Homework 3

2025-09-04

Note

I ended up using `dplyr` extensively in this homework and their cheat sheet[1] proved a great resource.

Boilerplate code

First, we load the necessary libraries:

```r
library(printr) # pretty print for Rmd
library(outliers)
library(ggplot2)
library(dplyr)
library(purrr, include.only = "accumulate")
library(lubridate)

# set seed for reproducibility
set.seed(42)
```

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.

First we need to load the data.

```r
crime_stats <- read.csv(
  "http://www.statsci.org/data/general/uscrime.txt",
  sep = "\t"
)
```

Then I wanted to get a bit of background on Grubbs' test to understand it. Per NIST, Grubbs' test is used to detect a single outlier in a univariate data set that follows an approximately normal distribution[2] and is the recommended test when testing for a single outlier. Recommended best practice is to test the nomality assumption which can be done using a normal probability plot. This is very easy to generate in R[3].

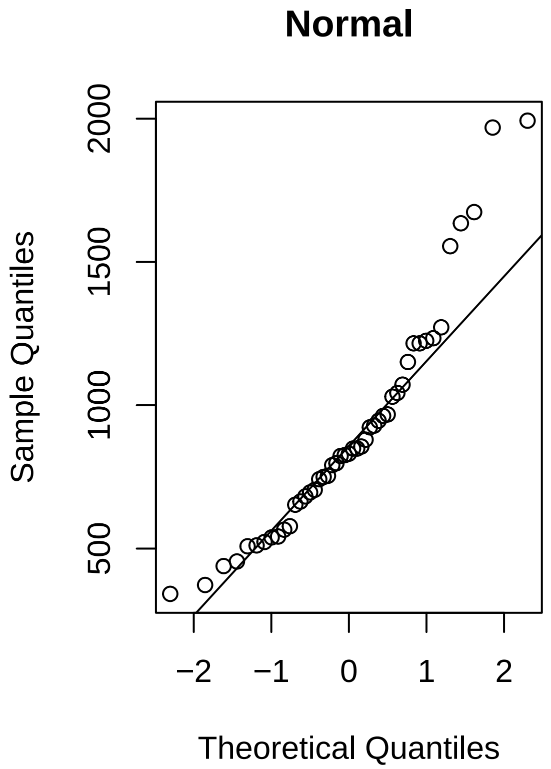
[1] https://github.com/rstudio/cheatsheets/blob/main/data-transformation.pdf

[2] https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h1.htm

[3] https://www.statology.org/test-for-normality-in-r/

```
#define plotting region
par(mfrow=c(1,2))

#create Q-Q plot for both datasets
qqnorm(crime_stats$Crime, main='Normal')
qqline(crime_stats$Crime)
```



By visual observation the majority of the data appears normal with deviations at the tails. Therefore, I believe it is appropriate to use Grubbs' test. As this is the case, let's see what the test says.

```
grubbs.test(crime_stats$Crime, opposite=TRUE)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime_stats$Crime
## G = 1.45589, U = 0.95292, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
```

```
grubbs.test(crime_stats$Crime)
```

```
##
##  Grubbs test for one outlier
##
## data:  crime_stats$Crime
## G = 2.81287, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
```

```
grubbs.test(crime_stats$Crime, type=11)
```

```
##
##  Grubbs test for two opposite outliers
##
## data:  crime_stats$Crime
## G = 4.26877, U = 0.78103, p-value = 1
## alternative hypothesis: 342 and 1993 are outliers
```

Using a p-value of 0.05 (obviously a controversal, but standard choice), we cannot use Grubbs' test to reject any of the null hypotheses and must conclude that neither tail 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?

In my job, I work extensively with time series data. A strong application of the CUSUM technique is monitoring battery state of charge (SOC) in lead-acid batteries. Low SOC harms long-term health due to sulfation, and while some vehicle programs operate with consistently lower-than-desired SOC, we still need to detect degradation. The complication is that self-discharge and off-power features also cause expected SOC loss. Static thresholds miss these changes, especially when SOC is already low.

To apply CUSUM, I would choose the critical value based on the historical variance of SOC measurements, ensuring it reflects the natural day-to-day fluctuations from self-discharge and off-power loads. The threshold would then be tuned using historical failure data and pilot testing: set high enough to avoid false positives from normal SOC drift, but low enough to catch sustained deviations that indicate degradation. Because our team has limited capacity for triaging alerts, I would bias the threshold toward reducing false positives (accepting some false negatives) to balance detection accuracy with operational constraints.

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).

I requested data from the National Oceanic and Atmospheric Administration (NOAA)'s online climate data portal[4] for Hartsfield-Jackson Atlanta Airport (station GHCND:USW00013874).

```r
temp <- read.csv("hartford-jackson.csv")
```

As average temperature was not available at this station until 1998-04-01, I will use the daily high to perform the analysis. First we need to do some data cleaning: making sure are dates are date types, also extracting year and month into. The month function is actually very helpful because it returns an enumeration rather than a string[5]

```r
temp$DATE <- as.Date(temp$DATE, "%Y-%m-%d")

temp <- temp %>%
  mutate(
    YEAR = year(DATE),
    MONTH = month(DATE, label = TRUE, abbr = TRUE),
    MONTH_DAY = format(DATE, "%b-%d"),
    YEARLESS_DATE = update(DATE, 2024) # leap year
  )
```

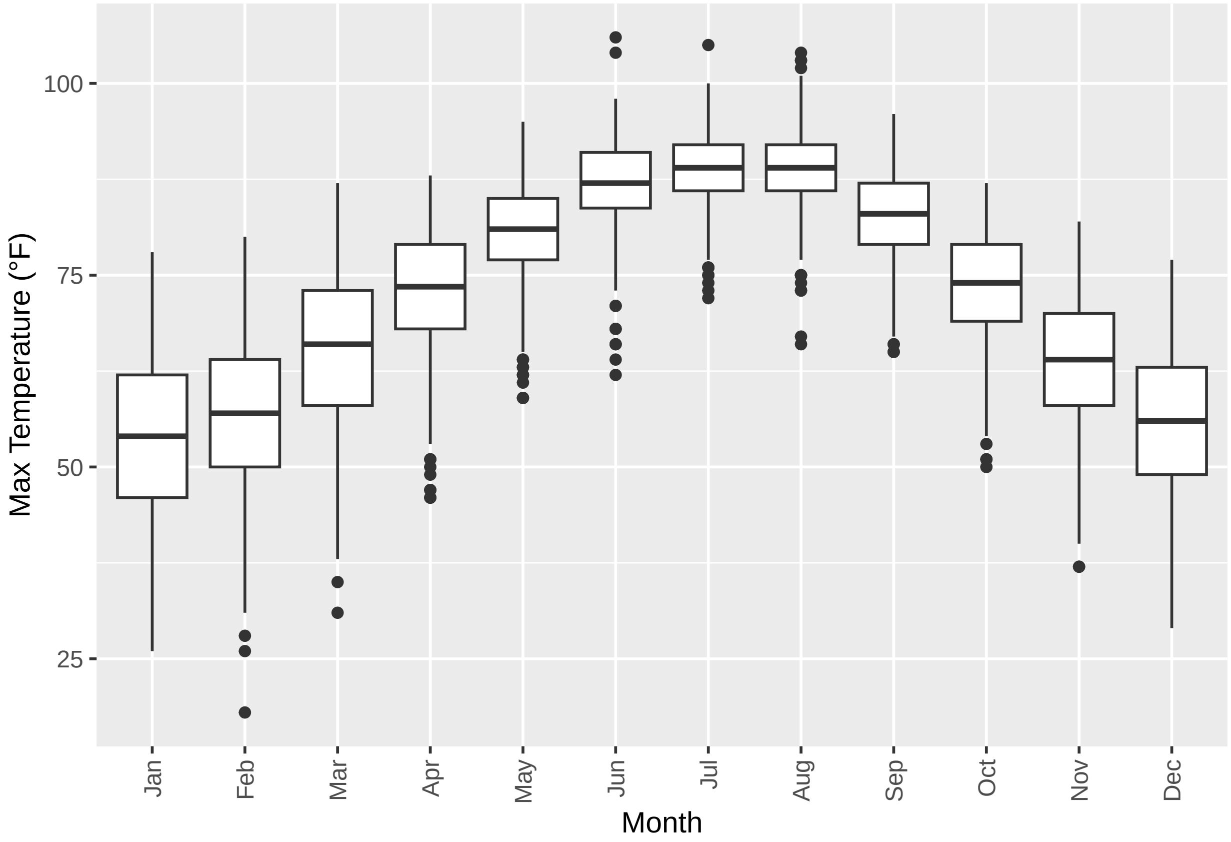
Part 1

First I would like to verify that July is the hottest month of the year

```r
ggplot(temp, aes(x = MONTH, y = TMAX)) +
  geom_boxplot() +
  labs(x = "Month", y = "Max Temperature (°F)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

[4] https://www.ncei.noaa.gov/cdo-web/search?datasetid=GHCND

[5] https://www.rdocumentation.org/packages/lubridate/versions/1.9.4/topics/month



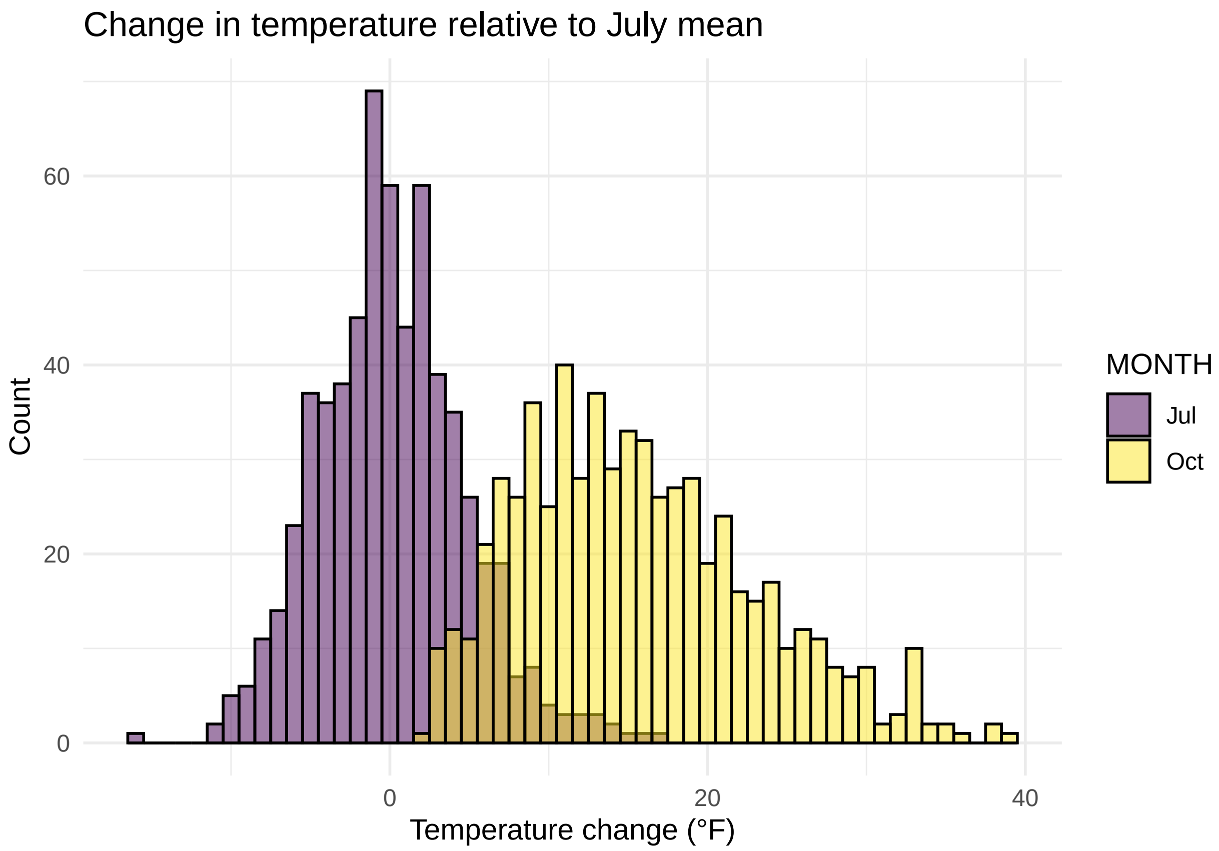
Not only is July the hottest month, it also has a very narrow IQR; however, there are a fair number of outliers on the cooler side. I am going to choose July as my reference and base the parameters relative to it. This makes selecting the average easy

```
stats <- temp %>%
  filter(MONTH == "Jul") %>%
  summarise(
    mu = mean(TMAX),
    std = sd(TMAX)
  )

mu <- stats$mu
std <- stats$std
```

The choice of C and T are harder. Implicit in the question is the idea that July is summer and October is fall. Therefore, let's compare the temperatures for those months using a histogram. I am going to perform the $\mu - x_t$ shift in the temperature so that we can use this to choose C.

```
ggplot(
  temp[(temp$MONTH == "Jul") | (temp$MONTH == "Oct"),],
  aes(x = mu - TMAX, fill = MONTH)
) +
geom_histogram(
  position = "identity",
  alpha = 0.5,
  binwidth = 1,
  color = "black"
) +
labs(
  x = "Temperature change (°F)",
  y = "Count",
  title = "Change in temperature relative to July mean"
) +
theme_minimal()
```



Given the assumption about October being definitively fall, I think a natural point for the sensitivity parameter is the distribution crossing points. Therefore,

```
C <- 6
```

With these parameters chosen we just need to select a decision interval. I'm going to calculate and plot the change in the CUSUM statistic and to see if there is a clear threshold. I found this very useful function to do a resetting cumulative sum[6]. It needed some modification, but pointed me in the right direction

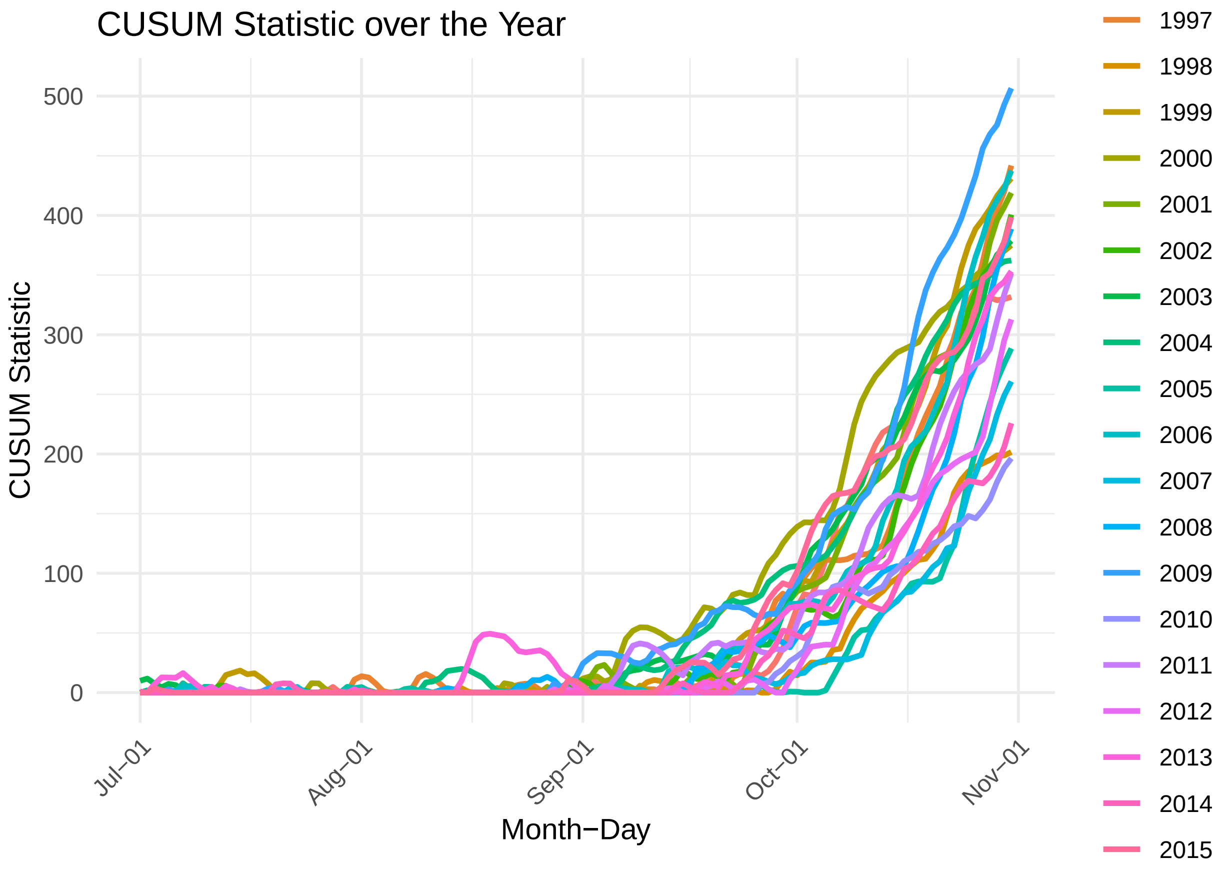
```
end_of_summer <- temp[("Jul" <= temp$MONTH) & (temp$MONTH <= "Oct"),]
end_of_summer <- end_of_summer %>%
  group_by(YEAR) %>%
  mutate(
    CUSUM_STAT = accumulate(
      TMAX,
      function(St_min1, xt, year) {
        max(0, St_min1 + (mu - xt - C))
      },
```

[6] https://stackoverflow.com/a/49076872/1543042

```
      .init = 0
    )[-1]
  )



ggplot(
  end_of_summer,
  aes(x = YEARLESS_DATE, y = CUSUM_STAT, color = factor(YEAR))
) +
  geom_line(linewidth = 1) +
  labs(
    x = "Month-Day",
    y = "CUSUM Statistic",
    color = "Year",
    title = "CUSUM Statistic over the Year"
  ) +
  scale_x_date(date_labels = "%b-%d") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



To avoid a false declaration of the start of fall in August 2013, we need to set the critical value to be at least 49.4; however, that seems to be overfitting our data. So let's add a buffer to that and set

```
T = 60
```

Now we have the parameters for the CUSUM model, let's see when fall started!

```
first_day_of_fall <- end_of_summer %>%
  filter(CUSUM_STAT >= T) %>%
  group_by(YEAR) %>%
  slice_min(DATE, n = 1) %>%
  ungroup() %>%
  select(
    YEAR,
    TMAX,
    MONTH_DAY
  )

left <- first_day_of_fall[1:10,]
right <- first_day_of_fall[11:20,]

cbind(left,right)
```

YEAR	TMAX	MONTH_DAY	YEAR	TMAX	MONTH_DAY
1996	66	Oct-01	2006	70	Sep-29
1997	65	Sep-27	2007	74	Oct-13
1998	72	Oct-09	2008	82	Oct-07
1999	80	Sep-28	2009	74	Sep-19
2000	73	Sep-17	2010	67	Oct-04
2001	71	Sep-29	2011	68	Oct-02
2002	74	Sep-29	2012	63	Oct-08
2003	67	Sep-29	2013	76	Sep-29
2004	73	Sep-20	2014	65	Oct-04
2005	74	Oct-12	2015	71	Sep-26

These seem reasonable, fall started sometime between Sept 17 and Oct 13 (encompassing the Autumnal equinox) over the two decades with temperatures on the first day of fall varying between 63°F and 80°F.

Discussion of Results

As was stated in the lecture choosing model parameters is inherently a trade off between false positives (such as the August 2013 case) and false negatives (saying that fall never occurred). This balancing act needs to be based on the data. So a couple weeks until fall starts, grab your pumpkin spice latte and enjoy the season!

Question 6.2

The CUSUM method is similar to the concept in climate called the cooling degree day[7]. However, the cooling degree day is strictly cumulative rather than resetting when the temperature cools down sufficiently. Therefore, the CUSUM metric can be considered the amount of excessive heat for the year. Because we are looking at above average days, it intuitively makes sense that $C = 0$.

```
C <- 0
```

Based on the fact that from the previous analysis fall started around the Autumnal equinox, I am going to assume that summer ends on Autumnal equinox. Further I am going to assume that summer starts on

[7] https://www.weather.gov/key/climate_heat_cool

May 15. I am going to “train” on the first 4 years of data, determining μ and T, then run inference on the remainder of the data.

```r
summer <- temp[
  ("2024-05-15" <= temp$YEARLESS_DATE)
  & (temp$YEARLESS_DATE <= "2024-09-21"),
]

train_ind <- (1996 <= summer$YEAR) & (summer$YEAR < 2000)
stats <- summer[train_ind,] %>%
    summarise(
      mu = mean(TMAX)
    )

mu <- stats$mu

print(paste("Average summer high from 1996 to 2000: ", format(mu, digits=1, nsmall=1), "°F"))
```

```
## [1] "Average summer high from 1996 to 2000:  86.5 °F"
```

Now we calculate the CUSUM statistic

```r
summer <- summer %>%
    group_by(YEAR) %>%
    mutate(
        CUSUM_STAT = accumulate(
            TMAX,
            function(St_min1, xt, year) {
                max(0, St_min1 + (xt - mu - C))
            },
            .init = 0
        )[-1]
    )
```

We need to choose T so that we are not flagging any of the years in our training data.

```r
summer[train_ind,] %>%
  group_by(YEAR) %>%
  slice_max(CUSUM_STAT, n = 1) %>%
  ungroup() %>%
  select(
    YEAR,
    TMAX,
    MONTH_DAY,
    CUSUM_STAT
  )
```

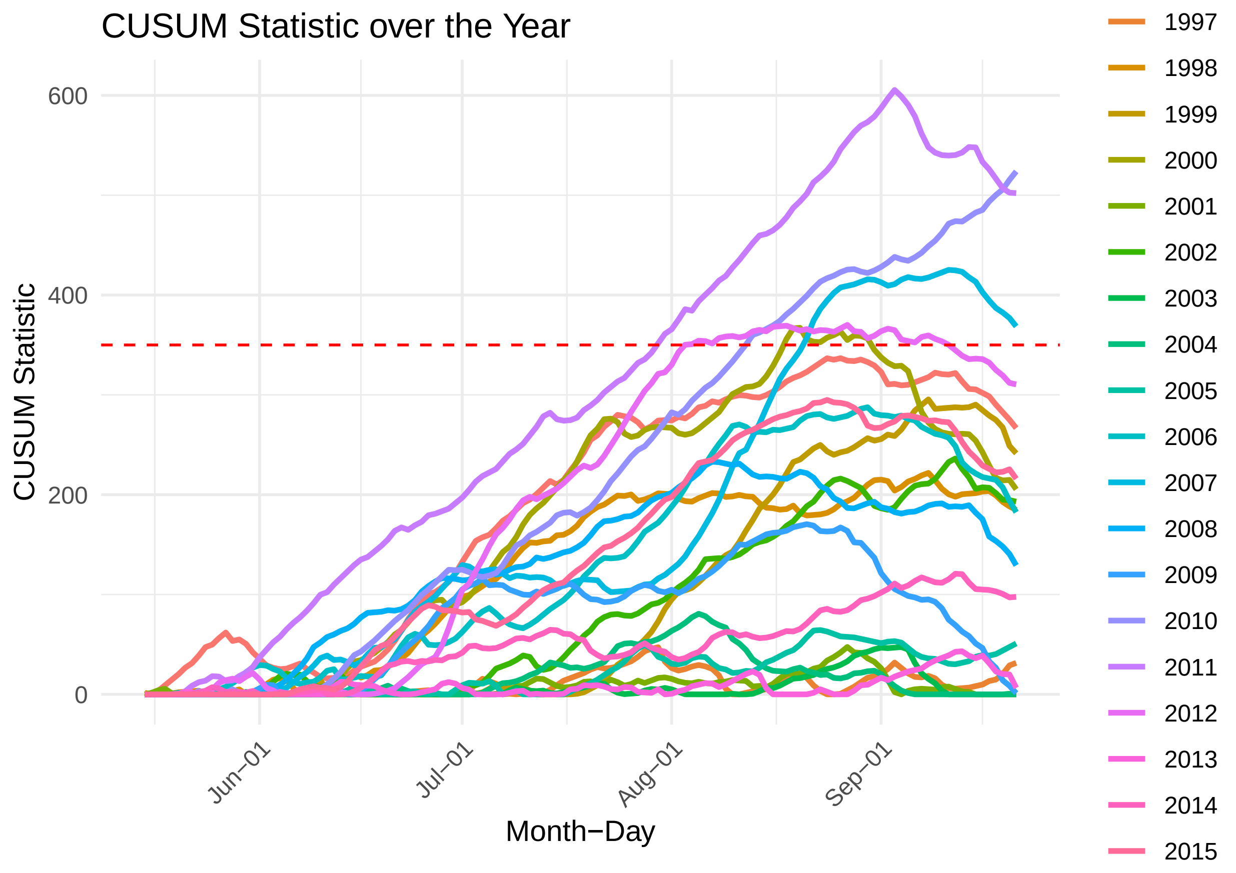
YEAR	TMAX	MONTH_DAY	CUSUM_STAT
1996	88	Aug-26	336.72692
1997	88	Jul-30	47.19423
1998	89	Sep-08	221.82500
1999	92	Sep-08	295.43654

To not flag our training data as warming we need to set $T > 336.7$, so let’s set it to 350.

```r
T <- 350
```

Now let’s see how high the CUSUM statistic gets each year in our inference years.

```r
ggplot(
  summer,
  aes(x = YEARLESS_DATE, y = CUSUM_STAT, color = factor(YEAR))
) +
  geom_line(linewidth = 1) +
  geom_hline(yintercept = T, linetype = "dashed", color = "red") +
  labs(
    x = "Month-Day",
    y = "CUSUM Statistic",
    color = "Year",
    title = "CUSUM Statistic over the Year"
  ) +
  scale_x_date(date_labels = "%b-%d") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



That plot is too busy, but we can clearly see that there are sum years that pass our threshold. Let’s characterize them by looking at the number of days the temperature exceeded T and what the maximum CUSUM statistic was.

```r
excess_heating <- summer %>%
  group_by(YEAR) %>%
  summarise(
    max_CUSUM = max(CUSUM_STAT, na.rm = TRUE),
    count_exceed = sum(CUSUM_STAT >= T, na.rm = TRUE),
  )

left <- excess_heating[1:10,]
right <- excess_heating[11:20,]

cbind(left,right)
```

YEAR	max_CUSUM	count_exceed	YEAR	max_CUSUM	count_exceed
1996	336.72692	0	2006	287.58077	0
1997	47.19423	0	2007	425.19038	32
1998	221.82500	0	2008	233.02115	0
1999	295.43654	0	2009	170.48462	0
2000	367.16731	13	2010	523.87308	41
2001	47.43654	0	2011	605.40962	54
2002	236.24038	0	2012	370.09423	40
2003	47.34038	0	2013	43.12115	0
2004	80.70577	0	2014	120.82500	0
2005	64.38846	0	2015	294.92308	0

Discussion of Results

There does appear to be some degree of warming in the data with 4/5 of the years with excessive heating occurring in the second half of the data with 3/5 occurring sequentially in 2010-2012 and the number of days above T is larger. This is less extreme than I was expecting, but still noticeable. I would say the warming starts in 2007 with the first excessive heat day in the second half of the data.