Outliers Exercise

Question

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.

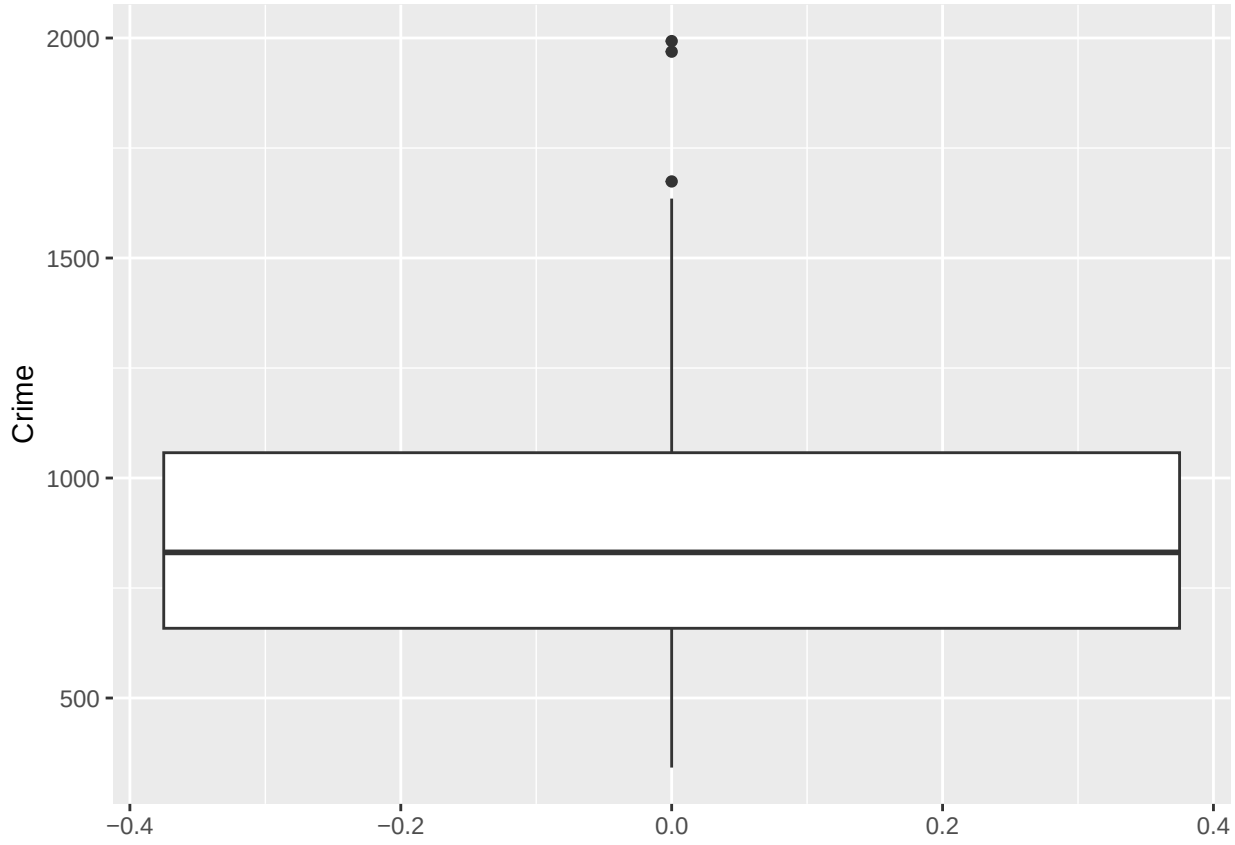
Solution

To check for outliers, the first step is to do an EDA of the data. I begin with the summary.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   342.0   658.5   831.0   905.1  1057.5  1993.0
```

First inspection shows a Max that is 1,100 points higher than the Median and 900 points higher than the third quantile. This suggets there are upper outliers. The lower numbers are less extreme.

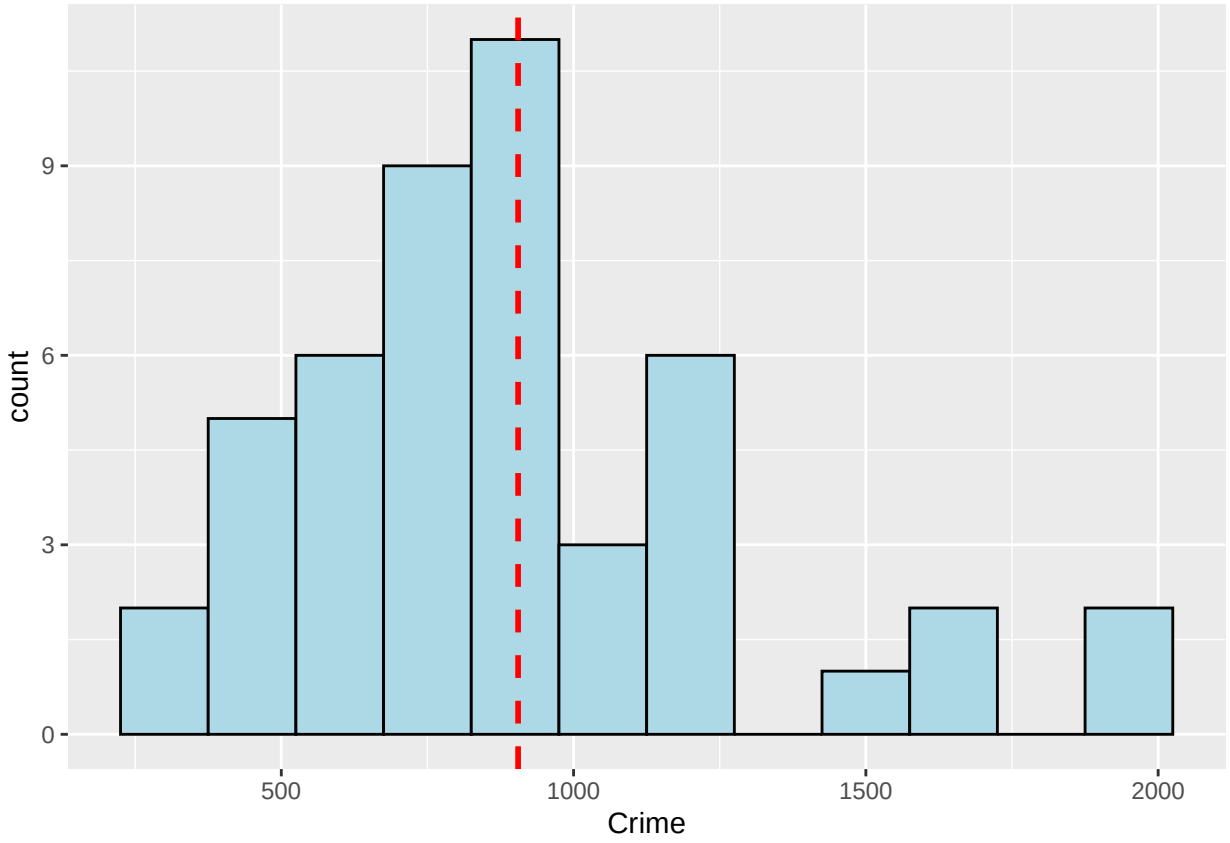
Next, I make a boxplot to inspect the data



The boxplot is similar to our summary. By visual inspection, I can posit that there are at least two upper outliers.

My last EDA is with a histogram

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



The histogram above plots the Crime data against count, with the red dashed line showing the mean of Crimes. Again, we can see that this histogram is right-side heavy, suggesting outliers.

Next, I will apply the grubbs test to see if these points are actually outliers, and to see if they will change the data.

```
## [1] "Grubbs Test for Highest Value:"
```

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

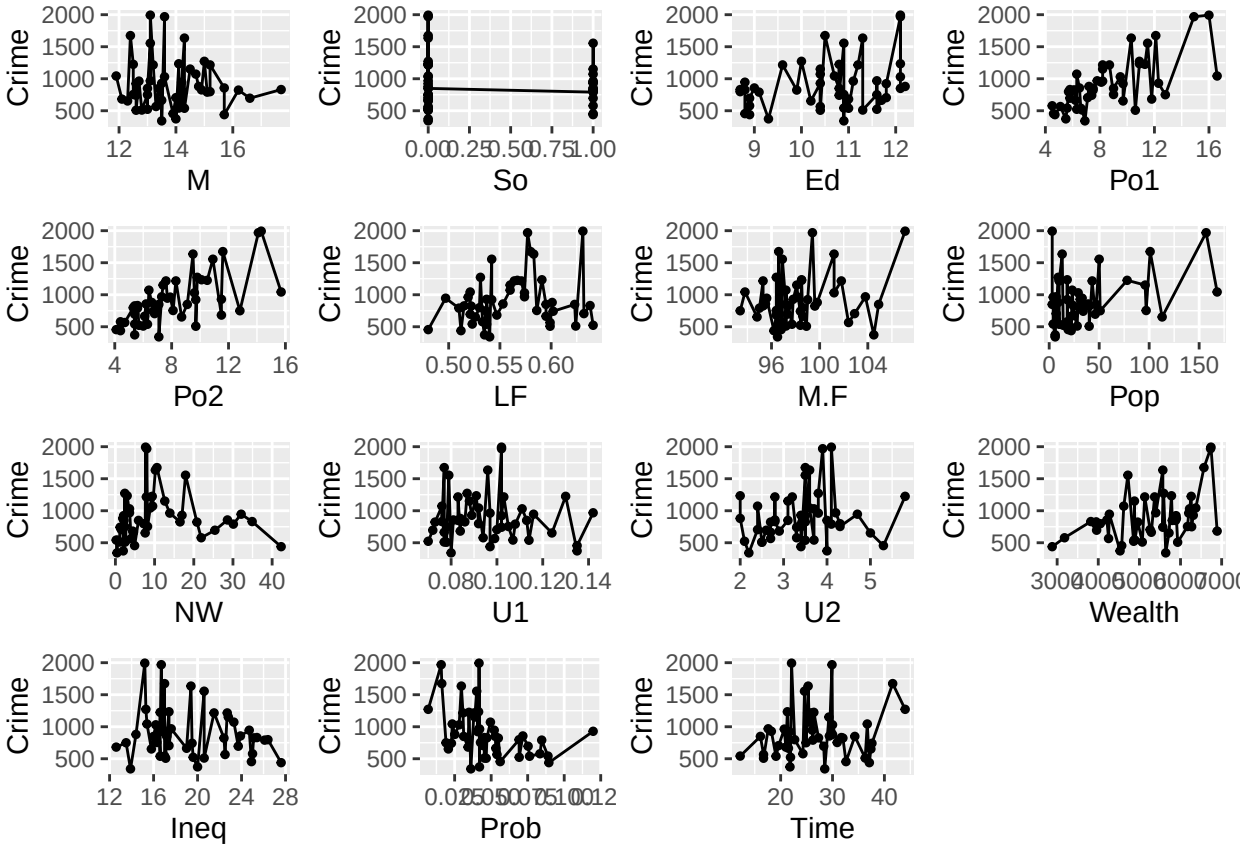
```
## [1] "Grubbs Test for Lowest Value"
```

```
##
##  Grubbs test for one outlier
```

```
##
## data:  crimedata$Crime
## G = 1.45589, U = 0.95292, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
```

Above, I ran the Grubbs Tests to check for outliers on the highest and lowest ends. According to this test, our high point 1993 is NOT an outlier with $\alpha = 0.05$. Watching Sokol's lecture, this would make sense based on what we know about large amounts of data and the likelihood of points outside the norm.

To be certain, I wanted to plot the crime data against the other predictors, one by one.



By quick inspection, plots against Po1 and Po2 imply some sort of linearity. While not getting too deep into this, I can reject the hypothesis that an outlier exists.

CUSUM Exercise

Question

Part 1

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.

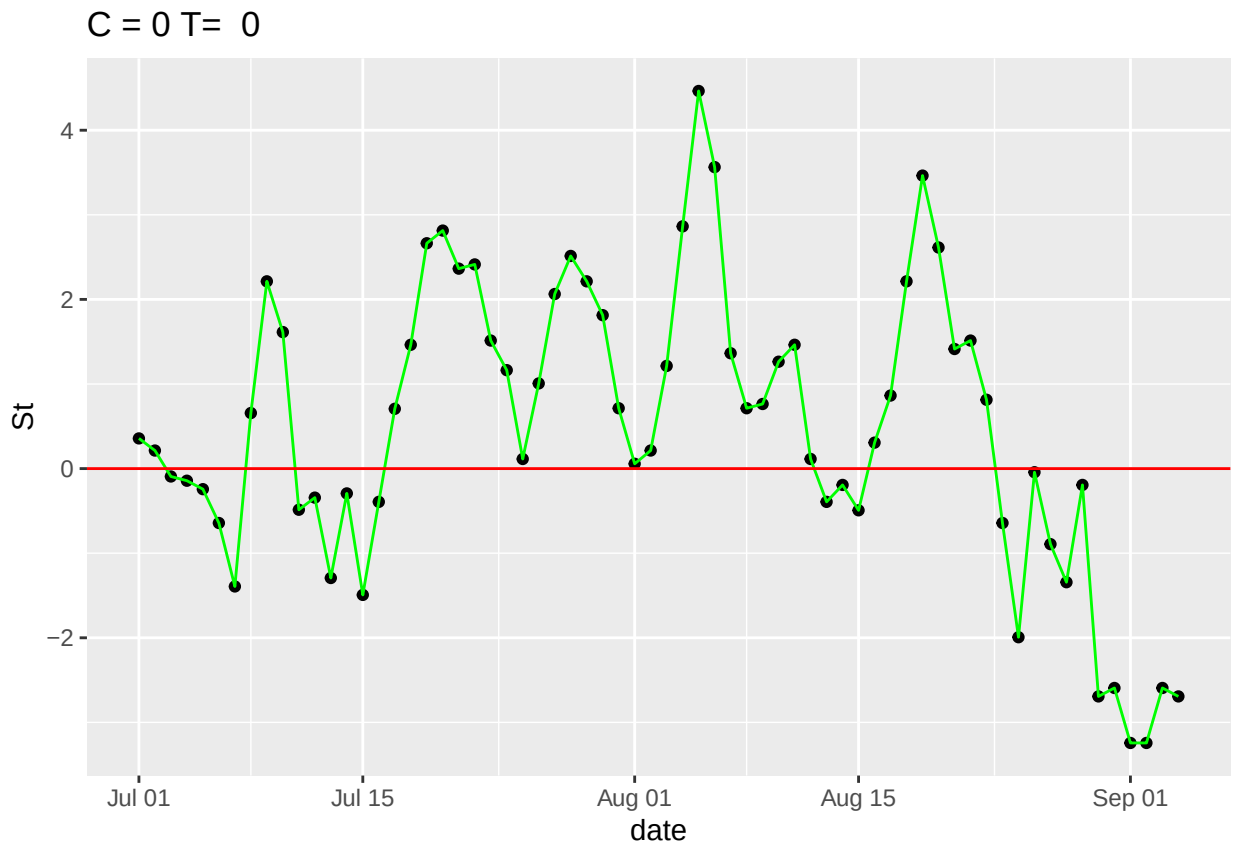
Part 1 Solution

For this I used an excel spreadsheet where I could input various values of C and T into my table. I used dates July 1 - September 4 for my "summer dates." I decided that summer would end after labor day, which is about September 4.

I wrote a function to calculate the St values using inputs C and T. The function is commented below.

```
##          date  mean      helper          St
## 1  2023-07-01 88.85  0.35681818  0.35681818
## 2  2023-07-02 88.35 -0.14318182  0.21363636
## 3  2023-07-03 88.40 -0.09318182 -0.09318182
## 4  2023-07-04 88.35 -0.14318182 -0.14318182
## 5  2023-07-05 88.25 -0.24318182 -0.24318182
## 6  2023-07-06 87.85 -0.64318182 -0.64318182
## 7  2023-07-07 87.10 -1.39318182 -1.39318182
## 8  2023-07-08 89.15  0.65681818  0.65681818
## 9  2023-07-09 90.05  1.55681818  2.21363636
## 10 2023-07-10 88.55  0.05681818  1.61363636
## 11 2023-07-11 87.95 -0.54318182 -0.48636364
## 12 2023-07-12 88.15 -0.34318182 -0.34318182
## 13 2023-07-13 87.20 -1.29318182 -1.29318182
## 14 2023-07-14 88.20 -0.29318182 -0.29318182
## 15 2023-07-15 87.00 -1.49318182 -1.49318182
## 16 2023-07-16 88.10 -0.39318182 -0.39318182
## 17 2023-07-17 89.20  0.70681818  0.70681818
## 18 2023-07-18 89.25  0.75681818  1.46363636
## 19 2023-07-19 90.40  1.90681818  2.66363636
## 20 2023-07-20 89.40  0.90681818  2.81363636
## 21 2023-07-21 89.95  1.45681818  2.36363636
## 22 2023-07-22 89.45  0.95681818  2.41363636
## 23 2023-07-23 89.05  0.55681818  1.51363636
## 24 2023-07-24 89.10  0.60681818  1.16363636
## 25 2023-07-25 88.00 -0.49318182  0.11363636
## 26 2023-07-26 89.50  1.00681818  1.00681818
## 27 2023-07-27 89.55  1.05681818  2.06363636
## 28 2023-07-28 89.95  1.45681818  2.51363636
## 29 2023-07-29 89.25  0.75681818  2.21363636
## 30 2023-07-30 89.55  1.05681818  1.81363636
## 31 2023-07-31 88.15 -0.34318182  0.71363636
## 32 2023-08-01 88.55  0.05681818  0.05681818
## 33 2023-08-02 88.65  0.15681818  0.21363636
## 34 2023-08-03 89.55  1.05681818  1.21363636
## 35 2023-08-04 90.30  1.80681818  2.86363636
## 36 2023-08-05 91.15  2.65681818  4.46363636
## 37 2023-08-06 89.40  0.90681818  3.56363636
## 38 2023-08-07 88.95  0.45681818  1.36363636
## 39 2023-08-08 88.75  0.25681818  0.71363636
## 40 2023-08-09 89.00  0.50681818  0.76363636
## 41 2023-08-10 89.25  0.75681818  1.26363636
## 42 2023-08-11 89.20  0.70681818  1.46363636
## 43 2023-08-12 87.90 -0.59318182  0.11363636
## 44 2023-08-13 88.10 -0.39318182 -0.39318182
## 45 2023-08-14 88.30 -0.19318182 -0.19318182
```

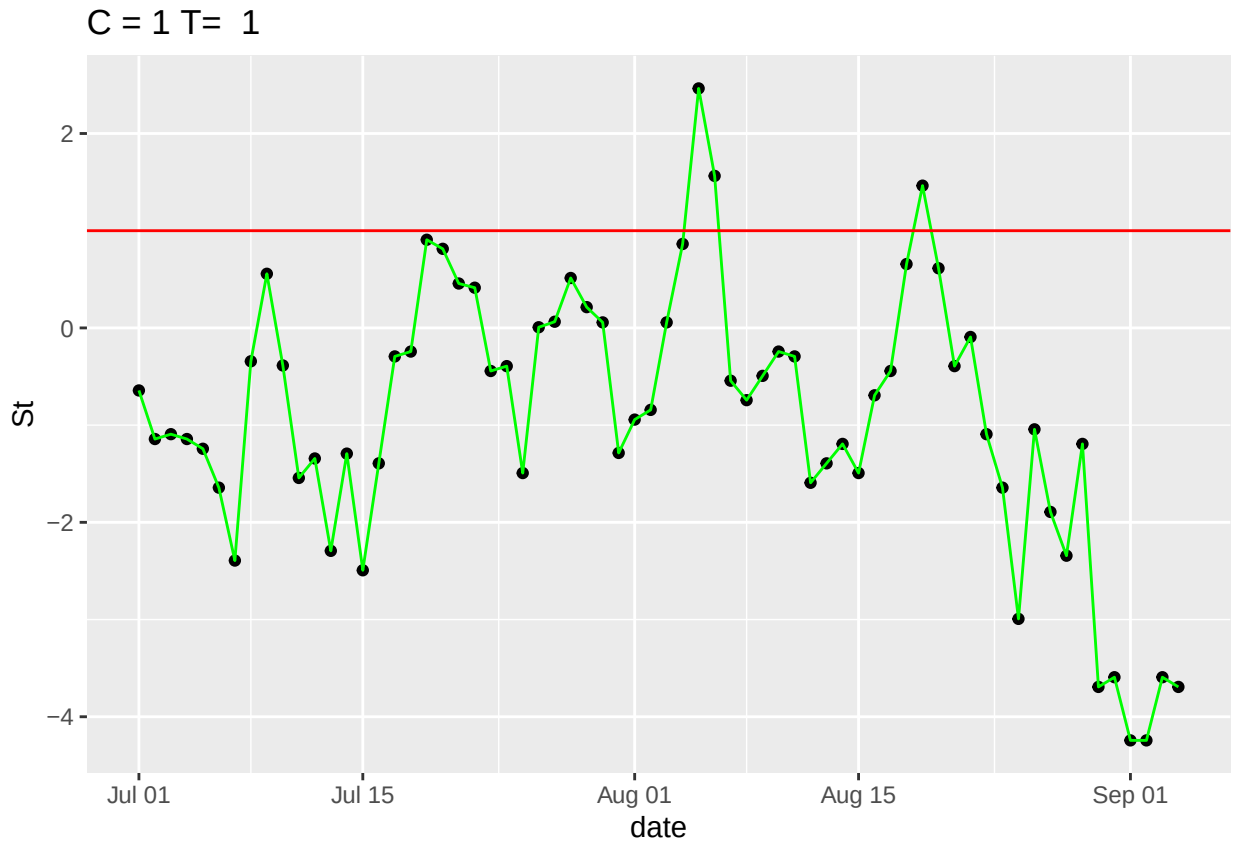
```
## 46 2023-08-15 88.00 -0.49318182 -0.49318182
## 47 2023-08-16 88.80  0.30681818  0.30681818
## 48 2023-08-17 89.05  0.55681818  0.86363636
## 49 2023-08-18 90.15  1.65681818  2.21363636
## 50 2023-08-19 90.30  1.80681818  3.46363636
## 51 2023-08-20 89.30  0.80681818  2.61363636
## 52 2023-08-21 89.10  0.60681818  1.41363636
## 53 2023-08-22 89.40  0.90681818  1.51363636
## 54 2023-08-23 88.40 -0.09318182  0.81363636
## 55 2023-08-24 87.85 -0.64318182 -0.64318182
## 56 2023-08-25 86.50 -1.99318182 -1.99318182
## 57 2023-08-26 88.45 -0.04318182 -0.04318182
## 58 2023-08-27 87.60 -0.89318182 -0.89318182
## 59 2023-08-28 87.15 -1.34318182 -1.34318182
## 60 2023-08-29 88.30 -0.19318182 -0.19318182
## 61 2023-08-30 85.80 -2.69318182 -2.69318182
## 62 2023-08-31 85.90 -2.59318182 -2.59318182
## 63 2023-09-01 85.25 -3.24318182 -3.24318182
## 64 2023-09-02 85.25 -3.24318182 -3.24318182
## 65 2023-09-03 85.90 -2.59318182 -2.59318182
## 66 2023-09-04 85.80 -2.69318182 -2.69318182
```




```
##          date  mean       helper           St
## 1  2023-07-01 88.85 -0.643181818 -0.643181818
## 2  2023-07-02 88.35 -1.143181818 -1.143181818
## 3  2023-07-03 88.40 -1.093181818 -1.093181818
```

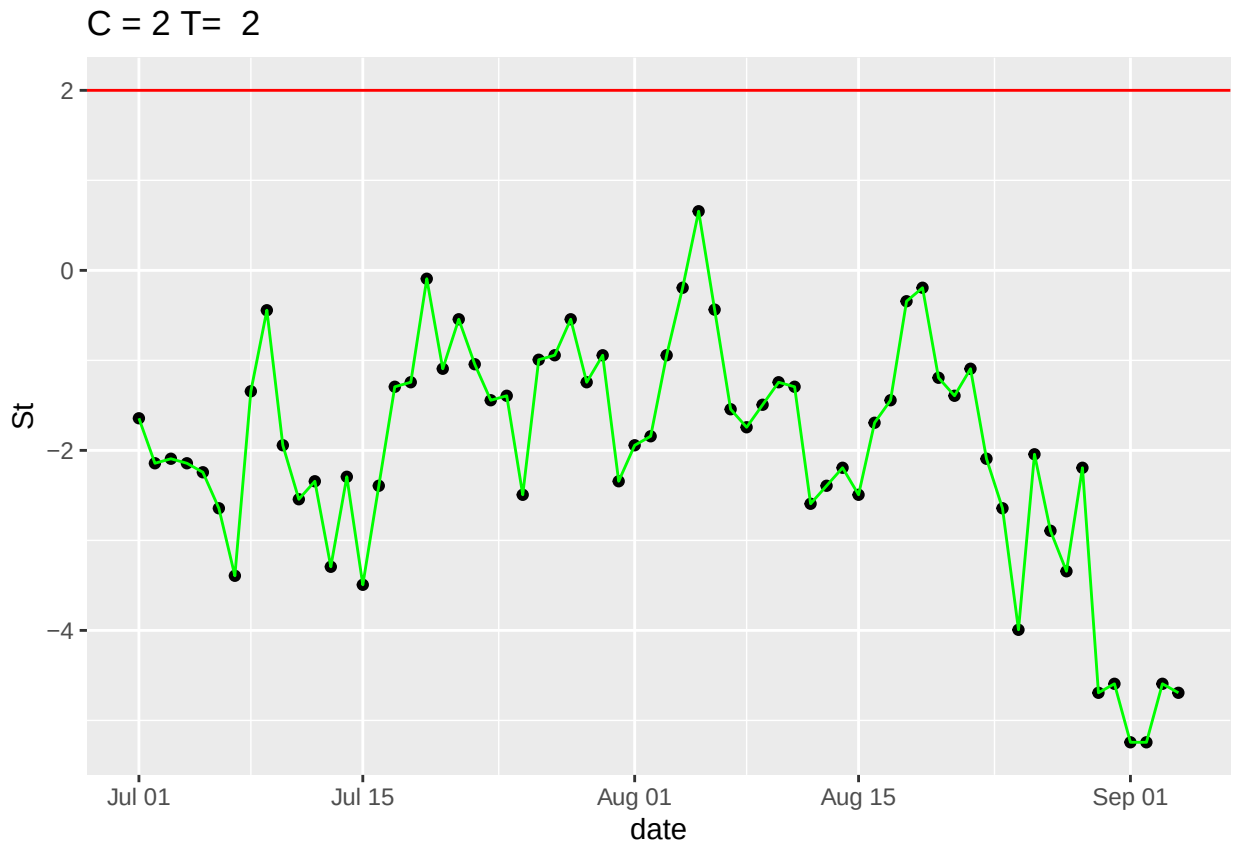
```
## 4  2023-07-04 88.35 -1.143181818 -1.143181818
## 5  2023-07-05 88.25 -1.243181818 -1.243181818
## 6  2023-07-06 87.85 -1.643181818 -1.643181818
## 7  2023-07-07 87.10 -2.393181818 -2.393181818
## 8  2023-07-08 89.15 -0.343181818 -0.343181818
## 9  2023-07-09 90.05  0.556818182  0.556818182
## 10 2023-07-10 88.55 -0.943181818 -0.386363636
## 11 2023-07-11 87.95 -1.543181818 -1.543181818
## 12 2023-07-12 88.15 -1.343181818 -1.343181818
## 13 2023-07-13 87.20 -2.293181818 -2.293181818
## 14 2023-07-14 88.20 -1.293181818 -1.293181818
## 15 2023-07-15 87.00 -2.493181818 -2.493181818
## 16 2023-07-16 88.10 -1.393181818 -1.393181818
## 17 2023-07-17 89.20 -0.293181818 -0.293181818
## 18 2023-07-18 89.25 -0.243181818 -0.243181818
## 19 2023-07-19 90.40  0.906818182  0.906818182
## 20 2023-07-20 89.40 -0.093181818  0.813636364
## 21 2023-07-21 89.95  0.456818182  0.456818182
## 22 2023-07-22 89.45 -0.043181818  0.413636364
## 23 2023-07-23 89.05 -0.443181818 -0.443181818
## 24 2023-07-24 89.10 -0.393181818 -0.393181818
## 25 2023-07-25 88.00 -1.493181818 -1.493181818
## 26 2023-07-26 89.50  0.006818182  0.006818182
## 27 2023-07-27 89.55  0.056818182  0.063636364
## 28 2023-07-28 89.95  0.456818182  0.513636364
## 29 2023-07-29 89.25 -0.243181818  0.213636364
## 30 2023-07-30 89.55  0.056818182  0.056818182
## 31 2023-07-31 88.15 -1.343181818 -1.286363636
## 32 2023-08-01 88.55 -0.943181818 -0.943181818
## 33 2023-08-02 88.65 -0.843181818 -0.843181818
## 34 2023-08-03 89.55  0.056818182  0.056818182
## 35 2023-08-04 90.30  0.806818182  0.863636364
## 36 2023-08-05 91.15  1.656818182  2.463636364
## 37 2023-08-06 89.40 -0.093181818  1.563636364
## 38 2023-08-07 88.95 -0.543181818 -0.543181818
## 39 2023-08-08 88.75 -0.743181818 -0.743181818
## 40 2023-08-09 89.00 -0.493181818 -0.493181818
## 41 2023-08-10 89.25 -0.243181818 -0.243181818
## 42 2023-08-11 89.20 -0.293181818 -0.293181818
## 43 2023-08-12 87.90 -1.593181818 -1.593181818
## 44 2023-08-13 88.10 -1.393181818 -1.393181818
## 45 2023-08-14 88.30 -1.193181818 -1.193181818
## 46 2023-08-15 88.00 -1.493181818 -1.493181818
## 47 2023-08-16 88.80 -0.693181818 -0.693181818
## 48 2023-08-17 89.05 -0.443181818 -0.443181818
## 49 2023-08-18 90.15  0.656818182  0.656818182
## 50 2023-08-19 90.30  0.806818182  1.463636364
## 51 2023-08-20 89.30 -0.193181818  0.613636364
## 52 2023-08-21 89.10 -0.393181818 -0.393181818
## 53 2023-08-22 89.40 -0.093181818 -0.093181818
## 54 2023-08-23 88.40 -1.093181818 -1.093181818
## 55 2023-08-24 87.85 -1.643181818 -1.643181818
## 56 2023-08-25 86.50 -2.993181818 -2.993181818
## 57 2023-08-26 88.45 -1.043181818 -1.043181818
```

```
## 58 2023-08-27 87.60 -1.893181818 -1.893181818
## 59 2023-08-28 87.15 -2.343181818 -2.343181818
## 60 2023-08-29 88.30 -1.193181818 -1.193181818
## 61 2023-08-30 85.80 -3.693181818 -3.693181818
## 62 2023-08-31 85.90 -3.593181818 -3.593181818
## 63 2023-09-01 85.25 -4.243181818 -4.243181818
## 64 2023-09-02 85.25 -4.243181818 -4.243181818
## 65 2023-09-03 85.90 -3.593181818 -3.593181818
## 66 2023-09-04 85.80 -3.693181818 -3.693181818
```



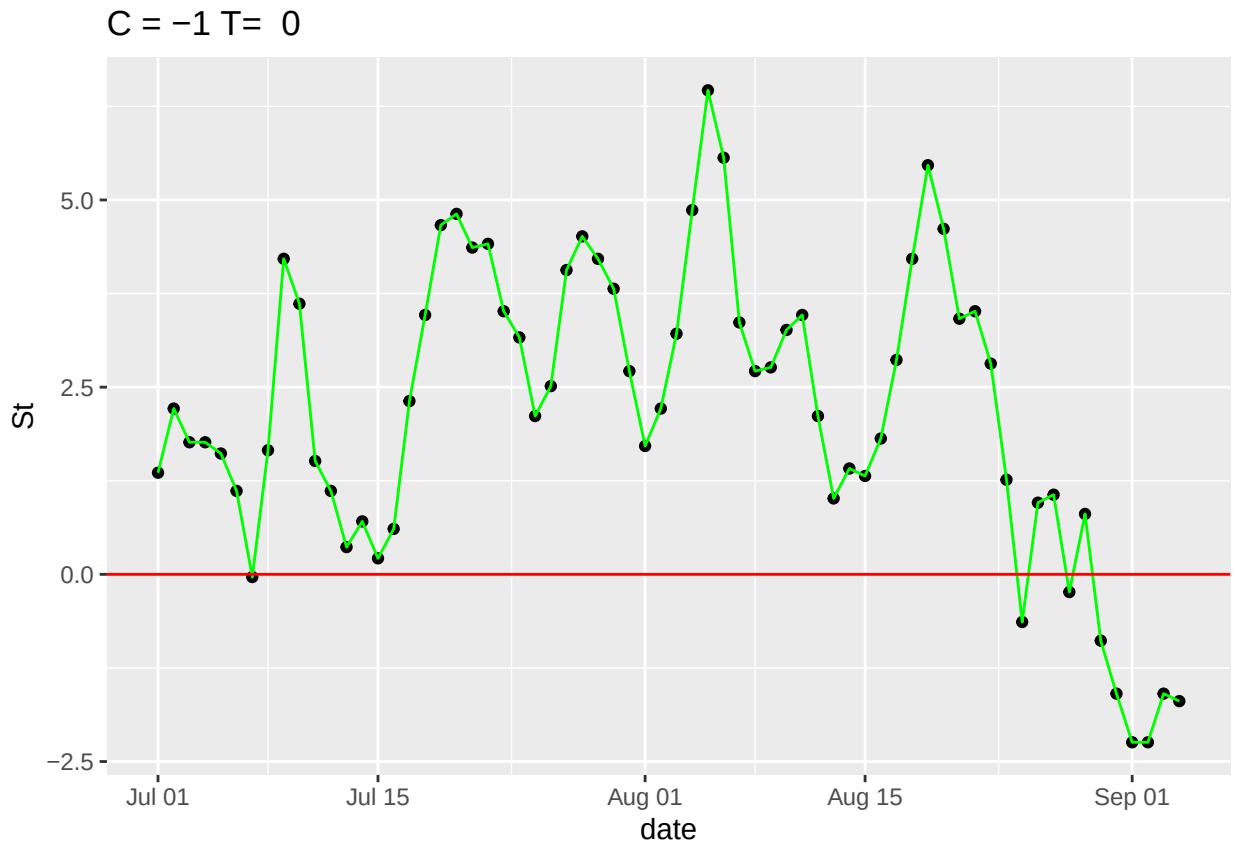

```
##          date  mean      helper          St
## 1  2023-07-01 88.85 -1.64318182 -1.64318182
## 2  2023-07-02 88.35 -2.14318182 -2.14318182
## 3  2023-07-03 88.40 -2.09318182 -2.09318182
## 4  2023-07-04 88.35 -2.14318182 -2.14318182
## 5  2023-07-05 88.25 -2.24318182 -2.24318182
## 6  2023-07-06 87.85 -2.64318182 -2.64318182
## 7  2023-07-07 87.10 -3.39318182 -3.39318182
## 8  2023-07-08 89.15 -1.34318182 -1.34318182
## 9  2023-07-09 90.05 -0.44318182 -0.44318182
## 10 2023-07-10 88.55 -1.94318182 -1.94318182
## 11 2023-07-11 87.95 -2.54318182 -2.54318182
## 12 2023-07-12 88.15 -2.34318182 -2.34318182
## 13 2023-07-13 87.20 -3.29318182 -3.29318182
## 14 2023-07-14 88.20 -2.29318182 -2.29318182
## 15 2023-07-15 87.00 -3.49318182 -3.49318182
```

```
## 16 2023-07-16 88.10 -2.39318182 -2.39318182
## 17 2023-07-17 89.20 -1.29318182 -1.29318182
## 18 2023-07-18 89.25 -1.24318182 -1.24318182
## 19 2023-07-19 90.40 -0.09318182 -0.09318182
## 20 2023-07-20 89.40 -1.09318182 -1.09318182
## 21 2023-07-21 89.95 -0.54318182 -0.54318182
## 22 2023-07-22 89.45 -1.04318182 -1.04318182
## 23 2023-07-23 89.05 -1.44318182 -1.44318182
## 24 2023-07-24 89.10 -1.39318182 -1.39318182
## 25 2023-07-25 88.00 -2.49318182 -2.49318182
## 26 2023-07-26 89.50 -0.99318182 -0.99318182
## 27 2023-07-27 89.55 -0.94318182 -0.94318182
## 28 2023-07-28 89.95 -0.54318182 -0.54318182
## 29 2023-07-29 89.25 -1.24318182 -1.24318182
## 30 2023-07-30 89.55 -0.94318182 -0.94318182
## 31 2023-07-31 88.15 -2.34318182 -2.34318182
## 32 2023-08-01 88.55 -1.94318182 -1.94318182
## 33 2023-08-02 88.65 -1.84318182 -1.84318182
## 34 2023-08-03 89.55 -0.94318182 -0.94318182
## 35 2023-08-04 90.30 -0.19318182 -0.19318182
## 36 2023-08-05 91.15  0.65681818  0.65681818
## 37 2023-08-06 89.40 -1.09318182 -0.43636364
## 38 2023-08-07 88.95 -1.54318182 -1.54318182
## 39 2023-08-08 88.75 -1.74318182 -1.74318182
## 40 2023-08-09 89.00 -1.49318182 -1.49318182
## 41 2023-08-10 89.25 -1.24318182 -1.24318182
## 42 2023-08-11 89.20 -1.29318182 -1.29318182
## 43 2023-08-12 87.90 -2.59318182 -2.59318182
## 44 2023-08-13 88.10 -2.39318182 -2.39318182
## 45 2023-08-14 88.30 -2.19318182 -2.19318182
## 46 2023-08-15 88.00 -2.49318182 -2.49318182
## 47 2023-08-16 88.80 -1.69318182 -1.69318182
## 48 2023-08-17 89.05 -1.44318182 -1.44318182
## 49 2023-08-18 90.15 -0.34318182 -0.34318182
## 50 2023-08-19 90.30 -0.19318182 -0.19318182
## 51 2023-08-20 89.30 -1.19318182 -1.19318182
## 52 2023-08-21 89.10 -1.39318182 -1.39318182
## 53 2023-08-22 89.40 -1.09318182 -1.09318182
## 54 2023-08-23 88.40 -2.09318182 -2.09318182
## 55 2023-08-24 87.85 -2.64318182 -2.64318182
## 56 2023-08-25 86.50 -3.99318182 -3.99318182
## 57 2023-08-26 88.45 -2.04318182 -2.04318182
## 58 2023-08-27 87.60 -2.89318182 -2.89318182
## 59 2023-08-28 87.15 -3.34318182 -3.34318182
## 60 2023-08-29 88.30 -2.19318182 -2.19318182
## 61 2023-08-30 85.80 -4.69318182 -4.69318182
## 62 2023-08-31 85.90 -4.59318182 -4.59318182
## 63 2023-09-01 85.25 -5.24318182 -5.24318182
## 64 2023-09-02 85.25 -5.24318182 -5.24318182
## 65 2023-09-03 85.90 -4.59318182 -4.59318182
## 66 2023-09-04 85.80 -4.69318182 -4.69318182
```




```
##          date  mean     helper          St
## 1  2023-07-01 88.85  1.3568182  1.35681818
## 2  2023-07-02 88.35  0.8568182  2.21363636
## 3  2023-07-03 88.40  0.9068182  1.76363636
## 4  2023-07-04 88.35  0.8568182  1.76363636
## 5  2023-07-05 88.25  0.7568182  1.61363636
## 6  2023-07-06 87.85  0.3568182  1.11363636
## 7  2023-07-07 87.10 -0.3931818 -0.03636364
## 8  2023-07-08 89.15  1.6568182  1.65681818
## 9  2023-07-09 90.05  2.5568182  4.21363636
## 10 2023-07-10 88.55  1.0568182  3.61363636
## 11 2023-07-11 87.95  0.4568182  1.51363636
## 12 2023-07-12 88.15  0.6568182  1.11363636
## 13 2023-07-13 87.20 -0.2931818  0.36363636
## 14 2023-07-14 88.20  0.7068182  0.70681818
## 15 2023-07-15 87.00 -0.4931818  0.21363636
## 16 2023-07-16 88.10  0.6068182  0.60681818
## 17 2023-07-17 89.20  1.7068182  2.31363636
## 18 2023-07-18 89.25  1.7568182  3.46363636
## 19 2023-07-19 90.40  2.9068182  4.66363636
## 20 2023-07-20 89.40  1.9068182  4.81363636
## 21 2023-07-21 89.95  2.4568182  4.36363636
## 22 2023-07-22 89.45  1.9568182  4.41363636
## 23 2023-07-23 89.05  1.5568182  3.51363636
## 24 2023-07-24 89.10  1.6068182  3.16363636
## 25 2023-07-25 88.00  0.5068182  2.11363636
```

```
## 26 2023-07-26 89.50  2.0068182  2.51363636
## 27 2023-07-27 89.55  2.0568182  4.06363636
## 28 2023-07-28 89.95  2.4568182  4.51363636
## 29 2023-07-29 89.25  1.7568182  4.21363636
## 30 2023-07-30 89.55  2.0568182  3.81363636
## 31 2023-07-31 88.15  0.6568182  2.71363636
## 32 2023-08-01 88.55  1.0568182  1.71363636
## 33 2023-08-02 88.65  1.1568182  2.21363636
## 34 2023-08-03 89.55  2.0568182  3.21363636
## 35 2023-08-04 90.30  2.8068182  4.86363636
## 36 2023-08-05 91.15  3.6568182  6.46363636
## 37 2023-08-06 89.40  1.9068182  5.56363636
## 38 2023-08-07 88.95  1.4568182  3.36363636
## 39 2023-08-08 88.75  1.2568182  2.71363636
## 40 2023-08-09 89.00  1.5068182  2.76363636
## 41 2023-08-10 89.25  1.7568182  3.26363636
## 42 2023-08-11 89.20  1.7068182  3.46363636
## 43 2023-08-12 87.90  0.4068182  2.11363636
## 44 2023-08-13 88.10  0.6068182  1.01363636
## 45 2023-08-14 88.30  0.8068182  1.41363636
## 46 2023-08-15 88.00  0.5068182  1.31363636
## 47 2023-08-16 88.80  1.3068182  1.81363636
## 48 2023-08-17 89.05  1.5568182  2.86363636
## 49 2023-08-18 90.15  2.6568182  4.21363636
## 50 2023-08-19 90.30  2.8068182  5.46363636
## 51 2023-08-20 89.30  1.8068182  4.61363636
## 52 2023-08-21 89.10  1.6068182  3.41363636
## 53 2023-08-22 89.40  1.9068182  3.51363636
## 54 2023-08-23 88.40  0.9068182  2.81363636
## 55 2023-08-24 87.85  0.3568182  1.26363636
## 56 2023-08-25 86.50 -0.9931818 -0.63636364
## 57 2023-08-26 88.45  0.9568182  0.95681818
## 58 2023-08-27 87.60  0.1068182  1.06363636
## 59 2023-08-28 87.15 -0.3431818 -0.23636364
## 60 2023-08-29 88.30  0.8068182  0.80681818
## 61 2023-08-30 85.80 -1.6931818 -0.88636364
## 62 2023-08-31 85.90 -1.5931818 -1.59318182
## 63 2023-09-01 85.25 -2.2431818 -2.24318182
## 64 2023-09-02 85.25 -2.2431818 -2.24318182
## 65 2023-09-03 85.90 -1.5931818 -1.59318182
## 66 2023-09-04 85.80 -1.6931818 -1.69318182
```



From here, I inspected the different graphs and say that T=0 and C= -1 gave me the best respoes. C=-1 allowed me to align the graph with the lowest temperature in July, and when the temperature was beginning to plummet in August. July is officially summer, so when the temperature drops below the July mark and stays below, the summer is over.. According to the graph, this was about August 28. Thus, the official end of summer is August 28.

Part 2

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).

Part 2 Solution

To start this question, I had to decided how I would approach it. The summer climate has great variation, and I didn't want to pick just one date and risk losing valuable information. I decided to chose the median date for each summer through the years. I chose the median because the mean could be skewed by heat waves or cold spells.

I can see how this is not a perfect method for this calculation, but it will work in this situation.

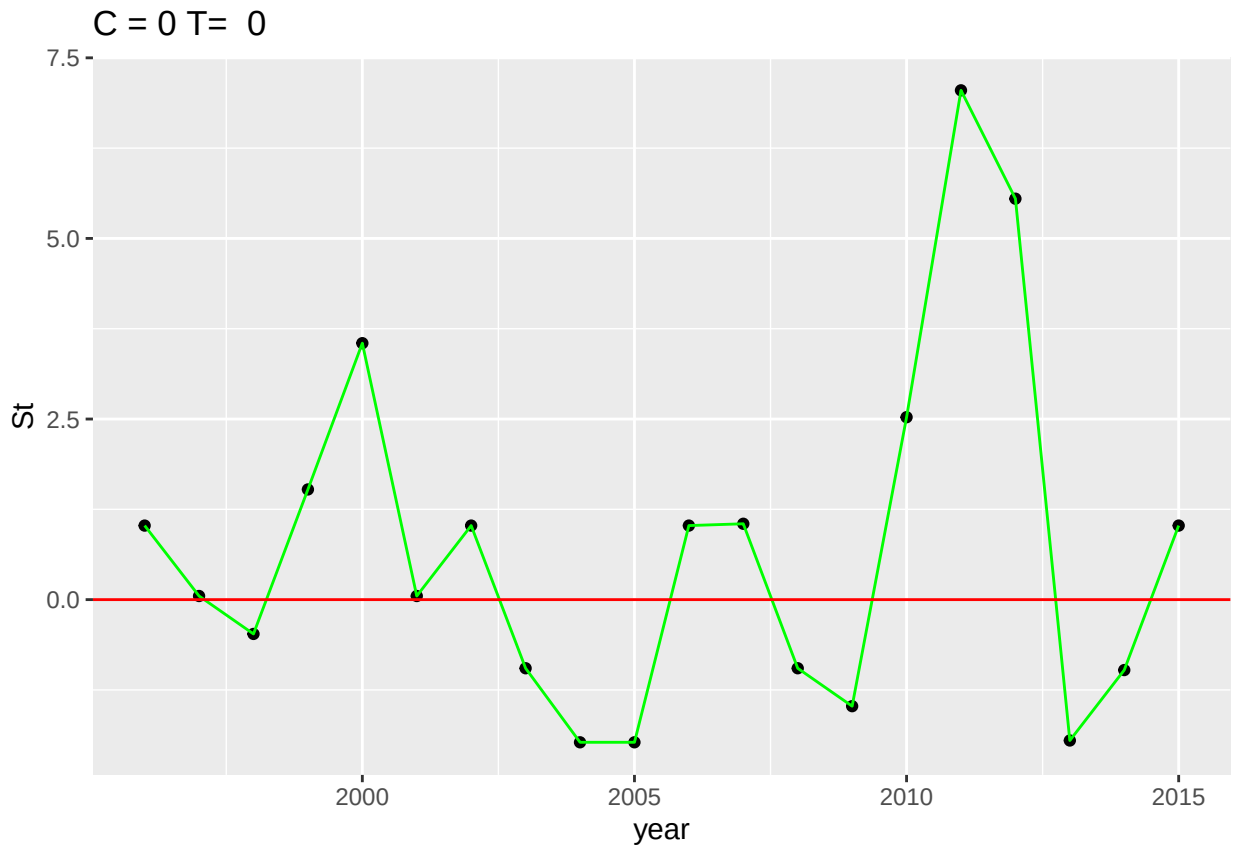
I started by transforming the data to be in a similar to what was done in the previous problem.

```
##       year median
## X1996 1996   90.0
## X1997 1997   88.0
## X1998 1998   88.5
```

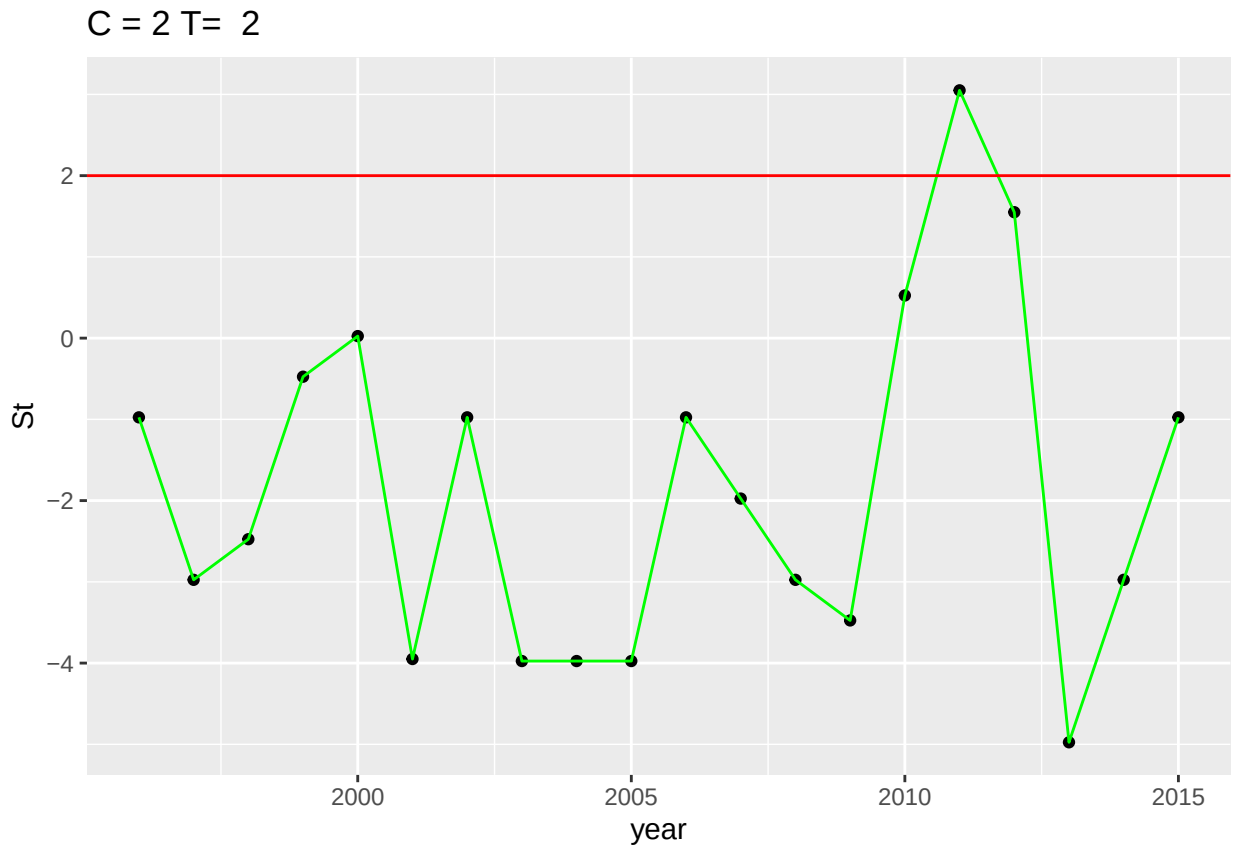
```
## X1999 1999   90.5
## X2000 2000   91.0
## X2001 2001   87.0
## X2002 2002   90.0
## X2003 2003   87.0
## X2004 2004   87.0
## X2005 2005   87.0
## X2006 2006   90.0
## X2007 2007   89.0
## X2008 2008   88.0
## X2009 2009   87.5
## X2010 2010   91.5
## X2011 2011   93.5
## X2012 2012   90.0
## X2013 2013   86.0
## X2014 2014   88.0
## X2015 2015   90.0
```

We now have a data frame that has the median temperature across the summer for all years. Using the function from the last question, I created a cusum function.

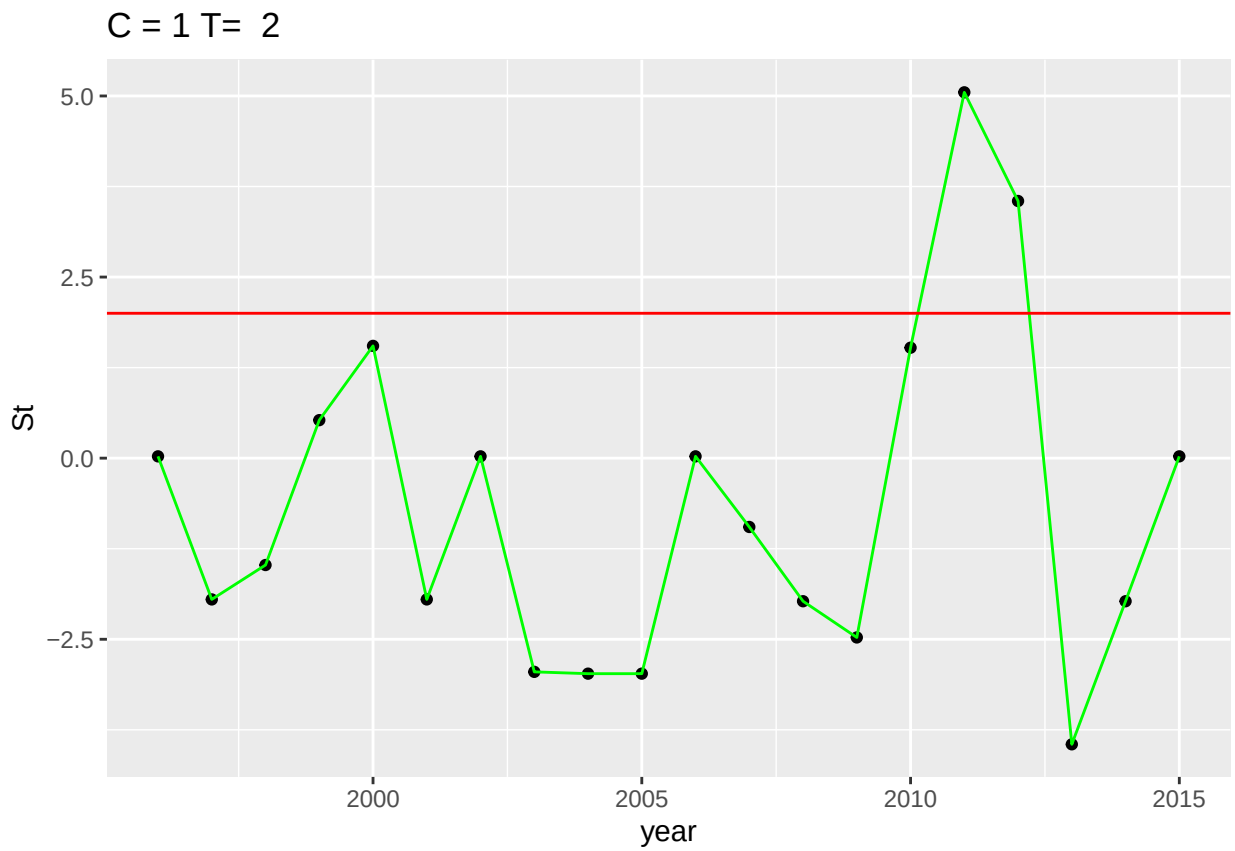
```
##       year median helper     St
## X1996 1996   90.0  1.025  1.025
## X1997 1997   88.0 -0.975  0.050
## X1998 1998   88.5 -0.475 -0.475
## X1999 1999   90.5  1.525  1.525
## X2000 2000   91.0  2.025  3.550
## X2001 2001   87.0 -1.975  0.050
```




```
##       year median helper     St
## X1996 1996   90.0 -0.975 -0.975
## X1997 1997   88.0 -2.975 -2.975
## X1998 1998   88.5 -2.475 -2.475
## X1999 1999   90.5 -0.475 -0.475
## X2000 2000   91.0  0.025  0.025
## X2001 2001   87.0 -3.975 -3.950
```




```
##       year median helper     St
## X1996 1996   90.0  0.025  0.025
## X1997 1997   88.0 -1.975 -1.950
## X1998 1998   88.5 -1.475 -1.475
## X1999 1999   90.5  0.525  0.525
## X2000 2000   91.0  1.025  1.550
## X2001 2001   87.0 -2.975 -1.950
```



The plots here are strange. The plot I most like is the one where C=1 and T=2. We see a huge spike in 2011, crossing the arbitrary threshold, but then it dips back down and seems to be rising again. The timeline is inconclusive for this period.

What is instead of taking the median of all days, I chose one day in each year? I start with a new dataset, holding a list of data frames

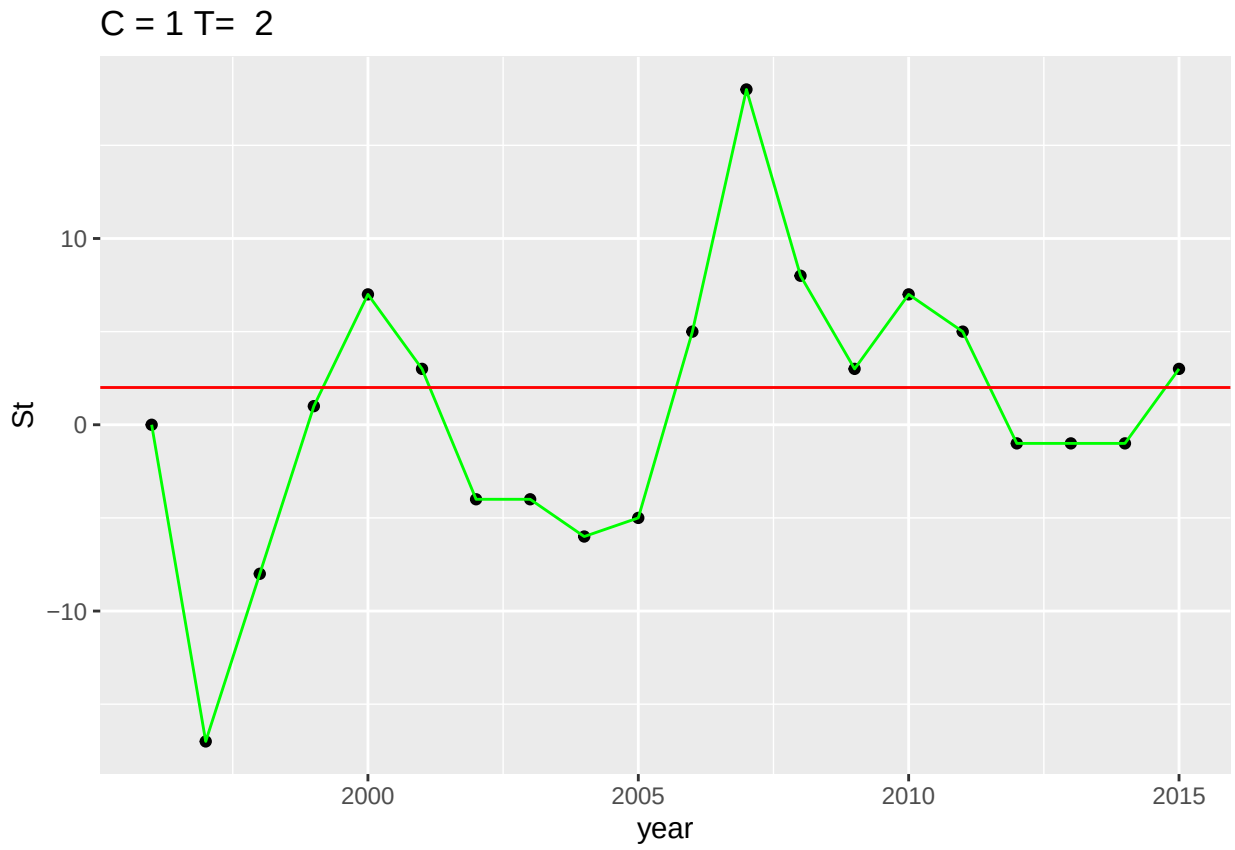
```
## [[1]]
## [[1]][[1]]
##       year X9.Aug
## X1996 1996     90
## X1997 1997     73
## X1998 1998     82
## X1999 1999     91
## X2000 2000     96
## X2001 2001     87
## X2002 2002     86
## X2003 2003     86
## X2004 2004     84
## X2005 2005     85
## X2006 2006     95
## X2007 2007    103
## X2008 2008     85
## X2009 2009     93
## X2010 2010     94
## X2011 2011     91
## X2012 2012     88
```

```
## X2013 2013     89
## X2014 2014     89
## X2015 2015     93
##
##
## [[2]]
##       year X20.Jul
## X1996 1996      99
## X1997 1997      90
## X1998 1998      91
## X1999 1999      90
## X2000 2000      99
## X2001 2001      87
## X2002 2002      91
## X2003 2003      88
## X2004 2004      88
## X2005 2005      89
## X2006 2006      93
## X2007 2007      86
## X2008 2008      94
## X2009 2009      82
## X2010 2010      91
## X2011 2011      91
## X2012 2012      84
## X2013 2013      86
## X2014 2014      76
## X2015 2015      93
##
## [[3]]
##       year X24.Aug
## X1996 1996      88
## X1997 1997      91
## X1998 1998      93
## X1999 1999      91
## X2000 2000      92
## X2001 2001      86
## X2002 2002      82
## X2003 2003      89
## X2004 2004      87
## X2005 2005      85
## X2006 2006      90
## X2007 2007      89
## X2008 2008      89
## X2009 2009      86
## X2010 2010      84
## X2011 2011      93
## X2012 2012      86
## X2013 2013      92
## X2014 2014      92
## X2015 2015      81
```

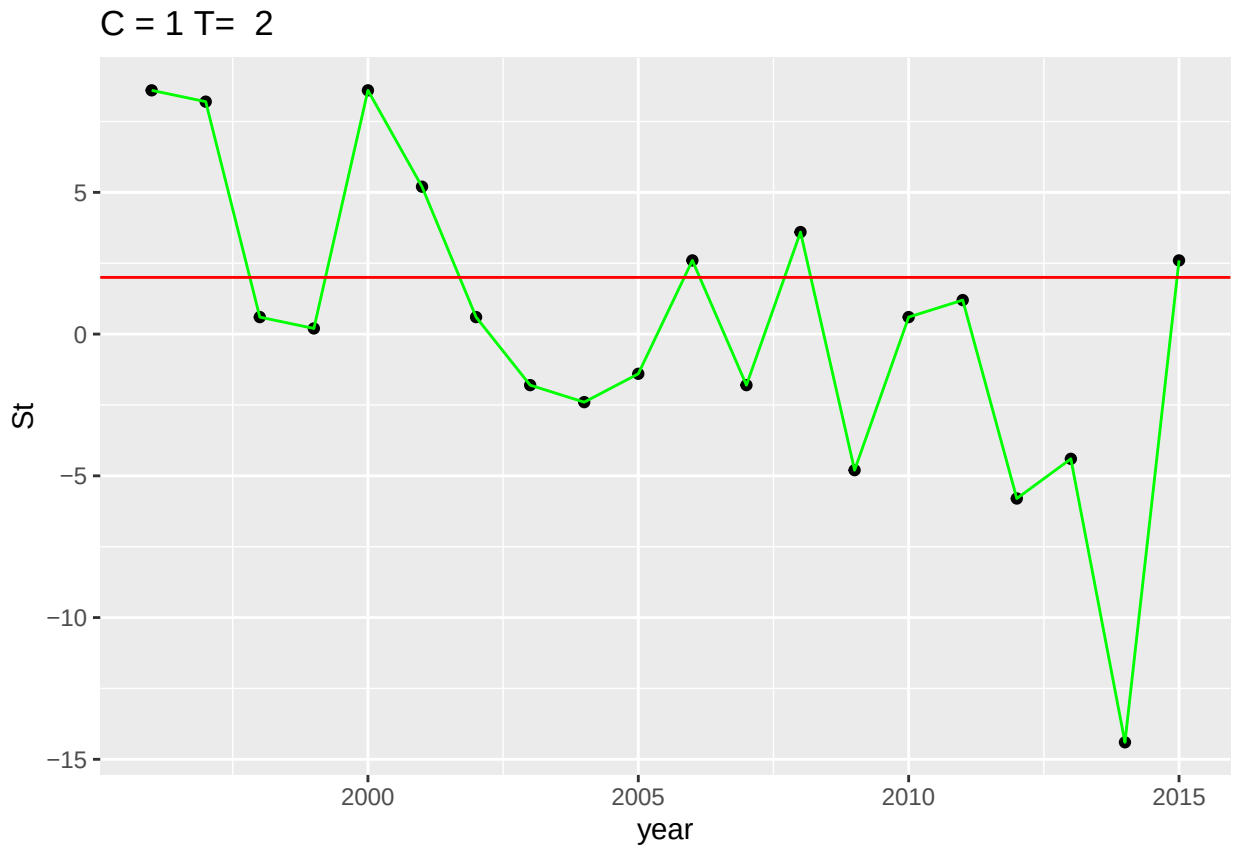
I can then use my previous function and apply it to three random dates

```
##       year median helper  St
```

```
## X1996 1996   90.0      0   0
## X1997 1997   88.0    -17 -17
## X1998 1998   88.5     -8  -8
## X1999 1999   90.5      1   1
## X2000 2000   91.0      6   7
## X2001 2001   87.0     -3   3
```




```
##       year median helper  St
## X1996 1996   90.0    8.6 8.6
## X1997 1997   88.0   -0.4 8.2
## X1998 1998   88.5    0.6 0.6
## X1999 1999   90.5   -0.4 0.2
## X2000 2000   91.0    8.6 8.6
## X2001 2001   87.0   -3.4 5.2
```




```
##       year median helper   St
## X1996 1996   90.0   -1.3 -1.3
## X1997 1997   88.0    1.7  1.7
## X1998 1998   88.5    3.7  5.4
## X1999 1999   90.5    1.7  5.4
## X2000 2000   91.0    2.7  4.4
## X2001 2001   87.0   -3.3 -0.6
```

Again, the data is inconclusive. I would love to say the world is getting warmer, but we are not showing data for enough years to make any well-informed decisions.