Reported Cases of Mumps Forecasting

Peter Chu

12/10/2022

Abstract

In this project, I want to forecast the reported number of cases of mumps in New York city for the years 1966 and 1967 based on data starting in 1953. The data comes from the tsdl library / package and is in the Health category. The techniques I used to be able to achieve this include first splitting the data into two sets, one training one testing, to assess model performance and analyzing the training set. Then I made it stationary by differencing at certain lags before fitting multiple possible models using the modified data's ACF and PACF graphs. Then I performed model diagnostics on a couple of the best models which involved checking for White Noise residuals and normality. Finally I used the best model that passed all diagnostic checks to forecast data to the training set and used the testing set, the actual values, to assess the model's performance. The end result was that I used a SARIMA$(3,1,2) \times (0,1,2)_{12}$ model which forecasted the last 12 data well.

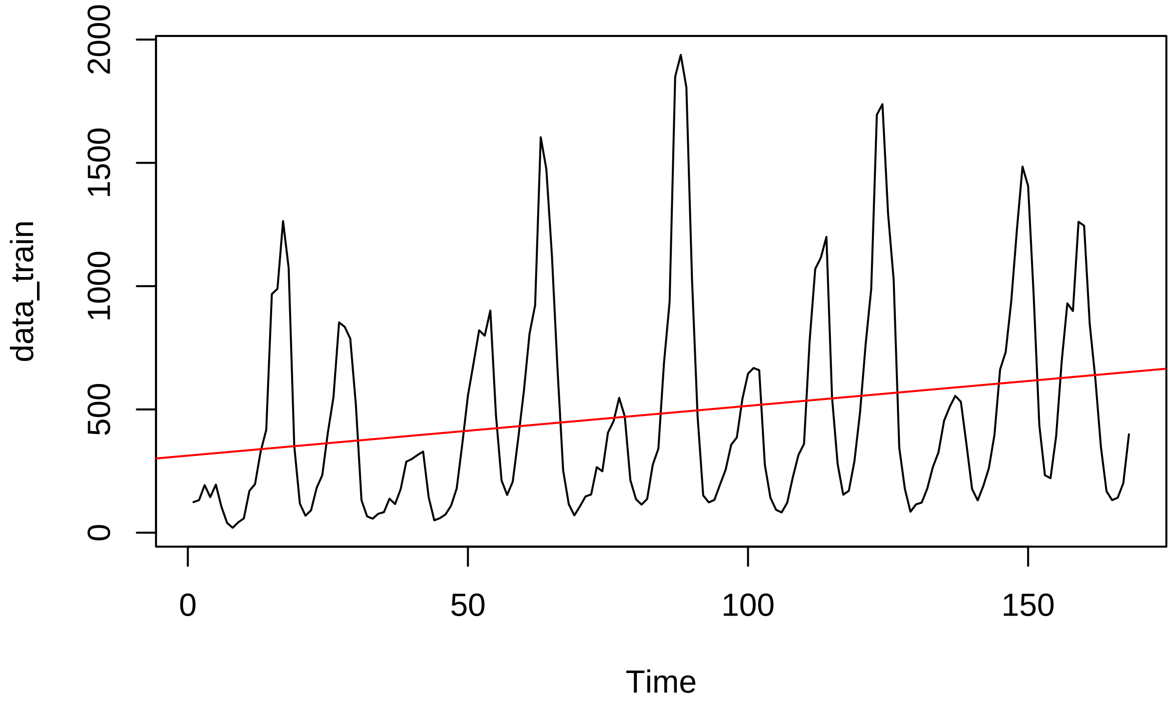
Main Body Of The Report

I used the dataset "Monthly reported number of cases of mumps, New York city, 1928-1972" in the tsdl library. As the title of the dataset suggests, it contains monthly data on the number of reported cases of mumps in New York City from 1928 to 1972. I chose to keep only 15 years from 1953 to 1968 as the larger dataset would take a while for models to load and test on. Also I believe that 15 years was a sufficient number of data points needed for this project. I chose this dataset as my mother is a nurse and has been for the past 32 years. She has experienced all the worst health epidemics especially recently the COVID 19 pandemic. She would work long hours risking her life to help others. My mother is the reason why I am so interested in bio-statistics which this project covers.

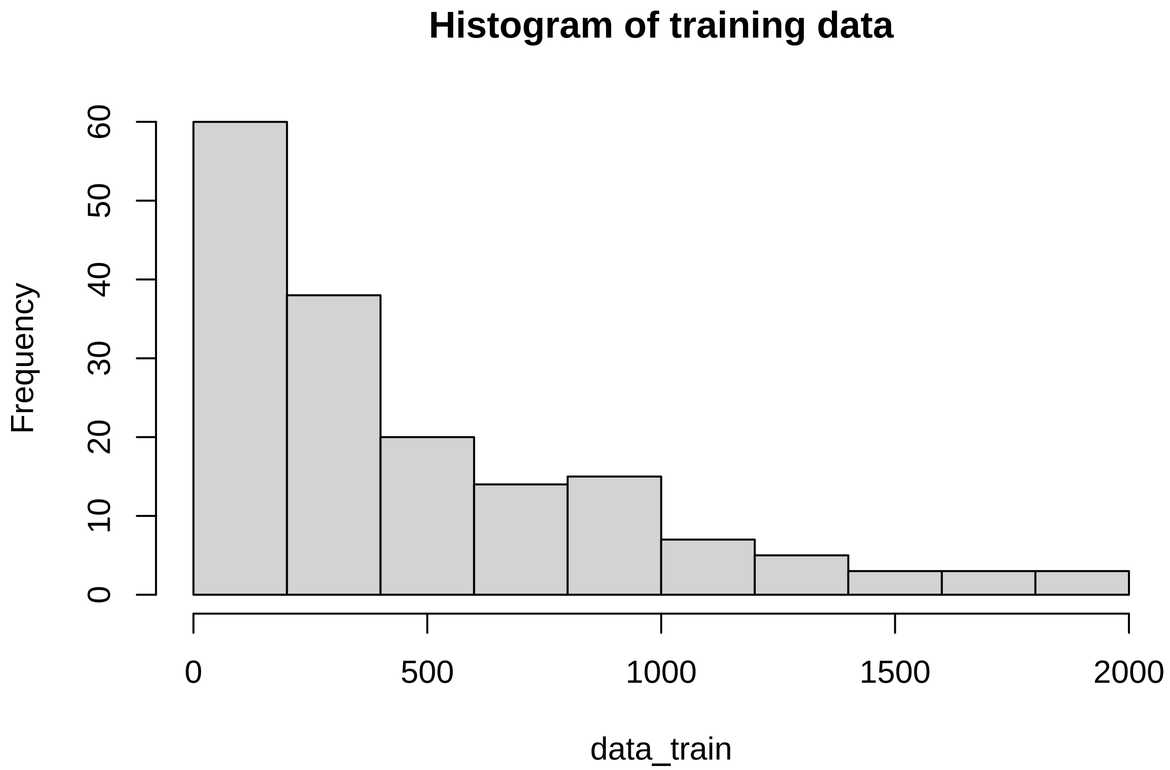
For the analysis, I started by first splitting data so that I could have data to train and test my models on. Upon initial inspection the data was non-stationary and had a slight trend. To make the data more stationary I applied a box-cox transformation and found λ to be 0.060606. Then I did a sqrt and log transformation. Overall the box-cox transformation was the most stable so I continued to use that data. I then differenced at lag = 1 and at lag = 12 to remove the trend and seasonality. The data looked much more stationary after doing so. In addition, comparing the means and variances of histograms along the way confirmed that the data was becoming more stationary.

After finalizing my transformation I looked at the ACF and PACF of the final data. From there I was able to figure out that a SARIMA$(3,1,2) \times (0,1,2)_{12}$ and SARIMA$(1,1,2) \times (0,1,2)_{12}$ models fit best. Since both models were good candidates I continued to perform model diagnostics on them. Both had White Noise residuals, appeared symmetrical / stable in a histogram, normal from Q-Q plots, and their ACF's were in the confidence intervals. However, I found that for the first model it had a Yule-Walker Selected Order of 0 while the second model had a Yule-Walker Selected Order of 2. Therefore, I could not use the second model. Thus I could only forecast using the first model. However, I ended up deciding to use both models and compare them. As expected the first model did better and had forecasted values closer to the actual values. There were some errors in the forecasted values that can be attributed to the model not being the best possible. Overall, the model performed well.

Plot and analyze the time series



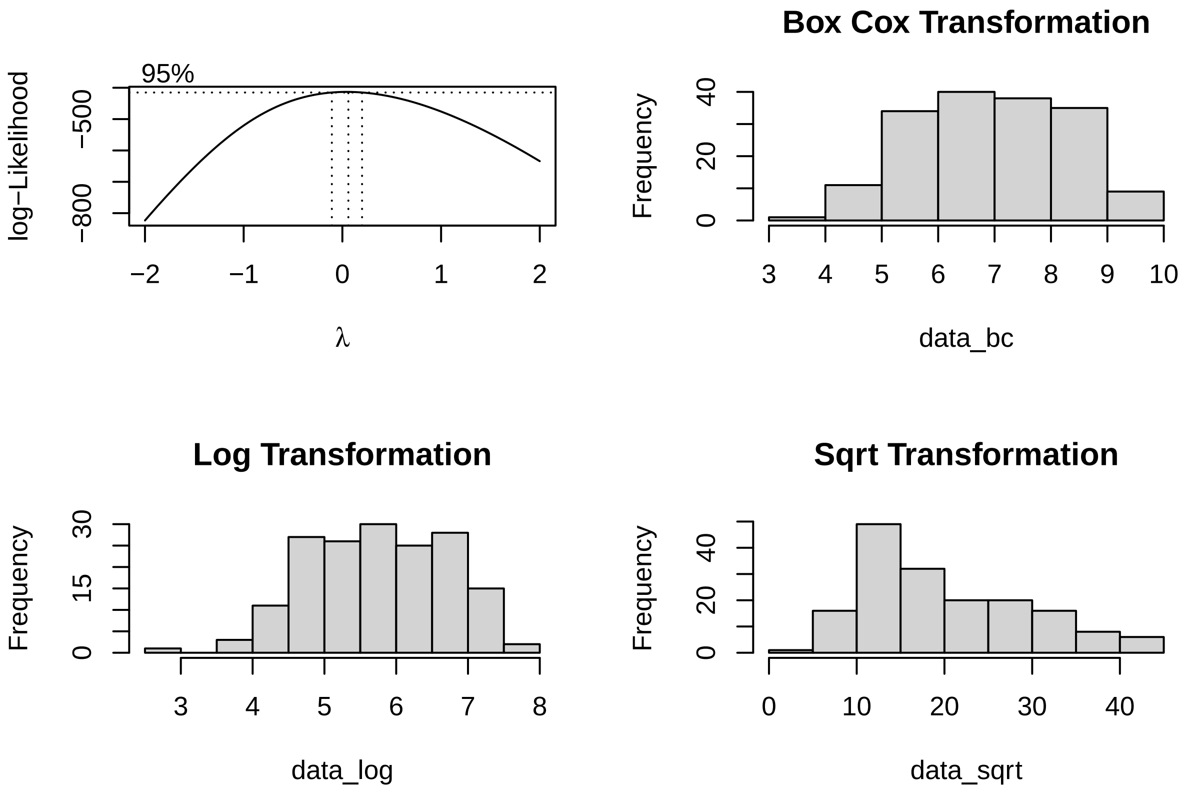
The graph appears to have some kind of seasonality and upward trend. The trend does not look large, so a simple differencing should get rid of it. To check for seasonality we will investigate the histogram of the data more.



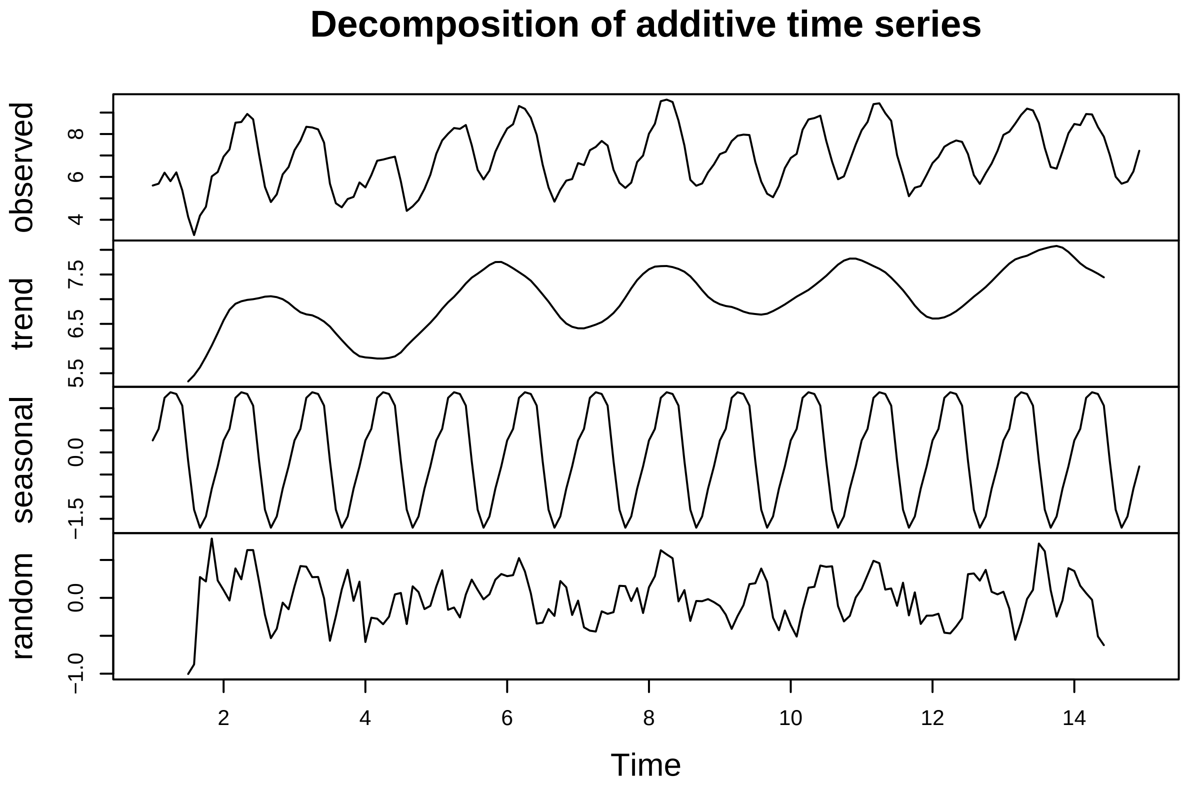
Statistics	Values
Mean of 1st half	439.8111
Mean of 2nd half	540.4778
Variance of 1st half	195922.5594
Variance of 2nd half	175643.4658

Clearly the histogram is not symmetric. In addition, the means and variances of the first and second half of the data are similar, but not enough. Thus we can conclude that this data is not stationary. I will attempt to make it as stationary as possible by performing a log, square root, and box-cox transformation on the data.

Transforming data



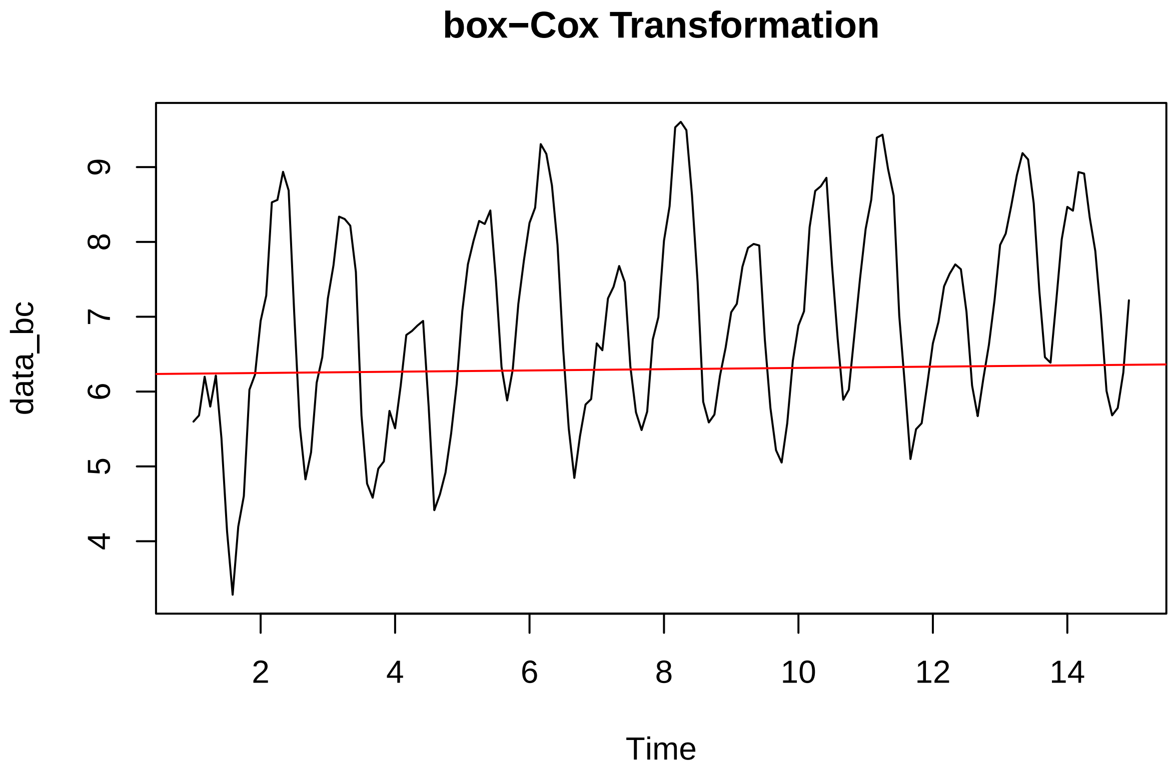
Here are the transformations I chose to use. Since $\lambda = 0.0606$ is close to 0 and $\lambda \in (0, 0.5)$, I used a log and sqrt transformation as well. Clearly the sqrt transformation didn't do as well as the Box-Cox and log transformation which was expected due to the value of λ. I chose to use the Box-Cox Transformation as it looks the most stable / symmetrical. Now to check for a trend and seasonality, we look at the decomposition graph of the Box-Cox transformed data

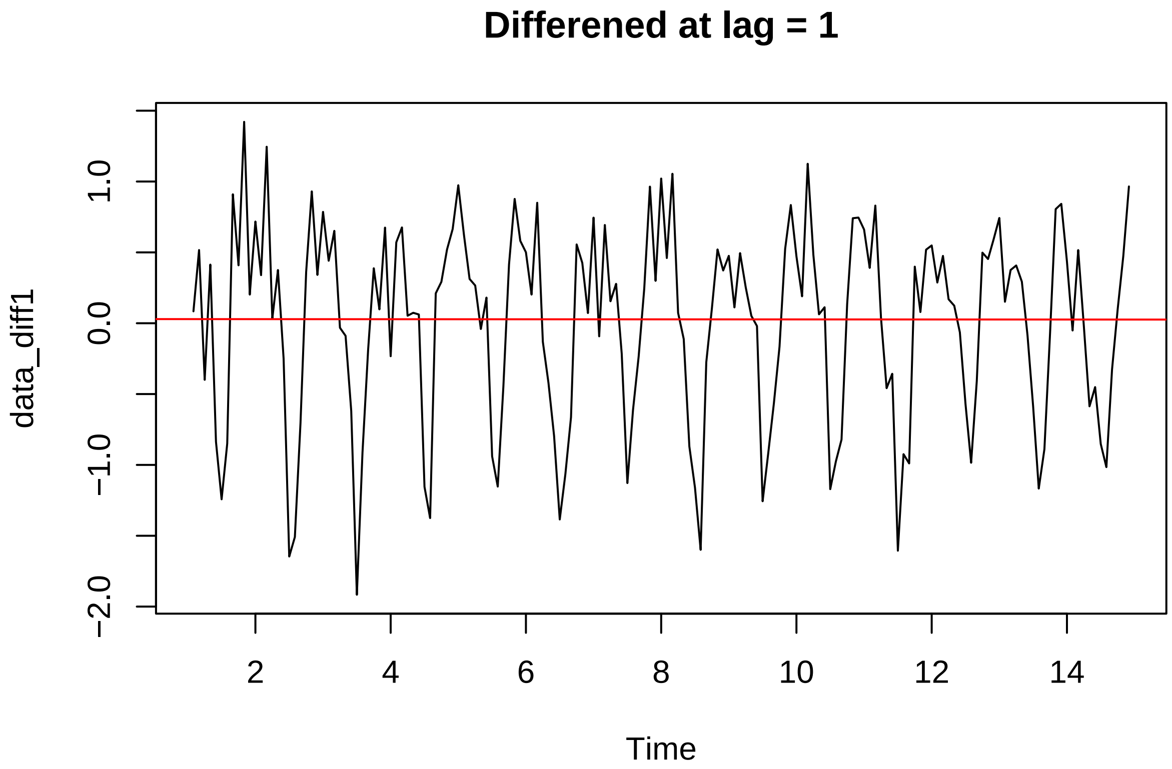


From this graph we can confirm there is some kind of trend and seasonality which we will need to get rid of to get our graph to be stationary. To do this I will difference at lag = 1 first de-trend and then difference at lag = 12 to deseasonalize it.

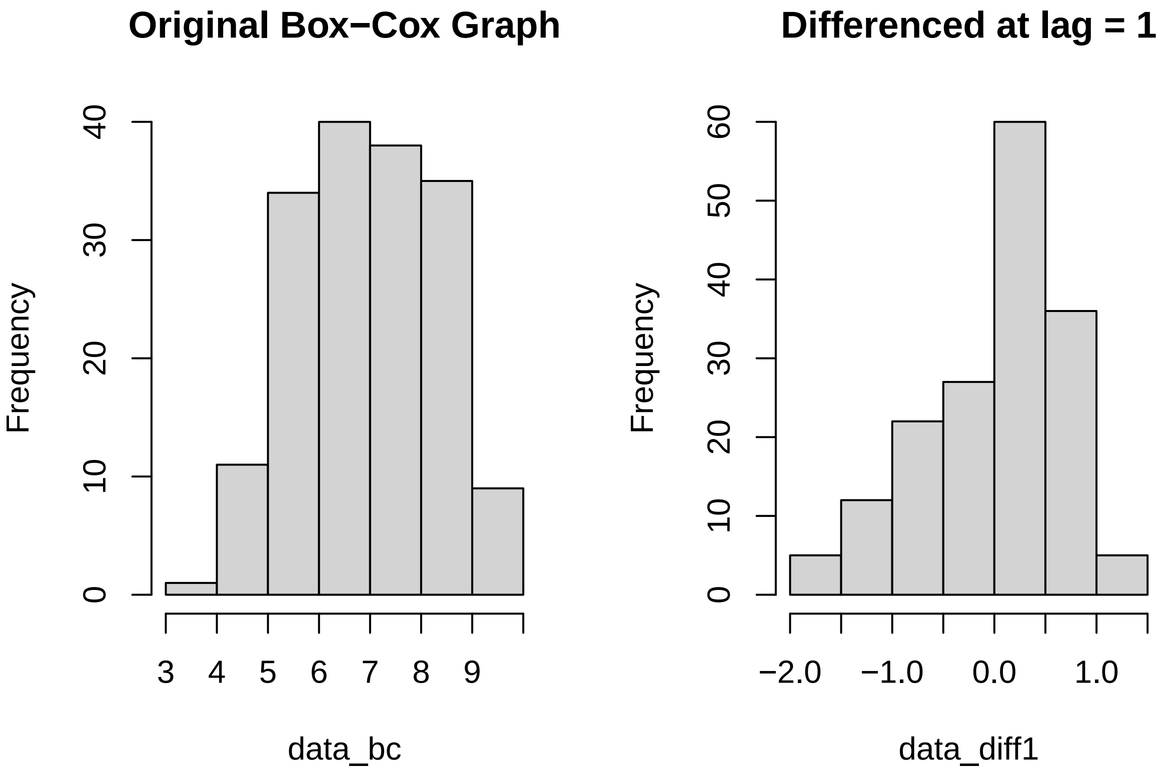
Detrend

To de-trend the data I will difference at lag = 1. Comparing the differenced graph to our original graph, we can clearly see that there was some effect. The data also appears more stationary and the trend appears less noticeable.


box−Cox Transformation
data_bc
4
5
6
7
8
9
2
4
6
8
10
12
14
Time



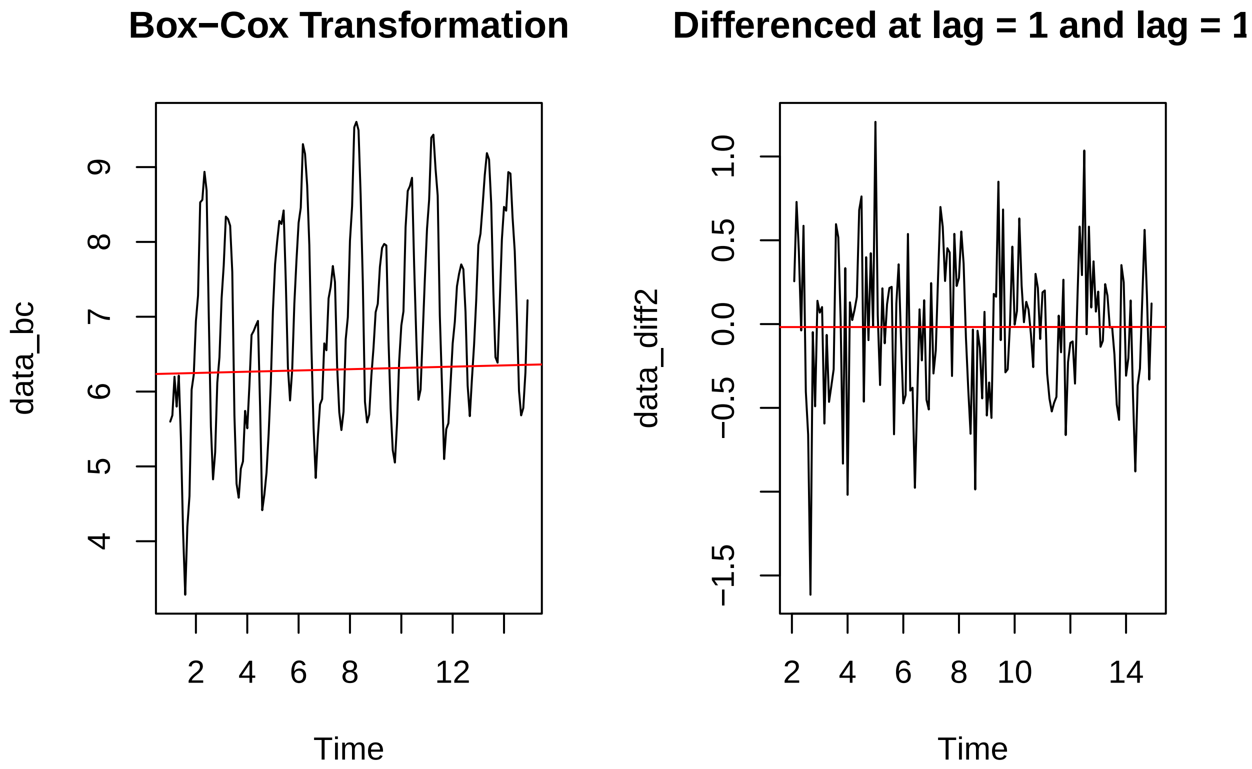
Furthermore, when comparing the histograms, we can see that it appears less stable when differenced at lag = 1. However, we will be further differencing to remove seasonality. The second differencing should provide a more stable graph. Moreover, the means and variances of the first and second half of the data are smaller, but farther apart in terms of magnitude.



Statistics	Original_Values	Differenced_Values
Mean of 1st half	6.553374	0.0168340
Mean of 2nd half	7.339013	0.0026755
Variance of 1st half	1.825975	0.5161351
Variance of 2nd half	1.522146	0.4335039

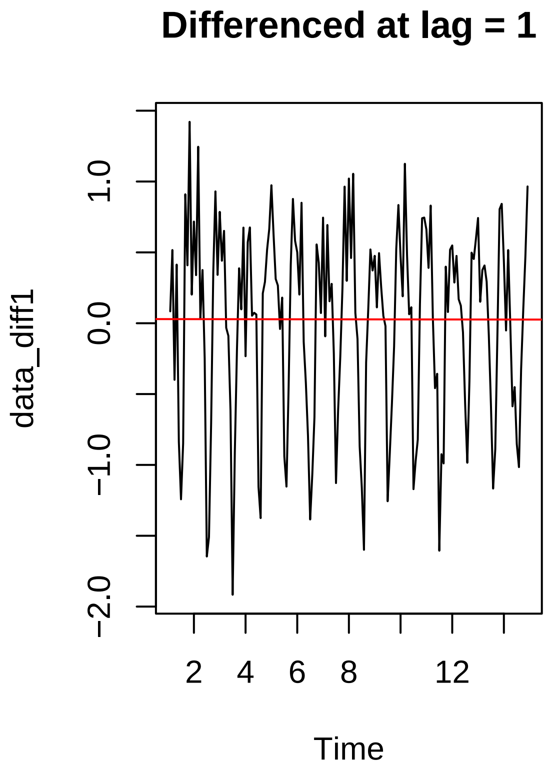
Deseasonalize

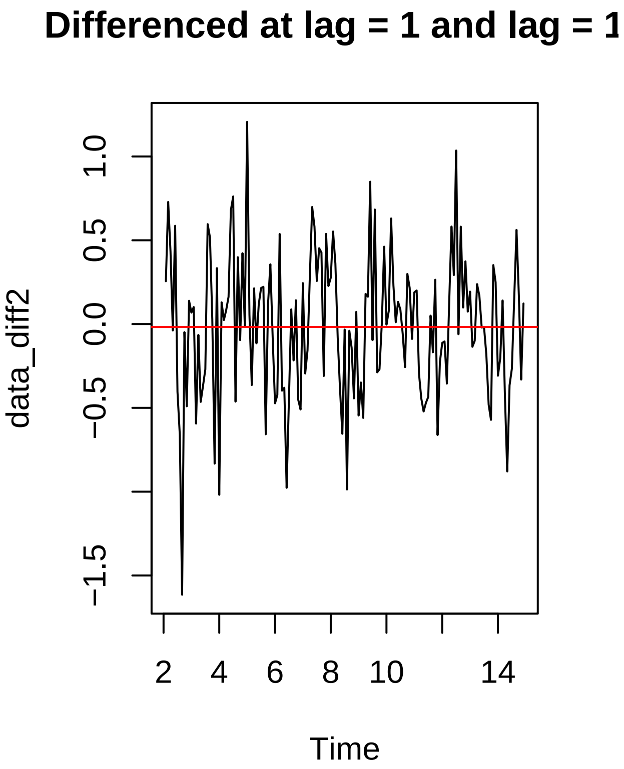
To remove the seasonality I will difference at lag = 12 since the data we have is monthly. The graphs below show that the data is further stationary when differencing at both lag = 1 and 12 compared to the original Box-Cox transformed data and just differencing at lag = 1.



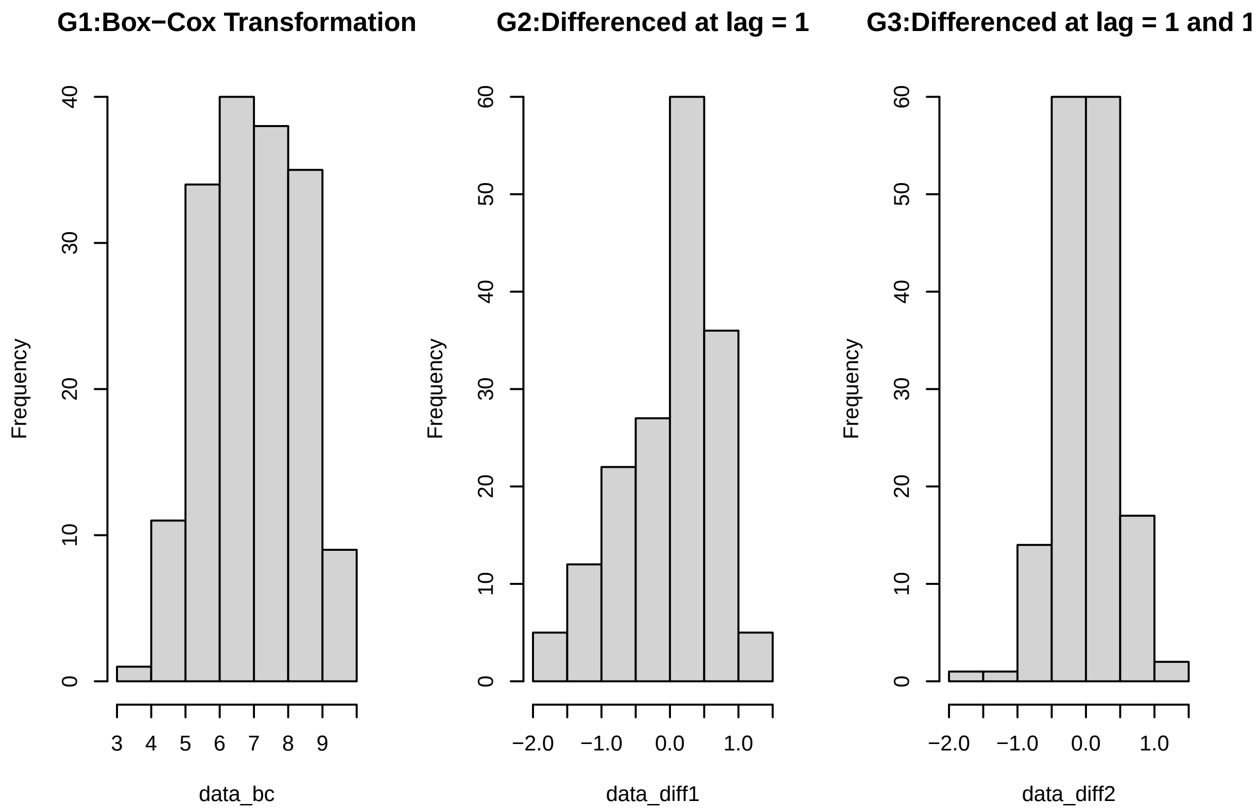
Box−Cox Transformation

Differenced at lag = 1 and lag = 1





Looking at our histograms we can see that the data is also further symmetric. While the mean was roughly the same, the variance became smaller.

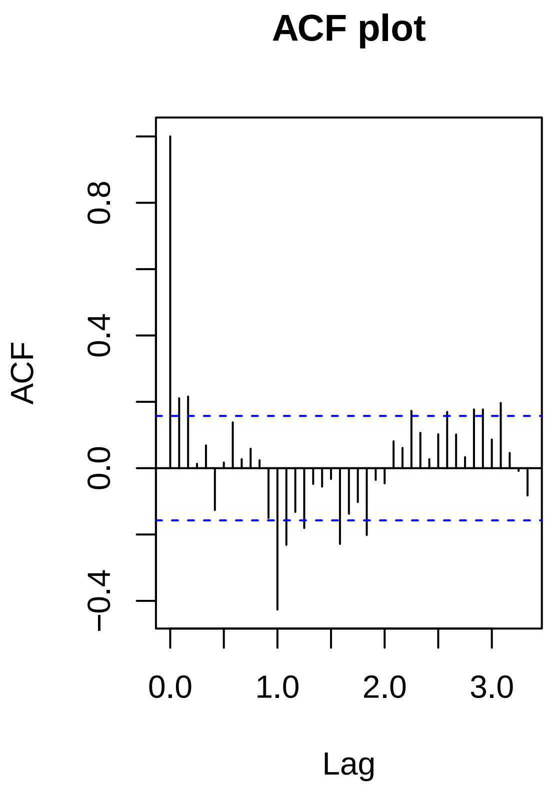


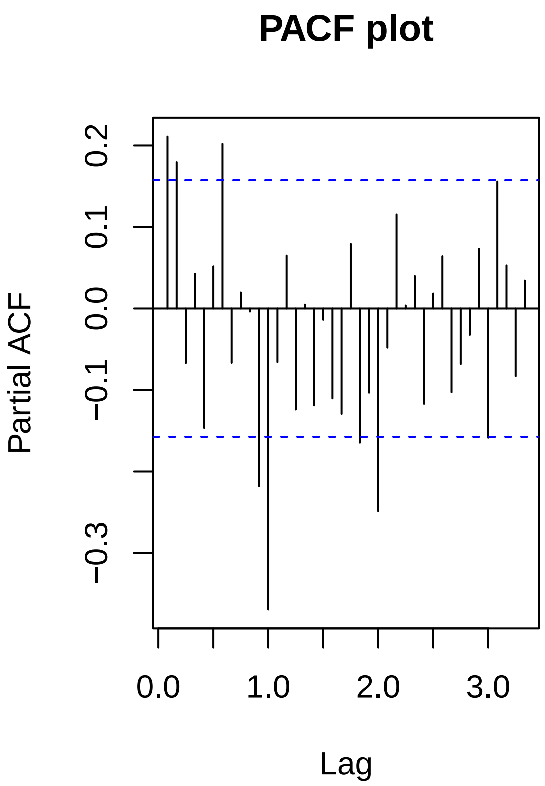
Statistics	G1_Values	G2_Values	G3_Values
Mean of 1st half	6.553374	0.0168340	0.0003589
Mean of 2nd half	7.339013	0.0026755	-0.0263325
Variance of 1st half	1.825975	0.5161351	0.2336146
Variance of 2nd half	1.522146	0.4335039	0.1432629

Because our histogram when differencing twice is more stable / symmetric than the original and when differenced once, we will use this model going forward.

Analysis of ACF and PACF Plots To Get Preliminary Models

Looking at the ACF and PACF of the data we can see that there are a couple possible models that come from them.





For model assumptions we know that we differenced once at lag = 1 to remove trend and once at lag = 12 to remove seasonality. Also we have monthly data. Therefore our d = 1, D = 1 and S = 12. We can conclude our P = 0 and Q = 2 from the plots as well. A possible model candidate is the Pure MA(2) model as our ACF lags cutoff after lag = 1, so p = 1. When looking at the PACF graph, we can see that at lags = 1,2,3 it is outside the confidence interval. However, it is also decreasing. Thus we could have q = 1 or 3. Therefore our possible model candidates are Pure MA(2), $\text{SARIMA}(1, 1, 2) \times (0, 1, 2)_{12}$, and $\text{SARIMA}(3, 1, 2) \times (0, 1, 2)_{12}$.

To select the best model we compare the AICc for each model.

Models	AICc
Pure MA(2)	173.0803
SARIMA(1,1,2) x (0,1,2) S = 12	165.6217
SARIMA(3,1,2) x (0,1,2) S = 12	161.0201

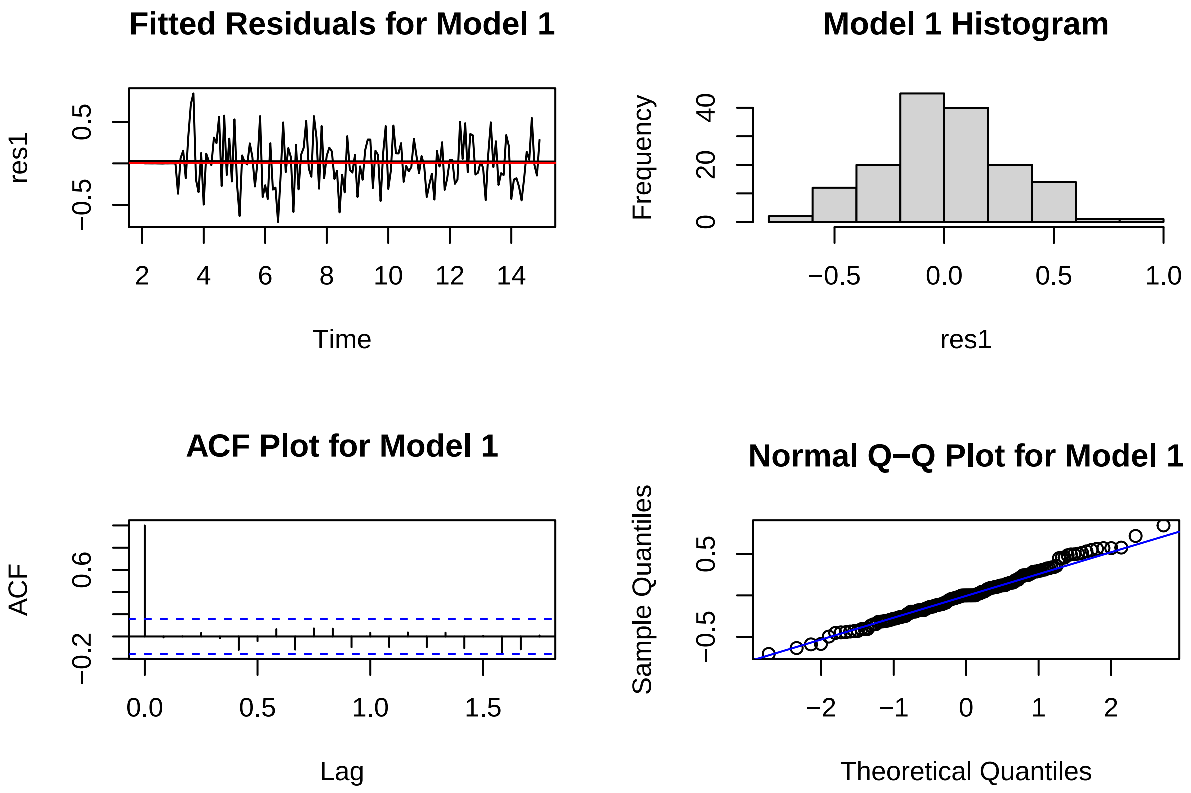
From our table we can see that our $\text{SARIMA}(3, 1, 2) \times (0, 1, 2)_{12}$ model is the best, due to having the lowest AICc value, and our $\text{SARIMA}(1, 1, 2) \times (0, 1, 2)_{12}$ the second best. We will use and compare both these models so now we need to perform model diagnostics on the residuals. We will set Model 1 = $\text{SARIMA}(3, 1, 2) \times (0, 1, 2)_{12}$ and Model 2 = $\text{SARIMA}(1, 1, 2) \times (0, 1, 2)_{12}$

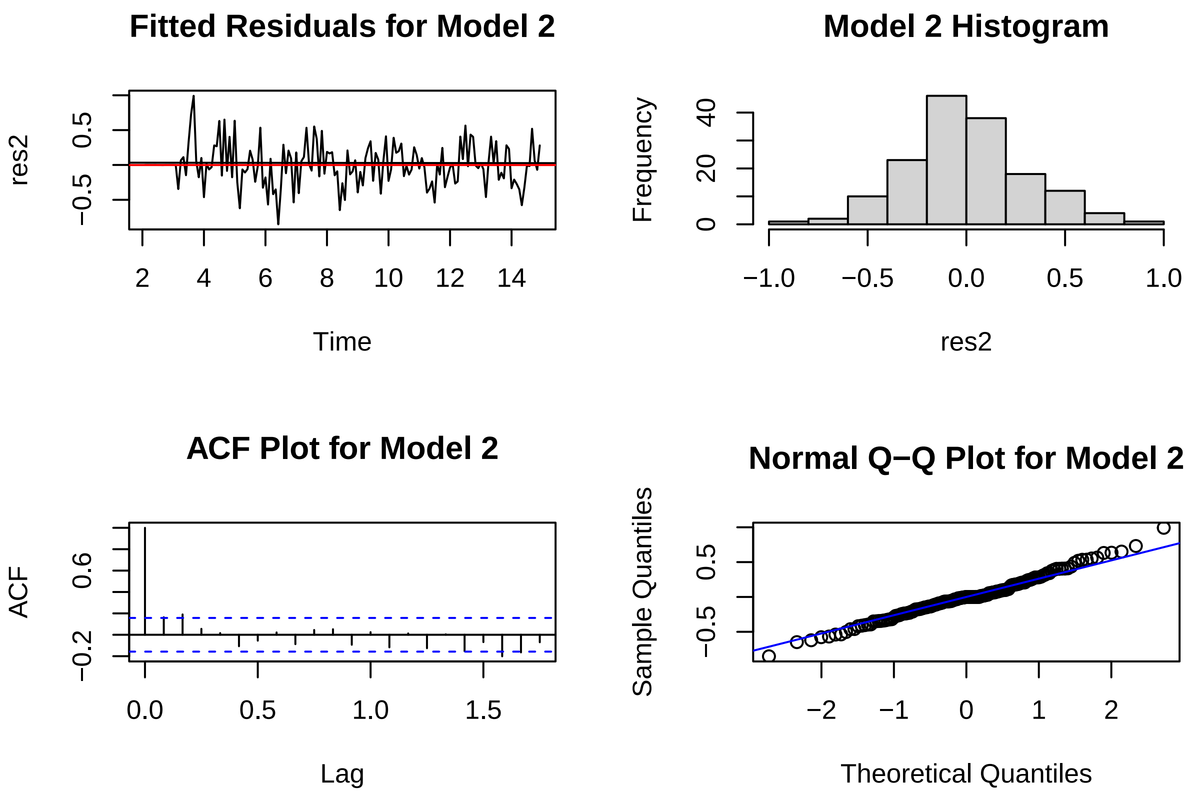
Diagnostic Checking

Our diagnostic checks will include:

(i) Plotting Residuals to see if they resemble White Noise

(ii) Plotting Histogram of the residuals to see if they resemble Gaussian
(iii) Examining Normal Q-Q Plot
(iv) Running a Shapiro-Wilk test of normality
(v) Checking sample ACF
(vi) Yule-Walker test





Tests	Model_1_Values	Model_2_Values
Shapiro-Wilk Normality Test	0.5377	0.08360
Box-Pierce Test	0.5155	0.07840
Box-Ljung test	0.4517	0.06119
Box-Ljung test with data^2	0.3322	0.40330
Yule-Walker Selected Order	0.0000	2.00000

For our first model we can see that the residuals appear to be White Noise, are normal, and are within the confidence interval on the ACF graph. We can also see that for all of our Box-Piece and Ljung-Box tests that p $>$ 0.05. Also our Yule-Walker selected order is 0. Therefore the residuals of Model 1 pass all diagnostic checks and we can continue to use it for forecasting / predicting. In addition, the model is stationary, but not invertible as $\Phi 2 = 1$

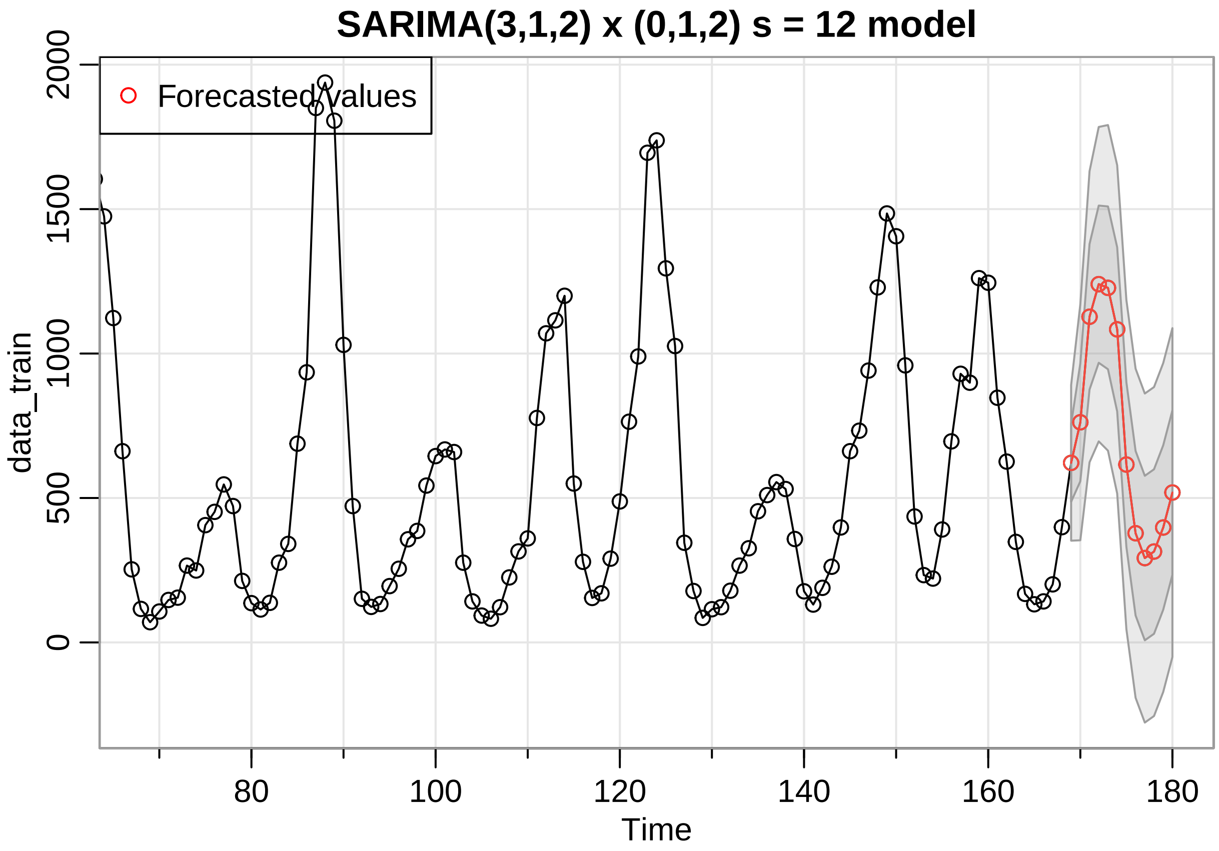
For our second model, it is similar to our first model. All the residuals appear to be White Noise, normal and are within the confidence interval for the ACF graph. All the Box-Pierce and Ljung-Box tests passed with p $>$ 0.05. However, our Yule-Walker selected order is 2, not 0. Therefore Model 2 does not pass all diagnostic checks and we can't use it for forecasting.

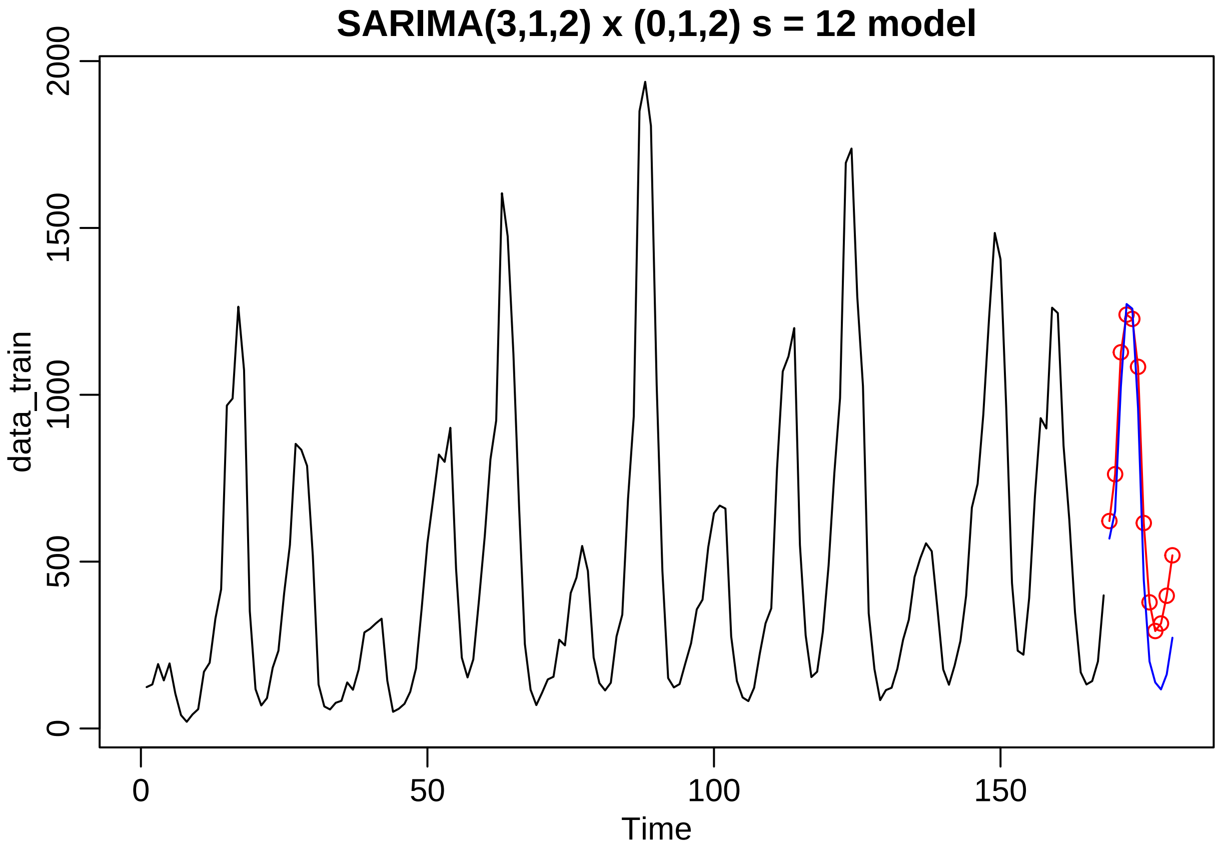
Therefore our final model is Model 1 with the equation $(1+0.781B-0.3158B^2-0.1477B^3)(1-B^{12})(1-B)X_t = (1-0.151B-0.849B^2)(1-1.9133B^{12}+B^{24})Z_t$

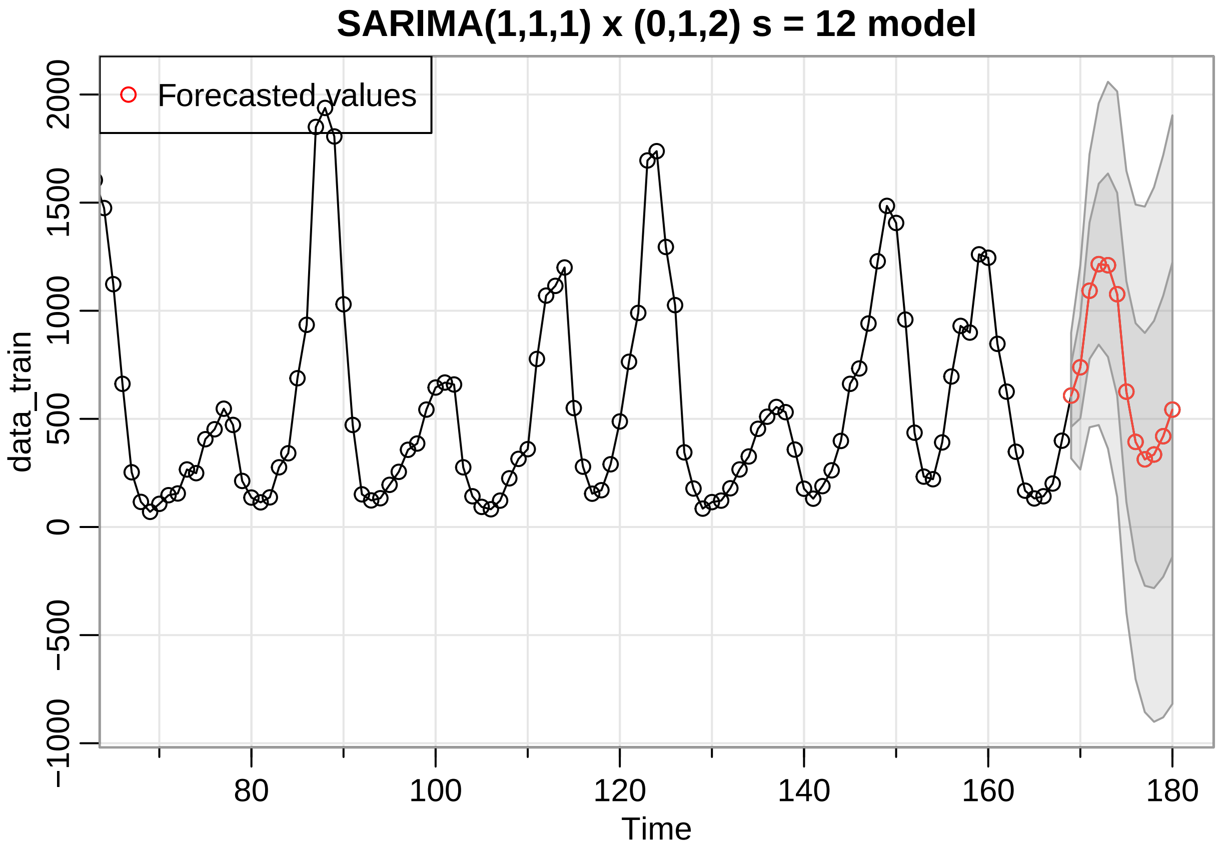
We will still use both models to forecast / predict values even though our second model is not acceptable just to compare it against our first model.

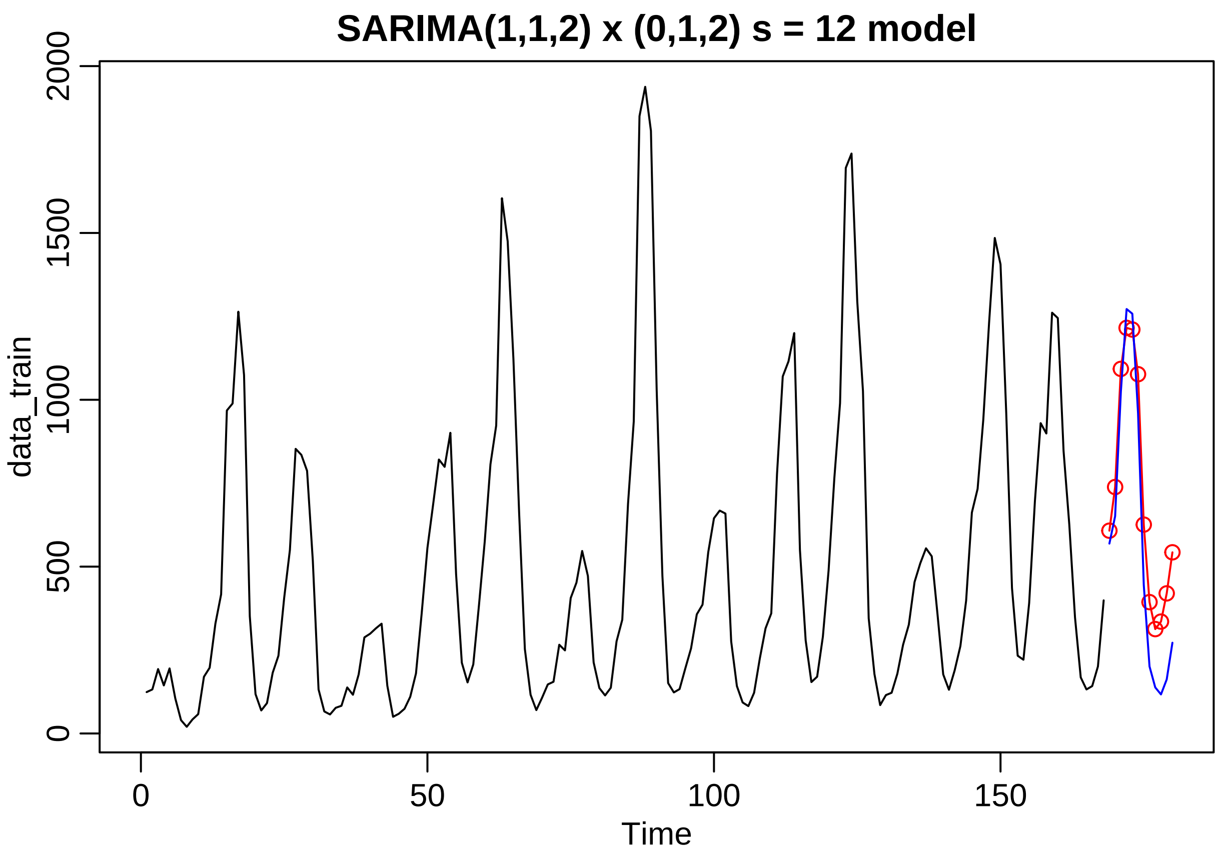
Forecasting values

We will be forecasting / predicting the next 12 values as shown by the graphs for each model.




SARIMA(3,1,2) x (0,1,2) s = 12 model
data_train
0
500
1000
1500
2000
0
50
100
150
Time


SARIMA(1,1,1) x (0,1,2) s = 12 model
○ Forecasted values
data_train
2000
1500
1000
500
0
−500
−1000
80
100
120
140
160
180
Time



We can see that they both follow the same trend as the actual values. However, our second model's values are much higher than our first models. Therefore we can see that although our second model almost passed all diagnostic checks, since the residuals weren't White Noise, we have an inaccurate, but close, model. Our first model fits the true data very well, but there are still some incorrect forecasts. From observation it appears that half the values were predicted correctly and half incorrectly. This could be fixed by having a better fitting model or by differencing at other lags.

Conclusion

Overall I believe that I achieved what I wanted to, which was being able to forecast the data. I was able to take a non-stationary graph, make it stationary, find an appropriate model that passed all diagnostic checks, and closely predicted 12 values. The errors can be explained and thus I believe the model which is a SARIMA$(3, 1, 2) \times (0, 1, 2)_{12}$ is satisfactory. I want to acknowledge my good friend Eric. He ended up dropping the class in week 2 or 3, but throughout the quarter he has continuously tried to help me despite not knowing anything about this class.

References

Lab 6 and 7 for general project template Lab 6 for creating SARIMA models in R Lab 5 for constructing SARIMA model equations Lab 4 for differencing Lecture 11 slides for diagnostic checking Week 4 lecture slides for SARIMA models All other lectures for basic foundations https://medium.com/@ooemma83/how-to-interpret-acf-and-pacf-plots-for-identifying-ar-ma-arma-or-arima-models-498717e815b6 https://www.baeldung.com/cs/acf-pacf-plots-arma-modeling https://towardsdatascience.com/interpreting-acf-and-pacf-plots-for-time-series-forecasting-af0d6db4061c https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/decompose

Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(tidymodels)
library(tsdl)
library(tidyverse)
library(forecast)
library(ggplot2)
library(MASS)
library(astsa)
library(MuMIn)
library(forecast)
library(knitr)
tidymodels_prefer()

#Subset all health datasets with frequency 12 from the tsdl library
tsdl_h <- subset(tsdl,12,'Health')

#Take 15 years of data
data <- ts(tsdl_h[[6]], start = c(1953,1), end = c(1967,12), frequency = 12)

#Split data into training and testing
data_train <- data[1:168]
data_test <- data[169:180]


#Plot data and put a trend line
ts.plot(data_train)
fit <- lm(data_train ~ as.numeric(1:length(data_train)))
abline(fit, col="red")

hist(data_train, main = 'Histogram of training data')

#Create Table showing summary stats of the data

stats <- c('Mean of 1st half', 'Mean of 2nd half', 'Variance of 1st half', 'Variance of 2nd half')
values <- c(mean(data[1:90]),mean(data[91:180]), var(data[1:90]), var(data[91:180]))

df <- data.frame(Statistics = stats, Values = values)
kable(df)

#Log transformation
data_log <- log(data_train)

#Sqrt transformation
data_sqrt <- sqrt(data_train)

par(mfrow = c(2,2))

#Boxcox transformation
#Code based off of Lab 4
boxcox_transformation <- boxcox(data_train ~ as.numeric(1:length(data_train)))
```

```r
lambda <- boxcox_transformation$x[which(boxcox_transformation$y == max(boxcox_transformation$y))]
data_bc <- (1/lambda) * (data_train^lambda-1) #lambda = 0.10101010....

#Plotting all histograms and log-likelihood curve
hist(data_bc, main = 'Box Cox Transformation')

hist(data_log, main = 'Log Transformation')
hist(data_sqrt, main = 'Sqrt Transformation')

data_bc <- ts(as.ts(data_bc), frequency = 12)
plot(decompose(data_bc))

#difference at lag 1 to remove trend
data_diff1 <- diff(data_bc, lag = 1)

#Comparison plots
plot.ts(data_bc, main = 'box-Cox Transformation')
fit <- lm(data_bc ~ as.numeric(1:length(data_bc)))
abline(fit, col="red")

plot.ts(data_diff1, main = 'Differened at lag = 1')
fit <- lm(data_diff1 ~ as.numeric(1:length(data_diff1)))
abline(fit, col="red")

par(mfrow = c(1,2))
hist(data_bc, main = 'Original Box-Cox Graph')

hist(data_diff1, main = 'Differenced at lag = 1')

#Creating table to display

values1 <- c(mean(data_bc[1:84]), mean(data_bc[85:168]), var(data_bc[1:84]), var(data_bc[85:168]))
values2 <- c(mean(data_diff1[1:83]), mean(data_diff1[84:167]), var(data_diff1[1:83]),var(data_diff1[84:

df <- data.frame(Statistics = stats, Original_Values = values1, Differenced_Values = values2)
kable(df)

#Difference at lag = 12
data_diff2 <- diff(data_diff1, lag = 12)

#Comparison plots between boxcox and differencing at lag 1 and 12
par(mfrow = c(1,2))
ts.plot(data_bc, main = 'Box-Cox Transformation')
fit <- lm(data_bc ~ as.numeric(1:length(data_bc)))
abline(fit, col="red")

ts.plot(data_diff2, main = 'Differenced at lag = 1 and lag = 12')
fit <- lm(data_diff2 ~ as.numeric(1:length(data_diff2)))
abline(fit, col="red")

#Comparison plots between differencing at lag 1 and differencing at lags 1 and 12
par(mfrow = c(1,2))
ts.plot(data_diff1, main = 'Differenced at lag = 1')
```

```r
fit <- lm(data_diff1 ~ as.numeric(1:length(data_diff1)))
abline(fit, col="red")

ts.plot(data_diff2, main = 'Differenced at lag = 1 and lag = 12')
fit <- lm(data_diff2 ~ as.numeric(1:length(data_diff2)))
abline(fit, col="red")

#Plot each histogram side by side
par(mfrow = c(1,3))
hist(data_bc, main = 'G1:Box-Cox Transformation')
hist(data_diff1, main = 'G2:Differenced at lag = 1')
hist(data_diff2, main = 'G3:Differenced at lag = 1 and 12')

#Creating table comparing stats between all models (original, differencing once, differencing twice)
values3 <- c(mean(data_diff2[1:72]), mean(data_diff2[73:155]), var(data_diff2[1:72]), var(data_diff2[73

df <- data.frame(Statistics = stats, G1_Values = values1, G2_Values = values2, G3_Values = values3)
kable(df)

par(mfrow = c(1,2))

acf(data_diff2, lag.max = 40, main = 'ACF plot')
pacf(data_diff2, lag.max = 40, main = 'PACF plot')

#Code from lab 6
fit_ma1 = arima(data_diff2, order = c(0, 0, 2))

AICc(fit_ma1)

fit_ma1

fit_sarima111 = arima(data_diff2, order = c(1, 1, 2),
                seasonal = list(order = c(0, 1,2),
                period = 12), method="ML")

AICc(fit_sarima111)

fit_sarima111

fit_sarima311 = arima(data_diff2, order = c(3, 1, 2),
                seasonal = list(order = c(0, 1, 2),
                period = 12), method="ML")

AICc(fit_sarima311)

fit_sarima311

res1 <- residuals(fit_sarima311)

par(mfrow = c(2,2))

plot.ts(res1, main = 'Fitted Residuals for Model 1')
t <- 1:length(res1)
```

```r
fit.res1 = lm(res1~t)
abline(fit.res1)
abline(h = mean(res1), col = 'red')

hist(res1, main = 'Model 1 Histogram')
acf(res1, main = 'ACF Plot for Model 1')

qqnorm(res1,main= "Normal Q-Q Plot for Model 1")
qqline(res1,col="blue")


res2 <- residuals(fit_sarima111)

par(mfrow = c(2,2))

plot.ts(res2, main = 'Fitted Residuals for Model 2')
t <- 1:length(res2)
fit.res2 = lm(res2~t)
abline(fit.res2)
abline(h = mean(res2), col = 'red')

hist(res2, main = 'Model 2 Histogram')
acf(res2, main = 'ACF Plot for Model 2')

qqnorm(res2,main= "Normal Q-Q Plot for Model 2")
qqline(res2,col="blue")

shapiro.test(res1)

Box.test(res1, lag = 13, type = c('Box-Pierce'), fitdf = 2)
Box.test(res1, lag = 13, type = c('Ljung-Box'), fitdf = 2)
Box.test(res1 ^ 2, lag = 13, type = c('Ljung-Box'), fitdf = 0)

ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))

shapiro.test(res2)

Box.test(res2, lag = 13, type = c('Box-Pierce'), fitdf = 2)
Box.test(res2, lag = 13, type = c('Ljung-Box'), fitdf = 2)
Box.test(res2 ^ 2, lag = 13, type = c('Ljung-Box'), fitdf = 0)

ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))

#Create Table for each test performed in code chunk above

tests <- c('Shapiro-Wilk Normality Test', 'Box-Pierce Test', 'Box-Ljung test', 'Box-Ljung test with data
values4 <- c(0.5377, 0.5155, 0.4517, 0.3322, 0)
values5 <- c(0.0836,0.0784, 0.06119, 0.4033, 2)

df <- data.frame(Tests = tests, Model_1_Values = values4, Model_2_Values = values5)
kable(df)

#Model 1 plot
```

```
#Don't use par(mfrow) because it will mush the graphs when knitted making it hard to see the difference
fit_pred <- sarima.for(data_train, n.ahead=12, plot.all=F, p=3, d=1, q=2, P=0, D=1, Q=2, S=12, main = 'S
legend("topleft", pch=1, col=c("red"), legend=c("Forecasted values"))

ts.plot(data_train, xlim = c(0,180), main = 'SARIMA(3,1,1) x (0,1,2) s = 12 model')
points(169:180, fit_pred$pred, col = 'red')
lines(169:180, fit_pred$pred, col = 'red')
lines(169:180, data_test, col = 'blue')


#Model 2 plot
#Don't use par(mfrow) because it will mush the graphs when knitted making it hard to see the difference
fit_pred <- sarima.for(data_train, n.ahead=12, plot.all=F, p=1, d=1, q=2, P=0, D=1, Q=2, S=12, main = 'S
legend("topleft", pch=1, col=c("red"), legend=c("Forecasted values"))

ts.plot(data_train, xlim = c(0,180), main = 'SARIMA(1,1,1) x (0,1,2) s = 12 model')
points(169:180, fit_pred$pred, col = 'red')
lines(169:180, fit_pred$pred, col = 'red')
lines(169:180, data_test, col = 'blue')
```