Homework Week 3a

2022-09-08

Homework 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.

> Using grubbs.test, we get a p value of .079. As this is not below the standard threshold of .05, we can accept the null hypothesis that there are no outliers in our data set that are statistically significant.

Analysis

First, lets load in the data and look at the structure

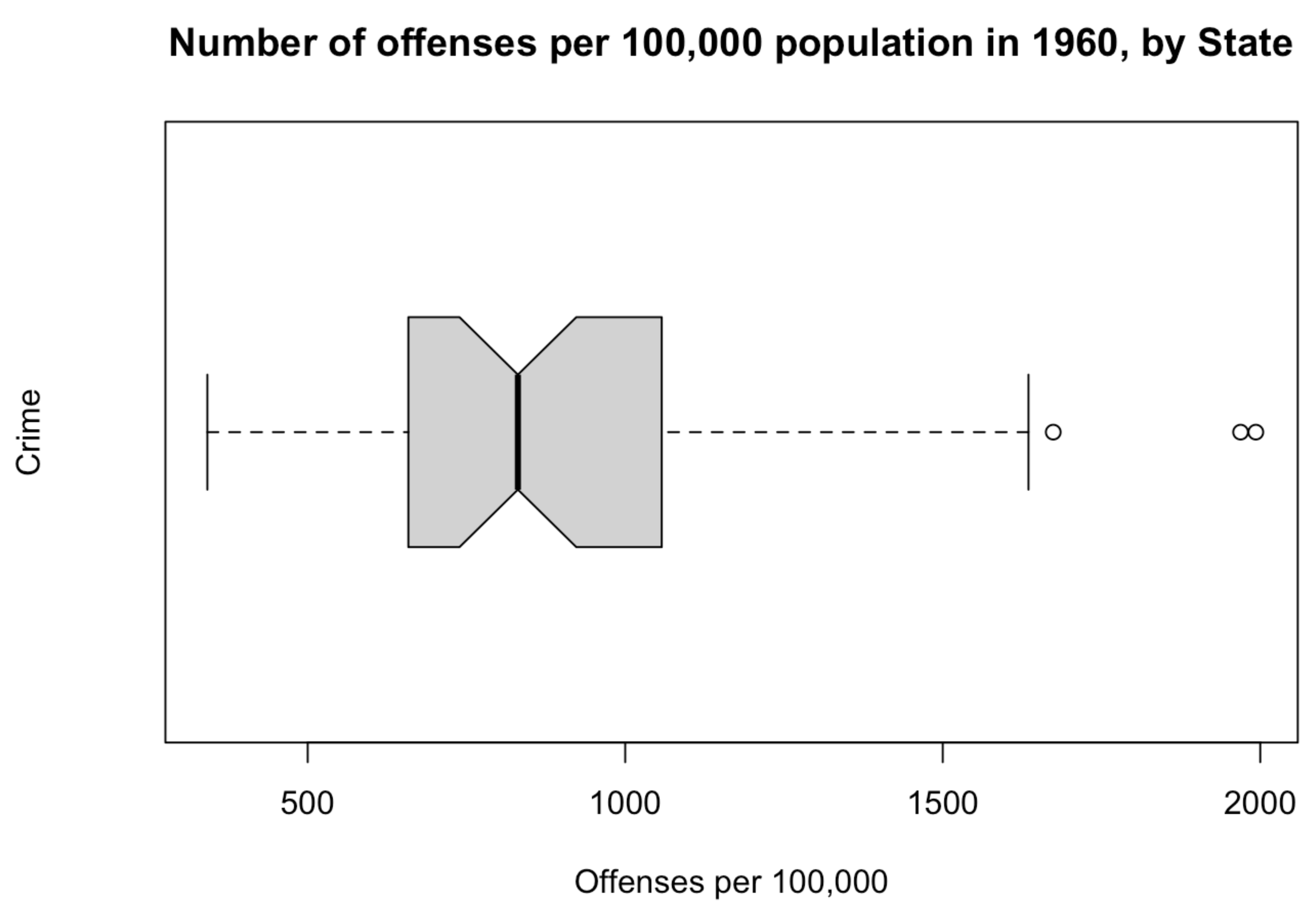
```
library(outliers)
```

```
crime <- read.csv('uscrime.txt', sep = "\t")
summary(crime)
```

```
##        M               So               Ed             Pol
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80
##  Mean   :13.86   Mean   :0.3404   Mean   :10.56   Mean   : 8.50
##  3rd Qu.:14.60   3rd Qu.:1.0000   3rd Qu.:11.45   3rd Qu.:10.45
##  Max.   :17.70   Max.   :1.0000   Max.   :12.20   Max.   :16.60
##       Po2               LF              M.F              Pop
##  Min.   : 4.100   Min.   :0.4800   Min.   : 93.40   Min.   :  3.00
##  1st Qu.: 5.850   1st Qu.:0.5305   1st Qu.: 96.45   1st Qu.: 10.00
##  Median : 7.300   Median :0.5600   Median : 97.70   Median : 25.00
##  Mean   : 8.023   Mean   :0.5612   Mean   : 98.30   Mean   : 36.62
##  3rd Qu.: 9.700   3rd Qu.:0.5930   3rd Qu.: 99.20   3rd Qu.: 41.50
##  Max.   :15.700   Max.   :0.6410   Max.   :107.10   Max.   :168.00
##        NW              U1                U2            Wealth
##  Min.   : 0.20   Min.   :0.07000   Min.   :2.000   Min.   :2880
##  1st Qu.: 2.40   1st Qu.:0.08050   1st Qu.:2.750   1st Qu.:4595
##  Median : 7.60   Median :0.09200   Median :3.400   Median :5370
##  Mean   :10.11   Mean   :0.09547   Mean   :3.398   Mean   :5254
##  3rd Qu.:13.25   3rd Qu.:0.10400   3rd Qu.:3.850   3rd Qu.:5915
##  Max.   :42.30   Max.   :0.14200   Max.   :5.800   Max.   :6890
##       Ineq            Prob              Time            Crime
##  Min.   :12.60   Min.   :0.00690   Min.   :12.20   Min.   : 342.0
##  1st Qu.:16.55   1st Qu.:0.03270   1st Qu.:21.60   1st Qu.: 658.5
##  Median :17.60   Median :0.04210   Median :25.80   Median : 831.0
##  Mean   :19.40   Mean   :0.04709   Mean   :26.60   Mean   : 905.1
##  3rd Qu.:22.75   3rd Qu.:0.05445   3rd Qu.:30.45   3rd Qu.:1057.5
##  Max.   :27.60   Max.   :0.11980   Max.   :44.00   Max.   :1993.0
```

Now, lets do some exploratory data analysis and see if there any data points that visually appear to be outliers.

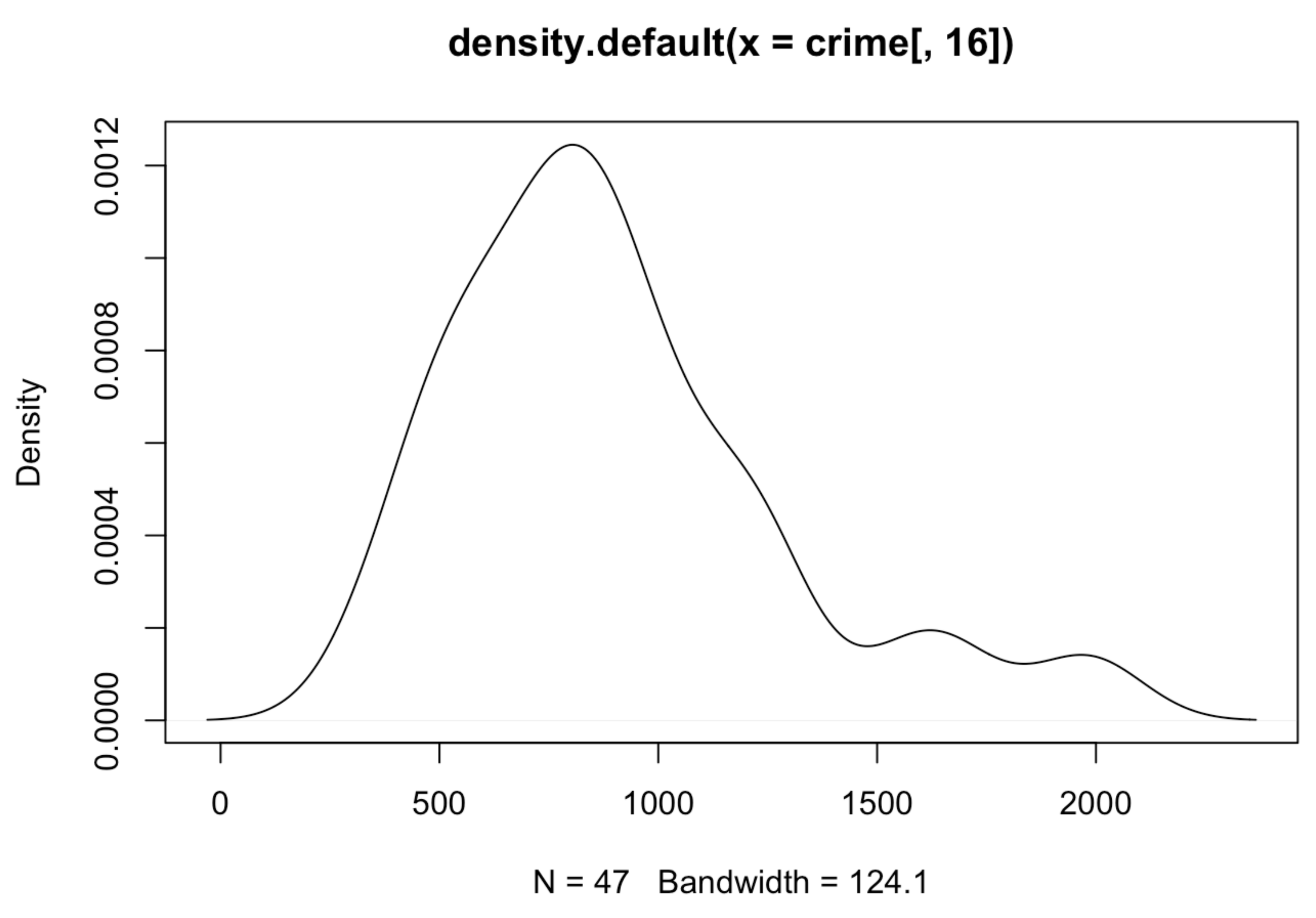
```
boxplot(crime[,16],
        main = "Number of offenses per 100,000 population in 1960, by State",
        notch = TRUE,
        ylab = "Crime",
        xlab = "Offenses per 100,000",
        horizontal = TRUE)
```



It does seem that there are several points that could be outliers, as there are three points that exceed the third quartile of other data points.

Now, lets do some more statistical measurements to see if these are statistically significant outliers. I'd like to use grubbs test, but first we need to exam the data to see if it is "psuedo normal", i.e. see if it follows a normal distribution even remotely, as that is a requirement of grubbs test.

```
d <- density(crime[,16])
plot(d)
```



Plotted on a density map, our data does appear to have a normal distribution with a slight right skew. However, it is normal enough for grubbs test.

First, lets look at outliers on the right:

```
grubbs.test(crime[,16], type = 10)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime[, 16]
## G = 2.81287, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
```

Our p value is greater than .05, thus we accept the null hypothesis that there are no outliers that are statistically significant.

```
grubbs.test(crime[,16], , opposite = TRUE)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime[, 16]
## G = 1.45589, U = 0.95292, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
```

On the left, there is nothing to consider as an outlier as the p value is 1.

Homework 3b

Chris Messer

2022-09-11

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?

> In my group at work, we are responsible for monitoring the engineering systems, and resolve issues when they happen. One such issue is the failure off the systems to generate an invoice. This can happen for a variety of reasons, most of the time due to one off issues with the customer's account. However, on occasion, we will see an issue where there is a broader issue impacting many customer accounts, and invoices will start to fail at a much higher rate.
>
> A change detection model would be useful for determining when a broader issue is occurring, and notify the appropriate department leads. For the threshold value, I would use the number of analysts in the group * the average number of tickets closed a day * 5. This would mean as long as the analysts can handle all the ticket volume in one business week, there is no need to sound an alarm.
>
> For the critical value, I would use 0. There is no rationale to dampen the ticket volume, and it is better to be more cautious here and sound the alarm early vs later.

Question 6.2a

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.

> I opted to use R for this exercise, as I am an expert in excel and fundamentally would like to practice in R. Using a CUSUM method, I determined that the unofficial summer end is September 16th. See "Analysis" for assumptions used.

Question 6.2b

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

> Using a CUSUM model, I have determined that starting in 2006, it is becoming hotter in the summer time in Atlanta.

Analysis

First, lets get our bearings with just building a CUSUM model for one year.

First, lets load the data. Since it looks like our data has the years as column 1 rather than the first year of data, lets reassign the row names to be the years and make 1996 (the first year of data) the first column.

```
library(qcc)
```

```
## Package 'qcc' version 2.7
```

```
## Type 'citation("qcc")' for citing this R package in publications.
```

```
library(outliers)
temps <- read.csv('temps.txt', sep = "\t")
str(temps)
data <- temps[,-1]
rownames(data) <- temps[,1]
```

That's better. Now lets build a CUSUM model for one year. Now, we have some decisions to make. What should our mu value be? Our T (threshold)? Our C (shift)? How do we determine when summer has officially ended, and is not just a cold front that blew in?

Choosing Mu

To find an appropriate value of mu, we first look to the question asked. What is the *unofficial* end date of summer? Since we are unofficially looking for a date, this implies there is an "official" end date of summer! A quick google search tells us in the northern hemisphere, summer officially starts on June 21st and ends on September 22. Since our data set starts on July 1st, we can assume the official summer are the dates the first 84 rows of data. As such, we can conclude the average temperature of summer is the mean temperature during these days.

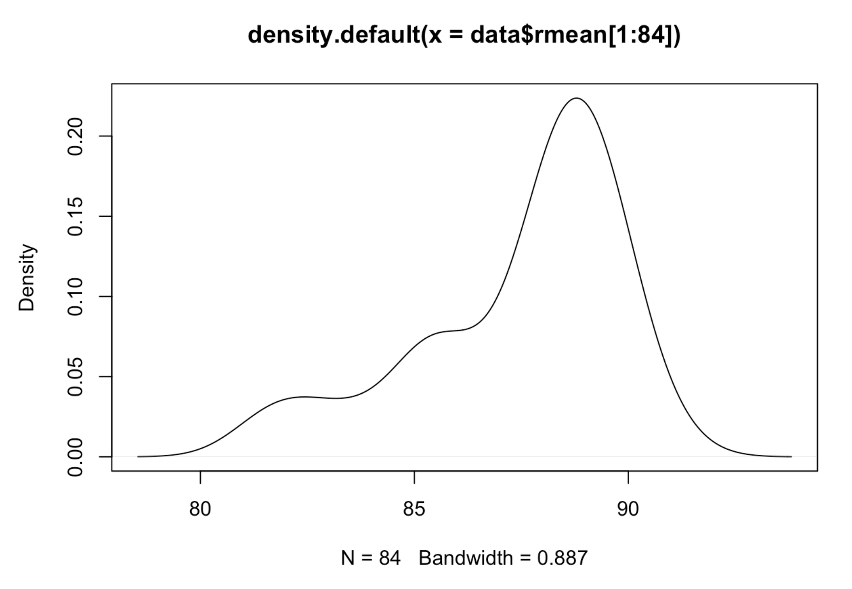
Choosing T

The qcc package makes this a little easier for us. It defaults to a cummulative 5 standard deviations from the mean before considering something as "out of threshold". This value is fine for now- if we see our unofficial summer date varying greatly from the official date, we can revisit this.

Choosing C

Now we must consider- at what point is a summer day *abnormally hot* or *abnormally cold*? Standard deviation is a great fence post for this metric. As long as our data is pseudo normally distributed, 1 standard deviation1 from the mean would encompass 68% of all data points. First, lets look at average temperatures across all years to see if they are normally distributed:

```
data$rmean <- rowMeans(data)
plot(density(data$rmean[1:84]))
```



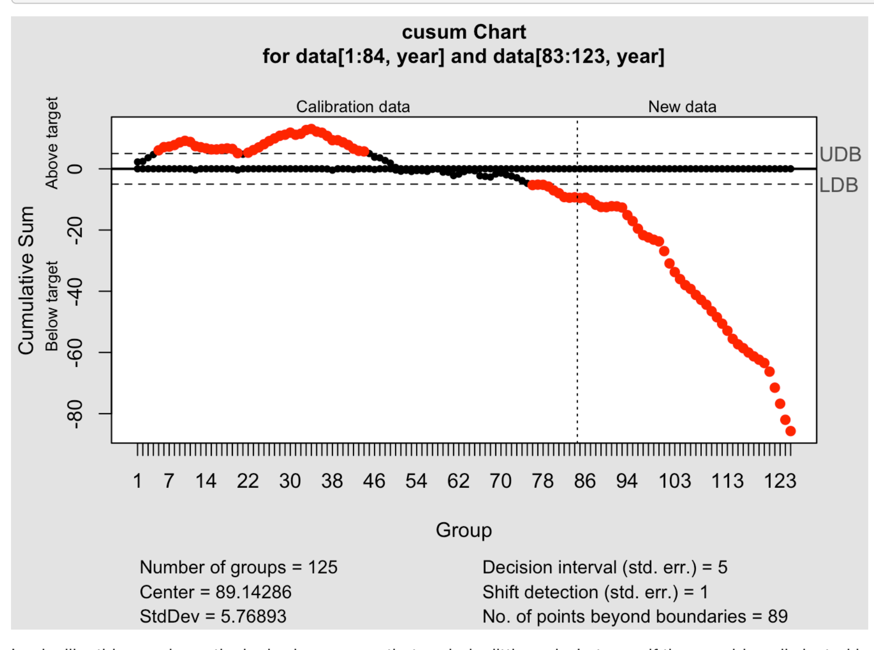
Our data does appear to have a pseudo normal distribution, with a left skew. As such, a C value of 1 standard deviations is appropriate. In other words, we will only add values to our running sum if they exceed 1 standard deviation from the mean for a given year, as a day that is hotter than 67% of other days is pretty warm day and definitely summer weather.

Choosing when summer is unofficially over

We must also quantify when summer is "over". To do this, we need to ignore any outliers in our data, i.e. if a cold front blew in and gave us a statistically significant cold week in the middle of summer. Because these events are unusual but expected, we cannot consider them the "end" of summer. Thus, I propose to only consider the summer over if there are 10 consecutive days of temperatures exceeding the minimum threshold. This works because we will 100% expect the fall to bring temperatures that are consistently below our threshold, and ignores short timespans in the summer below the threshold.

One year of CUSUM

```
year <-17 #choose a random year
summer_mean <- mean(data[1:84,year]) #calculate the mean temperature for our model
summer_sd <-  sd(data[1:84,year]) #calculate the standard deviation of the year
one_year <- cusum(data[1:84,year], std.dev = summer_sd, center = summer_mean, newdata = data[83:123,year], se.shi
ft =1) # generate the model with a C value of 2 std deviations
```



Looks like this year in particular had a summer that ended a little early. Lets see if those cold spells lasted longer than 10 days. We will do some programming tricks here - to find this value, we will check every value in the models lower bound violations, add 10 to it, and if that index is also in the lower bound violations, we can assume it has been 10 consecutive days of cold weather. We will store those days in a vector, then the first value will tell us which day was first to breach for 10 consecutive days. NOTE: this does leave some room for error if the first day breached, had no more breaches for 9 days, then breached again on the 10th day. We will ignore this for this exercise.

```
l <- c()
for (day in one_year$violations$lower){
  if ((day + 10) %in% one_year$violations$lower)
    {
    l<- c(l,day)
  }
}
c(l[[1]], temps[l[[1]],1])
```

```
## [1] "76"     "14-Sep"
```

Looks like summer unofficially ended on the 76th (Sept. 14th) day of our dataset!

All years of CUSUM

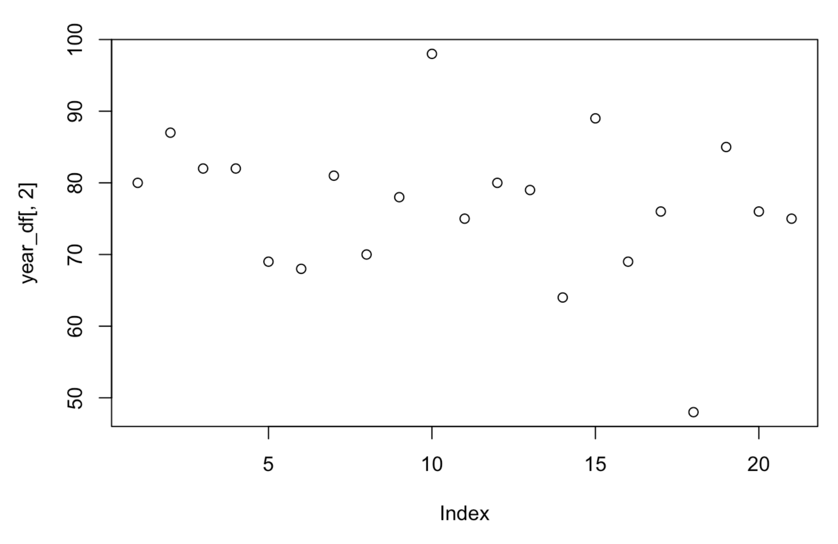
Now, lets do this and loop over all available years and look at our data. We will do this and store the first 10 day breach for each year into a data frame and analyze.

```
first_lower <- c()
day_map <-c()
std_summer <- c()
days_breached <- c()
for (year in seq(ncol(data)))
     {
  summer_mean <- mean(data[1:84,year])
  summer_sd <-  sd(data[1:84,year])
  year_cusum <- cusum(data[1:84,year], std.dev = summer_sd, plot = F, se.shift = 1, newdata = data[85:123,year])

  l <- c()
  for (day in year_cusum$violations$lower){
    if ((day + 10) %in% year_cusum$violations$lower)
      {
      l<- c(l,day)
    }
  }
  first_lower[year] <- l[[1]]
  day_map <- rownames(data[first_lower,])
  std_summer[year] <- summer_sd
  days_breached[year] <- length(year_cusum$violations$upper)
}

year_df <- do.call(rbind, Map(data.frame, year = colnames(data), first_lower_breach = first_lower, day = day_map,
std = std_summer, upper_breach = days_breached))



year_df
```

```
##        year first_lower_breach      day      std upper_breach
## X1996 X1996                 80   18-Sep 5.266874           16
## X1997 X1997                 87   25-Sep 4.675715            0
## X1998 X1998                 82   20-Sep 4.027373           10
## X1999 X1999                 82 20-Sep.1 6.283179           16
## X2000 X2000                 69    7-Sep 7.621073            6
## X2001 X2001                 68    6-Sep 3.758238            2
## X2002 X2002                 81   19-Sep 4.770267            0
## X2003 X2003                 70    8-Sep 3.910502           11
## X2004 X2004                 78   16-Sep 4.873265           17
## X2005 X2005                 98    6-Oct 3.634376            5
## X2006 X2006                 75   13-Sep 5.717822           19
## X2007 X2007                 80 18-Sep.1 6.640727           35
## X2008 X2008                 79   17-Sep 4.533992           11
## X2009 X2009                 64    2-Sep 5.124427           18
## X2010 X2010                 89   27-Sep 3.534925            0
## X2011 X2011                 69  7-Sep.1 6.501544            0
## X2012 X2012                 76   14-Sep 5.768930           39
## X2013 X2013                 48   17-Aug 5.136169            0
## X2014 X2014                 85   23-Sep 3.618576            0
## X2015 X2015                 76 14-Sep.1 5.001707           27
## rmean rmean                 75 13-Sep.1 2.390876           26
```

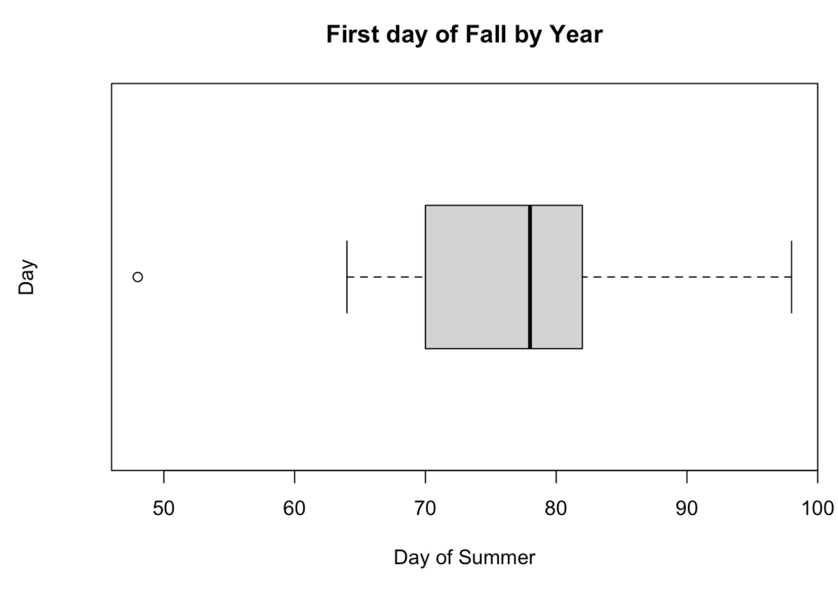
Lets plot these and examine.

```
plot(year_df[,2])
```



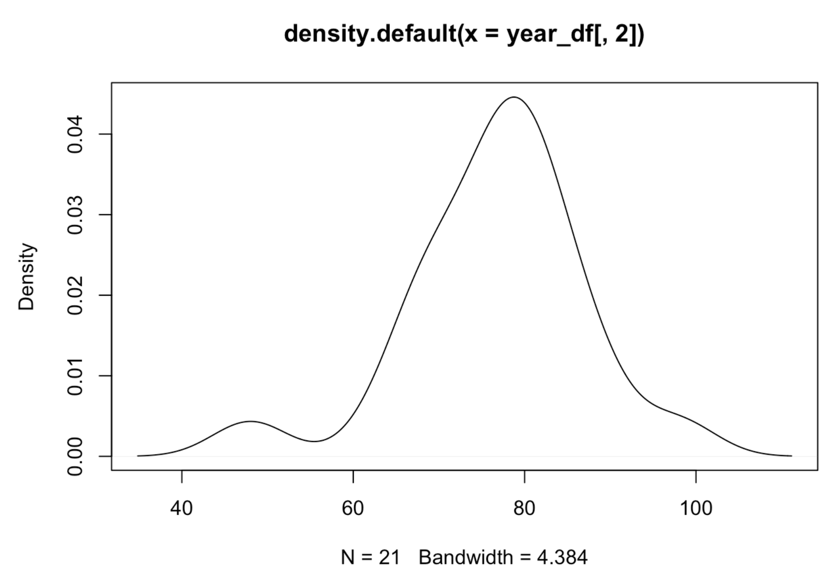
Nothing crazy jumping out here, though it looks like we may have an outlier. Lets look at it compared to the other values in a boxplot.

```
boxplot(year_df[,2],
        main = "First day of Fall by Year",
        notch = FALSE,
        ylab = "Day",
        xlab = "Day of Summer",
        horizontal = TRUE)
```



Definetly something happening here. Lets see how these points are distributed, then determine if we can use grubbs test to see if it is a statistically significant year we should remove.

```
plot(density(year_df[,2]))
```



Yes, it is normally distributed. Now lets look at grubbs test.

```
grubbs.test(year_df[,2])
```

```
##
##  Grubbs test for one outlier
##
## data:  year_df[, 2]
## G = 2.79274, U = 0.59053, p-value = 0.01873
## alternative hypothesis: lowest value 48 is an outlier
```

```
temps[48,1]
```

```
## [1] "17-Aug"
```

Looks like the year that had the unofficial summer end of the 48th day of summer in 2013 (August 17th) is a statistically significant outlier. What happened here though? A quick google search for "Atlanta summer 2013 coldfront" returns the first result, on August 16th, 2013: "**Atlanta cold snap: Why is it sweater weather in the South?**" **https://www.csmonitor.com/USA/2013/0816/Atlanta-cold-snap-Why-is-it-sweater-weather-in-the-South**

Wow- there really was a historic cold front that blew through the south that exact week! That said, lets remove it from our data, we do not want a one off historic cold front impacting our average summer end calculation.

```
c(mean(year_df[-18,2]), temps[78,1])
```

```
## [1] "78.15"  "16-Sep"
```

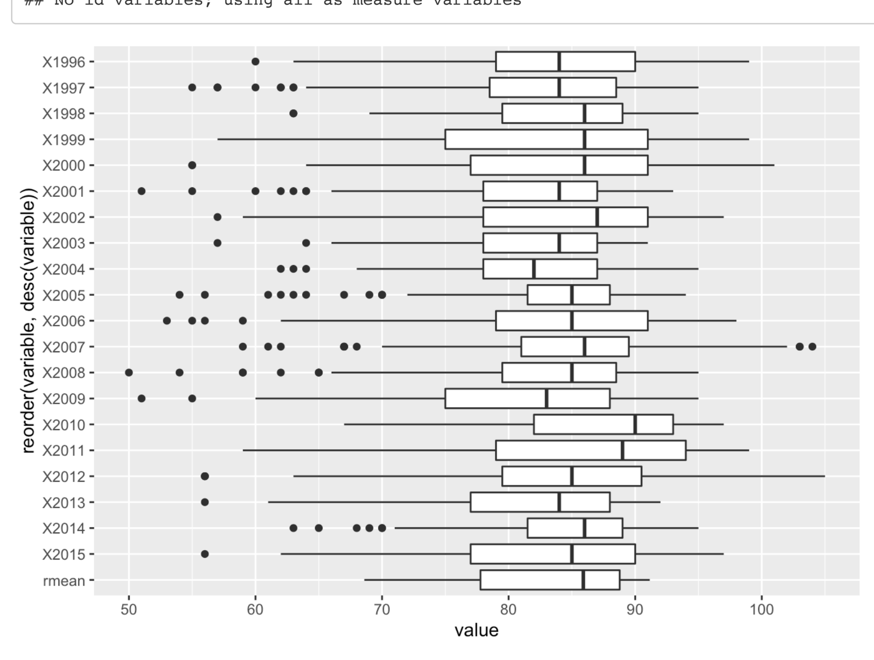
Looks like our unofficial summer end was on average, September 16th. Pretty close to the official summer end of September 22nd!

Question 6.2b Analysis

Now to answer the question of whether summers have been getting hotter, lets do some exploratory data analysis.

```
library(reshape2)
library(ggplot2)
library(plyr)
ggplot(melt(data), mapping=aes(x=reorder(variable,desc(variable)), y=value))+geom_boxplot()+coord_flip()
```

```
## No id variables; using all as measure variables
```



Looking at the box plot, it does not appear we have a clear picture that summers are getting hotter. We can see there is a few hotter summers in there, specifically 2010 and 2011, but nothing too unusual. With this, I would hypothesize that no, summers are not getting warmer.

CUSUM

Now, Lets prove our hypothesis. First we need to get the data in a workable format to put in one big CUSUM model. The idea here is we want to build a CUSUM model that excludes the non summer months that uses the mean of only the summer months. To do so, we need the data in one column. The below code transforms it appropriately.

```
temperatures <- data.frame(temp=unlist(data[1:84,1:20], use.names = FALSE))
days <- rep(temps[1:84,1], 20)
years <-c()
for (year in colnames(data[,1:20])){years <- c(years, rep(year,84))}
all_summers <- data.frame(years, days,temperatures)
head(all_summers)
```

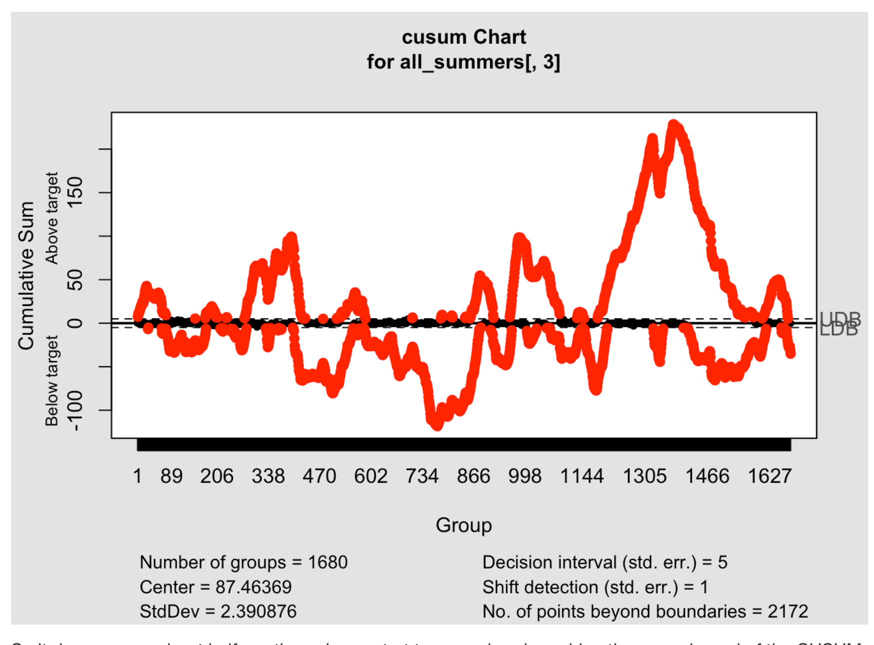
```
##   years  days temp
## 1 X1996 1-Jul   98
## 2 X1996 2-Jul   97
## 3 X1996 3-Jul   97
## 4 X1996 4-Jul   90
## 5 X1996 5-Jul   89
## 6 X1996 6-Jul   93
```

```
tail(all_summers)
```

```
##      years   days temp
## 1675 X2015 17-Sep   83
## 1676 X2015 18-Sep   83
## 1677 X2015 19-Sep   87
## 1678 X2015 20-Sep   89
## 1679 X2015 21-Sep   77
## 1680 X2015 22-Sep   76
```

Now let's look at the CUSUM model (using the same parameters as discussed above). A quick note - typically, it is not advisable to use a combined data set in this way with a CUSUM model, when all data points are not separable from each other. i.e. Summer of 1996 does not impact the summer of 1997, and mashing them up next to each other in a cusum model assumes the points are related, as it will calculate the last day of summer in 1996 into the CUSUM of the first day of summer in 1997. This is not a great way to analyze. However, for demonstration purposes, we will try.

```
all_summer_mean <- mean(all_summers[,3])
all_summer_sd <-  sd(all_summers[,3])
year_cusum <- cusum(all_summers[,3], std.dev = summer_sd, center = all_summer_mean, se.shift = 1)
```



So it does appear about halfway through, we start to more days breaching the upper bound of the CUSUM model in the latter half of the data than the first half. As such, I have concluded it is infact getting hotter, and it started to be statistically significantly hotter starting in 2006 (halfway through our data set.)