**STATS LAB**  
1Q

install.packages("tidyverse")

library(tidyverse)

# Load dataset

data(mtcars)

head(mtcars)

# Compute correlation between mpg and hp

correlation <- cor(mtcars$mpg, mtcars$hp)

print(paste("Correlation between mpg and hp is:", round(correlation, 2)))

# Scatter plot with regression line

library(ggplot2)

ggplot(data = mtcars, aes(x = hp, y = mpg)) +

geom\_point(shape = 4, color = "blue", size = 3) +

geom\_smooth(method = "lm", se = FALSE) +

labs(title = "MPG vs HorsePower",

x = "HorsePower (HP)",

y = "Miles per Gallon (mpg)") +

theme\_minimal()

View(mtcars)

--------------------------------------------------------------------------------------------------------------

2Q

data(mtcars)

# Correlation

correlation <- cor(mtcars$mpg, mtcars$hp)

print(paste("Correlation between mpg and hp is:", round(correlation, 2)))

# Q-Q Plot

library(ggplot2)

ggplot(data = mtcars, aes(sample = mpg)) +

stat\_qq() +

stat\_qq\_line() +

labs(title = "Q-Q plot for MPG",

x = "Theoretical Quantiles",

y = "MPG (sample)") +

theme\_minimal()

--------------------------------------------------------------------------------------------------------------

3Q

# Install tidyverse (only once; skip if already installed)

#install.packages("tidyverse")

# Load tidyverse

library(tidyverse)

# Read your CSV file into R

# (Make sure pricequotes.csv is in your working directory)

data <- read\_csv("C:/Users/HP/Downloads/pricequotes.csv")

# Quick check of first few rows

head(data)

# View full dataset in spreadsheet viewer

View(data)

# Summary statistics

print(summary(data))

# Find n

n\_barry <- length(data$Barry\_Price)

n\_mary <- length(data$Mary\_Price)

# Standard deviation & Standard error

sd\_barry <- sd(data$Barry\_Price)

sd\_mary <- sd(data$Mary\_Price)

se\_barry <- sd\_barry / sqrt(n\_barry)

se\_mary <- sd\_mary / sqrt(n\_mary)

cat("Mary: SD =", round(sd\_mary, 2), "| SE =", round(se\_mary, 2), "\n")

cat("Barry: SD =", round(sd\_barry, 2), "| SE =", round(se\_barry, 2), "\n")

# Boxplot comparison

ggplot(data, aes(x = "Barry", y = Barry\_Price)) +

geom\_boxplot(fill = "skyblue") +

geom\_boxplot(aes(x = "Mary", y = Mary\_Price), fill = "lightgreen") +

labs(title = "BoxPlot of Price Quotes",

x = "Person",

y = "Price") +

theme\_minimal()

# Suppose your dataset is stored in 'data'

#colnames(data)

--------------------------------------------------------------------------------------------------------------

4Q

# Load required libraries

library(dplyr)

library(ggplot2)

# Read the dataset

df <- read.csv("C:/Users/HP/Downloads/treatmentfacility.csv")

# Fix messy column name (rename "CI...." to "CI")

names(df)[names(df) == "CI...."] <- "CI"

# Convert Reengineer column to factor with order

df$Reengineer <- factor(df$Reengineer, levels = c("Prior", "Post"))

# Summary statistics

summary\_stats <- df %>%

group\_by(Reengineer) %>%

summarize(

n = n(),

mean\_turnover = mean(Employee\_Turnover, na.rm = TRUE),

sd\_turnover = sd(Employee\_Turnover, na.rm = TRUE),

mean\_TRFF = mean(TRFF, na.rm = TRUE),

sd\_TRFF = sd(TRFF, na.rm = TRUE),

mean\_CI = mean(CI, na.rm = TRUE),

sd\_CI = sd(CI, na.rm = TRUE)

)

print(summary\_stats)

# Boxplot for Employee Turnover

ggplot(df, aes(x = Reengineer, y = Employee\_Turnover, fill = Reengineer)) +

geom\_boxplot() +

labs(title = "Employee Turnover Before and After Reengineering") +

theme\_minimal()

# Boxplot for TRFF

ggplot(df, aes(x = Reengineer, y = TRFF, fill = Reengineer)) +

geom\_boxplot() +

labs(title = "TRFF Before and After Reengineering") +

theme\_minimal()

# Boxplot for CI

ggplot(df, aes(x = Reengineer, y = CI, fill = Reengineer)) +

geom\_boxplot() +

labs(title = "CI Before and After Reengineering") +

theme\_minimal()

#colnames(df)

--------------------------------------------------------------------------------------------------------------

5Q

library(readr)

library(ggplot2)

library(dplyr)

df <- read.csv("C:/Users/HP/Downloads/baggagecomplaints .csv")

df <- df %>%

mutate(Rate = 100 \* Baggage / Enplaned)

print(summary(df[c("Baggage", "Rate")]))

summary\_airline <- df %>%

group\_by(Airline) %>%

summarize(

total\_months = n(),

total\_passengers = sum(Enplaned),

mean\_complaints = mean(Baggage),

median\_complaints = median(Baggage),

sd\_complaints = sd(Baggage),

mean\_rate = mean(Rate),

median\_rate = median(Rate),

sd\_rate = sd(Rate),

min\_rate = min(Rate),

max\_rate = max(Rate)

)

print(summary\_airline)

# Yearly average

yearly\_avg <- df %>%

group\_by(Year, Airline) %>%

summarise(avg\_complaints = mean(Baggage), .groups = "drop")

ggplot(yearly\_avg, aes(x = Year, y = avg\_complaints, color = Airline)) +

geom\_line(linewidth = 1.2) +

geom\_point(size = 2, color = "black") +

theme\_minimal() +

labs(title = "Yearly Avg Baggage Complaints", x = "Year", y = "Avg Complaints")

#View(df)

-------------------------------------------------------------------------------------------------------------

6Q

library(ggplot2)

library(dplyr)

library(readr)

# Load dataset

data <- read.csv("C:/Users/HP/Downloads/medicalmalpractice.csv")

summary(data$Amount)

# Histogram (log scale)

ggplot(data, aes(x = log10(Amount))) +

geom\_histogram(fill = "lightblue", bins = 20) +

labs(title = "Histogram of Claim Amounts (Log Scale)",

x = "Log Amount", y = "Frequency")

# Boxplot for top 3 specialties

top3\_specialty <- data %>%

count(Specialty, name = "n") %>%

slice\_max(n, n = 3) %>%

pull(Specialty)

data %>%

filter(Specialty %in% top3\_specialty) %>%

ggplot(aes(x = Specialty, y = Amount, fill = Specialty)) +

geom\_boxplot() +

coord\_flip() +

theme\_minimal()

# Percentage of specific specialties

total <- length(data$Specialty)

spec\_percent <- data %>%

group\_by(Specialty) %>%

summarise(n = n(), pct = 100 \* n / total) %>%

filter(Specialty %in% c("Anesthesiology", "Dermatology", "Orthopedic Surgery"))

print(spec\_percent)

# Correlation between Age and Amount

cor.test(data$Age, data$Amount)

#View(data)

#head(data)

-------------------------------------------------------------------------------------------------------------

7Q

# Load libraries

library(readr)

library(tidyr)

library(dplyr)

library(ggplot2)

# ===============================

# 1. Load dataset

# ===============================

data <- read\_csv("C:/Users/HP/Downloads/fishstory.csv")

# Check column names

print(colnames(data))

# ===============================

# 2. Paired t-test (1980 vs 1970)

# ===============================

t\_test\_result <- t.test(data$Price1980, data$Price1970, paired = TRUE)

print(t\_test\_result)

# Mean difference

mean\_diff <- mean(data$Price1980 - data$Price1970, na.rm = TRUE)

cat("Mean difference in price (1980 - 1970):", mean\_diff, "\n")

# Confidence interval

cat("95% Confidence Interval:", t\_test\_result$conf.int, "\n")

# ===============================

# 3. Convert to long format

# ===============================

long\_data <- data %>%

pivot\_longer(cols = c(Price1970, Price1980),

names\_to = "Year",

values\_to = "Price") %>%

mutate(Year = ifelse(Year == "Price1970", 1970, 1980))

# ===============================

# 4. Line plot (average fish prices per year)

# ===============================

ggplot(long\_data, aes(x = Year, y = Price)) +

stat\_summary(fun = mean, geom = "line", color = "orange", linewidth = 1.2) +

stat\_summary(fun = mean, geom = "point", color = "orange", size = 3) +

labs(title = "Average Fish Prices (1970 vs 1980)",

x = "Year",

y = "Average Price") +

theme\_minimal()

# ===============================

# 5. Boxplot (distribution across species per year)

# ===============================

ggplot(long\_data, aes(x = factor(Year), y = Price)) +

geom\_boxplot(fill = "skyblue", color = "darkblue", outlier.colour = "red") +

labs(title = "Distribution of Fish Prices (1970 vs 1980)",

x = "Year",

y = "Price") +

theme\_minimal()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

correct code

#install.packages("tidyverse")

library(tidyverse)

# Read CSV

data <- read\_csv("C:/Users/HP/Downloads/fishstory.csv")

# Paired t-test

t\_test\_result <- t.test(data$Price1980, data$Price1970, paired = TRUE)

print(t\_test\_result)

# Mean difference

mean\_diff <- mean(data$Price1980 - data$Price1970, na.rm = TRUE)

cat("Mean difference in price (1980 - 1970):", mean\_diff, "\n")

# Confidence interval

cat("95% Confidence Interval:", t\_test\_result$conf.int, "\n")

-------------------------------------------------------------------------------------------------------------

8Q

#1

library(readr)

library(dplyr)

# Load the dataset

data <- read\_csv("C:/Users/HP/Downloads/patient-feedback.csv")

# Summary statistics for all numeric columns

summary(data)

#View(data)

#2

# Load required libraries

install.packages("GGally")

library(readr)

library(GGally)

# Load the Fitness dataset (you can replace with your file)

fitness <- read\_csv("C:/Users/HP/Downloads/CardioGoodFitness.csv")

# Select only continuous (numeric) variables

numeric\_data <- fitness[sapply(fitness, is.numeric)]

# Scatterplot matrix

ggpairs(numeric\_data)

# Compute correlation matrix

cor\_matrix <- cor(numeric\_data)

# Print the correlation matrix

print(cor\_matrix)

# Find strongest positive and negative correlation pairs

cor\_matrix[lower.tri(cor\_matrix, diag = TRUE)] <- NA # Keep only upper triangle

cor\_values <- as.data.frame(as.table(cor\_matrix))

cor\_values <- na.omit(cor\_values)

# Strongest positive correlation

strongest\_pos <- cor\_values[which.max(cor\_values$Freq), ]

# Strongest negative correlation

strongest\_neg <- cor\_values[which.min(cor\_values$Freq), ]

# Print results

cat("Strongest Positive Correlation:\n")

print(strongest\_pos)

cat("\nStrongest Negative Correlation:\n")

print(strongest\_neg)

#View(fitness)

-------------------------------------------------------------------------------------------------------------

9Q

import pandas as pd

from scipy.stats import spearmanr

# Scores of students

SMIP = [70, 46, 94, 34, 20, 86, 18, 12, 56, 64, 42]

DBMS = [60, 66, 90, 46, 16, 98, 24, 8, 32, 54, 62]

# Create a DataFrame

df = pd.DataFrame({

'SMIP': SMIP,

'DBMS': DBMS

})

# Calculate Spearman Rank Correlation

corr, p\_value = spearmanr(df['SMIP'], df['DBMS'])

print("Spearman Rank Correlation Coefficient:", corr)

print("P-value:", p\_value)

# Interpretation

if corr > 0:

relation = "positive"

elif corr < 0:

relation = "negative"

else:

relation = "no"

print(f"Interpretation: The Spearman correlation ({corr:.3f}) indicates a {relation} association between SMIP and DBMS scores.")

-------------------------------------------------------------------------------------------------------------

10Q

# Load the required libraries

import pandas as pd

import statsmodels.api as sm

import matplotlib.pyplot as plt

import warnings

# Ignore statistical test warnings for small sample sizes

warnings.filterwarnings("ignore")

# Load the dataset

data = {

'TV': [1, 2, 4, 7, 9, 11, 15],

'Sales': [2, 4, 6, 9, 12, 34, 45]

}

df = pd.DataFrame(data)

# Define X (independent variable) and Y (dependent variable)

X = df['TV']

Y = df['Sales']

# Add constant to predictor variable

X = sm.add\_constant(X)

# Fit the regression model

model = sm.OLS(Y, X).fit()

# Print summary of regression results

print(model.summary())

# Plot the regression line

plt.scatter(df['TV'], df['Sales'], color='blue', label='Actual Data')

plt.plot(df['TV'], model.predict(X), color='red', label='Fitted Line')

plt.title("Linear Regression: Sales vs TV Advertising Budget")

plt.xlabel("TV Advertising Budget")

plt.ylabel("Sales")

plt.legend()

plt.show()

-------------------------------------------------------------------------------------------------------------

11Q

import pandas as pd

from statsmodels.tsa.holtwinters import ExponentialSmoothing

import matplotlib.pyplot as plt

# Load dataset

df = pd.read\_csv("C:/Users/HP/Downloads/AusAntidiabeticDrug.csv", parse\_dates=['ds'])

# Set datetime index

df.set\_index('ds', inplace=True)

df.index.freq = 'MS' # Monthly start frequency

# Fit Holt-Winters model

model = ExponentialSmoothing(df['y'], trend="add", seasonal="add", seasonal\_periods=12)

fit = model.fit()

# Forecast next 24 months

forecast = fit.forecast(24)

# Plot

plt.figure(figsize=(12,6))

plt.plot(df['y'], label="Actual")

plt.plot(fit.fittedvalues, label="Fitted")

plt.plot(forecast, label="Forecast", color='red')

plt.legend()

plt.show()

-------------------------------------------------------------------------------------------------------------

12Q

# Load dataset

data(mtcars)

# Convert cylinders to factor

mtcars$cyl <- as.factor(mtcars$cyl)

# --- 1. ANOVA ---

anova\_result <- aov(mpg ~ cyl, data = mtcars)

summary(anova\_result)

# --- 2. Mann-Whitney Test (4 vs 6 cylinders) ---

cyl4 <- subset(mtcars, cyl == "4")$mpg

cyl6 <- subset(mtcars, cyl == "6")$mpg

mann\_whitney\_result <- wilcox.test(cyl4, cyl6)

mann\_whitney\_result

# --- 3. Kruskal-Wallis Test ---

kruskal\_result <- kruskal.test(mpg ~ cyl, data = mtcars)

kruskal\_result