PROJECT: Analysis and Prediction of Airbnb Listing Prices

BOARD INFINITY

A training report

Submitted in partial fulfillment of the requirements for the award of degree of

B.tech

(Computer science & engineering)

DATA SCIENCE

Submitted to

LOVELY PROFESSIONAL UNIVERSITY PHAGWARA, PUNJAB

From 05/06/23 to 12 /07/23

SUBMITTED BY

Name of student: ROSHAK KUMAR
Registration Number: 12203460

Signature of the student: Roshak

DECLARATION

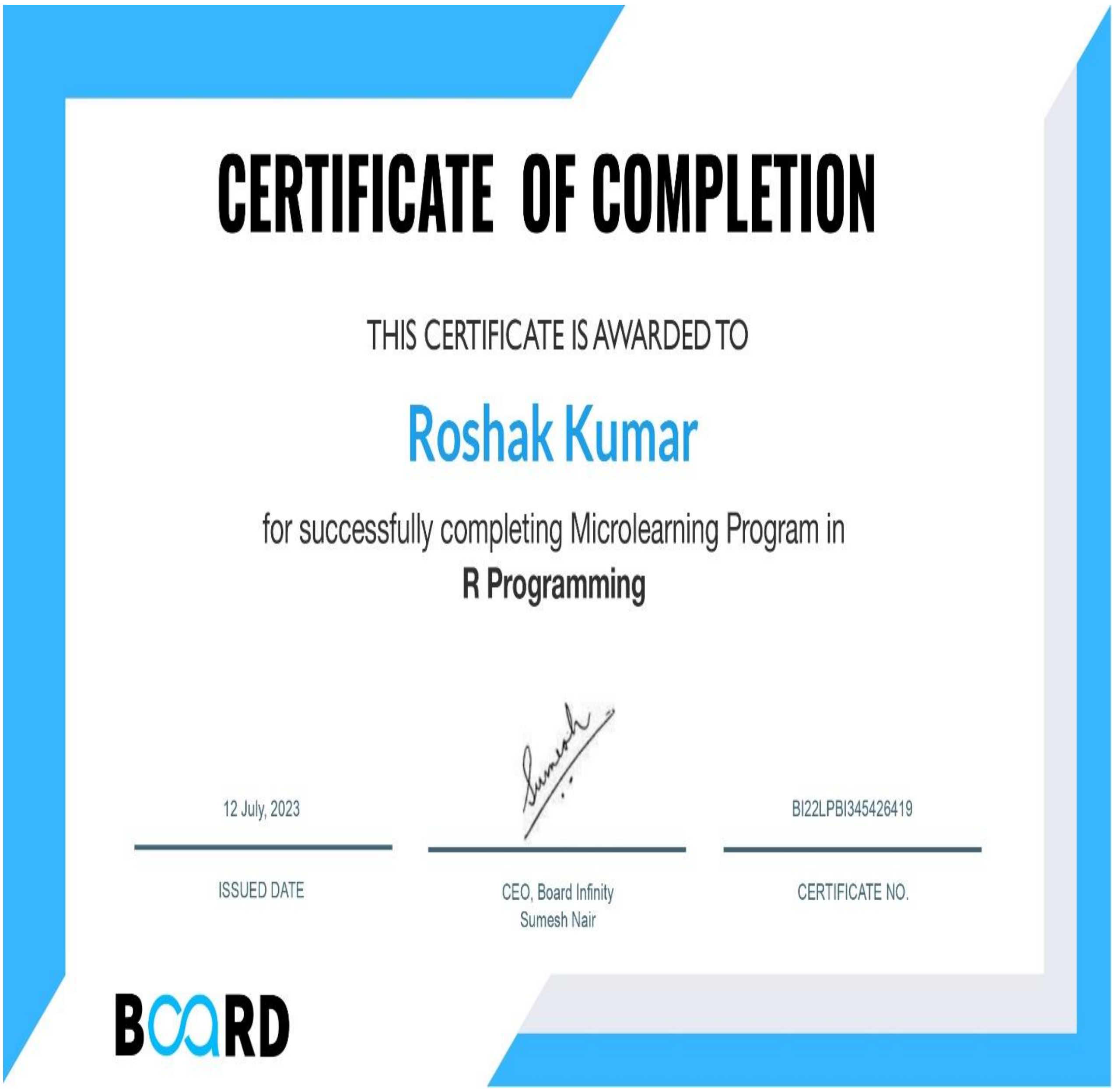
I, **ROSHAK KUMAR, 12203460,** hereby declare that the work done by me on "**Analysis and Prediction of Airbnb Listing Prices**" from **June, 2023** to **July, 2023**, is a record of original work for the partial fulfillment of the requirements for the award of the degree,**Btech(cse).**

Name of the Student (Registration Number) : ROSHAK KUMAR (12203460)

Signature of the student : Roshak

Dated: 20 August 2023

CERTIFICATE



CERTIFICATE OF COMPLETION

THIS CERTIFICATE IS AWARDED TO

Roshak Kumar

for successfully completing Microlearning Program in

R Programming

12 July, 2023

ISSUED DATE

Sumesh

CEO, Board Infinity
Sumesh Nair

BI22LPBI345426419

CERTIFICATE NO.

BOARD

ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to the following individuals and resources, without whom this project would not have been possible:

Board infinity faculty: I am deeply thankful for the guidance, expertise, and valuable insights provided throughout the entire duration of this project. Your mentorship has been instrumental in shaping the direction and quality of this analysis.

Board infinity : My appreciation goes to board infinity for their collaborative efforts in data collection, cleaning, and exploratory analysis. Their dedication significantly enriched the project's outcomes.

Airbnb: I acknowledge Airbnb for providing the Airbnb dataset used in this project. The availability of this comprehensive dataset was pivotal in conducting meaningful analyses and building accurate predictive models.

R Package Contributors: I want to express my thanks to the developers of various R packages, which played a critical role in data manipulation, visualization, and modeling tasks. These packages streamlined the coding process and enhanced the project's efficiency.

Online Communities and Forum: The insights and solutions gained from online programming communities and forums were invaluable in overcoming technical challenges. The willingness of members to share their knowledge greatly expedited problem-solving.

Family and Friends: My heartfelt gratitude goes to my family and friends for their unwavering support, patience, and encouragement throughout this journey. Your belief in my abilities motivated me to push the boundaries of this project.

This project represents the collaborative efforts of these remarkable individuals and resources. Their contributions have left an indelible mark on the quality and success of the "Analysis and Prediction of Airbnb Listing Prices" project.

LEARNING OUTCOMES

Engaging in a project like "Analysis and Prediction of Airbnb Listing Prices" using R programming can provide you with valuable learning outcomes that extend beyond just technical skills. Here are some potential learning outcomes you can expect from such a project:

1. Data Manipulation and Cleaning : You'll learn how to handle real-world datasets, deal with missing values, outliers, and inconsistent data, and use R's data manipulation packages like dplyr and tidyr to preprocess data effectively.

2. Exploratory Data Analysis (EDA): You'll gain experience in visualizing data using various plots and graphs to uncover patterns, trends, correlations, and potential insights within the data.

3. Feature Engineering: You'll learn how to engineer new features from existing data to enhance model performance, improving your understanding of domain-specific factors that affect the prediction task.

4. Machine Learning Algorithms: Implementing machine learning algorithms such as linear regression, decision trees, and random forests will give you hands-on experience in selecting and fine-tuning models for regression tasks.

5. Model Evaluation : You'll become familiar with evaluating model performance using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared, helping you understand how to assess the quality of predictions.

6. Statistical Analysis : Through analyzing the relationships between different variables and their impact on Airbnb listing prices, you'll gain insights into the application of statistical concepts.

7. Programming Proficiency : Your R programming skills will improve as you write code to preprocess data, create visualizations, develop models, and generate reports.

8. Data Visualization : Creating informative and meaningful visualizations will enhance your ability to communicate complex findings effectively to both technical and non-technical audiences.

9. Project Management : Managing a complete data science project from data collection to model deployment will teach you project planning, organization, and time management.

10. Critical Thinking : You'll develop the ability to critically assess the validity of insights derived from your analysis and understand when the results are actionable.

11. Domain Knowledge : As you delve into the Airbnb market, you'll gain domain-specific knowledge about factors affecting listing prices, such as location, property type, amenities, and seasonality.

12. Problem Solving : You'll encounter challenges and roadblocks during the project, allowing you to develop problem-solving skills by finding creative solutions to overcome obstacles.

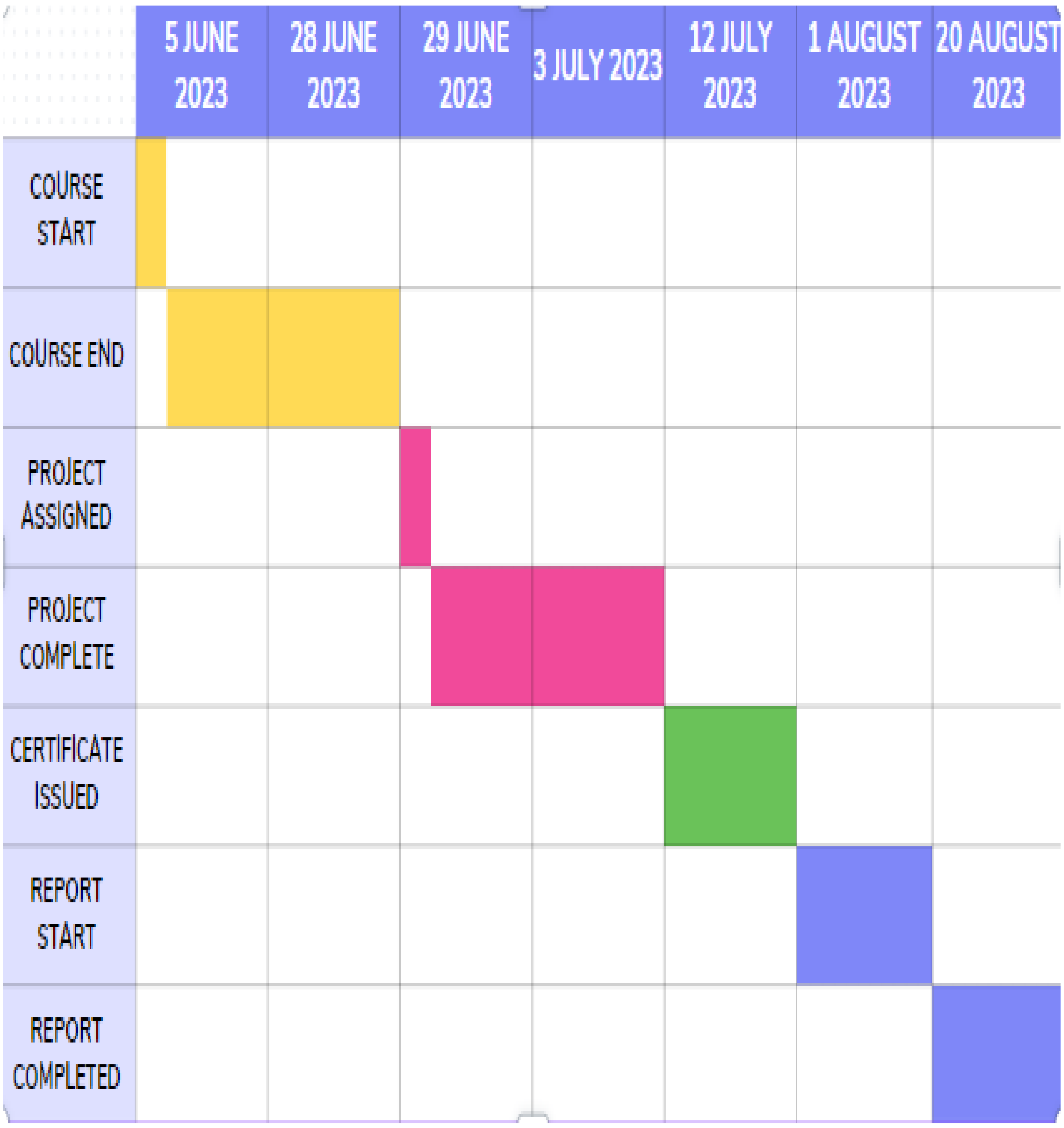
13. Communication Skills : Presenting your findings, insights, and predictions to different audiences will enhance your ability to communicate complex technical concepts clearly and concisely.

14. Reproducibility : Learning to document your code and analysis process using tools like R Markdown will teach you about reproducibility and the importance of well-documented work.

15. Ethical Considerations : You'll become aware of ethical considerations related to data privacy, responsible data usage, and potential biases in your analysis.

Overall, working on the "Analysis and Prediction of Airbnb Listing Prices" project in R programming will provide you with a comprehensive learning experience that combines technical skills with critical thinking, problem-solving, and effective communication.

GANTT CHART



	5 JUNE 2023	28 JUNE 2023	29 JUNE 2023	3 JULY 2023	12 JULY 2023	1 AUGUST 2023	20 AUGUST 2023
COURSE START	🟨						
COURSE END	🟨	🟨					
PROJECT ASSIGNED			🟥				
PROJECT COMPLETE			🟥	🟥			
CERTIFICATE ISSUED					🟩		
REPORT START						🟦	
REPORT COMPLETED							🟦

ABOUT ORGANISATION

BOARD INFINITY

Board Infinity is a full-stack career platform for students and jobseekers enabled by personalized learning paths, career coaches and access to opportunities. Our mission is to personalize your career journey, help you realise true potential and meet your career dreams. Be it a career transition, your first job, campus placements preparation or any career guidance. Board Infinity is a one-stop solution to all your career needs. We connect career aspirants with industry experts for focused learning, guidance, mentoring and support. We also prepare you and connect to relevant opportunities and help you realize your career dreams. Students ∞ Coaches - Enable meaningful connections between students and coaches. Colleges ∞ Employers - Coach and train college students and provide employers with business ready talent. Students ∞ Employers - Connect talent to the right opportunities. Board Infinity is a holistic platform connecting career aspirants, institutes to top industry coaches and companies for career guidance, focused training and opportunities.

Website

http://www.boardinfinity.com/

Industries

E-Learning Providers

Company size

51-200 employees

Headquarters

Vashi, Navi Mumbai , Maharashtra

Type

Educational

Founded

2017

Specialties

Career Coaching, Management Training, Placement Opportunities, Career Discovery, Personalized coaching, Recruitment, Online Training, and Online Courses

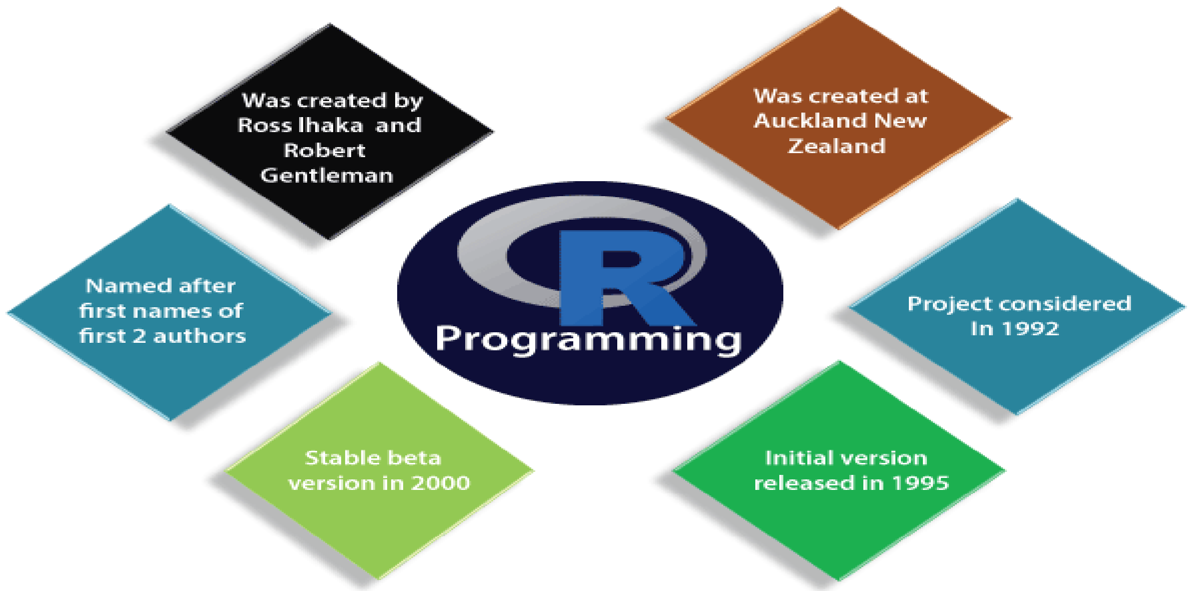
CHAPTER-1

INTRODUCTION OF THE PROJECT UNDERTAKEN

Before giving the introduction about my project I will let you know about the R PROGRAMMING language which is the root of this project. As whole of the project is based upon this language as I have opted the course of R programming provided by BOARD INFINITY.

Introduction about R programming language

R is a programming language for statistical computing and graphics supported by the R core team and the R foundation for statistical computing. Created by statisticians Ross lhaka and Robert Gentleman ,R is used among data miners , bioinformatics and statisticians for data analysis and developing statistical software .R is a open source programming language that is widely used as a statistical software and data analysis tools.


Was created by Ross Ihaka and Robert Gentleman
Was created at Auckland New Zealand
Named after first names of first 2 authors
Programming
Project considered In 1992
Stable beta version in 2000
Initial version released in 1995

INTRODUCTION OF PROJECT

PROJECT: Analysis and Prediction of Airbnb Listing Prices

The analysis and prediction of Airbnb listing prices is a fascinating and practical application of data science and statistical modelling. As the popularity of airbnb and other short-term rental platforms continues to grow ,understading the factor that influence listing prices and being able to predict those prices accurately becomes crucial for both hosts and guests. In this project I have used the R programming language to analyse airbnb listing data, Identify key features that affect pricing and build predictive models to estimate the prices of new listings.

OBJECTIVE

The goal of this project is to perform an exploratory data analysis (EDA) and predictive modelling on Airbnb listing data using the R programming language. The project will encompass all stages of the data science lifecycle, from data import and cleaning to visualisation and modelling.

DATA

I took the data from Airbnb's official website where they provide publicly available information about a city's Airbnb's listings. The data includes details such as the listing id, name, host id, host name, neighbourhood, latitude, longitude, room type, price, minimum nights, number of reviews, last review date, reviews per month, calculated host listings count, availability per year etc.

Project Steps:

1. Data Importing

- Import the Airbnb data using readr or other relevant packages. This may include .csv files or other formats.

2. Data Cleaning and Transformation

- Use dplyr and tidyr to clean the data and prepare it for analysis. This may include handling missing values, outliers, or erroneous data.
- Transform the data as necessary for analysis. This may include creating new variables, recoding variables, or restructuring the data.

3. Exploratory Data Analysis

- Use various R functions and packages to explore the data. This can include summary statistics, correlations, and distributions.
- Create visualizations using R's plotting capabilities. This can include scatter plots, boxplots, histograms, etc.

SCOPE

The scope of the analysis and prediction of airbnb listing prices in R programming is broad and encompasses several aspects of data science , statistics , and machine learning. This project can provide valuable insights for both hosts and guests on the airbnb platform.

The scope of the project can vary based on the depth of analysis and the specific goals of the analysis and prediction task. It offers opportunities to apply a wide range of data science techniques and tools available in R, making it a valuable and practical endeavour for anyone interested in the airbnb ecosystem, pricing strategies, and predictive modelling.

RELEVANCE

The analysis and prediction of Airbnb listing prices using R programming have significant relevance and practical applications in various contexts. Here are some key points highlighting the relevance of this endeavor:

Host Decision-Making: Hosts on Airbnb can benefit from accurate price predictions to optimize their listing prices. A data-driven approach helps hosts set competitive prices that attract guests while maximizing their revenue. By understanding the factors that influence pricing, hosts can adjust amenities, property features, and other attributes to align with market demand.

Guest Booking Decisions: For potential guests, having insights into estimated listing prices helps them make informed booking decisions. Predictive models can assist travelers in planning their trips and budgeting for accommodations, enhancing their overall booking experience.

Market Insights: The analysis of Airbnb listing prices contributes to a better understanding of the dynamics of the short-term rental market. This includes identifying pricing trends, seasonal variations, and how different factors impact prices across different locations. Such insights are valuable for investors, real estate professionals, and researchers interested in the sharing economy and hospitality industry.

Optimal Pricing Strategies: The project enables hosts to develop optimal pricing strategies based on demand fluctuations, local events, and competitor pricing. By continuously analyzing and adapting to market conditions, hosts can improve occupancy rates and revenue.

Competitive Advantage: Hosts who leverage data analysis and prediction models have a competitive advantage in a crowded marketplace. They can position their listings strategically by adjusting prices in real time to attract more bookings and stand out from competitors.

Personalized Experiences: Predictive modeling can contribute to personalized guest experiences. Hosts can offer tailored amenities or pricing based on predicted preferences, enhancing customer satisfaction and loyalty.

Real Estate Investments: Investors considering purchasing properties for short-term rentals can use predictive models to estimate potential rental income. This informs their investment decisions and helps them evaluate the profitability of different properties.

Regulatory Compliance: In locations with regulations or restrictions on short-term rentals, accurate pricing predictions can help hosts ensure compliance with pricing caps or other legal requirements.

Data Science Skill Development: This project provides individuals with an opportunity to apply and enhance their data science, statistical analysis, and machine learning skills using R. It serves as a practical exercise in data manipulation, visualization, model building, and interpretation.

Educational and Research Purposes: The project can serve as a valuable case study for educational institutions and researchers interested in teaching or studying data analysis, predictive modeling, and the sharing economy.

Innovation and Technology: By deploying predictive models through web applications or APIs, hosts and guests can access pricing predictions conveniently, embracing the technological advancements in the hospitality industry.

In essence, the analysis and prediction of Airbnb listing prices in R programming have wide-ranging relevance, impacting hosts, guests, investors, researchers, and the broader hospitality sector. It empowers stakeholders with data-driven insights that lead to better decision-making, improved customer experiences, and enhanced market competitiveness.

IMPORTANCE AND APPLICABILITY

The importance and applicability of analyzing and predicting Airbnb listing prices using R programming are significant and extend to various domains and stakeholders. Here's a closer look at the key importance and applicability of this endeavor:

1. **Informed Decision-Making for Hosts**

Optimized Pricing : Accurate price predictions empower Airbnb hosts to set competitive and attractive prices for their listings. This can lead to increased occupancy rates and revenue.

Customized Strategies : Hosts can tailor their pricing strategies based on demand patterns, local events, and property attributes, enhancing their overall business performance.

2. Enhanced Guest Experience

Budget Planning : Potential guests can make informed booking decisions by estimating the cost of their stay. This contributes to a better travel planning experience and increases customer satisfaction.

Value Assessment : Guests can assess the value of accommodations based on predicted prices and amenities, making more confident booking choices.

3. Real Estate Investment Evaluation

Profitability Assessment: Property investors considering short-term rentals can evaluate the potential return on investment (ROI) and assess the financial viability of acquiring or managing properties on platforms like Airbnb.

4. Market Insights and Research:

Pricing Trends : Analysis of historical data and price predictions offer insights into pricing trends, seasonality, and market dynamics, aiding researchers and analysts in understanding the short-term rental market.

5. Academic and Educational Purposes

Teaching Tool : This project serves as an excellent educational case study for data science, predictive modeling, and applied statistics courses.

Skill Development : Students and learners can practice and develop data analysis, visualization, and machine learning skills using real-world data and scenarios.

6. Regulatory Compliance

Legal Considerations : Accurate price predictions can help hosts ensure that their pricing complies with local regulations and restrictions on short-term rentals.

7. Data-Driven Insights

Understanding Impactful Factors : The analysis can reveal the key attributes that significantly influence listing prices, aiding hosts in making strategic decisions about property improvements or marketing efforts.

In summary, the analysis and prediction of Airbnb listing prices using R programming offer a range of practical benefits and applications across the Airbnb ecosystem. From improving the financial performance of hosts to aiding real estate investors and contributing to market research, this endeavor leverages data science to enhance decision-making, customer experiences, and business strategies in the short-term rental industry.

CHAPTER 2

VISUALIZATION AND MODELING

Visualization and modeling of data using R are essential components of data analysis and data science. R is a powerful programming language and environment that provides a wide range of tools for creating visualizations and building predictive models. Here's an overview of how you can perform visualization and modeling of data using R:

Data Visualization in R

1.Data Exploration and Summary

Use functions like `summary()` and `str()` to get an overview of your data's structure and basic statistics.

Create histograms, density plots, and box plots to understand the distribution of numerical variables using functions like `hist()`, `density()`, and `boxplot()`.

2. Scatter Plots and Correlation

Generate scatter plots to visualize relationships between two numerical variables using `plot()` or specialized packages like `ggplot2`.

Calculate and visualize correlations between variables using functions like `cor()` and correlation matrices.

3.Categorical Data Visualization

Create bar plots, pie charts to visualize categorical data and their proportions using functions like `barplot()`, `pie()`.

4.Time Series Visualization

Visualize time series data using line plots, area plots, or specialized time series plots from packages like `ggplot2` or `lubridate`.

5. Advanced Visualization

Utilize the `ggplot2` package for creating highly customizable and publication-quality visualizations.

CHAPTER 3

EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis (EDA) is a critical step in the data analysis process that involves visually and statistically summarizing, understanding, and exploring the main characteristics of a dataset. In R, there are several packages and functions that facilitate EDA. Here's a step-by-step guide to performing exploratory data analysis in R:

1. Load Required Packages:

```
library(dplyr)   # Data manipulation
library(ggplot2) # Data visualization
```

2. Load and Inspect the Data:

Load your dataset using functions like read.csv(), read_excel(), or any other appropriate function. Once loaded, examine the structure of the dataset using functions like str() and head()

```
data <- read.csv("your_data.csv")
str(data)
head(data)
```

3. Summary Statistics:

Generate summary statistics to get an overview of the central tendency, dispersion, and other key statistics of the variables.

```
summary(data)
```

4. Data Visualization:

a. Univariate Analysis:

Visualize the distribution of individual variables.

b. Bivariate Analysis:

Explore relationships between pairs of variables.

5. Missing Data Analysis:

Investigate missing values in the dataset.

6. Outlier Detection:

Identify potential outliers

7. Data Transformation:

Perform data transformations if needed (e.g., log transformation, normalization).

8. Multivariate Analysis:

Explore relationships among multiple variables.

These are just some of the key steps and techniques you can use for exploratory data analysis in R. The goal of EDA is to gain insights into the data, identify patterns, and generate hypotheses for further analysis. The specific techniques you choose will depend on the nature of your data and the questions you aim to answer.

CHAPTER 4

FIGURES AND CHARTS

BOXPLOT

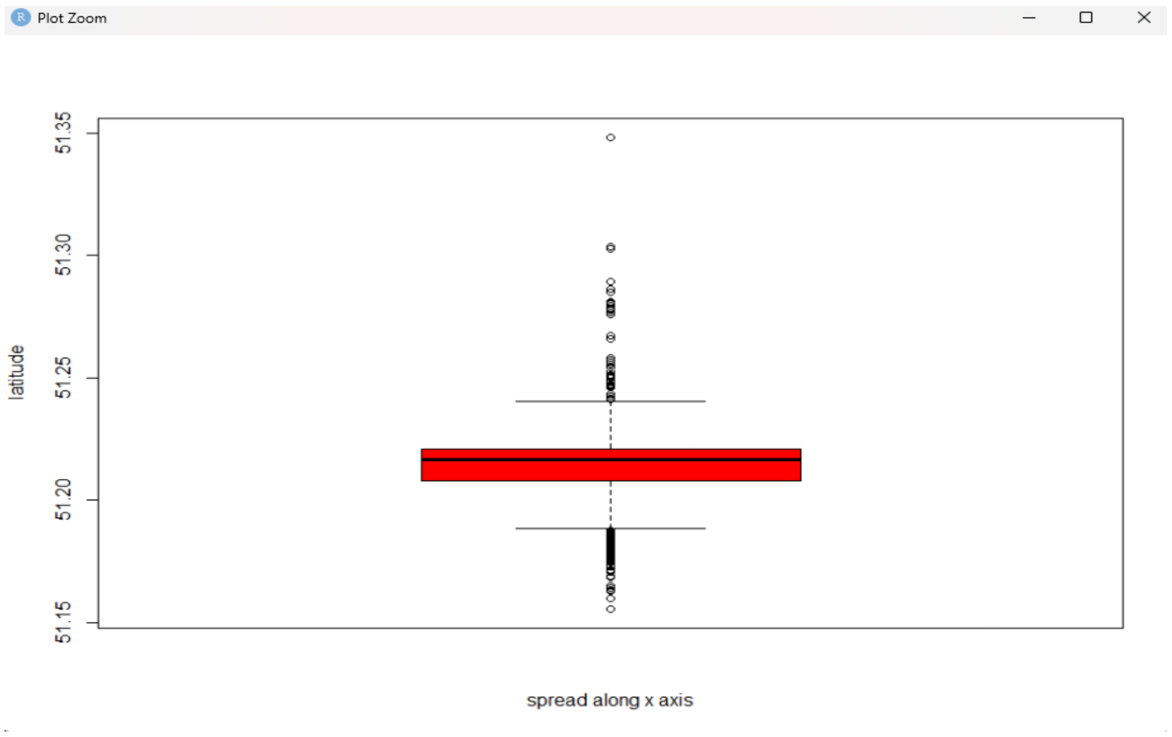
In R programming, a box plot (also known as a box-and-whisker plot) is a graphical representation of the distribution of a dataset. It provides a visual summary of the central tendency, spread, and potential outliers in the data. A box plot displays the data's quartiles, median, and any potential outliers in a concise manner.

Here's how a box plot is constructed and what its components represent:

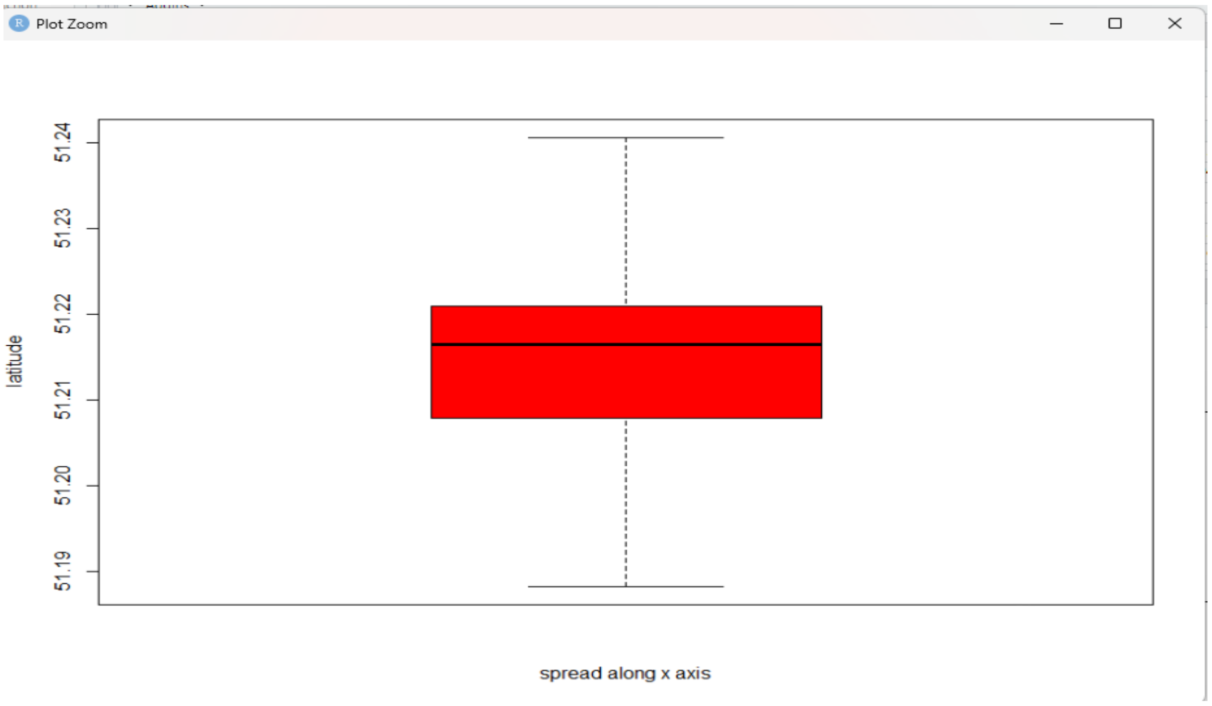
Box: The box in the plot represents the interquartile range (IQR), which is the range between the first quartile (Q1) and the third quartile (Q3). The box's vertical line inside it represents the median (Q2), which is the value that separates the lower and upper halves of the dataset.

Whiskers: The whiskers extend from the edges of the box to the minimum and maximum values within a certain range. The whiskers are often a multiple of the IQR, such as 1.5 times the IQR.

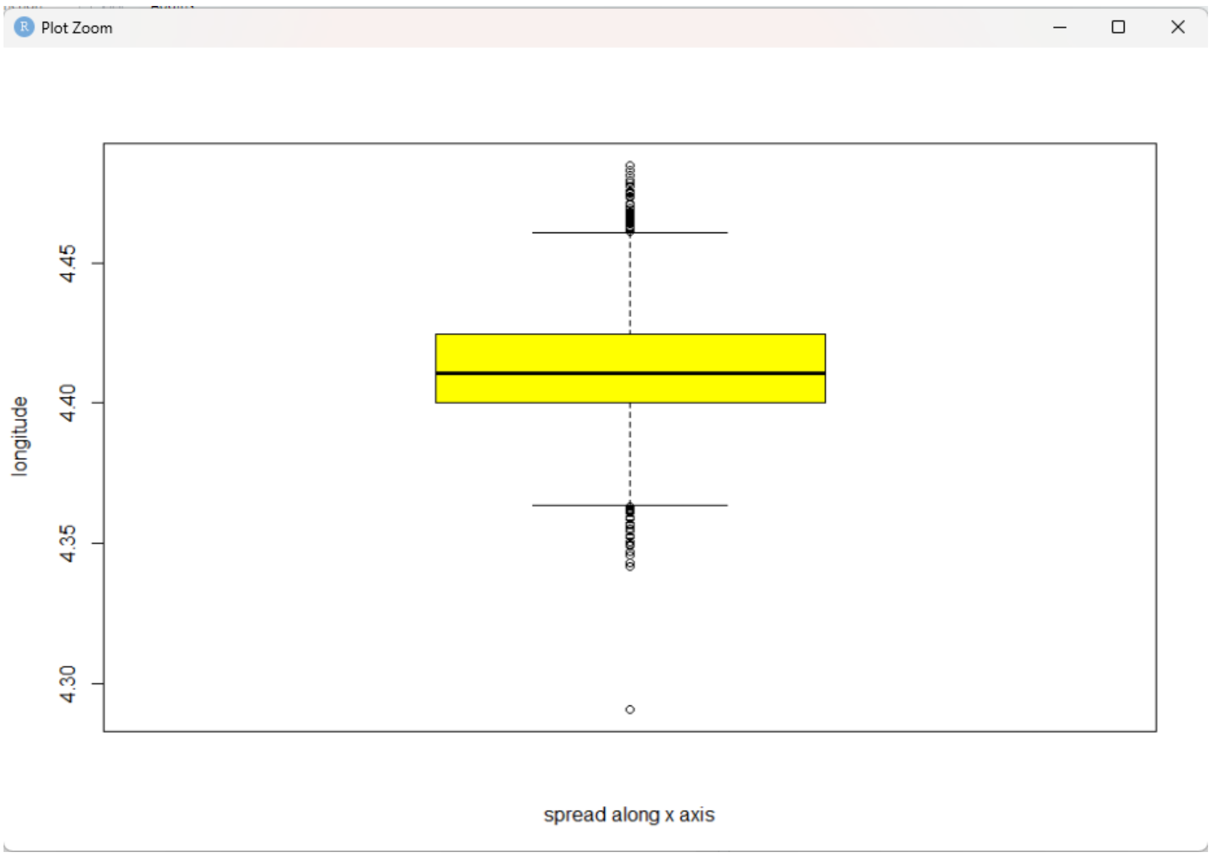
Outliers: Data points that fall outside the whiskers are considered potential outliers and are represented as individual points.



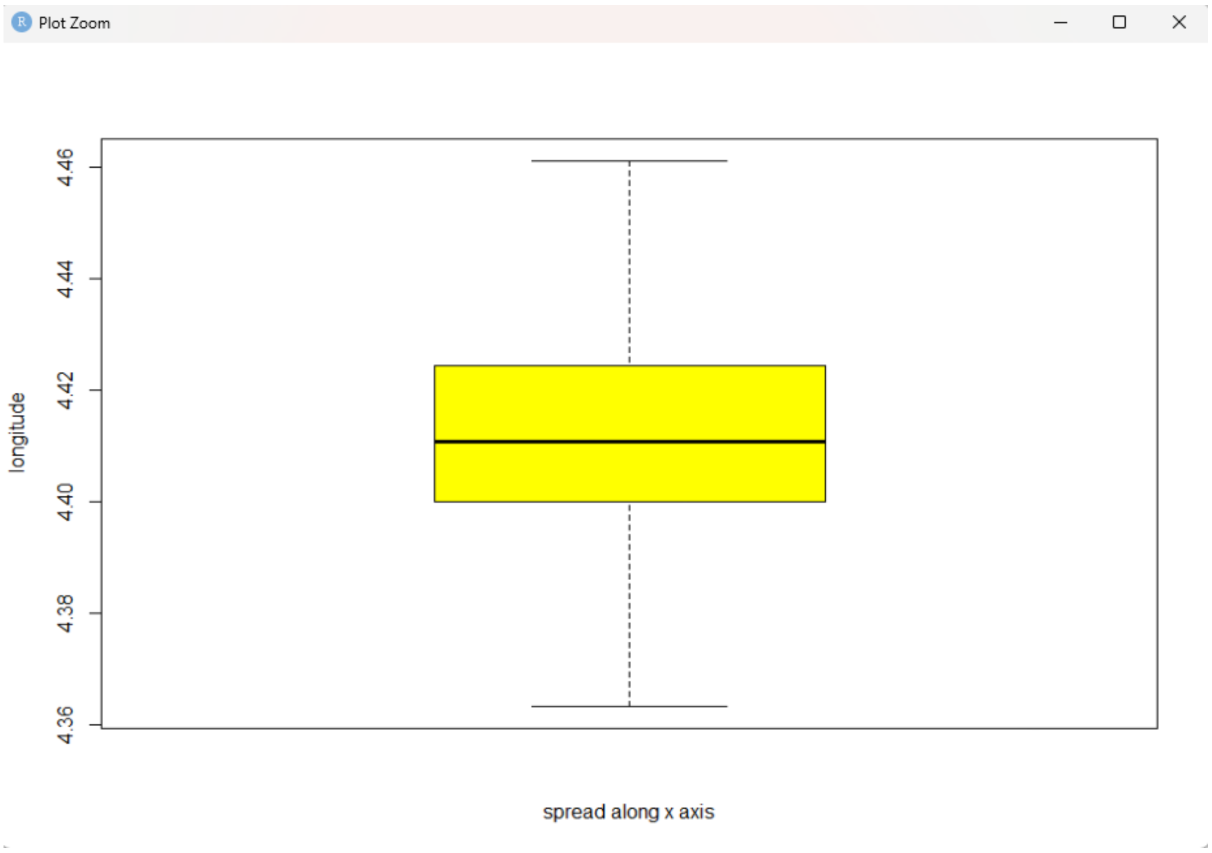
Boxplot of latitude column before handling outliers



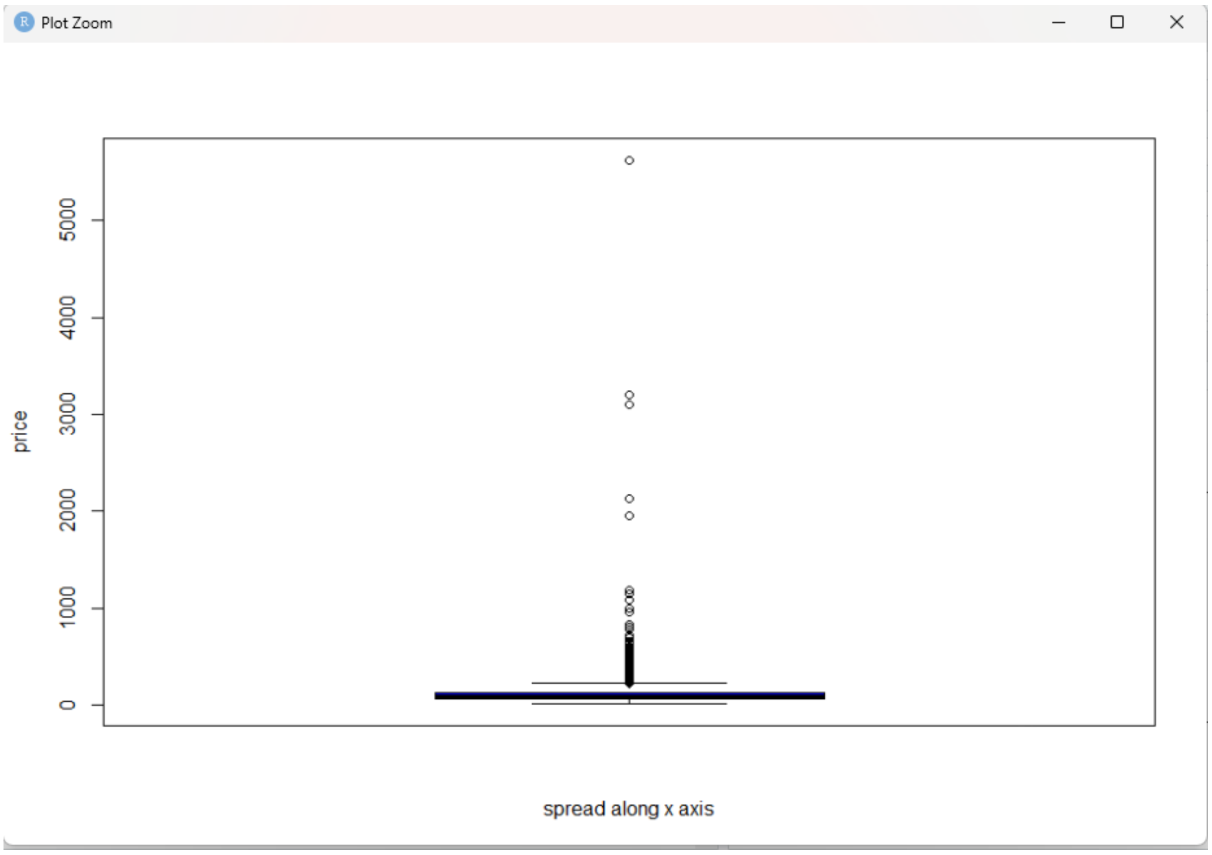
Box plot of latitude column after handling outliers



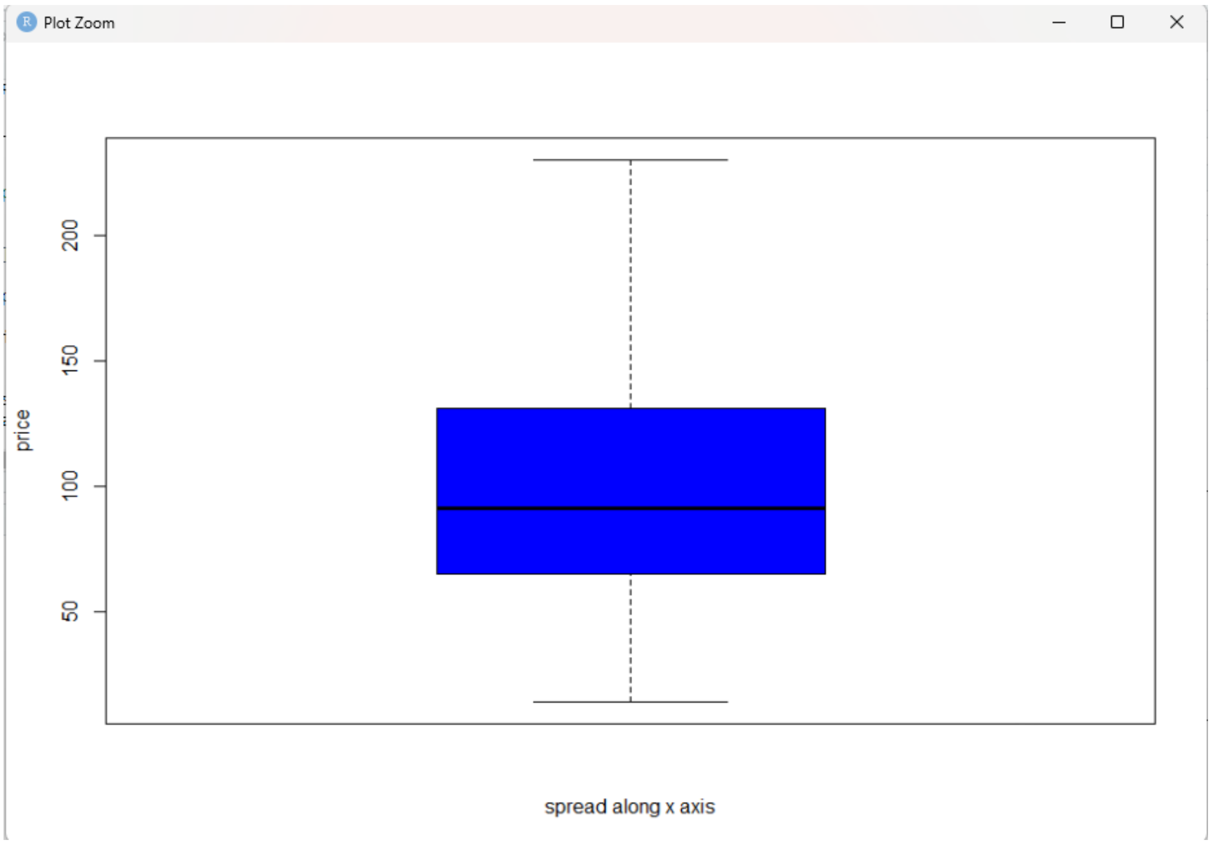
Boxplot of longitude column before handling outliers



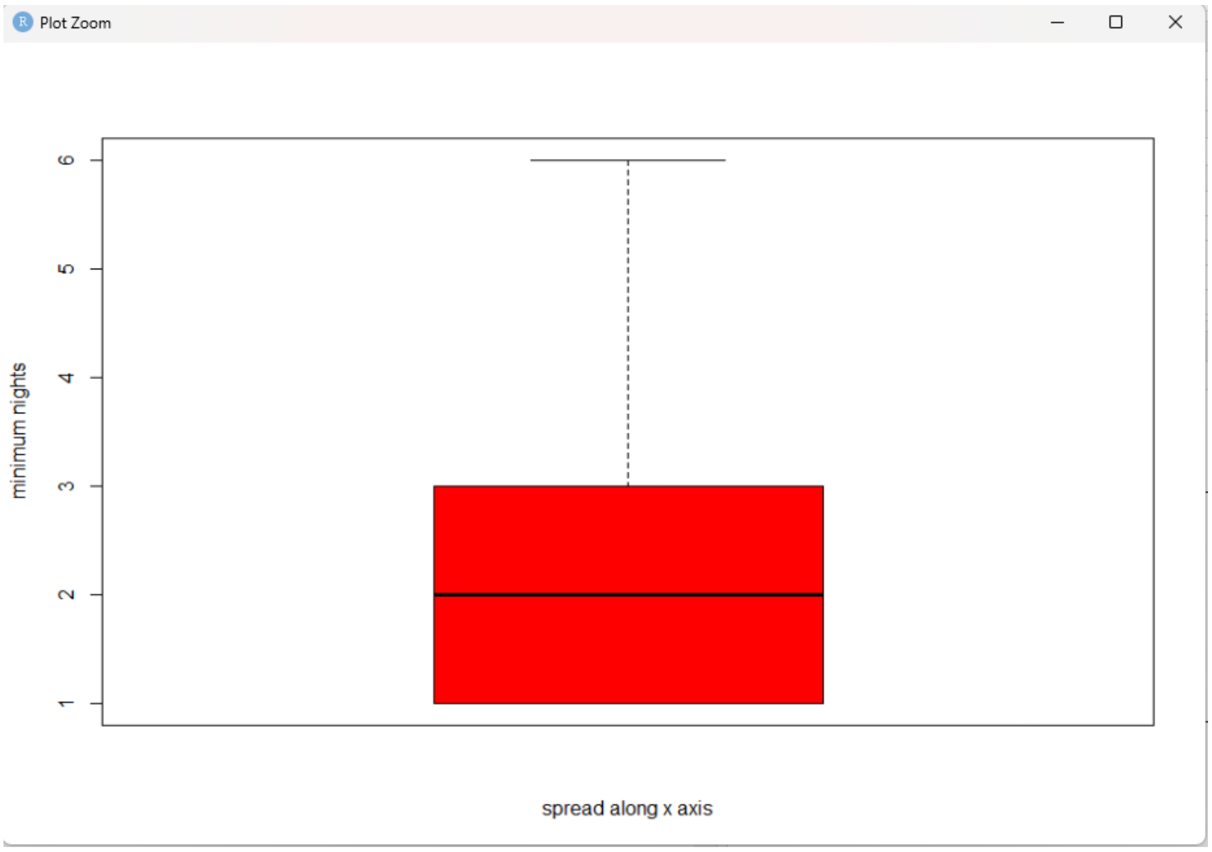
Boxplot of longitude column after handling outliers



Boxplot of price column before handling outliers



Boxplot of price column after handling outliers

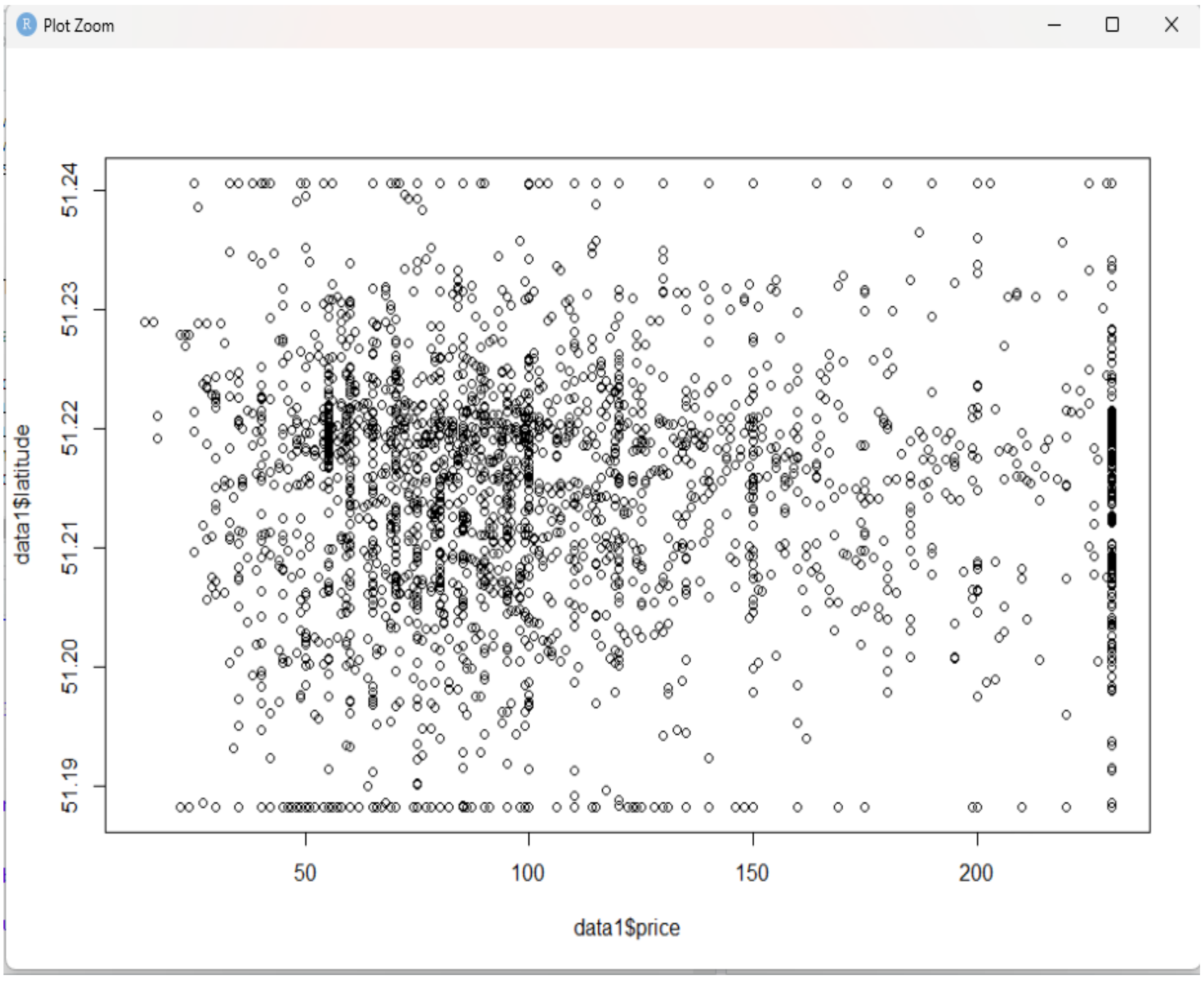


Boxplot of minimum nights column after handling outliers

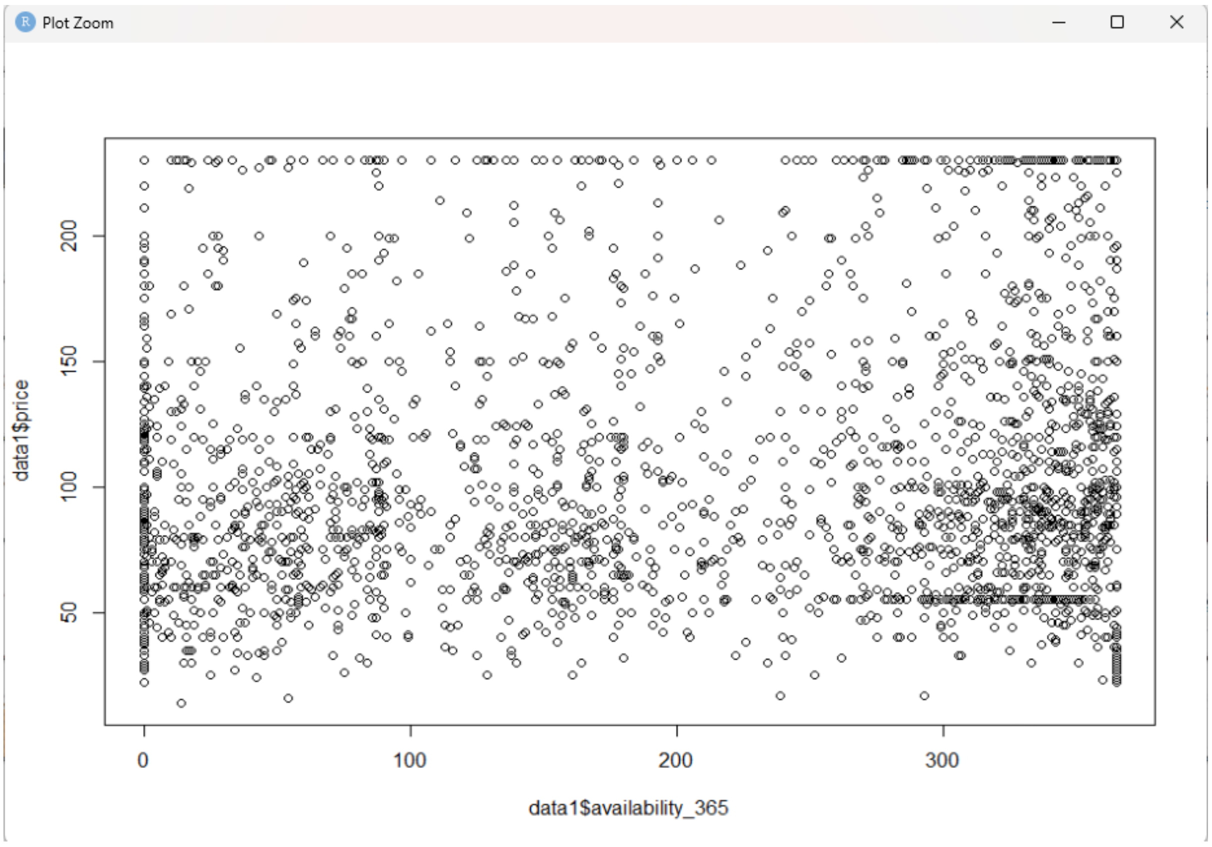
SCATTER PLOTS

A scatter plot is a type of data visualization that displays individual data points as dots on a two-dimensional plane. It's often used to explore the relationship between two variables and to identify patterns, trends, clusters, or outliers. In R programming, you can create scatter plots using the plot() function**.**

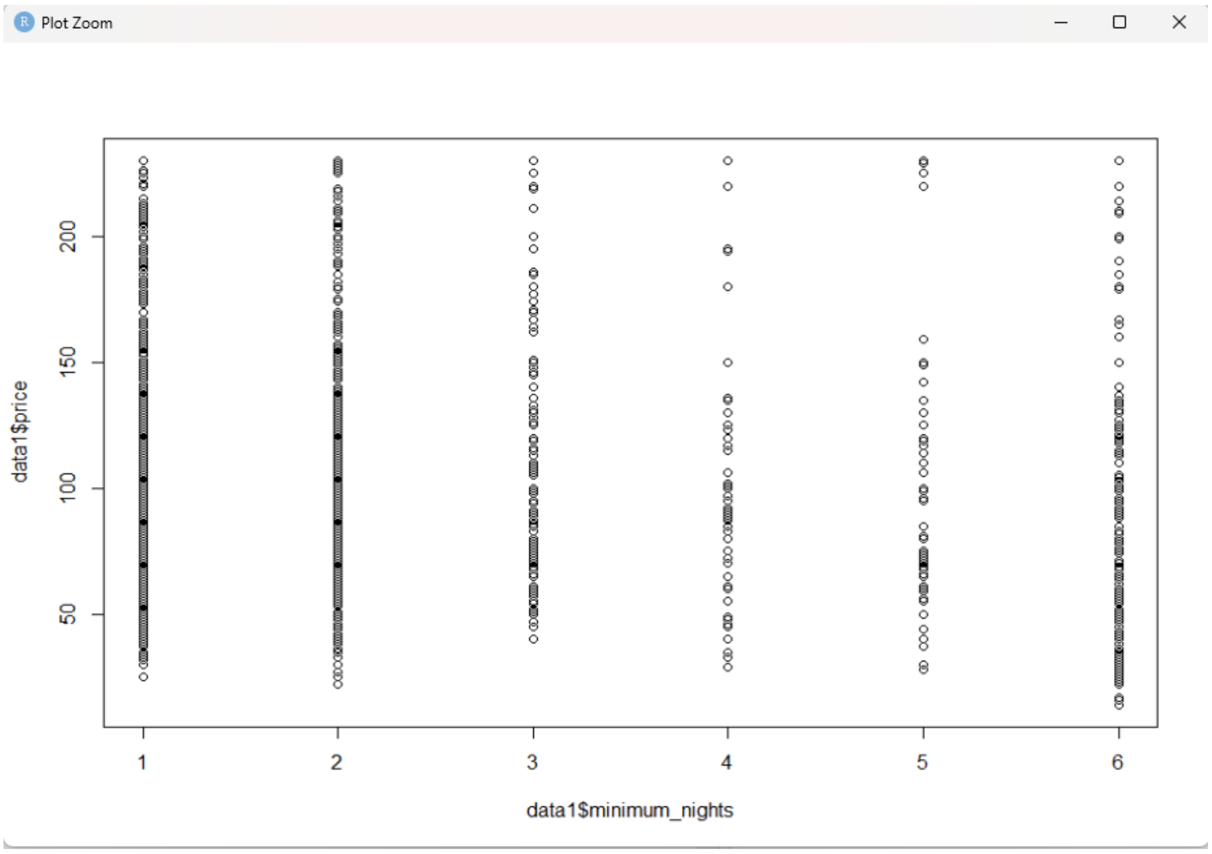
Scatter plots are particularly useful for showing the relationship between two numerical variables. You can easily identify trends, correlations, clusters, and outliers. Additionally, you can customize scatter plots by adding regression lines, different point shapes, colors, and labels, making them a versatile tool for exploratory data analysis and data visualization.



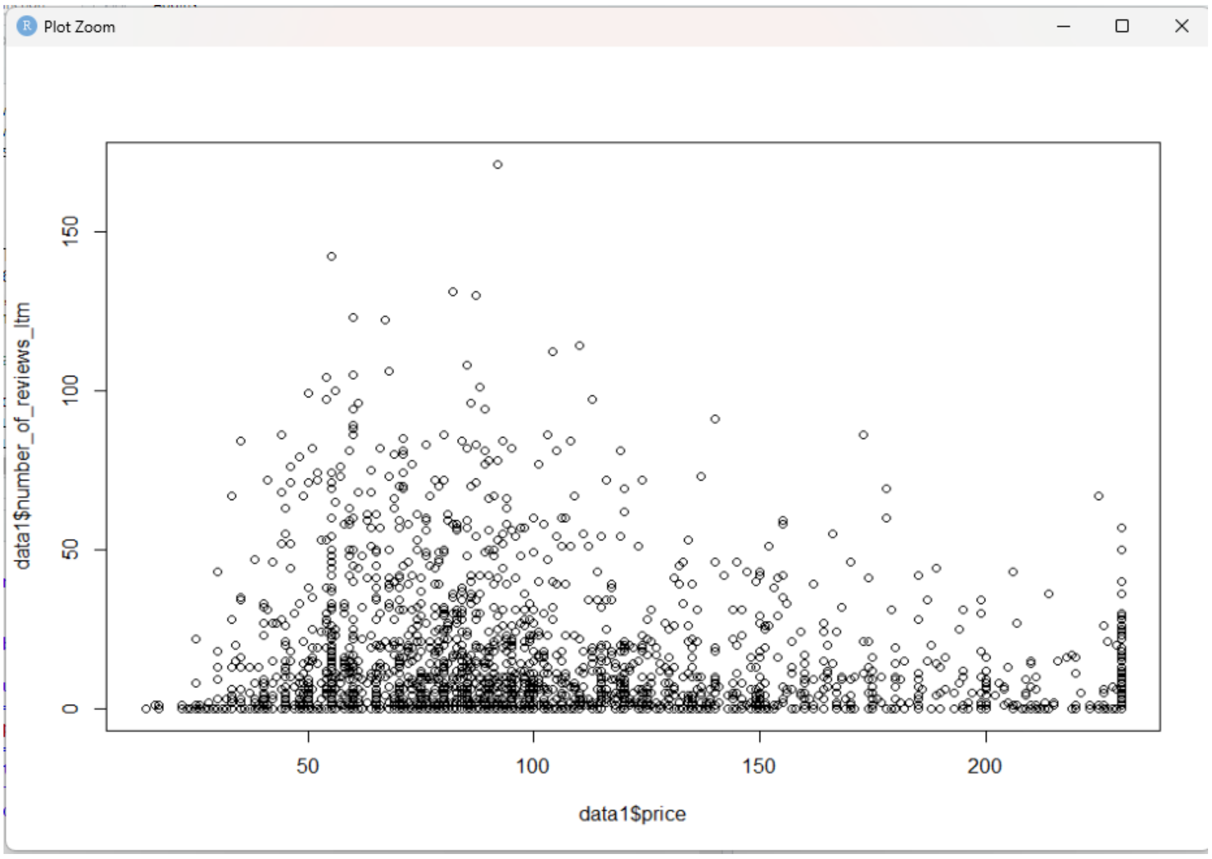
Scatter plots between latitude and price



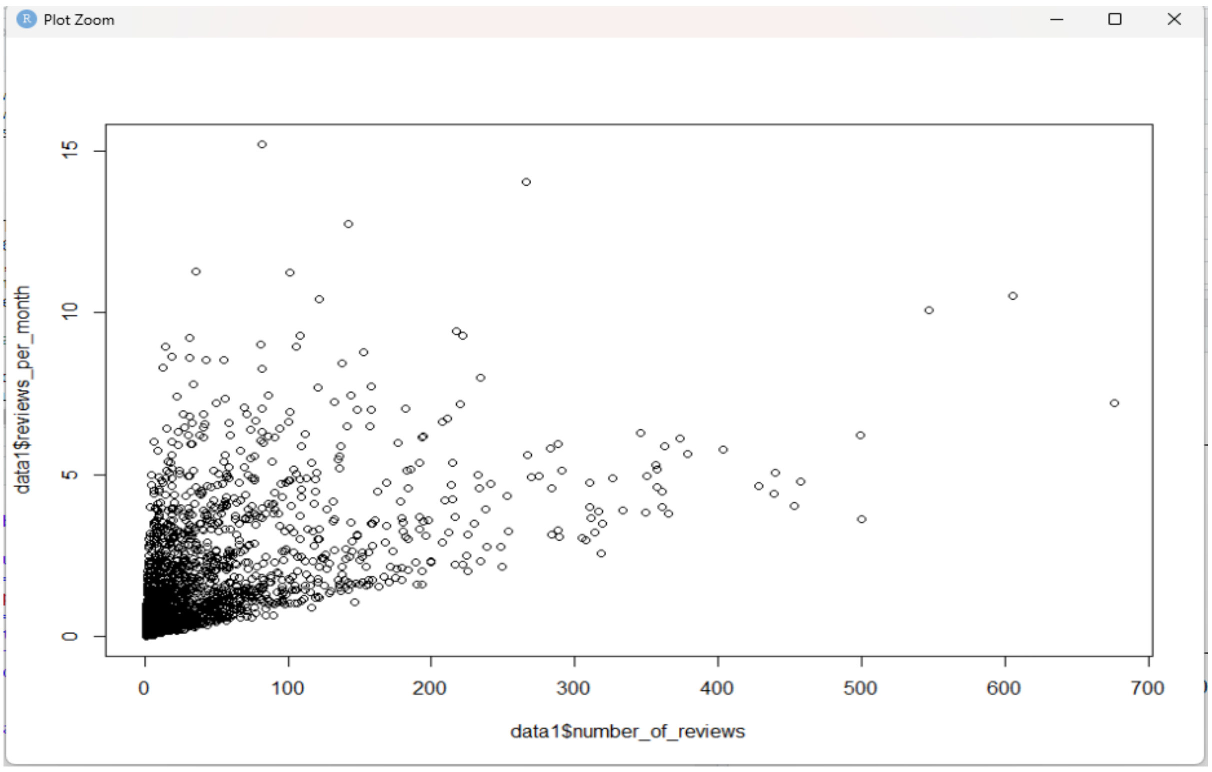
Scatter plots between price and availability



Scatter plots between price and minimum nights



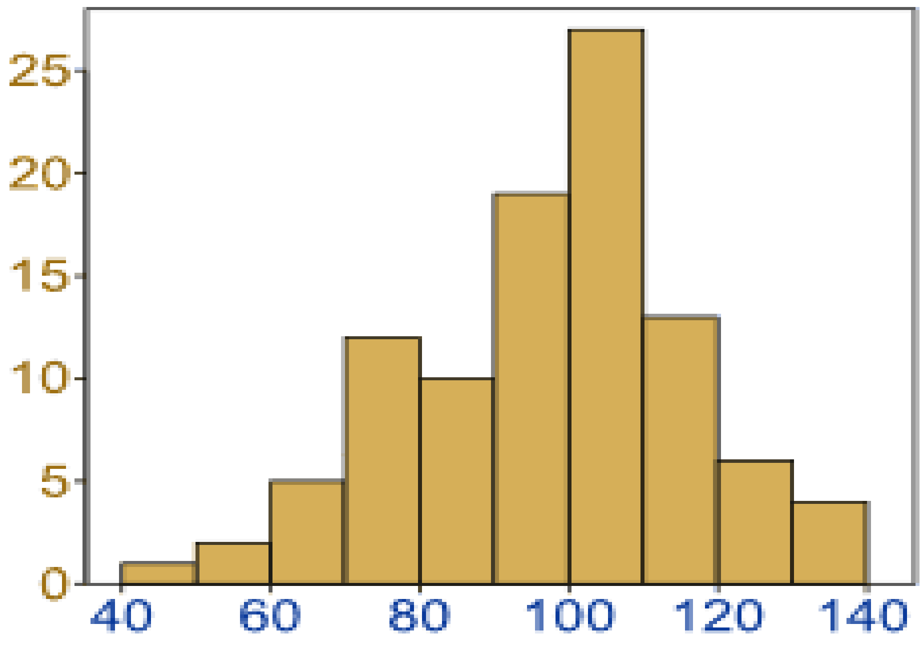
Scatter plots between price and number of reviews item



Scatter plots between reviews per month and number of reviews

HISTOGRAMS

A histogram is a graphical representation of the distribution of a dataset. It provides a visual overview of how the data is spread across different bins or intervals. In R programming, you can create histograms using the hist() function.



Histograms are particularly useful for visualizing the frequency distribution of continuous data. They help you understand the distribution's shape, center, and spread. Histograms can also highlight potential gaps or clusters in the data, making them an essential tool in exploratory data analysis and data visualization.

A histogram represents the frequencies of values of a variable bucketed into ranges. Histogram is similar to bar chat but the difference is it groups the values into continuous ranges. Each bar in histogram represents the height of the number of values present in that range.

The basic syntax for creating a histogram using R is −

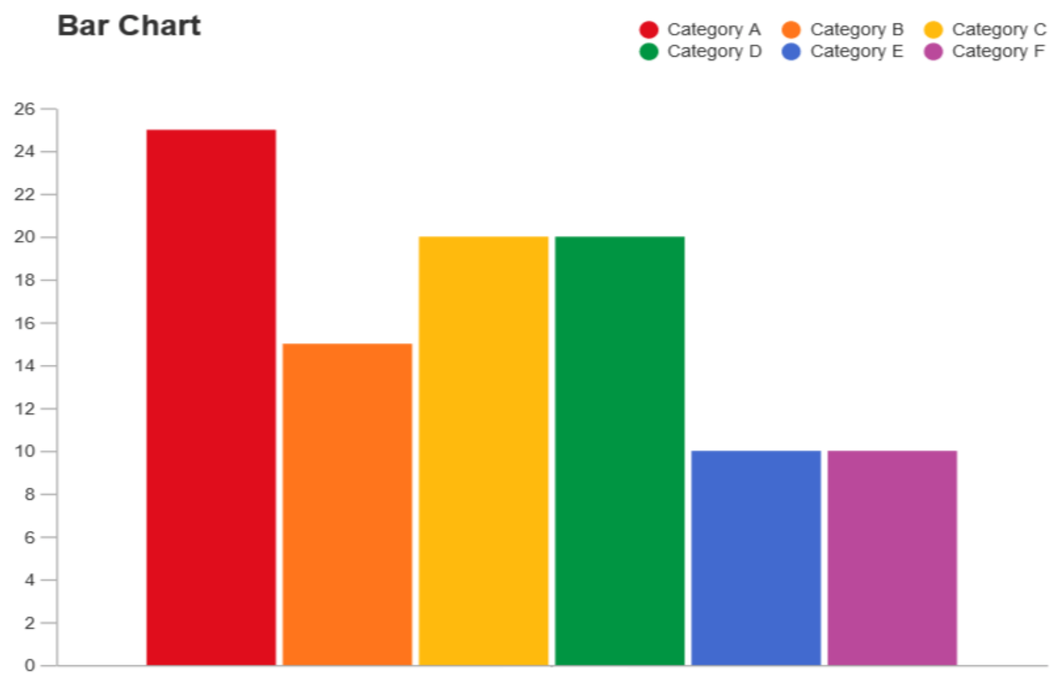
hist(v,main,xlab,xlim,ylim,breaks,col,border)

Following is the description of the parameters used −

- **v** is a vector containing numeric values used in histogram.
- **main** indicates title of the chart.
- **col** is used to set color of the bars.
- **border** is used to set border color of each bar.
- **xlab** is used to give description of x-axis.
- **xlim** is used to specify the range of values on the x-axis.
- **ylim** is used to specify the range of values on the y-axis.
- **breaks** is used to mention the width of each bar.

BAR CHART

A bar chart is a common type of data visualization that displays data using rectangular bars. Each bar's length is proportional to the value it represents. Bar charts are great for comparing categorical data or showing the frequency or count of different categories. In R programming, you can create bar charts using the barplot() function or other libraries like ggplot2.



A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable. R uses the function **barplot()** to create bar charts. R can draw both vertical and Horizontal bars in the bar chart. In bar chart each of the bars can be given different colors.

Syntax

The basic syntax to create a bar-chart in R is −

barplot(H,xlab,ylab,main, names.arg,col)

Following is the description of the parameters used −

- **H** is a vector or matrix containing numeric values used in bar chart.
- **xlab** is the label for x axis.
- **ylab** is the label for y axis.
- **main** is the title of the bar chart.
- **names.arg** is a vector of names appearing under each bar.
- **col** is used to give colors to the bars in the graph.

PIE CHART

A pie chart is a circular chart that displays data as slices of a pie, with each slice representing a category and its size proportional to the data it represents. In R programming, you can create pie charts using the pie() function.

Syntax

The basic syntax for creating a pie-chart using the R is −

pie(x, labels, radius, main, col, clockwise)

Following is the description of the parameters used −

- **x** is a vector containing the numeric values used in the pie chart.
- **labels** is used to give description to the slices.
- **radius** indicates the radius of the circle of the pie chart.(value between −1 and +1).
- **main** indicates the title of the chart.
- **col** indicates the color palette.
- **clockwise** is a logical value indicating if the slices are drawn clockwise or anti clockwise.

CHAPTER -5

DATA STRUCTURES IN R PROGRAMMING

A data structure is a particular way of organizing data in a computer so that it can be used effectively. The idea is to reduce the space and time complexities of different tasks. Data structures in R programming are tools for holding multiple values.

R's base data structures are often organized by their dimensionality (1D, 2D, or nD) and whether they're homogeneous (all elements must be of the identical type) or heterogeneous (the elements are often of various types). This gives rise to the six data types which are most frequently utilized in data analysis.

The most essential data structures used in R include:

- **Vectors**
- **Lists**
- **Dataframes**
- **Matrices**
- **Arrays**
- **Factors**

VECTORS IN R

A vector is simply a list of items that are of the same type.

To combine the list of items to a vector, use the c() function and separate the items by a comma.

In the example below, we create a vector variable called **fruits**, that combine strings:

FOR EXAMPLE

```
# Vector of strings
fruits <- c("banana", "apple", "orange")

# Print fruits
fruits

# Vector of numerical values
numbers <- c(1, 2, 3)

# Print numbers
numbers
```

To create a vector with numerical values in a sequence, use the : operator:

```
# Vector with numerical values in a sequence
numbers <- 1:10
```

```
numbers
```

You can also create numerical values with decimals in a sequence, but note that if the last element does not belong to the sequence, it is not used:

```
# Vector with numerical decimals in a sequence
numbers1 <- 1.5:6.5
numbers1



# Vector with numerical decimals in a sequence where the last element is not used
numbers2 <- 1.5:6.3
numbers2
```

Result:

```
[1] 1.5 2.5 3.5 4.5 5.5 6.5

[1] 1.5 2.5 3.5 4.5 5.5
```

In the example below, we create a vector of logical values:

```
# Vector of logical values
log_values <- c(TRUE, FALSE, TRUE, FALSE)

log_values
```

Vector Length

To find out how many items a vector has, use the length() function:

```
fruits <- c("banana", "apple", "orange")

length(fruits)
```

Sort a Vector

To sort items in a vector alphabetically or numerically, use the sort() function:

EXAMPLE

```
fruits <- c("banana", "apple", "orange", "mango", "lemon")
numbers <- c(13, 3, 5, 7, 20, 2)

sort(fruits)  # Sort a string
sort(numbers) # Sort numbers
```

ACCESS VECTORS

You can access the vector items by referring to its index number inside brackets []. The first item has index 1, the second item has index 2, and so on:

```
fruits <- c("banana", "apple", "orange")

# Access the first item (banana)
fruits[1]
```

You can also access multiple elements by referring to different index positions with the c() function:

```
fruits <- c("banana", "apple", "orange", "mango", "lemon")

# Access the first and third item (banana and orange)
fruits[c(1, 3)]
```

You can also use negative index numbers to access all items except the ones specified:

```
fruits <- c("banana", "apple", "orange", "mango", "lemon")

# Access all items except for the first item
fruits[c(-1)]
```

Change an Item

To change the value of a specific item, refer to the index number:

EXAMPLE

```
fruits <- c("banana", "apple", "orange", "mango", "lemon")

# Change "banana" to "pear"
fruits[1] <- "pear"
```

# Print fruits
fruits

Repeat Vectors

To repeat vectors, use the rep() function:

repeat_each <- rep(c(1,2,3), each = 3)

repeat_each

Repeat the sequence of the vectors:

repeat_times <- rep(c(1,2,3), times = 3)

repeat_times
Repeat each value independently:
repeat_indepent <- rep(c(1,2,3), times = c(5,2,1))

repeat_indepent

Generating Sequenced Vectors

One of the examples on top, showed you how to create a vector with numerical values in a sequence with the : operator:

example

numbers <- 1:10

numbers

To make bigger or smaller steps in a sequence, use the seq() function:

example

numbers <- seq(from = 0, to = 100, by = 20)

numbers

Note: The seq() function has three parameters: from is where the sequence starts, to is where the sequence stops, and by is the interval of the sequence.

DATA FRAME

Data Frames in R Language are generic data objects of R that are used to store tabular data. Data frames can also be interpreted as matrices where each column of a matrix can be of different data types. R DataFrame is made up of three principal components, the data, rows, and columns.

R program to create dataframe

```
# creating a data frame
friend.data <- data.frame(
   friend_id = c(1:5),
   friend_name = c("Sachin", "Sourav",
                  "Dravid", "Sehwag",
                  "Dhoni"),
   stringsAsFactors = FALSE
)
# print the data frame
print(friend.data)
```

OUTPUT

```
    friend_id friend_name
1        1     Sachin
2        2     Sourav
3        3     Dravid
4        4     Sehwag
5        5      Dhoni
```

To create an R data frame use **data.frame()** command and then pass each of the vectors you have created as arguments to the function.

Get the Structure of the R – Data Frame

One can get the structure of the R data frame using **str()** function in R. It can display even the internal structure of large lists which are nested. It provides one-liner output for the basic R objects letting the user know about the object and its constituents.

Summary of data in the R data frame

In the R data frame, the statistical summary and nature of the data can be obtained by applying **summary()** function. It is a generic function used to produce result summaries of the results of various model fitting functions. The function invokes particular methods which depend on the class of the first argument.

```
# R program to get the
# summary of the data frame

# creating a data frame
friend.data <- data.frame(
   friend_id = c(1:5),
   friend_name = c("Sachin", "Sourav",
                "Dravid", "Sehwag",
                "Dhoni"),
   stringsAsFactors = FALSE
)
# using summary()
print(summary(friend.data))
```

OUTPUT

```
friend_id friend_name
 Min.   :1   Length:5
 1st Qu.:2   Class :character
 Median :3   Mode  :character
 Mean   :3
 3rd Qu.:4
 Max.   :5
```

EXTRACT DATA FROM DATA FRAMES

Extracting data from an R data frame means that to access its rows or columns. One can extract a specific column from an R data frame using its column name.

```
# R program to extract
# data from the data frame

# creating a data frame
friend.data <- data.frame(
   friend_id = c(1:5),
   friend_name = c("Sachin", "Sourav",
               "Dravid", "Sehwag",
               "Dhoni"),
   stringsAsFactors = FALSE
)

# Extracting friend_name column
result <- data.frame(friend.data$friend_name)
print(result)
```

OUTPUT

```
friend.data.friend_name
1              Sachin
2              Sourav
3              Dravid
4              Sehwag
5               Dhoni
```

Expand Data Frame in R Language

A data frame in R can be expanded by adding new columns and rows to the already existing R data frame.

R program to expand

```
# the data frame

# creating a data frame
```

```
friend.data <- data.frame(
    friend_id = c(1:5),
    friend_name = c("Sachin", "Sourav",
                    "Dravid", "Sehwag",
                    "Dhoni"),
    stringsAsFactors = FALSE
)

# Expanding data frame
friend.data$location <- c("Kolkata", "Delhi",
                       "Bangalore", "Hyderabad",
                       "Chennai")
resultant <- friend.data
# print the modified data frame
print(resultant)
```

OUTPUT

```
friend_id friend_name  location
1         1      Sachin   Kolkata
2         2      Sourav     Delhi
3         3      Dravid Bangalore
4         4      Sehwag Hyderabad
5         5       Dhoni   Chennai
```

In R, one can perform various types of operations on a data frame like **accessing rows and columns, selecting the subset of the data frame, editing data frames, delete rows and columns in a data frame**, etc.

Remove Rows and Columns

A data frame in R removes columns and rows from the already existing R data frame.

```
library(dplyr)
# Create a data frame
data <- data.frame(
  friend_id = c(1, 2, 3, 4, 5),
  friend_name = c("Sachin", "Sourav", "Dravid", "Sehwag", "Dhoni"),
  location = c("Kolkata", "Delhi", "Bangalore", "Hyderabad", "Chennai")
)

# Remove a row with friend_id = 3
```

```
data <- subset(data, friend_id != 3)

# Remove the 'location' column
data <- select(data, -location)
```

OUTPUT

```
  friend_id friend_name
1         1      Sachin
2         2      Sourav
4         4      Sehwag
5         5       Dhoni
```

LIST

Lists are the R objects which contain elements of different types like − numbers, strings, vectors and another list inside it. A list can also contain a matrix or a function as its elements. List is created using **list()** function.

Creating a List

Following is an example to create a list containing strings, numbers, vectors and a logical values.

```
# Create a list containing strings, numbers, vectors and a logical

# values.

list_data <- list("Red", "Green", c(21,32,11), TRUE, 51.23, 119.1)

print(list_data)
```

OUTPUT

```
[[1]]

[1] "Red"


[[2]]

[1] "Green"


[[3]]

[1] 21 32 11


[[4]]

[1] TRUE
```

Naming List Elements

The list elements can be given names and they can be accessed using these names

CODE

```
# Create a list containing a vector, a matrix and a list.

list_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),

   list("green",12.3))


# Give names to the elements in the list.

names(list_data) <- c("1st Quarter", "A_Matrix", "A Inner list")


# Show the list.

print(list_data)
```

OUTPUT

```
$`1st_Quarter`

[1] "Jan" "Feb" "Mar"

$A_Matrix

     [,1] [,2] [,3]

[1,]    3    5   -2

[2,]    9    1    8

$A_Inner_list

$A_Inner_list[[1]]

[1] "green"

$A_Inner_list[[2]]

[1] 12.3
```

Accessing List Elements

Elements of the list can be accessed by the index of the element in the list. In case of named lists it can also be accessed using the names.

We continue to use the list in the above example –

CODE

```
# Create a list containing a vector, a matrix and a list.

list_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),

  list("green",12.3))


# Give names to the elements in the list.

names(list_data) <- c("1st Quarter", "A_Matrix", "A Inner list")


# Access the first element of the list.

print(list_data[1])


# Access the thrid element. As it is also a list, all its elements will be printed.

print(list_data[3])


# Access the list element using the name of the element.

print(list_data$A_Matrix)
```

When we execute the previous page code, it produces the following result –

```
$`1st_Quarter`
[1] "Jan" "Feb" "Mar"

$A_Inner_list
$A_Inner_list[[1]]
[1] "green"

$A_Inner_list[[2]]
[1] 12.3

     [,1] [,2] [,3]
[1,]    3    5   -2
[2,]    9    1    8
```

Manipulating List Elements

We can add, delete and update list elements as shown below. We can add and delete elements only at the end of a list. But we can update any element.

CODE

```
# Create a list containing a vector, a matrix and a list.

list_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),

   list("green",12.3))



# Give names to the elements in the list.

names(list_data) <- c("1st Quarter", "A_Matrix", "A Inner list")



# Add element at the end of the list.

list_data[4] <- "New element"

print(list_data[4])



# Remove the last element.

list_data[4] <- NULL
```

```
# Print the 4th Element.

print(list_data[4])



# Update the 3rd Element.

list_data[3] <- "updated element"

print(list_data[3])
```

When we execute the above code, it produces the following result −

```
[[1]]
[1] "New element"

$<NA>
NULL

$`A Inner list`
[1] "updated element"
```

MATRICES

A matrix is a rectangular arrangement of numbers in rows and columns. In a matrix, as we know rows are the ones that run horizontally and columns are the ones that run vertically. Matrices are two-dimensional, homogeneous data structures.
Now, let’s see how to create a matrix in R. To create a matrix in R you need to use the function called matrix. The arguments to this matrix() are the set of elements in the vector. You have to pass how many numbers of rows and how many numbers of columns you want to have in your matrix and this is the important point you have to remember that by default, matrices are in column-wise order. ACDDDD

```
# R program to create a matrix

A = matrix(

  # Taking sequence of elements
  c(1, 2, 3, 4, 5, 6, 7, 8, 9),

  # No of rows
  nrow = 3,
  # No of columns
  ncol = 3,

  # By default matrices are in column-wise order
  # So this parameter decides how to arrange the matrix
  byrow = TRUE
)

# Naming rows
rownames(A) = c("a", "b", "c")

# Naming columns
colnames(A) = c("c", "d", "e")

cat("The 3x3 matrix:\n")
print(A)
```

OUTPUT

```
The 3x3 matrix:
  c d e
a 1 2 3
b 4 5 6
c 7 8 9
```

ARRAYS

Arrays are the R data objects which store the data in more than two dimensions. Arrays are n-dimensional data structures. For example, if we create an array of dimensions (2, 3, 3) then it creates 3 rectangular matrices each with 2 rows and 3 columns. They are homogeneous data structures.

Now, let’s see how to create arrays in R. To create an array in R you need to use the function called array(). The arguments to this array() are the set of elements in vectors and you have to pass a vector containing the dimensions of the array.

SYNTAX FOR ARRAY

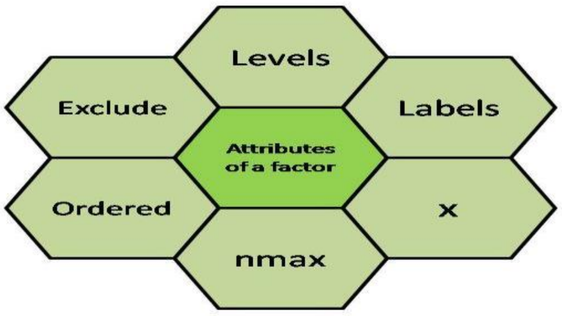
array_name <- array(data, dim= (row_size, column_size, matrices, dim_names))

FACTORS

Factors in R Programming Language are data structures that are implemented to categorize the data or represent categorical data and store it on multiple levels.
They can be stored as integers with a corresponding label to every unique integer. The R factors may look similar to character vectors, they are integers and care must be taken while using them as strings. The R factor accepts only a restricted number of distinct values. For example, a data field such as gender may contain values only from female, male, or transgender.
In the above example, all the possible cases are known beforehand and are predefined. These distinct values are known as levels. After a factor is created it only consists of levels that are by default sorted alphabetically.

Attributes of Factors in R Language

- **x:** It is the vector that needs to be converted into a factor.
- **Levels:** It is a set of distinct values which are given to the input vector x.
- **Labels:** It is a character vector corresponding to the number of labels.
- **Exclude:** This will mention all the values you want to exclude.
- **Ordered:** This logical attribute decides whether the levels are ordered.
- **nmax:** It will decide the upper limit for the maximum number of levels.



Creating a Factor in R Programming Language

The command used to create or modify a factor in R language is – **factor()** with a vector as input.
The two steps to creating an R factor :
- Creating a vector
- Converting the vector created into a factor using function factor()

CHAPTER-6

ABOUT AIRBNB DATA

BOOOKING DATA

The most basic type of Airbnb data is booking data. This includes information such as the dates your listing is booked, the length of stay, and the number of guests. This data can help you understand booking patterns, such as which days of the week are most popular, and which times of the year are busiest.

PRICING DATA

Pricing data is perhaps the most important type of Airbnb data for hosts. This includes information on the prices of other listings in your area, as well as the prices of your own listing over time. By analyzing pricing data, you can determine the optimal price for your listing based on the current market.

REVIEWS DATA

Reviews data is another important type of Airbnb data for hosts. This includes information on the ratings and reviews that guests leave for your listing. By analyzing reviews data, you can identify areas where your listing is excelling, as well as areas where it may need improvement.

CANCELLATION DATA

Cancellation data is another important type of Airbnb data. Cancellation data can provide valuable insights into the reasons guests may be canceling their bookings, and can help hosts take steps to reduce the number of cancellations they experience. This data includes information such as the number of cancellations, the reason for the cancellation, and the timing of the cancellation.

OCCUPANCY DATA

Occupancy data provides insight into how frequently your listing is being booked and can help you make decisions about pricing and availability. This data includes information such as the number of days your listing is booked, the number of days it’s available, and the occupancy rate.

CONCLUSION

The Analysis and Prediction of Airbnb Listing Prices in R programming is a data-driven project that combines data collection, preprocessing, exploratory analysis, feature engineering, model building, evaluation, and interpretation to uncover insights and create predictive models. By the end of this project, we will have a comprehensive understanding of the factors influencing Airbnb listing prices and the ability to make informed predictions for pricing new listings. This knowledge can empower hosts to set competitive prices and help guests make informed booking decisions.

The project "Analysis and Prediction of Airbnb Listing Prices" conducted using R programming has been an insightful and valuable endeavor that has provided significant insights into the dynamics of Airbnb listings and their associated prices. Through a combination of data analysis, feature engineering, we aimed to understand the factors that influence listing prices and develop predictive models to estimate those prices accurately.

Throughout the project, we undertook various stages of data processing, exploration, modeling, and evaluation, each contributing to a deeper understanding of the underlying trends in the data. The results obtained have shed light on the key drivers affecting listing prices and have enabled us to make informed predictions about the potential prices of new listings.

In conclusion, the "Analysis and Prediction of Airbnb Listing Prices" project has not only expanded our technical skills in R programming, data manipulation, and machine learning but has also deepened our understanding of the dynamics of the sharing economy and its implications for both hosts and guests. The insights gained from this project can serve as a foundation for further research and real-world applications in the realm of hospitality and pricing optimization.

REFERENCES

1. **BOOKS**

   Hadley Wickham and Garrett Grolemund , R for Data Science , O ‘REILLY MEDIA

   Garrett Grolemund , Hands-On Programming with R , O ‘REILLY MEDIA

2. **SITES**

   Geeks for geeks

   Tutorials point

   Airbnb data site

3. **COURSE PLATFORMS**

   Board infinity