Forecasting Crime Incidents in Boston using Time Series Modeling

STAT 5053

PARAG SASTURKAR
A20101974
parag.sasturkar@okstate.edu

Abstract

Crime is an unfortunate issue that societies need to deal with. Therefore, I believe that understanding it, and predicting the future incidents based on past, and then preparing for it is of utmost importance for societies to self-govern. "Prevention is always better than cure," says Romanian doctors and this goes hand in hand with my objective of forecasting number of crime incidents in Boston. This paper looks to build a time series model for analyzing crime data in Boston from 2015 - 2018. After model building, validation, and testing in practice, I chose the ARIMA(0,1,0)(0,1,0)[12] model as my champion model for forecasting purpose. Having said that some of the limitations of this paper and also recommendations for future analysis are also included.

Introduction

With democracy comes freedom and with freedom comes autonomy. This does not mean that you get a chance to disrupt others privacy or happy moments. Well, here I am talking about crime. In my opinion, with any type of crime comes to a lack of security and fear. It means if there is a crime in our societies, then people might lose their freedom to act, because they might always be in danger. In terms of application, accurate forecasting is beneficial for society because law enforcement can use data prediction to prepare for forecasted crime.

For the sake of my analysis, I collected the data from Kaggle website. You can refer to this data at the mentioned url - https://www.kaggle.com/ankkur13/boston-crime-data The dataset basically contains information about all the crimes happened in the Boston region from July 2015 to September 2018. This also contains information about what type of incidents happened, their specific location and so on. In order to use this data for time series modeling, I had to aggregate the number of incidents month-wise and save it in a flat file. The code for same is mentioned in Appendix B.

The final dataset that was prepared contains a total of 39 data points, one for each month from July 2015 to September 2018. Figure A mentioned below shows the original plot of the data over time.

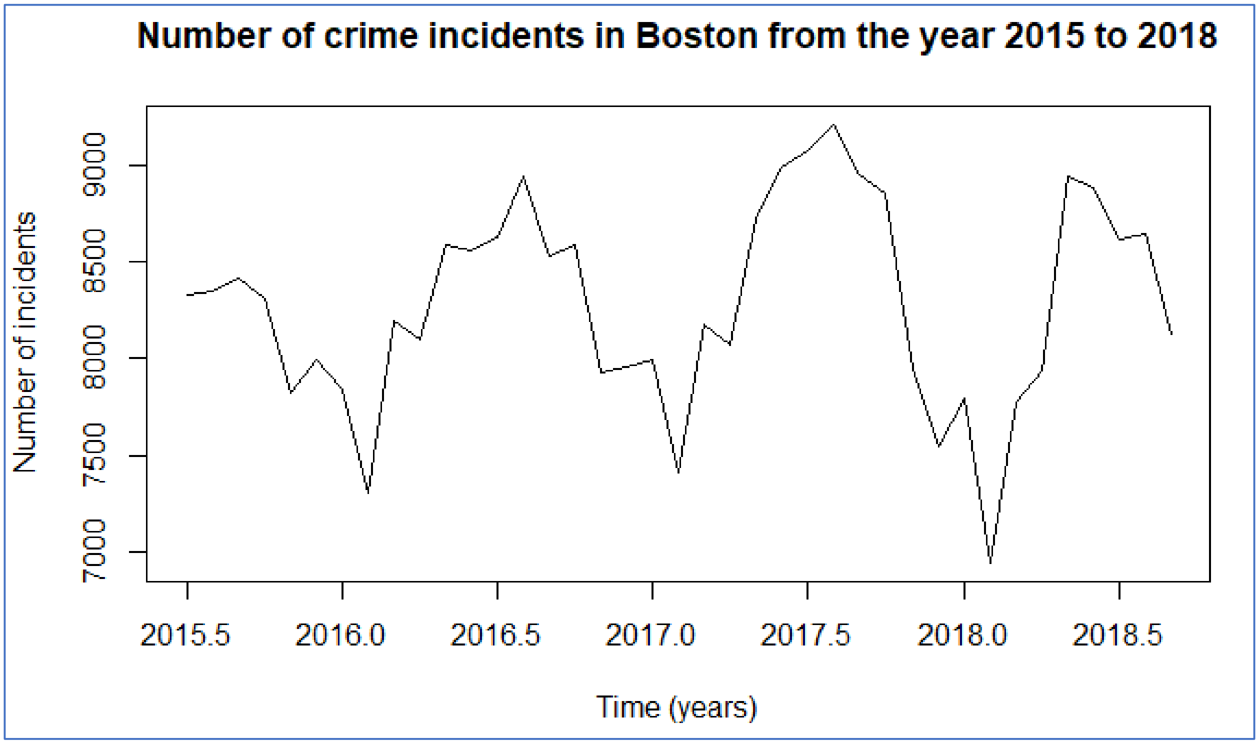

Figure A: Time Series Plot of the Original Data

Based on the nature of the data, multiple time series models like an ARIMA(0,1,0)(0,1,0)[12], ARIMA(1,1,0)(0,1,0)[12], ARIMA(0,1,1)(0,1,0)[12], and ARIMA(1,1,1)(0,1,0)[12] were built and then compared, tested, and validated against each other for the application of forecasting. The model ARIMA(0,1,0)(0,1,0)[12] was ultimately selected because of factors such as the analysis of over-parameterized models and the model's forecasting accuracy.

The methods section of this paper will argue the justification for these models. The models will be selected from plots and from the analysis of parameterized models. The models will then be validated and tested (in terms of forecasting). The results section will outline the differences between these models and how they forecast differently. The conclusion section of the paper will make a final selection of selecting a single model, offering justifications for the conclusion reached. Besides, the end of the paper will discuss future work and limitations of this paper. All plots and R output referenced for analysis can be located in Appendix A.

Methods

After generating an array of different models and testing them, an ARIMA(0,1,0)(0,1,0)[12] model was selected to model the crime incidents. The model is basically written as $\nabla\nabla_{12}(Y_t) = e_t$, where e_t is a white noise zero mean process with constant variance as 58745. The model has the following assumptions:

1. $E[e_t] = 0$
2. $Var[e_t] = \sigma^2$ (constant)
3. $Cov(e_t, e_{t-k}) = 0$ (White Noise)
4. $e_{t,s}$ are normally distributed

Based on the original plot of the data (Figure A), it seems there is seasonality in the data and hence, I had to account it for while building the models. As described in the model notation, I have first adjusted the data for seasonality and then worked on modeling the trend. The detailed procedure is explained in the further part of the report.

The process of arriving at the models involved looking at the original data and then making transformations. According to Figure 1 in Appendix A, the Box-Cox transformation plot suggests using the original data for the model building process. ACF plot of the original data (Figure 2 in Appendix A) suggests that the lags are oscillating and decaying over time so there might be seasonality in the data or the data has complex roots with AR(2) parameter in it. Having said that, it was clear that data were not stationary. Even the result from the Augmented Dickey Fuller test (Figure 3 in Appendix A) yielded a p-value greater than 0.05 (0.8822) at the higher lag order of 7 which suggested that the data were not stationary.

Further after looking at the ACF plot of the differenced original data (Figure 4 in Appendix A), I could easily notice significant lags at 6^{th} and 12^{th} points. It was clear that there is seasonality in the data. Hence, I went back and took the seasonal difference in the original data. The plot of the seasonally adjusted data is displayed in Appendix A (refer to Figure 5). It seemed like there was still a downward trend present in the data after adjusting it for seasonality. After performing the ADF test (Figure 6 in Appendix A) on this data, it yielded a p-value greater than 0.05 (0.4886) at the default order of 2 which suggested that the seasonally adjusted data were not stationary. Going ahead, I had to make the data stationary in order to obtain the model parameters correctly, hence I took a single difference on the seasonally adjusted data. The plot of the data is displayed in Appendix A (refer to Figure 7). Here, I noticed that there were not any trends in the data and hence it looked stationary. To be sure, I again performed the ADF test on it (Figure 8) that yielded a p-value of less than 0.05 (0.01). It supported the alternative hypothesis that the data were stationary. Now, I could use this data for further analysis.

According to Figure 9 in Appendix A, the ACF plot of the seasonally adjusted differenced data suggests that it is a white noise model. According to Figure 10 in Appendix A, the PACF plot of the seasonally adjusted differenced data also suggests that it is a white noise model. Besides, the EACF plot (Figure 11 in Appendix A) of the seasonally adjusted differenced data appears to support the ARMA(0,0) model. Further, the BIC plot (Figure 12) gave different results altogether. Based on the majority of the plots, I decided to go ahead and used ARMA(0,0) model for the seasonally adjusted differenced data forecasting. Therefore, the final selected model looked like ARIMA(0,1,0)(0,1,0)[12].

Next step in the modeling process was to compare this model results with other model results. So, I built other models like ARIMA(1,1,0)(0,1,0)[12], ARIMA(0,1,1)(0,1,0)[12], and ARIMA(1,1,1)(0,1,0)[12]. Basically, in these models, I had just added one extra parameter to compare the results using the overparameterized models technique. Over-parameterized models had total four assumptions as –

1. Models should have significant parameters means their confidence intervals should not contain zero in them

2. Models parameters should remain unchanged or similar

3. Lower AIC models are better

4. Parsimony supports lesser parameters models

Refer the appendix A for the actual results. Summary of the results is displayed in the following table –

Model	Parameters Significance	Change in Estimates	AIC	Parsimony	Result
ARIMA(0,1,0)(0,1,0)[12] (Figure 13 in Appendix A)	Zero parameters	No change	361.29	Supports this model	As all the overparameterized model assumptions support this model, I have chosen it as my champion model
ARIMA(1,1,0)(0,1,0)[12]	AR parameter was not	Very Similar	363.14	Does not Support	Overparameterized model

(Figure 14 in Appendix A)	significant as its confidence interval (-0.474583, 0.3174316) contained zero in it			this model over the first model	
ARIMA(0,1,1)(0,1,0)[12] (Figure 15 in Appendix A)	MA parameter was not significant as its confidence interval (-0.480884, 0.3204021) contained zero in it	Very Similar	363.14	Does not Support this model over the first model	Overparameterized model
ARIMA(1,1,1)(0,1,0)[12] (Figure 16 in Appendix A)	Both the parameters, MA and AR were not as significant as their confidence intervals (-1.665385, 2.841549) and (-2.787434, 1.474319) respectively contained zero in them	Parameter coefficients changed a lot as compared to last two models	365.13	Does not Support this model over the first model	Overparameterized model

After deciding the final model, it was the time to check if the model satisfies all the model assumptions or not. Basically, I performed the residual analysis. To check the constant mean and variance, I plotted the residuals over time (Figure 17 in Appendix A) and observed that the residuals vs. time plot show that variance is constant throughout the entire model and no increasing or decreasing trends are present in the residuals. This information suggests the assumptions of E[et] = 0 and the constant variance are valid. Then for the normality assumption, I performed the Shapiro-Wilk test (Figure 18 in Appendix A) and based on the result I can say that the normality assumption is fulfilled as the p-value turned out to be higher than 0.05 (0.2) and hence I could not reject the null hypothesis of normality. In contrast, the QQ plot (Figure 19 in Appendix A) showed some deviation from the normality as many data points did not fall on the 45-degree line. Having said it, in my case normality does not really affect the model results as I have enough data points and hence central limit theorem holds true. Lastly, the assumption for white noise [cov(e_t, e_{t-k}) = 0] appears to be fulfilled as most of the lags are within the tolerance interval as displayed in the ACF plot of the residuals (Figure 20 in Appendix A). Besides, the p-values in the Box-Ljung test are all more than the Bonferroni adjusted significance level (Figure 21 in Appendix A).

From the above analysis, I can state that the ARIMA(0,1,0)(0,1,0)[12] was indeed the better model for my data. It satisfied all the criteria of validation assessments. Just as a further check on the model, I decided to check if there are any outliers in the data which I can basically model, but based on the output in Appendix A (Figure 22), I can say that there were not any additive or innovative outliers left to model. In this way, I believe that my ultimate model results are more accurate than any other models and hence can be used for the crime incident forecasting purpose.

As mentioned earlier, the primary application of my analysis was to forecast future crime incidents in the Boston region. Forecasting future crime incidents could help the Boston Police Department and/or Law Makers to develop an expectation of how to prepare for future crimes in the region, including allocating resources to increase stoppage and prevention of any type of the crime incidents.

Results

To summarize my analysis, the ACF, PACF, and EACF plots of the seasonally adjusted differenced data suggested that the data can be best modeled as a white noise process and hence, I decided to use the ARIMA(0,1,0)(0,1,0)[12] model as my champion model to forecast the future crimes in the Boston.

The champion model was selected after comparing it with other 3 models based on overparameterized models technique. Further, the champion model also satisfied all the model assumptions for residual analysis. Finally, the model was used to forecast the crime incidents in the Boston region for the next two years (until September 2020). Please refer to Figure 23 in Appendix A. Here, the forecasting graph showed that the overall number of crime incidents are going down with strong seasonality present in it. Basically, the forecast followed the same pattern as before and it seems that the crime rate is increasing in the middle of the year and then again decreasing at the year-end. Besides, the actual point estimates and the 80% and 95% confidence interval values also been generated and can be looked at Figure 24 in Appendix A.

Conclusion

Police departments across the world have to deal with the reality that crime is prevalent. The data was collected in order to predict the future crime incidents in Boston that were to occur over the next two years. The ARIMA(0,1,0)(0,1,0)[12] model was ultimately selected as the model to forecast with, and the model was proven to forecast with accuracy and reliability.

As you know there might be n number of factors that are basically responsible for the increase or decrease in the crime incidents. In my opinion, the limitations of this model are that it does not consider any external factors such as for example political impact or say poverty rate or any such factors into consideration while forecasting the crime incidents.

So, the future work may include determining if incorporating external factors increases or decreases the forecasting abilities of this particular data set. Basically, the current forecasted numbers looked convincing based on the past and hence I could say that my model was working properly. But here a point should be noted that this forecast might change if you include external factors in the analysis. This would be a very good future scope to carry forward with this idea of forecasting crime incidents in the Boston region.

APPENDIX A

Box Cox Transformation

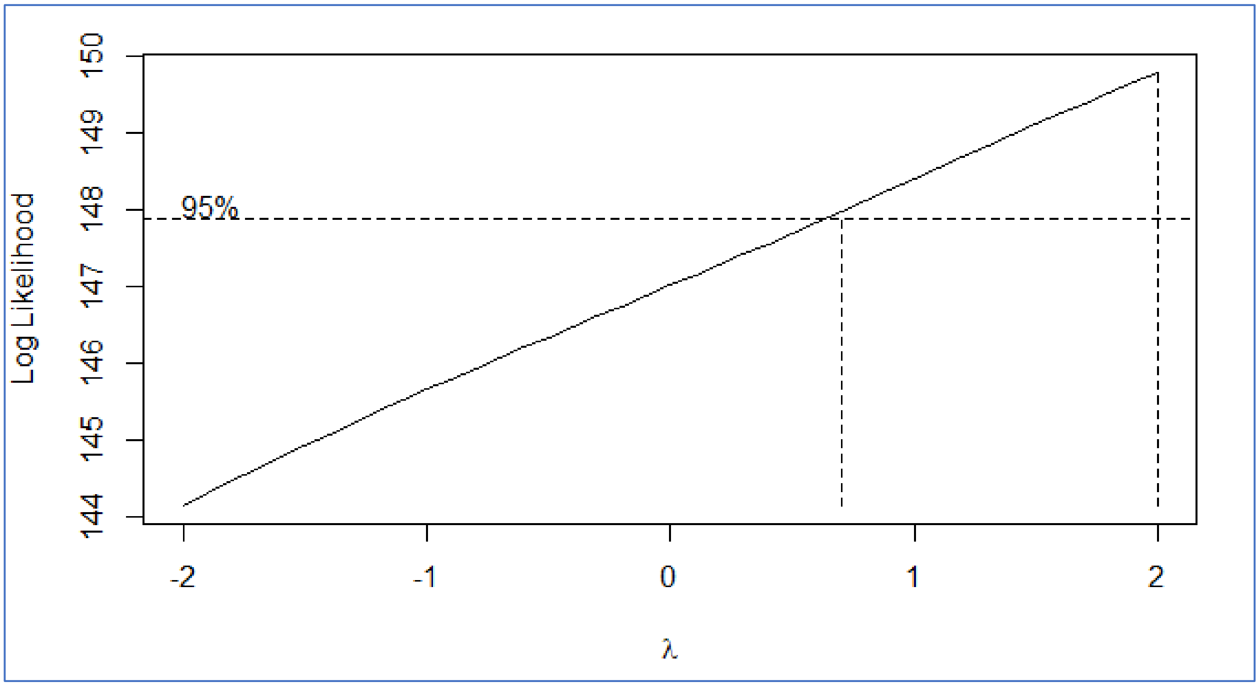

Figure 1 – Box-Cox transformation of the original data

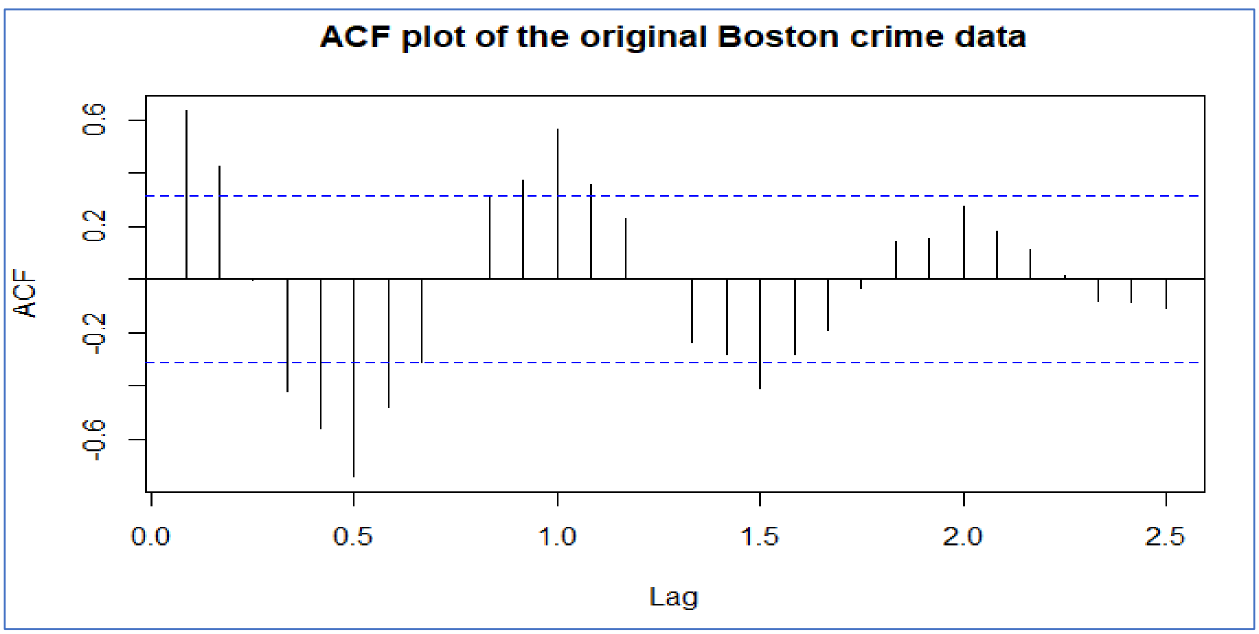

Figure 2 – ACF plot of the original data

```
        Augmented Dickey-Fuller Test

data:  bos.ts
Dickey-Fuller = -1.2115, Lag order = 7, p-value = 0.8822
alternative hypothesis: stationary
```

Figure 3 – ADF test on original data

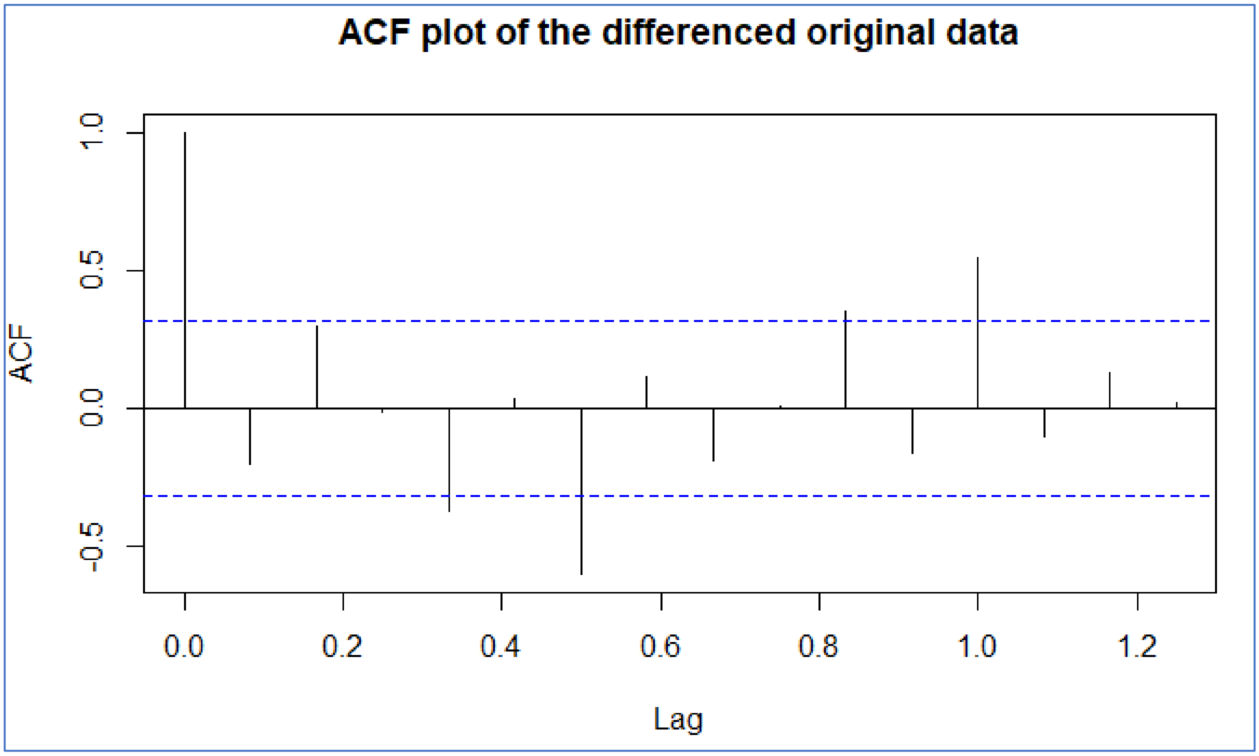

Figure 4 – ACF plot of the differenced original data

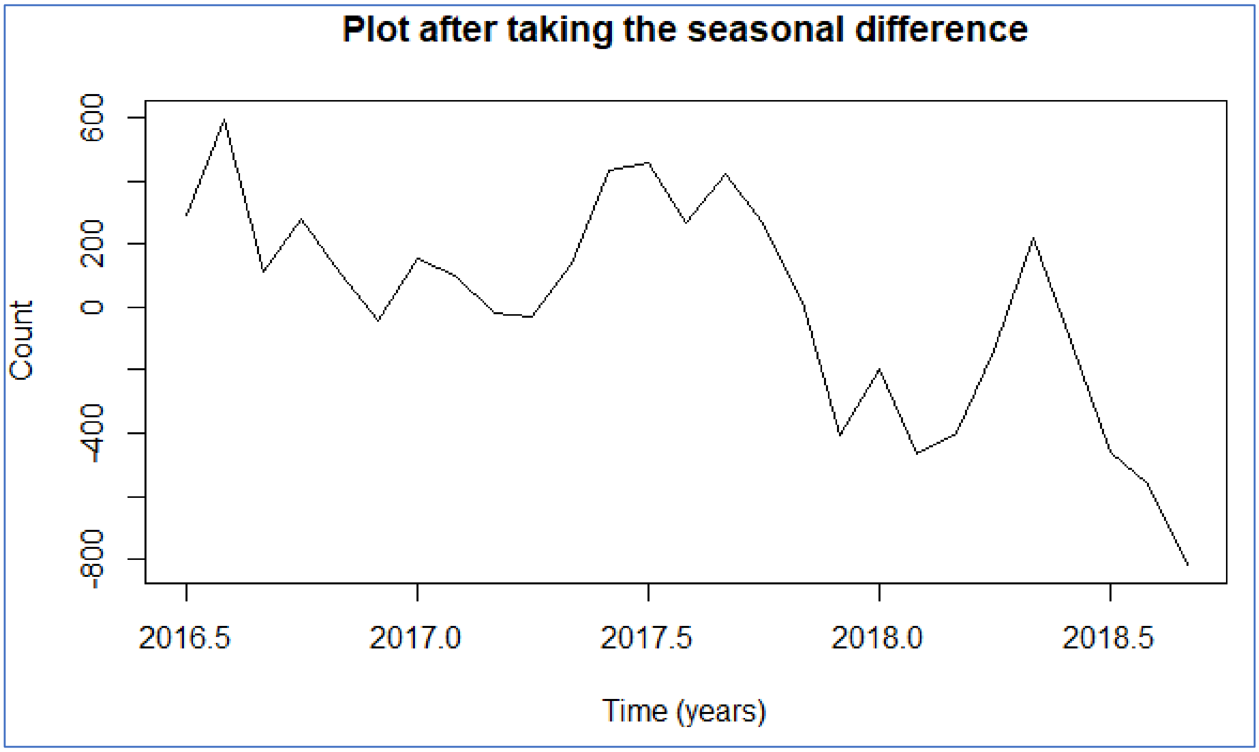

Figure 5 – Plot of the seasonally adjusted data

```
        Augmented Dickey-Fuller Test

data:  bos.sa.diff
Dickey-Fuller = -2.2196, Lag order = 2, p-value = 0.4886
alternative hypothesis: stationary
```

Figure 6 – ADF test on seasonally adjusted data

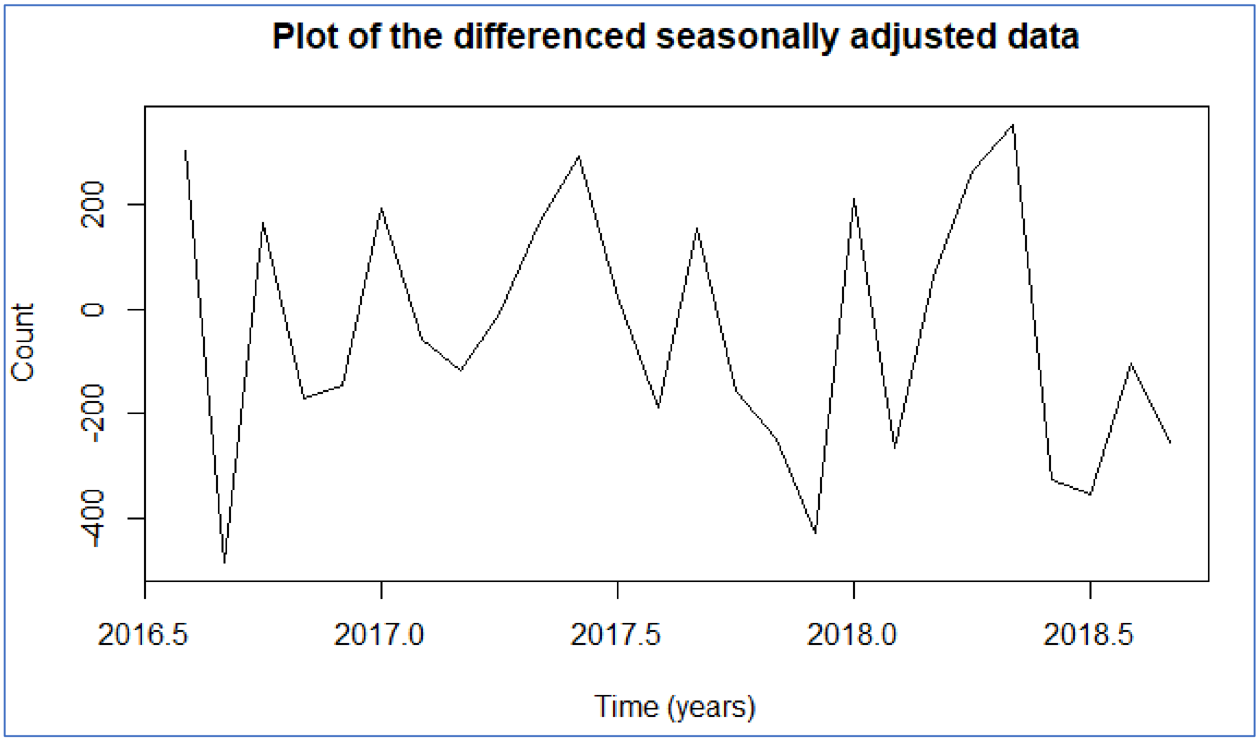

Figure 7 – Plot of the differenced seasonally adjusted data

```
        Augmented Dickey-Fuller Test

data:  bos.diff
Dickey-Fuller = -5.4275, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary
```

Figure 8 – ADF test on the seasonally adjusted differenced data

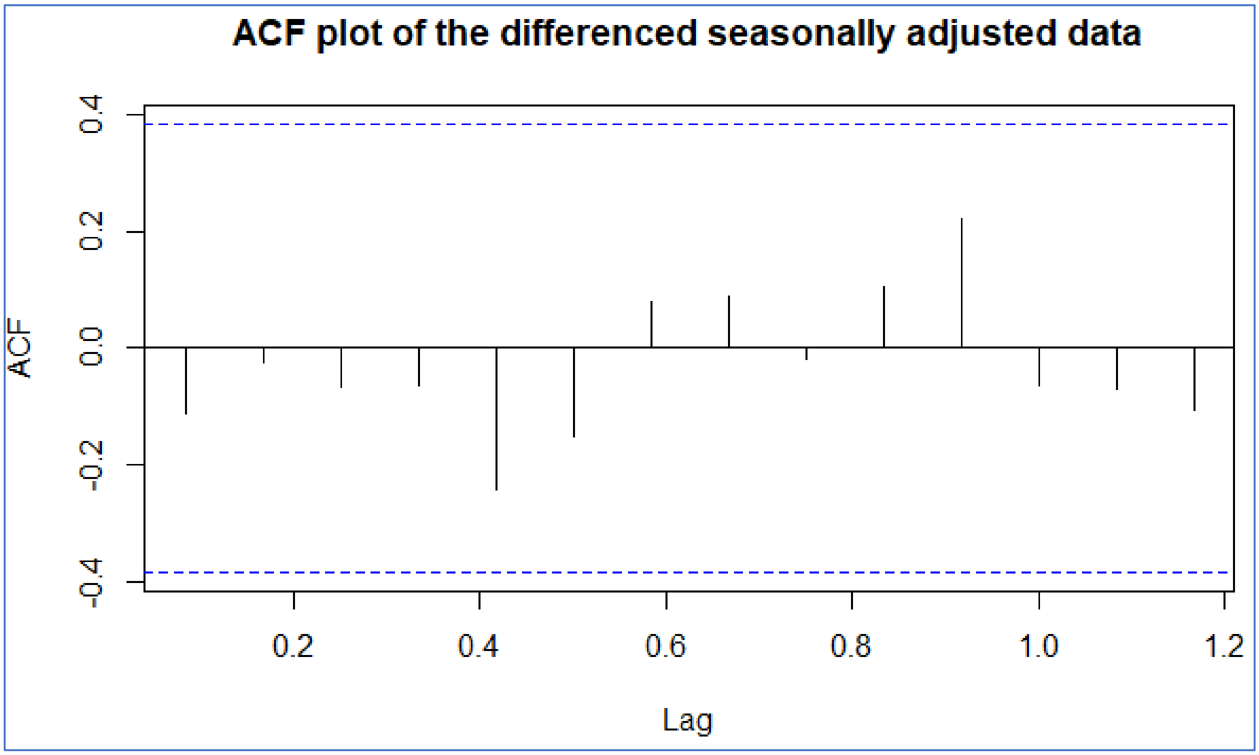

Figure 9 – ACF plot of the seasonally adjusted differenced data

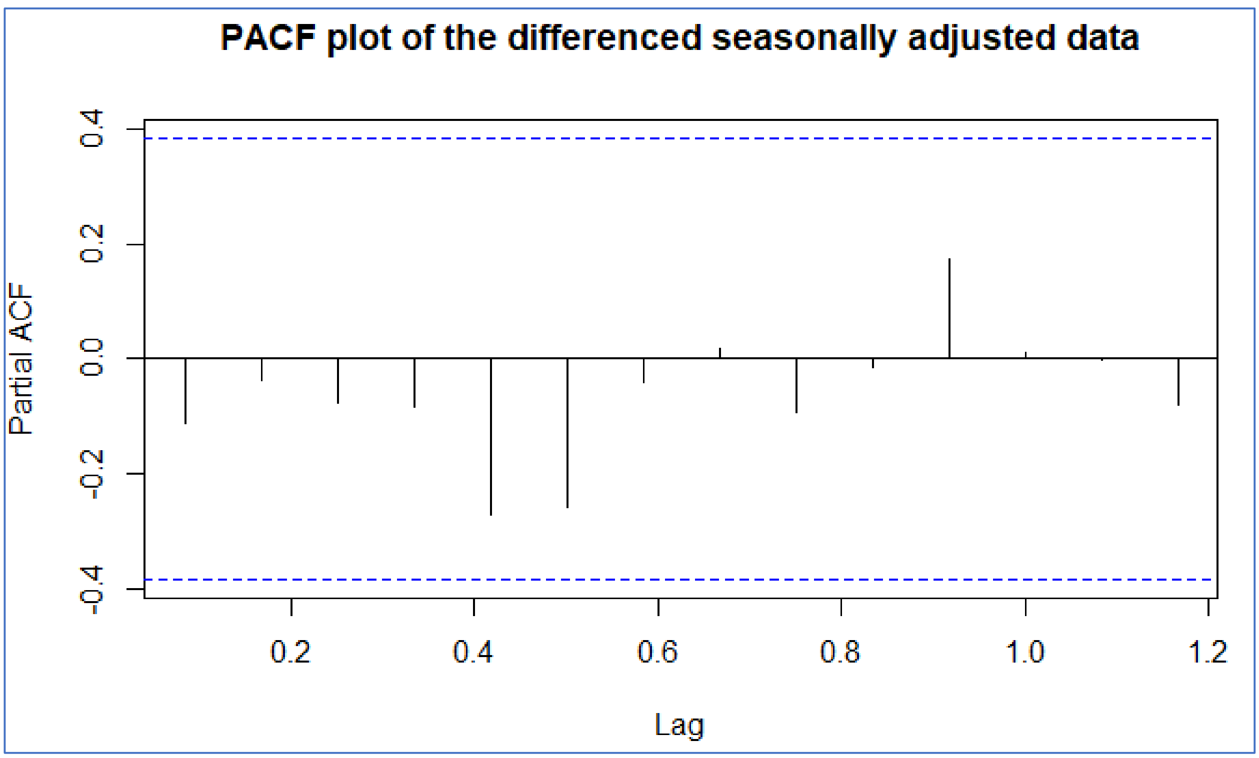

Figure 10 – ACF plot of the seasonally adjusted differenced data

EACF plot of the differenced seasonally adjusted data

AR/MA	0	1	2	3	4	5
0	o	o	o	o	o	o
1	o	o	o	o	o	o
2	o	o	o	o	o	o
3	x	o	o	o	o	o
4	o	o	o	o	o	o
5	x	o	o	o	o	o

Figure 11 – ACF plot of the seasonally adjusted differenced data

BIC plot of the differenced seasonally adjusted data

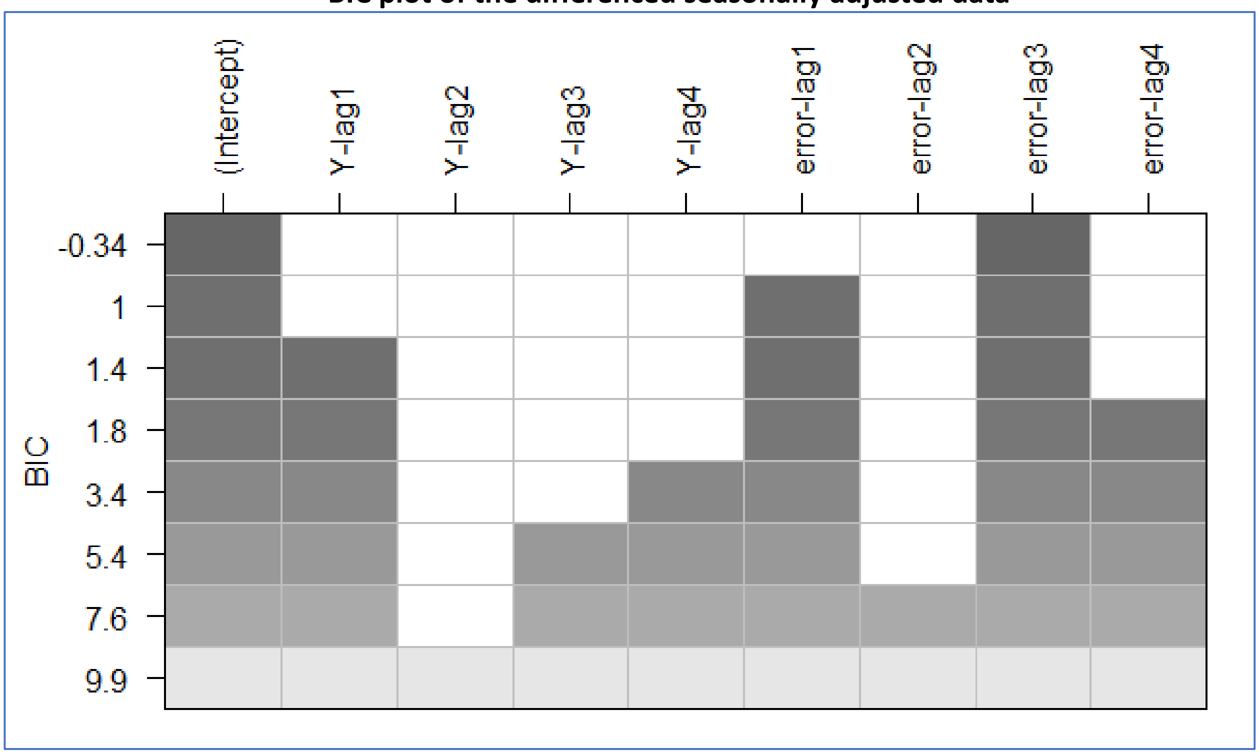

Figure 12 – BIC plot of the seasonally adjusted differenced data

```
> arima(bos.ts, order = c(0,1,0), seasonal = list(order = c(0,1,0), period = 12))

Call:
arima(x = bos.ts, order = c(0, 1, 0), seasonal = list(order = c(0, 1, 0), period = 12))


sigma^2 estimated as 58745:  log likelihood = -179.64,  aic = 361.29
```

Figure 13 – Summary result of the model ARIMA(0,1,0)(0,1,0)[12]

```
> arima(bos.ts, order = c(1,1,0), seasonal = list(order = c(0,1,0), period = 12))

Call:
arima(x = bos.ts, order = c(1, 1, 0), seasonal = list(order = c(0, 1, 0), period = 12))

Coefficients:
          ar1
      -0.0786
s.e.   0.2020

sigma^2 estimated as 58391:  log likelihood = -179.57,  aic = 363.14
> confint(arima(bos.ts, order = c(1,1,0), seasonal = list(order = c(0,1,0), period = 12)))
        2.5 %    97.5 %
ar1 -0.474583 0.3174316
```

Figure 14 – Summary result of the model ARIMA(1,1,0)(0,1,0)[12]

```
> arima(bos.ts, order = c(0,1,1), seasonal = list(order = c(0,1,0), period = 12))

Call:
arima(x = bos.ts, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 0), period = 12))

Coefficients:
          ma1
      -0.0802
s.e.   0.2044

sigma^2 estimated as 58384:  log likelihood = -179.57,  aic = 363.14
> confint(arima(bos.ts, order = c(0,1,1), seasonal = list(order = c(0,1,0), period = 12)))
         2.5 %    97.5 %
ma1 -0.4808841 0.3204021
```

Figure 15 – Summary result of the model ARIMA(0,1,1)(0,1,0)[12]

```
> arima(bos.ts, order = c(1,1,1), seasonal = list(order = c(0,1,0), period = 12))

Call:
arima(x = bos.ts, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 0), period = 12))

Coefficients:
          ar1     ma1
      -0.6566  0.5881
s.e.   1.0872  1.1497

sigma^2 estimated as 58348:  log likelihood = -179.56,  aic = 365.13
> confint(arima(bos.ts, order = c(1,1,1), seasonal = list(order = c(0,1,0), period = 12)))
        2.5 %   97.5 %
ar1 -2.787434 1.474319
ma1 -1.665385 2.841549
```

Figure 16 – Summary result of the model ARIMA(1,1,1)(0,1,0)[12]

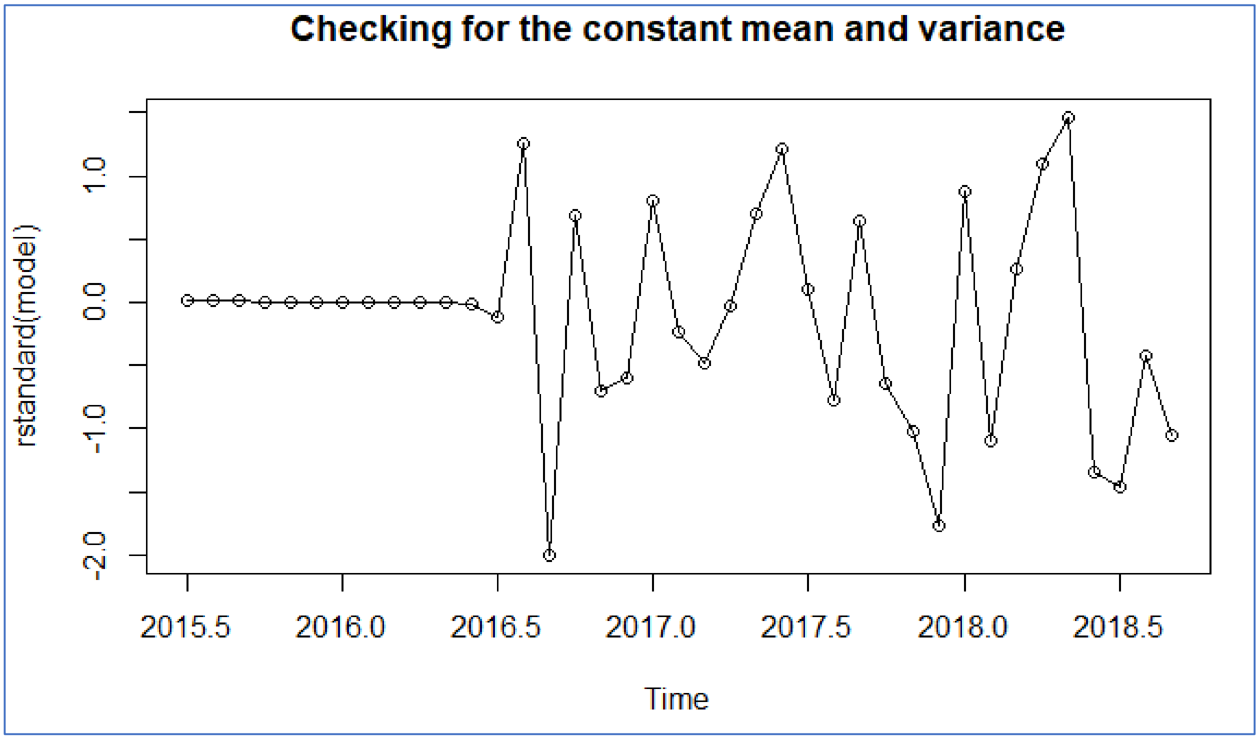

Figure 17 – Checking for the constant mean and variance

```
        Shapiro-Wilk normality test

data:  rstandard(model)
W = 0.96153, p-value = 0.2005
```

Figure 18 – Checking for the normality

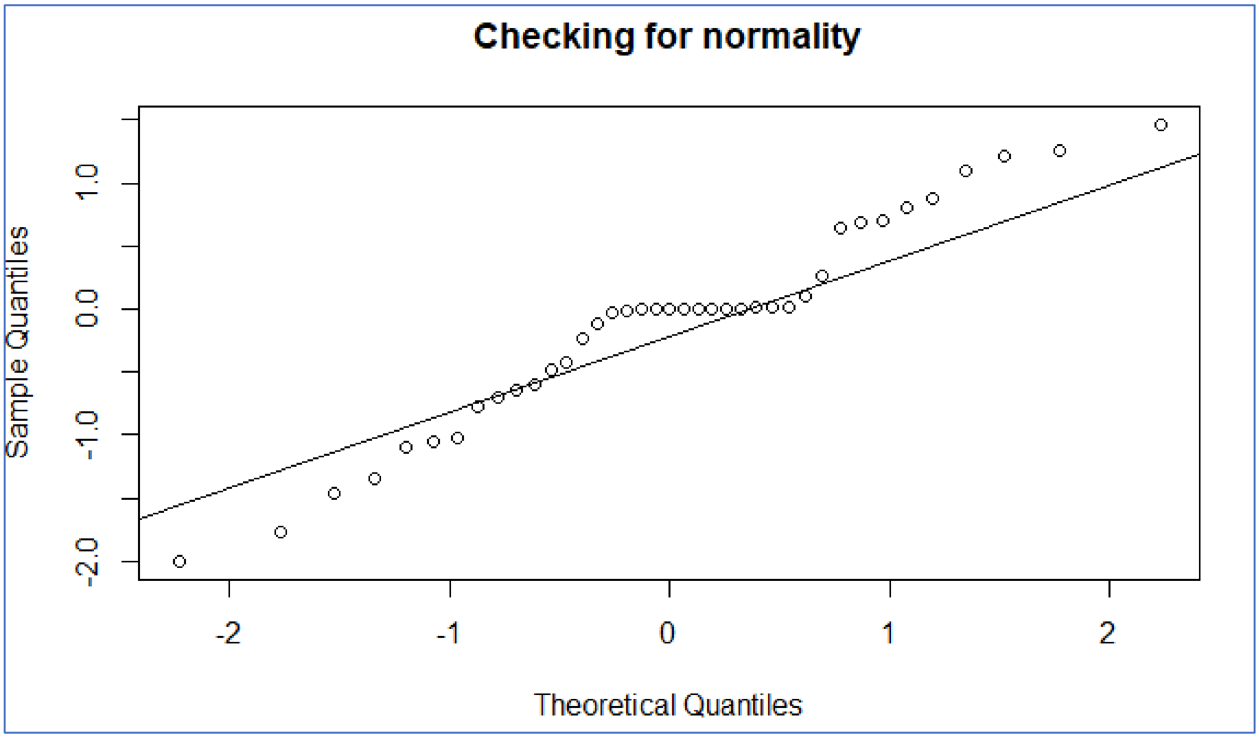

Figure 19 – Checking for the normality

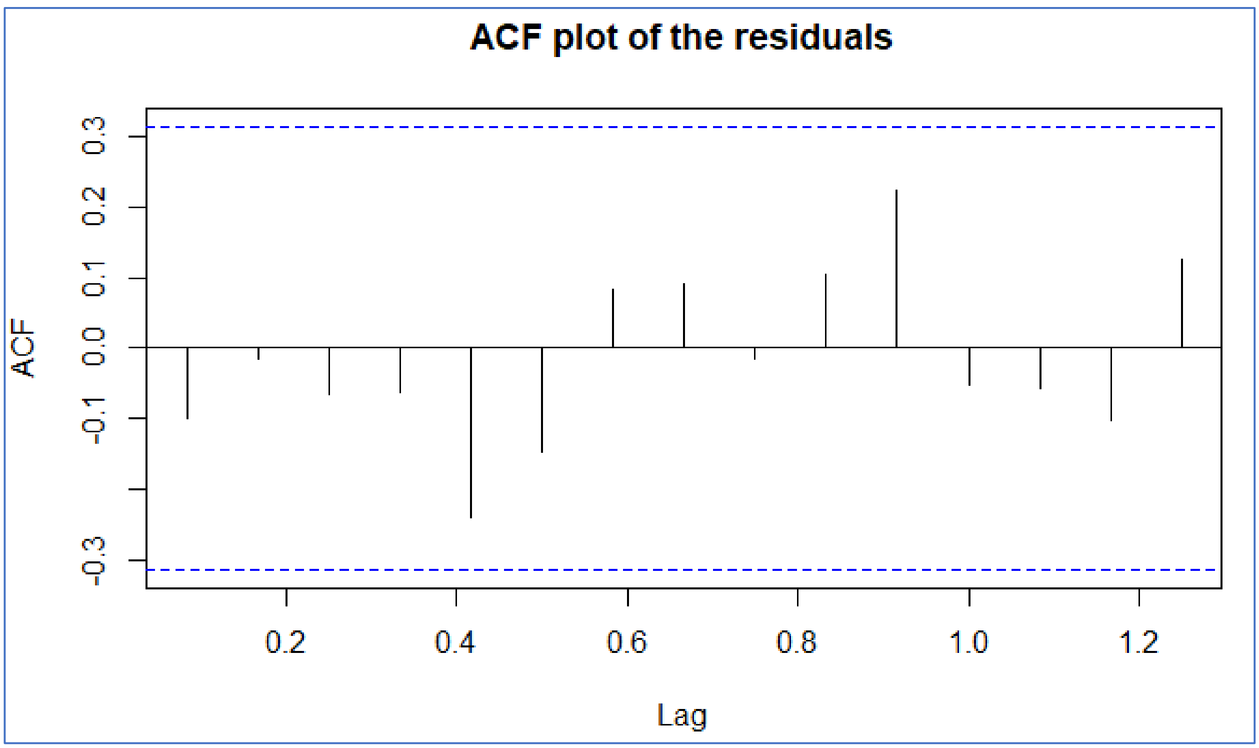

Figure 20 – Checking for independence of the residuals

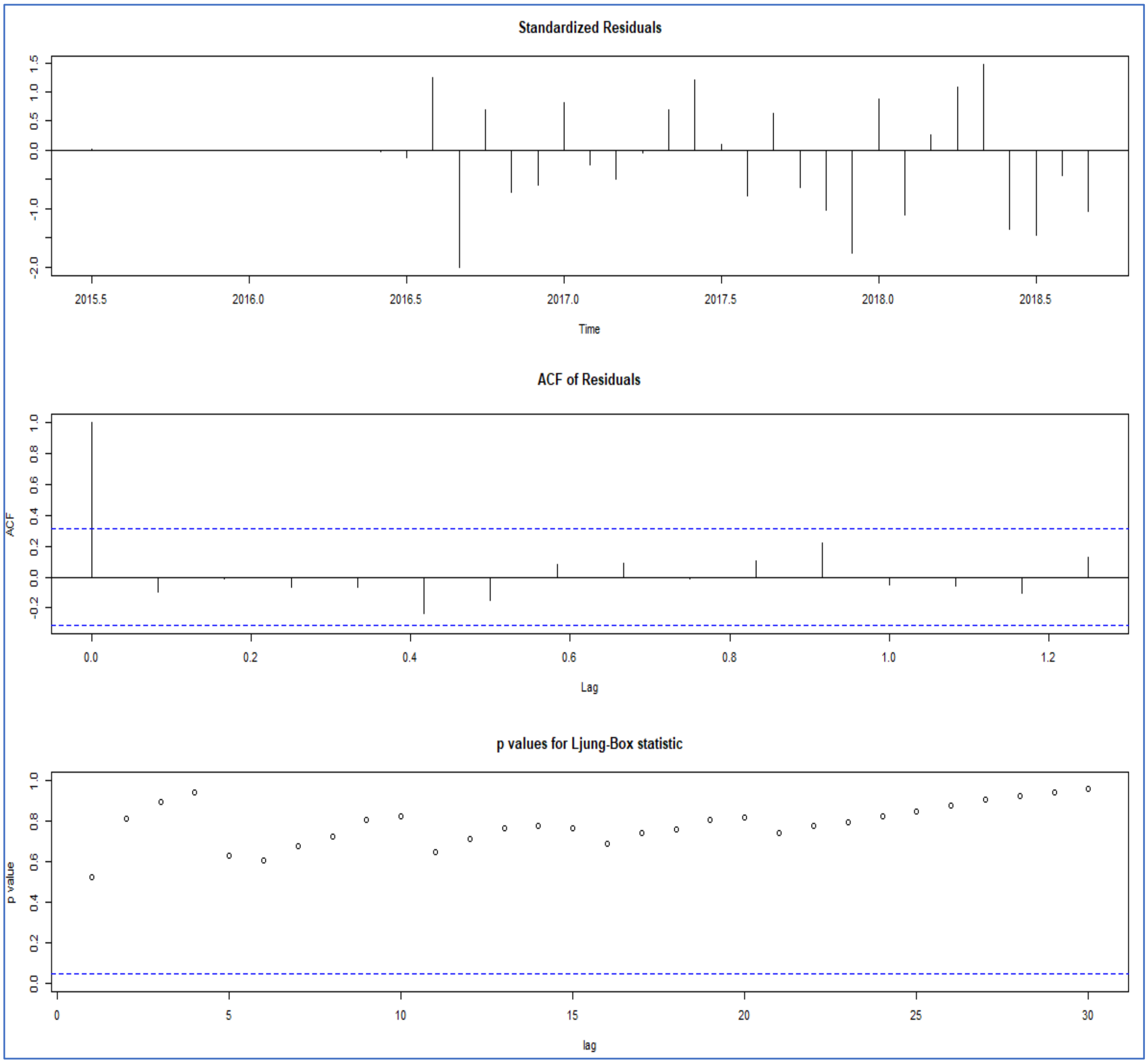

Figure 21 – Output for tsdiag – last plot represents the Box-Ljung test

```
> detectAO(model)
[1] "No AO detected"
> detectIO(model)
[1] "No IO detected"
```

Figure 22 – Checking for outliers

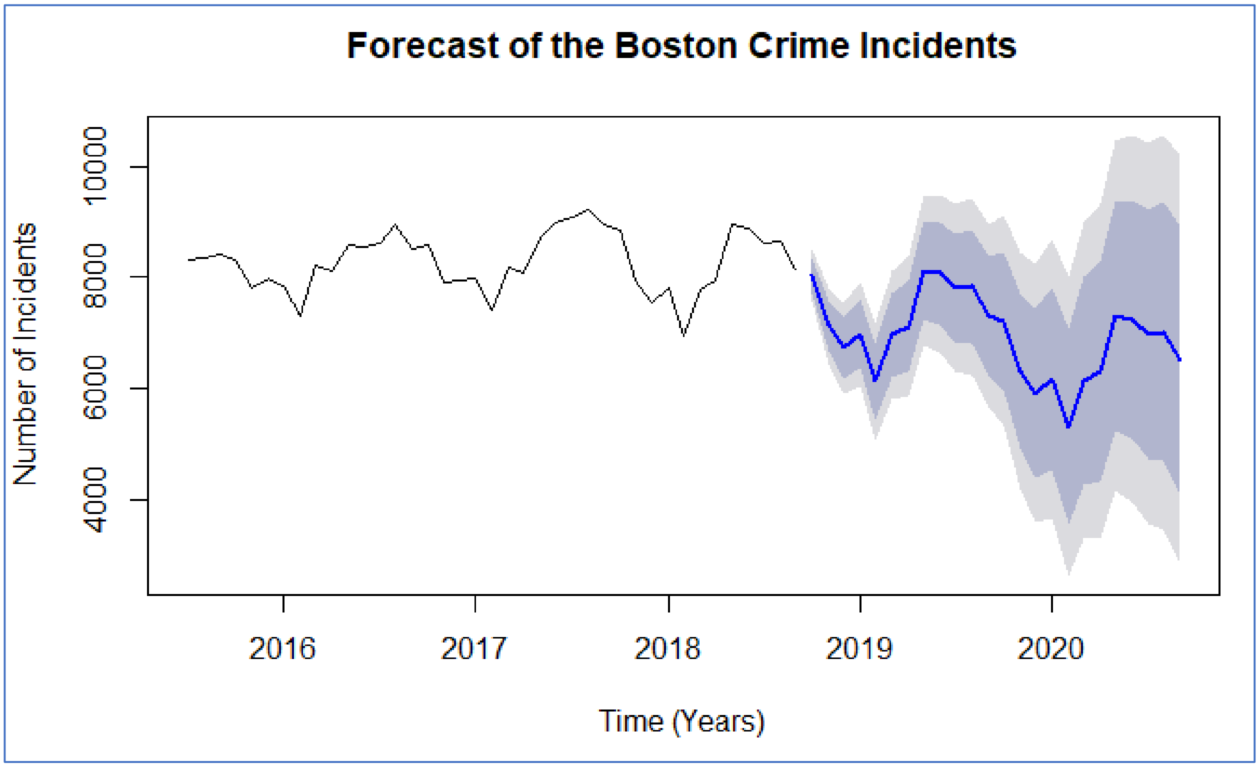

Figure 23 – Forecast of crime incidents in Boston

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Oct 2018	8036	7725.387	8346.613	7560.958	8511.042
Nov 2018	7125	6685.726	7564.274	6453.189	7796.811
Dec 2018	6724	6186.002	7261.998	5901.203	7546.797
Jan 2019	6977	6355.773	7598.227	6026.916	7927.084
Feb 2019	6125	5430.447	6819.553	5062.774	7187.226
Mar 2019	6960	6199.156	7720.844	5796.389	8123.611
Apr 2019	7117	6295.194	7938.806	5860.157	8373.843
May 2019	8121	7242.453	8999.547	6777.378	9464.622
Jun 2019	8065	7133.160	8996.840	6639.874	9490.126
Jul 2019	7799	6816.754	8781.246	6296.785	9301.215
Aug 2019	7827	6796.812	8857.188	6251.464	9402.536
Sep 2019	7314	6238.004	8389.996	5668.406	8959.594
Oct 2019	7218	5975.547	8460.453	5317.831	9118.169
Nov 2019	6307	4917.895	7696.105	4182.547	8431.453
Dec 2019	5906	4384.312	7427.688	3578.778	8233.222
Jan 2020	6159	4515.389	7802.611	3645.313	8672.687
Feb 2020	5307	3549.906	7064.094	2619.756	7994.244
Mar 2020	6142	4278.320	8005.680	3291.747	8992.253
Apr 2020	6299	4334.509	8263.491	3294.570	9303.430
May 2020	7303	5242.624	9363.376	4151.927	10454.073
Jun 2020	7247	5095.008	9398.992	3955.812	10538.188
Jul 2020	6981	4741.135	9220.865	3555.423	10406.577
Aug 2020	7009	4684.583	9333.417	3454.110	10563.890
Sep 2020	6496	4089.999	8902.001	2816.340	10175.660

Figure 24 – Point estimation of the forecast with 80% and 95% C.I.

Appendix B

Data preparation –

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

df = pd.read_csv(r"c:/Users/sastu/Downloads/crime.csv", encoding = 'latin-1')
df = df[['INCIDENT_NUMBER', 'YEAR', 'MONTH']]
df.MONTH = df.MONTH.astype('str')
df.YEAR = df.YEAR.astype('str')
df['MONTH'] = df['MONTH'].apply(lambda x: x.zfill(2))
df['Year_Month'] = df.YEAR + '-' + df.MONTH
df = pd.DataFrame(df.groupby(['Year_Month'])['INCIDENT_NUMBER'].count())
df = df.sort_values(by=['Year_Month'])
df = df.reset_index()
df = df.set_index('Year_Month')
df.columns = ['Count']
df_ts = df.drop(df.index[40])
df_ts = df_ts.drop(df.index[0])
df_ts.to_csv("c:/Users/sastu/Downloads/ts_boston_crime.csv", index=False)
```

Data Modeling –

```
library(TSA)
library(tseries)
library(forecast)
```

```
bos = read.csv("C:\\OSU\\Sem3\\STAT 5053\\Project\\ts_boston_crime.csv", sep = ',',
header = T)

bos.ts = ts(df, start = c(2015,7), frequency = 12)

plot(bos.ts, main = "Number of crime incidents in Boston from the year 2015 to 2018",
xlab = "Time (years)", ylab = "Number of incidents")

acf(bos.ts, lag.max = 30, main = "ACF plot of the original Boston crime data")
#seasonality is present

adf.test(bos.ts, k=7)

BoxCox.ar(bos.ts) #use original data

acf(diff(bos.ts), lag.max = 30,main = "ACF plot of the differenced original data")

bos.sa.diff = diff(df.ts, 12) #seasonal difference

plot(bos.sa.diff, main = 'Plot after taking the seasonal difference', xlab = "Time (years)")
#downward trend

adf.test(bos.sa.diff) #not stationary

acf(bos.sa.diff, main = "ACF plot of the seasonal differenced data")

bos.diff = diff(bos.sa.diff) #normal difference to make data stationary

plot(bos.diff, main = "Plot of the differenced seasonally adjusted data", xlab="Time
(years)")

adf.test(bos.diff, k = 0)

acf(bos.diff, main = "ACF plot of the differenced seasonally adjusted data")

pacf(bos.diff, main = "PACF plot of the differenced seasonally adjusted data")

eacf(bos.diff, 5,5) #ARMA(0,0)

plot(armasubsets(bos.diff, nar = 4, nma = 4)) #MA(3)


#Model

model = arima(bos.ts, order = c(0,1,0), seasonal = list(order = c(0,1,0), period = 12))

arima(bos.ts, order = c(1,1,0), seasonal = list(order = c(0,1,0), period = 12))

confint(arima(bos.ts, order = c(1,1,0), seasonal = list(order = c(0,1,0), period = 12)))

arima(bos.ts, order = c(0,1,1), seasonal = list(order = c(0,1,0), period = 12))
```

```
confint(arima(bos.ts, order = c(0,1,1), seasonal = list(order = c(0,1,0), period = 12)))
arima(bos.ts, order = c(1,1,1), seasonal = list(order = c(0,1,0), period = 12))
confint(arima(bos.ts, order = c(1,1,1), seasonal = list(order = c(0,1,0), period = 12)))

#Residual Analysis
plot(rstandard(model), type='o', main = "Checking for the constant mean and variance")
shapiro.test(rstandard(model))
qqnorm(rstandard(model), main = "Checking for normality")
qqline(rstandard(model))
acf(rstandard(model), main = "ACF plot of the residuals")
tsdiag(model, 30)

#Outlier detection
detectAO(model)
detectIO(model)

#Forecasting
get <- function (ts1,period){
  fit <- arima(ts1, order = c(0,1,0), seasonal = list(order = c(0,1,0), period = 12))
  fit$x <- ts1
  return(forecast(fit,period))
}
predictions = get(bos.ts, 24)
plot(predictions, main = "Forecast of the Boston Crime Incidents", xlab = "Time (Years)", ylab = "Number of Incidents")
predictions
```