Msc Statistics colleges in Pune | Msc ...

Internal component-3-Coding

Vivek Sharma – 23060641081

Shruti Shukla – 23060641083

Course: Linear Models

Under the Guidance of **Dr. Priya Deshpande**

Codes performed for:

1. Exploratory Data Analysis
2. Model Development
3. Comparison of Actual vs Fitted

**Code:**

#CODE FOR EDA

# Load required libraries

library(dplyr)

library(ggplot2)

library(reshape2)

library(rpart)

library(corrplot)

library(caret)

#Accessing Data

data = read.csv("C:\\Users\\ABHISHEK\\OneDrive\\Desktop\\LM Project\\Log Transform.csv")

# View the structure of your dataset

str(data)

# Summary statistics

summary(data)

# Correlation matrix

correlation\_matrix = cor(data[,c("GDP", "Import...", "Export...",

"Population", "Unemployment.rate...", "Minimum.Wage...month.",

"GDP.per.capita..in...", "Real.GDP.Growth..in..",

"Gov.Expenditure...of.GDP", "Tax.total.rate...")])

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

# Histogram for each variable

par(mfrow=c(3, 3))

for(i in 2:10) {

hist(data[,i], main = colnames(data)[i], xlab = colnames(data)[i], col = "skyblue", border = "darkblue")

}

# Box plot for each variable

par(mfrow=c(3, 3))

for(i in 2:10) {

boxplot(data[,i], main = colnames(data)[i], col = "skyblue", border = "darkblue")

}

#CODE FOR MODEL DEVELOPMENT

# MULTIPLE LINEAR REGRESSION

#storing GDP\_data in a dataframe called 'data'

dataframe = as.data.frame(data)

colnames(data)

# Multiple Linear Regression

model = lm(dataframe$GDP ~ dataframe$GDP.per.capita..in...

+ dataframe$Real.GDP.Growth..in..

+ dataframe$Gov.Expenditure...of.GDP + dataframe$Import...

+ dataframe$Export...+ dataframe$Tax.total.rate...

+ dataframe$Unemployment.rate...+

dataframe$Minimum.Wage...month. + dataframe$Population)

# Summary of the regression model

summary(model)

# Visualization of Residuals

residuals = resid(model)

fitted = fitted(model)

# Plotting Residuals vs Fitted Values

ggplot(data, aes(x = fitted, y = residuals)) +

geom\_point() +

geom\_hline(yintercept = 0, linetype = "dashed", color = "red") +

labs(title = "Residuals vs Fitted Values", x = "Fitted Values", y = "Residuals")

# Plotting Normal Q-Q Plot

qqnorm(residuals)

qqline(residuals)

# Plotting Residuals vs Predictors

par(mfrow = c(3, 3)) # Arrange plots in a 3x3 grid

for(i in 2:length(model$coefficients)) {

plot(data[, i], residuals, xlab = names(data)[i], ylab = "Residuals")

}

# COMPAIRISON OF ACTUAL VS FITTED

# Split the data into training and testing sets

set.seed(123) # for reproducibility

train\_index = createDataPartition(data$GDP, p = 0.8, list = FALSE)

train\_data = data[train\_index, ]

colnames(train\_data)

test\_data = data[-train\_index, ]

# Train the linear regression model

lm\_model = lm(GDP ~ ., data = train\_data)

# Train the multiple linear regression model

lm\_model = lm(GDP ~ GDP.per.capita..in... + Real.GDP.Growth..in.. + Gov.Expenditure...of.GDP

+ Import... + Export... + Tax.total.rate... + Unemployment.rate...

+ Minimum.Wage...month. + Population, data = train\_data)

# Predict GDP for training data

train\_predictions = predict(lm\_model, newdata = train\_data)

# Predict GDP for testing data

test\_predictions = predict(lm\_model, newdata = test\_data)

# Compare actual GDP with predicted GDP for training data

train\_comparison = data.frame(Actual\_GDP = train\_data$GDP, Predicted\_GDP = train\_predictions)

# Compare actual GDP with predicted GDP for testing data

test\_comparison = data.frame(Actual\_GDP = test\_data$GDP, Predicted\_GDP = test\_predictions)

# Output

cat("\nTesting Data Comparison:\n")

print(head(test\_comparison))

# Mean Absolute Error (MAE) for test data

test\_MAE = mean(abs(test\_comparison$Actual\_GDP - test\_comparison$Predicted\_GDP))

# Mean Squared Error (MSE) for test data

test\_MSE = mean((test\_comparison$Actual\_GDP - test\_comparison$Predicted\_GDP)^2)

# Root Mean Squared Error (RMSE) for test data

test\_RMSE = sqrt(test\_MSE)

# R-squared for test data

test\_Rsquared = summary(lm\_model)$adj.r.squared

cat("\nTest Data:\n")

cat("Mean Absolute Error (MAE):", test\_MAE, "\n")

cat("Mean Squared Error (MSE):", test\_MSE, "\n")

cat("Root Mean Squared Error (RMSE):", test\_RMSE, "\n")

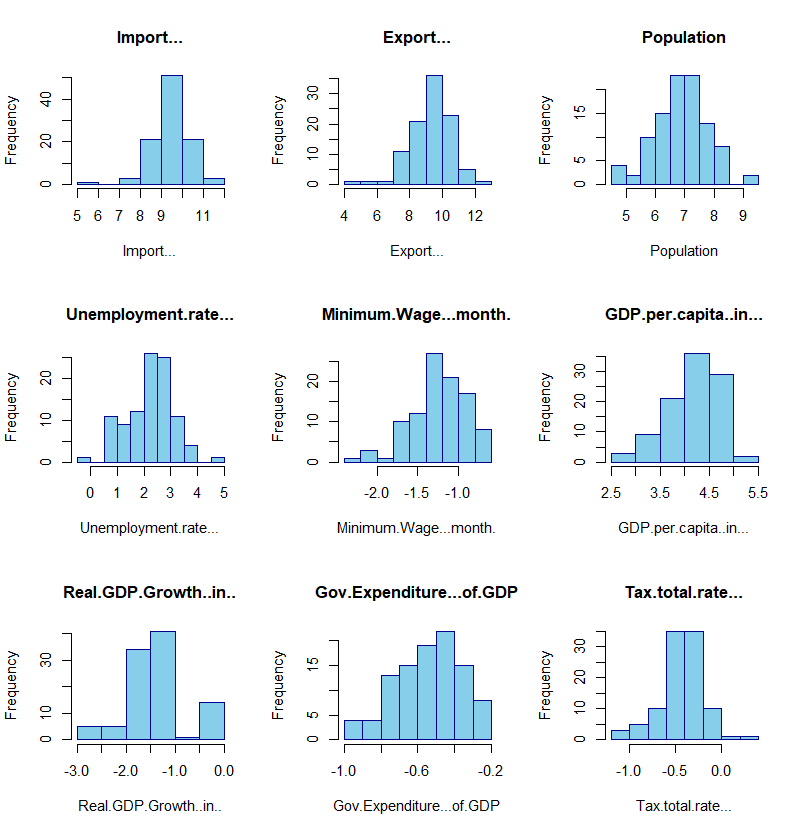
cat("R-squared:", test\_Rsquared, "\n")

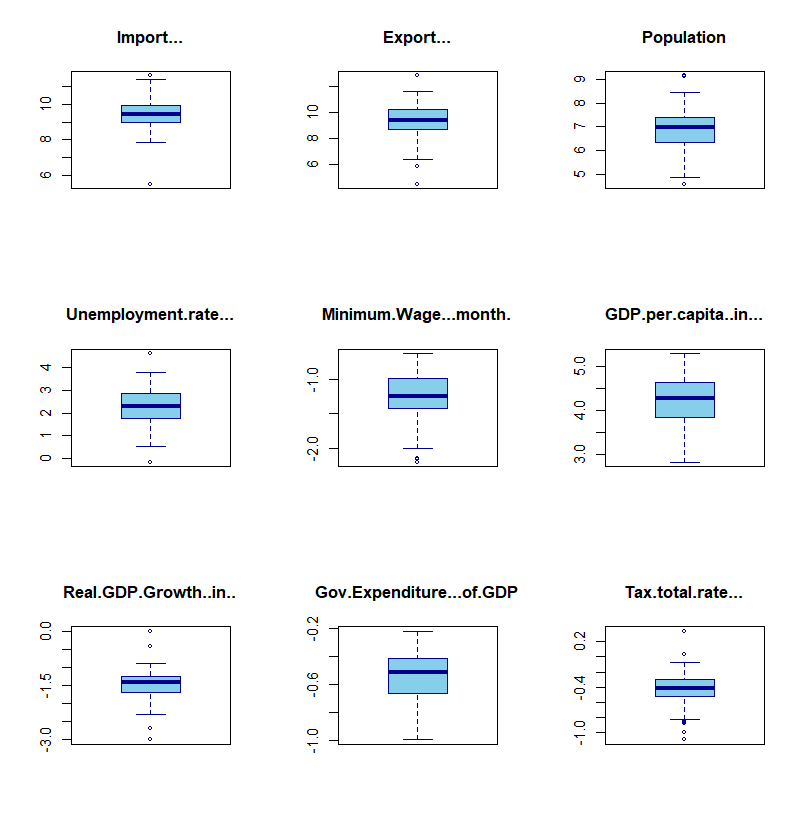
Output:

|  |
| --- |
| > #Accessing Data  > data = read.csv("C:\\Users\\ABHISHEK\\OneDrive\\Desktop\\LM Project\\Log Transform.csv")  >  > # View the structure of your dataset  > str(data)  'data.frame': 100 obs. of 10 variables:  $ GDP : num 10.3 10.2 11.2 9.5 11 ...  $ Import... : num 9.71 10.78 10.03 9.3 9.55 ...  $ Export... : num 9.02 10.47 10.12 8.4 9.99 ...  $ Population : num 7.58 6.46 7.63 4.89 7.5 ...  $ Unemployment.rate... : num 1.83 1.63 2.17 3.16 1.59 ...  $ Minimum.Wage...month. : num -0.954 -0.909 -0.932 -2 -1.162 ...  $ GDP.per.capita..in... : num 2.83 4.29 4.14 4.83 3.85 ...  $ Real.GDP.Growth..in.. : num 0 -1.44 -1.42 -1.68 -1.89 ...  $ Gov.Expenditure...of.GDP: num -0.53 -0.503 -0.506 -0.444 -0.661 ...  $ Tax.total.rate... : num -0.146 -0.437 -0.18 -1 -0.309 ...  >  > # Summary statistics  > summary(data)  GDP Import... Export... Population Unemployment.rate...  Min. : 8.775 Min. : 5.502 Min. : 4.434 Min. :4.580 Min. :-0.1524  1st Qu.:10.051 1st Qu.: 8.997 1st Qu.: 8.669 1st Qu.:6.362 1st Qu.: 1.7901  Median :10.603 Median : 9.424 Median : 9.344 Median :6.953 Median : 2.3091  Mean :10.665 Mean : 9.460 Mean : 9.311 Mean :6.872 Mean : 2.2622  3rd Qu.:11.375 3rd Qu.: 9.943 3rd Qu.:10.191 3rd Qu.:7.412 3rd Qu.: 2.8846  Max. :13.299 Max. :11.593 Max. :12.824 Max. :9.145 Max. : 4.6290  Minimum.Wage...month. GDP.per.capita..in... Real.GDP.Growth..in.. Gov.Expenditure...of.GDP  Min. :-2.2007 Min. :2.831 Min. :-3.000 Min. :-0.9929  1st Qu.:-1.4186 1st Qu.:3.844 1st Qu.:-1.683 1st Qu.:-0.6639  Median :-1.2522 Median :4.272 Median :-1.420 Median :-0.5169  Mean :-1.2508 Mean :4.173 Mean :-1.369 Mean :-0.5375  3rd Qu.:-1.0004 3rd Qu.:4.634 3rd Qu.:-1.252 3rd Qu.:-0.4137  Max. :-0.6306 Max. :5.296 Max. : 0.000 Max. :-0.2192  Tax.total.rate...  Min. :-1.0969  1st Qu.:-0.5161  Median :-0.4214  Mean :-0.4239  3rd Qu.:-0.3028  Max. : 0.3416  >  > # Correlation matrix  > correlation\_matrix = cor(data[,c("GDP", "Import...", "Export...",  + "Population", "Unemployment.rate...", "Minimum.Wage...month.",  + "GDP.per.capita..in...", "Real.GDP.Growth..in..",  + "Gov.Expenditure...of.GDP", "Tax.total.rate...")])  > corrplot(correlation\_matrix, method = "color")  >  > # Histogram for each variable  > par(mfrow=c(3, 3))  > for(i in 2:10) {  + hist(data[,i], main = colnames(data)[i], xlab = colnames(data)[i], col = "skyblue", border = "darkblue")  + }  >  > # Box plot for each variable  > par(mfrow=c(3, 3))  > for(i in 2:10) {  + boxplot(data[,i], main = colnames(data)[i], col = "skyblue", border = "darkblue")  + } |
| > |
|  |

**Correlation Matrix Heat Map**



**Individual Histogram For Each Independent Variables**

**Individual Box plot for each independent variables**

> #CODE FOR MODEL DEVELOPMENT

> # MULTIPLE LINEAR REGRESSION

>

> #storing GDP\_data in a dataframe called 'data'

> dataframe = as.data.frame(data)

> colnames(data)

[1] "GDP" "Import..."

[3] "Export..." "Population"

[5] "Unemployment.rate..." "Minimum.Wage...month."

[7] "GDP.per.capita..in..." "Real.GDP.Growth..in.."

[9] "Gov.Expenditure...of.GDP" "Tax.total.rate..."

> # Multiple Linear Regression

> model = lm(dataframe$GDP ~ dataframe$GDP.per.capita..in...

+ + dataframe$Real.GDP.Growth..in..

+ + dataframe$Gov.Expenditure...of.GDP + dataframe$Import...

+ + dataframe$Export...+ dataframe$Tax.total.rate...

+ + dataframe$Unemployment.rate...+

+ dataframe$Minimum.Wage...month. + dataframe$Population)

>

> # Summary of the regression model

> summary(model)

Call:

lm(formula = dataframe$GDP ~ dataframe$GDP.per.capita..in... +

dataframe$Real.GDP.Growth..in.. + dataframe$Gov.Expenditure...of.GDP +

dataframe$Import... + dataframe$Export... + dataframe$Tax.total.rate... +

dataframe$Unemployment.rate... + dataframe$Minimum.Wage...month. +

dataframe$Population)

Residuals:

Min 1Q Median 3Q Max

-0.59664 -0.12045 -0.02491 0.10534 0.81025

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -0.15942 0.34412 -0.463 0.6443

dataframe$GDP.per.capita..in... 0.98310 0.05471 17.970 <2e-16 \*\*\*

dataframe$Real.GDP.Growth..in.. -0.03779 0.03273 -1.155 0.2513

dataframe$Gov.Expenditure...of.GDP 0.32952 0.13372 2.464 0.0156 \*

dataframe$Import... 0.03162 0.03859 0.819 0.4147

dataframe$Export... 0.01292 0.03175 0.407 0.6852

dataframe$Tax.total.rate... 0.05379 0.10564 0.509 0.6119

dataframe$Unemployment.rate... 0.01446 0.02571 0.563 0.5752

dataframe$Minimum.Wage...month. -0.01624 0.06388 -0.254 0.7999

dataframe$Population 0.93100 0.03710 25.094 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2051 on 90 degrees of freedom

Multiple R-squared: 0.9584, Adjusted R-squared: 0.9542

F-statistic: 230.1 on 9 and 90 DF, p-value: < 2.2e-16

>

> # Visualization of Residuals

> residuals = resid(model)

> fitted = fitted(model)

>

> # Plotting Residuals vs Fitted Values

> ggplot(data, aes(x = fitted, y = residuals)) +

+ geom\_point() +

+ geom\_hline(yintercept = 0, linetype = "dashed", color = "red") +

+ labs(title = "Residuals vs Fitted Values", x = "Fitted Values", y = "Residuals")

>

>

> # Plotting Normal Q-Q Plot

> qqnorm(residuals)

> qqline(residuals)

>

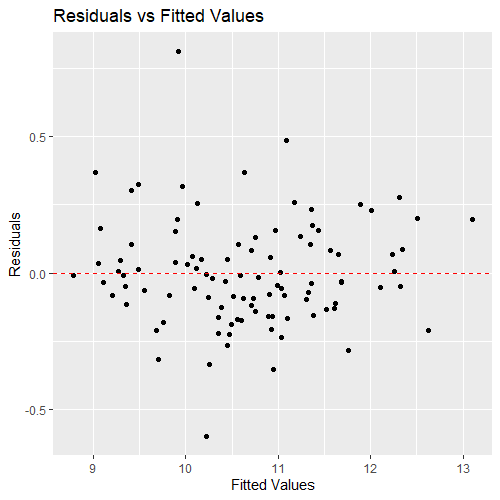
> # Plotting Residuals vs Predictors

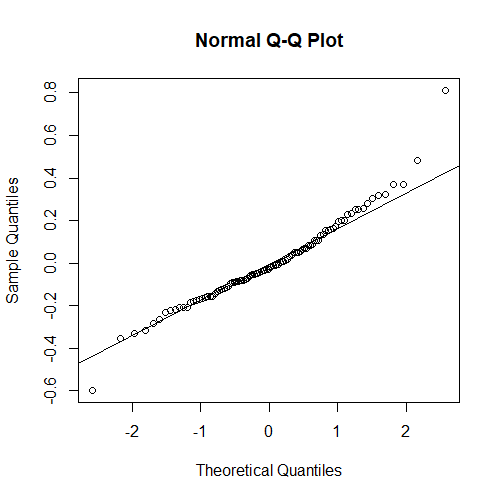
> par(mfrow = c(3, 3)) # Arrange plots in a 3x3 grid

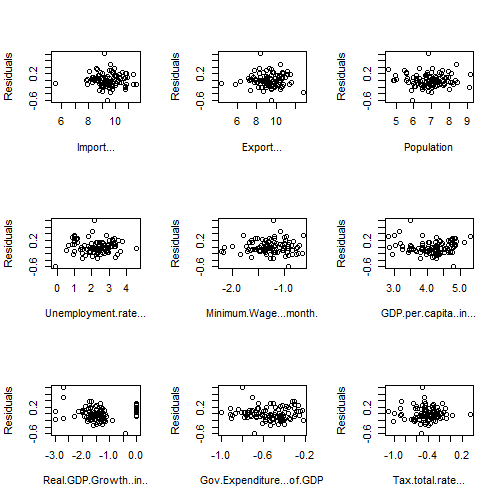
> for(i in 2:length(model$coefficients)) {

+ plot(data[, i], residuals, xlab = names(data)[i], ylab = "Residuals")

+ }







> # COMPAIRISON OF ACTUAL VS FITTED

> # Split the data into training and testing sets

> set.seed(123) # for reproducibility

> train\_index = createDataPartition(data$GDP, p = 0.8, list = FALSE)

> train\_data = data[train\_index, ]

> colnames(train\_data)

[1] "GDP" "Import..."

[3] "Export..." "Population"

[5] "Unemployment.rate..." "Minimum.Wage...month."

[7] "GDP.per.capita..in..." "Real.GDP.Growth..in.."

[9] "Gov.Expenditure...of.GDP" "Tax.total.rate..."

> test\_data = data[-train\_index, ]

>

> # Train the linear regression model

> lm\_model = lm(GDP ~ ., data = train\_data)

>

> # Train the multiple linear regression model

> lm\_model = lm(GDP ~ GDP.per.capita..in... + Real.GDP.Growth..in.. + Gov.Expenditure...of.GDP

+ + Import... + Export... + Tax.total.rate... + Unemployment.rate...

+ + Minimum.Wage...month. + Population, data = train\_data)

>

> # Predict GDP for training data

> train\_predictions = predict(lm\_model, newdata = train\_data)

>

> # Predict GDP for testing data

> test\_predictions = predict(lm\_model, newdata = test\_data)

>

> # Compare actual GDP with predicted GDP for training data

> train\_comparison = data.frame(Actual\_GDP = train\_data$GDP, Predicted\_GDP = train\_predictions)

>

> # Compare actual GDP with predicted GDP for testing data

> test\_comparison = data.frame(Actual\_GDP = test\_data$GDP, Predicted\_GDP = test\_predictions)

>

> # Output

>

> cat("\nTesting Data Comparison:\n")

Testing Data Comparison:

> print(head(test\_comparison))

Actual\_GDP Predicted\_GDP

1 10.281064 9.849134

2 10.184069 10.445883

6 9.237483 9.118459

13 10.586295 10.615463

18 9.274069 9.294846

30 9.297070 9.342214

>

> # Mean Absolute Error (MAE) for test data

> test\_MAE = mean(abs(test\_comparison$Actual\_GDP - test\_comparison$Predicted\_GDP))

>

> # Mean Squared Error (MSE) for test data

> test\_MSE = mean((test\_comparison$Actual\_GDP - test\_comparison$Predicted\_GDP)^2)

>

> # Root Mean Squared Error (RMSE) for test data

> test\_RMSE = sqrt(test\_MSE)

>

> # R-squared for test data

> test\_Rsquared = summary(lm\_model)$adj.r.squared

>

> cat("\nTest Data:\n")

Test Data:

> cat("Mean Absolute Error (MAE):", test\_MAE, "\n")

Mean Absolute Error (MAE): 0.1353863

> cat("Mean Squared Error (MSE):", test\_MSE, "\n")

Mean Squared Error (MSE): 0.0321294

> cat("Root Mean Squared Error (RMSE):", test\_RMSE, "\n")

Root Mean Squared Error (RMSE): 0.1792467

> cat("R-squared:", test\_Rsquared, "\n")

R-squared: 0.9501902