Time Series Analysis of United States Shootings

Samantha Lee
Professor Raya Feldman
PSTAT 274
Perm: 8112732
Section: T 7 PM

Abstract

Shootings have been a prevalent crime of violence in the United States, ranging from Columbine to Sandy Hill, to the most recent Las Vegas shooting. Gun violence has been seen to be one of the most common reasons for homicides in the country [1]. With data taken from a Github repository, a time series forecast has been done to predict the weekly deaths due to shootings from 2013 to 2015. This project utilizes autoregressive integrated moving average (ARIMA) models to forecast weekly deaths in the U.S due to shootings from 2013 to 2015.

The data was aggregated into weekly observations and transformed into a time series. It was then differenced and the autocorrelations and partial autocorrelations were examined for model selection. To determine if the model was a good fit, the model was used to predict the last thirteen observations and compared to the actual values. A spectral analysis of the data was the last step.

These forecasts are meant to anticipate future gun violence acts and thus prevent further occurrences.

Introduction

Data

The data is acquired off github.com from Buzzfeed as the recorded shootings in the United States from 2013 to 2015. Some variables include date, author and article source, number killed, and number injured. It was originally recorded daily data, but we aggregated the observations into weekly data to make the time series frequency more applicable. This is publicly available at https://github.com/BuzzFeedNews/2015-12-mass-shooting-intervals [2].

Software

All statistical analysis was done using the Rstudio integrated development environment. The included software libraries were tseries, forecast, tsa, GeneCycle, MASS, dplyr, and lubridate.

Results

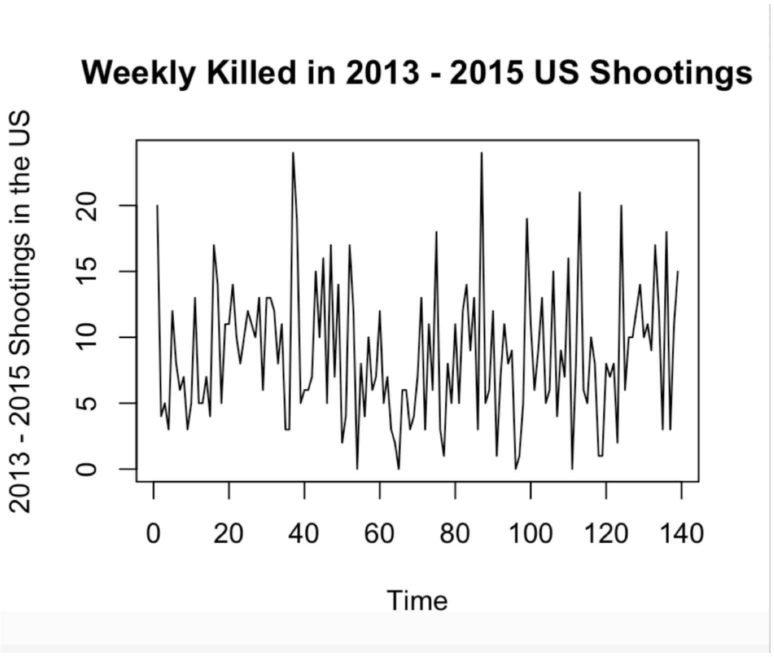
Using ARIMA models, we cannot conclude that deaths due to shootings in the United States, unfortunately, can be accurately forecasted. There is also a lack of dominant frequencies in the spectral analysis.

Summary of Analysis

After loading the data into R, the last thirteen weeks of data are removing in order to compare the observations against the forecasts. After exploratory analysis, we see that the data requires differencing to stabilize variance and remove trend. Next, to identify appropriate models, the autocorrelations (ACF) and partial autocorrelations (PACF) of the stationary series are examined. After constructing potential models, we conduct model diagnostic checking to make sure each meets the assumptions of ARIMA. When the 'best' model is chosen, it is used to forecast future observations and compared against the removed observations to evaluate model performance. The final step is a spectral analysis of the data.

Analysis

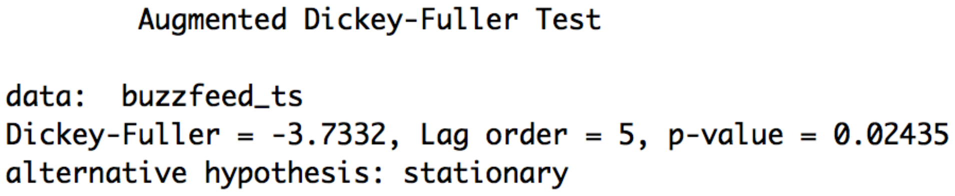
We begin by loading the formatted data into R and examining the time series to make some observations.



There is volatility in the data where more shootings occur – this affects the stationarity of the time series, so we will try to difference the data to reduce variance.

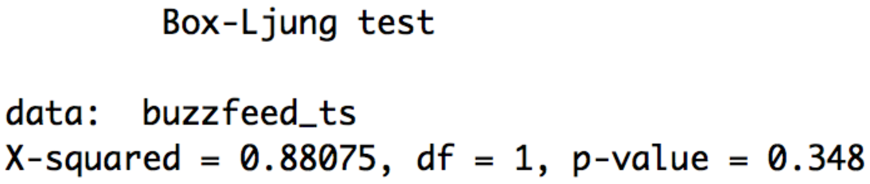
The last twelve weeks of observations have also been removed to later compare against the forecasted data.

The variance is 27.11, which we will reference later to see if it helps with volatility.



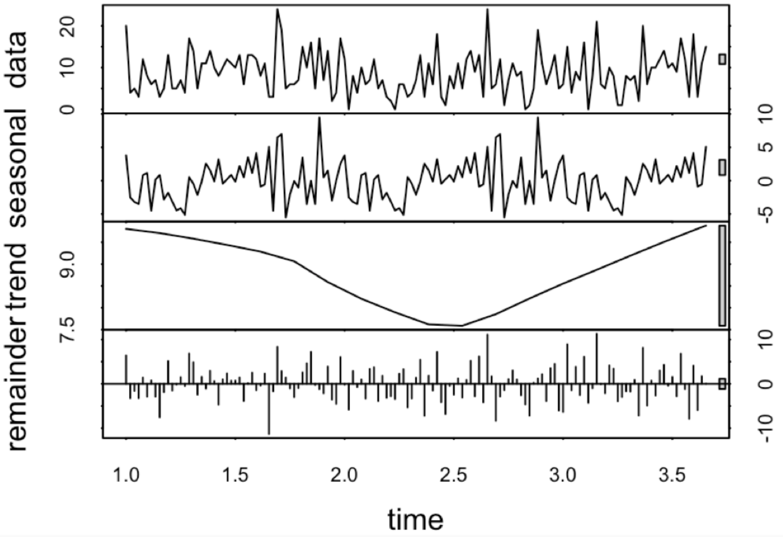
```
	Augmented Dickey-Fuller Test

data:  buzzfeed_ts
Dickey-Fuller = -3.7332, Lag order = 5, p-value = 0.02435
alternative hypothesis: stationary
```

The Augmented Dickey-Fuller test is frequently used to test stationary in time series analyses. Small p-values suggest the data is stationary and does not need to be differenced. Since our p-value is smaller than 0.05, we are able to reject, but move on to a different test for confirmation.

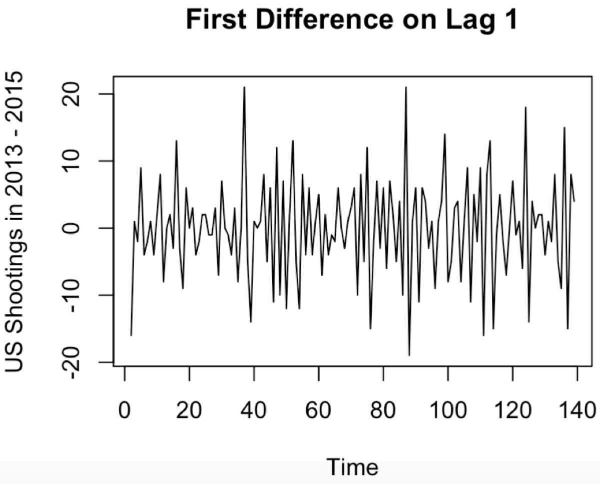


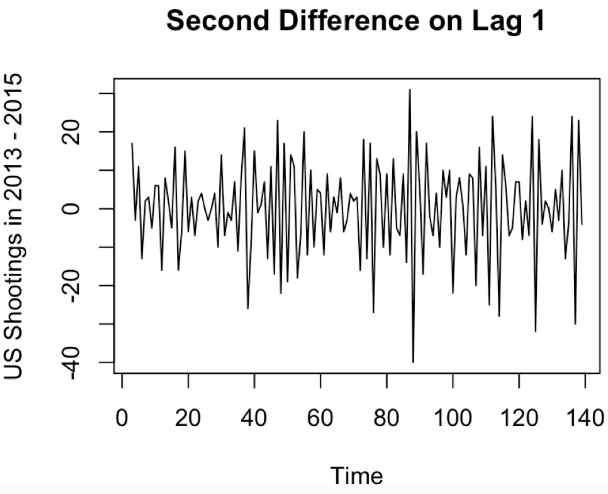
```
	Box-Ljung test

data:  buzzfeed_ts
X-squared = 0.88075, df = 1, p-value = 0.348
```

The Box-Ljung test is a second test used to confirm stationary at lags 1 – 20. In this case, there is not significant evidence to show that the data is stationary. Due to the lack of consistency between the two tests, we will decompose the data to see if there is anything we can do to further confirm the need for transformations or differencing.

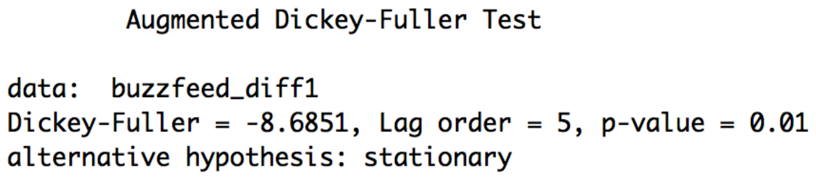


By decomposing the data with the stl() function, we can see the additive trend, seasonal, irregular components. There is a very obvious quadratic trend in the pattern. This tells us that we need to difference on lag 1 twice to try to mitigate this.

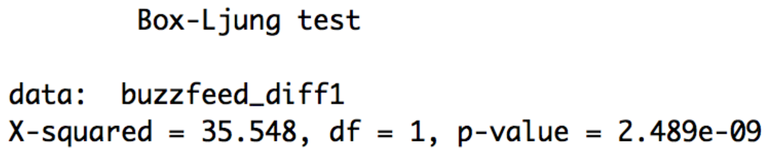




Looking at the plot for the first difference and second difference, we can see at first glance at the data looks more stationary. However, the variance taken on the second difference at lag 1 has increased to 127.597, in comparison to the first difference (0.577), which is a very obvious case of overdifferencing. We will take the first difference at lag 1, and by looking at the ADF test and the Ljung-Box test, we have shown stationarity with this step.

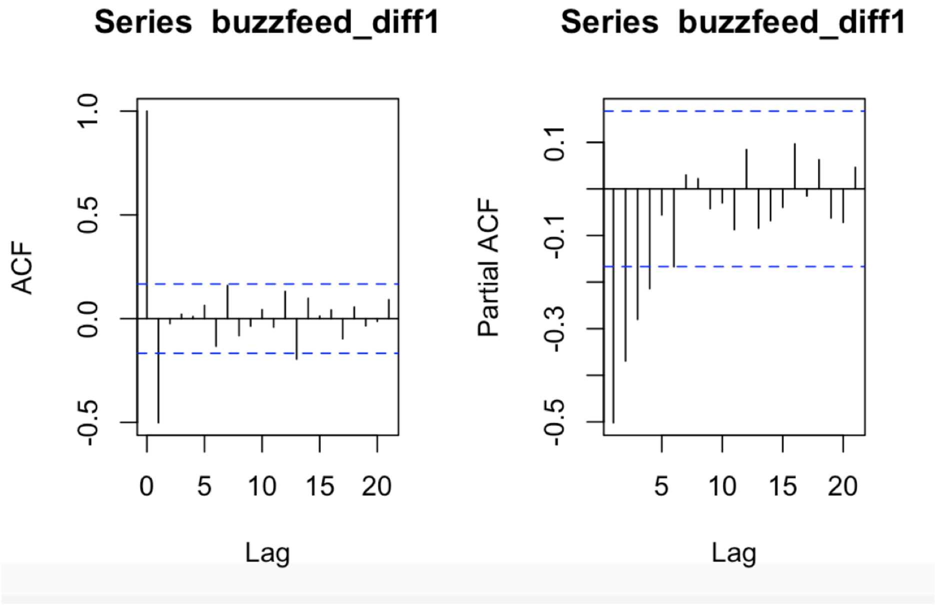


```
          Augmented Dickey-Fuller Test

data:  buzzfeed_diff1
Dickey-Fuller = -8.6851, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```



```
        Box-Ljung test

data:  buzzfeed_diff1
X-squared = 35.548, df = 1, p-value = 2.489e-09
```

Our next step is to examine the autocorrelations (ACF) and partial autocorrelations (PACF).



Since the spikes decay within the significant zone for both ACF and PACF plots, we can conclude that residuals are random with no information, so our ARIMA model is working fine.

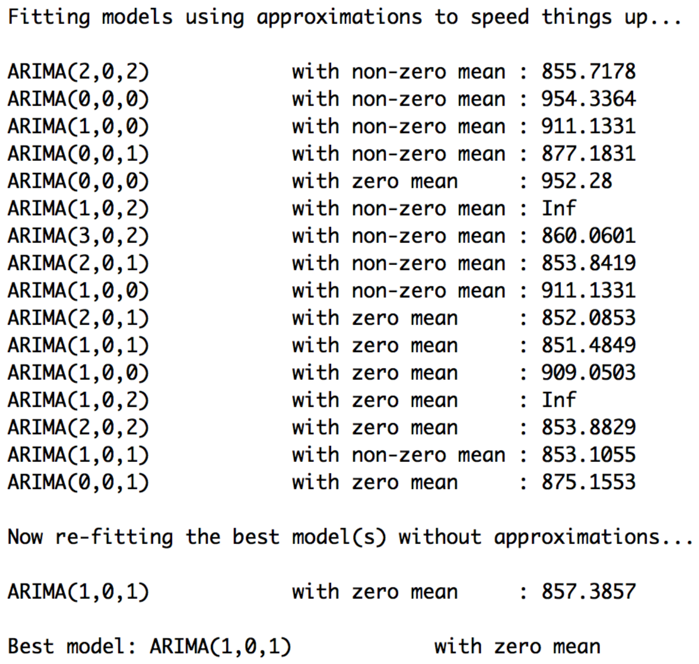
ACF plots display correlation between a series and its lags. The ACF cuts off after the second lag and exhibits exponential decay, which gives us reason to believe there is a moving average component of MA(2). We will also explore the possibility of the spike at lag 14 to be of significance.

Partial autocorrelation plots (PACF) display correlation between a variable and its lags that is not explained by previous lags. There is a pattern of decay in the partial ACF (PACF) that cuts off after lag 4. This exhibits an AR(4) component that we will consider.

Models

We will then move forward with model selection.

After running the auto.arima() function on the data, the best output model is ARIMA(1,0,1). The model corresponds to AR(1) and MA(1) components. This makes sense since there are significant spikes at lag 1 for both the ACF and the PACF, but it is suprising that the Integrated aspect is a 0, since we had differenced once at lag 1. In addition, we will explore the components explained above to see if we can acquire a better model for forecasting that take into account the other spikes that are seen in the ACF and PACF. With this base, we will continue to test other models.



```
Fitting models using approximations to speed things up...

ARIMA(2,0,2)           with non-zero mean : 855.7178
ARIMA(0,0,0)           with non-zero mean : 954.3364
ARIMA(1,0,0)           with non-zero mean : 911.1331
ARIMA(0,0,1)           with non-zero mean : 877.1831
ARIMA(0,0,0)           with zero mean     : 952.28
ARIMA(1,0,2)           with non-zero mean : Inf
ARIMA(3,0,2)           with non-zero mean : 860.0601
ARIMA(2,0,1)           with non-zero mean : 853.8419
ARIMA(1,0,0)           with non-zero mean : 911.1331
ARIMA(2,0,1)           with zero mean     : 852.0853
ARIMA(1,0,1)           with zero mean     : 851.4849
ARIMA(1,0,0)           with zero mean     : 909.0503
ARIMA(1,0,2)           with zero mean     : Inf
ARIMA(2,0,2)           with zero mean     : 853.8829
ARIMA(1,0,1)           with non-zero mean : 853.1055
ARIMA(0,0,1)           with zero mean     : 875.1553

Now re-fitting the best model(s) without approximations...

ARIMA(1,0,1)           with zero mean     : 857.3857

Best model: ARIMA(1,0,1)           with zero mean
```

The chosen models below were similar to the ones generated from the auto.arima() function as well as from the analysis of the ACF and PACF graphs.

AIC, which stands for Akaike Information Criterion, is a method of choosing between competing models via goodness of fit and simplicity of the model (less model parameters the better). It is calculated as:

$$-2 * \ln(L) + 2 * p$$

where L is the maximized value of the log-likelihood and p is the number of parameters in the model. This is how the values in the table are calculated. The lower the AIC value the better.

Another metric of choosing between competing models is the Bayesian Information Criterion (BIC). BIC tries to find the true model among the candidates and can be calculated as:

$$-2 * \ln(L) + \ln(N)*p$$

BIC is dependent on the number of observations. Again, we prefer the minimized BIC [3].

Model	**p**	**d**	**q**	**AIC**	**BIC**
1	3	1	2	863.56	883.078
2	1	1	14	869.93	921.698
3	1	1	2	858.21	874.848
4	2	1	14	867.5	922.189
5	4	1	2	865.55	887.992
6	4	1	14	875.24	932.716
7	4	0	1	860.75	883.237
8	1	1	1	912.97	923.735
9	2	1	4	865.86	888.302
Auto.arima	1	0	1	857.21	865.98

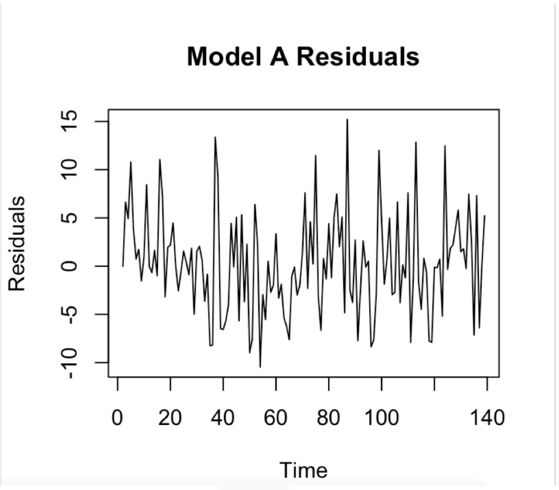
So with this table, we can see that the best models with the lowest AIC and BIC are

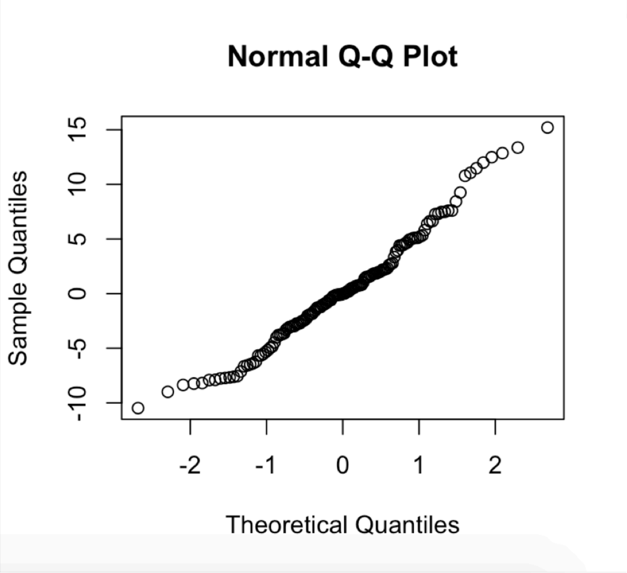
a. ARIMA(1, 1, 2)
b. ARIMA(3, 1, 2)
c. ARIMA(1, 0, 1) (auto.arima)
d. ARIMA(2, 1, 4)

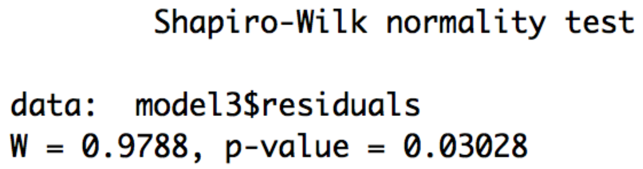
Now we will run residual diagnostics to see which model will be used for forecasting.

Residual Diagnostics

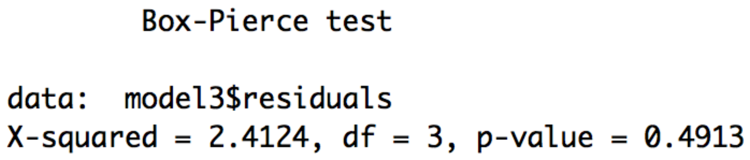
Model A


Model A Residuals

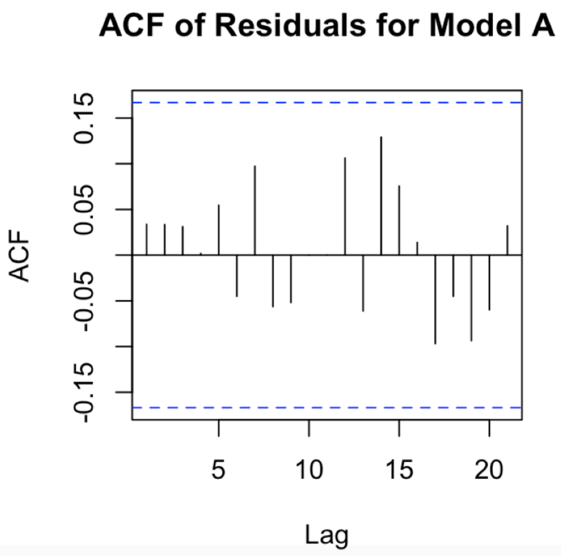

Normal Q-Q Plot

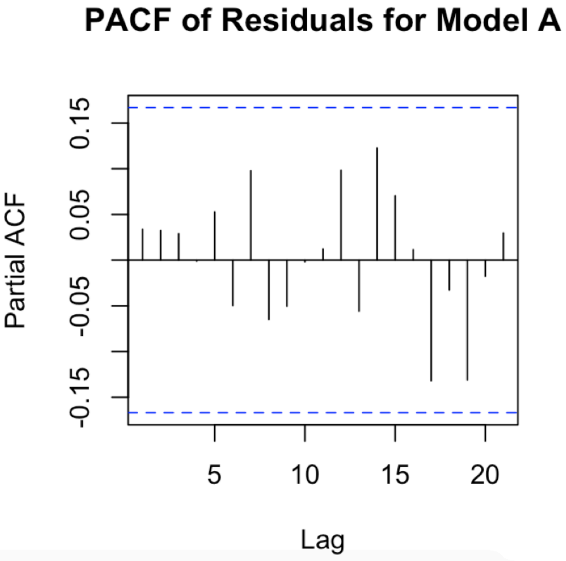


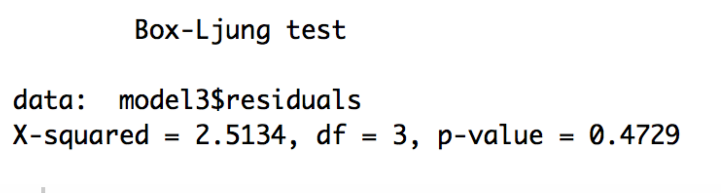
```
	Shapiro-Wilk normality test

data:  model3$residuals
W = 0.9788, p-value = 0.03028
```



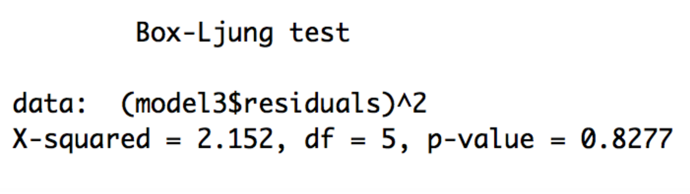
```
	Box-Pierce test

data:  model3$residuals
X-squared = 2.4124, df = 3, p-value = 0.4913
```


ACF of Residuals for Model A


PACF of Residuals for Model A



```
        Box-Ljung test

data:  model3$residuals
X-squared = 2.5134, df = 3, p-value = 0.4729
```



```
        Box-Ljung test

data:  (model3$residuals)^2
X-squared = 2.152, df = 5, p-value = 0.8277
```

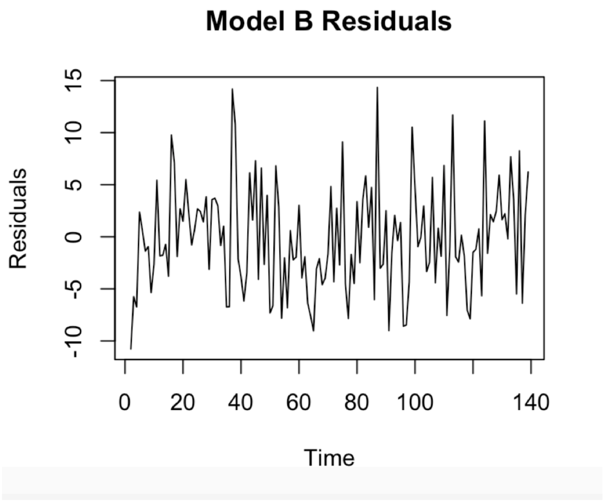
*Squaring the Box-Ljung test becomes the McLeod-Li test.

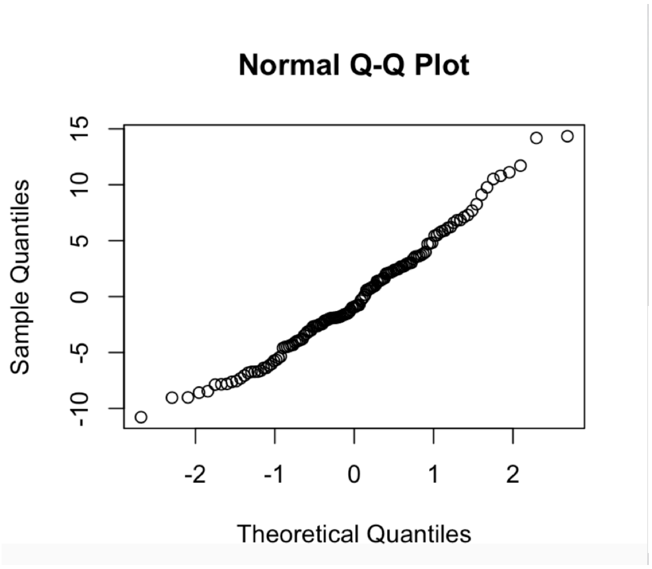
Model A fails the Shapiro-Wilk normality test since the p-value > 0.05. The ACF and PACF plots do resemble white noise and the Q-Q plot has the straight pattern of a normality plot.

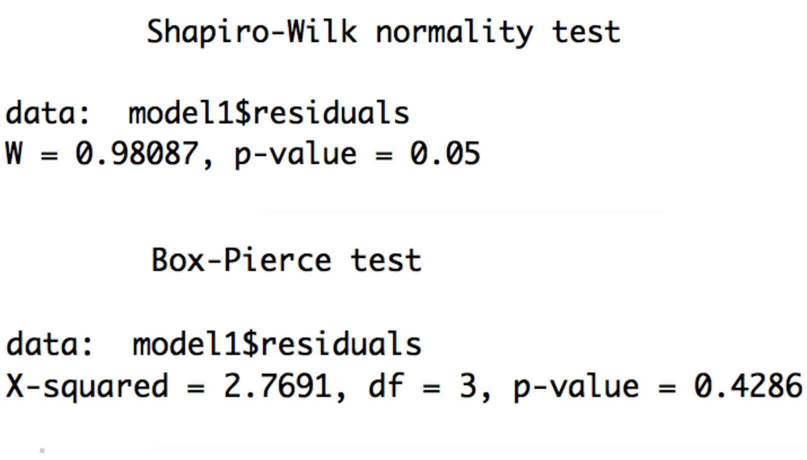
The Box-Pierce test and the Ljung-Box tests both test whether any of a group of autocorrelations of a time series differ from zero. The McLeod-Li test is used for autoregressive conditional heteroskedasticity in the model [4]. Here, we can see that all p-values are greater than 0.05, which is adequate.

Besides the result of the Shapiro-Wilk test, we are fairly confident about the normality of this model. We will move onto the next one.

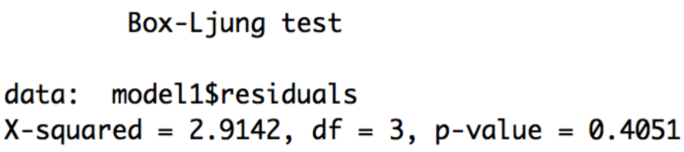
Model B



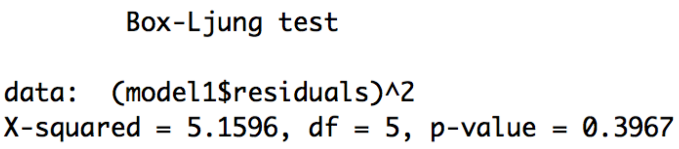




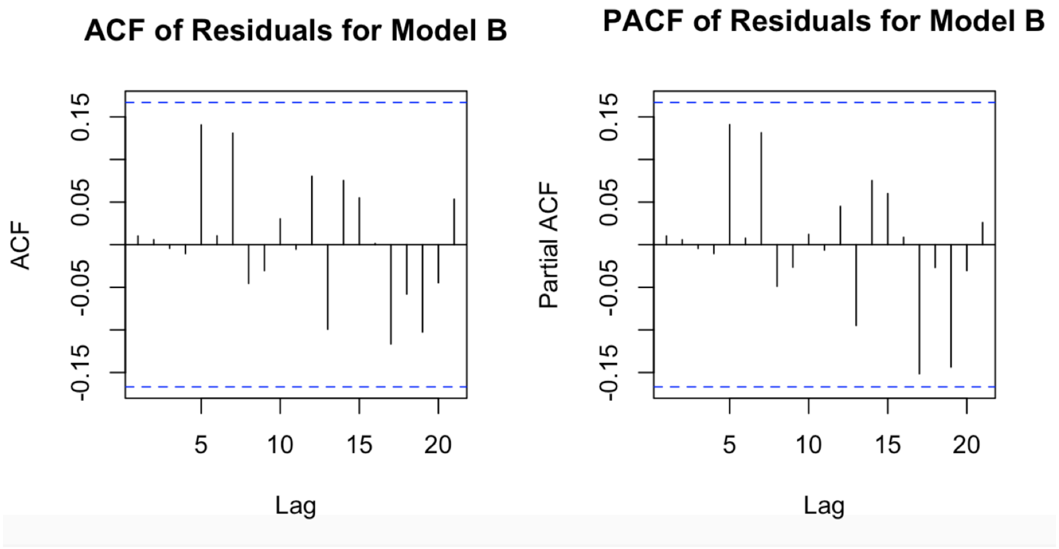
```
        Shapiro-Wilk normality test

data:  model1$residuals
W = 0.98087, p-value = 0.05

        Box-Pierce test

data:  model1$residuals
X-squared = 2.7691, df = 3, p-value = 0.4286
```



```
        Box-Ljung test

data:  model1$residuals
X-squared = 2.9142, df = 3, p-value = 0.4051
```



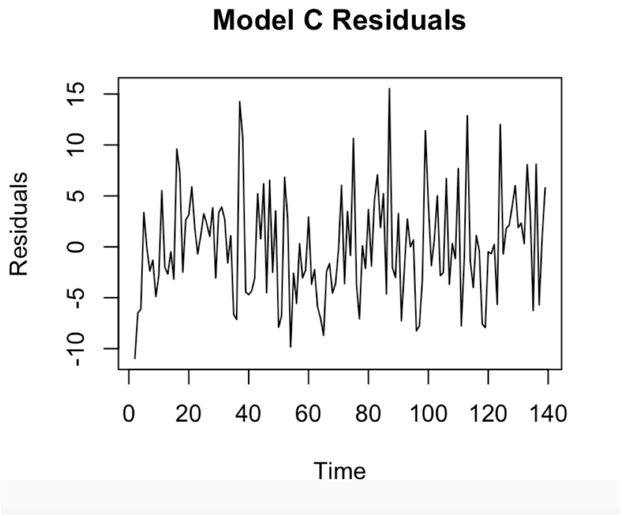
```
        Box-Ljung test

data:  (model1$residuals)^2
X-squared = 5.1596, df = 5, p-value = 0.3967
```

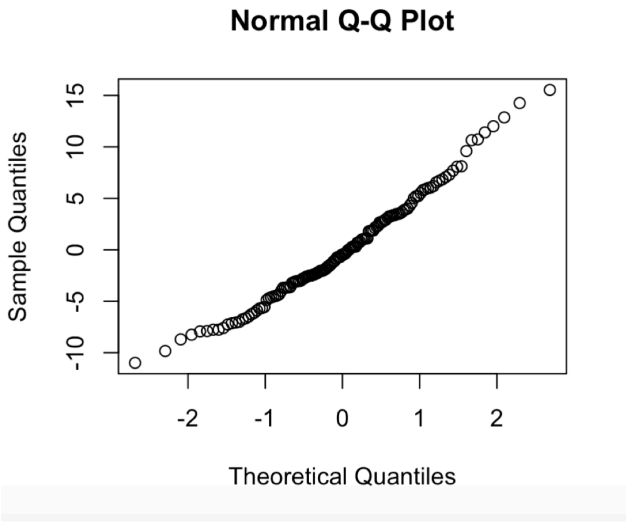


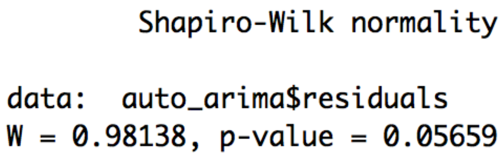
Model B passes all the tests required for normality, but again there is skepticism in the Shapiro-Wilk test since the p-value is equal to 0.05 The Q-Q plot has a linear pattern and the ACF/PACF both look like white noise. We can say that this model is fit for forecasting.

We will look at Model C.

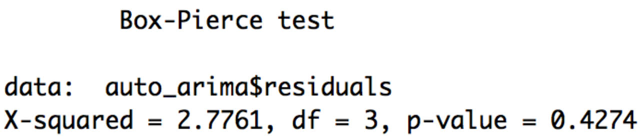
Model C



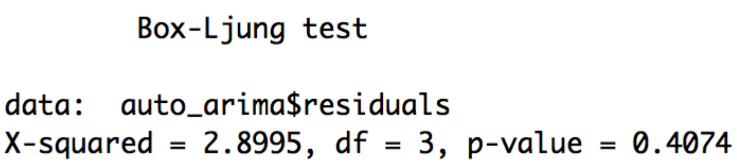




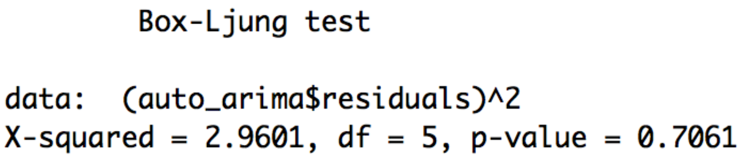
```
        Shapiro-Wilk normality test

data:  auto_arima$residuals
W = 0.98138, p-value = 0.05659
```



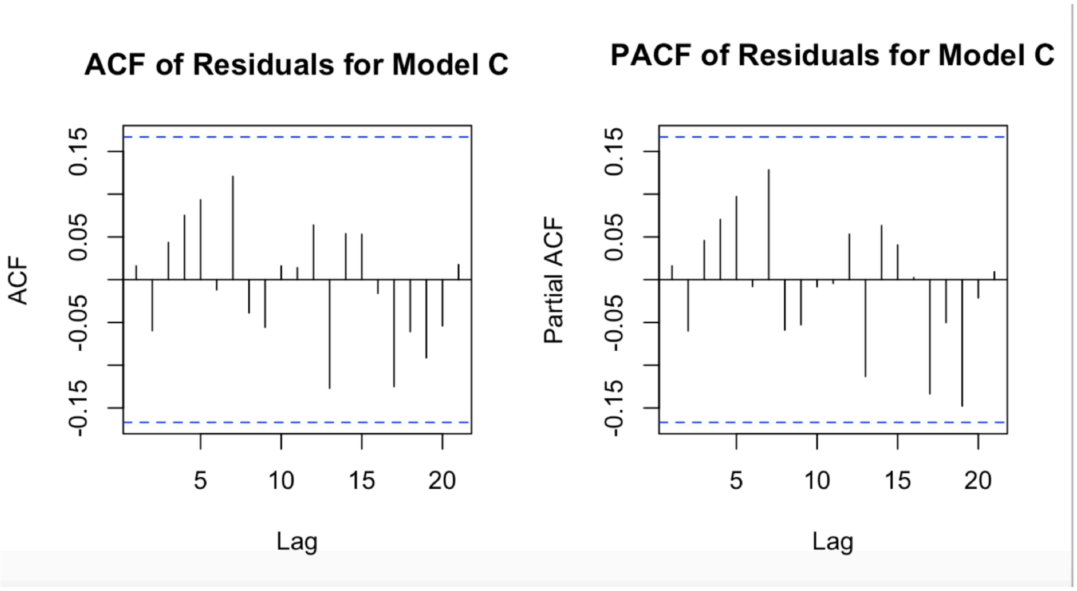
```
        Box-Pierce test

data:  auto_arima$residuals
X-squared = 2.7761, df = 3, p-value = 0.4274
```



```
        Box-Ljung test

data:  auto_arima$residuals
X-squared = 2.8995, df = 3, p-value = 0.4074
```

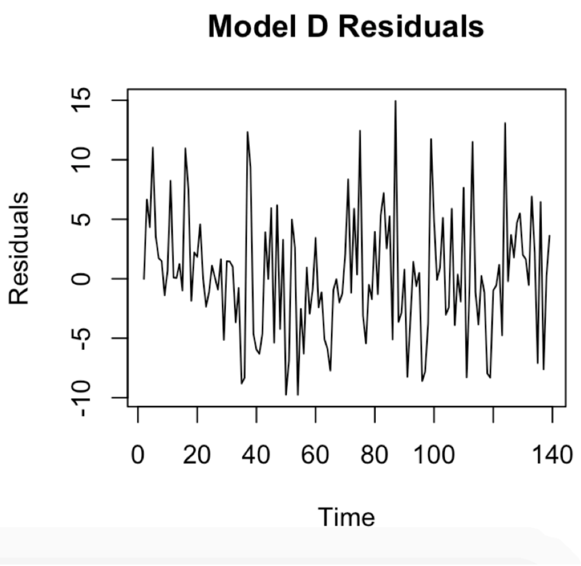


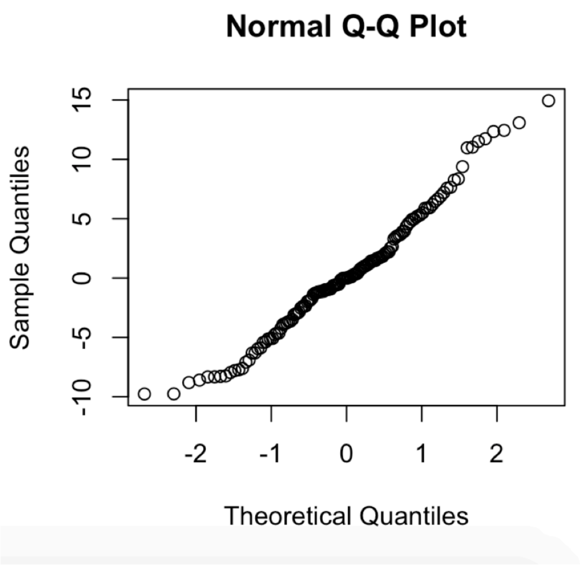
```
        Box-Ljung test

data:  (auto_arima$residuals)^2
X-squared = 2.9601, df = 5, p-value = 0.7061
```

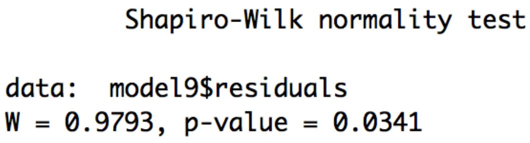


Model C also passes all diagnostic tests and has a linear Q-Q plot pattern and ACF/PACF plots that resemble white noise.

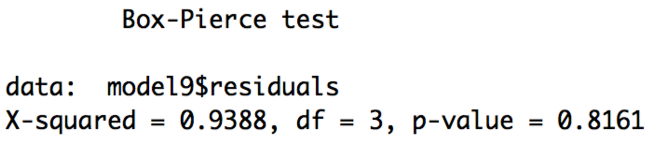
Model D



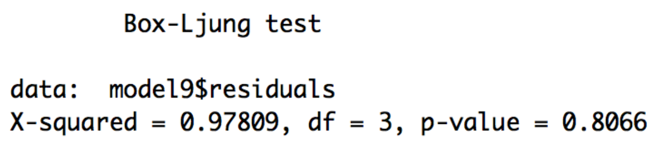




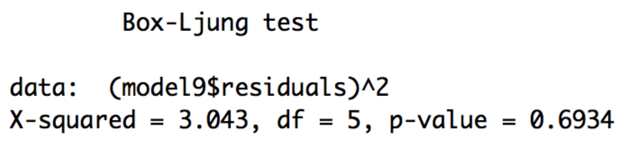
```
        Shapiro-Wilk normality test

data:  model9$residuals
W = 0.9793, p-value = 0.0341
```



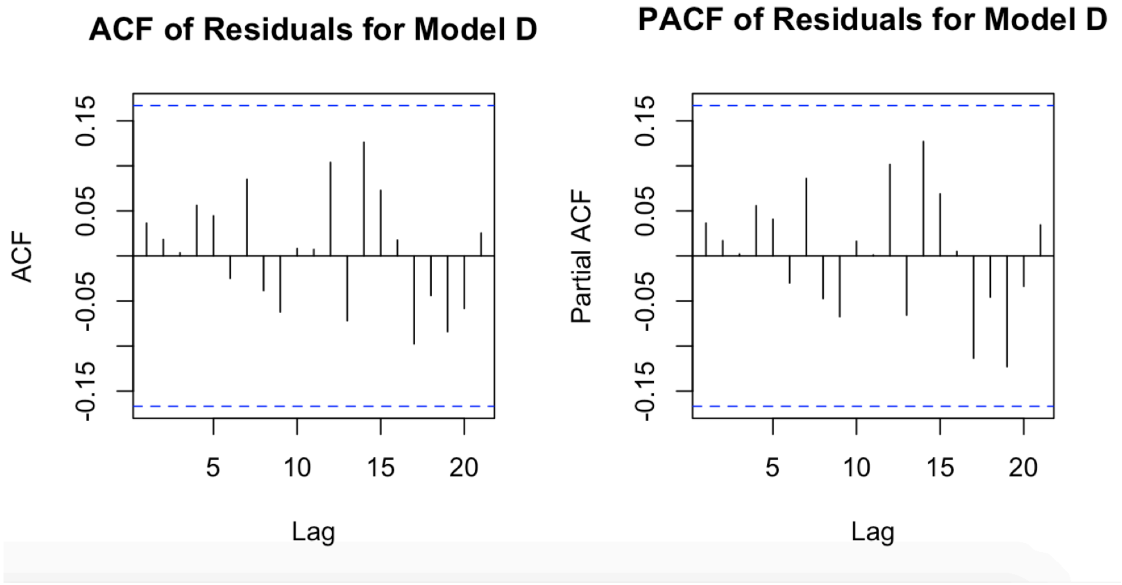
```
        Box-Pierce test

data:  model9$residuals
X-squared = 0.9388, df = 3, p-value = 0.8161
```



```
        Box-Ljung test

data:  model9$residuals
X-squared = 0.97809, df = 3, p-value = 0.8066
```



```
        Box-Ljung test

data:  (model9$residuals)^2
X-squared = 3.043, df = 5, p-value = 0.6934
```



Model D also adequately passes all tests except the Shapiro-Wilk normality test.

Based on the diagnostics of the residuals, Model C seems to be the best model for the data. However, for sake of curiosity, we are going to forecast Model D in addition to Model C.

The models used are

$$(1 + 0.0943B) \nabla Y_t + 8.588 = 8.588 + (1 - 0.1812B)Z_t$$

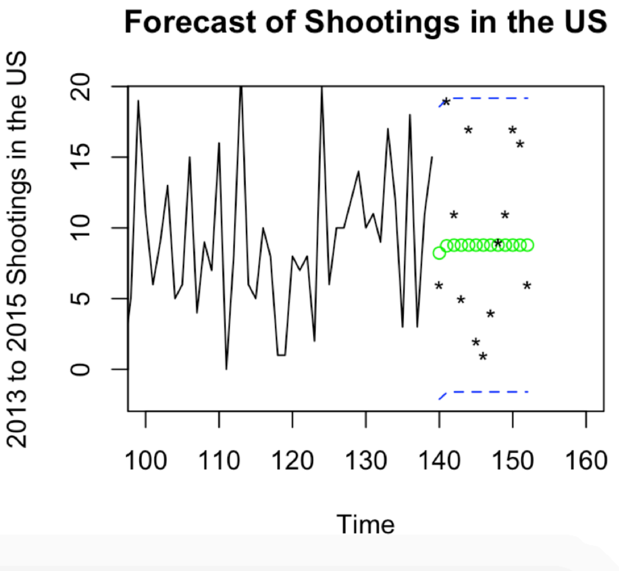
$$(1 - 0.109B - 0.9725B^2) \nabla Y_t = (1 - 0.9568B + 1.02B^2 - 1.011B^3 + 0.08B^4)Z_t$$

Forecasting

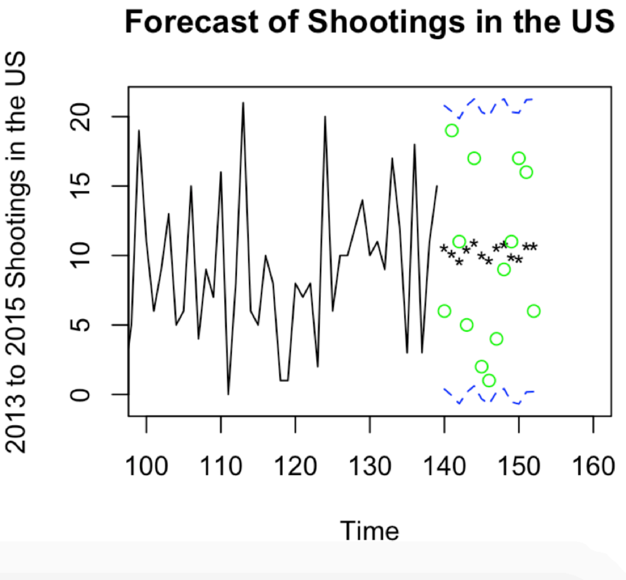
With Model C and D, we can then begin forecasting. Here, we can compare our model's forecast against the observations that were removed at the beginning of the analysis.

The blue dashed lines represent the confidence interval of our forecast and the green points represent the actual series. The asterisks are our forecast.

Forecast of Model C ARIMA(1, 0, 1)



Forecast of Model D ARIMA(2, 1, 4)



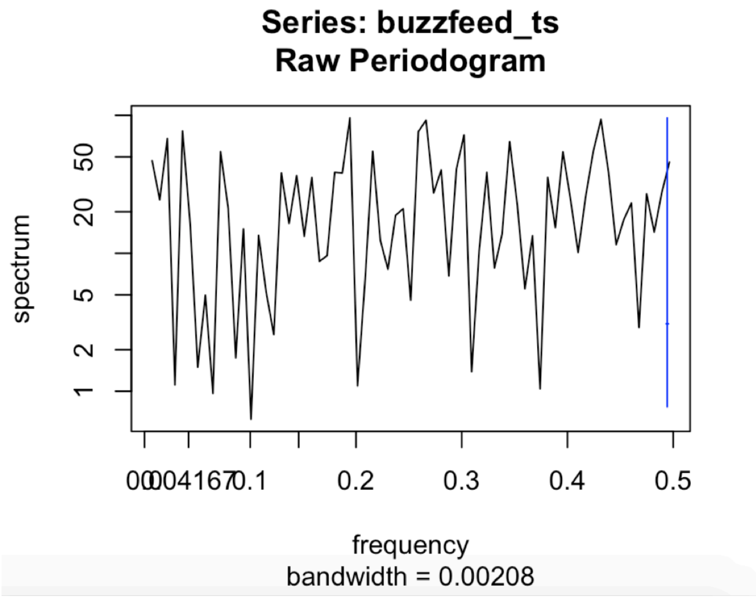
It is disappointing to see that the models have not done a sufficient job of forecasting the next thirteen observations for shootings in the U.S. Even after trying multiple forecasts of models (not listed), none have come close to the actual observed values in the data. The actual points are within the confidence intervals so the projection of the path is satisfactory, albeit there is a high standard error since the interval includes zero at a few points.

In the context of this situation, it is difficult to predict the next event of a shooting since each event is independent – there is no seasonal possibility or trend that can play into part, just the decision of the shooter.

Spectral Analysis

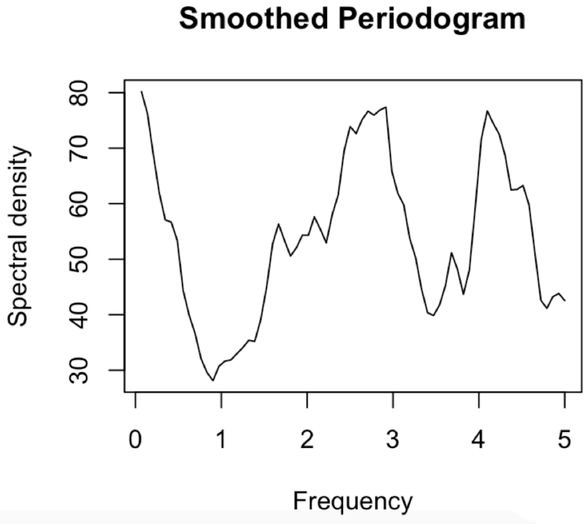
Spectral analysis is the decomposition of a time series into sine and cosine functions of different frequencies, and allows us to then determine the frequencies that are particularly strong or important. The frequency of a sine or cosine function is typically expressed in terms of the number of cycles per unit time. The period of a sine or cosine function is defined as the length of time required for one full cycle [5].

We will use a periodogram for our spectral analysis. The periodogram is ideal for identifying periodicity in data and estimating the frequency of the period.

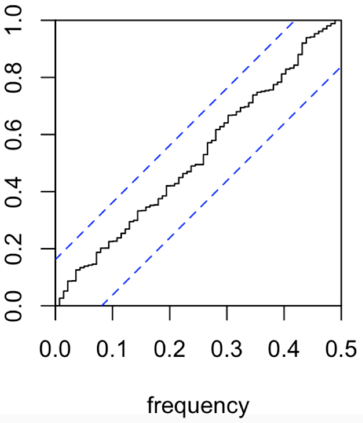


We can see that there are no dominant frequencies that can be modeled by trigonometric functions. But, in a periodogram, the precise set of frequencies is arbitrary and it does not become smoother over the length of the time series. Our next step is to smooth the periodogram (the drawback is that some of the features we are looking at will become vague).

In our smoothed periodogram, there are still high frequencies (many periods), which is expected from our non-seasonal data.



The Fisher test is used to test the data for hidden periodicities with unspecified frequency [6]. In this case, the Fisher test has a p-value of 0.879, which passes the Fisher test for periodicity. This is not surprising since the original data did not appear periodic.



The Kolmogorov-Smirnov test is applied to residuals to assess whether the residuals are Gaussian white noise. Here, the KS test for periodicity confirms the results of the Fisher test.

Conclusion

This project was very fun to do – especially starting out thinking we could predict the next shootings that were going to occur in the country. The chosen final mode that fit the data was

$$(1 - 0.109B - 0.9725B^2) \nabla Y_t = (1 - 0.9568B + 1.02B^2 - 1.011B^3 + 0.08B^4)Z_t$$

There is a strong possibility the models that I chose (ARIMA (1,0,1), ARIMA(2, 1, 4)) were not the best fit (even with most minimized AIC and BIC). The forecast did not directly match the actual test set, but it does fall within the confidence intervals and projects a path. Even so, an ARCH or GARCH model would not be adequate because each data point is not dependent on the previous points.

The reason for the incorrect prediction could be due to the unseen circumstances behind each event. The logic behind each shooting in the country has many factors behind it – dependent on the mindset of the shooter, gun control, locations of the victims, etc. With so many models available, it is difficult to find the best one, but it is interesting to analyze the data!

References

1. Simon, Mallory, and Ray Sanchez. "U.S. gun violence: The story in graphics." CNN. December 04, 2015. Accessed December 11, 2017. http://www.cnn.com/2015/12/04/us/gun-violence-graphics/index.html.
2. BuzzFeedNews. "BuzzFeedNews/2015-12-mass-shooting-intervals." GitHub. December 02, 2015. Accessed December 11, 2017. https://github.com/BuzzFeedNews/2015-12-mass-shooting-intervals.
3. "AIC vs. BIC." The Methodology Center. Accessed December 11, 2017. https://methodology.psu.edu/AIC-vs-BIC.
4. "Difference between Ljung Box and McLeod Li Test?" Time series - Difference between Ljung Box and McLeod Li Test? - Cross Validated. Accessed December 11, 2017. https://stats.stackexchange.com/questions/174934/difference-between-ljung-box-and-mcleod-li-test.
5. Wearing, Helen J. "Spectral Analysis in R." McMaster University. June 8, 2010. Accessed December 11, 2017. https://ms.mcmaster.ca/~bolker/eeid/2010/Ecology/Spectral.pdf.
6. Feldman, Raya. "Spectral Analysis." Gauchospace. November 27, 2017. Accessed December 1, 2017. https://gauchospace.ucsb.edu/courses/pluginfile.php/1571262/mod_resource/content/1/l17-slides-2017Spring-Fall.pdf.

Appendix

```
library(MASS)
library(GeneCycle)
library(TSA)
library(forecast)
library(astsa)
library(tseries)
library(lubridate)
library(dplyr)
library(ggplot2)

setwd("~/Desktop/PSTAT274Project")
#read in the data and set column names for consistency
data2013 <- read.csv("2013MASTER.csv")
colnames(data2013) <- c("Reported", "Date", "Shooter", "Killed", "Wounded", "Loc
ation", "Article1",
                        "Article2", "Article3", "Article4", "Article5", "Article
6", "Article7", "Article8",
                        "Article9")
data2014 <- read.csv("2014MASTER.csv")
colnames(data2014) <- c("Reported", "Date", "Shooter", "Killed", "Injured", "Loc
ation", "Article", "Article1",
                        "Article2", "Article3", "Article4", "Article5")
data2015 <- read.csv("CURRENT2015.csv")
data2015$X <- NULL #don't want this
colnames(data2015) <- c("Date", "Shooter", "Killed", "Injured", "Location", "Art
icle", "Article1", "Article2",
                        "Article3", "Article4")

#function to merge ALL 3 dataframes in R
MyMerge <- function(x, y){
  df <- merge(x, y, all.x= TRUE, all.y= TRUE)
  return(df)
}
dataTotal <- Reduce(MyMerge, list(data2013, data2014, data2015))

#arrange the entire dataset by date
dataTotal <- dataTotal %>% mutate(Date = as.Date(Date, "%m/%d/%Y")) %>%
  arrange(Date)

dataTotal$Date <- as.Date(dataTotal$Date, format="%d/%m/%Y")

# Aggregate over week number and number killed
dataTotal$weeknos <- (interval(min(dataTotal$Date), dataTotal$Date) %/% weeks(1)
) + 1
buzzfeed_killed <- aggregate(Killed~weeknos, FUN=sum, data=dataTotal, na.rm=TRUE
)
buzzfeed_killed <- data.frame(buzzfeed_killed)
```

```r
#Remove last 12 rows for forecasting purposes
buzzfeed_df_removed <- buzzfeed_killed[-c(140:152), ]
View(buzzfeed_df_removed)

#time series
buzzfeed_ts <- ts(buzzfeed_df_removed$Killed) #nonstationary!!
plot(buzzfeed_ts, ylab = "2013 - 2015 Shootings in the US", main = "Weekly Kille
d in 2013 - 2015 US Shootings")


#variance to reference for volatility
var(buzzfeed_ts)

#The Augmented Dickey–Fuller (ADF) t-statistic test:
#small p-values suggest the data is stationary and
#doesn’t need to be differenced stationarity.
adf.test(buzzfeed_ts, alternative = "stationary")

#The Ljung-Box test examines whether there is significant evidence for non-zero
#correlations at lags 1-20. Small p-values (i.e., less than 0.05)
#suggest that the series is stationary.
Box.test(buzzfeed_ts, type = "Ljung-Box")

########seasonal decomposition to find the additive trend, seasonal, irregular c
omponents
decomp <- stl(buzzfeed_ts, s.window = "period")
plot(decomp)

#Take difference at lag 1 twice. Variance of diff2 is greater.
#Overdifferencing according to the variances
buzzfeed_diff1 <- diff(buzzfeed_ts, lag = 1)
plot(buzzfeed_diff1, ylab = "US Shootings in 2013 - 2015", main = "First Differe
nce on Lag 1")

#second difference at lag 1
buzzfeed_diff2 <- diff(buzzfeed_diff1, lag = 1)
plot(buzzfeed_diff2, ylab = "US Shootings in 2013 - 2015", main = "Second Differ
ence on Lag 1")

#compare the variances
var(buzzfeed_diff1)
var(buzzfeed_diff2)

#Run stationarity tests
Box.test(buzzfeed_diff1, type = "Ljung-Box")
adf.test(buzzfeed_diff1, alternative = "stationary")

#decomposition
diff_decomp <- stl(buzzfeed_diff1, s.window = "periodic")
plot(diff_decomp)
```

```r
#ACF and PACF of differenced and transformed data
par(mfrow=c(1,2))
acf(buzzfeed_diff1)
acf(buzzfeed_diff1, type = "partial")

#automated arima forecast
auto_arima <- auto.arima(buzzfeed_diff1, trace = TRUE)
summary(auto_arima)
BIC(auto_arima)

#try out our own models
model1 <- arima(buzzfeed_diff1, order = c(3,1,2))
summary(model1)
BIC(model1)

model2 <- arima(buzzfeed_diff1, order = c(1,0,14))
summary(model2)
BIC(model2)

model3 <- arima(buzzfeed_diff1, order = c(1,0,2))
summary(model3)
BIC(model3)

model4 <- arima(buzzfeed_diff1, order = c(2,0,14))
summary(model4)
BIC(model4)

model5 <- arima(buzzfeed_diff1, order = c(4,1,2))
summary(model5)
BIC(model5)

model6 <- arima(buzzfeed_diff1, order = c(4,1,14))
summary(model6)
BIC(model6)

model7 <- arima(buzzfeed_diff1, order = c(4,0,1))
summary(model7)
BIC(model7)

model8 <- arima(buzzfeed_diff1, order = c(1,1,1))
summary(model8)
BIC(model8)

model9 <- arima(buzzfeed_diff1, order = c(2,1,4))
summary(model9)
BIC(model9)


#we will use model 3, model 1, auto.arima, model 9
```

```
#MODEL A DIAGNOSTICS
plot(model3$residuals, main="Model A Residuals", ylab = "Residuals")
shapiro.test(model3$residuals)

qqnorm(model3$residuals)

paste("Box-Pierce")
Box.test(model3$residuals, lag = 5,  type = "Box-Pierce", fitdf=2)
paste("Ljung-Box")
Box.test(model3$residuals, lag = 5, type = "Ljung-Box", fitdf=2)
paste("McLeod-Li")
Box.test((model3$residuals)^2, lag=5, type="Ljung-Box")

par(mfrow=c(1,1))
acf(model7$residuals, na.action=na.pass, main = "ACF of Residuals for Model A")
pacf(model7$residuals, na.action = na.pass, main = "PACF of Residuals for Model
A")

#MODEL B DIAGNOSTICS
plot(model1$residuals, main="Model B Residuals", ylab = "Residuals")
shapiro.test(model1$residuals)
qqnorm(model1$residuals)

acf(model1$residuals, na.action=na.pass, main = "ACF of Residuals for Model B")
pacf(model1$residuals, na.action=na.pass, main = "PACF of Residuals for Model B"
)

paste("Box-Pierce")
Box.test(model1$residuals, lag = 5, type = "Box-Pierce", fitdf=2)
paste("Ljung-Box")
Box.test(model1$residuals, lag = 5, type = "Ljung-Box", fitdf=2)
paste("McLeod-Li")
Box.test((model1$residuals)^2, lag=5, type="Ljung-Box")

#MODEL C DIAGNOSTICS
plot(auto_arima$residuals, main="Model C Residuals", ylab = "Residuals")
shapiro.test(auto_arima$residuals)
qqnorm(auto_arima$residuals)
acf(auto_arima$residuals, na.action=na.pass, main = "ACF of Residuals for Model
C")
pacf(auto_arima$residuals, na.action=na.pass, main = "PACF of Residuals for Mode
l C")

paste("Box-Pierce")
Box.test(auto_arima$residuals, lag = 5, type = "Box-Pierce", fitdf=2)
paste("Ljung-Box")
Box.test(auto_arima$residuals, lag = 5, type = "Ljung-Box", fitdf=2)
paste("McLeod-Li")
Box.test((auto_arima$residuals)^2, lag=5, type="Ljung-Box")
```

```
#MODEL D DIAGNOSTICS
plot(model9$residuals, main="Model D Residuals", ylab = "Residuals")
shapiro.test(model9$residuals)
qqnorm(model9$residuals)
acf(model9$residuals, na.action=na.pass, main = "ACF of Residuals for Model D")
pacf(model9$residuals, na.action=na.pass, main = "PACF of Residuals for Model D"
)

paste("Box-Pierce")
Box.test(model9$residuals, lag = 5, type = "Box-Pierce", fitdf=2)
paste("Ljung-Box")
Box.test(model9$residuals, lag = 5, type = "Ljung-Box", fitdf=2)
paste("McLeod-Li")
Box.test((model9$residuals)^2, lag=5, type="Ljung-Box")



#Begin forecasting
#match forecast to our removed data
#the last twelve points of our data to compare against the forecast
buzzfeed_last_12 <- buzzfeed_killed[-c(1:139),]
View(buzzfeed_last_12)
orig_removed <- buzzfeed_last_12$Killed
#fitted model C
fit <- arima(buzzfeed_ts, order= c(1,0,1), method = "ML", xreg = 1:length(buzzfe
ed_ts))
buzzfeed_predict <- predict(fit, n.ahead = 13, newxreg = (length(buzzfeed_ts)+1)
: length(buzzfeed_ts)+13)

#confidence interval bounds
upper <- buzzfeed_predict$pred + 2*buzzfeed_predict$se
lower <- buzzfeed_predict$pred - 2*buzzfeed_predict$se

#plot forecast
plot(buzzfeed_ts, xlim = c(100, 160), ylim = c(min(lower), max(upper)), ylab = "
2013 to 2015 Shootings in the US",
     main = "Forecast of Shootings in the US")
lines(upper, col = "blue", lty = "dashed")
lines(lower, col = "blue", lty = "dashed")
points(140:152, buzzfeed_predict$pred, col = "green")
points(140:152,orig_removed, pch = "*")

#fitted model D
fit2 <- arima(buzzfeed_ts, order= c(2,1,4), method = "ML", xreg = 1:length(buzzf
eed_ts))
buzzfeed_predict2 <- predict(fit2, n.ahead = 13, newxreg = (length(buzzfeed_ts)+
1): length(buzzfeed_ts)+13)
```

```
#confidence interval bounds
upper2 <- buzzfeed_predict2$pred + 2*buzzfeed_predict2$se
lower2 <- buzzfeed_predict2$pred - 2*buzzfeed_predict2$se

#plot forecast
plot(buzzfeed_ts, xlim = c(100, 160), ylim = c(min(lower2), max(upper2)), ylab =
 "2013 to 2015 Shootings in the US",
     main = "Forecast of Shootings in the US")
lines(upper2, col = "blue", lty = "dashed")
lines(lower2, col = "blue", lty = "dashed")
points(140:152, buzzfeed_predict2$pred, pch = "*")
points(140:152,orig_removed, col = "green")

#find coefficients
summary(fit)
summary(fit2)

#spectral analysis
periodogram(buzzfeed_ts)

#smoothed periodogram
del<-0.1 # sampling interval
x.spec <- spectrum(buzzfeed_ts,log="no",span=10,plot=FALSE)
spx <- x.spec$freq/del #cycles per unit of time
spy <- 2*x.spec$spec # multiply spectral density by 2 to match variance of time
series
plot(spy~spx,xlab="Frequency",ylab="Spectral density",type="l", main = "Smoothed
 Periodogram")

#fisher test
fisher.g.test(residuals(fit))

#ks test
cpgram(residuals(fit), main = "")
```