Outlier detection and statistics

Outlier detection and statistics

```r
##########################################################
## Question 1: VIX Regression Model
##########################################################
#install.packages("zoo")
#install.packages("xts")
#install.packages("utils")
#install.packages("tseries")
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(utils)
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```r
library(xts)
library(readr)
require(quantmod)
```

```
## Loading required package: quantmod

## Loading required package: TTR

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
require(graphics)
require(timeSeries)
```

```
## Loading required package: timeSeries

## Loading required package: timeDate

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##
##     time<-
```

```r
require(xts)
require(car)
```

```
## Loading required package: car

## Loading required package: carData
```

```
require(MLmetrics)
```

```
## Loading required package: MLmetrics

##
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':
##
##     Recall
```

Q1. Loading CSV file into "R" and using Data Frames

```
## Import data
#setwd("~/Desktop/UBS")
data <- read_csv("data.csv")
Vix_data<-data.frame(t(data[,-1]))
colnames(Vix_data)<-c("Eq_Ind1","Eq_Ind2","VIX","LIBOR")
rownames(Vix_data)<- c(colnames(data)[-1])

### Removing NA observations - results in quarterly data
Vix_data_TS_Qtrly<-na.omit(Vix_data)
```

Q2.Outlier Detection:

Outlier detection plays a crucial role in data analysis, in that outliers and influential points are distant from other points and tend to impact the model accuracy. Some points are potential outliers Outliers play a very significant part of any data analysis/cleaning steps. The presence of outliers can very much impact the final results of the regression model as they tend to pull the estimates towards themselves and hence the regression deviates from its task of fitting the best model. Some points are potential outliers and other are influential points. Influential points are those observations whose presence/absence can alter the regression result on a great scale. Usually, modellers try to identify(if any) outliers at the early stages of data cleaning and models are built in both the cases with and without these outliers to gauge the importance of these outliers and to further justify on the authenticity of the observation actually being an outlier or has a finite chance of occurence.

Simple Outlier detection can be done by the Help of Boxplots, Scatter plots, Quantile Plot, Checking the Distance from mean.

As a start, identifying observations which are 3-Std.Dev away from the mean might be a good start. It can help to assess the overall feel of the data and how the distribution looks like. Observations outside the 3-Std.Dev deviations definitely raises some concern as these are highly unusual and are very unlikely to occur.

```
attach(Vix_data_TS_Qtrly)
### STD Deviation Approach
# calculate summary statistics
find_outlier_obs<-function(var1){
  series1<-var1
  data_mean<-mean(series1)
  data_std <-sd(series1)
  # identify outliers
  cut_off = data_std * 3
  lower= data_mean - cut_off
  upper <-data_mean + cut_off
  obs_index<-which(series1<lower| series1 >upper)
```

```
  obs_value<-series1[obs_index]

  return(list(Obs_Index=obs_index,Value=obs_value))
}

VIX_outlier_obs<-find_outlier_obs(VIX)
Eq_Ind1_outlier_obs<-find_outlier_obs(Eq_Ind1)
Eq_Ind2_outlier_obs<-find_outlier_obs(Eq_Ind2)
LIBOR_outlier_obs<-find_outlier_obs(LIBOR)

## VIX - Appears that there are two Observations which are outside the 3Sigma Distance from the Mean of
VIX_outlier_obs
```

```
## $Obs_Index
## [1] 75 85
##
## $Value
## [1] 44.14 42.96
```

```
## Equity Index1 - No extreme Observations observed in this series
Eq_Ind1_outlier_obs
```

```
## $Obs_Index
## integer(0)
##
## $Value
## numeric(0)
```

```
## Equity Index2 - Appears that there is One Observation which are outside the 3Sigma Distance from the
Eq_Ind2_outlier_obs
```

```
## $Obs_Index
## [1] 109
##
## $Value
## [1] 6903.389
```

```
## Libor - No extreme Observations observed in this series
LIBOR_outlier_obs
```

```
## $Obs_Index
## integer(0)
##
## $Value
## numeric(0)
```

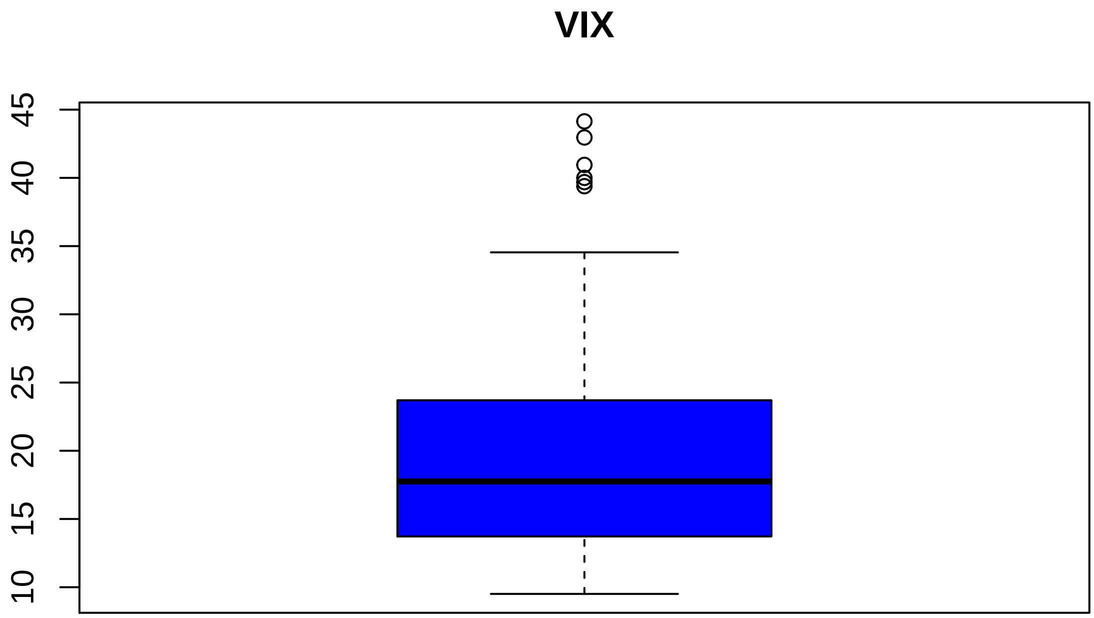
Visualization tools and diagnostic plots are essential in pre-processing of the data. Looking at plots of Boxplots, Scatterplots, and Normal-Quantile plots would help analyse the behavior of each individual series.

Boxplots help in understanding the distributions of the data, look at the range and IQR, quartiles and outliers. Extreme observations outside the whiskers of the Boxplot could be potential outliers.

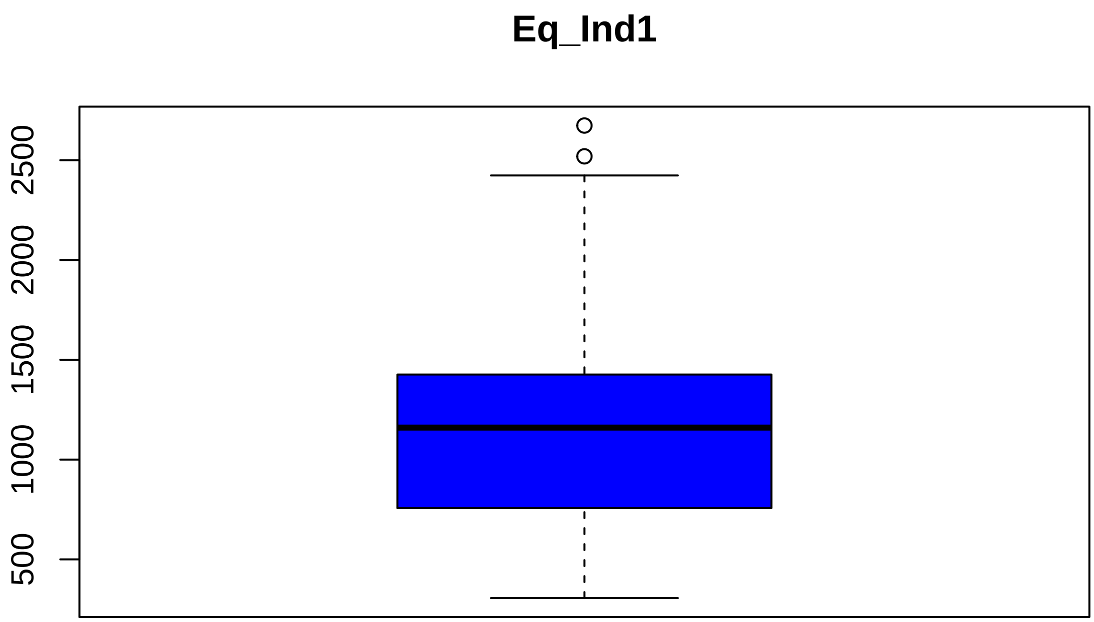
Scatterplots help in understanding the relationship between two variables or infer the trend in one or more variables. If an observation is distant from the major scatter of data then it might be a potential outlier.

Quantile-Quantile plots are basic diagnostic plots in understanding how the quantiles of the observations deviate from the theoretical normal quantiles.

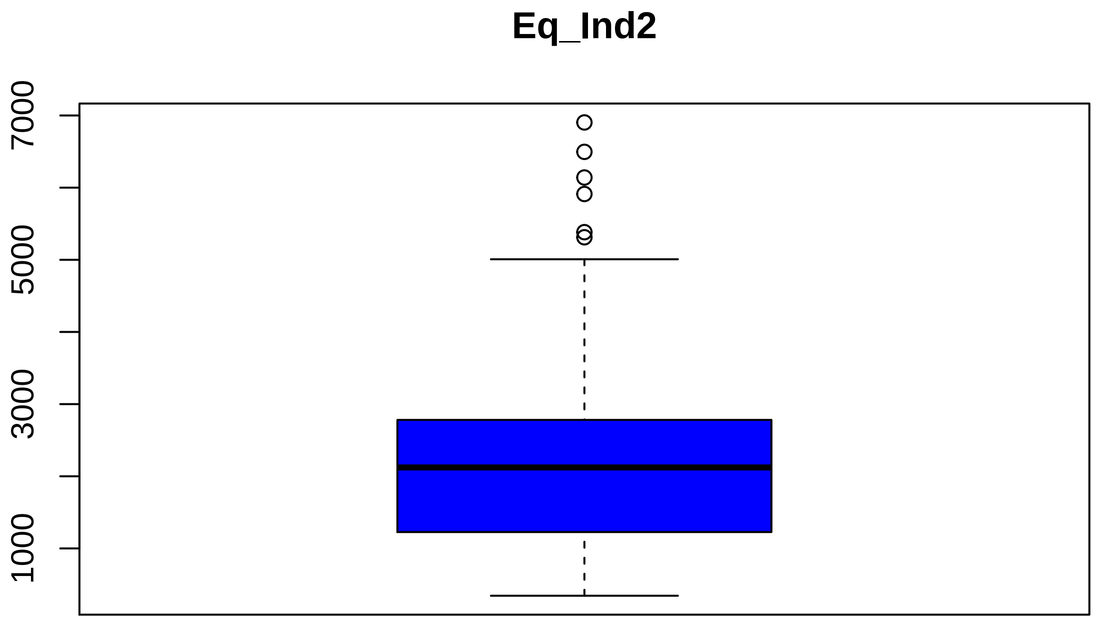
```r
attach(Vix_data_TS_Qtrly)

boxplot(VIX, col = "Blue",main="VIX")
```



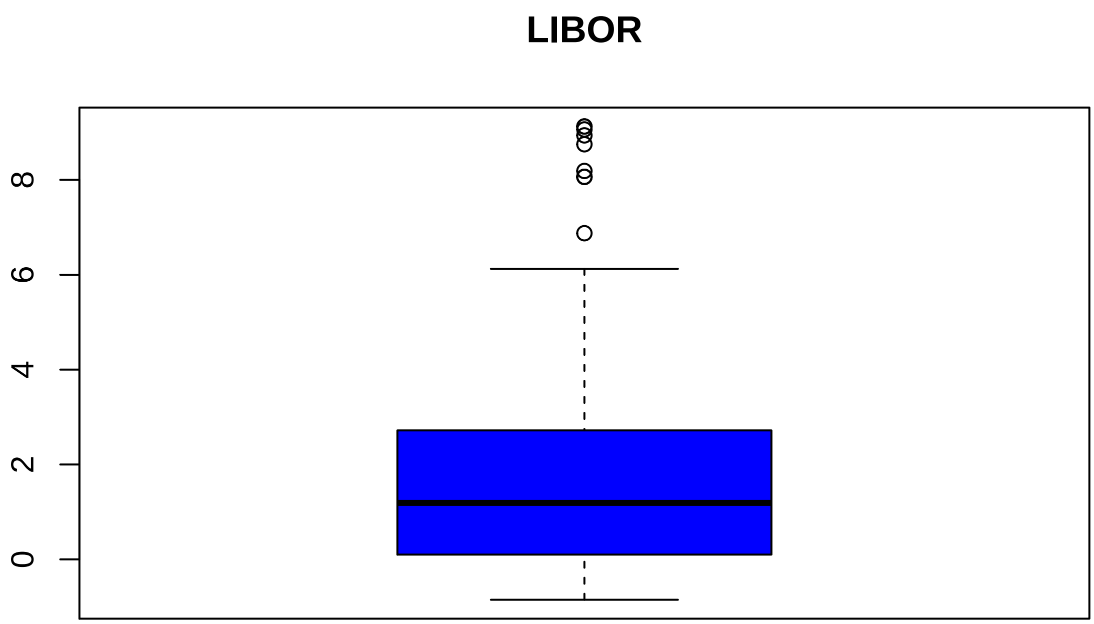

```r
boxplot(Eq_Ind1,col = "Blue",main="Eq_Ind1")
```



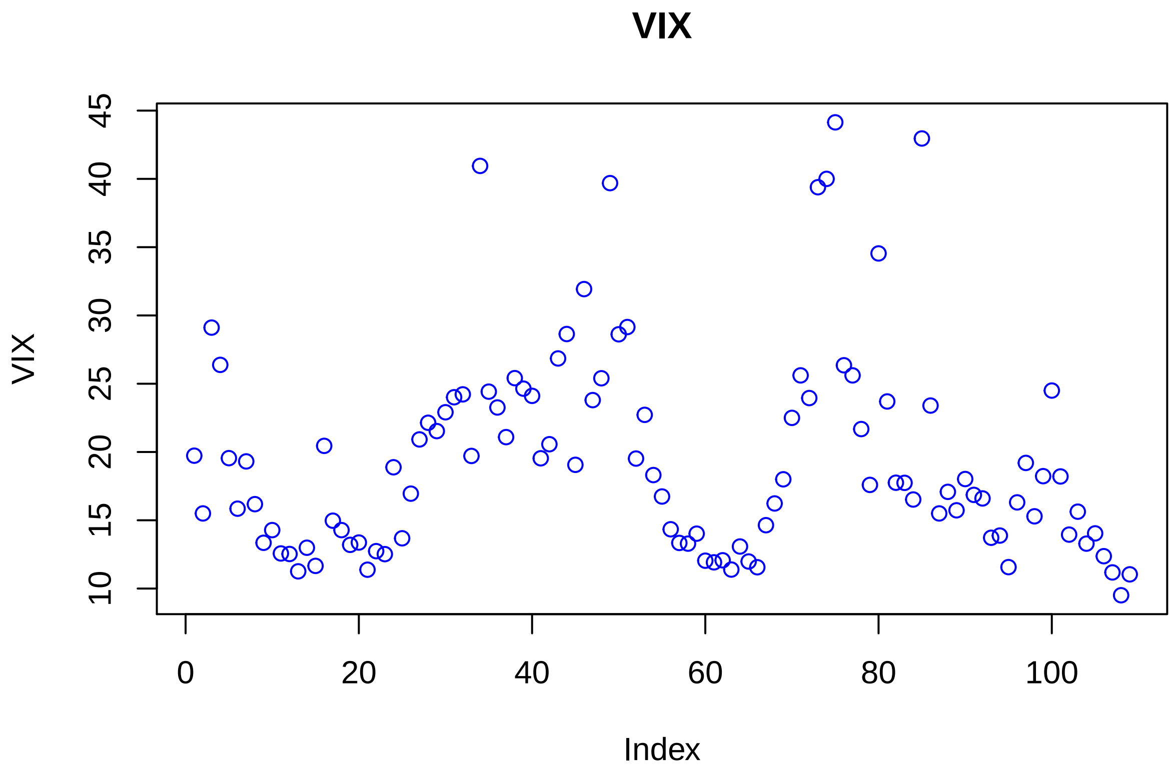

```r
boxplot(Eq_Ind2,col = "Blue",main="Eq_Ind2")
```



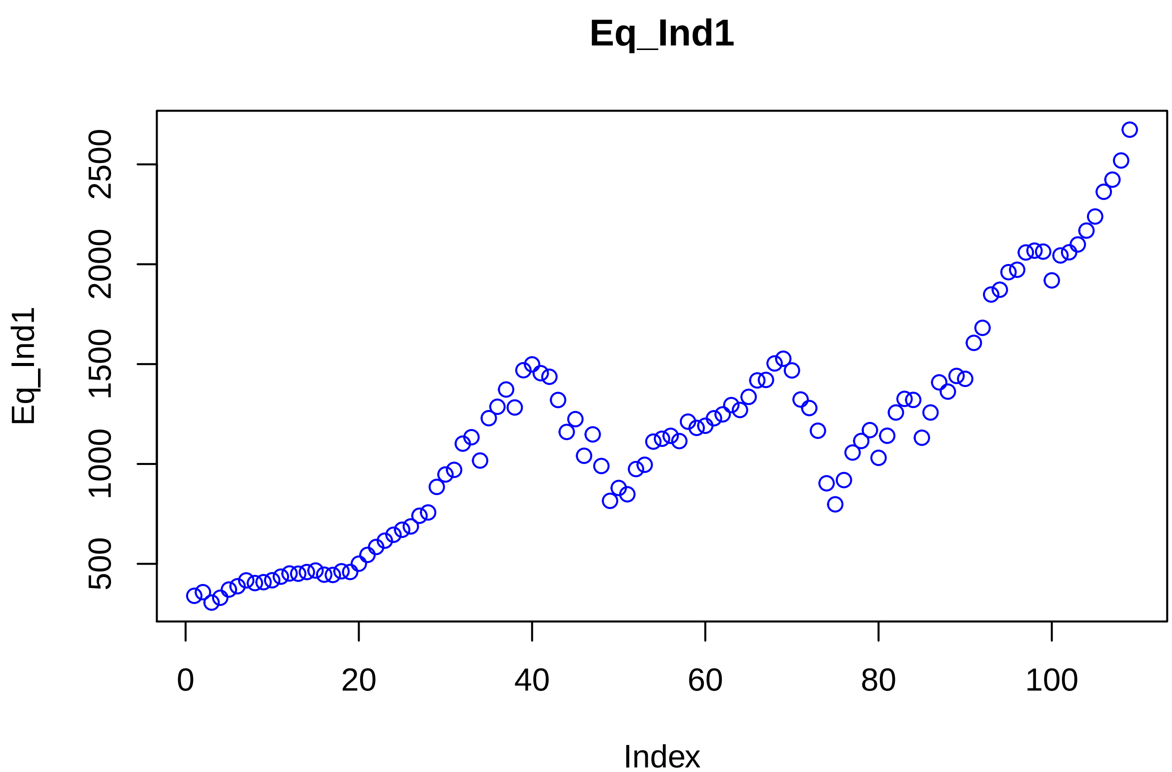

```r
boxplot(LIBOR,col = "Blue",main="LIBOR")
```



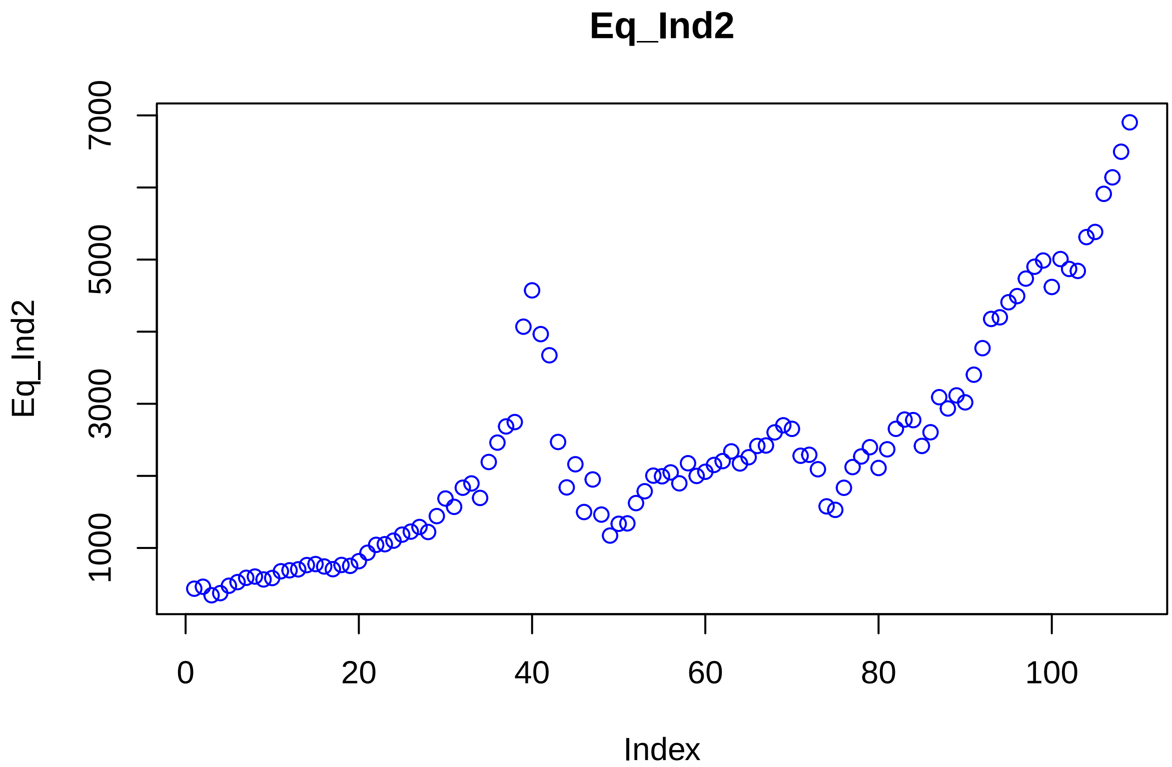

```r
## Scatterplot
plot(VIX, col = "Blue",main="VIX")
```



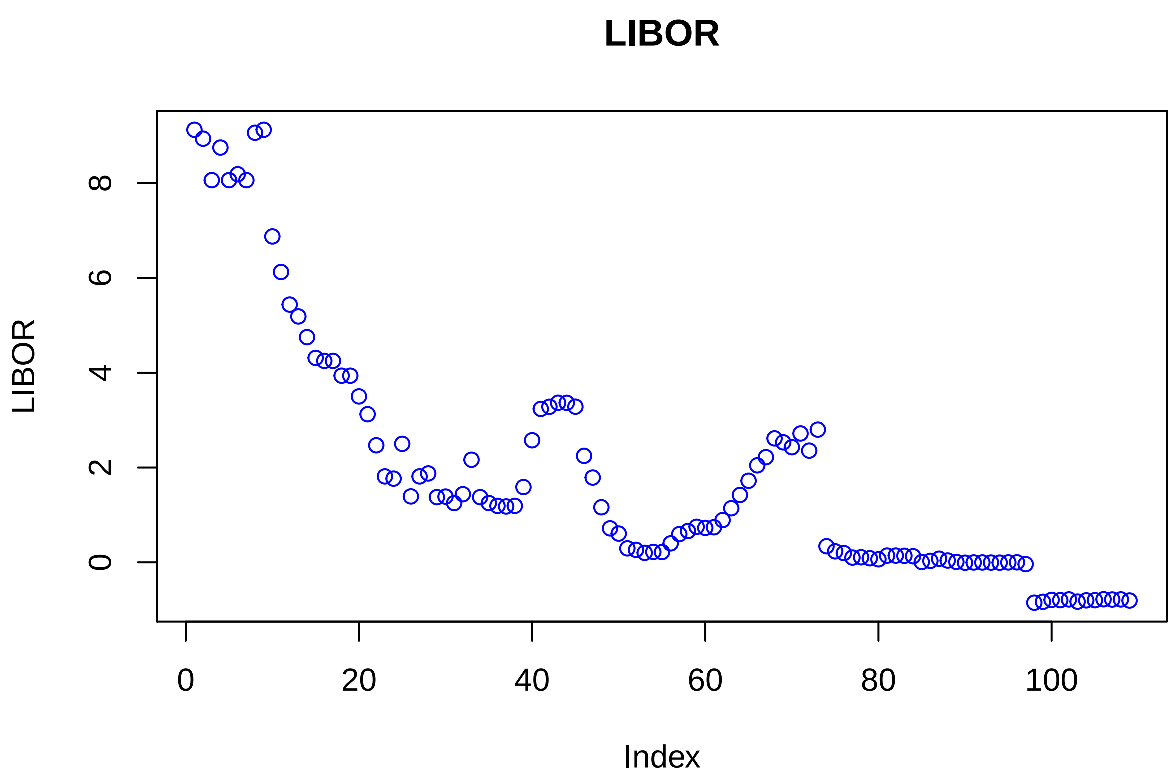

```r
plot(Eq_Ind1,col = "Blue",main="Eq_Ind1")
```



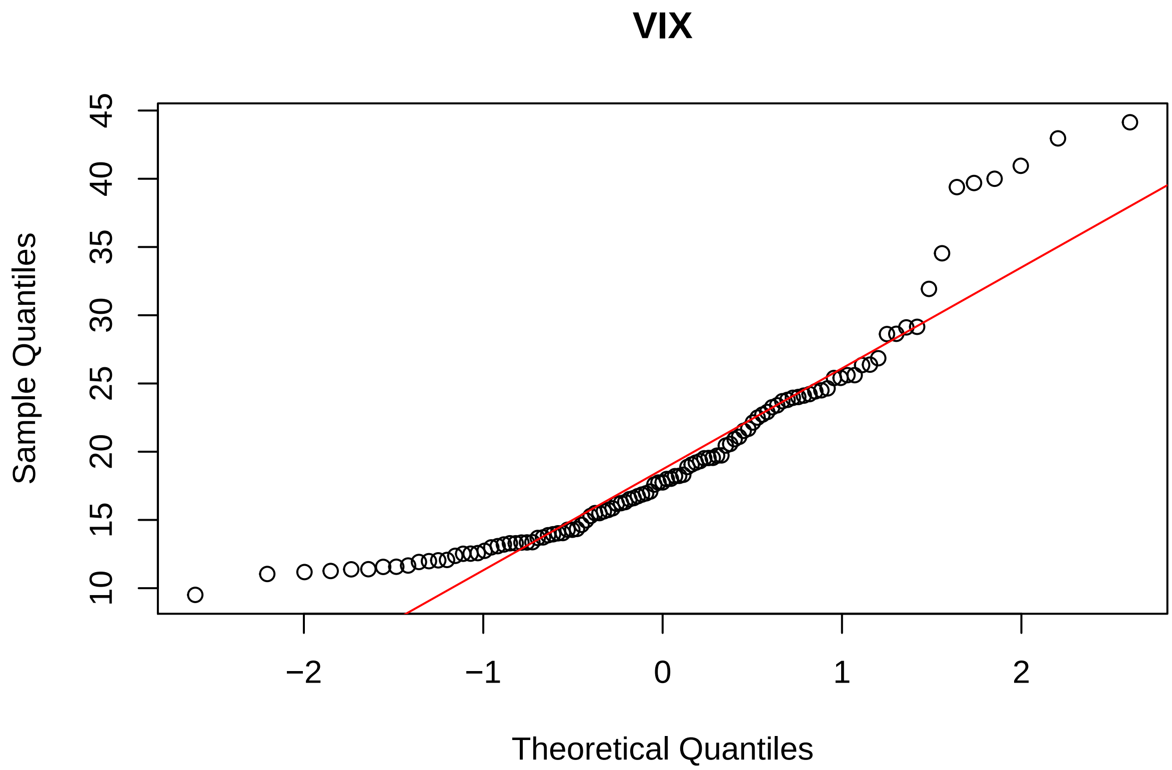

```r
plot(Eq_Ind2,col = "Blue",main="Eq_Ind2")
```



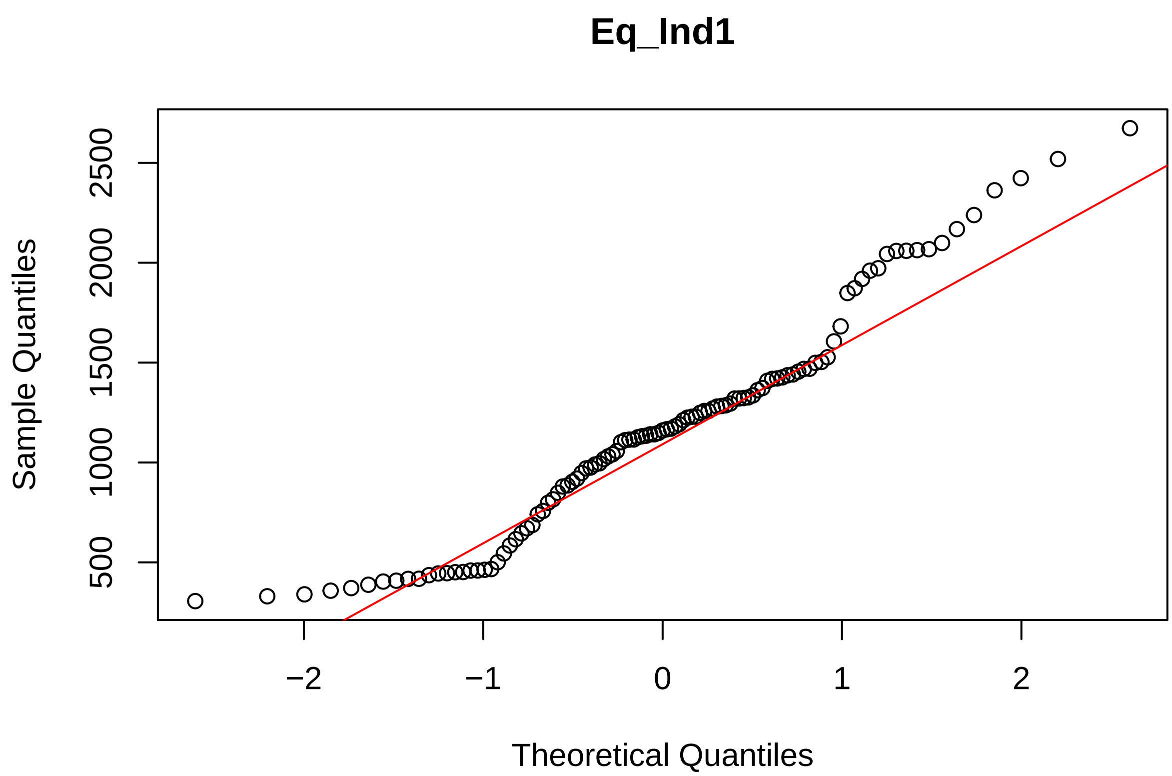

```r
plot(LIBOR,col = "Blue",main="LIBOR")
```



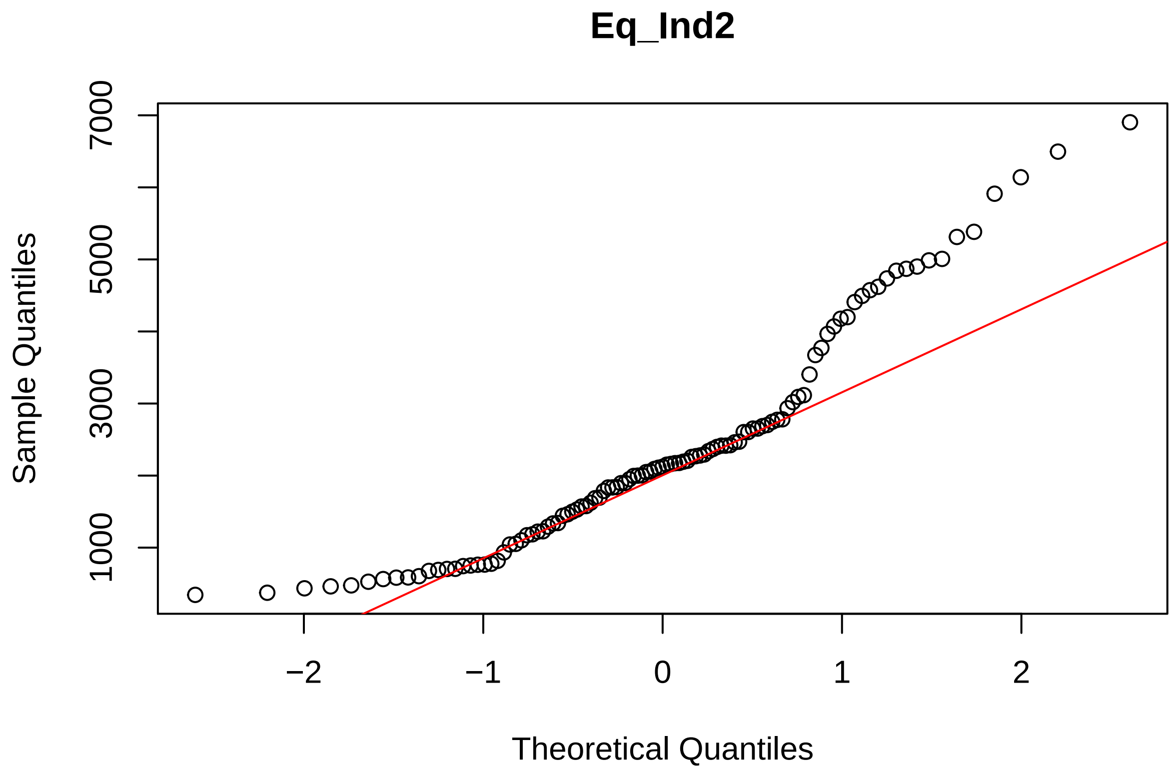

```r
## QQ plots
qqnorm(VIX,main = "VIX");qqline(VIX,col=2)
```



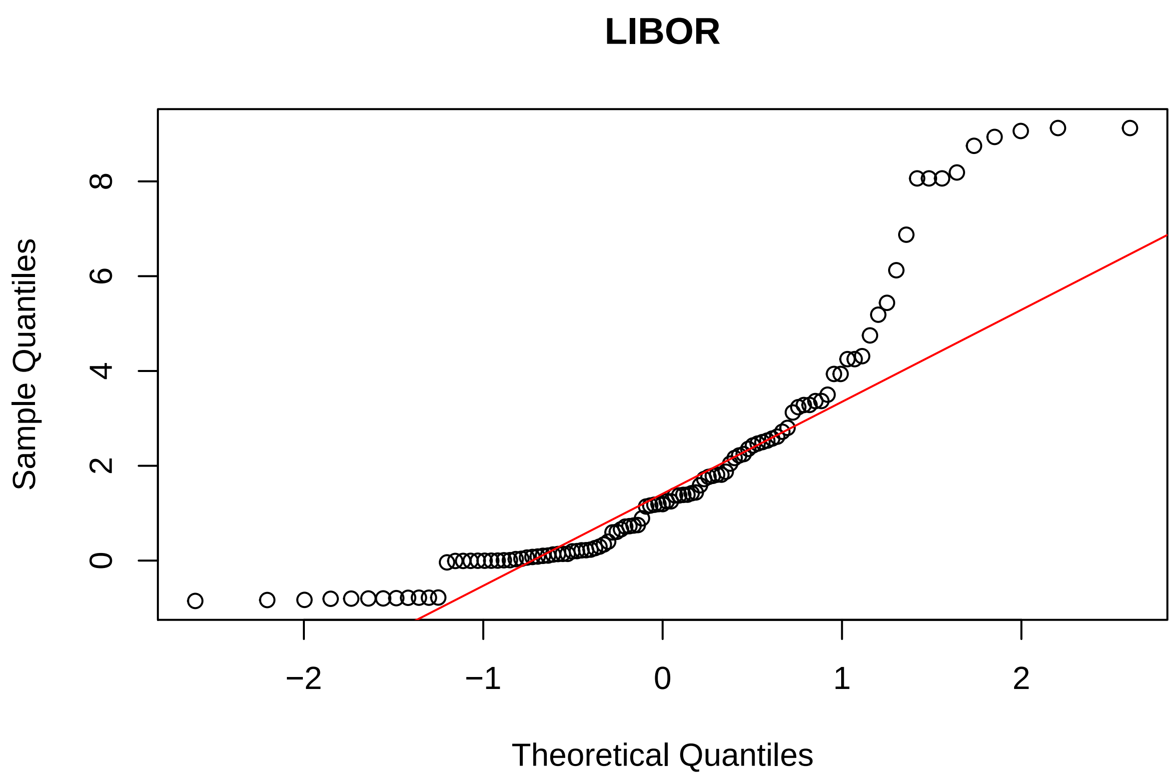

```
qqnorm(Eq_Ind1,main = "Eq_Ind1");qqline(Eq_Ind1,col=2)
```




```
qqnorm(Eq_Ind2,main = "Eq_Ind2");qqline(Eq_Ind2,col=2)
```

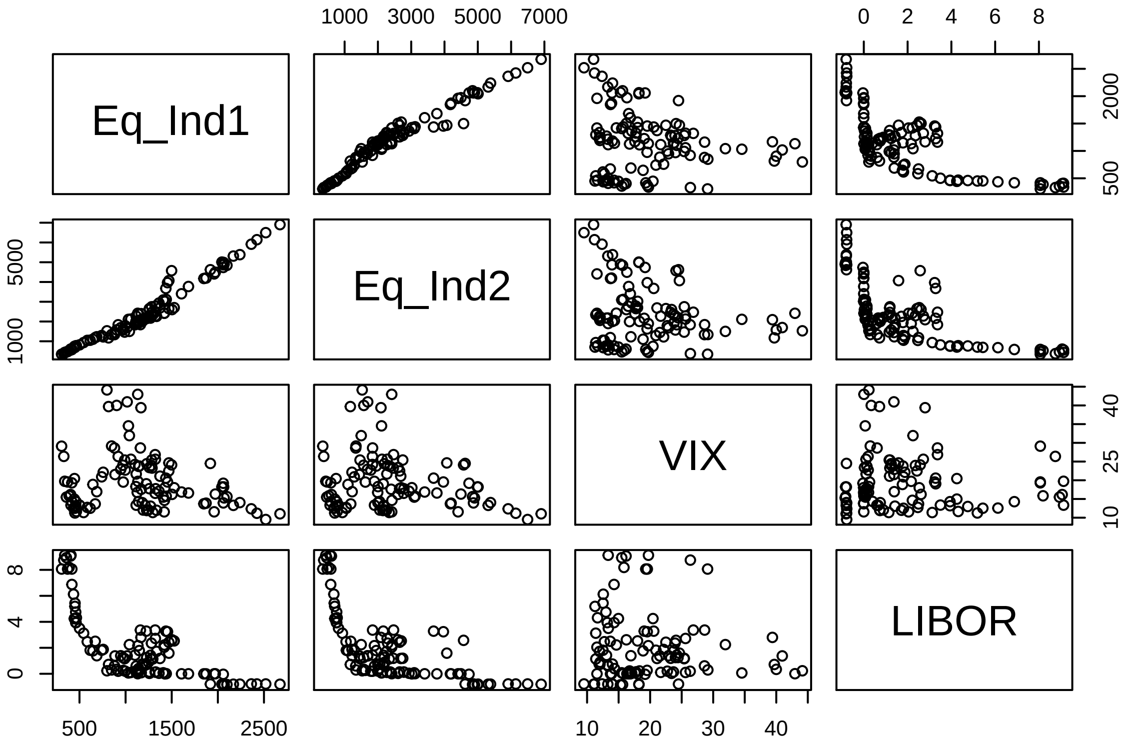



```
qqnorm(LIBOR,main = "LIBOR");qqline(LIBOR,col=2)
```



It can be inferred from the plots above that there are outliers in the box plots which can be looked into later during model selection and diagnostics. QQplots suggests that the data suffers from extreme tail observations.

```
## scatterplot to check for relationship between variables.

plot(Vix_data_TS_Qtrly)
```



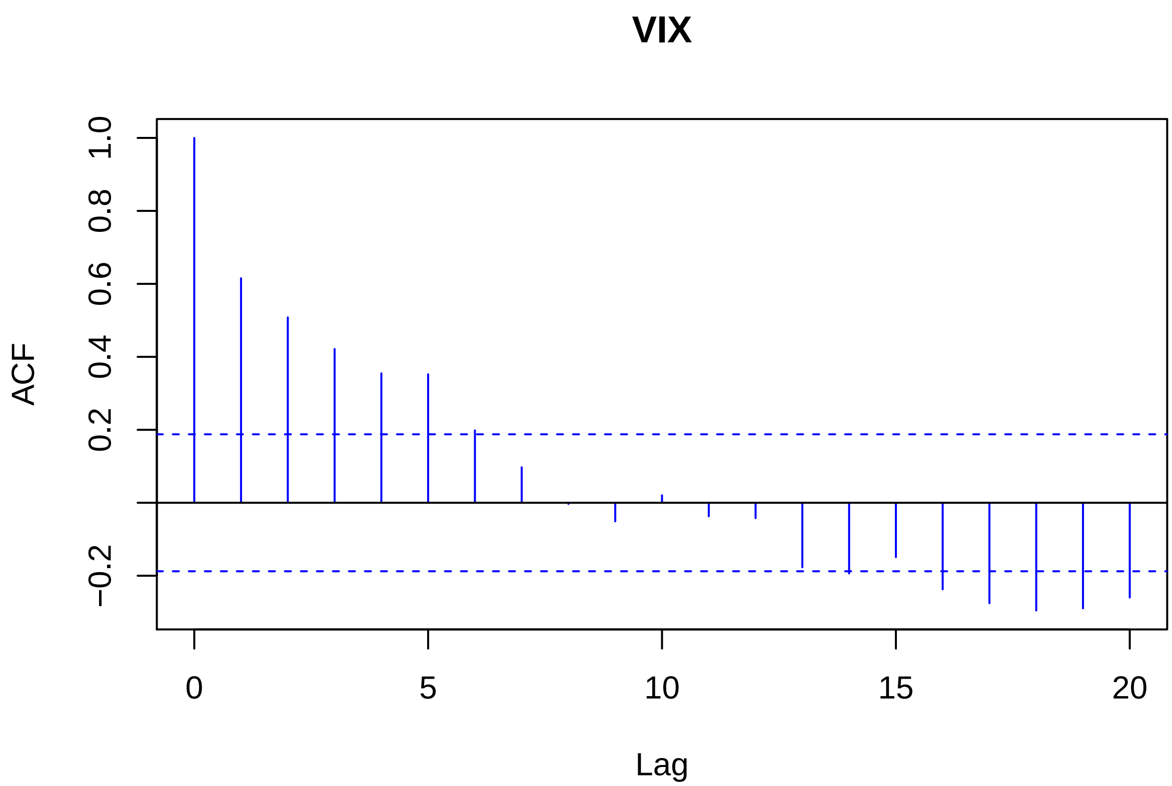

```
## Scatterplot shows that Vix has a decreasing relationship with Equity indexes and shows come correlat

cor(Vix_data_TS_Qtrly)
```

```
##             Eq_Ind1    Eq_Ind2         VIX       LIBOR
## Eq_Ind1  1.0000000  0.9744306 -0.16478200 -0.72190386
## Eq_Ind2  0.9744306  1.0000000 -0.18850687 -0.66168760
## VIX     -0.1647820 -0.1885069  1.00000000 -0.05957299
## LIBOR   -0.7219039 -0.6616876 -0.05957299  1.00000000
```

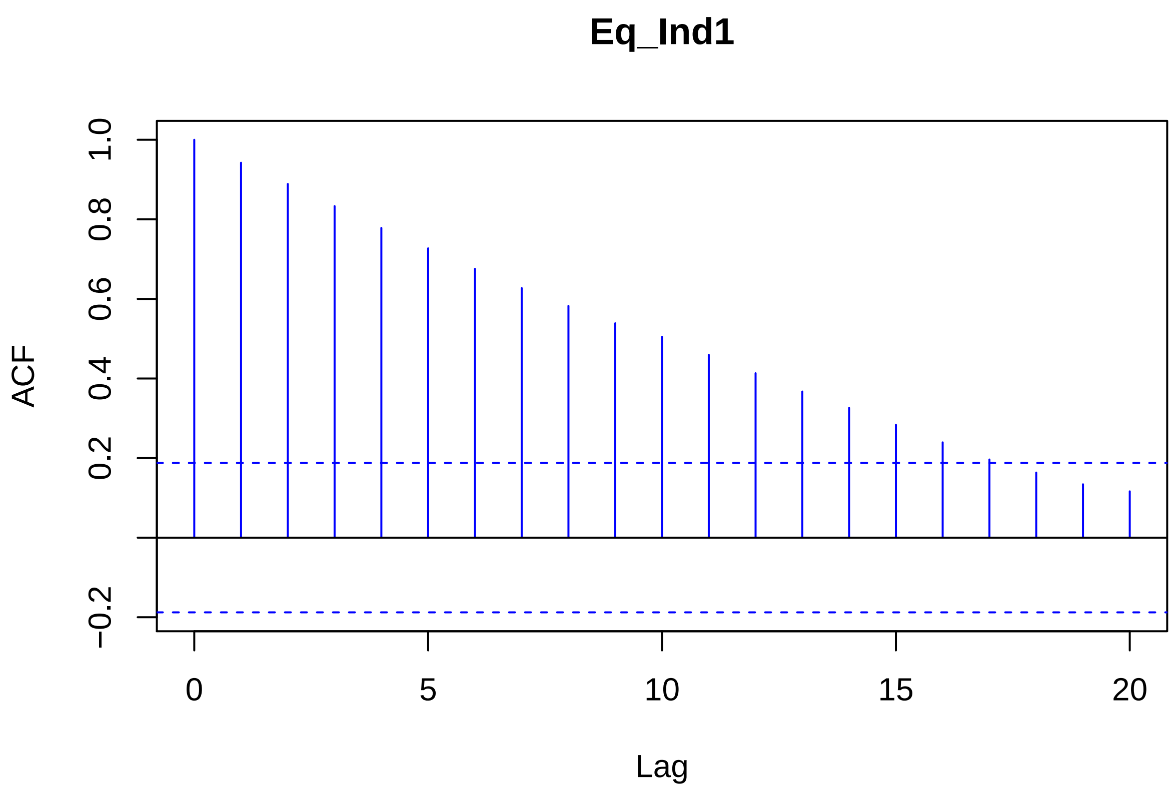
Both Equity indices show high correlation with LIBOR rate and less-to-no correlation with VIX at index levels.

Equity index1 and Equity Index 2 are highly correlated with each other and thus including both the variables in the model would result in the problem of multicollenearity.

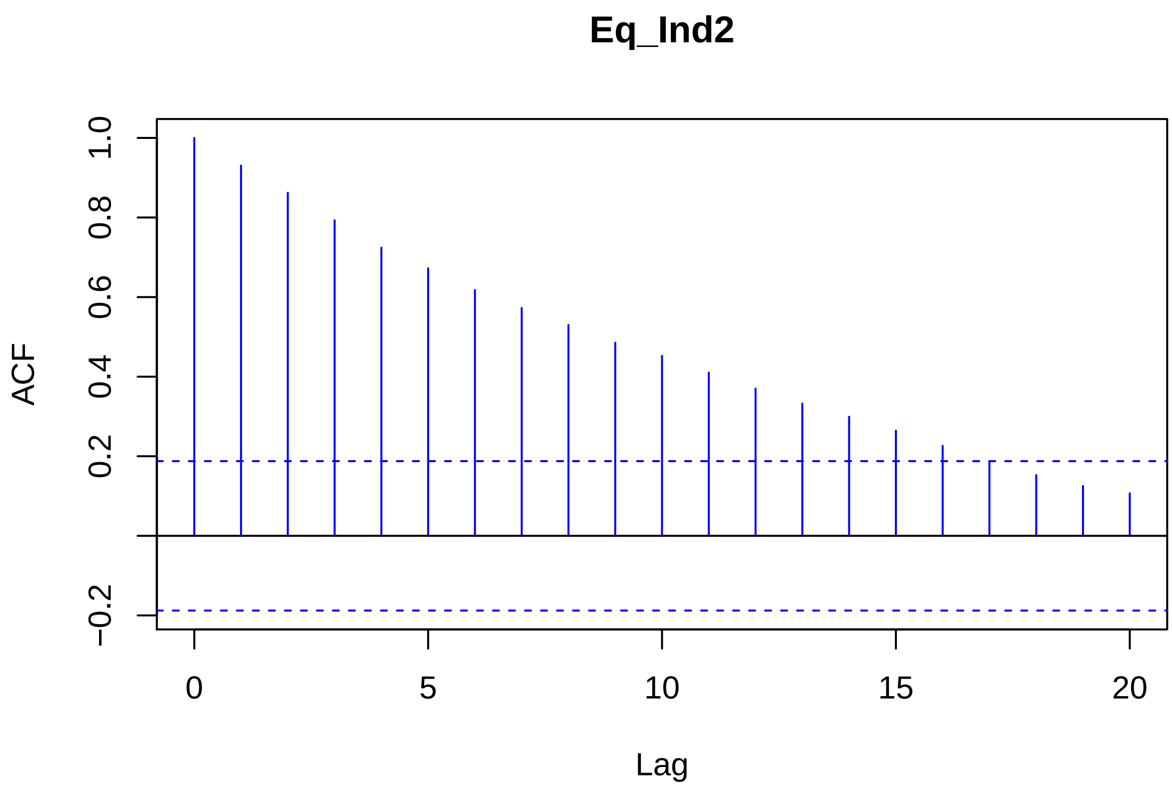
```
## This may be due to a trend present in the variables at Index level and also the indexes show non-sta
## Scatterplot
acf(VIX, col = "Blue",main="VIX")
```



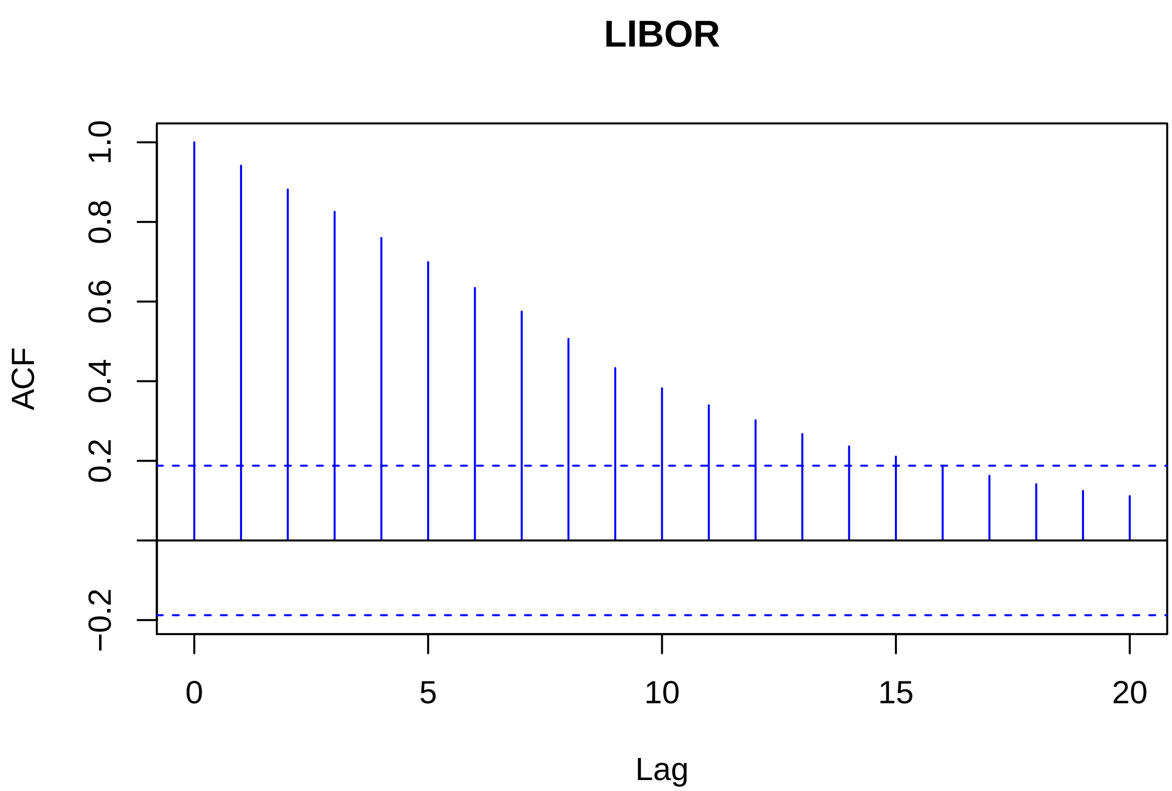

```r
acf(Eq_Ind1,col = "Blue",main="Eq_Ind1")
```



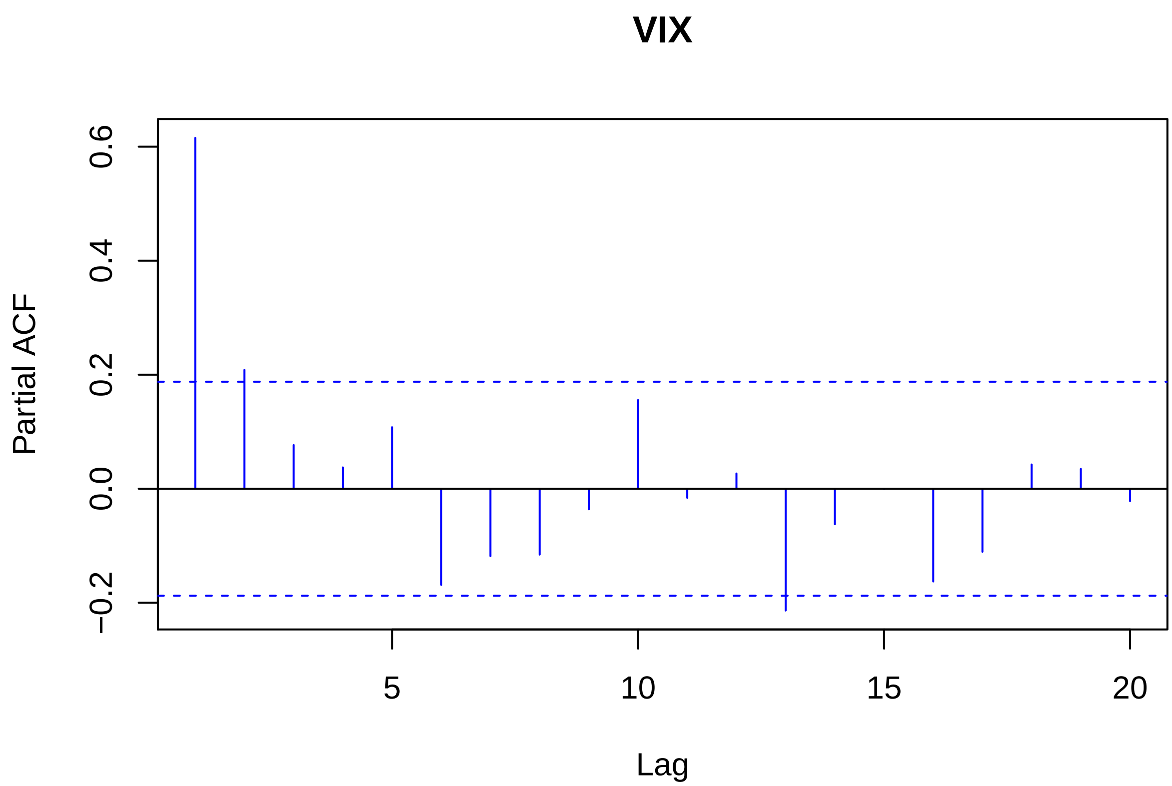

```r
acf(Eq_Ind2,col = "Blue",main="Eq_Ind2")
```



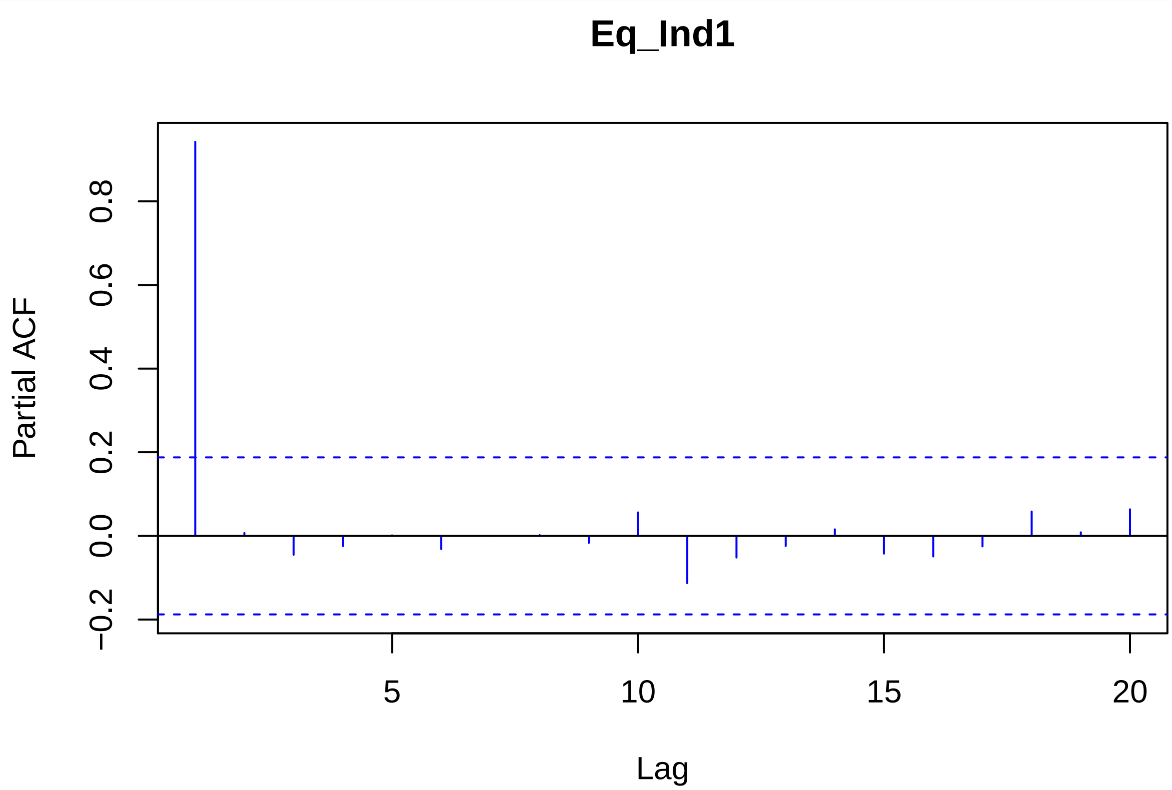

```r
acf(LIBOR,col = "Blue",main="LIBOR")
```



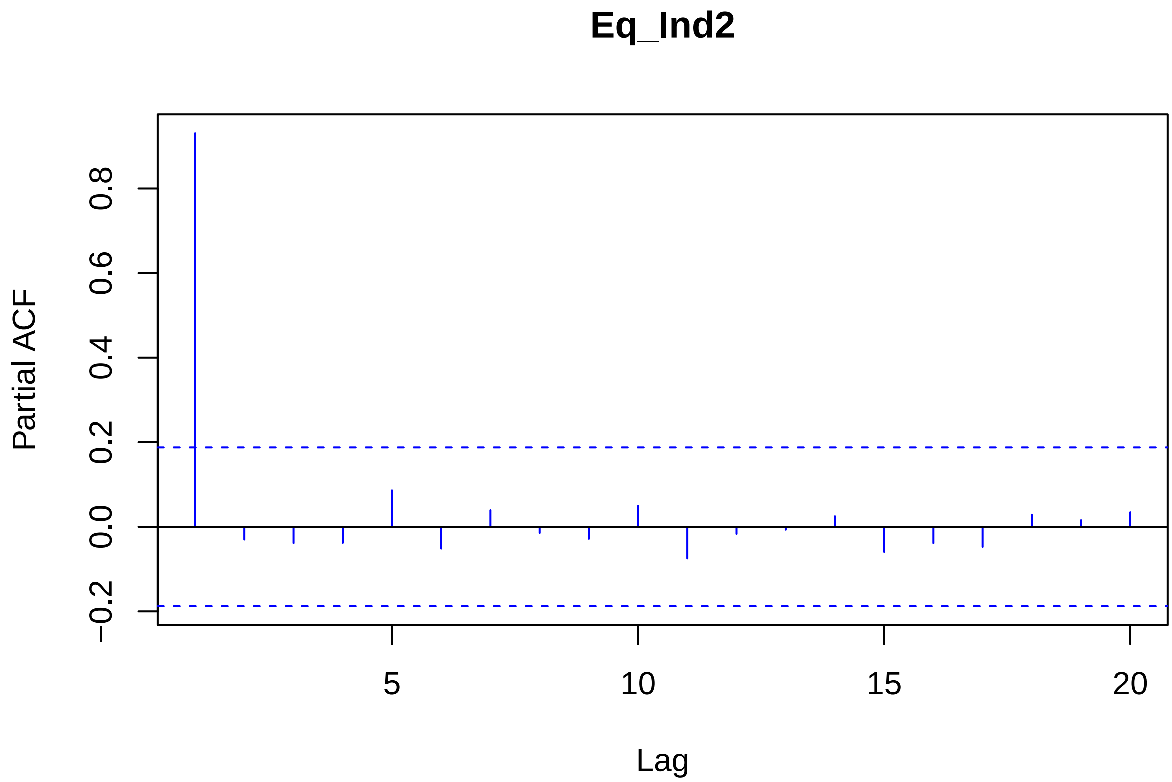

```r
pacf(VIX, col = "Blue",main="VIX")
```



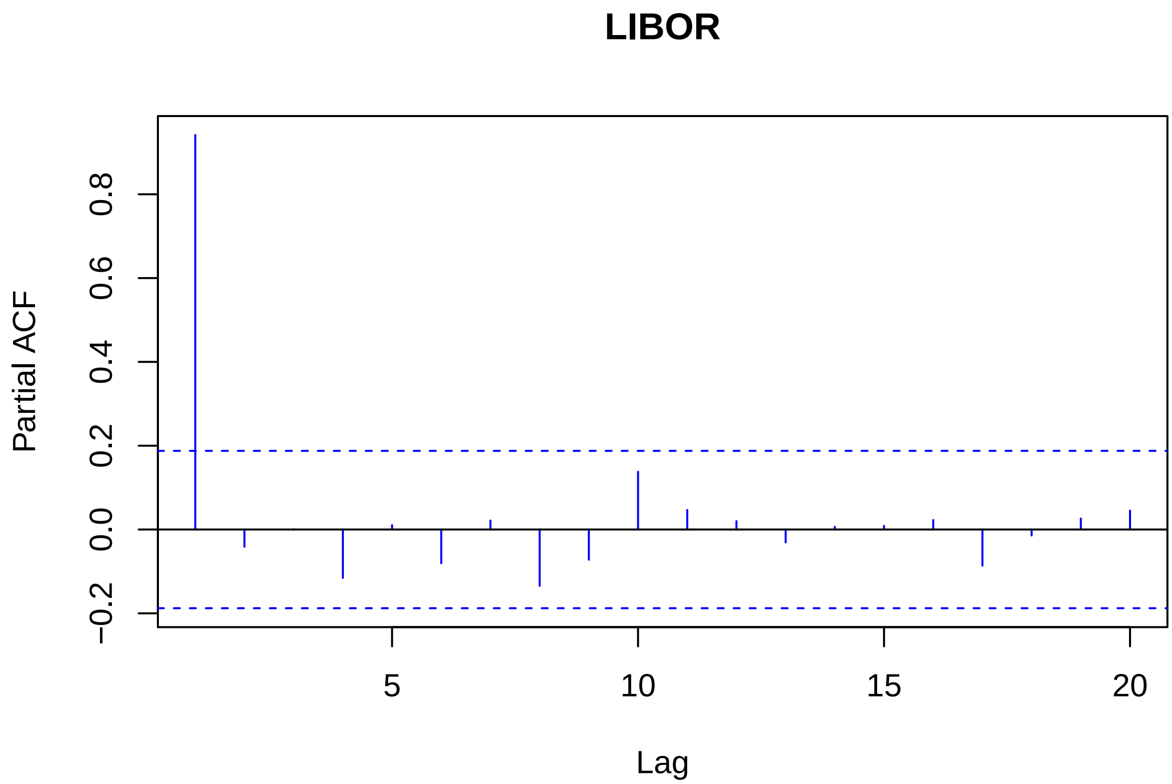

```r
pacf(Eq_Ind1,col = "Blue",main="Eq_Ind1")
```




```r
pacf(Eq_Ind2,col = "Blue",main="Eq_Ind2")
```




```r
pacf(LIBOR,col = "Blue",main="LIBOR")
```



ACF and PACF plots suggest non-stationary series.

```
## ACF and PACF plots suggest non-stationary series.

adf.test(VIX)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  VIX
## Dickey-Fuller = -2.1521, Lag order = 4, p-value = 0.5136
## alternative hypothesis: stationary
```

```
adf.test(Eq_Ind1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Eq_Ind1
## Dickey-Fuller = -1.5329, Lag order = 4, p-value = 0.7704
## alternative hypothesis: stationary
```

```
adf.test(Eq_Ind2)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Eq_Ind2
## Dickey-Fuller = -0.93778, Lag order = 4, p-value = 0.9443
## alternative hypothesis: stationary
```

```
adf.test(LIBOR)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  LIBOR
## Dickey-Fuller = -2.879, Lag order = 4, p-value = 0.212
## alternative hypothesis: stationary
```

```
## adf test results in non-stationary time series
```

Adf test results in non-stationary time series Possible transformations to consider are QoQ growth and QoQ diff.

```
### Take transformations on the dataset.

### QoQ Percent changes

QoQ_pcent_chg<-function(x){
  qoq<-c((x[2:length(x)]/x[1:length(x)-1])-1)
  return(qoq)
}

### QoQ Differences


QoQ_diff<-function(x){
  qoq<-c(x[2:length(x)]- x[1:length(x)-1])
```

```r
  return(qoq)
}


## Applying QoqPcent Change
Vix_qoq_pcent_chng<-as.data.frame(sapply(Vix_data_TS_Qtrly, function(x)QoQ_pcent_chg(x) ))
colnames(Vix_qoq_pcent_chng)<-paste0(colnames(Vix_qoq_pcent_chng),"_QoQ_pcentchg")

## Applying QoqPcent Difference
Vix_qoq_diff<-as.data.frame(sapply(Vix_data_TS_Qtrly, function(x)QoQ_diff(x) ))
colnames(Vix_qoq_diff)<-paste0(colnames(Vix_qoq_diff),"_QoQ_diff")

## Combinging all model Variabes - To use for all combinations model- Exhaustive Search.
VIX_with_Transf<-cbind(Vix_data_TS_Qtrly[-1,],Vix_qoq_pcent_chng,Vix_qoq_diff)
VIX_with_Transf<-as.data.frame(VIX_with_Transf[is.finite(rowSums(VIX_with_Transf)), ])

transf_names<-c(names(Vix_qoq_diff),names(Vix_qoq_pcent_chng))
VIX_transf<-VIX_with_Transf[transf_names]

## Checking for Correlation and Stationarity.
cor(VIX_transf)
```

```
##                     Eq_Ind1_QoQ_diff Eq_Ind2_QoQ_diff VIX_QoQ_diff
## Eq_Ind1_QoQ_diff          1.00000000       0.83014743   -0.6450241
## Eq_Ind2_QoQ_diff          0.83014743       1.00000000   -0.4639424
## VIX_QoQ_diff             -0.64502414      -0.46394236    1.0000000
## LIBOR_QoQ_diff            0.27208856       0.19288595   -0.1540489
## Eq_Ind1_QoQ_pcentchg      0.94044756       0.74382041   -0.6792112
## Eq_Ind2_QoQ_pcentchg      0.82879342       0.88825335   -0.5718103
## VIX_QoQ_pcentchg         -0.59867102      -0.43814529    0.9466422
## LIBOR_QoQ_pcentchg        0.05832665       0.08232148   -0.1574860
##                     LIBOR_QoQ_diff Eq_Ind1_QoQ_pcentchg Eq_Ind2_QoQ_pcentchg
## Eq_Ind1_QoQ_diff        0.27208856           0.94044756           0.82879342
## Eq_Ind2_QoQ_diff        0.19288595           0.74382041           0.88825335
## VIX_QoQ_diff           -0.15404886          -0.67921116          -0.57181029
## LIBOR_QoQ_diff          1.00000000           0.23497158           0.18478785
## Eq_Ind1_QoQ_pcentchg    0.23497158           1.00000000           0.87656486
## Eq_Ind2_QoQ_pcentchg    0.18478785           0.87656486           1.00000000
## VIX_QoQ_pcentchg       -0.16716011          -0.62185795          -0.52627549
## LIBOR_QoQ_pcentchg     -0.06155717           0.04445196           0.04530392
##                     VIX_QoQ_pcentchg LIBOR_QoQ_pcentchg
## Eq_Ind1_QoQ_diff          -0.5986710         0.05832665
## Eq_Ind2_QoQ_diff          -0.4381453         0.08232148
## VIX_QoQ_diff               0.9466422        -0.15748603
## LIBOR_QoQ_diff            -0.1671601        -0.06155717
## Eq_Ind1_QoQ_pcentchg      -0.6218580         0.04445196
## Eq_Ind2_QoQ_pcentchg      -0.5262755         0.04530392
## VIX_QoQ_pcentchg           1.0000000        -0.15455490
## LIBOR_QoQ_pcentchg        -0.1545549         1.00000000
```

```r
sapply(VIX_transf, function(x) adf.test(c(x)) )
```

```
## Warning in adf.test(c(x)): p-value smaller than printed p-value
```

```
## Warning in adf.test(c(x)): p-value smaller than printed p-value

## Warning in adf.test(c(x)): p-value smaller than printed p-value

## Warning in adf.test(c(x)): p-value smaller than printed p-value

## Warning in adf.test(c(x)): p-value smaller than printed p-value

##             Eq_Ind1_QoQ_diff               Eq_Ind2_QoQ_diff
## statistic   -3.270463                      -4.06412
## parameter   4                              4
## alternative "stationary"                   "stationary"
## p.value     0.07978949                     0.01
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "c(x)"                         "c(x)"
##             VIX_QoQ_diff                   LIBOR_QoQ_diff
## statistic   -5.287762                      -4.027948
## parameter   4                              4
## alternative "stationary"                   "stationary"
## p.value     0.01                           0.01048699
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "c(x)"                         "c(x)"
##             Eq_Ind1_QoQ_pcentchg           Eq_Ind2_QoQ_pcentchg
## statistic   -3.593909                      -4.582273
## parameter   4                              4
## alternative "stationary"                   "stationary"
## p.value     0.03704257                     0.01
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "c(x)"                         "c(x)"
##             VIX_QoQ_pcentchg               LIBOR_QoQ_pcentchg
## statistic   -4.875446                      -4.792235
## parameter   4                              4
## alternative "stationary"                   "stationary"
## p.value     0.01                           0.01
## method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
## data.name   "c(x)"                         "c(x)"
```

```
## Hence the series are stationary at 5% significance level. Hence we can use these series in our model

## VIX has good correlation with Equity index 1  and mild correlation with Equity index 2 after transfo

## Building Growth Model and First Difference model.
## Intuitively Growth model makes more sense as capturing the growth aspects would be very useful as th
chang_model<-VIX_transf[names(Vix_qoq_pcent_chng)]
VIX_Full_model<-lm(VIX_QoQ_pcentchg~.,data=chang_model,na.action=na.exclude)
summary(VIX_Full_model)
```

```
##
## Call:
## lm(formula = VIX_QoQ_pcentchg ~ ., data = chang_model, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66907 -0.16217 -0.03336  0.09507  1.12009
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.08895    0.02485   3.580 0.000526 ***
## Eq_Ind1_QoQ_pcentchg -2.80045    0.64079  -4.370 2.97e-05 ***
## Eq_Ind2_QoQ_pcentchg  0.21384    0.39862   0.536 0.592808
## LIBOR_QoQ_pcentchg   -0.01879    0.01120  -1.678 0.096440 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2462 on 103 degrees of freedom
## Multiple R-squared:  0.4045, Adjusted R-squared:  0.3872
## F-statistic: 23.32 on 3 and 103 DF,  p-value: 1.34e-11
```

```r
diff_model<-VIX_transf[names(Vix_qoq_diff)]
VIX_Full_model_diff<-lm(VIX_QoQ_diff~.,data=diff_model,na.action=na.exclude)
summary(VIX_Full_model_diff)
```

```
##
## Call:
## lm(formula = VIX_QoQ_diff ~ ., data = diff_model, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.4740  -2.7516   0.2363   2.8462  15.3089
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.969363   0.515949   1.879   0.0631 .
## Eq_Ind1_QoQ_diff -0.063814   0.010234  -6.235 1.01e-08 ***
## Eq_Ind2_QoQ_diff  0.005196   0.002967   1.751   0.0828 .
## LIBOR_QoQ_diff    0.430055   1.054728   0.408   0.6843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.018 on 103 degrees of freedom
## Multiple R-squared:  0.4334, Adjusted R-squared:  0.4169
## F-statistic: 26.26 on 3 and 103 DF,  p-value: 1.066e-12
```

```r
## In both the cases Equity Index 1 is significant hence we would build a model with Vix and Equity Ind
```

```r
## Building Models.

## Simulating Exhaustive Search for best model.

y_var<-as.data.frame(VIX_transf[,c("VIX_QoQ_diff","VIX_QoQ_pcentchg")])
x_var<-as.data.frame(VIX_transf[!names(VIX_transf) %in% names(y_var)])

### Function To Generate Combinations of Models
###### helper function
    factor_to_num<-function(x){
      x1<-as.numeric(as.character(x))
      return(x1)
    }

combination_of_models<-function(no_of_X,no_of_var){
```

```r
  combinations<-data.frame(combn(no_of_X,no_of_var))
  All_models<-NULL
  for (y in 1:dim(y_var)[2]) {
    for (x in 1:dim(combinations)[2]) {
      reg1<-cbind(y_var[c(y)],x_var[c(combinations)[[x]]])
      #x_data<-x_var[,c(combinations[,x])]
      #reg1<-cbind(y_var[c(y)],x_var[,c(combinations[,x])])
      reg1_sum<-summary(lm(reg1))
      model_var<-paste0(names(reg1)[1],"~",(paste0(names(reg1)[-1],collapse ="+")))
      r_sq<-reg1_sum$r.squared
      Adj_rsq<-reg1_sum$adj.r.squared
      d1<-data.frame(cbind(model_var,r_sq,Adj_rsq))
      All_models<-rbind(All_models,d1)
    }
  }

  All_models[,-1]<-sapply(All_models[,-1], function(x) factor_to_num(x))
  Ranked_rsq_model<-All_models[order(-All_models$r_sq),]

  return(Ranked_rsq_model)
}

## Best 2 Variable Factor Models
Model_2Var<-combination_of_models(6,2)
Model_2Var
```

```
##                                                       model_var       r_sq
## 14       VIX_QoQ_diff~Eq_Ind1_QoQ_pcentchg+LIBOR_QoQ_pcentchg 0.47756358
## 7          VIX_QoQ_diff~Eq_Ind2_QoQ_diff+Eq_Ind1_QoQ_pcentchg 0.46514018
## 13     VIX_QoQ_diff~Eq_Ind1_QoQ_pcentchg+Eq_Ind2_QoQ_pcentchg 0.46372461
## 3          VIX_QoQ_diff~Eq_Ind1_QoQ_diff+Eq_Ind1_QoQ_pcentchg 0.46166709
## 10           VIX_QoQ_diff~LIBOR_QoQ_diff+Eq_Ind1_QoQ_pcentchg 0.46136036
## 1              VIX_QoQ_diff~Eq_Ind1_QoQ_diff+Eq_Ind2_QoQ_diff 0.43251238
## 5            VIX_QoQ_diff~Eq_Ind1_QoQ_diff+LIBOR_QoQ_pcentchg 0.43047255
## 4          VIX_QoQ_diff~Eq_Ind1_QoQ_diff+Eq_Ind2_QoQ_pcentchg 0.42048033
## 2                VIX_QoQ_diff~Eq_Ind1_QoQ_diff+LIBOR_QoQ_diff 0.41655326
## 29   VIX_QoQ_pcentchg~Eq_Ind1_QoQ_pcentchg+LIBOR_QoQ_pcentchg 0.40284588
## 18     VIX_QoQ_pcentchg~Eq_Ind1_QoQ_diff+Eq_Ind1_QoQ_pcentchg 0.38836637
## 28 VIX_QoQ_pcentchg~Eq_Ind1_QoQ_pcentchg+Eq_Ind2_QoQ_pcentchg 0.38823696
## 22     VIX_QoQ_pcentchg~Eq_Ind2_QoQ_diff+Eq_Ind1_QoQ_pcentchg 0.38804060
## 25       VIX_QoQ_pcentchg~LIBOR_QoQ_diff+Eq_Ind1_QoQ_pcentchg 0.38717592
## 20       VIX_QoQ_pcentchg~Eq_Ind1_QoQ_diff+LIBOR_QoQ_pcentchg 0.37276872
## 16         VIX_QoQ_pcentchg~Eq_Ind1_QoQ_diff+Eq_Ind2_QoQ_diff 0.36954444
## 19     VIX_QoQ_pcentchg~Eq_Ind1_QoQ_diff+Eq_Ind2_QoQ_pcentchg 0.36130082
## 17           VIX_QoQ_pcentchg~Eq_Ind1_QoQ_diff+LIBOR_QoQ_diff 0.35842666
## 15       VIX_QoQ_diff~Eq_Ind2_QoQ_pcentchg+LIBOR_QoQ_pcentchg 0.34431612
## 8          VIX_QoQ_diff~Eq_Ind2_QoQ_diff+Eq_Ind2_QoQ_pcentchg 0.33612962
## 11           VIX_QoQ_diff~LIBOR_QoQ_diff+Eq_Ind2_QoQ_pcentchg 0.32939092
## 30   VIX_QoQ_pcentchg~Eq_Ind2_QoQ_pcentchg+LIBOR_QoQ_pcentchg 0.29408681
## 26       VIX_QoQ_pcentchg~LIBOR_QoQ_diff+Eq_Ind2_QoQ_pcentchg 0.28202621
## 23     VIX_QoQ_pcentchg~Eq_Ind2_QoQ_diff+Eq_Ind2_QoQ_pcentchg 0.28104020
## 9            VIX_QoQ_diff~Eq_Ind2_QoQ_diff+LIBOR_QoQ_pcentchg 0.22957058
## 6                VIX_QoQ_diff~Eq_Ind2_QoQ_diff+LIBOR_QoQ_diff 0.21957169
```

```
## 24       VIX_QoQ_pcentchg~Eq_Ind2_QoQ_diff+LIBOR_QoQ_pcentchg 0.20610605
## 21           VIX_QoQ_pcentchg~Eq_Ind2_QoQ_diff+LIBOR_QoQ_diff 0.19906595
## 27         VIX_QoQ_pcentchg~LIBOR_QoQ_diff+LIBOR_QoQ_pcentchg 0.05521967
## 12             VIX_QoQ_diff~LIBOR_QoQ_diff+LIBOR_QoQ_pcentchg 0.05171569
##       Adj_rsq
## 14 0.46751672
## 7  0.45485441
## 13 0.45341162
## 3  0.45131454
## 10 0.45100190
## 1  0.42159916
## 5  0.41952010
## 4  0.40933572
## 2  0.40533313
## 29 0.39136215
## 18 0.37660419
## 28 0.37647229
## 22 0.37627215
## 25 0.37539084
## 20 0.36070658
## 16 0.35742029
## 19 0.34901815
## 17 0.34608872
## 15 0.33170682
## 8  0.32336288
## 11 0.31649459
## 30 0.28051155
## 26 0.26821902
## 23 0.26721405
## 9  0.21475463
## 6  0.20456346
## 24 0.19083886
## 21 0.18366338
## 27 0.03705082
## 12 0.03347945
```

```r
## Best 1 Variable Models
Model_1Var<-combination_of_models(6,1)
Model_1Var
```

```
##                                model_var       r_sq    Adj_rsq
## 4      VIX_QoQ_diff~Eq_Ind1_QoQ_pcentchg 0.46132780 0.45619758
## 1          VIX_QoQ_diff~Eq_Ind1_QoQ_diff 0.41605614 0.41049477
## 10 VIX_QoQ_pcentchg~Eq_Ind1_QoQ_pcentchg 0.38670732 0.38086643
## 7      VIX_QoQ_pcentchg~Eq_Ind1_QoQ_diff 0.35840699 0.35229658
## 5      VIX_QoQ_diff~Eq_Ind2_QoQ_pcentchg 0.32696701 0.32055718
## 11 VIX_QoQ_pcentchg~Eq_Ind2_QoQ_pcentchg 0.27696590 0.27007986
## 2          VIX_QoQ_diff~Eq_Ind2_QoQ_diff 0.21524252 0.20776863
## 8      VIX_QoQ_pcentchg~Eq_Ind2_QoQ_diff 0.19197130 0.18427579
## 9        VIX_QoQ_pcentchg~LIBOR_QoQ_diff 0.02794250 0.01868481
## 6        VIX_QoQ_diff~LIBOR_QoQ_pcentchg 0.02480185 0.01551425
## 12   VIX_QoQ_pcentchg~LIBOR_QoQ_pcentchg 0.02388722 0.01459090
## 3            VIX_QoQ_diff~LIBOR_QoQ_diff 0.02373105 0.01443325
```

```r
# Possible Models.
## Growth Models - Economic Intuitiveness.
## Model , Rsq , Adj.rsq
# VIX_QoQ_pcentchg~Eq_Ind1_QoQ_pcentchg 0.35840699  0.35229658
# VIX_QoQ_pcentchg~Eq_Ind1_QoQ_pcentchg+LIBOR_QoQ_pcentchg  0.40284588 0.39136215

Fit1<-lm(VIX_QoQ_pcentchg~Eq_Ind1_QoQ_pcentchg,data = VIX_with_Transf)
Fit2<-lm(VIX_QoQ_pcentchg~Eq_Ind1_QoQ_pcentchg+LIBOR_QoQ_pcentchg,data = VIX_with_Transf)
Fit3<-lm(VIX_QoQ_pcentchg~Eq_Ind2_QoQ_pcentchg,data = VIX_with_Transf)

summary(Fit1)
```

```
##
## Call:
## lm(formula = VIX_QoQ_pcentchg ~ Eq_Ind1_QoQ_pcentchg, data = VIX_with_Transf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63978 -0.15556 -0.02783  0.09108  1.15280
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.08631    0.02489   3.468  0.00076 ***
## Eq_Ind1_QoQ_pcentchg -2.52214    0.30997  -8.137 8.77e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2475 on 105 degrees of freedom
## Multiple R-squared:  0.3867, Adjusted R-squared:  0.3809
## F-statistic: 66.21 on 1 and 105 DF,  p-value: 8.769e-13
```

```r
summary(Fit2)
```

```
##
## Call:
## lm(formula = VIX_QoQ_pcentchg ~ Eq_Ind1_QoQ_pcentchg + LIBOR_QoQ_pcentchg,
##     data = VIX_with_Transf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65417 -0.15854 -0.03328  0.10294  1.13522
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.08944    0.02474   3.614 0.000466 ***
## Eq_Ind1_QoQ_pcentchg -2.49922    0.30763  -8.124 9.86e-13 ***
## LIBOR_QoQ_pcentchg   -0.01871    0.01116  -1.677 0.096642 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2454 on 104 degrees of freedom
## Multiple R-squared:  0.4028, Adjusted R-squared:  0.3914
## F-statistic: 35.08 on 2 and 104 DF,  p-value: 2.272e-12
```

```r
summary(Fit3)
```

```
##
## Call:
## lm(formula = VIX_QoQ_pcentchg ~ Eq_Ind2_QoQ_pcentchg, data = VIX_with_Transf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44077 -0.16576 -0.05091  0.12466  1.35399
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.07506    0.02691   2.789  0.00627 **
## Eq_Ind2_QoQ_pcentchg -1.32774    0.20936  -6.342 5.82e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2687 on 105 degrees of freedom
## Multiple R-squared:  0.277,  Adjusted R-squared:  0.2701
## F-statistic: 40.22 on 1 and 105 DF,  p-value: 5.82e-09
```

```r
## Insample MAPE and MSE
Inssample_performance<-function(fitted_model){
      mape<-MAPE(fitted_model$fitted.values,fitted_model$model[,1])
      mse<-MSE(fitted_model$fitted.values,fitted_model$model[,1])
      rmse<-RMSE(fitted_model$fitted.values,fitted_model$model[,1])
    return(data.frame(mape,mse,rmse))

}

Fit1_insample<-Inssample_performance(Fit1)
Fit2_insample<-Inssample_performance(Fit2)
Fit3_insample<-Inssample_performance(Fit3)

Fit1_insample
```

```
##       mape        mse      rmse
## 1 3.288209 0.06010594 0.2451651
```

```r
Fit2_insample
```

```
##       mape        mse      rmse
## 1 3.176089 0.05852428 0.2419179
```

```r
Fit3_insample
```

```
##       mape        mse      rmse
## 1 2.452537 0.07086118 0.2661976
```

Final Model Selection.

Fit2 is the best Model as it is able to capture the VIX growth using an Equity Index Growth and Libor Growth. This helps to understand additional relationship between the variables and index movements. LIBOR growth is significant at 10% significance level and can be an accepted model in terms of its economic intuitiveness.

```r
## Final Model Selection.
```

```
# Fit2 is the best Model as it is able to capture the VIX growth using an Equity Index Growth and Libor
```

Question 1.d

```
## Question 1.d
## Since Forward values are provided only for equity index 1 and equity index 2
## Use Models - Fit1 and Fit3 to assess the value of Vix Index.

#The Equity Index 1 drops to 1300 2018Q1
#Equity Index 2 increases to level 10000

# Equity Index1
tail(VIX_with_Transf,1)
```

```
##        Eq_Ind1  Eq_Ind2   VIX    LIBOR Eq_Ind1_QoQ_pcentchg
## 201712 2673.61 6903.389 11.04 -0.80565           0.06122587
##        Eq_Ind2_QoQ_pcentchg VIX_QoQ_pcentchg LIBOR_QoQ_pcentchg
## 201712           0.06272053        0.1608833         0.02945311
##        Eq_Ind1_QoQ_diff Eq_Ind2_QoQ_diff VIX_QoQ_diff LIBOR_QoQ_diff
## 201712           154.25           407.43         1.53       -0.02305
```

```
#       VIX     Eq1       EQ2
#2017Q4 11.04   2673.61   6903.389

eq1_2018Q1_chng<-data.frame(Eq_Ind1_QoQ_pcentchg=((1300/2673.61)-1))
eq2_2018Q1_chng<- data.frame(Eq_Ind2_QoQ_pcentchg=((10000/6903.389)-1))
predict(Fit1,newdata=eq1_2018Q1_chng)
```

```
##        1
## 1.382105
```

```
predict(Fit3,newdata=eq2_2018Q1_chng)
```

```
##          1
## -0.5205186
```

```
## VIX_qoq_chng for 2018 Q1 is 1.382105 when Equity Index 1 Drops to 1300
## Vix at index level = 26.29844
(predict(Fit1,newdata=eq1_2018Q1_chng)+1)*11.04
```

```
##        1
## 26.29844
```

```
## VIX_qoq_chng for 2018 Q1 is -0.5205186 when Equity Index 2 Jumps to 10000
## Vix at index level = 5.293474
(predict(Fit3,newdata=eq2_2018Q1_chng)+1)*11.04
```

```
##        1
## 5.293474
```

Question 2: Run of Heads

```
##########################################################################################
## Question 2: Run of Heads
##########################################################################################
```

```
#install.packages("plyr")
#install.packages("randtests")
library(plyr)
library(base)
library(randtests)
```

Section 2

A Q.Write a function which generates random sequences of heads and tails for any value of n.

```
### Function to generate coin sequence of length n
Random_coin_seq<-function(n){

  coin_seq<-rbinom(n,1,0.5)
  mapped_values<-mapvalues(coin_seq, c(0, 1), c("T", "H"))
  return(mapped_values)

  }
```

2B Q.Write a function which computes the length of the longest run of heads in an arbitrary sequence of heads and tails.

```
### Function for Computing the length of longest run of heads

Function_longest_run_heads<-function(given_sequence){

  ## using "rle" function to calculate length all runs
  test1<-rle(given_sequence)

  Run_length_df<-data.frame(Run_length=test1$lengths,Value=test1$values)
  max_run_heads<-max(Run_length_df[Run_length_df$Value=="H",]$Run_length)
  return(max_run_heads)
}


### Alternate Implementation This function can also be used to calculate the length of max runs of head
Runs <- function(n){
  summary <- list(list())
  seq <- Random_coin_seq(1000)
  seq1 <- seq[-1]#Sequence without the first character
  seq2 <- seq[-length(seq)] #Sequence without the last character of the sequence
  # Comparing the two sequences is equivalent to comparing each character with its previous character
  TF <- seq1!=seq2
  # Add the first and the last character position since they were not included in seq1 and seq2
  runs <- c(0,which(TF),length(seq))
  # difference of the positions that are not runs
  # gives the length of runs
  runs_table <- cbind(seq[runs[-1]],as.numeric(as.character(diff(runs))))
  tab <- runs_table[which(runs_table[,1]=="H"),]
  max_len <- max(tab[,2])
  summary[[1]] <- seq;
  summary[[2]] <- runs_table;
  summary[[3]] <- as.numeric(max_len)
```

```
  summary[[4]] <- paste0("maximum number of heads: ", max_len)
  return(summary)
}
Runs_simulation<-Runs(1000)
```

2c.

Q.For a sequence of length n = 1000, we have observed a longest run of heads equal to Mn = 6. Based on this piece of information, do you believe that the coin is fair?

A prestep to this question is forming a hypothesis. Hypothesis: Let Null Hypothesis - H0: The coin is fair. i.e. P(H)=1/2 Alternate Hypothesis - Ha: The coin is not fair. i.e. P(H) != 1/2 Alpha - 5% confidence Level.

Assuming the coin is fair: We know that a fair coin has equal probability of either heads or tails. A coin toss problem is essentially a Binomial Distribution problem. The mean and variance of a Binomial Distribution is well known.

Since we are interested in finding if a coin is fair based on the longest run of heads observed (purely by chance).

Applying the above argument to our problem, we are interested in determining the likelihood of arriving at a longest run of heads = 6, i.e the probability argument can be framed as :- length of the sequence having run of heads no more than 6 is P(X<=6); from N trials.

There are two approaches to go about in this regards,

Step 1.

Given N=1000 and X(N) - longest run of heads(x) = 6, simulate random sequence of N trials, "M" times. At each simulation, calculate the longest run of heads for N trials. We arrive at a distribution of longest runs with M observations.

A simple proportions table would give us the probability of observing a given longest run of heads. Also, the calculating the quantile and checking if X(N) lies below the 5% quantile.

A simulation for Approach 1 is shown below. It is seen that the probability of observing a (longest run of heads=6) is less than 0.05. – Value of MAx_run_length=6 lies below the 2% quantile.

Referring to our original hypothesis, we now have enough evidence to reject the null hypothesis and in favor of the alternate hypothesis, based on the above simulation; and conclude that the Coin is unfair.

Alternatively, on purely chance basis, our best guess at the longest run of heads would be the mean of the distribution of run of heads for N trials. Articles show that the mean and variance of such a distribution approximate to (mu=log2(N)-2/3 ,var=ln(N)).

Using this approximate result, our expected value of the longest run would be around 9 to 10 (~9.3). Calculating how far away is 6 from the mean value of 10 can be done using a standard normal curve.

Step 2: This results in arbitary calculation of the probability of obtaining a run of heads no more than 6 leveraging a recurisve formula as described in the articles below.

https://www.csun.edu/~hcmth031/tlroh.pdf https://www.stat.wisc.edu/courses/st309-larget/heads-run.pdf http://tmcs.math.unideb.hu/load_doc.php?p=218&t=doc http://www.gregegan.net/QUARANTINE/Runs/Runs.html

Total Probability of Max_Run length =6 is prob(X<=6) - prob(X<=5) for 'N' trials (N=1000)

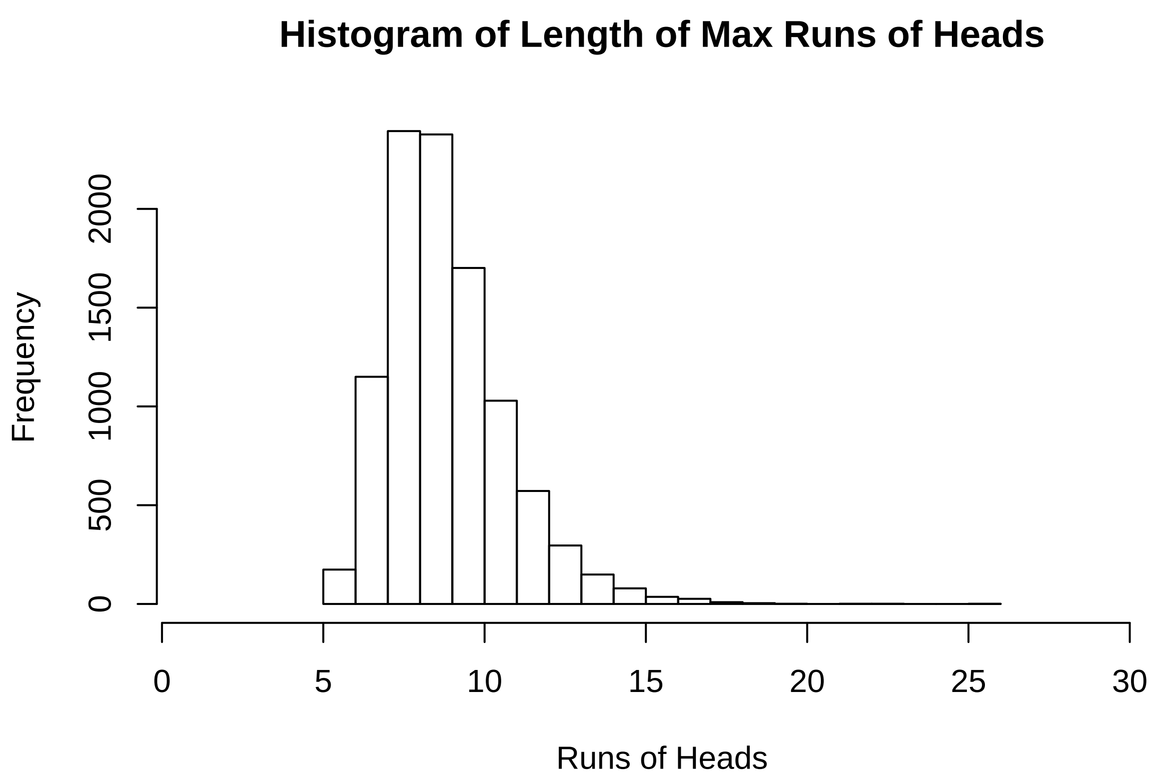
Applying the recursive formula to calculate #no. Sequence of no more than 6 , we obtain a probability of 0.0123. Using this to base our hypothesis, we now have enough evidence to reject null and conclude the coin is not fair .

Research shows that observing longest run of heads x increases with the order log(trials) as trials increases.

```r
## Step 1
### Simulation for Longest runs of "heads"
runs_vector<-data.frame()
for (i in 1:10000) {
  seq1<-Random_coin_seq(1000)
  longest_run<-Function_longest_run_heads(seq1)
  runs_vector<-rbind(longest_run,runs_vector)
  }

names(runs_vector)<-c("Max_Runs")
prop.table(table(runs_vector))
```

```
## runs_vector
##      5      6      7      8      9     10     11     12     13     14     15
## 0.0002 0.0172 0.1150 0.2394 0.2377 0.1701 0.1029 0.0572 0.0296 0.0149 0.0079
##     16     17     18     19     20     22     23     26
## 0.0036 0.0026 0.0009 0.0004 0.0001 0.0001 0.0001 0.0001
```

```r
hist(runs_vector[,1],main="Histogram of Length of Max Runs of Heads",xlim=c(1,30),xlab = "Runs of Heads
```




```r
#runs_vector
quantile(c(runs_vector[,1]),probs = c(0.1, 0.5, 1, 2, 5, 10, 25,50,75,100)/100)
```

```
## 0.1% 0.5%   1%   2%   5%  10%  25%  50%  75% 100%
##    6    6    6    7    7    7    8    9   10   26
```

```r
## The Quantile and Proportions for 6 are less than 5% and 0.011.
## This gives us enough evidence to reject the null at 5% confidence Level.
```

```r
## Step 2
```

```
# References: https://www.csun.edu/~hcmth031/tlroh.pdf
#             https://www.stat.wisc.edu/courses/st309-larget/heads-run.pdf
#             http://tmcs.math.unideb.hu/load_doc.php?p=218&t=doc
#             http://www.gregegan.net/QUARANTINE/Runs/Runs.html


###############################################################################
## for a coin toss: sample space(SS) 2^n
## eg. coin tossed 1 time: SS= 2^1 = 2
##    coin tossed 2 time: SS= 2^2 = 4
##    coin tossed 3 time: SS= 2^3 = 8
##    coin tossed 6 time: SS= 2^6 = 32
##    coin tossed 1000 time: SS= 2^1000
## probability_h<-d*(d^n-1)/(d-1)

## Recursive solution.
## A_n_Runs ; Prob = A_n_runs/2^n

maxruns=6
runs=maxruns+1
n=1000
total<-c()
a_n_runs<-c()
########### Simulation for MAXRun of length 6
for (i in 1:(n+1)) {
  if(i<=runs){
    a_n_runs[i]<-c(2^(i-1))
    total[i]<-a_n_runs[i]
  }else if(i>runs){
    a_n_runs[i]<-total[i-1]+total[i-2]+total[i-3]+total[i-4]+total[i-5]+total[i-6]+total[i-7]
        total[i]<-a_n_runs[i]
  }
}

## Total Probability of max_run_of_heads <= 6
total_prob_6<-total/(2^n)
total_prob_6[n+1]
```

```
## [1] 0.01821667
```

```
############# Simulation for MAXRun of length 5
maxruns=5
runs=maxruns+1
n=1000
total<-c()
a_n_runs<-c()
## Recursive loop formula for MAXRun of length 5
for (i in 1:(n+1)) {
  if(i<=runs){
    a_n_runs[i]<-c(2^(i-1))
    total[i]<-a_n_runs[i]
  }else if(i>runs){
    a_n_runs[i]<-total[i-1]+total[i-2]+total[i-3]+total[i-4]+total[i-5]+total[i-6]
    total[i]<-a_n_runs[i]
```

```r
  }
}
## Total Probability of max_run_of_heads <= 5
total_prob_5<-total/(2^n)
total_prob_5[n+1]
```

```
## [1] 0.0002724452
```

```r
## Total Probability of Max_Run length =6 is prob(X<=6)- prob(X<=5)
total_prob_6[n+1]-total_prob_5[n+1]
```

```
## [1] 0.01794422
```

```r
## Total Probability of Max_Run length =6 is prob(X<=6)- prob(X<=5) is < 0.05.
## Hence gives us evidence to reject the null and conclude the coin is unfair at 5% significance level
```

Results from Step1 and Step2 gives us enough evidence to reject the null and conclude that coin is unfair, at 5% significance level.