

# Crest Data Forecasting

Natasha Leodjaja

5/26/2021

## Abstract or Executive Summary

In this report, I wanted to forecast the data set called “Crest” which is data set number 563 in the `tsdl` package in RStudio. In order to forecast the next 12 data (in which case is the last 12 data of this data set), I plotted and analyzed the Crest time series, transformed data into a stationary series, plotted and analyzed ACF and PACF to preliminary identify model, fitted the model by choosing the model with the lowest AICc value, forecasted the model, and lastly, compared the forecasted data to the test data which is the last 12 data in the crest data set. As a result, I fitted my model to be a MA(1) model (Moving Average) and forecasted the next 12 data that shows a slow and steady increase.

## Main body of the report

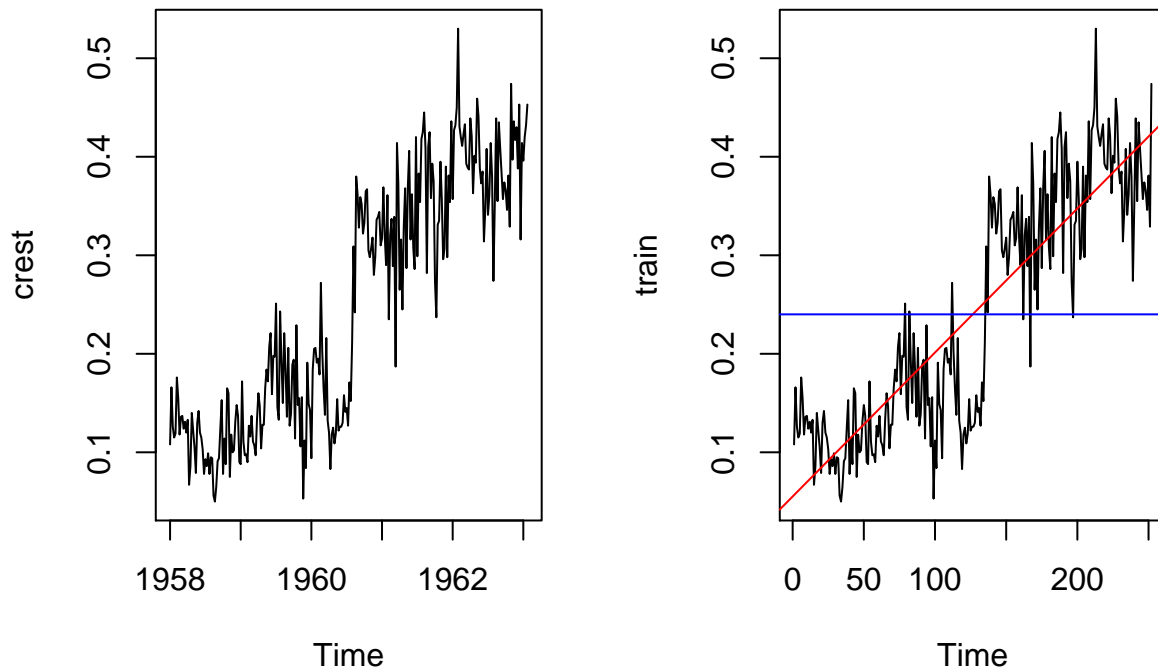
I chose the data set called “Crest Sales” as my data to analyze and forecast; which is data set number 563 in the `tsdl` package found in RStudio packages. This data set contains weekly market share data of Crest sales from January 1958 to April 1963. This data set is interesting to me because I myself am a customer of Crest products such as their toothpaste. So it is interesting to learn more about their market share. Additionally, this time series analysis overall could help me learn more on applying time series analysis in the marketing or sales department in the future as a Data Scientist.

To start with this analysis, I first checked if data itself is stationary, seasonal, or if there is a trend using the train data. Since I will be comparing the end result to the test data at the very end. As data looked non stationary, I applied transformation and de-trend at lag 1. However, I was unsure if data has seasonality component or not. Thus, I repeated the whole step above assuming data is “seasonal”. Now that I have 2 transformed models, where one was differenced once (non seasonal) and the other was differenced twice (seasonal). After obtaining two stable series, I then did model estimation by looking at the ACF/PACF of the data as well as parameter estimation. I noticed that both series shows a moving average model. Because one of the series was assumed “seasonal”, it was modeled to be SARIMA where the other was a pure MA model. Then, I performed diagnostic checking to find the model that fits best to my data by comparing their AICc values and choose the one with the lowest value and do forecasting with chosen model.

After applying all these techniques, I find out that my model is a MA(1) model where  $X_t = Z_t - 0.6404Z_{t-1}$ . I rejected the SARIMA model as not only it has a higher AICc value, it was also differenced twice as I assumed “seasonality”. Interestingly enough, when forecasting the next 12 data ahead, I noticed that all of my “forecasted” values slowly increases without any spikes. Where only 1 out of the 12 forecasted value matched closely to the test value. This is because a pure MA(q) forecast will be flat after q points.

## Plot and Analyze time series

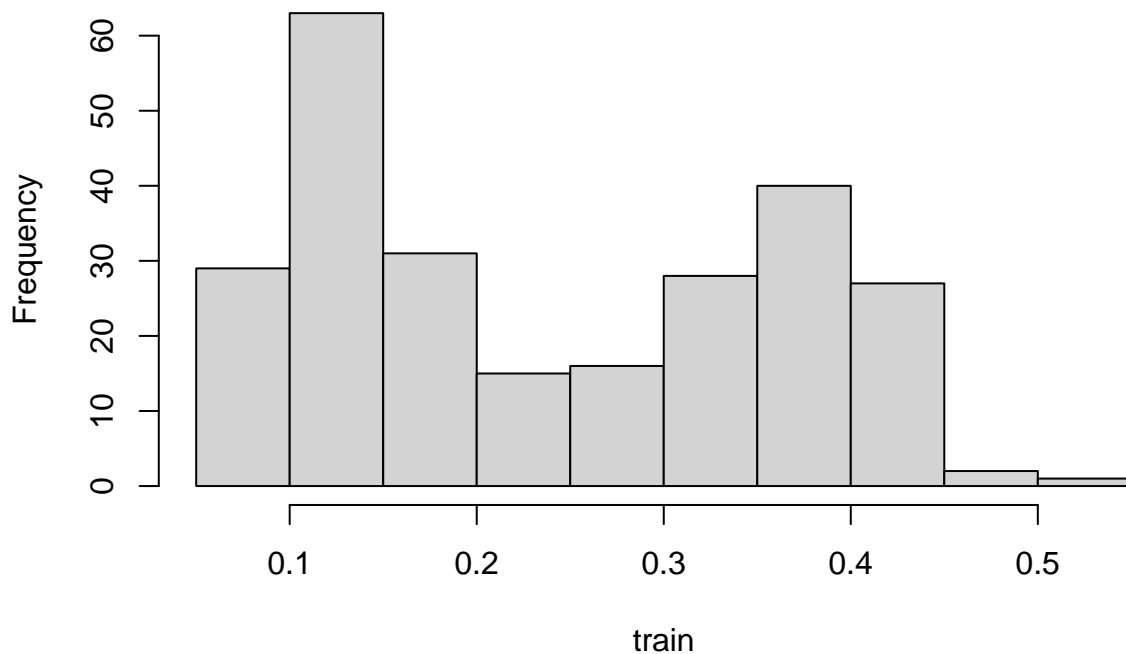
To start with this time series analysis, I first loaded R packages needed to use some of their commands. Such as `tsdl`, `astsa` and `forecast` libraries are very important to this analysis. After loading them, I separated the data set which I have named as ‘crest’ into 2 separate variables, one is called ‘train’ and the other is called ‘test’. Train consists the first 252 data out of the 264 and test contains the last 12 data from this data set. Next, I plotted crest to see if data is stationary.



The graph above shows that there is a linear trend and it is non stationary. It also looks like there is an upward trend as time goes by and there is also a sudden increase spike in the middle of 1960. However, we do observe slight seasonal trend. To confirm seasonality, we will have to compare histograms of both seasonal transformation and non seasonal transformation.

```
hist(train, main = "Histogram of Crest Data (Train)") # histogram
```

### Histogram of Crest Data (Train)



```
# check mean and variance
mean(crest[0:132]) # mean of first half of data
```

```
## [1] 0.1357803
mean(crest[233:264]) # mean of second half of data

## [1] 0.389875
var(crest[0:132]) # variance of first half of data

## [1] 0.001869058
var(crest[233:264]) # variance of second half of data

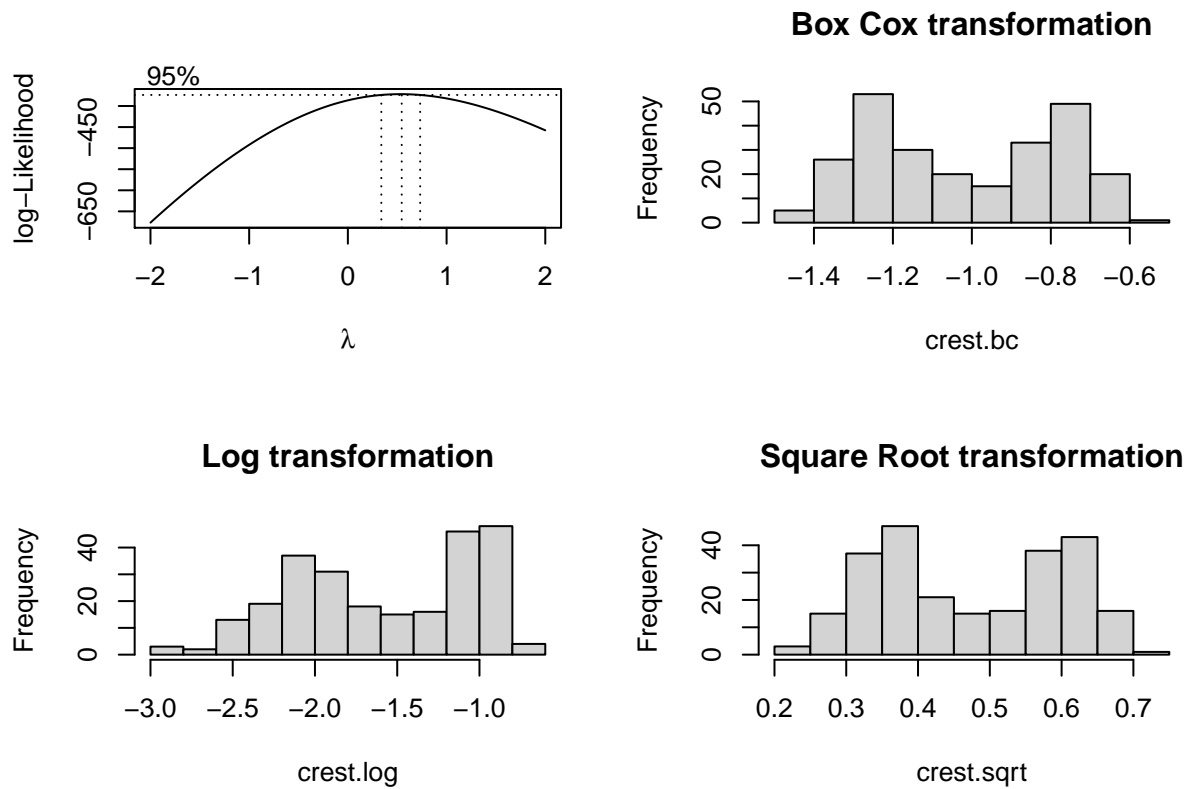
## [1] 0.002029597
```

From above histogram (Train), we can see that it is not stationary. Another indication that it is non stationary is the mean and variance calculated for the first and second half of the data. We can see that the mean is very different for the first and second half of the data. Unlike the mean, the variance does not differ too much for the first and second half of the data. Thus, I can confirm that data is not stationary.

### Transformation to get a stationary and stable series

Since data is not stable and stationary, I performed transformation and differencing to stabilize them.

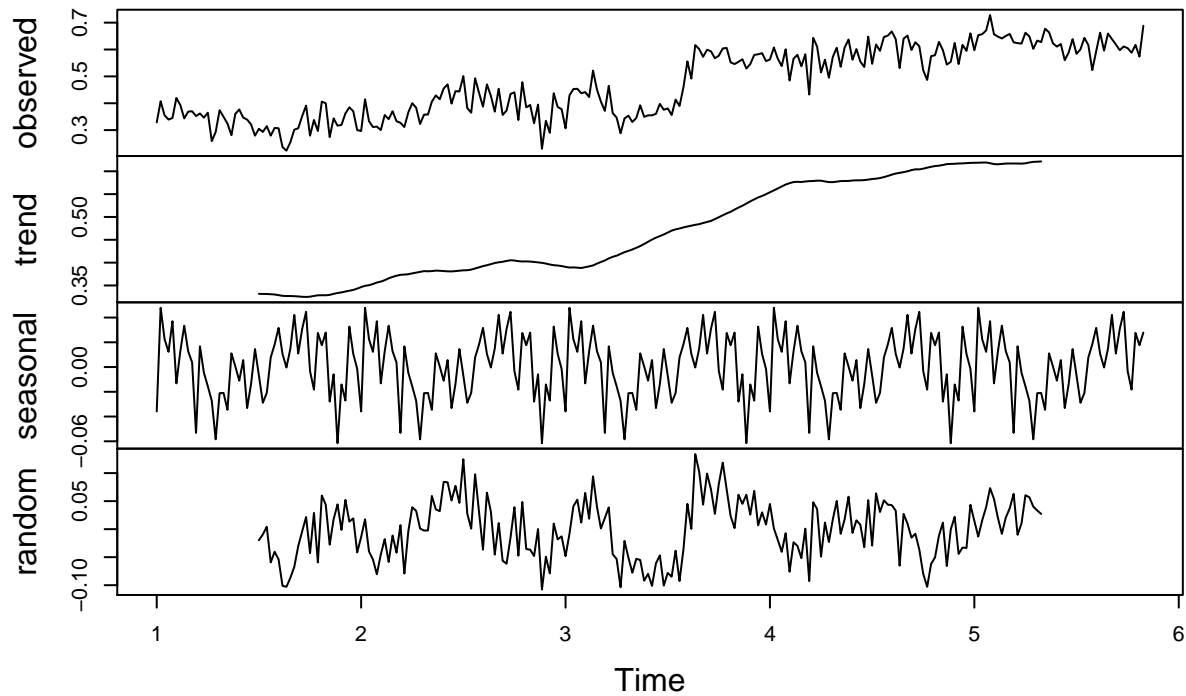
```
par(mfrow=c(2,2))
# variance stabilizing transformation
crest.log <- log(train)
# box cox transformation
bcTransform <- boxcox(train~ as.numeric(1:length(train))) # plots the graph
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))] # lambda = 0.6262626
crest.bc = (1/lambda)*(train^lambda-1)
hist(crest.bc, main="Box Cox transformation")
hist(crest.log, main="Log transformation")
crest.sqrt <- sqrt(train)
hist(crest.sqrt, main="Square Root transformation")
```



The first graph showed the log likelihood of the data and I obtained the lambda value to be 0.6262 which is close to 0.5 through the boxcox command. Thus, I applied 3 transformations: square root, log and boxcox to see which produces a more stable histogram. Looking at the histograms of the 3 transformations, I can see that square root transformation has the best outcome in terms of stability. Therefore, I will move forward with square root transformation.

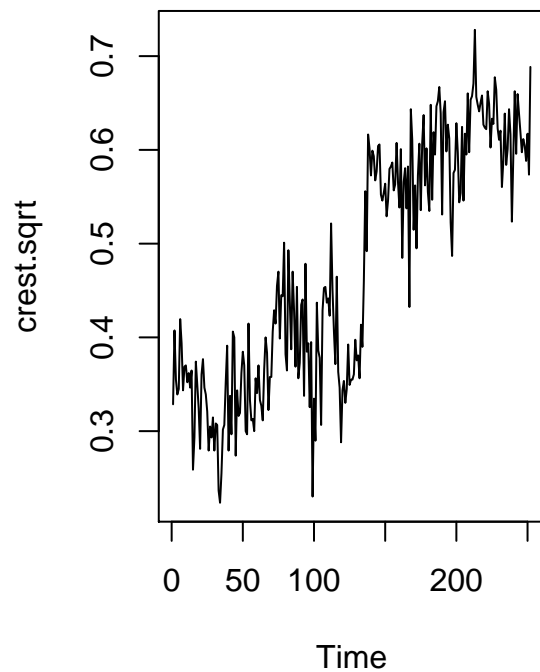
To confirm seasonality, I plotted the decomposition graph. The graph shows that there is a trend but no seasonality. To confirm this, I will compare the histograms of both seasonal transformation and non seasonal transformation.

## Decomposition of additive time series

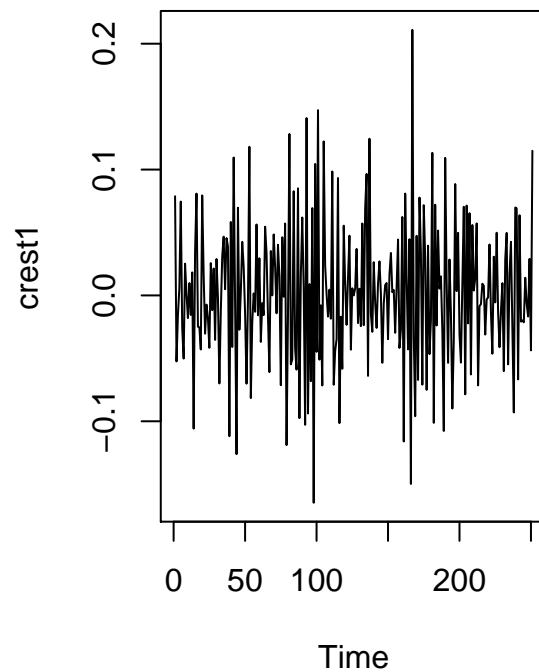


First, I plotted the histogram of the square root transformation alongside the histogram of data that was differenced at lag 1 to remove trend. The left plot (square root transformation) shows trend, where the right plot (de-trended at lag 1) shows a more stable series.

### Sqrt transformation



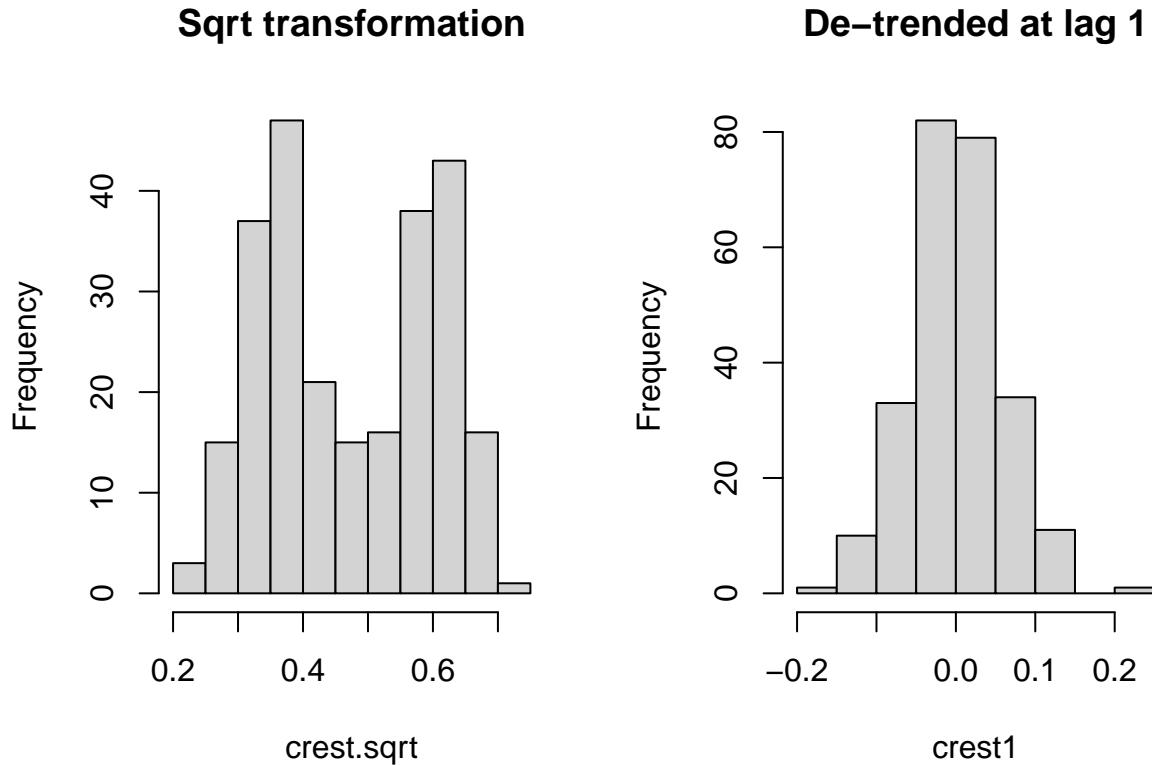
### De-trended at lag 1



The histogram of de-trended data also looks more stable compared to the square root transformation which

has not been differenced. I also compared the variance of the square root transformation and the de-trended data. Square root transformation has a higher variance compared to the latter. This shows that data is more stable now.

```
par(mfrow=c(1,2))
hist(crest.sqrt, main="Sqrt transformation")
hist(crest1, main="De-trended at lag 1")
```



```
var(crest.sqrt)
```

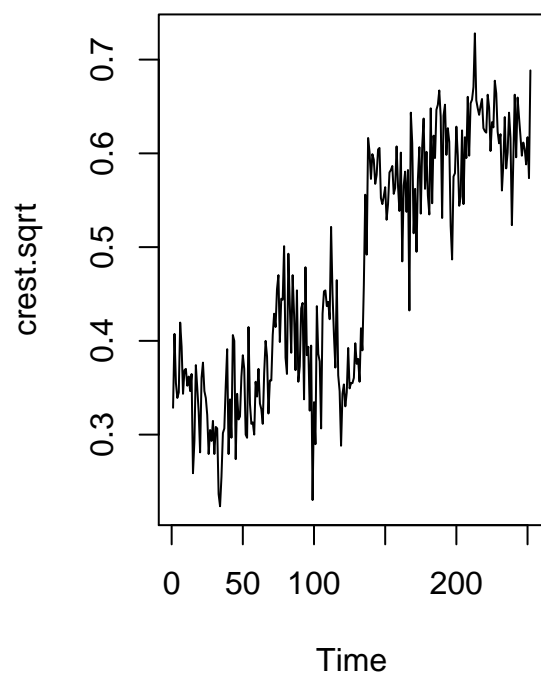
```
## [1] 0.01638277
```

```
var(crest1)
```

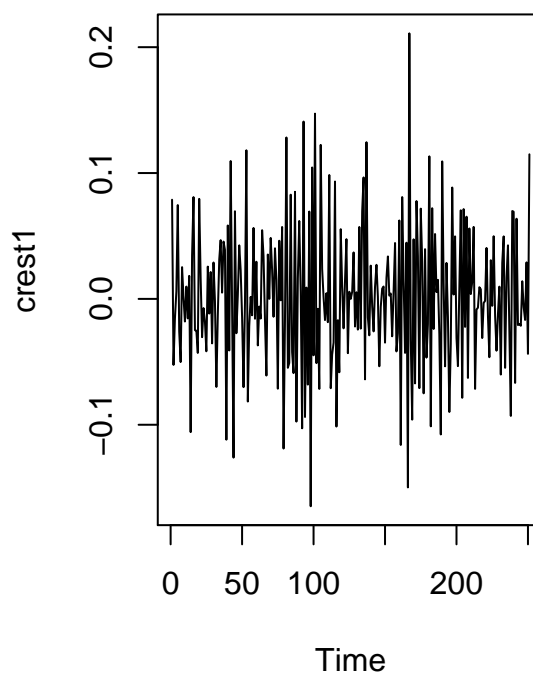
```
## [1] 0.00324855
```

Next, I performed differencing once to remove 'seasonality' and second to trend. Since this data set is a weekly data set, I will applied the first differencing at lag 52 to remove seasonality.

### Sqrt transformation



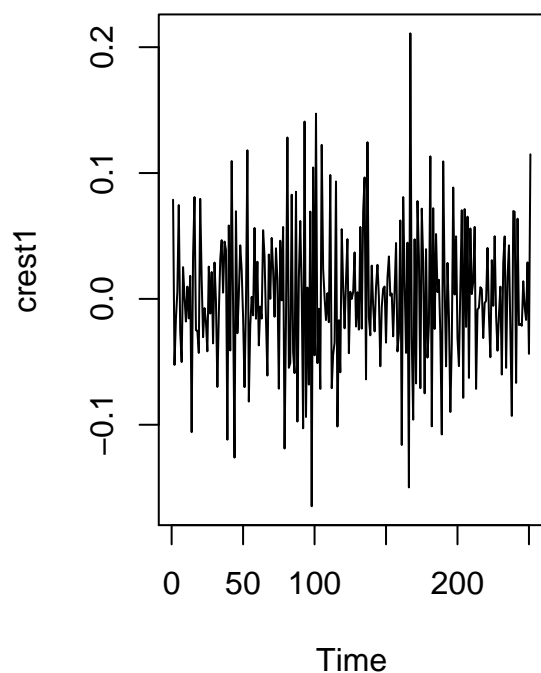
### Differenced at lag 52



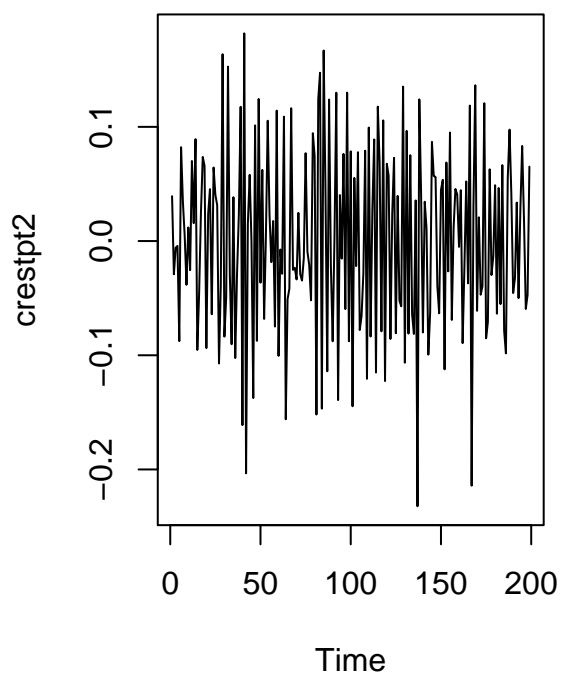
After the first differencing, I noticed that the plot still shows a slight trend thus I proceeded with second differencing to remove trend.

```
par(mfrow=c(1,2))
crestpt2 <- diff(crestpt1, lag=1) # second differencing at lag 1
ts.plot(crest1, main="Differenced at lag 52")
ts.plot(crestpt2, main="Differenced at lag 52 and lag 1")
```

**Differenced at lag 52**

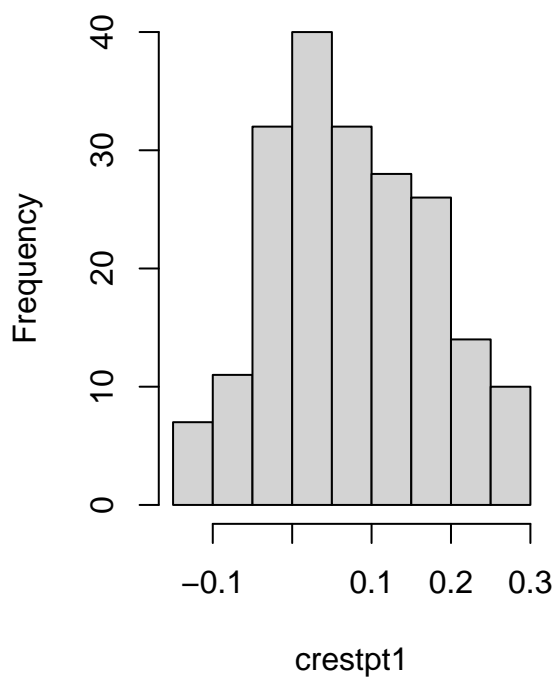


**Differenced at lag 52 and lag 1**

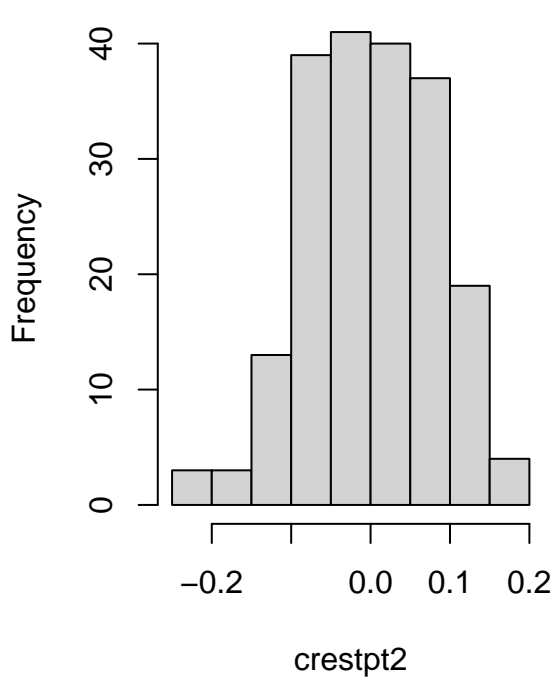


```
hist(crestpt1)
hist(crestpt2)
```

**Histogram of crestpt1**



**Histogram of crestpt2**



```
var(crest.sqrt)
```

```
## [1] 0.01638277
```



```
var(crestpt1)
```

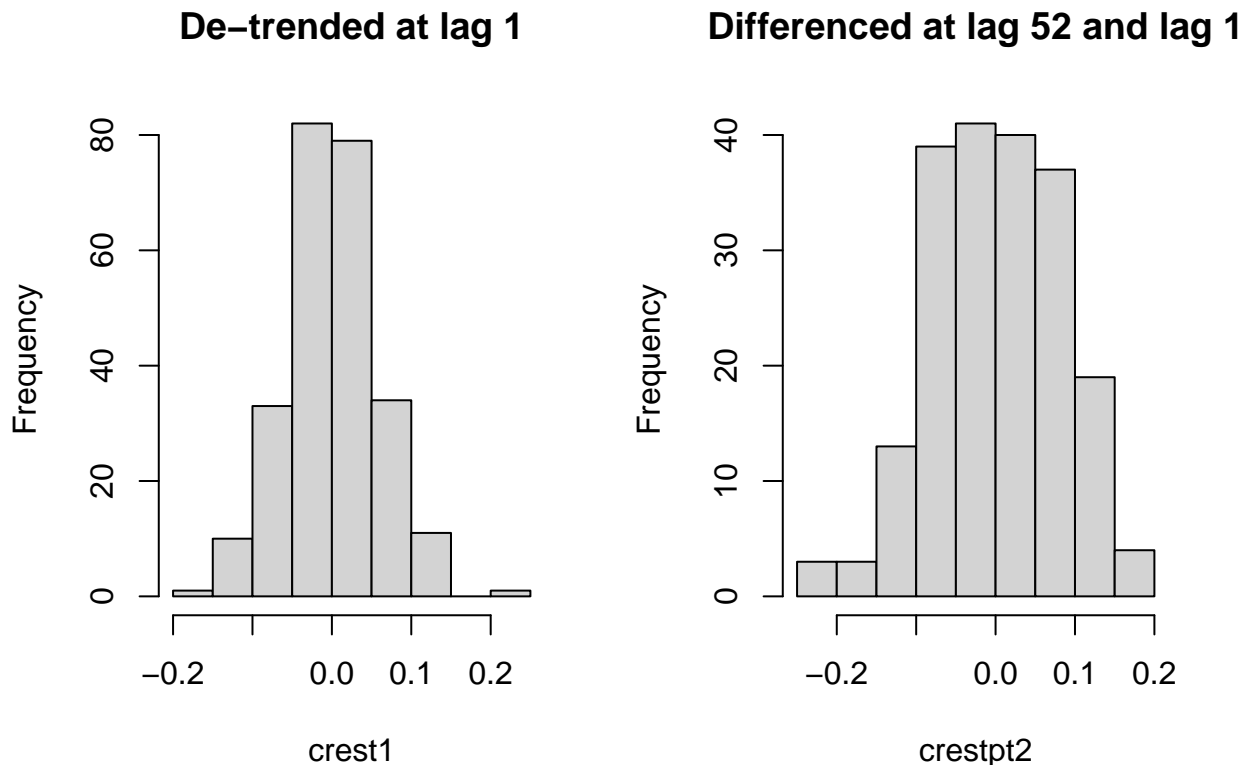
```
## [1] 0.009749634
```

```
var(crestpt2)
```

```
## [1] 0.006624294
```

Histogram shows that data looks more stationary after both differencing as compared to train data. Variance also keeps decreasing as differencing was applied. Train data has a variance of 0.016 whereas the data with both differencing has a variance of 0.0066. Although data is now stationary, we still need to check the ACFs (Autocorrelation function).

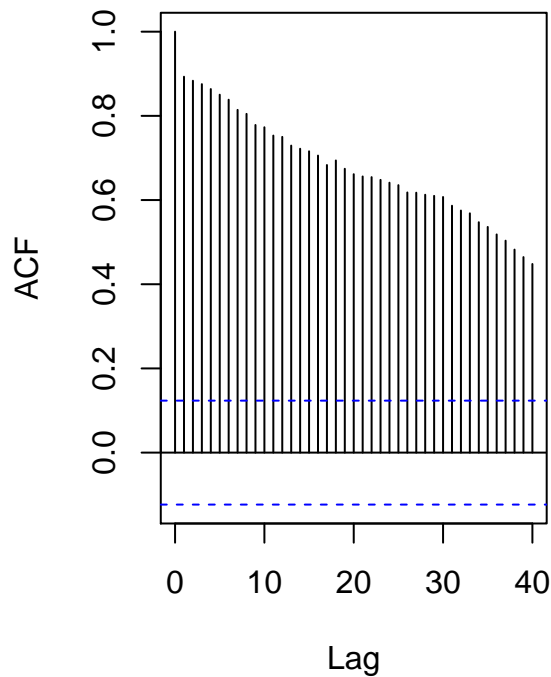
Now let's compare both histograms of seasonal and non seasonal transformation.



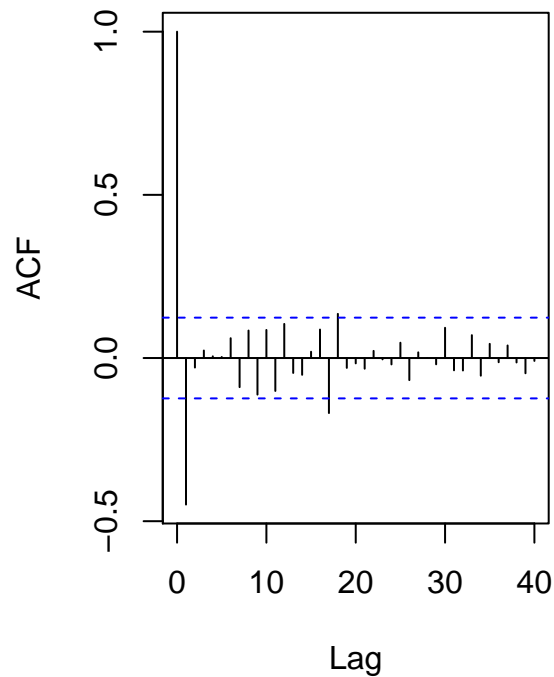
The first histogram looks more stable and stationary as compared to the second histogram. However, I will keep both models and try to see which model fits best by comparing AICc values.

```
par(mfrow=c(1,2))
acf(crest.sqrt, lag.max=40, main="Sqrt transformation")
acf(crest1, lag.max=40, main="De-trended at lag 1")
```

**Sqrt transformation**

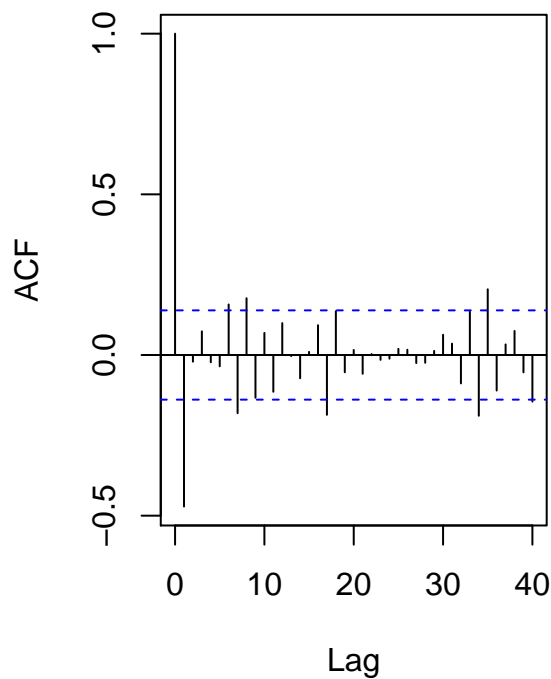


**De-trended at lag 1**



```
acf(crestpt2, lag.max=40, main="Differenced at lag 52 and lag 1")
```

**Differenced at lag 52 and lag 1**

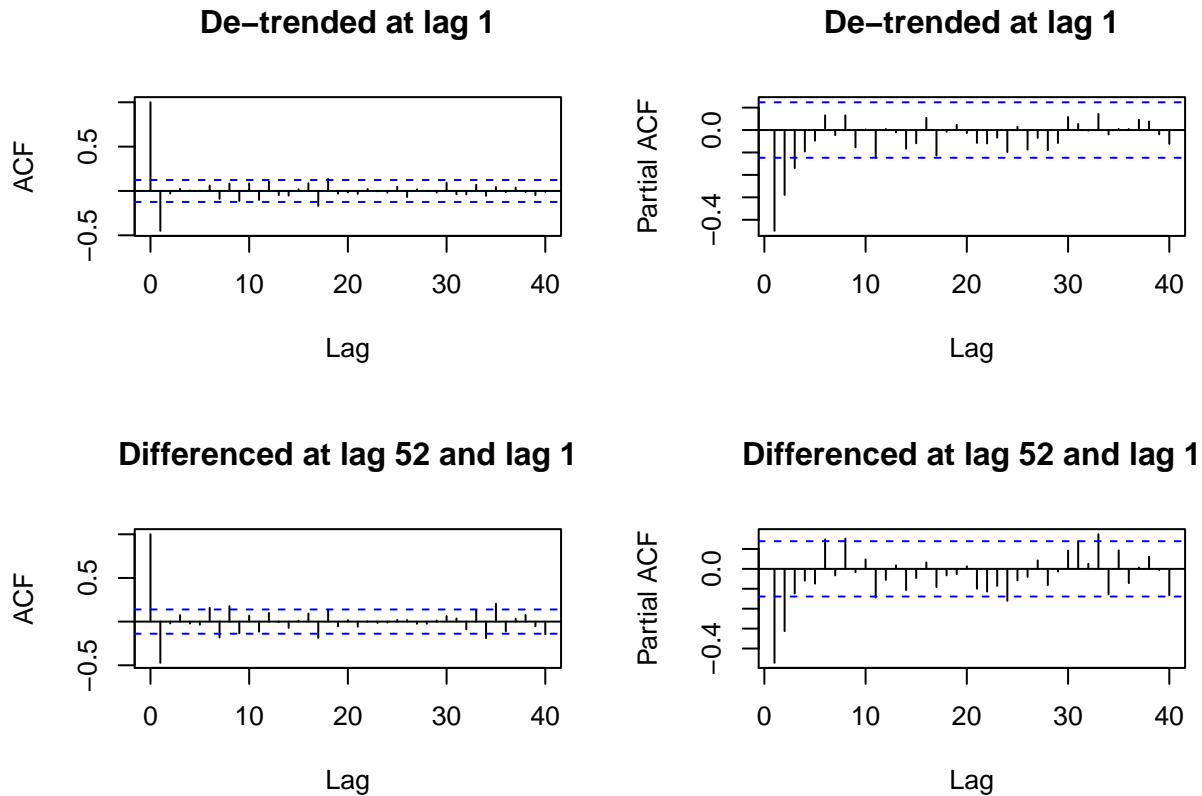


The ACF of the original data indicated it is non stationary and there is still a slow downward or decreasing trend. The second ACF (de trended at lag 1) showed no seasonality and is stationary. The third ACF (differenced at lag 52 and then at lag 1) shows that it is now stationary.

## Model Selection

```
par(mfrow=c(2,2))
acf(crest1, lag.max=40, main="De-trended at lag 1")
pacf(crest1, lag.max=40, main="De-trended at lag 1")

acf(crestpt2, lag.max=40, main="Differenced at lag 52 and lag 1")
pacf(crestpt2, lag.max=40, main="Differenced at lag 52 and lag 1")
```



**First Model** ACF cuts off after lag 1 which indicates MA(1) model.

**Second Model** ACF lies outside confidence interval at lag 1 maybe lag 6, 7, 8, 16, 30 and 31. Based on the PACF (partial autocorrelation function) plot, PACF lies outside confidence interval at lag 1, maybe 2, 6, 8, 11, 31 and 33.

Modeling the seasonal part:

1. I applied one seasonal differencing so  $D = 1$  at lag  $s = 52$  (weekly data).
2. The ACF shows a strong peak at  $h = 1s$ . A good choice for the MA part is  $Q=1$ .

Modeling the non seasonal part:

1. The ACF seems to be tailing off. A good choice for the MA part is  $q=1$ .

The first model suggests pure MA(1) and the second model suggests SARIMA(0,0,1) $\times$ (0,0,1) $_s=52$ .

## Parameter Estimation

To see which model fits best with my data set, I applied the arima function to both MA and SARIMA suggestion.

```
# Pure MA(1)
fit_ma = arima(crest1, order = c(0, 0, 1))
AICc(fit_ma)

## [1] -809.9658

fit_ma

##
## Call:
## arima(x = crest1, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       -0.6404    0.0011
## s.e.    0.0467    0.0011
##
## sigma^2 estimated as 0.002263:  log likelihood = 408.03,  aic = -810.06
```

```
# SARIMA(0,0,1)x(0,0,1)s=52
fit_sma = arima(crestpt2, order = c(0, 0, 1),
               seasonal = list(order = c(0, 0, 1),
                               period = 52), method="ML")
AICc(fit_sma)
```

```
## [1] -569.1711

fit_sma

##
## Call:
## arima(x = crestpt2, order = c(0, 0, 1), seasonal = list(order = c(0, 0, 1),
##      period = 52), method = "ML")
##
## Coefficients:
##          ma1      sma1  intercept
##       -0.6222  -1.0000      1e-04
## s.e.    0.0511   0.2655      8e-04
##
## sigma^2 estimated as 0.002129:  log likelihood = 288.69,  aic = -569.38
```

SARIMA model has a higher AICc of -569.17 whereas MA model has a lower AICc of -809.97. Therefore, we will move forward with the MA model as it has a lower AICc.

Model with the lowest AICc value:  $X_t = Z_t - 0.6404Z_{t-1}$ .

Model is stationary as it is a MA(1) model. It is also invertible as  $|\theta_1| < 1$ .

$\hat{\sigma}_{z^2} = 0.002263$ .

Model with the higher AICc value:  $X_t = (1 - 0.6222_{0.0511}B)(1 - 1_{0.2655}B^{52})Z_t$ .

This model is also stationary but not invertible as  $|\theta_1| < 1$  but  $|\phi_1| \not< 1$ .

$\hat{\sigma}_{z^2} = 0.002129$ .

## Diagnostic Checking

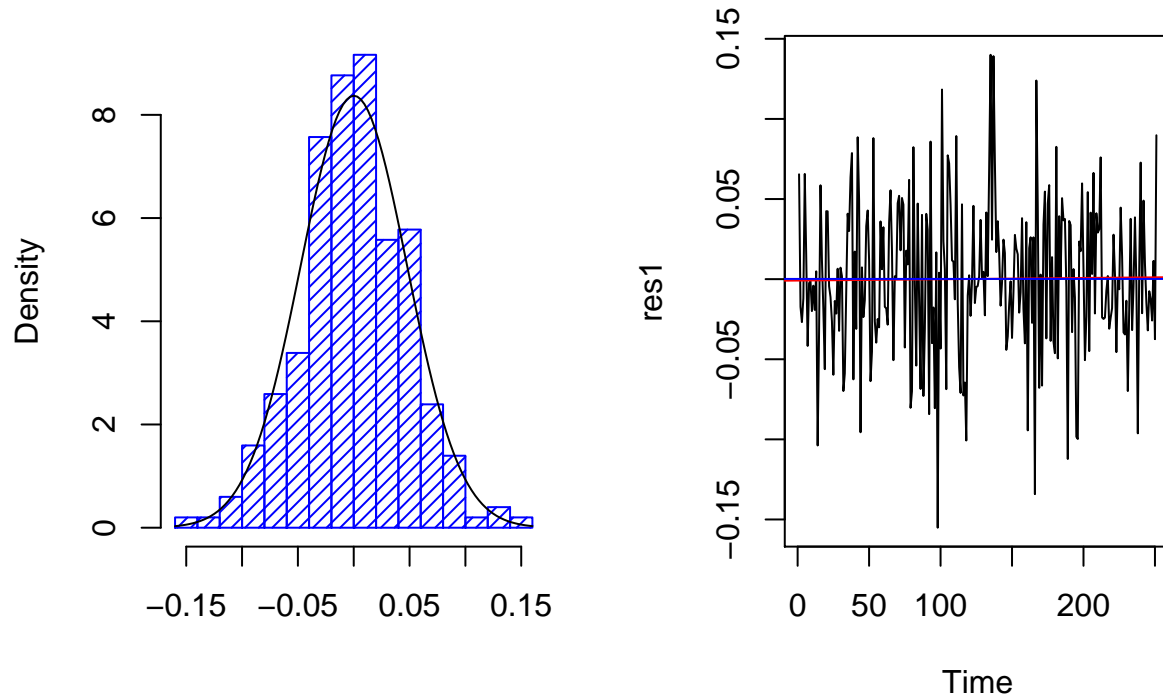
```
par(mfrow=c(1,2))
# Residuals for first model (MA)
res1 <- residuals(fit_ma)
```

```

hist(res1, main = "Residuals for MA", density=20, breaks=20, col="blue", xlab="", prob=TRUE)
m1 <- mean(res1)
std1 <- sqrt(var(res1))
curve(dnorm(x,m1,std1), add=TRUE )
plot.ts(res1)
fit1 <- lm(res1 ~ as.numeric(1:length(res1)))
abline(fit1, col="red")
abline(h=mean(res1), col="blue")

```

## Residuals for MA

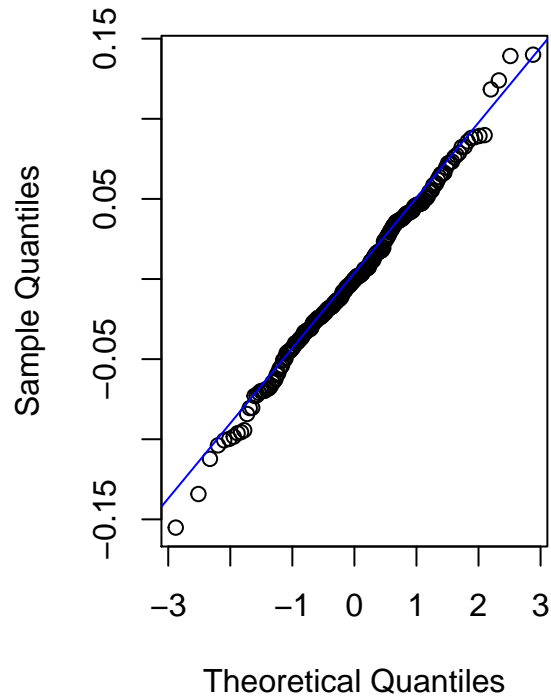


```

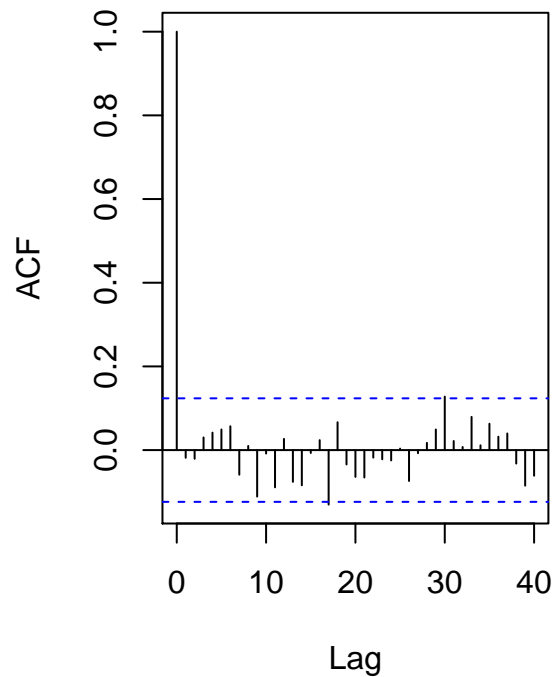
qqnorm(res1,main= "Normal Q-Q Plot for MA model")
qqline(res1,col="blue")
acf(res1, lag.max=40)

```

### Normal Q–Q Plot for MA model



### Series res1



```
pacf(res1, lag.max=40)
shapiro.test(res1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res1
## W = 0.99481, p-value = 0.5524
```

```
Box.test(res1, lag = 12, type = c("Box-Pierce"), fitdf = 1)
```

```
##
##  Box-Pierce test
##
## data:  res1
## X-squared = 8.4708, df = 11, p-value = 0.6706
```

```
Box.test(res1, lag = 12, type = c("Ljung-Box"), fitdf = 1)
```

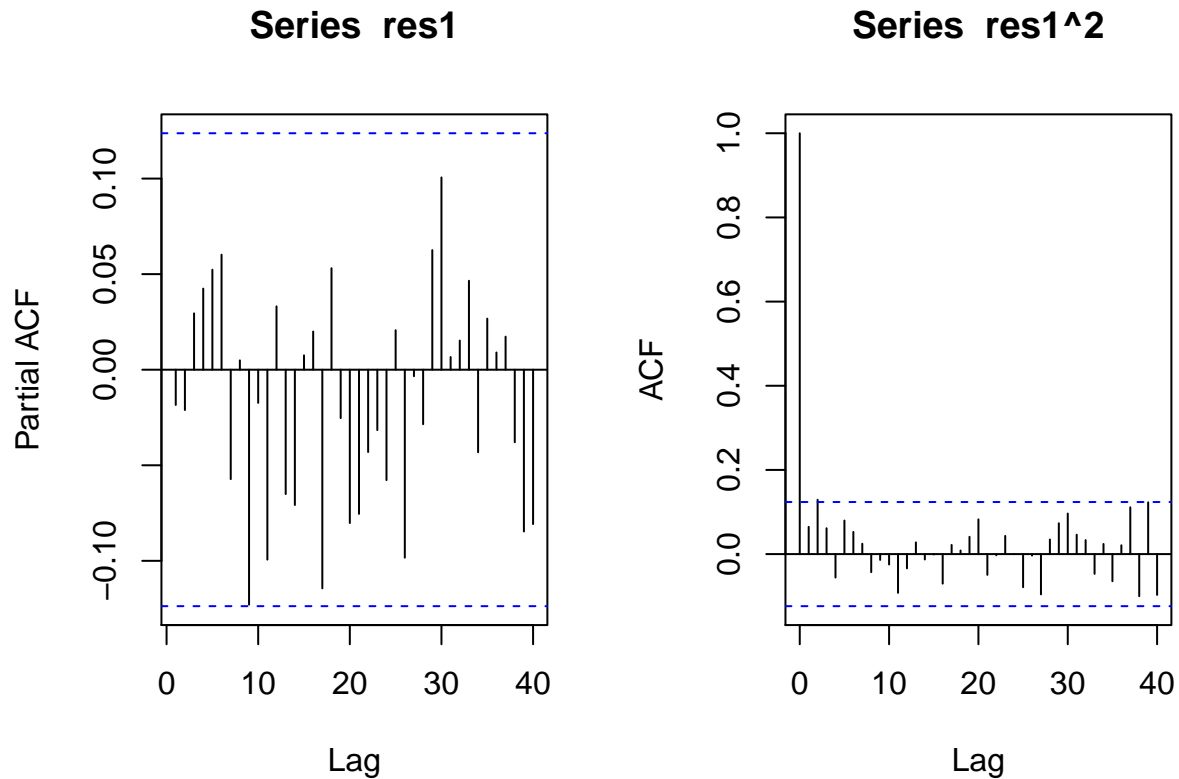
```
##
##  Box-Ljung test
##
## data:  res1
## X-squared = 8.8262, df = 11, p-value = 0.6379
```

```
Box.test(res1^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res1^2
```

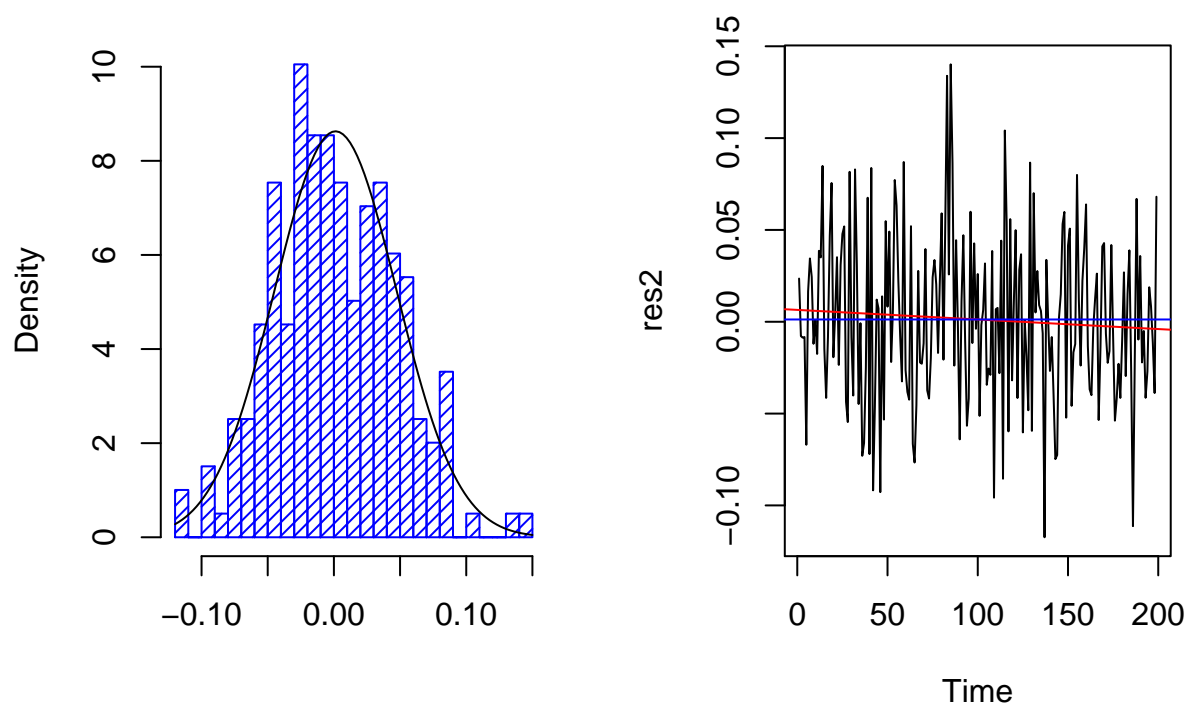
```
## X-squared = 12.844, df = 12, p-value = 0.3805
```

```
acf(res1^2, lag.max=40)
```



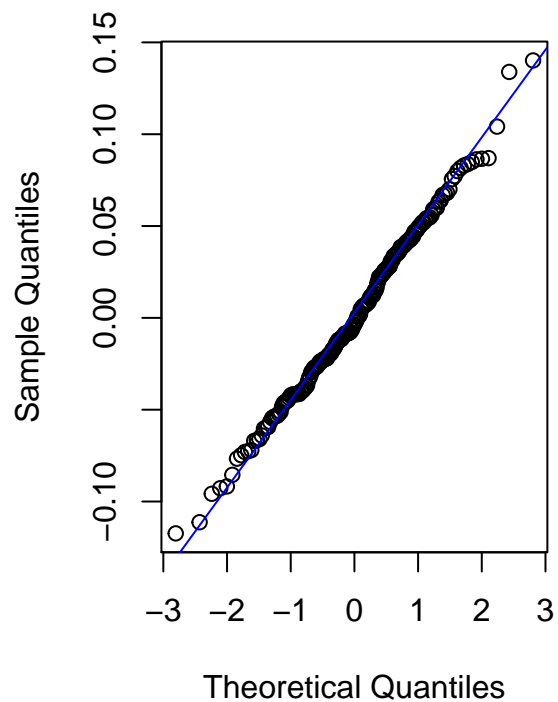
```
# Residuals for second model (SMA)
res2 <- residuals(fit_sma)
hist(res2, main = "Residuals for SARIMA", density=20, breaks=20, col="blue", xlab="", prob=TRUE)
m2 <- mean(res2)
std2 <- sqrt(var(res2))
curve(dnorm(x,m2,std2), add=TRUE )
plot.ts(res2)
fit2 <- lm(res2 ~ as.numeric(1:length(res2)))
abline(fit2, col="red")
abline(h=mean(res2), col="blue")
```

## Residuals for SARIMA

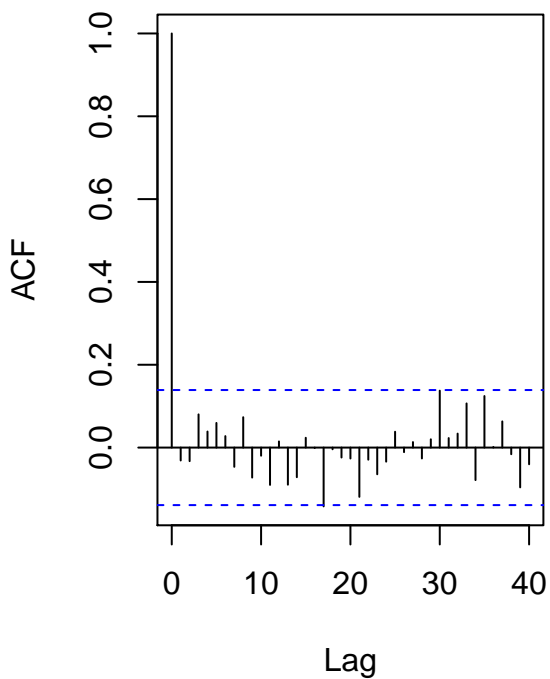


```
qqnorm(res2, main= "Normal Q-Q Plot for SARIMA model")
qqline(res2, col="blue")
acf(res2, lag.max=40)
```

## Normal Q-Q Plot for SARIMA model



## Series res2





```

pacf(res2, lag.max=40)
shapiro.test(res2)

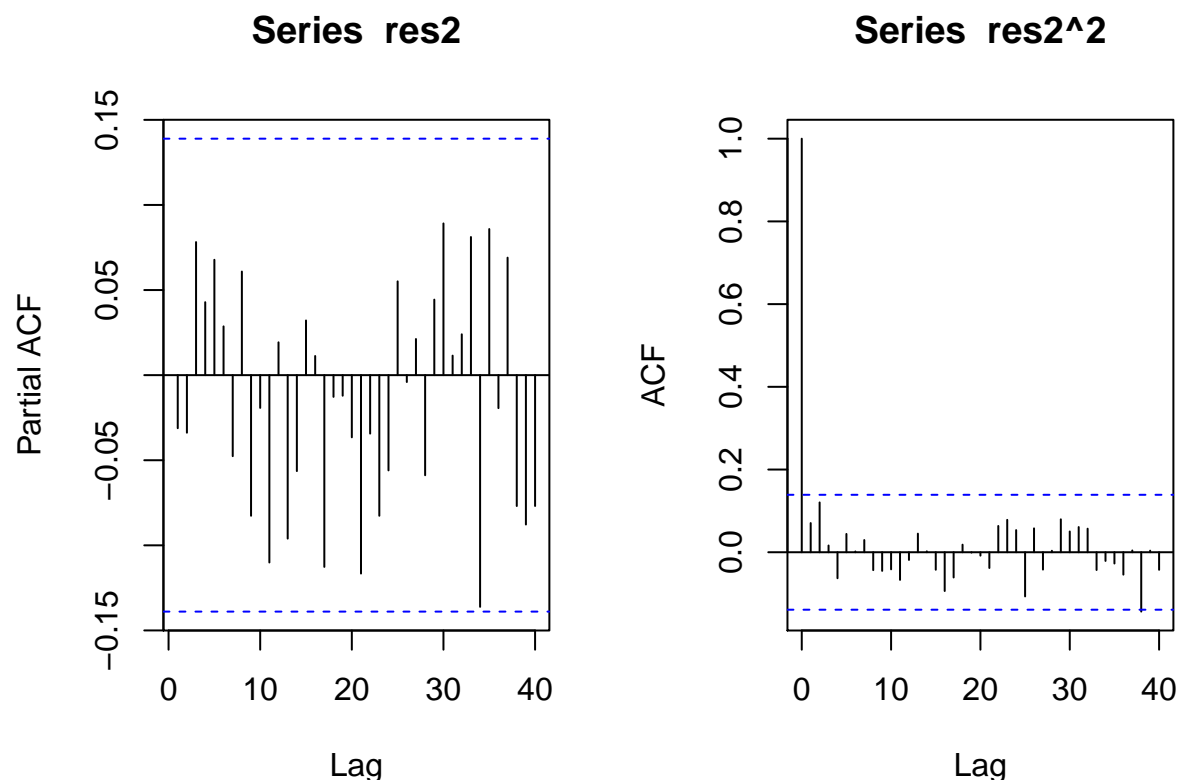
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.99456, p-value = 0.6878
Box.test(res2, lag = 12, type = c("Box-Pierce"), fitdf = 2)

##
##  Box-Pierce test
##
## data:  res2
## X-squared = 7.1273, df = 10, p-value = 0.7134
Box.test(res2, lag = 12, type = c("Ljung-Box"), fitdf = 2)

##
##  Box-Ljung test
##
## data:  res2
## X-squared = 7.465, df = 10, p-value = 0.6809
Box.test(res2^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)

##
##  Box-Ljung test
##
## data:  res2^2
## X-squared = 7.6076, df = 12, p-value = 0.815
acf(res2^2, lag.max=40)

```



The first model, MA(1) passes all diagnostic checking tests with all p-values larger than 0.05. All sample ACF and PACF are also within the confidence interval. The second model, SARIMA(0,0,1)x(0,0,1)s=52 also passes all diagnostic checking tests with p-values larger than 0.05. Both residual plots and histograms also resembles white noise and Gaussian. Q-Q plots are also close to straight line and both shapiro tests shows normality. However, since from previous parameter estimation, SARIMA model has the higher AICc value, I will go forward with the MA model.

The model obtained by using AICc is the same as the model suggested by ACF/PACF which is the MA(1) model.

## Model Forecasting

I will proceed with model forecasting on the chosen model, MA(1).

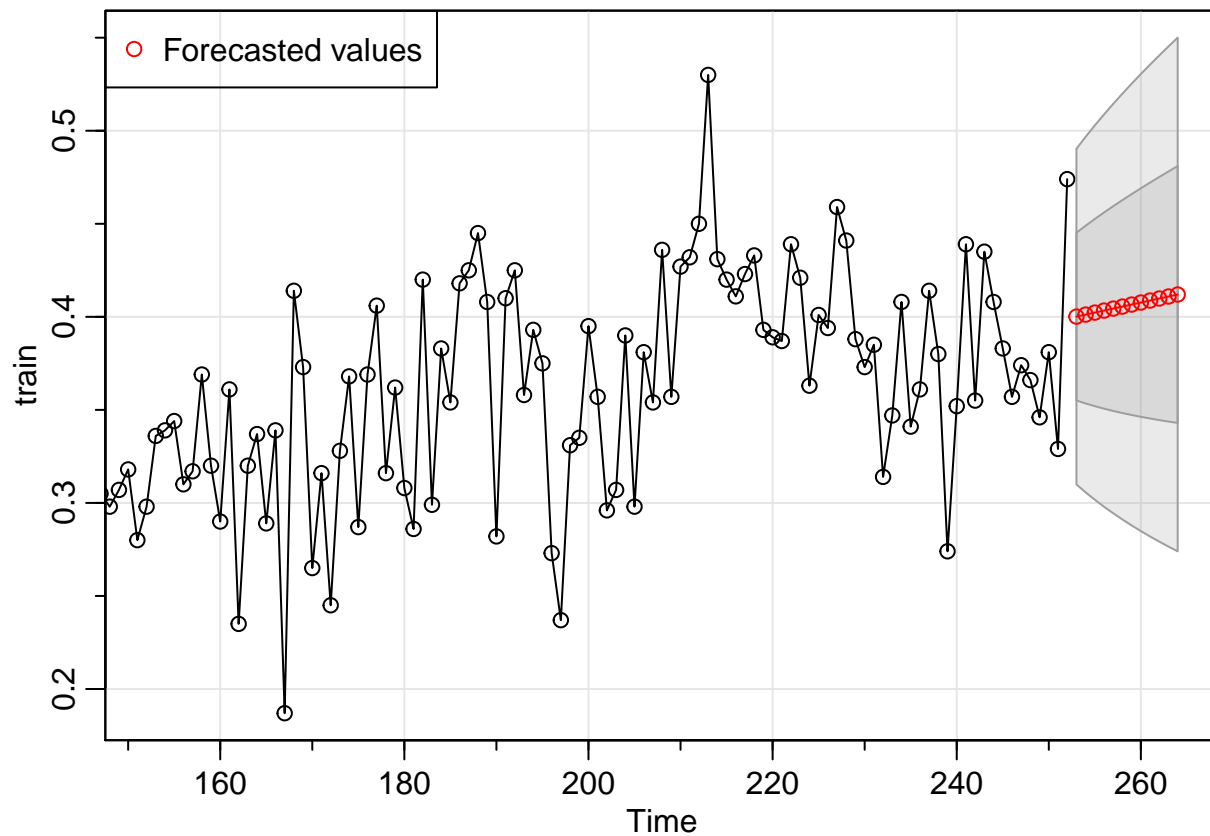
```
forecast(fit_ma) # forecast
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 252	-0.056424836	-0.11738455	0.004534873	-0.1496547	0.03680499
## 253	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214
## 254	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214
## 255	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214
## 256	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214
## 257	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214
## 258	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214
## 259	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214
## 260	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214
## 261	0.001091187	-0.07129882	0.073481189	-0.1096198	0.11180214

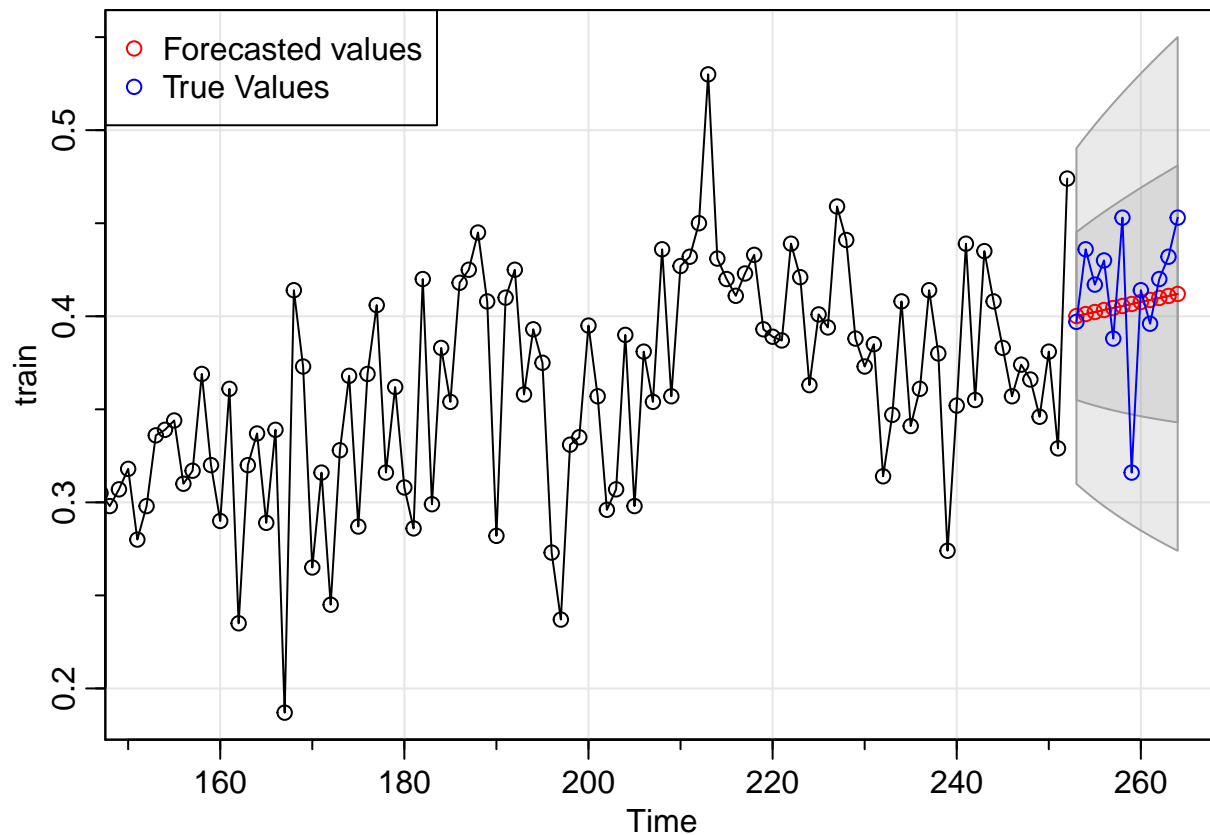
It shows from the table above that the values stopped changing after the first forecast. This is because a pure MA(q) forecast will be flat after q points. Which in this case, it stopped after the first point.

Predict 12 future observations.

```
fit_pred <- sarima.for(train, n.ahead=12, plot.all=F, p=0, d=1, q=1, P=0, D=0, Q=0, S=0)
legend("topleft", pch=1, col=c("red"), legend=c("Forecasted values"))
```



```
pred.tr <- sarima.for(train, n.ahead=12, plot.all=F, p=0, d=1, q=1, P=0, D=0, Q=0, S=0)
lines(253:264, pred.tr$pred, col="red")
lines(253:264, test, col="blue")
points(253:264, test, col="blue")
legend("topleft", pch=1, col=c("red", "blue"), legend=c("Forecasted values", "True Values"))
```



From the plots above, I can see that the first forecast value matches closely to the original data. The following forecast however, looks like the mean or average of the original data.

```
# check if fitted model matches the original model
auto.arima(train)
```

```
## Series: train
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##        -0.6446
## s.e.    0.0461
##
## sigma^2 estimated as 0.002051: log likelihood=420.88
## AIC=-837.75 AICc=-837.7 BIC=-830.7
```

## Conclusion

In conclusion, my goal of forecasting this model is achieved. I have fitted the exact model to this data set which is  $MA(1) X_t = Z_t - 0.6404Z_{t-1}$ . This can be checked by using `auto.arima`. There is also an explanation to why there is no spikes in the forecasted values, because MA models tend to show the mean or average of the data. Homework, Lecture notes and Labs has been very helpful in terms of helping me complete and perform this time series analysis.

## References

Homework 5 Lecture slides 12  
Lecture slides 15

Final project references Lab 5 solution

Lab 6 solution

Lab 7 solution

<https://api.github.com/repos/FinYang/tsdl/tarball/HEAD> (tsdl, 563)

<https://stackoverflow.com/questions/30350133/how-to-use-forecast-function-for-simple-moving-average-model-in-r>

<https://stats.stackexchange.com/questions/326193/why-do-i-get-constant-forecast-with-the-simple-moving-average-model>

## Appendix

```
# load packages
library(tsd1)
library(astsa)
library(MASS)
library(MuMIn)
library(ggplot2)
library(ggfortify)
library(forecast)

crest <- ts(tsd1[[563]], frequency = 52) # original data set
par(mfrow=c(1,2))
# plot data according to years
crest <- ts(crest, start = c(1958,1), end = c(1963,4), frequency = 52)
ts.plot(crest) # plot original data set

train <- crest[1:252] # training data set
test <- crest[253:264] # test data set
plot.ts(train)
fit <- lm(train ~ as.numeric(1:length(train)))
abline(fit, col="red")
abline(h=mean(train), col="blue")

hist(train, main = "Histogram of Crest Data (Train)") # histogram
# check mean and variance
mean(crest[0:132]) # mean of first half of data
mean(crest[233:264]) # mean of second half of data
var(crest[0:132]) # variance of first half of data
var(crest[233:264]) # variance of second half of data

par(mfrow=c(2,2))
# variance stabilizing transformation
crest.log <- log(train)
# box cox transformation
bcTransform <- boxcox(train~ as.numeric(1:length(train))) # plots the graph
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))] # lambda = 0.6262626
crest.bc = (1/lambda)*(train^lambda-1)
hist(crest.bc, main="Box Cox transformation")
hist(crest.log, main="Log transformation")
crest.sqrt <- sqrt(train)
hist(crest.sqrt, main="Square Root transformation")

# to produce decomposition of sqrt(U_t):
y <- ts(as.ts(crest.sqrt), frequency = 52)
```

```

decomp <- decompose(y)
plot(decomp) # decomposition shows upward trend

# Non seasonal transformation
par(mfrow=c(1,2))
# Differencing
ts.plot(crest.sqrt)
crest1 <- diff(crest.sqrt, lag=1) # de-trended at lag 1
ts.plot(crest1, main="De-trended at lag 1")
hist(crest.sqrt)
hist(crest1)
var(crest.sqrt)
var(crest1)

par(mfrow=c(1,2))
# Seasonal transformation
crestpt1 <- diff(crest.sqrt, lag=52) # de-trended at lag 52
ts.plot(crest1, main="Differenced at lag 52")

par(mfrow=c(1,2))
crestpt2 <- diff(crestpt1, lag=1) # second differencing at lag 1
ts.plot(crestpt2, main="Differenced at lag 52 and lag 1")
hist(crest.sqrt)
hist(crestpt1)
hist(crestpt2)
var(crest.sqrt)
var(crestpt1)
var(crestpt2)

par(mfrow=c(1,2))
hist(crest1, main="De-trended at lag 1") # non seasonal
hist(crestpt2, main="Differenced at lag 52 and lag 1") # seasonal

par(mfrow=c(1,2))
acf(crest.sqrt, lag.max=40, main="ACF of the sqrt(U_t)")
acf(crest1, lag.max=40, main="De-trended at lag 1")
acf(crestpt2, lag.max=40, main="Differenced at lag 52 and lag 1")

par(mfrow=c(1,2))
acf(crest1, lag.max=40, main="De-trended at lag 1")
pacf(crest1, lag.max=40, main="De-trended at lag 1")

acf(crestpt2, lag.max=40, main="Differenced at lag 52")
pacf(crestpt2, lag.max=40, main="Differenced at lag 52 and lag 1")

par(mfrow=c(1,2))
acf(crest1, lag.max=40, main="De-trended at lag 1")
pacf(crest1, lag.max=40, main="De-trended at lag 1")

acf(crestpt2, lag.max=40, main="Differenced at lag 52")
pacf(crestpt2, lag.max=40, main="Differenced at lag 52 and lag 1")

fit_ma = arima(crest1, order = c(0, 0, 1))

```

```

AICc(fit_ma)
fit_ma

# SARIMA(0,0,1)x(0,0,1)s=52
fit_sma = arima(crestpt2, order = c(0, 0, 1),
                seasonal = list(order = c(0, 0, 1),
                                period = 52), method="ML")
AICc(fit_sma)
fit_sma

par(mfrow=c(1,2))
# Residuals for first model (MA)
res1 <- residuals(fit_ma)
hist(res1, density=20, breaks=20, col="blue", xlab="", prob=TRUE)
m1 <- mean(res1)
std1 <- sqrt(var(res1))
curve(dnorm(x,m1,std1), add=TRUE )
plot.ts(res1)
fit1 <- lm(res1 ~ as.numeric(1:length(res1)))
abline(fit1, col="red")
abline(h=mean(res1), col="blue")
qqnorm(res1,main= "Normal Q-Q Plot for second model")
qqline(res1,col="blue")
acf(res1, lag.max=40)
pacf(res1, lag.max=40)
shapiro.test(res1)
Box.test(res1, lag = 12, type = c("Box-Pierce"), fitdf = 1)
Box.test(res1, lag = 12, type = c("Ljung-Box"), fitdf = 1)
Box.test(res1^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(res1^2, lag.max=40)
ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# Residuals for second model (SMA)
res2 <- residuals(fit_sma)
hist(res2, density=20, breaks=20, col="blue", xlab="", prob=TRUE)
m2 <- mean(res2)
std2 <- sqrt(var(res2))
curve(dnorm(x,m2,std2), add=TRUE )
plot.ts(res2)
fit2 <- lm(res2 ~ as.numeric(1:length(res2)))
abline(fit2, col="red")
abline(h=mean(res2), col="blue")
qqnorm(res2,main= "Normal Q-Q Plot for first model")
qqline(res2,col="blue")
acf(res2, lag.max=40)
pacf(res2, lag.max=40)
shapiro.test(res2)
Box.test(res2, lag = 12, type = c("Box-Pierce"), fitdf = 2)
Box.test(res2, lag = 12, type = c("Ljung-Box"), fitdf = 2)
Box.test(res2^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(res2^2, lag.max=40)
ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))

```

```

forecast(fit_ma) # forecast

fit_pred <- sarima.for(train, n.ahead=12, plot.all=F, p=0, d=1, q=1, P=0, D=0, Q=0, S=0)
legend("topleft", pch=1, col=c("red"), legend=c("Forecasted values"))

pred.tr <- sarima.for(train, n.ahead=12, plot.all=F, p=0, d=1, q=1, P=0, D=0, Q=0, S=0)
lines(253:264, pred.tr$pred, col="red")
lines(253:264, test, col="blue")
points(253:264, test, col="blue")
legend("topleft", pch=1, col=c("red", "blue"), legend=c("Forecasted values", "True Values"))

prediction <- sarima.for(crest1, n.ahead=12, plot.all=F, p=0, d=0, q=1, P=0, D=0, Q=0, S=0)
U.tr= prediction$pred + 2*prediction$se # % upper bound of prediction interval
L.tr= prediction$pred - 2*prediction$se # % lower bound
ts.plot(crest1, xlim=c(1,length(crest1)+12), ylim = c(min(crest1),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(crest1)+1):(length(crest1)+12), prediction$pred, col="red")

par(mfrow=c(1,3))
pred.orig <- exp(prediction$pred)
U= exp(U.tr)
L= exp(L.tr)
ts.plot(train, xlim=c(1,length(train)+12), ylim = c(min(train),max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")

# check fit of model
auto.arima(train)

```