Time Series Analysis of Bitcoin Index

Time Series Analysis Assignment 2

Bharath Narayanan Venkatesh

S4033348

30/04/25

Table of Contents

1. Introduction :

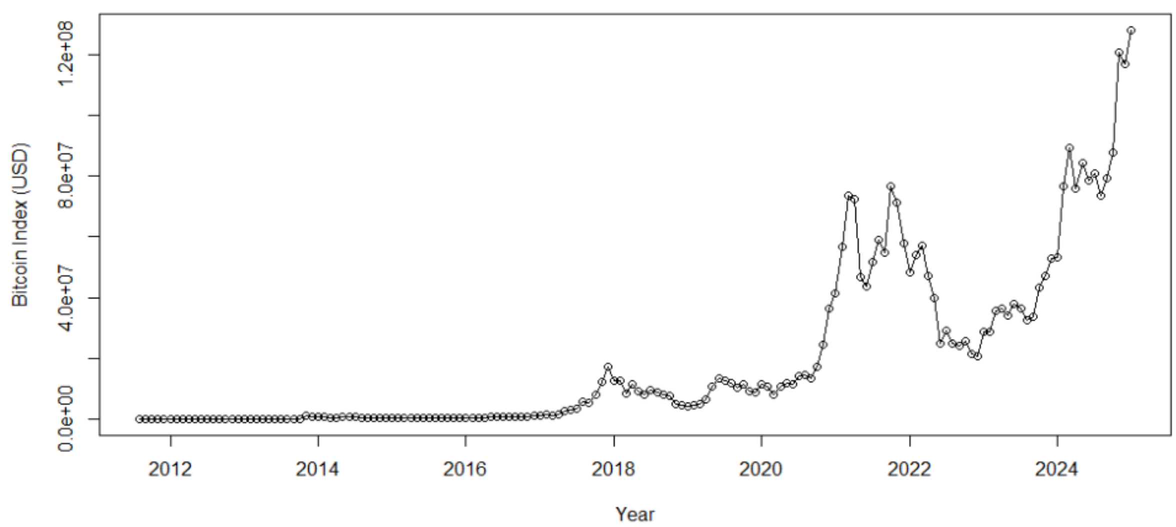
Bitcoin is one of the most important financial asset since its introduction in 2009. Bitocin has faced both incremental and decremental facets of investment. It has become one of the prime factors for doing time series analysis and modeling in the IT industry. With wide variety of data being available, lot of enthusiasts are using the data to do future predictions. Future prediction of Bitcoin is one of the critical aspects for investors, policymakers and financial analysts to make decisions on whether to invest in Bitcoins or not. In this report, I am going to analyze the Bitcoin index(USD) time series from August 2011 to January 2015 using ARIMA Models. We will conduct detailed descriptive analysis, then we will specify suitable ARIMA models using statistical tools, then fit them and finally we will select the best model based on the goodness of fit model criteria.

2. Descriptive Analysis:

The dataset consists of 162 monthly observations of Bitcoin prices. We have 2 variables, Date which holds value from Aug 2011 to Jan 2015, the class of this variable is character, and next one is Bitcoin variable which is numeric class. There are no missing values in the dataset. Finally, from summary, we can see that the Minimum value is 3988 and maximum is 128015000, median is 7930612. I am then creating time series object with ts() function. I am going to plot the object to see trend, seasonality, changing variance, behavior and change point.

From Figure 1, we can clearly see that there is a strong upward trend, there is some change in variance and seasonality especially in 2021 and 2022 and also after 2024. We could see 2 types of Behaviors both Autoregressive and Moving Average. There is no sudden change point available from the graph.

Figure 1: Bitcoin Index (USD) from Aug 2011 to Jan 2025



From Figures 2 and 3, the lag plots shows the points are closely clustered indicating strong autocorrelation at lags 1 and 2. This confirms that the data is not random and they are strongly dependent on past values.

Figure 2: Lag 1 Scatter Plot

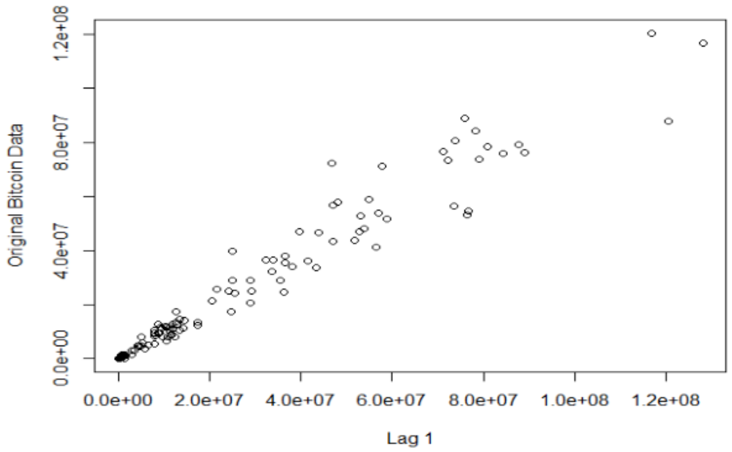
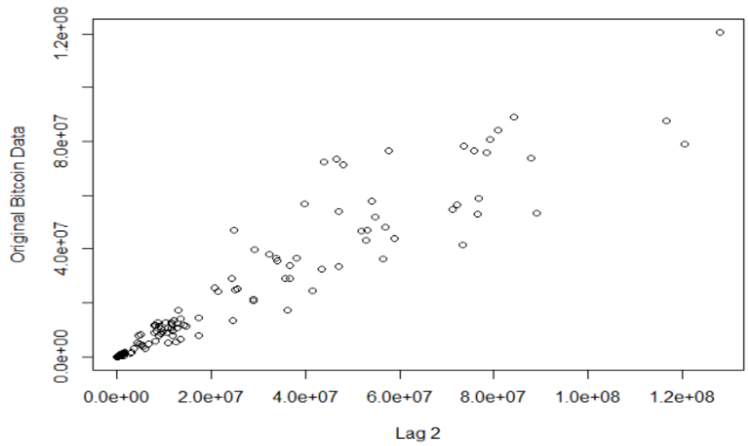


Figure 3: Lag 2 Scatter Plot



From ACF plot in Figure 4, we can see a clear pattern of slow decay and from PACF plot in Figure 5, we can see very high 1st lag. These are all typical indicators of non-stationarity. We will do some more tests to come to a clear conclusion on non-stationarity.

Figure 4: ACF of Bitcoin Index

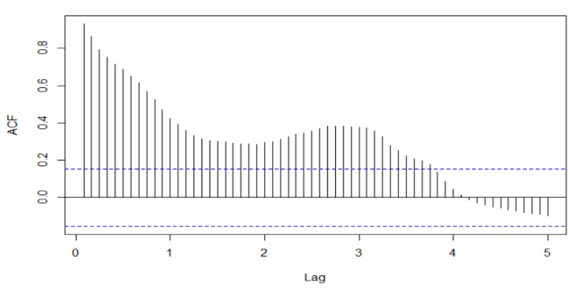
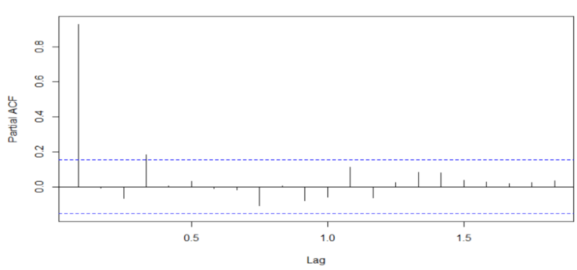


Figure 5: PACF of Bitcoin Index



In ADF test,

```
        Augmented Dickey-Fuller Test

data:  btc_ts
Dickey-Fuller = -0.27056, Lag order = 5, p-value = 0.99
```

Null Hypothesis: Series is non-stationary
Alternative Hypothesis: Series is stationary
p-value is 0.99, which is higher than 0.05, we fail to reject null hypothesis.

In PP Test,

```
        Phillips-Perron Unit Root Test

data:  btc_ts
Dickey-Fuller Z(alpha) = -2.4275, Truncation lag parameter = 4, p-value =
0.9557
```

Null Hypothesis: Series is non-stationary
Alternative Hypothesis: Series is stationary
p-value is 0.9557, which is higher than 0.05, we fail to reject null hypothesis

In KPSS Test,

```
        KPSS Test for Level Stationarity

data:  btc_ts
KPSS Level = 2.3395, Truncation lag parameter = 4, p-value = 0.01
```

Null Hypothesis: Series is stationary
Alternative Hypothesis: Series is non-stationary
p-value is 0.01, which is lesser than 0.05, we reject null hypothesis.

From all these tests, we could see that the series is non-stationary.

We will do QQ-plot and Shapiro Wilk Test to see the normality of residuals. Both the tails are way off in the QQ-Plot in Figure 6 and in the test, according to Null Hypothesis which is the data follows normal distribution and in alternative hypothesis the data does not follow normal distribution, p-value is much smaller than 0.05, so we reject null hypothesis. Finally, by looking at both QQ-plot and Shapiro Wilk Test results, we can come to a conclusion that normality is not achieved.

```
        Shapiro-Wilk normality test

data:  btc_ts
W = 0.73793, p-value = 1.102e-15
```

Figure 6: QQ plot of Bitcoin

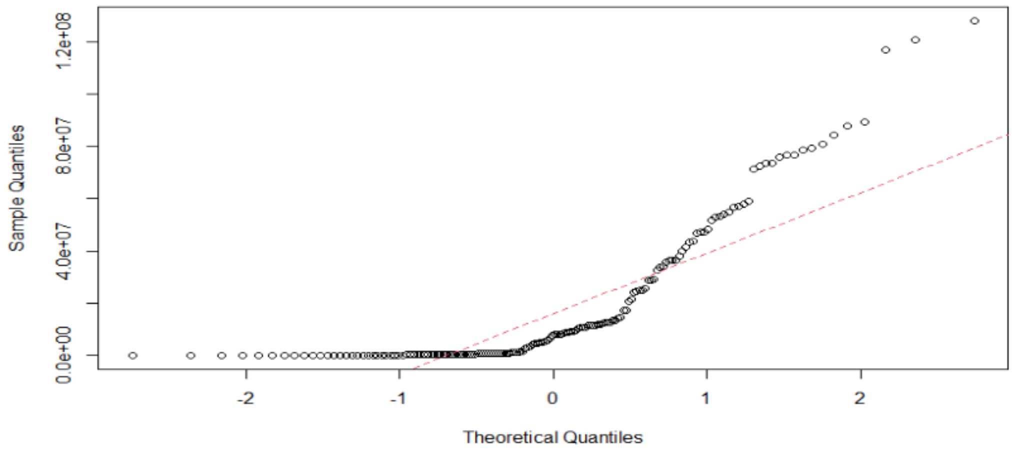


Figure 7: Log Transformation Bitcoin Index

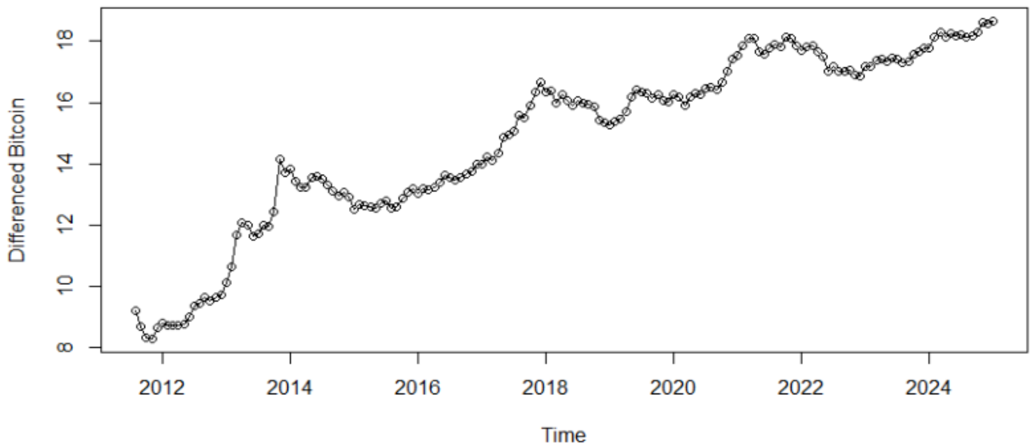


Figure 8: QQ plot of Log Transformed Data

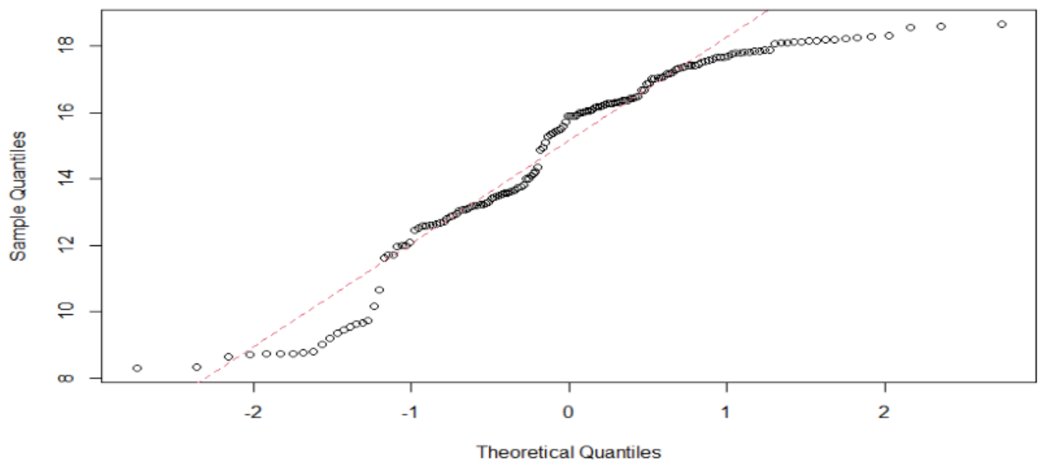


Figure 9: ACF plot of Log Transformed Data.

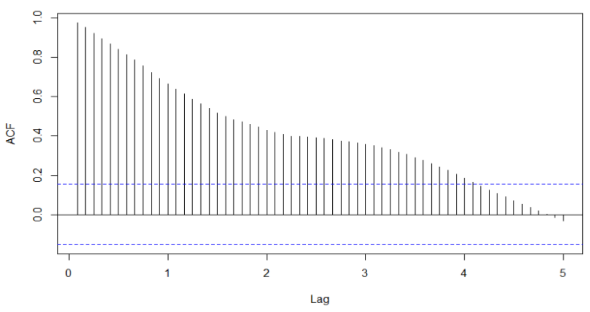
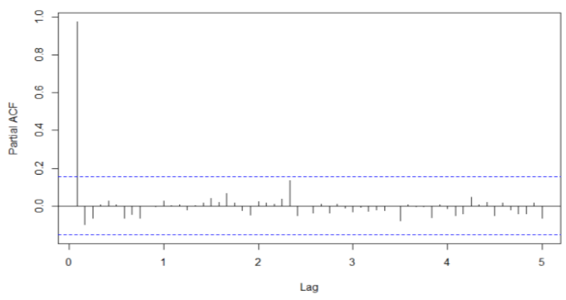


Figure 10: PACF plot of Log Transformed Data.



We are first proceeding with Log Transformation. Figure 7, shows that the log Transformation has changed the variance present in the graph and trend will remain the same as we are yet to proceed with differencing. From Figure 8, the QQ plot's tails are way off. Shapiro Wilk Test shows that p-value is less than 0.05 and we can come to a conclusion that normality is not achieved

```
        Shapiro-Wilk normality test

data:  btc_tslog
W = 0.9138, p-value = 3.363e-08
```

From Figures 9 and 10, ACF and PACF graph remains the same, as transformations will not change the characteristics of the trend.

We are proceeding with ADF Test, PP Test and KPSS Test to check non-stationarity.

```
        Augmented Dickey-Fuller Test

data:  btc_tslog
Dickey-Fuller = -2.3599, Lag order = 5, p-value = 0.4262


        Phillips-Perron Unit Root Test

data:  btc_tslog
Dickey-Fuller Z(alpha) = -8.1413, Truncation lag parameter = 4, p-value =
0.648
        KPSS Test for Level Stationarity

data:  btc_tslog
KPSS Level = 2.9962, Truncation lag parameter = 4, p-value = 0.01
```

In ADF Test and PPS Test, p-value is greater than 0.05, we fail to reject null hypothesis in both the tests. In KPSS Test, p-value is lesser than 0.05, we reject null hypothesis.
So stationarity still exists.

We will apply Box-Cox Transformation to the data. We are proceeding with sequence of values from -0.4 to 1. The true lambda value that stabilizes the variance is between 0.01 to 0.05 with 95% CI. From this, the exact lambda with maximum log likelihood is 0.05. This value is very close to 0(log transformation) but there is a slight difference of 0.05 from 0.

```
> BC$ci
[1] 0.01 0.05
> lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
> lambda
[1] 0.05
```

From Figure 11, we can see that Box-Cox transformation has also changed the variance present in the graph. Trend will remain the same as we are yet to perform differencing.

Figure 11: Time series plot of BC-transformed.

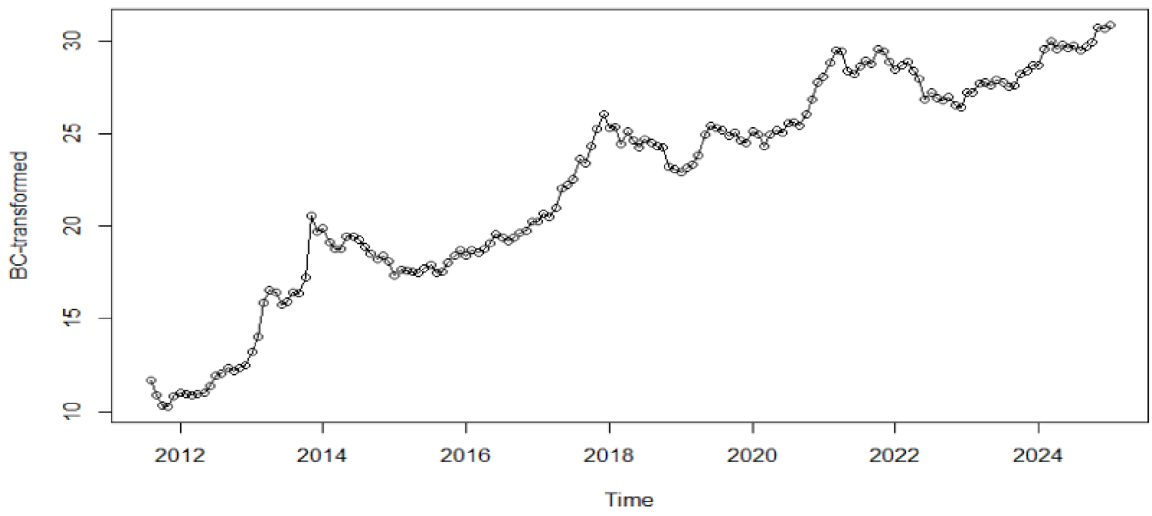
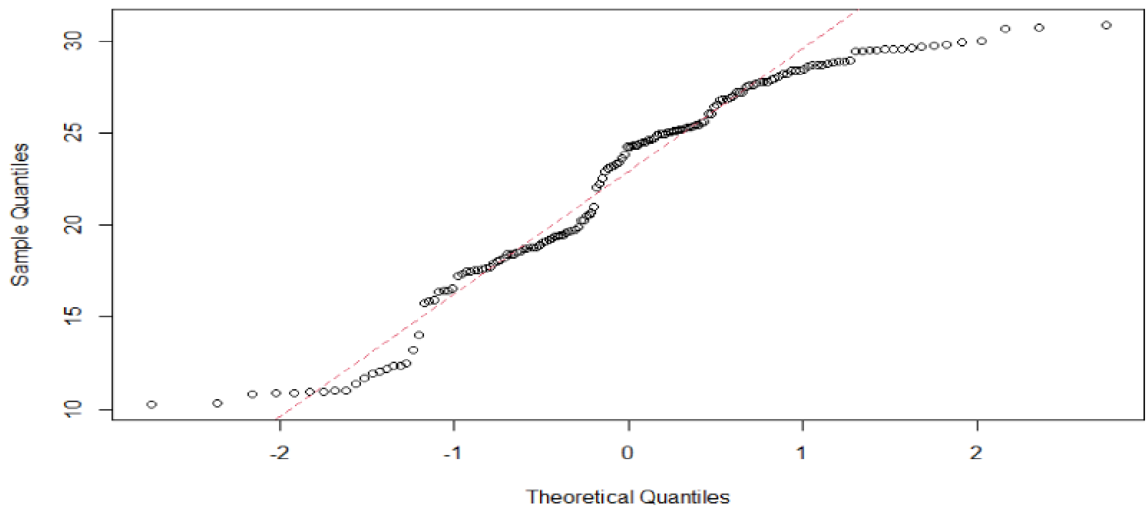


Figure 12: BC QQ plot of Bitcoin



From QQ Plot in Figure 12, we could see that both the tails are way off and Box-Cox Transformation has changed the variance present in the graph to some extent (refer Figure 11). Shapiro Wilk Test shows that the p-value is less than 0.05 and we can come to a conclusion that normality is not achieved.

```
        Shapiro-Wilk normality test

data:  BC.data
W = 0.93252, p-value = 6.461e-07
```

From Figures 13 and 14, ACF and PACF graph remains the same as transformations will not change the characteristics of the trend.

Figure 13: ACF plot of BC-transformed Data.

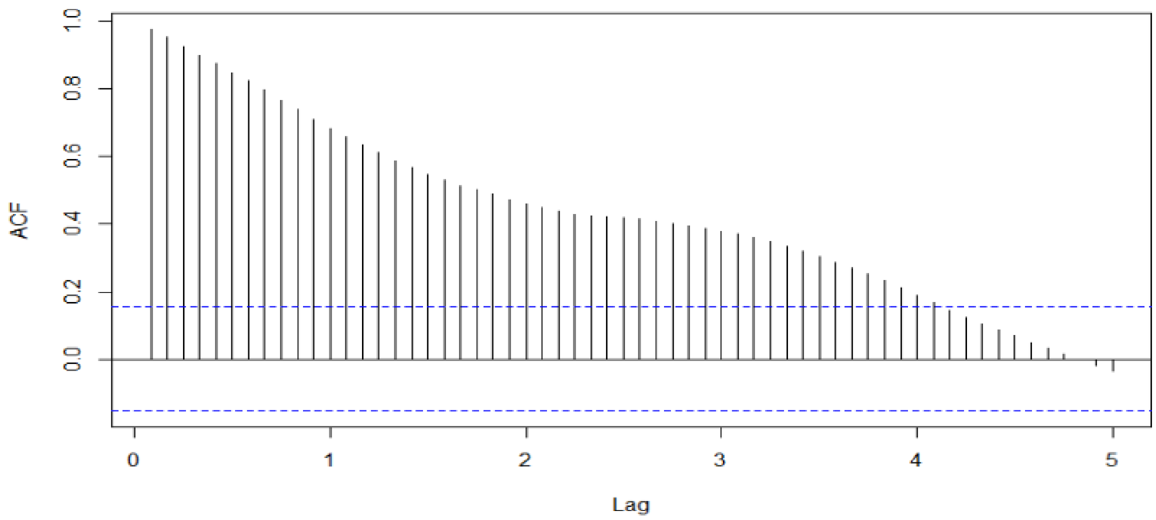
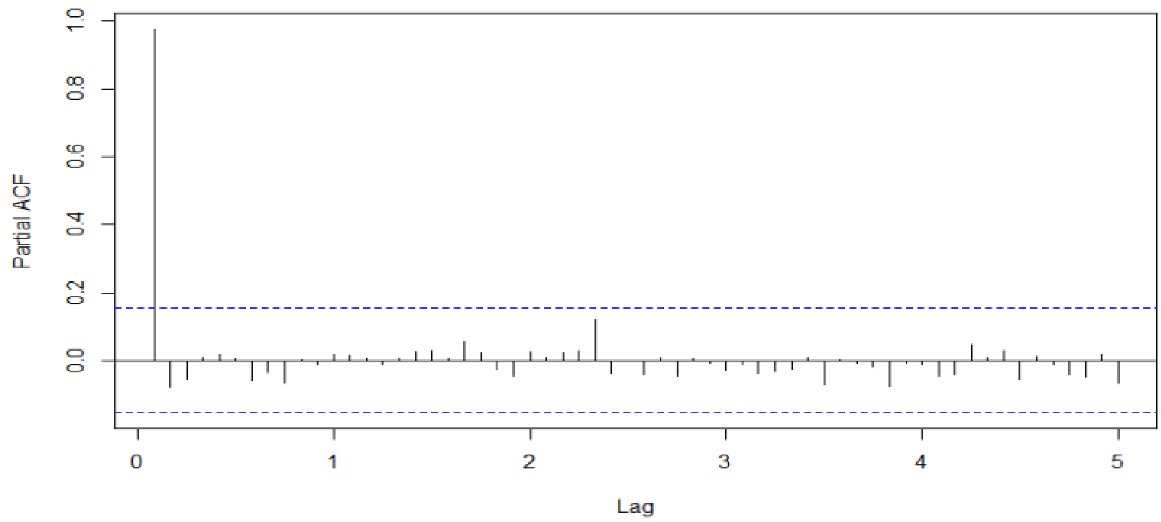


Figure 14: PACF plot of BC-transformed Data.



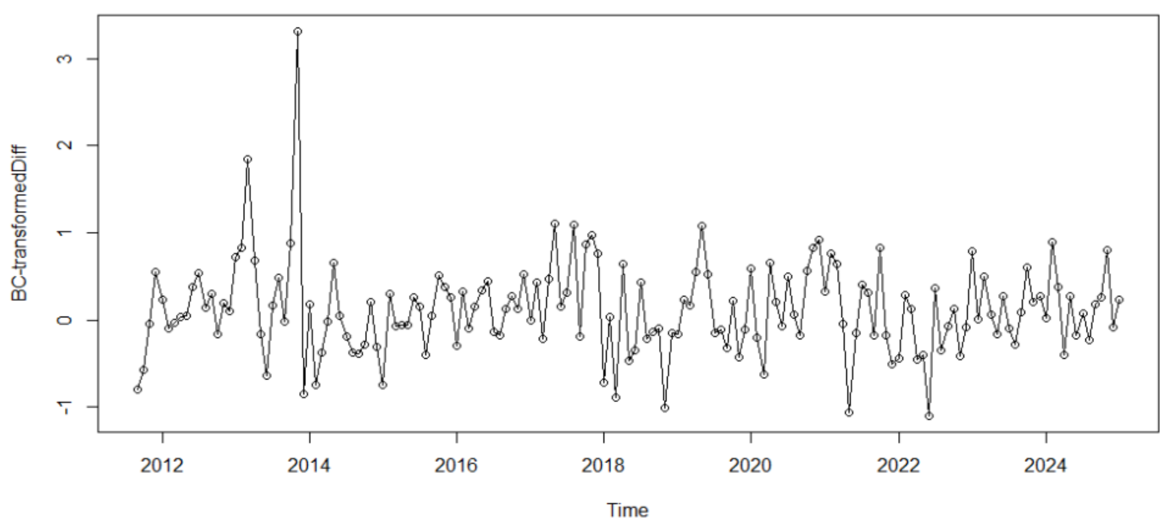

```
        Augmented Dickey-Fuller Test

data:  BC.data
Dickey-Fuller = -2.4479, Lag order = 5, p-value = 0.3895


        Phillips-Perron Unit Root Test

data:  BC.data
Dickey-Fuller Z(alpha) = -10.418, Truncation lag parameter = 4, p-value =
0.5175
        KPSS Test for Level Stationarity

data:  BC.data
KPSS Level = 3.0792, Truncation lag parameter = 4, p-value = 0.01
```

In ADF Test and PPS Test, p-value is greater than 0.05, we fail to reject null hypothesis in both the tests. In KPSS Test, p-value is lesser than 0.05, we reject null hypothesis.
So stationarity still exists.

From all the tests above, we can proceed with either Log Transformation or Box-Cox Transformation. I am proceeding with Box-Cox Transformation because there is no difference in plotting graph of both the transformed data. QQ-plot remains more or less similar as both of them are unable to capture the entire residuals especially in the tail part i.e both in upper and in lower ones. ACF and PACF of both the graph remains the same, the only difference I could see is in the 3 tests. In KPSS test, p-value remains the same at 0.01 for both the transformations. In ADF Test, p-value of Log Transformation is 0.4262, whereas in Box-Cox Transformation, p-value is 0.3895. This is very less compared to Log Transformation's p-value. In PP Test, p-value is 0.648 for Log Transformation and 0.5175 for Box-Cox Transformation. By comparing the p-values of both the tests, I proceeded with Box-Cox Transformation.

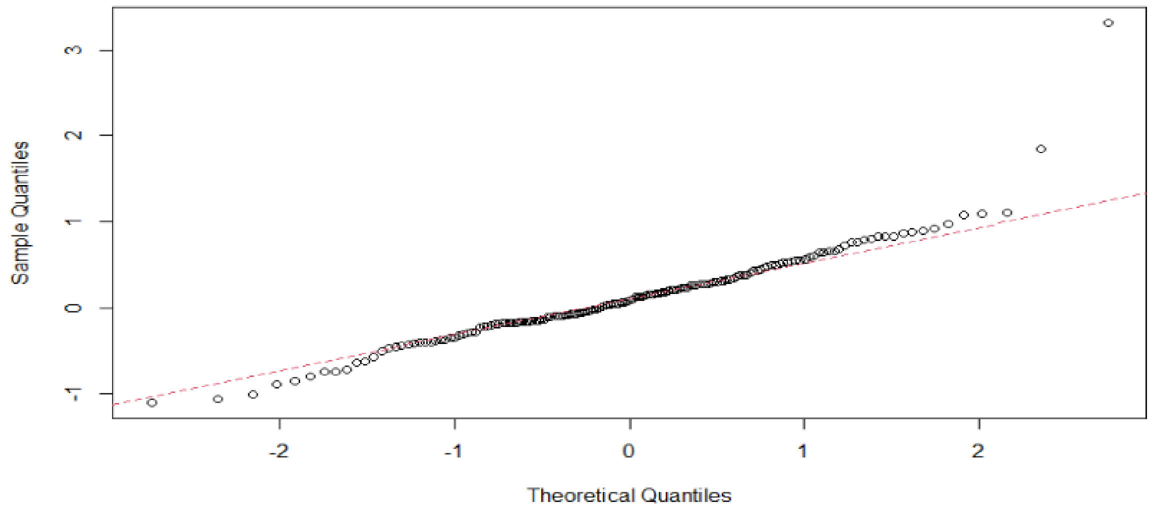
3. Model Specification:

Figure 15: Time series plot of BC-transformed Differencing Series.



We are proceeding with first order differencing of the Box-Cox transformed data. From Figure 15, we can see that there is no trend, also the series is now oscillating around a constant mean close to 0. We could see some visible spikes around late 2013 to early 2014 indicating a major Bitcoin event, after that the fluctuations remain stable, if we zoom out the Figure 15, we won't be noticing that too. So it is not a contender for change point. There is no visible seasonal pattern in the differenced series. Change in variance also got stabilized after differencing. In terms of Behaviour, we could see the presence of both Autoregressive and Moving Average. Before differencing, we could clearly say that there was no stationarity but after applying differencing, we can come to a conclusion that the series is stationary. We will further substantiate the stationarity of the series by analysing ACF, PACF plots and Unit Root tests or stationarity test which includes ADF, KPSS and PP test.

Figure 16: BC QQ plot of Bitcoin



From the QQ-plot in Figure 16, the differenced data has improved the accommodation of residuals compared to that of the transformed data in Figure 12. The differenced data graph is able to accommodate most of the residuals especially the ones present in the tail region too.

In Augmented Dickey-Fuller Test

```
data:  BC.data.diff
Dickey-Fuller = -4.896, Lag order = 5, p-value = 0.01
```

Null Hypothesis: Series is non-stationary
Alternative Hypothesis: Series is stationary
p-value is 0.01, which is lower than 0.05, we reject null hypothesis.

In Phillips-Perron Unit Root Test

```
data:  BC.data.diff
Dickey-Fuller Z(alpha) = -135.32, Truncation lag parameter = 4, p-value =
0.01
```

Null Hypothesis: Series is non-stationary
Alternative Hypothesis: Series is stationary
p-value is 0.01, which is lower than 0.05, we reject null hypothesis

In KPSS Test for Level Stationarity

```
data:  BC.data.diff
KPSS Level = 0.072676, Truncation lag parameter = 4, p-value = 0.1
```

Null Hypothesis: Series is stationary
Alternative Hypothesis: Series is non-stationary
p-value is 0.01, which is lesser than 0.05, we reject null hypothesis.

From all these tests, we could see that the series is now stationary. So, the differencing of the data has made the series to attain stationarity.

Figure 17: ACF of BC Differencing Transformation

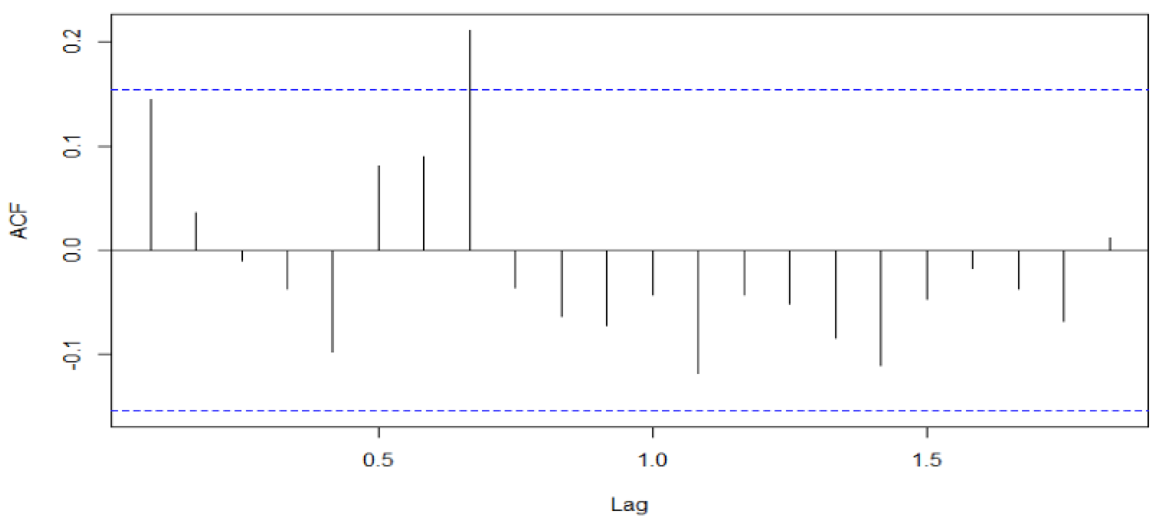
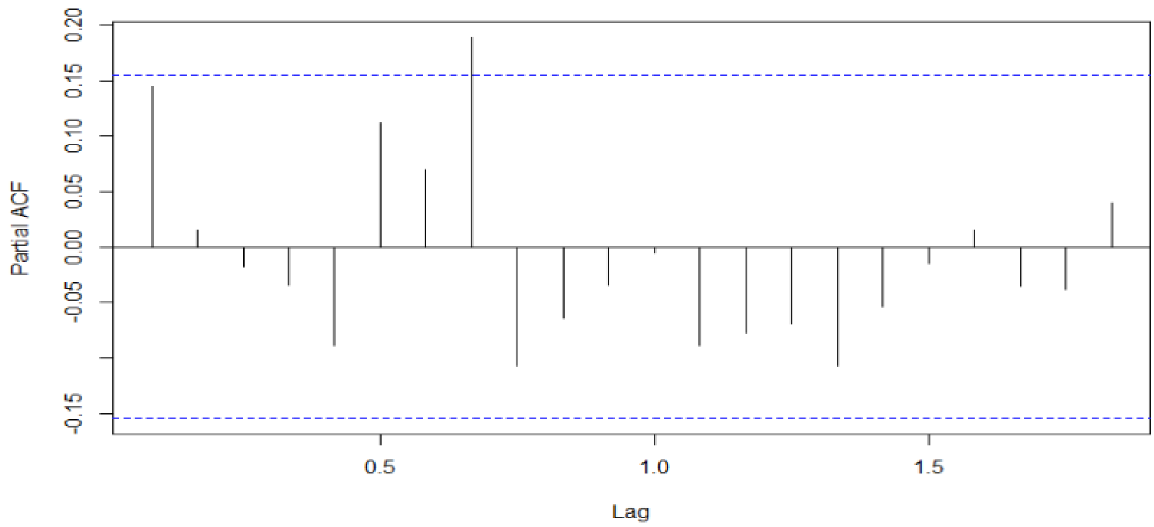


Figure 18: PACF of BC Differencing Transformation



From Figure 17, we can clearly see that the slow decay pattern no longer exists. In Figure 18 of PACF Plot, the initial lag is very close to the Confidence Interval line but it is not way past that. So a potential decrease in the 1st lag of PACF Plot is visible. From all these analysis, we can come to a potential conclusion that the stationarity of the series is achieved after differencing the Box-Cox transformed data.

Set of possible models obtained from ACF and PACF plots are

{ARIMA(1,1,1)} {ARIMA(1, 1, 2)} {ARIMA(2, 1, 1)} {ARIMA(2, 1, 2)}

In these models, the one with combination of 2 are candidate models, they are not the significant models. They are very close to the Confidence Interval, so we are considering them. These candidate models can be the potential model too. We will further proceed with EACF of the differenced data.

From EACF, the set of possible models we obtained are

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

{ARIMA(0,1,1)} {ARIMA(1,1,0)} {ARIMA(1,1,1)}

We already have {ARIMA(1,1,1)} from ACF and PACF plots, so in order to reduce duplicity, we will not consider this model from this graph.

In BIC graph from Figure 19, the set of possible models we obtained are

{ARIMA(1,1,0)} {ARIMA(1,1,4)} and {ARIMA(5,1,4)}

But we are not going to consider the model {ARIMA(5,1,4)} because this is not present in the top 2 models and also this model seems to be a complex fit with 5 AR and 4 MA which could lead to overfitting issue and according to the principle of parsimony we should always prefer models that are simple and adequately fit into our data.

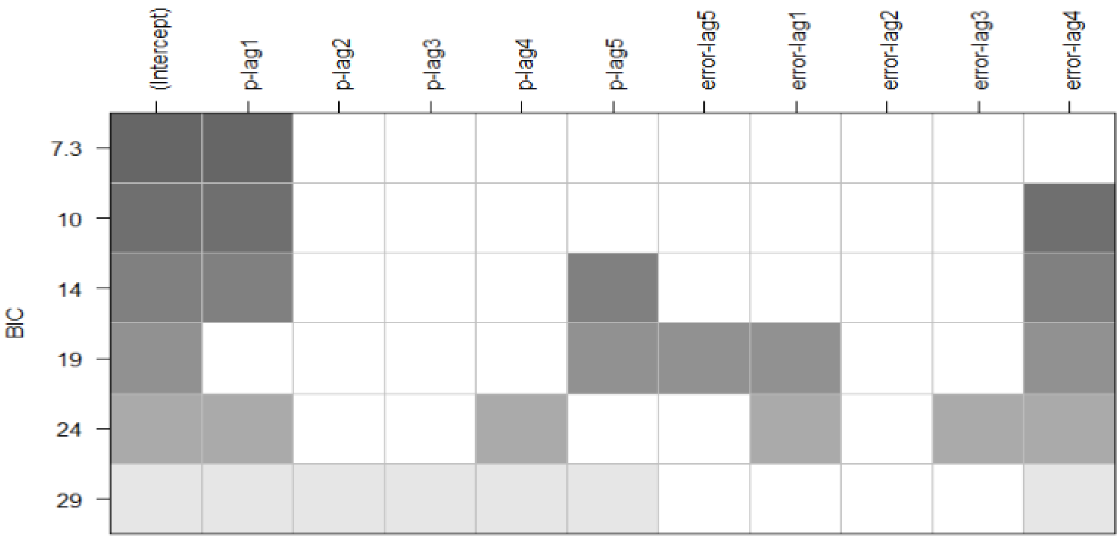


Figure 19: BIC of BC Differenced Data

Final set of ARIMA models including candidate models are,
{ARIMA(1,1,1)} {ARIMA(1, 1, 2)} {ARIMA(2, 1, 1)} {ARIMA(2, 1, 2)} {ARIMA(0,1,1)} {ARIMA(1,1,0)} {ARIMA(1,1,4)}

4. Model Fitting and Parameter Estimation

We are using Arima function to get parameter estimates and coeftest function to get the significance test of each parameter. If p-value is lesser than 0.05, they are significant or else they are nonsignificant. We will apply Arima() function on raw time series data btc_ts and see which model is the most significant among all the other models and also find out which model fits well to the data. We will use Least Squares(CSS), Maximum Likelihood (ML) and CSS-ML methods.

We will first consider the model ARIMA(1,1,1)

```
z test of coefficients(method='ML'):

      Estimate Std. Error z value Pr(>|z|)
ar1  0.1393832  0.4591484  0.3036   0.7615
ma1 -0.0062261  0.4607667 -0.0135   0.9892


z test of coefficients(method='CSS'):

      Estimate Std. Error z value Pr(>|z|)
ar1  0.1410742  0.4638558  0.3041   0.7610
ma1 -0.0094778  0.4659000 -0.0203   0.9838


z test of coefficients(method='CSS-ML'):

      Estimate Std. Error z value Pr(>|z|)
ar1  0.1411942  0.4615733  0.3059   0.7597
ma1 -0.0094247  0.4637748 -0.0203   0.9838
```

In all the methods, p-value was higher than 0.05. All 3 methods gave similar parameter estimates. None of the AR or MA terms were statistically Significant.

In ARIMA(1, 1, 2),

```
z test of coefficients(method='ML'):

     Estimate Std. Error z value  Pr(>|z|)
ar1 -0.858473   0.103248 -8.3147 < 2.2e-16 ***
ma1  1.048070   0.130906  8.0063 1.182e-15 ***
ma2  0.272386   0.098402  2.7681  0.005638 **

z test of coefficients(method='CSS'):

     Estimate Std. Error z value  Pr(>|z|)
ar1 -0.866984   0.102901 -8.4254 < 2.2e-16 ***
ma1  1.059757   0.130629  8.1127 4.951e-16 ***
ma2  0.280728   0.099493  2.8216  0.004779 **

z test of coefficients(method='CSS-ML'):

     Estimate Std. Error z value  Pr(>|z|)
ar1 -0.857484   0.103700 -8.2689 < 2.2e-16 ***
ma1  1.046861   0.131404  7.9667 1.629e-15 ***
ma2  0.271842   0.098434  2.7617  0.005751 **
```

In this model all the 3 methods are consistent. Ar1, ma1, ma2 are statistically significant with p-value lesser than 0.05. We can consider this as a potential contender for best model.

Model ARIMA(2, 1, 1)

```
z test of coefficients(method='ML'):

    Estimate Std. Error z value Pr(>|z|)
ar1 0.064386        NaN     NaN      NaN
ar2 0.015278        NaN     NaN      NaN
ma1 0.066888        NaN     NaN      NaN


z test of coefficients(method='CSS'):

     Estimate Std. Error z value  Pr(>|z|)
ar1 -0.866984   0.102901 -8.4254 < 2.2e-16 ***
ma1  1.059757   0.130629  8.1127 4.951e-16 ***
ma2  0.280728   0.099493  2.8216  0.004779 **


z test of coefficients(method='CSS-ML'):

    Estimate Std. Error z value Pr(>|z|)
ar1 0.063705        NaN     NaN      NaN
ar2 0.015436        NaN     NaN      NaN
ma1 0.066521        NaN     NaN      NaN
```

This model throws Nan for ML method so we move to CSS where we could see that there is consistency among all the coefficients, CSS-ML also shows NAN. Since we have good consistency among the coefficients in the CSS method, we can say that they are significant.

In Model ARIMA(2, 1, 2)

```
z test of coefficients(method='ML'):

     Estimate Std. Error  z value  Pr(>|z|)
ar1 -0.529398   0.091141  -5.8086 6.301e-09 ***
ar2 -0.835877   0.068231 -12.2507 < 2.2e-16 ***
ma1  0.662452   0.061559  10.7612 < 2.2e-16 ***
ma2  0.935219   0.053817  17.3777 < 2.2e-16 ***

z test of coefficients(method='CSS'):

     Estimate Std. Error  z value  Pr(>|z|)
ar1 -0.545799   0.098730  -5.5282 3.236e-08 ***
ar2 -0.848059   0.067947 -12.4812 < 2.2e-16 ***
ma1  0.676515   0.061934  10.9231 < 2.2e-16 ***
ma2  0.952984   0.053596  17.7810 < 2.2e-16 ***

z test of coefficients(method='CSS-ML'):

     Estimate Std. Error  z value  Pr(>|z|)
ar1 -0.529341   0.091131  -5.8086 6.301e-09 ***
ar2 -0.835679   0.068258 -12.2429 < 2.2e-16 ***
ma1  0.662463   0.061538  10.7650 < 2.2e-16 ***
ma2  0.935119   0.053868  17.3594 < 2.2e-16 ***
```

We achieved significance in ML method itself, I still continued to proceed with CSS and CSS-ML and they are significant in that too.

In Model ARIMA(0,1,1)

```
z test of coefficients(method='ML'):

    Estimate Std. Error z value Pr(>|z|)
ma1 0.126071   0.075546  1.6688  0.09516 .

z test of coefficients(method='CSS'):

    Estimate Std. Error z value Pr(>|z|)
ma1 0.126806   0.075739  1.6742  0.09408 .

z test of coefficients(method='CSS-ML'):

    Estimate Std. Error z value Pr(>|z|)
ma1 0.126071   0.075546  1.6688  0.09516 .
```

Eventhough we have only 1 coefficient but ma1 p-value is way higher than 0.05, so there is no significance in this model.

In model ARIMA(1,1,0)

```
z test of coefficients(method='ML'):

    Estimate Std. Error z value Pr(>|z|)
ar1 0.131888   0.078757  1.6746  0.09401 .

z test of coefficients(method='CSS'):

    Estimate Std. Error z value Pr(>|z|)
ar1 0.132726   0.079026  1.6795  0.09305 .

z test of coefficients(method='CSS-ML'):

    Estimate Std. Error z value Pr(>|z|)
ar1 0.131888   0.078757  1.6746  0.09401 .
```

p-value is way higher than 0.05 in all the 3 models, so there is no significance.

In Model ARIMA(1,1,4)

```
z test of coefficients(method='ML'):

     Estimate Std. Error z value  Pr(>|z|)
ar1 -0.714550   0.184622 -3.8703 0.0001087 ***
ma1  0.896413   0.184164  4.8675  1.13e-06 ***
ma2  0.147209   0.113101  1.3016 0.1930639
ma3 -0.052147   0.118222 -0.4411 0.6591424
ma4  0.127529   0.105913  1.2041 0.2285527

z test of coefficients(method='CSS'):

     Estimate Std. Error z value  Pr(>|z|)
ar1 -0.720819   0.190688 -3.7801 0.0001568 ***
ma1  0.903800   0.188593  4.7923 1.649e-06 ***
ma2  0.149802   0.114327  1.3103 0.1900960
ma3 -0.053474   0.120142 -0.4451 0.6562544
ma4  0.130248   0.108991  1.1950 0.2320716
```

```
z test of coefficients(method='CSS-ML'):

     Estimate Std. Error z value  Pr(>|z|)
ar1 -0.714459   0.184700 -3.8682 0.0001096 ***
ma1  0.896207   0.184194  4.8656 1.141e-06 ***
ma2  0.146979   0.113086  1.2997 0.1937013
ma3 -0.052182   0.118209 -0.4414 0.6588968
ma4  0.127625   0.105908  1.2051 0.2281824
```

There is no consistency in significance among the coefficients. Only ma1 is significant in all the 3 methods, so this model is also non-significant.
Models ARIMA(1, 1, 2), ARIMA(2, 1, 1), ARIMA(2, 1, 2) are the ones which showed Significance among all the other models.

5. Model Selection

We are going to check the Goodness of fit criteria. We will check AIC,BIC and MSE values of all the models. We are sorting both AIC and BIC based on the score. The model with least AIC and BIC values are considered as potential candidate models. The first block is AIC and the second block is BIC values of all the models. From both of them, we can see that model.212 is the one with least AIC value and model.110 is the one with least BIC value.

	Df	AIC
model.212	5	5477.276
model.114	6	5478.478
model.112	4	5479.077
model.110	2	5479.516
model.011	2	5479.620
model.111	3	5481.515
model.211	4	5483.509

	Df	BIC
model.110	2	5485.678
model.011	2	5485.783
model.111	3	5490.760
model.112	4	5491.403
model.212	5	5492.683
model.211	4	5495.834
model.114	6	5496.967

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
ARIMA(1,1,1)	693599.2	5853884	2951392	2.518	17.446	0.227	-0.014
ARIMA(1, 1, 2)	631986.9	5770071	2971860	2.438	18.057	0.229	-0.043
ARIMA(2, 1, 1)	690963.7	5853769	2951410	2.514	17.449	0.227	-0.011
ARIMA(2, 1, 2)	718956.8	5691449	2822234	2.497	18.725	0.217	0.003
ARIMA(0,1,1)	710175.5	5855822	2945637	2.550	17.415	0.227	-0.008
ARIMA(1,1,0)	695133.5	5853895	2950675	2.522	17.443	0.227	-0.012
ARIMA(1,1,4)	654373.5	5684633	2987766	2.448	19.358	0.230	-0.015

ACF1 is the first lag residual, we won't be measuring this. We will consider RMSE, MAE, MAPE, MASE as these are the most important ones to choose the best model. ME and MPE cancel out each other so we are not going to consider them too. We will first check the model with least RMSE and model ARIMA(1,1,4) has the least RMSE value among other models. In MAE, model ARIMA(2, 1, 2) is the one with least MAE value. For MAPE, model ARIMA(0,1,1) is the least one and for MASE model ARIMA(2, 1, 2) is the least one. The model ARIMA(2, 1, 2) identified by AIC was the one with least errors and if we check the Significance test results, they are significant too. This model is Significant in all the 3 methods ML, CSS and CSS-ML too. So model ARIMA(2, 1, 2) is the best model among all the other models. I then checked the neighbours of this model to see if I can get 1 or 2 more significant models. But their AR and MA parameters are not significant for any of the methods. The results are

This is for model ARIMA(3, 1, 2)

```
z test of coefficients for model (method='ML'):

     Estimate Std. Error  z value  Pr(>|z|)    
ar1 -1.381851   0.081640 -16.9262 < 2.2e-16 ***
ar2 -0.705607   0.124508  -5.6672 1.452e-08 ***
ar3  0.019240   0.081513   0.2360    0.8134    
ma1  1.616030   0.037926  42.6102 < 2.2e-16 ***
ma2  0.999993   0.043124  23.1889 < 2.2e-16 ***
---

z test of coefficients(method='CSS'):

      Estimate Std. Error  z value  Pr(>|z|)    
ar1 -1.6411345  0.0792131 -20.7180 < 2.2e-16 ***
ar2 -0.5973248  0.1451102  -4.1164 3.849e-05 ***
ar3  0.1336300  0.0820282   1.6291    0.1033    
ma1  1.8957234  0.0092146 205.7309 < 2.2e-16 ***
ma2  1.0275865  0.0110302  93.1613 < 2.2e-16 ***

z test of coefficients(method='CSS-ML'):

     Estimate Std. Error  z value  Pr(>|z|)    
ar1 -1.641968   0.080472 -20.4043 < 2.2e-16 ***
ar2 -0.578247   0.146721  -3.9411  8.11e-05 ***
ar3  0.148599   0.089284   1.6643   0.09605 .  
ma1  1.881087   0.036477  51.5698 < 2.2e-16 ***
ma2  0.999998   0.035828  27.9110 < 2.2e-16 ***
```

This is for model ARIMA(2, 1, 3)

```
z test of coefficients(method='ML'):

      Estimate Std. Error z value Pr(>|z|)   
ar1 -1.3294946  0.4798826 -2.7705 0.005598 **
ar2 -0.4861121  0.3414510 -1.4237 0.154543   
ma1  1.5062440  0.4804559  3.1350 0.001718 **
ma2  0.7347833  0.5013438  1.4656 0.142750   
ma3  0.0057355  0.2061035  0.0278 0.977799   

z test of coefficients(method='CSS'):

     Estimate Std. Error z value  Pr(>|z|)    
ar1 -1.293135   0.399110 -3.2400 0.0011951 ** 
ar2 -0.450365   0.310025 -1.4527 0.1463148    
ma1  1.470780   0.400241  3.6747 0.0002381 ***
ma2  0.689744   0.441206  1.5633 0.1179780    
ma3 -0.012099   0.177933 -0.0680 0.9457893    
```

```
z test of coefficients(method='CSS-ML'):

      Estimate Std. Error z value  Pr(>|z|)    
ar1 -1.3036291  0.3869695 -3.3688 0.0007549 ***
ar2 -0.4698591  0.2887062 -1.6275 0.1036385    
ma1  1.4806535  0.3881814  3.8143 0.0001366 ***
ma2  0.7098365  0.4179604  1.6983 0.0894447 .  
ma3 -0.0047668  0.1733358 -0.0275 0.9780607    
```

6. Conclusion:

I did a comprehensive descriptive analysis which indicated a strong upward trend and non-stationarity in the Bitcoin Index. Then we proceeded with transformation and differencing of the series. By doing so, we were able to stabilize the series and eventually we achieved stationarity. We then proceeded with model specification where we did the ACF, PACF, EACF and BIC. From these graphs, we identified significant and candidate models. Then we moved onto Model fitting and parameter estimation using Arima and Coeftest functions to see which model attains significance status. The last step is, we went on to see the goodness-of-fit of all the selected models, we checked multiple error measures and found out that the model ARIMA(2, 1, 2) is the best model among all the other models.

7. Appendix: R Codes

```
# Loading libraries
suppressPackageStartupMessages({
  suppressWarnings({
    library(tidyverse)
    library(lubridate)
    library(TSA)
    library(tseries)
    library(zoo)
    library(lmtest)
    library(forecast)
    library(fUnitRoots)
  })
})

# Loading the dataset
data <- read.csv("assignment2Data2025.csv")

# Checking Column Names in the dataset
colnames(data)
# Checking the class
class(data)
class(data$Date)
class(data$Bitcoin)
# Check for missing values
colSums(is.na(data))
#Checking Summary
```

```
summary(data)

# Creating time series object
btc_ts <- ts(data$Bitcoin, start = c(2011, 8), frequency = 12)

#Plotting the series
plot(btc_ts,
    type = "o",
    main = "Figure 1: Bitcoin Index (USD) from Aug 2011 to Jan 2025",
    xlab = "Year",
    ylab = "Bitcoin Index (USD)")

# Assigning Bitcoin column to y
y <- data$Bitcoin

x1 <- zlag(data$Bitcoin)
index1 <- 2:length(x1)

# Correlation at lag 1
cor(y[index1], x1[index1])

# Plotting lag 1
plot(y[index1], x1[index1],
    main = "Figure 2: Lag 1 Scatter Plot",
    xlab = "Lag 1",
    ylab = "Original Bitcoin Data")

# ----- Lag 2 -----
x2 <- zlag(zlag(data$Bitcoin))
index2 <- 3:length(x2)

# Correlation at lag 2
cor(y[index2], x2[index2])

# Plotting lag 2
plot(y[index2], x2[index2],
    main = "Figure 3: Lag 2 Scatter Plot",
    xlab = "Lag 2",
    ylab = "Original Bitcoin Data")

# ACF Plot
acf(btc_ts,main = "Figure 4: ACF of Bitcoin Index", lag.max = 60)

# PACF Plot
pacf(btc_ts,main = "Figure 5: PACF of Bitcoin Index")

adf.test(btc_ts)
pp.test(btc_ts)
```

```
kpss.test(btc_ts)

qqnorm(y=btc_ts, main = "Figure 6: QQ plot of Bitcoin")
qqline(y=btc_ts, col = 2, lwd = 1, lty = 2)
shapiro.test(btc_ts)

#Log Transformation
btc_tslog <- log(btc_ts)
plot(btc_tslog,
    type = "o",
    main = "Figure 7: Log Transformation Bitcoin Index",
    ylab = "Differenced Bitcoin",
    xlab = "Time")

qqnorm(y=btc_tslog, main = "Figure 8: QQ plot of Log Transformed Data")
qqline(y=btc_tslog, col = 2, lwd = 1, lty = 2)
shapiro.test(btc_tslog)

acf(btc_tslog, main ="Figure 9: ACF plot of Log Transformed Data.", lag.max = 60)
pacf(btc_tslog, main ="Figure 10: PACF plot of Log Transformed Data.", lag.max = 60)

adf.test(btc_tslog)
pp.test(btc_tslog)
kpss.test(btc_tslog)

#Box-Cox Transformation
BC = BoxCox.ar(btc_ts,lambda = seq(-0.4, 1, 0.01))
BC$ci
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
lambda
BC.data = (btc_ts^lambda-1)/lambda

plot(BC.data,type='o',ylab='BC-transformed', main = " Figure 11: Time series plot of BC-
transformed." )

qqnorm(y=BC.data, main = "Figure 12: BC QQ plot of Bitcoin")
qqline(y=BC.data, col = 2, lwd = 1, lty = 2)
shapiro.test(BC.data)

acf(BC.data, main ="Figure 13: ACF plot of BC-transformed Data.", lag.max = 60)
pacf(BC.data, main ="Figure 14: PACF plot of BC-transformed Data.", lag.max = 60)

adf.test(BC.data)
pp.test(BC.data)
kpss.test(BC.data)

#Box-Cox data differencing
BC.data.diff <- diff(BC.data, differences = 1)
```

```
plot(BC.data.diff, type='o',ylab='BC-transformedDiff', main = " Figure 15: Time series plot of BC-
transformed Differencing Series.")

qqnorm(y=BC.data.diff, main = "Figure 16: BC QQ plot of Bitcoin")
qqline(y=BC.data.diff, col = 2, lwd = 1, lty = 2)
shapiro.test(BC.data.diff)

adf.test(BC.data.diff)
pp.test(BC.data.diff)
kpss.test(BC.data.diff)

acf(BC.data.diff, main = "Figure 17: ACF of BC Differencing Transformation")
pacf(BC.data.diff, main = "Figure 18: PACF of BC Differencing Transformation")

eacf(BC.data.diff)

BC.bic = armasubsets(y= BC.data.diff, nar=5, nma=5, y.name='p', ar.method='ols')
plot(BC.bic)
mtext("Figure 19: BIC of BC Differenced Data", side = 1, line = 1, cex = 1)

# ARIMA(1,1,1)
model.111 = Arima(btc_ts,order=c(1,1,1), method='ML')
coeftest(model.111)

model.111CSS = Arima(btc_ts,order=c(1,1,1), method='CSS')
coeftest(model.111CSS)

model.111CSS_ML = Arima(btc_ts,order=c(1,1,1), method='CSS-ML')
coeftest(model.111CSS_ML)

#ARIMA(1, 1, 2)
model.112 = Arima(btc_ts,order=c(1,1,2), method='ML')
coeftest(model.112)

model.112CSS = Arima(btc_ts,order=c(1,1,2), method='CSS')
coeftest(model.112CSS)

model.112CSS_ML = Arima(btc_ts,order=c(1,1,2), method='CSS-ML')
coeftest(model.112CSS_ML)

#ARIMA(2, 1, 1)
model.211 = Arima(btc_ts,order=c(2,1,1), method='ML')
coeftest(model.211)

model.211CSS = Arima(btc_ts,order=c(2,1,1), method='CSS')
coeftest(model.112CSS)

model.211CSS_ML = Arima(btc_ts,order=c(2,1,1), method='CSS-ML')
```

```
coeftest(model.211CSS_ML)

#ARIMA(2, 1, 2)
model.212 = Arima(btc_ts,order=c(2,1,2), method='ML')
coeftest(model.212)

model.212CSS = Arima(btc_ts,order=c(2,1,2), method='CSS')
coeftest(model.212CSS)

model.212CSS_ML = Arima(btc_ts,order=c(2,1,2), method='CSS-ML')
coeftest(model.212CSS_ML)

#{ARIMA(0,1,1)}
model.011 = Arima(btc_ts,order=c(0,1,1), method='ML')
coeftest(model.011)

model.011CSS = Arima(btc_ts,order=c(0,1,1), method='CSS')
coeftest(model.011CSS)

model.011CSS_ML = Arima(btc_ts,order=c(0,1,1), method='CSS-ML')
coeftest(model.011CSS_ML)

# {ARIMA(1,1,0)}
model.110 = Arima(btc_ts,order=c(1,1,0), method='ML')
coeftest(model.110)

model.110CSS = Arima(btc_ts,order=c(1,1,0), method='CSS')
coeftest(model.110CSS)

model.110CSS_ML = Arima(btc_ts,order=c(1,1,0), method='CSS-ML')
coeftest(model.110CSS_ML)

#{ARIMA(1,1,4)}
model.114 = Arima(btc_ts,order=c(1,1,4), method='ML')
coeftest(model.114)

model.114CSS = Arima(btc_ts,order=c(1,1,4), method='CSS')
coeftest(model.114CSS)

model.114CSS_ML = Arima(btc_ts,order=c(1,1,4), method='CSS-ML')
coeftest(model.114CSS_ML)

sort.score <- function(x, score = c("bic", "aic")){
  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(model.111,model.112,model.211,model.212,model.011,model.110,model.114), score
= "aic")

sort.score(BIC(model.111,model.112,model.211,model.212,model.011,model.110,model.114), score
= "bic")

model_111_acc <- accuracy(model.111)[1:7]
model_112_acc <- accuracy(model.112)[1:7]
model_211_acc <- accuracy(model.211)[1:7]
model_212_acc <- accuracy(model.212)[1:7]
model_011_acc <- accuracy(model.011)[1:7]
model_110_acc <- accuracy(model.110)[1:7]
model_114_acc <- accuracy(model.114)[1:7]
df.models <- data.frame(

rbind(model_111_acc,model_112_acc,model_211_acc,model_212_acc,model_011_acc,model_110_
acc,
      model_114_acc)
)
colnames(df.models) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                         "MASE", "ACF1")
rownames(df.models) <- c("ARIMA(1,1,1)", "ARIMA(1, 1, 2)", "ARIMA(2, 1, 1)", "ARIMA(2, 1, 2)",
"ARIMA(0,1,1)", "ARIMA(1,1,0)", "ARIMA(1,1,4)")
round(df.models,  digits = 3)


model.213ml = Arima(btc_ts,order=c(2,1,3),method='ML')
coeftest(model.213ml)

model.213css = Arima(btc_ts,order=c(2,1,3),method='CSS')
coeftest(model.213css)

model.213mlcss = Arima(btc_ts,order=c(2,1,3),method='CSS-ML')
coeftest(model.213mlcss)

model.312ml = Arima(btc_ts,order=c(3,1,2),method='ML')
coeftest(model.312ml)

model.312css = Arima(btc_ts,order=c(3,1,2),method='CSS')
coeftest(model.312css)

model.312mlcss = Arima(btc_ts,order=c(3,1,2),method='CSS-ML')
coeftest(model.312mlcss)
```

8. Reference:

1. https://rmit.instructure.com/courses/140832/pages/week-2-after-class?module_item_id=7091840
2. https://rmit.instructure.com/courses/140832/files/44500708?module_item_id=7223237
3. https://rmit.instructure.com/courses/140832/pages/week-3-after-class?module_item_id=7092159
4. https://rmit.instructure.com/courses/140832/files/44536989?module_item_id=7225020
5. https://rmit.instructure.com/courses/140832/pages/week-4-after-class?module_item_id=7092165
6. https://rmit.instructure.com/courses/140832/files/43692306?module_item_id=7077746
7. https://rmit.instructure.com/courses/140832/pages/week-5-after-class?module_item_id=7092178
8. https://rmit.instructure.com/courses/140832/files/44854127?module_item_id=7250312
9. https://rmit.instructure.com/courses/140832/pages/week-6-after-class?module_item_id=7092201
10. https://rmit.instructure.com/courses/140832/files/45246826?module_item_id=7298823
11. Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. Journal of the Royal Statistical Society Series B (Statistical Methodology), 26(2), 211–243. https://doi.org/10.1111/j.2517-6161.1964.tb00553.x
12. Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723. https://doi.org/10.1109/TAC.1974.1100705

I did not use any AI tools for this assignment