Homework 3

Joe Diaz

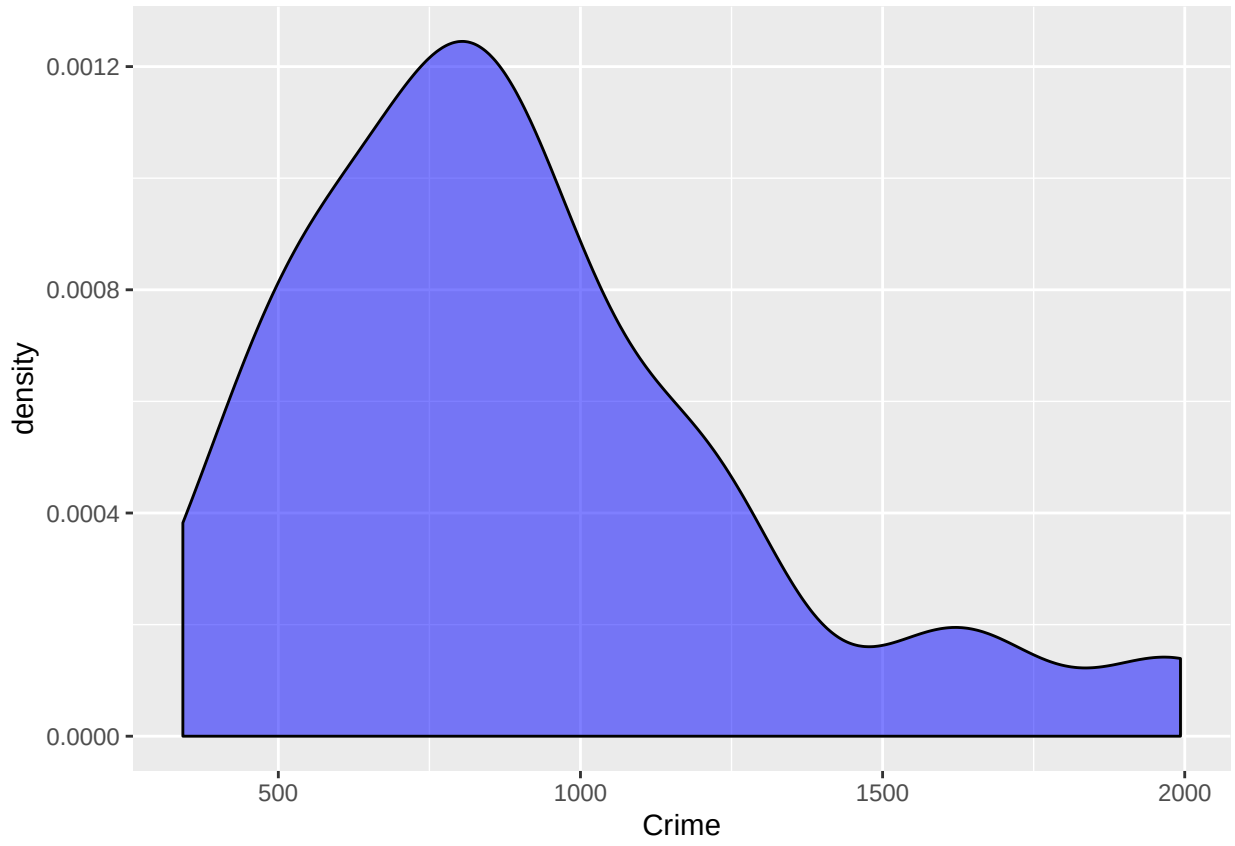
9/10/2019

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.

A good way to determine whether a data set has outliers is to visualize it. I think a density plot and some box-whsiker plots are in order!

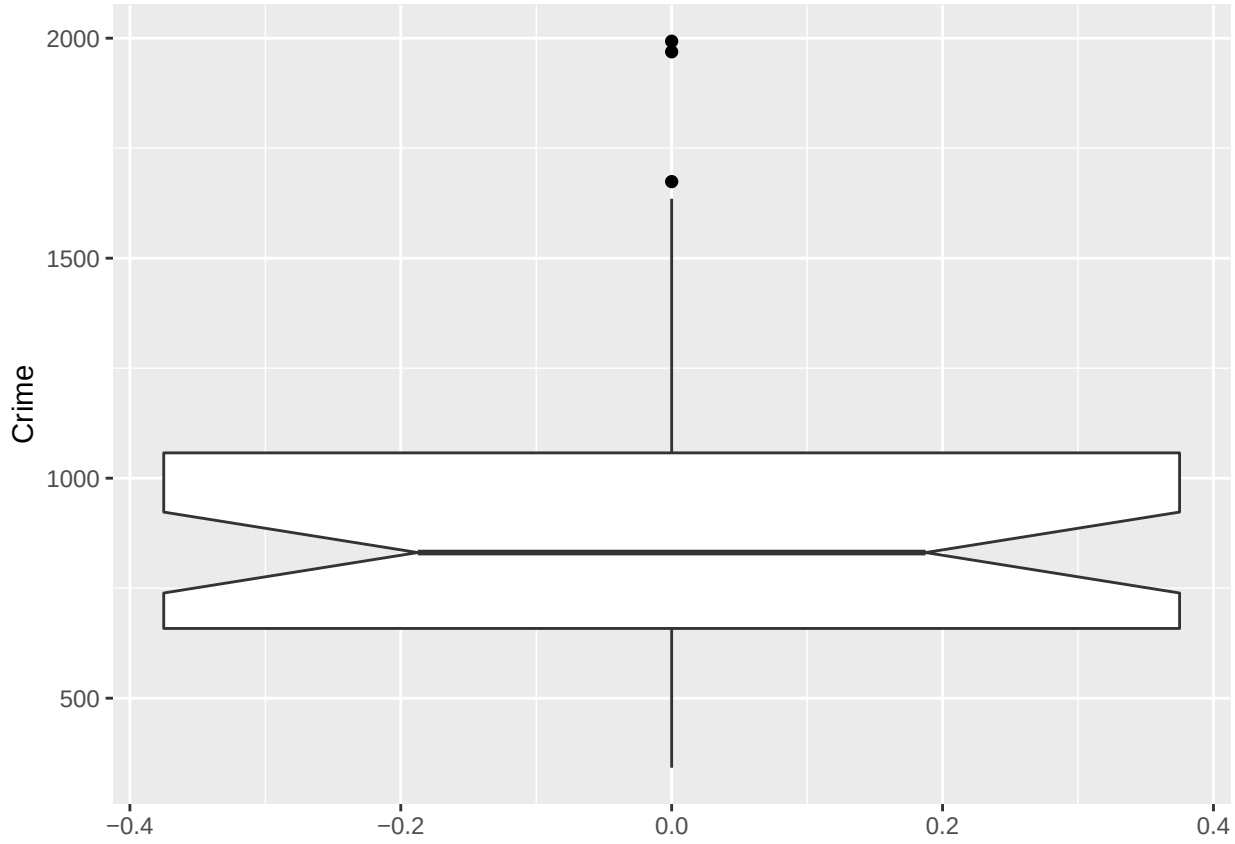
```
library(outliers)
library(ggplot2)
data <- read.table("uscrime.txt", stringsAsFactors = FALSE, header = TRUE)
ggplot(data, aes(x=Crime)) + geom_density(fill='blue', alpha = .5)
```



From the image above, it looks like that the data is not normally distributed. Though, without the tail to the right, it might be normally distributed. This tells me that the higher values are candidates for outliers.

Next up is a box-whisker plot.

```
ggplot(data, aes(y=Crime))+
  geom_boxplot(outlier.colour="black", outlier.shape=16,
               outlier.size=2, notch=TRUE)
```



According the the plots above, we have reason to believe that outliers exist in the data.

Up next, we will use the grubbs function from the outliers package.

According to the documentation, we can use `type = 11` to test both the lowest and highest points of the distribution.

```
grubbs.test(data[,'Crime'], type = 11)
```

```
##
##  Grubbs test for two opposite outliers
##
## data:  data[, "Crime"]
## G = 4.26880, U = 0.78103, p-value = 1
## alternative hypothesis: 342 and 1993 are outliers
```

Looks like at least the highest OR lowest points are outliers. Let's check them.

```
"Is top an outlier?"
```

```
## [1] "Is top an outlier?"
```

```
grubbs.test(data[,'Crime'], type = 10)
```

```
##
##  Grubbs test for one outlier
##
## data:  data[, "Crime"]
## G = 2.81290, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
```

```
"Is bottom an outlier?"
```

```
## [1] "Is bottom an outlier?"
```

```
grubbs.test(data[,'Crime'], type = 10, opposite = TRUE)
```

```
##
##  Grubbs test for one outlier
##
## data:  data[, "Crime"]
## G = 1.45590, U = 0.95292, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
```

Considering both P-values, the highest point seems to be more likely to be an outlier. Let's remove that point and check the next highest point.

```
to_test <- data[,'Crime'][-which.max(data[,'Crime'])]

grubbs.test(to_test, type = 10)
```

```
##
##  Grubbs test for one outlier
##
## data:  to_test
## G = 3.06340, U = 0.78682, p-value = 0.02848
## alternative hypothesis: highest value 1969 is an outlier
```

The next highest looks to be an outlier as well. Let's check the next one!

```
to_test2 <- to_test[-which.max(to_test)]
grubbs.test(to_test2, type = 10)
```

```
##
##  Grubbs test for one outlier
##
## data:  to_test2
## G = 2.56460, U = 0.84712, p-value = 0.1781
## alternative hypothesis: highest value 1674 is an outlier
```

This one does not seem to be an outlier.

Answer

Looks like the highest 2 points are outliers

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?

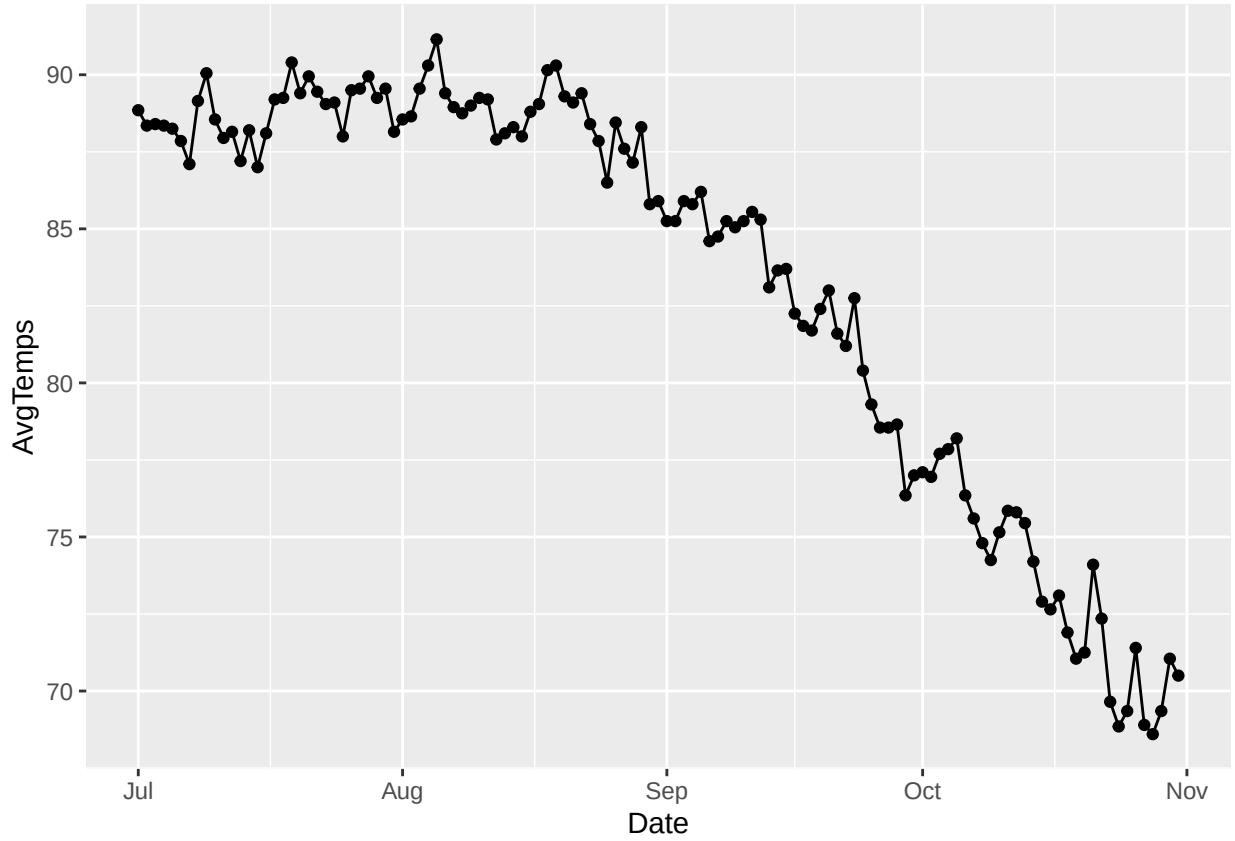
In the world of food supply chain, a change detection model would be beneficial in managning food packaging (i.e putting the correct amount of sauce in a packet). Since mostly machines do these sorts of tasks, a change would be detrimental to the business as it can result in wasted material, or too little of the product packaged. Since inspection whether or not there was a change that occured is relatively cheap (we can continue production even while checking as the consequence is minor), I'd imagine being able to set a relatively sensitive model. We can choose a relatively low C and a relatively low threshold for this.

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.

First, let's visualize the data.

```
temps <- read.table("temps.txt", stringsAsFactors = FALSE, header = TRUE)
temps[,"AvgTemps"] <- rowMeans(temps[,-1])

temps[,"Date"] <- as.Date(temps[,'DAY'], "%d-%B")

ggplot(data=temps, aes(x=Date,y=AvgTemps)) +
  geom_line()+
  geom_point()
```



So there is definitely a change present. The question is where.

With a bit of googling, it is suggested that C should be half of the standard deviation and the threshold should be 4 or 5 times the standard deviation. While these are suggested values, it makes sense because standard deviation is used as the metric to measure distance away from the mean.

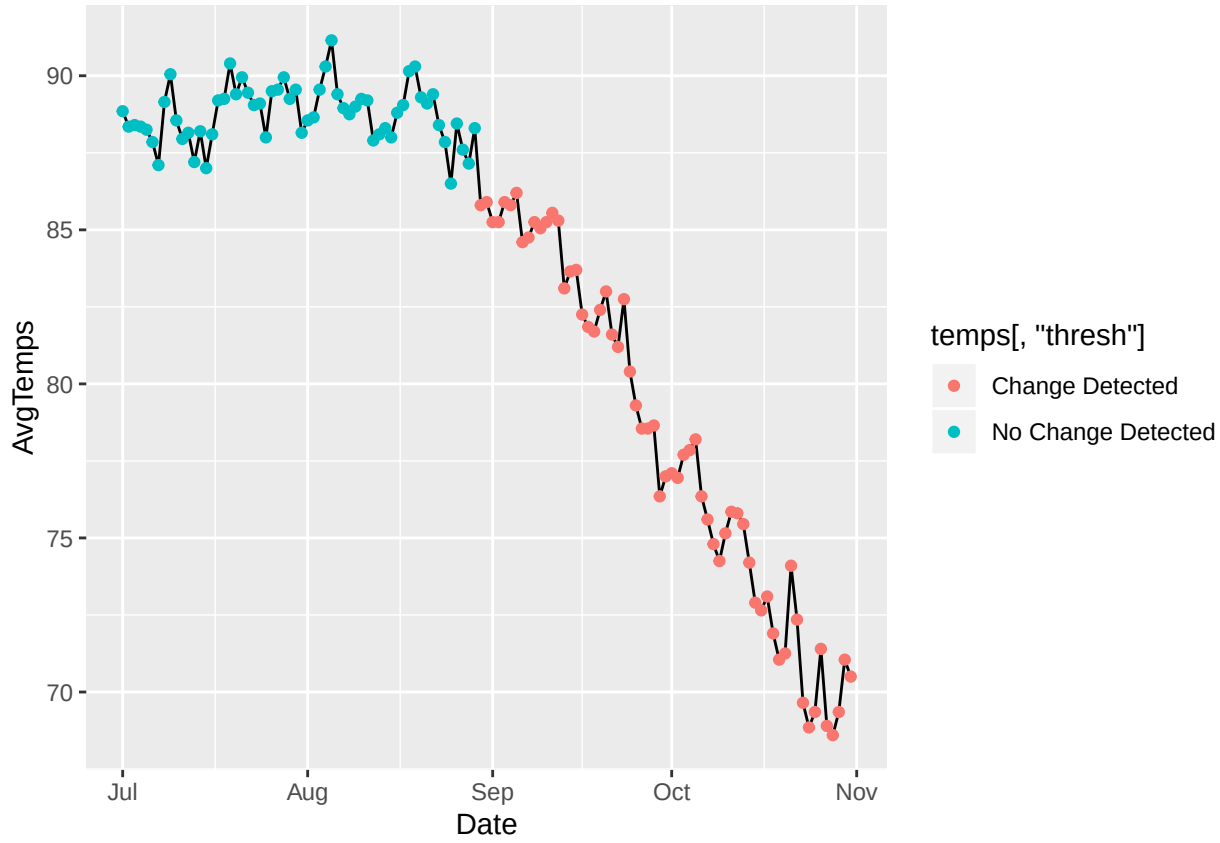
The code below is a manual cusum implementation using looping.

```
cusum <- function(xt,prev_st,mean,C) {


  return(max(c(0,prev_st+(mean-xt-C))))

}

temps_mean <- mean(temps[1:31,"AvgTemps"])

C <- sd(temps[1:31,"AvgTemps"])/2

thresh <- sd(temps[1:31,"AvgTemps"])*5

temps[,"st"] <- NULL
temps[1,"st"] <- 0

for(i in 2:nrow(temps)){
  temps[i,'st'] <- cusum(xt = temps[i,'AvgTemps']
                         , prev_st = temps[i-1,'st']
```

```
                          , mean = temps_mean
                          , C = C
                          )
}
```

In R, we can apply custom functions over rows of data. Let's leverage this to determine whether a data points falls under the threshold

```
is_above_thresh <- function(x,threshold=thresh){

  if(x<threshold){
    return("No Change Detected")
  } else{
    return("Change Detected")
  }

}

temps[,"thresh"] <- sapply(temps[,"st"],FUN = is_above_thresh)

ggplot(data=temps, aes(x=Date,y=AvgTemps)) +
  geom_line()+
  geom_point(aes(color = temps[,"thresh"]))
```



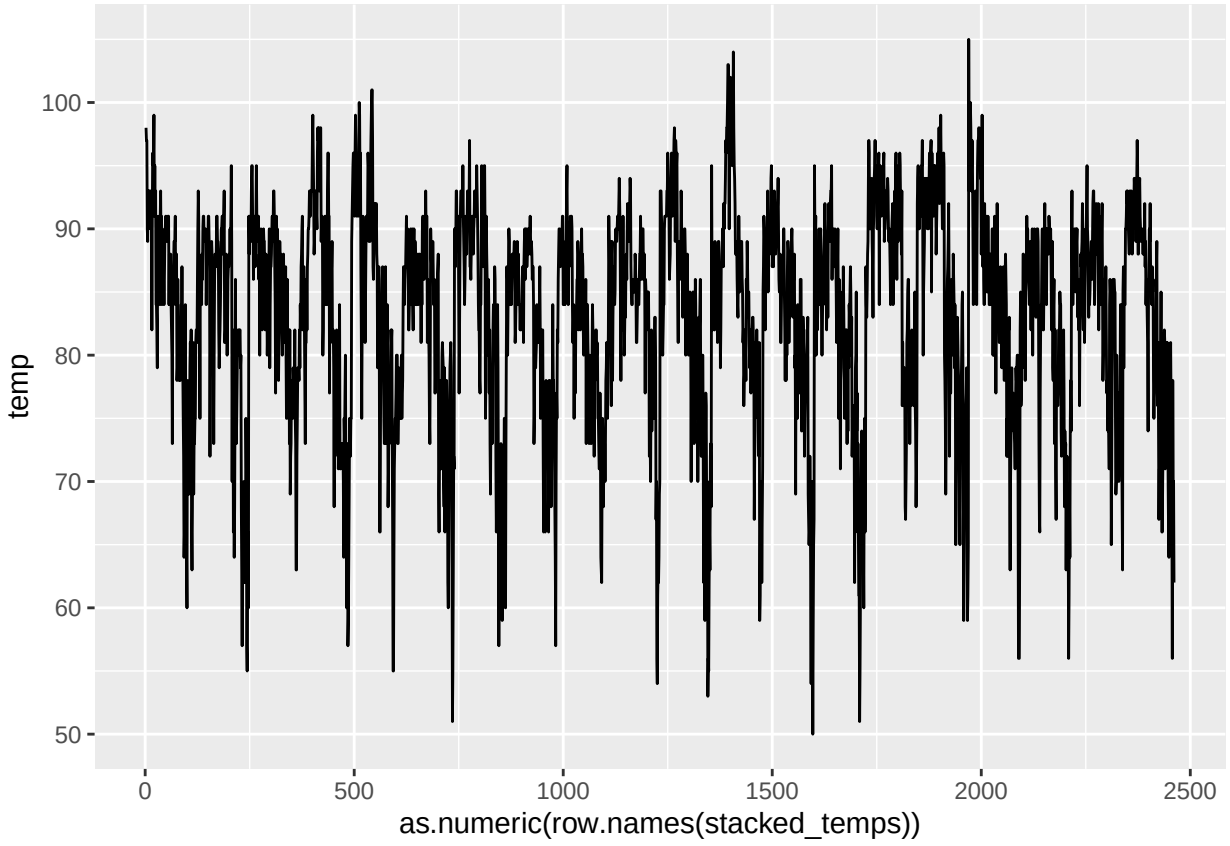
```
# 30th of August is when summer starts
```

According to cusum, there is a change detected on the very last day of august.

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).

Let's manipulate the data so all dates are stacked together.

```
# Plotting with index instead of date as there are missing dates


stacked_temps <- data.frame(ncol=2,nrow=0)

colnames(stacked_temps) <- c('date','temp')



for(col in colnames(temps[85:nrow(temps),2:21])){

  temp_df<-temps[,c("DAY",col)]
  temp_df['date'] <- sapply(temp_df[,"DAY"], paste,gsub("X","",col) , sep="")
  colnames(temp_df)[colnames(temp_df)==col] <- "temp"
  stacked_temps <- rbind(stacked_temps,temp_df[,c('date','temp')])

}
# removing first row as it was a result of the initialization of blank dataframe
stacked_temps<-stacked_temps[-1,]

stacked_temps[,'date'] <- as.Date(stacked_temps[,'date'], "%d-%B%Y")


ggplot(data=stacked_temps, aes(x = as.numeric(row.names(stacked_temps)),y=temp)) +
  geom_line()
```



The plot above tells me that a change may not be present. Maybe we can use average yearly temperatures.

```
# utilized dplyr and lubridate as it was more intuitive for me to use these packages when working with
library(lubridate)
```

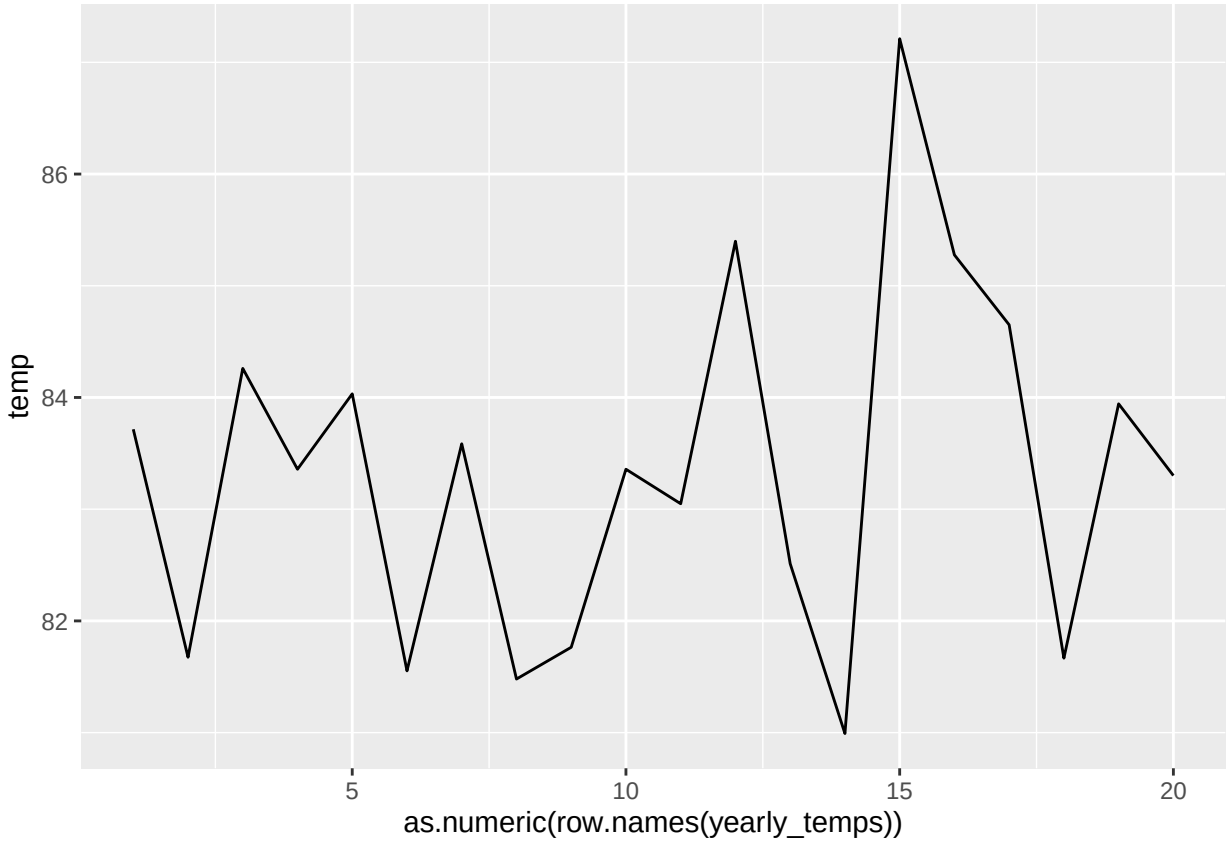
```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
yearly_temps<-stacked_temps %>%
                group_by(month=floor_date(date, "year")) %>%
                summarize(summary_variable=mean(temp))


colnames(yearly_temps) <- c('date', 'temp')

yearly_temps<-data.frame(yearly_temps)

ggplot(data=yearly_temps, aes(x = as.numeric(row.names(yearly_temps)),y=temp)) +
  geom_line()
```



From my perspective, it seems that there is no change. Let's try to prove this using cusum.

```
C <- 0
thresh <- 20
m <- 80 # data hovers around 80

yearly_temps[,"st"] <- NULL
yearly_temps[1,"st"] <- 0

for(i in 2:nrow(yearly_temps)){
```

```
  yearly_temps[i,'st'] <- cusum(xt = yearly_temps[i,'temp']
                           , prev_st = yearly_temps[i-1,'st']
                           , mean = m
                           , C = C
                           )
}

yearly_temps
```

```
##          date     temp st
## 1  1996-01-01 83.71545  0
## 2  1997-01-01 81.67480  0
## 3  1998-01-01 84.26016  0
## 4  1999-01-01 83.35772  0
## 5  2000-01-01 84.03252  0
## 6  2001-01-01 81.55285  0
## 7  2002-01-01 83.58537  0
## 8  2003-01-01 81.47967  0
## 9  2004-01-01 81.76423  0
## 10 2005-01-01 83.35772  0
## 11 2006-01-01 83.04878  0
## 12 2007-01-01 85.39837  0
## 13 2008-01-01 82.51220  0
## 14 2009-01-01 80.99187  0
## 15 2010-01-01 87.21138  0
## 16 2011-01-01 85.27642  0
## 17 2012-01-01 84.65041  0
## 18 2013-01-01 81.66667  0
## 19 2014-01-01 83.94309  0
## 20 2015-01-01 83.30081  0
```

After trying a bunch of thresholds and C, there seems to be no change detected.