library(GGally)

library(ggplot2) # For creating visualizations

library(dplyr) # For data manipulation

library(tidyr)

library(plm)

library(psych)

library(forecast)

library(corrplot)

library(readxl)

my\_data <- read\_excel("CD.xlsx")

data=my\_data

missing\_counts <- colSums(is.na(my\_data))

print(missing\_counts)

# Calculate summary statistics

# Custom mode function

calculate\_mode <- function(x) {

uniq\_x <- unique(x)

uniq\_x[which.max(tabulate(match(x, uniq\_x)))]

}

# Calculate summary statistics

summary\_stats <- data.frame(

Variable = colnames(data)[3:8],

Mean = sapply(data[, 3:8], mean, na.rm = TRUE),

Median = sapply(data[, 3:8], median, na.rm = TRUE),

Mode = sapply(data[, 3:8], calculate\_mode),

SD = sapply(data[, 3:8], sd, na.rm = TRUE),

Skewness = sapply(data[, 3:8], skew, na.rm = TRUE),

Kurtosis = sapply(data[, 3:8], kurtosi, na.rm = TRUE)

)

summary\_stats

# Identify numeric columns

numeric\_columns <- sapply(MISSING\_CD, is.numeric)

# Impute missing values with the mean for numeric columns

for (col in colnames(MISSING\_CD[, numeric\_columns])) {

MISSING\_CD[, col][is.na(MISSING\_CD[, col])] <- mean(MISSING\_CD[, col], na.rm = TRUE)

}

#Correlation analysis

# Calculate the correlation matrix

correlation\_matrix <- cor(my\_data[, c("GDP", "EEL", "PH", "UR", "P", "LE")],use = "complete.obs")

correlation\_matrix

# Linear Regression

lm\_model <- lm(LE ~ GDP + EEL, data = my\_data)

# Summary of the Linear Regression Model

summary(lm\_model)

# Perform Multiple Regression

multiple\_lm\_model <- lm(GDP ~ + EEL + PH + UR + P + LE, data = my\_data)

# Summary of the Multiple Regression Model

summary(multiple\_lm\_model)

# LOGISTICE REGRESSION

# Create a binary variable

my\_data$PH\_binary <- ifelse(my\_data$PH > 1.0, 1, 0)

# Logistic Regression

logistic\_model <- glm(PH\_binary ~ GDP + EEL, data = my\_data, family = binomial)

# Summary of the Logistic Regression Model

summary(logistic\_model)

# Create a correlation plot with a heatmap

corrplot(correlation\_matrix, method = "color")

# Print the correlation matrix

print(correlation\_matrix)

# Create two vectors for life expectancy (LE) based on PH groups

le\_below\_average <- my\_data$LE[my\_data$PH < mean(my\_data$PH)]

le\_above\_average <- my\_data$LE[my\_data$PH >= mean(my\_data$PH)]

# Perform the two-sample t-test

t\_test\_result <- t.test(le\_below\_average, le\_above\_average)

# Print the t-test result

print(t\_test\_result)

# Calculate the median GDP

median\_gdp <- median(my\_data$GDP)

# Use the median as the threshold

threshold <- median\_gdp

# Check the value of the threshold

threshold

# Hypothesis 2: Is there a significant difference in EEL between high GDP and low GDP countries?

# Null Hypothesis (H0): There is no significant difference.

# Alternative Hypothesis (H1): There is a significant difference.

# Subset the data into high GDP and low GDP countries based on a threshold

high\_gdp\_countries <- my\_data[my\_data$GDP >= threshold, ]

low\_gdp\_countries <- my\_data[my\_data$GDP < threshold, ]

# Perform a two-sample t-test to compare EEL between the two groups

t\_test\_result <- t.test(high\_gdp\_countries$EEL, low\_gdp\_countries$EEL)

# Display the results

t\_test\_result

# Load necessary libraries (if not already loaded)

# install.packages("dplyr") # Uncomment and run if you haven't installed the library

#TIME SERIE PLOT

# Load necessary libraries

gdp\_time\_series <- ts(my\_data$GDP, frequency = 1)

# Plot the GDP time series

plot(gdp\_time\_series, main = "GDP Time Series", ylab = "GDP")

# POVERTY TIME SERIES PLOT

PH\_time\_series <- ts(my\_data$PH, frequency = 1)

# Plot the Poverty Headcount(% Population) time series

plot(PH\_time\_series, main = "Poverty Headcount(% Population)

Time Series", ylab = "Poverty Headcount")

#UNEMPLOYMENT TIME SERIES PLOT

UR\_time\_series <- ts(my\_data$UR, frequency = 1)

# Plot the GDP time series

plot(UR\_time\_series, main = "Unemployment Rate

Time Series", ylab = "Unemployment Rate")

# Line plot for GDP over time

my\_data$Year = as.factor(my\_data$Year)

ggplot(my\_data, aes(x = Year, y = GDP, group = Country, color = Country)) +

geom\_line() +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "GDP Over Time",

x = "Year",

y = "GDP")+

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

# Scatter plot for GDP vs. Life Expectancy

ggplot(my\_data, aes(x = GDP, y = LE)) +

geom\_point() +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "GDP vs. Life Expectancy",

x = "GDP",

y = "Life Expectancy")

# Histogram for Unemployment Rate

ggplot(my\_data, aes(x = UR)) +

geom\_histogram(binwidth = 1, fill = "blue", color = "black") +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "Distribution of Unemployment Rate",

x = "Unemployment Rate",

y = "Frequency")

# Create a line plot for Life Expectancy over time

ggplot(my\_data, aes(x = Year, y = LE,group = Country, color = Country)) +

geom\_line() +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "Life Expectancy Over Time",

x = "Year",

y = "Life Expectancy")

# Create a bar plot for Population Growth Rate by Country

ggplot(my\_data, aes(x = Country, y = P, color = Country)) +

geom\_bar(stat = "identity") +

labs(title = "Population Growth Rate by Country",

x = "Country",

y = "Population Growth Rate") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))+

theme(plot.title = element\_text(hjust = 0.5))

# Create a box plot for Unemployment Rate by Country

ggplot(my\_data, aes(x = Country, y = UR, color = Country)) +

geom\_boxplot() +

labs(title = "Unemployment Rate by Country",

x = "Country",

y = "Unemployment Rate") +

theme(plot.title = element\_text(hjust = 0.5))+

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

# Create a line plot for Education Enrollment over time

my\_data$Year <- as.factor(my\_data$Year)

ggplot(my\_data, aes(x = Year, y = EEL,group = Country, color = Country)) +

geom\_line() +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "Education Enrollment Over Time",

x = "Year",

y = "Education Enrollment")

# Pair plot for multiple variables

my\_data %>%

select(GDP, EEL, PH, UR, LE) %>%

ggpairs() +

ggtitle("Pair Plot of Multiple Variables") +

theme(plot.title = element\_text(hjust = 0.5))

# Time series plot for Population

ggplot(my\_data, aes(x = Year, y = P, group = 1)) +

geom\_line() +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "Population Over Time",

x = "Year",

y = "Population")

# Correlation matrix

data=(my\_data[, c("GDP", "EEL", "PH", "UR", "LE")])

ggcorr(data, method = c("everything", "pearson"))

# Compute the correlation matrix

correlation\_matrix <- cor(data, method = "pearson")

# Create a scatterplot with correlation value

attach(data)

ggplot(data, aes(x = PH, y = GDP))+

geom\_point() +

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

annotate("text", x = max(data$PH), y = max(data$GDP),

label = paste("Correlation:", round(cor(data$PH, data$GDP), 2)),

hjust = 1, vjust = 1, size = 4, color = "blue") +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "Scatterplot of Poverty Headcount vs. GDP",

x = "Poverty Headcount(% Population)",

y = "GDP(US $)")

# Heatmap for the correlation matrix

ggplot(data = data.frame(x = colnames(cor\_matrix))) +

geom\_tile(aes(x = x, y = colnames(cor\_matrix), fill = cor\_matrix)) +

theme\_minimal() +

theme(plot.title = element\_text(hjust = 0.5))+

scale\_fill\_gradient(low = "white", high = "blue") +

labs(title = "Correlation Heatmap")

# Boxplot of Life Expectancy by Country

ggplot(my\_data, aes(x = Country, y = LE, color = Country)) +

geom\_boxplot() +

coord\_flip() +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "Life Expectancy by Country",

x = "Country",

y = "Life Expectancy")

# Boxplot of GDP by Year

ggplot(my\_data, aes(x = as.factor(Year), y = GDP, color = Country)) +

geom\_boxplot() +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "GDP by Year",

x = "Year",

y = "GDP")

# Create a scatter plot for Poverty Rate vs. GDP

ggplot(my\_data, aes(x = GDP, y = PH, color = "blue")) +

geom\_point() +

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "Poverty Rate vs. GDP",

x = "GDP",

y = "Poverty Rate")

# Perform a simple linear regression

model <- lm(LE ~ GDP, data = my\_data)

# Check the summary of the regression model

summary(model)

# Hypothesis testing for GDP's effect on life expectancy

# Null Hypothesis (H0): There is no association between GDP and life expectancy

# Alternative Hypothesis (H1): There is an association between GDP and life expectancy

# Perform a hypothesis test using the F-statistic

test\_result <- anova(model)

p\_value <- test\_result$"Pr(>F)"[1]

# Set the significance level (alpha)

alpha <- 0.05

# Check if the p-value is less than the significance level

if (p\_value < alpha) {

cat("Reject the null hypothesis: There is a statistically significant association between GDP and life expectancy.")

} else {

cat("Fail to reject the null hypothesis: There is no statistically significant association between GDP and life expectancy.")

}

# Visualize the regression line and scatter plot

ggplot(my\_data, aes(x = GDP, y = LE,color = "blue")) +

geom\_point() +

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

theme(plot.title = element\_text(hjust = 0.5))+

labs(title = "Regression of GDP on Life Expectancy",

x = "GDP",

y = "Life Expectancy")

# For panel data analysis

# Create a panel data model with fixed-effects

fixed\_effects\_model <- plm(LE ~ GDP + EEL + PH + UR + P, data = my\_data, model = "within")

# Summarize the fixed-effects model

summary(fixed\_effects\_model)

# Create a panel data model with random-effects

random\_effects\_model <- plm(LE ~ GDP + EEL + PH + UR + P, data = my\_data, model = "random")

# Summarize the random-effects model

summary(random\_effects\_model)

#Prediction

# Fit a multiple linear regression model

model <- lm(GDP ~ EEL + PH + UR + P + LE, data = my\_data)

# Summary of the regression model

summary(model)

# Make predictions based on the model

predicted\_gdp <- predict(model, newdata = my\_data)

# Add the predicted GDP values to the original dataset

my\_data$Predicted\_GDP <- predicted\_gdp

# View the updated dataset with predicted GDP values

View(my\_data)