```
## Error in library(forecast):  there is no package called 'forecast'
```

1 Chapter 3. Modern Methods of Data Visualisation

2 Chapter 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. This chapter tackles the modern data visualization methods, offering a comprehensive exploration of various visualization techniques, along with tangible examples accompanied by their underlying code.

3 Chapter 3.2 Scatter Plots and Bubble Charts

Scatter plots and bubble charts are fundamental data visualization techniques that provide valuable insights into the relationships and patterns within datasets. These visualizations are particularly effective for representing data points, making comparisons, and revealing trends. Below, we delve into the characteristics, peculiarities, and advantages of scatter plots and bubble charts.

3.1 3.2.1 Scatter Plots

A scatter plot is a simple yet powerful visualization that displays individual data points as dots on a two-dimensional plane. Each point represents a unique data entry, with one variable on the horizontal (x-axis) and another on the vertical (y-axis). Scatter plots are versatile and can accommodate various data types, including nominal, ordinal, interval, and ratio.

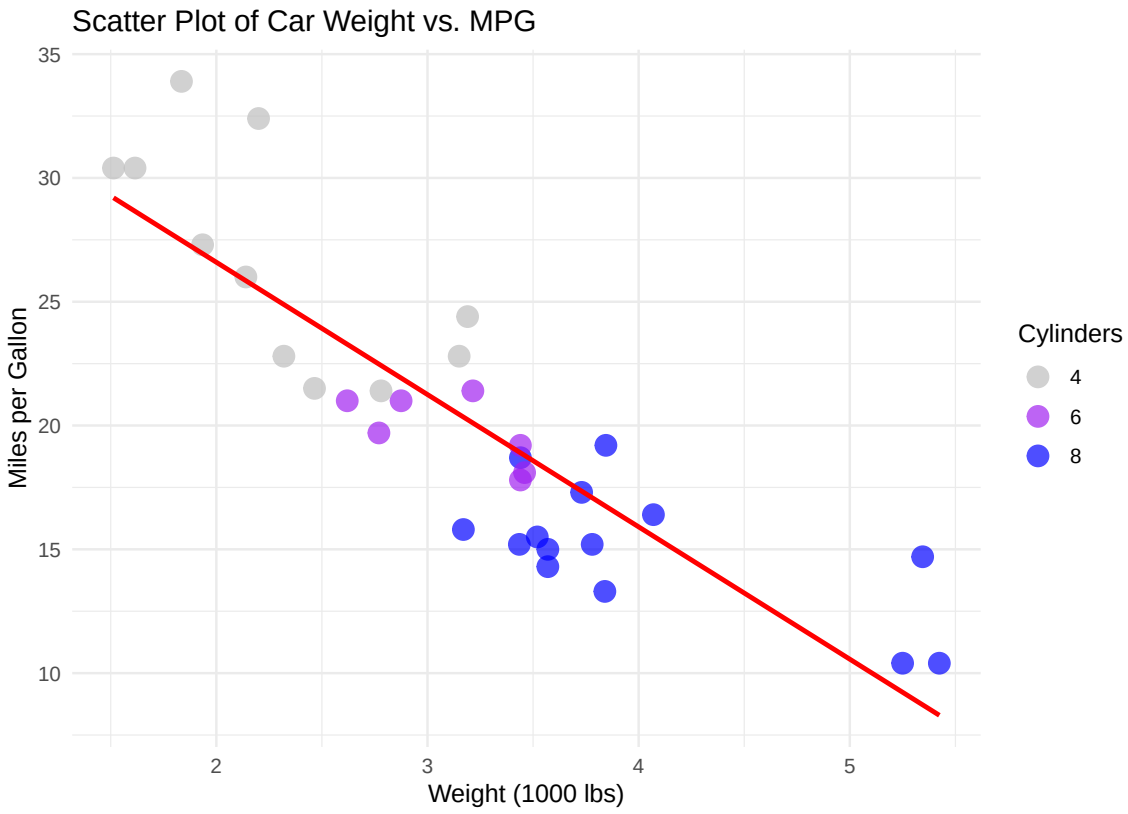
3.1.1 Advantages

- Easy identification of patterns and trends.
- Effective for outlier detection.
- Suitable for both bi-variate and multi-variate analysis.
- Offers a visual representation of the distribution of data points.

3.2 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, which contains information about various car models.

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



3.3 Analysis

The scatter plot displays the relationship between car weight (in thousands of pounds, "wt") and fuel efficiency (miles per gallon, "mpg") for different car models. While this graph plots two variable, through the use of colour, additional lines, and other subtle details, we can make of this basic graph, a more readable and visually interesting figure which the reader can get easily gather information from. Here are the key characteristics of the plot:

- Color-Coded Cylinders: The points are color-coded based on the number of cylinders in the engine (4, 6, and 8 cylinders). This allows for a quick visual differentiation of car types, enhancing the understanding of the data.
- Point Size and Transparency: Points have been enlarged and have a subtle degree of transparency for visual appeal. Larger, more prominent points are easier to see, while transparency helps in the visualization of overlapping points.
- Linear Regression Line: A blue linear regression trend line is included. It provides a visual representation of the overall relationship between car weight and fuel efficiency, indicating a negative correlation—cars tend to have lower fuel efficiency as their weight increases.

3.4 Regression and the Regression Line

Regression is a statistical technique used to model and understand the relationship between two or more variables. It is a fundamental tool in data analysis and predictive modeling. In simple linear regression, a linear equation is used to describe the relationship between a dependent variable (the one you want to predict) and one or more independent variables (the predictors). The equation takes the form of $Y = aX + b$, where Y is the dependent variable, X is the independent variable, a is the slope, and b is the intercept. Regression analysis calculates the best-fit line that minimizes the sum of squared differences between the predicted and actual values. This line provides valuable insights into the strength and direction of the relationship between variables, allowing us to make predictions and understand how changes in one variable affect another. Adding regression lines to scatter plots can enhance their value by providing a clearer depiction of trends and enabling more accurate predictions, turning the plot into a predictive tool rather than just a visual representation of data points.

3.5 3.2.2 Bubble Charts

A bubble chart is an extension of the scatter plot, where each data point is represented as a "bubble" or circle. In addition to the x and y axes, bubble charts introduce a third variable that is encoded by the size of the bubbles. This third variable allows for the simultaneous visualization of three data attributes.

3.6 Use Cases

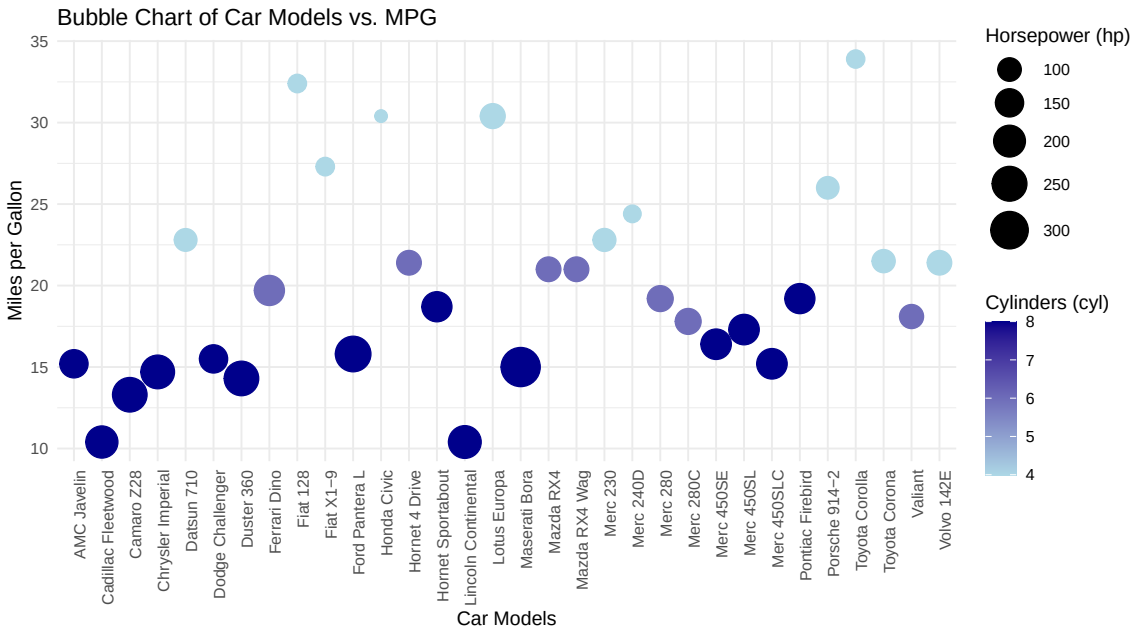
Bubble charts are particularly useful when you need to visualize the relationship between three variables. For example, in finance, a bubble chart can display the market capitalization, price-to-earnings ratio, and stock returns for different companies simultaneously. In public health, it can represent the population, vaccination coverage, and disease incidence for various regions. COULD INSERT REAL WORLD EXAMPLES

3.6.1 Advantages

- Visualising three data attributes in a single chart.
- Effective for exploring complex relationships and identifying patterns.

3.7 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).



4 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.

4.1 Key Considerations for Scatter Plots and Bubble Charts

1. **Scaling**: Consider the scaling of the axes and bubble sizes to prevent distortion of the data representation.
2. **Labeling**: Use clear labels, titles, and colour coding to make the visualisation more interpretable and meaning by decreasing the cognitive load.
3. **Interactivity**: Implement interactive features, such as tooltips or zooming, to allow users to explore individual data points in detail.

Chapter 3.3 Bar Charts and Histograms

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.

3.3.1 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.

3.3.2 Advantages of Bar Charts

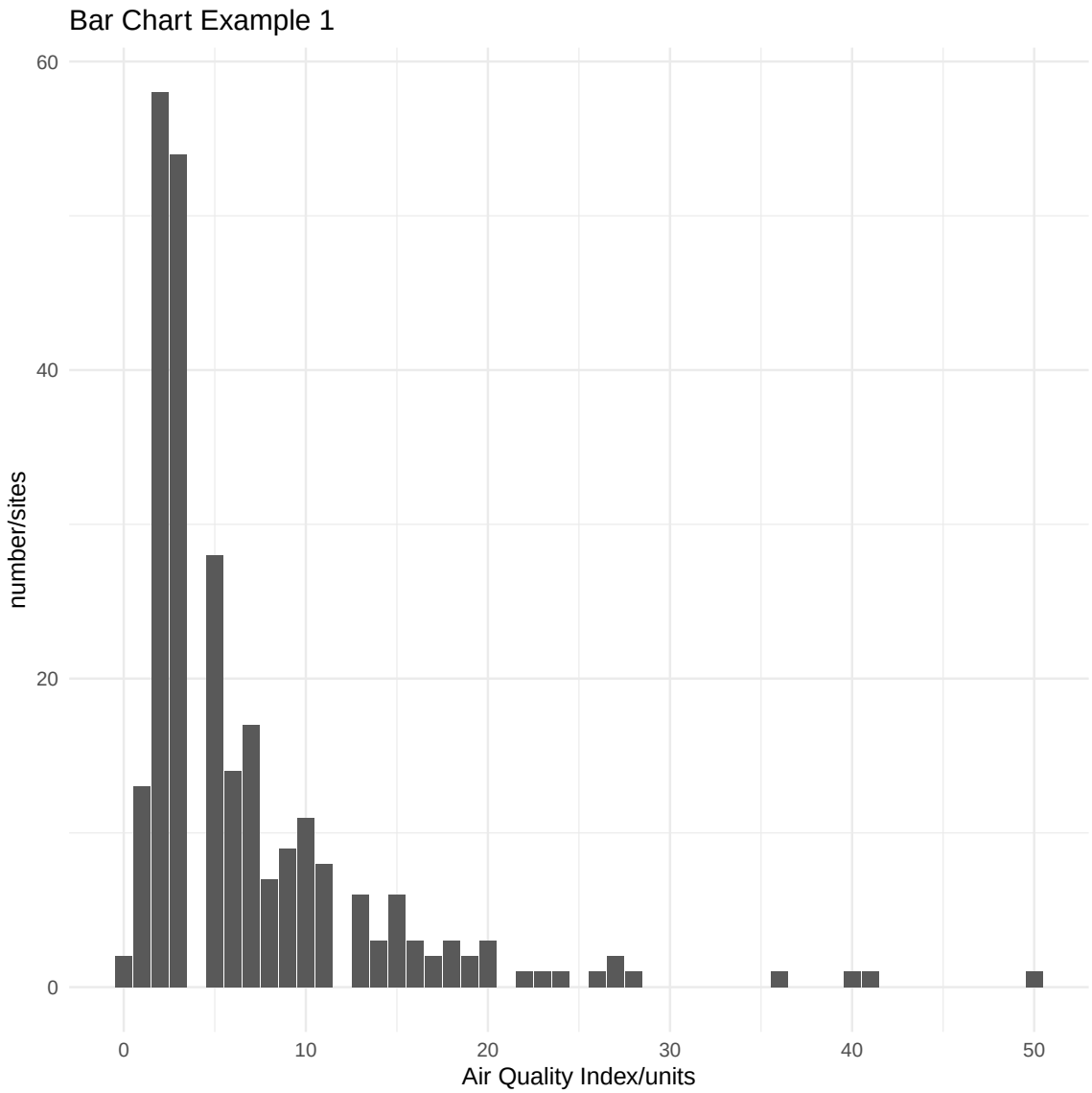
1. **Clarity and Simplicity**: Bar charts are structurally simple, making them easy to read and understand, allowing audiences to quickly grasp key information.
2. **Effective Comparison**: They provide a visual representation that makes comparing the size or value of different categories straightforward, 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.

3.3.3 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.

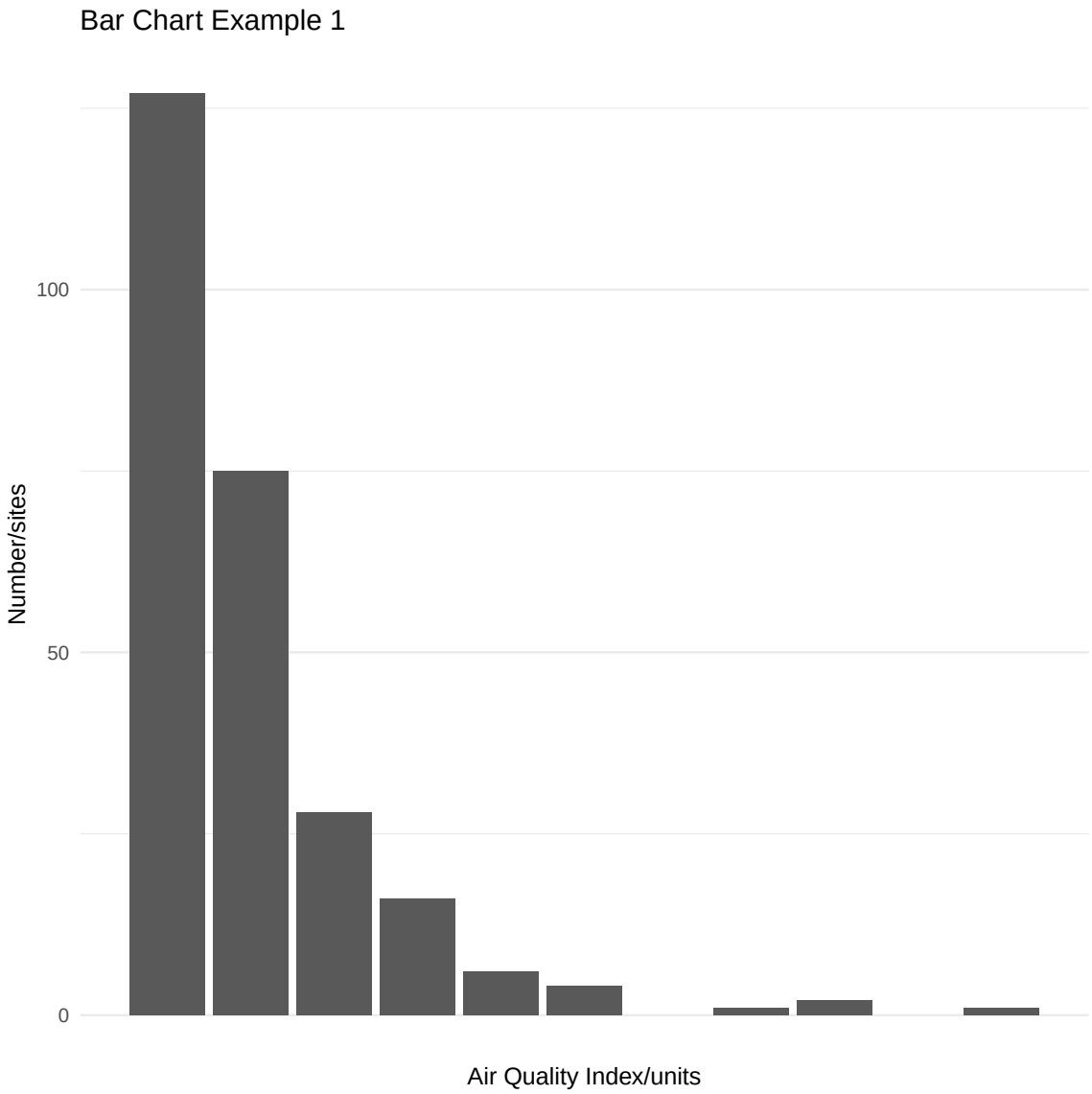
```
data <- read.csv("c4_epa_air_quality.csv")
ggplot(data, aes(x = aqi)) +
  geom_bar() +
  labs(title = "Bar Chart Example 1", x =
         "Air Quality Index/units", y = "number/sites") +
  theme_minimal()
```



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
```

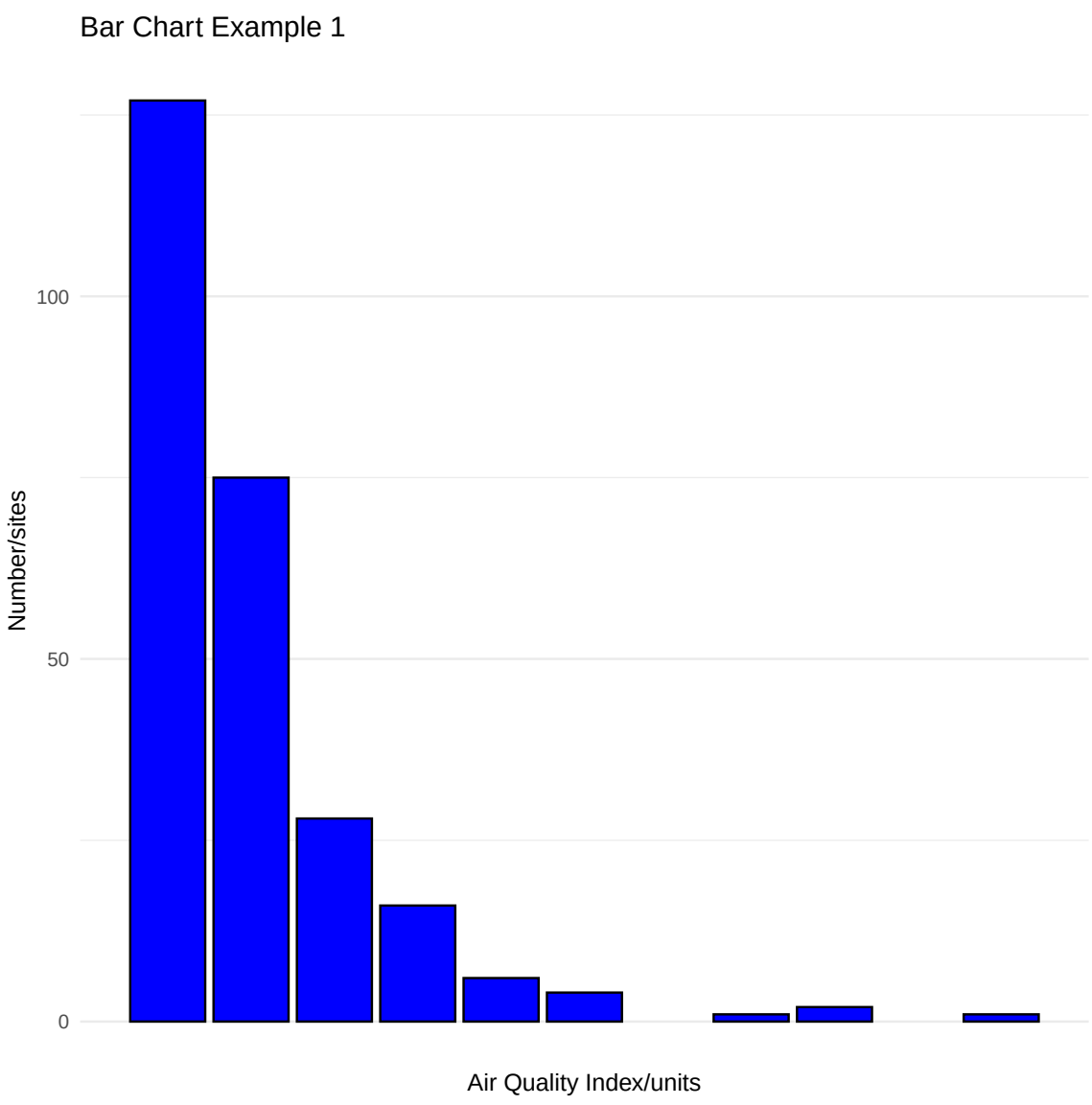
```r
data$group <- cut(data$aqi, breaks =
                    seq(0, max(data$aqi) + 5, by = 5),
                  right = FALSE, include.lowest = TRUE)
data$group <- as.numeric(data$group)

# 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.

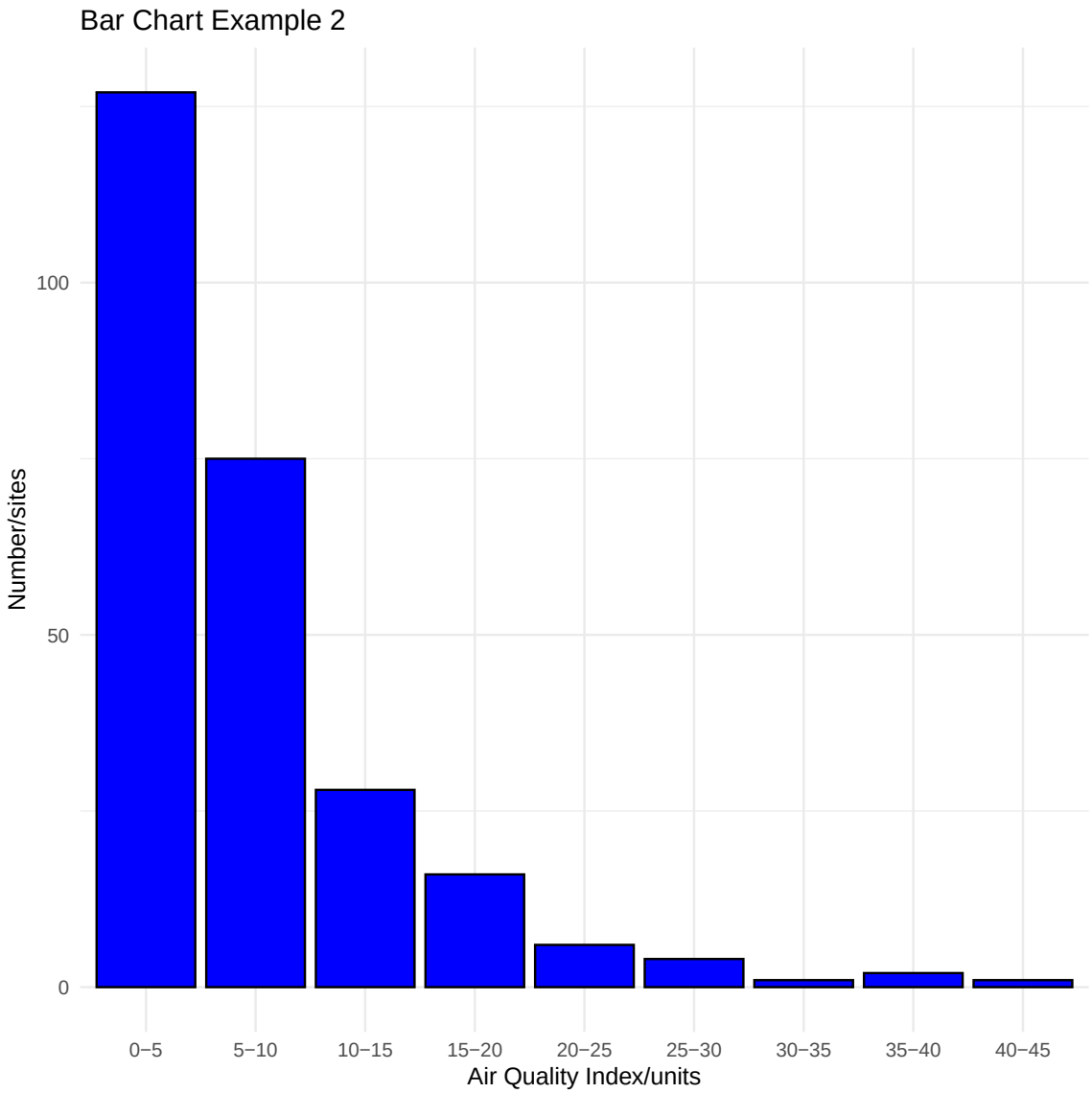
```
# read CSV dater set
data <- read.csv("c4_epa_air_quality.csv")

# 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]))
```



5 Chapter 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.

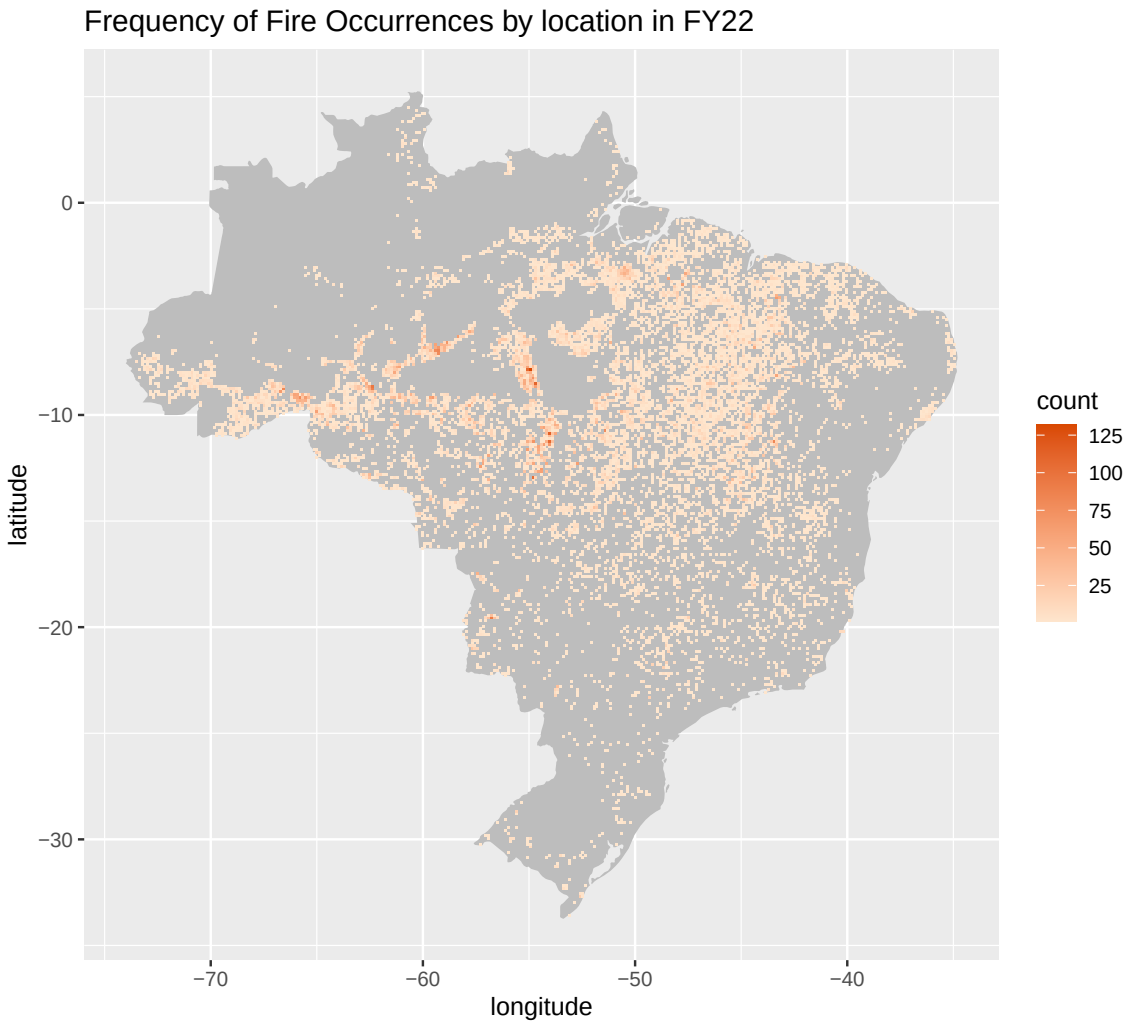
5.1 3.4.1 Heatmaps - Fire in Brazil

The heatmap is a data visualisation technique that uses color 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.

All data obtain from NASA. Each dataset from 2013 to 2022 contains more than 200,000 observations. Across the decade, there are over 3 million observations. The trend is impossible to analyze by eye. However, an exploratory analysis using heatmaps provides insights into this data.

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

# Create the heatmap of fire occurrences
fire_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 = "#d94801") +
  coord_fixed(ratio = 1) +
  labs(title = "Frequency of Fire Occurrences by location in FY22")

print(fire_heatmap)
```

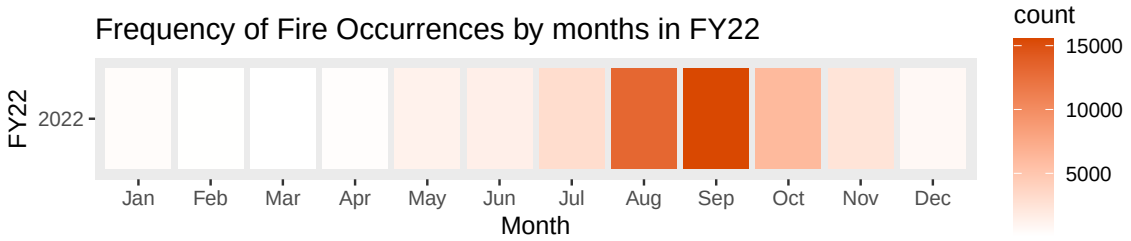


From the heatmap, we can observe that certain locations have significantly higher fires count. However, we do not know the cause of this. Is this due to geographical location, or is it because fires were mostly man-made and used to clear forest areas for agriculture use?

Colour selection from colorbrewer2.

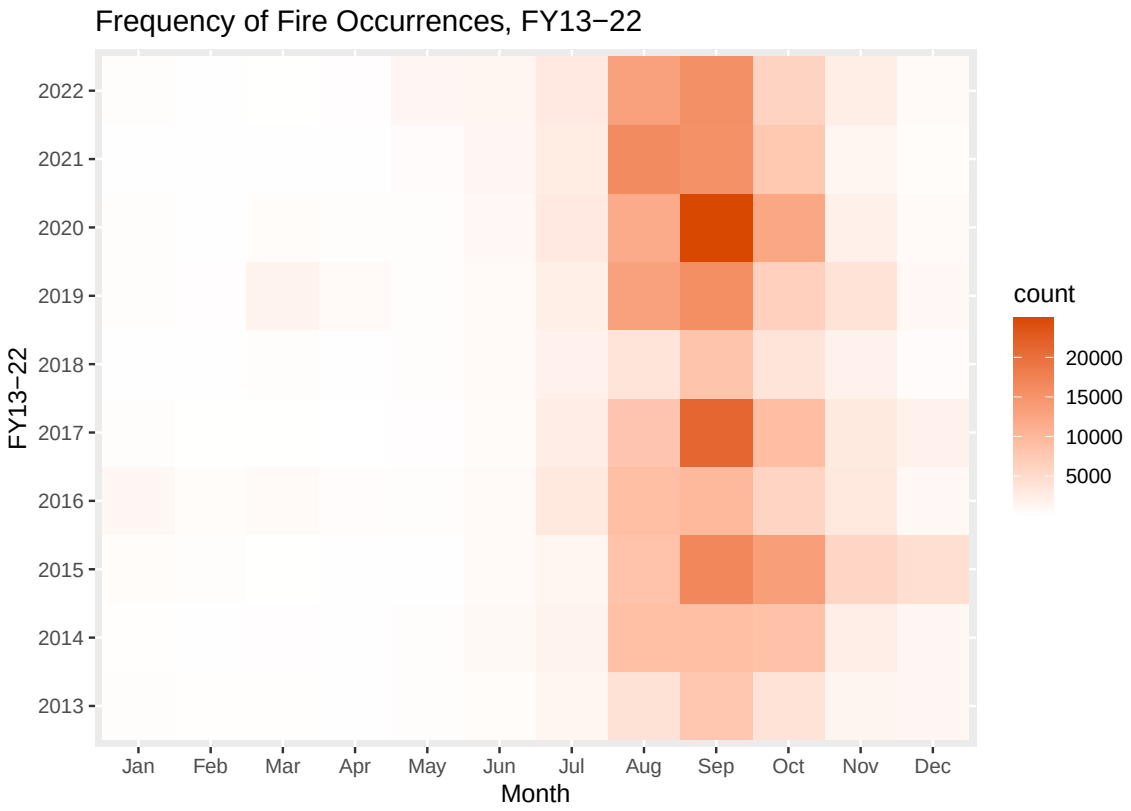
```
heatmap_plot <- ggplot(confident_fire_months_fy22,
                       aes(x = abb_month, y = as.character(2022), fill = count)) +
  geom_tile(width = 0.9, height = 1) +  # Create the heatmap tiles
  scale_fill_gradient(low = "white", high = "#d94801") +
  labs(title = "Frequency of Fire Occurrences by months in FY22",
```

```
       x = "Month", y = "FY22") +
  theme(panel.grid = element_blank() )

print(heatmap_plot)
```



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?

```
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 = "white", high = "#d94801") +
    labs(title = "Frequency of Fire Occurrences, FY13-22",x = "Month", y = "FY13-22")

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.

5.2 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.

5.2.1 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

Chapter 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.

5.3 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.

5.4 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.

5.5 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.

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.

Provide best practices for creating clear and informative line charts.

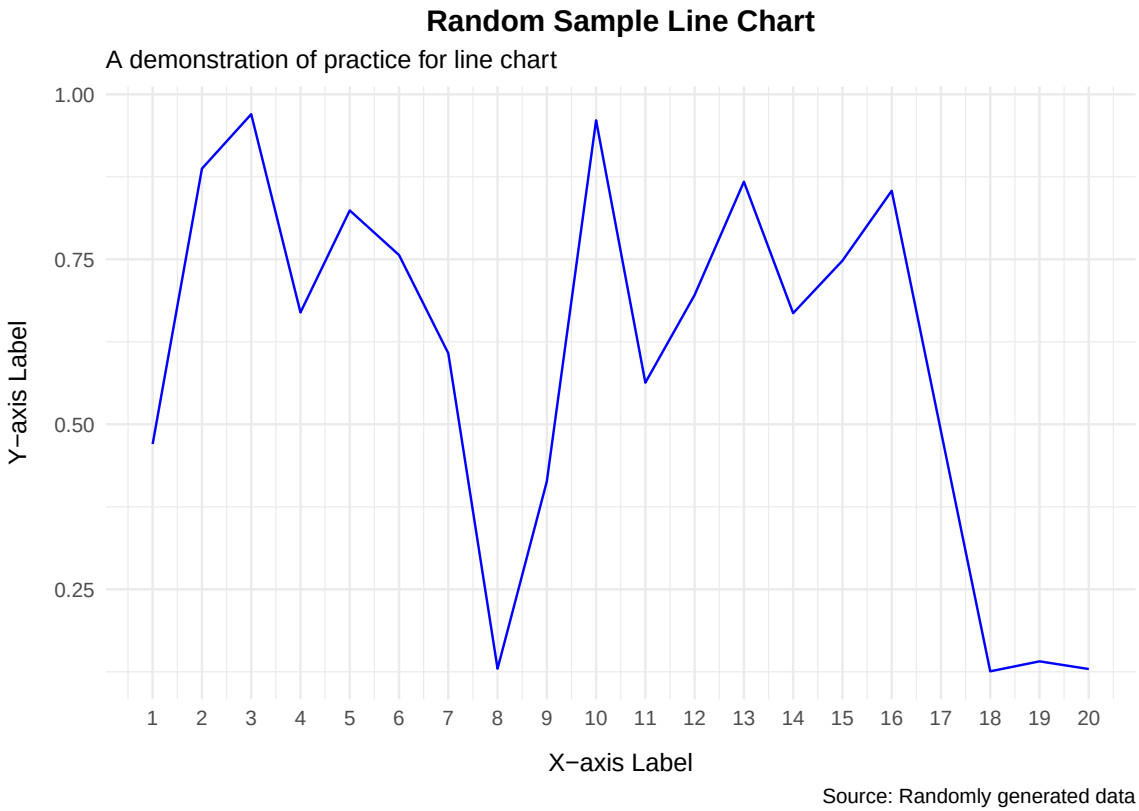
- Title and Labels: Every chart should have a descriptive title and axis labels to clearly convey the purpose of the visualization and the data being shown.
- Use of Colors: Colors should be chosen to clearly differentiate between different lines or data points but also be consistent with the overall 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:



Showcase real-world examples of time series visualisations.

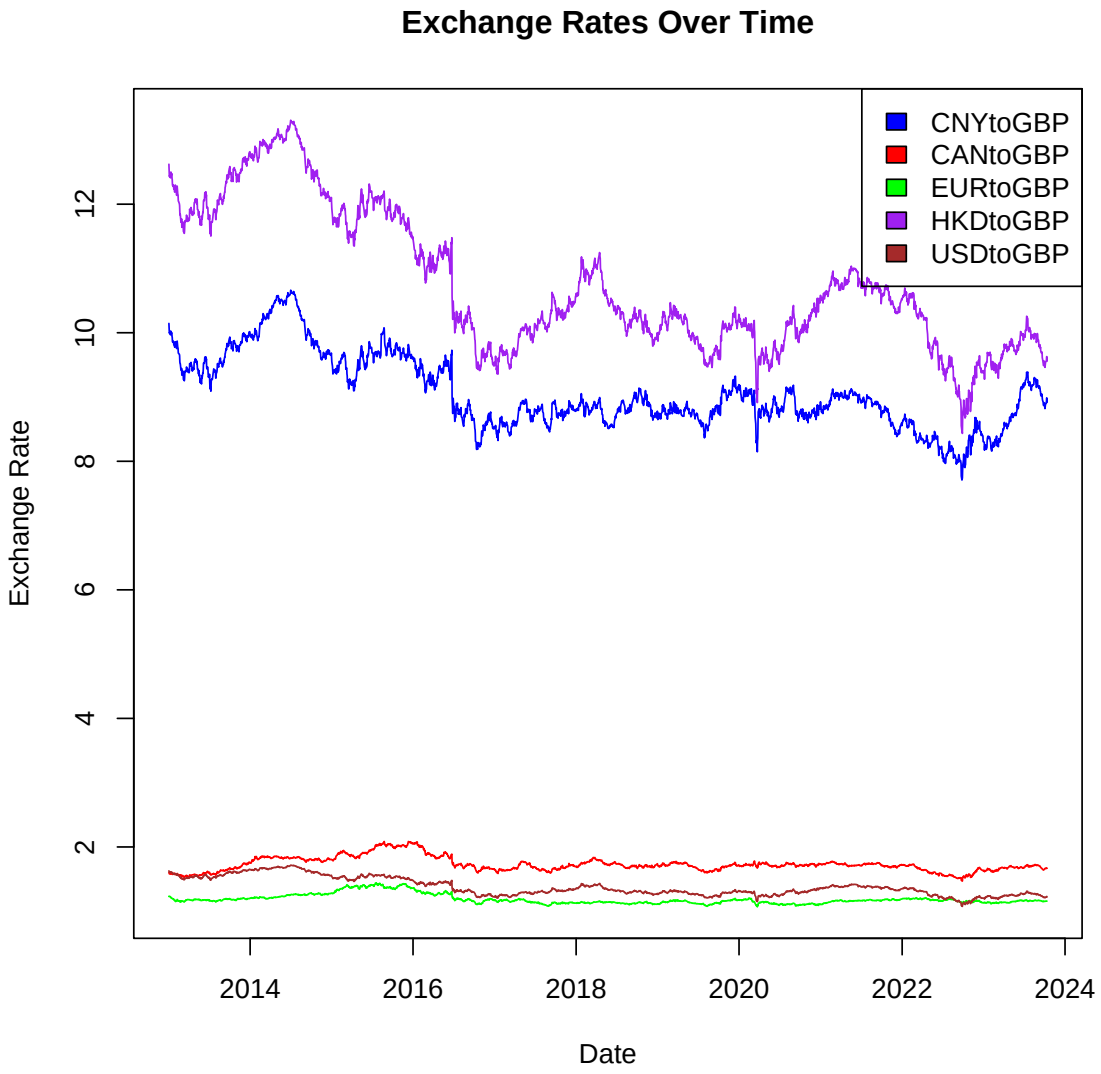
Time series of the daily CNY, CAN, EUR, HKD, USD versus GBP exchange reference rate data published by the European Central Bank over the time period from 01 Jan 2013 to 12 Oct 2023 (without weekends). The exchange rate tells you how many pounds you need to buy/sell 1 CNY, CAN, EUR, HKD, USD.

5.6 The data set has the format as below:

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

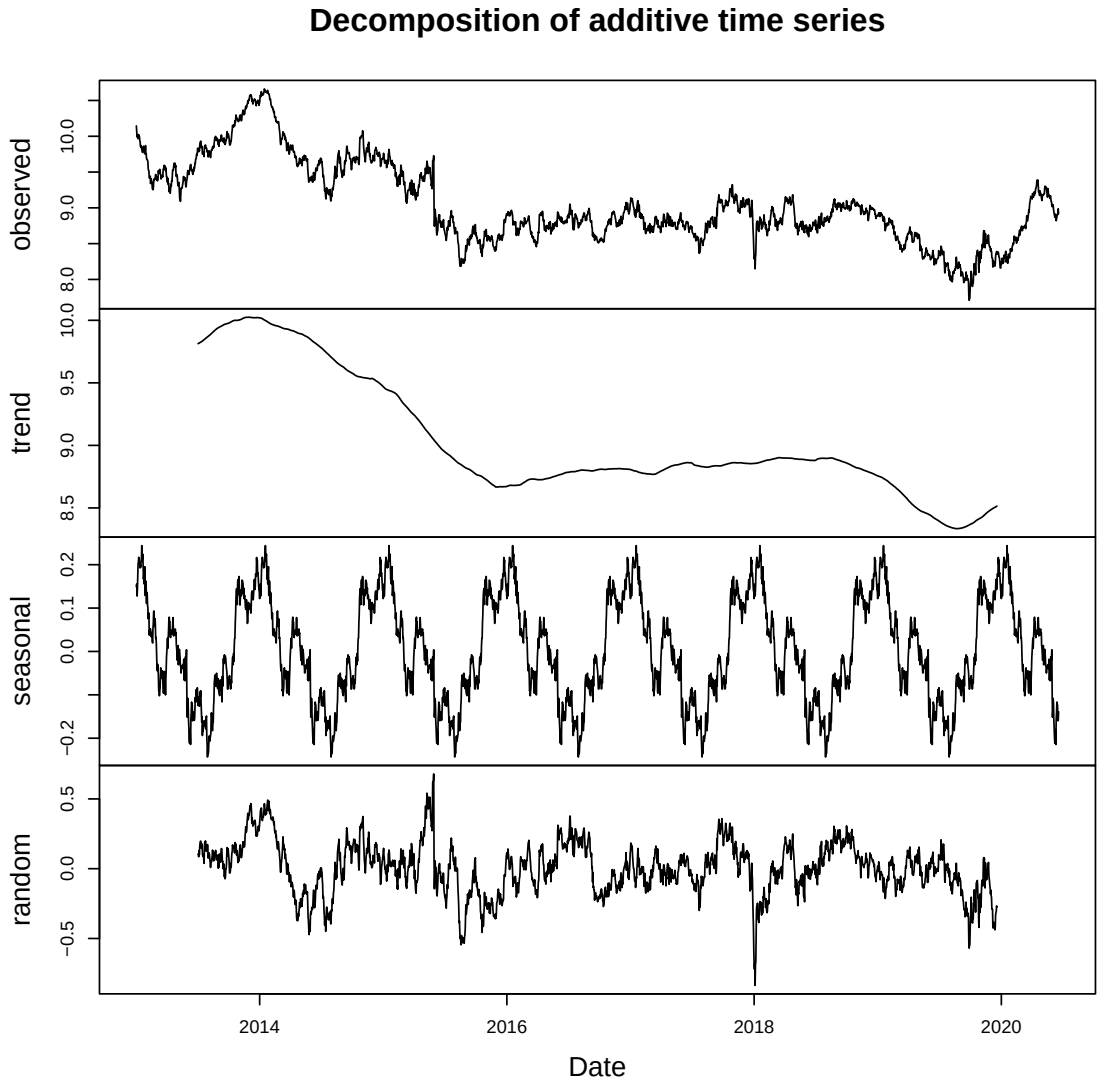
Table 1: Field Information: CNY, CAN, EUR, HKD, USD to GBP

5.7 Multiple time series in one plot:



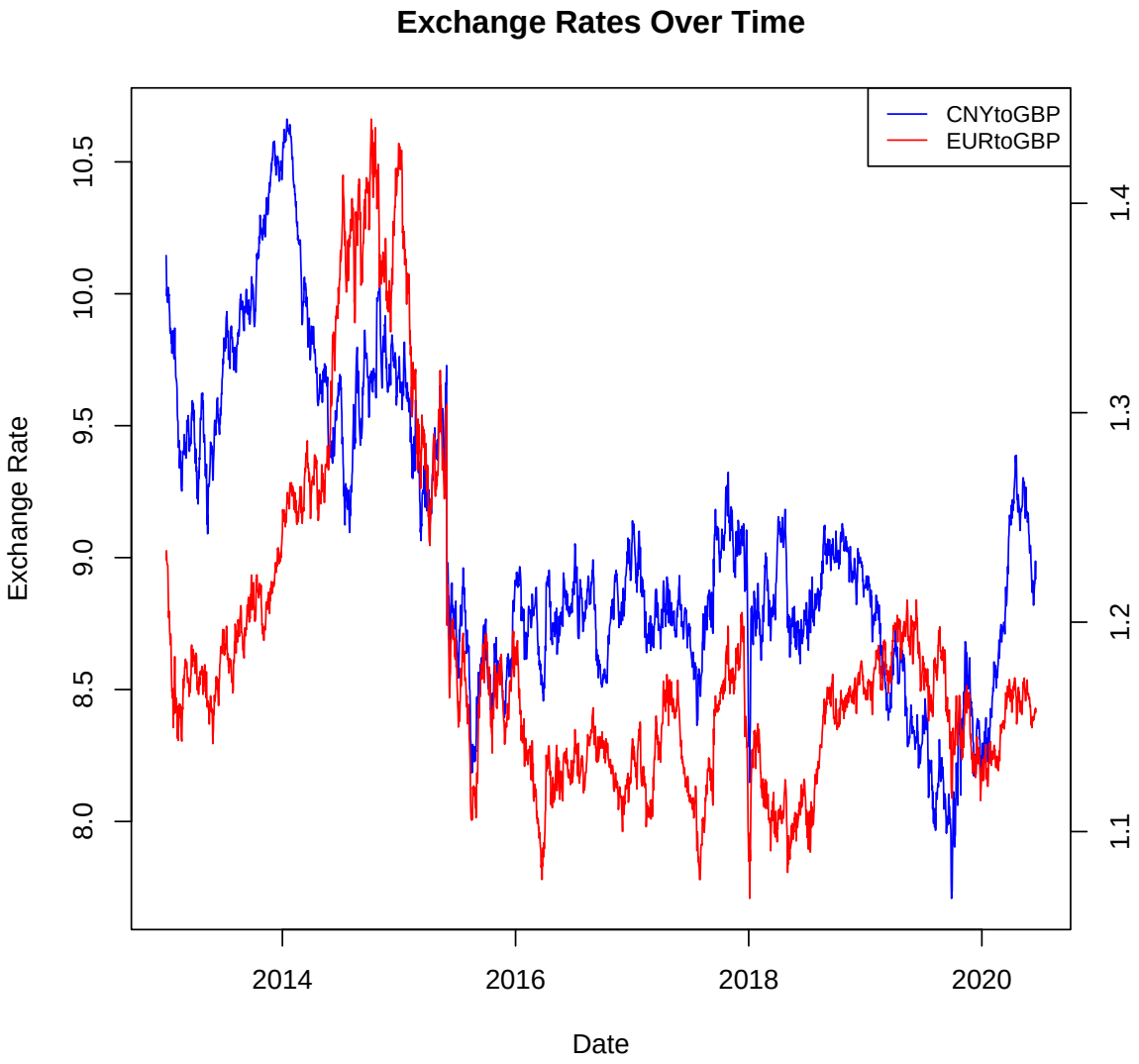
5.8 Decomposition of one time series into trend, seasonal, and random.

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.



5.9 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.



6 Chapter 3.5 Network Graphs and Sankey Diagrams

xxx

7 Chapter 3.6 Geographic Maps and Spatial Data Visualisation

xxx

8 Chapter 3.7 3D and Interactive Visualisations

xxx

9 Chapter 3.8 Advanced Visualisation Techniques

xxx