HW 3 (Includes my explanation using # and code)

Question 7.1

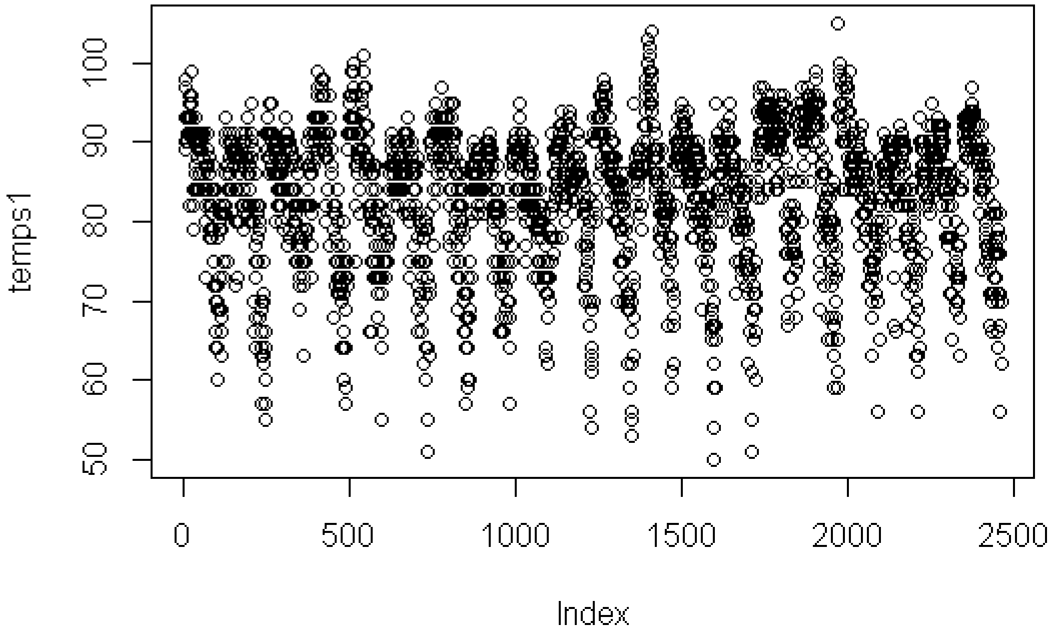
Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. What data would you need? Would you expect the value of α (the first smoothing parameter) to be closer to 0 or 1, and why?

Exponential smoothing could be used for measuring demand for a product like soda in a particular month. It is possible that the demand for soda changes per seasons like for example in the summer season there could be an increased demand for it due to the hot weather. Past data (ex: past 20 years) will be required to come up with a prediction. I would expect a to be closer to 0 because the demand will likely be dependent on the seasons or events like Super Bowl which will cause a lot of randomness in the system.

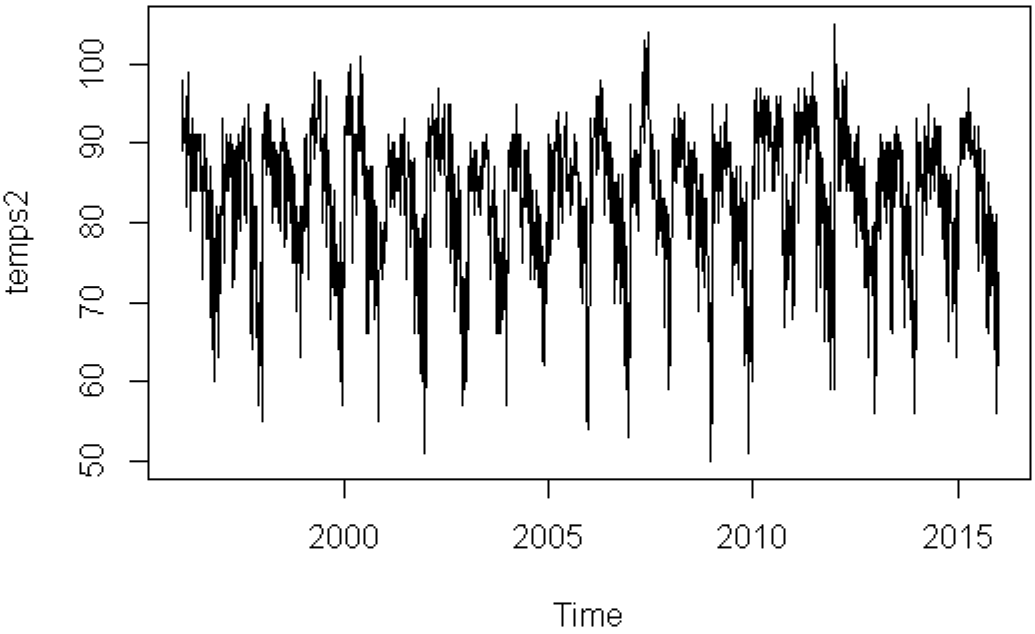
Question 7.2

Using the 20 years of daily high temperature data for Atlanta (July through October) from Question 6.2 (file `temps.txt`), build and use an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years. (Part of the point of this assignment is for you to think about how you might use exponential smoothing to answer this question. Feel free to combine it with other models if you'd like to. There's certainly more than one reasonable approach.)

```
#read table and show the head of the table
temps <- read.table("7.2tempsSummer2018.txt", stringsAsFactors = FALSE, header = TRUE)
head(temps)

#Convert vector to time series.  The below plot is a little hard to understand, hence I will use the time series function to plot the graph again.
temps1 <- as.vector(unlist(temps[,2:21]))
temps1
plot(temps1)
```

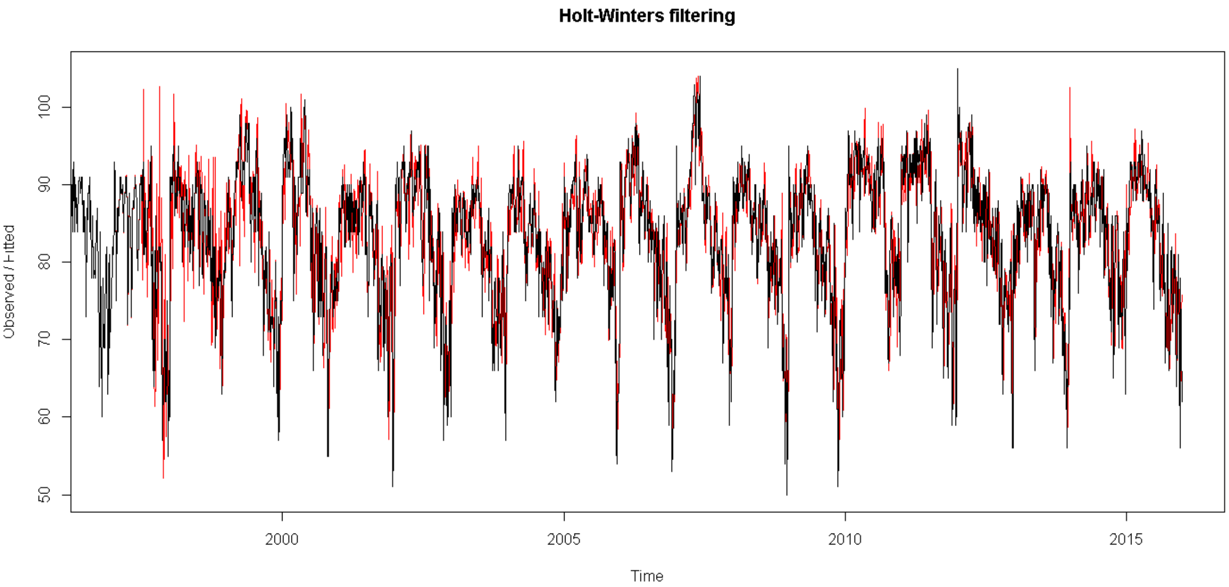


```
#Frequency of 123 is the number of days from July to October.
temps2 <- ts(temps1, start = 1996, frequency = 123)
temps2
plot(temps2)
```



#I will use triple exponential smoothing. The function will optimize alpha, beta, and gamma in order to minimize the deviation from the prediction from the true observations.

```
temps3 <- HoltWinters(temps2, alpha = NULL, beta = NULL, gamma = NULL, seasonal = "multiplicative")
temps3
plot(temp3)
#The red line is the smooth data. There is no smooth data for the first year since the seasonality coefficient is one.
```



```
summary(temps3)
# These are the optimized values of alpha: 0.615, beta : 0, gamma: 0.55. a  = 73.67952 (which is current estimate of baseline coefficient), b   =  -0.00436 (current estimate of trend coefficient).

#xhat is the smooth term. Since the trend term is close to zero there is not significant change.
head(temps3$fitted)
```

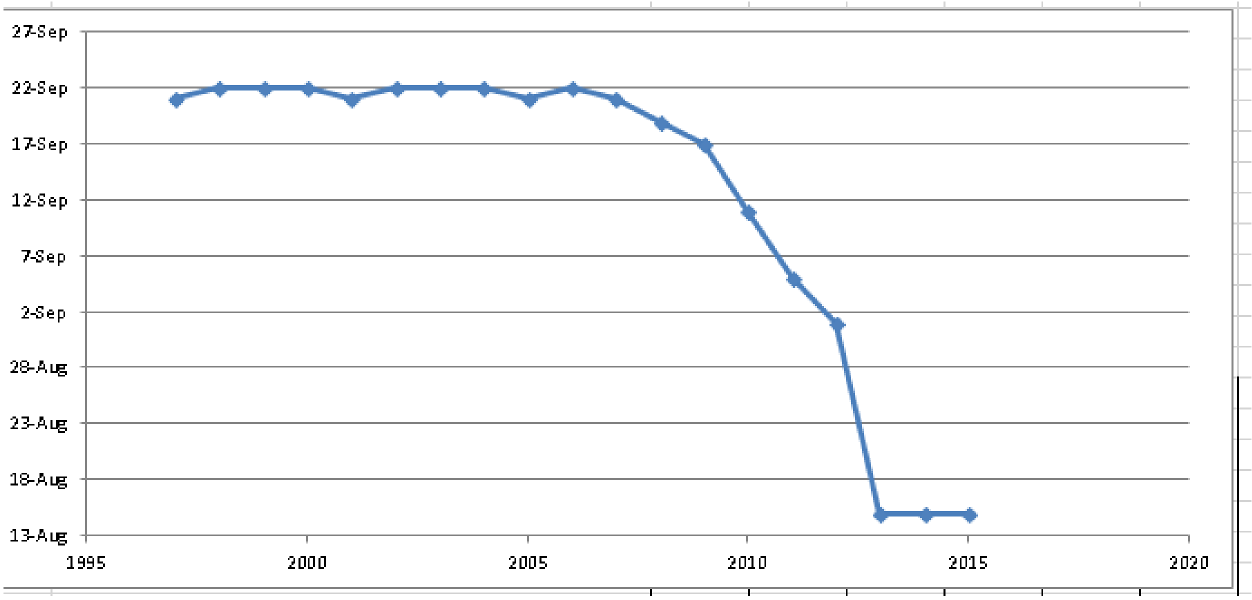
```
     xhat level    trend season
[1,] 87.2  82.9 -0.00436   1.05
[2,] 90.4  82.2 -0.00436   1.10
[3,] 93.0  81.9 -0.00436   1.14
[4,] 90.9  81.9 -0.00436   1.11
[5,] 84.0  81.9 -0.00436   1.03
[6,] 84.0  81.9 -0.00436   1.03
```

```
tail(temps3$fitted)
```

```
        xhat level    trend season
[2332,] 76.5  87.8 -0.00436  0.872
[2333,] 69.7  81.1 -0.00436  0.860
[2334,] 57.0  71.3 -0.00436  0.800
[2335,] 72.1  87.4 -0.00436  0.826
[2336,] 73.9  85.8 -0.00436  0.862
[2337,] 75.8  83.0 -0.00436  0.914
```

```
#The trend estimate seems to be closer to  zero, suggesting that the data don't show significant increase or decreases over the 20-year period.

#I would like to verify this using excel spreadsheet.
temps4 <- matrix(temps3$fitted[,4], nrow=123)
```

```
temps4

#Export the temps4 to excel
excel <- write.csv(temps4, "hw3q2.csv")
excel

#Please view the excel graph named Q2  to view my analysis. From the graph below, we can predict that the unofficial summer has not gotten later over the past 20 years.
```



<u>Question 8.1</u>

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

Linear regression model would be appropriate to use when trying to determine how to price a food item in a restaurant menu. For example, I recently went to a restaurant called Hawkers where I ordered a dish called Singapore Main Fun. I asked the server to customize my order by replacing meat with veggies and tofu. Since the order is custom made, Hawkers could potentially use linear regression model to come up with a price based on number of veggies added, type of veggies requested, cost of veggies, cost of tofu per grams, grams of tofu used, time taken to prepare the custom dish. The model would be trained on data from previous similar orders requested.

Question 8.2

Using crime data from http://www.statsci.org/data/general/uscrime.txt (file `uscrime.txt`, description at http://www.statsci.org/data/general/uscrime.html), use regression (a useful R function is `lm` or `glm`) to predict the observed crime rate in a city

```
#Read table
```

```
uscrime <- read.table("8.2uscrimeSummer2018.txt", stringsAsFactors = FALSE, header = TRUE)
```

```
# Used linear regression model with Crime as the response variable and the rest as predictors. Includes all 15 factors.
```

```
model1 <- lm(Crime~. , data = uscrime)
model1
summary(model1)
```

```
#The p value indicates weather the coefficients estimates is close to zero. The more the predictors the higher the r square value gets. Here its giving the training estimate of r square and I would not use this as the estimated model quality.
```

```
# This model is likely overfit since we have a lot of non-star coefficients. The high p value is also an indicator that the estimate if close to zero so we don’t need to include in the model.
```

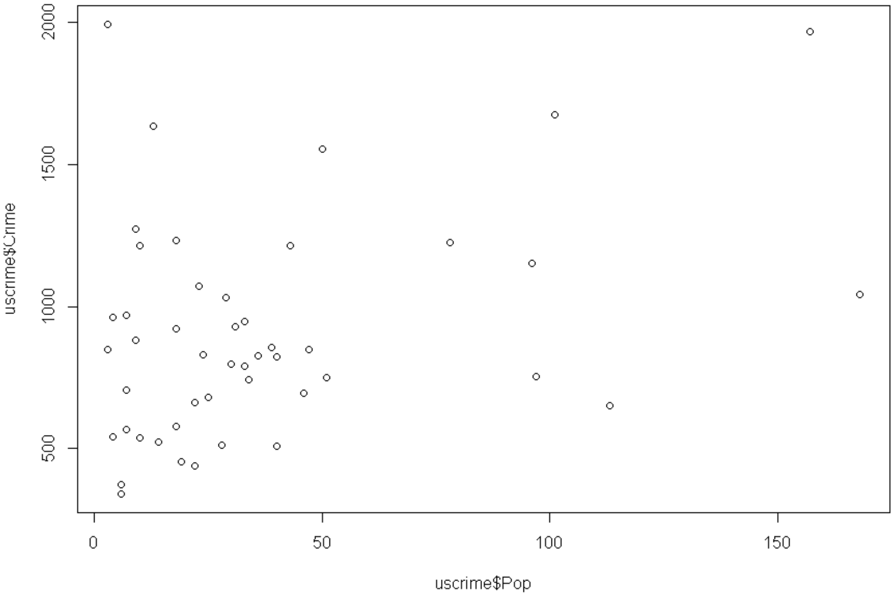
```
#Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5.98e+03   1.63e+03   -3.68  0.00089 ***
M            8.78e+01   4.17e+01    2.11  0.04344 *  
So          -3.80e+00   1.49e+02   -0.03  0.97977    
Ed           1.88e+02   6.21e+01    3.03  0.00486 ** 
Po1          1.93e+02   1.06e+02    1.82  0.07889 .  
Po2         -1.09e+02   1.17e+02   -0.93  0.35883    
LF          -6.64e+02   1.47e+03   -0.45  0.65465    
M.F          1.74e+01   2.04e+01    0.86  0.39900    
Pop         -7.33e-01   1.29e+00   -0.57  0.57385    
NW           4.20e+00   6.48e+00    0.65  0.52128    
U1          -5.83e+03   4.21e+03   -1.38  0.17624    
U2           1.68e+02   8.23e+01    2.04  0.05016 .  
Wealth       9.62e-02   1.04e-01    0.93  0.36075    
Ineq         7.07e+01   2.27e+01    3.11  0.00398 ** 
Prob        -4.86e+03   2.27e+03   -2.14  0.04063 *  
Time        -3.48e+00   7.17e+00   -0.49  0.63071    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 209 on 31 degrees of freedom
Multiple R-squared:  0.803,    Adjusted R-squared:  0.708 
F-statistic: 8.43 on 15 and 31 DF,  p-value: 3.54e-07
```

```
#Get prediction for the given point.
```

```
point <- data.frame(M = 14.0, So = 0, Ed = 10.0, Po1 = 12.0, Po2 = 15.5, LF = 0.640, M.F = 94.0, Pop = 150, NW = 1.1, U1 = 0.120, U2 = 3.6, Wealth = 3200, Ineq = 20.1, Prob = 0.04, Time = 39.0)
```

```
prediction1 <- predict(model1, point)
prediction1
plot(uscrime$Pop, uscrime$Crime)
```

#155.4349 is the crime value that this model is predicting. The crime value falls outside of normal range. The model here seems to be an overfit. The lowest range in the data set is 342. The problem is that the 15-factor model includes a lot of factors that aren’t significant – they have high p-values (see output below).



#Counteract overfitting. Step regression helps in giving the optimal coefficients (reduce the # of coefficients) to use in this model.

```
step(model1)
```

#after re running the model, more coefficients are important. R squared value has reduced from before.

```
model2 <- lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob, data = uscrime)
summary(model2)
```

```
#Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -6426.1     1194.6   -5.38  4.0e-06 ***
M               93.3       33.5    2.79   0.0083 ** 
Ed             180.1       52.8    3.41   0.0015 ** 
Po1            102.7       15.5    6.61  8.3e-08 ***
M.F             22.3       13.6    1.64   0.1087    
U1           -6086.6     3339.3   -1.82   0.0762 .  
U2             187.3       72.5    2.58   0.0137 *  
Ineq            61.3       14.0    4.39  8.6e-05 ***
Prob         -3796.0     1490.6   -2.55   0.0151 *  
---
```

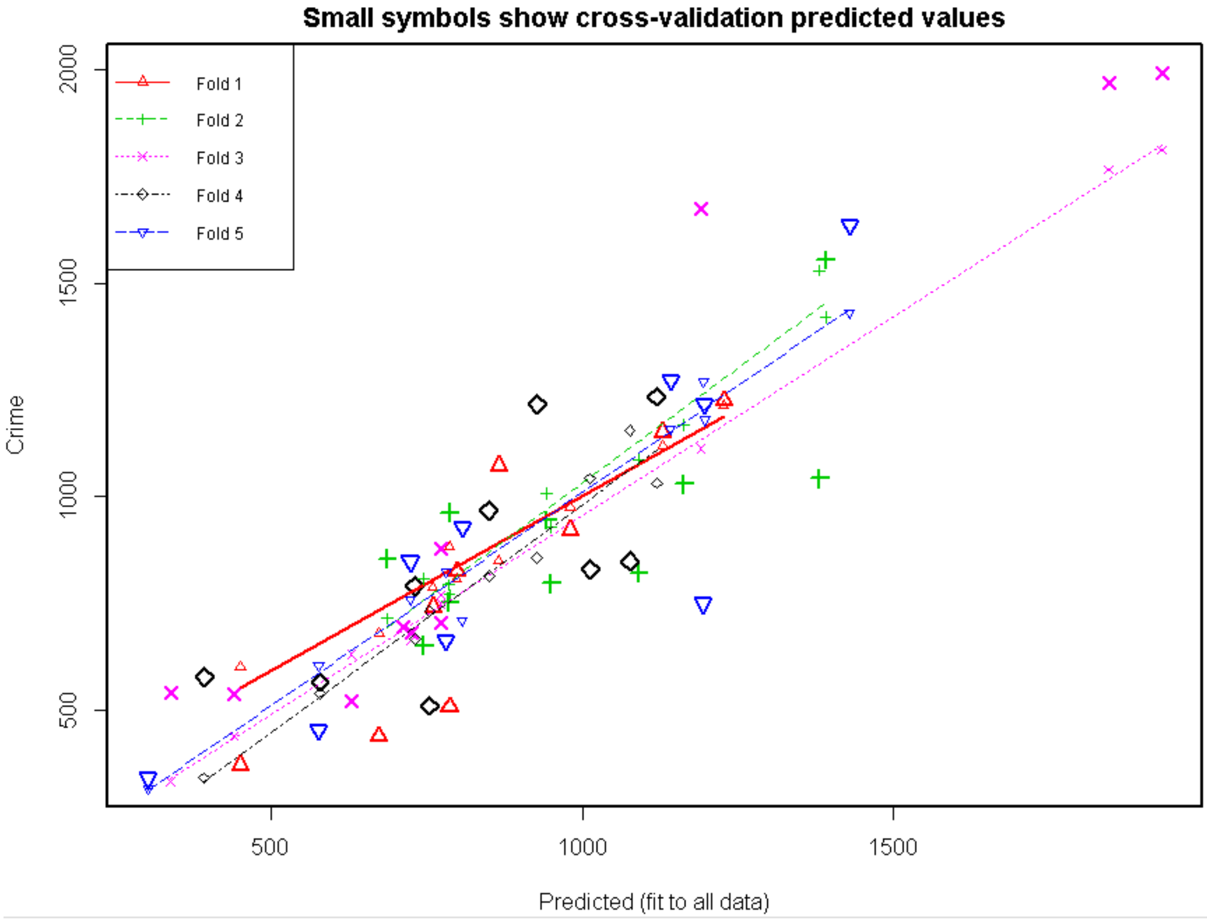
```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 196 on 38 degrees of freedom
Multiple R-squared:  0.789,    Adjusted R-squared:  0.744
F-statistic: 17.7 on 8 and 38 DF,  p-value: 1.16e-10
```

```
prediction2 <- predict(model2, point)
prediction2
```

```
#1038 is the crime value that this model is predicting. The crime value falls
within of normal range.
```

```
#Assess the quality of the model
library(DAAG)
layout(1,1,1)
model2cv <- cv.lm(uscrime, model2, m=5, seed = 42)
#ms = 44981

# To validate MSE, when I run the below model, I get the same answer 44981.
MSE = attr(model2cv, "ms")
MSE
```

#The first model’s R2 was 0.803, and the second model gave 0.789. Measuring on training data isn’t a good estimate, also because of the possibility of overfitting. I used cv.lm to do cross validation. The data set is split into 5 smaller data sets and you build a model of the 4 and test on the 5th one and you average the quality. Using 5 fold cross validation R2 was about 0.627.


Small symbols show cross-validation predicted values
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Crime
500
1000
1500
2000
500
1000
1500
Predicted (fit to all data)