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 α (the first smoothing parameter) to be closer to 0 or 1, and why?

Working in a manufacturing company, exponential smoothing would be taken into account when studying the correlation between machine calibration interval and the quality of goods. Theoretically, machine needs to be calibrated periodically to assure the produced goods meet the quality standards. To study, assembly machines will be chosen on the same production line, and the production goods dimension data will be collected to determine whether the machine calibration is within acceptable range. Machine calibration interval is set to 3 days, 7 days, 14 days to observe possible cyclical patterns, and exponential smoothing model would then be built using machine's types as cyclic factors with the production goods dimension data as the observed value. Value of α is expected to be closer to 1 since there shouldn't be too much variation in goods dimension for various time interval for the same machine, and therefore there shouldn't be much randomness in the system.

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.)

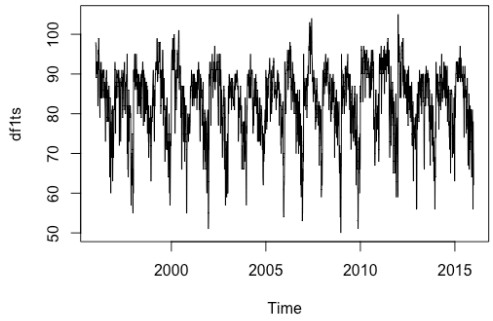
Note: in R, you can use either `HoltWinters` (simpler to use) or the `smooth` package's `es` function (harder to use, but more general). If you use `es`, the Holt-Winters model uses `model="AAM"` in the function call (the first and second constants are used "A"dditively, and the third (seasonality) is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

```
# Clear global environment, load package, load and preview dataset
> rm(list=ls())
> df1 <- read.table("/Users/…/temps.txt",header=T,stringsAsFactors = F)
> head(df1)
   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

Transform data into time series, with starting year being 1996 and frequency set to 123 which is the number of days from July to October. Plot to get a preview of the time series dataset.

```
> df1 <- as.vector(unlist(df1[,2:21]))
> df1ts <- ts(df1, start = 1996, frequency = 123)
> plot(df1ts)
```

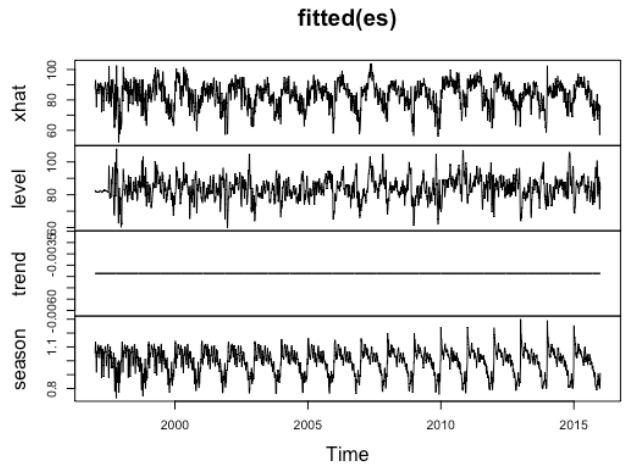


Run exponential smoothing model with Holt-Winters. Alpha, beta and gamma are set to NULL to find optimal values, and seasonal is set to multiplicative since seasonal variations exist for this dataset. Results from exponential smoothing has alpha equals to 0.6150, and gamma equals to 0.5495, with trend estimate beta being close to 0 and value of beta being 0.

```
> es <- HoltWinters(df1ts, alpha = NULL, beta = NULL, gamma = NULL, seasonal = "multiplicative")
> es
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.

Call:
HoltWinters(x = df1ts, 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…
```

By the looks of the following plot, variation for season seems to be increasing as time increases. To study whether the end of summer has gotten later, we put the seasonal factors into a separate matrix.

```
> plot(fitted(es))
```




```
> es_season <- matrix(es$fitted[,4], nrow = 123)
> head(es_season)
       [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
[1,] 1.052653 1.049468 1.120607 1.103336 1.118390 1.108172 1.140906 1.140574
[2,] 1.100742 1.099653 1.108025 1.098323 1.110184 1.116213 1.126827 1.154074
[3,] 1.135413 1.135420 1.139096 1.142831 1.143201 1.138495 1.129678 1.156092
[4,] 1.110338 1.110492 1.117079 1.125774 1.134539 1.126117 1.130758 1.137722
[5,] 1.025231 1.025233 1.044684 1.067291 1.084725 1.097239 1.115055 1.103877
[6,] 1.025838 1.025722 1.028169 1.042340 1.053954 1.067494 1.080203 1.094312
       [,9]    [,10]    [,11]    [,12]    [,13]    [,14]    [,15]    [,16]
[1,] 1.125438 1.122063 1.161415 1.198102 1.198910 1.243012 1.243781 1.238435
[2,] 1.142187 1.131889 1.144549 1.134661 1.153433 1.165431 1.172935 1.190735
[3,] 1.165657 1.147982 1.149459 1.135756 1.153310 1.155197 1.157286 1.169773
[4,] 1.150639 1.146992 1.142497 1.150162 1.151169 1.157751 1.163844 1.159343
[5,] 1.120818 1.133733 1.132167 1.142714 1.139244 1.112909 1.132435 1.132045
[6,] 1.102680 1.092178 1.075766 1.088547 1.082185 1.103092 1.115071 1.118575
       [,17]    [,18]    [,19]
[1,] 1.300204 1.290647 1.254521
[2,] 1.191956 1.219190 1.228826
[3,] 1.189915 1.172309 1.169045
[4,] 1.166605 1.167993 1.158956
[5,] 1.145230 1.168161 1.170449
[6,] 1.121598 1.134962 1.145475
```

To determine whether the end of summer got later, we can first look at whether the mean temperature gets cooler. According to the following result, mean temperature is indeed dropping. Next, we implement cusum function to observe the date in which the season factor falls below the baseline, indicating the temperature drops below the threshold and suggesting that summer has ended.

```
> for (i in 1:ncol(es_season)){
+     season_avg[i] = mean(es_season[,i])
+ }
> season_avg
 [1] 1.0000000 0.9981882 0.9980547 0.9975095 0.9971271 0.9961954 0.9955108
 [8] 0.9957393 0.9956360 0.9949667 0.9948162 0.9946495 0.9943300 0.9941211
```

[15] 0.9940949 0.9933907 0.9932476 0.9935215 0.9928824

Cusum function is applied as below, with T value defined as the 5 times the standard deviation of the first-year data (baseline), and the C value defined as one half the standard deviation of the first-year data. Result suggests that the summer end date has gotten later over the years from 2-Oct in 1997 to 7-Oct in 2015 with a maximum difference of 5 days.

```
> cusum <- function(df, mean, T, C){
+     mu = list()
+     cusum = 0
+     date = 1
+     while(date <= nrow(df)){
+         ini = df[date,]
+         cusum = max(0, (mean-C-ini)+cusum)
+         if (cusum >=T){
+             mu = date
+             break
+         }
+         date = date+1
+     }
+     return(mu)
+ }
> T = sd(es_season[,1])*5
> C = sd(es_season[,1])/2
> end_date = vector()
> for (i in 1:ncol(es_season)){
+     end_date[i] = cusum(df=as.matrix(es_season[,i]), mean=1, T=T, C=C)
+ }
> data.frame(Year = colnames(es_season),Day = df1[end_date,1])
    Year   Day
1  X1997 2-Oct
2  X1998 2-Oct
3  X1999 3-Oct
4  X2000 4-Oct
5  X2001 5-Oct
6  X2002 5-Oct
7  X2003 6-Oct
8  X2004 6-Oct
9  X2005 6-Oct
10 X2006 6-Oct
11 X2007 6-Oct
12 X2008 7-Oct
13 X2009 7-Oct
14 X2010 7-Oct
15 X2011 6-Oct
16 X2012 6-Oct
17 X2013 7-Oct
18 X2014 7-Oct
19 X2015 7-Oct
```