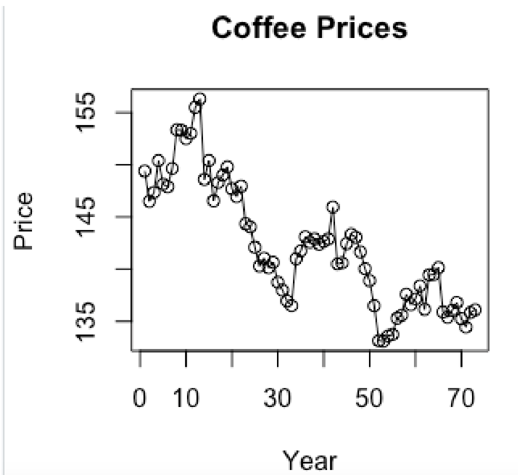
1: Make a time series plot (not scatterplot) of price against time. Use “Coffee Price” and “Time” as label for y-axis and x-axis, respectively. Which time series components are evident from the plot?

Answer: There appears to be an overall negative trend with two spikes in years 10, 40 and around 60, while there's a sharp decline from year 10 to 30, and from year 40 to 50. There also appears to be a seasonal component because the series follows the downward trend closely.

R-code:

```
plot(coffee$time, coffee$price, xlab = 'Year', ylab = 'Price',
    main = 'Coffee Prices')
lines(coffee$time, coffee$price)
```

R-output:



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

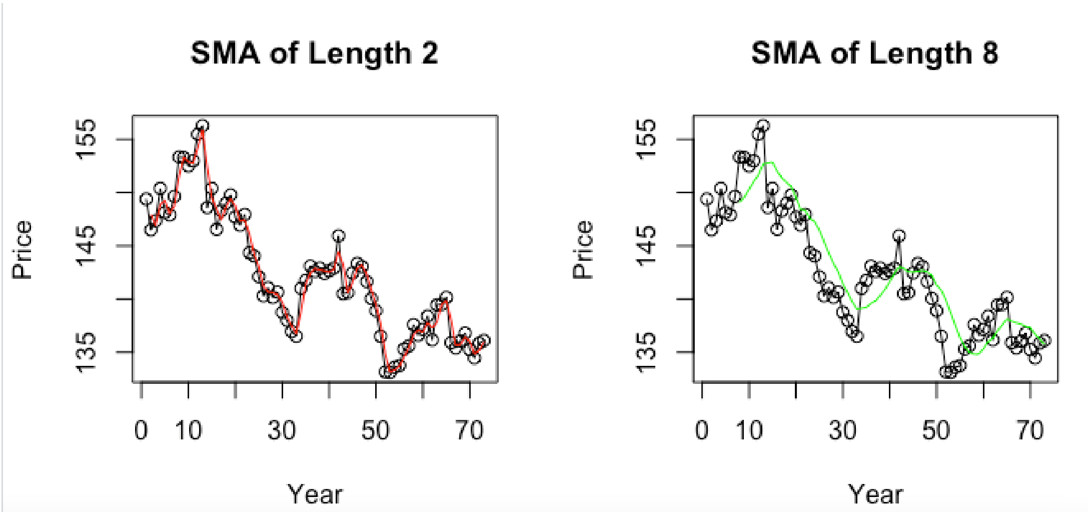
Answer: The SMA of length 8 demonstrates a smoother series, which smooths out the sharp ups and downs of the seasonal effects. Compared to the SMA of length 2, it captures the rapid changes of the series more closely and accurately.

R-code:

```
library(TTR)
sma1 = SMA(coffee$price, n = 2)
sma2 = SMA(coffee$price, n = 8)

par(mfrow = c(1, 2))
plot(coffee$time, coffee$price, xlab = 'Year', ylab = 'Price', main = 'SMA of Length 2')
lines(coffee$time, coffee$price)
```

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

plot(coffee$time, coffee$price, xlab = 'Year', ylab = 'Price', main = 'SMA of Length 8')
lines(coffee$time, coffee$price)
lines(coffee$time, sma2, col = 'green')
```

<u>R-output:</u>



<u>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.</u>

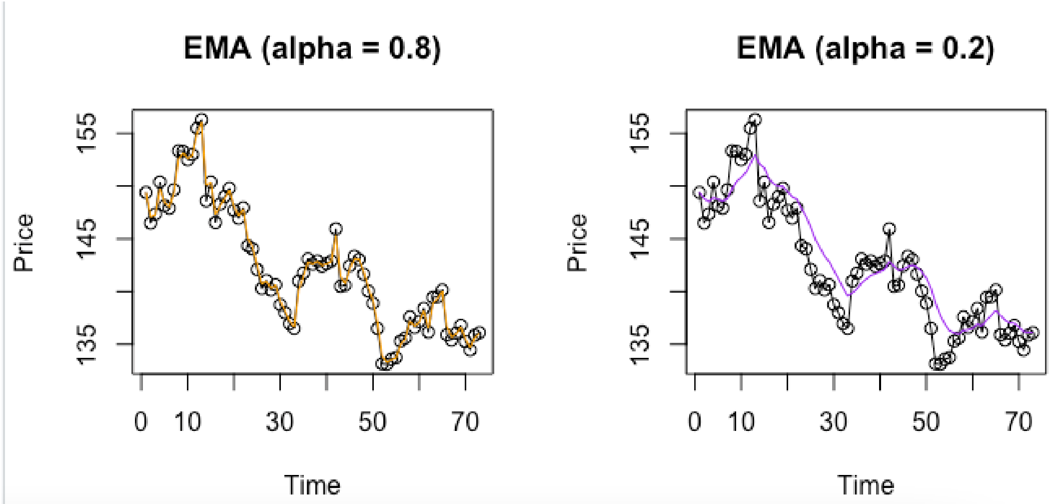
<u>Answer:</u>
The EMA with a larger alpha value closely follows the series, making it less smooth than the EMA with a smaller alpha. The EMA with alpha=0.2 is more smoother than the EMA with alpha=0.8. EMA with alpha=0.8 captures changes in the series more better, and the EMA with alpha=0.2 is too smooth to capture seasonal effects, and doesn't capture the immediate changes in series as well.

<u>R-code:</u>

```
ema1 = EMA(coffee$price, ratio = 0.8, n = 1)
ema2 = EMA(coffee$price, ratio = 0.2, n = 1)

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')
```

R-output:



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?

Answer:

Fitted model: y hat t = 7.801 + 0.8601(yt-1)+0.1180(yt-2)-0.0995(yt-3)+0.1573(yt-4)-0.0921(yt-5)

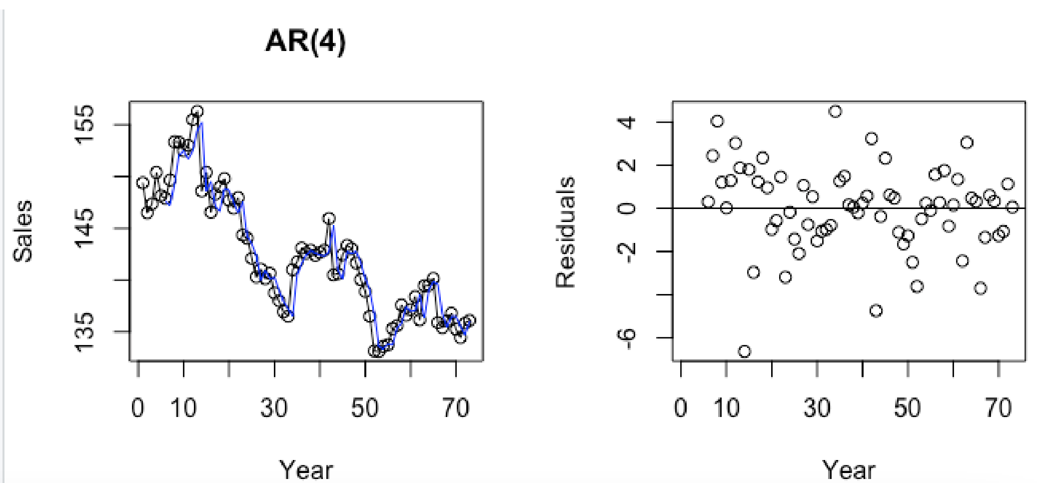
The model depends on the 1st lag the most because it has the largest coefficient of 0.8601.

R-code:

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

ar1 = ar(coffee$price, aic = FALSE, order.max = 5, demean = FALSE,
      intercept = TRUE, method = 'ols')
ar1

fitted.ar1 = coffee$price - ar1$resid
par(mfrow = c(1, 2))
plot(coffee$time, coffee$price, xlab = 'Year', ylab = 'Sales', main = 'AR(4)')
lines(coffee$time, coffee$price)
lines(coffee$time, fitted.ar1, col = 'blue')
plot(coffee$time, ar1$resid, xlab = 'Year', ylab = 'Residuals') #residuals
abline(0, 0)
```

R-output:

```
> 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 = 5, demean = FALSE,
+          intercept = TRUE, method = 'ols')
> ar1

Call:
ar(x = coffee$price, aic = FALSE, order.max = 5, method = "ols",     demean = FALSE, intercept
 = TRUE)

Coefficients:
      1        2        3        4        5
 0.8601   0.1180  -0.0995   0.1573  -0.0921

Intercept: 7.801 (6.083)

Order selected 5  sigma^2 estimated as  3.842
```



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?

Answer:
Autoregressive AR(4) is the best model because it has the smallest APE value of 1.95%.

R-code:
(yhat.sma1 = sma1[length(sma1)])
(yhat.sma2 = sma2[length(sma2)])
(yhat.ema1 = ema1[length(ema1)])

```
(yhat.ema2 = ema2[length(ema2)])
(yhat.ar1 = predict(ar1, n.ahead = 1, se.fit = FALSE))
y.true = 138.9
abs(y.true - yhat.sma1)/abs(y.true)*100
abs(y.true - yhat.sma2)/abs(y.true)*100
abs(y.true - yhat.ema1)/abs(y.true)*100
abs(y.true - yhat.ema2)/abs(y.true)*100
abs(y.true - yhat.ar1)/abs(y.true)*100
```

R-output:

```
> (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] 136.1884
> y.true = 138.9
> 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] 1.95219
```

Appendix:

```
plot(coffee$time, coffee$price, xlab = 'Year', ylab = 'Price',
   main = 'Coffee Prices')
lines(coffee$time, coffee$price)

library(TTR)
sma1 = SMA(coffee$price, n = 2)
sma2 = SMA(coffee$price, n = 8)
```

```r
par(mfrow = c(1, 2))
plot(coffee$time, coffee$price, xlab = 'Year', ylab = 'Price', main = 'SMA of Length 2')
lines(coffee$time, coffee$price)
lines(coffee$time, sma1, col = 'red')

plot(coffee$time, coffee$price, xlab = 'Year', ylab = 'Price', main = 'SMA of Length 8')
lines(coffee$time, coffee$price)
lines(coffee$time, sma2, col = 'green')

ema1 = EMA(coffee$price, ratio = 0.8, n = 1)
ema2 = EMA(coffee$price, ratio = 0.2, n = 1)

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')

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

ar1 = ar(coffee$price, aic = FALSE, order.max = 5, demean = FALSE,
        intercept = TRUE, method = 'ols')
ar1

fitted.ar1 = coffee$price - ar1$resid
par(mfrow = c(1, 2))
plot(coffee$time, coffee$price, xlab = 'Year', ylab = 'Sales', main = 'AR(4)')
lines(coffee$time, coffee$price)
lines(coffee$time, fitted.ar1, col = 'blue')
plot(coffee$time, ar1$resid, xlab = 'Year', ylab = 'Residuals')
abline(0, 0)

(yhat.sma1 = sma1[length(sma1)])
(yhat.sma2 = sma2[length(sma2)])
(yhat.ema1 = ema1[length(ema1)])
(yhat.ema2 = ema2[length(ema2)])

(yhat.ar1 = predict(ar1, n.ahead = 1, se.fit = FALSE))

y.true = 138.9
abs(y.true - yhat.sma1)/abs(y.true)*100
```

```
abs(y.true - yhat.sma2)/abs(y.true)*100
abs(y.true - yhat.ema1)/abs(y.true)*100
abs(y.true - yhat.ema2)/abs(y.true)*100
abs(y.true - yhat.ar1)/abs(y.true)*100
```