IS624 Week 6

Aaron Palumbo

Tuesday, July 14, 2015

Contents

HA - 8.1, 8.2, 8.6, 8.8

```r
library(knitr)
library(fpp)

# Suppress messages and warnings in all chuncks
opts_chunk$set(message=FALSE, warning=FALSE)
```

8.1

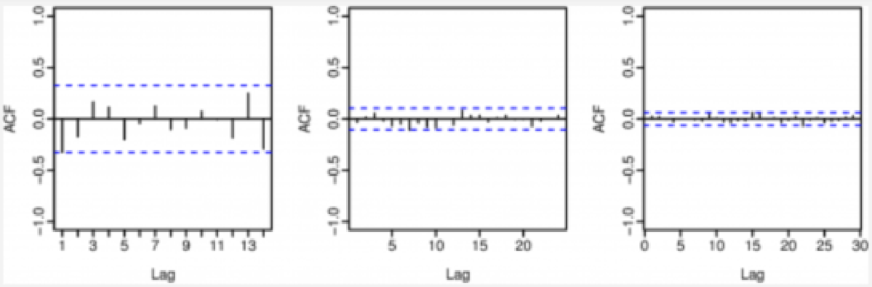


Figure 8.24: Left: ACF for a white noise series of 36 numbers. Middle: ACF for a white noise series of 360 numbers. Right: ACF for a white noise series of 1,000 numbers.

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

8.1 (a)

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

These figures show the correlation between different lags of the series (shown on the x axis). The y axis (the correlation) has the same scale for each plot, but the x axis shows an increasing number of lags as the series gets longer.

If the data are white noise (random) then we expect the correlations to be below the blue line, which indicates a significant lag.

For all the plots, the correlations of the lags shown are all below the significance level so they are all indicitive of white noise.

8.1 (b)

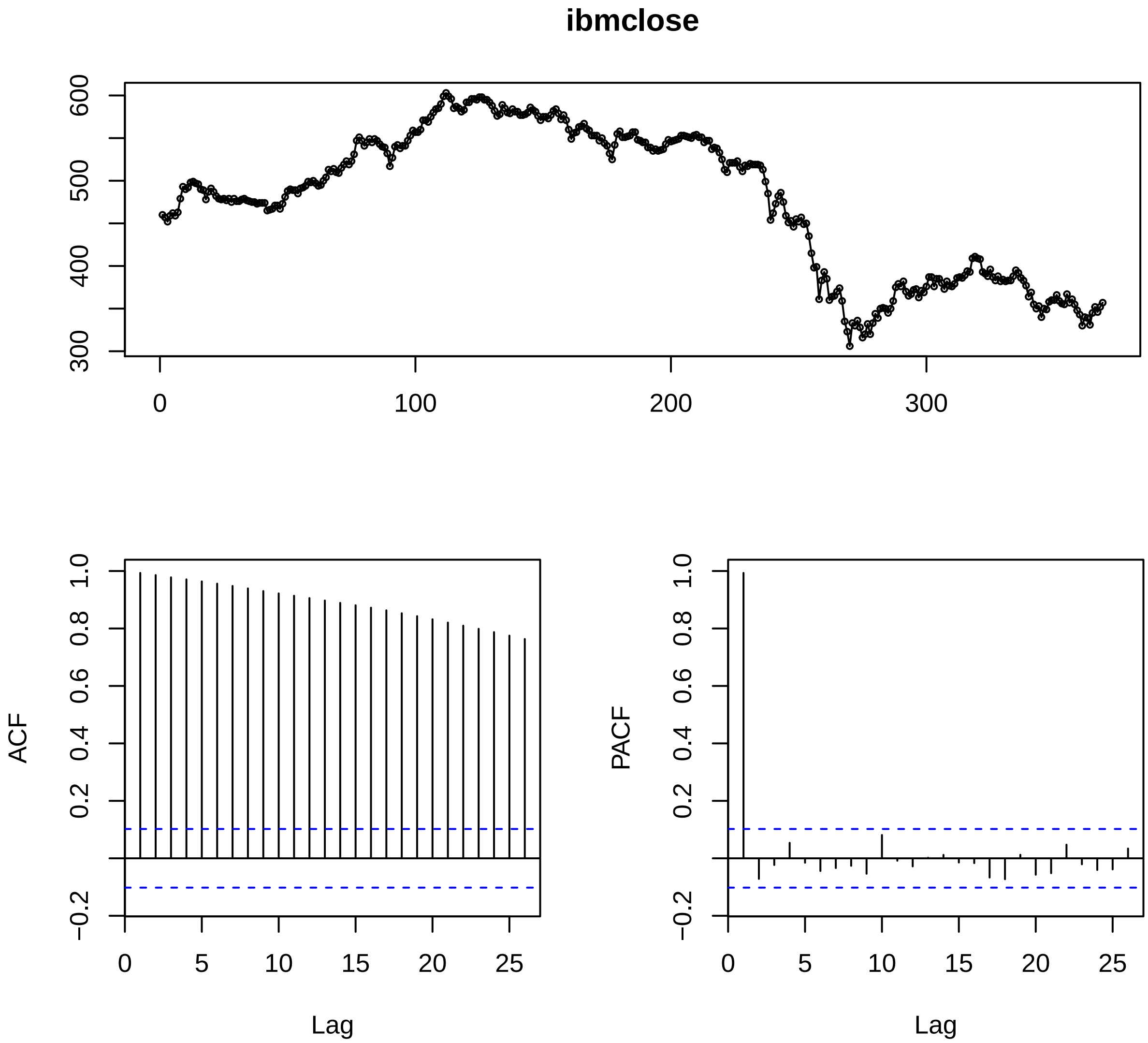
Why are the critical values at different distances from the mean of zero?

The value of a significant correlation is $\pm 1.96/\sqrt{T}$ where T is the number of data. From this we can see that as the number of data increase the value of a significant correlation decreases.

8.2

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.

```r
data(ibmclose)
tsdisplay(ibmclose)
```



When the ACF is slowly decaying, as it is in the graph above, it is a sign that the series may be auto regressive. We then look to the PACF to tell us of what degree. In this case we expect an AR(1) process that will need to be differenced once to be made stationary.

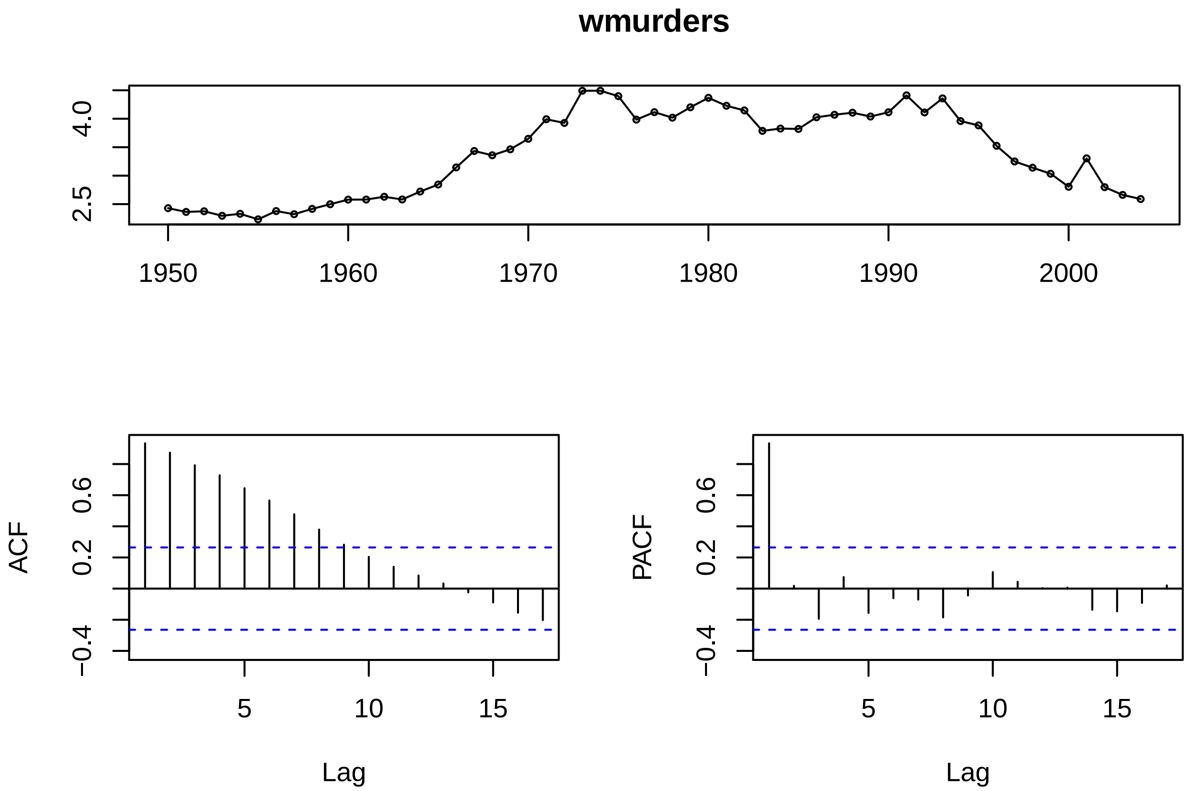
8.6

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

8.6 (a)

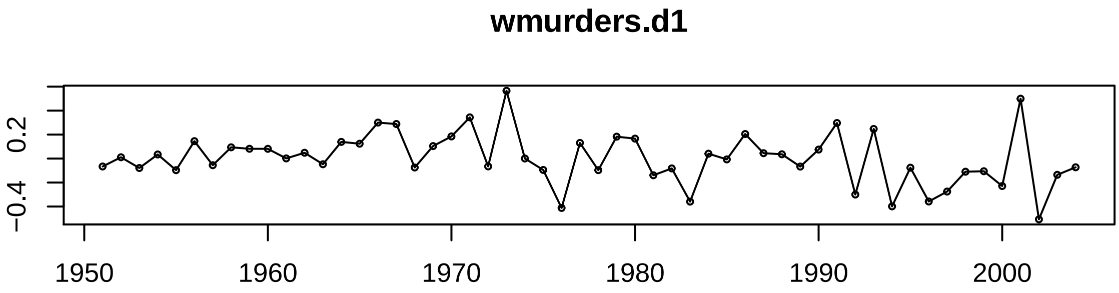
By studying appropriate graphs of the series in R, find an appropriate ARIMA$(\mathbf{p}, \mathbf{d}, \mathbf{q})$ model for these data.

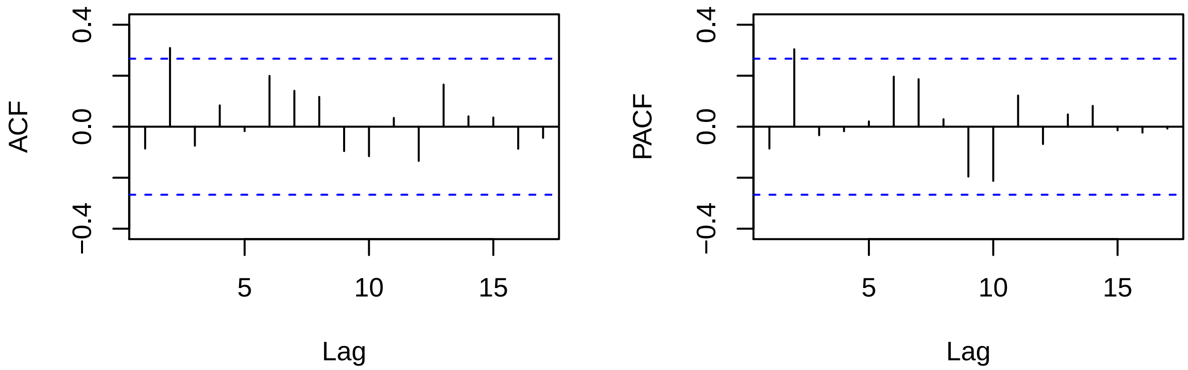
```r
data(wmurders)

tsdisplay(wmurders)
```



This is clearly not stationary. Let's start with taking the first difference.

```r
wmurders.d1 <- diff(wmurders)
tsdisplay(wmurders.d1)
```





This looks much better, but the ACF and PACF still show some significant spikes at a lag of two. Let's try a unit root test:

```
adf.test(wmurders.d1)
```

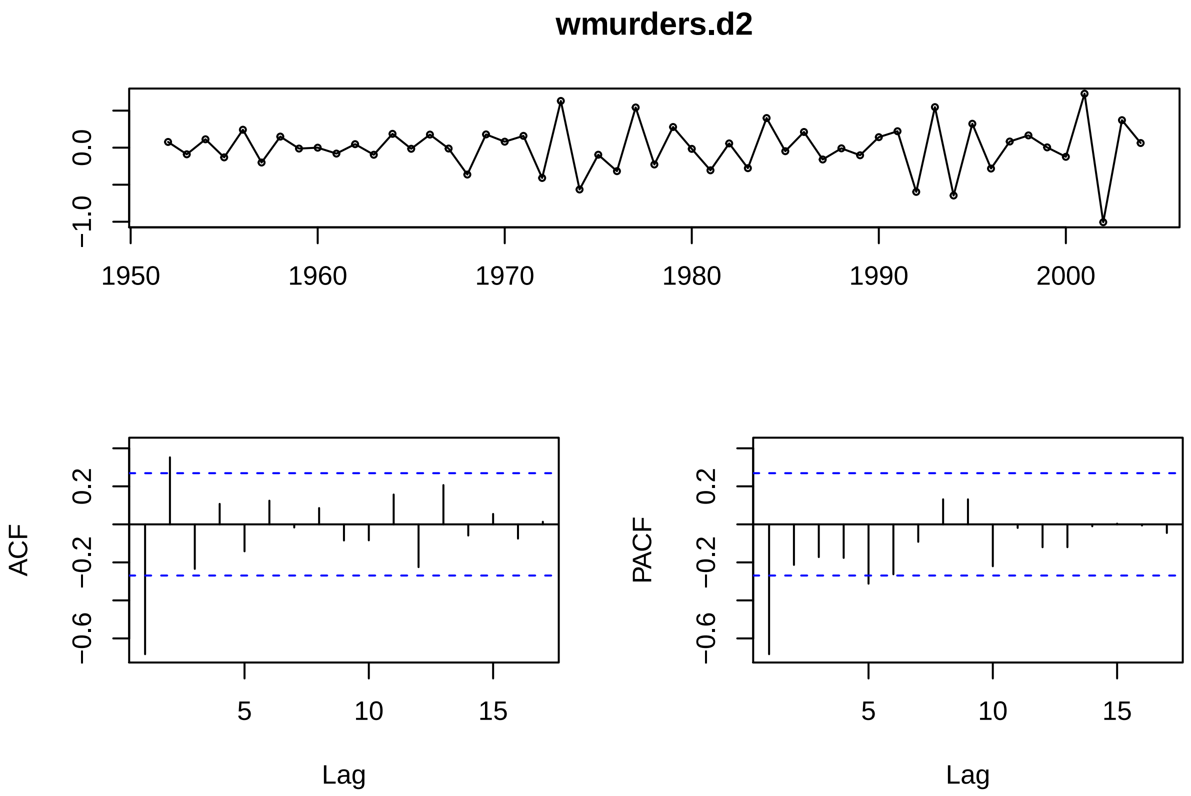
```
##
##  Augmented Dickey-Fuller Test
##
## data:  wmurders.d1
## Dickey-Fuller = -3.7688, Lag order = 3, p-value = 0.02726
## alternative hypothesis: stationary
```

```
kpss.test(wmurders.d1)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  wmurders.d1
## KPSS Level = 0.58729, Truncation lag parameter = 1, p-value =
## 0.02379
```

These tests are telling us different things. An ADF test < 0.05 indicates a stationary series, but a KPSS test < 0.05 indicates a non-stationary series. Take another difference and see what happens:

```r
wmurders.d2 <- diff(diff(wmurders))

tsdisplay(wmurders.d2)
```



```r
# unit root tests
adf.test(wmurders.d2)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  wmurders.d2
## Dickey-Fuller = -5.1646, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(wmurders.d2)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  wmurders.d2
## KPSS Level = 0.030483, Truncation lag parameter = 1, p-value = 0.1
```

Now both unit root tests tell us we have a stationary series.

Looking at the ACF and PACF plots, the large spike at 1 tells us we need either p or q to be 1. Let's start with $p = 1$ and test several ARIMA models in that neighborhood:

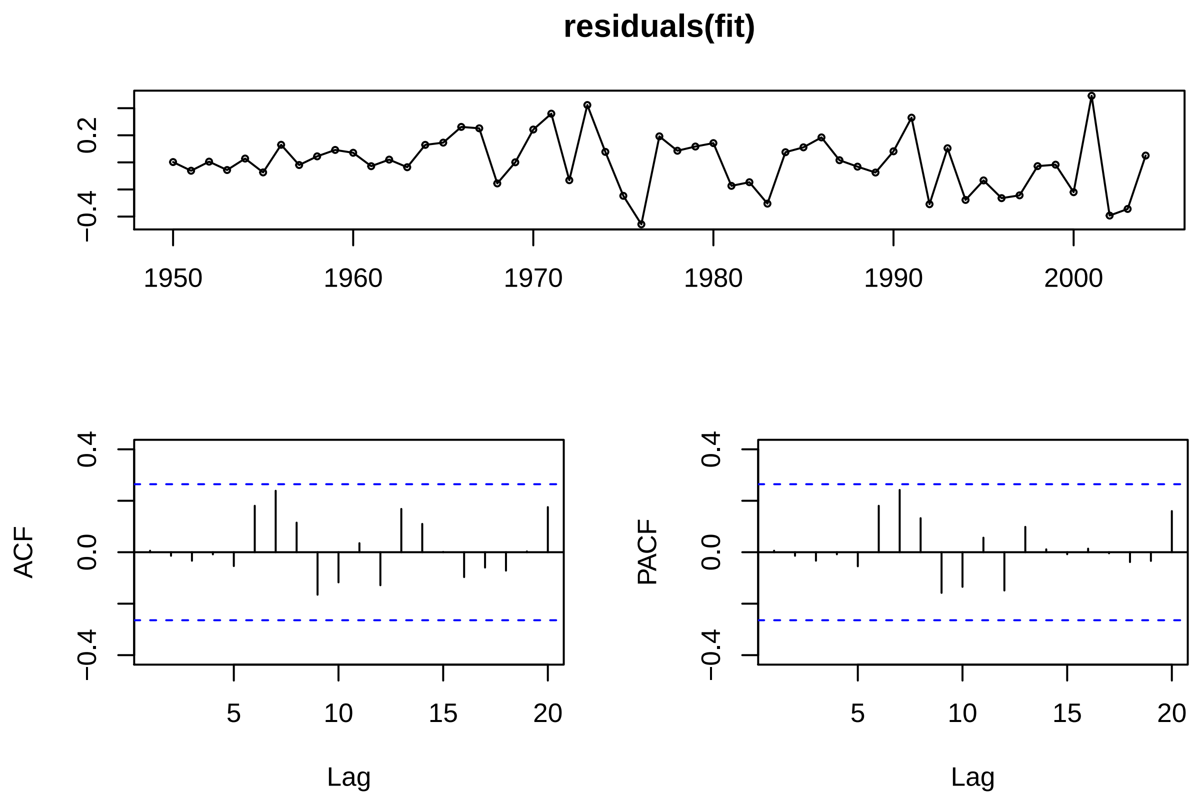
```r
test.arima <- function(t.series, order){
  df <- data.frame(model=paste0("ARIMA(",
                                paste0(order, collapse=","),
                                ")"),
                   AICc=Arima(t.series, order=order)$aicc)
  return(df)
}

gridSearch <- expand.grid(c(0, 1, 2),
                          c(1, 2),
                          c(0, 1, 2))
df.list <- apply(gridSearch, MARGIN=1, FUN=function(x) {test.arima(wmurders, x)})

df <- do.call(rbind, df.list)

kable(df)
```

model	AICc
ARIMA(0,1,0)	-11.3805724
ARIMA(1,1,0)	-9.6107488
ARIMA(2,1,0)	-12.4787165
ARIMA(0,2,0)	30.8394271
ARIMA(1,2,0)	0.2365660
ARIMA(2,2,0)	-0.1881352
ARIMA(0,1,1)	-9.4650495
ARIMA(1,1,1)	-9.8769992
ARIMA(2,1,1)	-10.1696633
ARIMA(0,2,1)	-6.2356449
ARIMA(1,2,1)	-6.3899721
ARIMA(2,2,1)	-6.2849867
ARIMA(0,1,2)	-12.9452661
ARIMA(1,1,2)	-10.6252520
ARIMA(2,1,2)	-12.4643160
ARIMA(0,2,2)	-5.5729679
ARIMA(1,2,2)	-5.9200628
ARIMA(2,2,2)	-3.9800752

Trying several models, including several with $d = 1$ since the residual tests were borderline, we find the model with the lowest AICc to be ARIMA(0,1,2)

Let's fit this model and test the residuals:

```r
fit <- Arima(wmurders, order=c(0, 1, 2))
tsdisplay(residuals(fit), lag.max=20)
```



There are no significant spikes in either the ACF or PACF plots. Let's confirm with a portmanteau test:

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

```
##
##  Box-Ljung test
##
## data:  residuals(fit)
## X-squared = 22.127, df = 20, p-value = 0.3337
```

The portmanteau test indicates the residuals are white noise so we conclude that the best model is ARIMA(0, 1, 2).

8.6 (b)

Should you include a constant in the model? Explain.

No. A constant introduces drift into the model, which we do not appear to have in these data.

8.6 (c)

Write this model in terms of the backshift operator.

$$(1 - B)^2 y_t = (1 + \theta_1 B + \theta_2 B^2) e_t$$

8.6 (d)

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

(see above)

8.6 (e)

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

```
fcast <- forecast(fit, h=3)
fcast$mean
```

```
## Time Series:
## Start = 2005
## End = 2007
## Frequency = 1
## [1] 2.458450 2.477101 2.477101
```

$$
\begin{aligned}
(1-B)^2 y_t &= (1+\theta_1 B+\theta_2 B^2)e_t \\
(1-2B+B^2)y_t &= e_t+\theta_1 Be_t+\theta_2 B^2 e_t \\
y_t-2By_t+B^2 y_t &= e_t+\theta_1 Be_t+\theta_2 B^2 e_t \\
y_t-2y_{t-1}+y_{t-2} &= e_t+\theta_1 e_{t-1}+\theta_2 e_{t-2} \\
y_t &= 2y_{t-1}-y_{t-2}+e_t+\theta_1 e_{t-1}+\theta_2 e_{t-2} \\
y_{t+1} &= 2y_t-y_{t-1}+e_{t+1}+\theta_1 e_t+\theta_2 e_{t-1} \\
y_{t+1} &= 2y_t-y_{t-1}+0+\theta_1 e_t+\theta_2 e_{t-1}
\end{aligned}
$$

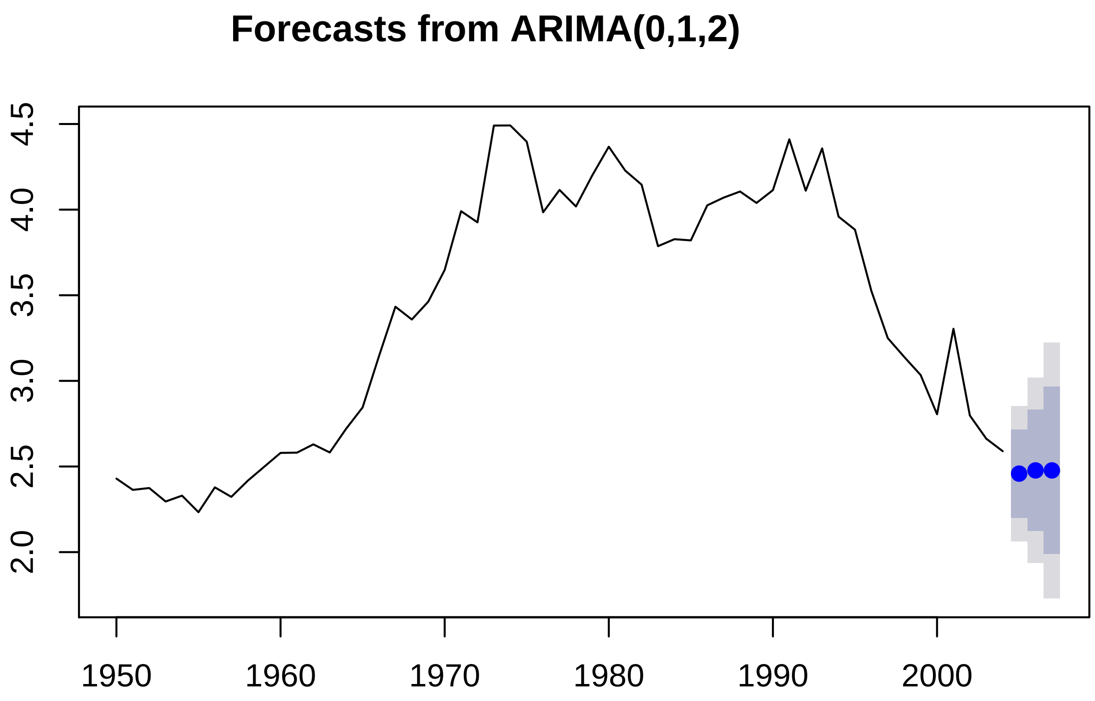
```
toforecast <- 3

yt <- fit$x
et <- fit$residuals
theta1 <- as.numeric(fit$coef['ma2'])
theta2 <- as.numeric(fit$coef['ma1'])

for (h in 1:toforecast){
  n <- length(yt)
  y_tp1 <- 2 * yt[n] - yt[n - 1] + theta1 * et[n] + theta2 * et[n - 1]
  yt <- c(yt, y_tp1)
  et <- c(et, 0)
}

f <- yt[(length(yt) - toforecast + 1):length(yt)]

plot(fcast)
lines(fit$x - fit$residuals, col='blue')
points(c(2005, 2006, 2007), f, col='red')
```

These points do not line up and I'm not sure where my mistake is.

8.6 (f)

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

```
plot(fcast)
```



8.6 (g)

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

```
auto.arima(wmurders)
```

```
## Series: wmurders
## ARIMA(1,2,1)
##
## Coefficients:
##           ar1      ma1
##       -0.2434  -0.8261
## s.e.   0.1553   0.1143
##
## sigma^2 estimated as 0.04457:  log likelihood=6.44
## AIC=-6.88   AICc=-6.39   BIC=-0.97
```

As I said above, the auto.arima function returns ARIMA(1, 2, 1) as the best model. We see in out table that the AICc for that model is -6.4 while the AICc for the model we selected is -11.4. As long as the residuals look okay, I don't see why we wouldn't go with the model with the lower AICc.

8.8

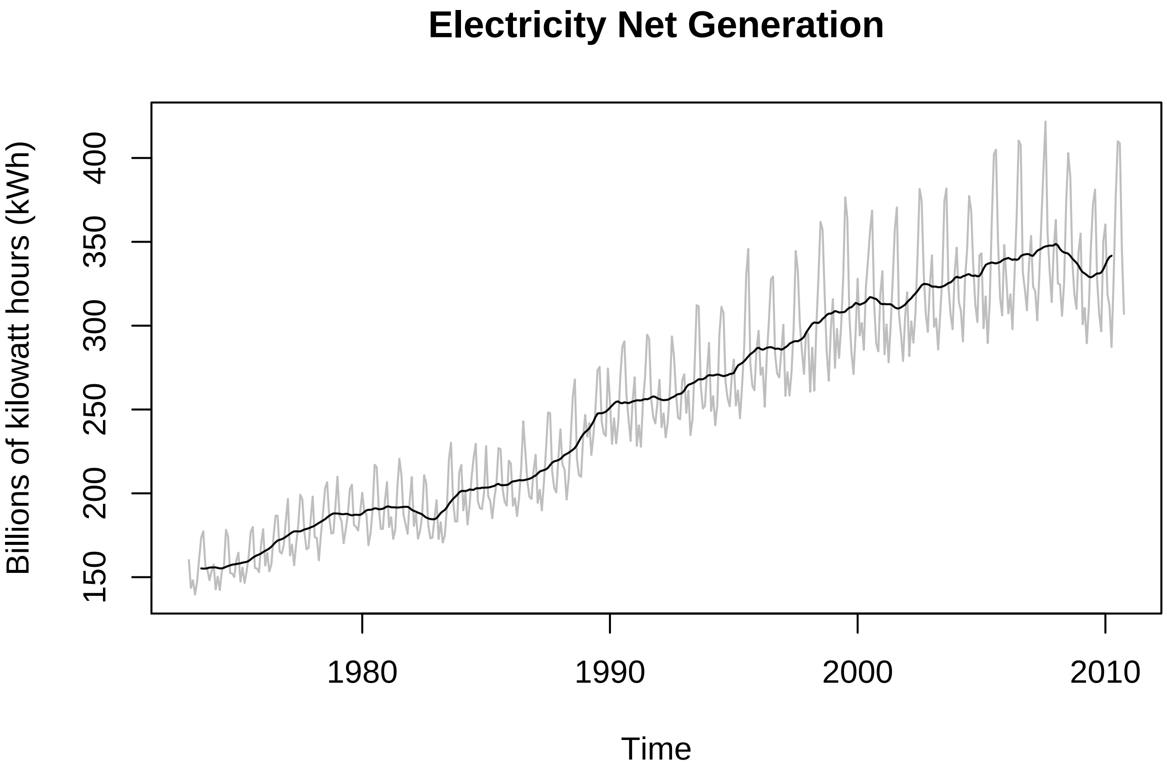
**Consider the total net generation of electricity (in billion kilowatt hours) by the U.S. electric industry (monthly for the period 1985-1996). (Data set usmelec.) In general there are two peaks per year: in mid-summer and mid-winter.

```r
rm(list=ls())
data(usmelec)
```

8.6 (a)

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

```r
movAvg <- ma(usmelec, order=12)
plot(usmelec, col='gray', main="Electricity Net Generation",
     ylab="Billions of kilowatt hours (kWh)")
lines(movAvg)
```



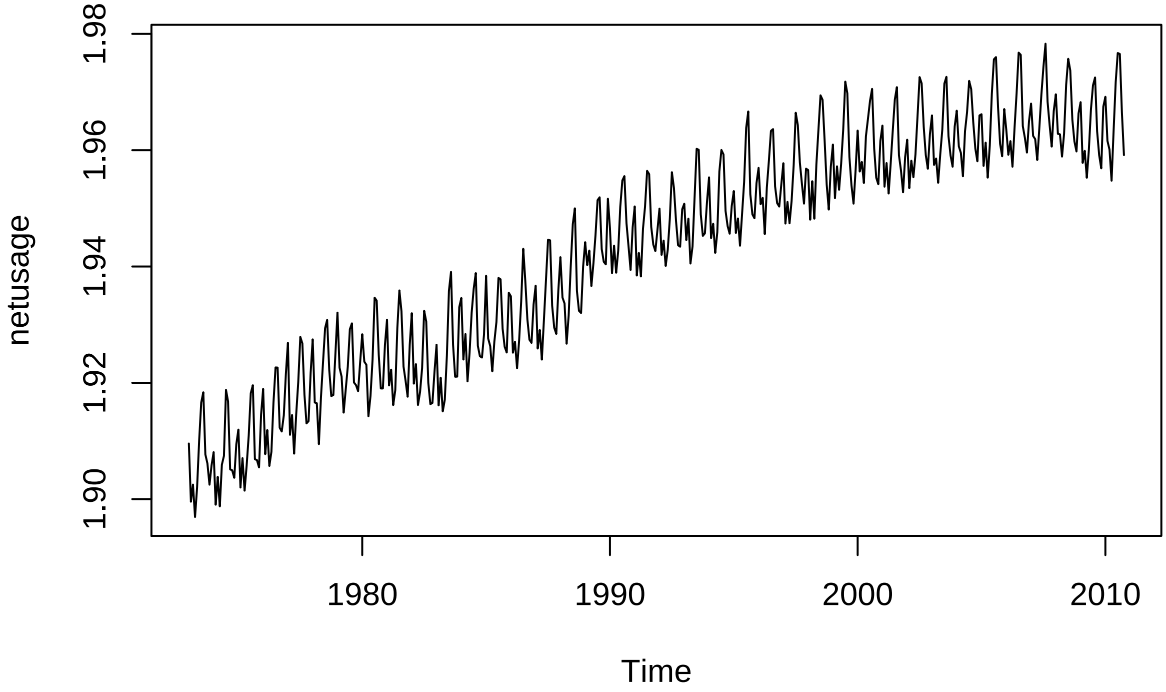
There is an increase in electricity generation over time. There is also a strong season component.

8.6 (b)

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

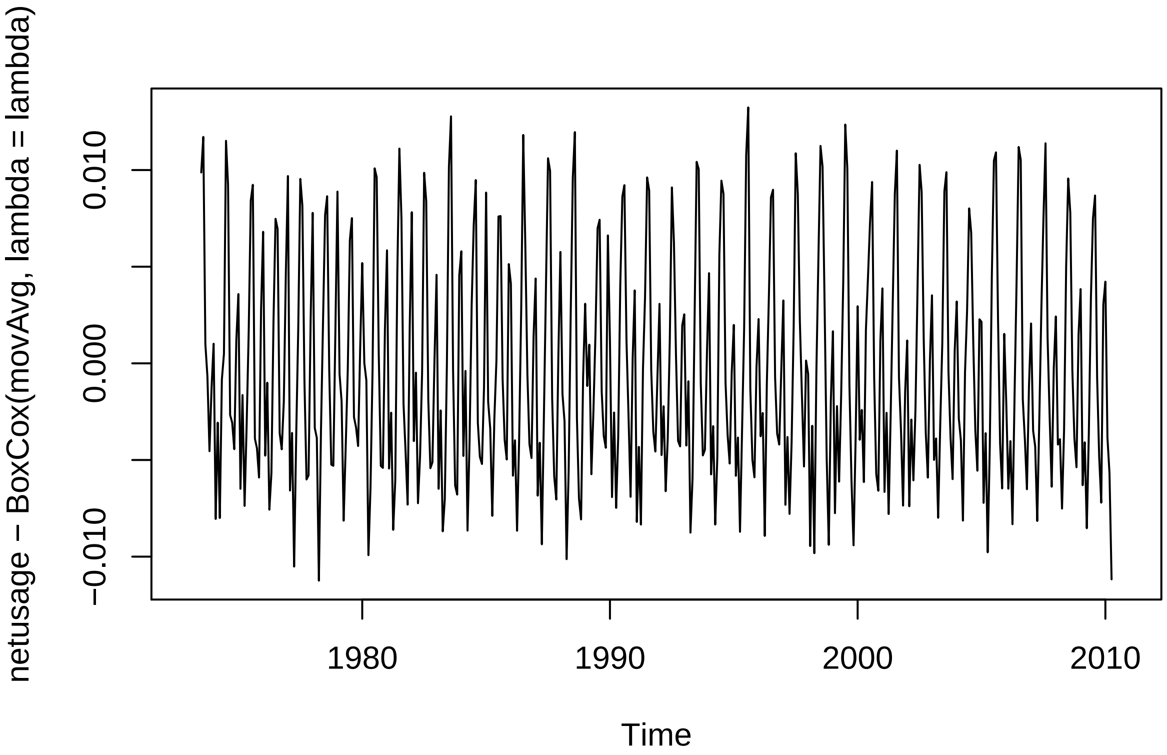
Yes the data need transforming. There is an increase in variance over time.

```
lambda <- BoxCox.lambda(usmelec)
netusage <- BoxCox(usmelec, lambda=lambda)

plot(netusage)
```



We can check this by using a seasonal difference

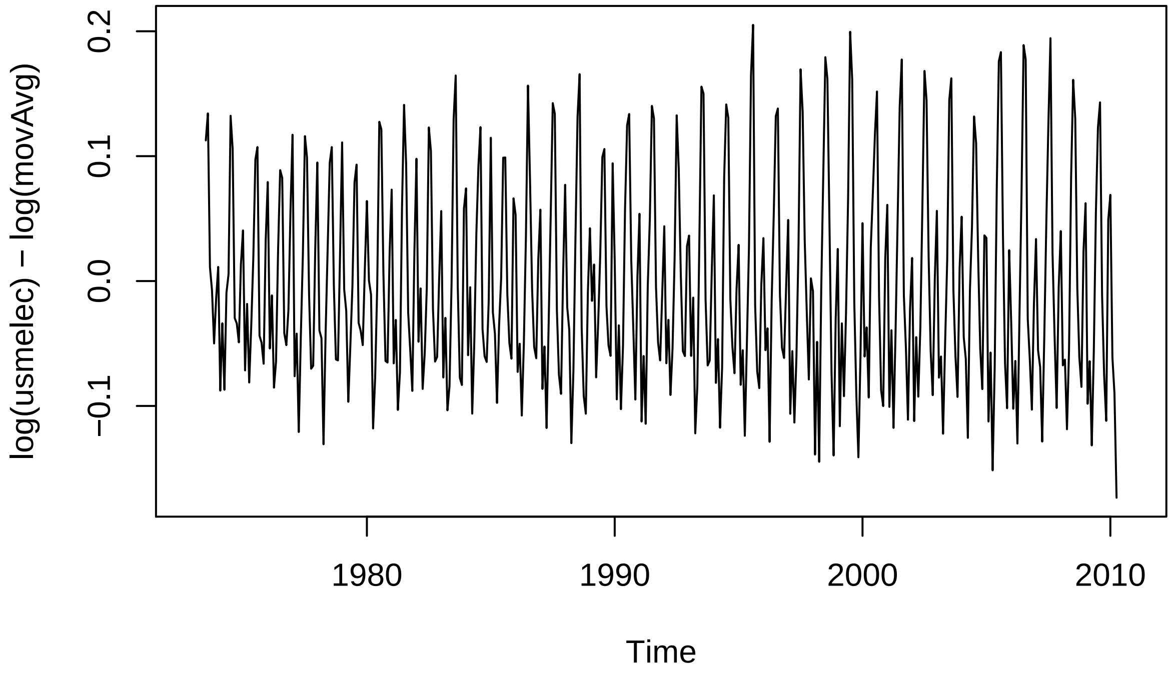
```
plot(netusage - BoxCox(movAvg, lambda=lambda))
```



It looks like the variance is not increasing.

The other option for this is using a log transform. Let's take a look at what this looks like when we subtract the moving avegage:

```
plot(log(usmelec) - log(movAvg))
```



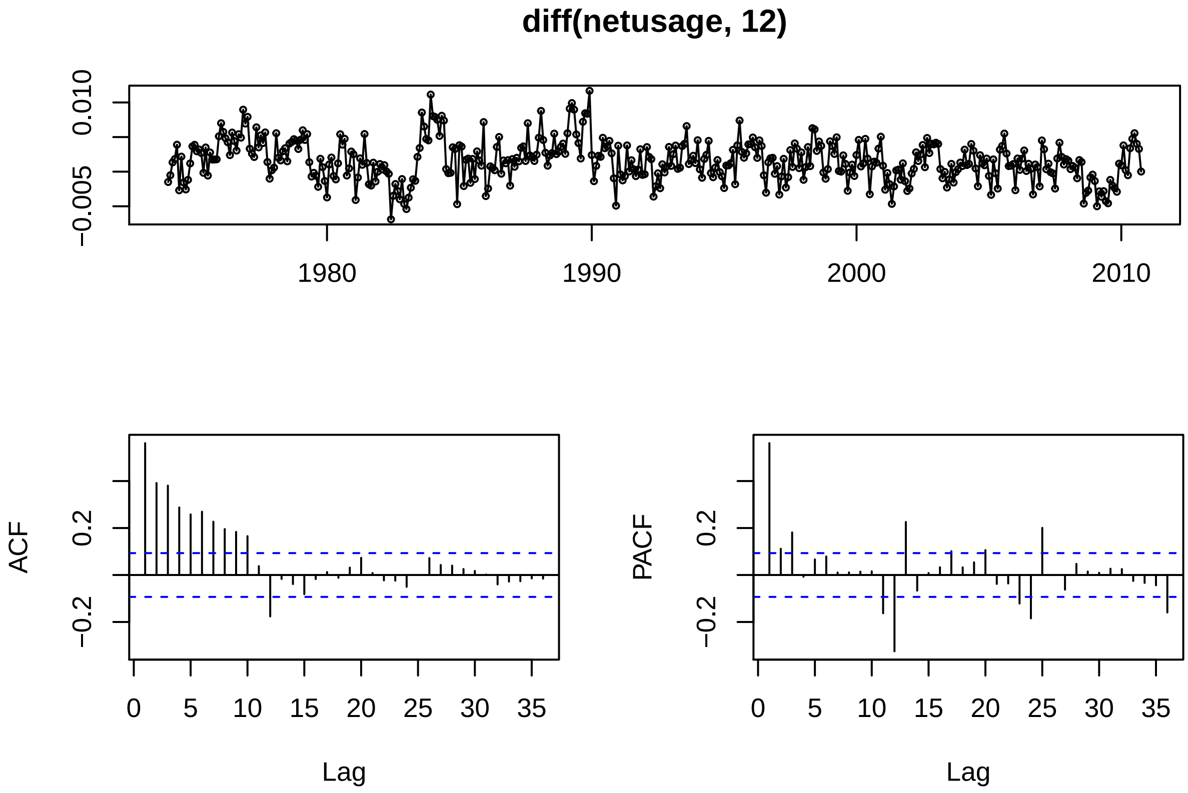
This still appears to have a trend in the variance, so we will go with the Box Cox transform.

8.6 (c)

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

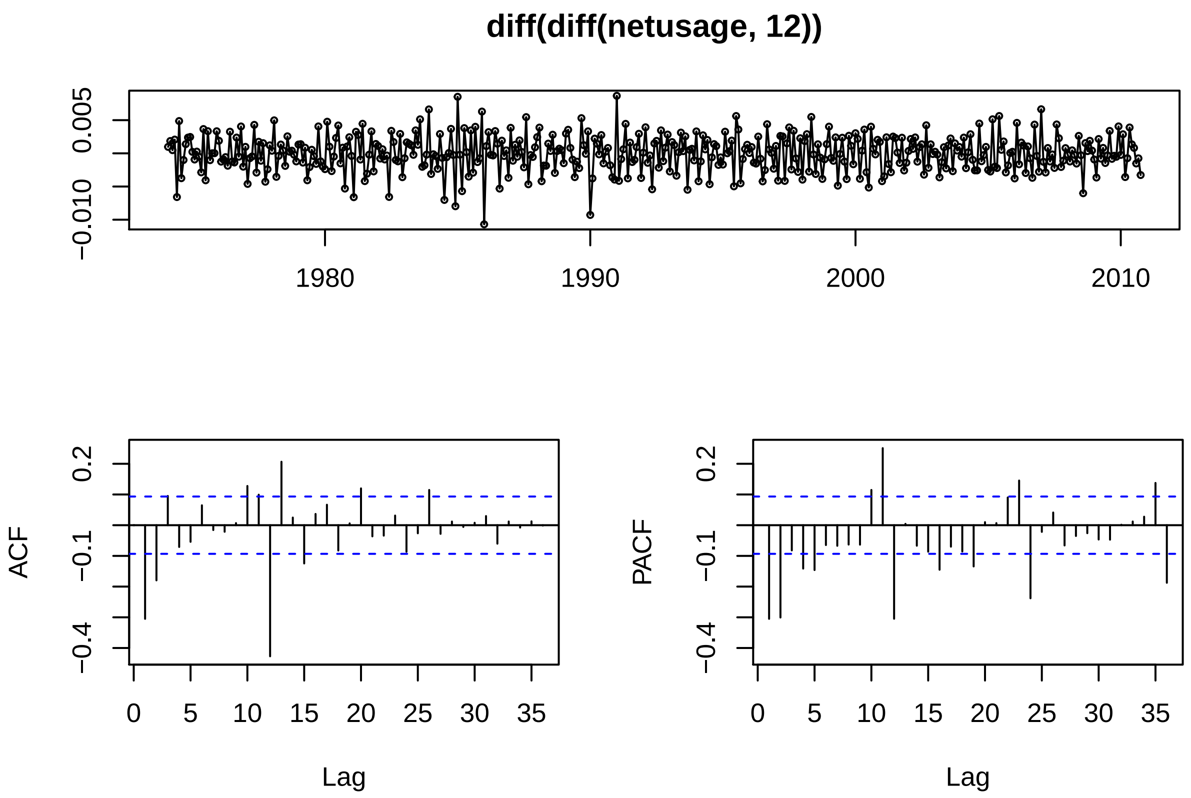
No the data are not stationary. Let's start with a seasonal difference:

```
tsdisplay(diff(netusage, 12))
```



The decaying trend in the ACF indicates a first order diff in necessary:

```
tsdisplay(diff(diff(netusage, 12)))
```



That looks better, but we still have some significant spikes. We can try our tests for stationary data now:

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

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

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

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

With a seasaonal-12 and second order diff, both of our unit root tests indicate a stationary series. I'm not sure what to make of the ACF and PACF plots. They still have some significant spikes.

8.6 (d)

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

```
test.arima <- function(t.series, params){
  order <- as.numeric(params[1:3])
  seasonal <- as.numeric(params[4:6])
  df <- data.frame(model=paste0("ARIMA(",
                                paste0(params, collapse=","),
                                ")"),
                   AICc=Arima(t.series,
                              order=order,
                              seasonal=seasonal,
                              lambda=lambda)$aicc)
  return(df)
}

gridSearch <- expand.grid(c(0, 1, 2), #p
                          c(2), #d
                          c(0, 1), #q
                          c(1, 2), #P
                          c(1), #D
                          c(0, 1)  #Q
                          )
df.list <- apply(gridSearch, MARGIN=1, FUN=function(x) {test.arima(usmelec, x)})

df <- do.call(rbind, df.list)

kable(df)
```

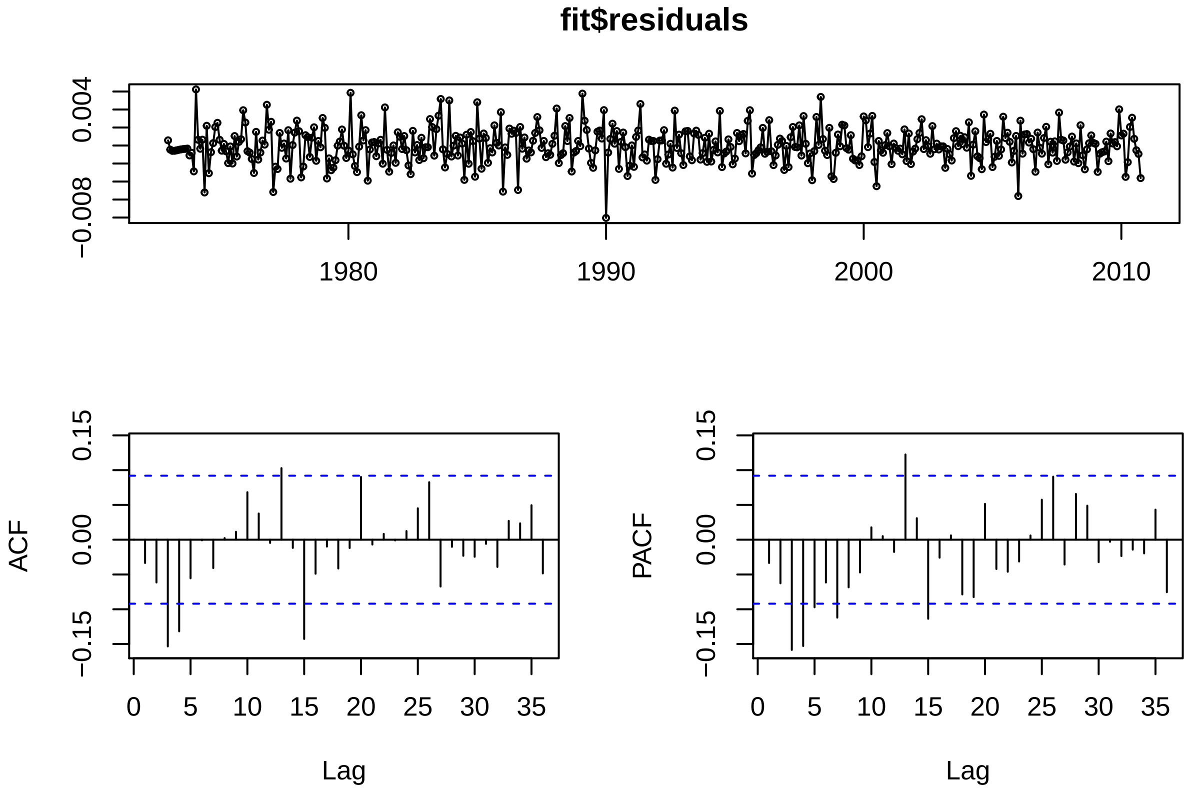
model	AICc
ARIMA(0,2,0,1,1,0)	-3607.482
ARIMA(1,2,0,1,1,0)	-3754.593
ARIMA(2,2,0,1,1,0)	-3876.260
ARIMA(0,2,1,1,1,0)	-4007.760
ARIMA(1,2,1,1,1,0)	-4041.012
ARIMA(2,2,1,1,1,0)	-4075.363
ARIMA(0,2,0,2,1,0)	-3657.899
ARIMA(1,2,0,2,1,0)	-3811.595
ARIMA(2,2,0,2,1,0)	-3918.264
ARIMA(0,2,1,2,1,0)	-4057.933
ARIMA(1,2,1,2,1,0)	-4091.944
ARIMA(2,2,1,2,1,0)	-4120.432
ARIMA(0,2,0,1,1,1)	-3750.161
ARIMA(1,2,0,1,1,1)	-3900.239
ARIMA(2,2,0,1,1,1)	-3996.371
ARIMA(0,2,1,1,1,1)	-4137.666
ARIMA(1,2,1,1,1,1)	-4165.397
ARIMA(2,2,1,1,1,1)	-4188.174
ARIMA(0,2,0,2,1,1)	-3752.429
ARIMA(1,2,0,2,1,1)	-3904.609
ARIMA(2,2,0,2,1,1)	-3996.740
ARIMA(0,2,1,2,1,1)	-4140.124
ARIMA(1,2,1,2,1,1)	-4168.028
ARIMA(2,2,1,2,1,1)	-4189.558

By AICc, the best model appears to be ARIMA(2, 2, 1)(2, 1, 1)

8.6 (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.

```r
fit <- Arima(usmelec, order=c(2, 2, 1),
             seasonal=c(2, 1, 1),
             lambda=lambda)

tsdisplay(fit$residuals)
```



```r
adf.test(fit$residuals)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  fit$residuals
## Dickey-Fuller = -10.743, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(fit$residuals)
```

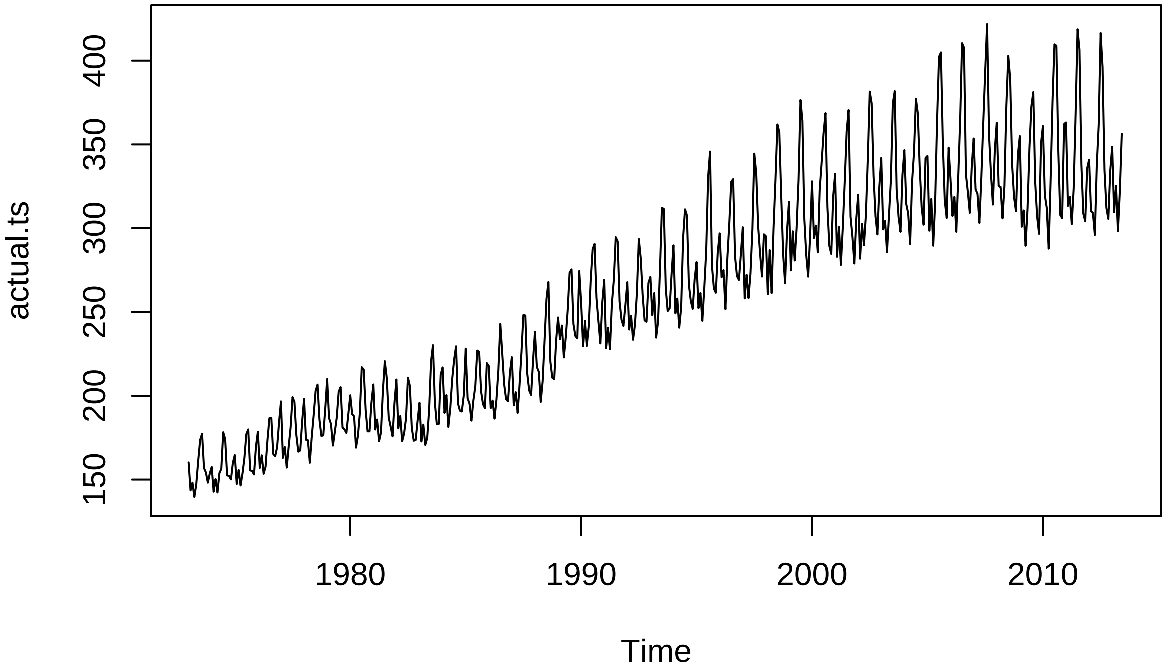
```
##
##  KPSS Test for Level Stationarity
##
## data:  fit$residuals
## KPSS Level = 0.033955, Truncation lag parameter = 4, p-value = 0.1
```

The graphs look pretty good. There are a few spikes in the ACF and PACF that I don't like but I'm not sure how many is too many. Both the unit root tests come back saying the series is stationary.

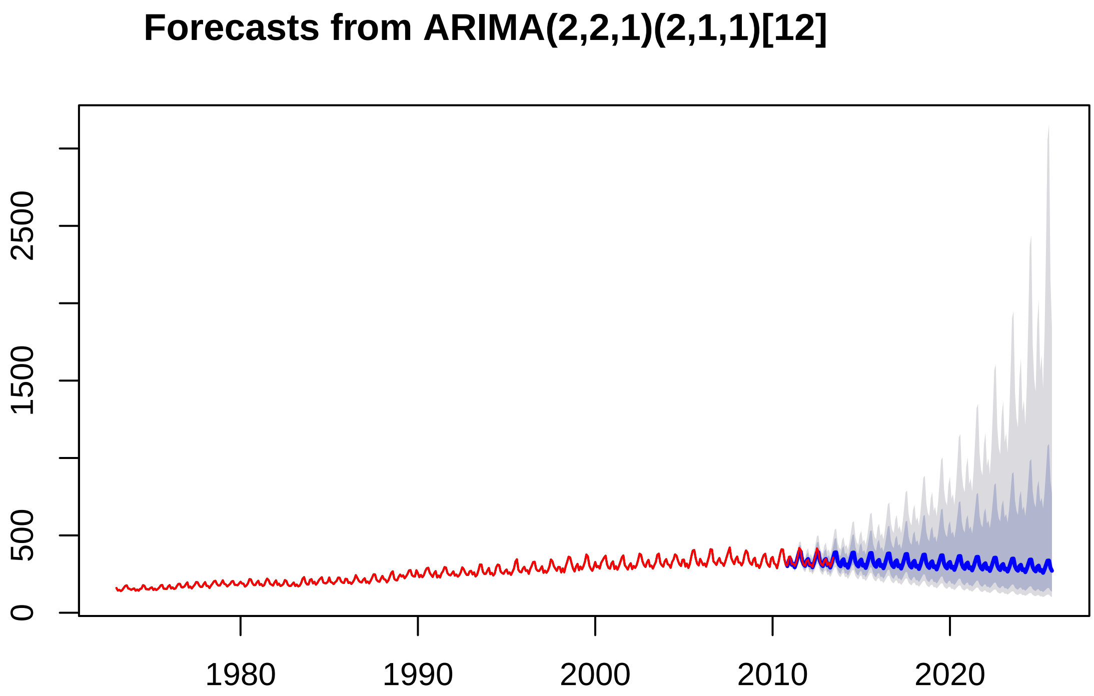
8.6 (f)

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

```
downloaded <- read.csv('electricity-overview.csv')
names(downloaded) <- c("month", "elec")

actual.ts <- ts(downloaded$elec, start=c(1973, 1), frequency = 12)
plot(actual.ts)
```



```
fcast <- forecast(fit, h=15*12)
plot(fcast)
lines(actual.ts, col='red')
```



That’s pretty impressive! To be honest, it worked better than I expected.

8.6 (g)

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

Judging from the confidence intervals, I would guess the next five years are usable, but much beyond that and things get really uncertain.