Data Visualisation: Theory and Practice

Yujie Chu, Pia Fullaondo, Qinqing Li, Jacko Zhou

November 16, 2023

Contents

1 Introduction

1.1 Historical Background and Misuses of Data Visualisation

Effective data visualisation is crucial for both exploratory data analysis and data communication. visualisations provide insight into data attributes, assisting data scientists in constructing models and addressing questions of interest. The field of data visualisation is covered in numerous method-specific publications, with one notable example being the Journal of Computational and Graphical Statistics. This journal publishes research on the latest techniques in computational and graphical methods in statistics, encompassing data analysis and numerical graphical displays. Spatial Statistics publishes articles on the theory and application of spatial and spatio-temporal statistics, and The R Journal publishes research articles in statistical computing that are of interest to R users.

Many exceptional design-oriented data visualisations are showcased on Information is Beautiful by David McCandless. Meanwhile, FlowingData is a blog on data visualisation and statistics by statistician Nathan Yau.

Below, an overview of the historical background is presented, accompanied by case examples of the misuses of data visualisation.

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 1.

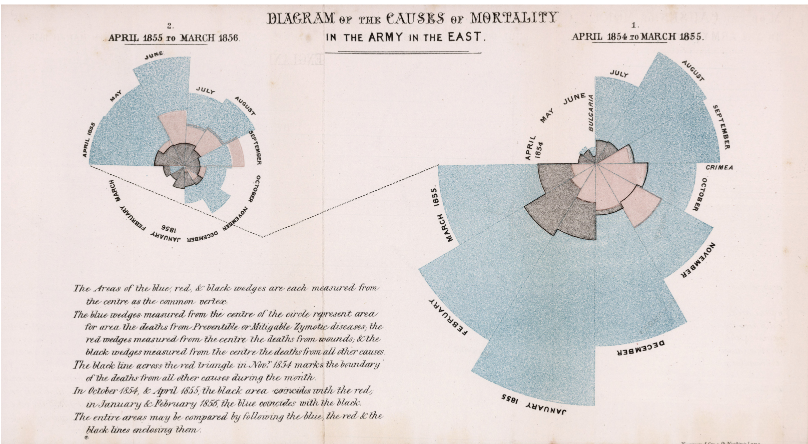


Figure 1: "Diagram of the causes of mortality in the army in the East", in 1858 by Florence Nightingale[9]

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 datavisualisations 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 2. This seemingly simple yet remarkably powerful chart visually illustrated the Earth's warming trend since 1850.

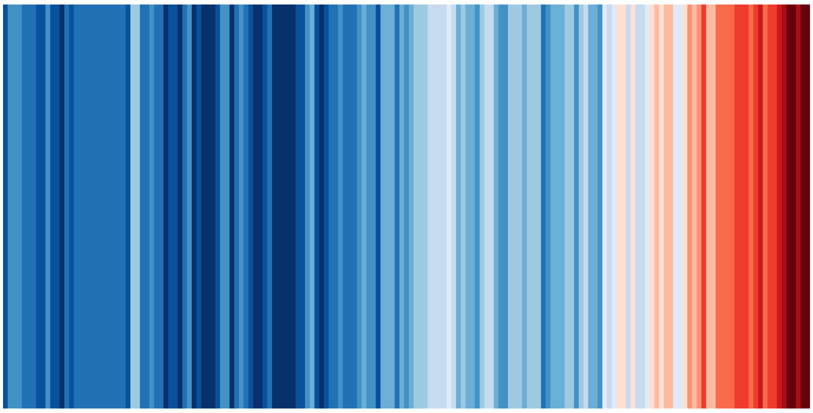


Figure 2: "Latest global stripes (1850-2020)", by Edward Hawkins[5]

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 datavisualisation conceals trends rather than revealing them. Figure 3 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[1].

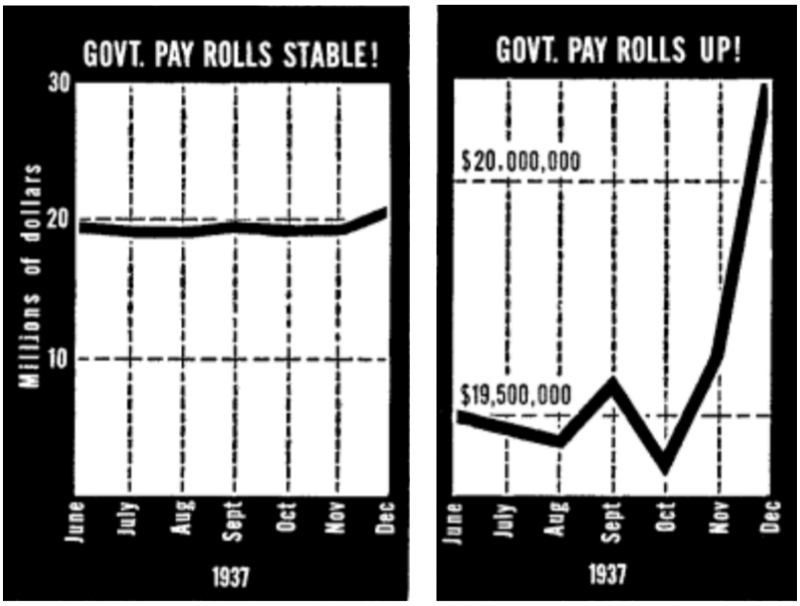


Figure 3: Inappropriate use of data visualisation[3]

Misuses of Data Visualisation - Case Example 2

Data visualisation 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 visualisation (as shown in Figure 4) 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[8].

Regrettably, the Kallikak family tree became a central element in the eugenics movement for decades afterward. Figure 4 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 [7].

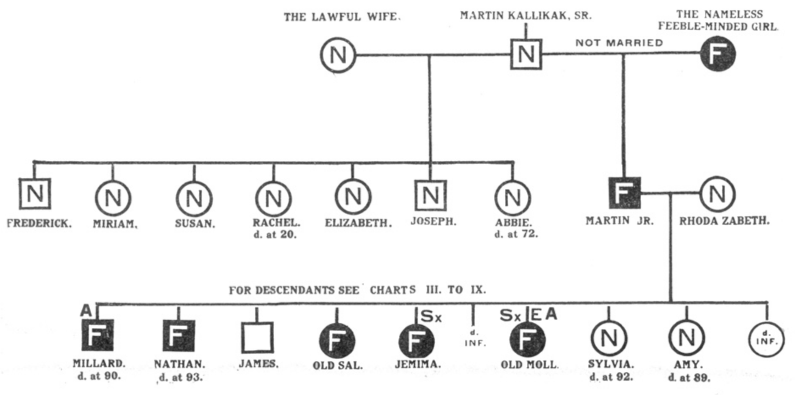


Figure 4: The Kallikak Family tree, in 1912 by Henry Goddard[2]

Case examples obtained from BBC Ideas[6].

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 visualisations 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 datavisualisation. 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.

Mtcars: 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 .

Tooth Growth: *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.

Iris: 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.

Annual Fire in Brazil: 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 $\geq 95\%$ [4].

Exchange Rate: 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.

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.

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 which will be studied in order: the 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 emphasises the importance of reducing mental effort needed to process information.

2.2 Data Types and Visualisation Techniques

To have a discussion about data representation, understanding the nature of the data is key. Data comes in various types, and selecting the appropriate visualisation technique is contingent upon recognising these distinctions. In this section, the data types are categorised and matched with their suitable visualisation techniques.

2.2.1 Categorisation of Data Types

Data types can be broadly categorised into four main types:

- **Nominal data**: represents categories or labels without any inherent order. Examples include colours, gender categories, and city names.
- **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**: 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**: includes ordered categories with equal intervals and a meaningful zero point. Examples are age, income, and weight.

2.2.2 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.

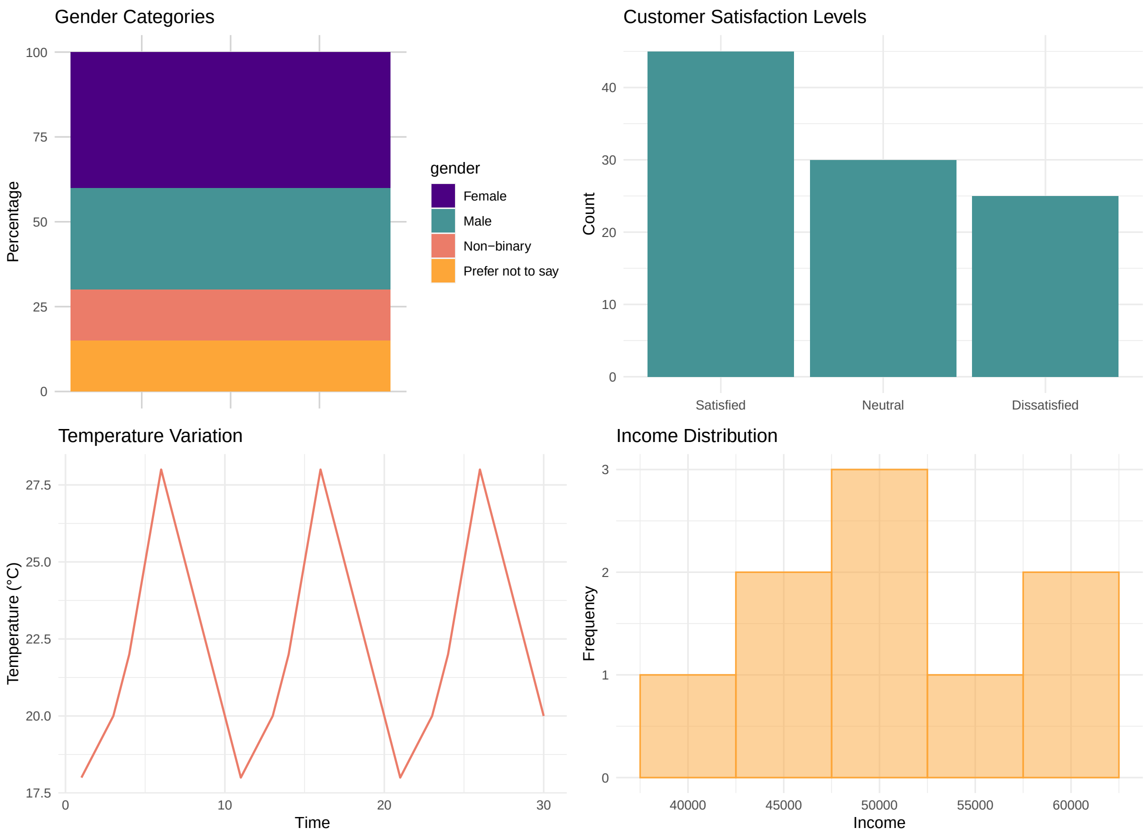


Figure 5: Different types of visualisation according to the data type used

2.3 Data Abstraction and Representation

The transformation of raw data into meaningful representations is a pivotal step in data representation. 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

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**: At the mid-level, 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.

These are represented in Figure 6. 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 visualisation 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.

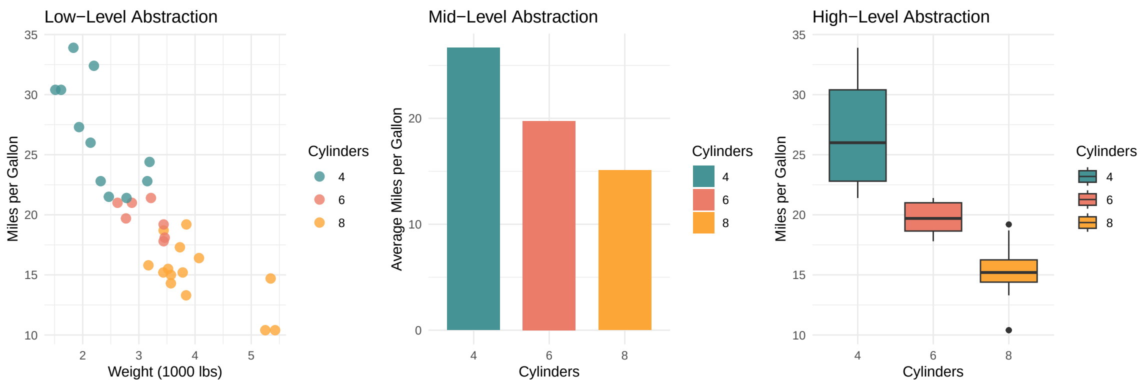


Figure 6: Mtcars dataset visualised on 3 different levels of abstraction

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, generalisation and specificity, and context versus precision. Abstraction increases clarity but may sacrifice crucial detailed information necessary for some analytical tasks. It offers a more generalised 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 Visual Perception and Cognition

In this first section, human visual perception is explored, along with the application of cognitive psychology principles in data visualisation.

Human Visual Perception: Decoding Visual Information
Human visual perception profoundly influences our understanding of the surrounding world. When applied to data visualisation, it shapes the way that 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.

2.4.1 The Gestalt Principles

Furthermore, the Gestalt Principles XXXCITEXXX play an important role in the realm of visual perception and design. 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.

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.

2.5 Colour Theory in Data Visualisation

In this section, 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 are explored.

The Importance of Colour in Conveying Information

Colour significantly enhances the impact and comprehension of data visualisations. It serves multiple purposes: distinguishing data points, emphasising trends, and offering contextual information. It is utilised to encode categorical data, differentiating between various groups with distinct colours, and to represent quantitative data by using colour 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 7. Careful selection of color schemes aligned with the intended message is essential — for instance, using warm colours like red and orange to indicate caution or warmth, and cool colours like blue and green to convey calmness or coldness. Additionally, attention should be paid to how colours interact when combined; certain combinations might create visual vibrations or impact text legibility.

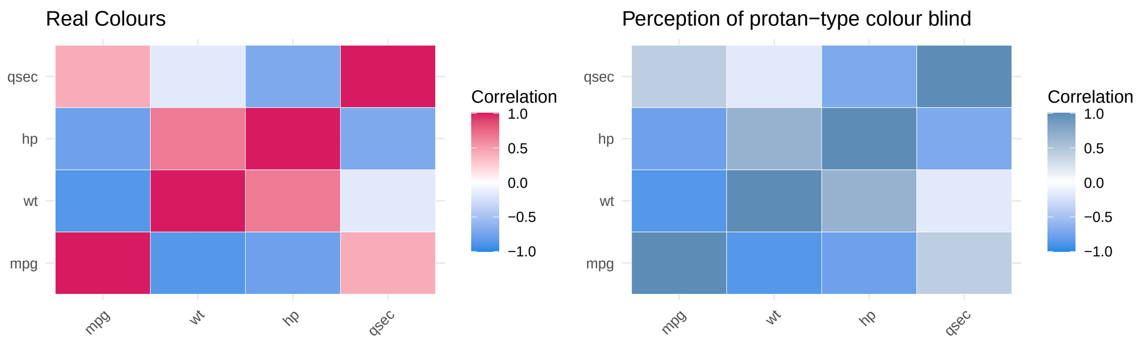


Figure 7: Colour perception of a heatmap for by a colour blind person

Avoiding Misleading Visualisations Due to Colour Choices

Misleading visualisations often stem from inappropriate or deceptive use of colour, requiring precautions to prevent such occurrences. First, maintaining consistency in colour usage throughout the visualisation is essential. Employing a uniform colour scheme for similar data categories or elements helps establish coherence and understanding. Furthermore, it's crucial to avoid colour choices that could distort or exaggerate the data. Overly intense or contrasting colours might mislead interpretations, emphasising the necessity for judicious colour selection.

Additionally, providing a clear and concise legend becomes imperative to explain the meaning of colours, especially when dealing with complex or unfamiliar colour schemes. A comprehensive legend helps viewers decipher the represented data accurately.

2.6 Cognitive Load and Visual Complexity

In data visualisation, achieving a balance between complexity and cognitive load is crucial. Cognitive load significantly influences how viewers engage with and comprehend presented data. Finding a balance is crucial to effectively convey information without overwhelming the viewer's cognitive capacity. 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.

2.6.1 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 ensures information is clear and easily digestible.

Furthermore, **employing interactive features** like tooltips and drill-down functionality assists in providing additional information when required, reducing the density of static visualisations. A final approach involves the use of **progressive disclosure**, presenting complex information gradually, beginning with an overview and allowing users to explore details as needed.

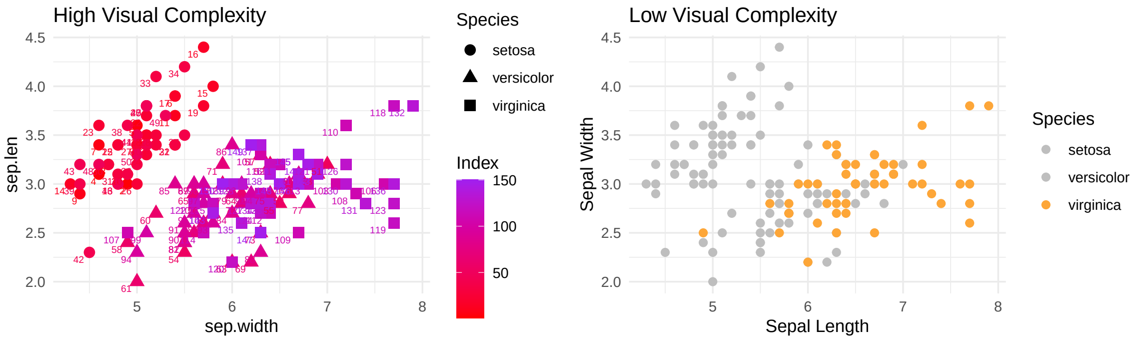


Figure 8: High vs low cognitive load demand through the reduction of visual complexity

2.6.2 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 prioritisation 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

This chapter explores a variety of powerful visualisation methods, from classic scatter plots and bar charts to advanced techniques like heatmaps and network graphs. Vivid examples will be used to demonstrate when and why each method is employed, and the theoretical and mathematical foundations that empower these visualisations to unveil insights hidden within the data will be delved into.

3.1 Bar Charts and Histograms

3.1.1 Bar Charts

The bar chart is a crucial tool in data presentation, arranging data into vertical or horizontal bars. Its advantages in data visualisation are manifold, particularly in showcasing data categories through frequency distribution. Bar charts excel in comparing classified data, especially when values are closely aligned. This superiority stems from human perception, as our visual acuity for height surpasses that of other visual elements like area or angle. The varying lengths of these bars directly correspond to the magnitude of the information they represent.

Bar charts represent a versatile tool for data visualisation, frequently employed to compare distinct categories. The **vertical bar chart**, commonly recognised, exhibits categories along the X-axis and their frequencies or counts along the Y-axis. **Horizontal bar charts**, rotated 90 degrees, prove beneficial for extended category names or numerous categories, displaying categories on the Y-axis and frequencies on the X-axis. **Multi-set or grouped bar charts** facilitate side-by-side comparisons of sub- groups within categories, available in both vertical and horizontal orientations. **Stacked bar charts** illustrate classes of values subdivided into sub-classes, often differentiated by colour, where each segment's size signifies its frequency or count, and the total bar length reflects the cumulative total.

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.

Bar Charts in Practice
The bar chart in Figure 9 depicts the relationship between tooth growth and different dosages of a vitamin, measured in milligrams per day. On the X-axis, three distinct levels of vitamin dosage are presented, while the Y-axis indicates the average tooth growth for each dosage. A key observation from the chart is a noticeable trend where increasing vitamin dosages correlate with increased tooth growth, suggesting that higher vitamin doses may enhance tooth growth.

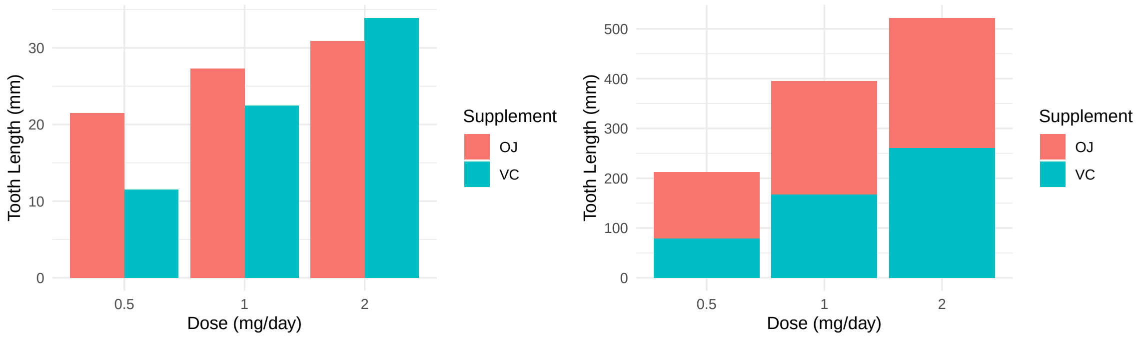


Figure 9: visualising Tooth Growth by Dosage: A Comparison of Grouped and Stacked Bar Chart Techniques

The bar chart on the left in Figure 9 illustrates the impact of varying vitamin dosages on tooth growth, further categorised by supplement type (supp). It shows three distinct dosage levels (0.5, 1, and 2 mg/day) on the X-axis and tooth growth length on the Y-axis, representing the average growth for both supplements at each dosage. The chart highlights that tooth growth, as indicated by the total height of each bar, varies not only with the dosage but also with the supplement type, suggesting a need for closer analysis to understand the relative effectiveness of each supplement at different dosages.

The structure of the second bar graph in Figure 9 is quite similar to that of the first one, with the main difference lying in the method of data representation. The stacked bar chart facilitates the understanding of the combined effects of the two supplements at each dosage level. However, compared to the normal bar chart, it becomes more challenging to differentiate the individual contributions of each supplement.

3.1.2 Histograms

Histograms, although visually similar to bar charts, convey different meanings. A histogram involves concepts of statistics. It requires data to be categorised 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.

Consider the following graph with created on ggplot2 with the use of to the `geom_histogram()` function and iris dataset.

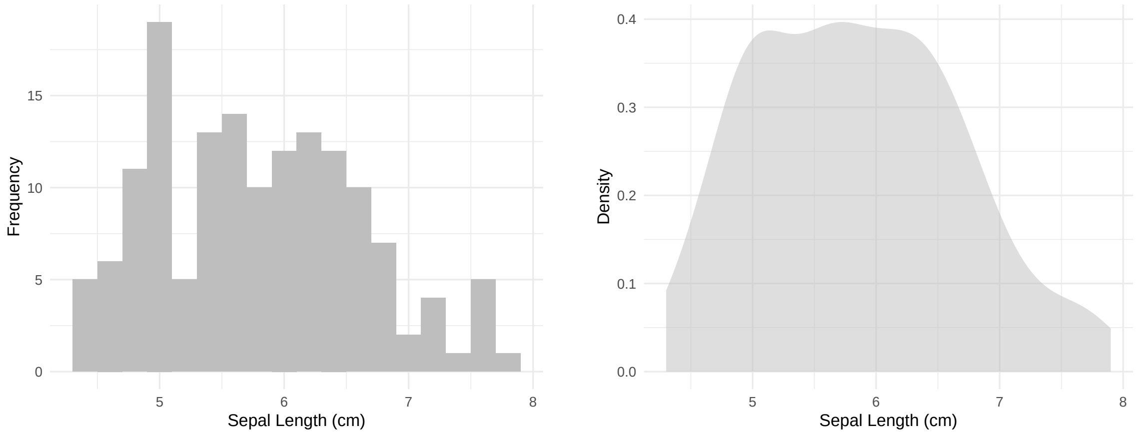


Figure 10: Histogram and Kernel Density Estimation of Sepal Length in Iris Dataset

3.1.3 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:

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

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

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

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

In Figure 10, the kernel density estimation of the dataset is presented. The distribution of sepal lengths is depicted by the kernel density curve. The peaks observed in the curve correspond to the primary concentration trends of sepal length within the data. A unimodal curve signifies a concentration of sepal lengths for most irises in that specific region. Conversely, a bimodal or multimodal curve suggests the existence of multiple concentration areas.

3.2 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.

3.2.1 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:

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

where n is the number of observations in the set.

3.2.2 Simple linear regression

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

Theory of Simple Linear Regression
Let $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ denote n explanatory variables and let $\mathbf{Y} = (Y_1, Y_2, ..., Y_n)^T$ denote n corresponding response variables.

In a simple linear model, it is assumed that the response variables $Y_1, Y_2, ..., Y_n$ are uncorrelated with a common variance σ^2, and their expectations are given by $E(Y_i|x_i) = \beta_0 + \beta_1 x_i$. The expectations generated by β_0 and β_1 given x_i can be expressed as:

$$\mathrm{E}(\mathbf{Y}|\mathbf{x}) = \begin{pmatrix} \beta_0 + \beta_1 x_1 \\ \beta_0 + \beta_1 x_2 \\ \vdots \\ \beta_0 + \beta_1 x_n \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \beta_0 + \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \beta_1 = \mathbf{1}_n \beta_0 + \mathbf{x}\beta_1 [11],$$

where $\mathbf{1}_n$ is an n-vector of 1's.

Given design matrix $\mathbf{X}$ where $X_i = (1, x_i)$ and$\beta = (\beta_0, \beta_1)^T$, then $E(Y_i|x_i) = X_i\beta$. These assumptions can be equivalently written in the vector form:

$$\mathrm{E}(\mathbf{Y}|\mathbf{x}) = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \mathbf{X}\beta, \quad \text{and} \quad \mathrm{var}(\mathbf{Y}|\mathbf{x}) = \begin{pmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{pmatrix} = \sigma^2 \mathbf{I}_n.$$

Least Squares Estimation

The residual sum of squares (RSS) is a measure of the goodness of fit in a regression model, where residuals are the differences between the response variables y_i and responses generated by the regression model $\mathrm{E}(\mathbf{Y}_i|\mathbf{x})$. In least squares estimation, the goal is to find values of parameters $\beta = (\beta_0, \beta_1)^T$ to minimise the RSS, denoted by Q:

$$\mathrm{Q} = \sum_{i=1}^{n} [y_i - \mathrm{E}(Y_i|\mathbf{x})]^2 = [\mathbf{y} - \mathrm{E}(\mathbf{Y}|\mathbf{x})]^T [\mathbf{y} - \mathrm{E}(\mathbf{Y}|\mathbf{x})] = [\mathbf{y} - \mathbf{X}\beta]^T [\mathbf{y} - \mathbf{X}\beta],$$

where $\mathbf{y}$ is n-vector of response variables and $\mathbf{X}$ is the $n \times 2$ design matrix. The partial derivative of Q with respect to vector β is:

$$\frac{\partial Q}{\partial \beta} = 2(\mathbf{X}^T\mathbf{X}\beta - \mathbf{X}^T\mathbf{y})[11],$$

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

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

The least squares estimate of β is given by:

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

Case example: 1970s automobiles

In this section, we will study the performance of 1970s automobiles using the mtcars dataset, employing the method of linear regression. Performance is measured in Miles per Gallon (mpg); the higher the mileage, the more efficient the automobile. We will start with the visualisation of a simple linear regression model, followed by the discussion of linear regression models and the model selection method.

In the preliminary stages of data exploration, calculating the correlation matrix is a crucial step before engaging in regression modeling. The correlation matrix provides valuable insights into the relationships between different variables in the dataset. This is also important for understanding multicollinearity issues, which occur when covariates are highly correlated.

Multicollinearity implies that the effects of individual covariates become intertwined. This intertwining can lead to erratic changes in the coefficient estimates of the regression model in response to small changes in the data. The accuracy of the model's predictions is undermined by the instability of coefficients[10].

The correlation matrix of all variables in the mtcars dataset is shown in Figure 11.

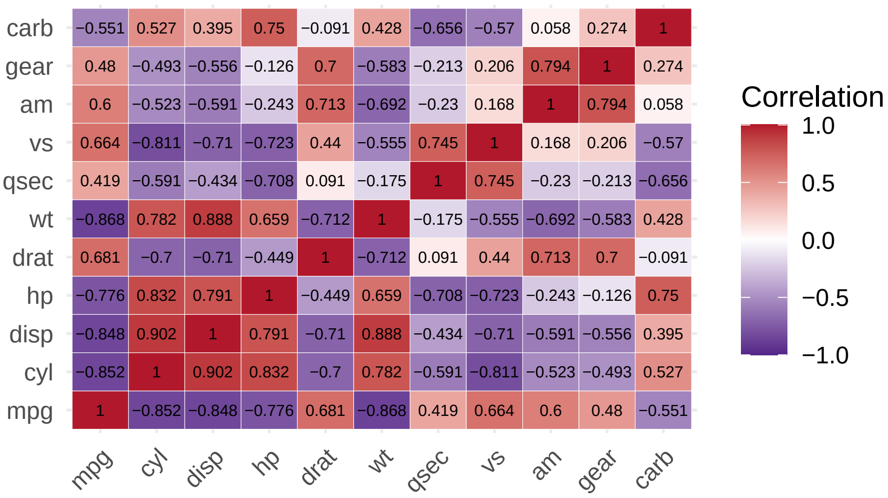


Figure 11: Correlation matrix of variables in mtcars dataset

For the simple linear regression model, the response variable is Miles per Gallon (mpg), and we select weight (wt) as the explanatory variable. Note that mpg and wt are highly correlated, with a correlation coefficient of -0.868. This suggests that wt may have strong predictive power for mpg. Use the R function `lm()` to calculate the linear regression model, with the summary displayed below. Observe that the t-test yields a p-value of 1.29×10^{-10}, which is less than 0.001. This indicates that the variable wt holds high statistical significance in this model. For the fitted model, the slope is $\beta_1 = -5.3445$, meaning that for every increase of 1000 lbs, the car efficiency decreases by 5 miles per gallon. The simple linear regression line is displayed in Figure 12.

```
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
```

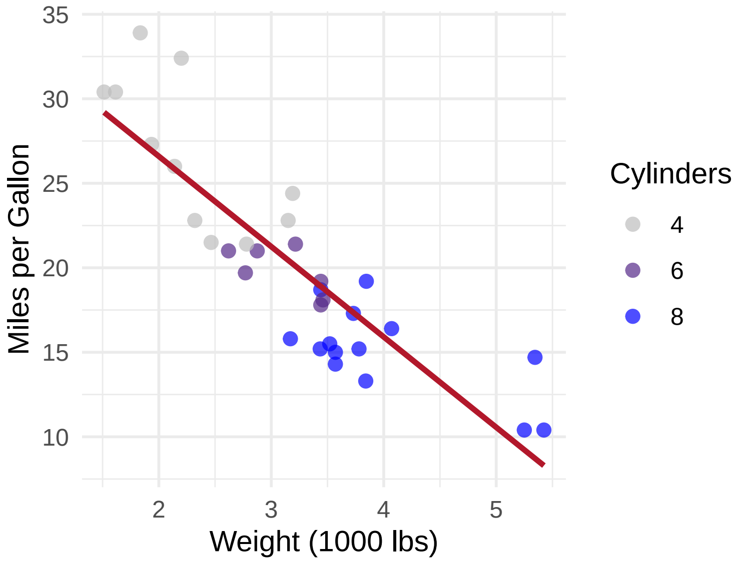


Figure 12: Scatter plot of car weights vs MPG

Model selection methods, such as the Akaike Information Criterion (AIC), play a crucial role in statistical modeling. The R algorithm `step()`, which minimises the AIC score, aids in selecting the most appropriate regression model from a set of explanatory variables. Starting with all explanatory variables, the algorithm iteratively excludes one covariate at each step. This algorithm aims to find the best-fitting model by balancing goodness of fit and simplicity. It's important to note that `step()` provides comparison between models rather than an absolute measure of model quality.

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

The procedure for selecting the best-fitted linear regression model using the `step()` algorithm is shown in Appendix A, where our best-fitted model is `formula = mpg ~wt + qsec + am`. Given 10 explanatory variables, Miles per Gallon is best predicted by weight (wt), quarter-mile time (qsec), and Transmission (am), where 0 represents automatic and 1 represents manual.

3.2.3 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.

Mathematical Intricacies of Bubble Charts

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:

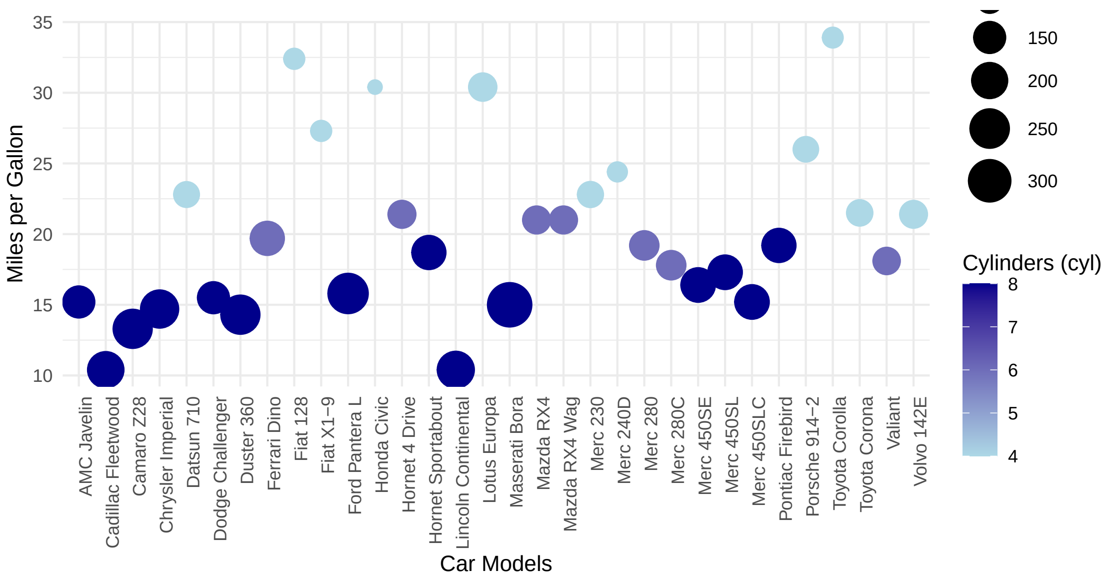
$$S = k \cdot \sqrt{V},$$

where S represents the size of the bubble, V 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

The bubble chart shown in the following figure **??** visualises data from the same dataset as above. It depicts 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).



The bubble plot reveals a clear pattern: as cylinder count increases, so does horsepower, but at the expense of lower fuel efficiency. Contrary to expectations, cars with fewer cylinders, despite lower horsepower, demonstrate better fuel economy, positioned higher on the graph. Meanwhile, vehicles with more cylinders, often associated with higher performance, cluster lower, indicating lower fuel efficiency. This plot highlights the trade-off between horsepower, cylinder count, and fuel efficiency, challenging the assumption that more cylinders equate to better performance, emphasising the significance of considering fuel economy in car selection.

3.3 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 organise the data into columns and rows. In Figure 14, 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 visualising relationships between covariates within a model. An example of the necessity to analyse 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 covariates, the regression model requires the removal of one covariate from the set of covariates. 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.

3.3.1 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 13, 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 14, 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.

```r
# 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]
```

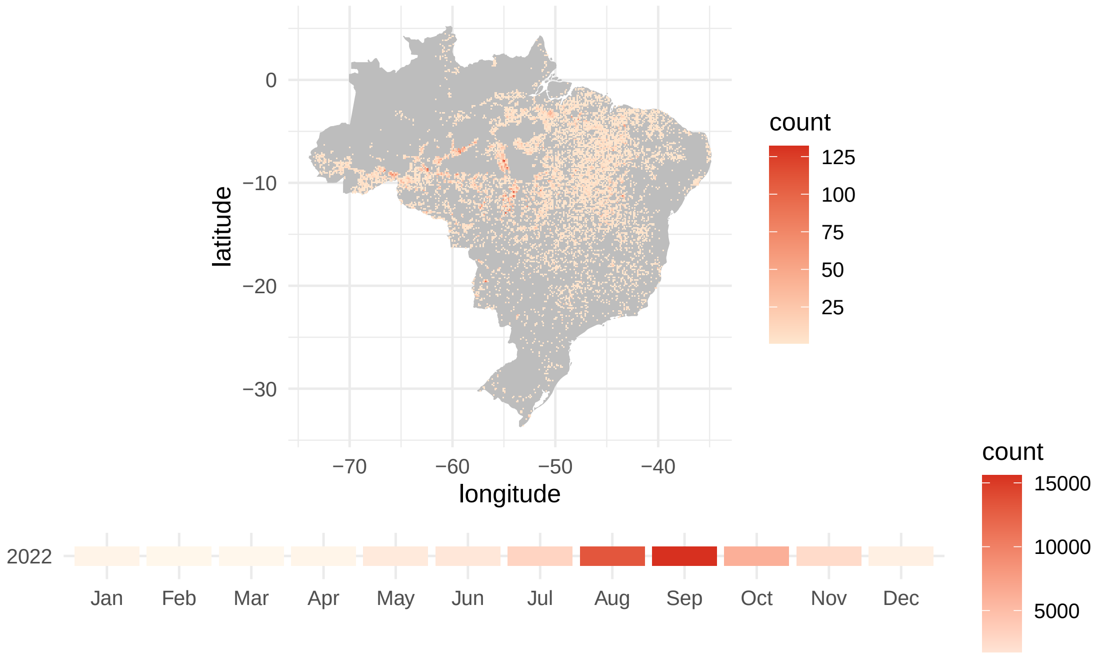


Figure 13: Frequency of Fire by Space and Time, FY22

```
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)
```

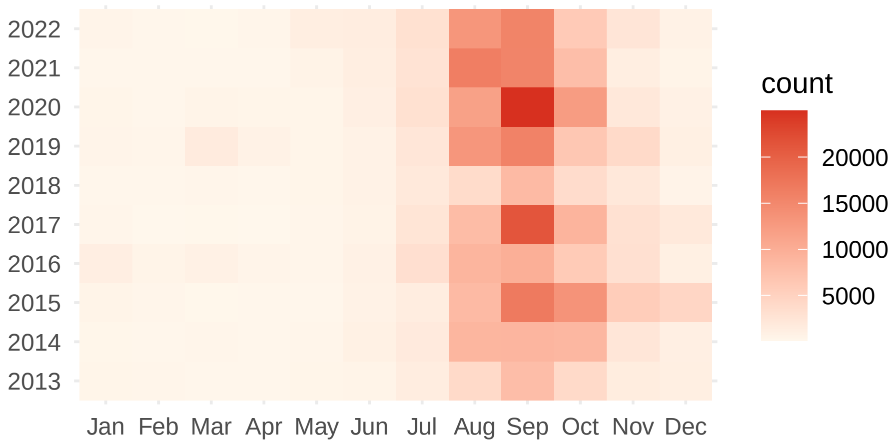


Figure 14: Frequency of Fire Occurrences, FY13-22

3.4 Line Charts and Time Series Visualisation

Line charts are fundamental tools in data visualisation, particularly useful for displaying time series data. A line chart represents n data points $\{(x_i, y_i)\}_{1 \le i \le n}$ on a Cartesian coordinate system, with the x-axis often denoting time intervals or ordered categories and the y-axis representing the measured values.

Basics of Line Chart

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. Also, a series of linear interpolations between pairs of data points could be used. These interpolations assume that the change between two points is uniform or linear. This linear approach is mathematically represented as:

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

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.

Time Series

Time series visualisation is particularly suited to line charts. A time series is a collection of observations x_t, where t denotes the time point at which the observation is recorded. An index set T_0 which collects all the time points when observations are available. For instance, we often have $T_0 = \{0, 1, 2, ..., n\}$ for $n \in N$. By plotting these data points over time, line charts help in identifying long-term trends, seasonal patterns, and anomalies.

A time series can be viewed as a realisation of a stochastic process. And a stochastic process $X = (X_t)_{t \in T_0}$ is a collection of random variables X_t, where t denotes the time index and T_0 the index set. For a fixed event $\omega \in \Omega$ we obtain the realisation of the stochastic process (sometimes also called a sample path) which is given by $x_t = X_t(\omega)$, $t \in T_0$. In practice, line charts of time series data provide insights into the behavior of such stochastic processes over time.

Time Series Visualisation of Exchange rates

Here, plot daily and 49-day moving average exchange rates of exchange rate data in one figure.

```
# Plot daily and 49-day moving average exchange rates of CAN, EUR, USD to GBP

# First plot
p1 <- ggplot(plot_dt, aes(x=Date, y=Rate, color=Currency)) + geom_line() +
  labs(y="Exchange Rate to GBP", x = "", color="Currency")+
  theme_minimal() + theme(legend.position="none")
# Second plot
p2 <- ggplot(plot_data, aes(x=Date, y=Rate, color=Currency)) + geom_line() +
  labs(y="Exchange Rate to GBP", x = "", color="Currency")+
  theme_minimal() + theme(legend.position="none")
# Extract the legend
p2_legend <- cowplot::get_legend(p2 + theme(legend.position="right"))
# Combine the plots
combined_plot <- cowplot::plot_grid(p1, p2, rel_widths = c(1, 1), nrow=2)
cowplot::plot_grid(combined_plot, p2_legend, nrow=1, rel_widths = c(2, 0.5))
```

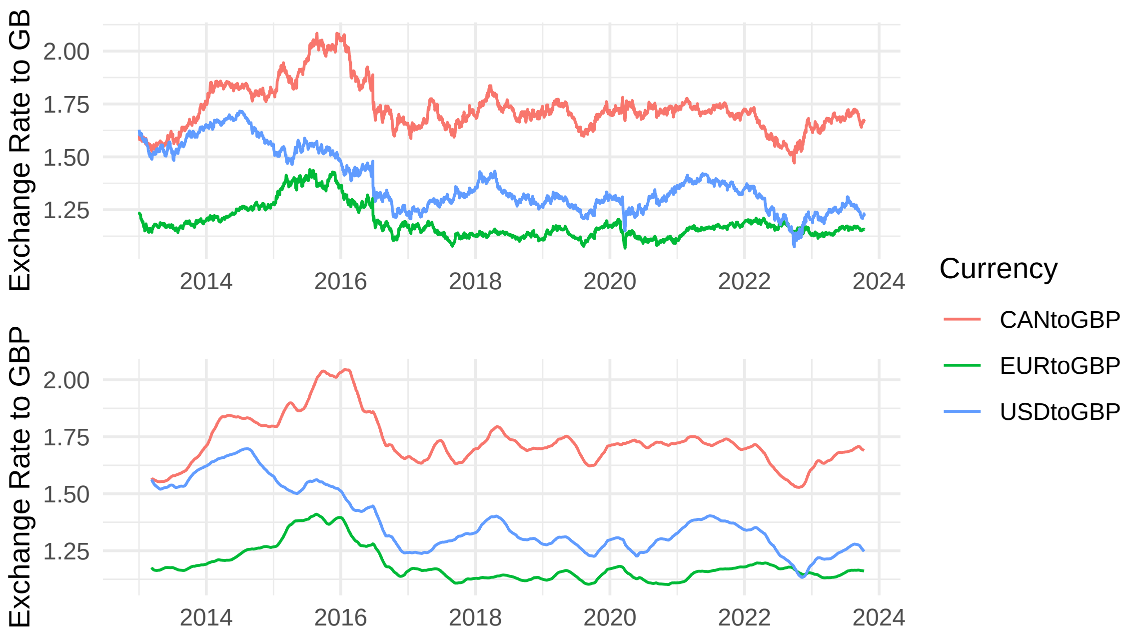


Figure 15: Daily (top) and 49-day moving average (bottom) exchange rates of CAN, EUR, USD to GBP

The first plot in Figure 15 presents a comparative visualisation of daily exchange rates for CAD, EUR, and USD against GBP, offering an overview of their trends and relative performance. This plot enables the identification of overall trends and periods of volatility for each currency pair, allowing for an assessment of their stability and strength relative to GBP. Notice that, there was a simultaneous and significant drop in all three currencies relative to the GBP. The concurrent nature of these declines across diverse currency pairs suggests that the driving factor is a depreciation of the GBP, rather than independent appreciations of the USD, EUR, and CAN. This notable depreciation of the GBP occurred around 2016, which coincides with the commencement of BREXIT process. Usually, the long term trend attracts financial analyst most. Therefore, filtering fluctuations and anomalies is important. Then, a 49-day moving average (MA49) was employed, shown in the second plot of Figure 15, to elucidate long-term trends while mitigating short-term fluctuations. This is mathematically represented as

$$\mathrm{MA}_{49}(t) = \frac{1}{49} \sum_{k=t-48}^{t} x_k,$$

where x_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 visualisations provides both a clear comparative and a quantitative perspective.

Decomposition of Time Series

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, X_t, can often be described as a combination of several distinct components: Trend component t_t: The underlying progression in the series, Seasonal component 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:

$$X_t = t_t + s_t + r_t.$$

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

$$X_t = t_t \times s_t \times r_t \quad \text{or} \quad \log(X_t) = \log(t_t) + \log(s_t) + \log(r_t).$$

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.

In R, a built in function named ”decompose()” is able to decompose the time series by addictive model or multiplicative model. And below is the demonstration of decomposition.

```r
# Plot decomposition of addictive time series model
ggplot(decomposed_df, aes(x = time, y = value)) + geom_line() +
  facet_wrap(~component, scales = "free_y", ncol = 1) + labs(x = "Date",y = "Exchange Rate to GBP") + theme_minimal()
```

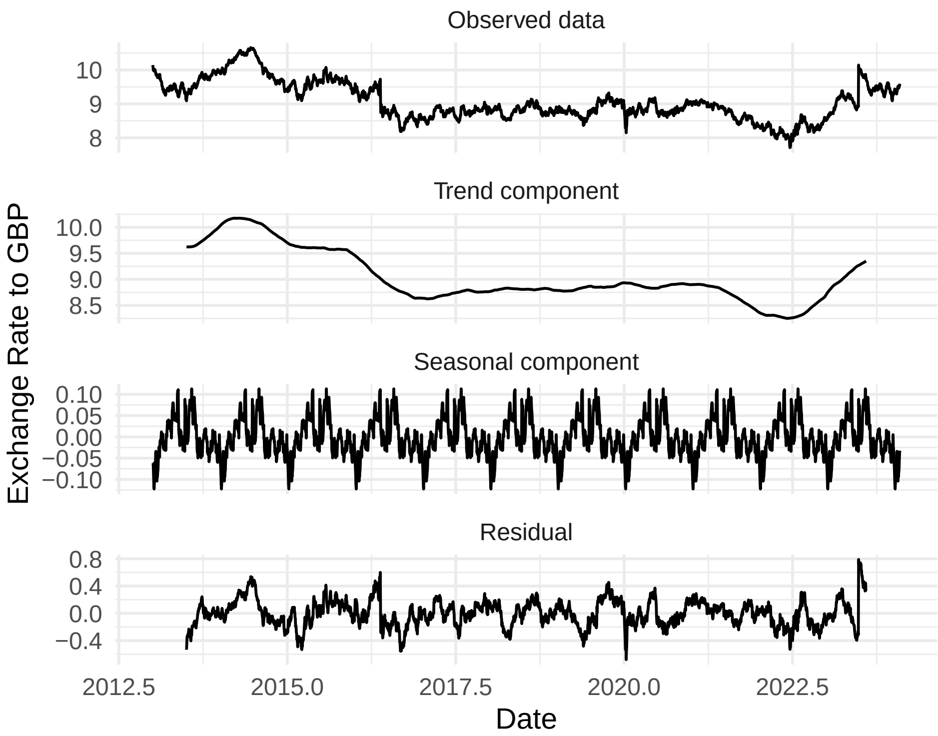


Figure 16: Decomposition of addictive time series model of CNY to GBP exchange rates

From Figure 16, the CNY to GBP exchange rate time series was decomposed into its fundamental components: trend, seasonality, and residual noise by additive model. This additive model, represented mathematically as $X_t = t_t + s_t + r_t$.

As illustrated in Figure 16, the trend component t_t of CNY to GBP exchange rate exhibits a distinct pattern over time: initially, it shows a gradual decrease and reached crest in 2022, followed by a sudden increase afterwards. It coincides with the tax reduction policy issued by UK government in 2022, which leads to a depreciation of 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 lag h, which is the time t with its values at a previous time $t-h$. The mathematical fomulation of ACF of time series will be given in the following paragraph.

Suppose we have a time series with observations denoted by $x_1, \ldots, x_n$. Then the sample mean is given by $\bar{x} = \frac{1}{n}\sum_{t=1}^{n} x_t$. And the sample autocovariance function at lag h in days of our time series is

$$\hat{\gamma}(h) := \frac{1}{n}\sum_{t=1}^{n-|h|}(x_{t+|h|}-\bar{x})(x_t-\bar{x}), \quad \text{for } -n<h<n.$$

Hence, the sample autocorrelation function at lag h in days is given by

$$\hat{\rho}(h) := \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} = \frac{\sum_{t=1}^{n-|h|}(x_{t+|h|}-\bar{x})(x_t-\bar{x})}{\sum_{t=1}^{n}(x_t-\bar{x})^2}, \quad \text{for } -n<h<n.$$

The value of $\hat{\rho}(h)$ 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, visualise the ACF for the CNY to GBP exchange rate to understand its time-dependent structure better.

```r
# Plot the Autocorrelation Function (ACF)
acf_data <- acf(MyData$CNYtoGBP, plot = FALSE)
# Plot using base R plotting
plot(acf_data, main="", xlab="Lag h", ylab="ACF")
```

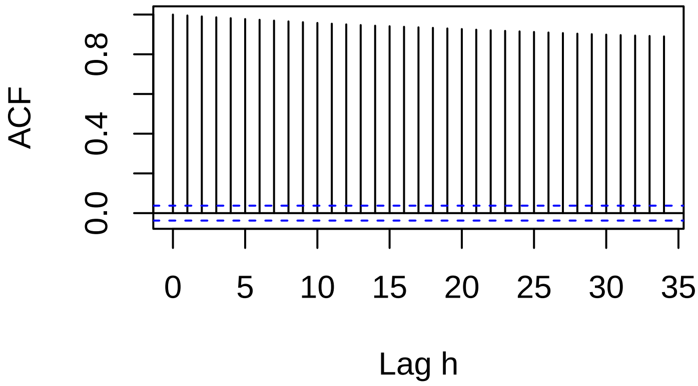


Figure 17: Autocorrelation Function at lag h in days of CNY to GBP Exchange Rate

From Figure 17, the 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.

References

[1] $500,000 in 1937 is how much today? https://www.calculateme.com/inflation/500000-dollars/from-1937/to-now, n.d. Accessed 9 Nov. 2023.

[2] K. Dakwa. The kallikak family – historical influences, current controversies, teaching resources. https://intelltheory.com/intelli/the-kallikak-family/, 2001. Accessed 9 Nov. 2023.

[3] M. de Carvahol. Incomplete data analysis. University of Edinburgh, Lecture 2, 9-10, 2023.

[4] Earth Science Data Systems. Firms frequently asked questions — earthdata. https://www.earthdata.nasa.gov/faq/firms-faq#ed-confidence, 2023. Accessed 12 Nov. 2023.

[5] E. Hawkins. Warming stripes — climate lab book. https://www.climate-lab-book.ac.uk/warming-stripes/, n.d. Accessed 9 Nov. 2023.

[6] A. Proctor. Five charts that changed the world - bbc ideas. https://www.bbc.co.uk/ideas/videos/five-charts-that-changed-the-world/p0fb69c1, 2023. Accessed 20 Oct. 2023.

[7] A. Rutherford. Where science meets fiction: the dark history of eugenics. https://www.theguardian.com/science/2022/jun/19/where-science-meets-fiction-the-dark-history-of-eugenics, 2022. Accessed 9 Nov. 2023.

[8] J.D. Smith and M.L. Wehmeyer. Who was deborah kallikak? *Intellectual and Developmental Disabilities*, 50(2):169–178, 2012.

[9] Wikipedia Contributors. Florence nightingale. https://en.wikipedia.org/wiki/Florence_Nightingale, 2019. Accessed 20 Oct. 2023.

[10] Wikipedia Contributors. Multicollinearity. https://en.wikipedia.org/wiki/Multicollinearity, 2019. Accessed 12 Nov. 2023.

[11] B. Worton. *Generalised Regression Models, Semester 1, 2023–2024*. University of Edinburgh, 2023.

Appendix A

```
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
```