HOMEWORK 4. ISYE 6501

Guillermo de la Hera Casado

September 17th, 2019

Question 7.1. Describe a situation for which exponential smoothing would be appropiate. Data to be used and alpha value.

I work for a large Pay-TV company in Africa. Time series analysis and forecasting of the customer base is done in a basic way by my peers. There is no visibility of the time serie components and only moving averages are used.

We could use exponential smoothing to improve the short term forecast predictions, as well as understanding better the trend and seasonality impact.

Regarding the data, we would need the daily customer base records, for at least the last 4-5 years. Alpha would be probably around 50%, meaning that:

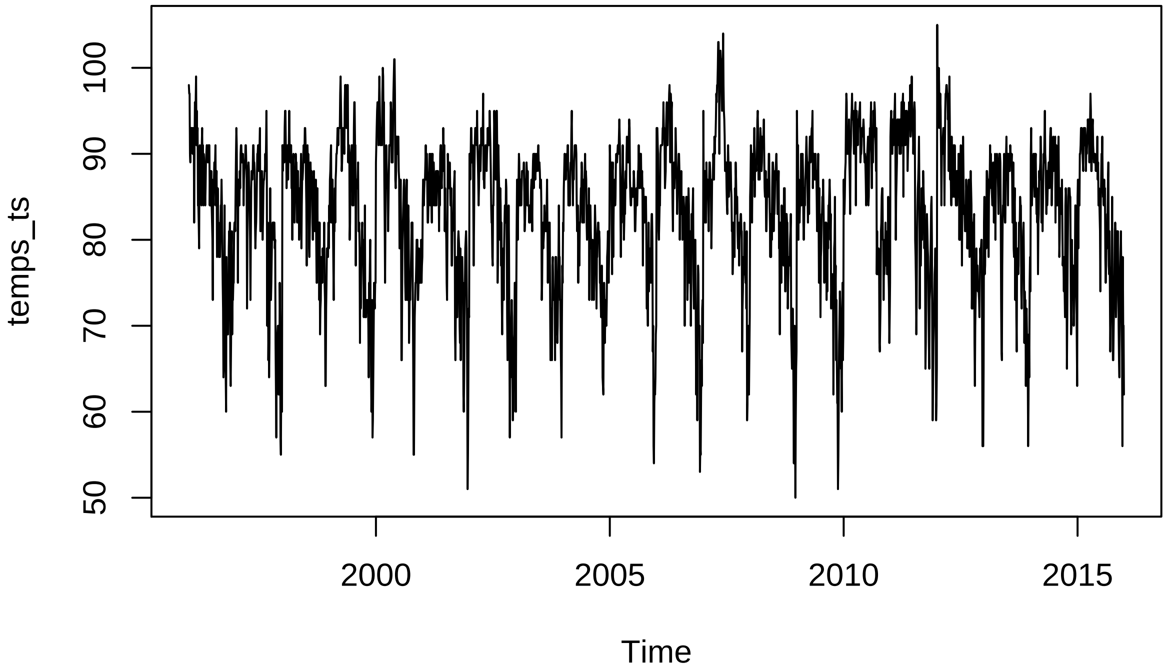
- Recent observations of the customer base size are indeed important to predict the next values.
- However quite a number of random factors can impact the base: price change, competitor movements, political issues, power cuts..

Question 7.2. Build an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years

The first step would be to read the data and store it within a single vector through *unlist*. After that, we create the time series object, mentioning that we want it to start in 1996 and have 123 observations as frequency for each period of interest (years)

```r
set.seed(101) # Set Seed so that same sample can be reproduced in future also

temps <- read.table("temps.txt", header = TRUE, stringsAsFactors = TRUE)
temps_vec <- as.vector(unlist(temps[,2:21]))
temps_ts <- ts(temps_vec, start = 1996, frequency = 123)
plot(temps_ts, plot.type = "s")
```



From the time series plot, we can observe the following:

- The data looks stationary, meaning that it's contained within a well defined range of values and statistics like mean and variance are very likely to remain constant.
- There doesn't seem to be a change in trend over time
- There is clearly seasonality going on every year
- From the picture it's not straightforward to tell whether the seasonality is additive (fixed amount regardless of the level) or multiplicative (proportional to the level). As I don't have the domain knowledge, I may need to let the model decides what's better to use based on SSE optimization.

As the next step, I will fit two different models, one with additive seasonality and another one with multiplicative seasonality. Alpha, beta and gamma values will be left as default, so that the algorithm automatically selects the best fit:

```
temps_Hw_add <- HoltWinters(temps_ts, alpha = NULL, beta = NULL,
                            gamma = NULL, seasonal = 'additive')
temps_Hw_add$SSE
```

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

```
temps_Hw_mul <- HoltWinters(temps_ts, alpha = NULL, beta = NULL,
                            gamma = NULL, seasonal = 'multiplicative')
temps_Hw_mul$SSE
```

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

The additive SSE for additive seasonality is actually smaller compared to the SSE obtained from multiplicative seasonality. Therefore we will continue with *temps_Hw_add* as the model of choice.

```
temps_Hw_add$alpha
```

```
##     alpha
## 0.6610618
```

```
temps_Hw_add$beta
```

```
## beta
##    0
```

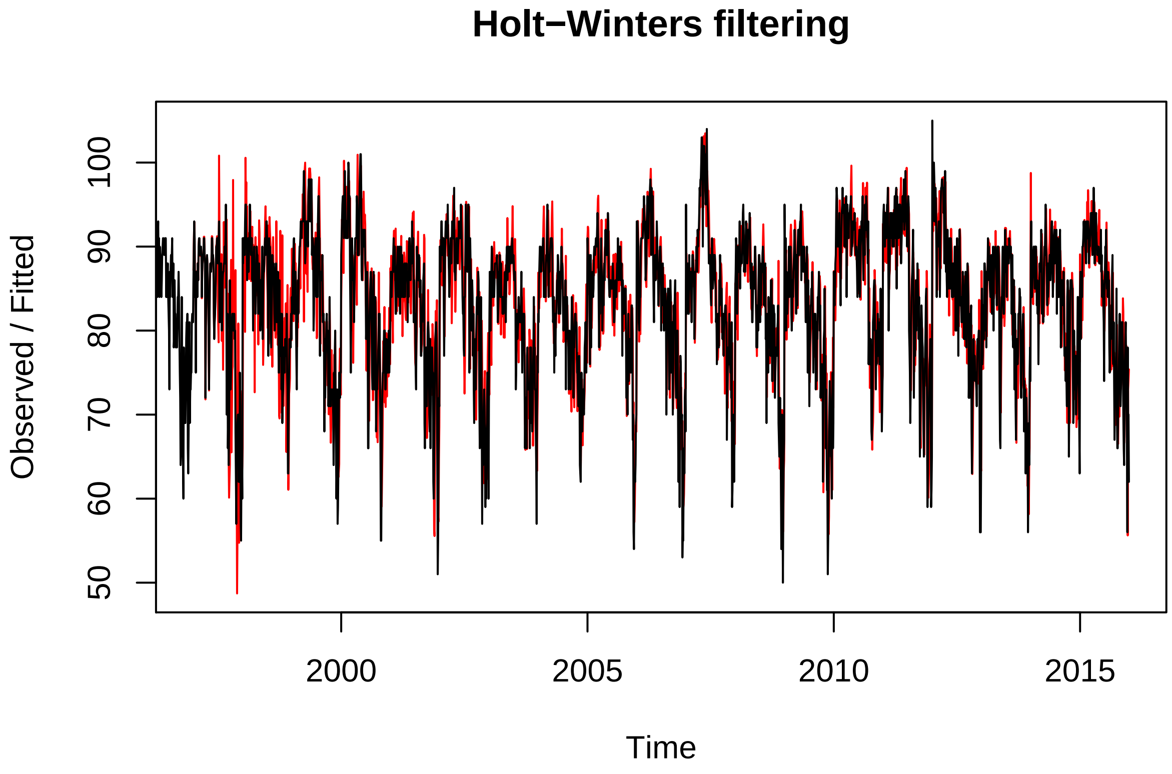
```
temps_Hw_add$gamma
```

```
##     gamma
## 0.6248076
```

By looking at the alpha, beta and gamma values we can confirm that the model doesn't appreciate any change in trend over time. With regards to alpha and gamma mixed feelings: the model appreciates some randomness although recent observations are also taken into consideration.

If we wanted to see how well the smoothed figures fit the actuals for the period, we can see that overall there is a great fit between the red line (fitted) and the black (observed). We can also observe that the first year has been taken as a baseline as there are no fitted values there.

```
plot(temps_Hw_add)
```



```
print(head(temps_Hw_add$fitted))
```

```
##          xhat    level        trend    season
## [1,] 87.17619 82.87739 -0.004362918  4.303159
## [2,] 90.32925 82.09550 -0.004362918  8.238119
## [3,] 92.96089 81.87348 -0.004362918 11.091777
## [4,] 90.93360 81.89497 -0.004362918  9.042997
## [5,] 83.99752 81.93450 -0.004362918  2.067387
## [6,] 84.04358 81.93177 -0.004362918  2.116168
```

Let's look at the seasonality factors, and bring that data into a tab delimited file. It will show the years as columns and the seasonality factor per day of the year as rows:

```
temps_Hw_sf <- matrix(temps_Hw_add$fitted[,4], nrow=123)
head(temps_Hw_sf)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,]  4.303159  4.054077  9.349191  8.130519  9.261932  8.532675 10.833750
## [2,]  8.238119  8.168393  8.457426  7.810603  8.686298  9.197265  9.837370
## [3,] 11.091777 11.100059 11.213419 11.470326 11.416575 11.012491 10.210645
## [4,]  9.042997  9.057058  9.529053 10.185524 10.863856 10.209557 10.532285
## [5,]  2.067387  2.067912  3.708913  5.588420  7.004562  8.024549  9.444664
## [6,]  2.116168  2.106939  2.232254  3.394700  4.340175  5.462823  6.487578
##           [,8]      [,9]     [,10]     [,11]    [,12]     [,13]     [,14]
## [1,] 10.596553  9.559008  9.526730 12.454249 15.18901 15.052357 17.963213
## [2,] 11.663073 10.874704 10.162519 11.021049 10.13469 11.583196 12.246533
## [3,] 12.021948 12.738520 11.468041 11.483482 10.45074 11.722773 11.707197
## [4,] 10.867244 11.779117 11.552601 11.117308 11.69283 11.650082 12.084663
## [5,]  8.518818  9.749682 10.738920 10.552850 11.29192 10.939712  8.908885
## [6,]  7.648008  8.250531  7.421824  6.156207  7.12694  6.619883  8.329696
##          [,15]     [,16]     [,17]    [,18]    [,19]
## [1,] 17.946705 17.796582 22.385105 21.31072 19.45438
## [2,] 12.817176 14.277046 14.037278 16.07719 17.19256
## [3,] 11.803979 12.726046 14.244787 12.98723 12.74993
## [4,] 12.461996 12.000781 12.458283 12.62307 11.98537
## [5,] 10.345553 10.214813 11.172561 12.77225 12.94898
## [6,]  9.151988  9.361362  9.444097 10.35325 11.13812
```

```
write.table(temps_Hw_sf, file = 'sf.txt', sep ='\t')
```

The next steps are performed in the spreadsheet, together with the plots - kindly check it for your reference. The approach is as follows:

- To create three different groups: Group 1 **(1997-2003)**, Group 2 **(2004-2010)** and Group 3 **(2011-2015)**. **If it's true that summers are ending later recently, then we should see the seasonality from Group 3 holding high values for longer.**
- To perform a exploratory analysis, just by looking at the avg seasonality factors for each one of the groups over time. From the plot, it doesn't really look like Group 3 holds high seasonality factors any better than Group 1 and Group 2, but we may want to verify it using CUSUM.
- To apply the CUSUM model to each one of the groups individually, in order to calculate S values.
- To keep C = 3 and critical value = 25, after adjusting seasonality in order to avoid making the model too exposed to false positives
- To plot the S values of the three groups vs critical value creating a control chart

From the control chart,**it is quite clear that Group 3 doesn't hold high seasonality values for longer** compared to Group 1 and Group 2. Actually, it's the first group crossing the critical threshold, indicating a trend change and pointing out that summer is ending.

- Group 3 crosses the threshold on Sep 18th
- Group 1 crosses the threshold on Sep 21st
- Group 2 crosses the threshold on Sep 22nd

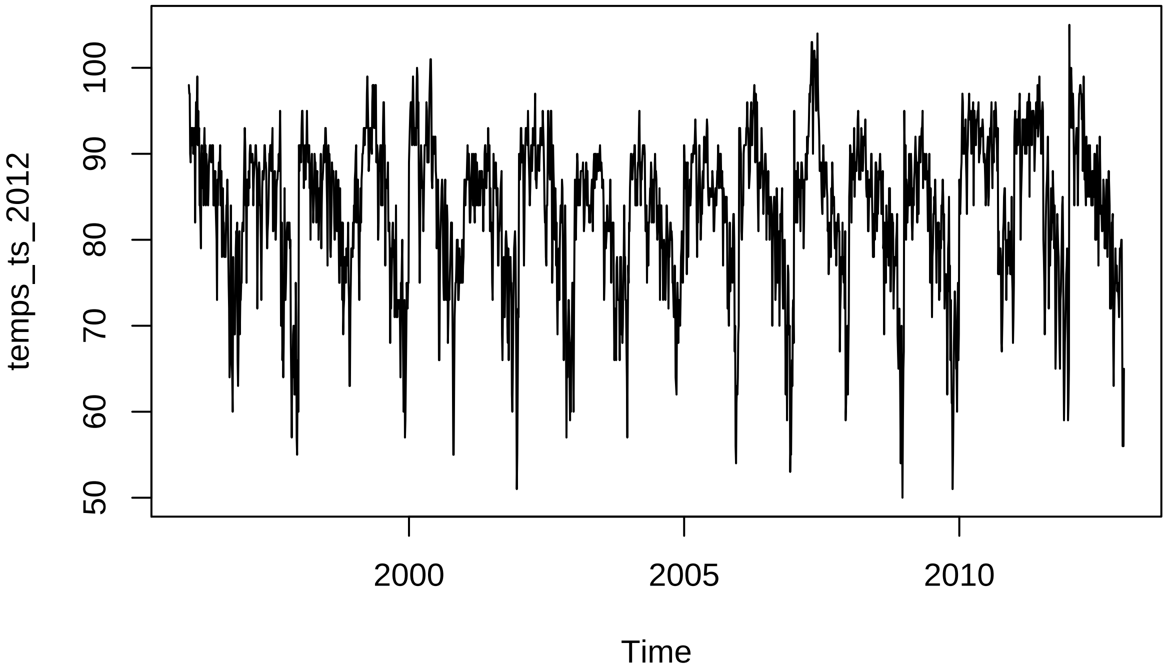
Based on this fact, **we can't say that unofficial end of summer has gotten later over the last 20 years**

Question 7.2. Added bonus. Is Holt Winters model as accurate for unseen observations?

We have seen that the fitted values from HoltWinters match quite accurately the observed values. However, there is the possibility that this happens just because we are evaluating the performance of the model on the data that has been trained to build it.

Let's make it interesting and train Holt Winters only with data up to 2012, additive seasonality.

```
temps_vec_2012 <- as.vector(unlist(temps[,2:18]))
temps_ts_2012 <- ts(temps_vec_2012, start = 1996, frequency = 123)
plot(temps_ts_2012)
```



```
temps_Hw_train <- HoltWinters(temps_ts_2012, alpha = NULL, beta = NULL,
                              gamma = NULL, seasonal = 'additive')
```

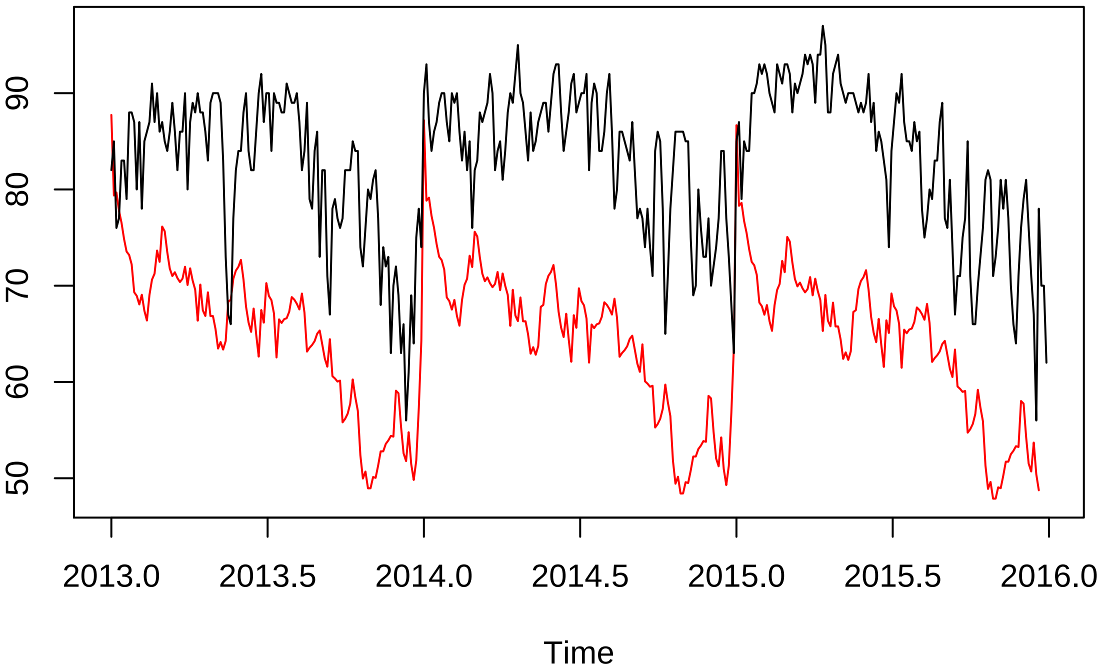
Now let's predict 2013-2015 values by using the *predict* function:

```
pred <- predict(temps_Hw_train, n.ahead = 366, prediction.interval = FALSE)
```

And now let's try to see the predicted values (red) vs the real ones (black): The seasonality is well captured, however the model predictions are consistently below the real observations

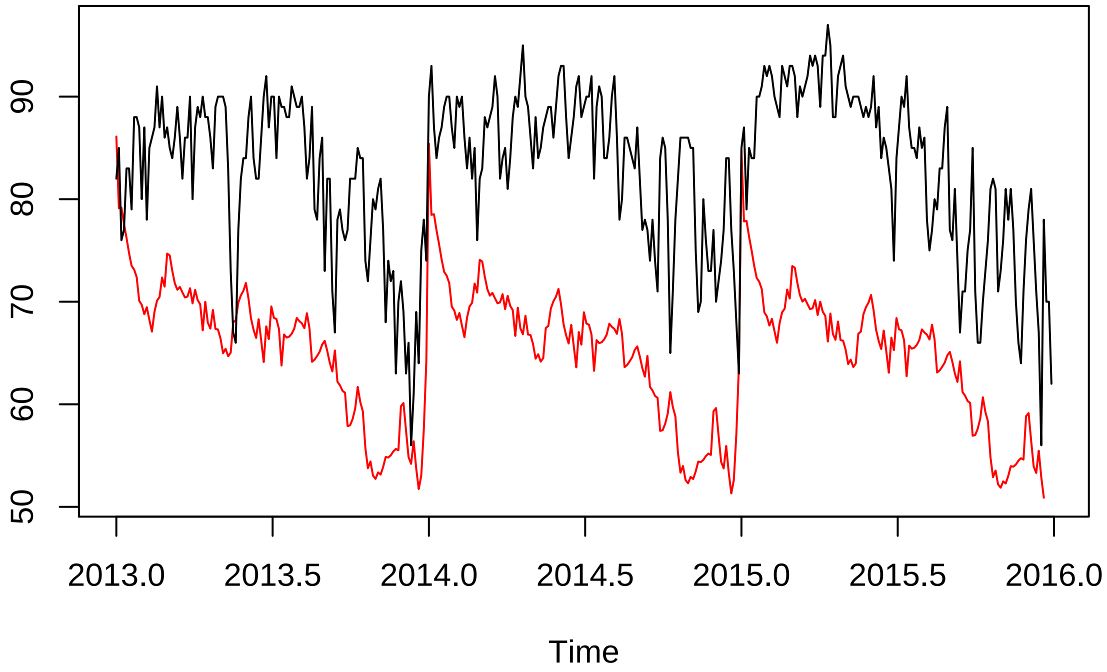
```
temps_vec_2013_2015 <- as.vector(unlist(temps[,19:21]))
temps_ts_2013_2015 <- ts(temps_vec_2013_2015, start = 2013, frequency = 123)
```

```
ts.plot(pred, temps_ts_2013_2015, gpars = list(col = c("red","black")))
```



What if we were to predict the values with multiplicative seasonality and check vs real observations?

```
temps_Hw_train <- HoltWinters(temps_ts_2012, alpha = NULL, beta = NULL,
                              gamma = NULL, seasonal = 'multiplicative')
pred <- predict(temps_Hw_train, n.ahead = 366, prediction.interval = FALSE)
ts.plot(pred, temps_ts_2013_2015, gpars = list(col = c("red","black")))
```



As we can see, the picture is quite similar, meaning that there is not a huge difference between both approaches. As a conclusion we can say that Holt Winters does a good job capturing the seasonality and overall shape, **but we may find later on in the course a model that can provide closer forecasts to the real value e.g. ARIMA.**