0.6 × 0.625 × 0.125 ≈ 0.0469 **Score(Negative) = 0.4 × 0.167 × 0.667 ≈ 0.0444**

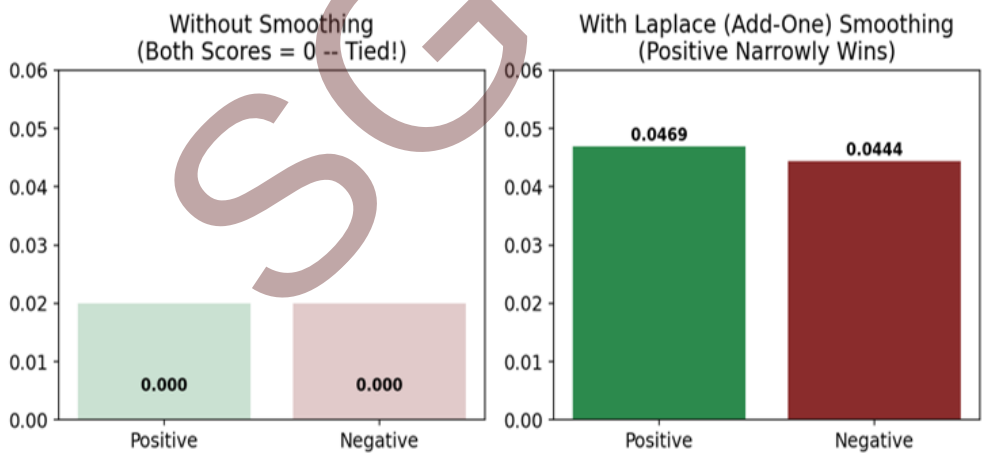


Fig.3.4.2 The Zero-Frequency Problem in Naive Bayes, and the Effect of Smoothing

As Figure 3.4.2 shows, after Laplace smoothing, the review is narrowly classified as Positive (normalized probability ≈ 51.3%, versus ≈ 48.7% for Negative), a far more sensible outcome for a review containing one clearly positive word and one clearly negative word than the uninformative zero-zero tie produced without smoothing. Laplace smoothing is applied routinely, essentially by default, in real-world implementations of the Naive Bayes classifier for exactly this reason.

SGOU

3.4.5 Challenges in Sentiment Analysis

Building on the mixed-sentiment caution first raised in Block II, Unit 3, several further challenges commonly complicate sentiment analysis in practice, for both lexicon-based and machine-learning-based approaches:

- **Negation:** The word 'not' or a similar negator can completely reverse the sentiment of the word that follows it -- 'not helpful' expresses the opposite sentiment of 'helpful', yet a simple lexicon that scores 'helpful' as positive without accounting for the preceding 'not' will score the phrase incorrectly.
- **Sarcasm and Irony:** A phrase such as 'oh great, another dropped call' uses an ordinarily positive word ('great') to express clearly negative sentiment, a pattern that both simple lexicons and many machine-learning classifiers struggle to detect without additional context.
- **Domain-Specific Language:** The same word can carry different sentiment in different contexts -- 'unpredictable' might be negative when describing a company's billing but positive when describing an exciting film plot, meaning a general-purpose lexicon may need to be adapted for a specific business domain such as telecom customer feedback.
- **Mixed Sentiment:** As seen in Sentence B of Section 3.4.3, a single piece of text can express both positive and negative sentiment about different aspects of a product or service simultaneously, and a single overall polarity score may oversimplify genuinely mixed feedback.

None of these challenges makes sentiment analysis unusable. They explain why production systems usually combine several safeguards: special handling of negation words, domain-adapted lexicons, and machine-learning models trained on data from the specific business. They do not rely on the simplest version of either approach introduced in this unit.

It is worth connecting these challenges back to the base-rate reasoning first introduced in Block I, Unit 4. Just as a highly accurate medical test can still produce mostly false positives when the underlying condition is rare, a sentiment classifier that is 'usually' correct can still mislabel a meaningful number of individual reviews, particularly the more linguistically tricky ones involving negation or sarcasm. For this reason, a business rarely relies on one sentiment score for a single review. Sentiment analysis is usually applied in aggregate, across hundreds or thousands of reviews. In a large group, occasional misclassifications tend to average out. The overall trend, whether sentiment is improving or worsening, is then much more reliable than any single review's score.

3.4.6 Forecasting: Simple Exponential Smoothing

Unit 3 of this block introduced three basic forecasting methods: the naive method, the moving-average forecast, and the trend-based forecast. Simple exponential smoothing offers a further refinement, particularly useful when a series shows no strong trend or seasonality: rather than weighting all recent observations equally, as a moving average does, exponential smoothing weights more recent observations more heavily, with the

SGOU

influence of older observations fading away ('decaying') exponentially the further back in time they occurred.

The technique maintains a running smoothed value, updated each period using a smoothing constant, α (alpha), between 0 and 1:

$$S_t = \alpha \cdot Y_t + (1 - \alpha) \cdot S_{t-1}$$

where Y_t is the actual observed value at time t, and S_{t-1} is the previous period's smoothed value. A higher α places more weight on the most recent observation (behaving more like the naive method as α approaches 1), while a lower α smooths more heavily and responds more slowly to recent changes.

Worked Example: Applying simple exponential smoothing, with α = 0.3, to the same 12 months of complaints data used throughout Unit 3 of this block (20, 22, 19, 23, 25, 24, 28, 30, 27, 32, 35, 33), starting with $S_1 = Y_1 = 20$:

Table 3.4.3 Simple Exponential Smoothing of Monthly Complaints (α = 0.3)

Month	1	2	3	4	5	6	7	8	9	10	11	12
Actual (Y)	20	22	19	23	25	24	28	30	27	32	35	33
Smoothed (S)	20.0	20.6	20.1	21.0	22.2	22.7	24.3	26.0	26.3	28.0	30.1	31.0

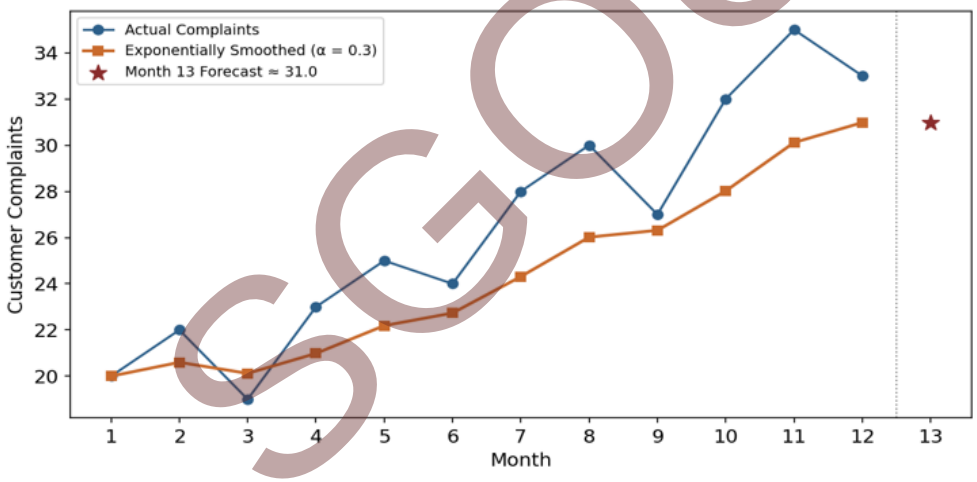


Fig.3.4.3 Simple Exponential Smoothing Forecast for Month 13

Forecast for Month 13 = $S_{12} \approx 30.98$ complaints

As shown in Figure 3.4.3, the smoothed series (S) lags noticeably behind the actual series (Y) whenever Y is rising quickly, since each new smoothed value is a blend of the latest observation and the entire smoothed history built up before it. This lagging behaviour is an important and honest limitation of simple exponential smoothing on a strongly trending series, examined further through actual forecast accuracy in Section 3.4.7.

3.4.7 Evaluating and Comparing Forecast Accuracy

Section 3.4.6 noted that exponential smoothing can lag behind a strongly trending

SGOU

series. Rather than relying on visual impressions alone, the RMSE technique introduced in Block III, Unit 2 can be used to measure this precisely, by comparing each method's one-step-ahead predictions (using only the data available up to that point) against what actually occurred.

For simple exponential smoothing, the prediction for Y_t is simply the previous period's smoothed value, S_{t-1}, giving eleven one-step forecast errors (for months 2 through 12). For the naive method, the prediction for Y_t is simply Y_{t-1}, so its forecast errors are identical to the first differences already calculated in Block III, Unit 3, Section 3.3.6.

$$\text{RMSE (Exponential Smoothing, } \alpha = 0.3) \approx 4.10 \text{ complaints}$$

$$\text{RMSE (Naive Method)} \approx 3.03 \text{ complaints}$$

This is a genuinely useful and slightly humbling result: for this particular, strongly and steadily trending series, the more sophisticated-sounding exponential smoothing method actually produces a larger typical error than the naive method of simply repeating last month's value. This occurs precisely because of the lagging behaviour noted in Section 3.4.6 -- with a low smoothing constant of $\alpha = 0.3$, exponential smoothing responds too slowly to keep pace with a series that is consistently climbing month after month.

This result should not be read as 'exponential smoothing is a poor technique'. It shows a lesson already established in Block III, Unit 2: no forecasting or modelling technique is best for all data. Each method must be checked against the specific data it is applied to, using an objective error metric such as RMSE, and not assumed in advance. A higher smoothing constant (giving more weight to the most recent observation) would track this trending series more closely. More advanced variants, such as double exponential smoothing (Holt's method), add a trend component to deal with this kind of lag. This is beyond the scope of this introductory unit, but it is a natural next step for a learner who continues with time series forecasting.

This same comparison also illustrates why an analyst should never adopt a forecasting method purely because it appears in more advanced textbooks or software packages by default. The telecom company's complaints series happens to have a strong, fairly steady trend, a situation in which the simple naive method performs surprisingly well precisely because last month's value is already a good proxy for a series that keeps moving consistently in one direction. A different series -- one that fluctuates around a broadly stable average with no clear trend -- would very likely reverse this result, with exponential smoothing outperforming the naive method by filtering out noise that the naive method would otherwise repeat directly into its forecast. The only reliable way to know which situation applies to a given dataset is to test multiple candidate methods against held-out data using RMSE, exactly as done in this section, rather than assuming in advance which method 'should' work best.

3.4.8 Combining Text Analytics and Forecasting: Tracking Sentiment Over Time

The final, and arguably most powerful, idea in this unit is that the sentiment-scoring techniques introduced in Sections 3.4.3 and 3.4.4 and the time series techniques

SGOU

introduced in Unit 3 of this block need not be used in isolation from one another. By applying a sentiment-scoring technique to every customer complaint received in a given month, and averaging the resulting scores, an analyst can construct an entirely new time series -- average monthly customer sentiment -- to which every technique introduced in Unit 3 of this block, and this unit, can then be applied directly.

For the telecom churn project, this means the intern could calculate an average lexicon-based or Naive-Bayes-based sentiment score for all customer complaints received each month over the past year, producing a 12-point sentiment time series exactly analogous in structure to the complaints-count series used throughout Unit 3 and this unit. A moving average could smooth out noisy month-to-month swings in sentiment. A linear trend line could show whether sentiment is steadily improving or worsening over the year. Exponential smoothing or a trend-based forecast could project where sentiment is likely to go next month. All of this uses the same formulas already introduced, applied to a new variable derived from text instead of one recorded directly as a number.

This combination closes an important loop that has run through this course. Block II, Unit 3 showed how to turn unstructured text into a numeric representation. This unit has shown how to turn that representation into a single, interpretable sentiment score. Unit 3 of this block, together with the forecasting techniques in this unit, shows how to track and project that score over time. Together, they give a business not just a snapshot of how customers feel today, but a calculated view of where customer sentiment is going next.

A brief practical note is worth adding here: when constructing a sentiment time series in this way, the sample size available in each period matters. A monthly sentiment average calculated from only three or four complaints is considerably less stable, and more easily distorted by a single unusually worded review, than one calculated from several hundred. An analyst combining sentiment analysis with time series forecasting should therefore keep half an eye on how many pieces of text contributed to each period's score, treating months with unusually small volumes of feedback with the same caution recommended for small sample sizes when interpreting a correlation coefficient in Block III, Unit 2.

SGOU

3.4.9 Applications of Text Analytics and Forecasting, and Conclusion of Block III

The techniques introduced in this unit are applied extensively across modern data analytics:

- Brand and Product Monitoring: Companies routinely apply sentiment analysis to social media mentions and product reviews, tracked over time using the forecasting techniques introduced in this block, to catch a worsening reputational trend early, before it escalates.
- Customer Support Quality Management: Sentiment trends calculated from support tickets or call transcripts, exactly as illustrated throughout this unit's

telecom case study, help a support organization detect a developing service quality problem well before it shows up in harder metrics such as churn.

- Financial Market Sentiment: Sentiment analysis applied to news articles and social media is widely used alongside traditional time series forecasting to inform trading and investment strategies.

- Demand and Resource Forecasting: Exponential smoothing and related techniques are used extensively across retail, logistics, and manufacturing to forecast demand and plan resources, often chosen over more complex methods precisely because of their simplicity and computational efficiency at scale.

This unit closes Block III of this course. The intern's telecom churn project, and your own analytical toolkit, have grown through this block. You began with the basic mechanics of machine learning (Unit 1). You then learned the statistical tools to evaluate any model rigorously (Unit 2), how to describe and forecast data ordered in time (Unit 3), and how to score and forecast sentiment from text (this unit). The project has grown from a single classification model into a much more complete picture. It now shows which customers are likely to churn, how confident that prediction should be, whether the underlying problem is getting better or worse over time, and what customers are actually saying about it. The next block of this course turns to how such results are effectively visualized and communicated to the business audiences who ultimately act on them.

Recap

- Sentiment analysis can be approached using a lexicon-based method (summing predefined word scores) or a machine-learning-based method (training a classifier such as Naive Bayes on labelled examples).

- Lexicon-based sentiment analysis sums the scores of every recognized word in a piece of text, correctly capturing mixed sentiment as long as both positive and negative words are represented in the lexicon.

- The zero-frequency problem occurs when a feature value never appears with a given class in the training data, forcing the Naive Bayes score for that class to exactly zero regardless of other evidence; Laplace (add-one) smoothing corrects this by ensuring no probability is ever exactly zero.

- Sentiment analysis faces several persistent challenges, including negation, sarcasm, domain-specific language, and mixed sentiment within a single piece of text.

- Simple exponential smoothing forecasts a time series by weighting recent observations more heavily than older ones, controlled by a smoothing constant α, but can lag behind a strongly and steadily trending series.

SGOU

- Forecasting methods should be compared objectively using RMSE (or other error metrics); a more sophisticated-sounding method, such as exponential smoothing, does not automatically outperform a simpler one, such as the naive method, on every dataset.
- Text analytics and forecasting can be combined by scoring sentiment for text collected over successive time periods, producing a sentiment time series to which moving averages, trend lines, and forecasting methods can then be applied directly.

Objective Type Questions

1. What are the two broad approaches to sentiment analysis introduced in this unit?
2. How is an overall sentiment score calculated in lexicon-based sentiment analysis?
3. What is the zero-frequency problem in Naive Bayes?
4. What technique is commonly used to correct the zero-frequency problem?
5. How does Laplace (add-one) smoothing adjust a probability calculation?
6. What effect does negation have on sentiment analysis?
7. Why is sarcasm difficult for both lexicon-based and machine-learning-based sentiment analysis to detect?
8. What does the smoothing constant, α, control in simple exponential smoothing?
9. What value does simple exponential smoothing predict for the next period?
10. Why can simple exponential smoothing lag behind a strongly trending series?
11. In the worked comparison in this unit, which forecasting method had the lower RMSE on the complaints data -- the naive method or exponential smoothing?
12. What lesson does that RMSE comparison illustrate about choosing a forecasting method?
13. What refinement to exponential smoothing explicitly accounts for a trend, addressing the lagging problem?
14. How can sentiment analysis and time series forecasting be combined?

SGOU

15. What four broad areas has the intern's telecom churn project addressed across the four units of Block III?

Answers

1. Lexicon-based sentiment analysis and machine-learning-based sentiment analysis
2. By summing (or averaging) the predefined scores of every word from the lexicon found in the text
3. When a feature value never appears with a given class in the training data, forcing that class's Naive Bayes score to exactly zero regardless of other evidence
4. Laplace (add-one) smoothing
5. It adds 1 to every count, and adds the number of possible categories for that feature to the denominator, so no probability is ever exactly zero
6. It can reverse the sentiment of the word it modifies (e.g., 'not helpful' versus 'helpful'), which simple methods may fail to account for
7. Because sarcasm uses ordinarily positive words to express negative sentiment, requiring context beyond the individual words themselves to detect correctly
8. How heavily the most recent observation is weighted relative to older observations
9. The most recent smoothed value, S_t
10. Because each smoothed value blends the latest observation with the entire smoothed history, causing it to respond more slowly than the series is actually changing
11. The naive method
12. That no forecasting technique is universally best; suitability must be checked against the specific data using an objective error metric such as RMSE
13. Double exponential smoothing (Holt's method)
14. By scoring sentiment for text collected in each time period, producing a sentiment time series that moving averages, trend lines, and forecasting methods can then be applied to
15. Building and evaluating a classification model (Units 1-2), understanding how the underlying metric changes over time (Unit 3), and scoring and forecasting customer sentiment from text (Unit 4)

SGOU

Assignments

1. Using the small lexicon in Table 3.4.1, plus your own reasonable scores for at least three additional words of your choosing, score the sentiment of a short product review you write yourself (at least two sentences), showing each word's contribution and the final total.
2. A training set of 8 reviews includes 5 labelled Positive and 3 labelled Negative. Among the positive reviews, 3 contain the word 'excellent' and 0 contain the word 'broken'. Among the negative reviews, 0 contain 'excellent' and 2 contain 'broken'. A new review contains both words. Calculate the Naive Bayes scores for Positive and Negative with and without Laplace smoothing, and explain what the zero-frequency problem looks like in this example.
3. Explain, using an example sentence of your own, how negation could cause a simple lexicon-based sentiment score to be calculated incorrectly.
4. Apply simple exponential smoothing, with $\alpha = 0.5$, to the monthly website visits data from Block III, Unit 3, Assignment 1 (50, 55, 52, 58, 60, 57, 63, 65, 62, 68, 70, 67), and calculate the forecast for month 13.
5. Explain, in your own words, why a business might want to combine sentiment analysis with time series forecasting, rather than using either technique on its own, and describe a specific business scenario where this combination would be valuable.

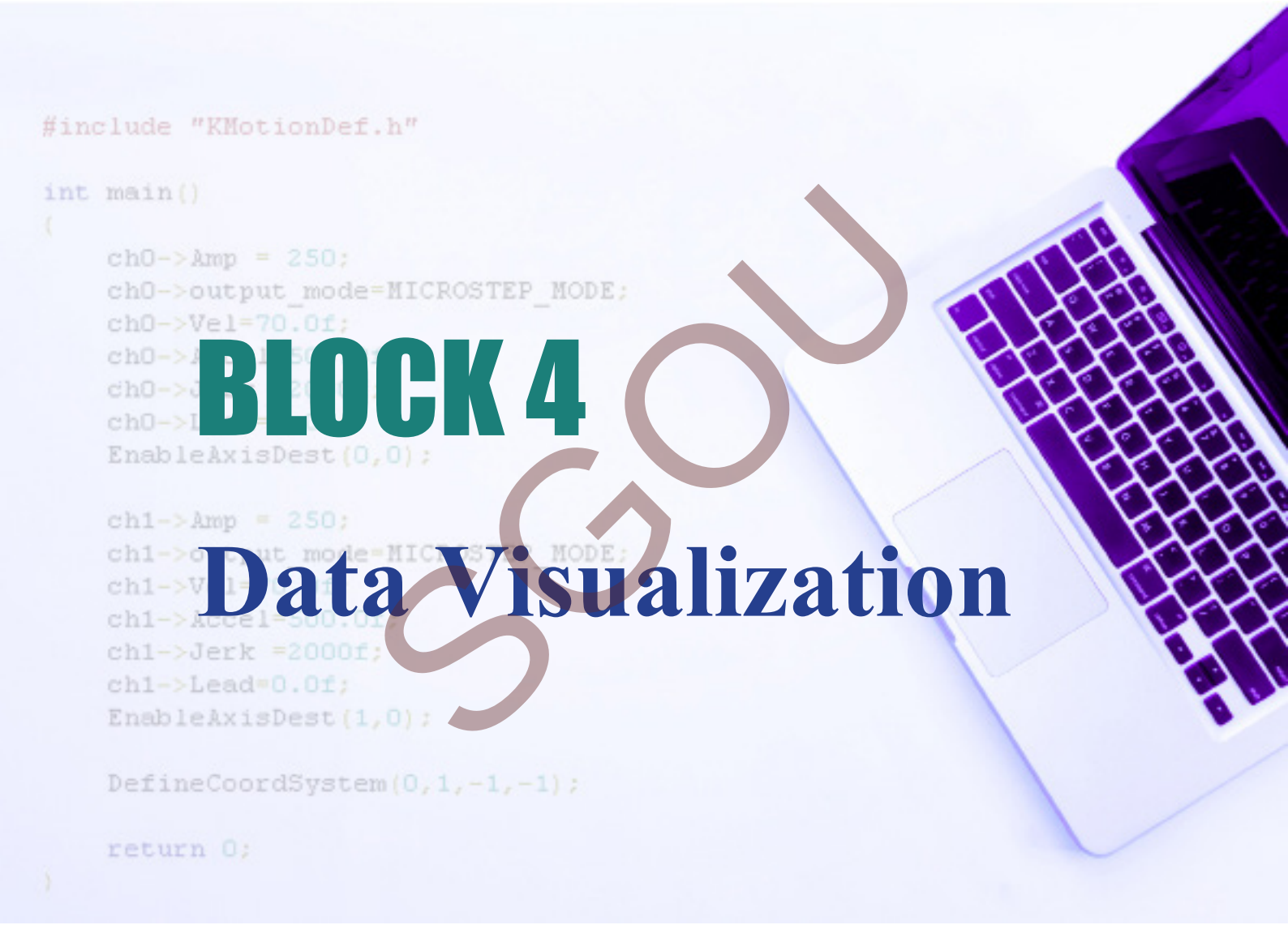
Suggested Reading

1. Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts.
2. Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed., draft). Stanford University.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

Reference

1. Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts.
2. Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed., draft). Stanford University.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

SGOU



BLOCK 4

Data Visualization

SGOU



Data Visualization Basics

Learning Outcomes

After completion of this unit, learner will be able to:

- explain data visualization and its importance in the data analytics process, including how it reveals patterns that summary statistics may hide
- identify common chart types and select appropriate visualizations for different analytical questions
- describe the principles of effective visualization design and explain how to avoid common pitfalls that can make visualizations misleading
- describe the challenges of visualizing real world data and explain the role of visualization in the data analytics life cycl

Prerequisite

Picture the intern opening a blank slide, unsure for the first time how to turn a page of numbers into something her audience will actually understand at a glance. She does not need any new statistics to do this, every number she might want to show (a churn rate, an RMSE, a monthly trend, a sentiment score) has already been calculated somewhere in her project so far. What she needs now is a way of thinking about presentation itself: which chart fits which kind of number, and which design choices help or hurt a viewer's understanding.

This unit assumes only that the learner is comfortable with the basic descriptive statistics, correlation, and time series concepts introduced in Blocks I and III of this course, since several worked examples in this unit revisit those same results purely from a presentation, rather than calculation, point of view. No new mathematical background is introduced.

SGOU

Keywords

Data Visualization, Chart Type, Bar Chart, Line Chart, Scatter Plot, Data-Ink Ratio, Chartjunk, Misleading Visualization, Truncated Axis, Visualization Challenge.

Discussion

4.1.1 Introduction to Data Visualization

Data visualization is the graphical representation of data and information, using visual elements such as charts, graphs, and maps to help people see and understand patterns, trends, and relationships that would be difficult or impossible to grasp from raw numbers alone. Every technique introduced across Blocks I to III of this course; a mean, a correlation coefficient, a decision tree, a monthly trend, a sentiment score and ultimately exists to answer a business question,remove a visualization is very often the final, and most important, step in actually delivering that answer to the people who will act on it.

Visualization is not a late add-on to data analytics, applied only once every other step is finished. As the intern's own project has shown throughout this course, visual tools such as scatter plots (Block III, Unit 2), box plots (Block I, Unit 2), and line charts with trend lines (Block III, Unit 3) were already being used earlier in the process, not just at the end, to build an initial understanding of data during exploratory analysis. This unit and the remainder of Block IV focus specifically on visualization used for the final, communicative purpose of presenting results clearly to an audience, building on the same chart types already encountered informally earlier in this course.

4.1.2 Why Data Visualization Matters

The value of visualization becomes clearest when summary statistics alone are shown to be insufficient on their own. Consider two small datasets, each containing 8 values.

Table 4.1.1 Two Datasets with Identical Summary Statistics

Dataset	Values
Dataset A	10, 12, 14, 16, 18, 20, 22, 24
Dataset B	9, 13, 15, 17, 17, 19, 21, 25

Calculating the mean and standard deviation for each dataset, using exactly the formulas introduced in Block I, Unit 1:

Dataset A: Mean = 17, Population Standard Deviation = 4.583

Dataset B: Mean = 17, Population Standard Deviation = 4.583

SGOU

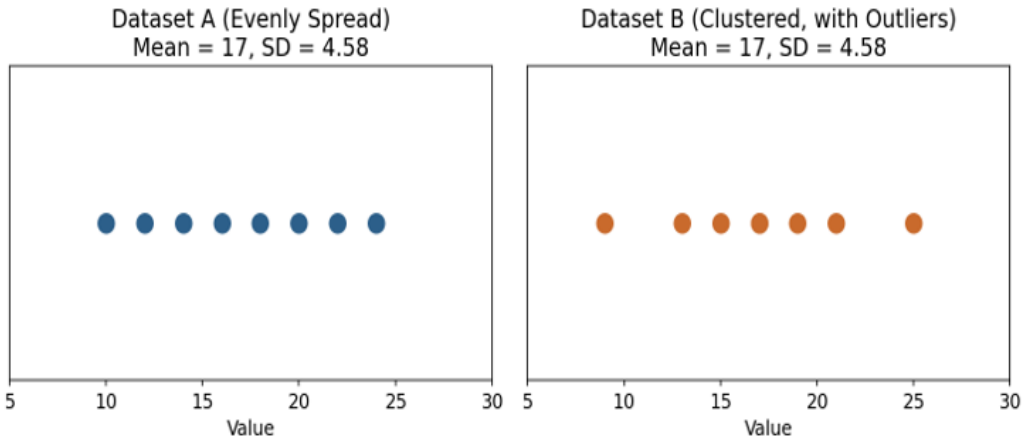


Fig. 4.1.1 Two Datasets with Identical Mean and Standard Deviation, but Very Different Shapes

Both datasets share an identical mean and an identical standard deviation, calculated to three decimal places. Judging only from these two summary numbers, an analyst might reasonably assume the two datasets are essentially the same. Yet, as Fig 4.1.1 shows, their underlying shapes are clearly different: Dataset A is evenly and uniformly spread across its range, while Dataset B is more tightly clustered around the centre, with two more extreme values at either end. No amount of staring at the mean and standard deviation alone would reveal this difference in shape, only plotting the data does. This is precisely why data visualization remains an essential complement to the descriptive statistics introduced in Block I of this course, rather than a mere afterthought once those statistics have been calculated.

This point is not just a classroom example. It echoes a famous demonstration in statistics known as Anscombe's Quartet, published by the statistician Francis Anscombe in 1973. It showed four datasets with nearly identical means, variances and even correlation coefficients, but very different patterns when plotted. One was roughly linear, one was clearly curved, one was dominated by a single outlier, and one had nearly identical x-values with one extreme exception. Anscombe's demonstration, much like the simpler two-dataset example in Table 4.1.1, remains a standard teaching example precisely because it so clearly shows that identical summary numbers can conceal meaningfully different underlying realities, and that visualization is not an optional extra step but a necessary safeguard against being misled by summary statistics alone.

4.1.3 Common Chart Types and What They Reveal

Different chart types are suited to answering different kinds of questions about data. Table 4.1.2 summarizes several chart types already encountered, in various worked examples, earlier in this course.

SGOU

Table 4.1.2 Common Chart Types, Their Purpose, and Earlier Examples from This Course

Chart Type	Best Suited For	Earlier Example in This Course
Bar Chart	Comparing values across discrete categories	Binomial and Poisson PMF bars (Block I, Unit 3)
Line Chart	Showing how a value changes continuously over time	Monthly complaints trend (Block III, Unit 3)
Scatter Plot	Showing the relationship between two numeric variables	Correlation strength examples (Block III, Unit 2)
Box Plot	Showing the spread, median, and outliers of a numeric variable	IQR outlier detection (Block I, Unit 2)
Pie / Stacked Bar Chart	Showing how a whole is divided into parts	Prior versus posterior probability (Block I, Unit 4)
Histogram	Showing the distribution (shape) of a single numeric variable	Normal distribution shape (Block I, Unit 3)

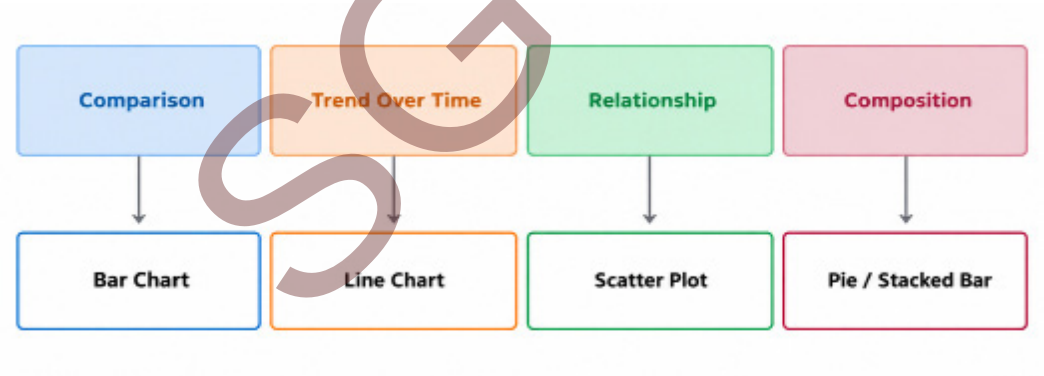


Fig. 4.1.2 Matching a Visualization Goal to a Suitable Chart Type

As Fig. 4.1.2 summarizes, the starting point for choosing a chart type should always be the specific question being asked of the data, rather than personal preference or habit. Asking 'am I comparing categories, showing change over time, showing a relationship, or showing how a whole breaks into parts?' before selecting a chart type helps ensure the resulting visualization actually supports the message it is meant to convey.

It is worth noting that these categories are starting guidelines rather than rigid rules. A single dataset can often be visualized correctly using more than one chart type, and the right choice can depend as much on the audience and the specific message being emphasized as on the data itself. The monthly complaints data used throughout Block

SGOU

III, Unit 3, for instance, was shown as a line chart to emphasize the trend over time, but the same twelve monthly values could equally well be shown as a bar chart if the intention were instead to let a viewer compare a handful of individual months directly against one another. Choosing between reasonable alternatives such as these is a matter of judgement, guided by the question in Fig 4.1.2, rather than a single mechanical rule that applies to every situation.

4.1.4 Principles of Effective Visualization

Beyond choosing the right chart type, several general design principles help ensure a visualization communicates clearly:

- **Maximize the Data-Ink Ratio:** A principle popularized by visualization theorist Edward Tufte, stating that as much of a chart's visual 'ink' as possible should be devoted to representing actual data, with decorative elements that convey no information such a unnecessary gridlines, background images, or heavy borders, kept to an absolute minimum.

- **Avoid Chartjunk:** Chartjunk refers to unnecessary visual embellishments, such as 3D effects on a simple bar chart, overly decorative fonts, or distracting background textures, which add visual clutter without adding any informational value, and often actively distort the reader's perception of the underlying data.

- **Use Color Purposefully:** Color should be used to highlight meaningful distinctions (for example, a different color for the single most important bar in a chart) rather than applied decoratively; overusing many bright, unrelated colors can overwhelm a viewer and obscure the actual message.

- **Label Directly and Clearly:** Wherever possible, labelling data points, bars, or lines directly on the chart itself, rather than relying solely on a separate legend the viewer must repeatedly refer back to, reduces the cognitive effort required to read a chart correctly.

- **Maintain Consistency:** Using consistent colors, scales, and formatting across a set of related charts (for example, in a single report or dashboard) helps a viewer transfer their understanding from one chart to the next without needing to re-learn the visual conventions each time.

These principles echo a theme already familiar from earlier in this course: just as the CW checklist for this course's own learning material calls for content that is clear, free of unnecessary clutter, and focused on what actually aids understanding, effective visualization design applies exactly the same standard to charts and graphs.

A further principle, closely related to maximizing the data-ink ratio, is choosing an appropriate level of precision for the audience. A chart for a technical audience of fellow analysts might show a correlation coefficient to three decimal places, as calculated in Block III, Unit 2. For a general business audience, it is usually better to round to one decimal place, or simply say 'a strong positive relationship' in plain language. Extra decimal places rarely change a business decision, but they can make a chart look harder to read at a glance. Matching the precision shown on a chart to what the audience

SGOU

actually needs is itself a form of maximizing the data-ink ratio: every digit displayed should earn its place by genuinely aiding the viewer's understanding.

4.1.5 Common Pitfalls and Misleading Visualizations

Just as a poorly designed chart can obscure a message through clutter, a carelessly (or deliberately) designed chart can actively mislead a viewer about what the underlying data actually shows. Common pitfalls include:

- **Truncated Axes:** Starting a bar chart's vertical axis at a value other than zero can dramatically exaggerate small differences between categories, making a minor difference appear far larger than it really is.
- **Inconsistent Scales:** Comparing two charts side by side that use different scales on their axes can make two genuinely similar patterns appear very different, or two genuinely different patterns appear deceptively similar.
- **Cherry-Picked Time Ranges:** Choosing a start and end date for a trend chart that happens to show the most favourable (or unfavourable) possible picture, while excluding data just outside that window that would tell a different story.
- **Overuse of 3D Effects:** Adding an unnecessary third dimension to a simple bar or pie chart distorts the visual proportions of each segment, making accurate comparison between categories more difficult rather than easier.

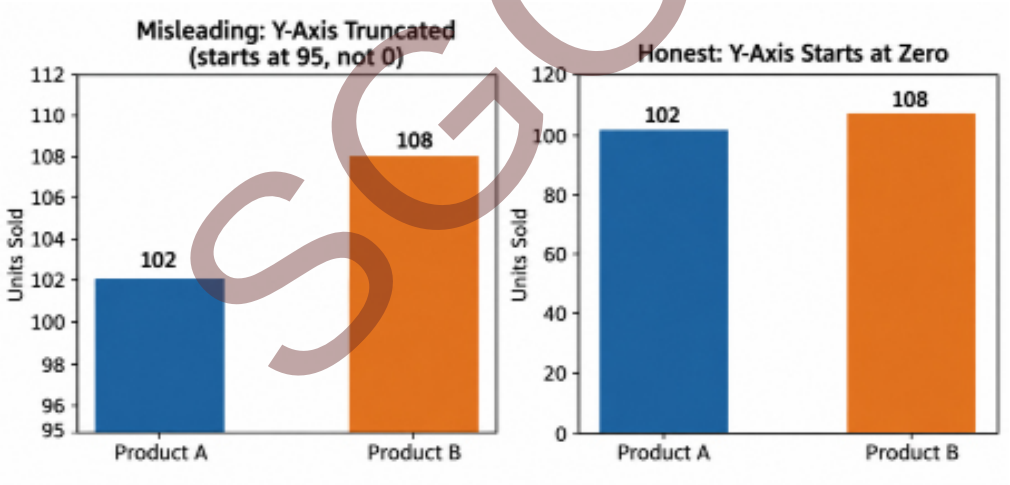


Fig. 4.1.3 The Same Data, Shown Honestly and Misleadingly

Figure 4.1.3 uses the same two figures (102 and 108 units sold) in both panels. In the truncated panel, the y-axis starts at 95 instead of 0. This makes Product B's sales look much higher than Product A's, suggesting a difference far larger than the true gap of 5.9%. The honest, zero-based axis on the right correctly shows the two bars as only slightly different in height. A responsible analyst must be alert to these pitfalls both when building their own visualizations and when critically reading visualizations produced by others, since a technically accurate chart can still convey a materially misleading impression through choices such as these.

SGOU

It is worth being fair to the truncated-axis technique, since it is not always used with an intention to mislead. Truncating an axis can occasionally be a reasonable choice when a genuinely small percentage change is itself the important story. For example, a central bank's interest rate moving from 6.00% to 6.25% is a meaningful policy change worth showing clearly, even though it would appear as a barely visible sliver on a chart with a full zero-based axis. The responsible practice in such cases is not to avoid truncation altogether, but to label the axis clearly, state the starting value explicitly, and, where practical, disclose the truncation directly in the chart's title or a footnote, so that a viewer is never left to assume, incorrectly, that a chart's axis begins at zero when it does not.

4.1.6 Challenges in Data Visualization

Beyond the risk of unintentionally (or intentionally) misleading a viewer, several further challenges commonly complicate data visualization in real-world practice, particularly as the scale and complexity of data grows in the ways described throughout Block II of this course:

- **High Volume:** Visualizing millions of individual data points directly, as might be generated by the big data sources introduced in Block II, Unit 1, can overwhelm both the rendering capability of visualization software and the perceptual capacity of a human viewer, typically requiring the data to be aggregated or sampled first.
- **High Dimensionality:** Data with many variables (dimensions) cannot be directly plotted on a two-dimensional chart; visualizing more than two or three variables at once requires techniques such as color, size, faceting (small multiples), or dimensionality reduction, each with its own limitations.
- **Diverse and Non-Technical Audiences:** A chart built for a fellow data analyst, comfortable with statistical terminology, may be entirely unsuitable for a business executive or a member of the public, requiring the same underlying data to be presented differently depending on who will be viewing it.
- **Real-Time and Streaming Data:** Continuously updating data, such as live website traffic or sensor readings, requires visualizations that can refresh smoothly and remain readable even as the underlying data changes from moment to moment, a requirement returned to in Unit 3 of this block on interactive dashboards.
- **Accessibility:** A visualization relying solely on color to distinguish categories may be difficult or impossible to interpret for viewers with color vision deficiencies (colour blindness), making the choice of color palette and the use of additional visual cues (such as patterns or direct labels) an important accessibility consideration.

These challenges are precisely why data visualization is treated as a substantial subject in its own right, worthy of the dedicated block that follows, rather than as a simple final step of 'making a chart' tacked onto the end of an analytics project.

SGOU

Several of these challenges will be addressed directly in the remaining units of this block. High volume and high dimensionality are revisited in Unit 2, especially when geospatial data is layered with several other variables. Real-time data and diverse audiences are addressed in Unit 3, through interactive dashboards that let different viewers explore the same data at their own pace and by their own interests. The overall discipline of managing these challenges step by step, from the first visualization goal to a finished, tested result, is formalized in the data visualization cycle in Unit 4. This unit's role has been to name these challenges clearly; the remainder of Block IV exists to address them in practice.

4.1.7 The Role of Visualization in the Data Analytics Life Cycle

Recall the six-phase data analytics life cycle first introduced in Block II, Unit 2: Discovery, Data Preparation, Model Planning, Model Building, Communicate Results, and Operationalize. Visualization plays an important role in at least two of these phases, serving genuinely different purposes in each.

- **Exploratory Visualization (during Data Preparation and Model Planning):** Charts such as scatter plots and box plots are used privately by the analyst to understand the data, spot anomalies and check assumptions before or during modelling. Box plots were used in this way to detect outliers in Block I, Unit 2, and scatter plots were used to assess correlation in Block III, Unit 2.

- **Explanatory Visualization (during Communicate Results):** Charts are designed deliberately for an external audience, to clearly and persuasively convey a specific finding or recommendation, the primary focus of this unit and the remainder of Block IV.

The design principles differ meaningfully between these two purposes. An exploratory chart, built quickly by an analyst for their own use, can reasonably prioritize speed over polish, since the only audience is the analyst themselves, already familiar with the data's context. An explanatory chart, intended for the intern's leadership meeting or a similar audience, must instead prioritize clarity, honesty, and an unambiguous message for viewers who may see the underlying data for the first time exactly the standard the principles introduced in Section 4.1.4 and the pitfalls introduced in Section 4.1.5 are designed to uphold.

Table 4.1.3 summarizes this distinction directly, drawing together the contrast introduced across this section.

Table 4.1.3 Comparing Exploratory and Explanatory Visualization

Aspect	Exploratory Visualization	Explanatory Visualization
Primary Audience	The analyst themselves	Business stakeholders or the public

SGOU

Typical Life Cycle Phase	Data Preparation, Model Planning	Communicate Results
Priority	Speed, flexibility, quick iteration	Clarity, honesty, a single clear message
Typical Polish Level	Minimal (default software formatting is often sufficient)	High (labelled, titled, carefully designed for the specific audience)

It is a useful habit to recognize which of these two modes a chart belongs to before building it. A chart built quickly for exploration should not be copied straight into a final report or presentation without first being checked against the explanatory standard. Equally, spending a lot of time polishing a chart that is only for the analyst's own exploration is usually time better spent elsewhere in the project.

4.1.8 Applications of Data Visualization in Practice

The principles introduced in this unit underlie visualization practice across every industry that uses data analytics:

- **Executive Reporting:** Dashboards and summary charts translate detailed analytical results, such as the churn model results developed throughout this course, into a form that business leaders can quickly understand and act on.
- **Journalism and Public Communication:** News organizations rely heavily on data visualization to communicate complex statistics, such as election results, public health data, economic indicators, to a broad, non-technical public audience.
- **Scientific Communication:** Researchers use carefully designed charts to communicate experimental findings clearly to peers, reviewers, and the public, with the same honesty principles introduced in Section 4.1.5 applying with particular force in this context.
- **Product and UX Design:** Effective visualization principles inform the design of everyday consumer applications, from fitness trackers displaying activity trends to financial apps displaying spending patterns.

Now that you have learned why visualization matters, how to choose a suitable chart type and how to avoid common design pitfalls, you are ready for Unit 2 of this block. That unit extends these foundations to two more kinds of data: geospatial (location-based) data, and mixed numerical and non-numerical (categorical) data. Both need visualization techniques beyond the standard chart types in this unit.

Bringing this unit to a close, we return one final time to the intern. Using the chart-selection guidance, design principles and pitfalls from this unit, she can now look back over everything her project produced in Blocks I to III and make a deliberate, defensible choice for each result she wants to present. For the monthly complaints data from Block III, Unit 3, she chooses a line chart with a trend line. For her model's accuracy against

SGOU

a naive baseline from Block III, Unit 2, she chooses a simple, clearly labelled bar chart. For her sentiment findings from Block III, Unit 4, she writes a plainly worded summary, not a chart cluttered with technical detail. Every one of these choices follows directly from asking the same question introduced in Fig.4.1.2, what is this chart actually trying to show, and to whom?

Recap

- Data visualization is the graphical representation of data, used both to explore data privately during analysis and to communicate findings clearly to an external audience.
- Two datasets can share identical summary statistics, such as mean and standard deviation, while having very different underlying shapes -- a difference that only visualization, not summary numbers alone, can reveal.
- Different chart types suit different purposes: bar charts for comparison, line charts for trends over time, scatter plots for relationships, box plots for spread and outliers, and pie or stacked bar charts for composition.
- Effective visualization design maximizes the data-ink ratio, avoids chartjunk, uses color purposefully, labels directly, and maintains consistency across related charts.
- Common pitfalls that can make a visualization misleading include truncated axes, inconsistent scales, cherry-picked time ranges, and unnecessary 3D effects.
- Real-world data visualization faces further challenges, including high data volume, high dimensionality, diverse audiences, real-time data, and accessibility for viewers with color vision deficiencies.
- Visualization serves two distinct purposes within the data analytics life cycle: exploratory visualization, used privately by an analyst during Data Preparation and Model Planning, and explanatory visualization, designed for an external audience during Communicate Results.

Objective Type Questions

1. What is data visualization?
2. What did Datasets A and B in this unit's worked example have in common, and what was different about them?
3. Which chart type is best suited to comparing values across discrete categories?
4. Which chart type is best suited to showing a relationship between two numeric variables?

SGOU

5. What does the data-ink ratio principle recommend?
6. What is chartjunk?
7. What is the effect of truncating a bar chart's y-axis so that it does not start at zero?
8. What is a cherry-picked time range in the context of misleading visualization?
9. Why can overusing 3D effects on a chart be misleading?
10. What visualization challenge arises when a dataset has more than two or three variables?
11. Why might a chart accessible to most viewers still be inaccessible to a viewer with color vision deficiency?
12. What is the difference between exploratory visualization and explanatory visualization?
13. During which phase of the data analytics life cycle is explanatory visualization primarily used?
14. During which phases of the data analytics life cycle is exploratory visualization primarily used?
15. Why does an explanatory chart typically require more care and polish than an exploratory chart?

Answers

1. The graphical representation of data and information using visual elements such as charts, graphs, and maps
2. They had an identical mean and standard deviation, but different underlying shapes (evenly spread versus clustered with outliers)
3. Bar chart
4. Scatter plot
5. That as much of a chart's visual ink as possible should represent actual data, minimizing decorative, non-informative elements
6. Unnecessary visual embellishments, such as 3D effects or decorative textures, that add clutter without adding informational value
7. It can exaggerate small differences between categories, making them appear far larger than they really are

SGOU

8. Choosing a start and end date for a chart that shows the most favourable or unfavourable possible picture, excluding data that would tell a different story

9. Because 3D effects distort the visual proportions of chart segments, making accurate comparison between categories more difficult

10. High dimensionality, since more than two or three variables cannot be directly plotted on a standard two-dimensional chart

11. Because a chart relying solely on color to distinguish categories may not be distinguishable to a viewer with color vision deficiency

12. Exploratory visualization is used privately by an analyst to understand data; explanatory visualization is designed for an external audience to communicate a finding

13. Communicate Results

14. Data Preparation and Model Planning

15. Because its audience may be seeing the underlying data for the first time and needs an unambiguous, honest message, unlike an analyst already familiar with the data's context

Assignments

1. Find or construct two small datasets of your own (at least 6 values each) that share the same mean and standard deviation but have visibly different shapes when plotted. Show your calculations and describe the difference in shape.

2. For each of the following business questions, state which chart type from Table 4.1.2 would be most appropriate, and justify your answer: (a) comparing total sales across five regions; (b) showing how a company's stock price has moved over the past year; (c) showing the relationship between advertising spend and sales.

3. Find (or describe from memory) an example of a misleading chart you have seen in the media or online, identify which pitfall from Section 4.1.5 it demonstrates, and redraw or redescribe how it could be presented more honestly.

4. Explain, using an example of your own, why a dashboard intended for company executives might need to look very different from a chart an analyst builds purely for their own exploratory use.

5. Describe a real-world visualization challenge (from Section 4.1.6) that you think would be most difficult to solve for a live, streaming dataset such as real-time website traffic, and suggest one possible way to address it.

SGOU

Suggested Reading

1. Tufte, E. R. (2001). The Visual Display of Quantitative Information (2nd ed.). Graphics Press.
2. Few, S. (2012). Show Me the Numbers: Designing Tables and Graphs to Enlighten (2nd ed.). Analytics Press.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

Reference

1. Tufte, E. R. (2001). The Visual Display of Quantitative Information (2nd ed.). Graphics Press.
2. Few, S. (2012). Show Me the Numbers: Designing Tables and Graphs to Enlighten (2nd ed.). Analytics Press.
3. Ahmed, M., & Pathan, A. K. (2020). Data Analytics: Concepts, Techniques and Applications (1st ed.). Taylor and Francis.

SGOU



UNIT 2

Visualization of Geospatial and Multitype Data

Learning Outcomes

After completion of this unit, learner will be able to:

- distinguish point, line, and polygon geospatial data
- describe and interpret common geospatial visualization techniques, including choropleth maps, point maps, and heat maps
- describe techniques for visualizing numerical data, non-numerical (categorical) data, and combinations of both together
- construct a multitype visualization that encodes several variables of different types within a single chart
- match an appropriate visual channel (position, color, size, shape) to a given type of data

SGOU

Prerequisite

Picture the intern with two files open: a spreadsheet of churn rates broken down by the telecom company's five operating regions, and another table combining contract type, average monthly charge, and churn rate together for the same set of customers. Neither file fits comfortably into the bar charts, line charts, and scatter plots introduced in Unit 1 -- one needs a sense of physical location, and the other needs to show three different kinds of information in one place. This unit gives her the tools for both.

This unit builds directly on the chart types and design principles introduced in Unit 1 of this block, and assumes no prior background in geographic information systems (GIS) or mapping software.

Here is a quick self-check before you begin. If you can look at a spreadsheet and say at once whether a column holds a number, a category or a location, you already have all you need to start this unit. The rest is learning which chart types exist for each of those cases and, more importantly, for combinations of them.

Keywords

Geospatial Data, Point Data, Line Data, Polygon Data, Choropleth Map, Point Map, Heat Map, Multitype Data, Bubble Chart, Visual Encoding, Preattentive Attribute.

Discussion

4.2.1 Introduction to Geospatial and Multitype Data Visualization

Unit 1 of this block introduced chart types suited to a single numeric variable, a relationship between two numeric variables, or a comparison across categories. Two further situations arise often enough in real-world data analytics to deserve dedicated attention: data tied to a physical location (geospatial data), and data that combines several variables of different types (numeric and categorical) within a single result (multitype data).

Both situations are direct extensions of ideas already introduced earlier in this course rather than entirely new subjects. Geospatial visualization is, at its core, still about representing a numeric or categorical value exactly as a bar chart does except that the value is now attached to a specific place on a map rather than to a labelled category on an axis. Multitype visualization, similarly, is about combining several of the encodings already introduced (position, color, size) within one chart, rather than choosing only one at a time.

It is worth being upfront about the scope of this unit. Building genuinely interactive, production-quality maps typically requires specialized geographic information system (GIS) software or dedicated mapping libraries capable of working with real geographic boundary files, well beyond what a general-purpose charting tool can produce directly. This unit focuses instead on the visual thinking behind maps: what a choropleth map is trying to achieve, and why. It uses simplified, clearly labelled schematics instead of production-grade maps. This way, the principles carry over to whichever mapping or business intelligence tool you eventually use, including Power BI and Tableau, which are covered in dedicated blocks later in this course.

4.2.2 Types of Geospatial Data

Geospatial data is data associated with a specific physical location which is typically classified into three basic types:

- **Point Data:** Represents a single, specific location, such as the coordinates of an individual retail store, a customer's billing address, or a cell tower.
- **Line Data:** Represents a path or connection between locations, such as a delivery route, a fibre-optic cable route, or the path of a customer's movement between cell towers.

SGOU

- **Polygon Data:** Represents an area or boundary, such as a state, a telecom operating circle, or a postal code region.

Table 4.2.1 The Three Basic Types of Geospatial Data

Type	Represents	Telecom Example
Point	A single specific location	Location of an individual retail outlet
Line	A path or connection	A planned fibre-optic cable route between cities
Polygon	An area or boundary	A telecom operating circle or district boundary

Identifying which of these three types a given dataset represents is the first step in choosing an appropriate geospatial visualization technique, several of which are introduced in the next section.

A single real-world dataset can sometimes be represented using more than one of these three types, depending on the level of detail required. A telecom company's retail outlets, for instance, might be represented as point data when the goal is to show their exact individual locations on a map, but the sales performance of an entire sales region containing many such outlets might instead be represented as polygon data, aggregating many individual points up into a single regional boundary and value. Choosing the right level such as individual points, connecting lines, or aggregated regions depends on the specific question being asked, echoing the same 'what question am I trying to answer' guidance introduced for chart-type selection in Unit 1 of this block.

4.2.3 Common Geospatial Visualization Techniques

Several standard techniques exist for visualizing geospatial data, each suited to a different combination of data type and message, with a fuller comparison, including the important rate-versus-count design choice, developed further in Section 4.2.4.

- **Choropleth Map:** Shades entire polygon regions (such as states or operating circles) using a color scale to represent a numeric value associated with each region for example, shading each telecom circle from light to dark according to its churn rate.
- **Point (Pin) Map:** Places a marker at each specific point location, useful for showing where individual events or entities (such as retail outlets, customer complaints, or service outages) are physically located.
- **Heat Map (Geospatial):** Uses color intensity to show the density or concentration of point data across an area, useful for quickly identifying where activity or events cluster most heavily, without needing to plot every single individual point marker.

SGOU

- **Flow Map:** Uses lines of varying thickness between locations to represent the volume or direction of movement or connection between them, useful for visualizing line data such as delivery routes or migration patterns.

A choropleth map is the most directly comparable to the bar and column charts introduced in Unit 1: both encode a numeric value using color or length, the only difference being that a choropleth map arranges its categories (regions) according to their real geographic position, rather than in an arbitrary row or column.

An important, easily overlooked design decision for a choropleth map is whether to shade regions by a raw count or by a rate. A large, densely populated region will naturally tend to have a larger raw count of almost anything more total complaints, more total churned customers, more total sales, simply because it contains more customers overall, regardless of whether that region is actually performing better or worse on a per-customer basis. Shading by a rate (such as churn rate, complaints per thousand customers, or sales per capita) rather than a raw count therefore usually produces a fairer and more genuinely informative choropleth map, correcting for the underlying difference in regional size that a raw-count map would otherwise conflate with genuine regional performance.

4.2.4 A Worked Example: Regional Churn Rates as a Choropleth-Style Schematic

Consider a telecom company operating across five regions, each with a different observed churn rate over the past quarter:

Table 4.2.2 Quarterly Churn Rate by Operating Region

Region	Churn Rate	Region	Churn Rate
North	8%	East	22%
West	12%	South	15%
Central	18%		

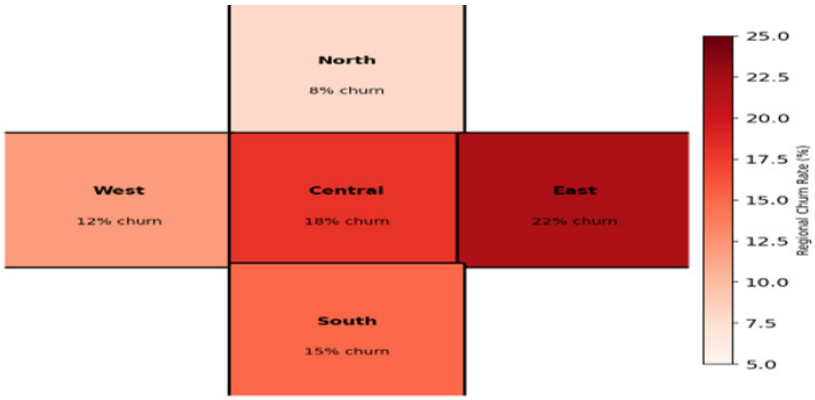


Fig 4.2.1 Illustrative Choropleth-Style Schematic
(Stylized Regions, Not an Actual Geographic Map)

SGOU

As Fig 4.2.1 illustrates using a simplified, schematic layout rather than an actual geographic map, darker shading immediately draws the eye to the East region, with the highest churn rate at 22%, while the lightest shading on the North region (8% churn) is immediately visible as the company's best-performing area. This is precisely the value a choropleth map adds over a plain table such as Table 4.2.2: a viewer can identify the best- and worst-performing regions in a fraction of a second, without needing to read and compare five individual numbers. In a real business setting, this same technique would be applied to an actual map of the telecom company's operating circles, using genuine geographic boundaries rather than the simplified schematic used here for illustration, typically built using dedicated mapping or business intelligence software rather than general-purpose charting tools.

This churn-rate choropleth also illustrates why geospatial visualization is rarely the final step of an analysis on its own, but rather a starting point for further investigation. Seeing that the East region has an unusually high churn rate raises a natural next question: why? The map cannot answer this, but it points the intern back to techniques from earlier in this course. She can check whether East's customer base differs in contract type (Block III, Unit 1). She can check whether East's complaint sentiment has been getting worse over time (Block III, Units 3 and 4). She can also check whether some other regional factor, not yet examined, is a confounding variable behind the pattern (Block III, Unit 2). A well-designed map, in this sense, functions less as a final answer and more as an effective way of directing attention toward the right follow-up questions.

4.2.5 Visualizing Numerical and Non-Numerical Data

Unit 1 of this block already introduced several techniques for visualizing a single numeric variable (histograms, box plots) and for comparing numeric values across categories (bar charts). Non-numerical (categorical or text-based) data calls for a few further, complementary techniques:

- **Bar and Column Charts:** Remain the standard, most reliable choice for comparing counts or proportions across categories, such as the number of customers on each contract type.
- **Treemaps:** Represent categories as nested rectangles, with each rectangle's area proportional to a numeric value, useful for showing both the relative size of many categories and any natural hierarchy or grouping among them (for example, product categories nested within product families).
- **Word Clouds (Word-Frequency Displays):** Represent the words appearing in a body of text, with each word's visual size scaled according to how frequently it occurs, providing an immediate, at-a-glance impression of the most prominent themes in unstructured text.

SGOU

Table 4.2.3 Comparing Techniques for Visualizing Non-Numerical (Categorical and Text) Data

Technique	Best Suited For	Limitation
Bar / Column Chart	Precise comparison of counts or proportions across a moderate number of categories	Becomes cluttered with a very large number of categories
Treemap	Showing relative size and hierarchy across many categories in limited space	Precise comparison between similarly sized rectangles can be difficult
Word Cloud	An immediate, at-a-glance impression of the most prominent terms in text	Poorly suited to precise comparison; size is a less exact visual channel

Worked Example: Recall the bag-of-words word frequencies calculated from five customer complaints in Block II, Unit 3: 'bill' (5 occurrences), 'support' (4), 'fix' (2), 'unhappy' (1), and 'late' (1).



Fig. 4.2.2 Word Size Scaled by Frequency (Bag-of-Words Counts from Block II, Unit 3)

As Fig. 4.2.2 shows, scaling each word's displayed size according to its frequency immediately draws attention to 'bill' and 'support' as the two dominant themes among these complaints, without requiring a viewer to read and compare the underlying frequency table first. This is precisely the kind of non-numerical (text-based, categorical) data visualization that a bar chart of word counts could also convey, but a word-size display often communicates the same relative ranking more immediately and compactly, particularly when many different words are involved.

However, word clouds have a real limitation alongside their visual appeal. They show frequency mainly through size, which Section 4.2.7 notes is less precise than position or length. A word cloud is therefore good at giving an approximate impression of which few words dominate. It is poor at supporting an accurate, fine-grained comparison between, say, a word appearing 12 times and another appearing 9 times. For a presentation where only the broad, most prominent themes matter, as in the intern's leadership meeting, this trade-off is usually acceptable. For a more detailed analytical report where the exact ranking and count of many words matter, a sorted bar chart of word frequencies is the more reliable choice, in the spirit of Fig 2.3.2 in Block II, Unit 3.

SGOU

4.2.6 Visualizing Multitype Data Together

Many of the most useful business results combine more than one type of data at once, for example, a categorical variable (such as contract type) alongside two or more numeric variables (such as average monthly charge and churn rate). Rather than building three separate charts, a single, carefully designed multitype visualization can often convey all of this information together, provided each variable is mapped to a distinct and appropriate visual channel.

Worked Example: Consider three contract types, each described by three variables: average monthly charge (numeric), churn rate (numeric), and number of customers (numeric, but naturally displayed as a size rather than a position).

Table 4.2.4 Contract Type, Monthly Charge, Churn Rate, and Customer Count

Contract Type	Avg. Monthly Charge	Churn Rate	Number of Customers
Month-to-Month	₹75	42%	400
One-Year	₹65	15%	250
Two-Year	₹60	5%	180

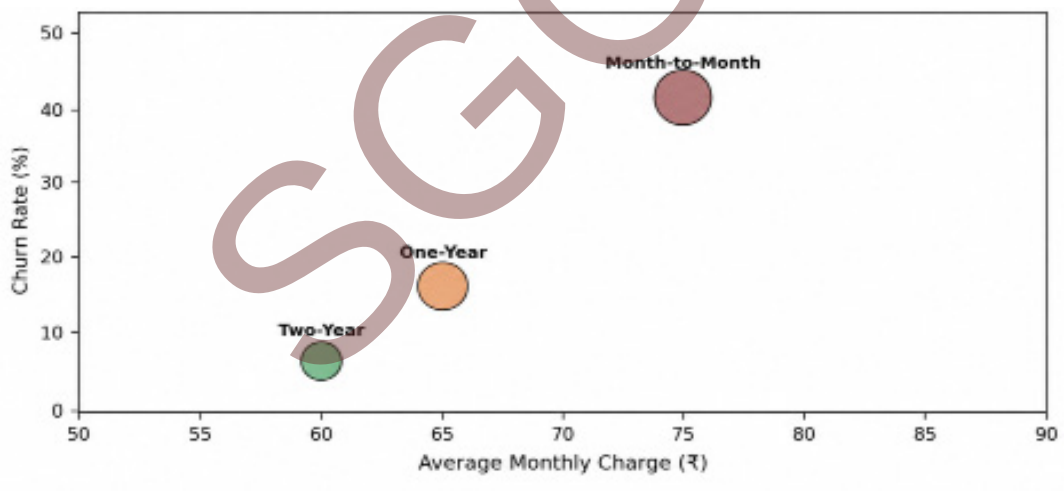


Fig. 4.2.3 Multitype Visualization - Contract Type (Color), Monthly Charge (X), Churn Rate (Y), and Customer Count (Bubble Size)

As Figure 4.2.3 shows, four distinct variables are encoded simultaneously within a single chart: horizontal position represents average monthly charge, vertical position represents churn rate, bubble size represents the number of customers, and color represents contract type (a categorical, non-numeric variable). A viewer can immediately see that month-to-month customers shown in red, with the highest charges, the highest churn, and the largest customer base represent the single greatest area of retention risk for the business, a conclusion that would otherwise require cross-referencing

SGOU

three separate charts or a full table of numbers to reach. This directly echoes, and now visually confirms, the decision-tree finding from Block III, Unit 1 that contract type was an unusually powerful predictor of churn.

It is worth noting how deliberately restrained this chart is, despite encoding four variables at once. Only three data points (one per contract type) are shown, each clearly labelled directly on the chart rather than through a separate legend the viewer must cross-reference both choices consistent with the data-ink and direct-labelling principles introduced in Unit 1 of this block. A tempting but poor alternative would have been to plot every individual customer as a separate bubble, all 830 of them, which would have produced a cluttered, overlapping mass of circles far harder to read than the three clean, aggregated bubbles shown in Fig 4.2.3. This shows a general principle to carry into the remaining units of this block. Adding more variables or detail to a chart is useful only when it helps the viewer understand. Aggregating data sensibly before charting it, as done here by contract type, is often more effective than showing every individual data point.

4.2.7 Choosing Appropriate Visual Encodings for Each Data Type

The multitype example in Section 4.2.6 succeeded because each variable was mapped to a visual channel well suited to the kind of comparison a viewer needs to make with it. Not every visual channel is equally effective for every purpose; research in visualization design has generally found position and length to be the most accurately and quickly judged by the human eye, followed by color and size, both of which are useful but noticeably less precise for comparing exact values.

Table 4.2.5 Matching Visual Channels to Their Best Use

Visual Channel	Best Suited For	Example from This Unit
Position (X/Y Axis)	Precise comparison of numeric values	Monthly charge and churn rate in Figure 4.2.3
Length / Height	Precise comparison of numeric values across categories	Standard bar charts (Unit 1 of this block)
Color	Distinguishing categories, or showing an approximate numeric intensity	Contract type and regional churn rate (Figures 4.2.1, 4.2.3)
Size	Showing an approximate relative magnitude, often a third or fourth variable	Customer count in Figure 4.2.3

This ranking reflects what visualization researchers call preattentive attributes visual properties, including position, color, and size, that the human visual system processes automatically and almost instantly, before conscious, deliberate attention is applied. Position and length are judged with the greatest precision. This is why they are kept, wherever possible, for the variable a viewer most needs to compare accurately. Color and size suit secondary variables, where an approximate impression is enough instead

SGOU

of an exact numeric comparison. This is the reasoning that put contract type on the color channel and customer count on the size channel, rather than the reverse, in Figure 4.2.3.

A further preattentive attribute worth briefly noting is shape, using different marker shapes (circles, squares, triangles) to distinguish categories, in much the same way color does. Shape is particularly useful as a supplementary channel alongside color, since relying on shape as well as color to distinguish categories helps a chart remain interpretable for viewers with color vision deficiencies, directly addressing the accessibility challenge raised in Unit 1 of this block. A multitype chart such as Figure 4.2.3 could, for instance, use both a distinct color and a distinct marker shape for each contract type, ensuring the categories remain distinguishable even for a viewer who cannot reliably tell the colors apart.

4.2.8 Applications of Geospatial and Multitype Visualization

The techniques introduced in this unit are applied extensively across data analytics practice:

- **Retail and Logistics:** Choropleth maps and point maps are used to plan store locations, delivery routes, and regional marketing campaigns based on geographic patterns in sales and demand.
- **Public Health:** Choropleth maps showing disease incidence by region are a standard tool for identifying outbreak hotspots and directing public health resources efficiently.
- **Telecommunications:** Network coverage, signal strength, and regional churn patterns, exactly as illustrated in this unit's worked example, are routinely visualized geographically to guide infrastructure investment and retention campaigns.
- **Marketing and Customer Analytics:** Multitype visualizations combining customer segments (categorical), spend (numeric), and engagement (numeric) help marketing teams identify which combinations of customer characteristics warrant the most attention.
- **Political and Social Science Research:** Choropleth maps of election results, census data, and survey responses by district are a long-established staple of visualizing how opinions and demographics vary across a country or region.

Now that you have learned how to visualize geospatial data and how to combine numerical and non-numerical data in one multitype visualization, you are ready for Unit 3 of this block. That unit introduces interactive dashboards. A dashboard brings together the map-based and multitype charts from this unit and the standard chart types from Unit 1 in a single, explorable interface for a business audience.

One further thread worth drawing out before moving on: every static chart introduced across this unit and Unit 1 shows a single, fixed view of the data, chosen in advance by whoever built the chart. A choropleth map answers 'which region has the highest churn rate?' clearly, but cannot, on its own, let a viewer then click on that region to see the same breakdown by contract type shown in Fig 4.2.3, without the analyst building an

SGOU

entirely separate chart in advance for that specific follow-up question. This limitation of static charts however well designed is precisely the gap that interactive dashboards, introduced next in Unit 3, are built to close, by letting a viewer ask their own follow-up questions of the data directly, rather than being limited to only the specific views an analyst anticipated and built in advance.

Returning one final time to the intern's leadership presentation: she now has a complete visual toolkit for every result her project has produced. The regional churn schematic from Section 4.2.4 lets her show, at a glance, which parts of the business need the most urgent attention. The multitype bubble chart from Section 4.2.6 lets her connect contract type, pricing, churn and customer volume in one image. Without it, she would need several separate charts or a dense table of numbers. The word-frequency display from Section 4.2.5 lets her explain, without technical jargon, what customers complain about most often. None of this required any new statistical calculation beyond what earlier blocks of this course had already produced only a more deliberate choice of how to show it.

Recap

- Geospatial data is classified into point data (a specific location), line data (a path or connection), and polygon data (an area or boundary).
- Common geospatial visualization techniques include choropleth maps (shading regions by value), point maps (marking specific locations), heat maps (showing density), and flow maps (showing movement between locations).
- A choropleth map is conceptually similar to a bar chart, encoding a numeric value using color, but arranges categories according to real geographic position rather than an arbitrary axis.
- Non-numerical (categorical or text-based) data can be visualized using bar and column charts, treemaps, and word-frequency displays such as word clouds.
- Multitype visualizations combine several variables of different types within a single chart by mapping each variable to a distinct visual channel, such as position, color, or size.
- Position and length are the most precisely judged visual channels by the human eye, followed by color and size; effective multitype visualization design assigns the most important variable to the most precise available channel.
- Preattentive attributes are visual properties processed automatically and almost instantly by the human visual system, underlying why certain visual channels communicate certain kinds of information more effectively than others.

SGOU

Objective Type Questions

1. What are the three basic types of geospatial data?
2. Which type of geospatial data represents an area or boundary, such as a state or district?
3. What does a choropleth map use to represent a numeric value associated with each region?
4. Which geospatial visualization technique is best suited to showing the density or concentration of point data across an area?
5. In what way is a choropleth map conceptually similar to a bar chart?
6. What is a treemap?
7. What does a word cloud (word-frequency display) show?
8. What is multitype data?
9. In the contract-type bubble chart worked example, which variable was encoded using bubble size?
10. In the same worked example, which variable was encoded using color?
11. According to visualization research, which two visual channels are judged most precisely by the human eye?
12. What is a preattentive attribute?
13. Why was customer count mapped to bubble size rather than to the x-axis or y-axis in the worked example?
14. What business conclusion did the multitype bubble chart in this unit help reveal about month-to-month contract customers?
15. Why is it important to label a schematic, illustrative map clearly as not being an actual geographic map?

SGOU

Answers

1. Point data, line data, and polygon data
2. Polygon data
3. Color (a color scale applied to shade each region)
4. A (geospatial) heat map

5. Both encode a numeric value using color or length; a choropleth map arranges its categories (regions) by geographic position rather than an arbitrary axis

6. A visualization that represents categories as nested rectangles, with each rectangle's area proportional to a numeric value

7. The words appearing in a body of text, with each word's size scaled according to how frequently it occurs

8. Data that combines variables of more than one type (e.g., numeric and categorical) within a single result or visualization

9. Number of customers

10. Contract type

11. Position and length

12. A visual property, such as position, color, or size, that the human visual system processes automatically and almost instantly, before deliberate attention is applied

13. Because size is a less precise channel best suited to an approximate magnitude, appropriate for a secondary variable rather than the primary comparison being made

14. That month-to-month customers represent the greatest area of retention risk, combining the highest charges, highest churn, and largest customer base

15. Because presenting a stylized diagram as if it were an accurate map could mislead a viewer about actual geographic boundaries, locations, or relationships

SGOU

Assignments

1. For each of the following, state whether it is best represented as point, line, or polygon geospatial data, and justify your answer: (a) the locations of ATMs operated by a bank; (b) the delivery route taken by a courier; (c) the sales territory assigned to each regional manager.

2. A retail chain records the following quarterly sales (₹ lakh) by region: North = 45, South = 62, East = 38, West = 55, Central = 70. Describe how you would design a choropleth-style schematic (similar to Figure 4.2.1) to represent this data, including which region would be shaded darkest and which lightest.

3. Design a multitype visualization (describe it in words, or sketch it) that combines at least one categorical variable and two numeric variables of your own choosing, explaining which visual channel (position, color, size) you would assign to each variable and why.

4. Explain, in your own words, why position and length are generally considered more precise visual channels than color and size, and describe a situation in which using color instead of position for a primary comparison could lead a viewer to a mistaken conclusion.

5. Using the word-frequency data from Block II, Unit 3 (or a small text sample of your own), describe how you would decide whether a bar chart or a word-cloud-style display would better communicate the most frequent words to a business audience.

Suggested Reading

1. Tufte, E. R. (2001). The Visual Display of Quantitative Information (2nd ed.). Graphics Press.

2. Few, S. (2012). Show Me the Numbers: Designing Tables and Graphs to Enlighten (2nd ed.). Analytics Press.

3. Munzner, T. (2014). Visualization Analysis and Design. CRC Press.

Reference

1. Tufte, E. R. (2001). The Visual Display of Quantitative Information (2nd ed.). Graphics Press.

2. Few, S. (2012). Show Me the Numbers: Designing Tables and Graphs to Enlighten (2nd ed.). Analytics Press.

3. Munzner, T. (2014). Visualization Analysis and Design. CRC Press.

SGOU



Interactive Dashboards

Learning Outcomes

After completion of this unit, learner will be able to:

- describe the key components of an interactive dashboard, including KPI cards, charts, filters, and tables, and distinguish common types of interactivity
- apply dashboard design principles, including visual hierarchy, consistent filtering, and effective layout
- design an interactive dashboard that combines multiple chart types to address a central business question and enables detailed exploration through drill down
- identify common pitfalls in dashboard design and explain how to avoid them

Prerequisite

Picture the intern sketching a rough layout on paper for the first time: a row of big numbers at the top, a map in one corner, a few charts below it, and, crucially, a way for her manager to click around and ask his own questions rather than waiting for her to build a new chart for every follow-up. She does not need any new calculation to do this; every chart in her sketch has already been built somewhere in Units 1 and 2 of this block. What she needs now is a way of arranging those charts together, and connecting them to one another, so a viewer can explore rather than just look.

This unit assumes familiarity with the chart types and design principles from Unit 1 and the geospatial and multitype visualizations from Unit 2 of this block. No prior hands-on experience with any specific dashboard software is assumed; general dashboard design principles are introduced here independently of any particular tool, with specific tools (Power BI and Tableau) covered in dedicated blocks later in this course.

SGOU

Keywords

Dashboard, KPI Card, Filter, Slicer, Drill-Down, Cross-Filtering, Tooltip, Visual Hierarchy, Single Source of Truth.

Discussion

4.3.1 Introduction to Interactive Dashboards

A dashboard is a single visual display that brings together several charts, summary figures, and controls, arranged to give a viewer a comprehensive, at-a-glance view of a business area, while also allowing that viewer to interact with the underlying data directly such as filtering, clicking, and exploring, rather than only looking at a fixed image. Every chart in Units 1 and 2 of this block was a static image that showed one fixed view of the data. A dashboard is different. It is a live, explorable interface, usually built with dedicated software such as Power BI or Tableau. It lets a viewer ask their own follow-up questions without needing an analyst to build a new chart each time.

This distinction matters because a static chart, however well designed, can only ever anticipate the questions its creator thought to ask in advance. A dashboard, by contrast, is designed from the outset to support a range of related questions a viewer might reasonably want to explore, exactly the kind of follow-up ('can I see this just for the East region?') that closed Unit 2 of this block. The remainder of this unit introduces the components, interactivity patterns, and design principles that make this possible.

It is worth being precise about what makes a dashboard genuinely 'interactive' rather than merely a collection of charts placed on the same page. Simply arranging several static images side by side into a single-page report, a practice sometimes informally called a 'dashboard' even though it offers no actual interactivity, does not deliver the core benefit described above, since a viewer still cannot filter, click, or explore beyond exactly what was placed on the page in advance. A genuinely interactive dashboard requires an underlying connection to live, queryable data, together with software capable of updating every chart on the page in response to a viewer's actions, which is precisely why dedicated dashboard software, rather than a static image-editing tool, is used to build them in practice.

4.3.2 Key Components of a Dashboard

Most well-designed dashboards, regardless of the specific software used to build them, are assembled from a small, recurring set of components:

- **KPI Cards :** Prominent, large-format displays of a single key number (a Key Performance Indicator), such as an overall churn rate, total revenue, or model accuracy, intended to be readable in under a second, without requiring any interpretation of a chart.

SGOU

- **Charts :** The bar charts, line charts, choropleth maps, and multitype visualizations introduced in Units 1 and 2 of this block, embedded within the dashboard rather than presented in isolation.

- **Filters (Slicers):** Interactive controls, dropdown menus, date-range selectors, or clickable buttons , that let a viewer narrow the entire dashboard down to a specific subset of the data, such as a single region, contract type, or time period.

- **Tables:** Detailed, row-level data, often included for viewers who want to inspect exact underlying figures behind a chart or KPI card, rather than only the summarized visual.

These four component types map naturally onto a rough hierarchy of detail: a KPI card offers the least detail but the fastest possible read; a chart offers an intermediate, visual level of detail; and a table offers the greatest detail but requires the most effort from a viewer to interpret. A well-designed dashboard usually leads with KPI cards and supports them with charts. Detailed tables are kept for viewers who want to dig deeper. They are often placed behind a drill-down interaction or a separate detail view, instead of being shown prominently on the main page. This is the progressive disclosure of detail that the data-ink principle from Unit 1 of this block would recommend.

These components are combined into a single-page (or occasionally multi-page) layout, designed so that the most important information is immediately visible without scrolling, and any further detail is available through interaction rather than requiring an entirely separate document.

4.3.3 Types of Dashboard Interactivity

Several distinct forms of interactivity are commonly built into dashboards, each supporting a different kind of exploration:

- **Filtering :** Selecting a value in a filter control (such as choosing 'East' from a region dropdown) narrows every chart and KPI card on the dashboard down to that specific subset of the data at once.

- **Drill-Down :** Clicking directly on part of a chart itself (such as a specific region on a map, or a specific bar in a bar chart) reveals a more detailed view focused on just that selection, without needing to use a separate filter control, explored in detail through a worked example in Section 4.3.6.

- **Tooltips (Hover Details) :** Hovering the mouse pointer over a specific data point reveals additional detail in a small pop-up box, without cluttering the main chart with every possible number at once, directly applying the data-ink principle from Unit 1 of this block, by keeping detailed numbers hidden until specifically requested.

- **Cross-Filtering (Highlighting) :** Clicking on part of one chart automatically highlights or filters the corresponding data in every other chart on the same dashboard, letting a viewer see, for instance, exactly which regions the highest-charge, highest-churn customers from a bubble chart are concentrated in, simply by clicking that bubble.

SGOU

Together, these interactivity types transform a fixed collection of charts into a genuinely explorable tool, a viewer can start from any chart, filter or drill into it, and immediately see every other chart on the dashboard update to reflect that same selection, all without needing an analyst to intervene.

It is worth noting that these interactivity types are not mutually exclusive alternatives, but are typically combined within a single, well-designed dashboard. A viewer might first use a top-level filter to narrow the whole dashboard to a specific time period. Next, the viewer might drill down by clicking a region on a map. Then the viewer might hover over a bar in the filtered chart to see its exact value in a tooltip. These are three different types of interactivity working together in one continuous exploration. The viewer never needs to ask an analyst for a new chart.

4.3.4 Dashboard Design Principles

Designing an effective dashboard extends the chart-level principles introduced in Unit 1 of this block (maximizing the data-ink ratio, avoiding chartjunk, labelling clearly) to the level of an entire page combining several charts at once:

- **Establish a Clear Visual Hierarchy:** The most important information typically the KPI cards, should occupy the most prominent position, usually the top of the page, with supporting charts arranged below in order of importance, so a viewer's eye naturally lands on the most critical figures first.
- **Limit the Number of Charts:** A dashboard crowded with ten or more competing charts overwhelms a viewer and defeats the purpose of a quick, at-a-glance summary; a smaller number of carefully chosen, well-designed charts, each earning its place, is almost always more effective.
- **Keep Filters Consistent and Prominent:** Filter controls should be placed in a consistent, easily located position (commonly along the top or left side of the dashboard) and should affect every relevant chart on the page uniformly, so a viewer never has to wonder whether a given filter has actually been applied to what they are currently looking at.
- **Avoid Requiring Scrolling for Key Information:** Wherever possible, the most important KPIs and charts should be visible without scrolling, since a viewer who must scroll to find the headline number may simply miss it altogether.
- **Maintain a Single Source of Truth:** Every chart and KPI card on a dashboard should be calculated from the same underlying, consistently defined data and business rules (for example, a single, agreed definition of 'churn'), so that two different charts on the same dashboard never appear to contradict one another due to inconsistent underlying definitions.

SGOU

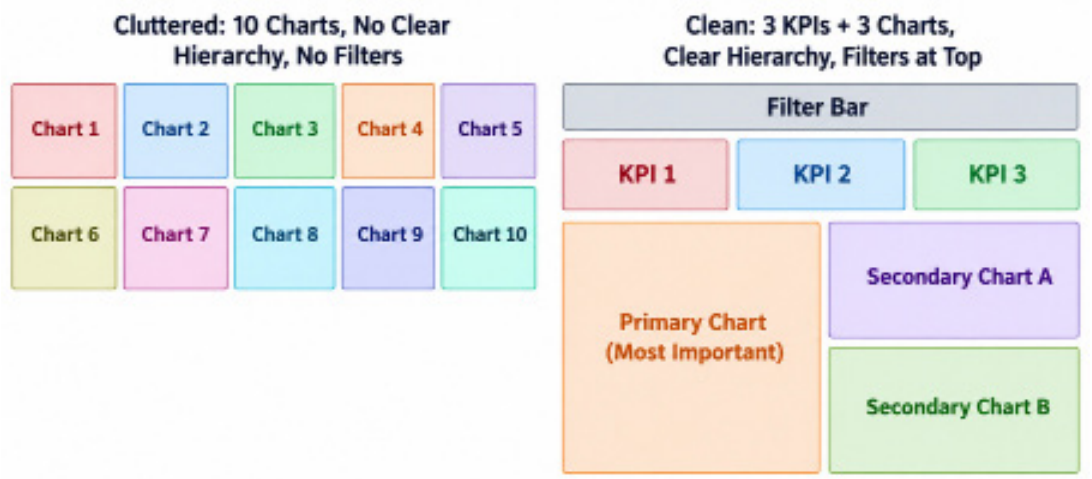


Fig. 4.3.1 Cluttered Versus Clean Dashboard Design

Figure 4.3.1 shows two layouts. The cluttered layout on the left has ten competing charts, no clear starting point and no filters, so the viewer has to hunt for the information that matters most. The clean layout on the right draws the eye to three prominent KPI cards, followed by one clearly primary chart and two clearly secondary charts. A consistent filter bar stays available at the top. The clean layout applies exactly the same underlying data, it simply organizes and prioritizes it far more deliberately.

A useful, practical test for evaluating a dashboard's design is what might be called the 'five-second rule': a viewer glancing at the dashboard for only five seconds should be able to state the single most important figure or finding it contains. Applying this test to the cluttered layout in Fig 4.3.1 fails immediately, since a viewer's attention has nowhere obvious to land among ten equally weighted charts; applying the same test to the clean layout succeeds easily, since the three KPI cards at the top are unmistakably the first thing a viewer's eye encounters. Designers who find themselves unable to answer their own five-second test for a dashboard they have built should generally treat that as a signal to simplify and re-prioritize, rather than to add further explanatory charts or text in an attempt to compensate.

4.3.5 A Worked Example: Designing a Churn Analysis Dashboard

Bringing together nearly every chart type introduced across this block, consider a dashboard designed around the telecom churn project followed throughout this course, intended for the intern's leadership presentation.

SGOU

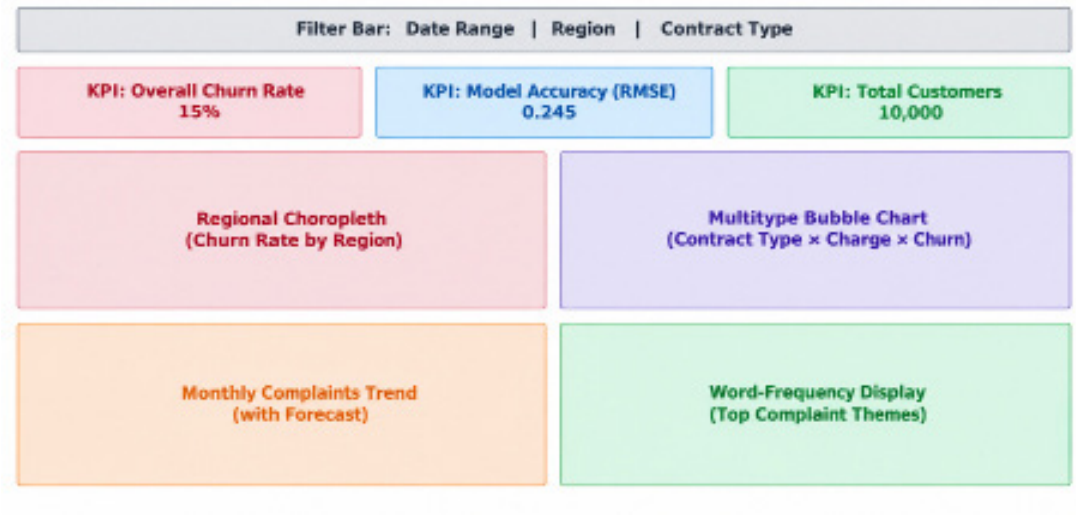


Fig. 4.3.2 Wireframe Layout for a Churn Analysis Dashboard

As shown in Fig 4.3.2, the dashboard is organized into three clear layers, directly applying the visual hierarchy principle from Section 4.3.4:

- **Filter Bar (Top):** Date range, region, and contract type filters, positioned prominently and consistently at the very top, affecting every chart below it.
- **KPI Row:** Three headline figures, overall churn rate (15%), model accuracy expressed via RMSE (0.245, from Block III, Unit 2), and total customer count (10,000), immediately visible without any interaction required.
- **Supporting Charts:** The regional choropleth and multitype bubble chart from Unit 2 of this block occupy the middle row, while the monthly complaints trend (with forecast, from Block III, Units 3 and 4) and the word-frequency display of complaint themes (from Unit 2 of this block) occupy the bottom row.

Every chart on this dashboard was already designed and justified individually earlier in this course. The contribution of this unit is to arrange the charts together, connect them through shared filters, and prioritize them with a clear visual hierarchy. A single page can then answer 'how is churn doing, where, and why?', which is a much broader question than any single chart could answer alone.

4.3.6 A Worked Example: Drill-Down Interactivity

Continuing with the churn dashboard from Section 4.3.5, suppose the intern's manager, viewing the regional choropleth, clicks directly on the East region the region with the highest churn rate identified back in Unit 2 of this block.

SGOU

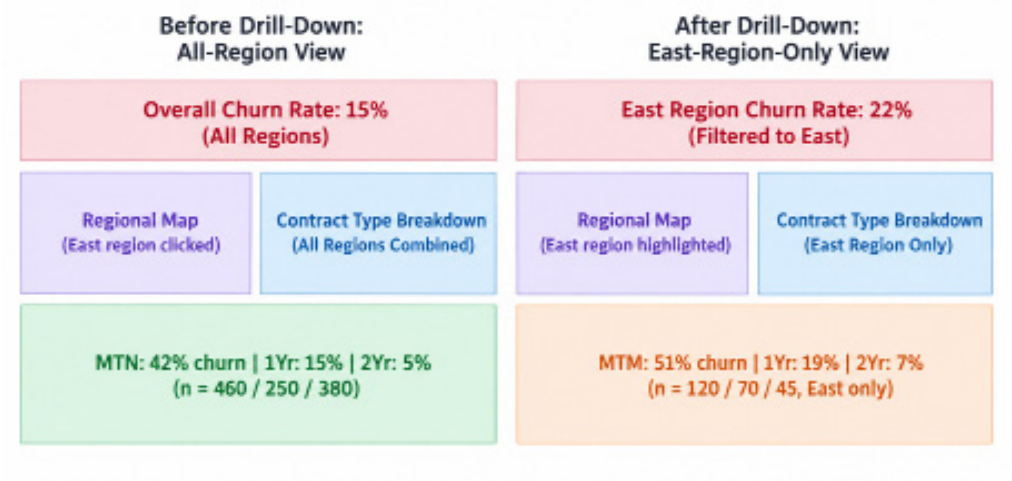


Fig. 4.3.3 Drill-Down Interactivity - Clicking a Region Filters the Entire Dashboard

As Figure 4.3.3 shows, this single click on the map triggers a cascade of updates across the whole dashboard. The intern does not need to build a single new chart. The top KPI card changes from the overall churn rate (15%, across all regions) to the East-specific churn rate (22%, matching the figure first shown in Unit 2's choropleth). The contract-type bubble chart changes from combined figures for all regions to figures for East alone. This shows that East's month-to-month churn rate (51%) is much worse than the company-wide month-to-month figure (42%) shown in Unit 2. East's two-year contract churn rate (7%) is only slightly above the company-wide figure (5%).

This single drill-down reveals a useful business insight that neither the company-wide bubble chart nor the regional choropleth could show alone. East's churn problem is not spread evenly across contract types. It is concentrated among month-to-month customers. A retention campaign aimed at month-to-month customers in the East region, instead of a broad company-wide campaign, may therefore be the most efficient way to address the region's churn. Producing this same insight without drill-down interactivity would have required the intern to build a separate, dedicated chart in advance for every region, on the chance that a viewer might ask about that specific region exactly the limitation of static charts that motivated this unit in the first place.

This example also illustrates why drill-down interactivity is often described as supporting a 'top-down' style of exploration: a viewer begins at the most aggregated, summary level (the company-wide churn rate), and progressively narrows in on more specific detail only where the summary view suggests something worth investigating further. This repeats, in a single interaction, the analytical habit that runs through this whole course: start with a broad question and progressively drill into the specific data, variables or techniques that best address it. Now the dashboard viewer can do this directly. It is no longer a process that only the original analyst can carry out.

4.3.7 Common Pitfalls in Dashboard Design

Beyond the general clutter illustrated in Figure 4.3.1, several further pitfalls specifically affect interactive dashboards:

SGOU

- **Inconsistent Filter Behaviour:** If some charts on a dashboard respond to a given filter while others do not, a viewer can easily draw an incorrect conclusion, believing they are looking at filtered data when part of the dashboard is, in fact, still showing the unfiltered total.

- **Overloading a Single Page:** Attempting to answer every possible business question on one dashboard, rather than designing focused, separate dashboards for genuinely distinct audiences or purposes, tends to produce an overwhelming, unfocused result, echoing the 'limit the number of charts' principle from Section 4.3.4 but applied at the level of an entire dashboard's overall scope.

- **Poor Performance with Large Datasets:** A dashboard connected directly to a very large underlying dataset, without appropriate aggregation or the kind of big-data infrastructure introduced in Block II, Unit 1, can become slow and unresponsive, undermining the very interactivity that makes a dashboard valuable in the first place.

- **Interactivity Without Purpose:** Adding filters, drill-downs, or other interactive elements simply because the software makes them easy to add, rather than because a genuine business question calls for them, adds complexity without adding value echoing the chartjunk caution from Unit 1 of this block, now applied to interactive features rather than static visual decoration.

Avoiding these pitfalls generally comes back to the same discipline introduced throughout this block: every component of a dashboard, whether a chart, a filter, or a drill-down interaction, should earn its place by genuinely helping a specific, identified viewer answer a specific, identified question, rather than being included simply because it was technically possible to add.

A practical safeguard against several of these pitfalls at once is to test a dashboard, before it is finalized, with someone who was not involved in building it, ideally a genuine representative of its intended audience, such as a business stakeholder rather than a fellow analyst. An analyst who has spent days building a dashboard knows every filter, chart and data definition very well. This makes it easy to miss points of confusion that a first-time viewer would notice at once. Examples are an unlabelled filter, a chart that is not filtered like the others, or a KPI card whose definition does not match what the viewer expects. This kind of informal usability testing, however brief, very often catches exactly the kind of pitfalls introduced in this section before they reach a real audience.

4.3.8 Tools for Building Interactive Dashboards

While this unit has deliberately introduced dashboard design principles independently of any specific software, in practice, interactive dashboards are built using dedicated business intelligence tools, several of which were first introduced in Block II, Unit 4 of this course:

- **Power BI:** Microsoft's business intelligence platform, offering native support for filters, drill-down, and cross-filtering, and covered in dedicated depth in Block V of this course.

SGOU

- **Tableau:** A dedicated visualization platform particularly well regarded for the flexibility and polish of its interactive dashboards, covered in dedicated depth in Block VI of this course.

- **Other Tools:** Spreadsheet applications such as Microsoft Excel offer basic dashboard capabilities through pivot tables and slicers, while cloud-based tools such as Google Looker Studio provide free, web-based dashboard building, each offering a different balance of cost, flexibility, and ease of use.

Every design principle introduced in this unit such as visual hierarchy, consistent filtering, limiting the number of charts, avoiding interactivity without purpose remove applies regardless of which specific tool ultimately builds the dashboard, since these principles concern how information should be organized and presented to a viewer, a question that exists independently of any particular piece of software.

This tool-independence is worth emphasizing precisely because it is easy to conflate learning a specific piece of software with learning to design an effective dashboard. A learner who becomes highly proficient at every button and menu in a particular tool, without internalizing the underlying design principles introduced in this unit, risks producing dashboards that are technically well built but poorly organized, cluttered, inconsistent, or lacking a clear point of focus. The reverse is far more valuable. A learner who understands visual hierarchy, purposeful interactivity and a single source of truth can use that understanding in Power BI, in Tableau, or in any future tool not yet invented. This is why this unit introduces these principles before either specific platform is covered in the blocks that follow.

4.3.9 Applications of Interactive Dashboards in Practice

Interactive dashboards are among the most widely used data analytics deliverables across virtually every industry:

- **Executive Reporting :** Company leadership routinely relies on dashboards summarizing sales, customer, and operational KPIs, allowing executives to explore performance by region, product line, or time period without needing to request a new report for every question.

- **Operations Monitoring :** Real-time dashboards tracking server performance, manufacturing output, or logistics status allow operations teams to spot and drill into emerging problems as they happen.

- **Marketing Analytics :** Dashboards combining campaign performance, customer segments, and sales figures let marketing teams explore which combinations of channel, audience, and message are performing best.

- **Public Sector and Open Data:** Government agencies and public health bodies increasingly publish interactive public dashboards, COVID-19 case tracking dashboards being a widely recognized recent example, allowing the public to explore data directly rather than relying solely on periodic static reports.

SGOU

Returning a final time to the intern's leadership meeting: with the dashboard designed in Section 4.3.5 and the drill-down capability demonstrated in Section 4.3.6, she is no longer limited to presenting a fixed sequence of static slides. Her manager, and any other executive in the room, can now explore the churn data directly during the meeting itself filtering to a region of particular interest, drilling into a contract type that catches their attention, all without the intern needing to have anticipated every possible question in advance. This is the central promise of interactive dashboards: not simply a more attractive way of showing the same static charts, but a genuinely different way of engaging an audience with data, letting them participate directly in the analysis rather than only receiving its conclusions.

Now that you have learned how to design and structure an interactive dashboard, using the chart types, geospatial and multitype techniques and design principles from this block, you are ready for Unit 4. That unit formalizes the complete process, called the data visualization cycle, by which a visualization or dashboard is planned, built, tested and refined. It runs from the first business question to a finished, well-communicated result.

Recap

- A dashboard is a single, live, explorable interface combining charts, KPI cards, filters, and tables, in contrast to the fixed, static charts introduced in Units 1 and 2 of this block.
- Key dashboard components include KPI cards (prominent single numbers), charts, filters (slicers), and detailed tables.
- Common types of dashboard interactivity include filtering, drill-down (clicking into a chart element for more detail), tooltips (hover details), and cross-filtering (highlighting related data across charts).
- Effective dashboard design establishes a clear visual hierarchy, limits the number of charts, keeps filters consistent and prominent, avoids requiring scrolling for key information, and maintains a single source of truth across all components.
- A worked dashboard layout for the telecom churn project combined KPI cards, a regional choropleth, a multitype bubble chart, a trend chart with forecast, and a word-frequency display, connected by a consistent filter bar.
- Drill-down interactivity lets a viewer click into a specific element (such as a region) to reveal a more detailed view, often uncovering patterns -- such as churn being concentrated in a specific contract type within a specific region -- that no single static chart could reveal alone.
- Common dashboard pitfalls include inconsistent filter behaviour, overloading a single page, poor performance with large datasets, and adding interactivity without a genuine underlying purpose.

SGOU

- Dashboard design principles apply independently of the specific software tool used to build the dashboard, whether Power BI, Tableau, or another platform.

Objective Type Questions

1. What is an interactive dashboard?
2. What is the main limitation of a static chart that dashboards are designed to address?
3. What is a KPI card?
4. What is the purpose of a filter (slicer) on a dashboard?
5. What is drill-down interactivity?
6. What is a tooltip in the context of a dashboard?
7. What is cross-filtering (highlighting)?
8. What does establishing a clear visual hierarchy on a dashboard mean?
9. What is a 'single source of truth' in dashboard design, and why does it matter?
10. In the worked churn dashboard example, what three figures were shown as KPI cards?
11. In the drill-down worked example, what business insight was revealed after clicking on the East region?
12. What problem can arise if some charts on a dashboard respond to a filter while others do not?
13. Why can connecting a dashboard directly to a very large, unaggregated dataset cause problems?
14. What does 'interactivity without purpose' mean, and why is it a pitfall?
15. Name two business intelligence tools commonly used to build interactive dashboards, both introduced earlier in this course.

SGOU

Answers

1. A single, live, explorable visual display combining charts, KPI cards, filters, and tables, allowing a viewer to interact with the underlying data directly
2. A static chart can only show one predetermined view and cannot answer follow-up questions the creator did not anticipate in advance
3. A prominent, large-format display of a single key number, readable at a glance without requiring interpretation of a chart
4. To let a viewer narrow the entire dashboard down to a specific subset of the data, such as a single region or time period
5. Clicking directly on part of a chart to reveal a more detailed view focused on just that selection
6. A small pop-up box revealing additional detail when a viewer hovers the mouse pointer over a specific data point
7. Clicking on part of one chart automatically highlights or filters the corresponding data in every other chart on the dashboard
8. Arranging the most important information (typically KPI cards) most prominently, with supporting charts arranged below in order of importance
9. Ensuring every chart and KPI card is calculated from the same underlying, consistently defined data and business rules, so that different charts never appear to contradict one another
10. Overall churn rate (15%), model accuracy via RMSE (0.245), and total customer count (10,000)
11. That East's churn problem is heavily concentrated among month-to-month customers specifically, rather than being evenly spread across contract types
12. A viewer can easily draw an incorrect conclusion, believing they are looking at filtered data when part of the dashboard is still showing the unfiltered total
13. It can become slow and unresponsive, undermining the interactivity that makes a dashboard valuable
14. Adding filters, drill-downs, or other interactive elements because the software makes them easy to add rather than because a genuine business question calls for them, adding complexity without adding value
15. Power BI and Tableau

SGOU

Assignments

1. Sketch (on paper or in a simple diagram) a wireframe layout, similar in style to Figure 4.3.2, for a dashboard summarizing a college's student attendance and performance data. Identify at least three KPI cards, two supporting charts, and one filter you would include.
2. Explain, in your own words, the difference between filtering and drill-down interactivity, using an example of your own (different from the telecom churn example used in this unit) for each.
3. Identify the cluttered dashboard design pitfall (from Figure 4.3.2 or Section 4.3.7) that you think would be easiest for a beginner to accidentally fall into, and explain, with a concrete example, how you would avoid it.
4. Describe a realistic scenario, different from the churn dashboard example in this unit, where drill-down interactivity would reveal an important insight that a single static chart could not.
5. Explain, in your own words, why maintaining a 'single source of truth' across a dashboard's charts and KPI cards is important, and describe what might go wrong for a business if two charts on the same dashboard used inconsistent underlying definitions.

Suggested Reading

1. Few, S. (2013). Information Dashboard Design: Displaying Data for At-a-Glance Monitoring (2nd ed.). Analytics Press.
2. Tufte, E. R. (2001). The Visual Display of Quantitative Information (2nd ed.). Graphics Press.
3. Munzner, T. (2014). Visualization Analysis and Design. CRC Press.

Reference

1. Few, S. (2013). Information Dashboard Design: Displaying Data for At-a-Glance Monitoring (2nd ed.). Analytics Press.
2. Tufte, E. R. (2001). The Visual Display of Quantitative Information (2nd ed.). Graphics Press.
3. Munzner, T. (2014). Visualization Analysis and Design. CRC Press.

SGOU



UNIT 4 Data Visualization Cycle

Learning Outcomes

After completion of this unit, learner will be able to:

- describe the data visualization reference pipeline and explain the role of each stage in the visualization process
- explain the iterative data visualization cycle, including understanding the data and audience, data preparation, visual encoding, interaction, interpretation, and communication
- select appropriate visual encodings based on the analytical purpose, audience, and characteristics of the data
- explain how evaluation and feedback improve visualizations through iteration and refinement

Prerequisite

The first three units of this block addressed the data visualization process piece by piece: Unit 1 introduced the basic purpose of visualization and the challenges involved in doing it well, Unit 2 examined how geospatial and mixed data types are represented visually, and Unit 3 explored how interactivity turns a static chart into an explorable dashboard. This unit draws these threads together into a single, coherent model of how a data visualization is actually produced, from the moment raw data is encountered to the moment a finished visualization is evaluated and refined.

To engage meaningfully with this unit, learners should be comfortable with the outcomes of Units 1 to 3 of this block, particularly the notion that different chart types suit different kinds of data, and that interactivity allows a single visualization to serve more than one analytical question. Learners should also recall the data analytics life cycle introduced in Block II, since the visualization cycle described in this unit can be understood as a detailed elaboration of the presentation and communication stages of that broader life cycle.

Keywords

Visualization Pipeline, Data Transformation, Visual Mapping, View Transformation, Human Interaction, Visual Encoding, Interpretation, Storytelling, Evaluation, Iteration.

SGOU

Discussion

4.4.1 Introduction to the Data Visualization Cycle

A finished chart or dashboard is the visible end product of a process that is rarely as linear as the final image suggests. Behind any effective visualization lies a sequence of decisions: what the underlying data actually contains, what question the audience is trying to answer, which visual form best represents the relevant pattern, how a viewer will interact with the result, and how the creator will know whether the visualization actually succeeded. The data visualization cycle is the name given to this sequence of decisions, viewed as a repeatable process rather than a one-off act of chart-making.

Understanding this cycle serves two purposes. First, it provides a checklist that helps a learner avoid the most common visualization failures, such as choosing an attractive chart type that nonetheless misrepresents the underlying data, or building a technically correct chart that no one outside its creator can actually interpret. Second, it emphasizes that visualization is rarely finished after a single attempt; real visualizations are refined through cycles of feedback and revision, in much the same way that a piece of writing is redrafted rather than published on the first attempt.

This unit examines the visualization cycle from two complementary angles. The first is a formal, technical model, known as the visualization reference pipeline, that describes how raw data is progressively transformed into a rendered view on screen. The second is a practical, workflow-oriented cycle that describes the broader human activities, understanding, preparing, visualizing, interacting, interpreting, communicating, and evaluating, that surround and give purpose to that technical pipeline.

It is worth noting that these two models are not competing descriptions of the same thing, but operate at different levels of detail. The reference pipeline describes what happens computationally, inside a visualization tool, each time data is rendered into a picture, and applies equally whether the tool in question is a general-purpose programming library, a dedicated business intelligence product such as Power BI or Tableau, discussed later in this course, or a simple spreadsheet chart. The practical cycle, by contrast, describes the surrounding human process, largely independent of any specific tool, that determines what data is fed into the pipeline in the first place and what is eventually done with the pipeline's output. A learner who understands both models is equipped not only to build a visualization correctly, but to understand why a particular visualization succeeded or failed to communicate its intended message.

4.4.2 The Data Visualization Reference Pipeline

One of the most influential models of how a visualization is produced was proposed by researchers Card, Mackinlay, and Shneiderman, who described visualization as a pipeline of transformations that progressively convert raw data into a viewable image. As illustrated in fig. 4.4.1, this pipeline consists of four principal stages, connected by three transformation steps, with human interaction able to feed back into every stage of the process.

SGOU

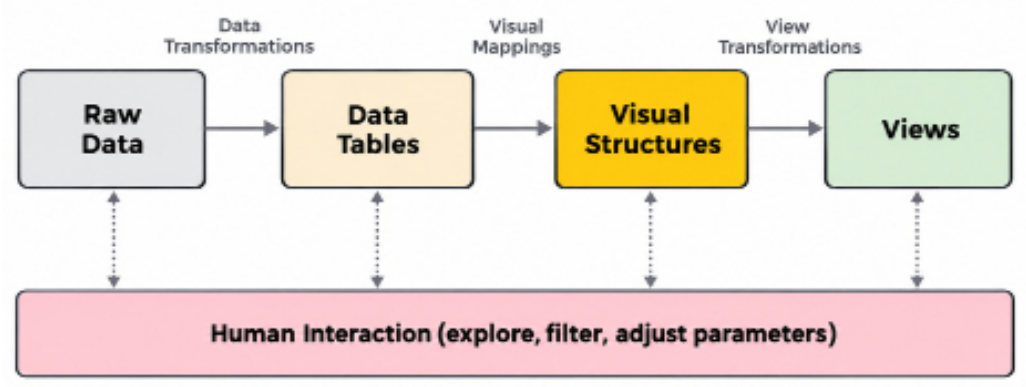


Fig. 4.4.1 The Data Visualization Reference Pipeline

As shown in Figure 4.4.1, raw data, which may be messy, incomplete, or stored in a form unsuited to direct analysis, is first converted through data transformations into data tables, structured collections of rows and columns similar to those discussed extensively in earlier blocks of this course. Visual mappings then convert this structured data into visual structures, which define how data values correspond to visual properties such as position, size, shape, and colour. Finally, view transformations render these visual structures into the actual views a viewer sees on screen, applying considerations such as scaling, zooming, or the specific window of data currently displayed.

4.4.2.1 Data Transformations

This first stage converts raw, often heterogeneous data into a consistent, structured data table. This closely parallels the data cleaning and transformation activities discussed in Block II and in Block V's treatment of Power BI's Power Query Editor, and includes tasks such as handling missing values, correcting data types, and reshaping data into rows and columns suited to analysis.

4.4.2.2 Visual Mappings

This stage assigns data fields to visual properties, commonly referred to as visual encodings or channels, such as position on an axis, length, area, colour hue or intensity, and shape. The choice of mapping is central to how effectively a visualization communicates its underlying data, and is discussed further in section 4.4.5 of this unit.

4.4.2.3 View Transformations

This final stage governs how the visual structure is actually rendered into a specific view, including decisions such as which portion of a large dataset is currently visible, the level of zoom applied, and how multiple views might be arranged together on a dashboard, echoing the interactive dashboard concepts introduced in Unit 3 of this block.

4.4.2.4 The Role of Human Interaction

As shown at the base of Figure 4.4.1, human interaction is not confined to the final

SGOU

view; a viewer's actions, such as applying a filter, changing a parameter, or selecting a different data field, can feed back into any stage of the pipeline, prompting new data transformations, new visual mappings, or a new view transformation. This is precisely what happens when a viewer interacts with a slicer or drill-through control in an interactive dashboard of the kind covered in Unit 3, and explains why interactive visualizations are often described as supporting a continuous, exploratory dialogue between the viewer and the data, rather than presenting a single, fixed image.

4.4.3 The Practical Data Visualization Cycle

While the reference pipeline in section 4.4.2 describes the technical transformations that occur within a visualization tool, a broader, more practical cycle describes the human activities that surround the creation of any real-world visualization, from an analyst's earliest engagement with a dataset through to the eventual evaluation of the finished result. As illustrated in Fig 4.4.2, this practical cycle can be described as seven interconnected stages, each of which is examined in the sections that follow.

This practical cycle is deliberately tool-independent. Whether a visualization is ultimately built using a general-purpose charting library, a spreadsheet application, or a dedicated platform such as Power BI or Tableau, covered later in this course, the same seven human activities apply. What changes from one tool to another is largely how quickly and easily each stage can be carried out, and how much of the underlying reference pipeline the tool automates on the creator's behalf, rather than the fundamental sequence of activities itself.

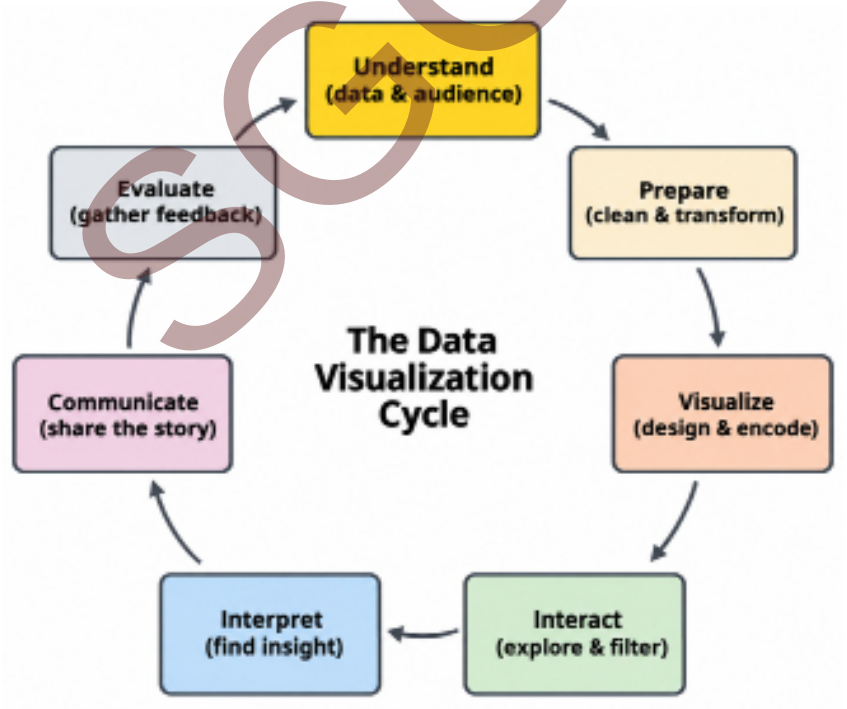


Fig. 4.4.2 The Data Visualization Cycle as an Iterative Process

As shown in Fig 4.4.2, the cycle is drawn as a closed loop rather than a straight line, reflecting the fact that real visualization projects rarely proceed in a single pass

from understanding to evaluation. A designer may return to the preparation stage after discovering a data quality issue while visualizing, or return to the visualize stage after feedback gathered during evaluation, making the cycle genuinely iterative rather than strictly sequential.

Table 4.4.1 relates the seven stages of the practical cycle to the more technical reference pipeline described in section 4.4.2, showing how the two models align even though they describe the process at different levels of detail.

Table 4.4.1 The seven stages of the practical cycle to the more technical reference pipeline

Practical Cycle Stage	Corresponding Pipeline Concept
Understand	Defines the requirements that later data transformations and visual mappings must satisfy
Prepare	Corresponds directly to the data transformations step
Visualize	Corresponds directly to the visual mappings step
Interact	Corresponds to the human interaction loop feeding back into the pipeline
Interpret / Communicate	Occurs after the view transformation step, using the rendered view
Evaluate	Assesses the overall pipeline output and triggers a new cycle

4.4.4 Stage 1: Understanding the Data and the Audience

Every visualization project begins, whether explicitly or implicitly, with a period of understanding. This involves two related activities. The first is understanding the data itself: its structure, its quality and the kinds of data it contains (numerical, categorical, geospatial or other, as discussed in Unit 2 of this block). The second is understanding the audience who will view the visualization: how familiar they are with the subject, the questions they need answered and the setting in which they will view the result, such as a printed report, a live presentation or an interactive dashboard.

Skipping this stage is one of the most common causes of ineffective visualization. A chart that is technically accurate but answers a question the audience was not actually asking, or that assumes a level of statistical familiarity the audience does not possess, is unlikely to succeed regardless of how carefully it is constructed in later stages of the cycle.

SGOU

4.4.5 Stage 2: Preparing the Data

Once the data and audience are understood, the data itself must be prepared, corresponding closely to the data transformations stage of the reference pipeline described in section 4.4.2. This includes cleaning the data of errors, duplicates, and missing values, converting it into a structure suited to the intended visualization, and, where necessary, aggregating or summarizing detailed records into the level of granularity the audience actually needs to see.

The preparation stage directly constrains what is possible in later stages: a visualization cannot show a trend over time if the underlying data lacks a properly structured date field, and a geospatial visualization cannot plot locations accurately if location data has not been cleaned and standardized. For this reason, experienced practitioners often report that data preparation consumes a disproportionate share of the total time spent on a visualization project, a pattern also noted in the discussion of the data analytics life cycle in Block II.

4.4.6 Stage 3: Choosing Visual Encodings and Designing the Visualization

With prepared data in hand, the visualize stage corresponds to the visual mappings step of the reference pipeline, translating data fields into visual properties such as position, length, colour, and shape. Some visual channels are better than others for precise numeric comparisons. Position along a common scale and length are generally easier for a viewer to compare accurately than area or colour intensity. This is why bar and column charts remain reliable choices for comparison tasks, as noted in the chart selection principles in earlier units of this course.

Beyond the choice of chart type itself, this stage also involves broader design decisions: selecting a colour palette that remains legible for viewers with colour vision deficiencies, choosing clear and sufficient labelling, and avoiding unnecessary decorative elements that add visual clutter without conveying additional information. These design choices determine not only whether a visualization is accurate, but whether it is genuinely usable by its intended audience.

This general principle, that some visual channels support more accurate judgement than others, was studied systematically by researchers William Cleveland and Robert McGill, whose findings are commonly summarized as a ranked ordering of visual encodings by accuracy. Table 4.4.2 presents a simplified version of this ordering, from the most to the least accurate for typical quantitative comparison tasks.

Table 4.4.2 Ranked ordering of visual encodings by accuracy

Rank	Visual Encoding	Example Chart Using This Encoding
1 (Most accurate)	Position along a common scale	Bar chart, dot plot

SGOU

2	Length	Bar chart (bar length)
3	Angle / Slope	Line chart slope
4	Area	Bubble chart, treemap
5	Colour intensity or saturation	Choropleth (heat) map
6(Least accurate)	Colour hue alone	Categorical colour coding

This ordering does not mean that lower-ranked encodings such as colour or area should never be used; area and colour intensity remain well suited to showing geographic patterns, as discussed in Unit 2 of this block, and to indicating a secondary variable alongside a primary one. Rather, the ordering is a guide for the visualization stage: when the audience needs to make a precise quantitative comparison, a designer should prefer position or length wherever practical, and reserve less accurate encodings for secondary or supporting information.

4.4.7 Stage 4: Interaction and Exploration

For interactive visualizations, such as the dashboards discussed in Unit 3 of this block, the cycle includes a distinct interaction stage, during which a viewer actively explores the data through filtering, drilling down into greater detail, hovering for additional information, or adjusting parameters. This stage corresponds to the human interaction loop described in the reference pipeline in section 4.4.2, and transforms visualization from a passive act of viewing into an active process of inquiry, allowing a single well-designed dashboard to answer many different questions for many different viewers.

4.4.8 Stage 5: Interpretation and Insight Generation

Producing a visualization is not the same as extracting insight from it. The interpretation stage involves examining the rendered visualization, whether static or interactive, and identifying the patterns, trends, comparisons, or anomalies it reveals. This stage draws on the audience's domain knowledge as much as on the visualization itself; two viewers looking at the same chart may draw different conclusions depending on their familiarity with the underlying subject matter, which is precisely why the understanding stage described in section 4.4.4 places such emphasis on knowing the intended audience in advance.

Interpretation is also where the risk of misreading a visualization is greatest, since a viewer's prior expectations can lead them to see a pattern that the data does not actually support, or to overlook a genuine pattern that contradicts what they expected to find. For this reason, careful interpretation often involves checking an apparent pattern against the underlying data table before treating it as a reliable finding, particularly when the pattern would lead to a significant decision, echoing the general caution

SGOU

around correlation and causation introduced in the statistical treatment of data earlier in this course.

4.4.9 Stage 6: Communication and Storytelling

Once insight has been identified, it must typically be communicated to others, whether through a written report, a live presentation, or a shared interactive dashboard. Effective communication often involves data storytelling, arranging visualizations and accompanying narrative text in a deliberate sequence that guides the audience toward a specific conclusion, rather than presenting every possible chart and leaving the audience to find the relevant pattern unaided. Annotations, titles, and carefully chosen highlighting can direct a viewer's attention to the specific insight the communicator wishes to convey.

The appropriate level of detail and interactivity in this stage depends heavily on the delivery context established during the understand stage. A visualization intended for a brief executive summary typically benefits from a single, clearly annotated chart that makes one point decisively, whereas a visualization intended for an analyst who will explore the data independently benefits from the fuller interactivity discussed in the previous stage. Recognizing which of these contexts applies is itself part of effective communication, since an overly simplified view can frustrate an analytical audience, while an overly complex, unguided dashboard can overwhelm an audience seeking a quick, clear answer.

4.4.10 Stage 7: Evaluation and Iteration

The final stage of the practical cycle involves evaluating whether the visualization actually achieved its purpose. This may take the form of informal feedback from colleagues, formal usability testing with representative members of the intended audience, or simply the creator's own critical review against the original understanding of the audience's needs established in the first stage of the cycle. Common evaluation questions include whether viewers can accurately read specific values from the visualization, whether they can correctly describe the main pattern or trend it shows, and whether they can complete a specific task, such as identifying the highest-performing category, without additional explanation.

As illustrated by the closed loop in Fig 4.4.2, evaluation is not the end of the process but a bridge back to the beginning of the next iteration. Feedback gathered at this stage commonly leads back to the understand stage, if the visualization answered the wrong question, back to the prepare stage, if a data quality issue was discovered, or back to the visualize stage, if the chosen visual encoding proved confusing to the audience. This capacity to loop back is what distinguishes the data visualization cycle from a simple, one-directional production process.

SGOU

4.4.11 Best Practices Across the Visualization Cycle

- Revisit the understand stage explicitly whenever a project's audience or purpose is unclear, rather than proceeding directly to chart construction.
- Treat data preparation as an integral part of the visualization process rather than a separate, purely technical task, since preparation choices directly determine what can later be visualized.
- Favour visual encodings, such as position and length, that support accurate quantitative comparison over encodings, such as colour intensity or area, that are more prone to misjudgement, particularly for precise comparisons.
- Design interactive elements with a specific analytical purpose in mind, rather than adding interactivity for its own sake.
- Build evaluation into the process from the outset, for example by sharing early drafts with a representative viewer, rather than treating evaluation as an afterthought once a visualization is considered finished.

Applying the data visualization cycle consistently helps ensure that a finished chart or dashboard is not merely technically correct, but genuinely fit for the audience and purpose it was created to serve, and that lessons learned from one visualization project carry forward productively into the next.

4.4.11.1 Common Pitfalls Across the Cycle

Many of the visualization failures discussed across this block can be traced back to a single stage of the cycle being neglected or rushed. Table 4.4.3 links several commonly observed pitfalls to the stage of the cycle in which they typically originate, reinforcing why treating visualization as a complete cycle, rather than as a single act of chart-making, helps prevent them.

Table 4.4.3 Links several commonly observed pitfalls to the relevant stages of the cycle

Common Pitfall	Stage Where It Originates	How the Cycle Helps Avoid It
Visualization answers the wrong question	Understand	Clarifying audience needs before designing
Misleading trends due to unclean data	Prepare	Systematic data cleaning and validation
Precise values hard to compare accurately	Visualize	Preferring position or length over colour or area
Interactive controls with no clear purpose	Interact	Designing interactivity around specific questions

SGOU

Audience misses the intended insight	Communicate	Deliberate storytelling and annotation
Flaws only discovered after wide release	Evaluate	Early and repeated testing with real viewers

4.4.12 An Illustrative Example

Consider an analyst asked to help a regional sales team understand why revenue has declined over the past two quarters. In the understand stage, the analyst would clarify that the audience consists of non-technical sales managers who need to identify which specific products or territories are driving the decline, rather than a detailed statistical explanation. In the prepare stage, the analyst would clean the sales transaction data, correct any inconsistent product or region naming, and aggregate individual transactions into monthly totals by product and territory.

In the visualize stage, the analyst might choose a line chart to show the overall revenue trend over time, alongside a bar chart comparing the most recent quarter's performance across territories, favouring position and length as the primary visual encodings for these quantitative comparisons. Publishing this as an interactive dashboard, as covered in Unit 3, would allow sales managers to filter by their own territory during the interact stage, while the analyst's own review during the interpret stage might reveal that the decline is concentrated in a single product category rather than being spread evenly.

During the communicate stage, the analyst would present this specific finding, the concentrated decline in one product category, directly and clearly, rather than simply handing over the dashboard without commentary. Finally, in the evaluate stage, feedback from the sales managers might reveal that they also need to see the decline broken down by individual sales representative, prompting the analyst to loop back to the prepare and visualize stages to extend the dashboard accordingly. This example illustrates how the seven stages of the cycle interact in practice, and why the process is more accurately described as a loop than a straight line.

It is also worth observing what this example demonstrates about the relationship between the two models discussed earlier in this unit. Each time the analyst revisits the prepare or visualize stage of the practical cycle, the underlying reference pipeline described in section 4.4.2 is effectively re-executed: new data transformations are applied to incorporate the sales representative field, a new visual mapping is chosen to represent it, and a new view is rendered for the sales managers to interact with. The practical cycle, in other words, is not a separate process running alongside the reference pipeline, but the human decision-making that determines when and why the pipeline is run again.

SGOU

Recap

- The data visualization reference pipeline describes how raw data is progressively converted into a rendered view through data transformations, visual mappings, and view transformations, with human interaction able to feed back into every stage.
- The practical data visualization cycle describes seven interconnected human activities: understand, prepare, visualize, interact, interpret, communicate, and evaluate.
- The understand stage clarifies both the underlying data and the intended audience before any visual design decisions are made.
- The prepare stage cleans and structures data, directly constraining what can later be visualized.
- The visualize stage selects visual encodings and design choices, favouring channels such as position and length for accurate quantitative comparison.
- The interact, interpret, and communicate stages transform a rendered visualization into genuine insight and, ultimately, into a shared understanding among its audience.
- The evaluate stage assesses whether a visualization achieved its purpose and commonly loops back to an earlier stage of the cycle, making the process genuinely iterative.

Objective Type Questions

1. Who proposed the influential data visualization reference pipeline model discussed in this unit?
2. What are the four principal stages of the data visualization reference pipeline?
3. Which transformation step converts raw data into a structured data table?
4. Which transformation step converts a data table into a visual structure?
5. Which transformation step converts a visual structure into a rendered view?
6. What role does human interaction play within the reference pipeline?
7. How many stages make up the practical data visualization cycle described in this unit?
8. Which stage of the practical cycle focuses on clarifying both the data and the intended audience?

SGOU

9. Which stage of the practical cycle most closely corresponds to the data transformations step of the reference pipeline?
10. Which visual encodings are generally considered most accurate for quantitative comparison?
11. What term describes arranging visualizations and narrative to guide an audience toward a conclusion?
12. What kind of activity typically occurs during the evaluate stage of the cycle?
13. Why is the data visualization cycle described as a loop rather than a straight line?
14. What might prompt a return to the prepare stage after evaluation?
15. What might prompt a return to the visualize stage after evaluation?

Answers

1. Card, Mackinlay, and Shneiderman
2. Raw data, data tables, visual structures, and views
3. Data transformations
4. Visual mappings
5. View transformations
6. It allows a viewer's actions to feed back into and modify any stage of the pipeline
7. Seven
8. Understand
9. Prepare
10. Position and length
11. Data storytelling
12. Gathering feedback and assessing whether the visualization achieved its purpose
13. Because feedback from evaluation commonly leads back to an earlier stage, prompting revision rather than a final, one-off output
14. The discovery of a data quality issue
15. Feedback that the chosen visual encoding proved confusing to the audience

SGOU

Assignments

1. Describe the data visualization reference pipeline and explain the purpose of each of its three transformation steps.
2. Explain the seven stages of the practical data visualization cycle and describe how they relate to one another.
3. Discuss why the understand stage is placed at the beginning of the visualization cycle and explain the risks of skipping it.
4. Explain why certain visual encodings, such as position and length, are generally preferred over others, such as colour intensity or area, for quantitative comparison tasks.
5. Using a hypothetical scenario of your choice, trace how a visualization project would move through all seven stages of the practical cycle, including at least one instance of looping back to an earlier stage.

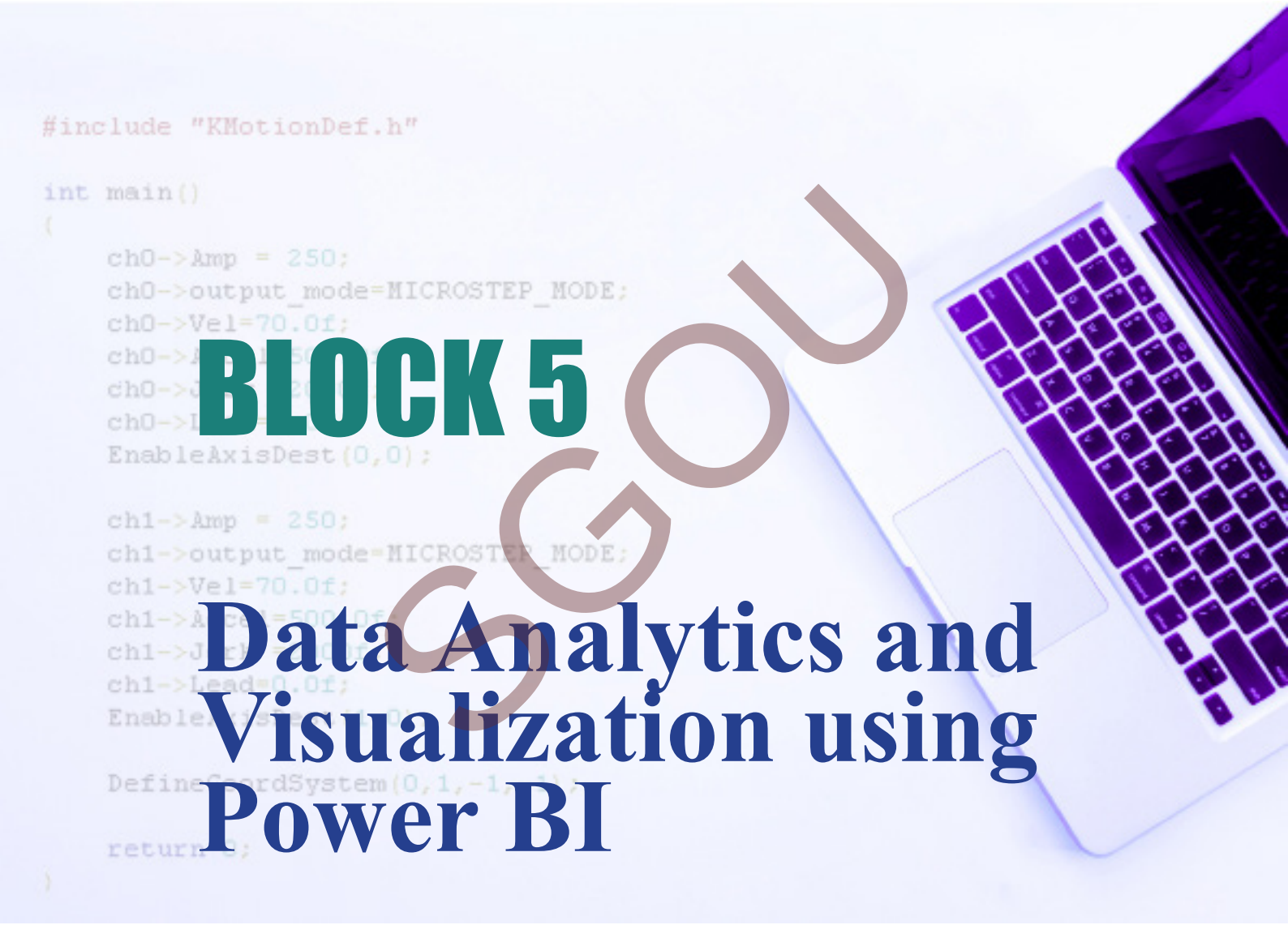
Suggested Reading

1. Campbell, A. (2020). Data Visualization: Clear Introduction to Data Visualization with Python. Proper Guide for Data Scientist.
2. Milligan, J. N. (2020). Learning Tableau 2020: Create effective data visualizations, build interactive visual analytics, and transform your organization (4th ed.). Packt Publishing.
3. Ahmed, M., & Pathan, A. K. (2020). Data analytics concepts, techniques and applications (1st ed.). Taylor and Francis.

Reference

1. Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann.
2. Campbell, A. (2020). Data Visualization: Clear Introduction to Data Visualization with Python.
3. Original illustrations created by the content writer for this Self-Learning Material (SLM).

SGOU



BLOCK 5

Data Analytics and Visualization using Power BI

SGOU



Power BI Basics and Data Preparation

Learning Outcomes

After completion of this unit, learner will be able to:

- explain the concept, purpose, and evolution of Power BI as a self-service business intelligence platform
- distinguish between the different versions of Power BI and identify suitable use cases for each
- describe the architecture and components of the Power BI ecosystem
- identify appropriate data connectors for different categories of data sources
- differentiate between Import, DirectQuery, and Live Connection modes of data loading
- apply data cleaning techniques to handle missing values, duplicates, and inconsistent data
- apply data transformation techniques such as merging, appending, pivoting, and grouping using Power Query Editor

SGOU

Prerequisite

Organizations today collect data from numerous sources - spreadsheets maintained by different departments, transactional databases, web services, and cloud applications. Before this data can be turned into insight, it must be brought together, examined for quality issues, and reshaped into a form suitable for analysis. Business intelligence tools such as Power BI were built precisely to simplify this journey for analysts who may not have a programming background.

To engage meaningfully with this unit, learners should be comfortable with the general notion of rows, columns, and tables, and should recall the discussion on data types, structured and unstructured data, and the data analytics life cycle from earlier blocks. This foundation will help learners understand why a dedicated stage for connecting, loading, cleaning, and transforming data is essential before any visualization or modelling can begin.

Keywords

Power BI, Business Intelligence, Power BI Desktop, Power BI Service, Data Connectors, Import Mode, DirectQuery, Live Connection, Power Query Editor, Data Cleaning, Data Transformation, M Language.

Discussion

5.1.1 Introduction to Power BI

Power BI is a self-service business intelligence and data visualization tool developed by Microsoft. It enables users to connect to a wide variety of data sources, prepare and transform raw data, build analytical data models, and present the results through interactive reports and dashboards. Power BI was designed to make advanced data analysis accessible not only to trained data analysts but also to business users who possess limited technical or programming expertise.

The core philosophy behind Power BI is to shorten the distance between raw data and actionable insight. Traditionally, generating a business report required a chain of specialists - database administrators to extract data, analysts to clean and process it, and designers to visualize it. Power BI consolidates much of this pipeline into a single integrated environment, allowing a single user to move from a spreadsheet or database table to a polished, interactive report within the same application.

Power BI is part of a broader movement in the analytics industry commonly referred to as self-service business intelligence, in which the responsibility for report creation shifts from centralized IT teams to individual business users. This shift has been made possible by intuitive drag-and-drop interfaces, built-in data preparation tools, and a formula language (DAX) that resembles spreadsheet functions familiar to most business users.

At a broad level, working with Power BI involves three recurring activities: connecting to and preparing data, modelling the data by defining relationships and calculations, and visualizing the data through charts, tables, and dashboards. This unit focuses on the first of these activities - understanding the Power BI platform itself and preparing data before it is modelled and visualized in subsequent units.

Power BI originated from a set of Excel add-ins - Power Query, Power Pivot, and Power View - that were developed by Microsoft to extend spreadsheet-based analysis with more powerful data connection, modelling, and visualization capabilities. As the demand for a standalone, more scalable analytics product grew, Microsoft combined these capabilities into Power BI Desktop and the associated Power BI Service, released as a distinct product. This lineage explains why many Power BI concepts, such as queries, measures, and pivot-style summarization, feel familiar to long-time spreadsheet users, even though Power BI itself is a considerably more capable platform intended for larger and more complex datasets.

SGOU

Since its release, Power BI has grown into one of the most widely adopted business intelligence platforms in the industry, in part because of its close integration with other Microsoft products such as Excel, SharePoint, and Azure, and in part because of the continuous addition of new data connectors and visualization capabilities. For learners preparing to work in data-driven roles, developing practical fluency with Power BI provides a transferable skill set, since the underlying concepts of connecting, cleaning, modelling, and visualizing data are shared by most modern business intelligence tools.

5.1.2 Power BI Versions / Editions

Power BI is offered in several editions, each designed to serve different stages of the analytics workflow, from individual report authoring to enterprise-wide distribution. Understanding these editions helps an organization choose the right combination of tools for its scale and budget.

5.1.2.1 Power BI Desktop

Power BI Desktop is a free, Windows-based application used primarily for authoring. It is where data connections are established, data is cleaned and transformed using Power Query Editor, data models are built, and visual reports are designed. Power BI Desktop is typically the starting point for any Power BI project and is installed locally on the analyst's computer.

5.1.2.2 Power BI Pro

Power BI Pro is a subscription-based, per-user license that enables report publishing, sharing, and collaboration through the Power BI Service (the cloud-based online platform). Pro users can create workspaces, share dashboards with colleagues who also hold a Pro license, and refresh datasets on a scheduled basis.

5.1.2.3 Power BI Premium

Power BI Premium provides dedicated cloud capacity for an organization rather than licensing individual users. It offers larger dataset storage limits, more frequent data refreshes, and the ability to share reports with users who do not themselves hold a Pro license, making it suitable for large enterprises with wide report distribution needs.

5.1.2.4 Power BI Report Server

Power BI Report Server is an on-premises reporting platform, allowing organizations that cannot or do not wish to store data in the cloud to host and share Power BI reports within their own infrastructure. It is commonly used by organizations with strict data residency or regulatory requirements.

5.1.2.5 Power BI Mobile

Power BI Mobile is an application available for Windows, iOS, and Android devices, allowing users to view and interact with published reports and dashboards while away from their desktop. It supports notifications and alerts when key metrics change.

SGOU

5.1.2.6 Power BI Embedded

Power BI Embedded allows independent software vendors and application developers to embed interactive Power BI reports and dashboards directly inside their own applications, so that end users can view analytics without needing a separate Power BI license.

Table 5.1.1 summarizes these editions and their principal use cases.

Edition	Deployment	Primary Use Case
Power BI Desktop	Local (Windows)	Authoring reports, data preparation and modelling
Power BI Pro	Cloud (per user)	Publishing, sharing, and collaboration
Power BI Premium	Cloud (capacity based)	Large-scale enterprise distribution
Power BI Report Server	On-premises	Regulated, in-house report hosting
Power BI Mobile	Mobile devices	Viewing reports and alerts on the go
Power BI Embedded	Cloud (developer)	Embedding analytics inside applications

5.1.3 Power BI Architecture and Components

The Power BI ecosystem is built around three principal components that work together to move data from its original source to the end viewer: Power BI Desktop, Power BI Service, and Power BI Mobile. As shown in Figure 5.1.1, these components are connected by a typical workflow of connecting, publishing, and accessing content.

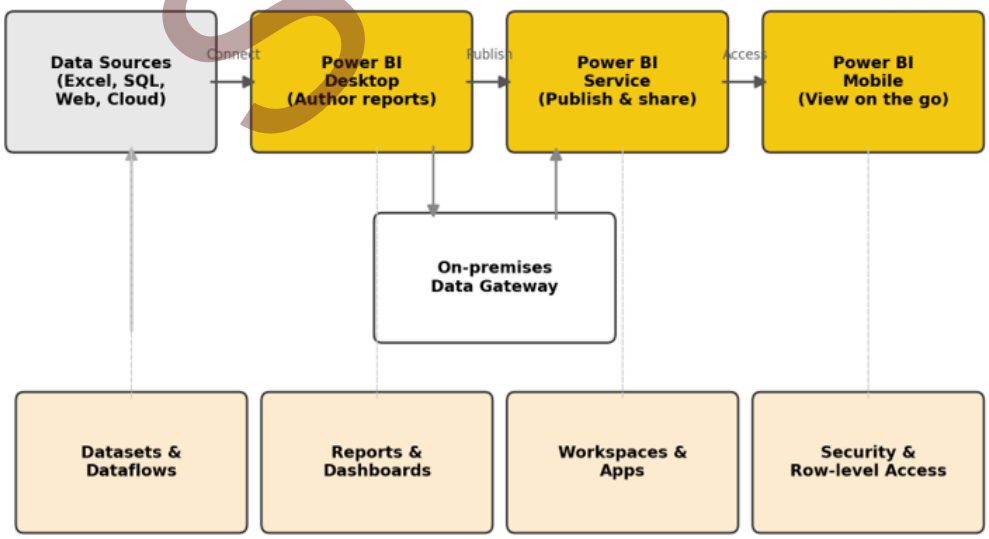


Fig. 5.1.1 Components of the Power BI Ecosystem

SGOU

As illustrated in Figure 5.1.1, an analyst begins by connecting to data sources using Power BI Desktop and authoring a report. This report is then published to the Power BI Service, a cloud-based platform where it can be organized into workspaces, combined with other reports into apps, and shared with colleagues. End users can then access these dashboards and reports either through a web browser or through the Power BI Mobile application.

Where the underlying data resides within an organization's internal network rather than the cloud, an On-premises Data Gateway acts as a secure bridge, allowing the Power BI Service to refresh datasets without exposing internal servers directly to the internet. The gateway is a critical component for organizations that keep sensitive data behind a corporate firewall while still wanting to benefit from cloud-based report distribution.

Within the Power BI Service, content is organized using datasets and dataflows (the prepared data), reports and dashboards (the visual output), and workspaces and apps (collections used for collaboration and distribution). Security is enforced through role-based access and row-level security, ensuring that different users see only the data they are authorized to view.

5.1.4 Data Connectors in Power BI

A data connector is the mechanism through which Power BI establishes a link with an external data source in order to retrieve data. Power BI provides an extensive library of built-in connectors, allowing it to draw data from files stored on a local computer, relational and non-relational databases, and a wide range of online and cloud-based services. This broad connectivity is one of the primary reasons Power BI has become popular across industries, as it removes the need to manually export and re-import data between systems.

Data connectors in Power BI can broadly be grouped into the following categories:

Table 5.1.2 Categories of Data Connectors in Power BI

Category	Description	Examples
File-based Sources	Data stored in files on a local machine or shared drive	Excel workbooks, CSV, XML, JSON, PDF, text files
Database Sources	Data stored in relational or NoSQL database systems	SQL Server, MySQL, Oracle, PostgreSQL, MongoDB
Online Services	Data hosted by cloud-based business applications	SharePoint Online, Dynamics 365, Google Analytics, Salesforce

SGOU

Azure Services	Data hosted on Microsoft's cloud computing platform	Azure SQL Database, Azure Synapse Analytics, Azure Data Lake
Web and Other Sources	Data retrieved directly from web pages or APIs	Web pages, OData feeds, ODBC/OLE DB connections

Regardless of the category, every connector ultimately performs the same basic function: it authenticates with the source (where required) and retrieves data in a form that Power BI can read into a query. The choice of connector depends entirely on where an organization's data resides, and it is common for a single Power BI report to combine data pulled from several different connector categories at once.

5.1.5 Connecting to Data Sources - The Get Data Experience

Power BI Desktop provides a consistent entry point for establishing any data connection, referred to as the Get Data experience. Although the underlying source may differ widely, from a spreadsheet to a cloud data warehouse, the general sequence of steps followed by the analyst remains largely the same:

1. **Select the source category:** The analyst chooses the appropriate connector, for example, Excel workbook, SQL Server database, or Web.
2. **Provide connection details:** Depending on the source, this may involve entering a file path, a server and database name, or a web address.
3. **Authenticate:** Power BI may prompt for credentials such as a Windows account, database username and password, an organizational account, or an API key, depending on the security configuration of the source.
4. **Preview the data:** A Navigator window displays the available tables, files, or worksheets, allowing the analyst to preview their contents before importing them.
5. **Choose to Load or Transform:** The analyst may load the data directly into the model, or choose Transform Data to open Power Query Editor and prepare the data before loading.

This consistent workflow means that once a learner is comfortable connecting to one type of source, connecting to a new and unfamiliar source generally follows a familiar pattern, differing mainly in the connection details and authentication method required.

Authentication deserves particular attention, since different sources support different mechanisms depending on how they are secured. Common authentication options in Power BI include a Windows account (using the credentials of the user currently logged into the computer), a database username and password configured specifically for that data source, an organizational account associated with a cloud service such as Microsoft 365, and an API key or access token used by many web-based services.

SGOU

Selecting the correct authentication method, and ensuring that the account used has adequate permission to read the required data, is often the first troubleshooting step when a new connection fails.

5.1.6 Data Loading in Power BI

Once a connection to a data source has been established, Power BI offers different modes for how the data is actually loaded and kept up to date within a report. The choice of loading mode has a significant impact on report performance, data freshness, and the size of the resulting Power BI file, and should be made deliberately based on the nature of the data source and the reporting requirement.

5.1.6.1 Import Mode

In Import mode, a copy of the data is extracted from the source and compressed into Power BI's own internal storage engine. Once imported, the report interacts only with this local copy, which allows for extremely fast query performance, since calculations do not need to travel back to the original source each time a visual is refreshed. The trade-off is that the data reflects the state of the source only as of the last refresh, and very large datasets may be constrained by available memory.

5.1.6.2 DirectQuery Mode

In DirectQuery mode, no data is imported into Power BI. Instead, every visual sends a live query back to the underlying database whenever the report is viewed or interacted with. This ensures that users always see the most current data and allows Power BI to work with datasets far larger than could be imported, but it depends heavily on the performance of the source system and the speed of the underlying network connection.

5.1.6.3 Live Connection

Live Connection mode is similar in spirit to DirectQuery but is used specifically to connect to an existing analytical model, such as a model already published to Power BI Service or hosted on SQL Server Analysis Services. Rather than reimporting or redefining data relationships, the report simply reuses the pre-built model, which promotes consistency when several reports need to draw on the same underlying analytical definitions.

Table 5.1.3 Comparison of Power BI Storage Modes

Aspect	Import Mode	DirectQuery Mode	Live Connection
Data location	Copied into Power BI	Remains in source	Remains in external model
Data freshness	As of last refresh	Always current	Always current

SGOU

Performance	Very fast	Depends on source	Depends on source
Dataset size limit	Limited by memory	Very large datasets supported	Determined by external model
Transformations	Fully supported	Limited	Not supported (model reused as-is)

5.1.6.4 Choosing the Appropriate Loading Mode

In practice, the choice of loading mode is guided by a small number of questions. If the dataset is of moderate size and does not need to reflect changes made to the source within the last few minutes, Import mode is usually preferred for its speed and its full support for Power Query transformations. If the dataset is extremely large, or if regulatory or business requirements demand that users always see live figures, DirectQuery becomes necessary, with the understanding that report performance will then depend on the responsiveness of the source system. Live Connection is chosen not so much for performance reasons as for governance reasons, ensuring that multiple reports across an organization remain consistent by relying on a single, centrally maintained data model.

5.1.7 Introduction to Power Query Editor

Power Query Editor is the dedicated environment within Power BI Desktop where all data cleaning and transformation activities take place, before the data is loaded into the report's data model. Every action a user performs in Power Query Editor, such as removing a column, filtering rows, or changing a data type, is recorded as a discrete, reorderable step in a list called Applied Steps. Collectively, these recorded steps form a query, and behind the scenes, Power BI translates each query into a script written in the M language (also known as Power Query Formula Language).

This step-based design offers two major advantages. First, it is non-destructive: the original data source is never altered, since Power BI only reads and reshapes a working copy. Second, it is repeatable: because every transformation is recorded as a step rather than applied manually and permanently, the entire sequence of cleaning and transformation operations is automatically re-executed each time the data is refreshed from its source, without the analyst needing to repeat any manual work.

As shown in Figure 5.1.2, data preparation in Power BI generally follows a five-stage workflow: connecting to the source, loading the data into Power Query, cleaning it to remove errors and inconsistencies, transforming it into an analysis-friendly shape, and finally handing it over to the data model, where relationships and calculations are defined.

SGOU

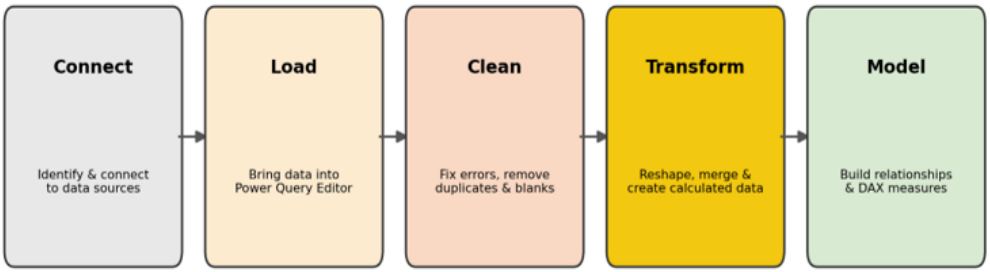


Fig. 5.1.2 Data Preparation Workflow in Power BI

5.1.8 Data Cleaning in Power BI

Data collected from real-world sources is rarely ready for immediate analysis. It often contains missing values, duplicate records, inconsistent formatting, or incorrect data types, any of which can distort calculations and mislead viewers of a report. Data cleaning refers to the set of operations performed to detect and correct such quality issues before the data reaches the reporting layer. Power Query Editor provides several built-in tools to support this process.

5.1.8.1 Handling Missing Values

Power Query allows blank or null values to be identified, removed, replaced with a default value, or filled using the value from a preceding row through the Fill Down option. The appropriate technique depends on why the value is missing and what impact it would have on subsequent analysis.

5.1.8.2 Removing Duplicate Records

Duplicate rows, which can occur due to repeated data entry or the combination of overlapping data sources, are removed using the Remove Duplicates option, ensuring that summary calculations such as counts and totals are not inflated.

5.1.8.3 Correcting Data Types

Every column in Power Query is assigned a data type, such as whole number, decimal number, text, date, or true/false. Incorrect data types, for example, a date stored as text, can prevent Power BI from performing calculations or sorting correctly, so verifying and correcting column data types is one of the earliest steps in any cleaning process.

5.1.8.4 Handling Errors

Some transformations, such as converting a non-numeric text value into a number, can produce errors within specific cells. Power Query flags such cells and provides options to remove rows with errors, replace erroneous values, or keep only the rows containing errors for closer inspection.

SGOU

5.1.8.5 Renaming and Organizing Columns

Columns imported from a source often carry technical or inconsistent names. Renaming columns to clear, business-friendly labels, removing unnecessary columns, and reordering columns logically all contribute to a dataset that is easier for both the analyst and the eventual report viewer to understand.

Table 5.1.4 summarizes some of the most frequently encountered data quality issues alongside the Power Query tool typically used to resolve each one.

Table 5.1.4 Common Data Quality Issues and Power Query Remediation Actions

Data Quality Issue	Typical Cause	Power Query Tool / Action
Blank or missing values	Incomplete data entry or export	Remove Blank Rows, Replace Values, Fill Down
Duplicate records	Repeated imports or merged sources	Remove Duplicates
Incorrect data type	Source stores numbers or dates as text	Change Type / Detect Data Type
Inconsistent text casing or spacing	Manual data entry by multiple users	Trim, Clean, Capitalize Each Word
Calculation errors after conversion	Non-numeric characters within a numeric field	Remove Errors, Replace Errors

5.1.9 Data Transformations in Power BI

While data cleaning focuses on correcting quality issues, data transformation focuses on reshaping otherwise correct data into a structure better suited for analysis and modelling. Power Query Editor offers a wide range of transformation operations, some of the most commonly used of which are described below.

5.1.9.1 Splitting and Merging Columns

A single column containing combined information, such as a full name or an address, can be split into multiple columns using a delimiter or a fixed number of characters. Conversely, multiple columns can be merged into one, which is useful when creating a single combined key or label.

5.1.9.2 Pivoting and Unpivoting

Pivoting converts unique values from a column into new columns, which is useful for producing summary-style layouts. Unpivoting performs the reverse operation,

SGOU

converting multiple columns into rows against a common attribute column, and is frequently required when source data (such as a spreadsheet with a separate column for each month) needs to be reshaped into a long, analysis-friendly format.

5.1.9.3 Grouping and Aggregating

The Group By operation allows rows to be grouped based on one or more columns, with an aggregate calculation, such as sum, average, count, minimum, or maximum, applied to another column. This is commonly used to summarize detailed transactional data into higher-level figures, such as total sales by region.

5.1.9.4 Merging and Appending Queries

Merging queries combines two queries based on matching values in one or more columns, similar to a join operation in a relational database, and is used to bring related information from two different tables into a single query. Appending queries, in contrast, stacks the rows of two or more queries with matching columns on top of one another, and is used when combining data of the same structure from multiple periods or sources.

5.1.9.5 Custom Columns and the M Language

Where the built-in transformation options are not sufficient, a custom column can be created using an expression written in the M language, Power Query's underlying formula language. Custom columns allow analysts to derive new values, such as calculating a profit margin from existing revenue and cost columns, directly within the data preparation stage.

For example, an analyst wishing to calculate the total value of each sales transaction from separate Quantity and UnitPrice columns would add a custom column with an expression similar to [Quantity] * [UnitPrice]. Although many such calculations can equally be performed later using DAX measures within the data model (as covered in Unit 2 of this block), performing straightforward, row-level calculations at the Power Query stage keeps the data model simpler and ensures that the calculated values are already present when the table is loaded.

5.1.10 Best Practices for Data Preparation in Power BI

Effective data preparation goes beyond simply applying the tools described above. A few practices are widely recommended to ensure that Power BI reports remain accurate, efficient, and maintainable over time:

- Remove unused columns and rows as early as possible in the query, since this reduces the volume of data carried through every subsequent step and improves refresh performance.
- Give each Applied Step a clear, descriptive name so that the transformation logic remains understandable when the query is revisited later or reviewed by another analyst.

SGOU

- Verify column data types immediately after connecting to a new source, before applying further transformations, to avoid compounding errors.
- Prefer built-in transformation options over complex custom M code wherever possible, as this keeps queries easier to maintain and less prone to error.
- Document the source and purpose of each query, particularly in reports that combine data from several different systems.

Taken together, these practices help ensure that a Power BI report continues to function correctly and refresh efficiently even as the underlying data grows in volume or as the report is handed over to another analyst for future maintenance. Since the query steps created during data preparation form the backbone on which every subsequent visual and calculation in the report depends, time invested at this stage is rarely wasted.

5.1.11 An Illustrative Example

Consider a small retail business that records daily sales in separate Excel workbooks maintained by each of its regional branches. To build a consolidated sales report in Power BI, an analyst would typically proceed as follows. First, the Excel connector is used to connect to each branch's workbook, and the individual sales tables are appended into a single combined query, since all branches use the same column structure. Next, the combined data is examined for missing values in the quantity or price columns, duplicate transaction entries arising from repeated exports, and any date column mistakenly imported as text.

Once the data has been cleaned, a custom column may be added to calculate the total value of each transaction by multiplying quantity and unit price, and the transaction data may be grouped by branch and month to produce a summary table of monthly sales per branch. Only after these steps are complete would the analyst proceed to build relationships between this sales table and supporting tables, such as a product table or a calendar table, in preparation for the modelling activities covered in the next unit.

This example illustrates a general principle that applies across almost any Power BI project: the quality of the eventual report depends directly on the rigour applied during the connection, loading, cleaning, and transformation stages, making this unit's content foundational to everything that follows in Block V.

Recap

- Power BI is a self-service business intelligence tool that allows users to connect to, prepare, model, and visualize data within a single integrated environment.
- Power BI is available in several editions - Desktop, Pro, Premium, Report Server, Mobile, and Embedded - each suited to different stages of authoring, sharing, or embedding analytics.

SGOU

- The Power BI ecosystem is built around Power BI Desktop, Power BI Service, and Power BI Mobile, linked together through publishing and, where required, an on-premises data gateway.
- Data connectors allow Power BI to draw data from file-based sources, databases, online services, Azure services, and the web.
- Data can be loaded using Import mode, DirectQuery mode, or Live Connection, each offering a different balance between performance and data freshness.
- Power Query Editor provides a non-destructive, step-based environment for cleaning and transforming data before it enters the data model.
- Data cleaning addresses missing values, duplicates, incorrect data types, and errors, while data transformation reshapes data through operations such as splitting, pivoting, grouping, merging, and appending.

Objective Type Questions

1. What is Power BI primarily used for?
2. Which Power BI edition is installed locally and used for authoring reports?
3. Which Power BI edition provides dedicated cloud capacity for an organization?
4. Which component allows on-premises data to be refreshed by the Power BI Service?
5. What is a data connector in Power BI?
6. Which category of connector would be used to access an Excel workbook?
7. What is the Get Data experience used for?
8. Which data loading mode stores a copy of data inside Power BI?
9. Which data loading mode always shows the most current data by querying the source directly?
10. What is the underlying formula language used by Power Query called?
11. What term describes the recorded list of transformation actions in Power Query?
12. Which Power Query option is used to remove repeated rows from a dataset?
13. What operation converts columns into rows in Power Query?

SGOU

14. Which operation combines two queries based on matching column values?
15. What is the main purpose of data cleaning before analysis?

Answers

1. Connecting to, preparing, modelling, and visualizing data
2. Power BI Desktop
3. Power BI Premium
4. On-premises Data Gateway
5. A mechanism used to connect Power BI to an external data source
6. File-based sources
7. Establishing a connection to a data source
8. Import mode
9. DirectQuery mode
10. M language (Power Query Formula Language)
11. Applied Steps
12. Remove Duplicates
13. Unpivoting
14. Merging queries
15. To detect and correct data quality issues such as missing values, duplicates, and errors

SGOU

Assignments

1. Describe the different versions of Power BI and discuss the type of organization or user best suited to each.
2. Explain the architecture of the Power BI ecosystem with reference to its principal components.
3. Compare Import mode, DirectQuery mode, and Live Connection as methods of loading data into Power BI, highlighting the advantages and limitations of each.

4. Discuss the role of Power Query Editor in data preparation and explain why its step-based approach is described as non-destructive.

5. Explain any three data cleaning operations and any three data transformation operations available in Power Query Editor, with suitable examples.

Suggested Reading

1. Powell, B. (2018). Mastering Microsoft Power BI: Expert techniques for effective data analytics and business intelligence. Packt Publishing.

2. Milligan, J. N. (2020). Learning Tableau 2020: Create effective data visualizations, build interactive visual analytics, and transform your organization (4th ed.). Packt Publishing.

3. Ahmed, M., & Pathan, A. K. (2020). Data analytics concepts, techniques and applications (1st ed.). Taylor and Francis.

Reference

1. Microsoft Corporation. Power BI Documentation.

2. Microsoft Corporation. Power Query M Formula Language Reference.

SGOU



Data Modelling and DAX in Power BI

Learning Outcomes

After completion of this unit, learner will be able to:

- explain the role of data modelling in transforming prepared data into an analysis-ready structure
- describe the concepts of tables, relationships, cardinality, and cross-filter direction in Power BI
- differentiate between star schema and snowflake schema designs and identify fact and dimension tables
- create and manage relationships between tables within the Power BI data model
- distinguish between calculated columns and measures and identify appropriate situations for each
- write basic DAX expressions using common aggregation, logical, and date functions
- apply the CALCULATE function to modify filter context within a DAX expression

SGOU

Prerequisite

Unit 1 of this block introduced the process of connecting to data sources and preparing data using Power Query Editor, covering data loading, cleaning, and transformation. Once data has been cleaned and reshaped, it must be organized into a coherent structure that Power BI can use to answer analytical questions efficiently. This organizing activity is known as data modelling, and it forms the foundation on which every report, chart, and calculation in a Power BI file ultimately depends.

To engage meaningfully with this unit, learners should be comfortable with the outcomes of Unit 1, particularly the idea that a Power BI file may contain multiple queries loaded from different sources. Learners should also recall the basic notion of a relational database, in which separate tables are linked together through common

key columns, since Power BI's data model is built on very similar principles. A working familiarity with spreadsheet formulas will also help learners relate to the DAX expressions introduced later in this unit.

Keywords

Data Model, Relationships, Cardinality, Cross-filter Direction, Star Schema, Snowflake Schema, Fact Table, Dimension Table, Calculated Column, Measure, DAX, CALCULATE, Filter Context, Row Context.

Discussion

5.2.1 Introduction to Data Modelling in Power BI

Data modelling is the process of organizing multiple tables of data and defining how they relate to one another, so that Power BI can combine information from different tables correctly when a report is built. A Power BI report is rarely powered by a single flat table; more commonly, it draws on several related tables, for example, a table of sales transactions, a table describing products, a table describing customers, and a table representing dates. Without a properly defined model, Power BI would have no way of knowing how a row in the sales table relates to a row in the product table.

The data model sits conceptually between the data preparation stage covered in Unit 1 and the visualization stage covered in Unit 3. Once queries have been cleaned and loaded from Power Query Editor into Power BI Desktop, the analyst moves to the Model view, where tables can be arranged, relationships drawn between them, and calculations defined. A well-designed data model makes report building in the next stage considerably simpler, since visuals can then draw fields from multiple tables as though they belonged to a single combined table.

Good data modelling practice also has a direct impact on report performance. A model organized according to well-established design principles, such as the star schema described later in this unit, generally performs faster and is easier to maintain than a model built from a small number of very wide, flat tables.

It is useful to think of data modelling as answering a specific kind of question on behalf of the business: given a particular fact, such as a sale, which descriptive attributes should be available to slice and summarize it by? A well-modelled dataset allows a business user to ask, for instance, what were total sales by region last quarter, and receive a correct answer instantly, because the relationships needed to connect the sales figures to region and date information have already been defined correctly within the model. A poorly modelled dataset, by contrast, may produce numbers that look plausible but are subtly incorrect, for example, double-counting sales because of an unintended many-to-many relationship.

SGOU

5.2.2 From Power Query to the Data Model

As discussed in Unit 1, each query prepared in Power Query Editor is loaded into Power BI Desktop as a table. When several such queries are loaded together, Power BI Desktop presents them in the Model view as a collection of separate tables, initially unconnected unless Power BI is able to automatically detect an obvious relationship based on matching column names.

It is generally good practice to complete the bulk of cleaning and shaping activity within Power Query before moving to the Model view, since transformations such as removing columns or correcting data types are more efficient when performed before the data is loaded. The Model view is then reserved primarily for defining relationships between tables and adding calculated columns and measures, which are described later in this unit.

5.2.3 Data Modelling Concepts - Tables, Relationships, and Cardinality

A relationship in Power BI is a defined link between a column in one table and a column in another table, typically based on a common key, such as a Product ID appearing in both a sales table and a product table. Once a relationship is created, Power BI is able to filter and summarize values across both tables automatically whenever a visual uses fields from each.

5.2.3.1 Cardinality

Cardinality describes the numerical relationship between matching rows in two related tables. Power BI supports several cardinality types:

Table 5.2.1 Cardinality Types in Data Modeling

Cardinality Type	Description	Typical Example
One-to-many	One row in the first table can relate to many rows in the second table	One product relates to many sales transactions
Many-to-one	The reverse of one-to-many, viewed from the other table's perspective	Many sales transactions relate to one product
One-to-one	Each row in one table relates to exactly one row in the other	One employee relates to one employee profile record
Many-to-many	Multiple rows in each table can relate to multiple rows in the other	Students and courses, where each student takes many courses

SGOU

5.2.3.2 Cross-filter Direction

Cross-filter direction determines how a filter applied to one table affects the other table in a relationship. A Single direction allows filters to flow only from the one side of a relationship to the many side, which is the most common and generally recommended setting. A Both direction allows filters to flow in either direction, which offers more analytical flexibility but can introduce ambiguity in more complex models and should be used selectively.

As a practical example, with a Single direction relationship between Dim_Product and Fact_Sales, selecting a product category in a slicer correctly filters the sales figures down to that category. However, a filter placed on the sales table, such as restricting to transactions above a certain value, would not, by default, flow back to filter the list of products shown in a separate visual, unless the relationship's cross-filter direction is explicitly set to Both.

5.2.4 Schema Design - Star Schema and Snowflake Schema

The way tables are arranged within a data model is referred to as its schema. Two schema designs are especially common in Power BI and in business intelligence more generally: the star schema and the snowflake schema.

5.2.4.1 Fact Tables and Dimension Tables

Before describing these schemas, it is necessary to distinguish between two categories of tables. A fact table stores measurable, transactional data, such as sales amounts, quantities, or transaction counts, and tends to be the largest table in the model, growing continuously as new transactions occur. A dimension table stores descriptive information used to filter, group, or label the facts, such as product names, customer details, store locations, or calendar dates, and is typically much smaller than the fact table it supports.

Table 5.2.2 Comparison of Fact and Dimension Tables

Characteristic	Fact Table	Dimension Table
Content	Numeric, measurable values	Descriptive, textual attributes
Row count	Very large, grows continuously	Comparatively small and stable
Typical columns	Amount, Quantity, foreign keys	Name, Category, Region, Description
Role in the model	The “many” side of relationships	The “one” side of relationships

SGOU

5.2.4.2 Star Schema

In a star schema, a single central fact table is connected directly to several surrounding dimension tables, with each dimension table related to the fact table through a one-to-many relationship. As illustrated in Figure 5.2.1, this arrangement resembles a star, with the fact table at the centre and dimension tables radiating outward.

As shown in Figure 5.2.1, the Fact_Sales table sits at the centre of the schema and connects to four dimension tables: Dim_Date, Dim_Product, Dim_Customer, and Dim_Store. Each dimension table supplies descriptive attributes that can be used to slice or group the sales figures, for example, viewing total sales by product category or by region, without those descriptive attributes needing to be repeated within the fact table itself.

The star schema is widely recommended for Power BI models because its simple, predictable relationship structure keeps calculations efficient and easy to reason about. Most Power BI modelling guidance encourages analysts to reshape their data, where necessary, into this pattern during the Power Query stage covered in Unit 1.

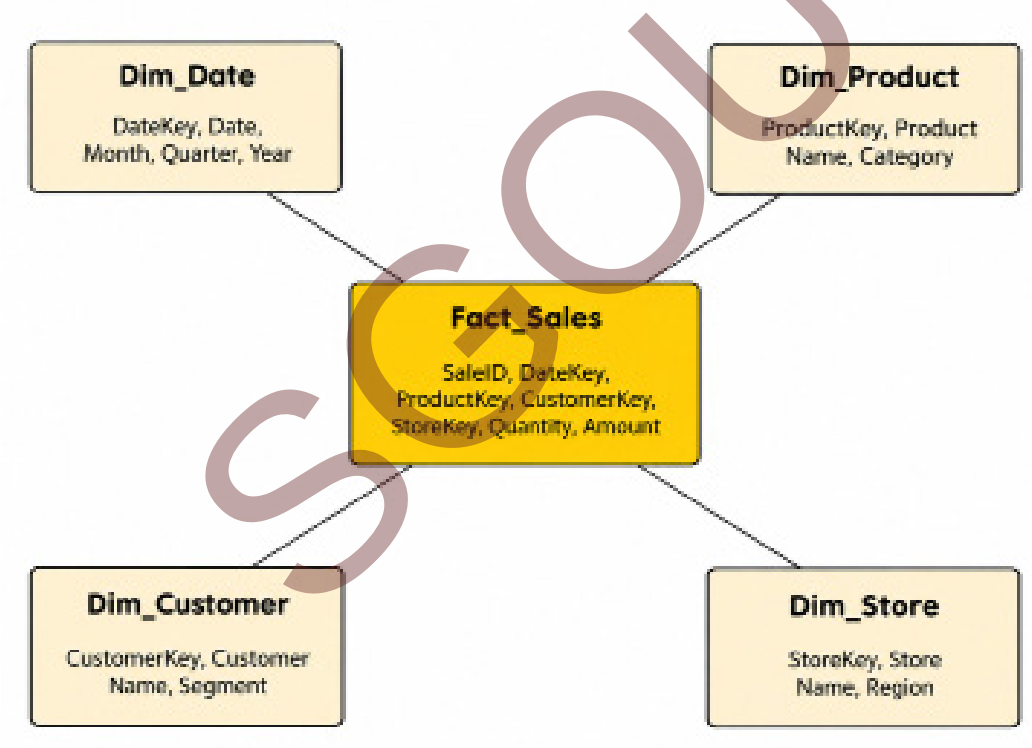


Fig. 5.2.1 Star Schema with a Central Fact Table and Surrounding Dimension Tables

5.2.4.3 Snowflake Schema

A snowflake schema extends the star schema by further splitting dimension tables into related sub-dimension tables. For example, rather than storing city, state, and country directly within a single Dim_Store table, a snowflake design might separate these into a linked Dim_Geography table. While this can reduce data duplication, it introduces additional relationships and generally increases the complexity of the model without

SGOU

a proportional performance benefit, which is why the simpler star schema is usually preferred in Power BI unless there is a specific reason to normalize the dimension tables further.

5.2.5 Creating and Managing Relationships in Power BI

Power BI Desktop provides a Model view in which relationships can be created either automatically or manually. When several tables containing similarly named key columns are loaded together, Power BI often detects and creates relationships automatically. However, it remains good practice for the analyst to review every automatically created relationship, since an incorrect automatic match can lead to misleading report results.

A relationship can also be created manually by dragging a column from one table onto the matching column in another table within the Model view diagram, or through the Manage Relationships dialog box, which allows the analyst to specify the two tables involved, the matching columns, the cardinality, and the cross-filter direction explicitly.

Each relationship is represented visually in the Model view as a line connecting the two related tables, with symbols indicating the cardinality (such as 1 and *) at each end. Reviewing this diagram view regularly helps the analyst confirm that the overall structure of the model still resembles the intended star schema as new tables are added over time.

5.2.5.1 Active and Inactive Relationships

Power BI allows only one active relationship (shown as a solid line) to exist between any two tables at a time, since an active relationship is the one used automatically by DAX calculations unless explicitly overridden. In some models, two tables may need to be related in more than one way, for example, a sales table containing both an order date and a delivery date, each of which could relate to a calendar table. In such cases, only one of the two relationships can be active at any given time, while the other is created as an inactive relationship (shown as a dashed line) and is used only when explicitly invoked from within a DAX expression using the USERELATIONSHIP function.

5.2.5.2 Hierarchies and Data Categories

Within a dimension table, related columns can be grouped into a hierarchy to support drilling from a general level down to a more detailed level, for example, a date hierarchy consisting of Year, Quarter, Month, and Day, or a geography hierarchy consisting of Country, State, and City. Power BI also allows individual columns to be assigned a data category, such as identifying a column as containing city names or geographic coordinates, which enables certain visuals, such as maps, to interpret and plot the column's values correctly.

5.2.6 Calculated Columns

A calculated column is a new column added to a table within the data model, whose values are computed using a DAX expression rather than being loaded directly from the data source. Once created, a calculated column behaves exactly like any other column

SGOU

in the table: it can be used in visuals, placed on rows or columns of a matrix, or used to filter and sort data.

The defining characteristic of a calculated column is that its expression is evaluated once for every row of the table, in the context of that row's own values, and the results are then stored permanently within the data model. This means calculated columns increase the file size of the Power BI report to some extent, since their computed values are stored physically alongside the imported data, in the same way as any other column.

Calculated columns are most appropriate when the result needs to be used for slicing, filtering, grouping, or sorting a visual, since these operations require an actual column of stored values, rather than a value that is computed only when a visual is displayed.

5.2.7 Measures

A measure is a calculation defined using a DAX expression that is evaluated dynamically at the moment a report is viewed, based on the filters, slicers, and visual context currently in effect. Unlike a calculated column, a measure does not add a new column of stored values to a table; instead, its result exists only for the duration of the calculation and changes automatically as the user interacts with the report.

For example, a measure that calculates Total Sales as the sum of an Amount column will return a different number depending on which product, region, or time period is currently selected in the report, without the analyst needing to write separate calculations for each possible combination of filters. This responsiveness to context is what makes measures the preferred approach for the great majority of numeric calculations in Power BI reports, particularly aggregations such as totals, averages, and ratios.

As illustrated in Figure 5.2.2, calculated columns and measures follow fundamentally different evaluation paths: a calculated column is computed row by row and stored in the table, while a measure is computed on demand and responds to the filter context of the surrounding visual.

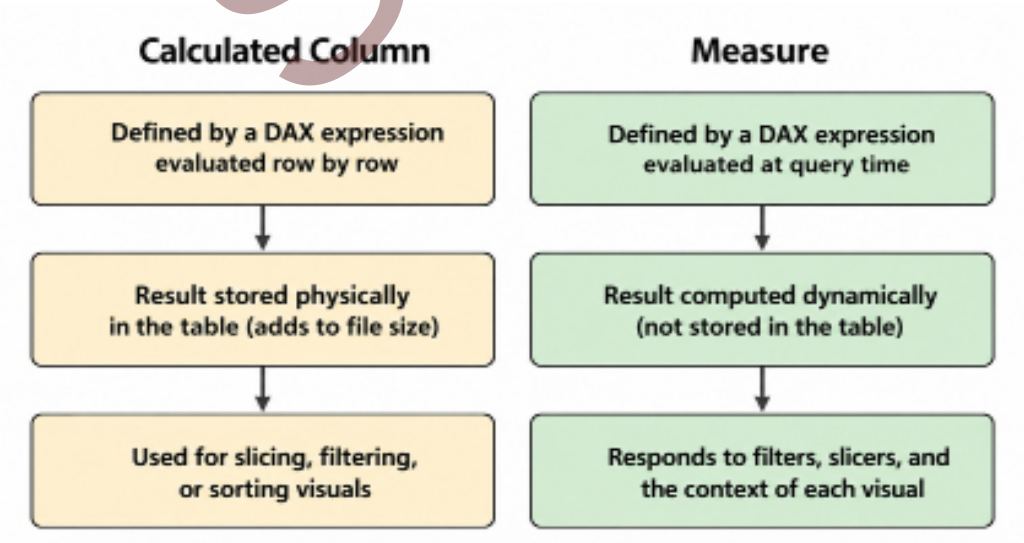


Fig.5.2.2 Calculated Column versus Measure - Evaluation and Storage Behaviour

SGOU

As shown in Figure 5.2.3, the choice between a calculated column and a measure is not simply a matter of preference; it is guided by how the resulting value will be used within the report. Table 5.2.3 summarizes the key distinctions between the two.

Table 5.2.3 Comparison of Calculated Columns and Measures

Aspect	Calculated Column	Measure
Evaluation time	Once, when data is refreshed	Dynamically, when a visual is rendered
Evaluation context	Row context	Filter context
Storage	Stored in the table (adds to file size)	Not stored; computed on demand
Typical use	Slicing, filtering, sorting, grouping	Totals, averages, ratios, KPIs
Responds to report filters	No (fixed once calculated)	Yes (recalculates with context)

5.2.8 Introduction to DAX (Data Analysis Expressions)

DAX, short for Data Analysis Expressions, is the formula language used throughout Power BI to define calculated columns, measures, and calculated tables. DAX was designed to feel familiar to users already comfortable with spreadsheet formulas, sharing many function names, such as SUM, AVERAGE, and IF, with Excel, while extending these with functions specifically built for working with related tables and filter context.

A DAX expression always evaluates to a single value or to a table, and every expression is written in the context of the data model, meaning it can refer not only to columns within its own table but, through relationships, to columns in related tables as well. This ability to reach across relationships is central to how DAX supports the kind of multi-table analysis that a well-designed star schema, as described earlier in this unit, is intended to enable.

5.2.8.1 Row Context and Filter Context

Two concepts are essential to understanding how DAX expressions behave: row context and filter context. Row context refers to the current row being evaluated, and applies primarily when a calculated column is being computed, since each row is evaluated individually. Filter context refers to the combination of filters, slicers, and visual fields currently applied when a measure is evaluated, and explains why the same measure can return different results in different parts of the same report.

SGOU

5.2.8.2 Basic DAX Syntax

A DAX expression typically references a column using the syntax TableName[ColumnName], and references an existing measure using square brackets alone, such as [Total Sales]. Expressions may combine columns, measures, constants, and functions using standard arithmetic operators (+, -, *, /) as well as comparison operators (=, >, <) within logical functions. Every measure and calculated column begins with a name followed by an equals sign, after which the DAX expression itself is written, in a style that will feel familiar to anyone who has written a formula in a spreadsheet cell.

5.2.9 Common DAX Functions

While DAX includes a very large library of functions, a relatively small set of function categories accounts for the majority of everyday Power BI calculations.

5.2.9.1 Aggregation Functions

Aggregation functions summarize values across a column or table, and include SUM, AVERAGE, MIN, MAX, and COUNT. For example, a Total Sales measure might be written as SUM(Fact_Sales[Amount]), which adds together the Amount column across whatever rows remain visible in the current filter context.

5.2.9.2 Logical Functions

Logical functions such as IF and SWITCH allow a calculation to return different results depending on a specified condition. These are commonly used to create categories, flags, or conditional labels, such as classifying a sale as High Value or Standard depending on the transaction amount.

5.2.9.3 Date and Time Functions

Date and time functions, including YEAR, MONTH, DATEDIFF, and a family of time intelligence functions such as TOTALYTD and SAMEPERIODLASTYEAR, allow calculations to be expressed in terms of calendar periods, enabling comparisons such as year-to-date sales or sales growth compared with the same period in the previous year.

5.2.9.4 The CALCULATE Function

CALCULATE is widely regarded as the most important function in DAX, because it allows the filter context of a calculation to be modified. CALCULATE takes an expression to evaluate, along with one or more filter conditions, and evaluates that expression within a new, modified filter context. For example, a measure to calculate sales specifically within a given product category, regardless of which category a viewer has currently selected in a report, would be written using CALCULATE together with a filter on the category column. Most advanced Power BI calculations, including comparisons across time periods or against a fixed target, rely on CALCULATE at their core.

SGOU

5.2.9.5 Table and Relationship Functions

Because a Power BI model consists of several related tables, DAX also provides functions that navigate across relationships. RELATED retrieves a value from a related table on the one side of a one-to-many relationship, for example, pulling a product's category into a row of the sales table. RELATEDTABLE returns all rows from a related table that correspond to the current row, and is typically used inside an aggregation function. DISTINCT returns a table of unique values from a column, and COUNTROWS counts the number of rows in a table or in the result of another table function. These functions allow a measure or calculated column to draw on information stored in a different table without that information needing to be duplicated or manually looked up.

Table 5.2.4 summarizes the main categories of DAX functions introduced in this unit, together with representative examples of each.

Table 5.2.4 Common DAX Function Categories and Example

Function Category	Purpose	Example Functions
Aggregation	Summarize values across a column or table	SUM, AVERAGE, MIN, MAX, COUNT
Logical	Return different results based on a condition	IF, SWITCH, AND, OR
Date and Time	Perform calculations based on calendar periods	YEAR, MONTH, DATEDIFF, TOTALYTD
Filter Modification	Modify the filter context of a calculation	CALCULATE, ALL, FILTER
Relationship / Table	Navigate across related tables	RELATED, RELATEDTABLE, DISTINCT, COUNTROWS

SGOU

5.2.9.6 Worked Examples of Common Measures

Table 5.2.5 presents a small set of measures that recur in a large proportion of business Power BI reports, illustrating how the function categories described above combine in practice.

Table 5.2.5 Examples of Common DAX Measures

Measure Name	DAX Expression (illustrative)	Description
Total Sales	SUM(Fact_Sales[Amount])	Adds the Amount column across the current filter context
Order Count	DISTINCTCOUNT(Fact_Sales[OrderID])	Counts the number of distinct customer orders
Average Order Value	DIVIDE([Total Sales],[Order Count])	Divides total sales by the number of distinct orders
Sales Year-to-Date	TOTALYTD([Total Sales], Dim_Date[Date])	Accumulates total sales from the start of the year to the current date context
High-Value Sales	CALCULATE([Total Sales], Fact_Sales[Amount] > 10000)	Restricts total sales to transactions above a threshold value

Note that the DIVIDE function is generally preferred over the ordinary division operator in DAX, since DIVIDE gracefully handles the case of dividing by zero, returning a blank value instead of an error, which is a common data quality safeguard in business reports.

5.2.9.7 DAX and the M Language Compared

Learners familiar with Unit 1 will recall the M language, used within Power Query Editor to define data preparation steps. It is worth clarifying how M and DAX differ, since both are formula languages within Power BI but serve distinct purposes at different stages of the workflow. M operates before data is loaded into the model, transforming and reshaping entire tables one query step at a time, and its results are what actually get stored as the tables of the data model. DAX, by contrast, operates after data has been loaded, defining calculated columns, measures, and calculated tables that build on top of the tables M has already produced. In short, M answers the question of what the tables in the model should contain, while DAX answers the question of what can be calculated from those tables once they exist.

5.2.10 Best Practices for Data Modelling and DAX

- Prefer measures over calculated columns for numeric aggregations, reserving calculated columns for values genuinely needed for slicing, filtering, or sorting.

SGOU

- Design the model as a star schema wherever possible, keeping fact tables narrow and dimension tables descriptive.
- Use Single cross-filter direction by default, switching to Both only where a specific analytical requirement justifies the added complexity.
- Build a dedicated calendar (date) table rather than relying on date columns scattered across multiple fact tables, since a single calendar table simplifies time intelligence calculations.
- Give measures clear, business-friendly names and organize related measures into display folders as a model grows in size.

Applying these practices consistently helps ensure that a Power BI data model remains fast, understandable, and easy to extend as new tables, relationships, and calculations are added while progressing through the visualization and reporting activities covered in the remaining units of this block.

5.2.11 An Illustrative Example

Returning to the retail business introduced in Unit 1, suppose the cleaned and transformed sales data, together with separate tables for products, customers, stores, and a calendar, have now been loaded into Power BI Desktop. Following the star schema principle described in this unit, the analyst would relate the sales table to each of the four supporting tables using a one-to-many relationship, with the sales table on the many side in each case.

With the relationships in place, the analyst could then define a Total Sales measure using SUM over the sales amount column, an Average Order Value measure that divides Total Sales by a count of distinct orders, and a Sales Year-to-Date measure built using CALCULATE together with a time intelligence function applied to the calendar table. Because these are defined as measures rather than calculated columns, each one would automatically recompute to reflect whichever product, store, or date range a report viewer selects, without the analyst needing to anticipate every possible combination of filters in advance.

This example illustrates why the modelling decisions covered in this unit, choosing a star schema, defining relationships with the correct cardinality, and preferring measures over calculated columns for aggregations, directly determine how flexible and reliable the visualizations built in Unit 3 will ultimately be.

Recap

- Data modelling organizes the tables prepared in Power Query into a connected structure, defined through relationships, that Power BI uses to combine information across tables.
- Relationships are characterized by their cardinality (one-to-many, many-to-one, one-to-one, or many-to-many) and their cross-filter direction (Single or Both).

SGOU

- A star schema arranges a central fact table around surrounding dimension tables and is the generally recommended design for Power BI models, offering a simpler and more efficient alternative to a snowflake schema.
- Calculated columns are evaluated row by row and stored physically within the table, making them suitable for slicing, filtering, and sorting.
- Measures are evaluated dynamically according to filter context and are not stored in the table, making them the preferred approach for most aggregations, totals, and KPIs.
- DAX (Data Analysis Expressions) is the formula language used to define calculated columns, measures, and calculated tables in Power BI.
- The CALCULATE function allows the filter context of a DAX expression to be modified and underlies most advanced calculations, including time intelligence.

Objective Type Questions

1. What is the purpose of data modelling in Power BI?
2. What is a relationship in a Power BI data model?
3. What term describes the numerical relationship between rows in two related tables?
4. Which cross-filter direction allows filters to flow in both directions between related tables?
5. What type of table stores measurable, transactional data such as sales amounts?
6. What type of table stores descriptive attributes such as product or customer details?
7. Which schema arranges a central fact table around surrounding dimension tables?
8. Which schema further splits dimension tables into related sub-dimension tables?
9. How frequently is a calculated column's expression evaluated?
10. When is a measure's expression evaluated?
11. Which type of calculation is stored physically within a table?
12. What is the formula language used to define calculated columns and measures in Power BI called?
13. What context describes the current row being evaluated in a calculated column?

SGOU

14. What context describes the filters and slicers currently applied when a measure is evaluated?

15. Which DAX function is used to modify the filter context of a calculation?

Answers

1. To organize related tables and define how they connect for analysis
2. A defined link between a column in one table and a column in another table
3. Cardinality
4. Both
5. Fact table
6. Dimension table
7. Star schema
8. Snowflake schema
9. Once, when the data is refreshed
10. Dynamically, at the time a visual is rendered
11. Calculated column
12. DAX (Data Analysis Expressions)
13. Row context
14. Filter context
15. CALCULATE

SGOU

Assignments

1. Explain the concepts of cardinality and cross-filter direction with reference to a suitable example.
2. Describe the star schema and explain why it is generally recommended for Power BI data models over the snowflake schema.
3. Distinguish between calculated columns and measures, explaining the evaluation and storage differences between them.

4. Explain the concepts of row context and filter context in DAX, and describe how the CALCULATE function makes use of filter context.
5. Using a hypothetical sales dataset, describe how you would design a star schema and identify at least two measures you would create.

Suggested Reading

1. Powell, B. (2018). Mastering Microsoft Power BI: Expert techniques for effective data analytics and business intelligence. Packt Publishing.
2. Ferrari, A., & Russo, M. (2019). The Definitive Guide to DAX (2nd ed.). Microsoft Press.
3. Ahmed, M., & Pathan, A. K. (2020). Data analytics concepts, techniques and applications (1st ed.). Taylor and Francis.

Reference

1. Microsoft Corporation. Power BI Documentation - Data Modelling.
2. Microsoft Corporation. DAX Function Reference.

SGOU



Building and Publishing Visualizations

Learning Outcomes

After completion of this unit, learner will be able to:

- select an appropriate visual type based on the analytical purpose of a chart, graph, map, or KPI
- create and format core visuals such as bar, column, line, pie, table, matrix, and map visuals
- apply slicers and report-, page-, and visual-level filters to control what a report displays
- distinguish between drill down and drill through and configure each within a report
- import and use custom visuals from Microsoft AppSource
- apply design principles to build clear, uncluttered, and effective report layouts
- publish a completed report to the Power BI Service and describe how it is organized and shared

SGOU

Prerequisite

Units 1 and 2 of this block established, in turn, how data is connected, cleaned, and transformed, and how the resulting tables are organized into a data model with relationships, calculated columns, and measures. With a working data model in place, the next natural step is to present that data visually, which is the focus of this unit.

To engage meaningfully with this unit, learners should be comfortable with the outcomes of Units 1 and 2, particularly the distinction between calculated columns and measures, since most visuals in this unit are built primarily around measures. Learners should also recall the general principles of data visualization introduced earlier in this course, such as choosing an appropriate chart type for a given kind of data and avoiding unnecessary visual clutter, since this unit applies those principles specifically within the Power BI environment.

Keywords

Visualizations Pane, Fields Pane, Bar Chart, Line Chart, Map Visual, KPI, Card Visual, Slicer, Report-level Filter, Page-level Filter, Visual-level Filter, Drill Down, Drill Through, Custom Visuals, Power BI Service, Workspace, App, Publish.

Discussion

5.3.1 Introduction to Visualization in Power BI

Visualization is the stage at which the carefully connected, cleaned, and modelled data discussed in Units 1 and 2 is finally translated into a form that a business user can read, interpret, and act upon. A well-built data model can still fail to communicate anything useful if the visuals built upon it are poorly chosen or cluttered, which is why Power BI treats report design as a distinct and important discipline in its own right, supported by a dedicated set of tools within Power BI Desktop.

Every visual added to a Power BI report draws its values from the underlying data model, most commonly from the measures defined in Unit 2, together with descriptive fields from dimension tables used to group or label those measures. Because Power BI visuals are interactive, selecting a value in one visual, such as a bar in a bar chart, automatically filters and highlights related values in every other visual on the same page, allowing a viewer to explore the data rather than simply read a static picture of it.

This unit introduces the principal categories of visuals available in Power BI, the tools used to filter and drill into data interactively, and the process of publishing a finished report so that it can be shared with others.

A completed Power BI report commonly spans several pages, each built around a specific analytical theme, for example, an overview page presenting headline figures, followed by more detailed pages addressing sales by product, sales by region, and customer behaviour. Navigation between these pages, together with the interactive filtering, drilling, and highlighting behaviour described later in this unit, is what distinguishes a Power BI report from a static, printed summary, allowing a single report to serve a wide range of questions that different viewers might bring to it.

5.3.2 The Report Canvas and Visualization Workflow

Report authoring in Power BI Desktop takes place on the report canvas, a blank page onto which visuals are placed and arranged. Three panes work together in this process. The Fields pane lists every table and column in the data model. The Visualizations pane offers the gallery of visual types and the formatting controls for the selected visual. The canvas is where visuals are placed, resized and viewed as a report reader will see them.

Building a visual generally follows a simple sequence: a visual type is selected from the Visualizations pane, and fields are then dragged from the Fields pane into the

SGOU

appropriate wells, such as Axis, Legend, or Values, depending on the visual chosen. Power BI immediately renders the visual using live data from the model, allowing the analyst to adjust field placement and formatting interactively until the visual communicates the intended message clearly. A report itself may span multiple pages, each addressing a different aspect of the underlying data, in the same way a printed report might be organized into separate sections.

5.3.3 Core Chart Types

Power BI provides a broad gallery of built-in visuals, the great majority of which fall into a small number of families based on the analytical question they are best suited to answer. Figure 5.3.1 groups the most commonly used visual types according to this analytical purpose.

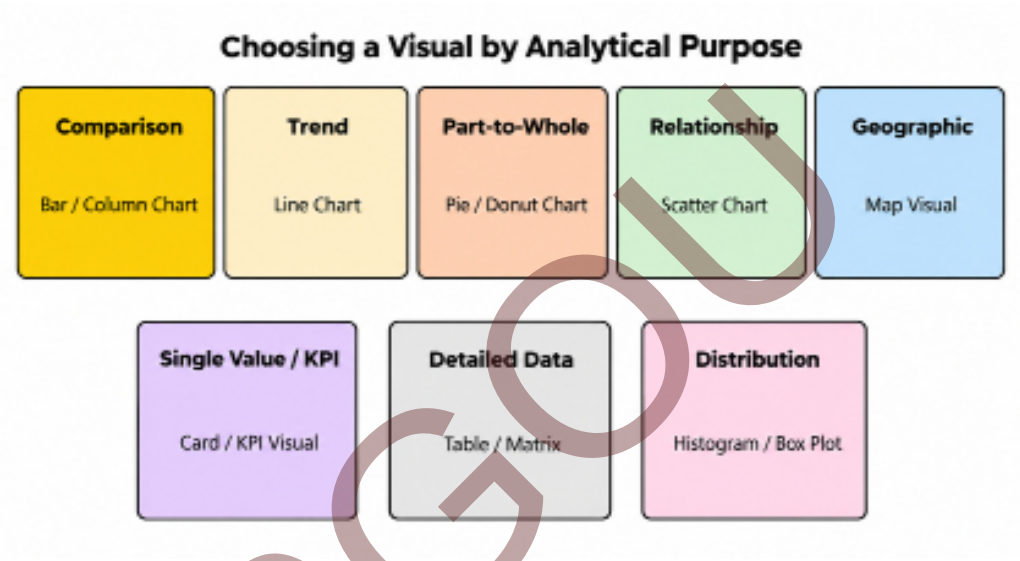


Fig. 5.3.1 Choosing a Visual by Analytical Purpose

As shown in Figure 5.3.1, bar and column charts are generally the preferred choice for comparison, since the human eye is well practised at judging differences in bar length or height. Line charts are best suited to showing a trend over a continuous scale, most commonly time. Pie and donut charts show a part-to-whole relationship but are best limited to a small number of categories, since they become difficult to read once more than four or five slices are involved. Scatter charts reveal the relationship between two numeric measures, and map visuals plot data against a geographic location.

Table 5.3.1 describes each of these core chart types in slightly more detail, together with the kind of field typically placed in each visual's principal wells.

Table 5.3.1 Common Power BI Visual Types and Their Applications

Visual Type	Best Suited For	Typical Fields Used
Bar / Column Chart	Comparing values across categories	Category on axis, measure in values

SGOU

Line Chart	Showing a trend over a continuous scale	Date/time on axis, measure in values
Pie / Donut Chart	Showing a part-to-whole breakdown (few categories)	Category as legend, measure in values
Scatter Chart	Examining the relationship between two measures	Two measures on axes, category as legend
Map Visual	Displaying data by geographic location	Location field, measure sized as a bubble or colour

5.3.3.1 Combo and Waterfall Charts

Beyond the core chart types summarized above, Power BI includes a small number of specialized visuals suited to more particular analytical questions. A combo chart overlays a column chart and a line chart on the same axis, which is useful when comparing two measures on different scales, such as showing monthly sales as columns alongside a profit margin percentage as an overlaid line. A waterfall chart shows how an initial value is increased or decreased by a sequence of positive and negative contributions to arrive at a final value, and is commonly used to explain a change in a figure, such as profit, from one period to the next in terms of its underlying components.

5.3.4 Tables and Matrix Visuals

Not every analytical question is best answered with a chart. A Table visual presents data in a simple grid of rows and columns, similar in appearance to a spreadsheet, and is useful when a viewer needs to see precise, detailed values rather than a visual comparison. A Matrix visual extends this idea by allowing rows and columns to be organized hierarchically, similar to a pivot table, so that values can be expanded or collapsed at different levels, such as viewing sales first by year, then by quarter, and then by month, within the same visual.

Both table and matrix visuals support conditional formatting, allowing cell backgrounds or font colours to change automatically based on the underlying value, which helps a viewer quickly identify figures that fall above or below an expected threshold without needing to convert the data into a chart at all.

5.3.5 Maps and Geographic Visuals

Power BI provides several visuals for displaying data geographically. The standard Map visual plots data as bubbles positioned over a background map, with bubble size representing the magnitude of a chosen measure. The Filled Map visual instead shades entire geographic regions, such as countries or states, according to a measure's value, which is useful for comparing performance across a defined set of territories. Power BI recognizes common geographic fields, such as country, state, or city names,

SGOU

automatically, though as noted in Unit 2, assigning an explicit data category to a column improves the accuracy with which Power BI plots it.

Because geographic visuals rely on an internet-based mapping service to render locations, they require an active internet connection, and organizations with strict data residency requirements should verify that this behaviour is acceptable before relying heavily on map visuals in a published report.

5.3.6 KPIs and Card Visuals

A Card visual displays a single summarized value, such as Total Sales for the current filter context, in large, prominent text, and is commonly used to draw attention to a headline figure at the top of a report page. A Multi-row Card extends this to display several such values together in a compact list.

A KPI (Key Performance Indicator) visual builds on the card concept by comparing a chosen measure against a defined target, displaying not only the current value but also a trend indicator and a visual cue, typically a colour or icon, showing whether performance is on track, ahead of, or behind the target. KPI visuals are particularly effective on management dashboards, where a viewer needs to assess performance against a goal at a glance, without needing to interpret a full chart.

A KPI visual is built from three underlying components: the indicator, which is the measure being tracked; the trend axis, typically a date field used to plot the indicator's recent history as a small sparkline; and the target, either a fixed value or a separate measure representing the goal against which the indicator is judged. Table 5.3.2 summarizes these components.

Table 5.3.2 Key Performance Indicator (KPI) Visual Components

KPI Component	Role	Typical Example
Indicator	The measure whose current performance is displayed	Total Sales
Trend Axis	A time-based field used to plot recent history	Month
Target	A fixed value or measure representing the goal	Sales Target measure or a fixed budget figure

5.3.7 Slicers and Filters

Slicers and filters allow a report viewer to narrow down the data displayed across some or all of the visuals on a page. Although related in purpose, they differ in how they are presented and in the scope over which they apply.

SGOU

5.3.7.1 Slicers

A slicer is a visual, placed directly on the report canvas, that allows a viewer to select one or more values from a field, such as a specific year, region, or product category, with the selection immediately filtering every other visual on the same page that uses the corresponding table. Slicers may be presented as a list, a dropdown, a set of buttons, or, for numeric and date fields, a range slider.

5.3.7.2 Filters

Filters are configured through the dedicated Filters pane rather than being placed visibly on the canvas, and can be applied at three distinct levels of scope. A visual-level filter restricts the data shown in a single selected visual only. A page-level filter applies to every visual on the current report page. A report-level filter applies across every page of the entire report. Choosing the appropriate level allows an analyst to, for example, exclude a test region from an entire report using a single report-level filter, while separately restricting one specific chart to a particular product category using a visual-level filter.

5.3.7.3 Cross-Highlighting Between Visuals

In addition to slicers and the Filters pane, Power BI visuals interact with one another automatically through a behaviour known as cross-highlighting. Selecting a data point in one visual, such as a single bar in a bar chart, does not remove the remaining data from other visuals on the page but instead highlights the portion of each related visual that corresponds to the selected value, while dimming the rest. This allows a viewer to compare a single category's contribution against the whole without losing sight of the broader picture. The precise behaviour, whether a selection highlights or fully filters other visuals, can be configured for each visual through its Edit Interactions settings.

Table 5.3.3 Filter Scopes and Slicers in Power BI

Filter Scope	Applies To	Typical Use
Visual-level filter	One selected visual only	Restricting a single chart to a specific subset of data
Page-level filter	All visuals on the current page	Focusing an entire page on one region or period
Report-level filter	All pages in the report	Excluding test or invalid data across the whole report
Slicer	All visuals using the filtered field (page or synced pages)	Letting the viewer interactively choose a value

SGOU

5.3.8 Drill Down and Drill Through

Power BI provides two related but distinct mechanisms for navigating from a summary view into greater detail: drill down and drill through.

5.3.8.1 Drill Down

Drill down allows a viewer to move to a more detailed level within the same visual, using a hierarchy such as the Year, Quarter, Month hierarchy discussed in Unit 2. Clicking the drill-down arrow on a chart and then selecting a particular year, for example, reveals the quarters within that year without leaving the current visual or page.

5.3.8.2 Drill Through

Drill through, by contrast, takes the viewer to an entirely different, typically more detailed, report page, carrying the selected value along as a filter. For example, right-clicking a specific product on a summary sales page might offer a drill-through option that opens a dedicated Product Detail page, pre-filtered to show only that product's figures. This allows a single detail page to be designed once and reused for every possible value in the summary visual, rather than requiring a separate page for each product.

Table 5.3.5 Key differences between these two navigation mechanisms

Aspect	Drill Down	Drill Through
Destination	Deeper level within the same visual	A separate, dedicated report page
Basis for navigation	A predefined hierarchy	A selected field value passed as a filter
Typical use	Exploring Year to Quarter to Month within one chart	Jumping from a summary to a detail page for one item

5.3.9 Custom Visuals

In addition to the built-in visual gallery, Power BI supports custom visuals, which extend the platform with chart types and interactive components not included by default, such as specialized gauges, word clouds, Sankey diagrams, and organizational charts. Custom visuals are most commonly obtained from Microsoft AppSource, a curated marketplace of visuals submitted by Microsoft and third-party developers, and can be imported directly into Power BI Desktop from within the Visualizations pane.

Because a custom visual runs code that is not part of Power BI's core, built-in library, organizations with strict governance requirements often review or restrict which custom visuals may be used, and Power BI administrators can disable custom visuals entirely for an organization if required. Once imported and added to a report, however, a custom visual behaves much like any built-in visual, drawing its data from fields dragged from the Fields pane.

SGOU

Popular examples of custom visuals available through AppSource include word clouds for summarizing text data, Sankey diagrams for illustrating flow between categories, gauge visuals for showing progress toward a single target, and organizational chart visuals for depicting reporting hierarchies. New custom visuals are added to AppSource on an ongoing basis, so it is good practice for an analyst to search AppSource for a suitable existing visual before considering a heavily customized alternative.

5.3.10 Formatting and Designing Effective Reports

Building individual visuals is only part of effective report design; how those visuals are arranged, formatted, and labelled on a page has a substantial impact on how easily a viewer can understand the report. A small number of widely followed principles apply consistently across Power BI reports:

- Lead with the most important figure or chart, positioning it prominently, typically near the top left of the page, since most viewers scan a page in that direction first.
- Limit each page to a focused set of related visuals rather than crowding many unrelated charts onto a single page, which tends to overwhelm rather than inform the viewer.
- Use consistent colours to represent the same category across every visual on a page, so that a viewer does not need to relearn a colour scheme from one chart to the next.
- Provide clear titles for every visual, and axis or field labels wherever a value's meaning would otherwise be ambiguous.
- Use KPIs and card visuals sparingly, reserving them for the handful of figures that genuinely represent headline performance indicators.

Power BI Desktop also allows a consistent theme, covering colours, fonts, and default visual formatting, to be applied across an entire report, which helps ensure that reports produced by different analysts within the same organization maintain a recognizable and professional appearance.

5.3.10.1 Bookmarks and Buttons for Interactive Navigation

Bookmarks allow an analyst to capture the current state of a report page, including its filter selections, slicer values, and the visibility of individual visuals, so that this exact state can be recalled later at the click of a button. Bookmarks are frequently combined with buttons and a technique sometimes referred to as layered navigation, in which several alternative views are stacked on the same page and swapped in and out using bookmarks, giving the appearance of separate screens within what is technically a single report page. This is commonly used to build guided, story-like report experiences, or to offer a toggle between, for example, a summary view and a detailed view of the same underlying data.

SGOU

5.3.11 Publishing a Report to the Power BI Service

Once a report has been built and reviewed within Power BI Desktop, it is typically published to the Power BI Service so that it can be shared with colleagues, as introduced in Unit 1's discussion of the overall Power BI ecosystem. Figure 5.3.2 illustrates the typical sequence of activities involved in taking a report from initial design through to being available to its intended audience.

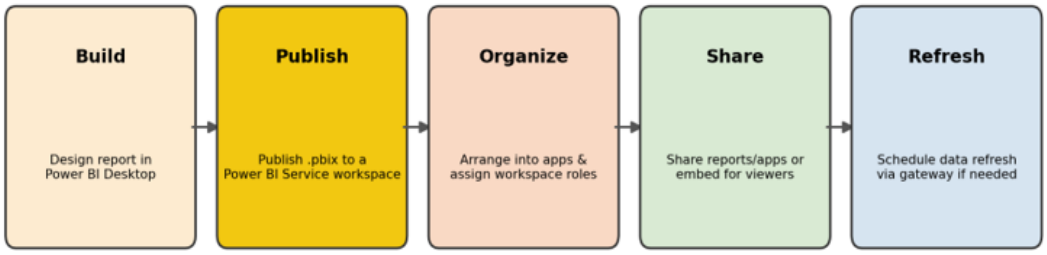


Fig. 5.3.2 Typical Workflow for Publishing and Sharing a Power BI Report

As shown in Figure 5.3.2, publishing begins with selecting the Publish command from within Power BI Desktop, which uploads the report, together with its underlying dataset, to a chosen workspace within the Power BI Service. A workspace is a container used to organize related reports, dashboards, and datasets, and to control which colleagues have access to them, typically corresponding to a team, department, or specific project.

Within the Power BI Service, one or more reports and dashboards from a workspace can be bundled together into an app, which presents a curated, read-only view of the content to a wider audience without exposing the underlying workspace itself. Access to a workspace or app is controlled through defined roles, such as Viewer, Contributor, or Admin, which determine whether a given colleague can merely view content, edit it, or manage workspace settings.

Finally, where the underlying dataset draws on a data source that changes over time, a scheduled refresh can be configured within the Power BI Service so that the published report continues to reflect current data without the analyst needing to manually republish it. As discussed in Unit 1, refreshing data that resides on an organization's internal network requires an on-premises data gateway to bridge the connection securely.

5.3.11.1 Sharing Options

Beyond distributing content through workspaces and apps, Power BI offers several further sharing mechanisms suited to different audiences. Direct share sends a link to a specific report or dashboard to named colleagues, suitable for one-off or informal sharing. Publish to web generates a publicly accessible link requiring no Power BI account at all, and should be used with considerable caution, since it is intended only for genuinely public, non-sensitive data. For audiences that need to view a report on a fixed schedule without interacting with the live version, a report can be exported to PDF or PowerPoint, producing a static snapshot of its current state. Organizations building their own customer-facing applications may instead rely on Power BI Embedded, briefly introduced in Unit 1, to surface reports directly inside a separate application.

SGOU

5.3.12 Best Practices for Building and Publishing Reports

- Choose the simplest visual capable of answering the analytical question at hand, rather than the most visually elaborate option available.
- Test drill down, drill through, and slicer interactions thoroughly before publishing, since these are the features viewers will rely on most to explore the report themselves.
- Organize workspaces logically, separating reports still under development from those approved for wider distribution.
- Review and restrict access roles carefully, granting Contributor or Admin access only to those who genuinely need to modify workspace content.
- Document the intended refresh schedule and data source dependencies for every published dataset, so that colleagues understand how current the figures in a report are likely to be.

5.3.13 An Illustrative Example

Returning once more to the retail business introduced in Unit 1 and continued in Unit 2, suppose the analyst has now built a data model containing the Total Sales, Average Order Value, and Sales Year-to-Date measures described earlier. A first report page might place a Card visual showing Total Sales prominently at the top, a column chart comparing Total Sales by product category beneath it, and a line chart showing the Sales Year-to-Date trend across the calendar year, with a slicer allowing the viewer to select a specific store.

A drill-down hierarchy on the line chart would allow the viewer to move from yearly to quarterly to monthly detail without leaving the page, while a drill-through configuration on the column chart could take the viewer to a dedicated Product Detail page, pre-filtered to a single category, for closer inspection. Once reviewed, the analyst would publish this report to a dedicated workspace for the retail business, bundle it into an app for the branch managers who need to view it, and configure a nightly scheduled refresh so that each manager sees up-to-date figures every morning.

This example illustrates how the visualization and publishing techniques covered in this unit build directly on the data preparation and modelling work carried out in Units 1 and 2, and prepares the ground for the security and deployment considerations examined in Unit 4.

SGOU

Recap

- Visualization translates the prepared and modelled data from Units 1 and 2 into charts, tables, maps, and KPIs that a business user can interpret and act upon.
- Core chart types can be grouped by analytical purpose: bar and column charts for comparison, line charts for trend, pie and donut charts for part-to-whole relationships, scatter charts for relationships between measures, and map visuals for geography.
- Table and matrix visuals present detailed, precise values, with matrix visuals supporting hierarchical rows and columns similar to a pivot table.
- Card and KPI visuals highlight single headline figures, with KPI visuals additionally comparing a measure against a defined target.
- Slicers and filters narrow down report data, with filters operating at visual, page, or report level depending on the intended scope.
- Drill down moves to greater detail within the same visual using a hierarchy, while drill through navigates to a separate, pre-filtered detail page.
- Custom visuals, obtained mainly through Microsoft AppSource, extend Power BI's built-in visual gallery.
- Publishing moves a report from Power BI Desktop into a Power BI Service workspace, where it can be organized into apps, shared under defined access roles, and kept current through scheduled data refresh.

SGOU

Objective Type Questions

1. Which chart type is generally preferred for comparing values across categories?
2. Which chart type is best suited to showing a trend over time?
3. Which visual type shows a part-to-whole relationship and is best limited to a few categories?
4. Which visual displays rows and columns organized hierarchically, similar to a pivot table?
5. Which type of visual compares a measure against a defined target?
6. What is a slicer used for in a Power BI report?
7. Which filter scope applies to every visual on the current report page only?

8. Which filter scope applies across every page of a report?
9. What is the term for moving to a more detailed level within the same visual using a hierarchy?
10. What is the term for navigating to a separate, pre-filtered detail page?
11. Where are custom visuals for Power BI most commonly obtained from?
12. What is a Power BI workspace used for?
13. What is a Power BI app used for?
14. What is required to refresh data hosted on an organization's internal network from the Power BI Service?
15. What command in Power BI Desktop uploads a report to the Power BI Service?

Answers

1. Bar or column chart
2. Line chart
3. Pie / donut chart
4. Matrix visual
5. KPI visual
6. To allow the viewer to interactively select values and filter the report page
7. Page-level filter
8. Report-level filter
9. Drill down
10. Drill through
11. Microsoft AppSource
12. To organize and control access to related reports, dashboards, and datasets
13. To bundle content from a workspace into a curated, read-only view for a wider audience
14. An on-premises data gateway
15. Publish

SGOU

Assignments

1. Explain how the choice of visual type should be guided by the analytical question a report page is intended to answer, with suitable examples.
2. Distinguish between visual-level, page-level, and report-level filters, and explain when each would be appropriate.
3. Explain the difference between drill down and drill through, describing a scenario in which each would be used.
4. Describe the process of publishing a report from Power BI Desktop to the Power BI Service, including the role of workspaces and apps.
5. Discuss any four design principles that contribute to an effective and uncluttered Power BI report layout.

Suggested Reading

1. Powell, B. (2018). Mastering Microsoft Power BI: Expert techniques for effective data analytics and business intelligence. Packt Publishing.
2. Knaflic, C. N. (2015). Storytelling with Data: A Data Visualization Guide for Business Professionals. Wiley.
3. Ahmed, M., & Pathan, A. K. (2020). Data analytics concepts, techniques and applications (1st ed.). Taylor and Francis.

Reference

1. Microsoft Corporation. Power BI Documentation - Visualizations.
2. Microsoft Corporation. Power BI Documentation - Publish and Share.

SGOU



Power BI Security and Deployment

Learning Outcomes

After completion of this unit, learner will be able to:

- explain the layers of security available within the Power BI ecosystem
- configure row-level security to restrict data access based on a user's role
- distinguish between static and dynamic row-level security and identify appropriate use cases for each
- describe the purpose of sensitivity labels in protecting Power BI content
- explain the role of workspaces, apps, and licensing in governing access to published reports
- describe Power BI Embedded and distinguish it from standard report sharing through the Power BI Service
- explain the purpose, architecture, and modes of the on-premises data gateway
- describe the role of deployment pipelines in managing a report's life cycle from development to production

SGOU

Prerequisite

Units 1 to 3 of this block traced the journey of data through Power BI, from connecting and preparing it, through modelling it with relationships and DAX, to presenting it as interactive visuals and publishing it to the Power BI Service. Once a report has been published and is being used by an organization, two further concerns become important: ensuring that only authorized users can see the data they are entitled to see, and ensuring that the underlying data can be refreshed and accessed reliably, including data that resides behind an organization's own firewall. This unit addresses both concerns under the combined themes of security and deployment.

To engage meaningfully with this unit, learners should be comfortable with the outcomes of Unit 3, particularly the concepts of workspaces, apps, and the publishing workflow, since this unit builds directly upon that foundation. A general awareness of how user accounts and permissions work within any shared computer system will also help learners relate to the access-control concepts introduced here.

Keywords

Row-Level Security, Object-Level Security, Sensitivity Labels, Workspace Roles, Power BI Embedded, On-premises Data Gateway, Gateway Cluster, Deployment Pipeline, Auditing.

Discussion

5.4.1 Introduction to Power BI Security and Deployment

As Power BI reports move from a single analyst's desktop into wider organizational use, two questions become unavoidable: who is allowed to see this data, and how does this data keep flowing reliably once the report is in production? Security in Power BI addresses the first question, governing which users can view, edit, or manage particular reports, datasets, and even particular rows within a dataset. Deployment addresses the second question, covering how reports are connected to data that may reside outside the cloud, how content moves through stages of development and testing before reaching end users, and how that content is embedded or distributed beyond the standard Power BI Service interface.

These concerns are not an afterthought bolted onto a finished report; they are typically planned for from the earliest stages of a Power BI project, since decisions made during data modelling, such as which fields are available for filtering, directly affect how security can later be applied. This unit surveys the principal mechanisms Power BI provides for security and deployment, building on the modelling and publishing concepts introduced in Units 2 and 3.

It is useful to think of security and deployment as complementary rather than separate concerns. Security determines who is permitted to see a given piece of data once a report is available, while deployment determines how reliably that report continues to be available, and how safely it moves from an analyst's own experimentation into something the wider organization can depend on. A report that is perfectly secured but refreshes unreliably is of little practical use, just as a report that refreshes flawlessly but exposes data to the wrong audience creates real organizational risk. Effective Power BI governance therefore treats both concerns together, rather than addressing one while neglecting the other.

5.4.2 Layers of Security in Power BI

Power BI security operates at several distinct layers, each controlling a different aspect of what a user can access. Table 5.4.1 summarizes these layers before each is discussed in more detail in the sections that follow.

SGOU

Table 5.4.1 Power BI Security Layers and Mechanisms

Security Layer	What It Controls	Typical Mechanism
Workspace / App Roles	Who can view, edit, or manage a workspace's content	Admin, Member, Contributor, Viewer roles
Row-Level Security (RLS)	Which rows of data a user can see within a report	DAX filter expressions applied per role
Object-Level Security (OLS)	Which tables or columns a user can see at all	Restrictions defined in the data model
Sensitivity Labels	How content is classified and protected once exported or shared	Labels such as Confidential or Internal
Tenant / Organizational Settings	Which features and sharing options are available across the organization	Administrator-controlled tenant settings

5.4.2.1 Identity and Access Management

Underneath every layer summarized in Table 5.4.1, Power BI relies on an organization's underlying identity platform to establish who a user actually is before any Power BI-specific permission is even evaluated. Every Power BI user signs in with an organizational account, and administrators can apply identity-level protections, such as multi-factor authentication or conditional access policies that restrict sign-in to trusted devices or networks, at this identity layer. Because these protections apply before Power BI's own role and row-level security checks are reached, they act as an additional, foundational safeguard beneath the Power BI-specific mechanisms described in the remainder of this unit.

5.4.3 Row-Level Security (RLS)

Row-level security allows a single report and a single dataset to be shared with multiple users, while ensuring that each user sees only the rows of data relevant to them. Rather than building a separate report for every region, department, or customer, an analyst defines one or more roles within Power BI Desktop, each associated with a DAX filter expression, and then assigns individual users or security groups to these roles once the report is published to the Power BI Service.

As illustrated in Figure 5.4.1, a single underlying report and dataset can serve several different viewers, with row-level security ensuring that each viewer's experience of the

SGOU

report is automatically filtered according to the role they have been assigned, without the analyst needing to maintain separate copies of the report.

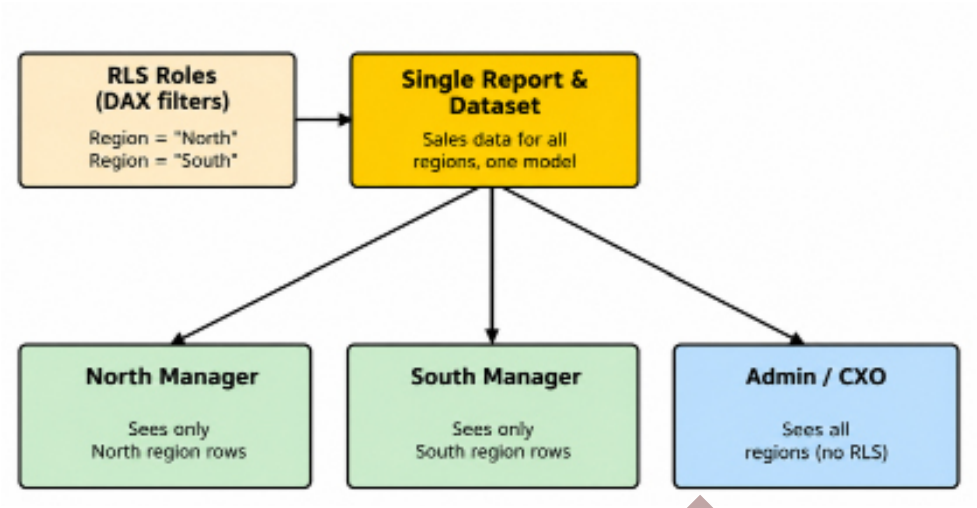


Fig. 5.4.1 Row-Level Security Applying Different Filters to the Same Report

As shown in Figure 5.4.1, the North Manager and South Manager both view the same underlying report, but row-level security filters restrict each of them to rows matching their own region, while an administrator assigned no restrictive role continues to see the full, unfiltered dataset.

5.4.3.1 Static Row-Level Security

In static row-level security, the filter condition for a role is written directly as a fixed value within the DAX expression, for example, restricting a role to rows where the Region column equals North. Static RLS is straightforward to set up but requires a separate role, and a corresponding DAX expression, to be created and maintained for every distinct value a user might need to be restricted to, which becomes cumbersome as the number of regions, territories, or departments grows.

5.4.3.2 Dynamic Row-Level Security

Dynamic row-level security instead compares a column in the data model against the identity of the currently signed-in user, typically using the USERPRINCIPALNAME function together with a mapping table that links each user to the region, department, or customer they are permitted to see. Because the filtering logic reads from a table rather than being hard-coded per role, dynamic RLS scales far more easily to organizations with many users and frequently changing assignments, at the cost of a slightly more involved initial setup.

5.4.3.3 Object-Level Security

Where row-level security restricts which rows a user can see, object-level security restricts which tables or columns are visible to a user at all, which is useful when an entire table or column, such as an employee salary column, should be hidden from certain roles rather than merely filtered.

SGOU

5.4.3.4 A Worked Example of an RLS Role

Continuing with the retail sales model introduced in Units 2 and 3, a static role named North Only might be defined against the Dim_Store table using a DAX filter expression such as Dim_Store[Region] = "North", which is entered in the Manage Roles dialog in Power BI Desktop. A dynamic equivalent would instead reference a mapping table, for example, a UserRegionMap table linking each manager's email address to a region, with a filter expression such as Dim_Store[Region] = LOOKUPVALUE(UserRegionMap[Region], UserRegionMap[Email], USERPRINCIPALNAME()). Once published, each user assigned to the underlying security group would automatically see only the rows permitted by whichever expression applies to them, without the report itself needing to change.

5.4.4 Sensitivity Labels and Data Protection

Beyond controlling who can open a report, organizations often need to classify and protect the content of that report once it is exported, downloaded, or shared outside Power BI, for instance as a PDF or an Excel file. Sensitivity labels, typically defined by an organization's information protection policy, can be applied to a Power BI report or dataset to indicate its classification, such as Public, Internal, Confidential, or Highly Confidential. Once applied, a sensitivity label can automatically travel with content that is exported from Power BI, and can be configured to restrict actions such as printing or forwarding, extending an organization's data protection policy consistently from its documents into its analytics.

Table 5.4.2 illustrates a typical sensitivity label scheme, of the kind many organizations adapt to their own policy requirements.

Table 5.4.2 Information Sensitivity Labels and Classification Levels

Label	Typical Meaning	Example Content
Public	Safe for release outside the organization	Published marketing dashboards
Internal	For use within the organization only	General operational reports
Confidential	Restricted to specific teams or roles	Departmental financial reports
Highly Confidential	Restricted to a named, small group of individuals	Board-level strategic reporting

5.4.5 Governance Within the Power BI Service

Unit 3 introduced workspaces and apps as the mechanisms through which reports

SGOU

are organized and distributed within the Power BI Service. From a security perspective, each workspace has its own set of assigned roles, commonly Admin, Member, Contributor, and Viewer, which determine whether a given colleague can merely view the workspace's content, edit and publish new reports into it, or manage the workspace's settings and membership entirely.

Licensing also plays a governance role: as discussed in Unit 1, colleagues generally require at least a Power BI Pro license to access shared workspace content directly, whereas content packaged into an app and backed by Premium capacity can typically be distributed more broadly without every recipient needing an individual Pro license. Administrators additionally control tenant-wide settings, such as whether report export or external sharing is permitted at all, providing a final, organization-wide layer of governance above the workspace level.

5.4.5.1 Endorsement and Certification

As an organization accumulates a growing number of datasets and reports, it becomes useful to signal which content is authoritative and trustworthy. Power BI supports this through endorsement, allowing a dataset or report to be marked as Promoted by its owner, indicating it is worth using as a starting point for further work, or Certified by a designated reviewer within the organization, indicating it has been vetted against organizational data quality and governance standards. Endorsed content is visually flagged when colleagues search for datasets to build new reports upon, helping steer users toward well-governed sources rather than ad hoc or duplicate datasets.

5.4.6 Power BI Embedded

As introduced briefly in Unit 1, Power BI Embedded allows Power BI reports and dashboards to be displayed inside a separate application, rather than being viewed through the standard Power BI Service website or mobile app. This capability is commonly used by software vendors who wish to offer analytics as a feature of their own product, and by organizations wishing to surface Power BI content within an internal portal or customer-facing website.

5.4.6.1 Embed for Your Organization

In this scenario, Power BI content is embedded into an internal application for an organization's own employees, who already hold Power BI accounts and licenses. Access control continues to follow the normal Power BI security model described earlier in this unit, including row-level security, since the viewer is still recognized as a licensed Power BI user.

5.4.6.2 Embed for Your Customers

In this scenario, an application developer embeds Power BI content for external users, such as customers of a software product, who do not themselves hold individual Power BI accounts. Instead, the embedding application authenticates to Power BI on behalf of these users using a dedicated service identity, and the developer's own application becomes responsible for controlling which customer sees which report or which filtered

SGOU

subset of data, often still leveraging row-level security to separate one customer's data from another's within a shared dataset.

For example, a company that provides inventory management software to several retail chains might embed one Power BI report in its own web application. It can use this scenario with dynamic row-level security based on a CustomerID column. Each retail chain that logs in then sees only its own inventory analytics, even though all the chains' data is stored in the same dataset. This approach allows the software vendor to build and maintain one report rather than a separate copy per customer, while still keeping each customer's data properly isolated.

5.4.7 The On-premises Data Gateway

Unit 1 introduced the on-premises data gateway briefly as the component that allows the Power BI Service to refresh data stored behind an organization's firewall. This unit examines the gateway in more detail, since it sits at the centre of most enterprise Power BI deployments that combine cloud-based reporting with on-premises data sources such as internal databases, file shares, or enterprise systems.

As illustrated in Figure 5.4.2, the gateway is installed on a server within the organization's own network and maintains a secure, outbound-only connection to the Power BI Service. Because the connection is initiated from inside the network outward, the organization does not need to open inbound ports through its firewall, which is an important security advantage of the gateway architecture.

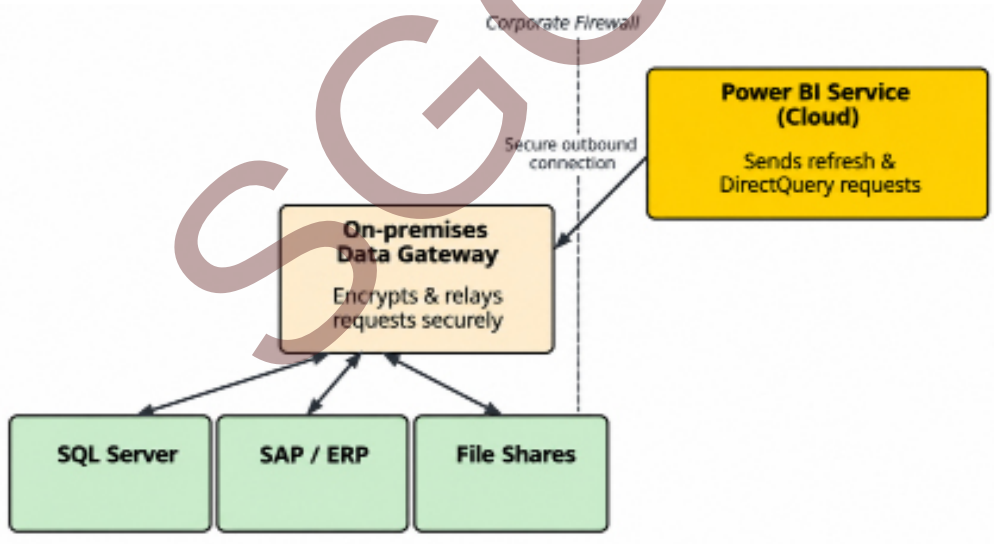


Fig. 5.4.2 On-premises Data Gateway Architecture

As shown in Figure 5.4.2, when the Power BI Service needs to refresh a dataset or execute a DirectQuery request against an on-premises source, it sends that request to the gateway across the secure connection. The gateway, running inside the corporate network, then queries the relevant source, such as SQL Server, an ERP system, or a file share, and relays the results back to the Power BI Service, all without the underlying data sources needing to be exposed directly to the internet.

SGOU

5.4.7.1 Gateway Modes

Power BI offers the gateway in two modes. The standard mode, intended for organizational use, allows multiple users to share a single gateway installation to connect to multiple data sources, and supports both scheduled refresh and DirectQuery. The personal mode is intended for a single user's own scheduled refreshes from their own machine, supports only Import mode refresh, and cannot be shared with colleagues, making it more suitable for individual use than for enterprise deployment.

Table 5.4.3 Comparison of Standard Mode and Personal Mode Gateways

Aspect	Standard Mode	Personal Mode
Intended use	Organizational, shared deployment	A single user's own reports
Number of users	Multiple users can share the gateway	Single user only
DirectQuery support	Supported	Not supported
Live Connection support	Supported	Not supported
Administration	Managed centrally by an administrator	Managed by the individual user

5.4.7.2 Gateway Clusters and High Availability

For organizations that depend heavily on scheduled refreshes and DirectQuery performance, a single gateway installation can represent a single point of failure. Power BI addresses this by allowing multiple gateway installations to be grouped into a gateway cluster, in which requests are automatically distributed across the available gateways and, should one gateway become unavailable, the remaining gateways in the cluster continue to service requests, improving both performance and reliability.

5.4.8 Deployment Pipelines

As a Power BI solution matures within an organization, changes to reports and datasets typically need to be tested before they reach the users who rely on them daily. Power BI deployment pipelines support this need by providing three linked stages, conventionally named Development, Test, and Production, each backed by its own workspace. An analyst can develop and refine a report in the Development stage, promote it to the Test stage for review and validation, and finally promote it to the Production stage, where it becomes available to its intended audience.

Deployment pipelines allow content, and optionally the underlying data source connections and parameters, to be compared and promoted between stages in a controlled manner, reducing the risk of an incomplete or incorrectly tested report reaching end users directly, in much the same way that a software development team might maintain separate development, staging, and production environments for an application.

SGOU

Table 5.4.4 Stages of Power BI Deployment Pipelines

Stage	Purpose	Typical Audience
Development	Build and refine reports and datasets	The report author only
Test	Validate calculations, security roles, and data refresh	A small group of reviewers
Production	Deliver the finished report for regular use	The full intended audience

5.4.9 Monitoring and Auditing

Alongside controlling access, organizations often need visibility into how Power BI content is actually being used and accessed. The Power BI Service provides usage metrics for individual reports, showing information such as how many times a report has been viewed and by which users, which helps report owners understand adoption and identify content that may no longer be needed. At the organizational level, administrators can review audit logs that record activities such as sharing, exporting, and permission changes across the tenant, supporting both security investigations and broader compliance requirements.

Table 5.4.5 lists a few of the activities commonly captured in Power BI audit logs, illustrating the breadth of visibility available to an administrator responsible for tenant-wide governance.

Table 5.4.5 Power BI Audit Log Activities and Their Significance

Logged Activity	Why It Matters
Report or dashboard viewed	Establishes usage patterns and adoption of a given report
Content shared externally	Flags data leaving the boundaries of the organization
Row-level security role membership changed	Confirms who currently has access to which filtered data
Dataset exported to Excel or PDF	Tracks content leaving the governed Power BI environment
Gateway data source credentials updated	Highlights changes to how on-premises data is accessed

SGOU

5.4.10 Best Practices for Security and Deployment

- Prefer dynamic row-level security over static role-per-value security for any dataset with more than a handful of security categories, to keep the model maintainable as the organization grows.
- Assign workspace roles and app permissions to security groups rather than individual users wherever possible, so that access changes as employees join, move, or leave the organization without requiring the report itself to be modified.
- Apply sensitivity labels consistently with the organization's broader information protection policy, rather than treating Power BI as a separate system.
- Use a dedicated, clustered gateway installation for production workloads rather than relying on a single, personally managed gateway.
- Adopt deployment pipelines, or an equivalent development and testing discipline, for any report that a significant number of users depend on regularly.
- Test row-level security thoroughly using the View As feature within Power BI Desktop and the Power BI Service before a report is shared, since an untested role can just as easily hide data that should be visible as expose data that should be hidden.

Taken together, these practices help ensure that the analytical value built up through Units 1 to 3 of this block, connecting, modelling, and visualizing data, is delivered securely and reliably to the people who depend on it, without exposing sensitive data beyond its intended audience or leaving report delivery dependent on a single point of failure.

5.4.11 An Illustrative Example

Returning once more to the retail business introduced in earlier units, suppose the sales report built across Units 1 to 3 is now ready to be shared with every regional manager across the country, as well as with senior management. Rather than building a separate report for each region, the analyst would define a dynamic row-level security role that compares each signed-in manager's email address against a mapping table linking managers to their assigned region, so that every manager automatically sees only their own region's figures when opening the single published report.

Because the underlying sales data resides in an on-premises SQL Server database, the analyst would also configure a scheduled refresh through a clustered on-premises data gateway, ensuring the report continues to reflect current data even if one gateway server requires maintenance. Finally, before rolling the report out organization-wide, the analyst would promote it through a deployment pipeline, from Development through Test to Production, allowing a small group of reviewers to validate the security roles and calculations before every regional manager gains access.

SGOU

This example illustrates how the security and deployment mechanisms covered in this unit, row-level security, gateways, and deployment pipelines, work together to take a report from a single analyst's workstation to a reliable, properly governed organizational asset.

Across all four units, this block has followed a single Power BI project from its first stage to final delivery across the organization. In Unit 1, raw data was connected to and prepared. In Unit 2, it was shaped into a reliable model with DAX calculations. In Unit 3, it was presented through interactive, well-designed visuals. In this unit, it is finally secured and deployed responsibly. This progression mirrors the life cycle that most real-world business intelligence projects follow, and the concepts introduced across these four units provide a practical foundation for working with Power BI, or with comparable self-service analytics tools, in a professional setting.

Recap

- Power BI security operates at several layers, including workspace and app roles, row-level security, object-level security, sensitivity labels, and tenant-level settings.
- Row-level security restricts the rows of data a user can see within a shared report, and can be implemented as static (fixed per role) or dynamic (based on the signed-in user's identity).
- Sensitivity labels classify and protect Power BI content in line with an organization's broader information protection policy, including content that is exported outside Power BI.
- Workspace roles and app permissions, together with tenant-level administrative settings, govern who can view, edit, or manage published Power BI content.
- Power BI Embedded allows reports to be displayed within another application, either for an organization's own licensed users or for external customers via a developer-managed identity.
- The on-premises data gateway provides a secure, outbound-only connection that allows the Power BI Service to refresh and query data sources located behind a corporate firewall, and can be clustered for reliability.
- Deployment pipelines support a controlled Development, Test, and Production life cycle for Power BI content, reducing the risk of untested changes reaching end users.

SGOU

Objective Type Questions

1. What does row-level security control within a Power BI report?
2. What is the key difference between static and dynamic row-level security?
3. Which DAX function is commonly used to implement dynamic row-level security based on the signed-in user?
4. What does object-level security restrict, as distinct from row-level security?
5. What is the purpose of a sensitivity label applied to a Power BI report?
6. Which workspace role would typically be assigned to a colleague who should only view, and not edit, a workspace's content?
7. In which Power BI Embedded scenario does the embedding application manage access using its own service identity rather than individual Power BI accounts?
8. What is the primary purpose of the on-premises data gateway?
9. Why does the gateway architecture not require inbound firewall ports to be opened?
10. What is the purpose of grouping multiple gateway installations into a gateway cluster?
11. Which gateway mode is intended for organizational, multi-user sharing rather than an individual user's own refreshes?
12. What are the three conventional stages of a Power BI deployment pipeline?
13. What does the usage metrics feature in the Power BI Service show a report owner?
14. What kind of records do organizational audit logs typically capture in Power BI?
15. Why is assigning workspace roles to security groups generally preferred over assigning them to individual users?

SGOU

Answers

1. Which rows of data a user is permitted to see within a shared report
2. Static RLS uses a fixed filter value per role, while dynamic RLS filters based on the identity of the signed-in user
3. USERPRINCIPALNAME

4. Which tables or columns a user can see at all, rather than which rows
5. To classify and protect content in line with the organization's information protection policy
6. Viewer
7. Embed for your customers
8. To securely connect the Power BI Service to data sources located behind an organization's firewall
9. Because the gateway initiates an outbound-only connection to the Power BI Service
10. To improve performance and reliability by distributing requests and providing failover
11. Standard mode
12. Development, Test, and Production
13. How many times a report has been viewed and by which users
14. Activities such as sharing, exporting, and permission changes across the tenant
15. Because access can then be managed by updating group membership rather than modifying the report or workspace directly

Assignments

1. Explain the different layers of security available in Power BI and describe what each layer controls.
2. Distinguish between static and dynamic row-level security, and discuss the circumstances under which each would be preferred.
3. Describe the two Power BI Embedded scenarios and explain how access control differs between them.
4. Explain the architecture and purpose of the on-premises data gateway, including why gateway clustering is recommended for production use.
5. Describe the purpose of deployment pipelines and explain how they support a controlled report life cycle from development to production.

SGOU

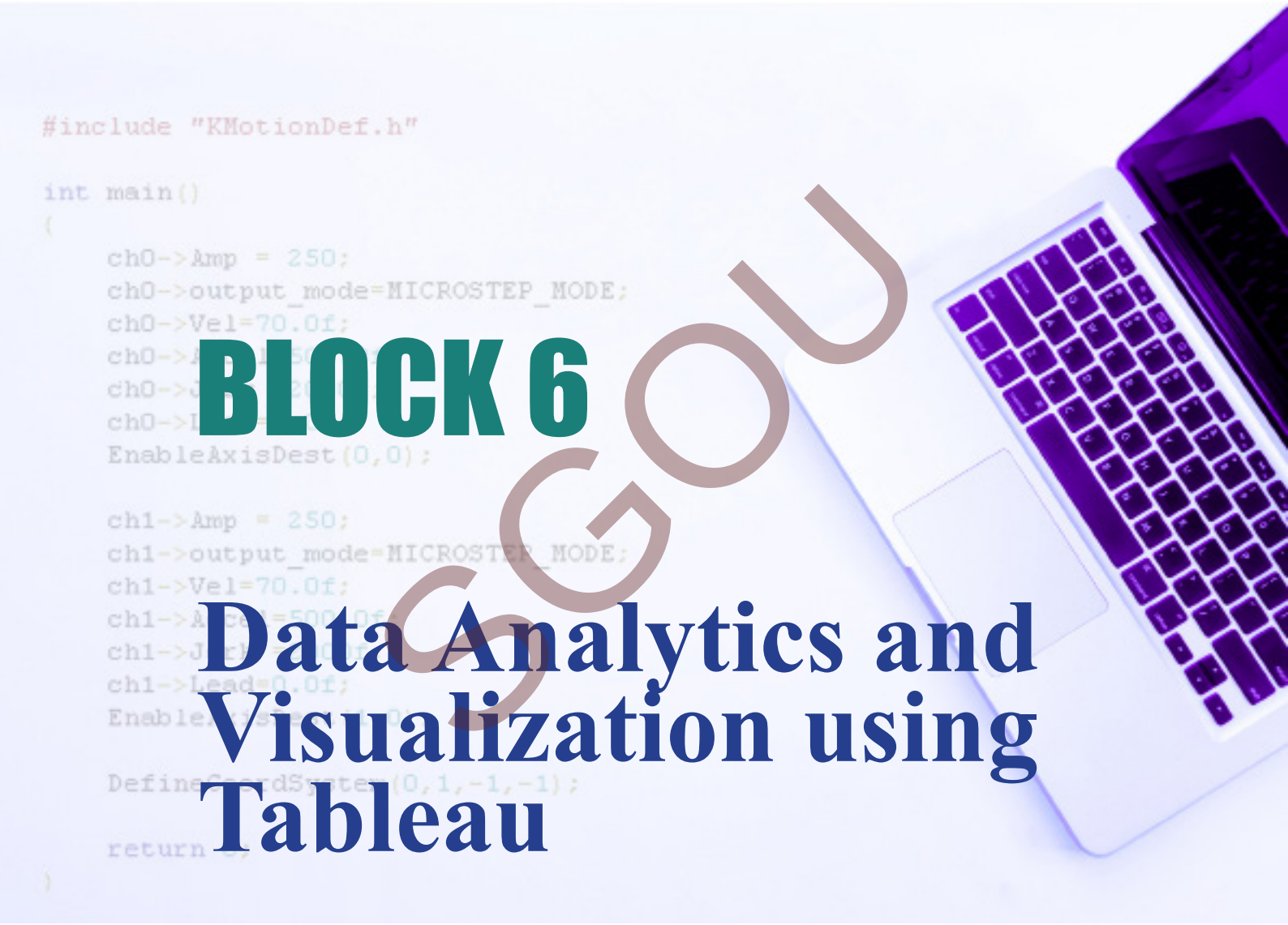
Suggested Reading

1. Powell, B. (2018). Mastering Microsoft Power BI: Expert techniques for effective data analytics and business intelligence. Packt Publishing.
2. Ferrari, A., & Russo, M. (2019). The Definitive Guide to DAX (2nd ed.). Microsoft Press.
3. Ahmed, M., & Pathan, A. K. (2020). Data analytics concepts, techniques and applications (1st ed.). Taylor and Francis.

Reference

1. Microsoft Corporation. Power BI Documentation - Security.
2. Microsoft Corporation. Power BI Documentation - On-premises Data Gateway.

SGOU



BLOCK 6

Data Analytics and Visualization using Tableau

SGOU



Tableau Basics and Data Preparation

Learning Outcomes

After completion of this unit, learner will be able to:

- explain what Tableau is and describe its role in modern data analytics and visualization
- distinguish between the different versions of Tableau and identify appropriate use cases for each
- identify the categories of data connectors available in Tableau and describe how each is used
- describe the process of loading data into Tableau, including the role of joins and relationships
- perform basic data cleaning operations within the Tableau environment
- apply fundamental data transformation techniques using Tableau Prep Builder
- recognize good practices for preparing data before it is used for visualization

SGOU

Prerequisite

This unit assumes a basic understanding of:

- Computer files, folders, and common file formats such as spreadsheets and text files
- Elementary data concepts such as rows, columns, records, and fields
- The general analytics and visualization ideas introduced in earlier blocks of this course

No prior experience with Tableau software is required. Concepts are explained at an introductory and conceptual level, and are illustrated with practical examples suitable for Open and Distance Learning.

Keywords

Tableau, Data Visualization, Data Source, Data Connector, Live Connection, Extract, Data Cleaning, Data Interpreter, Data Transformation, Tableau Prep, Pivot, Join, Union.

Discussion

6.1.1 Introduction to Tableau

Tableau is a visual analytics platform that allows users to connect to data, explore it interactively, and communicate findings through charts, maps, and dashboards. Rather than requiring users to write extensive code, Tableau relies on a drag-and-drop interface built on top of an internal query-generation engine known as VizQL, which automatically translates visual actions such as placing a field on a shelf into the underlying database queries needed to produce the corresponding chart.

The product originated from research at Stanford University on visualizing databases and was released commercially in 2003. Over time it grew into one of the most widely used self-service business intelligence tools, and it has, since 2019, been part of Salesforce. Its popularity stems from the fact that it lowers the technical barrier to exploring large and complex datasets: a learner who understands the structure of the data can begin building meaningful visual summaries within minutes, without first mastering a programming or query language.

In the context of this course, Tableau serves as a practical environment in which the statistical and analytical concepts studied in earlier blocks — such as distributions, correlation, and time-series trends — can be visualized and communicated. Before any of this exploration can begin, however, a learner must understand two things: which version of Tableau is appropriate for a given task, and how raw data is brought into Tableau and made analysis-ready. These two themes form the core of this unit.

Why Organizations Adopt Tableau

- It supports connections to a very wide range of data sources, from simple spreadsheets to large cloud data warehouses.
- Visualizations update interactively as filters, parameters, or selections change, which supports exploratory analysis.
- Dashboards can be published and shared so that non-technical stakeholders can interact with the same data without needing to write queries.
- It integrates data preparation, visualization, and dashboard publishing within a connected family of products, so a single dataset can move from raw file to a published, interactive report.

SGOU

A First Look at the Tableau Workspace

Although a detailed treatment of building visualizations is taken up in later units of this block, it is useful at this stage to recognize the basic building blocks of the Tableau Desktop workspace, since the same workspace is used to inspect and prepare data. A workbook is the file that a user saves and reopens, and it may contain any number of worksheets, dashboards, and stories. A worksheet is the canvas on which an individual chart is built by dragging fields from the Data pane onto shelves labelled Columns, Rows, Filters, Pages, and Marks. The Marks card controls how a field affects the appearance of a chart — for example, whether it changes colour, size, label, or shape. A dashboard combines several worksheets, along with text, images, and interactive controls such as filters, into a single composed view, while a story arranges a sequence of worksheets or dashboards to guide a viewer through a narrative. Recognizing this vocabulary early makes it easier to follow the data-connector and data-preparation discussion that follows, since every one of these elements ultimately depends on a properly connected and cleaned data source.

6.1.2 Versions of Tableau

Tableau is not a single application but a family of related products, each designed for a different stage of the analytics workflow or a different type of user. Understanding these versions is important because the choice of product determines how data can be connected, where a workbook can be stored, and who is able to view the resulting visualizations.

Tableau Desktop

Tableau Desktop is the primary authoring application used to connect to data, build visualizations, and design dashboards. It is installed on an individual's computer and offers the full range of connectors, calculation capabilities, and formatting options. Analysts and report developers typically use Desktop to create the workbooks that are later shared through Tableau Server or Tableau Cloud.

Tableau Public

Tableau Public is a free edition intended for learning and for sharing visualizations openly on the web. Workbooks created in Tableau Public are saved to a public online gallery rather than to a private location, which makes this edition suitable for students, hobbyists, and journalists who wish to publish work for open viewing, but unsuitable for confidential or organizational data.

Tableau Server

Tableau Server is an on-premises (self-hosted) platform that allows an organization to publish, manage, and share workbooks created in Tableau Desktop within a controlled environment. Because the organization hosts the server on its own infrastructure, it retains full control over data security, user access, and system configuration, which is often important for organizations with strict compliance requirements.

SGOU

Tableau Cloud (formerly Tableau Online)

Tableau Cloud provides the same publishing and sharing capabilities as Tableau Server, but is hosted and maintained by Tableau itself rather than by the organization. This removes the burden of installing and maintaining server infrastructure, and is often preferred by organizations that want to publish dashboards quickly without managing hardware.

Tableau Prep (Builder)

Tableau Prep Builder is a dedicated application for cleaning, shaping, and combining data before it is analyzed. It provides a visual, flow-based interface in which each step of a data transformation — such as filtering rows, splitting a column, or joining two tables — is represented as a node in a flow diagram. Prep Builder is discussed in more detail later in this unit, as it is central to the process of data preparation.

Tableau Reader

Tableau Reader is a free application that allows a user to open and interact with packaged workbooks (files with the extension .twbx) created in Tableau Desktop, without needing a Desktop licence. It provides limited interactivity — a Reader user can filter and explore an existing workbook but cannot create new visualizations or connect to new data sources.

Other Related Products

In addition to the core editions above, the Tableau family includes Tableau Mobile, which allows published dashboards to be viewed on smartphones and tablets, and Tableau Bridge, a lightweight utility that maintains a live connection between an organization's on-premises data and dashboards published to Tableau Cloud.

Table 6.1.1 Comparison of Tableau Editions

Edition	Hosting / Deployment	Primary Purpose	Typical User
Tableau Desktop	Installed on a local computer	Authoring visualizations and dashboards	Data analyst / report developer
Tableau Public	Cloud-hosted, publicly visible	Learning and open publishing of work	Student, hobbyist, journalist
Tableau Server	Self-hosted by the organization	Secure internal publishing and sharing	Enterprise IT-managed teams

SGOU

Tableau Cloud	Hosted by Tableau (SaaS)	Publishing and sharing without server maintenance	Organizations preferring managed hosting
Tableau Prep Builder	Installed on a local computer	Data cleaning, shaping, and combining	Data preparation specialist / analyst
Tableau Reader	Installed on a local computer	Viewing packaged workbooks	End user without authoring needs

As Table 6.1.1 illustrates, the choice of edition depends on whether the goal is to author content, host it securely within an organization, host it in a managed cloud environment, prepare data, or simply consume a finished dashboard. In practice, a typical analytics workflow uses several of these products together: data is cleaned in Tableau Prep, visualizations are authored in Tableau Desktop, and the finished dashboard is published to Tableau Server or Tableau Cloud for wider audiences to view.

6.1.3 Data Connectors in Tableau

Before any visualization can be built, Tableau must be connected to a source of data. A data connector is the mechanism through which Tableau communicates with a particular type of data source, translating the structure of that source — whether a spreadsheet, a database, or a cloud service — into a form that Tableau's engine can query. Tableau ships with a large library of built-in connectors, which can be broadly grouped into the following categories.

File-Based Connectors

File-based connectors allow Tableau to read data directly from files stored on a computer or network drive. Common examples include Microsoft Excel workbooks, comma-separated value (CSV) and other text files, PDF documents containing tables, JSON files, statistical files such as those produced by SPSS, SAS, or R, and spatial file formats such as shapefiles used for geographic mapping. File-based connectors are typically the starting point for learners because the data is self-contained and does not require network or server access.

Server and Database Connectors

Server-based connectors allow Tableau to connect directly to relational database management systems such as MySQL, Microsoft SQL Server, Oracle, and PostgreSQL. These connectors typically require the user to supply a server address, a database or schema name, and valid credentials. Because the data remains on the database server rather than being copied into a file, this category of connector is well suited to large, frequently updated datasets.

SGOU

Cloud and Web-Based Connectors

As organizations increasingly store data in cloud platforms, Tableau provides native connectors to cloud data warehouses and services such as Google BigQuery, Amazon Redshift, Snowflake, Microsoft Azure SQL Database, Google Sheets, and Salesforce. In addition, the Web Data Connector framework allows Tableau to retrieve data made available through a web-based application programming interface (API), extending Tableau's reach to sources that do not have a dedicated built-in connector.

Table 6.1.2 Examples of Connectors by Category

Category	Representative Examples	Typical Scenario
File-Based	Microsoft Excel, Text/CSV, PDF, JSON, Spatial files, Statistical files	A single analyst working with locally stored files
Server / Database	MySQL, Microsoft SQL Server, Oracle, PostgreSQL	Departmental reporting from an organizational database
Cloud / Web-Based	Google BigQuery, Amazon Redshift, Snowflake, Google Sheets, Salesforce, Web Data Connector	Enterprise analytics on data warehoused in the cloud

The examples in Table 6.1.2 are not exhaustive — Tableau periodically extends its connector library — but they illustrate the general principle that the choice of connector is determined by where the underlying data physically resides, rather than by the type of analysis to be performed.

Live Connection versus Extract

Regardless of the connector used, Tableau offers two fundamentally different ways of working with the connected data. A live connection sends queries directly to the original data source each time a visualization is built or refreshed, so the visualization always reflects the current state of the source data. An extract, by contrast, creates a compressed, optimized copy of the data (in Tableau's own .hyper format) that is stored locally and used instead of repeatedly querying the original source.

Aspect	Live Connection	Extract
Data Freshness	Always reflects the current source	Reflects the data as of the last refresh
Performance	Depends on source system performance	Often faster, since data is optimized locally

SGOU

Source Load	Every interaction queries the source	Source is queried only during extraction/refresh
Offline Use	Requires an active connection to the source	Can be used without a live connection
Best Suited For	Frequently changing, real-time data	Large datasets or sources with limited query capacity

Choosing between a live connection and an extract is one of the first practical decisions a Tableau user makes when preparing data, and this choice directly affects the performance and freshness of every dashboard built from that connection.

Choosing an Appropriate Connector and Connection Type

In practice, the choice of connector and connection type is guided by three questions. First, where does the data currently live — as a file on a computer, inside an organizational database, or within a cloud platform? This determines which connector category from Table 6.1.2 is relevant. Second, how frequently does the underlying data change, and how important is it that a dashboard reflect the very latest values? Data that changes every few minutes, such as a live operational dashboard, generally favours a live connection, whereas data that is updated daily or weekly, such as monthly sales summaries, is usually well served by a scheduled extract. Third, how large is the dataset, and how much query load can the source system tolerate? Very large datasets, or source systems already under heavy load from other applications, often benefit from an extract, since it shifts the burden of repeated querying away from the original system. Working through these three questions before building a single worksheet helps avoid the common mistake of leaving the default connection type unexamined until performance problems appear later.

6.1.4 Data Loading in Tableau

Once an appropriate connector has been chosen, the next step is to load the data into Tableau's data source page, where it can be inspected, arranged, and combined before analysis begins.

Steps in Connecting to a Data Source

1. Open Tableau Desktop and select the appropriate connector from the Connect pane (for example, Microsoft Excel, Text File, or a specific database connector).
2. Provide the required details, such as a file path, or a server address and login credentials for a database connection.
3. Once the connection succeeds, Tableau opens the data source page, listing the available tables, sheets, or objects found within that source.

SGOU

4. Drag one or more tables onto the canvas area of the data source page to include them in the data model.
5. If more than one table is used, define how the tables relate to one another using relationships or joins.
6. Review the data preview grid at the bottom of the page to confirm that fields, data types, and sample values look correct.
7. Navigate to a new worksheet to begin building visualizations from the connected data.

Understanding the Data Source Page

The data source page is organized into three main areas. The connections pane, on the left, lists the data source(s) currently connected and the tables available within each. The canvas, in the centre, is where tables are dragged and arranged, and where relationships or joins between tables are defined visually by drawing connecting lines. The preview grid, at the bottom, displays a sample of the actual rows and columns so that the learner can verify that the data has loaded correctly before proceeding further.

Joins, Unions, and Relationships

When a visualization requires fields from more than one table, Tableau provides several ways to combine them. A join combines tables side by side based on a common field, following the familiar inner, left, right, and full outer join logic used in relational databases. A union stacks tables on top of one another, appending the rows of one table to another when both share a similar structure — this is useful, for example, when combining twelve monthly sales files that each share the same columns. A relationship, which is the default and most flexible option in modern versions of Tableau, keeps tables logically connected without physically merging them, allowing Tableau to combine data at the correct level of detail automatically depending on which fields are used in a given visualization.

Data Types and Field Roles

Once loaded, every field in the data source is automatically assigned a data type (such as text/string, number, date, date and time, Boolean, or geographic role) and a field role. Tableau classifies each field as either a dimension, meaning a categorical field used to slice or group data (such as Region or Product Category), or a measure, meaning a numeric field that can be aggregated (such as Sales or Quantity). Fields are further classified as discrete or continuous: discrete fields, shown in blue, create distinct headers or labels, while continuous fields, shown in green, create a continuous axis. Correctly recognizing and, where necessary, correcting these classifications is an essential part of preparing data before visualization, since an incorrectly typed field can prevent a chart from being built as intended.

SGOU

Why does Tableau use its own extract format?

Tableau extracts are stored in a proprietary, column-oriented file format known as .hyper. This format is optimized specifically for the kind of aggregation and filtering queries that dashboards generate, which is why extracts often make a dashboard feel noticeably faster than an equivalent live connection to a large, general-purpose database. This information is provided for background interest and does not need to be memorized in technical detail.

6.1.5 Data Cleaning in Tableau

Real-world data is rarely ready for immediate analysis: it often contains missing values, inconsistent formatting, duplicate records, or fields that have been assigned the wrong data type. Data cleaning refers to the set of operations that correct these issues so that subsequent analysis and visualization are accurate and trustworthy. Tableau provides several built-in facilities for basic data cleaning directly on the data source page and within worksheets.

Renaming and Hiding Fields

Field names imported from a source file or database are not always clear to an end user. Tableau allows fields to be renamed to more descriptive labels, and allows fields that are not needed for analysis to be hidden from view, reducing clutter without deleting the underlying data.

Correcting Data Types

Because Tableau infers data types automatically when a source is connected, it can occasionally misclassify a field — for example, treating a numeric postal code as text, or a date stored as text rather than as an actual date field. The data type icon beside each field name can be used to manually correct such misclassifications, which is important because calculations, sorting, and date-based visualizations all depend on fields having the correct type.

Handling Missing (Null) Values

Missing values in a dataset can distort aggregations such as averages and totals if they are not handled deliberately. Tableau allows null values to be filtered out of a view, replaced with a specific default value, or indicated explicitly using a special mark, depending on what is appropriate for the analysis being performed.

Removing Duplicate Records

Duplicate rows — for instance, the same transaction recorded twice due to a data-entry or file-merging error — can inflate totals and mislead an analysis. While Tableau Desktop does not remove duplicates directly within a live connection, this can be addressed either before the extract is created or, more thoroughly, using Tableau Prep Builder, which provides a dedicated duplicate-removal step.

SGOU

Splitting and Grouping Fields

A single field sometimes contains information that would be more useful if separated into multiple fields — for example, a Full Name field that combines first and last names. Tableau's Split feature can automatically separate such a field based on a common delimiter such as a space or comma. Conversely, related but inconsistently labelled values (such as "USA", "U.S.A.", and "United States") can be combined into a single consistent category using Tableau's Group feature.

Using the Data Interpreter

Spreadsheets designed for human reading — with title rows, merged cells, subtotal rows, or multiple small tables on one sheet — are often difficult for software to read correctly. When connecting to such a file, Tableau offers a Data Interpreter option that automatically detects and removes these irregularities, producing a cleaner, more analysis-ready table. Learners are encouraged to always inspect the preview grid after enabling the Data Interpreter, since automatic detection is not always perfect for unusually formatted sheets.

Worked Example: Correcting a Misread Date Field

Suppose a CSV file records a Transaction Date column in the format "04/07/2026", and upon connection Tableau assigns this field a text (string) data type rather than a date type — a common issue when a source file mixes date formats or includes stray text in the column. Because the field is text, Tableau cannot use it to build a time-series chart or apply date-based filtering. The correction takes three steps. First, find the field in the Data pane and note the small data-type icon beside its name. Second, click the icon and choose the correct type, in this case Date. Third, if Tableau cannot read the values automatically because the order of day and month is unclear, use a calculated field with a date-parsing function to set the format explicitly before converting the result to a date type. Once corrected, the field can be placed on the Columns or Rows shelf to build a continuous time-series view, and this small correction — performed once during data preparation — benefits every subsequent worksheet built from the same data source.

6.1.6 Data Transformation Using Tableau Prep Builder

While basic cleaning can be performed directly within Tableau Desktop, more substantial data transformation — particularly when multiple files or tables must be reshaped, combined, and cleaned in a repeatable sequence — is best handled in Tableau Prep Builder. Prep Builder represents a data preparation task as a flow: a sequence of connected steps, each of which performs one clearly defined operation on the data, with the results visible at every stage.

Structure of a Prep Flow

A typical Prep flow begins with one or more input steps, where raw data sources are connected in the same way as in Tableau Desktop. This is followed by a series of transformation steps, and concludes with one or more output steps, which save the

SGOU

cleaned and transformed data either as a Tableau extract or back into a database or file format for use elsewhere.

Common Transformation Steps

- **Clean step:** Provides an interactive grid for renaming fields, correcting types, filtering rows, removing nulls, and standardizing inconsistent text values, all with an immediate visual preview of the effect on the data.
- **Aggregate step:** Summarizes data by grouping on selected fields and calculating totals, averages, counts, or other summary statistics — useful when raw transaction-level data must be reduced to a higher level of detail before analysis.
- **Pivot step:** Reshapes data between wide and long formats. Pivoting columns to rows converts several separate columns (for example, monthly sales columns for January through December) into two columns — one holding the category (month) and one holding the value (sales) — which is often the structure required for time-series visualization. Pivoting rows to columns performs the reverse transformation.
- **Join step:** Combines two data sources side by side based on a common field, following the same join logic used on the Tableau Desktop data source page, but with an interactive preview that highlights unmatched rows so that mismatches can be diagnosed easily.
- **Union step:** Stacks multiple similarly structured tables into one, which is particularly useful for combining a set of periodic files — such as monthly or regional extracts — into a single consolidated table.
- **Output step:** Defines where the finished, transformed data should be saved, such as a Tableau data extract (.hyper file), a published data source, or a file in a common format for use by other tools.

Illustrative Example

Consider an organization that receives twelve separate CSV files, one for each month, recording daily sales across several regional stores, with a few files containing extra blank columns and inconsistent region names. In Tableau Prep Builder, an analyst would connect all twelve files as inputs, use a Union step to stack them into a single table, apply a Clean step to remove blank columns and standardize region names using the Group feature, and finally use an Aggregate step to summarize sales by region and month. The resulting output could then be saved as a Tableau extract, ready to be opened directly in Tableau Desktop for visualization. This example illustrates how several of the individual techniques discussed earlier in this unit — connecting to files, combining tables, cleaning inconsistent values, and choosing between a live connection and an extract — come together in a realistic data preparation workflow.

Documenting a Prep Flow

Because a Prep flow may eventually contain a dozen or more steps, Tableau Prep Builder allows each step to be given a clear, descriptive name (for example, "Remove

SGOU

blank columns" rather than the default "Clean 1"), and allows a short description to be attached to the flow as a whole. This is not merely a cosmetic convenience: in an Open and Distance Learning or organizational setting, a data preparation flow is frequently reused, handed over to another analyst, or revisited months after it was first built, and clear naming is often the only practical documentation available at that point. Learners should adopt the habit of naming each step as it is created, rather than leaving this task until the end of the flow.

6.1.7 Good Practices for Data Preparation

The quality of any visualization or dashboard is limited by the quality of the data behind it. Before moving from data preparation to visualization, it is good practice to verify the following:

- **Accuracy :** Do sample values in the preview grid match what is expected from the original source?
- **Completeness :** Have missing values been identified and handled appropriately, rather than silently ignored?
- **Consistency :** Are categorical values (such as region or product names) spelled and labelled consistently throughout the dataset?
- **Correct Granularity :** Does each row represent the intended level of detail (for example, one row per transaction rather than an unintended mixture of transaction-level and summary rows)?
- **Correct Field Roles and Types :** Are dimensions, measures, and data types assigned in a way that matches how the field will actually be used?
- **Documentation :** Have renamed fields, applied filters, and transformation steps been named clearly enough that another person could understand the flow later?

Data preparation is rarely a single, one-time step; as new issues are discovered during visualization — such as an unexpected outlier or a mismatched join — analysts typically return to the data source page or the Prep flow to make further corrections. Treating data preparation as an iterative process, rather than a task to be finished once and forgotten, is one of the most important habits for producing reliable Tableau visualizations.

Data Preparation and the Rest of This Block

The connectors, loading process, and cleaning and transformation techniques introduced in this unit are not isolated skills. Unit 2 of this block builds directly on a properly prepared data source to create calculated fields and measures; Unit 3 assumes that dimensions and measures are already correctly classified when building charts, maps, and dashboards; and the case study in Unit 4 requires a dataset that has already been connected, cleaned, and shaped using the techniques covered here.

SGOU

Recap

- Tableau is a visual analytics platform that uses a drag-and-drop interface, built on the VizQL query-generation engine, to help users explore and communicate data.
- The Tableau product family includes Desktop, Public, Server, Cloud, Prep Builder, and Reader, each suited to a different stage of the analytics workflow or type of user.
- Data connectors allow Tableau to communicate with file-based sources, server/database sources, and cloud/web-based sources.
- A live connection always queries the original data source, while an extract works from a locally stored, optimized copy of the data.
- Loading data involves connecting to a source, arranging tables on the data source page, and defining relationships, joins, or unions where more than one table is used.
- Every field loaded into Tableau is assigned a data type and a role — dimension or measure, discrete or continuous — which can and often should be reviewed and corrected.
- Data cleaning within Tableau includes renaming and hiding fields, correcting data types, handling missing values, removing duplicates, splitting fields, and using the Data Interpreter.
- Tableau Prep Builder provides a flow-based environment for more substantial transformation, using steps such as Clean, Aggregate, Pivot, Join, and Union.
- Good data preparation practice involves checking data for accuracy, completeness, consistency, correct granularity, correct field roles, and clear documentation, as an ongoing rather than one-time activity.

SGOU

Objective Type Questions

1. What is Tableau?
2. Which query-generation engine underlies Tableau's drag-and-drop interface?
3. Which edition of Tableau is used mainly for authoring visualizations on a local computer?
4. Which free edition of Tableau publishes workbooks to a public online gallery?
5. What distinguishes Tableau Cloud from Tableau Server in terms of hosting?

6. Which application is used to open a packaged workbook without a Desktop licence?
7. What is a data connector in Tableau?
8. Which type of connection always queries the original data source directly?
9. What is an extract in Tableau?
10. What is the difference between a join and a union when combining tables?
11. How does Tableau classify a categorical field such as Region?
12. What colour represents a continuous field in Tableau's field list?
13. Which Tableau Desktop feature automatically cleans a poorly formatted spreadsheet on connection?
14. Which Tableau Prep step reshapes columns into rows, or rows into columns?
15. In what file format does Tableau typically store a data extract?

Answers

1. A visual analytics platform for exploring and communicating data
2. VizQL
3. Tableau Desktop
4. Tableau Public
5. Tableau Cloud is hosted by Tableau; Tableau Server is self-hosted
6. Tableau Reader
7. A mechanism that connects Tableau to a specific type of data source
8. Live connection
9. A locally stored, optimized copy of the data
10. A join combines tables side by side; a union stacks tables on top of one another
11. As a dimension
12. Green
13. Data Interpreter
14. Pivot
15. .hyper

SGOU

Assignments

1. Explain the purpose of each of the six main Tableau product editions discussed in this unit, and identify which edition would be most appropriate for (a) a student publishing a project online, and (b) a large enterprise that must keep all data on its own servers.
2. Distinguish between a live connection and an extract, and discuss the circumstances under which each would be the more suitable choice.
3. Describe the steps involved in connecting to and loading a multi-sheet Excel workbook into Tableau, including how relationships between sheets can be defined.
4. Explain, with examples, at least four data cleaning operations that can be performed within Tableau, and the type of data problem that each operation addresses.
5. Describe the structure of a Tableau Prep flow and explain the role of the Clean, Aggregate, Pivot, Join, and Union steps using a suitable example.
6. Discuss why data preparation should be treated as an iterative process rather than a single, one-time task, drawing on at least two data-quality checks discussed in this unit.

Suggested Reading

1. Meier, M. (2023). Mastering Tableau 2023 (4th ed.). Packt Publishing.
2. Loth, A. (2019). Visual Analytics with Tableau. Wiley.
3. Murray, D. G. (2016). Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software (2nd ed.). Wiley.

Reference

1. Tableau Software, LLC. Tableau Desktop and Tableau Prep Deployment Guide.
2. Tableau Software, LLC. Where's the Installer? — Product Download Documentation.
3. Tableau Software, LLC. All Technical Specifications — Tableau Product Family.

SGOU



Data Modelling and Calculations

Learning Outcomes

After completion of this unit, learner will be able to:

- describe the purpose of a data model in Tableau and distinguish its logical and physical layers
- differentiate between relationships, joins, and data blending as methods of combining data
- explain the concepts of cardinality and referential integrity and relate them to accurate aggregation
- construct and interpret Level of Detail (LOD) expressions using FIXED, INCLUDE, and EXCLUDE
- differentiate between table calculations and LOD expressions and select the appropriate technique for a given analytical need

Prerequisite

This unit assumes that the learner has already completed Unit 1 of this block and is comfortable with:

- Connecting to a data source and loading one or more tables into Tableau
- The distinction between dimensions and measures, and between discrete and continuous fields
- Basic aggregate functions such as sum, average, and count, as introduced in earlier blocks of this course
- Elementary algebraic notation, since calculated fields are written as formulas

Building on the connected and cleaned data source discussed in Unit 1, this unit introduces the two techniques that most strongly determine whether a Tableau workbook produces correct numbers: a well-designed data model, and carefully written calculations.

SGOU

Keywords

Logical Layer, Physical Layer, Relationship, Join, Data Blending, Cardinality, Referential Integrity, Level of Detail (LOD), FIXED, INCLUDE, EXCLUDE

Discussion

6.2.1 Introduction to Data Modelling in Tableau

A data model is the structure that defines how the tables connected to a Tableau workbook relate to one another, and at what level of detail each piece of information is recorded. Whenever a visualization draws on more than one table — for example, an Orders table and a separate Customers table — Tableau must know how a row in one table corresponds to a row in another before it can combine fields from both into a single chart. Getting this structure right is not a minor technical formality: an incorrectly modelled combination of tables is one of the most common causes of visibly wrong numbers in a dashboard, such as sales totals that are silently doubled or customer counts that are inflated.

This unit builds directly on the data loading skills covered in Unit 1. There, the focus was on connecting to and cleaning individual sources; here, the focus shifts to how multiple prepared sources are combined and summarized correctly, and how new, derived fields — calculated fields — are constructed from the fields that already exist in the model.

It is useful to think of data modelling and calculation as addressing two related but distinct questions. Data modelling answers the question, "how do the tables I have connected relate to one another, and at what level of detail is each fact recorded?" Calculation answers a different question: "given a correctly modelled set of fields, how do I derive a new value — a category, a ratio, a running total, or a per-customer figure — that is not already present as a raw column?" A dashboard can go wrong at either stage: an incorrectly modelled relationship can inflate or shrink totals before any calculation is even written, while a poorly written calculation can produce misleading figures even from a perfectly modelled data source. This unit therefore addresses the two topics in sequence, first ensuring that the underlying model is sound, and then building calculations on top of it.

6.2.2 The Tableau Data Model: Logical and Physical Layers

Modern versions of Tableau organize a data source into two layers, visible on the data source page. The logical layer consists of one or more logical tables, shown as boxes on the canvas, which are connected to one another using relationships. Each logical table preserves its own natural level of detail — for instance, an Orders logical table keeps one row per order line, while a Customers logical table keeps one row per customer, and Tableau automatically aggregates data to the correct level whenever fields from different logical tables are used together in a view.

SGOU

Within a single logical table, however, one or more physical tables may be combined using joins or unions, forming the physical layer. This distinction matters because the two layers behave differently. Combinations in the physical layer (joins and unions) create a single merged table at query time. They can change the number of rows if the join is not one-to-one. Combinations across the logical layer (relationships) do not merge rows at all. Instead, they calculate results on demand at whatever level of detail a worksheet needs.

Table 6.2.1 The comparison between logical layer and physical layer

Aspect	Logical Layer	Physical Layer
Connected Using	Relationships	Joins or Unions
Row-Level Merging	No — each table keeps its own grain	Yes — rows are physically combined
Risk of Row Duplication	Low, since aggregation adapts to the view	Higher, if cardinality is not one-to-one
Typical Use	Combining tables with different levels of detail (e.g., Orders and Customers)	Combining tables that share the same grain (e.g., twelve monthly extracts)

6.2.3 Relationships, Joins, and Data Blending Compared

Tableau offers three distinct ways of bringing fields from more than one table into a single analysis, and choosing the right one depends on how the tables are related and where they are stored.

6.2.3.1 Relationships

A relationship is a flexible, non-merging connection between two logical tables, defined by matching one or more common fields (for example, Customer ID appearing in both an Orders table and a Customers table). Because no physical merge takes place, each table retains its own row count, and Tableau computes the correct aggregation automatically depending on which fields from which table are placed in a given worksheet. Relationships are the default and generally recommended way to combine tables that describe different things, such as orders and the customers who placed them.

6.2.3.2 Joins

A join, defined within the physical layer, merges two tables row by row based on a matching field, following the familiar inner, left, right, and full outer join types used in relational databases. A join is appropriate when two tables share the same grain — for instance, joining a table of daily website visits to a separate table of daily marketing spend, where both are recorded once per day. If the join field is not unique in one of the

SGOU

tables, however, matching rows are duplicated for every match, a problem often called a fan-out, which can silently inflate sums and counts.

6.2.3.3 Data Blending

Data blending combines data from two entirely separate data sources — for example, a spreadsheet and a database that cannot be joined or related directly because they are connected independently. In blending, one data source is designated as primary and the other as secondary; Tableau aggregates the secondary source up to the level of one or more common linking fields before combining it with the primary source in the view. Because blending works on already-aggregated results, it is less flexible than relationships or joins and is generally used only when the data genuinely cannot be combined through the data source page.

Table 6.2.2 Comparing Relationships, Joins, and Data Blending

Aspect	Relationship	Join	Data Blending
Where Defined	Data source page (logical layer)	Data source page (physical layer)	Within a worksheet, across two data sources
Row Merging	No	Yes	No (secondary is pre-aggregated)
Best Suited For	Tables with different grains, same connection	Tables with the same grain, same connection	Tables from separate, unrelated connections
Risk of Duplication	Low	Possible if cardinality is not 1:1	Not applicable (aggregated first)

6.2.4 Cardinality and Referential Integrity

When two logical tables are connected by a relationship, Tableau allows the modeller to specify, or lets Tableau automatically detect, two important properties of that connection: cardinality and referential integrity.

6.2.4.1 Cardinality

Cardinality describes how many matching rows exist on each side of a relationship for a given value of the linking field. A one-to-one relationship means each value of the linking field appears at most once in both tables. A one-to-many relationship means one value on one side (for example, one customer) can match many rows on the other side (for example, many orders placed by that customer). A many-to-many relationship means a linking value can repeat on both sides. Correctly identifying cardinality allows Tableau to aggregate measures at the appropriate level automatically, without the learner needing to write special-case formulas.

SGOU

6.2.4.2 Referential Integrity

Referential integrity describes whether every value of the linking field on one side of a relationship is guaranteed to have a match on the other side. Tableau offers three settings for this: "all records match", used when every row is known to have a corresponding row in the other table; "some records may not match", used when unmatched rows are possible (for example, a promotional customer record that has not yet placed an order); and "related tables are unrelated", used when there is no guarantee of matching at all. Selecting the correct referential integrity setting can improve query performance, since Tableau can skip unnecessary matching logic when it is told that every record is guaranteed to match.

Why Cardinality Matters in Practice

A common classroom example shows the risk of ignoring cardinality. Suppose an Orders table (one row per order line) is joined, instead of related, to a Products table. If the Products table has two rows for the same product code because of a data-entry error, every order line for that product is duplicated in the joined result. This silently doubles its contribution to total sales. Relationships are more forgiving of this kind of error because they do not physically merge rows, which is one reason they are the recommended default when combining tables of different grains.

6.2.5 Building a Data Model: A Step-by-Step View

1. On the data source page, drag the first table (for example, Orders) onto the canvas to create the first logical table.
2. Drag a second table (for example, Customers) onto the canvas near the first; Tableau displays a relationship line connecting the two if it detects a likely matching field.
3. Click the relationship line (or the noodle connecting the two tables) to review or edit the fields used to match rows between the tables.
4. If needed, expand the cardinality and referential integrity options to confirm or correct how Tableau has interpreted the relationship between the two tables.
5. Repeat the process to add further tables, building up a network of logical tables that together represent the full data model for the analysis.
6. Use the preview grid to spot-check that row counts and sample values look reasonable after each table is added, rather than waiting until the entire model is built.

This incremental approach — adding one table at a time and checking the preview after each step — makes it far easier to identify exactly which addition caused an unexpected change in row counts, compared with adding several tables at once and only checking the result at the end.

SGOU

6.2.5.1 Worked Example: Modelling Orders, Customers, and Products

Consider a retail dataset consisting of three tables: Orders (one row per order line, containing an Order ID, a Customer ID, a Product ID, and a Sales amount), Customers (one row per customer, containing a Customer ID and a Region), and Products (one row per product, containing a Product ID and a Category). Each of these tables has a different grain — Orders is at the order-line level, while Customers and Products are each at their own one-row-per-entity level — which makes relationships the appropriate way to connect them, rather than joins.

The Orders table is first dragged onto the canvas as the starting logical table. Customers is then dragged alongside it, and Tableau proposes a relationship on the Customer ID field, which is confirmed after checking that the field names and data types match on both sides. Products is added in the same way, related on Product ID. Once all three tables are connected, one worksheet can combine Region (from Customers), Category (from Products) and SUM(Sales) (from Orders) without the learner writing a single join clause. Relationships also preserve each table's own grain. So the customer and product counts stay accurate even though Orders has many more rows than Customers or Products. This worked example illustrates, in a concrete setting, why the cardinality and referential integrity concepts discussed above matter well beyond the abstract definitions: they determine whether Region- and Category-level totals in the finished dashboard can be trusted.

6.2.6 Introduction to Calculations in Tableau

Once a data model is in place, the fields available for analysis are not limited to those that existed in the original source. A calculated field is a new field defined by writing a formula that combines existing fields, constants, and built-in functions. Calculated fields are created using the Calculation Editor, opened from the Data pane, where a formula is typed and validated before the new field is added alongside the other fields available for building views.

Tableau's calculation language has several broad categories of built-in functions. Numeric functions (such as ROUND and ABS) are for arithmetic. String functions (such as LEFT, MID and CONTAINS) are for working with text. Date functions (such as DATEDIFF and DATEPART) are for working with dates and times. Logical functions (such as IF, CASE and IIF) are for conditional logic. Aggregate functions (such as SUM, AVG and COUNTD) are for summarizing groups of rows. Type-conversion functions (such as STR and INT) are for changing a field from one data type to another. A learner does not need to memorize every function to begin writing useful calculations; the calculation editor lists available functions with a short description, and most calculations used in introductory analysis draw on a fairly small, recurring set of them.

Creating a Basic Calculated Field

1. In the Data pane, right-click and choose to create a new calculated field (or use the corresponding menu option).

SGOU

2. Give the field a clear, descriptive name, since this name will appear in the Data pane exactly as typed.
3. Type a formula in the editor, for example: [Sales] - [Cost], to compute a simple profit figure.
4. Confirm that the small status indicator at the bottom of the editor shows the formula is valid before saving.
5. Save the calculation; the new field then appears in the Data pane and can be dragged onto shelves like any other field.

Table 6.2.3 Common Function Categories Used in Calculated Fields

Category	Example Functions	Typical Purpose
Numeric	ROUND, ABS, SQRT	Perform arithmetic and rounding on numeric fields
String	LEFT, MID, CONTAINS, TRIM	Extract, search, or clean text values
Date	DATEDIFF, DATEPART, DATEADD	Calculate intervals or extract parts of a date
Logical	IF, CASE, IIF	Apply conditional logic to create categories or flags
Aggregate	SUM, AVG, COUNTD, MIN, MAX	Summarize many rows into a single value
Type Conversion	STR, INT, DATE	Convert a field from one data type to another

A learner does not need to master every function in every category before writing useful calculations. In practice, a small, recurring set of functions — an IF or CASE statement for categorization, a handful of aggregate functions for measures, and occasionally a date function for time-based analysis — accounts for the large majority of calculated fields used in typical coursework and workplace dashboards.

6.2.7 Calculated Dimensions and Calculated Measures

Not all calculated fields behave the same way once created, and it is important to distinguish between two broad kinds.

6.2.7.1 Calculated Dimensions (Calculated Columns)

A calculated dimension, sometimes referred to informally as a calculated column, is computed independently for every row of the data and typically produces a categorical result used to group or label data. A common example is a formula such as: IF [Age] < 18 THEN "Minor" ELSEIF [Age] < 60 THEN "Adult" ELSE "Senior" END, which

SGOU

creates a new Age Group field that can be used exactly like any other dimension — for instance, to colour a chart or to filter a view.

Calculated Measures

A calculated measure combines one or more existing measures using an aggregate function, and is evaluated after the underlying rows have been summarized, not row by row. A common example is a Profit Ratio field defined as SUM([Profit]) / SUM([Sales]), which must be written using the aggregate SUM function precisely because dividing two already-summarized totals is different from averaging many individual row-level ratios; the two approaches can give noticeably different results, particularly when order sizes vary widely.

A Common Beginner Mistake

A frequent error among new Tableau users is writing [Profit] / [Sales] instead of SUM([Profit]) / SUM([Sales]) when building a ratio measure. Without the aggregate functions, Tableau computes the ratio for every individual row before any aggregation happens, and then averages these individual ratios when the view is aggregated — which is mathematically different from, and usually less meaningful than, the ratio of the two totals. Always consider whether a calculation should be aggregated before or after the division, sum, or other operation is applied.

Recognizing whether a calculation is intended to behave like a dimension (evaluated per row) or like a measure (evaluated on aggregated totals) is one of the most important conceptual skills in this unit, since it directly determines how the resulting field can be used in a view.

6.2.8 Level of Detail (LOD) Expressions

Ordinary calculated measures are aggregated at whatever level of detail is currently defined by the dimensions placed in a view — a property sometimes described informally as the calculation being at "the level of the view". Level of Detail, or LOD, expressions allow a calculation to be computed at a level of detail that is coarser than, finer than, or entirely independent of the dimensions currently in the view. Tableau provides three LOD keywords.

6.2.8.1 FIXED

A FIXED expression computes its result using only the dimensions explicitly listed inside the expression, regardless of which other dimensions are present in the view. For example, {FIXED [Customer ID] : SUM([Sales])} calculates total sales per customer once, and this value remains the same even if the view is later broken down by additional dimensions such as Region or Order Date.

6.2.8.2 INCLUDE

An INCLUDE expression computes its result using the dimensions in the view plus one or more additional dimensions specified in the expression, even if those additional

SGOU

dimensions are not themselves placed in the view. For example, {INCLUDE [Order ID] : COUNTD([Product])} can be used to calculate the average number of distinct products per order, by first counting products at the order level and then allowing Tableau to average that count across whatever dimensions appear in the outer view.

6.2.8.3 EXCLUDE

An EXCLUDE expression removes one or more dimensions from the calculation that would otherwise be included because they are present in the view. For example, if a view is broken down by both Region and Year, {EXCLUDE [Year] : SUM([Sales])} calculates total sales by region only, ignoring the Year breakdown, which is useful for comparing a single year's figures against an unchanging regional total.

Table 6.2.4 Summary of LOD Expression Types

Keyword	Effect on Level of Detail	Typical Use Case
FIXED	Uses only the dimensions listed, ignoring the view	Customer lifetime totals, first purchase date
INCLUDE	Adds extra dimensions beyond the view	Average number of products per order
EXCLUDE	Removes a dimension that is present in the view	Comparing a detail figure against a coarser total

SGOU

Worked Example: Total Sales per Customer Using FIXED

Suppose a view lists individual order lines, broken down by Order Date and Product Category, and the analytical goal is to add a column showing each customer's total sales across all of their orders, so that individual order lines can be compared against that customer's overall spending. An ordinary calculated measure such as SUM([Sales]) would simply repeat the sales value of each order line, since it aggregates only to the level of detail already present in the view. Instead, the calculation {FIXED [Customer ID] : SUM([Sales])} computes total sales for each customer once, using only Customer ID, and then repeats that fixed total alongside every order line for that customer regardless of which other dimensions — Order Date, Product Category, or anything else — are subsequently added to the view. This is precisely the behaviour that distinguishes an LOD expression from an ordinary aggregate measure, and it is why FIXED expressions are commonly used for tasks such as customer lifetime value, first purchase date, or cohort-style analysis.

6.2.9 Table Calculations versus LOD Expressions

Table calculations are a separate mechanism that also modifies how a value is computed relative to the rest of the view, but they operate very differently from LOD expressions. A table calculation, such as a running total, percent of total, or rank, is computed after the underlying query has already returned aggregated results to Tableau

, that is, it works on the values already visible in the view, based on settings such as the order (addressing) and grouping (partitioning) of the table. An LOD expression, by contrast, is evaluated as part of the underlying query sent to the data source, before the results reach the view at all.

Table 6.2.5 Table Calculations versus LOD Expressions

Aspect	Table Calculation	LOD Expression
When Computed	After aggregation, on the values already in the view	As part of the query to the data source
Depends On	The structure of the current view (addressing/partitioning)	The dimensions specified in the expression itself
Typical Examples	Running total, percent of total, rank, moving average	Customer lifetime value, first order date, average order size
Reusability Across Views	Often needs reconfiguration if the view changes	Generally stable across different views

As a practical rule of thumb, use a table calculation when the calculation depends on the position or order of rows in the current view, such as "this month compared with the previous month". Use an LOD expression when the calculation depends on a fixed grouping of the underlying data, whatever the layout of the view, such as "total spend per customer".

The two techniques are also frequently combined. A common pattern is to first compute a stable, per-customer figure using a FIXED LOD expression, and then use a table calculation on top of that figure to express it as a running total or a percentage of the overall total within the current view. Recognizing that table calculations and LOD expressions solve different problems, one concerned with the order and grouping of what is already displayed, and the other concerned with the underlying grouping of the data itself, helps a learner avoid the common mistake of trying to force a single calculation type to do both jobs at once.

6.2.10 Best Practices for Data Modelling and Calculations

- Prefer relationships over joins when combining tables of different grains, and reserve joins for tables that are known to share the same level of detail.
- Check cardinality and referential integrity settings explicitly rather than accepting automatic detection without review, particularly for unfamiliar datasets.
- Give every calculated field a clear, descriptive name that indicates both what it measures and, where relevant, how it is aggregated.

SGOU

- Decide deliberately whether a calculation should behave as a row-level (calculated dimension) or aggregate-level (calculated measure) field before writing the formula.
- Test a new calculation on a small, familiar subset of data where the correct answer is already known, before applying it to the full dataset.
- Choose between a table calculation and an LOD expression based on whether the result should depend on the layout of the current view or remain fixed regardless of it.
- Add comments within complex calculations (using the // syntax) to explain the purpose of the formula for anyone who revisits it later.
- Keep the data model as simple as the analysis genuinely requires — adding tables or calculated fields that are never used makes a workbook harder to maintain without adding analytical value.

Following these practices consistently is particularly important in a shared or academic setting, where a workbook is often reviewed, reused, or extended by someone other than the person who originally built it. A clearly modelled set of tables and clearly named, well-documented calculations make that handover far more reliable than a workbook that happens to produce correct numbers but cannot easily be explained.

Data Modelling, Calculations, and the Rest of This Block

The data model and calculated fields built in this unit form the foundation for the rest of Block VI. Unit 3 relies on correctly modelled data and well-defined measures to build charts, maps, and interactive elements such as filters and drill-downs, while the end-to-end case study in Unit 4 requires a data model and a set of calculations that have already been verified for accuracy before any dashboard is assembled.

SGOU

Recap

- A data model in Tableau defines how connected tables relate to one another and at what level of detail each is recorded.
- The data source page distinguishes a logical layer, connected through relationships, from a physical layer, connected through joins or unions within a logical table.
- Relationships preserve each table's own grain and avoid row duplication, while joins physically merge rows and can cause a fan-out if cardinality is not one-to-one.
- Data blending combines two entirely separate data sources by aggregating a secondary source up to shared linking fields before combining it with a primary source.

- Cardinality describes how many matching rows exist on each side of a relationship, while referential integrity describes whether every row is guaranteed to have a match.
- A calculated field is a new field created by writing a formula using existing fields, constants, and Tableau's built-in functions.
- A calculated dimension is evaluated row by row and typically produces a categorical result, while a calculated measure combines existing measures using aggregate functions and is evaluated on already-summarized totals.
- Level of Detail (LOD) expressions — FIXED, INCLUDE, and EXCLUDE — allow a calculation to be computed at a level of detail different from the dimensions currently placed in the view.
- Table calculations operate on values already aggregated into the current view and depend on its addressing and partitioning, unlike LOD expressions, which are evaluated as part of the underlying query.
- Good practice includes preferring relationships for tables of different grains, checking cardinality explicitly, naming calculated fields clearly, and testing calculations on familiar data before wider use.

Objective Type Questions

1. What is a data model in the context of Tableau?
2. Which layer of the Tableau data model uses relationships to connect tables?
3. Which layer of the Tableau data model uses joins or unions to combine tables?
4. What problem can occur when two tables are joined on a field that is not unique in one of them?
5. Which technique is used to combine two entirely separate, unrelated data sources?
6. What does cardinality describe in a relationship between two tables?
7. Which referential integrity setting should be used when every row is known to have a match in the other table?
8. What is a calculated field in Tableau?
9. What is the key difference between a calculated dimension and a calculated measure?
10. Which LOD keyword computes a result using only the dimensions explicitly listed in the expression?

SGOU

11. Which LOD keyword removes a dimension present in the view from the calculation?
12. Which LOD keyword adds an extra dimension beyond those already in the view?
13. At what stage is a table calculation computed, relative to aggregation?
14. At what stage is an LOD expression computed, relative to the underlying query?
15. Why should SUM([Profit])/SUM([Sales]) generally be preferred over [Profit]/[Sales] for a ratio measure?

Answers

1. The structure defining how connected tables relate and at what level of detail
2. The logical layer
3. The physical layer
4. Row duplication, known as a fan-out
5. Data blending
6. How many matching rows exist on each side of the relationship
7. "All records match"
8. A new field defined using a formula over existing fields and functions
9. A calculated dimension is evaluated row by row; a calculated measure is evaluated on aggregated totals
10. FIXED
11. EXCLUDE
12. INCLUDE
13. After aggregation, on values already in the view
14. As part of the query sent to the data source
15. It computes the ratio of the correct totals rather than averaging individual row-level ratios

SGOU

Assignments

1. Explain the difference between the logical layer and the physical layer of the Tableau data model, and describe a scenario in which each would be used.
2. Compare relationships, joins, and data blending as methods of combining data, and identify which is most appropriate when combining an Orders table with a Customers table stored in the same database.
3. Discuss why understanding cardinality is important when building a data model, using an example of how a fan-out could distort a sales total.
4. Distinguish between a calculated dimension and a calculated measure, and construct one example of each using fields such as Sales, Cost, and Age.
5. Explain, with examples, the difference between FIXED, INCLUDE, and EXCLUDE Level of Detail expressions.
6. Discuss the difference between a table calculation and an LOD expression, and describe a scenario in which each would be the more appropriate choice.

Suggested Reading

1. Meier, M. (2023). Mastering Tableau 2023 (4th ed.). Packt Publishing.
2. Loth, A. (2019). Visual Analytics with Tableau. Wiley.
3. Murray, D. G. (2016). Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software (2nd ed.). Wiley.

Reference

1. Tableau Software, LLC. Tableau Desktop Online Help — Relate Your Data.
2. Tableau Software, LLC. Tableau Desktop Online Help — Create Level of Detail (LOD) Expressions.
3. Tableau Software, LLC. Tableau Desktop Online Help — Transform Values with Calculations.

SGOU



Creating and Publishing Visualizations

Learning Outcomes

After completion of this unit, learner will be able to:

- create common chart types in Tableau, including bar, line, area, scatter, and dual-axis charts
- build symbol maps and filled maps to visualize geographic data
- design simple Key Performance Indicator (KPI) style visualizations to highlight important figures
- apply filters and parameters to make a visualization interactive, and relate these to the 'slicer' concept used in other BI tools
- use hierarchies to enable drill-down analysis, and use Tableau actions to enable drill-through style interactivity
- describe how custom visuals can be incorporated into a Tableau workbook using extensions and custom shapes
- publish a completed workbook as a report to Tableau Server or Tableau Cloud, and configure basic sharing settings

SGOU

Prerequisite

This unit assumes that the learner is already comfortable with:

- Connecting to, loading, and cleaning data in Tableau, as covered in Unit 1
- Building a simple data model and writing basic calculated fields, as covered in Unit 2
- The distinction between dimensions and measures, and between discrete and continuous fields

With a properly modelled data source and a set of validated calculated fields already in place, this unit turns to the task these earlier units were building toward: constructing the charts, maps, and interactive elements that make up a finished visualization, and then publishing that work so that others can view and interact with it.

Keywords

Chart, Graph, Symbol Map, Filled Map, Key Performance Indicator (KPI), Filter, Quick Filter, Slicer, Parameter, Hierarchy, Drill Down, Drill Through, Action, Custom Visual, Extension, Tableau Server, Tableau Cloud, Packaged Workbook.

Discussion

6.3.1 From Prepared Data to Finished Visualization

Once a data source has been connected and cleaned (Unit 1) and organized into a sound data model with the necessary calculated fields (Unit 2), the next stage of work is to translate that data into visual form. In Tableau, this translation happens on a worksheet, where fields from the Data pane are placed onto shelves such as Columns, Rows, Filters, Pages, Marks, and the various encoding channels (colour, size, label, detail, and tooltip) available on the Marks card. The combination of fields and shelves used decides the chart type that Tableau draws. Tableau's recommendation engine, the Show Me panel, can suggest a suitable chart type based on the fields currently selected. This is a useful starting point for learners who are still getting familiar with which chart types suit which combinations of dimensions and measures.

It is useful to think of this unit's two halves — creating visualizations and publishing them — as addressing two different audiences. Creating a visualization is primarily a conversation between the analyst and the data: choosing the right chart type, map, or KPI to reveal a pattern that might otherwise remain hidden in a table of numbers. Publishing, by contrast, is primarily a conversation between the analyst and the eventual audience: deciding how much interactivity to expose, how the report should be organized and permissioned, and how it will be kept up to date once the analyst has moved on to other work. Both halves depend directly on the earlier units of this block, since neither a well-chosen chart nor a smoothly published dashboard can compensate for an incorrectly modelled data source or a poorly written calculation.

This unit is organized around three broad activities that together produce a finished, shareable piece of analysis: creating individual visualizations (charts, graphs, maps, and KPI-style views), making those visualizations interactive (through filters, parameters, drill-down, and drill-through actions, and by incorporating custom visuals where appropriate), and finally publishing the resulting workbook as a report that other people can access.

6.3.2 Creating Basic Charts and Graphs

Most of the visualizations built in Tableau are variations on a relatively small number of foundational chart types, each suited to a particular kind of comparison.

SGOU

Bar Charts

A bar chart is created by placing a dimension on one shelf (for example, Columns) and a measure on the other (for example, Rows), which produces one bar per category. Bar charts are well suited to comparing a measure across a moderate number of discrete categories, such as sales by product category, and remain one of the most frequently used chart types precisely because they are easy for a viewer to read accurately.

Line and Area Charts

A line chart is typically created by placing a continuous date field on Columns and a measure on Rows, which is well suited to showing how a measure changes over time. An area chart follows the same construction but fills the space beneath the line, which can help emphasize cumulative volume, particularly when several categories are stacked on top of one another using colour.

Scatter Plots

A scatter plot places one continuous measure on Columns and a second continuous measure on Rows, with each mark representing an individual data point (for example, one product or one customer). Scatter plots are useful for examining the relationship between two measures — for instance, the relationship between advertising spend and resulting sales — and connect directly to the correlation concepts introduced earlier in this course.

Dual-Axis and Combination Charts

A dual-axis chart overlays two measures that use different scales — for example, Sales in currency and Profit Ratio as a percentage — on the same set of categories, using two separate but aligned axes. This is created by placing a second measure on the Rows shelf and choosing to show it as a dual axis, which allows two related but differently scaled measures to be compared within a single view without one dominating the other visually.

Table 6.3.1 Common Chart Types and Their Typical Use

Chart Type	Typical Shelf Arrangement	Best Suited For
Bar Chart	Dimension on one shelf, Measure on the other	Comparing a measure across categories
Line Chart	Continuous date on Columns, Measure on Rows	Showing change in a measure over time
Area Chart	Same as line, with area fill and optional stacking	Emphasizing cumulative volume over time

SGOU

Scatter Plot	One measure on Columns, another on Rows	Examining the relationship between two measures
Dual-Axis Chart	Two measures on Rows, shown as a dual axis	Comparing two differently scaled measures together

Worked Example: Building a Sales-by-Category Bar Chart

Consider a prepared Orders data source that includes a Category dimension and a Sales measure. To build a simple comparison chart, Category is first dragged from the Data pane onto the Columns shelf, which creates one header for every distinct category, and SUM(Sales) is then dragged onto the Rows shelf, which produces one vertical bar per category with a height proportional to total sales. At this stage the chart already communicates its core message, but two small refinements usually improve it. First, sort the bars in descending order of sales (using the sort icon on the toolbar), so that the highest- and lowest-performing categories are easy to see. Second, drag the Category field a second time onto the Colour channel of the Marks card, so that individual categories can be told apart. This becomes especially useful when the chart is later placed beside others in a dashboard. This short sequence — placing a dimension and a measure, then refining with sorting and colour — is representative of how the large majority of Tableau charts are actually built in practice, regardless of how visually elaborate the finished result eventually looks.

Formatting, Labels, and Tooltips

Once the basic structure of a chart is in place, Tableau provides several ways to refine how it communicates. Axis titles, number formats (such as currency symbols, thousands separators, and percentage signs), and mark labels can all be adjusted through the Format menu, so that a viewer does not need to guess the unit or scale of what is displayed. The tooltip — the small pop-up that appears when a viewer hovers over a mark — can be customized to show additional context, such as an exact figure, a percentage of total, or a short explanatory sentence, without cluttering the main view itself. Thoughtful formatting of axes, labels, and tooltips is often what most clearly distinguishes a polished, presentation-ready chart from a technically correct but visually rough first draft.

6.3.3 Creating Maps

When a data source includes geographic fields — such as Country, State/Province, City, or Postal Code — Tableau can automatically assign these fields a geographic role, allowing them to be plotted on a map using built-in latitude and longitude values, without the learner needing to supply coordinates manually.

SGOU

Symbol Maps

A symbol map places a mark at the location corresponding to each geographic value, with the size or colour of the mark typically encoding a measure — for example, a circle at each city whose size represents total sales in that city. Symbol maps are well suited to point-based geographic data, such as store locations or delivery addresses.

Filled Maps

A filled map shades an entire geographic area — such as a state, province, or country — according to the value of a measure, using a colour gradient (often called a choropleth map in general cartographic terminology). Filled maps are well suited to comparing a measure across regions of unequal size, such as sales performance by state, where the shaded area draws the viewer's attention to regional patterns.

Because geographic roles depend on Tableau correctly recognizing a field as a place name, learners should verify, immediately after building a map, that no locations have been left unmatched (Tableau flags unmatched or ambiguous locations explicitly), correcting these where necessary by manually editing the location assignment for the affected values.

Maps can also combine both encodings at once: a filled map shading each state by average order value, with a layer of symbol marks sized by total order count added on top, gives a viewer two related geographic measures in a single view. Tableau supports this kind of layered map through its map layers panel, which allows additional geographic or non-geographic reference layers — such as roads, or a custom data layer — to be shown or hidden independently of the primary shaded or symbol layer, without requiring separate worksheets for each layer.

6.3.4 Key Performance Indicators (KPIs)

A Key Performance Indicator, or KPI, is a single, carefully chosen measure — such as Total Sales, Profit Ratio, or Customer Count — that is displayed prominently, often with an indicator of whether the current value is above or below a target or a prior period. In Tableau, a KPI-style view is usually built, not selected from a dedicated 'KPI chart type'. A large single number is shown as a text or Big Number style worksheet. Colour, shape or an icon (for example, an upward or downward triangle) shows performance against a target. This is often driven by a calculated field that compares the current value with the target.

A simple but effective KPI implementation combines a calculated field such as IF SUM([Sales]) >= [Sales Target] THEN "On Target" ELSE "Below Target" END with a colour encoding on the Marks card, so that the KPI's colour changes automatically as underlying data changes, without the analyst needing to manually update it. When several KPIs are needed together, they are commonly arranged side by side in a dashboard, as discussed further in the context of dashboard design in Unit 4.

A closely related technique is the use of reference lines and bands, which draw a horizontal or vertical line (or a shaded band) across a chart at a value defined by a constant, an average, or a parameter — for example, a reference line showing the

SGOU

sales target overlaid directly on a bar chart of actual monthly sales. Reference lines complement standalone KPI figures by showing, within the context of a full trend or comparison, exactly how far current performance sits from the target, rather than only stating whether the target has been met.

6.3.5 Filters, Quick Filters, and Parameters

Interactivity is one of the central advantages of Tableau over a static chart, and filters and parameters are the primary mechanisms through which a viewer can change what a visualization shows.

Filters

A filter restricts the rows used in a view to those matching a chosen condition — for example, showing only orders from a particular Region, or only orders placed within a chosen date range. A filter can be applied by an analyst while building the view (a fixed filter) or exposed to the end viewer as an interactive control, commonly referred to in Tableau as a Quick Filter.

Quick Filters and the 'Slicer' Concept

Learners who are also familiar with other business intelligence tools, such as Microsoft Power BI, may encounter the term 'slicer' used for an on-screen control that lets a viewer choose which categories to include in a view. In Tableau, this same interactive experience is provided by a Quick Filter — a filter card, placed beside a worksheet or dashboard, that can take the form of a list of checkboxes, a single dropdown, a search box, or a slider for a numeric or date range. Functionally, a Tableau Quick Filter and a Power BI slicer serve the same purpose: they let a viewer narrow down what is displayed, without needing to edit the underlying view.

Parameters

A parameter is a single, workbook-level value — such as a number, a date, or a selected string from a list — that the viewer can control and that can be referenced inside calculated fields, titles, or reference lines anywhere in the workbook. Unlike a filter, which simply hides rows, a parameter can change the underlying logic of a calculation; a common example is a parameter that lets a viewer choose which measure (Sales, Profit, or Quantity) is plotted, by referencing the parameter inside a calculated field that switches between the three measures using a CASE statement.

Aspect	Filter / Quick Filter	Parameter
Effect	Restricts which rows are shown	Supplies a value that calculations, titles, or logic can reference
Bound To	A specific field's values	A defined range or list of allowed values, independent of any one field
Typical Use	Letting a viewer choose a Region or date range	Letting a viewer switch which measure or scenario is displayed

SGOU

Worked Example: A Measure-Switching Parameter

Suppose a dashboard should let a viewer choose whether a chart displays Sales, Profit, or Quantity, without maintaining three separate charts. A parameter named Measure Selector is created with a list of three allowed string values: "Sales", "Profit", and "Quantity". A calculated field, Selected Measure, is then written using a CASE statement: CASE [Measure Selector] WHEN "Sales" THEN SUM([Sales]) WHEN "Profit" THEN SUM([Profit]) WHEN "Quantity" THEN SUM([Quantity]) END. This calculated field, rather than any single raw measure, is placed on the Rows shelf, and the parameter control is displayed beside the chart. When a viewer changes the parameter's value, the chart's axis, its underlying data, and — if the chart title references the parameter — even its title update immediately, all from a single worksheet. This example illustrates concretely why a parameter is a fundamentally different mechanism from a filter: a filter could never cause a chart to switch which measure it plots, since a filter only ever changes which rows are included, not which field is being visualized.

6.3.6 Drill Down and Drill Through

Drill Down Using Hierarchies

A hierarchy groups related dimensions into a nested structure — for example, Country, State, and City — allowing a viewer to expand a view from a coarser level of detail to a finer one by clicking the small expand icon (+) beside a dimension header. Tableau automatically detects some common hierarchies, such as parts of a date field (Year, Quarter, Month, Day), and additional hierarchies can be created manually by dragging one field onto another in the Data pane. Drill-down interactivity of this kind allows a single worksheet to serve both a summary view and a detailed view, without the analyst needing to build a separate chart for every level of detail.

Drill Through Using Actions

Some other BI tools provide a dedicated 'drill-through' feature that jumps from a summary report to a detailed report focused on a single selected item. Tableau achieves the same effect using Actions, configured at the dashboard level. A Filter Action, for example, can be set so that clicking a mark on a summary dashboard (such as a bar for a particular Region) navigates to, or filters, a second, more detailed sheet or dashboard showing only the orders from that Region. A URL Action can similarly open an external page relevant to the selected mark, and a Highlight Action can visually emphasize related marks across multiple views without changing what is filtered. Understanding Tableau Actions as the mechanism behind drill-through style navigation allows a learner to build the same kind of guided, click-to-explore experience described by that term in other tools.

SGOU

Matching Terminology Across BI Tools

Because this block sits alongside a companion block on Microsoft Power BI within the same course, it is worth explicitly noting the terminology mapping: a Power BI slicer corresponds to a Tableau Quick Filter, and Power BI drill-through corresponds to a Tableau Filter Action configured between dashboards. The underlying analytical goal — letting a viewer narrow down or navigate into data interactively — is the same in both tools, even though the menu names differ.

Worked Example: Drilling from Year into Month

Consider a line chart of Sales over time, built by placing the Order Date field on the Columns shelf. By default, Tableau recognizes the built-in date hierarchy, so the field initially appears as Year, with a small expand icon beside it. Clicking this icon expands the view to show Quarter within each Year, and clicking again expands further to Month within each Quarter, all without the analyst needing to rebuild the chart or add any new fields. This behaviour is drill-down in its simplest form: the same underlying view becomes progressively more detailed as the viewer chooses to expand it, which is particularly useful in a dashboard intended for viewers with varying needs — some wanting only an annual overview, and others wanting to inspect monthly fluctuation within a single year.

6.3.7 Custom Visuals

While Tableau's built-in chart types cover the large majority of everyday analytical needs, some scenarios call for a more specialized or branded visual than the standard set provides. Tableau supports several ways of extending its default visual vocabulary.

- Custom Shapes: Image files (such as company logos, icons, or simple pictograms) can be imported into Tableau's shape repository and used in place of the default circle or square marks, which is useful for pictogram-style charts or for reinforcing brand identity in a dashboard.
- Background Images: A custom image — such as a floor plan, a seating chart, or a non-standard map — can be added as a background, with data points plotted on top of it using x and y coordinates rather than geographic latitude and longitude.
- Custom Polygons: For geographic regions that do not correspond to a built-in map boundary (such as a custom sales territory), a table of polygon coordinates can be used to draw filled, irregularly shaped regions.
- Tableau Extensions: Extensions are small web-based applications that run inside a dashboard, built using Tableau's Extensions API, and can add functionality beyond what native Tableau provides — such as writing data back to an external system or embedding a specialized visual component built by a third party.

SGOU

Custom visuals should be used selectively: they typically require more setup effort and can be harder for a new viewer to interpret than a standard chart type, so they are generally most appropriate when a standard chart genuinely cannot communicate the intended message as effectively.

A representative use of extensions in an academic or organizational setting is a what-if input panel, where a small extension provides text boxes or sliders that a viewer can use to enter hypothetical values (for example, a hypothetical price change), with the resulting effect on revenue calculated and displayed elsewhere in the same dashboard using Tableau's native calculation engine. Because extensions run as embedded web content rather than as part of Tableau's own codebase, they should generally be sourced only from Tableau's official Extensions Gallery or from a trusted internal developer, since an extension can, in principle, interact with data passed to it from the dashboard.

6.3.8 Publishing a Report

Once a set of worksheets and, typically, a dashboard combining them have been built, the final stage of this unit is publishing that work so that it can be accessed outside the original Tableau Desktop session. Tableau supports several ways of doing this, depending on the audience and the infrastructure available.

Publishing to Tableau Server or Tableau Cloud

1. From Tableau Desktop, select the option to publish the workbook, and sign in to the target Tableau Server or Tableau Cloud site.
2. Choose a project (a folder-like container) in which the published workbook should appear, to keep content organized for other users.
3. Decide whether each data source should be embedded within the workbook or published separately, which affects whether the data source can be reused by other workbooks.
4. Configure the refresh schedule for any extract used by the workbook, so that published data stays reasonably current without the original author needing to republish manually.
5. Set permissions for the relevant users or groups, controlling who may view, interact with, or edit the published content.
6. Publish the workbook, and confirm that the resulting dashboard renders correctly and performs acceptably when opened through a web browser rather than Tableau Desktop.

Packaged Workbooks and Tableau Public

When a dedicated server is not available, or when a workbook must be shared with someone who has no server access, the workbook can be saved as a packaged workbook (a .twbx file). A packaged workbook bundles the data extract and the visual design into a single file. This file can be opened with Tableau Desktop or the free Tableau Reader.

SGOU

Alternatively, as discussed in Unit 1, work intended for open, public sharing rather than organizational distribution can be published directly to Tableau Public.

A packaged workbook is particularly useful for coursework and portfolio purposes: a learner can submit a single .twbx file that a reviewer opens with Tableau Reader, without either party needing access to a shared server. It is worth noting, however, that a packaged workbook embeds a static extract of the data at the moment it was saved; unlike a workbook published to Tableau Server or Tableau Cloud, it will not automatically refresh, so any figures within it should be understood as correct as of that specific save rather than continuously current.

Considerations Before Publishing

- Confirm that any sensitive or confidential fields have been removed, hidden, or restricted appropriately before publishing, particularly if the destination is a public gallery such as Tableau Public.
- Check that filters, parameters, and actions behave as intended once viewed through a browser, since some interactions can render slightly differently outside the Desktop authoring environment.
- Verify that the workbook performs acceptably — an overly complex view that is workable in Desktop can feel slow once several viewers access it simultaneously through a server.
- Provide a clear title and, where the platform supports it, a short description, so that viewers unfamiliar with the underlying data can understand what the report shows.

Understanding Permission Levels

When a workbook is published to Tableau Server or Tableau Cloud, access is controlled at the level of individual users or, more commonly in an organizational setting, groups of users. In broad terms, a viewer may be allowed to view and interact with a published dashboard (using filters and parameters) but not to change or download its underlying design. An explorer-level user may also be allowed to edit filters, save personal customized copies or download the data. A creator-level user, usually the original author or a chosen collaborator, has full permission to edit the workbook, republish it and manage its permissions for others. Assigning the narrowest permission level that still meets a given audience's needs — rather than granting broad editing access by default — is a widely recommended practice for maintaining the accuracy and consistency of a published report over time.

6.3.9 Good Practices in Visualization and Publishing

- Choose the chart type based on the comparison being communicated (category comparison, trend over time, relationship between measures, or geographic pattern), rather than defaulting to whichever chart looks most visually elaborate.

SGOU

- Use colour purposefully and sparingly — reserving strong colour for the field that matters most to the message of the view — rather than colouring every possible field.
- Test filters, parameters, and drill actions from the perspective of a first-time viewer, not only from the perspective of the person who built the workbook and already knows how it behaves.
- Prefer built-in chart types and native interactivity (filters, parameters, actions, hierarchies) over custom visuals unless a standard approach genuinely cannot express the required message.
- Before publishing, review the workbook as a whole for consistency — chart titles, number formatting, and terminology should match across every worksheet and dashboard in the workbook.
- Revisit permissions and refresh schedules periodically after publishing, since the audience and the underlying data both tend to change over the life of a published report.

This Unit and the Case Study in Unit 4

The chart types, maps, KPIs, filters, drill actions, and publishing workflow introduced in this unit are combined directly in the end-to-end dashboard case study covered in Unit 4 of this block. That case study assumes that a learner can already build the individual visual elements described here and knows how to make them interactive, before turning to the separate question of composing them into a single, well-designed dashboard.

Taken together, the topics in this unit move a learner from a validated data model and a set of calculated fields, as built in Unit 2, to a finished, interactive, and shareable piece of analysis. The natural next step, taken up in Unit 4, is to compose several such visualizations into a single dashboard built around a specific real-world question, and to extend the same data connection and calculation skills to a case study that also incorporates data originating from an IoT platform.

Recap

- Visualizations are built on a Tableau worksheet by placing fields from a prepared data source onto shelves such as Columns, Rows, Filters, and the Marks card.
- Bar, line, area, scatter, and dual-axis charts are foundational chart types, each suited to a different kind of comparison — categorical, trend-based, relational, or multi-scale.
- Symbol maps plot point-based geographic data using marks, while filled maps shade entire geographic areas according to a measure.

SGOU

- A KPI-style view highlights a single important measure, often combined with a calculated field and colour encoding to indicate performance against a target.
- Filters restrict which rows appear in a view, and Quick Filters expose this control interactively to a viewer — serving the same purpose as a 'slicer' in other BI tools.
- Parameters supply a single controllable value that can change the logic of a calculation, title, or reference line anywhere in the workbook.
- Hierarchies enable drill-down analysis by allowing a view to expand from a coarse level of detail to a finer one, while Actions enable drill-through style navigation between views.
- Custom visuals — including custom shapes, background images, custom polygons, and extensions — extend Tableau's default visual vocabulary but should be used selectively.
- A finished workbook is published to Tableau Server or Tableau Cloud, or shared as a packaged workbook or via Tableau Public, with attention to project organization, refresh schedules, and permissions.
- Good practice includes choosing chart types deliberately, using colour purposefully, testing interactivity from a viewer's perspective, and reviewing a workbook for consistency before publishing.

Objective Type Questions

1. Which panel in Tableau suggests a chart type based on the fields currently selected?
2. Which chart type is best suited to comparing a measure across several discrete categories?
3. Which chart type overlays two differently scaled measures using two aligned axes?
4. What type of map shades entire geographic areas using a colour gradient?
5. What is a Key Performance Indicator (KPI) in the context of a Tableau view?
6. What does a filter do to the rows shown in a view?
7. What term does Tableau use for an interactive filter control exposed to a viewer, corresponding to a 'slicer' in other BI tools?
8. What is a parameter able to affect that a simple filter cannot?

SGOU

9. Which Tableau feature allows a viewer to expand a view from a coarse to a finer level of detail?
10. Which Tableau feature is used to implement drill-through style navigation between dashboards?
11. Name one way in which a custom shape can be used in a Tableau visualization.
12. What file extension is used for a packaged Tableau workbook?
13. Which two platforms can a finished workbook be published to for organizational sharing?
14. What should be reviewed before publishing a workbook to a public gallery such as Tableau Public?
15. Why should custom visuals generally be used selectively rather than by default?

Answers

1. The Show Me panel
2. Bar chart
3. Dual-axis chart
4. Filled map
5. A single, prominently displayed measure indicating performance, often against a target
6. It restricts the view to rows matching a chosen condition
7. Quick Filter
8. The underlying logic of a calculation
9. Hierarchies (drill down)
10. Actions (specifically, a Filter Action)
11. Replacing the default mark with a logo, icon, or pictogram
12. .twbx
13. Tableau Server and Tableau Cloud
14. Sensitive or confidential fields, to confirm they are removed, hidden, or restricted
15. They require more setup and can be harder for viewers to interpret than standard chart types

SGOU

Assignments

1. Explain, with examples, which of the five chart types discussed in this unit would be most appropriate for (a) comparing profit across product categories, and (b) examining the relationship between advertising spend and sales.
2. Distinguish between a symbol map and a filled map, and describe a business scenario in which each would be the more informative choice.
3. Design a simple KPI calculation and explain how colour encoding could be used to indicate whether the KPI is above or below its target.
4. Explain the difference between a Quick Filter and a parameter, and describe a scenario that requires a parameter rather than a filter.
5. Describe how hierarchies enable drill-down analysis, and explain how Tableau Actions can be used to achieve drill-through style navigation between two dashboards.
6. Discuss the steps and considerations involved in publishing a workbook to Tableau Server or Tableau Cloud, including at least two precautions that should be taken before publishing.

Suggested Reading

1. Meier, M. (2023). Mastering Tableau 2023 (4th ed.). Packt Publishing.
2. Milligan, J. N. (2020). Learning Tableau 2020 (4th ed.). Packt Publishing.
3. Murray, D. G. (2016). Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software (2nd ed.). Wiley.

Reference

1. Tableau Software, LLC. Tableau Desktop Online Help — Which Chart Is Right for You?
2. Tableau Software, LLC. Tableau Desktop Online Help — Map, Geocode, and Answer Geographic Questions.
3. Tableau Software, LLC. Tableau Desktop Online Help — Publish Workbooks and Data Sources.

SGOU



Dashboard Development and IoT Implementation

Learning Outcomes

After completion of this unit, learner will be able to:

- explain the principles of effective dashboard design, including layout, consistency, and audience focus
- develop an interactive Tableau dashboard by connecting to datasets, modelling data, creating calculations, combining worksheets, and applying dashboard actions
- describe the characteristics of IoT data and explain the role of cloud IoT platforms in collecting and staging device data for analysis
- connect Tableau to cloud hosted IoT data and develop monitoring dashboards for visualizing device data
- apply good practices in dashboard development and IoT based visualization through a suitable case study

SGOU

Prerequisite

This unit assumes that the learner has already completed Units 1 to 3 of this block and is comfortable with:

- Connecting to, loading, and cleaning data in Tableau (Unit 1)
- Building a sound data model and writing calculated fields, including LOD expressions (Unit 2)
- Creating charts, maps, and KPI-style views, and applying filters, parameters, and drill actions (Unit 3)

This final unit of the block does not introduce many new Tableau mechanics. Instead, it draws together the skills built across Units 1 to 3 into one complete case study. It then extends the same workflow to a data source of a different kind: data streamed from connected devices, commonly called the Internet of Things (IoT).

It is worth pausing, at the start of this final unit, to notice how each of the earlier units has been building toward this point. Unit 1 ensured that a data source, however

messy at the outset, can be connected to and made analysis-ready. Unit 2 ensures that when more than one such source is involved, they can be combined correctly, and that new, derived figures can be produced through calculation with confidence in what those figures actually represent. Unit 3 ensured that this analysis-ready, correctly modelled data can be turned into charts, maps, and interactive elements that communicate clearly, and then shared with an audience beyond the original analyst. This unit does not ask the learner to acquire a further, separate skill so much as to practise combining the three skills already acquired, first on a familiar business scenario, and then on a less familiar, high-frequency data source, so that the whole workflow, rather than any single stage of it, becomes the object of practice.

Keywords

Dashboard, Container, Dashboard Action, Case Study, Internet of Things (IoT), Sensor, Telemetry, IoT Platform, AWS IoT Core, Azure IoT Hub, Cloud Data Warehouse, Streaming Data.

Discussion

6.4.1 From Individual Visualizations to a Complete Dashboard

A dashboard combines several worksheets, together with text, images, and interactive controls, into a single composed view designed to answer a related set of questions at a glance. Where a single worksheet typically addresses one question, for example, "which product category sells best?". A well-designed dashboard addresses a small family of related questions together, such as overall performance, its breakdown by category and region, and its trend over time, arranged so that a viewer can move naturally between the summary and the supporting detail.

Building an effective dashboard is therefore not simply a matter of placing several finished worksheets next to one another. It requires decisions about layout, about which elements should be interactive and how they should affect one another, and about how the dashboard will be viewed, for instance, on a desktop monitor as opposed to a phone screen. These are the concerns addressed in the first part of this unit, before the discussion turns to the end-to-end case study and the IoT extension.

6.4.2 Principles of Effective Dashboard Design

Define a Clear Purpose and Audience

Before opening Tableau, it is worth writing down, in a single sentence, the question the dashboard is meant to answer and for whom.A dashboard intended for a store manager

SGOU

reviewing daily performance has different priorities, such as simplicity, immediacy, and a small number of figures, compared with a dashboard intended for an analyst exploring quarterly trends in depth. Skipping this step is one of the most common causes of a cluttered dashboard that tries, unsuccessfully, to serve every possible audience at once.

Establish a Visual Hierarchy

The most important figure or chart on a dashboard should be the most visually prominent, typically the largest, or positioned at the top-left, since this is where a viewer's attention lands first in most reading conventions. Supporting detail, such as a breakdown chart or a smaller reference table, should be positioned so that it is clearly secondary, guiding the viewer from the headline figure toward increasingly specific detail rather than presenting every element with equal visual weight.

Maintain Consistency

Colour, font, and number formatting should be applied consistently across every worksheet included in a dashboard. If Sales is shown in blue on one chart and green on another, a viewer may reasonably assume the colours mean something different, even when they do not. Establishing a small, consistent colour palette and formatting convention before building individual worksheets, and applying it throughout, materially improves how quickly a viewer can interpret the finished dashboard.

Design for the Intended Device

Tableau allows a dashboard to be designed with a fixed size, a range of sizes, or automatically, and additionally allows separate layouts to be defined for desktop, tablet, and phone devices. A dashboard intended primarily for desktop viewing can make use of a wider layout with several charts side by side, while a phone layout generally needs to be simplified to a small number of stacked elements, since the available screen space is far smaller. Considering the likely viewing device early, rather than only when a dashboard is complete, avoids the need for extensive rework later.

Managing Whitespace and Density

A frequent mistake when building a first dashboard is to fill every available area of the canvas with a chart, on the assumption that more visible information is always more useful. In practice, deliberate empty space (often called whitespace) around and between charts helps a viewer's eye separate one element from another, and a dashboard with four clearly organized charts and generous spacing is usually easier to interpret quickly than one with eight charts packed edge to edge. A useful guiding question when a dashboard begins to feel crowded is whether every chart on it is necessary to answer the single question defined at the start of the design process, or whether some charts exist simply because the data was available.

SGOU

Table 6.4.1 Common Dashboard Action Types

Action Type	Trigger	Typical Effect
Filter Action	Selecting a mark on one worksheet	Filters one or more other worksheets to match the selection
Highlight Action	Selecting or hovering over a mark	Visually emphasizes related marks elsewhere without filtering them out
URL Action	Selecting a mark or a menu option	Opens an external web page relevant to the selected mark
Go to Sheet Action	Selecting a mark on a summary view	Navigates the viewer to a different, more detailed sheet or dashboard

Of these, the Filter Action and the Go to Sheet Action are the two most commonly used to build the kind of guided, click-to-explore experience described as 'drill-through' in Unit 3, while Highlight Actions are often used more subtly, to draw a viewer's attention across several charts on the same dashboard without narrowing what is displayed.

6.4.3 Building a Dashboard: A Step-by-Step View

1. Create a new dashboard from the dashboard menu and choose an initial size appropriate to the intended viewing device.
2. Drag existing worksheets from the sheets list onto the dashboard canvas, arranging them using horizontal and vertical containers to control how they resize relative to one another.
3. Add supporting objects such as text (for a title or short explanatory note), images (such as a logo), and blank spacers to improve visual organization.
4. Add Quick Filters or parameter controls to the dashboard, and decide which worksheets each control should apply to using the 'Apply to Worksheets' setting.
5. Configure dashboard actions such as a Filter Action, Highlight Action, or URL Action, to define how clicking or hovering over one worksheet affects others on the same dashboard.
6. Review the dashboard at each intended device size, adjusting the layout for tablet or phone as needed.
7. Add a final title and, where useful, a short caption or set of instructions, before proceeding to publish the dashboard as described in Unit 3.

As with building the data model in Unit 2, this process works best when carried out incrementally: adding one or two worksheets, checking the layout, and then continuing,

SGOU

rather than assembling every element at once and only reviewing the result at the very end.

6.4.4 End-to-End Case Study: A Sales Performance Dashboard

To consolidate the workflow covered across this block, consider a case study in which a retail organization wants a single dashboard summarizing sales performance, built from a raw transactional dataset containing order-level records with fields such as Order Date, Region, Category, Customer ID, Sales, and Profit. The following account walks through the case study end to end, referring back explicitly to the units in which each stage was introduced.

Stage 1: Connect and Clean (Unit 1)

The raw order data, supplied as a set of CSV files, is connected using the text file connector. The Data Interpreter is used to remove a stray title row present in one of the files, the Order Date field is corrected from a text data type to a proper date type, and the individual monthly files are combined using a union, since they share an identical structure.

Stage 2: Model and Calculate (Unit 2)

The cleaned Orders table is related to a separate Customers table on Customer ID, preserving the different grain of each table. A Profit Ratio measure is defined as SUM([Profit]) / SUM([Sales]), and a {FIXED [Customer ID] : SUM([Sales])} expression is used to calculate each customer's lifetime sales for later use in a customer-value segment.

Stage 3: Visualize (Unit 3)

A bar chart of Sales by Category, a line chart of Sales over time by Month, and a filled map of Profit Ratio by Region are each built as separate worksheets. A KPI-style text view displays overall Profit Ratio against a fixed target, coloured green or red depending on whether the target is met. A Region Quick Filter is added, and a Filter Action is configured so that clicking a bar in the Sales by Category chart filters the line chart to show the trend for that category alone.

Stage 4: Assemble and Publish (Unit 3 and this Unit)

The four worksheets and the KPI view are arranged on a single dashboard using a combination of horizontal and vertical containers, with the KPI positioned prominently in the top-left corner in line with the visual hierarchy principle discussed above, and the Region filter placed clearly at the top so that it is immediately visible. After checking the layout at both desktop and tablet sizes, the dashboard is published to Tableau Server with a weekly extract refresh schedule, and view-level permission is granted to the relevant sales team group.

SGOU

Why This Case Study Matters

This case study is deliberately built using only techniques already introduced in Units 1 through 3 of this block. Its purpose is not to introduce new mechanics, but to demonstrate that a real dashboard is rarely the product of a single skill; it is the combined result of careful data preparation, a correctly designed model, thoughtfully chosen visualizations, and a deliberate publishing decision, applied together in sequence.

Learners are encouraged to reproduce a similar case study using any dataset of their own choosing, following the same four stages, as this repetition, applied to unfamiliar data is often what most solidifies the individual skills covered earlier in the block.

Choosing a Data Set for Your Own Case Study

Any moderately sized, real or realistic tabular dataset is suitable for reproducing this case study, provided it contains at least one categorical dimension suitable for comparison (equivalent to Category or Region above), at least one date field suitable for a trend view, and at least one numeric measure worth summarizing. Publicly available sample datasets such as a retail superstore dataset, an open government dataset, or a dataset drawn from an earlier block of this course, all work well for this purpose. What matters most is not the specific subject of the data, but that the learner deliberately works through all four stages in sequence, rather than skipping directly to chart-building on data that has not been properly connected, modelled, and checked first.

6.4.5 Introduction to the Internet of Things (IoT)

The Internet of Things, or IoT, refers to networks of physical devices, sensors, meters, wearable devices, industrial machinery, and similar equipment that are connected to the internet and capable of collecting and transmitting data about their environment or their own operation without requiring direct human input for every reading. A temperature sensor in a warehouse, a smart electricity meter, and a fitness tracker are all examples of IoT devices, each continuously generating small pieces of data, often referred to as telemetry, that describe a measurement at a particular point in time.

Examples of IoT Applications Across Domains

IoT deployments span a wide range of domains, each generating data with a similar underlying character but a different set of measured values. In manufacturing, sensors attached to machinery monitor vibration, temperature, and operating hours to support predictive maintenance, flagging a machine for servicing before it fails rather than after. In agriculture, soil-moisture and weather sensors distributed across a field inform irrigation decisions in near real time. In logistics, GPS trackers fitted to vehicles report location and speed, supporting fleet-monitoring dashboards similar in spirit to the sales dashboard built earlier in this unit, but updated continuously rather than daily. In healthcare, wearable devices report heart rate and activity level, and in smart buildings, environmental sensors of the kind used in the case study later in this unit monitor

SGOU

temperature, humidity, and occupancy. Recognizing this variety is useful because, despite their different subject matter, all these examples share the same underlying data pattern: a device identifier, a timestamp, and one or more measured values. This is why the same Tableau techniques introduced throughout this block can be applied across different domains with only minor adaptations.

How IoT Data Differs from Conventional Business Data

IoT data has several characteristics that distinguish it from the transactional order and customer data used in the case study above. It is typically generated at high frequency and in large volume, with a single sensor potentially producing a new reading every few seconds rather than a new record only when a business event, such as an order, occurs. It often arrives as a continuous stream rather than as a periodically updated file, which affects how it is collected and stored before analysis. It also tends to have a relatively simple structure at the point of collection, commonly consisting of a device identifier, a timestamp, and one or a few measured values. Its analytical value increases when multiple readings are aggregated or compared over time.

6.4.6 Cloud Platforms for IoT

Because IoT devices generate data continuously and in large volume, this data is rarely connected to directly from a tool such as Tableau. Instead, a dedicated IoT platform, typically provided by a cloud vendor, is used to securely receive, authenticate, and stage device data before it is made available for analysis.

a. AWS IoT Core

AWS IoT Core, part of Amazon Web Services, provides secure, two-way communication between IoT devices and the cloud, using protocols such as MQTT and HTTPS. It also has a rules engine that can route incoming device messages to other AWS services. For example, it can store readings in a database or forward them to a data warehouse such as Amazon Redshift for later analysis.

b. Azure IoT Hub

Microsoft Azure IoT Hub provides an equivalent set of capabilities within the Azure ecosystem, including device provisioning, bi-directional messaging, and device management, and integrates closely with other Azure services such as Azure SQL Database and Power BI, making it a common choice for organizations already using Microsoft's cloud platform.

Choosing Between AWS and Azure for an IoT Case Study

For coursework and case-study purposes, the choice between AWS IoT Core and Azure IoT Hub is rarely dictated by technical necessity, since both provide broadly equivalent core capabilities: secure device connectivity, message routing, and integration with a wider set of cloud analytics services. In practice, the more important question for a learner is which cloud ecosystem is already in use elsewhere in the organization or course. For example, a learner who already uses Azure SQL Database as a Tableau data source, as introduced in Unit 1, may find it easier to stay in the Azure ecosystem

SGOU

and use Azure IoT Hub. Credentials, networking and familiarity are already in place. The underlying Tableau workflow described later in this unit is identical regardless of which platform stages the data, which is precisely the point of routing IoT data through a conventional cloud data warehouse before Tableau ever connects to it.

A Note on Google Cloud

Learners researching this area independently should be aware that Google formally retired its dedicated Google Cloud IoT Core service in August 2023. Organizations building IoT solutions on Google Cloud today typically rely on a combination of third-party device-connectivity partners together with Google's general-purpose data services, such as Cloud Pub/Sub for message ingestion and BigQuery for large-scale analytical storage, rather than a single dedicated Google IoT product. This distinction matters for a course such as this one, since a learner encountering an older textbook or article may reasonably, but incorrectly, assume that Google Cloud IoT Core remains an actively offered service.

Table 6.4.2 Current Status and Roles of Major Cloud IoT Platforms in IoT Analytics Pipelines

Platform	Current Status	Role in an IoT Analytics Pipeline
AWS IoT Core	Actively offered and maintained	Device connectivity, message routing to AWS storage and analytics services
Azure IoT Hub	Actively offered and maintained	Device connectivity and management, integration with Azure data and BI services
Google Cloud IoT Core	Retired in August 2023	Superseded by a combination of partner connectivity tools and Google's general data services (e.g., Pub/Sub, BigQuery)

6.4.7 Connecting Tableau to Cloud-Hosted IoT Data

Tableau does not typically connect directly to the high-frequency stream of raw device messages arriving at an IoT platform; instead, it connects to a data store where that stream has already been collected, and usually pre-aggregated, into an analysis-ready structure. A common architecture routes incoming device data from an IoT platform such as AWS IoT Core or Azure IoT Hub into a cloud data warehouse. Examples are Amazon Redshift, Google BigQuery or Azure SQL Database, all introduced as cloud connectors in Unit 1. In the warehouse, the data is organized into tables, such as one row per device per minute, or per hour, depending on the level of detail needed for analysis. Tableau then connects to this warehouse table using the same connector categories, data loading process, and choice between a live connection and an extract discussed in Unit 1, and the same data modelling and calculation techniques discussed in Unit 2 apply without modification.

SGOU

IoT tables can grow very large very quickly because devices report readings so often. For this reason, an extract built on pre-aggregated data (for example, hourly averages instead of every individual reading) is often preferred to a live connection to the full raw stream. This helps performance. It also suits most dashboard audiences, who care more about trends and thresholds than about every individual sensor reading.

This data staging pattern, from the device to the IoT platform, then to the cloud data warehouse and finally to Tableau, also provides practical benefits for data governance. Tableau connects at the warehouse table, not at the IoT platform itself. An organization can therefore apply the same access controls, naming conventions and data-quality checks to IoT data that it applies to its other analytical tables. Device data does not need to be treated as a special case with a separate set of tools and procedures. Learners who have worked through the data connector and data model discussions in Units 1 and 2 should recognize this as a natural extension of ideas already covered, rather than as a wholly new topic.

6.4.8 Case Study: An IoT Environmental Monitoring Dashboard

As a second, shorter case study illustrating the IoT-specific extension of this block's skills, consider an organization that has deployed temperature and humidity sensors across several warehouse locations, with readings collected through an IoT platform and staged, at five-minute intervals, in a cloud data warehouse table containing Device ID, Location, Timestamp, Temperature, and Humidity.

Stage 1: Connect and Prepare

Tableau connects to the cloud data warehouse table using the appropriate cloud connector, following the same process described in Unit 1. Because the raw data arrives at five-minute intervals, a calculated field is used to round each Timestamp down to the nearest hour, and an extract is built that aggregates Temperature and Humidity to hourly averages per Location, both to improve performance and to focus the resulting dashboard on hourly trends rather than every individual five-minute reading.

Stage 2: Model and Calculate

A calculated field flags any hourly reading where average Temperature exceeds a defined safe threshold, using a formula such as IF [Avg Temperature] > [Safe Threshold] THEN "Alert" ELSE "Normal" END, and a {FIXED [Location] : AVG([Temperature])} expression calculates each location's overall average temperature for comparison against individual hourly readings.

Stage 3: Visualize and Assemble

A line chart of hourly Temperature by Location, coloured by the Alert/Normal flag, allows a viewer to immediately spot periods where a location exceeded the safe threshold. A KPI view shows the number of locations currently in an Alert state, and a symbol map plots each warehouse location, sized by how far its most recent reading sits above or below the safe threshold. These three views are assembled into a single

SGOU

monitoring dashboard using the same container-based layout approach described earlier in this unit, published to Tableau Server with a refresh schedule set to match how frequently the underlying warehouse table itself is updated.

This case study shows that once IoT data has been staged into an analysis-ready cloud table, building a monitoring dashboard needs the same Tableau skills used throughout this block: connecting to data, modelling and calculating, visualizing and publishing. The main practical difference is a greater emphasis on aggregation and threshold-based calculated fields, which suit the continuous, high-frequency nature of sensor data.

Stage 4: Extending the Case Study

A more advanced version of this same case study could add a second Filter Action linking the symbol map to the line chart, so that clicking a specific warehouse location on the map filters the trend line to that location alone, directly reusing the drill-through pattern introduced earlier in this unit and in Unit 3. A further extension could add a parameter allowing a facilities manager to adjust the safe temperature threshold interactively, immediately recolouring the Alert/Normal flag throughout the dashboard, in the same way that the measure-switching parameter example was used in Unit 3. These extensions are left as an exercise, in the assignments below, precisely because they require no new Tableau concepts beyond what has already been introduced across this block, only the judgement to recognize where a previously learned technique applies to a new kind of data.

6.4.9 Good Practices for Dashboard Development and IoT Implementation

- Define the dashboard's purpose and intended audience in a single sentence before building any worksheet, and revisit that sentence if the design starts to feel cluttered.
- Establish a consistent colour palette and formatting convention across every worksheet before combining them into a dashboard.
- Design and test the dashboard at every intended device size, rather than assuming a desktop layout will translate acceptably to a phone or tablet.
- Treat an end-to-end case study as a sequence of stages such as connect and clean, model and calculate, visualize, assemble and publish and verify each stage before moving to the next.
- For IoT and other high-frequency data, aggregate to the coarsest level of detail that still supports the analytical question, rather than defaulting to the finest possible grain.
- Where an IoT platform or cloud service is referenced in coursework or documentation, verify its current status before relying on it, since offerings in this area as illustrated by the retirement of Google Cloud IoT Core — can change relatively quickly.

SGOU

- Document the overall data flow, from device to cloud platform to data warehouse to Tableau, so that anyone maintaining the dashboard later understands where each figure ultimately originates.

Reviewing a Dashboard Before It Is Considered Finished

Before treating any dashboard as finished, whether for sales performance or IoT monitoring, it is worth stepping through it as a first-time viewer would. Ask these questions. Do the title and initial layout make the dashboard's purpose obvious within a few seconds? Does every filter, parameter and action behave as expected? Are colours, number formats and terminology consistent across every worksheet? Does the layout remain usable at the smallest device size it is expected to be viewed on? This short review routine catches a large proportion of the issues that otherwise only surface once real viewers begin using the dashboard.

Reviewing a Dashboard Before It Is Considered Finished

Before treating any dashboard as finished, whether for sales performance or IoT monitoring, it is worth stepping through it as a first-time viewer would. Ask these questions. Do the title and initial layout make the dashboard's purpose obvious within a few seconds? Does every filter, parameter and action behave as expected? Are colours, number formats and terminology consistent across every worksheet? Does the layout remain usable at the smallest device size it is expected to be viewed on? This short review routine catches a large proportion of the issues that otherwise only surface once real viewers begin using the dashboard.

This unit concludes Block VI of the course. Across the four units of this block, the learner has progressed from connecting to and cleaning a single data source, through building a sound data model and writing calculations, to creating and publishing interactive visualizations, and finally to assembling those visualizations into complete, purpose-built dashboards including one that draws on the distinctive characteristics of IoT data. These skills, taken together, represent a complete, practical introduction to self-service data analytics and visualization using Tableau

Recap

- A dashboard combines several worksheets, together with supporting text and controls, into a single composed view addressing a related set of questions.
- Effective dashboard design depends on a clear purpose and audience, a deliberate visual hierarchy, consistent formatting, and a layout suited to the intended viewing device.
- A dashboard is built by arranging worksheets within containers, adding supporting objects and filter controls, and configuring dashboard actions that connect the worksheets interactively.

SGOU

- An end-to-end case study proceeds through four stages: connecting and cleaning the data, building a model and calculations, creating visualizations, and assembling and publishing the finished dashboard.
- The Internet of Things (IoT) refers to networks of connected devices that continuously generate high-frequency, high-volume telemetry data, distinct in character from conventional transactional business data.
- Cloud IoT platforms such as AWS IoT Core and Azure IoT Hub securely receive and stage device data, typically routing it into a cloud data warehouse before it is analyzed.
- Google Cloud IoT Core was retired in August 2023; current Google Cloud IoT approaches rely on partner connectivity tools combined with general-purpose services such as Pub/Sub and BigQuery.
- Tableau connects to staged, pre-aggregated IoT data in a cloud data warehouse using the same connector, modelling, and calculation techniques introduced earlier in this block.
- An IoT monitoring dashboard case study typically emphasizes aggregation to a manageable time interval and threshold-based calculated fields to flag readings outside a safe range.
- Good practice throughout dashboard development includes defining purpose and audience early, maintaining consistency, testing across device sizes, and verifying the current status of any cloud platform referenced.

Objective Type Questions

1. What is a Tableau dashboard, as distinct from a single worksheet?
2. Which dashboard design principle concerns making the most important chart the most visually prominent?
3. Which Tableau feature is used to arrange worksheets so that they resize predictably relative to one another on a dashboard?
4. Which dashboard action type is commonly used to filter one worksheet based on a selection made in another?
5. In the end-to-end case study, at which stage is a union used to combine several monthly files?
6. In the end-to-end case study, which technique calculates each customer's lifetime sales regardless of the view's other dimensions?

SGOU

7. What does the abbreviation IoT stand for?
8. What term is commonly used for the raw data generated by an IoT sensor?
9. Name one characteristic that distinguishes IoT data from conventional transactional business data.
10. Which AWS service provides secure, bi-directional communication between IoT devices and the cloud?
11. Which Microsoft cloud service provides an equivalent role to AWS IoT Core?
12. What happened to Google Cloud IoT Core in August 2023?
13. Why is IoT data typically aggregated before Tableau connects to it, rather than connected to at its rawest, highest frequency?
14. In the IoT case study, what type of calculated field flags a reading that exceeds a safe threshold?
15. Which four stages make up the end-to-end case study workflow described in this unit?

Answers

1. A combined, interactive view assembled from several worksheets to address a related set of questions
2. Establishing a visual hierarchy
3. Containers (horizontal and vertical layout containers)
4. A Filter Action
5. Stage 1 (Connect and Clean)
6. A FIXED Level of Detail (LOD) expression
7. Internet of Things
8. Telemetry
9. High frequency / large volume / streaming arrival / simple structure per reading (any one)
10. AWS IoT Core
11. Azure IoT Hub
12. It was retired (discontinued) by Google

SGOU

13. For performance, and because most audiences need trends rather than every individual reading

14. A logical (IF/THEN) calculated field comparing the reading against a threshold

15. Connect and clean; model and calculate; visualize; assemble and publish

Assignments

1. Explain the key principles of effective dashboard design discussed in this unit, and illustrate each with reference to a dashboard you have seen (in coursework, at work, or elsewhere).

2. Describe the four stages of the end-to-end case study covered in this unit, and, using a dataset of your choice, carry out each stage to build your own sales-style dashboard.

3. Explain how IoT data differs from the transactional data used in earlier units of this block, and discuss why this difference affects how the data is aggregated before visualization.

4. Compare AWS IoT Core and Azure IoT Hub as platforms for collecting and staging IoT device data, and explain why Google Cloud IoT Core is no longer part of this comparison.

5. Describe, step by step, how staged IoT data in a cloud data warehouse would typically be connected to and visualized in Tableau, referring back to relevant concepts from Units 1 and 2.

6. Design a simple IoT monitoring dashboard for a scenario of your choice (for example, a smart home, a vehicle fleet, or an agricultural sensor network), specifying at least one KPI, one chart, and one threshold-based calculated field.

SGOU

Suggested Reading

1. Meier, M. (2023). Mastering Tableau 2023 (4th ed.). Packt Publishing.

2. Milligan, J. N. (2020). Learning Tableau 2020 (4th ed.). Packt Publishing.

3. Loth, A. (2019). Visual Analytics with Tableau. Wiley.

Reference

1. Tableau Software, LLC. Tableau Desktop Online Help — Create Dashboards and Stories.
2. Amazon Web Services, Inc. AWS IoT Core — Product Documentation.
3. Microsoft Corporation. Azure IoT Hub — Product Documentation.

SGOU

MODEL QUESTION PAPER SETS

SGOU

SREENARAYANAGURU OPEN UNIVERSITY

QP CODE: ……

Reg. No. : .....

**Name: ……..... **

BACHELOR OF COMPUTER APPLICATION, SIXTH SEMESTER

END SEMESTER EXAMINATION

Discipline Specific Elective Course

B21CA03DE - Data Analytics & Visualization

2023-2024 - Admission Onwards (CBCS - UG)

MODEL QUESTION PAPER- SET A

Time: 3 Hour | Max Marks: 70

SECTION A

Answer any* ten *questions of the following. Each question carries* one *mark.

(10 × 1 = 10 Marks)

1. What is mode?
2. What is the Expansion of IQR
3. What are the 5 Vs of Big Data?
4. Name the six phases of the Data Analytics Life Cycle
5. Which Python library is used for numerical computation?
6. What are the four components of a time series?
7. How does Laplace (add-one) smoothing adjust a probability calculation?
8. Why can overusing 3D effects on a chart be misleading?
9. What is a treemap?
10. What might prompt a return to the preparation stage after evaluation?
11. What type of table stores descriptive attributes such as product or customer details?
12. Which filter scope applies to every visual on the current report page only?
13. Which type of connection always queries the original data source directly?
14. Which LOD keyword computes a result using only the dimensions explicitly listed in the expression?
15. What type of map shades entire geographic areas using a colour gradient?

SGOU

SECTION B

Answer any* five *questions of the following. Each question carries* two *marks.

(5×2 =10 Marks)

16. Differentiate between positive skewness and negative skewness
17. What are the characteristics of a Poisson distribution?
18. What is an analytic sandbox? State its purpose
19. What is the Correlation Coefficient (Pearson's r)?
20. What is the Naive Bayes Classifier?
21. Compare Exploratory and Explanatory Visualization.
22. Explain deployment pipelines.
23. How does the data loading process take place in PowerBI?
24. Describe LOD expressions.
25. Define Key Performance Indicators (KPIs)

SECTION C

Answer any* five *questions of the following. Each question carries* four *marks.

(5×4= 20 Marks)

26. Calculate the mean and median for the following data:

    12, 15, 18, 20, 22, 25, 28

27. A fair coin is tossed 5 times. Identify the appropriate probability distribution and write the expressions for calculating the probability of obtaining exactly 3 heads.
28. Compare Power BI and Tableau based on ecosystem integration, visualization flexibility, cost and typical organization.
29. Compare the Correlation Coefficient and RMSE.
30. Explain Machine Learning Based Sentiment Analysis Using Naive Bayes.
31. Explain the Common Pitfalls Across the Cycle.
32. Explain PowerBI architecture and its components.
33. Compare Star schema and Snowflake schema.
34. What are the different versions of Tableau?
35. How to publish a report in Tableau

SGOU

SECTION D

Answer any* two *questions of the following. Each question carries* fifteen *marks.

(2×15 = 30Marks)

36. What is data normalization? Explain min-max normalization, z-score standardization, decimal scaling and robust scaling with suitable formulas and numerical examples.
37. For the data X = 2, 4, 6, 8, 10 and Y = 3, 5, 6, 9, 10, calculate $\bar{x}$, $\bar{y}$, $\Sigma(x-\bar{x})(y-\bar{y})$, $\Sigma(x-\bar{x})^2$, and $\Sigma(y-\bar{y})^2$, and use them to calculate the covariance and the correlation coefficient r.
38. Detailed note on the concept of IoT and explain its implementation.
39. Write an account on PowerBI security and deployment.

SGOU

SREENARAYANAGURU OPEN UNIVERSITY

QP CODE: ……

Reg. No. : .....

**Name: ……..... **

BACHELOR OF COMPUTER APPLICATION, SIXTH SEMESTER

END SEMESTER EXAMINATION

Discipline Specific Elective Course

B21CA03DE - Data Analytics & Visualization

2023-2024 - Admission Onwards (CBCS - UG)

MODEL QUESTION PAPER- SET B

Time: 3 Hour Max Marks: 70

SECTION A

Answer any* ten *questions of the following. Each question carries* one *mark.

(10 × 1 = 10 Marks)

1. What is a probability distribution?
2. What are mutually exclusive events?
3. What is the Expansion of ETL
4. What is tokenization ?
5. What is AWS Glue?
6. What does the letter k represent in the K-Means algorithm?
7. What is the range of possible values for Pearson's correlation coefficient, r?
8. What does the data-ink ratio principle recommend?
9. What is multitype data?
10. What is a KPI card?
11. Which operation combines two queries based on matching column values?
12. What is the formula language used to define calculated columns and measures in Power BI called?
13. In what file format does Tableau typically store a data extract?
14. At what stage is an LOD expression computed, relative to the underlying query?
15. Which two platforms can a finished workbook be published to for organizational sharing?

SGOU

SECTION B

Answer any* five *questions of the following. Each question carries* two *marks.

(5×2 =10 Marks)

16. What is the interquartile range? Write the formula
17. Mention any four strategies for handling anomalies.
18. Write any two features of Tableau.
19. What is Time Series Forecasting ?
20. What is Root Mean Square Error (RMSE) ?
21. What does 'interactivity without purpose' mean, and why is it a pitfall ?
22. Define Power Query Editor.
23. What is the key difference between Drill Down and Drill Through methods?
24. What is the difference between cardinality and referential integrity?
25. Define filters and quick filters.

SECTION C

Answer any* five *questions of the following. Each question carries* four *marks.

(5×4= 20 Marks)

26. Explain min-max normalization with its formula and a suitable example.
27. Explain the 5 Vs of Big Data with suitable examples
28. Describe the important Python libraries used in data analysis.
29. Explain the Simple Linear Regression.
30. Explain the Applications of Machine Learning in Data Analytics.
31. Explain the Different Applications of Geospatial and Multitype Visualization.
32. Define Row-Level Security (RLS) and its types.
33. What are the steps to format and design effective reports?
34. Describe the Tableau Data Model.
35. Explain Monitoring and Auditing in PowerBI.

SGOU

SECTION D

Answer any* two *questions of the following. Each question carries* fifteen *marks.

(2×15 = 30Marks)

36. Explain the Data Analytics Life Cycle in detail. Discuss all six phases and their importance in completing a data analytics project.
37. Given four points A(1,1), B(2,1), C(8,8), D(9,8), apply one full iteration of K-Means with k = 2, using A and C as the initial centroids. Show the distance calculations, the resulting cluster assignments, and the updated centroids
38. For each of the following, state whether it is best represented as point, line, or polygon geospatial data, and justify your answer:

    a. The locations of ATMs operated by a bank.

    b. The delivery route taken by a courier.

    c. The sales territory assigned to each regional manager.
39. Write a detailed account on DAX (Data Analysis Expressions).

SGOU

സർവ്വകലാശാലാഗീതം

വിദ്യയാൽ സ്വതന്ത്രരാകണം
വിശ്വപൗരരായി മാറണം
ഗ്രഹപ്രസാദമായ് വിളങ്ങണം
ഗുരുപ്രകാശമേ നയിക്കണേ

കൂരിരുട്ടിൽ നിന്നു ഞങ്ങളെ
സൂര്യവീഥിയിൽ തെളിക്കണം
സ്നേഹദീപ്തിയായ്വിളങ്ങണം
നീതിവൈജയന്തി പാറണം

ശാസ്ത്രവ്യാപ്തിയെന്നുമേകണം
ജാതിഭേദമാകെ മാറണം
ബോധരശ്മിയിൽ തിളങ്ങുവാൻ
ജ്ഞാനകേന്ദ്രമേ ജ്വലിക്കണേ

കുരീപ്പുഴ ശ്രീകുമാർ

SGOU

SREENARAYANAGURU OPEN UNIVERSITY

Regional Centres

Kollam
Sreenarayanaguru Open University
Headquarters Building
BSNL Bhavan, Vellayittambalam,
Kollam, Kerala, Pin : 691012
Ph: 9188912554
email:rckollam@sgou.ac.in

Tripunithura
Govt. College
Tirpunithura, Ernakulam
Kerala, Pin : 682301
Ph: 04842927436
email:rcedirector@sgou.ac.in

Thalassery
Govt. Brennen College
Dharmadam, Thalassery
Kannur, Pin : 670106
Ph: 04952990494
email:rctdirector@sgou.ac.in

Pattambi
Sree Neelakanta Govt. Sanskrit College
Pattambi, Palakkad
Kerala, Pin : 679303
Ph: 04662912009
email:rcpdirector@sgou.ac.in

Kozhikode
Govt. Arts and Science College
Meenchantha, Kozhikode,
Kerala, Pin : 673002
Ph: 04952920228
email:rckdirector@sgou.ac.in

SAY NO TO RAGGING

BEFORE YOU EVEN THINK OF RAGGING



Humiliation

Blacklisting

Ruined Career

Download

ANTI RAGGING

App

Don't just stand and watch. Stop Ragging! Show Character

Visit UGC Website i.e. www.ugc.ac.in & www.antiraging.in to UGC Anti Ragging regulations.

Ministry of Education
Government of India

Are You Being Ragged ?

विश्वविद्यालय अनुदान आयोग
University Grants Commission

SSGOU

DON'T LET IT BE TOO LATE

SAY NO TO DRUGS

LOVE YOURSELF AND ALWAYS BE HEALTHY



SREENARAYANAGURU OPEN UNIVERSITY

The State University for Education, Training and Research in Blended Format, Kerala

Data Analytics and Visualization

COURSE CODE: B21CA04DE

SGOU

Sreenarayanaguru Open University
Kollam, Kerala Pin- 691601, email: info@sgou.ac.in, www.sgou.ac.in Ph: +91 474 2966841

ISBN 978-81-903304-3-5