Project_final

Enock Mbaraka

2024-07-29

Introduction

This document presents an analysis of Gross Capital Formation based on various economic factors.

#Loading and Preparing Data

```
# loading the necessary libraries
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(psych)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
library(reshape2)
library(knitr)
library(broom)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: ’zoo’

## The following objects are masked from ’package:base’:
##
##     as.Date, as.Date.numeric
```

```
# Checking the current working directory
getwd()
```

```
## [1] "/home/rstudio/Datasets"
```

```
# Change the working directory
setwd("/home/rstudio/Datasets")

# Verify the change
getwd()
```

```
## [1] "/home/rstudio/Datasets"
```

```
# list files in the working directory
list.files()
```

```
## [1] "project_dataset_final.csv" "project_final.html"
## [3] "project_final.pdf"         "project_final.Rmd"
## [5] "project_work.html"         "project_work.pdf"
## [7] "project_work.Rmd"          "project_work1.Rmd"
```

```
# Import the dataset
df <- read_csv("project_dataset_final.csv")
```

```
## Rows: 53 Columns: 6

## -- Column specification -------------------------------------------------------
## Delimiter: ","
## dbl (6): Year, Real_Interest_Rate, Domestic_Savings, Government_Consumption_...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
# View the first few rows of the dataset
head(df)
```

```
## # A tibble: 6 x 6
##    Year Real_Interest_Rate Domestic_Savings Government_Consumptio~1 Labour_Force
##   <dbl>              <dbl>            <dbl>                   <dbl>        <dbl>
## 1  1970               6.22             23.6                    16.3         73.2
## 2  1971              20.1              17.4                    18.0         73.4
## 3  1972               7.70             20.2                    17.6         73.5
## 4  1973              -1.09             24.5                    16.5         73.6
## 5  1974              -5.64             18.5                    17.0         73.6
## 6  1975              -1.64             13.5                    18.3         73.6
## # i abbreviated name: 1: Government_Consumption_Expenditure
## # i 1 more variable: Gross_Capital_Formation <dbl>
```

```r
# Print the column names of the dataframe
colnames(df)
```

```
## [1] "Year"                               "Real_Interest_Rate"
## [3] "Domestic_Savings"                   "Government_Consumption_Expenditure"
## [5] "Labour_Force"                       "Gross_Capital_Formation"
```

Data Cleaning

Handling missing values in dataset

```r
# Total number of missing values in the dataset
total_missing_values <- sum(is.na(df))
print(paste("Total missing values in the dataset:", total_missing_values))
```

```
## [1] "Total missing values in the dataset: 0"
```

```r
# Number of missing values in each column
missing_values_per_column <- colSums(is.na(df))
print("Missing values per column:")
```

```
## [1] "Missing values per column:"
```

```r
print(missing_values_per_column)
```

```
##                                Year                 Real_Interest_Rate
##                                   0                                  0
##                    Domestic_Savings Government_Consumption_Expenditure
##                                   0                                  0
##                        Labour_Force            Gross_Capital_Formation
##                                   0                                  0
```

```r
# Get a detailed summary of the modified dataset
dataset_summary <- describe(df)

# Print the summary statistics
print(dataset_summary)
```

```
##                                    vars  n    mean    sd  median trimmed   mad
## Year                                  1 53 1996.00 15.44 1996.00 1996.00 19.27
## Real_Interest_Rate                    2 53    6.22  7.17    6.27    6.36  4.48
## Domestic_Savings                      3 53   14.46  6.00   13.45   14.40  7.32
## Government_Consumption_Expenditure    4 53   15.83  2.43   16.25   15.89  2.79
## Labour_Force                          5 53   73.58  0.46   73.55   73.57  0.52
## Gross_Capital_Formation               6 53   20.75  3.27   20.46   20.71  3.44
##                                        min     max range  skew kurtosis   se
## Year                               1970.00 2022.00 52.00  0.00    -1.27 2.12
## Real_Interest_Rate                  -10.10   21.10 31.19 -0.10    -0.17 0.99
## Domestic_Savings                      4.31   27.15 22.84  0.09    -1.10 0.82
## Government_Consumption_Expenditure   11.74   19.80  8.06 -0.28    -1.31 0.33
## Labour_Force                         72.78   74.41  1.63  0.10    -0.97 0.06
## Gross_Capital_Formation              15.00   29.79 14.79  0.26    -0.41 0.45
```

Descriptive statistics

This section handles mean,median,range, std.deviation, variance

```
# Get a detailed summary of the modified dataset
dataset_summary <- describe(df)

# Print the summary statistics
print(dataset_summary)
```

```
##                                    vars  n    mean    sd  median trimmed   mad
## Year                                  1 53 1996.00 15.44 1996.00 1996.00 19.27
## Real_Interest_Rate                    2 53    6.22  7.17    6.27    6.36  4.48
## Domestic_Savings                      3 53   14.46  6.00   13.45   14.40  7.32
## Government_Consumption_Expenditure    4 53   15.83  2.43   16.25   15.89  2.79
## Labour_Force                          5 53   73.58  0.46   73.55   73.57  0.52
## Gross_Capital_Formation               6 53   20.75  3.27   20.46   20.71  3.44
##                                        min     max range  skew kurtosis   se
## Year                               1970.00 2022.00 52.00  0.00    -1.27 2.12
## Real_Interest_Rate                  -10.10   21.10 31.19 -0.10    -0.17 0.99
## Domestic_Savings                      4.31   27.15 22.84  0.09    -1.10 0.82
## Government_Consumption_Expenditure   11.74   19.80  8.06 -0.28    -1.31 0.33
## Labour_Force                         72.78   74.41  1.63  0.10    -0.97 0.06
## Gross_Capital_Formation              15.00   29.79 14.79  0.26    -0.41 0.45
```

Exploratory Data Analysis

Correlation Analysis

multicollinearity

Multicollinearity refers to the presence of a strong correlation among two or more of the predictor variables in the dataset. The presence of any correlation among predictors is detrimental to model quality for two reasons: It tends to increase the standard error of the coefficients estimates, making them less precise and leading to wider confidence intervals. It becomes difficult to estimate the effect of any one predictor variable on the response variable because multicollinearity makes the coefficients sensitive to small changes in the model or the data, which can lead to unstable coefficient estimates.

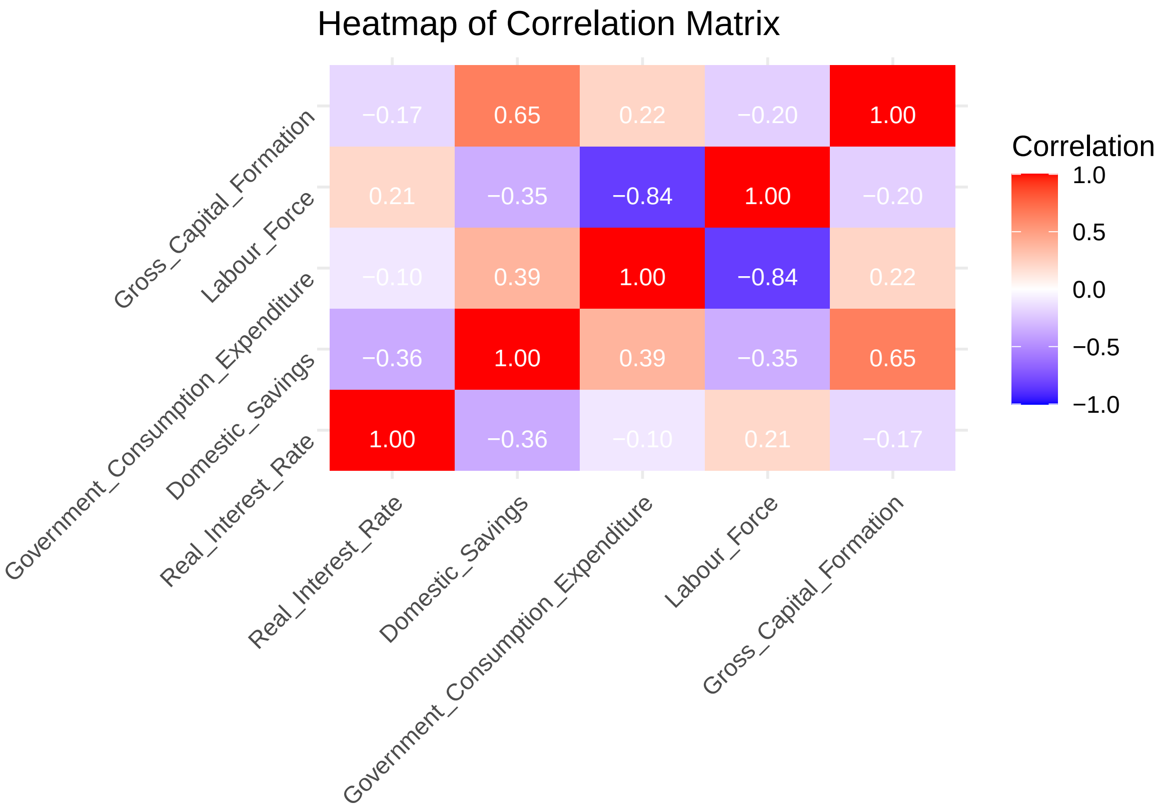
```
# Exclude the first column from the dataset
df_excl_first <- df[, -1]

# Calculate the correlation matrix
cor_matrix <- cor(df_excl_first, use = "complete.obs")  # Handling NA values if any

# Melt the correlation matrix for ggplot2, ensuring variable names
melted_cor_matrix <- melt(cor_matrix)

# Plot the heatmap with correlation coefficients
heatmap_plot <- ggplot(data = melted_cor_matrix, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +  # Add tiles for heatmap
  geom_text(aes(label = sprintf("%.2f", value)), vjust = 1, color = "white", size = 3) +  # Add text an
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1, 1), space
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), axis.text.y = element_text(angle = 45, vjust
  labs(x = "", y = "", title = "Heatmap of Correlation Matrix")

print(heatmap_plot)
```



```
# Display the correlation matrix as a formatted table
kable(cor_matrix, caption = "Correlation Matrix", format = "html", align = 'c')
```

Correlation Matrix

Real_Interest_Rate

Domestic_Savings

Government_Consumption_Expenditure

Labour_Force

Gross_Capital_Formation

Real_Interest_Rate

1.0000000

-0.3645549

-0.1036645

0.2060431

-0.1722458

Domestic_Savings

-0.3645549

1.0000000

0.3876681

-0.3505172

0.6490086

Government_Consumption_Expenditure

-0.1036645

0.3876681

1.0000000

-0.8352250

0.2221956

Labour_Force

0.2060431

-0.3505172

-0.8352250

1.0000000

-0.2025073

Gross_Capital_Formation

-0.1722458

0.6490086

0.2221956

-0.2025073

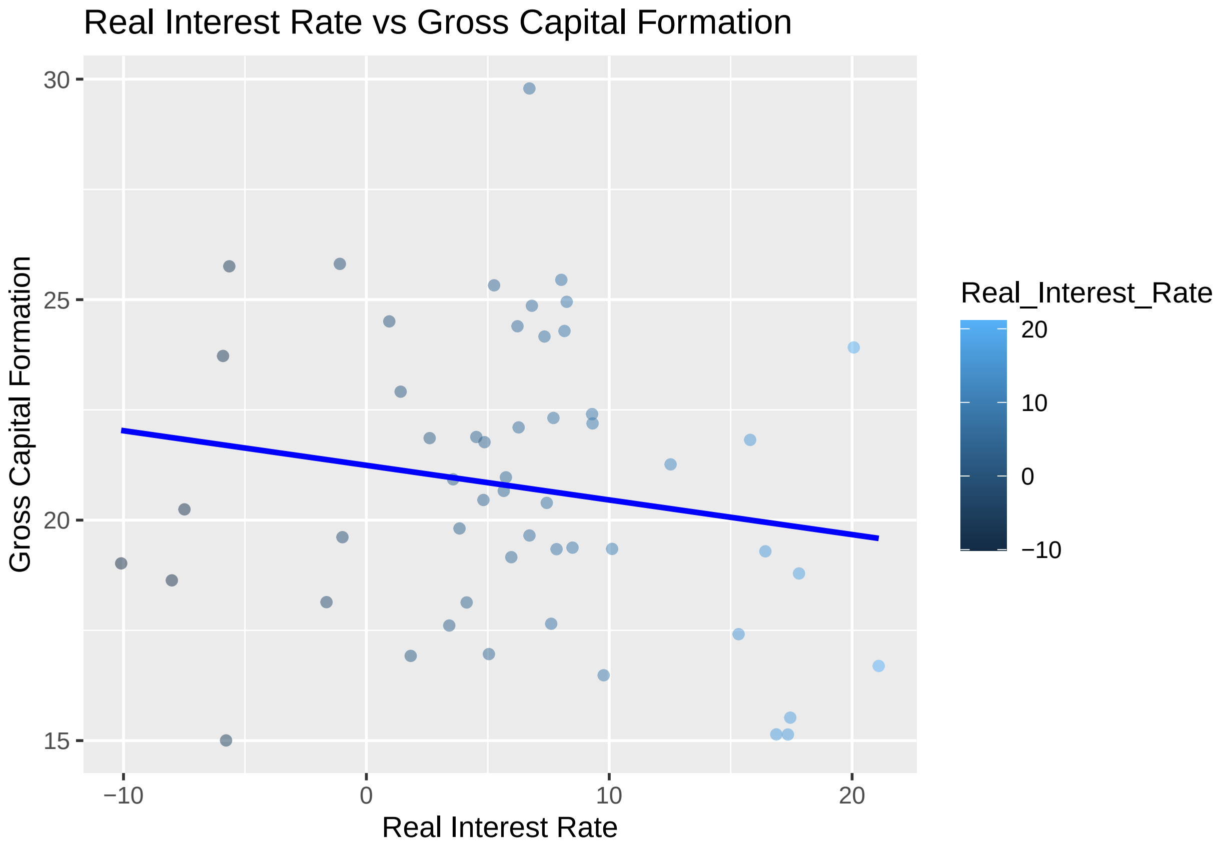
1.0000000

Checking Linearity

A linear relationship implies that the change in the response variable Y, resulting from a one-unit change in the predictor Xj,remains consistent across different values of Xj

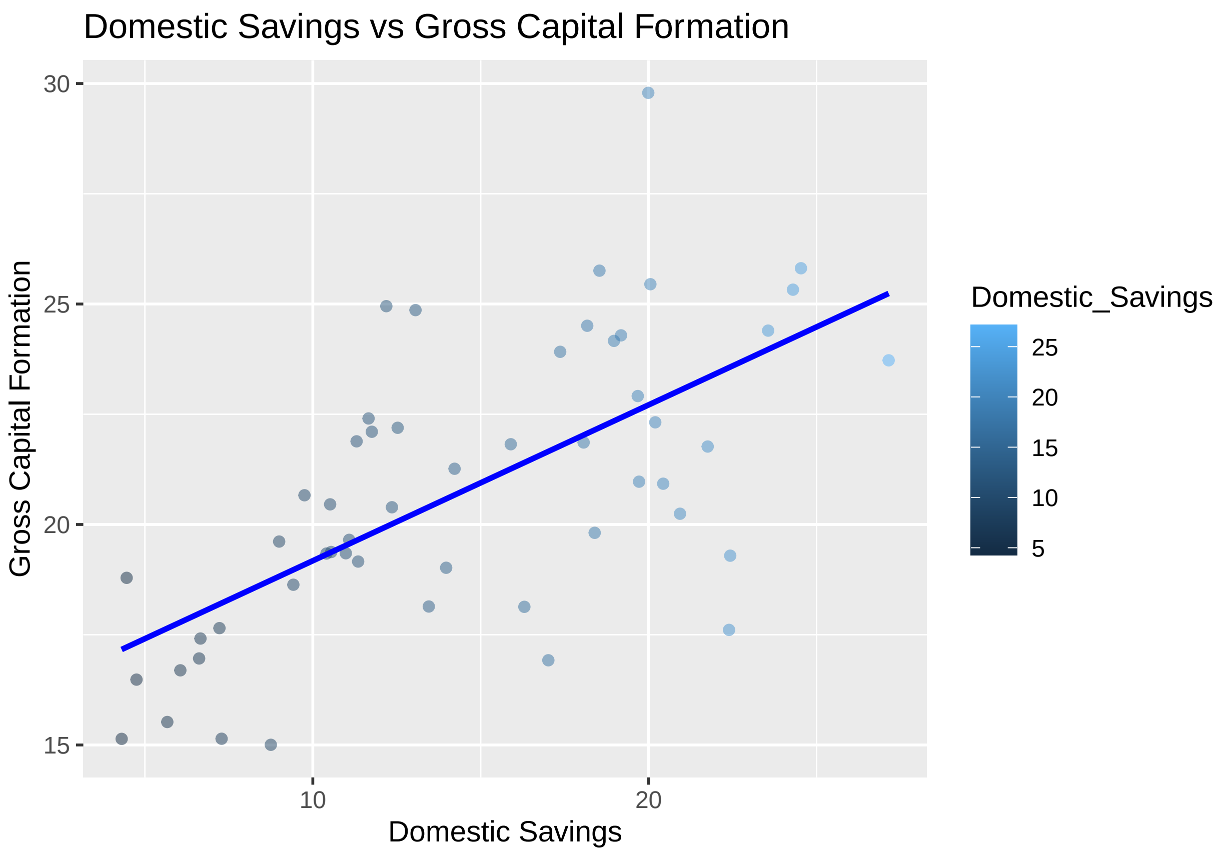
```
# Scatter plot for Real_Interest_Rate vs Gross_Capital_Formation
ggplot(df, aes(x=Real_Interest_Rate, y=Gross_Capital_Formation)) +
  geom_point(aes(color = Real_Interest_Rate), alpha = 0.5) +
  geom_smooth(method="lm", se=FALSE, color="blue") +
  labs(title="Real Interest Rate vs Gross Capital Formation",
       x="Real Interest Rate", y="Gross Capital Formation")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



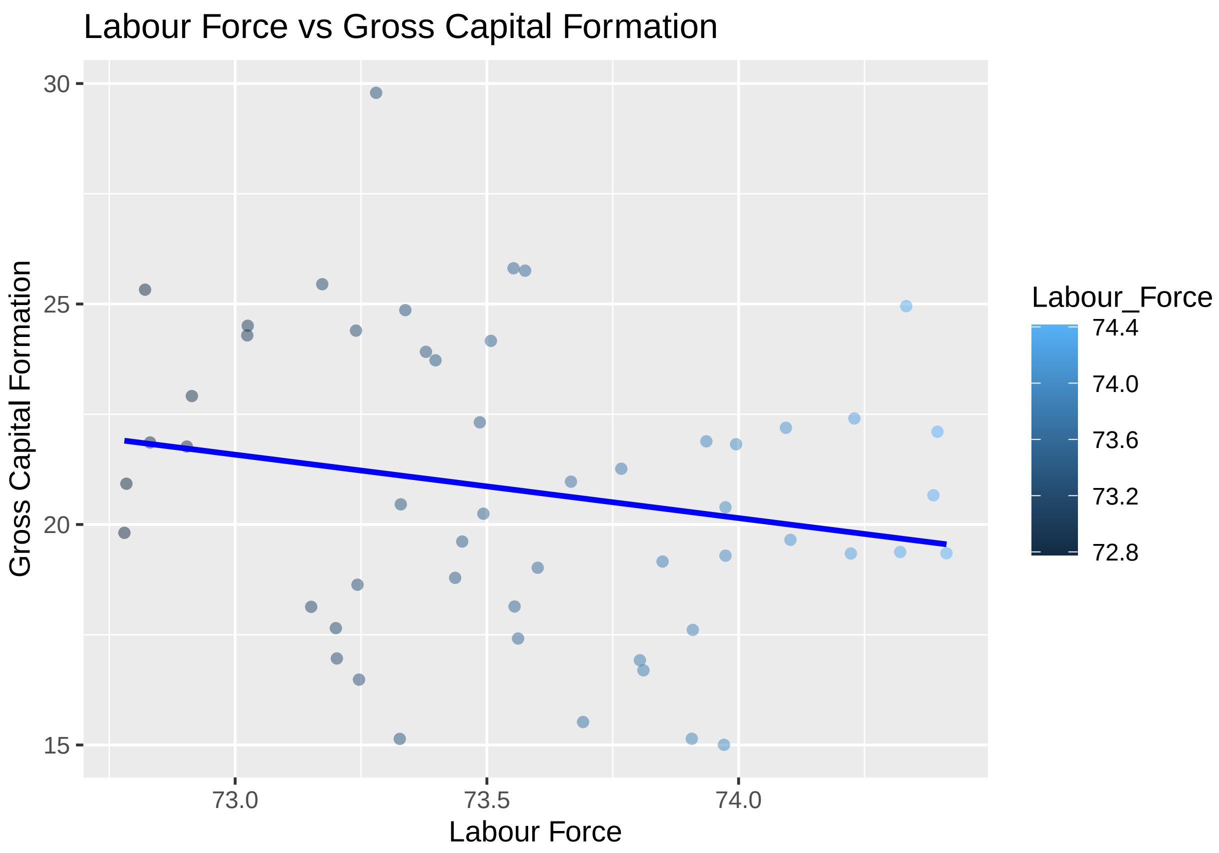
```
# Scatter plot for Domestic_Savings vs Gross_Capital_Formation
ggplot(df, aes(x=Domestic_Savings, y=Gross_Capital_Formation)) +
  geom_point(aes(color = Domestic_Savings), alpha = 0.5) +
  geom_smooth(method="lm", se=FALSE, color="blue") +
  labs(title="Domestic Savings vs Gross Capital Formation",
       x="Domestic Savings", y="Gross Capital Formation")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



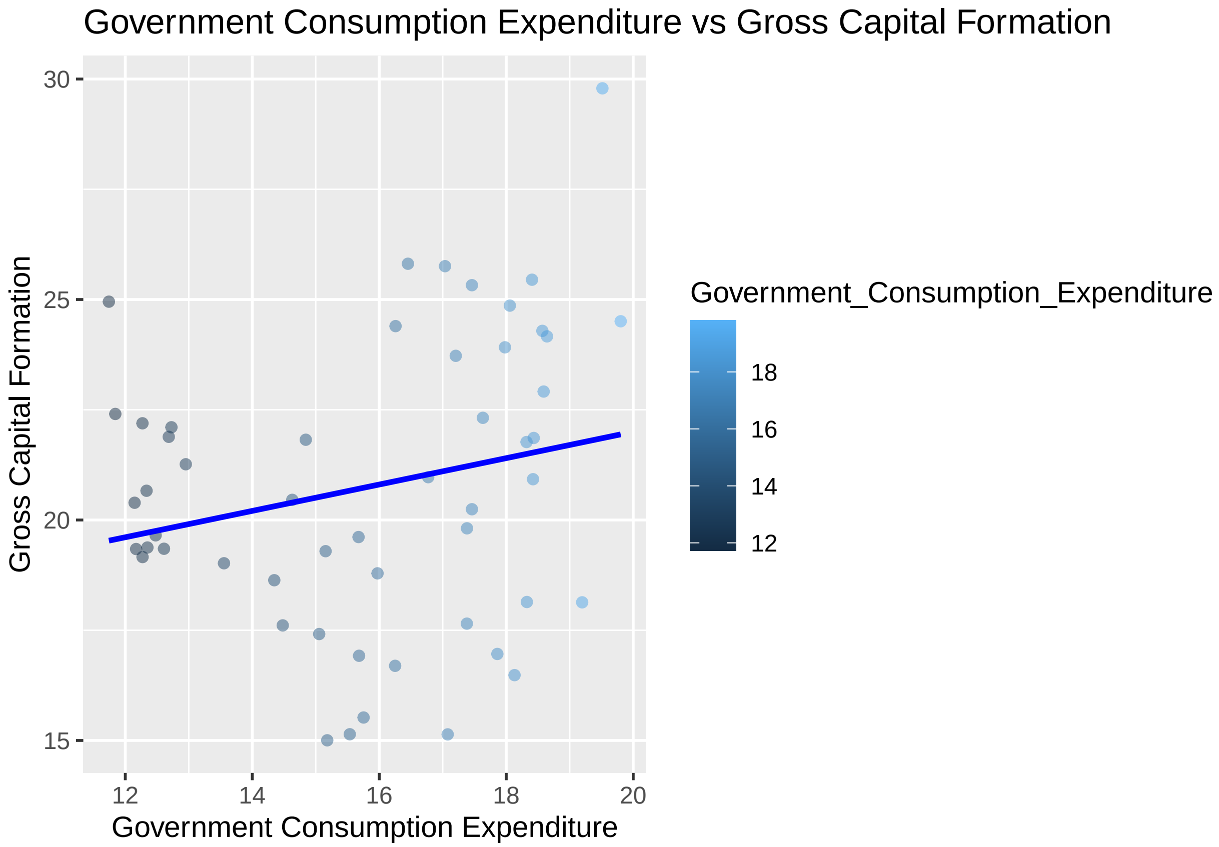
```
# Scatter plot for Labour_Force vs Gross_Capital_Formation
ggplot(df, aes(x=Labour_Force, y=Gross_Capital_Formation)) +
  geom_point(aes(color = Labour_Force), alpha = 0.5) +
  geom_smooth(method="lm", se=FALSE, color="blue") +
  labs(title="Labour Force vs Gross Capital Formation",
       x="Labour Force", y="Gross Capital Formation")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Scatter plot for Government_Consumption_Expenditure vs Gross_Capital_Formation
ggplot(df, aes(x=Government_Consumption_Expenditure, y=Gross_Capital_Formation)) +
  geom_point(aes(color = Government_Consumption_Expenditure), alpha = 0.5) +
  geom_smooth(method="lm", se=FALSE, color="blue") +
  labs(title="Government Consumption Expenditure vs Gross Capital Formation",
       x="Government Consumption Expenditure", y="Gross Capital Formation")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



# Regression Analysis It examines the relationship between two or more variables.

```
# Fit the linear regression model
model <- lm(Gross_Capital_Formation ~ Domestic_Savings + Real_Interest_Rate + Government_Consumption_Exp

# Extract coefficients
coeffs <- coefficients(model)

# Create the regression equation using the actual coefficients
regression_equation <- paste("Gross_Capital_Formation = ", round(coeffs[1], 2),
                             " + ", round(coeffs[2], 2), " * Domestic_Savings",
                             " + ", round(coeffs[3], 2), " * Real_Interest_Rate",
                             " + ", round(coeffs[4], 2), " * Government_Consumption_Expenditure",
                             " + ", round(coeffs[5], 2), " * Labour_Force",
                             " + ", "epsilon")

# Print the equation
cat("The regression model we are considering is as follows:\\\\")
```

```
## The regression model we are considering is as follows:\\
```

```
cat("$$")
```

```
## $$
```

```
cat(regression_equation)
```

```
## Gross_Capital_Formation =  29.99  +  0.38  * Domestic_Savings  +  0.04  * Real_Interest_Rate  +  -0.0
```

```
cat("$$")
```

```
## $$
```

```
# Display the regression results in a table
tidy_model <- broom::tidy(model)
knitr::kable(tidy_model, format = "html", caption = "Regression Results: Gross Capital Formation")
```

Regression Results: Gross Capital Formation

term

estimate

std.error

statistic

p.value

(Intercept)

29.9870091

109.9300368

0.2727827

0.7861904

Domestic_Savings

0.3768240

0.0690249

5.4592495

0.0000017

Real_Interest_Rate

0.0360822

0.0547823

0.6586467

0.5132698

Government_Consumption_Expenditure

-0.0801967

0.2775505

-0.2889446

0.7738680

Labour_Force

-0.1853338

1.4462244

-0.1281501

0.8985656

Homoscedasticity

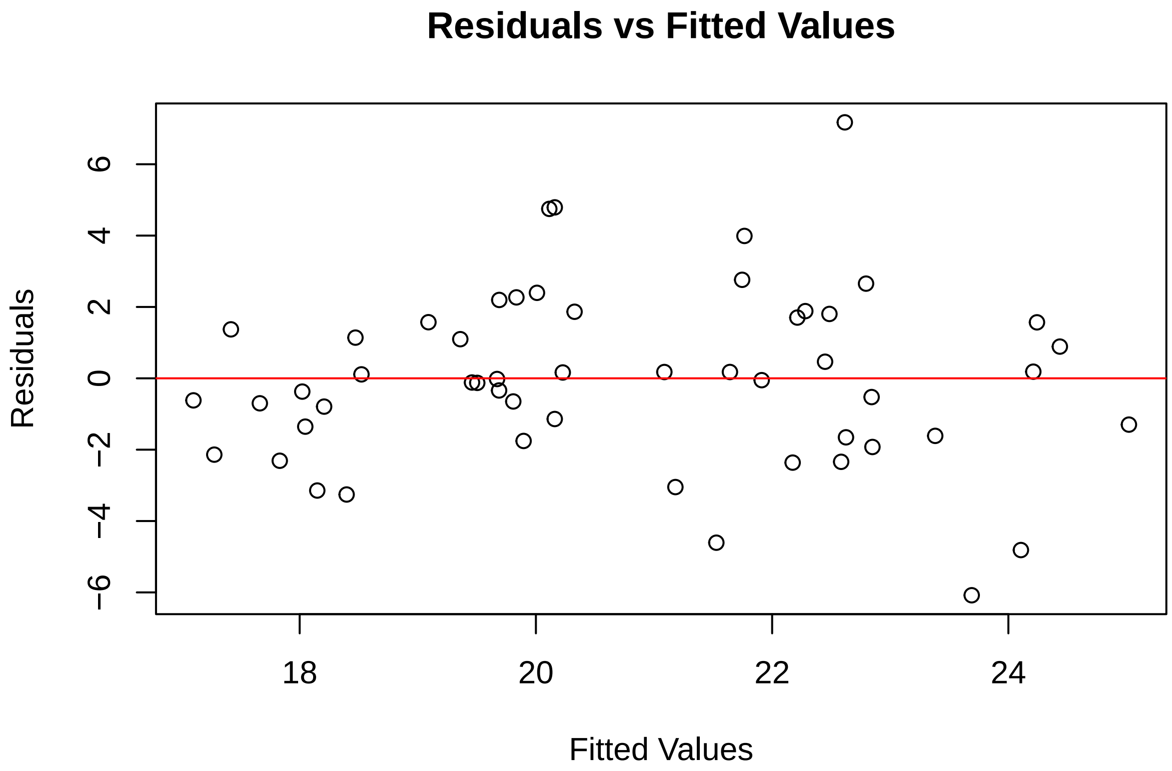
Homoscedasticity implies a constant variance of the residuals across different levels of the predictor variable(s), while heteroscedasticity indicates varying variances. It's essential to detect and address heteroscedasticity as it can affect the validity of statistical inference and prediction.

```r
# Graphical Methods

# Calculate fitted values and residuals
fitted_values <- fitted(model)
residuals <- resid(model)

# Plot residuals vs fitted values
plot(fitted_values, residuals, main="Residuals vs Fitted Values",
     xlab="Fitted Values", ylab="Residuals")

# Add a horizontal line at zero
abline(h = 0, col = "red")
```



```
## Statistical Tests for Homoscedasticity
#  Breusch-Pagan Test
## Perform the Breusch-Pagan test
bp_test <- bptest(model)

# Create a data frame for better presentation
bp_test_df <- data.frame(
  Test_Statistic = bp_test$statistic,
  P_Value = bp_test$p.value,
  Method = bp_test$method
)

# Use knitr::kable to create a nicely formatted table
kable(bp_test_df, caption = "Breusch-Pagan Test Results", align = 'c')
```

Table 1: Breusch-Pagan Test Results

	Test_Statistic	P_Value	Method
BP	9.994425	0.0405217	studentized Breusch-Pagan test

Normality Test

Here we attempt to confirm our assumption of normality amongst the residuals. If the residuals are non-normally distributed, confidence intervals can become too wide or too narrow, which leads to difficulty in estimating coefficients based on the minimisation of ordinary least squares.

```
# Shapiro-Wilk Test-check if the residuals from our linear regression model are normally distributed.

# Perform the Shapiro-Wilk test on residuals
residuals <- residuals(model)
shapiro_test <- shapiro.test(residuals)

# Create a data frame to store the test results
shapiro_results <- data.frame(
  Statistic = shapiro_test$statistic,
  P_Value = shapiro_test$p.value
)

# Use kable from knitr to create a table of the results
kable(shapiro_results, caption = "Shapiro-Wilk Test Results", format = "html", col.names = c("Test Stati
```

Shapiro-Wilk Test Results

Test Statistic

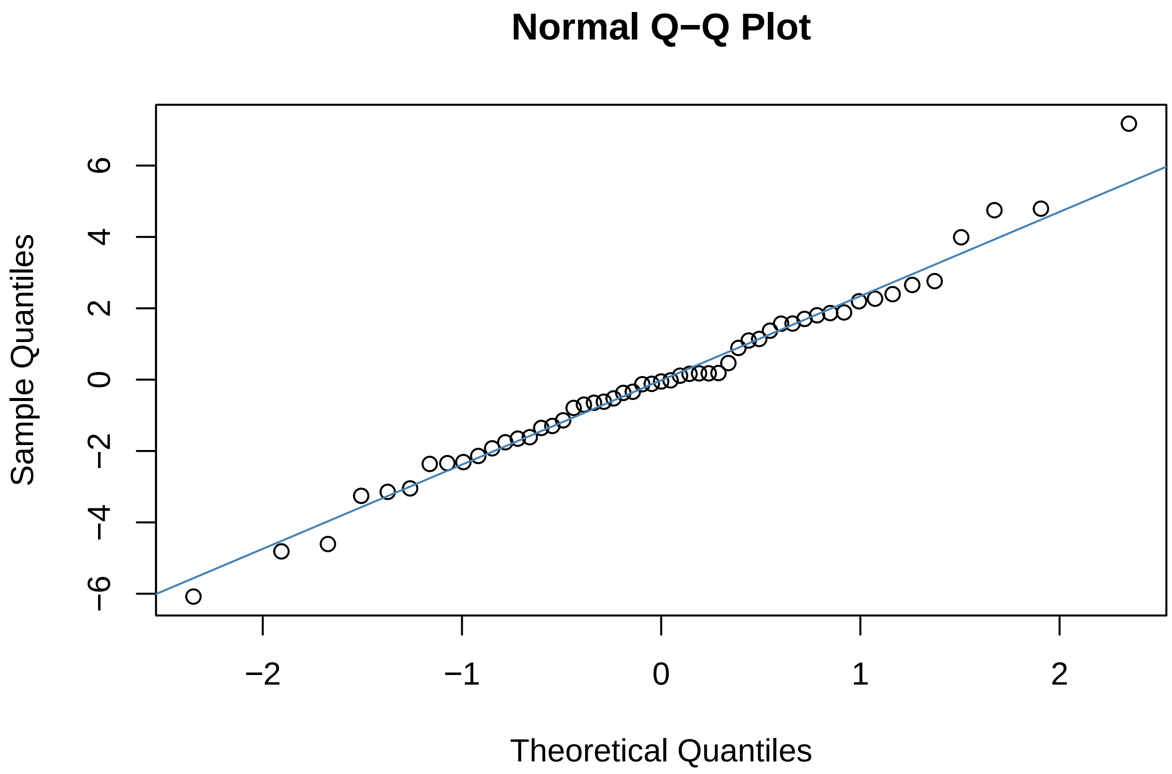
P-value

W

0.9854204

0.7608595

```
## Visual Inspection: QQ Plot
# Create a QQ plot of residuals
qqnorm(residuals)
qqline(residuals, col = "steelblue")
```



Serial Correlation

Serial correlation(autocorrelation) occurs when residuals are not independent of each other.

```
## Test for Serial Correlation

# The Durbin-Watson test helps us detect the presence of serial correlation in the residuals of our lin

# Perform the Durbin-Watson test using lmtest
dw_test <- dwtest(model)

# Create a data frame to store the test results
dw_results <- data.frame(
  Test_Statistic = dw_test$statistic,
  P_Value = dw_test$p.value
)

# Use kable from knitr to create a table of the results
kable(dw_results, caption = "Durbin-Watson Test Results", format = "html", col.names = c("Durbin-Watson
```

Durbin-Watson Test Results

Durbin-Watson Statistic

P-Value

DW

1.378657

0.0026698