Time Series Analysis Of Monthly Temperature in Aomori City

Richard (Kyaw) Paing

November 25, 2021

Abstract

This report consists of a time-series forecasting of monthly temperature in Aomori City in Japan. It divides the data into a training data of 1644 observations and attempts to predict the temperature for 24 observations (2 years). The motivation is to gain insight into the general temperature throughout the year. The data is differenced at lag 12 and 1 to remove seasonality and trend. Then it is tested by Shipiro-Wilk, Box-Ljung, Box-Pierce, and Mcleod-Li tests. The chosen model has passed all the tests except the McLeod-Li test. After testing other similar models, it is decided that SARIMA$(1,1,1)(0,1,1)_{12}$ is the best model out of the other models checked.

Introduction

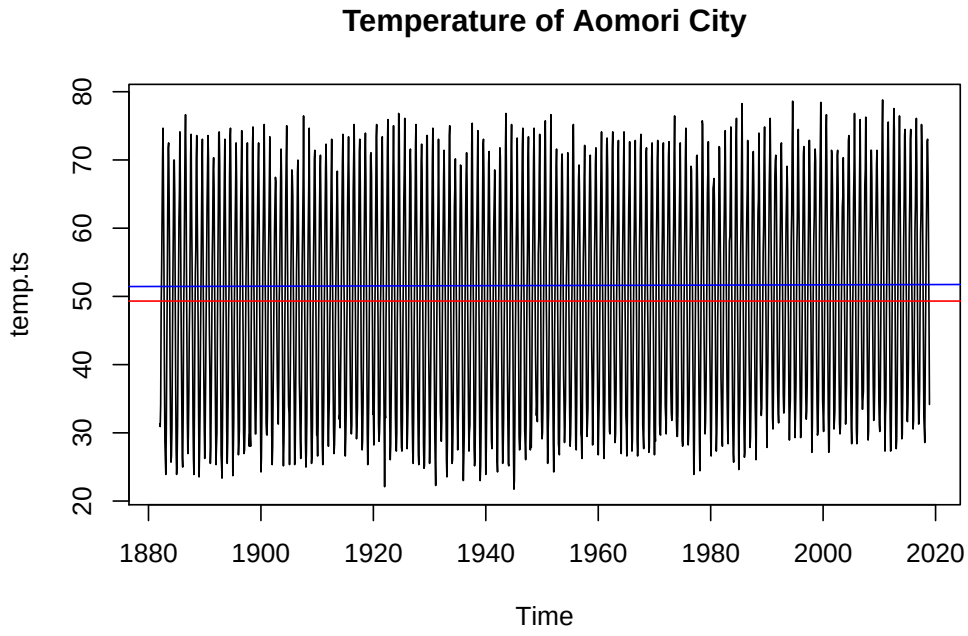
With climate variation and also climate change occuring in the world, there could be many fluctuations for the temperature within a year. This project seeks to perform a time series analysis on the monthly temperature within a year, which could be useful for planning out activities that are best performed within cetain temperatures. The data is the monthly average temperature in celcius of the city of Aomori, Japan, from 1880-2020. The data has been converted into farenheit to not include negative values and align with the standard unit of measure in the US. The software used for this analysis is R.

The data uses 1645 observations as the training data and 24 observations as the testing data. It is differenced at lags 12 and 1, however some models will not be differencing at lag 1 because the data does not seem to have a trend. The models are generated by examining the pacf and acf of the differenced data or are from examining the coefficients of the previous models.

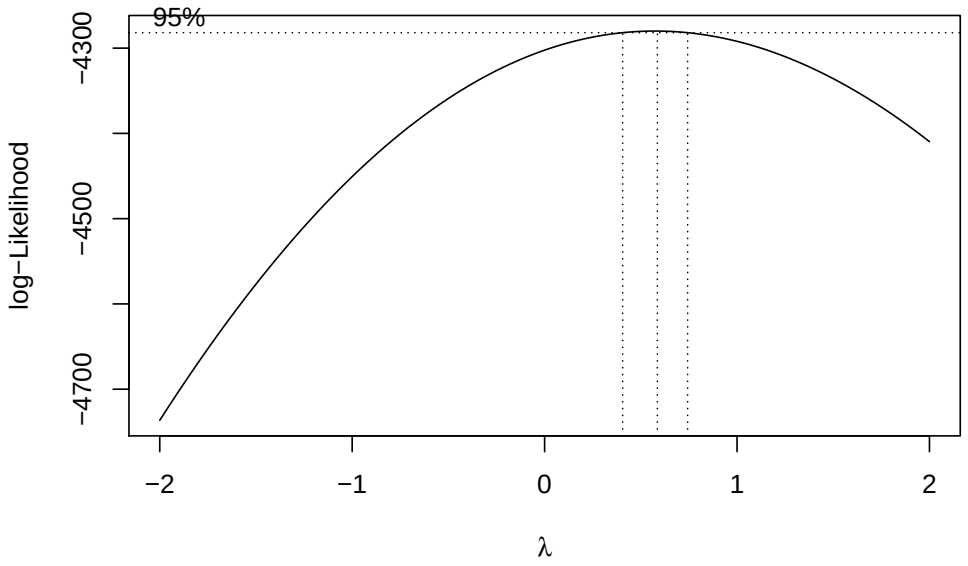
The data set can be found on kaggle.com, titled "Monthly temperature around Aomori City". The software used is R.

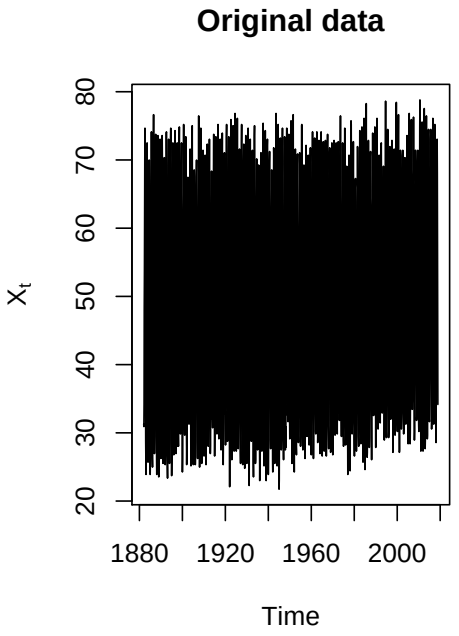
Analysis

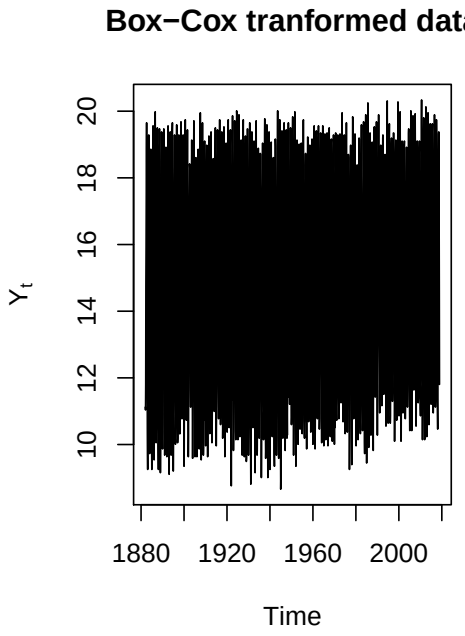
```
Warning: package 'MASS' was built under R version 4.0.4
```



The above is the original data, with the trend being the blue line and the mean being the red line. The original data has a mean of 49.31482 and a variance of 246.5621. The large variance is possibly due to the conversion from celcius to fareheit. The data does not seem to have a particular trend and looks fairly normal and stationary. We will perform a box-cox transformation to stabalize the variance.





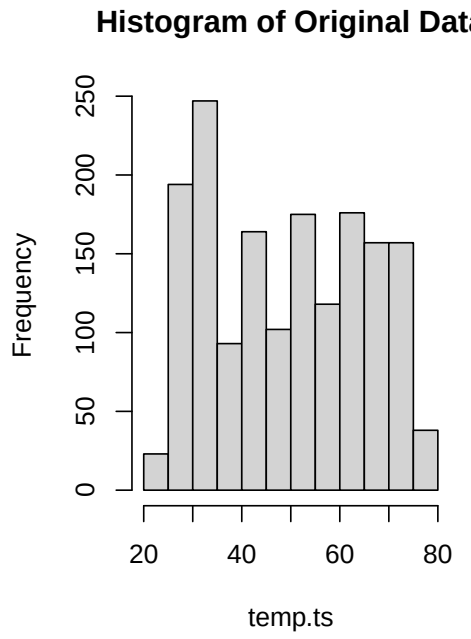


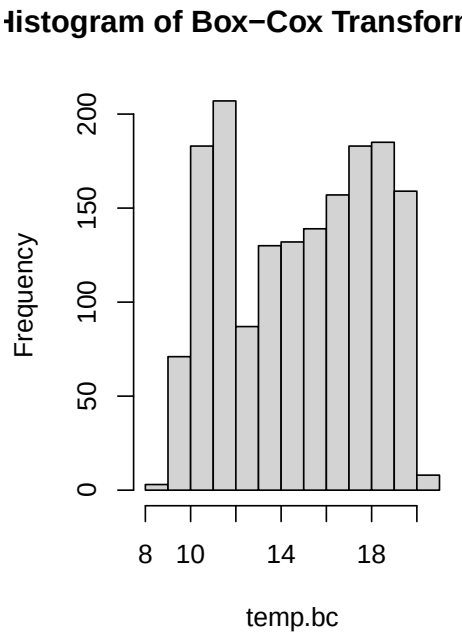
Variance of box-cox transformed data:

```
[1] 10.07401
```

Mean of box-cox transformed data;

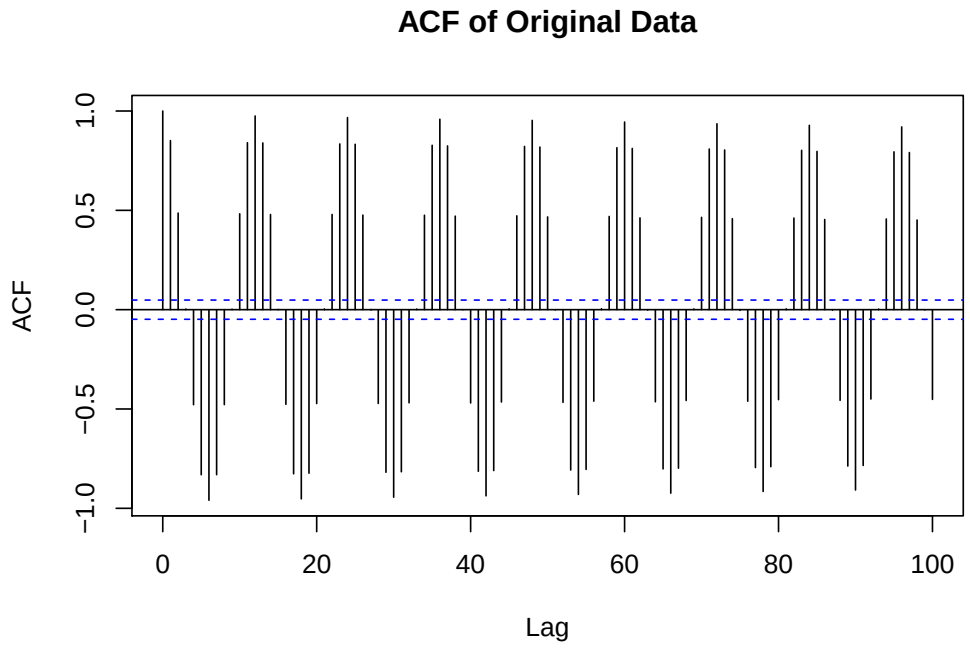
```
[1] 14.82958
```

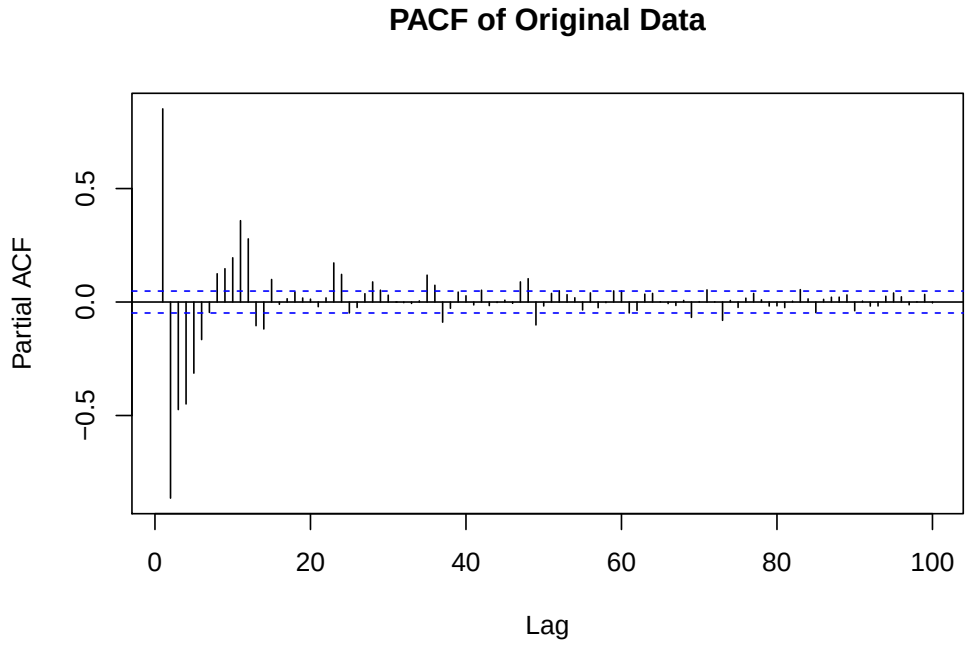




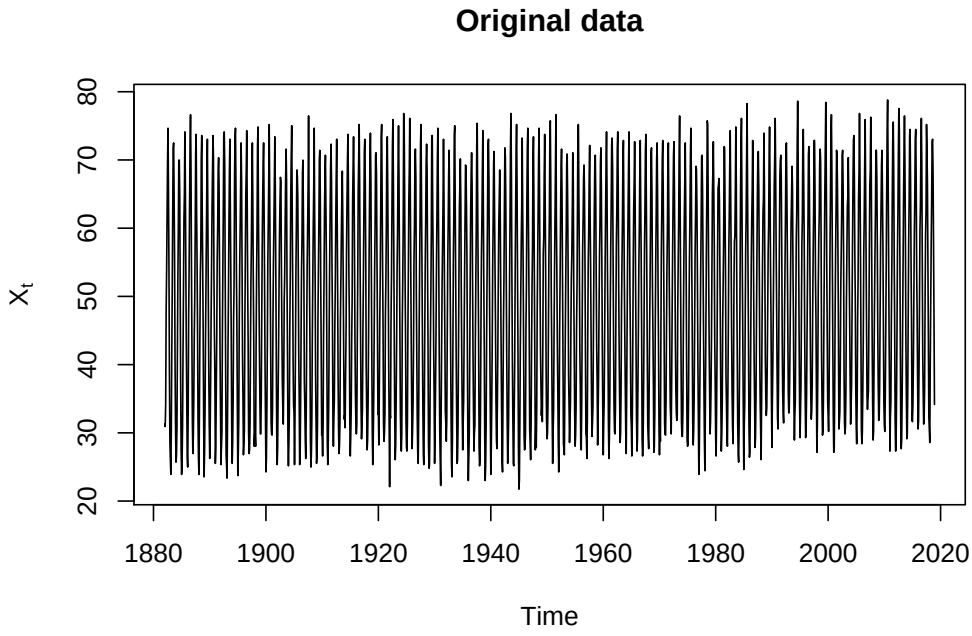
The transformation has greatly reduced the variance from the original which was

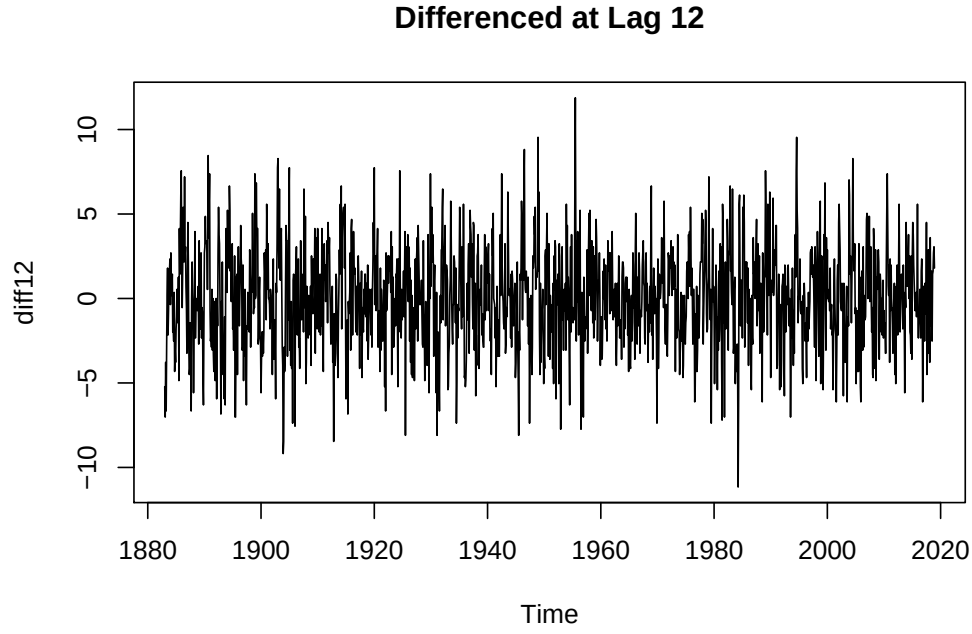
246.5621., as well as the mean which was 49.31482. However, because the original plot does not seem to have a mean or variance that increases with time and already seems stationary, the box-cox transformation will not be used.

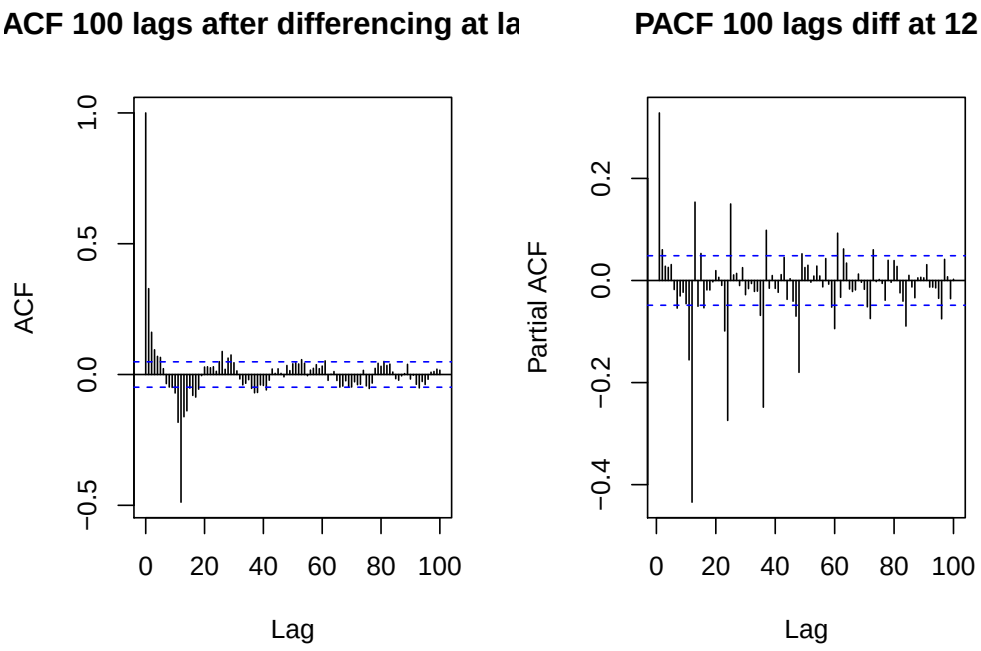




The acf and pacf show that there is seasonality, whereas the pacf indicates that lag 12 may be important.

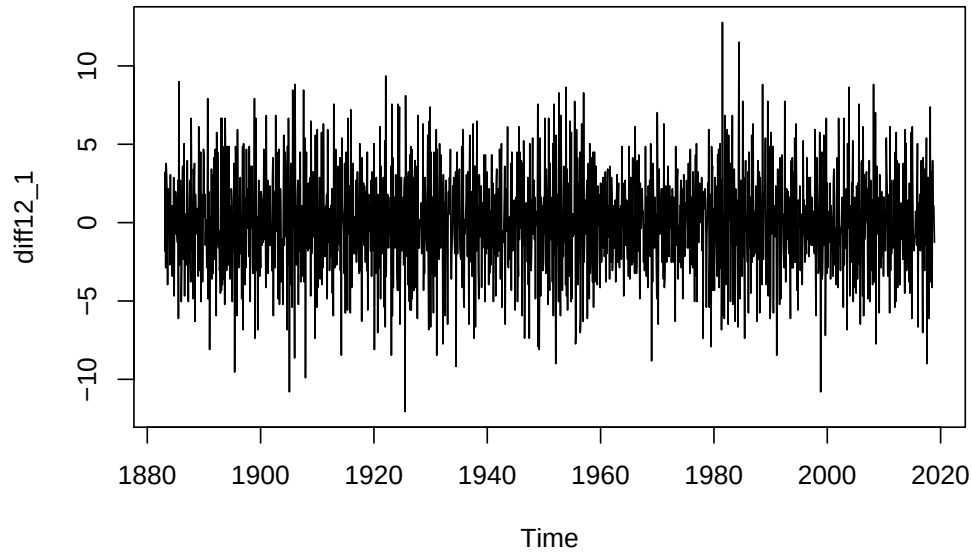

Original data
X_t
20
30
40
50
60
70
80
1880
1900
1920
1940
1960
1980
2000
2020
Time

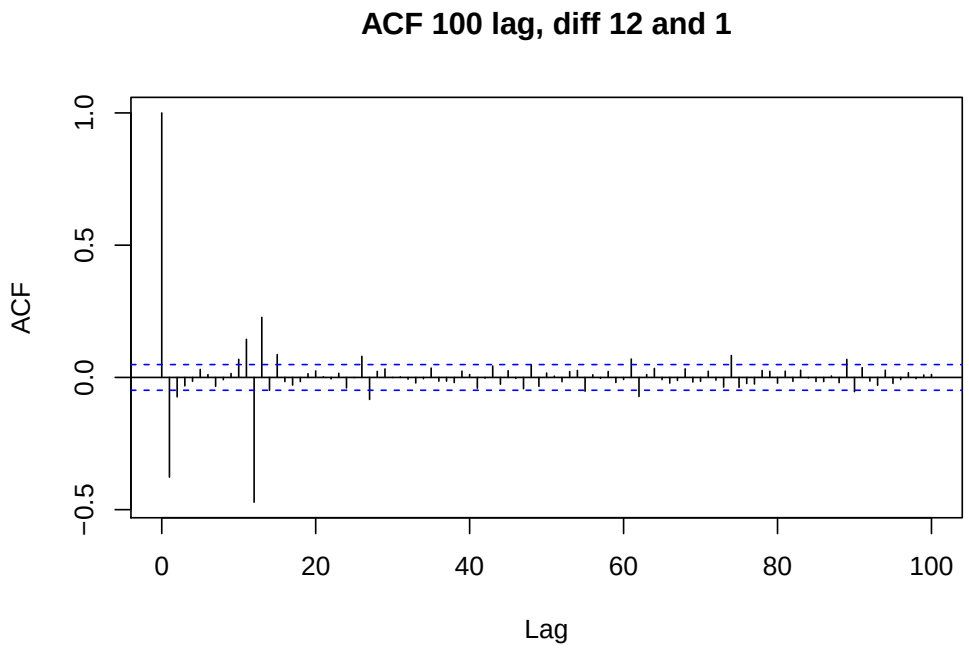

Differenced at Lag 12
diff12
−10
−5
0
5
10
1880
1900
1920
1940
1960
1980
2000
2020
Time

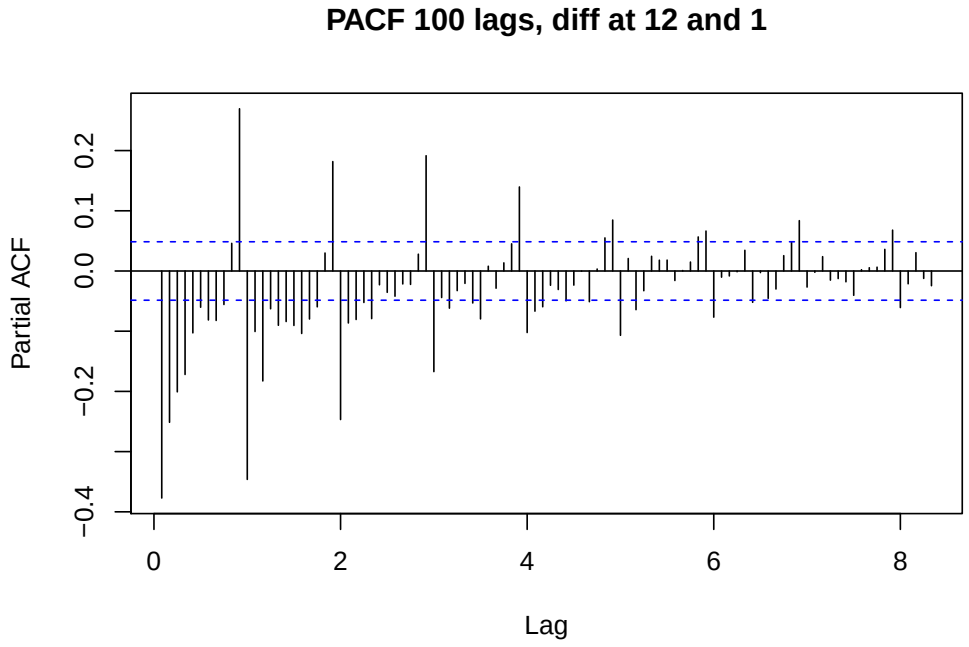



```
[1] 8.798336
```

Compared to the original data, the data looks much more stationary and like white noise. The acf seems to decay right at the beginning, but it may not be important. The pacf also seems to slowy decay. The variance is now 8.798336 compared to 246.5621 of the original data. Since the PACF seems to tail off, it may be that this is a moving average model.








```
[1] 11.80815
```

The variance is now 11.80815, which is higher than 8.798336, the variance for differencing at lag 12. Which may indicate that this differencing is uncessary. Differencing at lag 1 seemed to get rid of the decay for the acf, but not for the pacf. This may mean that the seasonal component for the AR part is 0 or it could be 3 or 6 which is unlikely. The acf was also significant at lag 1,2,10,11, and 12. The model Based on this the models chosen to test out are:

$\text{SARIMA}(2,0,2)(4,1,1)_{12}$

$\text{SARIMA}(2,0,2)(3,1,1)_{12}$

$\text{SARIMA}(1,1,1)(0,1,1)_{12}$

$\text{SARIMA}(2,1,2)(1,1,1)_{12}$

Model : $\textbf{SARIMA}(2,0,2)(4,1,1)_{12}$

```
Warning in arima(temp.ts, order = c(2, 0, 2), seasonal = list(order = c(4, :
possible convergence problem: optim gave code = 1


Call:
arima(x = temp.ts, order = c(2, 0, 2), seasonal = list(order = c(4, 1, 1), period = 12),
    method = "ML", optim.method = "Nelder-Mead")

Coefficients:
         ar1     ar2     ma1      ma2    sar1     sar2     sar3    sar4
      0.0534  0.3573  0.3169  -0.1815  0.0348  -0.0347  -0.0548  0.0404
s.e.  0.1851  0.0972  0.1927   0.0700  0.0273   0.0271   0.0269  0.0268
         sma1
      -0.9402
s.e.   0.0118

sigma^2 estimated as 4.206:  log likelihood = -3501.1,  aic = 7022.21
```

Model: $\textbf{SARIMA}(2,0,2)(3,1,1)_{12}$

```
Warning in arima(temp.ts, order = c(2, 0, 3), seasonal = list(order = c(3, :
possible convergence problem: optim gave code = 1


Call:
arima(x = temp.ts, order = c(2, 0, 3), seasonal = list(order = c(3, 1, 1), period = 12),
    method = "ML", optim.method = "Nelder-Mead")

Coefficients:

Warning in sqrt(diag(x$var.coef)): NaNs produced

         ar1     ar2     ma1      ma2     ma3    sar1     sar2     sar3
      0.1593  0.2548  0.1735  -0.1531  0.0544  0.0023  -0.0471  -0.0186
s.e.     NaN     NaN     NaN      NaN  0.0142  0.0271   0.0267   0.0267
         sma1
      -0.9338
s.e.   0.0118

sigma^2 estimated as 4.204:  log likelihood = -3500.68,  aic = 7021.36
```

Model: SARIMA$(1,1,1)(0,1,1)_{12}$

```
Call:
arima(x = temp.ts, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
    method = "ML")

Coefficients:
         ar1      ma1     sma1
      0.3378  -0.9829  -0.9764
s.e.  0.0244   0.0058   0.0079

sigma^2 estimated as 4.127:  log likelihood = -3491.67,  aic = 6991.33
```

Model: SARIMA$(2,1,2)(1,1,1)_{12}$

```
Call:
arima(x = temp.ts, order = c(2, 1, 2), seasonal = list(order = c(1, 1, 1), period = 12),
    method = "ML")

Coefficients:
         ar1      ar2      ma1     ma2    sar1     sma1
      0.6452  -0.0592  -1.3076  0.3167  0.0172  -0.9764
s.e.  0.2476   0.0905   0.2460  0.2422  0.0259   0.0082

sigma^2 estimated as 4.116:  log likelihood = -3489.42,  aic = 6992.85

Warning in arima(temp.ts, order = c(2, 1, 2), seasonal = list(order = c(1, :
some AR parameters were fixed: setting transform.pars = FALSE

Warning in log(s2): NaNs produced


Call:
arima(x = temp.ts, order = c(2, 1, 2), seasonal = list(order = c(1, 1, 1), period = 12),
    fixed = c(NA, 0, NA, 0, 0, NA), method = "ML")

Coefficients:
         ar1  ar2      ma1  ma2  sar1     sma1
      0.3378    0  -1.0174    0     0  -1.0241
s.e.  0.0244    0   0.0060    0     0   0.0083

sigma^2 estimated as 3.802:  log likelihood = -3491.67,  aic = 6991.33
```

Due to some of the standard deviations being NaN for SARIMA$(2,0,2)(3,1,1)_{12}$, this model may not be suitable. Out of the 4 models, two with the lowest AIC's will be chosen.

Invertiblility and Stationarity

SARIMA$(1,1,1)(0,1,1)_{12}$ and SARIMA$(2,1,2)(1,1,1)_{12}$ with some coefficients fixed to 0.

(A) $(1-0.3378B)\nabla_{12}\nabla_1 U_t = (1-0.9829B)(1-0.9764B^{12})Z_t$

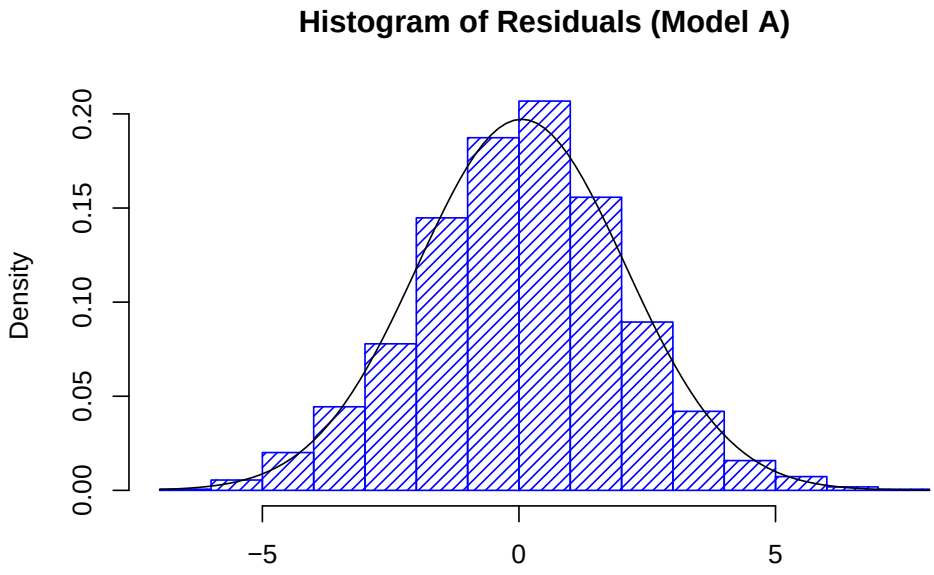
The roots of the autoregressive part is 2.960332 and the roots of the moving average part is 1.017397. Since both of the roots are outside of the unit circle, it is concluded that this model is both stationary and invertible.

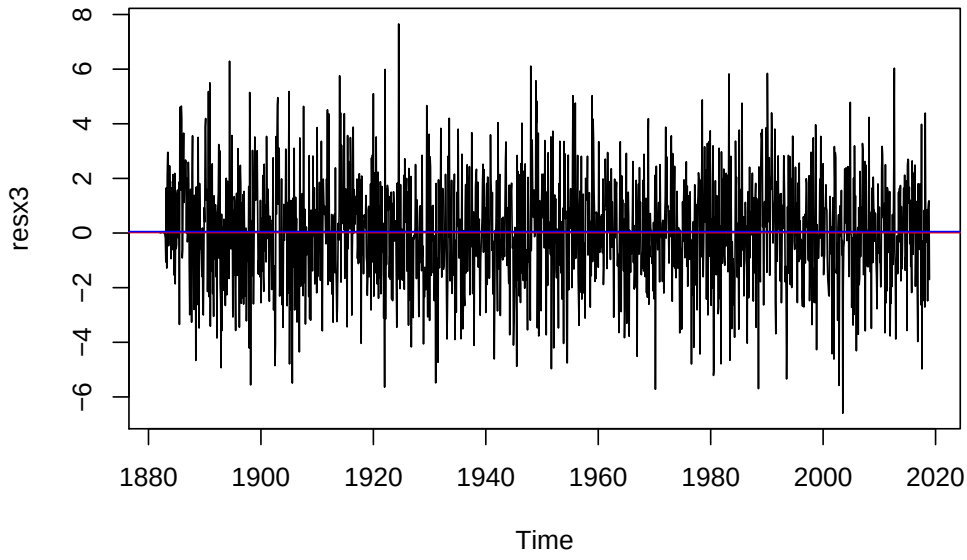
(B) $(1-0.3378B)\nabla_{12}\nabla_1 U_t = (1-1.0174B)(1-1.0241B^{12})Z_t$

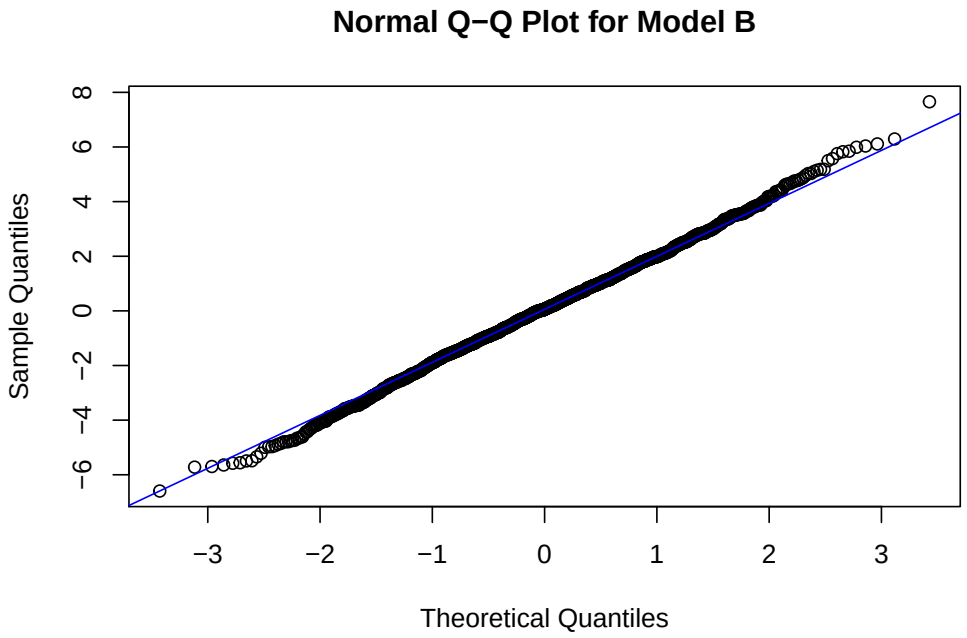
The roots of the autoregressive part is the same as the previous model, 2.960332. The roots of the moving average part is 0.9828976, which is inside the unit circle. Therefore, this model is not invertible but it is stationary.

Diagnostic Checking for Model A

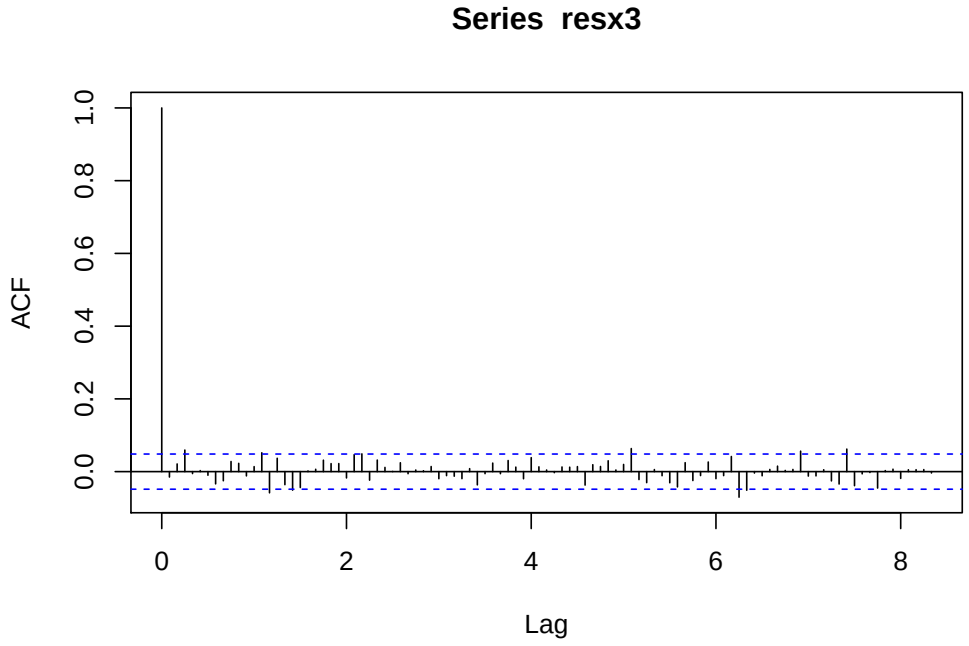
```
[1] 0.05221619
```

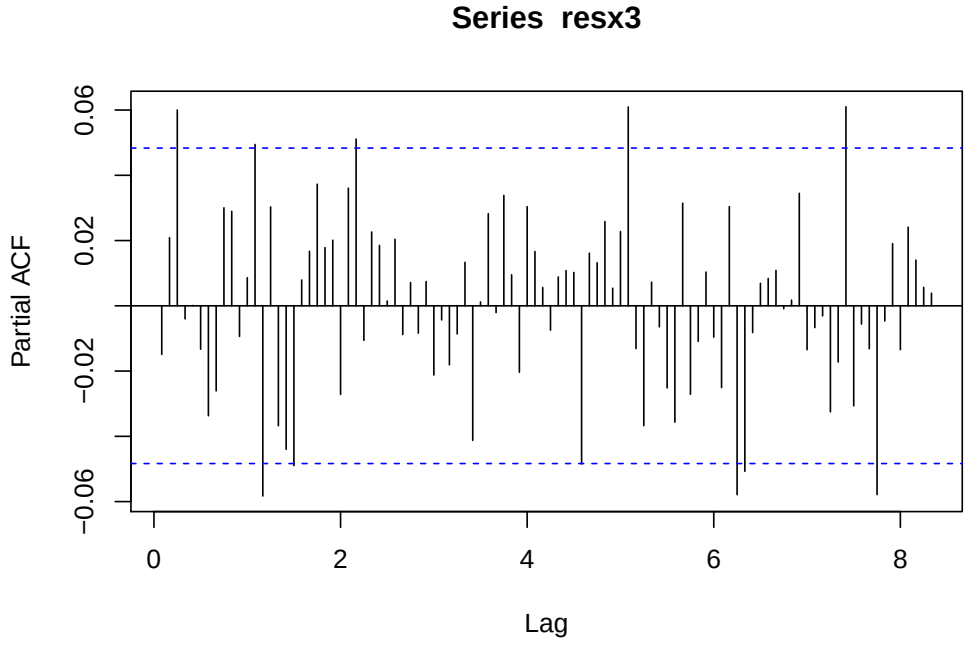






The histogram of the residuals look normally distributed and is symmetric. The sample mean is also very close to zero, being 0.05221619. The Q-Q plot also fits the line very well, which also indicates that it is normal. There also appears to be no trend.



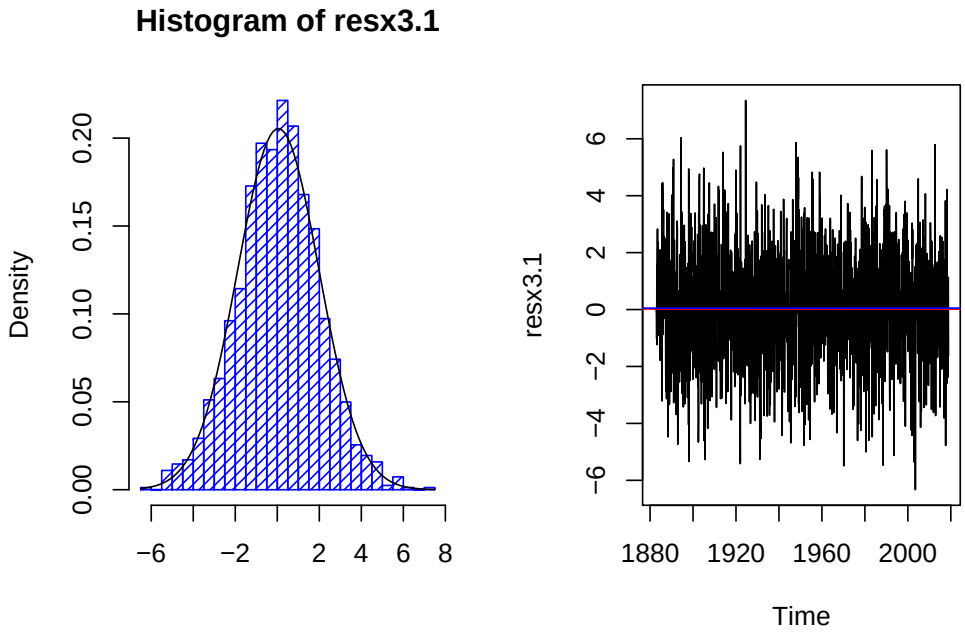


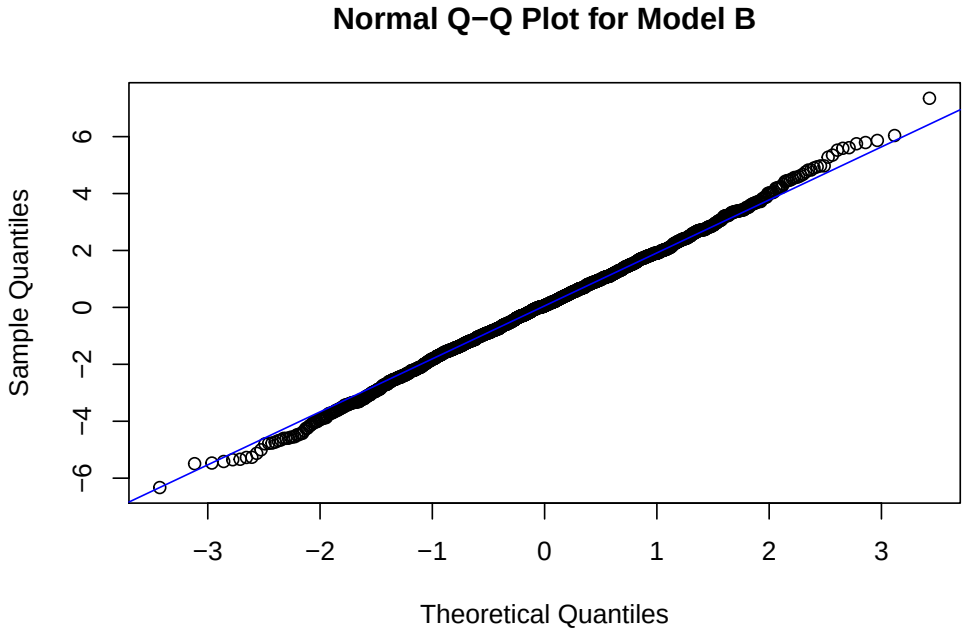
Although the acf and pacf has some lags being out of the confidence interval, it is out of 100 lags so it is most-likely fine. The residuals can be counted as zeros most of the time. Now we will try diagnostic checking for model B

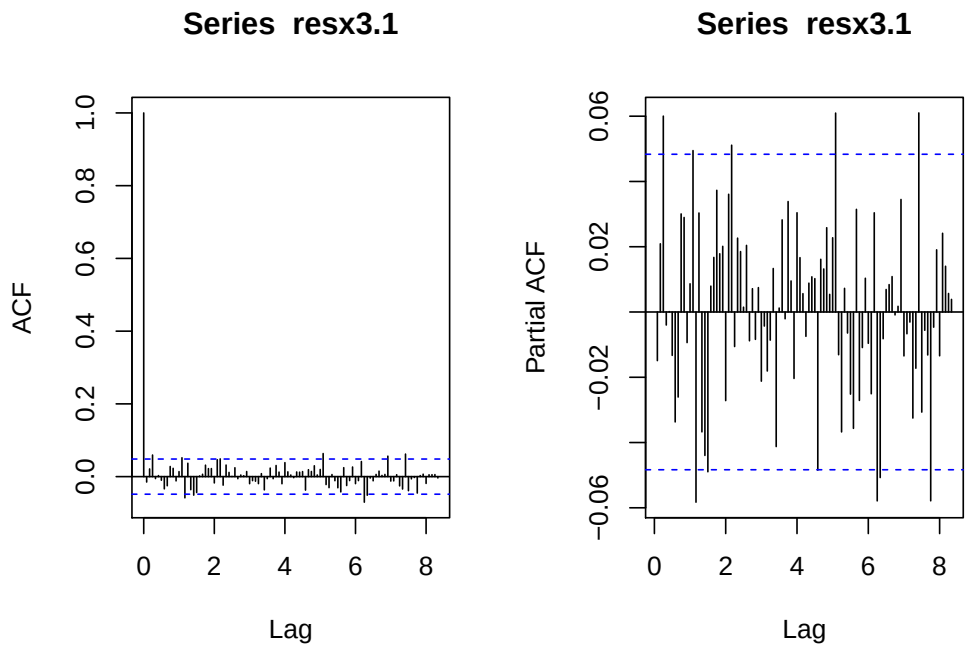
```
    Shapiro-Wilk normality test

data:  resx3
W = 0.99884, p-value = 0.3654


    Box-Pierce test

data:  resx3
X-squared = 51.747, df = 38, p-value = 0.06767


    Box-Ljung test

data:  resx3
X-squared = 52.344, df = 38, p-value = 0.06069


    Box-Ljung test

data:  (resx3)^2
X-squared = 68.322, df = 40, p-value = 0.003474
```

Model A passes all of the tests (has a p-value higher than 0.05) except the Mc-Leod-Li test where it had a very low p-value. Here lag = 40 because there are 1644 observations in the training data. This may mean that the residuals are not independent. However because the other tests pass, it may still resemble gaussian white noise.

Diagnostic Checking for Model B







Model B also looks quite similar to Model A, since their coefficients are very close in values along with having a similar structure. It looks normal and without a trend.

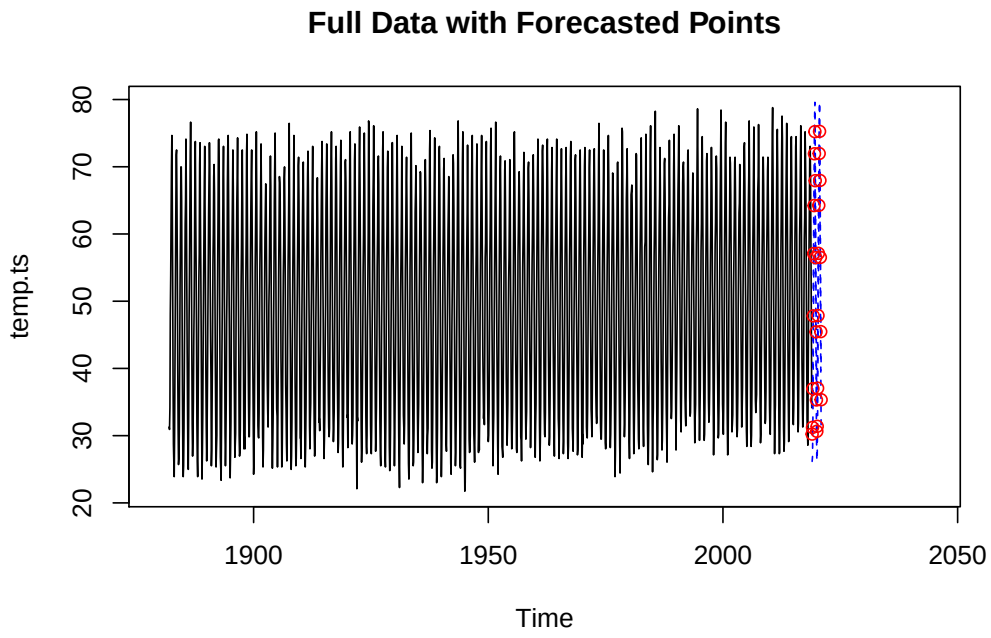
```
    Shapiro-Wilk normality test

data:  resx3.1
W = 0.99885, p-value = 0.3656
```

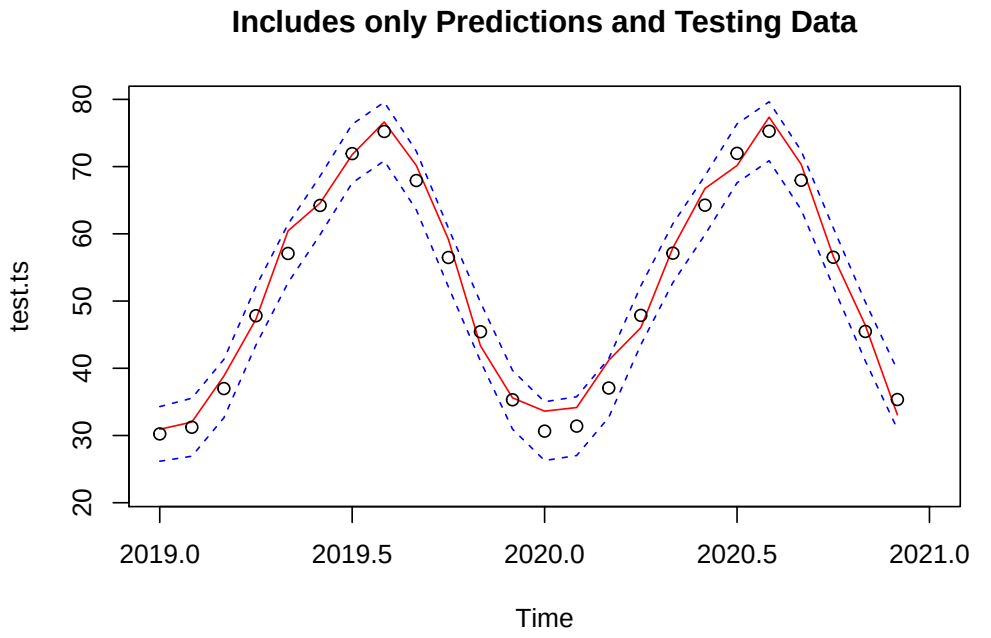
```
    Box-Pierce test

data:  resx3.1
X-squared = 51.749, df = 34, p-value = 0.02619
```

```
    Box-Ljung test

data:  resx3.1
X-squared = 52.346, df = 34, p-value = 0.02303
```

```
    Box-Ljung test

data:  (resx3.1)^2
X-squared = 68.315, df = 40, p-value = 0.003481
```

Model B seems to fail most tests except the Shapiro-Wilk test for normality. Model A seems to be the one that should be used for forecasting because even with similar results, Model B is more complex and not invertible.





24 data points were forecasted using the training data. The first graph shows all the data points but it is difficult to look at. In the second graph, the true data is the red line, the black dots are the predictions and the blue dotted lines are the confidence intervals. It seems that this model is fairly accurate at forecasting the data except for maybe at 2020 where the true data seems to deviate from the predictions.

Conclusion SARIMA$(1,1,1)(0,1,1)_{12}$

Model A, $(1-0.3378B)\nabla_{12}\nabla_1 U_t = (1-0.9829B)(1-0.9764B^{12})Z_t$, will be chosen to be the most accurate at forecasting the temperature for at least 2 years into the future. It has the lowest AIC and is able to pass most of the tests for normaliity and non-linear/linear dependence. It seems as though the beginning of the year and the end of the year are the coldest while the middle of the years will be the warmest.

References

- https://www.kaggle.com/akioonodera/monthly-temperature-of-aomori-city - Data Set link
- Professor Raya Feldmen's lectures

Appendix

```
library(MASS)
temp = read.csv("monthly_temperature_aomori_city.csv",header = TRUE)
train = temp[c(1:1644),] #divide data into training data
test = temp[c(1645:1668),]
test.ts = ts(test[,3],start = c(2019,1),frequency = 12) #changes where the data starts
test.ts = (test.ts*1.8)+32
temp.ts = ts(train[,3],start = c(1882,1),frequency = 12)
temp.ts = (temp.ts*1.8)+32 #convert to Farenheit so that it won't have negative values
ts.plot(temp.ts,main = "Temperature of Aomori City" )
abline(lm(temp.ts~as.numeric(1:length(temp.ts))),col = "blue")
abline(h=mean(temp.ts),col="red")
legend("topright", inset=c(-0.2,0), legend=c("Mean", "Trend"),
       col=c("red", "blue"), lty=1:2, cex=0.8)

op <- par(mfrow = c(1,1))
#===============box-cox==================
t=1:length(temp.ts)
fit = lm(temp.ts~t)
bcTransform= boxcox(temp.ts~t, plotit =TRUE)
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
temp.bc =(1/lambda)*(temp.ts^lambda-1)
#train = (1/lambda)*(temp.ts^lambda-1)

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

var(temp.bc)

mean(temp.bc)
op <- par(mfrow = c(1,2))
hist(temp.ts, main = "Histogram of Original Data")
hist(temp.bc,main = "Histogram of Box-Cox Transformed Data")

acf(ts(temp.ts,freq=1),lag.max = 100,main = "ACF of Original Data")
```

```r
#shows slight seasonality
pacf(ts(temp.ts,freq=1),lag.max= 100,main = "PACF of Original Data")
#======================differencing ====================
diff12 = diff(temp.ts,lag = 12, difference = 1)
ts.plot(temp.ts,main = "Original data",ylab = expression(X[t]))
ts.plot(diff12, main = "Differenced at Lag 12")
op <- par(mfrow = c(1,2))
acf(ts(diff12,freq=1),lag.max  = 100,main = "ACF 100 lags after differencing at lag 12")
pacf(ts(diff12,freq=1),lag.max = 100,main = "PACF 100 lags diff at 12")
var(diff12)

diff12_1 = diff(diff12,lag = 1,difference = 1)
ts.plot(diff12_1)#plot after differencing at lag 12 and 1
acf(ts(diff12_1,frequency=1),lag.max = 100,main = "ACF 100 lag, diff 12 and 1")#P = 12
pacf(diff12_1,lag.max = 100,main = "PACF 100 lags, diff at 12 and 1",)
var(diff12_1)
#==================trying out different models==========
x= arima(temp.ts, order=c(2,0,2), seasonal = list(order = c(4,1,1), period = 12), method="ML",o
x
x2= arima(temp.ts, order=c(2,0,3), seasonal = list(order = c(3,1,1), period = 12), method="ML",
x2
x3 <- arima(temp.ts, order=c(1,1,1), seasonal = list(order = c(0,1,1), period = 12), method="ML
x3
x3.1<- arima(temp.ts, order=c(2,1,2), seasonal = list(order = c(1,1,1), period = 12), method="M
x3.1
x3.1<- arima(temp.ts, order=c(2,1,2), seasonal = list(order = c(1,1,1), period = 12),fixed = c(
x3.1
#====================diagnostic checking==================
resx3<-residuals(x3)
hist(resx3,density=20,breaks=20, col="blue", xlab="", prob=TRUE,main = "Histogram of Residuals
m <- mean(resx3)
m
std <- sqrt(var(resx3))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(resx3)
fitt <- lm(resx3 ~ as.numeric(1:length(resx3))); abline(fitt, col="red")
abline(h=mean(resx3), col="blue")
qqnorm(resx3,main= "Normal Q-Q Plot for Model B")
qqline(resx3,col="blue")

acf(resx3,lag.max = 100)
pacf(resx3,lag.max = 100)

shapiro.test(resx3)#probably shouldn't use transformations
Box.test(resx3,lag = 40,type = c("Box-Pierce"),fitdf = 2)
Box.test(resx3, lag = 40, type = c("Ljung-Box"), fitdf = 2)
Box.test((resx3)^2, lag = 40, type = c("Ljung-Box"), fitdf = 0)

op <- par(mfrow = c(1,2))
resx3.1<-residuals(x3.1)
hist(resx3.1,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(resx3.1)
std <- sqrt(var(resx3.1))
curve( dnorm(x,m,std), add=TRUE )
```

```r
plot.ts(resx3.1)
fitt <- lm(resx3.1 ~ as.numeric(1:length(resx3.1))); abline(fitt, col="red")
abline(h=mean(resx3.1), col="blue")
op <- par(mfrow = c(1,1))
qqnorm(resx3.1,main= "Normal Q-Q Plot for Model B")
qqline(resx3.1,col="blue")

op <- par(mfrow = c(1,2))
acf(resx3.1,lag.max = 100)
pacf(resx3.1,lag.max = 100)

shapiro.test(resx3.1)
Box.test(resx3.1,lag = 40,type = c("Box-Pierce"),fitdf = 6)
Box.test(resx3.1, lag = 40, type = c("Ljung-Box"), fitdf = 6)
Box.test((resx3.1)^2, lag = 40, type = c("Ljung-Box"), fitdf = 0)
#=======================forecasting==================
library(forecast)
#forecast(x3)
pred.tr <- predict(x3, n.ahead = 24)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(temp.ts, xlim=c(1880,+2020+24), ylim = c(min(temp.ts),max(U.tr)),main = "Full Data with
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points( pred.tr$pred, col="red")

ts.plot(test.ts,xlim = c(2019,2021),ylim = c(min(temp.ts),max(U.tr)),col="red",main = "Includes
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points( pred.tr$pred, col="black")
```