Question 5.1

Using crime data from the file `uscrime.txt` (http://www.statsci.org/data/general/uscrime.txt, description at http://www.statsci.org/data/general/uscrime.html), 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.

Answer 5.1

```
# Clear global environment, load package, load and preview dataset
> rm(list = ls())
> install.packages("outliers")
> library(outliers)
> df1 <- read.table("/Users/…/uscrime.txt", header = T, stringsAsFactors = F)
> head(df1)
    M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
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
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
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
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
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
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
    Time Crime
1 26.2011   791
2 25.2999  1635
3 24.3006   578
4 29.9012  1969
5 21.2998  1234
6 20.9995   682

# Conduct grubbs test for two outliers on opposite tails, so type is set to 11. Since the goal is to
test whether there are any test points in the "Crime" column that falls in the category of
outliers, therefore opposite set to false in order to check the largest difference between the
test point and the mean. According to the results, p-value equals to 1, indicating at least one of
the max/min falls within the expected range of standard deviation and hence it is not outlier in
the "Crime" column.
> grubbs.test(df1[, 16], type = 11, opposite = F, two.sided = F)

        Grubbs test for two opposite outliers

data:  df1[, 16]
G = 4.26877, U = 0.78103, p-value = 1
alternative hypothesis: 342 and 1993 are outliers
```

To further verify if the test points contain an outlier, type is set to 10. As the p-value equals to 0.079 with threshold set to 0.05, result indicates the highest value is not an outlier in the "Crime" column.

```
> grubbs.test(df1[, 16], type = 10, opposite = F, two.sided = F)

        Grubbs test for one outlier

data:  df1[, 16]
G = 2.81287, U = 0.82426, p-value = 0.07887
alternative hypothesis: highest value 1993 is an outlier
```

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?

Answer 6.1

Working in a manufacturing company, the measurement of process capability index is crucial to determine the stability of manufacturing process. Change Detection model would be applied to determine the production stability by monitoring the changes in mean of measurement from inspection machine. When outliers are detected, corrective action can be put in place to maintain production stability. Critical value would be the ideal dimension from the design specification of the product, and the threshold would be the maximum and minimum dimension tolerance of the design specification in order to produce products with satisfactory quality.

Question 6.2

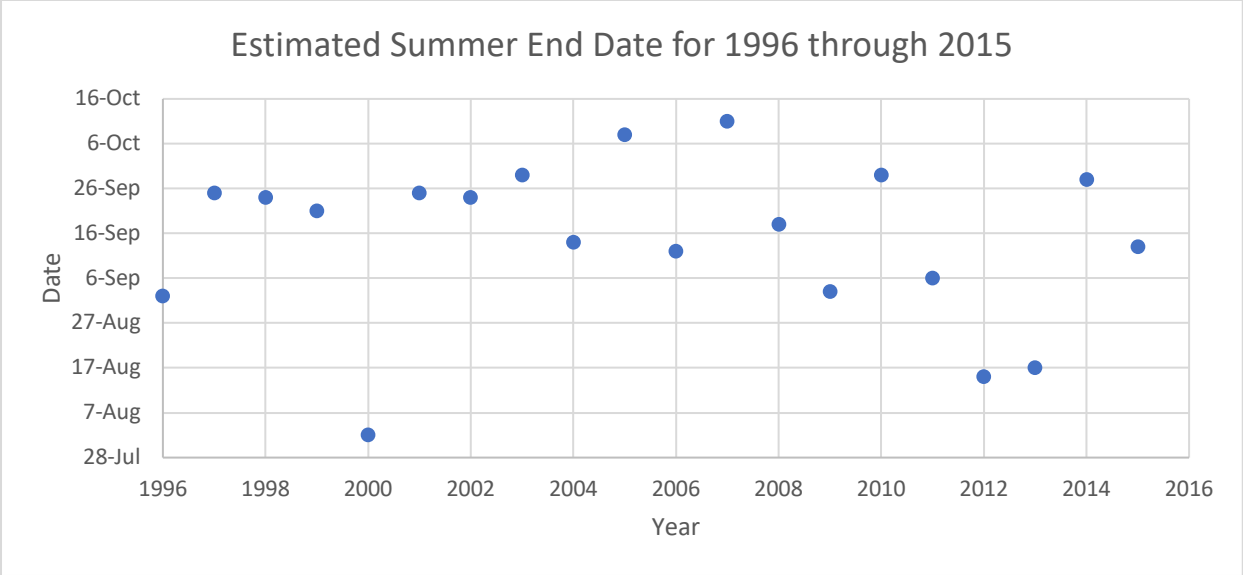
1. Using July through October daily-high-temperature data for Atlanta for 1996 through 2015, use a CUSUM approach to identify when unofficial summer ends (i.e., when the weather starts cooling off) each year. You can get the data that you need from the file `temps.txt` or online, for example at http://www.iweathernet.com/atlanta-weather-records 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).

Answer 6.2 (1)

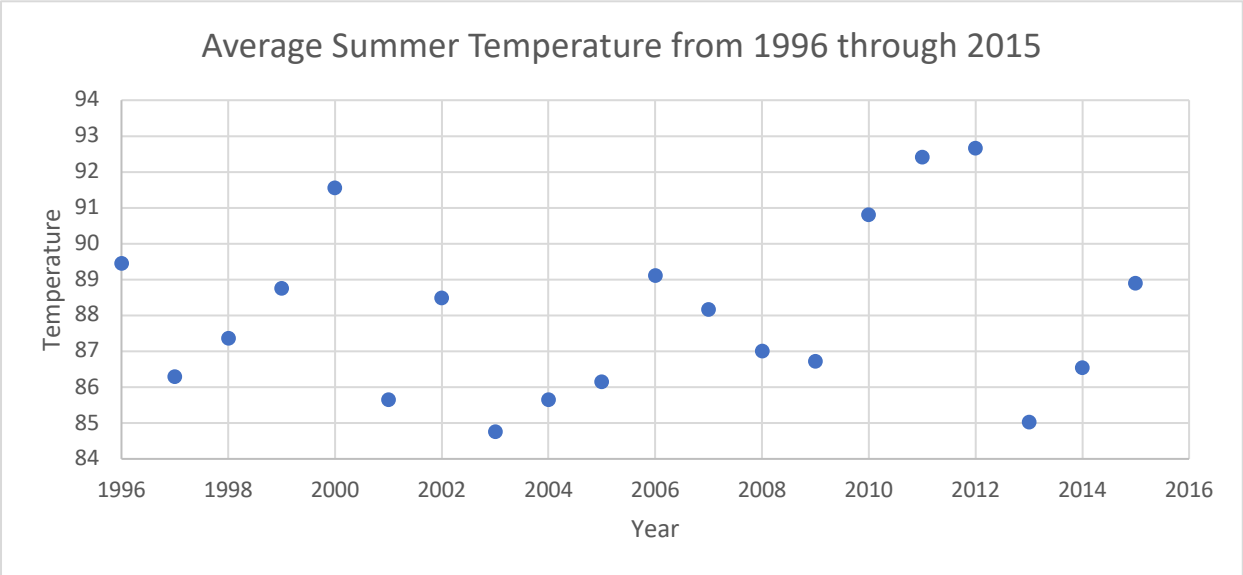
After importing temps.txt into 'sheet temps.txt', I calculated the standard deviation of temperature in 'sheet 6.2.1' for each day each year from 1996 through 2015 to observe the temperature distribution in order to determine when summer ends. Standard deviation was calculated using mean for July as a benchmark to observe the changes of temperature since

July. After using various C and T value, a 4 and 30 for C and T value respectively yielded a better summer end date distribution, which is around mid-September for each year. Following plot shows the summer end date distribution for 1996 through 2015.



Answer 6.2 (2)

In order to determine whether summer climate gets warmer, I decided to calculate the average of summer temperature, with summer end date determined by the previous question 6.2.1. The following plot is the average summer temperature distribution for each year. The plot trend did not show a significant change in average temperature, with temperature fluctuates from 2010 to 2013. With C value set to 1, the CUSUM result shows changes in year 2000, 2010 through 2013. When C value is set to 4, change is observed only in year 2012.

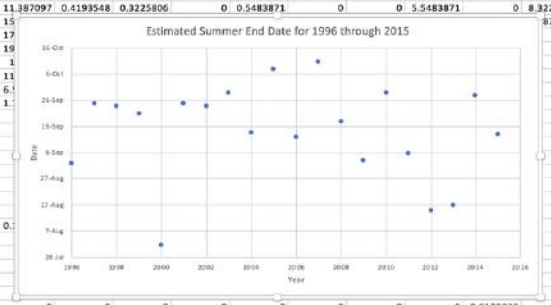


Appendix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	DAY	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	
2	1-Jul	98	86	91	84	89	84	90	73	82	91	93	95	85	95	87	92	105	82	90	85	
3	2-Jul	97	90	88	82	91	87	90	81	81	89	93	85	87	90	84	94	93	85	93	87	
4	3-Jul	97	93	91	87	93	87	87	87	86	86	93	82	91	89	83	95	99	76	87	79	
5	4-Jul	90	91	91	88	95	84	89	86	88	86	91	86	90	91	85	92	98	77	84	85	
6	5-Jul	89	84	91	90	96	86	93	80	90	89	90	88	88	80	88	90	100	83	86	84	
7	6-Jul	93	84	89	91	96	87	93	84	90	82	81	87	82	87	89	90	98	83	87	84	
8	7-Jul	93	75	93	82	96	87	89	87	89	76	80	82	88	86	94	94	93	79	89	90	
9	8-Jul	91	87	95	86	91	89	89	90	87	88	82	82	90	82	97	94	95	88	90	90	
10	9-Jul	93	84	95	87	96	91	90	89	88	89	84	89	89	84	96	91	97	88	90	91	
11	10-Jul	93	87	91	87	99	87	91	84	89	78	84	86	87	84	90	92	95	87	87	93	
12	11-Jul	90	84	91	82	96	90	84	84	90	83	90	85	89	86	93	95	90	80	85	92	
13	12-Jul	91	88	86	77	93	90	77	86	89	86	91	87	93	90	90	95	84	87	90	93	
14	13-Jul	93	86	88	73	91	86	82	87	91	84	91	86	85	84	91	97	90	78	89	92	
15	14-Jul	93	90	87	81	93	82	88	84	91	87	91	84	88	89	91	90	90	85	90	90	
16	15-Jul	82	91	91	81	93	82	91	86	84	84	91	81	89	89	94	80	90	86	86	89	
17	16-Jul	91	91	87	86	93	84	93	88	84	85	91	86	89	90	89	85	92	87	83	88	
18	17-Jul	96	89	90	82	91	87	93	88	84	89	93	89	88	88	87	87	93	91	86	93	
19	18-Jul	95	89	91	87	97	88	93	88	87	90	93	89	90	82	83	89	93	87	82	92	
20	19-Jul	96	89	95	88	100	90	93	88	84	89	96	88	91	80	90	94	91	90	85	91	
21	20-Jul	99	90	91	90	99	87	91	88	88	89	93	86	94	82	91	91	84	86	76	93	
22	21-Jul	91	89	91	90	93	84	95	89	89	90	93	86	95	86	94	92	90	87	82	93	
23	22-Jul	95	84	89	91	96	87	91	86	89	91	91	79	92	84	95	94	95	85	83	92	
24	23-Jul	91	87	91	93	87	90	89	81	93	91	86	82	87	87	97	92	97	84	88	88	
25	24-Jul	93	88	91	93	82	84	87	82	95	90	87	87	88	88	94	92	97	86	87	91	
26	25-Jul	84	89	86	91	75	82	84	84	89	92	88	87	89	90	95	90	98	89	88	90	
27	26-Jul	84	89	88	93	82	88	86	87	87	94	93	87	87	92	95	94	98	86	89	91	
28	27-Jul	82	91	80	93	88	90	89	87	84	92	95	90	90	90	93	94	97	82	92	92	
29	28-Jul	79	91	88	93	91	84	91	89	89	90	96	89	93	89	90	90	97	86	90	94	
30	29-Jul	90	89	89	93	89	89	91	88	87	83	91	87	92	85	94	93	94	86	82	93	
31	30-Jul	91	88	90	97	87	89	88	84	89	78	91	92	90	82	95	96	96	90	84	94	
32	31-Jul	87	72	86	99	86	87	90	88	90	84	94	90	88	85	95	96	88	80	85	93	
33	1-Aug	86	80	86	96	86	84	93	84	91	82	95	92	89	89	96	91	94	87	81	89	
34	2-Aug	90	84	82	93	81	84	91	84	90	86	95	92	92	83	84	96	99	89	84	94	
35	3-Aug	84	88	84	88	84	84	91	84	91	88	97	94	91	90	92	97	94	88	88	94	
36	4-Aug	91	89	86	89	88	86	91	82	91	91	98	97	91	92	95	85	87	90	90	97	
37	5-Aug	93	88	90	91	91	88	93	84	90	88	96	96	92	92	93	96	90	88	89	95	
38	6-Aug	88	84	89	93	91	84	97	82	84	86	89	98	94	89	93	93	86	88	92	88	

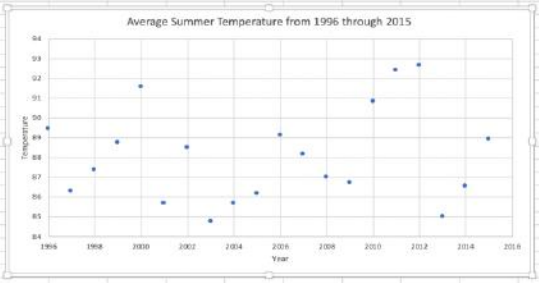
temps.txt | 6.2.1 | 6.2.2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
2	Avg(Jul)	91.193548	87.258065	89.709677	87.645161	91.741935	86.741935	89.258065	85.580645	87.83871	86.935484	90.193548	86.419355	89.16129	86.645161	91.258065	91.935484	94.096774	84.709677	86.612903	90.064516
3	C	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	T	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
5	EndDate	2-Sep	25-Sep	24-Sep	21-Sep	2-Aug	25-Sep	24-Sep	29-Sep	14-Sep	8-Oct	12-Sep	11-Oct	18-Sep	3-Sep	29-Sep	6-Sep	15-Aug	17-Aug	28-Sep	13-Sep
6	1-Jul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	2-Jul	0	0	0	1.6451613	0	0	0	0.5806452	2.8387097	0	0	0	0	0	3.2580645	0	0	0	0	0
8	3-Jul	0	0	0	0	0	0	0	0	0.6774194	0	0	0.4193548	0	0	7.516129	0	0	4.7096774	0	7.0645161
9	4-Jul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9.7741935	0	0	8.4193548	0	8.1290323
10	5-Jul	0	0	0	0	0	0	0	1.5806452	0	0	0	0	0	2.6451613	9.0322581	0	0	6.1290323	0	10.193548
11	6-Jul	0	0	0	0	0	0	0	0	0	0.9354839	5.1935484	0	3.1612903	0	7.2903226	0	0	3.3887097	0	12.258065
12	7-Jul	0	8.2580645	0	1.6451613	0	0	0	0	0	7.8709677	11.387097	0.4193548	0.3225806	0	0.5483871	0	0	5.5483871	0	8.3225806
13	8-Jul	0	4.516129	0	0	0	0	0	0	0	2.8064516	15									870968
14	9-Jul	0	3.7741935	0	0	0	0	0	0	0	0	17									0
15	10-Jul	0	0.0322581	0	0	0	0	0	0	0	4.9354839	19									0
16	11-Jul	0	0	0	1.6451613	0	0	1.2580645	0	0	4.8709677	1									0
17	12-Jul	0	0	0	8.2903226	0	0	9.516129	0	0	1.8064516	11									0
18	13-Jul	0	0	0	18.935484	0	0	12.774194	0	0	0.7419355	6.									0
19	14-Jul	0	0	0	21.580645	0	0.7419355	10.032258	0	0	0	1.									0
20	15-Jul	5.1935484	0	0	24.225806	0	1.483871	4.2903226	0	0	0										0
21	16-Jul	1.3870968	0	0	21.870968	0	0.2258065	0	0	0	0										0
22	17-Jul	0	0	0	23.516129	0	0	0	0	0	0										0
23	18-Jul	0	0	0	20.16129	0	0	0	0	0	0										0
24	19-Jul	0	0	0	15.806452	0	0	0	0	0	0										0
25	20-Jul	0	0	0	9.4516129	0	0	0	0	0	0										0
26	21-Jul	0	0	0	3.0967742	0	0	0	0	0	0										0
27	22-Jul	0	0	0	0	0	0	0	0	0	0										0
28	23-Jul	0	0	0	0	0.7419355	0	0	0.5806452	0	0	0.									0
29	24-Jul	0	0	0	0	6.483871	0	0	0.1612903	0	0										0
30	25-Jul	3.1935484	0	0	0	19.225806	0.7419355	1.2580645	0	0	0										0
31	26-Jul	6.3870968	0	0	0	24.967742	0	0.516129	0	0	0										0
32	27-Jul	11.580645	0	5.7096774	0	24.709677	0	0	0	0	0										0
33	28-Jul	19.774194	0	3.4193548	0	21.451613	0	0	0	0	0										0
34	29-Jul	16.967742	0	0.1290323	0	20.193548	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6129032	0
35	30-Jul	13.16129	0	0	0	20.935484	0	0	0	0	4.9354839	0	0	0	0.6451613	0	0	0	0	0	0
36	31-Jul	13.354839	11.258065	0	0	22.677419	0	0	0	0	3.8709677	0	0	0	0	0	0	2.0967742	0.7096774	0	0
37	1-Aug	14.548387	14.516129	0	0	24.419355	0	0	0	0	4.8064516	0	0	0	0	0	0	0	0	1.6129032	0
38	2-Aug	11.741935	13.774194	3.7096774	0	31.16129	0	0	0	0	1.7419355	0	0	0	0	3.2580645	0	0	0	0.2258065	0
39	3-Aug	14.935484	9.0322581	5.4193548	0	34.903226	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	4-Aug	11.129032	3.2903226	5.1290323	0	34.645161	0	0	0	0	0	0	0	0	0	0	2.9354839	3.0967742	0	0	0



temps.txt | 6.2.1 | 6.2.2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
2	EndDate	2-Sep	25-Sep	24-Sep	21-Sep	2-Aug	25-Sep	24-Sep	29-Sep	14-Sep	8-Oct	12-Sep	11-Oct	18-Sep	3-Sep	29-Sep	6-Sep	15-Aug	17-Aug	28-Sep	13-Sep
3	g(Jul-EndDat	89.4603175	86.3023256	87.3647059	88.7560976	91.5625	85.6511628	88.4823529	84.7555556	85.6533333	86.1515152	89.109589	88.1666667	87	86.71875	90.8111111	92.4179104	92.6666667	85.0212766	86.5393258	88.905405
4	Std	0	0	0	0	1.8125	0	0	0	0	0	0	0	0	0	1.06111111	3.72902156	6.64568823	1.91696482	0	0



temps.txt | 6.2.1 | 6.2.2