**Question 1,2,3**

library(readxl)

library(rstudioapi)

# Set working directory to the location of your Excel file

# Set the working directory to the folder where the file is located

setwd("C:/math")

# Verify if the file exists

file.exists("United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xls")

# If the file exists, this should return TRUE

# If the file extension is .xlsx, update the file name accordingly:

data\_file <- "United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xlsx"

# If the file extension is .xls, use:

data\_file <- "United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xls"

# Read the Excel file with the specified range

all\_data <- read\_excel(data\_file, range = "B2:W158")

all\_data

# Helper function to extract salary data by row

get\_salary\_wages <- function(row\_num, data = all\_data) {

return(na.omit(as.numeric(data[row\_num, -1])))

}

get\_salary\_wages

# Ensure that salary\_wages data has 16 points

# Extract salary data from the dataset using get\_salary\_wages()

salary\_wages\_snbodies <- get\_salary\_wages(6) # For small narrowbodies

salary\_wages\_lnbodies <- get\_salary\_wages(45) # For large narrowbodies

salary\_wages\_wbodies <- get\_salary\_wages(84) # For widebodies

salary\_wages\_tfleet <- get\_salary\_wages(123) # For total fleet

# Now, combine the extracted salary data into one sample

# Assuming you want a combined sample of all these salary data sets

salary\_wages\_sample <- c(salary\_wages\_snbodies, salary\_wages\_lnbodies, salary\_wages\_wbodies, salary\_wages\_tfleet)

salary\_wages\_sample

# Check the number of observations in the combined sample

length(salary\_wages\_sample)

# If you need exactly 16 observations, you can either take the first 16, sample 16 randomly, or apply some selection method.

# Randomly select 16 observations from the combined data

set.seed(123) # For reproducibility

salary\_wages\_sample\_16 <- sample(salary\_wages\_sample, 16, replace = FALSE)

salary\_wages\_sample\_16

# View the sample of 16 observations

print(salary\_wages\_sample\_16)

## Assuming these functions and salary data extraction methods have already been defined:

get\_modes <- function(data) {

freq\_table <- table(data)

max\_freq <- max(freq\_table)

modes <- as.numeric(names(freq\_table[freq\_table == max\_freq]))

if (length(modes) == length(data)) {

return(NULL)

}

return(modes)

}

get\_frequency\_distribution <- function(wage\_data) {

# Number of observations

n <- length(wage\_data)

# Calculate k directly as log2(n), and round it up

k <- ceiling(log2(n))

# Calculate class interval (interval >= (max - min)/k)

min\_salary <- min(wage\_data)

max\_salary <- max(wage\_data)

class\_interval <- (max\_salary - min\_salary) / k

class\_interval <- ceiling(class\_interval) # Ensure class interval is a whole number

# Create breakpoints

break\_points <- seq(

min\_salary - (class\_interval / 2), # Start the first break point slightly before the min value

max\_salary + (class\_interval / 2), # End the last break point slightly after the max value

by = class\_interval

)

# Create frequency distribution

salary\_bins <- cut(wage\_data, breaks = break\_points, right = TRUE)

frequency\_distribution <- table(salary\_bins)

return(frequency\_distribution)

}

# Example: Extract salary data from each category

salary\_wages\_snbodies <- get\_salary\_wages(6) # For small narrowbodies

salary\_wages\_lnbodies <- get\_salary\_wages(45) # For large narrowbodies

salary\_wages\_wbodies <- get\_salary\_wages(84) # For widebodies

salary\_wages\_tfleet <- get\_salary\_wages(123) # For total fleet

# Combine the extracted salary data

combined\_salary\_wages <- c(salary\_wages\_snbodies, salary\_wages\_lnbodies, salary\_wages\_wbodies, salary\_wages\_tfleet)

# If you want to take exactly 16 samples, you can sample from combined data

set.seed(123) # For reproducibility

salary\_wages\_sample\_16 <- sample(combined\_salary\_wages, 16, replace = FALSE)

# Get frequency distribution for the sample data

frequency\_distribution\_sample <- get\_frequency\_distribution(salary\_wages\_sample\_16)

# Print the frequency distribution for the sample

cat("Frequency Distribution for Sample of 16 Observations:\n")

print(frequency\_distribution\_sample)

# Perform analysis on the sample

print\_analysis <- function(wage\_data, title) {

mean <- mean(wage\_data)

median <- median(wage\_data)

modes <- get\_modes(wage\_data)

sample\_sd <- sd(wage\_data) # sample

sample\_var <- var(wage\_data) # sample

quartiles <- quantile(wage\_data, probs = c(0.25, 0.5, 0.75))

tenth\_percentile <- quantile(wage\_data, probs = 0.10)

ninth\_decile <- quantile(wage\_data, probs = 0.90)

range <- max(wage\_data) - min(wage\_data)

# Print results

cat("Analysis of ", title, "::\n")

cat("Mean:", mean, "\n")

cat("Median:", median, "\n")

if (is.null(modes) || length(modes) == 0) {

cat("Modes: None\n")

} else {

cat("Modes:", paste(modes, collapse = ", "), "\n")

}

cat("Sample Standard Deviation:", sample\_sd, "\n")

cat("Sample Variance:", sample\_var, "\n")

cat("Quartiles (Q1, Q2, Q3):", quartiles, "\n")

cat("10th Percentile:", tenth\_percentile, "\n")

cat("9th Decile:", ninth\_decile, "\n")

cat("Range:", range, "\n")

cat("\n\n")

}

# Perform analysis on the 16 sample

print\_analysis(salary\_wages\_sample\_16, "Salary Wages Sample of 13 Observations")

# Plot the histogram for the sample data

plot\_histogram <- function(frequency\_distribution, window\_title) {

barplot(frequency\_distribution,

xlab = "Salary Ranges",

ylab = "Frequency",

col = "lightblue",

border = "black",

space = 0, # No space between bars

width = 1, # Adjust width to fill the space better

main = window\_title

)

}

# Plot histogram for the sample frequency distribution

plot\_histogram(frequency\_distribution\_sample, "Histogram of Salary Wages Sample of 26 Observations")

**Output for 1 to 3**

Analysis of Salary Wages Sample of 16 Observations ::

Mean: 603.6513

Median: 584.5249

Modes: None

Sample Standard Deviation: 192.5641

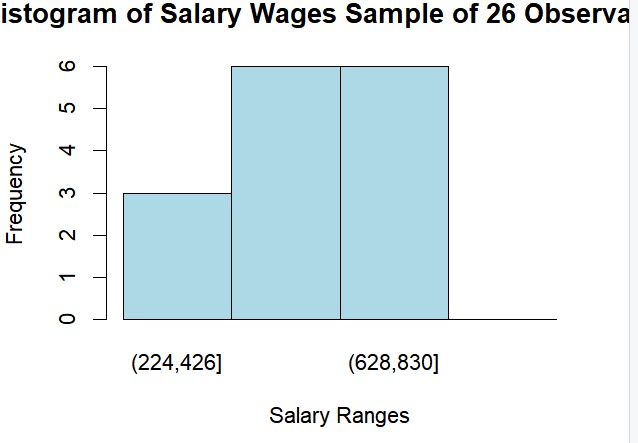
Sample Variance: 37080.93

Quartiles (Q1, Q2, Q3): 508.9872 584.5249 676.4091

10th Percentile: 381.6944

9th Decile: 748.9687

Range: 806.1665



**Question 4**

library(readxl)

library(rstudioapi)

# Set working directory to the location of your Excel file

# Set the working directory to the folder where the file is located

setwd("C:/math")

# Verify if the file exists

file.exists("United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xls")

# If the file exists, this should return TRUE

# If the file extension is .xlsx, update the file name accordingly:

data\_file <- "United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xlsx"

# If the file extension is .xls, use:

data\_file <- "United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xls"

# Read the Excel file with the specified range

all\_data <- read\_excel(data\_file, range = "B2:W158")

all\_data

maintenance\_categories <- c("labor", "materials", "third party", "burden")

years <- 1995:2015

# Function to extract data for a given row number (Maintenance/Load Factor)

get\_data\_by\_row <- function(row\_num) {

return(na.omit(as.numeric(all\_data[row\_num, -1])))

}

get\_maintenace\_category <- function(row\_num) {

labor <- get\_data\_by\_row(row\_num + 1)

materials <- get\_data\_by\_row(row\_num + 2)

third\_party <- get\_data\_by\_row(row\_num + 3)

burden <- get\_data\_by\_row(row\_num + 5)

return(setNames(

c(sum(labor), sum(materials), sum(third\_party), sum(burden)),

maintenance\_categories

))

}

# For ploting Load Factor bar plot

plot\_bar <- function(data, title) {

barplot(data,

main = title,

xlab = "Years",

ylab = "Load Factor (%)",

col = "lightblue",

border = "red"

)

}

# Maintenance and Load Factor row numbers

maintenance\_rows <- c(16, 55, 94, 133)

load\_factor\_rows <- c(34, 73, 112, 151)

fleet\_category <- c(

"small narrowbodies",

"large narrowbodies",

"widebodies",

"total fleet"

)

# Load necessary library

library(RColorBrewer) # For color palettes

# Pie chart for maintenance

windows(width = 1920 / 100, height = 1080 / 100) # Set window size

par(mfrow = c(2, 2), oma = c(0, 0, 3, 0))

# Define a color palette

colors <- brewer.pal(4, "Set3") # Using RColorBrewer for a set of 4 distinct colors

# Create pie charts for each maintenance category with enhancements

lapply(1:4, function(i) {

data <- get\_maintenace\_category(maintenance\_rows[i])

# Calculate percentages

percentages <- round(100 \* data / sum(data), 1)

# Create labels with category names and percentages

labels <- paste0(names(data), ": ", percentages, "%")

# Create pie chart

pie(data,

labels = labels, # Use labels with percentages

main = paste("Maintenance Costs for", fleet\_category[i]), # Descriptive title

col = colors, # Set colors for slices

border = "white") # Add border to slices

})

# Add an outer title for all pie charts

mtext("Maintenance Cost Distribution", outer = TRUE, cex = 1.5)

# Reset plotting parameters to default

par(mfrow = c(1, 1))

# bar chart for load factor

windows(width = 1920 / 100, height = 1080 / 100) # Set window size

par(mfrow = c(2, 2), oma = c(0, 0, 3, 0))

lapply(1:4, function(i) {

data <- setNames(get\_data\_by\_row(load\_factor\_rows[i]), years)

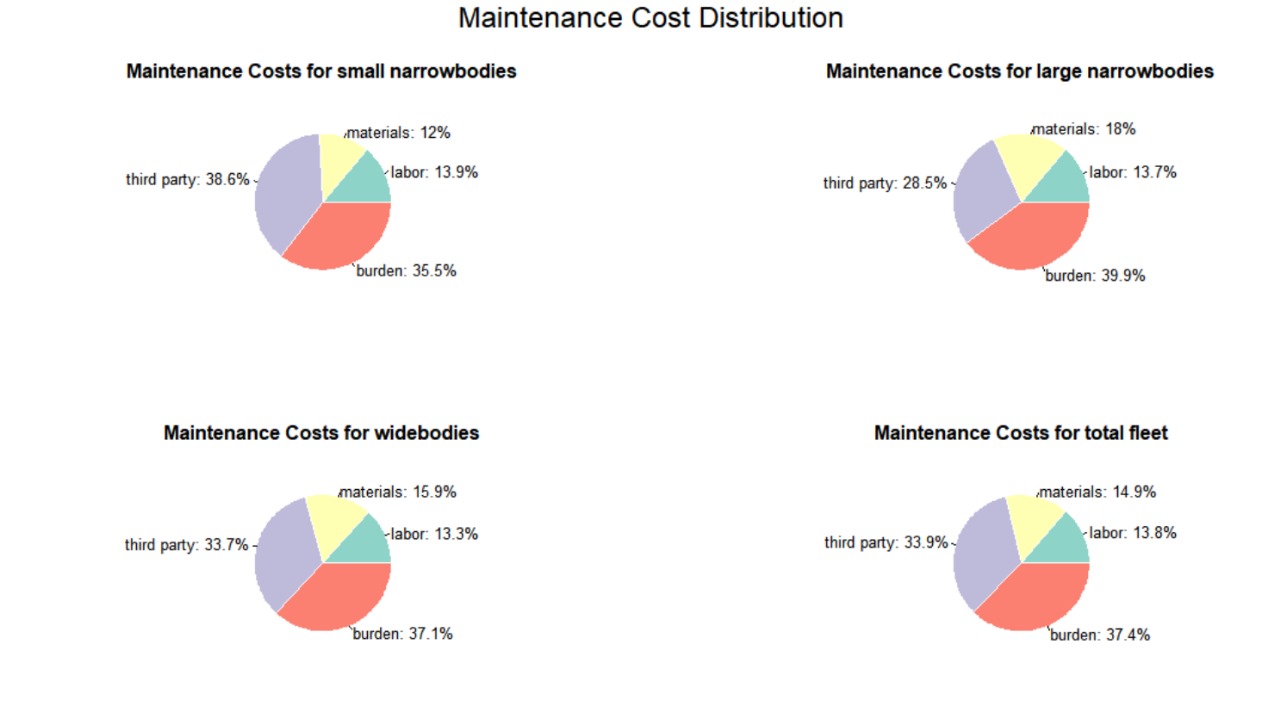
plot\_bar(data, fleet\_category[i])

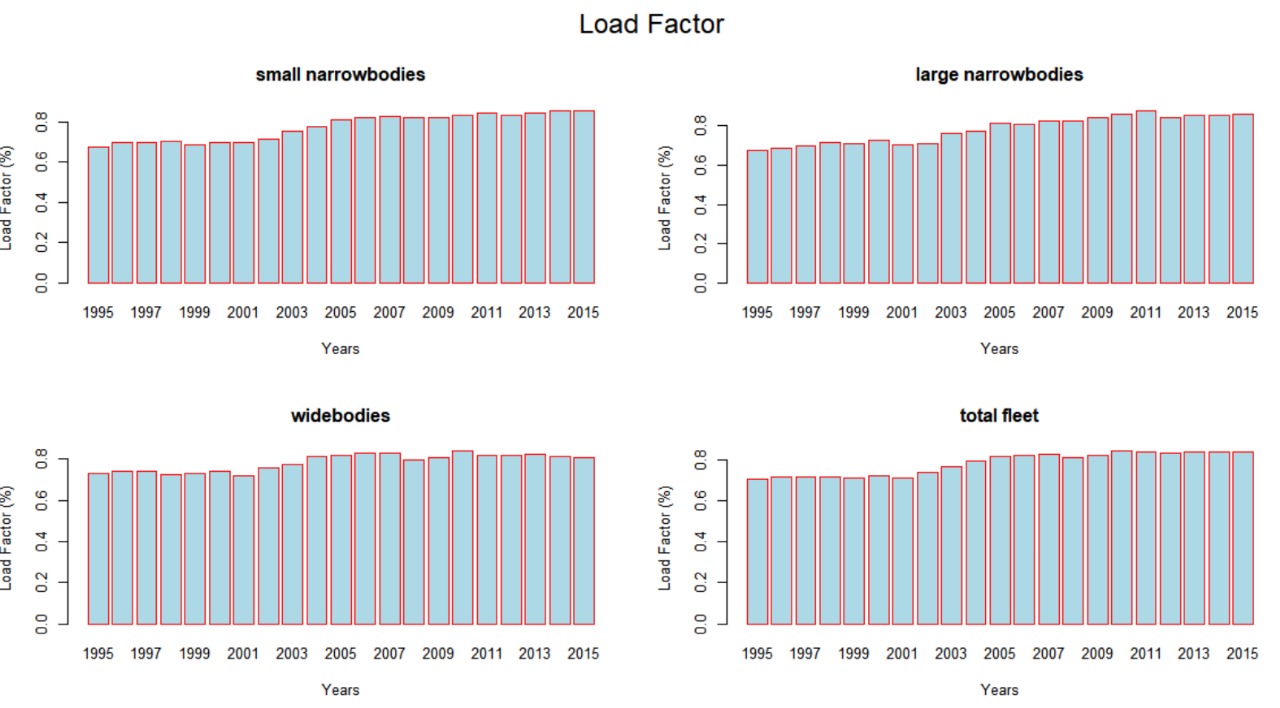
})

mtext("Load Factor", outer = TRUE, cex = 1.5)

par(mfrow = c(1, 1))

**Output for Q4:**





**Question 5**

library(readxl)

library(rstudioapi)

# Set working directory to the location of your Excel file

# Set the working directory to the folder where the file is located

setwd("C:/math")

# Verify if the file exists

file.exists("United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xls")

# If the file exists, this should return TRUE

# If the file extension is .xlsx, update the file name accordingly:

data\_file <- "United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xlsx"

# If the file extension is .xls, use:

data\_file <- "United\_Airlines\_Aircraft\_Operating\_Statistics-\_Cost\_Per\_Block\_Hour\_(Unadjusted).xls"

# Read the Excel file with the specified range

all\_data <- read\_excel(data\_file, range = "B2:W158")

all\_data

# define categories

daily\_utilization\_categories <- c("Block hours", "Airborne hours", "Departures")

ownership\_categories <- c("Rental", "Depreciation and Amortization")

purchased\_goods\_categories <- c("Fuel/Oil", "Insurance", "Other (inc. Tax)")

fleet\_category <- c(

"small narrowbodies",

"large narrowbodies",

"widebodies",

"total fleet"

)

# row numbers

purchased\_goods\_rows <- c(16, 55, 94, 133) - 5

ownership\_rows <- purchased\_goods\_rows + 12

daily\_utilization\_rows <- ownership\_rows + 13

get\_data\_by\_row <- function(row\_num) {

if (row\_num > nrow(all\_data)) {

stop("Row number exceeds data range.")

}

return(na.omit(as.numeric(all\_data[row\_num, -1])))

}

get\_category\_data <- function(row\_num, categories) {

rows\_data <- lapply(

seq\_along(categories),

function(i) get\_data\_by\_row(row\_num + i)

)

costs <- unlist(rows\_data)

category <- factor(rep(categories, sapply(rows\_data, length)))

return(data.frame(costs = costs, category = category))

}

box\_plot <- function(data, title, ylab) {

boxplot(costs ~ category,

data = data,

main = title,

col = "orange",

ylab = ylab,

border = "red"

)

}

plot\_category <- function(rows, categories, title, ylab) {

windows(width = 1920 / 100, height = 1080 / 100) # Set window size

par(mfrow = c(2, 2), oma = c(0, 0, 3, 0))

# Create the box plot using the formula interface

lapply(

seq\_along(rows),

function(i) {

box\_plot(

get\_category\_data(

rows[i], categories

), fleet\_category[i], ylab

)

}

)

mtext(title, outer = TRUE, cex = 1.5)

par(mfrow = c(1, 1))

}

plot\_category(

purchased\_goods\_rows,

purchased\_goods\_categories,

"Purchased Goods",

"Hours"

)

plot\_category(

ownership\_rows,

ownership\_categories,

"Aircraft Ownership",

"Cost ($)"

)

plot\_category(

daily\_utilization\_rows,

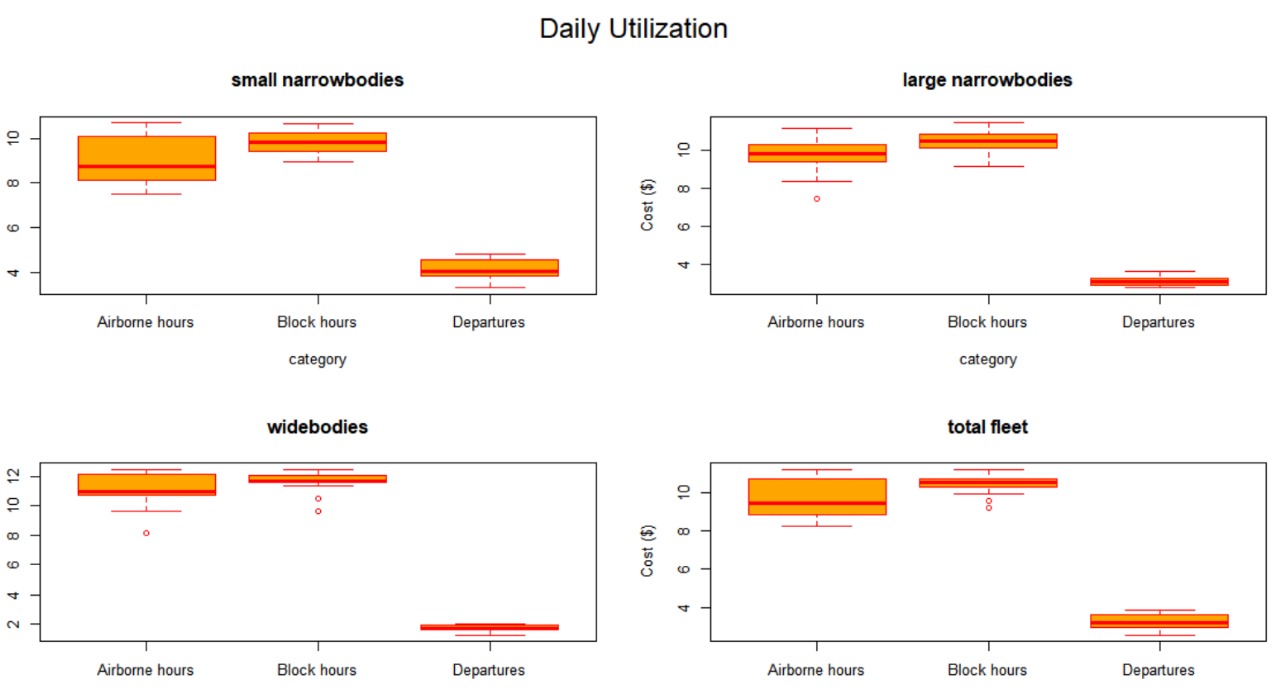
daily\_utilization\_categories,

"Daily Utilization",

"Cost ($)"

)

**Output for Q5:**

****

****

