Times Series: Popcorn Interest (2004 - 2021)

Jiaying Teng

December 3rd, 2021

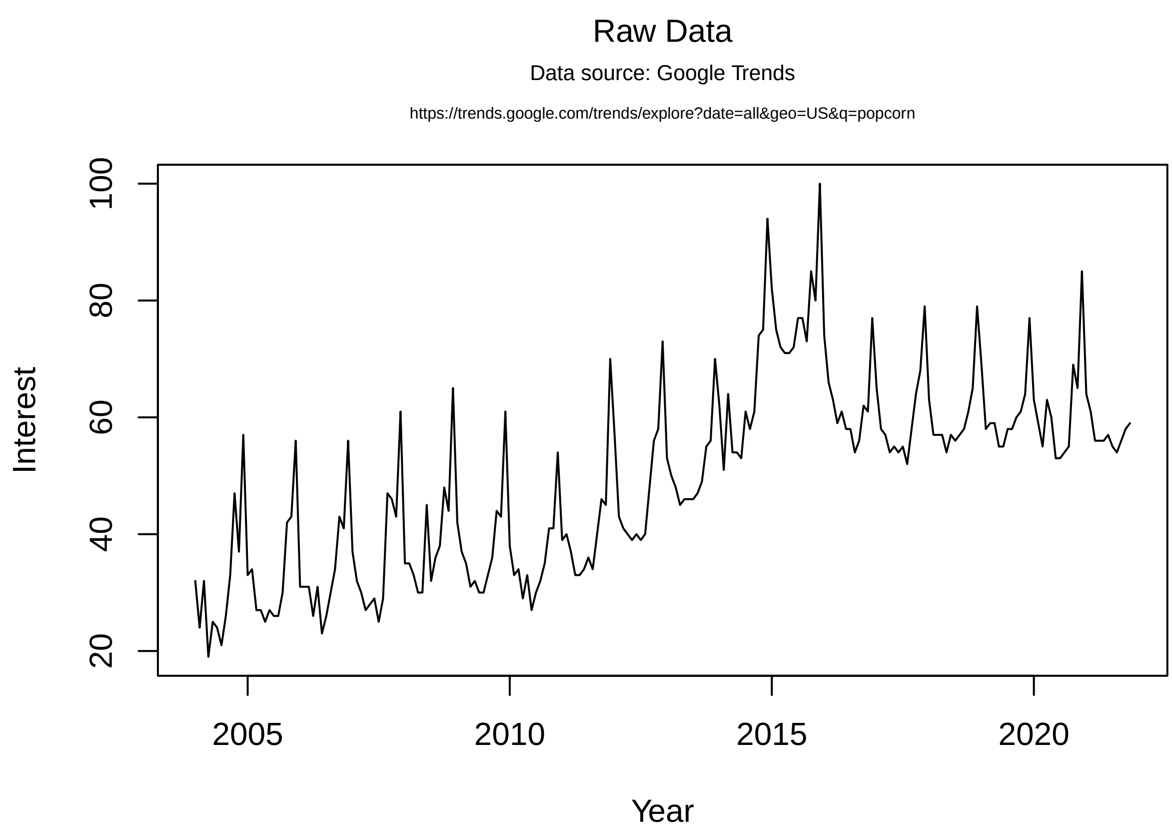
EXECUTIVE SUMMARY

A statistical analysis of the interest of Popcorn in the United States was done on a monthly data set ranging from 2004-2020 in an effort to predict and forecast upcoming trends of 2021. The results indicate that the data is not normal and thus, there may exists extraneous errors in the final forecast. An annual seasonal spike around the months of November-December is heavily suggested and is expected to continue for 2021. The interest of popcorn stayed steady across the years, discounting for an unusual rise during 2014-2015, and is expected to continue to have steady rise in the upcoming years.

INTRODUCTION

In an attempt to forecast the interest of popcorn in the United States, a time series analysis will be done on a Google Trends data set that measures popcorn interest in the US during 2004-2021. This report will attempt to analyze and provide reasonable hypothesis to when popcorn is popular throughout the years. Using R version 4.1.2 alongside RStudio, classical time series techniques such as transformation of the data set and differencing to remove seasonality will be used to forecast the results. The main goal of this report is to produce a model that accurately forecasts seasonal and non-seasonal trend of popcorn for the upcoming years.

To tackle the questions above, the criteria for how interest of popcorn is measured must be defined precisely. For our specific purposes, Google Trends carefully defines interest as absolute search volume for a term, relative to the number of searches received by Google. In other words, our data set will measure interest of popcorn as total number of searches for the exact term "popcorn" on Google relative to all other Google searches for a particular month.

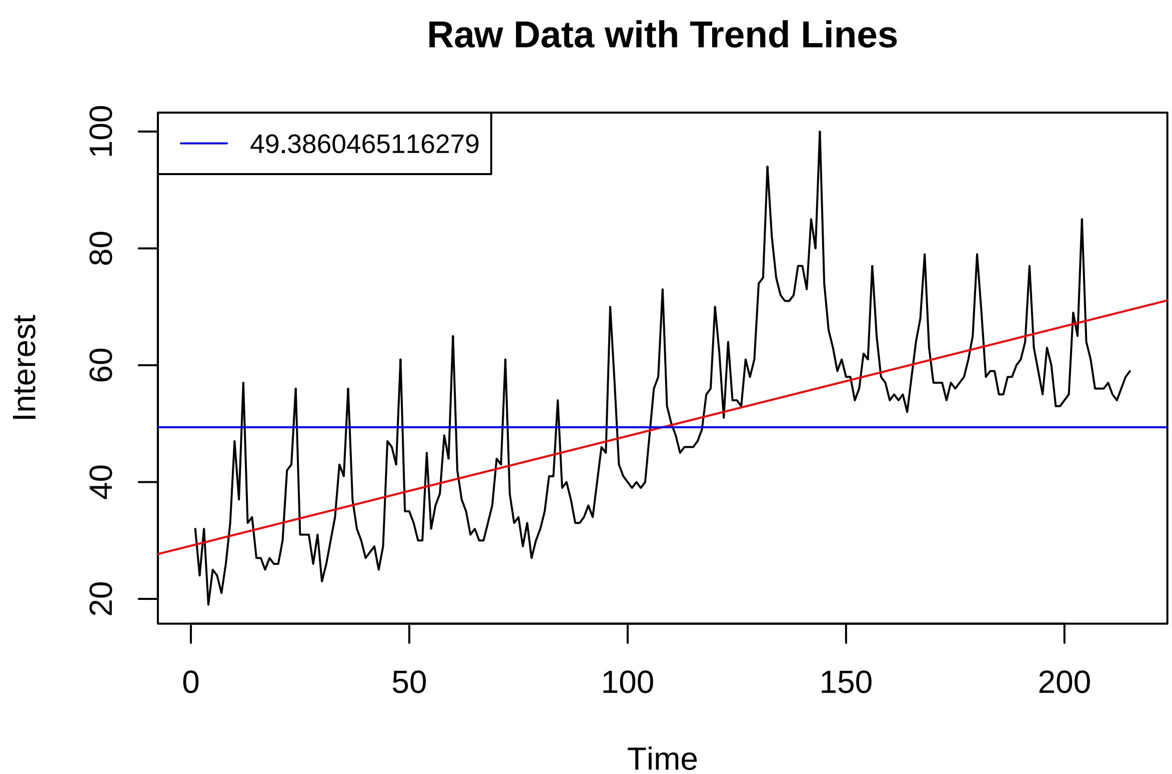


Analyzing the Time Series

Looking at the Raw Data, we can see a clear upward trend with distinct seasonality every year. The variance of the graph looks relatively stable except for the time period 2014-2016 where there exists a massive spike. Following 2016, the data shows consistent variance for the rest of the years.

The data can be divided into three time periods to examine three particular trends. For the time period 2004-2010, the interest of popcorn seem to stay relatively consistent and there may exists a slight upward trend. But during the years 2010-2016, there is clearly a massive upward trend followed by a sharp downward trend. During the years 2016-2020, the raw data shows little to no evidence of trend.

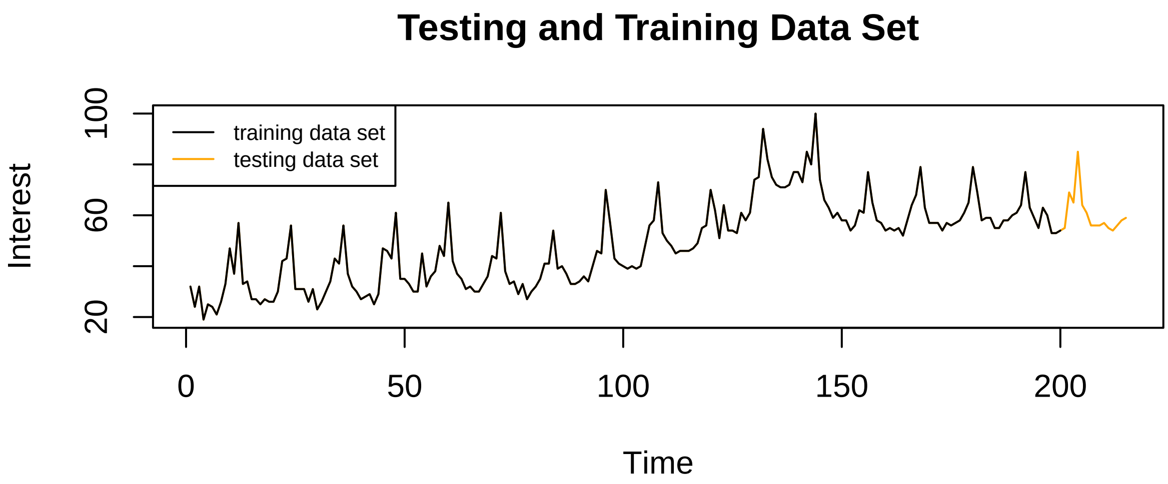
From the raw data, we are most concerned about the time period 2010-2016 as it shows an abnormal spike in comparisons to every other time period. This sharp change in behavior is the main contributor for the overall upward trend that the graph displays.



From the red trend line, we can see that there is a clear positive trend. Therefore, to eliminate trend and make the original data set more stable and consistent over the years, we will perform differencing.

Creating a Training Data Set

Before differencing however, we first must divide the data set into a test and training data set to evaluate the accuracy of our model for forecasting. We will divide the 215 observations into 200 observations for the training data set and 15 observations for the testing data set. We will produce a model using the training data set and test the model's accuracy by comparing its forecast results to the testing data set.

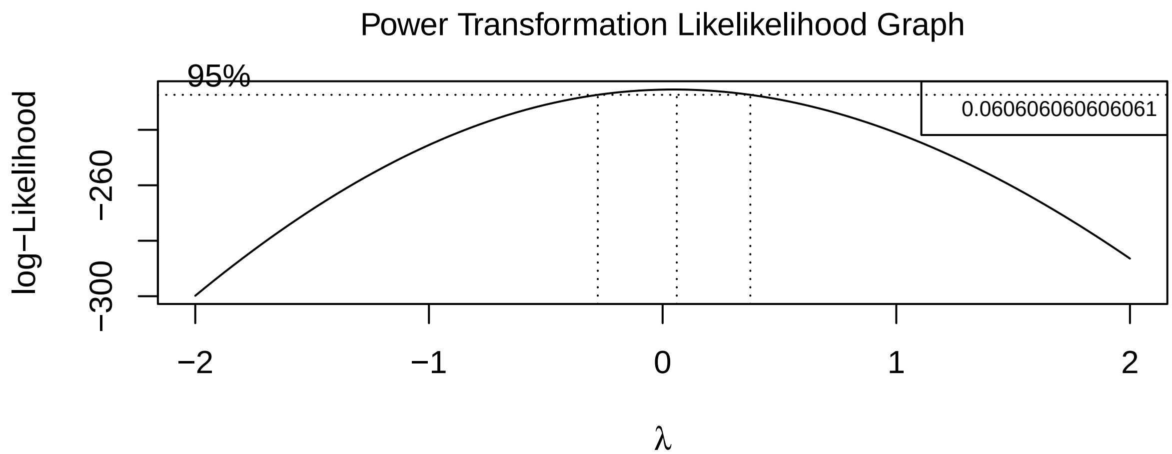


Using the training data set, we will produce a model that will be judged on its validity based on how accurately it forecasts the testing data set.

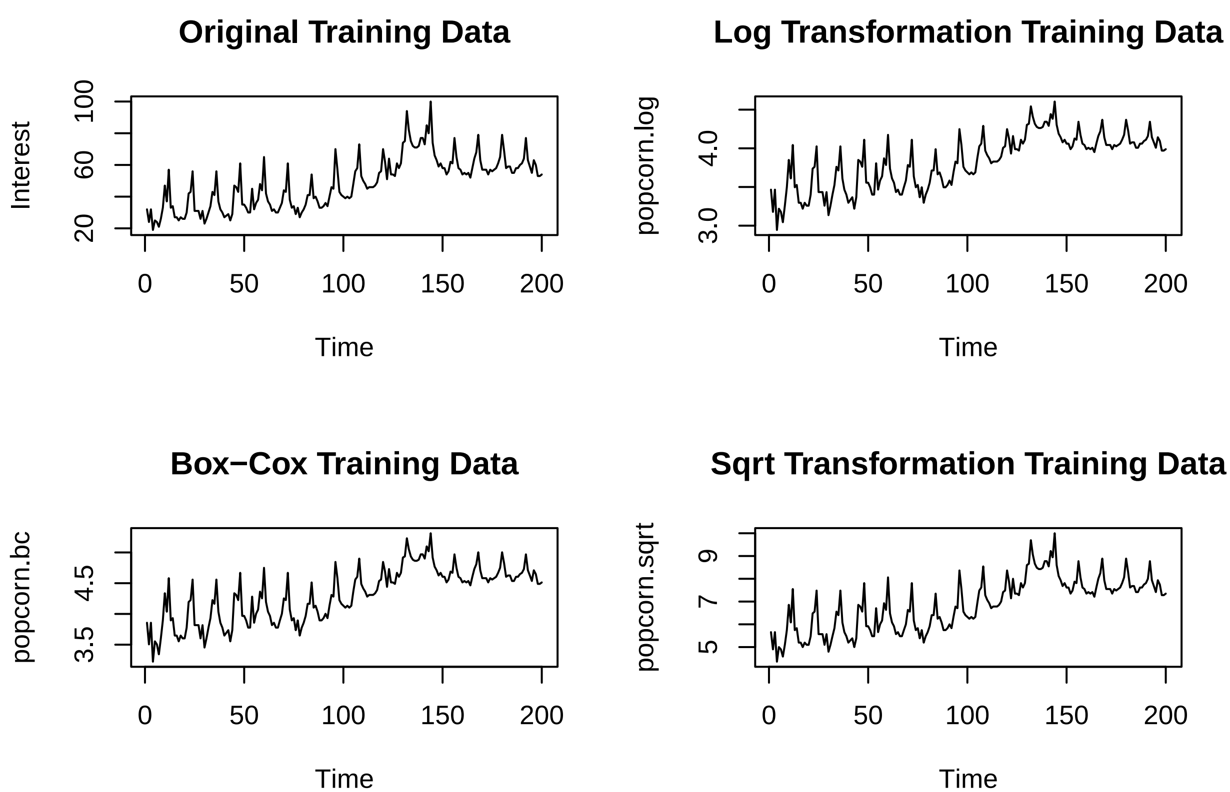
To begin, we first confirm our assumptions that the data is highly non-stationary and validate the possible existence of non-constant variance and mean.

From the graphs above, we can see that the histogram is right-skewed and that the ACF remains large and periodic. Thus, to deal with these two issues, we will perform transformation to stabilize variance and differencing to remove seasonality and trend.

Transformation on the Data



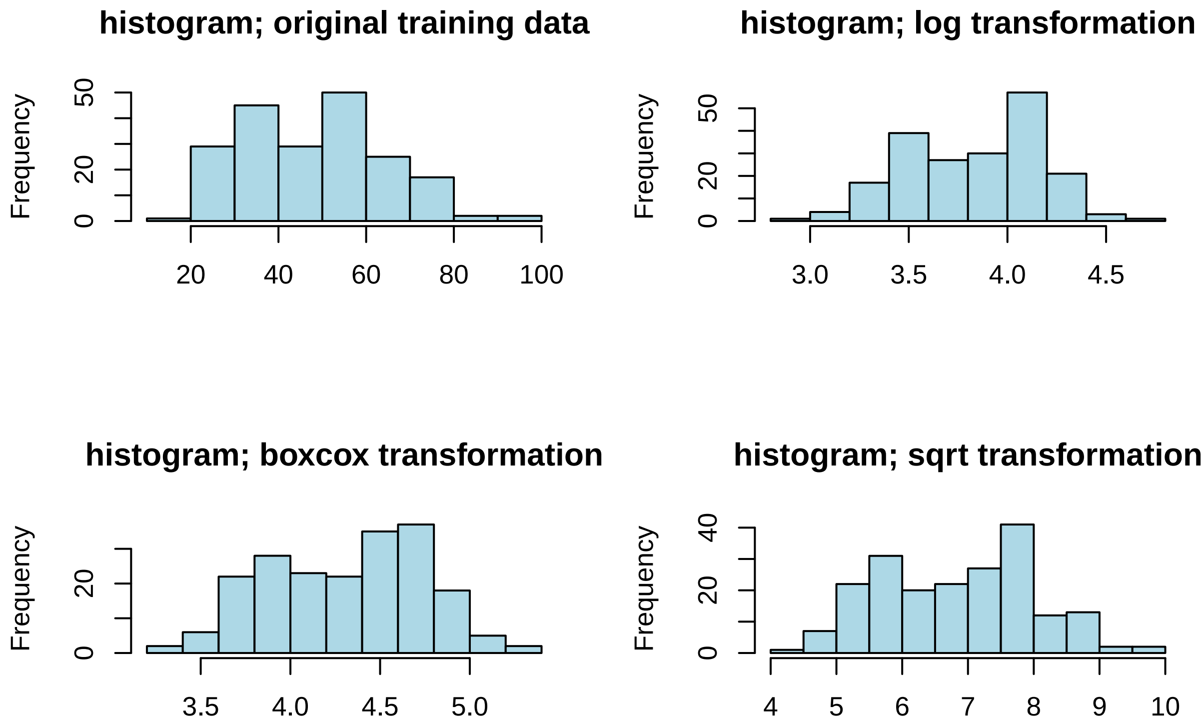
The box cox transformation returned $\lambda = 0.060606$. Since $\lambda = 0(log)$ is also in the confidence interval and is extremely close to λ, it may be most appropriate to perform log transformation on our data set.




```
> Variance of Original Training Data:  261.383316582915
> Variance of Sqrt Transformation:  1.35827248503342
```

```
> Variance of Box Cox Transformation:  0.18945059406569
> Variance of Log Transformation:  0.119668644068251
```

From the transformation graphs above, we see that the log transformation provides the lowest variance out of the three transformations. Thus, the log transformation best stabilizes the variance of our training data set.

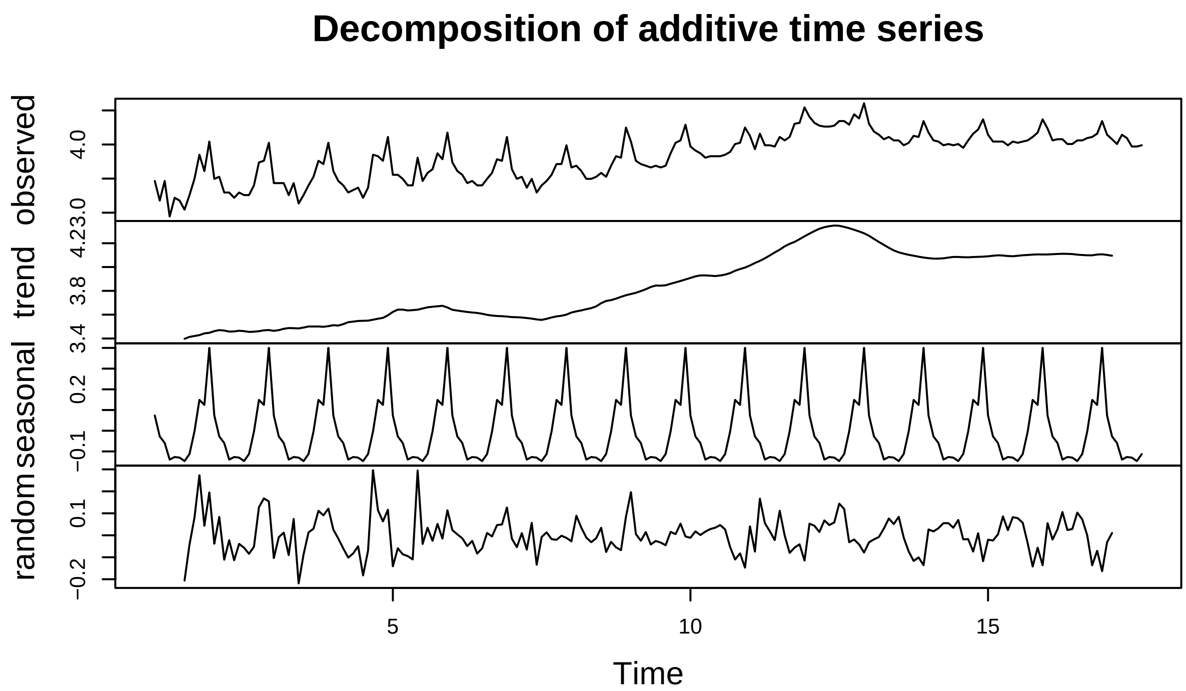


The histograms above shows that all of the transformations above fail to make the data follow a Gaussian distribution. This suggest the residuals may be non-Gaussian and thus the hypothesis that the data set above is random may be false. This suggests the possibility that no models we pick may pass the Shapiro-Wilk test during the diagnostic stage.

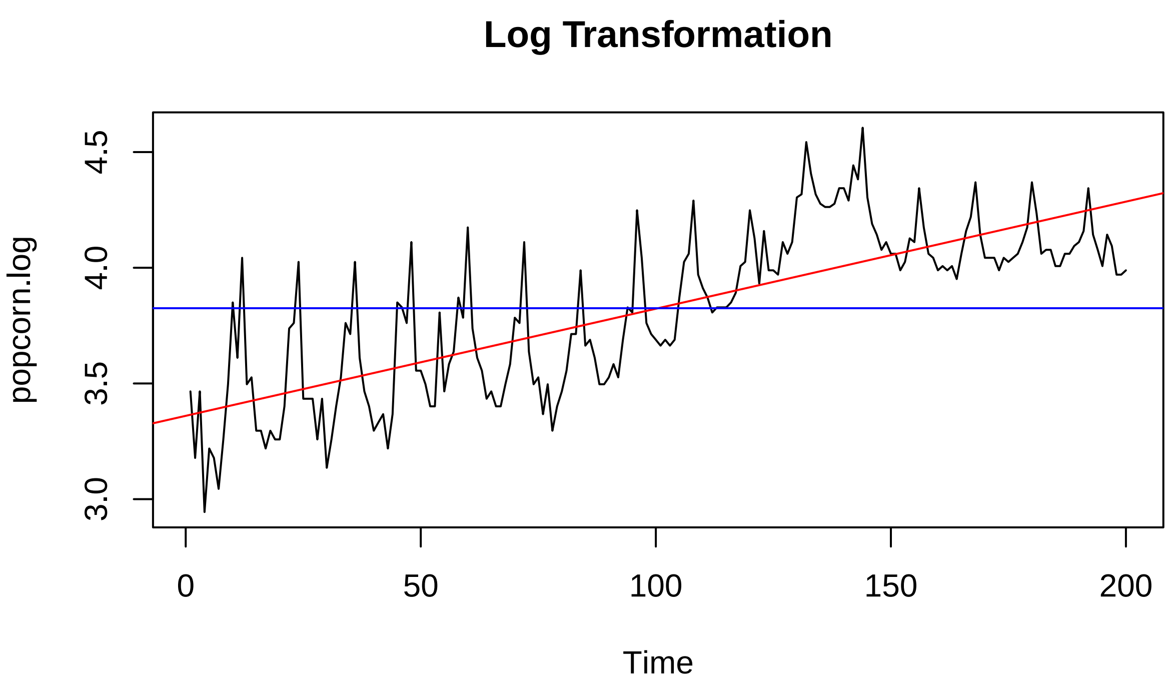
Since the log transformation and box-cox transformation are similar in their histograms, we pick the log transformation as it reduces our training model's variance more than the box-cox transformation does.

Differencing the Data

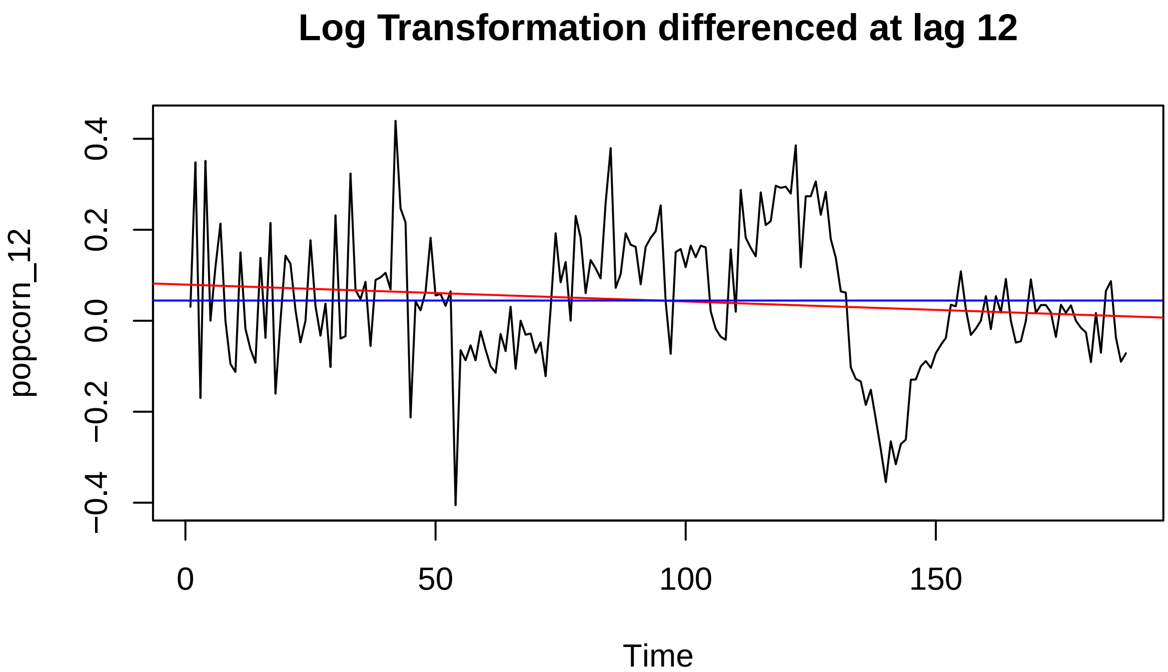
From the log transformation graph from above, we still witness a clear upward trend, suggesting non-stationarity. Thus performing differencing is required to remove possible seasonal and non-seasonal trend from our transformed data set.



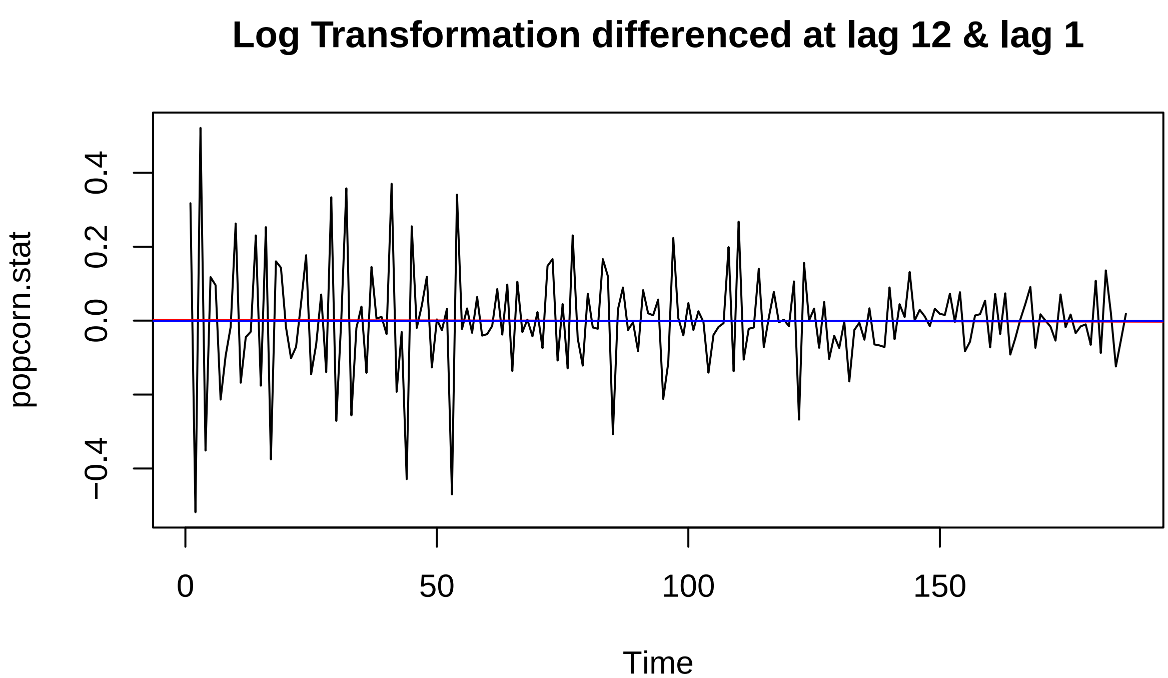
The decomposition table clearly shows that there is a upward trend from our transformed data set. Furthermore, there exists a heavy seasonal component that must be eliminated. Since our data set is a monthly data set, we will began performing differencing at lag = 12 since there are 12 months annually.



From our log transformation, we see that there still exists a clear upward trend.



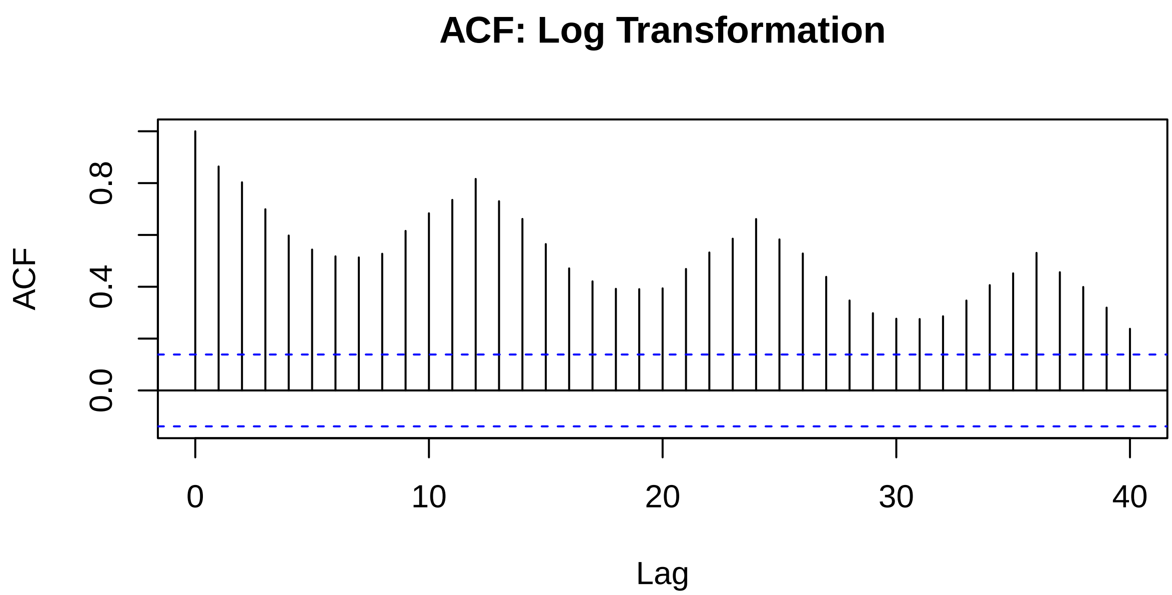
Differencing at lag = 12 significantly help reduce the trend of our data set but there still remains a slight downward trend that could be resolved if we difference again at lag = 1.



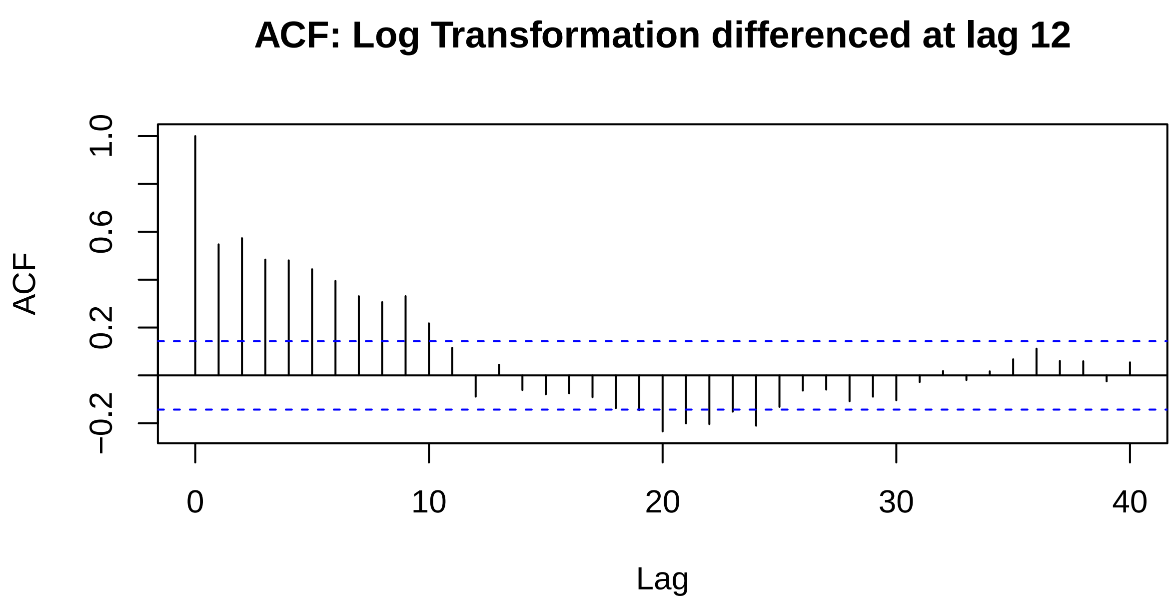
Differencing at lag = 12 and again at lag = 1 appears to resolve our problem with non-stationarity as the trend line closely follows the mean. But to assess if we over-differenced and if differencing actually made the data set stationary, we must look at the variance and ACFs of the graphs above.

```
> Variance of Log Transformation:  0.119668644068251
> Variance of Log Transformation (Differenced at Lag = 12):  0.0216011518072329
> Variance of Log Transformation (Differenced at Lag = 12 & Lag = 1):  0.0195787945498998
```

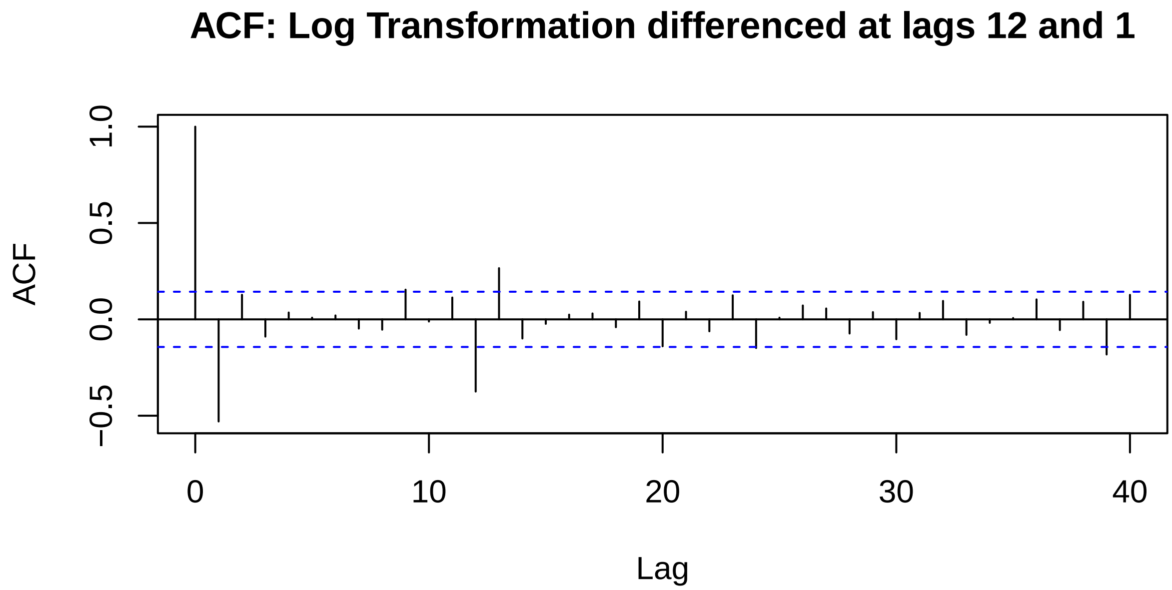
Since the variance decreased on differencing at lag = 12 and again when differenced at lag = 1, we can be sure that the model is not over differenced and proceed to analyze that ACF of our differenced and transformed data set.



The original ACF graph for log transformation portray slow decay, heavily suggesting a non-stationary process. Furthermore, consistent spike patterns at lags of 12, 24, and 36 strongly indicates seasonality.



Differencing at 12 removes the seasonality according to the ACF but there still exists a slow decay from lag 1 to lag 10 that indicates exisistence of trend.

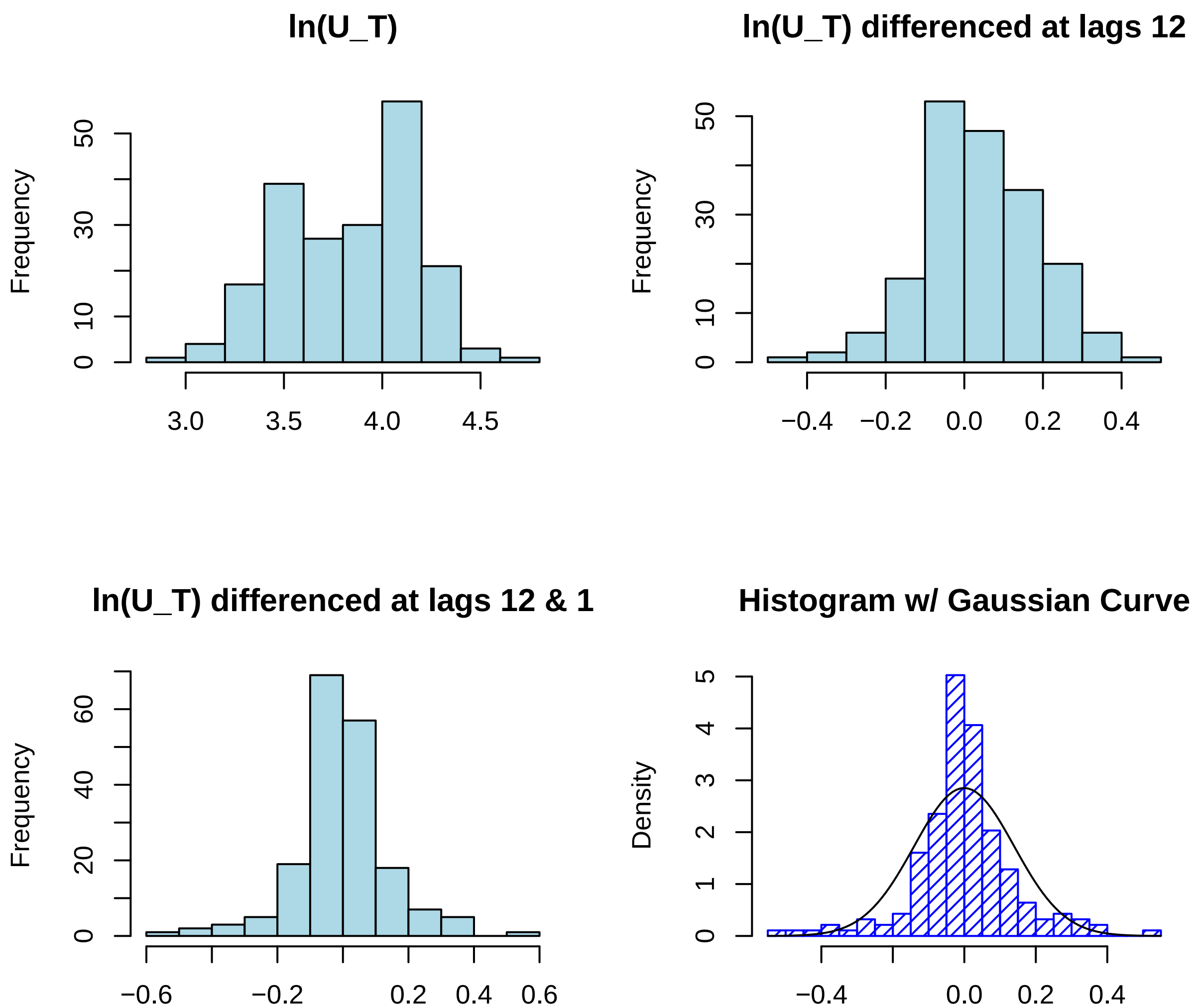


Differencing again at 1 gives us a ACF graph that corresponds well to a stationary process and thus, differencing at lag 12 and then again at lag 1 provides a indication of removal of trend, allowing for the start of identifying possible models for forecast.

Thus we will work with the data: $\nabla_1 \nabla_{12} ln(U_t)$
Where U_t is the first 200 observations of the original data.

Comparing Histograms

To see how well differencing help shape a normal distribution in our transformed data set, we look at the following histograms.



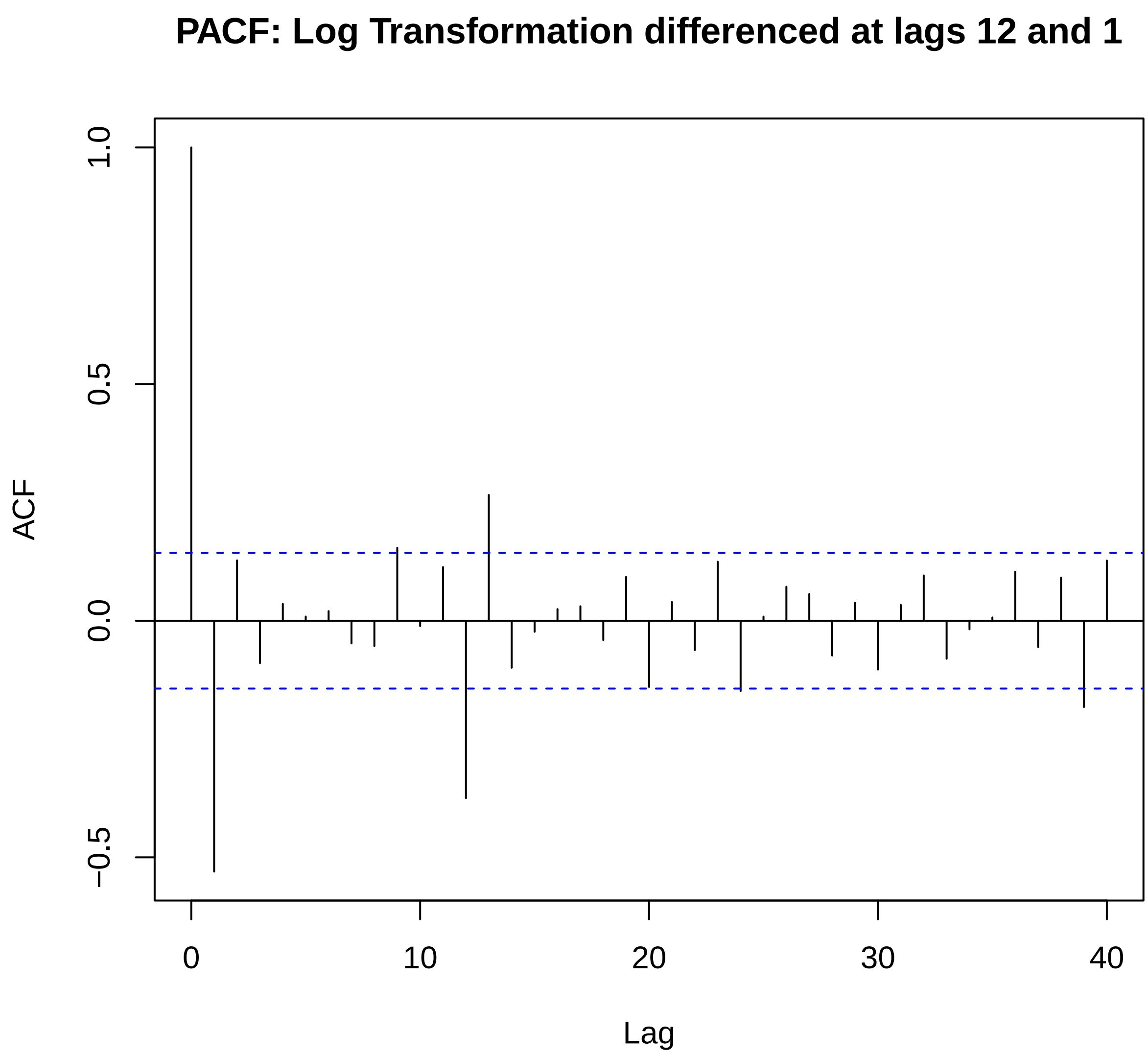
From the graphs above, it is clear that differencing at 12 & 1 help normalize our data but the histogram still shows a distribution that resembles more closely to that of a Leptokurtic Distribution rather than a Gaussian Distribution.

Thus, as stated earlier, predicted models may suffer from not having normally distributed error residuals which may suggest that the data set is not random.

Regardless, we are happy with the results of differencing at Lag = 12 and Lag = 1 on our log transformation and we move on to modeling.

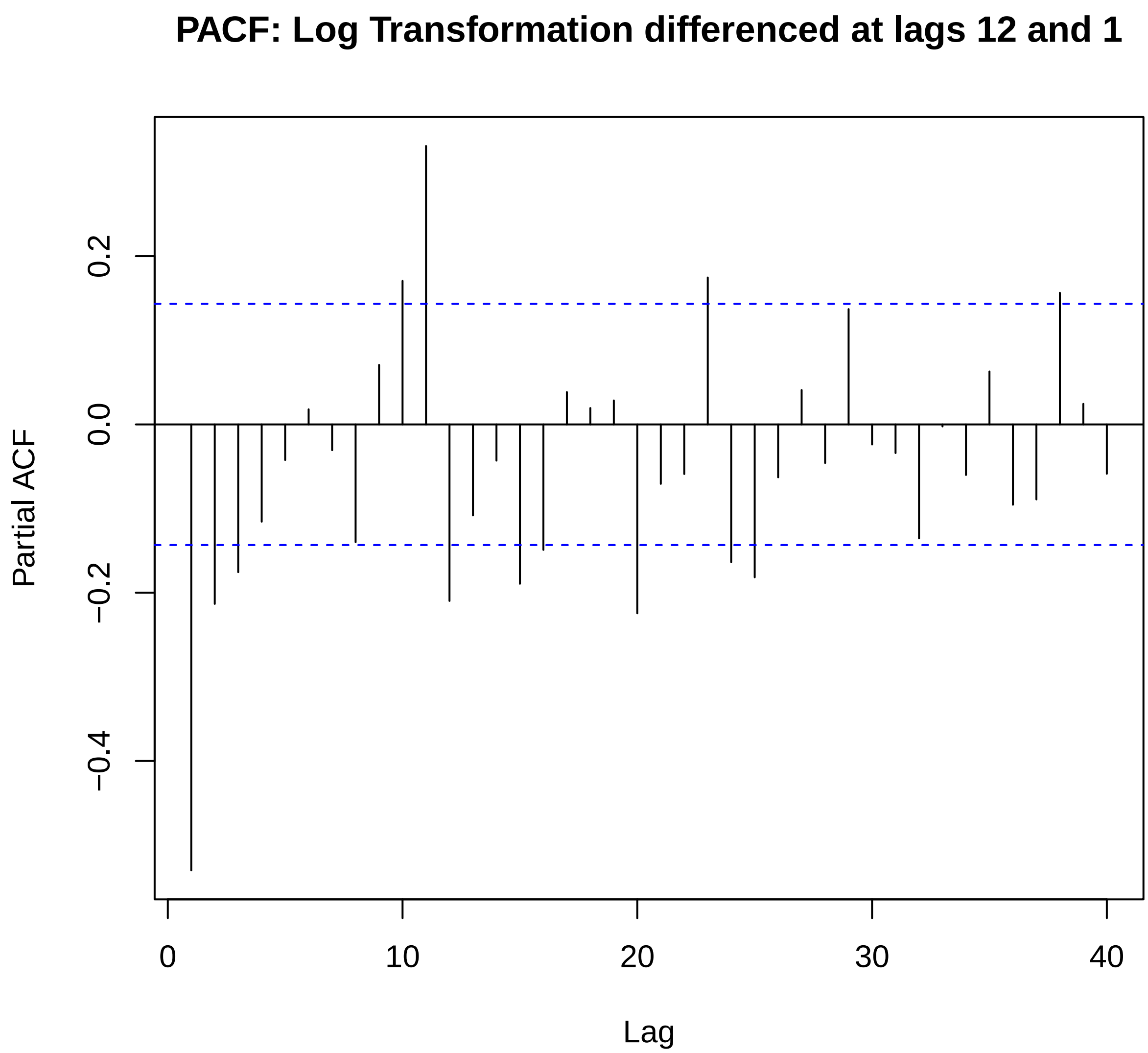
Choosing Possible Models

To choose an appropriate model for our data set, we must first look at the PACF and ACF of $\nabla_1 \nabla_{12} ln(U_t)$. Since differencing was done at lag 12 and again at lag 1, based on the two graphs, we will choose appropriate p, q, P, Q terms for a $SARIMA(p, 1, q) \times (P, 1, Q)$ model.



Looking at the ACF graph above, we see significant autocorrelation outside the confidence intervals at lags 1,9, 12, 13 and 39. To look for significant autocorrelation for non-seasonal trend, we look for spikes outside the confidence interval for lags under 12. Thus, suitable candidates for p is 1 or 9.

To look for significant lags for seasonal trend, we look for spikes outside the confidence interval for lags at 12 and beyond. Since there is a spike at lag 12 and lag 39 is relatively close to lag 36, possible candidates for Q are 1, or 3.



To pick possible candidates for the coefficients p and P, we repeat the same process with the PACF graph. Looking at the graph, it is clear that there are significant partial autocorrelation at lags 1, 2, 3, 10, 11, 12, 14, 20, 21, 23, 24,25 37. Since there are so many to choose from, we will narrow our scope to lags that has the largest of spikes.

Thus, for possible candidates of p, we will choose 1 or 11. For possible candidates of P, we will choose 1 or 3. We will also consider 0 since seasonal trend is not particularly strong.

Testing the Models

To determine the effectiveness of a particular model, we prioritize models with the lowest Akaike Information Criterion (AICc) score defined by the formula:

$$AICc = -2lnL(\theta_q, \phi_p, \frac{S(\theta_q, \phi_p)}{n}) + \frac{2n(p+q+1)}{(n-p-q-2)}$$

Where θ_q and ϕ_p are MLEs that minimize $-2lnL(...)$

From the coefficients we picked, p = 1, 3 ; P = 0, 1, 3 ; q = 1 ; Q = 1,3 The following SARIMA models scored lowest in AICc in the preliminary testings.
For code and a full list of AICc scores, please refer to the Section 9 of the Appendix

```
> Using AICc() from library(qpCR) in R, output of various AICc scores for SARIMA models
> Output of lowest AICc scores of tested SARIMA models:
> AICc of SARIMA(0,1,9)X(0,1,3):  -351.459676058847
> AICc of SARIMA(1,1,1)X(0,1,3):  -350.850669229957
> AICc of SARIMA(0,1,2)X(0,1,3):  -349.833425681831
> AICc of SARIMA(2,1,0)X(0,1,3):  -349.395702413376
```

From our testing, we observe that adding a seasonal AR component to our model often increased our AICc scores.

Furthermore, we observe that the model that gave the lowest AICc score was indeed within the parameters of our suggested coefficients. Since SARIMA(0,1,9)X(0,1,3) has the lowest AICc, we will chose this as model 1 as our primary model for diagnostic testings.

```
>
> Call:
> arima(x = popcorn.log, order = c(0, 1, 9), seasonal = list(order = c(0, 1, 3),
>     period = 12), method = "ML")
>
> Coefficients:
>           ma1     ma2      ma3     ma4     ma5     ma6      ma7     ma8     ma9
>       -0.6952  0.1914  -0.1029  0.0263  0.0242  0.0346  -0.0835  0.0219  0.2682
> s.e.   0.0728  0.0936   0.0968  0.1068  0.1220  0.1379   0.1367  0.1149  0.0851
>          sma1    sma2    sma3
>       -0.7663  0.0487  0.2112
> s.e.   0.0915  0.1145  0.0882
>
> sigma^2 estimated as 0.007276:  log likelihood = 189.56,  aic = -353.13
```

MODEL 1 | SARIMA(0, 1, 9) x (0, 1, 3)

From our arima call function above, we see that the coefficient of MA3 is quite small compared to its standard error. We check if 0 falls in the 95% confidence interval using the formula defined below:

$$Confidence.Int. = \ Coefficient \ \ \pm \ 1.96(\ Standard \ Err.)$$

Since MA3's coefficent is -0.103 and its S.E is 0.0968, we can see that 0 falls within the confidence interval. Thus it is possible that MA3 may be 0 and thus we test for if the AICc increases if we set MA3 = 0.

```
>
> Call:
> arima(x = popcorn.log, order = c(0, 1, 9), seasonal = list(order = c(0, 1, 3),
>     period = 12), fixed = c(NA, NA, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA),
>     method = "ML")
>
> Coefficients:
>           ma1     ma2  ma3      ma4     ma5     ma6      ma7     ma8     ma9
>       -0.6872  0.1394    0  -0.0368  0.0503  0.0154  -0.0705  0.0005  0.2881
> s.e.   0.0694  0.0753    0   0.0859  0.1118  0.1226   0.1207  0.0987  0.0801
>          sma1    sma2    sma3
>       -0.7550  0.0448  0.2100
> s.e.   0.0926  0.1160  0.0897
>
> sigma^2 estimated as 0.007324:  log likelihood = 188.98,  aic = -353.96

> AICc of SARIMA(0,1,9)X(0,1,3):  -351.459676058847
> AICc of SARIMA(0,1,9)X(0,1,3) w/ MA3 Removed:  -352.295643674403
```

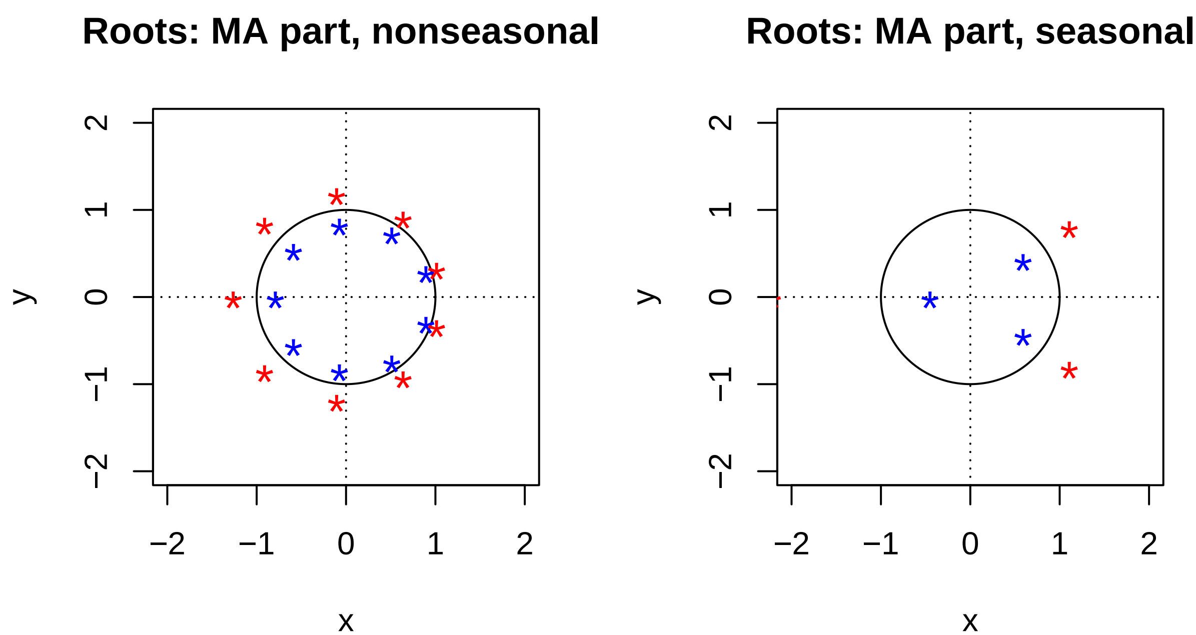
Since the AICc decreased, we set MA3 = 0 and we repeat the test process for every coefficient whose confidence interval contains 0. We do not set a coefficient to 0 if it does not decrease the AICc score. Thus, we get the following model:

```
>
> Call:
> arima(x = popcorn.log, order = c(0, 1, 9), seasonal = list(order = c(0, 1, 3),
>     period = 12), fixed = c(NA, NA, 0, 0, 0, 0, 0, 0, NA, NA, 0, NA), method = "ML")
>
> Coefficients:
>           ma1     ma2  ma3  ma4  ma5  ma6  ma7  ma8     ma9     sma1  sma2
>       -0.6899  0.1367    0    0    0    0    0    0  0.2549  -0.7283     0
> s.e.   0.0693  0.0673    0    0    0    0    0    0  0.0560   0.0669     0
>          sma3
>        0.2407
> s.e.  0.0658
>
> sigma^2 estimated as 0.00738:  log likelihood = 188.32,  aic = -364.64

> AICc of SARIMA(0,1,9)X(0,1,3) w/ Coefficients set to 0:  -362.969202376211
```

Thus, the SARIMA(0,1,9)x(0,1,3) model with MA3, MA4, MA5, MA6, MA7, MA8, and SMA2 set to 0 yields us with the lowest AICc score.

Since the model is pure MA, the model is stationary. Thus, we just need to check for invertibility.

We check to see the unit roots of the model above to see if it is invertible for both the seasonal and non-seasonal components.



From the graphs, we can see that both the MA seasonal and non-seasonal component has inverse roots (marked blue) inside the unit circle and unit roots (marked red) outside the unit circle.

Thus, we know the model is invertible and casual. From the above, we can write down the equation for model 1 as follows:

$$\nabla_1 \nabla_{12} ln(U_t) = (1 - 0.6899_{(0.0693)} B + 0.1367_{(0.0673)} B^2 + 0.2549_{(0.0560)} B^9)(1 - 0.7283_{(0.0669)} B^{12} + 0.2407_{(0.0658)} B^{36}) Z_t$$

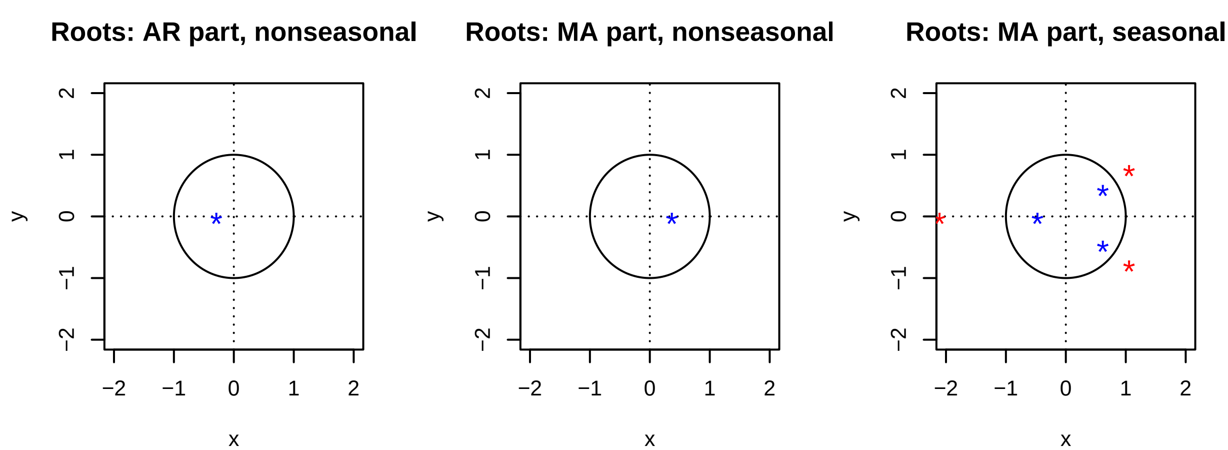
$$\hat{\sigma}_z^2 = 0.00738$$

MODEL 2 | SARIMA(1, 1, 1) x (0, 1, 3)

For our second model, let's work with SARIMA(1,1,1)X(0,1,3), our mode with the second lowest AICc score.

```
>
> Call:
> arima(x = popcorn.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 3),
>     period = 12), method = "ML")
>
> Coefficients:
>           ar1      ma1     sma1     sma2    sma3
>       -0.2935  -0.3707  -0.7291  -0.0578  0.3074
> s.e.   0.1273   0.1291   0.0844   0.1048  0.0873
>
> sigma^2 estimated as 0.007883:  log likelihood = 181.58,  aic = -351.16

>
> Call:
> arima(x = popcorn.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 3),
>     period = 12), fixed = c(NA, NA, NA, 0, NA), method = "ML")
>
> Coefficients:
>           ar1      ma1     sma1  sma2    sma3
>       -0.2930  -0.3669  -0.7590     0  0.2773
> s.e.   0.1271   0.1286   0.0656     0  0.0675
>
> sigma^2 estimated as 0.007899:  log likelihood = 181.43,  aic = -352.86

> AICc of SARIMA(1,1,1)X(0,1,3):  -350.850669229957
> AICc of SARIMA(1,1,1)X(0,1,3) w/ Coefficients set to 0:  -352.553837932826
```

Thus we work with SARIMA(1,1,1)x(0,1,3) model with SMA2 = 0. Now to check for invertiblity and stationarity.



From the graphs, we can see that there are no unit roots within or on the circle and no inverse roots are outside the unit circle. Thus this model is stationary and invertible.

$$(1 + 0.2930_{(0.1271)}B)\nabla_1\nabla_{12}ln(U_t) = (1 - 0.3669_{(0.1286)}B)(1 - 0.7590_{(0.0656)}B^{12} + 0.2773_{(0.0675)}B^{36})Z_t$$

$$\hat{\sigma}_z^2 = 0.007899$$

MODEL 3 | SARIMA (10, 1, 0) X (1, 1, 3)

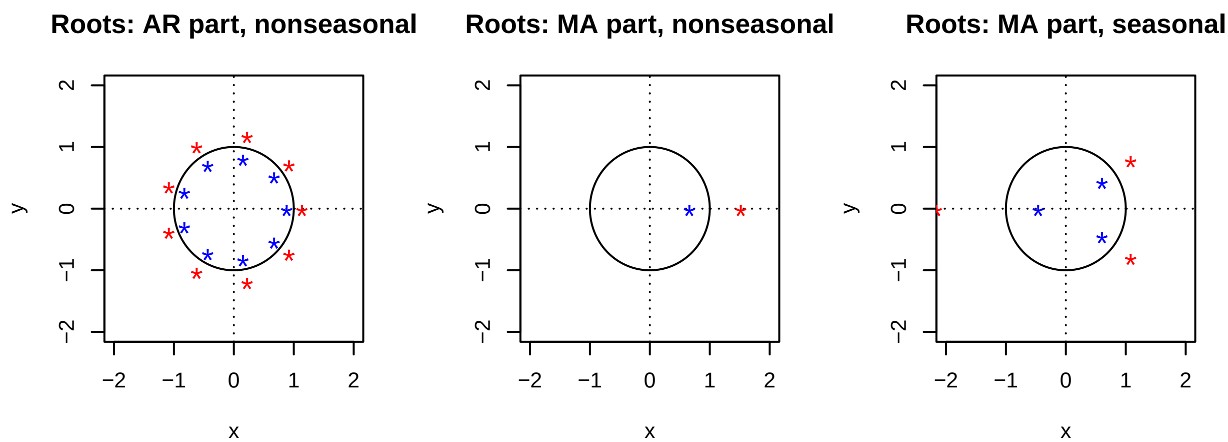
For model 3, let's look at a model that had the second lowest AICc score after setting all non-significant coefficients to 0.

```
>
> Call:
> arima(x = popcorn.log, order = c(10, 1, 1), seasonal = list(order = c(1, 1,
>     3), period = 12), method = "ML")
>
> Coefficients:
>          ar1     ar2     ar3     ar4      ar5      ar6     ar7      ar8     ar9
>       0.0452  0.2398  0.0711  0.0220  -0.0146  -0.0171  -0.037  -0.0632  0.2471
> s.e.  0.1732  0.1199  0.0844  0.0795   0.0789   0.0803   0.077   0.0778  0.0775
>         ar10      ma1     sar1     sma1     sma2    sma3
>       0.0820  -0.7162  -0.2177  -0.5666  -0.1425  0.2939
> s.e.  0.0862   0.1515   0.3256   0.3203   0.2644  0.1020
>
> sigma^2 estimated as 0.007392:  log likelihood = 187.82,  aic = -343.63
```

```
>
> Call:
> arima(x = popcorn.log, order = c(10, 1, 1), seasonal = list(order = c(1, 1,
>     3), period = 12), fixed = c(0, NA, 0, 0, 0, 0, 0, 0, NA, 0, NA, 0, NA, 0,
>     NA), method = "ML")
>
> Coefficients:
>       ar1     ar2  ar3  ar4  ar5  ar6  ar7  ar8     ar9  ar10      ma1  sar1
>         0  0.1989    0    0    0    0    0    0  0.2369     0  -0.6592     0
> s.e.    0  0.0844    0    0    0    0    0    0  0.0743     0   0.0640     0
>          sma1  sma2    sma3
>       -0.7423     0  0.2585
> s.e.   0.0636     0  0.0655
>
> sigma^2 estimated as 0.007508:  log likelihood = 186.34,  aic = -360.69
```

```
> AICc of SARIMA(10,1,1)X(1,1,3):  -341.021651173867
> AICc of SARIMA(10,1,1)X(1,1,3) w/ Coefficients Set to 0:  -358.077078053323
```

Note that AR10 was not needed, thus an indicator that p = 10 is excessive. Furthermore SAR1 = 0, thus a seasonal AR term is not required.



Just as the previous two models, this model is stationary and invertible as all unit roots are outside the unit circle and all inverse roots are within the unit circle.

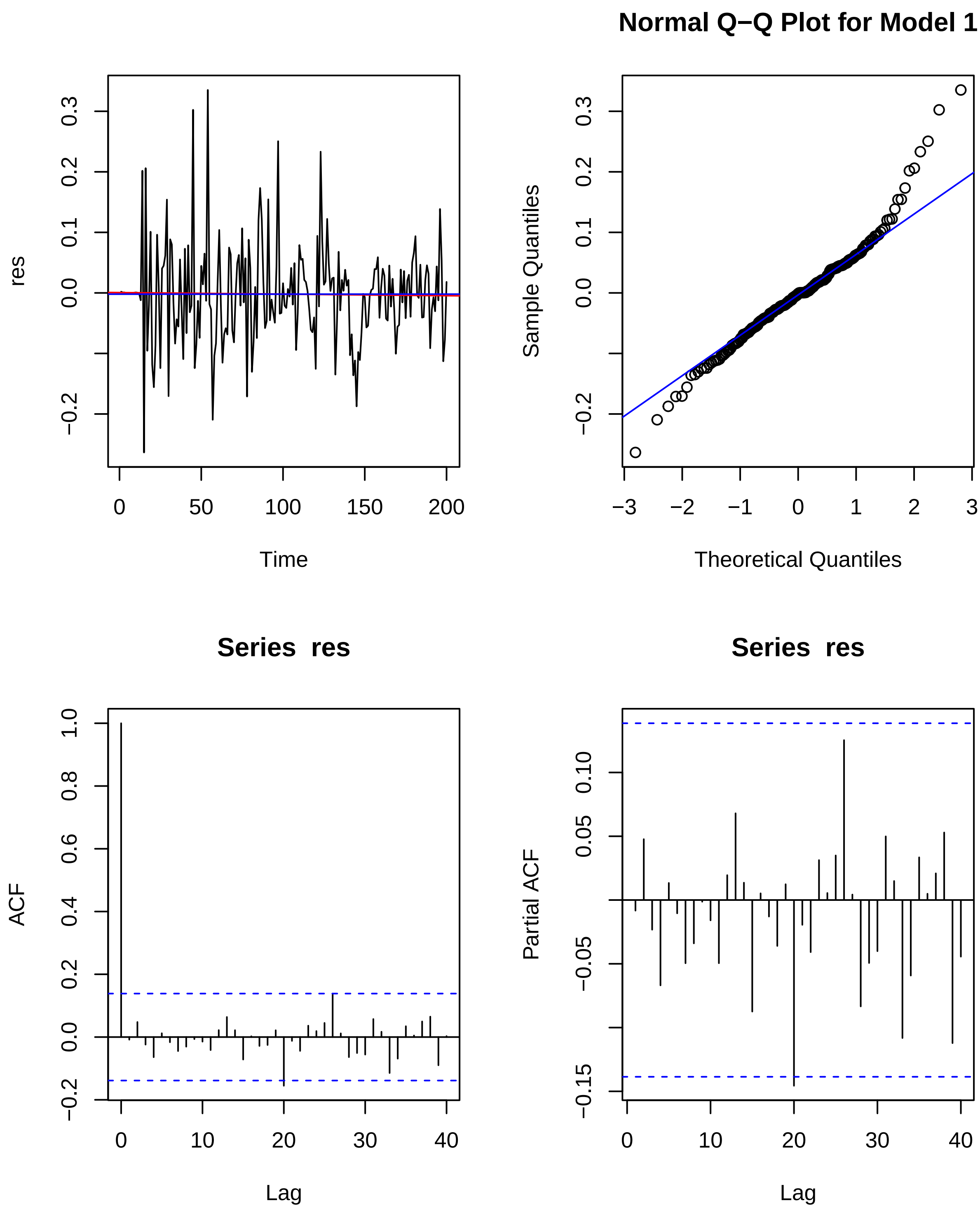
$$(1-0.1989_{(0.8444)}B^2-0.2369_{(0.0743)}B^9)\nabla_1\nabla_{12}ln(U_t) = (1-0.6592_{(0.0640)}B)(1-0.7423_{(0.0636)}B^{12}+0.2585_{(0.0655)}B^{36})Z_t$$

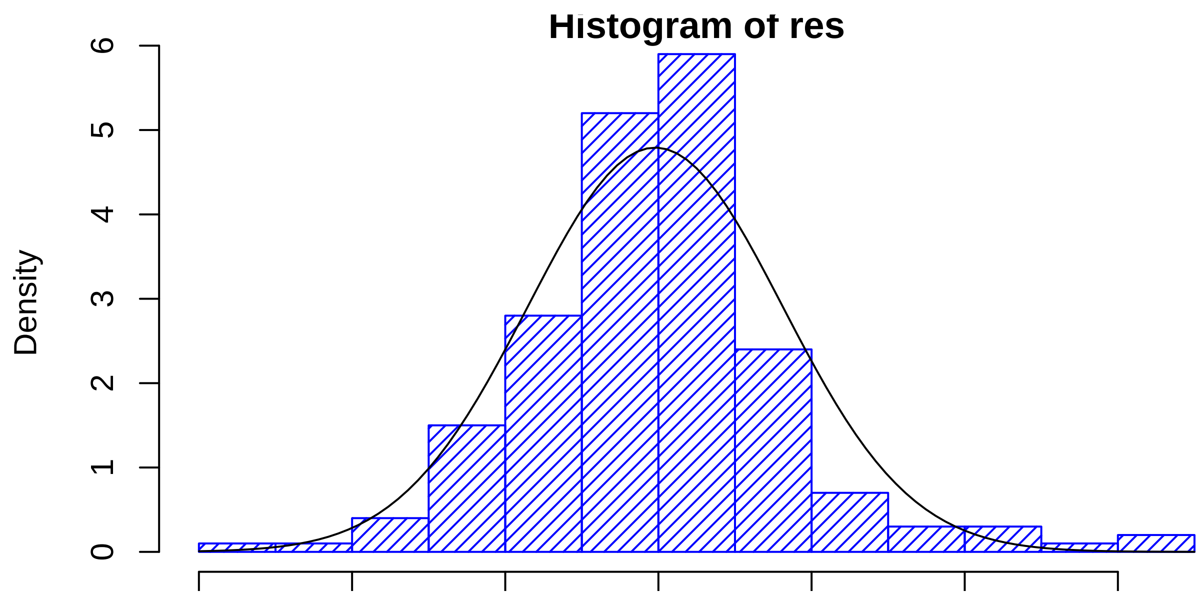
$$\hat{\sigma}_z^2 = 0.007508$$

Now that we have identified three potential models, let us perform some diagonistic checks for each particular SARIMA model.

DIAGONISTIC CHECKING: MODEL 1

To understand how well selective models perform, we must look at the residuals of our chosen models.






```
> Mean:  -0.00204699270757072
> Standard Deviation:  0.0832547267644329
```

From the graphs above, we can see that though the ACF and PACF of the residauls looks good, the ggplot and histogram does not show a well constructed normal distribution. Thus, this model may not have normally distributed residuals. Otherwise, the mean is close to 0 and the standard deviation is not that high. We now run some tests to assess normality and independence of residuals.

```
>
>   Shapiro-Wilk normality test
>
> data:  res
> W = 0.96262, p-value = 3.837e-05


>
>   Box-Pierce test
>
> data:  res
> X-squared = 3.4736, df = 9, p-value = 0.9425


>
>   Box-Ljung test
>
> data:  res
> X-squared = 3.6501, df = 9, p-value = 0.9329


>
>   Box-Ljung test
>
> data:  (res^2)
> X-squared = 16.288, df = 14, p-value = 0.2961


>
```

```
> Call:
> ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
>
>
> Order selected 0  sigma^2 estimated as  0.006931
```

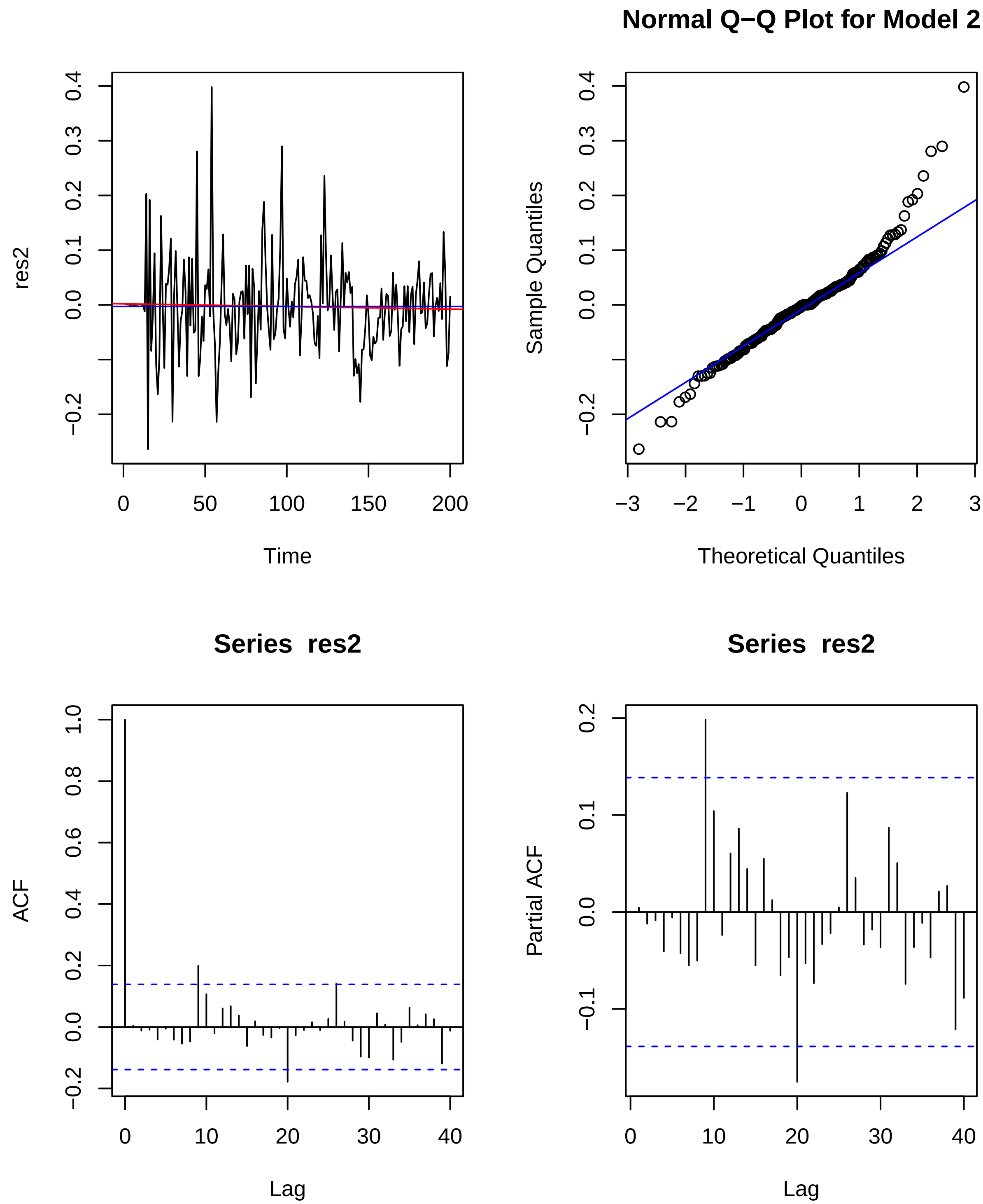
From the tests, we can see that the residuals of our model fail to pass the Shapiro-Wilk normality test but manages to pass the remaining Box-Pierce, Box-Ljung, and McLeod-Li tests.

Thus, for this model, we can assume that the residuals are not normal, the model passes the Box-Pierce test, they are highly uncorrelated. Furthermore, since the model also passes both the Box-Ljung and McLeod-Li test, we can be assured that the residuals are not dependent, and thus most likely are independent and non-correlated.

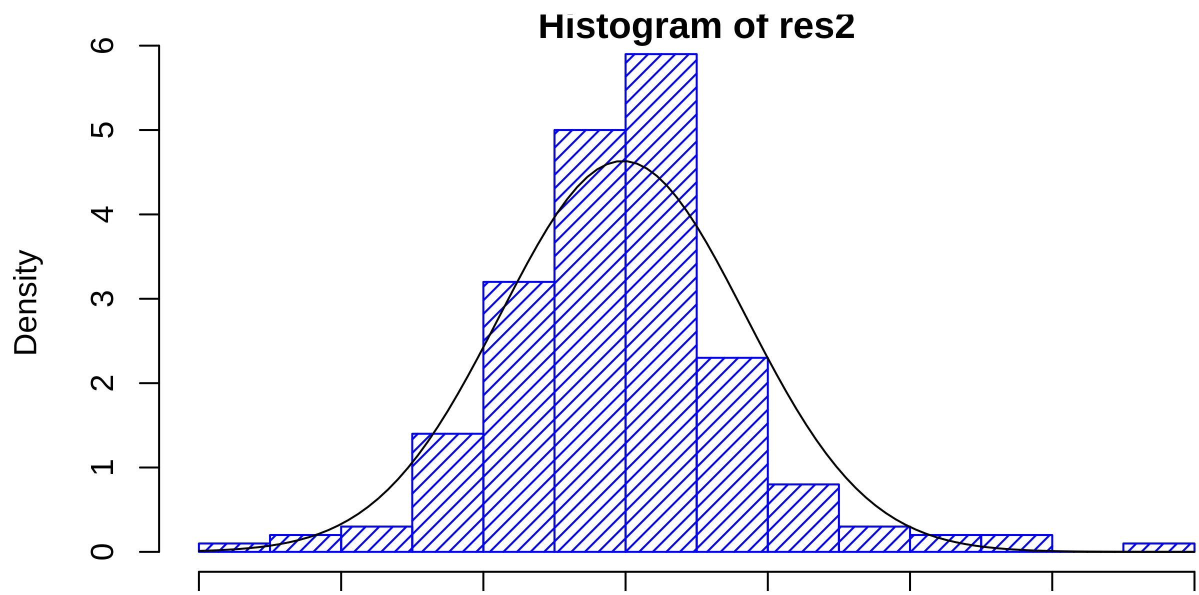
Furthermore, it is good that Yule-Walker identifies the residuals of our model as ar(0), suggesting that it resembles white noise.

Let's look at model 2 to see if it is more suitable for forecasting.

DIAGONISTIC CHECKING: MODEL 2



Notice how the normal Q-Q Plot, ACF and PACF of model 2's residuals already looks considerably worse than model 1's.




```
> Mean:  -0.0029380609051701
> Standard Deviation:  0.0861066993465473
```

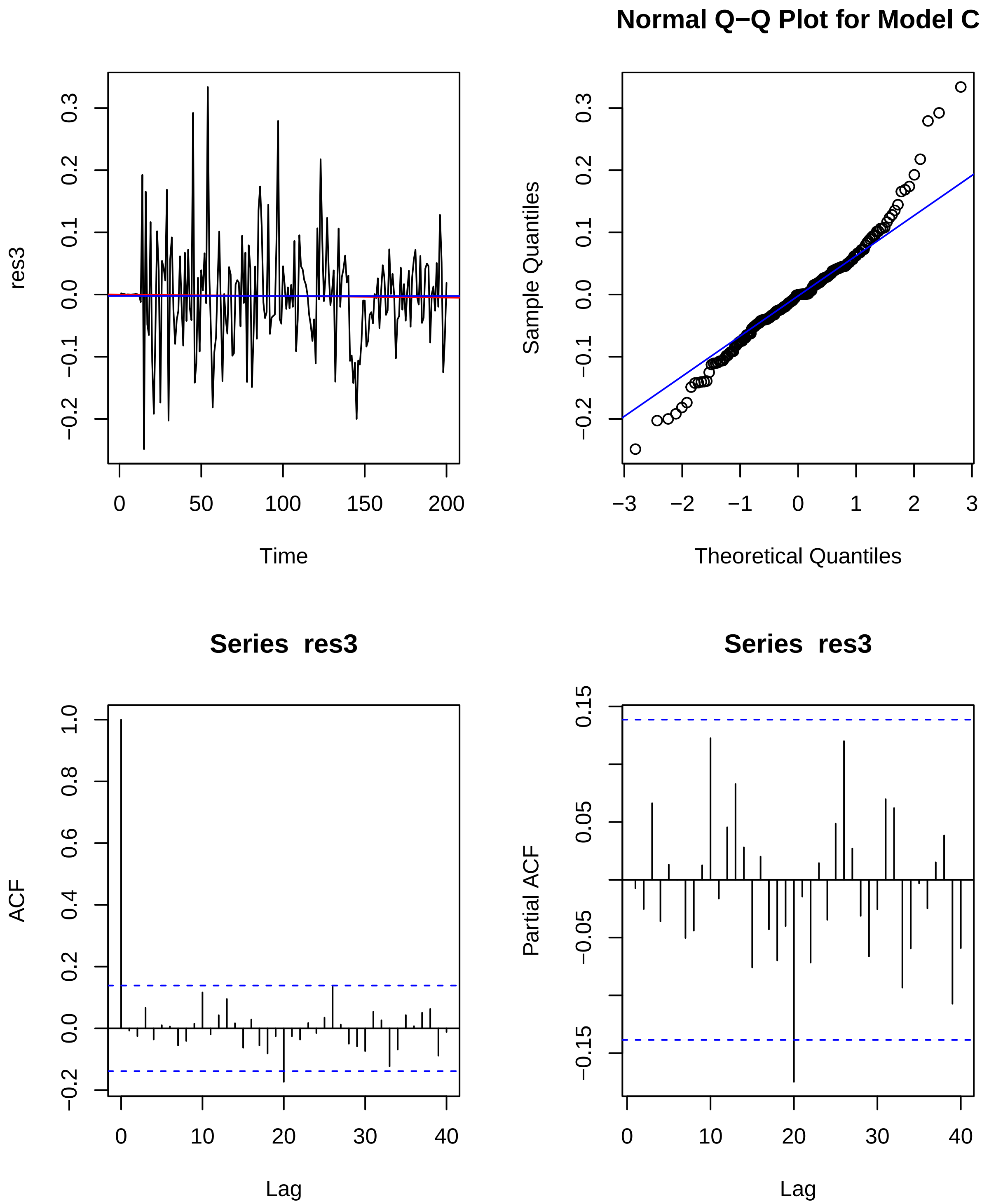
From this particular model, we can see that the ACF and PACF graphs have significant problems since there are several lags outside the confidence intervals. Thus, almost immediately, we sense that this model may not be the best for forecasting.

```
>
>   Shapiro-Wilk normality test
>
> data:  res2
> W = 0.95208, p-value = 3.006e-06

>
>   Box-Pierce test
>
> data:  res2
> X-squared = 13.977, df = 10, p-value = 0.1741

>
>   Box-Ljung test
>
> data:  res2
> X-squared = 14.811, df = 10, p-value = 0.1391

>
>   Box-Ljung test
>
> data:  (res2^2)
> X-squared = 12.511, df = 14, p-value = 0.5653

>
> Call:
```

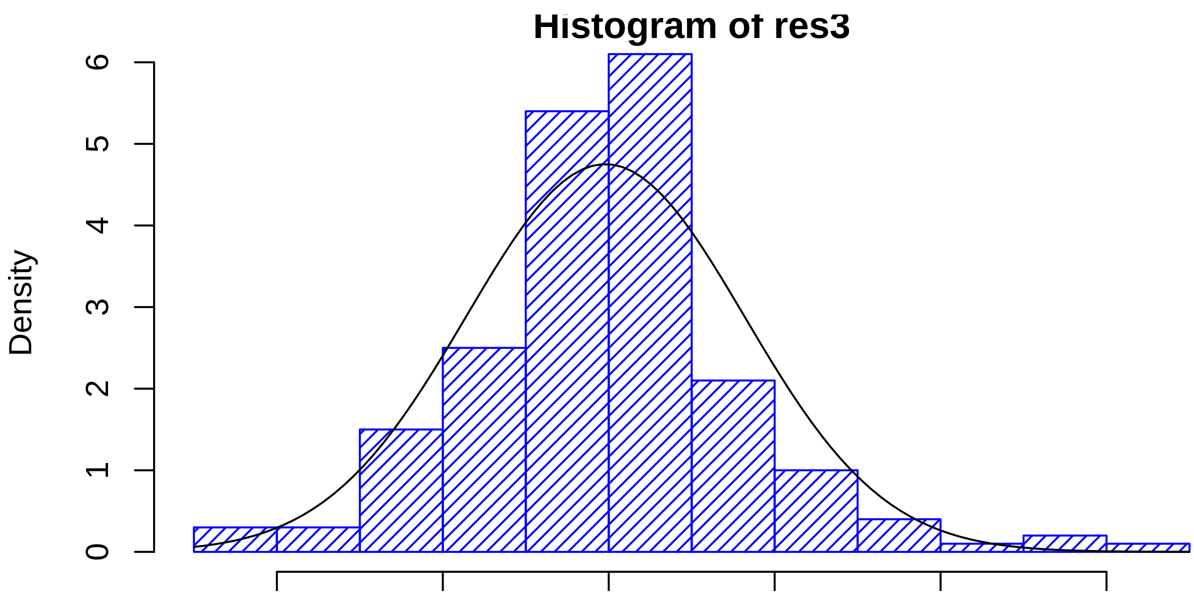
```
> ar(x = res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
>
>
> Order selected 0  sigma^2 estimated as  0.007414
```

From the above, we notice that the mean and the standard deviation is higher than model 1. Furthermore, the residuals of this model is even more unlikely to be normal compared to model 1 as indicated by the p-value of the Shapiro-Wilk test. The Box-Pierce test does pass but the p-value clearly does not offer as much confidence about the residauls being uncorrelated as that of model 1's. Furthermore, the same can be said about linear dependence from the results of the Box-Ljung test. The only test that scores better than model 1 is the McLeod-Li test. Given all the obvious flaws from all of what is mentioned above, we prefer model 1.

DIAGONISTIC CHECKING: MODEL 3



Model 3's ACF and PACF plots display several lags which are outside of the confidence interval, which is worrisome. The ggPlot also does not look great as it begans to deviate quickly away form a straight line past 1 standard deviation.




```
> Mean:  -0.00242888293936118
> Standard Deviation:  0.0839644170715699
```

From this particular model, we can see that the ACF and PACF graphs have significant problems since there are several lags outside the confidence intervals. Thus, almost immediately, we sense that this model may not be the best for forecasting.

```
>
>   Shapiro-Wilk normality test
>
> data:  res3
> W = 0.96494, p-value = 7.043e-05

>
>   Box-Pierce test
>
> data:  res3
> X-squared = 7.3041, df = 9, p-value = 0.6055

>
>   Box-Ljung test
>
> data:  res3
> X-squared = 7.7393, df = 9, p-value = 0.5606

>
>   Box-Ljung test
>
> data:  (res3^2)
> X-squared = 16.129, df = 14, p-value = 0.3055

>
> Call:
```

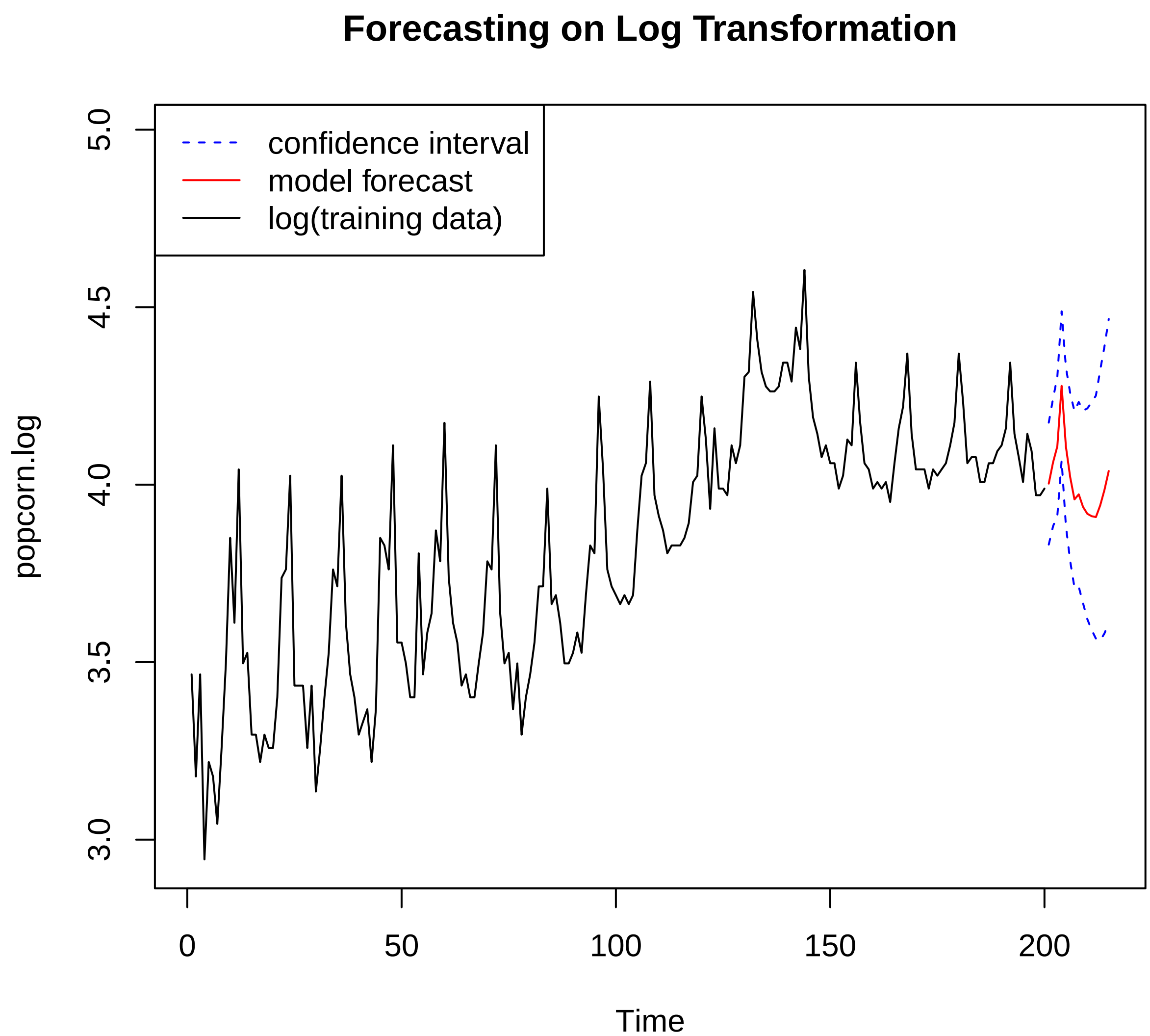
```
> ar(x = res3, aic = TRUE, order.max = NULL, method = c("yule-walker"))
>
>
> Order selected 0  sigma^2 estimated as  0.00705
```

From the results, we can see that model 3 suffer in similar ways compared to model 1 in ways that model 2 did. Model 3 is less normally distributed, and less confident of linear independence and non-correlation in its residuals. Notice how all three models fail to pass the Shapiro-Wilk test. This may be indicative of the fact that our data is not pulled at random and that there may exist influences outside of the time series that makes the data not normally distributed that using SARIMA, our models can not predict.

FORECASTING

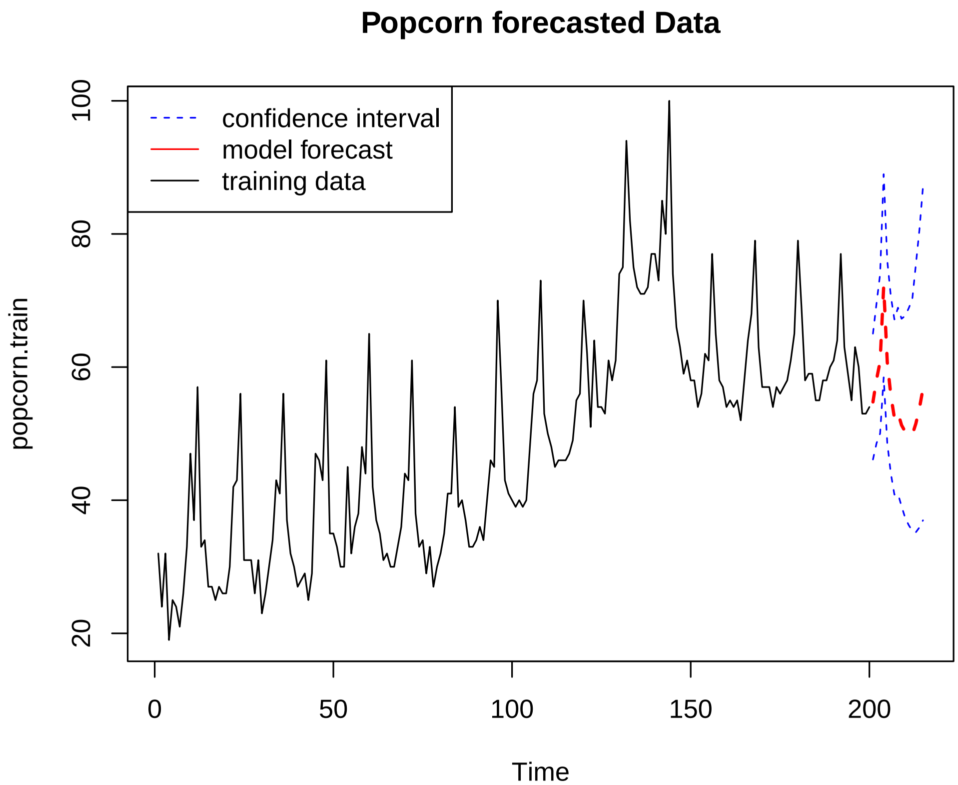
We will forecast the dataset following a SARIMA(0,1,9)X(0,1,3):
$\nabla_1 \nabla_{12} ln(U_t) = (1 - 0.6899_{(0.0693)} B + 0.1367_{(0.0673)} B^2 + 0.2549_{(0.0560)} B^9)(1 - 0.7283_{(0.0669)} B^{12} + 0.2407_{(0.0658)} B^{36}) Z_t$

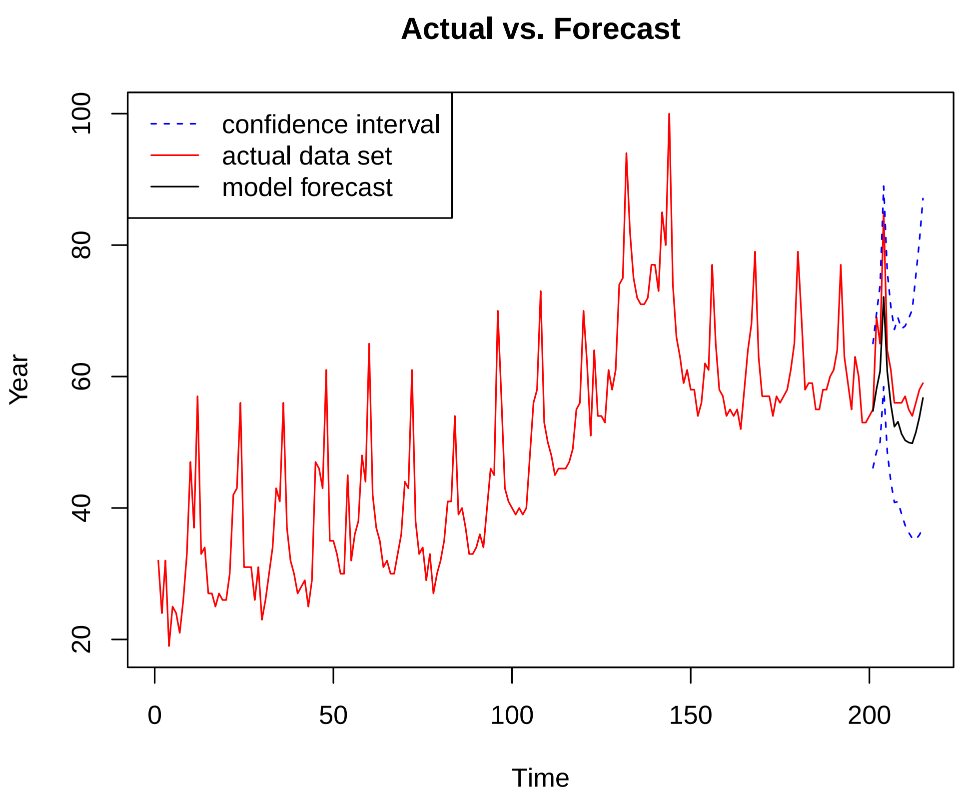
We first transform on the log transformation of the original dataset.



Looking at the graph, we see that the interval interval starts to expand after time 200. This is indicative of the model being possibly unstable and uncertain, therefore compensating its accuracy.

Next, we will perform forecasting on the original data set.


Popcorn forecasted Data
confidence interval
model forecast
training data
popcorn.train
Time
0
50
100
150
200
20
40
60
80
100


Actual vs. Forecast
confidence interval
actual data set
model forecast
Year
Time
0
50
100
150
200
20
40
60
80
100

CONCLUSION

Model Tested:

$$\nabla_1\nabla_{12}ln(U_t) = (1-0.6899_{(0.0693)}B+0.1367_{(0.0673)}B^2+0.2549_{(0.0560)}B^9)(1-0.7283_{(0.0669)}B^{12}+0.2407_{(0.0658)}B^{36})Z_t$$

In conclusion, the final SARIMA (0,1,9)X(0,1,3) model, defined as $\nabla_1\nabla_{12}ln(U_t) = (1 - 0.6899_{(0.0693)}B + 0.1367_{(0.0673)}B^2 + 0.2549_{(0.0560)}B^9)(1 - 0.7283_{(0.0669)}B^{12} + 0.2407_{(0.0658)}B^{36})Z_t$, does not seem to predict the actual data with high confidence. As seen from the forecast graph above, the forecast has clear visible deviations from the actual data set and furthermore, the actual data set is barely within the confidence interval. Thus, we can conclude that this model is not stable and thus should be used very sparingly to predict future trends of popcorn.

Furthermore, during testing, all of the selected models failed to pass the Shapiro-Wilk normality test. Therefore, there is a high probably chance that the data set above is not normally distributed and thus data pulled are not chosen at random. In other words, outside influences makes is nearly impossible to predict the time series above and thus SARIMA techniques may be adequate in predicting such a volatile data set.

Hence, I conclude that the goal to produce a suitable model for predicting future popcorn trend is not reached and thus, the data set above is not appropriate for our current time series techniques.

Acknowledgements

I humbly offer my appreciation and gratitude to Professor Raya Feldman for providing guidance and support towards the completition of this project. I offer many thanks to my peers, of which I highlight my deepest appreciation to Hannah Mok, for often providing snacks, and offering yours hours where we spent most laughing and tackling statistical semantics, and Max Reedy, who never fails to amaze and impress with his kindness and intellect, often rescuing and paving new paths for problems which were doomed to be dead ends. I save lastly my deepest sincerity and regards to thank my parents, for their earnest and hard-working nature to ensure that I can receive the education which I am blessed with today.

REFERENCES

- RStudio Documentation - https://docs.rstudio.com/
- RDocumentation - https://www.rdocumentation.org/
- Raya Feldman (Professor of Probability and Theory) - https://www.pstat.ucsb.edu/people/raya-feldman
- UCSB Lecture Series 174 - Time Series Analysis

APPENDIX

For AICc Score Values, please go to Section 9

1. Initial Steps

```
#reading in the dataset
popcorn <- read.csv("C:/Users/Jason/Desktop/174 project/popcorn.csv",header=TRUE,skip=1)

#changing column name for easier reference in the future
colnames(popcorn)[2] <- "interest"

#plotting the original dataset
tsdat <- ts(popcorn$interest, start=c(2004,1),end = c(2021,11),frequency=12)
ts.plot(tsdat,ylab="Interest",xlab="Year")

#minor formatting for the graph
title = "Raw Data"
mysubtitle = "Data source: Google Trends"
source = "https://trends.google.com/trends/explore?date=all&geo=US&q=popcorn"
mtext(side=3, line=3, cex=1, title)
mtext(side=3, line=2, cex=0.7, mysubtitle)
mtext(side=3, line=1, cex=0.5, source)

#plotting the data set as a time series
plot.ts(popcorn$interest, main="Raw Data with Trend Lines", ylab="Interest")
nt=length(popcorn$interest)
fit <- lm(popcorn$interest~as.numeric(1:nt));abline(fit,col="red")
abline(h=mean(popcorn$interest),col="blue")
legend("topleft", legend=c(mean(popcorn$interest)),col=c("blue"),lty=1:1,cex=0.8)
```

2. Creating the Training Data

```
#dividing the data set into training/testing
popcorn.train=popcorn$interest[c(1:200)]
popcorn.test=popcorn$interest[c(201:215)]

#plotting the time series
ts.plot(popcorn$interest,col="orange", main="Testing and Training Data Set",ylab="Interest")
lines(popcorn.train,col="black")
legend("topleft", legend = c("training data set","testing data set"), col=c("black","orange"),lty=1:1,c

#creating a matrix of tables
par(mfrow = c(2,2))
ts.plot(popcorn.train, main="Training Data", ylab="Interest")
fit <- lm(popcorn.train ~ as.numeric(1:length(popcorn.train))); abline(fit, col="red")
abline(h=mean(popcorn.train),col="blue")
hist(popcorn.train, col="light blue",xlab="",main="Histogram: Training Data")
acf(popcorn.train,lag.max=40, main="ACF: Training Data")
```

```
pacf(popcorn.train,lag.max=40, main="PACF: Training Data")
par(mfrow=c(1,1))
```

3. Transforming the Dataset

```
#transforming the dataset
library(MASS)

#transforming the data set with boxcox
bcTransform <- boxcox(popcorn.train~ as.numeric(1:length(popcorn.train)))
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
legend("topright", legend=lambda, cex=0.7)
mtext(side=3, line=1, cex=1, "Power Transformation Likelikelihood Graph")


par(mfrow = c(2,2))

#listing all of the transformations
popcorn.bc = (1/lambda)*(popcorn.train^lambda-1)
popcorn.log = log(popcorn.train)
popcorn.sqrt = sqrt(popcorn.train)

#plotting each transformation as a time series graph
ts.plot(popcorn.train, main="Original Training Data", ylab="Interest")
ts.plot(popcorn.log, main="Log Transformation Training Data", ylab="popcorn.log")
ts.plot(popcorn.bc, main="Box-Cox Training Data", ylab="popcorn.bc")
ts.plot(popcorn.sqrt, main="Sqrt Transformation Training Data", ylab="popcorn.sqrt")

#outputting variance of each
cat(paste("Variance of Original Training Data: ", var(popcorn.train)),
    paste("Variance of Sqrt Transformation: ", var(popcorn.sqrt)),
    paste("Variance of Box Cox Transformation: ", var(popcorn.bc)),
    paste("Variance of Log Transformation: ", var(popcorn.log))
    ,sep="\n")

#histogram of each one
par(mfrow = c(2,2))
hist(popcorn.train, col="light blue", xlab="", main="histogram; original training data")
hist(popcorn.log, col="light blue", xlab="", main="histogram; log transformation")
hist(popcorn.bc, col="light blue", xlab="", main="histogram; boxcox transformation")
hist(popcorn.sqrt, col="light blue", xlab="", main="histogram; sqrt transformation")
```

4. Differencing the Data

```
#decomposition graph
y <- ts(as.ts(popcorn.log),frequency=12)
decomp <- decompose(y)
plot(decomp)
```

```
#plotting the logarithmic transformation wiht trend line
plot.ts(popcorn.log, main="Log Transformation")
fit <- lm(popcorn.log ~ as.numeric(1:length(popcorn.log))); abline(fit, col="red")
abline(h=mean(popcorn.log),col="blue")

#differencing log transformation at lag 12
popcorn_12 <- diff (popcorn.log, lag=12)
plot.ts(popcorn_12, main="Log Transformation differenced at lag 12")
fit <- lm(popcorn_12 ~ as.numeric(1:length(popcorn_12))); abline(fit, col="red")
abline(h=mean(popcorn_12),col="blue")

#differencing again at lag 1
popcorn.stat <- diff(popcorn_12,lag=1)
plot.ts(popcorn.stat, main="Log Transformation differenced at lag 12 & lag 1")
fit <- lm(popcorn.stat ~ as.numeric(1:length(popcorn.stat))); abline(fit, col="red")
abline(h=mean(popcorn.stat),col="blue")

cat(paste("Variance of Log Transformation: ", var(popcorn.log)),
  paste("Variance of Log Transformation (Differenced at Lag = 12): ",var(popcorn_12)),
  paste("Variance of Log Transformation (Differenced at Lag = 12 & Lag = 1): ",var(popcorn.stat))
        , sep="\n")

#acf of log w and without differencing
  acf(popcorn.log, lag.max=40, main="ACF: Log Transformation")
  acf(popcorn_12, lag.max=40, main="ACF: Log Transformation differenced at lag 12")
  acf(popcorn.stat, lag.max=40, main="ACF: Log Transformation differenced at lags 12 and 1")

#histogram of log w and without differencing
  par(mfrow = c(2,2))
hist(popcorn.log, col="light blue",xlab="",main="ln(U_T)")
hist(popcorn_12, col="light blue",xlab="",main="ln(U_T) differenced at lags 12")
hist(popcorn.stat, col="light blue",xlab="",main="ln(U_T) differenced at lags 12 & 1")
hist(popcorn.stat, density=20,breaks=20, col="blue", xlab="", prob=TRUE, main = "Histogram w/ Gaussian C
m<-mean(popcorn.stat) #mean here
std<- sqrt(var(popcorn.stat)) #stat here
curve( dnorm(x,m,std), add=TRUE )

#acf and pacf of decided differencing
acf(popcorn.stat, lag.max=40, main="PACF: Log Transformation differenced at lags 12 and 1")
pacf(popcorn.stat, lag.max=40, main="PACF: Log Transformation differenced at lags 12 and 1")
```

5. Choosing Models

```
par(mfrow = c(2,2))
#looking once more at the different histograms
hist(popcorn.log, col="light blue",xlab="",main="ln(U_T)")
hist(popcorn_12, col="light blue",xlab="",main="ln(U_T) differenced at lags 12")
hist(popcorn.stat, col="light blue",xlab="",main="ln(U_T) differenced at lags 12 & 1")
hist(popcorn.stat, density=20,breaks=20, col="blue", xlab="", prob=TRUE, main = "Histogram w/ Gaussian C

#plotted histogram of difference at 12 and 1 with normal distribution curve
```

```r
m<-mean(popcorn.stat)
std<- sqrt(var(popcorn.stat))
curve( dnorm(x,m,std), add=TRUE )

#acf and pacf again
acf(popcorn.stat, lag.max=40, main="PACF: Log Transformation differenced at lags 12 and 1")
pacf(popcorn.stat, lag.max=40, main="PACF: Log Transformation differenced at lags 12 and 1")
```

6. Testing the Models

```r
acf(popcorn.stat, lag.max=40, main="PACF: Log Transformation differenced at lags 12 and 1")
pacf(popcorn.stat, lag.max=40, main="PACF: Log Transformation differenced at lags 12 and 1")
library(qpcR)

#writing in different AICc models from tested code, lowest AICcs
sar1 = AICc(arima(popcorn.log, order=c(0,1,9),
           seasonal = list(order = c(0,1,3), period = 12), method="ML"))
sar2 = AICc(arima(popcorn.log, order=c(1,1,1),
           seasonal = list(order = c(0,1,3), period = 12), method="ML"))
sar3 = AICc(arima(popcorn.log, order=c(0,1,2),
           seasonal = list(order = c(0,1,3), period = 12), method="ML"))
sar4 = AICc(arima(popcorn.log, order=c(2,1,0),
           seasonal = list(order = c(0,1,3), period = 12), method="ML"))

cat(paste("Using AICc() from library(qpCR) in R, output of various AICc scores for SARIMA models"),
    paste("Output of lowest AICc scores of tested SARIMA models:"),
    paste("AICc of SARIMA(0,1,9)X(0,1,3): ", sar1),
  paste("AICc of SARIMA(1,1,1)X(0,1,3): ", sar2),
  paste("AICc of SARIMA(0,1,2)X(0,1,3): ", sar3),
  paste("AICc of SARIMA(2,1,0)X(0,1,3): ", sar4),sep="\n")

# MODEL 1 #
####################################################################################################

#Showing an example of removing a non significant coefficient
sar_main = arima(popcorn.log, order=c(0,1,9),
           seasonal = list(order = c(0,1,3), period = 12), method="ML")
sar_main
sar_main_removed3 = arima(popcorn.log, order=c(0,1,9),
        seasonal = list(order = c(0,1,3), period = 12),
          method="ML",fixed=c(NA,NA,0,NA,NA,NA,NA,NA,NA,NA,NA,NA))
sar_main_removed3 #MA3 removed

cat(paste("AICc of SARIMA(0,1,9)X(0,1,3): ", sar1),
    paste("AICc of SARIMA(0,1,9)X(0,1,3) w/ MA3 Removed: ", AICc(sar_main_removed3)),sep="\n")

#Primary Model that we will work with
    model_main = arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3), period = 12), met

model_main
cat(paste("AICc of SARIMA(0,1,9)X(0,1,3) w/ Coefficients set to 0: ", AICc(model_main)),sep="\n")
```

```r
#Import library(UnitCircle)
library(UnitCircle)
source("plot.roots.R") #unitRoot Source
par(mfrow = c(1,2))

#Plotting the roots to check invertiblity and stationarity of model 1
plot.roots(NULL,polyroot(c(1, -0.6899,0.1367,0,0,0,0,0,0,0.2549)), main="Roots: MA part, nonseasonal ")
plot.roots(NULL,polyroot(c(1, -0.7283,0,0.2407)), main="Roots: MA part, seasonal")

# MODEL 2 #
##################################################################################################
model2 = arima(popcorn.log, order=c(1,1,1),
           seasonal = list(order = c(0,1,3), period = 12), method="ML")
model2

model2_2 = arima(popcorn.log, order=c(1,1,1),
               seasonal = list(order = c(0,1,3), period = 12), method="ML",fixed=c(NA,NA,NA,0,NA))
model2_2

cat(paste("AICc of SARIMA(1,1,1)X(0,1,3): ", AICc(model2)),
    paste("AICc of SARIMA(1,1,1)X(0,1,3) w/ Coefficients set to 0: ", AICc(model2_2)),sep="\n")

#Unit Root for Model 2
    par(mfrow=c(1,3))
plot.roots(NULL,polyroot(c(1, 0.2930)), main="Roots: AR part, nonseasonal")
plot.roots(NULL,polyroot(c(1, -0.3669)), main="Roots: MA part, nonseasonal")
plot.roots(NULL,polyroot(c(1, -0.7590,0,0.2773)), main="Roots: MA part, seasonal")

# MODEL 3 #
##################################################################################################
model3 <- arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3), period = 12), method="Ml
model3

model3_2 <- arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3), period = 12), method=
model3_2

cat(paste("AICc of SARIMA(10,1,1)X(1,1,3): ", AICc(model3)),
    paste("AICc of SARIMA(10,1,1)X(1,1,3) w/ Coefficients Set to 0: ", AICc(model3_2)),sep="\n")

    par(mfrow=c(1,3))

#Unit Root for Model 3
plot.roots(NULL,polyroot(c(1, 0, -0.1989, 0,0,0,0,0,0,-0.2369)), main="Roots: AR part, nonseasonal")
plot.roots(NULL,polyroot(c(1, -0.6592)), main="Roots: MA part, nonseasonal")
plot.roots(NULL,polyroot(c(1, -0.7423,0,0.25785)), main="Roots: MA part, seasonal")
```

7. Diagonistic Checking

```r
# MODEL 1 #
##################################################################################################
#letting fit be model_main
```

```r
fit <- model_main
res <- residuals(fit) #residuals of model 1

#plotting residuals as histogram, qqplot, etc.
par(mfrow=c(2,2))
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 1")
qqline(res,col="blue")

#acf/pacf of model 1
acf(res, lag.max=40)
pacf(res, lag.max=40)

par(mar = c(0.5, 4, 0.5, 0.5))
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 )
cat( paste("Mean: ", m), paste("Standard Deviation: ", std),sep="\n")

#Test for Model 1
shapiro.test(res)
Box.test(res, lag=14, type = c("Box-Pierce"), fitdf = 5)
Box.test(res, lag=14, type = c("Ljung-Box"), fitdf = 5)
Box.test((res^2), lag=14, type = c("Ljung-Box"), fitdf = 0)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# MODEL 2 #
##########################################################################################################
fit2 <- model2_2
res2 <- residuals(fit2)

#plotting model 2 residual graphs
par(mfrow=c(2,2))
plot.ts(res2)
fitt2 <- lm(res2 ~ as.numeric(1:length(res2))); abline(fitt2, col="red")
abline(h=mean(res2), col="blue")
qqnorm(res2,main= "Normal Q-Q Plot for Model 2")
qqline(res2,col="blue")
#acf/pacf for model 2
acf(res2, lag.max=40)
pacf(res2, lag.max=40)

#histgoram of model 2
par(mar = c(0.5, 4, 0.5, 0.5))
hist(res2,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m2 <- mean(res2)
std2 <- sqrt(var(res2))
curve( dnorm(x,m,std2), add=TRUE )
cat( paste("Mean: ", m2), paste("Standard Deviation: ", std2),sep="\n")
```

```
#tests for model 2
shapiro.test(res2)
Box.test(res2, lag=14, type = c("Box-Pierce"), fitdf = 4)
Box.test(res2, lag=14, type = c("Ljung-Box"), fitdf = 4)
Box.test((res2^2), lag=14, type = c("Ljung-Box"), fitdf = 0)
ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# MODEL 3 #
########################################################################################################

fit3 <- model3_2
res3 <- residuals(fit3)
#plotting residual graphs of model 3
par(mfrow=c(2,2))
plot.ts(res3)
fitt3 <- lm(res3 ~ as.numeric(1:length(res3))); abline(fitt3, col="red")
abline(h=mean(res3), col="blue")
qqnorm(res3,main= "Normal Q-Q Plot for Model C")
qqline(res3,col="blue")
acf(res3, lag.max=40)
pacf(res3, lag.max=40)

#plotting histograms of model 3
par(mar = c(0.5, 4, 0.5, 0.5))
hist(res3,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m3 <- mean(res3)
std3 <- sqrt(var(res3))
curve( dnorm(x,m,std3), add=TRUE )
cat( paste("Mean: ", m3), paste("Standard Deviation: ", std3),sep="\n")

#tests for model 3
shapiro.test(res3)
Box.test(res3, lag=14, type = c("Box-Pierce"), fitdf = 5)
Box.test(res3, lag=14, type = c("Ljung-Box"), fitdf = 5)
Box.test((res3^2), lag=14, type = c("Ljung-Box"), fitdf = 0)
ar(res3, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

8. Forecasting

```
par(mfrow=c(2:1))
#pred.orig is reverted back from log by taking e power
pred.orig <- exp(pred.tr$pred)
U= exp(U.tr) #upper conf.
L= exp(L.tr) #lower conf.
#plotting forecast for log
ts.plot(popcorn.train, xlim=c(1,length(popcorn.train)+15), ylim = c(min(popcorn.train),
                                                                    max(U)+10), main="Popcorn forecaste
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(popcorn.train)+1):(length(popcorn.train)+15), pred.orig, col="red", type="l", lwd = 2, l
legend("topleft", legend = c("confidence interval","model forecast","training data"),
```

```r
       col=c("blue","red", "black"),lty=c(2,1,1),cex=1)

#plotting forecast original
ts.plot(popcorn$interest,ylab="Year",xlab="Time", col="red", type="l", main = "Actual vs. Forecast")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(popcorn.train)+1):(length(popcorn.train)+15), pred.orig, col="black",
       type="l",lty=1)
legend("topleft", legend = c("confidence interval","actual data set","model forecast"),
       col=c("blue","red", "black"),lty=c(2,1,1),cex=1)
```

9. Checking the AICcs

```r
#Checking AICCs
AICc(arima(popcorn.log, order=c(0,1,0), seasonal = list(order = c(0,1,0),
                                                          period = 12), method="ML"))
```

```
> [1] -203.8454
```

```r
AICc(arima(popcorn.log, order=c(0,1,0), seasonal = list(order = c(0,1,1),
                                                          period = 12), method="ML"))
```

```
> [1] -272.9499
```

```r
AICc(arima(popcorn.log, order=c(0,1,0), seasonal = list(order = c(0,1,2),
                                                          period = 12), method="ML"))
```

```
> [1] -275.4675
```

```r
AICc(arima(popcorn.log, order=c(0,1,0), seasonal = list(order = c(0,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -274.7808
```

```r
AICc(arima(popcorn.log, order=c(0,1,0), seasonal = list(order = c(0,1,1),
                                                          period = 12), method="ML"))
```

```
> [1] -272.9499
```

```r
AICc(arima(popcorn.log, order=c(0,1,1), seasonal = list(order = c(0,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -348.1442
```

```r
#One of Lowest preliminary AICc

AICc(arima(popcorn.log, order=c(1,1,1), seasonal = list(order = c(0,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -350.8507
```

```r
AICc(arima(popcorn.log, order=c(1,1,2), seasonal = list(order = c(0,1,3),
                                                     period = 12), method="ML"))
```

```
> [1] -348.9997
```

```r
AICc(arima(popcorn.log, order=c(1,1,0), seasonal = list(order = c(0,1,3),
                                                     period = 12), method="ML"))
```

```
> [1] -346.2563
```

```r
AICc(arima(popcorn.log, order=c(2,1,0), seasonal = list(order = c(0,1,3),
                                                     period = 12), method="ML"))
```

```
> [1] -349.3957
```

```r
AICc(arima(popcorn.log, order=c(3,1,0), seasonal = list(order = c(0,1,3),
                                                     period = 12), method="ML"))
```

```
> [1] -348.4912
```

```r
AICc(arima(popcorn.log, order=c(1,1,0), seasonal = list(order = c(0,1,3),
                                                     period = 12), method="ML"))
```

```
> [1] -346.2563
```

```r
AICc(arima(popcorn.log, order=c(1,1,1), seasonal = list(order = c(1,1,3),
                                                     period = 12), method="ML"))
```

```
> [1] -349.0841
```

```r
#MA(39)
AICc(arima(popcorn.log, order=c(39,1,0), seasonal = list(order = c(0,1,0), period = 12), method="ML"))
```

```
> [1] -312.4038
```

```r
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                     period = 12), method="ML"))
```

```
> [1] -351.4597
```

```r
#Process of removing coefficients for SARIMA(11,1,0)x(0,1,3)
AICc(arima(popcorn.log, order=c(11,1,0), seasonal = list(order = c(0,1,3),
                                                      period = 12),
           method="ML", fixed=c(NA,NA,0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -343.8436
```

```r
AICc(arima(popcorn.log, order=c(11,1,0), seasonal = list(order = c(0,1,3),
                                                         period = 12),
          method="ML", fixed=c(NA,NA,0,0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -345.8319
```

```r
AICc(arima(popcorn.log, order=c(11,1,0), seasonal = list(order = c(0,1,3),
                                                         period = 12),
          method="ML", fixed=c(NA,NA,0,0,0,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -347.7196
```

```r
AICc(arima(popcorn.log, order=c(11,1,0), seasonal = list(order = c(0,1,3),
                                                         period = 12),
          method="ML", fixed=c(NA,NA,0,0,0,0,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -349.6706
```

```r
AICc(arima(popcorn.log, order=c(11,1,0), seasonal = list(order = c(0,1,3),
                                                         period = 12),
          method="ML", fixed=c(NA,NA,0,0,0,0,0,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -351.361
```

```r
AICc(arima(popcorn.log, order=c(11,1,0), seasonal = list(order = c(0,1,3),
                                                         period = 12),
          method="ML", fixed=c(NA,NA,0,0,0,0,0,0,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -352.9783
```

```r
AICc(arima(popcorn.log, order=c(11,1,0), seasonal = list(order = c(0,1,3),
                                                         period = 12),
          method="ML", fixed=c(NA,NA,0,0,0,0,0,0,NA,NA,NA,NA,0,NA)))
```

```
> [1] -354.6205
```

```r
AICc(arima(popcorn.log, order=c(11,1,0), seasonal = list(order = c(1,1,3),
                                                         period = 12), method="ML"))
```

```
> [1] -340.8324
```

```r
AICc(arima(popcorn.log, order=c(11,1,1), seasonal = list(order = c(0,1,3),
                                                         period = 12), method="ML"))
```

```
> [1] -340.6537
```

```r
AICc(arima(popcorn.log, order=c(11,1,1), seasonal = list(order = c(1,1,3),
                                                      period = 12), method="ML"))
```

```
> [1] -338.6798
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                      period = 12), method="ML"))
```

```
> [1] -341.0217
```

```r
AICc(arima(popcorn.log, order=c(3,1,11), seasonal = list(order = c(0,1,3),
                                                      period = 12), method="ML"))
```

```
> [1] -349.2305
```

```r
##Process of removing coefficients for SARIMA(10,1,0)x(0,1,3)
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                      period = 12), method="ML"))
```

```
> [1] -341.0217
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                      period = 12), method="ML",
           fixed=c(0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -342.9616
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                      period = 12), method="ML",
           fixed=c(0,NA,0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -344.3422
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                      period = 12), method="ML",
           fixed=c(0,NA,0,0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -346.2664
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                      period = 12), method="ML",
           fixed=c(0,NA,0,0,0,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -348.2583
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                         period = 12), method="ML",
          fixed=c(0,NA,0,0,0,0,0,NA,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -351.9776
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                         period = 12), method="ML",
          fixed=c(0,NA,0,0,0,0,0,0,NA,NA,NA,NA,NA,NA,NA)))
```

```
> [1] -353.1975
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                         period = 12), method="ML",
          fixed=c(0,NA,0,0,0,0,0,0,NA,0,NA,NA,NA,NA,NA)))
```

```
> [1] -354.2705
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3),
                                                         period = 12), method="ML",
          fixed=c(0,NA,0,0,0,0,0,0,NA,0,NA,0,NA,NA,NA)))
```

```
> [1] -356.0898
```

```r
AICc(arima(popcorn.log, order=c(10,1,1), seasonal = list(order = c(1,1,3), period = 12), method="ML", fi
```

```
> [1] -358.0771
```

```r
AICc(arima(popcorn.log, order=c(3,1,11), seasonal = list(order = c(0,1,3),
                                                         period = 12), method="ML"))
```

```
> [1] -349.2305
```

```r
AICc(arima(popcorn.log, order=c(1,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12), method="ML"))
```

```
> [1] -350.2177
```

```r
AICc(arima(popcorn.log, order=c(1,1,9), seasonal = list(order = c(1,1,3),
                                                        period = 12), method="ML"))
```

```
> [1] -347.9888
```

```r
AICc(arima(popcorn.log, order=c(11,1,9), seasonal = list(order = c(0,1,3),
                                                         period = 12), method="ML"))
```

```
> [1] -343.3237
```

```r
#Lowest base AICc and lowest ending AICc
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12), method="ML",
           fixed=c(NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

> [1] -351.4597

```r
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12), method="ML",
           fixed=c(NA,NA,0,NA,NA,NA,NA,NA,NA,NA,NA,NA)))
```

> [1] -352.2956

```r
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12), method="ML",
           fixed=c(NA,NA,0,0,NA,NA,NA,NA,NA,NA,NA,NA)))
```

> [1] -354.1128

```r
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12), method="ML",
           fixed=c(NA,NA,0,0,0,NA,NA,NA,NA,NA,NA,NA)))
```

> [1] -356.0661

```r
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12), method="ML",
           fixed=c(NA,NA,0,0,0,0,NA,NA,NA,NA,NA,NA)))
```

> [1] -357.6547

```r
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12), method="ML",
           fixed=c(NA,NA,0,0,0,0,0,NA,NA,NA,NA,NA)))
```

> [1] -359.3689

```r
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12), method="ML",
           fixed=c(NA,NA,0,0,0,0,0,0,NA,NA,NA,NA)))
```

> [1] -361.1653

```r
AICc(arima(popcorn.log, order=c(0,1,9), seasonal = list(order = c(0,1,3),
                                                        period = 12),
           method="ML",fixed=c(NA,NA,0,0,0,0,0,0,NA,NA,0,NA)))
```

> [1] -362.9692

```r
AICc(arima(popcorn.log, order=c(2,1,2), seasonal = list(order = c(0,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -346.9792
```

```r
AICc(arima(popcorn.log, order=c(1,1,2), seasonal = list(order = c(0,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -348.9997
```

```r
AICc(arima(popcorn.log, order=c(1,1,3), seasonal = list(order = c(0,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -346.9499
```

```r
AICc(arima(popcorn.log, order=c(2,1,0), seasonal = list(order = c(1,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -347.8191
```

```r
AICc(arima(popcorn.log, order=c(3,1,2), seasonal = list(order = c(0,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -344.8062
```

```r
AICc(arima(popcorn.log, order=c(1,1,3), seasonal = list(order = c(1,1,3),
                                                          period = 12), method="ML"))
```

```
> [1] -345.1107
```