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

Answer **7.1**

I want to analyze ice cream shop sales to better manage inventory and keep products fresh. Since sales are highly correlated with seasonal patterns, I plan to analyze five years of historical data to determine demand trends.

For this analysis, exponential smoothing would be appropriate because it helps identify long-term patterns while filtering out short-term fluctuations. Since sales follow a predictable seasonal trend, the smoothing parameter α should be closer to 0.

Choice of α:
For this scenario, α should be closer to 0 because:

1. **Strong Seasonality:** Ice cream sales are heavily influenced by seasonal patterns, which repeat year after year. A lower α gives more weight to historical data, ensuring the model captures these recurring seasonal trends effectively.
2. **Minor Trends:** While there may be minor trends (e.g., gradual growth in sales over the years), the primary driver of sales is seasonality. A lower α ensures the model focuses on the dominant seasonal patterns rather than overreacting to short-term fluctuations or minor trends.
3. **Stable Demand Patterns:** Since the seasonal changes are predictable and stable over time, a lower α helps smooth out noise and provides a more reliable forecast based on historical patterns.

This ensures that past data has a stronger influence on forecasts, preventing overreaction to minor daily or weekly variations. By focusing on long-term seasonal trends, I can make better stocking decisions and reduce waste.

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).

Answer 7.2

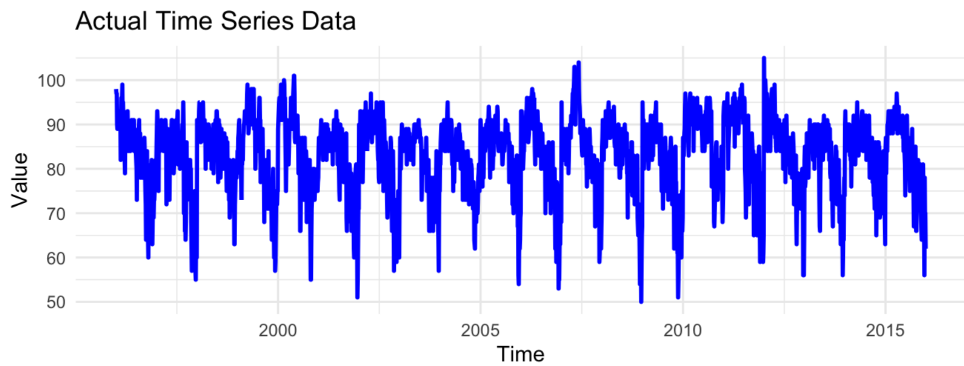
To determine if Atlanta's unofficial end of summer has shifted later over the past 20 years, I applied Holt-Winters exponential smoothing to daily high temperatures from July to October.

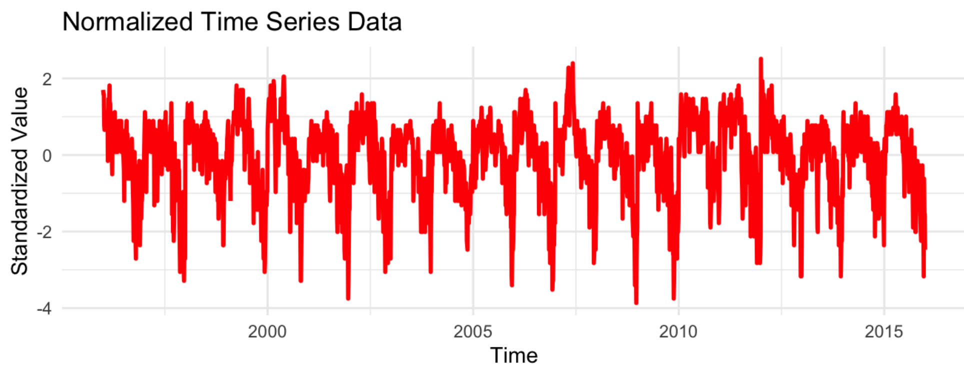
I tested three models:

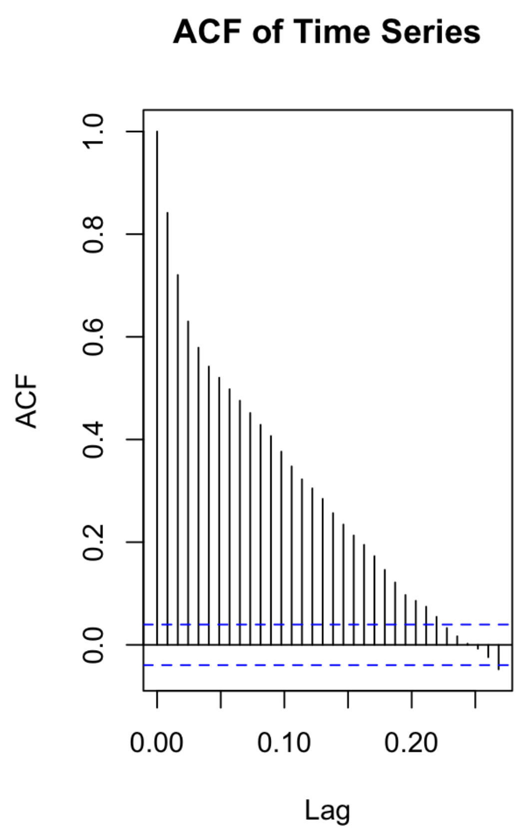
- Simple exponential smoothing (α): Captures only the level (short-term changes).
- Double exponential smoothing (Holt's method, α & β): Accounts for both level and trend.
- Triple exponential smoothing (Holt-Winters, α, β & γ): Incorporates seasonality in addition to level and trend.

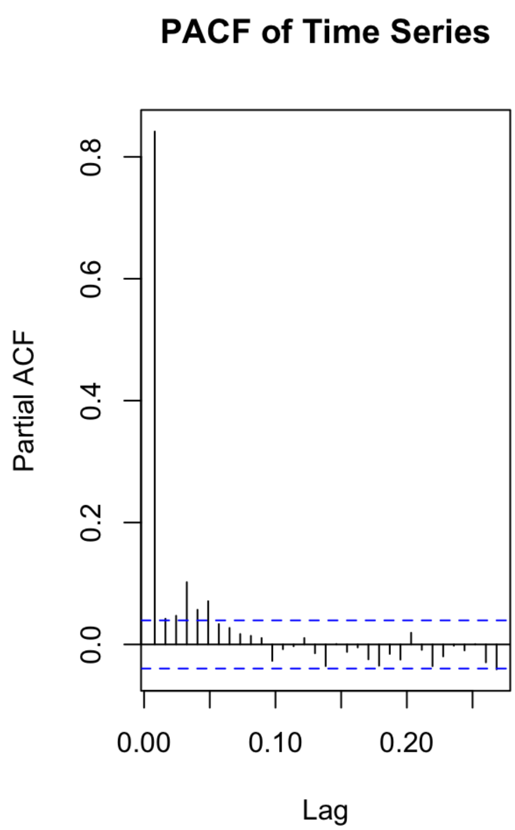
I tested single, double, and triple exponential smoothing models with both additive and multiplicative seasonality. In all cases, the trend component (β) remained close to zero (beta : 0.003720884 - 0), indicating no significant long-term increase in temperatures. While seasonal variations exist, the data does not support a shift in the end of summer over time.

Thus, based on this analysis, there is no strong evidence that summer is ending later than it did 20 years ago. Visualizing the smoothed data through plots helped confirm this, as no clear shift in seasonal patterns was observed.


Actual Time Series Data
Value
100
90
80
70
60
50
2000
2005
2010
2015
Time


Normalized Time Series Data
Standardized Value
2
0
-2
-4
2000
2005
2010
2015
Time


ACF of Time Series
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.10
0.20
Lag


PACF of Time Series
Partial ACF
0.8
0.6
0.4
0.2
0.0
0.00
0.10
0.20
Lag

----- R studio-----

```
# Set seed for reproducibility
> set.seed(1)
>
> # Define the file path
> file_path <- "~/Desktop/Master/Introduction to Analytics Modeling/HW4/hw4-SP22/temps.txt"
>
> # Check if the file exists and read the data
> if (!file.exists(file_path)) {
+     stop("Error: File not found. Check the path.")
+ }
>
> data <- read.table(file_path, header = TRUE)
> head(data)
    DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006 X2007 X2008 X2009
X2010
1 1-Jul    98    86    91    84    89    84    90    73    82    91    93    95    85    95    87
2 2-Jul    97    90    88    82    91    87    90    81    81    89    93    85    87    90    84
3 3-Jul    97    93    91    87    93    87    87    87    86    86    93    82    91    89    83
4 4-Jul    90    91    91    88    95    84    89    86    88    86    91    86    90    91    85
5 5-Jul    89    84    91    90    96    86    93    80    90    89    90    88    88    80    88
6 6-Jul    93    84    89    91    96    87    93    84    90    82    81    87    82    87    89
  X2011 X2012 X2013 X2014 X2015
1    92   105    82    90    85
2    94    93    85    93    87
3    95    99    76    87    79
4    92    98    77    84    85
5    90   100    83    86    84
6    90    98    83    87    84
>
> # Install necessary packages if not already installed
> packages <- c("gridExtra", "forecast", "ggplot2")
> new_packages <- packages[!(packages %in% installed.packages()[,"Package"])]
> if(length(new_packages)) install.packages(new_packages)
>
> # Load libraries
> library(ggplot2)
> library(gridExtra)
> library(forecast)
>
> # Convert data to time series
> data_vector <- as.vector(unlist(data[,2:21]))
> myts <- ts(data_vector, start = 1996, frequency = 123)
>
> # Apply Holt-Winters exponential smoothing models
> m1 <- HoltWinters(myts, beta = FALSE, gamma = FALSE)
> print(m1)
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = myts, beta = FALSE, gamma = FALSE)
```

```
Smoothing parameters:
 alpha: 0.8388021
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 63.30952
>
> m2 <- HoltWinters(myts, gamma = FALSE)
Warning in HoltWinters(myts, gamma = FALSE) :
  optimization difficulties: ERROR: ABNORMAL_TERMINATION_IN_LNSRCH
> print(m2)
Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = myts, gamma = FALSE)

Smoothing parameters:
 alpha: 0.8445729
 beta : 0.003720884
 gamma: FALSE

Coefficients:
        [,1]
a 63.2530022
b -0.0729933
>
> m3a <- HoltWinters(myts)
> print(m3a)
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = myts)

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
s11	3.123493477
s12	4.698228193
s13	2.730023168
s14	2.995935818
s15	1.714600919
s16	2.486701224
s17	6.382595268
s18	5.081837636
s19	7.571432660
s20	6.165047647
s21	9.560458487
s22	9.700133847
s23	8.808383245
s24	8.505505527
s25	7.406809208
s26	6.839204571
s27	6.368261304
s28	6.382080380
s29	4.552058253
s30	6.877476437
s31	4.823330209
s32	4.931885957
s33	7.109879628
s34	6.178469084
s35	4.886891317
s36	3.890547248
s37	2.148316257
s38	2.524866001
s39	3.008098232
s40	3.041663870
s41	2.251741386
s42	0.101091985
s43	-0.123337548
s44	-1.445675315
s45	-1.802768181
s46	-2.192036338
s47	-0.180954242
s48	1.538987281
s49	5.075394760
s50	6.740978049
s51	7.737089782
s52	8.579515859
s53	8.408834158
s54	4.704976718
s55	1.827215229
s56	-1.275747384
s57	1.389899699

s58	1.376842871
s59	0.509553410
s60	1.886439429
s61	-0.806454923
s62	5.221873550
s63	5.383073482
s64	4.265584552
s65	3.841481452
s66	-0.231239928
s67	0.542761270
s68	0.780131779
s69	1.096690727
s70	0.690525998
s71	2.301303414
s72	2.965913580
s73	4.393732595
s74	2.744547070
s75	1.035278911
s76	1.170709479
s77	2.796838283
s78	2.000312540
s79	0.007337449
s80	-1.203916069
s81	0.352397232
s82	0.675108103
s83	-3.169643942
s84	-1.913321175
s85	-1.647780450
s86	-5.281261301
s87	-5.126493027
s88	-2.637666754
s89	-2.342133004
s90	-3.281910970
s91	-4.242033198
s92	-2.596010530
s93	-7.821281290
s94	-8.814741200
s95	-8.996689798
s96	-7.835655534
s97	-5.749139155
s98	-5.196182693
s99	-8.623793296
s100	-11.809355220
s101	-13.129428554
s102	-16.095143067
s103	-15.125436350
s104	-13.963606549
s105	-12.953304848
s106	-16.097179844
s107	-15.489223470
s108	-13.680122300
s109	-11.921434142

```
s110 -12.035411347
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
>
> m3m <- HoltWinters(myts, seasonal = "multiplicative")
> print(m3m)
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.

Call:
HoltWinters(x = myts, 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
s11   1.041044609
s12   1.058139281
s13   1.032496529
s14   1.036257448
s15   1.019348815
s16   1.026754142
s17   1.071170378
s18   1.054819556
s19   1.084397734
s20   1.064605879
```

s21	1.109827336
s22	1.112670130
s23	1.103970506
s24	1.102771209
s25	1.091264692
s26	1.084518342
s27	1.077914660
s28	1.077696145
s29	1.053788854
s30	1.079454300
s31	1.053481186
s32	1.054023885
s33	1.078221405
s34	1.070145761
s35	1.054891375
s36	1.044587771
s37	1.023285461
s38	1.025836722
s39	1.031075732
s40	1.031419152
s41	1.021827552
s42	0.998177248
s43	0.996049257
s44	0.981570825
s45	0.976510542
s46	0.967977608
s47	0.985788411
s48	1.004748195
s49	1.050965934
s50	1.072515008
s51	1.086532279
s52	1.098357400
s53	1.097158461
s54	1.054827180
s55	1.022866587
s56	0.987259326
s57	1.016923524
s58	1.016604903
s59	1.004320951
s60	1.019102781
s61	0.983848662
s62	1.055888360
s63	1.056122844
s64	1.043478958
s65	1.039475693
s66	0.991019224
s67	1.001437488
s68	1.002221759
s69	1.003949213
s70	0.999566344
s71	1.018636837
s72	1.026490773

```
s73   1.042507768
s74   1.022500795
s75   1.002503740
s76   1.004560984
s77   1.025536556
s78   1.015357769
s79   0.992176558
s80   0.979377825
s81   0.998058079
s82   1.002553395
s83   0.955429116
s84   0.970970220
s85   0.975543504
s86   0.931515830
s87   0.926764603
s88   0.958565273
s89   0.963250387
s90   0.951644060
s91   0.937362688
s92   0.954257999
s93   0.892485444
s94   0.879537700
s95   0.879946892
s96   0.890633648
s97   0.917134959
s98   0.925991769
s99   0.884247686
s100  0.846648167
s101  0.833696369
s102  0.800001437
s103  0.807934782
s104  0.819343668
s105  0.828571029
s106  0.795608740
s107  0.796609993
s108  0.815503509
s109  0.830111282
s110  0.829086181
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
>
```

```
> # Extract fitted values
> m <- matrix(m3m$fitted[,4], ncol = 123)
>
> # Create time series data frame for visualization
> ts_df <- data.frame(
+     Time = as.numeric(time(myts)),
+     Actual = as.numeric(myts),
+     Normalized = as.numeric((myts - mean(myts)) / sd(myts))
+ )
>
> # Plot actual and normalized time series data
> p_actual <- ggplot(ts_df, aes(x = Time, y = Actual)) +
+     geom_line(color = "blue", linewidth = 1) +
+     ggtitle("Actual Time Series Data") +
+     xlab("Time") +
+     ylab("Value") +
+     theme_minimal()
>
> p_normalized <- ggplot(ts_df, aes(x = Time, y = Normalized)) +
+     geom_line(color = "red", linewidth = 1) +
+     ggtitle("Normalized Time Series Data") +
+     xlab("Time") +
+     ylab("Standardized Value") +
+     theme_minimal()
>
> # Arrange plots
> grid.arrange(p_actual, p_normalized, ncol = 1)
>
> # Decompose time series
> decomp <- decompose(myts, type = "additive")
> autoplot(decomp) + ggtitle("Decomposition of Time Series")
>
> # ACF and PACF plots
> par(mfrow = c(1,2))
> acf(myts, main = "ACF of Time Series")
> pacf(myts, main = "PACF of Time Series")
> par(mfrow = c(1,1))
```