Question

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).

Solution

Step-1: Load and check data

```
#load data
temp_df <- read.table("temps.txt", header = TRUE)
head(temp_df)
```

```
> head(temp_df)
    DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006 X2007 X2008 X2009 X2010 X2011 X2012
1 1-Jul    98    86    91    84    89    84    90    73    82    91    93    95    85    95    87    92   105
2 2-Jul    97    90    88    82    91    87    90    81    81    89    93    85    87    90    84    94    93
3 3-Jul    97    93    91    87    93    87    87    87    86    86    93    82    91    89    83    95    99
4 4-Jul    90    91    91    88    95    84    89    86    88    86    91    86    90    91    85    92    98
5 5-Jul    89    84    91    90    96    86    93    80    90    89    90    88    88    80    88    90   100
6 6-Jul    93    84    89    91    96    87    93    84    90    82    81    87    82    87    89    90    98
  X2013 X2014 X2015
1    82    90    85
2    85    93    87
3    76    87    79
4    77    84    85
5    83    86    84
6    83    87    84
```

Step-2: Identify columns that contains year values from temp_df

```
year_cols <- grep("^X", colnames(temp_df))
years <- as.numeric(sub("X", "", colnames(temp_df)[year_cols]))
```

grep identifies column names that start with **'X'**.

Step-3: Convert years into a separate column by converting these values into vector.

```
#convert data frame to matrix, matrix to vector
temp_data <- as.vector(as.matrix(temp_df[, year_cols]))
```

Step-4: Remove missing (na) data. Printing the total observations and days per season (or year)

```
#remove null values
temp_data <- temp_data[!is.na(temp_data)]
cat("Total observations:", length(temp_data),"\n")
days_per_season <- nrow(temp_df)
cat("Days per season:", days_per_season,"\n")
```

```
> cat("Total observations:", length(temp_data),"\n")
Total observations: 2460
> days_per_season <- nrow(temp_df)
> cat("Days per season:", days_per_season,"\n")
Days per season: 123
```

Step-5: Converting data frame into time series

```
start_year <- min(years)
ts_temps <- ts(temp_data, frequency = days_per_season, start = c(start_year, 1))
```

Step-6(a): Perform Exponential Smoothing using Holt-Winters Additive model

```
#additive fit model
hw_additive <- HoltWinters(ts_temps, seasonal = "additive")
print("=== ADDITIVE MODEL ===")
print(hw_additive)
#Sum of Squared Error - Additive
cat("SSE:", hw_additive$SSE, "\n\n")
```

Sample Output:

```
[1] "=== ADDITIVE MODEL ==="
> print(hw_additive)
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = ts_temps, seasonal = "additive")

Smoothing parameters:
 alpha: 0.6610618
 beta : 0
 gamma: 0.6248076

Coefficients:
              [,1]
a     71.477236414
b     -0.004362918
s1    18.590169842
s2    17.803098732
s3    12.204442890
s4    13.233948865
s5    12.957258705
s6    11.525341233
s7    10.854441534
s8    10.199632666
s9     8.694767348
s10    5.983076192
...
s111 -12.837047727
s112  -9.095808127
s113  -5.433029341
s114  -6.800835107
s115  -8.413639598
s116 -10.912409484
s117 -13.553826535
s118 -10.652543677
s119 -12.627298331
s120  -9.906981556
s121 -12.668519900
s122  -9.805502547
s123  -7.775306633
> #Sum of Squared Error - Additive
> cat("SSE:", hw_additive$SSE, "\n\n")
SSE: 66244.25
```

Step-6(b): Perform Exponential Smoothing using Holt-Winters Multiplicative model

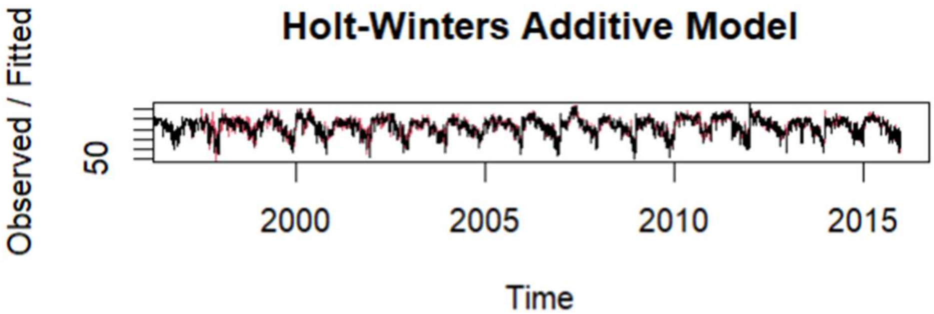
```
#multiplicative fit model
hw_multiplicative <- HoltWinters(ts_temps, seasonal = "multiplicative")
print("=== MULTIPLICATIVE MODEL ===")
print(hw_multiplicative)
#Sum of Squared Error - Multiplicative
cat("SSE:", hw_multiplicative$SSE, "\n\n")
```

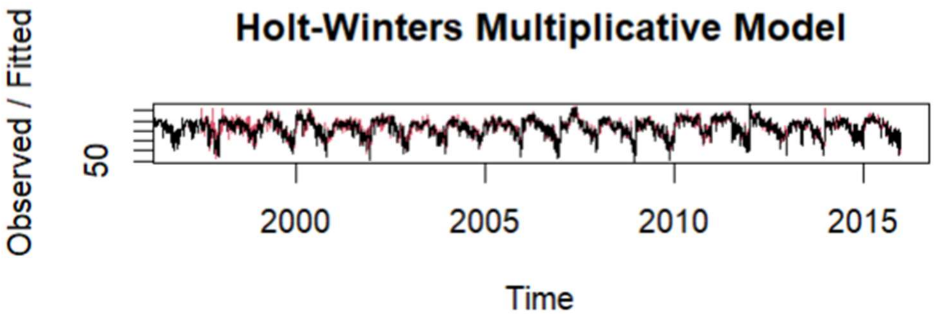
Sample Output:

```
> print("=== MULTIPLICATIVE MODEL ===")
[1] "=== MULTIPLICATIVE MODEL ==="
> print(hw_multiplicative)
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.

Call:
HoltWinters(x = ts_temps, 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
s6     1.151038408
s7     1.139383104
s8     1.130484528
s9     1.110487514
s10    1.076242879
…
s111   0.818367239
s112   0.863958784
s113   0.912057203
s114   0.898308248
s115   0.878723779
s116   0.848971946
s117   0.813891909
s118   0.846821392
s119   0.819121827
s120   0.851036184
s121   0.820416491
s122   0.851581233
s123   0.874038407
> #Sum of Squared Error - Multiplicative
> cat("SSE:", hw_multiplicative$SSE, "\n\n")
SSE: 68904.57
```

Step-7: Plot and compare both models visually.

```
#plotting both additive and multiplicative fit
#additive fit
par(mfrow = c(2, 1))
plot(hw_additive, main = "Holt-Winters Additive Model")
#multiplicative fit
plot(hw_multiplicative, main = "Holt-Winters Multiplicative Model")
par(mfrow = c(1, 1))
```





Step-8: Determine which model to use based on SSE (Sum of Squared Error). We use the model with lower SSE.

```
#compare additive and multiplicative fit
if(hw_additive$SSE < hw_multiplicative$SSE) {
  hw_model <- hw_additive
  model_type <- "ADDITIVE"
  cat("ADDITIVE model has lower SSE\n\n")
} else {
  hw_model <- hw_multiplicative
  model_type <- "MULTIPLICATIVE"
  cat("MULTIPLICATIVE model has lower SSE\n\n")
}
```

OUTPUT:

```
ADDITIVE model has lower SSE
```

Step-9: Extract trend component from the model

```
trend_component <- hw_model$fitted[,"trend"]
```

Step-10: Analyze last 50% of the data to find trend on when summer ends each year

```
temp_df2 <- data.frame(
  temp = temp_data,
  year = rep(years, each = days_per_season),
  day = rep(1:days_per_season, length(years))
)
```

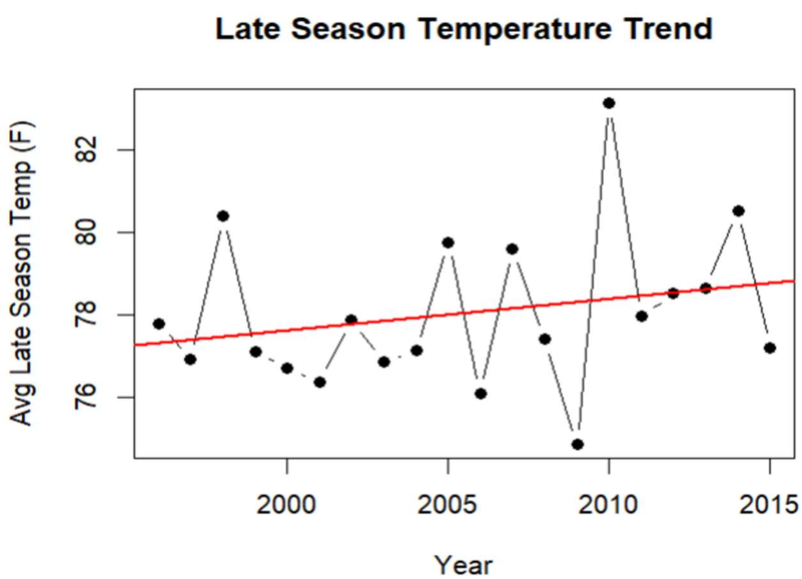
```
#focus on late season - last 50% of the season
late_day_start <- round(days_per_season * 0.5)
late_season <- temp_df2[temp_df2$day >= late_day_start, ]

#calculate average late season temp by year
late_season_avg <- aggregate(temp ~ year, data = late_season, mean)

#simple linear trend test
trend_test <- lm(temp ~ year, data = late_season_avg)
summary(trend_test)
```

OUTPUT:

```
Call:
lm(formula = temp ~ year, data = late_season_avg)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4660 -0.9261 -0.4531  0.7232  4.7638 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -75.74465  146.74263  -0.516    0.612
year          0.07668    0.07317   1.048    0.309

Residual standard error: 1.887 on 18 degrees of freedom
Multiple R-squared:  0.0575,	Adjusted R-squared:  0.005144 
F-statistic: 1.098 on 1 and 18 DF,  p-value: 0.3085
```

Step-11: Plot the trend- x-axis: year, y-axis: temperature

```
#plotting the trend
plot(late_season_avg$year, late_season_avg$temp,
     type = "b", pch = 19,
     xlab = "Year", ylab = "Avg Late Season Temp (F)",
     main = "Late Season Temperature Trend")
abline(trend_test, col = "red", lwd = 2)
```



Step-12: Calculate coefficient and p-value from trend. Conclude the analysis based on these values.

```
coeff_trend <- round(coef(trend_test)[2],3)
p_value <- round(summary(trend_test)$coefficients[2,4], 4)

cat("\n=== SUMMER END ANALYSIS ===\n")
cat("Slope of late season temperature trend:",coeff_trend, "°F per year\n")
cat("P-value:",p_value, "\n")

#analyzing the trend for conclusion
if(coeff_trend > 0 & p_value < 0.05) {
  print("Summer end appears to be getting LATER")
} else if(coeff_trend < 0 & p_value < 0.05) {
  print("Summer end appears to be getting EARLIER")
} else {
  print("No significant trend in summer end timing")
}
```

```
=== SUMMER END ANALYSIS ===
> cat("Slope of late season temperature trend:",coeff_trend, "°F per year\n")
Slope of late season temperature trend: 0.077 °F per year
> cat("P-value:",p_value, "\n")
P-value: 0.3085
> #analyzing the trend for conclusion
> if(coeff_trend > 0 & p_value < 0.05) {
+   print("Summer end appears to be getting LATER")
+ } else if(coeff_trend < 0 & p_value < 0.05) {
+   print("Summer end appears to be getting EARLIER")
+ } else {
+   print("No significant trend in summer end timing")
+ }
[1] "No significant trend in summer end timing"
```