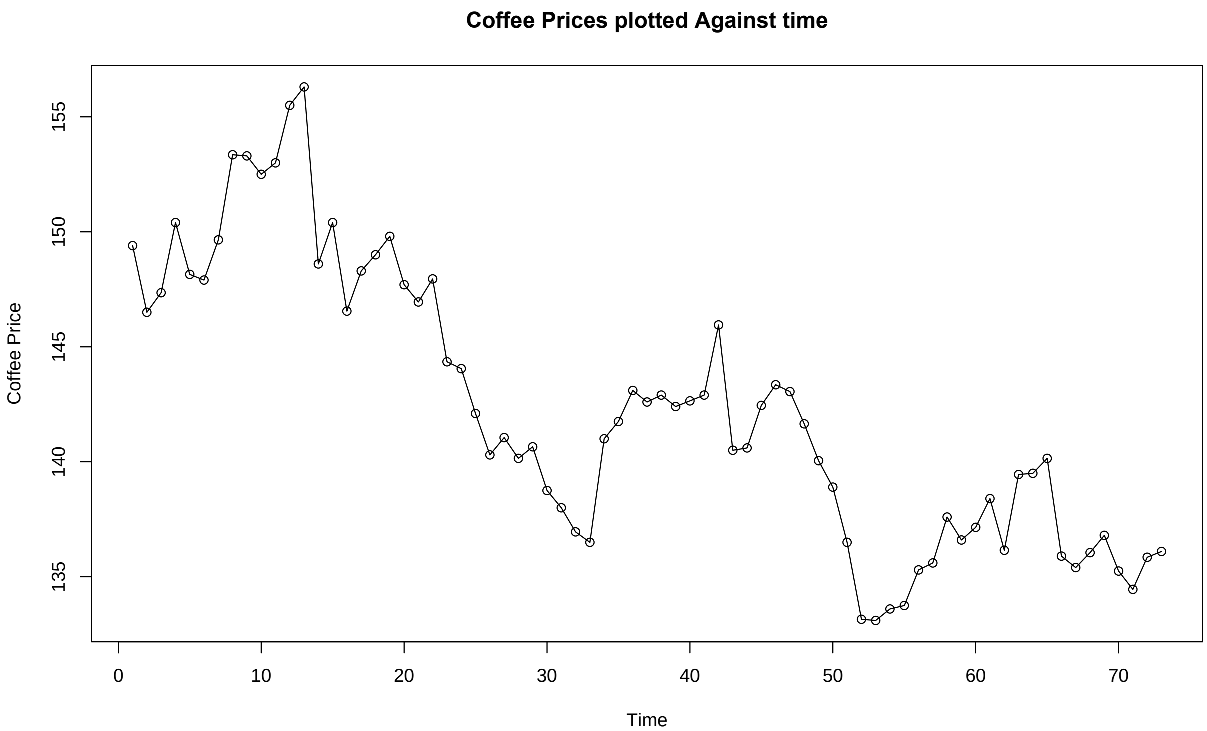
Marcus Crowder
Statistics-3155
R Homework Chapter 19

1. Make a time series plot (not scatterplot) of price against time. Use “Coffee Price” and “Time” as label for

y-axis and x-axis, respectively. Which time series components are evident from the plot?

Answer: The plot looks cyclical with prices going up but over time fall with a significant down trend having spikes on the way however this plot only uses data from January 2013 to April 2013 Making the forecast not a cyclical candidate since that component requires a time length greater than a year. Data does not appear to show seasonal effects

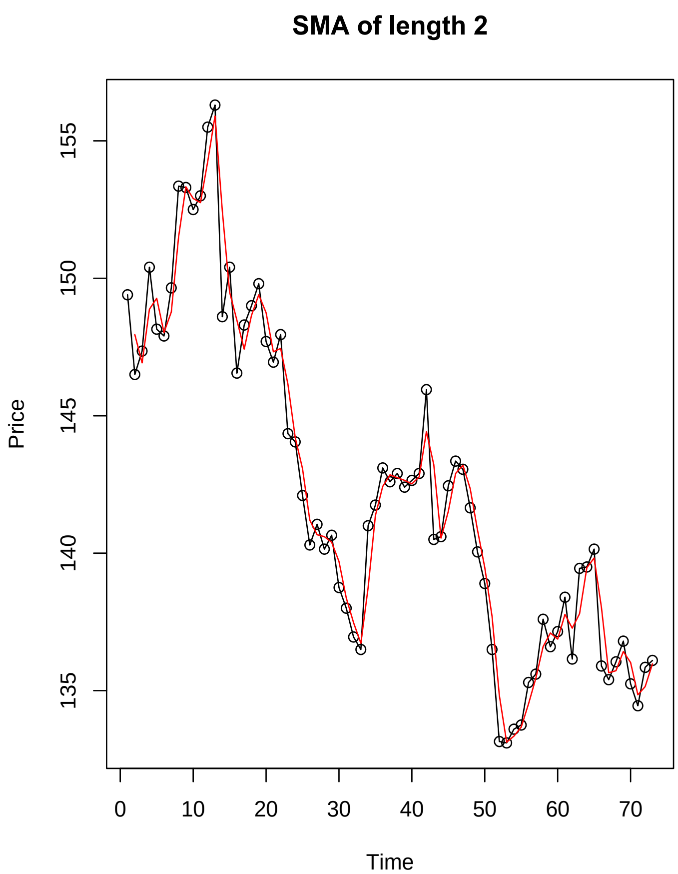


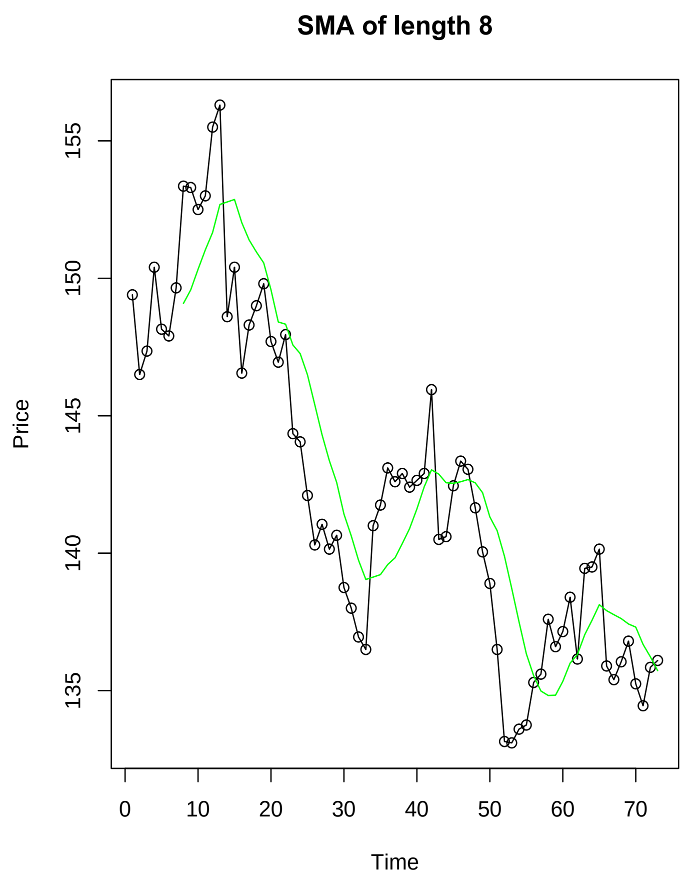
code:

```
> plot(coffee$time, coffee$price, xlab = 'Time', ylab = 'Coffee Price', main = 'Coffee Prices plotted Against time')
> lines(coffee$time, coffee$price)
```

2. Smooth the coffee price series using simple moving averages (SMA) of length 2 and 8. Add the two smoother curves (one in red and one in green) to the plot made in (a) and compare them.

The Lag of length 2 seems to be doing a better job of making predictions and following the trending lines while the Sma of length 8 smoothens out the fluctuations of the time series giving us a good description of the pattern.





Code:

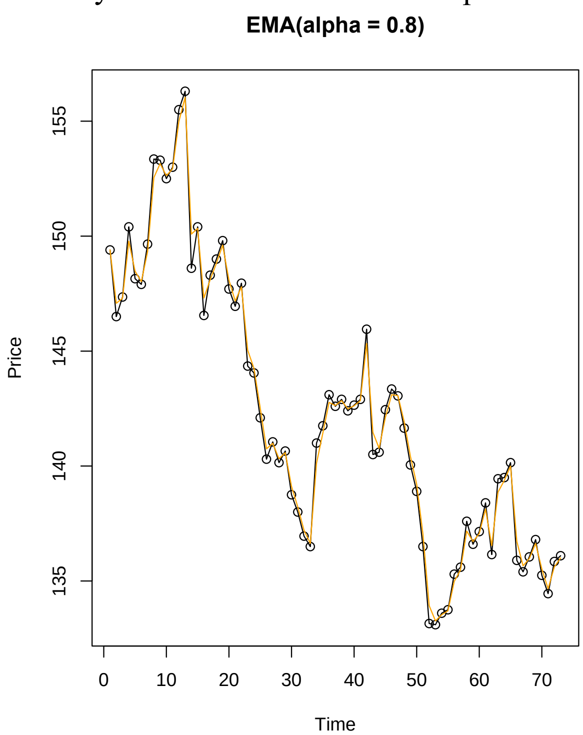
```
> library('TTR')

> sma1 <- SMA(coffee$price, n = 2)

> sma2 <- SMA(coffee$price, n = 8)

> plot(mfrow = c(1,2))

Error in xy.coords(x, y, xlabel, ylabel, log) :
```

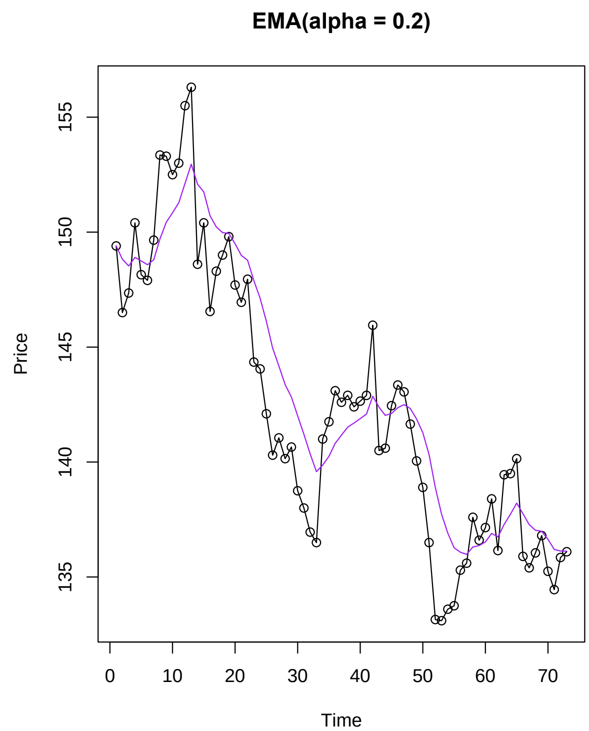
argument "x" is missing, with no default

```
> par(mfrow = c(1,2))

> plot(coffee$time, coffee$price, xlab = 'Time', ylab = 'Price', main = 'SMA of length 2')

> lines(coffee$time, coffee$price)

> lines(coffee$time, sma1, col = 'red')

> plot(coffee$time, coffee$price, xlab = 'Time', ylab = 'Price', main = 'SMA of length 8')

> lines(coffee$time, coffee$price)

> lines(coffee$time, sma2, col = 'green')
```

3. Apply single exponential smoothing (SES) to the coffee price series with weights α = 0.8 and α = 0.2, respectively. Add the two smoothed curves (one in orange and one in purple) to the plot made in (a) and compare them.

The EMA with a bigger alpha appears to move along the line and fluctuations but it is noticeably less smooth than the Time series plot with the smaller alpha of 0.2 which was to be expected given that a small alpha accounts for recent values less in the equation. The plot with alpha 0.2 is often incorrect and should receive a failing grade for responding too slowly to the trend leaving many over and underestimations present.





Code:
> par(mfrow = c(1,2))
> plot(coffee$time, coffee$price, xlab = 'Time', ylab = 'Price', main = 'EMA(alpha = 0.8)')
> lines(coffee$time, coffee$price)
> lines(coffee$time, ema1, col = 'orange')
> plot(coffee$time, coffee$price, xlab = 'Time', ylab = 'Price', main = 'EMA(alpha = 0.2)')
> lines(coffee$time, coffee$price)
> lines(coffee$time, ema2, col = 'purple')

4. Find autocorrelation between the original time series and each of the first 5 lags. Then, fit an autoregressive model with the lags whose autocorrelations are greater than 0.8. Write down the fitted model and add the smoothed curve in blue to the plot made in (a). Which lag does the model depend on most? Why?

Building the auto correlative Model with 5 lags we notice that as the lags become greater the less autocorrelation from the model which shows evidence of little to no seasonal effects.

Only the first 3 lags have autocorrelations above 8 so for the ar model we will only use order.max = 3 to represent those lags

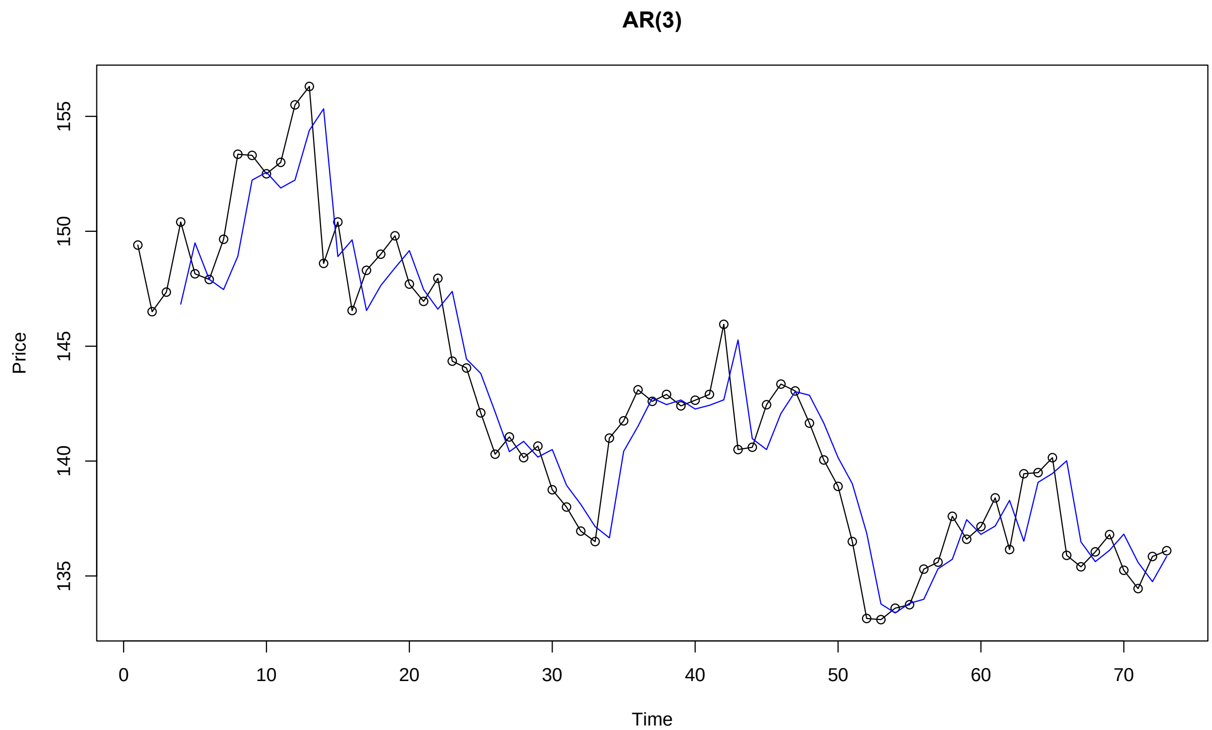
```
> acf(coffee$price, lag.max = 5, plot = FALSE)

Autocorrelations of series 'coffee$price', by lag

    0     1     2     3     4     5 
1.000 0.923 0.870 0.810 0.753 0.694 
> ar1 <- ar(coffee$price, aic = FALSE, order.max = 3, demean = FALSE, intercept = TRUE, method = 'ols')
> ar1

Call:
ar(x = coffee$price, aic = FALSE, order.max = 3, method = "ols",     demean = FALSE, intercept = TRUE)

Coefficients:
     1       2       3  
0.8456  0.1032  0.0004  

Intercept: 7.047 (5.952) 

Order selected 3  sigma^2 estimated as  4.009
```

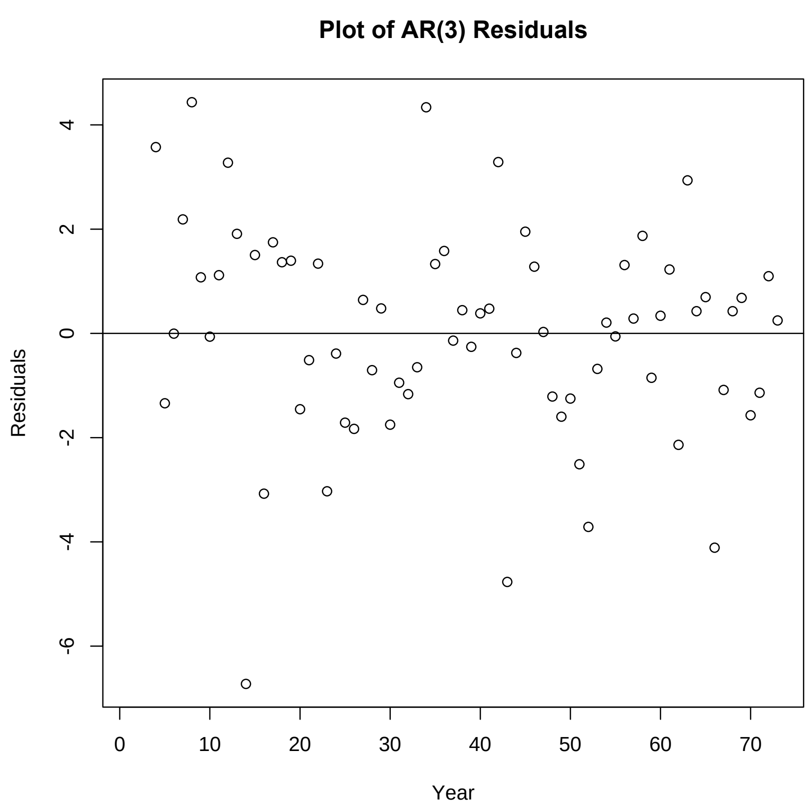
The fitted Model of this AR is

y^t = 7.047 + 0.8456yt-1 + 0.1032yt-2 + 0.0004yt-3

We can see that the model depends on lag 1 the most because as shown in previous example the moving average is predicted better in this time series when using a shorter lag value to account for the fluctuations that are no necessarily from any seasonal effects.

The plot for this Model seems to follow the data patterns but displays slowness in adjusting at the start but eventually predicts the model much better over time when following the downward trend with fluctuations.





Code for plots:

```
> plot(coffee$time, coffee$price, xlab= 'Time', ylab = 'Price', main = 'AR(3)')
> lines(coffee$time, coffee$price)
> lines(coffee$time, fitted.ar1, col = 'blue')

> plot(coffee$time, ar1$resid, xlab = 'Year', ylab = 'Residuals', main = 'Plot of AR(3) Residuals')
> abline(0, 0)
```

5. Suppose we know that the next value in the series was, in fact, 138.90. Compute the corresponding *absolute percentage error* (APE) for each of the models you have fitted before. Which model gives us the best prediction?

```
> (yhat.sma1 <- sma1[length(sma1)])
[1] 135.975
> (yhat.sma2 <- sma2[length(sma2)])
[1] 135.725
> (yhat.ema1 <- ema1[length(ema1)])
[1] 136.0026
> (yhat.ema2 <- ema2[length(ema2)])
[1] 136.1255
> (yhat.ar1 <- predict(ar1, n.ahead = 1, se.fit = FALSE))
Time Series:
Start = 74
End = 74
Frequency = 1
[1] 136.2088
> y.true <- 138.90
> abs(y.true - yhat.sma1)/abs(y.true)*100
[1] 2.105838
> abs(y.true - yhat.sma2)/abs(y.true)*100
[1] 2.28582
> abs(y.true - yhat.ema1)/abs(y.true)*100
[1] 2.085957
> abs(y.true - yhat.ema2)/abs(y.true)*100
[1] 1.997467
> abs(y.true - yhat.ar1)/abs(y.true)*100
Time Series:
Start = 74
End = 74
Frequency = 1
[1] 1.937486
>
```

The AR(3) Model seems to offer the best prediction, because it yields the smallest APE (1.937486)