Contents

1 Introduction

XXX

2 Theoretical Foundations of Data Visualisation

This chapter, "Theoretical Foundations of Data Visualisation," delves deep into the core principles and concepts that serve as the bedrock of this dynamic 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

In the pursuit of creating effective data visualisations, it is crucial to understand that behind every chart, graph or plot lies a solid theoretical framework. Theoretical underpinnings provide the foundation upon which data visualisation is built, shaping not only how we represent data but also how we perceive, understand, and interpret it.

2.1.1 Guiding Principles for Data Representation

Within this theoretical framework, we encounter a set of guiding principles that dictate how data should be represented visually. These principles encompass fundamental concepts such as:

- **Accuracy**: Data visualisations should accurately reflect the underlying data, minimising distortion or misinterpretation.
- **Simplicity**: The "less is more" principle applies to data visualisation. Simplified visuals often convey information more effectively than cluttered ones.
- **Clarity**: Visualisations should be clear and understandable to the intended audience, avoiding unnecessary complexity.
- **Relevance**: Information presented should be relevant to the message or question being addressed.
- **Consistency**: Visual elements, such as colour coding and labelling, should be used consistently throughout a visualisation.

2.1.2 Theoretical Framework and Visual Perception

One of the fundamental aspects of data visualisation theory is an understanding of how the human brain perceives visual information. This knowledge is instrumental in designing visualisations that resonate with viewers. It includes considerations like:

- **Gestalt Principles**: The Gestalt principles of visual perception, including proximity, similarity, and continuity, influence how we group and interpret visual elements in a visualisation.
- **Colour Theory**: The effective use of colour, including colour contrasts and harmonies, can enhance the clarity and impact of a visualisation.
- **Cognitive Load**: Minimising the mental effort required to process information is vital.

2.2 Visual Perception and Cognition

Understanding the intricacies of how humans interpret visual information is pivotal to the art and science of data visualisation. Thus, 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.

2.2.1 Human Visual Perception: Decoding Visual Information

Human visual perception is a remarkable cognitive process that allows us to decode and make sense of the world around us. When applied to data visualisation, it illuminates how viewers interact with and derive meaning from visual representations of data. Key aspects of human visual perception in the context of data visualisation include:

- **Pattern Recognition**: The human brain excels at recognizing patterns, making it adept at identifying trends, outliers, and relationships in data visualisations.
- **Perceptual Grouping**: We tend to group visually similar elements together, a principle known as perceptual grouping. This informs how we interpret clusters of data points and shapes in a visualisation.
- **Hierarchy of Perception**: Certain visual attributes are processed more quickly and efficiently than others. For example, colour is often 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.

2.2.2 Gestalt Principles

Gestalt psychology principles have long been recognised as fundamental to the field of visual perception and design. Gestalt psychology is a school of thought that emphasises how humans perceive and organise visual information. It posits that the mind seeks to create meaningful patterns and wholes from individual visual elements. The Gestalt principles thus, provide a framework for understanding how viewers naturally group and interpret visual stimuli.

Several key Gestalt principles play a pivotal role in shaping our perception of visual information. These principles include:

- **Proximity**: Elements that are close to each other are perceived as related or belonging to the same group. In data visualisation, proximity can be used to group data points or related information.
- **Similarity**: Elements that share similar visual attributes, such as colour, shape, or size, are perceived as belonging together. Similarity can be harnessed to encode categorical data or highlight relationships.
- **Continuity**: The human mind tends to perceive continuous patterns or lines as a single entity. In data visualisation, continuity can aid in representing trends or connections between data points.
- **Closure**: Viewers tend to mentally complete incomplete shapes or patterns. Closure can be employed to suggest relationships or connections even when not explicitly shown in the visualisation.
- **Symmetry**: Symmetrical elements are often perceived as more balanced and harmonious. Symmetry can be used for aesthetically pleasing and easily understandable visualisations.

2.2.3 Application of Gestalt Principles in Designing Visualisations

The application of Gestalt principles in designing data visualisations can lead to more intuitive and effective communication of information. Designers can strategically leverage these principles to:

- Group-related data points to enhance clarity and reduce visual clutter.
- Use colour or shape to signify meaningful categories or groupings.
- Create smooth, continuous lines or paths to guide the viewer's gaze through the visualisation.
- Suggest connections or patterns even in 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 is the art of simplifying and structuring raw data into formats that are comprehensible and insightful. It is the bridge that transforms numbers, text, and variables into visual elements that convey patterns, trends, and relationships. Effective data abstraction is at the heart of creating 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.

2.3.3 Trade-offs Between Abstraction and Information Loss

While abstraction is essential for simplifying complex data, it comes with trade-offs. Increasing levels of abstraction can lead to information loss, where fine-grained details or outliers are overlooked. It is crucial for data visualisation designers to strike a balance:

- **Clarity vs. Detail**: Increasing abstraction can enhance the clarity of a visualisation but may sacrifice detailed information that is important for certain analytical tasks.

- **Generalisation vs. Specificity**: Abstraction can provide a more generalised view of data, making it accessible to a wider audience. However, it may miss specific nuances that experts may require.
- **Context vs. Precision**: High-level abstraction can provide valuable context, but it may lack the precision needed 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 the right visualisation technique for your data type is pivotal to effective communication. Here are some examples of visualisation techniques matches with corresponding data types:

- **Nominal data**: Techniques such as bar charts and stacked bar charts are effective in displaying categorical information and relative proportions.
- **Ordinal data**: Ordinal data can be visualised using ordered bar charts, dot plots, or stacked bar charts, which maintain the ranking and order of the categories.
- **Interval data**: Interval data benefits from visualisation methods like line charts, histograms, and box plots, which highlight trends and distributions without assuming a true zero point.
- **Ratio data**: Ratio data can be effectively presented using scatter plots, histograms, and line charts, allowing for 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.

2.5.1 The Importance of Colour in Conveying Information

Colour is a potent tool for enhancing the understanding and impact of data visualisations. It enables the differentiation of data points, highlights trends, and provides context. Colour can be used to:

- **Encode Categorical Data**: Distinguish between different groups or classes using distinct colours.
- **Represent Quantitative Data**: Use colour intensity or gradients to represent values or magnitudes.
- **Add Context**: Apply colour to background elements, labels, or annotations to provide context and meaning to the visualisation.

2.5.2 Colour Perception and Colour Encoding in Visualisations

Understanding how colour is perceived by viewers is essential in data visualisation. Key principles include:

- **Colour Discrimination**: Consider that not all viewers may perceive colour in the same way. Ensure your colour choices are accessible to individuals with colour vision deficiencies (colour blindness).
- **Colour Encoding**: Select colour schemes that align with the message you want to convey. For example, warm colours like red and orange often signify caution or heat, while cool colours like blue and green convey calmness or coldness.
- **Colour Combinations**: Pay attention to how colours interact when placed adjacent to each other. Some combinations may create visual vibrations or make text difficult to read.

2.5.3 Avoiding Misleading Visualisations Due to Colour Choices

Misleading visualisations can result from inappropriate or deceptive use of colour. To avoid this:

- **Consistency**: Maintain consistency in colour usage throughout your visualisation. Use the same colour scheme for similar data categories or elements.
- **Avoid Distortion**: Ensure that colour choices do not exaggerate or distort the data. Overly intense or contrasting colours can lead to misinterpretation.
- **Legend Clarity**: Provide a clear and concise legend to explain the meaning of colours, especially when dealing with complex or unfamiliar colour schemes.
- **Test with Users**: Conduct user testing to ensure that your colour choices effectively convey the intended message and do not confuse or mislead the audience.

2.6 Theoretical Properties of Visualisations

Effective data visualisation is not solely about creating aesthetically pleasing graphics; it's also about adhering to key theoretical properties that optimise the expressiveness, precision, accuracy, and scalability of visual representations. In this section, we delve into these properties, including expressiveness and effectiveness, the data-ink ratio, and the principles of minimal ink, as well as precision, accuracy, and scalability.

2.6.1 Expressiveness and Effectiveness

- **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.

2.6.2 Data-Ink Ratio and the Principle of Minimal Ink

REVISE REFERENCES!! 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.

2.6.3 Precision, Accuracy, and Scalability

- **Precision**: Precision in data visualisation refers to the level of detail and granularity in the representation of data. Visualisations should strike a balance between showing enough detail to support accurate interpretation while avoiding overwhelming viewers with excessive complexity.
- **Accuracy**: Accuracy pertains to how faithfully the visualisation represents the true values in the data. Misleading or distorted visualisations can lead to incorrect conclusions. Therefore, maintaining accuracy is essential.
- **Scalability**: Scalability addresses how well a visualisation adapts to different data sizes or resolutions. Effective visualisations should be scalable, and capable of representing both small and large datasets without sacrificing 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.

2.7.1 Exploring the Concept of Cognitive Load in Visualisations

In data visualisations, cognitive load plays a significant role in how viewers engage with and understand the presented data. It is essential to strike a balance that ensures the visualisation conveys information effectively without overwhelming or overtaxing the viewer's cognitive capacity.

2.7.2 Strategies to Reduce Cognitive Load While Maintaining Complexity

- **Visual Hierarchy**: Establish a clear visual hierarchy that guides viewers' attention to the most important elements of the visualisation. Use techniques such as size, colour, and contrast to emphasise key information.
- **Simplify Labels and Text**: Reduce cognitive load by using concise labels and text. Avoid jargon and unnecessary complexity in annotations, ensuring that labels are informative and straightforward.
- **Interactive Features**: Implement interactive elements, such as tooltips and drill-down functionality, to provide additional information when viewers need it, reducing the need for dense, static visualisations.
- **Progressive Disclosure**: Present complex information gradually, allowing viewers to digest it in stages. Start with an overview and provide opportunities for users to explore details as needed.
- **Data Aggregation**: Consider aggregating data when it makes sense. Summarising data can reduce the cognitive load associated with interpreting fine-grained details.

2.7.3 Information Overload and Simplification Techniques

Information overload occurs when a visualisation overwhelms viewers with excessive data or visual elements, hindering comprehension. To combat information overload, the following simplification techniques can be applied:

- **Filtering**: Allow viewers to filter data based on specific criteria, enabling them to focus on the most relevant information.
- **Data Reduction**: Aggregate or summarise data to present overarching trends or patterns instead of inundating viewers with raw data points.
- **Studyboarding**: Use storytelling techniques to guide viewers through the data in a structured manner, helping them understand the context and relevance of the information presented.
- **Prioritisation**: Identify the most critical information and prioritise its display, relegating less essential data to secondary views or interactions.

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 Introduction to Modern Data Visualization Methods

As data grows increasingly complex and vast, the tools and techniques for effectively conveying this information continue to expand and refine. In this section we introduce the data sets that will be used to illustrate each of the different visualisatio techniques.

3.1.1 Data Sets

1. **Mtcars dataset**: The mtcars dataset in R is a built-in dataset that contains information about various car models. It provides data on the characteristics of 32 different car models, which were available in the early 1970s. The dataset includes a total of 11 variables, each representing different attributes of these cars, such as miles per gallon (mpg), horsepower (hp), number of cylinders (cyl), and more. The mtcars dataset is often used for data analysis, visualization, and statistical modeling, making it a useful resource for exploring and practicing data science techniques in R.

2. **Annual fire in Brazil**: it is obtained from NASA. Each dataset from 2013 to 2022 contains over 200,000 observations. Over the decade, there are more than 3 million observations. The trend is impossible to analyse by eye. However, an exploratory analysis using heatmaps provides insights into this data.

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 bring out easily identifiable comparisons, and reveals trends.

3.2.1 Scatter Plots

Scatter plots, also known as dot charts or dot density plots, offer a straightforward yet mathematically intriguing method for visualizing data. At their core, they display individual data points as dots along a single axis, where each dot representing a single observation.

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 (the most frequent value), skewness (asymmetry), and potential outliers.

They're particularly useful for comparing multiple data sets, identifying patterns, and detecting data irregularities. Their simplicity is what makes dot plots a valuable tool for both introductory statistics education and exploratory data analysis.

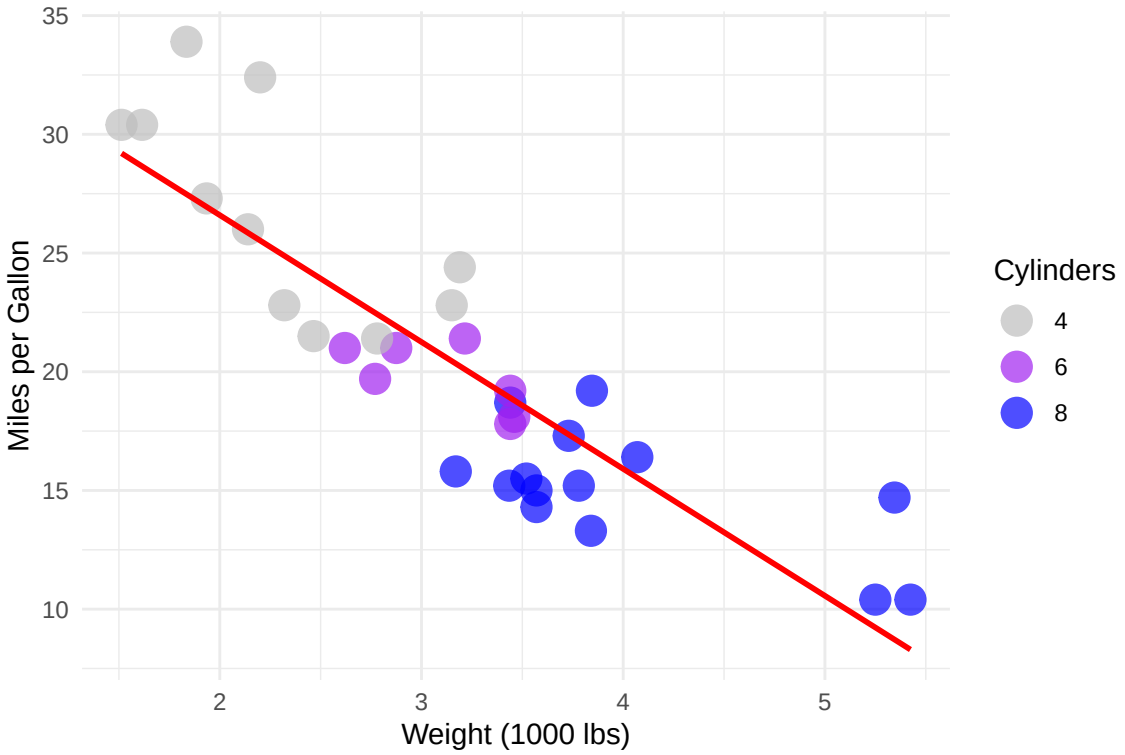


Figure 1: Scatter plot of car weights vs MPG

3.2.2 Scatter Plots in Practice

In this example, we'll create a scatter plot that visualizes the relationship between two variables - the weight of cars and the amount of miles traveled per gallon of petrol. We'll use the "mtcars" R dataset.

```
## `geom_smooth()` using formula = 'y ~ x'
```

3.2.3 Analysis

In Figure 1, provides insights into the relationship between cars' weight and their MPG, with the added dimension of color-coded cylinders.

Clusterings

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 in 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.

3.2.4 Regression and the Regression Line

Linear regression is a fundamental statistical method used to model the relationship between a dependent variable (often denoted as Y) and one or more independent variables (commonly denoted as X). The objective of linear regression is to find a linear equation that best represents this relationship. In simple linear regression, with one independent variable, the linear regression line is expressed as:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Here, β_0 is the intercept, β_1 is the slope, X is the independent variable, and ϵ represents the error term. The objective of the linear regression line is to minimise the sum of squared differences between the observed and predicted values of Y, which helps us understand how changes in X affect Y.

3.2.5 Bubble Charts

Bubble charts are a captivating data visualization 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 visualization 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 visualization 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:

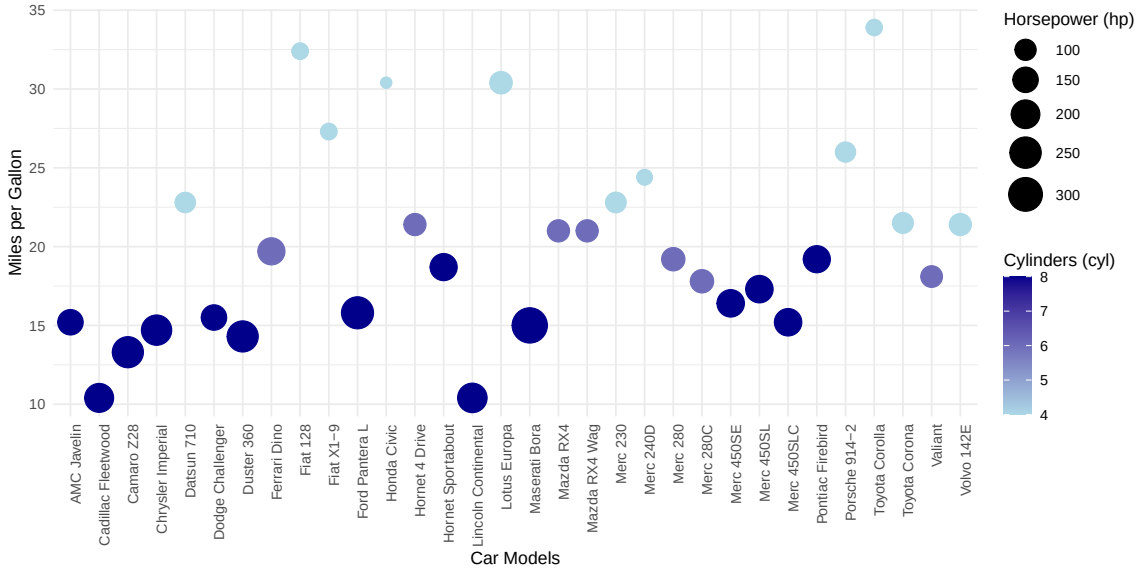
- S is the size of the bubble,
- V is the value of the variable being represented, and
- k is a scaling factor to control the bubble size.

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

3.2.6 Bubble Charts in Practice

This bubble plot visualizes 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))
```



3.2.7 Analysis

The plot's title, axis labels, and legends provide context and clarity to the visualization, 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 visualization, making it easy to differentiate between cars with different cylinder counts.

3.3 Bar Charts and Histograms

3.3.1 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 visualization. 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 visualization 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 visualizations. It offers a wealth of data visualization 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 visualization.

3.3.2 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.

3.3.3 Advantages of Bar Charts

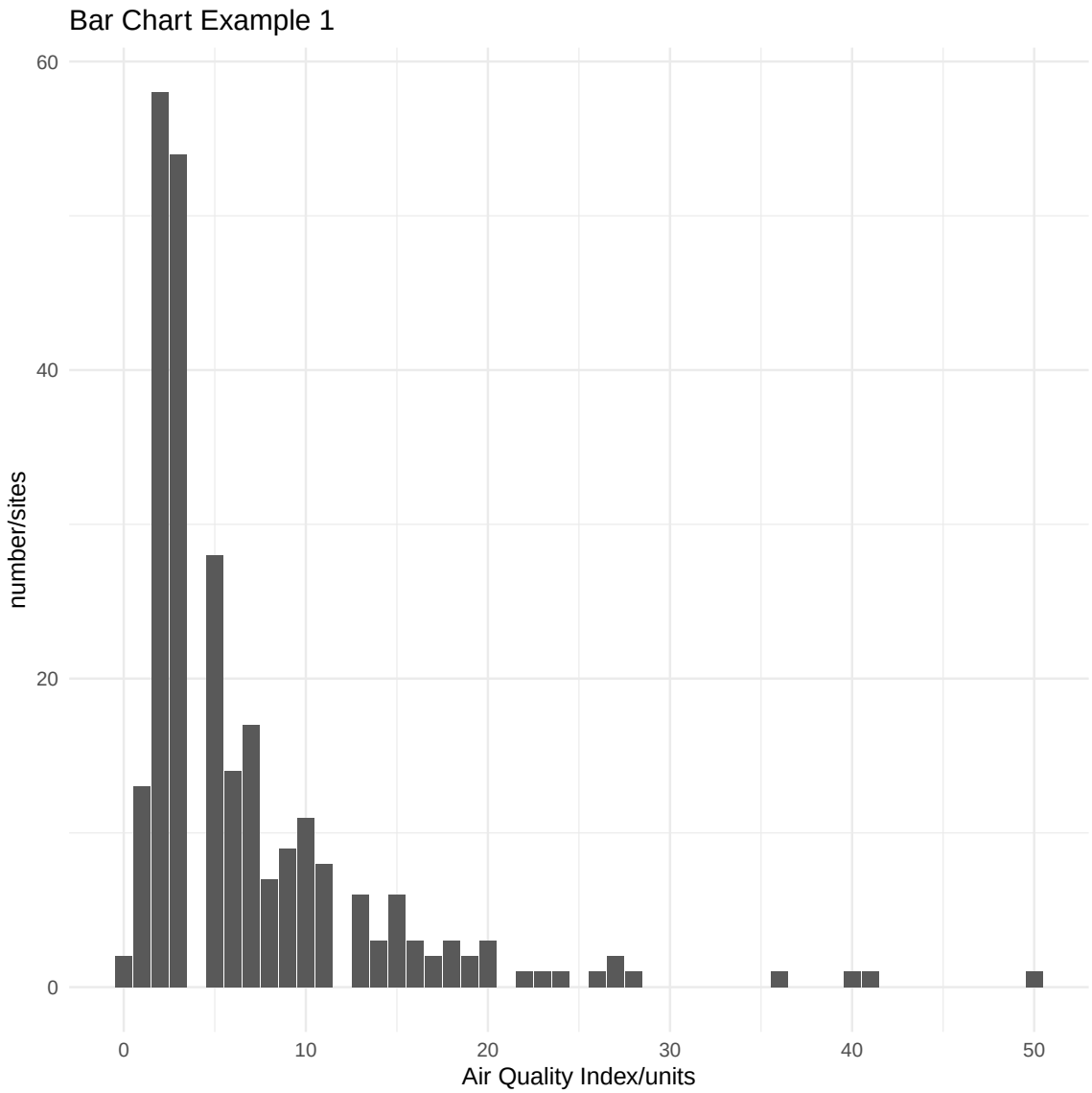
1. **Clarity and Simplicity**: Bar charts are structurally simple, making them easy to read and understand, allowing audiences to quickly grasp key information.
2. **Effective Comparison**: They provide a visual representation that makes comparing the size or value of different categories straightforward, especially when comparing a limited number of categorical data.
3. **High Flexibility**: They can be used to represent any type of data, be it continuous or discrete.
4. **Multilevel Representation**: Stacked or grouped bar charts can be used to represent multiple data series.

3.3.4 Disadvantages of Bar Charts

1. **Limited Data Representation**: They might not be suitable for representing large datasets as things can get cluttered.
2. **Potential Misinterpretation**: Without a zero baseline, bar charts can be misleading as they might exaggerate differences.
3. **Overcomplexity with Many Categories**: If there are too many bars, it can be challenging to discern information effectively.
4. **Requires Categorical Data**: Bar charts are not ideal for representing trends over continuous data, where line graphs might be more appropriate.

In this section, I will analyze the air quality dataset provided by the United States Environmental Protection Agency. In our dataset, we have data from over 200 locations. The Air Quality Index (AQI) ranges from 0 to 500. A higher AQI indicates increased levels of air pollution, leading to heightened health concerns. This implies that as the AQI rises, there is a greater risk to public health.

```
data <- read.csv("c4_epa_air_quality.csv")
ggplot(data, aes(x = aqi)) +
  geom_bar() +
  labs(title = "Bar Chart Example 1", x =
         "Air Quality Index/units", y = "number/sites") +
  theme_minimal()
```



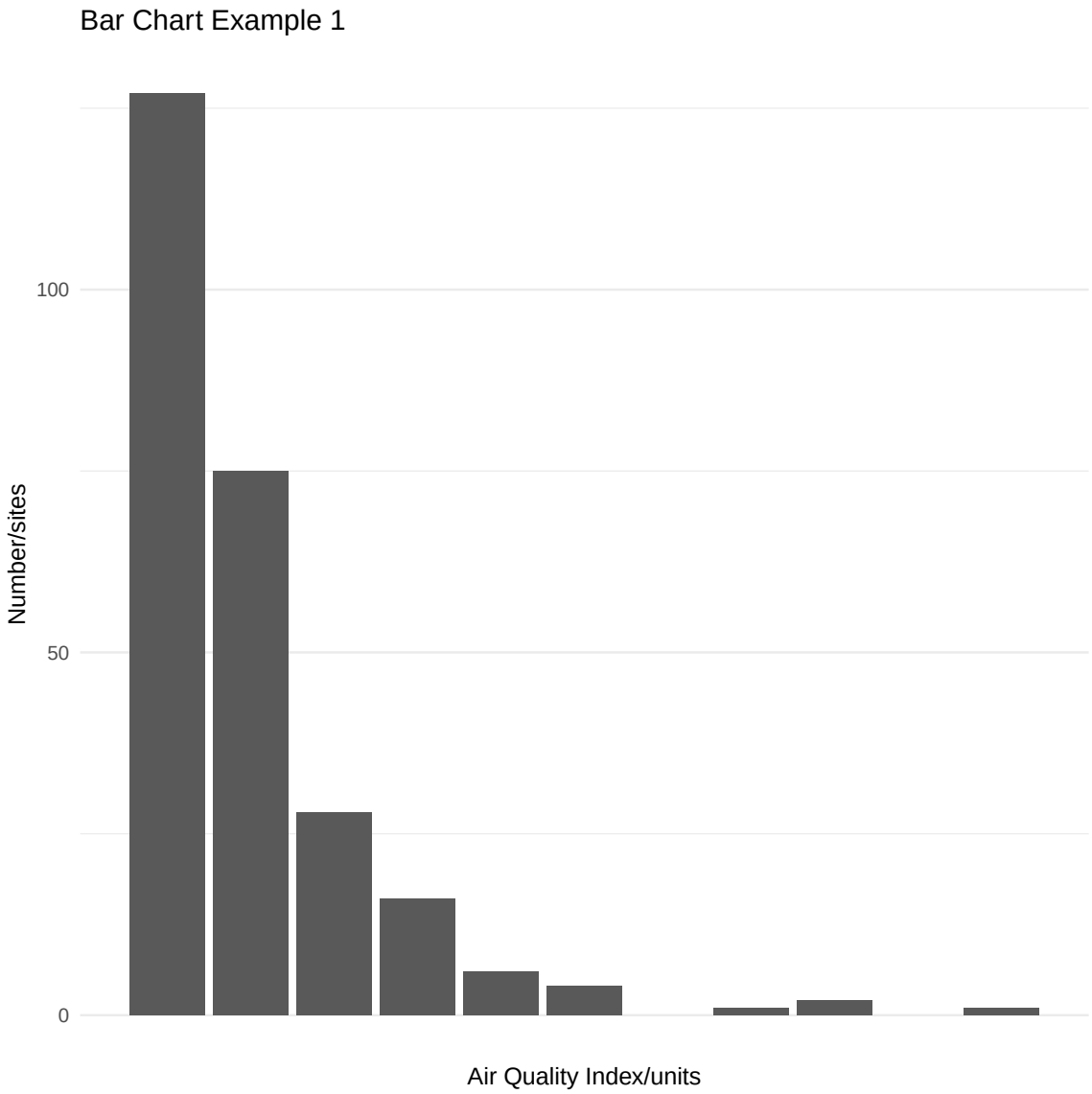
The above code is not optimal. Upon examinationan, we can see that there is an excussive number of different categories on the x-axis. Consequently, the multitude of vertical bars in the graph can potentially overwhelm and confuse readers. In order to solve the problem, we can use the "cut" function to divide the data into intervals of five units each. For instance, values from 0-5 would constitute one group, 6-10 another, 11-15 would form the next group, and so forth.

```
data <- read.csv("c4_epa_air_quality.csv") #
# load ggplot2 package
library(ggplot2)

# Use the cut function to divide the data into groups of five intervals
```

```r
data$group <- cut(data$aqi, breaks =
                    seq(0, max(data$aqi) + 5, by = 5),
                  right = FALSE, include.lowest = TRUE)
data$group <- as.numeric(data$group)

# Y-axis representing the number of occurrences of the X-axis label in the data
ggplot(data, aes(x = group)) +
  geom_bar() +
  labs(title = "Bar Chart Example 1",
       x = "Air Quality Index/units", y = "Number/sites") +
  theme_minimal() +
  scale_x_discrete(labels = scales::label_number(accuracy = 5))
```



We can also add some color to make our plot more attractive, here we can add sone color as well. In the code below, we set the color of the bar chart to blue while specifying the border color as black.

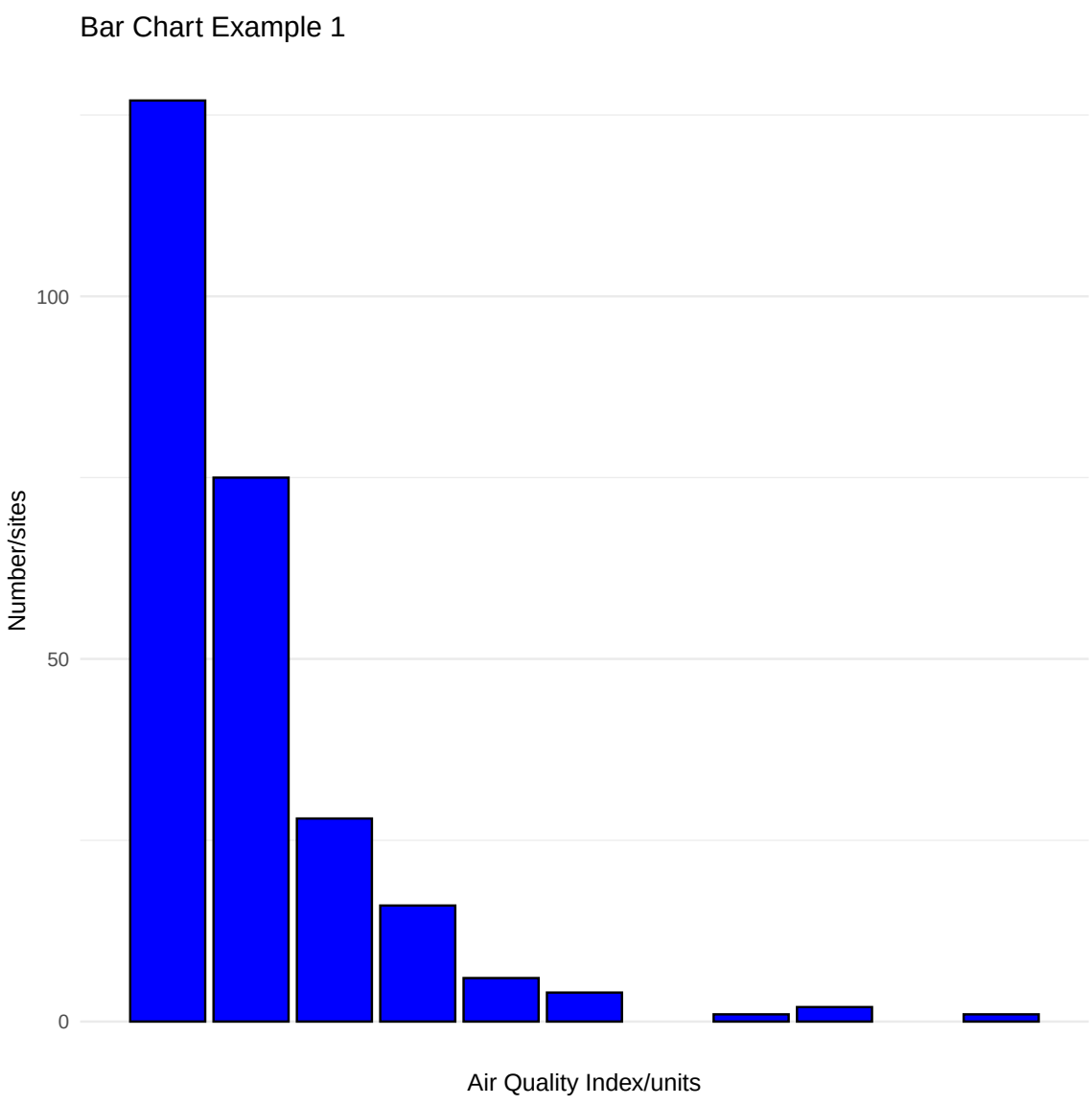
```
# read CSV dater set
data <- read.csv("c4_epa_air_quality.csv")

# load ggplot2
library(ggplot2)

# Use the cut function to divide the data into groups of five intervals
```

```r
data$group <- cut(data$aqi, breaks = seq(0, max(data$aqi) + 5, by = 5),
                  right = FALSE, include.lowest = TRUE)
data$group <- as.numeric(data$group)
# Converts the group column to numeric type

ggplot(data, aes(x = group)) +
  geom_bar(color="black",fill="blue") +
  labs(title = "Bar Chart Example 1", x = "Air Quality Index/units",
       y = "Number/sites") +
  theme_minimal() +
  scale_x_discrete(labels = scales::label_number(accuracy = 5))
```



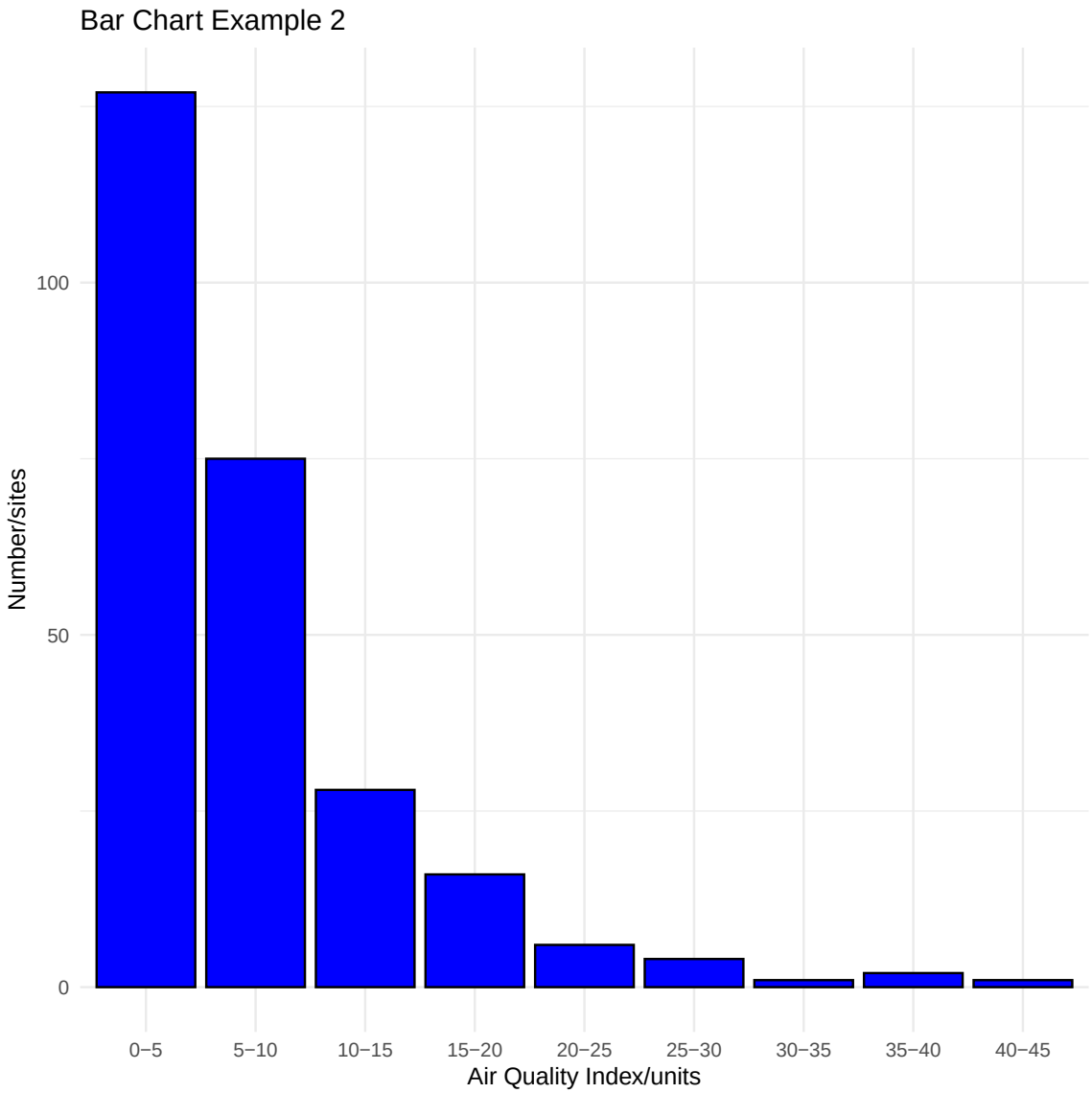
Finally, we add lables to the X-axis to define the range of each categories.

```
# read csv
data <- read.csv("c4_epa_air_quality.csv")

# load ggplot2
library(ggplot2)

# Use the cut function to divide the data into groups of five intervals
breaks_list <- seq(0, max(data$aqi) + 5, by = 5)
data$group <- cut(data$aqi, breaks = breaks_list,
                  right = FALSE, include.lowest = TRUE)


ggplot(data, aes(x = group)) +
  geom_bar(color="black", fill="blue") +
  labs(title = "Bar Chart Example 2", x = "Air Quality Index/units",
       y = "Number/sites") +
  theme_minimal() +
  scale_x_discrete(labels = paste0
                   (breaks_list[-length(breaks_list)], "-", breaks_list[-1]))
```



3.4 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 visualization 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.

3.4.1 Heatmaps - Fire in Brazil

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

In this illustrative example, heatmaps is used to visualize fire occurrences in Brazil. These heatmaps offer a spatially coherent representation, highlighting regions at high risk and seasonal patterns. Here, the heatmap is a power tool for identifying the risk of fire incidents. The data-driven insights empowers us to make informed decisions concerning preventive measures and strategies for firefighting.

From the heatmap, we can observe that certain locations have significantly higher fires count. However, we do not know the cause of this. Is this due to geographical location, or is it because fires were mostly man-made and used to clear forest areas for agriculture use?

Colour selection from colorbrewer2.

```
# 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 = 10))

time_heatmap <- ggplot(confident_fire_months_fy22,
                       aes(x = abb_month, y = as.character(2022), fill = count)) +
  geom_tile(width = 0.9, height = 1) +  # Create the heatmap tiles
  scale_fill_gradient(low = "#fff7ec", high = "#d7301f") +
  labs(x = "Month", y = "FY22", name = "count") +
  theme_minimal() +
  theme(axis.text = element_text(size = 10))

spacetime_fy22 <- grid.arrange(space_heatmap, time_heatmap, nrow = 2, heights = c(6,1.5))

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]
```

From the table, we can clearly see that August and September are the riskiest months in terms of fire hazard, whereas November to July hardly pose any risk at all. It's natural to ask the follow-up question: How does FY22 compare to previous years? Is it valid to claim that August and September are the fire hazard season?

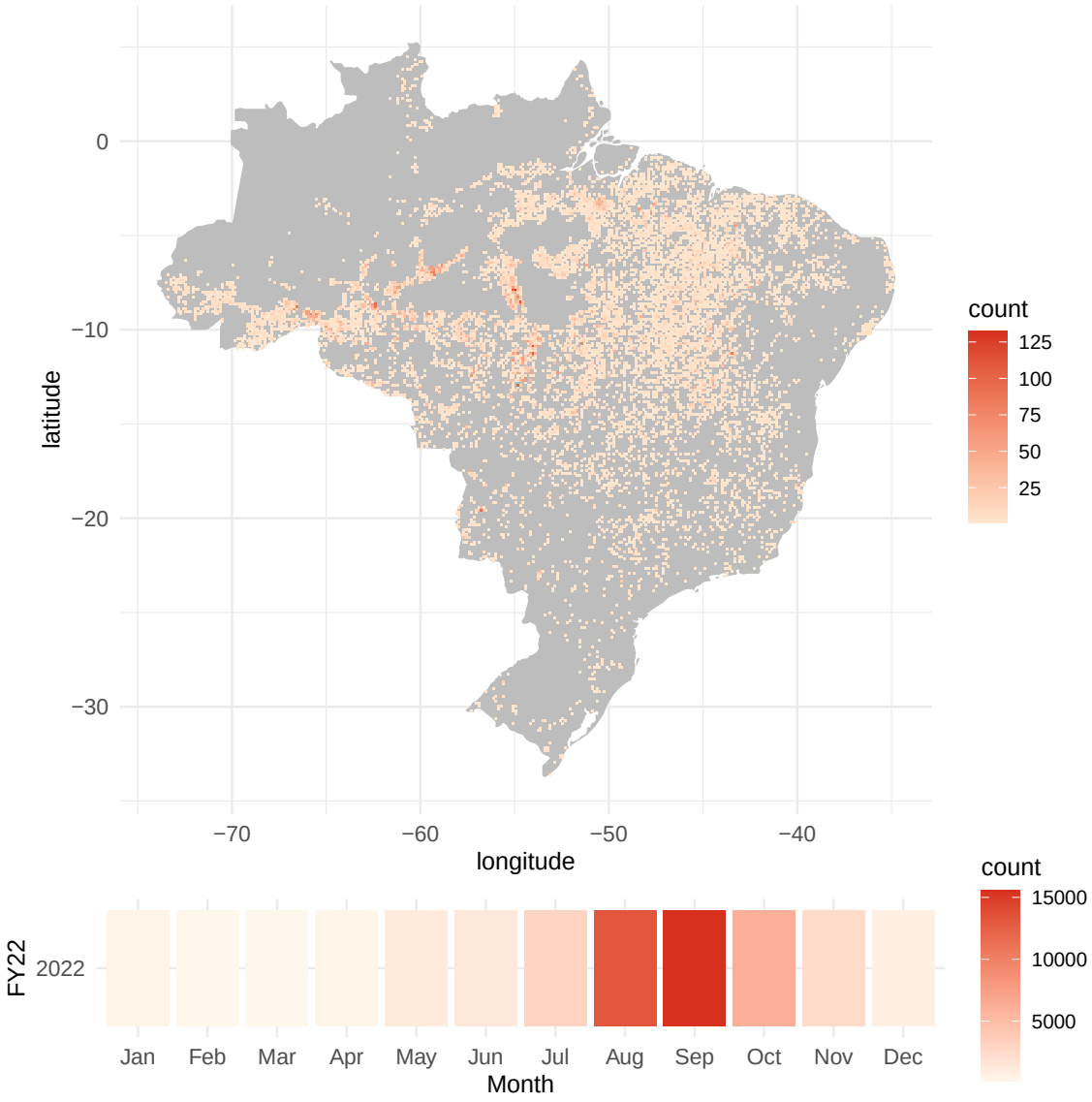


Figure 2: Frequency of Fire by Space and Time, FY22

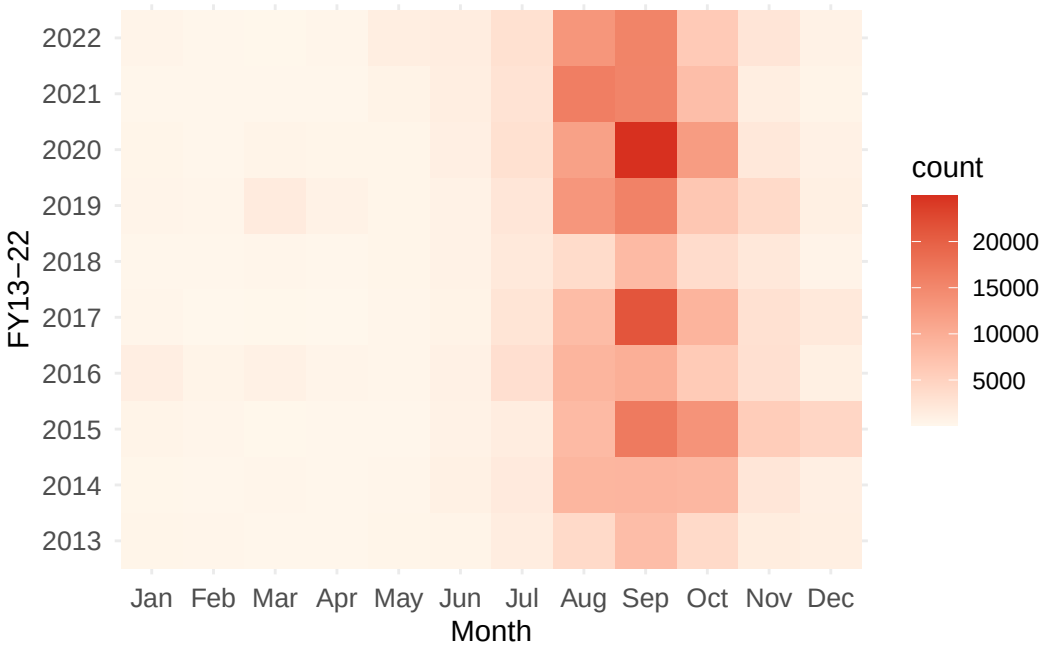


Figure 3: Frequency of Fire Occurrences, FY13-22

```
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 = "Month", y = "FY13-22") +
  theme_minimal() +
  theme(axis.text = element_text(size = 10))

print(heatmap_plot)
```

Indeed, the data showed a trend indicating that August to October have more fire occurrences compared to the rest of the year. There are clearly more fire hazards in those months.

The foundation of a heatmap is a data matrix, where each entry in this matrix represents an observation or measurement. Therefore, the first step to create a heatmap is to organize the data by columns and rows. In Figure 3, the structured data is displayed as a grid of coloured cells, where the colour intensity corresponds to the underlying frequency.

Heatmaps are powerful tools for visualizing relationships between covariables within a model. One example of why we need to analyse a matrix of correlations between variables is in regression models. In the real world, variables are often correlated, and complete independent relations are rare. Consequently, analysing pairwise correlations is essential. Highly correlated variables significantly impact the regression model. When faced with highly correlated variables, we need to choose one

variable from the correlated set. The selection is based on finding a regression model with the least Akaike Information Criterion (AIC) score among these variables. The AIC measures how much the linear model overfits the dataset. In other words, we want the regression model to explain the trend, and we do not want to overfit the model so that it explains the noise in the data set, which would lead to inaccurate predictions, as shown in Figure 4.

Figure 4: Danger of overfitting the regression model

3.4.2 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.

3.4.3 Use Cases for Treemaps

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

- **Disk Space Visualization**: Treemaps can be employed to visualize 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

3.5 Line Charts and Time Series Visualization

A **Line chart**, often referred to as a line graph or line plot, is a statistical chart composed of a Cartesian coordinate system, some points, and lines. It is commonly used to represent changes in numerical values over continuous time intervals or ordered categories. In a line graph, the x-axis is typically used for continuous time intervals or ordered categories (such as Stage 1, Stage 2, Stage 3). The y-axis is used for quantified data, and if it is negative, it is plotted below the y-axis. Lines are used to connect adjacent data points.

Line graphs are used to analyze trends in things that change over time or ordered categories. If there are multiple sets of data, they are used to analyze the interaction and impact of these data sets over time or ordered categories. The direction of the line represents positive/negative changes, and the slope of the line indicates the degree of change.

In terms of data, a line graph requires a continuous time field or a categorical field and at least one continuous data field.

3.5.1 Basic Components

- **X-Axis (Horizontal Axis):** Typically represents the independent variable, such as time or date.
- **Y-Axis (Vertical Axis):** Typically represents the dependent variable, like sales numbers, stock prices, or temperatures.
- **Line:** Connects the individual data points. In some line charts, multiple lines can represent different categories or sets of data.

3.5.2 Suitability for Displaying Trends Over Time:

- **Visual Clarity:** Line charts provide a clear and concise way to view changes over time. When data points are plotted over regular intervals (e.g., days, months, years), it becomes easy to see upward or downward trends.
- **Comparisons:** When you have multiple lines on a single chart, you can easily compare different sets of data. For instance, comparing sales data of two different products over time.
- **Identification of Patterns:** Line charts help in identifying patterns and anomalies. Seasonal patterns, cyclical events, and unexpected spikes or dips become evident.
- **Forecasting:** By viewing historical data trends on a line chart, analysts can make predictions or forecasts for future data points.
- **Simplicity:** They are easy to understand and interpret. Even if someone isn't data-savvy, they can grasp the general trend and major fluctuations from a line chart.

- **Flexibility:** They can be used for both short-term and long-term data. Whether you're looking at stock prices minute-by-minute over a single day or global temperature averages over a century, line charts can effectively represent the data.

3.5.3 Limitations:

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.

3.5.4 Discuss the importance of time series visualisation in data analysis.

Time series visualization refers to the graphical representation of time-ordered data points. In the world of data analysis, this form of visualization is invaluable for examining patterns, anomalies, and trends in datasets that evolve over time.

Uncovering Trends:
One of the primary advantages of time series visualization is the ease with which it allows analysts to identify long-term upward or downward trends in data. Recognizing these trends can help organizations make informed decisions about future strategies or interventions.

Detection of Seasonality:
Many datasets exhibit patterns that repeat over specific intervals, such as days, months, or years. Time series visualization makes it straightforward to spot such cyclical behaviors, which can be vital for businesses in sectors like retail or agriculture.

Identifying Anomalies:
Graphical representations can quickly highlight data points or periods that deviate significantly from the norm. These anomalies can indicate errors in data collection, or they may reveal significant events that need to be further investigated.

Forecasting and Predictions:
After identifying patterns in historical data, time series visualizations can aid in modeling future data points. Predictive modeling, underpinned by clear visualizations, allows businesses to make proactive decisions.

Facilitating Comparative Analysis:
Time series charts often allow for overlaying multiple data series on a single graph. This capability is useful for comparing different datasets or the same dataset under different conditions, leading to more comprehensive insights.

Conclusion:
Time series visualization is an indispensable tool in the arsenal of data analysts. It condenses large volumes of chronological data into easily interpretable graphics, enabling quick insights, better decision-making, and a deeper understanding of temporal dynamics in datasets. By providing a clear view of data trends, seasonality, and anomalies, time series visualization facilitates more informed and strategic actions in various domains.

3.5.5 Provide best practices for creating clear and informative line charts.

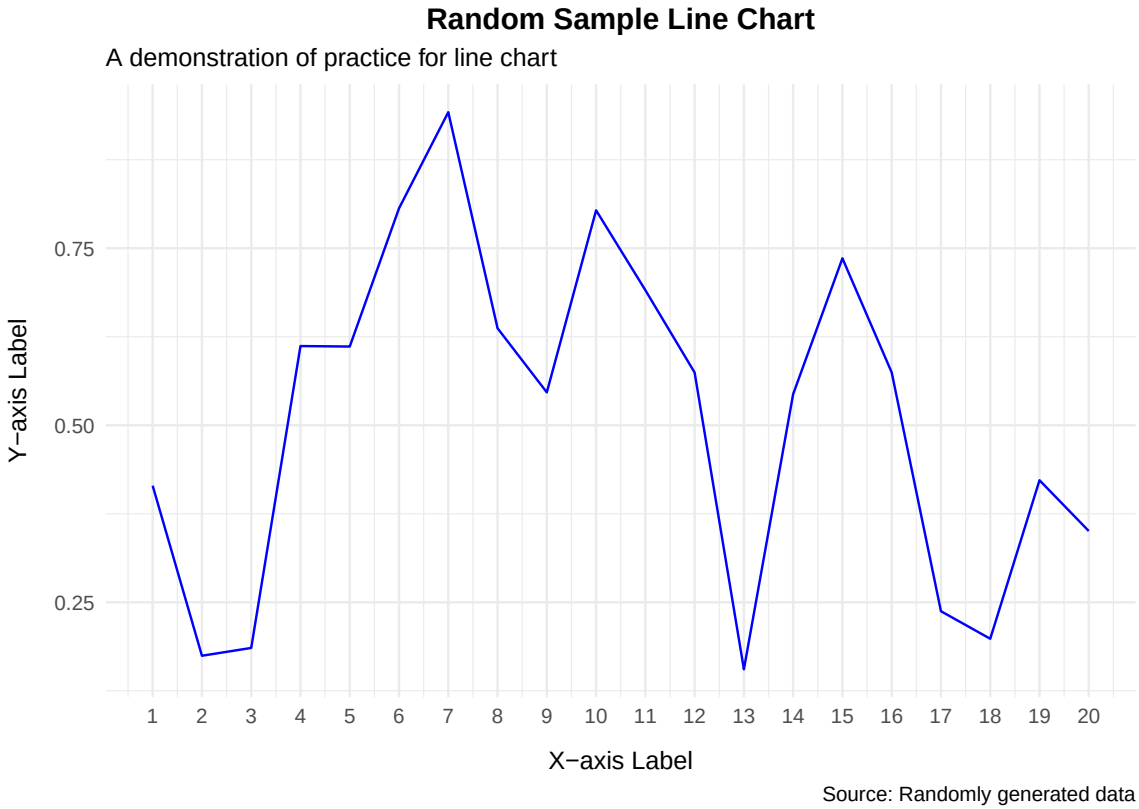
- Title and Labels: Every chart should have a descriptive title and axis labels to clearly convey the purpose of the visualization and the data being shown.

- Use of Colors: Colors should be chosen to clearly differentiate between different lines or data points but also be consistent with the overall theme or style.
- Gridlines and Background: Soft gridlines can help the viewer estimate values. A clean background aids in clarity.
- Line Types and Point Shapes: When multiple lines are on the same chart, use different line types and point shapes to differentiate between them.
- Consistent Scaling: The scale on the y-axis should be consistent so that the viewer isn't misled.
- Annotations: Important points or changes can be annotated directly on the graph.
- Legends: If there are multiple lines or data points with different colors/shapes, a legend should be provided.

Let's apply these practices:
First, we generate 2 series of random data.

```
x <- seq(1, 20)
y <- runif(20)
data <- data.frame(x = x, y = y)
```

Below is a line chart of the random sample:



3.5.6 Showcase real-world examples of time series visualisations

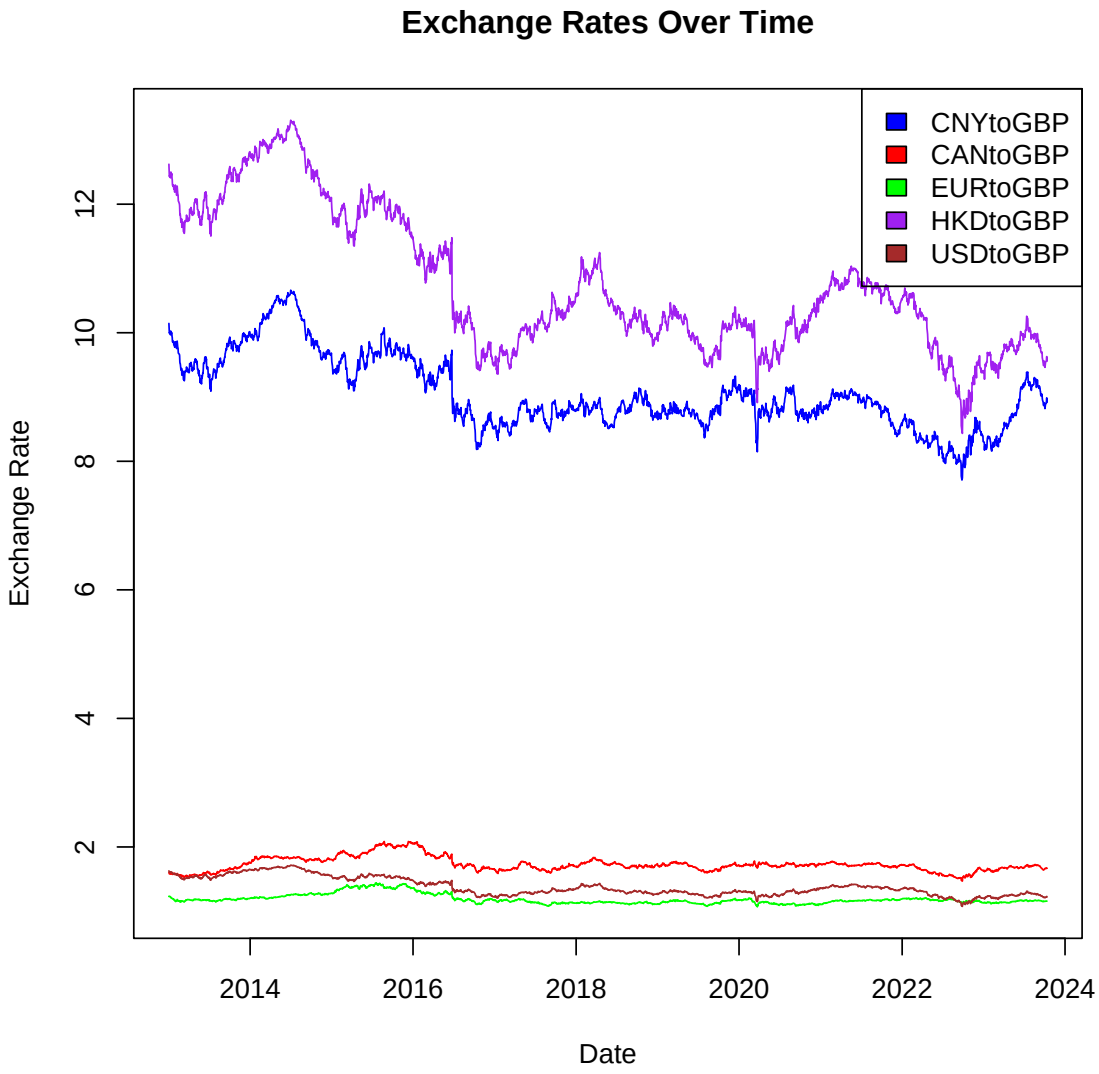
Time series of the daily CNY, CAN, EUR, HKD, USD versus GBP exchange reference rate data published by the European Central Bank over the time period from 01 Jan 2013 to 12 Oct 2023 (without weekends). The exchange rate tells you how many pounds you need to buy/sell 1 CNY, CAN, EUR, HKD, USD.

3.5.7 The data set has the format as below:

Date	**CNYtoGBP**	**CANtoGBP**	**EURtoGBP**	**HKDtoGBP**	**USDtoGBP**
%d-%m-%y	Value	Value	Value	Value	Value

Table 1: Field Information: CNY, CAN, EUR, HKD, USD to GBP

3.5.8 Multiple time series in one plot:

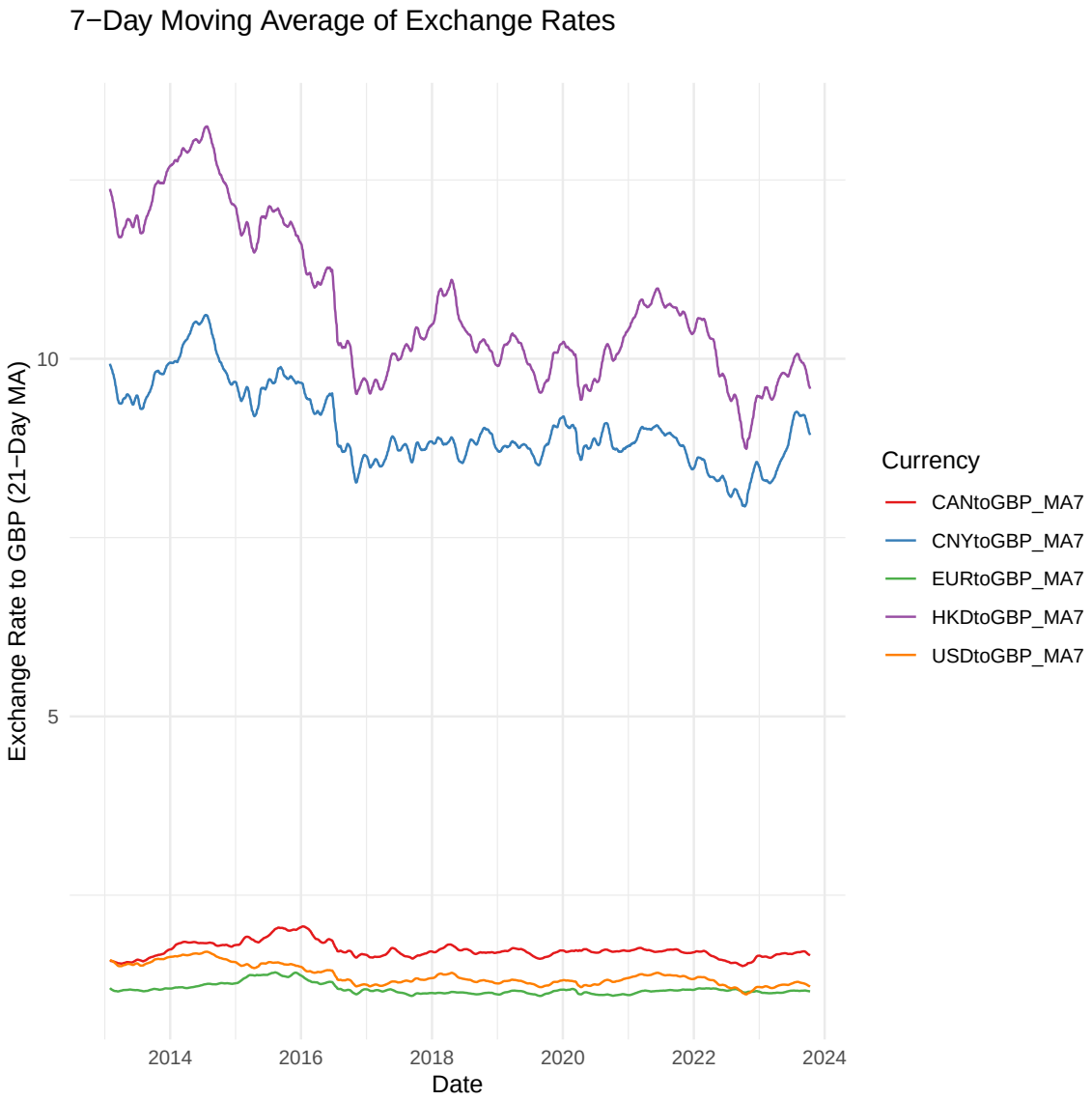



```r
# Calculate 21-day moving average for each currency
columns <- names(MyData)[!names(MyData) %in% "Date"]
for (col in columns) {
  new_col_name <- paste0(col, "_MA7")
  MyData[[new_col_name]] <- zoo::rollapply(MyData[[col]], width=21, FUN=mean, fill=NA, align='righ
}

# Plotting
plot_data <- MyData %>% gather(key="Currency", value="Rate", -Date) %>%
  filter(grepl("MA7", Currency))
```

```r
ggplot(plot_data, aes(x=Date, y=Rate, color=Currency)) +
  geom_line() +
  labs(title="7-Day Moving Average of Exchange Rates",
       subtitle="",
       y="Exchange Rate to GBP (21-Day MA)",
       x="Date",
       color="Currency") +
  theme_minimal() +
  scale_color_brewer(palette="Set1")

## Warning:  Removed 100 rows containing missing values (`geom_line()`).
```



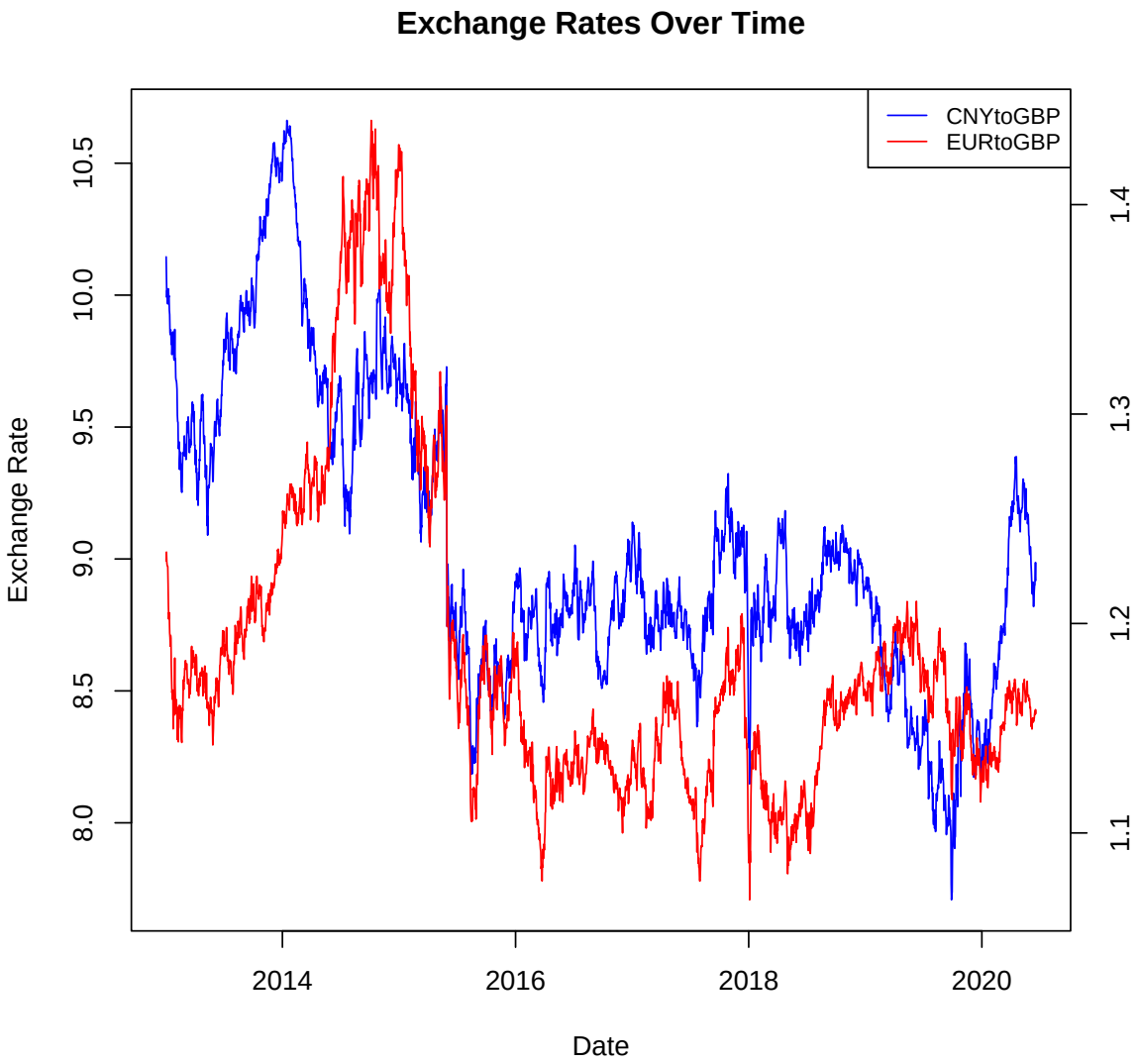
3.5.9 Decomposition of one time series into trend, seasonal, and random.

One of the primary advantages of time series visualization 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.

```
## Error in `ensure_igraph()`:
## !  Must provide a graph object (provided wrong object type).
```

3.5.10 Double y-axis time series plot.

If we want to display two different time series that measure two different quantities at the same time points, we can draw the second series again on the second Y-axis on the right side.



3.6 Network Graphs

Definition and Utility: Network graphs, often referred to as graphs or networks, are a powerful data visualization 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.

3.6.1 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.

3.6.2 Network Graphs in Practice

3.7 Sankey Diagrams

xxx

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

3.8 Geographic Maps and Spatial Data Visualisation

xxx

3.9 3D and Interactive Visualisations

xxx

3.10 Advanced Visualisation Techniques

xxx

4 Practical Implementations

XXX

5 Case Studies

5.1 Market Analysis Dashboards

XXX

5.2 Healthcare Data Visualisation

XXX

6 State-of-the-Art Approaches

XXX

7 Conclusion

XXX