DATA 624 Homework 5

Bin Lin

2018-3-26

8.11 1. Figure 8.24 shows the ACFs for 36 random numbers, 360 random numbers and for 1,000 random numbers.

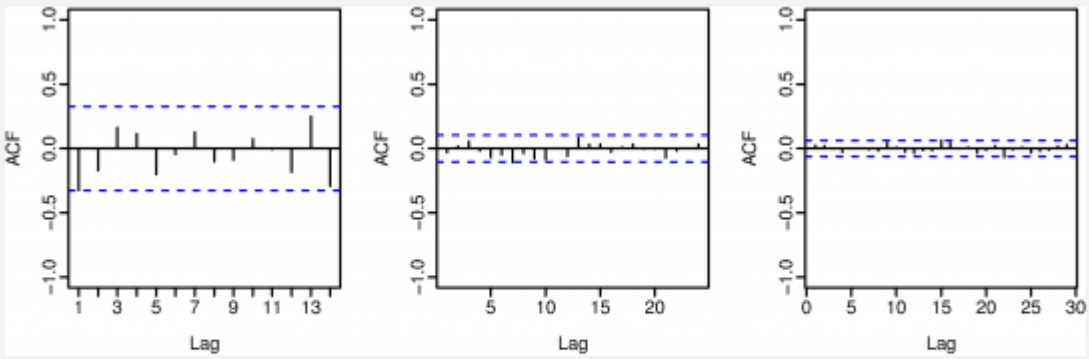


Figure 8.24: Left: ACF for a white noise series of 36 numbers. Middle: ACF for a white noise series of 360 numbers. Right: ACF for a white noise series of 1,000 numbers.

a. Explain the differences among these figures. Do they all indicate the data are white noise?

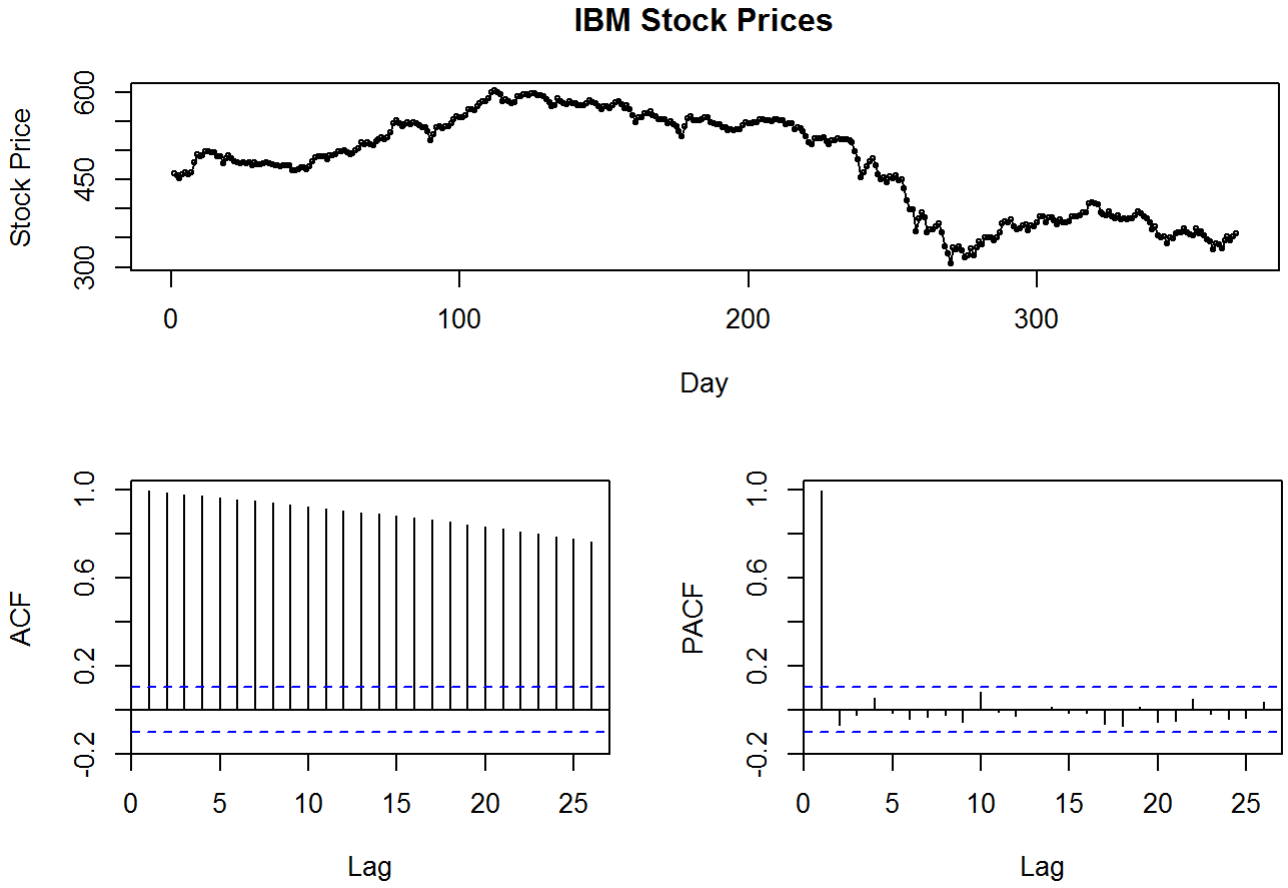
The autocorrelation coefficients are normally plotted to form the autocorrelation function or ACF. The plot is also known as a correlogram. Time series that show no autocorrelation are called “white noise”. If autocorrelations are small enough (less than the bounds), this is the evidence that the data are white noise. Therefore, they all indicate the data are white noise, since majority of the data (at least 95%) are under the bounds. The differences among these figures are the number of lags as shown on the x-axis.

b. Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?

The formular to calculate the bound is +/- (2/sqrt(T)), where T is the length of the time series. The larger the T, the smaller the bounds. FOr these three figures, they different number of data, that is why the critical values are at different distances from the mean of zero.

2 A classic example of a non-stationary series is the daily closing IBM stock prices (data set ibmclose). Use R to plot the daily closing prices for IBM stock and the ACF and PACF. Explain how each plot shows the series is non-stationary and should be differenced.

```
library(forecast)
library(fma)
library(ggplot2)
tsdisplay(ibmclose, main="IBM Stock Prices", ylab = "Stock Price", xlab="Day")
```



Approaches: Autocorrelation measures the linear relationship between lagged values of a time series. ACF is useful to identify non-stationary time series. for non-stationary data, ACF decreases slowly, and vice versa. In addition, the value of ACF tends to be large and positive for non-stationary data. Differencing is one way to make a time series stationary. Basically it means to compute the differences between consecutive observations, so that eliminate trend and seasonality.

Interpretation:

Time Series with trends, or with seasonality, are not stationary. The Time Series graph above shows upward then downward trend. Therefore, it is non-stationary. Moreover, the ACF values are all out of bounds, all of which are decreasing over time. This further proves the data is non-stationary. The PACF means partial autocorrelation function. PACF only describes the direct relationship between an observation and its lag. Based on what is shown above, there is no obvious partial correlation exist (besides the partial correlation with itself).

6. Consider the number of women murdered each year (per 100,000 standard population) in the United States (data set wmurders).

a. By studying appropriate graphs of the series in R, find an appropriate ARIMA(p,d,q) model for these data.

Approaches: Differencing can help stabilize the mean of a time series by removing changes in the level of a time series, and so eliminating trend and seasonality. This is one way to make a non-stationary time series stationary.

Interpretation:

The Time Series graph has no apparent seasonality, but it has strong upward trend before the year 1970 and strong downward trend after 1990. This data is non-stationary. The first difference graph looks stationary, however, the ACF and PACF figure each one of them has on spike that is going out of the bounds. We need to perform Unit Root test and KPSS test to check if more differencing is necessary or not.

```
library(forecast)
library(fma)
library(fpp)
```

```
## Loading required package: expsmooth
```

```
## Loading required package: lmtest
```

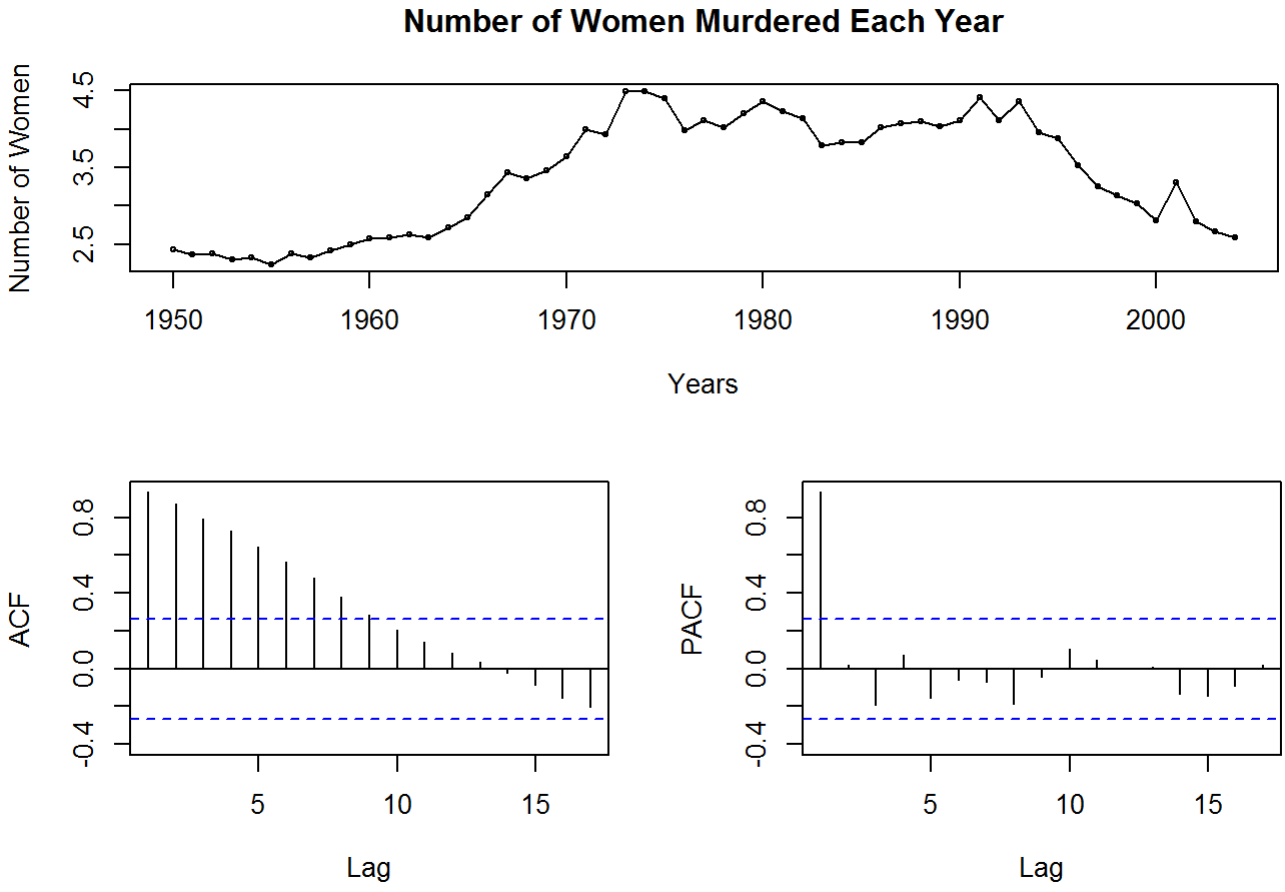
```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

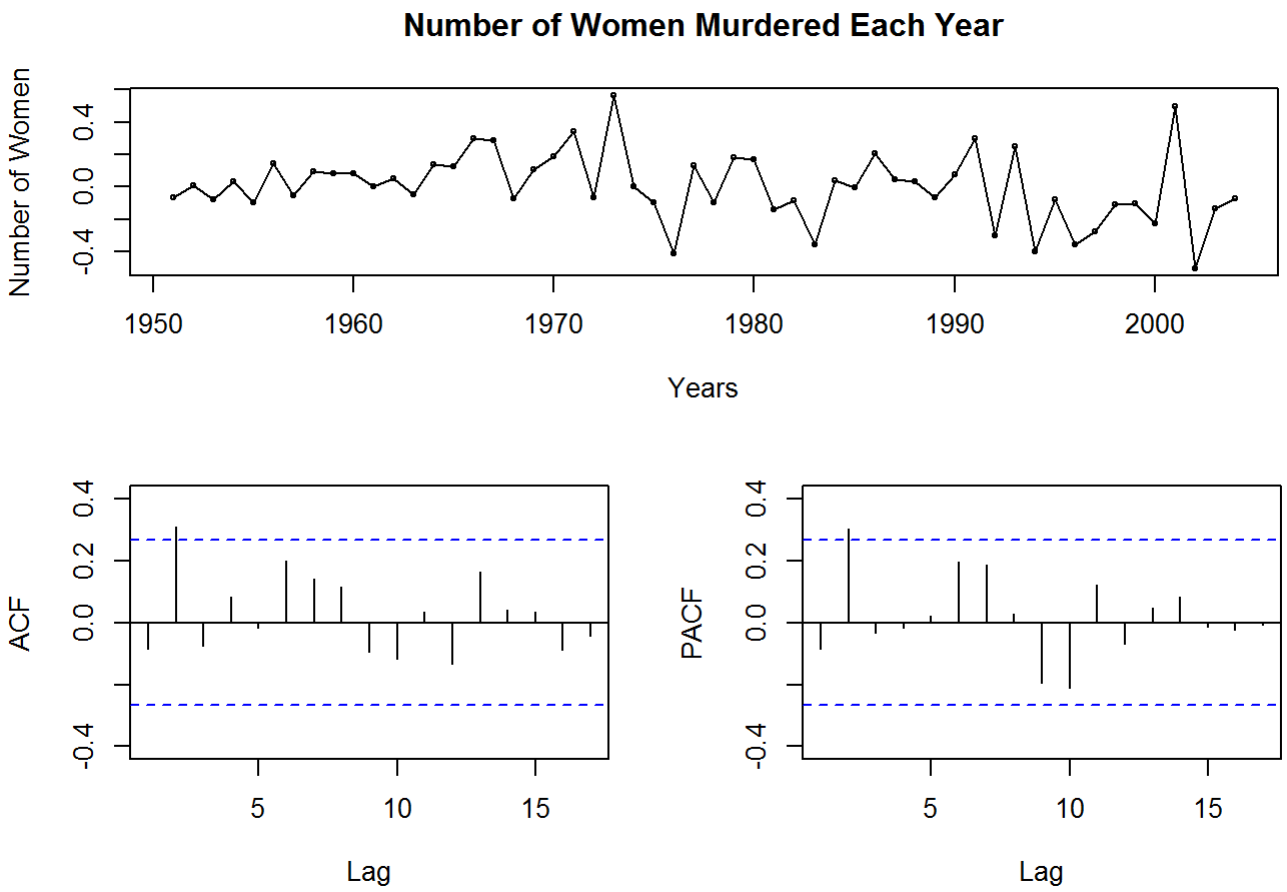
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: tseries
```

```
tsdisplay(wmurders, main="Number of Women Murdered Each Year", ylab = "Number of Women", xlab="Y
ears")
```



```
tsdisplay(diff(wmurders), main="Number of Women Murdered Each Year", ylab = "Number of Women", x
lab="Years")
```



The p-value from the Unit Root Test is 0.02726, which is less than 5%, which indicates that the data is stationary. The p-value from the KPSS Test is 0.02379, which is less than 5%, which indicate that the data is non-stationary. We need to perform second differencing.

```
adf.test(diff(wmurders))
```

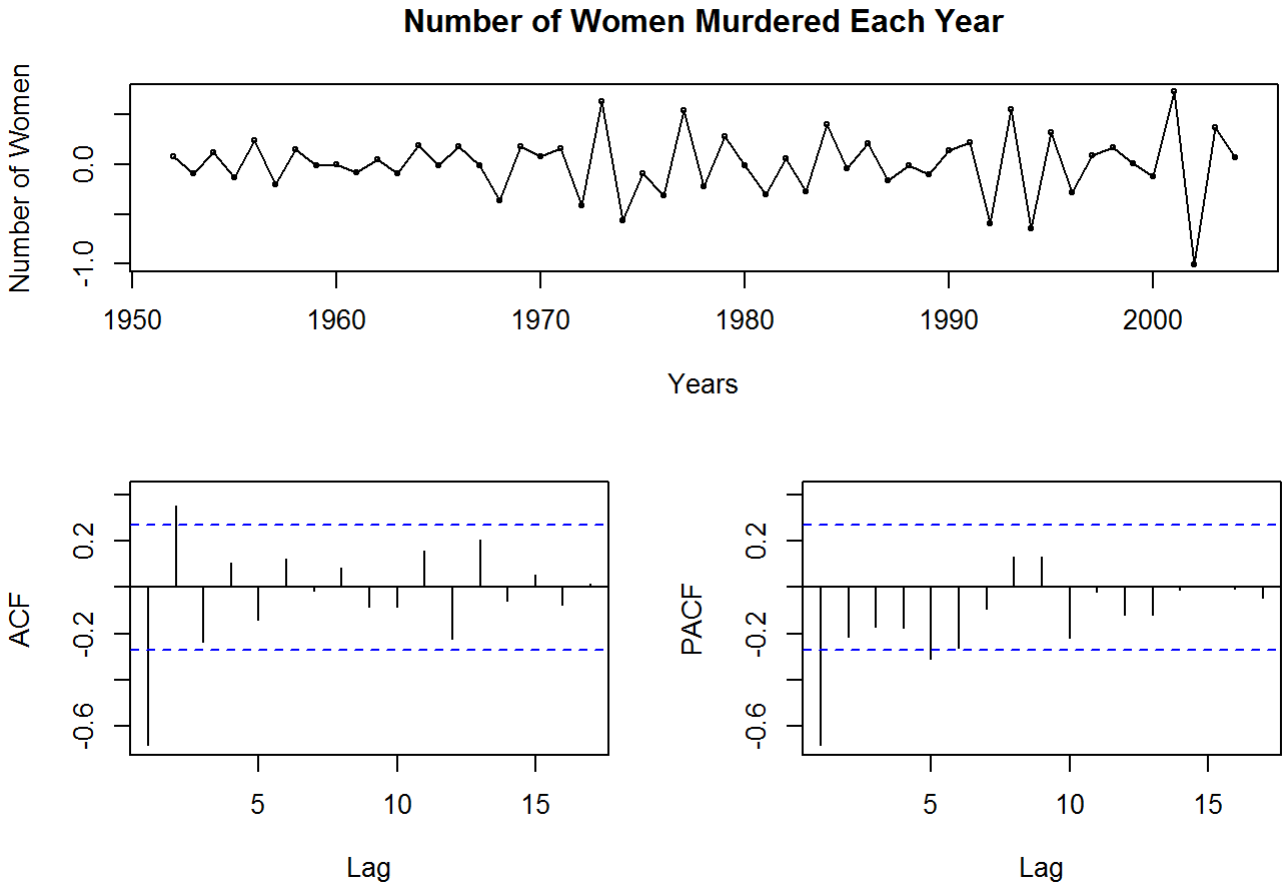
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(wmurders)
## Dickey-Fuller = -3.7688, Lag order = 3, p-value = 0.02726
## alternative hypothesis: stationary
```

```
kpss.test(diff(wmurders))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(wmurders)
## KPSS Level = 0.58729, Truncation lag parameter = 1, p-value =
## 0.02379
```

The second differencing graph appear to be much more stationary. On PACF graph, there is a significant spike at lag 1, but none beyond lag 1, so that we can determine that the p = 1. On ACF graph, there is a significant spike at lag 1, but none beyond lag 1, so that we can determine that the q = 1. Therefore, the appropriate ARIMA(p,d,q) model is ARIMA(1, 2, 1)

```
tsdisplay(diff(diff(wmurders)), main="Number of Women Murdered Each Year", ylab = "Number of Wom
en", xlab="Years")
```




```
adf.test(diff(diff(wmurders)))
```

```
## Warning in adf.test(diff(diff(wmurders))): p-value smaller than printed p-
## value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(wmurders))
## Dickey-Fuller = -5.1646, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(diff(wmurders)))
```

```
## Warning in kpss.test(diff(diff(wmurders))): p-value greater than printed p-
## value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(diff(wmurders))
## KPSS Level = 0.030483, Truncation lag parameter = 1, p-value = 0.1
```

b. Should you include a constant in the model? Explain.

If c=0 and d=2, the long-term forecasts will follow a straight line. If c???0 and d=2, the long-term forecasts will follow a quadratic trend. From the Time Series figure, the data seem to follow a quadratic trend, therefore a constant need to be included.

c. Write this model in terms of the backshift operator.

$$(1 - \quad \phi_1 B)(1 - B)^2 y_t = c + (1 + \theta_1 B)e_t$$

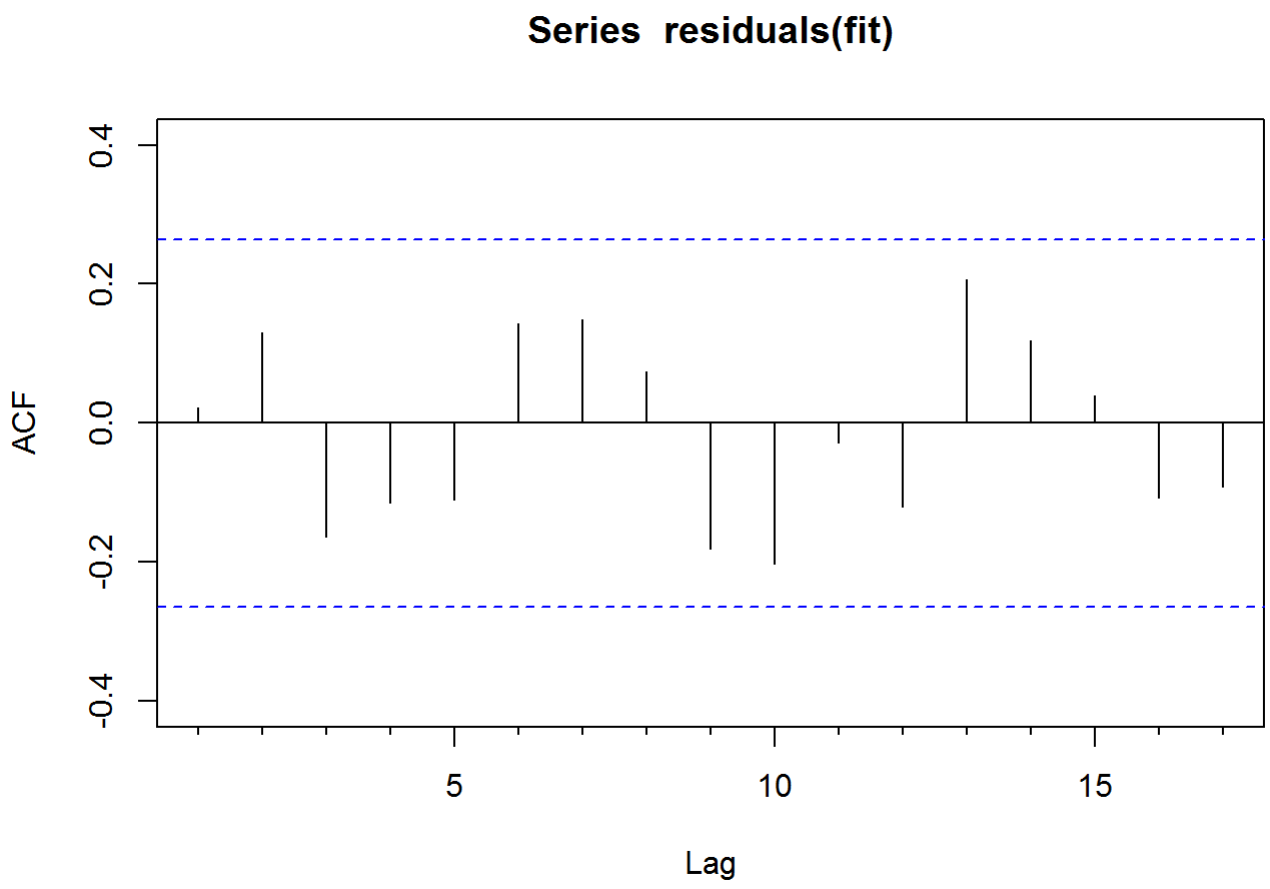
d. Fit the model using R and examine the residuals. Is the model satisfactory?

The ACF plot of the residuals shows all correlations within the threshold limits indicating that the residuals are behaving like white noise. A portmanteau test returns a p-value of 0.1039, which is also greater then the threshold 0.05. Therefore, the result indicates the residuals are white noise, so that the model is satisfactory.

```
fit <- Arima(wmurders, order = c(1, 2, 1), include.constant = TRUE)
summary(fit)
```

```
## Series: wmurders
## ARIMA(1,2,1)
##
## Coefficients:
##           ar1      ma1
##       -0.2434  -0.8261
## s.e.   0.1553   0.1143
##
## sigma^2 estimated as 0.04632:  log likelihood=6.44
## AIC=-6.88   AICc=-6.39   BIC=-0.97
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set -0.01065956 0.2072523 0.1528734 -0.2149476 4.335214 0.9400996
##                     ACF1
## Training set 0.02176343
```

```
Acf(residuals(fit))
```




```
Box.test(residuals(fit), lag=24, fitdf=4, type="Ljung")
```

```
##
##  Box-Ljung test
##
## data:  residuals(fit)
## X-squared = 28.238, df = 20, p-value = 0.1039
```

e. Forecast three times ahead. Check your forecasts by hand to make sure you know how they have been calculated.

Approached: 1. Expand the ARIMA equation so that yt is on the left hand side and all other terms are on the right. 2. Rewrite the equation by replacing t by T+h. 3. On the right hand side of the equation, replace future observations by their forecasts, future errors by zero, and past errors by the corresponding residuals.

```
murder_forecast <- forecast(fit, h = 3)

print(murder_forecast)
```

```
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2005       2.470660 2.194836 2.746484 2.048824 2.892496
## 2006       2.363106 1.986351 2.739862 1.786908 2.939304
## 2007       2.252833 1.765391 2.740276 1.507354 2.998313
```

```
summary(murder_forecast)
```

```
##
## Forecast method: ARIMA(1,2,1)
##
## Model Information:
## Series: wmurders
## ARIMA(1,2,1)
##
## Coefficients:
##           ar1      ma1
##       -0.2434  -0.8261
## s.e.   0.1553   0.1143
##
## sigma^2 estimated as 0.04632:  log likelihood=6.44
## AIC=-6.88   AICc=-6.39   BIC=-0.97
##
## Error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set -0.01065956 0.2072523 0.1528734 -0.2149476 4.335214 0.9400996
##                     ACF1
## Training set 0.02176343
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2005       2.470660 2.194836 2.746484 2.048824 2.892496
## 2006       2.363106 1.986351 2.739862 1.786908 2.939304
## 2007       2.252833 1.765391 2.740276 1.507354 2.998313
```

```
fit$coef["ar1"]
```

```
##        ar1
## -0.2434487
```

```
fit$coef["ma1"]
```

```
##        ma1
## -0.8260877
```

```
wmurders[length(wmurders)]
```

```
## [1] 2.589383
```

```
wmurders[length(wmurders)-1]
```

```
## [1] 2.662227
```

```
wmurders[length(wmurders)-2]
```

```
## [1] 2.797697
```

```
fit$residuals[55]
```

```
## [1] 0.03708172
```

$(1 - \quad \phi_1 B)(1-B)^2 y_t = c + (1 + \theta_1 B) e_t$

Where $\phi_1 = -0.2434$ and $\theta_1 = -0.8261$

$\left[1 - (2+\phi_1)B + (1+2\phi_1)B^2 - \phi_1 B^3\right] y_t = c + (1+\theta_1 B)e_t$

$[y_t - (2+\phi_1)y_{t-1} + (1+2\phi_1)y_{t-2} - \phi_1 y_{t-3}] = c + e_t + \theta_1 e_{t-1}$

$y_t = (2+\phi_1)y_{t-1} - (1+2\phi_1)y_{t-2} + \phi_1 y_{t-3} + e_t + \theta_1 e_{t-1} + c$

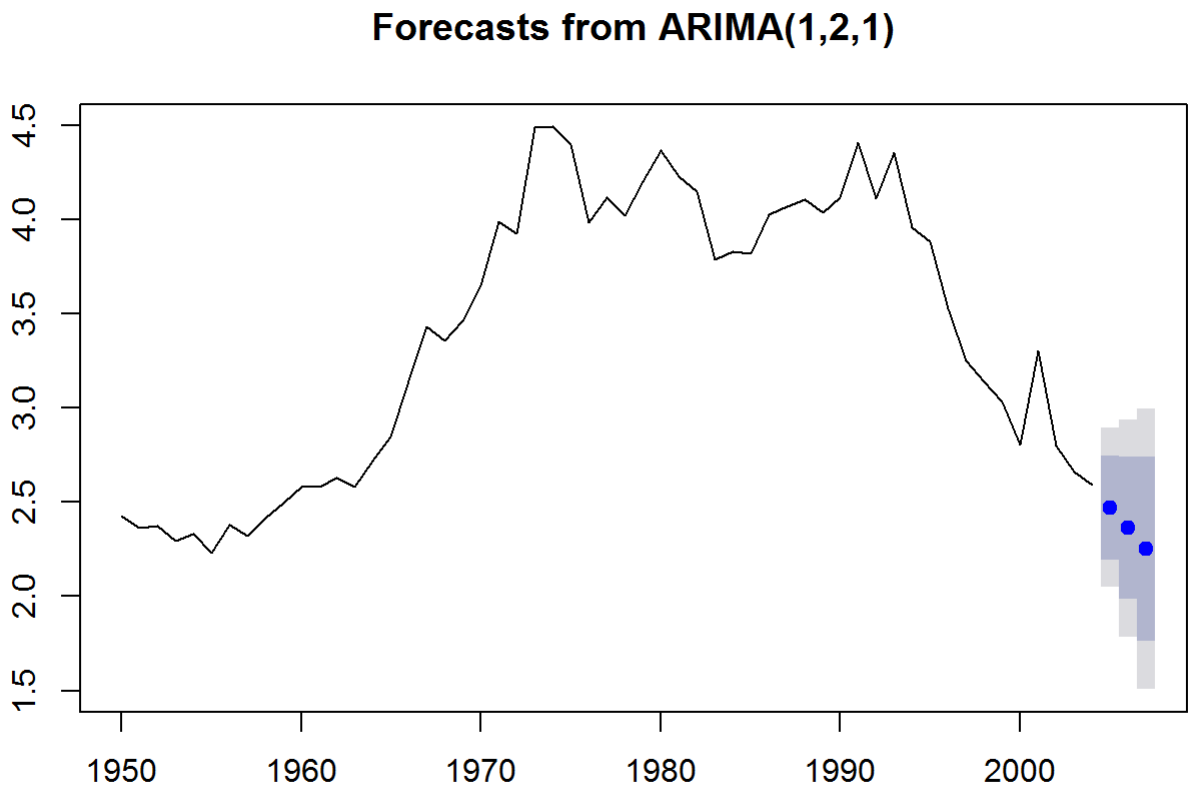
$y_{T+1} = (2+\phi_1)y_T - (1+2\phi_1)y_{T-1} + \phi_1 y_{T-2} + e_{T+1} + \theta_1 e_T + c$

$y_{T+2} = (2+\phi_1)y_{T+1} - (1+2\phi_1)y_T + \phi_1 y_{T-1} + e_{T+2} + \theta_1 e_{T+1} + c$

$y_{T+3} = (2+\phi_1)y_{T+2} - (1+2\phi_1)y_{T+1} + \phi_1 y_T + e_{T+3} + \theta_1 e_{T+2} + c$

f. Create a plot of the series with forecasts and prediction intervals for the next three periods shown.

```
plot(murder_forecast)
```



g. Does auto.arima give the same model you have chosen? If not, which model do you think is better?

The auto.arima give the same model I have chosen. The better model is the model that can minimize the AIC and BIC, since both models are the same. They are equally well.

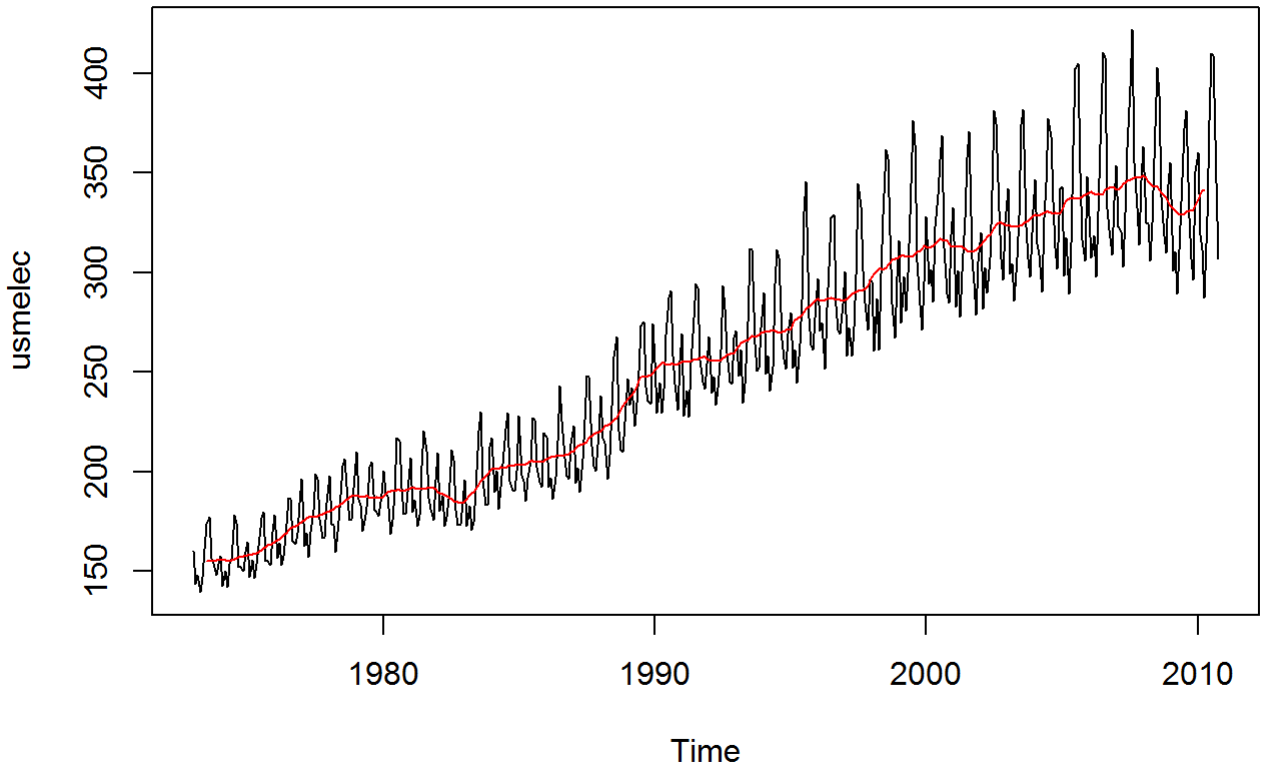
```
fit1 <- auto.arima(wmurders, seasonal=FALSE)
summary(fit1)
```

```
## Series: wmurders
## ARIMA(1,2,1)
##
## Coefficients:
##           ar1      ma1
##       -0.2434  -0.8261
## s.e.   0.1553   0.1143
##
## sigma^2 estimated as 0.04632:  log likelihood=6.44
## AIC=-6.88   AICc=-6.39   BIC=-0.97
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE      MASE
## Training set -0.01065956 0.2072523 0.1528734 -0.2149476 4.335214 0.9400996
##                    ACF1
## Training set 0.02176343
```

8. Consider the total net generation of electricity (in billion kilowatt hours) by the U.S. electric industry (monthly for the period 1985-1996). (Data set usmelec.) In general there are two peaks per year: in mid-summer and mid-winter.

a. Examine the 12-month moving average of this series to see what kind of trend is involved.

```
plot(usmelec)
lines(ma(usmelec, order = 12), col = "red")
```

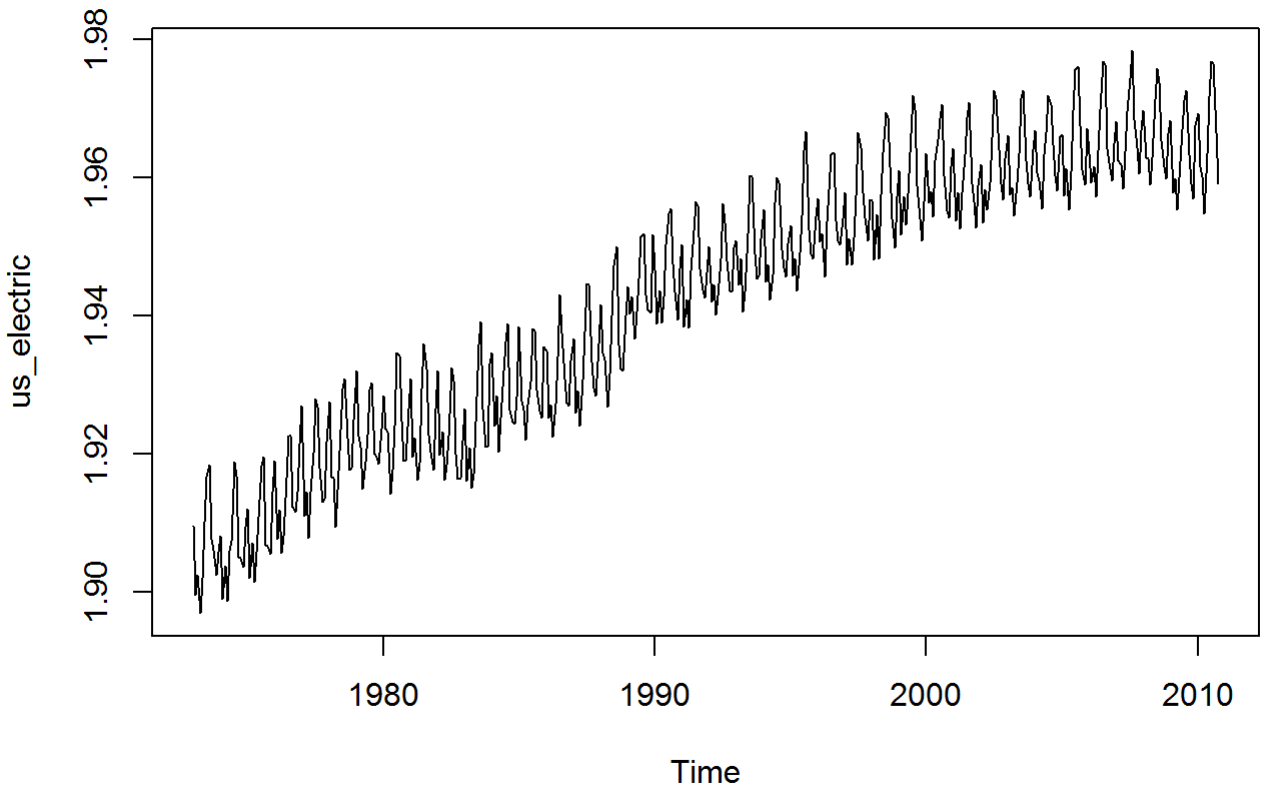


There is increase electricity net generation in the US over years. In addition, it is also associated with strong seasonality factors.

b. Do the data need transforming? If so, find a suitable transformation.

Yes, because the variance of the dataset keep increasing over time. Transformations is necessary to help stabilize the variance of a time series. The Box-Cox Transformation dramatically decrease its variances as shown in the figure. Therefore, it is a suitable transformation.

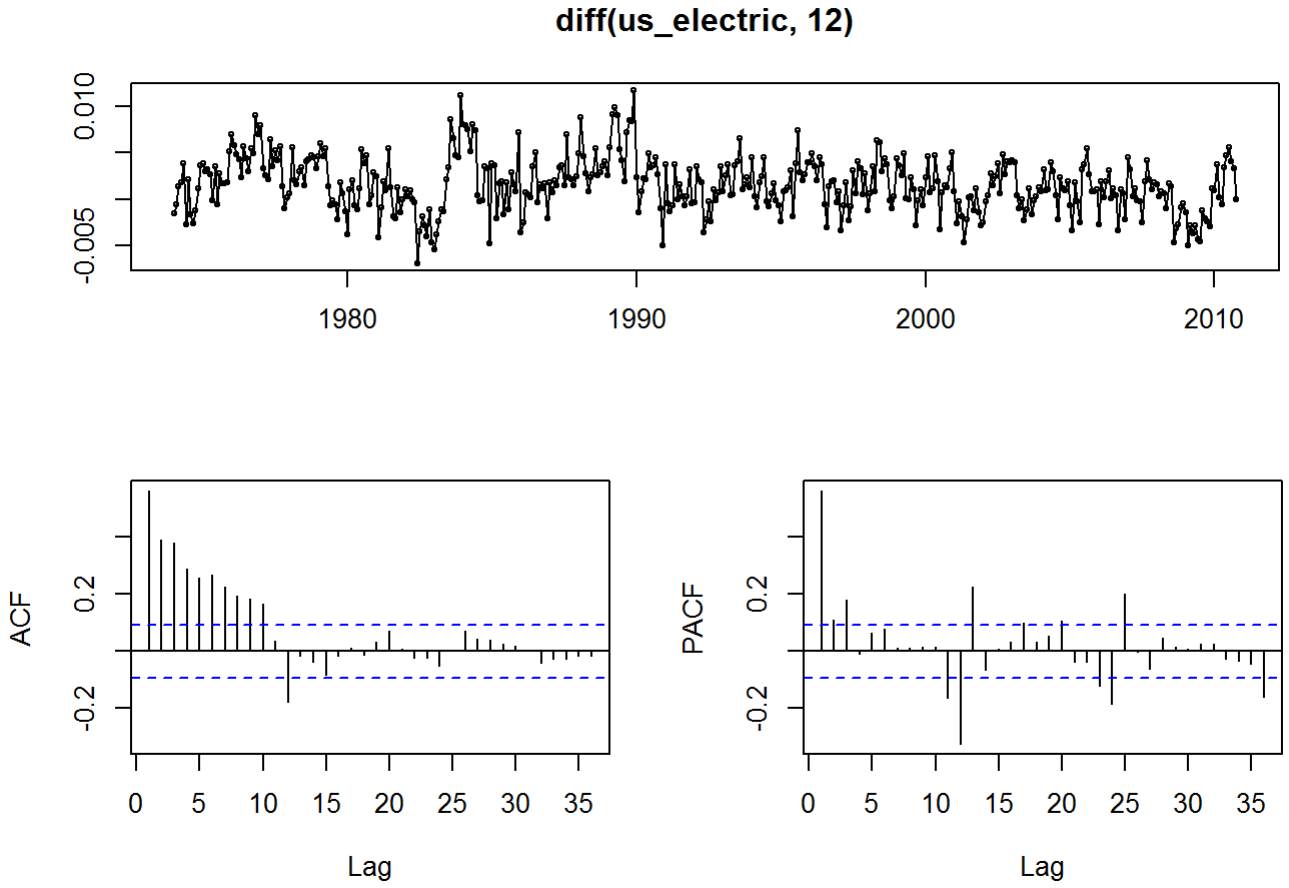
```
lambda <- BoxCox.lambda(usmelec)
us_electric <- BoxCox(usmelec, lambda = lambda)
plot(us_electric)
```



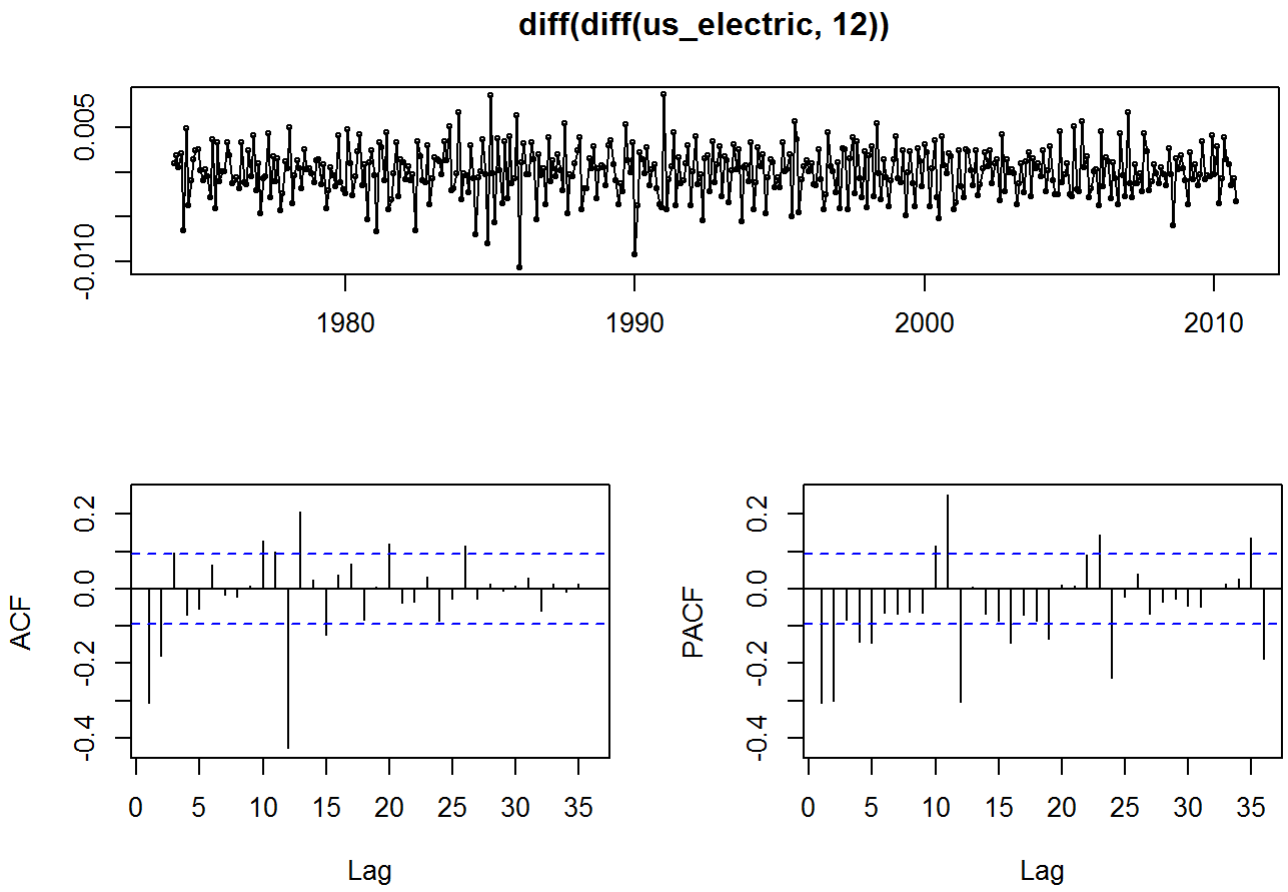
c. Are the data stationary? If not, find an appropriate differencing which yields stationary data.

Time series with trends or with seasonality are not stationary. Therefore, this dataset is not stationary neighther. Since the dataset has strong seasonal pattern, I want to try the seasonal differencing first. The ACF of this differencing drops slowly towards zero. This means the differenced data is still non-stationary. Therefore, I need to perform second-order differencing. The resulting graph looks much better, Both Unit Root test ($p<0.05$) and KPSS ($p>0.05$) test proved that the resulting dataset is stationary now.

```
tsdisplay(diff(us_electric, 12))
```




```
tsdisplay(diff(diff(us_electric, 12)))
```




```
adf.test(diff(diff(us_electric, 12)))
```

```
## Warning in adf.test(diff(diff(us_electric, 12))): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(us_electric, 12))
## Dickey-Fuller = -10.551, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(diff(us_electric, 12)))
```

```
## Warning in kpss.test(diff(diff(us_electric, 12))): p-value greater than
## printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(diff(us_electric, 12))
## KPSS Level = 0.012984, Truncation lag parameter = 4, p-value = 0.1
```

d. Identify a couple of ARIMA models that might be useful in describing the time series. Which of your models is the best according to their AIC values?

The significant spike at lag 1 in the ACF suggests non-seasonal MA component, and the significant spike at lag 12 in the ACF suggests the seasonal MA component. On the other hand, The significant spike at lag 1 in the PACF suggests non-seasonal AR component, and the significant spike at lag 12 in the ACF suggests a seasonal AR component.

Therefore, we can test on few models as shown in the following. ARIMA(0,1,2)(0,1,1)12 is the best models because it has the lowest level of Aic (-4257.311)

```
fit1 <- Arima(usmelec, order=c(0,1,1), seasonal=c(0,1,1), lambda = lambda)
fit2 <- Arima(usmelec, order=c(0,1,2), seasonal=c(0,1,1), lambda = lambda)
fit3 <- Arima(usmelec, order=c(0,1,3), seasonal=c(0,1,1), lambda = lambda)
fit4 <- Arima(usmelec, order=c(1,1,0), seasonal=c(1,1,0), lambda = lambda)
fit5 <- Arima(usmelec, order=c(2,1,0), seasonal=c(1,1,0), lambda = lambda)
fit6 <- Arima(usmelec, order=c(3,1,0), seasonal=c(1,1,0), lambda = lambda)


fit1$aic
```

```
## [1] -4231.601
```

```
fit2$aic
```

```
## [1] -4257.311
```

```
fit3$aic
```

```
## [1] -4256.383
```

```
fit4$aic
```

```
## [1] -4073.213
```

```
fit5$aic
```

```
## [1] -4108.789
```

```
fit6$aic
```

```
## [1] -4109.758
```

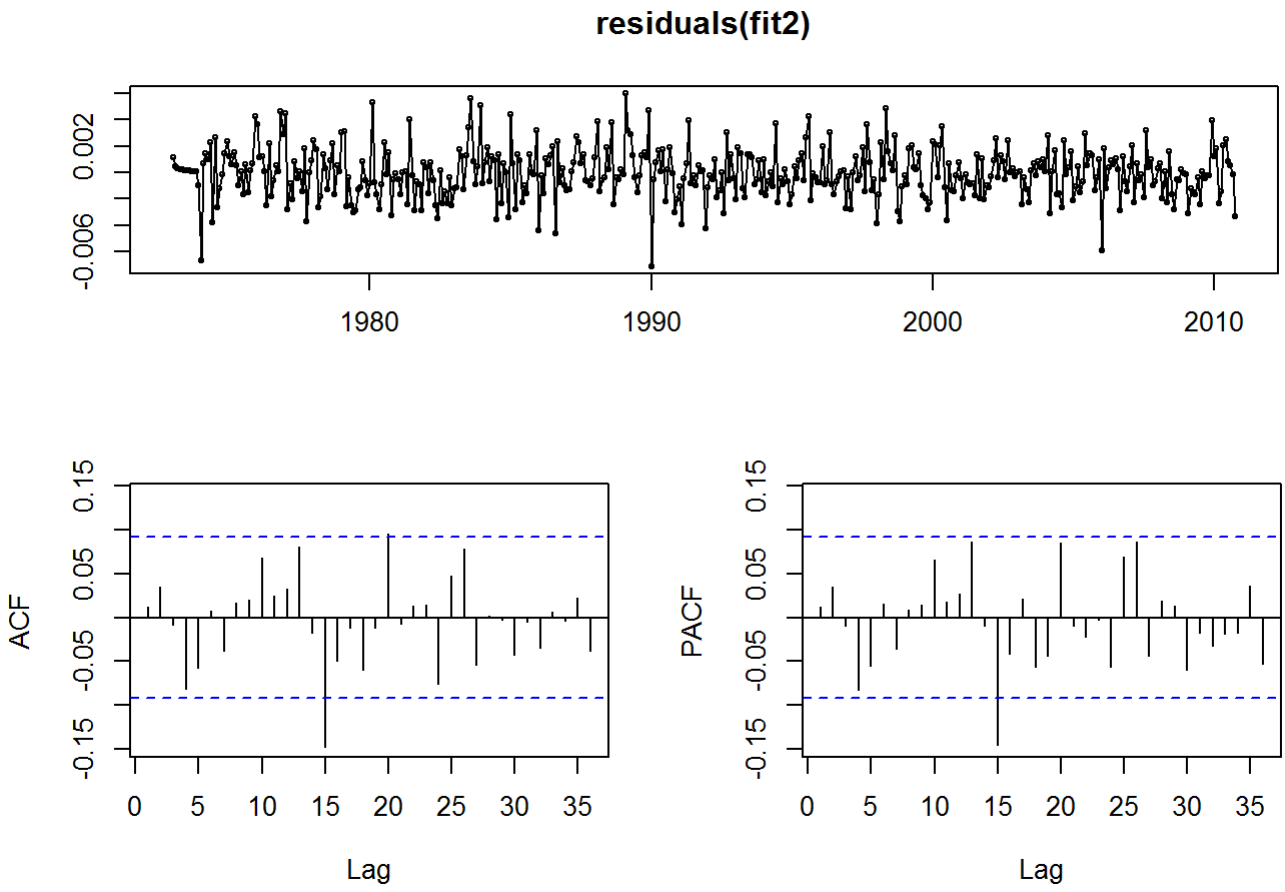
e. Estimate the parameters of your best model and do diagnostic testing on the residuals. Do the residuals resemble white noise? If not, try to find another ARIMA model which fits better.

The following shows the parameters of the model. The ACF and PACF plot of the residuals shows majority of correlations are within the threshold limits (<5%) indicating that the residuals are behaving like white noise. Although the portmanteau test returns a p-value of 0.03357, which is less than the threshold 0.05, the other models have even worse p-value after running each one of them. Therefore, the result indicates the residuals resemble white noise.

```
fit2$coef
```

```
##        ma1        ma2       sma1
## -0.4274796 -0.2570436 -0.8582555
```

```
tsdisplay(residuals(fit2))
```

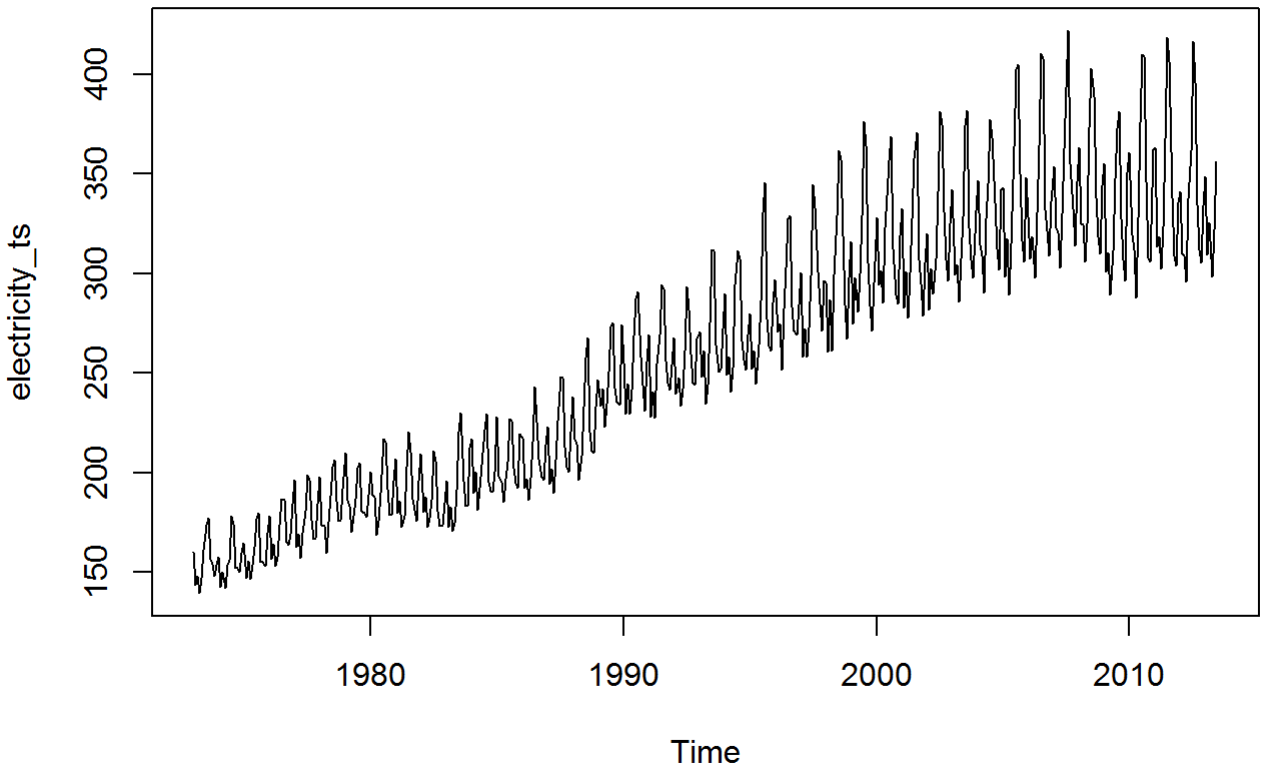



```
Box.test(residuals(fit2), lag=24, fitdf=4, type="Ljung")
```

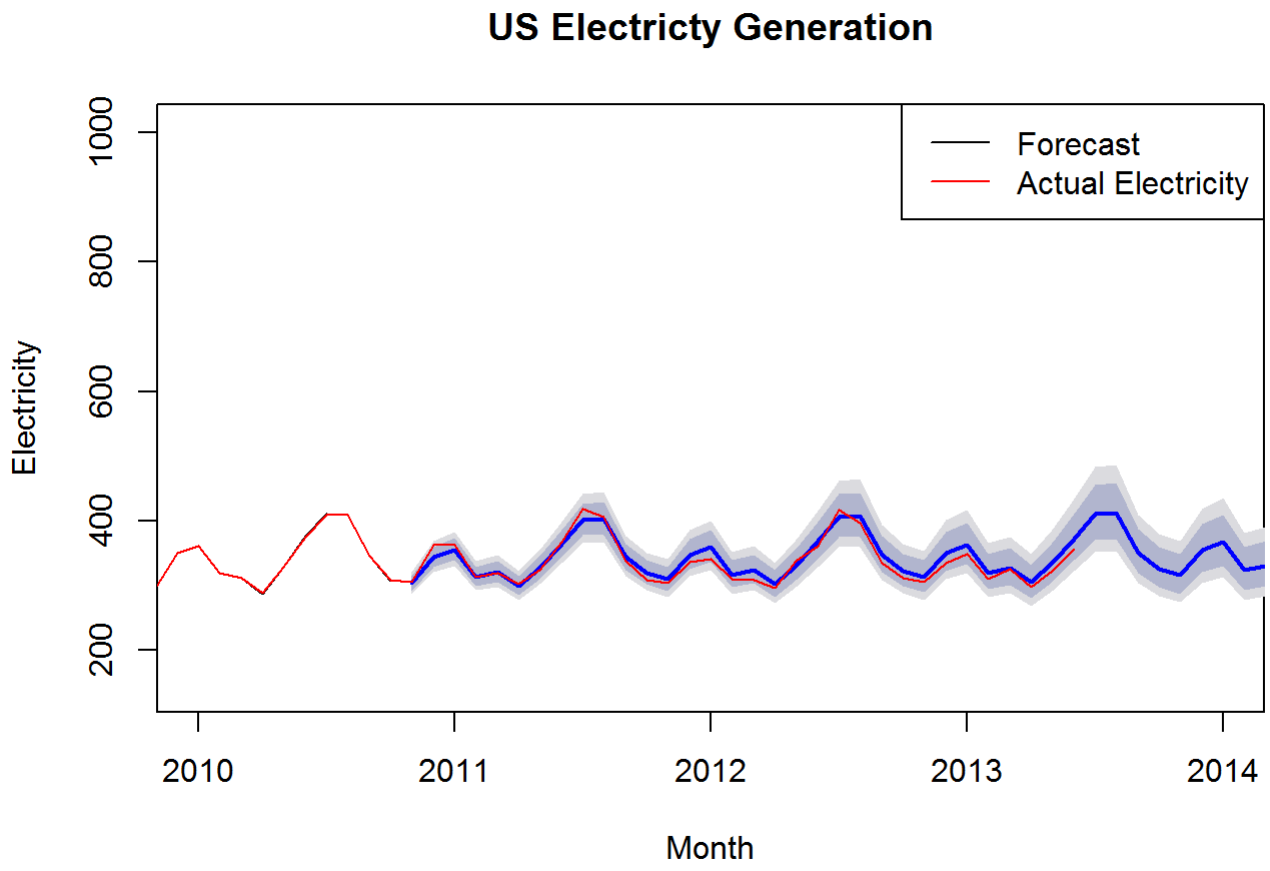
```
##
##  Box-Ljung test
##
## data:  residuals(fit2)
## X-squared = 33.02, df = 20, p-value = 0.03357
```

f. Forecast the next 15 years of generation of electricity by the U.S. electric industry. Get the latest figures from http://data.is/zgRWCO (http://data.is/zgRWCO) to check on the accuracy of your forecasts.

```
actual_data <- read.csv("electricity-overview.csv")
colnames(actual_data) <- c("Month", "Electricity")
electricity_ts <- ts(actual_data$Electricity, start = c(1973, 1), frequency = 12)
plot(electricity_ts)
```




```
e_forecast <- forecast(fit2, h = 180)
plot(e_forecast, main = "US Electricty Generation", ylab = "Electricity", xlab = "Month", xlim =
c(2010, 2014))
lines(electricity_ts, col = "red")
legend("topright", lty = 1, col = c(1, 2), c("Forecast", "Actual Electricity"))
```



The forecast has been very accurate as shown above, also as shown below (high RMSE and MAE values)

```
accuracy(e_forecast, electricity_ts)
```

```
##                      ME      RMSE      MAE       MPE     MAPE      MASE
## Training set -0.3499476  7.121762 5.194804 -0.150549 1.996946 0.5705842
## Test set     -4.3607938 10.349199 8.981722 -1.412878 2.644302 0.9865298
##                    ACF1 Theil's U
## Training set -0.01270233        NA
## Test set      0.45559428 0.3310075
```

g. How many years of forecasts do you think are sufficiently accurate to be usable?

Since the model only accurately predct the data from 10/2010 up to 06/2013 (the most recent available actual data), therefore, the forecast is only accurate for around 5 years.