ISYE 6501 HW 4

2024-02-04

Question 7.1

#Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. #What data would you need? Would you expect the value of ⍰ (the first smoothing parameter) to be closer to 0 or 1, and why?

Before starting this Spring in this program, I worked as a Sales Associate at Home Depot in the Garden department for almost a year. This department was always in charge of stocking holiday decorations for Halloween and Christmas. For Halloween, we always carried large inflatable decorations you could put in your front yard. However, for the last several years, supply has not matched demand for the inflatable decorations. There were several occasions where a customer would complain that our store, and all the other nearby locations, were out of a certain inflatable decoration.

In this case, each store only received a small quantity of the inflatable decorations, which usually sold out within the first two days. As a result, there would be many angry customers who were hoping for one of the inflatable decorations but none of the nearby locations had any, and online inventory was limited.

An exponential smoothing model for holiday decorations could be employed so Home Depot does not miss out on potential sales of large inflatable decorations. For the last several years, there have been numerous customer complaints about limited inventory of the decorations, and yet the shortage continued this past year. A value of the smoothing parameter would need to be close to 1, in order to weight recent missed sales on holiday inflatables to better be prepared for the holiday rush in the future. With higher weights, these recent observations could prompt Home Depot to get a larger quantity of inflatable decorations for Halloween so they do not miss out on more potential sales in the future, including someone like me who really wants a blowup Grim Reaper if I buy a house in the future.

Question 7.2

#Using the 20 years of daily high temperature data for Atlanta (July through October) from Question 6.2 (file temps.txt), build and use an #exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years. (Part of #the point of this assignment is for you to think about how you might use exponential smoothing to answer this question. Feel free to #combine it with other models if you'd like to. There's certainly more than one reasonable approach.)

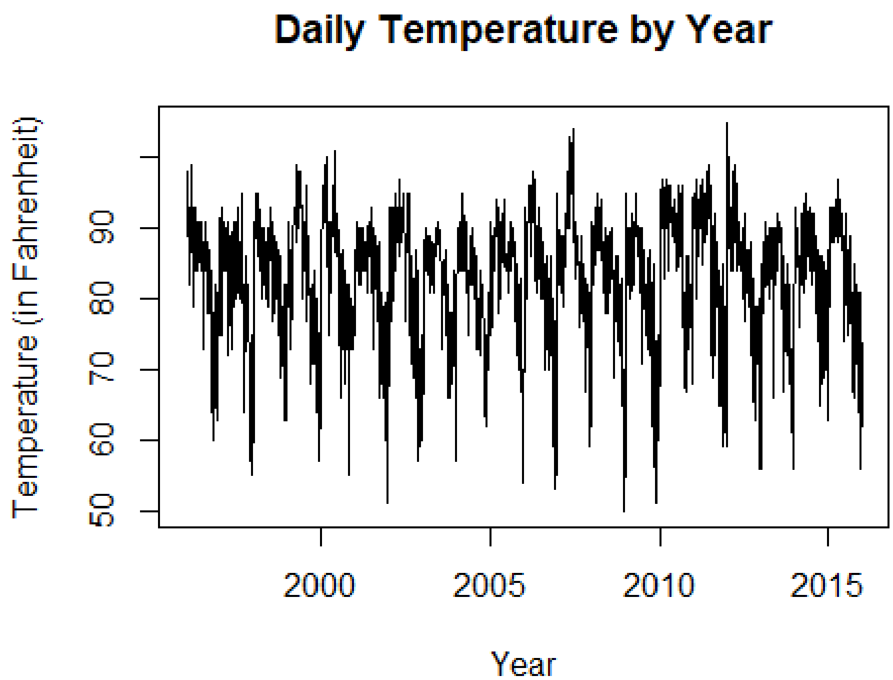
#Note: in R, you can use either HoltWinters (simpler to use) or the smooth package's es function (harder to use, but more general). If you #use es, the Holt-Winters model uses model=”AAM” in the function call (the first and second constants are used “A”dditively, and the third #(seasonality) is used “M”ultiplicatively; the documentation doesn’t make that clear).

The temps data set had to be transformed into a matrix and then converted into a time series object before creating an exponential smoothing model.

```
#read in the temps data set

Temperature <- read.table("/Users/ryanc/Downloads/temps.txt", header = TRUE)

#activate the necessary libraries
library(tidyverse)

library(tidyr)
library(dplyr)
library(lubridate)
library(ggplot2)
library(qcc)

library(plotly)

install.packages("tseries", repos = 'http://cran.us.r-project.org')

library(tseries)

install.packages("matrixStats", repos = 'http://cran.us.r-project.org')
```

I created a graph of the newly created time series object to get a feel for the data. From the graph, it appears that some years have larger fluctuations between July and October than others, especially after around 2009. However, I will try a single, double,, and triple exponential smoothing model before determining whether the end of summer has gotten later or not.

```
#convert the temperature data set into a vector to make it easier to convert
to a time series object
temperature_time_series <- as.vector(unlist(Temperature[,2:21]))

#convert to a time series object
temperature_time_series <- ts(data = temperature_time_series, start = 1996,
frequency = 123)

#see what the time series object looks like before creating a Holt-Winters
model
plot(temperature_time_series, ylab = "Temperature (in Fahrenheit)", xlab =
"Year", main = "Daily Temperature by Year")
```



I first attempted a single exponential smoothing model without trend or season. The alpha parameter had a high value above 0.8, meaning a ton of weight was being placed on more recent observations and the sum of squared errors (SSE) was 56198. However, since seasonality is a major consideration for determining the end of summer, it is best to consider models with a seasonality component, which are tested below.

```
#created a single exponential smoothing model and obtained the parameter and SSE values
Temperature_time_series_single <- HoltWinters(temperature_time_series, alpha=NULL, beta = FALSE, gamma = FALSE, seasonal ="additive")
Temperature_time_series_single

## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = temperature_time_series, alpha = NULL, beta = FALSE, gamma = FALSE, seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.8388021
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 63.30952
```

```
Temperature_time_series_single$SSE

## [1] 56198.1
```

The next model attempted was a double exponential smoothing model with a seasonal component and no trend. The alpha value was 0.66 and the gamma was 0.62. This means that the weight being placed on recent observations decreased. The SSE increased to 66244.54 as well but, since seasonality is a critical piece of the puzzle, I would choose this model over the single exponential smoothing model However, a lower SSE would be nice to have, so I tried a triple exponential smoothing model next.

```
#created a double exponential smoothing model with the addition of
seasonality. Parameter and error values are obtained
Temperature_time_series_double <- HoltWinters(temperature_time_series, alpha=
NULL, beta = FALSE, gamma = NULL, seasonal ="additive")
Temperature_time_series_double

## Holt-Winters exponential smoothing without trend and with additive
seasonal component.
##
## Call:
## HoltWinters(x = temperature_time_series, alpha = NULL, beta = FALSE,
gamma = NULL, seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.6610655
##  beta : FALSE
##  gamma: 0.624816
##
## Coefficients:
##             [,1]
## a     71.50455197
## s1    18.56504884
## s2    17.77792614
## s3    12.17913958
## s4    13.20850812
## s5    12.93164407
## s6    11.49954644
## s7    10.82847422
## s8    10.17350202
## s9     8.66850548
## s10    5.95671228
## s11    3.09704804
## s12    4.67174217
## s13    2.70350639
## s14    2.96939841
## s15    1.68805250
## s16    2.46014746
## s17    6.35605409
## s18    5.05529390
```

```
## s19    7.54489486
## s20    6.13851326
## s21    9.53392463
## s22    9.67360032
## s23    8.78183832
## s24    8.47895005
## s25    7.38024099
## s26    6.81263122
## s27    6.34168710
## s28    6.35550869
## s29    4.52549064
## s30    6.85092390
## s31    4.79678176
## s32    4.90533859
## s33    7.08334506
## s34    6.15191210
## s35    4.86032114
## s36    3.86396258
## s37    2.12173938
## s38    2.49830109
## s39    2.98153746
## s40    3.01510960
## s41    2.22518849
## s42    0.07453182
## s43   -0.14989789
## s44   -1.47225178
## s45   -1.82934308
## s46   -2.21860459
## s47   -0.20750788
## s48    1.51244778
## s49    5.04886273
## s50    6.71444925
## s51    7.71055864
## s52    8.55298247
## s53    8.38229339
## s54    4.67841989
## s55    1.80064706
## s56   -1.30232515
## s57    1.36333013
## s58    1.35027764
## s59    0.48298493
## s60    1.85987172
## s61   -0.83301232
## s62    5.19534086
## s63    5.35654173
## s64    4.23904546
## s65    3.81493788
## s66   -0.25779222
## s67    0.51619884
## s68    0.75357576
```

```
## s69    1.07013235
## s70    0.66396107
## s71    2.27474544
## s72    2.93935878
## s73    4.36718879
## s74    2.71799836
## s75    1.00873626
## s76    1.14416605
## s77    2.77030014
## s78    1.97376572
## s79   -0.01922276
## s80   -1.23048676
## s81    0.32583813
## s82    0.64854840
## s83   -3.19621115
## s84   -1.93988082
## s85   -1.67433949
## s86   -5.30783230
## s87   -5.15305944
## s88   -2.66421286
## s89   -2.36868892
## s90   -3.30846932
## s91   -4.26858186
## s92   -2.62254480
## s93   -7.84783931
## s94   -8.84129242
## s95   -9.02324314
## s96   -7.86221589
## s97   -5.77569533
## s98   -5.22274144
## s99   -8.65036212
## s100 -11.83593714
## s101 -13.15601816
## s102 -16.12173579
## s103 -15.15201594
## s104 -13.99017684
## s105 -12.97986374
## s106 -16.12373905
## s107 -15.51576781
## s108 -13.70665994
## s109 -11.94796730
## s110 -12.06194608
## s111 -12.86357634
## s112  -9.12232097
## s113  -5.45954883
## s114  -6.82736689
## s115  -8.44018483
## s116 -10.93897178
## s117 -13.58040079
## s118 -10.67911507
```

```
## s119 -12.65385459
## s120  -9.93351665
## s121 -12.69508208
## s122  -9.83204675
## s123  -7.80183886
```

```
Temperature_time_series_double$SSE
```

```
## [1] 66244.54
```

Here, I tried a triple exponential smoothing model with both trend and seasonality included. The trend parameter had a value of 0, but the SSE decreased slightly, to 66244.25. The values for alpha and gamma did not change from those in the double exponential smoothing model above, but due to the reduction in the error, this has the potential to be the model CUSUM will be performed on later. However, for completeness, I tried a triple exponential smoothing model with multiplicative seasonality.

```
#created a triple exponential smoothing model with trend added. Parameter and error values were obtained.
Temperature_time_series_triple <- HoltWinters(temperature_time_series, alpha= NULL, beta = NULL, gamma = NULL, seasonal ="additive")
Temperature_time_series_triple
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = temperature_time_series, alpha = NULL, beta = NULL, gamma = NULL, seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.6610618
##  beta : 0
##  gamma: 0.6248076
##
## Coefficients:
##               [,1]
## a     71.477236414
## b     -0.004362918
## s1    18.590169842
## s2    17.803098732
## s3    12.204442890
## s4    13.233948865
## s5    12.957258705
## s6    11.525341233
## s7    10.854441534
## s8    10.199632666
## s9     8.694767348
## s10    5.983076192
## s11    3.123493477
```

```
## s12    4.698228193
## s13    2.730023168
## s14    2.995935818
## s15    1.714600919
## s16    2.486701224
## s17    6.382595268
## s18    5.081837636
## s19    7.571432660
## s20    6.165047647
## s21    9.560458487
## s22    9.700133847
## s23    8.808383245
## s24    8.505505527
## s25    7.406809208
## s26    6.839204571
## s27    6.368261304
## s28    6.382080380
## s29    4.552058253
## s30    6.877476437
## s31    4.823330209
## s32    4.931885957
## s33    7.109879628
## s34    6.178469084
## s35    4.886891317
## s36    3.890547248
## s37    2.148316257
## s38    2.524866001
## s39    3.008098232
## s40    3.041663870
## s41    2.251741386
## s42    0.101091985
## s43   -0.123337548
## s44   -1.445675315
## s45   -1.802768181
## s46   -2.192036338
## s47   -0.180954242
## s48    1.538987281
## s49    5.075394760
## s50    6.740978049
## s51    7.737089782
## s52    8.579515859
## s53    8.408834158
## s54    4.704976718
## s55    1.827215229
## s56   -1.275747384
## s57    1.389899699
## s58    1.376842871
## s59    0.509553410
## s60    1.886439429
## s61   -0.806454923
```

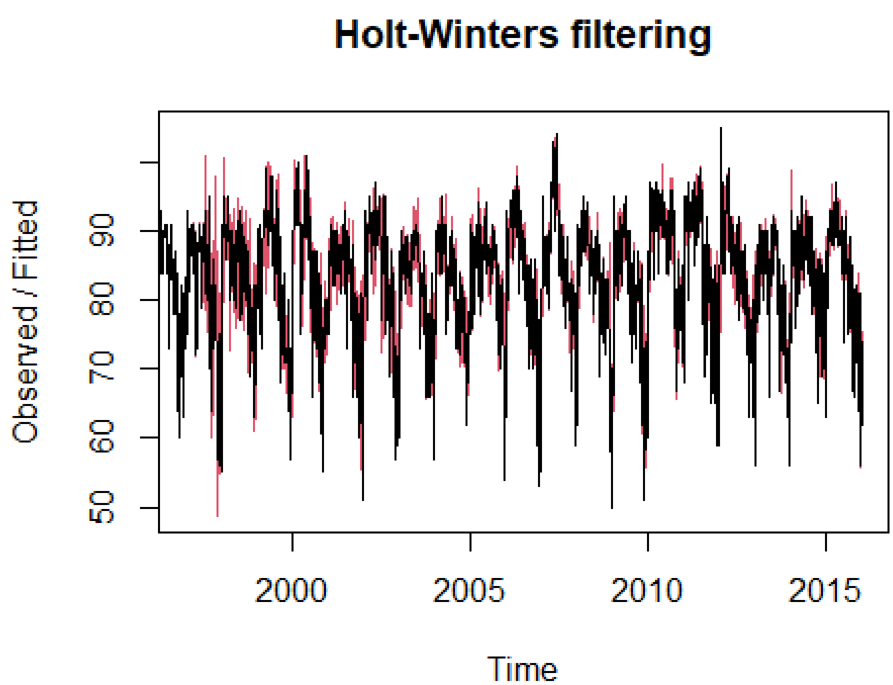
```
## s62    5.221873550
## s63    5.383073482
## s64    4.265584552
## s65    3.841481452
## s66   -0.231239928
## s67    0.542761270
## s68    0.780131779
## s69    1.096690727
## s70    0.690525998
## s71    2.301303414
## s72    2.965913580
## s73    4.393732595
## s74    2.744547070
## s75    1.035278911
## s76    1.170709479
## s77    2.796838283
## s78    2.000312540
## s79    0.007337449
## s80   -1.203916069
## s81    0.352397232
## s82    0.675108103
## s83   -3.169643942
## s84   -1.913321175
## s85   -1.647780450
## s86   -5.281261301
## s87   -5.126493027
## s88   -2.637666754
## s89   -2.342133004
## s90   -3.281910970
## s91   -4.242033198
## s92   -2.596010530
## s93   -7.821281290
## s94   -8.814741200
## s95   -8.996689798
## s96   -7.835655534
## s97   -5.749139155
## s98   -5.196182693
## s99   -8.623793296
## s100 -11.809355220
## s101 -13.129428554
## s102 -16.095143067
## s103 -15.125436350
## s104 -13.963606549
## s105 -12.953304848
## s106 -16.097179844
## s107 -15.489223470
## s108 -13.680122300
## s109 -11.921434142
## s110 -12.035411347
## s111 -12.837047727
```

```
## s112  -9.095808127
## s113  -5.433029341
## s114  -6.800835107
## s115  -8.413639598
## s116 -10.912409484
## s117 -13.553826535
## s118 -10.652543677
## s119 -12.627298331
## s120  -9.906981556
## s121 -12.668519900
## s122  -9.805502547
## s123  -7.775306633
```

```
Temperature_time_series_triple$SSE
```

```
## [1] 66244.25
```

```
#Plotted the triple exponential smoothing values to see visually how they
performed
plot(Temperature_time_series_triple)
```



Up to this point, the seasonality has been additive since there did not appear to be any change in the seasonal pattern. However, here I used a triple exponential smoothing model with multiplicative seasonality to determine whether it detected any change in seasonality. However, the SSE increased by nearly 2600 up to 68904.57, and the alpha parameter decreased from 0.66 with the additive seasonality to 0.61 and the gamma decreased from 0.62 to 0.54. This means that less weight is being placed on more recent observations which is not what we want when trying to determine if summer is ending later. As a result, the triple exponential smoothing model with additive seasonality will be chosen to perform CUSUM on.

```
#created a triple exponential smoothing model with multiplicative seasonality for comparison to the triple ES model above with additive seasonality. Parameter and error values were obtained.
Temperature_time_series_triple_multiplicative <-
HoltWinters(temperature_time_series, alpha= NULL, beta = NULL, gamma = NULL,
seasonal ="multiplicative")
Temperature_time_series_triple_multiplicative
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = temperature_time_series, alpha = NULL, beta = NULL,
gamma = NULL, seasonal = "multiplicative")
##
## Smoothing parameters:
##  alpha: 0.615003
##  beta : 0
##  gamma: 0.5495256
##
## Coefficients:
##              [,1]
## a    73.679517064
## b    -0.004362918
## s1    1.239022317
## s2    1.234344062
## s3    1.159509551
## s4    1.175247483
## s5    1.171344196
## s6    1.151038408
## s7    1.139383104
## s8    1.130484528
## s9    1.110487514
## s10   1.076242879
## s11   1.041044609
## s12   1.058139281
## s13   1.032496529
## s14   1.036257448
## s15   1.019348815
## s16   1.026754142
## s17   1.071170378
## s18   1.054819556
## s19   1.084397734
## s20   1.064605879
## s21   1.109827336
## s22   1.112670130
```

```
## s23   1.103970506
## s24   1.102771209
## s25   1.091264692
## s26   1.084518342
## s27   1.077914660
## s28   1.077696145
## s29   1.053788854
## s30   1.079454300
## s31   1.053481186
## s32   1.054023885
## s33   1.078221405
## s34   1.070145761
## s35   1.054891375
## s36   1.044587771
## s37   1.023285461
## s38   1.025836722
## s39   1.031075732
## s40   1.031419152
## s41   1.021827552
## s42   0.998177248
## s43   0.996049257
## s44   0.981570825
## s45   0.976510542
## s46   0.967977608
## s47   0.985788411
## s48   1.004748195
## s49   1.050965934
## s50   1.072515008
## s51   1.086532279
## s52   1.098357400
## s53   1.097158461
## s54   1.054827180
## s55   1.022866587
## s56   0.987259326
## s57   1.016923524
## s58   1.016604903
## s59   1.004320951
## s60   1.019102781
## s61   0.983848662
## s62   1.055888360
## s63   1.056122844
## s64   1.043478958
## s65   1.039475693
## s66   0.991019224
## s67   1.001437488
## s68   1.002221759
## s69   1.003949213
## s70   0.999566344
## s71   1.018636837
## s72   1.026490773
```

```
## s73   1.042507768
## s74   1.022500795
## s75   1.002503740
## s76   1.004560984
## s77   1.025536556
## s78   1.015357769
## s79   0.992176558
## s80   0.979377825
## s81   0.998058079
## s82   1.002553395
## s83   0.955429116
## s84   0.970970220
## s85   0.975543504
## s86   0.931515830
## s87   0.926764603
## s88   0.958565273
## s89   0.963250387
## s90   0.951644060
## s91   0.937362688
## s92   0.954257999
## s93   0.892485444
## s94   0.879537700
## s95   0.879946892
## s96   0.890633648
## s97   0.917134959
## s98   0.925991769
## s99   0.884247686
## s100  0.846648167
## s101  0.833696369
## s102  0.800001437
## s103  0.807934782
## s104  0.819343668
## s105  0.828571029
## s106  0.795608740
## s107  0.796609993
## s108  0.815503509
## s109  0.830111282
## s110  0.829086181
## s111  0.818367239
## s112  0.863958784
## s113  0.912057203
## s114  0.898308248
## s115  0.878723779
## s116  0.848971946
## s117  0.813891909
## s118  0.846821392
## s119  0.819121827
## s120  0.851036184
## s121  0.820416491
```

```
## s122  0.851581233
## s123  0.874038407
```

```
Temperature_time_series_triple_multiplicative$SSE
```

```
## [1] 68904.57
```

If I were to stop the analysis here, since there is no trend at all in the seasonality, I would say that there is no evidence to support the notion that the end of summer has gotten later in Atlanta over time.

However, to make sure this is the case, I extracted the seasonality factors from the chosen exponential smoothing model to perform CUSUM on.

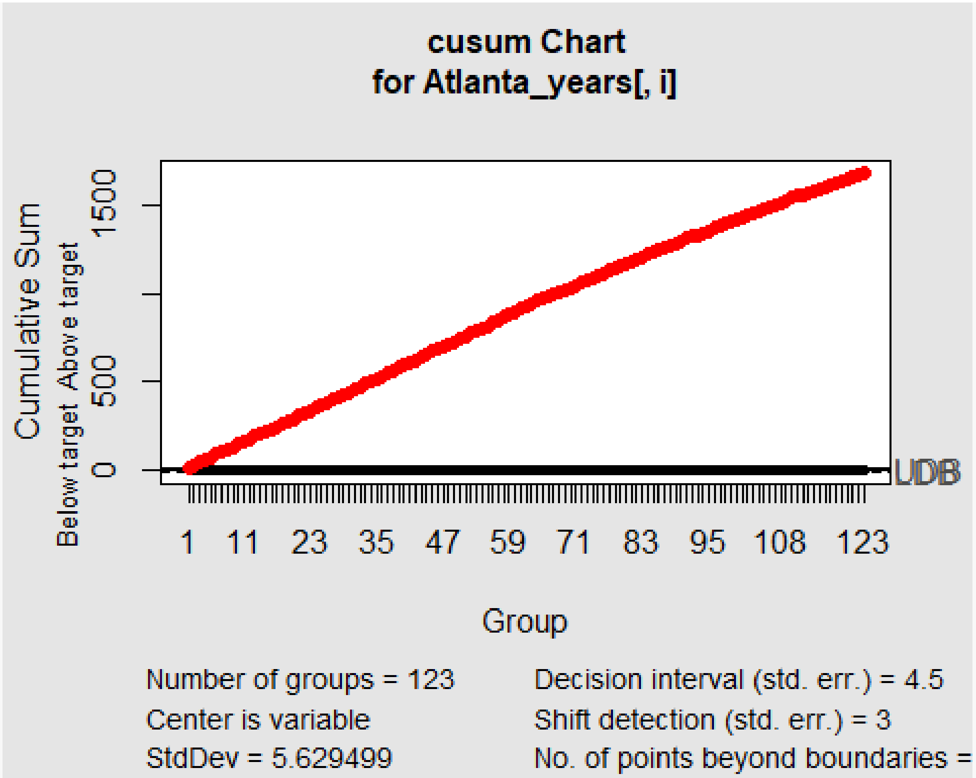
```
#obtained the fitted values for each day
fittedHW <- Temperature_time_series_triple$fitted
fittedHW[,4]
```

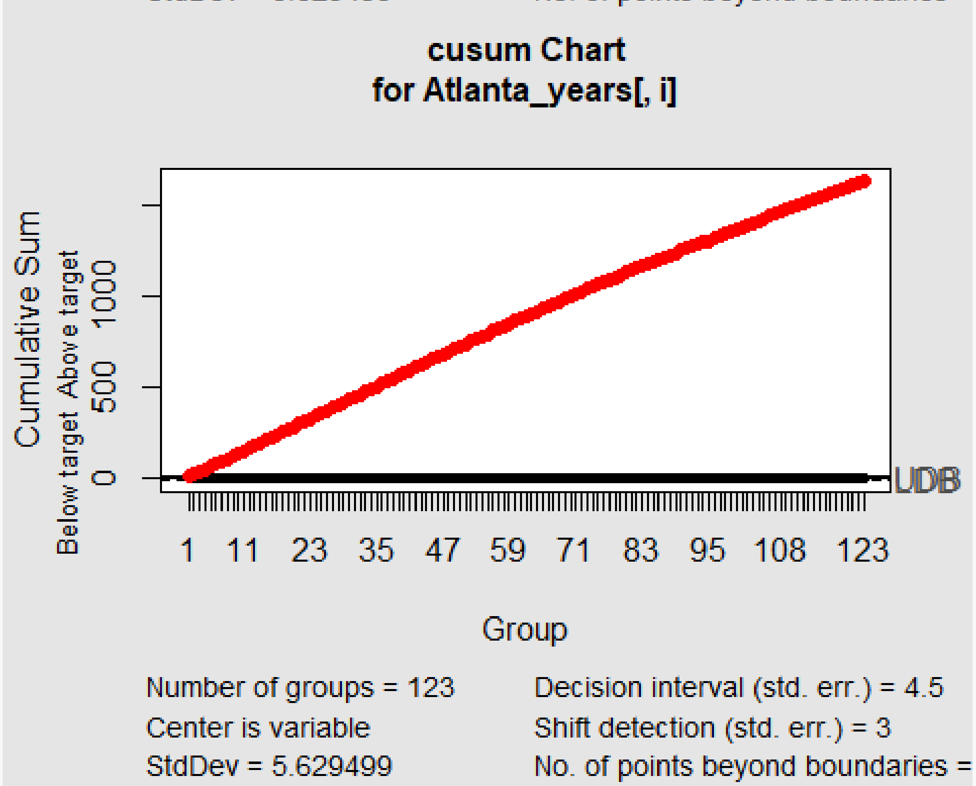
```
## Time Series:
## Start = c(1997, 1)
## End = c(2015, 123)
## Frequency = 123
##    [1]   4.303159495   8.238118845  11.091777381   9.042996893
2.067387137
##    [6]   2.116167625  -6.826921806   5.197468438   2.205598519
5.262509089
##   [11]   2.295029414   6.327549739   4.343809902   8.270639170
9.205598519
##   [16]   9.189338357   7.140557869   7.075517219   6.994216406
7.896655430
##   [21]   6.766574129   1.677143235   4.766574129   5.799094454
6.815354617
##   [26]   6.986086324   9.099907462   9.221858682   7.278769251
6.238118845
##   [31]  -9.940742944  -2.013913676   1.977956243   6.051126975
6.986086324
##   [36]   5.864135105   1.717793641   1.538931853  -2.493588472  -
9.574889285
##   [41]  -2.656190098   3.238118845   5.156818032   5.099907462
4.189338357
##   [46]   5.335679820   8.522671690   8.652752991   6.669013154
6.620232666
##   [51]   5.603972503  -0.322856765  -3.168385220  -1.119604733  -
0.030173838
##   [56]   1.953565999   4.986086324   8.132427788   8.343809902
9.490151365
##   [61]   9.620232666   2.770639170  -3.152125058 -10.095214489
3.937305836
##   [66]   0.929175755   3.969826162   6.042996893   6.189338357
6.221858682
##   [71]   8.295029414   1.343809902   3.392590389   5.416980633  -
```

```
4.526108798
##   [76]  -3.501718554   3.425110715  -0.574889285  -0.517978716  -
4.469198229
##   [81]  -3.412287659  -3.339116928  -4.322856765  -1.233425871
1.799094454
##   [86]   1.839744861   4.799094454   1.758444048  -3.314726684  -
7.412287659
##   [91] -10.404157578 -18.379767334 -16.257816115 -10.209035627
1.839744861
##   [96] -12.192775464 -16.176515302 -18.135864895 -22.103344570  -
4.046434001

#extracted the seasonality factors and converted them into a matrix format to
make it easier to perform CUSUM
HW_seasonality_factors <- matrix(fittedHW[,4], ncol = 123)
view(HW_seasonality_factors)
```

I computed the mean for each day between July and October with the 20 years of data present. Then, the standard deviation was calculated in order to then run CUSUM. The CUSUM output is shown below.

```
#computed the mean of each day's temperature between July and October
daily_mean <- colMeans(HW_seasonality_factors)
view(daily_mean)

#computed the standard deviation of the means
daily_standard_deviation <- sd(daily_mean)


Atlanta_years <- Temperature[, 2:21]
#create two vectors, one for years and another to store low temperature days
that fall below a
#threshold indicated in the cusum function
CUMSUM_years <- vector("list", ncol(Atlanta_years))
CUMSUMlow_temps <- vector("list", ncol(Atlanta_years))

#looped over each year in order for a cusum to be run for each year in the
data set
for (i in 1:ncol(Atlanta_years)) {
  #ran a cusum with a threshold of 4.5, right between the standard thresholds
of 4 and 5 which
  #seem to be common, and a shift of 3
 CUMSUM_years[[i]] <- cusum(Atlanta_years[, i], center = daily_mean, std.dev
= daily_standard_deviation,
decision.interval = 4.5, se.shift = 3, plot = TRUE)
#obtain low temperatures that fell below the threshold of 4.5
CUMSUMlow_temps[[i]] <- CUMSUM_years[[i]]$violations
```





Based on the output above, there is no conclusive evidence that the end of summer has gotten later over the last 20 years in Atlanta. There are occasional fluctuations that are higher than average, but not enough to indicate any trend over time.

As a result, the end of summer has appeared to be fairly consistent over the last 20 years. It would be cool to see if there was any change using data for each year since 2015. However, without more data, we cannot say that the end of summer is getting later.