Analyzing Time-Series Data using Exponential Smoothing Model

Kent Ng

June 2, 2018

Using the 20 years of daily high temperature data for Atlanta (file temps.txt), let's build and leverage an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years.

Let's first load the given data into a dataframe and set a seed value as best practice.

```
set.seed(42)
data_7<-read.table("7.2tempsSummer2018.txt",header=TRUE)
head(data_7)
```

```
##     DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006
## 1 1-Jul    98    86    91    84    89    84    90    73    82    91    93
## 2 2-Jul    97    90    88    82    91    87    90    81    81    89    93
## 3 3-Jul    97    93    91    87    93    87    87    87    86    86    93
## 4 4-Jul    90    91    91    88    95    84    89    86    88    86    91
## 5 5-Jul    89    84    91    90    96    86    93    80    90    89    90
## 6 6-Jul    93    84    89    91    96    87    93    84    90    82    81
##   X2007 X2008 X2009 X2010 X2011 X2012 X2013 X2014 X2015
## 1    95    85    95    87    92   105    82    90    85
## 2    85    87    90    84    94    93    85    93    87
## 3    82    91    89    83    95    99    76    87    79
## 4    86    90    91    85    92    98    77    84    85
## 5    88    88    80    88    90   100    83    86    84
## 6    87    82    87    89    90    98    83    87    84
```

For the HoltWinter (exponential smoothing) model, we require our data to be in a time series object rather than a dataframe.

```
data_7_vector<-as.vector(unlist(data_7[,2:21]))
data_ts<-ts(data_7_vector,start=1996, frequency=123)
class(data_ts)
```

```
## [1] "ts"
```

```
head(data_ts)
```

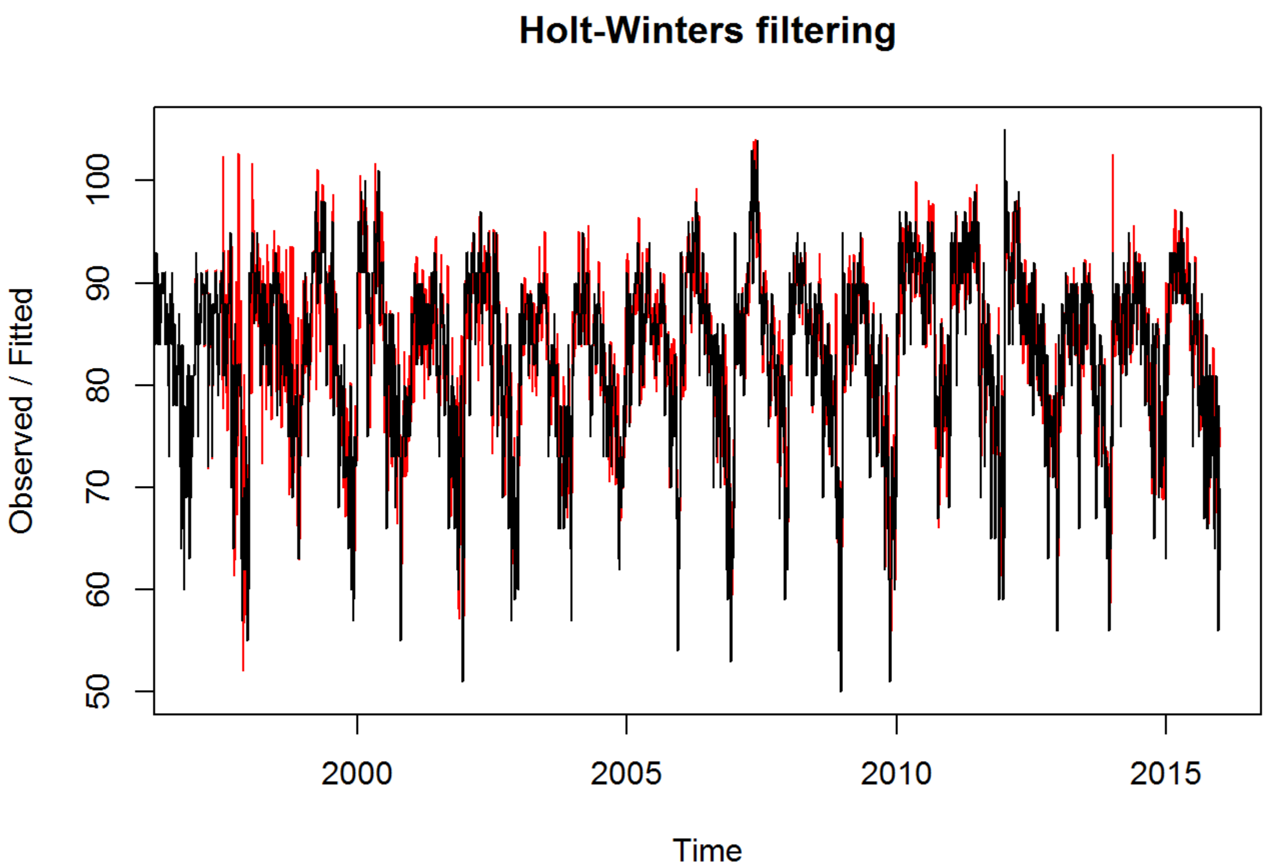
```
## [1] 98 97 97 90 89 93
```

Now that we have our time series object, let's call the HoltWinter method. By setting alpha, beta and gamma to NULL, R will find the optimal values for each of the parameters that gives our model the best fit. Further, as mentioned in the lectures, there are two ways exponential smoothing can deal with seasonality. One of the key differences between the two is that for additive seasonality, the amplitude of the seasonal variation is independent of the level; contrarily, multiplicative seasonality is dependent on the level. Here we will utilize the multiplicative method as taught in the lectures.

```
HoltWinterModel<-HoltWinters(data_ts, alpha=NULL,beta=NULL,gamma=NULL, seasonal="multiplicative"
)
```

We can plot our HoltWinterModel to see how the smoothing works (the red shows the plot of calculated baseline values). The STR() function will also show us the alpha, beta and gamma used in the model.

```
plot(HoltWinterModel)
```




```
str(HoltWinterModel)
```

```
## List of 9
##  $ fitted      : Time-Series [1:2337, 1:4] from 1997 to 2016: 87.2 90.4 93 90.9 84 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:4] "xhat" "level" "trend" "season"
##  $ x           : Time-Series [1:2460] from 1996 to 2016: 98 97 97 90 89 93 93 91 93 93 ...
##  $ alpha       : Named num 0.615
##   ..- attr(*, "names")= chr "alpha"
##  $ beta        : Named num 0
##   ..- attr(*, "names")= chr "beta"
##  $ gamma       : Named num 0.55
##   ..- attr(*, "names")= chr "gamma"
##  $ coefficients: Named num [1:125] 73.67952 -0.00436 1.23902 1.23434 1.15951 ...
##   ..- attr(*, "names")= chr [1:125] "a" "b" "s1" "s2" ...
##  $ seasonal    : chr "multiplicative"
##  $ SSE         : num 68905
##  $ call        : language HoltWinters(x = data_ts, alpha = NULL, beta = NULL, gamma = NULL,
    seasonal = "multiplicative")
##  - attr(*, "class")= chr "HoltWinters"
```

In regards to the original question of whether the unofficial end of summer has gotten later over the 20 years, there are many ways to answer this. For example, we could simply conduct CUSUM on the temperature data, as we did in last week's homework. Instead, here, we will run a CUSUM change detection analysis on the seasonality coefficient (which acts as a proxy) to determine our last day of summer for each year. The seasonality coefficient signifies that the baseline is expected to change solely because of the current time period in the cycle.

Below, we will extract the seasonality coefficients from the Holt-Winter model we developed and store it in a matrix

```
seasonality<-matrix(HoltWinterModel$fitted[,4],123)
head(seasonality)
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## [1,] 1.052653 1.049468 1.120607 1.103336 1.118390 1.108172 1.140906
## [2,] 1.100742 1.099653 1.108025 1.098323 1.110184 1.116213 1.126827
## [3,] 1.135413 1.135420 1.139096 1.142831 1.143201 1.138495 1.129678
## [4,] 1.110338 1.110492 1.117079 1.125774 1.134539 1.126117 1.130758
## [5,] 1.025231 1.025233 1.044684 1.067291 1.084725 1.097239 1.115055
## [6,] 1.025838 1.025722 1.028169 1.042340 1.053954 1.067494 1.080203
##          [,8]     [,9]    [,10]    [,11]    [,12]    [,13]    [,14]
## [1,] 1.140574 1.125438 1.122063 1.161415 1.198102 1.198910 1.243012
## [2,] 1.154074 1.142187 1.131889 1.144549 1.134661 1.153433 1.165431
## [3,] 1.156092 1.165657 1.147982 1.149459 1.135756 1.153310 1.155197
## [4,] 1.137722 1.150639 1.146992 1.142497 1.150162 1.151169 1.157751
## [5,] 1.103877 1.120818 1.133733 1.132167 1.142714 1.139244 1.112909
## [6,] 1.094312 1.102680 1.092178 1.075766 1.088547 1.082185 1.103092
##         [,15]    [,16]    [,17]    [,18]    [,19]
## [1,] 1.243781 1.238435 1.300204 1.290647 1.254521
## [2,] 1.172935 1.190735 1.191956 1.219190 1.228826
## [3,] 1.157286 1.169773 1.189915 1.172309 1.169045
## [4,] 1.163844 1.159343 1.166605 1.167993 1.158956
## [5,] 1.132435 1.132045 1.145230 1.168161 1.170449
## [6,] 1.115071 1.118575 1.121598 1.134962 1.145475
```

We can now export our seasonality into Excel to conduct our CUSUM change detection analysis.

```
library(xlsx)
```

```
## Loading required package: rJava
```

```
## Loading required package: xlsxjars
```

```
#workbook<-loadWorkbook("CUSUM.xlsx")
#sheets<-getSheets(workbook)
#sheet<-sheets[[1]]
#addDataFrame(seasonality,sheet,col.names=FALSE,row.names=FALSE,startRow=2,startColumn=3)
write.csv(seasonality, "CUSUM.csv")
```