I. **Question 6.2.a use a CUSUM approach to identify when unofficial summer ends**

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.

Results:

Step#1 : Identify best values of C and T

How : I used excel and obtained average temperature for all the days across 20 years to identify the best suited C and T values and also to validate the results in R. As part of analysis, I have selected C=5 and threshold as 82

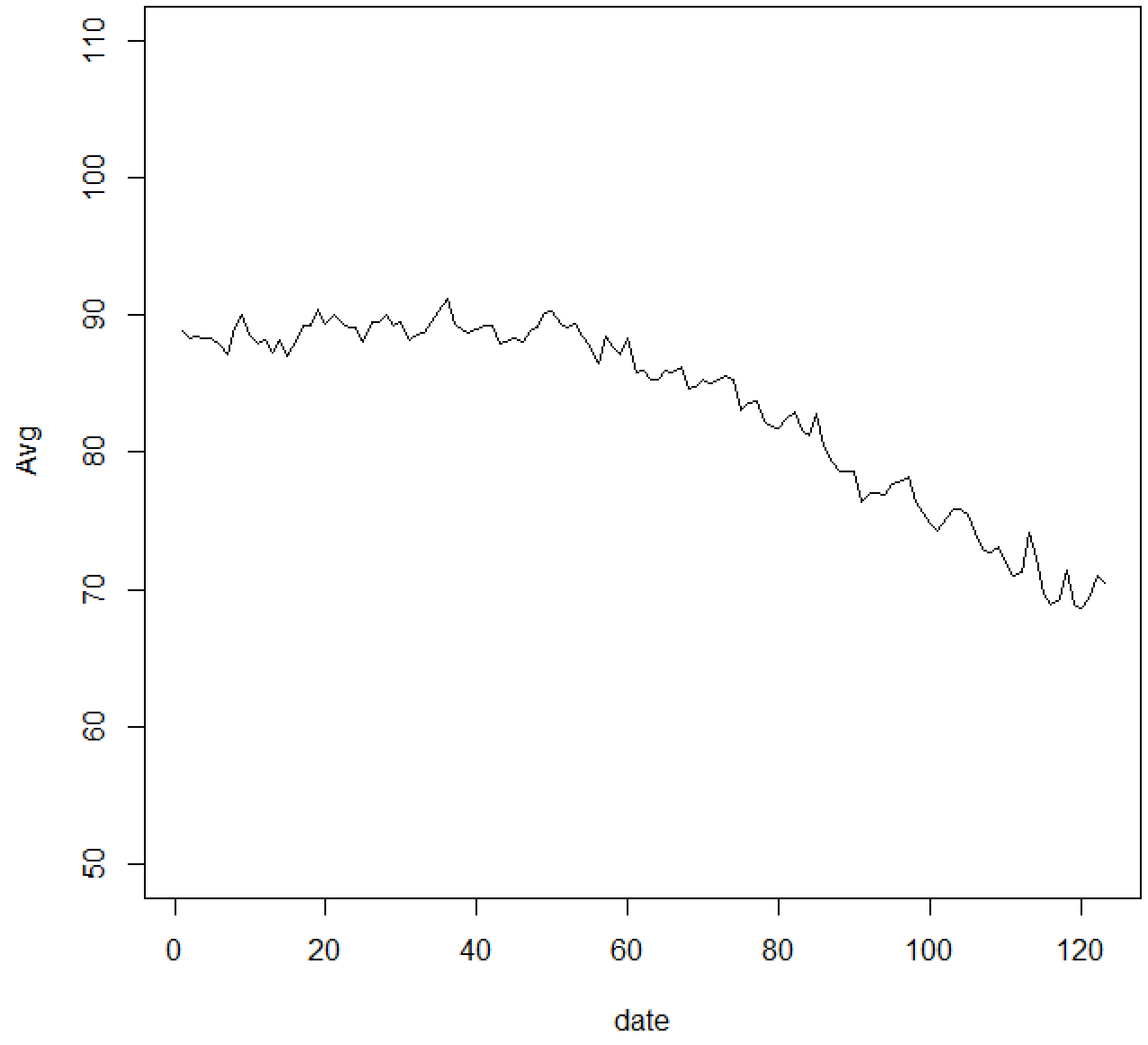
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
																						C=5;T=82				
DAY	19	19	19	19	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	Avg of All year;x(	Mean	mean-x(t)	C	mean-x(t)-c	s(t-1)+(mean-x(t)-c)
1-Jul	98	86	91	84	89	84	90	73	82	91	93	95	85	95	87	92	105	82	90	85	88.85	83.33902	-5.51098	0	-5.51098	0
2-Jul	97	90	88	82	91	87	90	81	81	89	93	85	87	90	84	94	93	85	93	87	88.35	83.33902	-5.01098	0	-5.01098	0
3-Jul	97	93	91	87	93	87	87	87	86	86	93	82	91	89	83	95	99	76	87	79	88.4	83.33902	-5.06098	0	-5.06098	0
4-Jul	90	91	91	88	95	84	89	86	88	86	91	86	90	91	85	92	98	77	84	85	88.35	83.33902	-5.01098	0	-5.01098	0
5-Jul	89	84	91	90	96	86	93	80	90	89	90	88	88	80	88	90	100	83	86	84	88.25	83.33902	-4.91098	0	-4.91098	0
6-Jul	93	84	89	91	96	87	93	84	90	82	81	87	82	87	89	90	98	83	87	84	87.85	83.33902	-4.51098	0	-4.51098	0
7-Jul	93	75	93	82	96	87	89	87	89	76	80	82	88	86	94	94	93	79	89	90	87.1	83.33902	-3.76098	0	-3.76098	0
8-Jul	91	87	95	86	91	89	89	90	87	88	82	82	90	82	97	94	95	88	90	90	89.15	83.33902	-5.81098	0	-5.81098	0
9-Jul	93	84	95	87	96	91	90	89	88	89	84	89	89	84	96	91	97	88	90	91	90.05	83.33902	-6.71098	0	-6.71098	0
10-Jul	93	87	91	87	99	87	91	84	89	78	84	86	87	84	90	92	95	87	87	93	88.55	83.33902	-5.21098	0	-5.21098	0
11-Jul	90	84	91	82	96	90	84	84	90	83	90	85	89	86	93	95	90	80	85	92	87.95	83.33902	-4.61098	0	-4.61098	0
12-Jul	91	88	86	77	93	90	77	86	89	86	91	87	93	90	90	95	84	87	90	93	88.15	83.33902	-4.81098	0	-4.81098	0
13-Jul	93	86	88	73	91	86	82	87	91	84	91	86	85	84	91	97	90	78	89	92	87.2	83.33902	-3.86098	0	-3.86098	0
14-Jul	93	90	87	81	93	82	88	84	91	87	91	84	88	89	91	90	90	85	90	90	88.2	83.33902	-4.86098	0	-4.86098	0
15-Jul	82	91	91	81	93	82	91	86	84	84	91	81	89	89	94	80	90	86	86	89	87	83.33902	-3.66098	0	-3.66098	0
16-Jul	91	91	87	86	93	84	93	88	84	85	91	86	89	90	89	85	92	87	83	88	88.1	83.33902	-4.76098	0	-4.76098	0
17-Jul	96	89	90	82	91	87	93	88	84	89	93	89	88	88	87	87	93	91	86	93	89.2	83.33902	-5.86098	0	-5.86098	0
18-Jul	95	89	91	87	97	88	93	88	87	90	93	89	90	82	83	89	93	87	82	92	89.25	83.33902	-5.91098	0	-5.91098	0
19-Jul	96	89	95	88	100	90	93	88	84	89	96	88	91	80	90	94	91	90	85	91	90.4	83.33902	-7.06098	0	-7.06098	0
20-Jul	99	90	91	90	99	87	91	88	88	89	93	86	94	82	91	91	84	86	76	93	89.4	83.33902	-6.06098	0	-6.06098	0
21-Jul	91	89	91	90	93	84	95	89	89	90	93	86	95	86	94	92	90	87	82	93	89.95	83.33902	-6.61098	0	-6.61098	0
22-Jul	95	84	89	91	96	87	91	86	89	91	91	79	92	84	95	94	95	85	83	92	89.45	83.33902	-6.11098	0	-6.11098	0
23-Jul	91	87	91	93	87	90	89	81	93	91	86	82	87	87	97	92	97	84	88	88	89.05	83.33902	-5.71098	0	-5.71098	0
24-Jul	93	88	91	93	82	84	87	82	95	90	87	87	88	88	94	92	97	86	87	91	89.1	83.33902	-5.76098	0	-5.76098	0
25-Jul	84	89	86	91	75	82	84	84	89	92	88	87	89	90	95	90	98	89	88	90	88	83.33902	-4.66098	0	-4.66098	0
26-Jul	84	89	88	93	82	88	86	87	87	94	93	87	87	92	95	94	98	86	89	91	89.5	83.33902	-6.16098	0	-6.16098	0
27-Jul	82	91	80	93	88	90	89	87	84	92	95	90	90	90	93	94	97	82	92	92	89.55	83.33902	-6.21098	0	-6.21098	0
28-Jul	79	91	88	93	91	84	91	89	89	90	96	89	93	89	90	90	97	86	90	94	89.95	83.33902	-6.61098	0	-6.61098	0
29-Jul	90	89	89	93	89	89	91	88	87	83	91	87	92	85	94	93	94	86	82	93	89.25	83.33902	-5.91098	0	-5.91098	0
30-Jul	91	88	90	97	87	89	88	84	89	78	91	92	90	82	95	96	96	90	84	94	89.55	83.33902	-6.21098	0	-6.21098	0
31-Jul	87	72	86	99	86	87	90	88	90	84	94	90	88	85	95	96	88	80	85	93	88.15	83.33902	-4.81098	0	-4.81098	0
1-Aug	86	80	86	96	86	84	93	84	91	82	95	92	89	89	96	91	94	87	81	89	88.55	83.33902	-5.21098	0	-5.21098	0

Step#2: Ingest data

```
> ###Question 6.2###
>
> ###Input : July through October daily-high-temperature data for Atlanta for 1996 through 2015,\
> ####ASK: use a CUSUM approach to identify when unofficial summer ends
> ####(i.e., when the weather starts cooling off) each year.
> ####OPTIONS: You can use R if you'd like, OR Excel spreadsheet
>
>
> ##########CLEAR##########
> rm(list = ls())
>
> ##########LIBRARY##########
>
>
> ##########INGEST FILE##########
>
> data<- read.table("6_2temps.txt",header=TRUE,stringsAsFactors = FALSE,sep="\t")
> #summary(data)
> length(data) # no of columns
[1] 21
>
> head(data,10)
      DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006
1   1-Jul    98    86    91    84    89    84    90    73    82    91    93
2   2-Jul    97    90    88    82    91    87    90    81    81    89    93
3   3-Jul    97    93    91    87    93    87    87    87    86    86    93
4   4-Jul    90    91    91    88    95    84    89    86    88    86    91
5   5-Jul    89    84    91    90    96    86    93    80    90    89    90
6   6-Jul    93    84    89    91    96    87    93    84    90    82    81
7   7-Jul    93    75    93    82    96    87    89    87    89    76    80
8   8-Jul    91    87    95    86    91    89    89    90    87    88    82
9   9-Jul    93    84    95    87    96    91    90    89    88    89    84
10 10-Jul    93    87    91    87    99    87    91    84    89    78    84
   X2007 X2008 X2009 X2010 X2011 X2012 X2013 X2014 X2015
1     95    85    95    87    92   105    82    90    85
2     85    87    90    84    94    93    85    93    87
3     82    91    89    83    95    99    76    87    79
4     86    90    91    85    92    98    77    84    85
5     88    88    80    88    90   100    83    86    84
6     87    82    87    89    90    98    83    87    84
7     82    88    86    94    94    93    79    89    90
8     82    90    82    97    94    95    88    90    90
9     89    89    84    96    91    97    88    90    91
10    86    87    84    90    92    95    87    87    93
```

Step#3: Obtain average of all days and show averages by day

```
> ##########ggplot(data)##########
> date_avgs <- rowMeans(data[c(2:length(data))], dims=1, na.rm=T)
> cbind(data[1],date_avgs)
       DAY date_avgs
1    1-Jul     88.85
2    2-Jul     88.35
3    3-Jul     88.40
4    4-Jul     88.35
5    5-Jul     88.25
6    6-Jul     87.85
7    7-Jul     87.10
8    8-Jul     89.15
9    9-Jul     90.05
10  10-Jul     88.55
11  11-Jul     87.95
12  12-Jul     88.15
13  13-Jul     87.20
14  14-Jul     88.20
15  15-Jul     87.00
16  16-Jul     88.10
17  17-Jul     89.20
18  18-Jul     89.25
19  19-Jul     90.40
20  20-Jul     89.40
21  21-Jul     89.95
22  22-Jul     89.45
23  23-Jul     89.05
24  24-Jul     89.10
25  25-Jul     88.00
26  26-Jul     89.50
27  27-Jul     89.55
28  28-Jul     89.95
29  29-Jul     89.25
30  30-Jul     89.55
31  31-Jul     88.15
32   1-Aug     88.55
33   2-Aug     88.65
34   3-Aug     89.55
35   4-Aug     90.30
36   5-Aug     91.15
37   6-Aug     89.40
38   7-Aug     88.95
39   8-Aug     88.75
40   9-Aug     89.00
41  10-Aug     89.25
```



Step#4 setup C and Threhold . calculate CUSUM

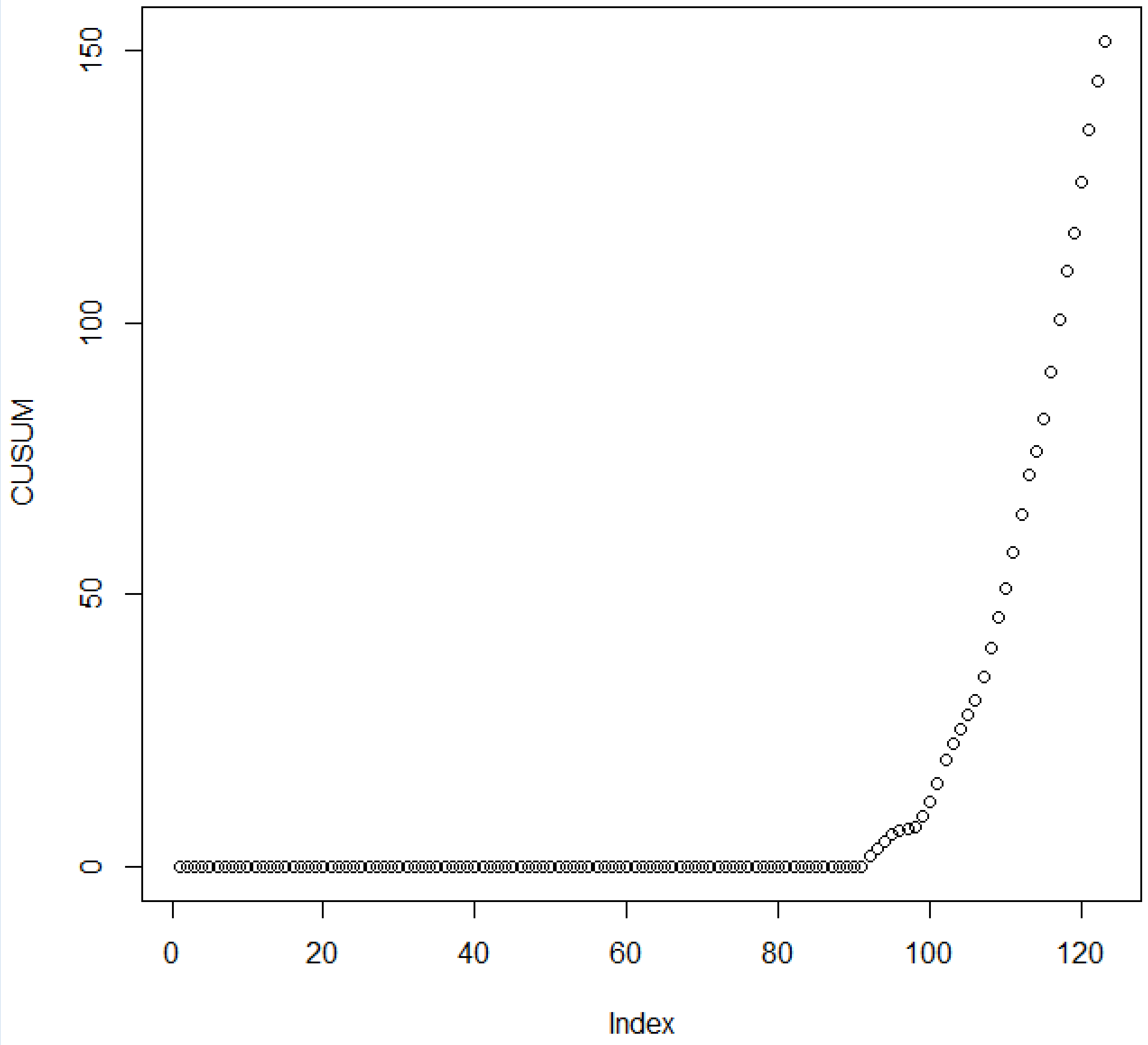
```
> ##########CHECK NUMBER OF ROWS##########
> n <- length(date_avgs) # 123 data points
> ntest <- nrow(data[1])
>
> x_t <- date_avgs
>
> mean_x_t <- mean(x_t)
>
> mean_x_t
[1] 83.33902
>
> # set up based on analysis in excel
> C <- 5
>
> #Threshold for Temp drop  set to 82.
>
> #plot average
> #plot(date_avgs,ylim=c(50,110),xlab="date",ylab="Avg",type="l")
>
> # as we are seeing decrease in temperature, we calculate mean - data
>
> mean_data <- mean_x_t-date_avgs
>
> # subtract C from the difference score
> s_t <- mean_data - C
>
> s_t
  [1] -10.5109756 -10.0109756 -10.0609756 -10.0109756  -9.9109756  -9.5109756  -8.7609756 -10.8109756
  [9] -11.7109756 -10.2109756  -9.6109756  -9.8109756  -8.8609756  -9.8609756  -8.6609756  -9.7609756
 [17] -10.8609756 -10.9109756 -12.0609756 -11.0609756 -11.6109756 -11.1109756 -10.7109756 -10.7609756
 [25]  -9.6609756 -11.1609756 -11.2109756 -11.6109756 -10.9109756 -11.2109756  -9.8109756 -10.2109756
 [33] -10.3109756 -11.2109756 -11.9609756 -12.8109756 -11.0609756 -10.6109756 -10.4109756 -10.6609756
 [41] -10.9109756 -10.8609756  -9.5609756  -9.7609756  -9.9609756  -9.6609756 -10.4609756 -10.7109756
 [49] -11.8109756 -11.9609756 -10.9609756 -10.7609756 -11.0609756 -10.0609756  -9.5109756  -8.1609756
 [57] -10.1109756  -9.2609756  -8.8109756  -9.9609756  -7.4609756  -7.5609756  -6.9109756  -6.9109756
 [65]  -7.5609756  -7.4609756  -7.8609756  -6.2609756  -6.4109756  -6.9109756  -6.7109756  -6.9109756
 [73]  -7.2109756  -6.9609756  -4.7609756  -5.3109756  -5.3609756  -3.9109756  -3.5109756  -3.3609756
```

```
> #precusum <- 0 * s_t
> cusum <- append(0, 0)
> cusum
[1] 0 0
> cusum[1]
[1] 0
>
> for (i in 1:length(s_t))
+       {
+   ifelse(cusum[i] + s_t[i-1] > 0, cusum[i+1] <- cusum[i] + s_t[i-1], cusum[i+1] <- 0)
+ }
>
> cusum
  [1]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [10]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [19]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [28]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [37]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [46]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [55]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [64]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [73]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [82]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [91]   0.000000   0.000000   1.989024   3.328049   4.567073   5.956098   6.595122   7.084146   7.223171
[100]   9.212195  11.951220  15.490244  19.579268  22.768293  25.257317  27.796341  30.685366  34.824390
[109]  40.263415  45.952439  51.191463  57.630488  64.919512  72.008537  76.247561  82.236585  90.925610
[118] 100.414634 109.403659 116.342683 125.781707 135.520732 144.509756 151.798780
> length(cusum[-1])
[1] 123
> cbind(data[1],cusum[-1])
      DAY  cusum[-1]
1     1-Jul   0.000000
2     2-Jul   0.000000
3     3-Jul   0.000000
4     4-Jul   0.000000
5     5-Jul   0.000000
```

Step#5 : find the day when the threshold has reached

```
> cusum[1]
[1] 0
>
> for (i in 1:length(s_t))
+       {
+   ifelse(cusum[i] + s_t[i-1] > 0, cusum[i+1] <- cusum[i] + s_t[i-1], cusum[i+1] <- 0)
+ }
>
> cusum
  [1]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [10]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [19]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [28]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [37]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [46]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [55]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [64]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [73]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [82]   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
 [91]   0.000000   0.000000   1.989024   3.328049   4.567073   5.956098   6.595122   7.084146   7.223171
[100]   9.212195  11.951220  15.490244  19.579268  22.768293  25.257317  27.796341  30.685366  34.824390
[109]  40.263415  45.952439  51.191463  57.630488  64.919512  72.008537  76.247561  82.236585  90.925610
[118] 100.414634 109.403659 116.342683 125.781707 135.520732 144.509756 151.798780
> length(cusum[-1])
[1] 123
> cbind(data[1],cusum[-1])
       DAY  cusum[-1]
1     1-Jul   0.000000
2     2-Jul   0.000000
3     3-Jul   0.000000
4     4-Jul   0.000000
5     5-Jul   0.000000
6     6-Jul   0.000000
>
> which(cusum >= 82)
[1] 116 117 118 119 120 121 122 123 124
>
> data[116, 1]
[1] "24-Oct"
```

The date returned for unofficial summer end is 10/24



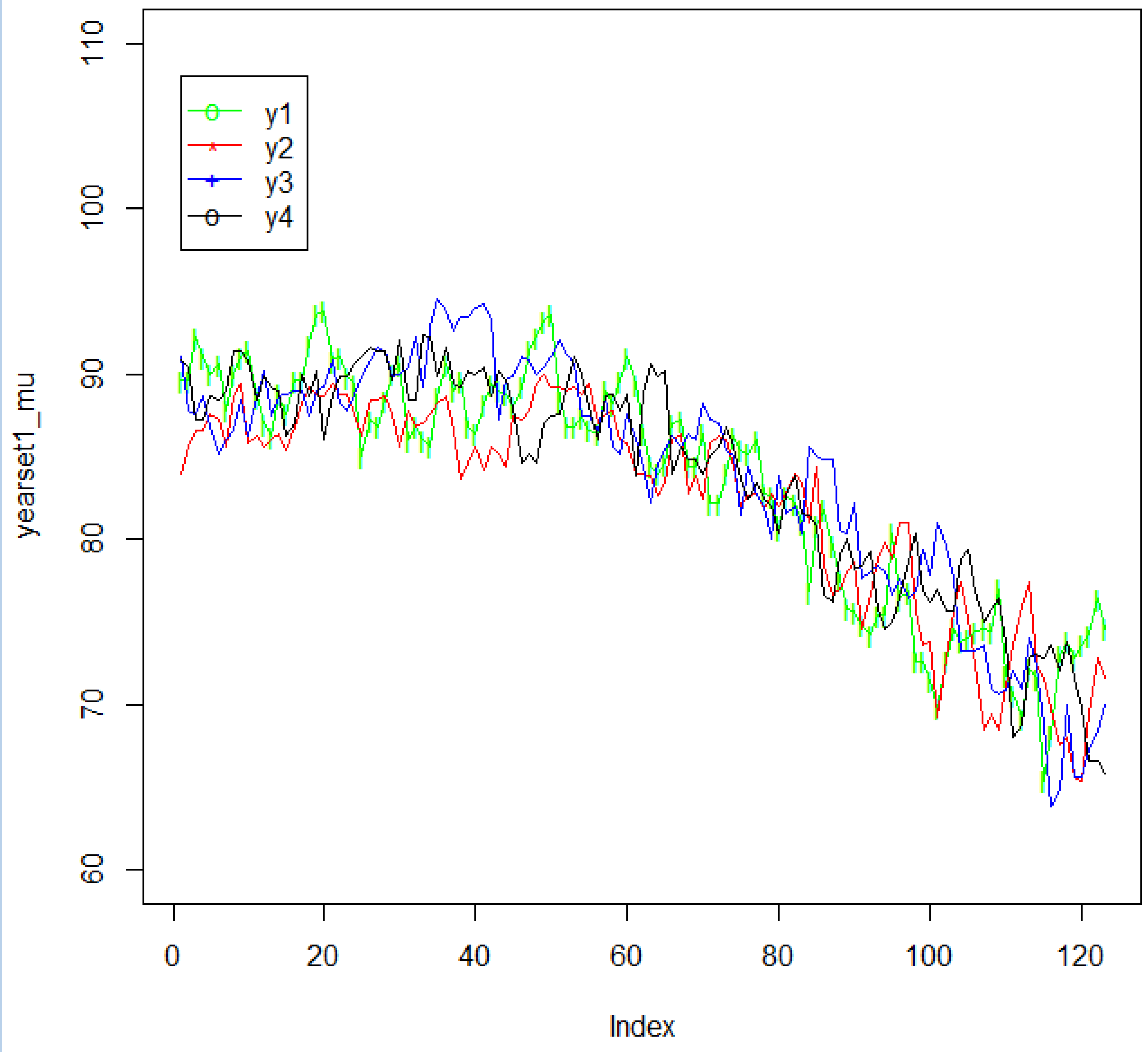
2. **Question 6.2.b** Use a CUSUM approach to make a judgment of whether Atlanta's summer climate has gotten warmer in that time (and if so, when).

Step#1 Understand data

Yearly datasets were hard to visualize. Hence I categorized the datasets into 5 year buckets and obtained mean for each bucket

```
>
> #######HAS ATLANTA CLIMATE GOT WARMER BY YEARS?##########
> # categorize the 20 years data into 5 buckets of 5 years
>
> yearset_1 <-data[2:6]
> head(yearset_1)
  X1996 X1997 X1998 X1999 X2000
1    98    86    91    84    89
2    97    90    88    82    91
3    97    93    91    87    93
4    90    91    91    88    95
5    89    84    91    90    96
6    93    84    89    91    96
> yearset_2 <-data[7:11]
> head(yearset_2)
  X2001 X2002 X2003 X2004 X2005
1    84    90    73    82    91
2    87    90    81    81    89
3    87    87    87    86    86
4    84    89    86    88    86
5    86    93    80    90    89
6    87    93    84    90    82
> yearset_3 <-data[12:16]
> head(yearset_3)
  X2006 X2007 X2008 X2009 X2010
1    93    95    85    95    87
2    93    85    87    90    84
3    93    82    91    89    83
4    91    86    90    91    85
5    90    88    88    80    88
6    81    87    82    87    89
> yearset_4 <-data[17:length(data)]
> head(yearset_4)
  X2011 X2012 X2013 X2014 X2015
1    92   105    82    90    85
2    94    93    85    93    87
3    95    99    76    87    79
4    92    98    77    84    85
5    90   100    83    86    84
6    90    98    83    87    84
```

```
> yearset1_mu= rowMeans(yearset_1, dims=1, na.rm=T)
> yearset1_mu
  [1] 89.6 89.6 92.2 91.0 90.0 90.6 87.8 90.0 91.0 91.4 88.6 87.0 86.2 88.8 87.6 89.6 89.6 91.8 93.6 93.8
 [21] 90.8 91.0 89.8 89.4 85.0 87.2 86.8 88.4 90.0 90.6 86.0 86.8 86.0 85.6 88.6 90.6 89.0 89.6 86.8 86.4
 [41] 88.2 89.4 89.0 88.8 88.0 89.2 91.4 92.2 93.2 93.6 88.4 86.8 86.8 87.4 86.6 86.4 89.0 88.2 89.6 91.0
 [61] 89.4 86.4 84.4 84.0 84.6 87.0 87.2 84.4 84.4 86.4 82.2 82.2 84.0 86.2 85.4 85.2 86.0 82.8 82.6 80.6
 [81] 82.6 82.2 81.0 76.8 80.8 81.8 79.6 77.6 75.8 75.6 74.8 74.2 75.4 75.4 80.4 76.4 76.8 72.6 72.6 71.4
[101] 69.8 72.6 74.6 73.8 74.0 74.4 74.6 74.4 77.0 71.8 70.6 69.2 72.2 71.6 65.4 68.2 73.0 73.8 72.8 73.6
[121] 74.2 76.4 74.6
> yearset2_mu= rowMeans(yearset_2, dims=1, na.rm=T)
> yearset3_mu= rowMeans(yearset_3, dims=1, na.rm=T)
> yearset4_mu= rowMeans(yearset_4, dims=1, na.rm=T)
>
>
> Day=data[1]
> #Day
>
> #plot(yearset1_mu,type="l",ylab="Avg. Temperature")
> plot(yearset1_mu, type="o", col="green", pch="1", lty=1, ylim=c(60,110) )
>
> ### Use RED for 2nd 5 years.
> #points(yearset2_mu, col="red", pch="*")
> lines(yearset2_mu, col="red",lty=1)
>
> ### Use GREEN for 3nd 5 years.
> lines(yearset3_mu, col="blue",lty=1)
>
> ### Use GREEN for 4th 5 years.
> lines(yearset4_mu, col="black",lty=1)
>
>
>
> ###add legend
> legend(1,108,legend=c("y1","y2","y3","y4"), col=c("green","red","blue","black"),
+                                     pch=c("o","*","+"),lty=c(1,1,1), ncol=1)
```



Year#1 category :1996 -2000
Year#2 category :2001 -2005
Year#3 category :2006-2010
Year#4 category :2011 -2015

Visually it looks like last 2 set (9 years average) is showing spikes

```
> # average of all days in each year set
>
> yearset1_overall_mu=mean(yearset1_mu)
> yearset2_overall_mu=mean(yearset2_mu)
> yearset3_overall_mu=mean(yearset3_mu)
> yearset4_overall_mu=mean(yearset4_mu)
>
>
> rbind(yearset1_overall_mu,yearset2_overall_mu,yearset3_overall_mu,yearset4_overall_mu)
                        [,1]
yearset1_overall_mu 83.40813
yearset2_overall_mu 82.34797
yearset3_overall_mu 83.83252
yearset4_overall_mu 83.76748
```

Overall Avg. temperature of the year as well shows increase in the last 2 data sets.

3. I developed CUSUM increase approach in spreadsheet with the below parameters and noticed the threshold 4 reaches in **2011**.

	A	B	C	D	E	F	G	H	I
1					C=0.65	T=4			
2	year	yearly Avg. Temp	Mean	x(t)-mean	C	x(t)-mean-c	s(t-1)+(x(t)-mean-c)		
3	1996	83.71544715	83.33902	0.376422764	0.65	-0.273577236	0		
4	1997	81.67479675	83.33902	-1.664227642	0.65	-2.314227642	0		
5	1998	84.2601626	83.33902	0.921138211	0.65	0.271138211	0.271138211		
6	1999	83.35772358	83.33902	0.018699187	0.65	-0.631300813	0		
7	2000	84.03252033	83.33902	0.693495935	0.65	0.043495935	0.043495935		
8	2001	81.55284553	83.33902	-1.786178862	0.65	-2.436178862	0		
9	2002	83.58536585	83.33902	0.246341463	0.65	-0.403658537	0		
10	2003	81.4796748	83.33902	-1.859349593	0.65	-2.509349593	0		
11	2004	81.76422764	83.33902	-1.574796748	0.65	-2.224796748	0		
12	2005	83.35772358	83.33902	0.018699187	0.65	-0.631300813	0		
13	2006	83.04878049	83.33902	-0.290243902	0.65	-0.940243902	0		
14	2007	85.39837398	83.33902	2.059349593	0.65	1.409349593	1.409349593		
15	2008	82.51219512	83.33902	-0.826829268	0.65	-1.476829268	0		
16	2009	80.99186992	83.33902	-2.347154472	0.65	-2.997154472	0		
17	2010	87.21138211	83.33902	3.872357724	0.65	3.222357724	3.222357724		
18	2011	85.27642276	83.33902	1.937398374	0.65	1.287398374	4.509756098	<=Threshold=4	
19	2012	84.6504065	83.33902	1.311382114	0.65	0.661382114	5.171138211		
20	2013	81.66666667	83.33902	-1.672357724	0.65	-2.322357724	2.848780488		
21	2014	83.94308943	83.33902	0.604065041	0.65	-0.045934959	2.802845528		
22	2015	83.30081301	83.33902	-0.038211382	0.65	-0.688211382	2.114634146		

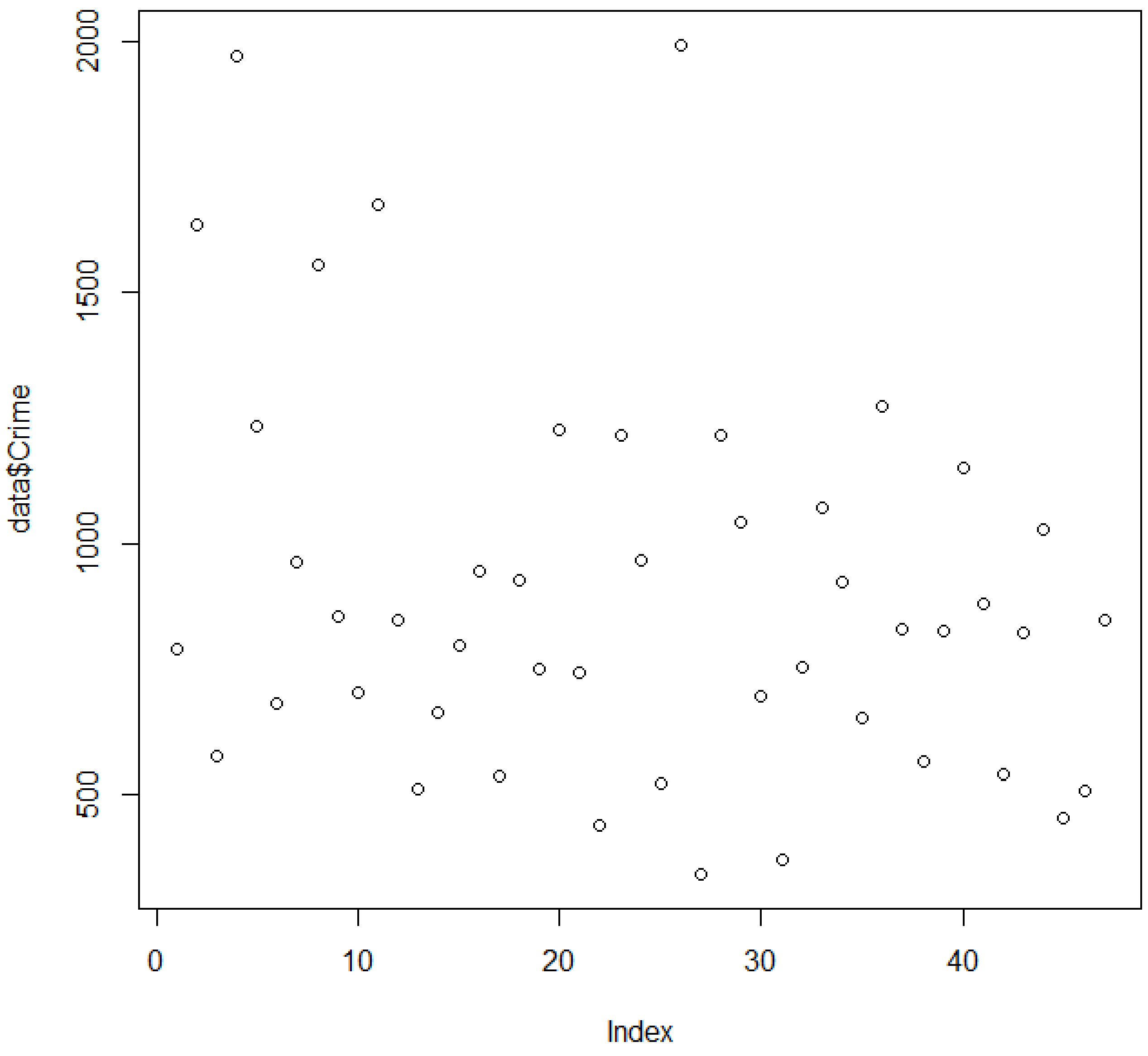
Question 5.1

Input : uscrime.txt

Objective: test to see whether there are any outliers in the last column (number of crimes per 100,000 people). Use the grubbs.test function in the outliers package in R.

A. **Step#1: Visualize outliers**

```
> ###Question 5.1
>
> ###Input : http://www.statsci.org/data/general/uscrime.txt
>
> ####ASK: test to see whether there are any outliers in the last column (# of crimes per 100k)
>
> ####OPTIONS: Use the grubbs.test function in the outliers package in R.
>
>
> ##########CLEAR##########
> rm(list = ls())
> set.seed(100)
>
> ##########LIBRARY##########
> #install.packages("outliers")
> library(outliers)
> data<- read.table("5_1uscrime.txt",header=TRUE,stringsAsFactors = FALSE,sep="\t")
> head(data,10)
      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob    Time Crime
1  15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602 26.2011   791
2  14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599 25.2999  1635
3  14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401 24.3006   578
4  13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801 29.9012  1969
5  14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399 21.2998  1234
6  12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201 20.9995   682
7  12.7  1 11.1  8.2  7.9 0.519  98.2   4 13.9 0.097 3.8   6200 16.8 0.042100 20.6993   963
8  13.1  1 10.9 11.5 10.9 0.542  96.9  50 17.9 0.079 3.5   4720 20.6 0.040099 24.5988  1555
9  15.7  1  9.0  6.5  6.2 0.553  95.5  39 28.6 0.081 2.8   4210 23.9 0.071697 29.4001   856
10 14.0  0 11.8  7.1  6.8 0.632 102.9   7  1.5 0.100 2.4   5260 17.4 0.044498 19.5994   705
>
> #Scatterplot
> plot(data$Crime)
```


data$Crime
2000
1500
1000
500
0
10
20
30
40
Index

B. **Results : There seems to be highest outlier point**

```
>
> # Grubbs test allows to detect whether the highest or lowest value in a dataset is an outlier.
> ###DETECT HIGH OUTLIER###
> Outhigh <- grubbs.test(data$Crime)
> Outhigh

        Grubbs test for one outlier

data:  data$Crime
G = 2.81287, U = 0.82426, p-value = 0.07887
alternative hypothesis: highest value 1993 is an outlier

>
> ## if the p-value <= (α=0.05),then the null hypothesis is rejected
> ## THEN we will conclude that the lowest/highest value is an outlier.
>
> ### p-value >=0.05,null hypothesis is not rejected,
> ### we do not reject the hypothesis that the lowest/highest value is not an outlier.
>
>
> ###DETECT LOW OUTLIER
> Outlow <- grubbs.test(data$Crime,opposite = TRUE)
> Outlow

        Grubbs test for one outlier

data:  data$Crime
G = 1.45589, U = 0.95292, p-value = 1
alternative hypothesis: lowest value 342 is an outlier

>
> dataY=data$Crime
> which(dataY== 1993)
[1] 26
> which(dataY== 342)
[1] 27
>
```

Row#26 holds the highest Outlier point and Row#27 is the lowest Outlier point

```
> head(data,27)
      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob    Time Crime
1  15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602 26.2011   791
2  14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599 25.2999  1635
3  14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401 24.3006   578
4  13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801 29.9012  1969
5  14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399 21.2998  1234
6  12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201 20.9995   682
7  12.7  1 11.1  8.2  7.9 0.519  98.2   4 13.9 0.097 3.8   6200 16.8 0.042100 20.6993   963
8  13.1  1 10.9 11.5 10.9 0.542  96.9  50 17.9 0.079 3.5   4720 20.6 0.040099 24.5988  1555
9  15.7  1  9.0  6.5  6.2 0.553  95.5  39 28.6 0.081 2.8   4210 23.9 0.071697 29.4001   856
10 14.0  0 11.8  7.1  6.8 0.632 102.9   7  1.5 0.100 2.4   5260 17.4 0.044498 19.5994   705
11 12.4  0 10.5 12.1 11.6 0.580  96.6 101 10.6 0.077 3.5   6570 17.0 0.016201 41.6000  1674
12 13.4  0 10.8  7.5  7.1 0.595  97.2  47  5.9 0.083 3.1   5800 17.2 0.031201 34.2984   849
13 12.8  0 11.3  6.7  6.0 0.624  97.2  28  1.0 0.077 2.5   5070 20.6 0.045302 36.2993   511
14 13.5  0 11.7  6.2  6.1 0.595  98.6  22  4.6 0.077 2.7   5290 19.0 0.053200 21.5010   664
15 15.2  1  8.7  5.7  5.3 0.530  98.6  30  7.2 0.092 4.3   4050 26.4 0.069100 22.7008   798
16 14.2  1  8.8  8.1  7.7 0.497  95.6  33 32.1 0.116 4.7   4270 24.7 0.052099 26.0991   946
17 14.3  0 11.0  6.6  6.3 0.537  97.7  10  0.6 0.114 3.5   4870 16.6 0.076299 19.1002   539
18 13.5  1 10.4 12.3 11.5 0.537  97.8  31 17.0 0.089 3.4   6310 16.5 0.119804 18.1996   929
19 13.0  0 11.6 12.8 12.8 0.536  93.4  51  2.4 0.078 3.4   6270 13.5 0.019099 24.9008   750
20 12.5  0 10.8 11.3 10.5 0.567  98.5  78  9.4 0.130 5.8   6260 16.6 0.034801 26.4010  1225
21 12.6  0 10.8  7.4  6.7 0.602  98.4  34  1.2 0.102 3.3   5570 19.5 0.022800 37.5998   742
22 15.7  1  8.9  4.7  4.4 0.512  96.2  22 42.3 0.097 3.4   2880 27.6 0.089502 37.0994   439
23 13.2  0  9.6  8.7  8.3 0.564  95.3  43  9.2 0.083 3.2   5130 22.7 0.030700 25.1989  1216
24 13.1  0 11.6  7.8  7.3 0.574 103.8   7  3.6 0.142 4.2   5400 17.6 0.041598 17.6000   968
25 13.0  0 11.6  6.3  5.7 0.641  98.4  14  2.6 0.070 2.1   4860 19.6 0.069197 21.9003   523
26 13.1  0 12.1 16.0 14.3 0.631 107.1   3  7.7 0.102 4.1   6740 15.2 0.041698 22.1005  1993
27 13.5  0 10.9  6.9  7.1 0.540  96.5   6  0.4 0.080 2.2   5640 13.9 0.036099 28.4999   342
>
```

C. View sorted results

```
> newdata <- data[order(Crime),]
> newdata
      M So   Ed  Pol  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob    Time Crime
27 13.5  0 10.9  6.9  7.1 0.540  96.5   6  0.4 0.080 2.2   5640 13.9 0.036099 28.4999   342
31 14.0  0  9.3  5.5  5.4 0.535 104.5   6  2.0 0.135 4.0   4530 20.0 0.041999 21.7998   373
22 15.7  1  8.9  4.7  4.4 0.512  96.2  22 42.3 0.097 3.4   2880 27.6 0.089502 37.0994   439
45 13.9  1  8.8  4.6  4.1 0.480  96.8  19  4.9 0.135 5.3   4570 24.9 0.056202 32.5996   455
46 12.6  0 10.4 10.6  9.7 0.599  98.9  40  2.4 0.078 2.5   5930 17.1 0.046598 16.6999   508
13 12.8  0 11.3  6.7  6.0 0.624  97.2  28  1.0 0.077 2.5   5070 20.6 0.045302 36.2993   511
25 13.0  0 11.6  6.3  5.7 0.641  98.4  14  2.6 0.070 2.1   4860 19.6 0.069197 21.9003   523
17 14.3  0 11.0  6.6  6.3 0.537  97.7  10  0.6 0.114 3.5   4870 16.6 0.076299 19.1002   539
42 14.1  0 10.9  5.6  5.4 0.523  96.8   4  0.2 0.107 3.7   4890 17.0 0.088904 12.1996   542
38 13.3  0 10.4  5.1  4.7 0.599 102.4   7  4.0 0.099 2.7   4250 22.5 0.053998 16.6999   566
3  14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401 24.3006   578
35 12.3  0 10.2  9.7  8.7 0.526  94.8 113  7.6 0.124 5.0   5720 15.8 0.020700 37.4011   653
14 13.5  0 11.7  6.2  6.1 0.595  98.6  22  4.6 0.077 2.7   5290 19.0 0.053200 21.5010   664
6  12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201 20.9995   682
30 16.6  1  8.9  5.8  5.4 0.521  97.3  46 25.4 0.072 2.6   3960 23.7 0.075298 28.3011   696
10 14.0  0 11.8  7.1  6.8 0.632 102.9   7  1.5 0.100 2.4   5260 17.4 0.044498 19.5994   705
21 12.6  0 10.8  7.4  6.7 0.602  98.4  34  1.2 0.102 3.3   5570 19.5 0.022800 37.5998   742
19 13.0  0 11.6 12.8 12.8 0.536  93.4  51  2.4 0.078 3.4   6270 13.5 0.019099 24.9008   750
32 12.5  0 10.9  9.0  8.1 0.586  96.4  97  8.2 0.105 4.3   6170 16.3 0.042698 30.9014   754
1  15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602 26.2011   791
15 15.2  1  8.7  5.7  5.3 0.530  98.6  30  7.2 0.092 4.3   4050 26.4 0.069100 22.7008   798
43 16.2  1  9.9  7.5  7.0 0.522  99.6  40 20.8 0.073 2.7   4960 22.4 0.054902 31.9989   823
39 14.9  1  8.8  6.1  5.4 0.515  95.3  36 16.5 0.086 3.5   3950 25.1 0.047099 27.3004   826
37 17.7  1  8.7  5.8  5.6 0.638  97.4  24 34.9 0.076 2.8   3820 25.4 0.045198 31.6995   831
12 13.4  0 10.8  7.5  7.1 0.595  97.2  47  5.9 0.083 3.1   5800 17.2 0.031201 34.2984   849
47 13.0  0 12.1  9.0  9.1 0.623 104.9   3  2.2 0.113 4.0   5880 16.0 0.052802 16.0997   849
9  15.7  1  9.0  6.5  6.2 0.553  95.5  39 28.6 0.081 2.8   4210 23.9 0.071697 29.4001   856
41 14.8  0 12.2  7.2  6.6 0.601  99.8   9  1.9 0.084 2.0   5900 14.4 0.025100 30.0001   880
34 12.6  0 11.8  9.7  9.7 0.542  99.0  18  2.1 0.102 3.5   5890 16.6 0.040799 21.6997   923
18 13.5  1 10.4 12.3 11.5 0.537  97.8  31 17.0 0.089 3.4   6310 16.5 0.119804 18.1996   929
16 14.2  1  8.8  8.1  7.7 0.497  95.6  33 32.1 0.116 4.7   4270 24.7 0.052099 26.0991   946
7  12.7  1 11.1  8.2  7.9 0.519  98.2   4 13.9 0.097 3.8   6200 16.8 0.042100 20.6993   963
24 13.1  0 11.6  7.8  7.3 0.574 103.8   7  3.6 0.142 4.2   5400 17.6 0.041598 17.6000   968
44 13.6  0 12.1  9.5  9.6 0.574 101.2  29  3.6 0.111 3.7   6220 16.2 0.028100 30.0001  1030
29 11.9  0 10.7 16.6 15.7 0.521  93.8 168  8.9 0.092 3.6   6370 15.4 0.023400 36.7009  1043
33 14.7  1 10.4  6.3  6.4 0.560  97.2  23  9.5 0.076 2.4   4620 23.3 0.049499 25.5005  1072
40 14.5  1 10.4  8.2  7.4 0.560  98.1  96 12.6 0.088 3.1   4880 22.8 0.038801 29.3004  1151
23 13.2  0  9.6  8.7  8.3 0.564  95.3  43  9.2 0.083 3.2   5130 22.7 0.030700 25.1989  1216
28 15.2  0 11.2  8.2  7.6 0.571 101.8  10  7.9 0.103 2.8   5370 21.5 0.038201 25.8006  1216
20 12.5  0 10.8 11.3 10.5 0.567  98.5  78  9.4 0.130 5.8   6260 16.6 0.034801 26.4010  1225
5  14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399 21.2998  1234
36 15.0  0 10.0 10.9  9.8 0.531  96.4   9  2.4 0.087 3.8   5590 15.3 0.006900 44.0004  1272
8  13.1  1 10.9 11.5 10.9 0.542  96.9  50 17.9 0.079 3.5   4720 20.6 0.040099 24.5988  1555
2  14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599 25.2999  1635
11 12.4  0 10.5 12.1 11.6 0.580  96.6 101 10.6 0.077 3.5   6570 17.0 0.016201 41.6000  1674
4  13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801 29.9012  1969
26 13.1  0 12.1 16.0 14.3 0.631 107.1   3  7.7 0.102 4.1   6740 15.2 0.041698 22.1005  1993
> D
```

D. **Results: the highest outlier is 1993 with the next closet point in the scatterplot noted by 1969**

Question 6.1 **Describe a situation or problem from your job, everyday life, current events, etc., for which a Change Detection model would be appropriate. Applying the CUSUM technique, how would you choose the critical value and the threshold?**

While listening to the tutorial for CUSUM process, the very first situation that I could relate is with an ongoing initiative at work for COVID Vaccination.
To encourage vaccination with our workforce, our organization has introduced a random drawing every couple of week that will select 10 winners and award them $2000.
To enroll in this program, we had to register our vaccination details to HR for management to understand if there is any positive impact in motivating employees towards vaccination.

For example, we have 10,000 employees and before this initiative, we have less 1% employee vaccination records that organization holds.
Organization was able to obtain more visibility on vaccination entries after this initiative. We can use historical data along with the time series data collected after the initiative.
Using CUSUM, we can detect the increase in employee vaccination trending over time.

<u>Selecting T & C:</u>
Considering goal of the organization as 50% for employee vaccination, I will set this as threshold “T’

Coming to C value, between tradeoff on False positive vs. Late detection, I will choose late detection and hence I will start with C=0 and choose a <u>BIGGER value for “C”</u> to prefer a <u>less sensitive CUSUM program</u>.