Time Series: Yearly Electricity Consumption (US), 1920 - 1970

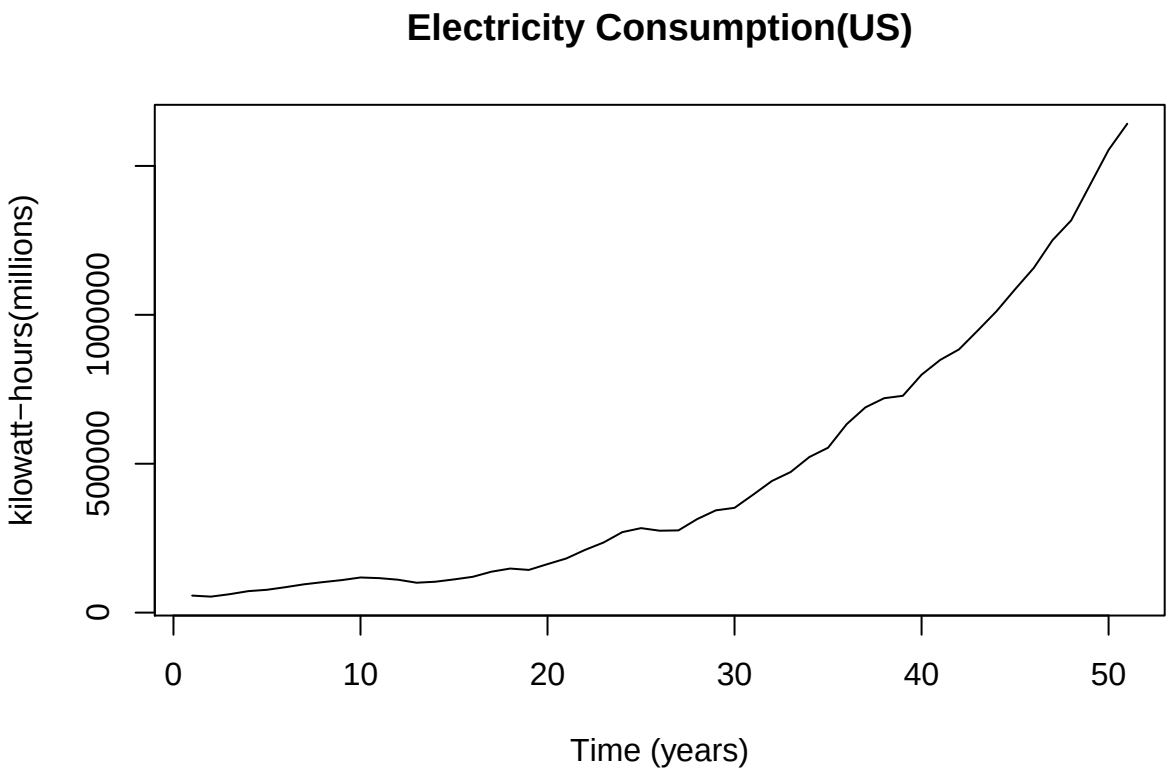
Phillip Kim

Project Date: November 2017 - December 2017

Executive Summary
Currently, the United States stand in second place in the ranking of countries that have the highest yearly electric energy consumption (in units of kilowatts multiplied by hour per year), consuming a total of 3,913,000,000,000 kW · h/yr as of 2014. In this time series project, we will be able to witness whether the data for electric energy consumption for 2014 is within the confidence interval of the forecasted data points. Some time series techniques that will be used to analyze this are the following: box-cox transformation, differencing, model-fitting, diagnostic checking, forecasting, and spectral analysis.

Original Data
Here is the graph of the time series data:

```
setwd("/Users/pureInfinitas/Documents/School/Fall 2017/PSTAT 174/untitled folder")
elec = read.csv("total-electricity-consumption-us.csv", header = TRUE)
elect = ts(elec[, 2])
elec2 = read.csv("estimate_total_elec_consume.csv", header = FALSE)
elect2 = ts(elec2[, 5])
ts.plot(elect, ylab = "kilowatt-hours(millions)", xlab = "Time (years)",
    main = "Electricity Consumption(US)")
```



Using the var( ) function in R, we find that the variance of this time series is: 198484636051. From figure 1, we can see that there is a clear positive trend; however, there is no seasonality. The trend doesn't appear to be linear and looks more likely to be an exponential trend. Because of this exponential trend, the variance

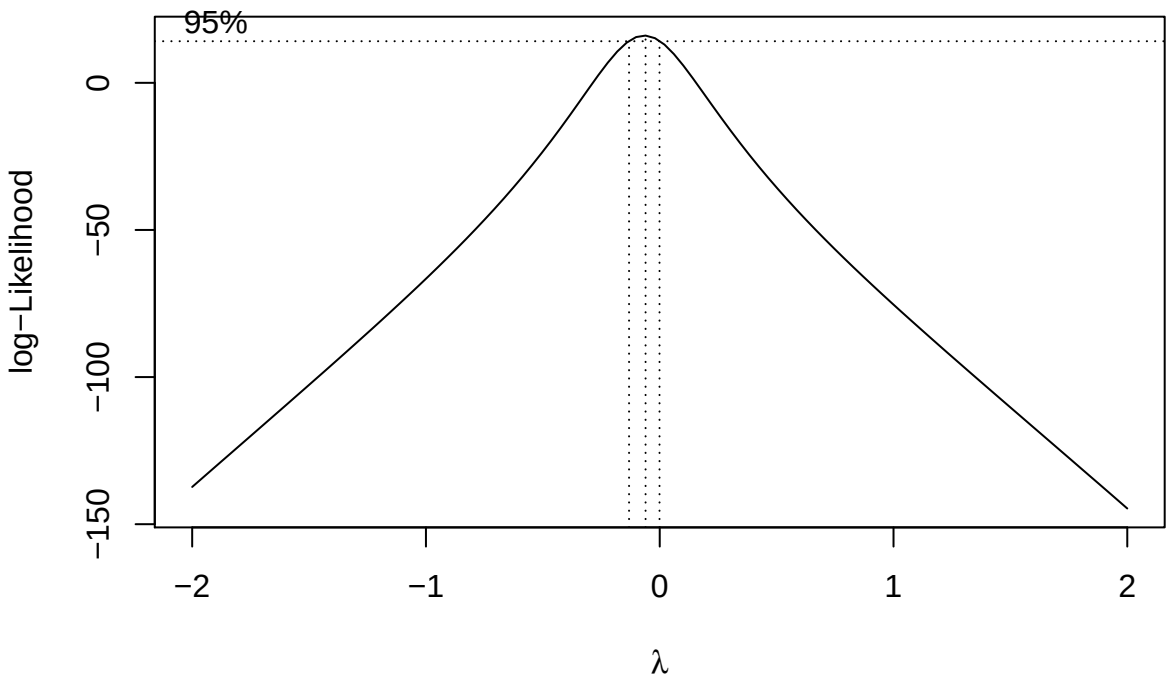
of the time series seems to increase as time passes. Therefore, it would be necessary to use the Box-Cox transformation to stabilize the variance of the time series. Differencing at lag 1 would also improve the stationarity of the time series.

Data Transformation

Box-Cox Transformation

We now proceed to use the Box-Cox Transformation.

```
require(MASS)
bcElec <- boxcox(elect ~ as.numeric(1:length(elect)))
```




```
optimalLambda <- bcElec$x[which(bcElec$y == max(bcElec$y))]
optimalLambda
```

```
## [1] -0.06060606
```

From the above figure, we observe that although the optimal to use for the box-cox transformation is -0.06060606, because this value is less than 0 and 0 is within the 95% confidence interval of the log-likelihood plot, we use the following transformation: f (Ut) = ln Ut, where Ut represents our original data. Therefore, we obtain the following time series:

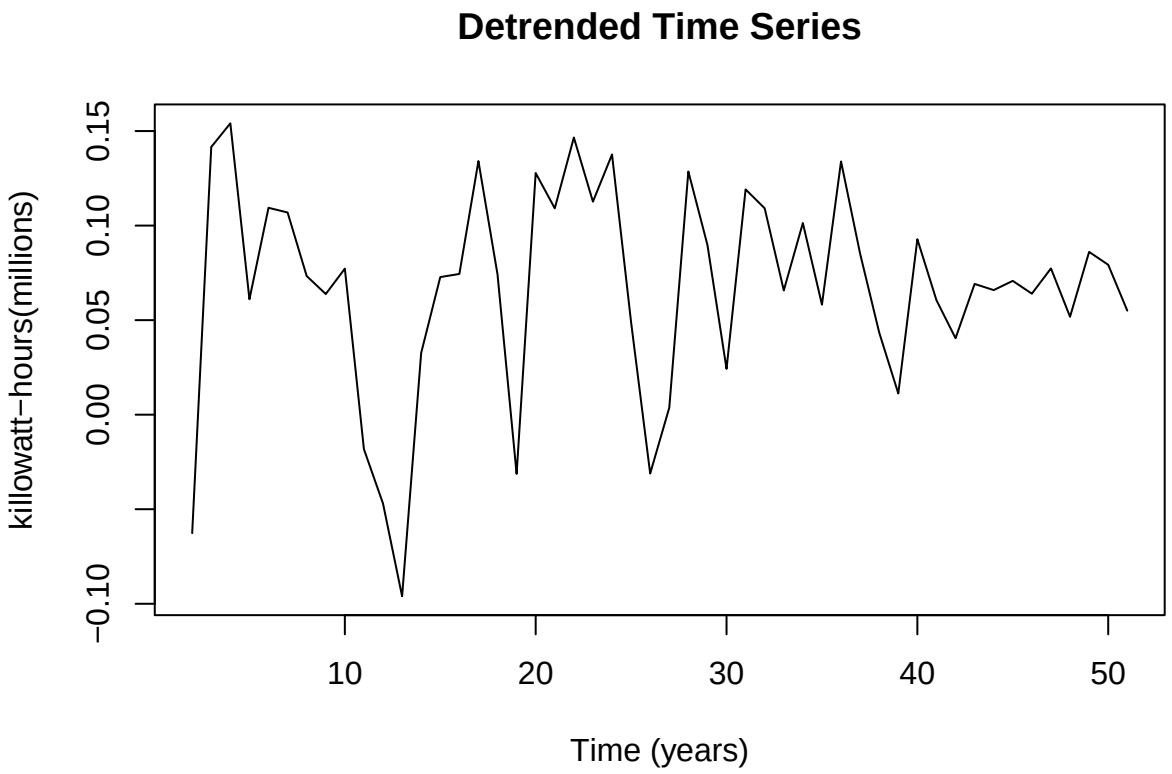
```
elect_bc = log(elect)
ts.plot(elect_bc, main = "Box-Cox transformed data", ylab = "kilowatt-hours(millions)",
    xlab = "Time (years)")
```


```
var(elect_bc)
```

```
## [1] 1.045185
```

Using the var( ) function in R, we find that the variance of this time series is: 1.045185, therefore stabilizing the variance of the time series. We also observe that the trend of the time series data has become more linear than exponential due to the box-cox transformation. Because we would want to remove a linear trend, we proceed by differencing the time series once at lag 1.

```
elect_diff1 = diff(elect_bc, 1)
plot(elect_diff1, main = "Detrended Time Series", ylab = "killowatt-hours(millions)",
    xlab = "Time (years)")
```

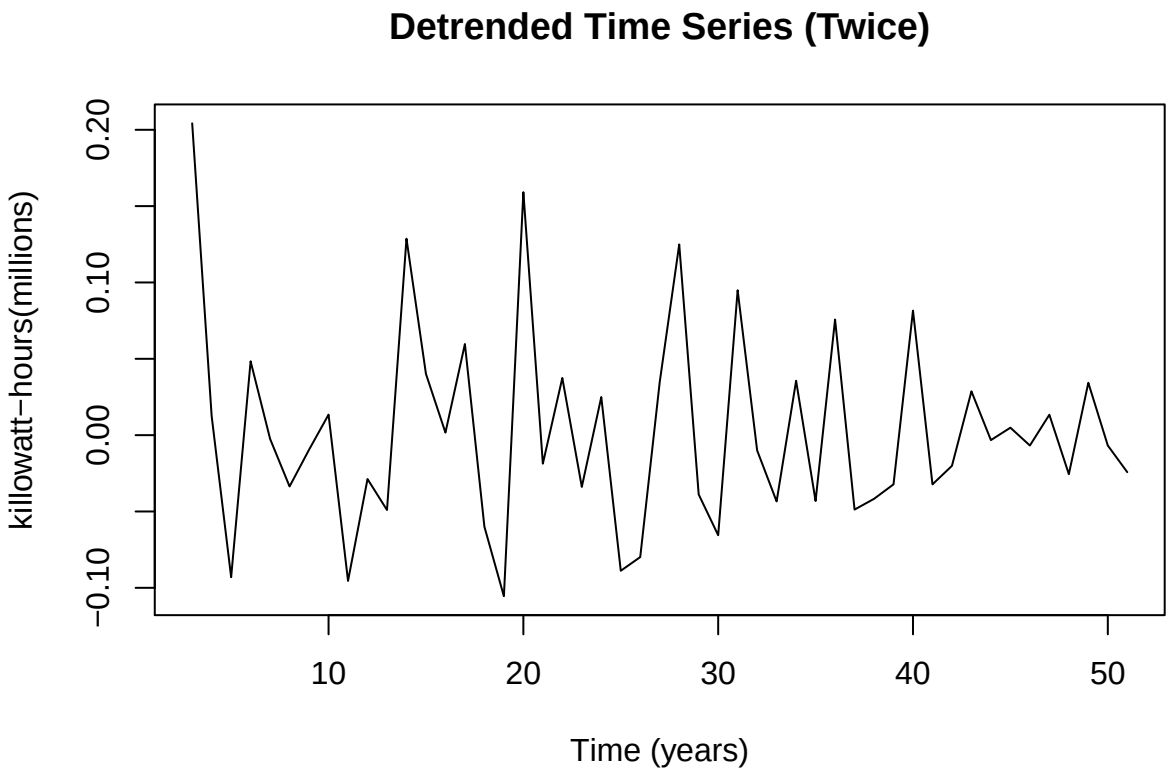



```
var(elect_diff1)
```

```
## [1] 0.003092171
```

From the above, we observe that there no longer exists a trend and there is barely any presence of seasonality. The variance of this time series plot is: 0.003092171. For the purpose of experimenting, I decide to difference the time series once again at lag 1 to be sure that I have the optimal transformed data:

```
elect_diff1_2 = diff(elect_diff1, 1)
plot(elect_diff1_2, main = "Detrended Time Series (Twice)", ylab = "killowatt-hours(millions)",
    xlab = "Time (years)")
```




```
var(elect_diff1_2)
```

```
## [1] 0.004274764
```

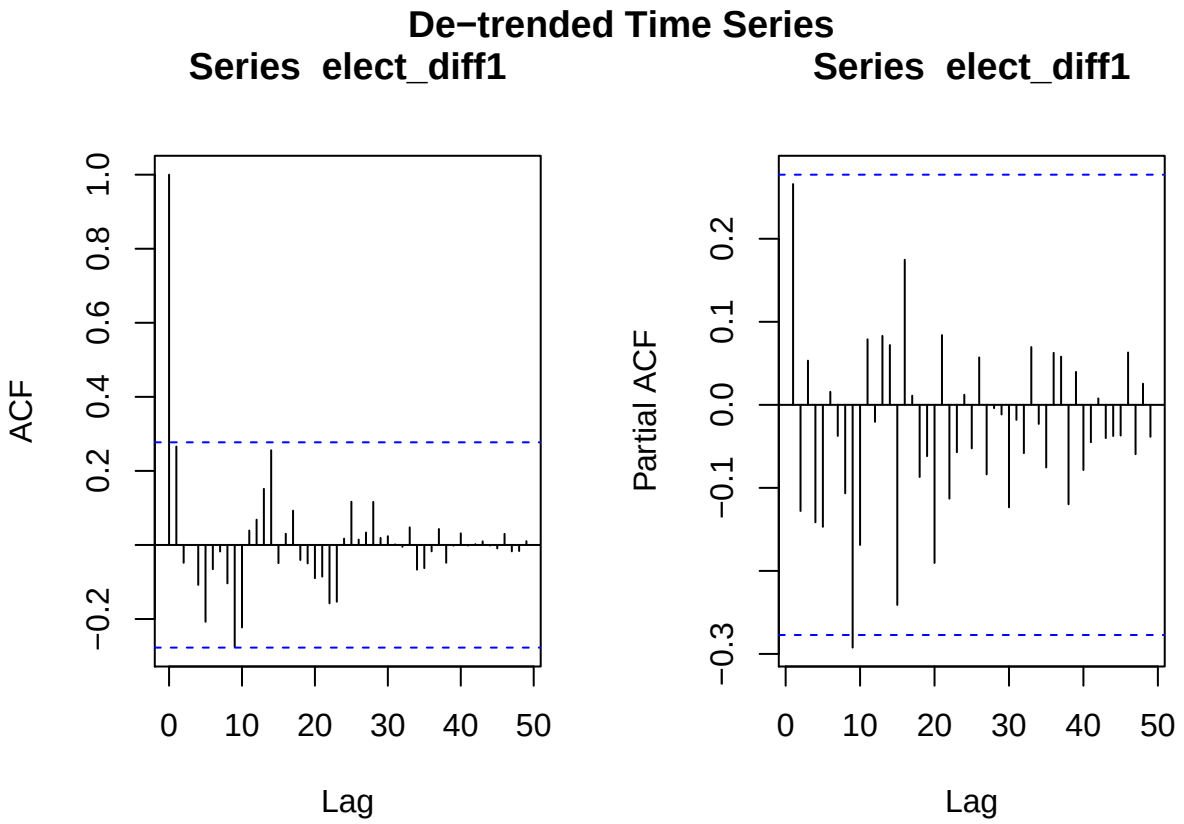
From the above figure, we observe that there appears to be a stronger presence of seasonality than before; however, we also take into account the increase in variance of this time series data, in which after differencing at lag 1 twice, we obtain a variance of: 0.004274764. Therefore, we conclude that differencing the time series at lag 1 only once provides us with the optimal transformed data.

Model Building

ACF and PACF speculation

We now plot the ACF and PACF graphs of the transformed data:

```
op = par(mfrow = c(1, 2))
acf(elect_diff1, lag.max = 100)
pacf(elect_diff1, lag.max = 100)
title("De-trended Time Series", line = -1, outer = TRUE)
```



From the above figures, I speculate that the model is: ARIMA(9, 0, 9).

Comparisons The next step is to compare various ARIMA(p, 0, q) models for p : 0 p 9 and q : 0 q 9 using the AICc. We therefore work towards building a total of 100 models and comparing 100 AICc values. Using a for loop in R and the AICc( ) function, we obtain the following AICc matrix:

```
aiccs <- matrix(NA, nr = 10, nc = 10)
dimnames(aiccs) = list(p = 0:9, q = 0:9)
for (p in 0:9) {
    for (q in 0:9) {
        aiccs[p + 1, q + 1] = AICc(arima(elect_diff1, order = c(p,
            0, q), method = "ML", optim.control = list(maxit = 500)))
    }
}

aiccs
```

```
##    q
## p           0         1         2         3         4         5         6
##   0 -143.9770 -147.4544 -145.3592 -143.3042 -140.8770 -146.4630 -146.0994
##   1 -145.8734 -145.3197 -146.2121 -143.7773 -141.2111 -145.6874 -144.8784
##   2 -145.1625 -143.1093 -143.7830 -141.9222 -139.2427 -143.2413 -141.9282
##   3 -142.9671 -140.6524 -141.2282 -142.3670 -140.6401 -140.3252 -138.8421
##   4 -141.5628 -142.0008 -143.1376 -138.2689 -138.6398 -137.2396 -136.2513
##   5 -140.4529 -137.9725 -136.9986 -138.6544 -132.3004 -137.9050 -134.4478
##   6 -137.8677 -135.4229 -134.5975 -135.5479 -133.0347 -134.4342 -130.8856
##   7 -135.2187 -134.0401 -131.4642 -132.2564 -131.4746 -130.9444 -127.0869
##   8 -132.8746 -131.8312 -127.5928 -129.7142 -128.2405 -127.3392 -123.7491
##   9 -134.4416 -131.3693 -133.4652 -129.9161 -126.3648 -125.4426 -120.2800
##    q
```

```
## p           7         8         9
##   0 -143.7517 -141.4761 -138.8958
##   1 -141.9215 -138.8389 -135.2742
##   2 -138.9793 -135.6849 -134.9590
##   3 -135.6839 -135.0392 -131.2088
##   4 -135.1972 -131.5696 -128.6390
##   5 -130.8839 -129.7329 -124.2305
##   6 -127.0697 -125.6991 -119.0025
##   7 -122.9813 -121.4930 -114.7648
##   8 -122.1398 -117.7705 -109.9563
##   9 -117.6951 -115.3249 -112.4427
```

It should be noted that p corresponds to the indices of the rows of the matrix while q corresponds to the indices of the columns of the matrix. Using the which( ) function, we determine the entry of the matrix which contains the minimal AICc, which would be in the first row and second column of the aiccs matrix. Because we would want to use AICc as our main method of selecting a model, we therefore adopt the ARIMA(0, 0, 1) model, which differs from our speculation of the ARIMA(9, 0, 9) model.

```
which(aiccs == min(aiccs), arr.ind = TRUE)
```

```
##   p q
## 0 1 2
```

```
arima(elect_diff1, order = c(0, 0, 1), method = "ML")
```

```
##
## Call:
## arima(x = elect_diff1, order = c(0, 0, 1), method = "ML")
##
## Coefficients:
##          ma1  intercept
##       0.4016     0.0659
## s.e.  0.1435     0.0102
##
## sigma^2 estimated as 0.002697:  log likelihood = 76.85,  aic = -147.71
```

Using the arima( ) function, we obtain the following ARIMA(0, 0, 1) model: $X_t = Z_t + 0.4016Z_{t1} + 0.0659$.

Diagnostics Check

Now we perform diagnostic checking on the fitted model residuals:

```
fit = arima(elect_diff1, order = c(0, 0, 1), method = "ML")
Box.test(residuals(fit), type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  residuals(fit)
## X-squared = 0.29175, df = 1, p-value = 0.5891
```

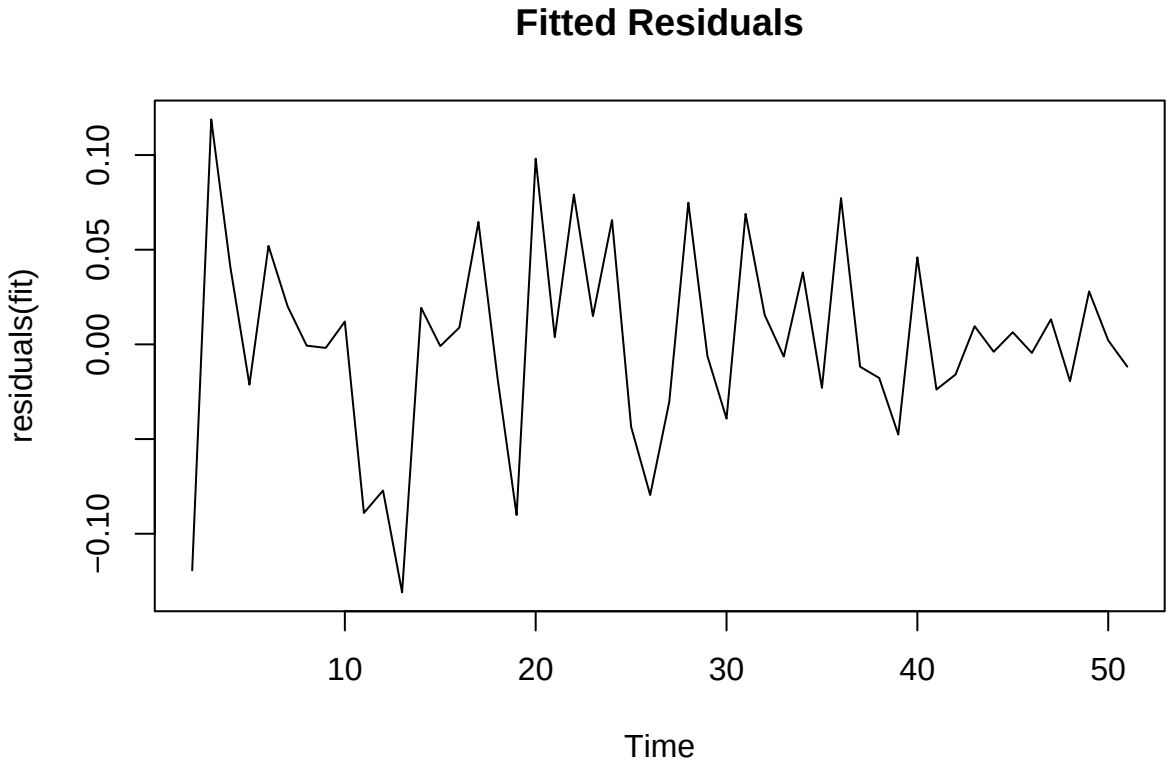
```
shapiro.test(residuals(fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit)
## W = 0.97221, p-value = 0.284
```

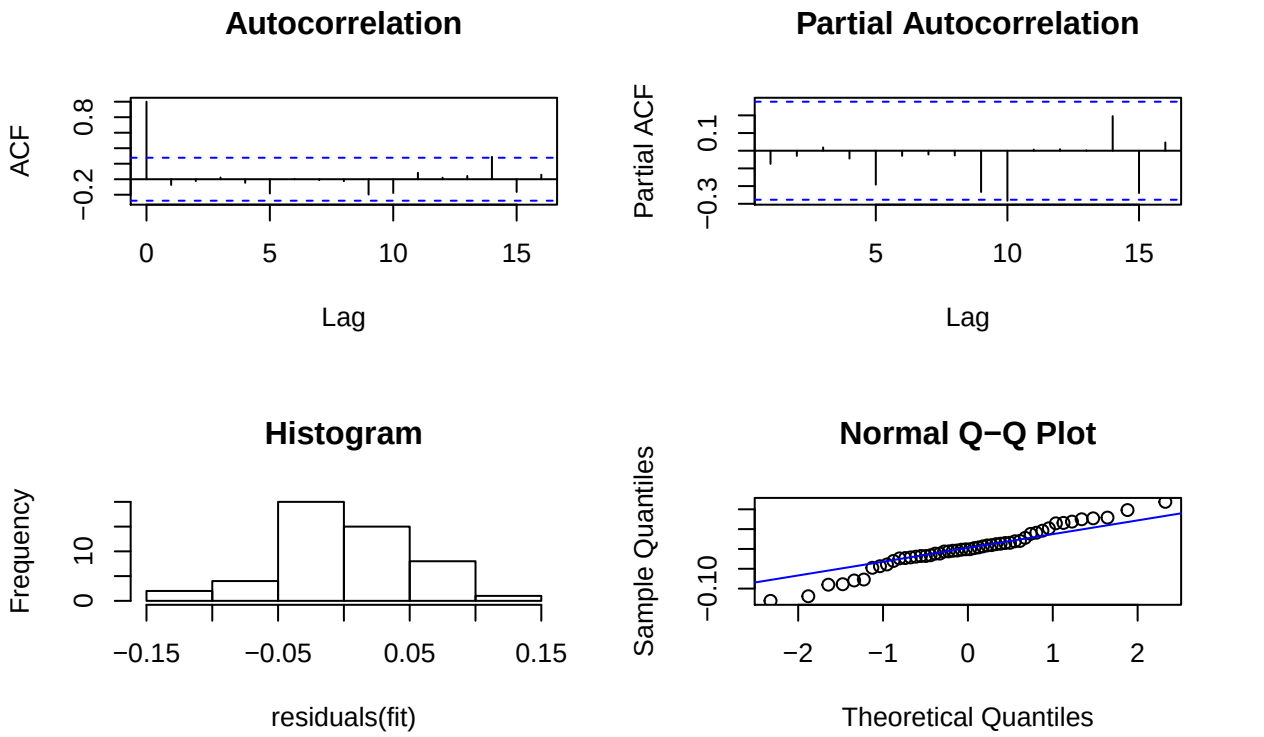
From the Box-Ljung test, we find that the p-value is greater than 0.05, which means that the residuals are

independent on each other. We also see that from the Shapiro-Wilk normality test, the p-value is greater than 0.05, which means that the residuals are approximately IID Gaussian. Now, we plot the residuals:

```
ts.plot(residuals(fit), main = "Fitted Residuals")
```




```
par(mfrow = c(1, 2), oma = c(0, 0, 2, 0))
# Plot diagnostics of residuals
op <- par(mfrow = c(2, 2))
# acf
acf(residuals(fit), main = "Autocorrelation")
# pacf
pacf(residuals(fit), main = "Partial Autocorrelation")
# Histogram
hist(residuals(fit), main = "Histogram")
# q-q plot
qqnorm(residuals(fit))
qqline(residuals(fit), col = "blue")
title("Fitted Residuals Diagnostics", outer = TRUE)
```




```
par(op)
```

Although we use AICc as our prime criterion for choosing the best model, I decide to use another model, the model we have currently contains only MA components, and without the presence of AR components, we wouldn't be able to take into account for the increase in total electricity consumption over time. Originally, I had decided to use a model which has the second minimum AICc, and found this model using the which( ) function. I had found that this model is the following: ARIMA(0, 0, 5). However, because this model doesn't take into account increase in total electricity consumption, it is necessary to attach an AR component to my ARIMA model. Therefore, in efforts to minimize AICc, we decide to use an ARIMA(1, 0, 2) model, which reduces the number of parameters used. Using the arima( ) function, we obtain the following ARIMA(1, 0, 2) model:

```
fit2 = arima(elect_diff1, order = c(1, 0, 2), method = "ML")
fit2
```

```
##
## Call:
## arima(x = elect_diff1, order = c(1, 0, 2), method = "ML")
##
## Coefficients:
##          ar1      ma1      ma2  intercept
##       0.8042  -0.5456  -0.4544     0.0674
## s.e.  0.1012   0.1484   0.1389     0.0028
##
## sigma^2 estimated as 0.002431:  log likelihood = 78.55,  aic = -147.1
```

Our model is the following: $X_t - 0.8042X_{t-1} = Z_t - 0.5456Z_{t-1} - 0.4544Z_{t-2} + 0.0674$

```
fit2 = arima(elect_diff1, order = c(1, 0, 2), method = "ML")
Box.test(residuals(fit2), type = "Ljung")
```

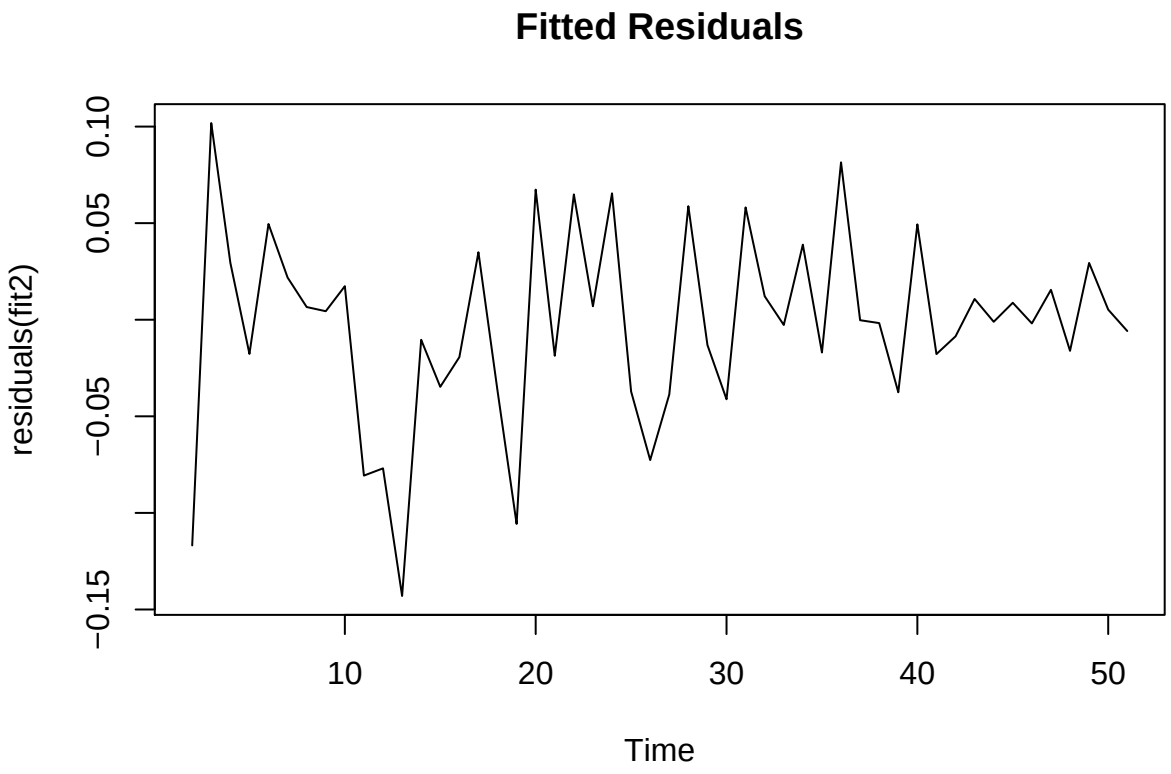
```
##
##  Box-Ljung test
##
## data:  residuals(fit2)
## X-squared = 0.058183, df = 1, p-value = 0.8094
```

```
shapiro.test(residuals(fit2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit2)
## W = 0.96636, p-value = 0.1639
```

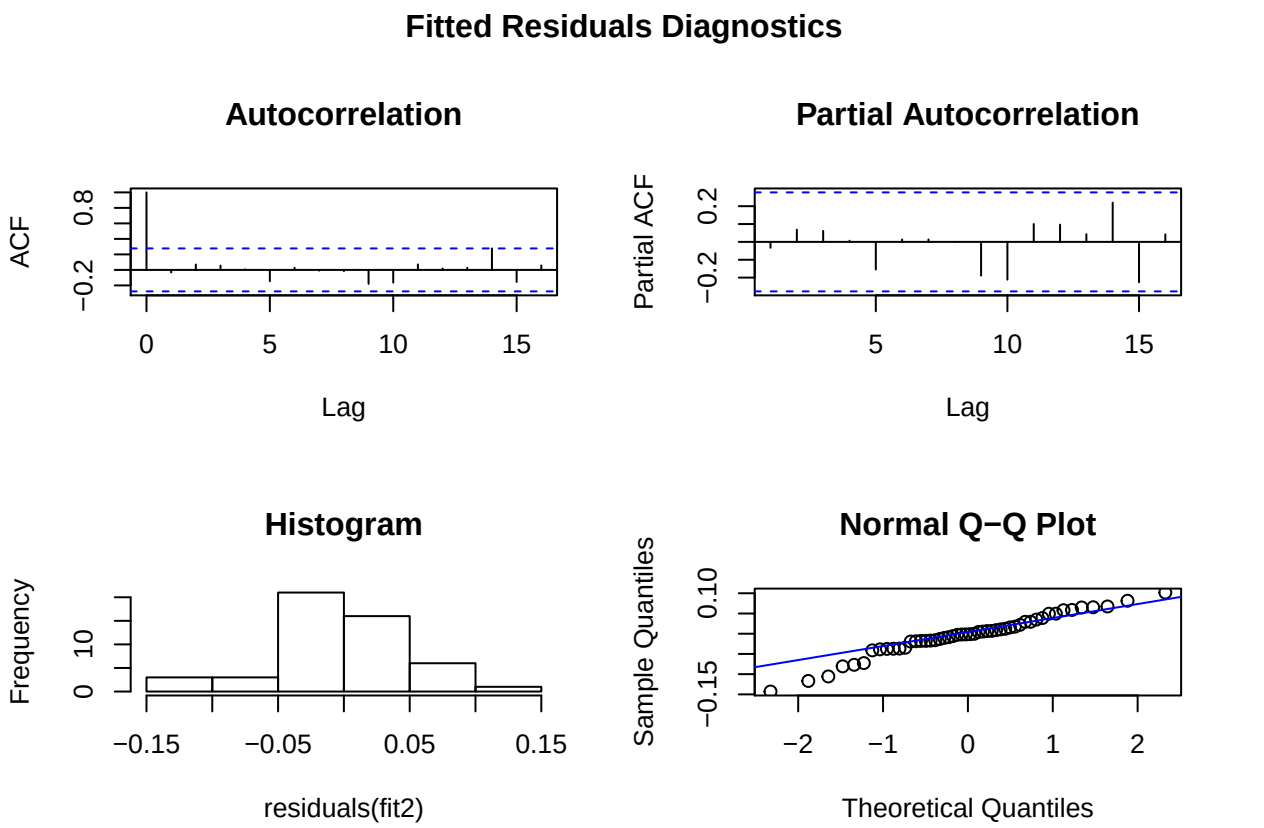
The p-value for the Box-Ljung test and Shapiro-Wilk normality test are both greater than 0.05, which implies that the residuals are independent from each other and are also IID Gaussian. The data is therefore normal and we adopt our ARIMA(1, 0, 2) model to be our optimal model.

```
ts.plot(residuals(fit2), main = "Fitted Residuals")
```




```
par(mfrow = c(1, 2), oma = c(0, 0, 2, 0))
# Plot diagnostics of residuals
op <- par(mfrow = c(2, 2))
# acf
acf(residuals(fit2), main = "Autocorrelation")
# pacf
pacf(residuals(fit2), main = "Partial Autocorrelation")
# Histogram
hist(residuals(fit2), main = "Histogram")
# q-q plot
qqnorm(residuals(fit2))
qqline(residuals(fit2), col = "blue")
```

```
title("Fitted Residuals Diagnostics", outer = TRUE)
```

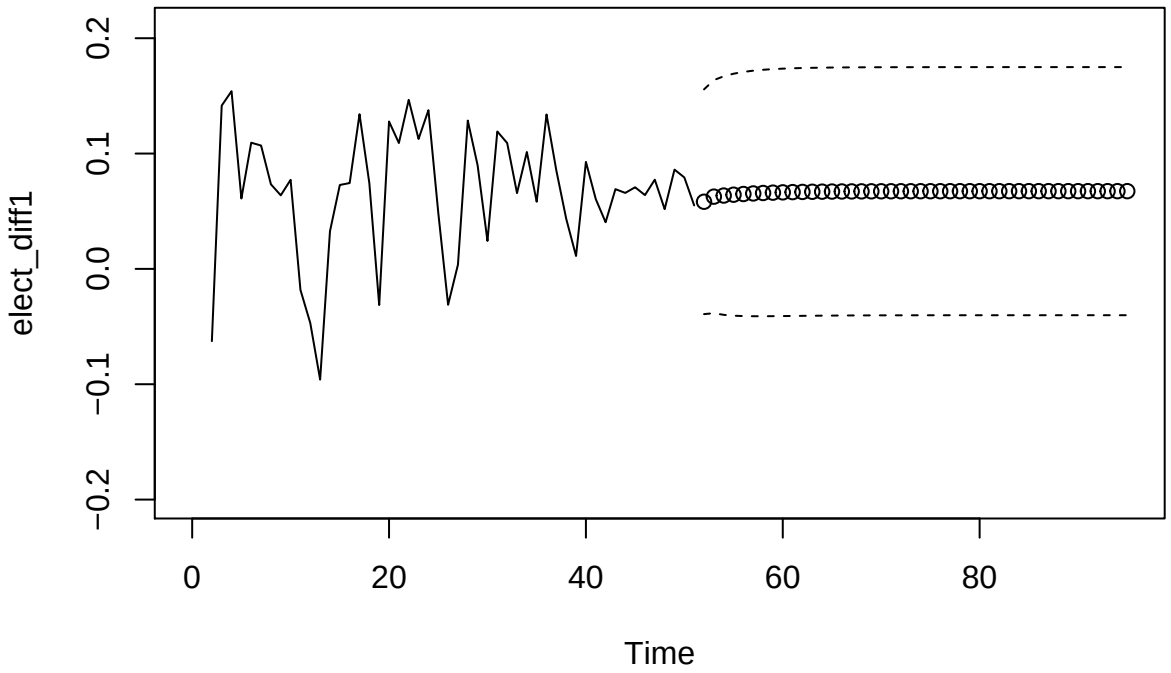



```
par(op)
```

Forecasting

Given our ARMA model, we use the predict( ) function to forecast the next 44 data points, which represent 44 years worth of total electricity consumption in the US.The below, predictions, however, are based on predictions of our transformed time series.

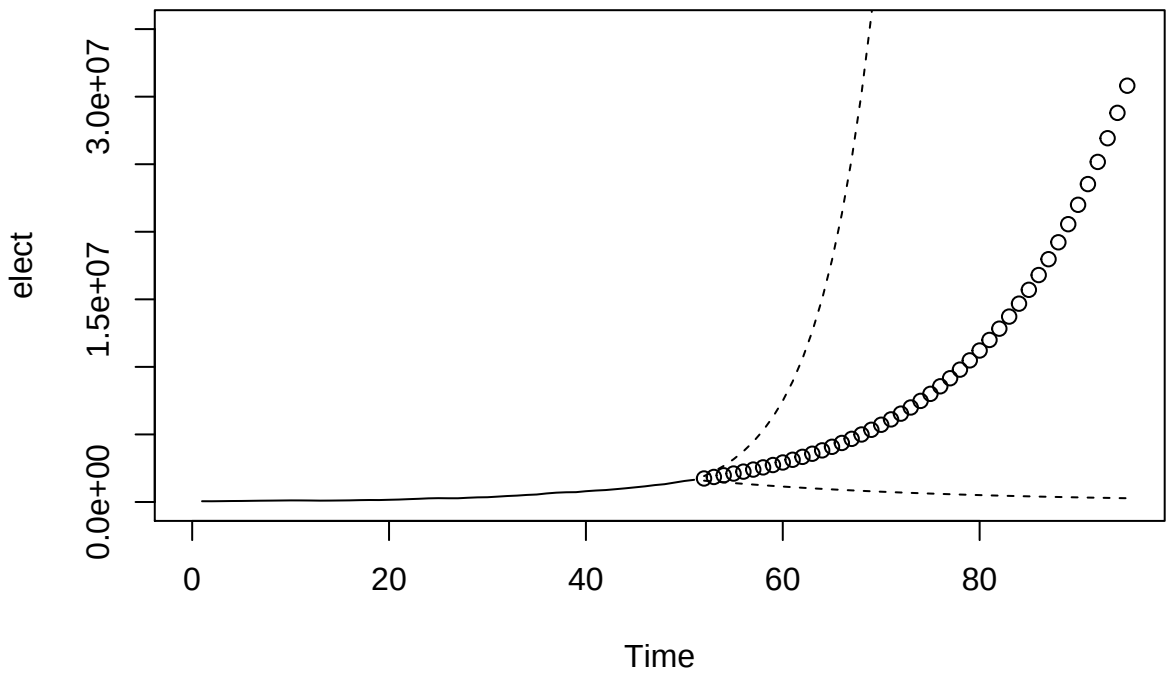
```
mypred <- predict(fit2, n.ahead = 44)
ts.plot(elect_diff1, xlim = c(0, 95), ylim = c(-0.2, 0.21))
points(52:95, mypred$pred)
lines(52:95, mypred$pred + 1.96 * mypred$se, lty = 2)
lines(52:95, mypred$pred - 1.96 * mypred$se, lty = 2)
```



The below code and results takes into account the predictions and standard deviations of the differences of the transformed time series, and transforms both the original differences plus the prediction of the differences back into the original form of the data. We make use of the standard deviation of the differences to create 95% confidence intervals of the predictions as well.

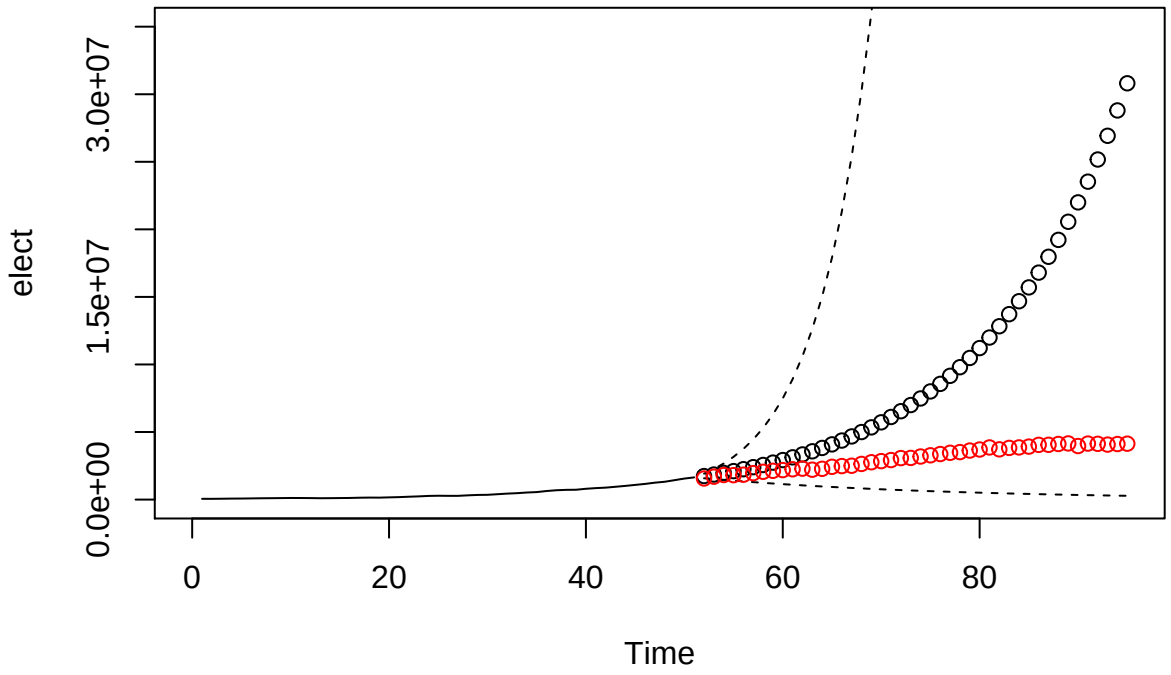
```
first_actual <- elect_bc[1]
first_actual2 <- elect_bc[1]
first_actual3 <- elect_bc[1]
bc_vec <- c(first_actual)
differences_vec <- c(elect_diff1, mypred$pred)
for (i in 1:length(differences_vec)) {
    first_actual <- first_actual + differences_vec[i]
    bc_vec <- c(bc_vec, first_actual)
}

bc_vec_upper <- c(first_actual2)
bc_vec_lower <- c(first_actual3)
mypred_upper <- mypred$pred + 1.96 * mypred$se
mypred_lower <- mypred$pred - 1.96 * mypred$se
differences_vec_upper <- c(elect_diff1, mypred_upper)
differences_vec_lower <- c(elect_diff1, mypred_lower)
for (i in 1:length(differences_vec)) {
    first_actual2 <- first_actual2 + differences_vec_upper[i]
    bc_vec_upper <- c(bc_vec_upper, first_actual2)
    first_actual3 <- first_actual3 + differences_vec_lower[i]
    bc_vec_lower <- c(bc_vec_lower, first_actual3)
}

bc_vec_exp <- exp(bc_vec)
bc_vec_upper_exp <- exp(bc_vec_upper)
bc_vec_lower_exp <- exp(bc_vec_lower)

ts.plot(elect, xlim = c(0, 95), ylim = c(0, 3.5e+07))
```

```
points(52:95, bc_vec_exp[52:95])
lines(52:95, bc_vec_upper_exp[52:95], lty = 2)
lines(52:95, bc_vec_lower_exp[52:95], lty = 2)
```



I was unable to find data which directly states annual total electricity consumption from 1971 - 2014. There was data on annual electricity consumption per capita, however, and so making use of data on US population (annual), we can obtain approximate data for total electricity consumption from 1971 - 2014 (annually). We therefore have the following figure:

```
ts.plot(elect, xlim = c(0, 95), ylim = c(0, 3.5e+07))
points(52:95, bc_vec_exp[52:95])
lines(52:95, bc_vec_upper_exp[52:95], lty = 2)
lines(52:95, bc_vec_lower_exp[52:95], lty = 2)
points(52:95, elect2[52:95], col = "red")
```



We take a more in-depth look into the confidence intervals of electricity consumption in the US from 1971 - 2014.

```
for (i in 52:95) {
    print(c(bc_vec_lower_exp[i], bc_vec_upper_exp[i]))
}
```

```
## [1] 1578572 1918242
## [1] 1519099 2259414
## [1] 1459798 2670003
## [1] 1401771 3162274
## [1] 1345579 3751132
## [1] 1291475 4454520
## [1] 1239540 5293944
## [1] 1189765 6295118
## [1] 1142089 7488743
## [1] 1096429 8911439
## [1]  1052694 10606867
## [1]  1010792 12627067
## [1]   970631.2 15034048.4
## [1]   932128.2 17901697.4
## [1]   895202.8 21318038.4
## [1]   859780.5 25387936.6
## [1]   825792.2 30236315.3
## [1]   793172.9 36011989.1
## [1]   761862.3 42892230.4
## [1]   731803.4 51088209.9
## [1]   702942.8 60851476.4
## [1]   675230 72481677
## [1]   648617.3 86335752.5
## [1]    623059.3 102838891.7
## [1]    598512.8 122497580.0
## [1]    574936.9 145915142.8
## [1]    552292.5 173810261.0
```

```
## [1]    530541.9 207039026.7
## [1]    509649.7 246621215.7
## [1]    489581.3 293771583.5
## [1]    470304.2 349937144.9
## [1]    451786.9 416841581.8
## [1]    433999.2 496538140.8
## [1]    416912.3 591472645.9
## [1]    400498.5 704558558.2
## [1]    384731.2 839266387.0
## [1]    369584.8 999730196.8
## [1]     355034.9 1190874477.6
## [1]     341057.9 1418565272.5
## [1]     327631.3 1689790201.0
## [1]     314733.3 2012872902.4
## [1]     302343.1 2397728479.9
## [1]     290440.7 2856167785.8
## [1]     279007 3402259885
```

Now we take a look at the estimated total electricity consumption in the US from 1971 - 2014:

```
for (i in 52:95) {
    print(elect2[i])
}
```

```
## [1] 1561051
## [1] 1695206
## [1] 1816737
## [1] 1807051
## [1] 1840607
## [1] 1955507
## [1] 2056531
## [1] 2128034
## [1] 2183190
## [1] 2240976
## [1] 2289312
## [1] 2211108
## [1] 2277603
## [1] 2424841
## [1] 2477790
## [1] 2503249
## [1] 2637766
## [1] 2762431
## [1] 2846300
## [1] 2923917
## [1] 3069715
## [1] 3082005
## [1] 3187003
## [1] 3277277
## [1] 3370975
## [1] 3462871
## [1] 3514503
## [1] 3628794
## [1] 3706173
## [1] 3857457
## [1] 3717880
```

```
## [1] 3824317
## [1] 3860609
## [1] 3920250
## [1] 4049930
## [1] 4052974
## [1] 4114051
## [1] 4154966
## [1] 3961560
## [1] 4143407
## [1] 4127198
## [1] 4069169
## [1] 4110051
## [1] 4137101
```

We now verify to see at which points the actual (or rather estimated) data lies within the confidence interval or not.

```
true_lie_within <- c()
for (i in 52:95) {
    if ((elect2[i] > bc_vec_lower_exp[i]) & (elect2[i] < bc_vec_upper_exp[i])) {
        true_lie_within <- c(true_lie_within, TRUE)
    } else {
        true_lie_within <- c(true_lie_within, FALSE)
    }
}
true_lie_within
```

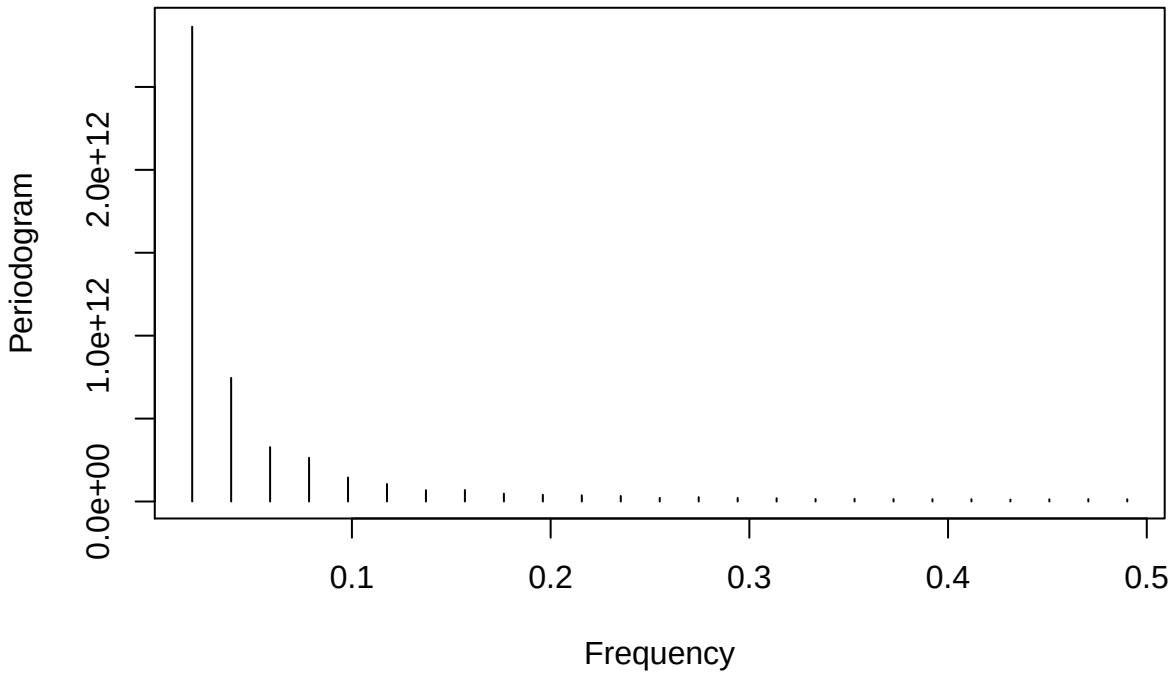
```
##  [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [34]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

We can see from above that the estimated total electricity consumption in only 1971 does not lie within the confidence interval; however, the rest of the 43 estimated points lie within their respective confidence intervals, including the estimated 44th point (actual total electricity consumption for 2014).

Spectral Analysis

Using the periodogram( ) function, we obtain the following periodogram:

```
periodogram_elect <- periodogram(elect)
plot(periodogram_elect$freq, periodogram_elect$spec, xlab = "Frequency",
    ylab = "Periodogram", type = "h")
```



The following are the estimated frequencies:

```
periodogram(elect)$freq
```

```
##  [1] 0.01960784 0.03921569 0.05882353 0.07843137 0.09803922 0.11764706
##  [7] 0.13725490 0.15686275 0.17647059 0.19607843 0.21568627 0.23529412
## [13] 0.25490196 0.27450980 0.29411765 0.31372549 0.33333333 0.35294118
## [19] 0.37254902 0.39215686 0.41176471 0.43137255 0.45098039 0.47058824
## [25] 0.49019608
```

We now perform the Fisher test:

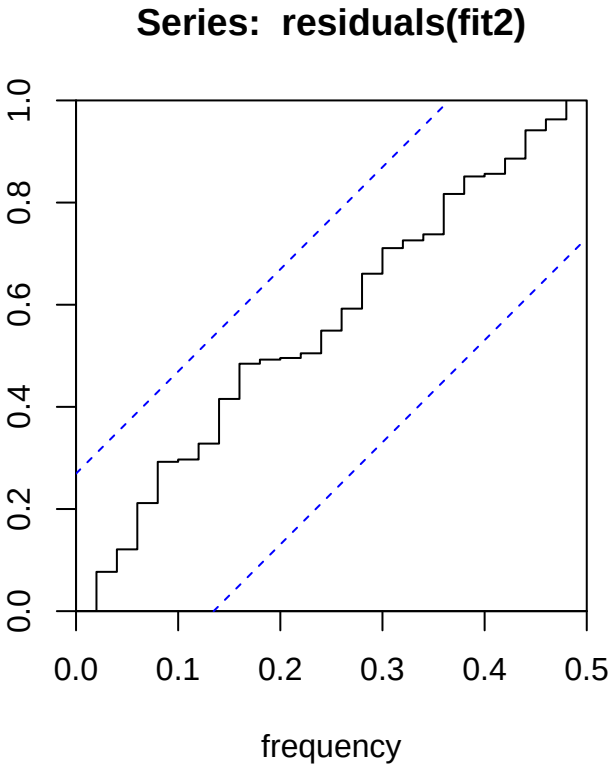
```
fisher.g.test(elect)
```

```
## [1] 2.68546e-08
```

Using the fisher.g.test( ) function, we obtain a p-value of 2.68546e-08, which is less than 0.05. Therefore, our original data set does not pass Fisher's test for Gaussian White Noise, and so our data set is not Gaussian White Noise.

We now perform the Kolmogorov-Smirnov Test, in which we are checking as to whether the residuals are Gaussian White Noise. Using the cpgram( ) function, we obtain the following plot:

```
cpgram(residuals(fit2))
```



From the above graph, we can see that our function, outlined in black, does not exit the blue- dotted line boundaries, which means we do not reject the hypothesis that the residuals follow the Gaussian white noise model.

Conclusion

My final model of the transformed data is: $X_t - 0.8042X_{t-1} = Z_t - 0.5456Z_{t-1} - 0.4544Z_{t-2} + 0.0674$. Although I was able to find a model taking into consideration AICc and diagnostic check, we also had to take into consideration the increasing trend in total electricity consumption as time passes. Therefore, my new model takes into account the increase in total electric consumption as time passes as evident in the data of my estimations in years 1971 - 2014. In addition, my model satisfies the Box-Ljung test, Shapiro-Walk Normality test, and Kolmogorov-Smirnov Test. My model also has one of the lowest AICc value when comparing 99 other possible models.

I thank my TA, Nhan Huynh, and my colleague, Syen Yang Lu, for assisting me with this project.

References

https://datamarket.com/data/set/22vi/total-electricity-consumption-us-kilowatt-hours-millions- 1920-1970#!ds=22vi&display=line

https://data.worldbank.org/indicator/EG.USE.ELEC.KH.PC

https://fred.stlouisfed.org/series/POPTOTUSA647NWDB