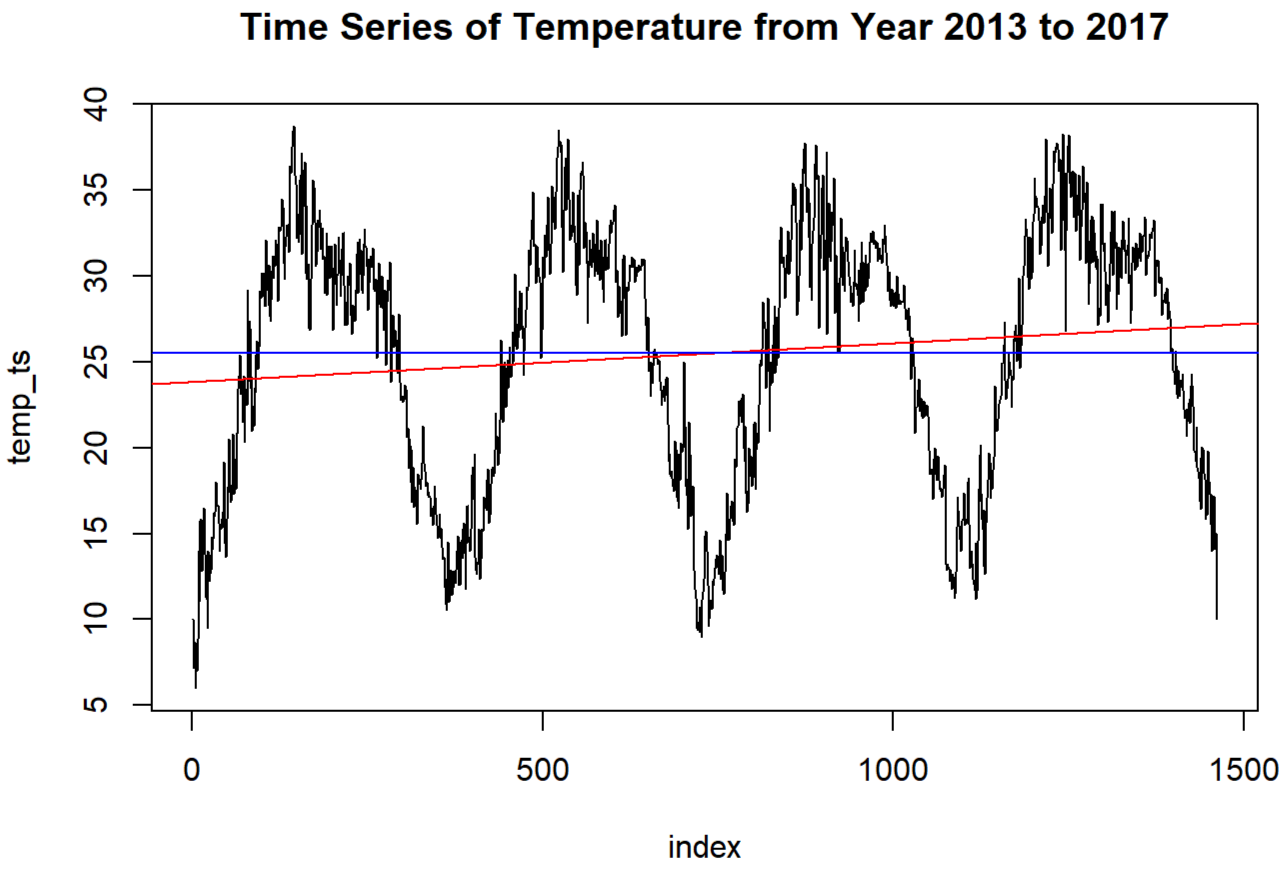
# Forecasting Temperatures in Delhi Using Time Series

Ethan Tran

2025-03-21

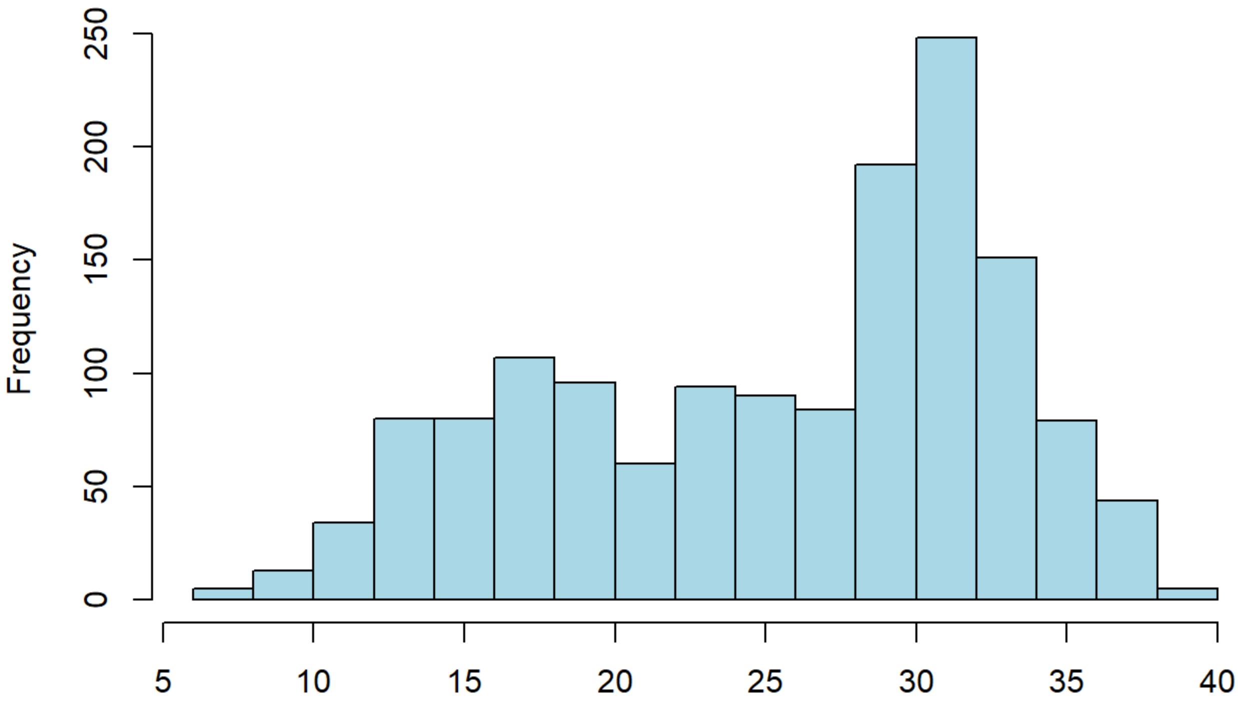
## Introduction

In this project, I will attempt to forecast the temperature in the city of Delhi using previous data. The two most vital variables in the dataset I will be using are the date and average daily temperature between 2013 and 2017 in Delhi, which I will be using to formulate a time series. This dataset is interesting and important because after thorough analysis, one can undeniably prove that the climate is changing and attempt to predict where our planet is headed in the coming years. I used many techniques and testing many models in this project in an attempt to accurately forecast the temperature of Delhi after 2017, which you will see throughout the project report. There were many positives and the project went smoothly until I ran into some issues with the diagnostics checking. I spent all night trying to but unfortunately I could not figure out how to pass the Box-Ljung and Box-Pierce test, and could not figure out how to properly plot my forecasts. From my project, I can conclude that the average temperatures of Delhi did slightly increase over time. I used R for my project and found my dataset on Kaggle.

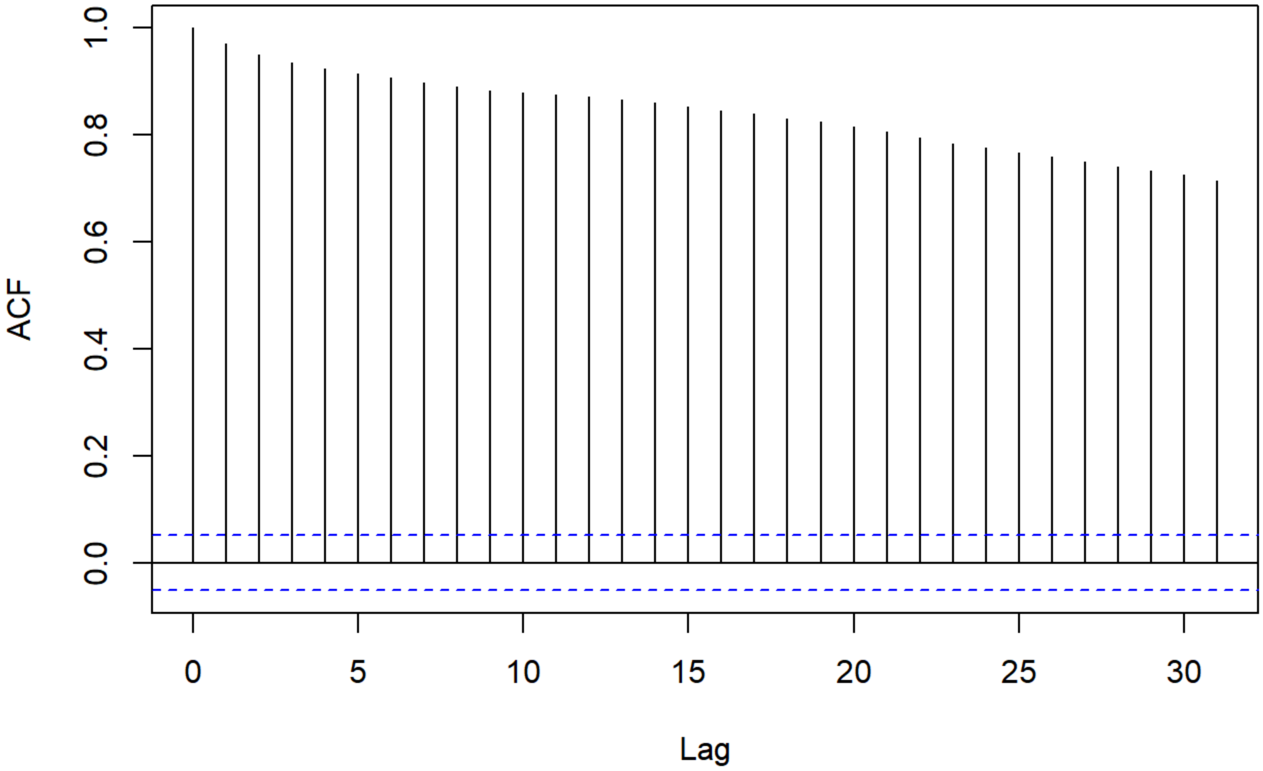


Looking at the plot of the time series, there is a clear positive trend and seasonal component in this time series. There are no apparent sharp changes in behavior. The variance seems to slightly increase over time. I will try a log transform to stabilize the variance. Differencing will need to be applied in order to make this a stationary time series for forecasting.

Histogram of Temperature Time Series

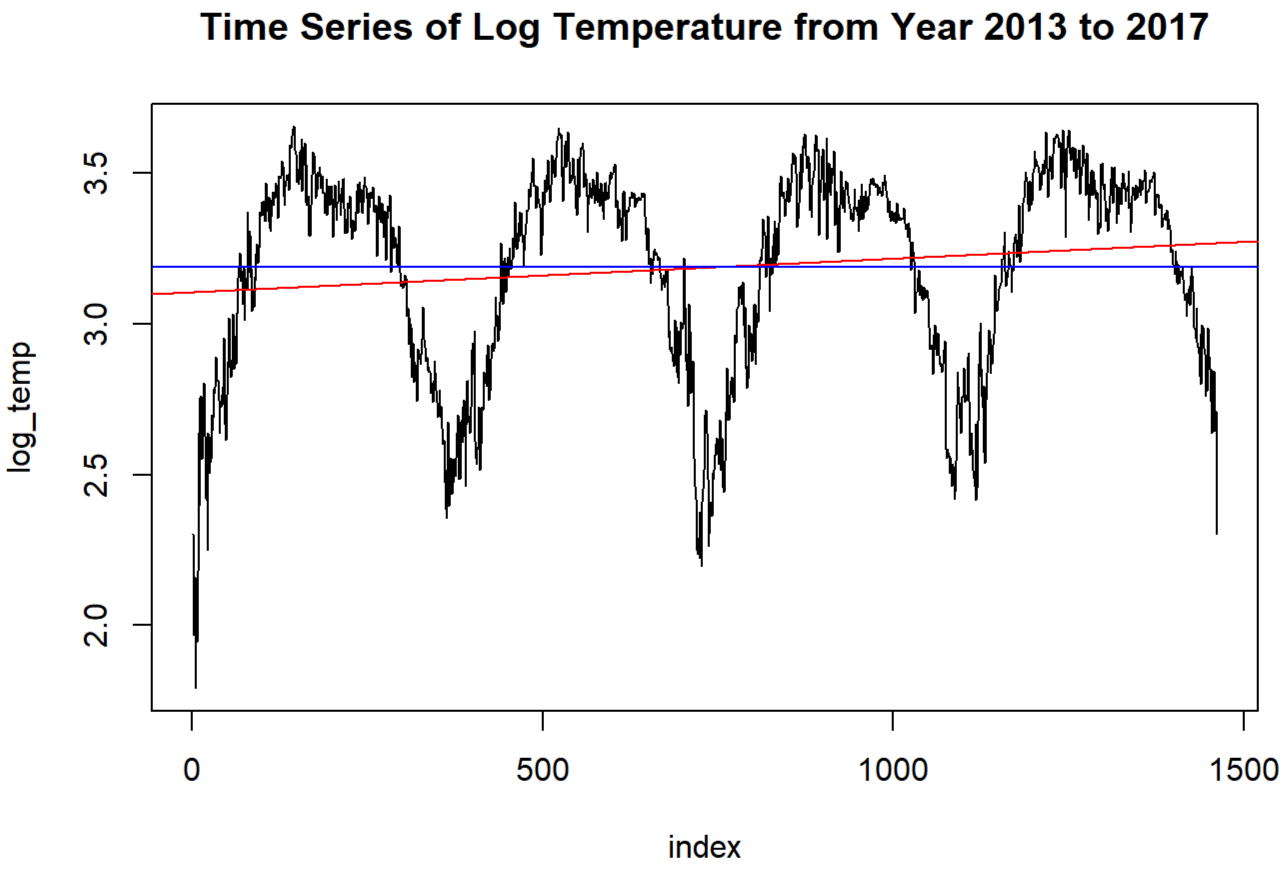


ACF of Temperature Time Series



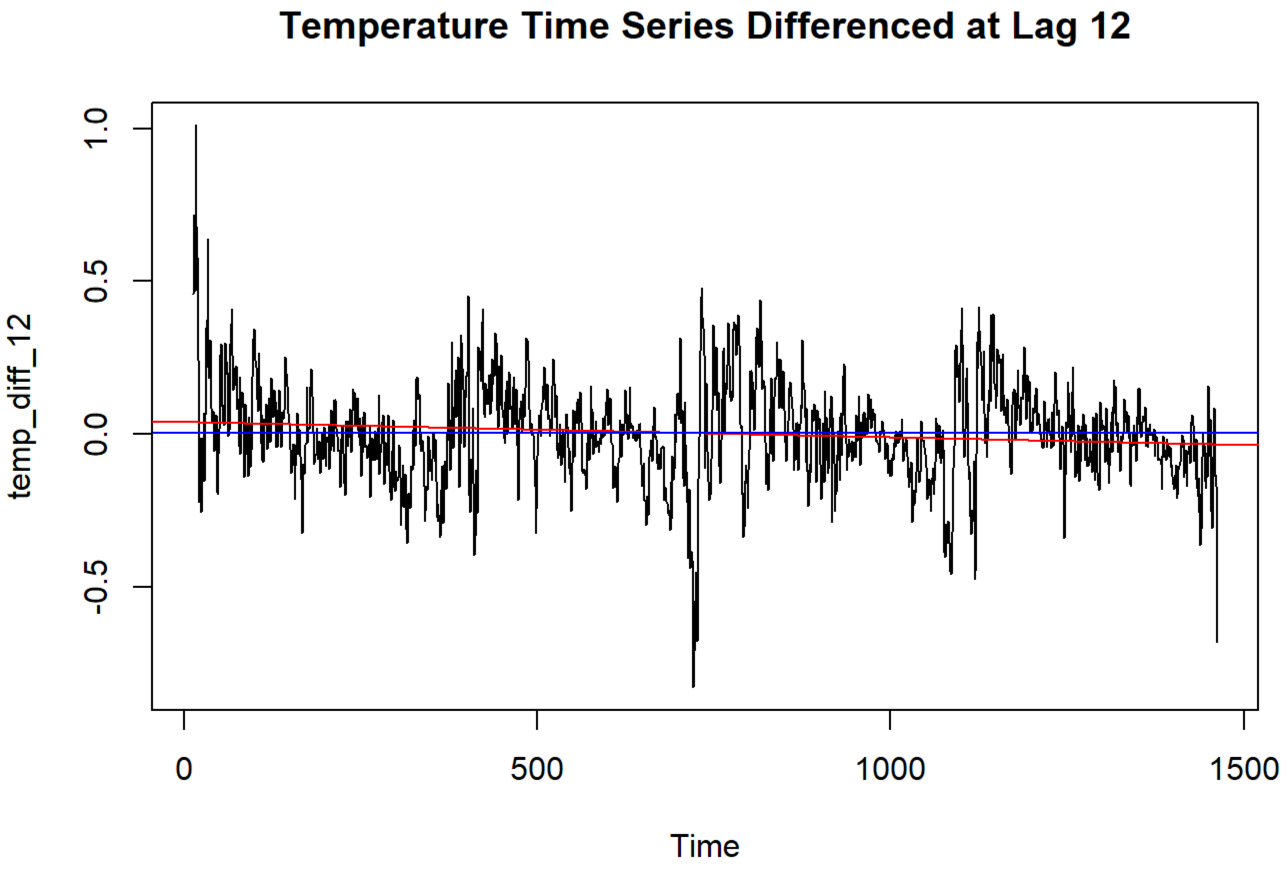
```
## [1] 53.99461
```

The histogram has a slight left skew and the ACF plot shows slow decay. The variance is about 54. If the variance increases after differencing, this indicates over-differencing.



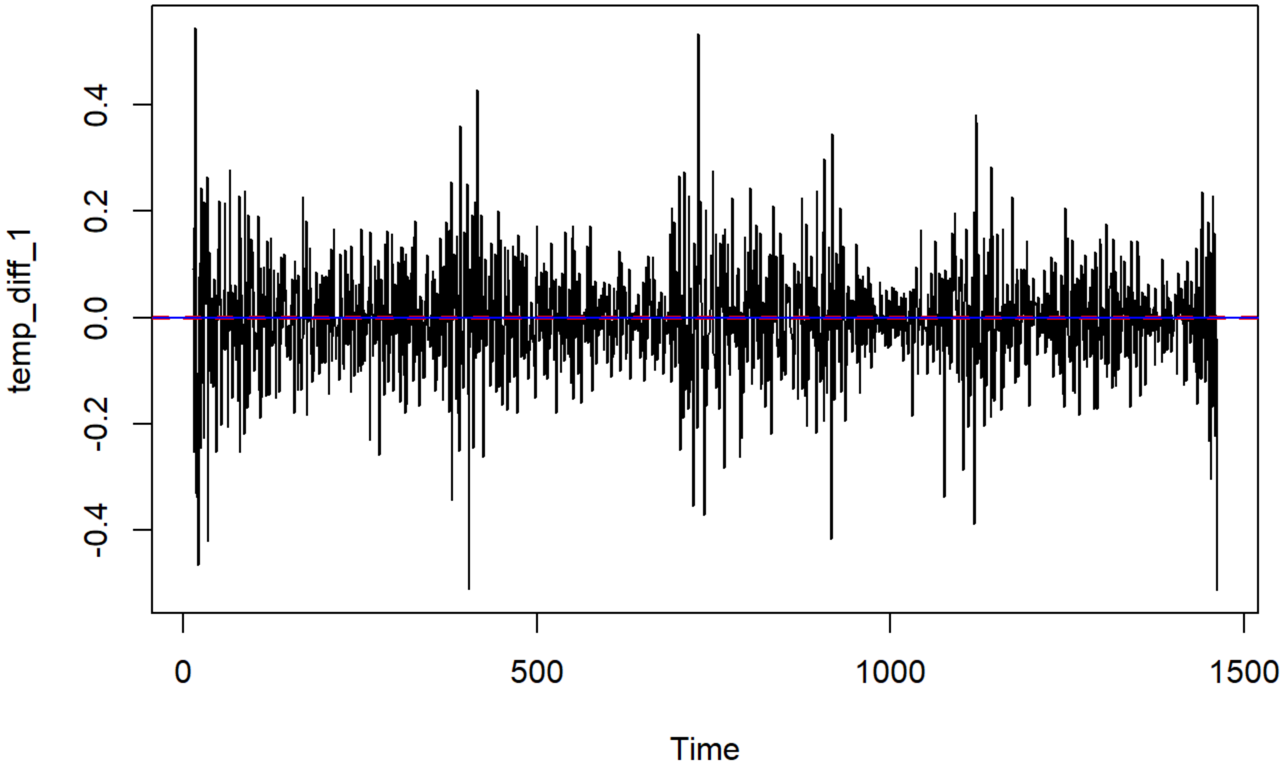
```
## [1] 0.1126211
```

After the log transform, the variance decreased from about 54 to about 0.1126.



```
## [1] 0.02594788
```

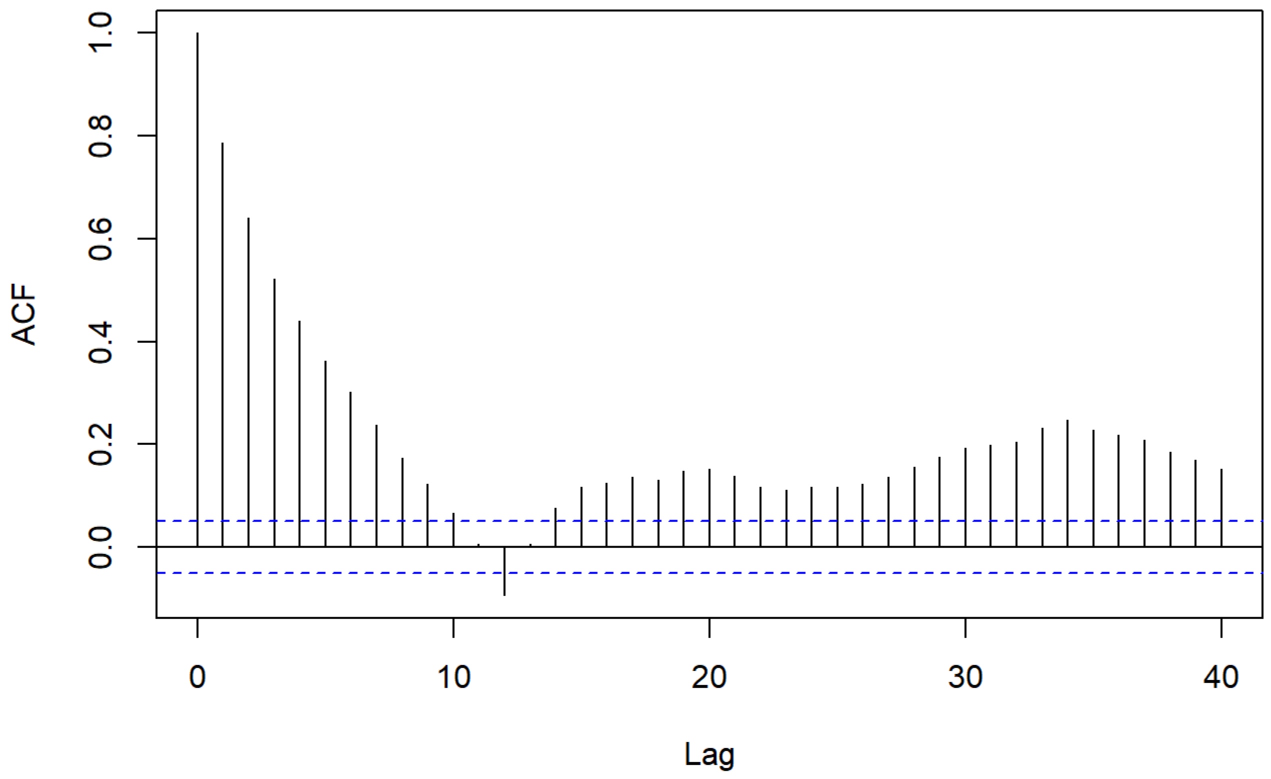
I then differenced at lag 12, from which the variance reduced to about 0.0259. The trend was essentially gone and the seasonal component was greatly reduced, though still visible.



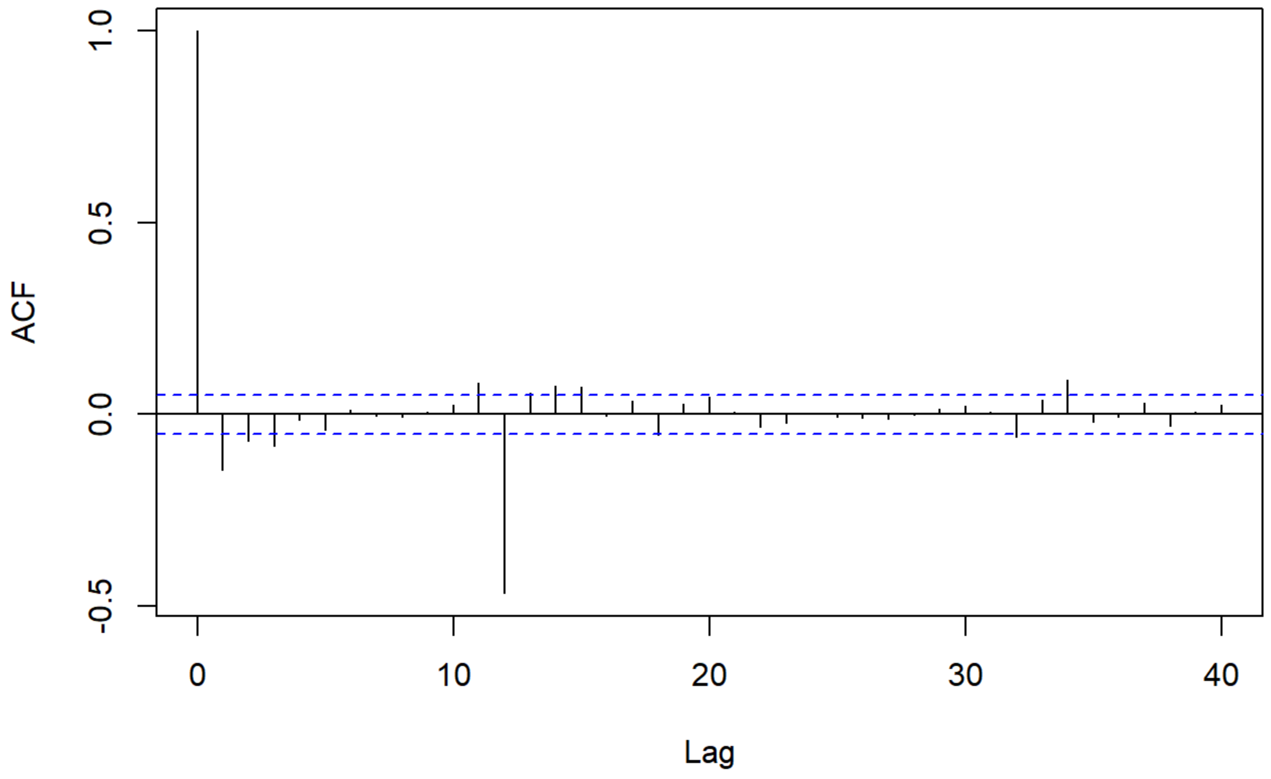
```
## [1] 0.01063484
```

Finally, I differenced again at lag 1 and the seasonal component and trend were no longer visible. The variance reduced to about 0.0106. Visually, the data exhibits stationary traits.

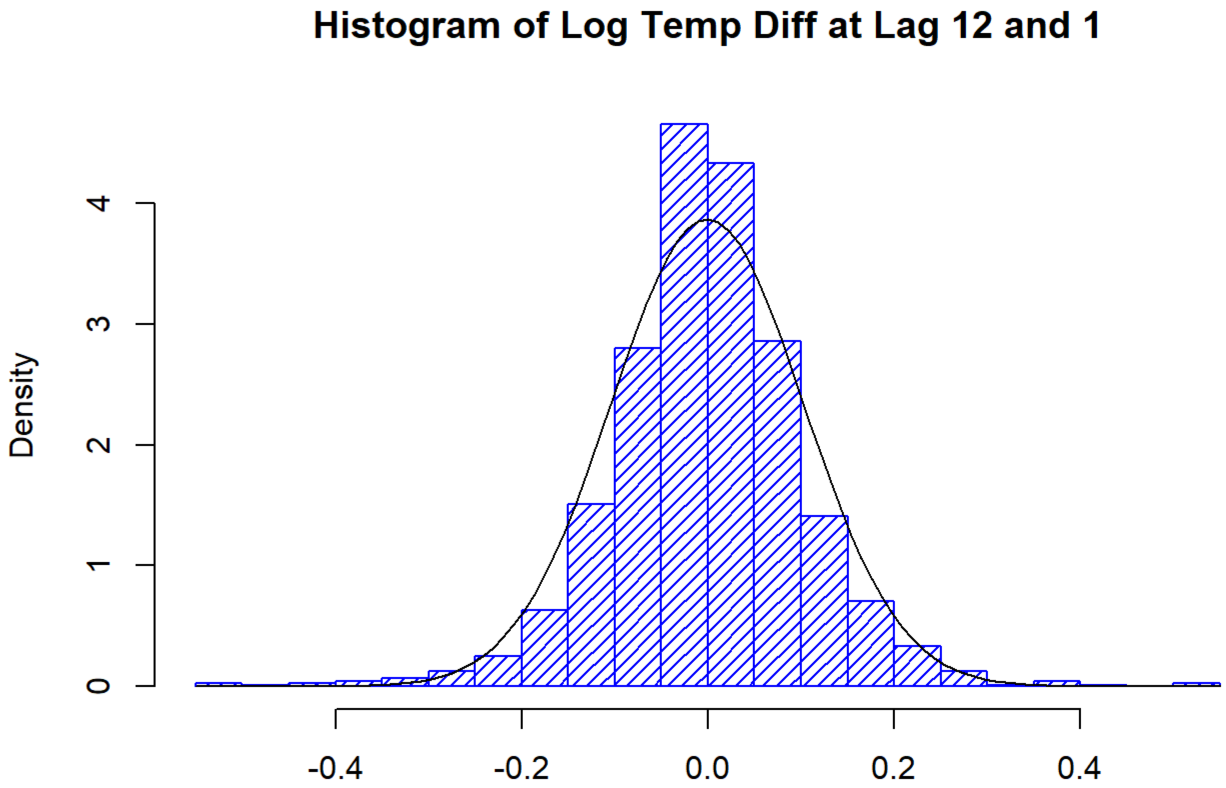
ACF of Temp Differenced at Lag 12



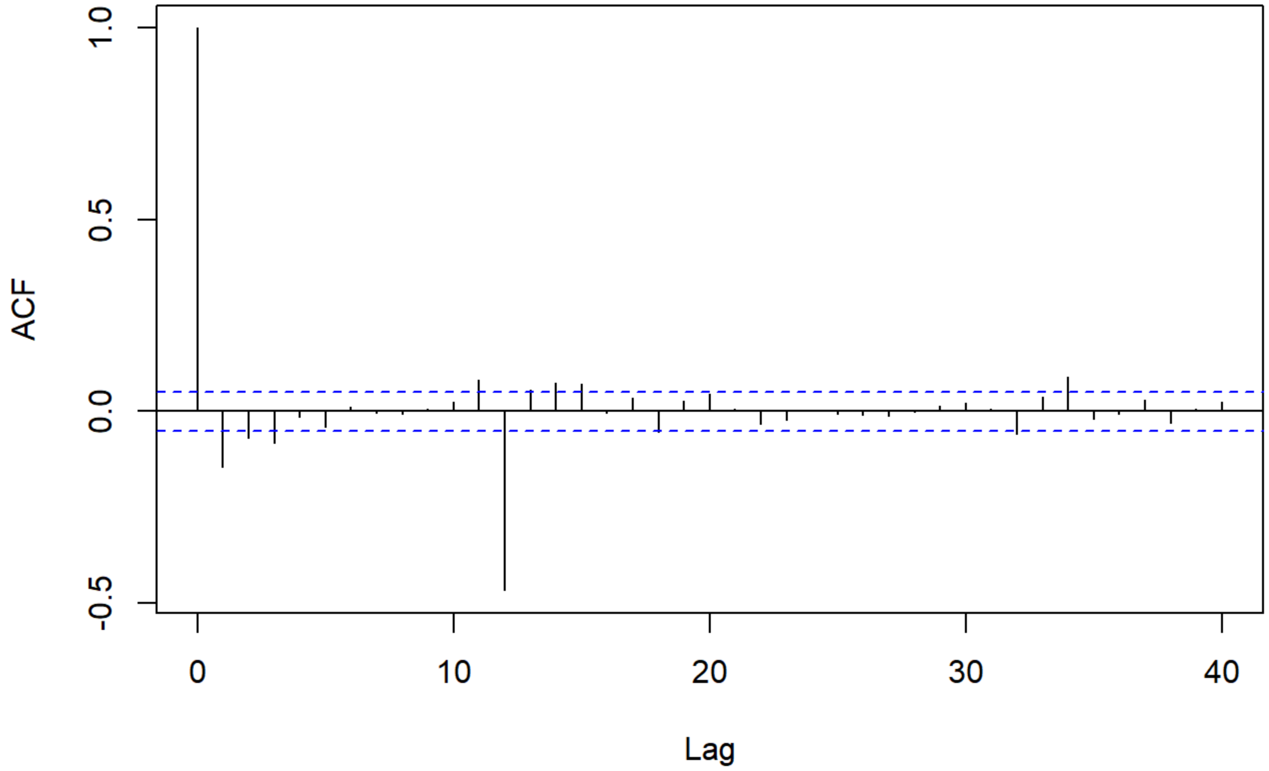
ACF of Temp Differenced at Lag 12 and 1



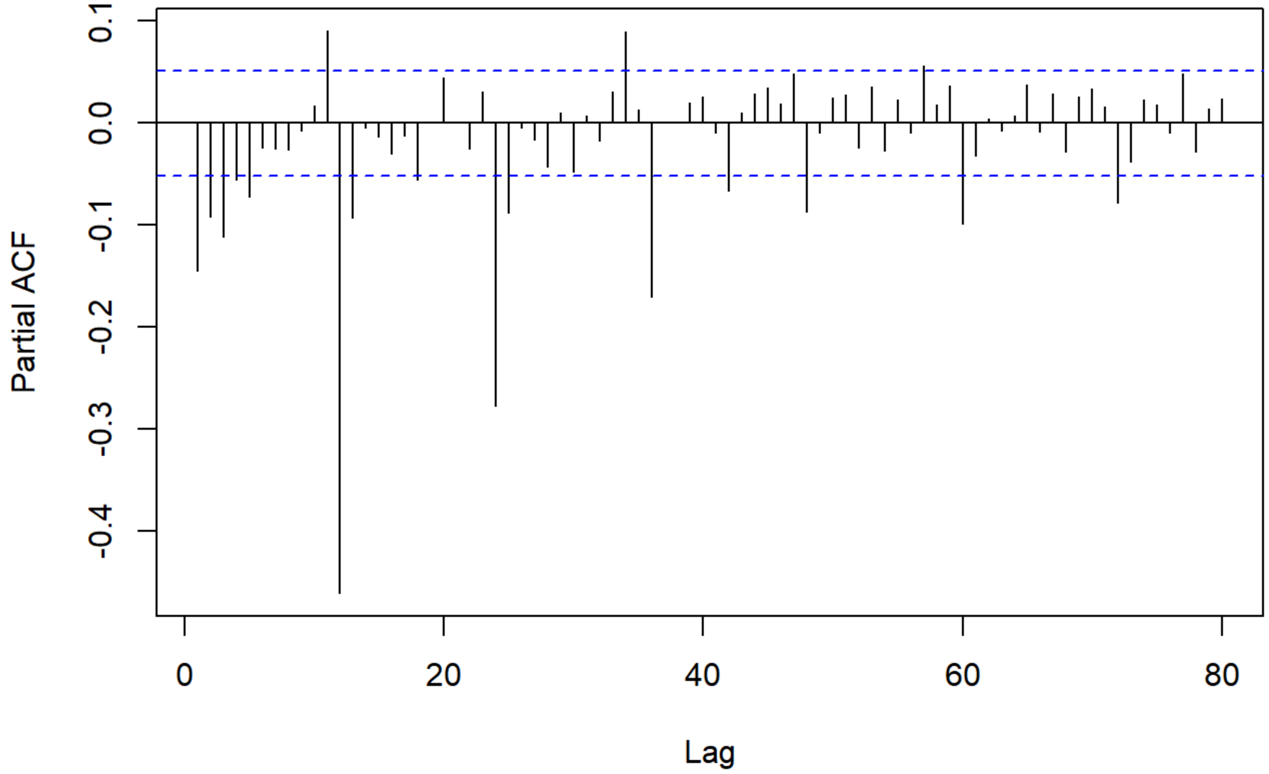
After looking at the ACF plots, we can see that after differencing at lag 12, the seasonal component is less visible but there is still a slow decay and significant ACF at high lags, indicating non stationarity. After differencing again at lag 1, we see a nice ACF plot that looks like a stationary process.



The histogram of the data resembles a normal distribution, which is ideal.



ACF of Temp Differenced at Lag 12 and 1



PACF of Temp Differenced at Lag 12 and 1

Plausible Models to Test SARIMA: s = 12, p = 1 or 3, P = 1-6 d = 1, D = 1, q = 1, 2, or 3, Q = 1 s=12 and D=1 as we differenced at lag 12 one time, and d=1 since we differenced at lag 1 once. The ACF is outside the confidence intervals at multiple points, but most noticeably are at lag 1, 3, 11, and 12. Q=1 and q could either be 1, 2, or 3. The PACF is noticeably outside the confidence intervals at 1-4, 11-13, 24-25, and 36, so P could be 1, 2, or 3. The PACF is significant at and around multiples of s=12. p could be 1 or 3. SMA: set P and p to 0, no AR component SAR: set Q and q to 0, no MA component

```
##
## Call:
## arima(x = log_temp, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##          ar1      ma1    sar1     sma1
##       0.5834  -0.7814  0.0123  -0.9856
## s.e.  0.0565   0.0431  0.0291   0.0119
##
## sigma^2 estimated as 0.00544:  log likelihood = 1700.29,  aic = -3390.57
```

```
##
## Call:
## arima(x = log_temp, order = c(3, 1, 1), seasonal = list(order = c(1, 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##          ar1     ar2     ar3      ma1    sar1     sma1
##       0.5546  0.0425  -0.044  -0.7654  0.0113  -0.9862
## s.e.  0.0692  0.0334   0.031   0.0643  0.0291   0.0121
##
## sigma^2 estimated as 0.005427:  log likelihood = 1701.81,  aic = -3389.62
```

After testing every possible model, my best one was a SARIMA model (1, 1, 1) x (1, 1, 1) x 12. I know this because it has the lowest AIC value of -3390.57. Because the sample size was so large, I chose to use AIC value. My second lowest AIC value was another SARIMA model (3, 1, 1) x (1, 1, 1) x 12, with a value of -3389.62. I will be using these two models for diagnostic testing.

Model A: $\boldsymbol{\nabla}\mathbf{1}$ $\boldsymbol{\nabla}$12 ln($U_t$) = (1 - $0.5834_{(0.0565)}$B)(1 - $0.7814_{(0.0431)}$B)(1 - $0.0123_{(0.0291)}B^{12}$)(1- $0.9856_{(0.0119)}B^{12}$)

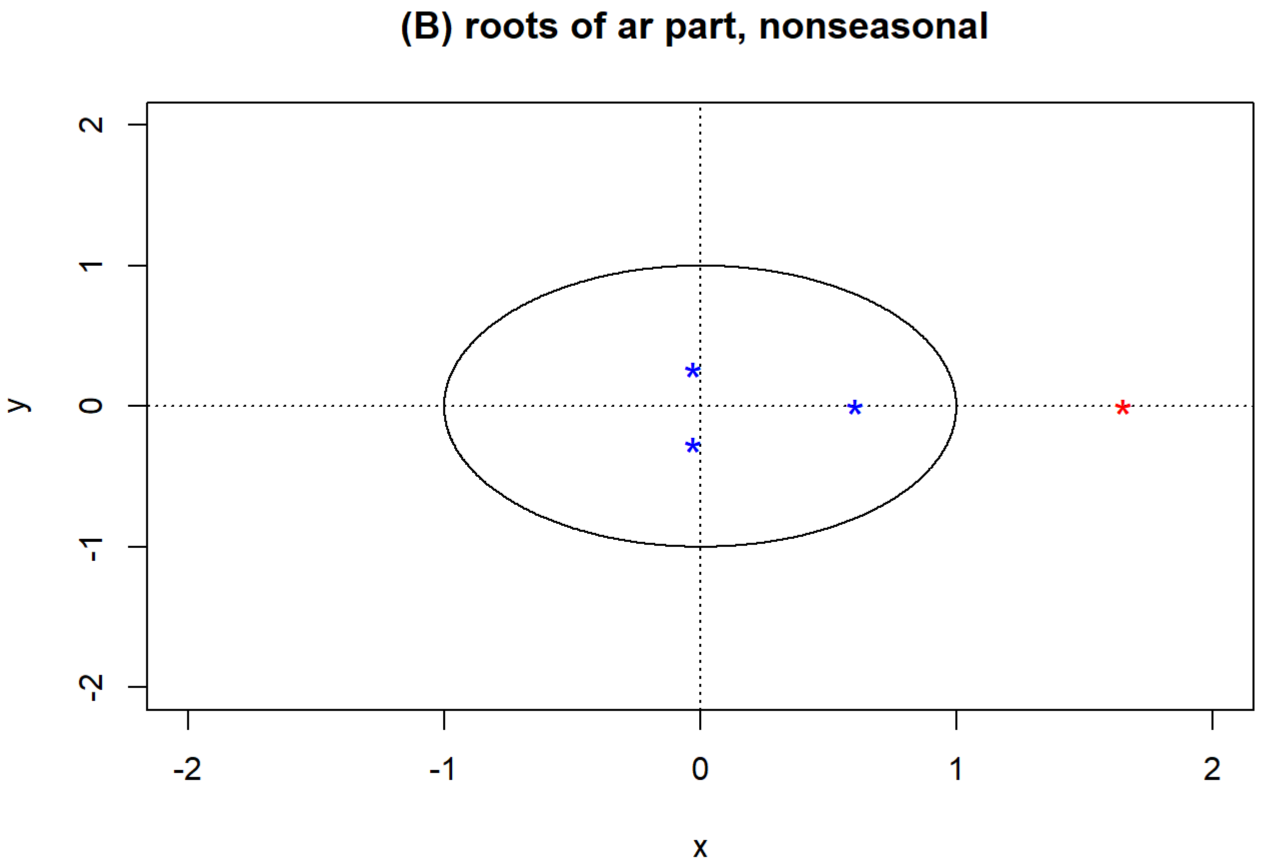
$\hat{\sigma}^2$ = 0.00544

Model B: $\boldsymbol{\nabla}\mathbf{1}$ $\boldsymbol{\nabla}$12 ln($U_t$) = (1 - $0.5546_{(0.0692)}$B - $0.0425_{(0.0334)}B^2$ + $0.044_{(0.031)}$ $B^3$)$(1 - 0.7654_{(0.0643)}$B)(1 - $0.0113_{(0.0291)}B^{12}$)(1 - $0.9862_{(0.0121)}B^{12}$)

$\hat{\sigma}^2$ = 0.005427

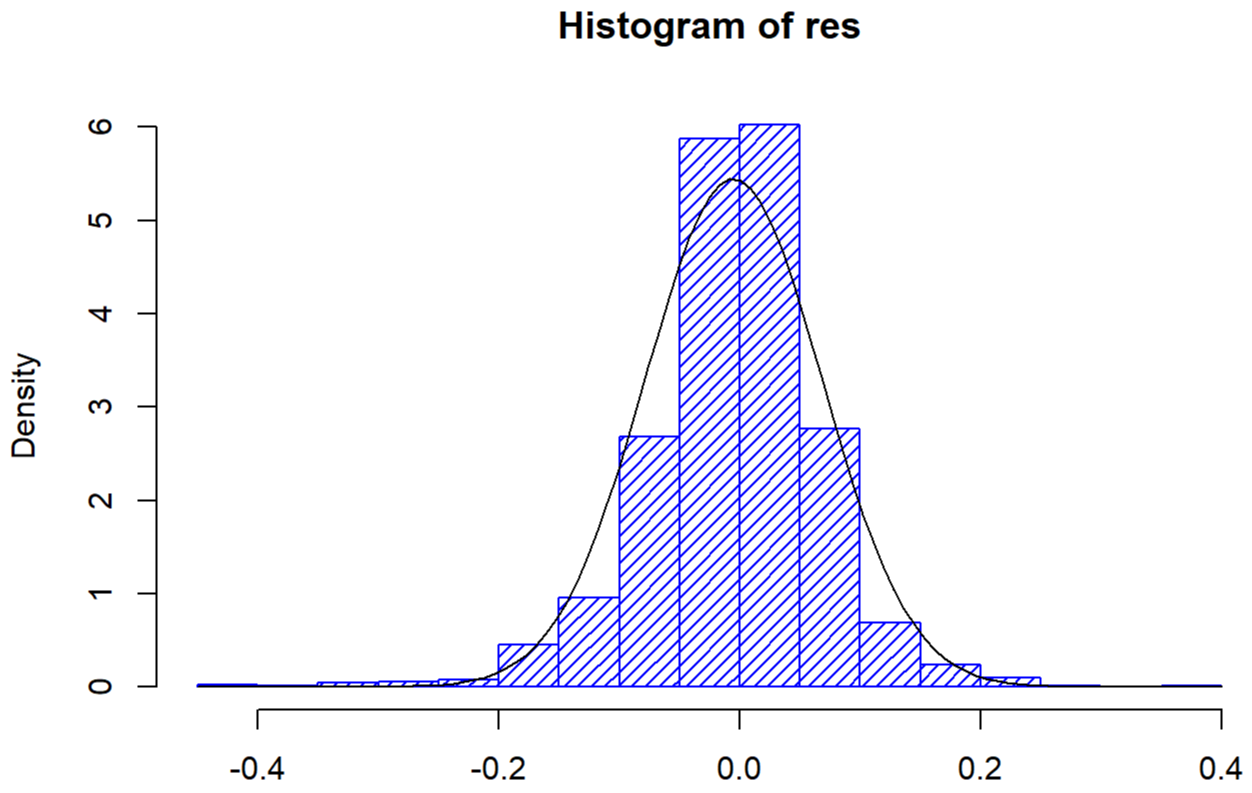
Model A is stationary and invertible because |ar1| < 1, |sar1| < 1, |ma1| < 1, and |sma1| < 1.

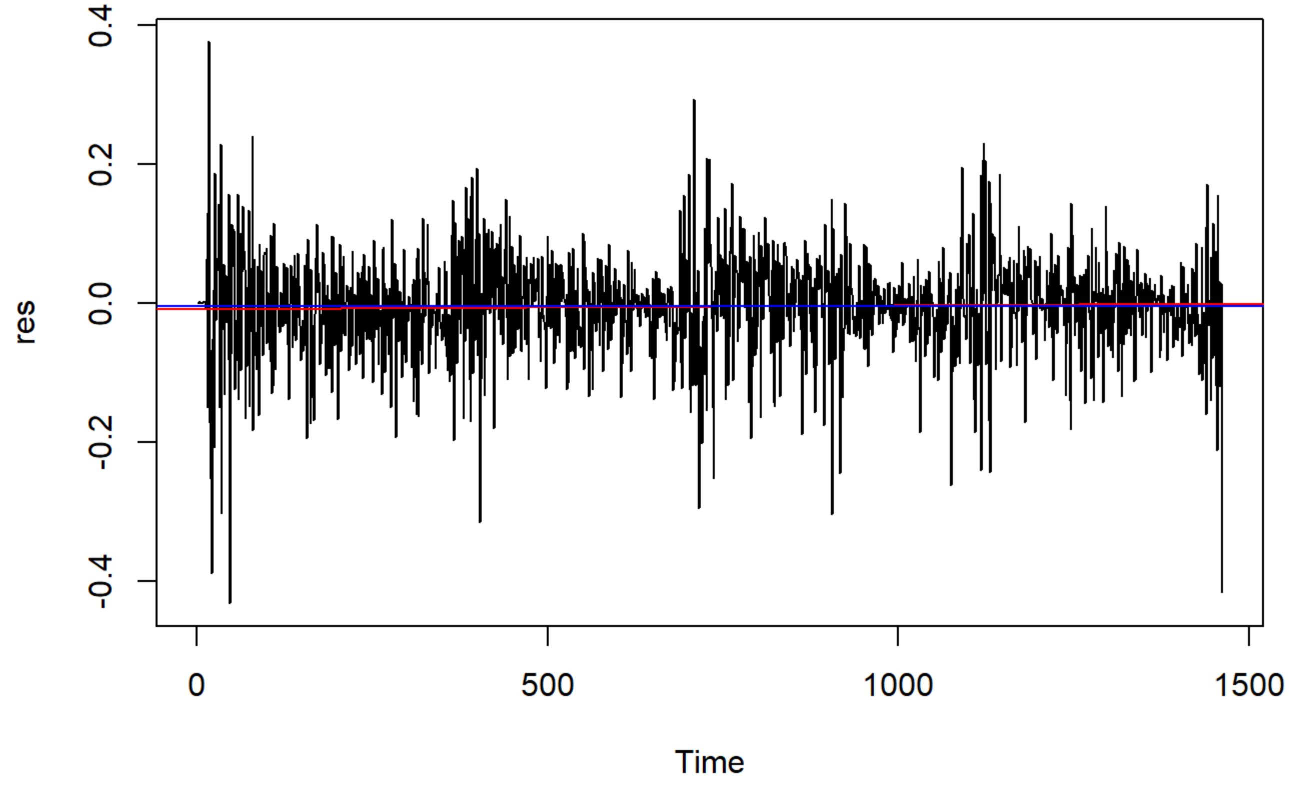
For Model B, we will have to check the non seasonal AR polynomial for if the roots are all greater than 1 for stationarity. We already know it is invertible because |ma1| and |sma1| < 1. If the red points, the roots of the polynomial, are outside the unit circle, then we know it is stationary.

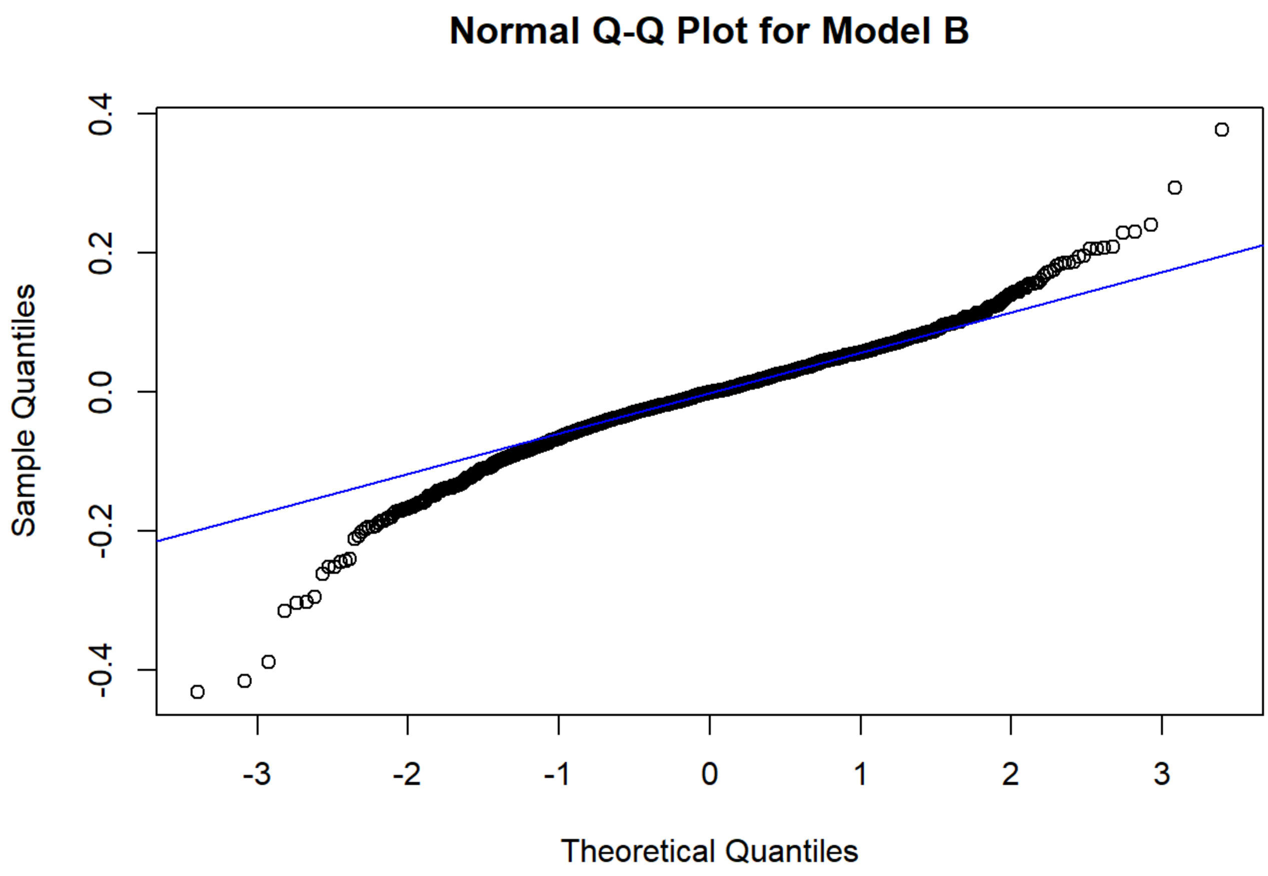


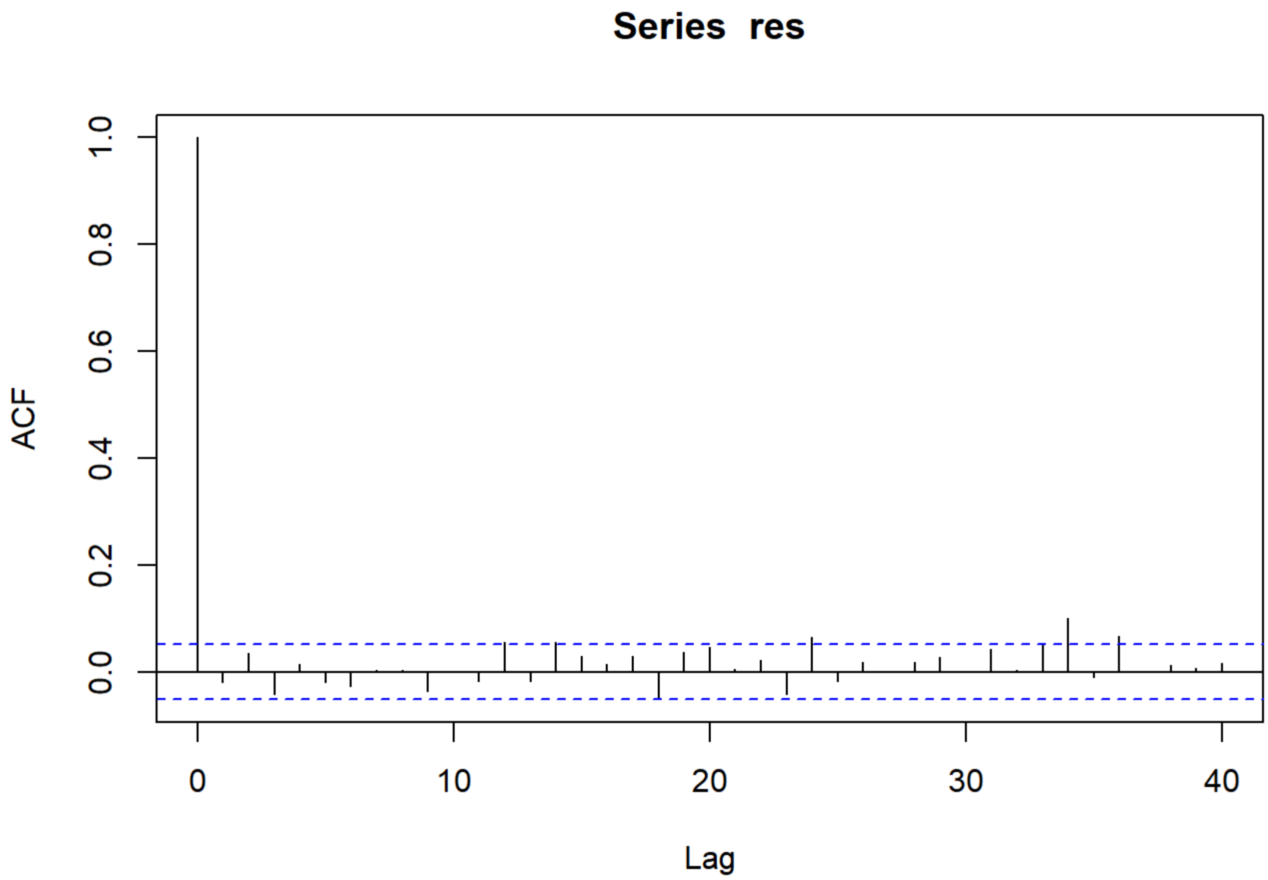
```
## [1]  1.6538259+6.923502e-22i -0.3439584+3.691061e+00i -0.3439584-3.691061e+00i
```

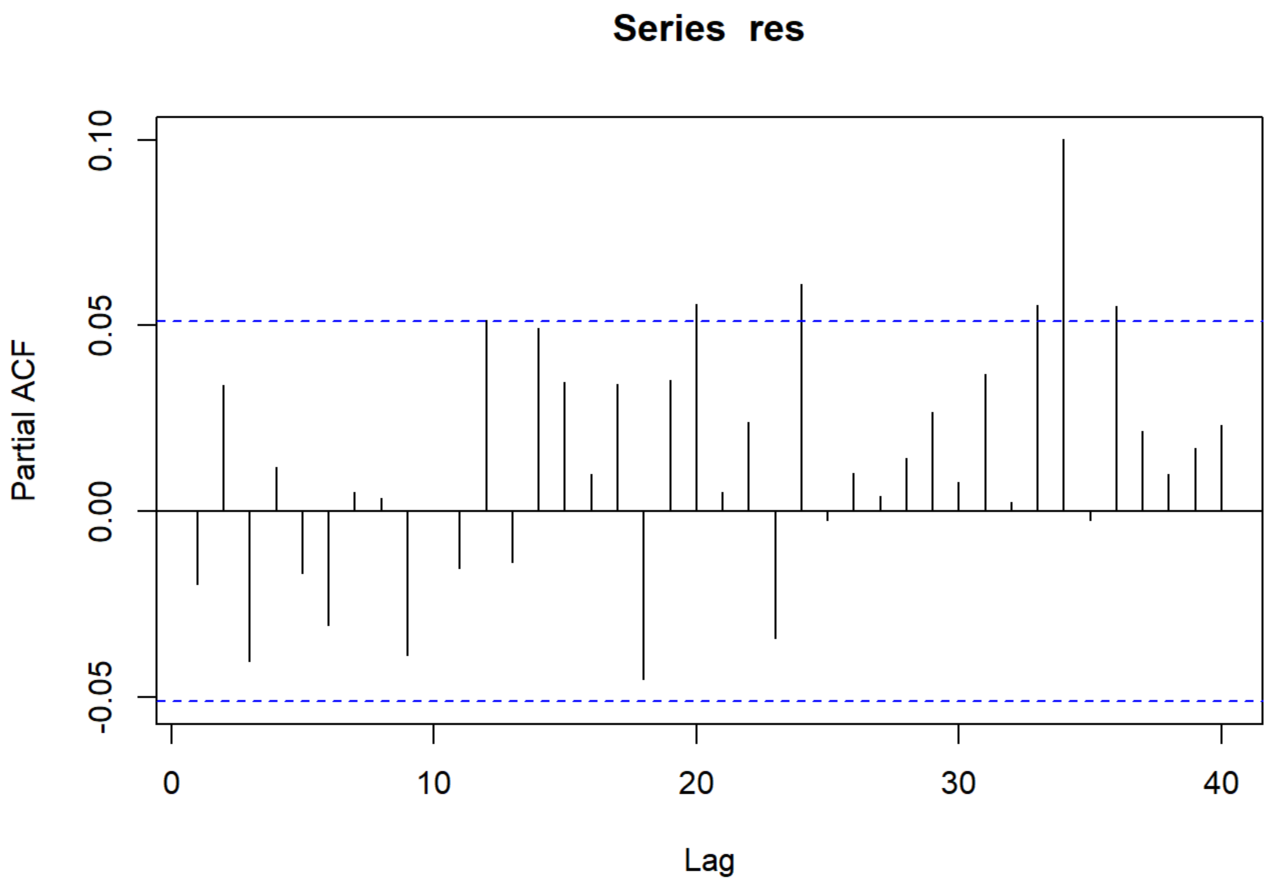
From the graph, we can see that both models A and B are stationary and invertible.


Histogram of res
Density
0
1
2
3
4
5
6
-0.4
-0.2
0.0
0.2
0.4


res
-0.4
-0.2
0.0
0.2
0.4
0
500
1000
1500
Time


Normal Q-Q Plot for Model B
Sample Quantiles
Theoretical Quantiles
0.4
0.2
0.0
-0.2
-0.4
-3
-2
-1
0
1
2
3


Series res
ACF
Lag
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40

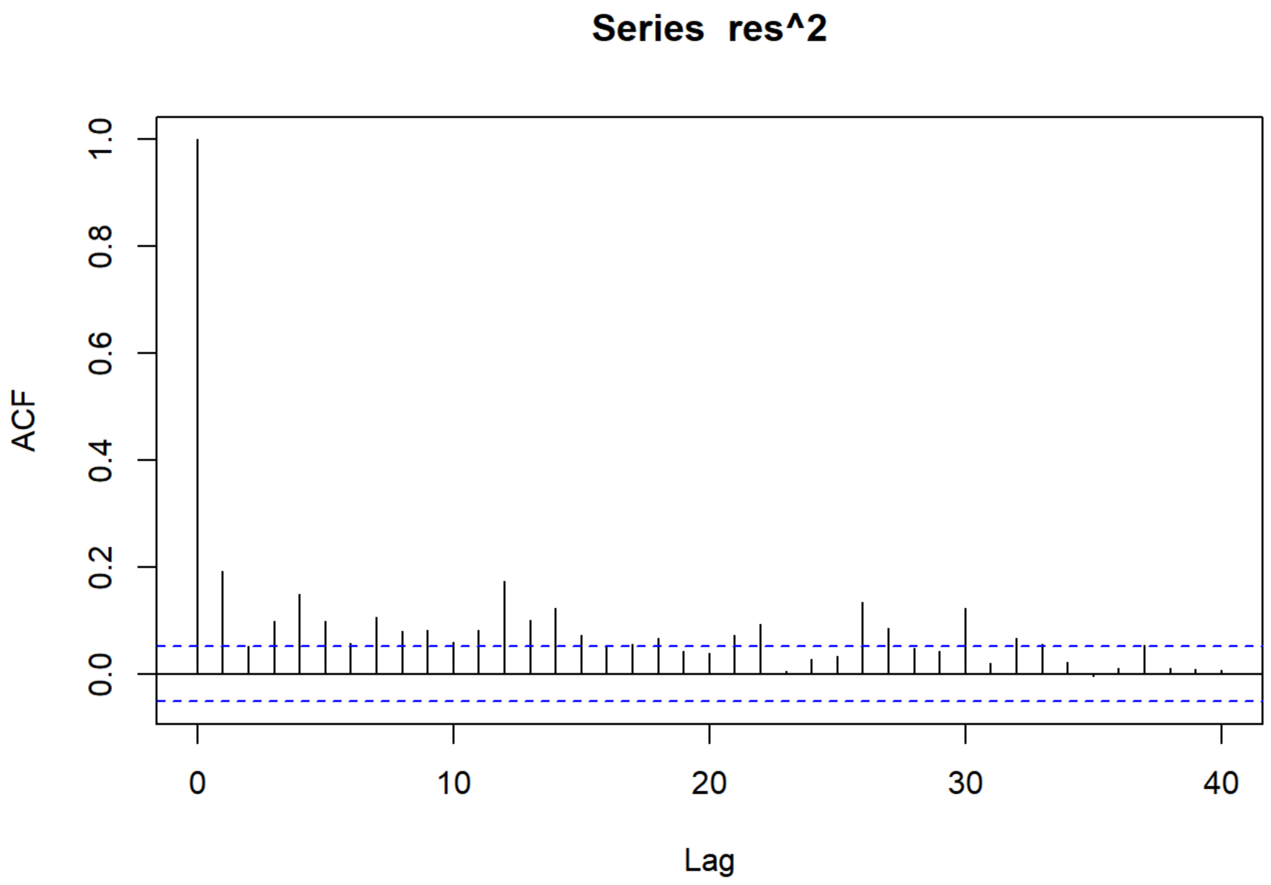


```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.9605, p-value < 2.2e-16
```

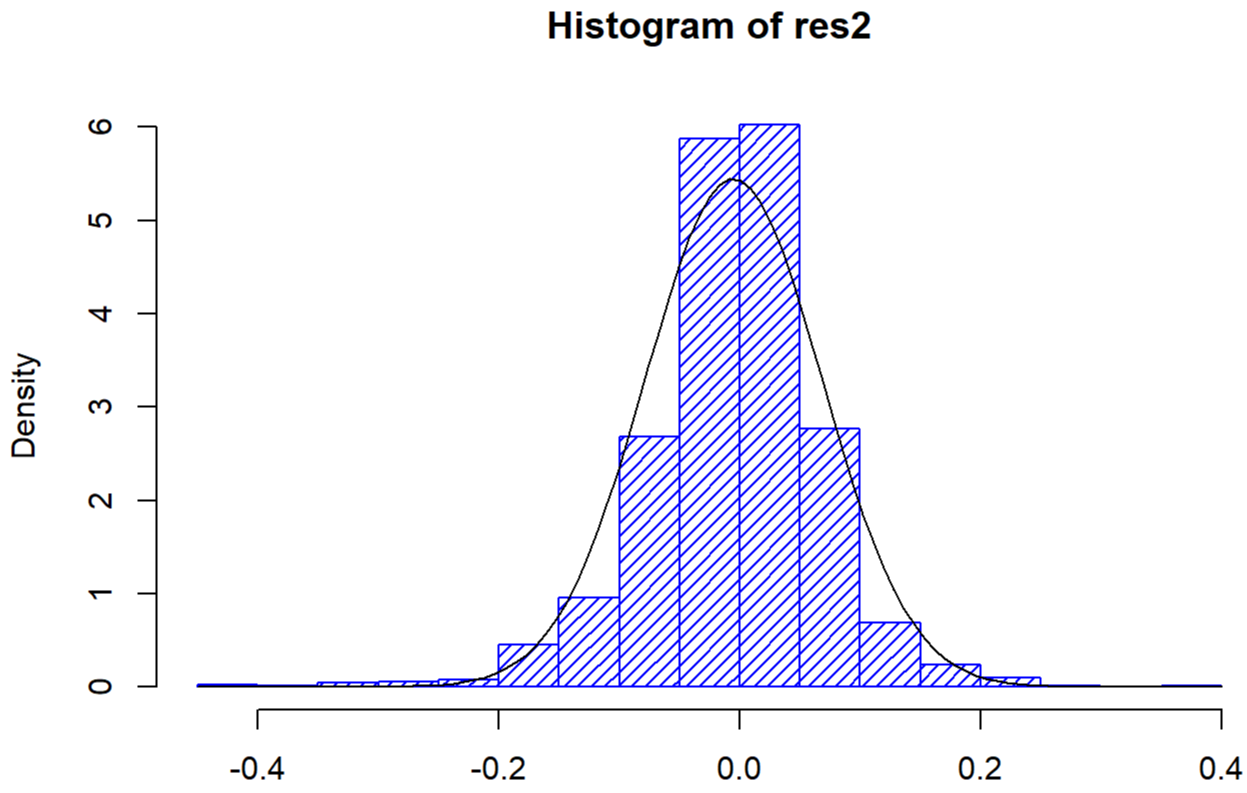
```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 70.33, df = 34, p-value = 0.0002471
```

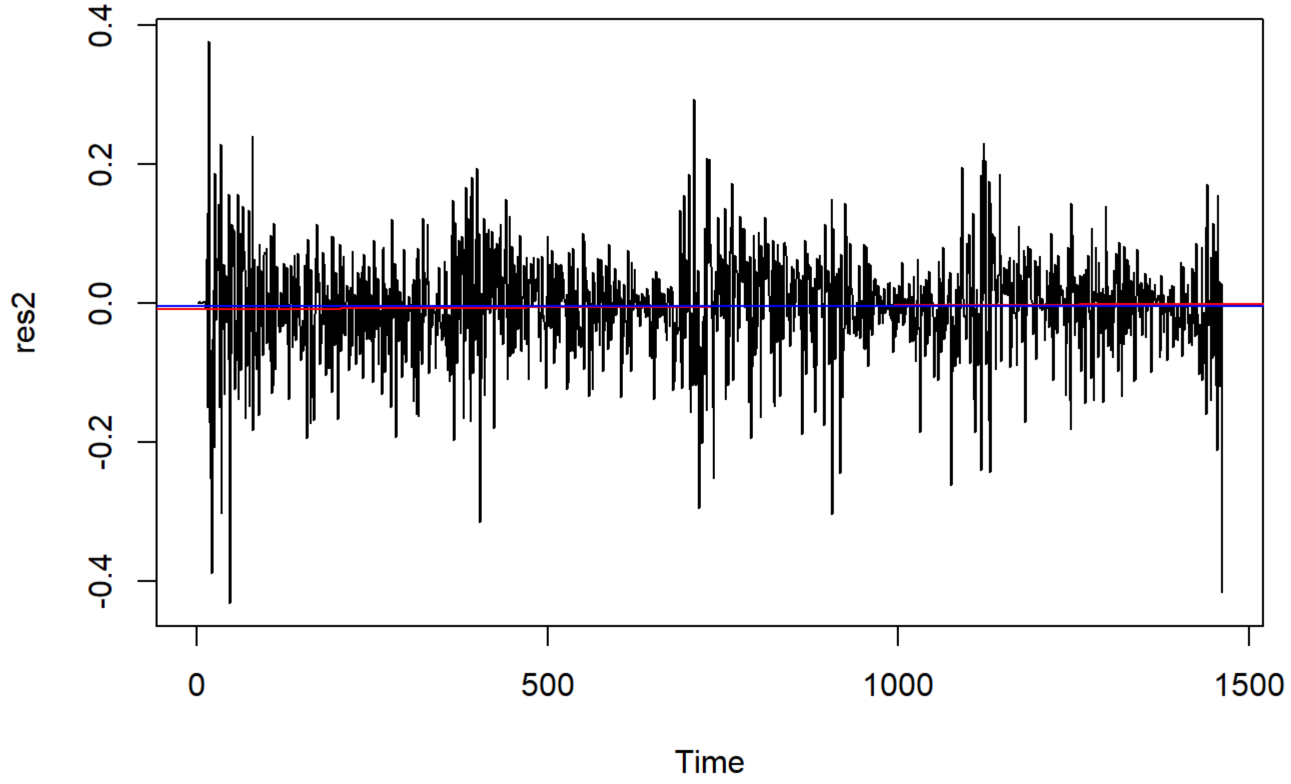
```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 71.566, df = 34, p-value = 0.0001739
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 218.9, df = 12, p-value < 2.2e-16
```

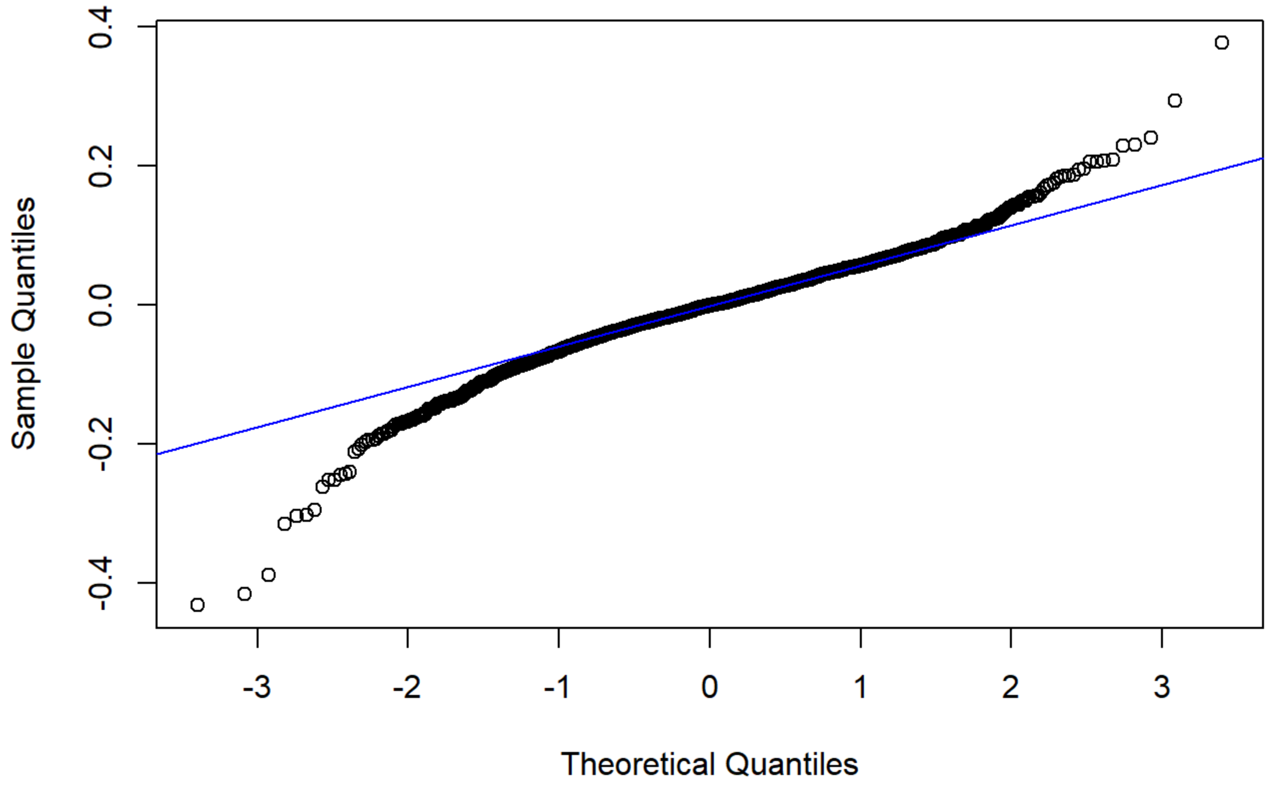


```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.005368
```

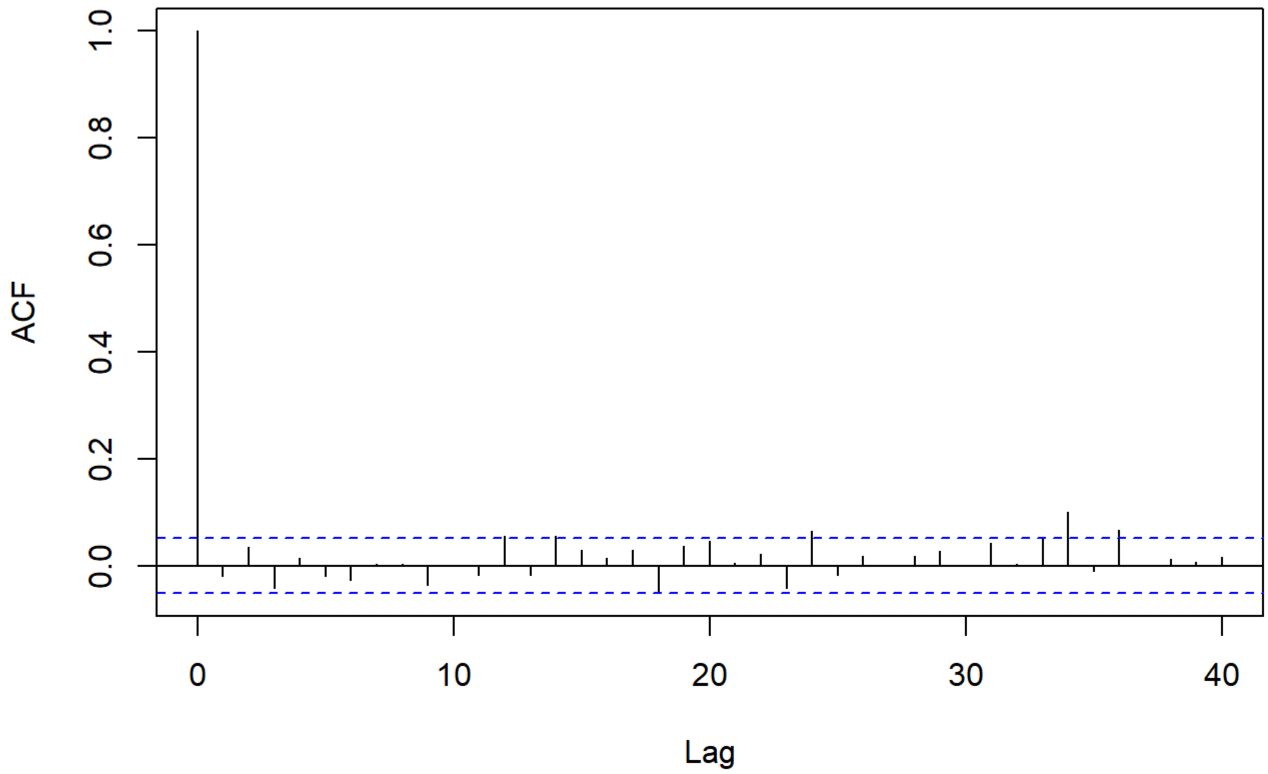

Histogram of res2
Density
0
1
2
3
4
5
6
-0.4
-0.2
0.0
0.2
0.4

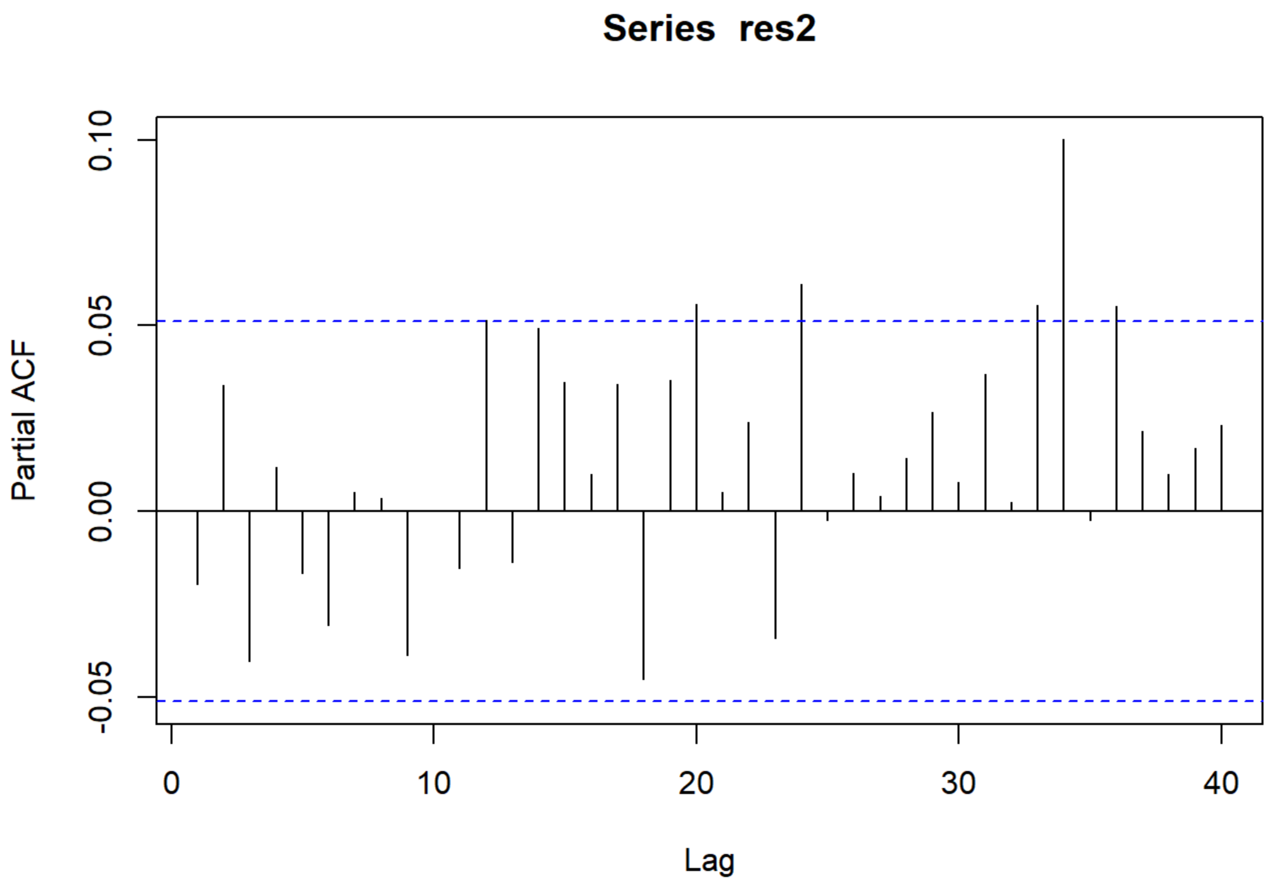

res2
-0.4
-0.2
0.0
0.2
0.4
0
500
1000
1500
Time

Normal Q-Q Plot for Model B



Series res2



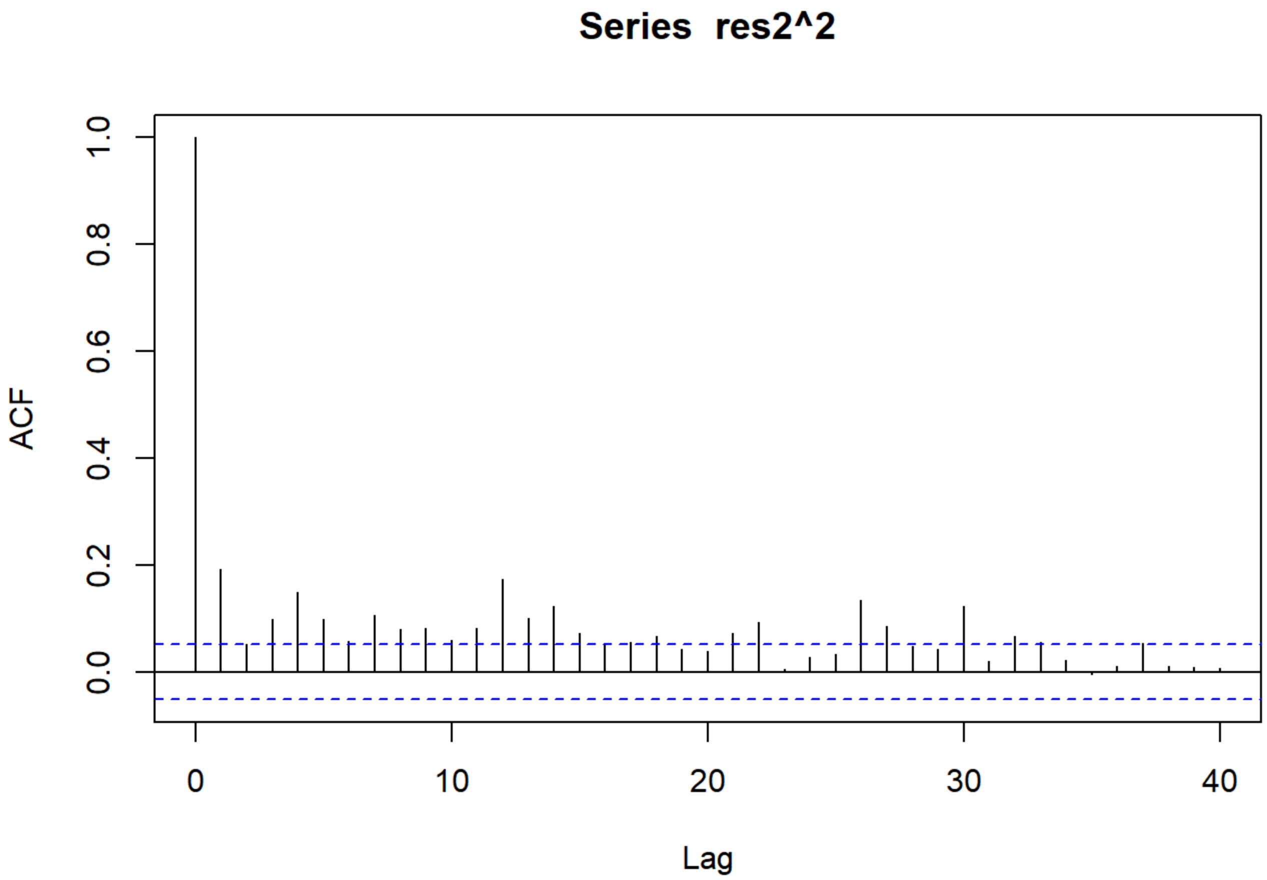


```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.9605, p-value < 2.2e-16
```

```
##
##  Box-Pierce test
##
## data:  res2
## X-squared = 70.33, df = 32, p-value = 0.0001074
```

```
##
##  Box-Ljung test
##
## data:  res2
## X-squared = 71.566, df = 32, p-value = 7.436e-05
```

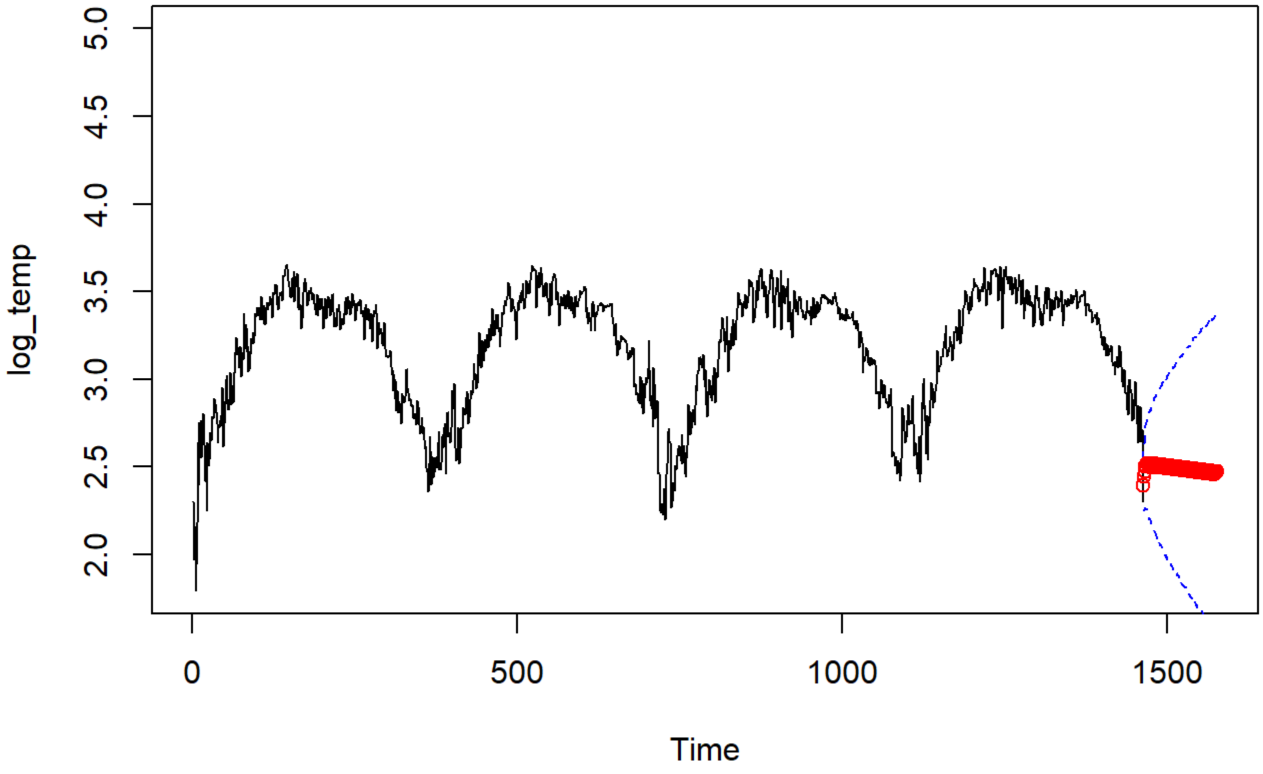
```
##
##  Box-Ljung test
##
## data:  res2^2
## X-squared = 218.9, df = 12, p-value < 2.2e-16
```



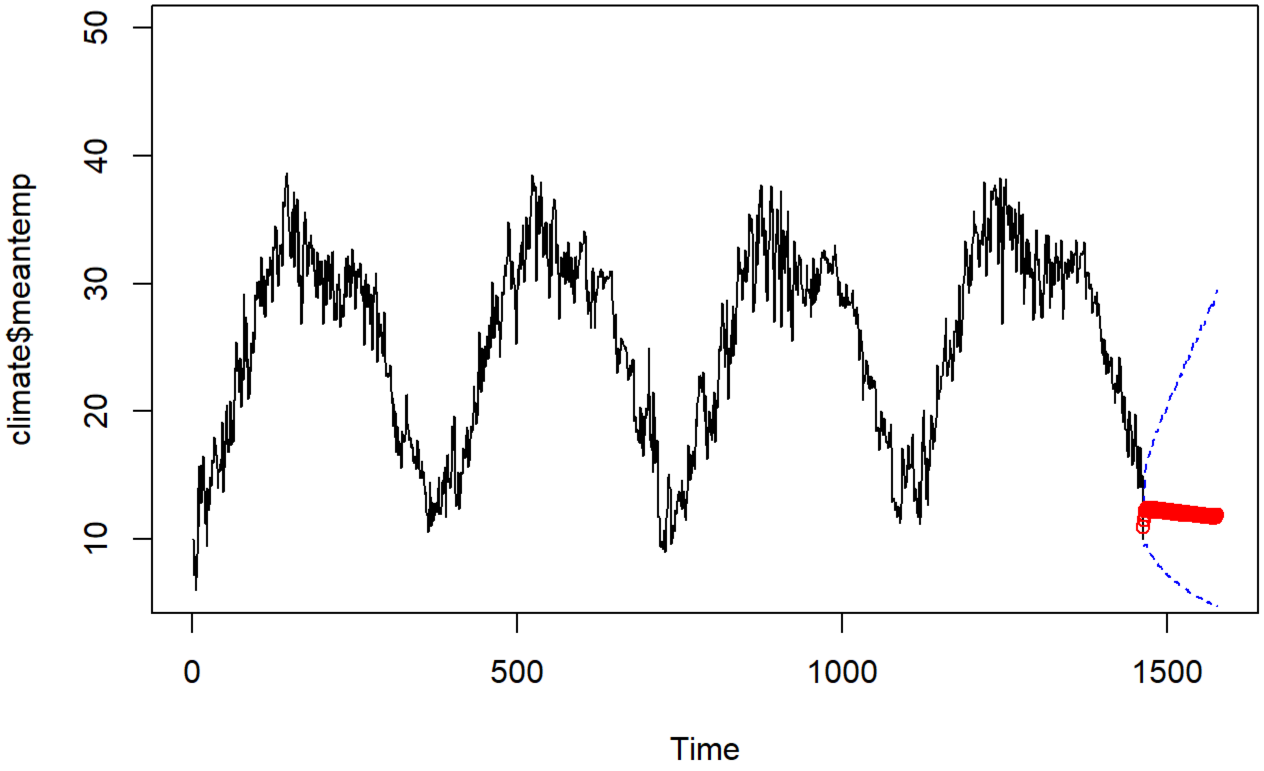
```
##
## Call:
## ar(x = res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.005368
```

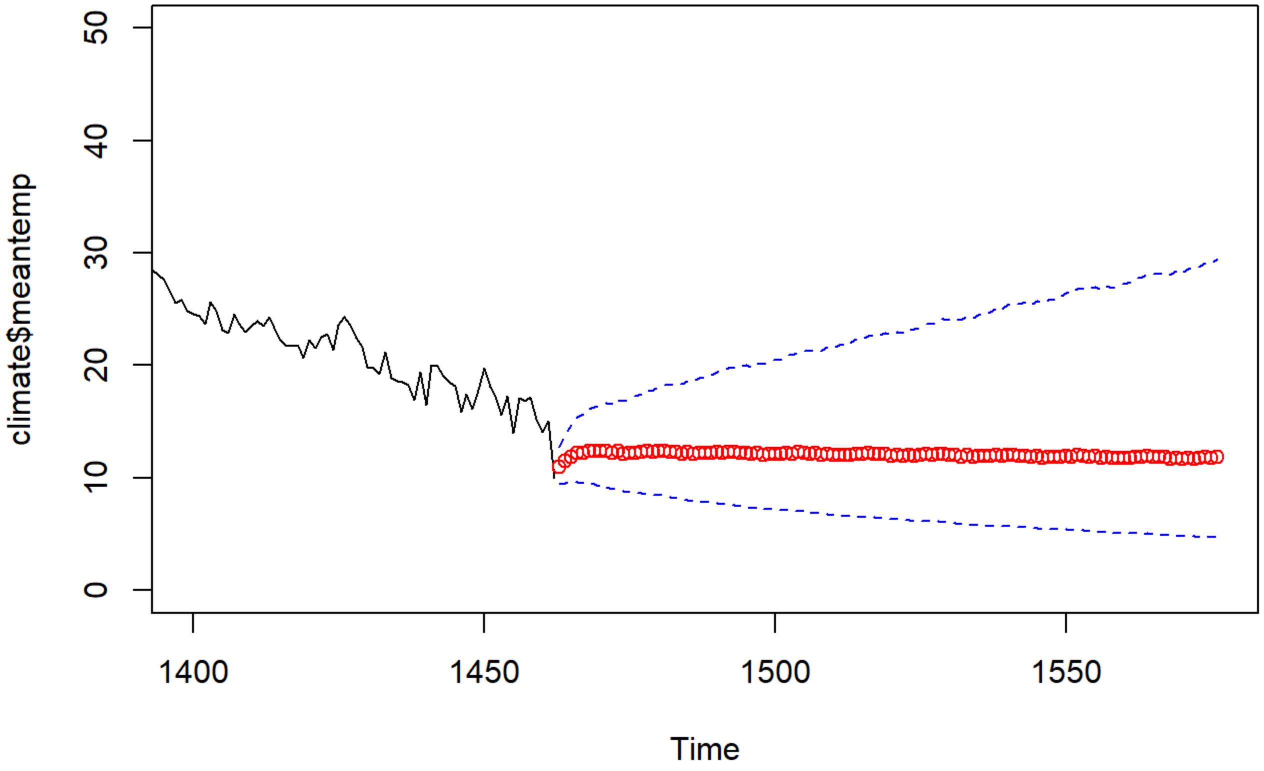
For models A and B, there is no visible trend, the histogram of residuals is fairly normal with sample mean almost 0, 95% of the data definitely resides between positive and negative 2 when looking at the QQ plot, sample ACF and PACF look good, but we cannot entirely attribute the statistically significant ACFs and PACFs to white noise. Residuals squared show nonlinear dependence in the plot as well as in the McLeod Li test. Shapiro-Wilk test was failed, which is to be expected with highly seasonal data such as temperatures. Unfortunately the Box-Pierce and Box-Ljung tests were not passed. Both models had order selected 0, so they passed the yule-walker test. Since both my models A and B passed and failed the same diagnostic tests, I will be using model A for forecasting since it had a lower AIC score.

```
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1463       2.394998 2.300430 2.489567 2.250368 2.539628
## 1464       2.443417 2.322194 2.564640 2.258022 2.628811
## 1465       2.473479 2.335969 2.610989 2.263176 2.683782
## 1466       2.499865 2.350413 2.649317 2.271298 2.728433
## 1467       2.502048 2.342858 2.661239 2.258588 2.745509
## 1468       2.518959 2.351287 2.686631 2.262527 2.775391
## 1469       2.519187 2.343817 2.694557 2.250982 2.787392
## 1470       2.516122 2.333585 2.698660 2.236956 2.795289
## 1471       2.517408 2.328089 2.706727 2.227869 2.806947
## 1472       2.504583 2.308781 2.700385 2.205129 2.804036
## 1473       2.513866 2.311825 2.715907 2.204871 2.822861
## 1474       2.498090 2.290018 2.706163 2.179871 2.816310
## 1475       2.505531 2.290989 2.720073 2.177417 2.833644
## 1476       2.504178 2.283484 2.724872 2.166656 2.841700
## 1477       2.506216 2.279609 2.732822 2.159651 2.852781
## 1478       2.514281 2.281954 2.746608 2.158967 2.869595
## 1479       2.508997 2.271110 2.746884 2.145180 2.872814
## 1480       2.517733 2.274427 2.761040 2.145628 2.889838
## 1481       2.514736 2.266135 2.763336 2.134534 2.894938
## 1482       2.509362 2.255583 2.763142 2.121240 2.897485
## 1483       2.510992 2.252140 2.769845 2.115111 2.906874
## 1484       2.498284 2.234457 2.762112 2.094796 2.901773
## 1485       2.506476 2.237767 2.775184 2.095522 2.917430
## 1486       2.495296 2.221793 2.768798 2.077010 2.913581
## 1487       2.501557 2.223070 2.780044 2.075647 2.927467
## 1488       2.499514 2.216183 2.782845 2.066197 2.932831
## 1489       2.501162 2.213100 2.789223 2.060609 2.941714
## 1490       2.508975 2.216277 2.801673 2.061332 2.956618
## 1491       2.503584 2.206332 2.800835 2.048977 2.958190
## 1492       2.512210 2.210480 2.813941 2.050753 2.973667
## 1493       2.509168 2.203027 2.815309 2.040966 2.977370
## 1494       2.503763 2.193277 2.814250 2.028915 2.978611
## 1495       2.505396 2.190624 2.820167 2.023995 2.986797
## 1496       2.492688 2.173690 2.811685 2.004823 2.980553
## 1497       2.500865 2.177697 2.824034 2.006622 2.995109
## 1498       2.489741 2.162455 2.817027 1.989200 2.990282
## 1499       2.495988 2.164404 2.827571 1.988875 3.003101
## 1500       2.493936 2.158156 2.829717 1.980405 3.007468
## 1501       2.495579 2.155680 2.835478 1.975748 3.015410
## 1502       2.503389 2.159437 2.847342 1.977359 3.029420
## 1503       2.497996 2.150046 2.845947 1.965853 3.030140
## 1504       2.506622 2.154725 2.858519 1.968441 3.044802
## 1505       2.503579 2.147781 2.859377 1.959434 3.047725
## 1506       2.498174 2.138520 2.857828 1.948130 3.048217
## 1507       2.499806 2.136337 2.863275 1.943929 3.055684
## 1508       2.487098 2.119855 2.854342 1.925449 3.048748
## 1509       2.495276 2.124297 2.866255 1.927912 3.062639
## 1510       2.484152 2.109475 2.858830 1.911133 3.057172
## 1511       2.490399 2.111854 2.868944 1.911464 3.069333
## 1512       2.488347 2.106014 2.870680 1.903620 3.073075
## 1513       2.489990 2.103930 2.876050 1.899562 3.080418
## 1514       2.497800 2.108063 2.887537 1.901748 3.093852
## 1515       2.492407 2.099035 2.885779 1.890797 3.094018
## 1516       2.501033 2.104064 2.898002 1.893921 3.108144
## 1517       2.497990 2.097459 2.898521 1.885430 3.110549
## 1518       2.492584 2.088524 2.896645 1.874627 3.110541
## 1519       2.494217 2.086659 2.901775 1.870910 3.117523
## 1520       2.481509 2.070483 2.892535 1.852899 3.110119
## 1521       2.489686 2.075222 2.904150 1.855818 3.123555
## 1522       2.478563 2.060689 2.896437 1.839480 3.117646
## 1523       2.484809 2.063366 2.906253 1.840267 3.129352
## 1524       2.482758 2.057812 2.907704 1.832859 3.132657
## 1525       2.484401 2.056002 2.912799 1.829222 3.139580
## 1526       2.492211 2.060401 2.924020 1.831815 3.152607
## 1527       2.486818 2.051631 2.922005 1.821257 3.152379
## 1528       2.495443 2.056909 2.933977 1.824763 3.166123
## 1529       2.492401 2.050547 2.934254 1.816644 3.168157
## 1530       2.486995 2.041848 2.932142 1.806202 3.167788
## 1531       2.488628 2.040213 2.937043 1.802836 3.174419
## 1532       2.475920 2.024261 2.927579 1.785167 3.166673
## 1533       2.484097 2.029217 2.938977 1.788418 3.179776
## 1534       2.472974 2.014896 2.931052 1.772404 3.173544
## 1535       2.479220 2.017793 2.940647 1.773528 3.184912
## 1536       2.477169 2.012450 2.941887 1.766443 3.187894
## 1537       2.478811 2.010845 2.946778 1.763119 3.194504
## 1538       2.486622 2.015443 2.957800 1.766016 3.207227
## 1539       2.481229 2.006866 2.955592 1.755753 3.206704
## 1540       2.489854 2.012332 2.967376 1.759547 3.220161
## 1541       2.486811 2.006153 2.967469 1.751708 3.221914
## 1542       2.481406 1.997633 2.965178 1.741540 3.221272
## 1543       2.483038 1.996172 2.969905 1.738441 3.227636
## 1544       2.470331 1.980390 2.960271 1.721032 3.219629
## 1545       2.478508 1.985513 2.971502 1.724538 3.232478
## 1546       2.467384 1.971354 2.963415 1.708772 3.225997
## 1547       2.473631 1.974421 2.972841 1.710155 3.237107
## 1548       2.471579 1.969242 2.973917 1.703321 3.239838
## 1549       2.473222 1.967795 2.978649 1.700238 3.246206
## 1550       2.481032 1.972547 2.989517 1.703371 3.258693
## 1551       2.475639 1.964120 2.987158 1.693339 3.257940
## 1552       2.484265 1.969734 2.998796 1.697357 3.271172
## 1553       2.481222 1.963698 2.998746 1.689738 3.272706
## 1554       2.475817 1.955319 2.996315 1.679784 3.271850
## 1555       2.477449 1.953994 3.000904 1.676894 3.278004
## 1556       2.464741 1.938347 2.991136 1.659691 3.269792
## 1557       2.472919 1.943601 3.002236 1.663398 3.282440
## 1558       2.461795 1.929571 2.994020 1.647828 3.275762
## 1559       2.468042 1.932772 3.003311 1.649418 3.286665
## 1560       2.465990 1.927724 3.004256 1.642783 3.289197
## 1561       2.467633 1.926403 3.008862 1.639893 3.295372
## 1562       2.475443 1.931278 3.019608 1.643215 3.307671
## 1563       2.470050 1.922972 3.017128 1.633366 3.306734
## 1564       2.478675 1.928703 3.028648 1.637565 3.319786
## 1565       2.475633 1.922783 3.028483 1.630122 3.321144
## 1566       2.470227 1.914516 3.025939 1.620340 3.320114
## 1567       2.471860 1.913303 3.030417 1.617620 3.326099
## 1568       2.459152 1.897764 3.020541 1.600582 3.317722
## 1569       2.467329 1.903124 3.031535 1.604452 3.330207
## 1570       2.456206 1.889198 3.023214 1.589042 3.323370
## 1571       2.462452 1.892509 3.032396 1.590799 3.334105
## 1572       2.460401 1.887566 3.033235 1.584326 3.336476
## 1573       2.462043 1.886348 3.037738 1.581594 3.342493
## 1574       2.469854 1.891324 3.048383 1.585069 3.354638
## 1575       2.464461 1.883116 3.045805 1.575371 3.353550
## 1576       2.473086 1.888944 3.057229 1.579717 3.366455
```



Looking at the plot of the forecast, the forecasted points all seem to follow the previously observed seasonality.


climate$meantemp
10
20
30
40
50
0
500
1000
1500
Time


climate$meantemp
0
10
20
30
40
50
1400
1450
1500
1550
Time

Looking at the plot of the forecast, the forecasted points all seem to follow the previously observed seasonality and trend. Our model performed decently.

# Conclusion

In this project, I attempted to forecast the temperature in the city of Delhi based on data about previous daily temperatures in the city. Many techniques were used in this project, such as transforming data, differencing data, and testing many models until finding one that fits best. There are many plots for visualization. In the end, the forecasting model was not the most precise, but the techniques used to get achieve a model are much more important than the end results. All material used in this project was learned from Professor Feldman. Thank you.

# Appendix

```r
climate <- read.csv("C:/Users/Ethan Tran/PSTAT174/Project/DailyDelhiClimateTrain.csv")
climate_test <- read.csv("C:/Users/Ethan Tran/PSTAT174/Project/DailyDelhiClimateTest.csv")
#reading in the datasets, already partitioned into training and testing sets in Kaggle

climate$date <- as.Date(climate$date)
climate_test$date <- as.Date(climate_test$date)
#changing the dates to date objects

climate2 <- rbind(climate, climate_test)
```

```r
temp_ts <- ts(climate$meantemp, frequency = 1)
#creating time series of mean daily temperature in Delhi

plot(1:length(temp_ts),temp_ts, main =
"Time Series of Temperature from Year 2013 to 2017", type = 'l',xlab='index')
index = 1: length(temp_ts)
trend <- lm(temp_ts ~ index)
abline(trend, col="red")
abline(h=mean(temp_ts) , col='blue')
#plotting the average daily temperature in Delhi from 2013 to 2017
```

```r
hist(climate$meantemp, col="light blue", xlab="", main="Histogram of Temperature Time Series")
#plotting histogram

acf(climate$meantemp, main = "ACF of Temperature Time Series")
#plotting ACF

var(climate$meantemp)
#getting the variance of the time series
```

```
log_temp <- log(temp_ts)
plot(1:length(log_temp),log_temp, main =
"Time Series of Log Temperature from Year 2013 to 2017", type = 'l',xlab='index')
index = 1: length(log_temp)
trend <- lm(log_temp ~ index)
abline(trend, col="red")
abline(h=mean(log_temp) , col='blue')
#plotting the log average daily temperature in Delhi from 2013 to 2017
var(log_temp)
#getting the variance after log transform
```

```
temp_diff_12 <- diff(log_temp, lag=12)
plot.ts(temp_diff_12, main = "Temperature Time Series Differenced at Lag 12")
#differencing and plotting at lag 12
fit <- lm(temp_diff_12 ~ as.numeric(1:length(temp_diff_12))); abline(fit, col="red")
abline(h=mean(temp_diff_12), col="blue")
var(temp_diff_12)
#variance after differencing at 12
```

```
temp_diff_1 <- diff(temp_diff_12, lag=1)
plot.ts(temp_diff_1)
fit <- lm(temp_diff_1 ~ as.numeric(1:length(temp_diff_1))); abline(fit, col="red", lwd=2, lty=2)
abline(h=mean(temp_diff_1), col="blue")
#differencing and plotting at lag 12 and then 1
var(temp_diff_1)
#variance after differencing at lag 1
```

```
acf(temp_diff_12, lag.max = 40, main = "ACF of Temp Differenced at Lag 12")
acf(temp_diff_1, lag.max = 40, main = "ACF of Temp Differenced at Lag 12 and 1")
```

```
hist(temp_diff_1, density=20,breaks=20, col="blue", xlab="", prob=TRUE, main = "Histogram of Log
Temp Diff at Lag 12 and 1")
m<-mean(temp_diff_1)
std<- sqrt(var(temp_diff_1))
curve( dnorm(x,m,std), add=TRUE )
```

```
acf(temp_diff_1, lag.max = 40, main = "ACF of Temp Differenced at Lag 12 and 1")
pacf(temp_diff_1, lag.max = 80, main = "PACF of Temp Differenced at Lag 12 and 1")
```

```
arima(log_temp, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 12), method =
"ML")
#SARIMA (1, 1, 1) x (1, 1, 1) x 12, best
arima(log_temp, order = c(3, 1, 1), seasonal = list(order = c(1, 1, 1), period = 12), method =
"ML")
#SARIMA (3, 1, 1) x (1, 1, 1) x 12, 2nd best
```

```r
source("plot.roots.R")
plot.roots(NULL,polyroot(c(1, -0.5546, 0.0425, -0.044)), main="(B) roots of ar part, nonseasonal")
polyroot(c(1, -0.5546, 0.0425, -0.044))
#finding roots of non seasonal ar polynomial
```

```r
#Diagnostics for Model A
fit <- arima(log_temp, order=c(1,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")
res <- residuals(fit)
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res)
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model B")
qqline(res,col="blue")
acf(res, lag.max=40)
pacf(res, lag.max=40)
shapiro.test(res)
Box.test(res, lag = 38, type = c("Box-Pierce"), fitdf = 4)
Box.test(res, lag = 38, type = c("Ljung-Box"), fitdf = 4)
Box.test(res^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(res^2, lag.max=40)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

#Diagnostics for Model B
fit2 <- arima(log_temp, order=c(3,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")
res2 <- residuals(fit)
hist(res2,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m2 <- mean(res2)
std2 <- sqrt(var(res2))
curve( dnorm(x,m2,std2), add=TRUE )
plot.ts(res2)
fitt2 <- lm(res2 ~ as.numeric(1:length(res2))); abline(fitt, col="red")
abline(h=mean(res2), col="blue")
qqnorm(res2,main= "Normal Q-Q Plot for Model B")
qqline(res2,col="blue")
acf(res2, lag.max=40)
pacf(res2, lag.max=40)
shapiro.test(res2)
Box.test(res2, lag = 38, type = c("Box-Pierce"), fitdf = 6)
Box.test(res2, lag = 38, type = c("Ljung-Box"), fitdf = 6)
Box.test(res2^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(res2^2, lag.max=40)
ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
fit_A <- arima(log_temp, order=c(1,1,1), seasonal = list(order = c(1,1,1), period = 12), method
="ML")
forecast(fit_A, h = 114)
```

```
pred_tr <- predict(fit_A, n.ahead = 114)
U_tr = pred_tr$pred + 2*pred_tr$se
L_tr = pred_tr$pred - 2*pred_tr$se
ts.plot(log_temp, xlim=c(1,length(log_temp)+114), ylim = c(min(log_temp),5))
lines(U_tr, col="blue", lty="dashed")
lines(L_tr, col="blue", lty="dashed")
points((length(log_temp)+1):(length(log_temp)+114), pred_tr$pred, col="red")
```

```
pred.orig <- exp(pred_tr$pred)
U= exp(U_tr)
L= exp(L_tr)
ts.plot(climate$meantemp, xlim=c(1,length(climate$meantemp)+114), ylim = c(min(climate$meantem
p),50))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(climate$meantemp)+1):(length(climate$meantemp)+114), pred.orig, col="red")

ts.plot(climate$meantemp, xlim = c(1400,length(climate$meantemp)+114), ylim = c(0,50))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(climate$meantemp)+1):(length(climate$meantemp)+114), pred.orig, col="red")
```