Time Series Analysis Module 1 Assignment

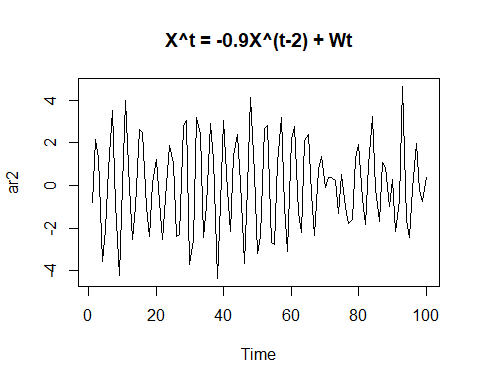
Filipp Krasovsky

Textbook Exercises (Pages 14, 35)

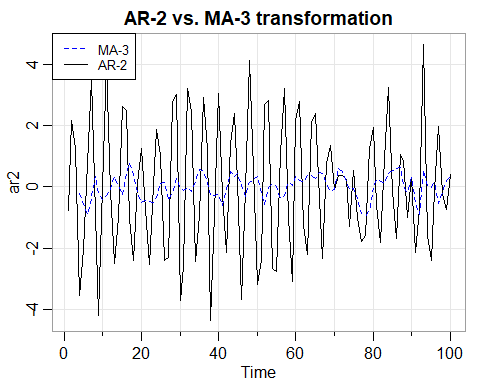
## Loading required package: astsa

1.1. [8 points] a) Generate n=100 observations from the autoregression xt = -.9xt-2 + wt

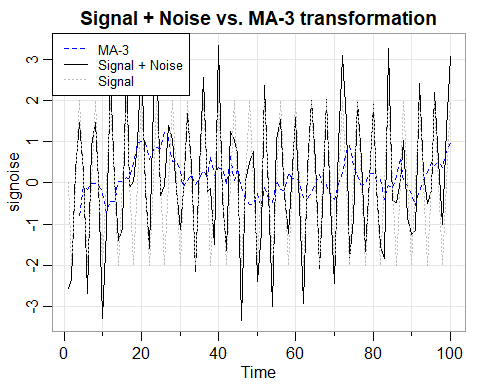
#create an AR-2 series with coefficients B1 = 0 and B2=-0.9,  
#creating xt = 0(xt-1) - 0.9(xt-2) + wt  
set.seed(1002)  
ar2 <- arima.sim(list(order = c(2,0,0), ar = c(0,-0.9)), n = 100)  
ts.plot(ar2,main="X^t = -0.9X^(t-2) + Wt")

 Apply a moving average filter (xt + xt-1 + xt-2 + xt-3) / 4 to the series.

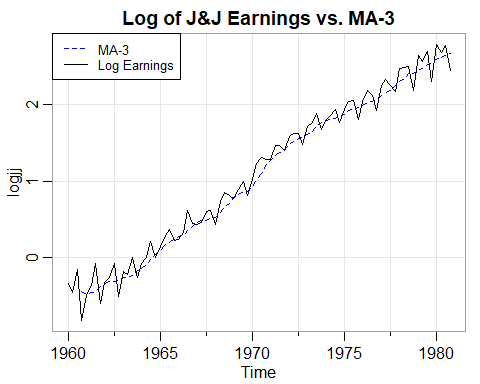
#use the filter command to create a moving average  
ma3 = filter(ar2,sides=1,filter=rep(1/4,4))  
#plot and superimpose  
tsplot(ar2,col="black",main="AR-2 vs. MA-3 transformation")  
lines(ma3,col="blue",lty=2)  
legend(x="topleft",legend=c("MA-3","AR-2"),col=c("blue","black"),cex=0.8,lty=2:1)

 Repeat with X^t = 2cos(2pi \* (t/4)) + wt.

#First, create our signal + noise model as well as an index for time.  
t = 1:100  
sig = 2 \* cos(2\*pi\*(t/4))  
noise = rnorm(100)  
signoise = sig + noise   
  
#use the filter command to create a moving average  
signoise\_ma3 = filter(signoise,sides=1,filter=rep(1/4,4))  
#plot and superimpose  
tsplot(signoise,col="black",main="Signal + Noise vs. MA-3 transformation")  
lines(signoise\_ma3,col="blue",lty=2)  
lines(sig,col="gray",lty=3)  
legend(x="topleft",legend=c("MA-3","Signal + Noise","Signal"),col=c("blue","black","gray"),cex=0.8,lty=c(2,1,3))

 Repeat with the log of the johnson and johnson data in example 1.2 from the textbook.

#long-form syntax to make origin clear.  
logjj = log(astsa::jj)  
#use the filter command to create a moving average  
jj\_ma3 = filter(logjj,sides=1,filter=rep(1/4,4))  
#plot and superimpose  
tsplot(logjj,col="black",main="Log of J&J Earnings vs. MA-3")  
lines(jj\_ma3,col="blue",lty=2)  
legend(x="topleft",legend=c("MA-3","Log Earnings"),col=c("blue","black"),cex=0.8,lty=c(2,1))

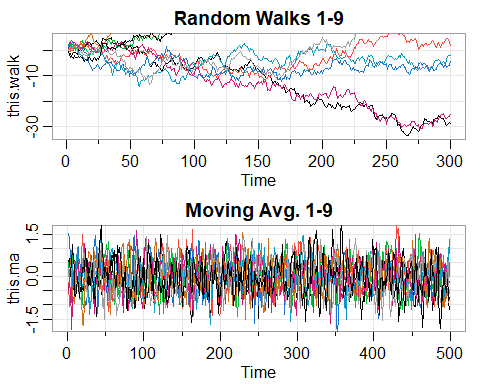
 What is seasonal adjustment? Seasonal adjusment is defined as a method of data smoothing for economic performance over a given period, the primary goal of which is to remove cyclical trends and provide a nonseasonal view.

Conclusions: I learned how to generate simulated time series, and as a corollary, how to identify when time series data is behaving in a way that can be modeled as either an AR or MA process. Furthermore, I learned that moving averages are able to smooth noisy data.

Exercise 1.3 - Working with Random Walk and Moving Average

Generate and plot nine series that are random walks of length n=300 without drift (theta=0) and variance = 1. Then, generate nine series of length n = 500 that are moving averages of the form discussed in example 1.8

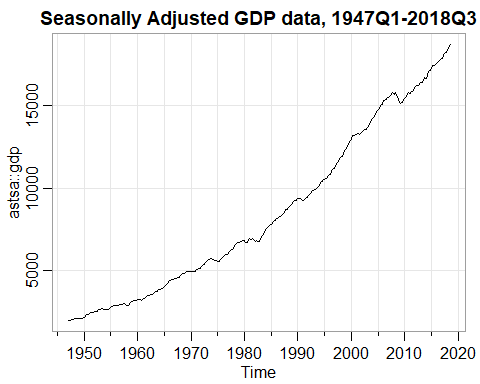
par(mfrow=c(2,1))  
for (i in c(1:9)){  
 this.walk = rnorm(300)  
 this.walk = cumsum(this.walk)  
   
 if(i==1){  
 tsplot(this.walk,main="Random Walks 1-9",col=i)  
 }else{  
 lines(this.walk,col=i)  
 }  
}  
  
for(i in c(1:9)){  
 w = rnorm(500)  
 this.ma = filter(w,sides=2,filter=rep(1/3,3))  
 if(i==1){  
 tsplot(this.ma,main="Moving Avg. 1-9",col=i)  
 }else{  
 lines(this.ma,col=i)  
 }  
}



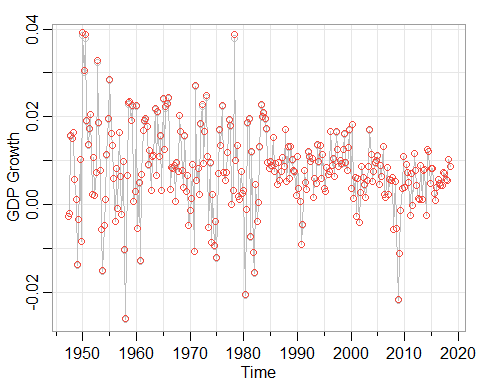
The primary differences between the two types of time series are that Random Walks are less stationary than moving averages. More importantly, the variance of a random walk increases without bound as time increases, whereas the variance of a moving average does not.

1.4. [7 points] The data in gdp are the seasonally adjusted quarterly U.S. GDP from 1947-I to 2018-III. The growth rate is shown in Figure 1.4. a) Plot the data and compare it to one of the models discussed in Section 1.3.

#we reference the gdp dataset from astsa::gdp   
#plot the data   
tsplot(astsa::gdp, main="Seasonally Adjusted GDP data, 1947Q1-2018Q3")

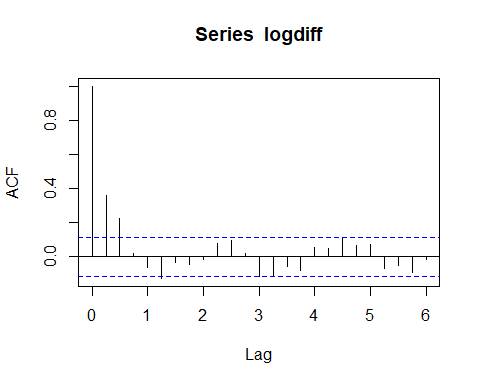
 b) Reproduce Figure 1.4 using your colors and plot characters (pch) of your own choice. Then, comment on the difference between the two methods of calculating growth rate.

logdiff = diff(log(gdp))  
returns = diff(gdp)/lag(gdp,-1)  
  
tsplot(logdiff,type="o",col="gray",ylab="GDP Growth")  
points(returns,pch=1,col=2)

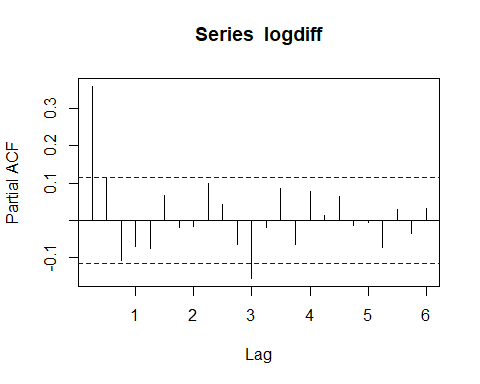
 The output itself is negligibly different - however, the process differs in that the first method uses a log transformation and takes successive differences while the other calculates the percentage difference by computing r = (xt - x(t-1))/x(t-1)

1. Which of the models discussed in Section 1.3 best describe the behavior of the growth in U.S. GDP?

#plot the ACF and PACF to identify if this is AR or MA.  
acf(logdiff)



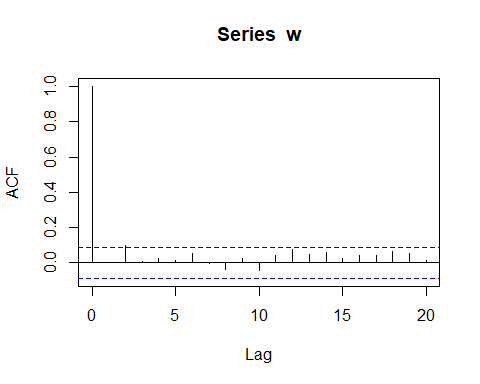
pacf(logdiff)

 Our correlograms show a gradually decreasing ACF and a PACF that cuts after one spike. As a result, we can make the argument this is an AR(1) model.

2.11. [7 points]

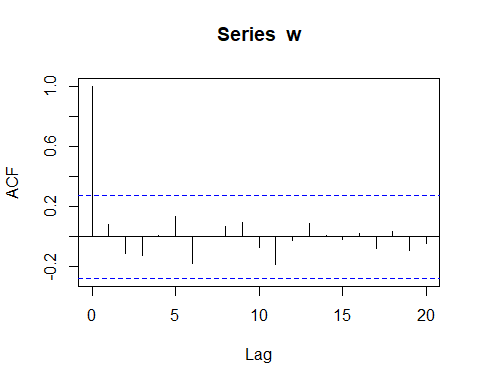
1. Simulate a series of n=500 moving average observations as in Example 1.8 and compute the sample ACF to lag 20. Compare the sample ACF you obtain to the actual ACF [Hint: Recall Example 2.17]

#reproduce an MA of n=500  
w = rnorm(500)  
v = filter(w,sides=2,filter=rep(1/3,3))  
  
#compute the sample ACF to lag twenty.  
#we can use the sample ACF formula:  
#sigma(1,n-h){(x^t+h - mean(x)) \* (x^t - mean(x))} / sigma(1,n){(x^t-mean(x))^2}  
#...or we can just use the library.  
  
acf(w,lag.max = 20)

 The actual ACF of an MA series would have gradually decreasing values as the absolute lag moves from 0 to 2, and then becomes zero after any lags > 2. In this instance, we only have a strong spike at zero.

1. Repeat part(a) using only n=50. How does changing n affect the results?

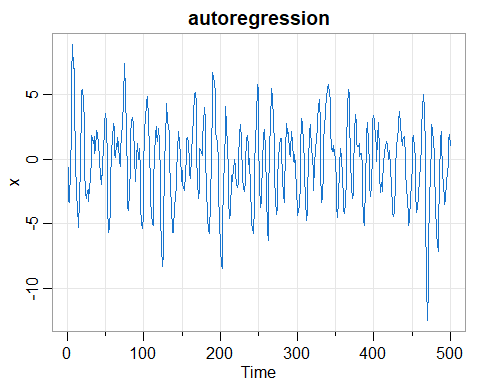
#reproduce an MA of n=50  
w = rnorm(50)  
v = filter(w,sides=2,filter=rep(1/3,3))  
  
#compute the sample ACF to lag twenty.  
#we can use the sample ACF formula:  
#sigma(1,n-h){(x^t+h - mean(x)) \* (x^t - mean(x))} / sigma(1,n){(x^t-mean(x))^2}  
#...or we can just use the library.  
  
acf(w,lag.max = 20)

 The key differences are rooted in (1) a higher standard for significance, (2) changing magnitudes for spikes, and (3) a change in directionality of the lag spikes. Making n smaller creates more variability in the ACF.

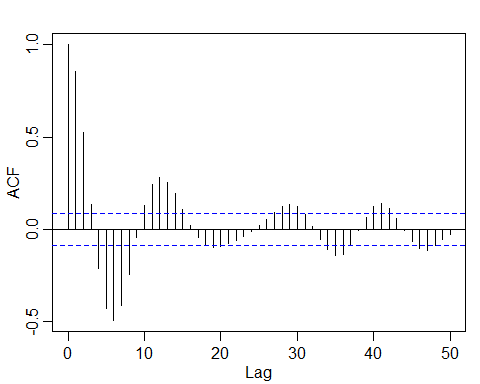
2.12. [7 points]

Simulate 500 observations from the AR model specified in Example 1.9 and then plot the sample ACF to lag 50. What does the sample ACF tell you about the approximate cyclic behavior of the data? [Hint: Recall Example 2.32]

set.seed(90210)   
w = rnorm(500 + 50)   
#50 extra to avoid startup problems   
x = filter(w, filter=c(1.5,-.75), method="recursive")[-(1:50)]   
tsplot(x, main="autoregression", col=4)

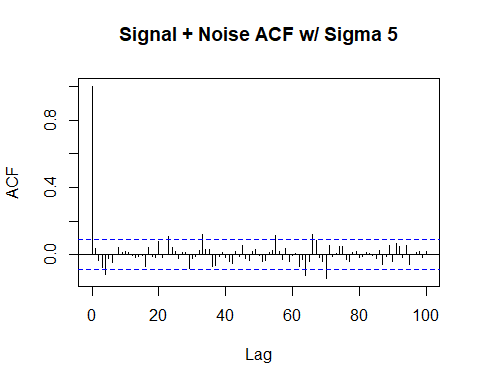
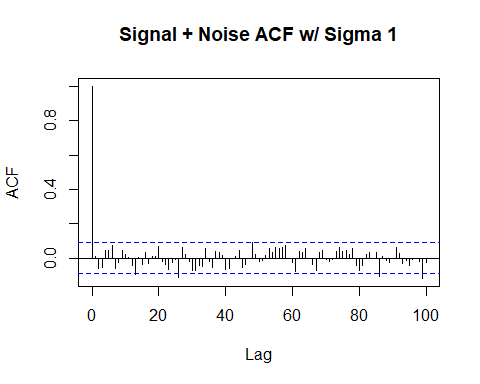
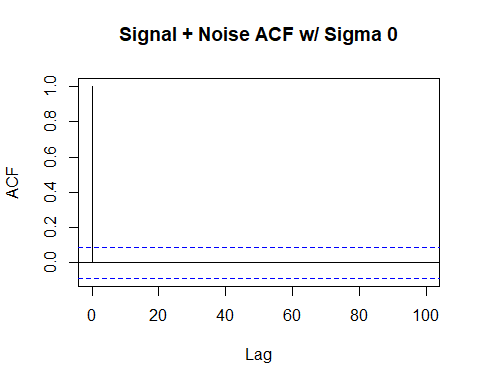


acf(x,lag.max = 50)

 Lags eventually decrease in siginificance after k = 30; however, we can confirm that observations 12 moments in time apart are positively correlated. Moments that are separated by around six lags tend to be negatively correlated. Positive values of the AR process tend to be associated with negative values 6 lags apart.

2.14. [7 points] Simulate a series of n=500 observations from the signal-plus-noise model presented in Example 1.11 with standard deviations of 0,1,and 5.Compute the sample ACF to lag 100 of the three series you generated and comment.

t=500  
for ( i in c(0,1,5)){  
 cs = 2\*cos(2\*pi\*(t+15)/50)  
 w = rnorm(n=500,mean=0,sd=i)  
 x = cs + w   
 x[500]=0.00001  
 #tsplot(sn,main=paste("Signal+Noise with Sigma:",i))  
 acf(x,main=paste("Signal + Noise ACF w/ Sigma",i),lag.max = 100)  
}

 Remarks: The first time series has no autocorrelative properties because the value is a constant with no variance. As the standard deviation becomes larger, the number of significant lag spikes increases as well - such as in the case of the final series, with potentially significance spikes at lag ~60 and ~90.