Question 6.2

1. Using July through October daily-high-temperature data for Atlanta for 1996 through 2015, use a CUSUM approach to identify when unofficial summer ends (i.e., when the weather starts cooling off) each year. You can get the data that you need from the file `temps.txt` or online, for example at http://www.iweathernet.com/atlanta-weather-records or https://www.wunderground.com/history/airport/KFTY/2015/7/1/CustomHistory.html . You can use R if you'd like, but it's straightforward enough that an Excel spreadsheet can easily do the job too.
2. Use a CUSUM approach to make a judgment of whether Atlanta's summer climate has gotten warmer in that time (and if so, when).

Solution 6.2.1:

Code Used:

```
setwd("C:/Users/jtejeda/OneDrive - Clarios/Documents/R_projects/hw3-SP22")

temp_data <- read.table("temps.txt", stringsAsFactors = FALSE, header = T)
head(temp_data)

#compute daily averages across the years
daily_average <- rowMeans(temp_data[c(2:length(temp_data))], dims=1, na.rm=T)

# calculate mu of the daily averages. This will be used to compute the comulative sums # nolint
mu <- mean(daily_average)

# compute the difference between the mean of the time series and each "day"
#da_minus_mu <- date_avgs - da_mu

# Set an arbitrary C that will be used to calibrate the sensitivity of the model
C <- 0

# calculate the sum of each individual x_t
S_sub_t <- daily_average - mu - C

# create an empty vector for looping
# include an additional zero to help with indexing
preCUSUM <- 0 * S_sub_t
cusum <- append(preCUSUM, 0)

# loop through each day (i) to calculate the CUSUM (S_sub_t) and
# update the accumulator. The below loop is assigning a value to the accumulator
# variable which is in turn carried onto the next iteration (or not) via the ifelse statement
```

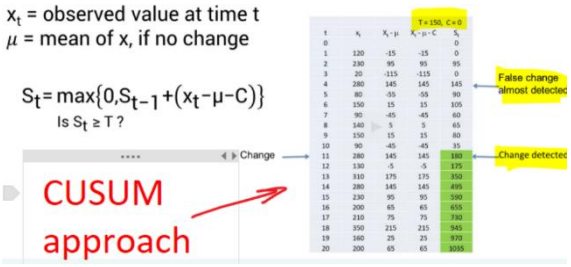
```
for (i in 1:length(S_sub_t))
    {
  accumulator <- cusum[i] + S_sub_t[i]
  ifelse(accumulator > 0, cusum[i+1] <- accumulator, cusum[i+1] <- 0)}

plot(cusum)

#This finds the max value on the cumsum graph.
cummax(cusum)
temp_data[78,1]

#Results of when "the summer unofficially ends" based on different values of C
#August 24 @ C=5
#August 30 @ C=4
#Sept 13 @ C=1
#Sept 16 @ C=0
```

In the code above each day for the many years are averaged and then those values are averaged once more to calculate mu. Mu and an arbitrary value of C are then used to calculate S_sub_t which the value from each data point that will add or subtract from the accumulator. The loop is following the below equation from lectures.



Cade Results:

> setwd("C:/Users/jtejeda/OneDrive - Clarios/Documents/R_projects/hw3-SP22")

> temp_data <- read.table("temps.txt", stringsAsFactors = FALSE, header = T)

> head(temp_data)

	DAY	X1996	X1997	X1998	X1999	X2000	X2001	X2002	X2003	X2004	X2005	X2006	X2007
1	1-Jul	98	86	91	84	89	84	90	73	82	91	93	95
2	2-Jul	97	90	88	82	91	87	90	81	81	89	93	85
3	3-Jul	97	93	91	87	93	87	87	87	86	86	93	82
4	4-Jul	90	91	91	88	95	84	89	86	88	86	91	86
5	5-Jul	89	84	91	90	96	86	93	80	90	89	90	88
6	6-Jul	93	84	89	91	96	87	93	84	90	82	81	87

	X2008	X2009	X2010	X2011	X2012	X2013	X2014	X2015
1	85	95	87	92	105	82	90	85
2	87	90	84	94	93	85	93	87
3	91	89	83	95	99	76	87	79
4	90	91	85	92	98	77	84	85
5	88	80	88	90	100	83	86	84
6	82	87	89	90	98	83	87	84

```
> #compute daily averages across the years
> daily_average <- rowMeans(temp_data[c(2:length(temp_data))], dims=1, na.rm=T)

> # calculate mu of the daily averages. This will be used to compute the comulative sums # nolint
> mu <- mean(daily_average)

> # compute the difference between the mean of the time series and each "day"
> #da_minus_mu <- date_avgs - da_mu
>
> # Set an arbitrary C that will  .... [TRUNCATED]

> # calculate the sum of each individual x_t
> S_sub_t <- daily_average - mu - C

> # create an empty vector for looping
> # include an additional zero to help with indexing
> preCUSUM <- 0 * S_sub_t

> cusum <- append(preCUSUM, 0)

> # loop through each day (i) to calculate the CUSUM (S_sub_t) and
> # update the accumulator. The below loop is assigning a value to the accumulator  .... [TRUNCATED]

> plot(cusum)

> #This finds the max value on the cumsum graph.
> cummax(cusum)
  [1]   0.000000   5.510976  10.521951  15.582927  20.593902  25.504878
  [7]  30.015854  33.776829  39.587805  46.298780  51.509756  56.120732
 [13]  60.931707  64.792683  69.653659  73.314634  78.075610  83.936585
 [19]  89.847561  96.908537 102.969512 109.580488 115.691463 121.402439
 [25] 127.163415 131.824390 137.985366 144.196341 150.807317 156.718293
 [31] 162.929268 167.740244 172.951220 178.262195 184.473171 191.434146
 [37] 199.245122 205.306098 210.917073 216.328049 221.989024 227.900000
 [43] 233.760976 238.321951 243.082927 248.043902 252.704878 258.165854
 [49] 263.876829 270.687805 277.648780 283.609756 289.370732 295.431707
 [55] 300.492683 305.003659 308.164634 313.275610 317.536585 321.347561
 [61] 326.308537 328.769512 331.330488 333.241463 335.152439 337.713415
 [67] 340.174390 343.035366 344.296341 345.707317 347.618293 349.329268
 [73] 351.240244 353.451220 355.412195 355.412195 355.484146 355.845122
 [79] 355.845122 355.845122 355.845122 355.845122 355.845122 355.845122
 [85] 355.845122 355.845122 355.845122 355.845122 355.845122 355.845122
 [91] 355.845122 355.845122 355.845122 355.845122 355.845122 355.845122
 [97] 355.845122 355.845122 355.845122 355.845122 355.845122 355.845122
[103] 355.845122 355.845122 355.845122 355.845122 355.845122 355.845122
[109] 355.845122 355.845122 355.845122 355.845122 355.845122 355.845122
[115] 355.845122 355.845122 355.845122 355.845122 355.845122 355.845122
[121] 355.845122 355.845122 355.845122 355.845122

> temp_data[78,1]
[1] "16-Sep"

> #Results of when "the summer unofficially ends" based on different values of C
> #August 24 @ C=5
> #August 30 @ C=4
> #Sept 13 @ C=1
> #Sept 16 @ C = 0
```

Conclusion:

Since the data follows a very normal distribution with low variability and It's already averaged daily over several years, I didn't see the need to use the C parameter. This parameter is meant to regulate the sensitivity of the model to avoid detecting false changes and this data set does not seem to need that.

The results of the model with C = 0 arrive at the conclusion that the summer unofficially ends on September 16th, right at index 78 in the graph below.