# Analysing Summer Temperatures in Atlanta

Juan Antonio Jiménez Cobo

## Introduction

In this project, we will use change detection methods and time series models to identify when unofficial summer ended using July through October daily-high-temperature data for Atlanta for 1996 through 2015. These results will serve us to explore further insights, such as if summer temperature has gotten warmer, or whether summer has ended later over the eyars.

The data set can be found using the link: [http://www.iweathernet.com/atlanta-weather-records](http://www.iweathernet.com/atlanta-weather-records).

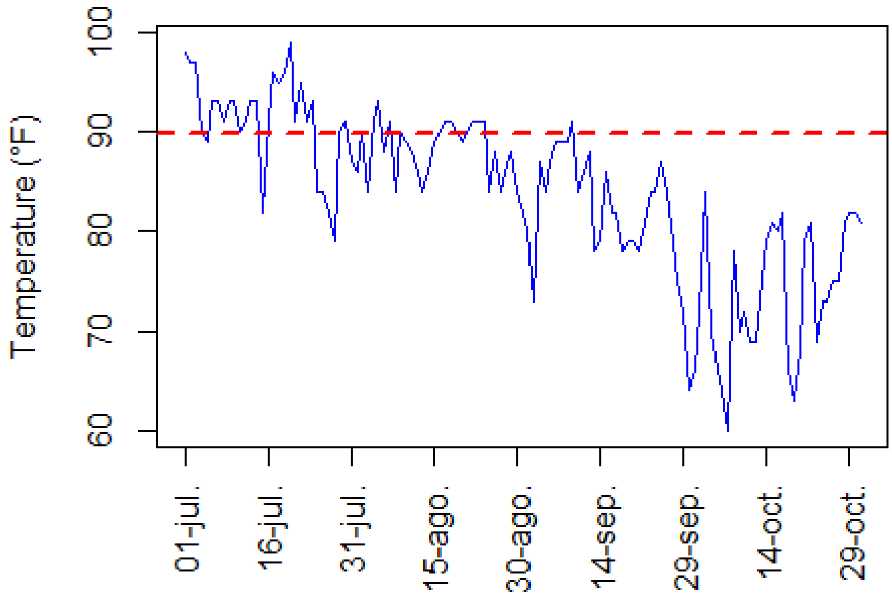
## 1. Import data set and initial exploration

```
# Clean the previous data
rm(list = ls())

# Import the new data
temperature_data <- read.table("temps.txt", header = TRUE)
head(temperature_data)

##     DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005
X2006 X2007
## 1 1-Jul    98    86    91    84    89    84    90    73    82    91
93    95
## 2 2-Jul    97    90    88    82    91    87    90    81    81    89
93    85
## 3 3-Jul    97    93    91    87    93    87    87    87    86    86
93    82
## 4 4-Jul    90    91    91    88    95    84    89    86    88    86
91    86
## 5 5-Jul    89    84    91    90    96    86    93    80    90    89
90    88
## 6 6-Jul    93    84    89    91    96    87    93    84    90    82
81    87
##   X2008 X2009 X2010 X2011 X2012 X2013 X2014 X2015
## 1    85    95    87    92   105    82    90    85
## 2    87    90    84    94    93    85    93    87
```

```
## 3    91    89    83    95    99    76    87    79
## 4    90    91    85    92    98    77    84    85
## 5    88    80    88    90   100    83    86    84
## 6    82    87    89    90    98    83    87    84
```

The data set consists of time-series data track the temperature in Atlanta every day from July 1st until October 31st from 1996 to 2015. First, we will analyze the temperatures recorded in 1996 by plotting the temperatures from every day in 1996 and compare them with the mean of the temperatures during that period to visualize the variation of the temperature during those months.

```
temp_1996 <- temperature_data$X1996
summer_temp_1996 <- temp_1996[1:61]

# Create a sequence of days from July 1st to October 31st
dates <- seq(as.Date("1996-07-01"), as.Date("1996-10-31"), by = "day")

# Mean of the temperature
mean_1996 <- mean(summer_temp_1996)
sd_1996 <- sd(temp_1996)

# Plot the data for 1996
plot(dates, temp_1996, type = "l", col = "blue",
     xlab = "",
     ylab = "Temperature (°F)",
     xaxt = "n")

# Add a red line for the mean temperature
abline(h = mean_1996, col = "red", lwd = 2, lty = 2)

# Show the dates at the horizontal axis in a range of 15 days
axis(1, at = seq(dates[1], dates[length(dates)], by = "15 days"),
     labels = format(seq(dates[1], dates[length(dates)], by = "15 days"),
"%d-%b"),
     las = 2)
```




As we can see in the graph, although there are some local minimums under the mean threshold before, for example at day 60 or around day 80, the temperature starts to fall consistently around day 80-90 (mid-to-late September).

## 2. CUSUM to detect summer-ending date

We now use a CUSUM approach to determine the date when summer ended each year. Since we have to determine when the temperatures decrease, we will subtract the mean minus the observed values $x_t$ at the formula of the cumulative sums. Then, the sequence of cumulative sums will be

$$S_t = \max\{0, S_{t-1} + (\mu - x_t - C)\}$$

and the date wanted shall be the date corresponding to the index $t$ such that $S_t \geq T$. Therefore, we have to set the parameters $C$ and $T$ such that the results are close to those deduced previously. A general threshold to use is

$$C = \frac{\sigma}{2}, \quad T = 5\sigma$$

where $\sigma$ denotes the standard deviation of the temperatures. We will use these parameters to compute the cumulative sum sequence. We will start with 1996.

```
# Define a function that construct the sequence of cumulative sums
cusum <- function(temp_data, mu, C){
```

```
  # Initialize the sequence of cumulative sums and set the first term of
the sequence
  s <- numeric(length(temp_data))
  s[1] <- max(0, mu-temp_data[1]-C)
  # Set the rest of the terms
  for (t in 2:length(s)){
    s[t] <- max(0, s[t-1]+(mu-temp_data[t]-C))
  }
  # Return the sequence of cumulative sums
  return(s)
}
C <- sd_1996/2
T <- sd_1996*5

# Compute the CUSUM sequence for the year 1996 using C=2
cusum_1996 <- cusum(temp_1996, mean_1996, C)
cusum_1996

##   [1]   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
0.0000000
##   [7]   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
0.0000000
##  [13]   0.0000000   0.0000000   3.4635353   0.0000000   0.0000000
0.0000000
##  [19]   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
0.0000000
##  [25]   1.4635353   2.9270706   6.3906059  12.8541412   8.3176765
2.7812118
##  [31]   1.2447471   0.7082824   0.0000000   1.4635353   0.0000000
0.0000000
##  [37]   0.0000000   0.0000000   1.4635353   0.0000000   0.0000000
0.0000000
##  [43]   0.0000000   1.4635353   0.9270706   0.0000000   0.0000000
0.0000000
##  [49]   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
0.0000000
##  [55]   0.0000000   1.4635353   0.0000000   1.4635353   0.9270706
0.0000000
##  [61]   1.4635353   4.9270706  10.3906059  22.8541412  21.3176765
22.7812118
##  [67]  21.2447471  17.7082824  14.1718177  10.6353530   5.0988883
6.5624236
##  [73]   6.0259589   3.4894942  10.9530295  17.4165648  16.8801001
20.3436354
##  [79]  23.8071707  31.2707060  37.7342413  44.1977766  51.6613119
56.1248472
##  [85]  57.5883825  59.0519178  57.5154531  58.9789884  65.4425237
75.9060590
##  [91]  89.3695943 110.8331296 130.2966649 143.7602002 145.2237355
160.6872708
```

```
##  [97] 180.1508061 201.6143414 227.0778767 234.5414120 250.0049473
263.4684826
## [103] 279.9320179 296.3955532 308.8590885 315.3226238 319.7861591
325.2496944
## [109] 328.7132297 348.1767650 370.6403003 388.1038356 394.5673709
399.0309062
## [115] 415.4944415 427.9579768 440.4215121 450.8850474 461.3485827
465.8121180
## [121] 469.2756533 472.7391886 477.2027239

# Define a function that, given a CUSUM sequence, returns the index of
the first term that is greater than the threshold
summer_end <- function(cusum_seq, T){
  t <- 1
  while (cusum_seq[t]<T && t <= length(cusum_seq)){
    t <- t+1
  }
  return(t)
}

# Set the value of the temperatures of 1996 that surpass the threshold
index_1996 <- summer_end(cusum_1996, T)

# Set the date and the temperature recorded on that day
summer_end_1996 <- temperature_data$DAY[index_1996]
temp_summer_end_1996 <- temp_1996[index_1996]

# Print the results
cat("The summer in 1996 ended on", summer_end_1996, " and the temperature
recorded on that day was", temp_summer_end_1996)

## The summer in 1996 ended on 20-Sep  and the temperature recorded on
that day was 79
```

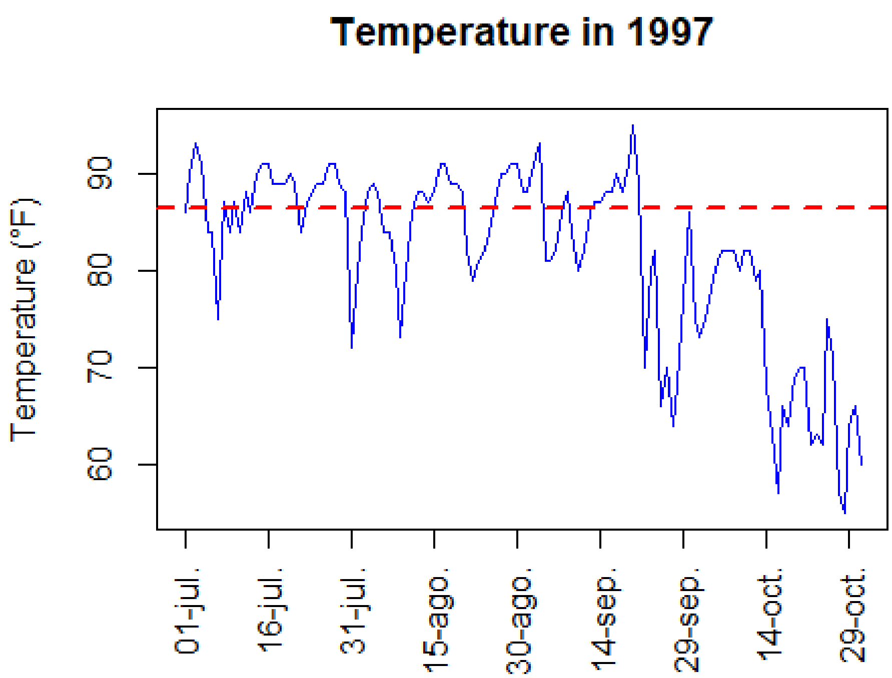
The method detected that temperatures decreased significantly from September 20th, which aligns with our previous insights.

Next, we will do the same for the rest of the years. We will build a function to plot a comparison between the cumulative sums and the mean of each year, and then detect when the decrease in temperatures became significant.

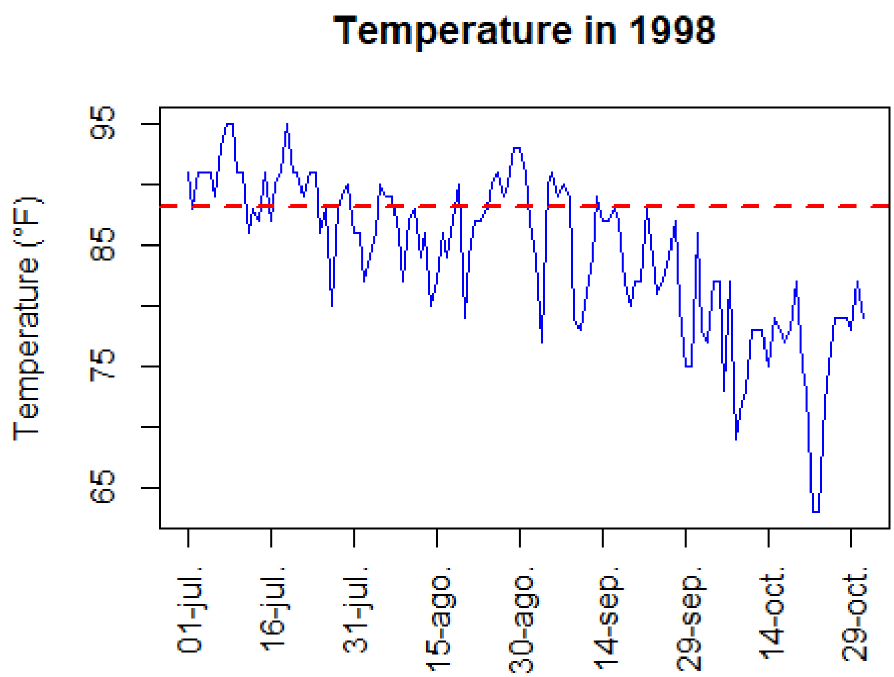
```
process_years_recursive <- function(years, results = list()) {
  # base case: no years left
  if (length(years) == 0) {
    # combine results list into a data.frame and return
    if (length(results) == 0) return(data.frame())
    return(do.call(rbind, results))
  }

  # process the first year in the vector
  year <- years[1]
```

```r
  colname <- paste0("X", year)

  # If the column doesn't exist, add NA row and recurse
  if (!colname %in% names(temperature_data)) {
    warning("Column ", colname, " not found. Skipping ", year)
    row <- data.frame(
      year = year,
      summer_end = NA,
      temp_on_summer_end = NA,
      C = NA,
      T = NA,
      stringsAsFactors = FALSE
    )
    return(process_years_recursive(years[-1], c(results, list(row))))
  }

  temp_year <- temperature_data[[colname]]
  mean_year <- mean(temp_year[1:61])
  sd_year <- sd(temp_year)
  C <- sd_year / 2
  T <- sd_year * 5

  plot(dates, temp_year, type = "l", col = "blue",
       xlab = "",
       ylab = "Temperature (°F)",
       xaxt = "n",
       main = paste("Temperature in", year))

  # Add a red line for the mean temperature
  abline(h = mean_year, col = "red", lwd = 2, lty = 2)

  # Show the dates at the horizontal axis in a range of 15 days
  axis(1, at = seq(dates[1], dates[length(dates)], by = "15 days"),
       labels = format(seq(dates[1], dates[length(dates)], by = "15
days"), "%d-%b"),
       las = 2)

  cusum_year <- cusum(temp_year, mean_year, C)
  index_year <- summer_end(cusum_year, T)

  summer_end_date <- temperature_data$DAY[index_year]
  temp_summer_end <- temp_year[index_year]

  # Print the results exactly like in your original code
  cat("The summer in", year, "ended on", summer_end_date,
      " and the temperature recorded on that day was", temp_summer_end,
      "\nC=", C, ",  T=", T, "\n")

  # Build a row of results (keeps same information as your original)
```

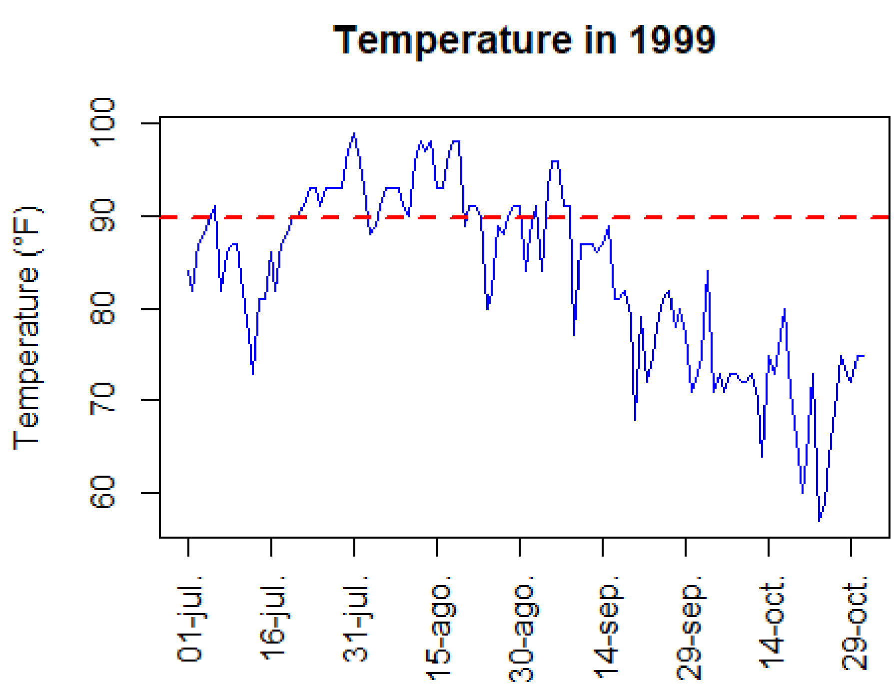
```r
  row <- data.frame(
    year = year,
    summer_end = ifelse(length(index_year) >= 1,
as.character(summer_end_date[1]), NA),
    temp_on_summer_end = ifelse(length(index_year) >= 1,
as.numeric(temp_summer_end[1]), NA),
    C = C,
    T = T,
    stringsAsFactors = FALSE
  )

  # recurse on remaining years, appending this year's row to results
  process_years_recursive(years[-1], c(results, list(row)))
}

# Call the recursive function for years 1997 through 2015
results_df <- process_years_recursive(1997:2015)
```



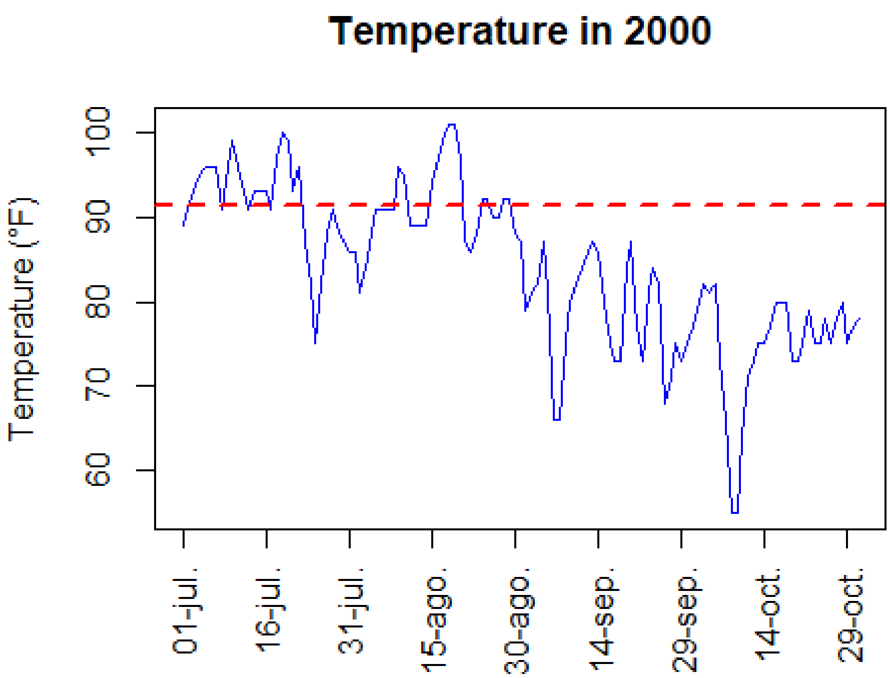

```
## The summer in 1997 ended on 27-Sep  and the temperature recorded on
that day was 64
## C= 4.659512 ,  T= 46.59512
```



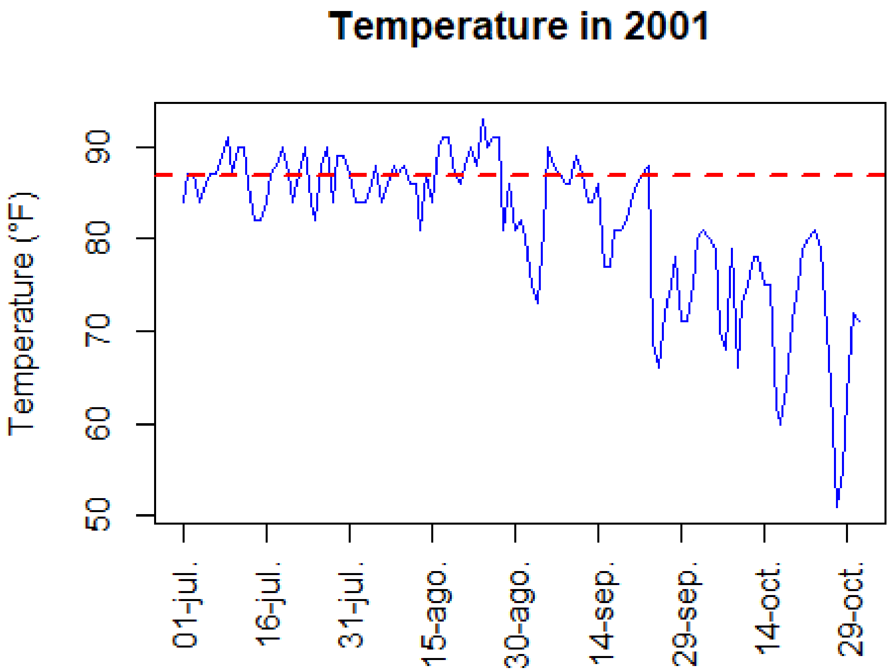

```
## The summer in 1998 ended on 29-Sep  and the temperature recorded on
that day was 75
## C= 3.204657 ,  T= 32.04657
```



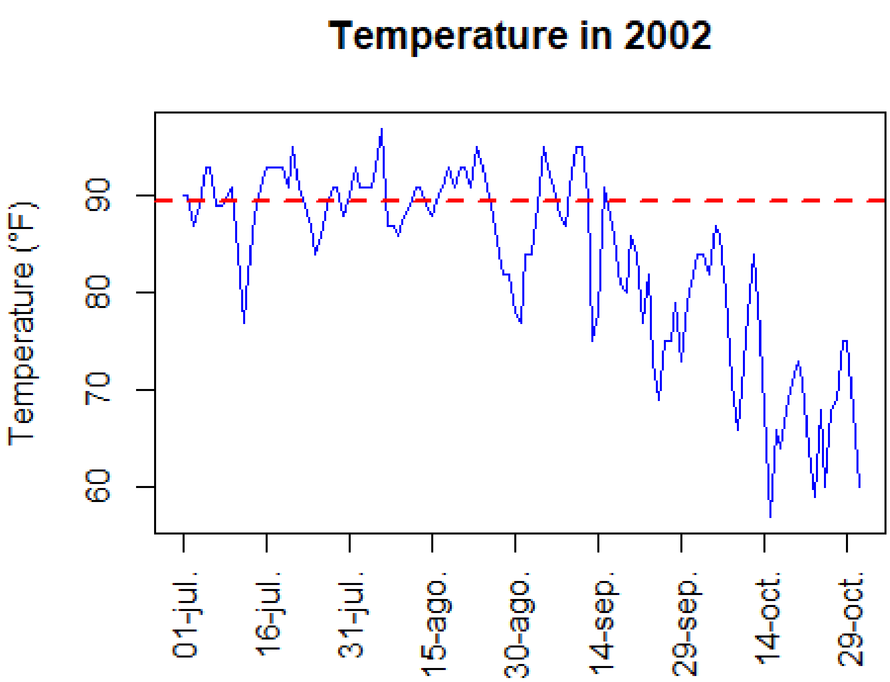
```
## The summer in 1999 ended on 22-Sep  and the temperature recorded on that day was 72
## C= 4.861664 ,  T= 48.61664
```



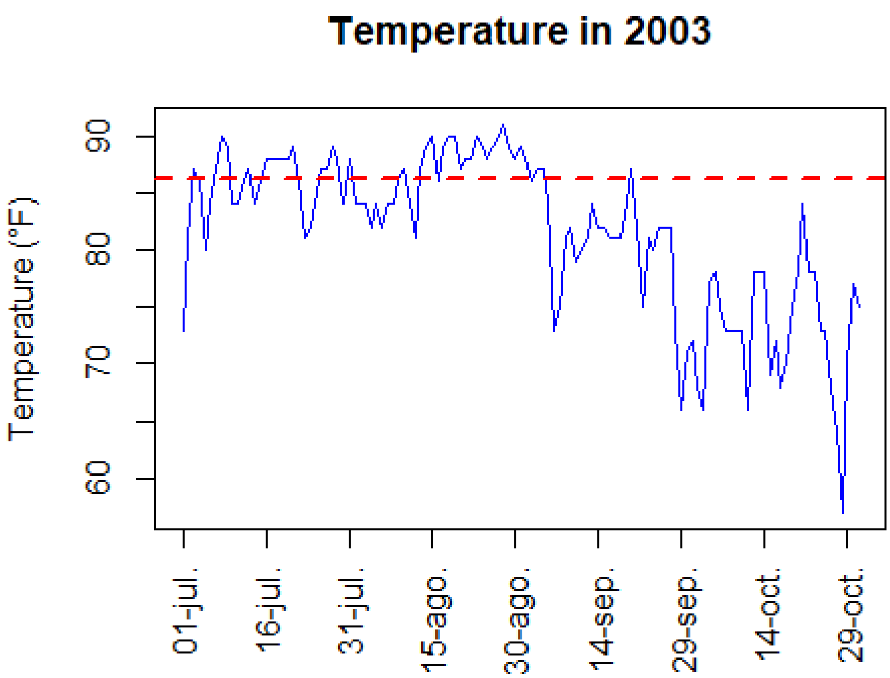

```
## The summer in 2000 ended on 7-Sep  and the temperature recorded on that day was 66
## C= 4.759346 ,  T= 47.59346
```



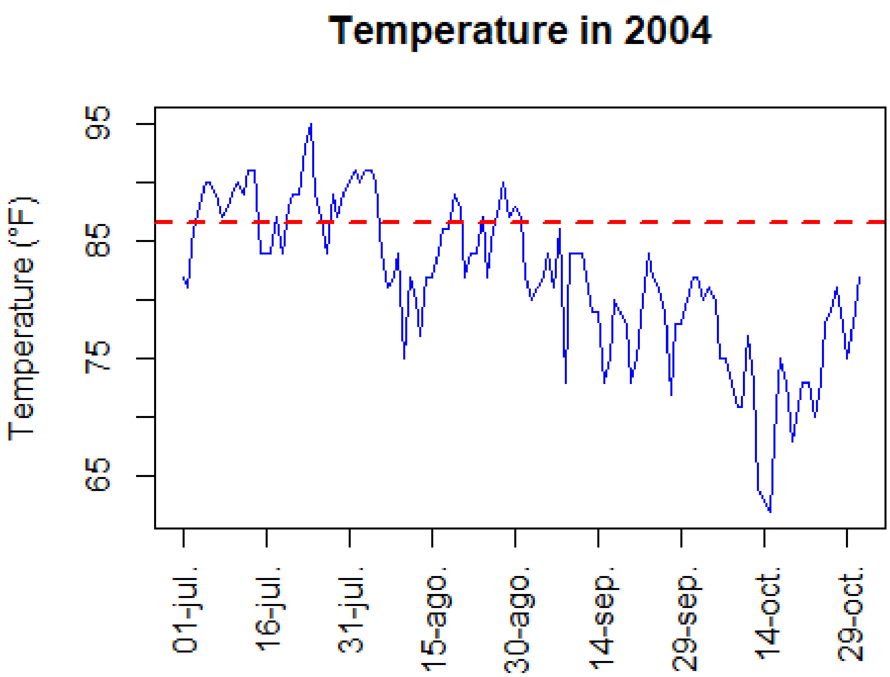

```
## The summer in 2001 ended on 26-Sep  and the temperature recorded on
that day was 72
## C= 4.112258 ,  T= 41.12258
```



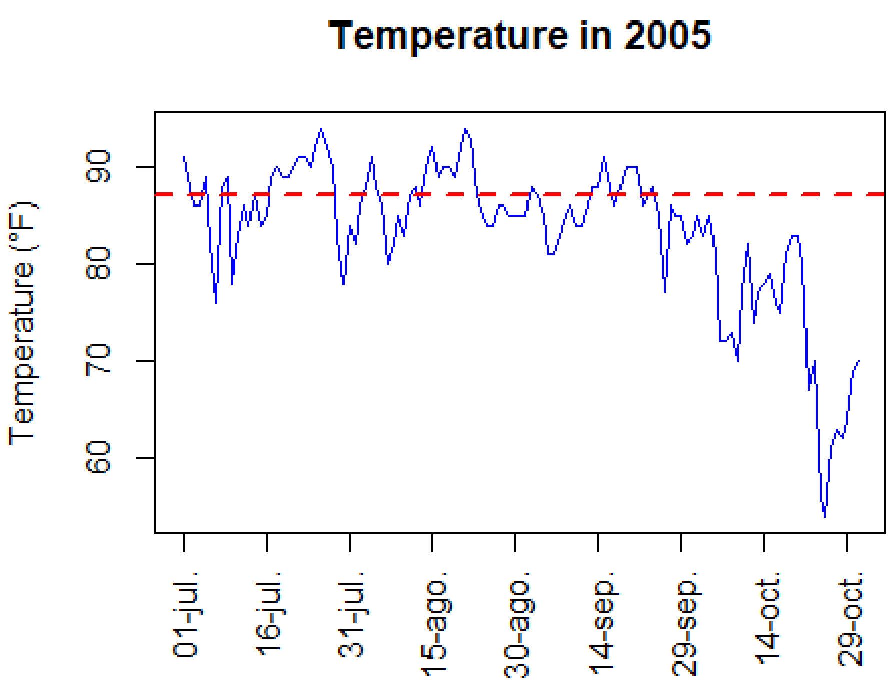
```
## The summer in 2002 ended on 25-Sep  and the temperature recorded on that day was 69
## C= 4.713047 ,  T= 47.13047
```



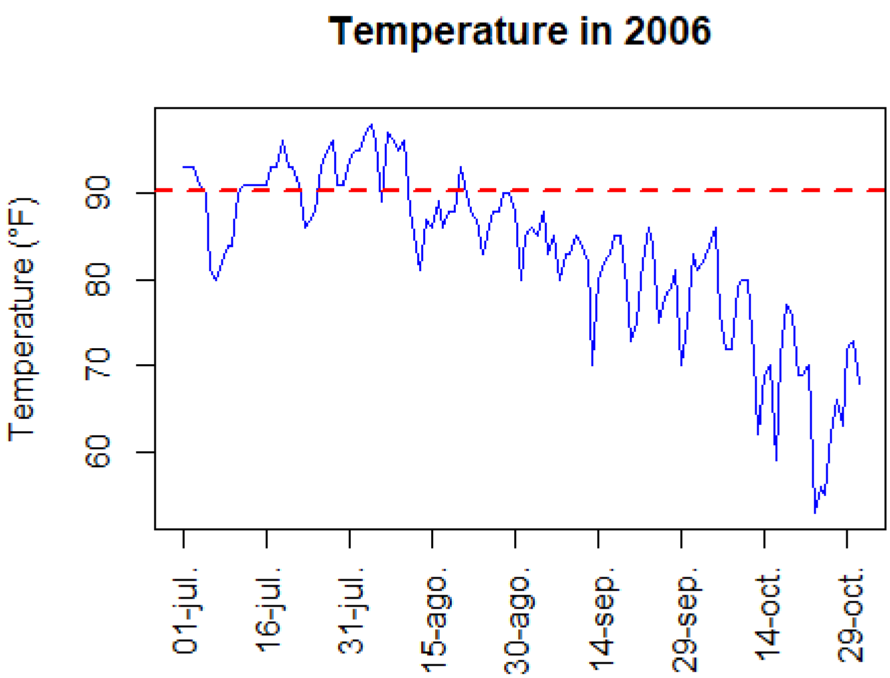

```
## The summer in 2003 ended on 22-Sep  and the temperature recorded on that day was 75
## C= 3.508975 ,  T= 35.08975
```




```
## The summer in 2004 ended on 15-Sep  and the temperature recorded on that day was 73
## C= 3.33147 ,  T= 33.3147
```

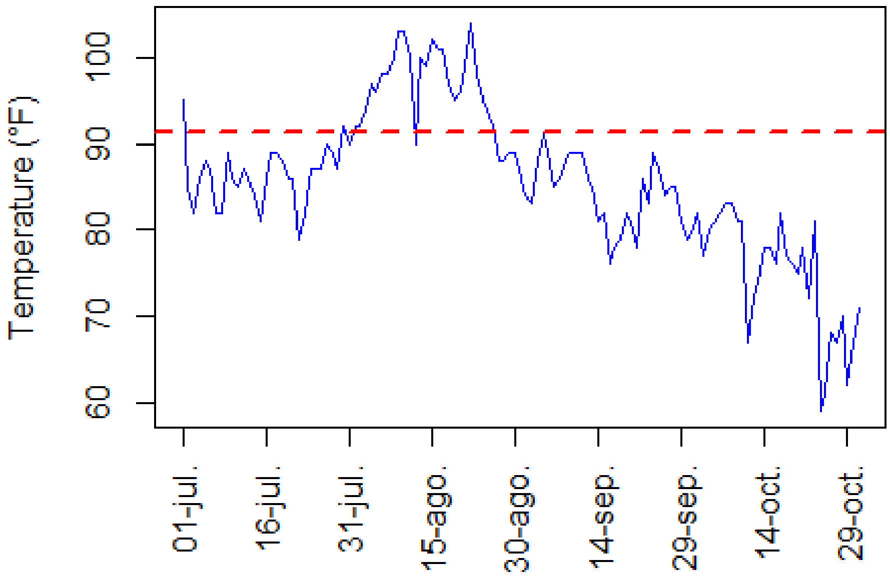


```
## The summer in 2005 ended on 9-Oct  and the temperature recorded on
that day was 70
## C= 3.866698 ,  T= 38.66698
```



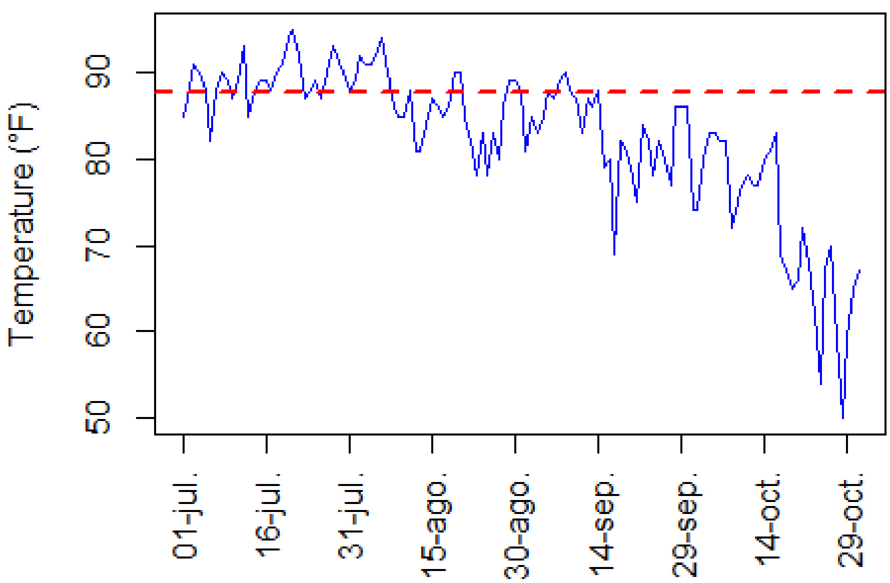

```
## The summer in 2006 ended on 19-Sep  and the temperature recorded on
that day was 79
## C= 4.896826 ,  T= 48.96826
```

**Temperature in 2007**

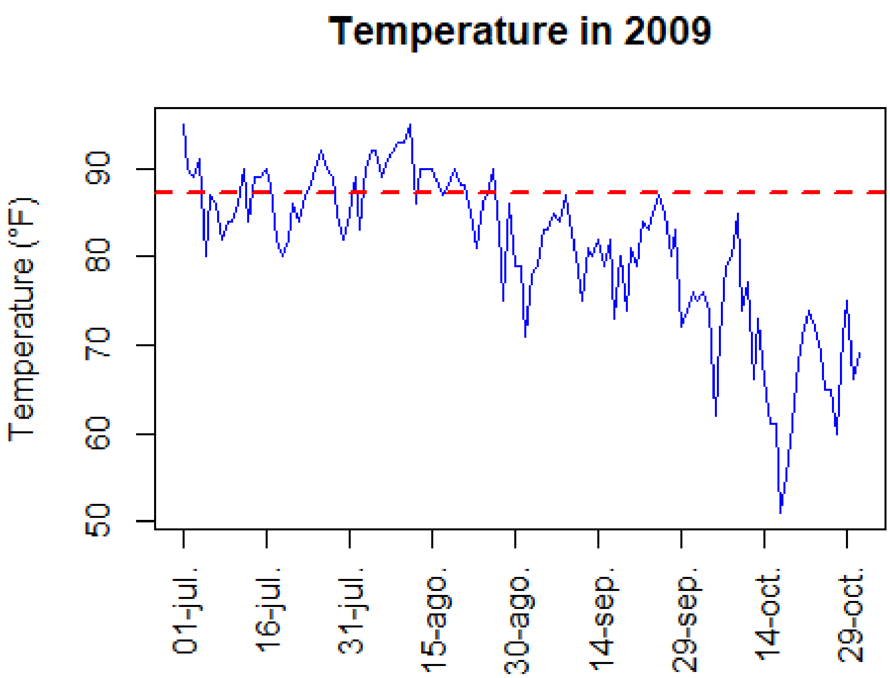



```
## The summer in 2007 ended on 19-Sep  and the temperature recorded on
that day was 82
## C= 4.516699 ,  T= 45.16699
```

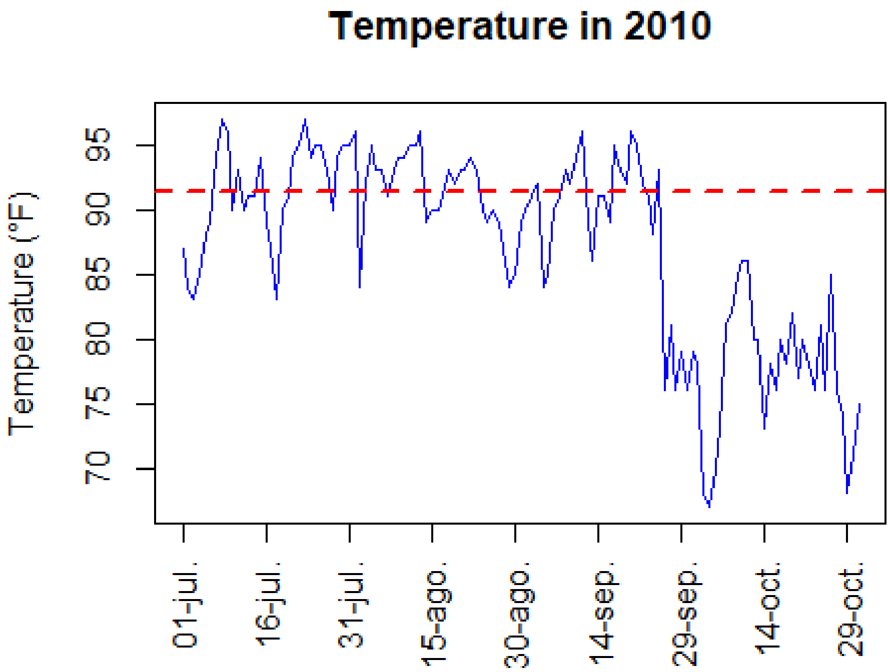
**Temperature in 2008**



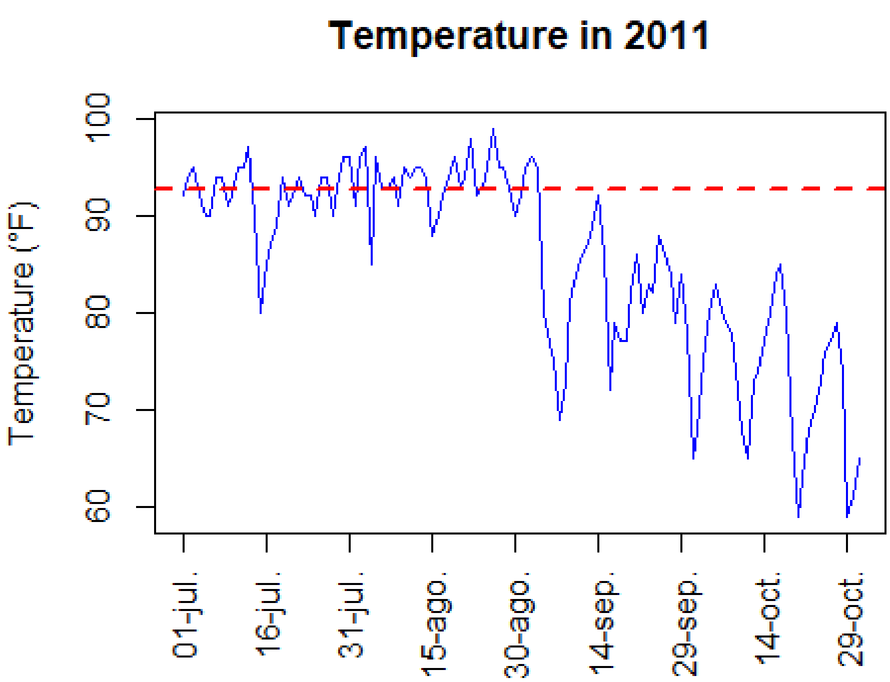
```
## The summer in 2008 ended on 24-Sep  and the temperature recorded on
that day was 78
## C= 4.366586 ,  T= 43.66586
```



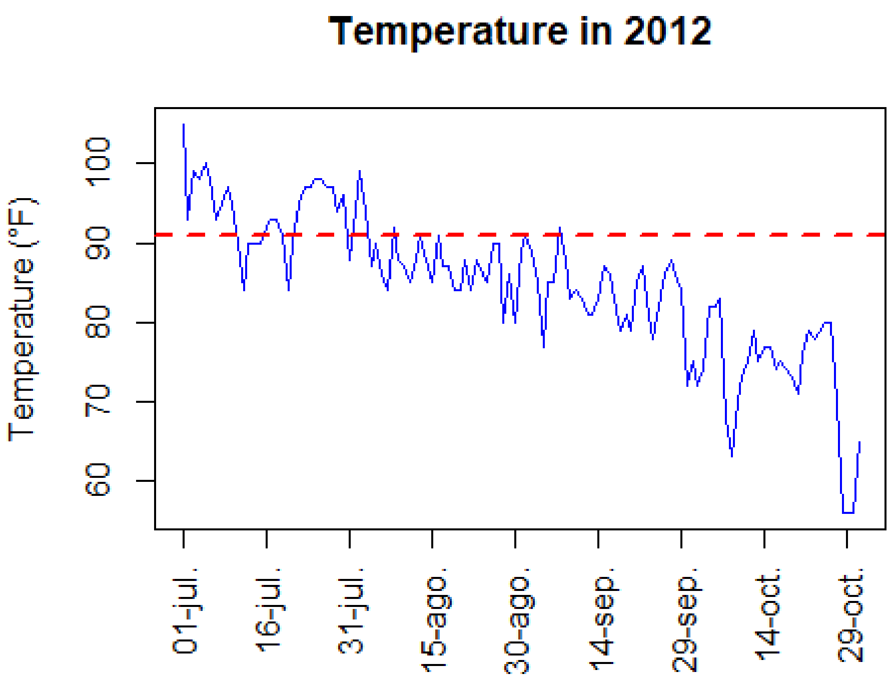

```
## The summer in 2009 ended on 17-Sep  and the temperature recorded on
that day was 73
## C= 4.506596 ,  T= 45.06596
```



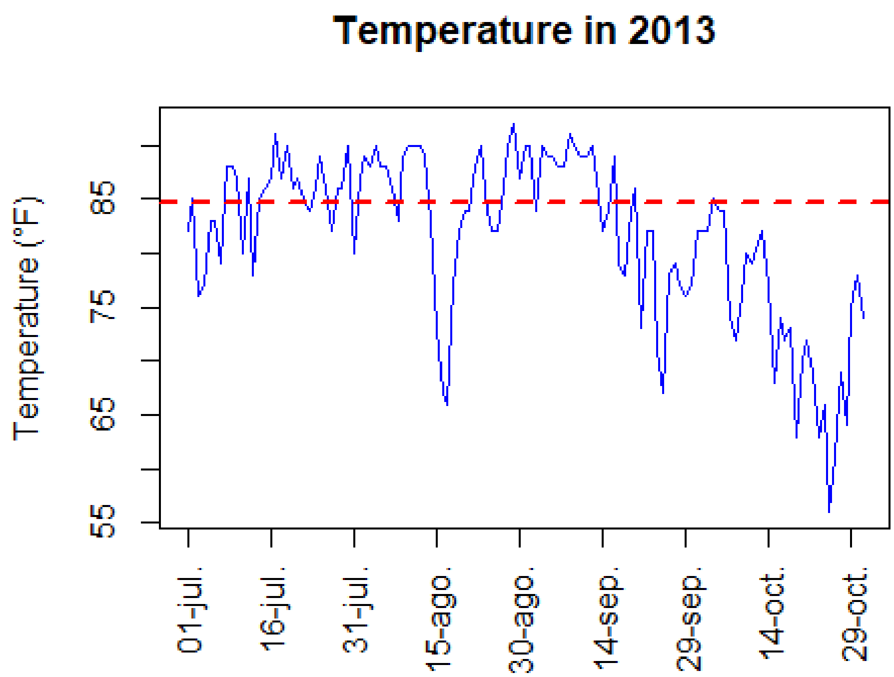

```
## The summer in 2010 ended on 29-Sep  and the temperature recorded on that day was 79
## C= 3.722578 ,  T= 37.22578
```



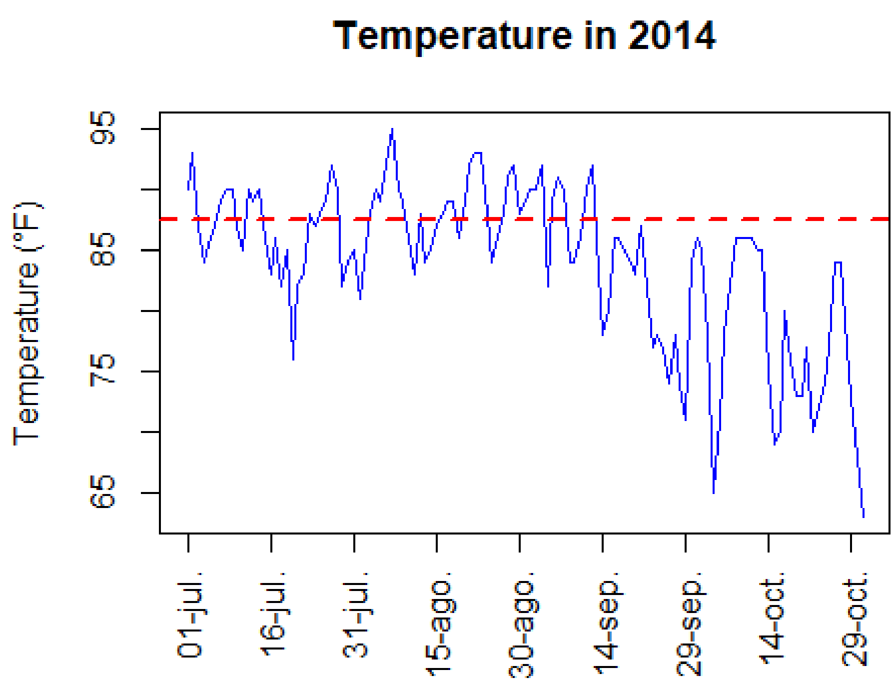
```
## The summer in 2011 ended on 8-Sep  and the temperature recorded on that day was 73
## C= 4.965579 ,  T= 49.65579
```



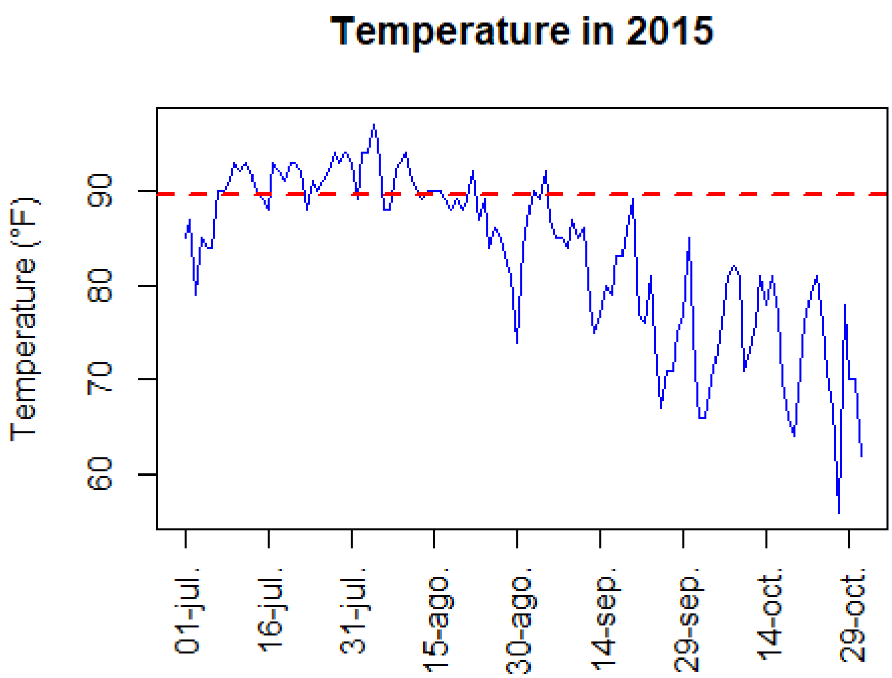

```
## The summer in 2012 ended on 19-Sep  and the temperature recorded on that day was 81
## C= 4.626183 ,  T= 46.26183
```




```
## The summer in 2013 ended on 18-Aug  and the temperature recorded on
that day was 77
## C= 3.863271 ,  T= 38.63271
```



```
## The summer in 2014 ended on 26-Sep  and the temperature recorded on that day was 74
## C= 3.295738 ,  T= 32.95738
```




```
## The summer in 2015 ended on 22-Sep  and the temperature recorded on that day was 76
## C= 4.354635 ,  T= 43.54635

# View the combined results
print(results_df)

##    year summer_end temp_on_summer_end        C        T
## 1  1997     27-Sep                 64 4.659512 46.59512
## 2  1998     29-Sep                 75 3.204657 32.04657
## 3  1999     22-Sep                 72 4.861664 48.61664
## 4  2000      7-Sep                 66 4.759346 47.59346
## 5  2001     26-Sep                 72 4.112258 41.12258
## 6  2002     25-Sep                 69 4.713047 47.13047
## 7  2003     22-Sep                 75 3.508975 35.08975
## 8  2004     15-Sep                 73 3.331470 33.31470
## 9  2005      9-Oct                 70 3.866698 38.66698
## 10 2006     19-Sep                 79 4.896826 48.96826
## 11 2007     19-Sep                 82 4.516699 45.16699
## 12 2008     24-Sep                 78 4.366586 43.66586
## 13 2009     17-Sep                 73 4.506596 45.06596
## 14 2010     29-Sep                 79 3.722578 37.22578
## 15 2011      8-Sep                 73 4.965579 49.65579
```

```
## 16 2012     19-Sep                       81 4.626183 46.26183
## 17 2013     18-Aug                       77 3.863271 38.63271
## 18 2014     26-Sep                       74 3.295738 32.95738
## 19 2015     22-Sep                       76 4.354635 43.54635
```

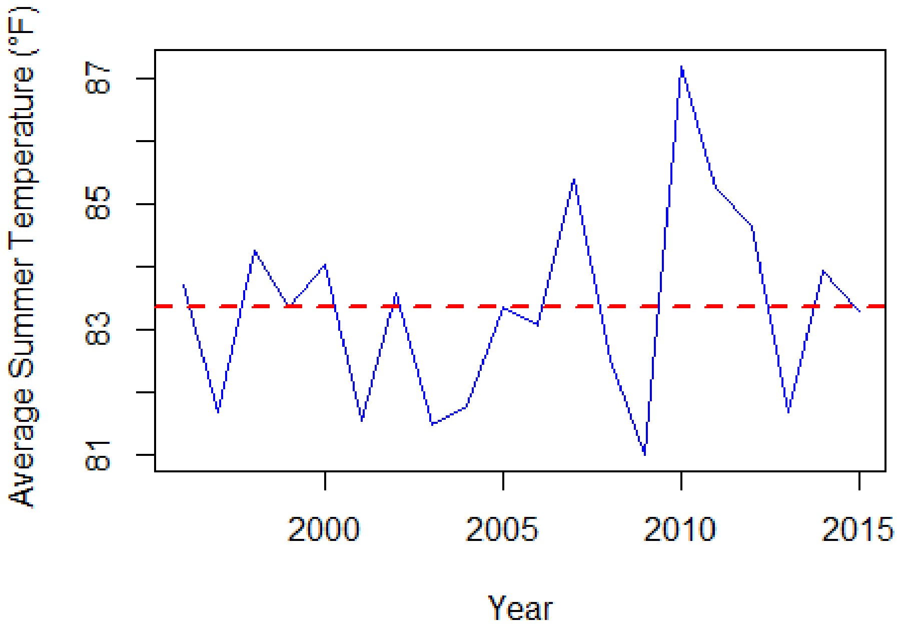
In the data frame we can observe the summer ending date from 1997 to 2015, the temperature of that day, and the values of the parameters $C$ and $T$ that were used in the CUSUM method.

**Note:** Although CUSUM is a valid technique for change detection, it has some limitations that are evidenced during this project. One of them is its sensibility to outliers or extreme values. If we look at the year 2013, we will notice that summer seems to end around mid-September. However, the extreme decrease of temperatures around mid-August made CUSUM to incorrectly detect that summer ended on August 18th. We will explore approaches to address this issue later on.

# 3. Determine whether the summer temperature in Atlanta has gotten warmer

Once we have identified the date of summer end from 1996 to 2015, we can analyze whether the temperature of Atlanta in summer has gotten warmer over the years. To this end, we will implement a CUSUM approach for the average temperature of each year, and detect whether and when there was a significant variation.

```r
# Define the range of years
years <- 1996:2015

# Calculate the mean temperature for each year and store in a list
mean_temps <- sapply(years, function(y) {
  colname <- paste0("X", y)
  mean(temperature_data[[colname]], na.rm = TRUE)
})

# Mean of the average temperatures and vector of years
temp_mean <- mean(mean_temps)

plot(years, mean_temps, type = "l", col = "blue",
     xlab = "Year", ylab = " Average Summer Temperature (°F)")

# Add a horizontal red line for the mean temperature
abline(h = temp_mean, col = "red", lwd = 2, lty = 2)
```




We can observe natural variation in the average temperatures, likely influenced by the natural climate changes, with 2010 being the year with the highest average temperature. However, no clear trend can be appreciated in the plot, although it might be discussed that the variance has slightly increased over the years.

To obtain a more objective conclusion, we will apply the CUSUM model to determine when the average temperatures have increased by adjusting the threshold $T$. Since the focus is on detecting increasing temperatures, the formula for the cumulative sums will be:

$$S_t = \max\{0, S_{t-1} + (x_t - \mu - C)\}$$

```
# Standard deviation of the temperatures
sd <- sd(mean_temps)

# Define CUSUM parameters
C <- sd/2
T <- 5*sd

# Define a function that construct the new sequence of cumulative sums
cusum2 <- function(temp_data, mu, C){
  # Initialize the sequence of cumulative sums and set the first term of
the sequence
  s <- numeric(length(temp_data))
  s[1] <- max(0, temp_data[1]-mu-C)
  # Set the rest of the terms
  for (t in 2:length(s)){
```

```
    s[t] <- max(0, s[t-1]+(temp_data[t]-mu-C))
  }
  # Return the sequence of cumulative sums
  return(s)
}

# Set the sequence of cumulative sums
cusum_years <- cusum2(mean_temps, temp_mean, C)
print(cusum_years)

##  [1] 0.0000000 0.0000000 0.1299097 0.0000000 0.0000000 0.0000000
0.0000000
##  [8] 0.0000000 0.0000000 0.0000000 0.0000000 1.2681211 0.0000000
0.0000000
## [15] 3.0811292 4.2272991 4.7474527 2.2838665 2.0967031 1.2672632

# Set the value of the temperatures of each year that surpass the
threshold
index <- summer_end(cusum_years, T)

# Set the date and the temperature recorded on that year
warmer_year <- 1995+index
warmer_temp <- mean_temps[index]

# Print the results
cat("The summer climate in Atlanta has gotten warmer in", warmer_year,
"and its average temperature in summer was", warmer_temp)

## The summer climate in Atlanta has gotten warmer in 2016 and its
average temperature in summer was NA
```

The output of the model indicates that no cumulative sum exceeded the threshold of the model, confirming that there is no significant evidence to conclude that the summer in Atlanta has gotten warmer over the years during the period 1996-2015.

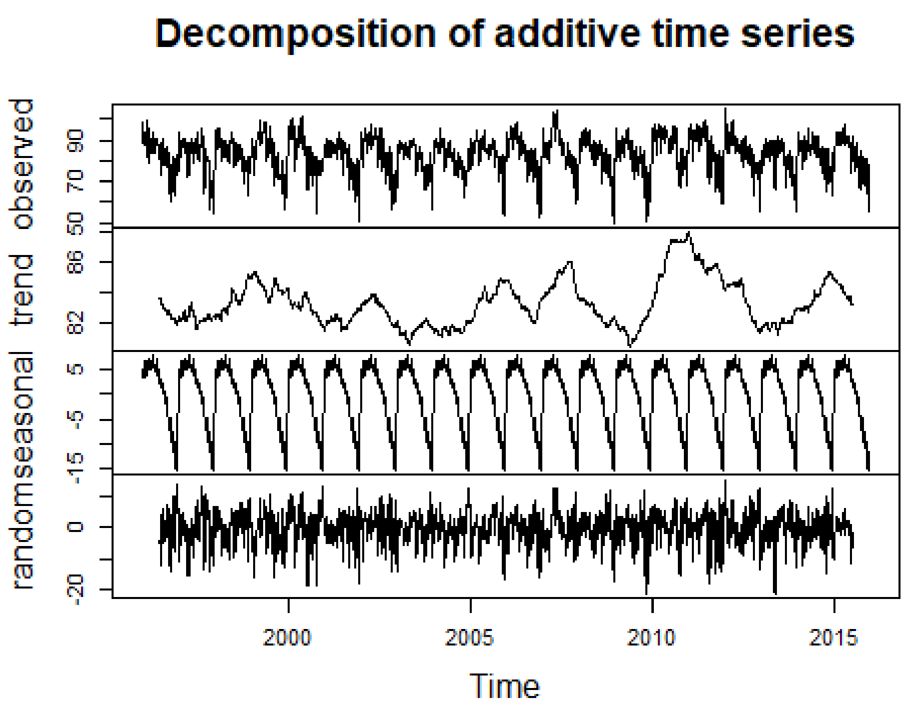
# 4. Applying Exponential Smoothing

```
# Load packages
library(smooth)
```

As we saw previously, CUSUM had some limitations when dealing with sudden extreme values, as it happened when detecting the date in 2013. One possible approach is to employ Exponential Smoothing (ES) to reduce the noise within data. Before we run the model, we have to transform the data into time series format, and decompose the data to determine the best approach to analyze it.

```
# Convert to time series
days_per_year <- nrow(temperature_data)
```

```r
# Convert to long format
temp_long <- data.frame(
  Day = rep(1:days_per_year, length(years)),
  Year = rep(years, each = days_per_year),
  Temp = as.vector(as.matrix(temperature_data[, -1])) # Convert to
numeric vector
)

# Convert data to time series data
temp_ts <- ts(temp_long$Temp, start = c(1996, 1), frequency =
days_per_year)

temp_components <- decompose(temp_ts)
plot(temp_components)
```




The decomposition provides two insights about the time series components:

1. There is no clear increasing/decreasing trend within the data, although the variance seems to have increased over the years
2. A seasonal pattern can be clearly appreciated, which is expected since we are considering annual temperatures.

Since there is no clear trend over the years, the best possible approach is to employ an additive Exponential Smoothing model.

```r
# Apply HoltWinters Exponential Smoothing
hw_model <- HoltWinters(temp_ts, seasonal = "additive")
hw_model
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal
component.
##
## Call:
## HoltWinters(x = temp_ts, 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
```

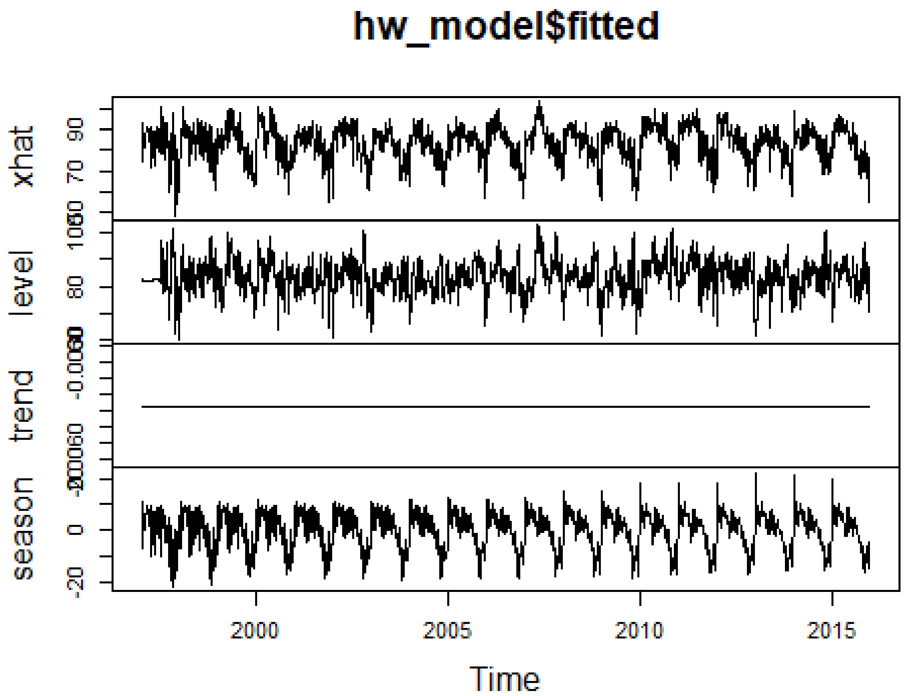
```
## s32    4.931885957
## s33    7.109879628
## s34    6.178469084
## s35    4.886891317
## s36    3.890547248
## s37    2.148316257
## s38    2.524866001
## s39    3.008098232
## s40    3.041663870
## s41    2.251741386
## s42    0.101091985
## s43   -0.123337548
## s44   -1.445675315
## s45   -1.802768181
## 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
```

```
hw_model$SSE
```

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

```
plot(hw_model$fitted)
```

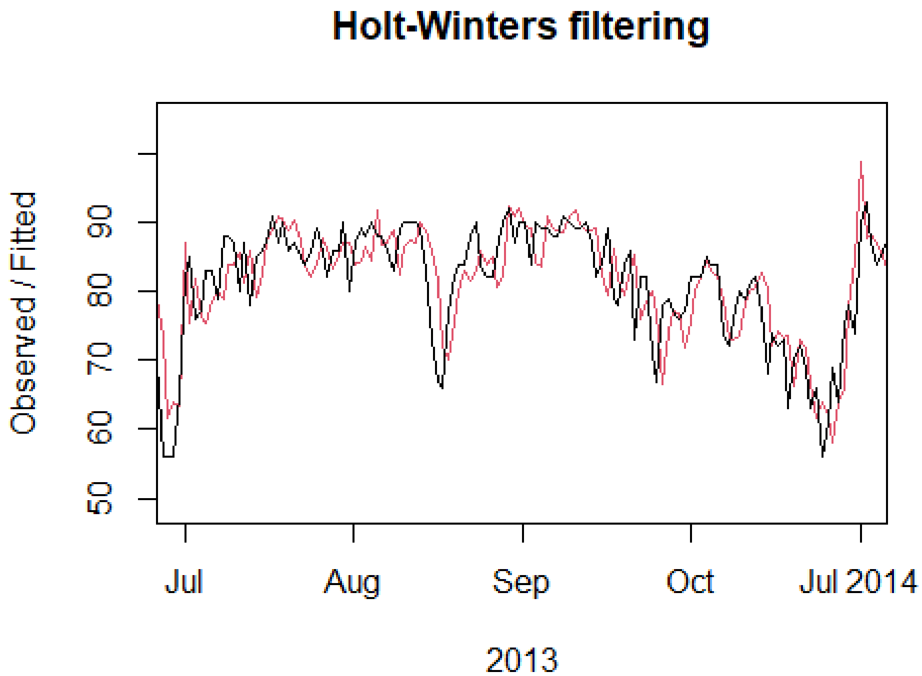



The smoothing parameters of the model offers some insights:

- $alpha$: represents the proportion of the new estimate that comes from the previous estimate, and how much comes from previous estimates. Since its value is closer to 1, it means that the model gives more importance to recent observations.
- $beta$: its value is 0 since we are using an additive model with no trend.
- $gamma$: determines how the seasonal pattern is updated based on previous observations. Its value of 0.625 allows the model to quickly adapt to the seasonal changes caused by the end of summer.

Once we have fitted the ES model, we analyze how much the extreme values have been smoothed. For example, let’s compare the original temperatures of 2013 with the smoothed values after applying the ES model.

```
# Plot the data for 2013 with the fitted values
plot(hw_model, xlab = "2013", xlim = c(2013, 2014), xaxt= "n")
axis(1, at = seq(2013, 2014, by = 0.25), labels = c("Jul", "Aug", "Sep",
"Oct", "Jul 2014"))
```



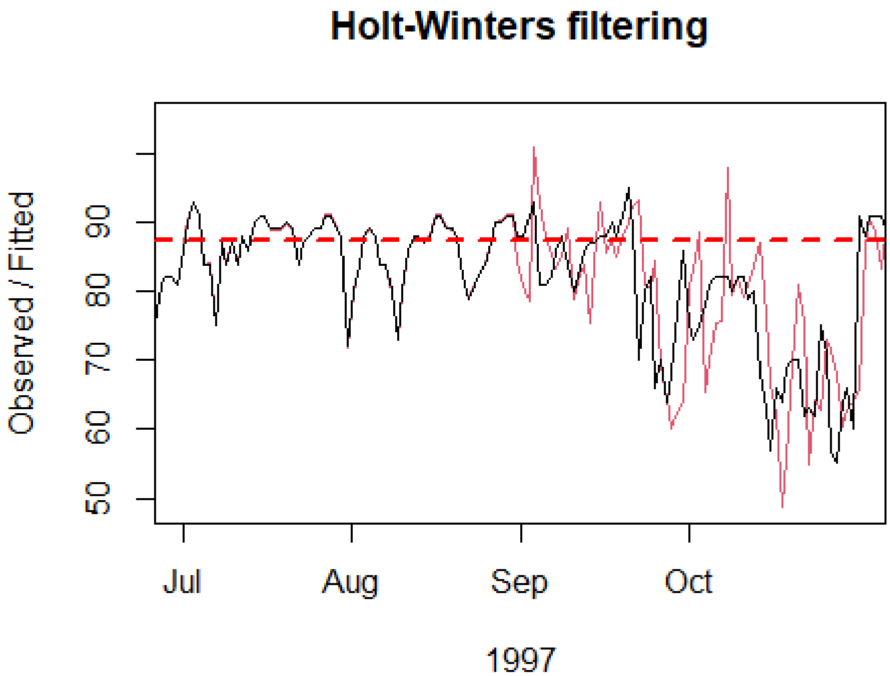

As we can observe, the extreme values around mid-August have been smoothed.

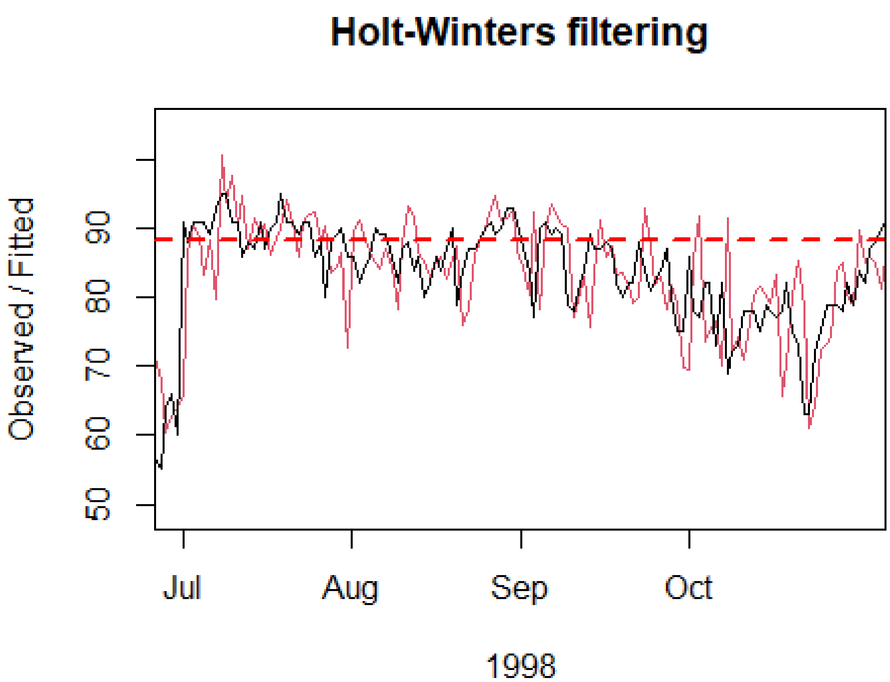
## 4.1. CUSUM to detect summer-ending date

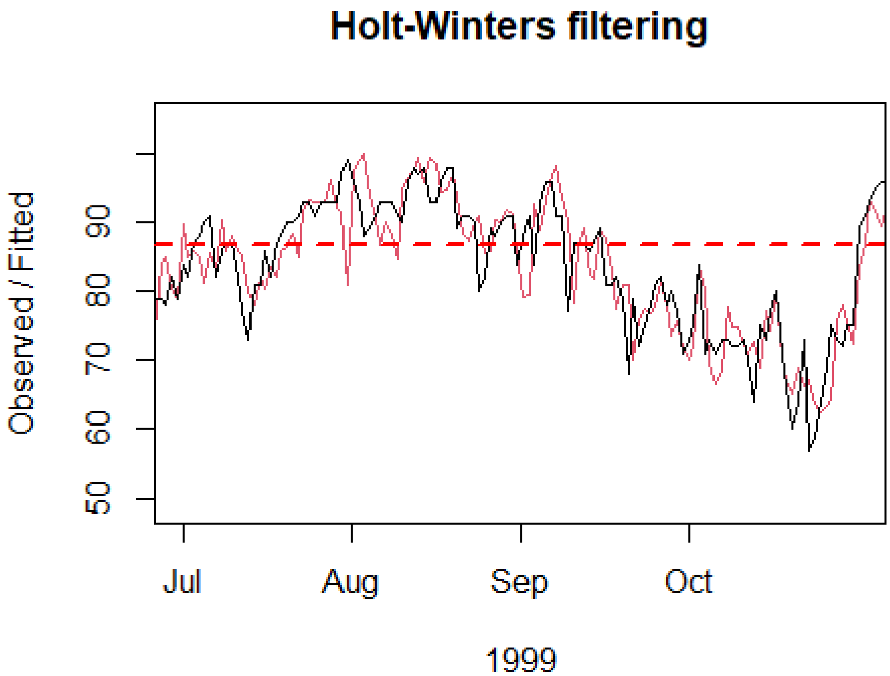
Now, we can apply again CUSUM on the smoothed time series to determine the date when summer ended each year and compare it with our previous results.

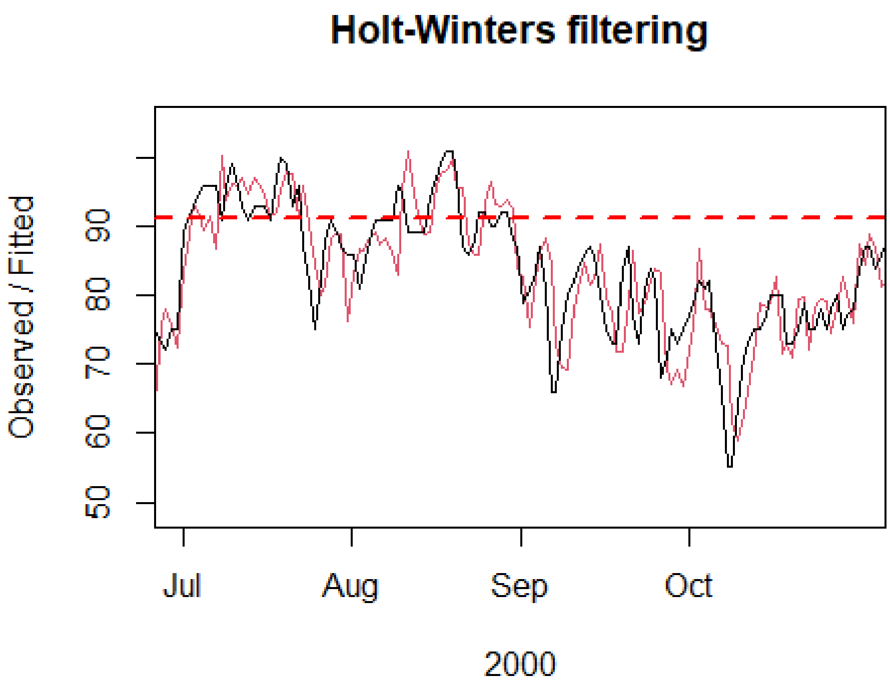
```
# Set smoothed temperatures
smoothed_temps <- hw_model$fitted[,1]

# Set a vector of the dates when summer ended each year
summer_end_dates <- numeric(ncol(temperature_data)-2)

# Initialize results list
results_list <- vector("list", length = 19)
k <- 1

for (i in 0:18) {
  # year for this iteration
  yr <- 1997 + i

  # Set the smoothed temperatures for each year
  smoothed_temps_years <- smoothed_temps[(123 * i + 1):(123 * (i + 1))]

  # Set the mean of temperatures in July (first 31 days)
```

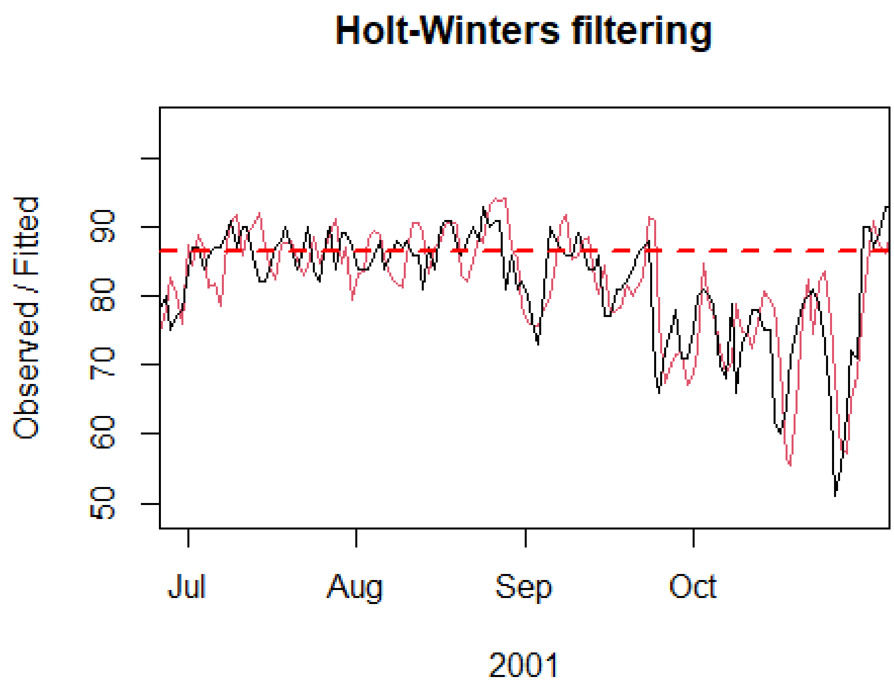
```r
  mean_july <- mean(smoothed_temps_years[1:31], na.rm = TRUE)

  # Set the standard deviation of all the points and set the values of
parameters C and T
  sd_all <- sd(smoothed_temps_years, na.rm = TRUE)
  C <- sd_all / 2
  T <- sd_all * 5

  # Call CUSUM function
  cusum_model <- cusum(smoothed_temps_years, mean_july, C)

  # Determine which values are greater than the threshold T
  index <- summer_end(cusum_model, T)

  # Take the first index if there are multiple
  index1 <- if (length(index) >= 1 &&
                !all(is.na(index))) index[1] else NA_integer_

  # Set date summer ended and the temperature recorded on that day
  date_val <- if (!is.na(index1)) temperature_data$DAY[index1] else NA
  temp_val <- if (!is.na(index1)) temperature_data[index1, 3 + i] else NA

  # Plot the data for the year with fitted values
  plot(hw_model, xlab = paste(yr), xlim = c(yr, yr + 1), xaxt = "n")
  axis(1, at = seq(yr, yr + 0.75, by = 0.25), labels = c("Jul", "Aug",
"Sep", "Oct"))
  abline(h = mean_july, col = "red", lwd = 2, lty = 2)

  # store row info as a data.frame
  results_list[[k]] <- data.frame(
    year = yr,
    summer_end = ifelse(is.na(index1), NA, as.character(date_val)),
    temp_on_summer_end = ifelse(is.na(index1), NA, as.numeric(temp_val)),
    C = C,
    T = T,
    stringsAsFactors = FALSE
  )
  k <- k + 1
}
```

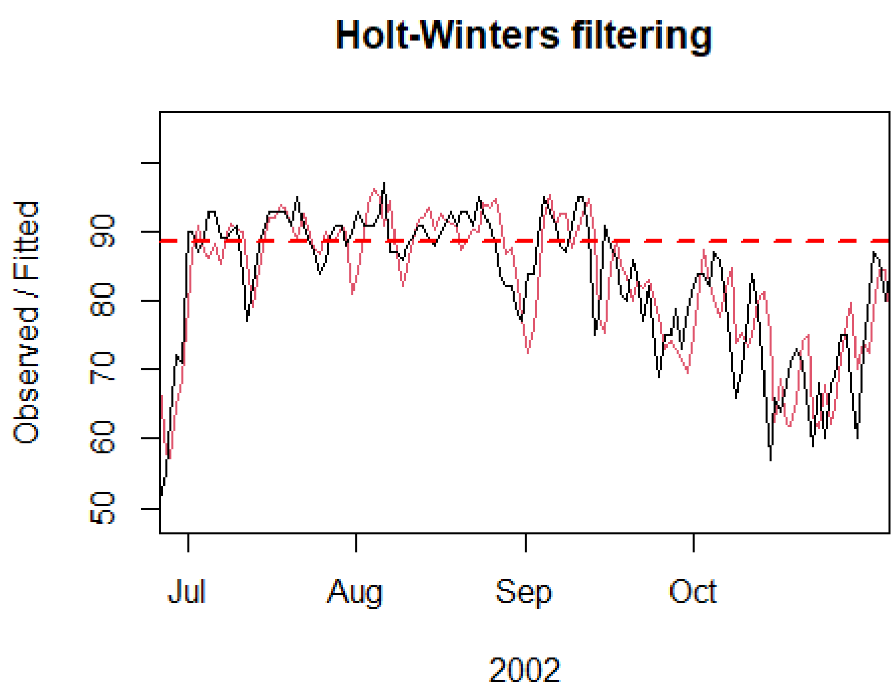

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
1997

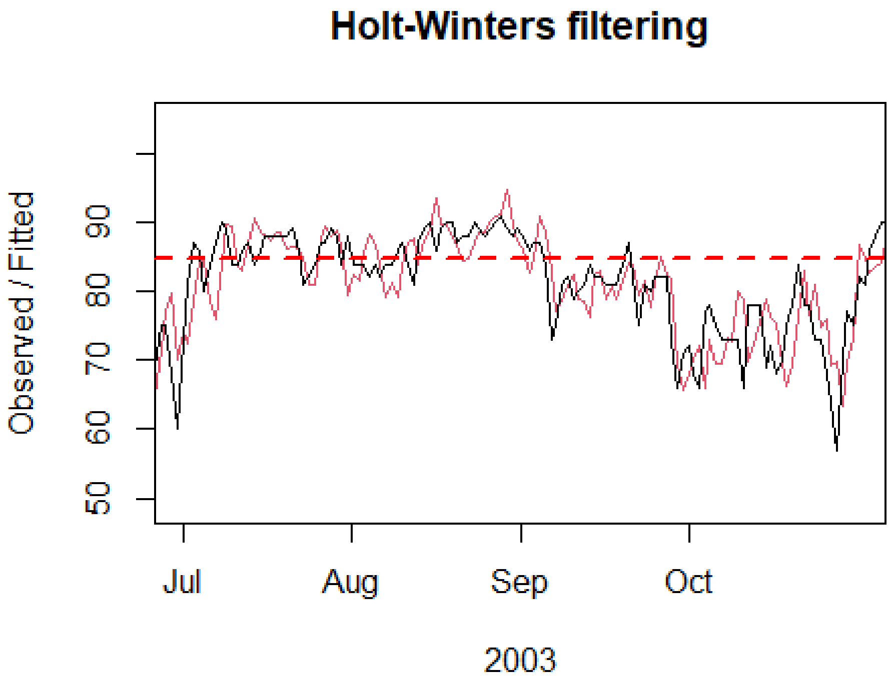

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
1998

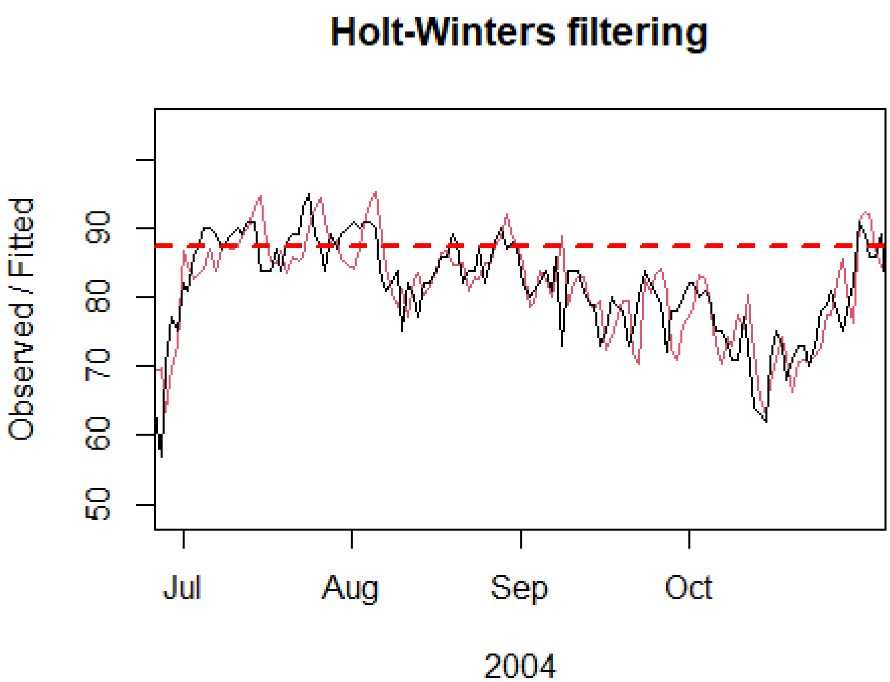

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
1999

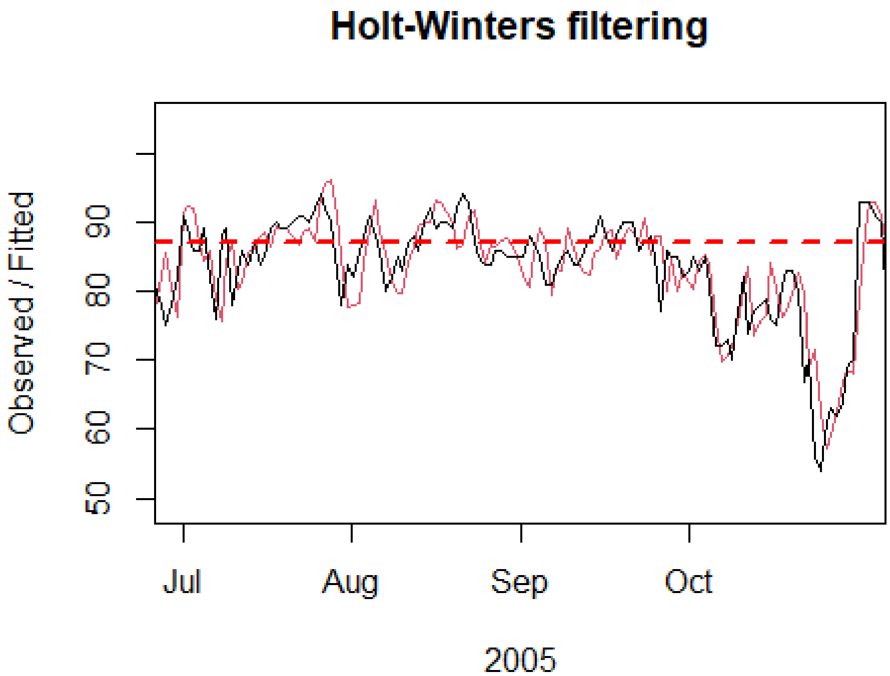

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2000

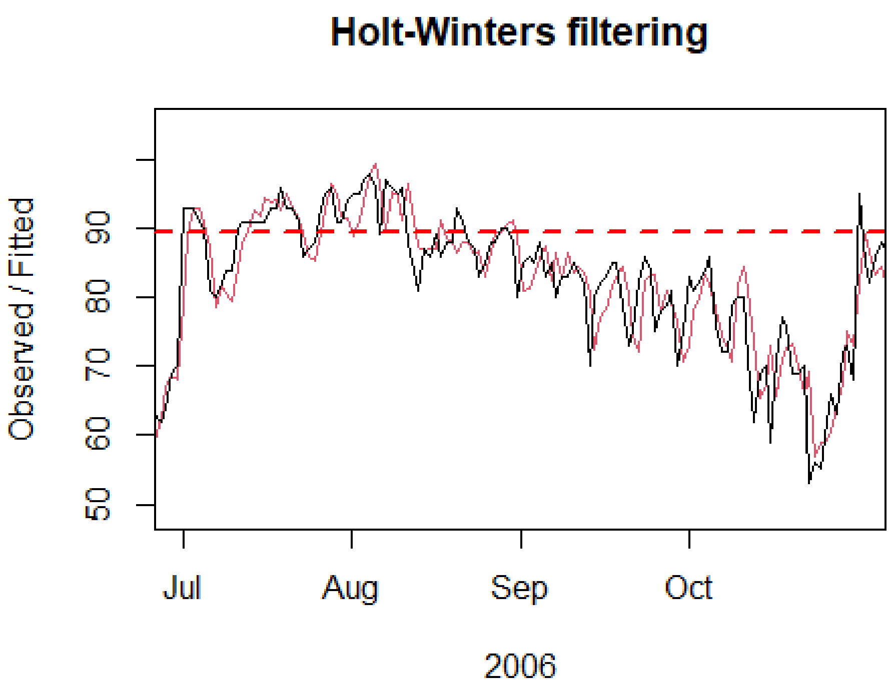

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2001

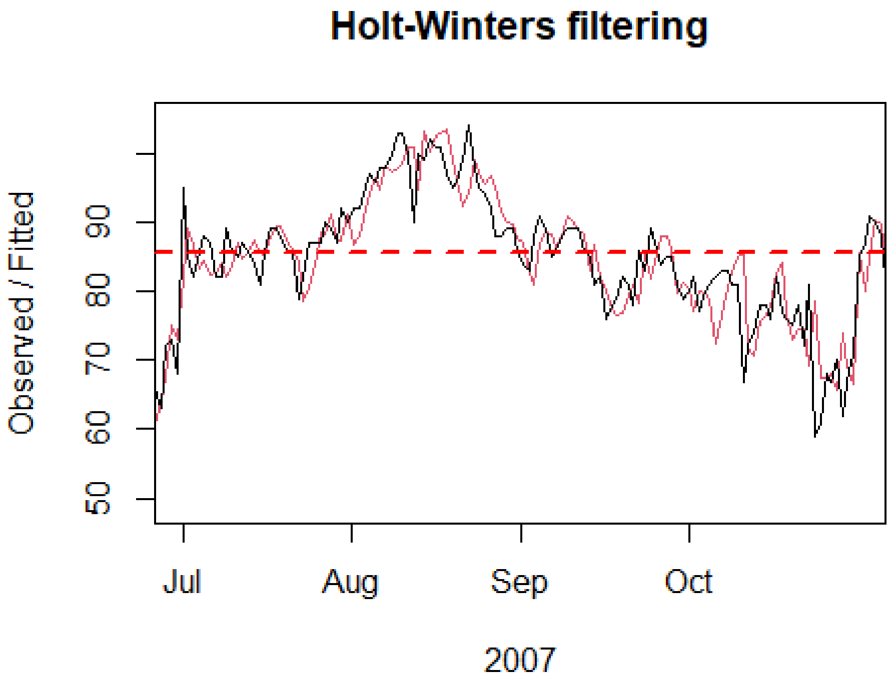

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2002

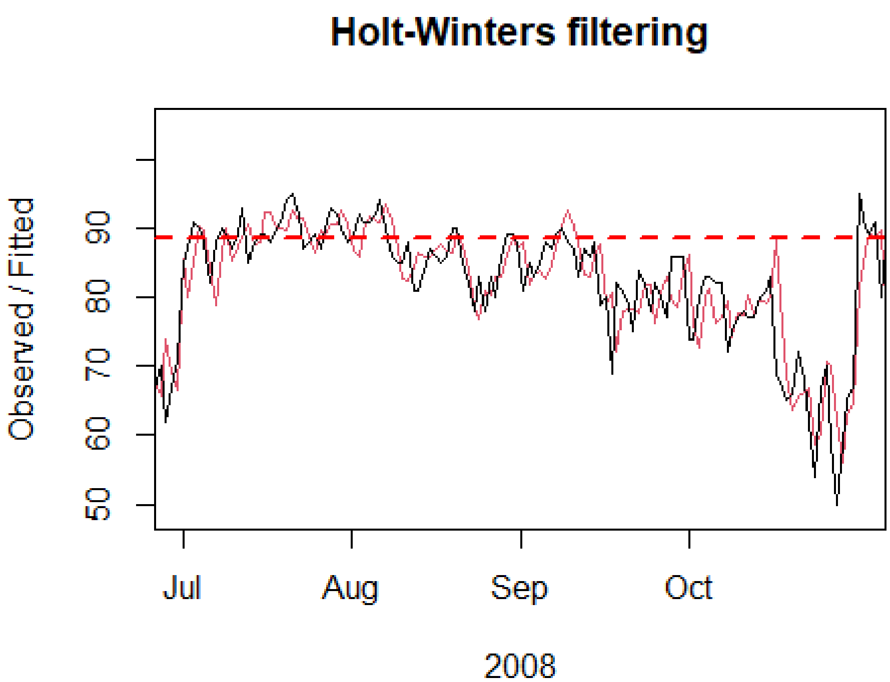

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2003

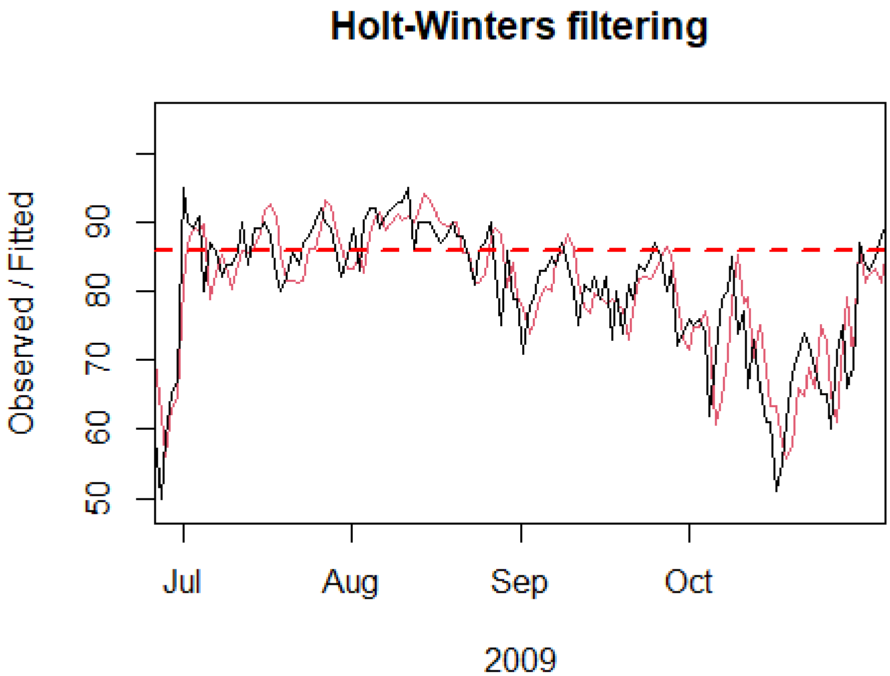

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2004

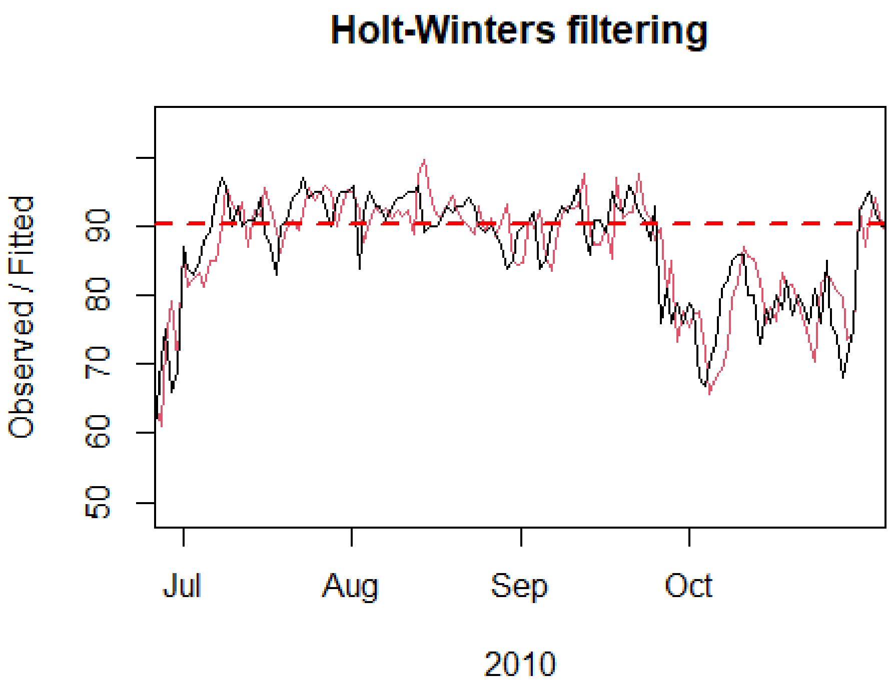

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2005

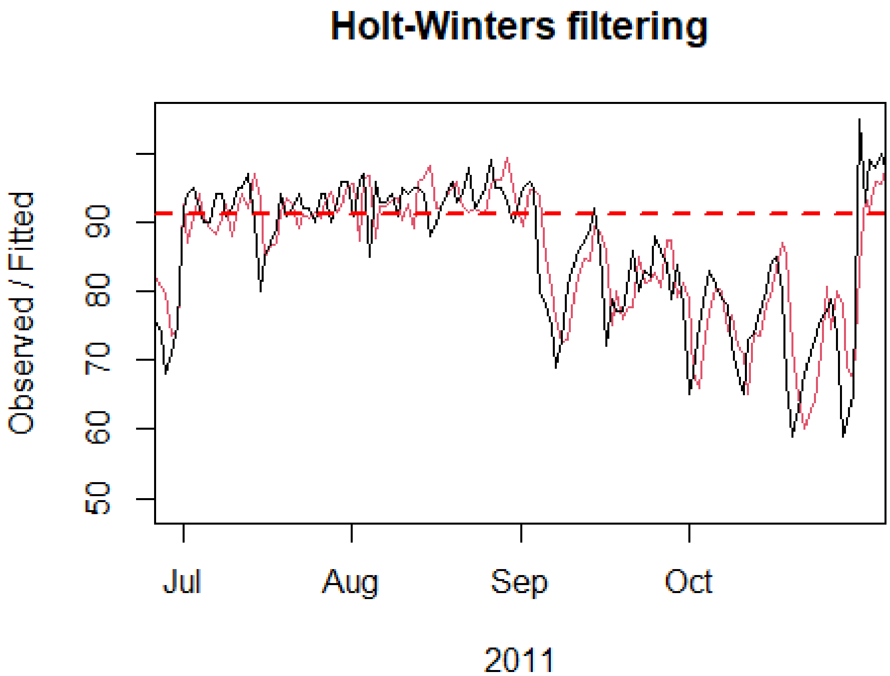

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2006

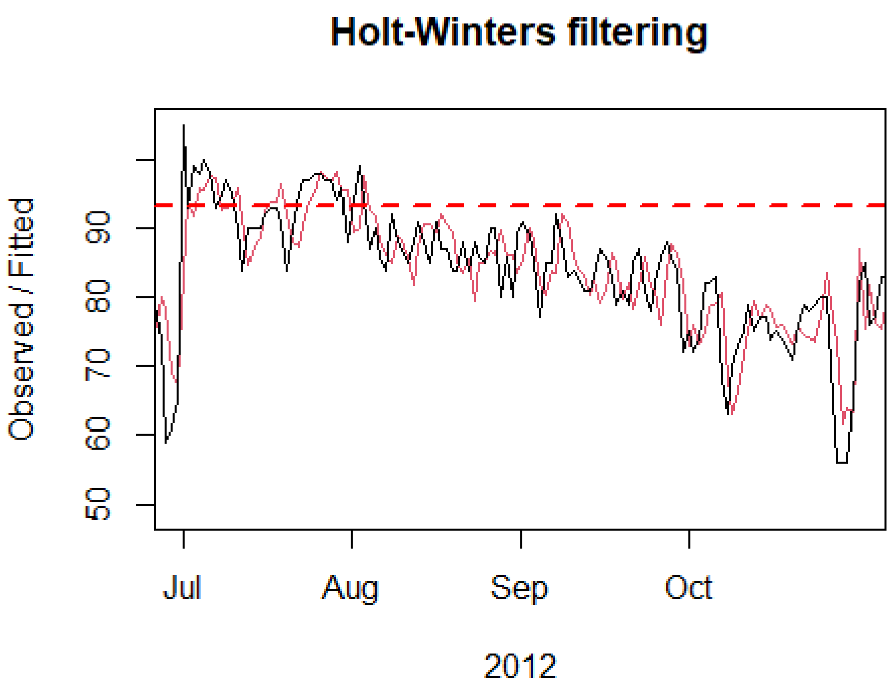

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2007

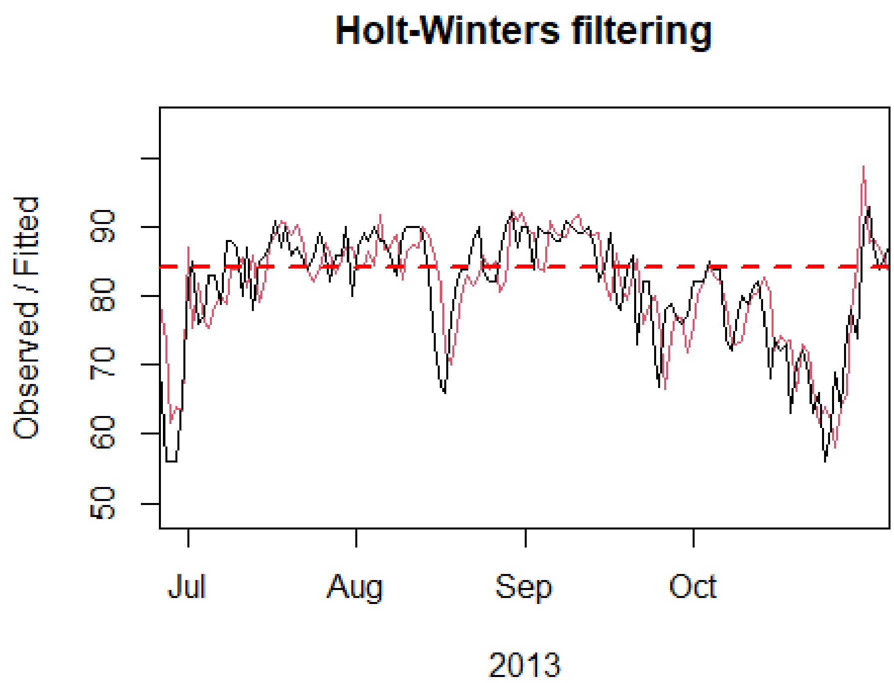

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2008

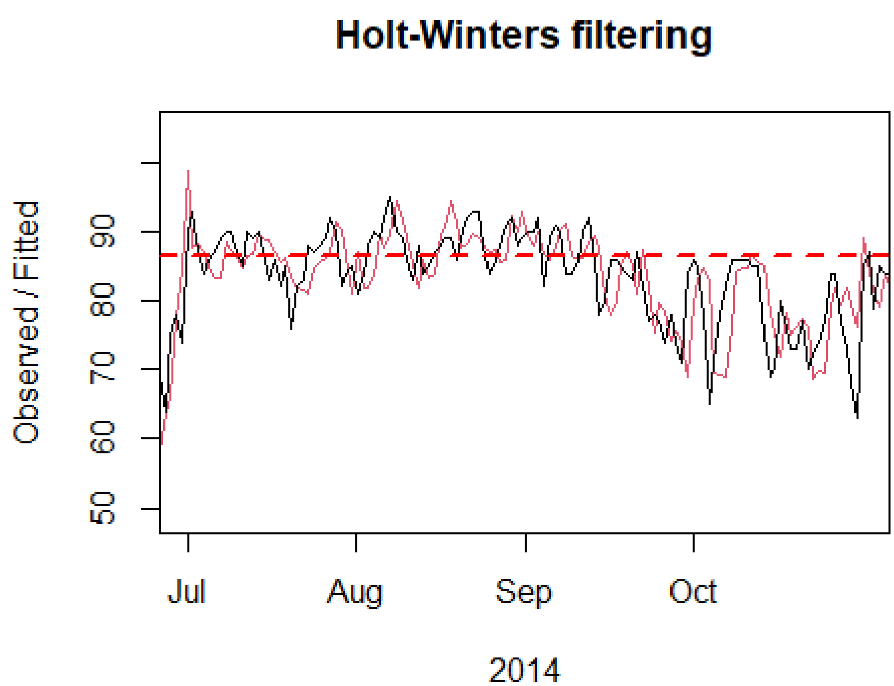

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2009

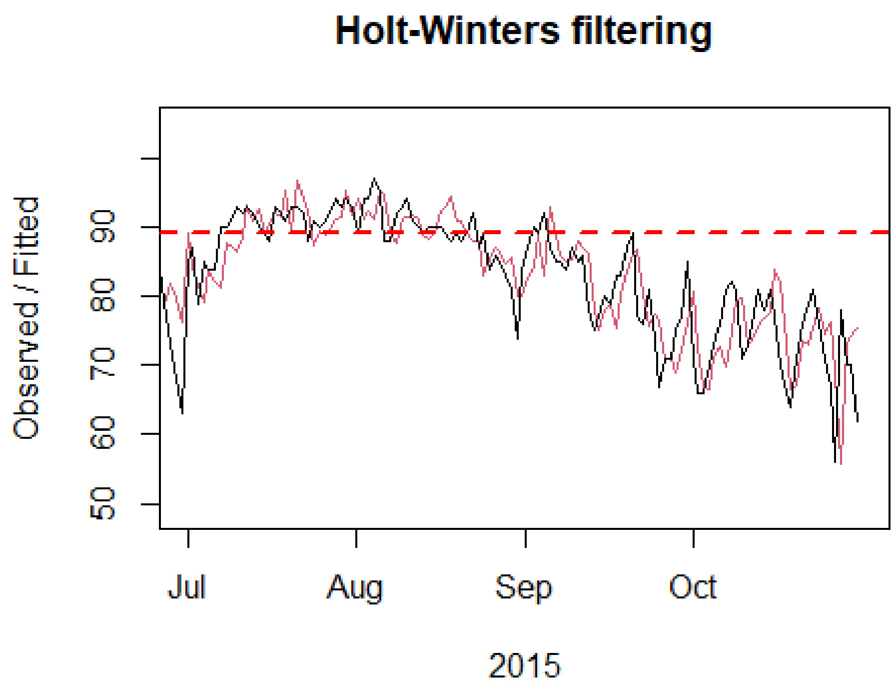

Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2010


Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2011


Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2012


Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2013


Holt-Winters filtering
Observed / Fitted
50
60
70
80
90
Jul
Aug
Sep
Oct
2014




```
# Combine into a single data.frame
results_df <- do.call(rbind, results_list)
rownames(results_df) <- NULL

# View results
print(results_df)

##    year summer_end temp_on_summer_end        C        T
## 1  1997     28-Sep                 68 4.915031 49.15031
## 2  1998     30-Sep                 75 3.775604 37.75604
## 3  1999     28-Sep                 80 4.797224 47.97224
## 4  2000      8-Sep                 75 4.727593 47.27593
## 5  2001     28-Sep                 78 4.177851 41.77851
## 6  2002     27-Sep                 75 4.463523 44.63523
## 7  2003     30-Sep                 71 3.503098 35.03098
## 8  2004     15-Sep                 73 3.364925 33.64925
## 9  2005      8-Oct                 73 3.689549 36.89549
## 10 2006     20-Sep                 73 4.621253 46.21253
## 11 2007     14-Oct                 78 4.291111 42.91111
## 12 2008     21-Sep                 75 4.038332 40.38332
## 13 2009     20-Sep                 81 4.320583 43.20583
## 14 2010      1-Oct                 79 3.618830 36.18830
## 15 2011     10-Sep                 84 4.628044 46.28044
## 16 2012     25-Aug                 85 4.124719 41.24719
## 17 2013     29-Sep                 76 3.683235 36.83235
## 18 2014     28-Sep                 74 3.037728 30.37728
## 19 2015     18-Sep                 83 4.121925 41.21925
```

Some of the results are different compared to previous results. For example, the model detected that summer in 2013 ended on September 29th, which is more aligned with the insights provided by the plots.

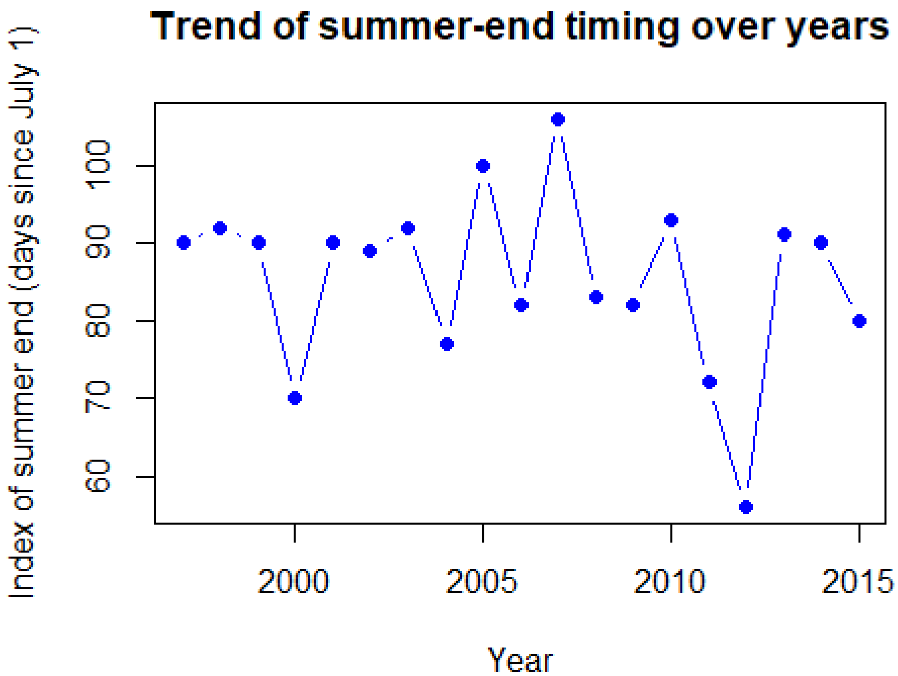
## 4.2. Determine whether summer has ended later over the years

Finally, we can employ the results to judge whether summer in Atlanta has gotten later over the years. To approach this, since we already know when summer ended each year, we can get the positional index of each date from July 1st - for example, if summer ended on July 31st, the positional index will be 31. Then, the list of indexes will serve as the input of a linear regression model that will determine whether summer has been progressively ending later.

```
# Get the index of each summer_end date in the main data$DAY column
summer_end_indices <- match(results_df$summer_end, temperature_data$DAY)

# View or store the list of indices
summer_end_indices

##  [1]  90  92  90  70  90  89  92  77 100  82 106  83  82  93  72  56
91  90  80
```

We can conduct a visual inspection of the dates.

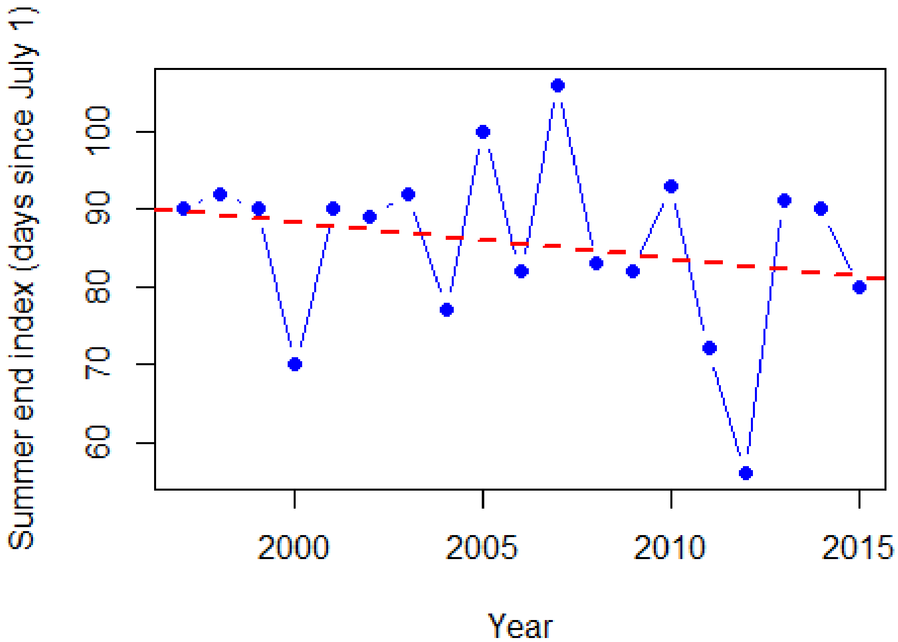
```
years <- 1997:2015

plot(years, summer_end_indices, type = "b", pch = 19, col = "blue",
     xlab = "Year", ylab = "Index of summer end (days since July 1)",
     main = "Trend of summer-end timing over years")
```




No significant trend can be extracted from the plot. We can fit a linear regression model for a more objective conclusion.

```
results_df$summer_end_index <- summer_end_indices
model <- lm(summer_end_index ~ year, data = results_df)
summary(model)

##
## Call:
## lm(formula = summer_end_index ~ year, data = results_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.695  -2.818   1.170   6.654  20.946
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1032.2175   954.4825   1.081    0.295
## year          -0.4719     0.4758  -0.992    0.335
##
## Residual standard error: 11.36 on 17 degrees of freedom
## Multiple R-squared:  0.0547, Adjusted R-squared:  -0.0009037
## F-statistic: 0.9837 on 1 and 17 DF,  p-value: 0.3352

plot(results_df$year, results_df$summer_end_index,
     type = "b", pch = 19, col = "blue",
```

```
     xlab = "Year", ylab = "Summer end index (days since July 1)")
abline(model, col = "red", lwd = 2, lty = 2)
```




The linear regression model shows a negative slope for the variables `years`, which indicates that the summer ending dates have been slightly getting sooner over the years. However, the p-value of 0.33 shows that the trend detected by the model is not statistically significant.

Therefore, we can conclude that visual inspection might suggest that summer has ended sooner over the years. However, there is no statistical evidence that supports this conclusion.