Time Series Report

Time Series Analysis and Spatial Statistics - Umeå University

Artem Shiryaev

2024-03-15

Introduction

In this report we will familiarize ourselves with time series analysis methods, applied to forecasting of log Bitcoin prices as our case study of choice. The format of the report is as follows, firstly we introduce the methods and provide a brief overview − the curious reader may refer to the textbook *Introduction to Time Series and Forecasting Third Edition* by Peter J. Brockwell and Richard A. Davis, for more detailed explanations. Secondly, we will present the results of applying the methods covered on our case study of choice and lastly a brief discussion and conclusion will be drawn by judging the results of the analysis in a concise and clear manner.

The following material can be accessed from my GitHub repo, forked and ran by yourself using R. Link: https://github.com/ArtemShiryaev/TimeSeriesVT24UMU/

Methods

Working with time series, we usually start with investigating properties of the dataset. Does it include a seasonal compontent? Perhaps there is a deterministic trend or drift? Ideally we would like to work with stationary time series right of the bat, however, most likely than not we must apply methods and/or transform the data in order to stationarize it.

Most commonly as a first step, one may transform the dataset using a one-to-one function, generally being the logarithmic tranformation, using the addative properties

$$data \Rightarrow ln(data)$$

If applying a tranformation is not sufficient, one may proceede with either the S1 method or S2 method to remove any drift, deterministic trend and seasonal components, to finally get stationary residuals.

The S1 method

Let there be a times series $\{X_t, t \in \mathbb{Z}\}$ with a seasonality compontent of d, and assume $d = 2q$, then the formula we utilize to compute for each observation.

$$m_t = \frac{0.5x_{t-q/2} + \cdots + x_{t-2} + x_{t-1} + x_t + x_{t+1} + x_{t+2} + \cdots + 0.5x_{t+q/2}}{q}$$

The second step in the S1 process is to subtract the filtered mean from the dataset. Thirdly, by subtracting each seasonal component.

$$s_t = w_k - \frac{1}{d}\Sigma_i^d w_i \quad , i, k = 1, 2, \dots d$$

Then we obtain de-seasonalized data by

$$d_t = x_t - s_t$$

Lastly we re-estimate the means using the de-seasonalized data

$$\hat{m}_t = \frac{0.5d_{t-3} + d_{t-2} + d_{t-1} + d_t + d_{t+1} + d_{t+2} + 0.5d_{t+3}}{6}$$

Followed by obtaining the residuals using the

$$\hat{Y}_t = x_t - \hat{m}_t - s_t$$

Followed by an inspection of the residuals. Typically reviewing the autocovariance fucntion plot to try spotting dependence structures remaining after de-trending and de-seasonalizing the data using S1.

The S2 method

Method S2 consist of elimination of trend and seasonal component by differencing.

The **lag-d** difference operator ∇_d is defined as

$$\nabla_d X_t = X_t - X - t - d = (1 - \mathcal{B}^d)X_t$$

where $\mathcal{B}$ is the backward shit operator defined as

$$\mathcal{B}X_t = X_{t-1}$$

Applying the classical decomposition model$X_t = m_t + s_t + Y_t$ where m_t is a slowly changing function known as a trend component, s_t is a function with known period d referred to as a seasonal component, and Y_t is a random noise component that is stationary, if Y_t is iid Gaussian White Noise then $\mathbf{E}[Y_t] = 0$. Applying the difference operator we get a de-seasonalized series with trend compontent $m_t - m_{t-d}$ and residual $Y_t - Y_d$.

Put mathematically:

$$\nabla_d X_t = m_t - m_{t-d} + Y_t - Y_{t-d}$$

Spectral Analysis

Assume $\{X_t, t \in \mathbb{Z}\}$ has a underlying seasonality or 'periodic' component, although X_t is represented in a noisy matter where the periodic element is challenging to spot. Then applying spectral analysis is a way to filter out the noise whilst keeping the periodic component intact. This method is typically done in the following steps

- Calculating the perriodogram using the spectral density
- Extracting the frequencies at which the spectral density is estimated
- Scaling and then plotting the spectral densities to review if a frequency is dominating the plot

Naturally, if a frequency can be spotted to exibit periodic behavior. We may utilize that to remove any underlying seasonal component.

Methods for testing iid noise and checking normality

Reviewing whether or not the differenced time series are iid Noise, after having applied either the S1 or S2 method we proceed with these steps.

- Visually checking the sample autocorrelation function
- Protmanteau test
- Turning point test
- Difference-sign test
- Mann-Kendall Rank test
- Augmented Dicker Fuller test
- Checking for normality
  - Histogram
  - qq plot
  - Normality test
    - Shapiro-Wilks test
    - Shapiro–Francia test
    - Jarque–Beratest
    - Anderson–Darling test

ACF and PACF Firstly, it is always wise to plot an ACF and PACF plot to see visually the behavior of the residuals. Since the ACF measures and plots the average correlation between data points in time series and previous values of the series measured for different lag lengths. PACF plots partial correlation controls for any correlation between observations of a shorter lag length. Both are used to determine in a general sense what order we should use for our ARMA(p,q) model. ACF cutoff on the plot provides us with the AR(p) and PACF with the MA(q)

Statistical Tests We will below lay the mathematical foundation all of these methods to see whether or not the time series residuals has a dependence structure or not. The typical hypothesis tests reviews if

H_0 = The Time Series is iid Noise, e.g. no dependence structure among residuals

H_1 = The Time Series is NOT iid Noise, e.g. there is a possible dependece structure among residuals

Protmanteau test The Portmanteau test continues and builds upon the idea of autorrelation. Let $\hat{\rho}(j)$ denote the sample autocorrelation value of lag j. Then if $Y_1, Y_2, \ldots, Y_n$ is an iid sequence, for large n

$$Q = n\Sigma_{j=1}^{h}\hat{\rho}(j) \quad , j = 1, \ldots, h$$

Q is approximately distributed as the sum of squares of the independent $\mathcal{N}(0,1)$ random variables, yielding that $\sqrt{n}\hat{\rho}(j)$ for $j = 1, \ldots, h$ is $\chi^2(h)$ distributed with h degrees of freedom.\ Ljung and Box refined this test with an better approximation of the χ^2 distribution using

$$Q_{LB} = n(n+2)\Sigma_{j=1}^{h}\frac{\hat{\rho}(j)}{n-j} \quad , j = 1, \ldots, h$$

Turning point test The method by which the turning point test the residual for an iid sequence is as follows. If $Y_1, \dots Y_n$ is a sequence of observations, then there is a turning point at time i if $Y_{i-1} < Y_i$ and $Y_i > Y_{i+1}$, or alternatively $Y_{i-1} > Y_i$ and $Y_i < Y_{i+1}$. Then

If T is the number of turning points of an iid sequence of length n, the for large n

$$T \in \mathcal{N}(\frac{2(n-2)}{3}, \frac{16n-29}{90})$$

Thus, we reject H_0 whenever $\frac{|T-\mu_T|}{\sigma_T} > \Phi_{1-\frac{\alpha}{2}}$ where $\Phi_{1-\frac{\alpha}{2}}$ is the 1- $\frac{\alpha}{2}$ quantile of the standard normal distribution $\mathcal{N}(0,1)$.

Difference-sign test Let $Y_1, \dots Y_n$ be a sequence of observations, then we count the number S of values i such that $Y_i > Y_{i-1}$.

If $Y_1, \dots Y_n$ is an iid sequence, then for large n

$$S \in \mathcal{N}(\frac{n-1}{2}, \frac{n+1}{12})$$

Thus, we reject H_0 whenever $\frac{|S-\frac{n-1}{2}|}{|\sqrt{\frac{n+1}{12}}|} > \Phi_{1-\frac{\alpha}{2}}$ where $\Phi_{1-\frac{\alpha}{2}}$ is the 1- $\frac{\alpha}{2}$ quantile of the standard normal distribution $\mathcal{N}(0,1)$.

Time Series literature argues however that the the difference-sign test must be used with caution. A set of observations exhibiting a cyclic component will pass the difference-sign test for randomness, since roughly half of the observations will be points of increase.

Mann-Kendall Rank test The rank test is especially useful for detecting a linear trend in the data. Define $\mathcal{P}$ to be the number of pairs (i,j) such that $Y_j > Y_i$ and $j > i$, $i = 1, \dots, n-1$ If $Y_1, \dots Y_n$ is an iid sequence, then for large n

$$\mathcal{P} \in \mathcal{N}(\frac{n(n-1)}{4}, \frac{n(n-1)(2n+5)}{72})$$

We would reject H_0 if $\frac{|\mathcal{P}-\frac{n(n-1)}{4}|}{|\sqrt{\frac{n(n-1)(2n+5)}{72}}|} > \Phi_{1-\frac{\alpha}{2}}$ where $\Phi_{1-\frac{\alpha}{2}}$ is the 1- $\frac{\alpha}{2}$ quantile of the standard normal distribution $\mathcal{N}(0,1)$.

Augmented Dicker Fuller test The testing procedure for the ADF test is applied to the model

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t,$$

where α is a constant, β the coefficient on a time trend and p the lag order of the autoregressive process. Imposing the constraints $\alpha = 0$ and $\beta = 0$ corresponds to modelling a random walk and using the constraint $\beta = 0$ corresponds to modeling a random walk with a drift.

The null hypothesis that a unit root is present in a time series, however on the contrary the alternative hypothesis is stationarity within the time series.

Forecasting Models

The ARMA model we consider is can be estimated in various manners, we shall consider Yule-Walker techniques for preliminary estimation of the autoregressive parameters $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)', \boldsymbol{\theta} = (\theta_1, \ldots, \phi_p)'$, and σ^2 from observations $x_1, \ldots, x_n$ of the causal ARMA(p, q) process defined by

$$\phi(B)X_t = \theta(B)Z_t, \quad \{Z_t\} \sim \text{WN}\left(0, \sigma^2\right).$$

For the details of the estimation procedure, we refer the reader to the aforementioned textbook literature. Naturally, we wish to evaluate the goodness of fit, initially by reviewing the residuals using the previously mentioned methods as well as using the Akaike Information Criteria defined as

$$\text{AIC} = -2\log(L) + 2(p + q + k)$$

where L is the likelihood of the data, p is the order of the autoregressive part and q is the order of the moving average part. The k represents the intercept of the ARIMA model.

Finally, after having picked the most suitable models based on the various evaluation methods, we shall try to forecast the subsetted dataset that was not included in the training phase of the model. Each models forecasted predictions will be evaluated using mean squared error MSE, defined as

$$\text{MSE} = (Predicted - Actual\,Values)^2$$

The lowest MSE, the better predictive ability.

Results of Case Study: Forecasting Bitcoin

Dataset

Our task was to use time series is an underlying dependence structure, in order to review and try to utilize it in order to forecast accurately our predictions. We know from financial literature that pricing of assets can be strongly dependent on financial performance - likewise are cryptocurrencies heavily dependent on the overall markets performance and typically experience cycles. Thus, we continue by finding downloading daily prices of Bitcoin as our data set of choice from Yahoo Finance.

The reader may see the aforementioned time series dataset in Figure 1.

To reduce the complexity of the time series and 'smooth' it, we average the prices of each month in accordance with the formula:

$$[\text{Monthly Prices}]_i = \Sigma_{i=1}^{30}[\text{Daily Prices}]_i \cdot \frac{1}{30} \quad , i = 1, 2, \ldots$$

Reducing the observations from 3401 to 101.

Plotting the transformed monthly data we can see a smoother time series, with less variation and spikes. Whilst keeping the overall trends as shown in Figure 2.

We can examine from Figure 3 an overview of the split dataset into train and test, here we indeed seem to have an upwards deterministic trend, seems as if a positive drift can be examined in the second plot. A seasonal component could perhaps be thought of, in addition US regulatory interventions has affected the pricing positively in 2024 - which can be seen in sharp increase in the opening prices. The 2018 'first' exposure to public and the pandemics type has also had a drastic influence on the pricing as can be observed in the change of pricing behavior.

Therefore, we perform an natural logarithmic transformation to reduce the yet still - drastic fluctuations. Judging by the Figure 4 below, we see a substantial improvement of the training dataset - reseabling more of a linear times series with drift and various seasonalities.

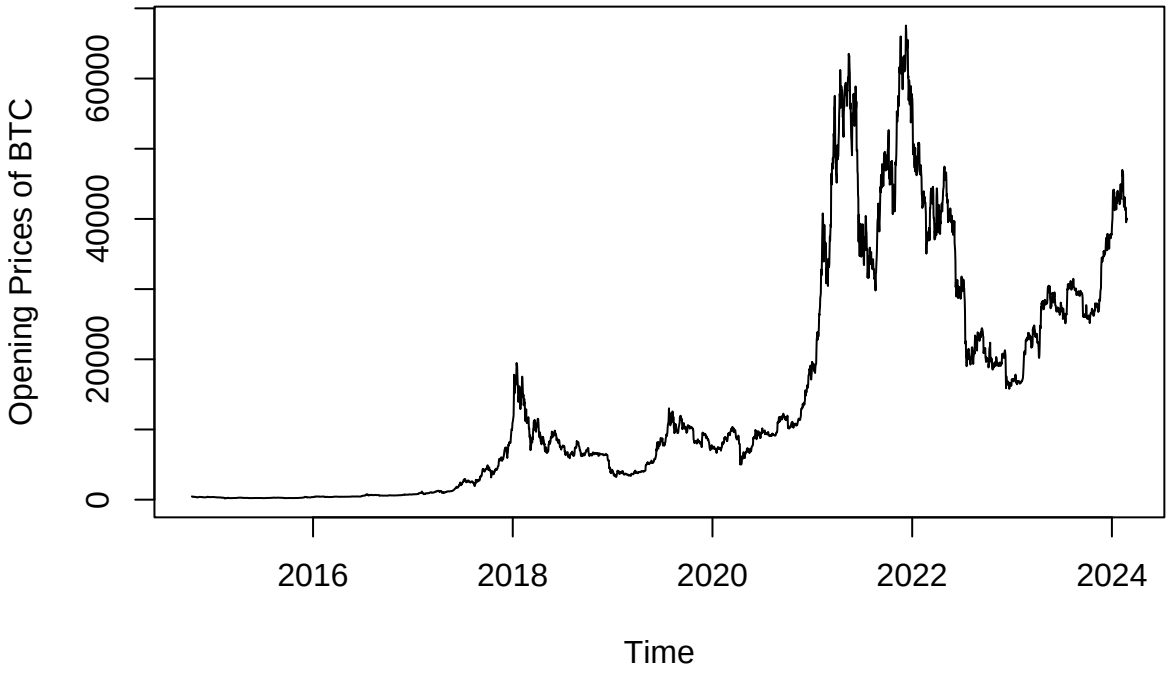

Figure 1: Daily Opening Prices of Bitcoin

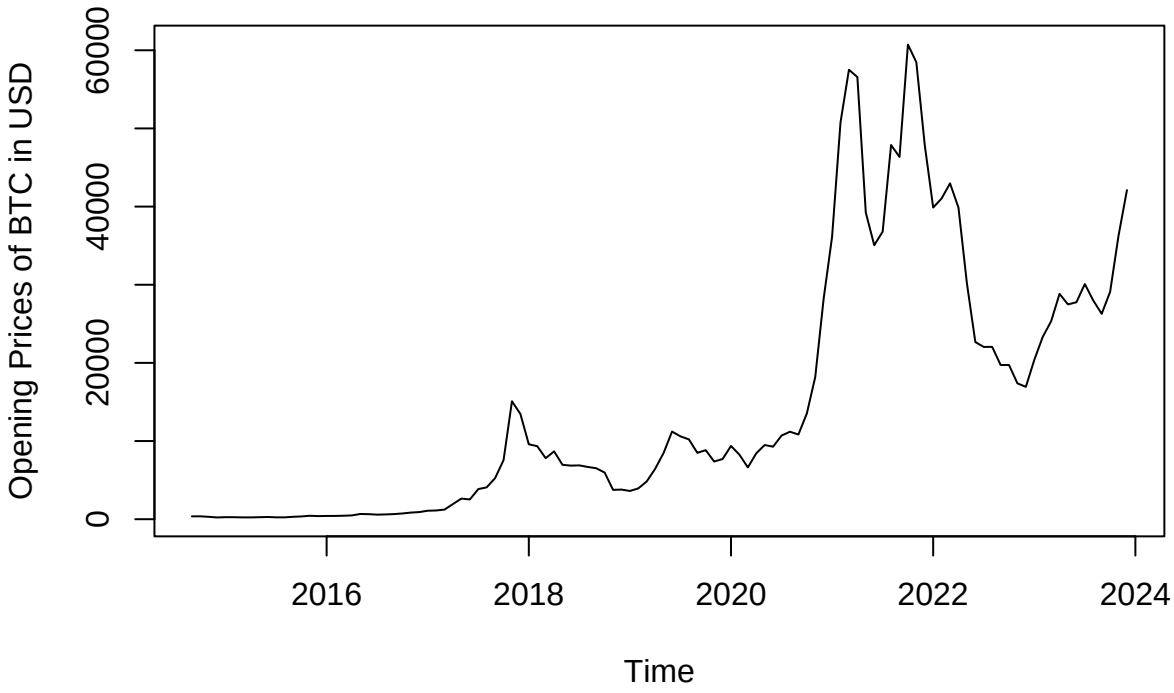

Figure 2: Monthly Opening Prices of Bitcoin

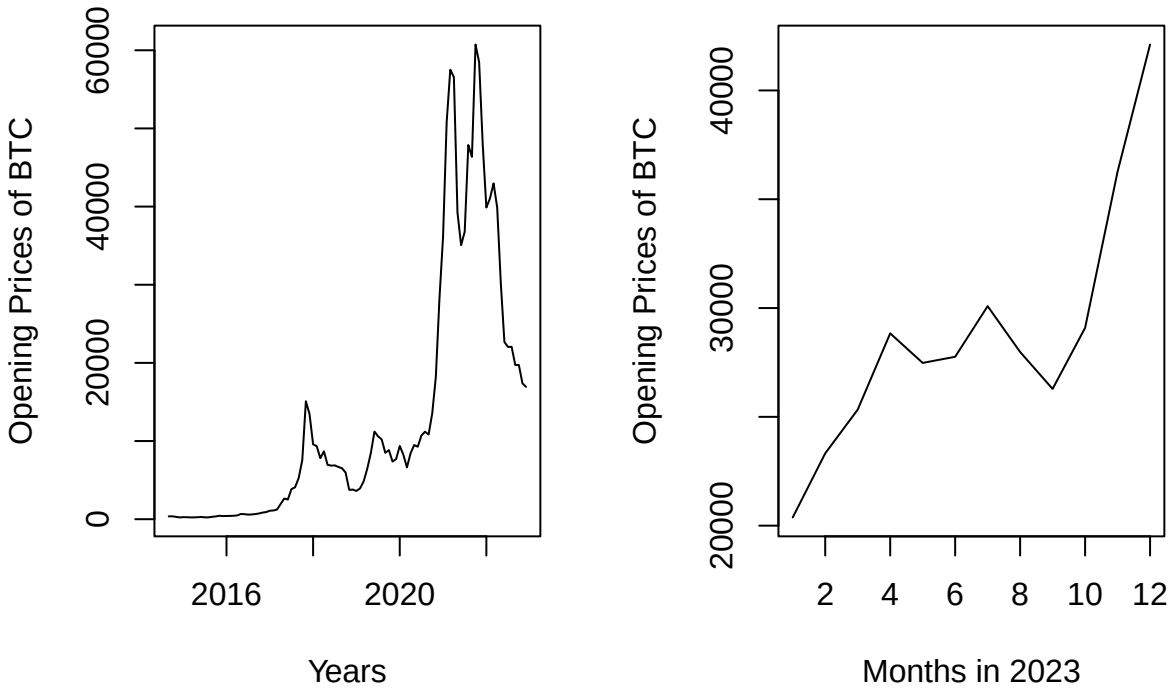

Figure 3: Daily Bitcoin Prices Split Data

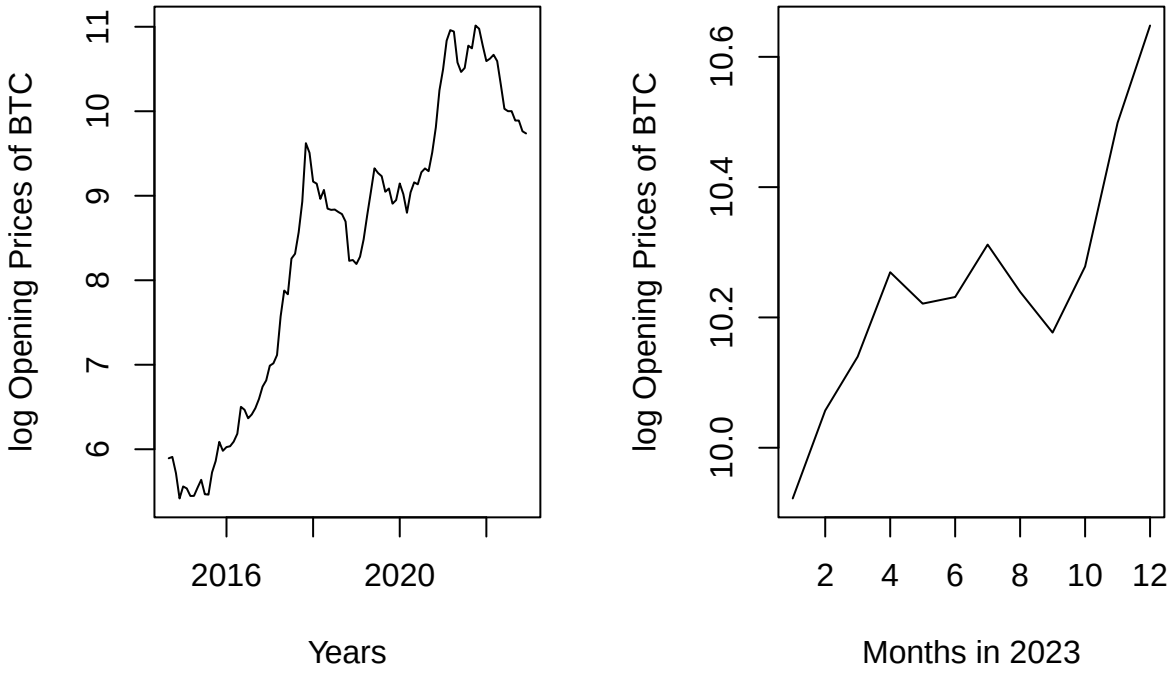

Figure 4: Daily Log Prices of Bitcoin

Stationarity

The logarthimic dataset is then proceed to remove any drift, deterministic trend and seasonal components in order to get stationary residuals by both using methods **S1** and **S2**.

Solution using S1 method

We have 101 observations and 12 monthly seasons yielding $d = 12 = 2q \Rightarrow q = 6$, thus according to the formula we are to compute for each observation.

$$m_t = \frac{0.5x_{t-3} + x_{t-2} + x_{t-1} + x_t + x_{t+1} + x_{t+2} + 0.5x_{t+3}}{6}$$

Following the logic described within the methods sections we initially get the filtered time series, seen in Figure 6.

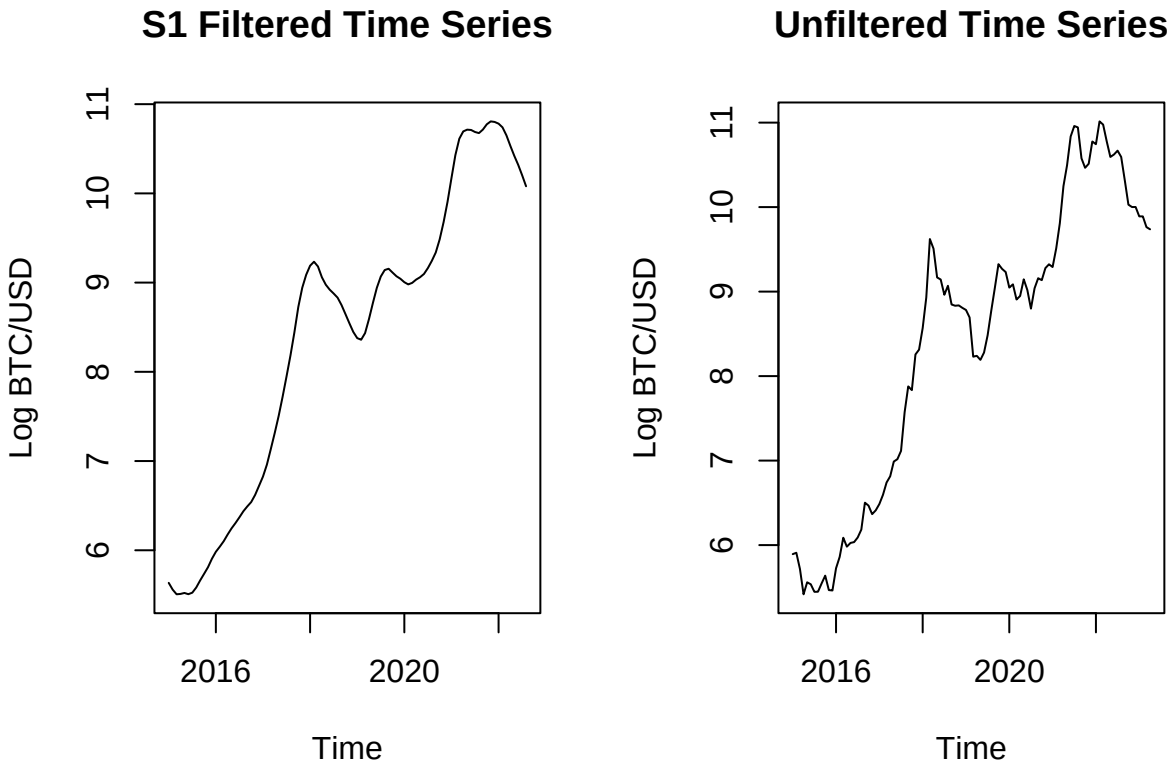

Figure 5: Filtered vs. Unfiltered Test Time Series

Calculating the seasonal component we get the following results as an output, which is subsequently used to remove and de-seasonalize the time series.

```
##        Jan        Feb      March      April        May       June       July
## -0.3421268 -0.3371797 -0.4081076 -0.1866173 -0.1347215 -0.1733557 -0.2558456
##        Aug       Sept        Oct        Nov        Dec
## -0.3120946 -0.3925010 -0.4275887 -0.4453251 -0.5095459
```

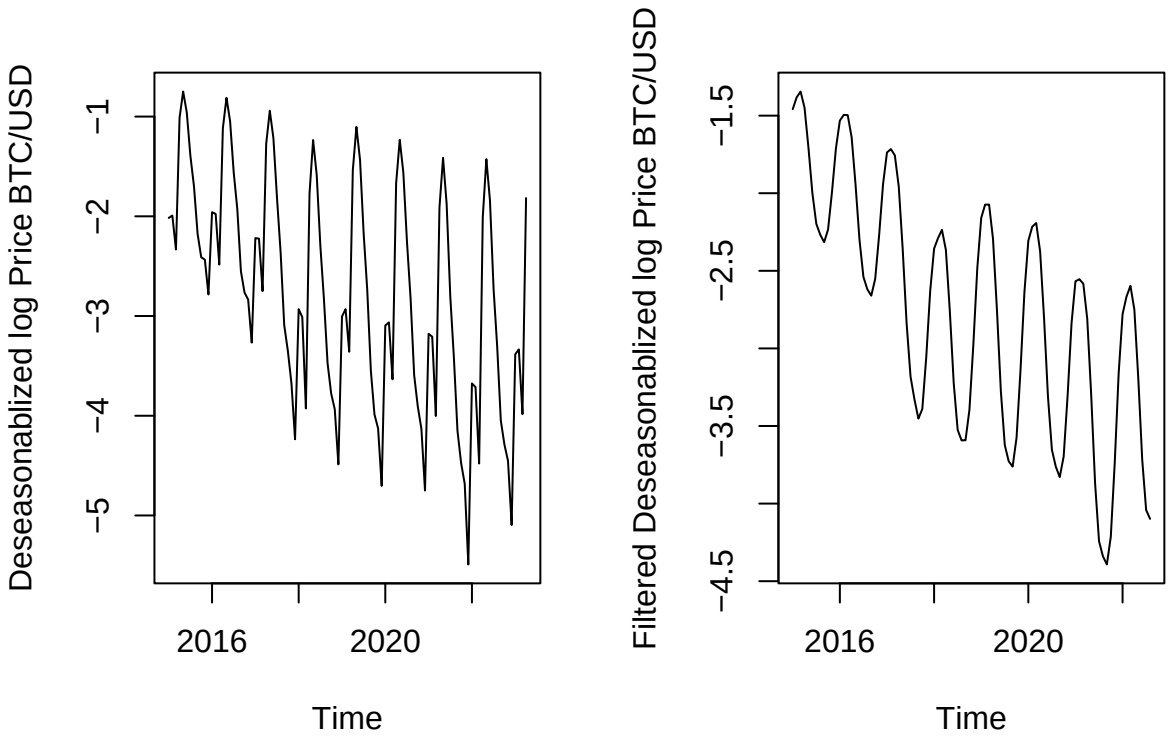

Figure 6: Deseasonalized Test Dataset

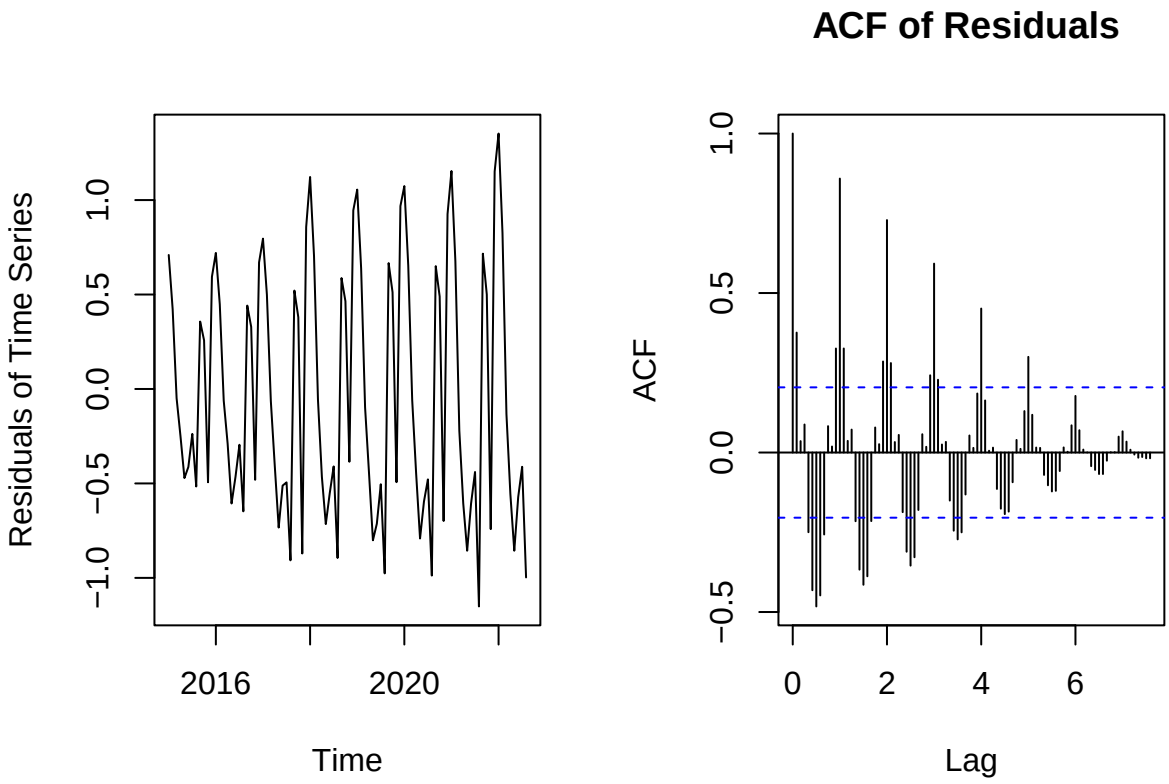

Figure 7: Residuals of Test Dataset

Following the aforementioned steps in the methods section we get following plot showing the deseasonalized log prices of Bitcoin.

Judging from Figure 7, the residuals are spread around 0, however there is clearly a pattern remaining which is also seen from the autocovariance function plot on the right hand side. Where up until lag 6 (e.g. six months) there is a statistical significant impact on the prices.

Solution using S2 Method

Method S2 consist of elimination of trend and seasonal component by differencing, applying the steps described in the methods section we can examine the autocovariance plot and the deseasonalized and detrended time series in Figure 8

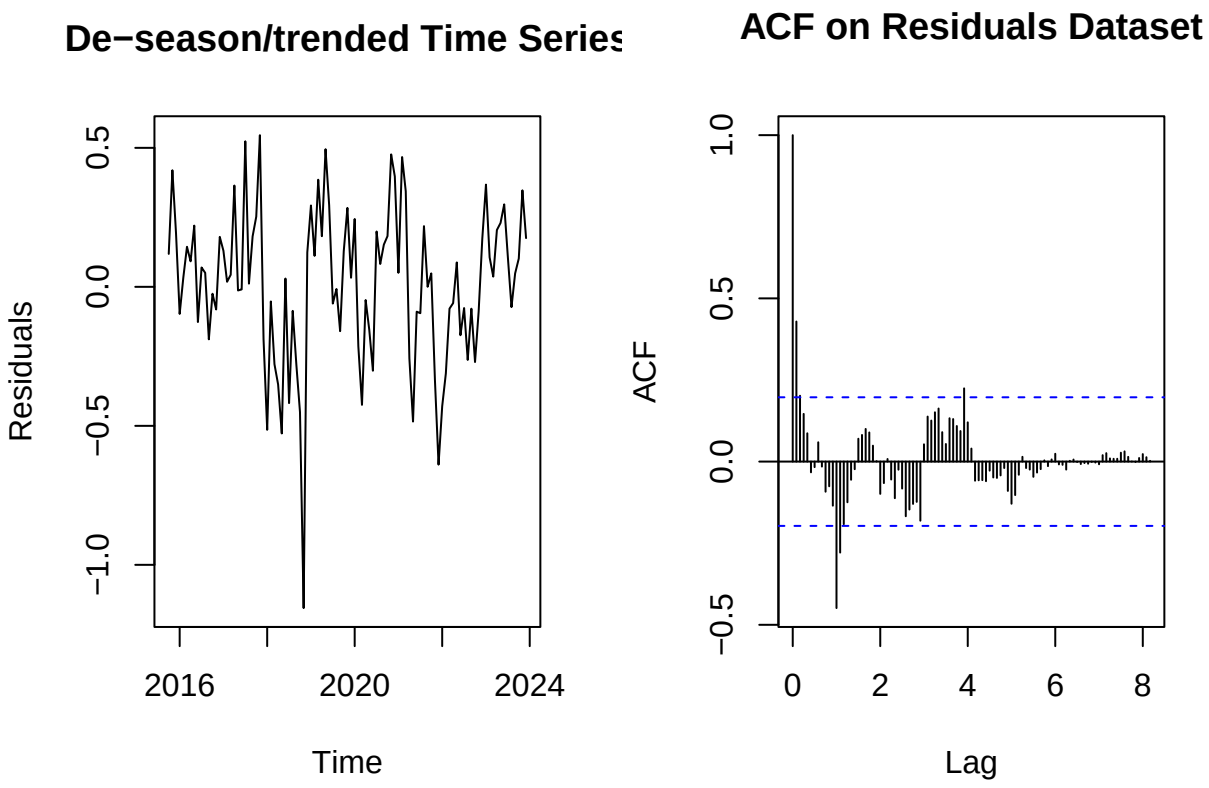

Figure 8: Differenced Method for de-seasonlized and de-trending Time series

Testing Differenced Time Series

We proceed with test if the residuals of the differenced series are iid noise according to the methods provided. We will systematically check all of these methods to see whether or not the time series residuals has a dependence structure or not.

The typical hypothesis tests reviews if

H_0 = The Time Series is iid Noise, e.g. no dependence structure among residuals

H_1 = The Time Series is NOT iid Noise, e.g. there is a possible dependece structure among residuals

Sample autocorrelation function Figure 9 present the autocorrelation functions above for S1 and S2 methods yielding slightly different results. However, one might point out to the reader that altought both ACF vary, they both record independent structure at around lag 4-5 mark - which indicate that both methods produce an independent residual structure after lag 6, in our case - after 6 months the opening prices of Bitcoin are independent of each other.

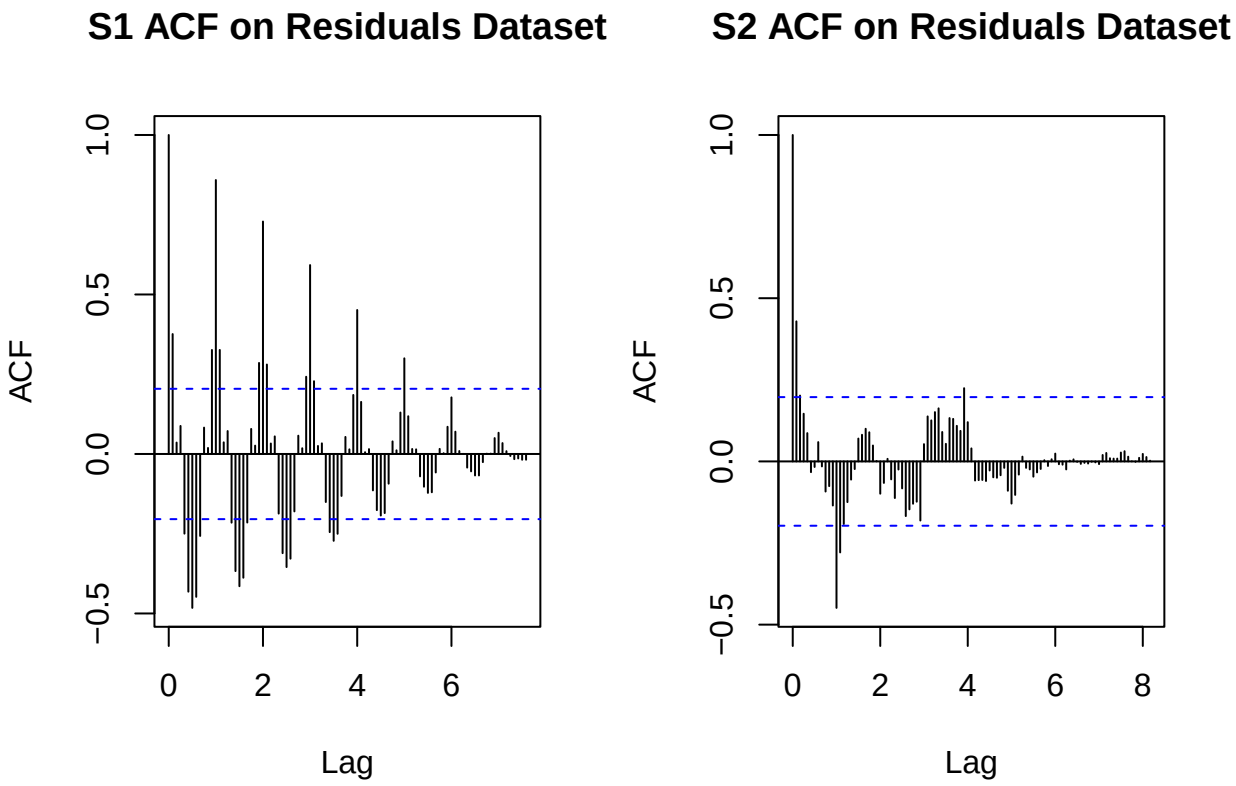

Figure 9: ACF for S1 and S2 methods on time series

Protmanteau test The resulst of both methods of Box-ljung and Box-Pierce reject the null-hypothesis H_0 with p-value < 0.000 for both residual time series. Implying that these series are not an iid sequence and has a clear dependence structure.

```
##
##  Box-Pierce test
##
## data:  S1.Residuals
## X-squared = 13.004, df = 1, p-value = 0.0003109

##
##  Box-Pierce test
##
## data:  S2.Residuals
## X-squared = 18.228, df = 1, p-value = 1.96e-05

##
```

```
##  Box-Ljung test
##
## data:  S1.Residuals
## X-squared = 13.432, df = 1, p-value = 0.0002473
```

```
##
##  Box-Ljung test
##
## data:  S2.Residuals
## X-squared = 18.786, df = 1, p-value = 1.463e-05
```

Turning point test The results from the turning point test output, H_0 was rejected for method S1, however the test was unable to reject the S2 methods residuals. Implying that they may be an underlying iid sequence.

```
##
##  Turning Point Test
##
## data:  S1.Residuals
## statistic = -3.9958, n = 92, p-value = 6.447e-05
## alternative hypothesis: non randomness
```

```
##
##  Turning Point Test
##
## data:  S2.Residuals
## statistic = -0.88212, n = 99, p-value = 0.3777
## alternative hypothesis: non randomness
```

Difference-sign test Results for the difference-sign test were similar to the turning point test output, H_0 was rejected for method S1 residuals, however the test was unable to reject the S2 methods residuals. Implying that they may be an iid sequence.

```
##
##  Difference Sign Test
##
## data:  S1.Residuals
## statistic = -3.0533, n = 92, p-value = 0.002263
## alternative hypothesis: nonrandomness
```

```
##
##  Difference Sign Test
##
## data:  S2.Residuals
## statistic = 0.34641, n = 99, p-value = 0.729
## alternative hypothesis: nonrandomness
```

Mann-Kendall Rank test The Mann-Kendall Rank test was unable to reject both H_0 for both S1 and S2 methods of the residuals. Implying that both of them may be an iid sequence.

```
##
##  Mann-Kendall Rank Test
##
## data:  S1.Residuals
## statistic = -0.82971, n = 92, p-value = 0.4067
## alternative hypothesis: trend

##
##  Mann-Kendall Rank Test
##
## data:  S2.Residuals
## statistic = -0.21464, n = 99, p-value = 0.83
## alternative hypothesis: trend
```

Checking for normality In case the series has a normal distribution, we would then be able to infer stronger assumptions and make better predictions.

We begin visually, reviewing the normality assumption of the residuals using quantile plots

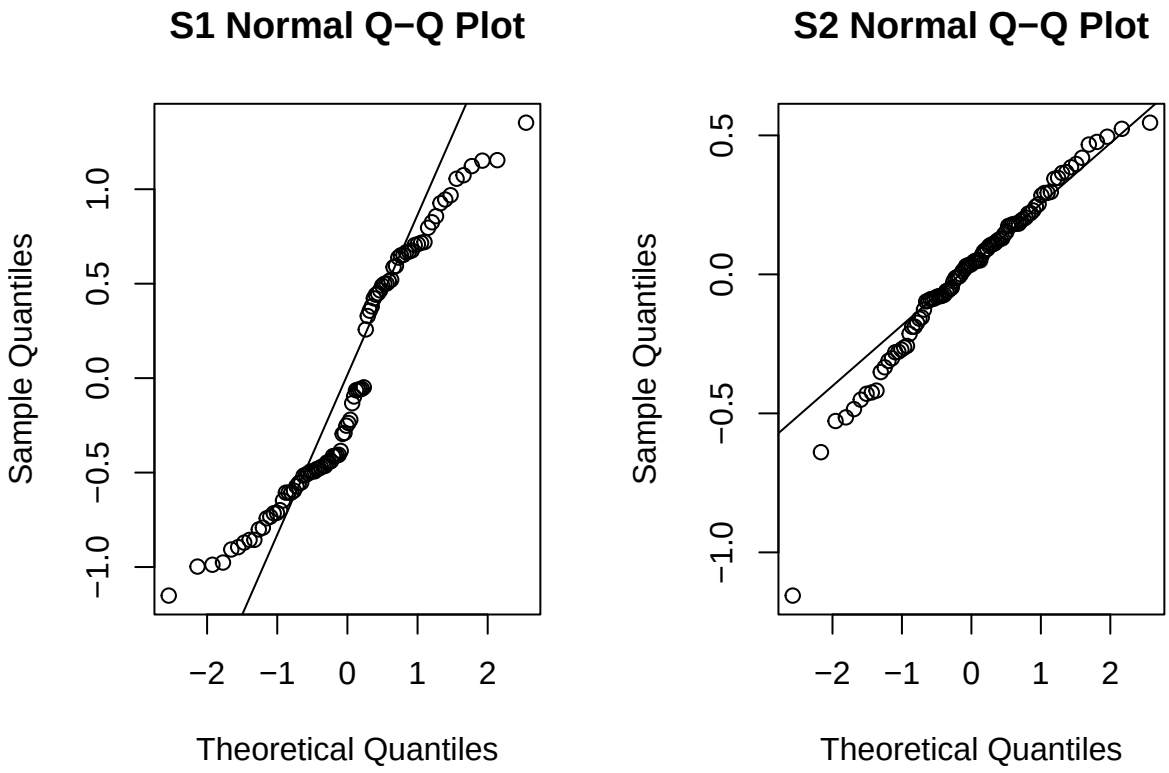

Figure 10: Q-Q plots for Normality of Residuals of S1 and S2 Method

From Figure 10, it seems that the time series may be normally distributed indeed for S2, however it may be a stretch to assume the S1 residuals are normal. We must perform a statistical hypothesis test to evaluate it more carefully and accurately.

Proceeding with the statistical tests, the Jarque–Bera statistic tests the residuals of the fit for normality based on the observed skewness and kurtosis. Atleast for S1 residuals it appears that the residuals have

some non-normal skewness and kurtosis to the time series. The Shapiro–Wilk statistic tests the residuals of the fit for normality based on the empirical order statistics. Below we see the results of both tests

```
##
##  Shapiro-Wilk normality test
##
## data:  S1.Residuals
## W = 0.92675, p-value = 6.651e-05
```

```
##
##  Shapiro-Wilk normality test
##
## data:  S2.Residuals
## W = 0.96096, p-value = 0.004993
```

```
##
##  Jarque Bera Test
##
## data:  S1.Residuals
## X-squared = 7.2991, df = 2, p-value = 0.026
```

```
##
##  Jarque Bera Test
##
## data:  S2.Residuals
## X-squared = 27.084, df = 2, p-value = 1.315e-06
```

```
##
##  Anderson-Darling normality test
##
## data:  S1.Residuals
## A = 2.8171, p-value = 3.834e-07
```

```
##
##  Anderson-Darling normality test
##
## data:  S2.Residuals
## A = 0.62743, p-value = 0.09938
```

```
##
##  Shapiro-Francia normality test
##
## data:  S1.Residuals
## W = 0.93301, p-value = 0.0003143
```

```
##
##  Shapiro-Francia normality test
##
## data:  S2.Residuals
## W = 0.95682, p-value = 0.00361
```

Spectral Analysis

Proceeding with the steps using spectral analysis to see whether or not there may be underlying periodicity within the transformed time series using S1 and S2 methods. We proceed with the above mentioned steps, by calculating a perriodogram, extracting the frequencies and reviewing the spectral density plot as can be seen in figure 11.

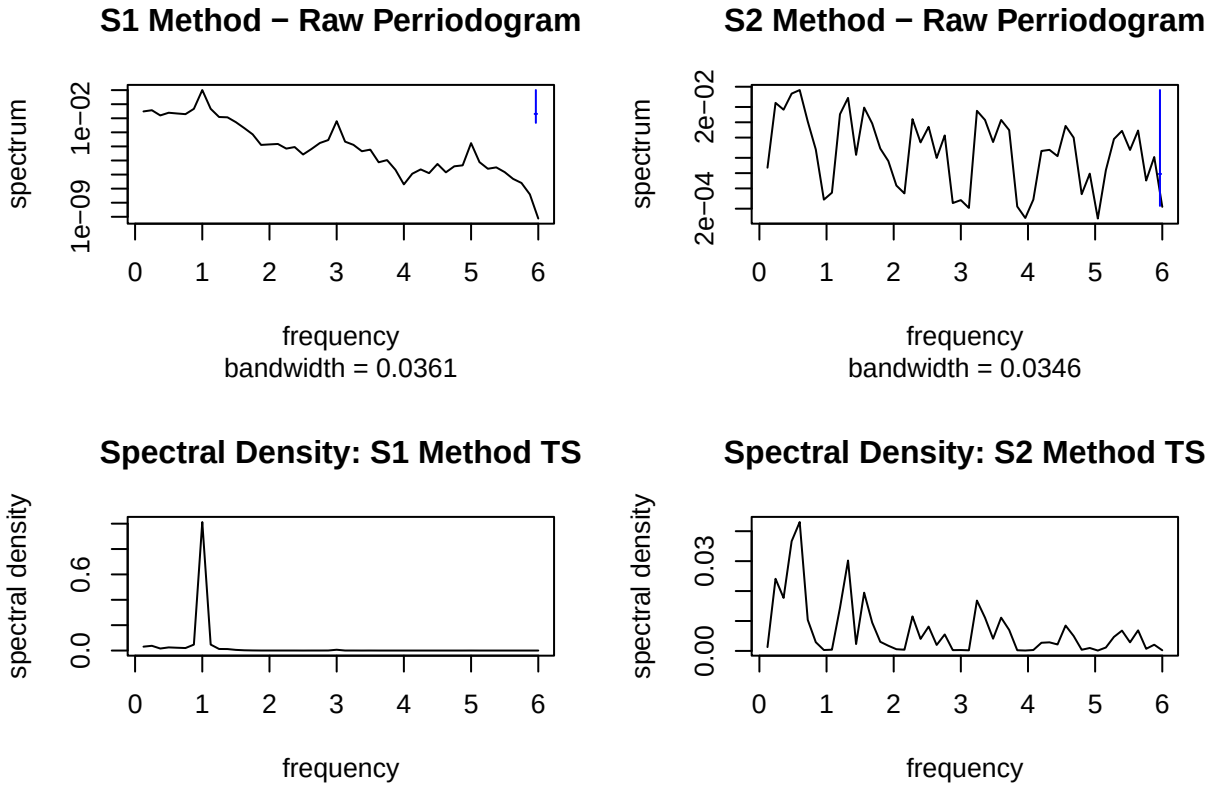

Figure 11: Spectral Analysis on S1 and S2 data

For the method S1 the spectral density clearly shows a frequency of 1 e.g. 1 year frequency. Although, It seems as S2 method doesn't produce a clear frequency, a possible interpretation is that there's a more complex underlying process that is not accounted for within the data. Thus, we proceed with differencing the time series once, in accordance with the frequency.

```
library(forecast)
nsdiffs(S1.Method.ts); ndiffs(S1.Method.ts); nsdiffs(S2.Method.ts); ndiffs(S2.Method.ts)
```

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

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

```
## [1] 0
```

```
## [1] 0
```

Since poor results are shown for S2 method, firstly with the incomplete spectral density est. and now with the subsequent 0 differencing required for stationarity we omit the S2 method times series and proceed with solely the S1 time series.

We conduct the augment dicker fuller test, to check for stationarity within our differenced time series. The null hypothesis is whether the time series is a random walk versus the alternative hypothesis which implies the time series is a stationary process as mentioned within the methods section.

```
adf.test(TS.diff)
```

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

##
##  Augmented Dickey-Fuller Test
##
## data:  TS.diff
## Dickey-Fuller = -18.949, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Therefore the null hypothesis is rejected due to the p-value being smaller than 0.05.

This indicates that the time series is stationary. To put it another way, it has some time-independent structure and does exhibit constant variance over time.

However, we proceed with checking the time series visually using ACF and PACF plot seen in figure 12. These plots provide us with an intuition that there may still be an underlying process unaccounted for.

Fitting the ARMA model

We proceed with fitting an ARMA model firstly checking systematically for the AR process, due to uncertain results from the ACF and PACF plots. Which can be studied closer in the R code below.

```
# Finding the best order using AIC, AR
ar_mod = ar(TS.diff,order=25,aic=T) # Search among p=1,2,,...,25
ar_mod$aic # differences in AIC between each model and the best-fitting mode
```

```
##          0          1          2          3          4          5          6
## 322.353895 222.027874 147.663205 149.569157  92.949588  59.964603  61.917061
##          7          8          9         10         11         12         13
##  37.473650   5.253333   0.000000   1.295797   2.122424   4.122366   6.049363
##         14         15         16         17         18         19         20
##   8.048536   9.100220  10.877405  12.676852  14.501922  16.473642  18.459517
##         21         22         23         24         25
##  20.281768  22.239821  24.210410  26.184270  28.099085
```

```
# 'Best', aka lowest AIC is 0 for AR(9)


# Regular ARMA(1,1)
Model_1 <- arima(TS.diff,
                 order=c(1,0,1),
                 method="ML",
                 include.mean=F)
Model_1$aic
```

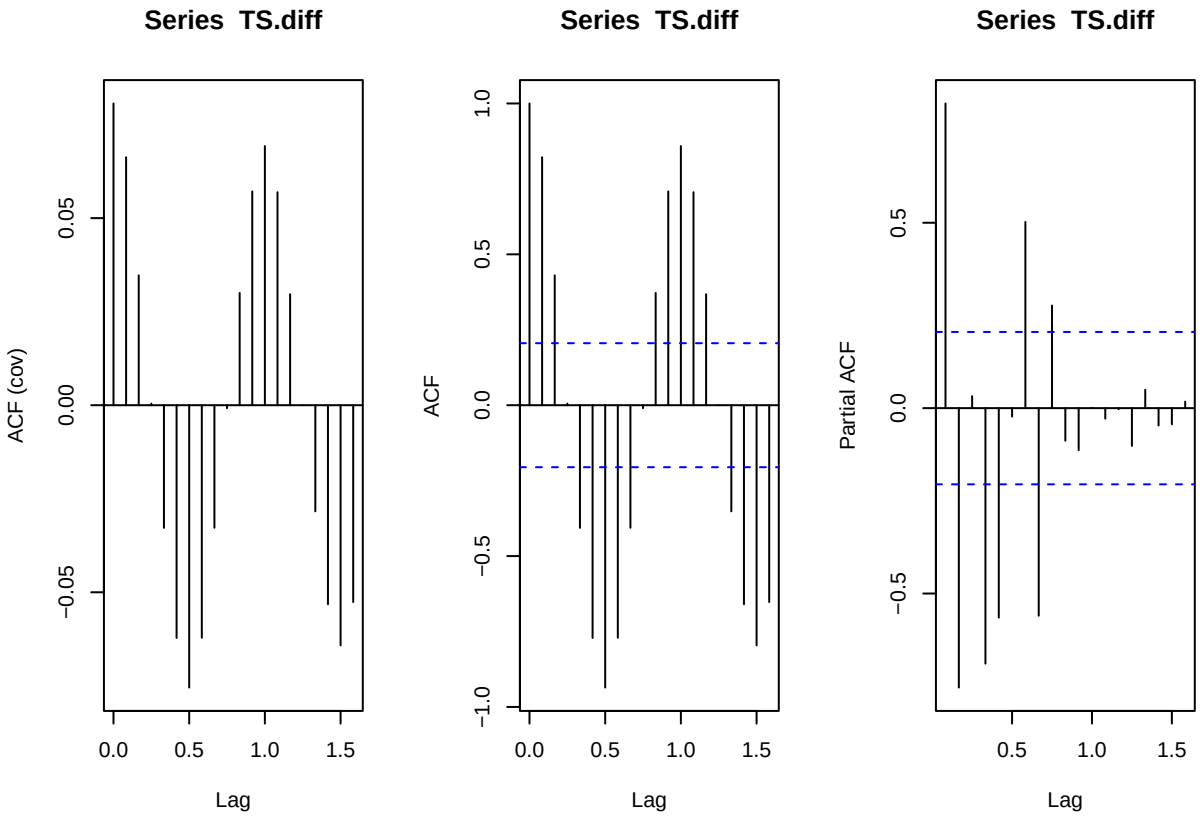

Figure 12: ACF for differenced Time Series

```
## [1] -146.1612
```

```
# AIC -146


# Fitting ARMA(9,1) model with ML-estimation
Model_2 <- arima(TS.diff,order=c(9,0,1),method="ML", include.mean=F)
Model_2$aic
```

```
## [1] -482.7571
```

```
# Lowest AIC -482.75
```

We end up selecting the ARMA(9,1) model based on the plot and AIC information criterion.

Testing Residuals

```
##
##  Box-Pierce test
##
## data:  Mod2_res
## X-squared = 0.073265, df = 1, p-value = 0.7866
```

```
##
##  Box-Ljung test
##
## data:  Mod2_res
## X-squared = 0.075707, df = 1, p-value = 0.7832
```

```
##
##  Turning Point Test
##
## data:  Mod2_res
## statistic = -0.083712, n = 91, p-value = 0.9333
## alternative hypothesis: non randomness
```

```
##
##  Difference Sign Test
##
## data:  Mod2_res
## statistic = 0, n = 91, p-value = 1
## alternative hypothesis: nonrandomness
```

```
##
##  Mann-Kendall Rank Test
##
## data:  Mod2_res
## statistic = 2.1289, n = 91, p-value = 0.03326
## alternative hypothesis: trend
```

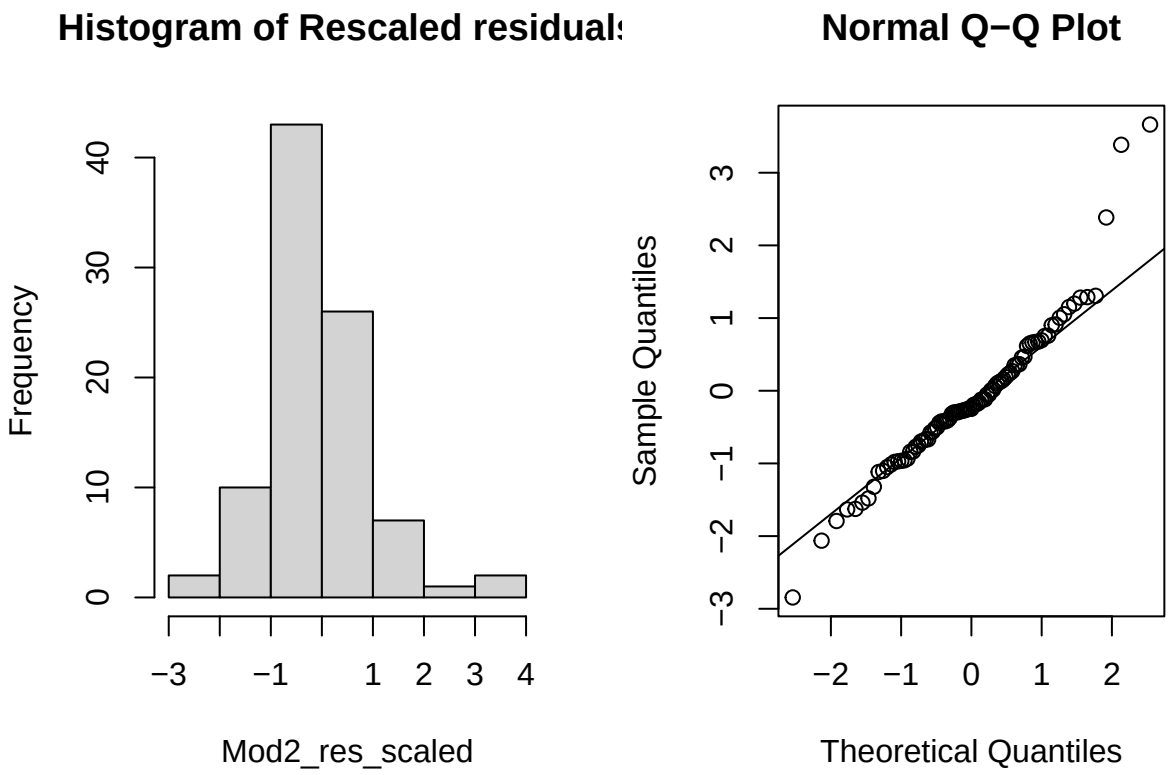

Figure 13: Normality Assumption checks

```
##
##  Shapiro-Wilk normality test
##
## data:  Mod2_res
## W = 0.94047, p-value = 0.0004174
```

```
##
##  Jarque Bera Test
##
## data:  Mod2_res
## X-squared = 42.533, df = 2, p-value = 5.809e-10
```

```
##
##  Anderson-Darling normality test
##
## data:  Mod2_res
## A = 1.1852, p-value = 0.004072
```

```
##
##  Shapiro-Francia normality test
##
## data:  Mod2_res
## W = 0.93242, p-value = 0.0003178
```

```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 12.471, df = 18, p-value = 0.822
##
## Model df: 0.   Total lags used: 18
```

Judging from the hypothesis tests and rescaled residuals, as can be seen in figure 13 and figure 14 we can see an approximately normally distributed residual - although the tails may be too wide, thus making the prediction less reliable. However, we proceed with the function coeftest within the lmtest package to see each significance level for each parameter. See blow for output.

```
coef.mod2 <- coeftest(Model_2)
coef.mod2
```

```
##
## z test of coefficients:
##
##        Estimate  Std. Error  z value Pr(>|z|)
## ar1  0.89825550  0.10727743   8.3732   <2e-16 ***
## ar2 -0.08306871  0.14693646  -0.5653   0.5718
## ar3  0.10096279  0.10861497   0.9295   0.3526
## ar4 -0.00067667  0.01884597  -0.0359   0.9714
## ar5 -0.00049181  0.01625149  -0.0303   0.9759
## ar6 -0.98968371  0.01177085 -84.0792   <2e-16 ***
## ar7  0.88833321  0.10810768   8.2171   <2e-16 ***
```

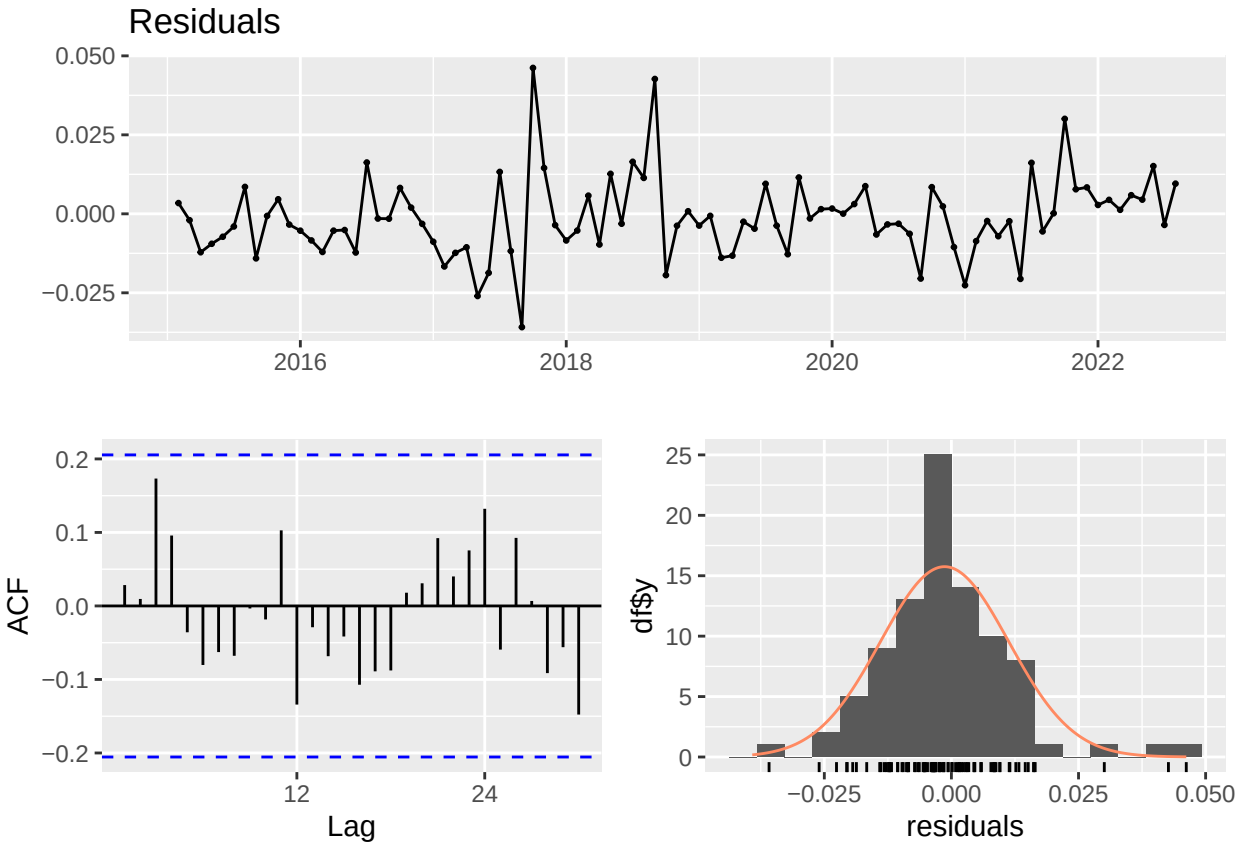

Figure 14: Residual checks

```
## ar8 -0.08013205  0.14778993  -0.5422   0.5877
## ar9  0.10113555  0.10859669   0.9313   0.3517
## ma1  0.99999595  0.03764587  26.5632   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results of the function coeftest yields significant autoregressive orders for AR(1), AR(6), AR(7) and for the moving average MA(1). This is somewhat inconsistent with the minimum order selected by the AR estimation using AIC previously, as it yielded 9.

Reviewing causality and invertibility

To inspect whether or not the model is stationary. We plot all of the inverse complex roots. From the figure we all can inspect that the roots are within the unit circle for both inverse AR and MA roots (i.e < 1), which implies that the model is stationary, causal and invertible. We can see from figure 15 that all roots are within the unit circle.

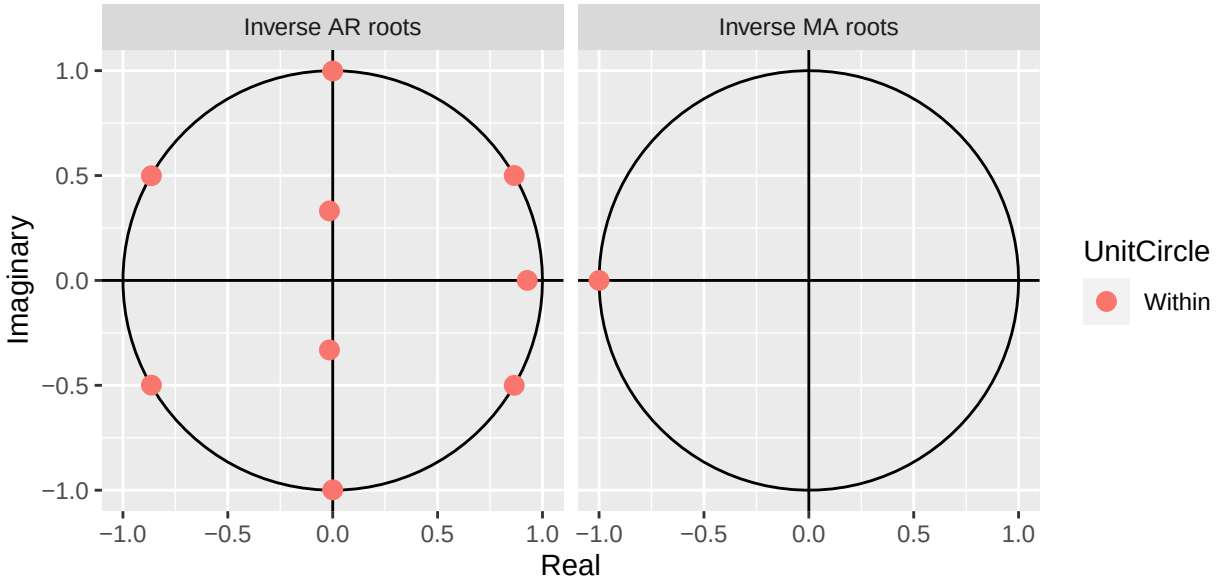

Figure 15: Inverse Roots AR and MA plots

Seasonality ARIMA

Extending the ARMA model we used to include a seasonal component we train our thrid model on the previously undifferenced dataset.

```
##
## Call:
## arima(x = log.train.data, order = c(9, 0, 1), seasonal = c(1, 0, 1))
##
## Coefficients:
##          ar1     ar2      ar3     ar4      ar5     ar6     ar7      ar8
##       0.4451  0.8413  -0.2538  0.0406  -0.1442  0.1314  0.0970  -0.0836
## s.e.  0.1568  0.1986   0.1496  0.1427   0.1400  0.1595  0.1532   0.1286
##           ar9     ma1    sar1     sma1  intercept
##       -0.0926  0.8953  0.8612  -0.9999     8.1582
## s.e.   0.1075  0.1227  0.1224   0.1426     0.8712
##
## sigma^2 estimated as 0.02932:  log likelihood = 29.35,  aic = -30.7

##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1        0.445052   0.156816  2.8380  0.004539 **
## ar2        0.841272   0.198636  4.2352 2.283e-05 ***
## ar3       -0.253849   0.149634 -1.6965  0.089797 .
## ar4        0.040558   0.142658  0.2843  0.776177
## ar5       -0.144169   0.139955 -1.0301  0.302957
## ar6        0.131428   0.159521  0.8239  0.410000
## ar7        0.096981   0.153231  0.6329  0.526793
## ar8       -0.083590   0.128613 -0.6499  0.515733
## ar9       -0.092581   0.107546 -0.8608  0.389321
## ma1        0.895277   0.122733  7.2945 2.997e-13 ***
## sar1       0.861237   0.122374  7.0378 1.953e-12 ***
## sma1      -0.999950   0.142589 -7.0128 2.336e-12 ***
## intercept  8.158242   0.871179  9.3646 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Having fitted the SARIMA model, we review the output of the aforementioned coeftest function, and see that the 'sar1' and 'sma1' are both significant with a p value of 0.000.

Figure 16 illustrates the inverse roots for AR and MA processes and provides us with a stable SARIMA estimation, due to the fact that all roots are within the unit circle.

Checking the residual diagnostics of the SARIMA model in figure 17, we can see an approximately iid noise.

```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 10.03, df = 20, p-value = 0.9676
##
## Model df: 0.   Total lags used: 20
```

Forecasting Predictions and Evaluating Models

We continue with forecasting our test dataset using our fitted third SARIMA model below in figure 18. We can see that although the predictions may not align completely, the predictions are within the 95% confidence interval.

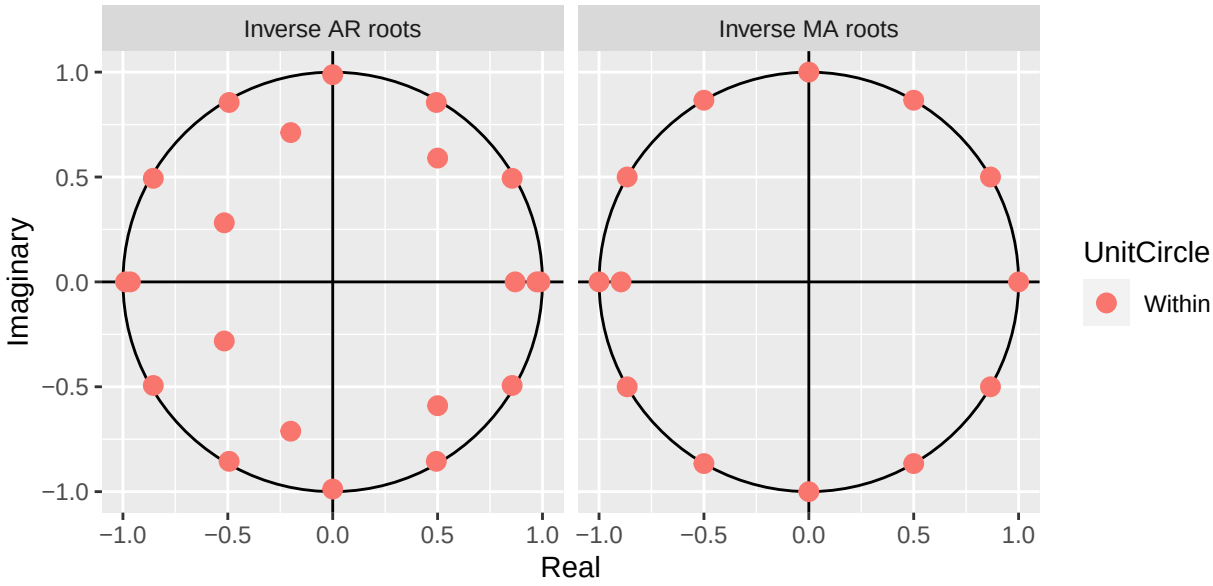

Figure 16: Checking Roots of AR and MA

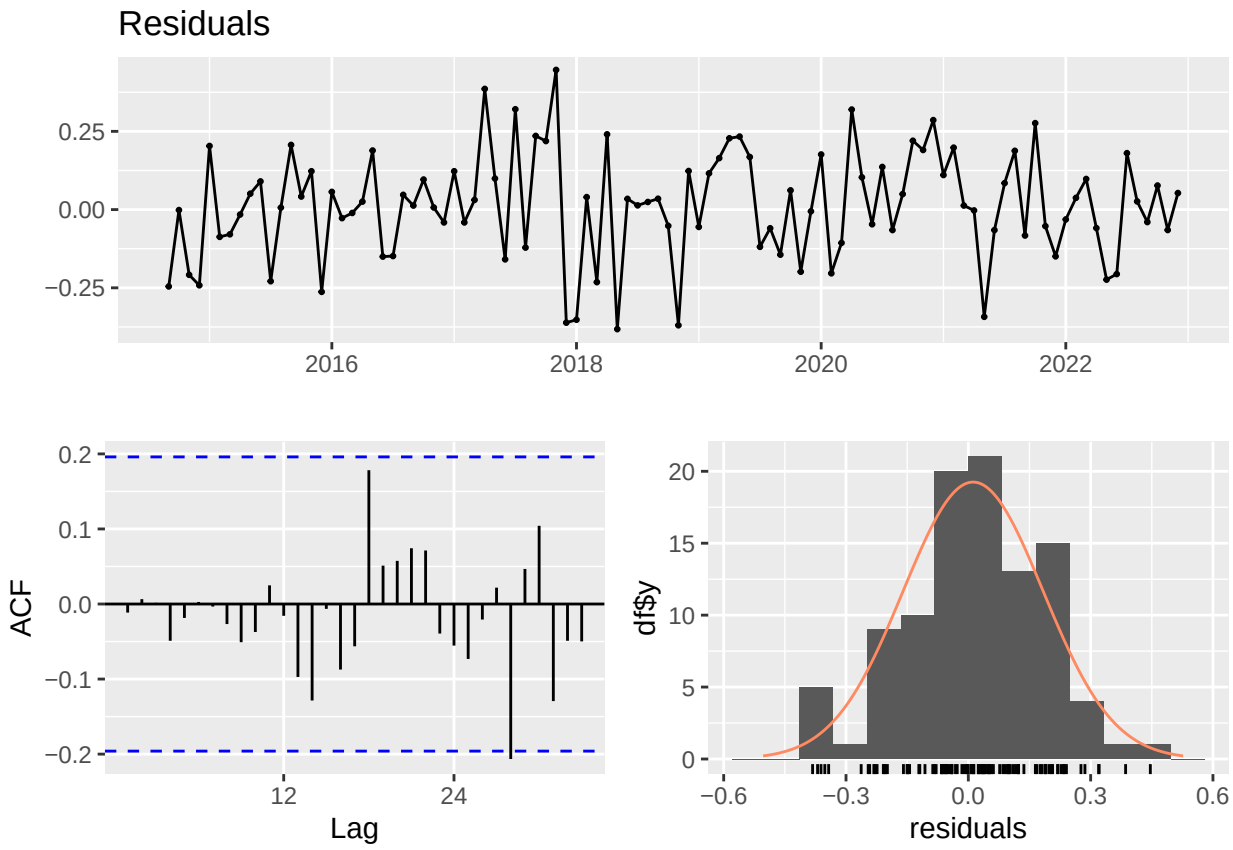

Figure 17: SARIMA Diagnostics of Residuals

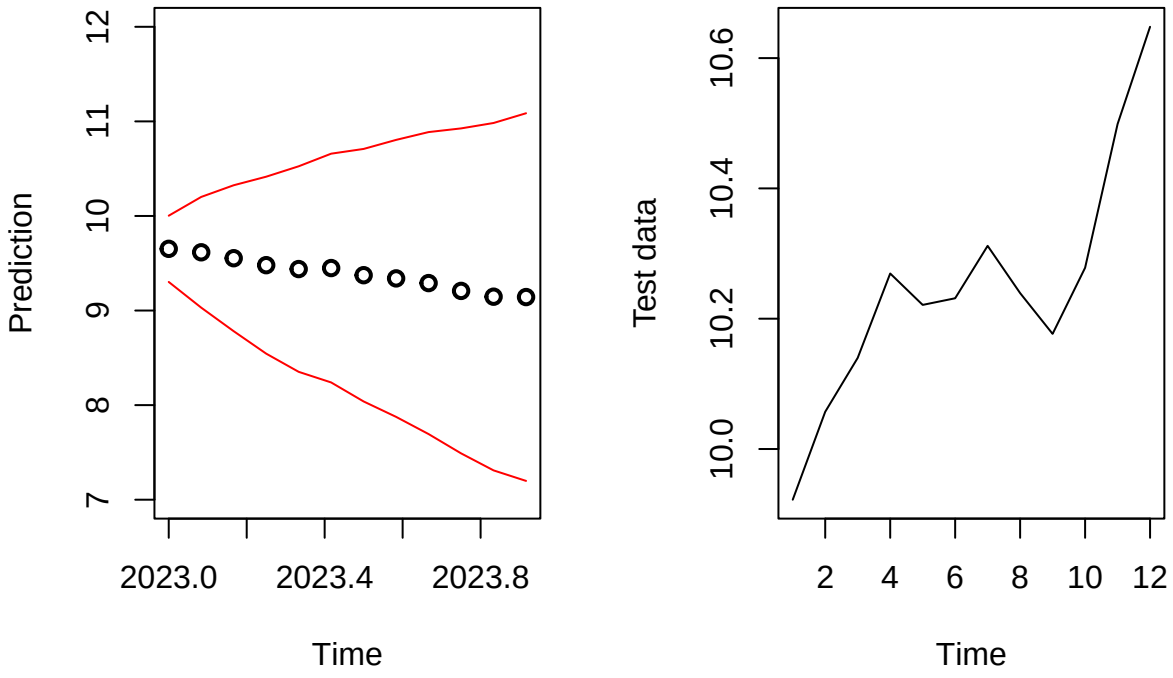

Figure 18: Forecast with SARIMA(9,0,1) Model with 95 CI

Calculating MSE yields from the model, we get

```
MSE3 = sum((as.numeric(forecast_mod3$pred) - as.numeric(log.test.data))^2)
MSE3
```

```
## [1] 10.17593
```

Benchmarking

In order to evaluate our performace additionally we use auto.arima which selects orders automatically and we compare our results by forecasting our test dataset. Our

```
## Series: log.train.data
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.3655
## s.e.  0.0928
##
## sigma^2 = 0.03369:  log likelihood = 27.78
## AIC=-51.56   AICc=-51.44   BIC=-46.37
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.365528   0.092775    3.94 8.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

auto.arima selects an AR(1) process as the best option. As can be studied in the output above. We subsequently replicate the previous forecast and evaluate the perfomance with MSE. In figure 19 we see the predicted values together with the real test data set.

Calculating MSE yields

```
MSE4 = sum((as.numeric(forecast_mod4$pred) - as.numeric(log.test.data))^2)
MSE4
```

```
## [1] 3.722912
```

So far the AR(1) process selected by auto.arima produces the lowest MSE with 3.72. We proceed with evaluating all of our created models, and measure the MSE performance for each.

```
MSE1;MSE2;MSE3;MSE4
```

```
## [1] 1261.122
```

```
## [1] 1258.881
```

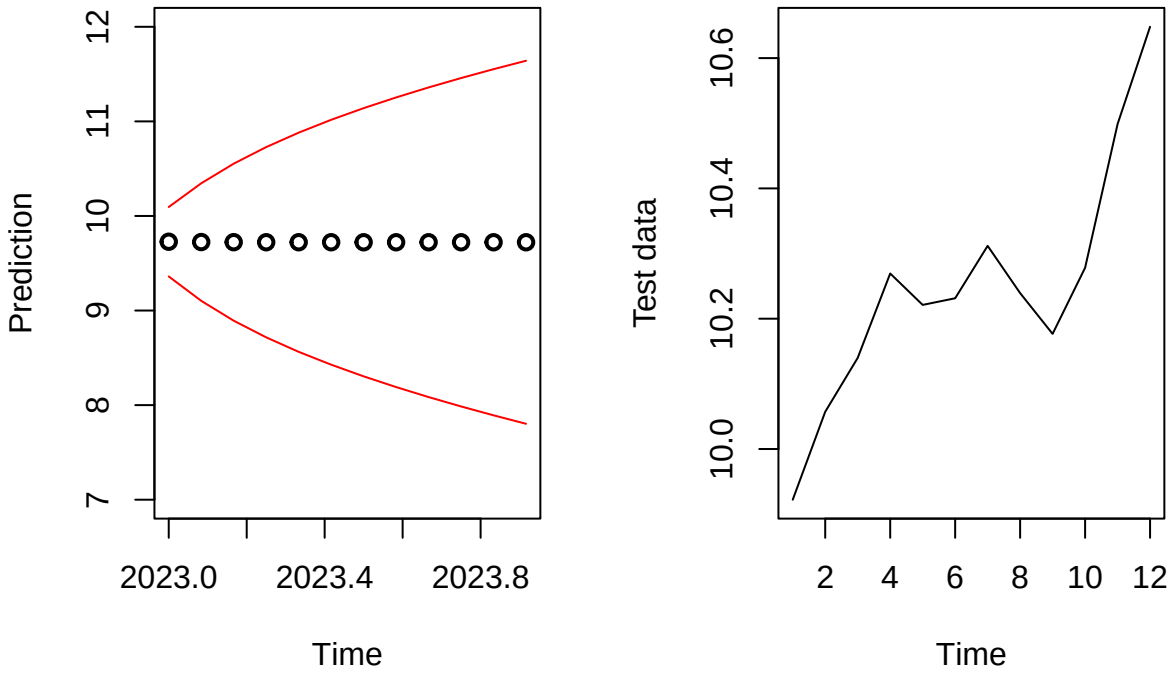

Figure 19: Forecast with auto.arima AR(1) Model with 95 CI

```
## [1] 10.17593
```

```
## [1] 3.722912
```

Overall, the less complicated model choosen by AUTO.ARIMA scores the lowest, e.g. has the best predictive power.

General Conclusion

The Residuals are not normal, nor are they independent. Although some visual plots show resemblance of normality and some test are unable to reject H_0 in many cases e.g. implying the residuals are iid. However, several methods points to the fact that there is a clear dependence structure and many p-values suggest strongly non-normality is present within the time series. However, many tests show stationarity within the differenced data which is also confirmed using the inverse AR and MA roots plots.

In general I tried to estimate the most reliable model using the AIC criterion, perhaps that was a poor choice on my part, as the dataset is highly dependent on each previous observation as shown by the selection of a simple AR(1) model by auto.arima. In addition, selecting a more simple model is always perferential to a more complicated model such as the ARMA(9,1), with half of the parameters being unsignificant during the coeftest z tests.

Futhermore, I am very suspicious of my initial 'down sampling' and smoothing of the dataset. Using 30 days as an average price point for bitcoin is naive at best. Especially since the concurrency market is highly volatile and adjusting rapidly around regulatory realities as well as other impacts for instance Elon Musks tweets. Resulting in this classical estimation methods being unsuitable to forecast accurately monthly prices of Bitcoin, or perhaps I made a fatal mistake which made the data silly and simply unusable.