Problem Set 1

Kai Mao

Table of contents

1 Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

2 Problem 1 - Wine Data

2.1 a. Code on how to import the data

```
# Import the data
#' @description Import the wine dataset from a specified path and assign column names.
#' @param filePath The path to the wine dataset file.
#' @return No return value, directly creates a dataFrame
wine <- read.csv("C:\\Users\\Feixing\\Desktop\\stats 506\\data\\Problem Set 1\\wine\\wine.da
# Give appropriate column names
colnames(wine) <- c("Class", "Alcohol", "Malic_acid", "Ash","Alcalinity_of_ash", "Magnesium"
                    "Total_phenols", "Flavanoids", "Nonflavanoid_phenols", "Proanthocyanins"
                    "Color_intensity", "Hue", "OD280/OD315_of_diluted_wines", "Proline")
# Display the first few lines of data to confirm that the import is correct
head(wine)
```

```
  Class Alcohol Malic_acid  Ash Alcalinity_of_ash Magnesium Total_phenols
1     1   14.23       1.71 2.43              15.6       127          2.80
2     1   13.20       1.78 2.14              11.2       100          2.65
3     1   13.16       2.36 2.67              18.6       101          2.80
4     1   14.37       1.95 2.50              16.8       113          3.85
5     1   13.24       2.59 2.87              21.0       118          2.80
6     1   14.20       1.76 2.45              15.2       112          3.27
  Flavanoids Nonflavanoid_phenols Proanthocyanins Color_intensity  Hue
```

```
1       3.06                    0.28            2.29            5.64 1.04
2       2.76                    0.26            1.28            4.38 1.05
3       3.24                    0.30            2.81            5.68 1.03
4       3.49                    0.24            2.18            7.80 0.86
5       2.69                    0.39            1.82            4.32 1.04
6       3.39                    0.34            1.97            6.75 1.05
  OD280/OD315_of_diluted_wines Proline
1                         3.92    1065
2                         3.40    1050
3                         3.17    1185
4                         3.45    1480
5                         2.93     735
6                         2.85    1450
```

2.2 b. Check and report that the number of wines

```
# Use the table function to calculate the amount of each type of wine
#' @description Calculate and report the count of each wine class from the dataset.
wine_class <- table(wine$Class)
#' @details Outputs are formatted using cat and sprintf for clear presentation.
cat(sprintf("Class 1: %d\nClass 2: %d\nClass 3: %d\n", wine_class[1], wine_class[2], wine_cla
```

```
Class 1: 59
Class 2: 71
Class 3: 48
```

2.3 c. Use the data to answer the following questions:

2.3.1 1.Find the correlation between alcohol content and color intensity

```
#' @description Calculate and output the correlation coefficient between alcohol content and
whole_cor <- cor(wine$Alcohol,wine$Color_intensity)
# Output the correlation coefficient
cat(sprintf("The correlation between alcohol content and color intensity is %.4f\n", whole_co
```

```
The correlation between alcohol content and color intensity is 0.5464
```

2.3.2 2. Find the highest and lowest correlation

```r
# Calculate the correlations of the three categories separately
class1_cor <- cor(wine$Alcohol[wine$Class==1], wine$Color_intensity[wine$Class==1])
class2_cor <- cor(wine$Alcohol[wine$Class==2], wine$Color_intensity[wine$Class==2])
class3_cor <- cor(wine$Alcohol[wine$Class==3], wine$Color_intensity[wine$Class==3])
corlist <- c(class1_cor, class2_cor, class3_cor)
max_cor <- max(corlist)
min_cor <- min(corlist)
max_class <- which(corlist == max_cor)
min_class <- which(corlist == min_cor)
# Output the highest and lowest correlation values and their corresponding categories
cat(sprintf("Class %d has the highest correlation: %.4f\n", max_class, max_cor))
```

```
Class 1 has the highest correlation: 0.4083
```

```r
cat(sprintf("Class %d has the lowest correlation: %.4f\n", min_class, min_cor))
```

```
Class 2 has the lowest correlation: 0.2698
```

2.3.3 3. Find the alcohol content of the wine with the highest color intensity

```r
max_color_index <- which.max(wine$Color_intensity)
max_color_alcohol <- wine$Alcohol[max_color_index]
# Find and display the alcohol content of the wine with the highest color intensity
cat(sprintf("The alcohol content of the wine with the highest color intensity is: %.4f, the h
```

```
The alcohol content of the wine with the highest color intensity is: 14.3400, the highest col
```

2.3.4 4. Find the percentage of wines had a higher content of proanthocyanins compare to ash

```r
higher_proanthocyanins_number <- sum(wine$Proanthocyanins > wine$Ash)
percentage <- (higher_proanthocyanins_number / nrow(wine)) * 100
# Output the percentage of wines had a higher content of proanthocyanins compare to ash
cat(sprintf("The percentage of wines had a higher content of proanthocyanins compared to ash
```

```
The percentage of wines had a higher content of proanthocyanins compared to ash: 8.4270%
```

2.4 d. Create a table identifying the average value of each variable, providing one row for the overall average, and one row per class with class averages.

```r
# Compute the overall averages
average <- t(colMeans(wine[,-1]))

# Calculate average values for each variable by class
class_average <- aggregate(. ~ Class, data = wine, FUN = mean)
class_average <- class_average[, -1]

# Set column names to maintain consistency with the overall averages
colnames(class_average) <- colnames(average)

# Combine the overall average and class-specific averages into one table
average_value <- rbind(average, class_average)

# Output
print(average_value)
```

```
   Alcohol Malic_acid      Ash Alcalinity_of_ash Magnesium Total_phenols
1 13.00062   2.336348 2.366517          19.49494  99.74157      2.295112
2 13.74475   2.010678 2.455593          17.03729 106.33898      2.840169
3 12.27873   1.932676 2.244789          20.23803  94.54930      2.258873
4 13.15375   3.333750 2.437083          21.41667  99.31250      1.678750
  Flavanoids Nonflavanoid_phenols Proanthocyanins Color_intensity       Hue
1  2.0292697            0.3618539        1.590899        5.058090 0.9574494
2  2.9823729            0.2900000        1.899322        5.528305 1.0620339
3  2.0808451            0.3636620        1.630282        3.086620 1.0562817
4  0.7814583            0.4475000        1.153542        7.396250 0.6827083
  OD280/OD315_of_diluted_wines   Proline
1                     2.611685  746.8933
2                     3.157797 1115.7119
3                     2.785352  519.5070
4                     1.683542  629.8958
```

2.5 e. Carry out a series of t-tests to examine whether the level of phenols differs across the three classes.

```r
# t-test between Class 1 and Class 2
t_test_1_2 <- t.test(wine$Total_phenols[wine$Class == 1],
                     wine$Total_phenols[wine$Class == 2])

# t-test between Class 1 and Class 3
t_test_1_3 <- t.test(wine$Total_phenols[wine$Class == 1],
                     wine$Total_phenols[wine$Class == 3])

# t-test between Class 2 and Class 3
t_test_2_3 <- t.test(wine$Total_phenols[wine$Class == 2],
                     wine$Total_phenols[wine$Class == 3])

# Display t-test results
t_test_1_2
```

```

    Welch Two Sample t-test

data:  wine$Total_phenols[wine$Class == 1] and wine$Total_phenols[wine$Class == 2]
t = 7.4206, df = 119.14, p-value = 1.889e-11
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.4261870 0.7364055
sample estimates:
mean of x mean of y 
 2.840169  2.258873 
```

```r
t_test_1_3
```

```

    Welch Two Sample t-test

data:  wine$Total_phenols[wine$Class == 1] and wine$Total_phenols[wine$Class == 3]
t = 17.12, df = 98.356, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.026801 1.296038
sample estimates:
mean of x mean of y 
 2.840169  1.678750 
```

```r
t_test_2_3
```

```
	Welch Two Sample t-test

data:  wine$Total_phenols[wine$Class == 2] and wine$Total_phenols[wine$Class == 3]
t = 7.0125, df = 116.91, p-value = 1.622e-10
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.4162855 0.7439610
sample estimates:
mean of x mean of y
 2.258873  1.678750
```

```r
# Additional for minor extra credit: Manually calculate t-statistics and p-values
# Calculate means and standard deviations for each class
mean1 <- mean(wine$Total_phenols[wine$Class == 1])
mean2 <- mean(wine$Total_phenols[wine$Class == 2])
mean3 <- mean(wine$Total_phenols[wine$Class == 3])
sd1 <- sd(wine$Total_phenols[wine$Class == 1])
sd2 <- sd(wine$Total_phenols[wine$Class == 2])
sd3 <- sd(wine$Total_phenols[wine$Class == 3])

# Calculate sample sizes
n1 <- sum(wine$Class == 1)
n2 <- sum(wine$Class == 2)
n3 <- sum(wine$Class == 3)

# Manually compute t-statistics
t_stat_1_2 <- (mean1 - mean2) / sqrt((sd1^2 / n1) + (sd2^2 / n2))
t_stat_1_3 <- (mean1 - mean3) / sqrt((sd1^2 / n1) + (sd3^2 / n3))
t_stat_2_3 <- (mean2 - mean3) / sqrt((sd2^2 / n2) + (sd3^2 / n3))

# Compute degrees of freedom
df_1_2 <- ((sd1^2 / n1) + (sd2^2 / n2))^2 /
          (((sd1^2 / n1)^2 / (n1 - 1)) + ((sd2^2 / n2)^2 / (n2 - 1)))

df_1_3 <- ((sd1^2 / n1) + (sd3^2 / n3))^2 /
          (((sd1^2 / n1)^2 / (n1 - 1)) + ((sd3^2 / n3)^2 / (n3 - 1)))

df_2_3 <- ((sd2^2 / n2) + (sd3^2 / n3))^2 /
```

```
          (((sd2^2 / n2)^2 / (n2 - 1)) + ((sd3^2 / n3)^2 / (n3 - 1)))

# Compute p-values
p_value_1_2 <- 2 * pt(-abs(t_stat_1_2), df = df_1_2)
p_value_1_3 <- 2 * pt(-abs(t_stat_1_3), df = df_1_3)
p_value_2_3 <- 2 * pt(-abs(t_stat_2_3), df = df_2_3)


# Output results
cat(sprintf("t-statistics and p-values between classes:\nClass 1 vs Class 2: t = %.4f, p = %g
            t_stat_1_2, p_value_1_2, t_stat_1_3, p_value_1_3, t_stat_2_3, p_value_2_3))
```

```
t-statistics and p-values between classes:
Class 1 vs Class 2: t = 7.4206, p = 1.88933e-11
Class 1 vs Class 3: t = 17.1202, p = 2.91504e-31
Class 2 vs Class 3: t = 7.0125, p = 1.62157e-10
```

3 Problem 2 - AskAManager.org Data

3.1 a. Import the data into a data.frame in R.

```
# Import data from a CSV file located on the local machine
#' @description Imports the dataset from a specified CSV file path into an R dataframe.
#' @param filePath String path to the dataset file.
#' @return No return value, directly creates a dataframe named 'manager'.
manager <- read.csv("C:\\Users\\Feixing\\Desktop\\stats 506\\data\\Problem Set 1\\AskAManage
head(manager)
```

```
  X          Timestamp How.old.are.you. What.industry.do.you.work.in.
1 1 4/27/2021 11:02:10            25-34  Education (Higher Education)
2 2 4/27/2021 11:02:22            25-34             Computing or Tech
3 3 4/27/2021 11:02:38            25-34 Accounting, Banking & Finance
4 4 4/27/2021 11:02:41            25-34                    Nonprofits
5 5 4/27/2021 11:02:42            25-34 Accounting, Banking & Finance
6 6 4/27/2021 11:02:46            25-34  Education (Higher Education)
                                 Job.title
1        Research and Instruction Librarian
2 Change & Internal Communications Manager
3                     Marketing Specialist
```

```
4                         Program Manager
5                      Accounting Manager
6            Scholarly Publishing Librarian
  If.your.job.title.needs.additional.context..please.clarify.here.
1
2
3
4
5
6
  What.is.your.annual.salary...You.ll.indicate.the.currency.in.a.later.question..If.you.are.
1
2
3
4
5
6
  How.much.additional.monetary.compensation.do.you.get..if.any..for.example..bonuses.or.over
1
2
3
4
5
6
  Please.indicate.the.currency If..Other...please.indicate.the.currency.here..
1                         USD
2                         GBP
3                         USD
4                         USD
5                         USD
6                         USD
  If.your.income.needs.additional.context..please.provide.it.here.
1
2
3
4
5
6
  What.country.do.you.work.in.
1              United States
2             United Kingdom
3                         US
4                        USA
```

```
5                          US
6                         USA
  If.you.re.in.the.U.S...what.state.do.you.work.in. What.city.do.you.work.in.
1                                     Massachusetts                    Boston
2                                                                   Cambridge
3                                         Tennessee               Chattanooga
4                                         Wisconsin                 Milwaukee
5                                    South Carolina                Greenville
6                                     New Hampshire                   Hanover
  How.many.years.of.professional.work.experience.do.you.have.overall.
1                                                           5-7 years
2                                                        8 - 10 years
3                                                         2 - 4 years
4                                                        8 - 10 years
5                                                        8 - 10 years
6                                                        8 - 10 years
  How.many.years.of.professional.work.experience.do.you.have.in.your.field.
1                                                                 5-7 years
2                                                                 5-7 years
3                                                               2 - 4 years
4                                                                 5-7 years
5                                                                 5-7 years
6                                                               2 - 4 years
  What.is.your.highest.level.of.education.completed. What.is.your.gender.
1                                    Master's degree               Woman
2                                     College degree          Non-binary
3                                     College degree               Woman
4                                     College degree               Woman
5                                     College degree               Woman
6                                    Master's degree                 Man
  What.is.your.race...Choose.all.that.apply..
1                                       White
2                                       White
3                                       White
4                                       White
5                                       White
6                                       White
```

3.2 b. Clean up the variable names.

```
#' @description Cleans and simplifies column names in the dataframe to enhance readability ar
#' @param dataFrame The dataframe object containing the original data.
#' @return Modifies the dataframe in place, changing column names.
names(manager)
```

```
 [1] "X"
 [2] "Timestamp"
 [3] "How.old.are.you."
 [4] "What.industry.do.you.work.in."
 [5] "Job.title"
 [6] "If.your.job.title.needs.additional.context..please.clarify.here."
 [7] "What.is.your.annual.salary...You.ll.indicate.the.currency.in.a.later.question..If.you.a
 [8] "How.much.additional.monetary.compensation.do.you.get..if.any..for.example..bonuses.or.o
 [9] "Please.indicate.the.currency"
[10] "If..Other...please.indicate.the.currency.here.."
[11] "If.your.income.needs.additional.context..please.provide.it.here."
[12] "What.country.do.you.work.in."
[13] "If.you.re.in.the.U.S...what.state.do.you.work.in."
[14] "What.city.do.you.work.in."
[15] "How.many.years.of.professional.work.experience.do.you.have.overall."
[16] "How.many.years.of.professional.work.experience.do.you.have.in.your.field."
[17] "What.is.your.highest.level.of.education.completed."
[18] "What.is.your.gender."
[19] "What.is.your.race...Choose.all.that.apply.."
```

```
names(manager) <- c("ID","Timestamp","Age","Job","Job_Title","Job_Context","Annual_Salary",
                    "Additional_Compensation","Currency","Other_Currency","Income_Context",
                    "Country","State","City","Overall_Professional_experience_Years",
                    "Field_Professional_experience_Years","Highest_Education","Gender","Race'
# Confirm changes by displaying the new column names
names(manager)
```

```
 [1] "ID"
 [2] "Timestamp"
 [3] "Age"
 [4] "Job"
 [5] "Job_Title"
 [6] "Job_Context"
 [7] "Annual_Salary"
 [8] "Additional_Compensation"
 [9] "Currency"
[10] "Other_Currency"
[11] "Income_Context"
[12] "Country"
[13] "State"
[14] "City"
[15] "Overall_Professional_experience_Years"
[16] "Field_Professional_experience_Years"
[17] "Highest_Education"
[18] "Gender"
[19] "Race"
```

3.3 c. Restrict the data to those being paid in US dollars.

```r
# Number of observations before restricting the data
initial_number <- nrow(manager)
cat("Initial number of observations before restricting the data:", initial_number, "\n")
```

```
Initial number of observations before restricting the data: 28062
```

```r
# Restrict the data to those being paid in USD
#' @description Filters entries based on currency, retaining only those with salaries in USD
#' @param dataFrame The dataframe with salary and currency information.
#' @return Creates a filtered dataframe 'manager_usd' containing only USD entries.
manager_usd <- subset(manager, Currency == "USD")
usd_number <- nrow(manager_usd)
# Output
cat("Number of observations after restricting the data:", usd_number, "\n")
```

```
Number of observations after restricting the data: 23374
```

3.4 d. Assume no one starts working before age 18.

```r
# Create a function to calculate the midpoint of age and work experience
#' Convert age and experience range strings to numerical midpoints
#' @description This function processes strings containing numeric ranges, converting them to
#' @param year A string containing a numeric range or a single numeric value.
convert_to_midpoint <- function(year) {
    num <- as.numeric(unlist(regmatches(year, gregexpr("[[:digit:]]+", year))))
    if (length(num) > 1) {
        return(mean(num))
    } else {
        return(num)
    }
}

# Restrict the data to those worked after age 18
# Apply the conversion function to age and experience data
manager_usd$Age <- sapply(manager_usd$Age, convert_to_midpoint)
manager_usd$Overall_Professional_experience_Years <- sapply(manager_usd$Overall_Professional_

manager_usd$Field_Professional_experience_Years <- sapply(manager_usd$Field_Professional_expe

# Filter the dataset to remove entries that do not meet logical age and experience criteria
manager_usd_age <- subset(manager_usd, Age >= 18 &
                          Overall_Professional_experience_Years >= Field_Professional_experie
                          & (Age - 18) >= Overall_Professional_experience_Years)

# Number of observations before restricting the data working after age 18
usd_number <- nrow(manager_usd)
cat("Number of observations before restricting age 18:", usd_number, "\n")
```

```
Number of observations before restricting age 18: 23374
```

```r
# Number of observations after restricting the data working after age 18
usd_age_number <- nrow(manager_usd_age)
cat("Number of observations after restricting age 18:", usd_age_number, "\n")
```

```
Number of observations after restricting age 18: 20206
```

3.5 e. Eliminate any rows with extremely low or extremely high salaries.

To focus on a realistic range of salaries and remove potential outliers, we determined the upper salary limit using the IQR method. This method helps identify and exclude extreme values that might skew our analysis.

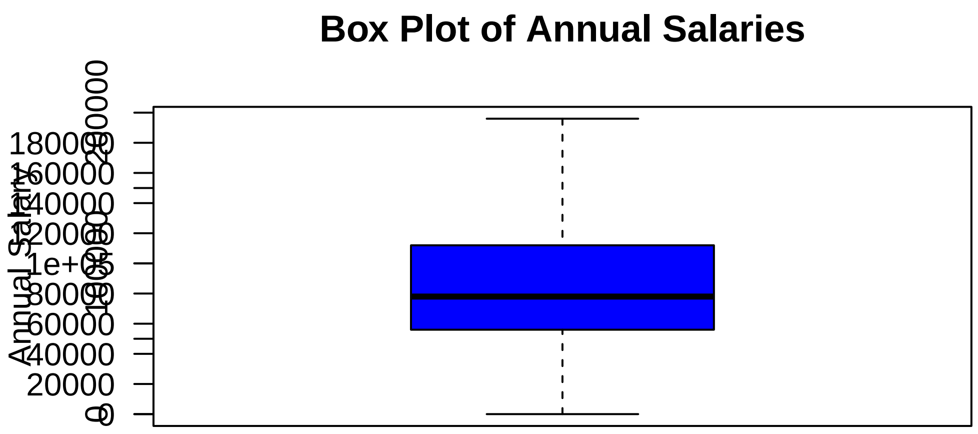
```r
# Calculate the first and third quartiles and IQR
Q1 <- quantile(manager_usd$Annual_Salary, 0.25)
Q3 <- quantile(manager_usd$Annual_Salary, 0.75)
IQR <- Q3 - Q1

# Calculate the upper bound for salaries using the 1.5*IQR rule
upper_bound <- Q3 + 1.5 * IQR

# Print the calculated upper bound to verify
cat("Calculated upper bound for salaries based on IQR method:", upper_bound, "\n")
```

```
Calculated upper bound for salaries based on IQR method: 196000
```

```r
# Create a boxplot to visualize salary distribution with the identified upper bound
boxplot(manager_usd$Annual_Salary, main="Box Plot of Annual Salaries",
        ylab="Annual Salary", col="blue", ylim=c(0, upper_bound), outline=FALSE)
axis(2, at=seq(0, upper_bound, by=20000), labels=seq(0, upper_bound, by=20000), las=1)
```



```r
# Exclude data with annual salaries below $1,000 and above the calculated upper bound
salary_lower <- 1000
manager_usd_age_salary <- subset(manager_usd_age, Annual_Salary >= salary_lower & Annual_Sala
# Output the final sample size after filtering
final_sample_size <- nrow(manager_usd_age_salary)
cat("The final sample size after filtering is:", final_sample_size, "\n")
```

```
The final sample size after filtering is: 19386
```

4 Problem 3 - Palindromic Numbers

4.1 a. Write function isPalindromic that checks if a given positive integer is a palindrome.

```r
#' This function checks if a given positive integer is the same forward and backward,
#' indicating if the number is palindromic.
#' @param number A positive integer to be checked for palindromicity.
#' @return A list containing:
#' @return isPalindromic A logical value indicating if the input is palindromic.
#' @return reversed The input number with its digits reversed.
#' @examples
#' isPalindromic(121) # returns list(isPalindromic = TRUE, reversed = 121)
#' isPalindromic(123) # returns list(isPalindromic = FALSE, reversed = 321)
#' @export
isPalindromic <- function(number) {
  if (!is.numeric(number) || number != as.integer(number) || number <= 0) {
    stop("Input is not a positive integer.")
  }

  # Converts a number to a string
  number_str <- as.character(number)

  # Reverse number
  reversed_str <- rev(strsplit(number_str, "")[[1]])
  reversed_number <- as.integer(paste(reversed_str, collapse = ""))

  # Checks if the number is the same forward and backward
  is_palindromic <- (number_str == paste(reversed_str, collapse = ""))
  return(list(isPalindromic = is_palindromic, reversed = reversed_number))
}
```

```r
# Test
isPalindromic(39951)
```

```
$isPalindromic
[1] FALSE

$reversed
[1] 15993
```

4.2 b. Create a function nextPalindrome that finds the next palindromic number strictly greater than the input.

```r
#' This function calculates the smallest palindromic number that is strictly greater
#' than the provided positive integer.
#' @param number A positive integer to find the next palindrome for.
#' @return The next palindromic number.
#' @examples
#' nextPalindrome(123) # returns 131
#' nextPalindrome(738) # returns 747
#' @export
nextPalindrome <- function(number) {
  if (!is.numeric(number) || number < 1 || number != as.integer(number)) {
    stop("Input is not a positive integer.")
  }
  # Start checking from the next number
  number <- number + 1

  # Adjust numbers that end in 0 to avoid unnecessary checks
  while (number %% 10 == 0) {
    number <- number + 1
  }

  # Find the next palindrome
  while(TRUE) {
    num_str <- as.character(number)
    if (num_str == paste(rev(strsplit(num_str, "")[[1]]), collapse = "")) {
      return(number)
    }
    number <- number + 1
  }
}
```

4.3 c. Find the next palindrome for each of the following

```
#' Apply the nextPalindrome function to a list of numbers
#' @export
numbers <- c(391, 9928, 19272719, 109, 2)
result <- sapply(numbers, nextPalindrome)
for (i in 1:length(numbers)) {
  cat(sprintf("The next palindrome after %d is %d.\n", numbers[i], result[i]))
}
```

```
The next palindrome after 391 is 393.
The next palindrome after 9928 is 9999.
The next palindrome after 19272719 is 19277291.
The next palindrome after 109 is 111.
The next palindrome after 2 is 3.
```

5 References and Data Sources

The datasets used in this project are obtained from the following sources:

- **Wine Data**: This dataset is from the UCI Machine Learning Repository, primarily used for analyzing chemical properties of different types of wine.
- **AskAManager.org Salary Data**: This salary data is from an ongoing salary survey at AskAManager.org, used to explore how various factors affect salary levels.
- **Palindromic Numbers**: Demonstrates programming techniques and logical reasoning through the computation and verification of palindromic numbers.

6 Methods and Implementation

All data analyses were performed in the R environment, employing a range of techniques including data import, data cleaning, statistical analysis, and result visualization.

7 Code and Documentation Repository

- This document and related code are hosted on GitHub for review and sharing purposes. Access link: GitHub Repository Link

8 Acknowledgements

Thanks to the course instructors and teaching assistants for their guidance on this assignment. Thanks to all data providers for supporting open data.

9 Notes

The analyses in this document are for academic purposes only, intended to fulfill the requirements of a statistics course. While every reasonable effort has been made to ensure the accuracy of the analysis results, the content of this document represents only the views of the author.