Group 174: Time-Series Model Creation for Bitcoin Price

First Name	Last Name	Monday or Tuesday class	Share project with ITMD 525? (Y or N)
Nastaran	Ghane	Tuesday (I attended on Monday Class)	N
Milad	Sabouri	Monday	N

Table of Contents

1. Introduction

Blockchain is a new phenomenon which is going to change many aspects of modern societies. Most popular application of blockchain is digital crypto currencies. Digital cryptocurrencies are one of the hottest topics in data science, economics, finance, and technology. Currently, Bitcoin is the most famous digital cryptocurrency. So, exploring and discovering the relations and finding the patterns between data which are produced by selling and buying the Bitcoin can be valuable. In this project, we are going to find a prediction model for the Bitcoin price. We have the previous data of Bitcoin market and try to use Time-Series technique to predict the future price in the market.

2. Dataset

Data Set which is used for this project is downloaded from Kaggle website.

However, it originally is collected from https://coinmarketcap.com/ and is accessible for Kaggle website's users. As mentioned earlier, the goal for this project is to create a Time-Series model to forecast Bitcoin's price in future.

There are eight variables in the dataset which are:

```
> str(bitcoin)
Classes 'tbl_df', 'tbl' and 'data.frame':       1793 obs. of  13 variables:
 $ slug       : chr  "bitcoin" "bitcoin" "bitcoin" "bitcoin" ...
 $ symbol     : chr  "BTC" "BTC" "BTC" "BTC" ...
 $ name       : chr  "Bitcoin" "Bitcoin" "Bitcoin" "Bitcoin" ...
 $ date       : chr  "4/28/2013" "4/29/2013" "4/30/2013" "5/1/2013" ...
 $ ranknow    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ open       : num  135 134 144 139 116 ...
 $ high       : num  136 147 147 140 126 ...
 $ low        : num  132.1 134 134.1 107.7 92.3 ...
 $ close      : num  134 145 139 117 105 ...
 $ volume     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ market     : num  1.50e+09 1.49e+09 1.60e+09 1.54e+09 1.29e+09 ...
 $ close_ratio: num  0.544 0.781 0.384 0.288 0.388 ...
 $ spread     : num  3.88 13.49 12.88 32.17 33.32 ...
>
```

The following are descriptions of each variables:

$ **open**: The $ amount in US Dollars that the day started at
$ **high**: The highest $ amount it got to in US dollars that day
$ **low**: The lowest $ amount it got to in US dollars that day
$ **close**: The $ amount in US dollars that the day finished at
$ **volume**: The $ value in US dollars of how many were exchanged that day
$ **market**: The total amount of market capital (combined worth) in US dollars
$ **Close**_ratio = The daily close rate, min-maxed with the high and low values for the day.
$ **Spread** = The $USD difference between the high and low values for the day.

Moreover, data is collected from 28/04/2013 to 25/03/2018 and it contains 1793 records or 257 weeks. The basic statistics values for the dataset are as following:

nobs	1.793000e+03
NAs	0.000000e+00
Minimum	6.843000e+01
Maximum	1.949740e+04
1. Quartile	2.780900e+02
3. Quartile	9.197500e+02
Mean	1.635309e+03
Median	4.981700e+02
Sum	2.932110e+06
SE Mean	7.396932e+01
LCL Mean	1.490234e+03
UCL Mean	1.780385e+03
Variance	9.810329e+06
Stdev	3.132144e+03
Skewness	3.122398e+00
Kurtosis	9.786500e+00

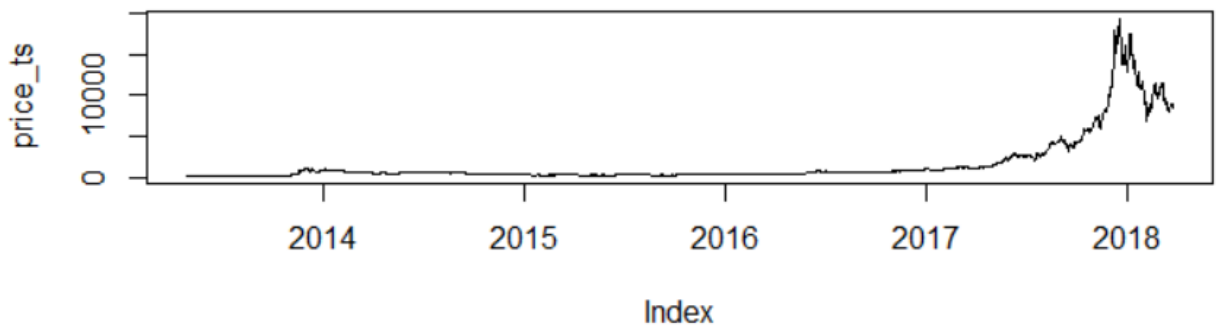
Among all available variable, close price is chosen as independent variable.

3. Problems to be Solved

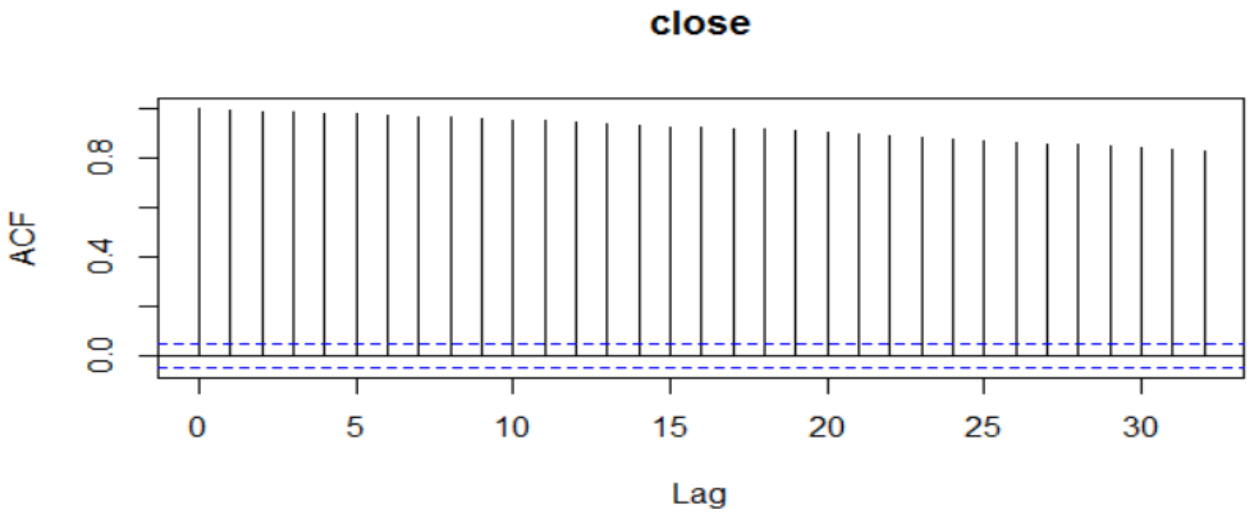
Bitcoin has attracted attention of so many investors these days. It has had so much variation from the first day of its creation. Creating a time-series model to find a specific pattern will be so effective on investment decisions. So, in this project we are going to apply data analytic techniques to investigate the data and figure out whether forecasting the trend is possible or not.

4. Data Processing

Before making any time-series model we must make sure that time-series data is stationary and serial dependence. To do so, for stationary test linear time series is plotted.



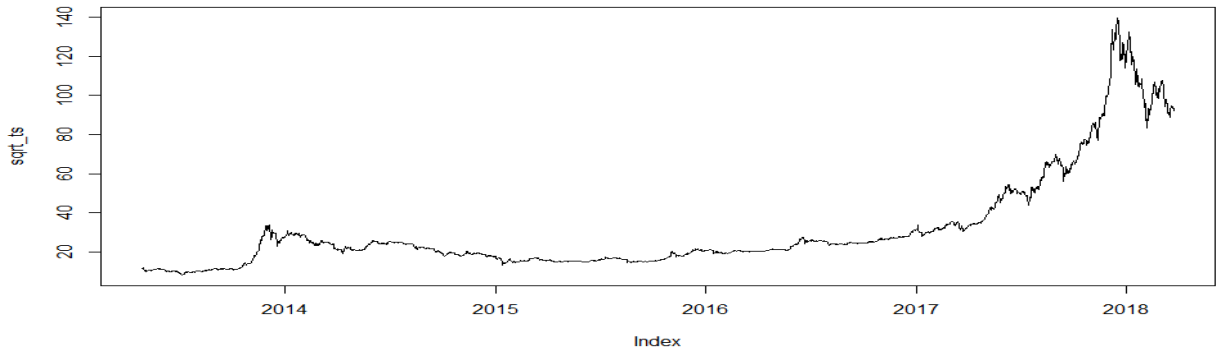
It is cleared that the plot is not stationary. Both mean and variance is changed over time. Also, we must check serial dependency. ACF plot is needed to figure out whether data is stationary or not.



As it is clear from the plot, it is not decayed quickly. So, we need to apply data processing. To achieve to a reliable data, we applied different types of data transformation on the dataset. To see whether each transformation technique makes the data reliable or not the time-series plot is created for each one.

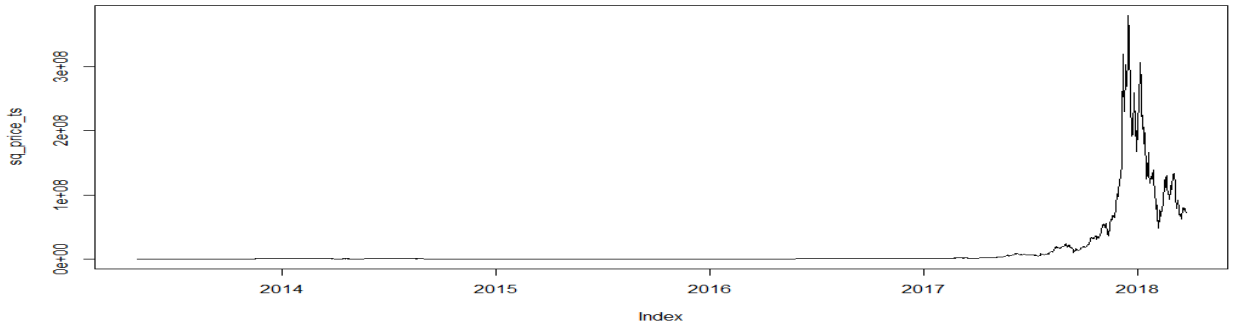
4.1.Transformation

4.1.1. SQRT



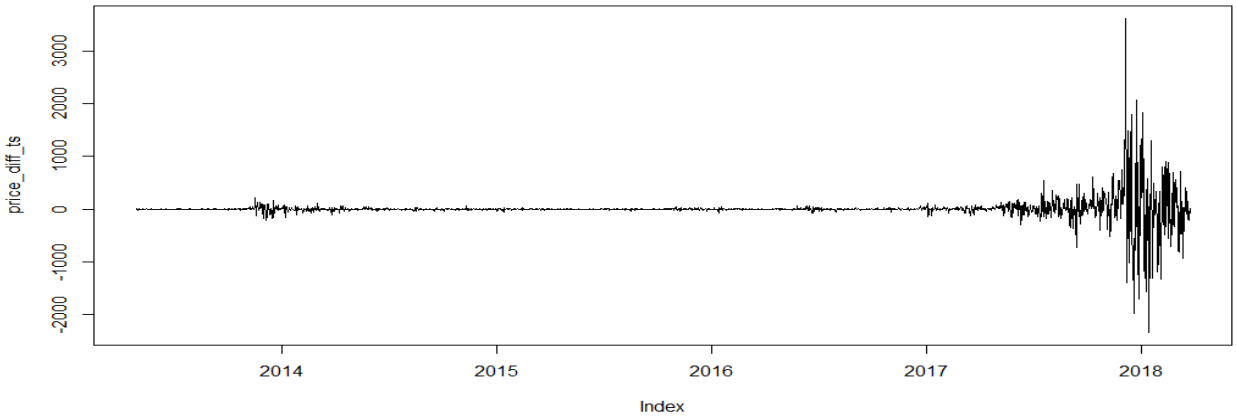
As it is clear, there is not a meaningful change on the data by applying sqrt transformation.

4.1.2. Square



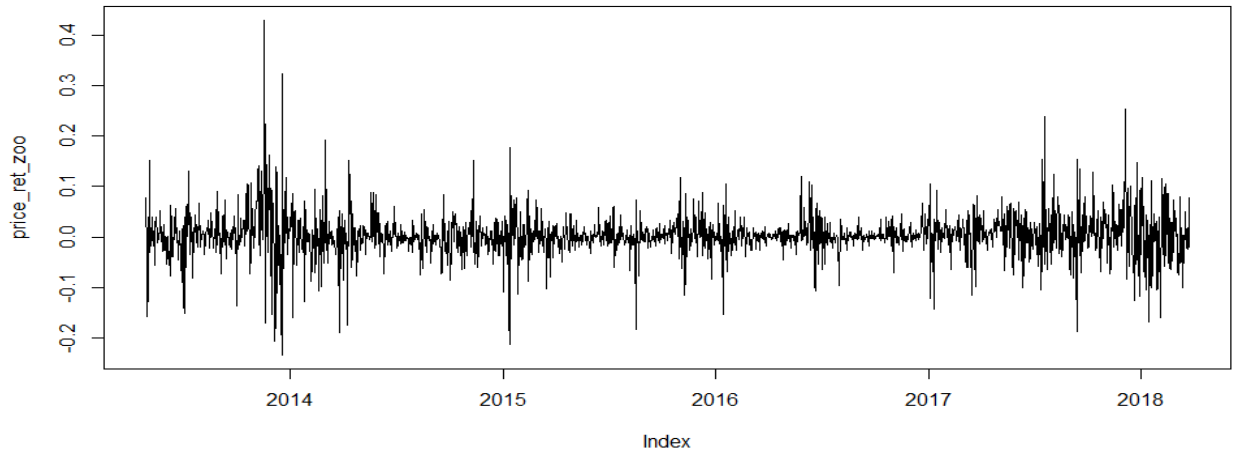
Also, the square transformation does not make a difference on the result.

4.1.3. Differencing



After applying differencing, although mean does not change over time, variance does. So, data still is not stationary.

4.1.4. Return



Return transformation has made the data more reliable. The mean does not change over time and variance changes are smoother. To make sure if there is other transformation with better result, return transformation is applied on weekly data set. I mean daily data is converted to weekly data and each week has average value of the seven days of the week.

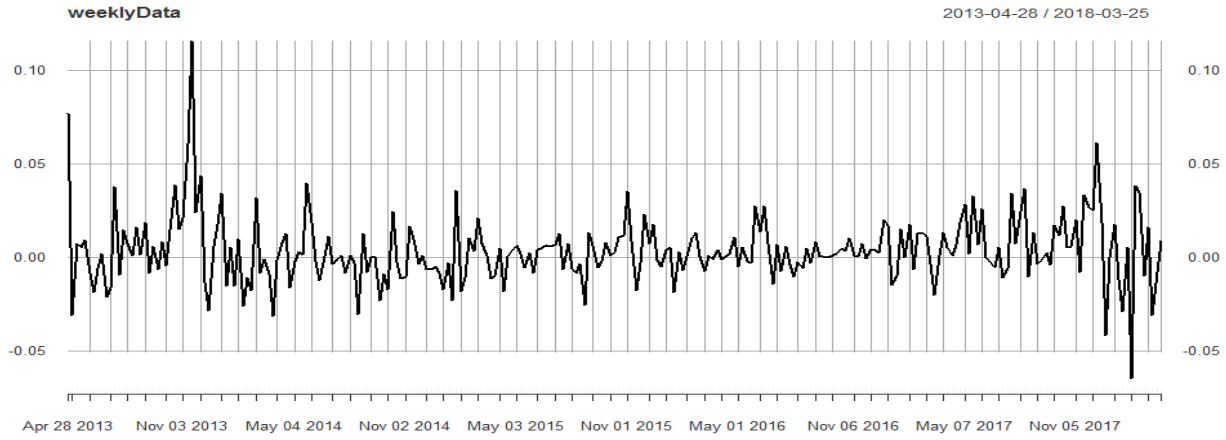
4.2. Convert Daily Data to Weekly Data

ACF plot for return daily data did not meet the stationary requirement. So, we transformed the return daily data to weekly to see whether data become stationary or not.

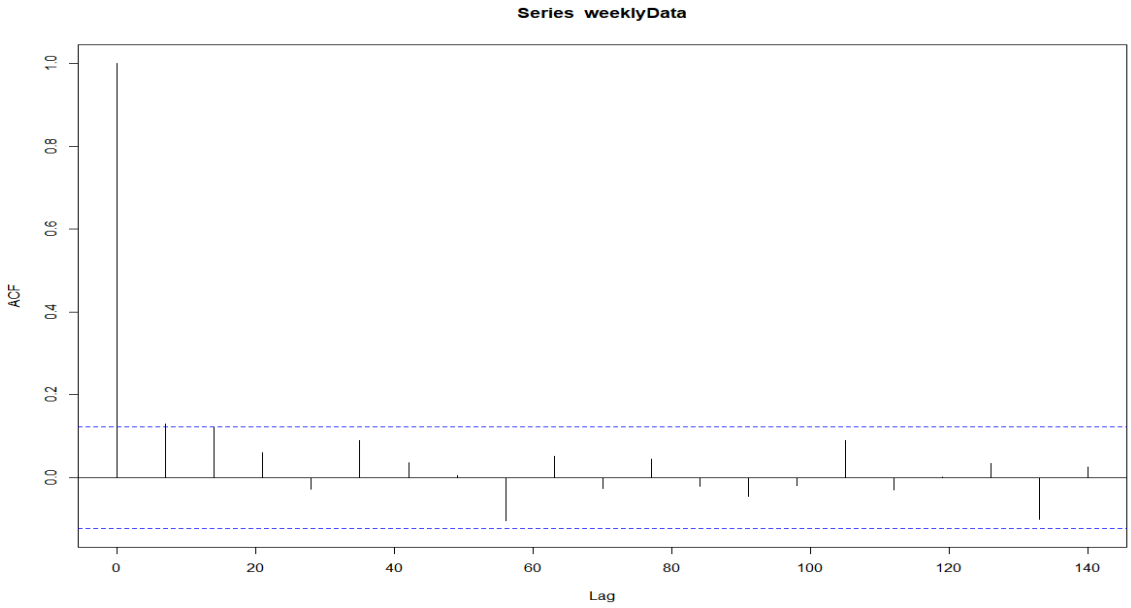
4.3. Check the Requirements

4.3.1. Stationary

To check the stationary of data set we need to draw the time-series plot and ACF plot.



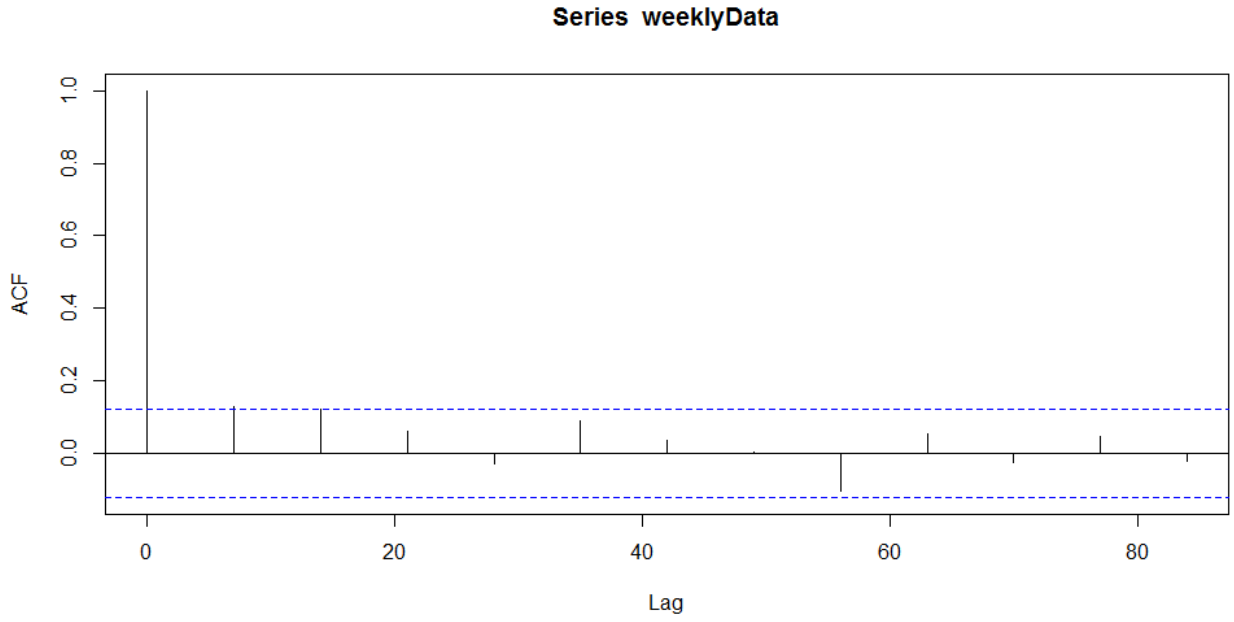
Return transformation on weekly data has made the data smoother. As seen on the plot, mean does not change over time and variance changes are reliable to make sure the data is stationary. Also, the ACF plot is as below.



According to the above ACF plot, after lag 2 all other values are zero so we can confirm the stationary requirement.

4.3.2. Serial Dependency

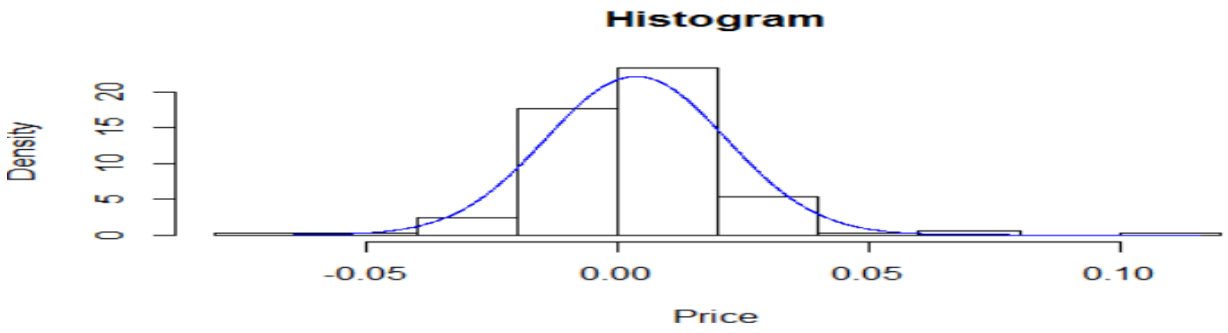
To figure whether the data is serially dependent or not, ACF plot is created.

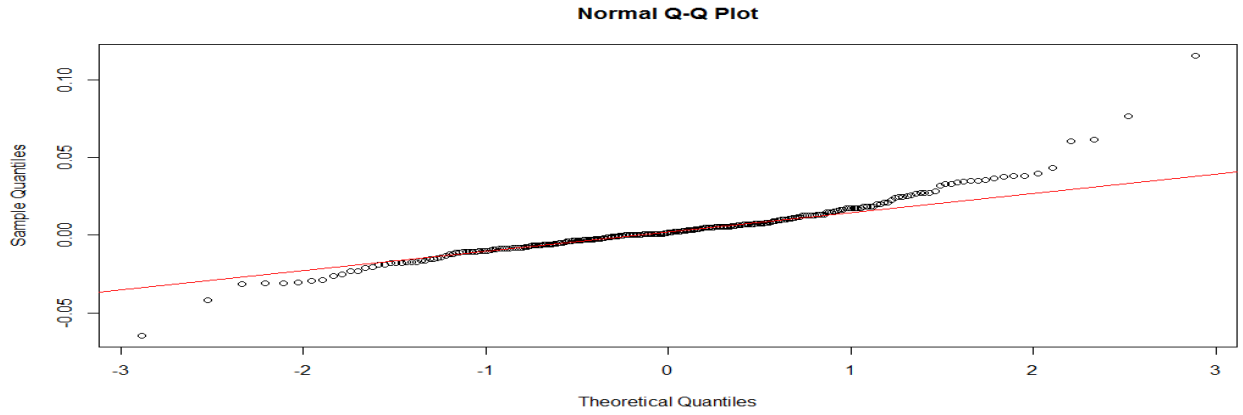


According to the ACF plot, there are some significantly non-zero auto correlation values. Thus, it is serially correlated.

4.3.3. Normality Test

To figure out the data is normally distributed or not, Histogram is created.





According to the above plots, we can confirm that our data follows normal distribution.

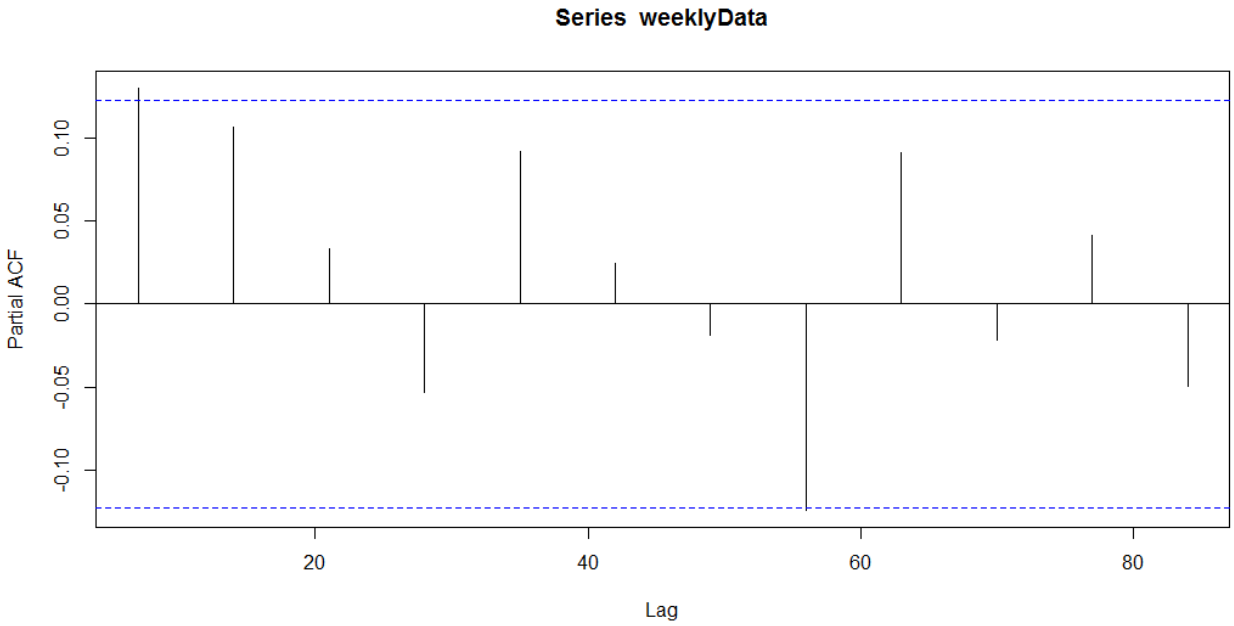
5. Methods and Process

To find best model for the price in future, different models are built such as AR, MA, ARMA and ARIMA model.

5.1.AR Model

5.1.1. Model Creation

To build AR model, determination of value of p is required. To find the value, PACF plot is created.



According to the plot, PACF tails off at lag 7. So, the AR model with the p value of 7 is built.

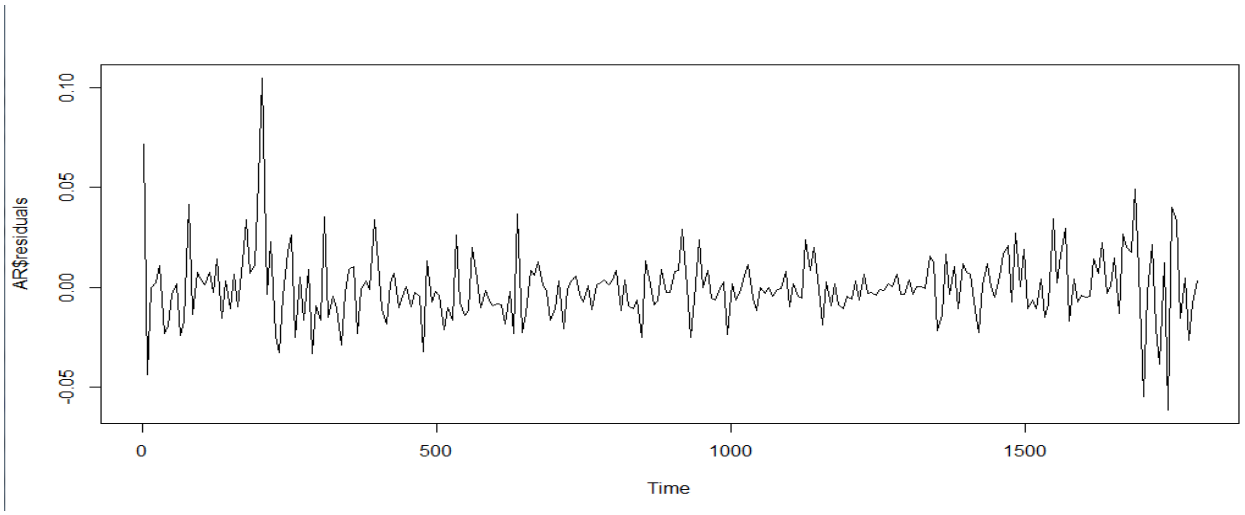
```
Call:
arima(x = weeklyData, order = c(7, 0, 0))

Coefficients:
          ar1     ar2     ar3      ar4     ar5
      0.1226  0.1122  0.0281  -0.0783  0.1043
s.e.  0.0649  0.0662  0.0664   0.0663  0.0666
          ar6      ar7  intercept
      0.0239  -0.0258     0.0037
s.e.  0.0669   0.0666     0.0015

sigma^2 estimated as 0.0003078:  log likelihood = 674.
31,  aic = -1330.63
```

AR model

$X_t = 2.6381 * 10^{-3} + 0.1226 * X_{t-1} + 0.1122 * X_{t-2} + 0.0281 * X_{t-3} - 0.0783 * X_{t-4} + 0.1043 * X_{t-5} + 0.0239 * X_{t-6} - 0.0258 * X_{t-7} + a_t$

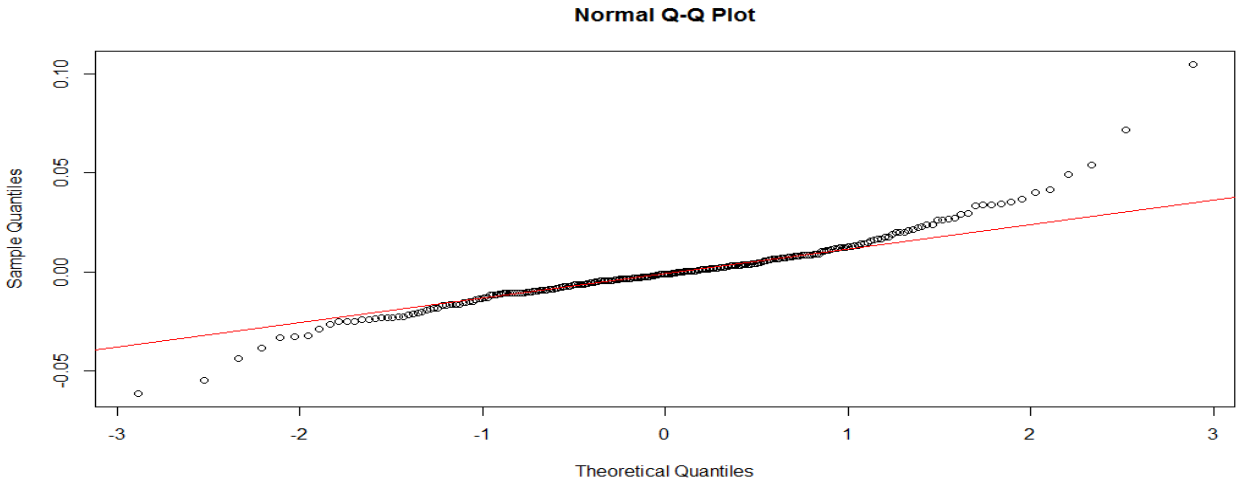
5.1.2. Residual analysis

Residual time plot is created.



5.1.2.1. *Normality test*

Normality for the model is validated by QQ plot.

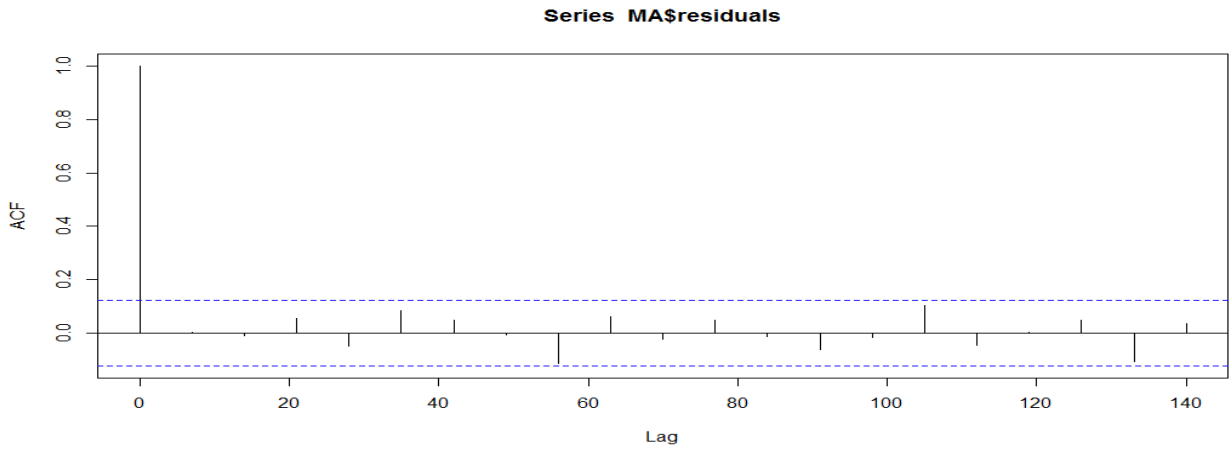


According to the plot, most of the spots are fallen on the line. So, it validates normally distribution.

```
> Box.test(AR$residuals, lag=20, type='Ljung')

        Box-Ljung test

data:  AR$residuals
X-squared = 15.964, df = 20, p-value = 0.7189
```

Result of Ljung normality test (p-value>0.05) confirms the normally distribution of the residuals.

To check whether residuals are white noise or not ACF plot is created.

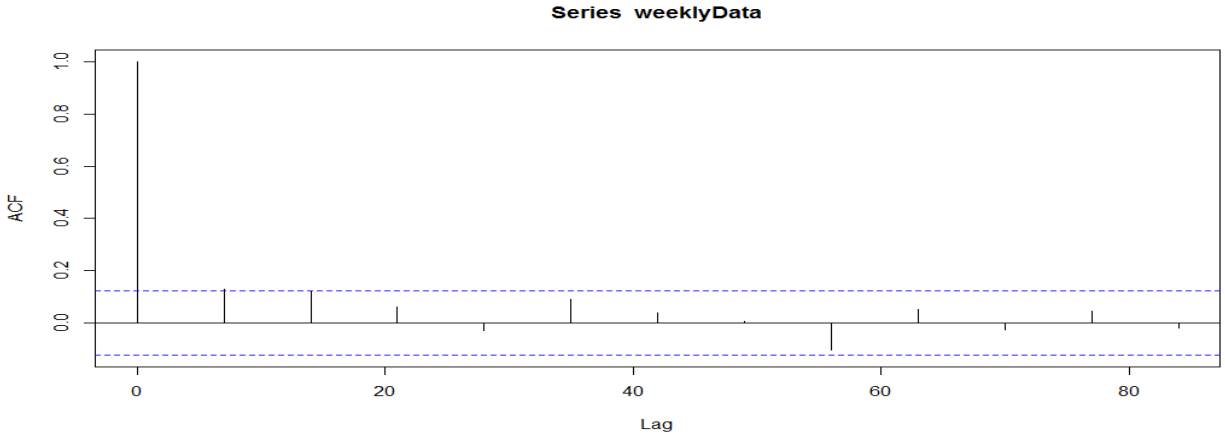


Since all the values of the residual ACF plot are zero, it validates that residuals are white noise which meets the assumption on residual analysis.

5.2.MA Model

5.2.1. Model Creation

To build AR model, determination of value of q is required. To find the value, ACF plot is created.



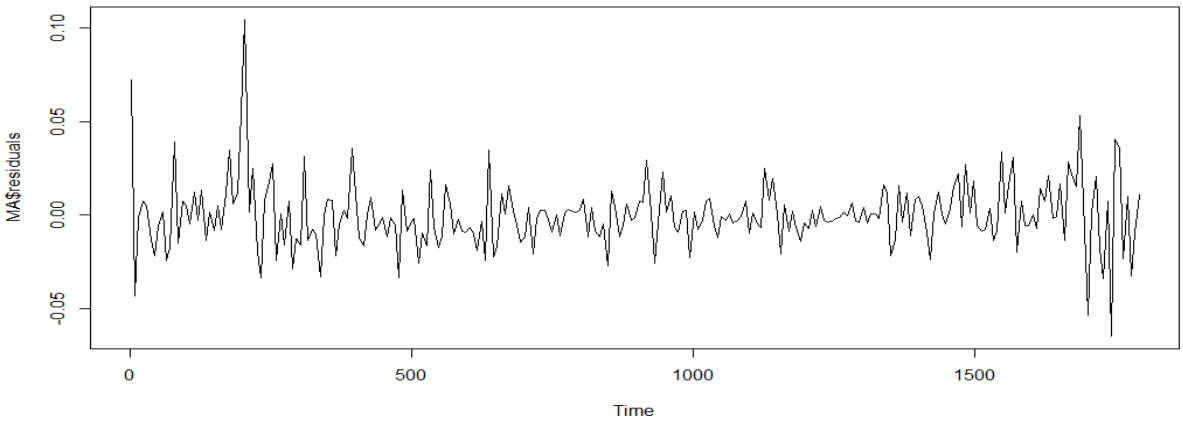
According to the plot, ACF cuts off at lag 2. So, MA (2) is created.

```
> MA

Call:
arima(x = weeklyData, order = c(0, 0, 2))

Coefficients:
          ma1     ma2  intercept
       0.1081  0.1362     0.0037
s.e.   0.0637  0.0715     0.0014

sigma^2 estimated as 0.0003128:  log likelihood = 672.31,  aic = -1336.63
```

MA model:

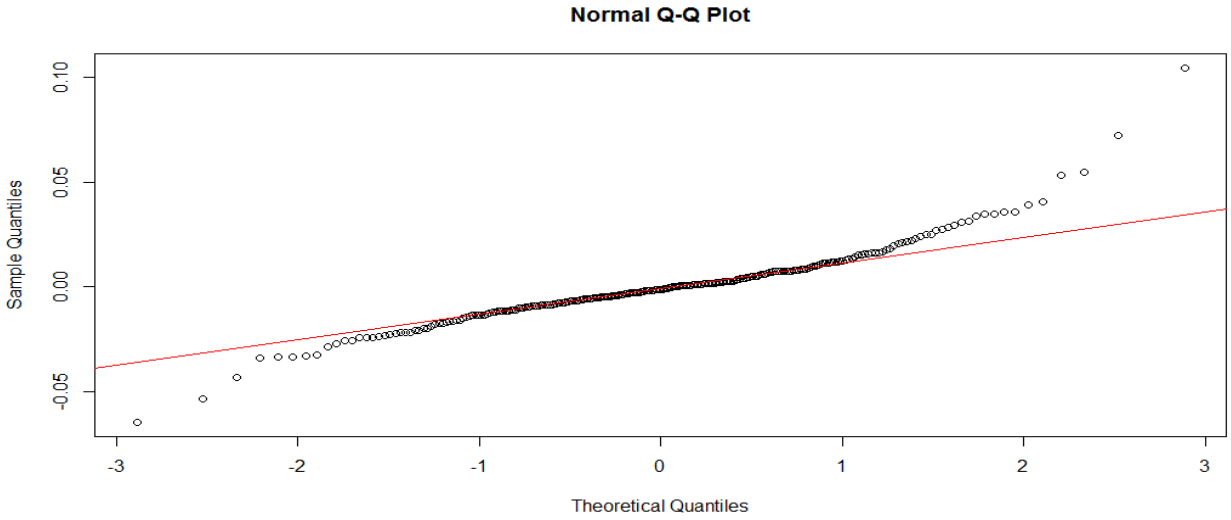
$r_t = 3.62 * 10^{-3} + a_t - 0.1081 * a_{t-1} - 0.1362 * a_{t-2}$

5.2.2. Residual Analysis



5.2.2.1. Normality test

Normality for the model is validated by QQ plot.



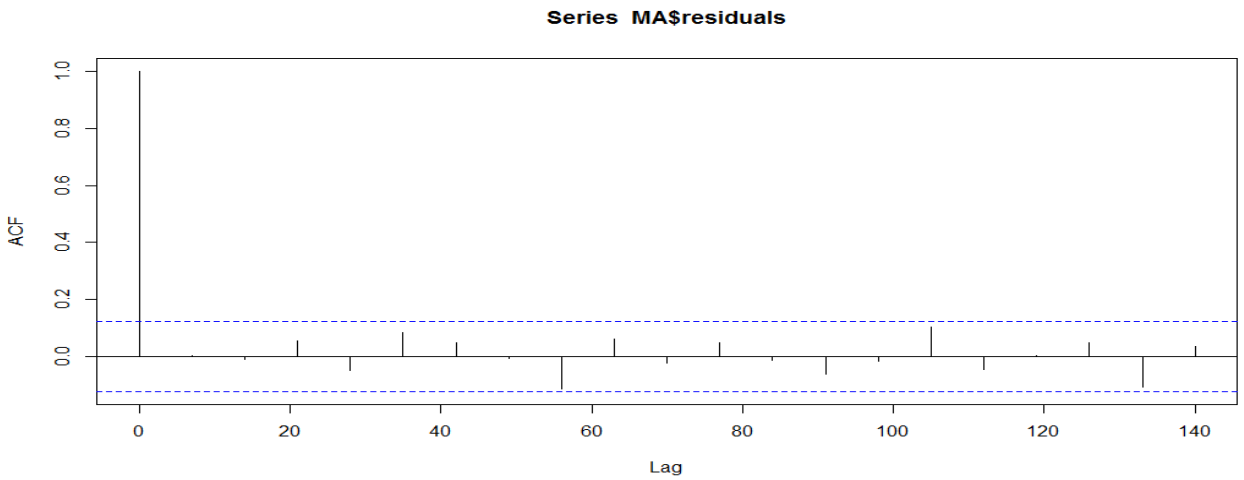
According to the plot, most of the spots are fallen on the line. So, it validates normally distribution.

```
> Box.test(MA$residuals, lag=20, type='Ljung')

        Box-Ljung test

data:  MA$residuals
X-squared = 18.251, df = 20, p-value = 0.5708
```

Result of Ljung normality test (p-value>0.05) confirms the normally distribution of the residuals.

5.2.2.2. White Noise Check

To check whether residuals are white noise or not ACF plot is created.



Since all the values of the residual ACF plot are zero, it validates that residuals are white noise which meets the assumption on residual analysis.

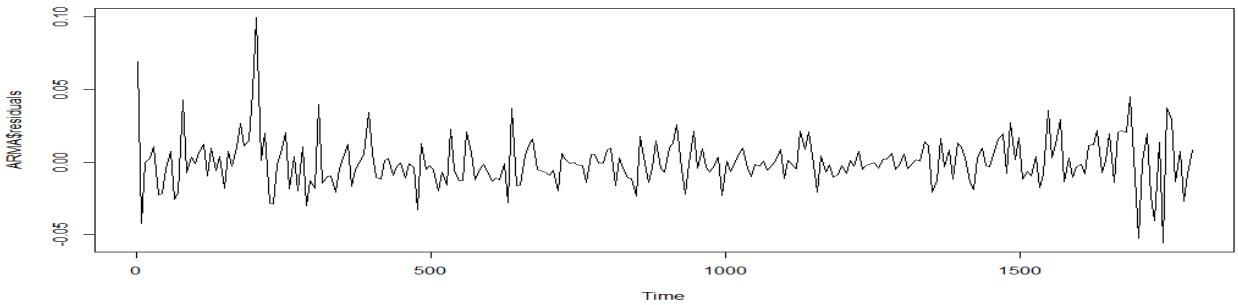
5.3.ARMA Model

5.3.1. Model Creation

As seen on MA and AR models, ACF plot tails of at lag 7 and PACF plot tails off at lag 2. So, p and q for the ARMA model are 7 and 2, respectively.

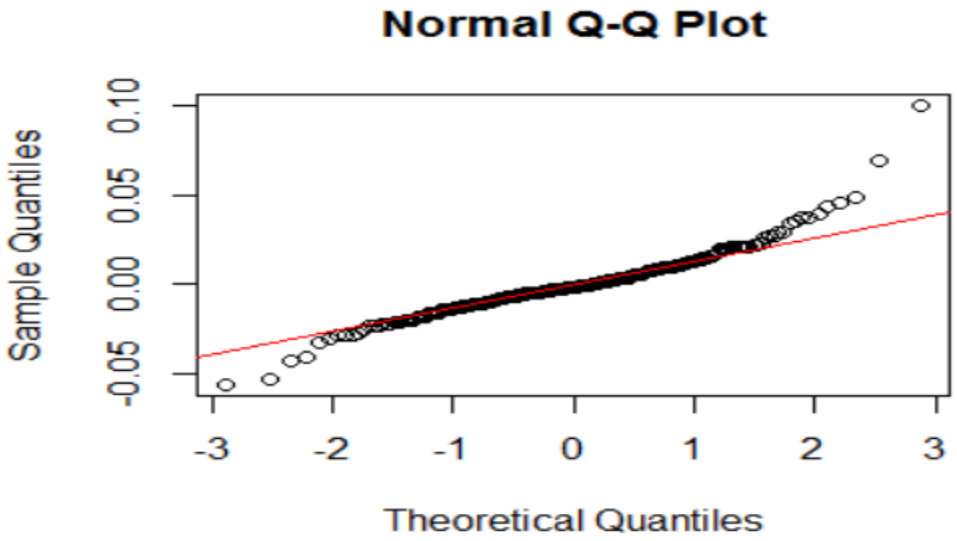
```
> ARMA

Call:
arima(x = weeklyData, order = c(7, 0, 2))

Coefficients:
          ar1      ar2     ar3     ar4     ar5     ar6     ar7     ma1
      -0.0837  -0.8010  0.1671  0.0225  0.1062  0.0042  0.1211  0.2063
s.e.   0.0657   0.0648  0.0830  0.0841  0.0834  0.0655  0.0664  0.0151
          ma2  intercept
       1.0000     0.0037
s.e.   0.0165     0.0016

sigma^2 estimated as 0.0002885:  log likelihood = 680.27,  aic = -1338.55
```

5.3.2. Residual Analysis



5.3.2.1. *Normality test*

Normality for the model is validated by QQ plot.



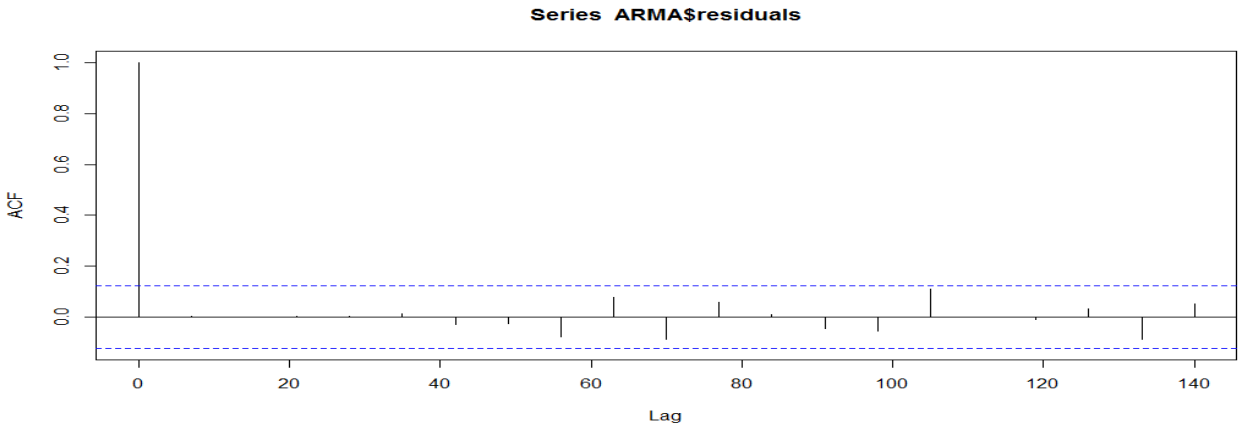
According to the plot, most of the spots are fallen on the line. So, it validates normally distribution.

```
> Box.test(ARMA$residuals, lag=20, type='Ljung')

        Box-Ljung test

data:  ARMA$residuals
X-squared = 14.319, df = 20, p-value = 0.8139
```

Result of Ljung normality test (p-value>0.05) confirms the normally distribution of the residuals.

5.3.2.2. *White Noise Check*

To check whether residuals are white noise or not ACF plot is created.



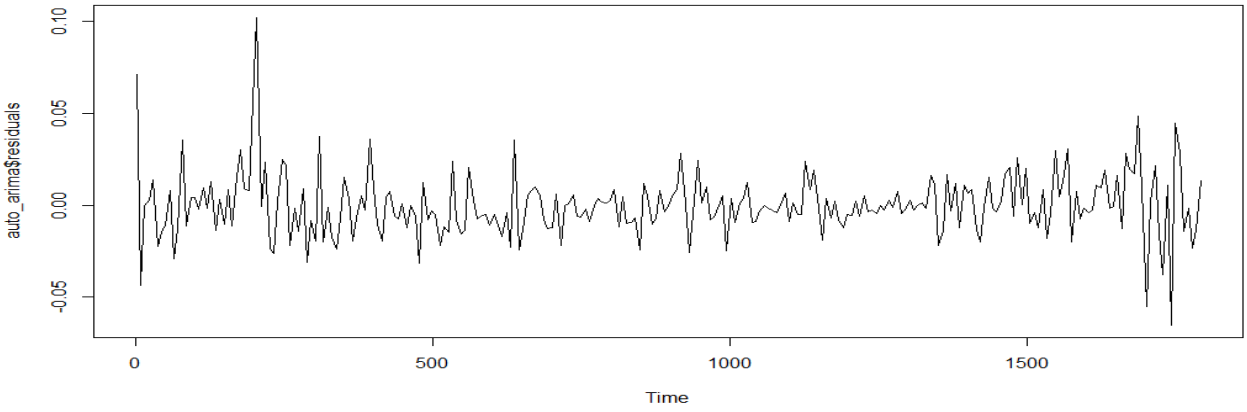
Since all the values of the residual ACF plot are zero, it validates that residuals are white noise which meets the assumption on residual analysis.

5.4.ARMA Model Using auto.arima Function

5.4.1. Model Creation

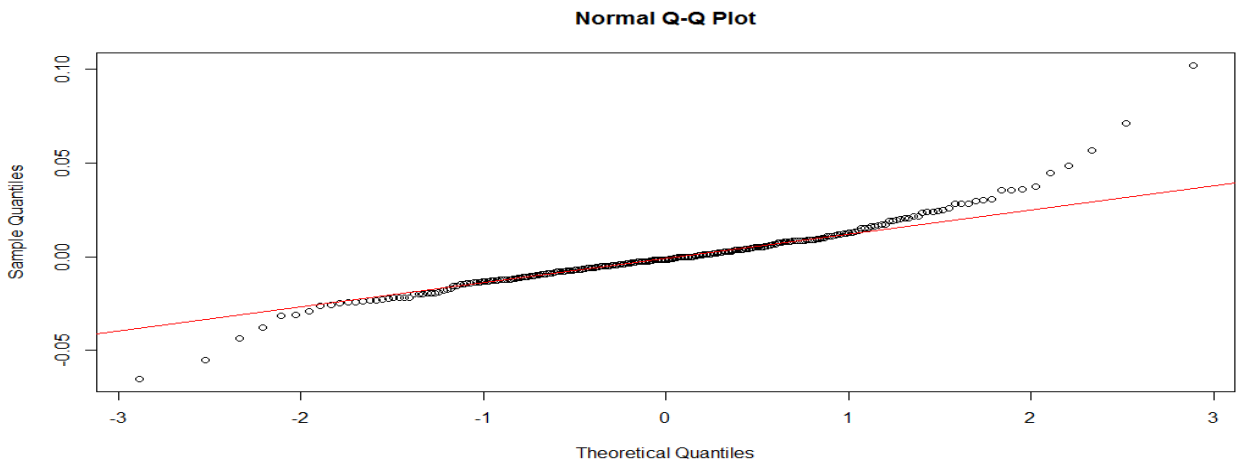
```
> auto_arima = auto.arima(weeklyData, max.p = 12, max.q = 12, ic = "aic")
> auto_arima
Series: weeklyData
ARIMA(2,0,3) with non-zero mean

Coefficients:
          ar1      ar2     ma1     ma2     ma3    mean
      -0.9564  -0.1565  1.1009  0.4080  0.2147  0.0037
s.e.   0.2672   0.2417  0.2595  0.2713  0.0763  0.0014

sigma^2 estimated as 0.0003114:  log likelihood=675.8
AIC=-1337.61   AICc=-1337.16   BIC=-1312.76
```

5.4.2. Residual Analysis



5.4.2.1. Normality test

Normality for the model is validated by QQ plot.



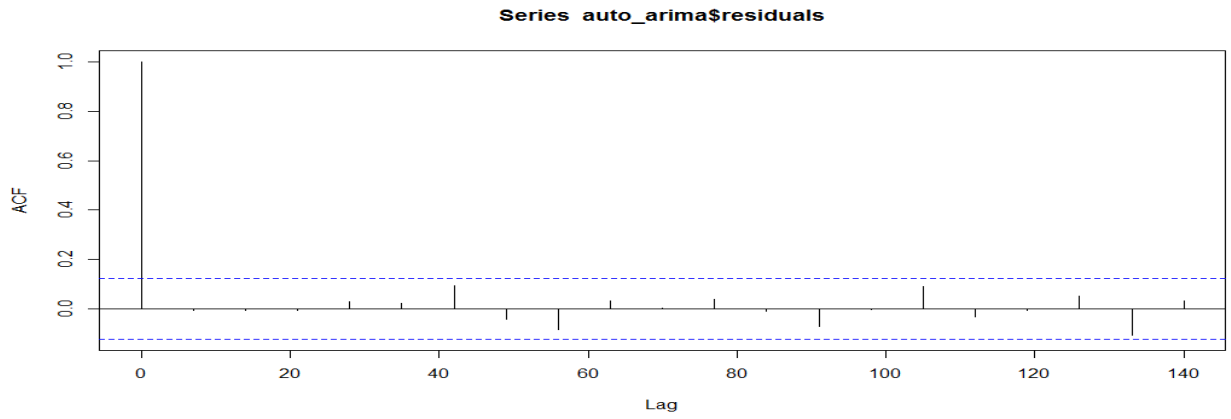
According to the plot, most of the spots are fallen on the line. So, it validates normally distribution.

```
> Box.test(auto_arima$residuals, lag=20, type='Ljung')

        Box-Ljung test

data:  auto_arima$residuals
X-squared = 14.04, df = 20, p-value = 0.8285
```

Result of Ljung normality test (p-value>0.05) confirms the normally distribution of the residuals.

5.4.2.2. White Noise Check

To check whether residuals are white noise or not ACF plot is created.



Since all the values of the residual ACF plot are zero, it validates that residuals are white noise which meets the assumption on residual analysis.

5.5.ARMA Model Using EACF Method

For this model creation, p and q are determined by EACF method. The following table show that p and q are 1 and 2, respectively.

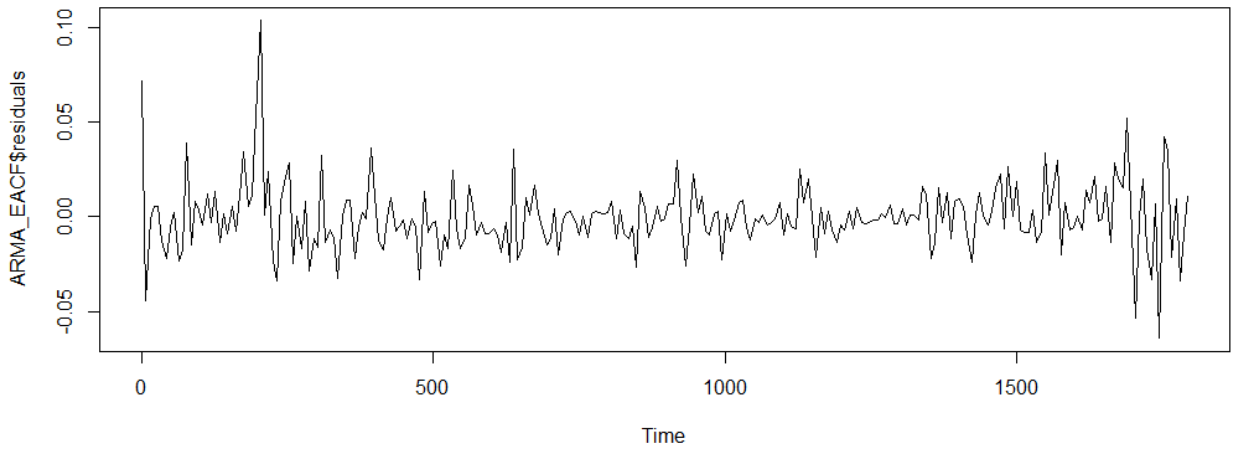
```
[1] "Simplified EACF: 2 denotes significance"
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    2    0    0    0    0    0
[2,]    2    0    0    0    0    0
[3,]    2    2    0    0    0    0
[4,]    2    0    0    2    0    0
[5,]    2    0    2    2    0    0
[6,]    2    2    2    2    2    0
>
```

p=1 q=2

5.5.1. Model Creation

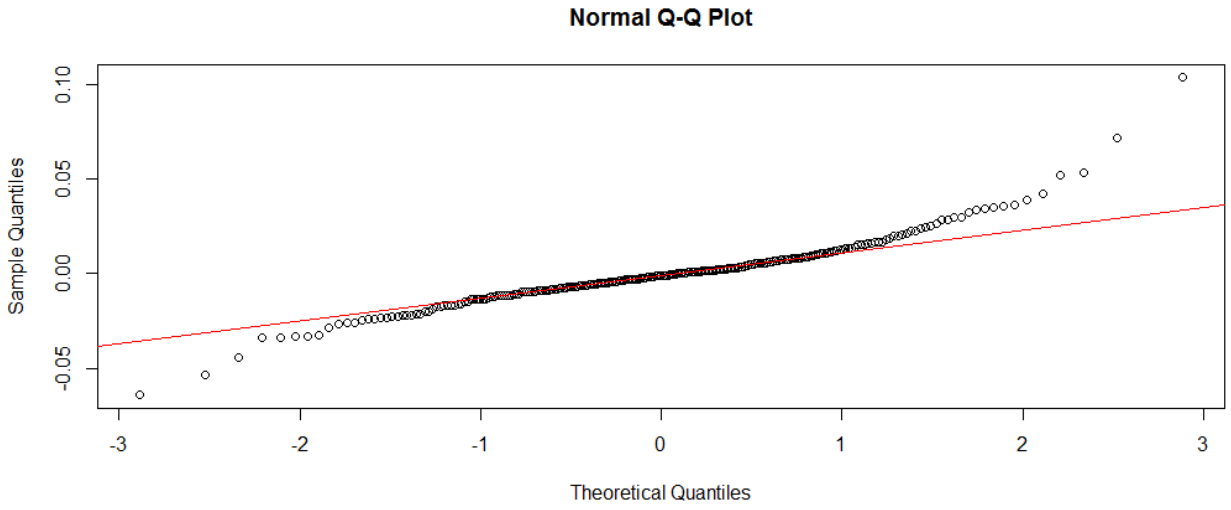
```
Call:
arima(x = weeklyData, order = c(1, 0, 2))

Coefficients:
         ar1      ma1     ma2  intercept
      0.2230  -0.1074  0.1083     0.0037
s.e.  0.4847   0.4776  0.1076     0.0014

sigma^2 estimated as 0.0003124:  log likelihood = 672.48,  aic
= -1334.96
```

5.5.2. Residual Analysis



5.5.2.1. Normality Test

Normality for the model is validated by QQ plot.



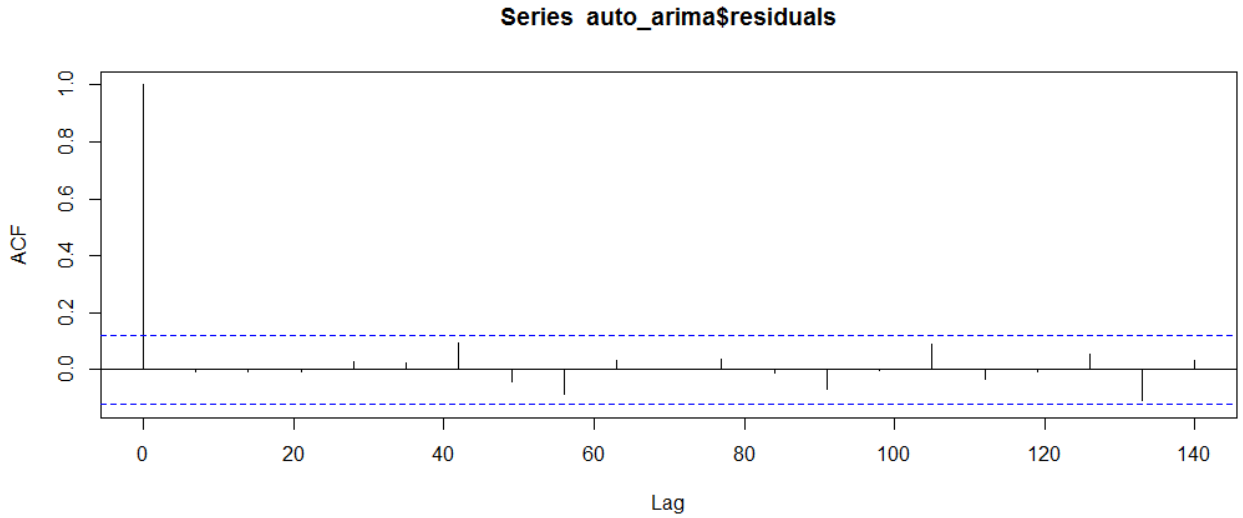
According to the plot, most of the spots are fallen on the line. So, it validates normally distribution.

```
> Box.test(ARMA_EACF$residuals, lag=20, type='Ljung')

        Box-Ljung test

data:  ARMA_EACF$residuals
X-squared = 18.23, df = 20, p-value = 0.5722
```

Result of Ljung normality test (p-value>0.05) confirms the normally distribution of the residuals.

5.5.2.2. White Noise Check

To check whether residuals are white noise or not ACF plot is created.



Since all the values of the residual ACF plot are zero, it validates that residuals are white noise which meets the assumption on residual analysis.

5.6. Models Selection

To select which model is the best fit for the data, AIC is chosen as a metric.

	AR (7)	MA (2)	ARMA (7,2)	ARMA (2,3)	ARMA (1,2)
AIC	-1330.63	-1336.63	-1338.55	-1337.61	-1334.96

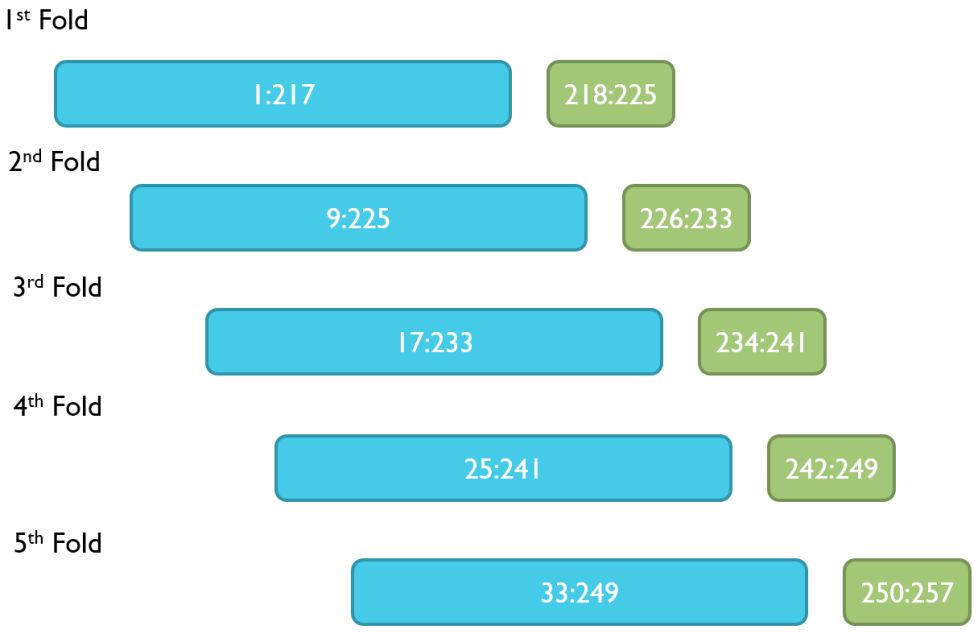
AIC values for different models are so close together. So, at this time we cannot claim which one is the best fit. We must apply evaluation on all the models to figure out the best one.

6. Evaluations and Results

6.1. Five-fold cross validation

Because number of the data are too small, validation method must be N-fold cross validation. To do so, data is separated to 5 fold. Each fold contains of 217 training data and 8 test data. For example, first fold's training data set starts from first data and ends to 217th data and its test data set start from 218th data and ends to 225th data.

In a same way, second fold's training data set starts from 9th data and ends to 225th data and its test data set start from 226th data and ends to 233rd data. Other folds follow the same pattern.



The validation method is applied for all the models.

A sample of cross folding is shown:

```
# Fold = 3
train3 = weeklyData[17:233, ]
test3 = weeklyData[234:241, ]

# AR model
AR3=arima(train3, order = c(7,0,0))

# MR model
MA3=arima(train3, order = c(0,0,2))

# ARIMA model
ARMA3=arima(train3, order = c(7,0,2))

# auto Arima model
auto_arima3 = auto.arima(train3, max.p = 30, max.q = 30, ic = "aic")

#ARMA_EACF
ARMA_EACF3 = arima(train3, order = c(1,0,2))
```

Then accuracy is applied for each fold with eight predictions ahead to compare with the actual data of the test data set.

```
accuracy(forecast(AR3, 8), test3)
accuracy(forecast(MA3, 8), test3)
accuracy(forecast(ARMA3, 8), test3)
accuracy(forecast(auto_arima3, 8), test3)
accuracy(forecast(ARMA_EACF3, 8), test3)
```

To figure out which model fits the best, MAE is considered as a metric. Since there are five folds for each model, mean value of five related MAE is considered as each model's MAE.

6.2.Results and Findings

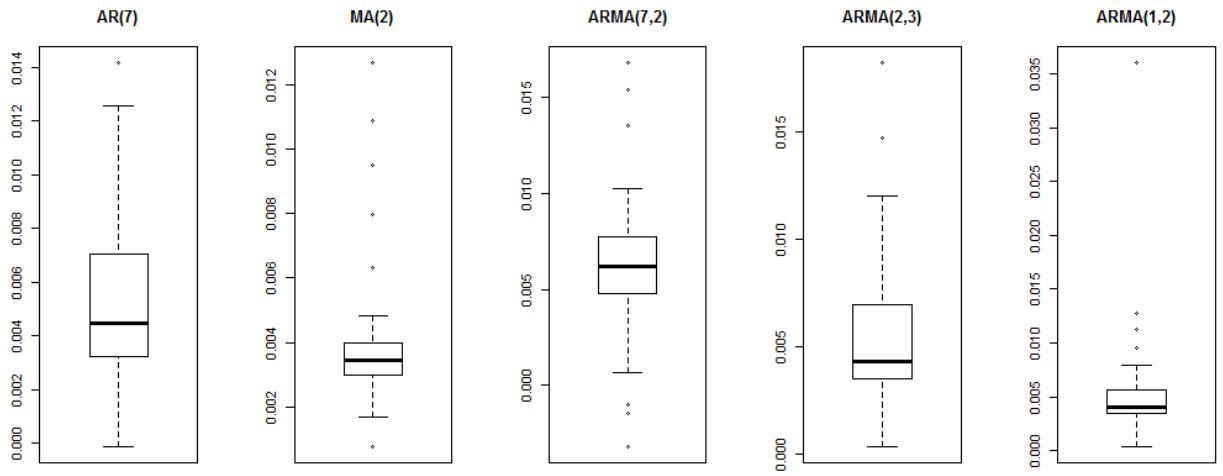
As mentioned earlier, mean MAE value of five folds of each model is considered as the model's MAE. The result is shown in the following table:

	AR	MA	ARMA(7,2)	ARMA(2,3)	ARMA(1,2)
MAE	0.02670786	0.026707	0.02649215	0.02662835	0.02670202

6.2.1. Two-Paired Sample Hypothesis Testing

As seen in the table, MAE values for all models are too close. However, ARMA(7,2) is maintaining the least value of MAE. So, we need to apply two-paired hypothesis testing on the ARMA(7,2) with all other models.

At first, we need to create box plot of the predicted values of different models to see is it possible to compare the different models by their median values or not.



Because group variance is large in the models' box plot it is not reliable to compare the groups based on their medians. In other words, we cannot use box plot to make solid conclusion.

As we mentioned earlier, we need to apply hypothesis testing on the group of data.

```
ARMA_1 = ARMA
ARMA_2 = ARIMA
ARMA_3 = ARMA_EACF

diff_1 = ARMA_1 - ARMA_2
diff_2 = ARMA_1 - ARMA_3
diff_3 = ARMA_1 - AR
diff_4 = ARMA_1 - MA

z.test(diff_1, NULL, alternative = "two.sided", mu = 0, sigma.x=sd(diff_1), sigma.y=NULL, conf.level = 0.95)
z.test(diff_2, NULL, alternative = "two.sided", mu = 0, sigma.x=sd(diff_2), sigma.y=NULL, conf.level = 0.95)
z.test(diff_3, NULL, alternative = "two.sided", mu = 0, sigma.x=sd(diff_3), sigma.y=NULL, conf.level = 0.95)
z.test(diff_4, NULL, alternative = "two.sided", mu = 0, sigma.x=sd(diff_4), sigma.y=NULL, conf.level = 0.95)
```

ARMA(7,2) with ARMA(2,3)

Null hypothesis => μ ARMA (7, 2) = μ ARMA (2, 3)

Alternative hypothesis => μ ARMA (7, 2) ≠ μ ARMA (2, 3)

```
> z.test(diff_1, NULL, alternative = "two.sided", mu = 0, sigma.x=sd(diff_1), sigma.y=NULL, conf.level =
0.95)

        One-sample z-Test

data:  diff_1
z = 0.7727, p-value = 0.4397
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.0004739454  0.0010908527
sample estimates:
   mean of x
0.0003084536
```

As it is clear, since our confidence level is 95% and p-value is greater than 0.05, so we can accept the null hypothesis and say that the mean of these two groups of data are equal.

ARMA(7,2) with ARMA(1,2)

Null hypothesis => μ ARMA (7, 2) = μ ARMA (1, 2)

Alternative hypothesis => μ ARMA (7, 2) $\neq$ μ ARMA (1, 2)

```
> z.test(diff_2, NULL, alternative = "two.sided", mu = 0, sigma.x=sd(diff_2), sigma.y=NULL, conf.level =
0.95)

        One-sample z-Test

data:  diff_2
z = -1.067, p-value = 0.286
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.005286921  0.001559564
sample estimates:
   mean of x
-0.001863678
```

As it is clear, since our confidence level is 95% and p-value is greater than 0.05, so we can accept the null hypothesis and say that the mean of these two groups of data are equal.

ARMA(7,2) with AR (7)

Null hypothesis => $\mu_{ARMA\ (7,\ 2)} = \mu_{AR\ (7)}$

Alternative hypothesis => $\mu_{ARMA\ (7,\ 2)} \neq \mu_{AR\ (7)}$

```
> z.test(diff_3, NULL, alternative = "two.sided", mu = 0, sigma.x=sd(diff_3), sigma.y=NULL, conf.level =
0.95)

        One-sample z-Test

data:  diff_3
z = 1.3002, p-value = 0.1935
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.000309975  0.001531737
sample estimates:
   mean of x
0.0006108808
```

As it is clear, since our confidence level is 95% and p-value is greater than 0.05, so we can accept the null hypothesis and say that the mean of these two groups of data are equal.

ARMA(7,2) with MA(2)

Null hypothesis => $\mu_{ARMA\ (7,\ 2)} = \mu_{MA\ (2)}$

Alternative hypothesis => $\mu_{ARMA\ (7,\ 2)} \neq \mu_{MA\ (2)}$

```
> z.test(diff_4, NULL, alternative = "two.sided", mu = 0, sigma.x=sd(diff_4), sigma.y=NULL, conf.level =
0.95)

        One-sample z-Test

data:  diff_4
z = 3.8883, p-value = 0.0001009
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.001022116 0.003099835
sample estimates:
  mean of x
0.002060976
```

Our confidence level is 95% and p-value is less than 0.05, so we reject the null hypothesis and say that the mean of these two groups of data are equal.

7. Conclusions and Future Work

7.1. Conclusions

Bitcoin data set had some limitation. The number of data was too small, and it was not stationary. Different types of data processing applied on the data set. Return transformation on weekly data make the met requirements for stationary and serial dependency.

Five different models built according to values of p and q based on manually selection, EACF technique and auto.arima function. Then five different models built under p and q values. Since the data was too small, we had to apply N-Fold Cross validation Technique with 217 values on training data test and 8 values on test data set with five folds. So, because test data set were different in each fold, we ended up with 217 values on training data set and 40 values on test data set. Mean Absolut Error in different considered models considered as evaluation method.

MAE values were too close together and we couldn't confidently say which model was the best. So, we applied two paired hypothesis testing between the model with lowest MAE value with all others.

Final result was that among the five created models ARMA(7, 2), ARMA(2, 3), ARMA(1, 2) and AR(7) were equally qualified better than MA(2).

7.2. Limitations

AS I mentioned earlier number of data was too small. Also, the data did not follow stationary requirements.

7.3. Potential Improvements and Future Work

Since the Bitcoin is relatively a new research case study, there are different potential improvements and future works. To be more specific, some of them are:

- Applying GARCH Model to analyze variance of data.
- Applying Regression Model to find out the relations between prices and volume of the market.
- Analyzing the relations between Bitcoin price and other effective factors such as oil and gold prices.