

Question 7.1

Exponential smoothing is a time series forecasting method for univariate data. Exponential smoothing forecasting methods are similar in that a prediction is a weighted sum of past observations, but the model explicitly uses an exponentially decreasing weight for past observations. Specifically, past observations are weighted with a geometrically decreasing ratio.

The exponential smoothing forecasting method can be applied to the hotel industry. Hotels need to look at trending data to determine the future hotel room reservation rate. The exponential smoothing forecasting could be used to analyze monthly or yearly booking trends to determine the sales price. I would use the yearly sales to forecast our annual performance and the monthly sales to track how we perform each month. I think the alpha changes base on the circumstances. If I want to predict the monthly sales, I will want the alpha close to 1 because I want the forecast to be based on previous records. However, in the case of a pandemic where nothing is predictable, I would want the alpha closer to 0 because there is more randomness and different circumstances in terms of sales. Because hotel bookings are definitely impacted by seasonality, I would say that seasonality and trend do play a role in the forecast.

HW 4

2022-09-20

As we've learned in this module, exponential smoothing allows for weighted averages where greater weight can be on recent observations and lesser weight on older observations. As always, the first step is to take a quick look on the data set.

```
# Basic Data Info
rm(list = ls())
temps <- read.table("/Users/xiaofanjiao/Desktop/temps.txt", stringsAsFactors = FALSE, header = TRUE)
head(temps)
```

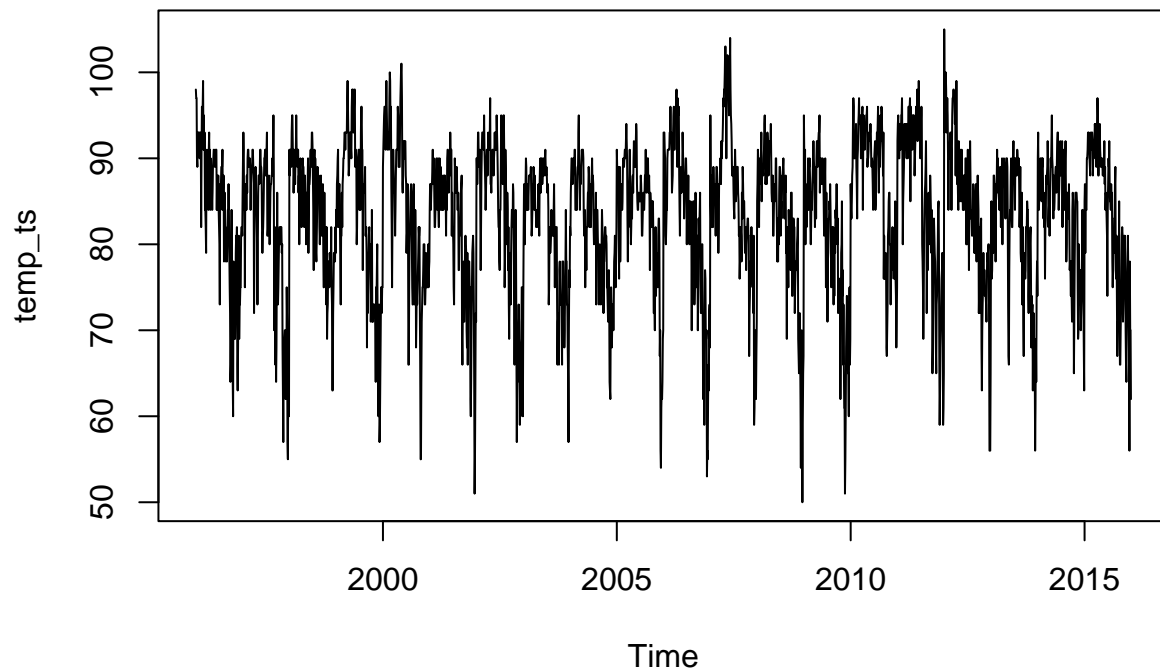
```
##      DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006 X2007
## 1 1-Jul   98    86    91    84    89    84    90    73    82    91    93    95
## 2 2-Jul   97    90    88    82    91    87    90    81    81    89    93    85
## 3 3-Jul   97    93    91    87    93    87   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

To create a timeseries, the first step is to create a temperature vector named `temps_vec`. I would concatenate all the columns. Then I would build the time series data named `temps_ts` and removing the first column. I would plot the data for visualization.

```
# Convert to time series
temp_vec <- as.vector(unlist(temps[,2:21]))
temp_ts <- ts(temp_vec, start = 1996, frequency = 123)
summary(temp_ts)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    50.00   79.00   85.00   83.34   90.00  105.00
```

```
plot(temp_ts)
```



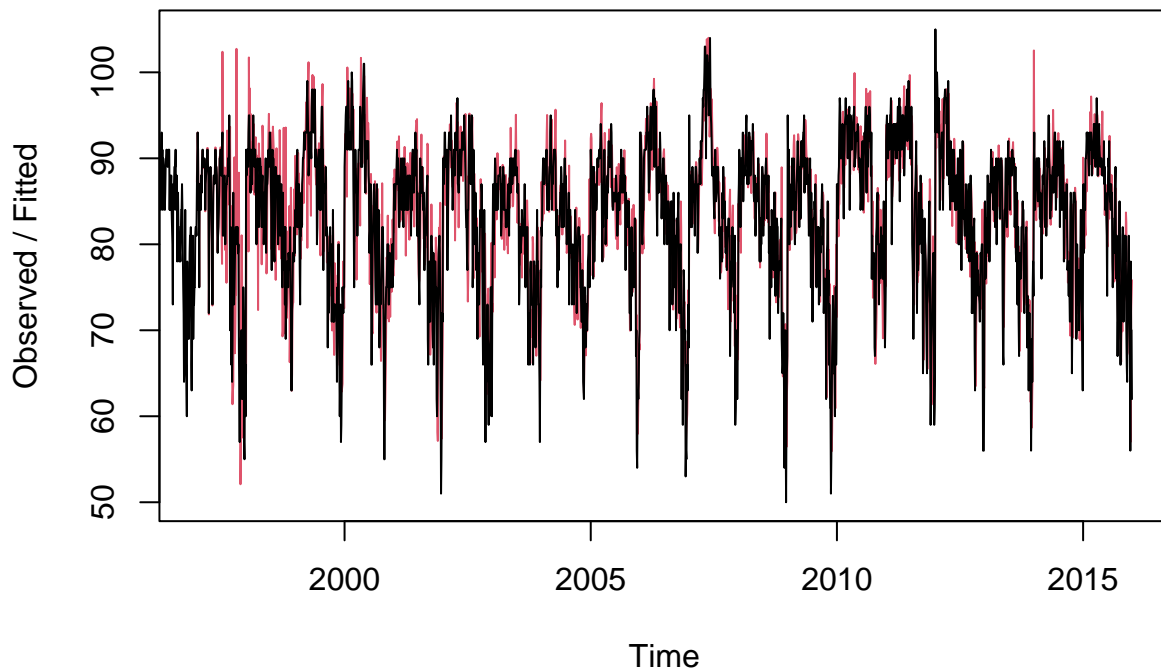
To perform forecasting, I used the `HoltWinters` function, with seasonality, and applied the exponential smoothing to the data. The seasonal option can be “additive” or “multiplicative”. When the seasonal trend is consistent in magnitude over the entire collection of data, the additive model performs well. On the other hand, when seasonality fluctuates in strength over time, the multiplicative model is favoured. For this assignment, I used “multiplicative” model. To have the model calculate certain parameters, such as alpha, beta, and gamma, NULL values are explicitly applied to them.

```
# Apply Holt
temps_HW <- HoltWinters(temp_ts, alpha = NULL, beta = NULL, gamma = NULL, seasonal = "multiplicative")
summary(temps_HW)
```

```
##           Length Class  Mode
## fitted      9348   mts    numeric
## x           2460    ts     numeric
## alpha         1  -none-  numeric
## beta          1  -none-  numeric
## gamma         1  -none-  numeric
## coefficients 125  -none-  numeric
## seasonal      1  -none-  character
## SSE           1  -none-  numeric
## call          6  -none-  call
```

```
plot(temps_HW)
```

Holt-Winters filtering



From the summary, we observe that α is equal to 1 which indicates less randomness in the prediction, therefore there are more trust towards our current observations. Beta and gamma are all equal to 1 as well so it is likely to find trends or cycle in the data. After plotting the data, we see that the Holt-Winters prediction (in red), is quite similar to our observed data (in black), especially on later cycles. This is because the model becomes more accurate at projecting future outcomes as more data points are utilised to develop the trends and seasonality components.

```
head(temps_HW$fitted)
```

```
##          xhat    level      trend   season
## [1,] 87.23653 82.87739 -0.004362918 1.052653
## [2,] 90.42182 82.15059 -0.004362918 1.100742
## [3,] 92.99734 81.91055 -0.004362918 1.135413
## [4,] 90.94030 81.90763 -0.004362918 1.110338
## [5,] 83.99917 81.93634 -0.004362918 1.025231
## [6,] 84.04496 81.93247 -0.004362918 1.025838
```

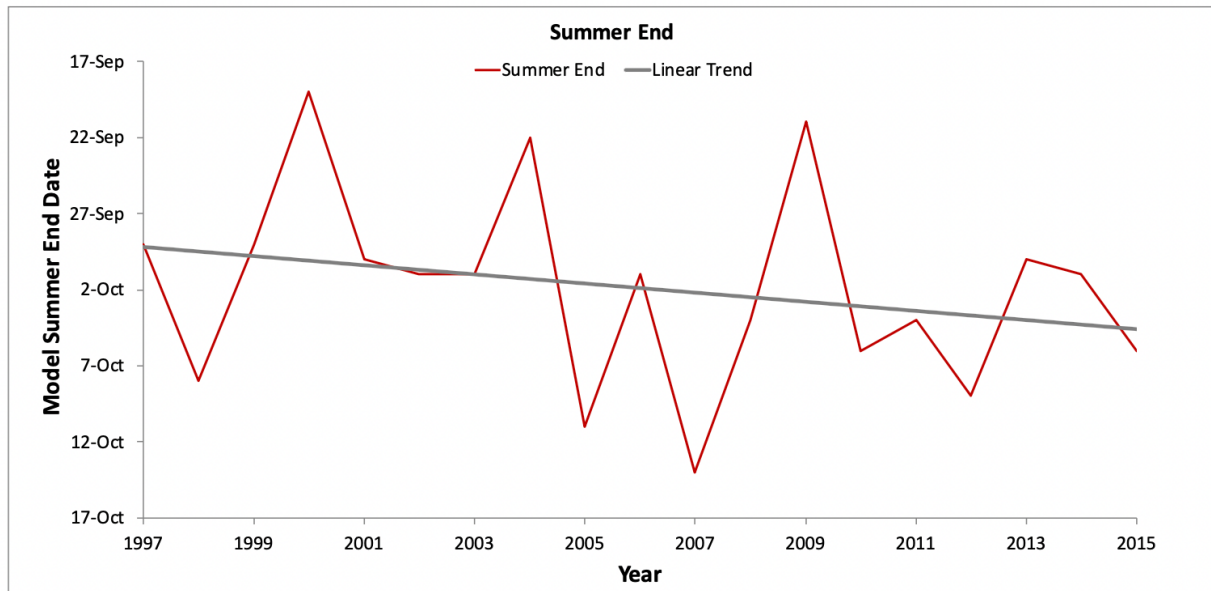
```
temps_HW_sf <- matrix(temps_HW$fitted[,4], nrow=123)
temp_HW_smoothed <- matrix(temps_HW$fitted[,1], nrow =123)
x <- data.frame(temp_HW_smoothed)
```

After the calculation of the smoothed data, I exported the file to excel to apply CUSUM on the data set.

```
#Export to Excel  
library(writexl)  
write_xlsx(x,"/Users/xiaofanjiao/Desktop/book1.xlsx")
```

Question 7.2

By applying the CUSUM model to this data and using $C = 6$ and $T = 45$, we can see that there is a slight trend that summer is ending a little later each year. The ending date of the summer fluctuates each year from late September to early October. However, this trend does not occur last very long as we see the next year the date often changes back and be closer to the trend line.



JUL Average	87.86475252
Std of July averages	2.090330792
C =	6
T =	45

