BHao\_HW5

## 8.1 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?

The differences between the three figures is the number of random values used, 36, 360 and 1000, respectively. Yes, all figures indicate the data are white noise because there are no significant spikes?

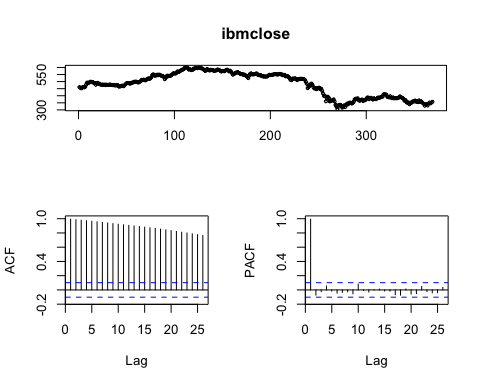
### 8.1.b. Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?

When there are more random values, the closer the critical values are to the mean of zero. I assume the reason is similar to why standard errors decrease with the number of observations. Each figure refers to white noise, the when the range of random values is wider, the lower the autocorrelations between the values.

## 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.

The plot of IBM's closing price shows a downward trend. The ACF plot shows significant autocorrelation between values and their lagged counterparts. The PACF plot shows the correlation between values and their lagged counterparts but with the effects of the in-between values removed. The downward trend and significant autocorrelation in the ACF chart indicate that the series is non-stationary and should be differenced.

tsdisplay(ibmclose)

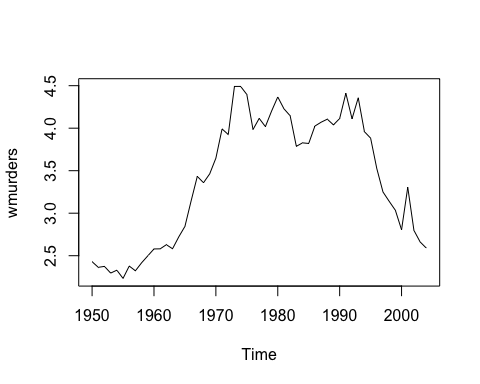


## 8.6 Consider the number of women murdered each year (per 100,000 standard population) in the United States (data set wmurders).

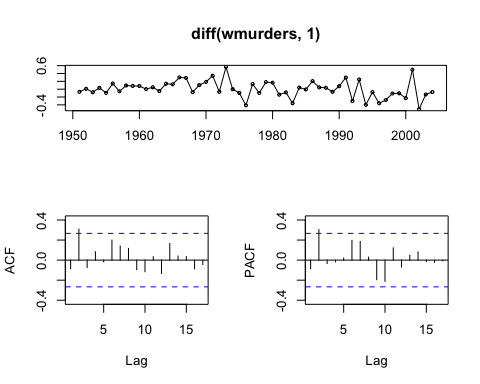
### 8.6.a. By studying appropriate graphs of the series in R, find an appropriate ARIMA(p,d,q) model for these data.

The data are clearly non-stationary as it trends up and then down over a long period of time; thus, we take the first difference which looks stationary.

plot(wmurders)



tsdisplay(diff(wmurders, 1))



The spike in lag2 of both ACF and PACF suggest evaluating ARIMA(2,1,0) and ARIMA(0,1,2) models along with a few similar variants. Based on the table below, the ARIMA(0,1,2) model produced the lowest AICc.

fit310 = Arima(wmurders, order=c(3,1,0))  
fit210 = Arima(wmurders, order=c(2,1,0))  
fit211 = Arima(wmurders, order=c(2,1,1))  
fit112 = Arima(wmurders, order=c(1,1,2))  
fit012 = Arima(wmurders, order=c(0,1,2))  
fit013 = Arima(wmurders, order=c(0,1,3))  
  
model = c('fit310', 'fit210', 'fit211', 'fit112', 'fit012', 'fit013')   
aicc = c(fit310$aicc, fit210$aicc, fit211$aicc, fit112$aicc, fit012$aicc, fit013$aicc)   
data.frame(model, aicc) %>% arrange(aicc)

## model aicc  
## 1 fit012 -12.94527  
## 2 fit210 -12.47872  
## 3 fit013 -10.63595  
## 4 fit112 -10.62525  
## 5 fit310 -10.17350  
## 6 fit211 -10.16966

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

No, when d >= 1 then c should be set to zero.

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

(1 - *B*)y = (1 + *B* + *B*^2)

where = -0.0660 and = 0.3712

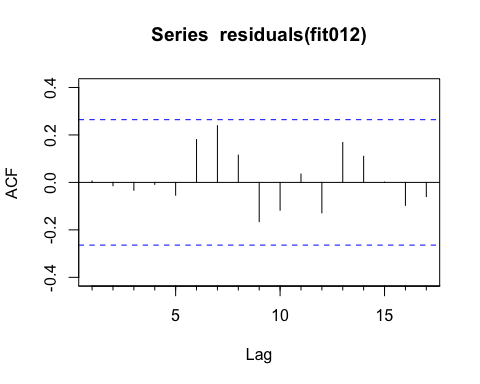
summary(fit012)

## Series: wmurders   
## ARIMA(0,1,2)   
##   
## Coefficients:  
## ma1 ma2  
## -0.0660 0.3712  
## s.e. 0.1263 0.1640  
##   
## sigma^2 estimated as 0.0422: log likelihood=9.71  
## AIC=-13.43 AICc=-12.95 BIC=-7.46  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE  
## Training set 0.0007242355 0.1997392 0.1543531 -0.08224024 4.434684  
## MASE ACF1  
## Training set 0.9491994 0.005880608

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

Yes, the residuals do look satisfactory.

Acf(residuals(fit012))



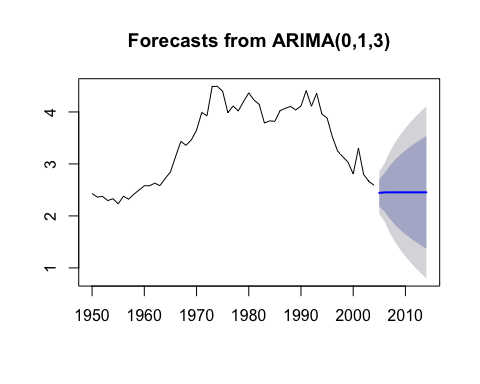
### 8.6.e. Forecast three times ahead. Check your forecasts by hand to make sure you know how they have been calculated.

forecast(fit012, h = 3)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2005 2.458450 2.195194 2.721707 2.055834 2.861066  
## 2006 2.477101 2.116875 2.837327 1.926183 3.028018  
## 2007 2.477101 1.979272 2.974929 1.715738 3.238464

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

plot(forecast(fit013))



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

If I set d=1, then auto.arima finds the same model; however, without that , auto.arima selects ARIMA(0,2,3) which has a higher AICc.

auto.arima(wmurders, d = 1, stepwise = F, approximation = F)

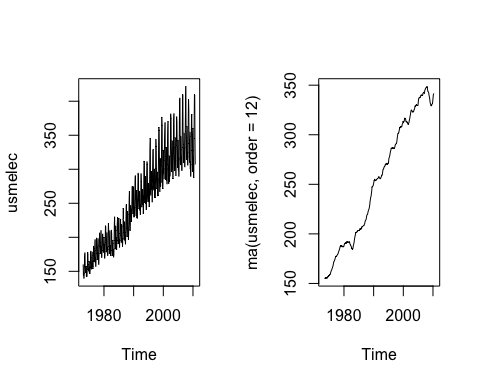
## Series: wmurders   
## ARIMA(0,1,2)   
##   
## Coefficients:  
## ma1 ma2  
## -0.0660 0.3712  
## s.e. 0.1263 0.1640  
##   
## sigma^2 estimated as 0.0422: log likelihood=9.71  
## AIC=-13.43 AICc=-12.95 BIC=-7.46

## 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.

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

Clearly, there is an upward trend in the data, and the variance seems to increase with time.

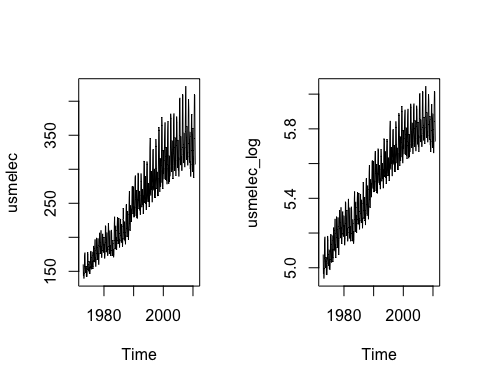
par(mfrow = c(1,2))  
plot(usmelec)  
plot(ma(usmelec, order = 12))



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

Given the increasing variance over time, taking the log will help stabilize the variance.

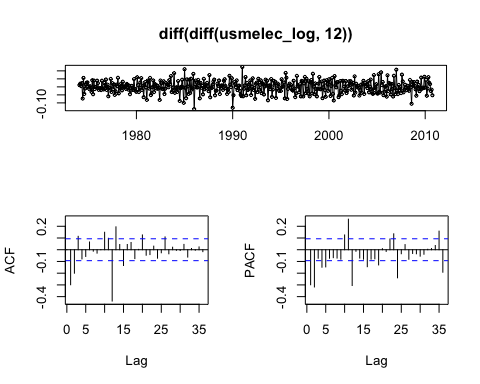
usmelec\_log = log(usmelec)  
par(mfrow=c(1,2))  
plot(usmelec)  
plot(usmelec\_log)



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

No, the data are not stationary. Given the seasonality, we first take the seasonal difference and then take an additional first difference.

tsdisplay(diff(diff(usmelec\_log, 12)))



### 8.8.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?

In the ACF chart, the significant spike at lag 1 suggessts a non-seasonal MA(1) component and the significant spike at lag 12 suggests a seasonal MA(1) component, so we start with an ARIMA(0,1,1)(0,1,1). We see similar spikes on the PACF chart, and thus can also try an ARIMA(1,1,0)(1,1,0) model. Of the models I tested, the ARIMA(1,1,1)(1,1,1) model produced the lowest AICc value.

However, that model did show a few spikes in its residual chart, so I took a look at a few more options, but it still produced the lowest AICc value.

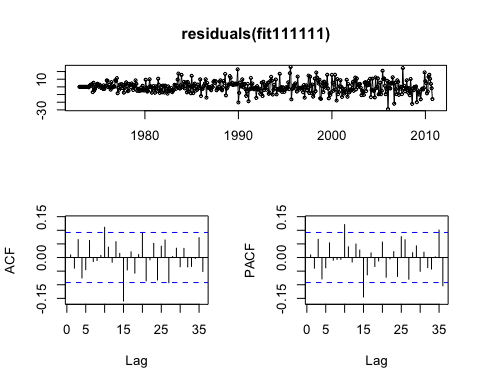
fit013011 = Arima(usmelec, order = c(0,1,3), seasonal = c(0,1,1))  
fit012011 = Arima(usmelec, order = c(0,1,2), seasonal = c(0,1,1))  
fit011011 = Arima(usmelec, order = c(0,1,1), seasonal = c(0,1,1))  
fit111111 = Arima(usmelec, order = c(1,1,1), seasonal = c(1,1,1))  
  
fit110310 = Arima(usmelec, order = c(1,1,0), seasonal = c(3,1,0))  
fit210210 = Arima(usmelec, order = c(2,1,0), seasonal = c(2,1,0))  
fit012012 = Arima(usmelec, order = c(0,1,2), seasonal = c(0,1,2))  
  
model2 = c('fit013011', 'fit012011', 'fit011011', 'fit111111', 'fit110310', 'fit210210', 'fit012012')   
aicc2 = c(fit013011$aicc, fit012011$aicc, fit011011$aicc, fit111111$aicc, fit110310$aicc, fit210210$aicc, fit012012$aicc)   
data.frame(model2, aicc2) %>% arrange(aicc2)

## model2 aicc2  
## 1 fit111111 3048.295  
## 2 fit012011 3048.427  
## 3 fit013011 3049.515  
## 4 fit012012 3050.158  
## 5 fit011011 3082.082  
## 6 fit110310 3113.753  
## 7 fit210210 3125.140

### 8.8.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.

Despite the few significant spikes in the residual charts, the ARIMA(1,1,1)(1,1,1) model produced the best AICc values of the models I tested. This model also produced a better AICc value than anything the auto.arima() function produced even with stepwise and approximation set to False.

tsdisplay(residuals(fit111111))



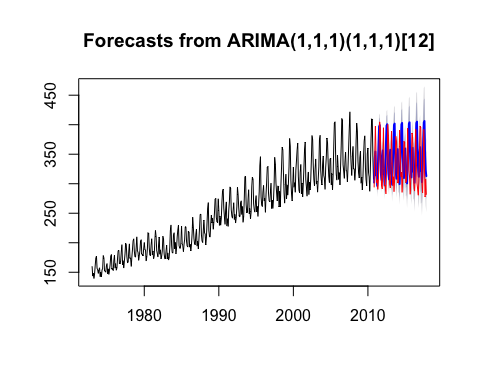
auto.arima(usmelec)

## Series: usmelec   
## ARIMA(1,0,2)(0,1,1)[12] with drift   
##   
## Coefficients:  
## ar1 ma1 ma2 sma1 drift  
## 0.9586 -0.4153 -0.2735 -0.7041 0.4268  
## s.e. 0.0210 0.0522 0.0525 0.0312 0.0676  
##   
## sigma^2 estimated as 56.23: log likelihood=-1519.25  
## AIC=3050.49 AICc=3050.68 BIC=3075.04

#auto.arima(usmelec, stepwise = F, approximation = F)

### 8.8.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.

forecast111111 = forecast(fit111111, h = 85) # to match actual data   
  
# format actual data to compare with forecast   
actual = read.csv('/Users/brucehao/Google Drive/CUNY/git/DATA624/MER\_T07\_01.csv', stringsAsFactors = FALSE)  
actual\_recent = actual[actual$YYYYMM >= 201011, ]  
actual\_recent$Value = as.numeric(actual\_recent$Value)  
# there appears to be problem in the data where every 13 months the value is 10x what it should be   
actual\_recent$Value = ifelse(actual\_recent$Value > 1000, actual\_recent$Value/10, actual\_recent$Value)  
actual\_ts = ts(actual\_recent$Value, start=c(2010, 11), end=c(2017, 11), frequency=12)  
  
plot(forecast111111)  
lines(actual\_ts, col=2)



# calculate RMSE of prediction   
rmse = function(error){ sqrt(mean(error^2)) }  
mae = function(error){ mean(abs(error)) }  
error = forecast111111$mean - actual\_ts  
rmse(error)

## [1] 49.6056

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

Usability depends on the use case, but at least over the ~7 year period for which actual data is available, the model's error does not seem to increase over time.