Data Visualisation: Theory and Practice

Yujie Chu, Pia Fullaondo, Qinqing Li, Jacko Zhou

November 13, 2023

Contents

1 Introduction

1.1 Motivation and Background

Motivations for having Data Visualisations - Case Example 1

Florence Nightingale was not only a social reformer and the founder of modern nursing but also a pioneering statistician. It was her application of data visualisation during the Crimean War that transformed the field of healthcare and pushed for social reform.

During the Crimean War, Nightingale recognised that unsanitary hospital conditions were claiming more lives than the battlefield itself. With the help of William Farr, Nightingale created the coxcomb aimed to illustrate the toll of preventable mortality on soldiers, as shown in Figure **??**.

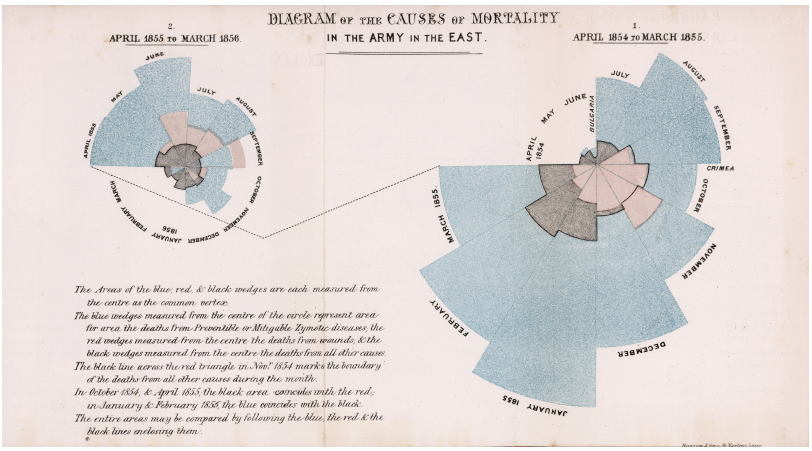


Figure 1: "Diagram of the causes of mortality in the army in the East", in 1858 by Florence Nightingale

The coxcomb, resembling an unconventional pie chart, partitioned mortality by causes. Blue indicates preventable deaths, red indicates deaths by wounds, and black indicates other causes. The blue areas outweighed the red and black sections combined, highlighting the disproportionate impact of unsanitary hospital conditions on the mortality rate.

Nightingale leveraged the compelling visualisations in her advocacy efforts, presenting them to MPs and government officials who otherwise are unlikely to read or understand statistical reports. Nightingale successfully persuaded Queen Victoria, head of the British Army at the time, to allocate funding for the improvement of better conditions in military hospitals.

Motivations for having Data Visualisations - Case Example 2

Sometimes, one glance is enough to convey the most powerful idea. Edward Hawkins, a British climate scientist and Professor of climate science at the University of Reading, is renowned for his exceptional data visualizations of climate change.

In 2018, Edward Hawkins was invited to deliver a lecture on climate change in Wales to an audience with diverse backgrounds. It was important to effectively convey the growing urgency surrounding global warming. To achieve this, he created a chart that used just colours, without any words, titles, or legends, as shown in Figure **??**. This seemingly simple yet remarkably powerful chart visually illustrated the Earth's warming trend since 1850.

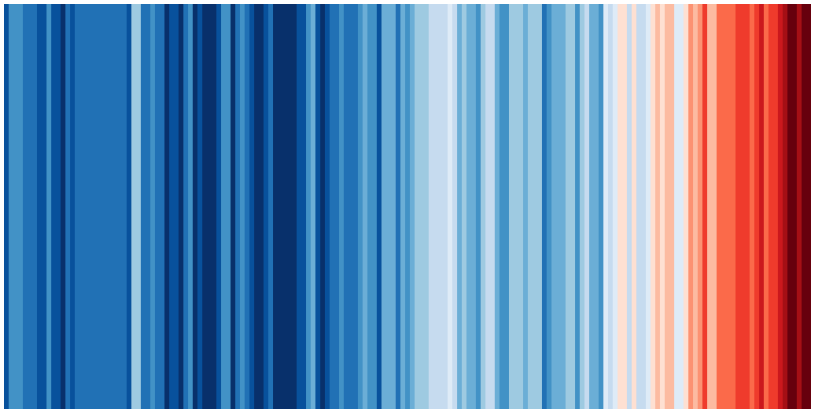


Figure 2: "Latest global stripes (1850-2020)", by Edward Hawkins

Known as the "warming stripes," this chart cleverly employs blues to indicate cooler-than-average years and reds to signify hotter-than-average years. Its influence reached far and wide, gracing the front pages of major media outlets and featured in news broadcasts worldwide. It became a symbol in climate change demonstrations. Arguably, it stands as one of the most iconic graphics in modern times.

Misuses of Data Visualisation - Case Example 1

Inappropriate data visualization conceals trends rather than revealing them. Figure **??** illustrates an instance of this issue. On the left-hand side, an inappropriate scale was used — the y-scale ranging from 0 to 30 million dollars, obscuring the fluctuations in payroll spending. Conversely, on the right-hand side, observe that there's a significant increase of over 500,000 dollars in just two months. This revelation is substantial; considering inflation, 500,000 dollars in 1937 is worth well over 10 million dollars today.

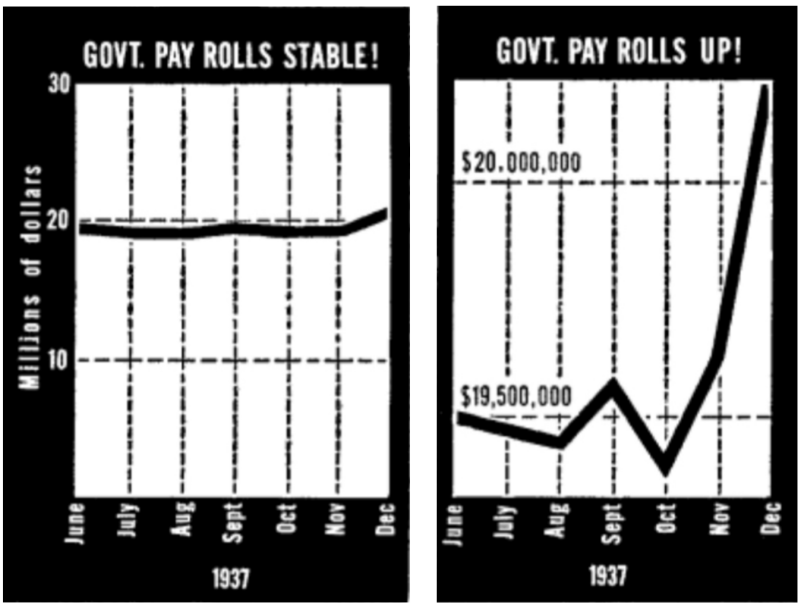


Figure 3: Inappropriate use of data visualisation

Misuses of Data Visualisation - Case Example 2

Data visualization can be misused, leading to disastrous consequences. One striking example of such misuse is found in the Kallikak Family tree, which was one of the most prominent eugenic narratives of the 20th century.

The visualization (as shown in Figure **??**) was created by the psychologist Henry Goddard and presented in his 1912 book, "The Kallikak Family: A Study in the Heredity of Feeble-Mindedness." Goddard's narrative centered around Martin Kallikak, a soldier who, in addition to his marriage to a respected citizen, had a one-night stand with a "feeble-minded" maid. Goddard believed that intellectual disabilities were inherited traits. In Goddard's account, the legitimate family was successful, while the children of the "feeble-minded" maid were labeled as "the lowest types of human beings." However, research has since revealed that the entire story was fictitious, as there was no record of the maid's existence.

Regrettably, the Kallikak family tree became a central element in the eugenics movement for decades afterward. It was featured in the 1935 Nazi propaganda film "Das Erbe" (The Inheritance), which was used to promote public acceptance of Nazi eugenics laws. This propaganda laid the groundwork for the forced sterilization of approximately 400,000 people under Nazi eugenics policies.

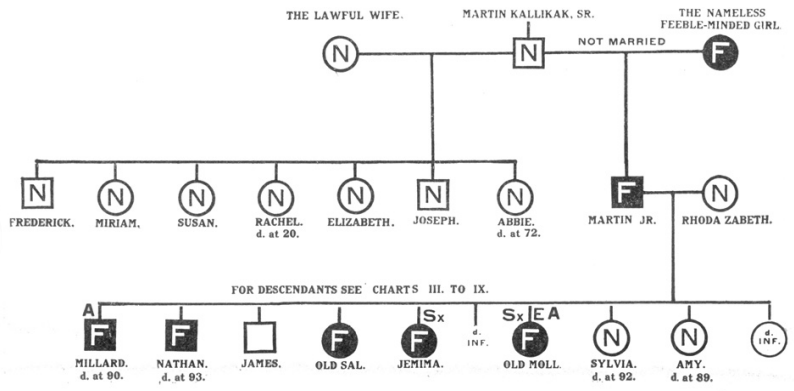


Figure 4: The Kallikak Family tree, in 1912 by Henry Goddard

1.2 Computing and Data Visualisation

In data visualisation, we mostly use ggplot as our useful tool to create so many greaet plots to represent our dataset. ggplot2 is based on the Grammar of Graphics, which simply means that you can draw each part of the graph first, and then add the parts together to form a complete graph.
As we will explore in subsequent sections, we can achieve numerous visualizations effortlessly by utilizing data in R together with ggplot. When using ggplot2, the following objects are used repeatedly, Such as geom, scale, coord, aes, stat, theme labs and so on.

"ggplot2: Elegant Graphics for Data Analysis" is a book written by Hadley Wickham, focused on teaching the use of the ggplot2 package in R for data visualization. The book thoroughly covers the principles, usage, and advanced techniques of ggplot2, making it an essential resource for learning and mastering this tool.
geom refers to Geometric Objects. Geometric objects are key components of ggplot2 and are used to define how data is visually represented in a plot. Each geom function corresponds to a specific type of graphical representation in a chart.
Scales map data to the aesthetic attributes of a graphic, such as color, size, and shape. In ggplot2, scale functions allow you to adjust the details of these mappings, such as the choice of colors, the format of labels, the layout of legends, and more.
Chord talks about how data coordinates are mapped to the plane of the graphic. It provdes axis and gridlines to make it possible to read the graph. We can use Cartesian coordinate system, polar coordinates and map projections and so on.
Faceting is a powerful feature that allows you to split one plot into multiple plots based on a factor (or factors) included in the dataset. This is particularly useful for exploring and presenting data that has multiple groups or categories.
The **theme** function plays a crucial role in customizing the non-data components of your plots. The theme system in ggplot2 allows you to fine-tune the aesthetic details of your plot, such as fonts, labels, legends, and background colors. It is an essential tool for making your plots more readable and for creating visually appealing graphics that can be tailored to specific audiences or publication requirements.

1.3 Datasets

In this section, we unveil the datasets used throughout our study. This section delves into the comprehensive depiction of the diverse datasets employed. Each dataset is meticulously introduced, elucidating its source, structure, and relevance to our investigation.

The Mtcars dataset: The *Mtcars* dataset, available as a built-in dataset in R, offers a glimpse into the automotive world of the early 1970s. This dataset encompasses 11 attributes for 32 distinct car models. Some of the variables included are: mpg: Miles per Gallon, cly: Number of cylinders , hp: Horsepower , and wt: Weight of the car in tons .

The Tooth Growth dataset: *The ToothGrowth dataset*, available as a built-in dataset in R, offers the impact of vitamin C on the tooth growth of Guinea pigs. The dataset consists of 60 observations and 3 variables: len: Length of the Guinea pigs' teeth, supp: Method of vitamin C supplementation, and dose: Dose of vitamin C in milligrams per day.

The Edgar Anderson's Iris dataset: The Anderson's iris data, available as a built-in dataset in R, offers the measurements in centimeters of sepal length and width, petal length and width, along with the species name for 50 flowers from each of three species of iris. The dataset consists of 5 variables: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species: Species name.

The annual fire in Brazil dataset: Open-source fire observation data is provided on a global scale by NASA. For the purpose of our project, the analysis was focused on Brazil. The dataset covers the year 2013 to 2022, with over 200,000 observations annually. Each observation includes crucial information such as latitude, longitude and date of observation. Notably, the dataset contains the variable confidence, ranging from 0% to 100%. This variable quantifies the level of confidence associated with each observation being a fire occurrence. For the reliability of the result, we filtered all observations with a confidence level $\geq 95\%$ [**?**].

The exchange rate data: The exchange rate data, available at the Bank of England, provides daily spot exchange rates against GBP over the time period from 2005 to now (without weekends). A subset of daily spot exchange rate of CNY (Chinese Yuan), CAD (Canadian Dollar), EUR (Euro), HKD (Hongkong Dollar), and USD (US Dollar) against GBP (Pounds Sterling) from January 2013 to October 2023 was used in this report. The dataset contains 6 variables: 'Date': Date of spot exchange rate and spot exchange rate of CNY, CAN, EUR, HKD, and USD against GBP.

Statistical GIS Boundary Files for London: This file, offered by Greater London Autority (GLA) and available at London Datastore, provided a range of key GIS boundary files for ESRI and Map Info covering Greater London, including shape files of London Wards and London Boroughs. This data set includes Name: Name of ward, District: Name of borough, Geometry: Pair of longitude and latitude.

The crime levels by borough: This data set, offered by Metropolitan Police Service and available at London Datastore, counted the number of crimes at three different geographic levels of London (borough, ward, LSOA) per month, according to crime type from 2010 to 2021. This data set contain variables like LookUp_BoroughName: Name of Borough and X201004: count of crimes during April 2010.

The Population by Ward and Borough: This file, offered by Greater London Autority (GLA) and available at London Datastore, provided population for 2001 to 2050 for London wards and boroughs. This data set includes variables like Name: Borough name, Year: population of the year, Population: population of the borough at that year.

Trees dataset: The trees dataset, available as a built-in dataset in R, offers measurements from 31 felled black cherry trees and provides insights into the relationship between a tree's girth, its height, and the volume of timber it can produce. The dataset contains 3 variables: Girth: The diameter of the tree, Height: The height of the tree, Volume: The volume of timber that the tree can produce.

1.4 Structure and Organisation of the Thesis

The remainder of this thesis is composed by, firstly, Chapter 2: "Theoretical Foundations of Data Visualisation", an introductory section that lays the theoretical foundation for the subsequent discussions.

The crux of this document, Chapter 3: "Modern Methods of Data Visualisation", conducts a detailed exploration of various modern methods of data visualisation. This chapter offers an in-depth analysis and critical evaluation of their applications, strengths, and limitations.

Chapter 4: "Practical Implementation" ventures into the practical application of Python Dash and R Shiny for constructing interactive data visualisation dashboards. Subsequently, Chapter 5: "Case Studies" presents case studies, which serve as practical demonstrations of the efficacy and relevance of the discussed visualisation methods in resolving real-world problems. Finally, Chapter 6 "State-of-the-Art Approaches" critically examines state-of-the-art approaches in data visualisation, highlighting emerging trends, methodologies, and technologies in the field.

2 Theoretical Foundations of Data Visualisation

This chapter, "Theoretical Foundations of Data Visualisation," delves into the core principles and concepts that serve as the base of this field. We seek to understand not only the "how" but also the "why" behind the creation of visualisations that captivate and inform.

2.1 Introduction to Data Visualisation Theory

Creating effective data visualisations requires a robust theoretical framework underlying every chart, graph, or plot. These theoretical underpinnings not only form the basis of data visualisation but also influence how we represent, perceive, understand, and interpret data.

Guiding Principles for Data Representation
The theoretical framework of data visualisation involves guiding principles dictating visual representation of data. These principles include accuracy, emphasizing faithful reflection of underlying data to reduce distortion or misinterpretation; simplicity, advocating for streamlined visuals to convey information effectively; clarity, ensuring visuals are easily understood without unnecessary complexity; relevance, presenting information pertinent to the message or question addressed; and consistency, maintaining uniform use of visual elements like color coding and labeling throughout a visualisation.

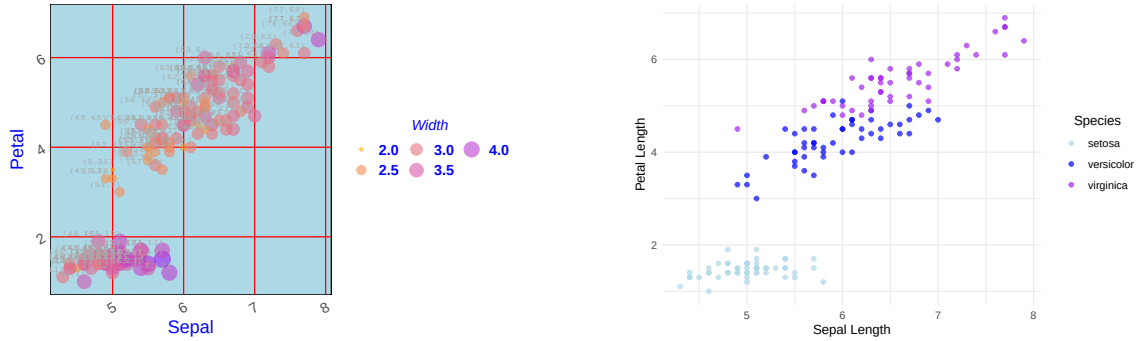


Figure 5: Comparison of visualisations of the distribution of sepal and petal legth across three flower species

Theoretical Framework and Visual Perception
Understanding how the human brain processes visual information is a fundamental aspect of data visualisation theory. This knowledge plays a crucial role in designing visualisations that effectively connect with viewers. It encompasses several key considerations: Gestalt Principles, which encompass proximity, similarity, and continuity, affecting how visual elements are grouped and interpreted; Color Theory, involving the strategic use of color contrasts and harmonies to improve clarity and impact; and the management of Cognitive Load, which emphasizes the importance of reducing mental effort needed to process information.

2.2 Visual Perception and Cognition

Here, we explore human visual perception, along with the application of cognitive psychology principles in data visualisation and highlight the crucial role of pre-attentive attributes in shaping our perception of data.

Human Visual Perception: Decoding Visual Information
Human visual perception, a remarkable cognitive process, profoundly influences our understanding of the surrounding world. When applied to data visualisation, it elucidates how individuals engage with and derive meaning from visual data representations. Significant aspects of human visual perception within data visualisation encompass pattern recognition, adept at identifying trends, outliers, and relationships in data representations. Additionally, perceptual grouping, where visually similar elements are grouped together, influences the interpretation of data clusters and shapes. Moreover, the hierarchy of perception dictates that certain visual attributes are processed more swiftly and effectively than others, such as color being processed faster than text, influencing the viewer's attention hierarchy.

By harnessing the principles of human visual perception, applying insights from cognitive psychology, and leveraging pre-attentive attributes, data visualisation designers can create visualisations that are not only aesthetically pleasing but also cognitively efficient.

2.2.1 The Gestalt Principles

The Gestalt principles play an important role in the realm of visual perception and design. We'll particularly focus on their relevance to data visualisation and strategies for creating more effective visualisations. Key Gestalt principles crucial in shaping visual information perception include proximity, which groups related elements, similarity that links similar attributes, continuity aiding trend representation, closure for implying connections, and symmetry for balance and aesthetics in visualisations.

Application of Gestalt Principles in Designing Visualisations
Leveraging Gestalt principles in data visualisation design enhances intuitive and effective information communication. Designers strategically employ these principles to group related data for clarity, minimize visual clutter, use color or shape to denote meaningful categories, establish smooth visual paths guiding the viewer's gaze, and imply connections or patterns within complex datasets.

2.3 Data Abstraction and Representation

The transformation of raw data into meaningful representations is a pivotal step in data visualisation. This process, known as data abstraction, involves distilling complex datasets into visual forms that convey insights. In this section, we explore data abstraction, the hierarchies and levels of abstraction in data visualisation, and the critical trade-offs between abstraction and the potential loss of information.

2.3.1 Data Abstraction: Transforming Raw Data

Data abstraction involves simplifying and structuring raw data into comprehensible and insightful formats. This process serves as the bridge, transforming numbers, text, and variables into visual

elements that convey patterns, trends, and relationships, forming the core of informative data visualisations.

2.3.2 Hierarchies and Levels of Abstraction

In data visualisation, abstraction operates on multiple levels of granularity. Hierarchies of abstraction allow us to represent data at varying levels of detail:

1. **Low-Level Abstraction**: At the lowest level, raw data is preserved in its most detailed form. This might include individual data points, measurements, or unprocessed text.

2. **Mid-Level Abstraction**: As we move up the hierarchy, data is grouped or aggregated to provide a broader overview. For example, hourly data points may be aggregated into daily or weekly averages.

3. **High-Level Abstraction**: At the highest level, data is represented in a condensed and abstracted form, often as summary statistics or key insights. This level provides a big-picture view.

This is represented in Figure **??**. The first visualisations of the mtcars dataset is a scatter plot that provides detailed information about the relationship between car weight and miles per gallon, with points colored by the number of cylinders. The second is an abstract visualization using a box-and-whisker plot to provide a high-level summary of the distribution of miles per gallon for different numbers of cylinders. Finally, the third visualisation is a bar plot presenting aggregated information about the average miles per gallon for different numbers of cylinders.

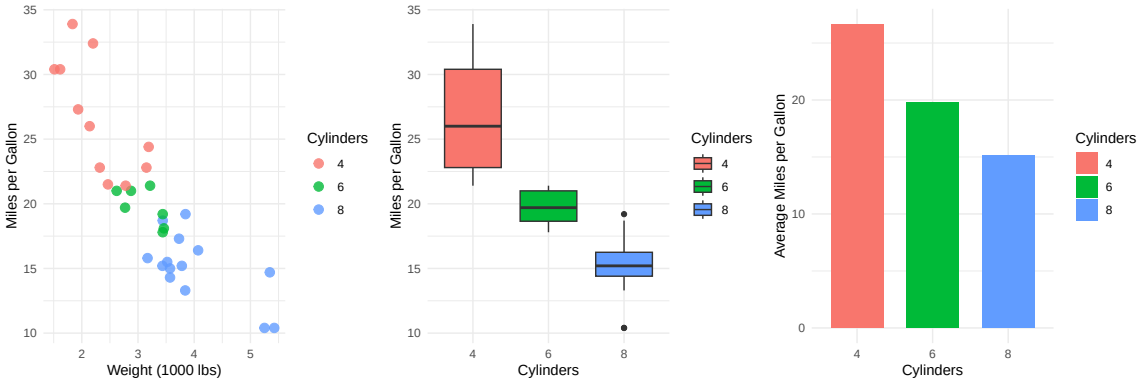


Figure 6: Mtcars dataset visualised on 3 different levels of abstraction

Trade-offs Between Abstraction and Information Loss
While abstraction simplifies complex data, it presents trade-offs. Designers of data visualisation must strike a balance between clarity and detail, generalization and specificity, and context versus precision. Abstraction increases clarity but may sacrifice crucial detailed information necessary for some analytical tasks. It offers a more generalized view accessible to a wider audience but might overlook specific nuances essential for experts. While providing valuable context, high-level abstraction may lack the precision required for precise decision-making.

In data visualisation, the art of data abstraction lies in finding the right level of detail that effectively conveys the intended message while minimising the risk of information loss. This balancing act is a critical consideration in the design of informative and meaningful data visualisations.

2.4 Data Types and Visualisation Techniques

In the world of data visualisation, understanding the nature of your data is key. Data comes in various types, and selecting the appropriate visualisation technique is contingent upon recognising these distinctions. In this section, we categorise data types, and demonstrate how to match each data type with suitable visualisation techniques.

2.4.1 Categorisation of Data Types

Data types can be broadly categorised into four main types:

- **Nominal data**: nominal data represents categories or labels without any inherent order. Examples include colours, gender categories, and city names.
- **Ordinal data**: ordinal data implies a meaningful order or ranking among categories but lacks equal intervals between them. Examples include survey responses (eg. "very satisfied", "satisfied", "neutral", "dissatisfied", "very dissatisfied")
- **Interval data**: interval data possesses ordered categories with equal intervals between them, but it lacks a true zero point. Temperature is measured in Celsius or Fahrenheit as an example.
- **Ratio data**: ratio data includes ordered categories with equal intervals and a meaningful zero point. Examples are age, income, and weight.

2.4.2 Matching Data Types with Appropriate Visualisation Techniques

Selecting appropriate visualisation techniques is essential for effective data communication. Various data types demand specific visualisation methods for optimal representation. For nominal data, bar charts and stacked bar charts are effective in displaying categorical information and relative proportions. Ordinal data benefits from ordered bar charts, dot plots, or stacked bar charts, maintaining the ranking and order of categories. Interval data is best visualised using line charts, histograms, and box plots, showcasing trends and distributions without assuming a true zero point. Ratio data finds effective representation through scatter plots, histograms, and line charts, enabling precise comparisons and measurements due to the presence of a meaningful zero point.

2.5 Colour Theory in Data Visualisation

Here, we explore the significance of colour in data visualisation, the principles of colour perception and encoding, and the importance of avoiding misleading visualisations through thoughtful colour choices.

The Importance of Colour in Conveying Information
Color significantly enhances the impact and comprehension of data visualisations. It serves multiple purposes: distinguishing data points, emphasizing trends, and offering contextual information. It is utilized to encode categorical data, differentiating between various groups with distinct colors, and to represent quantitative data by utilizing color intensity or gradients to portray values or magnitudes. Additionally, color is instrumental in adding context to visualisations through background

elements, labels, or annotations, imparting meaning to the data.

Colour Perception and Colour Encoding in Visualisations
Understanding color perception in data visualisation is crucial. Key principles involve considering color discrimination, ensuring accessibility for individuals with color vision deficiencies, as is illustrated by figure **??**. Careful selection of color schemes aligned with the intended message is essential—for instance, using warm colors like red and orange to indicate caution or warmth, and cool colors like blue and green to convey calmness or coldness. Additionally, attention should be paid to how colors interact when combined; certain combinations might create visual vibrations or impact text legibility.

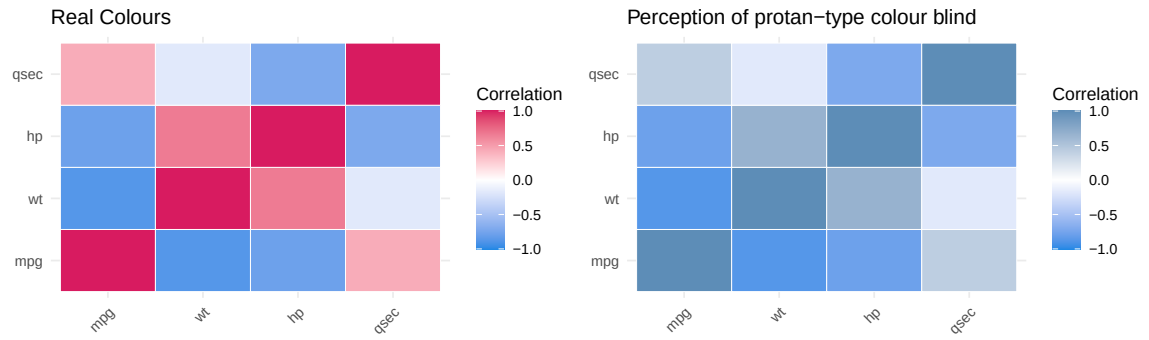


Figure 7: Colour perception of a heatmap for by a colour blind person

Avoiding Misleading Visualisations Due to Colour Choices
Misleading visualisations often stem from inappropriate or deceptive use of color, requiring precautions to prevent such occurrences. First, maintaining consistency in color usage throughout the visualisation is essential. Employing a uniform color scheme for similar data categories or elements helps establish coherence and understanding. Furthermore, it's crucial to avoid color choices that could distort or exaggerate the data. Overly intense or contrasting colors might mislead interpretations, emphasizing the necessity for judicious color selection.
Additionally, providing a clear and concise legend becomes imperative to explain the meaning of colors, especially when dealing with complex or unfamiliar color schemes. A comprehensive legend helps viewers decipher the represented data accurately. User testing stands as another crucial step in the process. Conducting thorough user testing ensures that the chosen color palette effectively conveys the intended message without confusion or misleading the audience. This step validates the visual interpretation and aids in making necessary adjustments to enhance clarity and accuracy in data representation.

2.6 Theoretical Properties of Visualisations

Effective data visualisation extends beyond aesthetically pleasing graphics; it involves adhering to crucial theoretical properties that enhance expressiveness, precision, accuracy, and scalability in visual representations. This section examines key properties such as expressiveness, effectiveness, the data-ink ratio, principles of minimal ink, as well as precision, accuracy, and scalability.

To begin with, we'll define the key concepts that frame this section:

- **Expressiveness**: Visualisations should be expressive, meaning they should effectively communicate the intended message or insights within the data. Expressive visualisations capture the richness and complexity of the underlying data, revealing patterns, trends, and relationships.
- **Effectiveness**: An effective visualisation is one that successfully conveys information to its audience. It allows viewers to understand the data, draw meaningful conclusions, and make informed decisions based on the presented information.

2.6.1 Data-Ink Ratio and the Principle of Minimal Ink

This **Data-Ink Ratio principle**, introduced by Edward Tufte, emphasises maximising the ink (or pixels in digital formats) used to represent the actual data while minimising non-essential ink. A higher data-ink ratio results in a cleaner, more efficient visualisation that reduces clutter and enhances comprehension.
The **Principle of Minimal Ink** builds on the data-ink ratio. This principle advocates for the removal of any visual elements that do not contribute to the viewer's understanding of the data. Eliminating unnecessary ink (e.g., excessive gridlines or decorations) simplifies the visualisation without sacrificing its effectiveness.

Precision, Accuracy, and Scalability
The concepts that can be mobilised in order to comply by these principles are precision, accuracy, and scalability. Specifically in the context of data visualisation, precision involves striking a balance between presenting sufficient detail for accurate interpretation while avoiding overwhelming complexity. Accuracy is also vital as its role is ensuring faithful representation of true data values to prevent misleading conclusions. Scalability addresses a visualisation's adaptability to varying data sizes and resolutions, demanding the capability to represent both small and large datasets without compromising clarity or performance.

2.7 Cognitive Load and Visual Complexity

In data visualisation, achieving a balance between complexity and cognitive load is crucial. This section explores the concept of cognitive load in visualisations, strategies to reduce cognitive load while maintaining complexity, and techniques to combat information overload through simplification.

Exploring the Concept of Cognitive Load in Visualisations
In data visualisations, cognitive load significantly influences how viewers engage with and comprehend presented data. Striking a balance is crucial to effectively convey information without overwhelming the viewer's cognitive capacity.

2.7.1 Strategies to Reduce Cognitive Load While Maintaining Complexity

To reduce cognitive load while maintaining complexity in data visualisation, several strategies can be employed. Firstly, establishing a clear visual hierarchy using size, color, and contrast helps direct attention to crucial elements. Additionally, simplifying labels and text by avoiding unnecessary complexity and jargon ensures information is clear and easily digestible. Employing interactive features like tooltips and drill-down functionality assists in providing additional information when

required, reducing the density of static visualisations. Another approach involves the use of progressive disclosure, presenting complex information gradually, beginning with an overview and allowing users to explore details as needed. Lastly, considering data aggregation where appropriate can help summarize information and alleviate the cognitive load associated with interpreting intricate details.

These strategies aim to maintain complexity while lessening the cognitive burden on viewers by directing attention effectively, simplifying content, offering interactive elements, gradually revealing information, and summarizing data where feasible.

2.7.2 Information Overload and Simplification Techniques

Addressing information overload in visualisations necessitates the strategic application of simplification techniques. Filtering enables focused data selection, while data reduction aggregates information to highlight overarching trends. Storyboarding structures data presentation, aiding in contextual comprehension, and prioritization ensures critical information is prominently displayed, elevating the visualisation's clarity and impact. These strategies collectively combat overwhelming data or excessive visual elements, enhancing comprehension and the effective communication of insights to viewers.

3 Modern Methods of Data Visualisation

In this chapter, we explore a variety of powerful visualisation methods, from classic scatter plots and bar charts to advanced techniques like heatmaps and network graphs. Through vivid examples, we'll show when and why each method is used, and delve into the theoretical and mathematical foundations that empower these visualisations to unveil insights hidden within the data.

3.1 Scatter Plots and Bubble Charts

Scatter plots and bubble charts are fundamental data visualisation techniques that provide valuable insights into the relationships and patterns within datasets. These visualisations are particularly effective for representing discrete data through data points, since this brings out easily identifiable comparisons, and reveals trends.

3.1.1 Scatter Plots

A scatter plot is a graphical representation of a set of data points in a two-dimensional coordinate system. Each data point is represented by a dot, and the position of the dot is determined by the values of two variables.
In general, Y denotes the response variable and X denotes the explanatory variable. Let (x_i, y_i) represent the coordinates of the i-th data point on the scatter plot. The scatter plot can be mathematically described as a set of points:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}.$$

Here, n is the number of observations in the set.

The mathematical interest of dot plots lies in their ability to provide a simple visual representation of data distribution, center, and spread. While they don't rely on complex equations or statistical principles, dot plots make it easy to observe important characteristics of data, such as the mode, median, variance, skewness and outliers.

In addition, the arrangement of the points on the plot can provide insights into the nature of the relationship between X and Y.When points show a linear trend, it implies a strong correlation between the variables. When the points are scattered with no clear pattern, it signifies a low correlation between the variables.

3.1.2 Regression and the Regression Line

Regression models are statistical tools that provide functions to estimate the relationship between the response variable and one or more independent variables. Regression analysis is widely adopted by data scientists, who use large datasets to build predictive models for trend forecasting. In the following paragraph, we will introduce linear regression models and demonstrate their usage using the mtcars dataset.

Simple Linear Models

In the simple linear model, we assume that the responses $\mathbf{Y} = (Y_1, Y_2, ..., Y_n)$ are uncorrelated, with a common variance σ^2. The explanatory variable is represented as a vector $\mathbf{x} = (x_1, x_2, ..., x_n)$, where x_i is the i-th observation. We can express the n expectations in vector form:

$$\mathrm{E}(\mathbf{Y}|\mathbf{x}) = \begin{pmatrix} \beta_0 + \beta_1 x_1 \\ \beta_0 + \beta_1 x_2 \\ \vdots \\ \beta_0 + \beta_1 x_n \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \beta_0 + \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \beta_1 = \mathbf{1}_n \beta_0 + \mathbf{x}\beta_1 [\mathbf{?}].$$

The expectation can be rewritten as:

$$\mathrm{E}(\mathbf{Y}|\mathbf{x}) = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \mathbf{X}\beta.$$

Least Squares Estimation

The coefficients β_0 and β_1 are calculated using the method of least squares estimation. The residuals are the differences between the response variables y_i and responses generated by the linear regression model $\mathrm{E}(\mathbf{Y}_i|\mathbf{x})$. The goal is to minimise the residual sum of squares (RSS), denoted by Q:

$$\mathrm{Q} = \sum_{i=1}^{n} \{y_i - \mathrm{E}(Y_i|\mathbf{X})\}^2.$$

The RSS is a measure of the goodness of fit in a regression model, representing the sum of squared residuals. The least squares estimator is the method used to find the parameter values that minimize the RSS, providing the best-fitting linear regression model.

After expanding Q in terms of vectors and matrices, the partial derivative of Q with respect to vector β is:

$$\frac{\partial Q}{\partial \beta} = 2(\mathbf{X}^T\mathbf{X}\beta - \mathbf{X}^T\mathbf{y})[\mathbf{?}],$$

where $\mathbf{y}$ is the vector of response variables.

Equating $\frac{\partial Q}{\partial \beta} = \mathbf{0}$, the vector $\hat{\beta}$, the least squares estimate of β, can be written as:

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\beta}) = \mathbf{0}.$$

Since, in the case of simple linear models, $\mathbf{X}$ is of full rank n, $(\mathbf{X}^T\mathbf{X})^{-1}$ exists, and there is a unique least square estimate of β given by:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

The Band of Confidence:

When discussing regression lines, it is essential to consider the band of confidence. This is represented by confidence intervals around the regression line, reflecting the uncertainty associated with predictions. The wider the interval, the greater the uncertainty. Commonly, a 95% confidence interval is used, indicating that there is a 95% probability that the true regression line lies within the specified interval.

Relation to Scatter Plots

Regression lines find their visual counterpart in scatter plots. A scatter plot displays the observed data points on a graph, with the regression line weaving through them. The closeness of the data points to the regression line indicates the strength of the relationship between the variables. Additionally, the spread of the data points around the line provides insights into the variability and potential outliers.

Scatter Plots in Practice

In the following example, the mtcars dataset will be used to demonstrate a linear regression model. The focus lies on assessing the efficiency of automobiles, with the aim of constructing a linear regression model that utilizes various attributes to predict the performance of the automobile, measured in Miles Per Gallon (mpg).

For exploratory investigation, the correlation matrix is to be calculated, rounded to 3 significant figures, see Figure **??**.

Excluding highly correlated covariates is a critical consideration in constructing a regression model. When explanatory variables in a regression model are highly correlated, this will lead to multicollinearity. Multicollinearity implies that the individual effects of correlated variables on the response variable become intertwined. This intertwinement can result in erratic changes in the coefficient estimates of the regression model in response to small changes in the data. Such instability makes it challenging to interpret and trust the specific contributions of each variable to the model's predictions[**?**].

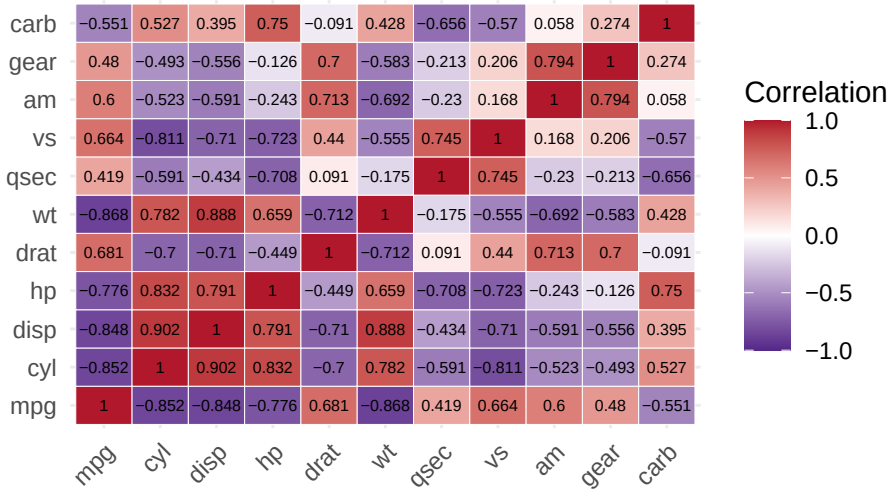


Figure 8: Correlation matrix of variables in mtcars dataset

Examining the correlation matrix reveals a high correlation between the response variable, Miles Per Gallon (mpg), and the explanatory variable, weight (wt). This is advantageous, suggesting that wt may have strong predictive power for mpg. In the summary of the linear regression model `Modelwt`, note that the t-test yields a p-value less than 0.001. This indicates that the variable wt is of high statistical significance in this model.

```
Modelwt <- lm(formula = mpg ~ wt, data = mtcars)
summary(Modelwt)

##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528,Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

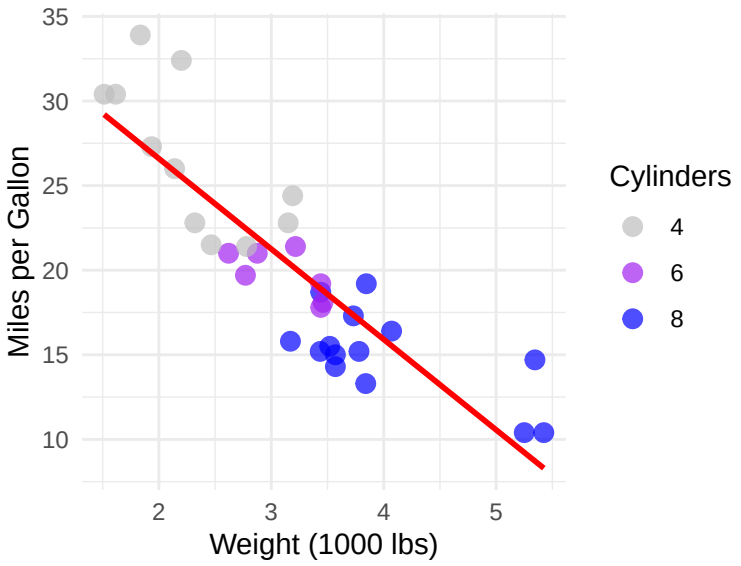


Figure 9: Scatter plot of car weights vs MPG

Model selection methods, such as the Akaike Information Criterion (AIC), play a crucial role in statistical modeling. An algorithm which minimises AIC score helps in selecting the most appropriate regression model from a set of explanatory variables. Further details on AIC can be found in section **??**.

The R algorithm `step()` minimises the AIC score at each step for optimal regression model selection. Starting with the regression model which contains all explanatory variables `ModelAll`, the algorithm iteratively exclude one covariate per step.

This algorithm aims to find the best-fitting model by balancing goodness of fit and simplicity. Note that `step()` provides a competitive ranking among models, not an absolute measure of model quality.

For detailed procedure of this AIC algorithm, using `summary(ModelBest)`, please see the output in **??**.

```
ModelAll <- lm(formula = mpg ~ ., data = mtcars)
ModelBest <- step(ModelAll)
```

Analysis of the scatter plot

Figure **??**, provides insights into the relationship between cars' weight and their MPG, with the added dimension of color-coded cylinders.Particularly it visually highlights and make accessible to the viewer features of the data set, that would otherwise go unnoticed. These are some of the following elements:

- **Clustering**: The scatter plot reveals distinct clustering of data points, highlighting specific patterns within the dataset. Cars with four cylinders (color "grey") are predominantly clustered in the lower weight and higher MPG region, representing smaller and more fuel-efficient

vehicles. In contrast, cars with eight cylinders (color "blue") tend to be clustered in the higher weight and lower MPG area, indicating larger and less fuel-efficient cars. The identification of this clustering aids iAIC algorithmn visualising how the number of cylinders influences the trade-off between weight and fuel efficiency.

- **Linear Regression Line**: The regression line provides a visual representation of the overall relationship between car weight and fuel efficiency.If the line has a positive slope, it indicates that as car weight increases, MPG decreases. Conversely, a negative slope suggests that heavier cars tend to have higher MPG. The steepness of the line represents the strength of this relationship. In this case, the reed regression line indicates a negative correlation—cars tend to have lower fuel efficiency as their weight increases.

3.1.3 Bubble Charts

Bubble charts are a captivating data visualisation tool that extends beyond the typical two-dimensional scatter plot by introducing an extra dimension. They represent data points as bubbles or circles on a two-dimensional plane, where the size of each bubble encodes a third variable. This technique enhances data visualisation by facilitating the exploration of multivariate data and uncovering patterns that may be hidden in traditional scatter plots.

Bubble Chart's Utility in Visualising Data
Bubble charts excel in scenarios where three key variables need to be conveyed simultaneously. The x-axis and y-axis represent two variables, as in a standard scatter plot, while the size of the bubble encodes a third variable, often a quantitative one. This allows for the visualisation of relationships between three variables in a single, intuitive graphic.
For instance, in economics, bubble charts can illustrate economic indicators, with the x-axis showing time, the y-axis displaying GDP growth, and the bubble size representing a related factor like population or inflation.

Mathematical Intricacies
The mathematical intricacies of constructing bubble charts involve scaling the data values to determine the size of each bubble accurately. The size of the bubble is typically proportional to the square root of the variable it represents. The choice of scaling method depends on the data distribution and the message the chart aims to convey.

The formula for calculating the bubble size (S) often involves applying a linear or nonlinear scaling function:

$$S = k \cdot \sqrt{V}$$

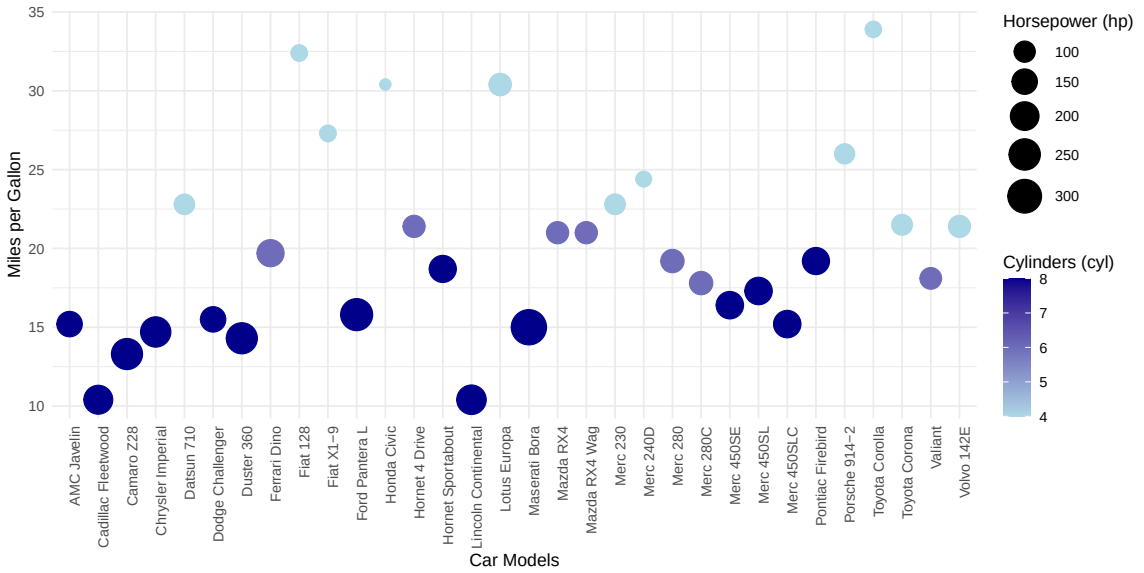
Where:

- S is the size of the bubble,
- V is the value of the variable being represented, and
- k is a scaling factor to control the bubble size.

Selecting an appropriate scaling factor (k) is critical for maintaining the proportionality between the bubble size and the variable being represented.

Bubble Charts in Practice This bubble plot visualises data from the same dataset as above. The purpose of this plot is to depict the relationship between car models and their fuel efficiency (mpg) while using the size of the bubbles to represent the car's horsepower (hp) and color-coding the bubbles based on the number of cylinders (cyl).

```
#Create bubble plot
ggplot(mtcars, aes(x = rownames(mtcars), y = mpg, size = hp, color = cyl)) +
  geom_point() +
  labs(
    x = "Car Models",
    y = "Miles per Gallon",
    size = "Horsepower (hp)",
    color = "Cylinders (cyl)"
  ) +
  scale_size_continuous(range = c(3, 10)) +
  scale_color_gradient(low = "lightblue", high = "darkblue") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust =1))
```



Analysis of the bubble chart

The plot's title, axis labels, and legends provide context and clarity to the visualisation, making it accessible and informative. Additionally, the choice of a gradient color scale for the number of cylinders enhances the visual appeal and aids in interpreting the data.This bubble plot allows for quick comparisons between multiple characteristics of different car models.The resulting bubble plot effectively conveys several key insights:

1. **Car Model vs. MPG**: The x-axis displays the car models, offering a clear representation of each vehicle in the dataset. The bubble plot is particularly useful for displaying nominal data, such as car model names, as it allows easy identification and comparison.

2. **Miles per Gallon (MPG)**: The y-axis measures miles per gallon, representing the fuel efficiency of each car model. Higher bubbles indicate better fuel efficiency. This variable, which is continuous, is positioned vertically to demonstrate how each car model's fuel efficiency relates to others.

3. **Horsepower (HP)**: The size of each bubble represents the car's horsepower (hp). Larger bubbles correspond to higher horsepower, providing an additional dimension to the data. The size encoding helps identify more powerful cars.

4. **Cylinders (Cyl)**: The color of each bubble is determined by the number of cylinders (cyl) in the car's engine. The color scheme adds a categorical aspect to the visualisation, making it easy to differentiate between cars with different cylinder counts.

subsectionBar Charts and Histograms

3.1.4 Bar Charts

A bar chart is a very important method to present data. It organizes information into vertical bars. Bar charts have lots of advantages in data visualisation. It can present data categories in a frequency distribution. A bar chart is best for comparing classified data. Especially when the values are close, because the human perception of height is better than other visual elements (such as area, angle, etc.), the use of a bar chart is more appropriate. These bars usually have different lengths, and every length is proportional to the size of the information they present.
R uses the function *barplot*() to create bar charts. R can draw both vertical and Horizontal bars in the bar chart. In the bar chart, each of the bars can be given different colors.
R is a programming language for data analysis and statistical computing, and its advent has made data visualisation more straightforward and accessible. Among the various tools available in R, ggplot2 stands out as one of the most renowned and powerful tools for creating data visualisations. It offers a wealth of data visualisation capabilities and is celebrated for its versatility and aesthetic appeal. In this chapter, we will focus on how to use ggplot2 to create bar charts for data visualisation.

3.1.5 Different Types of Bar Charts

Here is an overview of the different types of bar charts.

Vertical Bar Chart This is the most common bar chart. We use different vertical columns to display and compare the values of different categories in the same dimension, where the X-axis represents the contrasting categories and the Y-axis represents the frequency or count of their categories.

Horizontal Bar Chart This is very similar to a vertical bar chart but rotated 90 degrees. Categories are shown on the y-axis and frequency or count are shown on the x-axis. Horizontal bar charts are especially useful when category names are long or when there are numerous categories.

Multi-set Bar Chart Also known as a grouped bar chart or clustered bar chart. A multi-set bar chart is used to represent and compare different sub-groups within individual categories. This type of chart is useful when you want to show and compare multiple sets of data side-by-side. Multi-set Bar charts can be horizontal or vertical like the other normal bar charts, and the length of each bar represents the frequency or count of their categories.

Stacked bar chart Similar to bar charts, stacked bar charts are often used to compare different classes of values and, within each class of values, are divided into sub-classes, which are often referred to by different colors. Each segment's size is proportional to the frequency or count that it represents from the sub-category. The entire bar's length represents the cumulative total of all the sub-categories. However, it is very easy to get confused when there are too many categories. Bar charts excel due to their structural simplicity, ease of comprehension, straightforward comparison of different data categories, and versatility for representing various data types and multilevel information

The disadvantages of bar charts include limited suitability for large datasets, potential misinterpretation when lacking a zero baseline, difficulty in handling numerous categories, and their preference for categorical data over continuous data trends, where line graphs are more suitable.

3.1.6 ToothGrowth Dataset

This bar chart below illustrates the tooth growth in relation to varying doses of a vitamin. The key observations are:

1. **X-axis Description:** The X-axis represents different dosages of the vitamin (mg/day). There are three distinct dosage levels.
2. **Y-axis Description:** The Y-axis signifies the length of tooth growth (len). This represents the average tooth growth at the given vitamin dosage.
3. **Data Observation:** From the heights of the bars, it is evident that as the vitamin dosage increases, the tooth growth also appears to increase. This might suggest that higher doses of the vitamin may promote tooth growth.

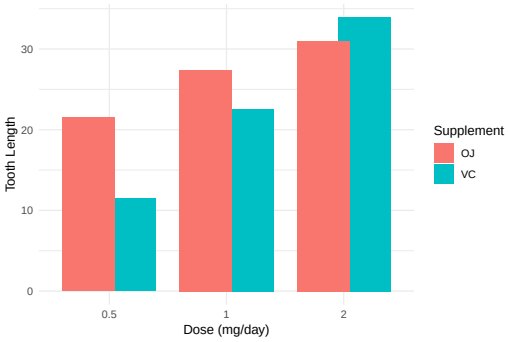


Figure 10: Tooth Growth by Dose and Supplement (grouping bar chart)

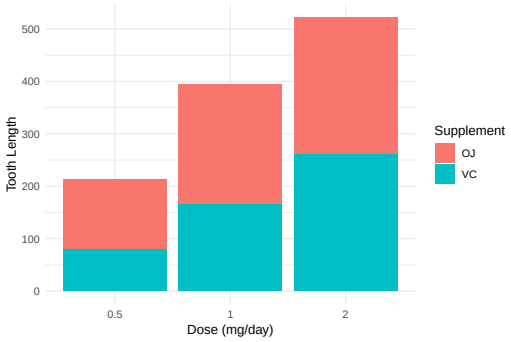


Figure 11: Tooth Growth by Dose and Supplement (stacked bar chart)

The displayed bar chart **??** provides insights into tooth growth influenced by varying doses of a vitamin, further categorized by the type of supplement ('supp'). The key insights from this chart are:

1. **X-axis Description:** The X-axis demarcates different vitamin dosages, categorized into three distinct levels: 0.5, 1, and 2 mg/day.

2. **Y-axis Description:** The Y-axis quantifies tooth growth length, representing the combined average growth for both supplements at the respective vitamin dosages.

3. **Data Observation:** The total height of each bar signifies the combined tooth growth for both supplements at the given dosage. From the stacked sections, it's evident that the impact on tooth growth varies based on the supplement type. A detailed inspection might elucidate the relative effectiveness of the supplements at each dosage level.

The structure of the second figure **??** is quite similar to that of the first one, with the main difference lying in the method of data representation. Forming a bar chart, it facilitates the understanding of the combined effects of the two supplements at each dosage level. However, compared to the grouped bar chart, it becomes more challenging to differentiate the individual contributions of each supplement.

In the next part of our section, we will look at another plot which called Histogram.

3.1.7 Histograms

Histograms, although visually similar to bar charts, convey different meanings. A histogram involves concepts of statistics. It requires data to be categorized into groups and then counts the data points within each of those groups. On a Cartesian coordinate system, the x-axis shows the endpoints of each group, and the y-axis represents frequency. The height of each rectangle indicates the corresponding frequency, making it a frequency distribution histogram. In order to determine the quantity of each group in the histogram, a multiplication of the frequency by the group interval is necessary. Since every histogram has a fixed group interval, if we use the y-axis to directly show quantity and each rectangle's height indicates the number of data points, we can both retain the distribution and simultaneously see the number in each group at a glance. All examples in this text use the non-standard histogram depiction with the y-axis denoting quantity.
Uses of Histograms: Histograms demonstrates the distribution of frequency or quantity across groups. Facilitates the visualisation of differences in frequency or quantity among groups. The R language uses the `hist()` function to create histograms. This function takes vectors as input and uses a few more parameters to plot the histogram.
Now, we want to create a better graph with ggplot2 thanks to the `geom_histogram()` function and iris dataset.
The iris dataset is a classic dataset in the field of statistics. It was introduced by the British biologist Ronald A. Fisher in 1936 as an example of discriminant analysis. The dataset consists of 150 samples from three species of iris flowers: setosa, versicolor, and virginica. For each sample, four features were measured: the lengths and the widths of the sepals and petals, all in centimeters. The dataset is often used for classification tasks to differentiate between the three species based on the given measurements. It has become a standard test case for many classification algorithms and is widely recognized in the data science community.

```r
#Create a histogram using ggplot2 for Sepal Length in the Iris dataset
bar_diagram <- ggplot(iris, aes(x = Sepal.Length)) +
  geom_histogram(
    binwidth = 0.2,  # Adjust the box width to 0.2 for smaller data sets
    fill = "grey",
    color = "black"
  ) +
  labs(
    x = "Sepal Length/cm",
    y = "Frequency"
  ) +
  theme_minimal()
print(bar_diagram)
```

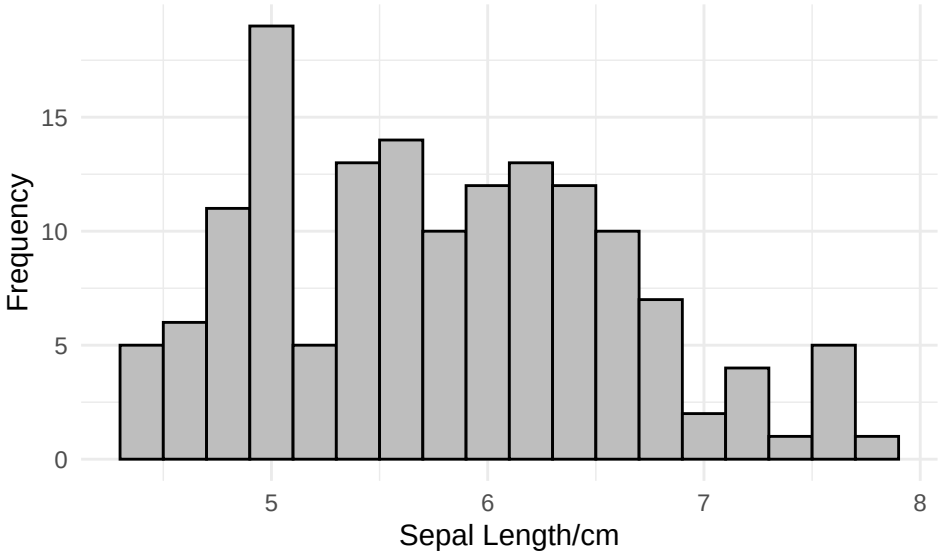


Figure 12: Histogram of Sepal Length in Iris Dataset

3.1.8 Kernel Density Estimation

Kernel Density Estimation(KDE) is an very useful tool in statistics. In stead of discrete histograms, it helps us to create a smooth curve given by a dataset. KDE is used to infer the distribution of a population based on a limited sample. Thus, the result of the kernel density estimation is an estimate of the sample's probability density function. Based on this estimated probability density function, we can ascertain certain characteristics of the data distribution, such as the regions where data is concentrated. The KDE algorithm takes a parameter, bandwidth, that affects how "smooth" the resulting curve is. Changing the bandwidth changes the shape of the kernel: a lower bandwidth means only points very close to the current position are given any weight, which leads to the estimate looking squiggly; a higher bandwidth means a shallow kernel where distant points can contribute.

We can express KDE as follows,where the K represent the kernel function.

$$\hat{f}(x) = \sum_{observations} K\left(\frac{x - observation}{bandwidth}\right)$$

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

The kernel function $K(u)$ is a normalized non-negative function that satisfies:

$$\int K(u)\, du = 1$$

```r
# Create a Kernel Density Estimate plot using ggplot2 for Sepal Length in the Iris dataset
kde_diagram <- ggplot(iris, aes(x = Sepal.Length)) +
  geom_density(
    fill = "grey",
    alpha = 0.5,  # Adjust the transparency for better visualisation
    adjust = 1  # This parameter can be used to control the smoothness
  ) +
  labs(
    x = "Sepal Length/cm",
    y = "Density"
  ) +
  theme_minimal()
print(kde_diagram)
```

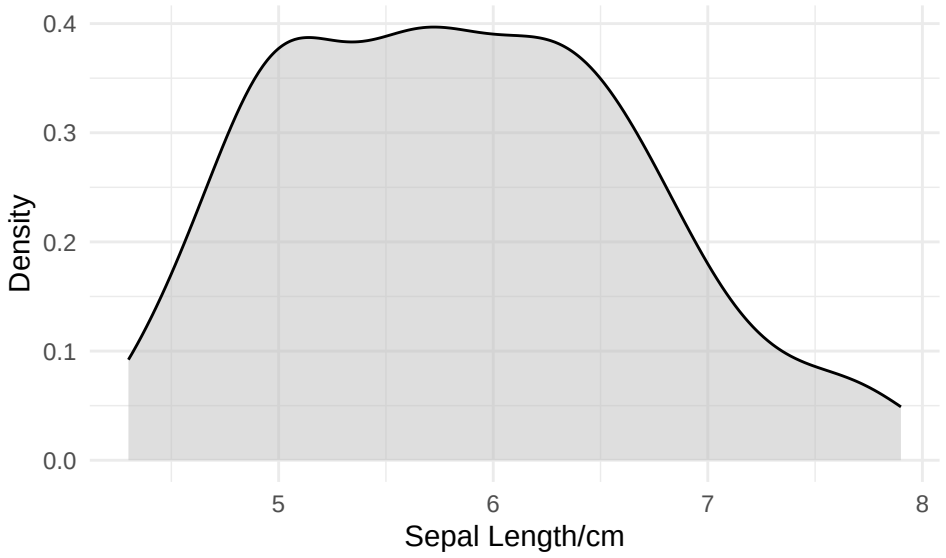


Figure 13: Kernel Density Estimation of Sepal Length in Iris Dataset

In figure **??** we present the kernel density estimation of our dataset. The kernel density curve will display the distribution of sepal lengths. The peaks of the curve correspond to the main concentration trends of sepal length in the data. If the curve is unimodal, it means that the sepal lengths of most irises are concentrated in that region; if it is bimodal or multimodal, this indicates the presence of multiple such concentration areas.

3.2 Heatmaps and Tree Maps

In this chapter, we explore two powerful data visualisation techniques: heatmaps and treemaps. These methods are instrumental for conveying intricate data structures and patterns, offering unique ways to represent multivariate information, making them indispensable tools for data scientists.

We will delve into the theory behind heatmaps and treemaps, understand how to create them using popular data visualisation libraries, and demonstrate their practical applications with real-world examples. By the end of this chapter, you will be well-equipped to leverage heatmaps and treemaps to gain insights from complex and hierarchical datasets.

3.2.1 Heatmaps - Fire in Brazil

The heatmap is a data visualisation technique that uses colour coding to represent different intensity.

In this illustrative example, heatmaps are used to visualise fire occurrences in Brazil. These heatmaps provide a spatially coherent representation, highlighting regions at high risk and seasonal patterns. Here, the heatmap is a powerful tool for identifying the occurrence of fire incidents. The data-driven insights could empower policymakers to make informed decisions regarding preventive measures and firefighting strategies.

In Figure **??**, it can be observed that significantly higher fire counts are found in certain locations. The presence of two strips with high frequencies of fires are highly unusual. The vertical trend corresponds to the location of BR-230 (Trans-Amazonian Highway) passing through the city of Apuí, State of Amazonas, where a high frequency of fire occurrence is observed. The horizontal trend corresponds to BR-163 (Brazil highway) passing through Três Pinheiros in Novo Progresso, State of Pará. The western coastal area with a high frequency of fire occurrence corresponds to regions in close proximity to the cities of Vista Alegre do Abunã and Rio Branco. Research has indicated that 95 % of active fires and the most intense ones (FRP ¿ 500 megawatts) occurred at the edges in forests.

From the same figure, it can be observed that August and September are the riskiest months in terms of fire hazard, whereas little risk is posed from November to July. The follow-up question naturally arises: How does FY22 compare to previous years? Is it valid to claim that August and September constitute the fire hazard season?

In Figure **??**, the data shows a higher number of fire occurrences in the months of August to October compared to the rest of the year, indicating a greater number of fire hazards during these months.

```
# Obtain the Brazil map data
brazil_map <- map_data("world", region = "Brazil")

# Create the heatmap of fire occurrences
space_heatmap <- ggplot(confident_fire_fy22, aes(x = longitude, y = latitude)) +
  geom_polygon(data = brazil_map, aes(x = long, y = lat, group = group),
               fill = "#bdbdbd") +
  geom_bin2d(bins = 300) +
  scale_fill_gradient(low = "#fee6ce", high = "#d7301f") +
  coord_fixed(ratio = 1) +
  theme_minimal()+
  theme(axis.text = element_text(size = 9))

interactive_plot <- ggplotly(space_heatmap)

time_heatmap <- ggplot(confident_fire_months_fy22,
                       aes(x = abb_month, y = as.character(2022), fill = count)) +
  geom_tile(width = 0.9, height = 0.5) +  # Create the heatmap tiles
  scale_fill_gradient(low = "#fff7ec", high = "#d7301f") +
  labs(x = " ", y = " ", name = "count") +
  theme_minimal() +
  theme(axis.text = element_text(size = 9))

spacetime_fy22 <- grid.arrange(space_heatmap, time_heatmap, nrow = 2,
                               heights = c(2,0.5))

print(spacetime_fy22)

## TableGrob (2 x 1) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
```

```
## 2 2 (2-2,1-1) arrange gtable[layout]
```

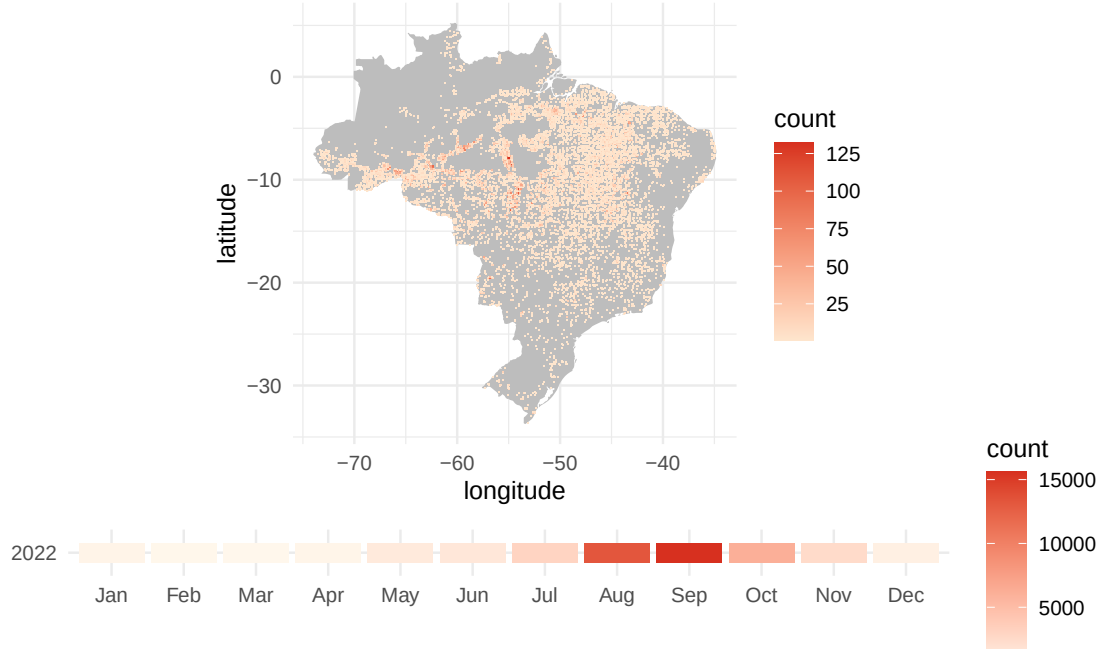


Figure 14: Frequency of Fire by Space and Time, FY22

```
heatmap_plot <- ggplot(pivot_table,
                       aes(x = factor(abb_month, levels = custom_order),
                           y = as.character(year), fill = count)) +
  geom_tile() +
  scale_fill_gradient(low = "#fff7ec", high = "#d7301f") +
  labs(x = " ", y = " ") +
  theme_minimal() +
  theme(axis.text = element_text(size = 9))

print(heatmap_plot)
```

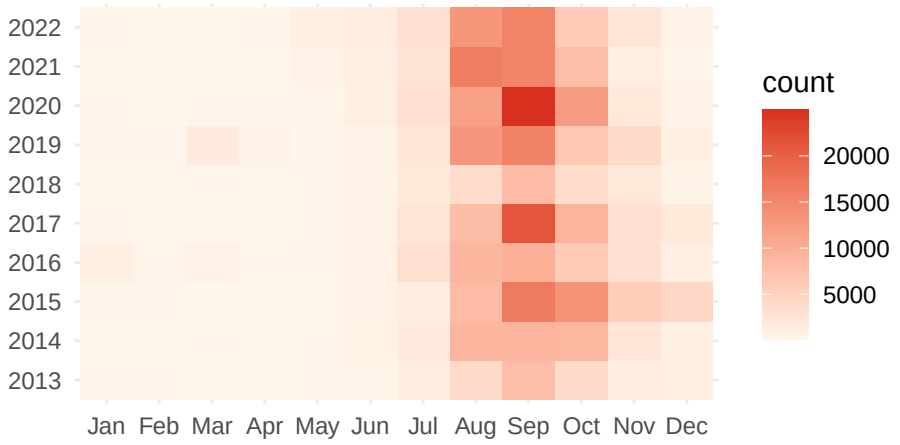


Figure 15: Frequency of Fire Occurrences, FY13-22

3.2.2 Heatmaps, correlation matrix and AIC score

The foundation of a heatmap is a data matrix M, where each entry in this matrix represents an observation:

$$M = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1j} \\ M_{21} & M_{22} & \dots & M_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ M_{i1} & M_{i2} & \dots & M_{ij} \end{bmatrix}.$$

Therefore, the first step to create a heatmap is to organize the data into columns and rows. In Figure **??**, the structured data is displayed as a grid of coloured cells, where the colour intensity corresponds to the underlying frequency.

Heatmaps serve as powerful tools for visualizing relationships between covariables within a model. An example of the necessity to analyze a matrix of correlations between variables is found in regression models. In the real world, variables are often correlated, and completely independent relationships are seldom encountered. Therefore, the analysis of pairwise correlations becomes essential. Significantly impacted by highly correlated variables, the regression model requires the selection of one variable from the correlated set. The selection is based on the identification of a regression model with the lowest Akaike Information Criterion (AIC) score among these variables:

$$AIC = -2l(\hat{\theta}) + 2\dim(\theta),$$

where $l(\hat{\theta})$ is the log-likelihood function, which is used to find the Maximum Likelihood Estimator (MLE) of a distribution.

The AIC measures the extent to which the linear model fits the dataset. To obtain the best model,

minimise the AIC score. In other words, the objective is to have the trend explained by the regression model, while avoiding overfitting that captures the noise in the dataset, ultimately leading to inaccurate predictions.

3.2.3 Treemaps

Treemaps are a visualisation method specifically designed for hierarchical data structures. They represent data as nested rectangles, where each rectangle represents a part of the whole. Treemaps offer a visually appealing and efficient way to convey the hierarchical composition of data. The size and color of each rectangle can be used to encode additional information.

3.2.4 Use Cases for Treemaps

Treemaps are highly effective when dealing with hierarchical data. Some common use cases include:

- **Disk Space Visualisation**: Treemaps can be employed to visualise disk space usage, where the outermost rectangle represents the entire disk, and inner rectangles represent folders and files. The size of each rectangle reflects the space they occupy.
- **Market Share Analysis**: In business, treemaps are useful for visualizing market share data. The top-level rectangle represents the total market, and inner rectangles represent individual segments, brands, or products. The size and color of each segment can represent its share and performance.

XXX

3.3 Line Charts and Time Series Visualisation

In this chapter, we are going to investigate the intricacies of the line chart and its most common application: time series. First, a line chart is a statistical representation that uses a Cartesian coordinate system, where each point on the chart corresponds to a pair of coordinates (x, y), to depict changes in numerical values over continuous time intervals or ordered categories. The x-axis typically represents these intervals or categories, while the y-axis conveys quantified data. Hence, data points are represented by coordinates $\{(x_i, y_i)\}_{1 \leq i \leq n}$, with n being the total number of data points. In a line chart, consecutive data points are typically connected by straight lines. The line segment between two points (x_i, y_i) and (x_{i+1}, y_{i+1}) can be described by the equation of a line in the slope-intercept form: $y = mx + b$, where m is the slope and b is the y-intercept.

A line chart is a series of linear interpolations between pairs of data points. These interpolations assume that the change between two points is uniform or linear, which is a significant simplification of real-world data. This linear approach is mathematically represented as:

$$y = y_i + \frac{(y_{i+1} - y_i)}{(x_{i+1} - x_i)} \cdot (x - x_i) \quad for \quad x_i \leq x \leq x_{i+1}$$

This equation highlights that for any point x between x_i and x_{i+1}, the corresponding value of y on the line chart is determined by a linear relation. This method effectively 'fills the gaps' between actual observed data points and provides a continuous view of the data.

3.3.1 Suitability for Displaying Trends Over Time

Line charts effectively visualise data trends over time. By plotting data at intervals like days or years, they highlight trends and patterns. Multiple lines on one chart enable easy data comparison, such as contrasting sales of two products. They aid in recognizing seasonal changes, cyclic events, and unexpected shifts, making them invaluable for forecasting. Due to their simplicity, they're accessible to those with minimal data analysis background. Line charts are a prime choice for time series visualisation.

Time series visualisation is essential in data analysis, showcasing time-ordered data. It reveals long-term trends, helping analysts discern patterns for future planning. It's crucial for spotting seasonality in datasets. This method also identifies anomalies, suggesting areas needing investigation. Predictive modeling, based on historical patterns, becomes feasible, fostering proactive choices. Overlaying multiple data series offers richer comparative analysis. Overall, time series visualisation provides quick insights into chronological data, driving informed decisions by highlighting trends, seasonal changes, and outliers.

While line charts are excellent for displaying trends over time, they have limitations. They may not be suitable for showing individual data distributions or for data where there's no logical order. eg. too many points, too many lines, too many zeros.

3.3.2 Showcase real-world examples of time series visualisations

Here, we are going to investigate 'exchange rate dataset' and plot all daily and 21-day moving average exchange rates in one figure.

```r
# Plot daily and 21-day moving average exchange rates of CNY, CAN,
# EUR, HKD, USD to GBP

# First plot
p1 <- ggplot(plot_dt, aes(x=Date, y=Rate, color=Currency)) + geom_line() +
  labs(title="Daily exchange rates", y="Exchange Rate to GBP",
       x="Date", color="Currency")+
  theme_minimal() + theme(legend.position="none")

# Second plot
p2 <- ggplot(plot_data, aes(x=Date, y=Rate, color=Currency)) + geom_line() +
  labs(title="21-Day moving average \n exchange rates",
       y="Exchange Rate to GBP", x="Date", color="Currency")+
  theme_minimal() + theme(legend.position="none")

# Extract the legend
p2_legend <- cowplot::get_legend(p2 + theme(legend.position="bottom"))
# Combine the plots with adjusted widths using cowplot
combined_plot <- cowplot::plot_grid(p1, p2, labels = c("1", "2"),
                                    rel_widths = c(1, 1), nrow=1)
# Combine the plots and the legend
cowplot::plot_grid(combined_plot, p2_legend, ncol=1, rel_heights = c(1, .1))
```

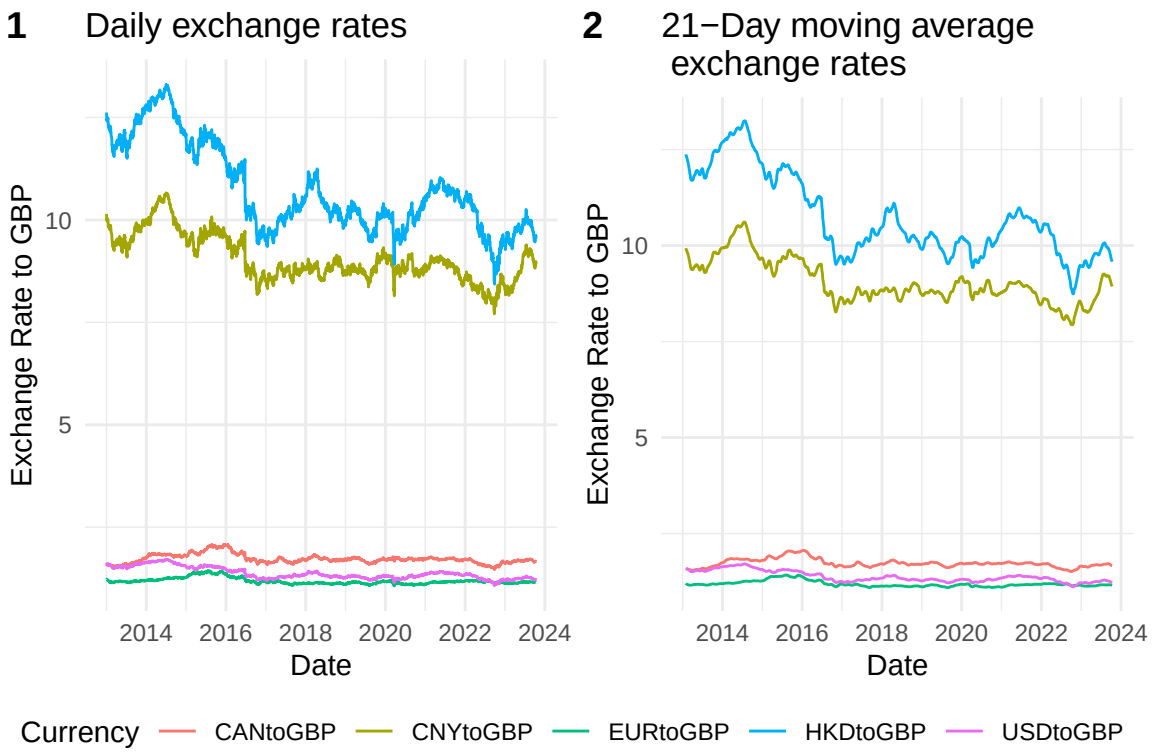


Figure 16: Daily and 21-day moving average exchange rates of CNY, CAN, EUR, HKD, USD to GBP

From the first plot of Figure **??**, we can see the the daily CNY, CAN, EUR, HKD, USD versus GBP exchange in the same plot, which provide us an overview of the trend and comparison. Then we employed a 21-day moving average (MA21) to elucidate long-term trends while mitigating short-term

fluctuations. This is mathematically represented as

$$MA_{21}(t) = \frac{1}{21} \sum_{k=t-20}^{t} y_k$$

, where y_k denotes the exchange rate on day k. This method effectively filters out daily noise, allowing a clearer view of overarching trends in currency movements against the GBP. The overlay of these moving averages on the daily exchange rates in our visualizations provides both a clear comparative and a quantitative perspective.

Decomposition of time series into trend, seasonal, and random:

One of the primary advantages of time series visualisation is the ease with which it allows analysts to identify long-term upward or downward trends in data and patterns that repeat over specific intervals. By decomposing the time series, it would be easy to see those features.

Time series data, Y_t, can often be described as a combination of several distinct components:

- **Trend (T_t):** The underlying progression in the series.
- **Seasonal (S_t):** Periodic fluctuations due to seasonal factor.
- **Residual (R_t):** The irregular or error component.

The decomposition of a time series can be described in two main models:
Additive Model: In the additive model, the components are added together:

$$Y_t = T_t + S_t + R_t$$

Multiplicative Model: In the multiplicative model, the components are multiplied together:

$$Y_t = T_t \times S_t \times R_t$$

In practice, the choice between the additive and multiplicative models often depends on the nature of the time series. If the magnitude of the seasonal fluctuations or the variation around the trend does not vary with the level of the time series, then an additive model is appropriate. If the magnitude of the seasonal fluctuations or the variation around the trend increases or decreases as the time series level changes, then a multiplicative model may be more suitable.

```r
# Plot decomposition of addictive time series model
decomposed_ts <- stats::decompose(ts_data$CNYtoGBP)
plot(decomposed_ts, xlab="Date")
```

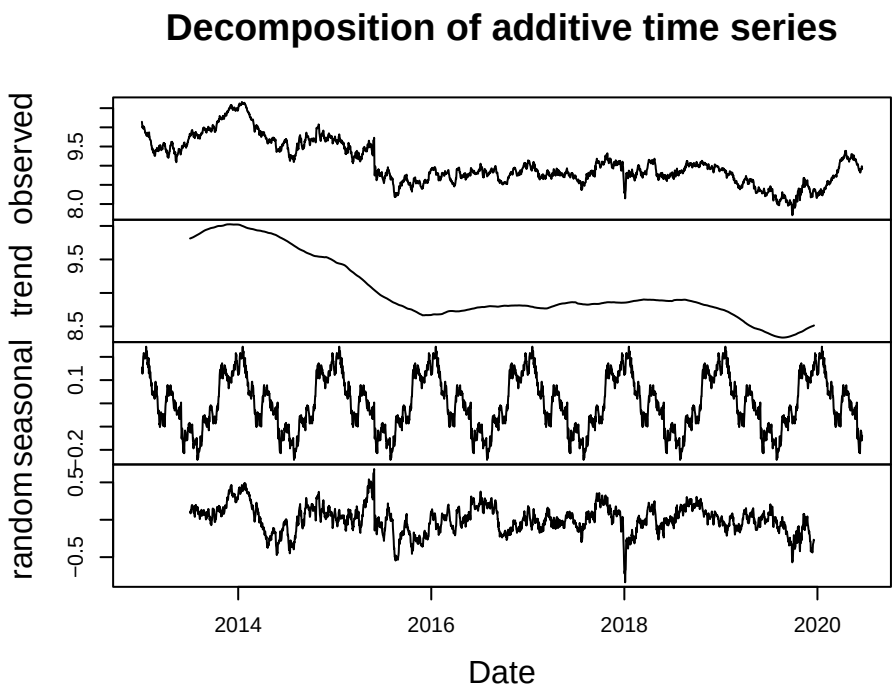


Figure 17: Decomposition of addictive time series model of CNY to GBP exchange rates

In the decomposition of the CNY to GBP exchange rate series using an additive model, we disentangle the data into its fundamental components: trend, seasonality, and residual noise. This additive model, represented mathematically as $Y_t = T_t + S_t + R_t$, allows us to scrutinize each element in isolation. As illustrated in Figure **??**, the trend component T_t reveals a gradual decrease in the CNY to GBP exchange rate over time, signifying a long-term depreciation of CNY against GBP. This trend is pivotal for understanding the broader economic relationship between these currencies.

Moreover, the seasonal component S_t of the decomposition highlights cyclical fluctuations, indicative of recurrent patterns within the year. These could be attributed to seasonal economic activities, policy changes, or other cyclical factors influencing the currency market. The clear demarcation of these cyclical trends in the seasonal component helps in isolating such effects from the overarching trend.

Lastly, the residual component R_t encompasses the random, unexplained variations after accounting for the trend and seasonal factors. Analyzing these residuals is crucial for understanding the unpredictability in the exchange rate and can be pivotal in risk management and forecasting.

Autocorrelation Analysis of CNY to GBP Exchange Rate

Autocorrelation, also referred to as serial correlation, is a crucial concept in time series analysis. It describes the correlation of a time series with its own past and future values. The autocorrelation

function (ACF) measures the linear predictability of the series at time t with its values at a previous time $t - k$.

The autocorrelation function at lag k is defined as:

$$R(k) = \frac{\sum_{t=1}^{n-k}(y_t - \bar{y})(y_{t+k} - \bar{y})}{\sum_{t=1}^{n}(y_t - \bar{y})^2}$$

where y_t is the value of the series at time t, $\bar{y}$ is the mean of the series, and n is the total number of observations.

The value of $R(k)$ lies between -1 and +1. A value close to +1 indicates a strong positive correlation, while a value close to -1 indicates a strong negative correlation. A value near 0 suggests little to no linear correlation. A slow decay in the ACF plot indicates a strong relationship between past and present values, while spikes at specific lags may suggest seasonality. Autocorrelations outside the 95% confidence interval are considered statistically significant.

Next, we'll visualize the ACF for the CNY to GBP exchange rate to understand its time-dependent structure better.

```
# Plot the Autocorrelation Function (ACF)
acf(MyData$CNYtoGBP, main="ACF of CNY to GBP Exchange Rate")
```

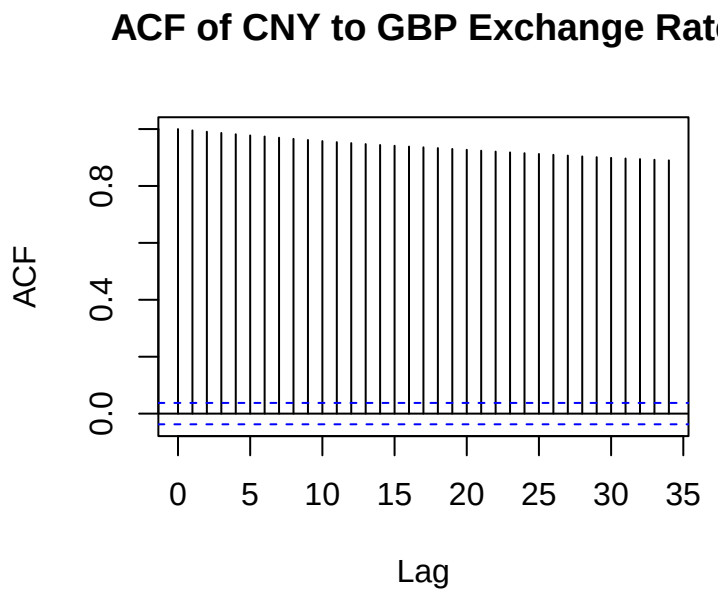


Figure 18: Autocorrelation Function (ACF) of CNY to GBP Exchange Rate

The Autocorrelation Function (ACF) plot for the CNY to GBP exchange rate series reveals a compelling feature: the ACF starts near 1 and decreases gradually. This pattern suggests a strong persistence in the time series, indicating that past values have a significant influence on future values.

In time series analysis, such a slow decay in the ACF is indicative of a non-stationary series, where the mean, variance, and autocorrelation structure do not remain constant over time.

This persistent autocorrelation suggests that short-term movements in the CNY to GBP exchange rate are heavily influenced by its recent history. Such a characteristic is crucial for forecasting models, as it implies that recent historical data can be a powerful predictor of near-future trends. Models like ARIMA (Autoregressive Integrated Moving Average), which are well-suited for data with high autocorrelation, may be particularly effective in this context.

3.4 Network Graphs

Definition and Utility: Network graphs, often referred to as graphs or networks, are a powerful data visualisation method used to depict relationships between entities. These entities, known as nodes, are interconnected by edges or links, which represent relationships, connections, or interactions. Network graphs find extensive utility in various fields, such as social network analysis, transportation systems, and even biological networks like protein-protein interactions. They excel at revealing complex dependencies and structures, making them a critical tool for understanding relational data.

3.4.1 The Mathematics behind Network Graphs:

Constructing network graphs involves several mathematical intricacies. Here we present just a few of the many concepts that play a role in the creation of such graphs:

1. **Nodes and Edges**: Mathematically, a network graph, G, is defined as $G = (V, E)$, where V represents the set of nodes and E represents the set of edges connecting these nodes.
2. **Node Degree**: The degree of a node is the number of edges connected to it. In a directed graph, nodes can have both in-degrees and out-degrees.
3. **Centrality Measures**: Centrality metrics like degree centrality, betweenness centrality, and closeness centrality provide insights into the relative importance or influence of nodes within a network.
4. **Graph Metrics**: Graph theory concepts like shortest paths, connected components, and clustering coefficients are used to analyze the network's structure.

Formulas used in Network Graphs:

1. **Degree of a Node (Undirected Graph)**:

$$Degree(v) = \sum_{w \in V} A(v, w)$$

where $A(v, w)$ is the adjacency matrix element, indicating whether there is a connection between nodes v and w.

2. **Degree of a Node (Directed Graph)**:

$$In - Degree(v) = \sum_{w \in V} A(w, v)$$

$$Out - Degree(v) = \sum_{w \in V} A(v, w)$$

3. **Betweenness Centrality (for unweighted graphs)**:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the number of shortest paths from node s to t, and $\sigma_{st}(v)$ is the number of those paths passing through node v.

3.4.2 Network Graphs in Practice

```
# Plot the graph
#plot(lesmis_graph, layout = layout, vertex.label.cex = 0.7, main = "Character Interactions in Les Misérables")
```

3.5 Sankey Diagrams

xxx

3.6 Geographic Maps and Spatial Data Visualisation

A **geographical maps** is a visual representation of an area—a symbolic depiction highlighting relationships between elements of that space, such as objects, regions, or themes. Maps have been used for centuries to navigate and explore the world, and they play a crucial role in understanding our environment, both locally and globally. These maps serve as canvases on which spatial data is painted, allowing for a visual comprehension of information that might otherwise remain abstract.

Before moving on to the spatial data visualisation. It is essential to understand how we map the the earth on a plane. The Earth, a three-dimensional spheroid, can be transformed on a plane through map projections. Each projection offers a different way to "flatten" the Earth, and as a result, each has its strengths and distortions. For instance, the Mercator projection preserves angles but distorts areas as you move towards the poles. Beyond projections, coordinate systems, like the commonly used latitude and longitude, provide a standardized way to pinpoint any location on Earth.

- **Latitude** measures the angle between a point on the Earth's surface and the equator, moving north or south. And latitude values range from -90 (South Pole) to +90 (North Pole). The equator, which divides the Earth into the Northern and Southern Hemispheres, is at 0 latitude.
- **Longitude** measures the angle between a point on the Earth's surface and the prime meridian, moving east or west. And longitude values range from -180 to +180. The prime meridian, which is at 0 longitude, runs from the North Pole through Greenwich, England, to the South Pole. It divides the Earth into the Eastern and Western Hemispheres.

Together, lines of longitude and latitude create a grid system over the Earth's surface. By providing both a latitude and longitude value, one can specify an exact location on the Earth's surface. For example, the coordinates (0 N, 0 E) would indicate the intersection of the equator and the prime meridian, located in the Gulf of Guinea off the west coast of Africa.

Spatial data visualisation

Spatial data visualisation are powerful tools that transform raw, often complex datasets into visual representations, revealing patterns, relationships, and insights rooted in location. At their core, maps provide a spatial context, allowing us to see the world's intricate web of interconnectedness. Today, with the surge in big data and advanced visualisation tools, spatial data visualisation is not just about presenting information but also about telling compelling stories, guiding decision-making, and predicting future trends based on geographical patterns.
Here, we are going to construct a geographical map of the Greater London and show each ward.

```
# Plot London map by ward
ggplot(data = london_boroughs) + geom_sf(fill = "lightblue", color = "black") +
  theme_minimal() + labs(y="Latitude", x="Longitude")
```

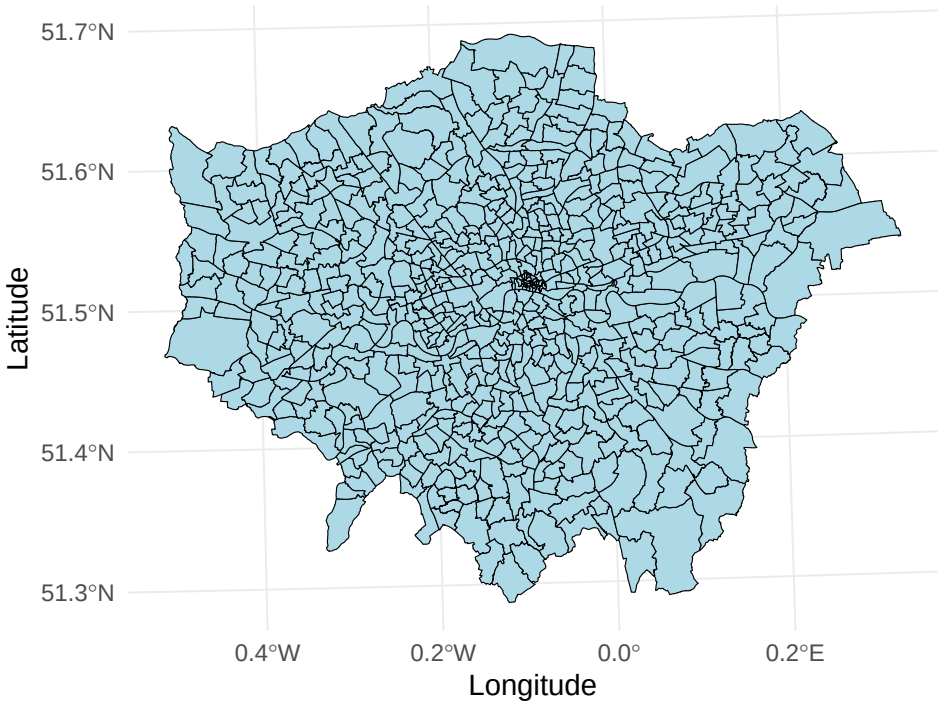


Figure 19: London Wards Map

From Figure **??**, we are introduced to a detailed layout of all wards within Greater London, pinpointed by their geographical coordinates. While this map provides a clear depiction of location and boundaries, it offers limited insight into the dynamics of crime distribution.

Hence, we will use a more informative visualization: a choropleth map showcasing crime rates by borough. Here, crime rates are calculated by dividing the total crime count in each borough

by its respective population as of 2020. This per capita approach normalizes the data, ensuring comparability across boroughs with different population sizes.

```
# Plot the Crime rate in London by boroughs
ggplot(data=aggregated_data) + geom_sf(aes(fill=Crime_rate)) +
  geom_sf_text(aes(label = DISTRICT, geometry = centroid),
               size = 2.5, check_overlap = TRUE) +
  scale_fill_gradient(low="Green", high="red") + theme_minimal() +
  labs(fill="Crime rate")
```

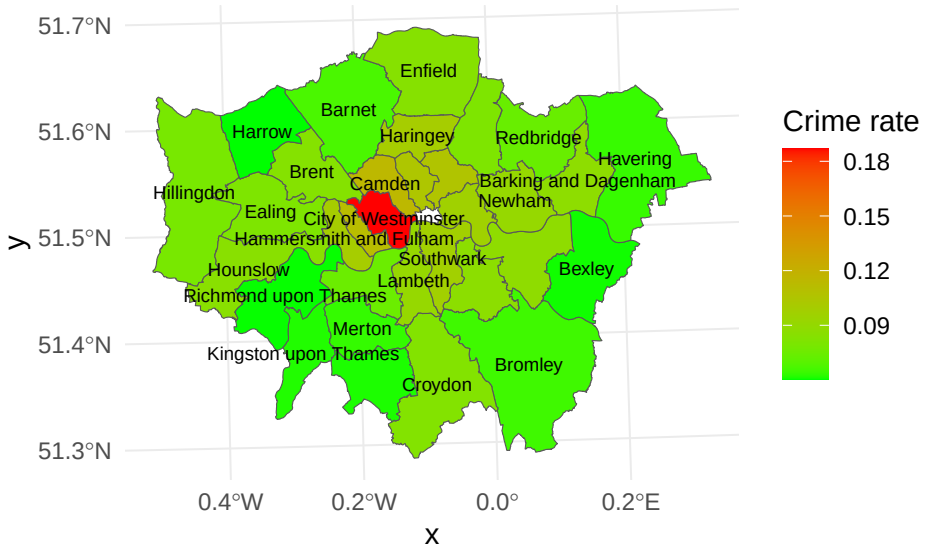


Figure 20: Crime rate by boroughs in London in 2020

From Figure **??**, Westminster stands out, labeled in red, indicating a higher crime rate. This visual marker highlights Westminster as an area of particular concern regarding criminal activity. This could be attributed to factors like its status as a central, densely populated area with significant tourist traffic and commercial activity, which often correlate with higher crime rates.

In contrast, the map reveals a trend where rural or less densely populated areas tend to be safer, as evidenced by their lighter coloration. These areas typically experience lower crime rates, possibly due to factors such as smaller populations, less anonymity for potential offenders, and different

socio-economic dynamics compared to urban centers.

The choropleth map of London's crime rates elucidates key spatial patterns in crime distribution. It highlights the variance in crime rates from urban centers to rural areas, accentuating the higher crime rates in central, densely populated areas like Westminster, depicted in red, compared to the lighter shades marking safer, rural boroughs. This visual tool facilitates an understanding of how urbanization, socio-economic status, and population density impact crime rates across the boroughs.

3.7 3D and Interactive Visualisations

ggplot2 is one of the most popular data visualisation libraries in R, but it is primarily designed for 2D data visualisation. Directly creating 3D views with ggplot2 can be challenging.

R provides several packages for 3D visualization, such as rgl, plot3D, rayshader, and others, which are specifically designed for three-dimensional data. These packages offer the capability to create 3D scatter plots, surface plots, heat maps, contour maps, and more.

rgl: This is one of the most popular R packages for creating interactive 3D charts. It supports various types of 3D graphics including points, lines, and surfaces, and allows users to interactively rotate, zoom, and pan the view.
scatterplotThis package provides a function to create 3D scatter plots. It does not support interactive manipulation, but the generated graphics are well-suited for display in static reports.

3D data visualisation is an approach that employs three-dimensional graphics to represent complex data structures, allowing for an immersive exploration of information. Unlike traditional 2D visualisations (like bar graphs or line charts), 3D visualisations can convey an additional dimension of data, making them particularly valuable in specific contexts.

Our first example will be a scatter plot. We can use scatterplot3d package to help us for data visualisation.

```
# Generate colors based on the Volume variable
colors <- colorRampPalette(c("blue", "red"))(length(unique(trees$Volume)))
color_assign <- colors[as.numeric(as.factor(trees$Volume))]
# Create 3d scatter plot with colors
scatterplot3d(trees$Girth, trees$Height, trees$Volume,
              color=color_assign,
              main="3D Scatterplot of trees data",
              xlab="Girth (inches)",
              ylab="Height (ft)",
              zlab="Volume (cubic ft)")
```

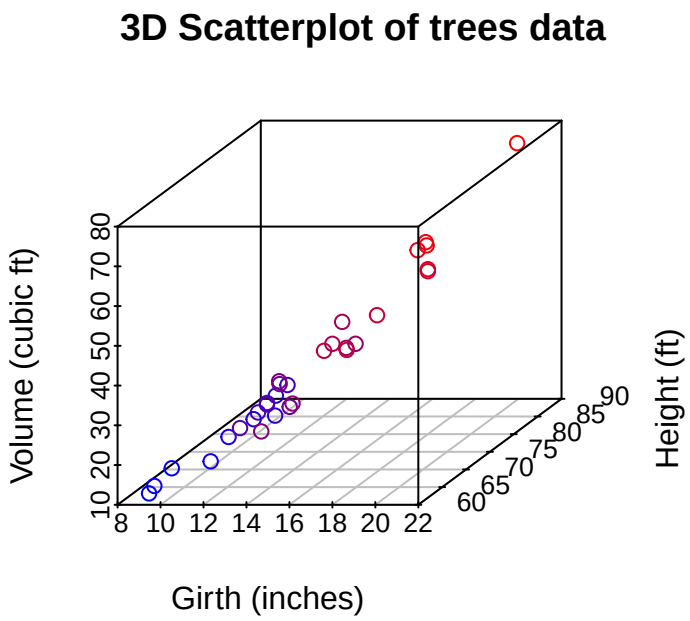


Figure 21: 3d scatter plot

4 Trees Dataset in R

4.1 Advanced Visualisation Techniques

xxx

5 Practical Implementations

XXX

6 Case Studies

6.1 Market Analysis Dashboards

XXX

6.2 Healthcare Data Visualisation

XXX

7 State-of-the-Art Approaches

XXX

8 Conclusion

XXX

References

[1] Earth Science Data Systems. (2023) *FIRMS Frequently Asked Questions — Earthdata*. Available at: https://www.earthdata.nasa.gov/faq/firms-faqed-confidence [Accessed 12 Nov. 2023].

[2] Worton, B. (2023) *Generalised Regression Models, Semester 1, 2023–2024*. University of Edinburgh. Chp. 3.1.

[3] Wikipedia Contributors. (2019) *Multicollinearity*. Available at: https://en.wikipedia.org/wiki/Multicollinearity [Accessed 12 Nov. 2023].

Chapter 1-3

A Using `step()` function to select Linear Regression Models

```
Start:  AIC=70.9
mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb

       Df Sum of Sq    RSS    AIC
- cyl   1    0.0799 147.57 68.915
- vs    1    0.1601 147.66 68.932
- carb  1    0.4067 147.90 68.986
- gear  1    1.3531 148.85 69.190
- drat  1    1.6270 149.12 69.249
- disp  1    3.9167 151.41 69.736
- hp    1    6.8399 154.33 70.348
- qsec  1    8.8641 156.36 70.765
<none>              147.49 70.898
- am    1   10.5467 158.04 71.108
- wt    1   27.0144 174.51 74.280

Step:  AIC=68.92
mpg ~ disp + hp + drat + wt + qsec + vs + am + gear + carb

       Df Sum of Sq    RSS    AIC
- vs    1    0.2685 147.84 66.973
- carb  1    0.5201 148.09 67.028
- gear  1    1.8211 149.40 67.308
- drat  1    1.9826 149.56 67.342
```

```
- disp  1    3.9009 151.47 67.750
- hp    1    7.3632 154.94 68.473
<none>              147.57 68.915
- qsec  1   10.0933 157.67 69.032
- am    1   11.8359 159.41 69.384
- wt    1   27.0280 174.60 72.297

Step:  AIC=66.97
mpg ~ disp + hp + drat + wt + qsec + am + gear + carb

       Df Sum of Sq    RSS    AIC
- carb  1    0.6855 148.53 65.121
- gear  1    2.1437 149.99 65.434
- drat  1    2.2139 150.06 65.449
- disp  1    3.6467 151.49 65.753
- hp    1    7.1060 154.95 66.475
<none>              147.84 66.973
- am    1   11.5694 159.41 67.384
- qsec  1   15.6830 163.53 68.200
- wt    1   27.3799 175.22 70.410

Step:  AIC=65.12
mpg ~ disp + hp + drat + wt + qsec + am + gear

       Df Sum of Sq    RSS    AIC
- gear  1     1.565 150.09 63.457
- drat  1     1.932 150.46 63.535
<none>              148.53 65.121
- disp  1    10.110 158.64 65.229
- am    1    12.323 160.85 65.672
- hp    1    14.826 163.35 66.166
- qsec  1    26.408 174.94 68.358
- wt    1    69.127 217.66 75.350

Step:  AIC=63.46
mpg ~ disp + hp + drat + wt + qsec + am

       Df Sum of Sq    RSS    AIC
- drat  1     3.345 153.44 62.162
- disp  1     8.545 158.64 63.229
<none>              150.09 63.457
- hp    1    13.285 163.38 64.171
- am    1    20.036 170.13 65.466
- qsec  1    25.574 175.67 66.491
- wt    1    67.572 217.66 73.351

Step:  AIC=62.16
mpg ~ disp + hp + wt + qsec + am
```

```
       Df Sum of Sq    RSS    AIC
- disp  1     6.629 160.07 61.515
<none>              153.44 62.162
- hp    1    12.572 166.01 62.682
- qsec  1    26.470 179.91 65.255
- am    1    32.198 185.63 66.258
- wt    1    69.043 222.48 72.051

Step:  AIC=61.52
mpg ~ hp + wt + qsec + am

       Df Sum of Sq    RSS    AIC
- hp    1     9.219 169.29 61.307
<none>              160.07 61.515
- qsec  1    20.225 180.29 63.323
- am    1    25.993 186.06 64.331
- wt    1    78.494 238.56 72.284

Step:  AIC=61.31
mpg ~ wt + qsec + am

       Df Sum of Sq    RSS    AIC
<none>              169.29 61.307
- am    1    26.178 195.46 63.908
- qsec  1   109.034 278.32 75.217
- wt    1   183.347 352.63 82.790

Call:
lm(formula = mpg ~ wt + qsec + am, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4811 -1.5555 -0.7257  1.4110  4.6610

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.6178     6.9596   1.382 0.177915
wt           -3.9165     0.7112  -5.507 6.95e-06 ***
qsec          1.2259     0.2887   4.247 0.000216 ***
am            2.9358     1.4109   2.081 0.046716 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.459 on 28 degrees of freedom
Multiple R-squared:  0.8497,Adjusted R-squared:  0.8336
F-statistic: 52.75 on 3 and 28 DF,  p-value: 1.21e-11
```