Time Series Report

Time Series - Umeå University

Artem Shiryaev

2024-01-29

Introduction

The goal of this assignment is to familiarize us with time series analysis, and present the results of the analysis in a concise and clear manner.

The tasks and answers will be presented below in a systematic fashion, where the question will be posted - followed by the solution. The following material can be accessed from my GitHub repo, forked and ran by yourself using R. Link: https://github.com/ArtemShiryaev

Tasks

Task 1

Find a time series data that (can) includes a seasonal component, for example, quarterly data, monthly data. Make sure that the series is fairly long, at least 10 years.

Solution

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.

```r
# Load time series data, in this case daily Bitcoin prices
data <- read.csv("~/University/Education/Mathematics/Avancerade Kurser/VT24 Time Series/Assignments/Lab

# Format the dataframe into a time series object
formated.data <- ts(data = data[,2:7],
                    start = c(2014, 288),
                    frequency = 365
)
```

Here we can see the aforementioned time series in Figure 1.

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

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

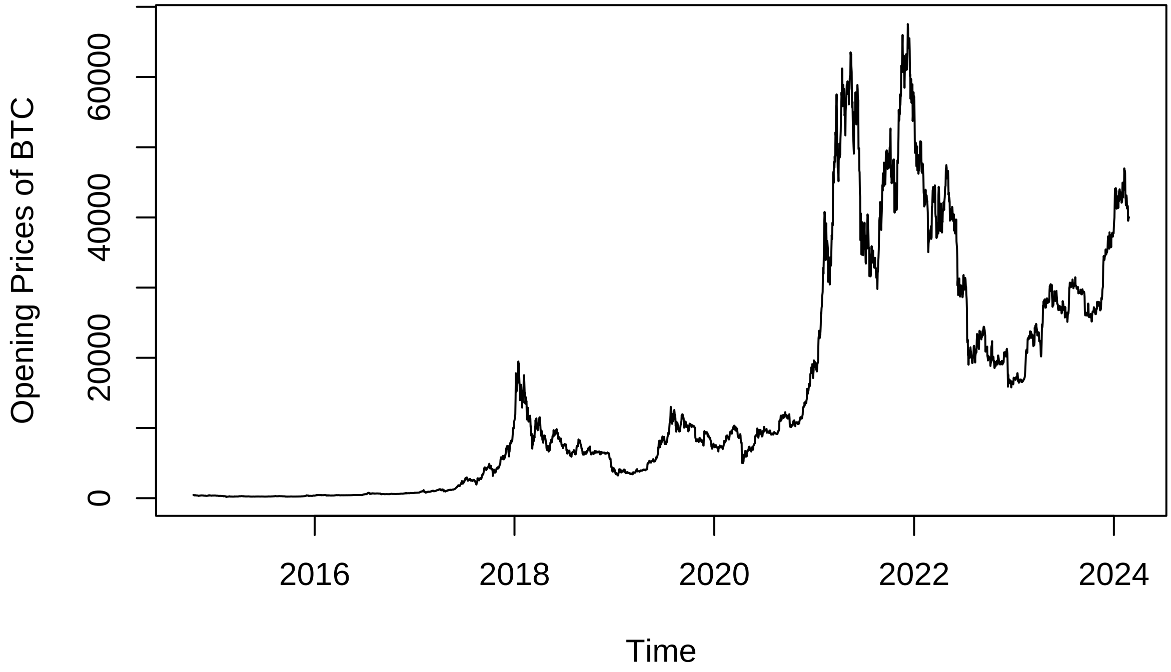

Figure 1: Daily Opening Prices of Bitcoin

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

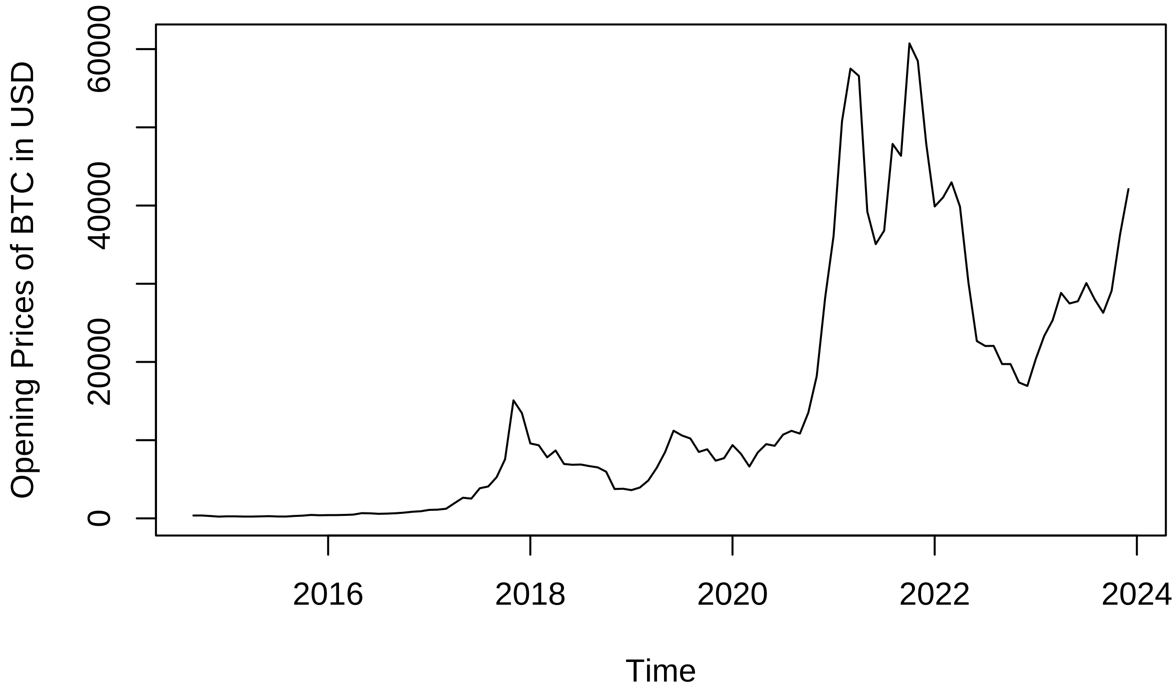

Figure 2: Monthly Opening Prices of Bitcoin

Task 2

Split the data in **two parts**, the most recent year, and the previous years. Time series analysis will be applied on the second part, and the last year data will be used as the "correct answer" when forecasting.

Solution:

```
# Selecting the previous year as test data, and remaining as training data.
train.data <- ts(monthly.data[1:100],
                    start = c(2014,9),
                    frequency = 12)

test.data <- as.ts(monthly.data[101:112],
                   start = c(2023,1),
                    frequency = 12)
```

Task 3

Start by plotting the time series and examine the main features of the graphs, i.e., check whether there is a drift, a deterministic trend, a combination of drift and a deterministic trend, a seasonal component, any apparent sharp changes in behavior, any outlying observations,

Solution

Plotting these two time series we see

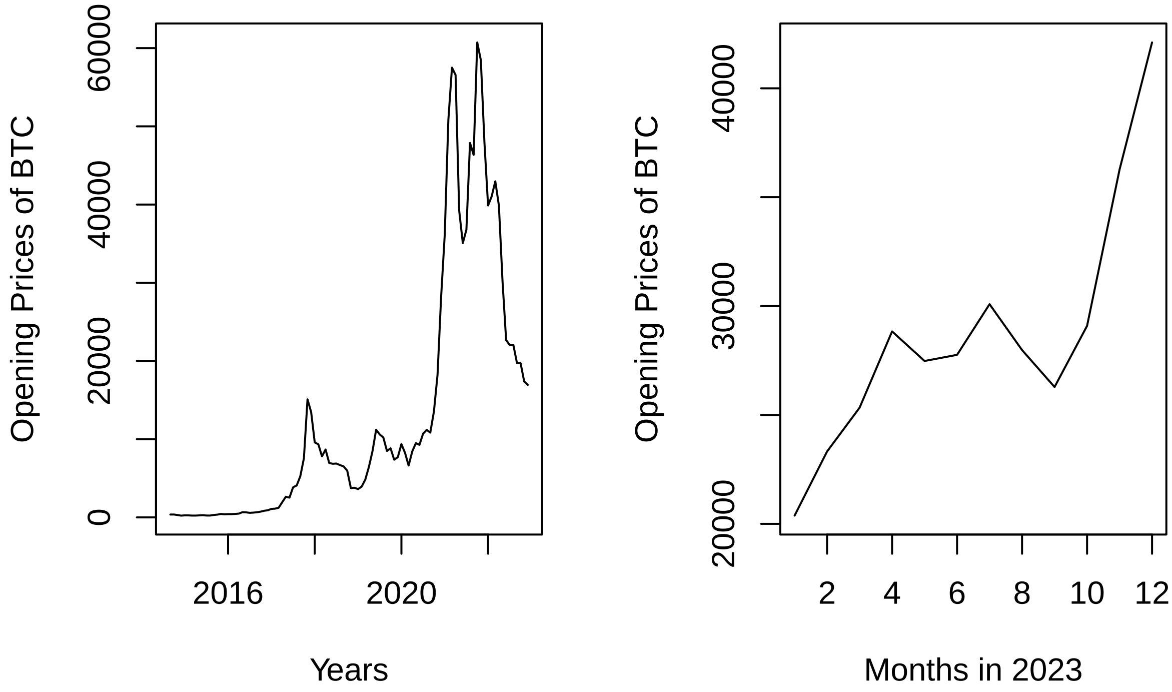

Figure 3: Daily Bitcoin Prices Split Data

We can examine from Figure 3, that we indeed 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 below we see a substantial improvement of the training dataset - reseabling more of a linear times series with drift and various seasonalities.

```
log.test.data <- log(test.data, base = exp(1))
log.train.data <- log(train.data, base = exp(1))
par(mfrow=c(1,2))
plot(log.train.data,
     ylab = "log Opening Prices of BTC",
     xlab = "Years"
)

plot(log.test.data ,
     ylab = "log Opening Prices of BTC",
     xlab = "Months in 2023"
)
```

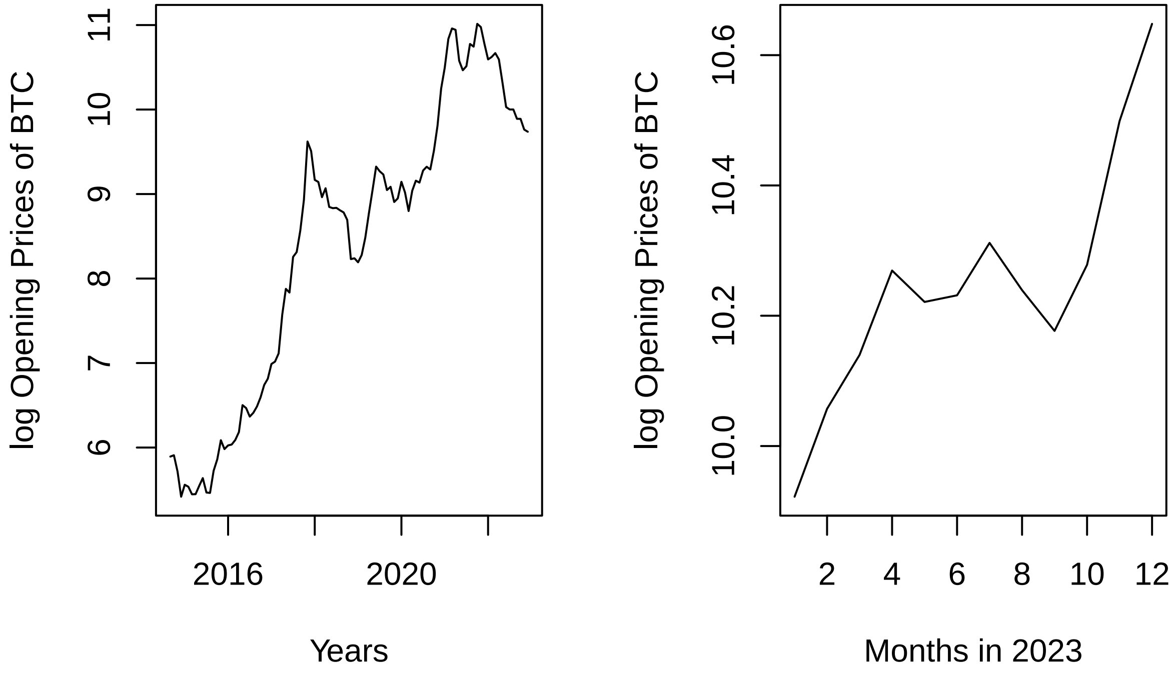

Figure 4: Daily Log Prices of Bitcoin

Task 4

Remove any drift, deterministic trend and seasonal components in order to get stationary residuals. Do that, by both using methods **S1** and **S2** (see Chapter 1.5).

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.

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

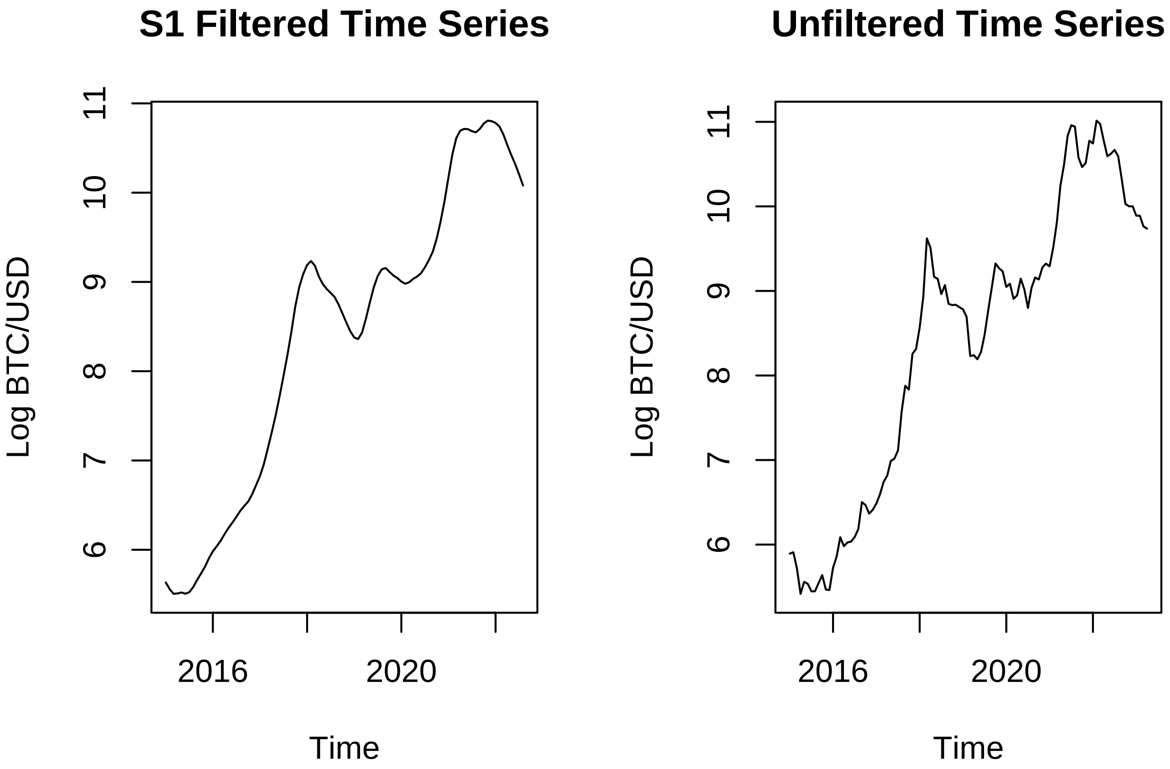

Figure 5: Filtered vs. Unfiltered Test Time Series

The second step in the S1 process is to subtract the filtered mean

Followed by subtracting each

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

We get the following results;

```
##        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
```

We obtain de-seasonalized data by

$$d_t = x_t - s_t$$

The 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

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

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.

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:

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

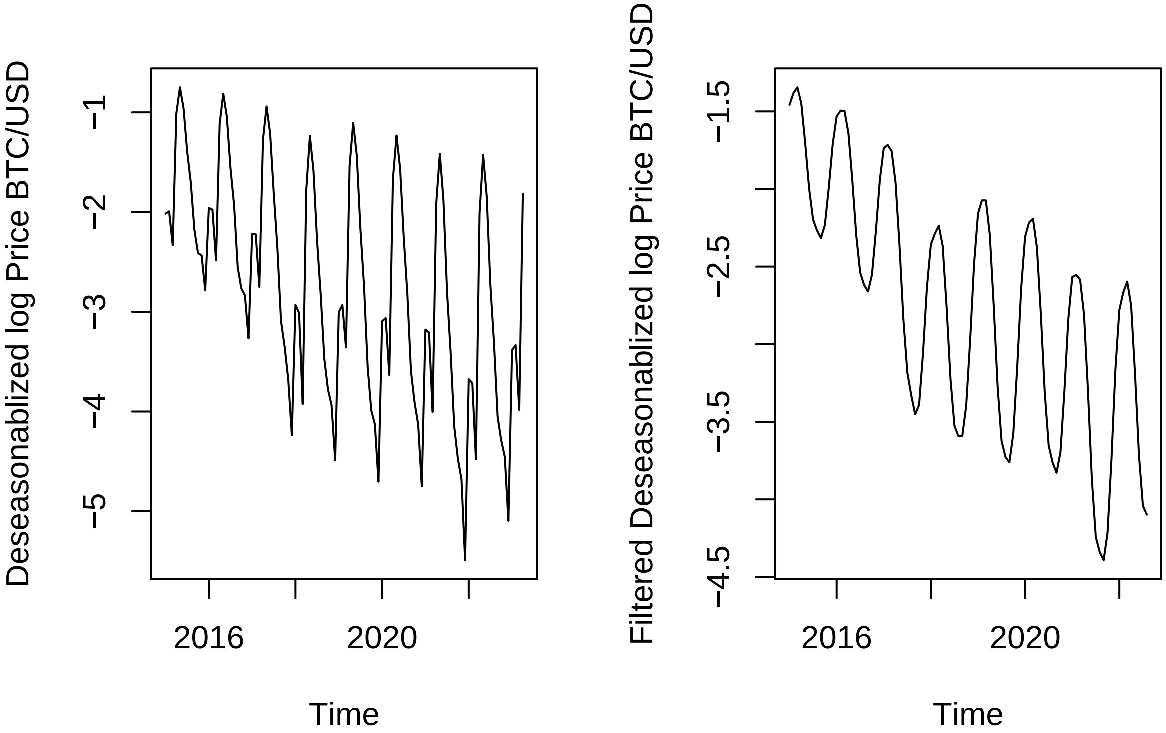

Figure 6: Deseasonalized Test Dataset

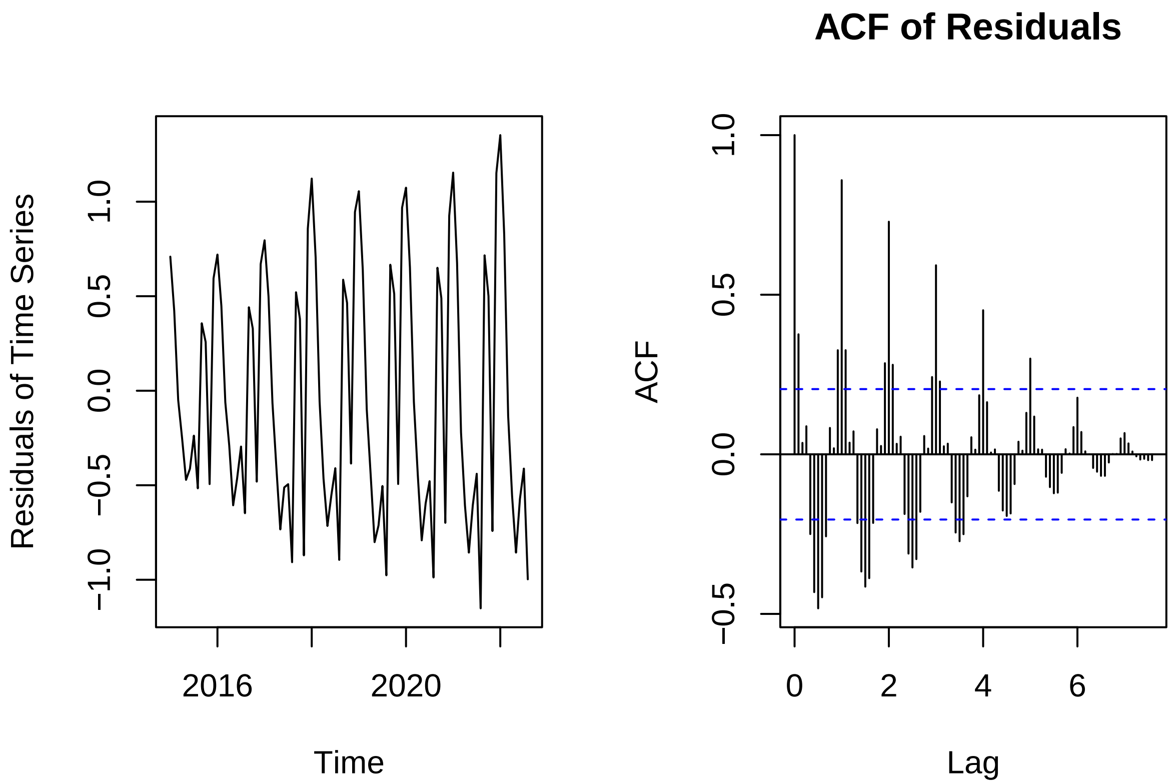

Figure 7: Residuals of Test Dataset

```r
# eliminate the seasonal component

log.data <- log(monthly.data, base = exp(1))

# eliminate the seasonal component
S2.lag12.diffed <- diff(log.data, lag = 12, differences = 1)


# eliminate the trend form the deseasonalized series
S2.trend.diffed <- diff(S2.lag12.diffed,differences = 1)
```

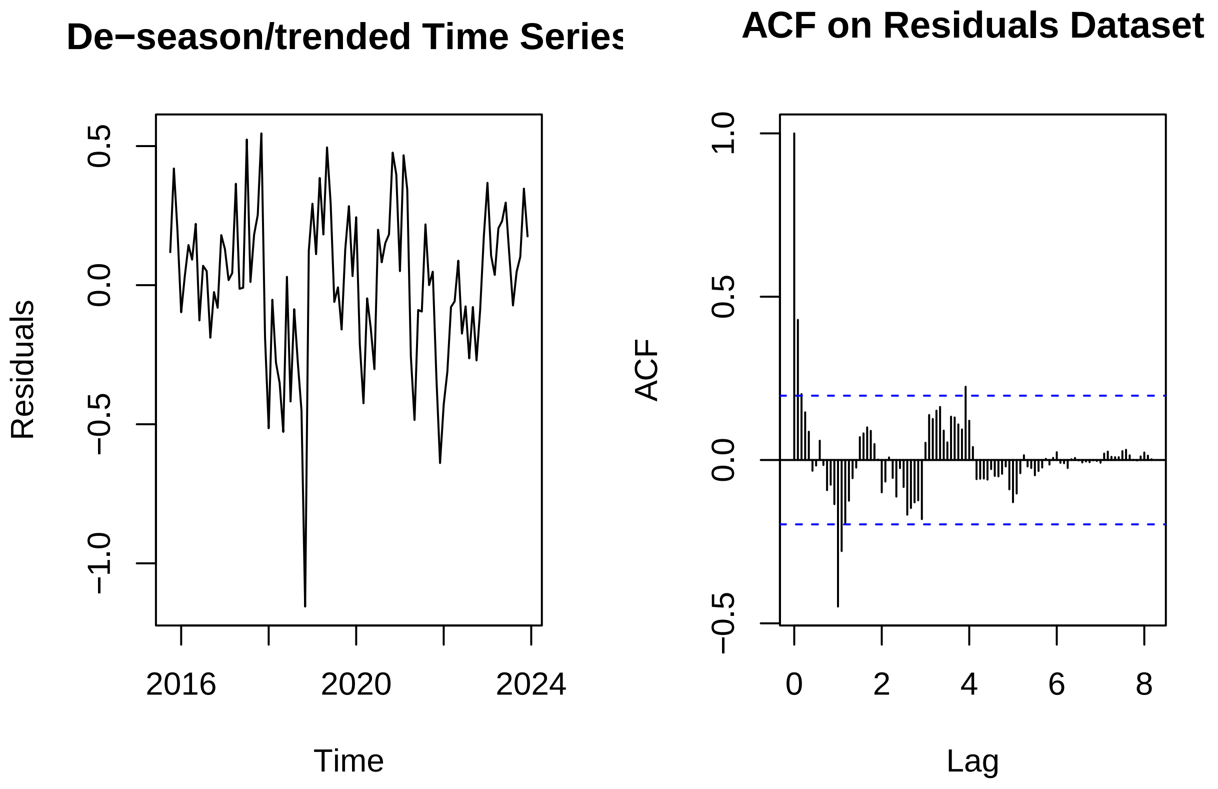

Figure 8: Differenced Method for de-seasonlized and de-trending Time series

Task 5

Test if the residuals or the differenced series are iid Noise according to the methods in Chapter 1.6.

Solution

The available testing methods we will be where we will be reviewing some of them are as follows

- Visually checking the sample autocorrelation function

- Protmanteau test
- Turning point test
- Difference-sign test
- Mann-Kendall Rank test
- Checking for normality
  - Histogram
  - qq plot
  - Normality test
    - Shapiro-Wilks test
    - Shapiro–Francia test
    - Jarque–Beratest
    - Anderson–Darling test

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

$$\mathrm{H}_0 = \text{The Time Series is iid Noise, e.g. no dependence structure among residuals}$$
$$\mathrm{H}_1 = \text{The Time Series is NOT iid Noise, e.g. there is a possible dependece structure among residuals}$$

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.

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

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

We shall now test both methods to check for independence

```r
# Box-Pierce Version of iid Sequence test
Box.test(S1.Residuals,type = "Box-Pierce", lag = 1)
```

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

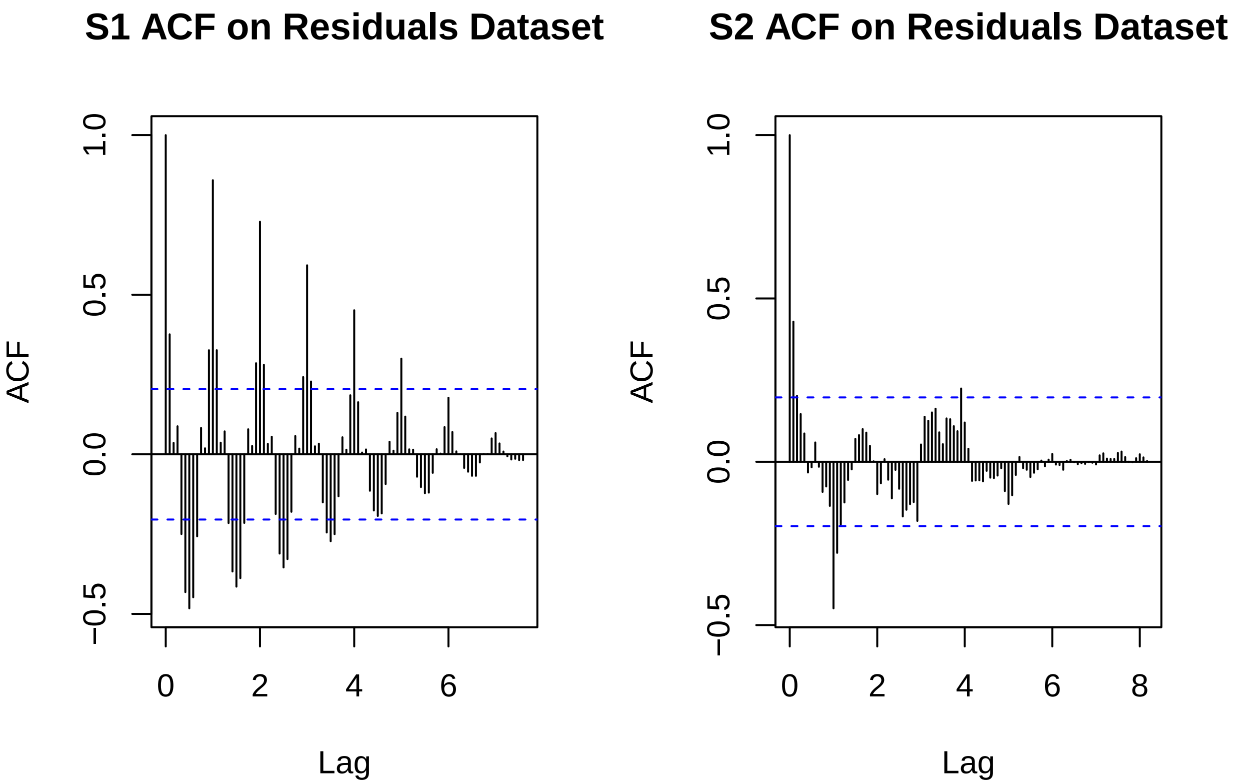

Figure 9: ACF for S1 and S2 methods on time series

```r
Box.test(S2.Residuals,type = "Box-Pierce", lag = 1)
```

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

```r
# Ljung-Box Version of iid Sequence test
Box.test(S1.Residuals,type = "Ljung-Box", lag = 1)
```

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

```r
Box.test(S2.Residuals,type = "Ljung-Box", lag = 1)
```

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

Both methods 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.

Turning point test The method by which the turning point test the residual for an iid sequence is as follows. If $Y_1, \ldots Y_n$ 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 $\mathcal{N}(0,1)$.

```r
library(randtests)
turning.point.test(S1.Residuals)
```

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

```r
turning.point.test(S2.Residuals)
```

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

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 iid sequence

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.

```r
difference.sign.test(S1.Residuals)
```

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

```r
difference.sign.test(S2.Residuals)
```

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

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.

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 $\mathcal{N}(0, 1)$.

```
rank.test(S1.Residuals)
```

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

```
rank.test(S2.Residuals)
```

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

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.

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

```
# Normality Plots
par(mfrow=c(1,2))
qqnorm(S1.Residuals,
       main = "S1 Normal Q-Q Plot")
qqline(S1.Residuals)
qqnorm(S2.Residuals,
       main = "S2 Normal Q-Q Plot")
qqline(S2.Residuals)
```

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

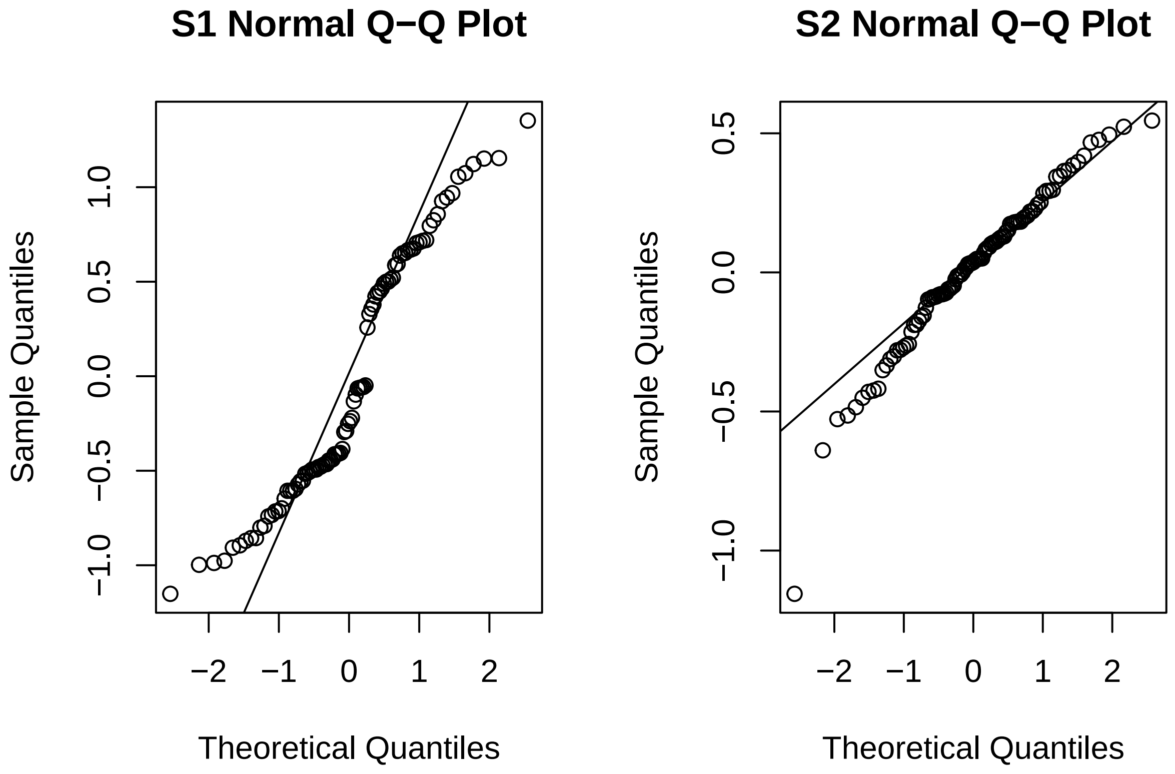

Figure 10: Q-Q plots for Normality of Residuals of S1 and S2 Method

```r
shapiro.test(S1.Residuals)
```

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

```r
shapiro.test(S2.Residuals)
```

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

```r
jarque.bera.test(S1.Residuals)
```

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

```r
jarque.bera.test(S2.Residuals)
```

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

```r
# library(nortest)
# ad.test(S1.Residuals)
# ad.test(S2.Residuals)
# sf.test(S1.Residuals)
# sf.test(S2.Residuals)
```

Normality and iid sequence 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 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.

Task 6

Regardless of the conclusion from 5, use the results from method S1 to forecast the forthcoming year (just using the estimated trend and the estimated seasonal component) and compare it to the "correct answer".

Solution

Proceeding with forecasting the year 2023 using the S1 Method for decomposition from 2014 - 2022, we firstly re-familiarize the reader with the derivations of the S1 Method.

$$X_t = m_t + s_t + Y_t,\ \ t = 1, \dots n,\ \ \text{where }, \mathbf{E}[Y_t] = 0 \tag{1}$$

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

$$s_t = w_k - \frac{1}{d}\Sigma_i^d w_i \quad , i, k = 1, 2, \dots d \tag{3}$$

$$d_t = x_t - s_t \tag{4}$$

$$\text{Then re-estimate the means using the de-seasonalized data} \tag{5}$$

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

$$\hat{Y_t} = x_t - \hat{m_t} - s_t \tag{7}$$

In our case, because we assume $\mathbf{E}[Y_t] = 0$, which due to our non-normality in the residuals as seen above in Task 5 is in fact incorrectly assumed - we will be producing a biased forecasting result. Nevertheless, suppose $\mathbf{E}[Y_t] = 0$ is true, then we may see from Equation 7 that in order to forecast x_t we simply need to plugin the values for $\hat{m_t}$ and s_t, mathematically speaking

$$\hat{x}_t^{Forcast} = \hat{m_t} + s_t$$

Recalling the results from Task 4, our s_t values are

Months	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov	Dec
s_t	-0.34	-0.33	-0.41	-0.18	-0.14	-0.17	-0.25	-0.31	-0.39	-0.43	-0.45	-0.51

The results are presented in the from of Figure 11. The reader may instantly spot both plots in-curing different shapes and trends. However, reviewing the Figure carefully, one may see the magnitude of error that the forecast achieves compared to the real data. The Forecasted line varies around $(0.4, 1.8)$ whilst the real data varies from $(-5.5, -1.5)$ on the natural logarithimc scale of the opening prices of Bitcoin in USD.

General Conclusion

One may thus conclude that such a linear time series model is insufficient or poorly specified for using on this kind of data set. As we saw under Task 5, the stationary assumption was violated as well as the normality assumption. In addition, intuitively the assumption of the model that $s_t = s_{t+d}$ may be too strong - 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 decomposition being unsuitable to forecast accurately monthly prices of Bitcoin.

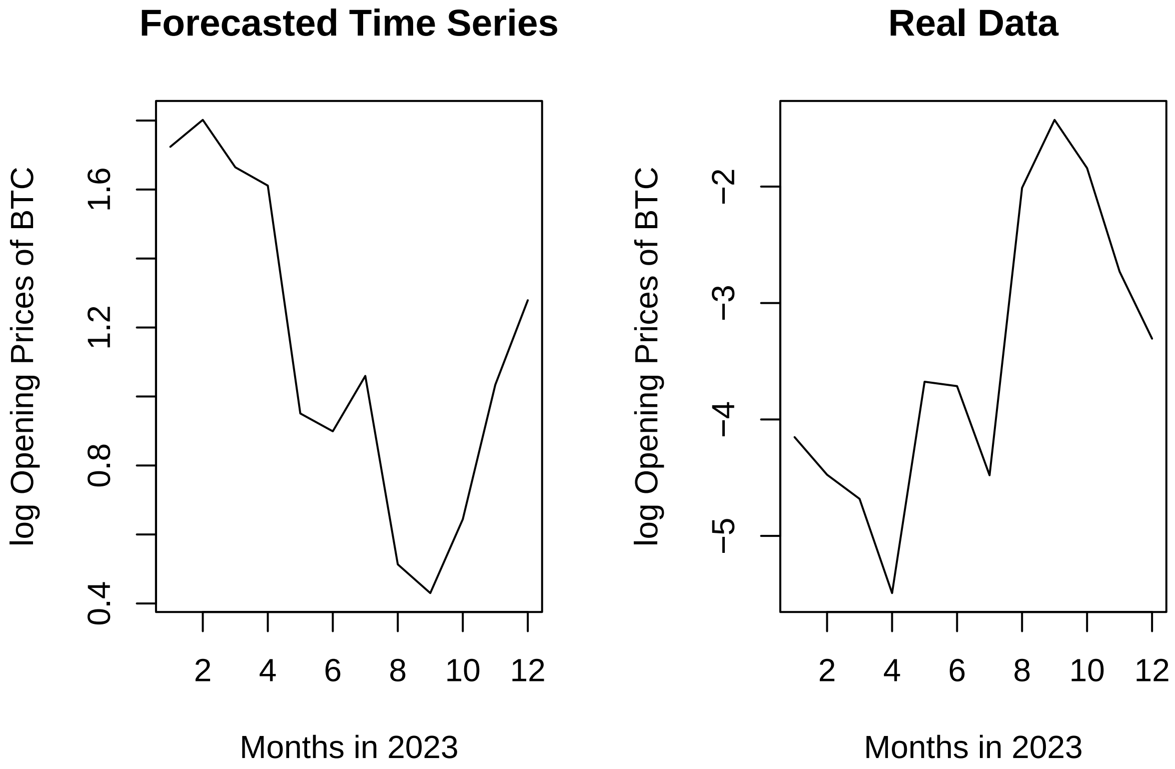

Figure 11: 12 Month Forecast of Monthly BTC/USD Prices in 2023