

Future stock trends- Analysis and Forecast

5th April, 2023

1 Abstract

The data set I plan to use for my project displays date, open, high, low, close, volume, and name of JPM stock throughout 5 years. The main objective of my project is to perform time series forecasting of JPM stock data by using the last 5 years of stock data to make informed estimates that will provide predictive direction of the future stock trends for JPM stock. This is important because it would allow consumers and the company to understand the future of their stock price which is relevant to a multitude of financial decisions that will be made.

For analysis and forecast, the data has been divided into two sections; namely 'training data' and 'test data'. This helps the model to make accurate predictions. After the analysis, the creation of the time series plot shows an obvious trend in which there is a seasonal component and the variance varied. In the next step, the model has been made approximately normal. The Arima model has been selected as the best model for forecast of future closing stock trends.

2 Introduction

Stock price, opening and closing rates fluctuates throughout the year. Buyers and sellers of financial assets such as bonds, currencies, and equities trade in the stock market. Shares are one of the most popular assets. They are units of ownership in a company that investors purchase in order to receive dividends or profit from future price increases. Companies can also raise capital by selling their shares.

A company cannot trade on the stock exchange unless it has a certain amount of capital, shareholders, and other requirements. Brokers are essential players in the stock market. They are qualified professionals who connect buyers and sellers and execute trades for a commission. Companies and individuals interested in investing in the stock market should use a broker. The price of shares changes based on supply and demand and in that aspect it's similar to any other product. In addition, there are several other factors such as information, the economy, a company's financial health and some external events such as the impression about the company's performance etc. So, my analysis is basically based on showing the futuristic trends of the stock closing price so that buyers and sellers can look at those trends and decide whether to sell or keep the shares. After running the time series analysis, the model with the lowest AIC has been selection and the predictive analysis has been run. The forecasted points that lie within the 95% confidence interval have been carefully examined. The results depict that the model is accurate enough to predict future stock closing price values.

3 Time series analysis:

Time series is a series of data points in which each data point is associated with a timestamp. The analysis has been carried out in the following sections.

3.1 Original dataset plots

I started by plotting a time series plot with the original data. However, I have modeled the data by only having closing prices and also by excluding the dates. The reason of cutting off those dates is to plot the daily JPM closing stock price with ease and smoothness. Including the dates would skew the results, and would cause inconsistency with the rest of the data.

First of all, the data has been imported into R studio and the table below shows the overview of the imported dataset. The dataset has 1259 observations and 7 columns.

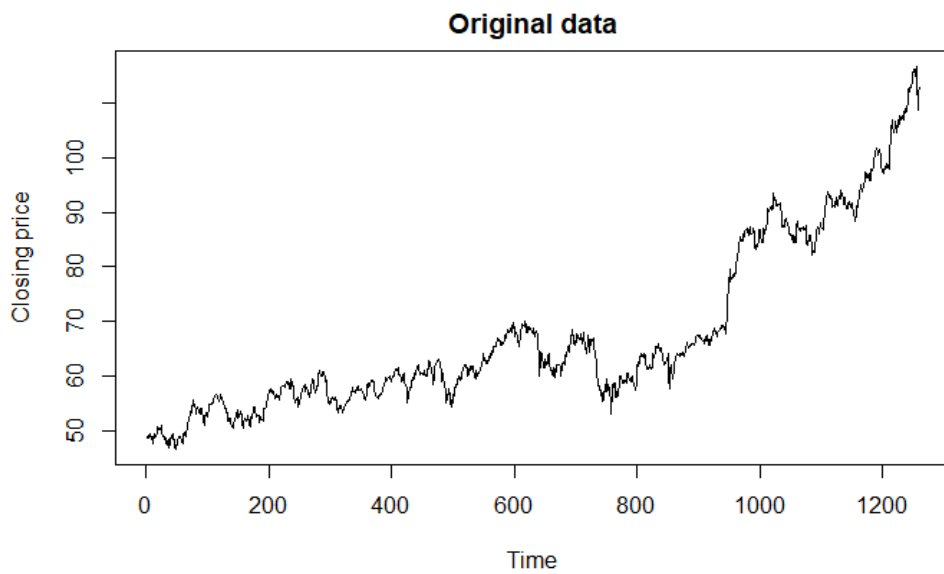
	date	open	high	low	close	volume	Name
1	2013-02-08	48.3300	48.6900	48.2600	48.63	15217458	JPM
2	2013-02-11	48.5100	48.9000	48.3800	48.66	13934328	JPM
3	2013-02-12	48.8000	49.3100	48.6000	49.14	16387366	JPM
4	2013-02-13	49.3500	49.4500	48.5000	48.68	21635732	JPM
5	2013-02-14	48.4000	49.2899	48.3700	49.22	18017634	JPM
6	2013-02-15	49.3500	49.4600	48.5700	48.88	20015681	JPM
7	2013-02-19	49.1200	49.6800	49.1200	49.45	20445200	JPM
8	2013-02-20	49.3500	49.5400	48.3600	48.61	24787660	JPM
9	2013-02-21	48.4100	48.4500	47.8300	48.25	24436578	JPM
10	2013-02-22	48.6300	48.9100	48.4100	48.91	23583917	JPM
11	2013-02-25	49.1000	49.2000	47.6500	47.70	32749013	JPM

Showing 1 to 12 of 1,259 entries, 7 total columns

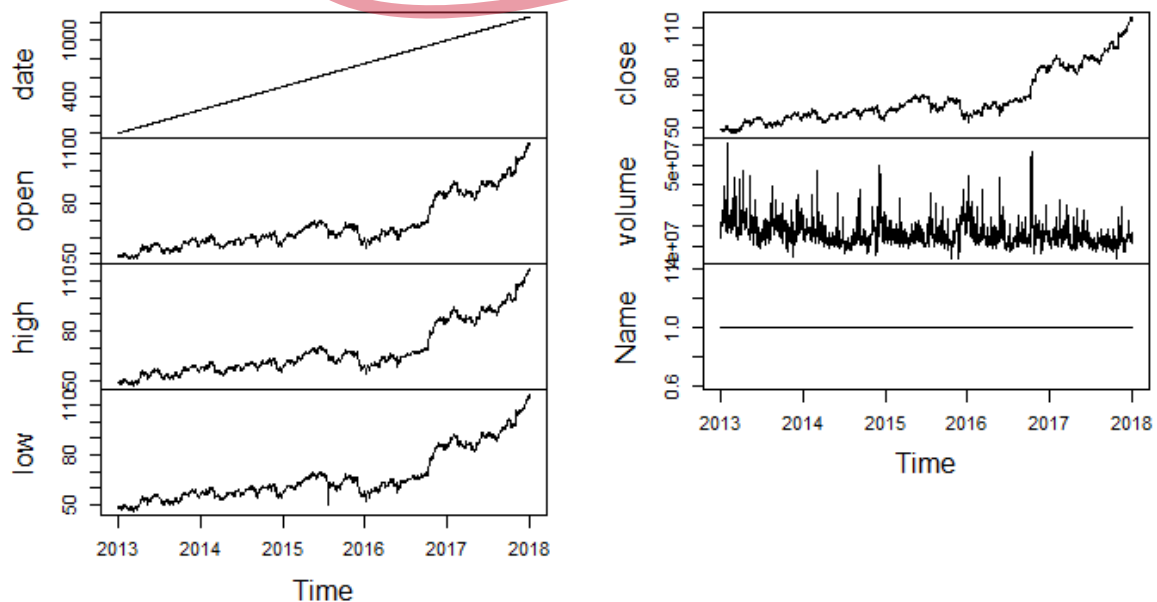
The null values have been removed by using na.omit function in the pre-processing step and the dataset has been divided into training data and test data. The training dataset has 1247 observations while the test dataset has 12 observations.

The time series has been plotted with a frequency of 251 to generate 1261 observations.

The original dataset and the time series plots are shown below:

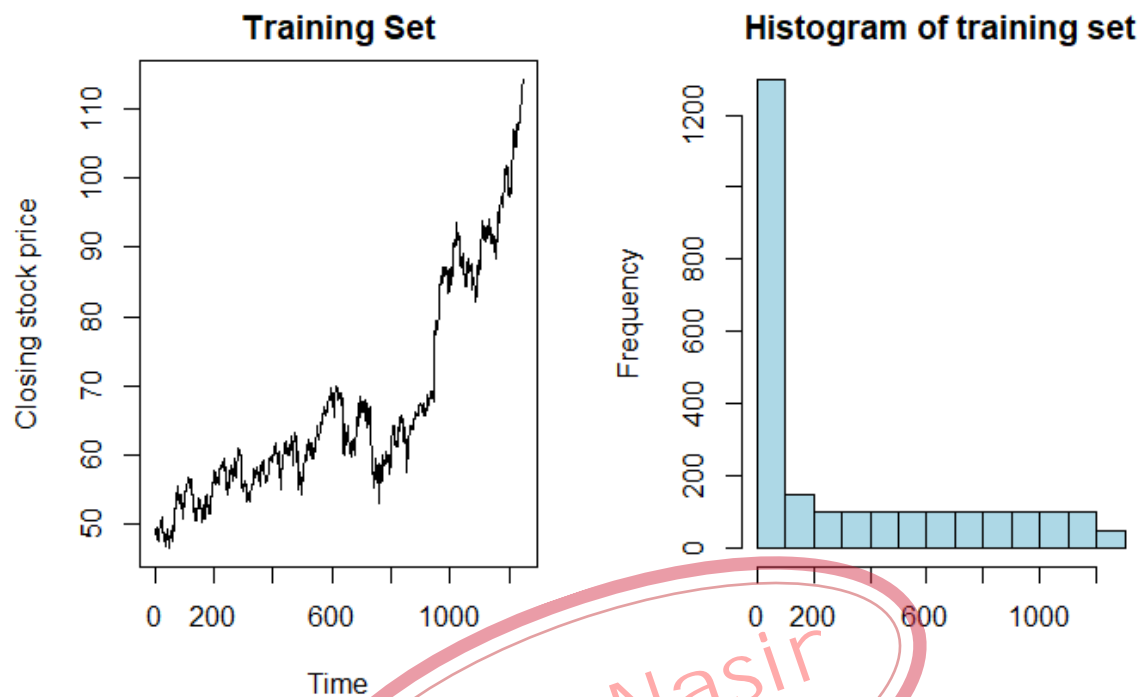


I started by plotting a time series plot with the original data. However, I have modeled the data by only having stock closing price and also by excluding the dates between the years 2013 and 2018. Instead of these years, the time series has been created of 1261 observations. rest of the data.

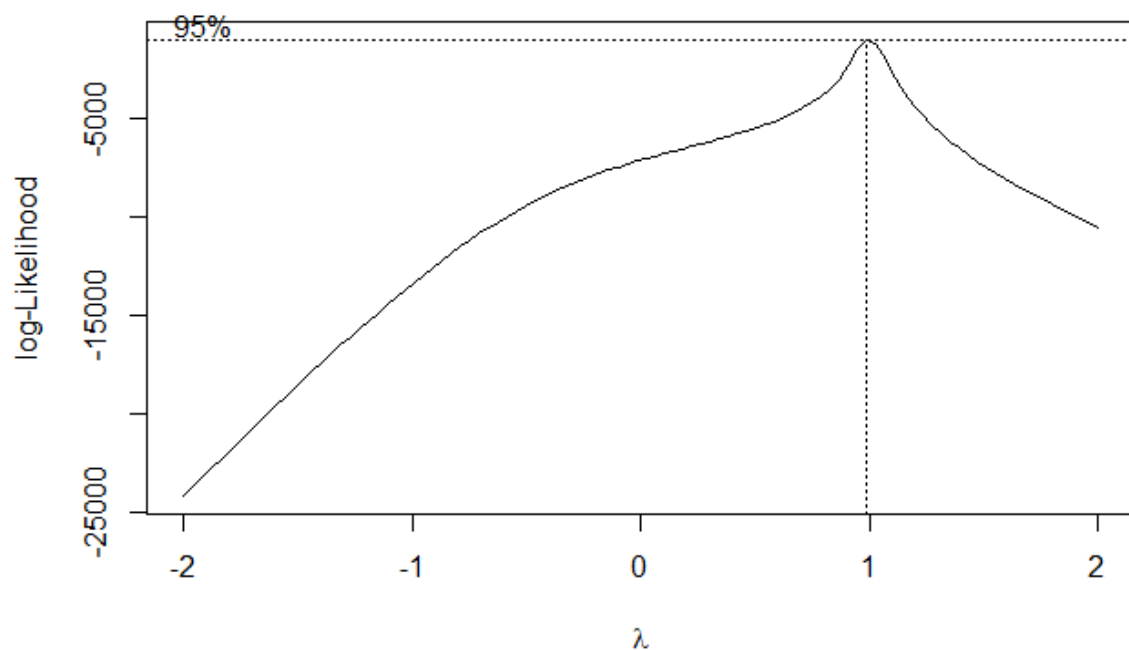


3.2 Plotting the training dataset

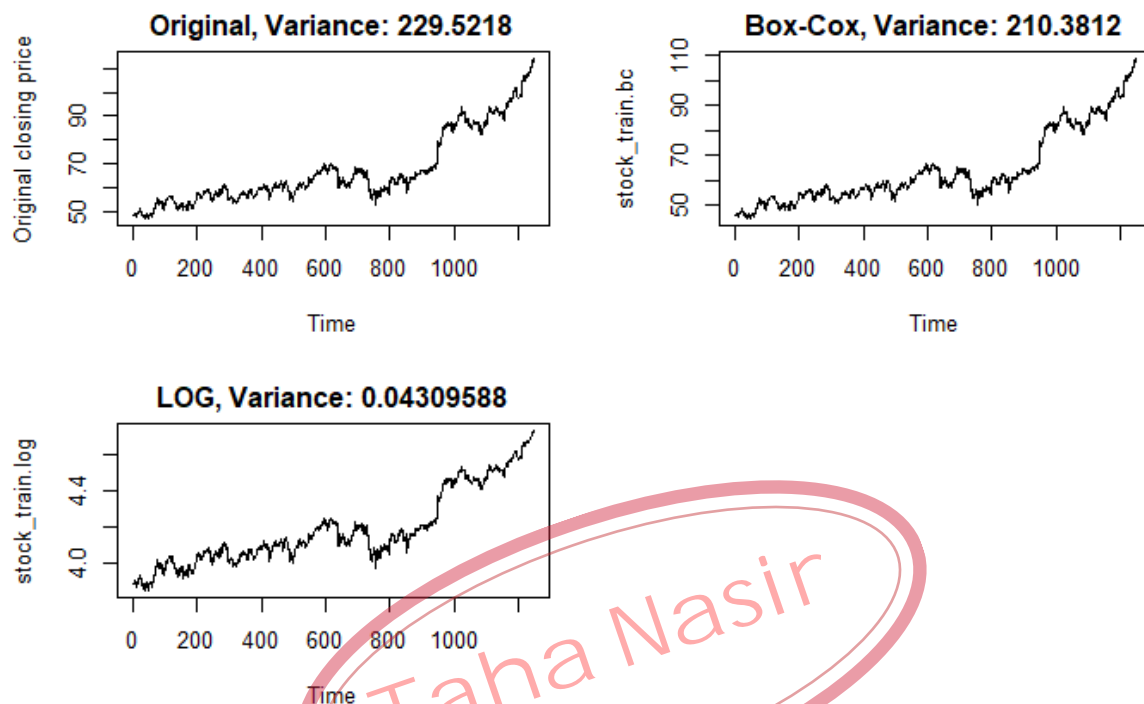
In order to make sure our model was relatively accurate; the data has been sectioned into training dataset and test dataset. I removed the last 12 observations, in order to compare the forecasted values with the original data values later in the analysis.



After observing the time series plot, it can be said that the trend is generally increasing and decreasing but the overall trend is increasing especially at the end of the interval, the value of closing stock price reaches the highest value. There is also a seasonal component. The histogram is right skewed and the variance is non-constant. Based on these observations, I would say it would be best to perform a Box-Cox transformation.



In the next step, the variance of the original stock_train dataset, the boxcox stock_train dataset and the log transformed stock_train dataset has been compared after the transformation process.



> var(stock_train[,2]) has the highest variance

[1] Variance= 229.5218

> var(stock_train.bc)

[2] Variance= 210.3812

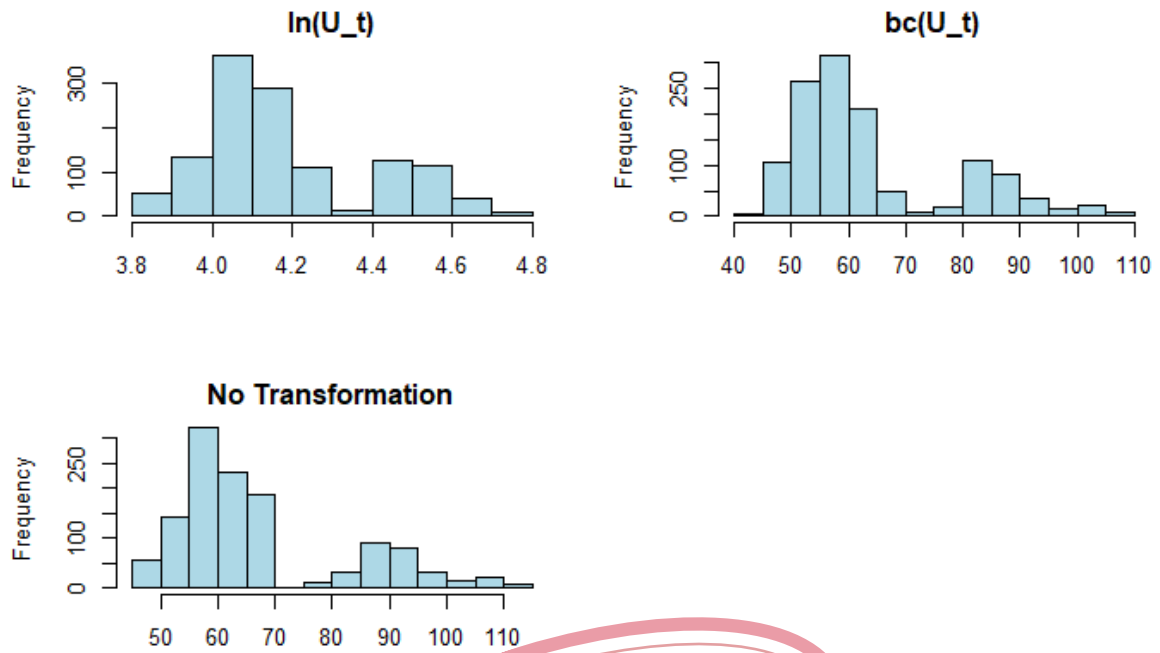
> var(stock_train.log) has the lowest variance. So, we will choose the logarithmically transformed dataset for further analysis.

[3] Variance= 0.04309588

Looking at the 95% confidence interval for the true lambda and by looking at all the plots, we could see that either no transformation or logarithmic transformation is suggested. Looking at the plots we could see that the LOG plot actually didn't have as sharp dips and tips. This was most prominent around time 200, 400 and 600. Also, when looking at the variances, we could see the Box-Cox transformation slightly decreases the variance whereas the log transform drastically decreases the variance.

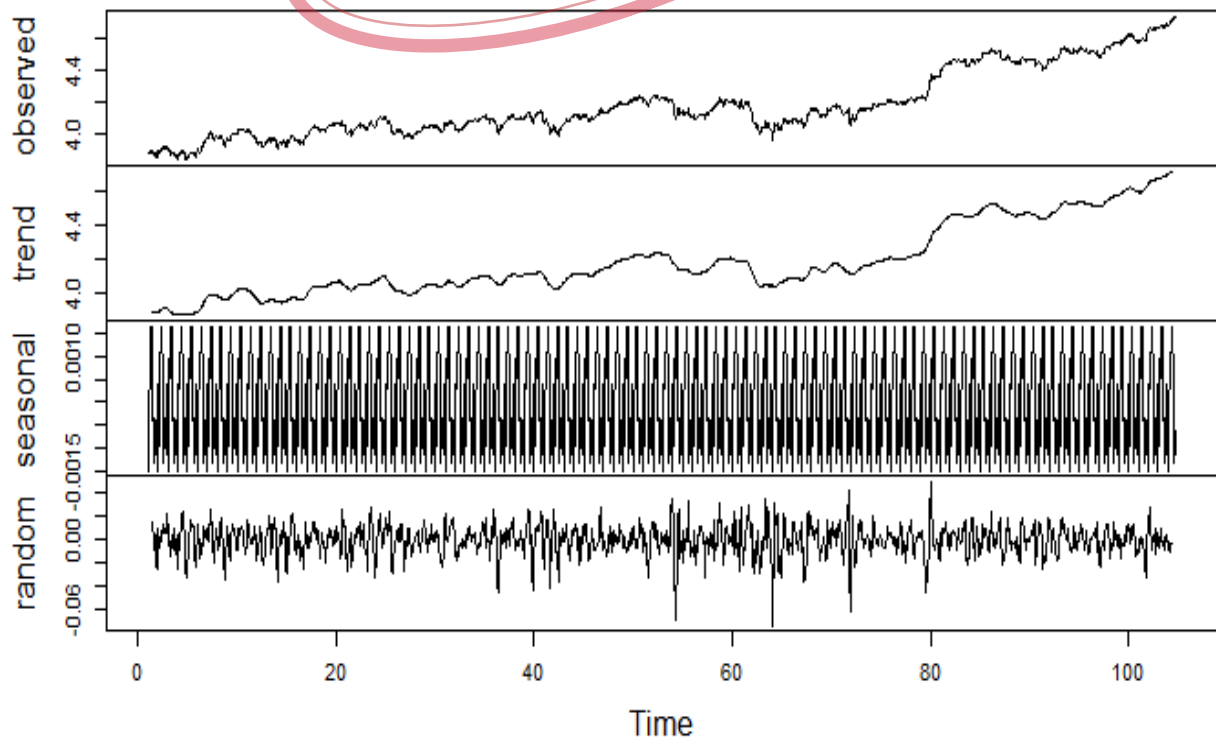
The histograms showing the trends of the various transformations are shown below:

The histogram that least resembles the normal distribution is that of the logarithmic transformation. Hence, based on the variances, histograms and the plots, the logarithmic transformation has been selected for further analysis.



3.3 Decomposition of log transformed training dataset

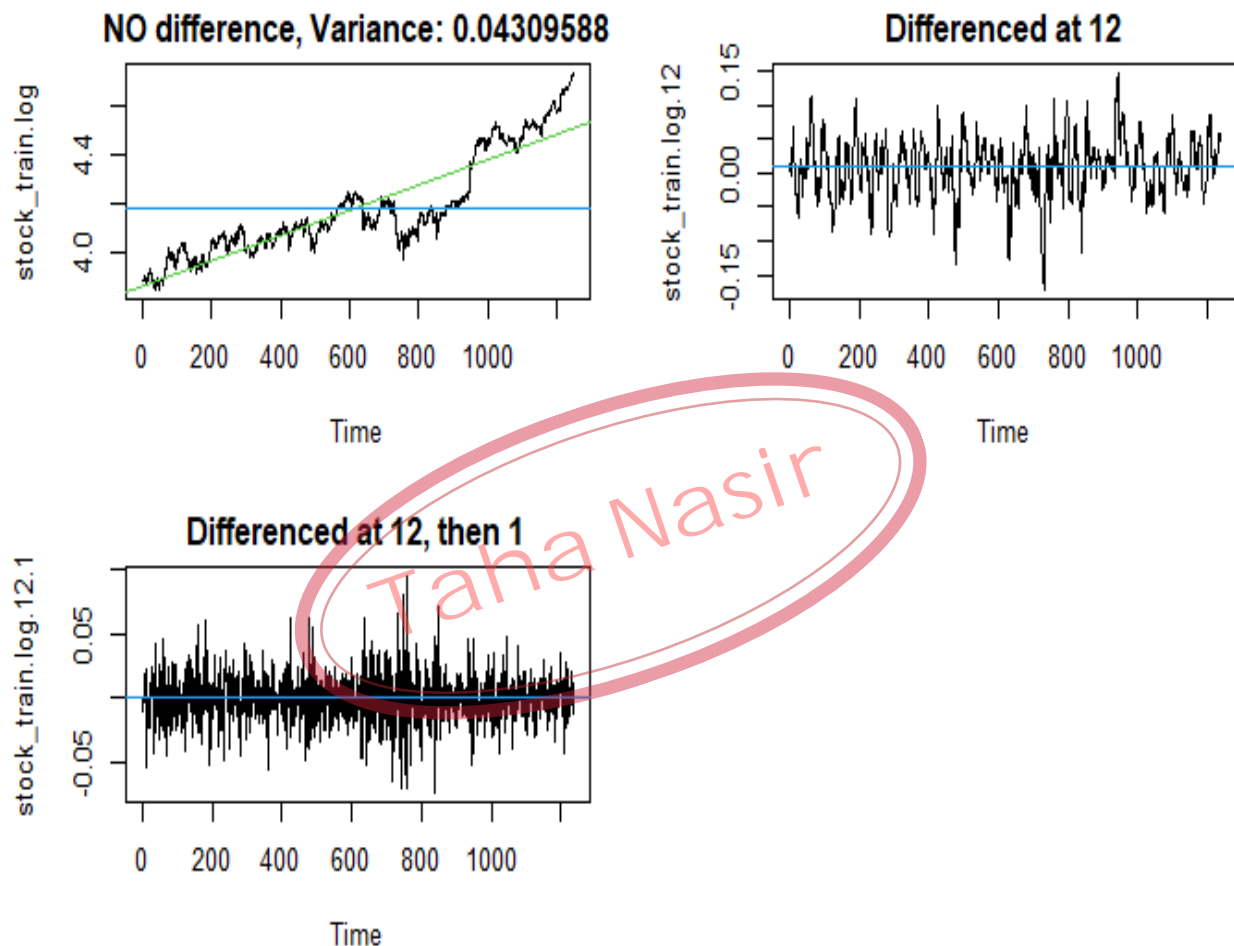
Decomposition of additive time series



In the aforementioned graph, there is clear trend and a jumbled up seasonal component. To correct this, differencing has been performed which will be explained in the following steps.

3.4 Differencing

The differencing has been done two times. The plot of no difference and differenced at 12 and at 12, then 1 are shown below:

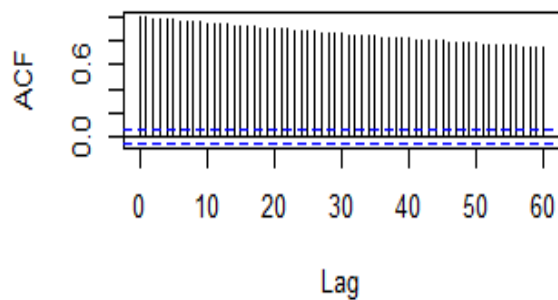


The green line is the fitted line whereas the blue line shows the mean value. By looking at the three plots mentioned above, it can be clearly seen that the plot 'Differenced at lag 12, then 1' had the least common or apparent trend and the least apparent seasonality. The variance of the plot differenced at 12, then 1 is lower (approx. 0.0003) than the variance of the plot differenced at 12 (0.00177) but the visual depiction of the data is clearer for the plot differenced at 12. The mean of the plot 'Differenced at 12, then 1' is closest to 0 which is 0.00003495.

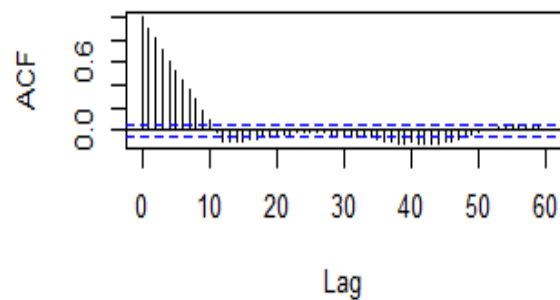
3.5 Plots of ACF differences

The ACF plots of no difference, differenced at 12 and differenced at 12, then 1 are shown below:

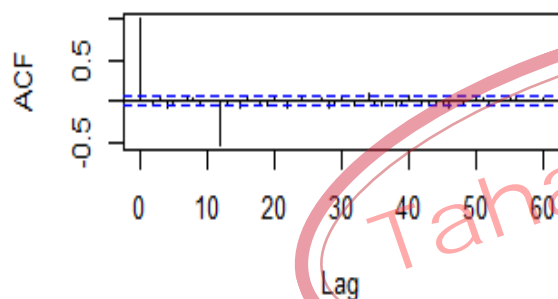
ACF of LOG



ACF of LOG difference at 12

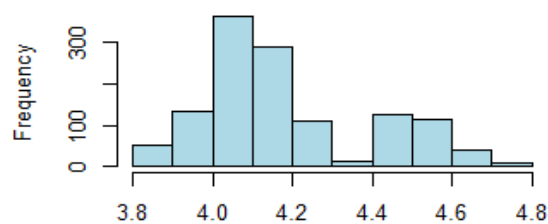


ACF of LOG difference at 12 then 1

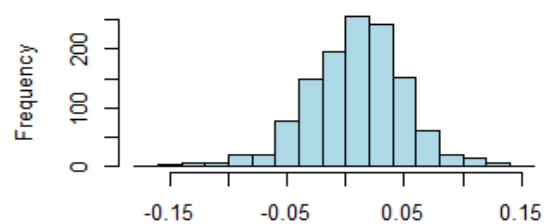


Comparing the ACF's of the three graphs, the non-differenced and differenced at 12 still contains some seasonality, and the ACF has decayed slowly which indicated non-stationarity. The ACF of the log plot differenced at 12, then 1 had no apparent seasonality and the decay corresponded to a stationary process. The histograms of the three processes are as follows:

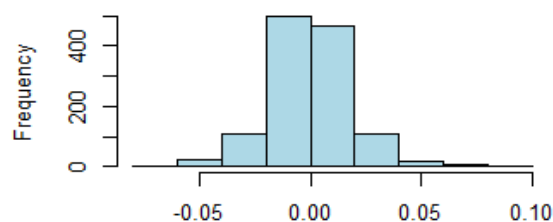
$\ln(U_t)$



$\ln(U_t)$, differenced at 12



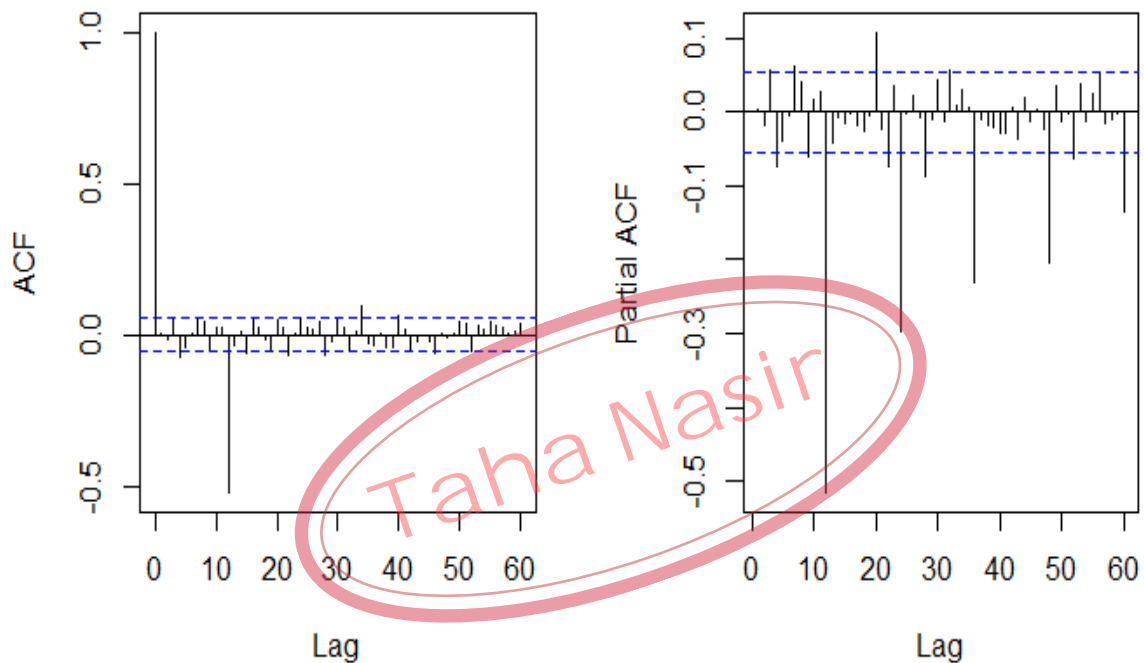
$\ln(U_t)$, differenced at 12 then 1



After analyzing the histograms, it is clear that the histogram of log differenced at 12, then 1 has the most normal like distribution so it has been chosen for further analysis.

3.6 ACF and PACF of log differenced at 12, then 1

ACF of LOG difference at 12 then 1 PACF of LOG difference at 12 then 1



Looking at the ACF and PACF plots, I have made the following deductions:

ACF outside confidence intervals: 1 and 12

PACF chosen outside confidence intervals: 2 and 12

SARIMA for log: $s=12$, $D=1$, $d=1$, $Q=0$ or 1 , $q=0$, $P=0$ or 1 , $p=0$.

3.7 Trying models:

	p <int>	q <int>	P <int>	Q <int>	AICc <dbl>
1	0	0	0	0	-6374.694
2	1	0	0	0	-6372.712
3	0	1	0	0	-6372.712
4	1	1	0	0	-6370.702
5	0	0	1	0	-6771.910
6	1	0	1	0	-6769.902
7	0	1	1	0	-6769.902
8	1	1	1	0	-6767.889
9	0	0	0	1	-7200.961
10	1	0	0	1	-7199.006

1-10 of 16 rows

Previous **1** 2 Next

	p <int>	q <int>	P <int>	Q <int>	AICc <dbl>
11	0	1	0	1	-7199.008
12	1	1	0	1	-7196.994
13	0	0	1	1	-7199.601
14	1	0	1	1	-7197.647
15	0	1	1	1	-7197.649
16	1	1	1	1	-7195.635

The best model with the lowest AICc is when $p=0$, $q=0$, $P=0$ and $Q=1$. The AICc in this case is -7200.961 and the second lowest is -7199.601 when $p=0$, $q=0$, $P=1$ and $Q=1$. The second one will be taken as an alternative model.

3.8 Models

The first model is:

```
fit4.1
##
## Call:
## arima(x = stock_train.log, order = c(0, 1, 0), seasonal = list(order =
##      1, 1), period = 12), method = "ML")
##
## Coefficients:
##          sma1
##       -1.0000
## s.e.    0.0161
##
## sigma^2 estimated as 0.000163:  log likelihood = 3602.49,  aic = -7200.
## 97
```

The second model is:

```
fit4
```

```
##
## Call:
## arima(x = stock_train.log, order = c(0, 1, 0), seasonal = list(order =
##      1, 1), period = 12), method = "ML")
##
## Coefficients:
##          sar1      sma1
##      -0.0232  -1.0000
## s.e.    0.0287   0.0182
##
## sigma^2 estimated as 0.0001628:  log likelihood = 3602.81,  aic = -7199
## .62
```

3.9 Diagnostic checking 1st model:

```
## Shapiro-Wilk normality test
##
## data:  res
## W = 0.96574, p-value < 2.2e-16

# p-value is less than 0.05 so it does not pass the shapiro.test
Box.test(res, lag = 9, type = c("Box-Pierce"), fitdf = 3)

##
## Box-Pierce test
##
## data:  res
## X-squared = 10.845, df = 6, p-value = 0.0933

Box.test(res, lag = 9, type = c("Ljung-Box"), fitdf = 3)

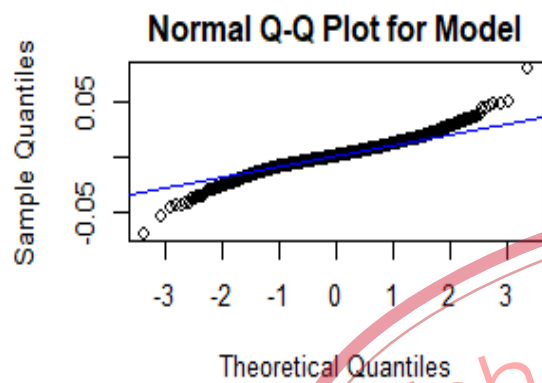
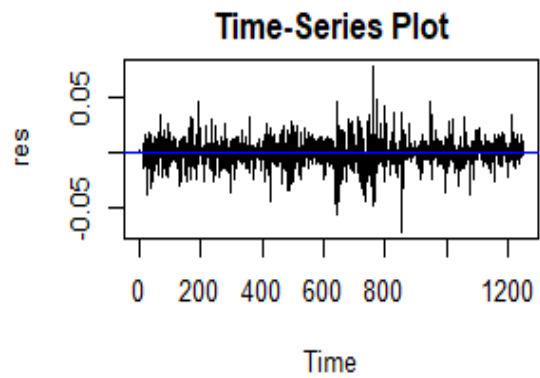
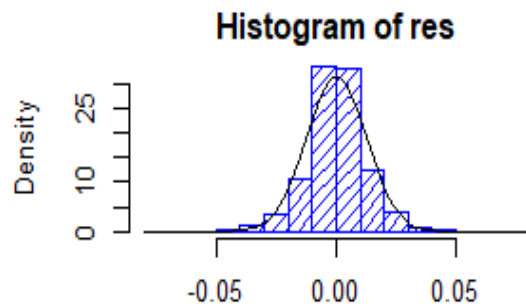
##
## Box-Ljung test
##
## data:  res
## X-squared = 10.916, df = 6, p-value = 0.091

Box.test(res^2, lag = 9, type = c("Ljung-Box"), fitdf = 0)

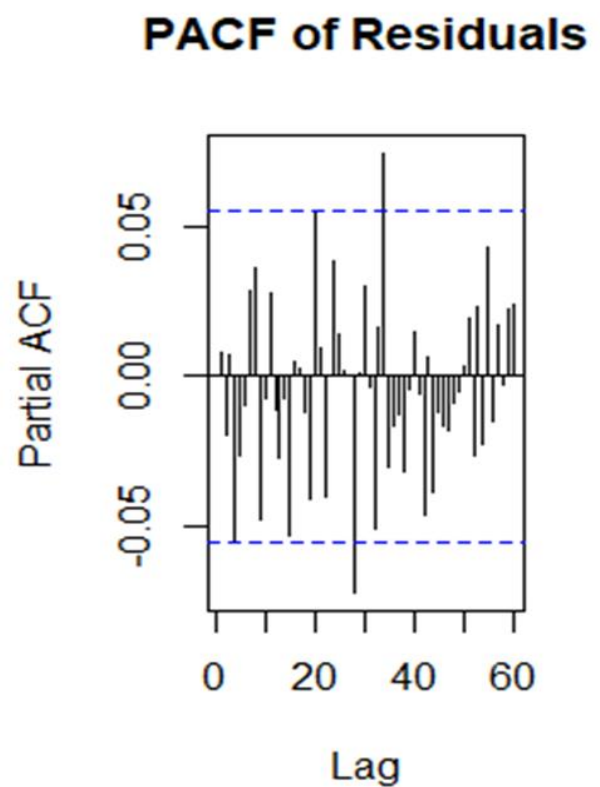
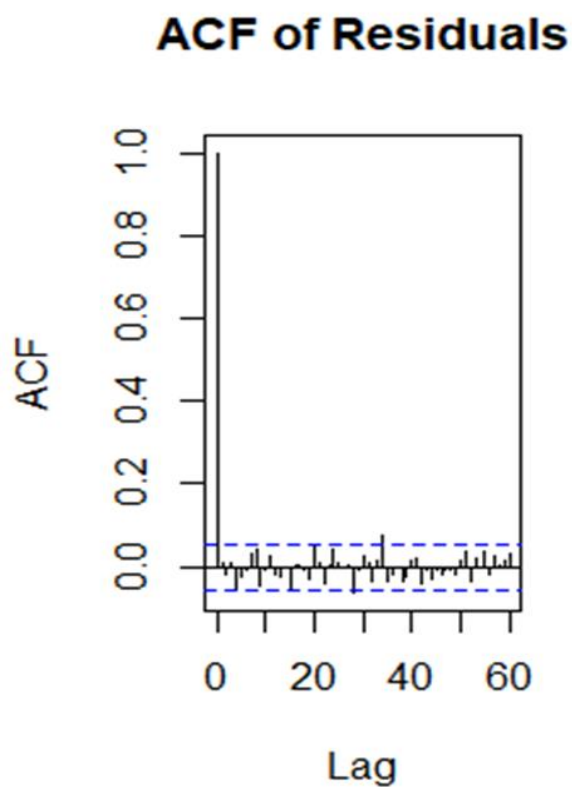
##
## Box-Ljung test
##
## data:  res^2
## X-squared = 98.369, df = 9, p-value < 2.2e-16
```

Passes the Box_Pierce and Ljung-Box tests, since p-values of these two are larger than 0.05.

The histogram of residual, time series plot and normal Q-Q plot for model 1 has been plotted below:



The new ACF and PACF are:



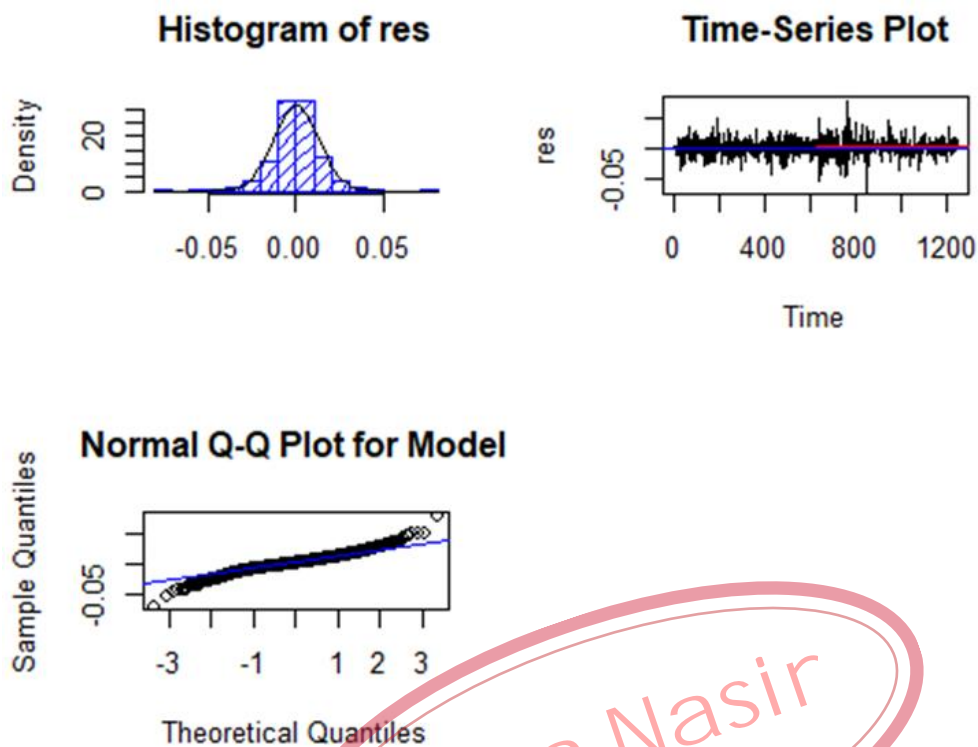
Call:

```
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))  
##  
##  
## Order selected 0 sigma^2 estimated as 0.0001616
```

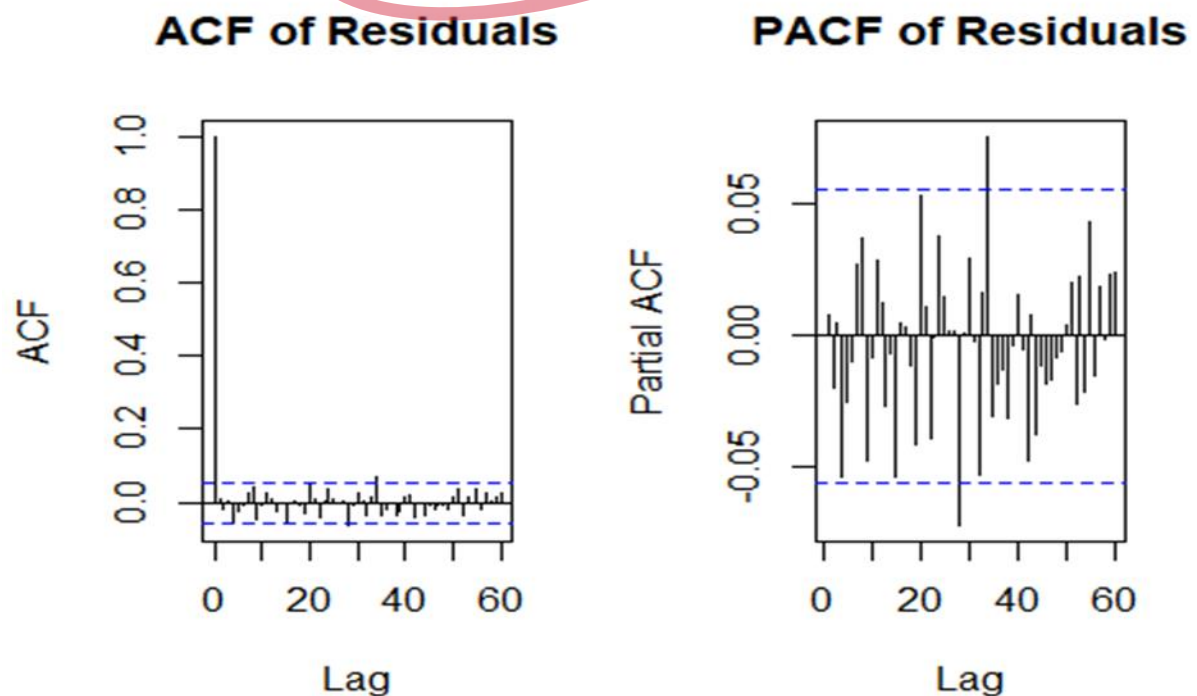
3.10 Diagnostic checking of Model 2:

```
## Shapiro-Wilk normality test  
##  
## data: res  
## W = 0.96591, p-value < 2.2e-16  
  
# p-value is less than 0.05 so it does not pass the Shapiro test  
Box.test(res, lag = 9, type = c("Box-Pierce"), fitdf = 2)  
  
##  
## Box-Pierce test  
##  
## data: res  
## X-squared = 10.538, df = 7, p-value = 0.1601  
  
Box.test(res, lag = 9, type = c("Ljung-Box"), fitdf = 2)  
  
##  
## Box-Ljung test  
##  
## data: res  
## X-squared = 10.608, df = 7, p-value = 0.1567  
  
Box.test(res^2, lag = 9, type = c("Ljung-Box"), fitdf = 0)  
  
##  
## Box-Ljung test  
##  
## data: res^2  
## X-squared = 98.234, df = 9, p-value < 2.2e-16
```

It passes the first and second Box test i.e., Box-Pierce and Ljung-Box test because the p-values for these tests are larger than 0.05. The histogram of residual, time series plot and normal Q-Q plot for model 1 has been plotted below:



The ACF and PACF plots are:



```
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-
walker"))
```

##

Order selected 0 sigma^2 estimated as 0.0001614

When performing the different types of tests on the two models, we saw that the models passed two of the Box Cox tests because it's p-values were greater than the significance level of 0.05. Moreover, both the models failed the Shapiro test and residual squared Ljung-Box test. Looking at the time series plot, we saw that there were no trends, no change in variances nor seasonal components. After carefully analyzing the Q-Q plots and histograms, we could see that they were approximately normally distributed. In the final ACF and PACF there were minor lags outside the confidence intervals. Based on all the evidence, I have decided to go with the model with the least AICc which is the first model, although the second model could have also been chosen since both the models almost resemble the normal distribution.

Another benefit of the first model is that it had one less coefficient as shown in the analysis of the first model. Hence the first model is best for forecast.

3.11 Forecast

According to the forecast function, the forecasted dataset is as follows:

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1248	4.741280	4.724839	4.757721	4.716135	4.766424
1249	4.740064	4.716813	4.763316	4.704505	4.775624
1250	4.741786	4.713310	4.770263	4.698235	4.785337
1251	4.743925	4.711043	4.776806	4.693637	4.794213
1252	4.744116	4.707353	4.780879	4.687893	4.800339
1253	4.746000	4.705729	4.786272	4.684411	4.807590
1254	4.746738	4.703240	4.790236	4.680214	4.813262
1255	4.746158	4.699657	4.792659	4.675041	4.817275
1256	4.745083	4.695762	4.794405	4.669652	4.820515
1257	4.746689	4.694699	4.798678	4.667177	4.826200

1-10 of 24 rows

Previous ☒ 1 ☐ 2 ☐ 3 Next

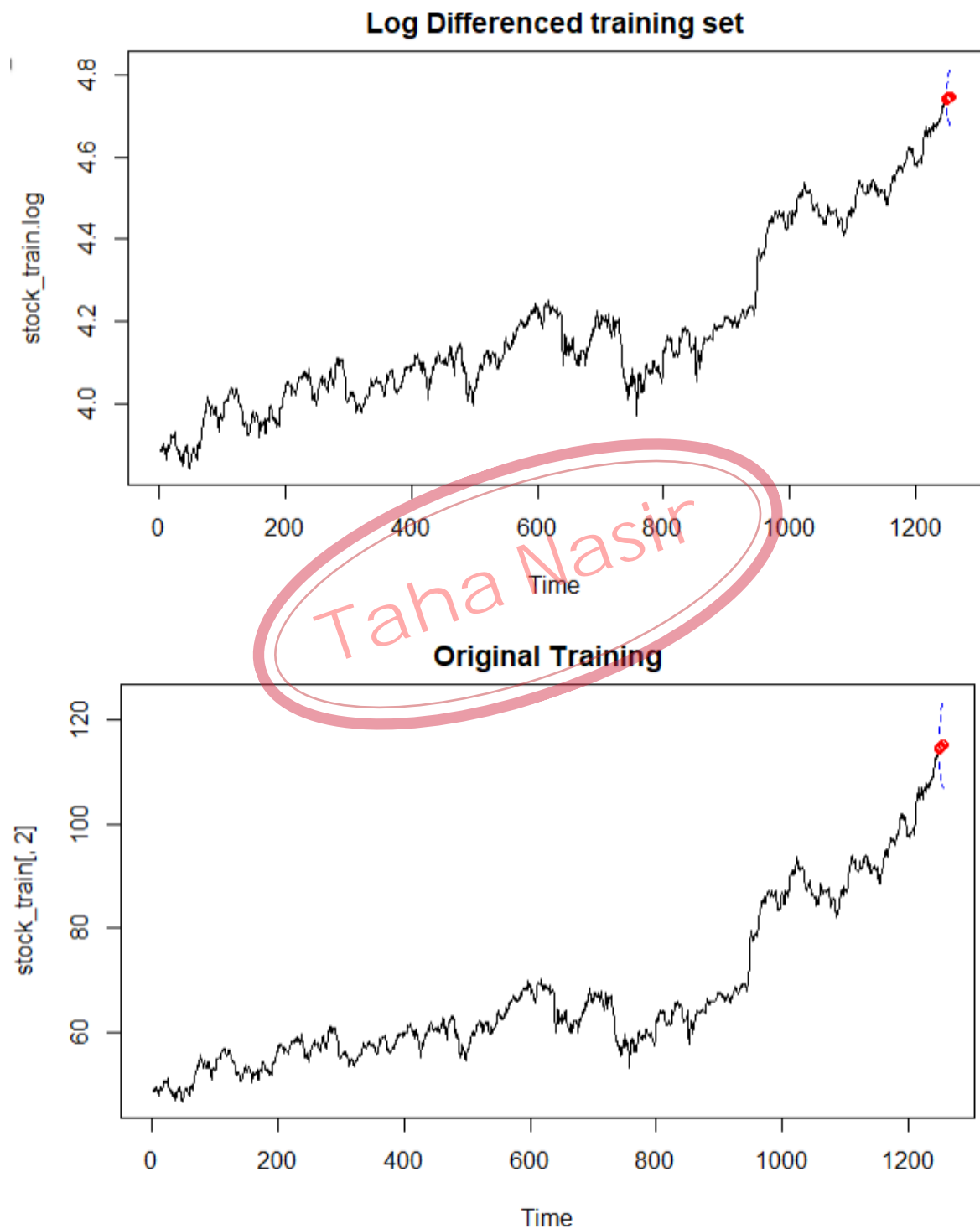
	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1258	4.746808	4.692281	4.801335	4.663416	4.830200
1259	4.747321	4.690370	4.804273	4.660221	4.834422
1260	4.749512	4.690191	4.808833	4.658788	4.840236
1261	4.748297	4.686697	4.809897	4.654088	4.842505
1262	4.750019	4.686222	4.813815	4.652451	4.847586
1263	4.752157	4.686238	4.818077	4.651342	4.852973
1264	4.752348	4.684372	4.820325	4.648387	4.856310
1265	4.754233	4.684260	4.824206	4.647218	4.861248
1266	4.754970	4.683056	4.826885	4.644987	4.864954
1267	4.754390	4.680586	4.828195	4.641516	4.867265

11-20 of 24 rows

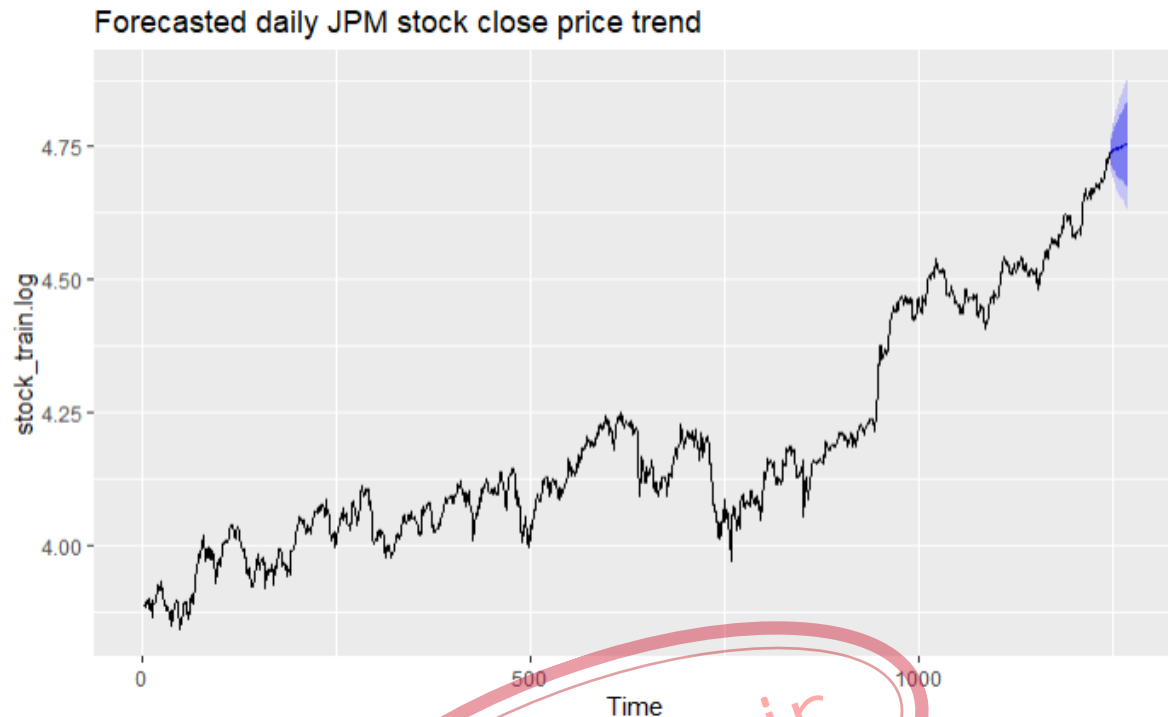
Previous ☐ 1 ☒ 2 ☐ 3 Next

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
1268	4.753316	4.677668	4.828964	4.637623	4.869009
1269	4.754921	4.677474	4.832368	4.636477	4.873366
1270	4.755041	4.675836	4.834246	4.633907	4.876174
1271	4.755554	4.674629	4.836479	4.631790	4.879318

The 12 points have been forecasted for the daily closing price of stocks. The plots of the tlog differenced and original training datasets are as follows:



The plot of original data points along with the forecasted data is as follows:



4 Conclusion

To conclude, the forecasted daily closing stock price has been evaluated. The goal has been achieved by using the first model instead of the second one. In the evaluation process, firstly, the dataset was divided into two parts; training data and test data. Then it was log transformed and then differenced at 12 and differenced at 12, then 1. After that, diagnostic checking has been done and after confirmation, the model has been used to predict the future closing price values which are close to the original data. With this analysis the customers can make informed decisions about the stocks they are interested in.

5 References

Data source: [Kaggle.com](https://www.kaggle.com)

Appendix:

```
#reading the data
library(tidyverse)
library(dplyr)
stock.csv <- read.csv(file = "JPM_data.csv")

#Time series
stock=ts(stock.csv,c(2013,1),c(2018,1),252)
ts.plot(stock.csv$close,main = "Original data", ylab = "Closing price")
plot.ts(stock)
nt=length(stock)
abline(h = mean(stock), col="4")
legend("bottomright", legend = c("Fitted Line", "Mean"), pch = rep(15, 4), col = 3:4)

#Training and test dataset
stock_train <- stock[c(1:1247),c(1,5)]
stock_train.csv<-stock[c(1:1247)]
stock_test <- stock[c(1247:1261),c(1,5)]

#Plotting the training dataset
par(mfrow=c(1,2))
ts.plot(stock_train[,2], main = "Training Set", ylab = "Closing stock price")
#fit <- lm(stock_train ~ as.numeric(1:length(stock_train[,2])));abline(fit, col="3")
#legend("bottomright", legend = c("Fitted Line"),pch , col = 3)
hist(stock_train, col = "light blue", xlab = "", main = "Histogram of training set")

## Transformations:
library(MASS)
bcTransform <- boxcox(stock_train~ as.numeric(1:length(stock_train.csv)))
lambda=bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
stock_train.bc = (1/lambda)*(stock_train[,2]^lambda-1)
stock_train.log <- log(stock_train[,2])
par(mfrow=c(2,2))
plot.ts(stock_train[,2],ylab="Original closing price", main = "Original, Variance: 229.5218")
plot.ts(stock_train.bc, main = "Box-Cox, Variance: 210.3812")
plot.ts(stock_train.log, main = "LOG, Variance: 0.04309588")
var(stock_train[,2])# highest variance
```

```

var(stock_train.bc)

var(stock_train.log) # lowest variance

par(mfrow=c(2,2))

hist(stock_train.log, col="light blue", xlab="", main="ln(U_t)") # Has the lowest variance

hist(stock_train.bc, col="light blue", xlab="", main="bc(U_t)") #looks the most symmetric and normally distri

hist(stock_train[,2], col = "light blue", xlab = "", main = "No Transformation")

## Decomposition of LOG transformed training set:

y <- ts(as.ts(stock_train.log), frequency = 12)

decomposition <- decompose(y)

plot(decomposition)

## Differencing:

par(mfrow=c(2,2))

plot.ts(stock_train.log, main = "NO difference, Variance: 0.04309588") #0.04309588
fit1 <- lm(stock_train.log ~ as.numeric(1:length(stock_train.log)));abline(fit1, col=3)
abline(h=mean(stock_train.log), col=4)
stock_train.log.12 <- diff(stock_train.log, lag =12)
plot.ts(stock_train.log.12, main = "Differenced at 12")
fit2 <- lm(stock_train.log.12 ~ as.numeric(1:length(stock_train.log.12)));abline(fit1, col=3)
abline(h=mean(stock_train.log.12), col=4)
stock_train.log.12.1 <- diff(stock_train.log.12, lag =1)
plot.ts(stock_train.log.12.1, main = "Differenced at 12, then 1")
fit3 <- lm(stock_train.log.12.1 ~ as.numeric(1:length(stock_train.log.12.1)));abline(fit1, col=3)
abline(h=mean(stock_train.log.12.1), col=4)

var(stock_train.log)

mean(stock_train.log)

var(stock_train.log.12)

mean(stock_train.log.12) #-0.003156617

var(stock_train.log.12.1) #variance went higher, 0.04487819 went higher

mean(stock_train.log.12.1) #0.008926943 went lower

#ACF differencing plots

par(mfrow=c(2,2))

acf(stock_train.log, lag.max = 60, main = "ACF of LOG")

acf(stock_train.log.12, lag.max = 60, main = "ACF of LOG difference at 12")

```

```

acf(stock_train.log.12.1, lag.max = 60, main = "ACF of LOG difference at 12 then 1") #ACF decays corresponding
## Comparing the histograms at different differences:
par(mfrow=c(2,2))
hist(stock_train.log, col="light blue", xlab="", main="ln(U_t)")
hist(stock_train.log.12, col="light blue", xlab="", main="ln(U_t), differenced at 12")
hist(stock_train.log.12.1, col="light blue", xlab="", main="ln(U_t), differenced at 12 then 1")
## ACF and PACF of log then differenced at 12 then 1:
par(mfrow=c(1,2))
acf(stock_train.log.12.1, lag.max = 60, main = "ACF of LOG difference at 12 then 1")
#ACF outside confidence intervals: 1,12
pacf(stock_train.log.12.1, lag.max = 60, main = "PACF of LOG difference at 12 then 1")
#PACF outside the confidence intervals: 1,12?
#List of candidate models to try:
#SARIMA for log: s=12, D=1, d=1, Q=0 or 1, q=0, P=0 or 1, p=0.
## Trying models now:
df <- expand.grid(p=0:1, q=0:1, P=0:1, Q=0:1)
df <- cbind(df, AICc=NA)
# Compute AICc:
library(MASS)
library(MuMIn)
for (i in 1:nrow(df)) {
  sarima.obj <- NULL
  try(arima.obj <- arima(stock_train.log, order=c(df$p[i], 1, df$q[i]),
    seasonal=list(order=c(df$P[i], 1, df$Q[i]), period=12),
    method="ML"))
  if (!is.null(arima.obj)) { df$AICc[i] <- AICc(arima.obj) }
  # print(df[i, ])
}
#df[which.min(df$AICc), ]
df[(df$AICc),] # second lowest AIC -45.75403 when p=1, q=1, P=0, Q=1
# when p=0, q=1, P=0, Q=1 we get the lowest AIC -47.91584
final <- which.min(df$AICc)
fit4.1 <- arima(stock_train.log, order=c(0, 1, 0),

```

```

seasonal=list(order=c(0, 1, 1), period=12),
method="ML")
fit4.1
fit4 <- arima(stock_train.log, order=c(0,1,0),
seasonal=list(order=c(1,1,1), period=12),
method="ML")
fit4
## Diagnostic Checking 1st model:
res <- residuals(fit4.1)
shapiro.test(res)
# p-value is less than 0.05 so it does not pass the shapiro.test
Box.test(res, lag = 9, type = c("Box-Pierce"), fitdf = 3)
Box.test(res, lag = 9, type = c("Ljung-Box"), fitdf = 3)
Box.test(res^2, lag = 9, type = c("Ljung-Box"), fitdf = 0)
#passes the Box_Pierce and Ljung-Box tests, since p-values are larger than 0.05
par(mfrow=c(2,2))
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
# Approximately normal and almost symmetric.
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res,main = "Time-Series Plot")
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model")
qqline(res,col="blue")
# Looks approximately normal
par(mfrow=c(1,2))
acf(res, lag.max=60, main = "ACF of Residuals")
pacf(res, lag.max=60, main = "PACF of Residuals")
# All ACF and PACF are within confidence intervals and can be counted as zero.
#acf(res^2, lag.max=60, main = "ACF of Residuals^2")
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

```

#2 diagnostic checking

```
res <- residuals(fit4)
```

```
shapiro.test(res)
```

p-value is less than 0.05 so it does not pass the Shapiro test

```
Box.test(res, lag = 9, type = c("Box-Pierce"), fitdf = 2)
```

```
Box.test(res, lag = 9, type = c("Ljung-Box"), fitdf = 2)
```

```
Box.test(res^2, lag = 9, type = c("Ljung-Box"), fitdf = 0)
```

#passes the first and second Box test i.e Box-Pierce and Ljung-Box test because the p-values for these test are larger than 0.05.

```
par(mfrow=c(2,2))
```

```
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
```

Approximately normal and almost symmetric.

```
m <- mean(res)
```

```
std <- sqrt(var(res))
```

```
curve( dnorm(x,m,std), add=TRUE )
```

```
plot.ts(res,main = "Time-Series Plot")
```

```
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
```

```
abline(h=mean(res), col="blue")
```

```
qqnorm(res,main= "Normal Q-Q Plot for Model")
```

```
qqline(res,col="blue")
```

Looks approximately normal

```
par(mfrow=c(1,2))
```

```
acf(res, lag.max=60, main = "ACF of Residuals")
```

```
pacf(res, lag.max=60, main = "PACF of Residuals")
```

All ACF and PACF are within confidence intervals and can be counted as zero.

```
#acf(res^2, lag.max=60, main = "ACF of Residuals^2")
```

```
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##Forecast
```

```
library(forecast)
```

```
library(dplyr)
```

```
library(ggplot2)
```

```
fit4.1
```

```
forecast(fit4.1)
```

```

#par(mfrow=c(1,2))

pred.tr <- predict(fit4.1, n.ahead = 8)

U.tr = pred.tr$pred + 2*pred.tr$se #upper bound

L.tr = pred.tr$pred - 2*pred.tr$se #lower bound

ts.plot(stock_train.log, xlim=c(1,length(stock_train.log)+8), ylim = c(min(stock_train.log),max(U.tr)), main = "Log
Differenced training set")

lines(U.tr, col="blue", lty="dashed")

lines(L.tr, col="blue", lty="dashed")

points((length(stock_train.log)+1):(length(stock_train.log)+8), pred.tr$pred, col="red")

pred.orig <- exp(pred.tr$pred)

U = exp(U.tr)

L = exp(L.tr)

ts.plot(stock_train[,2], xlim=c(1, length(stock_train[,2])+8), ylim = c(min(stock_train[,2]), max(U)), main =
"Original Training")

lines(U, col="blue", lty="dashed")

lines(L, col="blue", lty="dashed")

points((length(stock_train[,2])+1):(length(stock_train[,2])+8), pred.orig, col="red")

forecast.plot<-forecast(fit4.1) %>% autoplot(main="Forecasted daily JPM stock close price trend")

forecast.plot

```