USE CASE STUDY REPORT

GROUP NO: 19
STUDENT NAMES: MRINAL PAYANNAVAR, LI-HSUAN LIN

TIME SERIES ANALYSIS:
VIOLENT CRIME IN BOSTON

PROFESSOR: XUEMIN JIN
IE 7275: DATA MINING IN ENGINEERING

Northeastern University
College *of* Engineering

TABLE OF CONTENTS

I. BACKGROUND AND INTRODUCTION

Boston is a very large coastal city (i.e. on the ocean, a bay, or inlet) located in the state of Massachusetts. The city covers 48 square miles and has an approximate population of 673,184 as of 2016. One of the biggest issues the city has faced are the unlawful acts that take place throughout the year. The rate of crime in Boston is substantially higher when compared to national average across all American communities.

A) PROBLEM STATEMENT

Boston is more violent than New York and Seattle, but less violent than Chicago and Las Vegas, according to numbers from the FBI, based on crimes committed in 2015. For every 100,000 people, there are 7.83 daily crimes that occur in Boston. In Boston, you have a 1 in 35 chances of becoming a victim of any crime.

With such statistics, it is very natural that one might want to know, which areas are safest and which are prone to more crimes. That can help the government of Boston, to decide which areas they can assign more security or station police officers to keep a check on the surroundings.

B) GOAL OF STUDY

In this case study, we study a Time Series Analysis of the rates of violent crimes, over a period from July 2012- till date, in the city of Boston. The data has been provided by the Government of Boston, website. The study of the level of crimes, the nature and the frequency of crimes in different neighborhoods of Boston, and over the period of time will help generalize the statistics and assist in making decisions about the security measures to be taken.

C) POSSIBLE SOLUTION

The data has two parts, one part was collected from July 2012 to August 2015, which will be used as the training set. The other part has been collected from August 2015 till date. This set will be used as the validation set. Our solution includes an extensive time series analysis by plotting a time series plot to analyze the trends in different neighborhoods as well as the seasonality distribution over the years and developing ARIMA model for predicting the future cases.

II. DATA EXPLORATION AND VISUALIZATION

The data can be used to group similar crimes in categories and analyze them to find what areas are prone to what type of crime, the time of the year when most crimes happen, the crimes that can be curbed easily.

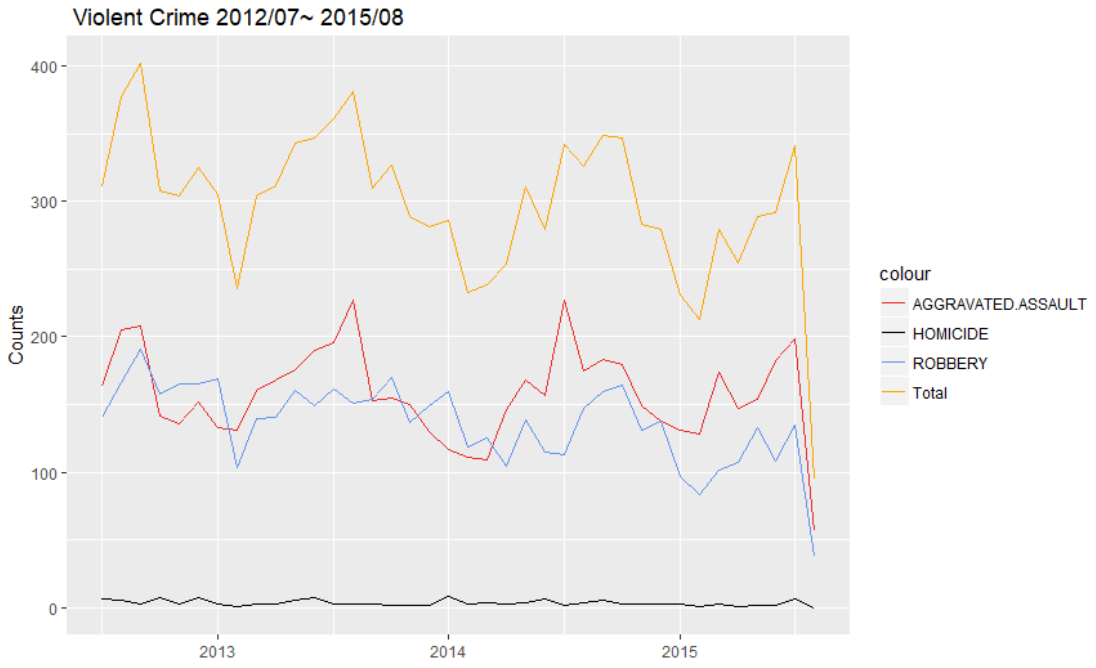
In this case, we study the trends of violent crimes, that include a combination of crimes like, aggravated assault, homicide, and robbery.

The following are the plots used to visualize the seasonality and trends of occurrences of the violent crimes, throughout the years and in different neighborhoods.

A) Time-Series Plot:

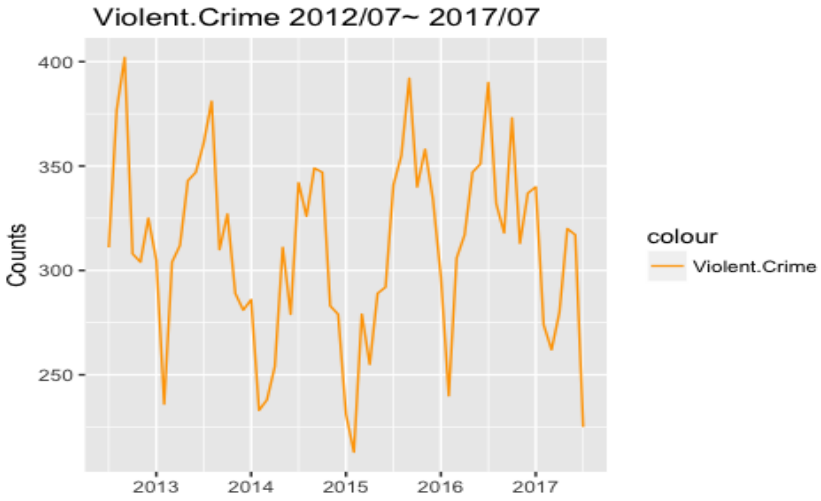
A time series graph is a great way to evaluate patterns and behavior in data over time. It plots the year of the violent crime against the frequency of the violent crime occurrences. The line chart helps us delineate patterns in terms of trends, seasonality and find the level i.e. the average rate of the violent crime in Boston.

The time series plot of the training set, from year 2012 to year 2015, is derived as follows,

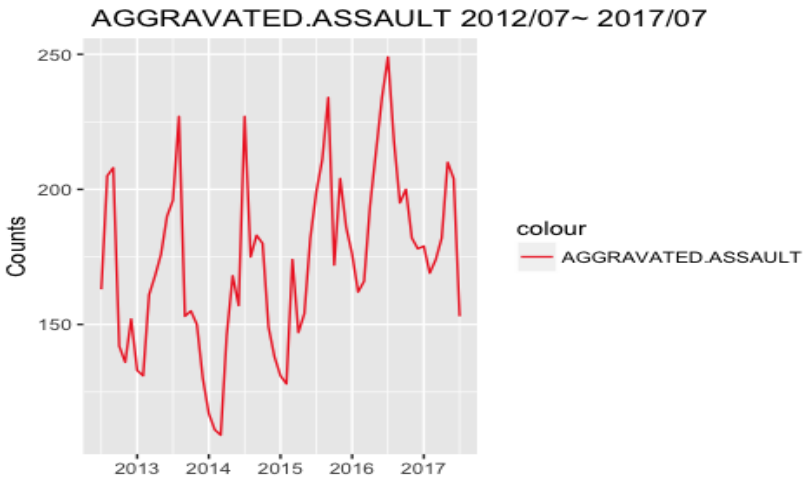


From the time series plot, we could see that violent crime increases during the summer, and during the winter, it drops.

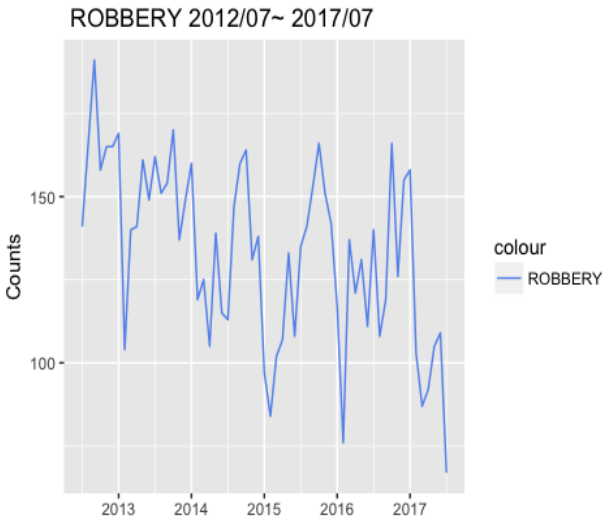
The amount of homicides in Boston is not that significant, as compared to the robbery and aggravated assaults crimes. In totality, the level of violent crimes is around 350 cases per year. An improvement can be noted, as the rate of violent crime is on a decrease as seen in 2015 compared to 2012.

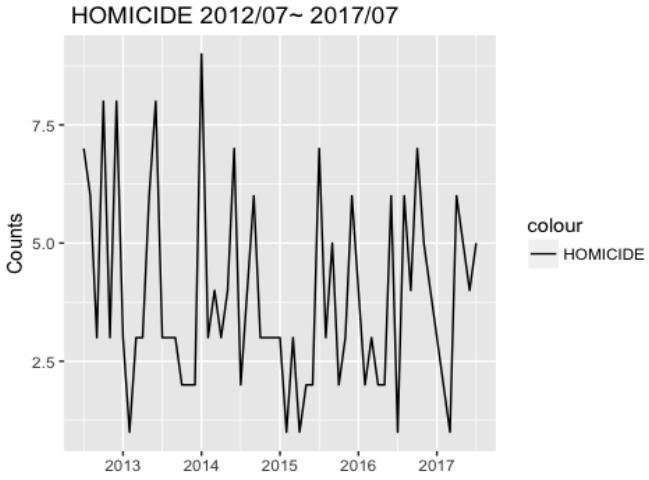


The time series plot for aggravated assault can be seen as below, where the level is 175 assaults per year for 5 years. The U-shaped trend is slightly discernable for every year, with seasonality having peak values during summer (May, June, July).



Similarly, for Robbery and Homicide.

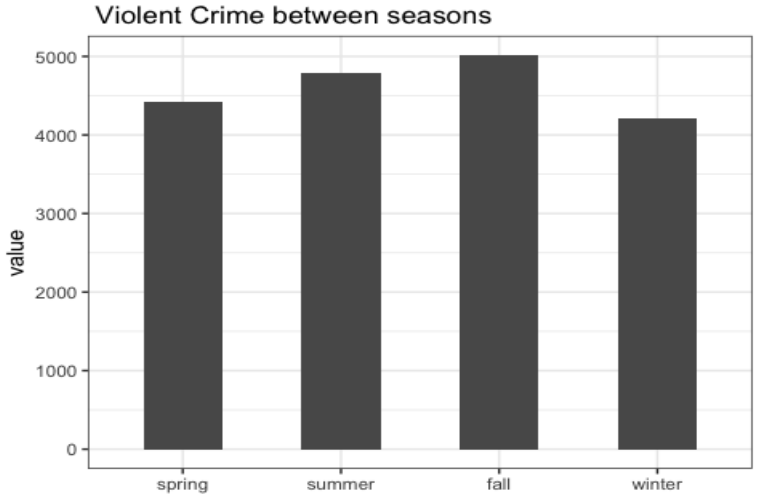




B) BAR PLOT

1. Violent Crimes between seasons:

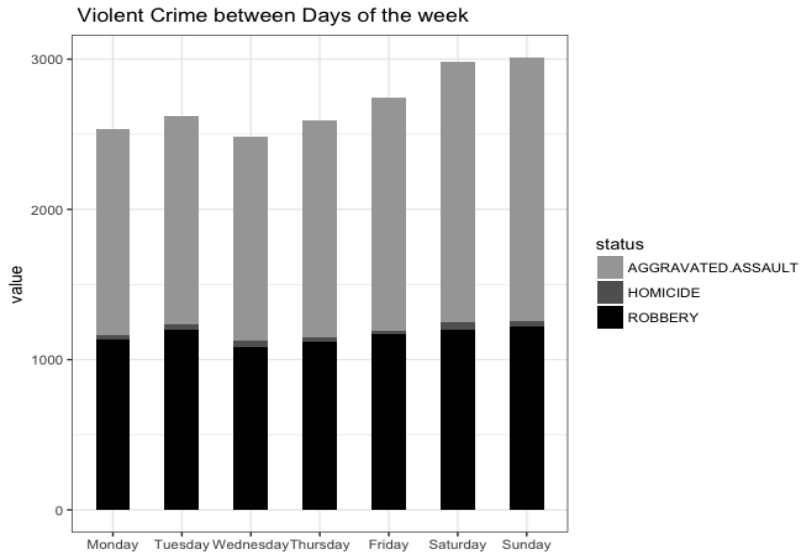
A year has been divided in 4 categories representing seasons namely, Spring, Summer, Fall, Winter. A box plot is a chart representing the frequency violent crimes occurring in of these categories. Deriving a box plot of violent crimes based on seasons gives the following result,



According to the bar plot, it is obvious that summer and fall has a higher crime rate compare to winter and spring.

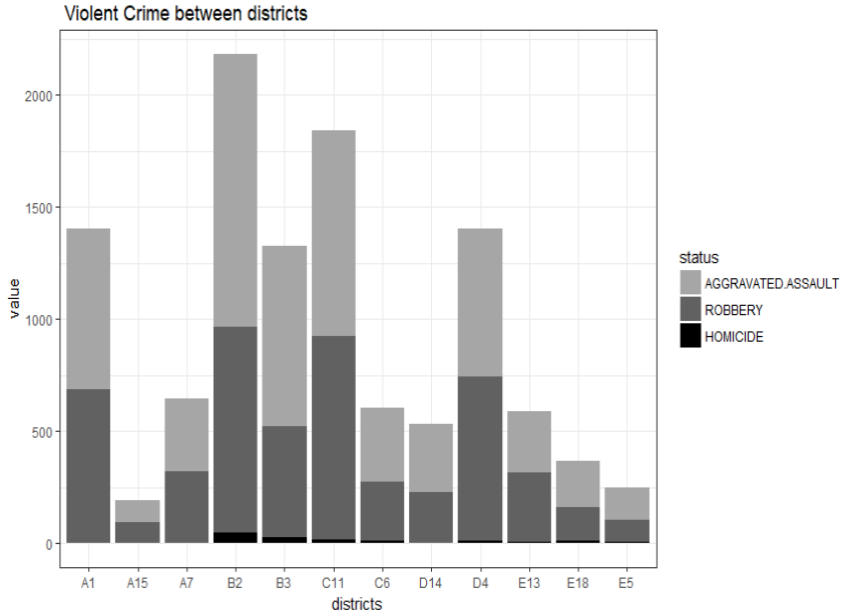
2. Violent crimes between days of the week

The bar plot of violent crime between days of the week shows that weekends possess more incidents. However, the rest of the week days still have incidents around 2500.



3. Violent crimes based on neighborhoods

From the bar plot, B2(Roxbury), C11(Dorchester), A1(Dorchester) and D4(South End) are the most dangerous districts. Also, the plot shows that B2(Roxbury) holds the most homicide cases.



A1= Downtown
A15= Charlestown
A7= East Boston
B2= Roxbury
B3= Mattapan
C11= Dorchester
C6= South Boston
D14= Brighton
D4= South End
E13= Jamaica Plain
E18= Hyde Park
E5= West Roxbury

III. Data Preparation and Preprocessing

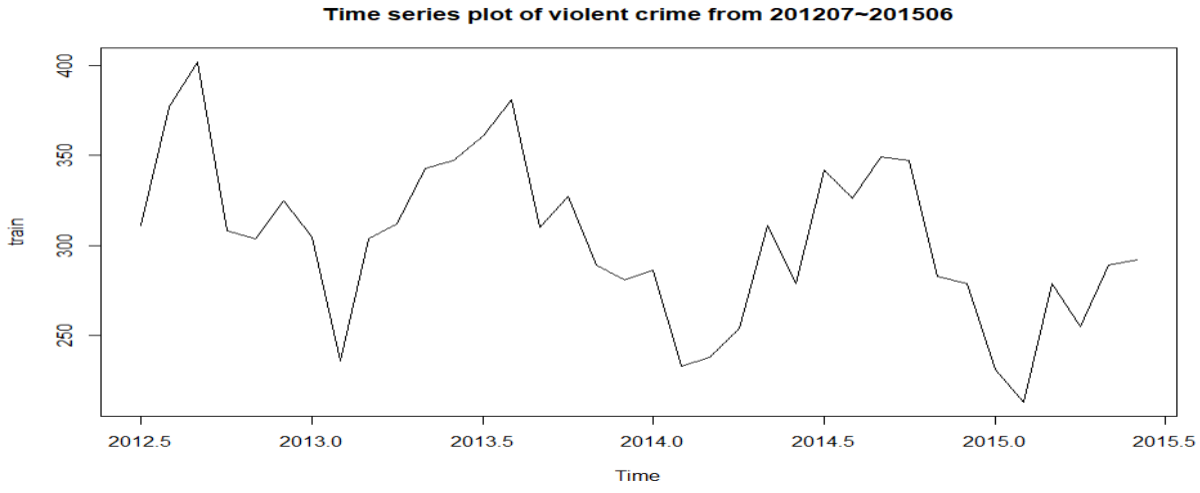
This Dataset contains 268000 instances and 20 attributes. The attributes include, nature of crime, incident description, district, the time and date of the crime and related columns. The dataset will be divided in two parts, one will be the training set and the other will be the validation set.
The training set contains data from the years 2012 to 2015. The validation set contains data from year 2015 to 2017.

The pre-processing step of developing the predictive model is categorizing the types of crimes and extract the types of violent crime, i.e., aggravated crime, robbery and homicide. The rate of these crimes is then split into different sets in terms of days of the week, seasons of the year, neighborhoods in Boston.

IV. Data Mining Techniques and Implementation

1. Time Series plot of the training set

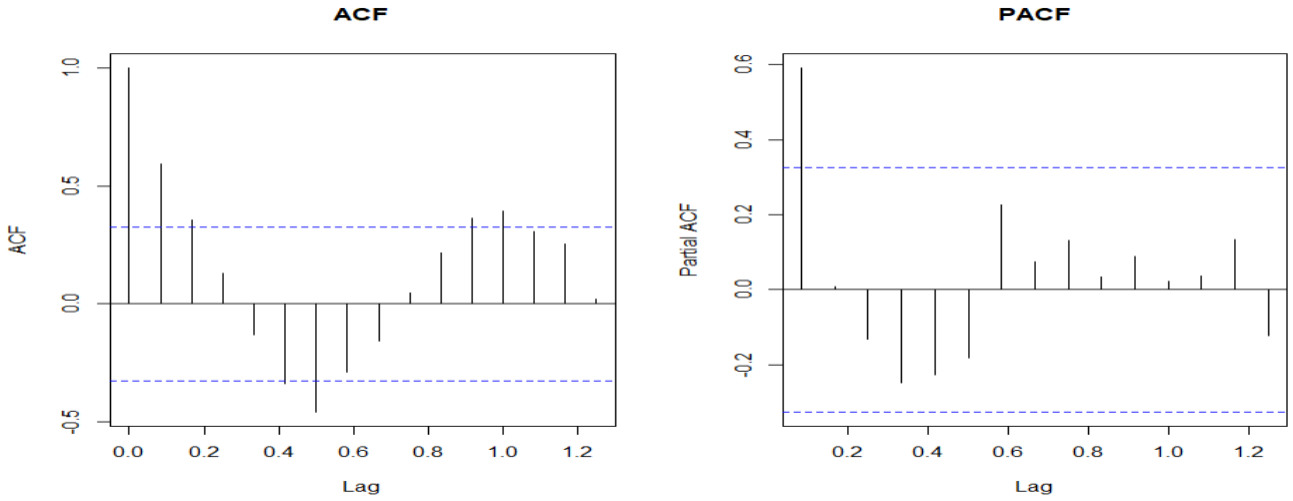
A time series plot has 4 components, namely, level, trend, seasonality and noise. Level is the average in the series, trend is the change between consecutive periods, and seasonality describes a short-term cyclical behavior of the series.
The function plot.ts() is used to generate a time-series plot.



There is a peak every summer, and a trough every winter. This time series could probably be described using an additive model, because of the approximate consistent seasonality in terms of size and does not to depend on the level of the time series. The trend of this model is a linear one.

2. ACF AND PACF PLOT

To select an appropriate ARIMA model, values of p and q for an ARIMA(p,d,q) model need to be selected. To do this the correlogram and partial correlogram of the stationary time series. are examined.
The function **ACF** computes (and by default plots) an estimate of the autocorrelation function of a multivariate time series data. From the graph we see that, the auto-correlation exceed at lag 1, 2
Function **PACF** computes (and by default plots) an estimate of the partial autocorrelation function of a multivariate time series. The partial correlation exceed the lags 1.



3. ARIMA Model Selection

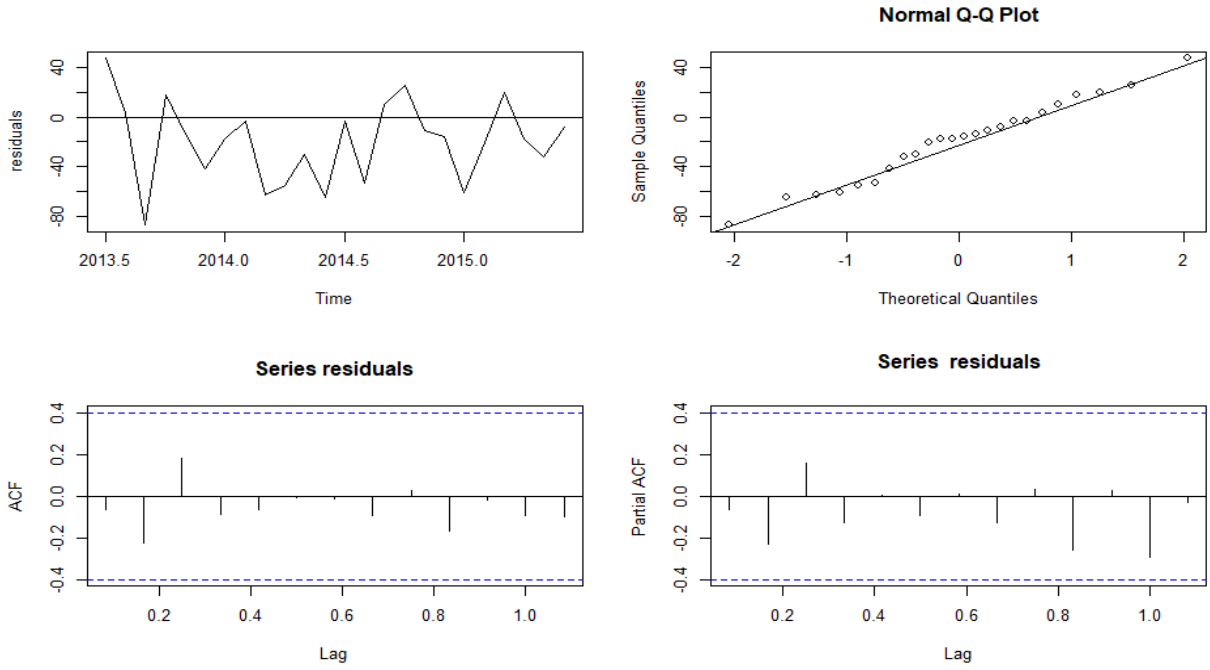
Autoregressive Integrated Moving Average (ARIMA) models include an explicit statistical model for the irregular component of a time series, that allows for non-zero autocorrelations in the irregular component.
Using the principal of parsimony, we select the seasonal ARIMA model with Seasonal AR order=1, seasonal differencing=1 and repeating seasonal period=12(12 months period).

```
Series: train
ARIMA(0,0,0)(1,1,0)[12] with drift

Coefficients:
          sar1    drift
      -0.5737  -1.3507
s.e.   0.1938   0.4058

sigma^2 estimated as 977.1:  log likelihood=-118.02
AIC=242.04   AICc=243.24   BIC=245.57
```

4. Residual plots



5. Test for Stationarity

To check whether the sequence is stationary, we perform the Phillips-Peron Unit Root test.
H_0 = The Sequence is explosive
H_1 = The Sequence is stationary
If p-value > 0.05, Do not reject H_0
In our model, we reject H_0, hence our model is stationary.

6. Test for Normality

To check whether the residuals are normally distributed, the SHapior- Wilk Test is performed.
H_0: The population is normally distributed
H_1: The population is not normally distributed
In our model,
P=0.9406 >0.05, does not reject H_0.
There is enough evidence to prove that the residuals are normally distributed

7. Test for independence of residuals.

Box- Pierce test is performed to check for independence of residuals.
H_0: The data are independently distributed
H_1: The data are not independently distributed; they exhibit serial correlation.
In our model,
P=0.7587 and P=0.744 >0.05, both tests does not reject H_0.
There is enough evidence to prove that the residuals are independently distributed.

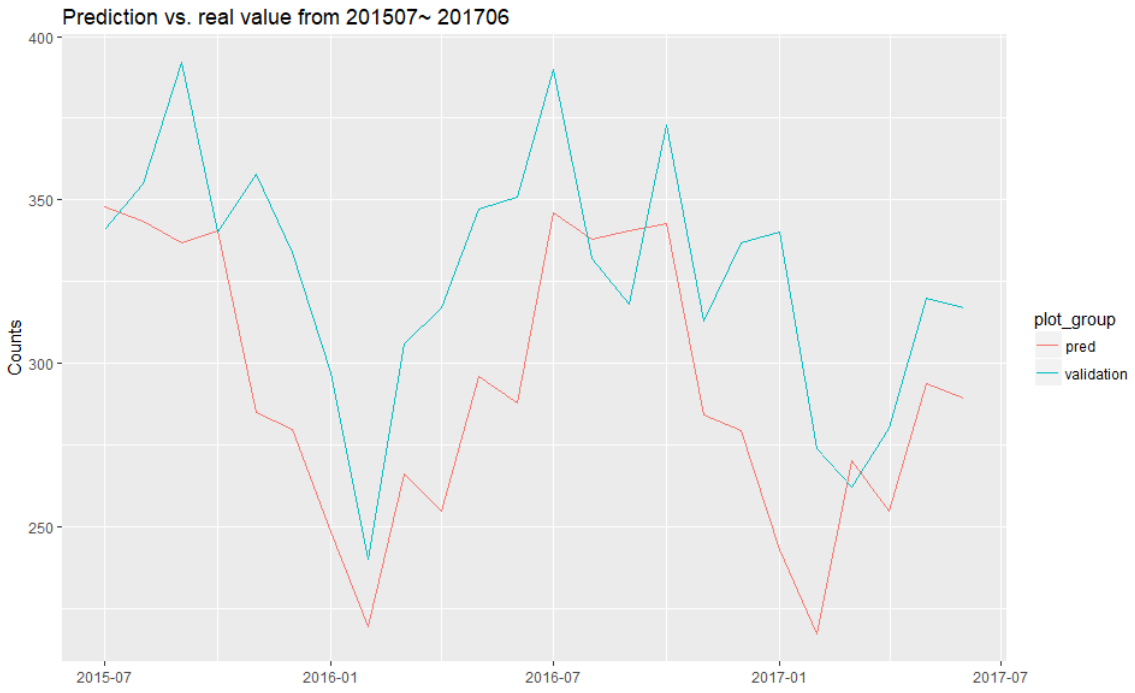
V. Performance Evaluation

Since this study is performing time series analysis, the data is divided into the training set with the first 36 consecutive months and the validation set with the following 24 months. Then, we use the validation set, that is, the crimes between 07/2015 to 06/2017 to evaluate the predictive performance of the training set. Also, the predictive performance of the ARIMA model will be evaluated by the MAE and MAPE performance measures.

Based on the ARIMA (0,0,0)(1,1,0)[12] model, we predicted the crimes between 07/2015 to 06/2017 as following:

	Validation	pred
Jul 2015	341	347.9837
Aug 2015	355	343.3211
Sep 2015	392	336.7178
Oct 2015	340	340.7014
Nov 2015	358	284.8896
Dec 2015	334	279.6299
Jan 2016	297	248.3211
Feb 2016	240	219.2986
Mar 2016	306	266.0879
Apr 2016	317	254.6851
May 2016	347	295.9284
Jun 2016	351	287.9059
Jul 2016	390	346.0992
Aug 2016	332	337.8662
Sep 2016	318	340.5858
Oct 2016	373	342.6850
Nov 2016	313	284.2945
Dec 2016	337	279.4315
Jan 2017	340	242.8662
Feb 2017	274	217.3150
Mar 2017	262	270.1543
Apr 2017	280	254.7843
May 2017	320	293.7465
Jun 2017	317	289.1953

From the time series plot, we could see that our prediction basically follows the actual value. Moreover, from MAPE, it shows that our mean absolute percentage error is 0.116506, which means that our prediction is off by, on an average, 11% of the number of the actual violent crime.



VI. Discussion and Recommendation

This case study uses time series analysis to forecast violent crime incidents of Boston city. The final ARIMA model that has been chosen includes a seasonal AR (1) term, a one-time seasonal differencing and a seasonal period S=12 (a twelve-month period).

With a 11% MAPE, our model can be considered as a model with good predictive performance. The result of the model shows that violent crime of Boston displaying winter troughs and summer peaks. This information could help police departments to deploy their officers and make the city safer. However, based on the dataset from Boston police department, we only took consideration of the seasonal pattern of violent crime.

To make a more precise prediction, further research could be focused upon the spatial pattern, which is, the crime rates between the neighborhoods of violent crime.

VII. Summary

This case study mainly sheds light on violent crime in Boston. The study begins by exploring violent crimes with descriptive statistics such as line chart, bar chart and box plot, based on the data provided by Boston Police Department from July 2012 to June 2017.

The case study goes on to focus on the seasonal pattern of violent crime and utilize time series analysis to effectively forecast the number of crime incidents.

The final seasonal ARIMA model with a 11% mean absolute percentage error could be considered as a model with well predictive performance.

The result of the model shows that violent crime of Boston displaying winter troughs and summer peaks. With such predictive model, it could assist Boston police department in making decisions about security controls.

Appendix: R Code for use case study

```
# Data visualization
library(xlsxjars)
library(xlsx)

violent <-read.xlsx("E:\\NEU\\IE 7275 Data mining\\R\\case\\violent crime.xlsx", sheetIndex =
3, header = T,stringsAsFactors = F)

str(violent)

# Parsing the date format
library(zoo)
violent$DATE1 <- as.yearmon(violent$DATE,"%Y/%m")
colnames(violent)[5] <- "Violent.Crime"

# Line chart

library(ggplot2)
# Time series plot using ggplot
ggplot(violent, aes(x=DATE1)) +
  geom_line(aes(y=AGGRAVATED.ASSAULT, color="AGGRAVATED.ASSAULT"))+
  geom_line(aes(y=ROBBERY, color="ROBBERY"))+
  geom_line(aes(y=HOMICIDE, color="HOMICIDE"))+
  geom_line(aes(y=Violent.Crime, color="Violent.Crime"))+
  scale_color_manual(values = c("firebrick2", "black", "cornflowerblue", "orange"))+
  labs(title =" Violent Crime 2012/07~ 2017/07", x = "", y = "Counts")

# Line plot for aggregate violent crime
ggplot(violent, aes(x=DATE1)) +
  geom_line(aes(y=Violent.Crime, color="Violent.Crime"))+
  scale_color_manual(values = c("orange"))+
  labs(title =" Violent.Crime 2012/07~ 2017/07", x = "", y = "Counts")

# Line plot for aggravated assault over time
ggplot(violent, aes(x=DATE1)) +
  geom_line(aes(y=AGGRAVATED.ASSAULT, color="AGGRAVATED.ASSAULT"))+
  scale_color_manual(values = c("firebrick2"))+
  labs(title =" AGGRAVATED.ASSAULT 2012/07~ 2017/07", x = "", y = "Counts")
```

```
# Line plot for robbery over time
ggplot(violent, aes(x=DATE1)) +
  geom_line(aes(y=ROBBERY, color="ROBBERY"))+
  scale_color_manual(values = c("cornflowerblue"))+
  labs(title =" ROBBERY 2012/07~ 2017/07", x = "", y = "Counts")


# Line plot for homicide over time
ggplot(violent, aes(x=DATE1)) +
  geom_line(aes(y=HOMICIDE, color="HOMICIDE"))+
  scale_color_manual(values = c("black"))+
  labs(title =" HOMICIDE 2012/07~ 2017/07", x = "", y = "Counts")



# Bar plot for violent crime over the seasons
str(violent)
sea <- c("spring","summer","fall","winter")
ggplot(violent.season, aes(x=season, y=Violent.crime,))+
  theme_bw()+
  geom_bar(stat = "identity",position="stack", width = 0.5)+
  scale_x_discrete(limits = sea)+
  scale_fill_manual(values=c('#fee8c8','#fdbb84','#e34a33'))+
  labs(title =" Violent Crime between seasons", x = "", y = "value")


# Bar plot for violent crime over the week
library(reshape2)
dat = melt(violent.week[-5], id.var="DAY_WEEK", variable.name="status")
week.or <- c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")
ggplot(dat, aes(x=DAY_WEEK, y=value, fill=status))+
  theme_bw()+
  geom_bar(stat = "identity",position="stack", width = 0.5)+
  scale_x_discrete(limits = week.or)+
  scale_fill_manual(values=c("gray65","gray38","black"))+
  labs(title =" Violent Crime between Days of the week", x = "", y = "value")
```

```
# Bar plot for violent crime in districts
library(reshape2)
dat = melt(district[-5], id.var="district", variable.name="status")
ggplot(dat, aes(x=district, y=value, fill=status))+
  theme_bw()+
  geom_bar(stat = "identity",position="stack")+
  scale_fill_manual(values=c(AGGRAVATED.ASSAULT="gray65", ROBBERY="gray38",
HOMICIDE="black"))+
  labs(title =" Violent Crime between districts", x = "districts", y = "value")


# Time series plot using ts()
Violent.Crime=ts(violent$Violent.Crime)


# Data partition 60%40%
violent.ts <- ts(Violent.Crime, start = c(2012,7), end = c(2017,6), frequency = 12)
train <- window(violent.ts, start = c(2012,7), end = c(2015,6), frequency = 12)
validation <- window(violent.ts, start = c(2015,7), end = c(2017,6), frequency = 12)


# Time series plot for training set
plot.ts(train, main="Time series plot of violent crime from 201207~201506")


# ACF PACF plot
opar <- par(no.readonly = TRUE)
par(mfrow = c(1,2))
acf(train, main="ACF")
pacf(train,main="PACF")
par(opar)


# ARIMA model selection
library(forecast)
auto.arima(train)

#ARIMA(0,0,0)(1,1,0)[12]

train_model <- arima0(train, order = c(0,0,0),seasonal = list(order =c(1,1,0), period=12))
```

```
# Residual diagnostic
residuals <- train_model$residuals
par(mfrow = c(2,2))
ts.plot(residuals)
abline(h=0)
qqnorm(residuals)
qqline(residuals)
acf(residuals)
pacf(residuals)


# Test for Stationarity
library(TSA)
adf.test(residuals)
pp.test(residuals)
kpss.test(residuals)
# Test for normality of residuals
shapiro.test(residuals)



# Test for independence of residuals
Box.test(residuals, type = "Box-Pierce")
Box.test(residuals, type = "Ljung-Box")



# Developing a predictive model
pred <- predict(train_model,n.ahead = 24)
pred$pred

pred.real.ts <- ts(data.frame(validation, pred=pred$pred),start = c(2015,7), end = c(2017,6),
frequency = 12)
pred.real.ts

library(ggfortify)
autoplot(pred.real.ts, facets = FALSE,ts.linetype = 1,xlim = , ylab = "Counts", main =
"Prediction vs. real value from 201507~ 201706")


# Performance evaluation
library(Metrics)
mae(validation,pred$pred)
mape(validation,pred$pred)
```