Homework4
2023-09-19
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?

In my job as an analyst, we frequently work with historical daily stock price data to generate short-term forecasts. The stock prices often exhibit volatility and fluctuations from day-to-day that make it difficult to discern the underlying trends. Exponential smoothing could be useful here to smooth out some of this daily noise. Specifically, we could employ single exponential smoothing (SES) to forecast the closing price over a 1-week horizon. The data required would be a time-series of historical daily closing prices for a given stock. I would expect the smoothing parameter α to be relatively low, likely between 0.05 to 0.20. A lower α gives more weight to the historical data and less weight to the most recent observation. This helps filter out some of the daily fluctuations. An α close to 1 would react too quickly to daily changes. For example, for Apple stock, an α of 0.1 would be reasonable. This puts 10% weight on the most recent closing price, with the rest distributed exponentially across the historical data. With the frequent volatility in Apple, we don't want today's price to overly influence the forecast. We could then use SES with α=0.1 to generate 1-week ahead forecast prices. The smoothed trend should provide a less noisy forecast compared to just using the last observation. This can help guide short-term trading decisions and analysis of price momentum. For fluctuations over longer periods seasonality and trend becomes more relevant, especially for long term, buy and hold investors.

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).

Use longer HW code aplha = null, beta = null nd gama = null Multiplicative – expects large peaks and throughs Emphasize that there is no clear trend Determine why you used seasonality vs bseline for CUSUM You can plot the seasonaliy / bseline values for each year

```
library(IRkernel)
library(forecast)
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(ggplot2)
library(reshape)
# Load temps data
temps <- read.table('/Users/barovierallybose/Documents/Intro to analytics modeling /HW 4/hw4-FA23/temps.txt', header=TRUE)
# Convert a data frame to a time series object
# Extract data from columns 2 to 21 of the 'temps' data frame and convert it into a one-dimensional vector
data_vector <- as.vector(unlist(temps[, 2:21]))

# Create a time series object 'data' with a start year of 1996 and a frequency of 123
data <- ts(data_vector, start = c(1996, 1), frequency = 123)

# Using the 'plot.ts' function to create a time series plot
plot.ts(data, main = "Time Series Plot", xlab = "Time", ylab = "Value", ylim = c(50, 100))
```

So far I have converted the data into time series format, since the visualization shows yearly repeated patters then it would make sense to use the triple exponential smoothing method, which is also called winters method or Holt-winters method.
Though I expected temperatures to be roughly the same year after year during the summer months, I decided to explore between both the additive and multiplicative seasonal components to see how the data fares.

```
# Set a random seed for reproducibility
set.seed(1)

# Perform Holt-Winters forecasting using additive decomposition
HW_A <- HoltWinters(data, alpha=NULL, beta=NULL, gamma=NULL, seasonal = "additive")

# Create a plot of the Holt-Winters results
plot(HW_A)

# Set a random seed for reproducibility
set.seed(1)

# Perform Holt-Winters forecasting using multiplicative decomposition
HW_M <- HoltWinters(data, alpha=NULL, beta=NULL, gamma=NULL, seasonal = "multiplicative")
# Create a plot of the Holt-Winters results

plot(HW_M)

# Print results for the additive method
cat("Using additive method:\n")
## Using additive method:
cat("\tBase factor:", HW_A$alpha,"\n")      # Print the base factor (alpha)
```

```
##  Base factor: 0.6610618
cat("\tTrend factor:",HW_A$beta,"\n")       # Print the trend factor (beta)
##  Trend factor: 0
cat("\tSeasonal factor:",HW_A$gamma,"\n")   # Print the seasonal factor (gamma)
##  Seasonal factor: 0.6248076
cat("\tSum of Squared Errors:", HW_A$SSE)    # Print the Sum of Squared Errors
##  Sum of Squared Errors: 66244.25
# Print results for the multiplicative method
cat("\nFor multiplicative method:\n")
##
## For multiplicative method:
cat("\tBase factor:", HW_M$alpha,"\n")      # Print the base factor (alpha)
##  Base factor: 0.615003
cat("\tTrend factor:",HW_M$beta,"\n")       # Print the trend factor (beta)
##  Trend factor: 0
cat("\tSeasonal factor:",HW_M$gamma,"\n")   # Print the seasonal factor (gamma)
##  Seasonal factor: 0.5495256
cat("\tSum of Squared Errors:", HW_M$SSE)    # Print the Sum of Squared Errors
##  Sum of Squared Errors: 68904.57
```

In either the additive y(t) = Level(t) + Trend(t) + Seasonality(t) + Remainder(t) or the multiplicative model y(t) = Level(t) * Trend(t) * Seasonality(t) * Remainder(t):
Level (L(t)): Represents the baseline or average level of the time series. It captures the overall mean or baseline behavior of the data. Trend (T(t)): Represents any systematic, long-term increase or decrease in the data. It captures the direction and magnitude of a linear or nonlinear trend. Seasonality (S(t)): Represents the repeating patterns or cycles in the data that occur with a fixed frequency (e.g., daily, monthly, yearly). It captures seasonal fluctuations.
Nb. The xhat represents the raw(fitted) data

```
# Set up a layout with 2 rows and 1 column
par(mfrow = c(2, 1))

# Plot the fitted values from HW_A in the first row
plot(fitted(HW_A))

# Plot the fitted values from HW_M in the second row
plot(fitted(HW_M))
 #Decision is to use the additive method
# Extract and display the first ten fitted values from the additive Holt-Winters model
(HW_A)
first_ten_fitted <- HW_A$fitted[1:10,]
print(first_ten_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
## [7,] 75.06732 81.89860 -0.004362918 -6.826922
## [8,] 87.04284 81.84974 -0.004362918  5.197468
## [9,] 84.01829 81.81705 -0.004362918  2.205599
## [10,] 87.05875 81.80060 -0.004362918  5.262509
# Extracting seasonal factors from the additive Holt-Winters model (HW_A)
seasonal_factor <- matrix(HW_A$fitted[, 4], nrow = 123)
fitted <- matrix(HW_M$fitted[,1], nrow=123)

#Let's add the row and column names back into the dataset for ease of use.
colnames(seasonal_factor) <- colnames(temps[,3:21])
rownames(seasonal_factor) <- temps[,1]

colnames(fitted) <- colnames(temps[,3:21])
rownames(fitted) <- temps[,1]
avg_SF_yr <- vector()
for (i in 1:ncol(seasonal_factor)){
  avg_SF_yr[i] = mean(seasonal_factor[,i])
}
avg_SF_yr
## [1]  1.855414e-16 -5.311666e-02 -3.631361e-03 -1.493990e-02 -6.627846e-03
## [6] -2.497262e-02 -4.262708e-02 -7.328313e-03  7.848572e-03 -2.186189e-02
## [11] -2.050925e-02 -1.696484e-02 -2.664637e-02 -1.766348e-02 -1.557198e-03
## [16] -2.418875e-02 -2.499924e-02  9.649320e-03 -9.342363e-03
cusum_decrease = function(data, mean, T, C){
  results = list()
  cusum = 0
  rowCounter = 1
  while (rowCounter <= nrow(data)){
    current = data[rowCounter,]
    cusum = max(0, cusum + (mean - current - C))
    # print(cusum)
    if (cusum >= T) {
      results = rowCounter
      break
    }
    rowCounter = rowCounter + 1
    if (rowCounter >= nrow(data)){
      results = NA
      break
    }
  }
  return(results)
}
```

```
C = sd(seasonal_factor[,1])*0.5
T = sd(seasonal_factor[,1])*5

result_vector = vector()
for (col in 1:ncol(seasonal_factor)){
  result_vector[col] = cusum_decrease(data = as.matrix(seasonal_factor[,col]), mean = 1,
T, C)
}

result_df = data.frame(Year = colnames(seasonal_factor),Day = temps[result_vector,1])
result_df
##     Year   Day
## 1  X1997 1-Oct
## 2  X1998 2-Oct
## 3  X1999 2-Oct
## 4  X2000 2-Oct
## 5  X2001 4-Oct
## 6  X2002 5-Oct
## 7  X2003 5-Oct
## 8  X2004 6-Oct
## 9  X2005 6-Oct
## 10 X2006 6-Oct
## 11 X2007 6-Oct
## 12 X2008 6-Oct
## 13 X2009 7-Oct
## 14 X2010 6-Oct
## 15 X2011 6-Oct
## 16 X2012 6-Oct
## 17 X2013 6-Oct
## 18 X2014 7-Oct
## 19 X2015 7-Oct
```

After some exploratory data analysis, I have decided to use the additive method. Looking at the data overtime, the variability in fluctuations remains consistent. This makes sense for climate data, as I wouldn't expect any drastic change in climate within a 20 year span. While at first it could be argued that temperature data exhibits seasonal patterns that change in proportion to the overall level, for example, daily high temperatures in summer are a larger deviation from the annual mean compared to winter, we are looking at a segment of the entire year where temperatures are expected to be at one consistent level, particularly the Summer period, into early Fall. Within the summer season, the temperature fluctuations tend to be more constant day-to-day. For example, daily highs may vary by a fixed amount around a mean. The variability is more consistent day to day in the summer compared to the full year with winter. With a more stable base temperature level in summer, the additive deviations may be more stationary. The HoltWinters function in R makes this analysis easy to accomplish.

To support my logic, I carried out a comparison of the sum of squares between the two different models and the additive method returned a slightly lower sum of squares: 66244 vs 68905. Other things that are noteworthy from the additive method is that it gives more weighting to the base factor, meaning that the model will be more sensitive to recent changes in the data. Also, noteworthy, and can also be seen from the visualization, there is no discernible trend base on either model. And finally, making the additive method more ideal, it places a higher weighting on the seasonal aspect of the data.

But what determines the unofficial end of summer? For this analysis, we are going to assume that summer comes to an end as we begin to notice an effective change in temperatures, typically a drop below 80 degrees seems reasonable but a reasonable drop in temperature, around 4-5 degrees or greater, somewhere in the region at or below the 80 degrees mark, seems like a good proxy. Bsed on the results of my analysis, we noticed this in 1997, where there was a significant drop from 86 degrees to 75 degrees, >10 degrees drop, marking the unofficial start of the cooling period between September 30th and October 1st. And again in 2004, between October 5th and 6th, where temperatures dropped to 75 degrees, from 80 degrees the day before, and there are many more examples of these abrupt and sustained drops in temperature. This approach works in the analysis of the unofficial transition from summer to fall using the average seasonal factor of each year, where we are comparing subsequent years' seasonal factors to the first year's average. By establishing a threshold based on the first year's average seasonal factor for the transition period, I was able determine the unofficial end of summer when the subsequent years' seasonal factors consistently fall below this threshold. This approach was effective since there was a clear and consistent seasonal pattern for the transition.

Note that my model is built on the seasonal factor instead of the fitted values. The seasonal component directly captures the seasonal patterns in the data. This makes the summer effect more interpretable and explicit to analyze.

So using the additive method, and applying CUSUM to the seasonal factor, showed that summer got longer over the 20 year span, from Oct 1 in 1997 to Oct 7 in 2015. With the only notabe exception, it was only 4 years after 2009 that we saw a slight reversion back to a shorter summer by one day, but since then, the end of summer became later. Further testing could reveal if the difference over the 20 year span is statistically significant and perhaps a larger data set could add more robustness to our analysis. Some details to note about my CUSUM model, the C and T values were calculated from standard deviation of the first year, using 1996 as a baseline makes sense in the context of comparison with later data to see if there may be a discernible difference. C is calculated as .5 times the sd of the first year and T, 5 times. As temperature data can be noisy, exponential smoothing has shown to be a useful tool in reducing noise and improving predictions.