Pstat 174 Final Project

Kenneth Villatoro

2023-12-8

Abstract

Throughout this final project, I analyzed the United Kingdom Monthly Driver Deaths from 1969 to 1984. The questions I addressed in this final project was how many deaths will occur in the United Kingdom due to car accidents in the next 10 months. This is very important as the goal is to predict these outcomes and from there indicate what measurements are needed in order to continue lowering that number. Car accidents are imperative, it happens throughout and is something that cannot be changed as it is a part of daily life. However, the amount of deaths can be reduced significantly through strict measurements and a change of transportation habits. The transformations I used in order to forecast these outcomes are box cox transformations and differencing from lags 1 and 12. The conclusions that come with the forecasting is that it was dramatically decrease, however, the deaths will pick up again over time.

Introduction

The Data set that I will be analyzing in this final project is the UK Monthly Drivers Death data set from 1969-1984. This is an important Data set as it highlights the risks that involve driving and overall,g oing out to do daily tasks. It is important to state with Monthly Driver Death is very important not just in today's society, but in general for the safety of all citizens. There are many benefits in forecasting Drivers Deaths, as seeing what the expected amount is for each month will help prioritize sending a message to all citizens to not drive recklessly and most of all, not under the influence.

Explatory Data Analysis

We start by loading the Data Set in the R Mark Down:

```
data("UKDriverDeaths")
uk_data <- UKDriverDeaths
head(uk_data)
```

```
##       Jan  Feb  Mar  Apr  May  Jun
## 1969 1687 1508 1507 1385 1632 1511
```

Training/Testing Split

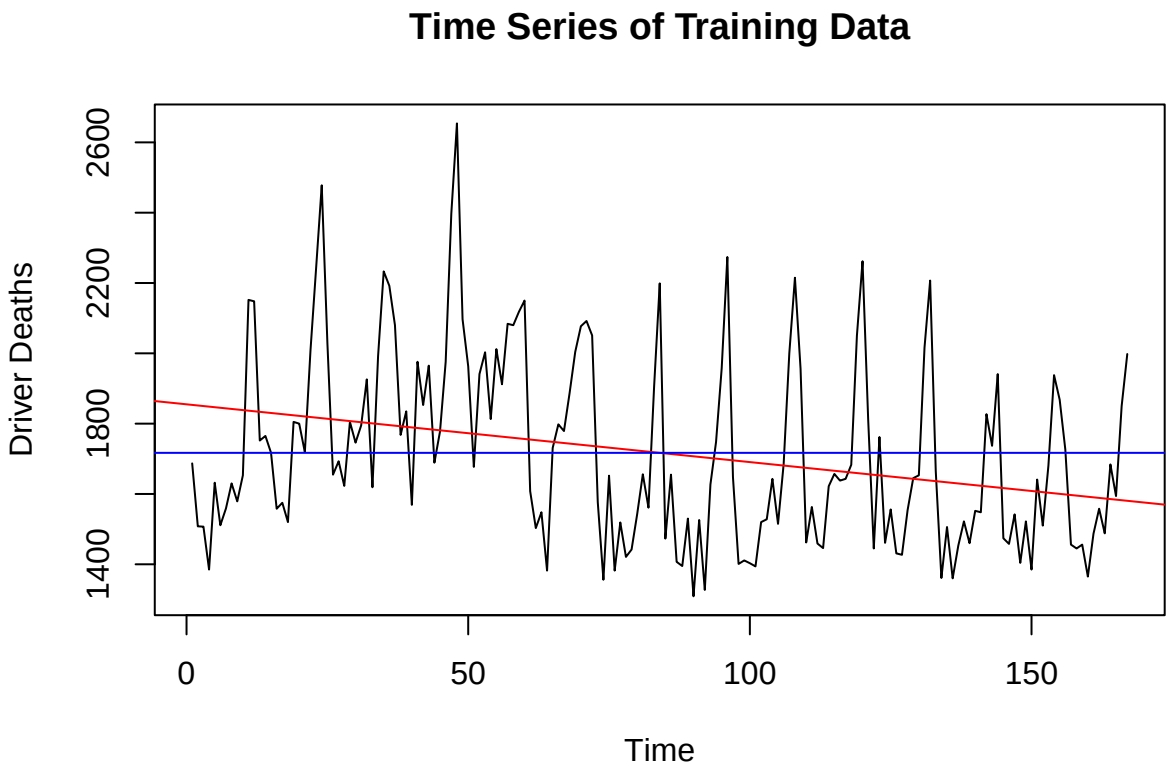
We then divide data to training and test sets. We will use training set for modeling and tests set for validation. For the splits, the years from 1969-1984 will be for the training set and 1984 will be for the testing set.

```
training_uk = uk_data[c(1:167)]
testing_uk= uk_data[c(168:180)]
```

Data Vizualation of Time Series

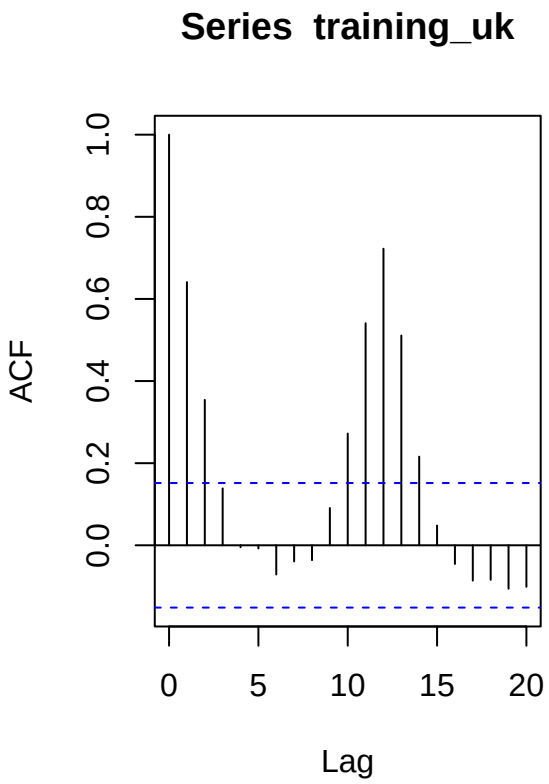
We plot the Time Series of the Training Set down below. We see that the Linear Trend is decreasing as the Years go on which is great, while the mean is constant at approximately 1700.

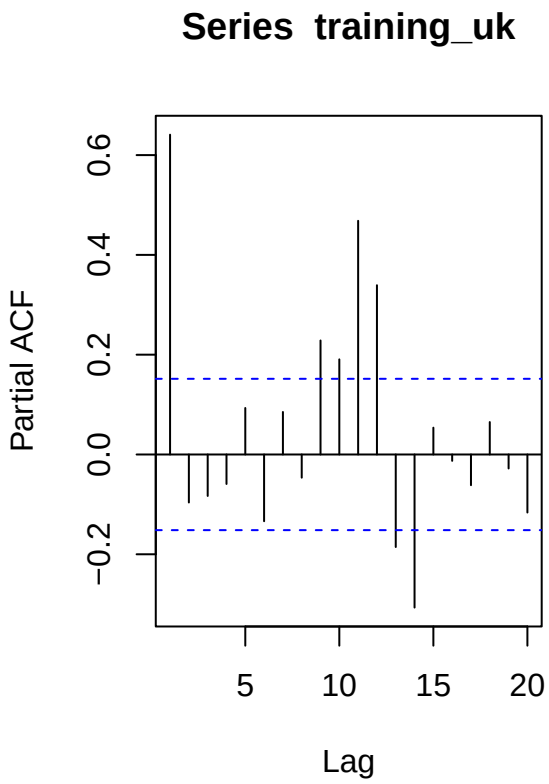
```
plot.ts(as.numeric(training_uk),main = "Time Series of Training Data", ylab = "Driver Deaths")
ntr=length(as.numeric(training_uk))
fit_train <- lm(as.numeric(training_uk) ~ as.numeric(1:ntr))
abline(fit_train, col="red")
abline(h=mean(as.numeric(training_uk)), col="blue")
```



Analyzing the ACF and PACF of the Training Data set, we see that the ACF and PACF are well outside the confidence interval at around lags 10-15, lag 2 and lag 3. With these results, it is imperative to either difference at either lag 1 or both at lag 1 and lag 12.

```
par(mfrow = c(1,2))
acf(training_uk, lag.max = 20)
pacf(training_uk, lag.max = 20)
```





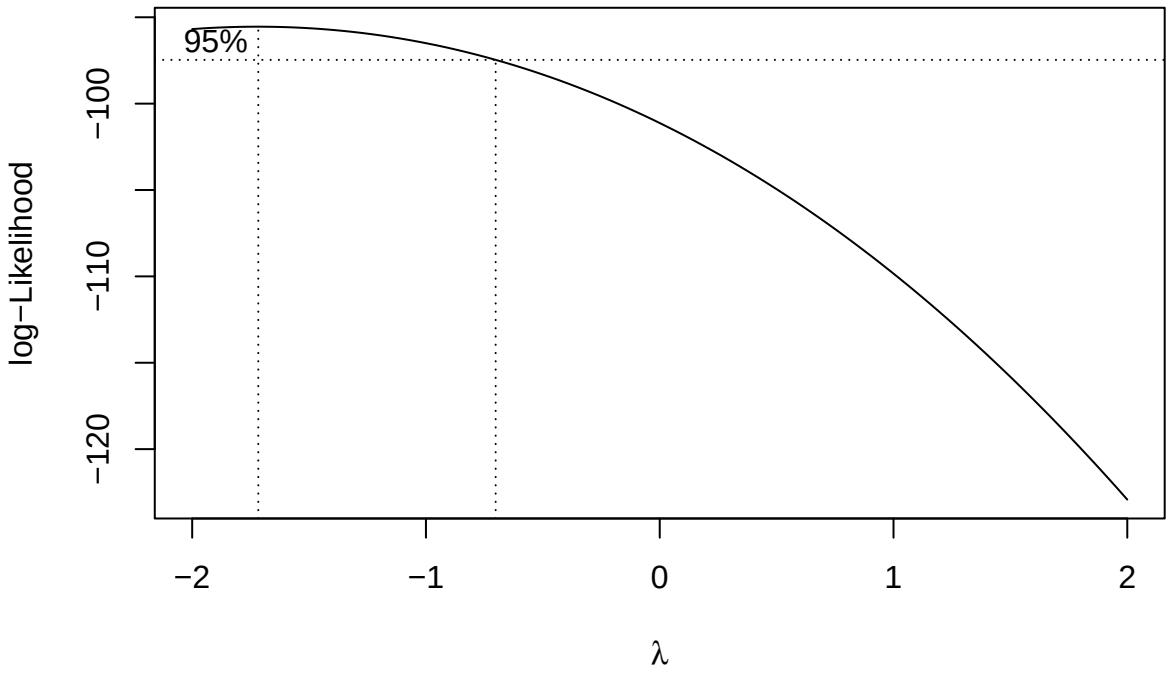
Transformations of Time Series

The transformation that will be used in order to make the time series a stationary series is a Box cox transformation. The optimal

$$\lambda$$

received in this transformation is -1.717.

```
#difference at lag 1 and 4
t <- 1:length(training_uk)

boxcox_uk <- boxcox(training_uk ~ t, plotit = T)
```

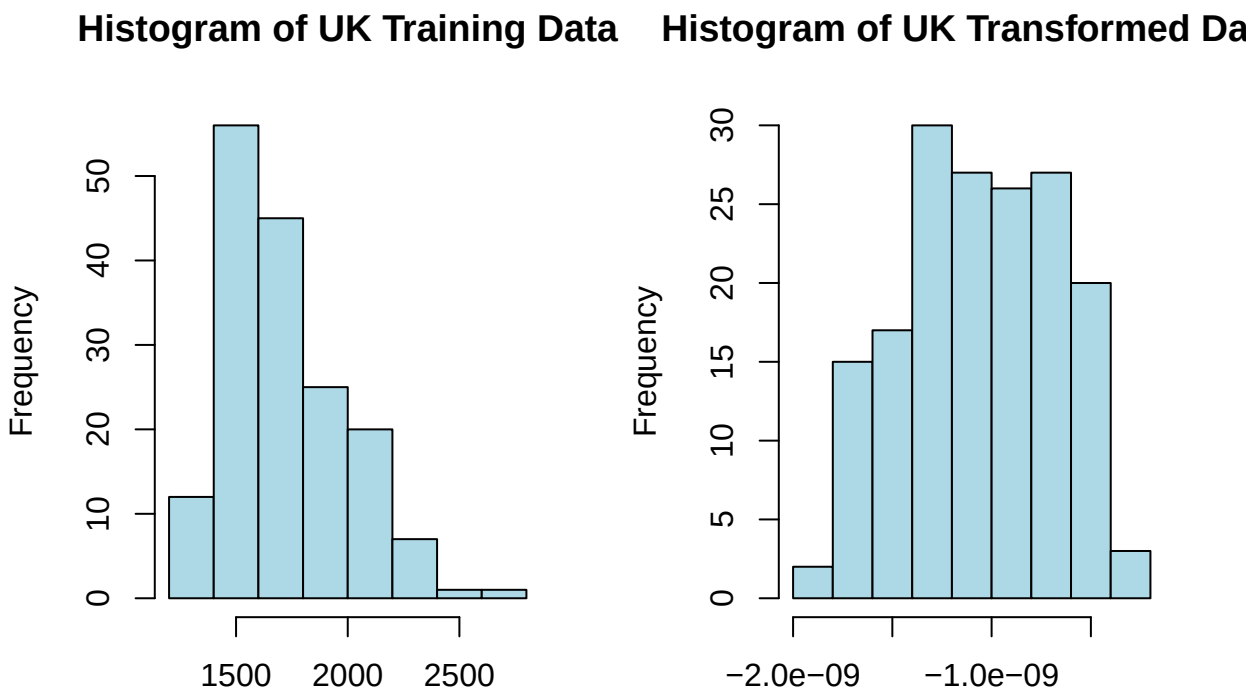



```
lambda <- boxcox_uk$x[which.max(boxcox_uk$y)]

uk_bc <- (1/lambda) * (training_uk**(lambda-1))

lambda
```

```
## [1] -1.717172
```

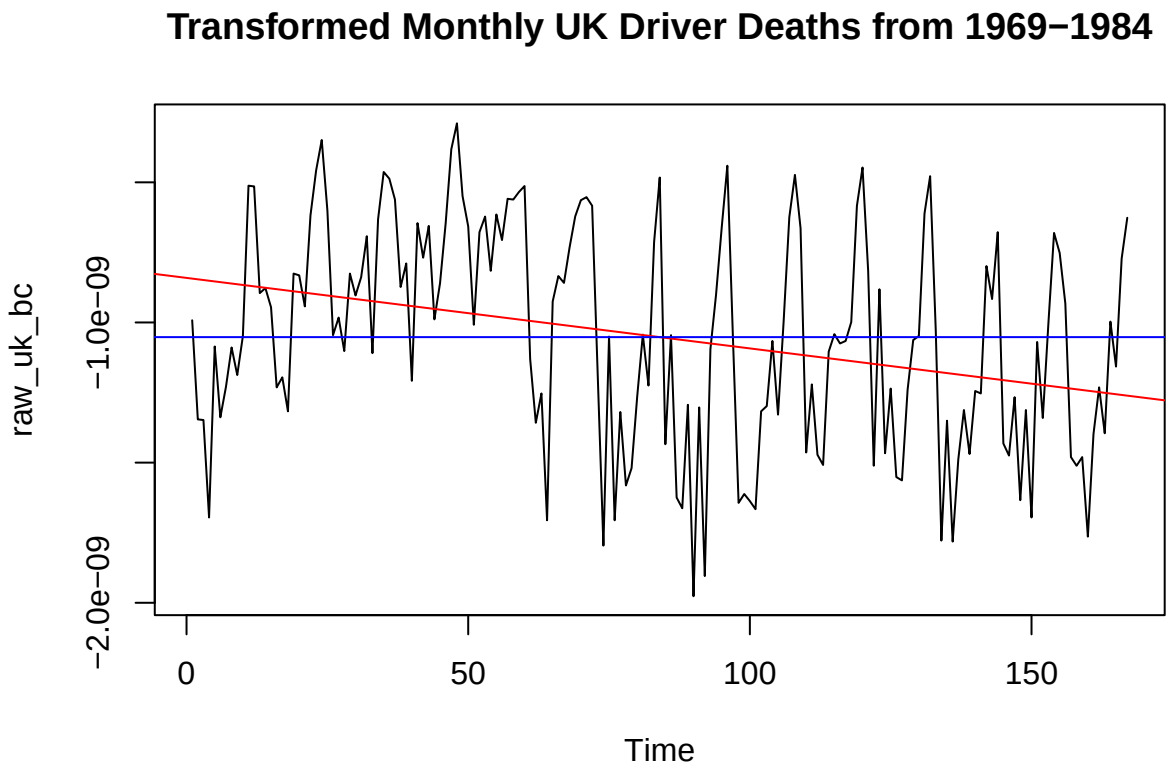
Then we compare a histogram of the training data alongside with the box cox transformed data. We analyze that there are more bins in the transformed variable as the box cox method is trying to make the data more Gaussian.

```
par(mfrow=c(1,2))
hist(training_uk, col="light blue", xlab="", main="Histogram of UK Training Data")
hist(uk_bc, col="light blue", xlab="", main="Histogram of UK Transformed Data",
     breaks=10)
```



Continuing with the box cox transformation, raw_uk_bc is a numeric form of the box cox time series with the purpose of plotting and seeing how well the transformed time series performed. We analyze that the mean is constant around zero which is a good sign and the linear trend decreasing as time goes on.

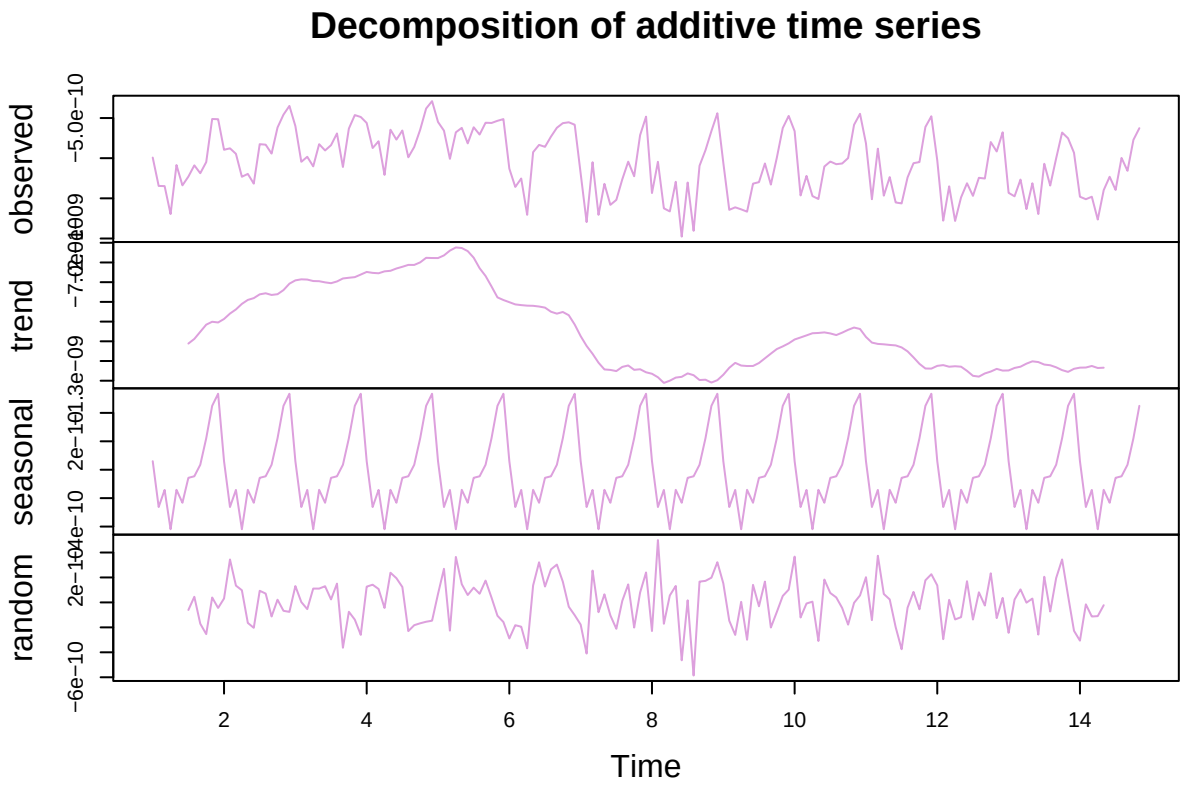
```
raw_uk_bc <- as.numeric(uk_bc)
plot.ts(raw_uk_bc, main="Transformed Monthly UK Driver Deaths from 1969-1984") # to generate trend and
nt=length(raw_uk_bc)
fit <- lm(raw_uk_bc ~ as.numeric(1:nt))
abline(fit, col="red")
abline(h=mean(raw_uk_bc), col="blue")
```



Decomposition of the Boxcox Transformation

We then decompose the box cox transformation with the frequency equaling to 12 due to the time series being monthly. Observing the decomposition of the box cox transformation, we still see there is some linear trend with the addition of having a seasonality component as well. In order to continue with this transformation, there must be differencing at lags 1 and 12.

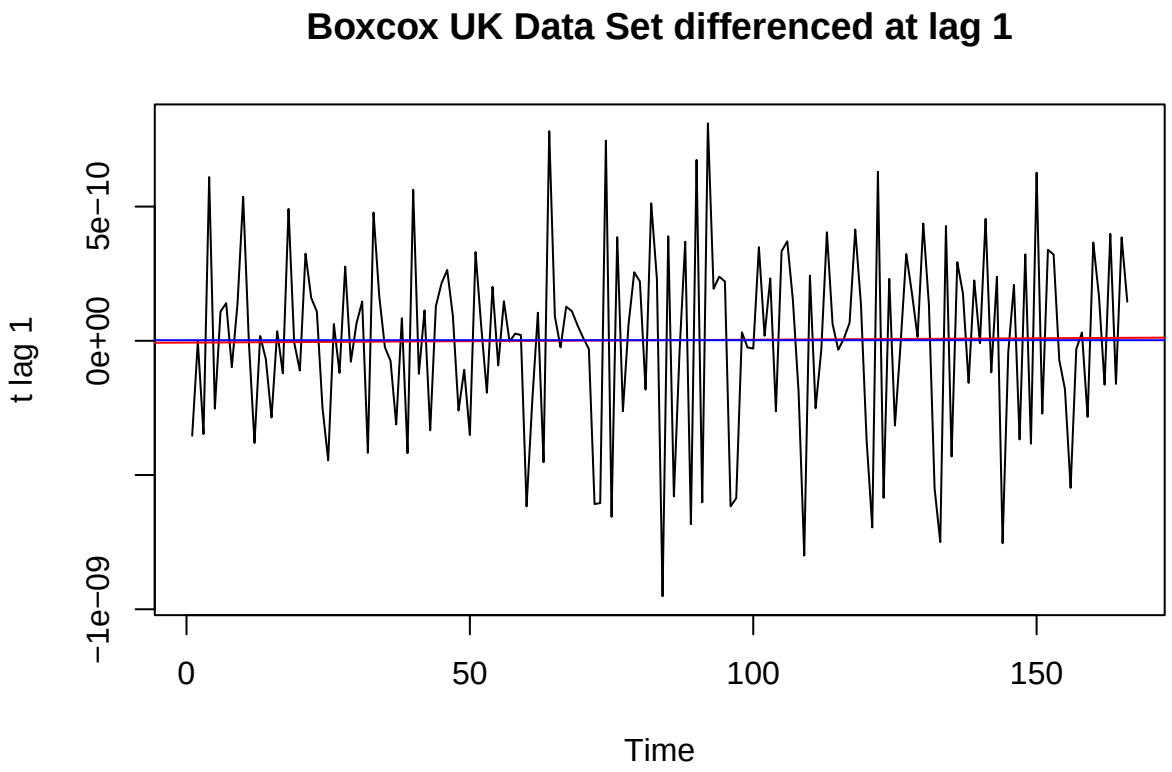
```
y <- ts(as.ts(raw_uk_bc), frequency = 12)

decom <- decompose(y)

plot(decom, col = "plum")
```



Differencing Boxcox Transformation at Lags 1 and 12 with Checking

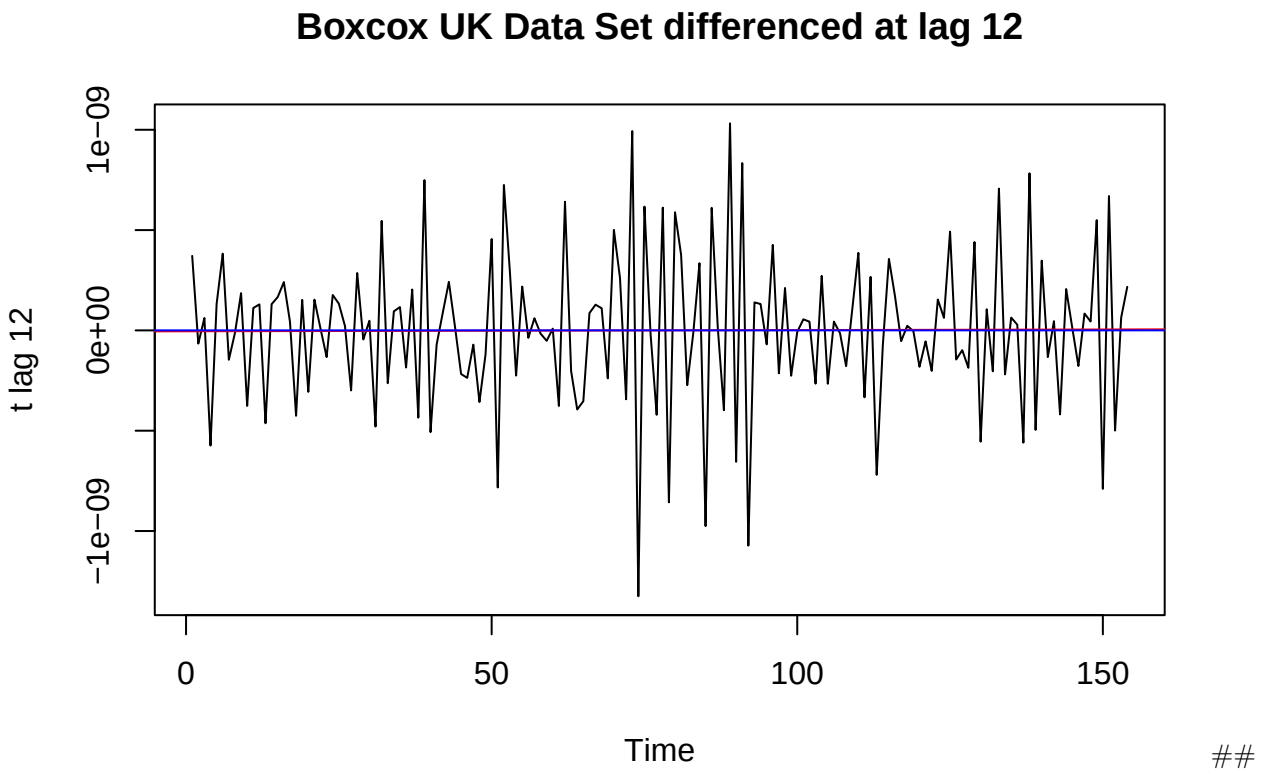
We first start with the differencing of the box cox transformation at lag 1. The result of the Linear Model plot shown below is white noise, however, we analyze significant ups and down of the data. This is not a good sign as the transformed time series may be not be stationary. On the positive end, we do see the mean and fitted line almost the same which does show that the linear trend was eliminated.

```
y_1 <- diff(uk_bc, 1)
plot.ts(y_1, main= "Boxcox UK Data Set differenced at lag 1",ylab="GDP differenced a
t lag 1")
fit_11 <- lm(y_1 ~ as.numeric(1:length(y_1)))
abline(fit_11, col="red")
abline(h=mean(y_1), col="blue")
```



Proceeding onto differencing to lag 12, this differencing is remvoing the seasonal component of the box cox transformed time series. Analyzing the graph shown below,

```
#Plot of y_12

y_12 <- diff(y_1,12)

plot.ts(y_12, main= "Boxcox UK Data Set differenced at lag 12", ylab="GDP differenced a
t lag 12")
fit_12 <- lm(y_12 ~ as.numeric(1:length(y_12)))
abline(fit_12, col="red")
abline(h=mean(y_12), col="blue")
```



##

Variance Checking ##

We then proceed onto analyzing the variance between The Original Data, The box cox transformation differenced at lag 1, and the box cox transformation differenced at lags 1 and 12. We see that we significantly lowered the variance when transforming and differencing the time series. Y_1 has the lowest variance, which is the model we will proceed with.

```
Variance_Comparison <- c(var(uk_data), var(y_1), var(y_12))

analyze_1 <- as.data.frame(Variance_Comparison, row.names = c("Original", "Y_1", "Y_12"))

analyze_1
```

```
##          Variance_Comparison
## Original        8.387451e+04
## Y_1             1.195401e-19
## Y_12            1.505463e-19
```

Analyzing ACF and PACF of Differencing Transformation

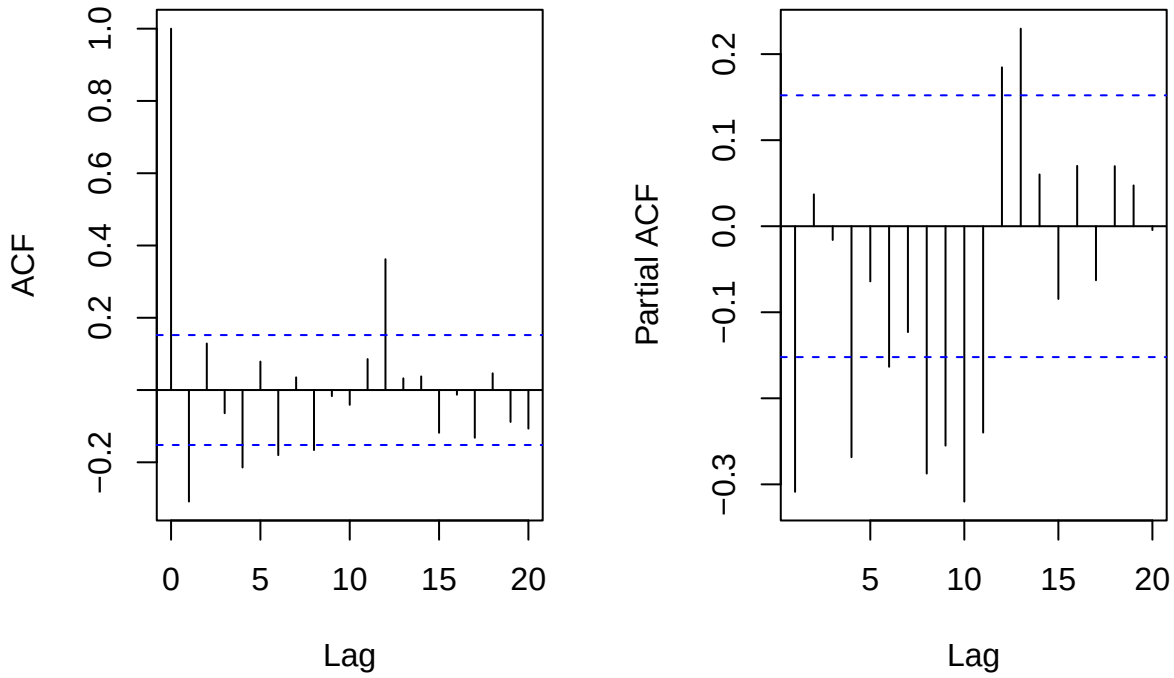
The ACF plot shows that at lag 12, the ACF are outside the confidence interval. On the other hand, for the PACF we see that the PACF at lag 12 and 13 is outside the confidence interval. Our values for p, d, q and P, D, and Q will be:

q = 1 d = 1 p = 4

P = 1 or 2 D = 0 Q = 1 or 2

```
par(mfrow = c(1,2))
acf(y_1, lag.max = 20, main = "Training GDP Dataset after Lag 12 Differencing")
pacf(y_1, lag.max = 20, main = "Training GDP Dataset after Lag 12 Differencing")
```

ning GDP Dataset after Lag 12 Diffening GDP Dataset after Lag 12 Diffe



Fitting Models according to Time Series Data

Estimating AR parameters of Boxcox transformation

We first start off estimating the AR parameters in order to identify p. We see that the simulation concludes that an AR(13) model is best suited for this data.

```
ar(y_1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = y_1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
## Coefficients:
##       1        2        3        4        5        6        7        8
## -0.6139  -0.2497  -0.2387  -0.3968  -0.2588  -0.3023  -0.2714  -0.3176
##       9       10       11       12       13
## -0.2773  -0.3076  -0.0622   0.3158   0.2296
##
## Order selected 13  sigma^2 estimated as  6.894e-20
```

Fitting Models/Model Identifitication

Three models will be fitted and then put to the test based on AICc in order to determine which is the best model to move forward with. For the first model, we will fit an AR(14) model. For the second model, we will fit an SARIMA(1,1,4) * (1,0,0) ^12 model. And finally, for the third model, we will fit a SARIMA(4,1,12) * (3,0,0)^12

Model 1: AR(13)

```
model_1 <- arima(uk_bc, order=c(13,1,0), seasonal = list(order = c(0,1,0), period = 12), method="ML")

model_1
```

```
##
## Call:
## arima(x = uk_bc, order = c(13, 1, 0), seasonal = list(order = c(0, 1, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.7830  -0.4081  -0.3104  -0.4844  -0.3539  -0.4188  -0.3195  -0.3142
## s.e.   0.0789   0.0925   0.0984   0.1020   0.1083   0.1098   0.1113   0.1086
##           ar9     ar10     ar11     ar12     ar13
##       -0.1391  -0.0772  -0.0705  -0.4859  -0.2013
## s.e.   0.1079   0.1007   0.0976   0.0924   0.0801
##
## sigma^2 estimated as 5.691e-20:  log likelihood = 3191.29,  aic = -6354.58
```

Model 2: SARIMA(1,1,4) * (2,0,0) ^12

```
model_2 <- arima(uk_bc, order=c(1,1,4), seasonal = list(order = c(2,0,0), period = 12), method="ML")

model_2
```

```
##
## Call:
## arima(x = uk_bc, order = c(1, 1, 4), seasonal = list(order = c(2, 0, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1     ma1      ma2      ma3      ma4    sar1    sar2
##       -0.7662  0.0778  -0.3932  -0.0175  -0.3928  0.4495  0.2889
## s.e.   0.0798  0.0937   0.0790   0.0803   0.0712  0.0776  0.0800
##
## sigma^2 estimated as 5.425e-20:  log likelihood = 3441.98,  aic = -6867.96
```

Model 3: SARIMA(4,1,12) * (3,0,0)^12

```
model_3 <- arima(uk_bc, order=c(4,1,12), seasonal = list(order = c(3,0,0), period = 12), method="ML")
```

```
## Warning in arima(uk_bc, order = c(4, 1, 12), seasonal = list(order = c(3, :
## possible convergence problem: optim gave code = 1
```

```
model_3
```

```
##
## Call:
## arima(x = uk_bc, order = c(4, 1, 12), seasonal = list(order = c(3, 0, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1     ar2      ar3      ar4      ma1      ma2     ma3     ma4
##       -0.4261  0.5813  -0.0952  -0.7337  -0.3326  -0.8372  0.5402  0.3651
## s.e.   0.1062  0.1197   0.0873   0.0726   0.1466   0.1162  0.1399  0.1516
##           ma5     ma6      ma7      ma8     ma9     ma10    ma11    ma12
```

```
##       -0.4481  0.1085  -0.1908  -0.0423  0.2088  -0.1260  0.3027  0.1743
## s.e.   0.1356  0.1276   0.1275   0.1195  0.1328   0.1269  0.1001  0.1178
##         sar1    sar2    sar3
##       0.1926  0.3560  0.2194
## s.e.  0.0874  0.0906  0.0957
##
## sigma^2 estimated as 4.248e-20:  log likelihood = 3455.34,  aic = -6870.67
```

Comparing AICc of all three models

Comparing the three models that we fitted, we conclude that model_2 has the lowest AICc and therefore contiune with this model.

```
AICc(model_1)
```

```
## [1] -6352.197
```

```
AICc(model_2)
```

```
## [1] -6867.253
```

```
AICc(model_3)
```

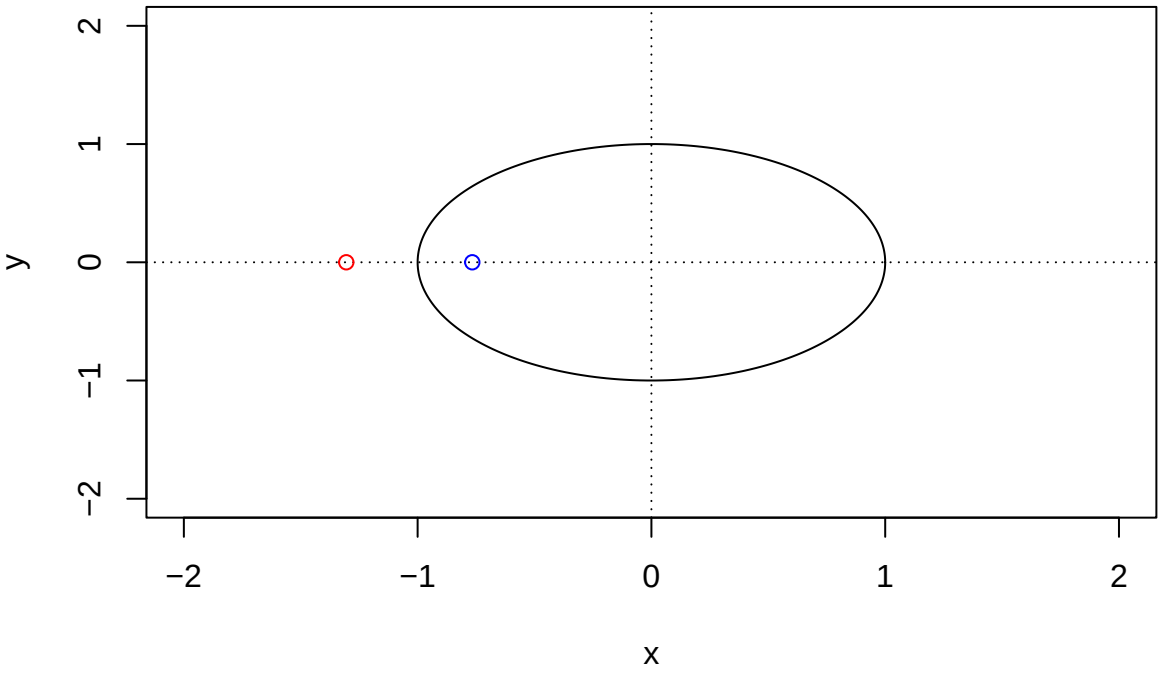
```
## [1] -6865.502
```

Checking Model Stationary/Invertibility

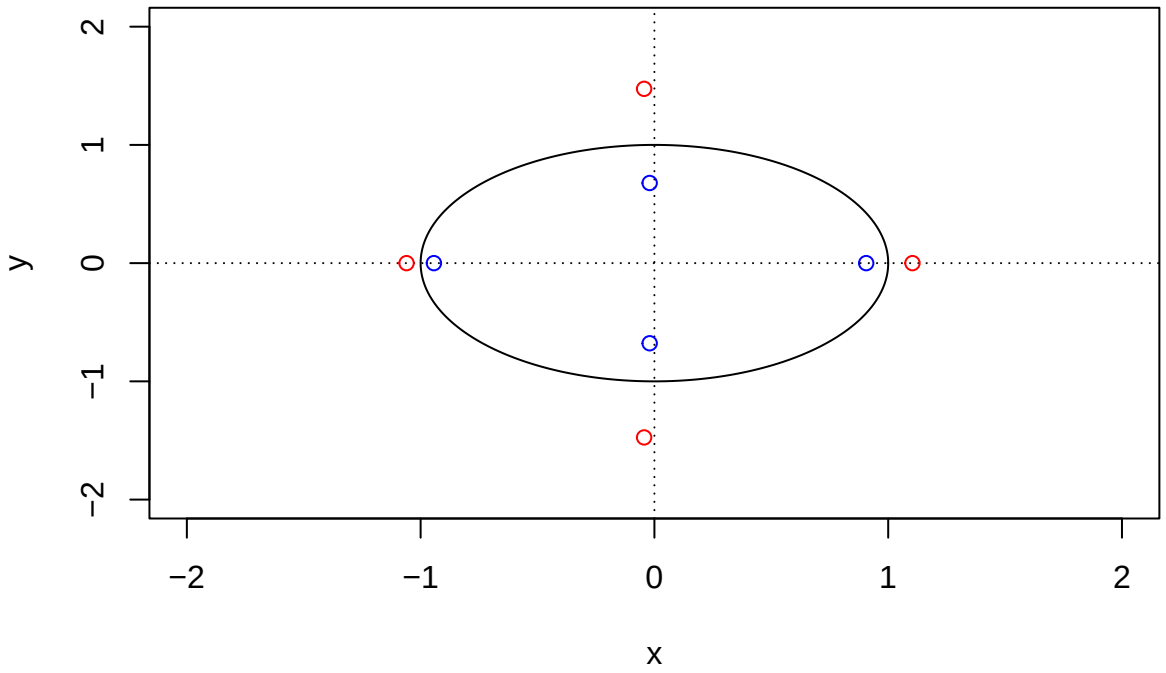
We then check for stationary and invertibility for Model 2:

Model 2:

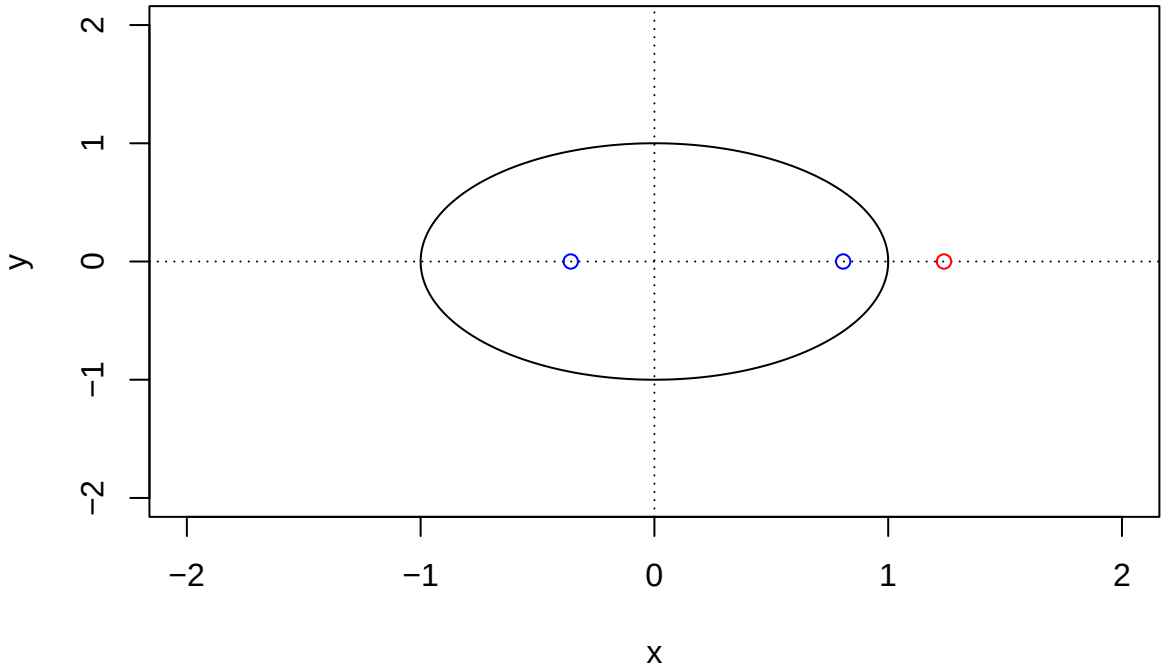
```
#AR
plot.roots(polyroot(c(1, 0.7662)))
```




```
#MA
plot.roots(polyroot(c(1, 0.0778 , -0.3932 , -0.0175 , -0.3928)))
```




```
#SAR
plot.roots(polyroot(c(1, -0.4495 , -0.2889)))
```



We see that based on the polyroots, Model 2 is stationary and invertible.

Final Model and Formula

The final model that will be used in order to forecast with be Model 3 as it has the lowest AICc and is stationary and invertible.

The formula of this model is:

$$(1 - 0.4495B^{12} - 0.2889B^{24})(1 + 0.7662B)Y_t = (1 + 0.0778B - 0.3932B^2 - 0.017B^3 - 0.3928B^4)Z_t$$

where:

$$Y_t = U_t^{1/(-1.71)}$$

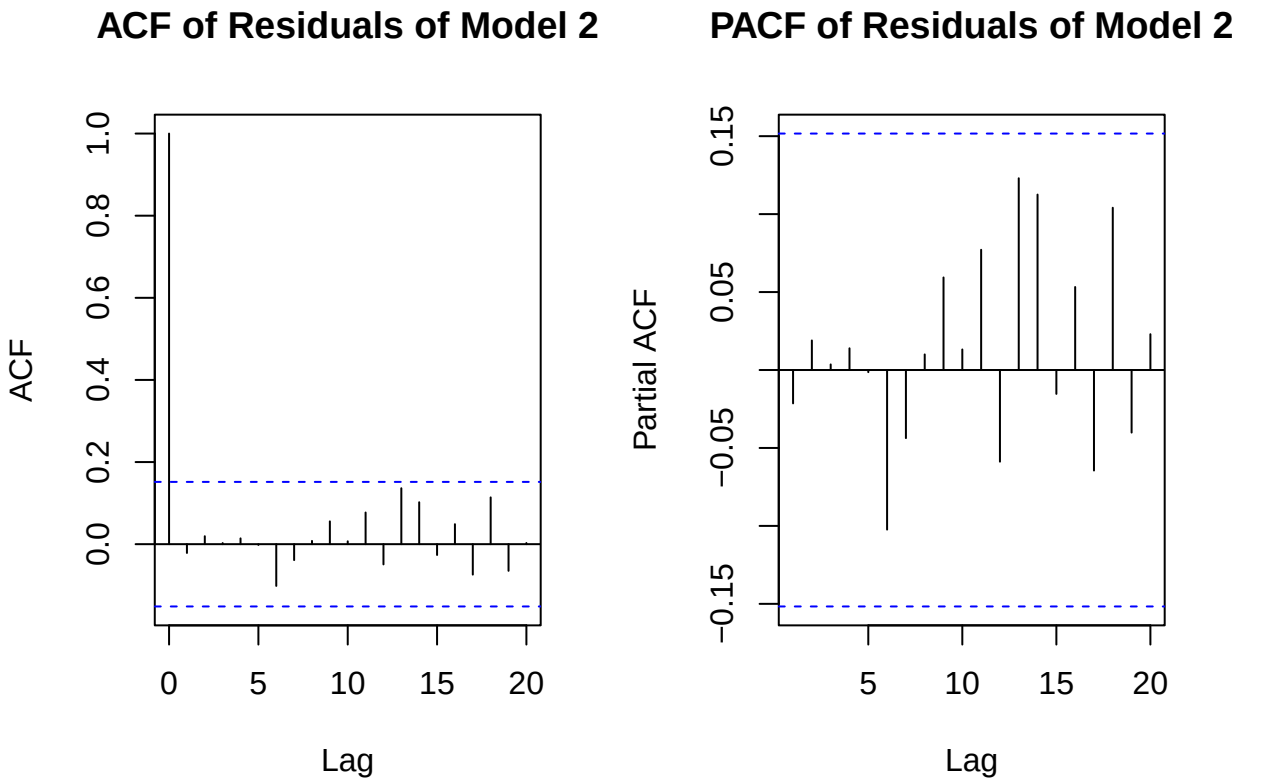
$$U_t$$

is the original time series.

Diagonsitic Checking

We then proceed to Diagnostic Checking in order to see if the residuals align with white noise and pass all Portmanteau Statistics. We first check if Model 2 is white noise by taking the residuals and analyzing the ACF and PACF plots. We see that all lags are inside the confidence interval. We can also conclude the residuals are white noise as the AR estimates of the residuals is 0.

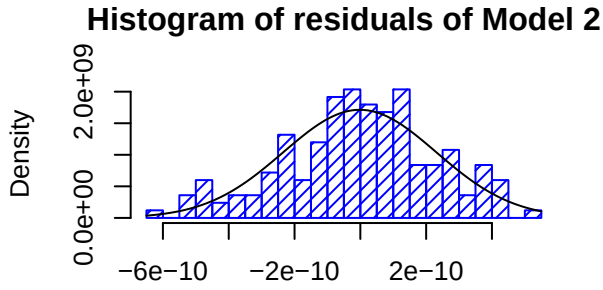
```
residuals_2 = residuals(model_2)
par(mfrow = c(1,2))
acf(residuals_2, lag.max = 20, main = "ACF of Residuals of Model 2")
pacf(residuals_2, lag.max = 20, main = "PACF of Residuals of Model 2")
```

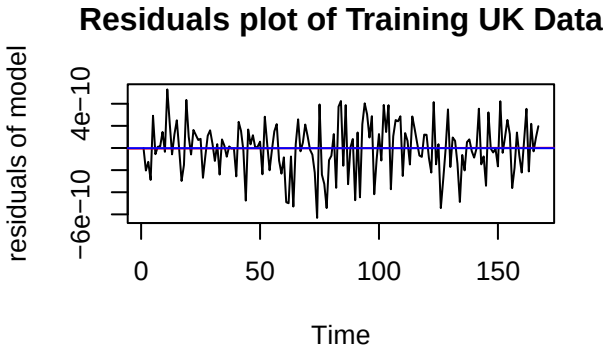


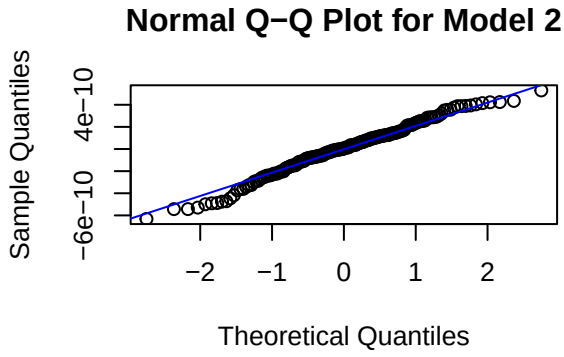

```
ar(residuals_2, aic = TRUE, order.max = NULL, message = c("yule-walker"))
```

```
##
## Call:
## ar(x = residuals_2, aic = TRUE, order.max = NULL, message = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  5.425e-20
```

```r
#Model 2

res_2 = residuals(model_2)
par(mfrow=c(2,2))
hist(res_2,density=20,breaks=20, col="blue", xlab="", prob=TRUE,
     main="Histogram of residuals of Model 2")

m <- mean(res_2)
std <- sqrt(var(res_2))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res_2,ylab= "residuals of model",
        main="Residuals plot of Training UK Data")
fitt <- lm(res_2 ~ as.numeric(1:length(res_2)))
abline(fitt, col="red")
abline(h=mean(res_2), col="blue")
qqnorm(res_2,main= "Normal Q-Q Plot for Model 2")
qqline(res_2,col="blue")
```







Portmanteau Statistics

We then proceed to check the Portmanteau Statistics in order to ensure that the model is ready for forecasting. We see that Model 2 passes normality, Box-Pierce, Ljung-Box, and Mcleod-Li test. Thus, Model 2 is ready for forecasting.

```r
#Model 2 Checking

#Shapiro test for normality
shapiro.test(res_2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_2
## W = 0.98629, p-value = 0.1011
```

```
#Box-Pierce test
Box.test(res_2, type = c("Box-Pierce"), lag = 13, fitdf = 7)
```

```
##
##  Box-Pierce test
##
## data:  res_2
## X-squared = 7.1887, df = 6, p-value = 0.3038
```

```
#Ljung-Box test
Box.test(res_2, type = c("Ljung-Box"), lag = 13, fitdf = 7)
```

```
##
##  Box-Ljung test
##
## data:  res_2
## X-squared = 7.7516, df = 6, p-value = 0.2569
```

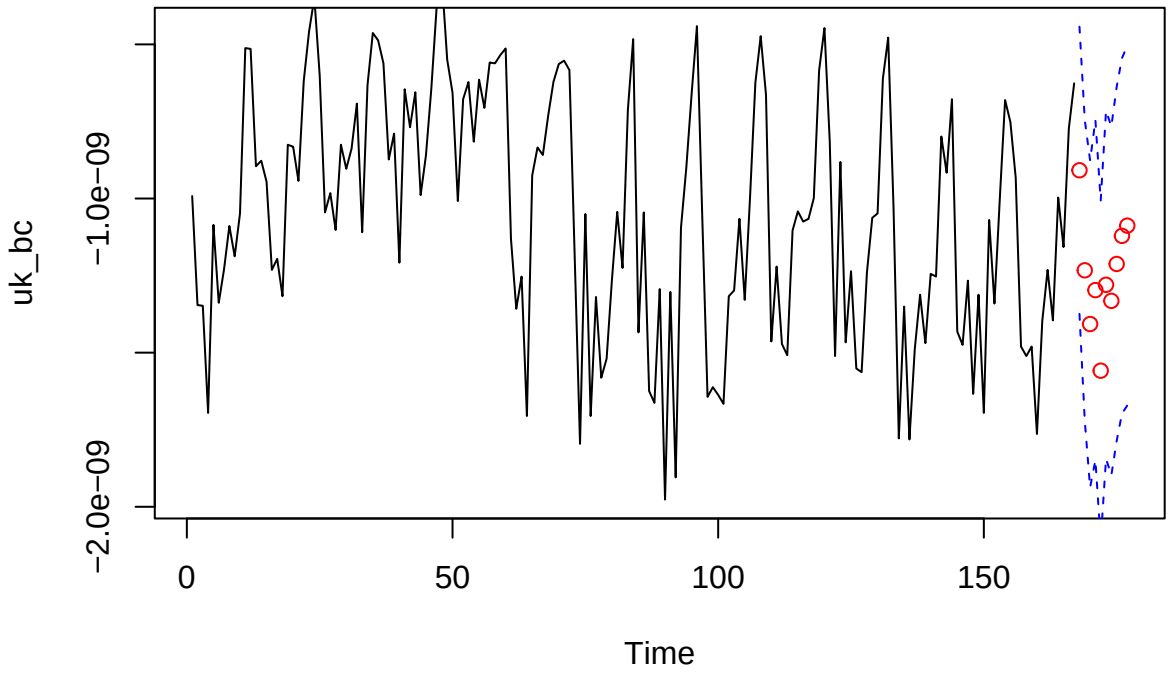
```
#McLeod-Li test
Box.test(res_2**2, type = c("Ljung-Box"), lag = 13, fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res_2^2
## X-squared = 19.688, df = 13, p-value = 0.1033
```

Forecasting of the Time Series Data

We then proceed to forecasting of Model 2, predicting the outcome for the next 10 months. We see that all the outcomes are inside the confidence interval, which is a good sign of our predicting in terms of the model.

```
#plot of forecasting using box-cox transformation
library(forecast)

pred.tr <- predict(model_2, n.ahead = 10)

U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(uk_bc, xlim=c(1,length(uk_bc)+10), ylim = c(min(uk_bc),max(U.tr))) #plot y.tr and forecast
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(uk_bc)+1):(length(uk_bc)+10), pred.tr$pred, col="red")
```



Conclusion

In conclusion, the forecasting of the predicted values did match the SARIMA model we fitted previously. However, we see that there is an increase in terms of the Deaths in the UK moving forward. This is not good for our outcome as we would like for our monthly Deaths to decrease as time goes on. This calls for action in terms of transportation regulations, a spread of message calling all drivers to drive safely, and bring up this outcome to a national level. This observation will heavily benefit those who value the safety of all citizens and those who care about their community.

Contributers to Project: Kenneth Villatoro, Professor Raisa Feldman

Data Libraries used in Final Project: plot.roots.r()