statistical_project

2024-05-27

Project Introduction

In this project we aim to study the influence of macroeconomics indicators on the house price index in Colombia. This indicator measure the evolution through time of the market prices of residential properties as a percentage change. The idea of the analysis is to present and study the possible effects of the macroeconomics indicators on the house prices through the construction of regression models, evaluating their results and interpreting the models in the context of the data.

Dataset Introduction and Preprocessing

The dataset was constructed gattering information from multiple sites: the Colombian department of statistics (DANE) , the Colombian central bank (Banco de la Repúlbica), Google trends and the Federal Reserve Economic Data (FRED). In general, all the visited sites present the data as a .csv file with tables according to filters related with the time span of interest. With this we construct a consolidated database in .csv format with the following variables:

Variable	Description
House Price Index	Measure of the average change in the residential properties price as an index
Industrial Inputs Index	Measure of the average changes in the industry input costs as an index
Metals Price Index	Measure of the average price change in metal costs excluding gold as an index
Energy Price Index	Measure of the average price change of energy including Crude oil, Natural Gas, Coal Price and Propane Indices
Shipping Price Index	Measure the average price change of shipping costs
Forex Index	Indicator of the foreign exchange rate of the Colombian Peso (COP) with respect of the USD
Unemployment Rate	Indicator of the percentage of unemployment population in Colombia
Industrial Production Index	Measure of the level of production on industrial sectors as an index
Interest rate	Reference interest rate emited by the colombian central bank with respect of the others financial institutions
Construction Licences Area	Measure the total of squared meters given for construction licences in Colombia
Finished Constructions	Measure of the total constructions that have been finished in Colombia
Google Trends Housing	Google search trends on housing for Colombia

These variables present real values,and were selected as they have an initial coherent relation with the housing sector. We present the import process of the data

```
#`echo = FALSE`: Hides the R code in the final document.
#`results = 'hide'`: Hides the output of the R code.
```

Data Uploading

```
#Read the data using the read_excel function
#Change here for current data path
data <- read_excel("C:/Users/CAMILO/Documents/GitHub/unipd_sl_24/data/data.xlsx",sheet = "dataframe_col
#View(data)
#Present the first 5 rows of the data
#head(data,5)
```

Data pre-processing and cleaning

After have an initial version of the data we proceed with the pre-processing and cleaning steps. As a first approximation we identify the number of NA values in each one of the columns.

```
#We first change the variables names to have more consistency

colnames(data) <- c('MY','date','Home_Price_Index','Industrial_Inputs_Index','Metals_Price_Index','Energ
# Counting NA values in each column
na_counts <- apply(data, 2, function(x) sum(is.na(x)))
# Print the counts of NA values per column
print(na_counts)
```

```
##                          MY                        date
##                           0                           0
##           Home_Price_Index     Industrial_Inputs_Index
##                           0                           0
##         Metals_Price_Index         Energy_Price_Index
##                           0                           0
##       Shipping_Price_Index                Forex_Index
##                           0                           0
##          Unemployment_Rate Industrial_Production_Index
##                           0                          12
##              Interest_Rate  Construction_Licences_Area
##                           0                          12
##     Finished_Constructions       Google_Trends_Housing
##                         158                           0
```

```
# Construction and Industrial prod have nans,
# they are at the extremes
# Convert date column to Date type if it's not already
data$date <- as.Date(data$date)

# Copy the dataframe to avoid modifying the original one
data_filled <- data
```

We realize that the variable finished_constructions requires an additional pre-processing step because it is recorded by quarters while the other are registered by months. We propose the following reconstruction steps:

```
# Interpolate finished_constructions
# Find the indices where the date is at the end of a trimester
trimester_end_indices <- which(format(data$date, "%m") %in% c("03", "06", "09", "12"))

# Loop through each trimester end and distribute the value to the previous three months
for (i in trimester_end_indices) {
  if (i - 2 > 0) {
    # Distribute the value to the current month and the previous two months
    value_to_distribute <- data$Finished_Constructions[i] / 3
    data_filled$Finished_Constructions[i] <- value_to_distribute
    data_filled$Finished_Constructions[i - 1] <- value_to_distribute
    data_filled$Finished_Constructions[i - 2] <- value_to_distribute
  }
}

data_filled$Finished_Constructions <- na.locf(data_filled$Finished_Constructions, na.rm = FALSE)

#View(data_filled)
```

Summary stats and final dataset

We proceed to remove the left rows with NA values eliminating from the dataset the registers before January 2005, also we drop the column MY also referred to the date in form of MONTH()YEAR(). Finally, we present a summary of the dataset after these transformations. In particular we see that the Housing Price Index range present values on [87.35,203.50] with an IQR=70.38.

```
# Removing rows with any NA values
clean_data <- na.omit(data)

# Drop columns ------------

data_reduced <- subset(clean_data, select = -c(MY))
#head(data_reduced,10)
```

```
# View description ---------------
# Get the summary of the dataframe
summary_stats <- summary(select(data_reduced,-date))

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

Variable	Min	1st Qu.	Median	Mean	3rd Qu.	Max
Home_Price_Index	87.53	111.17	158.02	147.84	180.79	203.50
Industrial_Inputs_Index	82.63	118.00	133.95	137.42	158.04	210.75
Metals_Price_Index	78.87	121.60	141.59	145.37	170.82	233.75
Energy_Price_Index	72.99	128.96	161.54	173.97	223.62	331.78

Variable	Min	1st Qu.	Median	Mean	3rd Qu.	Max
Shipping_Price_Index	-213.5	243.7	262.8	279.8	294.3	478.3
Forex_Index	0.1712	1.1934	2.376	2.671	3.196	4.788
Unemployment_Rate	0.08499	0.09684	0.10490	0.10853	0.11402	0.20359
Industrial_Production_Index	62.19	79.09	96.86	95.27	108.54	132.46
Interest_Rate	-0.0175	0.03844	0.04512	0.05494	0.06688	0.13250
Construction_Licences_Area	675268	958182	1260614	1276163	1452165	2883878
Finished_Constructions	1413931	2329601	2554152	2618787	2886264	4007432
Google_Trends_Housing	17.50	22.50	33.25	34.59	45.38	63.00

Exploratory Analysis

Outliers based on the IQR

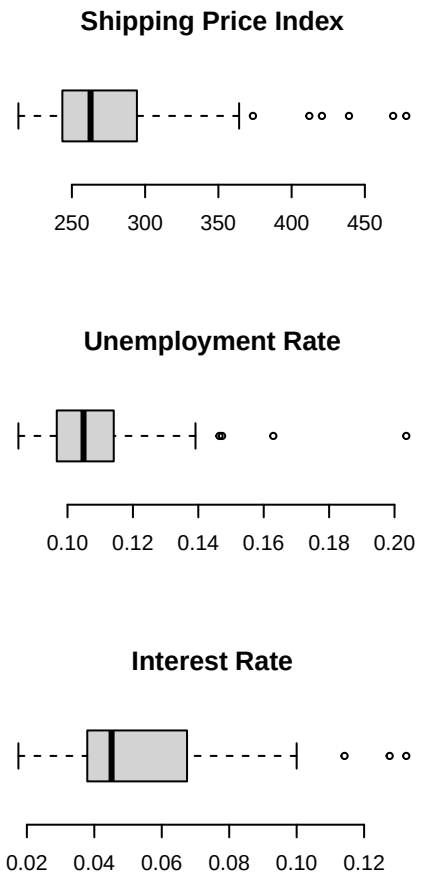
As an initial step to better understand the data and its behavior we find the outliers of each one of the variables using the IQR to select those points outside the bounds. As a result we found that the variables that present the most quantity of outliers are the Shipping Price Index, the Interest Rate and the Unemployment Rate.

```
# Function to detect outliers based on IQR
detect_outliers_single_var <- function(column) {
  column <- na.omit(column)
  Q1 <- quantile(column, 0.25, na.rm = TRUE)
  Q3 <- quantile(column, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR

  outliers <- column[column < lower_bound | column > upper_bound]
  return(outliers)
}

# List to store outliers for each variable
outliers_list <- list()

# Loop through each column in the dataset
for (var in colnames(select(data_reduced,-date))) {
  if (is.numeric(data[[var]])) {
    outliers_list[[var]] <- detect_outliers_single_var(data[[var]])
  }
}

# Print outliers for each variable
#print(outliers_list)
```

Variable	Values
Home_Price_Index	0
Industrial_Inputs_Index	0
Metals_Price_Index	0

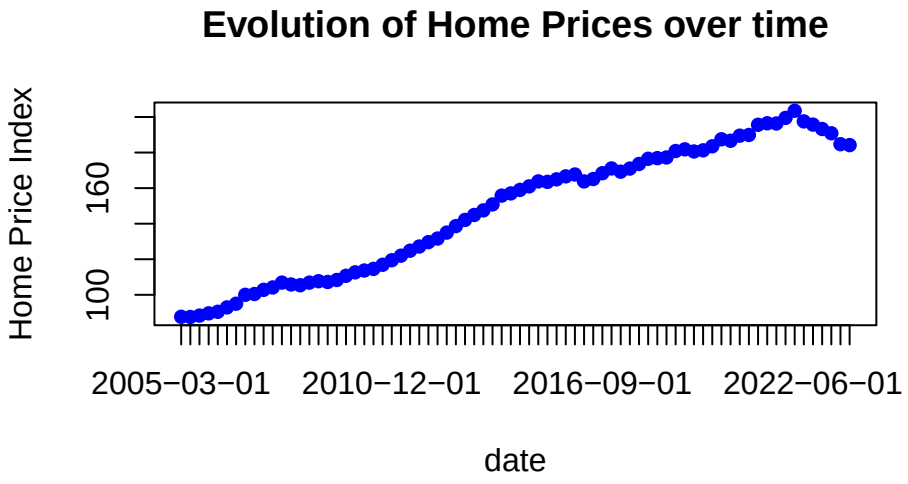
Variable	Values
Energy_Price_Index	376.4121
Shipping_Price_Index	374.969, 375.187, 395.268, 446.983, 439.181, 454.407, 468.721, 478.263, 478.216, 471.877, 469.290, 442.591, 456.584, 412.082, 409.599, 443.799, 420.694, 446.214, 462.462
Forex_Index	0
Unemployment_Rate	0.1725640, 0.2048530, 0.2197200, 0.2035910, 0.2091470, 0.1744270, 0.1629280, 0.1756237
Industrial_Production_Index	0
Interest_Rate	0.1141935, 0.1204839, 0.1275000, 0.1275806, 0.1300000, 0.1324194, 0.1325000, 0.1325000
Construction_Licences_Area	2493085, 1988396, 2838778, 2337743, 2331210, 2859035, 2372414, 2052607
Finished_Constructions	4007432, 3762696, 1413931
Google_Trends_Housing	0

In addition present the box plot for the variables with the highest quantity of outliers



Ploting the Home Price Index

Continuing with the exploratory analysis we plot the variable of Home Price Index to see its behavior through time. It is possible to see a clear increase tendency starting from the year 2005 until the year 2021, then a little decay was recorded.



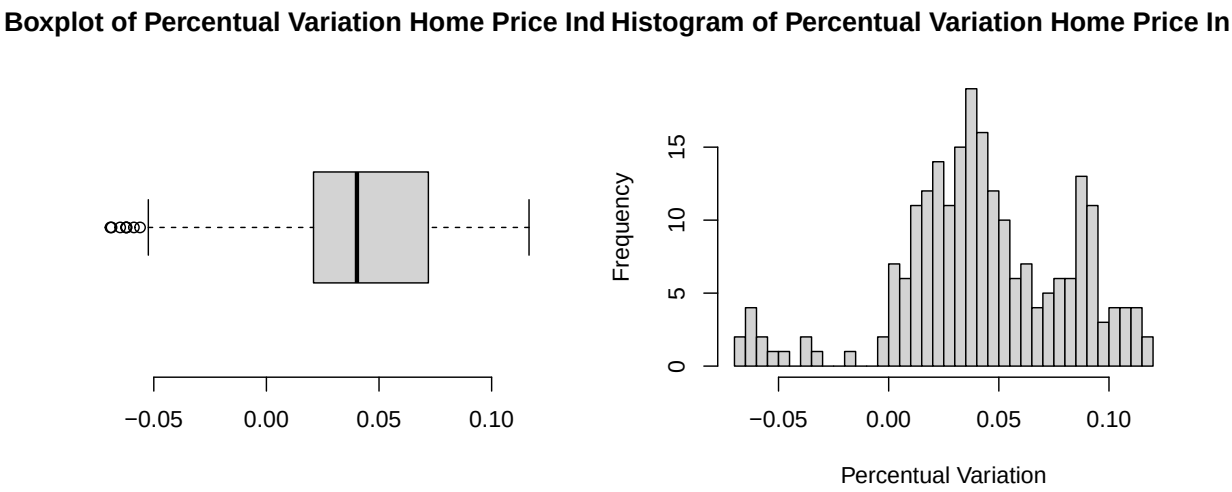
Ploting percentual variation of the Home Price Index

Another useful information can be found in the box plots and histograms for the percentual variations. We take the percentual variation of the variables with the intention of identify patterns and tendencies in a clearly way, considering that the majority of it is composed by index, and the study of it changes is fundamental for a complete understanding of their behavior throught time. We start with the Home Price Index variable.

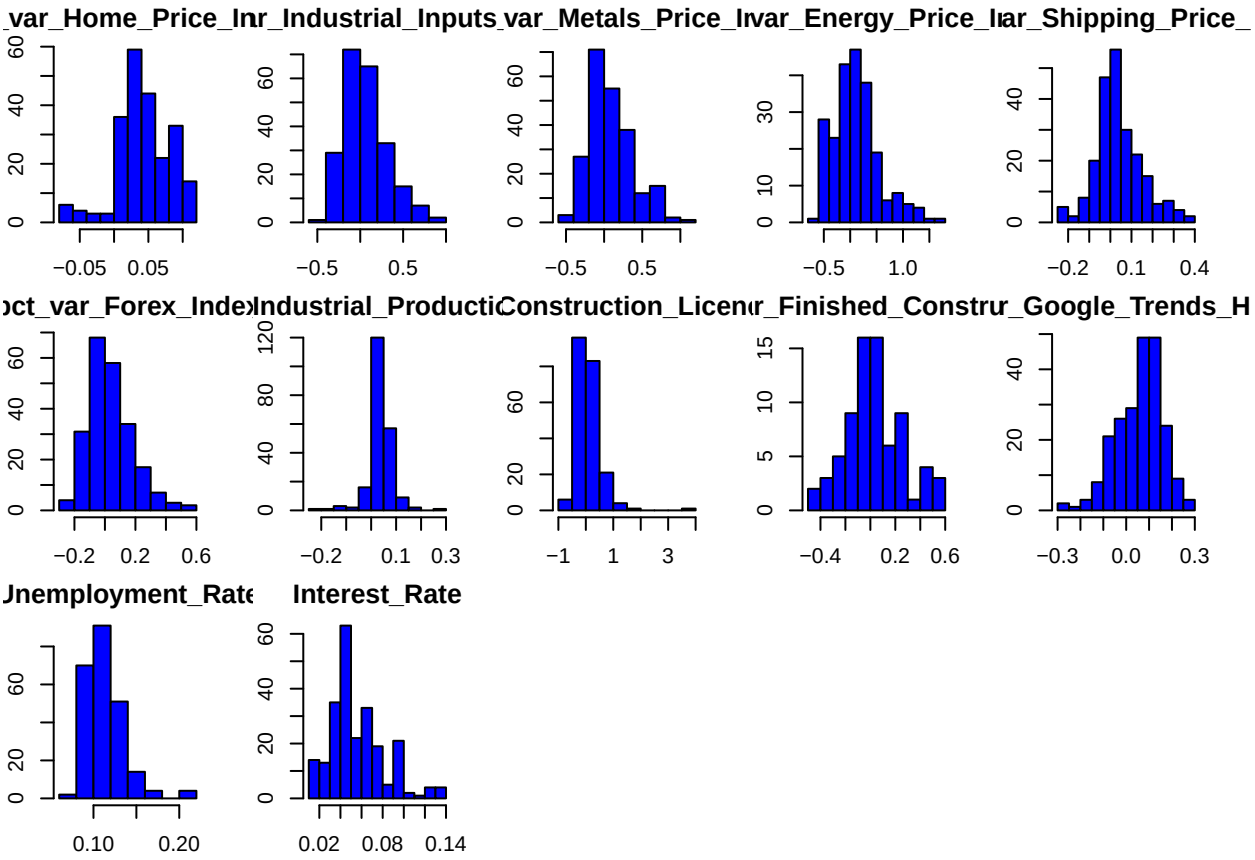
In these plots we see that the central value of the percentual variations is positive, indicating and overall increase in the Home Price Index through time. Also the negatives outliers were expected because those negatives variations due to the decrease after the 2021.In the histogram it is possible to see a right skewed behavior.

```
# Create Differences (Inflation)

# Select subset of variables to difference
variables_to_calculate <- c("Home_Price_Index", "Industrial_Inputs_Index", "Metals_Price_Index",
                            "Energy_Price_Index", "Shipping_Price_Index", "Forex_Index",
                            "Industrial_Production_Index", "Construction_Licences_Area",
                            "Finished_Constructions", "Google_Trends_Housing")

# Create a function to calculate the 12-month percentual variation
percentual_variation_12_months <- function(x) {
  return((x / lag(x, n = 12) - 1))
}

# Helper function to lag data
```

```r
lag <- function(x, n) {
  c(rep(NA, n), head(x, -n))
}

# Initialize the result data frame with the original data
data_percentual_variation <- data

# Apply the function to the subset of variables
for (var in variables_to_calculate) {
  new_var_name <- paste0("pct_var_", var)
  data_percentual_variation[[new_var_name]] <- percentual_variation_12_months(data[[var]])
}
```

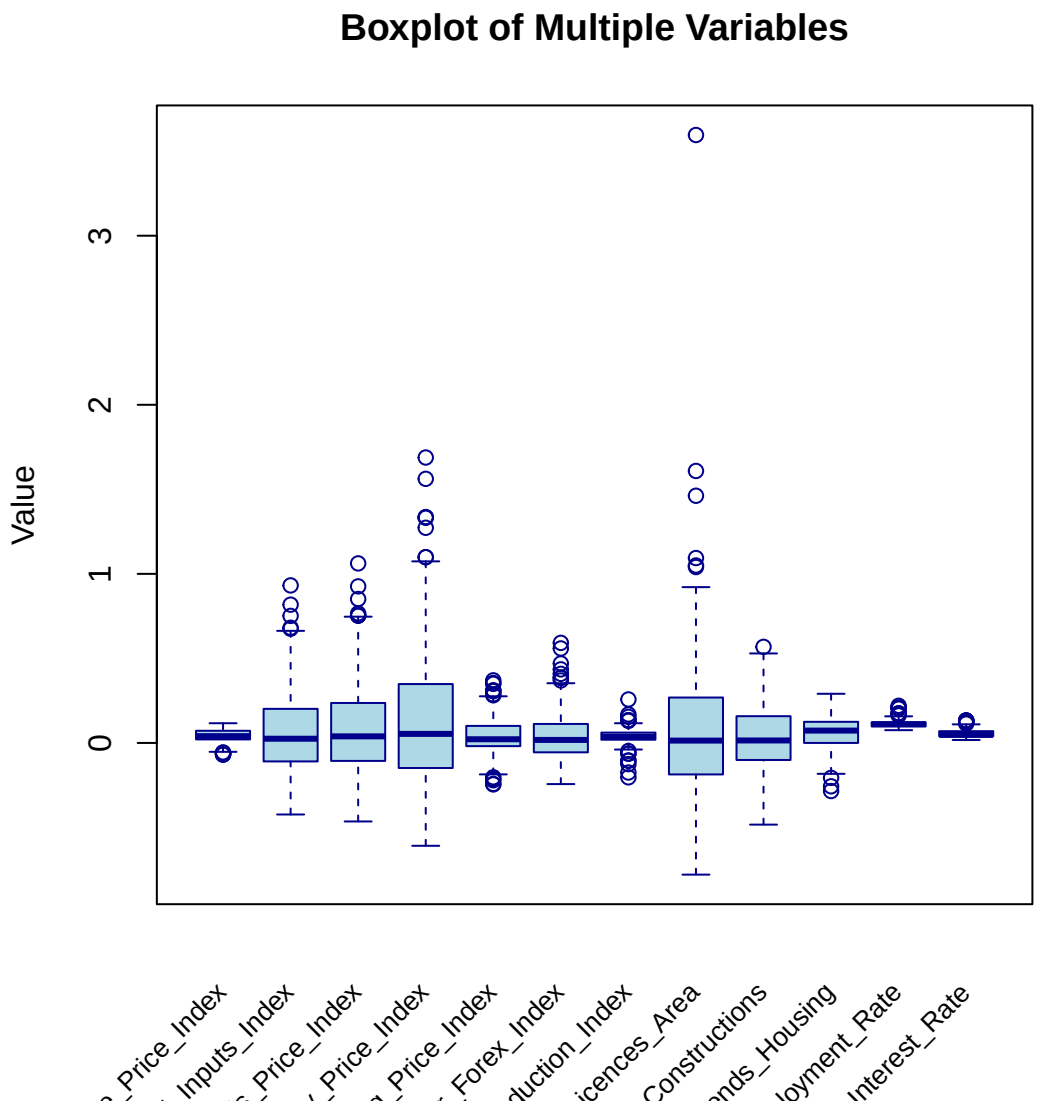


We did a similar procedure for the rest of the variables with respect to their percentual variation

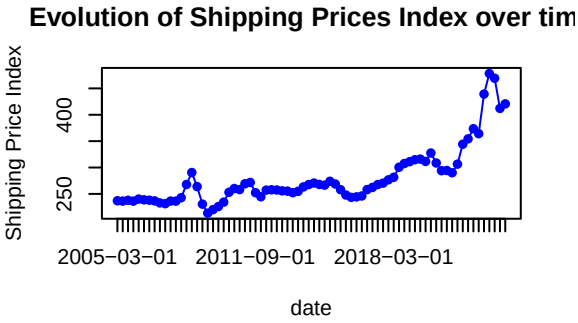


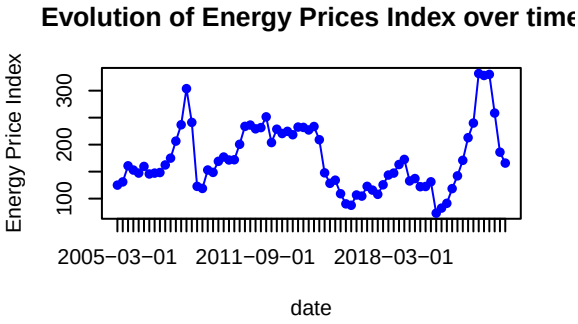
For an easier interpretation we plot the boxplot for the percentual variation of each variable in a same graph. This plot shows that the index variables with a high variability in percentual variation are the shipping index price and the energy index price. Also, the index variable that show less variability in the percentual variation is the one related with the industrial production.

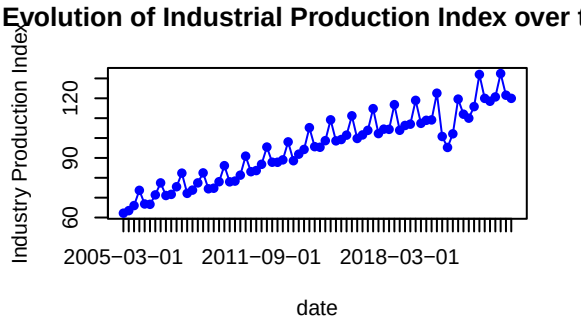
```
## No id variables; using all as measure variables
```



We plot these four variables to see its evolution through time. It is possible to see that the industrial production Index and the Shipping price Index have an increase tendency while the energy price have a less defined behavior over time with certain peaks in specific years like 2008 and 2022







Log-transformation

We apply a log transformation with the intention of normalize the data as some variables are index that change over time and others area real values in a wide range. Our idea is to reduce the variability of the data, compressing the scale and helping to linearize those relationships between the variables for the interpretability of the results in the models. We apply log transformation to all variables except those that are already percentages: the unemployment rate and the interest rate.

```
#Log transformation of the variables
variables_to_log <- c("Home_Price_Index", "Industrial_Inputs_Index", "Metals_Price_Index",
                      "Energy_Price_Index", "Shipping_Price_Index", "Forex_Index",
                      "Industrial_Production_Index", "Construction_Licences_Area",
                      "Finished_Constructions", "Google_Trends_Housing")
```

```r
for (var in variables_to_log) {
  log_var_name <- paste0(var, "_log")
  data_reduced[[log_var_name]] <- log(data_reduced[[var]])
}

# Subset the dataframe
selected_columns <- c(paste0(variables_to_log, "_log"), "Unemployment_Rate", "Interest_Rate")
data_final <- data_reduced[, selected_columns]
```

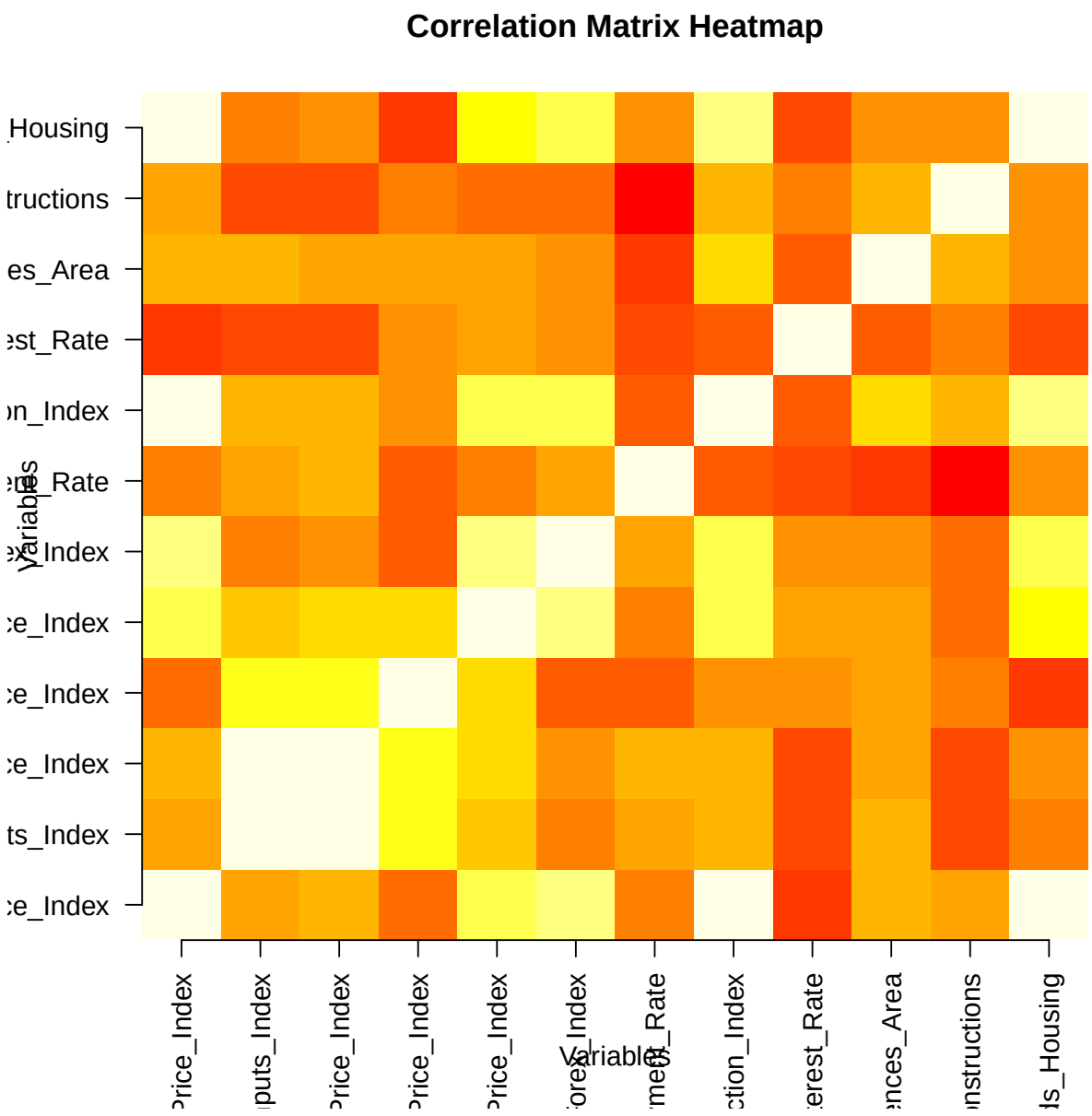
Covariance and correlation

Now we present the covariance and correlation between the variables of the dataset in order to understand the strength of their linear relation. In particular we show in the following table the result for the correlation of the variables with respect to our variable y (Housing Prices Index):

Variable	Correlation with Housing Prices Index
Industrial_inputs	0.1786
metals	0.2442
energy	-0.0339
shipping	0.6880
fx	0.7649
unemployment	-0.0114
industrial_prod	0.9447
interest_rate	-0.2695
construction_licences_area	0.2864
finished_constructions	0.1637
google_trends	0.9318

From the results we see a strong positive relationship between the house price index and the google trend one,a more moderate one with the shipping price index and the fx indicator. While the weaker relationships are found with the energy price index and the unemployment rate. These correlation serve as a point of reference for the linear regression models that we implement in the corresponding section. Also, we found a high positive correlation between the metal price index and the industrial inputs.

Revisar si tiene sentido esta matris



Multicolinearity Check

Now present a multicollinearity test in top of the results related with the correlation of the variables in relation with the dependent variable of House Price Index. We first calculate the VIF for each predictor variable. We found that some variables like Shipping, Industrial Production Index, Metals Price Index and Industrial Inputs present a high VIF.

```
##     Industrial_Inputs_Index_log          Metals_Price_Index_log
##                       262.712138                      271.834237
##          Energy_Price_Index_log        Shipping_Price_Index_log
##                         5.830493                       10.775109
##                Forex_Index_log Industrial_Production_Index_log
```

```
##                       10.263374                       18.804197
##  Construction_Licences_Area_log     Finished_Constructions_log
##                        1.852463                        2.188996
##       Google_Trends_Housing_log             Unemployment_Rate
##                        9.148957                        3.750313
##                  Interest_Rate
##                        2.052118
```

At this point we consider to remove the Metals Price Index as they show a strong correlation with Industrial Inputs, they both have a similar behavior and at the end the metal sector is consider on the Industrial Inputs Index. After this reduction we re run the VIF test.

```
##     Industrial_Inputs_Index_log          Energy_Price_Index_log
##                        3.443407                        5.830433
##        Shipping_Price_Index_log                 Forex_Index_log
##                       10.619605                       10.253715
## Industrial_Production_Index_log  Construction_Licences_Area_log
##                       15.170557                        1.645044
##      Finished_Constructions_log       Google_Trends_Housing_log
##                        2.061951                        9.130925
##              Unemployment_Rate                    Interest_Rate
##                        3.001225                        1.907184
```

Modelling the data

We start with a simple regression model for the data, remember that this process will be executed with the variables after the log normalization. In this first iteration we consider all the explanatory variables.

```
# OLS 1 Log-Log-----------------
# Perform linear regression
model <- lm(Home_Price_Index_log ~ ., data = data_final)
summary(model)
```

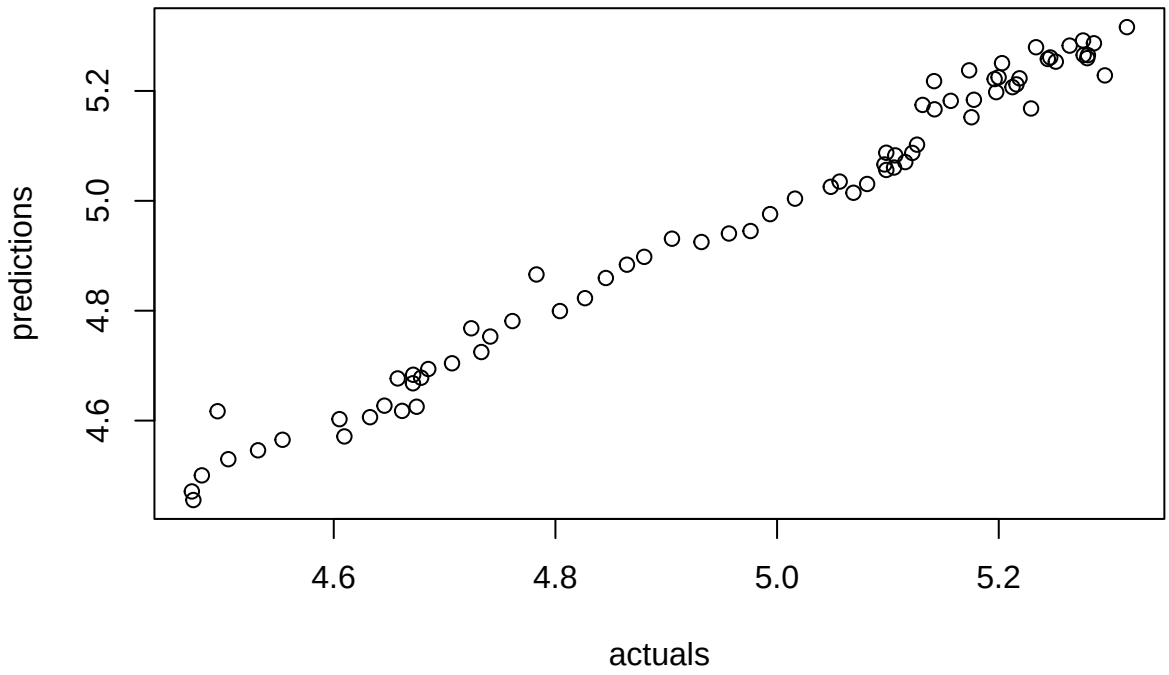
```
##
## Call:
## lm(formula = Home_Price_Index_log ~ ., data = data_final)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121869 -0.018344  0.001657  0.021147  0.067455
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      1.064351   0.544682   1.954   0.0551 .
## Industrial_Inputs_Index_log     -0.030898   0.033726  -0.916   0.3631
## Energy_Price_Index_log           0.045513   0.029890   1.523   0.1328
## Shipping_Price_Index_log         0.074346   0.078451   0.948   0.3469
## Forex_Index_log                 -0.068880   0.046670  -1.476   0.1450
## Industrial_Production_Index_log  0.658104   0.085562   7.692 1.24e-10 ***
## Construction_Licences_Area_log  -0.000204   0.017490  -0.012   0.9907
## Finished_Constructions_log      -0.030569   0.030628  -0.998   0.3221
```

```
## Google_Trends_Housing_log        0.400933   0.034058  11.772  < 2e-16 ***
## Unemployment_Rate                0.366194   0.387079   0.946   0.3477
## Interest_Rate                   -0.463352   0.226089  -2.049   0.0446 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03606 on 63 degrees of freedom
## Multiple R-squared:  0.9837, Adjusted R-squared:  0.9812
## F-statistic: 381.3 on 10 and 63 DF,  p-value: < 2.2e-16
```

In this first model we see a high R-squared, this mean that the model is powerful in terms of explain the variance in the House Price Index, but the model is no completely satisfactory as it includes several predictors without statistical significance, and other with different degree of signifance as industrial production, google trends and interest rate. This can be explain because the R-squared statistic will always increase when more variables are added to the model.

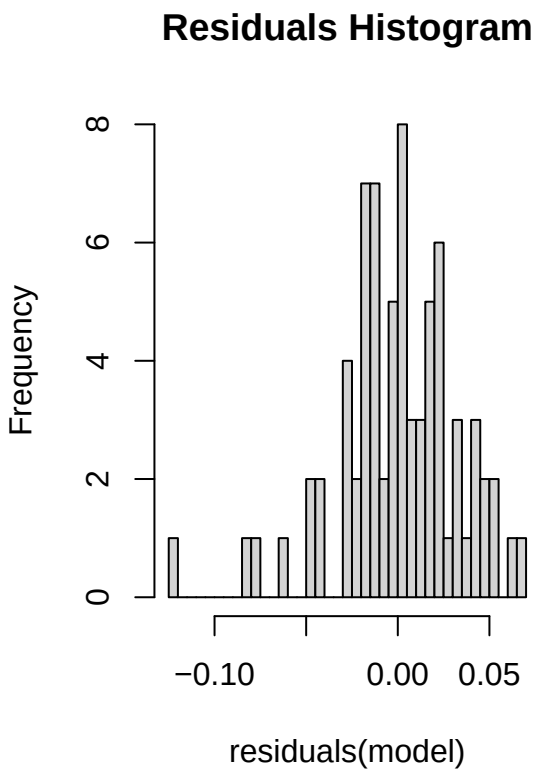
After having the model we fit it to evaluate its performance. First we determine the MSE which has a value of 0.001106908 suggesting the the predictions are close to the actual values, this could be observed in the plot of the predicted vs the actual values which presents an approximately 45° degrees line.

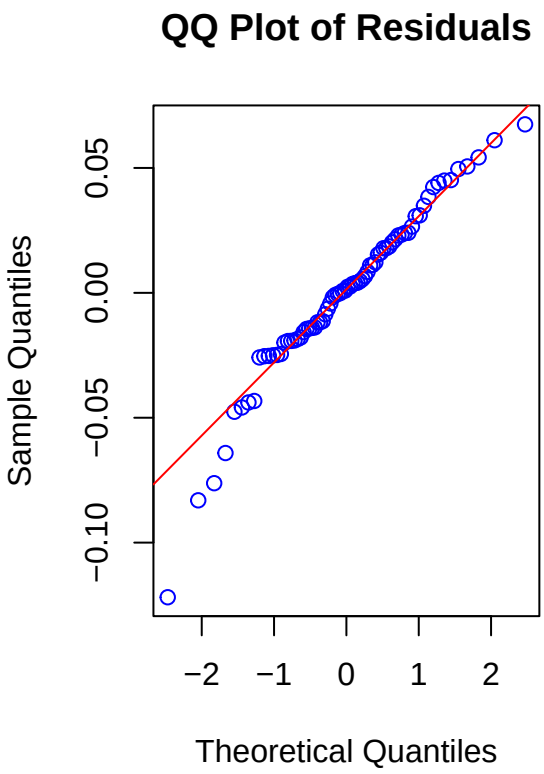
```
## [1] 0.001106908
```



Then we plot the residuals Histogram and QQ plot. The histogram shows an approximately normally distribution centered around zero. However, the QQ plot show some deviations outside the reference line.

```
## [1] 4.77049e-17
```





Drop variables according to their significance

After the first iteration we consider the process of eliminating variables based on its significance to the model. The first variable that we consider to discard is the Energy Price Index, for this we proceed with an ANOVA F-test for the comparison of regression models. As the obtained value is 0.1328 is greater than 0.05 the Energy Price Index is not a significant predictor in the presence of the other variables. With this we remove it

```
##
## Call:
## lm(formula = Home_Price_Index_log ~ ., data = data_final)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121869 -0.018344  0.001657  0.021147  0.067455
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.064351   0.544682   1.954   0.0551 .
## Industrial_Inputs_Index_log    -0.030898   0.033726  -0.916   0.3631
## Energy_Price_Index_log          0.045513   0.029890   1.523   0.1328
## Shipping_Price_Index_log        0.074346   0.078451   0.948   0.3469
## Forex_Index_log                -0.068880   0.046670  -1.476   0.1450
## Industrial_Production_Index_log 0.658104   0.085562   7.692 1.24e-10 ***
```

```
## Construction_Licences_Area_log  -0.000204   0.017490  -0.012   0.9907
## Finished_Constructions_log      -0.030569   0.030628  -0.998   0.3221
## Google_Trends_Housing_log        0.400933   0.034058  11.772  < 2e-16 ***
## Unemployment_Rate                0.366194   0.387079   0.946   0.3477
## Interest_Rate                   -0.463352   0.226089  -2.049   0.0446 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03606 on 63 degrees of freedom
## Multiple R-squared:  0.9837, Adjusted R-squared:  0.9812
## F-statistic: 381.3 on 10 and 63 DF,  p-value: < 2.2e-16

## Analysis of Variance Table
##
## Model 1: Home_Price_Index_log ~ Industrial_Inputs_Index_log + Shipping_Price_Index_log +
##     Forex_Index_log + Industrial_Production_Index_log + Construction_Licences_Area_log +
##     Finished_Constructions_log + Google_Trends_Housing_log +
##     Unemployment_Rate + Interest_Rate
## Model 2: Home_Price_Index_log ~ Industrial_Inputs_Index_log + Energy_Price_Index_log +
##     Shipping_Price_Index_log + Forex_Index_log + Industrial_Production_Index_log +
##     Construction_Licences_Area_log + Finished_Constructions_log +
##     Google_Trends_Housing_log + Unemployment_Rate + Interest_Rate
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1     64 0.084926
## 2     63 0.081911  1 0.0030146 2.3186 0.1328
```

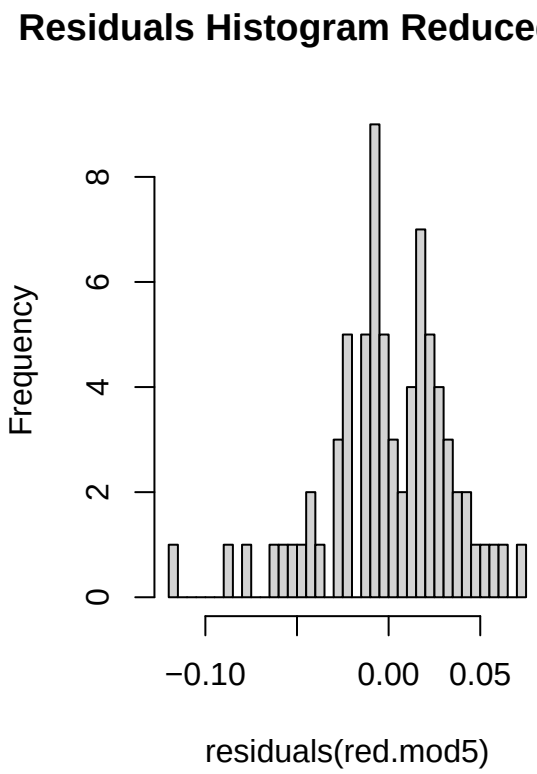
We observe again the significance of the variables, and the next no significant variables that we decide to drop were Industrial Inputs Index, Finished Constructions, Construction Licence Area and Unemployment. All of this after eliminating step by step each one following the previous procedure with the ANOVA F-test until all the predictors are significative.

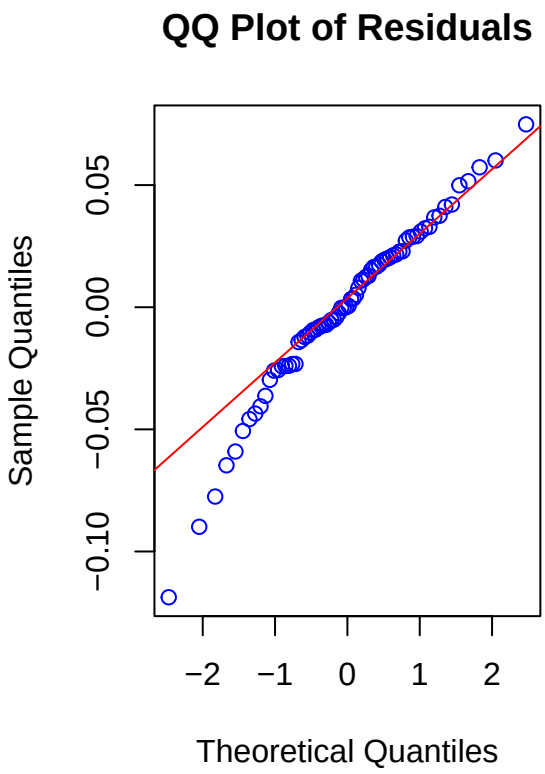
```r
#Process of Variable dropping according to ther significance

# Energy does not improve model
summary(red.mod)
red.mod2 <- update(red.mod, . ~ . -Industrial_Inputs_Index_log)
anova(red.mod, red.mod2)
# Can safely remove industrial inputs log
summary(red.mod2)

# Remove finished_constructions_log (only price variable so far)
red.mod3 <- update(red.mod2, . ~ . -Finished_Constructions_log)
anova(red.mod2, red.mod3)
# Can safely remove finished constructions (but makes no sense)
summary(red.mod3)
# All predictors are statistically significant

red.mod4 <- update(red.mod3, . ~ . -Construction_Licences_Area_log)
anova(red.mod3, red.mod4)
# Can safely remove constructions_licences_area_log
summary(red.mod4)

red.mod5 <- update(red.mod4, . ~ . -Unemployment_Rate)
anova(red.mod4, red.mod5)
```

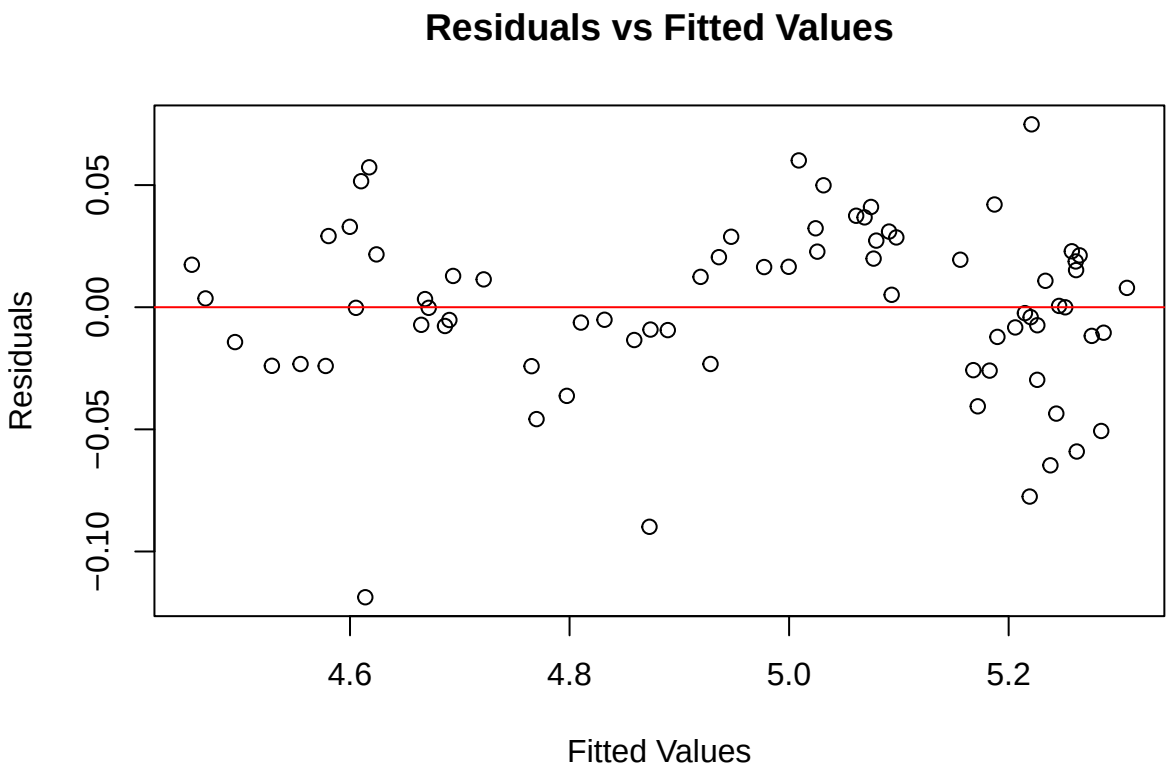
```r
# Can safely remove finished constructions (but makes no sense)
summary(red.mod5)

# All predictors are statistically significant
```

```
##
## Call:
## lm(formula = Home_Price_Index_log ~ Shipping_Price_Index_log +
##     Forex_Index_log + Industrial_Production_Index_log + Google_Trends_Housing_log +
##     Interest_Rate, data = data_final)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.118730 -0.014063  0.000251  0.021531  0.074866
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.62553    0.18580   3.367 0.001255 **
## Shipping_Price_Index_log        0.17067    0.04385   3.892 0.000229 ***
## Forex_Index_log                -0.08127    0.02793  -2.910 0.004887 **
## Industrial_Production_Index_log  0.59242    0.05484  10.803  < 2e-16 ***
## Google_Trends_Housing_log       0.39168    0.02860  13.695  < 2e-16 ***
## Interest_Rate                  -0.61174    0.19211  -3.184 0.002190 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0358 on 68 degrees of freedom
## Multiple R-squared:  0.9827, Adjusted R-squared:  0.9814
## F-statistic: 772.6 on 5 and 68 DF,  p-value: < 2.2e-16
```

The final reduced model continue to show a high R-squared, while the QQ-Plot shows more points outside the reference line. At this point even the fact that now all the variables are significative the model appears to has less explanatory power as the histogram of the residuals show a more skewed behavior even though it is centered in zero.


Residuals Histogram Reduced Mo
Frequency
0
2
4
6
8
−0.10
0.00
0.05
residuals(red.mod5)


QQ Plot of Residuals
Sample Quantiles
−0.10
−0.05
0.00
0.05
−2
−1
0
1
2
Theoretical Quantiles



Correction of the model with lagged value

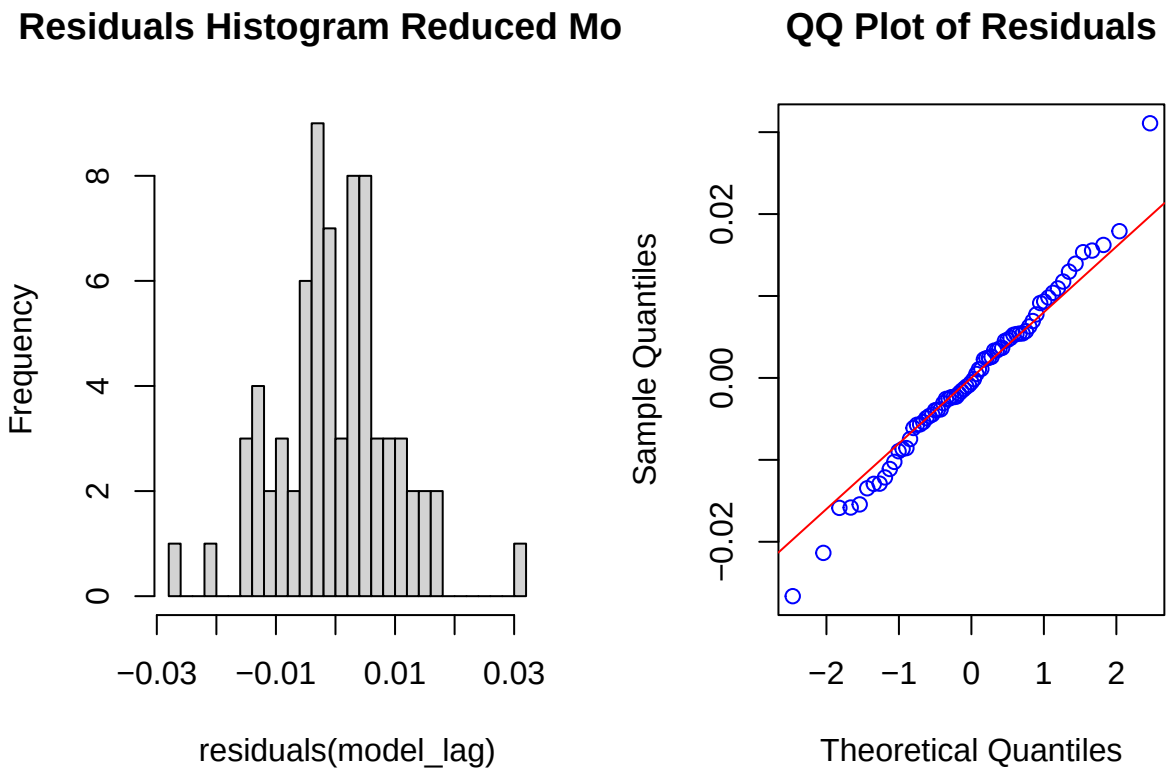
As we continue tosee some variability in the residuals, we decided to use a lagged version of the dependent variable as an explanatory variable. This can help to account for the lagged effects between the variables.

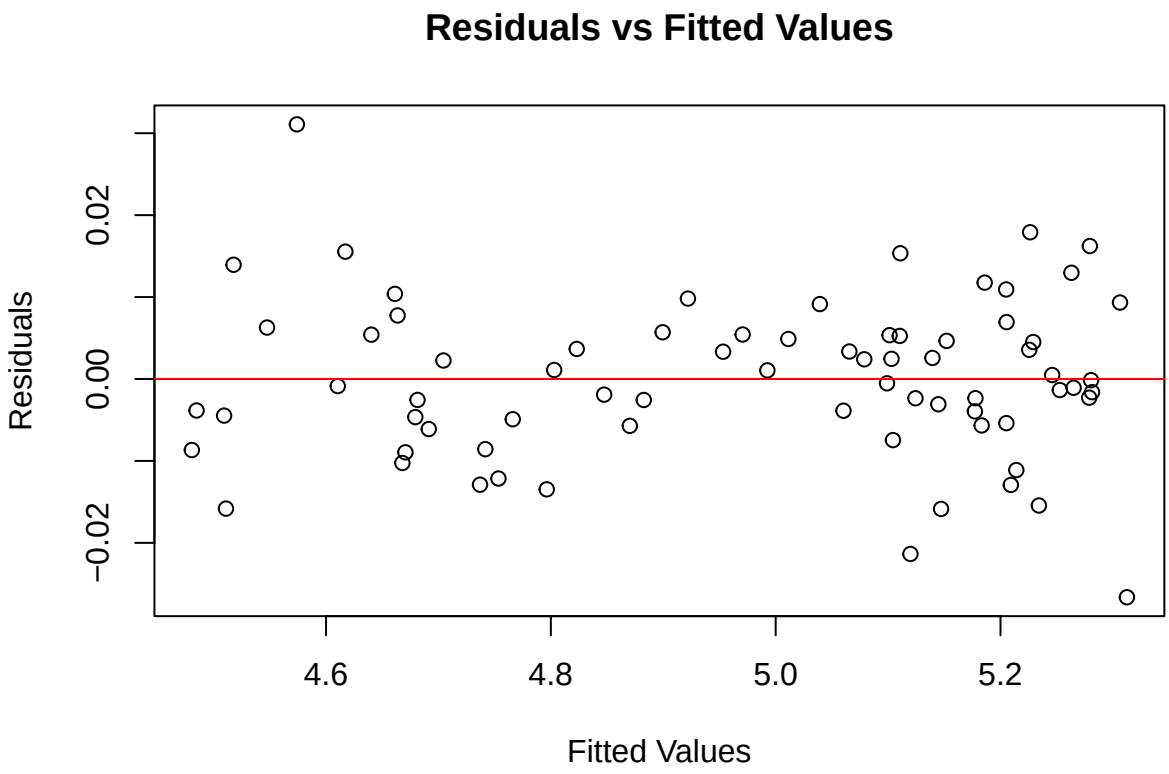
```
# Using a a lag version of the Home Price Index

data_final$Home_Price_Index_log_lag1 <- c(NA, head(data_final$Home_Price_Index_log, -1))

# Remove rows with NA values
data_final_with_lag <- na.omit(data_final)

# Fit the linear regression model
model_lag <- lm(Home_Price_Index_log ~ Home_Price_Index_log_lag1 + Forex_Index_log +
                Industrial_Production_Index_log +
                Google_Trends_Housing_log +
                Interest_Rate, data = data_final_with_lag)

# Summarize the model
summary(model_lag)
```

```
##
## Call:
## lm(formula = Home_Price_Index_log ~ Home_Price_Index_log_lag1 +
```

```
##     Forex_Index_log + Industrial_Production_Index_log + Google_Trends_Housing_log +
##     Interest_Rate, data = data_final_with_lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0266383 -0.0053892 -0.0005291  0.0054113  0.0310817
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      0.207420   0.054453   3.809 0.000305 ***
## Home_Price_Index_log_lag1        0.865665   0.027755  31.189  < 2e-16 ***
## Forex_Index_log                 -0.021747   0.007014  -3.100 0.002826 **
## Industrial_Production_Index_log  0.116086   0.023331   4.976  4.8e-06 ***
## Google_Trends_Housing_log        0.035754   0.013045   2.741 0.007850 **
## Interest_Rate                   -0.226246   0.054481  -4.153  9.5e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0101 on 67 degrees of freedom
## Multiple R-squared:  0.9986, Adjusted R-squared:  0.9985
## F-statistic:  9395 on 5 and 67 DF,  p-value: < 2.2e-16
```

After include the lagged effect we consider the reduced model showing a high R-squared. Then we proceed to show the histogram of the residuals and their QQ plot. From the histogram it is possible to see that now the residuals show a more





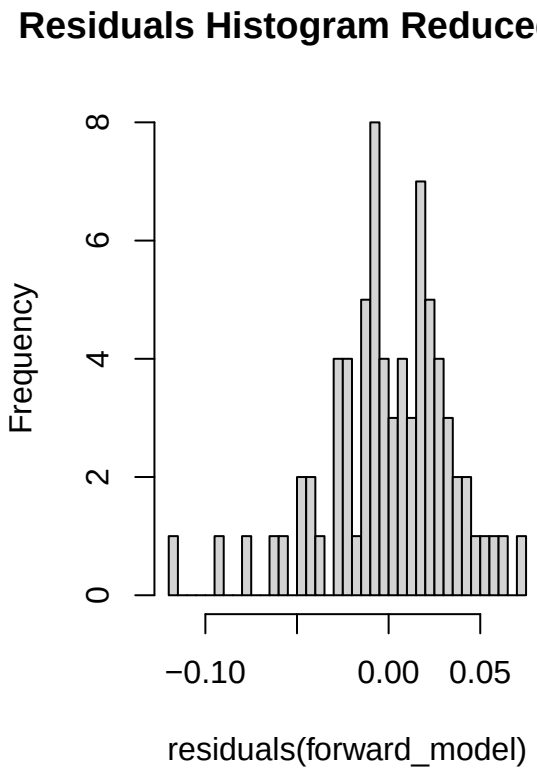
This process consider the model with robust std errors, we could think in leave it or not as we did not see it on class.

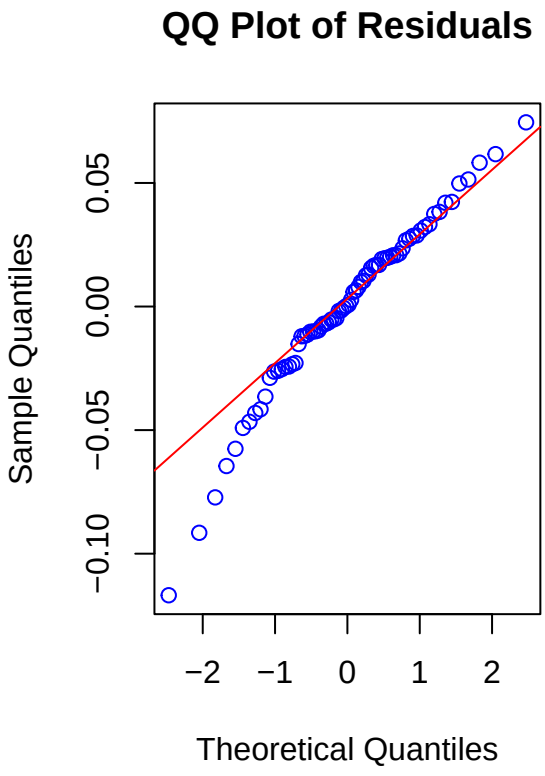
Forward AIC Selection

We Implement the forward selection based on the AIC obtained by the scores. The AIC approach allows a model comparison with the intention to find a good balance between the model fit and the complexity. The process of forward selection stars with no predictors and adding one by one. In thes case we have a no significative predictor the Industrial Inputs Index, this happens because with the AIC criterion is possible to obtain non-significant predictors if their inclusion improves the general performance of the model.

```
# Forward AIC Stepwise selection-------------

# Fit the null model (intercept only)
null_model <- lm(Home_Price_Index_log ~ 1, data = data_reduced)

# Specify the full model with all potential predictors
full_model <- lm(Home_Price_Index_log ~ Industrial_Inputs_Index_log + Shipping_Price_Index_log + Forex_
                   Construction_Licences_Area_log + Google_Trends_Housing_log +
                   Unemployment_Rate + Interest_Rate, data = data_reduced)

# Perform forward selection based on AIC
forward_model <- step(null_model,
                      scope = list(lower = null_model, upper = full_model),
                      direction = "forward")
```

```
##
## Call:
## lm(formula = Home_Price_Index_log ~ Google_Trends_Housing_log +
##     Industrial_Production_Index_log + Industrial_Inputs_Index_log +
##     Interest_Rate + Shipping_Price_Index_log + Forex_Index_log,
##     data = data_reduced)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.116812 -0.014398  0.000464  0.020767  0.074518
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.609109   0.194960   3.124  0.00263 **
## Google_Trends_Housing_log       0.392540   0.028937  13.565  < 2e-16 ***
## Industrial_Production_Index_log 0.590355   0.055640  10.610 5.51e-16 ***
## Industrial_Inputs_Index_log     0.007937   0.026551   0.299  0.76593
## Interest_Rate                  -0.593246   0.203071  -2.921  0.00475 **
## Shipping_Price_Index_log        0.160363   0.056019   2.863  0.00560 **
## Forex_Index_log                -0.076066   0.033066  -2.300  0.02454 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03605 on 67 degrees of freedom
## Multiple R-squared:  0.9827, Adjusted R-squared:  0.9812
## F-statistic: 635.3 on 6 and 67 DF,  p-value: < 2.2e-16
```





Backward AIC selection

We implement the backward selection based also based in the AIC obtained by the different models. In this case, contrary to forward selection, all the variables left were in the reduced model obtained by leaving only the most significative variables. so for this case the results are the same obtained before.

```
# Backward AIC Stepwise selection-------------

full_model <- lm(Home_Price_Index_log ~ Industrial_Inputs_Index_log + Shipping_Price_Index_log + Forex_
                   Construction_Licences_Area_log + Google_Trends_Housing_log +
                   Unemployment_Rate + Interest_Rate, data = data_reduced)

# Perform backward selection based on AIC
backward_model <- step(full_model,
                       direction = "backward")
```

```
##
## Call:
## lm(formula = Home_Price_Index_log ~ Shipping_Price_Index_log +
##     Forex_Index_log + Industrial_Production_Index_log + Google_Trends_Housing_log +
##     Interest_Rate, data = data_reduced)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.118730 -0.014063  0.000251  0.021531  0.074866
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      0.62553    0.18580   3.367 0.001255 **
## Shipping_Price_Index_log         0.17067    0.04385   3.892 0.000229 ***
## Forex_Index_log                 -0.08127    0.02793  -2.910 0.004887 **
## Industrial_Production_Index_log  0.59242    0.05484  10.803  < 2e-16 ***
## Google_Trends_Housing_log        0.39168    0.02860  13.695  < 2e-16 ***
## Interest_Rate                   -0.61174    0.19211  -3.184 0.002190 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0358 on 68 degrees of freedom
## Multiple R-squared:  0.9827, Adjusted R-squared:  0.9814
## F-statistic: 772.6 on 5 and 68 DF,  p-value: < 2.2e-16
```

Lasso regresion model

```
# Regularization------------
# Done over full model
# Load the packages
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
library(caret)
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
```

```
## The following objects are masked from 'package:caret':
##
##     precision, recall
```

```r
# Subset the dataframe by selecting specific columns
data_final <- subset(data_reduced, select = c(Home_Price_Index_log, Industrial_Inputs_Index_log, Metals_
                                              Energy_Price_Index_log, Shipping_Price_Index_log, Forex_In
                                              Industrial_Production_Index_log, Construction_Licences_Ar
                                              Finished_Constructions, Google_Trends_Housing_log,
                                              Unemployment_Rate, Interest_Rate))

# Sample data preparation
# Assuming your data frame is `data_final` and your response variable is `y_log`
set.seed(123)  # For reproducibility
trainIndex <- createDataPartition(data_final$Home_Price_Index_log, p = .8,
                                  list = FALSE,
                                  times = 1)
data_train <- data_final[ trainIndex,]
data_test  <- data_final[-trainIndex,]

# Extract predictors and response
X_train <- model.matrix(Home_Price_Index_log ~ ., data_train)[, -1]  # Remove the intercept column
y_train <- data_train$Home_Price_Index_log

X_test <- model.matrix(Home_Price_Index_log ~ ., data_test)[, -1]  # Remove the intercept column
y_test <- data_test$Home_Price_Index_log

# Fit linear regression model
lm_model <- lm(Home_Price_Index_log ~ ., data = data_train)

# Predict and evaluate linear regression model
lm_predictions <- predict(lm_model, newdata = data_test)
lm_mse <- mse(y_test, lm_predictions)
lm_r2 <- R2(y_test, lm_predictions)
cat("Linear Regression MSE:", lm_mse, "\n")
```

```
## Linear Regression MSE: 0.0008267574
```

```r
cat("Linear Regression R2:", lm_r2, "\n")
```

```
## Linear Regression R2: 0.9927676
```

```r
# Fit LASSO regression model
lasso_model <- cv.glmnet(X_train, y_train, alpha = 1)  # alpha = 1 for LASSO
lasso_best_lambda <- lasso_model$lambda.min
```

```r
lasso_predictions <- predict(lasso_model, s = lasso_best_lambda, newx = X_test)
lasso_mse <- mse(y_test, lasso_predictions)
lasso_r2 <- R2(y_test, lasso_predictions)

cat("LASSO Regression MSE:", lasso_mse, "\n")
```

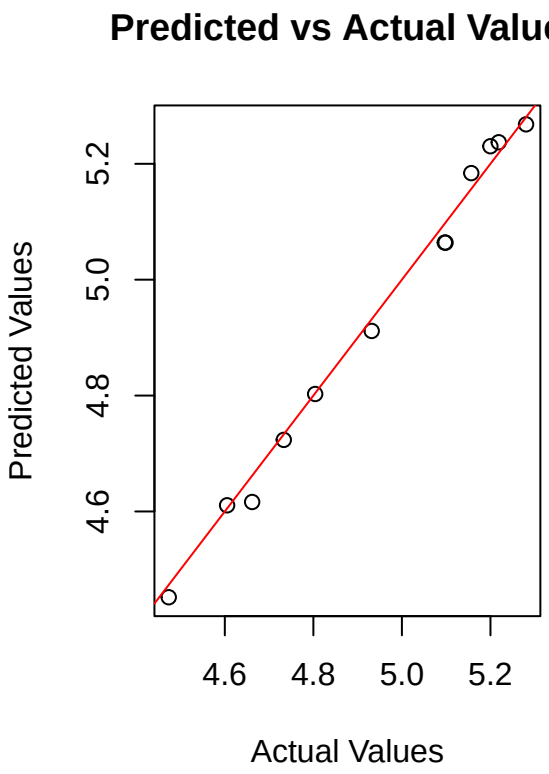
```
## LASSO Regression MSE: 0.000624037
```

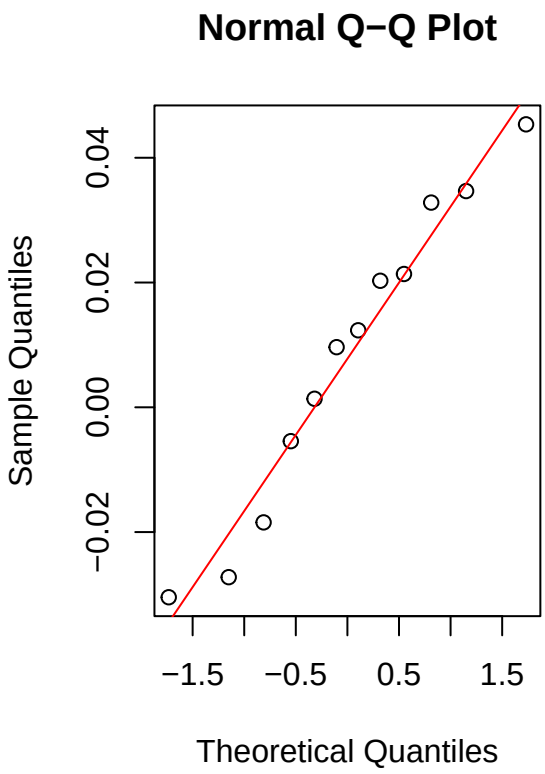
```r
cat("LASSO Regression R2:", lasso_r2, "\n")
```

```
## LASSO Regression R2: 0.9934312
```

```r
# Coefficients at best lambda
lasso_coefficients <- coef(lasso_model, s = lasso_best_lambda)
print(lasso_coefficients)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                          s1
## (Intercept)                    4.806159e-01
## Industrial_Inputs_Index_log    .
## Metals_Price_Index_log         8.738383e-03
## Energy_Price_Index_log         3.889603e-02
## Shipping_Price_Index_log       3.161120e-02
## Forex_Index_log                .
## Industrial_Production_Index_log 6.042338e-01
## Construction_Licences_Area_log  .
## Finished_Constructions         -5.498884e-09
## Google_Trends_Housing_log      3.921892e-01
## Unemployment_Rate              .
## Interest_Rate                  -5.326852e-01
```





Ridge regression model

```
# Fit Ridge regression model
ridge_model <- cv.glmnet(X_train, y_train, alpha = 0)  # alpha = 0 for Ridge
ridge_best_lambda <- ridge_model$lambda.min
ridge_predictions <- predict(ridge_model, s = ridge_best_lambda, newx = X_test)
ridge_mse <- mse(y_test, ridge_predictions)
ridge_r2 <- R2(y_test, ridge_predictions)
cat("Ridge Regression MSE:", ridge_mse, "\n")
```

```
## Ridge Regression MSE: 0.000872958
```

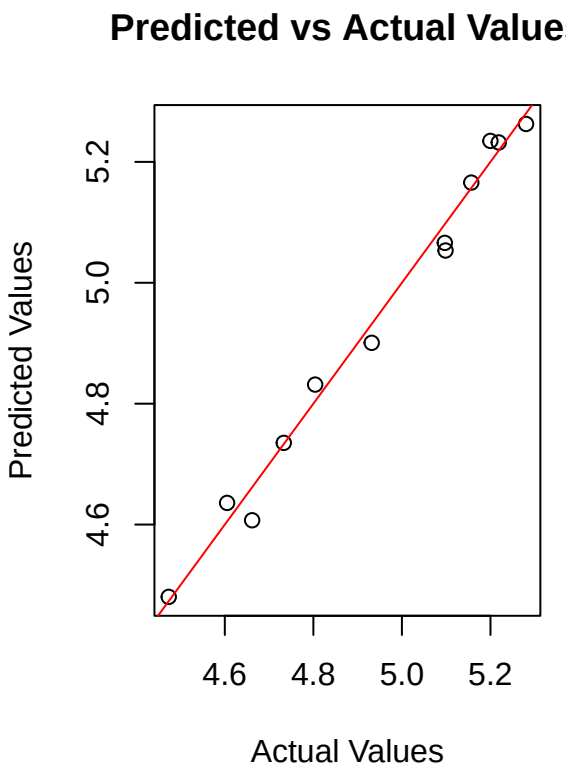
```
cat("Ridge Regression R2:", ridge_r2, "\n")
```

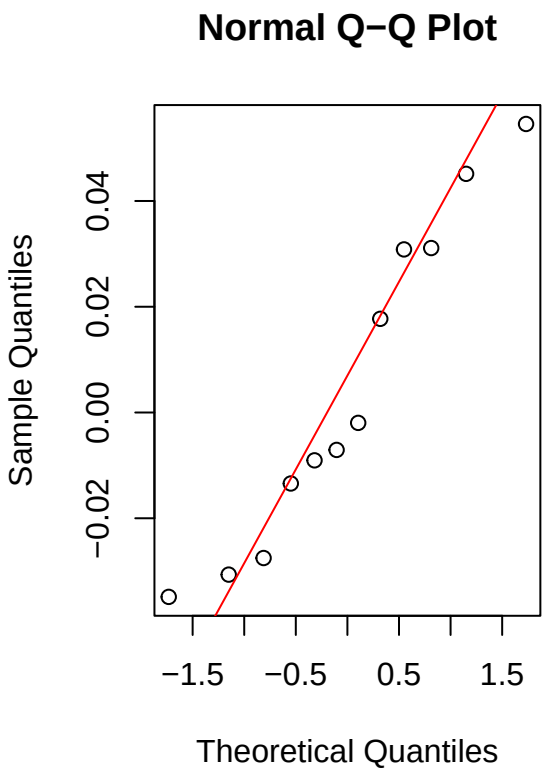
```
## Ridge Regression R2: 0.9876594
```

```
# Coefficients at best lambda
ridge_coefficients <- coef(ridge_model, s = ridge_best_lambda)
print(ridge_coefficients)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                                        s1
## (Intercept)                  1.318884e-01
## Industrial_Inputs_Index_log -1.549222e-03
## Metals_Price_Index_log       4.576378e-02
## Energy_Price_Index_log      -7.198777e-03
## Shipping_Price_Index_log     1.556973e-01
## Forex_Index_log              3.337002e-02
## Industrial_Production_Index_log  5.045049e-01
## Construction_Licences_Area_log   1.241637e-02
## Finished_Constructions       3.290680e-09
## Google_Trends_Housing_log    3.286521e-01
## Unemployment_Rate           -4.514011e-01
## Interest_Rate               -9.771575e-01
```






```r
results <- data.frame(
  Model = c("Linear Regression", "LASSO Regression", "Ridge Regression"),
  MSE = c(lm_mse, lasso_mse, ridge_mse),
  R2 = c(lm_r2, lasso_r2, ridge_r2)
)
print(results)
```

```
##               Model          MSE        R2
## 1 Linear Regression 0.0008267574 0.9927676
## 2  LASSO Regression 0.0006240370 0.9934312
## 3  Ridge Regression 0.0008729580 0.9876594
```