Homework 3

Jeff Tilton

9/7/2018

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.

Data

```
library(kableExtra)
library(data.table)
raw = read.table('uscrime.txt', header = TRUE, sep = '\t')
range01 = function(x){(x-min(x))/(max(x)-min(x))}
data = as.data.table(apply(raw, 2, range01))

dims = dim(data)

kable(head(data))
```

M	So	Ed	Po1	Po2	LF	M.F	Pop	NW	U1	
0.5517241	1	0.1142857	0.1074380	0.1293103	0.1863354	0.1167883	0.1818182	0.7102138	0.5277778	0.55
0.4137931	0	0.7428571	0.4793388	0.4655172	0.6397516	0.5693431	0.0606061	0.2375297	0.3611111	0.42
0.3965517	1	0.0571429	0.0000000	0.0258621	0.3291925	0.2554745	0.0909091	0.5154394	0.3333333	0.34
0.2931034	0	0.9714286	0.8595041	0.8620690	0.6024845	0.4379562	0.9333333	0.1852732	0.4444444	0.50
0.3793103	0	0.9714286	0.5289256	0.5172414	0.6894410	0.3722628	0.0909091	0.0665083	0.2916667	0.00
0.0344828	0	0.6571429	0.6033058	0.6379310	0.4161491	0.2189781	0.1333333	0.0997625	0.1944444	0.23

The data is a 47 by 16 table that I have scaled between 0 and 1 by column.

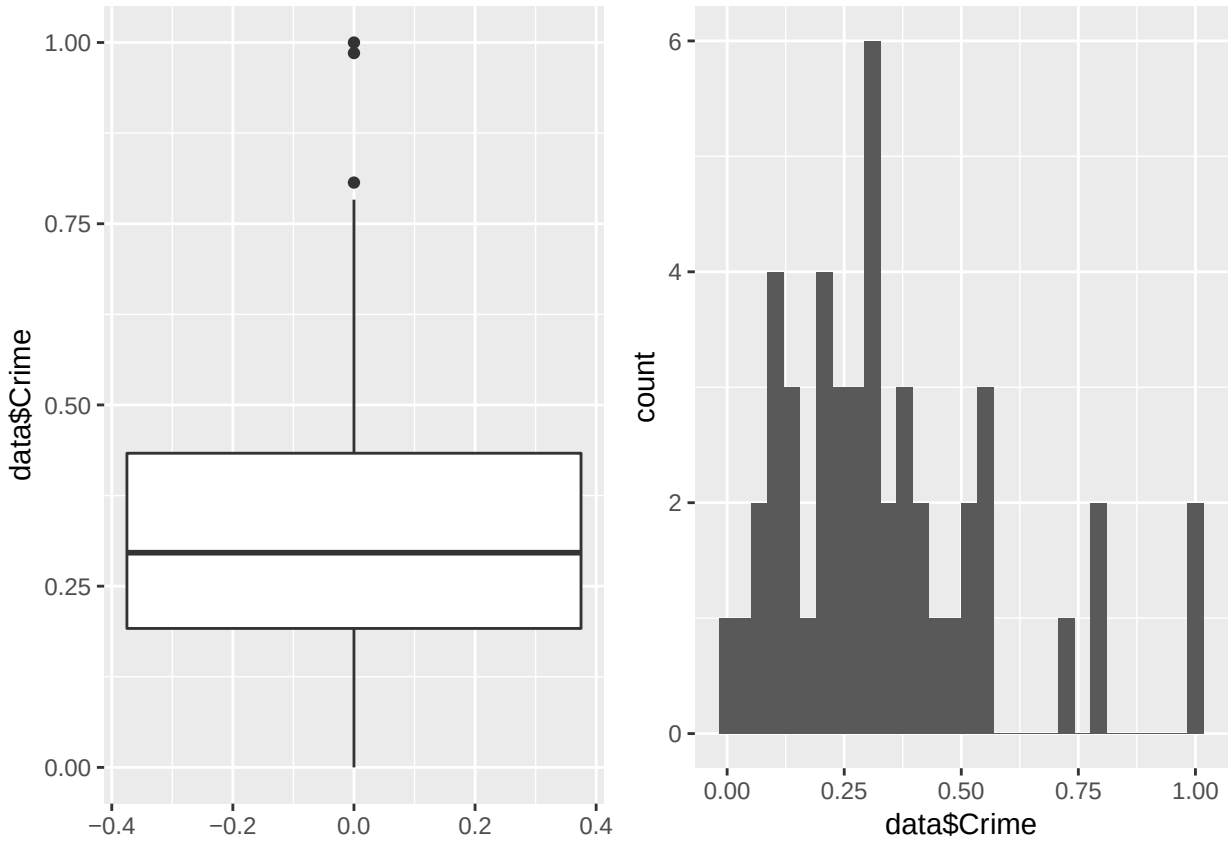
Goal

The goal of this exercise is to find any outliers with the grubbs.test function in the outliers package.

Visualize Data

```
library(gridExtra)
library(ggplot2)
bp = ggplot(data, aes(y = data$Crime)) + geom_boxplot()
hist = ggplot(data=data, aes(data$Crime)) + geom_histogram()
grid.arrange(bp,hist, nrow = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The boxplot and histogram show that there may be an outlier in the dataset. However, it is important to state that no value, no matter how extreme, is necessarily an outlier. The extreme value is part of the dataset and may be critical data in understanding an extrema. The goal of this exercise is to determine if there is an outlier in the statistical sense through hypothesis testing. The grubbs test will let us know if we can reject the null, there are no outliers, and conclude that the most extreme point is a statistical outlier.

grubbs.test Description

Performs Grubbs' test for one outlier, two outliers on one tail, or two outliers on opposite tails, in small sample.

```
library(outliers)
x = data$Crime
type = 10

grubbs.test(x=x,type=type)
```

```
##
##  Grubbs test for one outlier
##
## data:  x
## G = 2.81290, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1 is an outlier
```

Discussion

Although we did not set an alpha value, the p-value is small enough that it is not unreasonable to conclude that an outlier exists. Therefore we can reject the null hypothesis, there is no outlier, and accept the

alternative, the highest value 1 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?

I work as a hydraulic engineer in an office that controls dams on a major US river. Part of the job is to control how much a dam spills. Although a dam does not produce energy when it spills, there is a higher fish survival rate when they go over the dam than throguh the turbines. Spill has an effect on the total dissolved gas (TDG) of the river water, whcih can harm the fish. Therefore spill will go up and down depending on the TDG levels. The CUSUM method can be used to determine if a particular action (spill reduction) has improved the TDG levels in the river. Determining the critical value and threchold would be determined through a partnership with fish biologists and water quality experts.

Question 6.2

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.

Data

```
temps = data.frame(read.table('temps.txt', header = TRUE, sep = '\t'))
names(temps) = append(c('DAY'),as.character(c(1996:2015)))
yearly_mean = data.frame(year = 1996:2015, temp = colMeans(temps[,2:length(temps)]))
temps$mean = rowMeans(temps[,2:length(temps)])

dims = dim(temps)
kable(head(temps))
```

DAY	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	201
1-Jul	98	86	91	84	89	84	90	73	82	91	93	95	85	95	87	9
2-Jul	97	90	88	82	91	87	90	81	81	89	93	85	87	90	84	9
3-Jul	97	93	91	87	93	87	87	87	86	86	93	82	91	89	83	9
4-Jul	90	91	91	88	95	84	89	86	88	86	91	86	90	91	85	9
5-Jul	89	84	91	90	96	86	93	80	90	89	90	88	88	80	88	9
6-Jul	93	84	89	91	96	87	93	84	90	82	81	87	82	87	89	9

Goal

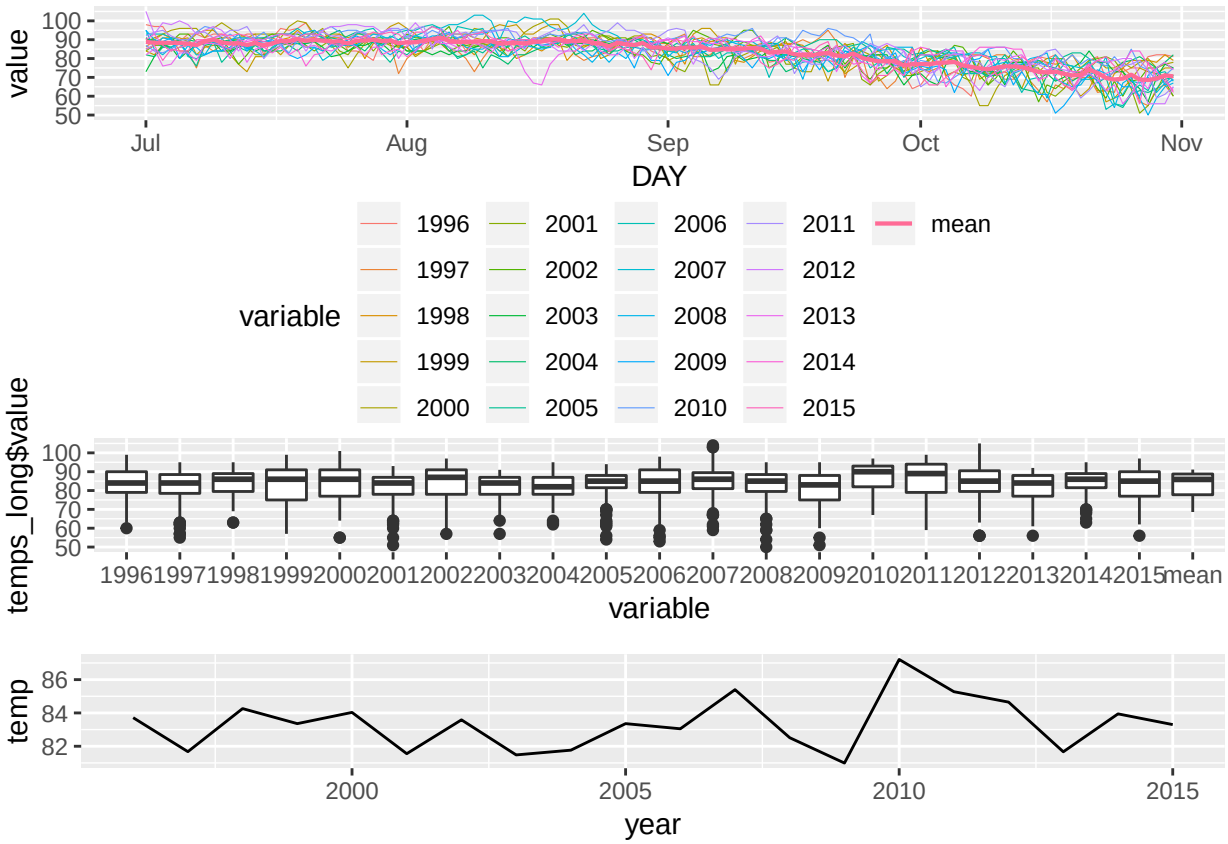
The goal of this exercise is to use a CUSUM approach to identify when unofficial summer ends.

Visualize Data

```
library(tidyr)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
get_legend<-function(myggplot){
  tmp <- ggplot_gtable(ggplot_build(myggplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)
}


temps_long = melt(temps, id = 'DAY')
temps_long$DAY = as.Date(temps_long$DAY, "%d-%B")
size = c(rep(.2, length(temps)-2))
size[length(temps)-1] =.75

p = ggplot(data = temps_long,
           aes(x = DAY, y = value, colour = variable, size =variable)) +
            geom_line() +
            scale_size_manual(values=size) +
            theme(legend.position="bottom")
legend = get_legend(p)
p = p + theme(legend.position="none")

p_yearly = ggplot(data = yearly_mean,
           aes(x = year, y = temp)) +
            geom_line()


bp = ggplot(temps_long, aes(x = variable, y = temps_long$value)) + geom_boxplot()


grid.arrange(p,legend,bp,p_yearly, ncol=1)
```



CUMSUM

$$S_t = max\{0, S_{t-1} + (\mu - X_t - C)\}$$

Looking at the plot of temperatures through the season, the mean of all years looks to drop in mid August. I plan on running the CUSUM algorithm on the mean data. Because the C and t values are arbitrary, I will try and find a combination that gives me a change around mid August for the mean and see how well that works for the other years.

```
library(tidyr)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
s = temps[2]


cusum = function(s, C, decrease = FALSE, MEAN = FALSE){
  x = 1
  if (decrease){
    x = -1
  }
```

```
  if (MEAN){
    s_mean = MEAN
  }else{
    s_mean = sum(s)/length(s)
  }
  s_t = 0
  l = c(s_t)
  for(t in 2:length(s)){
    s_t = max(0, s_t + (s[t]*x - s_mean*x - C))
    l[t] = s_t
  }
  return(l)
}


t = 2
C = 3
MEAN = sum(temps$mean)/nrow(temps)

change = apply(temps[2:length(temps)], 2, cusum, C= C, decrease = TRUE, MEAN = MEAN) %>%
        as.data.frame
change$DAY = temps$DAY


change_day = as.data.frame(change[1:length(change)-1]>t)
get_true = function(v){
  return(min(which(v == TRUE)))
}

index = apply(change_day, 2, get_true)
days = change$DAY[index]
year = c(as.character(seq(1996, 2015, 1)))
year[length(year)+1] = 'mean'
change_df = data.frame(year = year, days = days)



# Use a rolling mean on the s_t values
#to see that summer has truly ended and
#that a cold front hadn't just come through

window = 10

roll = rollmean(
  as.zoo(change[1:length(change)-1]),
  window, align = 'center')

roll = roll>t
roll$day = as.vector(change$DAY)
roll = drop_na(as.data.frame(roll))


roll_index = apply(roll[1:length(roll)-1], 2, get_true)
```

```
roll_days = roll$day[index]
roll_df = data.frame(year = year, days = roll_days)



df = data.frame(year = roll_df$year, change = change_df$days, roll = roll_df$days)


kable(df)
```

year	change	roll
1996	2-Sep	6-Sep
1997	7-Jul	11-Jul
1998	3-Sep	7-Sep
1999	12-Jul	16-Jul
2000	25-Jul	29-Jul
2001	2-Sep	6-Sep
2002	12-Jul	16-Jul
2003	6-Sep	10-Sep
2004	10-Aug	14-Aug
2005	7-Jul	11-Jul
2006	13-Sep	17-Sep
2007	16-Sep	20-Sep
2008	23-Aug	27-Aug
2009	28-Aug	1-Sep
2010	26-Sep	30-Sep
2011	5-Sep	9-Sep
2012	4-Sep	8-Sep
2013	3-Jul	7-Jul
2014	20-Jul	24-Jul
2015	30-Aug	3-Sep
mean	26-Sep	30-Sep

Discussion

I ran several combinations of C and t, but I was not able to get the mean to show a change in mid august even with C and t set to 0. I then settled on finding a C, t combination that gave me days around the August September range. There were still quite a few July values as well, however.

I next decided to use a rolling average on the s_t values as well. The reason is I thought a cold front could have moved in for a day or two, maybe a rain storm, but not truly an end of summer. This did not have a dramatic effect on any of the years, so maybe the first day value is acceptable.

Lastly, I decided to use the mean of all years, which might be considered the expected value of this dataset and ran the CUSUM again. This gave me a result that fit with what I would expect. There were more Augusts and Septembers, fewer July's. I understand that I would not have this value oahead of time for the majority of these dates, but I could use it moving forward.

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

```
t = 0
C = 1
```

```
kable(data.frame(year = year[1:20], change  = cusum(yearly_mean$temp, C=C) > t))
```

year	change
1996	FALSE
1997	FALSE
1998	FALSE
1999	FALSE
2000	FALSE
2001	FALSE
2002	FALSE
2003	FALSE
2004	FALSE
2005	FALSE
2006	FALSE
2007	TRUE
2008	FALSE
2009	FALSE
2010	TRUE
2011	TRUE
2012	TRUE
2013	TRUE
2014	TRUE
2015	TRUE

Discussion

I gave a very low threshold for t and c on this exercise because I felt yearly temperatures are going to be much more stable than seasonal ones. It looks like yearly temperatures began to increase in earnest starting in 2010.