Time Series Analysis - Bitcoin Competition

Loan Huynh - s3655461

Introduction

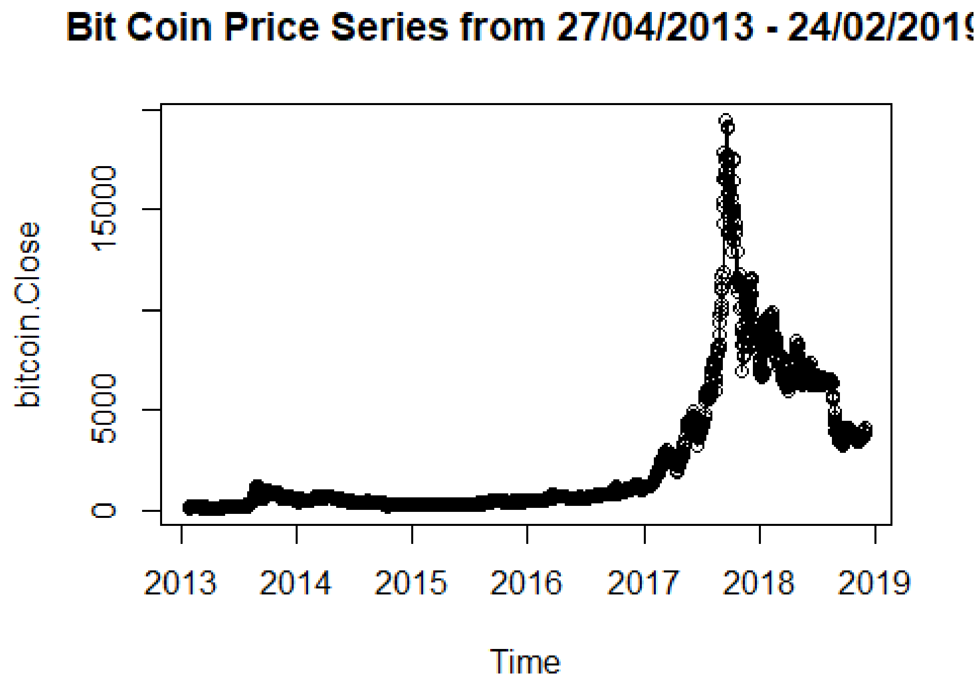
The aim of this project is trying to find the best model for forecasting Bit Coin price in next 10 days.

Dataset is collected from https://coinmarketcap.com/ website between 27 April 2013 to 24 Feb 2019.

Data Exploration

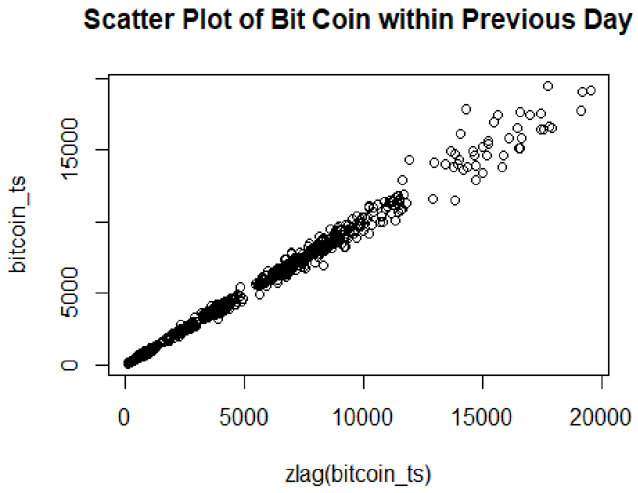
Loading data and library

```r
# Load libraries
library(TSA)
library(fUnitRoots)
library(forecast)
library(CombMSC)
library(lmtest)
library(fGarch)
library(rugarch)
library(tseries)
source("residual.analysis.R")
source("TSHandy.r")
source("MASE.R")

# This function is used to fiting model
modelfit <- function(garchOrder, armaOrder, data){
  model <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = ga
rchOrder),
                   mean.model = list(armaOrder = armaOrder, include.mean = FA
LSE),
                   distribution.model = "norm")
  model.fit <- ugarchfit(spec=model,data=data, out.sample = 100)
  return(model.fit)
}


# sortAIC_garch function is sort Garchmodels based on AIC values
sortAIC_garch <- function(list_GarchOrder, resdata)
{
  rs <- data.frame(matrix(ncol = 2))
  colnames(rs) <- c("GarchOrder", "AIC")
  for (i in 1: length(list_GarchOrder))
  {
    garchOrder <- sapply(list_GarchOrder,function(x) unlist(x))[,i]
    model <- garch(resdata, order = garchOrder)
    aic <- AIC(model)
    rs[i,] <- list(toString(garchOrder), aic)
  }
  return(rs[order(rs$AIC),])
}


# This function transform fitted residuals from forcasting ARIMA+GARCH model
into fitted BitCoin Price
fitted.values <- function(forecastResult, transformed_series)
{
  fitted.res <- forecastResult@forecast$seriesFor
  invdiff <- diffinv(fitted.res)
  latest_value <- transformed_series[length(transformed_series)]
  fitted.value.list <- c()
  for(i in 2:length(invdiff))
  {
    trans.fitted.value <- latest_value + invdiff[i]
    fitted.value.list <- c(fitted.value.list, exp(trans.fitted.value))
    latest_value <- trans.fitted.value
  }
  return(fitted.value.list)
}


#This function create model and return MASE results for 10 forecasts
checkMASE <- function(model, data_model, data.trans, observed)
{
 # model <- modelfit(garchOrder, armaOrder, data_model)
  forcRes = ugarchforecast(model, data = data_model, n.ahead = 10)
  predicted_result <- fitted.values(forcRes, data.trans)
  rs <- MASE(observed,predicted_result)$MASE
  return(rs)
}

# Load dataset
bitcoin <- read.csv("Bitcoin_Historical_Price.csv")
bitcoin_ts <- data.frame(bitcoin$Close)
rownames(bitcoin_ts) <- bitcoin$Date
bitcoin_ts <- ts(as.vector(bitcoin_ts), start = c(2013,27,4), frequency = 365
) # Convert Bitcoin price data into Time Series Object
plot(bitcoin_ts, type= 'o', main="Bit Coin Price Series from 27/04/2013 - 24/
02/2019")
```



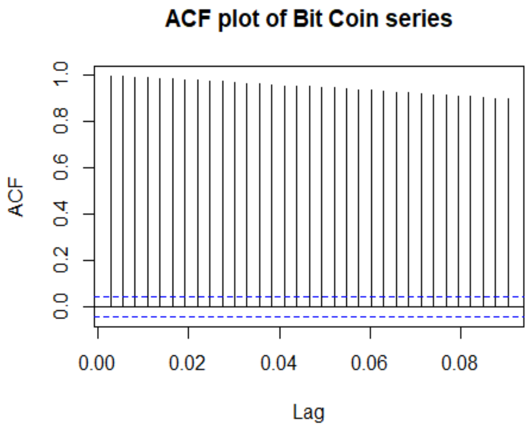
Bit Coin series plot illustrates that there is no seasonality. This series fluctuates from 2013 to 2017 and then go up significantly. It reaches a peak at nearly 20000 USD for some days in the end of 2017. After that it falls considerably in 2018 and 2019. It has succeeding points and changes of variance.

```r
# Scatter plot
plot(y = bitcoin_ts, x = zlag(bitcoin_ts), main = "Scatter Plot of Bit Coin w
ithin Previous Day")
```

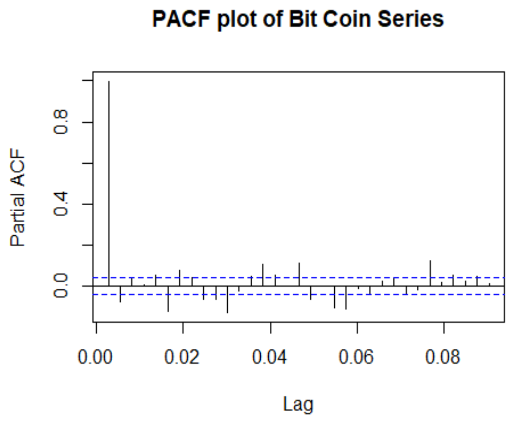


Scatter plot shows a strong relationship between Bit Coin Price and its lag.

```r
acf(bitcoin_ts, na.action = na.pass, main ="ACF plot of Bit Coin series") # ACF Plot
```



```r
pacf(bitcoin_ts, na.action = na.pass, main = "PACF plot of Bit Coin Series")
# PACF plot
```



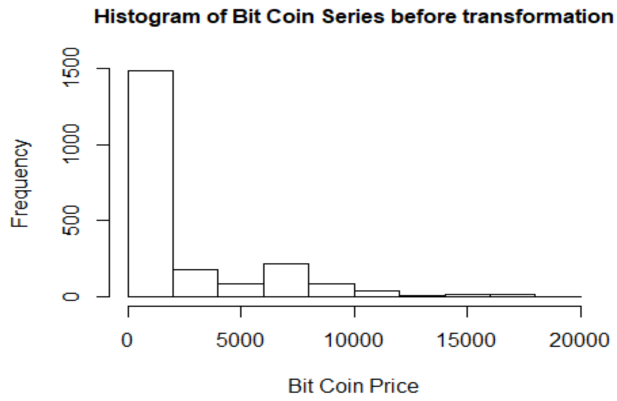
ACF plot has decaying pattern. In conclusion, Bit Coin price is a non-stationary series. In the next parts, transformation and difference are applied to transfer non-stationary series into stationary series.

Data Transformation and Difference

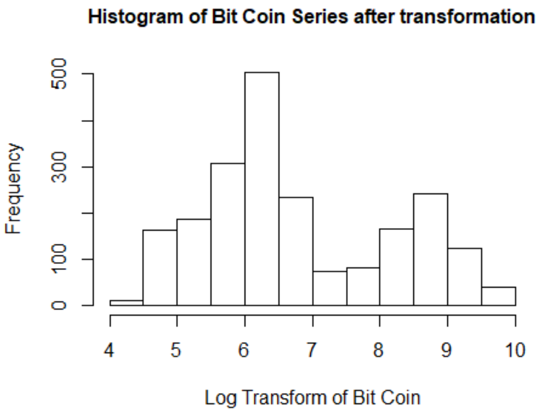
Data transformation

Log is applied for transforming data because this series skews to the right in its histogram plot and has positive values.

```r
# Log Transform
log.bc <- log(bitcoin_ts)
hist(bitcoin_ts, main = "Histogram of Bit Coin Series before transformation",
xlab = "Bit Coin Price", cex.main=1)
```

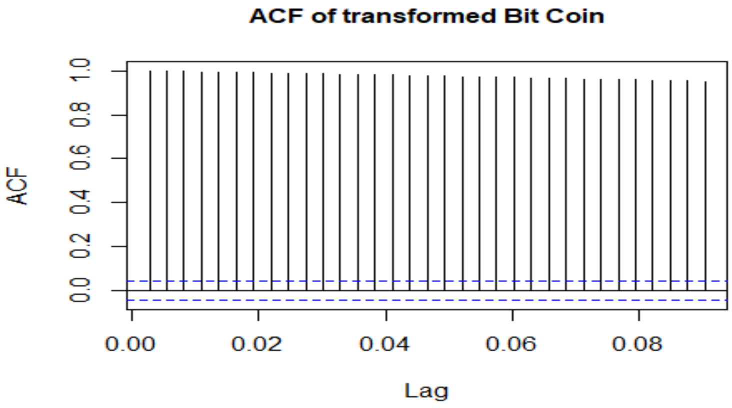


```
hist(log.bc, main = "Histogram of Bit Coin Series after transformation", xlab
= "Log Transform of Bit Coin", cex.main=1)
```

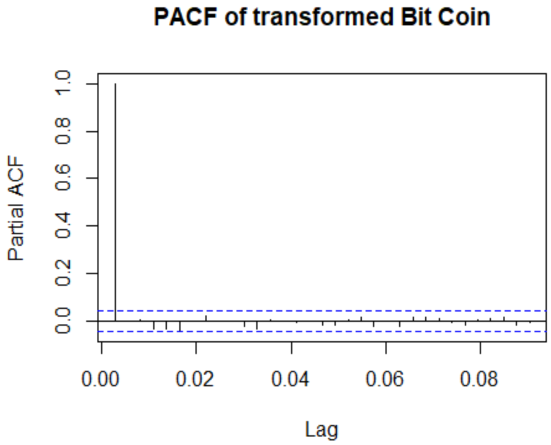


ACF and PACF of transformed data

```
acf(log.bc, na.action = na.pass, main = "ACF of transformed Bit Coin", cex.ma
in= 1)
```



```
pacf(log.bc, na.action = na.pass,  main = "PACF of transformed Bit Coin", cex
.main= 1)
```



After transformation, the series still got decaying pattern in ACF. It means that it is still a non-stationary series.

ADF Test is applied to validate the stationary of series

```
ar(diff(log.bc)) # Order 31

##
## Call:
## ar(x = diff(log.bc))
##
## Coefficients:
##        1        2        3        4        5        6        7        8
## -0.0012  -0.0140   0.0069   0.0226   0.0383   0.0607  -0.0215  -0.0010
##        9       10       11       12       13       14       15       16
## -0.0042   0.0495   0.0563  -0.0066   0.0058   0.0052   0.0024  -0.0111
##       17       18       19       20       21       22       23       24
##  0.0688   0.0064  -0.0117   0.0552  -0.0230   0.0292  -0.0450  -0.0203
##       25       26       27       28       29       30       31
## -0.0139   0.0343   0.0198  -0.0436  -0.0376  -0.0472   0.0386
##
## Order selected 31  sigma^2 estimated as  0.001853

adfTest(log.bc, lags = 31)

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 31
##   STATISTIC:
```

```
##     Dickey-Fuller: 1.0713
##   P VALUE:
##     0.9232
##
## Description:
##  Sun Jun 09 16:01:45 2019 by user: loanh
```

The p-value of ADF Test is 0.9232 larger than 0.05. Therefore, Null hypothesis cannot be rejected. It means that series is non-stationary series.

Ljung-Box Test

```
Box.test(x = log.bc, type = "Ljung-Box")
```

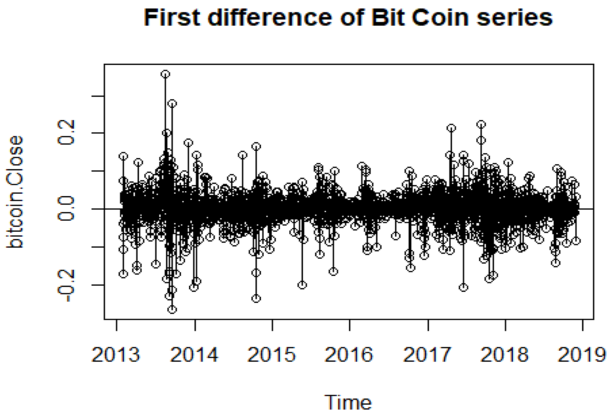
```
##
##  Box-Ljung test
##
## data:  log.bc
## X-squared = 2127.9, df = 1, p-value < 2.2e-16
```

The p-value is less than 5% of significance level. Therefore, null hypothesis of independence is rejected. It means that the series values are dependents or correlated.

Data Difference

Apply first difference for transformed series.

```
diff.log.bc <- (diff(log.bc, difference =1))
plot(diff.log.bc, type= 'o', main = "First difference of Bit Coin series")
abline(h=0)
```



The above series plot looks like moving average model. Apply ADF Test for this series

```
# ADF Test
ar(diff(diff.log.bc), na.action = na.pass) # Order is 32
```

```
##
## Call:
## ar(x = diff(diff.log.bc), na.action = na.pass)
##
## Coefficients:
##       1        2        3        4        5        6        7        8
## -0.9421  -0.9044  -0.8482  -0.7827  -0.7065  -0.6114  -0.6052  -0.5769
##       9       10       11       12       13       14       15       16
## -0.5536  -0.4762  -0.3997  -0.3867  -0.3642  -0.3395  -0.3179  -0.3147
##      17       18       19       20       21       22       23       24
## -0.2289  -0.2100  -0.2085  -0.1389  -0.1495  -0.1071  -0.1405  -0.1441
##      25       26       27       28       29       30       31       32
## -0.1403  -0.0856  -0.0503  -0.0772  -0.0950  -0.1201  -0.0560  -0.0465
##
## Order selected 32  sigma^2 estimated as  0.001961
```

```
adfTest(diff.log.bc, lags = 32)
```

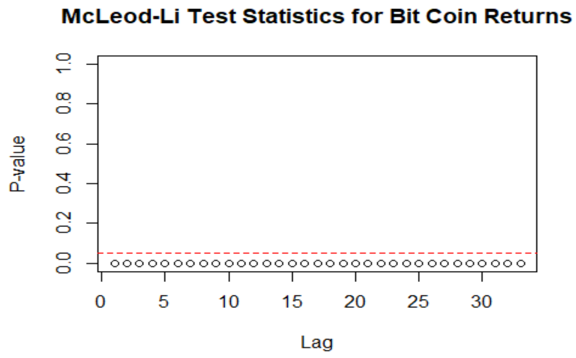
```
## Warning in adfTest(diff.log.bc, lags = 32): p-value smaller than printed p
-
## value
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 32
##   STATISTIC:
##     Dickey-Fuller: -7.4259
##   P VALUE:
##     0.01
##
## Description:
##  Sun Jun 09 16:01:45 2019 by user: loanh
```

With order of lag is 32, the differenced series becomes stationary

Apply McLeod-Li Test

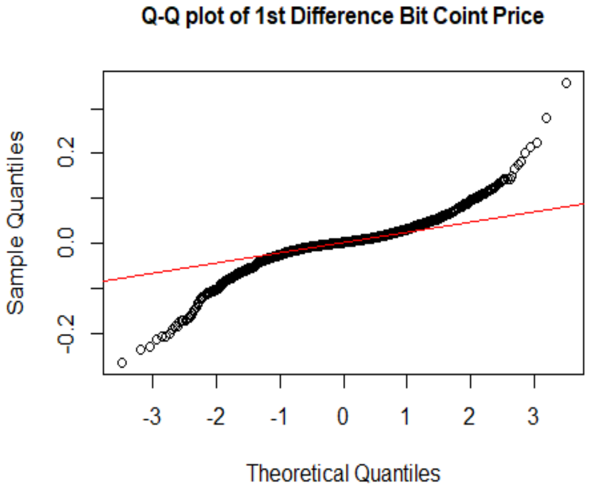
```
McLeod.Li.test(y=diff.log.bc, main="McLeod-Li Test Statistics for Bit Coin Re
turns")
```



It is clearly observed in McLeod-Li Test result that all lags are significant at 5% significance level. It implies the existence of volatility clustering.

Q-Q plot

```
qqnorm(diff.log.bc, main="Q-Q plot of 1st Difference Bit Coint Price", cex.ma
in= 1)
qqline(diff.log.bc, col="red")
```

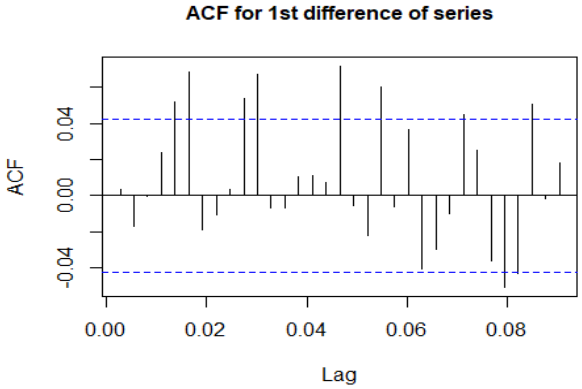


```
shapiro.test(diff.log.bc)

## 
##  Shapiro-Wilk normality test
## 
## data:  diff.log.bc
## W = 0.88509, p-value < 2.2e-16
```

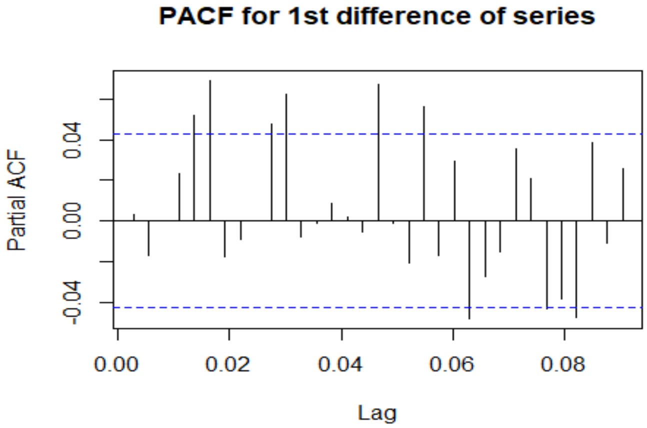
Q-Q normal plot shows that there are so many outliers at both tails. The p-value of Shapiro-Wilk test is less than 0.05. It implies that the first difference of bit coin violates normality assumption.

ACF and PACF plots

```
acf(diff.log.bc , na.action = na.pass, main = "ACF for 1st difference of seri
es", cex.main= 1)
```



```
pacf(diff.log.bc , na.action = na.pass, main = "PACF for 1st difference of se
ries", cex.main= 1)
```



There are so many significant lags in both ACF and PACF plots. There are no distinct between both plots. Due to stationarity and changes of variances problems in original series. In this report, ARMA + GARCH model is applied.

Data Modeling

ARIMA model

EACF

```
eacf(diff.log.bc)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o o o x x o o o x x  o  o  o
## 1 x o o o o x o o o o x  o  o  o
## 2 o x o o o x o o o o x  o  o  o
## 3 o x o o o x o o o o x  o  o  o
## 4 x x o x o x o o o o x  o  o  o
## 5 x x x x x o o o o o x  o  o  o
## 6 x x x x x o o o o o o  o  o  o
## 7 x x o x x x x o o o o  o  o  o
```

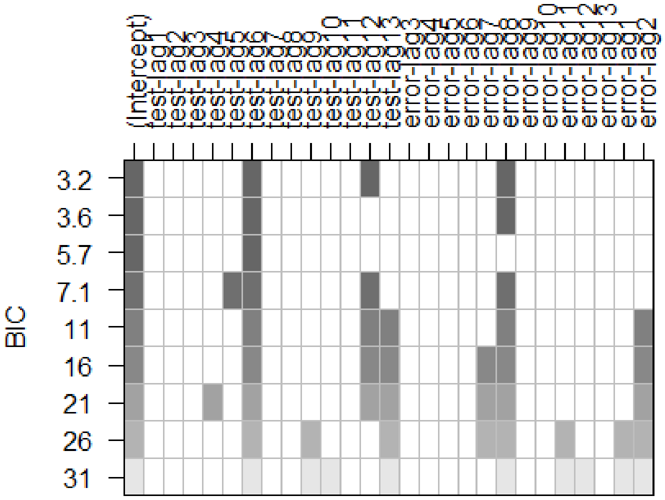
EACF suggests possibles models like ARIMA(0,1,11), ARIMA(6,1,7), ARIMA(6,1,8), ARIMA(7,1,7)

BIC Table

```r
bic = armasubsets(y=diff.log.bc,nar=13,nma=13,y.name='test',ar.method='ols')
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 2 linear dependencies found

## Reordering variables and trying again:
```

```r
plot(bic)
```



BIC table suggests ARIMA(6,1,8), ARIMA(12,1,8)

The set of possible ARIMA models is {ARIMA(0,1,11), ARIMA(6,1,7), ARIMA(6,1,8), ARIMA(7,1,7), ARIMA(12,1,8)}

Parameter Estimation

Maximum Likelihood method is considered to estimate parameters

```
armaList <- list(c(0,1,11), c(6,1,7), c(6,1,8) ,c(7,1,7),c(12,1,8))
armaEstimation <- myCandidate(bitcoin_ts, orderList = armaList, methodType =
"ML")
armaEstimation$IC
```

```
##    p d  q      AIC     AICc      BIC
## 5 12 1  8 28951.50 28951.94 29070.44
## 3  6 1  8 28992.29 28992.51 29077.24
## 2  6 1  7 29137.67 29137.87 29216.96
## 4  7 1  7 29139.09 29139.32 29224.04
## 1  0 1 11 29141.98 29142.13 29209.94
```

ARIMA(12,1,8) got smallest AIC and BIC result.

Coefficient Test

Coefficient Test is applied for all possible models based on AIC and BIC result orderly.

ARIMA(12,1,8)

```
arima.1218 <- arima(bitcoin_ts, order = c(12,1,8))
coeftest(arima.1218)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   0.336787   0.081830   4.1157 3.860e-05 ***
## ar2   0.049864   0.079912   0.6240 0.5326353
## ar3  -0.236650   0.040764  -5.8054 6.422e-09 ***
## ar4   0.285492   0.036289   7.8672 3.628e-15 ***
## ar5  -0.309696   0.038398  -8.0655 7.295e-16 ***
## ar6  -0.563696   0.048624 -11.5929 < 2.2e-16 ***
## ar7   0.379273   0.065010   5.8341 5.408e-09 ***
## ar8  -0.185615   0.072869  -2.5472 0.0108578 *
## ar9  -0.026413   0.026000  -1.0159 0.3096754
## ar10  0.124967   0.023953   5.2172 1.817e-07 ***
## ar11  0.092357   0.026588   3.4736 0.0005135 ***
## ar12 -0.228153   0.026211  -8.7046 < 2.2e-16 ***
## ma1  -0.300932   0.083136  -3.6197 0.0002949 ***
## ma2  -0.116125   0.080528  -1.4421 0.1492870
## ma3   0.252682   0.041717   6.0570 1.387e-09 ***
## ma4  -0.307463   0.031663  -9.7105 < 2.2e-16 ***
## ma5   0.470048   0.030140  15.5956 < 2.2e-16 ***
## ma6   0.569370   0.048667  11.6994 < 2.2e-16 ***
## ma7  -0.448122   0.072671  -6.1664 6.985e-10 ***
## ma8   0.328366   0.078303   4.1935 2.747e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There are only three coefficients of ar2, ar9 and ma2 not significant at 5%

ARIMA(6,1,8)

```
arima.618 <- arima(bitcoin_ts, order = c(6,1,8))
coeftest(arima.618)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.875232   0.093195  9.3914 < 2.2e-16 ***
## ar2  0.138037   0.092949  1.4851  0.137520
## ar3 -0.666490   0.090507 -7.3640 1.785e-13 ***
## ar4 -0.259465   0.093069 -2.7879  0.005306 **
## ar5  0.547189   0.131103  4.1737 2.997e-05 ***
## ar6 -0.375159   0.073673 -5.0922 3.538e-07 ***
## ma1 -0.844764   0.092324 -9.1500 < 2.2e-16 ***
## ma2 -0.225483   0.091845 -2.4550  0.014088 *
## ma3  0.720366   0.096541  7.4618 8.535e-14 ***
## ma4  0.279387   0.101792  2.7447  0.006057 **
## ma5 -0.399955   0.140533 -2.8460  0.004427 **
## ma6  0.225664   0.071024  3.1773  0.001487 **
## ma7 -0.015076   0.040233 -0.3747  0.707866
## ma8  0.218731   0.033925  6.4475 1.137e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Most of coefficients for this model are significant, except ma7 and ar2.

ARIMA(6,1,7)

```
arima.617 <- arima(bitcoin_ts, order = c(6,1,7))
coeftest(arima.617)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.3579926  0.0709701   5.0443 4.552e-07 ***
## ar2 -0.3935201  0.1276471  -3.0829 0.0020501 **
## ar3 -0.6191569  0.1016227  -6.0927 1.110e-09 ***
## ar4 -0.2848556  0.0995241  -2.8622 0.0042074 **
## ar5  0.2255639  0.0979622   2.3026 0.0213036 *
## ar6 -0.7925021  0.0674990 -11.7409 < 2.2e-16 ***
## ma1 -0.2814151  0.0740698  -3.7993 0.0001451 ***
## ma2  0.3439157  0.1385974   2.4814 0.0130867 *
## ma3  0.7001192  0.1268680   5.5185 3.419e-08 ***
## ma4  0.2505341  0.1312875   1.9083 0.0563542 .
## ma5 -0.1123529  0.1382334  -0.8128 0.4163463
## ma6  0.7000980  0.1207694   5.7970 6.752e-09 ***
## ma7 -0.0097353  0.0545837  -0.1784 0.8584437
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Most of coefficients for this model are significant, except ma4, ma5 and ma7

ARIMA(7,1,7)

```
arima.717 <- arima(bitcoin_ts, order = c(7,1,7))
coeftest(arima.717)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 -0.28178    0.25733 -1.0950 0.273507
## ar2 -0.25295    0.21251 -1.1903 0.233938
## ar3 -0.20868    0.24150 -0.8641 0.387525
## ar4 -0.11402    0.24763 -0.4604 0.645211
## ar5  0.56828    0.24461  2.3232 0.020169 *
## ar6  0.23785    0.22824  1.0421 0.297355
## ar7  0.31009    0.12941  2.3962 0.016564 *
## ma1  0.37880    0.25226  1.5016 0.133198
## ma2  0.26988    0.22240  1.2135 0.224946
## ma3  0.25043    0.25462  0.9836 0.325334
## ma4  0.10500    0.26628  0.3943 0.693334
## ma5 -0.45251    0.25764 -1.7563 0.079029 .
## ma6 -0.31727    0.22940 -1.3830 0.166659
## ma7 -0.40238    0.13319 -3.0212 0.002518 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The majority of coefficients in this model is not significant.

ARIMA(0,1,11)

```
arima.0111 <- arima(bitcoin_ts, order = c(0,1,11))
coeftest(arima.0111)
```

```
##
## z test of coefficients:
##
##         Estimate Std. Error z value  Pr(>|z|)
## ma1   0.0913312  0.0221350  4.1261 3.690e-05 ***
## ma2  -0.0048507  0.0219150 -0.2213   0.82482
## ma3   0.0158047  0.0221970  0.7120   0.47645
## ma4  -0.0307127  0.0231146 -1.3287   0.18394
## ma5   0.1264469  0.0239024  5.2901 1.222e-07 ***
## ma6  -0.0275749  0.0260263 -1.0595   0.28937
## ma7  -0.0694753  0.0272005 -2.5542   0.01064 *
## ma8   0.0651726  0.0277051  2.3524   0.01865 *
## ma9   0.0130509  0.0242927  0.5372   0.59111
## ma10  0.1183611  0.0208654  5.6726 1.407e-08 ***
## ma11  0.0592755  0.0237377  2.4971   0.01252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

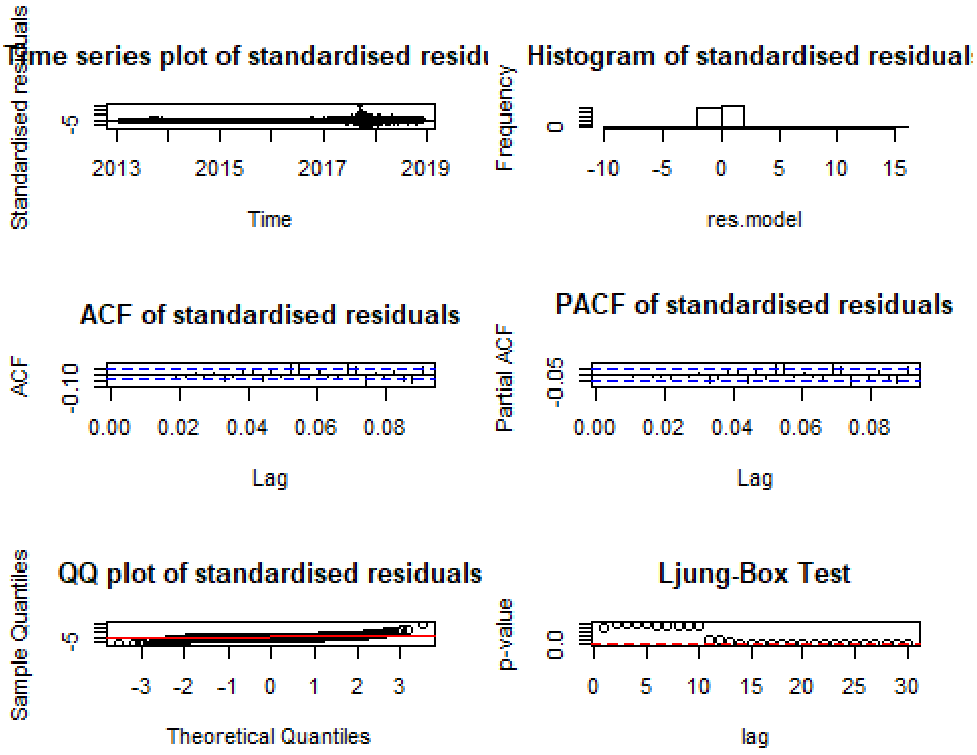
Coefficients for ma2, ma3, ma4, ma6, ma9 in arima (0,1,11) are not significant.

From coefficients test result and AIC/BIC table, the set of arima model {ARIMA(6,1,8), ARIMA(12,1,8)} is considered for diagnostics check.

MODEL DIAGNOTICS

ARIMA(6,1,8)

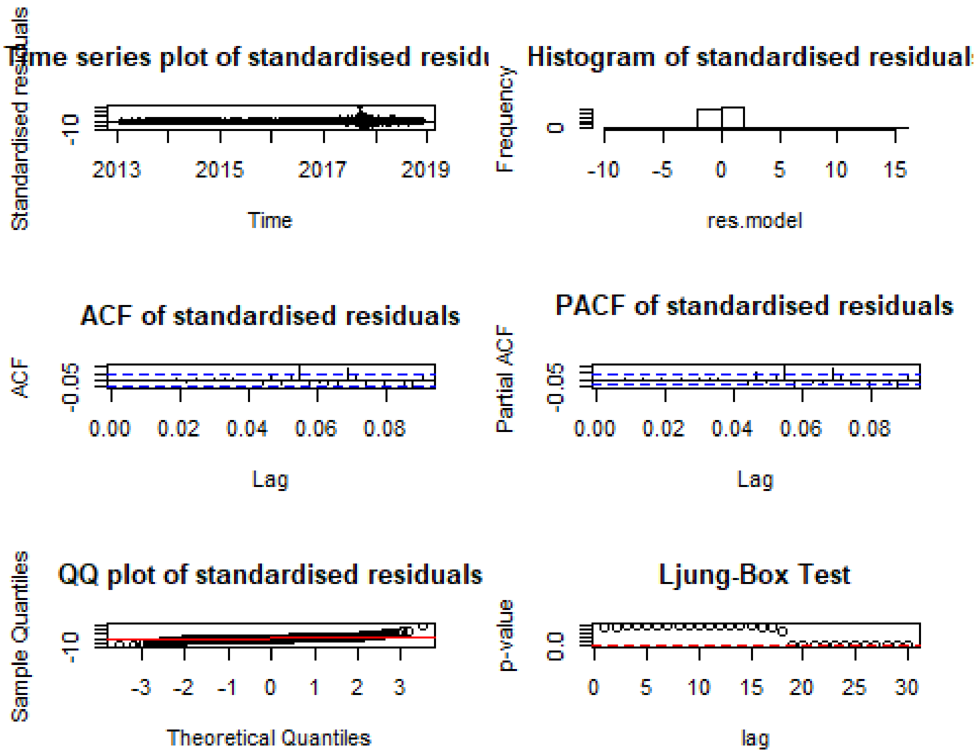
```
residual.analysis(model = arima.618, class = "ARIMA")

##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.55067, p-value < 2.2e-16
```



ARIMA(12,1,8)

```
residual.analysis(model = arima.1218, class = "ARIMA")

##
##  Shapiro-Wilk normality test
##
```

```
## data:  res.model
## W = 0.56365, p-value < 2.2e-16
```



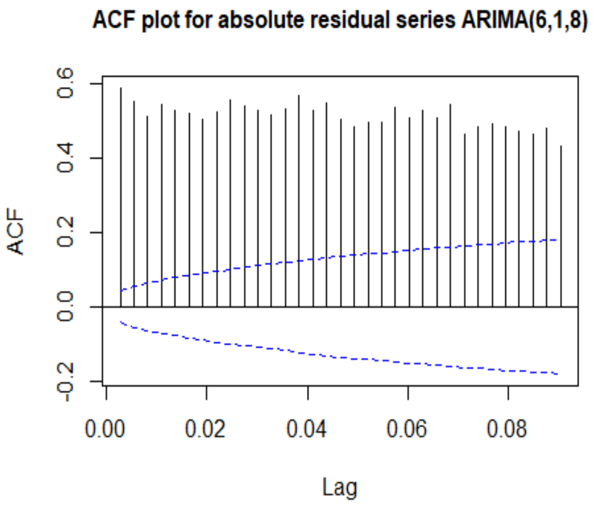
It is clearly to see that both ARIMA(12,1,8) and ARIMA(6,1,8) failed in Ljung-Box Test and violate normality of residuals. In next part, garch models are selected based on residuals of two these models.

GARCH

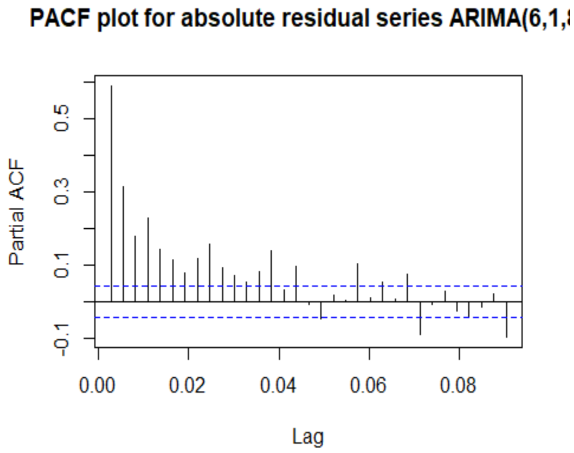
ARIMA(6,1,8)

Absolute transform

```
res.618 <- arima.618$residuals
# Apply absolute transform
ab.res.618 <- abs(res.618)
# ACF and PACF plots
acf(ab.res.618,  ci.type="ma",main="ACF plot for absolute residual series ARI
MA(6,1,8)", cex.main= 1)
```



```
pacf(ab.res.618, main="PACF plot for absolute residual series ARIMA(6,1,8)",
cex.main= 1)
```

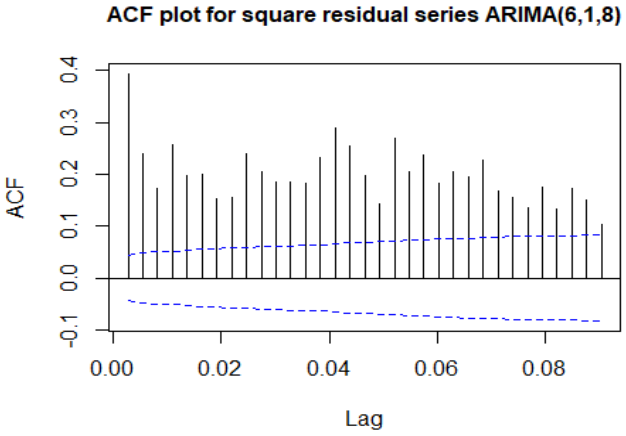


```
eacf(ab.res.618)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x 
## 1 x o x x o o x o x o o  o  o  x 
## 2 x o x x x o o o x o o  o  o  x 
## 3 x x x x o o o o x o o  o  o  x 
## 4 x x o x x o o o x o o  o  o  x 
## 5 x x o x x o x o o o o  o  o  x 
## 6 x x x x x x x o o o o  o  o  x 
## 7 x x x o x x o o o o o  o  o  x
```

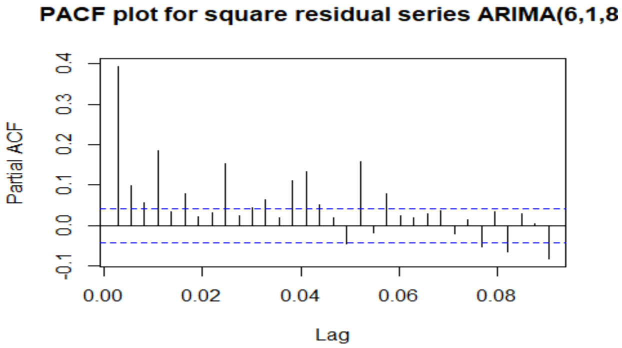
No AR/MA is suggested from eacf of absolute residuals

Squared Transform

```
sq.res.618 <- res.618^2
acf(sq.res.618, ci.type="ma",main="ACF plot for square residual series ARIMA(
6,1,8)", cex.main= 1)
```



```
pacf(sq.res.618, main="PACF plot for square residual series ARIMA(6,1,8)", ce
x.main= 1)
```



```
eacf(sq.res.618)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x x x x x x o o x o o  o  o  o
## 2 x o x x x o o x x o o  o  o  o
## 3 x x x x x o o o x o o  o  o  o
## 4 x x o o o o x x x o o  o  o  o
## 5 x x x o x x x o o o o  o  o  o
## 6 x x o o x x x o o x o  o  o  o
## 7 x o x o x o x o o x o  o  o  o
```

EACF plot suggests ARMA(5,7), ARMA(6,7), ARMA(5,8). The possible GARCH models are GARCH(7,5), GARCH(7,6), GARCH(8,5).

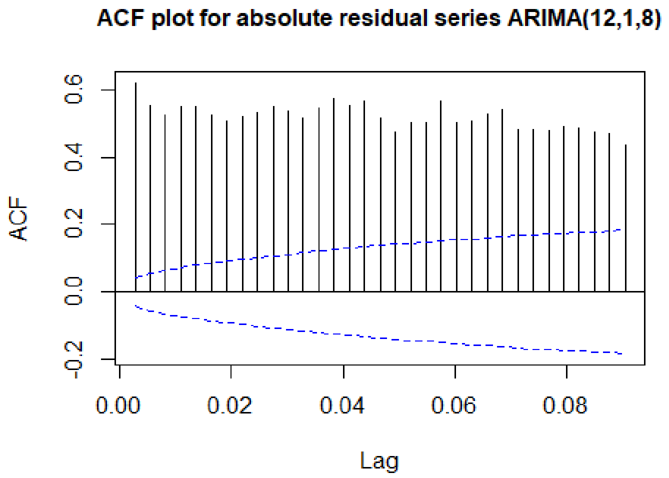
```
list_garchOrder <- list(c(7,5), c(7,6), c(8,5))
sortAIC_garch(list_garchOrder, arima.618$residuals)
```

```
##   GarchOrder      AIC
## 2       7, 6 26391.62
## 3       8, 5 26419.38
## 1       7, 5 26591.81
```

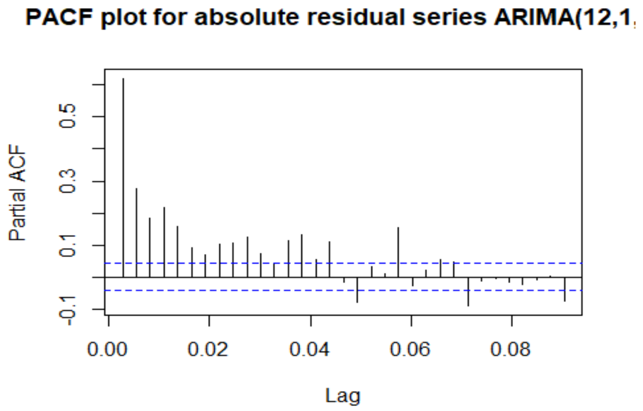
GARCH(7,6) has smallest AIC value in comparison with other models with residuals of ARIMA(6,1,8).

ARIMA(12,1,8)

```
res.1218 <- arima.1218$residuals
ab.res.1218 <- abs(res.1218)
# ACF and PACF plots
acf(ab.res.1218,  ci.type="ma",main="ACF plot for absolute residual series AR
IMA(12,1,8)", cex.main= 1)
```



```
pacf(ab.res.1218, main="PACF plot for absolute residual series ARIMA(12,1,8)"
, cex.main= 1)
```



```
eacf(ab.res.1218)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x x x o o o o o o o o  x  o  x
## 2 x x x o x o o o o o o  x  o  x
## 3 x x x o o o o o o o o  o  o  x
## 4 x x x x o o o o o o o  o  o  x
## 5 x x x x x o x o o o o  o  o  o
## 6 x x x x o x o o o o o  o  o  x
## 7 x x x x x x x o o o o  o  o  x
```

EACF suggest ARMA(5,7), ARMA(6,7), ARMA(5,8). The possible GARCH models are GARCH(7,5), GARCH(7,6), GARCH(8,5)

```
sq.res.1218 <- res.1218^2
eacf(sq.res.1218)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x x x x o o x o x o o  o  o  o
## 2 x o x x x o o x x o o  o  o  o
## 3 x x x x x o o x x o o  o  o  o
## 4 x x x x x o o x o o o  o  o  o
## 5 x x x o x x x x o o o  o  o  o
## 6 x x x o x x o o o x o  o  o  o
## 7 x o x o x x o o o x o  o  o  o
```

EACF suggests ARMA(1,9), ARMA(4,8), ARMA(5,8), ARMA(4,9) . The possible GARCH models are GARCH(9,1), GARCH(8,4), GARCH(8,5), GARCH(9,4)

The set of possible GARCH model for residuals of ARIMA(12,1,8): {GARCH(7,5), GARCH(7,6), GARCH(8,5),GARCH(9,1), GARCH(8,4), GARCH(9,4) }

```
list_garchOrder1218 <- list(c(7,5), c(7,6), c(8,5), c(9,1), c(8,4), c(9,4))
sortAIC_garch(list_garchOrder1218, arima.1218$residuals)

##   GarchOrder      AIC
## 2       7, 6 26365.62
## 3       8, 5 26389.47
## 6       9, 4 26521.77
## 1       7, 5 26560.56
## 5       8, 4 26695.10
## 4       9, 1 27195.88
```

GARCH(7,6) has smallest AIC value in comparision with other models with residuals of ARIMA(12,1,8).

ARIMA + GARCH

Some combine possible models are considered as below

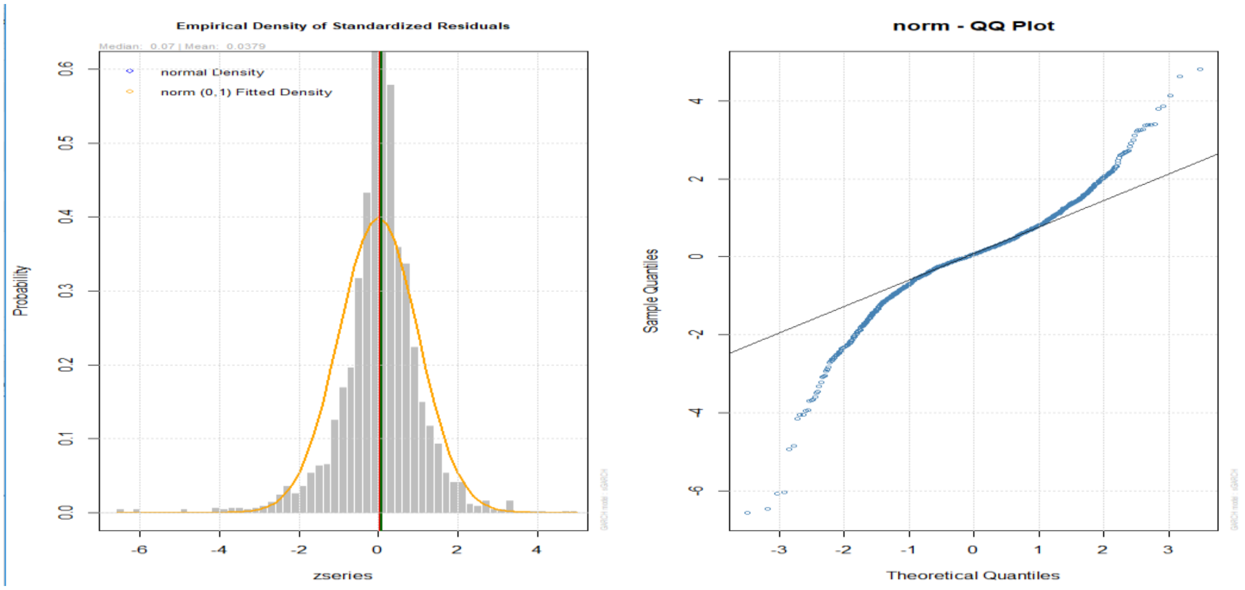
ARIMA(6,1,8)

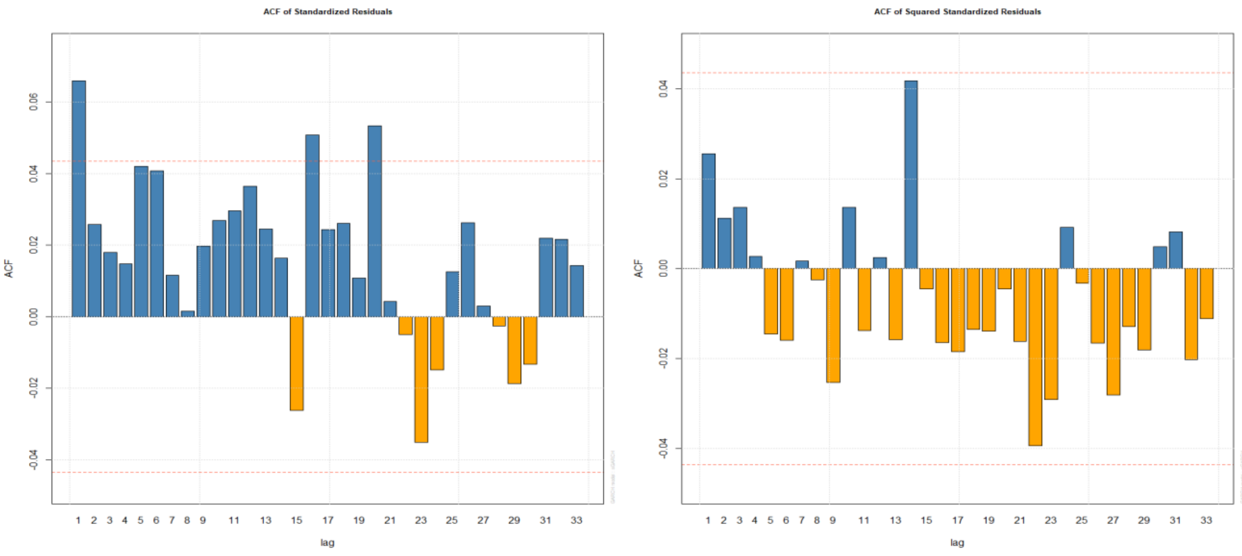
ARIMA(6,1,8) + GARCH(7,6)

```
model.618_76 <- modelfit(c(7,6), c(6,8), diff.log.bc)
model.618_76
```

```
## 
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
## 
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(7,6)
## Mean Model   : ARFIMA(6,0,8)
## Distribution : norm
## 
## Optimal Parameters
## ------------------------------------
##          Estimate  Std. Error     t value Pr(>|t|)
## ar1    -0.151572    0.032564   -4.654635 0.000003
## ar2     0.117641    0.044735    2.629716 0.008546
## ar3    -0.488581    0.029406  -16.614843 0.000000
## ar4     0.055808    0.039727    1.404775 0.160088
## ar5    -0.205361    0.042630   -4.817322 0.000001
## ar6    -0.849000    0.013807  -61.490958 0.000000
## ma1     0.148148    0.042081    3.520524 0.000431
## ma2    -0.126905    0.045978   -2.760121 0.005778
## ma3     0.499325    0.021420   23.311055 0.000000
## ma4    -0.061440    0.036243   -1.695223 0.090033
## ma5     0.174393    0.038930    4.479677 0.000007
## ma6     0.905007    0.000752 1203.518496 0.000000
## ma7    -0.004955    0.026574   -0.186460 0.852084
## ma8    -0.026451    0.023087   -1.145708 0.251916
## omega   0.000084    0.000010    8.099854 0.000000
## alpha1  0.149665    0.022259    6.723822 0.000000
## alpha2  0.046510    0.007753    5.998621 0.000000
## alpha3  0.057408    0.021811    2.632104 0.008486
## alpha4  0.000000    0.001685    0.000013 0.999990
## alpha5  0.167803    0.024220    6.928214 0.000000
## alpha6  0.000000    0.021414    0.000000 1.000000
## alpha7  0.000000    0.045593    0.000000 1.000000
## beta1   0.000000    0.150701    0.000000 1.000000
## beta2   0.000000    0.130693    0.000000 1.000000
## beta3   0.000000    0.119215    0.000000 1.000000
## beta4   0.064456    0.025830    2.495427 0.012581
## beta5   0.506628    0.050888    9.955742 0.000000
## beta6   0.000000    0.015112    0.000001 0.999999
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## ar1    -0.151572    0.036658  -4.13476 0.000036
## ar2     0.117641    0.083603   1.40715 0.159384
## ar3    -0.488581    0.057084  -8.55893 0.000000
## ar4     0.055808    0.062910   0.88710 0.375025
## ar5    -0.205361    0.085028  -2.41520 0.015726
## ar6    -0.849000    0.020142 -42.14972 0.000000
## ma1     0.148148    0.053467   2.77081 0.005592
## ma2    -0.126905    0.075026  -1.69148 0.090746
## ma3     0.499325    0.047738  10.45963 0.000000
## ma4    -0.061440    0.057752  -1.06387 0.287388
## ma5     0.174393    0.069878   2.49567 0.012572
## ma6     0.905007    0.001119 808.52598 0.000000
## ma7    -0.004955    0.031082  -0.15942 0.873341
## ma8    -0.026451    0.026028  -1.01624 0.309513
## omega   0.000084    0.000070   1.20181 0.229436
## alpha1  0.149665    0.050377   2.97089 0.002969
## alpha2  0.046510    0.055518   0.83774 0.402178
## alpha3  0.057408    0.045147   1.27158 0.203521
## alpha4  0.000000    0.074520   0.00000 1.000000
## alpha5  0.167803    0.059438   2.82315 0.004755
## alpha6  0.000000    0.149395   0.00000 1.000000
## alpha7  0.000000    0.181235   0.00000 1.000000
## beta1   0.000000    0.321297   0.00000 1.000000
## beta2   0.000000    0.543262   0.00000 1.000000
## beta3   0.000000    0.570375   0.00000 1.000000
## beta4   0.064456    0.315186   0.20450 0.837961
## beta5   0.506628    0.180456   2.80748 0.004993
## beta6   0.000000    0.340906   0.00000 1.000000
##
## LogLikelihood : 3856.091
##
## Information Criteria
## ------------------------------------
##
## Akaike       -3.7734
## Bayes        -3.6959
## Shibata      -3.7738
## Hannan-Quinn -3.7449
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                          statistic  p-value
## Lag[1]                       8.838 0.002951
## Lag[2*(p+q)+(p+q)-1][41]    36.889 0.000000
## Lag[4*(p+q)+(p+q)-1][69]    49.818 0.001138
## d.o.f=14
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                              statistic p-value
## Lag[1]                           1.326  0.2495
## Lag[2*(p+q)+(p+q)-1][38]        12.125  0.9432
## Lag[4*(p+q)+(p+q)-1][64]        19.922  0.9844
## d.o.f=13
##
```

Optimal parameters in this model shows that omega, alpha1, alpha2, alpha3, alpha5, beta4 and beta5 are siginifcant at 5% level.

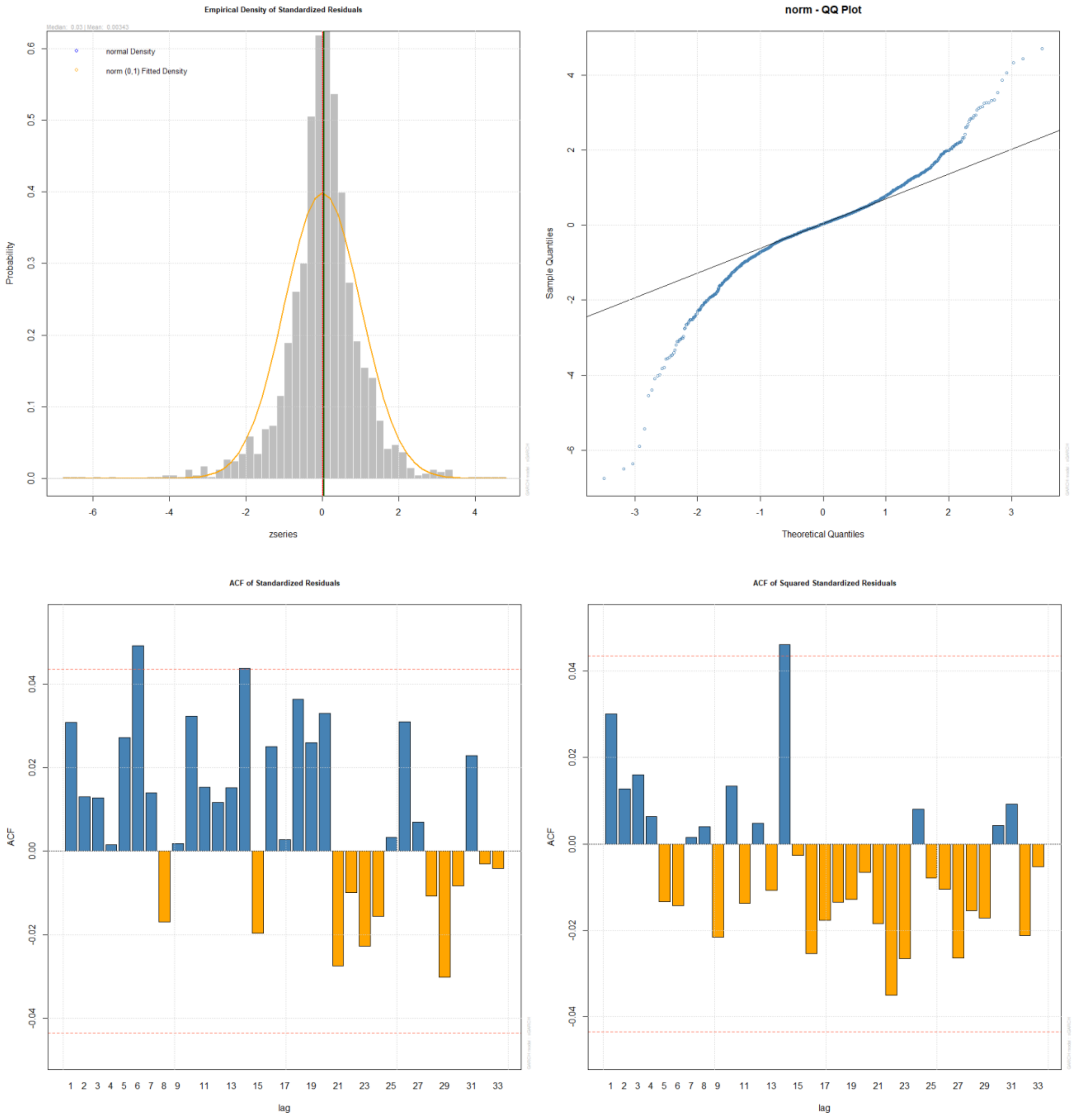




ARIMA(6,1,8) + GARCH(8,5)

```
model.618_85 <- modelfit(c(8,5), c(6,8), diff.log.bc)
model.618_85

## 
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
## 
## Conditional Variance Dynamics    
## -----------------------------------
## GARCH Model  : sGARCH(8,5)
## Mean Model   : ARFIMA(6,0,8)
## Distribution : norm 
## 
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error     t value Pr(>|t|)
## ar1    -0.541081    0.000119 -4.5393e+03 0.000000
## ar2     0.427888    0.000102  4.2098e+03 0.000000
## ar3    -0.311620    0.000083 -3.7427e+03 0.000000
## ar4     0.011588    0.000097  1.1929e+02 0.000000
## ar5     1.019275    0.000223  4.5657e+03 0.000000
## ar6     0.397696    0.000105  3.7710e+03 0.000000
## ma1     0.566747    0.000103  5.5176e+03 0.000000
## ma2    -0.416867    0.000095 -4.3959e+03 0.000000
## ma3     0.310334    0.000101  3.0735e+03 0.000000
## ma4    -0.000351    0.001049 -3.3452e-01 0.737986
## ma5    -1.048153    0.000391 -2.6805e+03 0.000000
## ma6    -0.400462    0.000176 -2.2780e+03 0.000000
## ma7    -0.005004    0.000417 -1.2012e+01 0.000000
## ma8    -0.029713    0.000127 -2.3398e+02 0.000000
## omega   0.000084    0.000041  2.0665e+00 0.038779
## alpha1  0.137185    0.025181  5.4480e+00 0.000000
## alpha2  0.049521    0.021596  2.2931e+00 0.021843
## alpha3  0.039190    0.020594  1.9030e+00 0.057042
## alpha4  0.000000    0.015349  3.0000e-06 0.999997
## alpha5  0.161144    0.029213  5.5161e+00 0.000000
## alpha6  0.000000    0.105916  0.0000e+00 1.000000
## alpha7  0.000000    0.053343  0.0000e+00 1.000000
## alpha8  0.000000    0.021971  0.0000e+00 1.000000
## beta1   0.000000    0.133192  0.0000e+00 1.000000
## beta2   0.000000    0.069725  0.0000e+00 1.000000
## beta3   0.000000    0.086530  0.0000e+00 1.000000
## beta4   0.075316    0.012566  5.9935e+00 0.000000
## beta5   0.522803    0.060268  8.6746e+00 0.000000
## 
## Robust Standard Errors:
##         Estimate  Std. Error     t value Pr(>|t|)
## ar1    -0.541081    0.001404 -3.8542e+02 0.000000
## ar2     0.427888    0.001045  4.0952e+02 0.000000
## ar3    -0.311620    0.000628 -4.9606e+02 0.000000
## ar4     0.011588    0.000148  7.8449e+01 0.000000
## ar5     1.019275    0.002797  3.6445e+02 0.000000
## ar6     0.397696    0.000846  4.7013e+02 0.000000
## ma1     0.566747    0.000054  1.0507e+04 0.000000
## ma2    -0.416867    0.000188 -2.2141e+03 0.000000
## ma3     0.310334    0.000369  8.4179e+02 0.000000
## ma4    -0.000351    0.035831 -9.7910e-03 0.992188
## ma5    -1.048153    0.006730 -1.5575e+02 0.000000
## ma6    -0.400462    0.000811 -4.9365e+02 0.000000
## ma7    -0.005004    0.027015 -1.8522e-01 0.853055
## ma8    -0.029713    0.004599 -6.4604e+00 0.000000
## omega   0.000084    0.000311  2.6943e-01 0.787596
## alpha1  0.137185    0.083171  1.6494e+00 0.099058
## alpha2  0.049521    0.084291  5.8750e-01 0.556865
## alpha3  0.039190    0.063041  6.2165e-01 0.534169
## alpha4  0.000000    0.048665  1.0000e-06 0.999999
## alpha5  0.161144    0.103171  1.5619e+00 0.118309
## alpha6  0.000000    0.889684  0.0000e+00 1.000000
## alpha7  0.000000    0.458733  0.0000e+00 1.000000
## alpha8  0.000000    0.092953  0.0000e+00 1.000000
## beta1   0.000000    0.984905  0.0000e+00 1.000000
## beta2   0.000000    0.280333  0.0000e+00 1.000000
## beta3   0.000000    0.185010  0.0000e+00 1.000000
## beta4   0.075316    0.213495  3.5278e-01 0.724254
## beta5   0.522803    0.371560  1.4070e+00 0.159413
## 
## LogLikelihood : 3863.108 
## 
## Information Criteria
## ------------------------------------
##                     
## Akaike       -3.7803
## Bayes        -3.7028
```

Optimal parameters in this model shows that omega, alpha1, alpha2, alpha5, beta4 and beta5 are siginifcant at 5% level.



Looking at Q-Q plot between two above models, there are still many points at both tails. However ACF Plots show that ARIMA(6,1,8) + GARCH(7,6) looks better than the other.

As a result, ARIMA(6,1,8) + GARCH(7,6) is the suitable model for residuals of ARIMA(6,1,8).

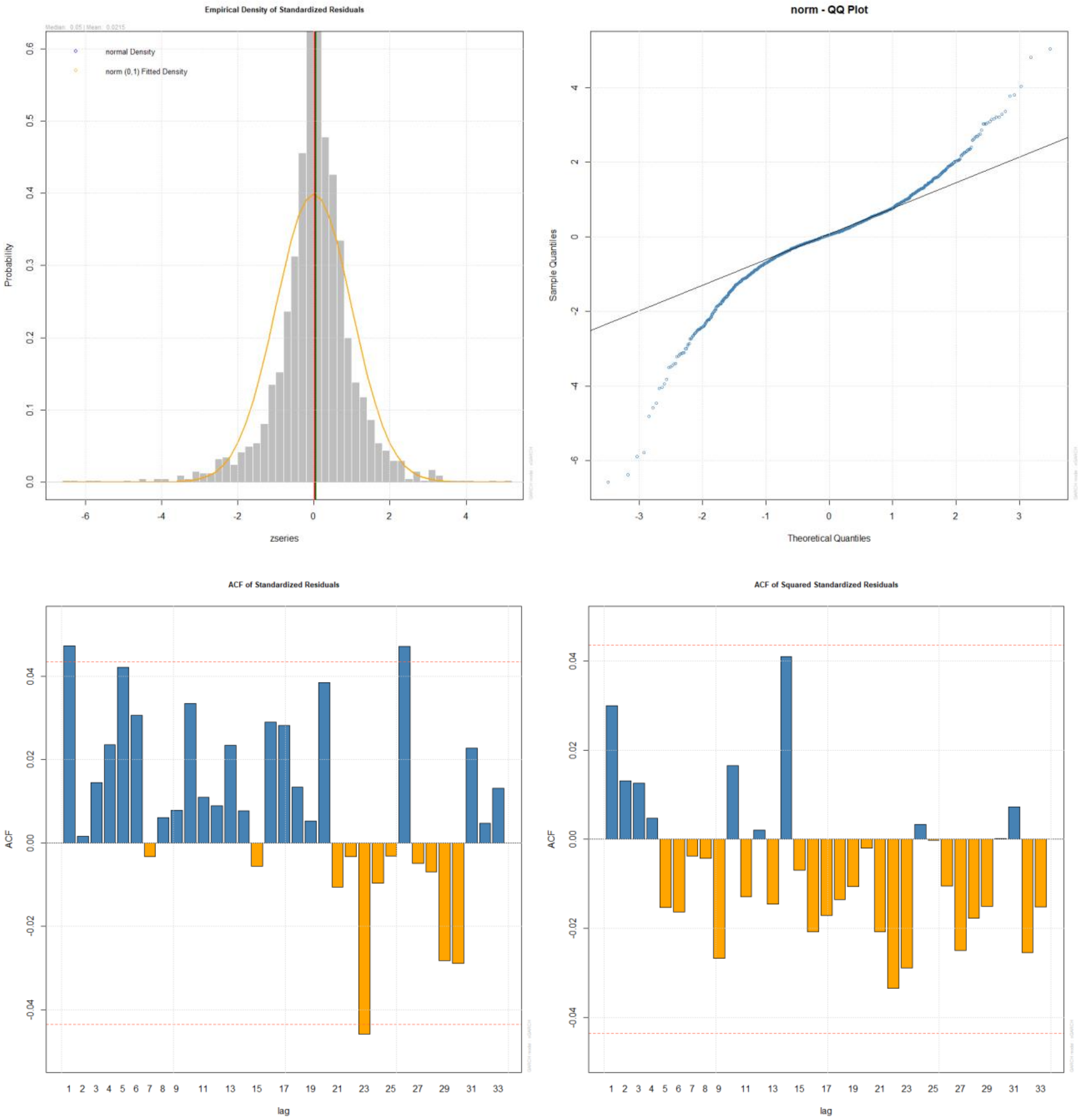
ARIMA(12,1,8)

ARIMA(12,1,8) + GARCH(7,5)

```
model.1218_75 <- modelfit(c(7,5), c(12,8), diff.log.bc)
model.1218_75
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(7,5)
## Mean Model   : ARFIMA(12,0,8)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error     t value Pr(>|t|)
## ar1     0.318825    0.023811    13.38989 0.000000
## ar2    -0.912131    0.021884   -41.67957 0.000000
## ar3     0.768702    0.026262    29.27019 0.000000
## ar4    -0.496245    0.030571   -16.23269 0.000000
## ar5     0.902116    0.026042    34.64058 0.000000
## ar6    -0.237092    0.031407    -7.54914 0.000000
## ar7     0.348885    0.031304    11.14496 0.000000
## ar8     0.265777    0.028491     9.32855 0.000000
## ar9    -0.060022    0.026587    -2.25755 0.023974
## ar10    0.038951    0.030054     1.29604 0.194961
## ar11   -0.044153    0.020279    -2.17726 0.029461
## ar12    0.038659    0.020288     1.90553 0.056711
## ma1    -0.298547    0.001199  -248.93877 0.000000
## ma2     0.928335    0.000115  8082.09670 0.000000
## ma3    -0.743552    0.008359   -88.94787 0.000000
## ma4     0.475362    0.000211  2250.77294 0.000000
## ma5    -0.914446    0.000139 -6595.46535 0.000000
## ma6     0.271511    0.005816    46.68351 0.000000
## ma7    -0.371798    0.005636   -65.97150 0.000000
## ma8    -0.231062    0.001758  -131.41794 0.000000
## omega   0.000087    0.000031     2.78856 0.005294
## alpha1  0.145549    0.024077     6.04508 0.000000
## alpha2  0.059584    0.018995     3.13685 0.001708
## alpha3  0.061059    0.022041     2.77031 0.005600
## alpha4  0.000000    0.004453     0.00001 0.999992
## alpha5  0.183687    0.035478     5.17749 0.000000
## alpha6  0.000000    0.063537     0.00000 1.000000
## alpha7  0.000000    0.031865     0.00000 1.000000
## beta1   0.000000    0.050608     0.00000 1.000000
## beta2   0.000000    0.186844     0.00000 1.000000
## beta3   0.000000    0.143345     0.00000 1.000000
## beta4   0.062073    0.003810    16.29164 0.000000
## beta5   0.486550    0.025464    19.10729 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error     t value Pr(>|t|)
## ar1     0.318825    0.031630  1.0080e+01 0.000000
## ar2    -0.912131    0.027026 -3.3750e+01 0.000000
## ar3     0.768702    0.032909  2.3359e+01 0.000000
## ar4    -0.496245    0.033433 -1.4843e+01 0.000000
## ar5     0.902116    0.037336  2.4162e+01 0.000000
## ar6    -0.237092    0.036486 -6.4982e+00 0.000000
## ar7     0.348885    0.035272  9.8913e+00 0.000000
## ar8     0.265777    0.042802  6.2095e+00 0.000000
## ar9    -0.060022    0.041025 -1.4631e+00 0.143450
## ar10    0.038951    0.044739  8.7062e-01 0.383963
## ar11   -0.044153    0.028495 -1.5495e+00 0.121265
## ar12    0.038659    0.027381  1.4119e+00 0.157983
## ma1    -0.298547    0.001158 -2.5782e+02 0.000000
## ma2     0.928335    0.000216  4.2911e+03 0.000000
## ma3    -0.743552    0.010382 -7.1618e+01 0.000000
## ma4     0.475362    0.000458  1.0379e+03 0.000000
## ma5    -0.914446    0.000176 -5.2023e+03 0.000000
## ma6     0.271511    0.007569  3.5873e+01 0.000000
## ma7    -0.371798    0.004536 -8.1973e+01 0.000000
## ma8    -0.231062    0.002068 -1.1175e+02 0.000000
## omega   0.000087    0.000112  7.7961e-01 0.435622
## alpha1  0.145549    0.040709  3.5754e+00 0.000350
## alpha2  0.059584    0.089228  6.6777e-01 0.504282
## alpha3  0.061059    0.081664  7.4769e-01 0.454648
## alpha4  0.000000    0.085970  1.0000e-06 1.000000
## alpha5  0.183687    0.109046  1.6845e+00 0.092087
## alpha6  0.000000    0.290952  0.0000e+00 1.000000
## alpha7  0.000000    0.288467  0.0000e+00 1.000000
## beta1   0.000000    0.780766  0.0000e+00 1.000000
## beta2   0.000000    0.827476  0.0000e+00 1.000000
## beta3   0.000000    0.699257  0.0000e+00 1.000000
## beta4   0.062073    0.241548  2.5698e-01 0.797193
## beta5   0.486550    0.181250  2.6844e+00 0.007266
##
```

Optimal parameters in this model shows that omega, alpha1, alpha2, alpha3, alpha5, beta4 and beta5 are significant at 5% level.

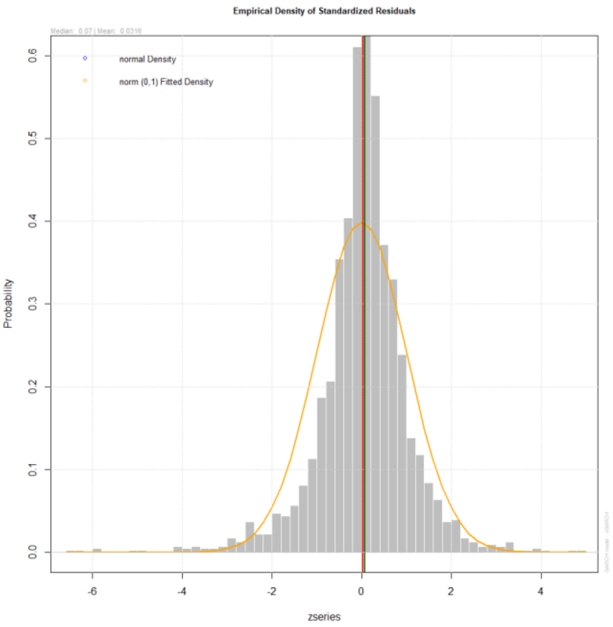


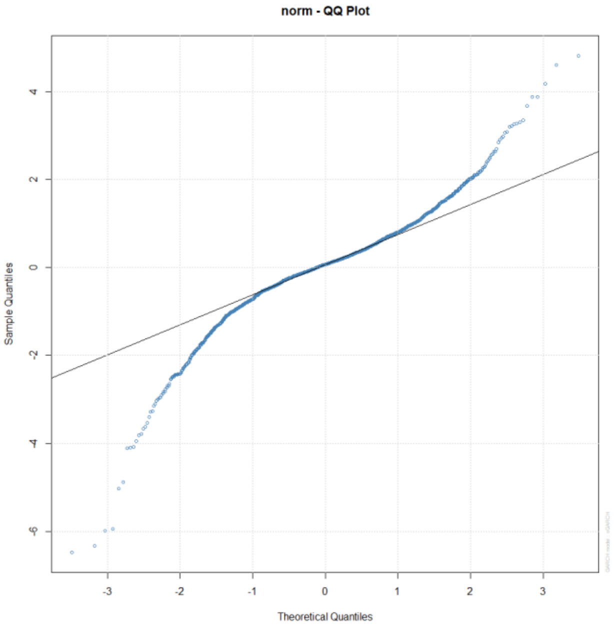
ARIMA(12,1,8) + GARCH(8,5)

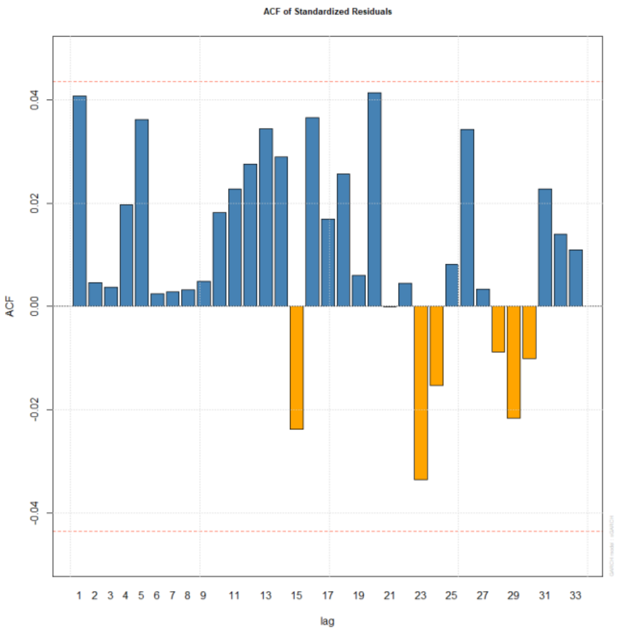
```
model.1218_85<- modelfit(c(8,5), c(12,8), diff.log.bc)
model.1218_85
```

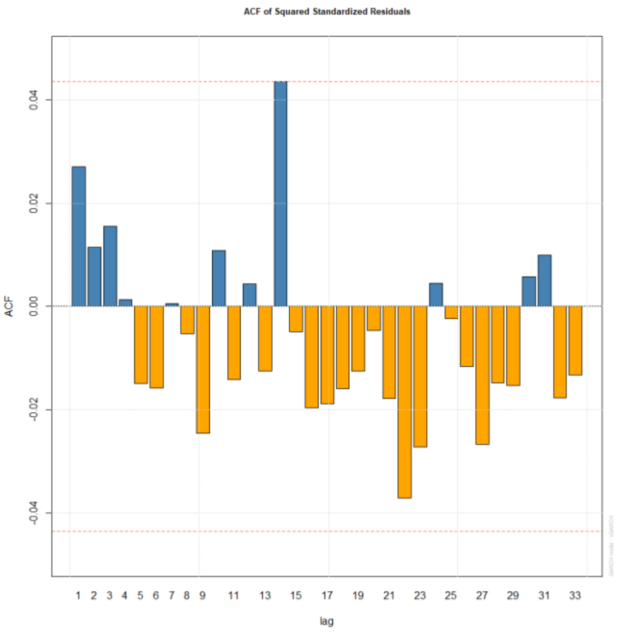
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(8,5)
## Mean Model   : ARFIMA(12,0,8)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error     t value Pr(>|t|)
## ar1    -0.177018    0.105555   -1.677026 0.093537
## ar2    -0.476569    0.027320  -17.443757 0.000000
## ar3    -0.757160    0.061515  -12.308632 0.000000
## ar4    -0.089520    0.063729   -1.404699 0.160111
## ar5    -0.546938    0.056996   -9.596035 0.000000
## ar6    -0.801475    0.034768  -23.052312 0.000000
## ar7    -0.010695    0.100408   -0.106519 0.915170
## ar8    -0.306216    0.047565   -6.437877 0.000000
## ar9     0.071575    0.033833    2.115534 0.034384
## ar10    0.007715    0.026229    0.294139 0.768651
## ar11    0.041739    0.024148    1.728484 0.083901
## ar12    0.057338    0.027191    2.108755 0.034966
## ma1     0.201010    0.101706    1.976380 0.048112
## ma2     0.495956    0.014254   34.795074 0.000000
## ma3     0.798430    0.054596   14.624376 0.000000
## ma4     0.102806    0.054361    1.891196 0.058598
## ma5     0.553493    0.048565   11.396963 0.000000
## ma6     0.928562    0.000353 2626.842856 0.000000
## ma7     0.038155    0.086943    0.438848 0.660772
## ma8     0.359016    0.036827    9.748733 0.000000
## omega   0.000088    0.000047    1.872452 0.061144
## alpha1  0.149495    0.024541    6.091662 0.000000
## alpha2  0.047620    0.027591    1.725915 0.084363
## alpha3  0.062021    0.028711    2.160181 0.030759
## alpha4  0.000000    0.014576    0.000002 0.999998
## alpha5  0.170017    0.036566    4.649633 0.000003
## alpha6  0.000000    0.110468    0.000000 1.000000
## alpha7  0.000000    0.042314    0.000000 1.000000
## alpha8  0.000000    0.025711    0.000000 1.000000
## beta1   0.000000    0.151663    0.000000 1.000000
## beta2   0.000000    0.062547    0.000000 1.000000
## beta3   0.000000    0.107703    0.000000 1.000000
## beta4   0.059888    0.031857    1.879899 0.060122
## beta5   0.503893    0.056554    8.909974 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## ar1    -0.177018    0.227525  -0.778017 0.436559
## ar2    -0.476569    0.037659 -12.654732 0.000000
## ar3    -0.757160    0.142086  -5.328878 0.000000
## ar4    -0.089520    0.117956  -0.758933 0.447893
## ar5    -0.546938    0.113011  -4.839696 0.000001
## ar6    -0.801475    0.059514 -13.466984 0.000000
## ar7    -0.010695    0.335074   -0.031919 0.974536
## ar8    -0.306216    0.081377   -3.762934 0.000168
## ar9     0.071575    0.064310    1.112966 0.265723
## ar10    0.007715    0.056141    0.137419 0.890699
## ar11    0.041739    0.028173    1.481514 0.138470
## ar12    0.057338    0.055927    1.025236 0.305252
## ma1     0.201010    0.211228    0.951622 0.341289
## ma2     0.495956    0.019234   25.786004 0.000000
## ma3     0.798430    0.153346    5.206721 0.000000
## ma4     0.102806    0.085122    1.207760 0.227140
## ma5     0.553493    0.101393    5.458900 0.000000
## ma6     0.928562    0.001136  817.291226 0.000000
## ma7     0.038155    0.248327    0.153647 0.877888
## ma8     0.359016    0.071860    4.996082 0.000001
## omega   0.000088    0.000274    0.322405 0.747146
## alpha1  0.149495    0.042632    3.506632 0.000454
## alpha2  0.047620    0.109143    0.436307 0.662614
## alpha3  0.062021    0.084150    0.737032 0.461103
## alpha4  0.000000    0.072304    0.000000 1.000000
## alpha5  0.170017    0.152242    1.116760 0.264097
## alpha6  0.000000    0.652665    0.000000 1.000000
## alpha7  0.000000    0.295622    0.000000 1.000000
## alpha8  0.000000    0.150127    0.000000 1.000000
## beta1   0.000000    0.798105    0.000000 1.000000
## beta2   0.000000    0.307960    0.000000 1.000000
## beta3   0.000000    0.256563    0.000000 1.000000
## beta4   0.059888    0.145743    0.410914 0.681135
## beta5   0.503893    0.264128    1.907758 0.056422
```

Optimal parameters in this model shows that alpha1, alpha3, alpha5 and beta5 are significant at 5% level.





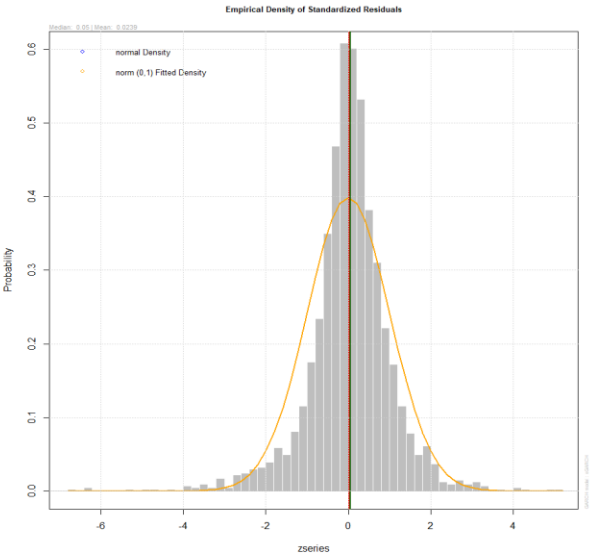


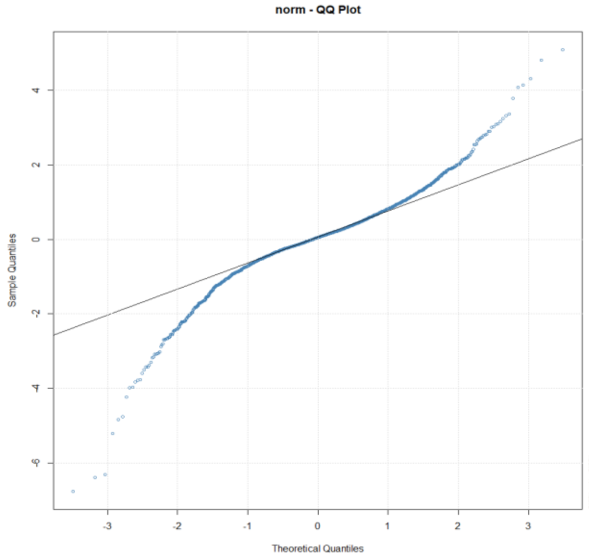


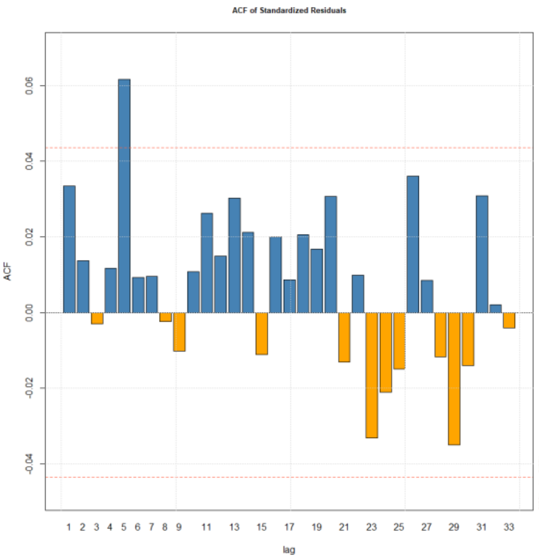
ARIMA(12,1,8) + GARCH(7,6)

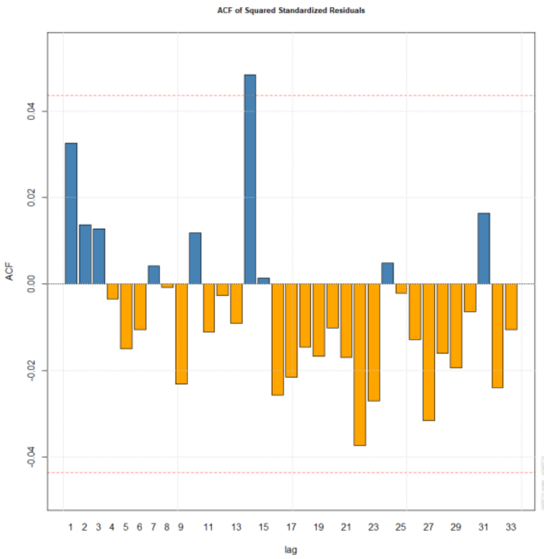
```
model.1218_76 <- modelfit(c(7,6), c(12,8), diff.log.bc)
model.1218_76

##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(7,6)
## Mean Model   : ARFIMA(12,0,8)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## ar1     0.115408    0.000028  4.1551e+03 0.000000
## ar2     0.816645    0.000105  7.7834e+03 0.000000
## ar3    -0.618975    0.000084 -7.3577e+03 0.000000
## ar4    -0.248550    0.000043 -5.7852e+03 0.000000
## ar5     0.389884    0.000060  6.5499e+03 0.000000
## ar6    -0.178599    0.000035 -5.1499e+03 0.000000
## ar7     0.370237    0.000062  5.9426e+03 0.000000
## ar8     0.234662    0.000044  5.3701e+03 0.000000
## ar9     0.045525    0.000018  2.4671e+03 0.000000
## ar10    0.071095    0.000023  3.1447e+03 0.000000
## ar11   -0.056466    0.000021 -2.7394e+03 0.000000
## ar12    0.003468    0.000009  3.9663e+02 0.000000
## ma1    -0.092192    0.000037 -2.4951e+03 0.000000
## ma2    -0.819347    0.000121 -6.7630e+03 0.000000
## ma3     0.639333    0.000100  6.4234e+03 0.000000
## ma4     0.258129    0.000058  4.4678e+03 0.000000
## ma5    -0.463171    0.000082 -5.6335e+03 0.000000
## ma6     0.297329    0.000069  4.3118e+03 0.000000
## ma7    -0.341556    0.000078 -4.3638e+03 0.000000
## ma8    -0.383115    0.000074 -5.1741e+03 0.000000
## omega   0.000064    0.000004  1.4843e+01 0.000000
## alpha1  0.133884    0.010987  1.2186e+01 0.000000
## alpha2  0.054275    0.022633  2.3981e+00 0.016482
## alpha3  0.051108    0.004346  1.1759e+01 0.000000
## alpha4  0.000131    0.005244  2.5074e-02 0.979996
## alpha5  0.167232    0.016632  1.0055e+01 0.000000
## alpha6  0.000099    0.044712  2.2210e-03 0.998228
## alpha7  0.000073    0.048336  1.5110e-03 0.998795
## beta1   0.000191    0.254705  7.5100e-04 0.999401
## beta2   0.000129    0.205221  6.2600e-04 0.999500
## beta3   0.000154    0.196459  7.8600e-04 0.999373
## beta4   0.062904    0.091038  6.9097e-01 0.489588
## beta5   0.528089    0.035299  1.4960e+01 0.000000
## beta6   0.000148    0.166509  8.9200e-04 0.999289
## 
## Robust Standard Errors:
##          Estimate  Std. Error     t value Pr(>|t|)
## ar1     0.115408    0.000121  9.5494e+02  0.00000
## ar2     0.816645    0.001246  6.5549e+02  0.00000
## ar3    -0.618975    0.001234 -5.0169e+02  0.00000
## ar4    -0.248550    0.000169 -1.4709e+03  0.00000
## ar5     0.389884    0.000361  1.0811e+03  0.00000
## ar6    -0.178599    0.000123 -1.4539e+03  0.00000
## ar7     0.370237    0.000150  2.4683e+03  0.00000
## ar8     0.234662    0.000163  1.4390e+03  0.00000
## ar9     0.045525    0.000066  6.9498e+02  0.00000
## ar10    0.071095    0.000091  7.8545e+02  0.00000
## ar11   -0.056466    0.000076 -7.3825e+02  0.00000
## ar12    0.003468    0.000012  2.9443e+02  0.00000
## ma1    -0.092192    0.000129 -7.1656e+02  0.00000
## ma2    -0.819347    0.000479 -1.7117e+03  0.00000
## ma3     0.639333    0.000393  1.6257e+03  0.00000
## ma4     0.258129    0.000151  1.7114e+03  0.00000
## ma5    -0.463171    0.000207 -2.2357e+03  0.00000
## ma6     0.297329    0.000313  9.5035e+02  0.00000
## ma7    -0.341556    0.000356 -9.5830e+02  0.00000
## ma8    -0.383115    0.000255 -1.5003e+03  0.00000
## omega   0.000064    0.000082  7.9022e-01  0.42940
## alpha1  0.133884    0.099640  1.3437e+00  0.17905
## alpha2  0.054275    0.100596  5.3954e-01  0.58952
## alpha3  0.051108    0.051733  9.8792e-01  0.32319
## alpha4  0.000131    0.078701  1.6710e-03  0.99867
## alpha5  0.167232    0.102870  1.6257e+00  0.10402
## alpha6  0.000099    0.279021  3.5600e-04  0.99972
## alpha7  0.000073    0.333197  2.1900e-04  0.99982
## beta1   0.000191    0.703202  2.7200e-04  0.99978
## beta2   0.000129    1.217294  1.0600e-04  0.99992
## beta3   0.000154    1.177146  1.3100e-04  0.99989
## beta4   0.062904    0.567052  1.1093e-01  0.91167
## beta5   0.528089    0.075630  6.9825e+00  0.00000
## beta6   0.000148    0.687601  2.1600e-04  0.99983
##
```

Optimal parameters in this model shows that omega, alpha1, alpha2, alpha3, alpha5 and beta5 are significant at 5% level.









Looking at Q-Q plot between three above models, there are still many points at both tails. However ACF Plots show that ARIMA(12,1,8) + GARCH(8,5) looks better than the other. Furthermore, ARIMA(12,1,8) + GARCH(7,5) has number of significant paramters larger than these one in other model.

As a result, ARIMA(12,1,8) + GARCH(7,5) and ARIMA(12,1,8) + GARCH(8,5) are the suitable models for residuals of ARIMA(12,1,8)

Validation

In this step, three models ARIMA(6,1,8) + GARCH(7,6) , ARIMA(12,1,8) + GARCH(7,5) and ARIMA(12,1,8) + GARCH(8,5) are considered to validate by using MASE for forecasting values.

```
actual_result <- read.csv("Bitcoin_Prices_Forecasts.csv")
observed <- as.numeric(actual_result$Closing.price)
```

ARIMA(6,1,8) + GARCH(7,6)

```
checkMASE(model.618_76, diff.log.bc, log.bc, observed)

##       MASE
## 1 1.356283
```

ARIMA(12,1,8) + GARCH(7,5)

```
checkMASE(model.1218_75, diff.log.bc, log.bc, observed)

##       MASE
## 1 2.396508
```

ARIMA(12,1,8) + GARCH(8,5)

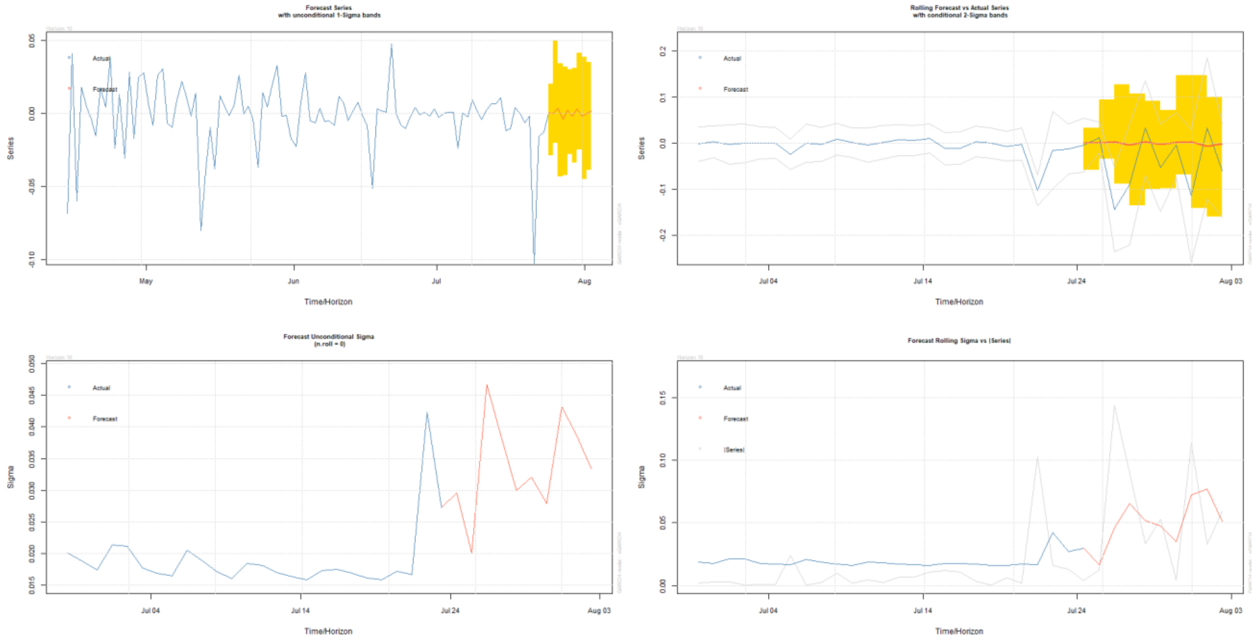
```
checkMASE(model.1218_85, diff.log.bc, log.bc, observed)

##       MASE
## 1 4.892222
```

MASE of ARIMA(6,1,8) + GARCH(7,6) is smaller than MASE of other models. Therefore ARIMA(6,1,8) + GARCH(7,6) is the good one

CONCLUSION

Model ARIMA(6,1,8) + GARCH (7,6) is the best model. Its predicted values for next 10 days as below

```
forcRes = ugarchforecast(model.618_76, data = data_model, n.ahead = 10)
data.frame(fitted.values(forcRes, log.bc))

##    fitted.values.forcRes..log.bc.
## 1                        3812.285
## 2                        3813.473
## 3                        3827.240
## 4                        3825.148
## 5                        3830.352
## 6                        3827.937
## 7                        3836.334
## 8                        3837.198
## 9                        3837.165
## 10                       3842.687
```



Forecast Series
with unconditional 1-Sigma bands

Rolling Forecast vs Actual Series
with conditional 2-Sigma bands

Forecast Unconditional Sigma
(n.roll = 0)

Forecast Rolling Sigma vs |Series|

MASE of ARIMA(6,1,8) + GARCH(7,6) is 1.356283