Economic modeling Project 2 Group Xavier, Titouan, Xavier

INTRODUCTION In this project we took CAC 40 as the main country stock index because it has a huge dataset and it is include the top 40 major stocks of europe.Representing major companies, the CAC 40 is a crucial benchmark for the French stock market. Because stock markets frequently reflect and react to broader economic trends, it is imperative that analysts, investors, and policymakers understand these relationships in order to make well-informed decisions.

```r
# Define indicators for GDP, interest rate, inflation, fiscal and monetary policies, consumer confidenc
# Load required libraries
library(readxl)
library(plyr)
library(dplyr)
```

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

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

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

```r
library(tidyr)
library(ggplot2)
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```r
library(nlme)
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```r
library(rcompanion)
```

```
## Warning: package 'rcompanion' was built under R version 4.3.2
```

```r
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.3.2

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```r
library(e1071)
library (moments)
```

```
##
## Attaching package: 'moments'

## The following objects are masked from 'package:e1071':
##
##     kurtosis, moment, skewness
```

```r
library (ADGofTest)
library (faraway)
```

```
##
## Attaching package: 'faraway'

## The following object is masked from 'package:plyr':
##
##     ozone
```

```r
library(dplyr)
library(tidyr)
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following objects are masked from 'package:faraway':
##
##     logit, vif

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(xts)
```

```
##
## ######################### Warning from 'xts' package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## ###############################################################################

##
## Attaching package: 'xts'

## The following objects are masked from 'package:data.table':
##
##     first, last

## The following objects are masked from 'package:dplyr':
##
##     first, last
```

HERE WE WILL LOAD THE DATA

```r
# MUTUTATE function create year column which convert the whole column from character into date.
# ARRANGE reorder data in ascending form
stock_data_Monthly_budget <- read_excel("C:/Users/Siddharth Sharma/OneDrive/Desktop/Semester1/GROUP pro
  mutate(YEAR = as.Date(as.character(YEAR), format = "%Y-%m-%d")) %>%

  arrange(YEAR)
stock_data_GDP<- read.csv("C:/Users/Siddharth Sharma/OneDrive/Desktop/Semester1/GROUP project 2 codes/C
```

```r
   mutate(YEAR = as.Date(as.character(DATE), format = "%Y-%m-%d")) %>%

  arrange(YEAR)
stock_data_inflation <- read.csv("C:/Users/Siddharth Sharma/OneDrive/Desktop/Semester1/GROUP project 2 c
  mutate(YEAR = as.Date(as.character(Date), format = "%B %d,%Y")) %>%

  arrange(YEAR)
stock_data_Exchang_rate <- read.csv("C:/Users/Siddharth Sharma/OneDrive/Desktop/Semester1/GROUP project
  mutate(YEAR = as.Date(as.character(Date), format = "%Y-%m-%d")) %>%

  arrange(YEAR)
stock_data_Consumer_confidence_index<- read.csv("C:/Users/Siddharth Sharma/OneDrive/Desktop/Semester1/GR
  mutate(YEAR = as.Date(as.character(TIME), format = "%Y-%m-%d")) %>%

  arrange(YEAR)
stock_data_Unemployment <- read.csv("C:/Users/Siddharth Sharma/OneDrive/Documents/unemployement.csv")%>%
  mutate(YEAR = as.Date(as.character(Date), format = "%B %d,%Y")) %>%

  arrange(YEAR)

CAC_40<- fread("C:/Users/Siddharth Sharma/OneDrive/Desktop/Semester1/GROUP project 2 codes/^FCHI.csv",,c
  mutate(YEAR = as.Date(as.character(Date), format = "%Y-%m-%d")) %>%
  arrange(YEAR)
```

After loading the data to avoid any complications we remove the unnecacery columns from the datasets

```r
# deleting the unnecesory test
choose=c("Date","Volume","Open","High","Low","Adj.Close")
stock_data_Exchang_rate<-stock_data_Exchang_rate%>%select(-choose)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(choose)
##
##   # Now:
##   data %>% select(all_of(choose))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
choose1=c("LOCATION","INDICATOR","SUBJECT","MEASURE","FREQUENCY","TIME","Flag.Codes")
stock_data_Consumer_confidence_index<-stock_data_Consumer_confidence_index%>%select(-choose1)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(choose1)
##
##   # Now:
```

```
##   data %>% select(all_of(choose1))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
choose3=c("Date","Volume","Open","Adj Close")
CAC_40<-CAC_40%>%select(-choose3)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(choose3)
##
##   # Now:
##   data %>% select(all_of(choose3))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
choose4="DATE"
stock_data_GDP<-stock_data_GDP%>%select(-choose4)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(choose4)
##
##   # Now:
##   data %>% select(all_of(choose4))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
choose5="Date"
stock_data_inflation<-stock_data_inflation%>%select(-choose5)
```

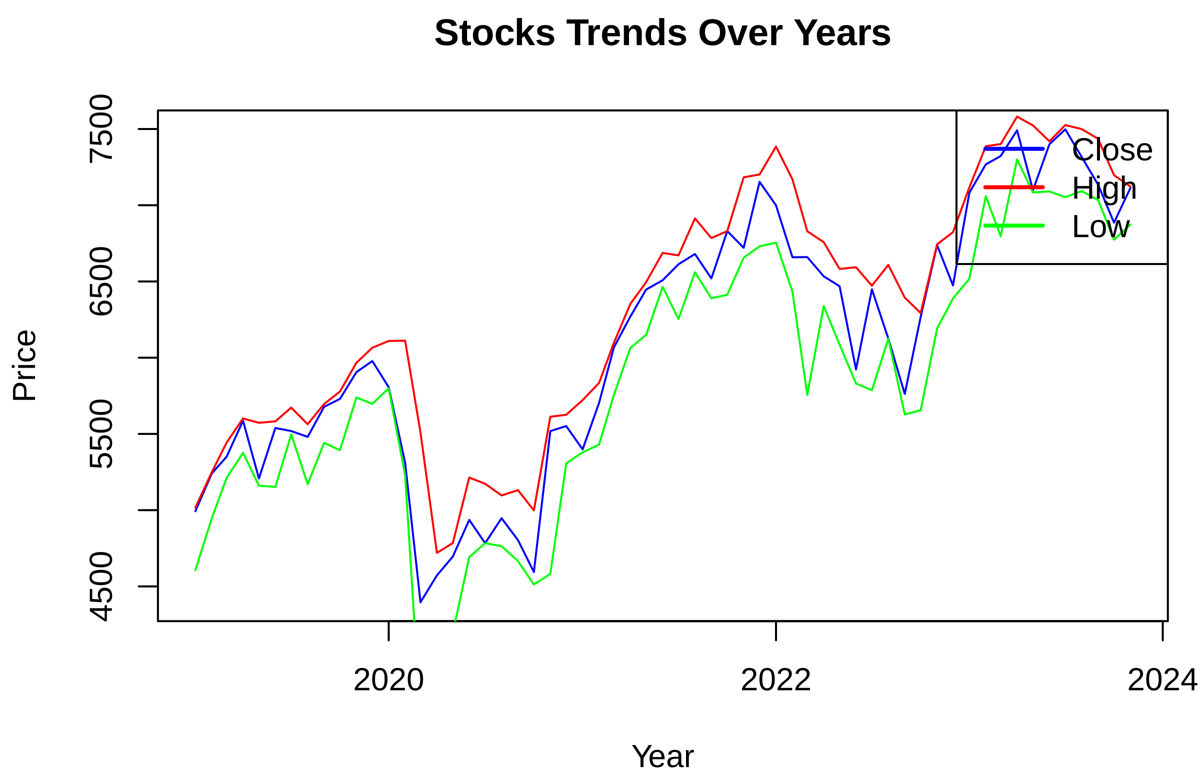
```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(choose5)
##
##   # Now:
##   data %>% select(all_of(choose5))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
stock_data_Unemployment<-stock_data_Unemployment%>%select(-choose5)
```

Here we are plot the closing, low and high to better understand our data

```r
plot(CAC_40$YEAR, CAC_40$Close, type = "l", col = "blue", xlab = "Year", ylab = "Price",
     main = "Stocks Trends Over Years")
# Add lines for NSEI$High and NSEI$Low
lines(CAC_40$YEAR, CAC_40$High, type = "l", col = "red")
lines(CAC_40$YEAR, CAC_40$Low, type = "l", col = "green")

# Add a legend to differentiate the lines
legend("topright", legend = c("Close", "High", "Low"), col = c("blue", "red", "green"), lwd = 2)
```



From the above line graph we understand that during COVID-19 pendemic the shares started to drop a lot but it recover itself with a slow growth

Now we will check for basic analysis for CAC_40

```r
cat("Summary of Closing Value", paste(names(CAC_40$Close), collapse = ", "), ":\n")
```

```
## Summary of Closing Value  :
```

```r
print(summary(CAC_40$Close))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4396    5500    6125    6097    6730    7498
```

```r
print(summary(CAC_40$High))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4720    5607    6394    6304    7016    7581
```

```r
print(summary(CAC_40$Low))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3632    5221    5786    5824    6491    7300
```

```r
print(mean(na.omit(CAC_40$Close, trim=0.1)))
```

```
## [1] 6096.668
```

```r
print(mean(na.omit(CAC_40$Close, trim=0.2)))
```

```
## [1] 6096.668
```

```r
print("variance:")
```

```
## [1] "variance:"
```

```r
variance_var<-(var(na.omit(CAC_40$Close)))
print("Standard deviation:")
```

```
## [1] "Standard deviation:"
```

```r
print(sd(na.omit(CAC_40$Close)))
```

```
## [1] 859.8601
```

```r
print(quantile(na.omit(CAC_40$Close)))
```

```
##       0%      25%      50%      75%     100%
## 4396.120 5499.515 6125.100 6729.855 7497.780
```

```r
print("percentile 5% & 95%")
```

```
## [1] "percentile 5% & 95%"
```

```r
print(quantile(na.omit(CAC_40$Close),prob=c(0.05,0.95),na.rm=TRUE))
```

```
##       5%      95%
## 4685.320 7330.157
```

```r
skewness_var<-skewness(na.omit(CAC_40$Close))
kurtosis_var<-kurtosis(na.omit(CAC_40$Close))
MAD_var<-mad(na.omit(CAC_40$Close))
```

The dataset provide a summary of the closing values. With a mean of 6097, values for the first dataset range from a minimum of 4396 to a maximum of 7498. The second dataset displays a mean of 6304 and a range of 4720 to 7581. The third dataset has a mean of 5824 and a range of 3632 to 7300. 6096.668 is the mean closing value overall. The 5th and 95th percentiles are 4685.320 and 7330.157, respectively, and the variance is 859.8601. The distribution and central tendency of the closing values in the datasets are revealed by these statistics.

'Merging the Datasets for having a good regression analysis with the common column as year

```r
merged_df <- merge(CAC_40, stock_data_Consumer_confidence_index, by="YEAR")
merged_df<-merge(merged_df,stock_data_GDP)
merged_df<-merge(merged_df,stock_data_Exchang_rate)
merged_df<-merge(merged_df,stock_data_inflation)
merged_df<-merge(merged_df,stock_data_Monthly_budget)
merged_df<-merge(merged_df,stock_data_Unemployment)
```

```r
# Creating a function to convert percent (string) to numeric
percentStrToNumeric <- function(percent_str) {
  as.numeric(sub("%", "", percent_str)) / 100
}
```

```r
#converting char into numeric percentage
merged_df$Value <- sapply(merged_df$Value, percentStrToNumeric)
merged_df$Value.y <- sapply(merged_df$Value.y, percentStrToNumeric)
```

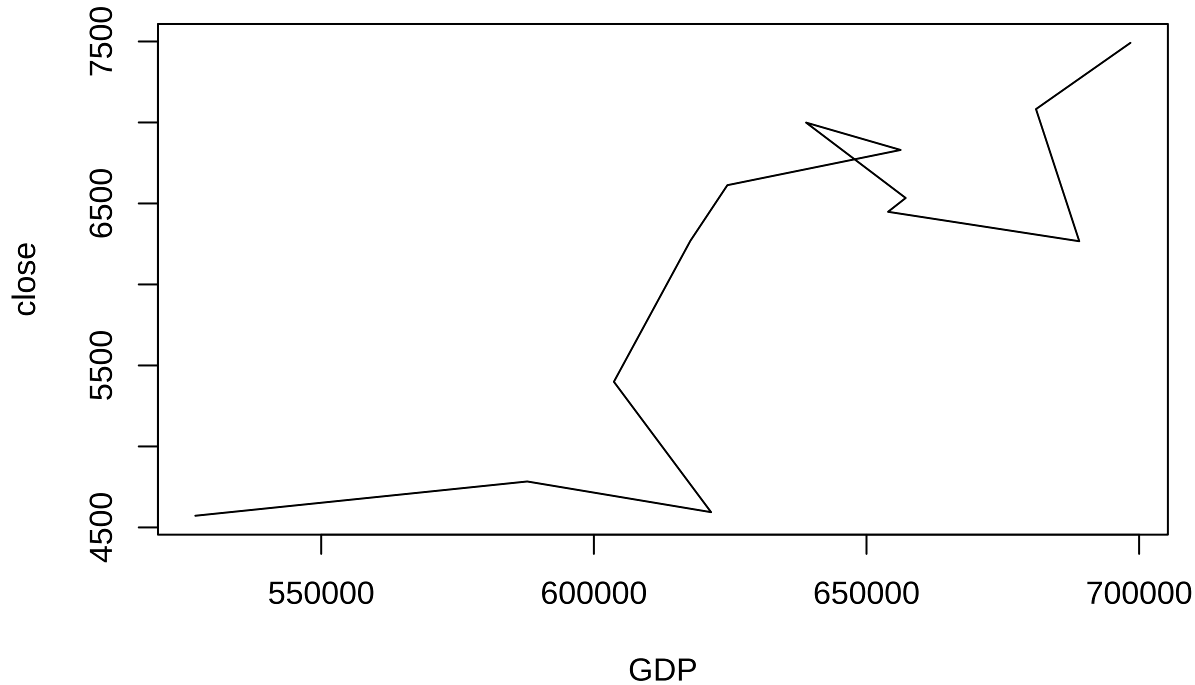
#assigning values to variables

```r
#assigning values to variables
close=merged_df$Close.x
GDP=merged_df$CPMNACNSAB1GQFR
CCI=merged_df$Value.x
Exchange=merged_df$Close.y
Unem=merged_df$Value
Inflation=merged_df$Value.y
budget=merged_df$`Budget monthly statement`
```

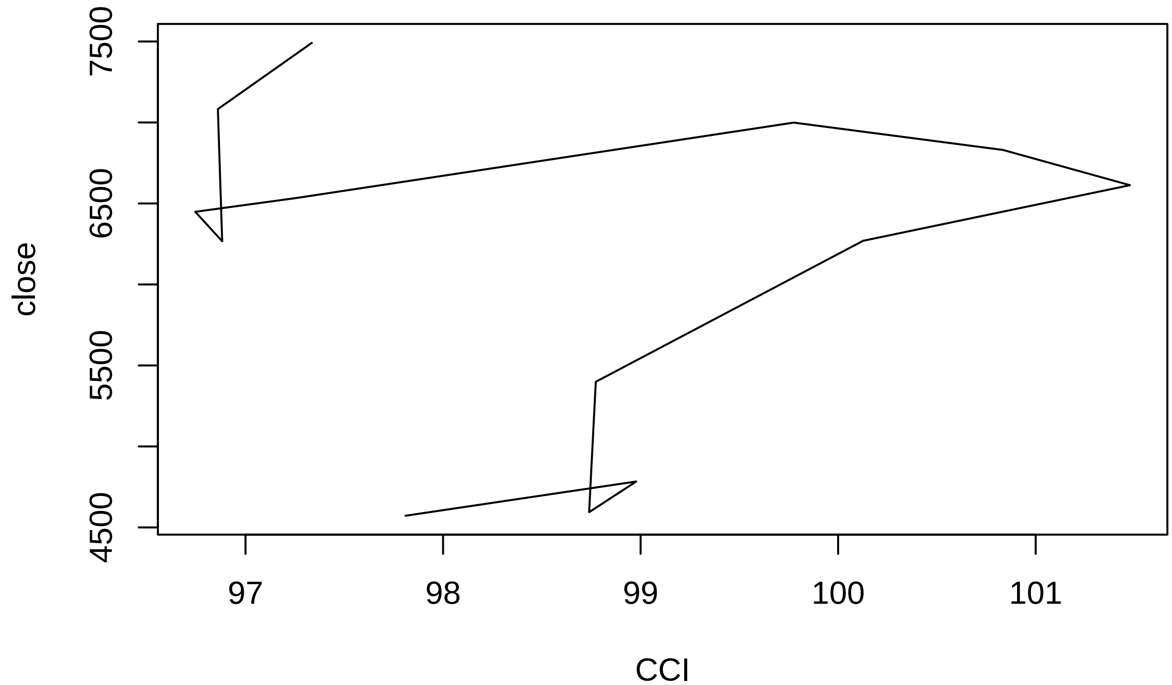
```r
#setting limits
maximun<-100
minimum<-0
```

#Plotting and checking each coorelation individually

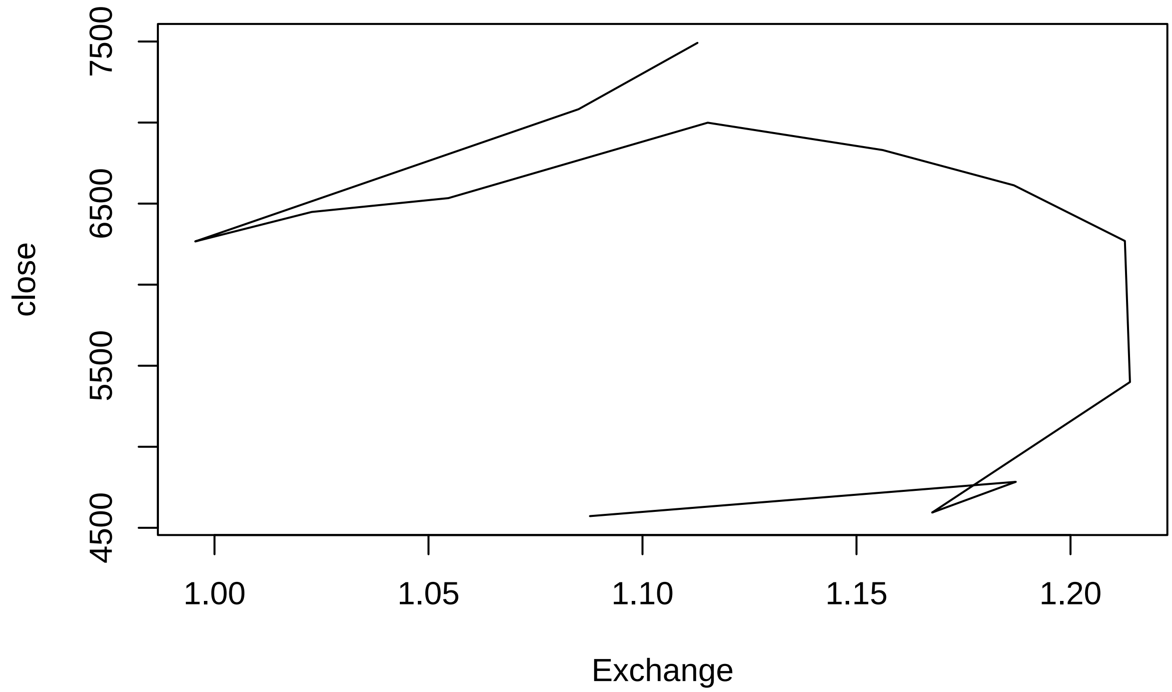
```r
#Plotting and checking each coorelation individually
plot (GDP, close, type = "l")
```



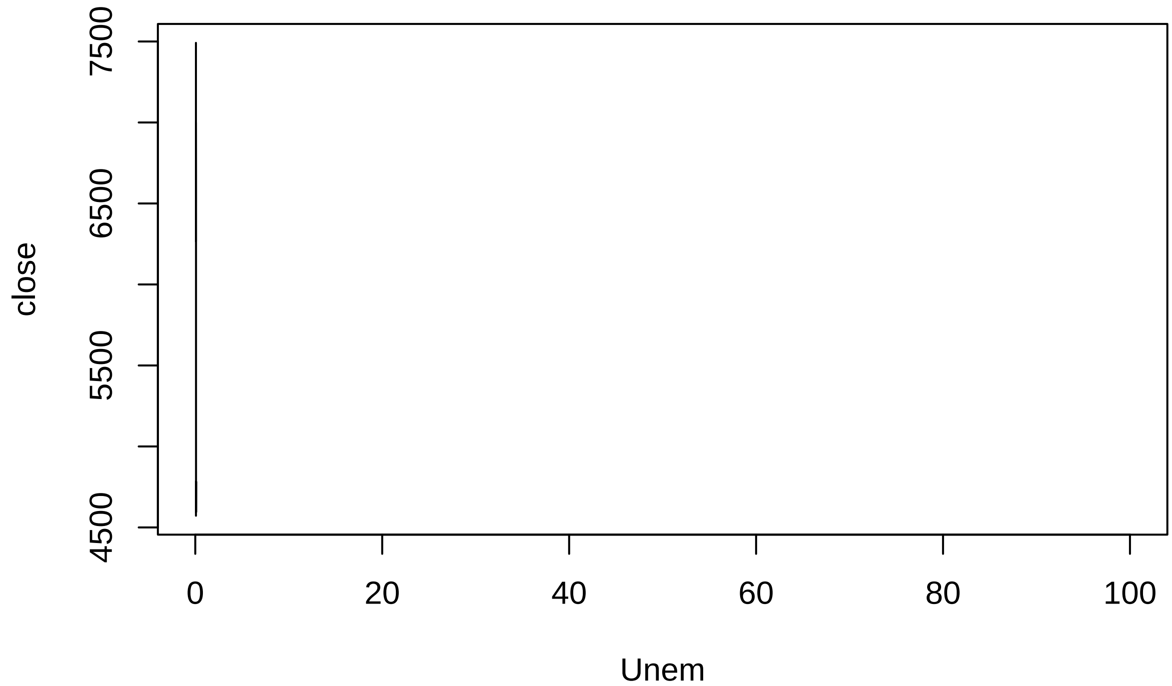

```r
plot (CCI, close, type = "l")
```



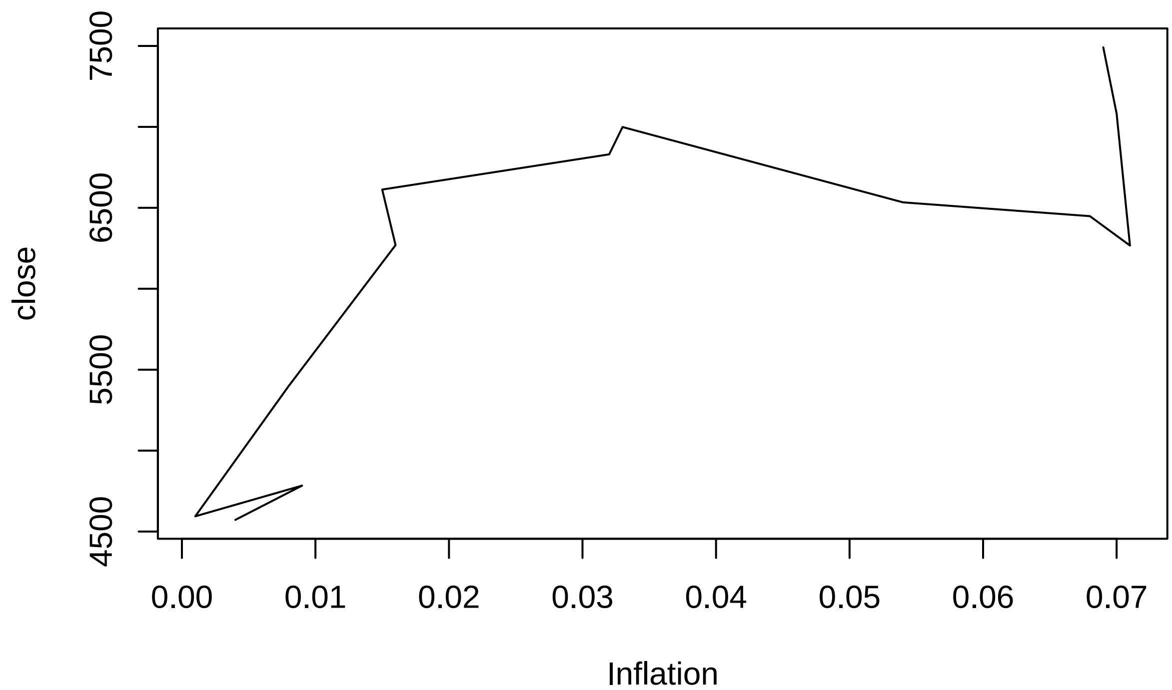

```
plot (Exchange, close, type = "l")
```



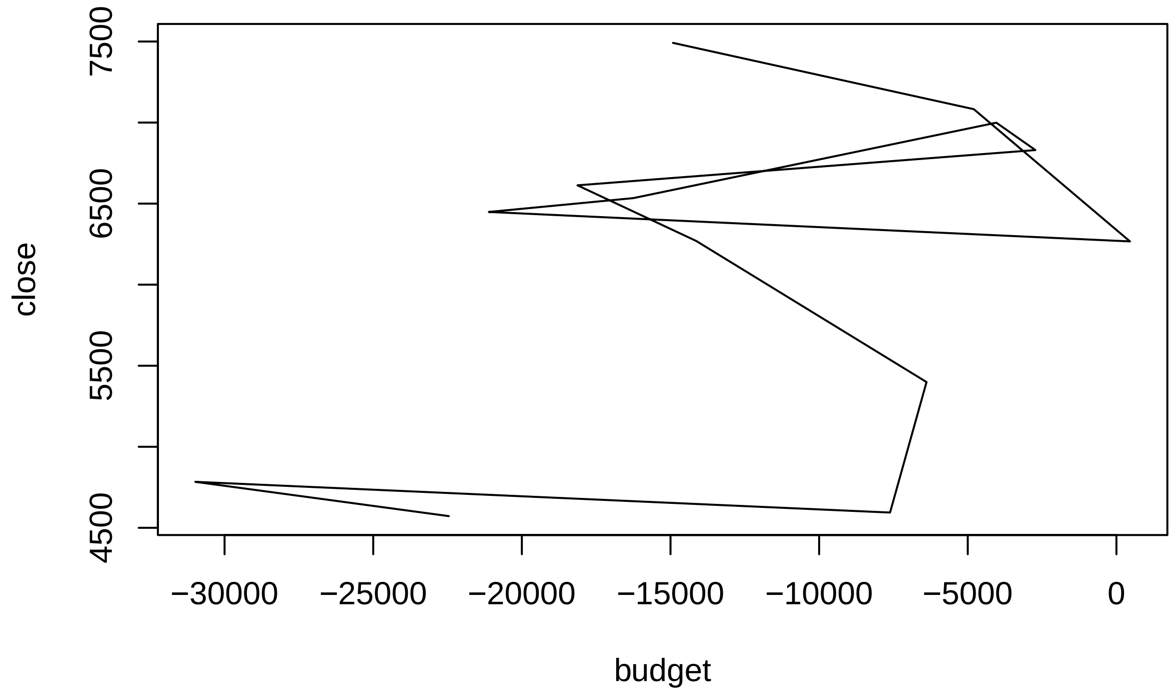

```
plot(Unem, close, type = "l", xlim = c(minimum,maximun))
```




```r
plot (Inflation, close, type = "l")
```




```r
plot (budget, close, type = "l")
```

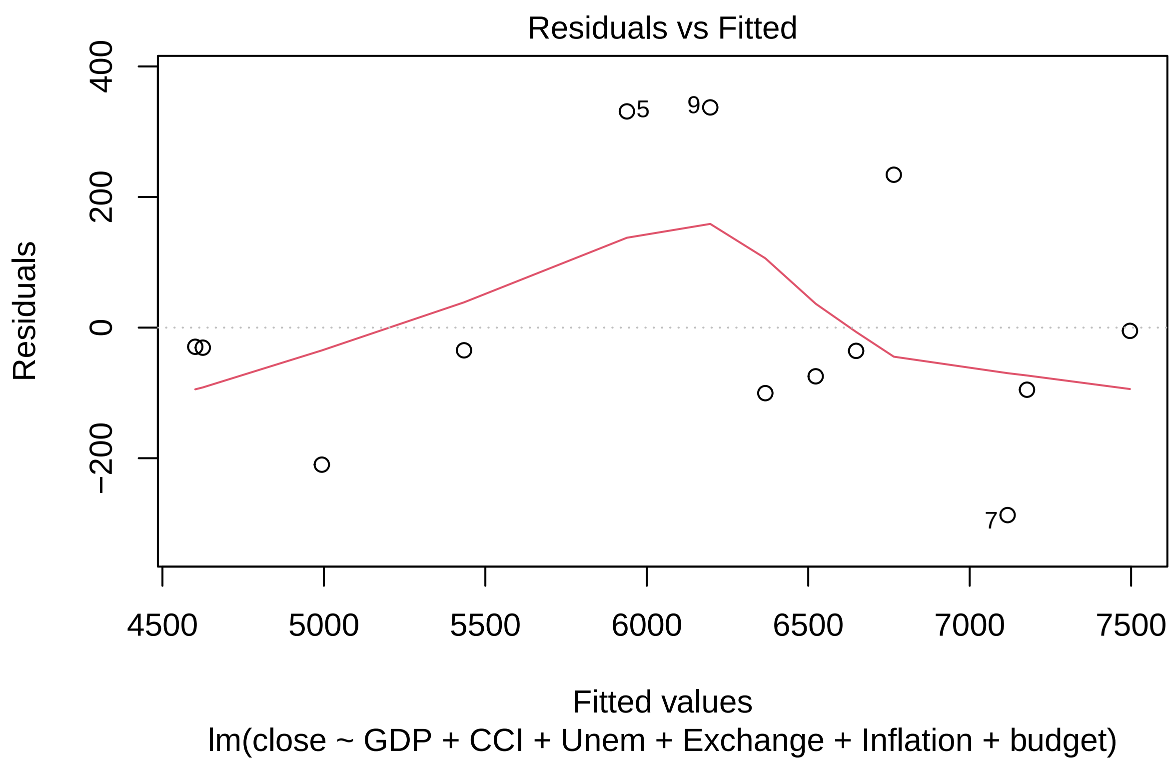


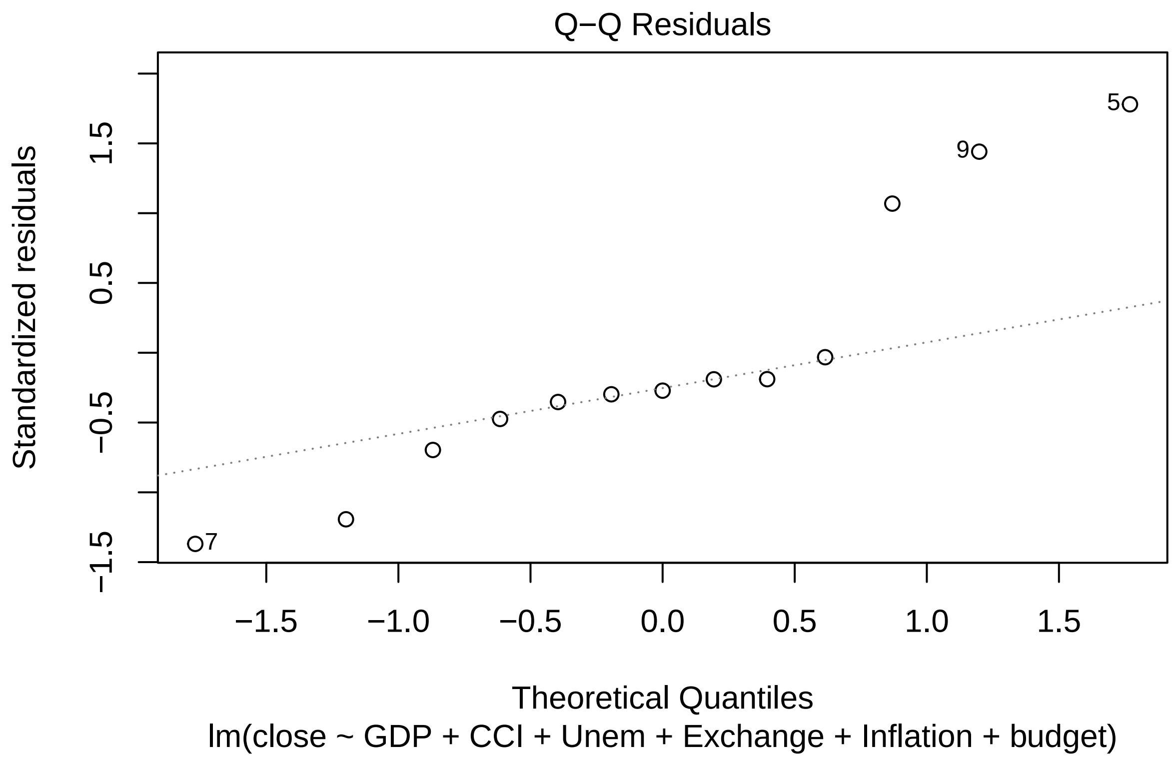

```r
#forming a linear regression model
mod.ols <- lm(close~GDP+CCI+Unem+Exchange+Inflation+budget)
summary (mod.ols)
```

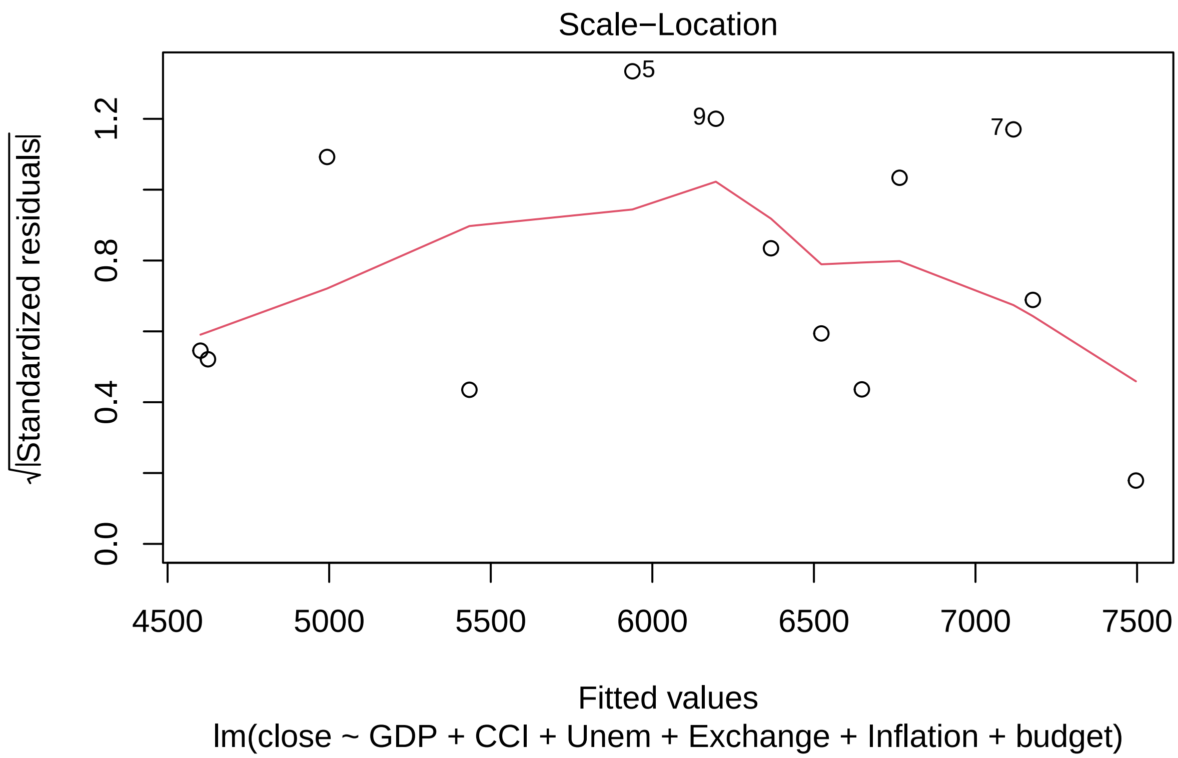
```
##
## Call:
## lm(formula = close ~ GDP + CCI + Unem + Exchange + Inflation +
##     budget)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -287.09  -95.13  -34.76   -5.00  337.23
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.300e+04  9.378e+03  -2.452   0.0496 *
## GDP          5.860e-03  7.694e-03   0.762   0.4751
## CCI          2.765e+02  9.953e+01   2.778   0.0321 *
## Unem        -1.430e+05  5.598e+04  -2.554   0.0433 *
## Exchange     7.167e+03  2.487e+03   2.882   0.0280 *
## Inflation    2.398e+04  1.912e+04   1.254   0.2565
## budget      -1.943e-02  1.728e-02  -1.124   0.3038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 268.8 on 6 degrees of freedom
```

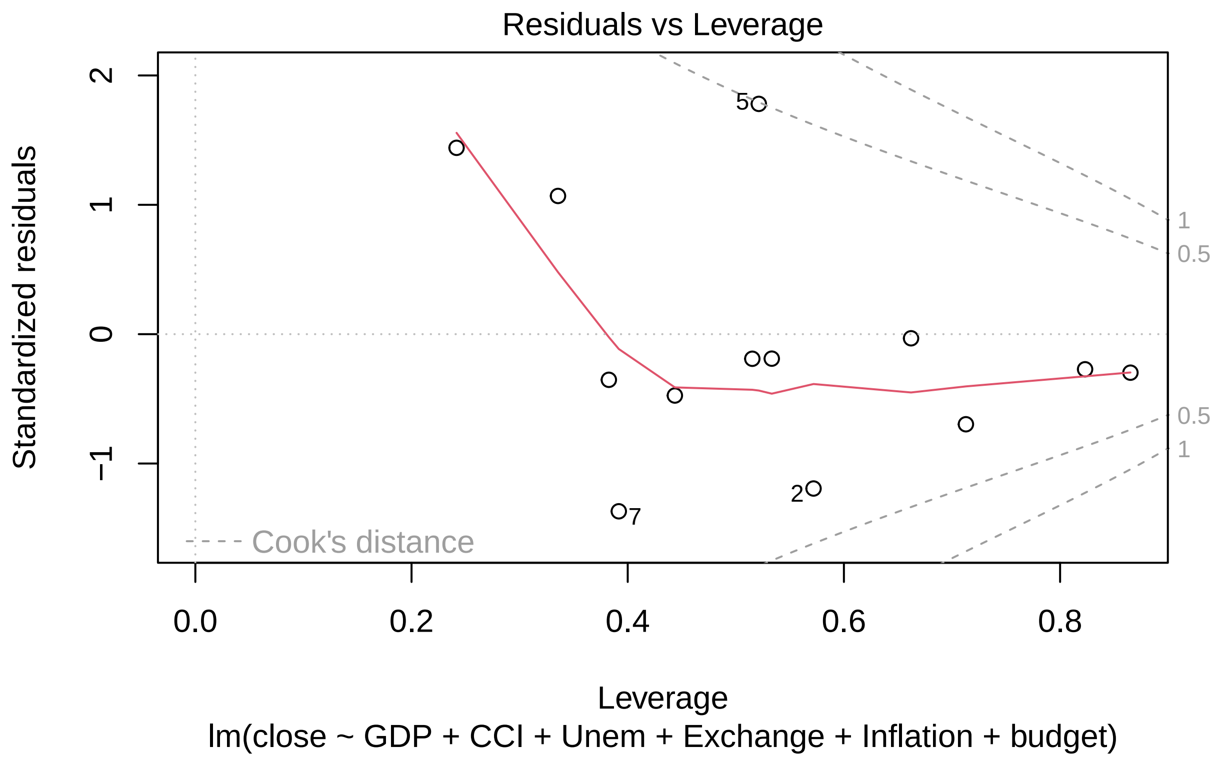
```
## Multiple R-squared:  0.9628, Adjusted R-squared:  0.9257
## F-statistic: 25.92 on 6 and 6 DF,  p-value: 0.0004847
```

```r
plot (mod.ols)
```




Q−Q Residuals
Standardized residuals
−1.5
−0.5
0.5
1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
5
9
7
Theoretical Quantiles
lm(close ~ GDP + CCI + Unem + Exchange + Inflation + budget)


Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
4500
5000
5500
6000
6500
7000
7500
Fitted values
lm(close ~ GDP + CCI + Unem + Exchange + Inflation + budget)
5
9
7



After analyzing the linear regression model we got to know that there is a greater p value in GDP INFLATION and BUDGET than 5% despite 0.9 R Square thus this model is incorrect ALSO the Scaled-location Graph Shows a non linear line depicting a error in the coeficient or any other error and residuals vs levereage confirms some possible outliers

So we will try to see if log transormed linear model works or not

```r
# Checking log transformed linear model
demand.log.linear = lm (log(close)~log(GDP)+log(CCI)+log(Unem)+log(Exchange)+log(Inflation)+log(budget)
```

```
## Warning in log(budget): NaNs produced
```

```r
summary(demand.log.linear)
```

```
##
## Call:
## lm(formula = log(close) ~ log(GDP) + log(CCI) + log(Unem) + log(Exchange) +
##     log(Inflation) + log(budget))
##
## Residuals:
## ALL 1 residuals are 0: no residual degrees of freedom!
##
## Coefficients: (6 not defined because of singularities)
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.743        NaN     NaN      NaN
## log(GDP)             NA         NA      NA       NA
```

```
## log(CCI)               NA          NA      NA       NA
## log(Unem)              NA          NA      NA       NA
## log(Exchange)          NA          NA      NA       NA
## log(Inflation)         NA          NA      NA       NA
## log(budget)            NA          NA      NA       NA
##
## Residual standard error: NaN on 0 degrees of freedom
##   (12 observations deleted due to missingness)
```

```
return(demand.log.linear)
```

```
##
## Call:
## lm(formula = log(close) ~ log(GDP) + log(CCI) + log(Unem) + log(Exchange) +
##     log(Inflation) + log(budget))
##
## Coefficients:
##    (Intercept)        log(GDP)        log(CCI)       log(Unem)   log(Exchange)
##          8.743              NA              NA              NA              NA
## log(Inflation)     log(budget)
##             NA              NA
```

The incorrect model cannot be corrected even after log transforming the incorrect linear model

So now we will check what is wrong with the linear model by Doing the anova test and After that AIC test

```
# Doing the anovea test
f1=anova(lm((close~GDP)))
f2=anova(lm((close~GDP+CCI)))
f3=anova(lm((close~GDP+CCI+Unem)))
f4=anova(lm((close~GDP+CCI+Unem+Exchange)))
f5=anova(lm((close~GDP+CCI+Unem+Exchange+Inflation)))
f6=anova(lm((close~GDP+CCI+Unem+Exchange+Inflation+budget)))

summary (f1) # to have the coefficients and R
```

```
##        Df          Sum Sq           Mean Sq           F value
##  Min.   : 1.0   Min.   :4222314   Min.   : 383847   Min.   :19.41
##  1st Qu.: 3.5   1st Qu.:5029039   1st Qu.:2150189   1st Qu.:19.41
##  Median : 6.0   Median :5835765   Median :3916531   Median :19.41
##  Mean   : 6.0   Mean   :5835765   Mean   :3916531   Mean   :19.41
##  3rd Qu.: 8.5   3rd Qu.:6642490   3rd Qu.:5682873   3rd Qu.:19.41
##  Max.   :11.0   Max.   :7449216   Max.   :7449216   Max.   :19.41
##                                                     NA's   :1
##      Pr(>F)
##  Min.   :0.001054
##  1st Qu.:0.001054
##  Median :0.001054
##  Mean   :0.001054
##  3rd Qu.:0.001054
##  Max.   :0.001054
##  NA's   :1
```

```r
summary (f2)
```

```
##        Df          Sum Sq           Mean Sq          F value
##  Min.   : 1.0   Min.   : 655061   Min.   : 356725   Min.   : 1.836
##  1st Qu.: 1.0   1st Qu.:2111157   1st Qu.: 505893   1st Qu.: 6.598
##  Median : 1.0   Median :3567253   Median : 655061   Median :11.359
##  Mean   : 4.0   Mean   :3890510   Mean   :2820334   Mean   :11.359
##  3rd Qu.: 5.5   3rd Qu.:5508234   3rd Qu.:4052138   3rd Qu.:16.121
##  Max.   :10.0   Max.   :7449216   Max.   :7449216   Max.   :20.882
##                                                     NA's   :1
##      Pr(>F)
##  Min.   :0.001027
##  1st Qu.:0.052070
##  Median :0.103114
##  Mean   :0.103114
##  3rd Qu.:0.154158
##  Max.   :0.205201
##  NA's   :1
```

```r
summary(f3)
```

```
##        Df        Sum Sq           Mean Sq          F value
##  Min.   :1   Min.   : 655061   Min.   : 187085   Min.   : 3.501
##  1st Qu.:1   1st Qu.:1426590   1st Qu.: 538067   1st Qu.: 6.784
##  Median :1   Median :1783626   Median :1269274   Median :10.068
##  Mean   :3   Mean   :2917882   Mean   :2543712   Mean   :17.795
##  3rd Qu.:3   3rd Qu.:3274918   3rd Qu.:3274918   3rd Qu.:24.942
##  Max.   :9   Max.   :7449216   Max.   :7449216   Max.   :39.817
##                                                  NA's   :1
##      Pr(>F)
##  Min.   :0.0001393
##  1st Qu.:0.0057259
##  Median :0.0113124
##  Mean   :0.0351898
##  3rd Qu.:0.0527151
##  Max.   :0.0941177
##  NA's   :1
```

```r
summary(f4)
```

```
##        Df          Sum Sq           Mean Sq          F value
##  Min.   :1.0   Min.   : 308746   Min.   : 171878   Min.   : 1.796
##  1st Qu.:1.0   1st Qu.: 655061   1st Qu.: 308746   1st Qu.: 3.307
##  Median :1.0   Median :1375021   Median : 655061   Median : 7.385
##  Mean   :2.4   Mean   :2334306   Mean   :2093677   Mean   :14.977
##  3rd Qu.:1.0   3rd Qu.:1883486   3rd Qu.:1883486   3rd Qu.:19.054
##  Max.   :8.0   Max.   :7449216   Max.   :7449216   Max.   :43.340
##                                                    NA's   :1
##      Pr(>F)
##  Min.   :0.0001724
##  1st Qu.:0.0080628
##  Median :0.0486928
```

```
##  Mean   :0.0786331
##  3rd Qu.:0.1192632
##  Max.   :0.2169743
##  NA's   :1
```

```r
summary(f5)
```

```
##        Df         Sum Sq           Mean Sq           F value
##  Min.   :1   Min.   : 308746   Min.   :  75001   Min.   : 4.117
##  1st Qu.:1   1st Qu.: 557522   1st Qu.: 395325   1st Qu.: 8.734
##  Median :1   Median : 752537   Median : 752537   Median :11.333
##  Mean   :2   Mean   :1945255   Mean   :1870254   Mean   :29.724
##  3rd Qu.:1   3rd Qu.:1625118   3rd Qu.:1625118   3rd Qu.:25.113
##  Max.   :7   Max.   :7449216   Max.   :7449216   Max.   :99.321
##                                                  NA's   :1
##      Pr(>F)
##  Min.   :0.0000219
##  1st Qu.:0.0015456
##  Median :0.0119774
##  Mean   :0.0233665
##  3rd Qu.:0.0212453
##  Max.   :0.0820425
##  NA's   :1
```

```r
summary(f6)
```

```
##        Df             Sum Sq           Mean Sq           F value
##  Min.   :1.000   Min.   :  91382   Min.   :  72271   Min.   :  1.264
##  1st Qu.:1.000   1st Qu.: 371186   1st Qu.: 200064   1st Qu.:  5.470
##  Median :1.000   Median : 655061   Median : 655061   Median : 10.413
##  Mean   :1.714   Mean   :1667361   Mean   :1615739   Mean   : 25.916
##  3rd Qu.:1.000   3rd Qu.:1366749   3rd Qu.:1366749   3rd Qu.: 22.486
##  Max.   :6.000   Max.   :7449216   Max.   :7449216   Max.   :103.073
##                                                      NA's   :1
##      Pr(>F)
##  Min.   :0.0000531
##  1st Qu.:0.0051528
##  Median :0.0188304
##  Mean   :0.0713249
##  3rd Qu.:0.0691067
##  Max.   :0.3037765
##  NA's   :1
```

ANOVA TEST 1 The group means appear to differ significantly, as indicated by the F-value (F value) of 19.41. With a p-value of 0.001054, the observed differences are not likely to be the result of chance. Overall, the results of the ANOVA point to statistical significance in the group variability.

ANOVA TEST 2 The group means appear to differ significantly, as indicated by the F-value (F value) of 11.359. With a p-value of 0.103114, the observed differences are likely to be the result of chance. ANOVA TEST 3 The group means appear to differ significantly, as indicated by the F-value (F value) of 17.795 . With a p-value of 0.0351898, the observed differences are not likely to be the result of chance. Overall, the results of the ANOVA point to statistical significance in the group variability. ANOVA TEST 4 The group means appear to differ significantly, as indicated by the F-value (F value) of 14.977 . With a p-value of

0.0786331, the observed differences are likely to be the result of chance. ANOVA TEST 5 The group means appear to differ significantly, as indicated by the F-value (F value) of 25.916 . With a p-value of 0.0233665, the observed differences are not likely to be the result of chance. Overall, the results of the ANOVA point to statistical significance in the group variability.

ANOVA TEST 6 The group means appear to differ significantly, as indicated by the F-value (F value) of 25.916. With a p-value of 0.0713249 , the observed differences are likely to be the result of chance.

Setting linear regression with 1 variable and adding each variable till complete regression to find best model

```
sat.lm1 <- lm(close ~ GDP) # estimate the regression with 1 varible
sat.lm2 <- lm((close~GDP+CCI)) # estimate the regression with 2 varibles
sat.lm3<-lm((close~GDP+CCI+Unem))
sat.lm4<-lm((close~GDP+CCI+Unem+Exchange))
sat.lm5<-lm((close~GDP+CCI+Unem+Exchange+Inflation))
sat.lm6<-lm((close~GDP+CCI+Unem+Exchange+Inflation+budget))
```

NOW we Do the AIC TEST to make a better linear regression model

```
#AIC VERIFICATION
sat.n <- length(close) # number of observations
sat.sse1 <- sum(resid(sat.lm1) ^2) # the sum of squared residuals
AIC.selfmade <- sat.n + sat.n*log(2*pi) + sat.n * log(sat.sse1 / sat.n) + 2 * (2+1)
AIC.selfmade
```

```
## [1] 207.8747
```

```
AIC(sat.lm1, k=2)
```

```
## [1] 207.8747
```

```
#AIC function
AIC(sat.lm1, k=2)
```

```
## [1] 207.8747
```

```
AIC(sat.lm2, k=2)
```

```
## [1] 207.683
```

```
#AIC  for 2 variable regression
AIC(sat.lm2, k=2)
```

```
## [1] 207.683
```

```
#AIC and BIC for  3 variable model
AIC(sat.lm3, k=2)
```

```
## [1] 199.9231
```

```r
#AIC  for 4 variables
AIC(sat.lm4,k=2)
```

```
## [1] 199.2898
```

```r
#AIC for 5 variables
AIC(sat.lm5,k=2)
```

```
## [1] 188.7733
```

```r
#AIC for all variables
AIC(sat.lm6,k=2)
```

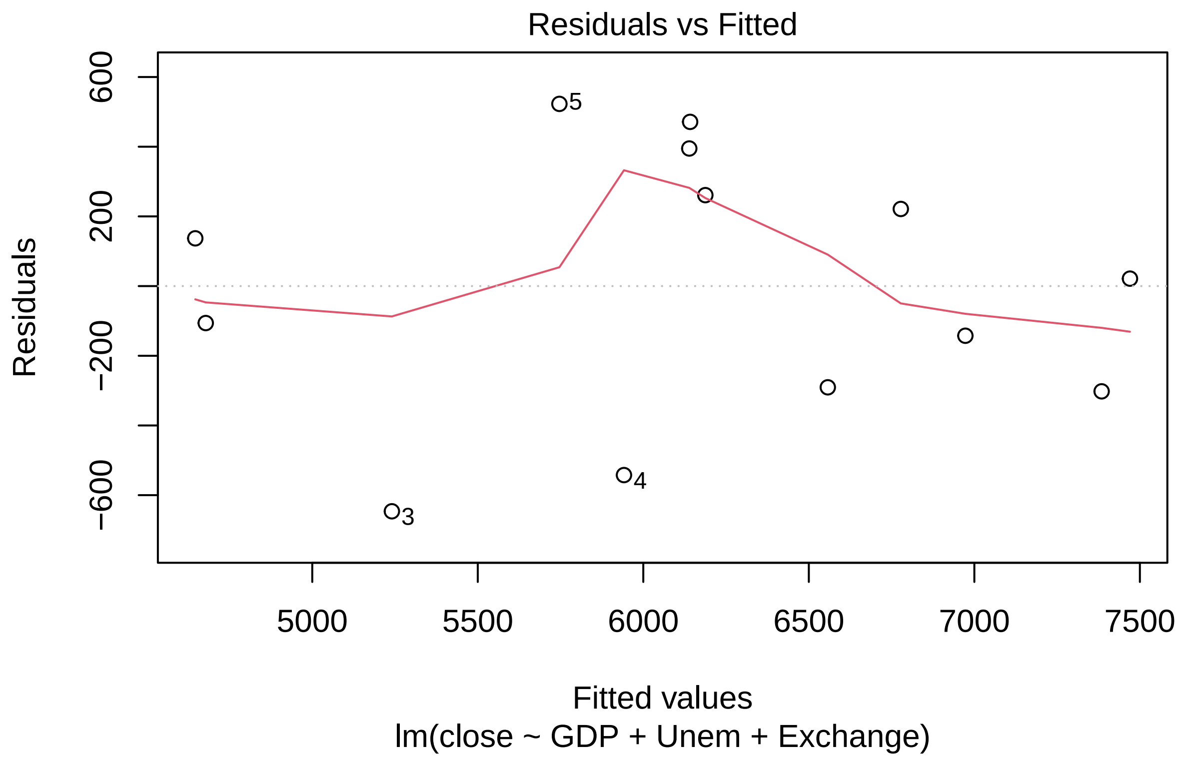
```
## [1] 188.2873
```

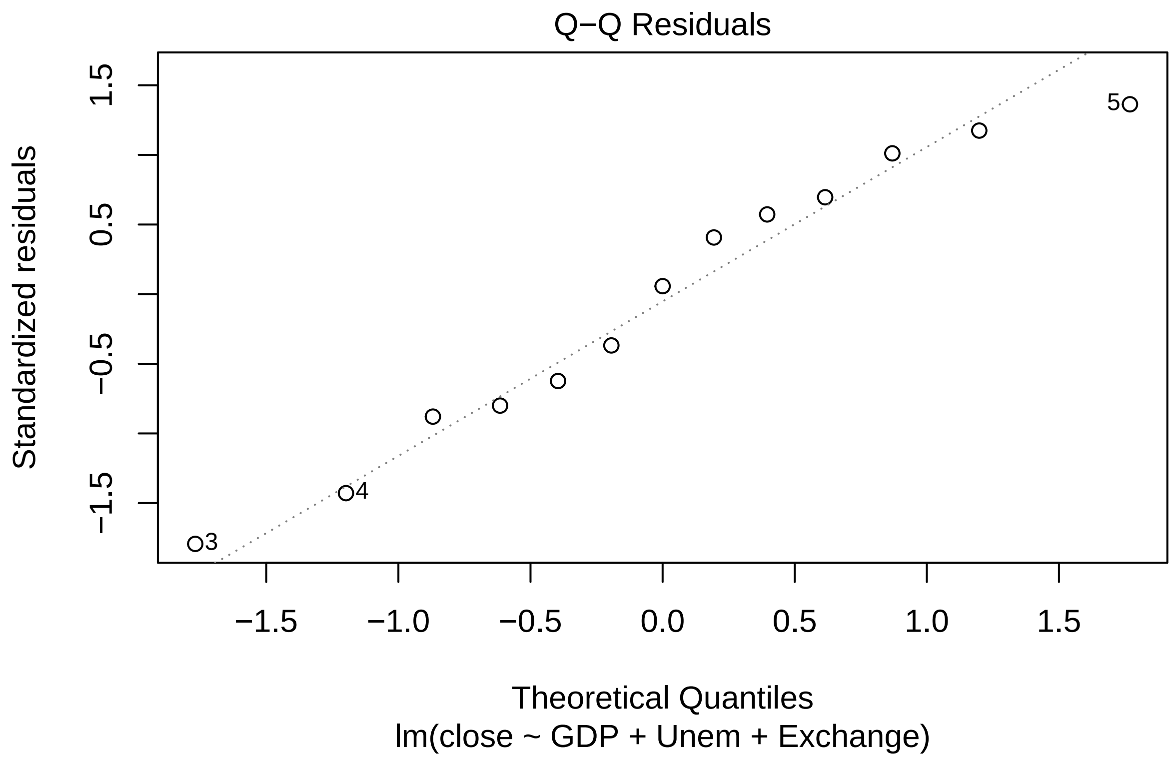
Looking at the AIC analysis we found that there is a problem with CCI and Budget as it has only slight difference from AIC 1 to AIC 2 (from 207.8747 to 207.683) and from AIC 5 and AIC 6.So we remove both of them from the linear regression.Also, by applying the theory we found out that inflation directly affects GDP. And, Due to the rule that there should be no correlation between independent variables. So, we remove inflation also.

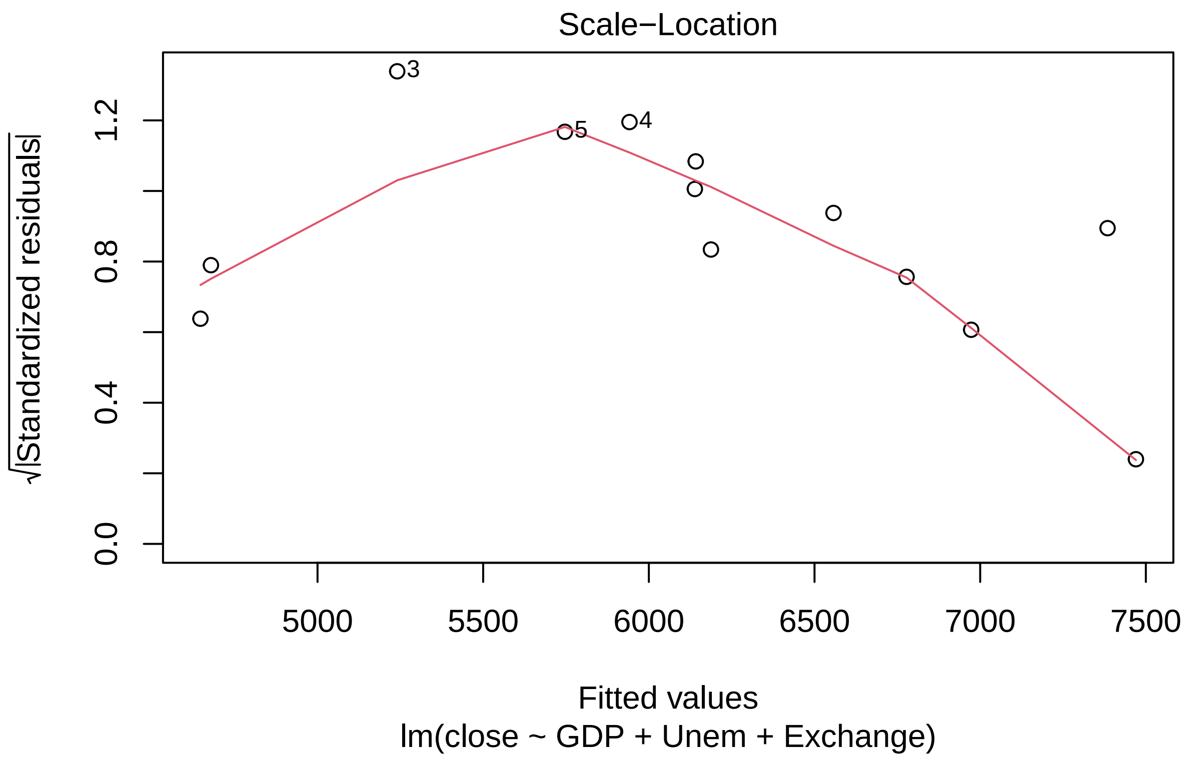
```r
# as after AIC test We know that monthly budget and CCI  are not that effective , so we make new mode;
new_model<-lm(close~GDP+Unem+Exchange)
summary(new_model)
```

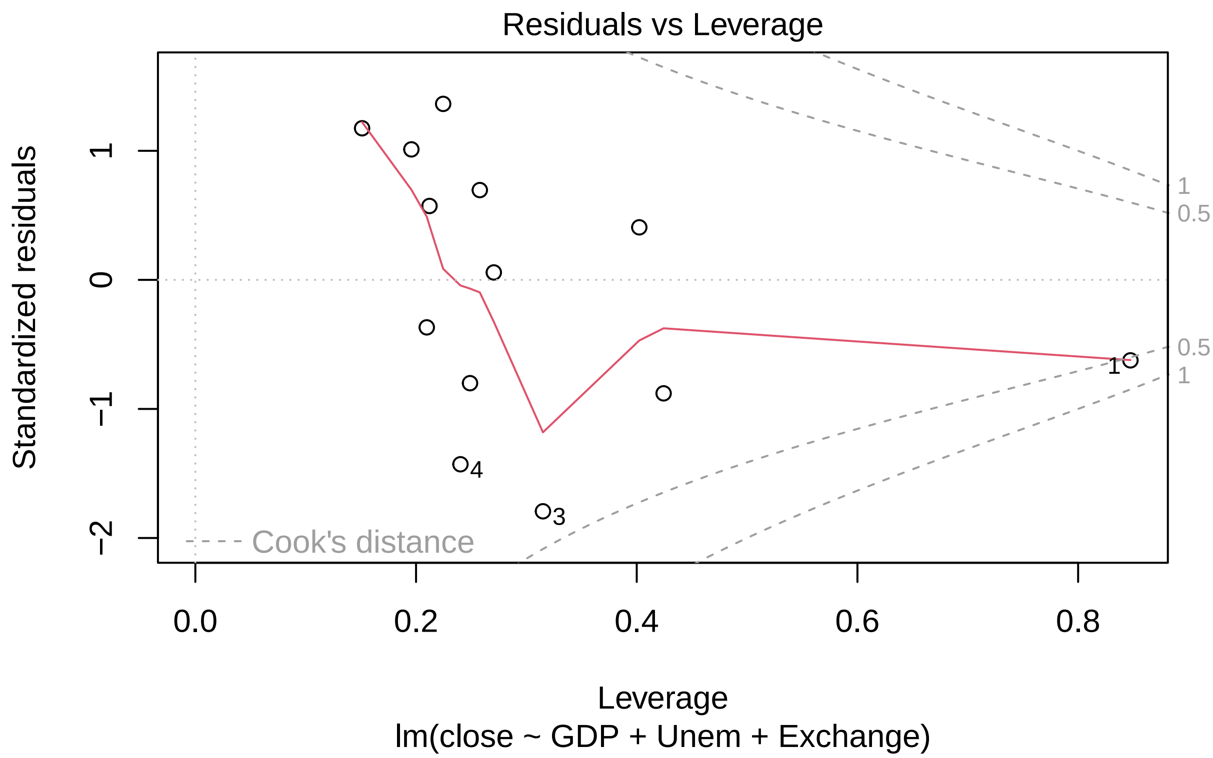
```
##
## Call:
## lm(formula = close ~ GDP + Unem + Exchange)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -646.34 -290.55   21.41  261.04  522.90
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.730e+03  4.257e+03   0.641  0.53730
## GDP          1.197e-02  3.383e-03   3.539  0.00632 **
## Unem        -1.765e+05  4.903e+04  -3.600  0.00574 **
## Exchange     8.326e+03  2.800e+03   2.974  0.01561 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 435.5 on 9 degrees of freedom
## Multiple R-squared:  0.8537, Adjusted R-squared:  0.805
## F-statistic: 17.51 on 3 and 9 DF,  p-value: 0.0004247
```

```r
plot(new_model)
```


Residuals vs Fitted
Residuals
600
200
−200
−600
5
4
3
5000
5500
6000
6500
7000
7500
Fitted values
lm(close ~ GDP + Unem + Exchange)


Q−Q Residuals
Standardized residuals
1.5
0.5
−0.5
−1.5
5
4
3
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(close ~ GDP + Unem + Exchange)

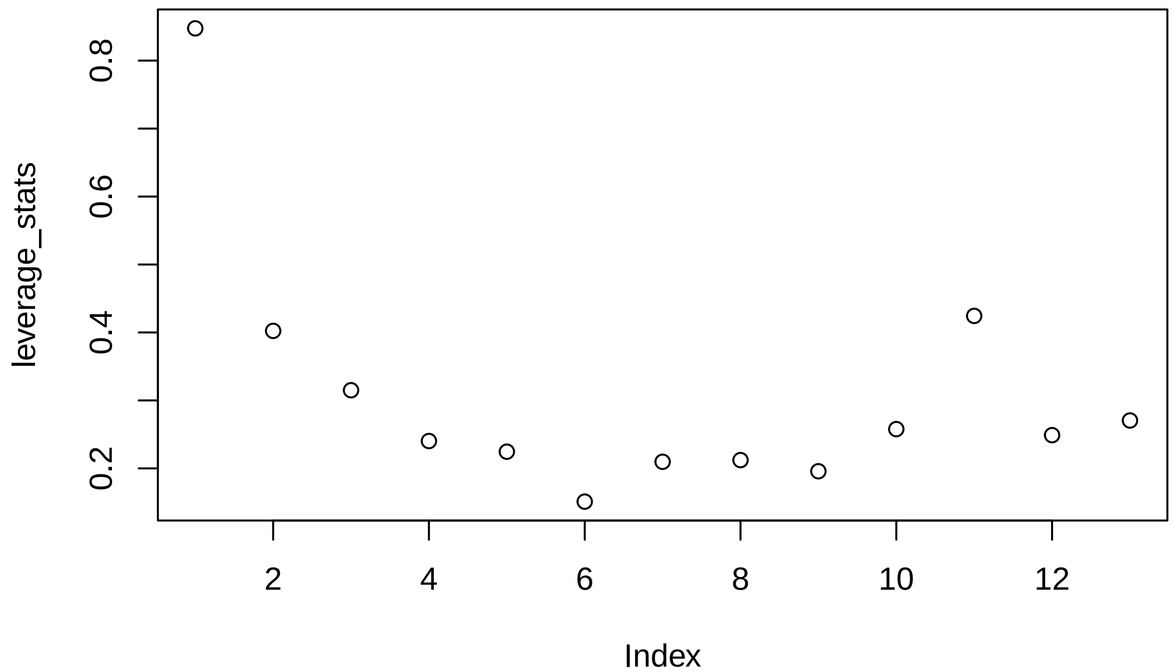

Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
3
5
4
5000
5500
6000
6500
7000
7500
Fitted values
lm(close ~ GDP + Unem + Exchange)



In this Model we analysed that the new regression model has all the p values below 5%. Also, the scaled-location graph is much better than the previous model.Lastly, there is only one possible outlier according to residuals vs leverage graph with a R squared as 0.805

Now we check the leverage for the new model

```
leverage_stats <- hatvalues(new_model)
plot(leverage_stats)
```



now we will also check if the log transformed model is better fit for the linear model or nor

```r
# Checking log transformed linear moddel
demand.log.linear = lm (log(close)~log(GDP)+log(Unem)+log(Exchange))
summary(demand.log.linear)
```

```
##
## Call:
## lm(formula = log(close) ~ log(GDP) + log(Unem) + log(Exchange))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.13046 -0.03695 -0.01195  0.04927  0.10349
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -14.3934     4.4362  -3.245  0.01009 *
## log(GDP)        1.2761     0.3704   3.445  0.00733 **
## log(Unem)      -2.2902     0.6827  -3.354  0.00846 **
## log(Exchange)   1.5062     0.5595   2.692  0.02472 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07898 on 9 degrees of freedom
## Multiple R-squared:  0.8392, Adjusted R-squared:  0.7856
## F-statistic: 15.66 on 3 and 9 DF,  p-value: 0.0006464
```

CONCLUSION As the log transformed model has r squared = 0.785,we conclude that in the new model the closing stocks of CAC 40 is correlated to GDP UNEMPLOYEMENT AND EXCHANGE RATe CONCLUSION