Homework-4

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

In Retail Banking, tracking of credit card sign-up rate would be a potential use case for exponential smoothing model. Outside of actual sign-up rate for a specific card, information on card specific promotions, marketing campaigns, change in customer-value-propositions (CVP), and even existing roadshows all provide valuable data that will influence the model. On top of that, credit card sales tend to have seasonal trends and cyclical patterns that follow the market. A newly launched card will have strong upward trend due to the promotions held by the bank, and on the other hand, older cards with less attractive or competitive CVP tend to have a downward trend overtime.

I would expect the value of a to be closer to 0 as there is a lot of randomness in the credit card market. Promotions by competitors will also affect the observed sign-up rate in a particular day. Instead, I think it more importance should be placed on identifying and optimizing the trend and cyclical parameters as they will be the true drivers of sign-up rates.

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).

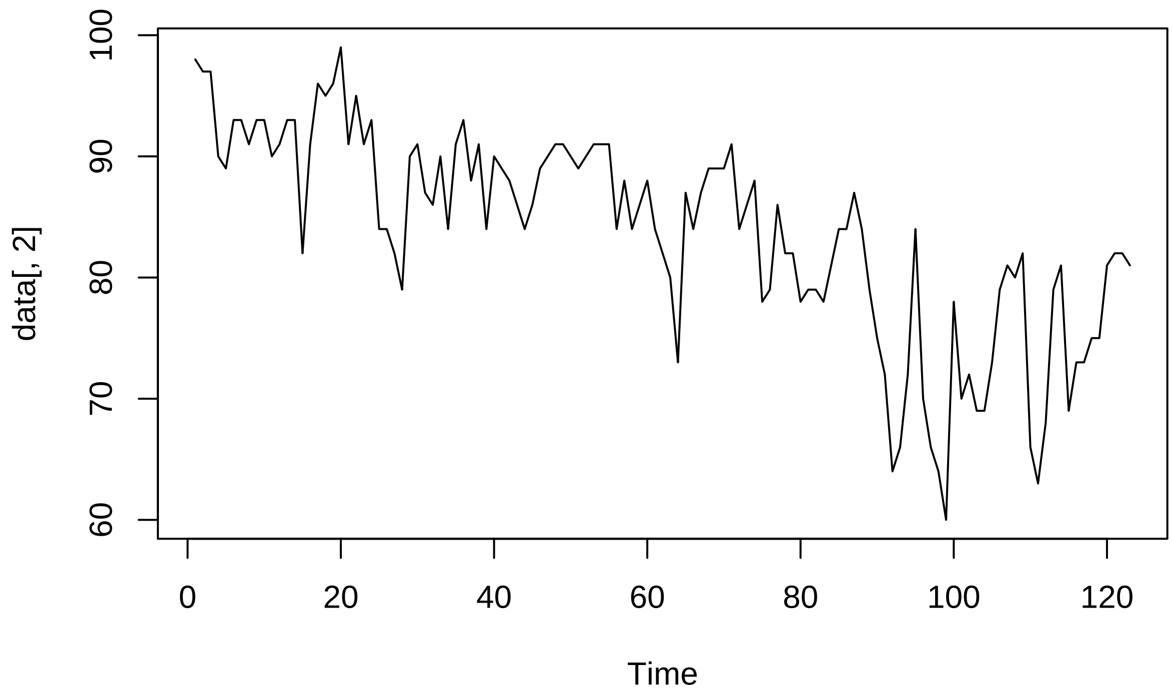
```
library(kernlab)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:kernlab':
##
##     alpha
```

```
set.seed(42069)

# load data
data = read.table("C:/Users/Admin/Desktop/MM/Homework 4/temps.txt",
                  stringsAsFactors = FALSE,
                  header = TRUE)
head(data)
```

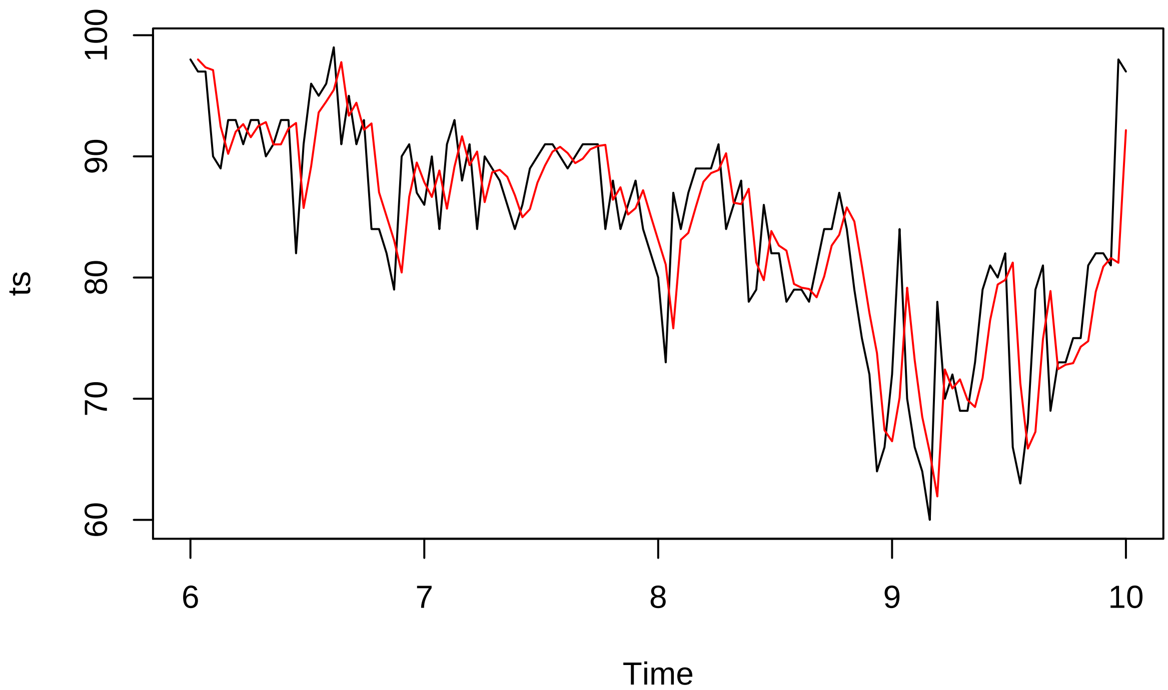
```
##     DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006 X2007
## 1 1-Jul    98    86    91    84    89    84    90    73    82    91    93    95
## 2 2-Jul    97    90    88    82    91    87    90    81    81    89    93    85
## 3 3-Jul    97    93    91    87    93    87    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
```

```
plot.ts(data[,2])
```



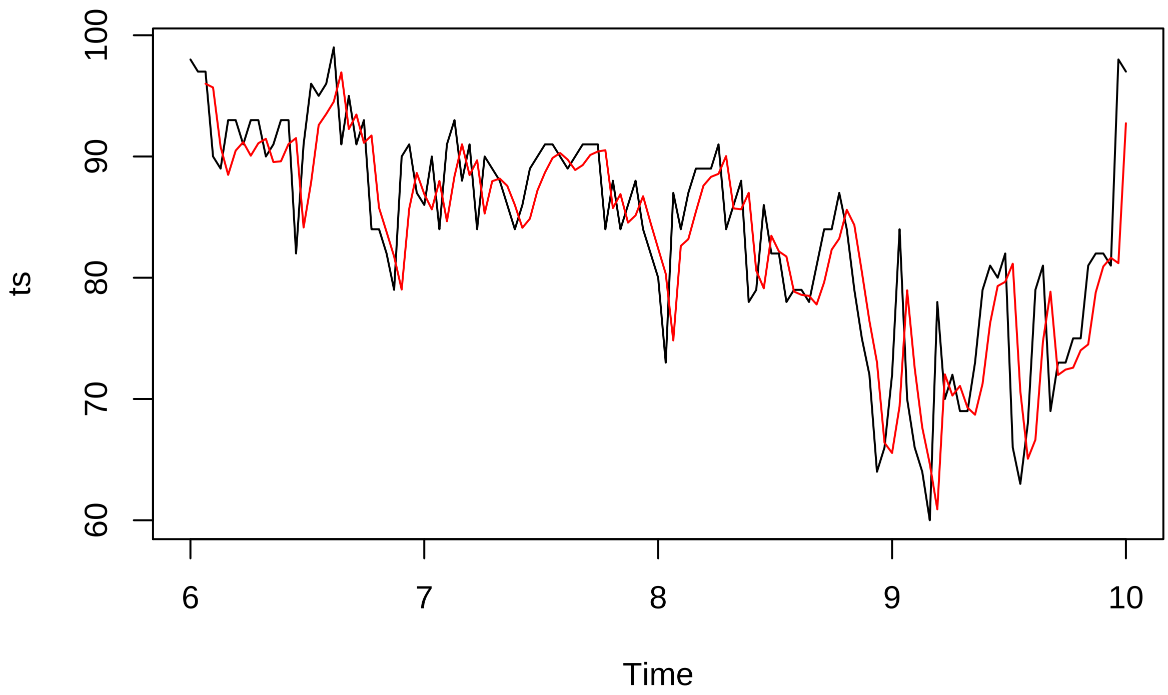

```r
# holtwinter's model function
model = function(a, b, g, s){
  ts = ts(unlist(data$X1996, use.names=FALSE),
          start = 6,
          end = 10,
          frequency = 31)
  data.mean = HoltWinters(ts,
                          alpha = a,
                          beta = b,
                          gamma = g,
                          seasonal = s)
  print(paste("a =", data.mean$alpha))
  print(paste("b =", data.mean$beta))
  print(paste("g =", data.mean$gamma))
  plot(ts)
  lines(data.mean$fitted[,1], col='red')
}

# testing model fit
model(NULL, FALSE, FALSE, 'additive')
```

```
## [1] "a = 0.652338589742349"
## [1] "b = FALSE"
## [1] "g = FALSE"
```



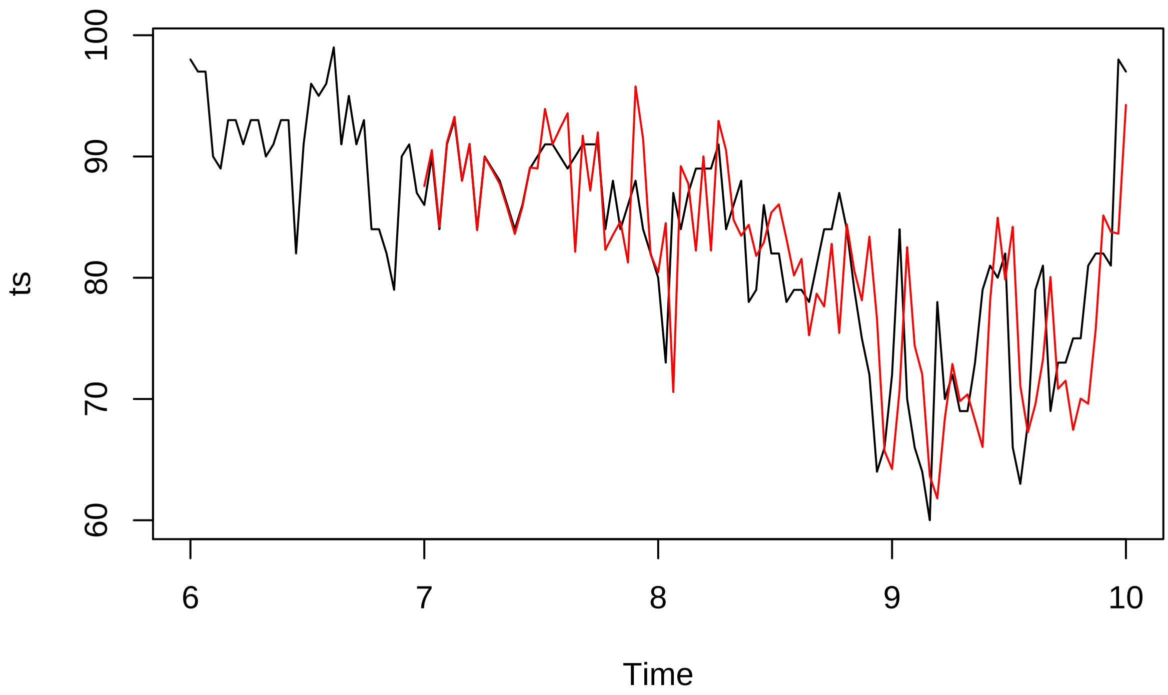

```r
# second order with trend
model(NULL, NULL, FALSE, 'additive')
```

```
## [1] "a = 0.671599098040214"
## [1] "b = 0.0239407279462201"
## [1] "g = FALSE"
```



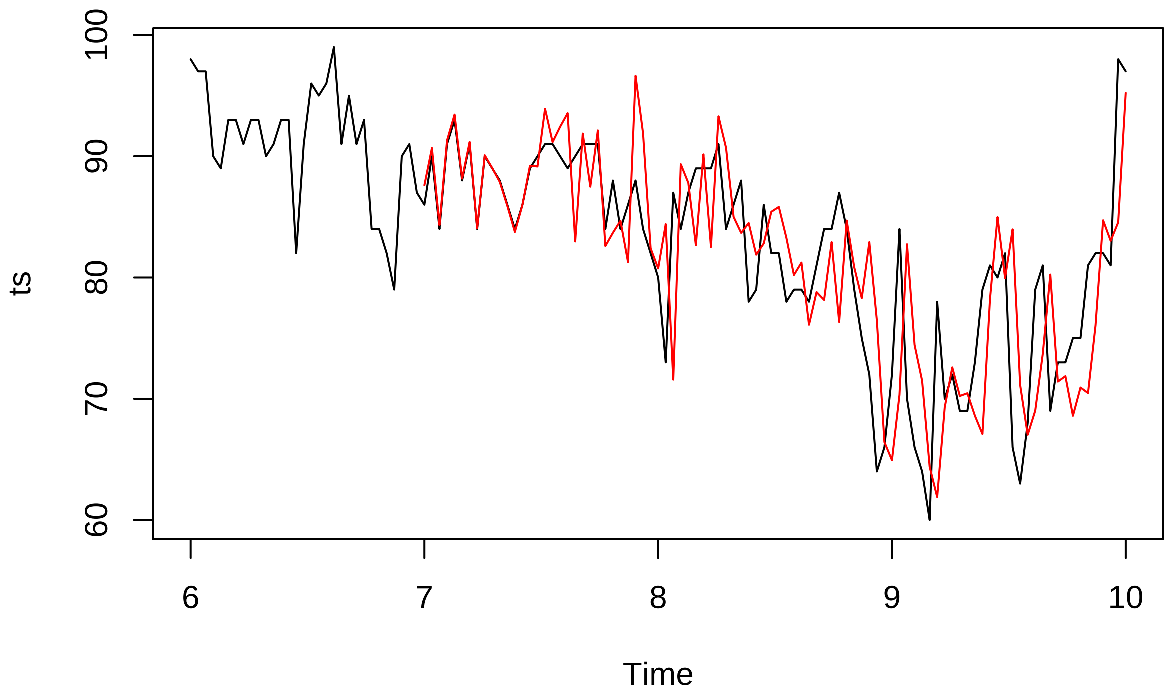

```
# third order with seasonality
model(NULL, NULL, NULL, 'additive')
```

```
## [1] "a = 0.701184396186389"
## [1] "b = 0"
## [1] "g = 1"
```



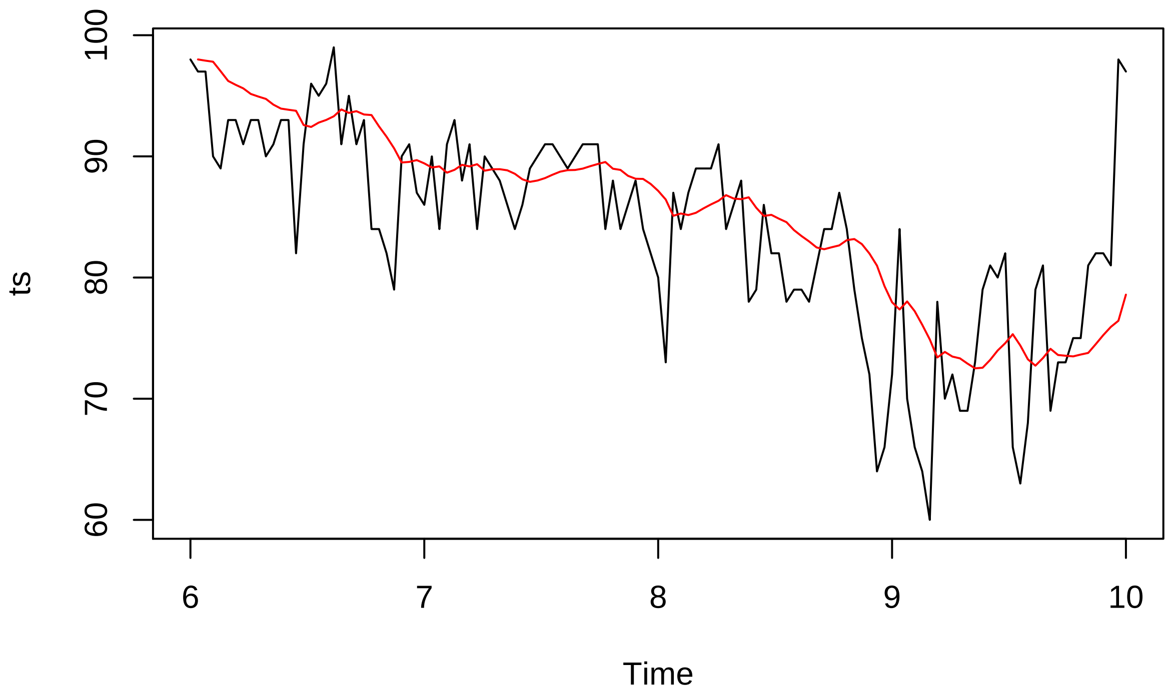

```
# third order with multiplicative seasonality
model(NULL, FALSE, NULL, 'multiplicative')
```

```
## [1] "a = 0.693992217527573"
## [1] "b = FALSE"
## [1] "g = 1"
```



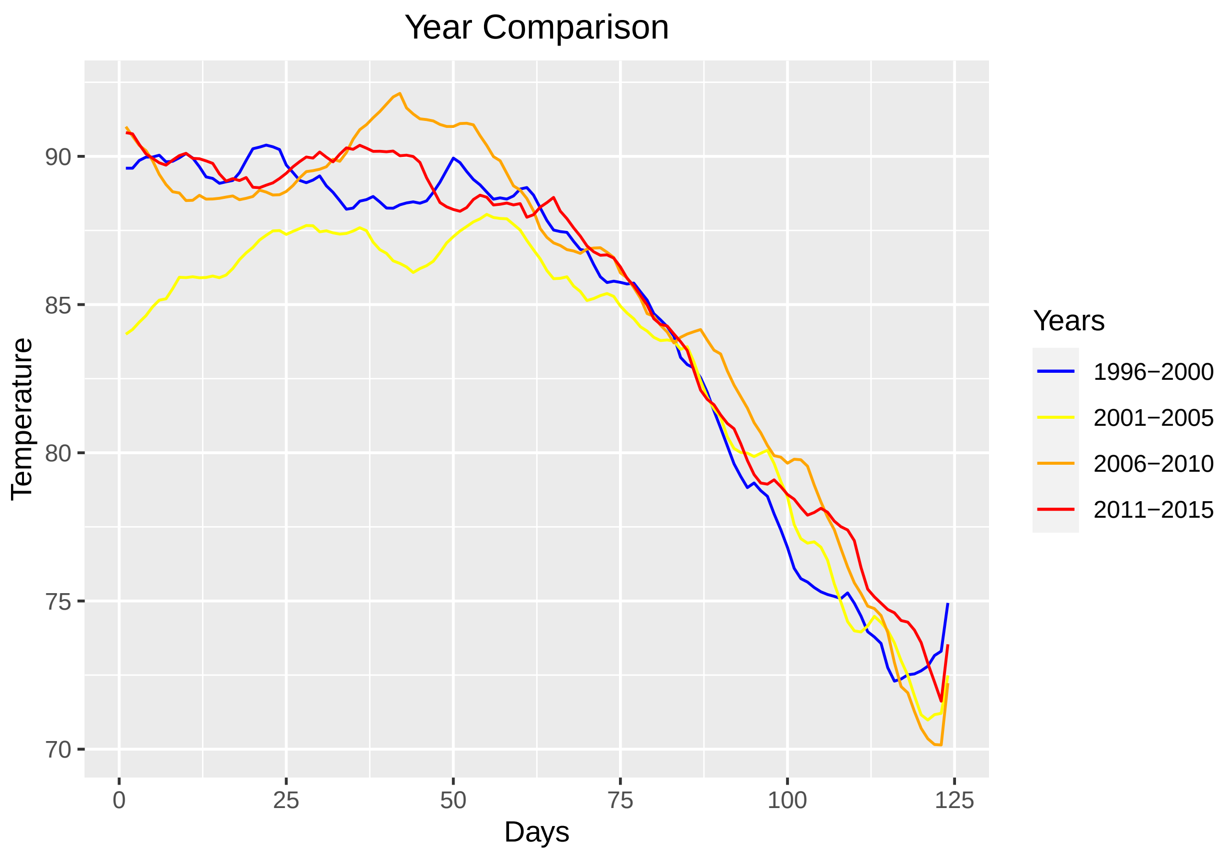

```
# lower alpha value selected since daily temp is highly varying
# trend is always close to 0 indicating no significant trend
# seasonality does not smooth the data properly
model(0.1, FALSE, FALSE, NULL)
```

```
## [1] "a = 0.1"
## [1] "b = FALSE"
## [1] "g = FALSE"
```




```
# loading predictions into data frame
ls = list()
for (i in 2:21){
  ts = ts(unlist(data[,i], use.names=FALSE),
          start = 6,
          end = 10,
          frequency = 31)
  data.mean = HoltWinters(ts,
                          alpha = 0.1,
                          beta = FALSE,
                          gamma = FALSE)
  ls[[i]] = as.numeric(data.mean$fitted[,1])
}
mx = do.call(cbind, ls)
df = as.data.frame(mx)

# visualizing smoothed temperature change across years
df$MEAN96_00 = (df$V1 + df$V2 + df$V3 + df$V4 + df$V5)/5
df$MEAN01_05 = (df$V6 + df$V7 + df$V8 + df$V9 + df$V10)/5
df$MEAN06_10 = (df$V11 + df$V12 + df$V13 + df$V14 + df$V15)/5
df$MEAN11_15 = (df$V16 + df$V17 + df$V18 + df$V19 + df$V20)/5

ggplot(df, aes(x = (1:nrow(df)))) +
  geom_line(aes(y = MEAN96_00, color = '1')) +
  geom_line(aes(y = MEAN01_05, color = '2')) +
  geom_line(aes(y = MEAN06_10, color = '3')) +
```

```
geom_line(aes(y = MEAN11_15, color = '4')) +
labs(title = 'Year Comparison',
     x = 'Days',
     y = 'Temperature',
     color = "Legend") +
theme(plot.title = element_text(hjust = 0.5)) +
scale_color_manual(name = 'Years',
                   values = c('1'='blue', '2'='yellow', '3'='orange', '4'='red'),
                   labels = c('1996-2000', '2001-2005', '2006-2010', '2011-2015'))
```



According to exponentially smoothed temperature curve averaged across 4 sets of 5-year periods, we see that while summer might unofficially end with a slightly higher temperature from 1996 to 2015, the result might not be significant enough to support the claim that summer is ending later. More detailed tests will be required.