MODELING TRAFFIC FATALITIES VIA TIME SERIES ANALYSIS

Elisa Cui

Rho

PSTAT 174

Professor Sudeep Bapat

June 6, 2018

Table of Contents

Abstract

We used a monthly traffic fatality data set of fifteen years (1960-1974) to perform a time series analysis that can predict the traffic fatality rates in Ontario, Canada. In our analysis, we used various methods and techniques like diagnostic checking, forecasting, data differencing, model selection, and seasonality differencing to test multiple SARIMA models. We used Box Cox Transformation, autocorrelation function (ACF), partial autocorrelation function (PACF), and other functions to manipulate and interpret the original data. In the end, our best model is SARIMA (2,0,1) x $(2,1,2)_{12}$ that can be used to make monthly forecasts. We split the data into a training set and a test set to check how well the predicted values would compare with the true values.

Introduction

Currently, there is an estimated 1.25 million deaths caused by traffic accidents worldwide. That's practically one death every 25 seconds. It is even considered one of the top 10 global causes of death by the World Health Organization. These incidents vary from drunk driving, speeding, lack of vehicle protection, crashes, and countless of other possibilities. These accidents become a tragedy for the victims involved and a problem for the city itself financially. Even though traffic fatalities affect everyone around the world, we are focusing on the data set of 180 observations provided by DataMarket.com, of monthly traffic accident fatalities in Ontario, Canada from 1960-1974. The data set ranges from 55 to 256 deaths each year in this fifteen year period. The data set and forecast can be used to understand the changes in accidents, why it is occurring, and ways to prevent it to increase in the future.

The goal of this project is to forecast traffic accident fatalities in Ontario, Canada in the future by using statistical analysis for model selection. By using the statistical software R, we first applied different transformations and seasonality differencing to reduce variability and any trends that may have skewed the original dataset. Through exploratory analysis, we continued with the log transformation of our data set and de-seasonalized it at lag 12 and lag 1. After seeing the results, we went forward with differencing at lag 12 only because, after differencing first at lag 12 then at lag 1, the data showed signs of over differencing. By referencing the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots, we narrowed down our selected models. Lastly, through diagnostic checking and forecasting, we concluded that the best model for our project is the SARIMA (2,0,1) x $(2,1,2)_{12}$. We can interpret this model in the form

$$Y_t\ (1-\phi_1 B)(1-\ \phi_2 B)(1-\ \Phi_1 B^{12})(1-\ \Phi_2 B^{12})(1-B^{12}) = (1-\theta_1 B)(1-\Theta_1 B^{12})(1-\Theta_2 B^{12})\ Z_t$$

so our final model is

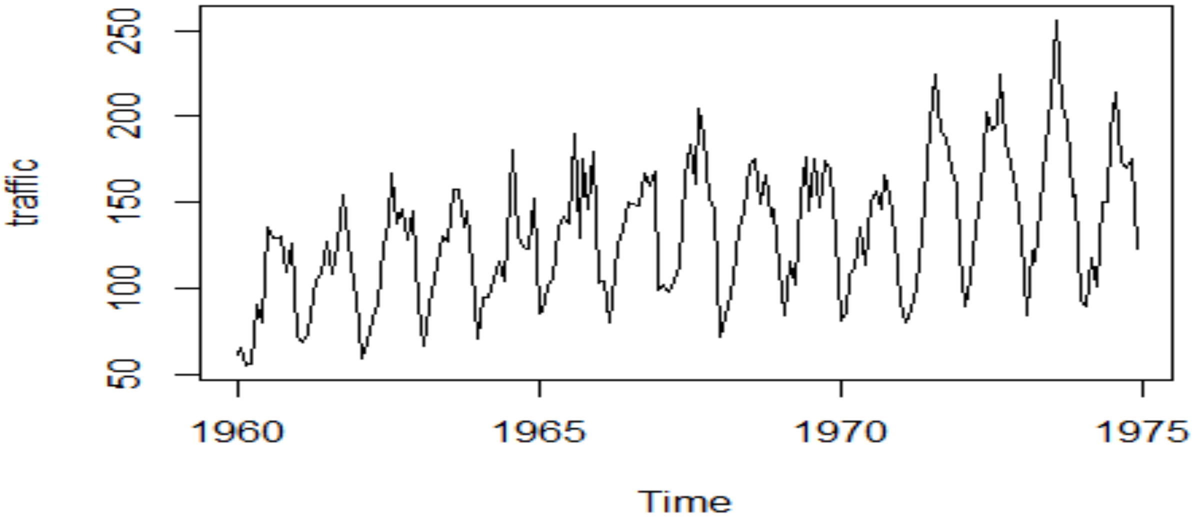
Y_t (1-0.89B)(1-0.07B)(1+0.29B^{12})(1- 0.17B^{12})(1-B^{12}) = (1+ 0.74B)(1+ 0.46B^{12})(1+ 0.43B^{12})Z_t

where $Y_t = \log(X_t) - \log(X_{t-12})$ and Z_t is white noise. This model is stationary, as proven by the Augmented Dickey-Fuller Test since small p-values suggest stationarity and ours was less than 0.01. It is also invertible as the roots of θ_1, Θ_1, and Θ_2 are outside of the unit circle.

We then separated our data into a training set and test set to compare our predicated values to the original data. The original data's values are within the confidence intervals and our predicted values are similar and match the seasonality well enough so we can conclude that the model is sufficient in predicting future values of this data set.

Raw Data

The monthly traffic fatalities from Ontario, Canada has 180 observations from 1960 until 1975. Looking at the raw data, there seems to be a seasonality trend as well as an increase in variance after 1971. The lows of the trend tends to happen in February while the highs appear in August/September. We first need to transform the data to decrease the variance and then difference it to remove seasonality and trend, it needed.

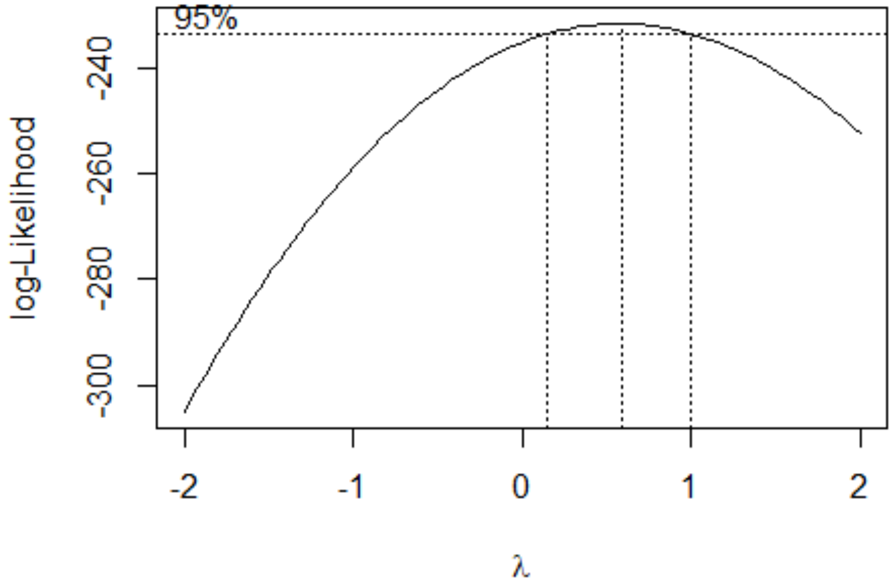


Transformation of Data

Based on the averages of the data, there are 132 traffic fatalities per month in Ontario. The mean rate is increasing with time which means that traffic fatalities are definitely a growing issue in Ontario. From our training set 1960 to mid-1973, the average is 128.87 and from our test set, which is the last 18 months of the data, the average is 162.5. The variance of the training set is 13.56 and the variance of the test set is 21.10. There is a clear increase in our average and variances from our training set to the test set which means the data needs to be transformed in order to stabilize the variance.

Box-Cox Transformation

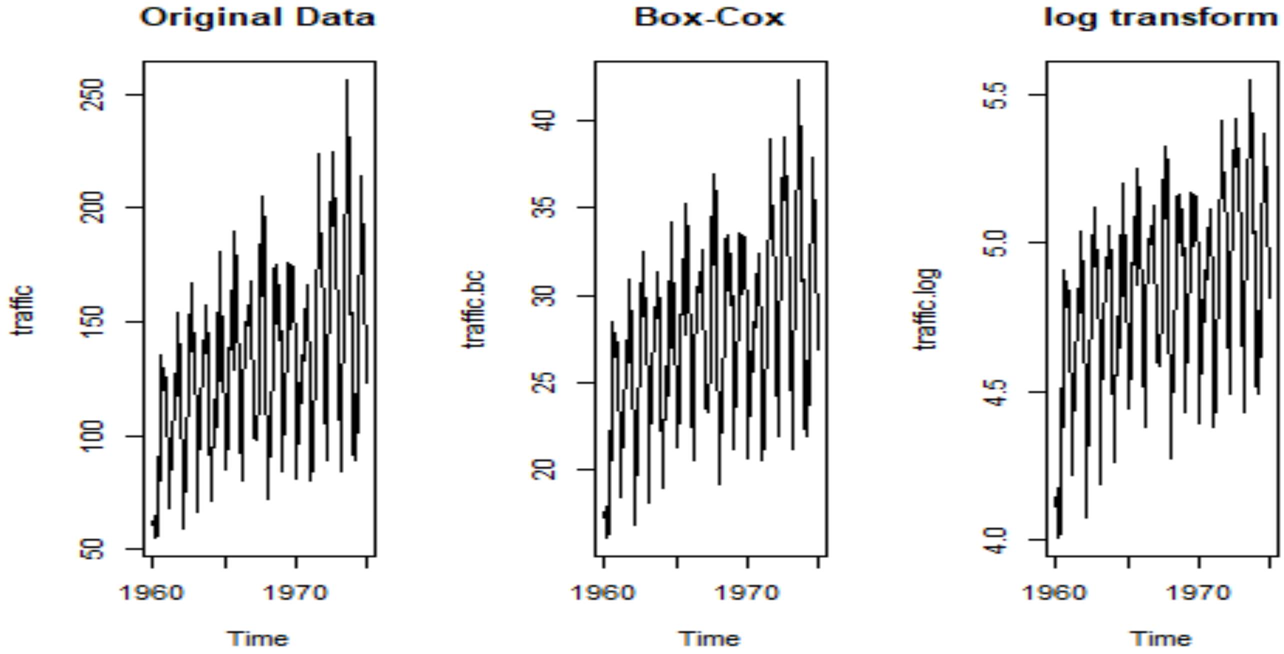
The increasing variance from 13.56 to 21.10 from our two datasets clearly shows that a transformation needs to be arranged on the dataset in order to lower the variance. We performed a Box-Cox transformation with λ = .586 as our value to maximize the log-likelihood. Our variance of the entire data resulted in 27.02 which means that this transformation did not prove to be helpful since our variance is still seen to be unstable, in which we decided to use another transformation to further analyze our data.



Logarithmic Transformation

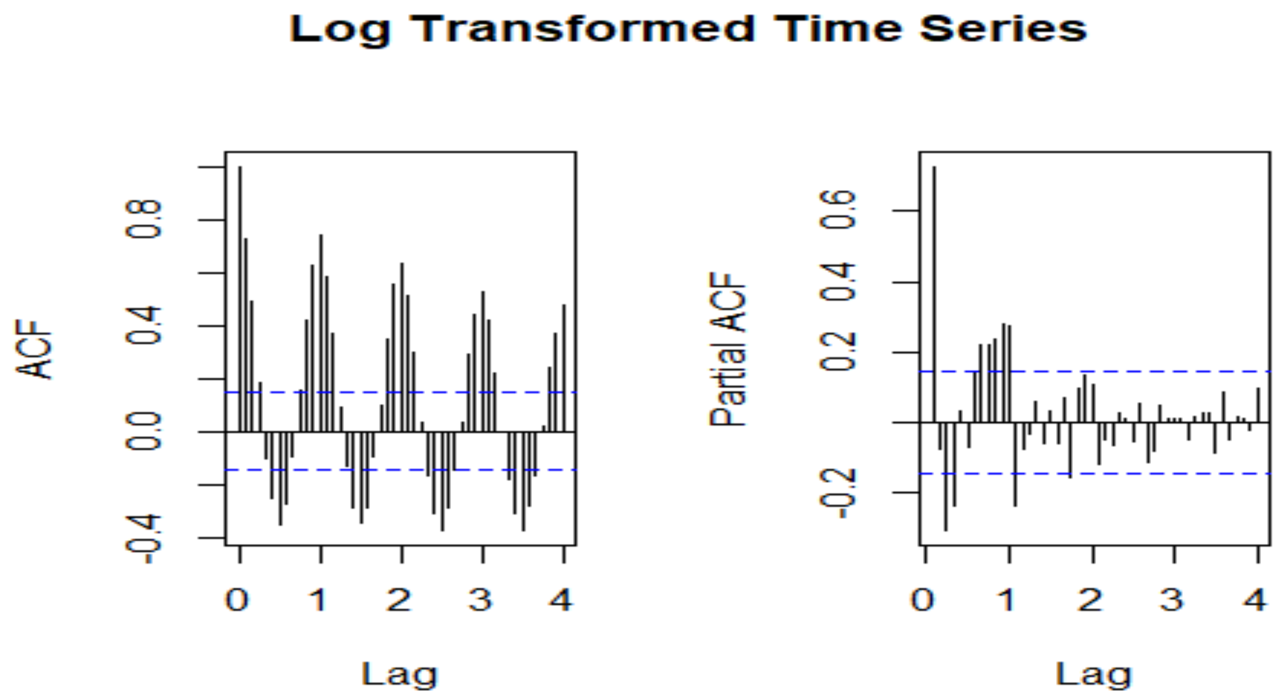
With the logarithmic transformation, our variance decreased to 0.096 which was much smaller than the variance Box-Cox transformation of the data. We also tried a square root transformation but again, the variance increased which helped us conclude that the logarithmic transformation was the best. We also used the same transformation on our training set of the data from 1965 to mid-1974. This produced a very similar variance of 0.092 as well as alike ACF/PACF plots, as shown below. In the end, our model was

Y_t (1-0.89B)(1-0.07B)(1+0.29B^{12})(1- 0.17B^{12})(1-B^{12}) = (1+ 0.74B)(1+ 0.46B^{12})(1+ 0.43B^{12})Z_t.



ACF/PACF of Log Transformed Data

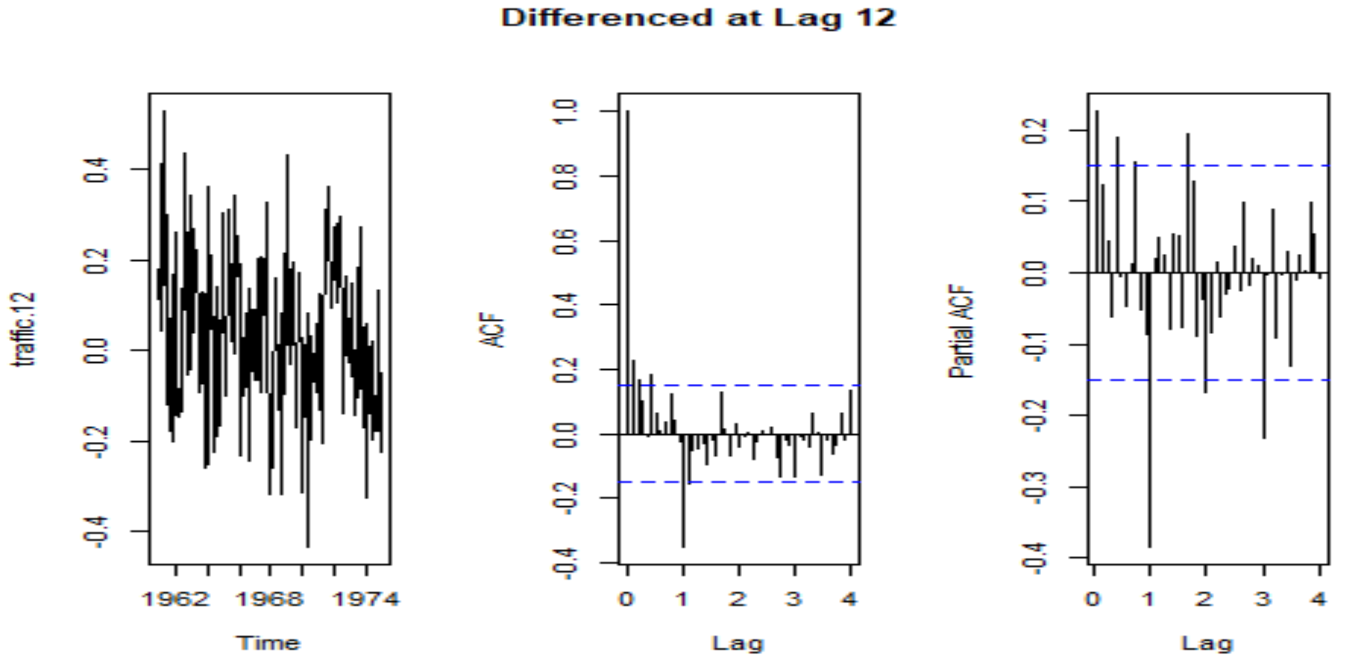
Since we are set on using the logarithmic transformation of the dataset, we now want to visually analyze the autocorrelation and partial autocorrelation functions in order to further examine the seasonal trend within the data. The ACF plot has an annual cycle with large correlations at large lags which means we need to difference the data for a more accurate analysis.



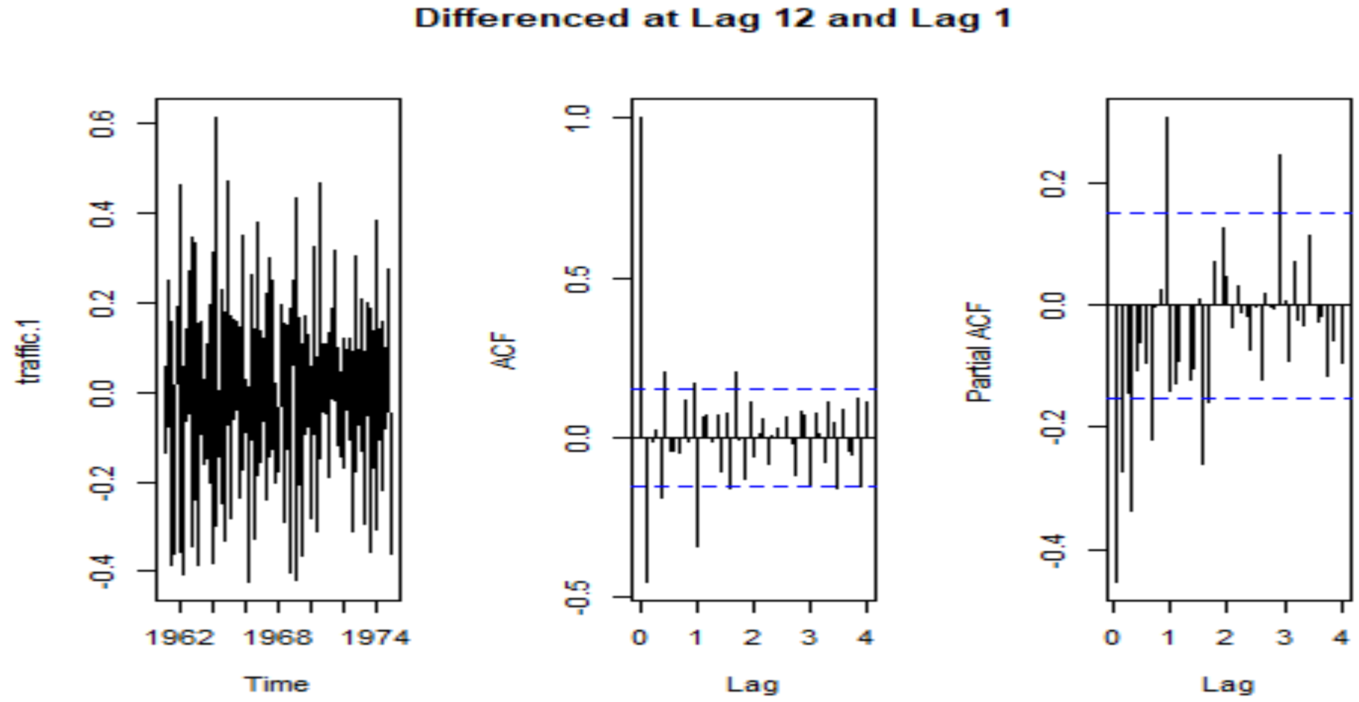
Data Differencing

We have chosen to difference the data at lag 12 in order to show that the data will not have a seasonal trend as well as giving us a more stable variance. After differencing, the data will appear to be more stationary with the removal of the seasonality because of the constant mean and reduction of large correlations at large lags that we observed from the ACF prior to differencing at lag 12.

Now we want to difference at lag 1 to see if it is necessary or if it will result in over-differencing. Differencing at lag 1 made the data appear more noisy and gives our data a higher variance which means that differencing at lag 1 did indeed result in over-differencing which means we will keep the data that is differenced at lag 12. From visual analysis, we can see that the ACF/PACF show significant correlations at lags that are all multiples of 12 which means that our data could potentially be modeled by a seasonal autoregressive integrated moving average model but due to the ACF showing less correlation after lag 12, we cannot state a clear model for the data based only on the ACF/PACF figures.



Differenced at Lag 12



Differenced at Lag 12 and Lag 1

Model Selection

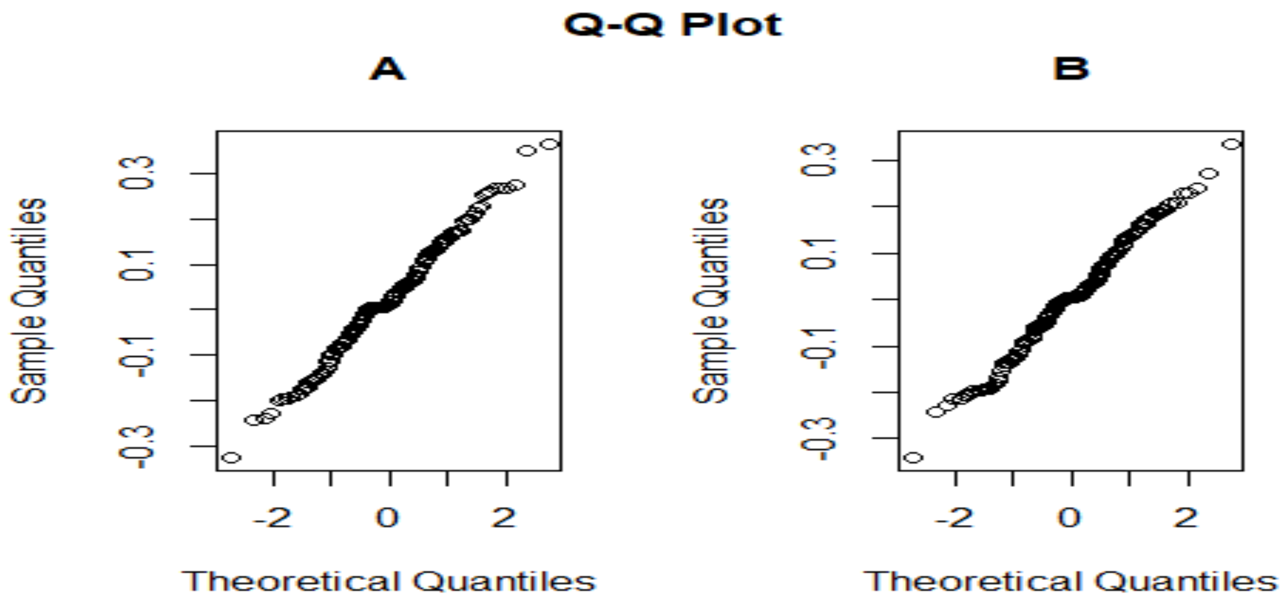
Now that our training dataset has been differenced at lag 12 and we have visually analyzed the ACF/PACFs of the data, we will continue with model selection via Akaike's Second-

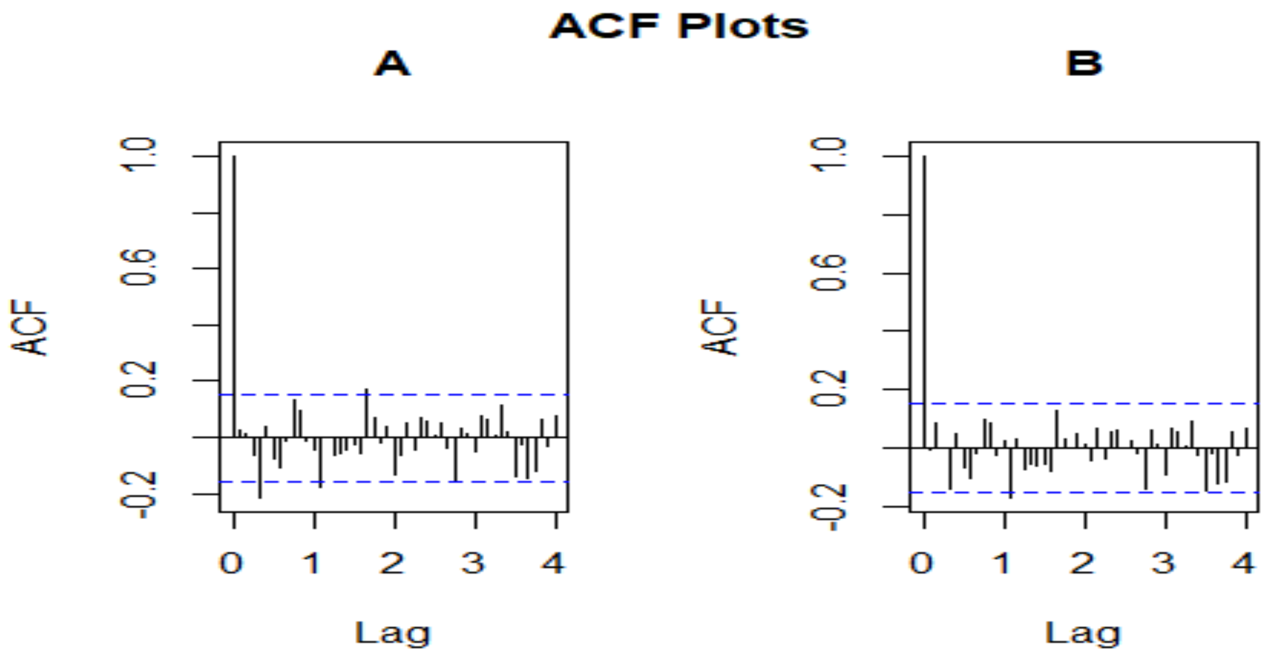
Order Corrected Information Criterion. Using the function *auto.arima* in the *forecast* library in R Studio, we were able to choose 6 models with the lowest AICc's that the computer could estimate without error. From this, we want to model our training set by a seasonal autoregressive integrated moving average model (SARIMA(p, d, q) x (P, D, Q)$_{12}$). Our data is only differenced at lag 12 to create a more stationary dataset, which means d = 0 and D = 1. Based on the results, we chose the 6 models listed below:

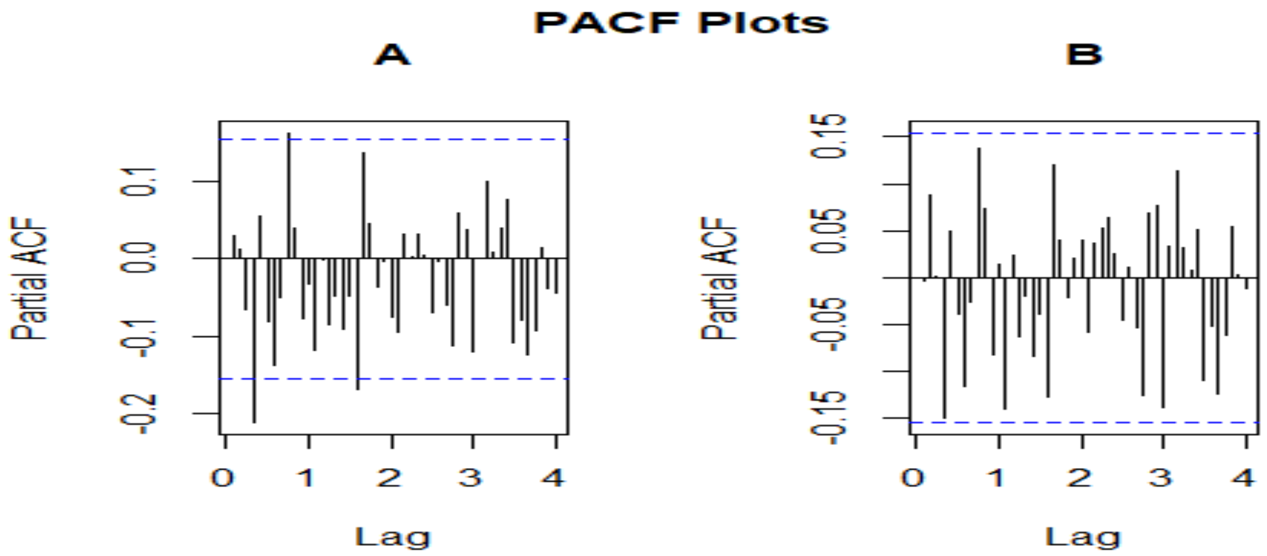
SARIMA (2,0,2)x(2,1,1)$_{12}$	with AICc -150.6762
SARIMA (1,0,1)x(2,1,2)$_{12}$	with AICc -152.7634
SARIMA (2,0,1)x(2,1,2)$_{12}$	with AICc -142.7743
SARIMA (1,0,1)x(1,1,2)$_{12}$	with AICc -154.045
SARIMA (1,0,1)x(2,1,1)$_{12}$	with AICc -154.3796
SARIMA (1,0,2)x(2,1,1)$_{12}$.	with AICc -152.3415

Diagnostic Checking

The function used to choose the best six models only estimates the AICc which means we needed to fit the model and re-check the values. Since AICc is insufficient to decide the optimal model, we began diagnostic checks on the models we found with the two lowest AICc values (SARIMA (2,0,1)x(2,1,2)$_{12}$ which we will refer to in this section as A, and SARIMA (1,0,2)x(2,1,1)$_{12}$ which we will refer to in this section as B). We began by checking the Q-Q plot where we found the residuals of both plots to be similar with a mild perturbation in the second quantile and small deviations at the ends of the first and fourth quantile but mostly lined up with the ideal 45° Q-Q plot. Next, we plotted the ACF & PACF of the models. Model A had an ACF plot with noise outside the confidence interval at lags 4,13 and 20 and a PACF plot with noise outside the confidence interval at lags 4, 9, and 19. This shows that A may not have constant error variance. Model B had an ACF plot mostly in the confidence interview with the only notable exception at lag 13 and a PACF plot entirely in the confidence interval. Thus, B is found to have constant error variance. Both models were found to have Gaussian histograms with only minor deviations. We then tested our models with the Shapiro-Wilk normality test which checks for normality of the residuals. Model A had a p-value of 0.6649 and Model B had a p-value of 0.4506, both of which are above 0.05 so the residuals of both models are approximately IID Gaussian. Next, we used the Box-Pierce test and the Box-Ljung test to check that the residuals are independent from each other. Model A had a Box-Pierce p-value of 0.09683 and a Box-Ljung p-value of 0.07582. Model B had a Box-Pierce p-value of 0.4305 and a Box-Ljung p-value of 0.3863. Since all of these values are above 0.05, the residuals of both models are uncorrelated. After analyzing the fit of the models, we decided on Model A, SARIMA (2,0,1)x(2,1,2)$_{12}$, as it performed well on all of the diagnostic checks and it is also stationary and invertible.


Q-Q Plot
A
B
Sample Quantiles
-0.3
-0.1
0.1
0.3
-2
0
2
Theoretical Quantiles

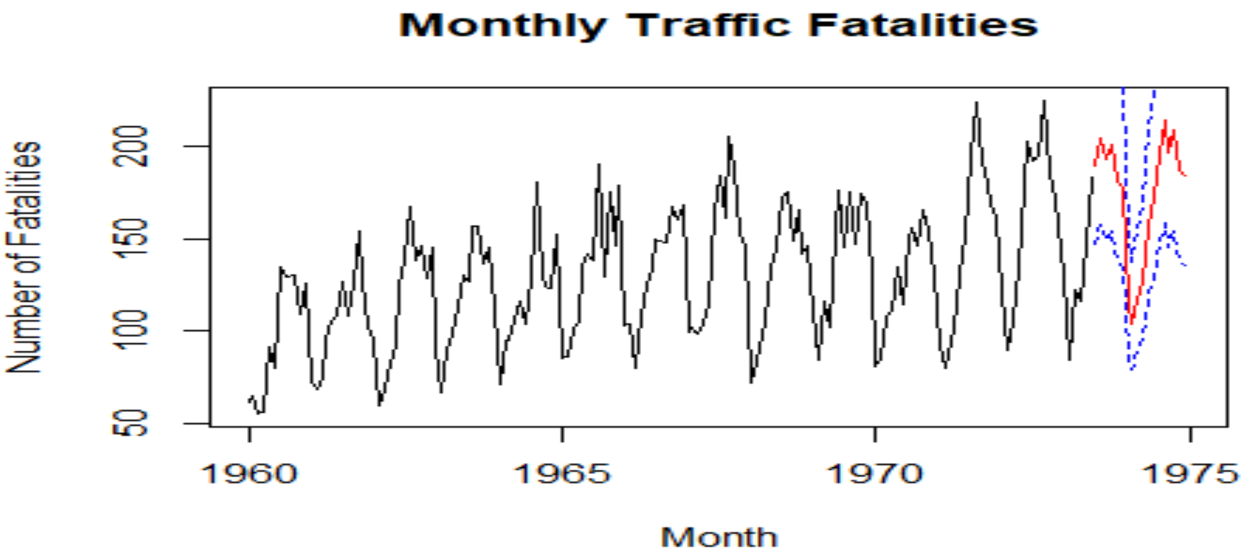

ACF Plots
A
B
ACF
-0.2
0.2
0.6
1.0
0
1
2
3
4
Lag

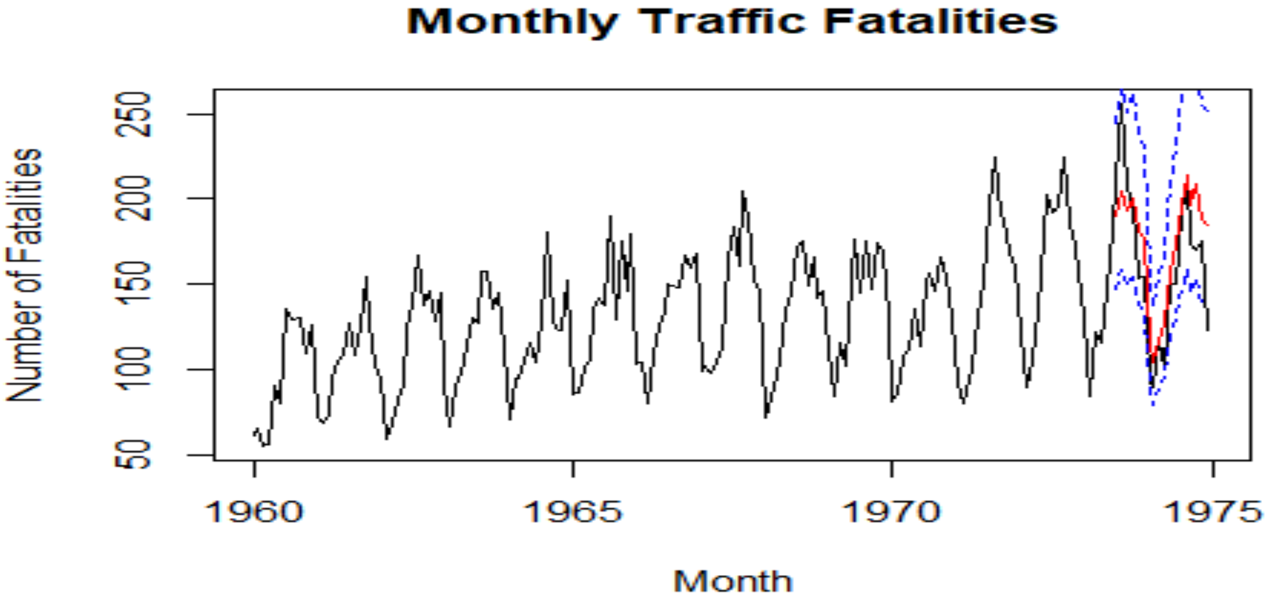


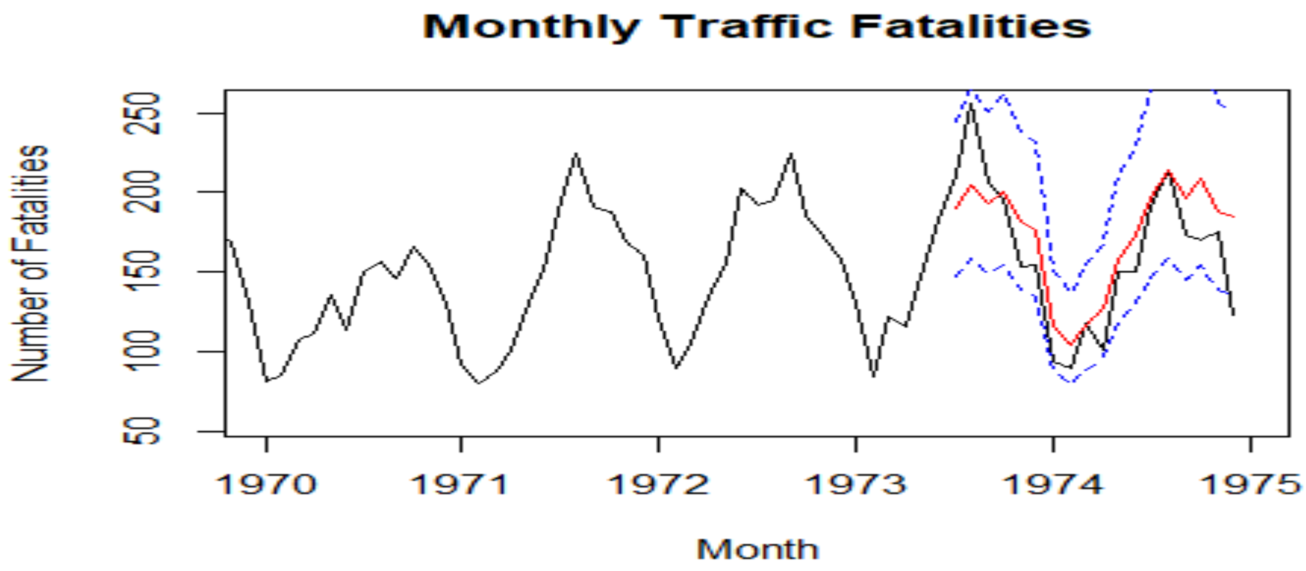
Forecasting

Since we split up our dataset into the training set being 1960 to mid-1973 and the test set being the last 18 months of our observations, we now want to forecast those 18 months ahead from the training set and see how that compares to the actual data from the test set. Based on our analysis, we want to forecast in order to conclude that the model we have chosen is correct at predicting the future values of the data. After model selection and diagnostic checking, we have chosen our model to be SARIMA (1,0,2)x(2,1,1)$_{12}$ and we will predict the next 18 months by using the *predict* function to give us the values. We also created functions to give us the 95% confidence interval that the predicted values should lie within. We then plot the training set values alongside the predicted values in order to show if our prediction is correct or insufficient. Forecasts were initially made on the log transformed data and then predicted values and confidence intervals were transformed back to the original data.

The values from the test set do appear to closely follow the predicted values and the actual values do fall within the 95% confidence interval which results in our model of SARIMA (1,0,2)x(2,1,1)$_{12}$ being able to successfully forecast the last 18 months of the original data well enough.


Monthly Traffic Fatalities
Number of Fatalities
50
100
150
200
1960
1965
1970
1975
Month


Monthly Traffic Fatalities
Number of Fatalities
50
100
150
200
250
1960
1965
1970
1975
Month



Conclusion

Our final model of the Monthly Traffic Fatalities data set from Ontario, Canada was found to predict the set accurately. The final model was

Y_t (1-B)(1- 0.046B^{12})(1+.0598B^{12})(1+0.9806B^{12}) = (1+0.8173B)(1+0.0274B^{12}) Z_t,

where Y_t = log(X_t) – log(X_{t-12}) and Z_t is white noise. This model captures the relationship between the autoregressive component and the moving average component. The AR element depends on the rates of traffic fatalities in the last two months and the MA element shows the seasonality of the entire data set. We undertook this in the hopes of predicting future traffic fatalities since they seem to be growing in number due to an increased population and thus more vehicles on the road. Our model accurately forecasted the last 18 months of the model so we can conclude that, if there was similar data that was much more recent, it would be able to predict the future values of monthly traffic fatalities in Ontario.

Lastly, we would like to thank Professor Bapat for allowing us this opportunity to gain knowledge in how to adapt and transform Time Series Data to forecast as well as analyze it. Zhipu Zhou also offered his expertise in fitting the correct model and how it uses certain data sets to automatically difference it at the best lag possible. We thank you for all the help and this opportunity.

References

Data: https://datamarket.com/data/set/22ty/monthly-traffic-fatalities-in-ontario-1960-1974#!ds=22ty&display=line

Software: R Studio

Software Packages: MASS, forecast, qpcR

Text: *Introduction to Time Series Analysis,* by P.S.P. Cowpertwait, A.V. Metcalfe

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
setwd("C:/Users/Chelsea/Desktop/Chelsea School/Pstat 174") #sets my working directory
traffic.csv = read.table('data.txt', header = FALSE) #reads the txt file for the time series
#head(traffic.csv)

traffic = ts(traffic.csv[,2], start = c(1960,1), frequency = 12) #converts the table to a time series data set

min(traffic)
max(traffic)

ts.plot(traffic)
#increasing-ish trend, strong seasonal component, variability depends on time

#box-cox transformation
library(MASS)
t = 1:length(traffic)
fit = lm(traffic ~ t)
bcTransform = boxcox(traffic ~ t,plotit = TRUE)
#corresponds to a 95% confidence interval for the true value of lambda in the Box-Cox tranformation

#transforming with lambda from box-cox
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda
traffic.bc = (1/lambda)*(traffic^lambda-1) #transforms the time series data to the best lambda

#plot of the transformed data
op <- par(mfrow = c(1,2))
ts.plot(traffic,main = "Original data",ylab = expression(X[t]))
ts.plot(traffic.bc,main = "Box-Cox tranformed data", ylab = expression(Y[t]))
```

```
par(op)

#comparing the variance of both original data and transformed data
var(traffic)
var(traffic.bc)

#acf/pacf of the transformed data
op = par(mfrow = c(1,2))
acf(traffic.bc,lag.max = 60,main = "")
pacf(traffic.bc,lag.max = 60,main = "")
title("Box-Cox Transformed Time Series", line = -1, outer=TRUE)
par(op)

#variance unstable - log transform next
traffic.log <- log(traffic)
op <- par(mfrow = c(1,3))
ts.plot(traffic, main = "Original Data")
ts.plot(traffic.bc, main = "Box-Cox")
ts.plot(traffic.log, main = "log transform")
par(op)

#checking var of both log and box-cox transform
var(traffic.bc)
var(traffic.log) #traffic.log does not LOOK much different than traffic.bc BUT var is MUCH smaller

#checking acf/pacf of log and box-cox
op = par(mfrow = c(1,2))
acf(traffic.log,lag.max = 48,main = "")
pacf(traffic.log,lag.max = 48,main = "")
title("Log Transformed Time Series", line = -1, outer=TRUE)
par(op)

#checking sqrt tranformation of data
traffic.sqrt <- sqrt(traffic)
op <- par(mfrow = c(1,3))
ts.plot(traffic, main = "Original Data")
ts.plot(traffic.log, main = "Log Transform")
ts.plot(traffic.sqrt, main = "Square-Root Transform")
par(op) #again, graph doesn't LOOK much different

#checking variance of log and sqrt tranform
var(traffic.log)
var(traffic.sqrt) #traffic.sqrt has higher variance, use traffic.log for transformed data
```

```
#difference at lag 12 to remove seasonality
traffic.12 <- diff(traffic.log, lag = 12)
op = par(mfrow = c(1,3))
ts.plot(traffic.12, main = "")
acf(traffic.12,lag.max = 48,main = "")
pacf(traffic.12,lag.max = 48,main = "")
title("Differenced at Lag 12", line = -1, outer=TRUE)
par(op)

#difference at lag 1 to try and remove trend
traffic.1 <- diff(traffic.12, lag = 1)
op = par(mfrow = c(1,3))
ts.plot(traffic.1, main = "")
acf(traffic.1,lag.max = 48,main = "")
pacf(traffic.1,lag.max = 48,main = "")
title("Differenced at Lag 12 and Lag 1", line = -1, outer=TRUE)
par(op)

#checking var of diff lag 12 and diff lag 12 and lag 1
var(traffic.12)
var(traffic.1) #shows signs of overdifferencing, use traffic.12

#splitting data into train/test set and checking var/acf/pacf to double check it's similar to original data
t.train <- ts(traffic[1:162], start = c(1960,1), frequency = 12)
t.test <- traffic[163:180]
ts.plot(t.train)
ts.plot(traffic)

mean(t.train)
mean(t.test)

var(t.train)
var(t.test)

#transforming the training set and checking the var and acf/pacf
train.log <- log(t.train)
var(train.log)

#checking stationarity
train.12 <- diff(train.log, lag = 12)#differences the training data
adf.test(train.12, alternative = "stationary")#checks for stationarity, small p-values suggest stationarity

op = par(mfrow = c(1,2))
acf(train.log,lag.max = 48,main = "")
```

```
pacf(train.log,lag.max = 48,main = "")
title("Log Transform - Training Set", line = -1, outer=TRUE)
par(op)

library(forecast)
#fits the data with approx for best model
auto.arima(train.log, allowdrift = FALSE, trace = TRUE, allowmean = FALSE, ic = "aicc")

#best according to auto.arima
fit.202.211 <- arima(train.log, order = c(2,0,2), seasonal = list(order = c(2,1,1), period = 12), method =
'ML', include.mean = FALSE)
#next 5 best
fit.101.212 <- arima(train.log, order = c(1,0,1), seasonal = list(order = c(2,1,2), period = 12), method =
"ML", include.mean = FALSE)
fit.201.212 <- arima(train.log, order = c(2,0,1), seasonal = list(order = c(2,1,2), period = 12), method =
"ML", include.mean = FALSE)
fit.101.112 <- arima(train.log, order = c(1,0,1), seasonal = list(order = c(1,1,2), period = 12), method =
"ML", include.mean = FALSE)
#fit.102.212 <- arima(train.log, order = c(1,0,2), seasonal = list(order = c(2,1,2), period = 12), method =
"ML", include.mean = FALSE) - error code
fit.101.211 <- arima(train.log, order = c(1,0,1), seasonal = list(order = c(2,1,1), period = 12), method =
"ML", include.mean = FALSE)
fit.102.211 <- arima(train.log, order = c(1,0,2), seasonal = list(order = c(2,1,1), period = 12), method =
"ML", include.mean = FALSE)


library(qpcR)
#comparing AICc for best model
matrix(c(AICc(fit.202.211), AICc(fit.101.212), AICc(fit.201.212), AICc(fit.101.112), AICc(fit.101.211),
AICc(fit.102.211)), nrow = 1, dimnames = list("AICc", c("fit.202.211", "fit.101.212", "fit.201.212",
"fit.101.112", "fit.101.211","fit.102.211")))

#choose fit.201.212 as it has the lowest AICc
#compare fit.102.211 and fit.101.212
fit.201.212
fit.102.211
res.fit.212 <- residuals(fit.201.212)
res.fit.211 <- residuals(fit.102.211)

op = par(mfrow = c(1,2))
qqnorm(res.fit.212, main = "A")
qqnorm(res.fit.211, main = "B")
title("Q-Q Plot", line = -1, outer=TRUE)
par(op)
```

```
op = par(mfrow = c(1,2))
acf(res.fit.212, lag.max = 48, main = "A")
acf(res.fit.211, lag.max = 48, main = "B")
title("ACF Plots", line = -1, outer = TRUE)
par(op)

op = par(mfrow = c(1,2))
pacf(res.fit.212, lag.max = 48, main = "A")
pacf(res.fit.211, lag.max = 48, main = "B")
title("PACF Plots", line = -1, outer = TRUE)
par(op)

op = par(mfrow = c(1,2))
hist(res.fit.212, breaks = 10, main = "SARIMA(2,0,1)x(2,1,2)[12]")
hist(res.fit.211, breaks = 10, main = "SARIMA(1,0,2)x(2,1,1)[12]")
par(op)

shapiro.test(res.fit.212)
shapiro.test(res.fit.211)
h1 <- min(2*12,length(res.fit.212)/5)
h2 <- min(2*12,length(res.fit.211)/5)
h1
h2
Box.test(res.fit.212, lag = 12, type = "Box-Pierce", fitdf = 2)
Box.test(res.fit.211, lag = 12, type = "Box-Pierce", fitdf = 2)
Box.test(res.fit.212, lag = h1, type = "Box-Pierce", fitdf = 2)
Box.test(res.fit.211, lag = h2, type = "Box-Pierce", fitdf = 2)
Box.test(res.fit.212, lag = 12, type = "Ljung", fitdf = 2)
Box.test(res.fit.211, lag = 12, type = "Ljung", fitdf = 2)
Box.test(res.fit.212, lag = h1, type = "Ljung", fitdf = 2)
Box.test(res.fit.211, lag = h2, type = "Ljung", fitdf = 2)
#tests at lag 12 and h to test for serial correlation, but fails to reject null as p >0.05 so lags are not
serially correlated,thus residuals are independent.

#The AICc is lower on fit.201.212 but most of the p-values for the diagnostic checking are higher for
fit.102.211 as some of the p-values for fit.201.212 are just above the 0.05 value to not reject the null
hypothesis. Also, the Q-Q plot is not as heavy tailed on the fit.102.211 resid plot so, after comparison on
the 2 best fits, we choose fit.102.211

#forecasting
pred.tr <- predict(fit.101.212, n.ahead = 18) #predicts last 18 months
#95% CI
up.tr <- pred.tr$pred + 1.96*pred.tr$se
```

```
low.tr <- pred.tr$pred - 1.96*pred.tr$se
#plots the training data along with the predicted values
ts.plot(train.log, xlim = c(1960,1975), main = "Transformed Monthly Traffic Fatalities", ylab = "Log of
Number of Fatalities", xlab = "Month")
lines(up.tr, col = "blue", lty = "dashed")
lines(low.tr, col = "blue", lty = "dashed")
points(pred.tr$pred, col = "red", type = "l")

#plots all the data along with the predicted values
ts.plot(traffic.log, xlim = c(1960,1975), main = "Transformed Monthly Traffic Fatalities", ylab = "Log of
Number of Fatalities", xlab = "Month")
lines(up.tr, col = "blue", lty = "dashed")
lines(low.tr, col = "blue", lty = "dashed")
points(pred.tr$pred, col = "red", type = "l")
#the predicted values fall in line with the 95% CI and are very similar to real data with slight differences

#bringing predicted values back to original data
pred.og <- exp(pred.tr$pred)
up.og <- exp(up.tr)
low.og <- exp(low.tr)

ts.plot(t.train, xlim = c(1960,1975), main = "Monthly Traffic Fatalities", ylab = "Number of Fatalities",
xlab = "Month")
lines(up.og, col = "blue", lty = "dashed")
lines(low.og, col = "blue", lty = "dashed")
points(pred.og, col = "red", type = "l")

ts.plot(traffic, xlim = c(1960,1975), main = "Monthly Traffic Fatalities", ylab = "Number of Fatalities", xlab
= "Month")
lines(up.og, col = "blue", lty = "dashed")
lines(low.og, col = "blue", lty = "dashed")
points(pred.og, col = "red", type = "l")

#zooming in on the last 5 years
ts.plot(traffic, xlim = c(1970,1975), main = "Monthly Traffic Fatalities", ylab = "Number of Fatalities", xlab
= "Month")
lines(up.og, col = "blue", lty = "dashed")
lines(low.og, col = "blue", lty = "dashed")
points(pred.og, col = "red", type = "l")
```