An Analysis and Forecasting of Crime Rates in Los Angeles from 2010-2017

Author: Nihaal Kumar
Perm # : 4517678
Class: PSTAT 174
Professor: Raya Feldman

Abstract

This project aims to use the Box Jenkins method of time series analysis to examine crime rates in Los Angeles from 2010 to September of 2017. The dataset for this report was taken from Kaggle, an online forum with datasets posted for analysis,competitions, and for general use by the public. My goal for this project was to take the dataset and use it to forecast future crime levels, and to observe whether or not there were noticeable factors that affected the levels of crime in the city. The dataset I used had information on the counts of all types of crimes in Los Angeles, which would have needed separate time series to analyze each distinct type. As a result, I chose to analyze the crime with the highest number of cases, which was "Battery- Simple Assault." These counters were grouped by month, leading to a time series of monthly data over the roughly 7 years of crime information.

The following analysis of this data was split into four parts, each part involving a different stage of the model building process. It should be noted that while I also had to clean the dataset to only include the one type of crime before proceeding to start the time series analysis, it will not be included in the main part of this report. However, all changes and modifications made to the dataset will be included in the appendix at the end of this report. As a result, the four sections of this report were split into preliminary analysis, model identification, diagnostic checking, and finally forecasting.

The preliminary analysis consists of taking the data and trying to achieve a stationary process. This was done through various checks for transformation, as well as differencing the data to account for any trends or seasonality components. The model identification phase involved taking the ACF/PACF of the stationary process from part 1, and trying to estimate parameters for a potential model. These estimates are then combined with estimates generated by the auto.arima() function within R, and the top 2 candidate models are chosen based on lowest AICc to be compared to each other with diagnostic tests.

These diagnostics tests involved visual plots to check for normality, as well as utilization of the well known portmanteau tests, which were used to analyze the residuals of each of the 2 candidate models to determine a winner. Upon finding the optimal model, it is then used to forecast the number of simple assault cases in Los Angeles both during the last 12 months of the data set, as well as a forecast of the 12 months after the end of the data. The resulting predictions indicated similar behavior to the past years of data, indicating that the behavior of crime rates in Los Angeles would stay the same as it has been, without much change.

Introduction

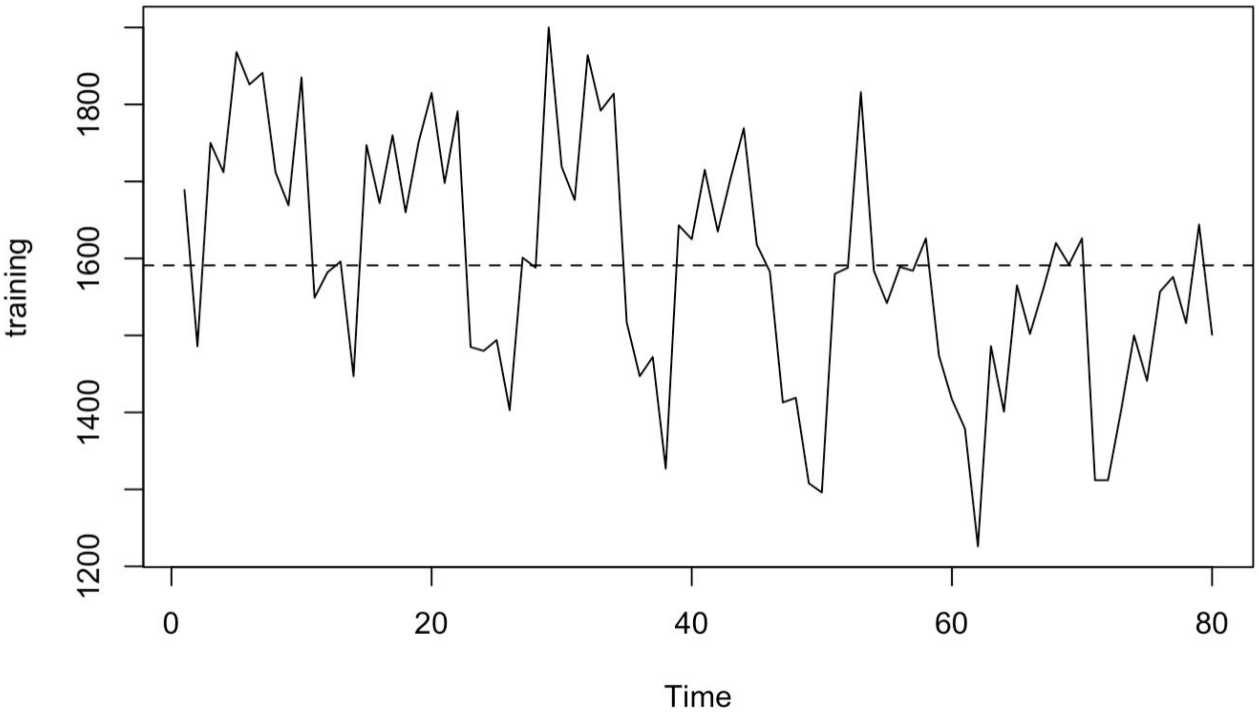
Los Angeles has been home to some of the highest crime rates in the country. Covering such a large area of land, the city encapsulates a large portion of the population in California. With all that has been going on in the world right now, I chose to analyze monthly crime rates in Los Angeles from 2010-2017, due to my passion for this topic during this time. By analyzing the number of occurrences of "Battery-Simple Assault" Cases, which was the most committed crime within the dataset, I hoped to find any insights into the behavior of crime throughout the year, and if there were any factors related to time which affected it. I expected the crime rates to be highest in the winter months, due to it being dark for longest during that time. However, as will be shown in this report, that was the opposite of the reality.

The summer months turned out to have the highest rates of crime, which at first did not make sense to me, but after some thought came into a better perspective. Summer is when the most amount of people are "free" from their obligations; students are out for break, and parents take time off to spend time with their families. As a result, the likelihood of crime rates being higher in summer actually ends making a lot of sense. But how was this conclusion reached?

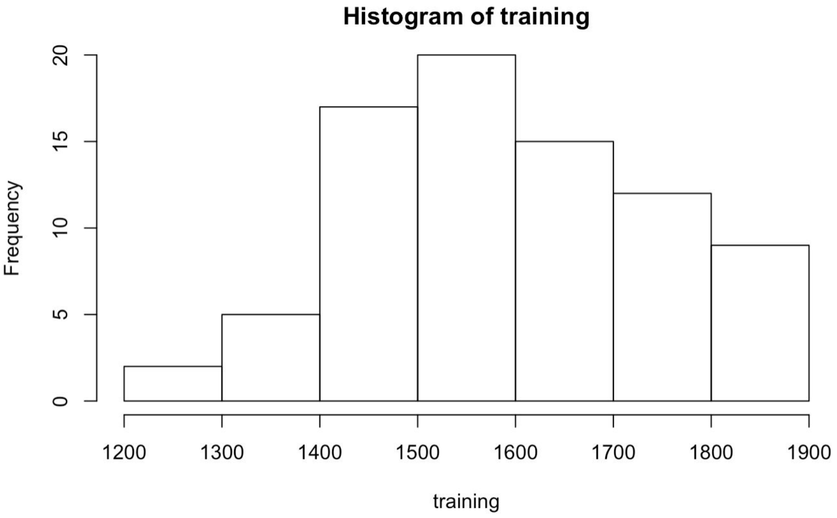
By applying the Box Jenkins methodology of time series, I predicted future crime rates in Los Angeles after completing a number of steps. I first performed necessary transformations and differences to achieve a stationary process. From there I used the ACF and PACF plots to estimate parameters for my seasonal arima model. Taking those estimations into account with estimations created by online software methods, I compared a large list of possible models to each other. The two models with the lowest AICc were then compared using diagnostic tests as well as the portmanteau tests, from which I arrived at a final SARIMA model, which had parameters (0,0,1) x (1,1,0)[12]. The majority of this project went without issue, but there is one big negative which should be addressed here. The number of observations in this data set was not adequate enough to meet the standards required for analysis of this level. With normality being a large factor in assessing the strength of a model, and sample size being a huge influencer on that normality, throughout this report you will notice the models not being as normal as would be preferred. I unfortunately made this realization far into the project, and didn't turn back. However, in the future I would make sure to check the observation count earlier on so that this issue doesn't occur. Even with this obstacle, the resulting model and predictions were well in line with past data, indicating that if the normality assumption was fully met, this model would be even more potent. The data set for this project(called "Crime in Los Angeles 2010-2017") was found on Kaggle, a public database for competitions and exploratory analysis, and the entire project was coded and created in Rstudio. We now move to the report itself.

Preliminary Analysis

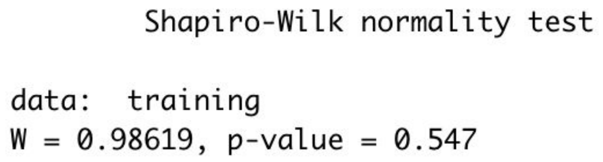
The first step of this analysis is to plot the monthly counts of simple assault battery cases as a time series.



Looking at the time series, there is no apparent increasing or decreasing trend. There are, however, spikes and dips in the data, which might indicate the presence of seasonality in the data. In terms of variance, it does not appear to be constant, but the mean does appear to be relatively constant. The goal of this stage of the analysis was to achieve a stationary process from which to estimate parameters for our model, and so I was given enough reason to check other visual plots to determine whether or not transformations were needed. To check if we needed to perform transformations, we first plot the histogram of the data.



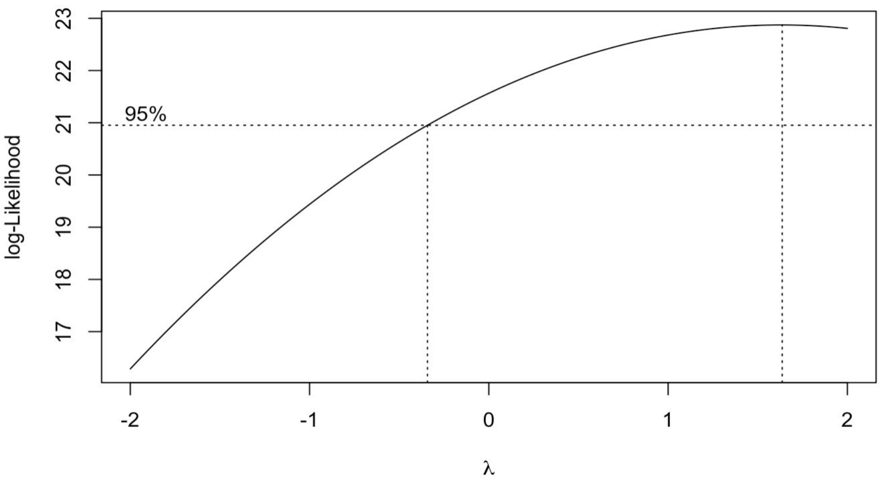
The histogram does appear to be close to normal, however the intended "bell curve" could be more even around the mean. In order to affirm or negate the normalcy of the histogram, I used the Shapiro-Wilks test for normalcy. Under this test, if the resulting p-value is less than 0.05, then the null hypothesis of a normal distribution is rejected. However, if the p-value is greater than 0.05, under the scope of the test it would mean the data is normally distributed.



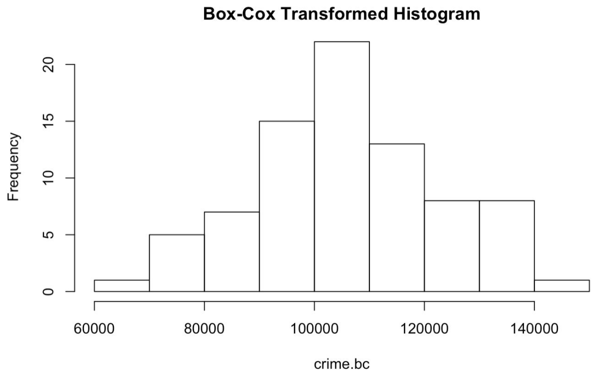
```
        Shapiro-Wilk normality test

data:  training
W = 0.98619, p-value = 0.547
```

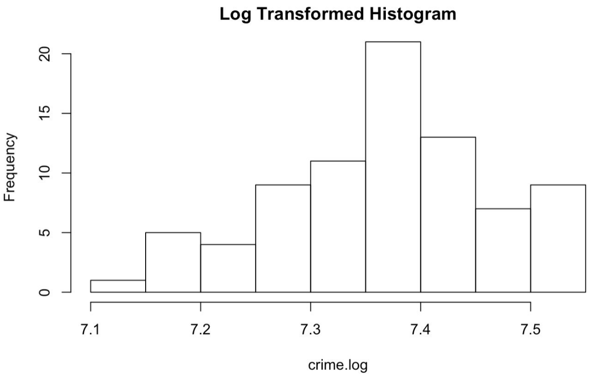
The resulting p-value of 0.547 is greater than 0.05, meaning that the null hypothesis of the data having a normal distribution could not be rejected. Since both the histogram and normality test indicated a normal distribution, this was a good sign that no transformations were needed. However, due to the variance of the series not being constant throughout the whole time series, I decided to attempt different transformations on the data to see if the resulting series was better than what it is right now,

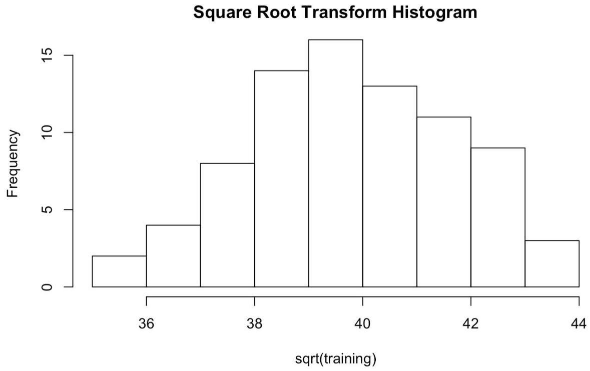
These transformations ranged from taking the square root of the data, the logarithm of the data, as well as performing the Box-Cox transformation. After performing each of these transformations, I planned to compare the resulting respective histograms to determine if any of the transformations made for a more stationary process overall. While the square root and logarithmic transformations are straight forward, the Box-Cox transformation required me to check the Box-Cox plot. The resulting plot is shown below.



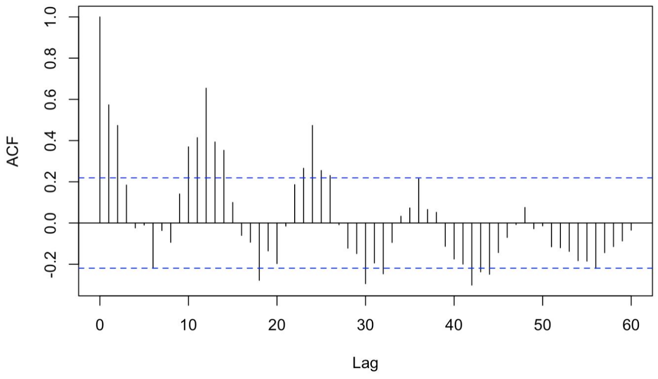
Looking at the Box-Cox plot, I checked to see what potential values of λ were within the 95% confidence interval. Something noticeable here is that the interval contains a rather large range of values.   Because of this, I used the Bc.transform function to obtain the value for λ to transform the data by, which came out to be λ= 1.63634. With the potential transformations complete, it was time to compare the resulting histograms with each other.


Box-Cox Transformed Histogram

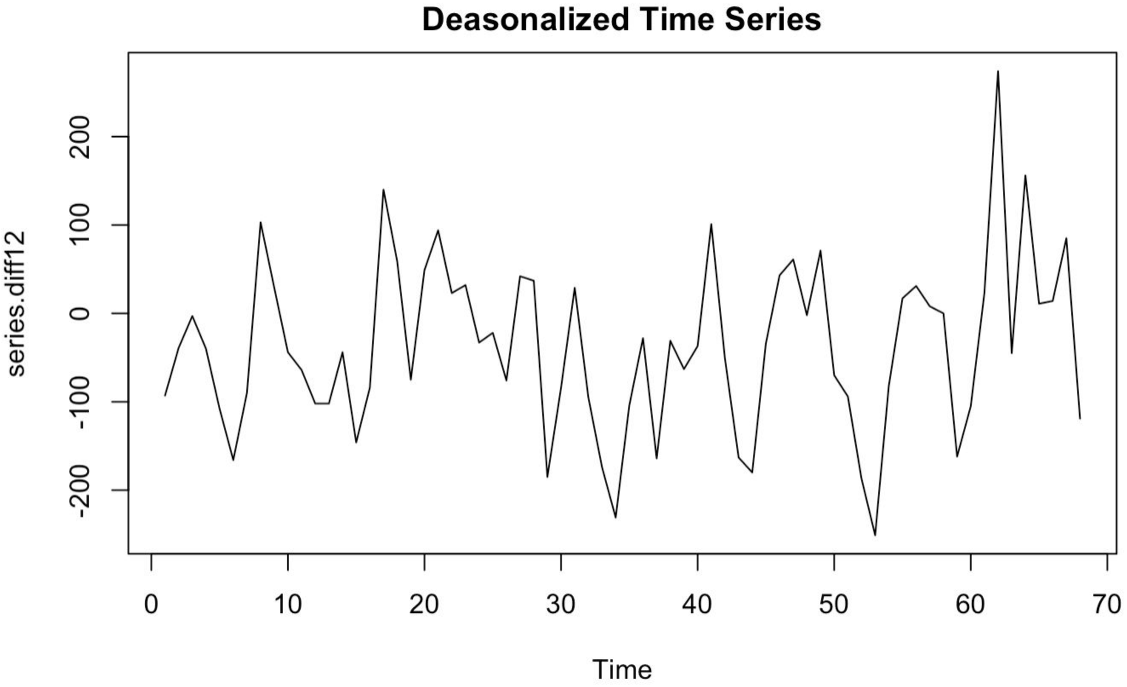

Log Transformed Histogram

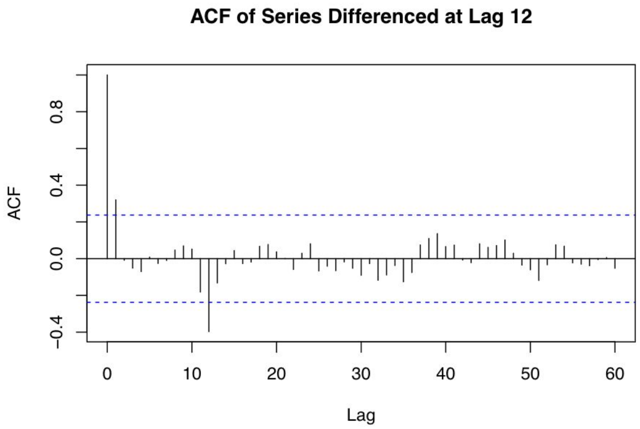

Square Root Transform Histogram

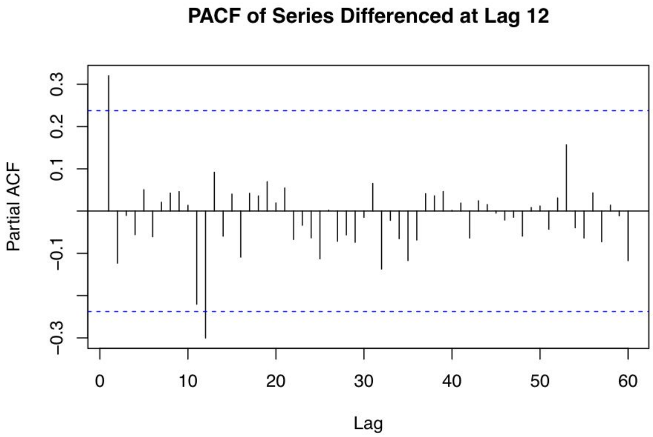
In comparison to the histogram of the original data , the log transformed histogram was visibly worse, and was immediately disregarded. Between the square root transform and the Box-Cox, the square root transformation histogram looked to be more visibly normal. At this point, I compared the winning transformation with the histogram of the original data, and although the square root transform did look better, both of the histograms seemed to be adequate. As a result, I compared the variance of the two. The original data had a variance of 23,824.16, and the square rooted data had a variance of 3.78. However, this was misleading because by square rooting the data, the smaller values would have a smaller variance no matter what. Consequently, I chose to stick with the original data.The next step was to see whether or not differencing was required, and upon plotting the ACF, it became clear that differencing was needed..



By looking at the ACF, there is a very clear sinusoidal pattern present, indicating seasonality of the data. To remedy this, the data was differenced at lag 12, since the data was monthly and the wave-like patterns of the data occurred in sets of 12. After differencing the data once, the resulting ACF and time series plot looked much more stationary, and with no real trend present in the data, differencing once was enough to achieve an adequate series from which I could estimate model parameters. The resulting time series plot, as well as the ACF and PACF from which I would select parameters are shown below.



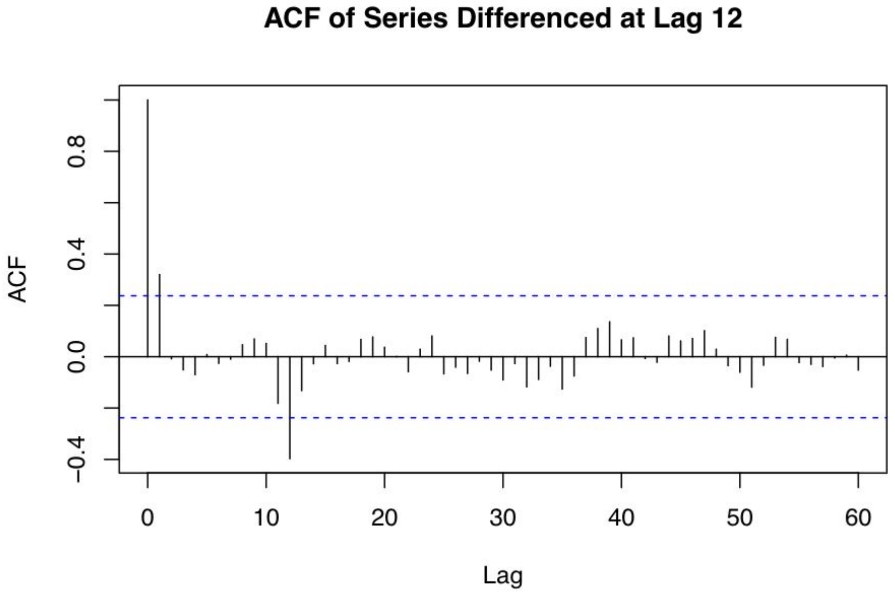


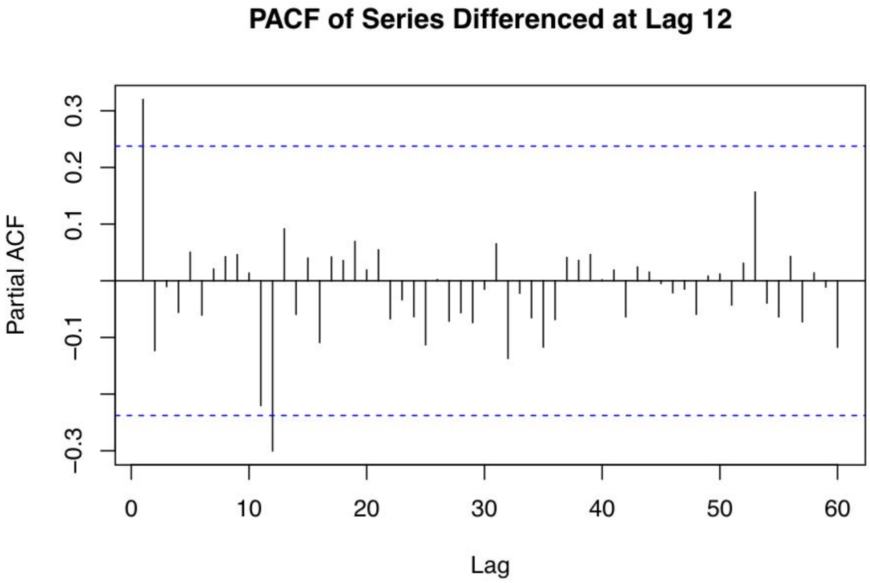


Looking at the time series now, it does appear to be more stationary than the original data. More importantly,the new ACF and PACF look much better, and are more in line with properties of a stationary process. With this now deseasonalized data, we proceed to the next stage of the report.

Model Identification

Using the ACF and PACF shown at the bottom of the previous page, I tried estimating potential parameters for my model. For ease of access they are posted below once again.

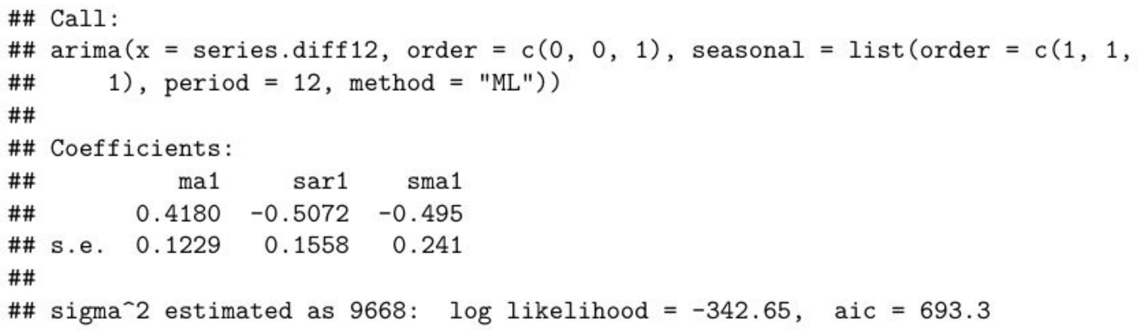




As mentioned earlier, due to the seasonal component of the original data, I was looking for the parameters of a SARIMA model. Since the series was differenced at lag 12, S =12 and D = 1. Since there was no other differencing done, d = 0. Looking at the ACF, we notice a seasonal moving average with a strong peak at h = 1, and so Q = 1. Along the same logic, the nonseasonal counterpart of q looked to be either q = 0 or q = 1. Looking at the PACF, the only “strong peak” present was at lag h = 1 as well, and so the seasonal autoregressive portion of the model indicated that a possible value for P was 1. By looking at the within year lags, due to the assumptions of within year lags and normal lags having the same behavior for SARIMA models, I thought possible values of p could be 0 or 1 as well.

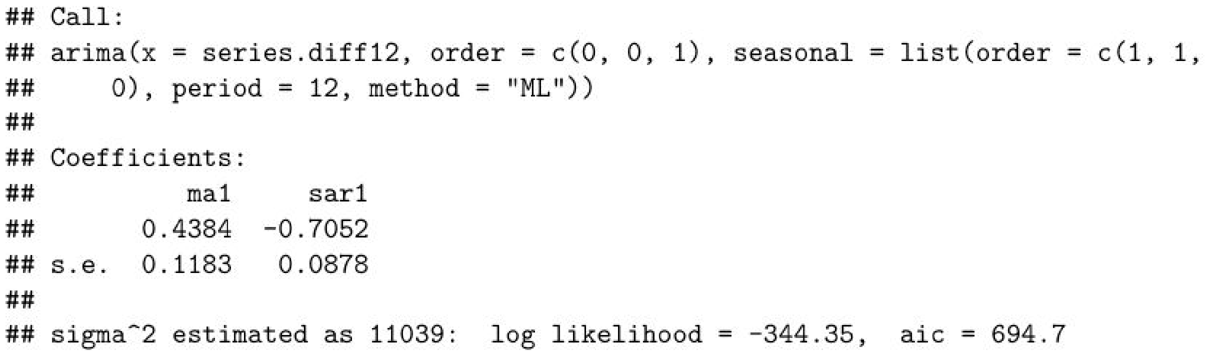
With this in mind, I now had a few potential candidate models to use. However, to include more possibilities, I also utilized the auto.arima() function to generate more potential models. By taking the possible models provided by the auto.arima() and also including the theoretical model estimations I had made from observing the ACF and PACF, I ended up with 14 possible candidates. All 14 models and their parameters are listed in the appendix, so as not to be inefficient. These models were then all compared by their AICc values, and from the list of AICc values I created, I chose the top two models which I would then take to diagnostic testing before choosing a final model to forecast with. These two models were the ones with the lowest AICcs of the bunch.

Within the context of the code, the top two models were fit.9(AICc 693.) and fit.5(AICc 694.7), which I took to the next step of the process: diagnostics. For future reference, I changed the name of fit.9 to model A and fit.5 to model B. Model A came out to be:



```
## Call:
## arima(x = series.diff12, order = c(0, 0, 1), seasonal = list(order = c(1, 1,
##     1), period = 12, method = "ML"))
##
## Coefficients:
##          ma1     sar1    sma1
##       0.4180  -0.5072  -0.495
## s.e.  0.1229   0.1558   0.241
##
## sigma^2 estimated as 9668:  log likelihood = -342.65,  aic = 693.3
```

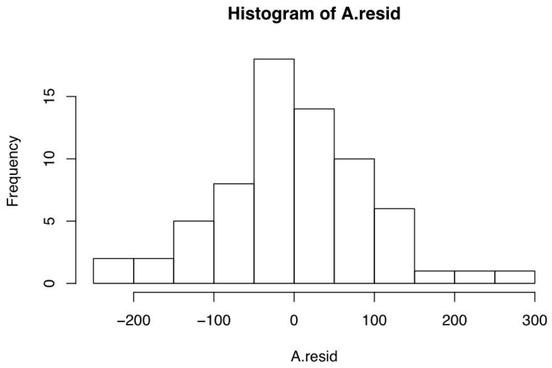
Model B is as follows:

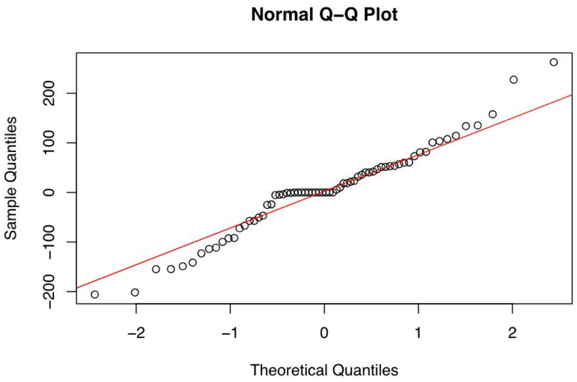


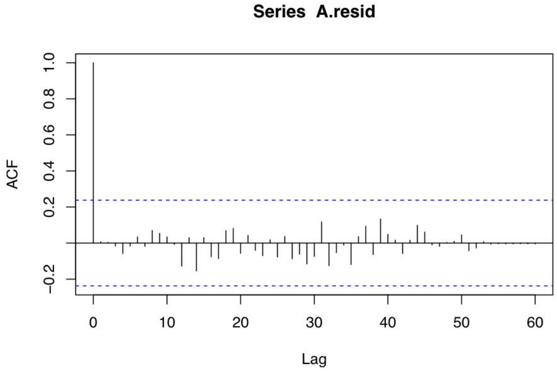
```
## Call:
## arima(x = series.diff12, order = c(0, 0, 1), seasonal = list(order = c(1, 1,
##     0), period = 12, method = "ML"))
##
## Coefficients:
##          ma1     sar1
##       0.4384  -0.7052
## s.e.  0.1183   0.0878
##
## sigma^2 estimated as 11039:  log likelihood = -344.35,  aic = 694.7
```

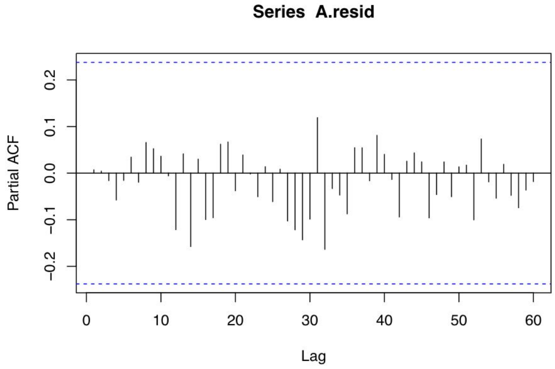
Diagnostic Testing

With model A and B now in hand, the next step was to perform diagnostic tests to determine which of the two would be my final model to proceed to forecast the data with. These diagnostic tests would be performed on the residuals of each of the models, and the respective plots were also drawn. For the sake of space not all of the plots will be in the main report, but can all be found in the appendix if needed. The plots I created consisted of histograms, a QQ plot, and the respective ACF/PACFS of each model's residuals. We will start with model A. The four aforementioned plots are shown below.

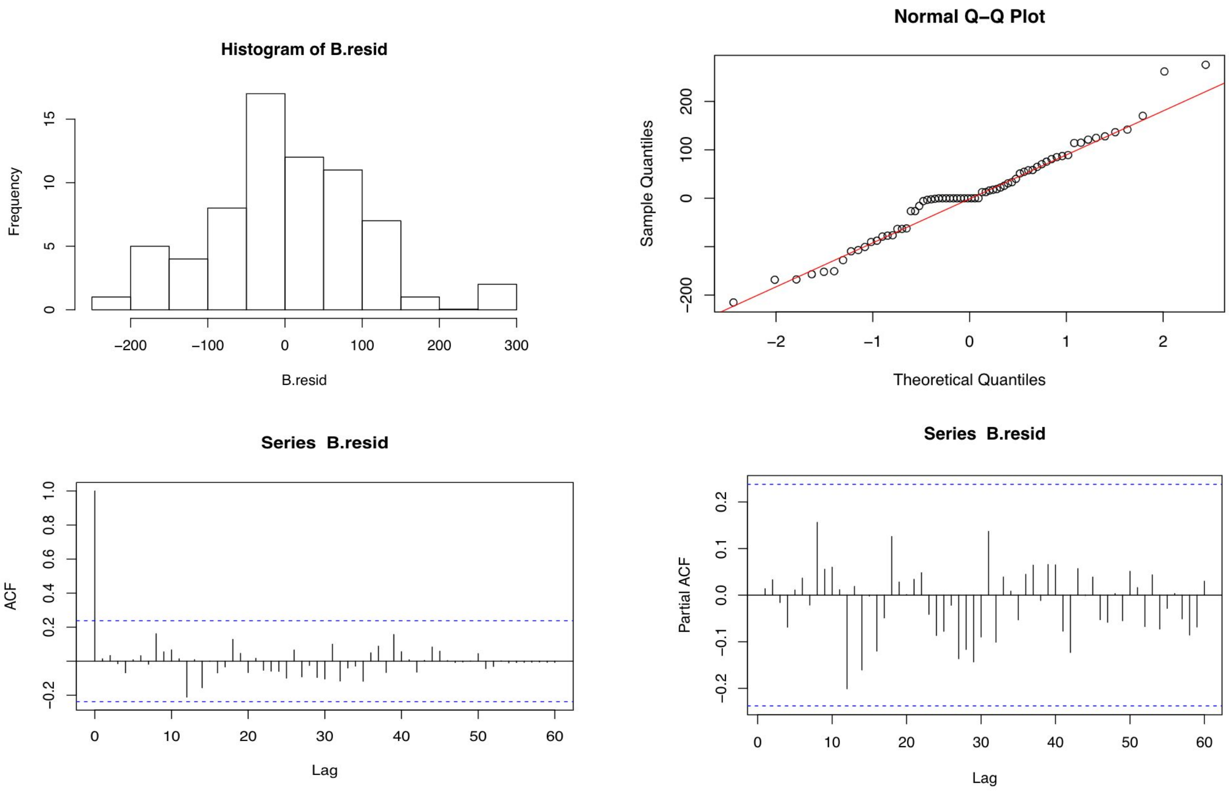




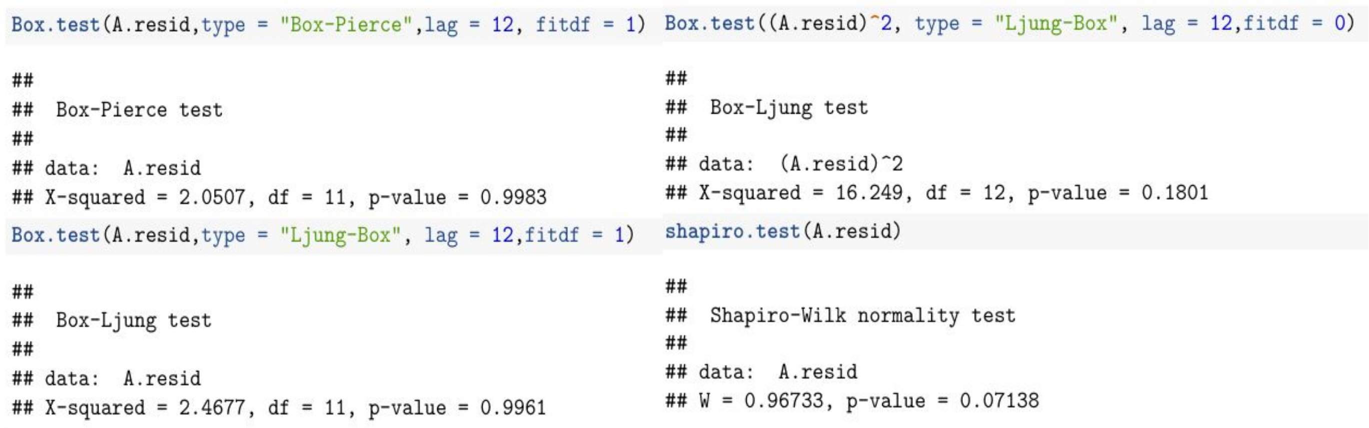




Both the ACF(left) and PACF(right) of the residuals of model A have no significant lags outside of the confidence interval of the plots, which was a good sign. In addition, the histogram did appear to display characteristics of being normal. However, the QQ plot was not nearly as straight as I would've liked. It was now that I realized that my original data set had an issue I didn't notice at first. With the data being monthly from 2010 to 2017, there still aren't enough observations for an ideal time series analysis. Furthermore, it is known in statistics that lower sample sizes lead to a smaller and smaller likelihood of a normal distribution. As a result, for both model A and model B, even though the ACF/PACFs might be ideal, one should take note of the fact that the lack of normality of the residuals is largely due to the inadequate observation size. However, there were still many positive aspects of my model, and so having said that I proceeded with the rest of the project, just with this observation in the back of my mind. With model A's plots overall being adequate, I now looked at model B.



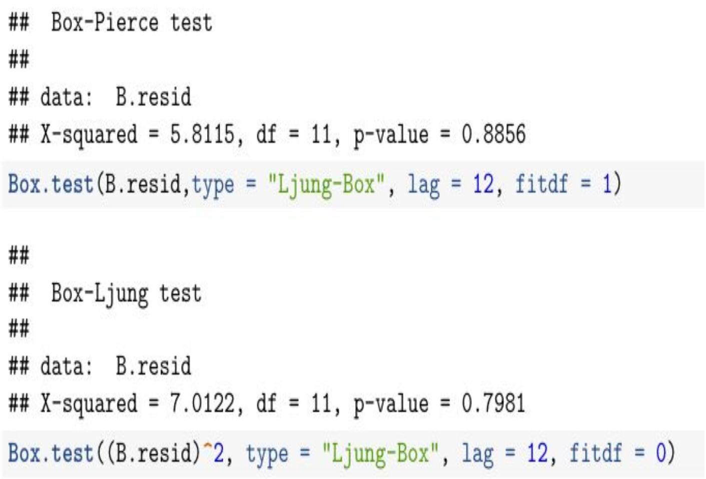
Once again, model B's residuals had no significant lags, which was a good sign. However, the histogram of model B was less favorable than model A's, with an overall less normal curve of the distribution. With both models passing through visual diagnostics, I then compared them using the portmanteau tests. These tests are the Box-Ljung, Box-Pierce, and Mcleod-Li tests. The Box-Ljung and Box-Pierce tests check for the hypothesis of white noise. Should the p-value of the two tests be above 0.05, then we would fail to reject the white noise hypothesis, which is what I was looking for. Similarly, the Mcleod-Li test checked for non-linear dependence of the residuals, with a p-value greater than 0.05 being desirable as well. In addition to the portmanteau tests, I also used the previously used Shapiro-Wilks test for normality. The results of the four tests for model A are below:



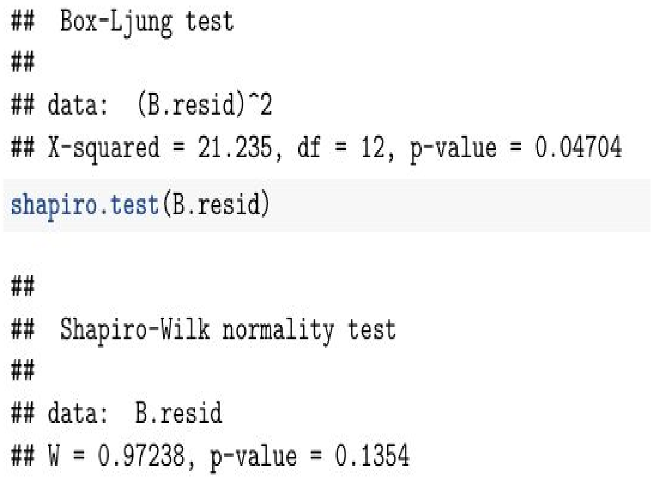
```
Box.test(A.resid,type = "Box-Pierce",lag = 12, fitdf = 1)

##
##  Box-Pierce test
##
## data:  A.resid
## X-squared = 2.0507, df = 11, p-value = 0.9983

Box.test(A.resid,type = "Ljung-Box", lag = 12,fitdf = 1)

##
##  Box-Ljung test
##
## data:  A.resid
## X-squared = 2.4677, df = 11, p-value = 0.9961
```

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

##
##  Box-Ljung test
##
## data:  (A.resid)^2
## X-squared = 16.249, df = 12, p-value = 0.1801

shapiro.test(A.resid)

##
##  Shapiro-Wilk normality test
##
## data:  A.resid
## W = 0.96733, p-value = 0.07138
```

With p-values greater than 0.05 for all 4 tests, model A passed the portmanteau tests. It was now time to check model B.



```
##  Box-Pierce test
##
## data:  B.resid
## X-squared = 5.8115, df = 11, p-value = 0.8856
Box.test(B.resid,type = "Ljung-Box", lag = 12, fitdf = 1)

##
##  Box-Ljung test
##
## data:  B.resid
## X-squared = 7.0122, df = 11, p-value = 0.7981
Box.test((B.resid)^2, type = "Ljung-Box", lag = 12, fitdf = 0)
```



```
##  Box-Ljung test
##
## data:  (B.resid)^2
## X-squared = 21.235, df = 12, p-value = 0.04704
shapiro.test(B.resid)

##
##  Shapiro-Wilk normality test
##
## data:  B.resid
## W = 0.97238, p-value = 0.1354
```

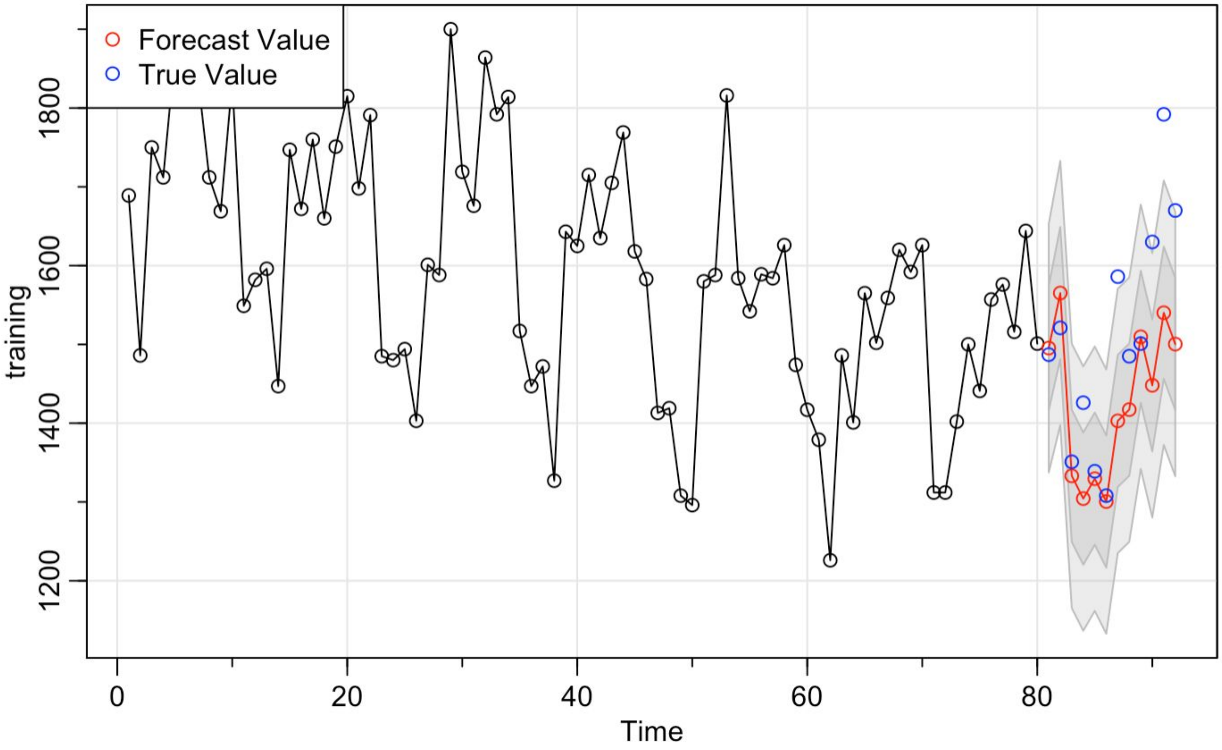
Model B ended up a p-value greater than 0.05 for only three out of the four tests. It did not pass the Mcleod-Li test for nonlinear dependence, and with all of that in mind, model A became the final model of choice. Model A was a SARIMA(0,0,1) x (1,1,1)[12], and I now would use it to forecast the data. The algebraic equation of the model is written below using the coefficients provided by the arima function:

$$X_t - 0.5072X_{t-12} = Z_t + 0.4180Z_{t-1} - 0.495Z_{t-12}$$

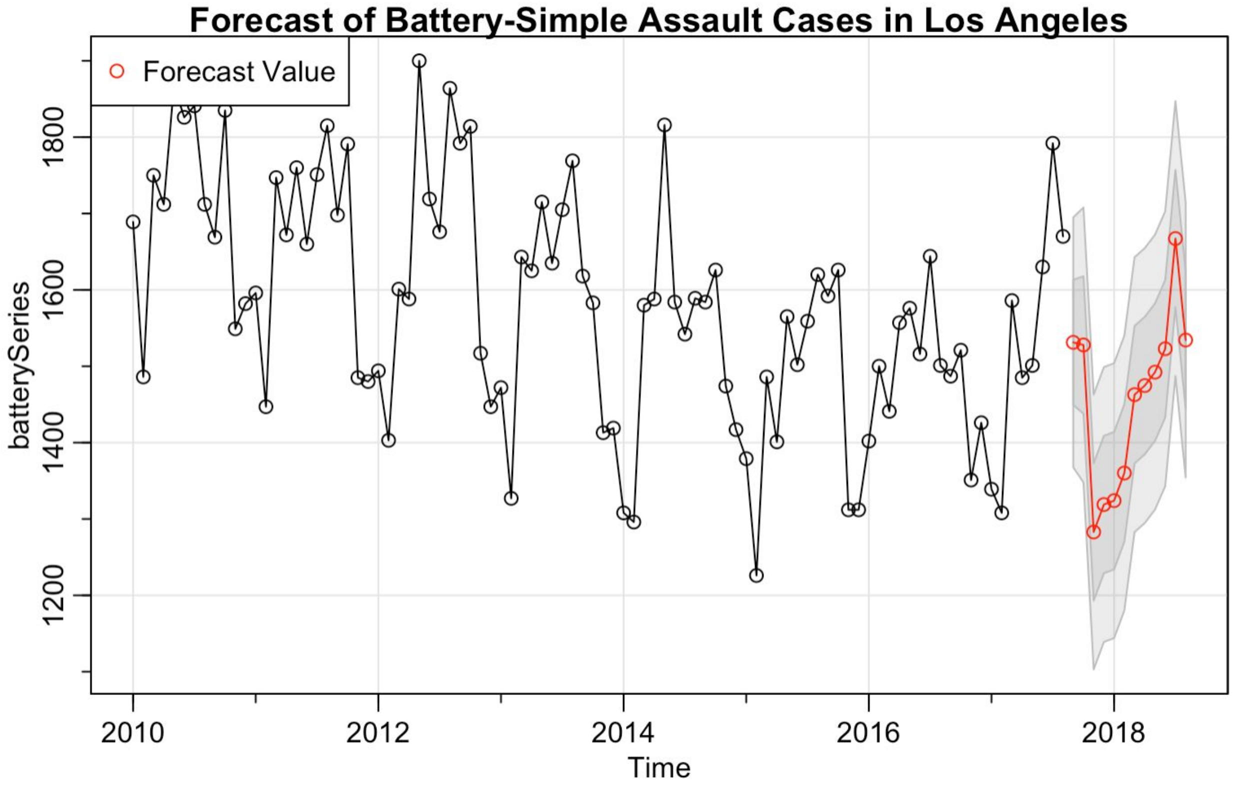
Given that model A passed all diagnostic tests, the fact that the visual plots could have been more normal with a larger observation size was not enough reason for me to turn back at this point. For future analyses I plan to make note of this much earlier on than I did for this dataset, so I won’t have to run into this issue so late in the process. With that reiterated, I moved to the last part of this report, which was using the chosen model to forecast the number of simple assault battery cases in LA in 2017, as well as beyond.

Forecasting

Using model A to forecast, I first aimed to forecast the data over a time period contained in the original data. As a result, I had split up the data into a training and testing set. The training set was from the start of 2010 to August of 2016, and the testing set was from September 2016 to September 2017. Using the sarima.for function, the forecast I achieved is shown below, along with the confidence interval of the prediction showing the range of potential values. These predictions were then included with the true data points taken from the testing set, so I could get a visual representation of the accuracy of the model.



Most of the prediction points in red fall within the confidence interval depicted by the gray region. However, a few of the true data points fall outside or on the edge of the interval, meaning it would theoretically be out of range of possible prediction values for this model. Since most of the predictions did fall within the interval, I still deemed this model to be suitable for this data. The last thing to do was to forecast the number of cases of simple assault for time that was not in the original data set. I chose to forecast a year ahead of the scope of this data, and the resulting plot is on the next page.



To my delight, the predictions for simple assault for the 12 months following this data set all fall within the corresponding confidence interval and seems to accurately reflect the previous patterns of assault cases over the past 7 years. As a result, I believe the accuracy of these predictions is actually rather potent, meaning the selected final model of choice adequately forecasted this data. Looking at the prediction values in the plot compared to the real values, there seems to be a spike in cases of this crime in the summer months. This was a surprise to me, as generally most people think of crimes to happen when it is dark, and summer days have the longest amount of sunlight than any other time of year. With all of this in mind, I move to concluding remarks.

Conclusion

In conclusion, the application of the Box Jenkins Methods in time series analysis was able to successfully predict the number of Battery-Simple Assault cases per month in Los Angeles. However, the implications of my results contradicted what I originally expected to see within this data. I expected to see higher rates of crime around winter due to it being dark for longer, but it turns out that summer held the highest rates of this crime. This actually makes sense in the context that overall less people are working like they normally do, meaning a higher possible chance for crimes to occur, especially with students out of school for summer break.

In terms of the model itself, the various visual and numeric tests and processes I used to reach my final model all showed positive signs throughout the course of this project. I was able to achieve a stationary process by differencing the data to remove the seasonality component, and from there I was able to test a handful of candidate models through application of my theoretical knowledge as well as through the auto.arima() function built in R. The big, glaring issue with this project is that I did not have enough observations in total. This resulted in my model indicating positive attributes, yet almost consistently not being as “normal” as I would hope for in an ideal model. As mentioned multiple times before, normality is heavily affected by sample size, and since the other tests I used all indicated a strong model, I believe this model could reach its full potency with an adequate number of observations. For future reference, I could split up the data by weeks instead and train my model over a shorter number of years, or I could also try to find the data for the past 2.5 years that have passed since 2017 to stay with my monthly framework. With all that being said, the final SARIMA model for this project was shown to be:

$$X_t - 0.5072X_{t-12} = Z_t + 0.4180Z_{t-1} - 0.495Z_{t-12}$$

I hope to one day revisit this dataset and act on the changes I mentioned earlier in this conclusion. With all that’s happening in the country right now, analyzing crime rates for public safety seems to be a top priority, and using tools such as the ones utilized in this report, I believe more insightful conclusions can be made, potentially to be used in real life applications.

APPENDIX ON THE NEXT PAGE AND BEYOND

Forecasting Battery Assault Crime Rates in LA: Appendix

Nihaal Kumar

```
library(tidyverse)
library(lubridate)
library(DataExplorer)
library(dplyr)
library(tseries)
library(forecast)
library(MASS)
library(dse)
library(qpcR)
```

```
## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display

## Warning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'.
```

```
library(astsa)
```

Loading and Cleaning the Data:

```
crimes <- read.csv('Crime_Data_2010_2017.csv',header = TRUE)
```

```
names(crimes)
```

```
##  [1] "DR.Number"              "Date.Reported"          "Date.Occurred"
##  [4] "Time.Occurred"          "Area.ID"                "Area.Name"
##  [7] "Reporting.District"     "Crime.Code"             "Crime.Code.Description"
## [10] "MO.Codes"               "Victim.Age"             "Victim.Sex"
## [13] "Victim.Descent"         "Premise.Code"           "Premise.Description"
## [16] "Weapon.Used.Code"       "Weapon.Description"     "Status.Code"
## [19] "Status.Description"     "Crime.Code.1"           "Crime.Code.2"
## [22] "Crime.Code.3"           "Crime.Code.4"           "Address"
## [25] "Cross.Street"           "Location"
```

```
crimes %>% group_by(Crime.Code.Description) %>% summarize(Count_Incident = n()) %>% arrange(desc(Count_
```

```
## # A tibble: 135 x 2
##    Crime.Code.Description                                   Count_Incident
##    <fct>                                                             <int>
##  1 BATTERY - SIMPLE ASSAULT                                         145767
##  2 VEHICLE - STOLEN                                                 121329
##  3 BURGLARY FROM VEHICLE                                            121318
##  4 BURGLARY                                                         114751
##  5 THEFT PLAIN - PETTY ($950 & UNDER)                               113709
##  6 THEFT OF IDENTITY                                                100653
##  7 INTIMATE PARTNER - SIMPLE ASSAULT                                 85908
##  8 VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS) 0114      79433
##  9 VANDALISM - MISDEAMEANOR ($399 OR UNDER)                          71523
## 10 ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT                    67631
## # ... with 125 more rows
```

```
BatteryCases <- filter(crimes, Crime.Code.Description == "BATTERY - SIMPLE ASSAULT")

BatteryCases <- dplyr::select(BatteryCases, c(Crime.Code.Description,Date.Occurred))

head(BatteryCases)
```

```
##     Crime.Code.Description Date.Occurred
## 1 BATTERY - SIMPLE ASSAULT    01/02/2013
## 2 BATTERY - SIMPLE ASSAULT    01/02/2013
## 3 BATTERY - SIMPLE ASSAULT    01/05/2013
## 4 BATTERY - SIMPLE ASSAULT    01/06/2013
## 5 BATTERY - SIMPLE ASSAULT    01/20/2013
## 6 BATTERY - SIMPLE ASSAULT    02/09/2013
```

```
colnames(BatteryCases)
```

```
## [1] "Crime.Code.Description" "Date.Occurred"
```

```
BatteryCases = BatteryCases %>% rename(Date = "Date.Occurred") %>% rename(CrimeType = "Crime.Code.Descr
colnames(BatteryCases)
```

```
## [1] "CrimeType" "Date"
```

```
BatteryCases$Date <- as.Date(BatteryCases$Date, "%m/%d/%Y")
```

```
str(BatteryCases)
```

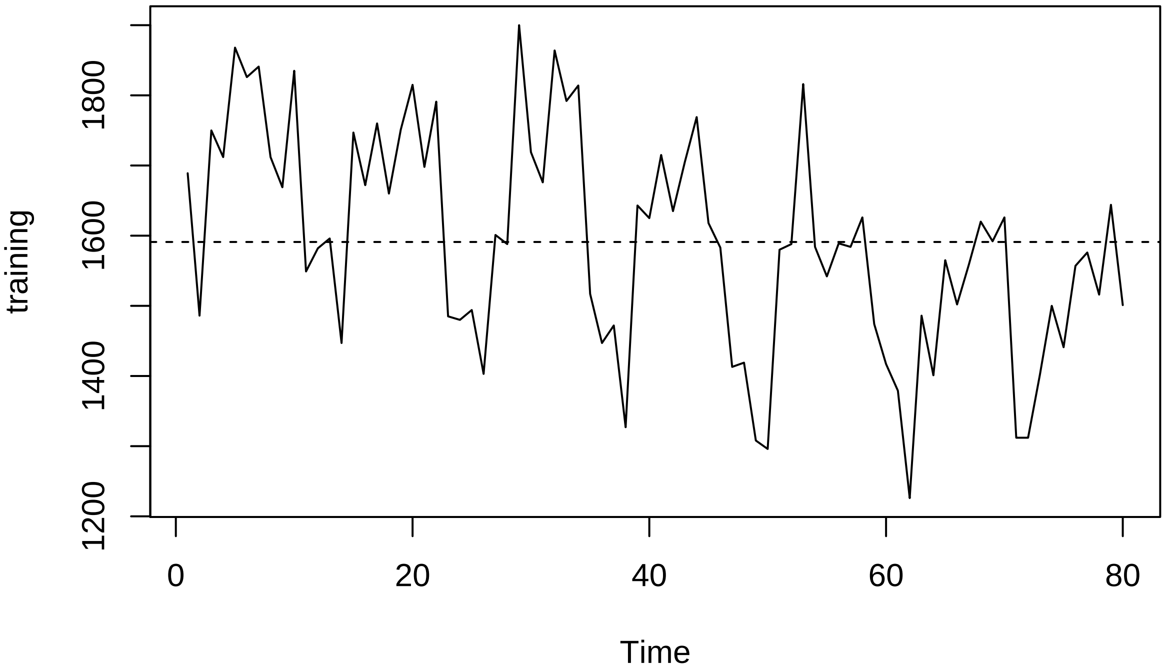
```
## 'data.frame':    145767 obs. of  2 variables:
##  $ CrimeType: Factor w/ 135 levels "","ABORTION/ILLEGAL",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ Date     : Date, format: "2013-01-02" "2013-01-02" ...
```

```
series_data <- BatteryCases %>%
  mutate(year_month = make_date(year = year(Date),month = month(Date)))%>%
  filter(!is.na(year_month))%>%
  filter(year_month != "2017-09-01")%>%
  group_by(year_month) %>%
  summarize(crime_count = n())

series_data <- series_data$crime_count
```

```
batterySeries <- ts(series_data,start = c(2010,1),freq = 12)
training <- batterySeries[1:80]
testing  <- batterySeries[81:92]
```

Beginning of Preliminary Analysis

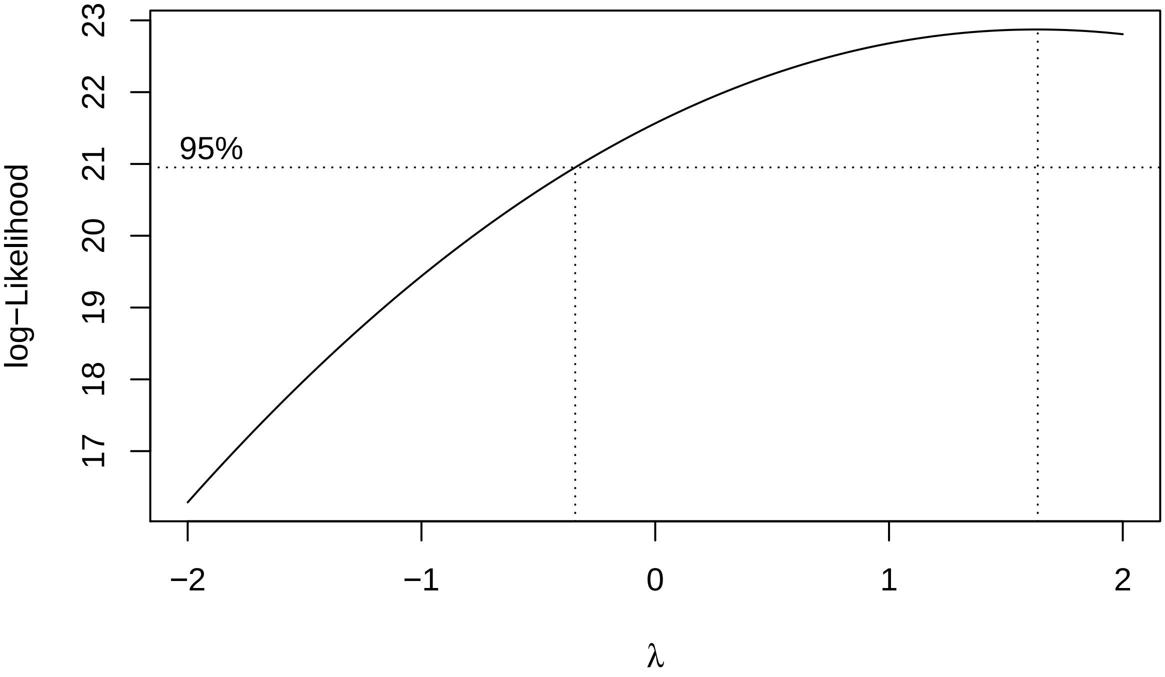
```
plot.ts(training)
abline(h = mean(training),lty = 2)
```



```r
shapiro.test(training)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  training
## W = 0.98619, p-value = 0.547
```

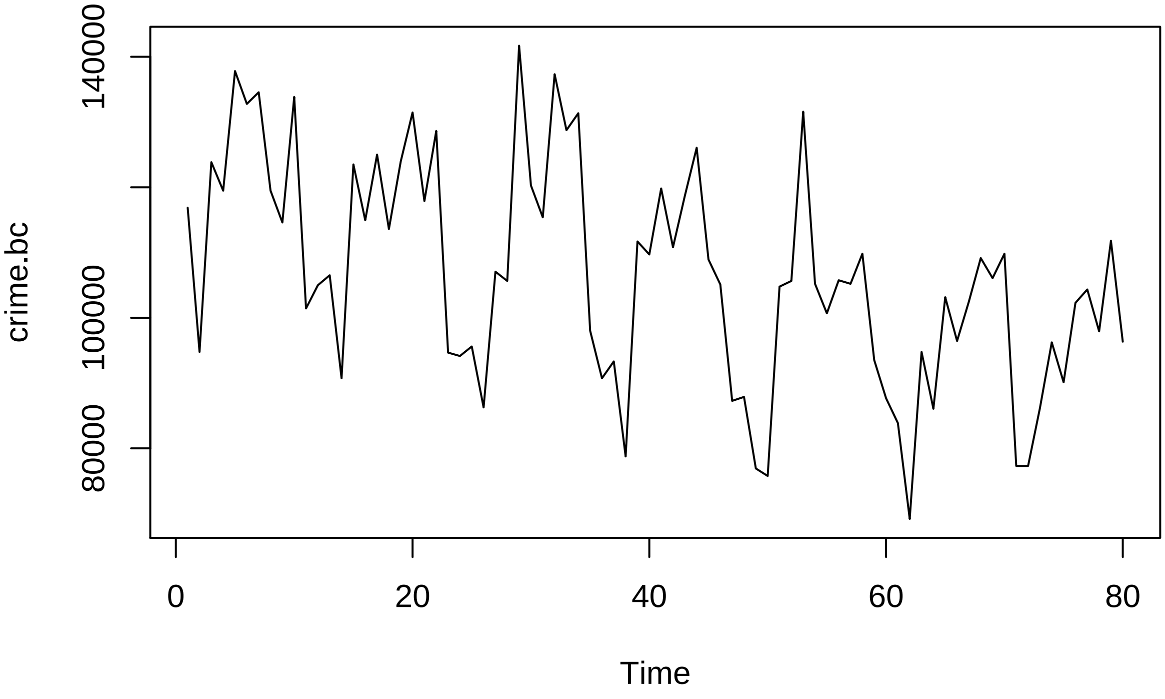
```r
#box cox test
t = 1:length(training)
bc.transform = boxcox(training~t, plotit = TRUE)
```



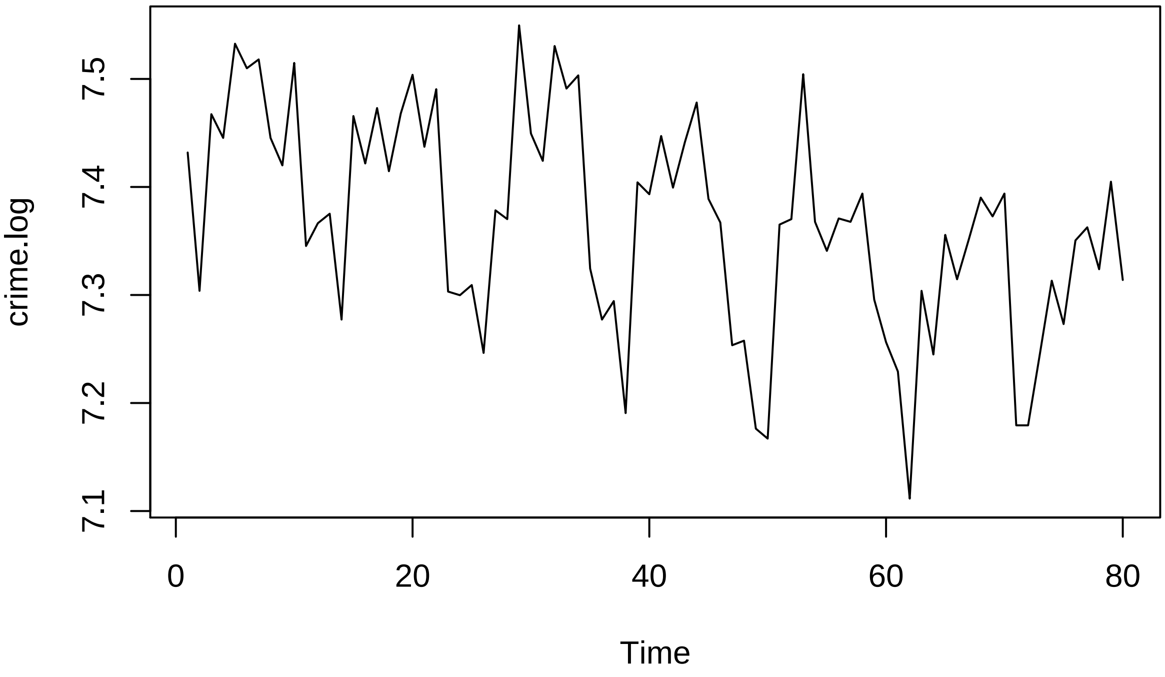
```
lambda = bc.transform$x[which(bc.transform$y == max(bc.transform$y))]
lambda
```

```
## [1] 1.636364
```

```
crime.bc = (1/lambda) * ((training^lambda)-1)
crime.log = log(training)
plot.ts(crime.bc)
```

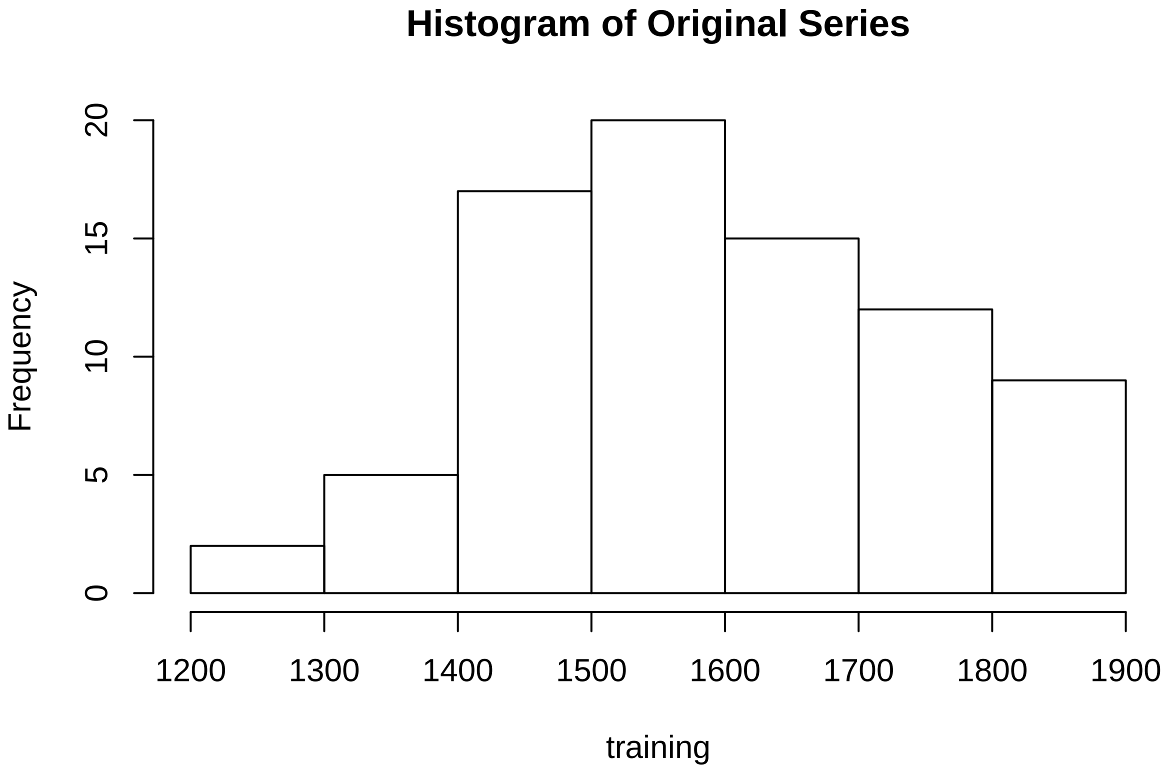


```
plot.ts(crime.log)
```

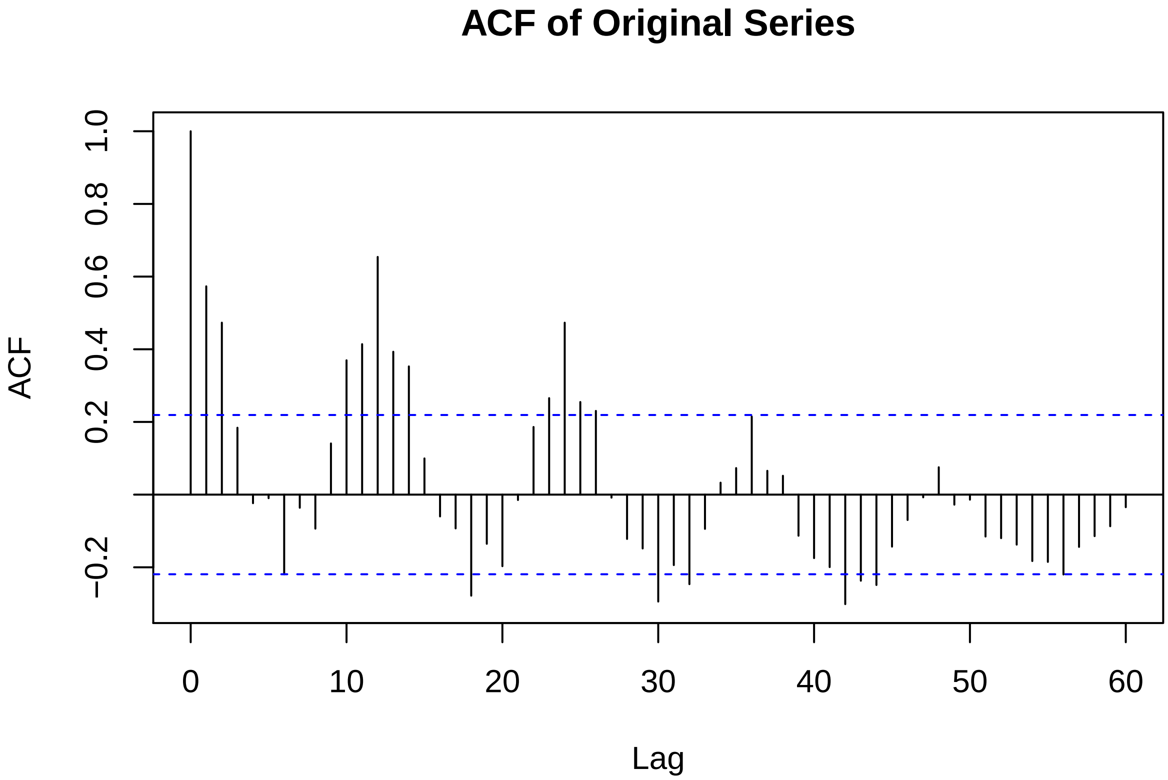


Visual Plots of Unchanged Time Series Data

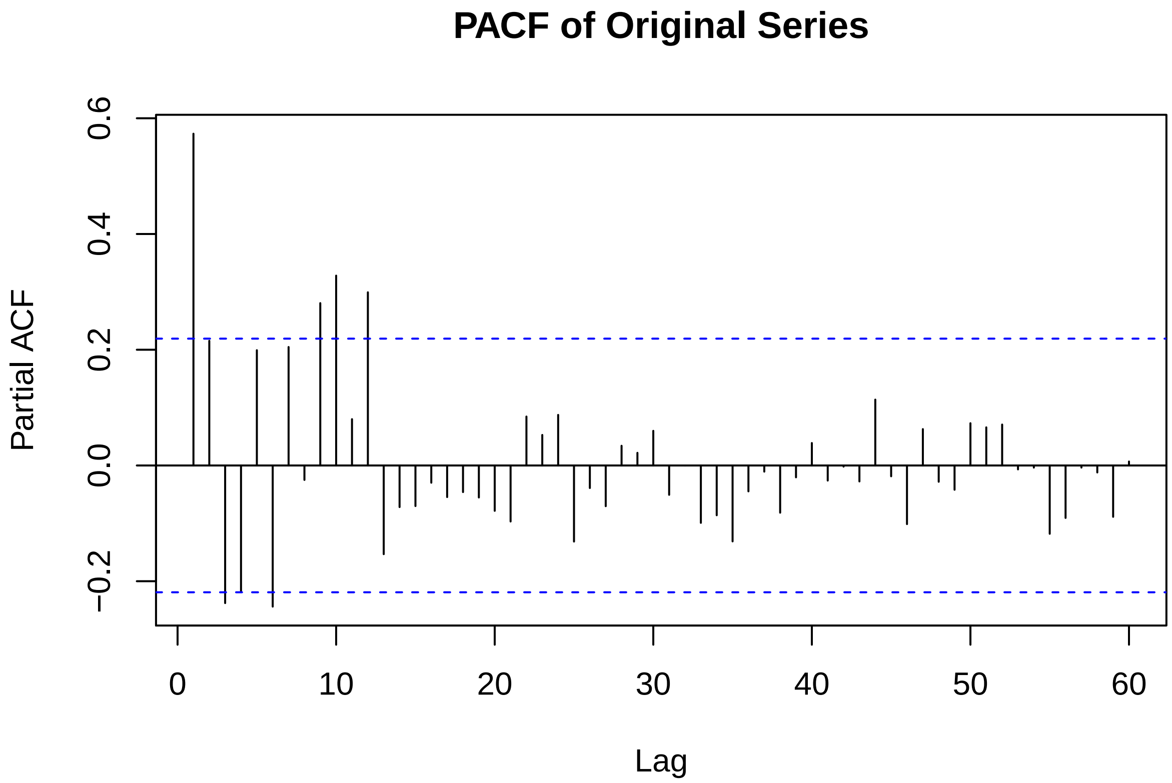
```r
hist(training,main = "Histogram of Original Series")
```



```r
acf(training,lag.max = 60,main ="ACF of Original Series")
```

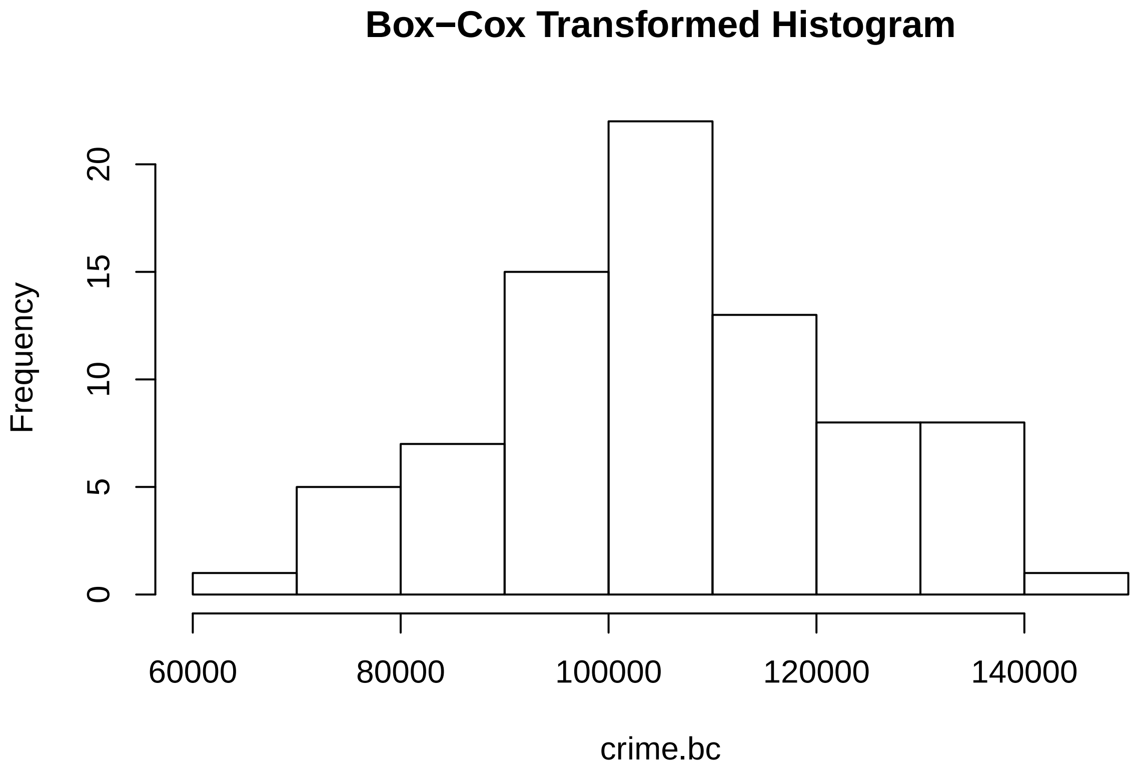


```
pacf(training,lag.max = 60, main =  "PACF of Original Series")
```

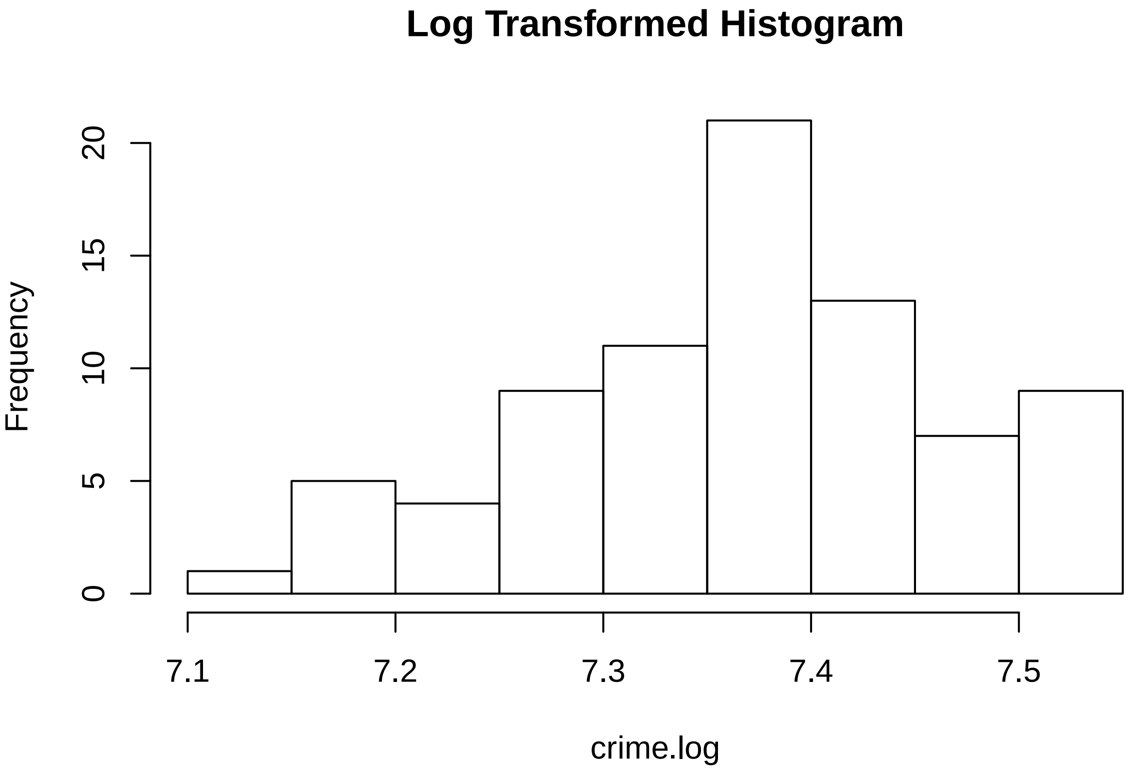


Histograms of Potential Transformations:

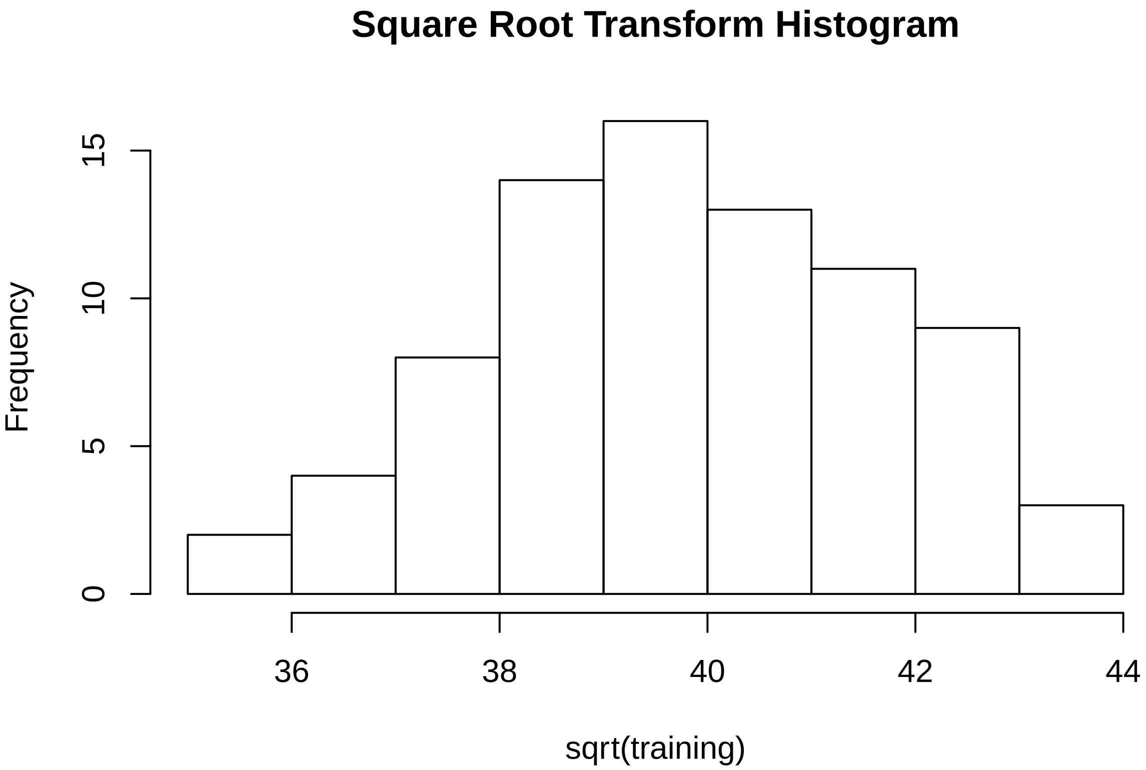
```
hist(crime.bc,main = "Box-Cox Transformed Histogram")
```



```r
hist(crime.log, main = "Log Transformed Histogram")
```



```r
hist(sqrt(training),main = "Square Root Transform Histogram")
```



Comparison of Variance

```r
var(training)
```

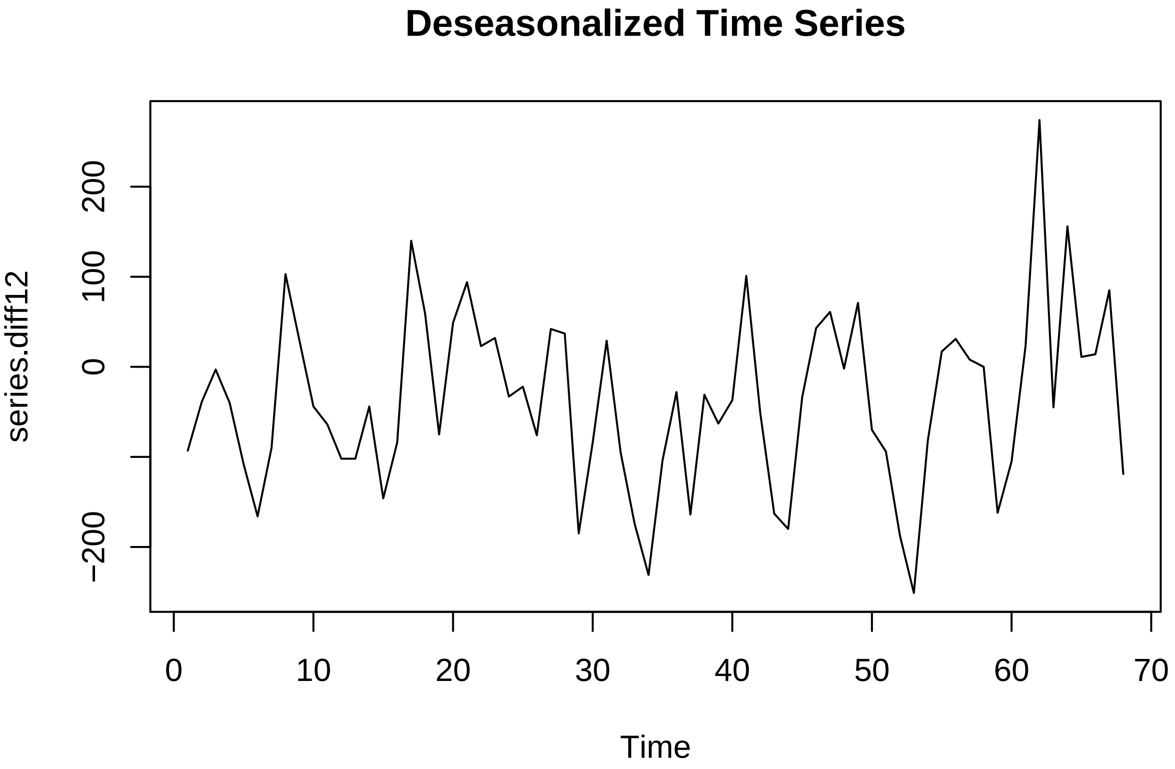
```
## [1] 23824.16
```

```r
var(sqrt(training))
```

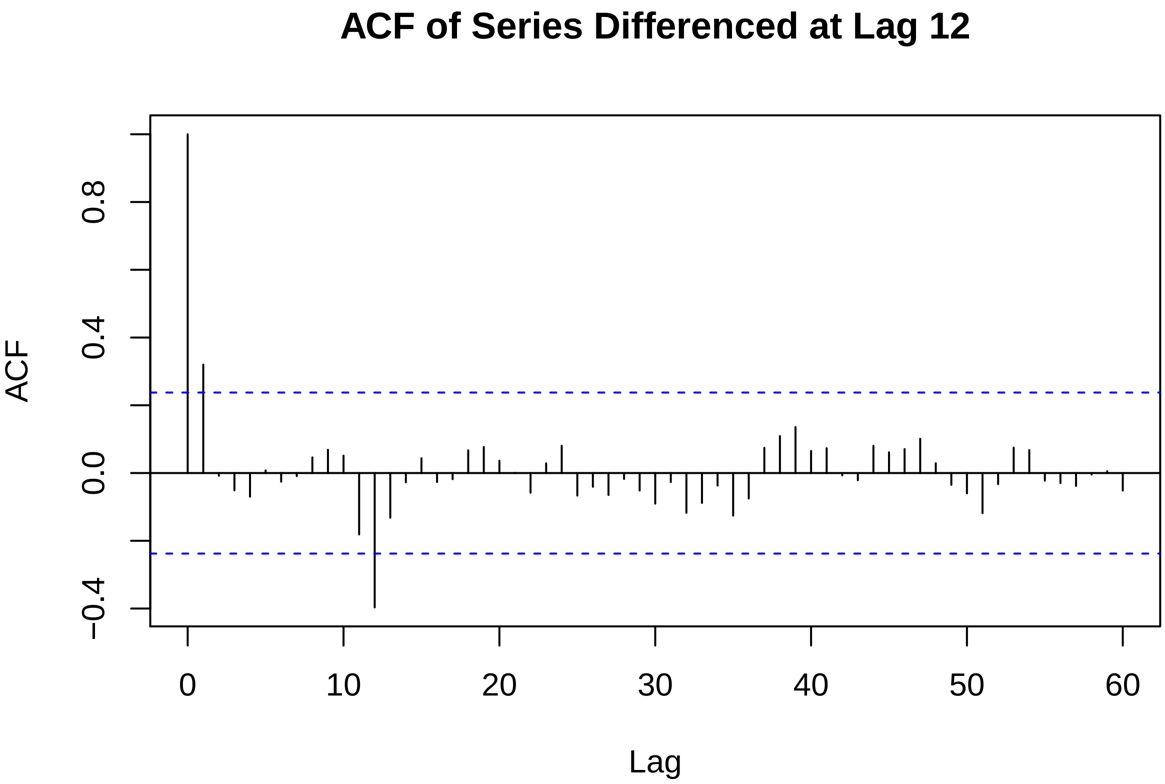
```
## [1] 3.785055
```

Differencing to Remove Seasonality

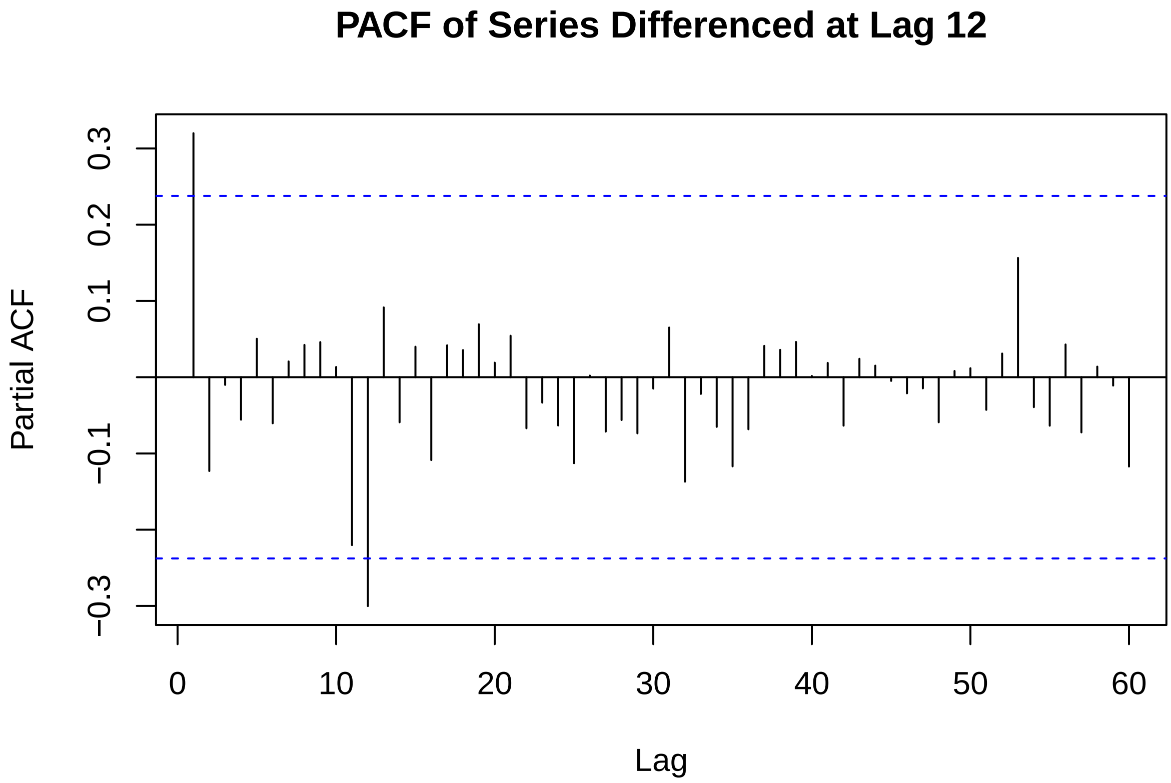
```r
series.diff12<- diff(training,lag = 12)
plot.ts(series.diff12, main = "Deseasonalized Time Series")
```



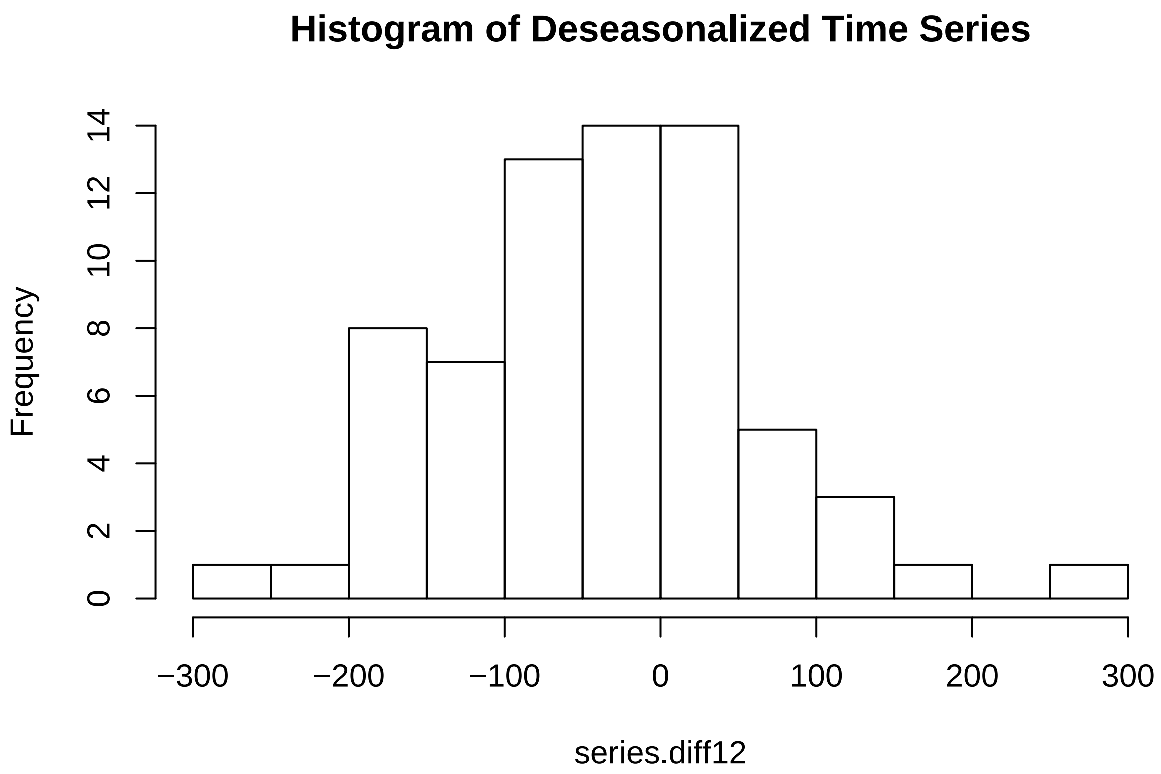
```r
acf(series.diff12, lag.max = 60, main = "ACF of Series Differenced at Lag 12")
```



```r
pacf(series.diff12,lag.max = 60,main = "PACF of Series Differenced at Lag 12")
```



```r
hist(series.diff12,main =  "Histogram of Deseasonalized Time Series")
```



More Potential Models with auto.arima()

```
par(mfrow=c(2,1))
auto.arima(series.diff12, test="kpss", ic="aic", trace = T)
```

```
##
##  ARIMA(2,0,2) with non-zero mean : Inf
##  ARIMA(0,0,0) with non-zero mean : 817.9343
##  ARIMA(1,0,0) with non-zero mean : 812.5718
##  ARIMA(0,0,1) with non-zero mean : 811.4062
##  ARIMA(0,0,0) with zero mean     : 825.4932
##  ARIMA(1,0,1) with non-zero mean : 813.4038
##  ARIMA(0,0,2) with non-zero mean : 813.4038
##  ARIMA(1,0,2) with non-zero mean : 815.4009
##  ARIMA(0,0,1) with zero mean     : 815.4783
##
##  Best model: ARIMA(0,0,1) with non-zero mean
```

```
## Series: series.diff12
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1      mean
##       0.3711  -38.2671
## s.e.  0.1160   14.9422
##
## sigma^2 estimated as 8383:  log likelihood=-402.7
## AIC=811.41   AICc=811.78   BIC=818.06
```

Total List of Possible Models

```
fit.1  <- arima(series.diff12, c(1,0,0), seasonal = list(order = c(1,1,0), period = 12, method = "ML"))
fit.2  <- arima(series.diff12, c(0,0,0), seasonal = list(order = c(1,1,0), period = 12, method = "ML"))
fit.3  <- arima(series.diff12, c(1,0,1), seasonal = list(order = c(1,1,0), period = 12, method = "ML"))
fit.4  <- arima(series.diff12, c(0,0,1), seasonal = list(order = c(1,0,0), period = 12, method = "ML"))
fit.5  <- arima(series.diff12, c(0,0,1), seasonal = list(order = c(1,1,0), period = 12, method = "ML"))
fit.6  <- arima(series.diff12, c(1,0,0), seasonal = list(order = c(1,1,1), period = 12, method = "ML"))
fit.7  <- arima(series.diff12, c(0,0,0), seasonal = list(order = c(1,1,1), period = 12, method = "ML"))
fit.8  <- arima(series.diff12, c(0,0,1), seasonal = list(order = c(1,0,1), period = 12, method = "ML"))
fit.9  <- arima(series.diff12, c(0,0,1), seasonal = list(order = c(1,1,1), period = 12, method = "ML"))
fit.10 <- arima(series.diff12, c(2,0,0), seasonal = list(order = c(1,1,1), period = 12, method = "ML"))
fit.11 <- arima(series.diff12, c(2,0,1), seasonal = list(order = c(1,1,1), period = 12, method = "ML"))
fit.12 <- arima(series.diff12, c(2,0,0), seasonal = list(order = c(1,1,0), period = 12, method = "ML"))
fit.13 <- arima(series.diff12, c(0,0,2), seasonal = list(order = c(1,1,0), period = 12, method = "ML"))
fit.14 <- arima(series.diff12, c(0,0,2), seasonal = list(order = c(1,1,1), period = 12, method = "ML"))
```

```
AICc_List <- c(AICc(fit.1),AICc(fit.2),AICc(fit.3),AICc(fit.4),AICc(fit.5),AICc(fit.6),AICc(fit.7),AICc
```

AICc List From which two best were selected

```
AICc_List
```

```
##  [1] 695.9704 703.1745 697.0164 799.1587 694.8879 694.9515 700.2900 796.5969
##  [9] 693.6766 696.0884 698.2639 697.1466 697.0234 695.9347
```

Based on the AICc values, the two potential models to proceed with to compare with diagnostic checking are fit.9, and fit.5, which havve the lowest and second lowest AICc values. We will rename these to model A and model B for reason of comparison before proceeding.

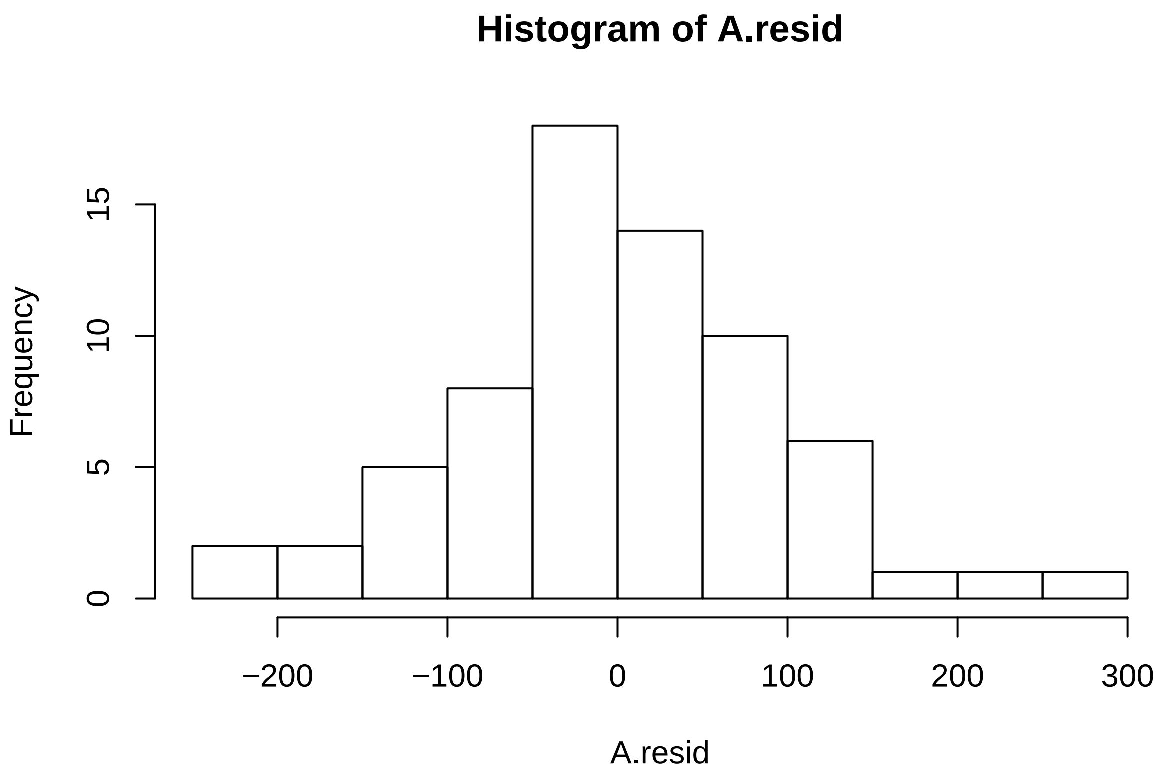
```
A <- fit.9
B <-fit.5
```

```
A
```

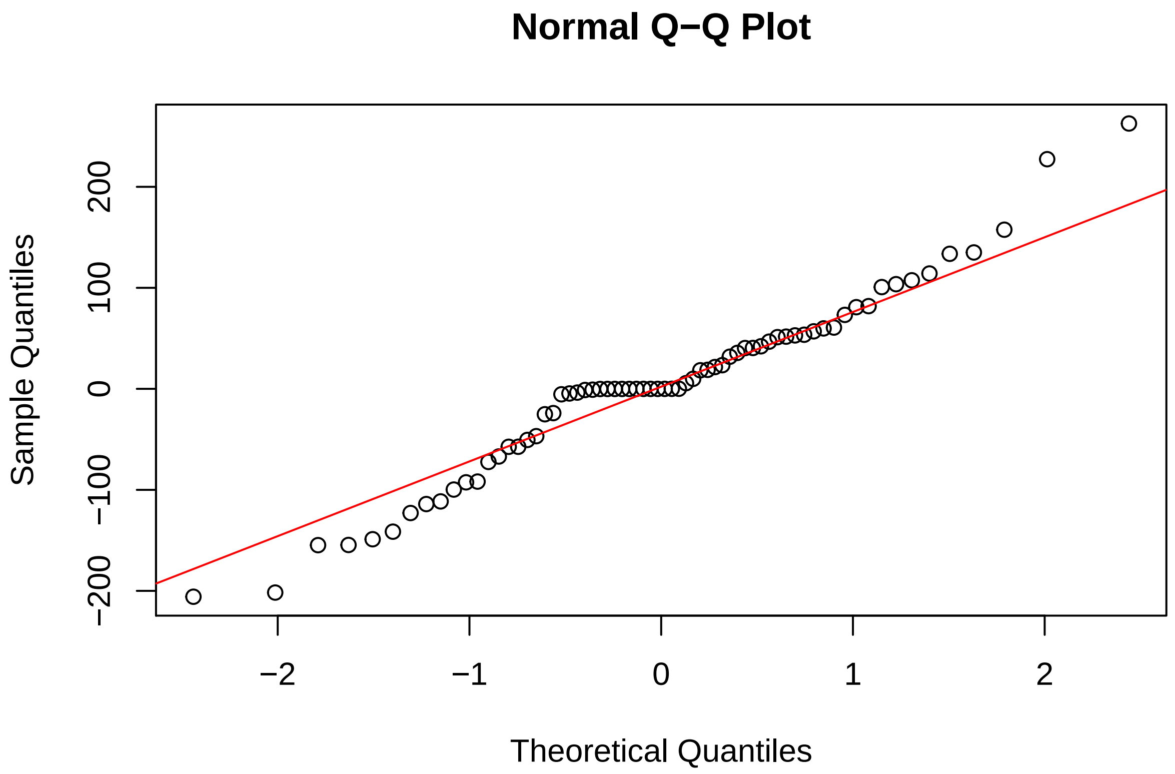
```
##
## Call:
## arima(x = series.diff12, order = c(0, 0, 1), seasonal = list(order = c(1, 1,
##     1), period = 12, method = "ML"))
##
## Coefficients:
##          ma1     sar1    sma1
##       0.4180  -0.5072  -0.495
## s.e.  0.1229   0.1558   0.241
##
## sigma^2 estimated as 9668:  log likelihood = -342.65,  aic = 693.3
```

Model A Diagnostics

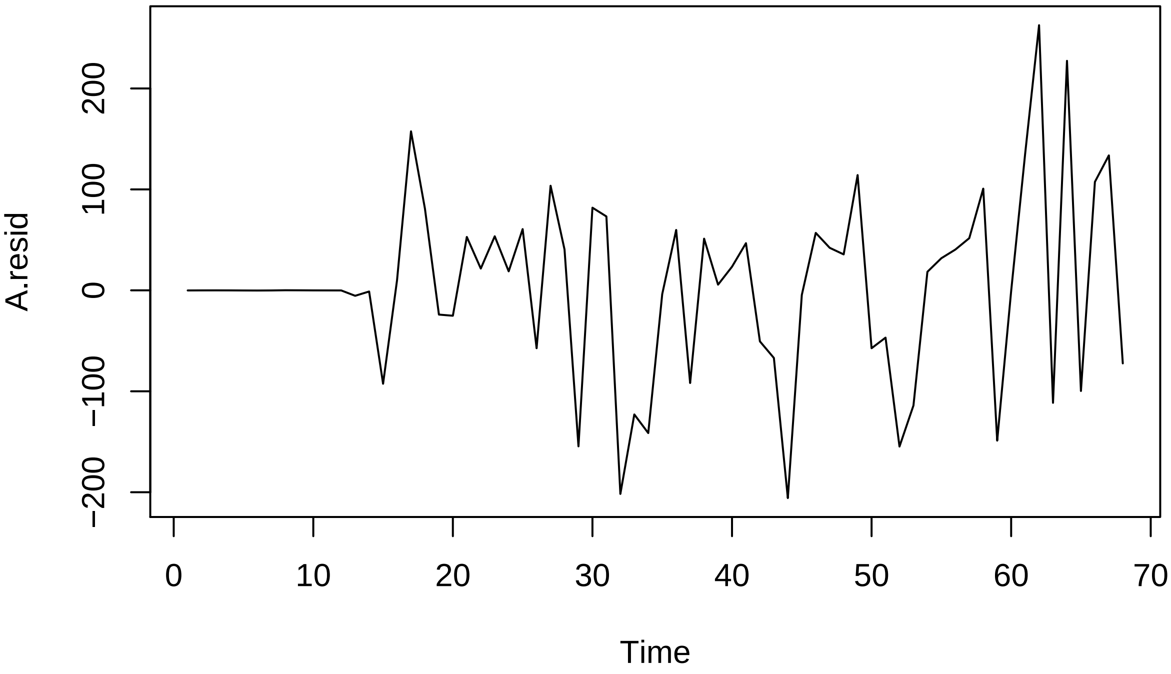
```
A.resid <- residuals(A)
hist(A.resid)
```



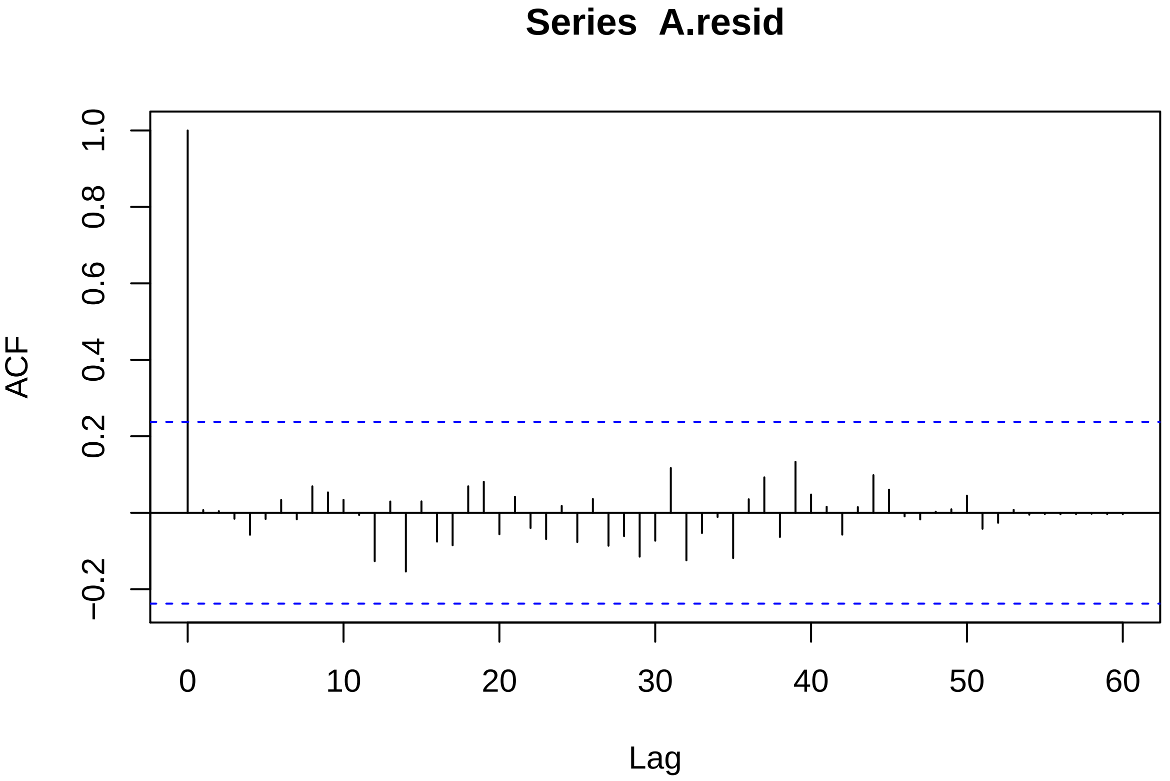
```
qqnorm(A.resid)
qqline(A.resid,col = "red")
```



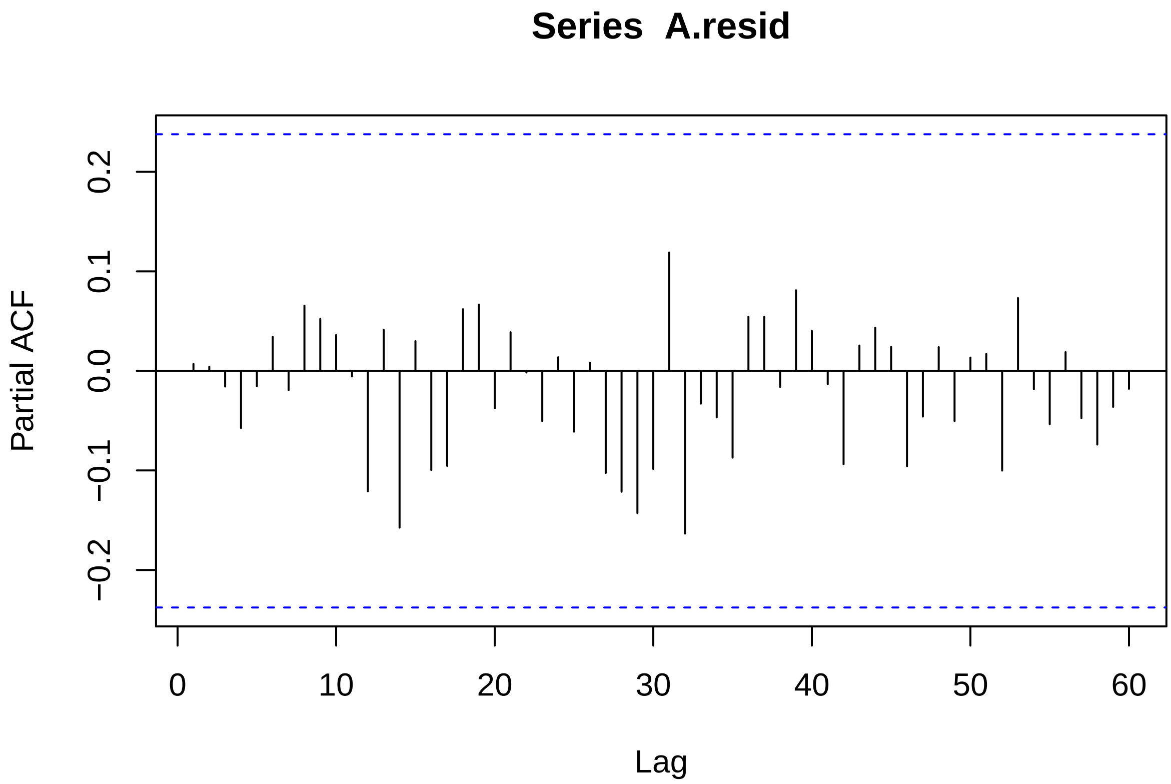
```
plot.ts(A.resid)
```



```r
acf(A.resid,lag.max = 60)
```



```r
pacf(A.resid,lag.max = 60)
```



Model A Portmanteau Tests

```r
Box.test(A.resid,type = "Box-Pierce",lag = 12, fitdf = 1)
```

```
##
##  Box-Pierce test
##
## data:  A.resid
## X-squared = 2.0507, df = 11, p-value = 0.9983
```

```r
Box.test(A.resid,type = "Ljung-Box", lag = 12,fitdf = 1)
```

```
##
##  Box-Ljung test
##
## data:  A.resid
## X-squared = 2.4677, df = 11, p-value = 0.9961
```

```r
Box.test((A.resid)^2, type = "Ljung-Box", lag = 12,fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  (A.resid)^2
## X-squared = 16.249, df = 12, p-value = 0.1801
```

```r
shapiro.test(A.resid)
```

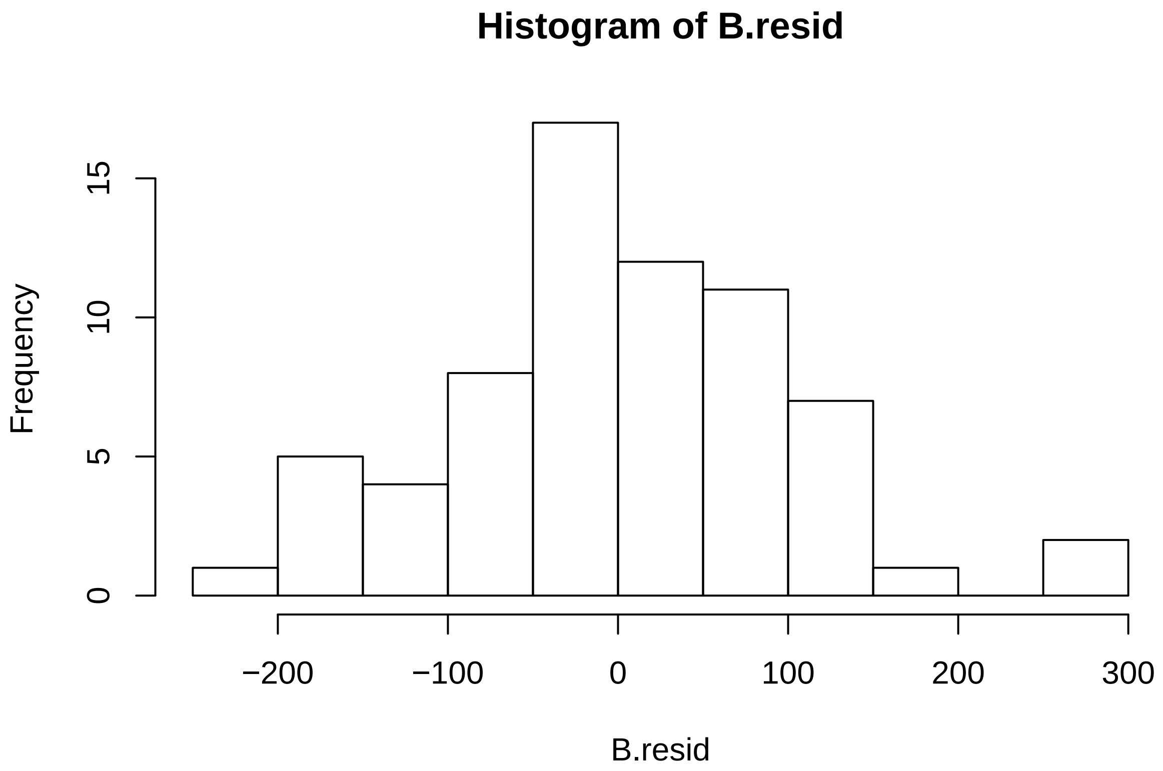
```
##
##  Shapiro-Wilk normality test
##
## data:  A.resid
## W = 0.96733, p-value = 0.07138
```

```
B
```

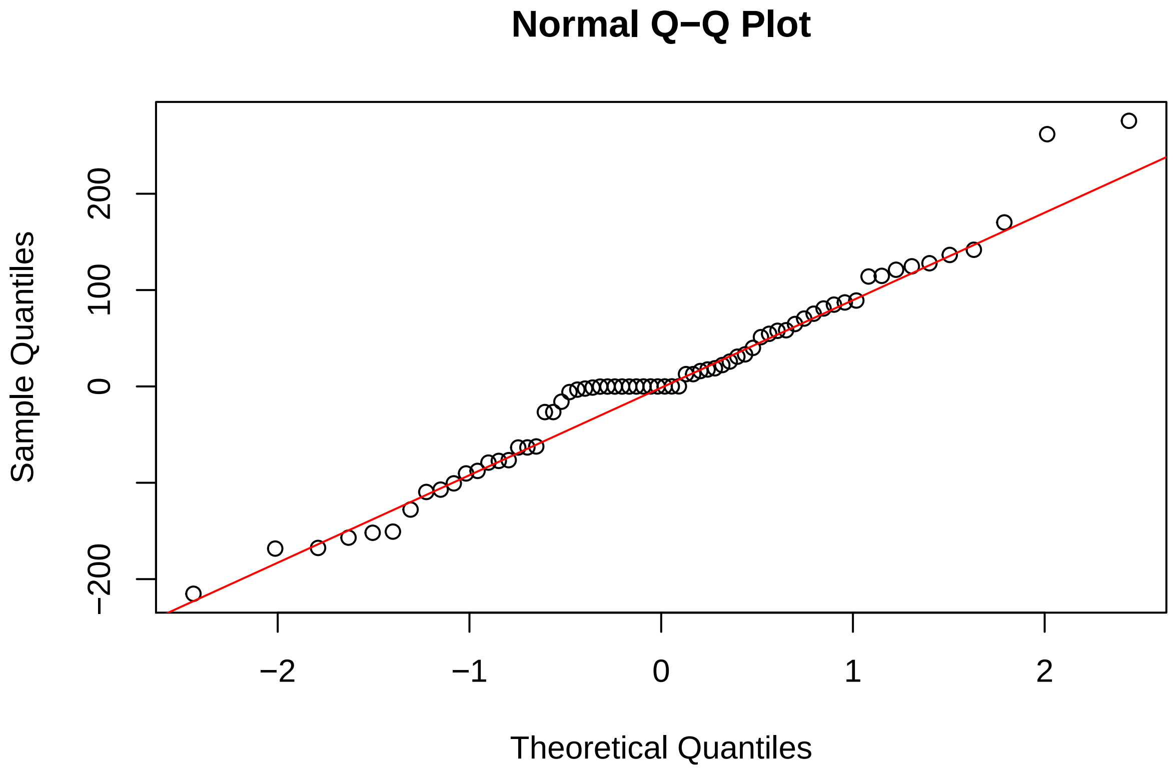
```
##
## Call:
## arima(x = series.diff12, order = c(0, 0, 1), seasonal = list(order = c(1, 1,
##     0), period = 12, method = "ML"))
##
## Coefficients:
##          ma1     sar1
##       0.4384  -0.7052
## s.e.  0.1183   0.0878
##
## sigma^2 estimated as 11039:  log likelihood = -344.35,  aic = 694.7
```

Model B Diagnostics

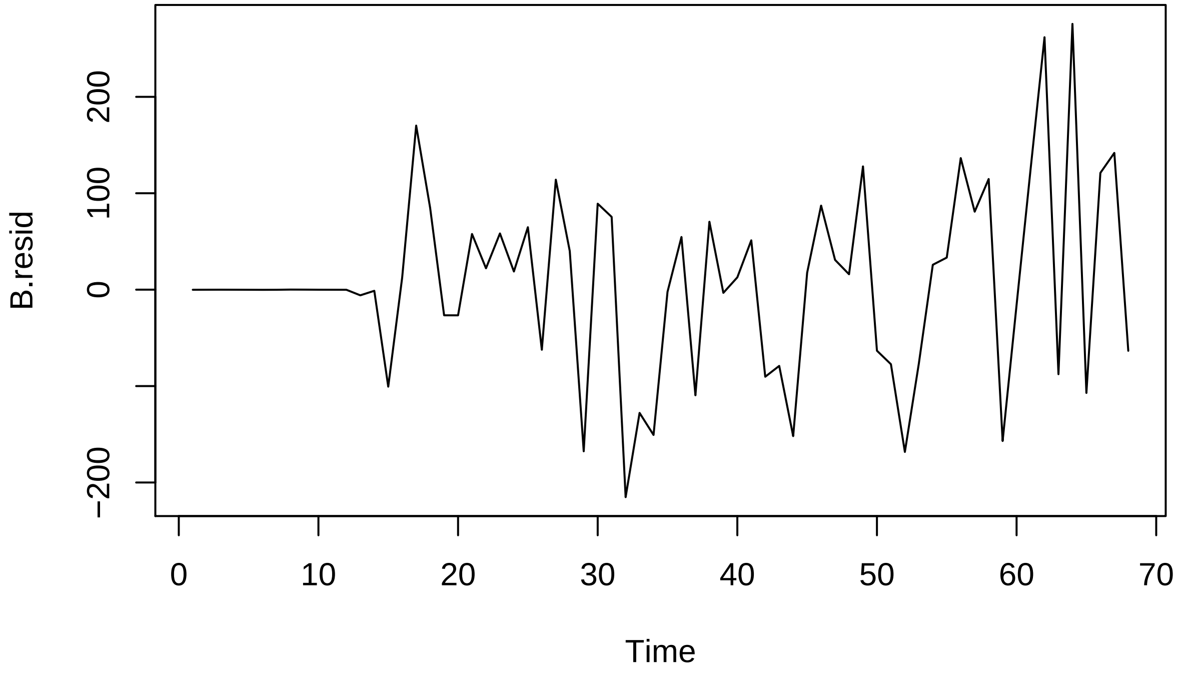
```
B.resid <-residuals(B)
hist(B.resid)
```



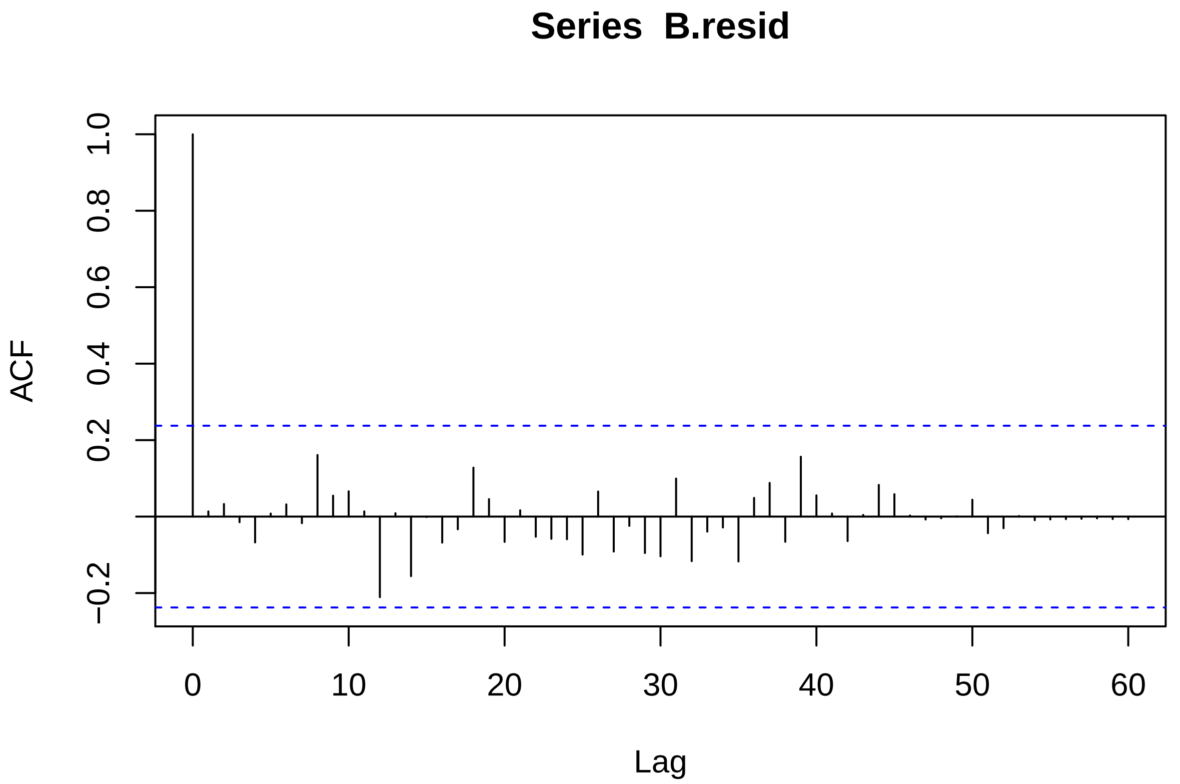
```
qqnorm(B.resid)
qqline(B.resid,col = "red")
```



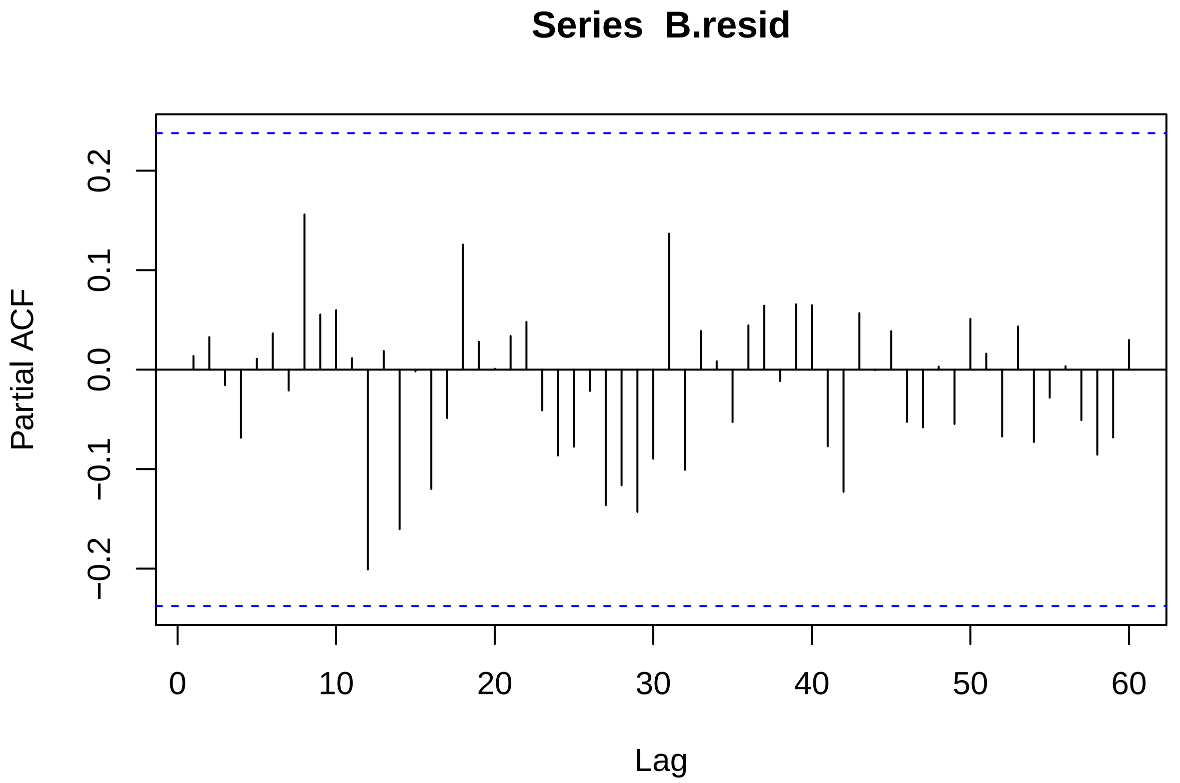
```
plot.ts(B.resid)
```



```
acf(B.resid,lag.max = 60)
```



```r
pacf(B.resid,lag.max = 60)
```



Model B Portmanteau Tests

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

```
##
##  Box-Pierce test
##
## data:  B.resid
## X-squared = 5.8115, df = 11, p-value = 0.8856
```

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

```
##
##  Box-Ljung test
##
## data:  B.resid
## X-squared = 7.0122, df = 11, p-value = 0.7981
```

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

```
##
##  Box-Ljung test
##
## data:  (B.resid)^2
## X-squared = 21.235, df = 12, p-value = 0.04704
```

```
shapiro.test(B.resid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  B.resid
## W = 0.97238, p-value = 0.1354
```

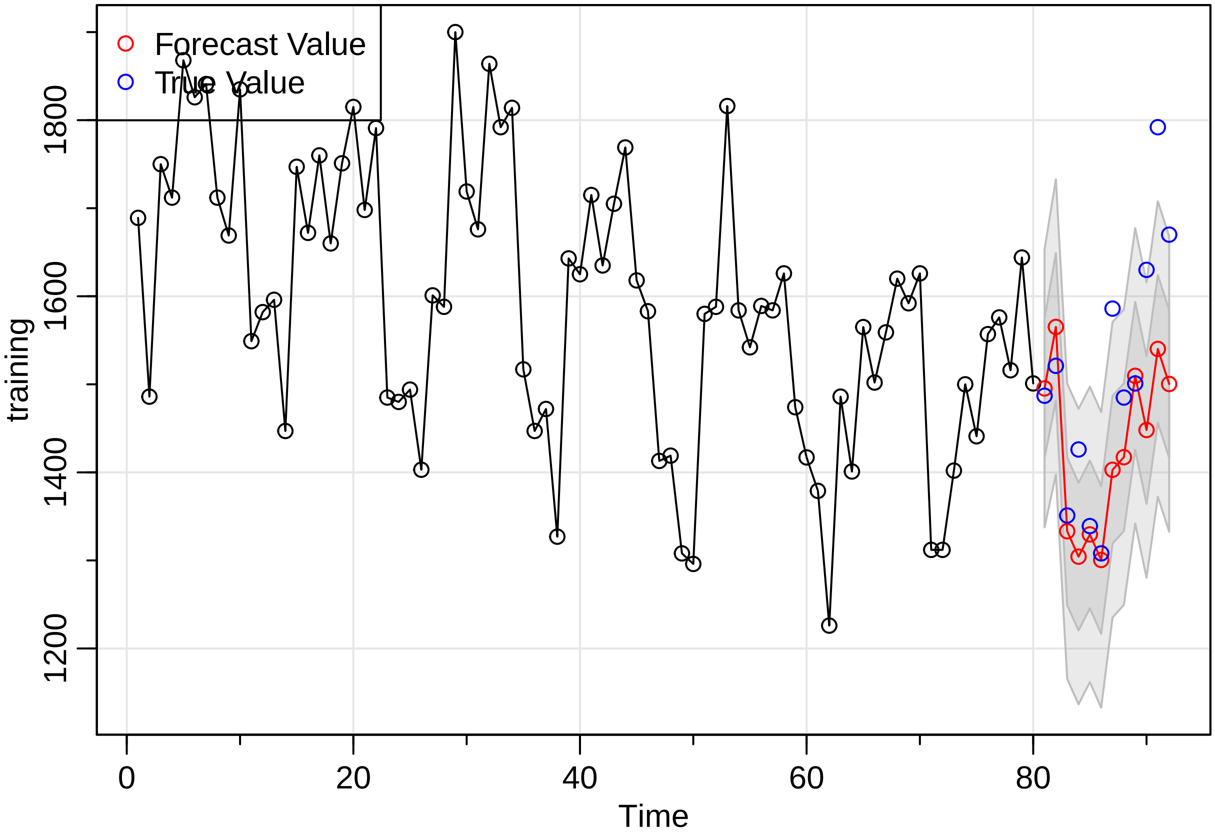
Based on the respective diagnostics performed on the Model A and Model B, which were the two models with the lowest AICc scores of the bunch created using auto.arima,we performed the diagnostic tests above. Upon performing the portmanteau tests, we see that the residuals of model B don't pass the Mcloed Li test for the squared residuals, and so the model to proceed with for forecasting is going to be model A. Model is a SARIMA (0,0,1) x (1,1,0)[12],and we will forecast with this model.

```
fit.A <-arima(training,order = c(0,0,1),seasonal = list(order = c(1,1,0),period = 12),method = "ML")
forecast::forecast(fit.A)
```

```
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 81       1525.932 1412.021 1639.842 1351.721 1700.143
## 82       1626.000 1500.674 1751.326 1434.331 1817.669
## 83       1368.669 1243.343 1493.994 1177.000 1560.338
## 84       1348.730 1223.404 1474.055 1157.061 1540.399
## 85       1393.954 1268.629 1519.280 1202.285 1585.624
## 86       1404.153 1278.827 1529.479 1212.484 1595.822
## 87       1456.741 1331.416 1582.067 1265.072 1648.410
## 88       1502.430 1377.105 1627.756 1310.761 1694.099
## 89       1572.152 1446.826 1697.478 1380.483 1763.821
## 90       1511.103 1385.777 1636.428 1319.434 1702.772
## 91       1614.266 1488.941 1739.592 1422.597 1805.936
## 92       1542.627 1417.301 1667.953 1350.958 1734.296
## 93       1549.043 1403.468 1694.617 1326.406 1771.680
## 94       1626.000 1476.513 1775.487 1397.379 1854.621
## 95       1348.846 1199.358 1498.333 1120.224 1577.467
##  96       1335.881 1186.394 1485.369 1107.260 1564.503
##  97       1396.769 1247.281 1546.256 1168.148 1625.390
##  98       1437.681 1288.194 1587.168 1209.060 1666.302
##  99       1451.235 1301.747 1600.722 1222.614 1679.856
## 100       1521.519 1372.032 1671.007 1292.898 1750.140
## 101       1573.498 1424.011 1722.986 1344.877 1802.119
## 102       1512.816 1363.328 1662.303 1284.195 1741.437
## 103       1624.667 1475.180 1774.155 1396.046 1853.289
## 104       1528.066 1378.578 1677.553 1299.444 1756.687
```

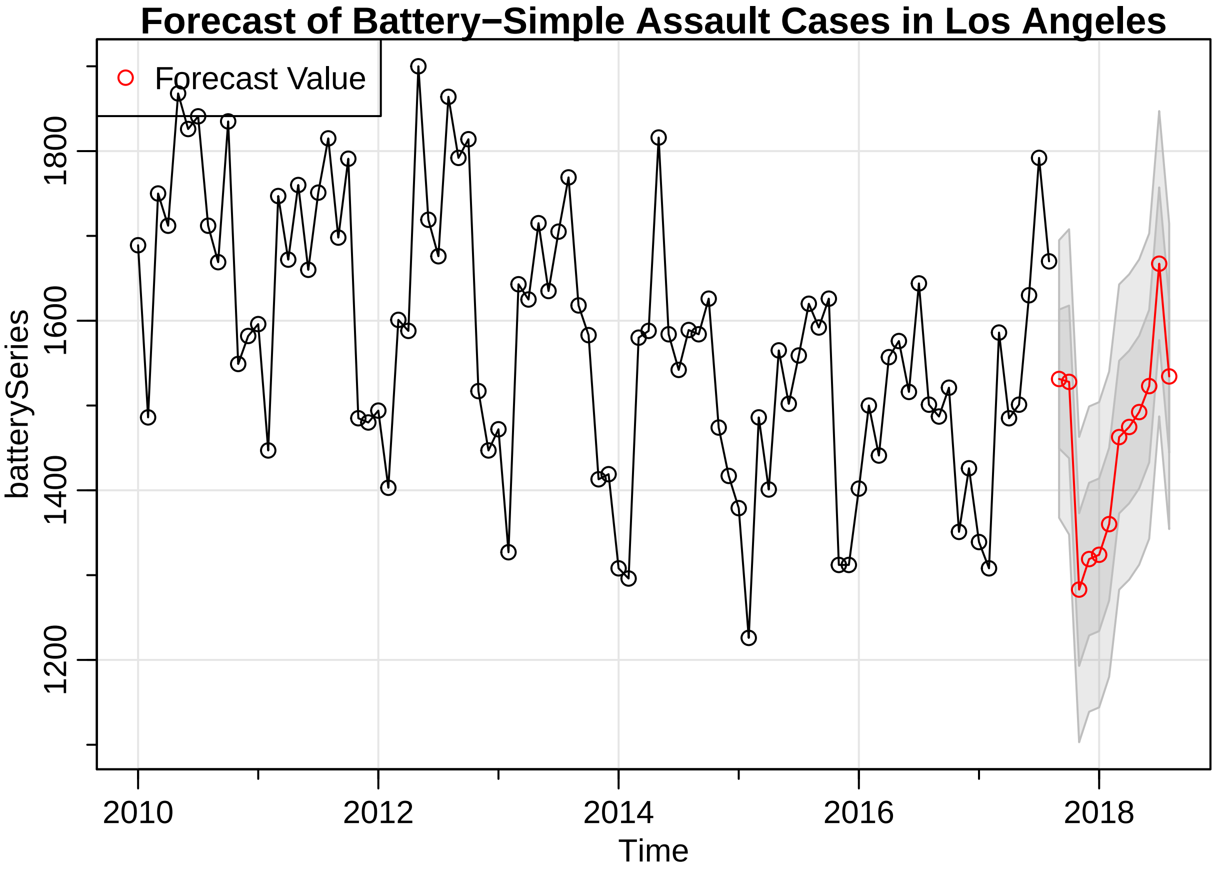
Forecast on Training Set

```
pred = sarima.for(training,12,0,0,1,1,1,0,S = 12,no.constant = FALSE)
points(81:92,testing,col = "blue")
legend("topleft",pch = 1, col = c("red","blue"),legend = c("Forecast Value","True Value"))
```



Forecast 12 Years Beyond End of Data

```
fc.Battery <- sarima.for(batterySeries, 12, 0,0,1,1,1,0, S = 12, no.constant = FALSE,plot.all = F)
legend("topleft",pch = 1, col = c("red"),legend = c("Forecast Value"))
title("Forecast of Battery-Simple Assault Cases in Los Angeles")
```


Forecast of Battery−Simple Assault Cases in Los Angeles
○ Forecast Value
batterySeries
1200
1400
1600
1800
2010
2012
2014
2016
2018
Time