

HW4 - Kelly “Scott” Sims

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?

With home automation at all time highs, everyone is turning to automated central air to keep their house at the desired climate while being energy efficiency. Take a nest for example. You program the desired temperatures at certain times of the day. You might have your air turned off during the early morning hours when the sun is just rising and it is still relatively cool. But by Mid to late afternoon, you might have it programmed to turn on and keep your house cool. The nest thermostat basically monitors the ambient temperature, and when it sufficiently rises above or drops below the desired temperature, it either shuts off or turns on, depending on the situation. Let's say for instance you have it set for 70 degrees Fahrenheit as the desired temperature, if someone is constantly running in or out of the house, has a window open, a ceiling fan, etc, the temperature might fluctuate alot around that 70 degree mark. If the nest turned on at 70.5 or above, and turned off at 69.5 or less, then your central air would constantly be turning off and on and wouldn't be very effecient. Therefore, an exponetial smoothing transformation of the ambient temperature would be appropriate here to prevent the air from turning on just because someone opened the door and entered the house. I'd imagine the alpha value would be closer 0, meaning the previous temperature has more weight over the current temperature. This is because, in those instances that someone opens the door or any of the other aforementioned stimuli, it would cause a momentary spike in readings, and then re-stabilize. By giving more weight to past readings that are more consistent, this would properly smooth the temperature readings to those that were more consistent with one another.

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).

While trying to exponentially smooth the data, we will need to first get an idea of an optimal alpha and beta parameter for the HoltWinters function. Let's plot the data for 1996 and experiment with every combination of alpha from 0.2 to 0.5 and beta from 0.0 to 0.5

```
library(TTR)
temps <- read.table('temps.txt', header = TRUE, stringsAsFactors = FALSE)
#temps$DAY <- format(as.Date(temps$DAY, format = '%d-%b'), '%d-%b')
days <- temps$DAY
temps <- t(temps[,2:ncol(temps)])
colnames(temps) <- days
rownames(temps) <- 1996:2015
temps_ts <- ts(temps, start = 1996, frequency=1)
```

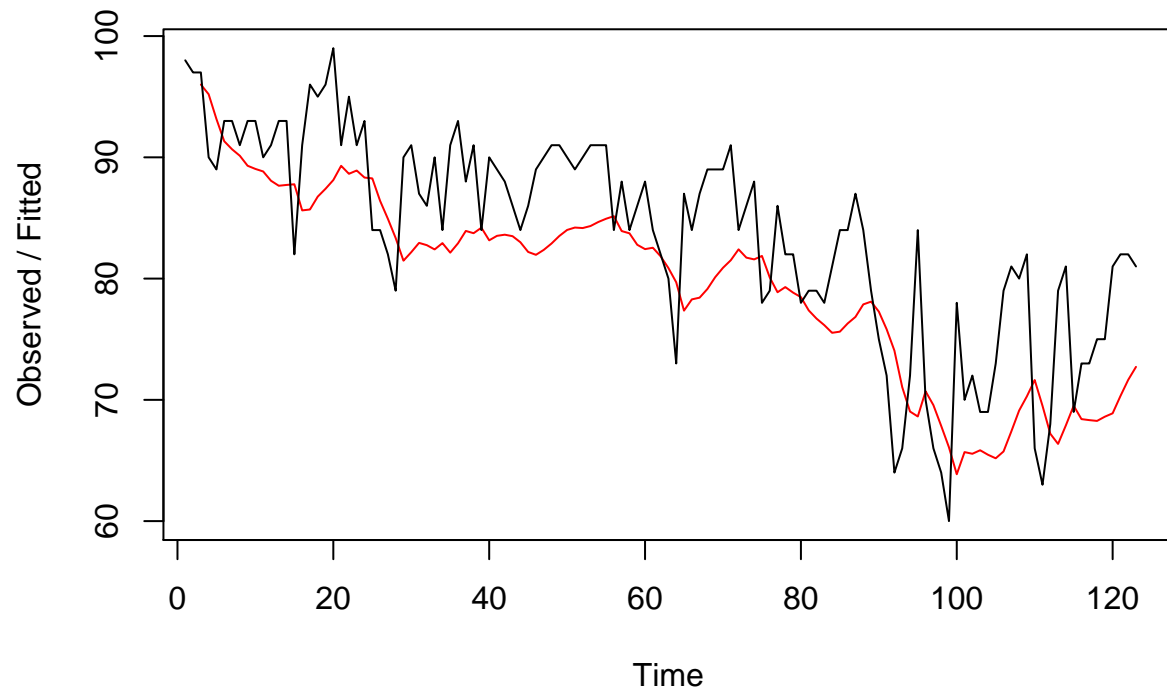
```

for(i in seq(from = 0.2, to = 0.5, by = 0.1)){
  for(j in seq(from = 0.00, to = 0.5, by = 0.1)){
    cat('Alpha: ',i, 'Beta: ', j,'\n')
    temp_h <- HoltWinters(temps_ts[1,],alpha=i, beta = j, gamma = FALSE)
    plot(temp_h)
  }
}

```

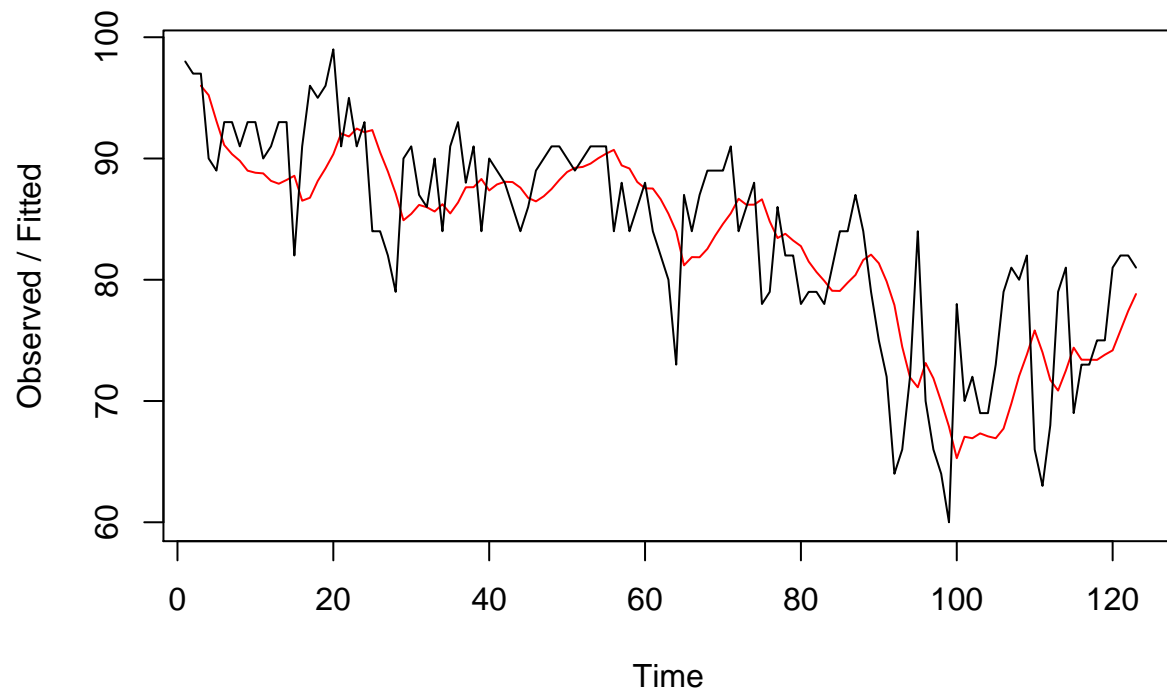
Alpha: 0.2 Beta: 0

Holt-Winters filtering



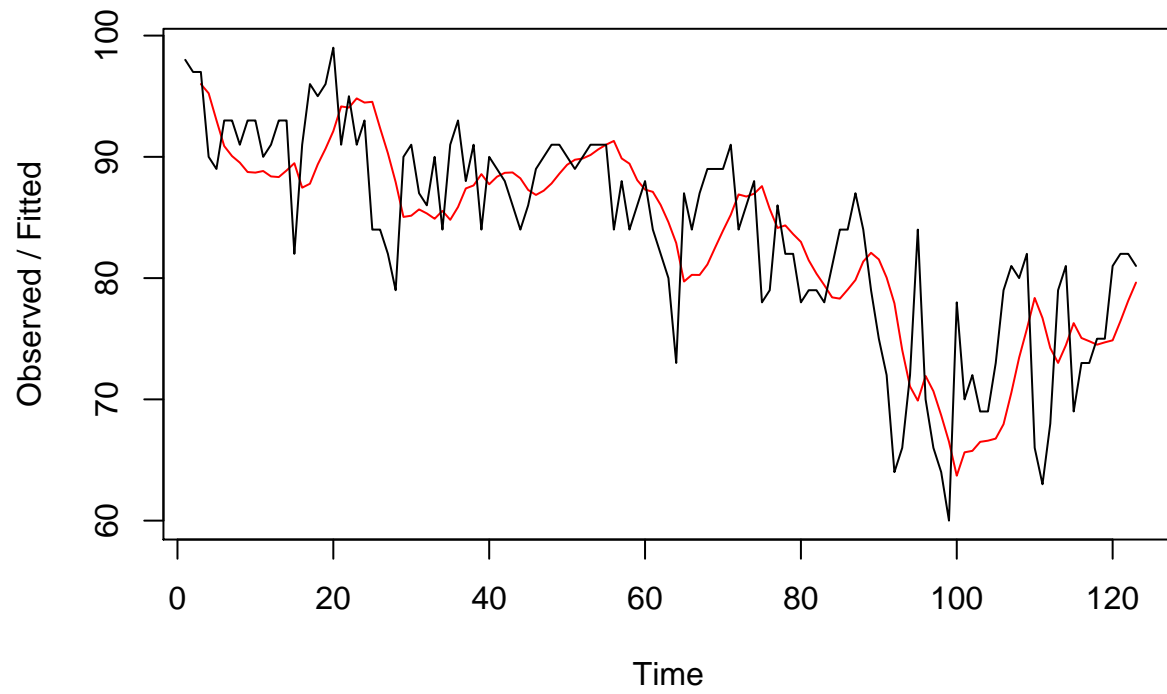
Alpha: 0.2 Beta: 0.1

Holt-Winters filtering



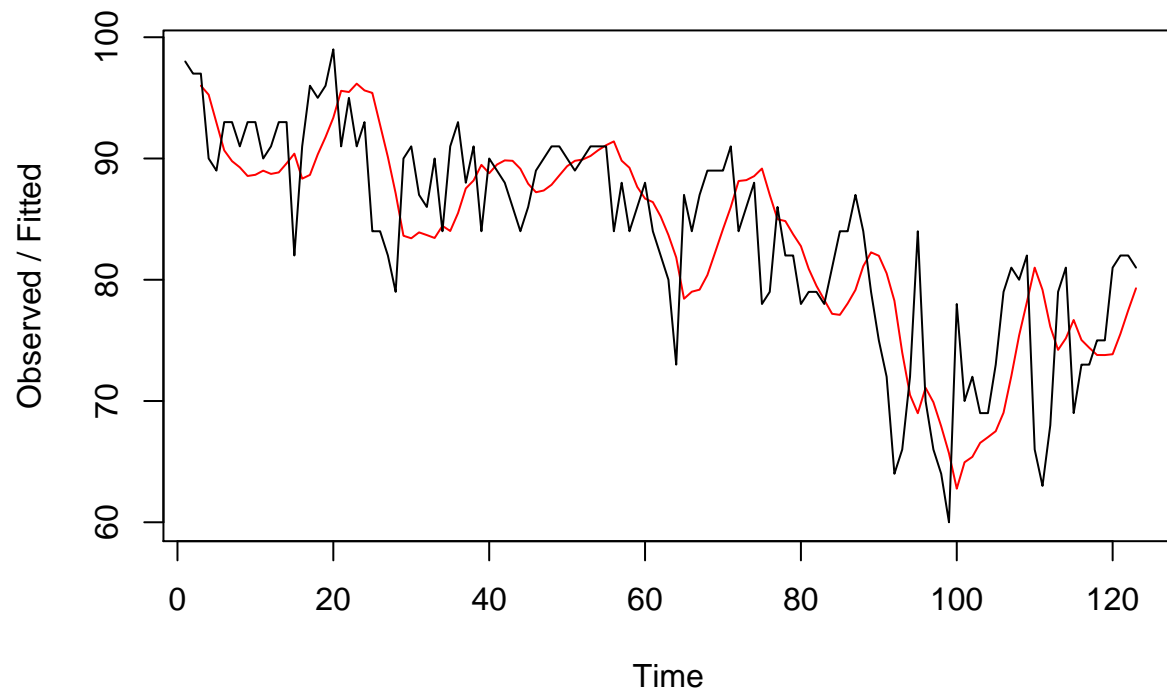
Alpha: 0.2 Beta: 0.2

Holt-Winters filtering



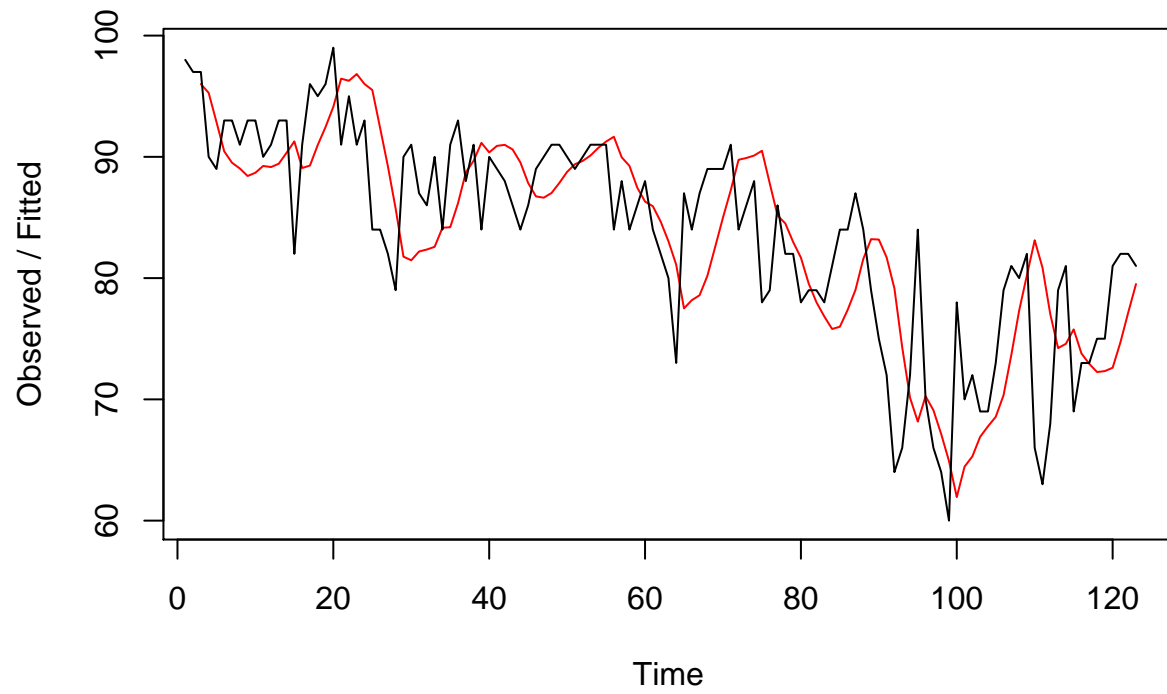
Alpha: 0.2 Beta: 0.3

Holt-Winters filtering



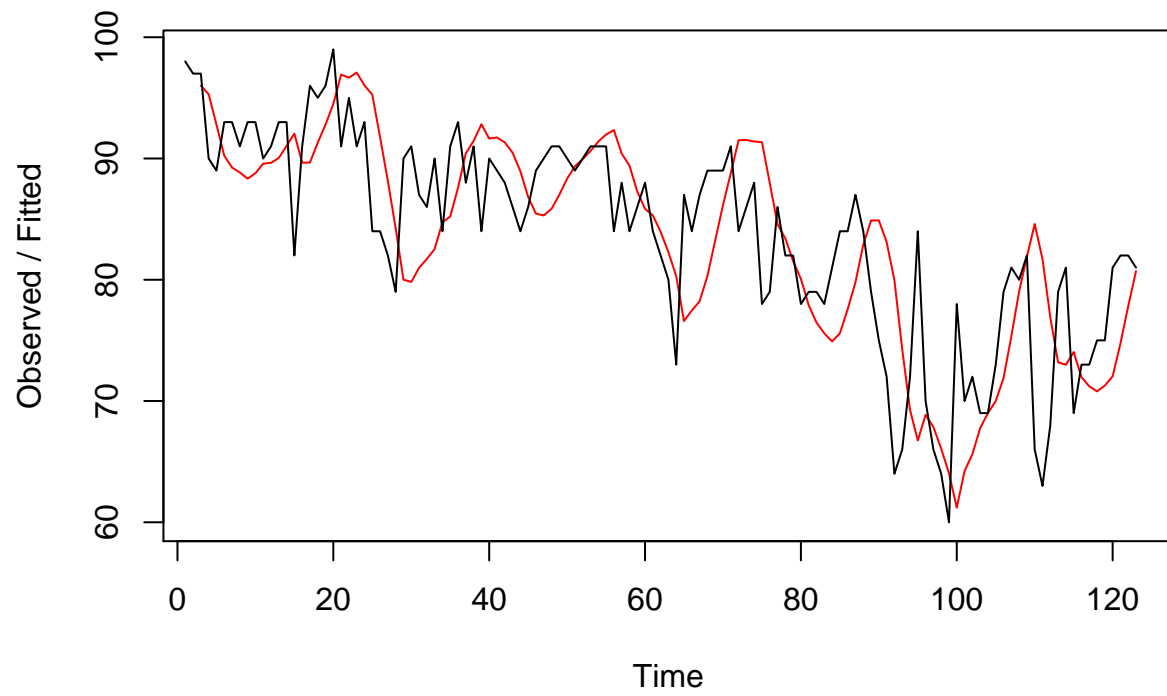
Alpha: 0.2 Beta: 0.4

Holt-Winters filtering



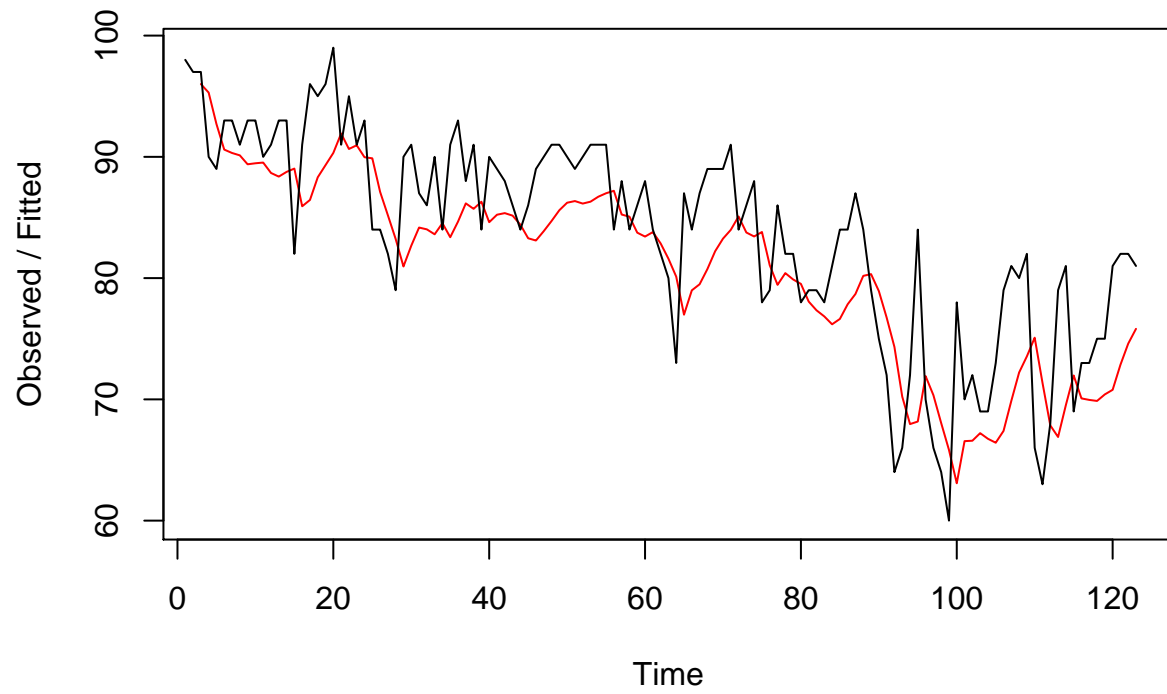
Alpha: 0.2 Beta: 0.5

Holt-Winters filtering



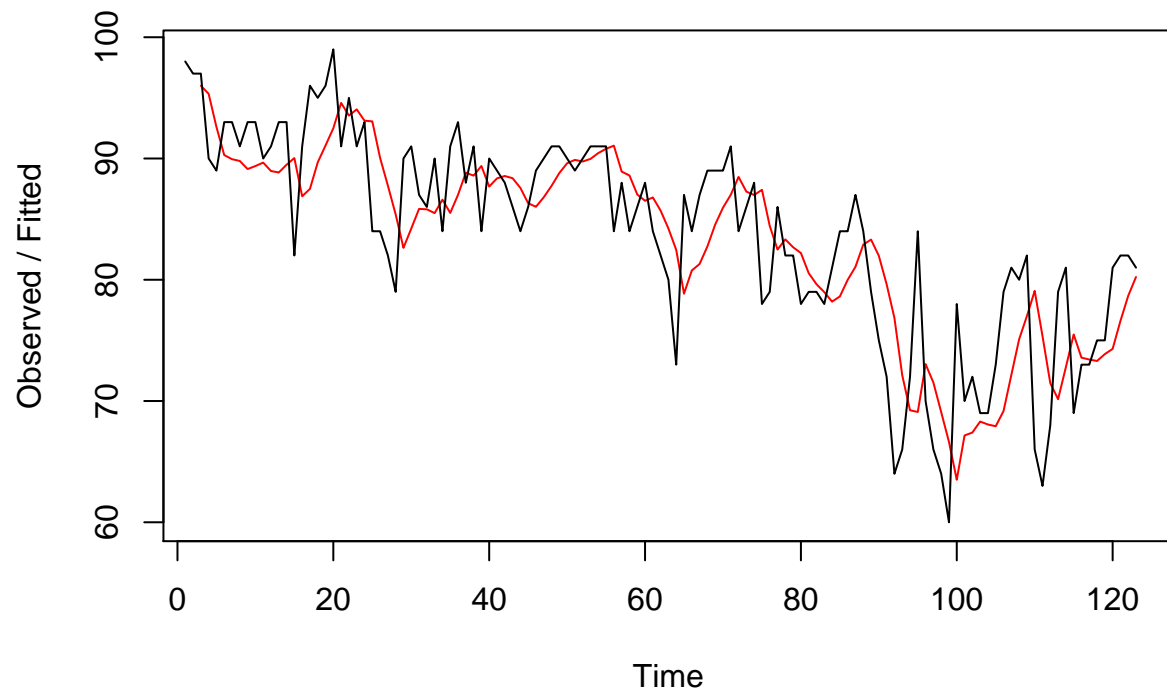
Alpha: 0.3 Beta: 0

Holt-Winters filtering



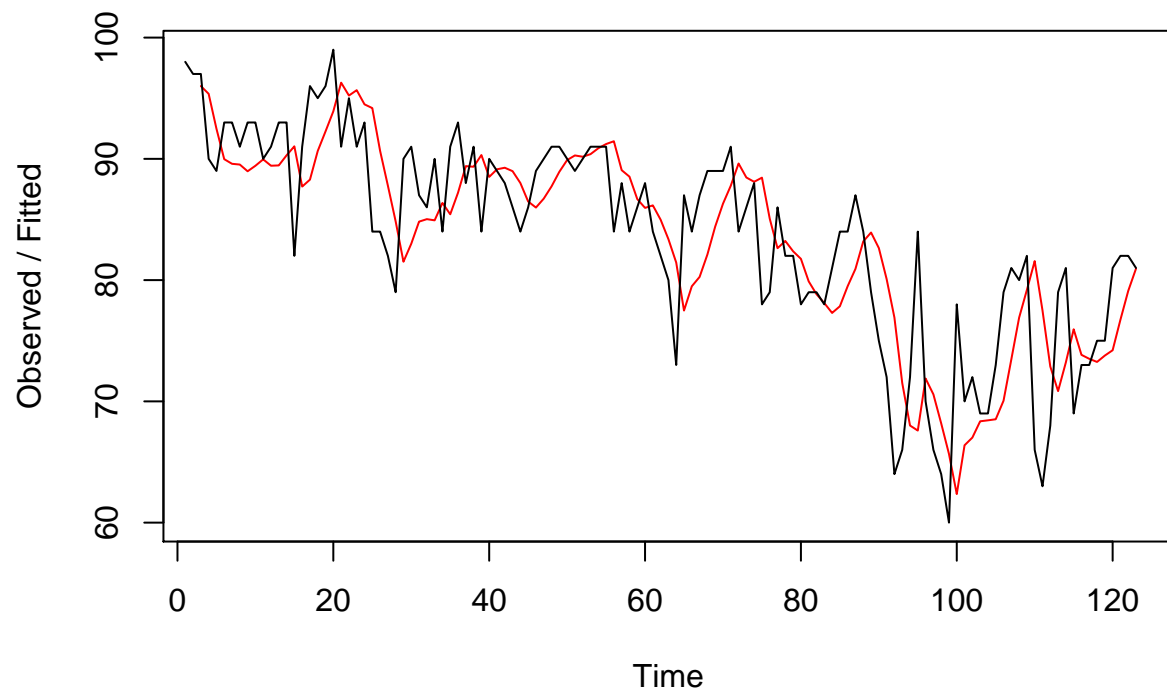
Alpha: 0.3 Beta: 0.1

Holt-Winters filtering



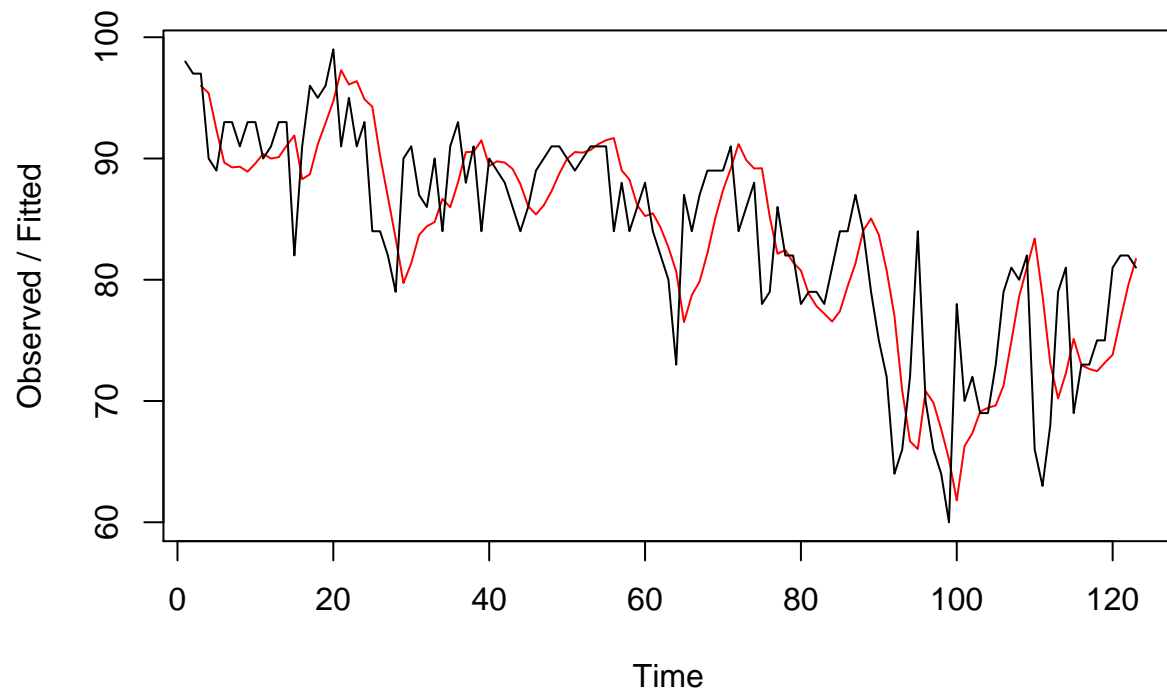
Alpha: 0.3 Beta: 0.2

Holt-Winters filtering



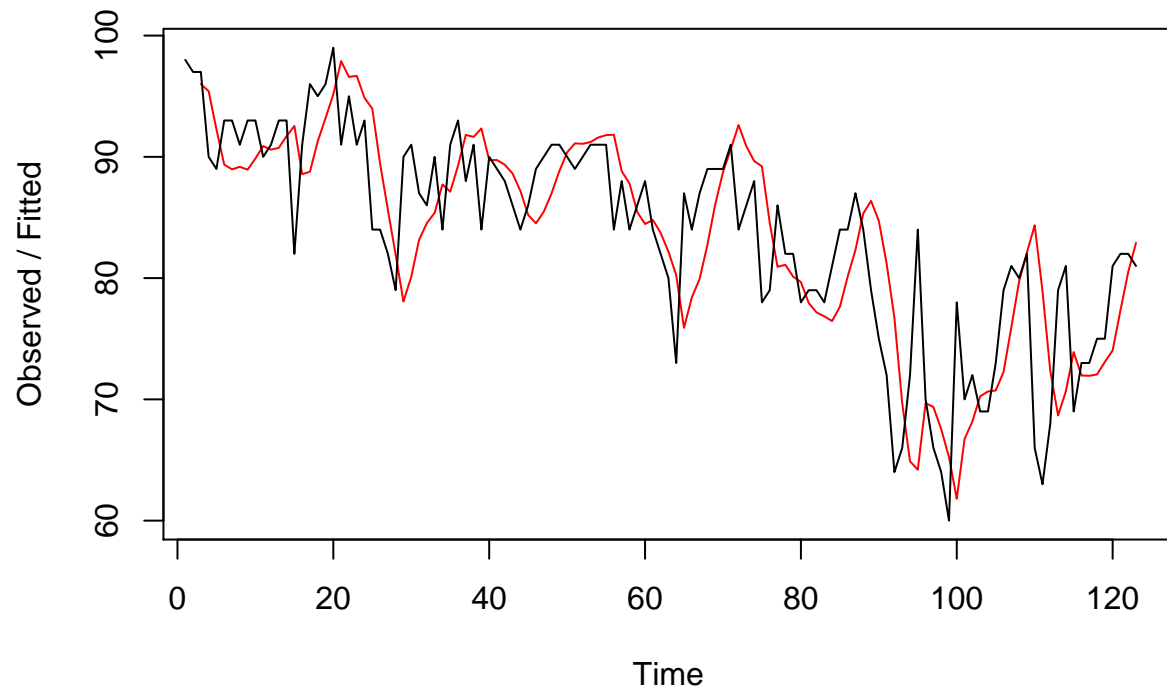
Alpha: 0.3 Beta: 0.3

Holt-Winters filtering



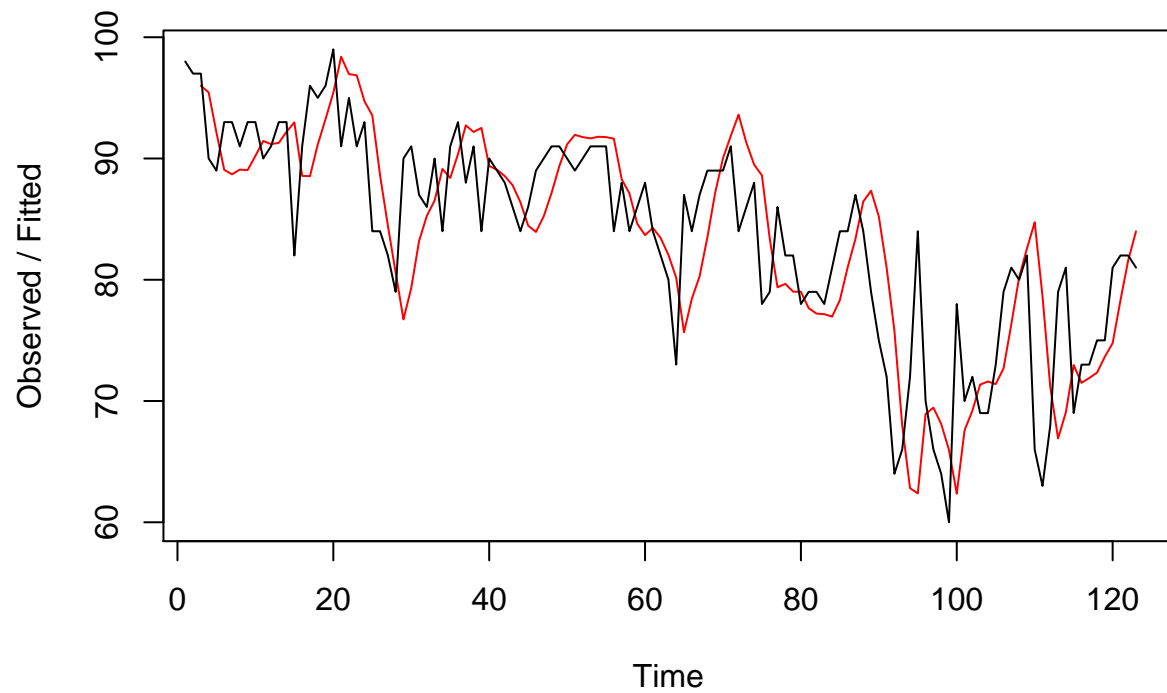
Alpha: 0.3 Beta: 0.4

Holt-Winters filtering



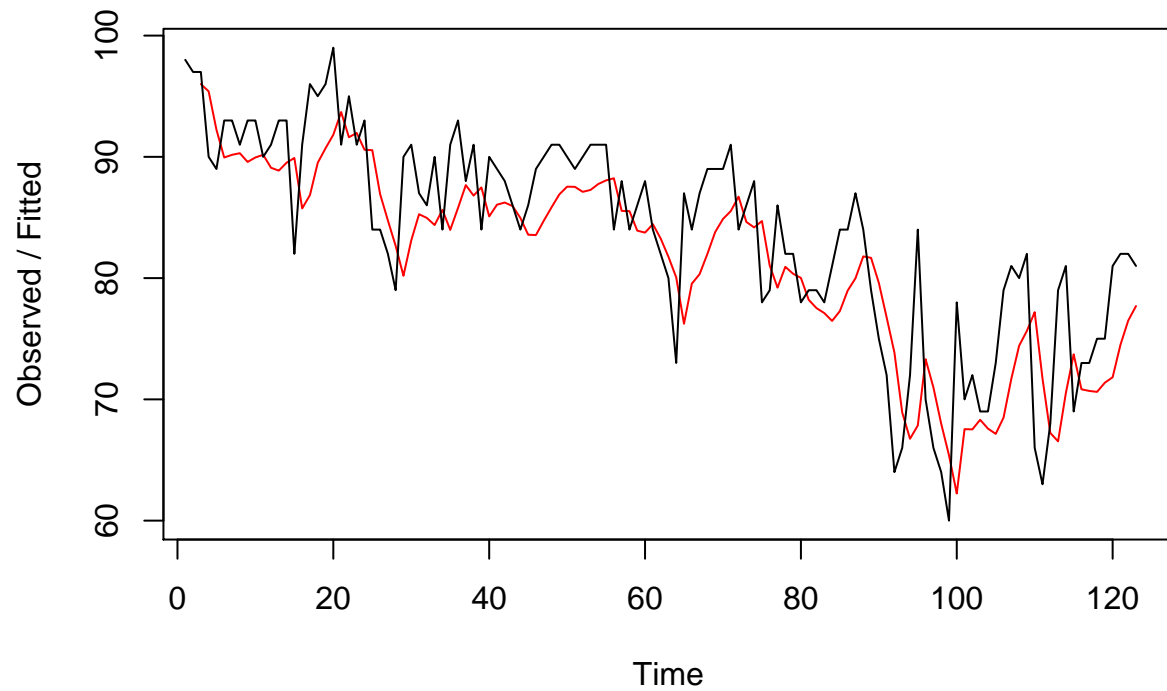
Alpha: 0.3 Beta: 0.5

Holt-Winters filtering



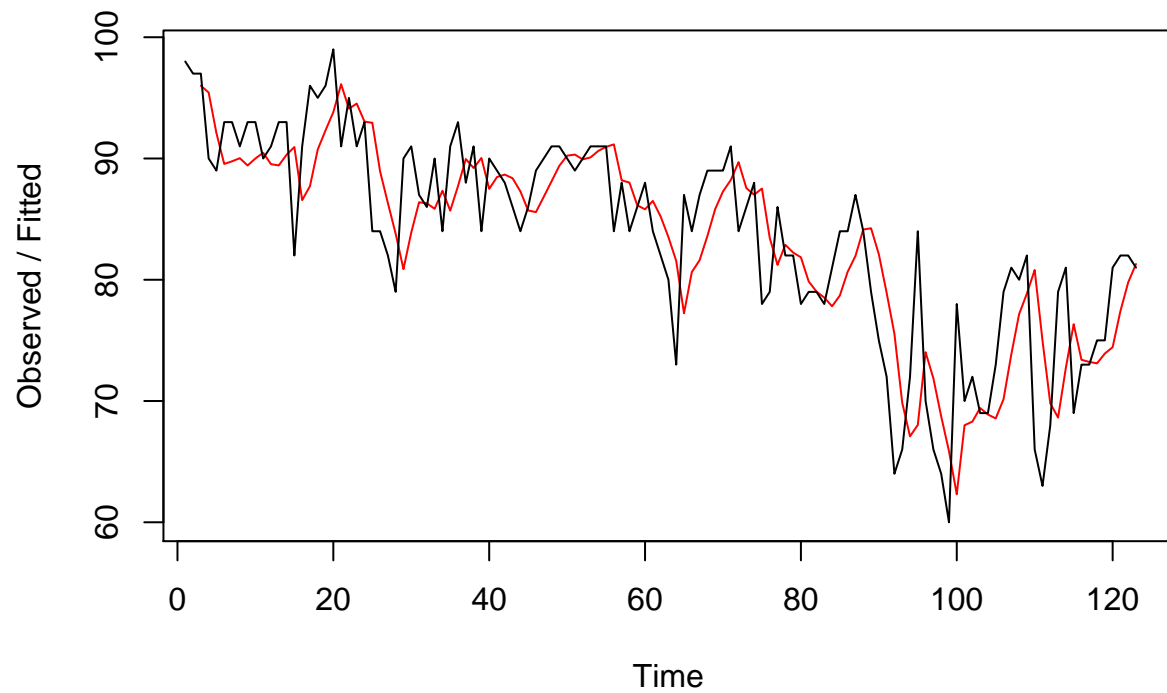
Alpha: 0.4 Beta: 0

Holt-Winters filtering



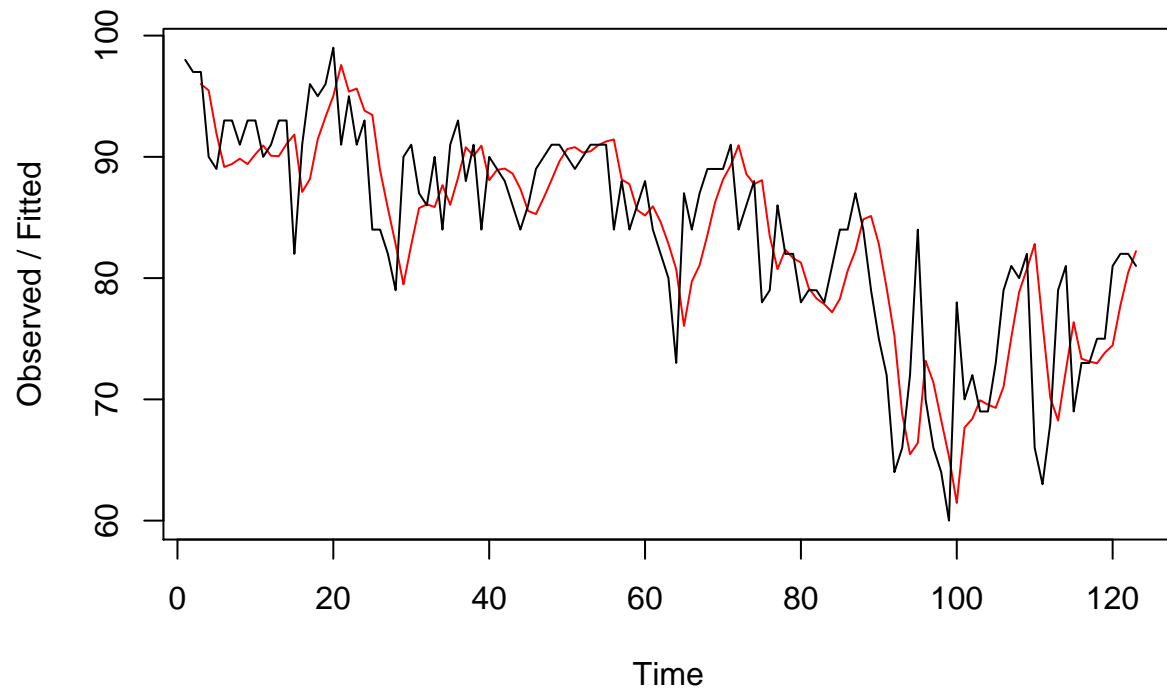
Alpha: 0.4 Beta: 0.1

Holt-Winters filtering



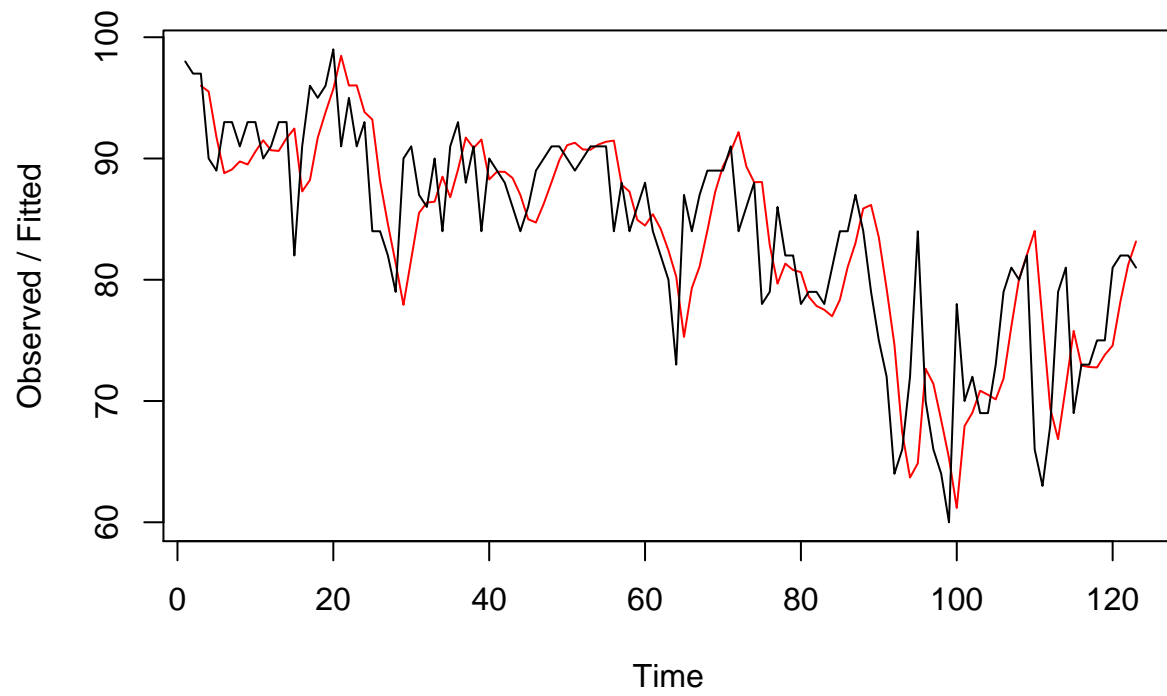
Alpha: 0.4 Beta: 0.2

Holt-Winters filtering



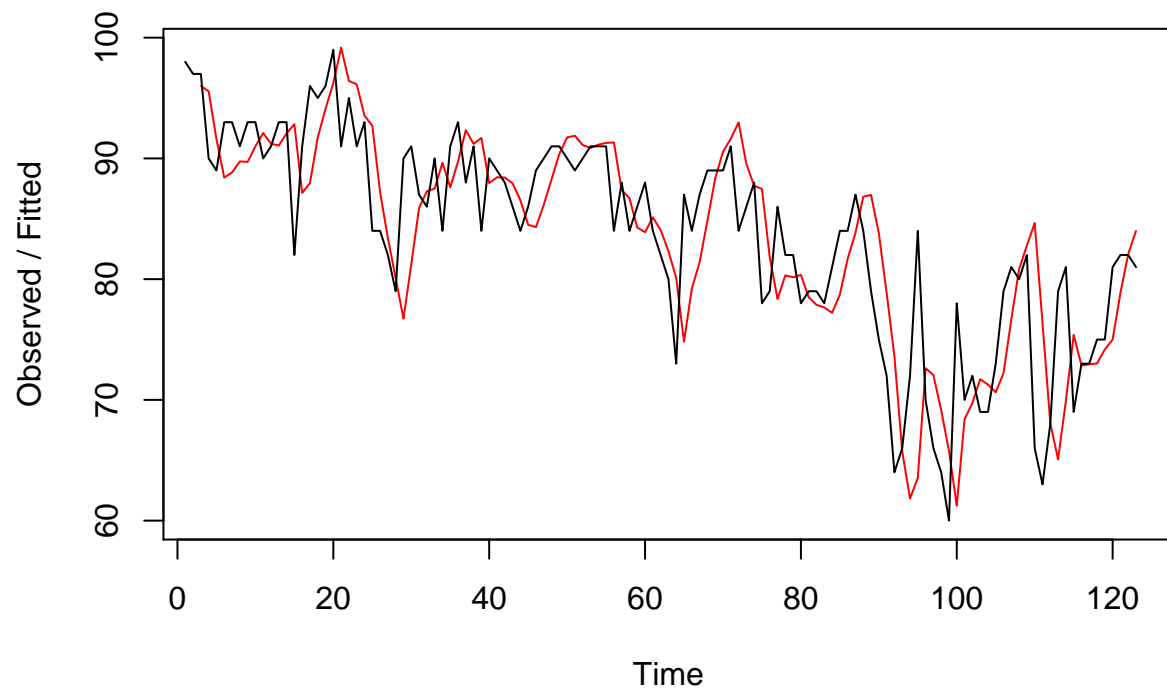
Alpha: 0.4 Beta: 0.3

Holt-Winters filtering



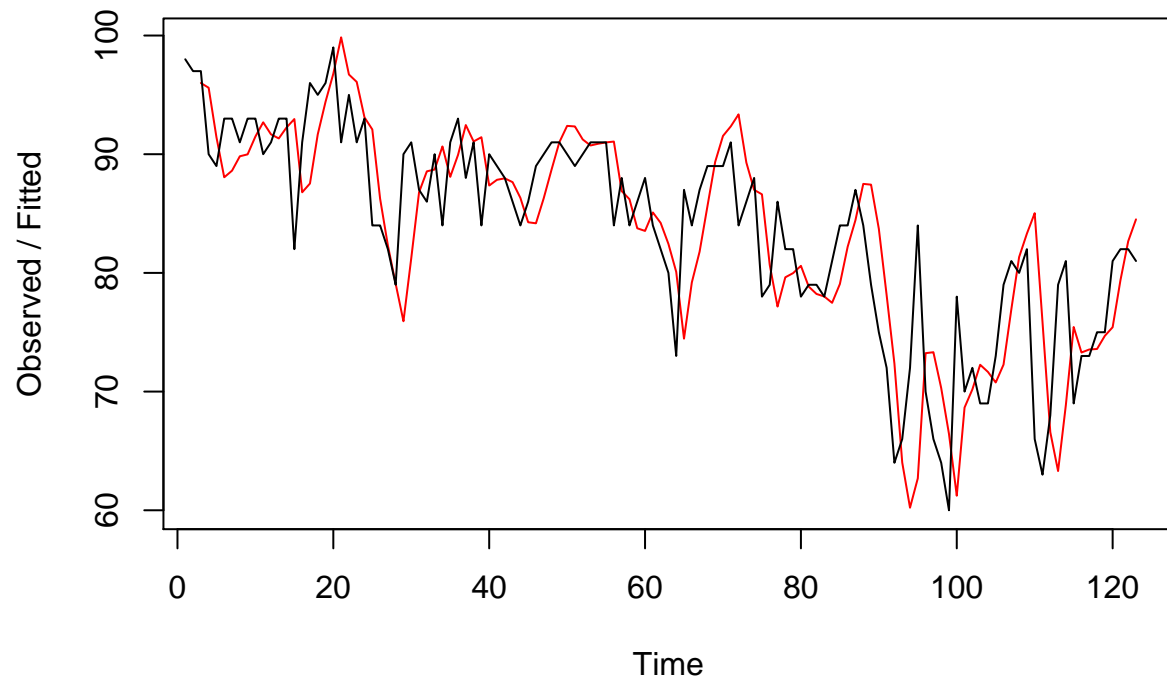
Alpha: 0.4 Beta: 0.4

Holt-Winters filtering



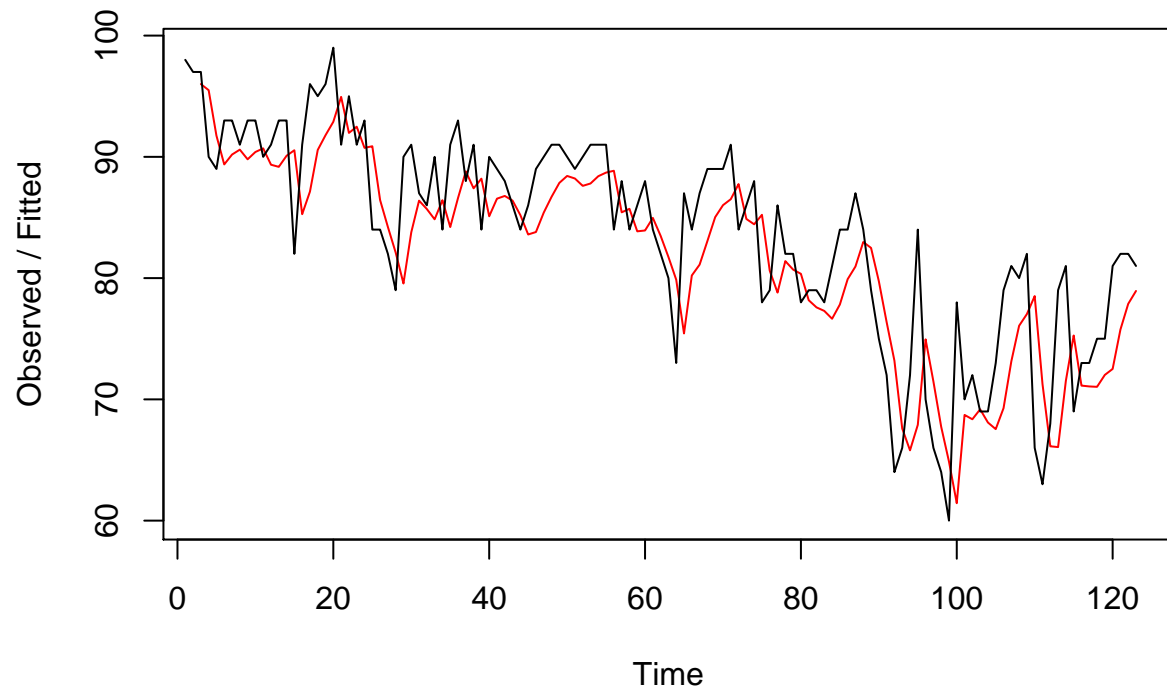
Alpha: 0.4 Beta: 0.5

Holt-Winters filtering



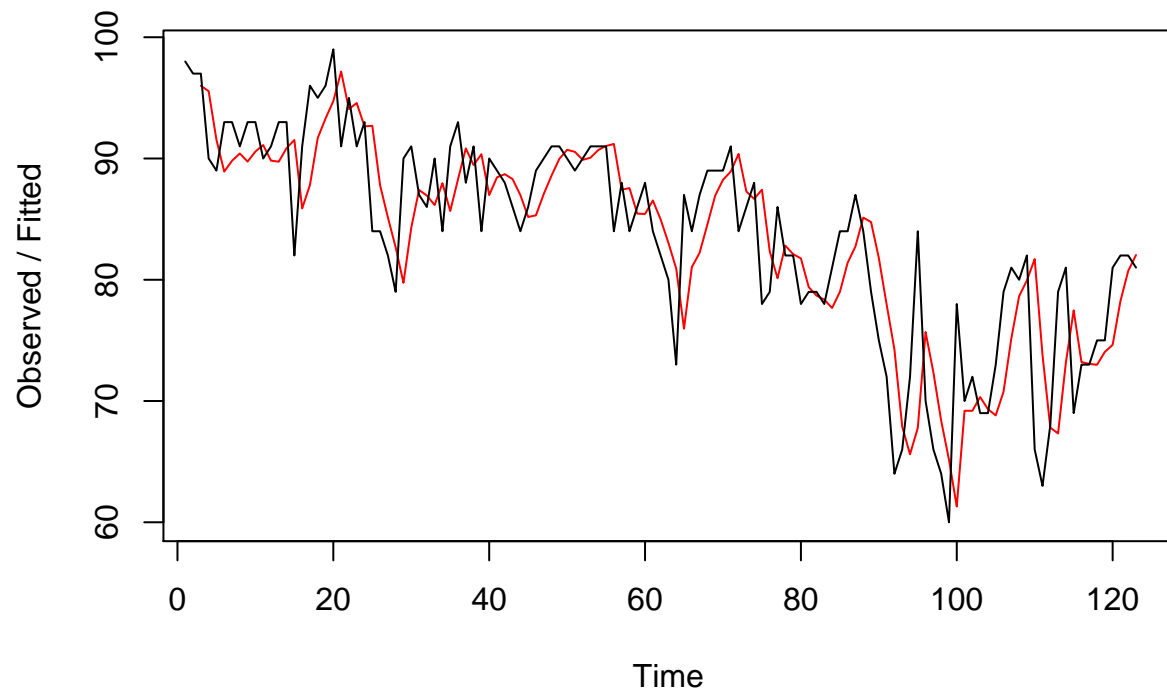
Alpha: 0.5 Beta: 0

Holt-Winters filtering



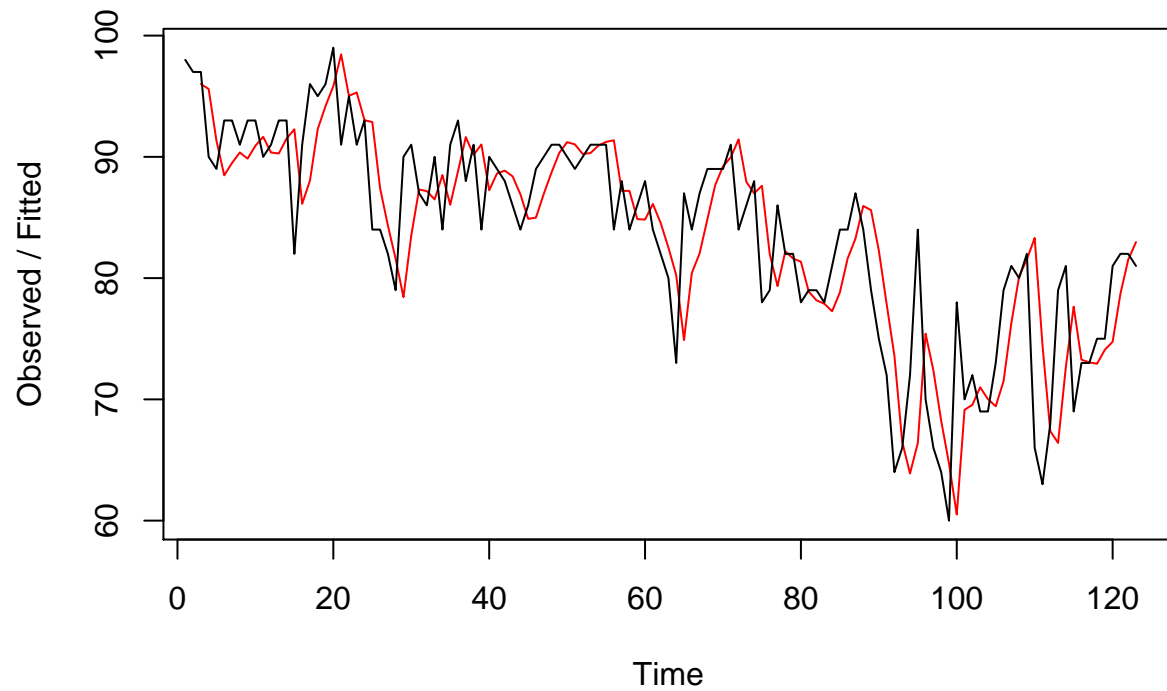
Alpha: 0.5 Beta: 0.1

Holt-Winters filtering



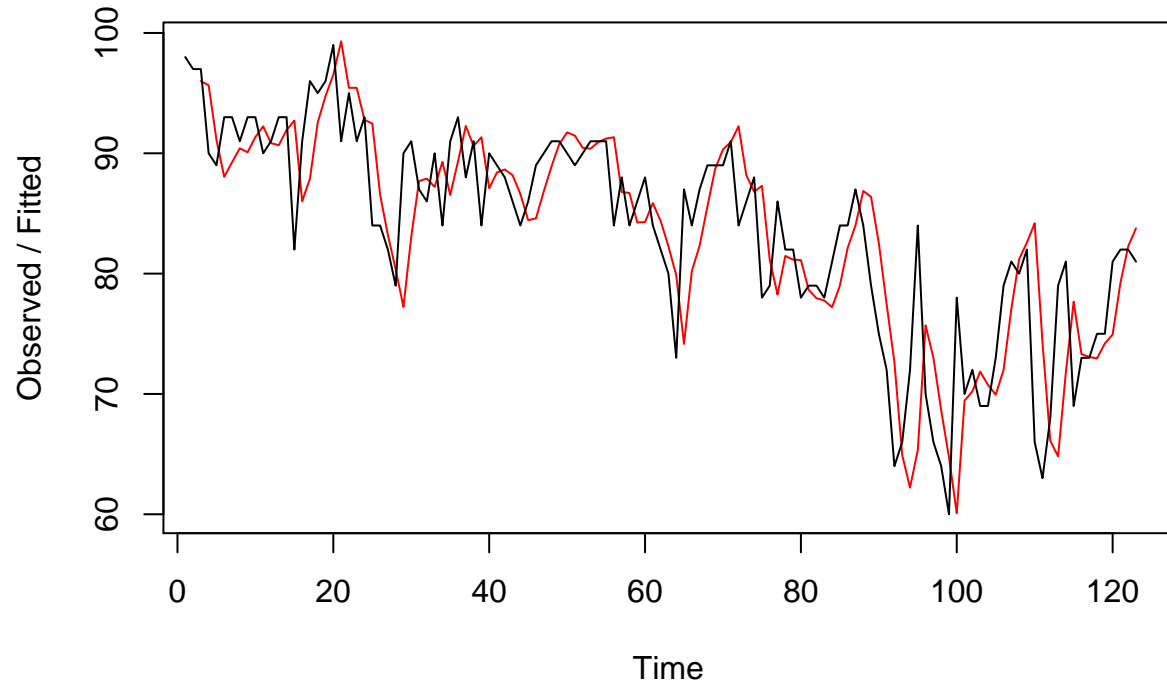
Alpha: 0.5 Beta: 0.2

Holt-Winters filtering



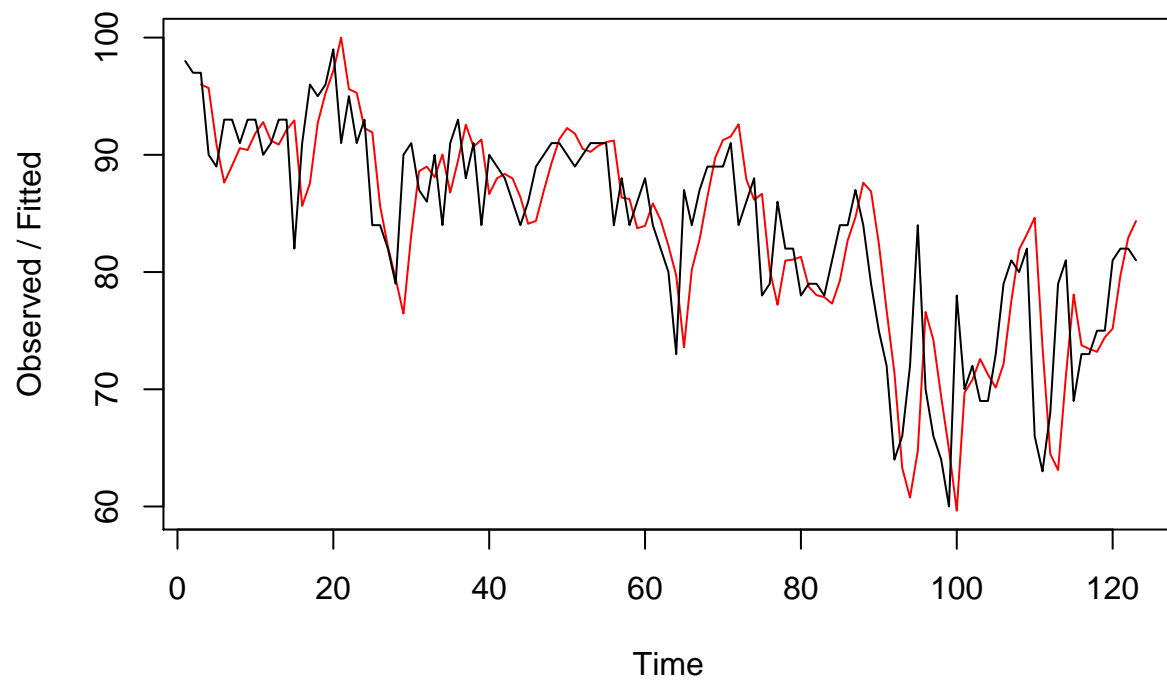
Alpha: 0.5 Beta: 0.3

Holt-Winters filtering



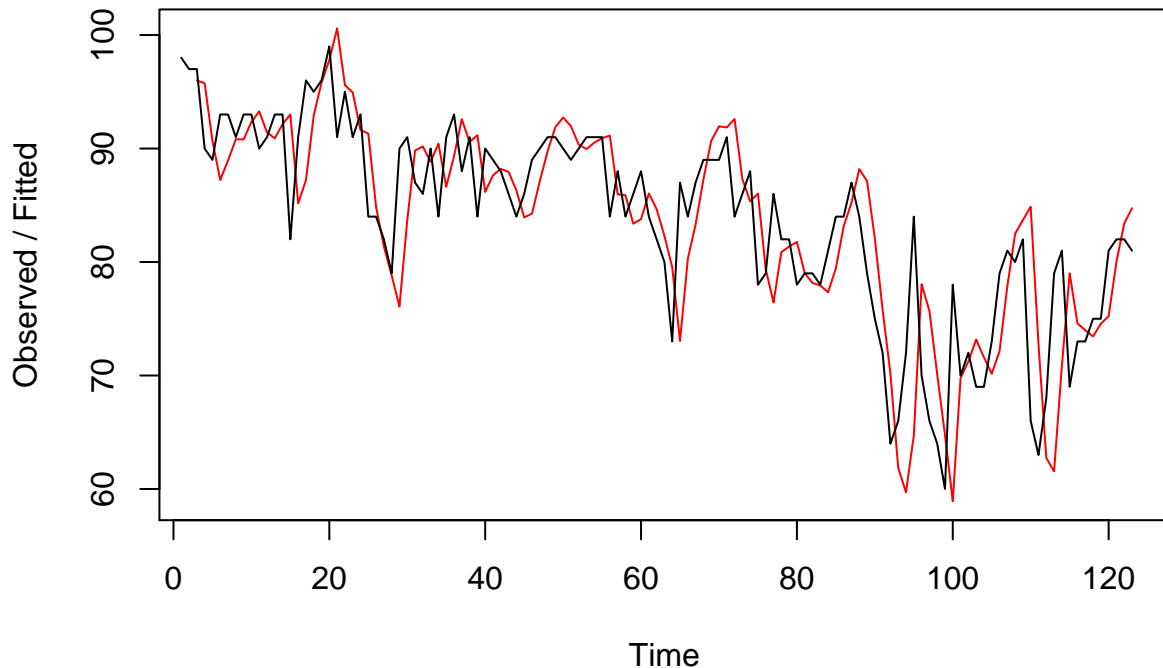
Alpha: 0.5 Beta: 0.4

Holt-Winters filtering



Alpha: 0.5 Beta: 0.5

Holt-Winters filtering



An alpha value of 0.2 and beta of 0.2 seems to best smooth the data without losing any major characteristic changes. It also doesn't appear to amplify any characteristic changes. Let's Attempt a CUMSUM algorithm on the 1996 data with a 0.2 alpha and beta.

```
temp_h <- HoltWinters(temps_ts[1,],alpha=0.2, beta = 0.2, gamma = FALSE)
forecast <- temp_h$fitted[,1]
```

```
avg = mean(forecast)
s = forecast[1]
C = 1
threshold = -30
for(i in seq(from = 2, to = length(forecast))){
  s = s + (forecast[i] - avg + C)

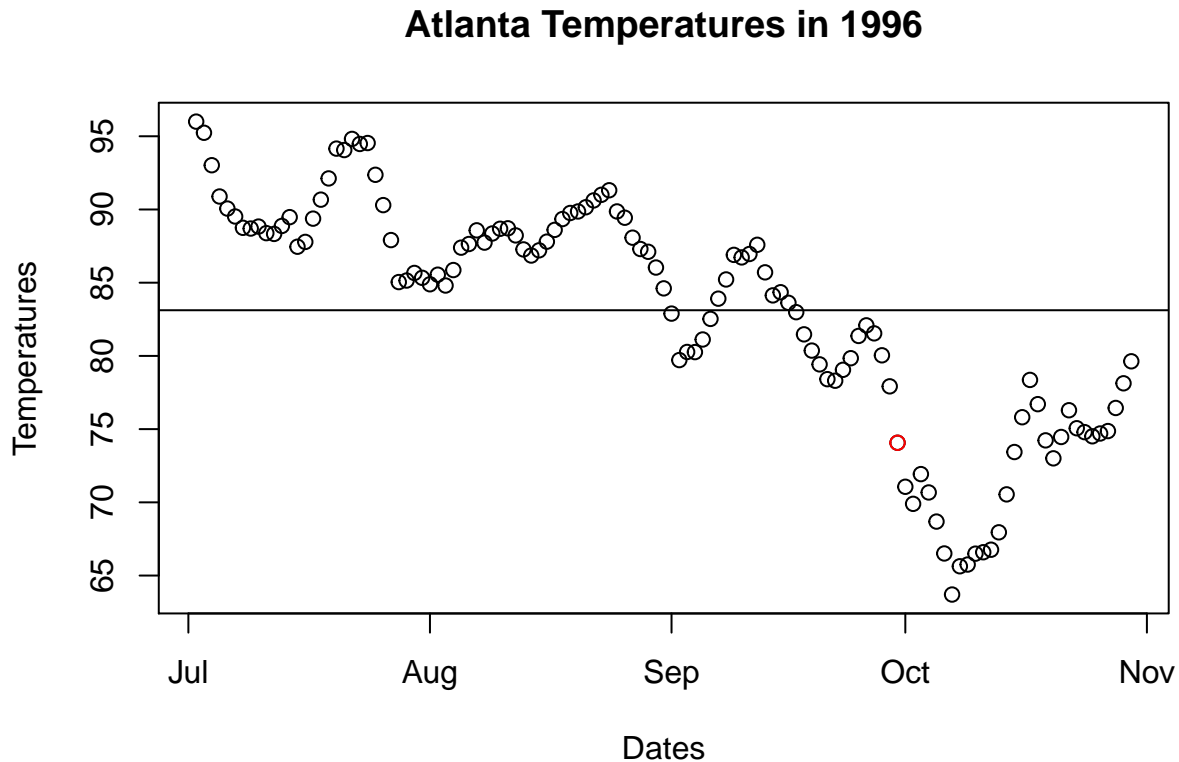
  s = min(0, s)

  if (s < threshold){
    day_change = days[i]
    day_index = i
    break
  }
}
print(day_change)
```

```
## [1] "29-Sep"
```

```
xp = as.Date(strptime(days, format = '%d-%b'))
max = length(xp) - 1
xp = xp[2:max]
```

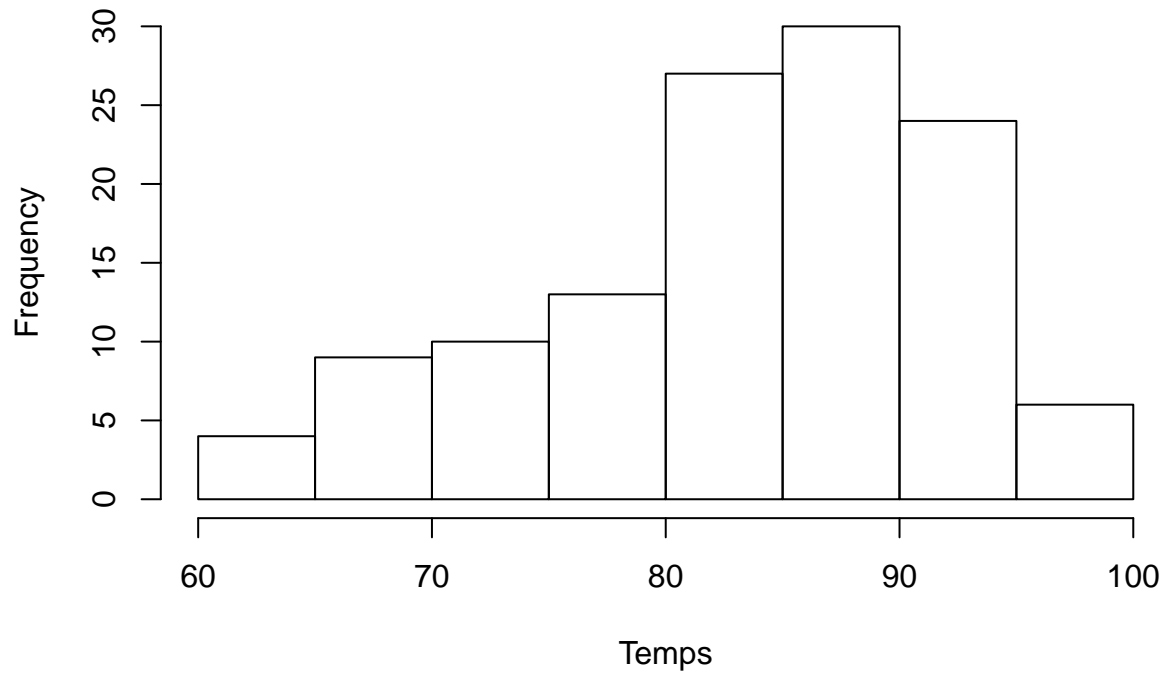
```
plot(x = xp, y = forecast, xlab = 'Dates', ylab='Temperatures', main='Atlanta Temperatures in 1996')
points(x = xp[day_index], y= forecast[day_index], col = 'red')
abline(h = avg)
```



The smoothing appears to have helped characterize the event change of summer to fall at Sept 29th. Before that date, the temperature did spike up to just below the average temperature, but never quite made it past. It then fell back down to which it never returned higher than the average temperature. As a quality check, let's see if the average temperature is the right value to use. If the data has a lot of outliers or is greatly skewed left or right, then this will erroneously pull the mean away from the true center of that data. If that is the case, then the median temperature of the data is the more appropriate metric to use

```
hist(temps_ts[1,], main = '1996 Atlanta Temperatures', xlab = 'Temps')
```

1996 Atlanta Temperatures



The data is greatly left skewed. Let's rerun the analysis again, but using the median instead of the mean in the CUSUM algorithm

```
temp_h <- HoltWinters(temps_ts[1,],alpha=0.2, beta = 0.2, gamma = FALSE)
forecast <- temp_h$fitted[,1]
```

```
median = median(forecast)
s = forecast[1]
C = 1
threshold = -30
for(i in seq(from = 2, to = length(forecast))){
  s = s + (forecast[i] - median + C)

  s = min(0, s)

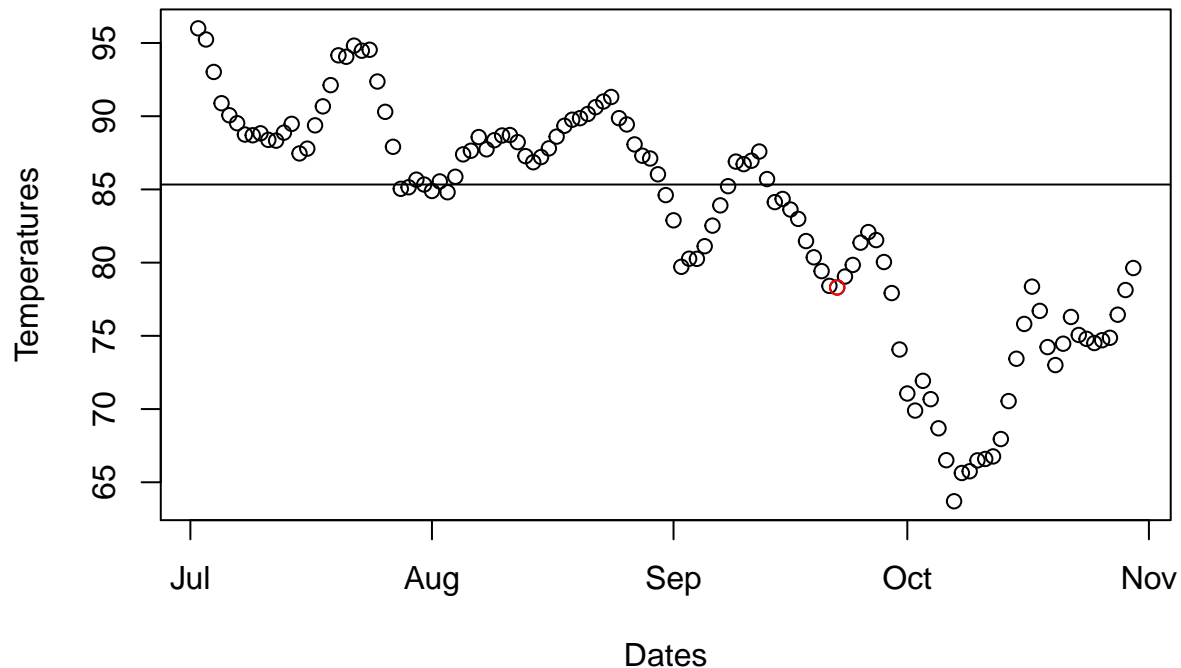
  if (s < threshold){
    day_change = days[i]
    day_index = i
    break
  }
}

print(day_change)
```

```
## [1] "21-Sep"
```

```
plot(x = xp, y = forecast, xlab = 'Dates', ylab='Temperatures', main='Atlanta Temperatures in 1996 Using
points(x = xp[day_index], y= forecast[day_index], col = 'red')
abline(h = median)
```


Atlanta Temperatures in 1996 Using Median Temperature



One can argue that the median was a better metric to use. The CUMSUM found the appropriate point at which the event change dropped below the data's median value sooner than the mean metric did. Although the temperature slightly increased from that point, it never crossed the median line again for the rest of the year. Let's use this analysis to guide the rest of our investigation on the remaining years of data.

```
weather_change_index = rep(0, 20)

for (i in seq(1,20)){

  temp_h <- HoltWinters(temps_ts[i,],alpha=0.2, beta = 0.2, gamma = FALSE)
  x <- temp_h$fitted[,1]
  median = median(x)
  C = 1
  threshold = -30
  s = x[1] - median + C

  for(j in seq(from = 2, to = length(x))){
    s = s + (x[j] - median + C)
    s = min(0, s)

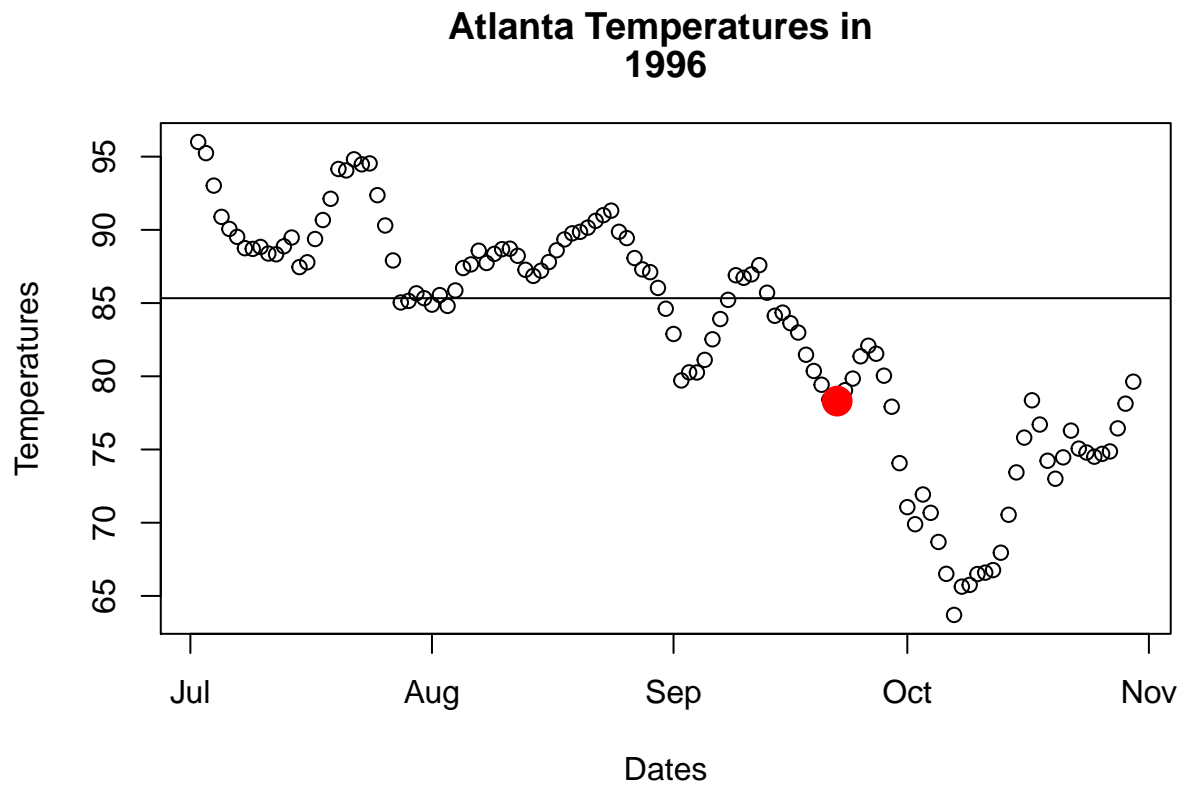
    if (s < threshold){
      day_change = days[j]
      day_index = j
      weather_change_index[i] = day_index
      break
    }
  }
}
```

```

}

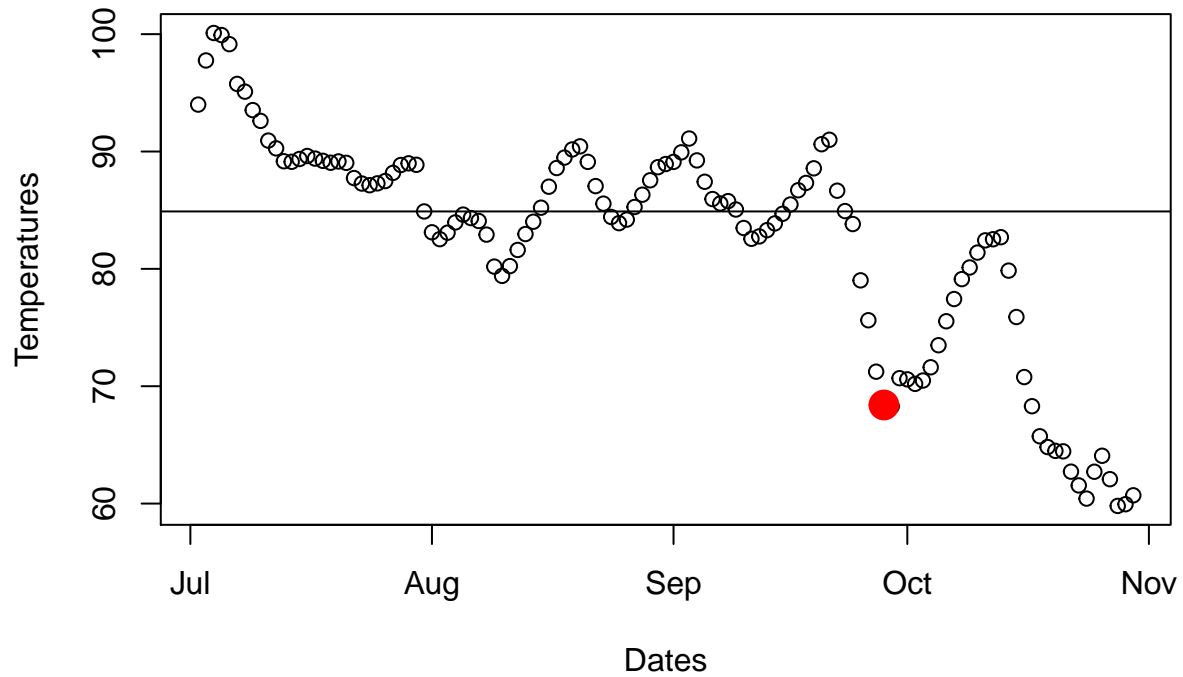
plot(x = xp, y = x, xlab = 'Dates', ylab='Temperatures', main=c('Atlanta Temperatures in ', as.character(day_change)),
points(x = xp[day_index], y= x[day_index], col = 'red', pch=19, cex = 2)
abline(h = median)
print(as.character(day_change))
}

```



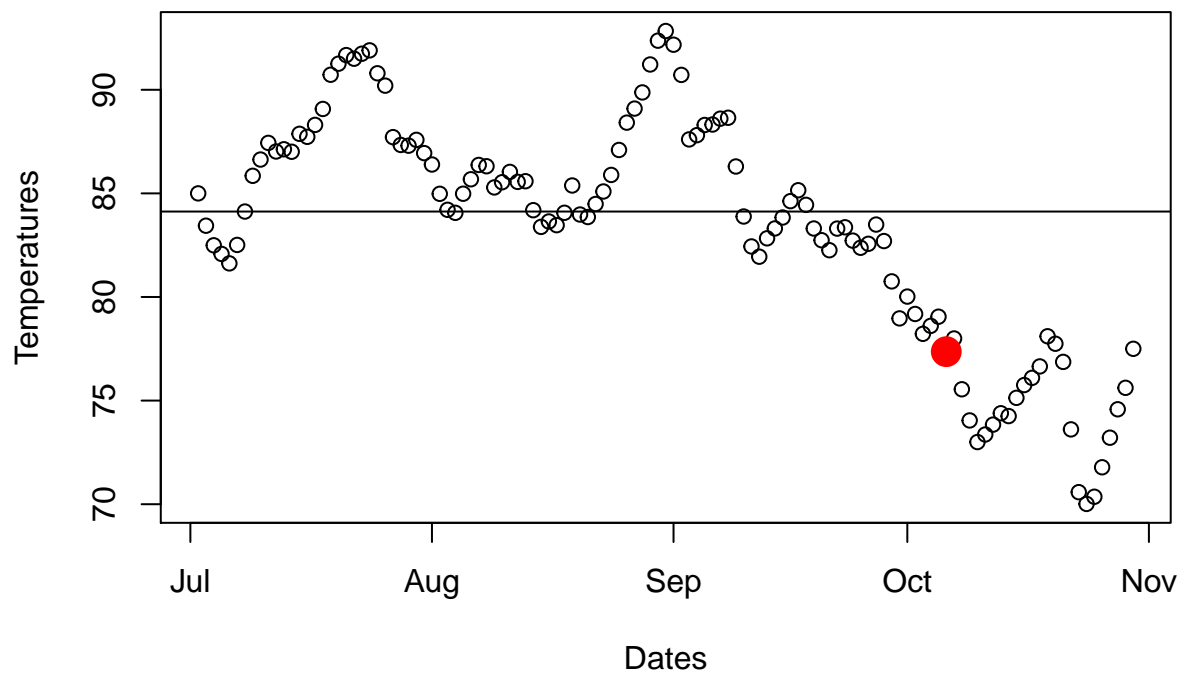
```
## [1] "21-Sep"
```

Atlanta Temperatures in 1997



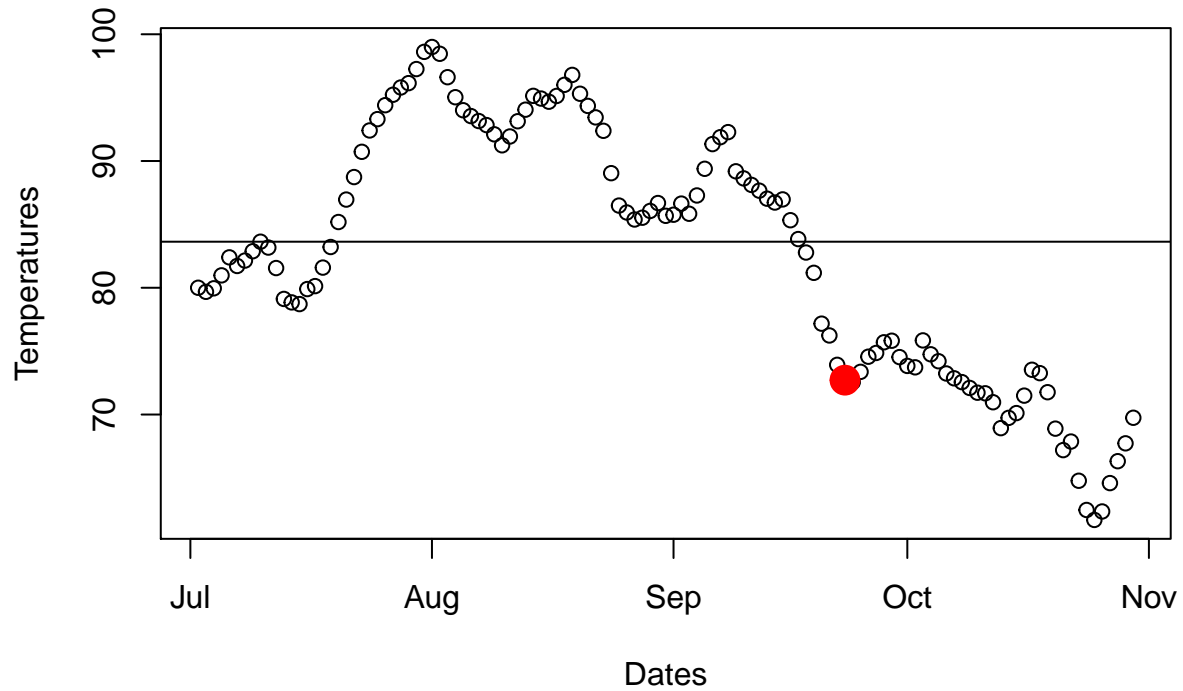
[1] "27-Sep"

Atlanta Temperatures in 1998



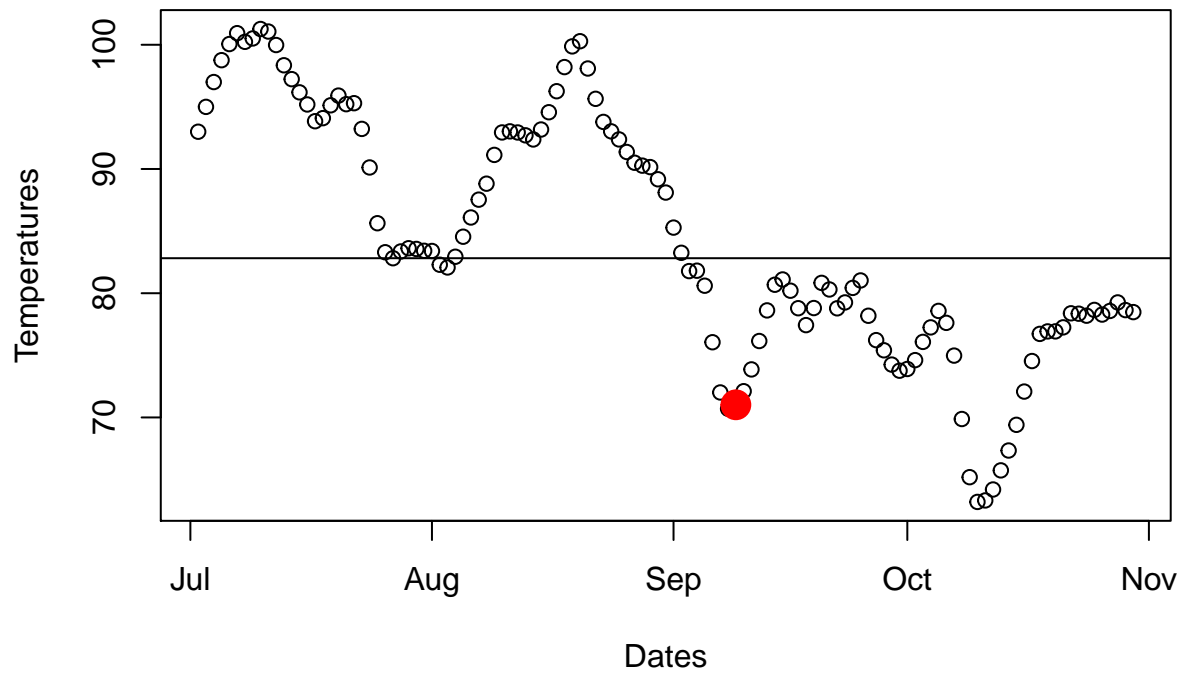
[1] "5-Oct"

Atlanta Temperatures in 1999



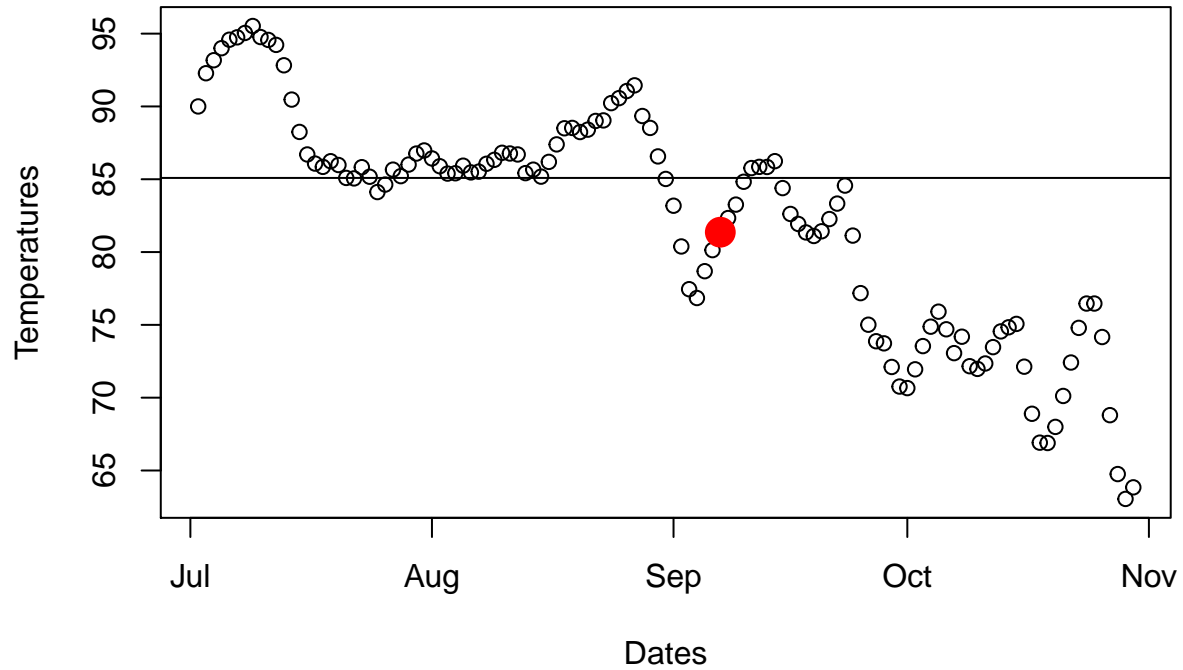
[1] "22-Sep"

Atlanta Temperatures in 2000



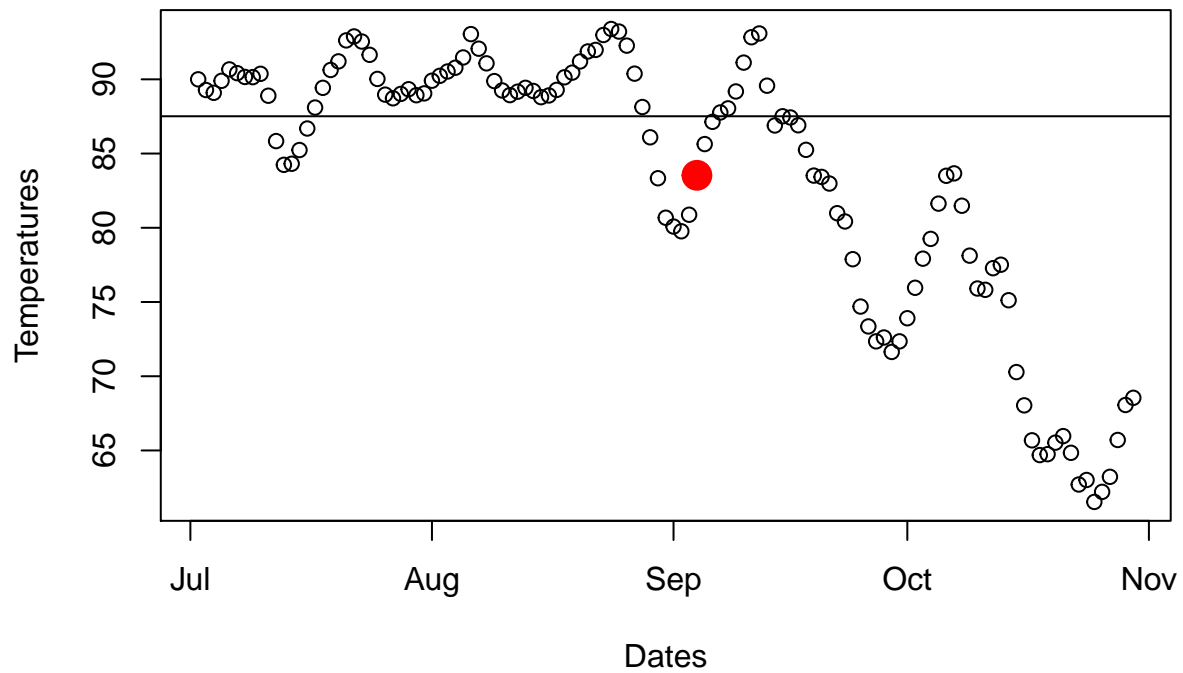
[1] "8-Sep"

Atlanta Temperatures in 2001



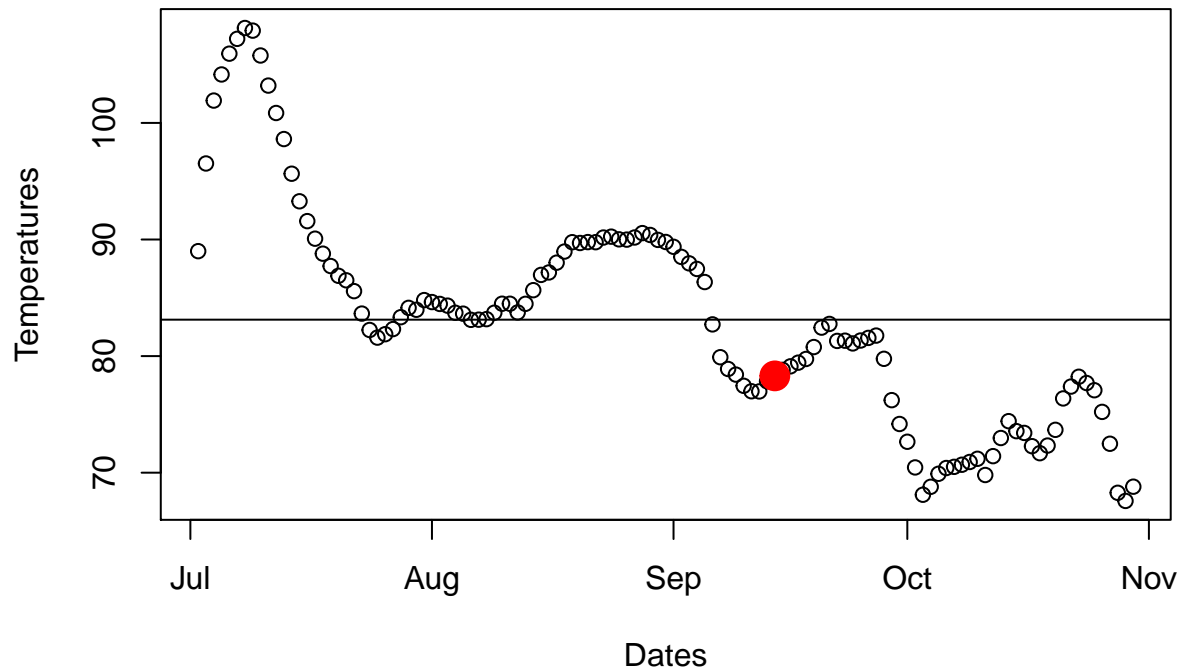
[1] "6-Sep"

Atlanta Temperatures in 2002



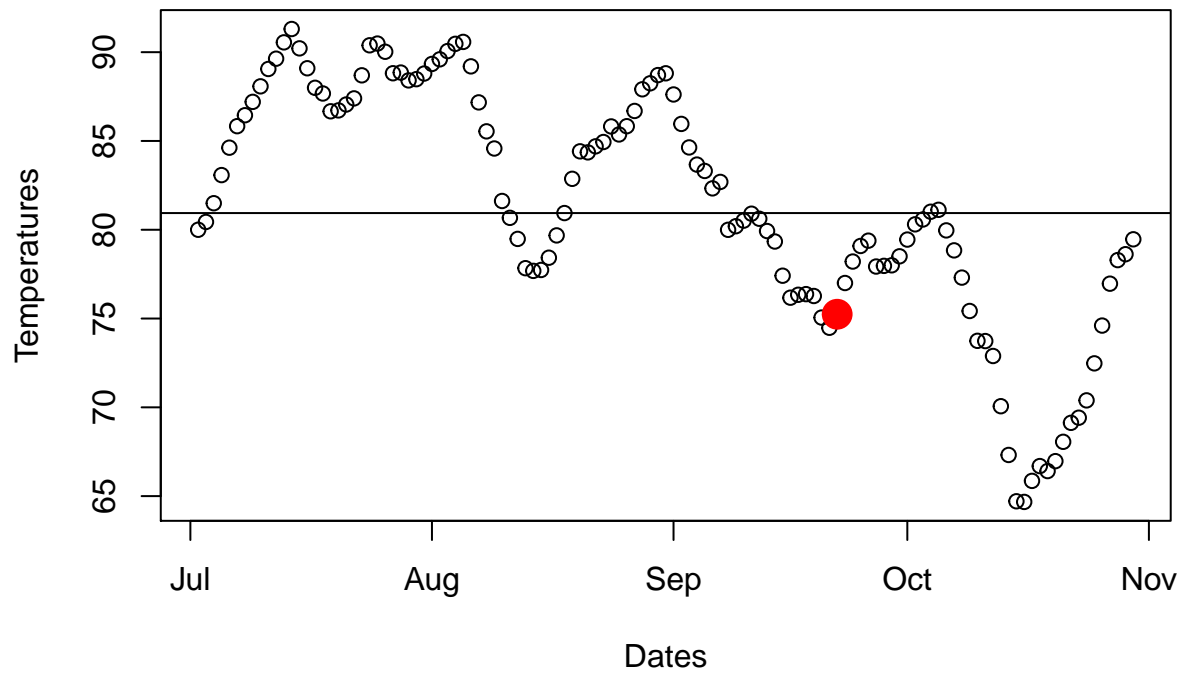
[1] "3-Sep"

Atlanta Temperatures in 2003



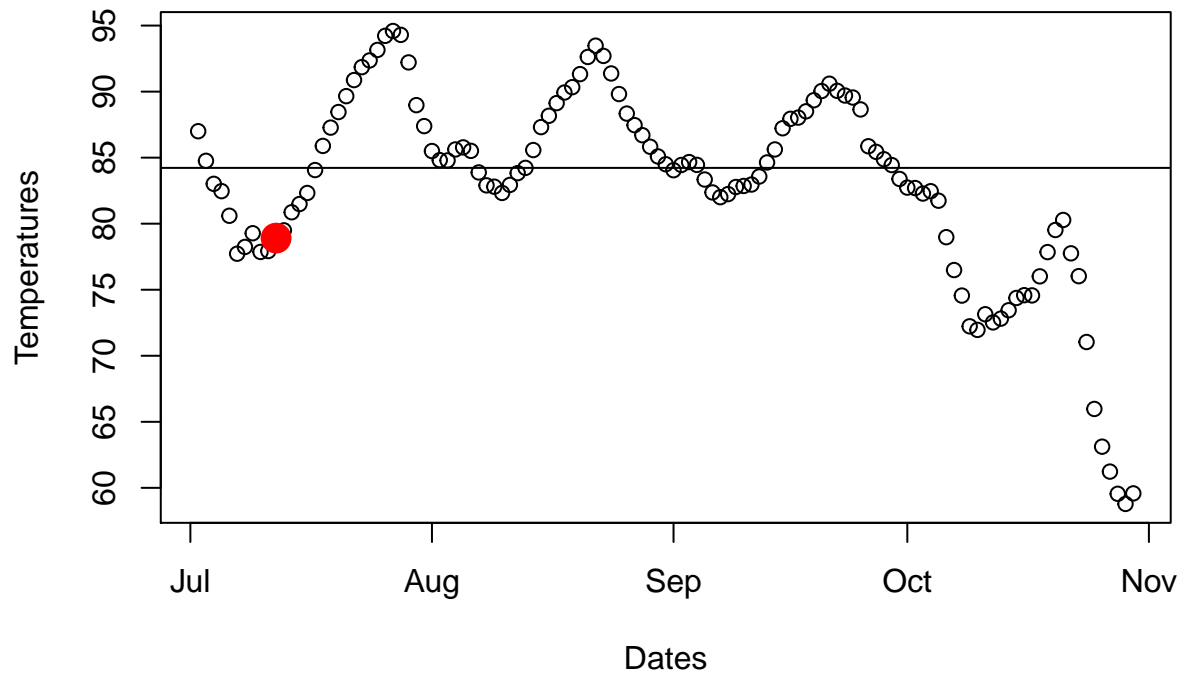
[1] "13-Sep"

Atlanta Temperatures in 2004



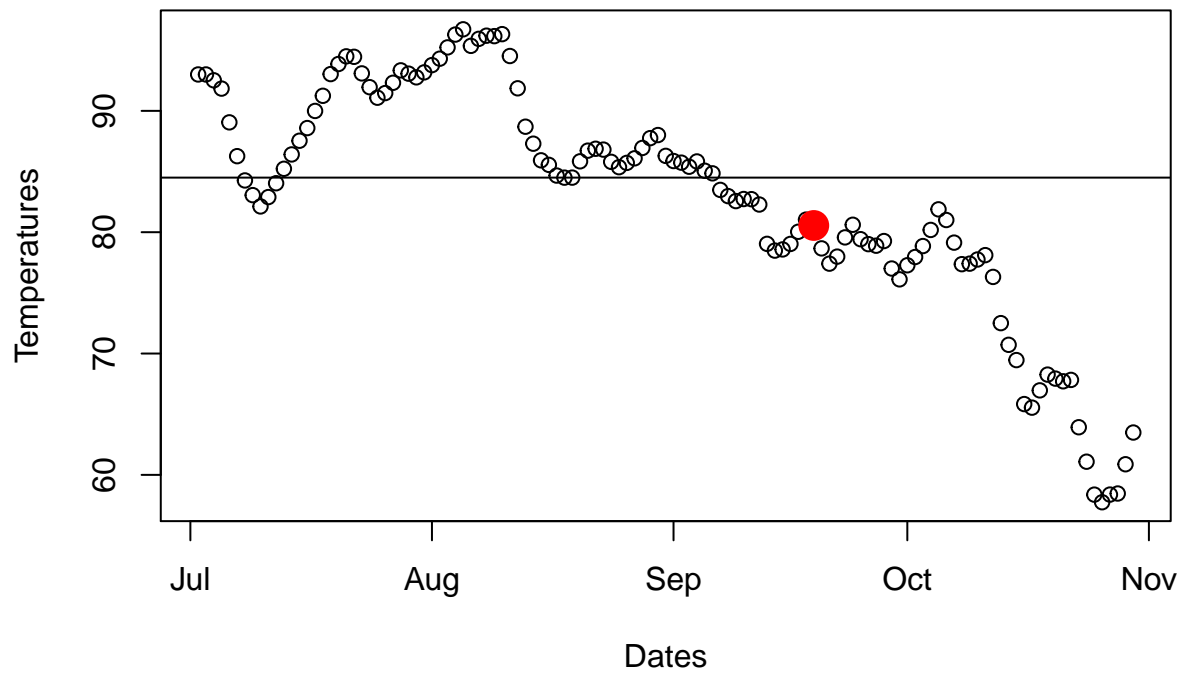
[1] "21-Sep"

Atlanta Temperatures in 2005



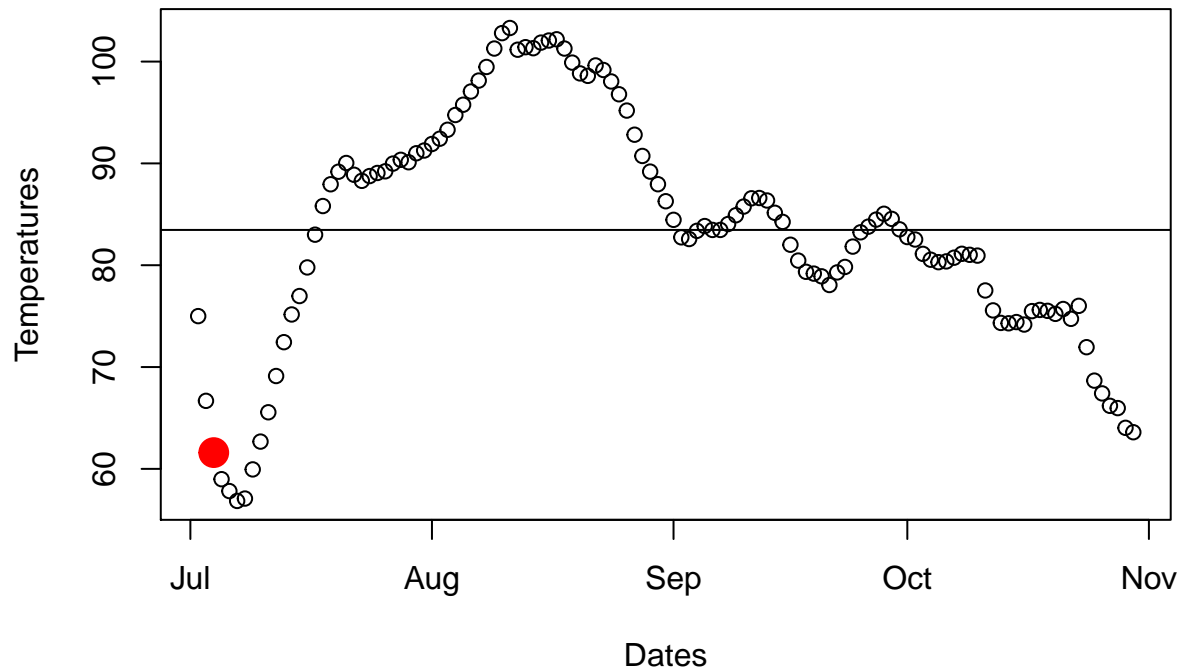
[1] "11-Jul"

Atlanta Temperatures in 2006



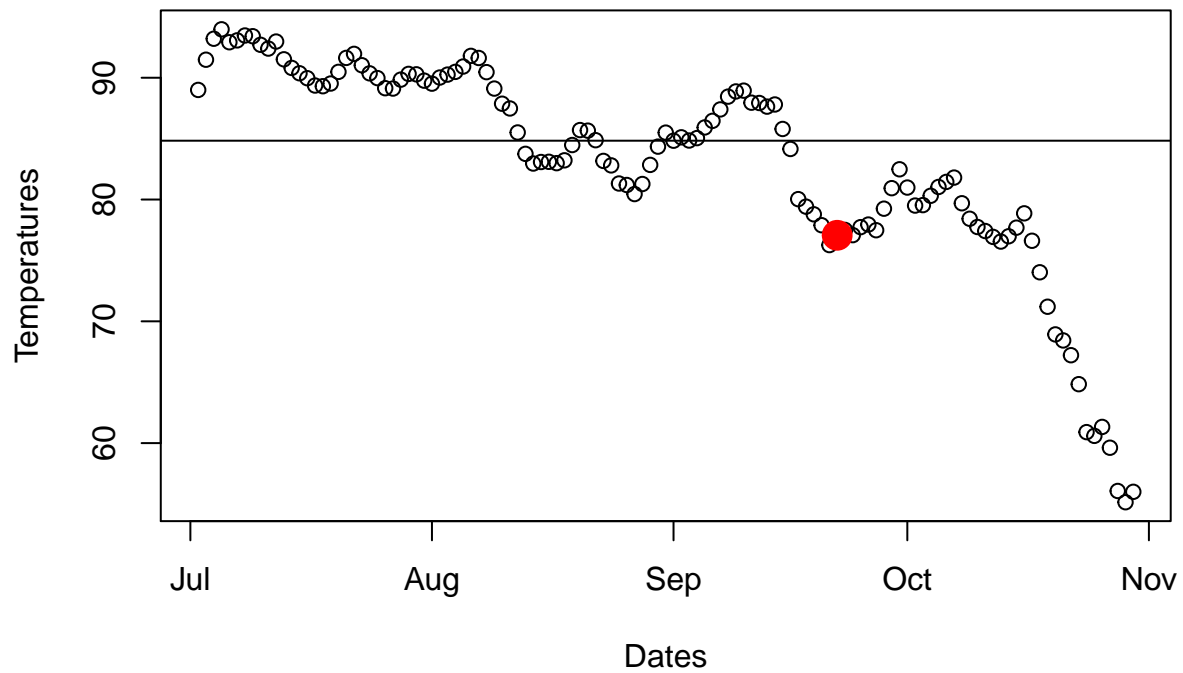
[1] "18-Sep"

Atlanta Temperatures in 2007



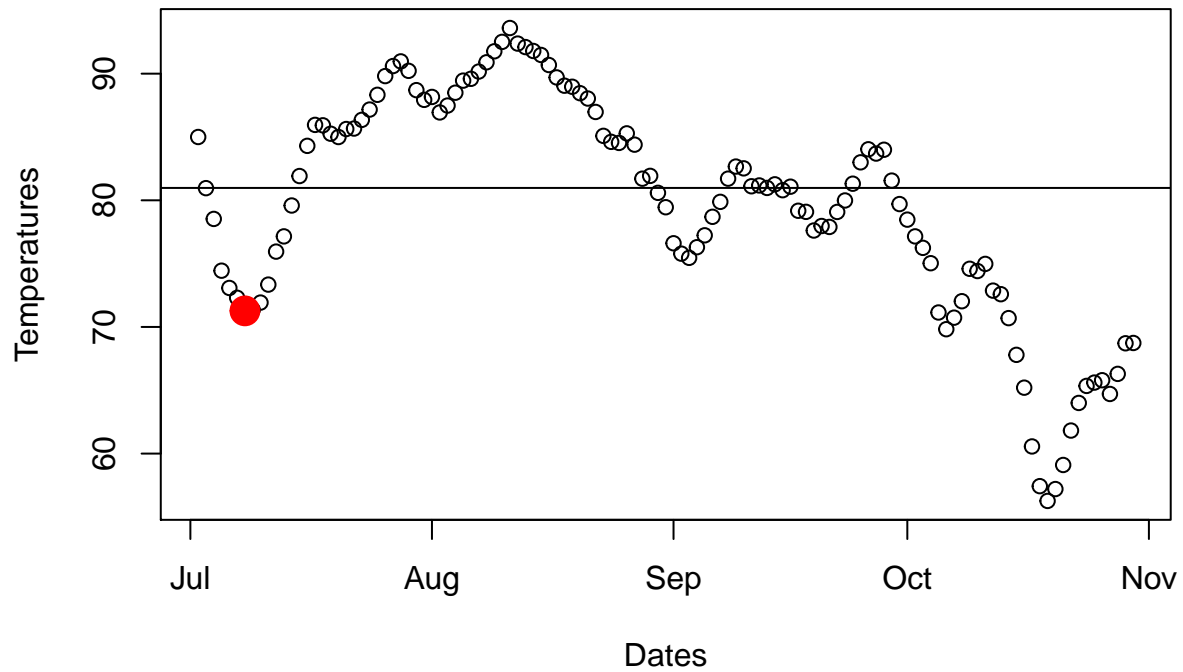
[1] "3-Jul"

Atlanta Temperatures in 2008



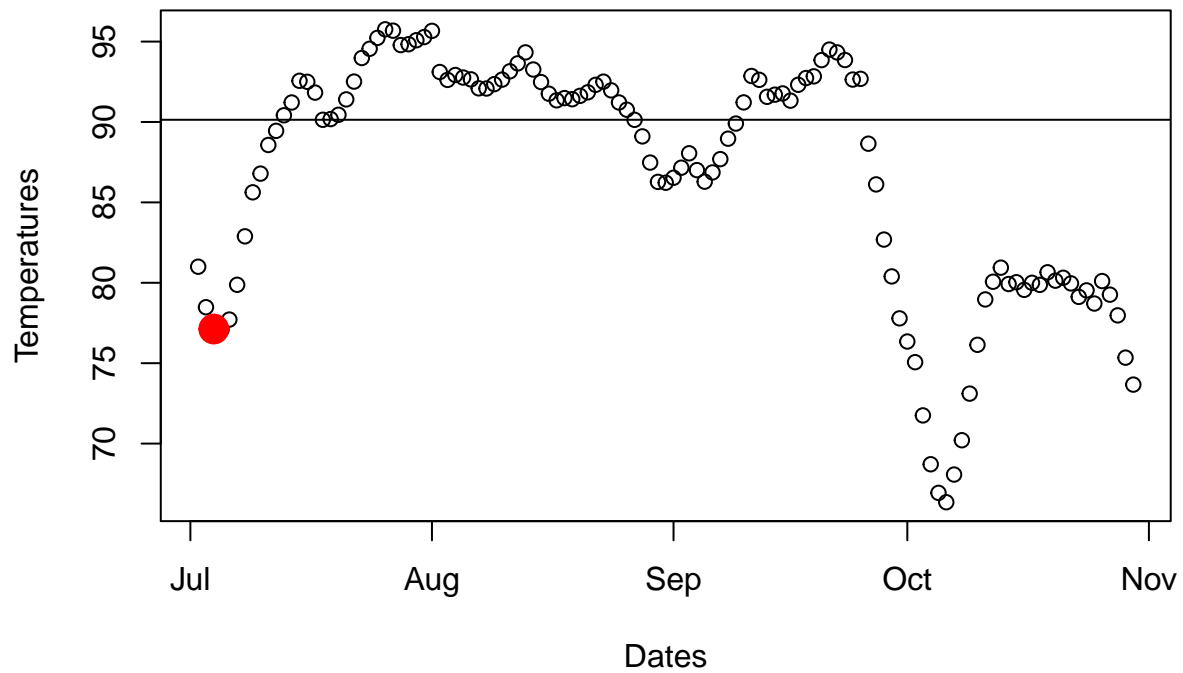
[1] "21-Sep"

Atlanta Temperatures in 2009



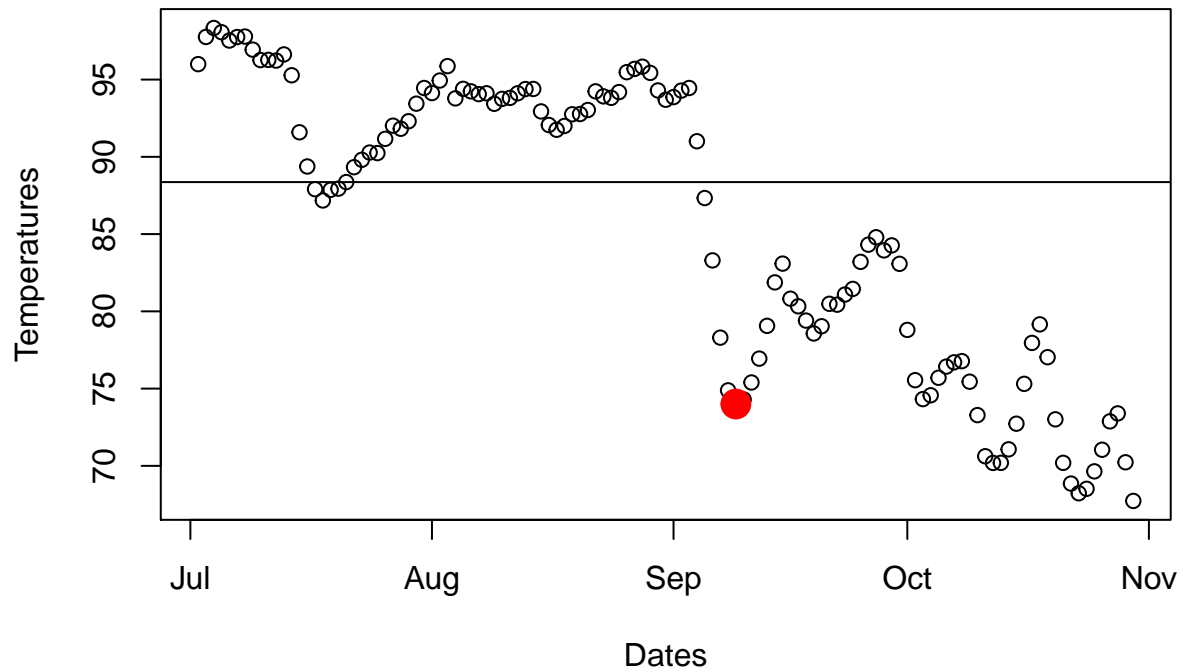
[1] "7-Jul"

Atlanta Temperatures in 2010



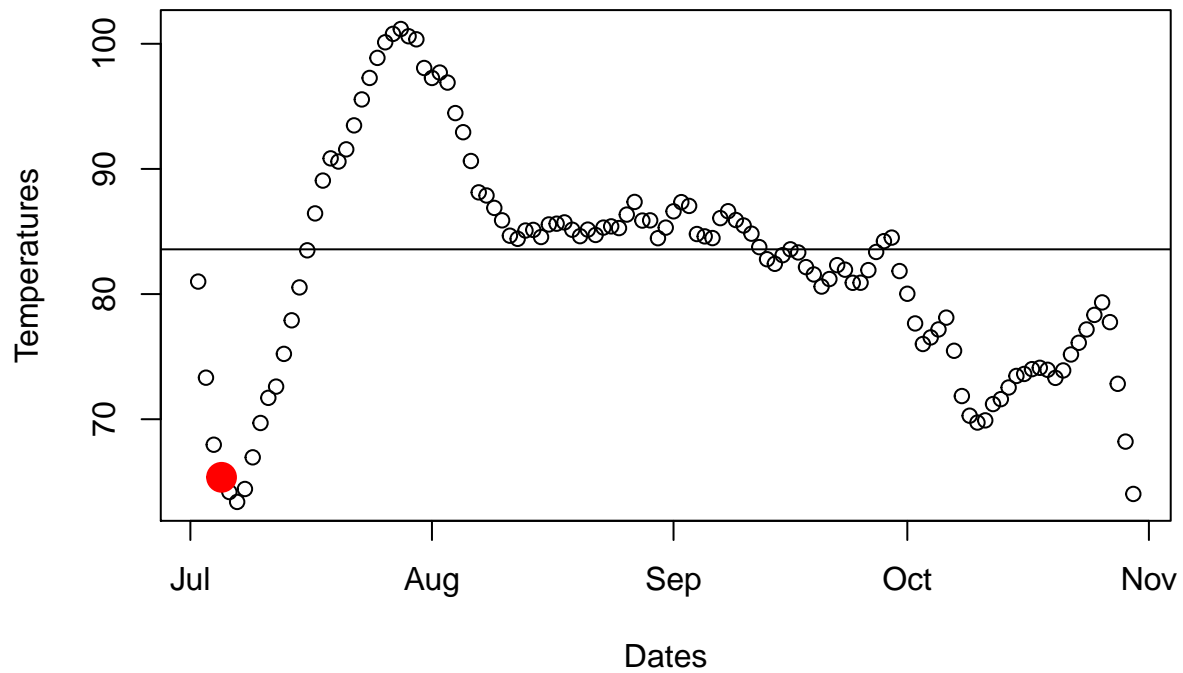
[1] "3-Jul"

Atlanta Temperatures in 2011



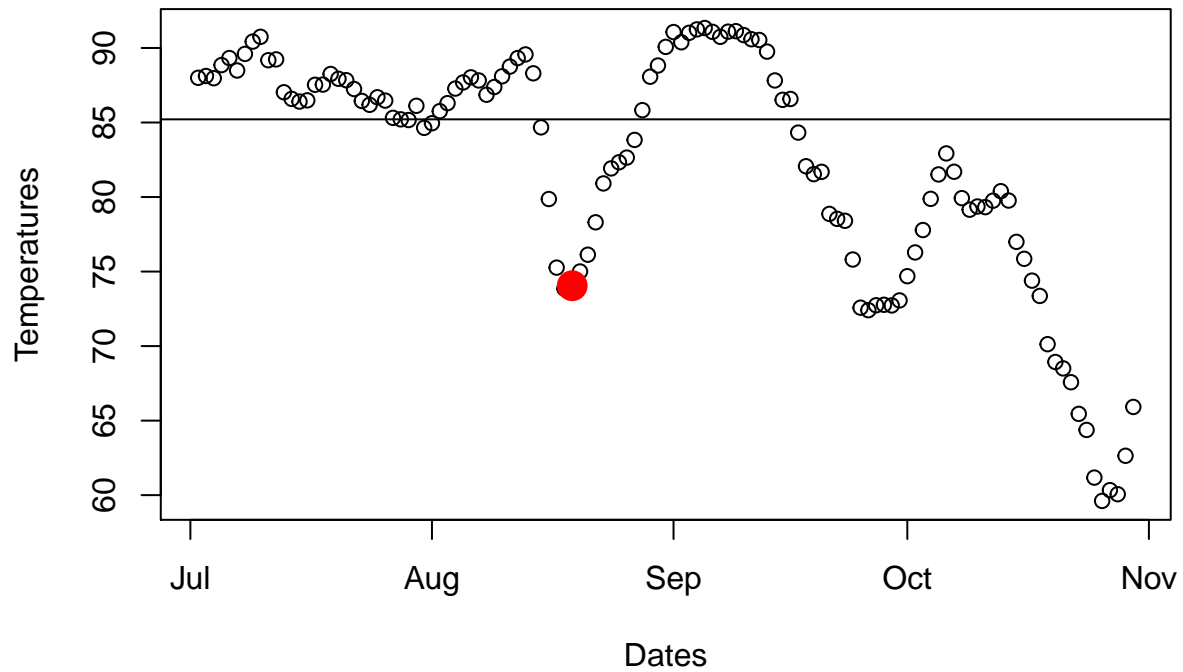
```
## [1] "8-Sep"
```

Atlanta Temperatures in 2012



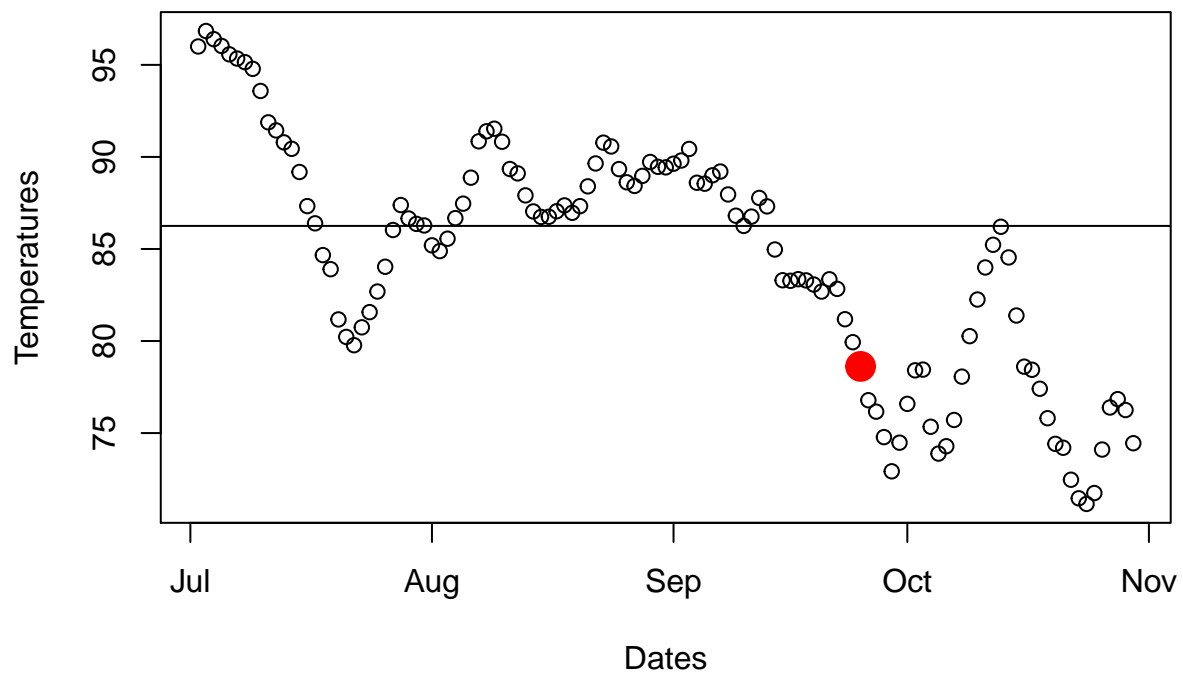
```
## [1] "4-Jul"
```

Atlanta Temperatures in 2013



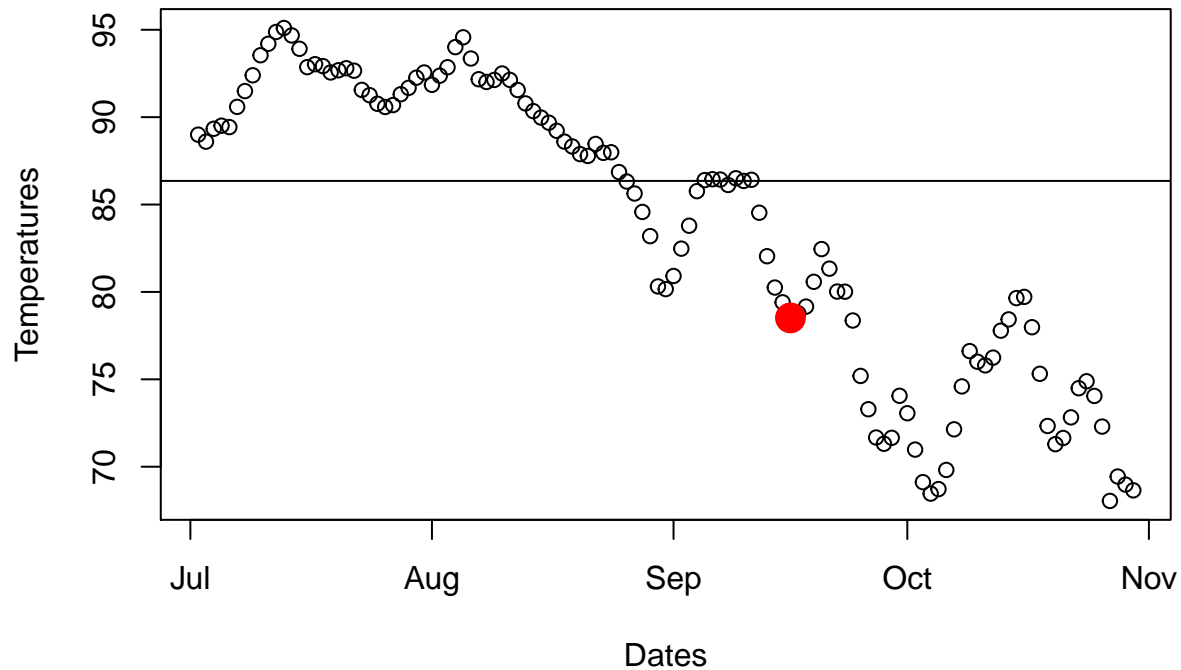
[1] "18-Aug"

Atlanta Temperatures in 2014



[1] "24-Sep"

Atlanta Temperatures in 2015

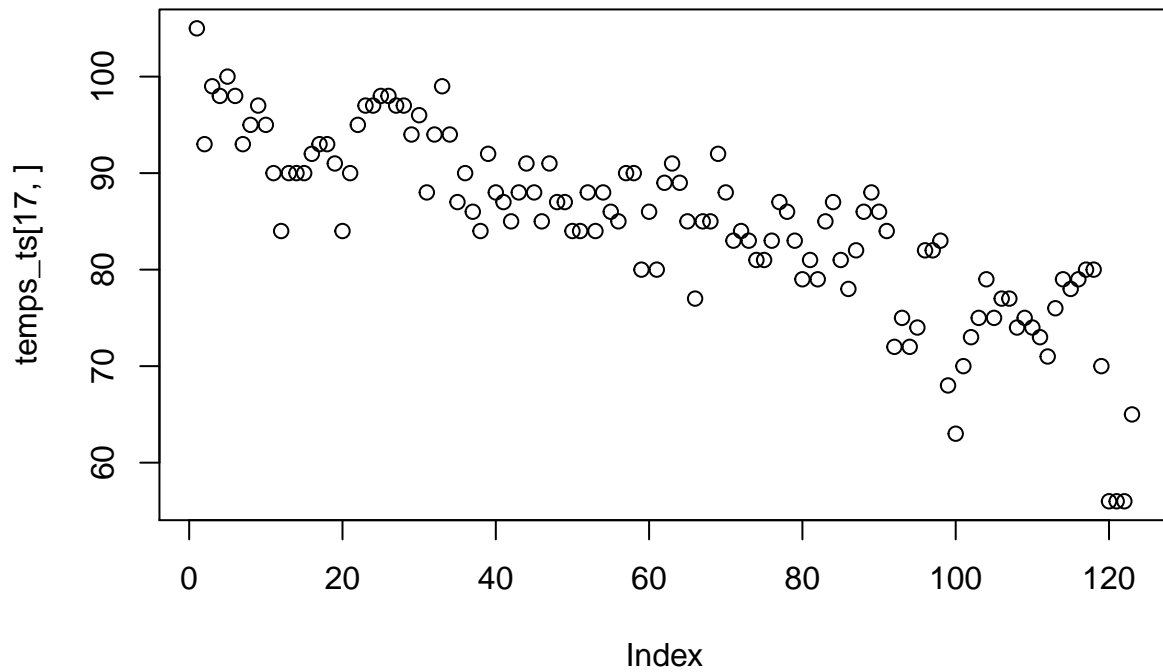


```
## [1] "15-Sep"
```

The exponential smoothing is causing issues at the beginning of a few years where the temperatures drop off tremendously in as early as July (2005, 2007, 2009, 2010 and 2012). This is because the trend at the beginning of these years are decreasing, so the exponential smoothing is amplifying this drop off. As an effect, this is causing an extremely early change detection within our CUSUM algorithm. Let's examine 2012 and see what we can do

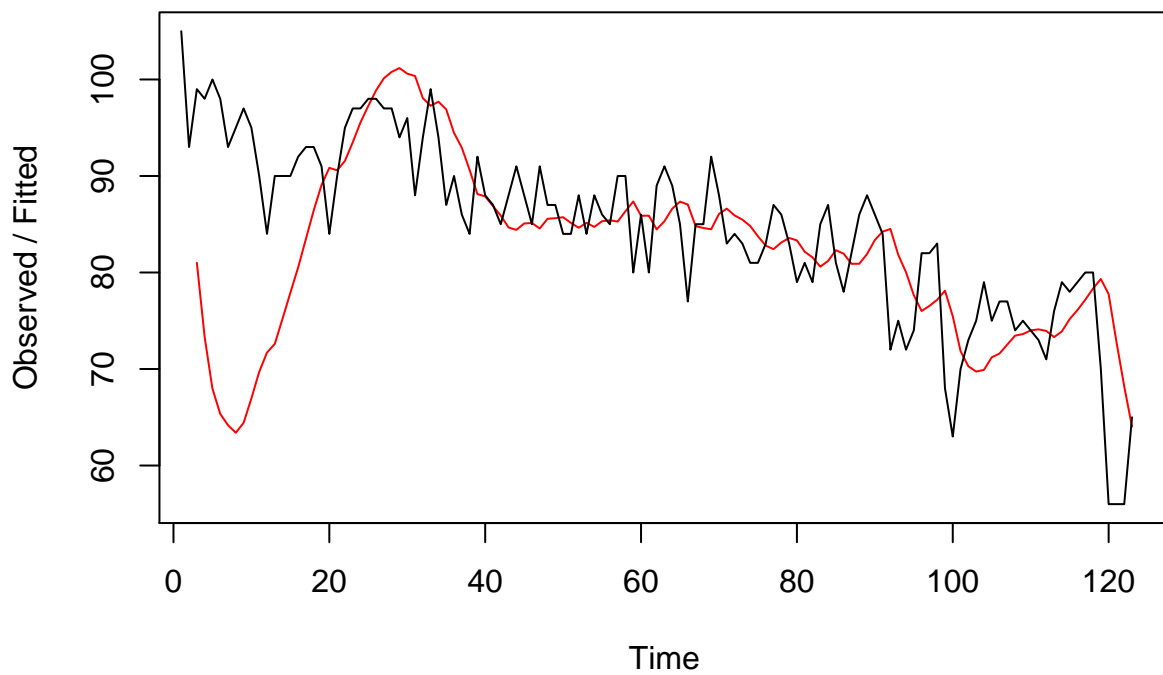
```
plot(temps_ts[17,], main = 'Atlanta Temperatures 2012')
```

Atlanta Temperatures 2012



```
plot(HoltWinters(temps_ts[17,], alpha = 0.2, beta = 0.2, gamma = FALSE))
```

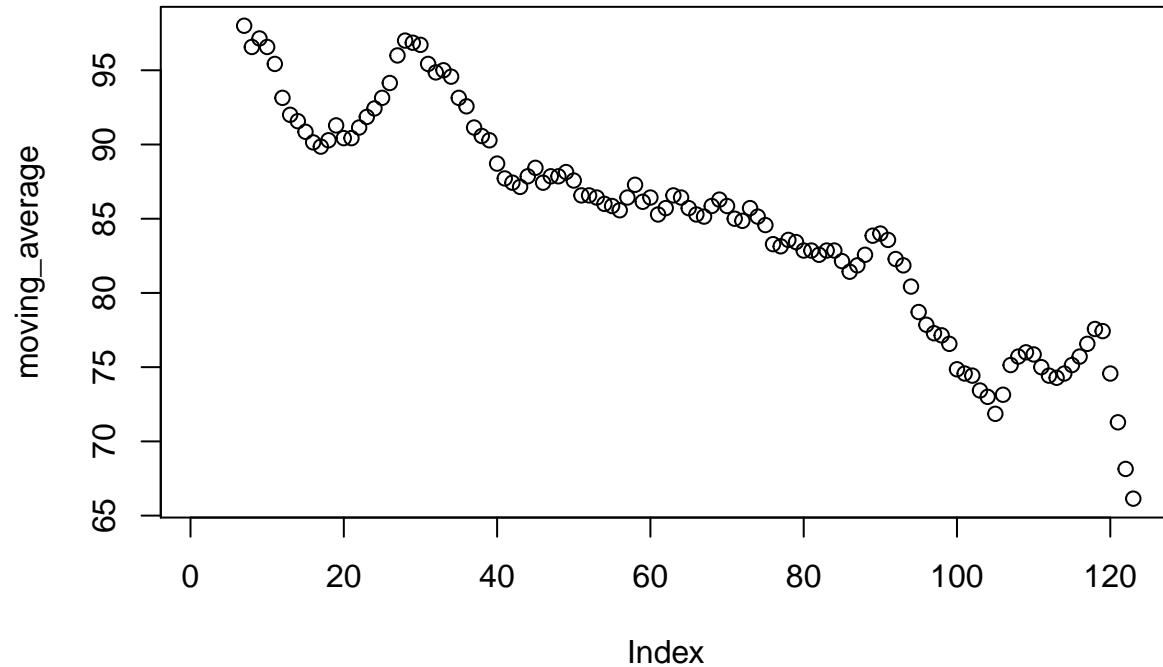
Holt-Winters filtering



The issue with the Holt-Winters filtering is pretty clear. Let's try to smooth the data with a 7 day moving average, and then apply an exponential smoothing to try and fix this issue

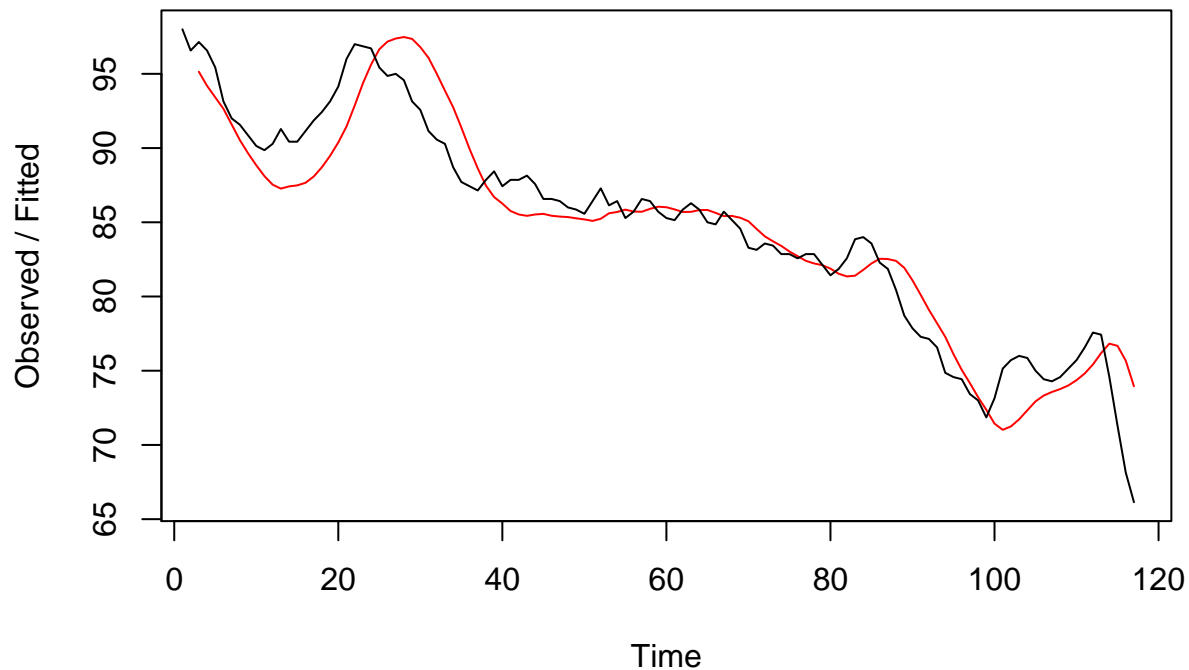
```
moving_average = SMA(temps[17:], n=7)
plot(moving_average, main = '7 Day Moving Average for 2012 Temperatures in Atlanta')
```

7 Day Moving Average for 2012 Temperatures in Atlanta



```
holt = HoltWinters(moving_average[7:length(moving_average)], alpha = 0.2, beta = 0.2, gamma = FALSE)
plot(holt)
```

Holt-Winters filtering



The moving average appears to have fixed the massive drop off at the beginning of the month.
Let's apply CUSUM now to see the results for 2012

```
forecast <- holt$fitted[,1]

median = median(forecast)
s = forecast[1]
C = 0.2
threshold = -10
for(i in seq(from = 2, to = length(forecast))){
  s = s + (forecast[i] - median + C)

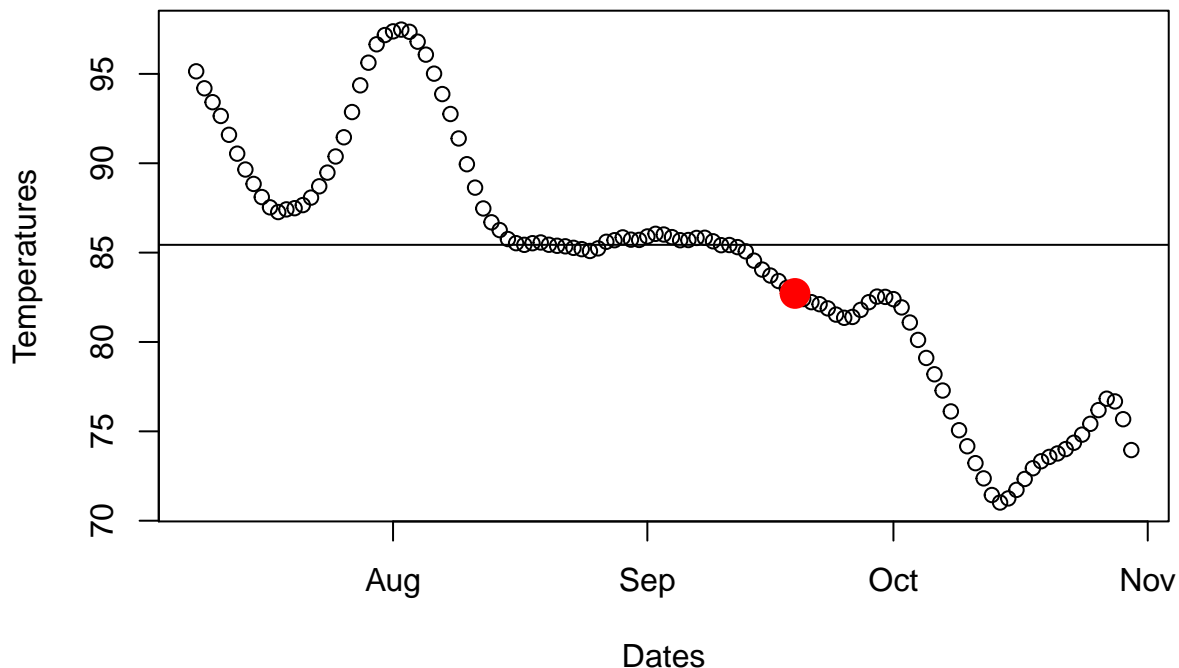
  s = min(0, s)

  if (s < threshold){
    day_change = days[i+6]
    day_index = i
    break
  }
}
print(day_change)

## [1] "18-Sep"

plot(x = xp[7:length(xp)], y = forecast, xlab = 'Dates', ylab='Temperatures', main='Temperatures in 2012',
points(x = xp[day_index+6], y= forecast[day_index], col = 'red',pch=19, cex = 2)
abline(h = median)
```

Temperatures in 2012 w/ Median Temperature and 7 Day Moving Average



This is much better. Let's try again and apply a 7 day moving average to all data before exponentially smoothing and applying a CUSUM algorithm. We will also be saving the index

locations of the dates where summer officially ended to use to determine if the unofficial summer is getting later. That variable will be called “weather_change_index”

```
weather_change_index = rep(0, 20)
for (i in seq(1,20)){
  moving_average = SMA(temps[i,], n=7)
  temp_h <- HoltWinters(moving_average[7:length(moving_average)],alpha=0.2, beta = 0.2, gamma = FALSE)
  x <- temp_h$fitted[,1]
  median = median(x)
  C = 1
  threshold = -25
  s = x[1] - median + C

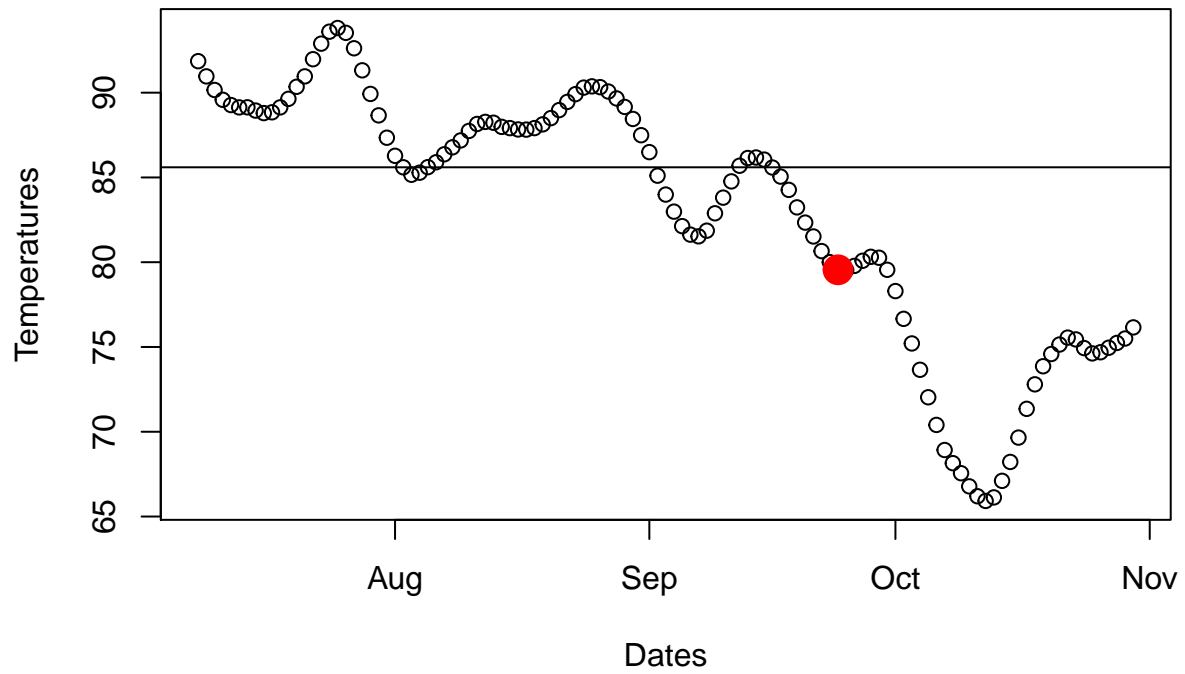
  for(j in seq(from = 2, to = length(x))){
    s = s + (x[j] - median + C)
    s = min(0, s)

    if (s < threshold){
      day_change = days[j+6]
      day_index = j
      weather_change_index[i] = day_index
      break
    }
  }

}

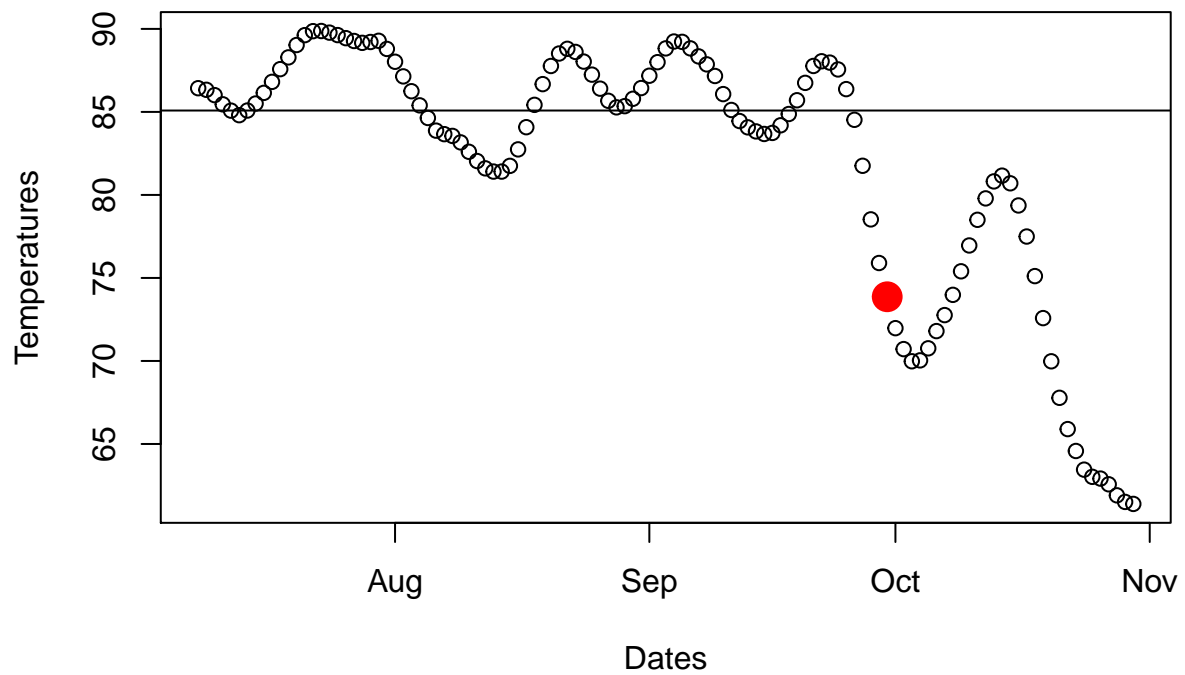
plot(x = xp[7:length(xp)], y = x, xlab = 'Dates', ylab='Temperatures', main=c('Atlanta Temperatures in'))
points(x = xp[day_index+6], y= x[day_index], col = 'red', pch=19, cex = 2)
abline(h = median)
print(as.character(day_change))
}
```


Atlanta Temperatures in 1996



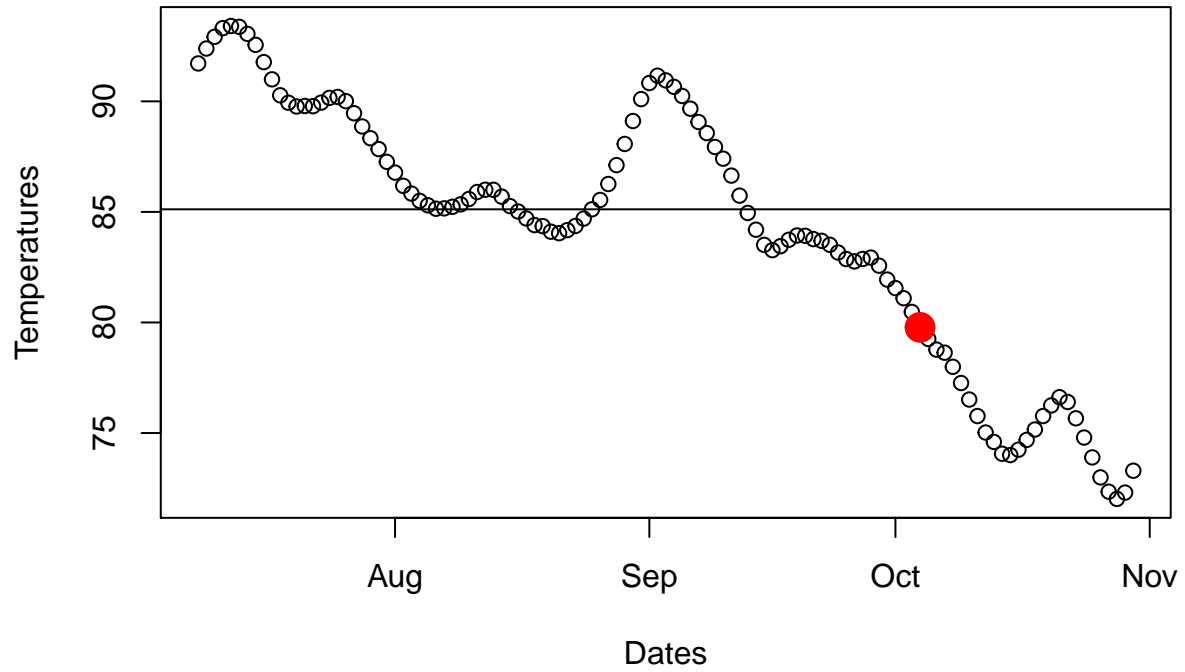
[1] "23-Sep"

Atlanta Temperatures in 1997



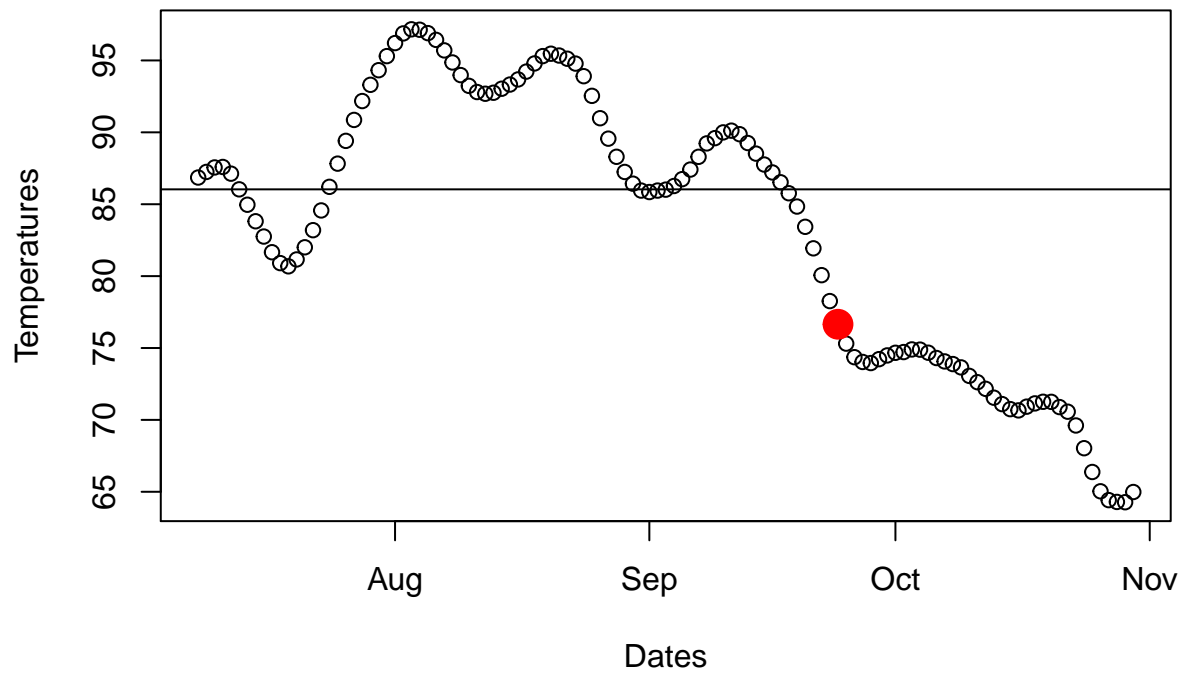
[1] "29-Sep"

Atlanta Temperatures in 1998



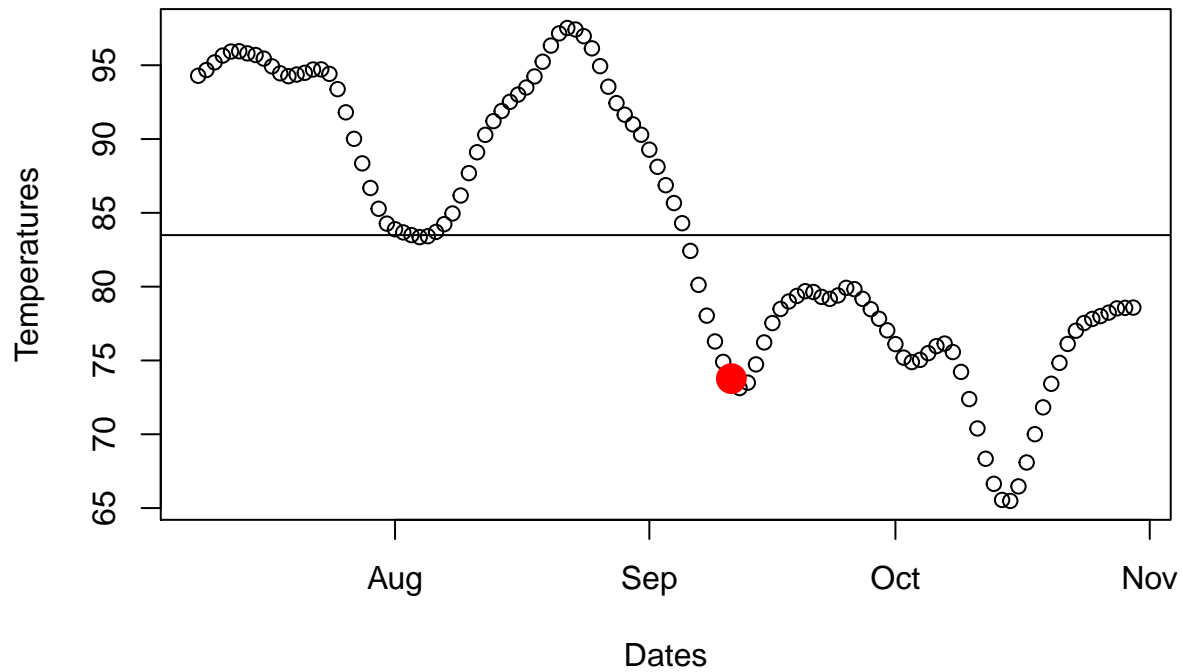
[1] "3-Oct"

Atlanta Temperatures in 1999



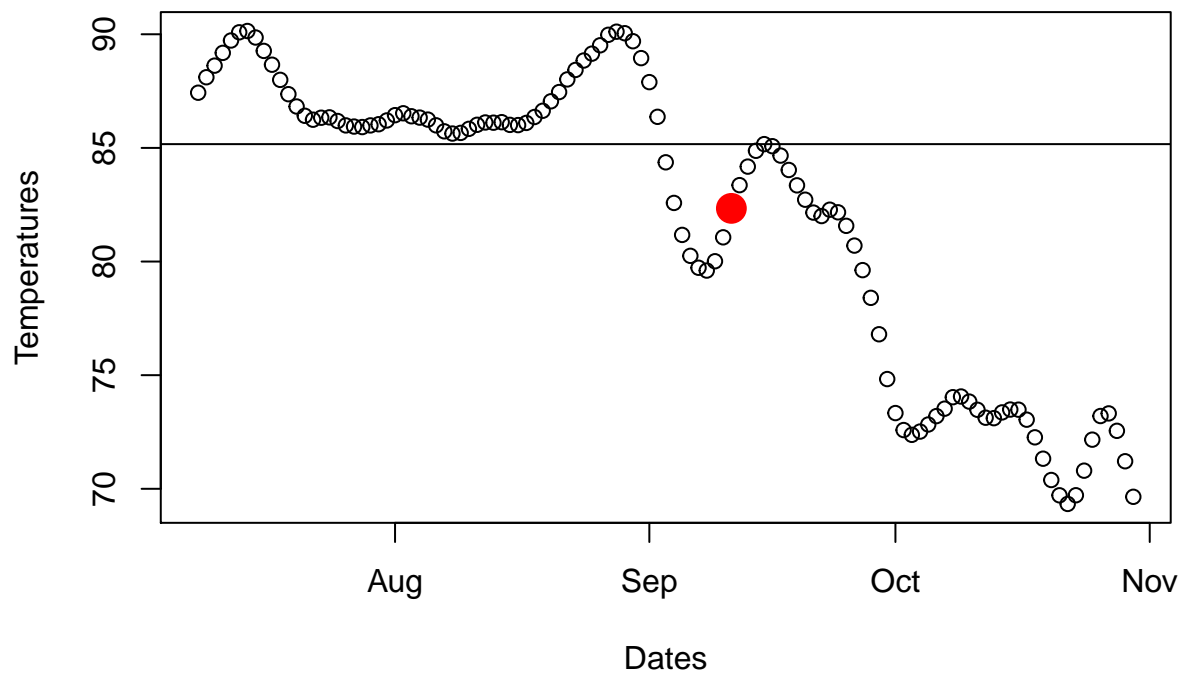
[1] "23-Sep"

Atlanta Temperatures in 2000



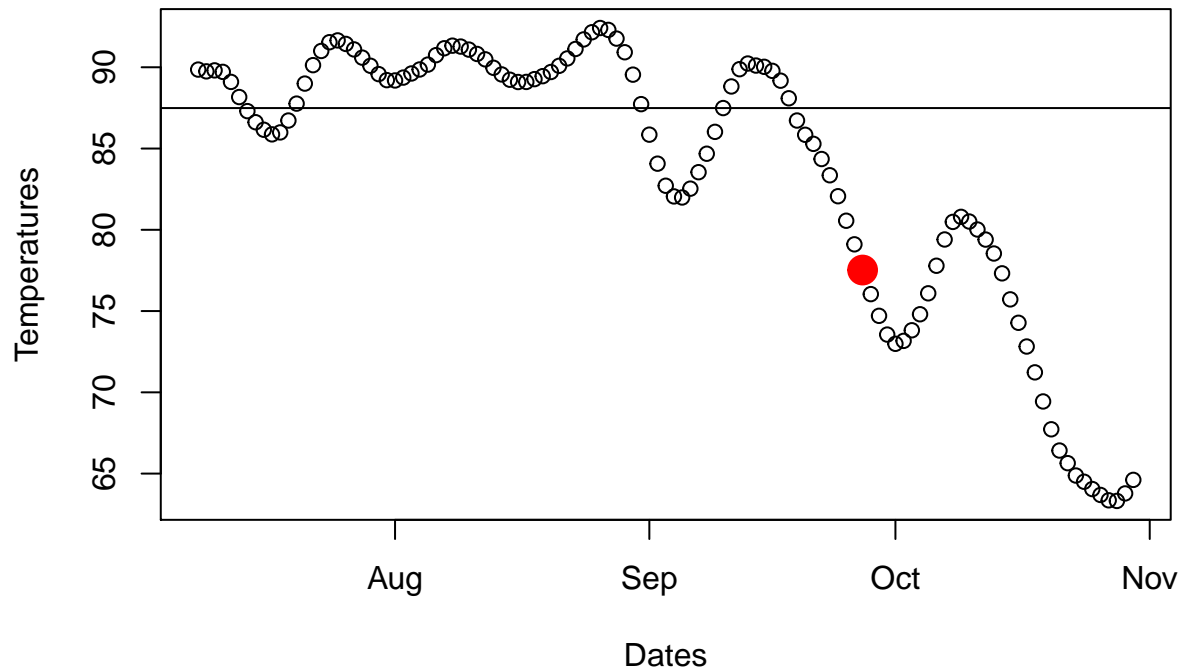
[1] "10-Sep"

Atlanta Temperatures in 2001



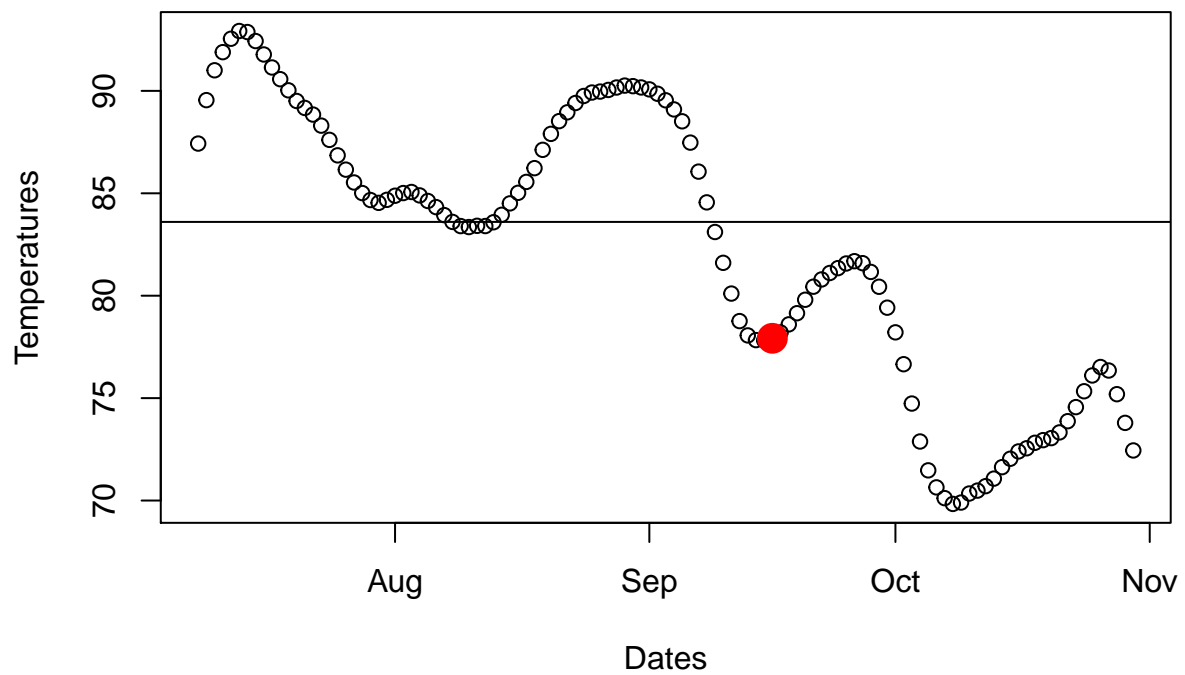
[1] "10-Sep"

Atlanta Temperatures in 2002



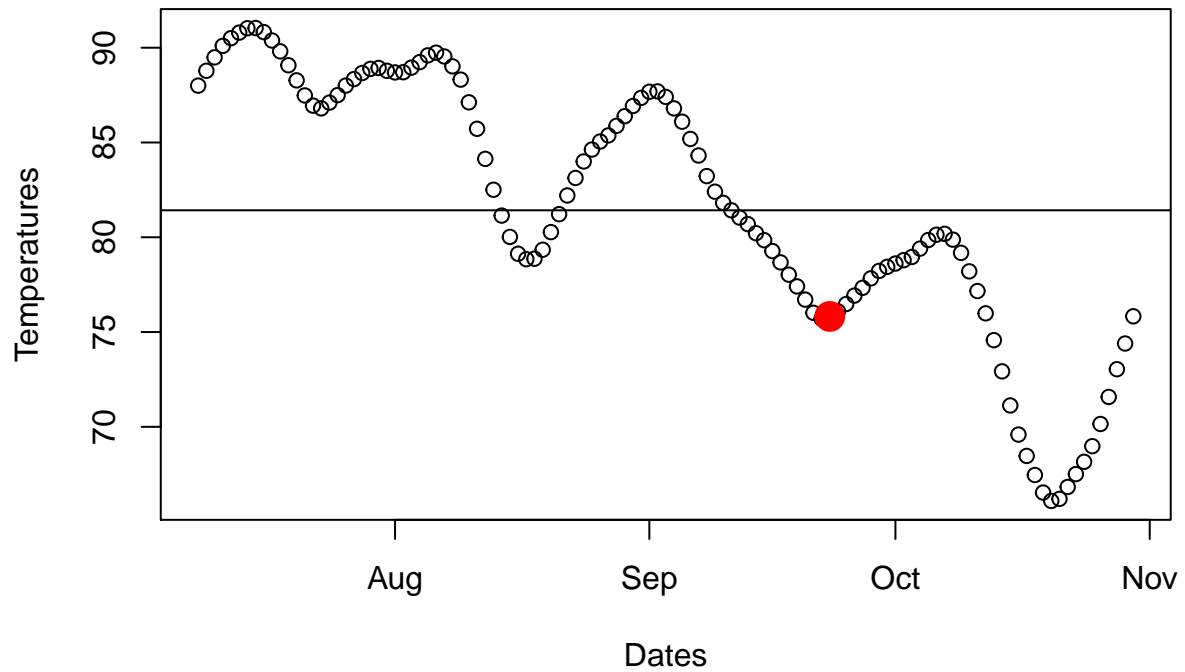
[1] "26-Sep"

Atlanta Temperatures in 2003

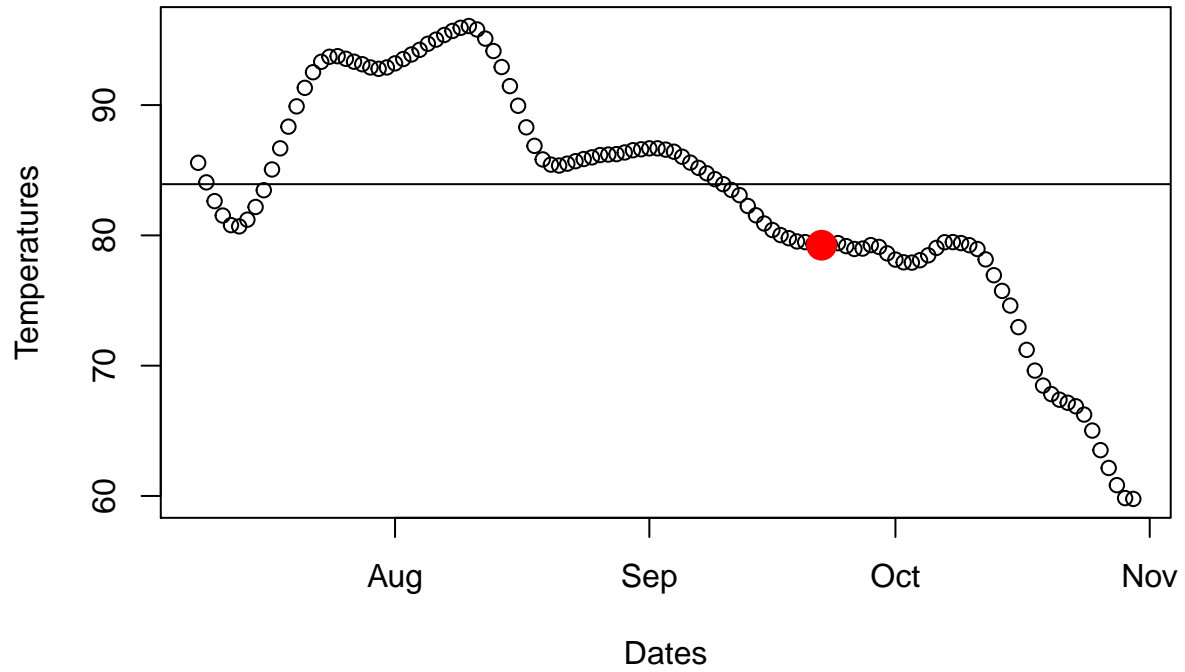


[1] "15-Sep"

Atlanta Temperatures in 2004

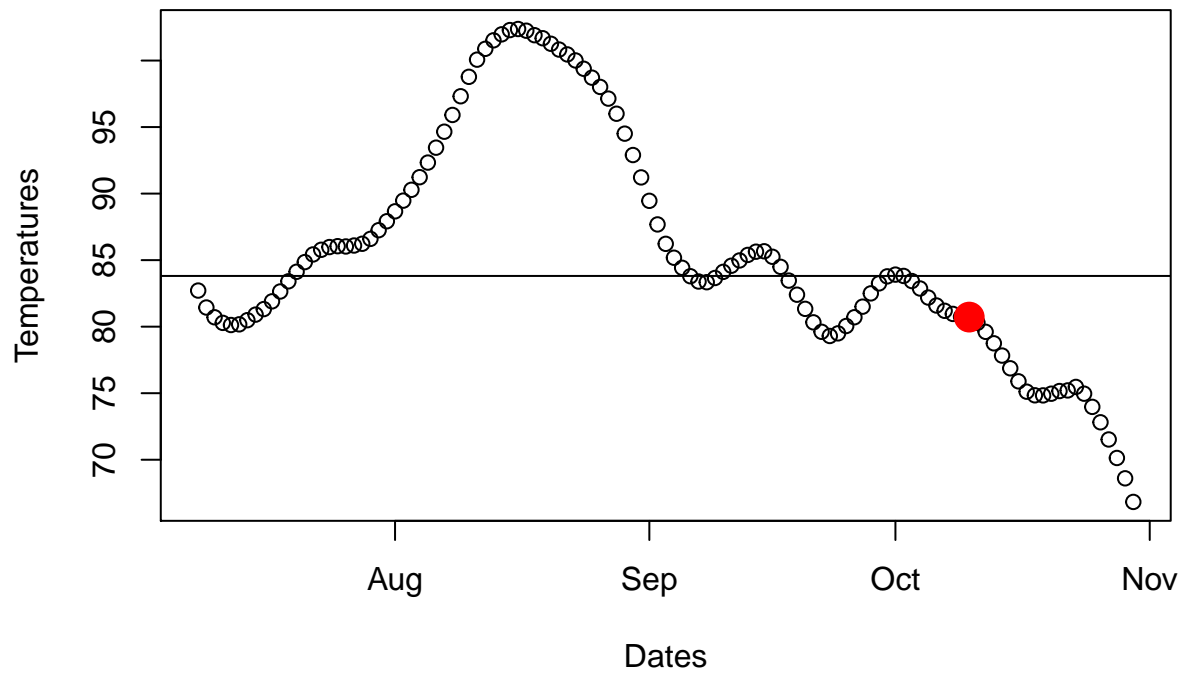


Atlanta Temperatures in 2006



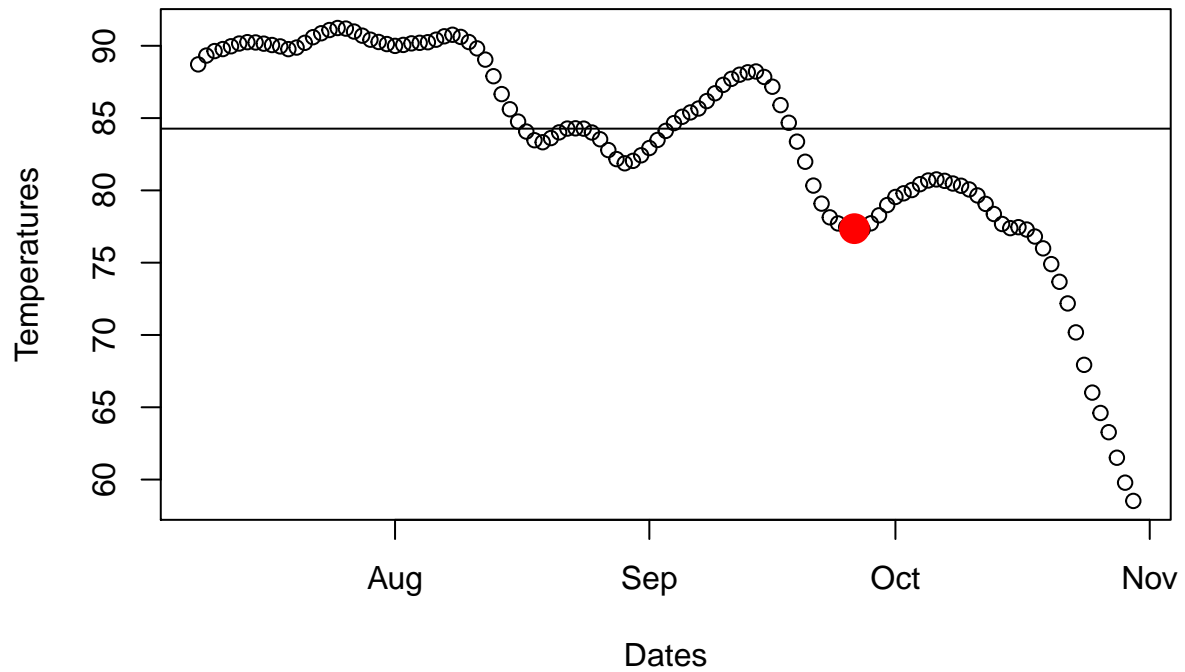
[1] "21-Sep"

Atlanta Temperatures in 2007



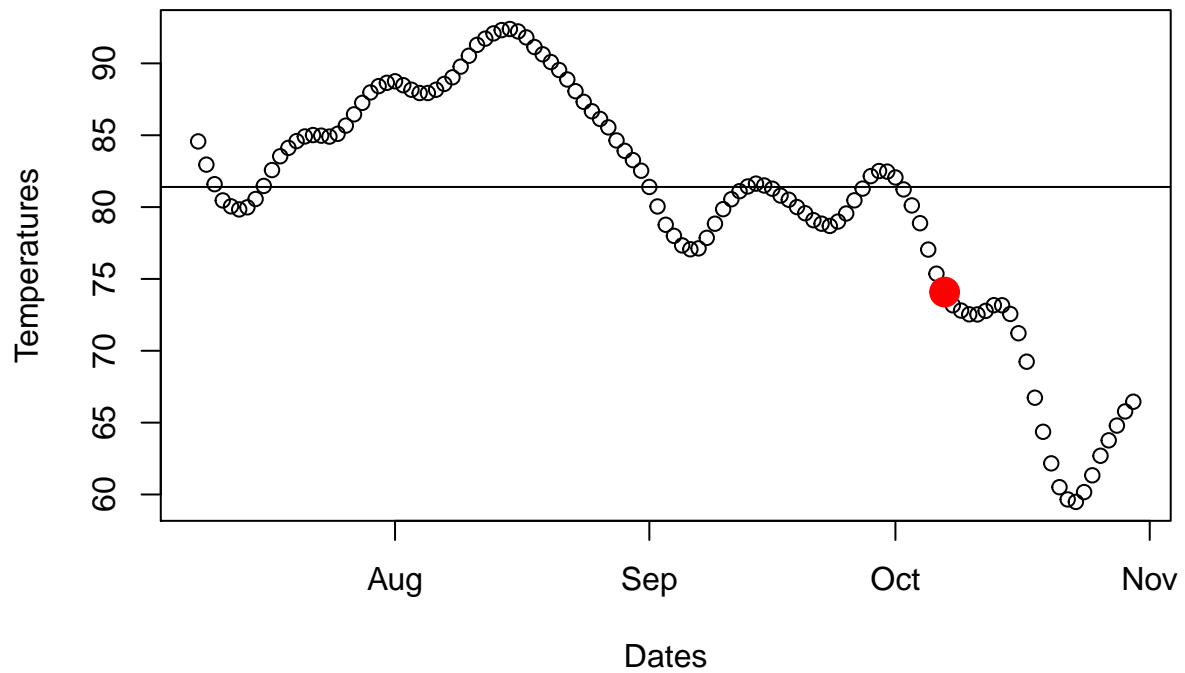
[1] "9-Oct"

Atlanta Temperatures in 2008



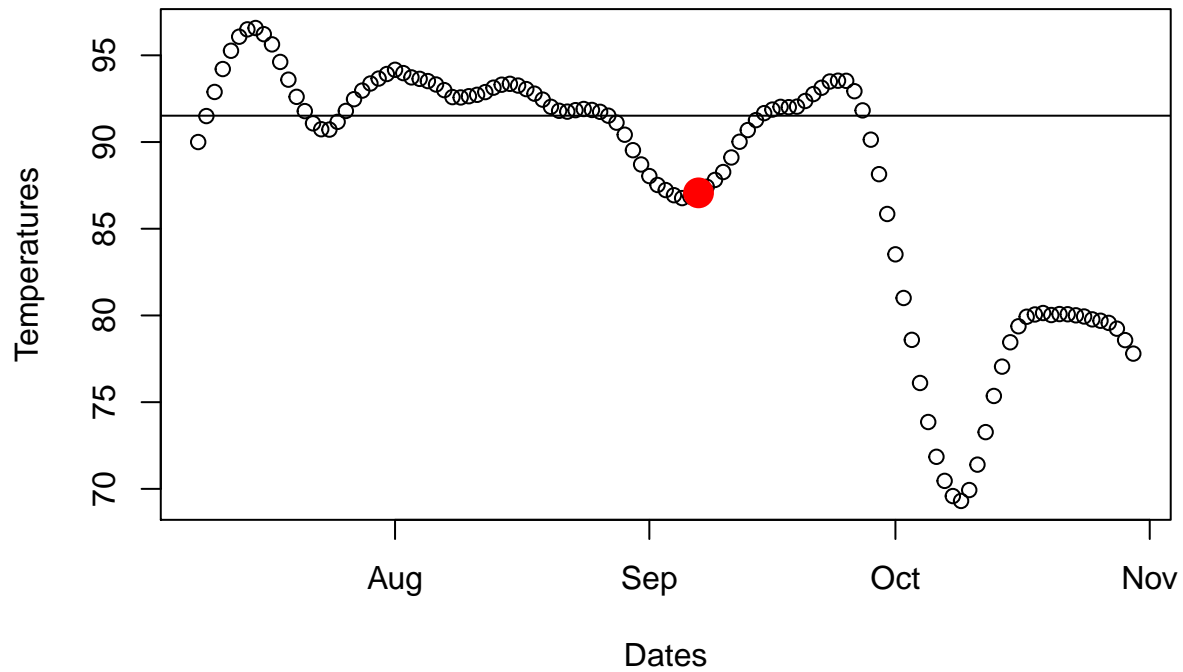
[1] "25-Sep"

Atlanta Temperatures in 2009



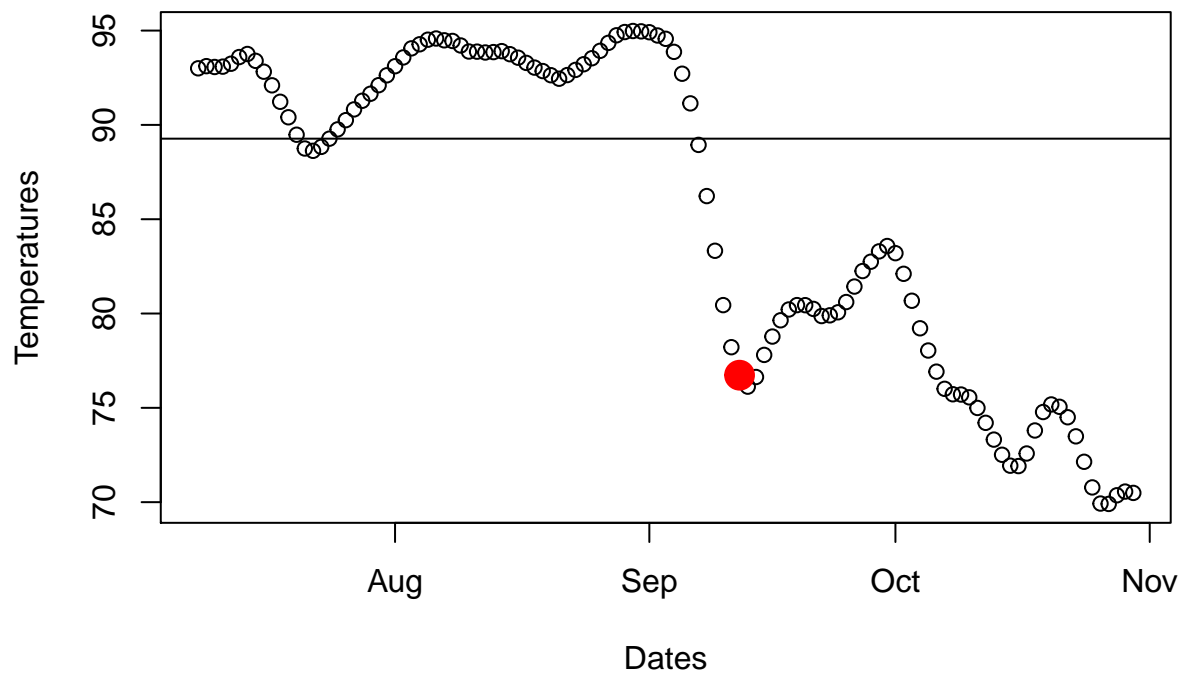
[1] "6-Oct"

Atlanta Temperatures in 2010



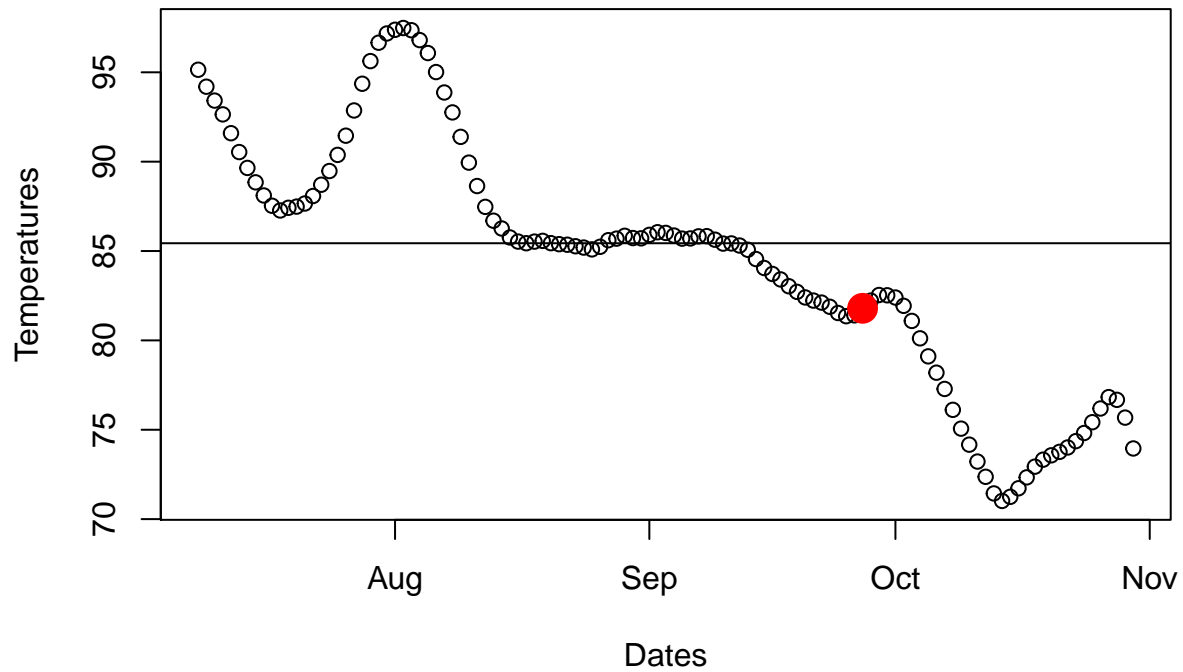
[1] "6-Sep"

Atlanta Temperatures in 2011



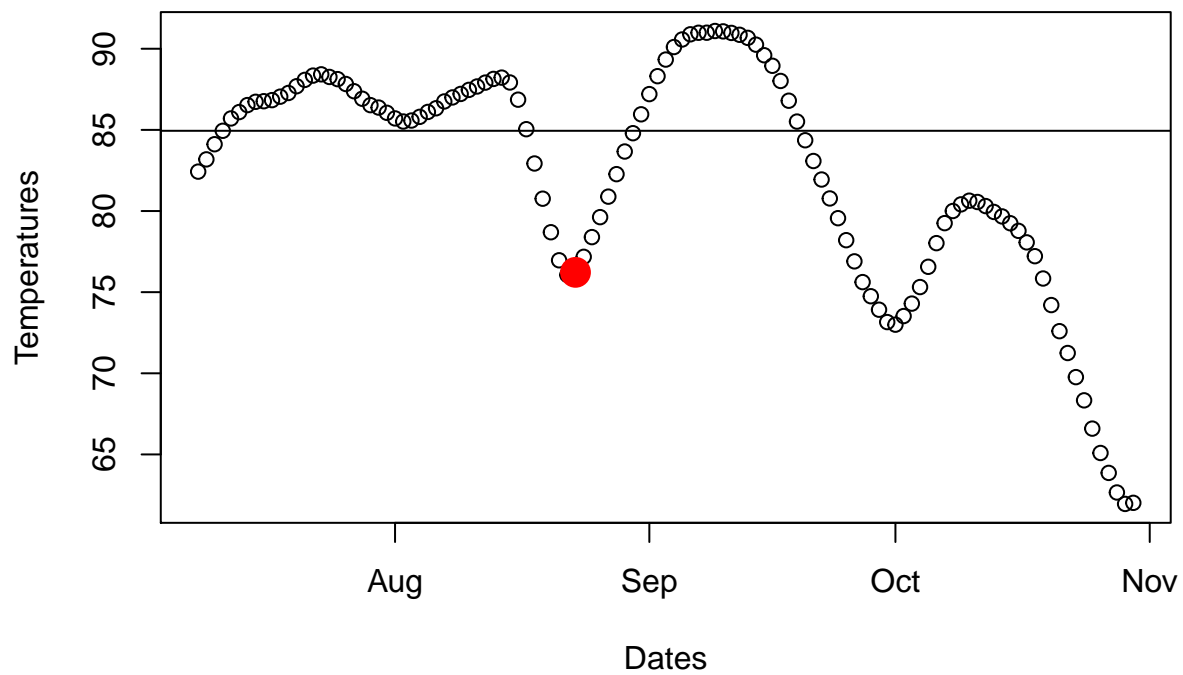
[1] "11-Sep"

Atlanta Temperatures in 2012



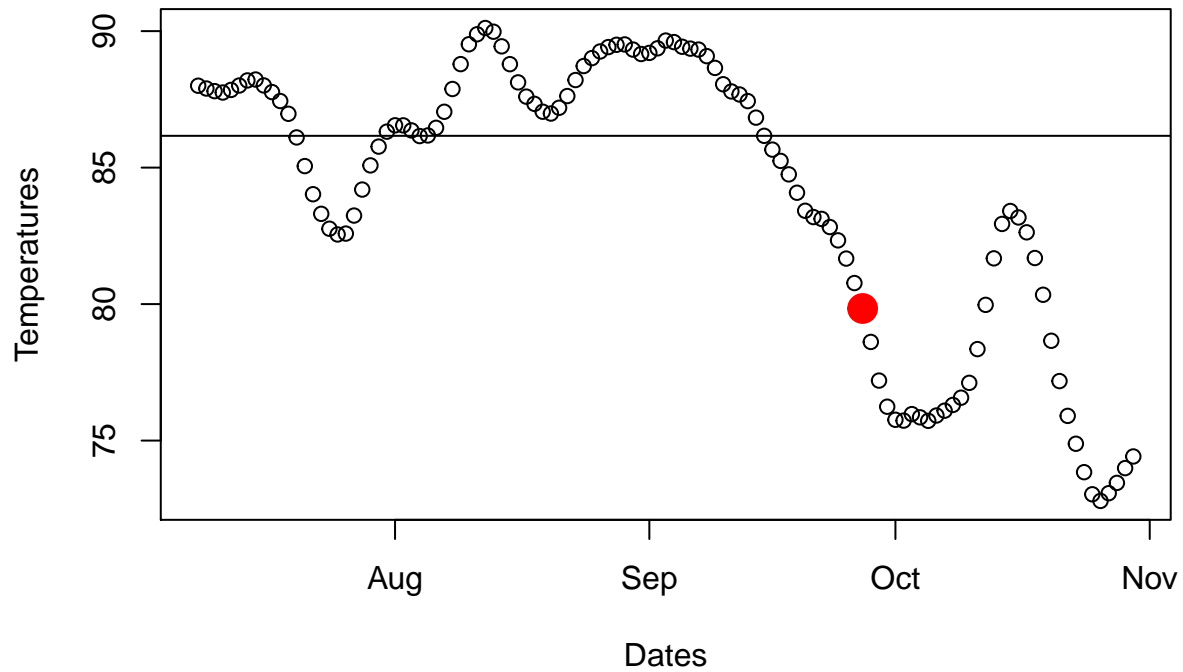
[1] "26-Sep"

Atlanta Temperatures in 2013



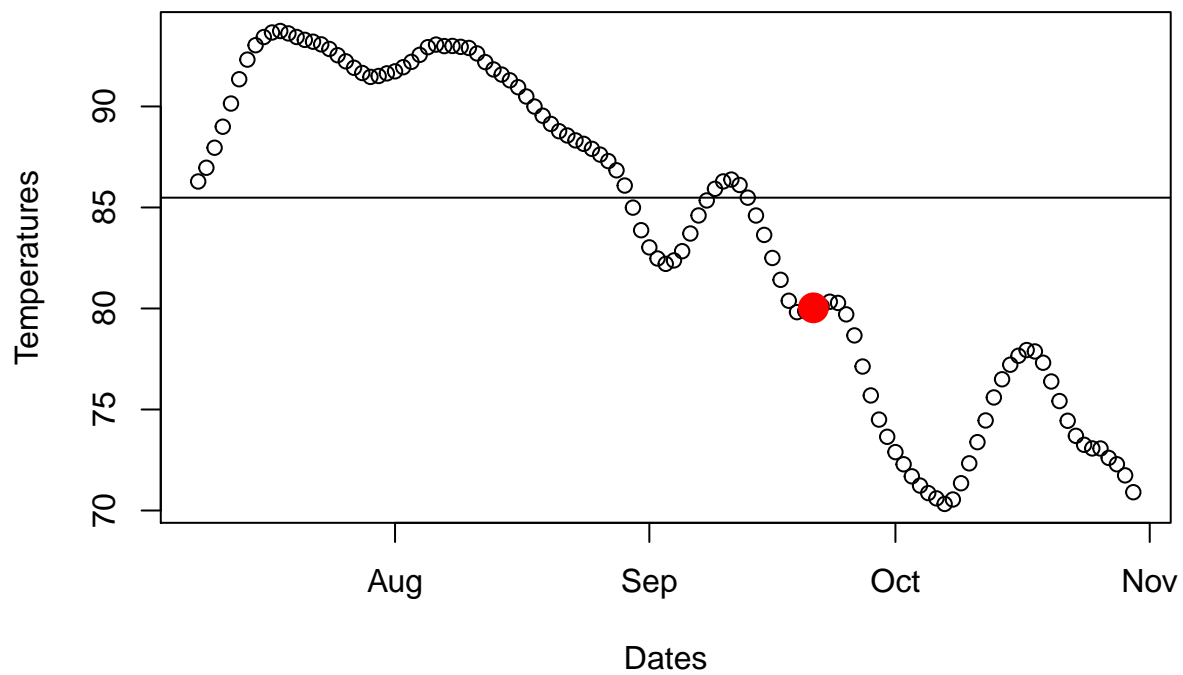
[1] "22-Aug"

Atlanta Temperatures in 2014



```
## [1] "26-Sep"
```

Atlanta Temperatures in 2015



```
## [1] "20-Sep"
```

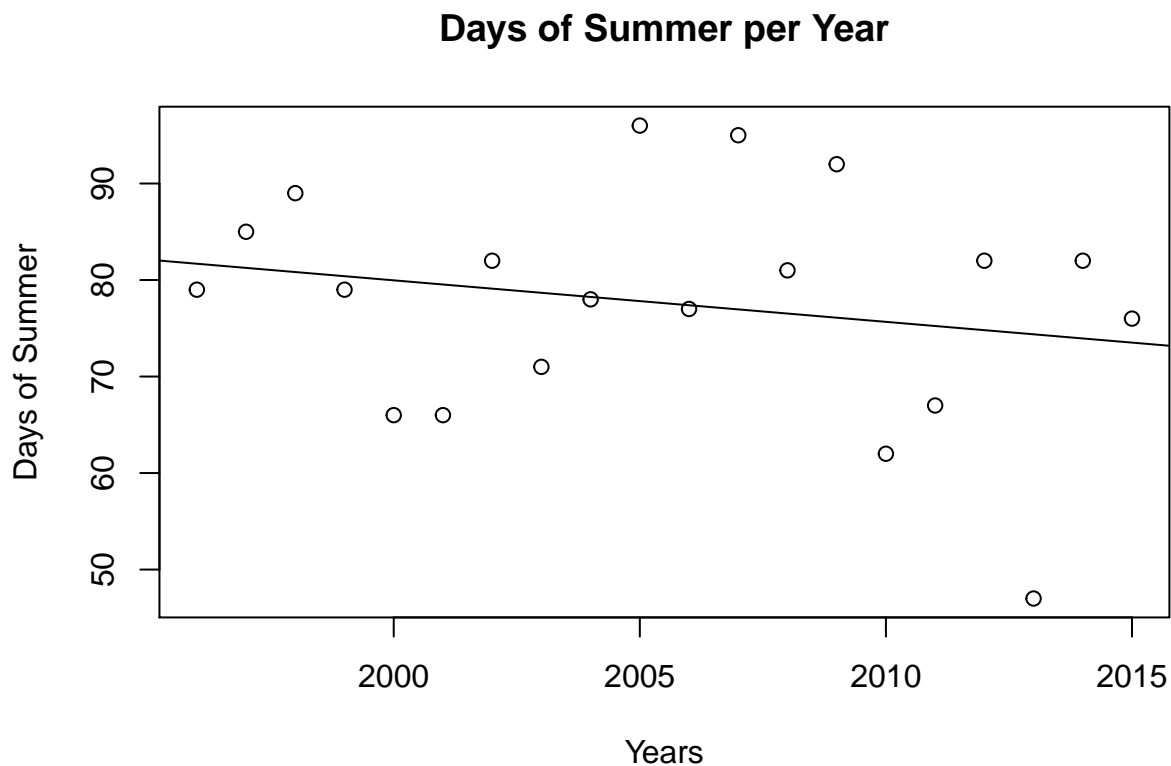
It appears that our algorithm works very well with the exception of 2013. There was an outlier of

a week for temperature in August where the temps dropped to nearly 75 degrees (moving average altered) before climbing to back over 90 degrees. It was hard to account for this in the cumsum algorithm. As such, the year will just be marked as an outlier for the algorithm. Let's now use the `weather_change_index` to extract for each year, all the days before that index as summer, and count how many days there are. If the unofficial end of summer is getting later, then with increasing years, we should see more days of summer.

```
days_of_summer = rep(0,20)
for (i in seq(1,20)){
  x = as.matrix(temps_ts[i,])
  x = x[1:weather_change_index[i]]
  days_of_summer[i] = length(x)
}

years = seq(1996, 2015)
data = c(years, days_of_summer)

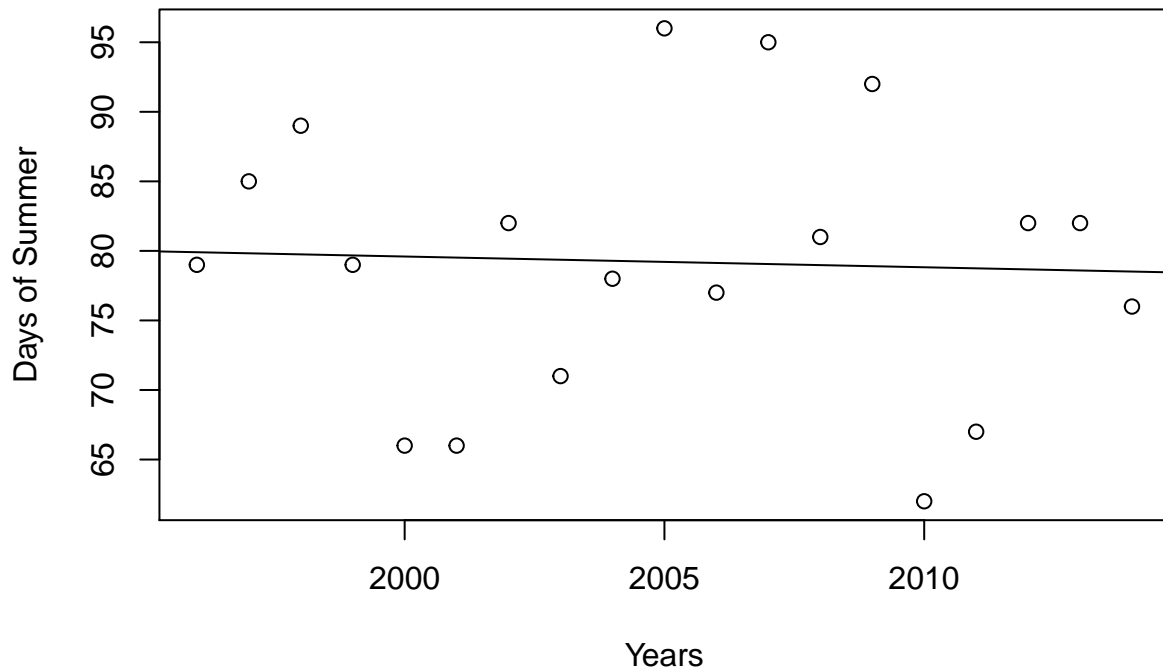
plot(x = years, y = days_of_summer, xlab='Years', ylab = 'Days of Summer', main = 'Days of Summer per Year')
abline(lm(formula = days_of_summer~years))
```



The outlier from 2013 appears to be dragging the regression down. Let's remove that outlier and replot the regression line

```
years = 1996:2014
days_of_summer = days_of_summer[days_of_summer != 47]
plot(x = years, y = days_of_summer, xlab='Years', ylab = 'Days of Summer', main = 'Days of Summer per Year')
abline(lm(formula = days_of_summer~years))
```

Days of Summer per Year



*Conclusion

It appears that the official end of summer is about consistent throughout the last 20 years. There is a little bit of a downwards trend, but it wouldn't be erroneous to interpret the trend line as flat. Notice how the days of summer per year are pretty random as well.