Forecasting For Bitcoin price

MATH 1318 Time Series Competitive project (Bitcoin data)

Mohammad Mamun(s3571301)

29th May,2018

Table of Contents

[Data Set 2](#_Toc515481648)

[Loading Data and Visual observation of data 2](#_Toc515481649)

[Visual Observation of Graph 2](#_Toc515481650)

[ACF and PACF Analysis 3](#_Toc515481651)

[Diagnostic test and Differencing 3](#_Toc515481652)

[ACF And PACF for transformed series 7](#_Toc515481653)

[Absolute value and square transformations to figure out this ARCH effect. 7](#_Toc515481654)

[##Squaring Transformation 8](#_Toc515481655)

[Diagonistic Check 24](#_Toc515481656)

[Overfitting models- 26](#_Toc515481657)

[overfitting model with GARCH part 31](#_Toc515481658)

[Forecast 37](#_Toc515481659)

[Model Accuracy check (MASE) 37](#_Toc515481660)

[Conclusion 37](#_Toc515481661)

[Appendix 38](#_Toc515481662)

Introduction

The objective of this project was to find a suitable model to do forecasting for Bitcoin time series data for next10 days. The data sets were sourced from the [coin marke](https://coinmarketcap.com/)t. In this project we will have building model, parameter estimation and diagnostic check, forecasting and accuracy check.

# Data Set

The [coin market](https://coinmarketcap.com/) provides two data set for bitcoin time series, one dataset we will use use for building model which has 1772 observations and another one is for forecasting and it has 10 observations.

The data is the daily bitcoin closing price data from 27/04/2013 to 03/03/2018. Bitcoin is a cryptocurrency with no physical note denomination. Bitcoin data has been very volatile with sharp rises/falls on daily basis, a quick internet search will show many headlines on the volatility of Bitcoin.

## Packages

We will use following Rpackages for this project

knitr::opts\_chunk$set(echo = TRUE)  
library(TSA)  
library(fUnitRoots)  
library(forecast)  
library(CombMSC)  
library(lmtest)  
library(fGarch)  
library(rugarch)  
library(truncnorm)  
library(forecast)  
library(truncnorm)  
library(FitAR)

## Loading Data and Visual observation of data

W read the data set by using following code,then we need to convert this data set to a time series object

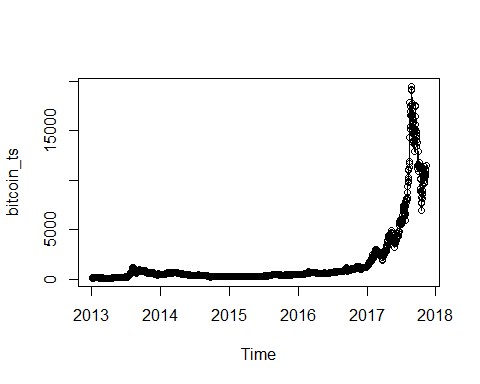
bitcoin<- read.csv("Bitcoin\_Historical\_Price.csv")$Close  
bit\_forecast<-read.csv("Bitcoin\_Prices\_Forecasts.csv")$Closing.price  
  
bitcoin\_ts= ts(bitcoin,start=c(2013,4,27),frequency=365)  
bitcoin\_ts\_nxt= ts(bit\_forecast,start=c(2018,3,4),frequency=365)

# Visual Observation of Graph

From visual observation of graph, we can see that from 2013 to 2017 there is a trend but no seasonality. From early 2017 there is a trend to mid 2017 with rapid rise in price of bitcoin price to almost 20,000. From mid 2017 there is a big drop in price to about 6,000 and a trend.

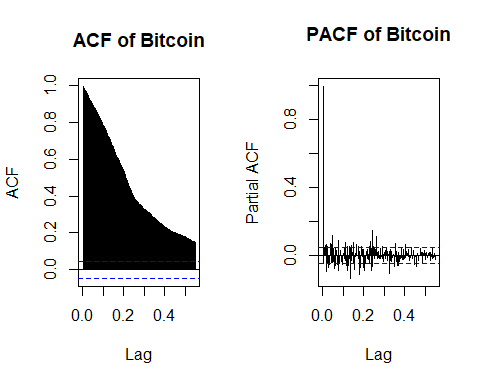
Further visual observation of the plot indicates no autoregressive behaviour but there is changing variance and as mentioned no seasonality throughout the Bitcoin time series data.

par(mfrow=c(1,1))  
plot.ts(bitcoin\_ts,yax.flip=T,type='o')



## ACF and PACF Analysis

par(mfrow=c(1,2))  
acf(bitcoin\_ts,lag.max = 200,main="ACF of Bitcoin")  
pacf(bitcoin\_ts,lag.max = 200,main="PACF of Bitcoin")



## ACF and PACF Plot Analysis

Visual observation of the ACF plot indicates a decaying pattern which points to a trend in this series.

Upon observation of the PACF indicicates one significant correllation in the first leg which confirms the existence of trend in this data set.

y=bitcoin\_ts  
x=zlag(bitcoin\_ts)  
index = 2:length(x)   
cor(y[index],x[index])

## [1] 0.9970915

##Correlation coeficient : 0.9970915.

**##Observation of correllation**

Upon observation of the correllation score of 99.7% indicates strong correllation from the previous days closing price.

## Diagnostic test and Differencing

##adf test--  
adf.test(bitcoin\_ts)

##   
## Augmented Dickey-Fuller Test  
##   
## data: bitcoin\_ts  
## Dickey-Fuller = -1.6976, Lag order = 12, p-value = 0.7063  
## alternative hypothesis: stationary

##Transformation  
#bitcoin\_ts\_log = log(bitcoin\_ts)  
#bitcoin\_ts\_log  
bitcoin\_ts\_d1 = diff(log(bitcoin\_ts))  
adf.test(bitcoin\_ts\_d1, alternative = "stationary")

## Warning in adf.test(bitcoin\_ts\_d1, alternative = "stationary"): p-value  
## smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: bitcoin\_ts\_d1  
## Dickey-Fuller = -10.167, Lag order = 12, p-value = 0.01  
## alternative hypothesis: stationary

## Analysis of Dickey Fuller results and Differencing

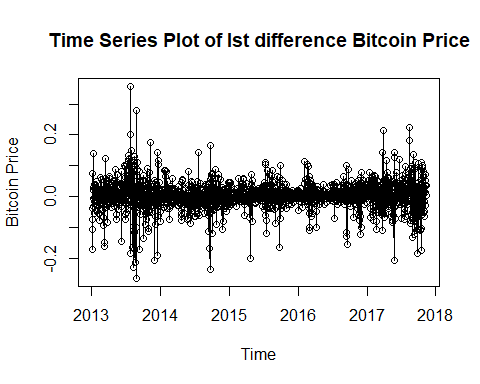
With a p value of 0.7063 from Dickey Fuller we can not reject the null hypothesis and we can say the series is non-stationary. We have transformend the series with log transformation that helped to stabilize the series, Since there is still changing of variance exists in the series we have done differencing.

After we have done the first differencing, ADF test confirms the series is stationary.

The ADF test is statistically significant (at 5% significance) which indicates Stationarity of the series after first differencing

**##Visual observation of first difference**

plot(bitcoin\_ts\_d1,type='o',ylab= 'Bitcoin Price',  
 main="Time Series Plot of Ist difference Bitcoin Price")



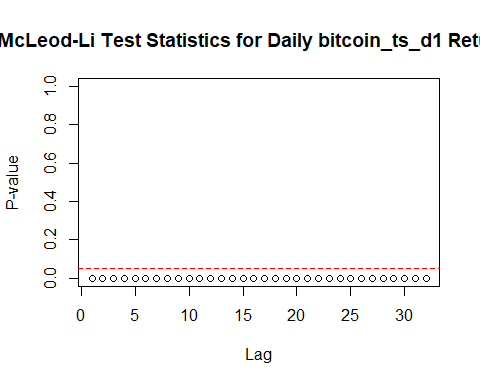
**## Observation of the First Log Transformation Difference**

The time series plot shows a change in variance in various parts of the graph. There is no seasonality as the p value as mentioned was statistically significant.

We can see that there is a volatile clustering in the series, as larger variance is followed by the smaller variance and vice versa.

We are performing a mcleod test to check whether there is a volatile clustering in the series

McLeod.Li.test(y=bitcoin\_ts\_d1,main="McLeod-Li Test Statistics for Daily bitcoin\_ts\_d1 Returns")

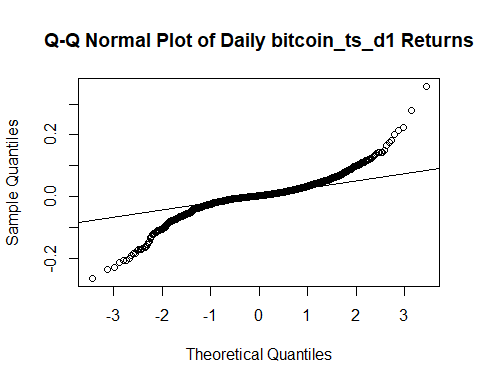


**## Mcleod-Li test analysis**

McLeod-Li test is significnat at 5% level of significance for all lags. This gives a strong idea about existence of volatiliy clustering.

**##QQplot Analysis**

qqnorm(bitcoin\_ts\_d1,main="Q-Q Normal Plot of Daily bitcoin\_ts\_d1 Returns")  
qqline(bitcoin\_ts\_d1)

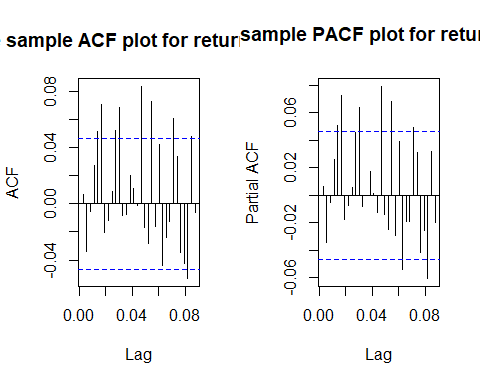


**## Analysis of QQplot**

Fat tails is in accordance with heteroscadasticity exists within the observations

## ACF And PACF for transformed series

par(mfrow=c(1,2))  
acf(bitcoin\_ts\_d1, main="The sample ACF plot for return series")  
pacf(bitcoin\_ts\_d1, main="The sample PACF plot for return series")



eacf(bitcoin\_ts\_d1)

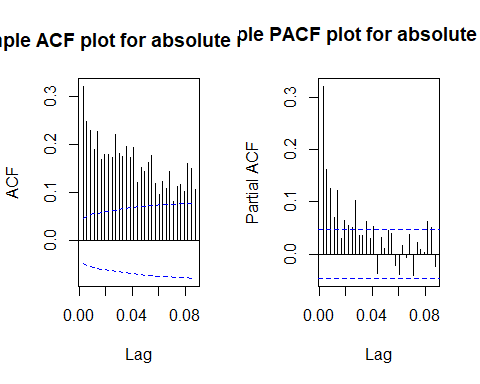
## AR/MA  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13  
## 0 o o o o x x o o o x x o o o   
## 1 x o o o o x o o o o x o o o   
## 2 x x o o o x o o o o x o o o   
## 3 x x x o o x o o o o x o o o   
## 4 x x o x o x o o o o x o o o   
## 5 x x x x x o o o o o x o o o   
## 6 x x o x x x o o o o o o o o   
##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

**## Analysis of Plots**

EACF all shows pattern of white noise for the correlation structure. However, there is an ARCH effect present in the series. From the EACF, we can identify ARMA(1,1), ARMA(2,2), and ARMA(3,3) models.

## Absolute value and square transformations to figure out this ARCH effect.

abs.bitcoin\_ts\_d1 = abs(bitcoin\_ts\_d1)  
sq.bitcoin\_ts\_d1 = bitcoin\_ts\_d1^2  
par(mfrow=c(1,2))  
acf(abs.bitcoin\_ts\_d1, ci.type="ma",main="The sample ACF plot for absolute return series")  
pacf(abs.bitcoin\_ts\_d1, main="The sample PACF plot for absolute return series")



eacf(abs.bitcoin\_ts\_d1)

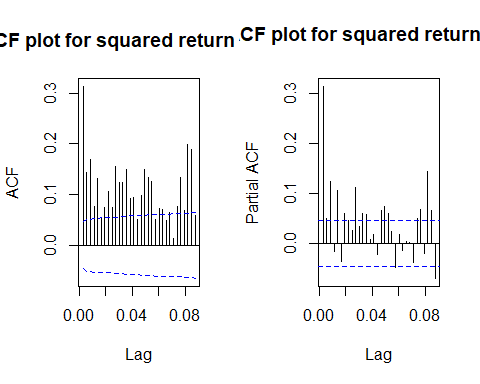
## AR/MA  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13  
## 0 x x x x x x x x x x x x x x   
## 1 x o o x x x o o o x o o o o   
## 2 x x o o o o o o o x o o o o   
## 3 x x x o o o o o o x o o o o   
## 4 x x x x o o o o o x o o o o   
## 5 x x x x x o o o o o o o o o   
## 6 x x x x x o o o o o o o o o   
##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

**## Analysis of Plots**

After the absolute value transformation, we observe many signficicant lags in both ACF and PACF. Also, EACF do not suggest an ARMA(0,0) model. From the EACF, we can identify ARMA(1,1), ARMA(2,2), and ARMA(3,3) models for absolute value series. These models correspond to parameter settings of [max(2,2),2], [max(1,1),1] and [max(3,3),3]. So the corresponding tentative GARCH models are GARCH(1,1), GARCH(2,2), GARCH(3,3).

## ##Squaring Transformation

par(mfrow=c(1,2))  
acf(sq.bitcoin\_ts\_d1, ci.type="ma",main="ACF plot for squared return series")  
pacf(sq.bitcoin\_ts\_d1, main="PACF plot for squared return series")



eacf(sq.bitcoin\_ts\_d1)

## 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 x o x o o x o   
## 2 x x o o o x o o o x o o x o   
## 3 x x x o o x o o o x o o x o   
## 4 x x x o o o o o o o o o x o   
## 5 x x x o x x o o o o o o o o   
## 6 x x x x x x o o o o o o o o   
## 7 x x x o x x x x o o o o o o

**##Analysis of Square Transformation**

After the square transformation, we boserve many signficicant lags in both ACF and PACF. Also, EACF do not suggest an ARMA(0,0) model. From the EACF, we can identify ARMA(2,2),ARMA(3,3), and ARMA(2,3) models for squared series. These models correspond to parameter settings of [max(2,2),2], [max(3,3),3], [max(2,3),2]. So the corresponding tentative GARCH models are GARCH(2,2), GARCH(3,3), GARCH(3,2).

mod.22 = garch(bitcoin\_ts\_d1,order=c(2,2),trace = FALSE)

## Warning in sqrt(pred$e): NaNs produced

summary(mod.22)

All the coefficients but a2 are significant at 5% level of significance.

##   
## Call:  
## garch(x = bitcoin\_ts\_d1, order = c(2, 2), trace = FALSE)  
##   
## Model:  
## GARCH(2,2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.74488 -0.33250 0.06947 0.53368 4.60743   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## a0 4.863e-05 6.861e-06 7.088 1.36e-12 \*\*\*  
## a1 2.019e-01 1.825e-02 11.059 < 2e-16 \*\*\*  
## a2 1.973e-02 2.304e-02 0.856 0.392   
## b1 1.876e-01 8.065e-02 2.326 0.020 \*   
## b2 5.942e-01 6.731e-02 8.827 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 5506.8, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 0.026848, df = 1, p-value = 0.8698

mod.22\_2 = garchFit(formula = ~garch(2,2), data =bitcoin\_ts\_d1 )

##   
## Series Initialization:  
## ARMA Model: arma  
## Formula Mean: ~ arma(0, 0)  
## GARCH Model: garch  
## Formula Variance: ~ garch(2, 2)  
## ARMA Order: 0 0  
## Max ARMA Order: 0  
## GARCH Order: 2 2  
## Max GARCH Order: 2  
## Maximum Order: 2  
## Conditional Dist: norm  
## h.start: 3  
## llh.start: 1  
## Length of Series: 1771  
## Recursion Init: mci  
## Series Scale: 0.04499265  
##   
## Parameter Initialization:  
## Initial Parameters: $params  
## Limits of Transformations: $U, $V  
## Which Parameters are Fixed? $includes  
## Parameter Matrix:  
## U V params includes  
## mu -0.55869489 0.5586949 0.05586949 TRUE  
## omega 0.00000100 100.0000000 0.10000000 TRUE  
## alpha1 0.00000001 1.0000000 0.05000000 TRUE  
## alpha2 0.00000001 1.0000000 0.05000000 TRUE  
## gamma1 -0.99999999 1.0000000 0.10000000 FALSE  
## gamma2 -0.99999999 1.0000000 0.10000000 FALSE  
## beta1 0.00000001 1.0000000 0.40000000 TRUE  
## beta2 0.00000001 1.0000000 0.40000000 TRUE  
## delta 0.00000000 2.0000000 2.00000000 FALSE  
## skew 0.10000000 10.0000000 1.00000000 FALSE  
## shape 1.00000000 10.0000000 4.00000000 FALSE  
## Index List of Parameters to be Optimized:  
## mu omega alpha1 alpha2 beta1 beta2   
## 1 2 3 4 7 8   
## Persistence: 0.9   
##   
##   
## --- START OF TRACE ---  
## Selected Algorithm: nlminb   
##   
## R coded nlminb Solver:   
##   
## 0: 2276.2288: 0.0558695 0.100000 0.0500000 0.0500000 0.400000 0.400000  
## 1: 2247.9835: 0.0558672 0.0773139 0.0552636 0.0495434 0.387486 0.387320  
## 2: 2229.9289: 0.0558634 0.0730970 0.0784165 0.0648966 0.393423 0.393248  
## 3: 2218.0766: 0.0558605 0.0533761 0.0800700 0.0627795 0.384556 0.384243  
## 4: 2205.0845: 0.0558535 0.0502299 0.0981058 0.0738019 0.391533 0.391348  
## 5: 2202.4562: 0.0558501 0.0377075 0.100118 0.0728041 0.387744 0.387514  
-----Omitted some lines  
##   
## Final Estimate of the Negative LLH:  
## LLH: -3307.239 norm LLH: -1.867441   
## mu omega alpha1 alpha2 beta1   
## 1.658196e-03 4.741829e-05 2.046052e-01 2.258521e-02 1.876304e-01   
## beta2   
## 5.919380e-01   
##   
## R-optimhess Difference Approximated Hessian Matrix:  
## mu omega alpha1 alpha2 beta1  
## mu -1.982454e+06 -3392668 3004.546 5984.73 1109.323  
## omega -3.392668e+06 -28858339031 -12737554.873 -13845985.33 -24196137.284  
## alpha1 3.004546e+03 -12737555 -10320.368 -10738.82 -15318.676  
## alpha2 5.984730e+03 -13845985 -10738.822 -13459.23 -17398.100  
## beta1 1.109323e+03 -24196137 -15318.676 -17398.10 -26558.737  
## beta2 -2.774372e+01 -24652189 -15517.933 -17222.51 -26865.778  
## beta2  
## mu -2.774372e+01  
## omega -2.465219e+07  
## alpha1 -1.551793e+04  
## alpha2 -1.722251e+04  
## beta1 -2.686578e+04  
## beta2 -2.749135e+04  
## attr(,"time")  
## Time difference of 0.3500469 secs  
##   
## --- END OF TRACE ---  
##   
##   
## Time to Estimate Parameters:  
## Time difference of 1.316645 secs

summary(mod.22\_2)

##   
## Title:  
## GARCH Modelling   
##   
## Call:  
## garchFit(formula = ~garch(2, 2), data = bitcoin\_ts\_d1)   
##   
## Mean and Variance Equation:  
## data ~ garch(2, 2)  
## <environment: 0x000000002ba8a5f0>  
## [data = bitcoin\_ts\_d1]  
##   
## Conditional Distribution:  
## norm   
##   
## Coefficient(s):  
## mu omega alpha1 alpha2 beta1 beta2   
## 1.6582e-03 4.7418e-05 2.0461e-01 2.2585e-02 1.8763e-01 5.9194e-01   
##   
## Std. Errors:  
## based on Hessian   
##   
## Error Analysis:  
## Estimate Std. Error t value Pr(>|t|)   
## mu 1.658e-03 7.128e-04 2.326 0.02000 \*   
## omega 4.742e-05 1.480e-05 3.205 0.00135 \*\*   
## alpha1 2.046e-01 2.995e-02 6.832 8.38e-12 \*\*\*  
## alpha2 2.259e-02 3.160e-02 0.715 0.47477   
## beta1 1.876e-01 8.186e-02 2.292 0.02189 \*   
## beta2 5.919e-01 6.674e-02 8.869 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Log Likelihood:  
## 3307.239 normalized: 1.867441   
##   
## Description:  
## Wed May 30 18:56:15 2018 by user: Mohammad   
##   
##   
## Standardised Residuals Tests:  
## Statistic p-Value   
## Jarque-Bera Test R Chi^2 5385.598 0   
## Shapiro-Wilk Test R W 0.9063864 0   
## Ljung-Box Test R Q(10) 37.96594 3.847335e-05  
## Ljung-Box Test R Q(15) 44.46622 9.292393e-05  
## Ljung-Box Test R Q(20) 54.66252 4.611045e-05  
## Ljung-Box Test R^2 Q(10) 7.565348 0.6712095   
## Ljung-Box Test R^2 Q(15) 10.16227 0.8094103   
## Ljung-Box Test R^2 Q(20) 12.73401 0.8885026   
## LM Arch Test R TR^2 8.543814 0.7413214   
##   
## Information Criterion Statistics:  
## AIC BIC SIC HQIC   
## -3.728107 -3.709544 -3.728130 -3.721249

mod.11 = garch(bitcoin\_ts\_d1,order=c(1,1),trace = FALSE)

## Warning in sqrt(pred$e): NaNs produced

summary(mod.11)

All the coefficients are significant at 5% level of significance.

##   
## Call:  
## garch(x = bitcoin\_ts\_d1, order = c(1, 1), trace = FALSE)  
##   
## Model:  
## GARCH(1,1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.55811 -0.33740 0.06798 0.53194 4.76716   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## a0 3.632e-05 3.816e-06 9.518 <2e-16 \*\*\*  
## a1 1.485e-01 9.555e-03 15.546 <2e-16 \*\*\*  
## b1 8.522e-01 7.025e-03 121.310 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 5626.9, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 1.7522, df = 1, p-value = 0.1856

mod.11\_1 = garchFit(formula = ~garch(1,1), data =bitcoin\_ts\_d1, trace = FALSE )  
summary(mod.11\_1)

##   
## Title:  
## GARCH Modelling   
##   
## Call:  
## garchFit(formula = ~garch(1, 1), data = bitcoin\_ts\_d1, trace = FALSE)   
##   
## Mean and Variance Equation:  
## data ~ garch(1, 1)  
## <environment: 0x000000002c6e7798>  
## [data = bitcoin\_ts\_d1]  
##   
## Conditional Distribution:  
## norm   
##   
## Coefficient(s):  
## mu omega alpha1 beta1   
## 1.7761e-03 3.4971e-05 1.5137e-01 8.5168e-01   
##   
## Std. Errors:  
## based on Hessian   
##   
## Error Analysis:  
## Estimate Std. Error t value Pr(>|t|)   
## mu 1.776e-03 7.169e-04 2.477 0.013235 \*   
## omega 3.497e-05 9.961e-06 3.511 0.000447 \*\*\*  
## alpha1 1.514e-01 2.029e-02 7.459 8.7e-14 \*\*\*  
## beta1 8.517e-01 1.852e-02 45.998 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Log Likelihood:  
## 3302.659 normalized: 1.864855   
##   
## Description:  
## Wed May 30 18:56:19 2018 by user: Mohammad   
##   
##   
## Standardised Residuals Tests:  
## Statistic p-Value   
## Jarque-Bera Test R Chi^2 5531.756 0   
## Shapiro-Wilk Test R W 0.9051734 0   
## Ljung-Box Test R Q(10) 38.76437 2.791702e-05  
## Ljung-Box Test R Q(15) 45.18041 7.171091e-05  
## Ljung-Box Test R Q(20) 55.28813 3.718178e-05  
## Ljung-Box Test R^2 Q(10) 8.12715 0.6164186   
## Ljung-Box Test R^2 Q(15) 11.47579 0.7181952   
## Ljung-Box Test R^2 Q(20) 13.62341 0.8490653   
## LM Arch Test R TR^2 8.943048 0.7077871   
##   
## Information Criterion Statistics:  
## AIC BIC SIC HQIC   
## -3.725193 -3.712818 -3.725204 -3.720621

mod.33 = garch(bitcoin\_ts\_d1,order=c(3,3),trace = FALSE)  
summary(mod.33)

Higher order parameters are insignificant but a1,a3 and b2 is significant

##   
## Call:  
## garch(x = bitcoin\_ts\_d1, order = c(3, 3), trace = FALSE)  
##   
## Model:  
## GARCH(3,3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.53751 -0.32699 0.06485 0.53062 4.70153   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## a0 8.965e-05 2.380e-05 3.766 0.000166 \*\*\*  
## a1 1.681e-01 1.840e-02 9.132 < 2e-16 \*\*\*  
## a2 8.004e-02 6.194e-02 1.292 0.196287   
## a3 1.035e-01 3.456e-02 2.995 0.002742 \*\*   
## b1 1.008e-09 3.346e-01 0.000 1.000000   
## b2 4.201e-01 1.245e-01 3.374 0.000740 \*\*\*  
## b3 2.281e-01 2.097e-01 1.087 0.276867   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 5702.1, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 0.37393, df = 1, p-value = 0.5409

mod.33\_3 = garchFit(formula = ~garch(3,3), data =bitcoin\_ts\_d1, trace = FALSE, cond.dist = "QMLE" )  
summary(mod.33\_3)

##   
## Title:  
## GARCH Modelling   
##   
## Call:  
## garchFit(formula = ~garch(3, 3), data = bitcoin\_ts\_d1, cond.dist = "QMLE",   
## trace = FALSE)   
##   
## Mean and Variance Equation:  
## data ~ garch(3, 3)  
## <environment: 0x000000002bc567a8>  
## [data = bitcoin\_ts\_d1]  
##   
## Conditional Distribution:  
## QMLE   
##   
## Coefficient(s):  
## mu omega alpha1 alpha2 alpha3 beta1   
## 1.5781e-03 6.5737e-05 1.9802e-01 5.6504e-02 5.1498e-02 1.0000e-08   
## beta2 beta3   
## 5.6444e-01 1.4050e-01   
##   
## Std. Errors:  
## robust   
##   
## Error Analysis:  
## Estimate Std. Error t value Pr(>|t|)   
## mu 1.578e-03 7.026e-04 2.246 0.024694 \*   
## omega 6.574e-05 4.627e-05 1.421 0.155421   
## alpha1 1.980e-01 5.504e-02 3.598 0.000321 \*\*\*  
## alpha2 5.650e-02 6.630e-02 0.852 0.394067   
## alpha3 5.150e-02 2.429e-02 2.120 0.033999 \*   
## beta1 1.000e-08 3.482e-01 0.000 1.000000   
## beta2 5.644e-01 1.241e-01 4.547 5.44e-06 \*\*\*  
## beta3 1.405e-01 2.394e-01 0.587 0.557285   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Log Likelihood:  
## 3307.403 normalized: 1.867534   
##   
## Description:  
## Wed May 30 18:56:22 2018 by user: Mohammad   
##   
##   
## Standardised Residuals Tests:  
## Statistic p-Value   
## Jarque-Bera Test R Chi^2 5193.371 0   
## Shapiro-Wilk Test R W 0.9065786 0   
## Ljung-Box Test R Q(10) 38.38302 3.254477e-05  
## Ljung-Box Test R Q(15) 44.95669 7.778704e-05  
## Ljung-Box Test R Q(20) 55.38758 3.592769e-05  
## Ljung-Box Test R^2 Q(10) 7.727321 0.6554521   
## Ljung-Box Test R^2 Q(15) 10.4622 0.789704   
## Ljung-Box Test R^2 Q(20) 13.03699 0.8757915   
## LM Arch Test R TR^2 8.710492 0.7274391   
##   
## Information Criterion Statistics:  
## AIC BIC SIC HQIC   
## -3.726034 -3.701283 -3.726075 -3.716890

mod.22 = garch(bitcoin\_ts\_d1,order=c(2,2),trace = FALSE)

## Warning in sqrt(pred$e): NaNs produced

summary(mod.22) # a1 and b2 significant

##   
## Call:  
## garch(x = bitcoin\_ts\_d1, order = c(2, 2), trace = FALSE)  
##   
## Model:  
## GARCH(2,2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.74488 -0.33250 0.06947 0.53368 4.60743   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## a0 4.863e-05 6.861e-06 7.088 1.36e-12 \*\*\*  
## a1 2.019e-01 1.825e-02 11.059 < 2e-16 \*\*\*  
## a2 1.973e-02 2.304e-02 0.856 0.392   
## b1 1.876e-01 8.065e-02 2.326 0.020 \*   
## b2 5.942e-01 6.731e-02 8.827 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 5506.8, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 0.026848, df = 1, p-value = 0.8698

mod.22\_2 = garchFit(formula = ~garch(2,2), data =bitcoin\_ts\_d1, trace = FALSE, cond.dist = "QMLE" )  
summary(mod.22\_2)

##   
## Title:  
## GARCH Modelling   
##   
## Call:  
## garchFit(formula = ~garch(2, 2), data = bitcoin\_ts\_d1, cond.dist = "QMLE",   
## trace = FALSE)   
##   
## Mean and Variance Equation:  
## data ~ garch(2, 2)  
## <environment: 0x000000002b8d21b8>  
## [data = bitcoin\_ts\_d1]  
##   
## Conditional Distribution:  
## QMLE   
##   
## Coefficient(s):  
## mu omega alpha1 alpha2 beta1 beta2   
## 1.6582e-03 4.7418e-05 2.0461e-01 2.2585e-02 1.8763e-01 5.9194e-01   
##   
## Std. Errors:  
## robust   
##   
## Error Analysis:  
## Estimate Std. Error t value Pr(>|t|)   
## mu 1.658e-03 7.016e-04 2.364 0.018101 \*   
## omega 4.742e-05 4.027e-05 1.177 0.239025   
## alpha1 2.046e-01 5.437e-02 3.763 0.000168 \*\*\*  
## alpha2 2.259e-02 5.934e-02 0.381 0.703484   
## beta1 1.876e-01 9.841e-02 1.907 0.056563 .   
## beta2 5.919e-01 7.051e-02 8.395 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Log Likelihood:  
## 3307.239 normalized: 1.867441   
##   
## Description:  
## Wed May 30 18:56:24 2018 by user: Mohammad   
##   
##   
## Standardised Residuals Tests:  
## Statistic p-Value   
## Jarque-Bera Test R Chi^2 5385.598 0   
## Shapiro-Wilk Test R W 0.9063864 0   
## Ljung-Box Test R Q(10) 37.96594 3.847335e-05  
## Ljung-Box Test R Q(15) 44.46622 9.292393e-05  
## Ljung-Box Test R Q(20) 54.66252 4.611045e-05  
## Ljung-Box Test R^2 Q(10) 7.565348 0.6712095   
## Ljung-Box Test R^2 Q(15) 10.16227 0.8094103   
## Ljung-Box Test R^2 Q(20) 12.73401 0.8885026   
## LM Arch Test R TR^2 8.543814 0.7413214   
##   
## Information Criterion Statistics:  
## AIC BIC SIC HQIC   
## -3.728107 -3.709544 -3.728130 -3.721249

mod.23 = garch(bitcoin\_ts\_d1,order=c(2,3),trace = FALSE)

## Warning in sqrt(pred$e): NaNs produced

summary(mod.23)

a1,a3 and b2 is significant

##   
## Call:  
## garch(x = bitcoin\_ts\_d1, order = c(2, 3), trace = FALSE)  
##   
## Model:  
## GARCH(2,3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.77573 -0.32992 0.06924 0.53528 4.56469   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## a0 5.090e-05 7.364e-06 6.912 4.78e-12 \*\*\*  
## a1 2.380e-01 2.471e-02 9.630 < 2e-16 \*\*\*  
## a2 7.656e-03 2.870e-02 0.267 0.790   
## a3 8.571e-08 2.544e-02 0.000 1.000   
## b1 2.076e-01 8.996e-02 2.308 0.021 \*   
## b2 5.602e-01 7.773e-02 7.207 5.72e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 6011.6, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 0.007324, df = 1, p-value = 0.9318

mod.23\_2 = garchFit(formula = ~garch(3,2), data =bitcoin\_ts\_d1, trace = FALSE, cond.dist = "QMLE" )  
summary(mod.23\_2)

##   
## Title:  
## GARCH Modelling   
##   
## Call:  
## garchFit(formula = ~garch(3, 2), data = bitcoin\_ts\_d1, cond.dist = "QMLE",   
## trace = FALSE)   
##   
## Mean and Variance Equation:  
## data ~ garch(3, 2)  
## <environment: 0x0000000029cd8330>  
## [data = bitcoin\_ts\_d1]  
##   
## Conditional Distribution:  
## QMLE   
##   
## Coefficient(s):  
## mu omega alpha1 alpha2 alpha3 beta1   
## 0.00162181 0.00004551 0.20771275 0.01217701 0.00000001 0.20654104   
## beta2   
## 0.57978937   
##   
## Std. Errors:  
## robust   
##   
## Error Analysis:  
## Estimate Std. Error t value Pr(>|t|)   
## mu 1.622e-03 6.922e-04 2.343 0.019123 \*   
## omega 4.551e-05 6.858e-05 0.664 0.506915   
## alpha1 2.077e-01 5.553e-02 3.741 0.000184 \*\*\*  
## alpha2 1.218e-02 6.437e-02 0.189 0.849968   
## alpha3 1.000e-08 1.602e-01 0.000 1.000000   
## beta1 2.065e-01 1.141e-01 1.811 0.070183 .   
## beta2 5.798e-01 1.269e-01 4.568 4.92e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Log Likelihood:  
## 3307.156 normalized: 1.867395   
##   
## Description:  
## Wed May 30 18:56:26 2018 by user: Mohammad   
##   
##   
## Standardised Residuals Tests:  
## Statistic p-Value   
## Jarque-Bera Test R Chi^2 5336.817 0   
## Shapiro-Wilk Test R W 0.9065388 0   
## Ljung-Box Test R Q(10) 37.71505 4.253928e-05  
## Ljung-Box Test R Q(15) 44.20356 0.0001021781  
## Ljung-Box Test R Q(20) 54.35769 5.118924e-05  
## Ljung-Box Test R^2 Q(10) 7.561235 0.6716088   
## Ljung-Box Test R^2 Q(15) 10.07274 0.8151397   
## Ljung-Box Test R^2 Q(20) 12.66778 0.8911789   
## LM Arch Test R TR^2 8.559359 0.7400346   
##   
## Information Criterion Statistics:  
## AIC BIC SIC HQIC   
## -3.726885 -3.705227 -3.726916 -3.718883

m.11 = garch(bitcoin\_ts\_d1,order=c(1,1),trace = FALSE)

## Warning in sqrt(pred$e): NaNs produced

summary(m.11)

All the coefficients are significant at 5% level of significance.

##   
## Call:  
## garch(x = bitcoin\_ts\_d1, order = c(1, 1), trace = FALSE)  
##   
## Model:  
## GARCH(1,1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.55811 -0.33740 0.06798 0.53194 4.76716   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)   
## a0 3.632e-05 3.816e-06 9.518 <2e-16 \*\*\*  
## a1 1.485e-01 9.555e-03 15.546 <2e-16 \*\*\*  
## b1 8.522e-01 7.025e-03 121.310 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 5626.9, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 1.7522, df = 1, p-value = 0.1856

m.11\_2 = garchFit(formula = ~garch(1,1), data =bitcoin\_ts\_d1 )

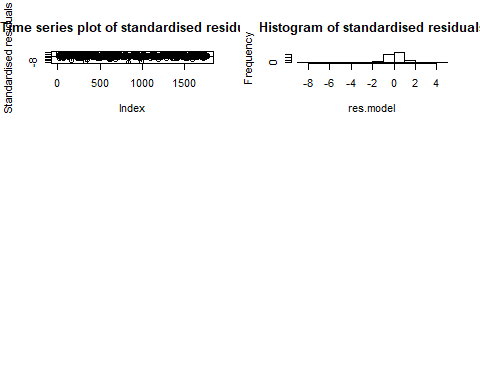
##   
## Series Initialization:  
## ARMA Model: arma  
## Formula Mean: ~ arma(0, 0)  
## GARCH Model: garch  
## Formula Variance: ~ garch(1, 1)  
## ARMA Order: 0 0  
## Max ARMA Order: 0  
## GARCH Order: 1 1  
## Max GARCH Order: 1  
## Maximum Order: 1  
## Conditional Dist: norm  
## h.start: 2  
## llh.start: 1  
## Length of Series: 1771  
## Recursion Init: mci  
## Series Scale: 0.04499265  
##   
## Parameter Initialization:  
## Initial Parameters: $params  
## Limits of Transformations: $U, $V  
## Which Parameters are Fixed? $includes  
## Parameter Matrix:  
## U V params includes  
## mu -0.55869489 0.5586949 0.05586949 TRUE  
## omega 0.00000100 100.0000000 0.10000000 TRUE  
## alpha1 0.00000001 1.0000000 0.10000000 TRUE  
## gamma1 -0.99999999 1.0000000 0.10000000 FALSE  
## beta1 0.00000001 1.0000000 0.80000000 TRUE  
## delta 0.00000000 2.0000000 2.00000000 FALSE  
## skew 0.10000000 10.0000000 1.00000000 FALSE  
## shape 1.00000000 10.0000000 4.00000000 FALSE  
## Index List of Parameters to be Optimized:  
## mu omega alpha1 beta1   
## 1 2 3 5   
## Persistence: 0.9   
##   
##   
## --- START OF TRACE ---  
## Selected Algorithm: nlminb   
##   
## R coded nlminb Solver:   
##   
## 0: 2263.1542: 0.0558695 0.100000 0.100000 0.800000  
## 1: 2236.8980: 0.0558668 0.0738548 0.103287 0.787343  
## 2: 2218.5770: 0.0558610 0.0662325 0.129436 0.797959  
## 3: 2212.9766: 0.0558595 0.0590437 0.131085 0.796531  
## 4: 2205.7258: 0.0558548 0.0467430 0.139439 0.798686  
## 5: 2201.8551: 0.0558502 0.0473228 0.149570 0.807139  
## 6: 2198.8553: 0.0558424 0.0342513 0.150769 0.808597  
## 7: 2194.8871: 0.0558248 0.0346154 0.153253 0.821560  
## 8: 2191.6688: 0.0557783 0.0193849 0.148489 0.842593  
## 9: 2191.4159: 0.0557769 0.0233957 0.150252 0.845168  
## 10: 2190.4138: 0.0557065 0.0196974 0.148829 0.848089  
## 11: 2190.3208: 0.0556414 0.0166560 0.149295 0.851963  
## 12: 2190.3168: 0.0556303 0.0181444 0.149750 0.852825  
## 13: 2190.1939: 0.0556195 0.0173652 0.149664 0.852439  
## 14: 2190.1880: 0.0555760 0.0171357 0.150035 0.852371  
## 15: 2190.1780: 0.0554822 0.0173521 0.150538 0.852072  
## 16: 2190.0838: 0.0531097 0.0185355 0.156622 0.845573  
## 17: 2189.7386: 0.0453188 0.0170279 0.152945 0.851162  
## 18: 2189.7372: 0.0453185 0.0171304 0.152934 0.851176  
## 19: 2189.7364: 0.0453179 0.0171298 0.152855 0.851108  
## 20: 2189.7357: 0.0453086 0.0172396 0.152794 0.851103  
## 21: 2189.7344: 0.0452859 0.0171926 0.152715 0.851071  
## 22: 2189.7142: 0.0442999 0.0174356 0.151011 0.851655  
## 23: 2189.6675: 0.0403352 0.0173049 0.151363 0.851583  
## 24: 2189.6660: 0.0394539 0.0172956 0.151472 0.851647  
## 25: 2189.6659: 0.0394189 0.0172562 0.151302 0.851731  
## 26: 2189.6659: 0.0394790 0.0172760 0.151375 0.851682  
## 27: 2189.6659: 0.0394757 0.0172754 0.151373 0.851685  
##   
## Final Estimate of the Negative LLH:  
## LLH: -3302.659 norm LLH: -1.864855   
## mu omega alpha1 beta1   
## 1.776118e-03 3.497116e-05 1.513733e-01 8.516849e-01   
##   
## R-optimhess Difference Approximated Hessian Matrix:  
## mu omega alpha1 beta1  
## mu -1954855.312 -8799736 4417.885 -1473.551  
## omega -8799735.943 -61052788385 -27359445.398 -52424061.717  
## alpha1 4417.885 -27359445 -22556.487 -33343.872  
## beta1 -1473.551 -52424062 -33343.872 -57359.325  
## attr(,"time")  
## Time difference of 0.1277189 secs  
##   
## --- END OF TRACE ---  
##   
##   
## Time to Estimate Parameters:  
## Time difference of 0.5521641 secs

summary(m.11\_2)

##   
## Title:  
## GARCH Modelling   
##   
## Call:  
## garchFit(formula = ~garch(1, 1), data = bitcoin\_ts\_d1)   
##   
## Mean and Variance Equation:  
## data ~ garch(1, 1)  
## <environment: 0x000000002ad2a500>  
## [data = bitcoin\_ts\_d1]  
##   
## Conditional Distribution:  
## norm   
##   
## Coefficient(s):  
## mu omega alpha1 beta1   
## 1.7761e-03 3.4971e-05 1.5137e-01 8.5168e-01   
##   
## Std. Errors:  
## based on Hessian   
##   
## Error Analysis:  
## Estimate Std. Error t value Pr(>|t|)   
## mu 1.776e-03 7.169e-04 2.477 0.013235 \*   
## omega 3.497e-05 9.961e-06 3.511 0.000447 \*\*\*  
## alpha1 1.514e-01 2.029e-02 7.459 8.7e-14 \*\*\*  
## beta1 8.517e-01 1.852e-02 45.998 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Log Likelihood:  
## 3302.659 normalized: 1.864855   
##   
## Description:  
## Wed May 30 18:56:27 2018 by user: Mohammad   
##   
##   
## Standardised Residuals Tests:  
## Statistic p-Value   
## Jarque-Bera Test R Chi^2 5531.756 0   
## Shapiro-Wilk Test R W 0.9051734 0   
## Ljung-Box Test R Q(10) 38.76437 2.791702e-05  
## Ljung-Box Test R Q(15) 45.18041 7.171091e-05  
## Ljung-Box Test R Q(20) 55.28813 3.718178e-05  
## Ljung-Box Test R^2 Q(10) 8.12715 0.6164186   
## Ljung-Box Test R^2 Q(15) 11.47579 0.7181952   
## Ljung-Box Test R^2 Q(20) 13.62341 0.8490653   
## LM Arch Test R TR^2 8.943048 0.7077871   
##   
## Information Criterion Statistics:  
## AIC BIC SIC HQIC   
## -3.725193 -3.712818 -3.725204 -3.720621

residual.analysis(mod.22,class="GARCH",start=2)

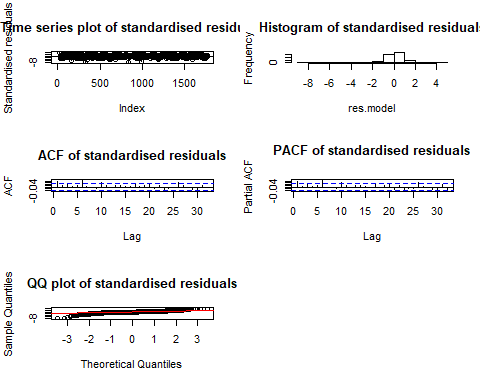
## Error in na.omit.ts(as.ts(x)): time series contains internal NAs



Histogram looks symmetric

residual.analysis(mod.11,class="GARCH",start=2)

## Error in LBQPlot(res.model, na.action = na.omit, lag.max = 200, StartLag = k + : unused argument (na.action = na.omit)

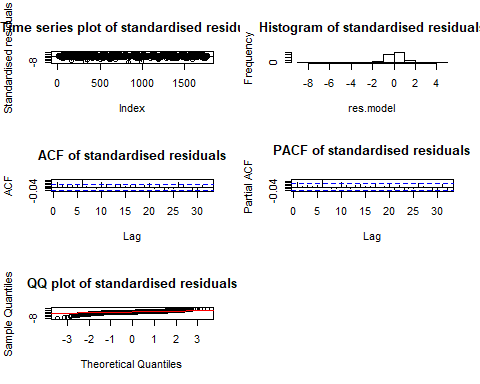


##   
## Shapiro-Wilk normality test  
##   
## data: res.model  
## W = 0.9021, p-value < 2.2e-16

Histogram looks symmetric,ACF and PACF looks insignificant,time series plot is bouncing around Zero.

residual.analysis(mod.33,class="GARCH",start=2)

## Error in LBQPlot(res.model, na.action = na.omit, lag.max = 200, StartLag = k + : unused argument (na.action = na.omit)

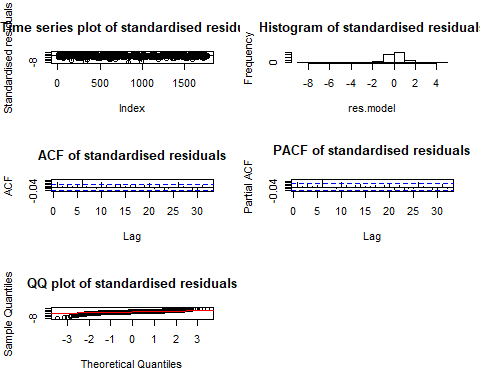


##   
## Shapiro-Wilk normality test  
##   
## data: res.model  
## W = 0.90221, p-value < 2.2e-16

Histogram looks symmetric,ACF and PACF looks insignificant,time series plot is bouncing around Zero.

residual.analysis(mod.23,class="GARCH",start=2)

## Error in LBQPlot(res.model, na.action = na.omit, lag.max = 200, StartLag = k + : unused argument (na.action = na.omit)



##   
## Shapiro-Wilk normality test  
##   
## data: res.model  
## W = 0.90403, p-value < 2.2e-16

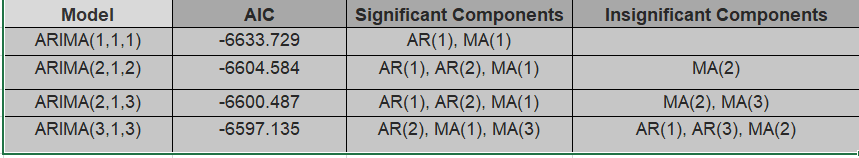
Histogram looks symmetric,ACF and PACF looks insignificant,time series plot is bouncing around Zero.

sort.score(AIC(mod.23,mod.33,mod.11,mod.22), score = "aic")#we find model mod.11 with lowest AIC (-6633.729)

## df AIC  
## mod.11 3 -6633.729  
## mod.22 5 -6604.584  
## mod.23 6 -6600.487  
## mod.33 7 -6597.135

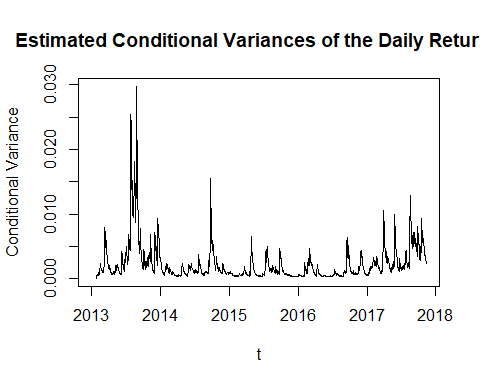
#sort.score(BIC(mod.23,mod.33,mod.11,mod.22), score = "bic")#didnot get any BIC value

## Diagonistic Check



All the models looks good, but we will consider GARCH(1,1) for further analysis based on its lower AIC Value

par(mfrow=c(1,1))  
plot((fitted(mod.11)[,1])^2,type='l',ylab='Conditional Variance',xlab='t',main="Estimated Conditional Variances of the Daily Returns")

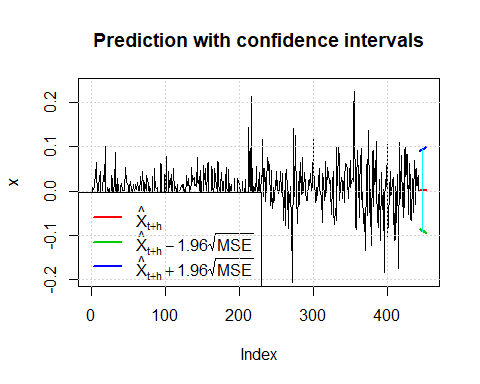


**##Fitted value for conditional variances**

Now we have consider fitted value to plot conditional variances. It looks after 2015 the series settles down.

**##Prediction with confidence intervals**

fGarch::predict(mod.11\_1,n.ahead=10,trace=FALSE,plot=TRUE)



## meanForecast meanError standardDeviation lowerInterval upperInterval  
## 1 0.001776118 0.04525420 0.04525420 -0.08692049 0.09047272  
## 2 0.001776118 0.04570751 0.04570751 -0.08780896 0.09136120  
## 3 0.001776118 0.04615774 0.04615774 -0.08869139 0.09224363  
## 4 0.001776118 0.04660498 0.04660498 -0.08956796 0.09312019  
## 5 0.001776118 0.04704931 0.04704931 -0.09043883 0.09399106  
## 6 0.001776118 0.04749082 0.04749082 -0.09130418 0.09485642  
## 7 0.001776118 0.04792960 0.04792960 -0.09216417 0.09571641  
## 8 0.001776118 0.04836572 0.04836572 -0.09301895 0.09657119  
## 9 0.001776118 0.04879926 0.04879926 -0.09386868 0.09742091  
## 10 0.001776118 0.04923029 0.04923029 -0.09471348 0.09826572

.

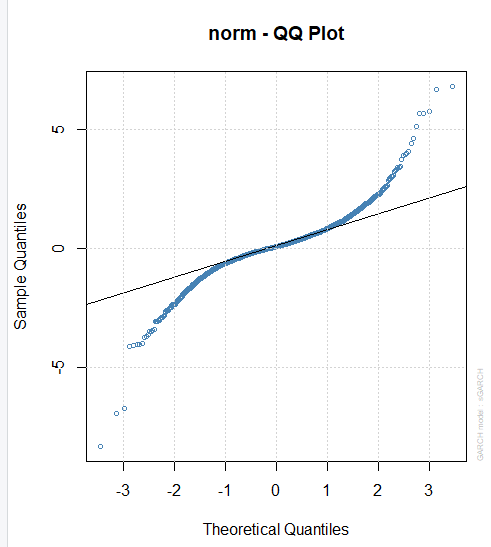
## Overfitting models-

**##overfitting model with arma part**

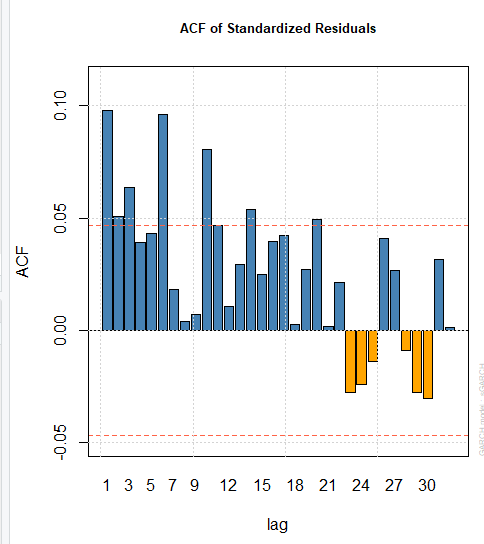
model11<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),   
 mean.model = list(armaOrder = c(2,2), include.mean = FALSE),   
 distribution.model = "norm")  
mo.22\_11<-ugarchfit(spec=model11,data=bitcoin\_ts,sample=100)  
mo.22\_11

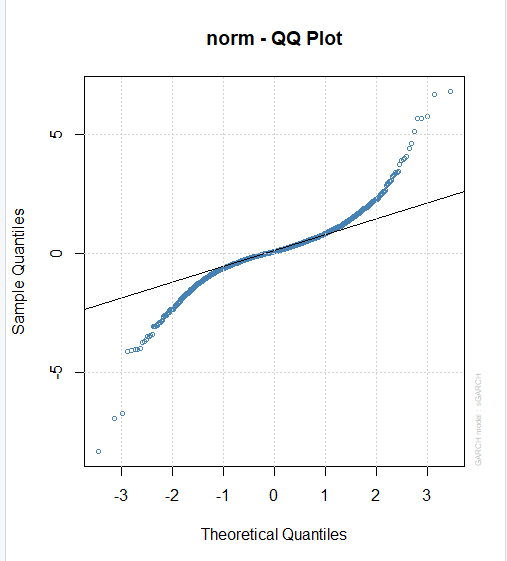
##   
## \*---------------------------------\*  
## \* GARCH Model Fit \*  
## \*---------------------------------\*  
##   
## Conditional Variance Dynamics   
## -----------------------------------  
## GARCH Model : sGARCH(1,1)  
## Mean Model : ARFIMA(2,0,2)  
## Distribution : norm   
##   
## Optimal Parameters  
## ------------------------------------  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 1.403757 0.000532 2637.8874 0.000000  
## ar2 -0.403517 0.000444 -908.4404 0.000000  
## ma1 -0.395935 0.029924 -13.2313 0.000000  
## ma2 -0.038134 0.030161 -1.2643 0.206106  
## omega 5.343524 1.138585 4.6931 0.000003  
## alpha1 0.195093 0.014610 13.3537 0.000000  
## beta1 0.803907 0.015793 50.9036 0.000000  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 1.403757 0.000693 2025.4088 0.000000  
## ar2 -0.403517 0.000409 -987.2835 0.000000  
## ma1 -0.395935 0.039800 -9.9481 0.000000  
## ma2 -0.038134 0.035131 -1.0855 0.277710  
## omega 5.343524 4.732839 1.1290 0.258885  
## alpha1 0.195093 0.052018 3.7505 0.000177  
## beta1 0.803907 0.069663 11.5400 0.000000  
##   
## LogLikelihood : -8187.083   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike 9.2484  
## Bayes 9.2700  
## Shibata 9.2484  
## Hannan-Quinn 9.2564  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 16.99 3.764e-05  
## Lag[2\*(p+q)+(p+q)-1][11] 42.79 0.000e+00  
## Lag[4\*(p+q)+(p+q)-1][19] 56.88 0.000e+00  
## d.o.f=4  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 1.071 0.3008  
## Lag[2\*(p+q)+(p+q)-1][5] 1.915 0.6386  
## Lag[4\*(p+q)+(p+q)-1][9] 2.530 0.8332  
## d.o.f=2  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[3] 0.2315 0.500 2.000 0.6304  
## ARCH Lag[5] 1.3944 1.440 1.667 0.6205  
## ARCH Lag[7] 1.6169 2.315 1.543 0.7976  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 5.8904  
## Individual Statistics:   
## ar1 0.16274  
## ar2 0.16571  
## ma1 0.02791  
## ma2 0.02047  
## omega 0.22261  
## alpha1 1.08883  
## beta1 0.34362  
##   
## Asymptotic Critical Values (10% 5% 1%)  
## Joint Statistic: 1.69 1.9 2.35  
## Individual Statistic: 0.35 0.47 0.75  
##   
## Sign Bias Test  
## ------------------------------------  
## t-value prob sig  
## Sign Bias 0.1446 0.8850   
## Negative Sign Bias 0.4681 0.6398   
## Positive Sign Bias 0.4760 0.6342   
## Joint Effect 0.5657 0.9042   
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 291.2 1.291e-50  
## 2 30 301.9 3.380e-47  
## 3 40 325.2 8.076e-47  
## 4 50 342.8 1.064e-45  
##   
##   
## Elapsed time : 27.56583

We have started fitting GARCH(1,1) and ARMA(1,1) Model and from the parametr estimate we can see that all (ar) are and alha and beta are significant but(ma2) componen is not significant. Now,we will try to iterate through to find the best possible model for model fitting.

model12<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),   
 mean.model = list(armaOrder = c(1,2), include.mean = FALSE),   
 distribution.model = "norm")  
mo.12\_11<-ugarchfit(spec=model12,data=bitcoin\_ts\_d1)  
mo.12\_11

|  |
| --- |
|  |
|  |





The ACF for Standard residuals ans squared residuals looks significant and Q-Q plot has slightly improved to align around the red line

##   
## \*---------------------------------\*  
## \* GARCH Model Fit \*  
## \*---------------------------------\*  
##   
## Conditional Variance Dynamics   
## -----------------------------------  
## GARCH Model : sGARCH(1,1)  
## Mean Model : ARFIMA(1,0,2)  
## Distribution : norm   
##   
## Optimal Parameters  
## ------------------------------------  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 0.951166 0.015516 61.3018 0.000000  
## ma1 -0.907643 0.009036 -100.4444 0.000000  
## ma2 -0.028386 0.011871 -2.3912 0.016795  
## omega 0.000036 0.000010 3.4680 0.000524  
## alpha1 0.143673 0.018735 7.6686 0.000000  
## beta1 0.855327 0.018156 47.1100 0.000000  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 0.951166 0.010925 87.0605 0.000000  
## ma1 -0.907643 0.007854 -115.5641 0.000000  
## ma2 -0.028386 0.005483 -5.1770 0.000000  
## omega 0.000036 0.000032 1.1142 0.265212  
## alpha1 0.143673 0.045653 3.1471 0.001649  
## beta1 0.855327 0.052818 16.1940 0.000000  
##   
## LogLikelihood : 3300.611   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike -3.7206  
## Bayes -3.7021  
## Shibata -3.7206  
## Hannan-Quinn -3.7138  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 1.874 1.710e-01  
## Lag[2\*(p+q)+(p+q)-1][8] 7.425 2.482e-05  
## Lag[4\*(p+q)+(p+q)-1][14] 14.236 3.852e-03  
## d.o.f=3  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 1.070 0.3009  
## Lag[2\*(p+q)+(p+q)-1][5] 1.915 0.6387  
## Lag[4\*(p+q)+(p+q)-1][9] 2.772 0.7959  
## d.o.f=2  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[3] 0.1359 0.500 2.000 0.7123  
## ARCH Lag[5] 1.5283 1.440 1.667 0.5849  
## ARCH Lag[7] 1.8018 2.315 1.543 0.7592  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 0.6825  
## Individual Statistics:   
## ar1 0.03178  
## ma1 0.02547  
## ma2 0.03001  
## omega 0.25118  
## alpha1 0.07172  
## beta1 0.14764  
##   
## Asymptotic Critical Values (10% 5% 1%)  
## Joint Statistic: 1.49 1.68 2.12  
## Individual Statistic: 0.35 0.47 0.75  
##   
## Sign Bias Test  
## ------------------------------------  
## t-value prob sig  
## Sign Bias 1.0782 0.2811   
## Negative Sign Bias 0.1318 0.8952   
## Positive Sign Bias 0.0200 0.9840   
## Joint Effect 2.0908 0.5538   
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 310.2 1.646e-54  
## 2 30 328.6 1.664e-52  
## 3 40 349.5 1.545e-51  
## 4 50 363.8 1.191e-49  
##   
##   
## Elapsed time : 1.047131

We have reduced (ma) component in this model.Every component including (ar),(ma) components are significant

model13<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),   
 mean.model = list(armaOrder = c(1,1), include.mean = FALSE),   
 distribution.model = "norm")  
mo.11\_11<-ugarchfit(spec=model13,data=bitcoin\_ts\_d1)  
mo.11\_11

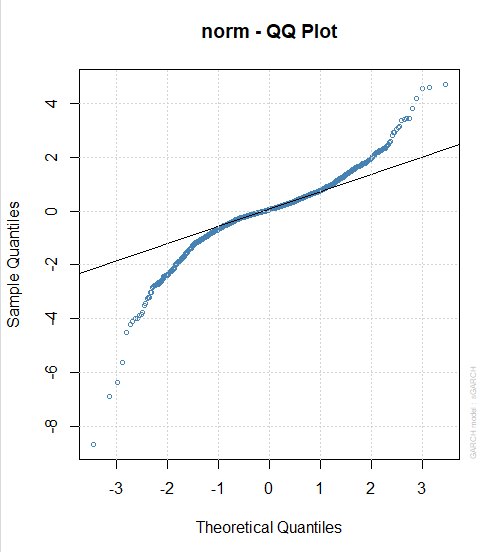
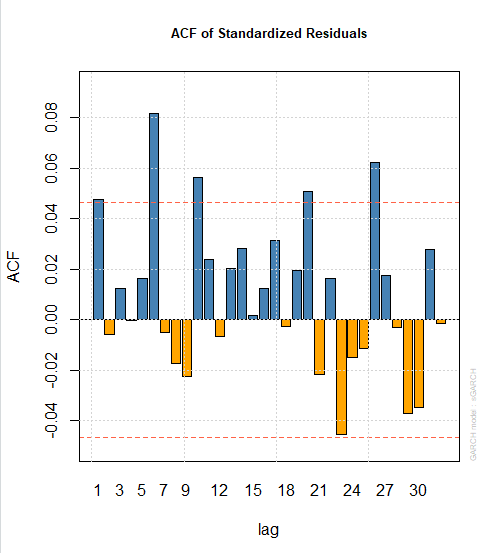
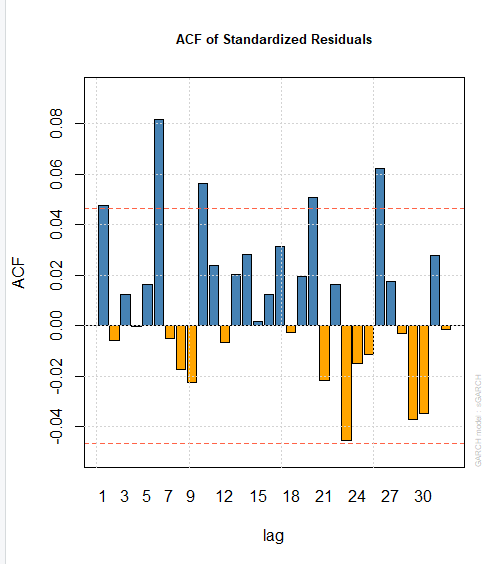
##   
## \*---------------------------------\*  
## \* GARCH Model Fit \*  
## \*---------------------------------\*  
##   
## Conditional Variance Dynamics   
## -----------------------------------  
## GARCH Model : sGARCH(1,1)  
## Mean Model : ARFIMA(1,0,1)  
## Distribution : norm   
##   
## Optimal Parameters  
## ------------------------------------  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 0.931548 0.119612 7.7881 0.000000  
## ma1 -0.911578 0.137030 -6.6524 0.000000  
## omega 0.000037 0.000010 3.5083 0.000451  
## alpha1 0.145607 0.019062 7.6386 0.000000  
## beta1 0.853392 0.018401 46.3778 0.000000  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 0.931548 0.149320 6.2386 0.000000  
## ma1 -0.911578 0.173184 -5.2636 0.000000  
## omega 0.000037 0.000032 1.1334 0.257060  
## alpha1 0.145607 0.045849 3.1758 0.001494  
## beta1 0.853392 0.052719 16.1874 0.000000  
##   
## LogLikelihood : 3301.325   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike -3.7226  
## Bayes -3.7071  
## Shibata -3.7226  
## Hannan-Quinn -3.7168  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 4.290 0.038326  
## Lag[2\*(p+q)+(p+q)-1][5] 4.616 0.011616  
## Lag[4\*(p+q)+(p+q)-1][9] 10.289 0.006926  
## d.o.f=2  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.9156 0.3386  
## Lag[2\*(p+q)+(p+q)-1][5] 1.7891 0.6690  
## Lag[4\*(p+q)+(p+q)-1][9] 2.6482 0.8153  
## d.o.f=2  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[3] 0.1716 0.500 2.000 0.6787  
## ARCH Lag[5] 1.5474 1.440 1.667 0.5799  
## ARCH Lag[7] 1.8288 2.315 1.543 0.7535  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 0.578  
## Individual Statistics:   
## ar1 0.02473  
## ma1 0.02023  
## omega 0.25015  
## alpha1 0.07676  
## beta1 0.14923  
##   
## Asymptotic Critical Values (10% 5% 1%)  
## Joint Statistic: 1.28 1.47 1.88  
## Individual Statistic: 0.35 0.47 0.75  
##   
## Sign Bias Test  
## ------------------------------------  
## t-value prob sig  
## Sign Bias 1.43305 0.1520   
## Negative Sign Bias 0.06355 0.9493   
## Positive Sign Bias 0.15912 0.8736   
## Joint Effect 2.93309 0.4021   
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 309.9 1.891e-54  
## 2 30 343.1 2.142e-55  
## 3 40 377.0 6.887e-57  
## 4 50 382.7 3.038e-53  
##   
##   
## Elapsed time : 0.7904601

All components here are significant

## overfitting model with GARCH part

## all components are significant  
model14<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 2)),   
 mean.model = list(armaOrder = c(1,1), include.mean = FALSE),   
 distribution.model = "norm")  
mo.11\_12<-ugarchfit(spec=model14,data=bitcoin\_ts\_d1)  
mo.11\_12

##   
## \*---------------------------------\*  
## \* GARCH Model Fit \*  
## \*---------------------------------\*  
##   
## Conditional Variance Dynamics   
## -----------------------------------  
## GARCH Model : sGARCH(1,2)  
## Mean Model : ARFIMA(1,0,1)  
## Distribution : norm   
##   
## Optimal Parameters  
## ------------------------------------  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 0.932466 0.083766 11.1319 0.000000  
## ma1 -0.912607 0.096211 -9.4855 0.000000  
## omega 0.000046 0.000013 3.5613 0.000369  
## alpha1 0.203575 0.025702 7.9206 0.000000  
## beta1 0.224043 0.065653 3.4125 0.000644  
## beta2 0.571382 0.061434 9.3008 0.000000  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 0.932466 0.076773 12.1458 0.000000  
## ma1 -0.912607 0.089065 -10.2466 0.000000  
## omega 0.000046 0.000038 1.2163 0.223880  
## alpha1 0.203575 0.053684 3.7921 0.000149  
## beta1 0.224043 0.084911 2.6386 0.008326  
## beta2 0.571382 0.101910 5.6067 0.000000  
##   
## LogLikelihood : 3305.93   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike -3.7266  
## Bayes -3.7081  
## Shibata -3.7267  
## Hannan-Quinn -3.7198  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 4.021 0.044938  
## Lag[2\*(p+q)+(p+q)-1][5] 4.332 0.027130  
## Lag[4\*(p+q)+(p+q)-1][9] 10.092 0.008421  
## d.o.f=2  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.08053 0.7766  
## Lag[2\*(p+q)+(p+q)-1][8] 1.14643 0.9629  
## Lag[4\*(p+q)+(p+q)-1][14] 3.72429 0.9050  
## d.o.f=3  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[4] 1.106 0.500 2.000 0.2930  
## ARCH Lag[6] 1.187 1.461 1.711 0.6947  
## ARCH Lag[8] 1.428 2.368 1.583 0.8529  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 0.6788  
## Individual Statistics:   
## ar1 0.02261  
## ma1 0.02618  
## omega 0.21450  
## alpha1 0.07222  
## beta1 0.13368  
## beta2 0.14683  
##   
## Asymptotic Critical Values (10% 5% 1%)  
## Joint Statistic: 1.49 1.68 2.12  
## Individual Statistic: 0.35 0.47 0.75  
##   
## Sign Bias Test  
## ------------------------------------  
## t-value prob sig  
## Sign Bias 1.4355 0.1513   
## Negative Sign Bias 0.5332 0.5940   
## Positive Sign Bias 0.2236 0.8231   
## Joint Effect 3.1235 0.3730   
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 320.0 1.592e-56  
## 2 30 350.1 8.564e-57  
## 3 40 373.1 3.970e-56  
## 4 50 377.9 2.493e-52  
##   
##   
## Elapsed time : 0.7393129



ACF plots for residuals are mostly significant,Q-Q plot is not that aligned with red line.However since all other parameters are significant,we will consider this model for forecasting

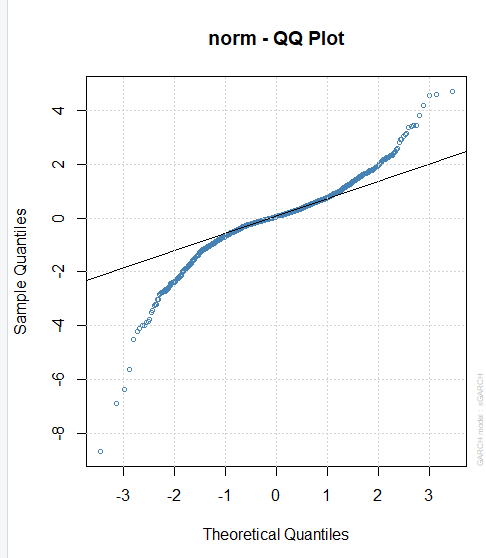
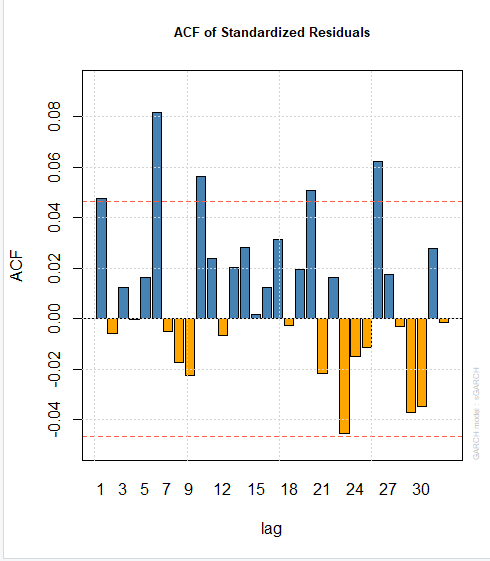
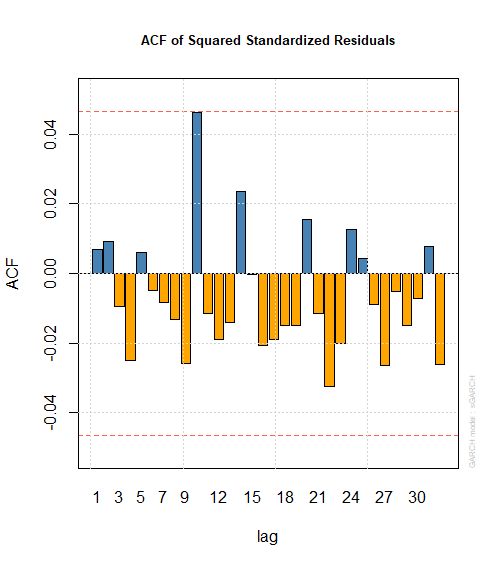
All components here are significant

model15<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 3)),   
 mean.model = list(armaOrder = c(1,1), include.mean = FALSE),   
 distribution.model = "norm")  
mo.11\_13<-ugarchfit(spec=model15,data=bitcoin\_ts\_d1)  
mo.11\_13

##   
## \*---------------------------------\*  
## \* GARCH Model Fit \*  
## \*---------------------------------\*  
##   
## Conditional Variance Dynamics   
## -----------------------------------  
## GARCH Model : sGARCH(1,3)  
## Mean Model : ARFIMA(1,0,1)  
## Distribution : norm   
##   
## Optimal Parameters  
## ------------------------------------  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 -0.144135 0.949149 -0.151857 0.879300  
## ma1 0.183174 0.941540 0.194547 0.845748  
## omega 0.000045 0.000013 3.449108 0.000562  
## alpha1 0.201856 0.027030 7.467797 0.000000  
## beta1 0.225525 0.100604 2.241703 0.024981  
## beta2 0.571619 0.063460 9.007552 0.000000  
## beta3 0.000000 0.078031 0.000004 0.999997  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## ar1 -0.144135 1.610164 -0.089516 0.928672  
## ma1 0.183174 1.600973 0.114414 0.908909  
## omega 0.000045 0.000039 1.151985 0.249327  
## alpha1 0.201856 0.048280 4.180921 0.000029  
## beta1 0.225525 0.192818 1.169629 0.242150  
## beta2 0.571619 0.094800 6.029740 0.000000  
## beta3 0.000000 0.143865 0.000002 0.999998  
##   
## LogLikelihood : 3305.253   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike -3.7247  
## Bayes -3.7031  
## Shibata -3.7248  
## Hannan-Quinn -3.7167  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 2.566 0.1091548  
## Lag[2\*(p+q)+(p+q)-1][5] 4.606 0.0119645  
## Lag[4\*(p+q)+(p+q)-1][9] 12.769 0.0004993  
## d.o.f=2  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.07996 0.7774  
## Lag[2\*(p+q)+(p+q)-1][11] 2.44728 0.9263  
## Lag[4\*(p+q)+(p+q)-1][19] 5.64747 0.9000  
## d.o.f=4  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[5] 0.05052 0.500 2.000 0.8222  
## ARCH Lag[7] 0.15723 1.473 1.746 0.9790  
## ARCH Lag[9] 1.03070 2.402 1.619 0.9271  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 1.5628  
## Individual Statistics:   
## ar1 0.03188  
## ma1 0.03576  
## omega 0.22411  
## alpha1 0.06849  
## beta1 0.13595  
## beta2 0.14745  
## beta3 0.12976  
##   
## Asymptotic Critical Values (10% 5% 1%)  
## Joint Statistic: 1.69 1.9 2.35  
## Individual Statistic: 0.35 0.47 0.75  
##   
## Sign Bias Test  
## ------------------------------------  
## t-value prob sig  
## Sign Bias 0.7432 0.4575   
## Negative Sign Bias 0.2221 0.8242   
## Positive Sign Bias 0.4835 0.6288   
## Joint Effect 1.5023 0.6817   
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 303.7 3.446e-53  
## 2 30 327.3 3.100e-52  
## 3 40 338.3 2.346e-49  
## 4 50 354.2 7.746e-48  
##   
##   
## Elapsed time : 0.8796411

since We get bunch of insignificant alpha,beta so we stop this iterative process for Garch part

So we will consider GARCH(1,2) part for further analysis



The ACF plots for residuals are mostly significant and Q-Q plot looks better as well.

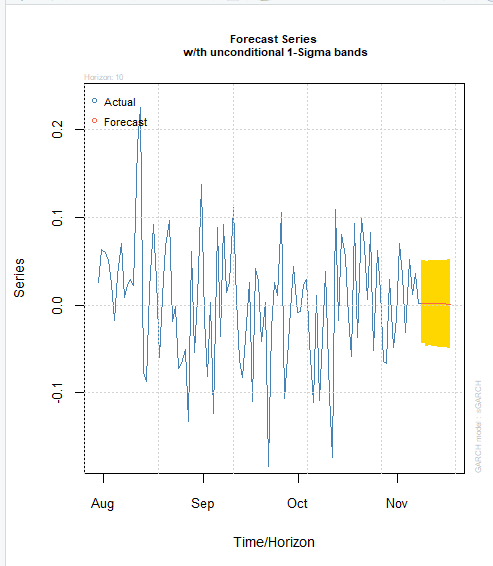
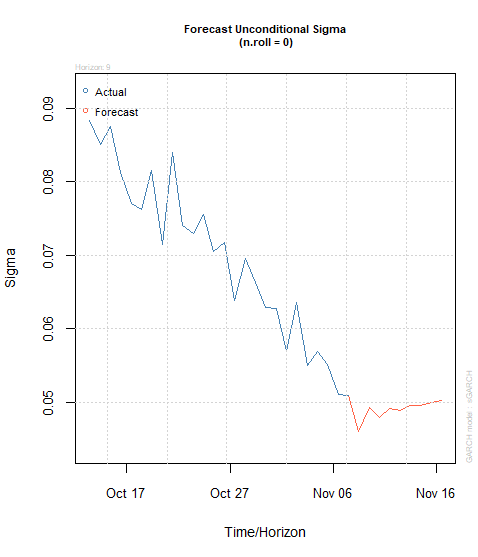
So we will consider ARMA(1,1)+GARCH(1,2) for forecasting

forc= ugarchforecast(mo.11\_12, data = bitcoin\_ts, n.ahead = 10)  
#plot(forc)  
forc@forecast

## $n.ahead  
## [1] 10  
##   
## $N  
## [1] 1771  
##   
## $n.start  
## [1] 0  
##   
## $n.roll  
## [1] 0  
##   
## $sigmaFor  
## 1974-11-07 11:00:00  
## T+1 0.04602436  
## T+2 0.04938297  
## T+3 0.04795373  
## T+4 0.04922564  
## T+5 0.04895436  
## T+6 0.04955567  
## T+7 0.04965758  
## T+8 0.05004044  
## T+9 0.05026077  
## T+10 0.05057079  
##   
## $seriesFor  
## 1974-11-07 11:00:00  
## T+1 0.002110943  
## T+2 0.001968382  
## T+3 0.001835449  
## T+4 0.001711494  
## T+5 0.001595909  
## T+6 0.001488131  
## T+7 0.001387631  
## T+8 0.001293919  
## T+9 0.001206535  
## T+10 0.001125053

## Forecast

forc = ugarchforecast(mo.11\_12, data = bitcoin\_ts\_nxt, n.ahead = 10)  
  
forc <- ts(fitted(forc)[,1],start = c(2018,03,04),frequency = 365.25)

## 

## Model Accuracy check (MASE)

we need to back transformation for getting the MASE value

forc = ugarchforecast(mo.11\_12, data = bitcoin\_ts\_nxt, n.ahead = 9)

forc\_diffinv <- diffinv(forc,differences = 1,xi = bitcoin\_ts[length(bitcoin\_ts)])  
forc\_diffinv

## Time Series:  
## Start = c(2018, 3)   
## End = c(2018, 12)   
## Frequency = 365   
## [1] 11512.60 11573.30 10779.90 9965.57 9395.01 9337.55 8866.00  
## [8] 9578.63 9205.12 9194.85

Y.t = as.vector(bitcoin\_ts\_nxt)  
n = length(as.vector(forc\_diffinv))  
e.t = Y.t - as.vector(forc\_diffinv)  
sum = 0   
for (i in 2:n){  
 sum = sum + abs(Y.t[i] - Y.t[i-1] )  
}  
q.t = e.t / (sum/(n-1))  
q.t

## [1] 0.0000000 0.1413621 -1.7064278 -3.6029622 -4.9317693 -5.0655942  
## [7] -6.1638119 -4.5041388 -5.3740262 -5.3979473

mean(abs(q.t))

## [1] 3.688804

*The mean Mase value we have got MASE—3.6888*

## Conclusion

For the Bitcoin dataset we performed Time Series Analysis.It was very challenging task due to the volatile clustering in that series. However,after fitting the ARIMA models, evidences for conditional heteroscedasticity were observed in the residues. After further investigation we could fit GARCH component to the residuals of ARIMA model sucessfully. The best model fit through this process is ARIMA(1,1) + GARCH(1,2).

The Bitcoin price has forecasted for the next 10 days.

## Appendix

library(TSA)

library(fUnitRoots)

library(forecast)

library(CombMSC)

library(lmtest)

library(fGarch)

library(truncnorm)

library(forecast)

library(itsa)

library(truncnorm)

bitcoin<- read.csv("Bitcoin\_Historical\_Price.csv")$Close

bit\_forecast<-read.csv("Bitcoin\_Prices\_Forecasts.csv")$Closing.price

bitcoin\_ts= ts(bitcoin,start=c(2013,4,27),frequency=365)

bitcoin\_ts\_nxt= ts(bit\_forecast,start=c(2018,3,4),frequency=365)

View(bitcoin\_ts\_nxt)

bitcoin<- read.csv("Bitcoin\_Historical\_Price.csv")$Close

bitcoin\_ts= ts(bitcoin,start=c(2013,4,27),frequency=365.25)

# bitcoin <- read.csv("Bitcoin\_Historical\_Price.csv.csv")

class(bitcoin\_ts)

par(mfrow=c(1,1))

plot.ts(bitcoin\_ts,yax.flip=T,type='o')

par(mfrow=c(1,2))

acf(bitcoin\_ts,lag.max = 200,main="The sample ACF of landings series")

pacf(bitcoin\_ts,lag.max = 200,main="The sample PACF of landings series")

# Calculation of correlation

y=bitcoin\_ts

x=zlag(bitcoin\_ts)

index = 2:length(x)

cor(y[index],x[index])

##Correaltion coeficient : 0.9970915.

##We will do trasnformation using log,since the variance is high at the higher lag that's why we have chosen log transformation

##adf test--

adf.test(bitcoin\_ts)

#With a p-value of 0.7063, we cannot reject the null hypothesis,so we can say the series is non-stationary.

##Transformation

#bitcoin\_ts\_log = log(bitcoin\_ts)

#bitcoin\_ts\_log

bitcoin\_ts\_d1

bitcoin\_ts\_d1 = diff(log(bitcoin\_ts))

adf.test(bitcoin\_ts\_d1, alternative = "stationary")

Box.test(bitcoin\_ts\_d1,type="Ljung",lag=20,fitdf=1)

#since p value is significant (1.343e-05),The data are not independently distributed; they exhibit serial correlation.

plot(bitcoin\_ts\_d1,type='o',ylab= 'Bitcoin Price',

main="Time Series Plot of Ist difference Bitcoin Price")

#It seems there is volatile clustering on the series

McLeod.Li.test(y=bitcoin\_ts\_d1,main="McLeod-Li Test Statistics for Daily bitcoin\_ts\_d1 Returns")

# McLeod-Li test is significnat at 5% level of significance for all lags. This gives a strong idea about existence of volatiliy clustering.

qqnorm(bitcoin\_ts\_d1,main="Q-Q Normal Plot of Daily bitcoin\_ts\_d1 Returns")

qqline(bitcoin\_ts\_d1) # Fat tails is in accordance with volatiliy clustering

par(mfrow=c(1,2))

acf(bitcoin\_ts\_d1, main="The sample ACF plot for return series")

pacf(bitcoin\_ts\_d1, main="The sample PACF plot for return series")

eacf(bitcoin\_ts\_d1)

# EACF all shows pattern of white noise for the correlation structure. However, there is an ARCH effect present in the series.

# From the EACF, we can identify ARMA(1,1), ARMA(2,2), and ARMA(3,3) models for absolute

#So we'll use absolute value and square transformations to figure out this ARCH effect.

abs.bitcoin\_ts\_d1 = abs(bitcoin\_ts\_d1)

sq.bitcoin\_ts\_d1 = bitcoin\_ts\_d1^2

par(mfrow=c(1,2))

acf(abs.bitcoin\_ts\_d1, ci.type="ma",main="The sample ACF plot for absolute return series")

pacf(abs.bitcoin\_ts\_d1, main="The sample PACF plot for absolute return series")

eacf(abs.bitcoin\_ts\_d1)

# After the absolute value transformation, we boserve many signficicant lags in

#both ACF and PACF. Also, EACF do not suggest an ARMA(0,0) model.

# From the EACF, we can identify ARMA(1,1), ARMA(2,2), and ARMA(3,3) models for absolute

#value series.

# These models correspond to parameter settings of [max(2,2),2], [max(1,1),1] and [max(3,3),3].

# So the corresponding tentative GARCH models are GARCH(2,2), GARCH(1,1), GARCH(3,3).

par(mfrow=c(1,2))

acf(sq.bitcoin\_ts\_d1, ci.type="ma",main="The sample ACF plot for squared return series")

pacf(sq.bitcoin\_ts\_d1, main="The sample PACF plot for squared return series")

eacf(sq.bitcoin\_ts\_d1)

#str(mo.11)

# After the square transformation, we boserve many signficicant lags in both ACF and PACF. Also, EACF do not suggest an ARMA(0,0) model.

# From the EACF, we can identify ARMA(2,2),ARMA(3,3), and ARMA(2,3) models for squared series.

# These models correspond to parameter settings of [max(2,2),2], [max(3,3),3], [max(2,3),2]. So the corresponding

# tentative GARCH models are GARCH(2,2), GARCH(3,3), GARCH(3,2).

mo.22 = garch(bitcoin\_ts\_d1,order=c(2,2),trace = FALSE)

summary(mo.22) # All the coefficients but a2 are significant at 5% level of significance.

mo.22\_2 = garchFit(formula = ~garch(2,2), data =bitcoin\_ts\_d1 )

summary(mo.22\_2)

mo.11 = garch(bitcoin\_ts\_d1,order=c(1,1),trace = FALSE)

summary(mo.11)# All the coefficients are significant at 5% level of significance.

mo.11\_1 = garchFit(formula = ~garch(1,1), data =bitcoin\_ts\_d1, trace = FALSE )

summary(mo.11\_1)

#[max(2,2),2], [max(2,3),2] and [max(3,3),3]

mo.33 = garch(bitcoin\_ts\_d1,order=c(3,3),trace = FALSE)

summary(mo.33) # Higher order parameters are insignificant but a1,a3 and b2 is significant

mo.33\_3 = garchFit(formula = ~garch(3,3), data =bitcoin\_ts\_d1, trace = FALSE, cond.dist = "QMLE" )

summary(mo.33\_3)

mo.22 = garch(bitcoin\_ts\_d1,order=c(2,2),trace = FALSE)

summary(mo.22) # a1 and b2 significant

mo.22\_2 = garchFit(formula = ~garch(2,2), data =bitcoin\_ts\_d1, trace = FALSE, cond.dist = "QMLE" )

mo.23 = garch(bitcoin\_ts\_d1,order=c(2,3),trace = FALSE)

summary(mo.23) # a1,a3 and b2 is significant

mo.23\_2 = garchFit(formula = ~garch(3,2), data =bitcoin\_ts\_d1, trace = FALSE, cond.dist = "QMLE" )

summary(mo.33\_3)

summary(mo.22)

fitted(mod.11)

residual.analysis(mo.22,class="GARCH",start=2)

#jarque.bera.test(mo.22)

#shapiro.test(m.11)

residual.analysis(mo.11,class="GARCH",start=2)

residual.analysis(mo.33,class="GARCH",start=3)

residual.analysis(mo.23,class="GARCH",start=3)

?residual.analysis

mo.23

sort.score <- function(x, score = c("aic","bic")){

if (score == "aic"){

x[with(x, order(AIC)),]

} else if (score == "bic") {

x[with(x, order(BIC)),]

} else {

warning('score = "x" only accepts valid arguments ("aic","bic")')

}

}

sort.score(AIC(mo.23,mo.33,mo.11,mo.22), score = "aic")#we find model mo.11 with lowest AIC (-6633.729)

sort.score(BIC(mo.23,mo.33,mo.11,mo.22), score = "bic")#didnot get any BIC value

#For all models, we get suitable diagnostic check results. So, we will go on with GARCH(1,1) model.

par(mfrow=c(1,1))

plot((fitted(mo.11)[,1])^2,type='l',ylab='Conditional Variance',xlab='t',main="Estimated Conditional Variances of the Daily Returns")

# Changes in conditional variance at the beginning of the series and after 2016 it becomes better

fGarch::predict(mo.11\_1,n.ahead=10,trace=FALSE,plot=TRUE)

# Forecasts for the confidance limits are based on the forecasts of conditional variance.

par(mfrow=c(1,1))

library(rugarch)

model11<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),

mean.model = list(armaOrder = c(2,2), include.mean = FALSE),

distribution.model = "norm")

mo.22\_11<-ugarchfit(spec=model11,data=bitcoin\_ts,sample=100)

mo.22\_11

plot(mo.22\_11)

library(rugarch)

model12<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),

mean.model = list(armaOrder = c(1,2), include.mean = FALSE),

distribution.model = "norm")

mo.12\_11<-ugarchfit(spec=model12,data=bitcoin\_ts\_d1)

mo.12\_11

plot(mo.12\_11)

#Every component including (ar),(ma) components are significant

library(rugarch)

model13<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),

mean.model = list(armaOrder = c(1,1), include.mean = FALSE),

distribution.model = "norm")

mo.11\_11<-ugarchfit(spec=model13,data=bitcoin\_ts\_d1)

mo.11\_11

plot(mo.11\_11)

## all components are significant

model14<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 2)),

mean.model = list(armaOrder = c(1,1), include.mean = FALSE),

distribution.model = "norm")

mo.11\_12<-ugarchfit(spec=model14,data=bitcoin\_ts\_d1)

mo.11\_12

plot(mo.11\_12)

## all components are significant

model15<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 3)),

mean.model = list(armaOrder = c(1,1), include.mean = FALSE),

distribution.model = "norm")

mo.11\_13<-ugarchfit(spec=model15,data=bitcoin\_ts\_d1)

mo.11\_13

plot(mo.11\_13)

## Here we can see bunch of components are not significant

#since We get bunch of insignificant alpha,beta so we stop this iterative process for Garch part

##overfitting model with arma part

model16<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 2)),

mean.model = list(armaOrder = c(1,2), include.mean = FALSE),

distribution.model = "norm")

mo.12\_12<-ugarchfit(spec=model16,data=bitcoin\_ts\_d1)

mo.12\_12

plot(mo.11\_12)

model17<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 2)),

mean.model = list(armaOrder = c(2,1), include.mean = FALSE),

distribution.model = "norm")

mo.21\_12<-ugarchfit(spec=model17,data=bitcoin\_ts\_d1)

mo.21\_12

model19<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 2)),

mean.model = list(armaOrder = c(1,1), include.mean = FALSE),

distribution.model = "norm")

mo.11\_12<-ugarchfit(spec=model19,data=bitcoin\_ts\_d1)

mo.11\_12##Better model to fit for forecast

plot(mo.11\_12)

forc= ugarchforecast(mo.11\_12, data = bitcoin\_ts, n.ahead = 10)

plot(forc)

forc@forecast

## we need to back transformation for geting the mase value

data.diff1.back=diffinv(bitcoin\_ts\_d1,xi=bitcoin\_ts[1])

# Forecasts for the confidance limits are based on the forecasts of conditional variance.

forc = ugarchforecast(mo.11\_12, data = bitcoin\_ts\_nxt, n.ahead = 9)

forc <- ts(fitted(forc)[,1],start = c(2018,03,04),frequency = 365.25)

forc

plot(forc)

xi = log(bitcoin\_ts)[1]

## To Find MASE we need to back transformation

#log.data.diff1.back = diffinv(bitcoin\_ts\_d1, xi = log(bitcoin\_ts)[1])#difference back

#log.data.diff1.back = exp(log.data.diff1.back)#log back

forc\_diffinv <- diffinv(forc,differences = 1,xi = bitcoin\_ts[length(bitcoin\_ts)])

forc\_diffinv

##exp(forc\_diffinv)

bitcoin\_ts\_nxt

Y.t = as.vector(bitcoin\_ts\_nxt)

n = length(as.vector(forc\_diffinv))

e.t = Y.t - as.vector(forc\_diffinv)

sum = 0

for (i in 2:n){

sum = sum + abs(Y.t[i] - Y.t[i-1] )

}

q.t = e.t / (sum/(n-1))

q.t

mean(abs(q.t))

---------------------------------------------------------------------------The End------------------------------------------------------------------------------------------