Growth of Renewable Energy Consumption in the United States

Angel Chen

6/5/2020

Abstract

Renewable energy consumption has been increasing in the United States over the past 20 years as people sought alternatives to fossil fuels. Sources of renewable energy include hydroelectric power, geothermal, solar, wind, and biomass. In this report, I investigate the total monthly consumption of renewable energy in the United States from January 2001 to January 2020 and try to figure out a model that can be used to forecast how much renewable energy will be used in the future.

In order to achieve this goal, I used the Box-Jenkins methodology. This method involves data transformation, differencing, examining autocorrelation and partial autocorrelation functions, model parameter estimation, checking for stationarity and invertibility, and diagnostic checking. To summarize my results, I came up with three plausible models for the data, but ended up with two satisfactory models that passed all checking. I concluded that one model was better than the other because it had lower AICc, an information criterion, and used it to predict renewable energy consumption for February 2019 to January 2020. My predictions followed the actual data values well, meaning that my model can be used for future forecasting, and it also supports the conclusion that renewable energy consumption is on an upward trend.

Introduction

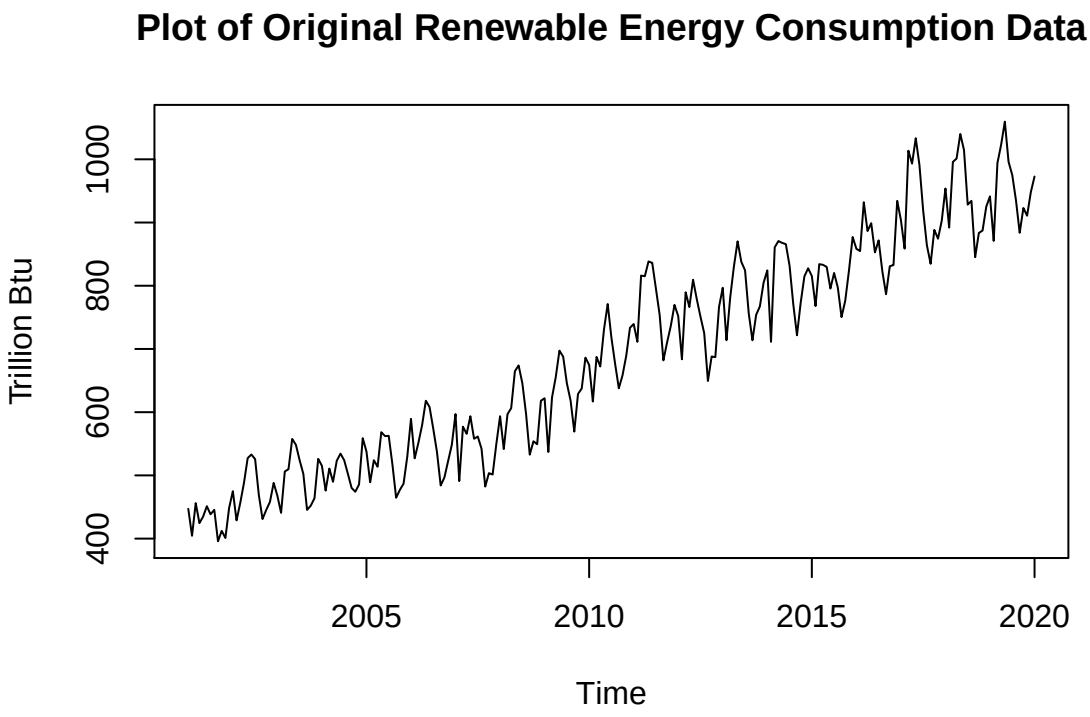
To restate, the purpose of this report is to find a suitable model that can be used to forecast total monthly renewable energy consumption in the United States. The data set was acquired from the US Energy Information Administration and be found at https://www.eia.gov/totalenergy/data/monthly/ under "Renewable Energy - Production and consumption by source." Total renewable energy consumption includes these sources: hydroelectric power, geothermal, solar, wind, and biomass (wood, waste, and biofuels). Monthly energy consumption is recorded in trillions of British thermal units (Btu), which is the amount of heat required to raise one pound of water by 1 degree Fahrenheit. The data set is from January 2001 to January 2020 with 229 observations.

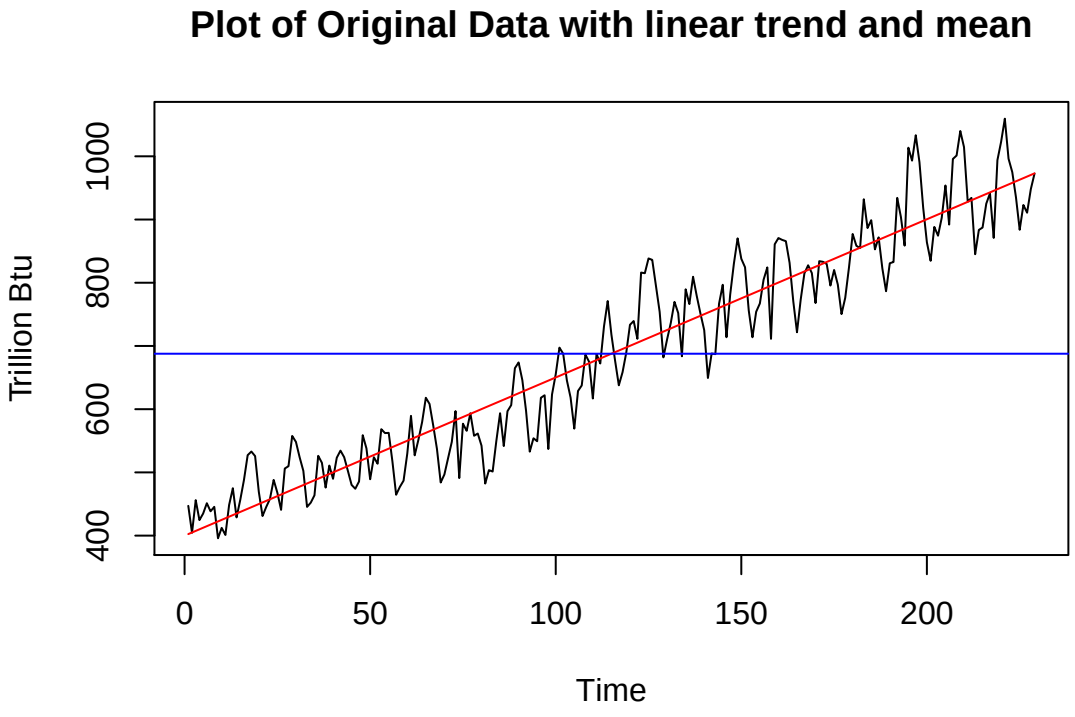
I decided to work on this data set because I wish for more people to realize that fossil fuels like coal, oil, and natural gas are slowly but surely becoming a relic of the past as more countries are making the switch to clean energy. The huge rise in renewable energy consumption in recent years gives hope that we will be able to dramatically reduce our carbon footprint on Earth someday.

To find a good model for this data set, I followed the Box-Jenkins methodology with the help of RStudio. First, I split the data set into a training set and a test set. Next, I chose an appropriate transformation for the training data to stabilize variance and differenced at lag 12 and lag 1 to remove seasonality and trend. Then, I examined the autocorrelation (ACF) and partial autocorrelation functions (PACF) to prepare three different Seasonal Autoregressive Integrated Moving Average (SARIMA) models to test. After checking for stationarity and invertibility of the models, I found that only two of them were actually stationary and invertible.

I performed diagnostic checking on both models by inspecting plots of their residuals and making sure that they pass the Shapiro-Wilk, Box-Pierce, Ljung-Box, and McLeod-Li tests. In the end, I chose the model with the lower AICc, an information criterion, and made periodograms of the data and the model's residuals, which showed a period of 12 months. Next, I used my model to make predictions of renewable energy consumption for February 2019 to January 2020 and compared them with the actual values of the testing data. Since my predictions followed the actual values of the testing data well, I concluded that I had an adequate model that can be used for forecasting. Finally, I used that model to forecast even further in the future from February 2020 to January 2021 and my model supported the idea that renewable energy consumption is on an upward trend.

The Time Series



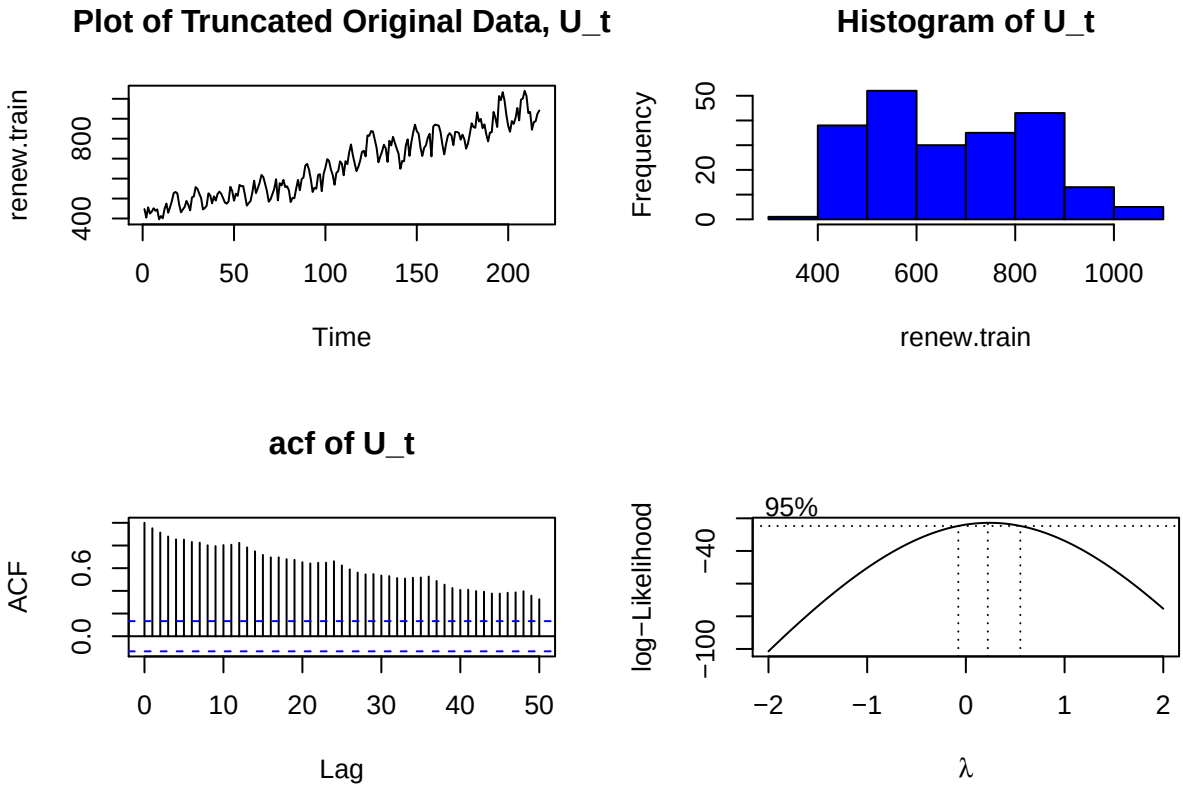


To start off, I plotted the time series from January 2001 to January 2020 and realized that there was a

positive, linear trend that I would have to remove later. I also noticed that there seems to be seasonality in this data set because the plot oscillates. Perhaps there were certain months in the year when Americans consumed more renewable energy, like during the summer for instance. Additionally, the variance increased from 2010 to 2020 because the plot varied a lot more compared to 2001 to 2009. Other than that, there are no apparent changes in behavior.

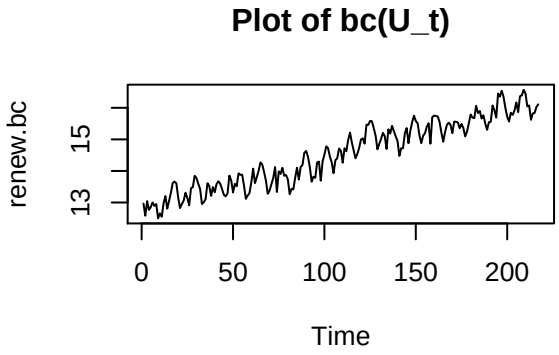
Transformations

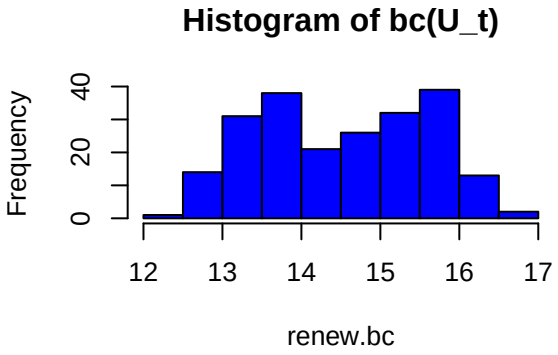
I wanted to test out how well my model fits the data later so I split the data into training and test sets. The last 12 months of the data set became my test set, so my training set, U_t, contained data from January 2001 to January 2019. Looking at the histogram of U_t, I could tell that the variance was not stable because the histogram was skewed and non-symmetric. The autocorrelation function (ACF) of U_t decreases slowly, and showed a pattern, which meant that there was a seasonal component involved. To stabilize the variance a bit, I needed to transform the data. I plotted the confidence interval to help me choose the right lambda for Box-Cox transformation.

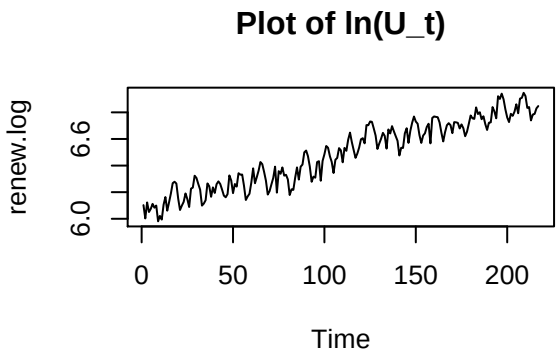


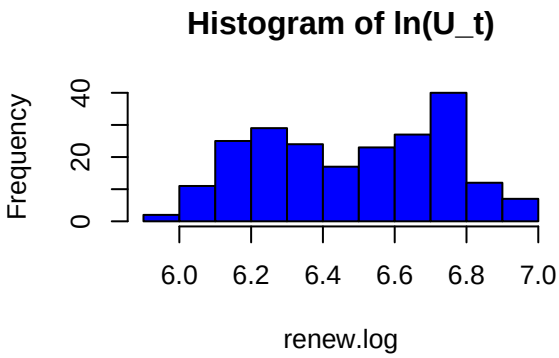

```
## [1] 0.2222222
```

Box-Cox transformation suggested that $\lambda = 0.2222$, but the histogram of the Box-Cox transformed data did not look that good. Since 0 was in the confidence interval as well, I decided to try $\lambda = 0$, which is a log transform. Although not perfect, the histogram for the log transformed data seemed like a slight improvement over the histogram of the Box-Cox transformed data. The plots of both transformed data sets were very similar anyway, so I chose the log transformation.

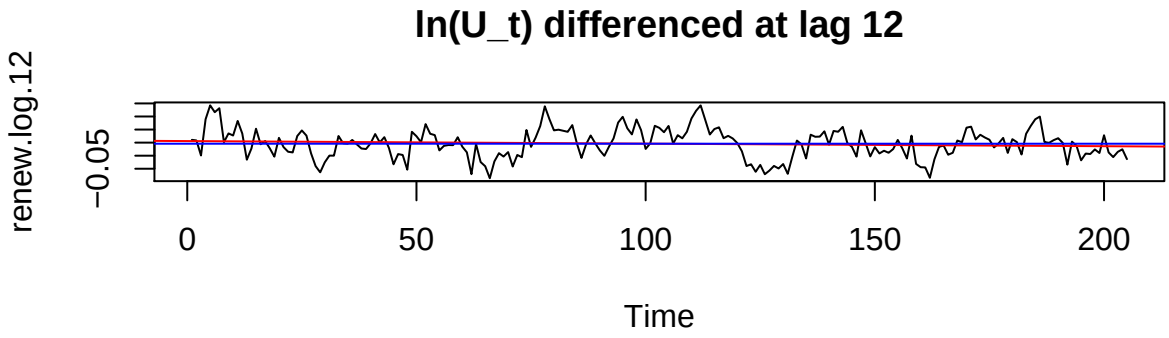


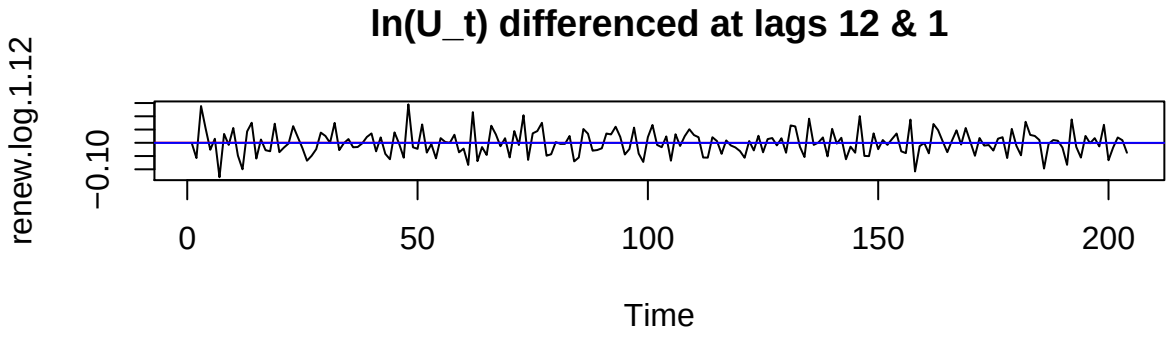




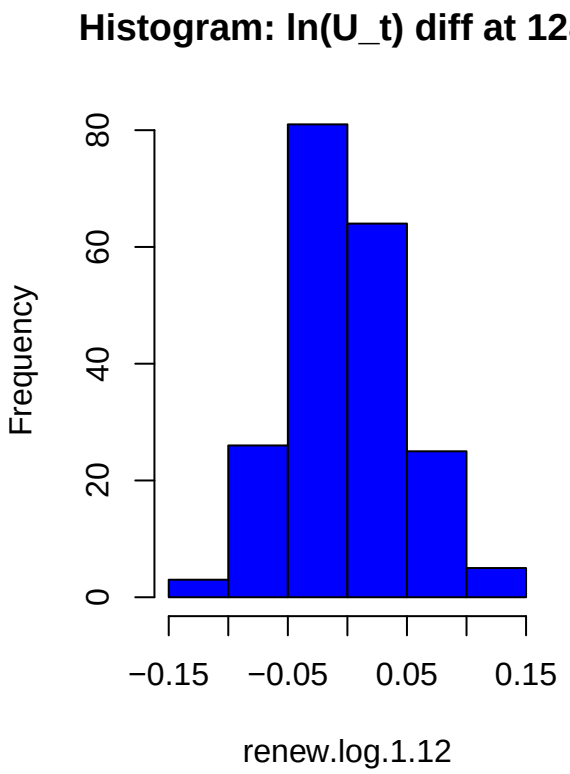


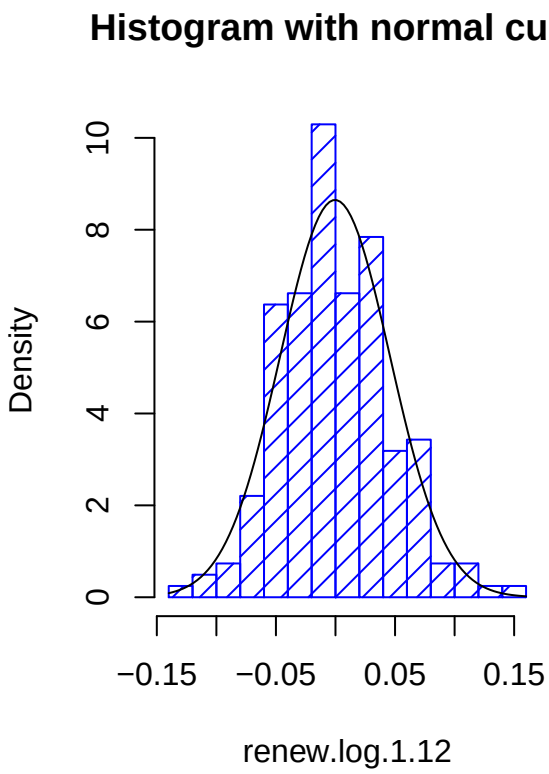
Since the Box-Jenkins methodology works well for Gaussian time series, I must make sure that my data is stationary with no trend, no change of variance, and no seasonality. Since the log transform took care of the unstable variance, I differenced $ln(U_t)$ at lag 12 to remove seasonality because this was a monthly data set. After differencing at lag 12, there was a small trend (in red) remaining, so I differenced again at lag 1 to remove trend. At this point, the time series is stationary. To avoid over-differencing, I kept track of the variance and it decreased with each difference, which was a good sign. If I differenced at lag 1 again, the variance would increase, so I stopped.






```
##                                                dataset          mean    variance
## 1                                              ln(U_t)  6.4806295415 0.062664425
## 2                          ln(U_t) differenced at lag 12  0.0457837251 0.003154825
## 3               ln(U_t) differenced at lag 12 and lag 1 -0.0003618073 0.002127804
## 4 ln(U_t) differenced at lag 12 once and lag 1 twice -0.0001741697 0.005268722
```

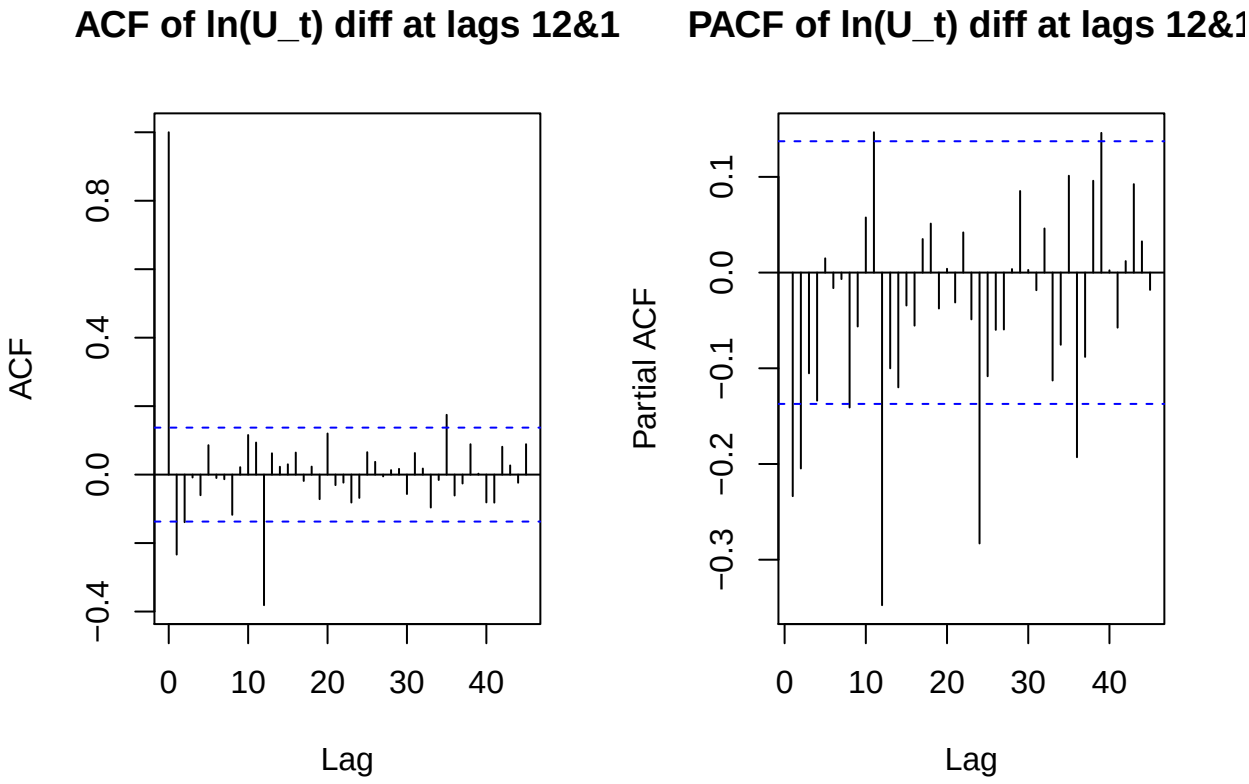




Since the histogram of $\nabla_1 \nabla_{12} ln(U_t)$ looks roughly symmetric and Gaussian, I proceeded to the next step and analyzed its ACF and PACF to do some preliminary identifications for possible models.

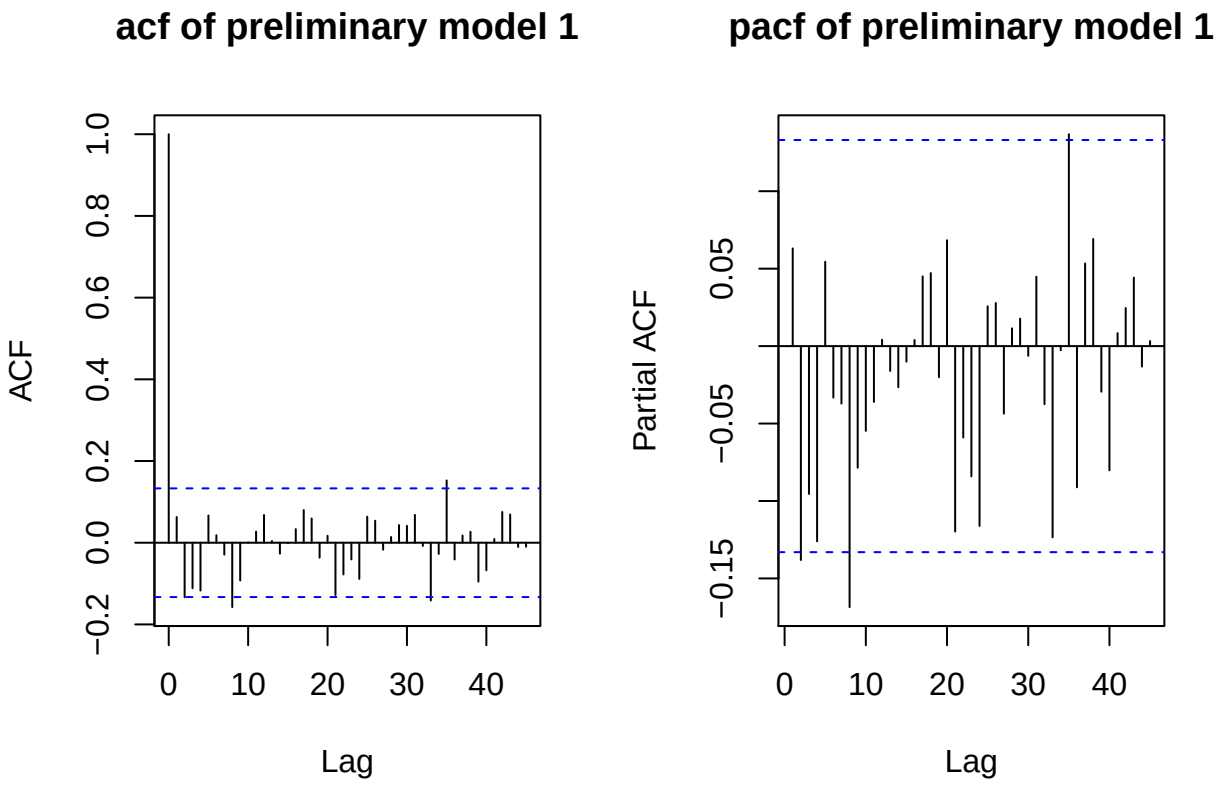
Preliminary Model Identifications

My data had a seasonal component, so I knew that I wanted to use SARIMA models, with s=12 because of a period of 12 months, d=1 because of differencing at lag 1, and D=1 because of differencing at lag 12. The difficult part is determining the order for the other coefficients, like p, P, q, and Q. I plotted the sample ACF and PACF to help me decide. On the ACF plot, I can see that lags 1, 12 and possibily 35 were sticking out of the confidence interval, which meant that they were significant. Perhaps q=1 and Q=1 or 3. I also noticed that on the PACF plot, lags 1, 2, 12, 24, and 36 were significant. Perhaps p=1 or 2 and P=3.



SARIMA models were tricky to fit since there were many coefficients I had to consider, so I started with a small pure seasonal moving average (SMA) model with q=1, Q=1. The residuals' ACF and PACF plots will be able to guide me into improving my model.

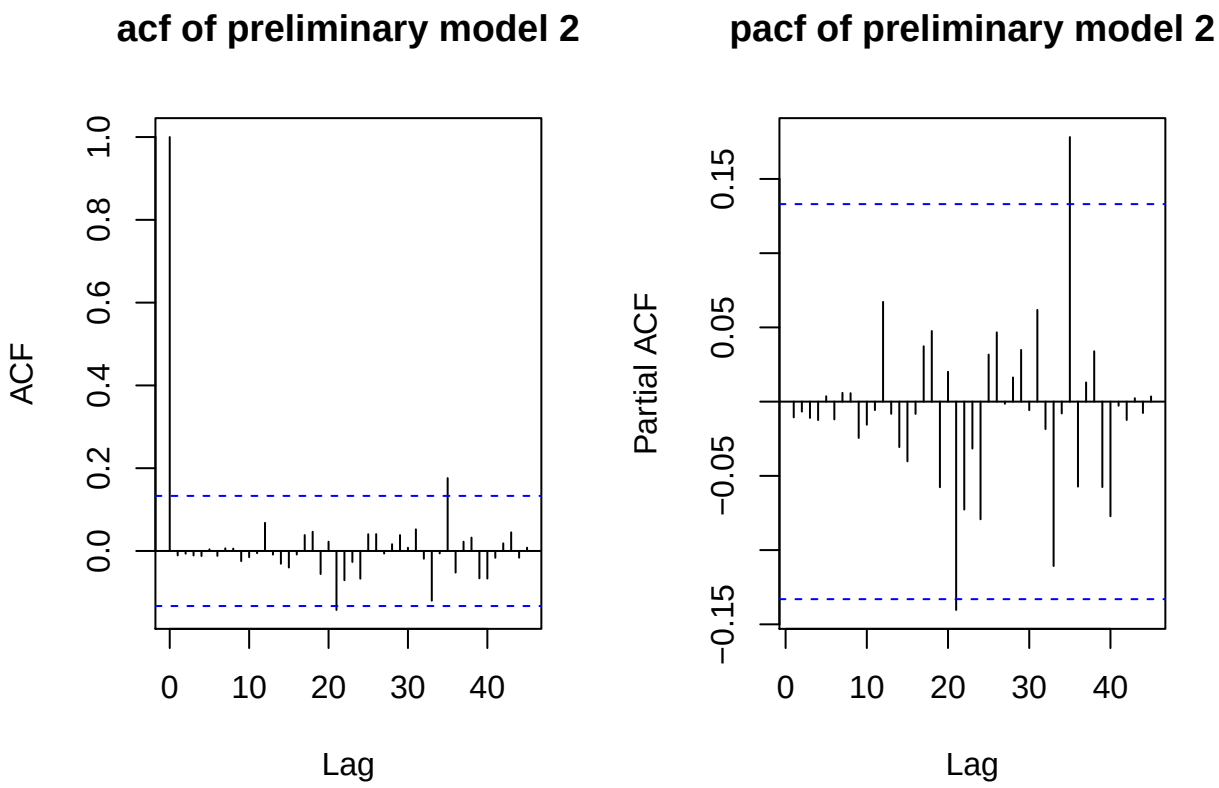
```
##
## Call:
## arima(x = renew.log, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -0.3773  -0.7870
## s.e.   0.0843   0.0532
##
## sigma^2 estimated as 0.001229:  log likelihood = 388.19,  aic = -770.38
```



Looking at these plots, I could tell that there were lots of improvements to be made. ACF sticks up from the confidence interval at lags 8 and 35 while PACF sticks up at lag 8. I did not want to try fitting a high order of coefficients and I wanted to see the effects of a mixed model so I added p=8 to see if my model would improve.

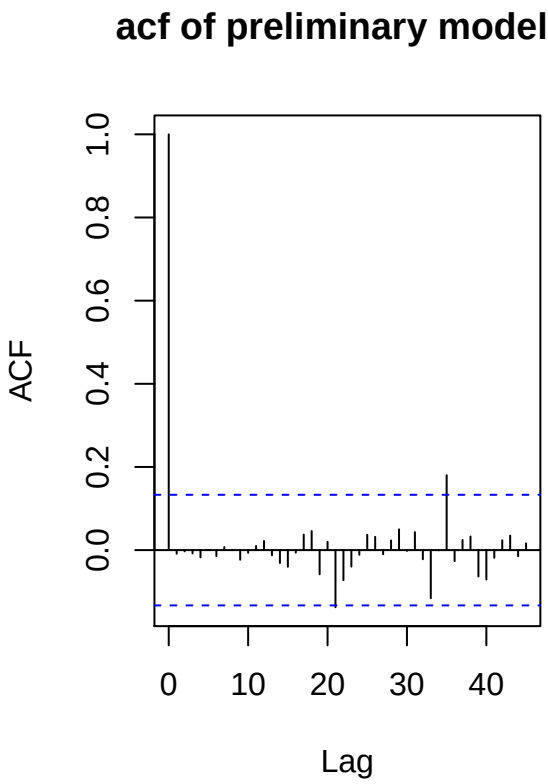
```
##
## Call:
## arima(x = renew.log, order = c(8, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      ar4     ar5      ar6      ar7      ar8
##       0.3747  -0.0665  -0.0058  -0.0900  0.0958  -0.0376  -0.0264  -0.1575
## s.e.  0.1779   0.0898   0.0887   0.0831  0.0831   0.0776   0.0774   0.0784
##           ma1     sma1
```

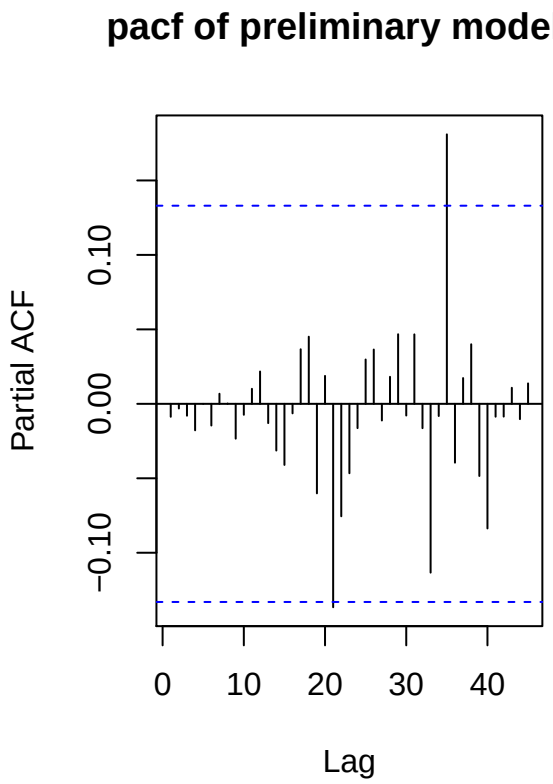
```
##       -0.7004  -0.8616
## s.e.   0.1734   0.0649
##
## sigma^2 estimated as 0.001102:  log likelihood = 396.91,  aic = -771.81
```



Now, both ACF and PACF stick up at lag 35, demonstrating that I could improve the model again. Also, all coefficients for the autoregressive (AR) part of the model except for the first and last coefficients are insignificant (since 0 is within their 95% confidence interval), so I may consider setting them to zero later on. Since ACF sticks up at lag 35, I decided to change Q to 4.

```
##
## Call:
## arima(x = renew.log, order = c(8, 1, 1), seasonal = list(order = c(0, 1, 4),
##     period = 12), method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      ar4     ar5      ar6      ar7      ar8
##       0.4113  -0.0584  -0.0072  -0.0698  0.1016  -0.0325  -0.0155  -0.1486
## s.e.  0.2110   0.0967   0.0951   0.0906  0.0878   0.0790   0.0796   0.0812
##           ma1     sma1     sma2    sma3    sma4
##       -0.7333  -0.8116  -0.1076  0.0355  0.0430
## s.e.   0.2086   0.0816   0.1011  0.1047  0.0904
##
## sigma^2 estimated as 0.001096:  log likelihood = 397.62,  aic = -767.24
```





The result is that the ACF and PACF did not change at all. They are still significant at lag 35. Perhaps Q should not be 4 after all. Additionally, the second, third, and fourth coefficients for the SMA part of the model are insignificant. I suspect that Q is best left as 1. I suspect that I may be able to lower the order for p as well, due to how so many coefficients for the AR part have 0 in their confidence intervals.

Model Fitting

Model 1

Using the results of the preliminary model identification, I have a good idea on what p, q, P and Q should be. I decided to try p=3, q=3, P=0, and Q=1. I wanted to lower the AICc of the model and find a way to make all coefficients significant, so I experimented with setting coefficients to zero, and I found out that setting the second AR coeffcient to zero resulted in lowering the AICc from -771.8 to -775.5. This model with the lower AICc is Model 1.

```
##
## Call:
## arima(x = renew.log, order = c(3, 1, 3), seasonal = list(order = c(0, 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##           ar1     ar2     ar3     ma1      ma2      ma3     sma1
##       -0.8272  0.2296  0.3248  0.5049  -0.6816  -0.4470  -0.8237
## s.e.   1.1553  0.4101  0.4041  1.1746   0.1495   0.7861   0.0592
##
## sigma^2 estimated as 0.001145:  log likelihood = 394.16,  aic = -772.32
```

```
##
## Call:
## arima(x = renew.log, order = c(3, 1, 3), seasonal = list(order = c(0, 1, 1),
##     period = 12), fixed = c(NA, 0, NA, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##           ar1  ar2     ar3     ma1     ma2      ma3     sma1
##       -0.9648    0  0.6254  0.6779  -0.437  -0.8923  -0.8243
## s.e.   0.0120    0  0.0034  0.0372   0.054   0.0385   0.0559
##
## sigma^2 estimated as 0.001125:  log likelihood = 395.01,  aic = -776.02
```

```
## [1] -771.7838
```

```
## [1] -775.4874
```

Model 2

Out of curiosity, I wanted to see what would happen if I set p to a high order, like 35, because during my preliminary investigation, lag 35 kept sticking up no matter what. After experimenting with setting insignificant coefficients to zero for a while, I found that this combination of coefficients gave me an AICc of -771, which was not as good as the previous model. This is Model 2 with p=35, q=1, P=0, and Q=1.

```
##
## Call:
## arima(x = renew.log, order = c(35, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 12), fixed = c(NA, 0, 0, 0, 0, 0, 0, NA, 0, 0, 0, 0, 0, 0, 0, 0,
##     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, NA, NA, NA), method = "ML")
##
## Coefficients:
##          ar1  ar2  ar3  ar4  ar5  ar6  ar7      ar8  ar9  ar10  ar11  ar12
##       0.3826    0    0    0    0    0    0  -0.1582    0     0     0     0
## s.e.  0.1035    0    0    0    0    0    0   0.0648    0     0     0     0
##       ar13  ar14  ar15  ar16  ar17  ar18  ar19  ar20  ar21  ar22  ar23  ar24
##          0     0     0     0     0     0     0     0     0     0     0     0
```

```
## s.e.     0     0     0     0     0     0     0     0     0     0     0     0
##       ar25  ar26  ar27  ar28  ar29  ar30  ar31  ar32  ar33  ar34    ar35
##          0     0     0     0     0     0     0     0     0     0  0.2028
## s.e.     0     0     0     0     0     0     0     0     0     0  0.0723
##           ma1     sma1
##       -0.7325  -0.8791
## s.e.   0.0860   0.0613
##
## sigma^2 estimated as 0.001064:  log likelihood = 399.38,  aic = -786.75
```

Model 3

Finally, I wondered if a mixed model or a non-mixed model would be better. I made a pure SMA model by switching things around and set p=0, q=35, P=0, Q=1. I tried setting many different combinations of insignificant coefficients to zero, and discovered that this combination gave me an AICc of -773.5. This is Model 3.

```
##
## Call:
## arima(x = renew.log, order = c(0, 1, 35), seasonal = list(order = c(0, 1, 1),
##     period = 12), fixed = c(NA, NA, 0, 0, 0, 0, 0, NA, 0, 0, 0, 0, 0, 0, 0,
##     0, 0, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, 0, 0, 0, 0, NA, 0, NA, NA), method = "ML")
##
## Coefficients:
##           ma1      ma2  ma3  ma4  ma5  ma6  ma7      ma8  ma9  ma10  ma11  ma12
##       -0.6381  -0.3184    0    0    0    0    0  -0.2928    0     0     0     0
## s.e.   0.0863   0.1006    0    0    0    0    0   0.0609    0     0     0     0
##       ma13  ma14  ma15  ma16  ma17  ma18  ma19  ma20     ma21  ma22  ma23  ma24
##          0     0     0     0     0     0     0     0  -0.2465     0     0     0
## s.e.     0     0     0     0     0     0     0     0   0.0858     0     0     0
##         ma25  ma26  ma27  ma28  ma29  ma30  ma31  ma32     ma33  ma34    ma35
##       0.2812     0     0     0     0     0     0     0  -0.2380     0  0.3701
## s.e.  0.1113     0     0     0     0     0     0     0   0.0825     0  0.0965
##          sma1
##       -0.8747
## s.e.   0.0645
##
## sigma^2 estimated as 0.0007698:  log likelihood = 403.13,  aic = -788.26
```

Summary of Models

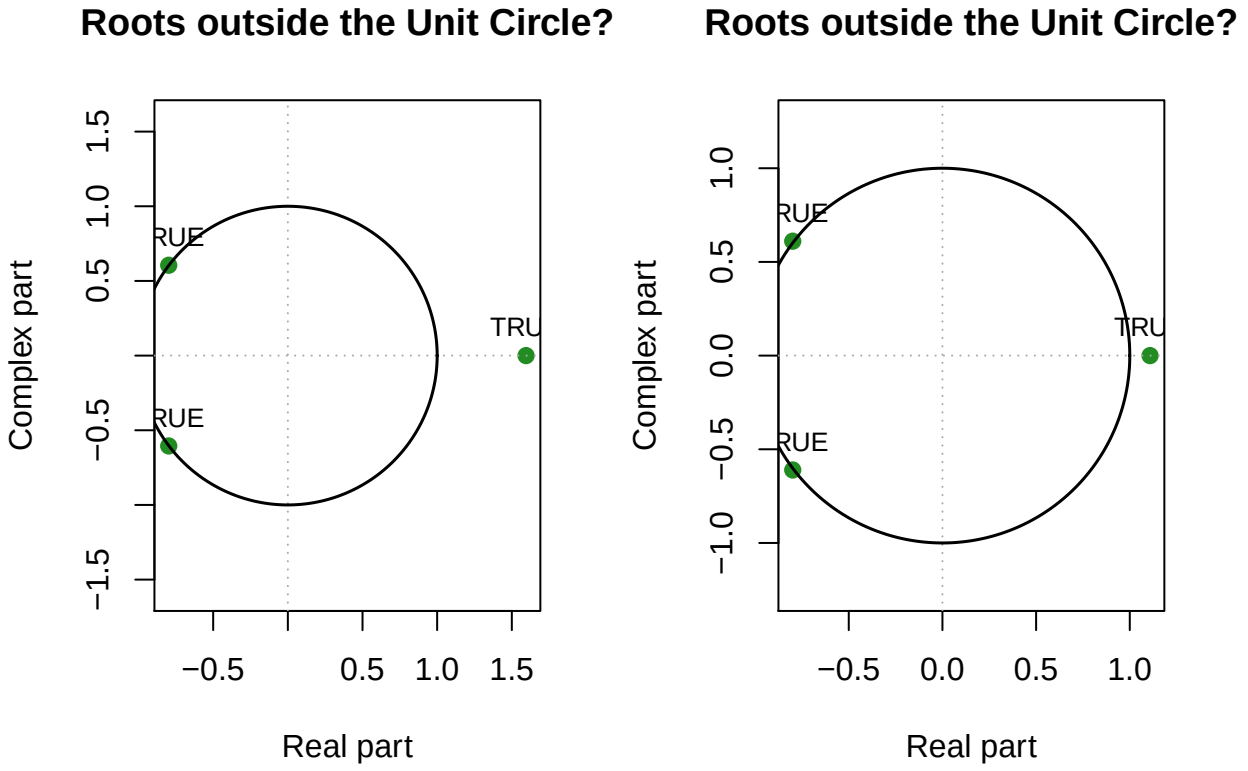
Here is a quick summary of the models I am considering:

```
##     model                       type nonzero_coefficients      AICc
## 1 Model 1  SARIMA(3,1,3)x(0,1,1)_12                     6 -775.4874
## 2 Model 2 SARIMA(35,1,1)x(0,1,1)_12                     5 -771.0443
## 3 Model 3 SARIMA(0,1,35)x(0,1,1)_12                     8 -773.4608
```

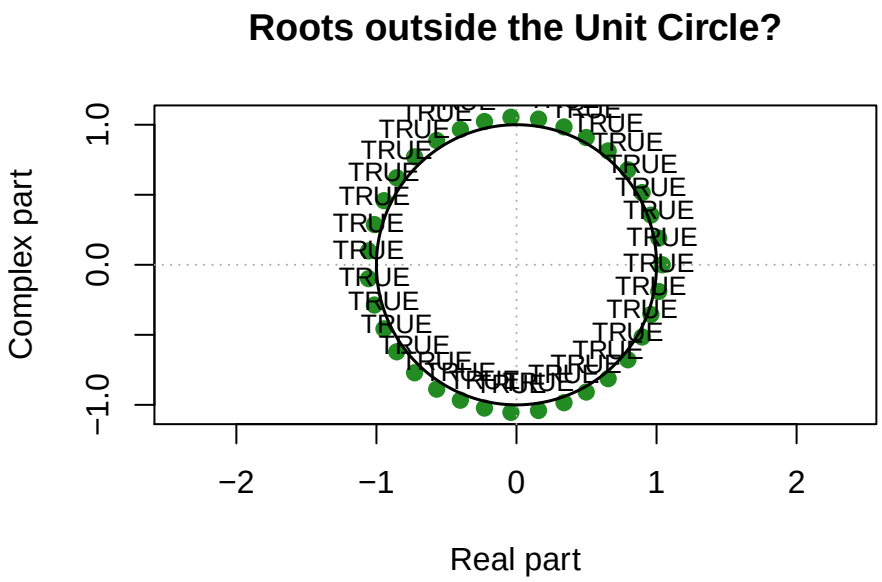
Diagnostic Checking

Before moving on, I needed to check that the models I picked were stationary and invertible. This meant checking whether the roots of the polynomials in the model fall inside or outside the unit circle. Polynomials making up the AR part were invertible, but may not be stationary. Meanwhile, polynomials for the MA part were stationary, but not necessarily invertible.

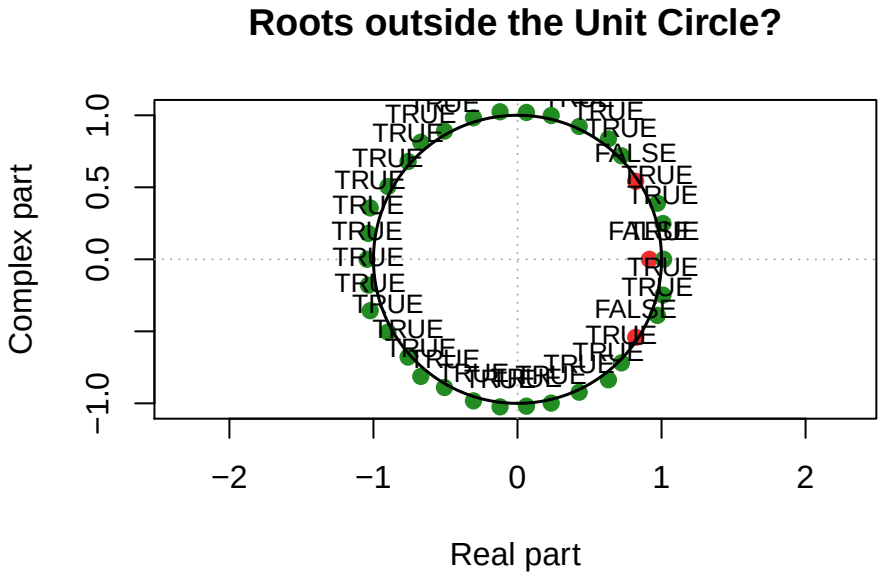
For Model 1, the roots of the non-seasonal AR part were outside the circle, so the AR part is stationary. The absolute value of the seasonal MA coefficient -0.8237 is less than 1, and the roots of the non-seasonal MA part were outside the unit circle, as well, so the MA part is invertible. Model 1 is stationary and invertible.



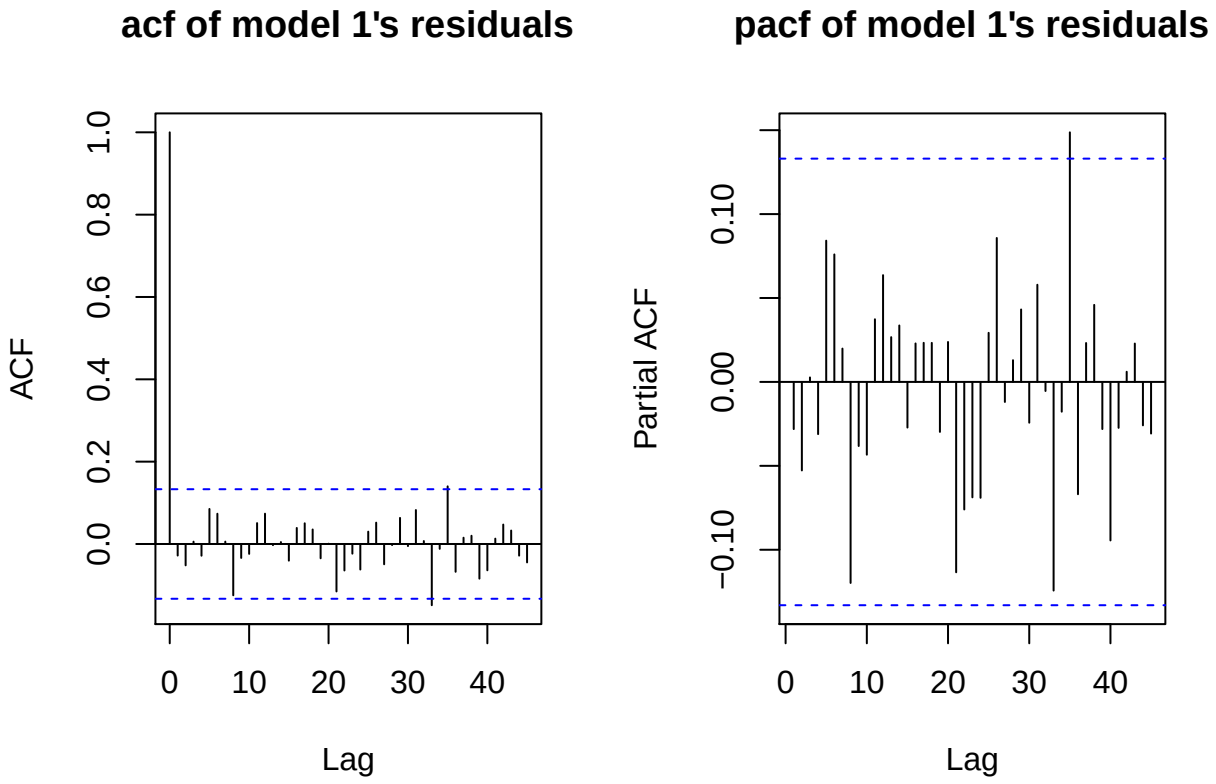
For Model 2, the roots of the non-seasonal AR part were outside the unit circle, so the AR part is stationary. Since both the non-seasonal and seasonal MA coefficients, -0.7325 and -0.8791 respectively, have absolute values that were less than 1, the MA part is invertible. Therefore, Model 2 is stationary and invertible.



For Model 3, some of the roots of the non-seasonal MA part were inside the unit circle, which meant that the MA part was not invertible. Therefore, Model 3 is not stationary and invertible. I discard Model 3.



With Model 3 out of the way, I move onto checking diagnostics for Models 1 and 2. If the fit was good, then the model's residuals should resemble Gaussian white noise. Starting with Model 1, I checked whether the residuals' ACF and PACF stayed within the confidence interval. Both plots looked alright, except that PACF stuck out at lag 35, but it should be fine since RStudio overestimates p.



The histogram of Model 1's residuals look approximately normal with sample mean zero and the time series looks like white noise with no visible trend or seasonality. The QQ-plot has most of its points falling into a straight line, which is good. The p-values for the Shapiro-Wilk normality test, Box-Pierce test, Ljung-Box test, and McLeod-Li test were all greater than 0.05, so my residuals were approximately normal and uncorrelated. To top it off, the residuals were fitted to AR(0), white noise, so Model 1 is ready for forecasting.

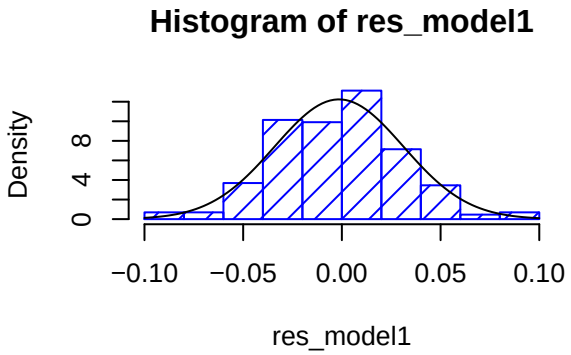
```
##
##  Shapiro-Wilk normality test
##
## data:  res_model1
## W = 0.99148, p-value = 0.2361
```

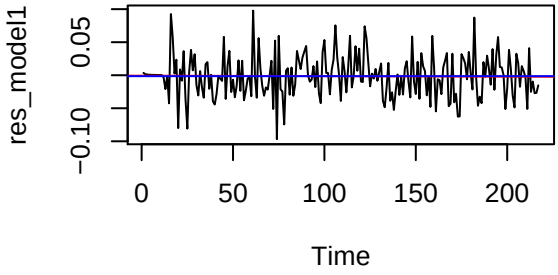
```
##
##  Box-Pierce test
##
## data:  res_model1
## X-squared = 9.5347, df = 9, p-value = 0.3895
```

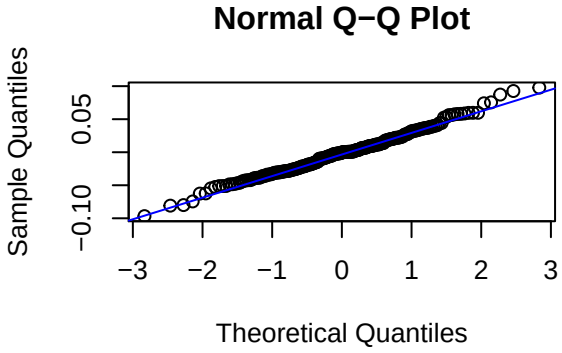
```
##
##  Box-Ljung test
##
## data:  res_model1
## X-squared = 9.9775, df = 9, p-value = 0.3523
```

```
##
##  Box-Ljung test
##
## data:  (res_model1)^2
## X-squared = 10.878, df = 15, p-value = 0.7612
```

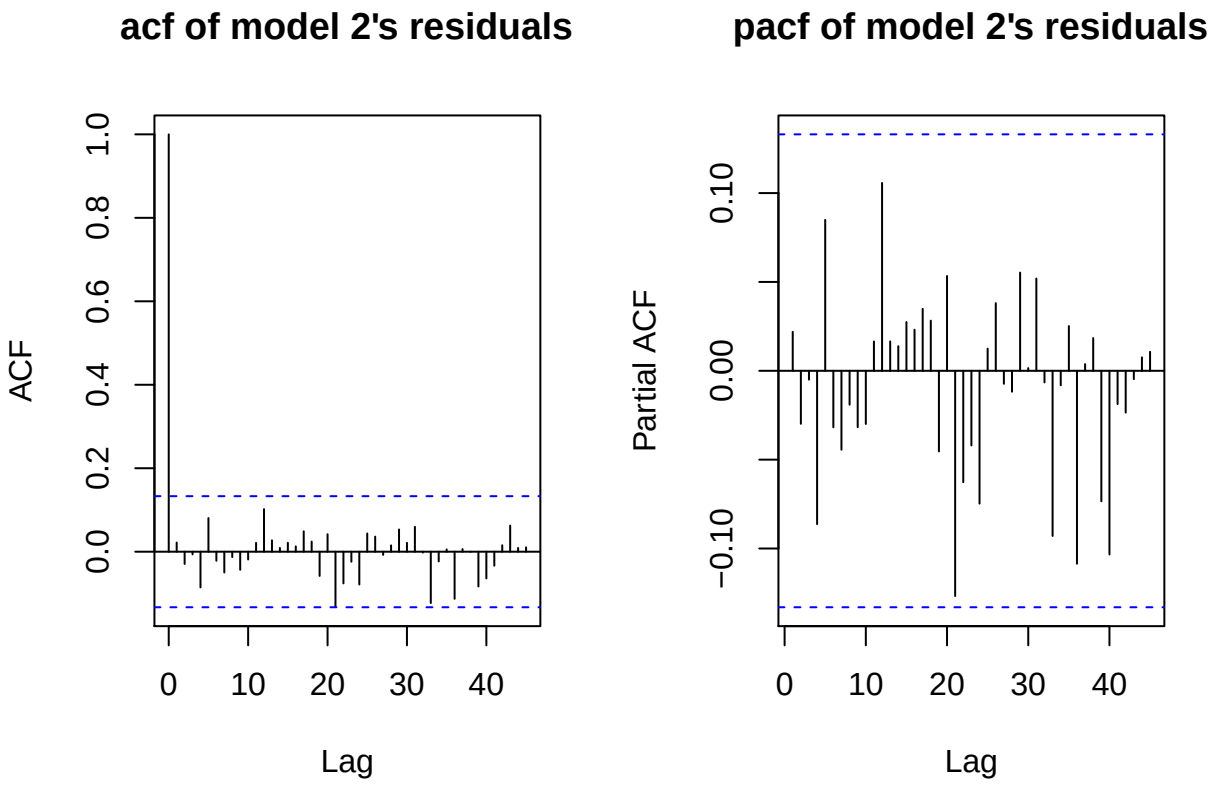
```
##
## Call:
## ar(x = res_model1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.001063
```







I repeat the same procedure for Model 2's residuals. Its ACF and PACF stayed within the confidence interval.



The histogram looked approximately normal with mean zero, while the time series shows no trend or seasonality. The QQ-plot looks straight, too. Once again, the p-values for all tests were greater than 0.05, so my residuals were approximately normal and uncorrelated at 95% confidence level. The residuals were able to fit to AR(0), and now Model 2 is ready for forecasting.

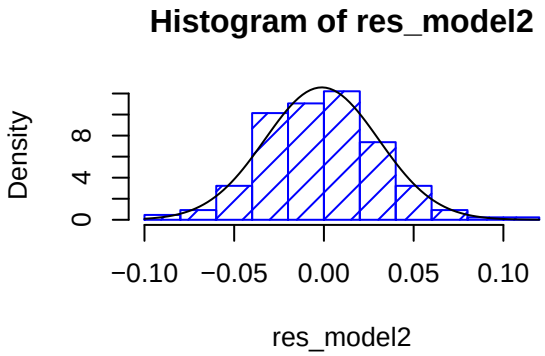
```
##
##  Shapiro-Wilk normality test
##
## data:  res_model2
## W = 0.99304, p-value = 0.4002
```

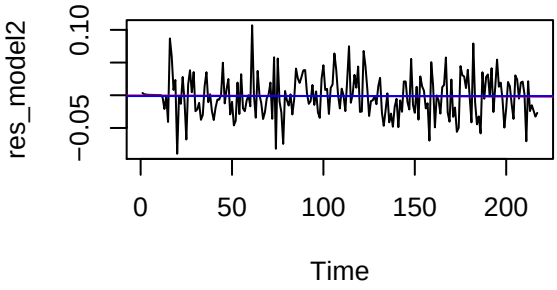
```
##
##  Box-Pierce test
##
## data:  res_model2
## X-squared = 7.1012, df = 10, p-value = 0.7159
```

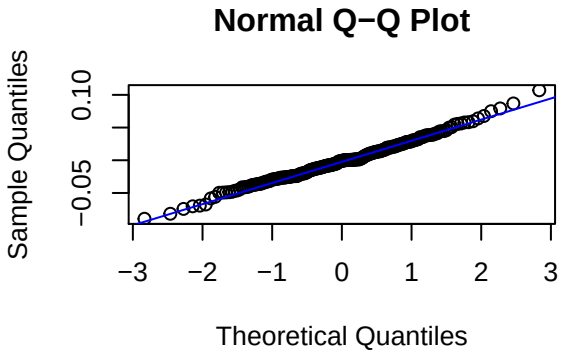
```
##
##  Box-Ljung test
##
## data:  res_model2
## X-squared = 7.4348, df = 10, p-value = 0.6839
```

```
##
##  Box-Ljung test
##
## data:  (res_model2)^2
## X-squared = 5.0993, df = 15, p-value = 0.9913
```

```
##
## Call:
## ar(x = res_model2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.001005
```







Now that both Model 1 and Model 2 passed all diagnostic checking, I must pick one to actually forecast with. Model 1 had an AICc of -775.5 while Model 2 had an AICc of -771, so strictly going by the lower AICc, I pick Model 1 as my final model. It is SARIMA(3,1,3)x$(0,1,1)_{12}$ with 6 non-zero coefficients.

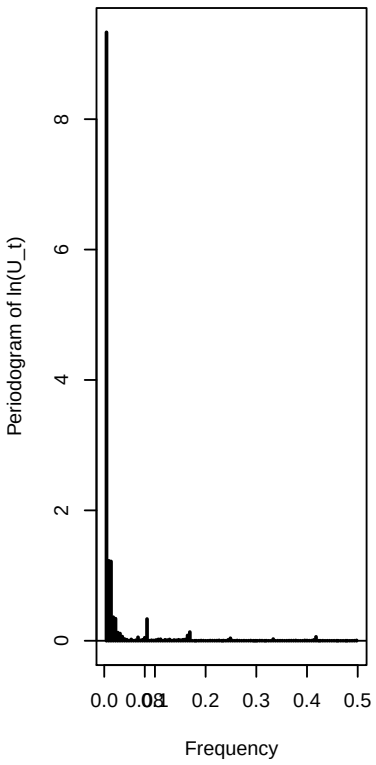
It can be written as: $(1+0.9648B-0.6254B^3)(1-B)(1-B^{12})ln(U_t) = (1+0.6779B-0.437B^2-0.8923B^3)(1-0.8243B^{12})Z_t$ with $\hat{\sigma}_Z^2 = 0.001125$.

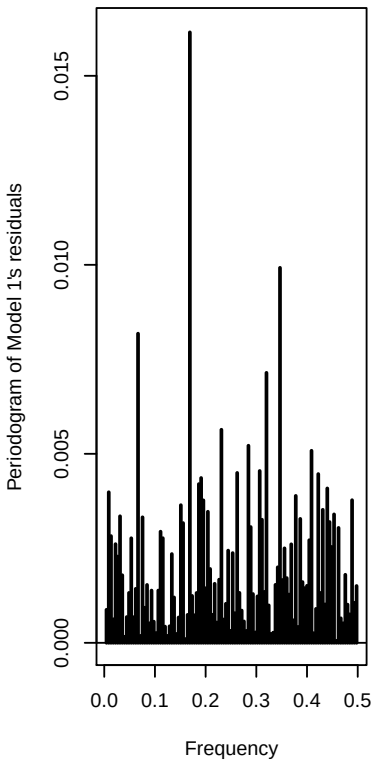
At first during my preliminary model identification, the ACF and PACF suggested that p=1 or 2, q=1, P=3, and Q=1, which is different than what I ended up with using the AICc. I found that generally, one seasonal part was enough, and there was no need to add in both seasonal MA and AR parts, otherwise the AICc would increase.

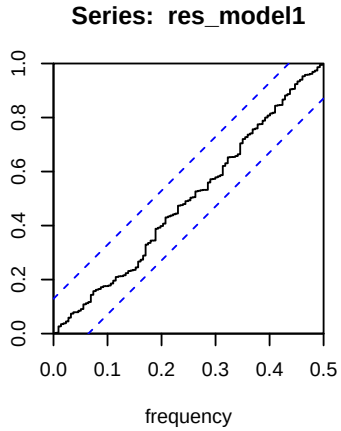
Spectral Analysis

Spectral analysis is helpful in determining periods of a seasonal time series. In the previous parts, I have assumed that my period is 12 because it was monthly data, but I can plot periodograms to confirm. The periodogram of $ln(U_t)$ showed a significant spike at around frequency 0.084. Period is 1/frequency, so the periodogram selected a period of 12 months. No frequency dominates the periodogram of the residuals because it is all over the place, hence the residuals of Model 1 resemble white noise.

I have also used Fisher's test for periodicity detection on Model 1's residuals, and since the p-value, 0.4681, is greater than 0.05, I fail to reject the null hypothesis that the residuals are Gaussian white noise. I conducted Kolmogorov-Smirnov test to the residuals as well, and since nothing is outside the boundaries, I fail to reject the same null hypothesis. Model 1 is completely suitable for forecasting.

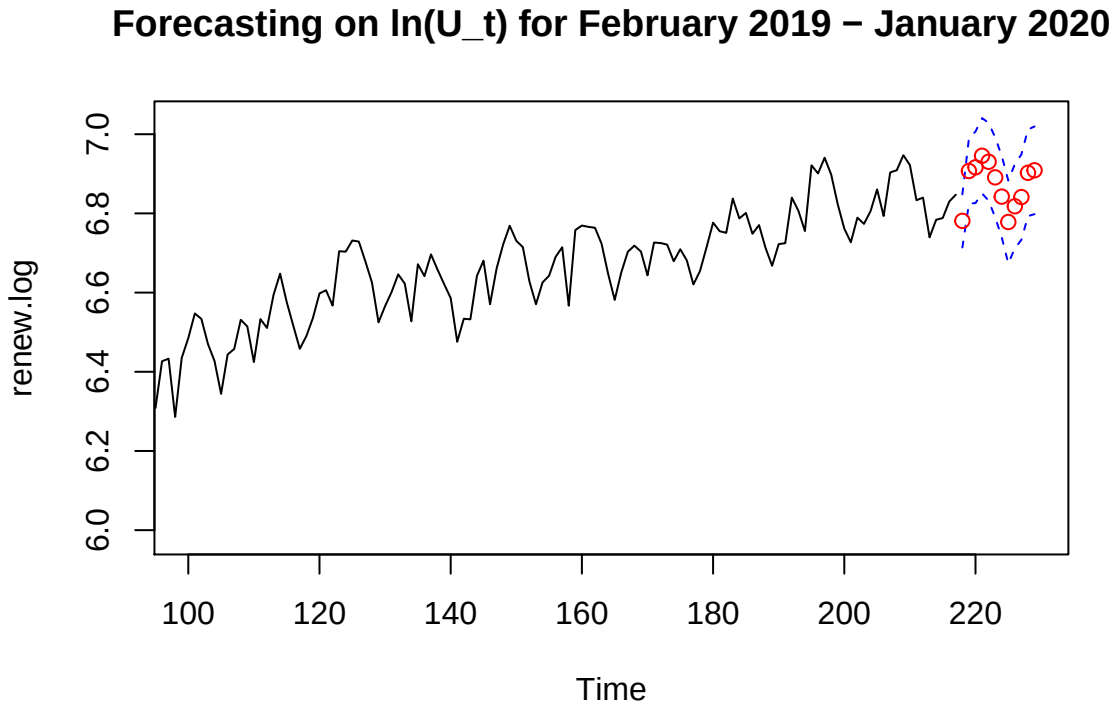


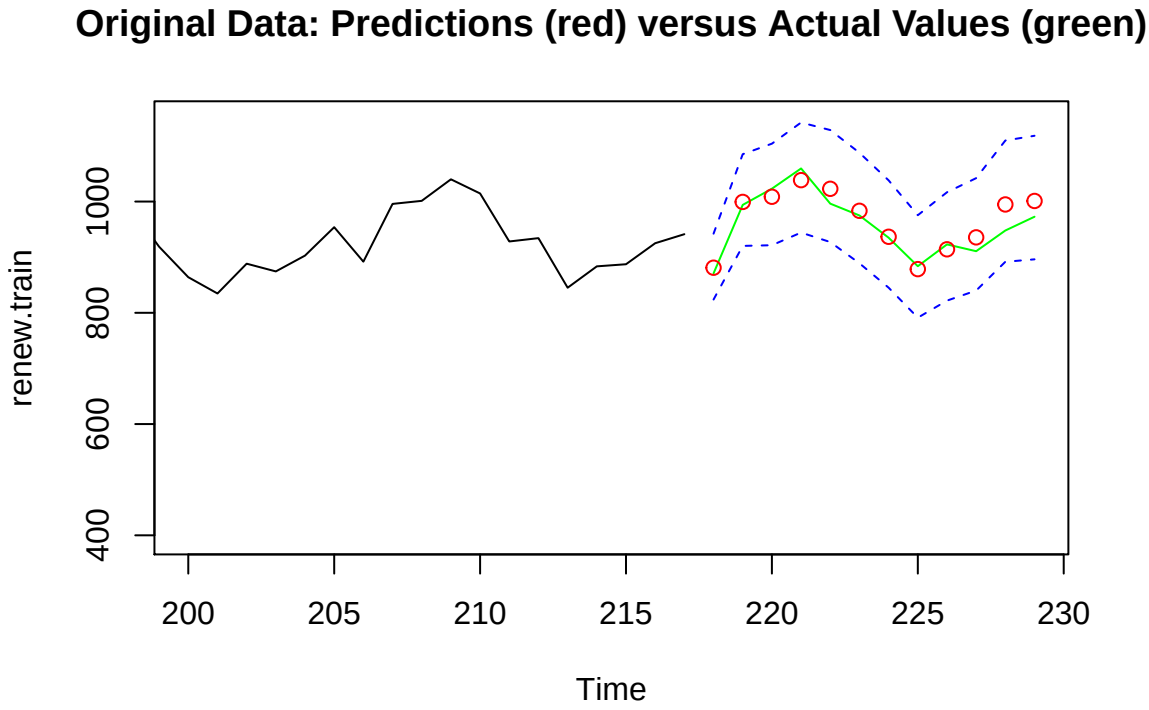





```
## [1] 0.468141
```

Forecasting





Comparing the model's predictions versus the actual values in the test set for renewable energy consumption from February 2019 to January 2020, I can see that they are very similar. Model 1 is able to capture the trend in the original data well.

```
##    original_pred renew.test
## 1       881.0520    870.879
## 2       999.3257    993.965
## 3      1008.6566   1023.090
## 4      1038.4907   1059.511
## 5      1022.9179    996.219
## 6       983.3600    975.007
## 7       936.6895    934.983
## 8       878.5303    883.759
## 9       914.1385    922.997
## 10      935.6220    910.772
## 11      994.7712    947.972
## 12     1001.0733    972.738
```

For fun, I forecasted the renewable energy consumption for February 2020 to January 2021. I can tell that the data is on an upward trend because the point forecasts are generally greater than the actual values for February 2019 to January 2020.

```
##    lower_CI point_forecast upper_CI
## 1  838.6715       938.3872 1049.959
## 2  934.9177      1049.0881 1177.201
## 3  937.4540      1054.3678 1185.862
## 4  967.9866      1091.0863 1229.841
## 5  938.5550      1059.8460 1196.812
## 6  910.3637      1029.8997 1165.131
## 7  859.4337       973.9646 1103.758
## 8  803.3706       911.8898 1035.068
## 9  841.5780       956.8849 1087.990
## 10 849.2373       967.0627 1101.235
## 11 911.5571      1039.6828 1185.817
## 12 910.9323      1040.5912 1188.705
```

Conclusion

My goal for this project was to find a satisfactory model that can be used to forecast monthly renewable energy consumption, and I believe that Model 1 fits the criteria because it is stationary, invertible, and passes all diagnostic tests. To reiterate, Model 1 is a SARIMA(3,1,3)x$(0,1,1)_{12}$ model with 6 non-zero coefficients. It can be written in the formula, $(1 + 0.9648B - 0.6254B^3)(1 - B)(1 - B^{12})ln(U_t) = (1 + 0.6779B - 0.437B^2 - 0.8923B^3)(1 - 0.8243B^{12})Z_t$ with $\hat{\sigma}_Z^2 = 0.001125$, where $ln(U_t)$ is the log transformed training data. Additionally, Model 1 supports the conclusion that the United States will continue to use various types of renewable energy at an increasing rate.

I want to thank Professor Raya Feldman from the Department of Statistics and Applied Probability at the University of California, Santa Barbara for teaching me time series analysis techniques and procedures. Your help and advice for this project was much appreciated!

References

U.S. Energy Information Administration. *Renewable Energy Production and Consumption by Source, 2001-2020.* U.S. Energy Information Administration, 2020. Web. 15 May 2020. https://www.eia.gov/totalenergy/data/browser/index.php?tbl=T10.01#/?f=M&start=200101&end=202001&charted=6-16

Appendix

```
library(readr)
#read in data
renew.csv <- read_csv("C:/Users/angel/OneDrive/Documents/renew_en.csv")
renew <- ts(renew.csv[,3], start = c(2001,1), frequency = 12)

#plotting time series of original data
plot.ts(renew, main = "Plot of Original Renewable Energy Consumption Data", ylab="Trillion Btu")

#plotting time series with linear trend and mean
l <- as.numeric(1:length(renew))
r <- renew[c(1:229)]
trend <- lm(renew ~ l)
plot(x=l, y=r, type="l", ylab = "Trillion Btu", xlab="Time", main = "Plot of Original Data with linear
curve(predict(trend, newdata=data.frame(l=x)), add=TRUE, col="red")
abline(h=mean(renew), col="blue")
```

```
#splitting into training and test sets
renew.train <- renew[c(1:217)]
renew.test <- renew[c(218:229)]
par(mfrow=c(2,2))

#plotting time series, histogram, and acf of U_t
plot.ts(renew.train, main = "Plot of Truncated Original Data, U_t")
hist(renew.train, col="blue", main="Histogram of U_t")
acf(renew.train, lag.max=50, main="acf of U_t")
require(MASS)

#plotting confidence interval for lambda
bcTransform <- boxcox(renew.train~as.numeric(1:length(renew.train)))
```

```
#Box-Cox says lambda should be 0.2222
lambda <- bcTransform$x[which(bcTransform$y==max(bcTransform$y))]
lambda
```

```
par(mfrow=c(2,2))
#plotting bc(U_t) and its histogram
renew.bc <- (1/lambda)*(renew.train^lambda-1)
plot.ts(renew.bc, main = "Plot of bc(U_t)")
hist(renew.bc, col="blue", main = "Histogram of bc(U_t)")

#plotting ln(U_t) and its histogram
renew.log <- log(renew.train)
plot.ts(renew.log, main = "Plot of ln(U_t)")
hist(renew.log, col="blue", main = "Histogram of ln(U_t)")
```

```
par(mfrow=c(2,1))
#plotting time series of ln(U_t) diff at lag 12
renew.log.12 <- diff(renew.log, lag=12)
plot.ts(renew.log.12, main = "ln(U_t) differenced at lag 12")
fit1 <-lm(renew.log.12 ~ as.numeric(1:length(renew.log.12)))
abline(fit1, col="red")
abline(h=mean(renew.log.12), col="blue")
```

```r
#plotting time series of ln(U_t) diff at lags 12 and 1
renew.log.1.12 <- diff(renew.log.12, lag=1)
plot.ts(renew.log.1.12, main = "ln(U_t) differenced at lags 12 & 1")
fit2 <-lm(renew.log.1.12 ~ as.numeric(1:length(renew.log.1.12)))
abline(fit2, col="red")
abline(h=mean(renew.log.1.12), col="blue")

renew.log.1.1.12 <- diff(renew.log.1.12, lag=1)

#comparing variances
dataset <- c("ln(U_t)", "ln(U_t) differenced at lag 12", "ln(U_t) differenced at lag 12 and lag 1",
             "ln(U_t) differenced at lag 12 once and lag 1 twice")
mean <- c(mean(renew.log), mean(renew.log.12), mean(renew.log.1.12), mean(renew.log.1.1.12))
variance <- c(var(renew.log), var(renew.log.12),var(renew.log.1.12), var(renew.log.1.1.12))
data.frame(dataset, mean, variance)

#plotting histograms of ln(U_t) diff at lags 12 and 1
par(mfrow=c(1,2))
hist(renew.log.1.12, col="blue", main="Histogram: ln(U_t) diff at 12&1")
hist(renew.log.1.12, density = 10, breaks = 10, col="blue", prob=TRUE,
     main="Histogram with normal curve")
m <- mean(renew.log.1.12)
std <- sqrt(var(renew.log.1.12))
curve(dnorm(x,m,std),add=TRUE)
```

```r
par(mfrow=c(1,2))

#plotting acf and pacf of ln(U_t) diff at lags 12 and 1
acf(renew.log.1.12, main = "ACF of ln(U_t) diff at lags 12&1",lag.max=45)
pacf(renew.log.1.12, main= "PACF of ln(U_t) diff at lags 12&1",lag.max=45)
#acf sticks up lags 1, 12, and 35
#pacf sticks up at lags 1,2, 12, 24, 36
```

```r
#starting with a pure SMA model
prelim_model1 <- arima(renew.log, order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12), method="ML"
prelim_model1

par(mfrow=c(1,2))
res <- residuals(prelim_model1)
acf(res, lag.max=45, main="acf of preliminary model 1")
pacf(res,lag.max=45, main="pacf of preliminary model 1")
#acf sticks up at lags 8 and 35
#pacf at lag 8
```

```r
#I changed p to 8 to hopefully improve the model
prelim_model2 <- arima(renew.log, order=c(8,1,1), seasonal=list(order=c(0,1,1), period=12), method="ML"
prelim_model2

par(mfrow=c(1,2))
res <- residuals(prelim_model2)
acf(res, lag.max=45, main="acf of preliminary model 2")
pacf(res,lag.max=45, main="pacf of preliminary model 2")
#both acf and pacf stick up at lag 35
```

```r
#I changed Q to 4
prelim_model3 <- arima(renew.log, order=c(8,1,1), seasonal=list(order=c(0,1,4), period=12), method="ML")
prelim_model3

par(mfrow=c(1,2))
res <- residuals(prelim_model3)
acf(res, lag.max=45, main="acf of preliminary model 3")
pacf(res,lag.max=45, main="pacf of preliminary model 3")
#both acf and pacf stick up at lag 35
```

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

#fit Model 1 by setting insignificant coefficients to zero
model1 <- arima(renew.log, order=c(3,1,3), seasonal=list(order=c(0,1,1), period=12),
                fixed=c(NA,0,NA,NA,NA,NA,NA),method="ML")
model1

library(qpcR)
#comparing AICc where the only difference is setting the second AR coefficient to zero
AICc(arima(renew.log, order=c(3,1,3), seasonal=list(order=c(0,1,1), period=12), method="ML"))
AICc(model1)
```

```r
#fit Model 2 by setting insignificant coefficients to zero
model2 <- arima(renew.log, order=c(35,1,1), seasonal=list(order=c(0,1,1), period=12),
                fixed=c(NA,0,0,0,0,0,0,NA,0,0,0,0,0,0,0,0,0,0,
                        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NA,NA,NA),
                method="ML")
model2
```

```r
#what if I change the model into pure SMA and set all insignificant coefficients to 0?
#fitting Model 3
model3 <- arima(renew.log, order=c(0,1,35), seasonal=list(order=c(0,1,1), period=12),
                fixed=c(NA,NA,0,0,0,0,0,NA,0,0,0,0,0,0,0,0,0,0,
                        0,0,0,NA,0,0,0,NA,0,0,0,0,0,0,0,0,NA,0,NA,NA),
                method="ML")
model3
```

```r
#summary of models so far
model <- c("Model 1", "Model 2", "Model 3")
type <- c("SARIMA(3,1,3)x(0,1,1)_12", "SARIMA(35,1,1)x(0,1,1)_12", "SARIMA(0,1,35)x(0,1,1)_12")
nonzero_coefficients <- c(6,5,8)
AICc <- c(AICc(model1), AICc(model2),AICc(model3))
data.frame(model, type, nonzero_coefficients, AICc)
```

```r
library(UnitCircle)
par(mfrow=c(1,2))
#checking roots of Model 1
#checking roots of non-seasonal AR part
uc.check(pol_ = c(1,0.9648, 0 ,-0.6254), plot_output = TRUE, print_output = FALSE)
#checking roots of non-seasonal MA part
uc.check(pol_ = c(1,0.6779, -0.437 ,-0.8923), plot_output = TRUE, print_output = FALSE)
```

```r
#checking roots of Model 2
#checking roots of non-seasonal AR part
uc.check(pol_ = c(1,-0.3826,0,0,0,0,0,0,0.1582,0,0,0,0,0,0,0,0,0,0,
                  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-0.2028),
```

```
                  plot_output = TRUE, print_output = FALSE)
```

```
#checking roots of Model 3
#checking roots of non-seasonal MA part
uc.check(pol_ = c(1,-0.6381,-0.3184,0,0,0,0,0,-0.2928,0,0,0,0,0,0,0,0,0,
                  0,0,0,-0.2465,0,0,0,0.2812,0,0,0,0,0,0,0,-0.2380,0,0.3701),
                  plot_output = TRUE, print_output = FALSE)
```

```
#checking diagnostics for Model 1
#plotting acf and pacf of Model 1's residuals
res_model1 <- residuals(model1)
par(mfrow=c(1,2))
acf(res_model1, lag.max=45, main="acf of model 1's residuals")
pacf(res_model1,lag.max=45, main="pacf of model 1's residuals")
```

```
par(mfrow=c(2,2))

#plotting histogram and time series of Model 1's residuals
hist(res_model1, density=10, breaks=10, col="blue", prob=TRUE)
m <- mean(res_model1)
std <- sqrt(var(res_model1))
curve(dnorm(x,m,std), add=TRUE)
plot.ts(res_model1)
fit3 <- lm(res_model1~as.numeric(1:length(res_model1)))
abline(fit3, col="red")
abline(h=mean(res_model1), col="blue")

#qq-plot
qqnorm(res_model1)
qqline(res_model1, col="blue")

#running Shapiro-Wilk, Box-Pierce, Ljung-Box, and McLeod-Li tests
shapiro.test(res_model1)
Box.test(res_model1, lag=15, type=c("Box-Pierce"), fitdf=6)
Box.test(res_model1, lag=15, type=c("Ljung-Box"), fitdf=6)
Box.test((res_model1)^2, lag=15, type=c("Ljung-Box"), fitdf=0)

ar(res_model1, aic=TRUE, order.max=NULL, method=c("yule-walker"))
```

```
#checking diagnostics for Model 2
#plotting acf and pacf of Model 2's residuals
res_model2 <- residuals(model2)
par(mfrow=c(1,2))
acf(res_model2, lag.max=45, main="acf of model 2's residuals")
pacf(res_model2,lag.max=45, main="pacf of model 2's residuals")
```

```
par(mfrow=c(2,2))

#plotting histogram and time series of Model 2's residuals
hist(res_model2, density=10, breaks=10, col="blue", prob=TRUE)
m <- mean(res_model2)
std <- sqrt(var(res_model2))
curve(dnorm(x,m,std), add=TRUE)
plot.ts(res_model2)
fit4 <- lm(res_model2~as.numeric(1:length(res_model2)))
```

```r
abline(fit4, col="red")
abline(h=mean(res_model2), col="blue")

#qq-plot
qqnorm(res_model2)
qqline(res_model2, col="blue")

#running Shapiro-Wilk, Box-Pierce, Ljung-Box, and McLeod-Li tests
shapiro.test(res_model2)
Box.test(res_model2, lag=15, type=c("Box-Pierce"), fitdf=5)
Box.test(res_model2, lag=15, type=c("Ljung-Box"), fitdf=5)
Box.test((res_model2)^2, lag=15, type=c("Ljung-Box"), fitdf=0)

ar(res_model2, aic=TRUE, order.max=NULL, method=c("yule-walker"))
```

```r
library(TSA)
library(GeneCycle)
par(mfrow=c(1,3))

#periodograms of ln(U_t) and Model 1's residuals
TSA::periodogram(renew.log, ylab="Periodogram of ln(U_t)")
abline(h=0)
axis(1, at=0.08)
TSA::periodogram(res_model1, ylab="Periodogram of Model 1's residuals")
abline(h=0)

#Kolmogorov-Smirnov test
cpgram(res_model1)
```

```r
#Fisher's test
fisher.g.test(res_model1)
```

```r
library(forecast)
final_fit <- model1

#forecasting for February 2019 - January 2020 with transformed data
pred <- predict(final_fit, n.ahead=12)
U <- pred$pred + 2*pred$se
L <- pred$pred - 2*pred$se
ts.plot(renew.log, xlim=c(100,length(renew.log)+12), ylim=c(min(renew.log),max(U)),
        main = "Forecasting on ln(U_t) for February 2019 - January 2020")
lines(U, col="blue", lty= "dashed")
lines(L, col="blue", lty="dashed")
points((length(renew.log)+1):(length(renew.log)+12),pred$pred, col="red")
```

```r
#forecasting for February 2019 - January 2020 with original data
original_pred <- exp(pred$pred)
U <- exp(U)
L <- exp(L)
ts.plot(renew.train, xlim=c(200,length(renew.train)+12), ylim=c(min(renew.train),max(U)+8),
        main = "Original Data: Predictions (red) versus Actual Values (green)")
lines(U, col="blue", lty= "dashed")
lines(L, col="blue", lty="dashed")
lines((length(renew.train)+1):(length(renew.train)+12), renew.test, col="green")
```

```r
points((length(renew.train)+1):(length(renew.train)+12), original_pred, col="red")
```

```r
#forecasting for February 2020 - January 2021 with original data
forecasting <- forecast(final_fit, h=24)
point_forecast <- exp(forecasting$mean[13:24])
lower_CI <- exp(forecasting$lower[13:24,2])
upper_CI <- exp(forecasting$upper[13:24,2])
data.frame(lower_CI, point_forecast, upper_CI)
```