

Homework_4

Claire Kraft

2024-09-18

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?

One of my hobbies is distance running. I run casually but I would like to use exponential smoothing to help me create a training program. This way I can increase my mileage gradually and not injure myself. I would need to import all my running records from Garmin or Nike Run tracking apps from at least a year ago. Maybe I can import in the last 5 years of data. In addition to the miles, some other columns can include elevation, weather (temperature), and pace.

In terms of exponential smoothing I can see if there are seasonal patterns where I am running more efficiently than other seasons. I know that I tend to run faster in the cold. So using the pace + distance / mileage i can figure out my efficiency. Exponential smoothing could observe my efficiency in the winter is better than the summer. For the Holt linear trend model maybe it can detect my distance accrual as I get into the consistent habit of running. Running is a sport that is easy to fall out of and easy to get back into shape.

The alpha value determines the weight on observations. It ranges from 0 and 1. When the alpha is closer to 0 then the data is more random. When the alpha is closer to 1 then the data is less random. When there is more randomness then trust the previous/historically estimates. If the randomness is less then trust what you see currently/recently. In other words if my running data is *mostly stable* with a some fluctuations (*less random*) over the years then I should pick an alpha closer to 1. If my running data is *mostly inconsistent*, with many variances, (*more random*) then i should choose an alpha closer to 0.

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).

The additive and basic holt winter models had the lowest (and same) sum of squared errors. After smoothing the additive model i exported the data for cusum. It seems summer ends around the 76th out of 123rd day.

I set 12.82 as my baseline mean (μ). For critical values, I wanted to use 2 standard deviations away. According to the normal bell curve principle, most data falls within ± 2 standard deviations, covering about 95% of the data[5]. Therefore, I used the formula $k=2*\text{std dev}$ for my critical value. For the thresholds, I calculated an upper

threshold (UH) and a lower threshold (LH). The upper threshold formula is $uh = \text{mean} + k$ and the lower threshold formula is $lh = \text{mean} - k$.

```
# for all code snippets look at reference 1
#----- Load and explore data
rm(list = ls())

temps_df <- read.table("C://Users//Clair//OneDrive//Documents//GitHub//omsa//ISYE 6501//Homework
4//temps.txt", header=TRUE)
#temps_df

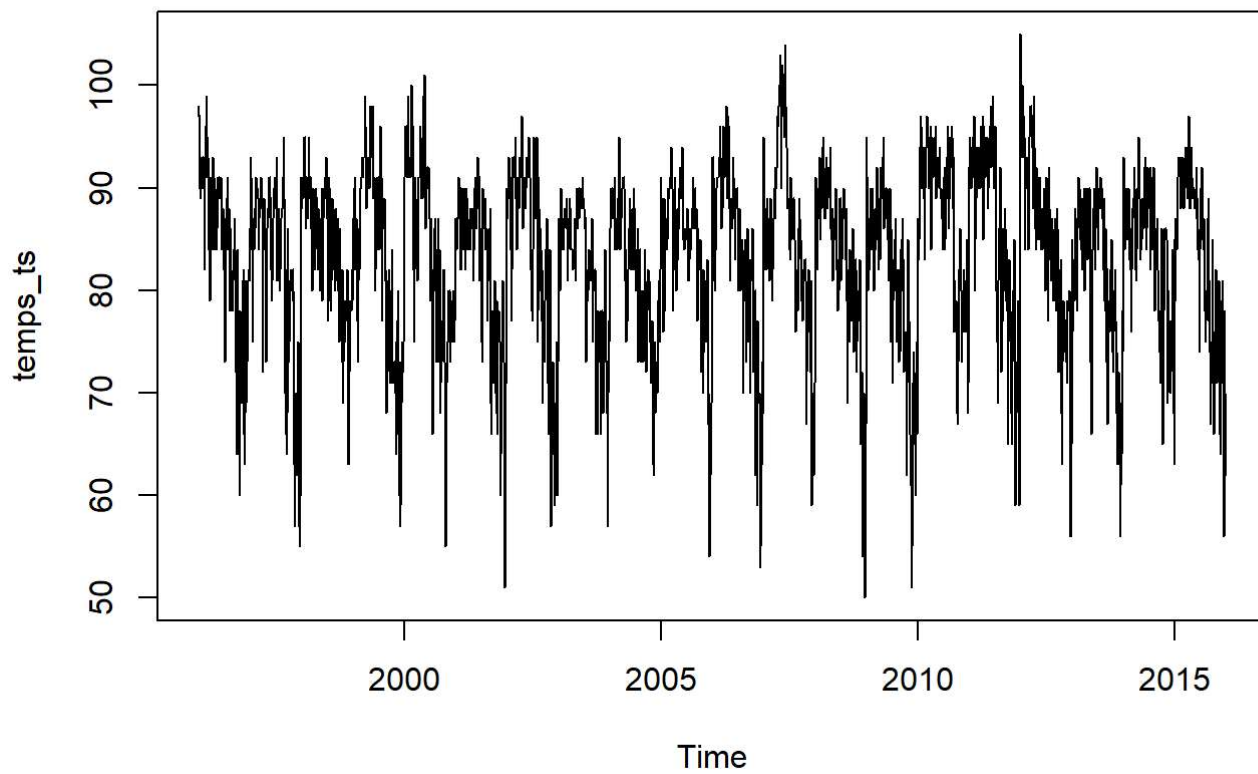
# Convert the data frame to a time series vector
temps_vec <- as.vector(unlist(temps_df[,2:21])) # excluding the first date column
#temps_vec
tail(temps_vec)
```

```
## [1] 67 56 78 70 70 62
```

```
# Convert the vector to a time series object
temps_ts <- ts(temps_vec, start=1996, frequency=123) # starting from 1996 with 123 observations
#temps_ts
tail(temps_ts)
```

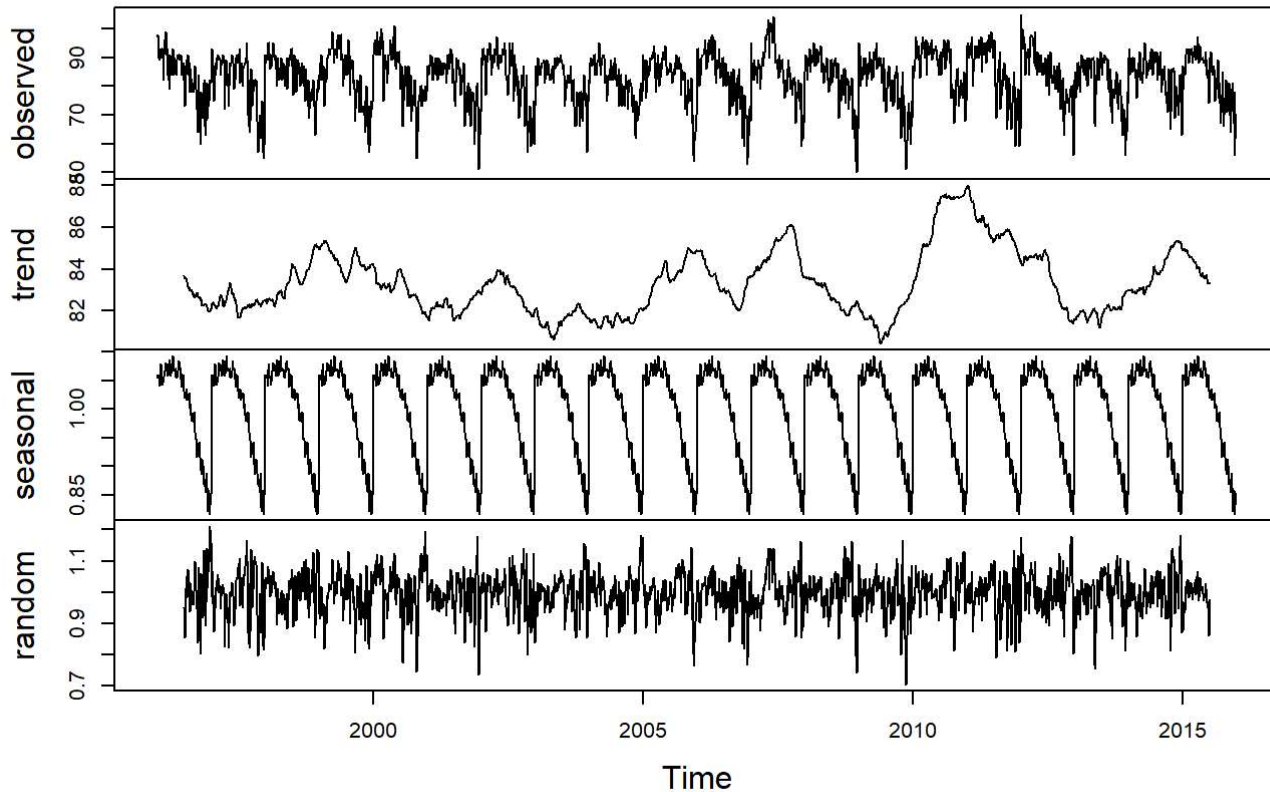
```
## [1] 67 56 78 70 70 62
```

```
# Plot the time series data
plot(temps_ts)
```



```
# Decompose the time series into its components (trend, seasonal, and random)
plot(decompose(temps_ts, type = "mult"))
```

Decomposition of multiplicative time series



```
# Holt winters model
# multiplicative
?HoltWinters
```

```
## starting httpd help server ... done
```

```
temp_holt_basic <- HoltWinters(temps_ts, alpha = NULL, beta = NULL, gamma = NULL)
temp_holt_basic$alpha
```

```
##      alpha
## 0.6610618
```

```
temp_holt_basic$beta
```

```
## beta
##      0
```

```
temp_holt_basic$gamma
```

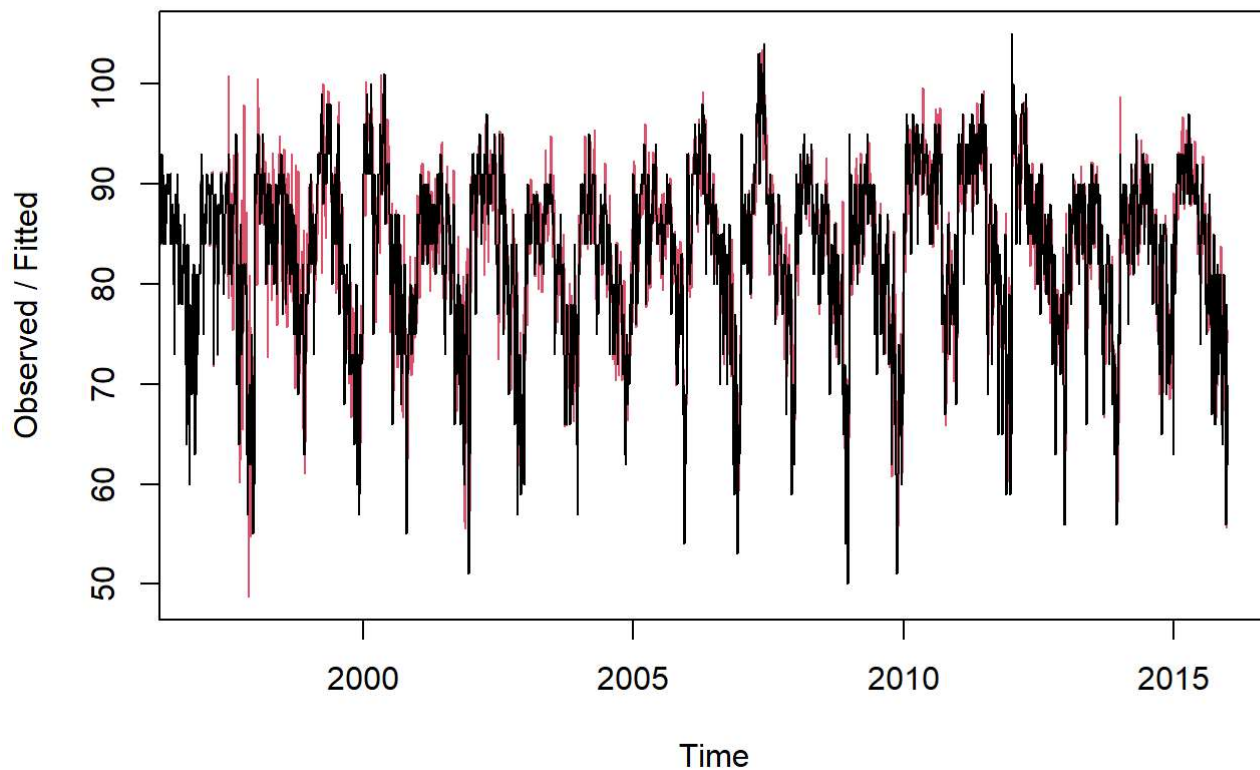
```
##      gamma
## 0.6248076
```

```
temp_holt_basic$SSE # 66244.25
```

```
## [1] 66244.25
```

```
plot(temp_holt_basic)
```

Holt-Winters filtering



```
# Holt winters model
# additive
?HoltWinters
temp_holt_a <- HoltWinters(temps_ts, alpha = NULL, beta = NULL, gamma = NULL, seasonal = "additive")
temp_holt_a$alpha
```

```
##      alpha
## 0.6610618
```

```
temp_holt_a$beta
```

```
## beta
## 0
```

```
temp_holt_a$gamma
```

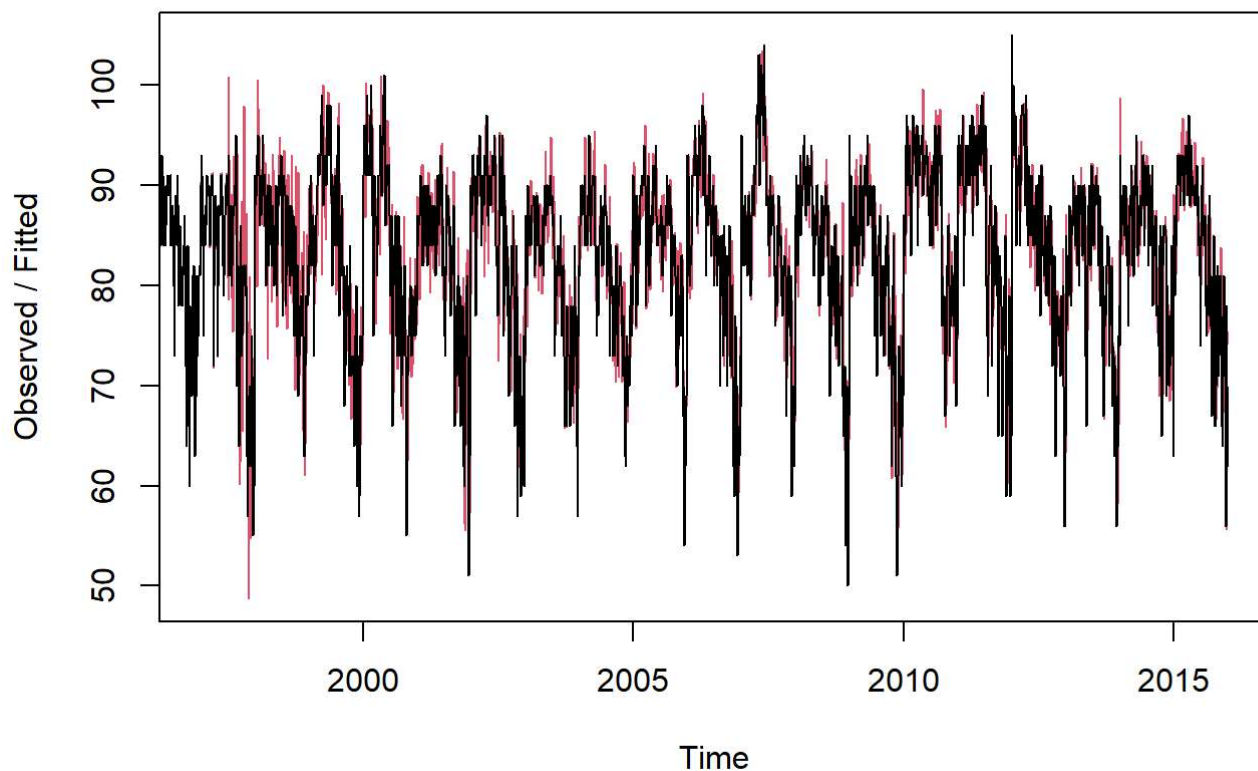
```
## gamma
## 0.6248076
```

```
temp_holt_a$SSE # 66244.25
```

```
## [1] 66244.25
```

```
plot(temp_holt_a)
```

Holt-Winters filtering



```
# Holt winters model
#
?HoltWinters
temp_holt_m <- HoltWinters(temps_ts, alpha = NULL, beta = NULL, gamma = NULL, seasonal = "multiplicative")
temp_holt_m$alpha
```

```
##    alpha  
## 0.615003
```

```
temp_holt_m$beta
```

```
## beta  
##    0
```

```
temp_holt_m$gamma
```

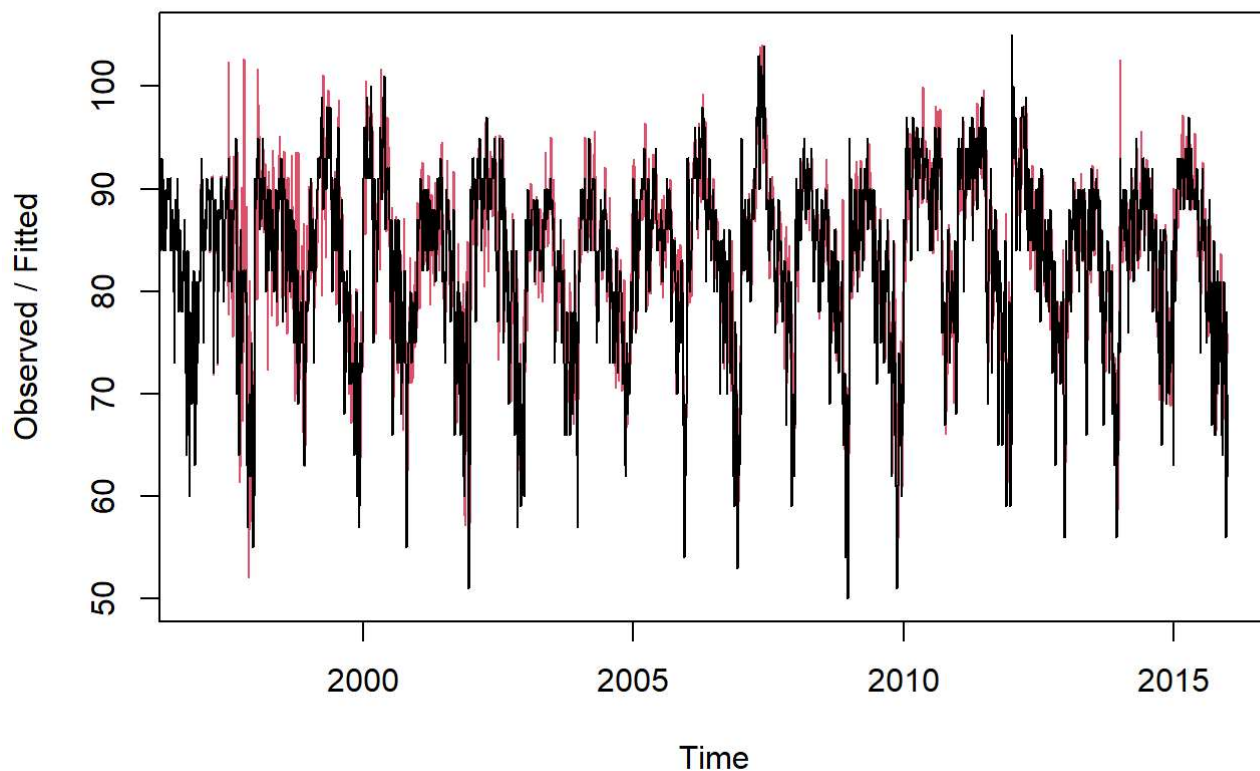
```
##    gamma  
## 0.5495256
```

```
temp_holt_m$SSE # 68904.57
```

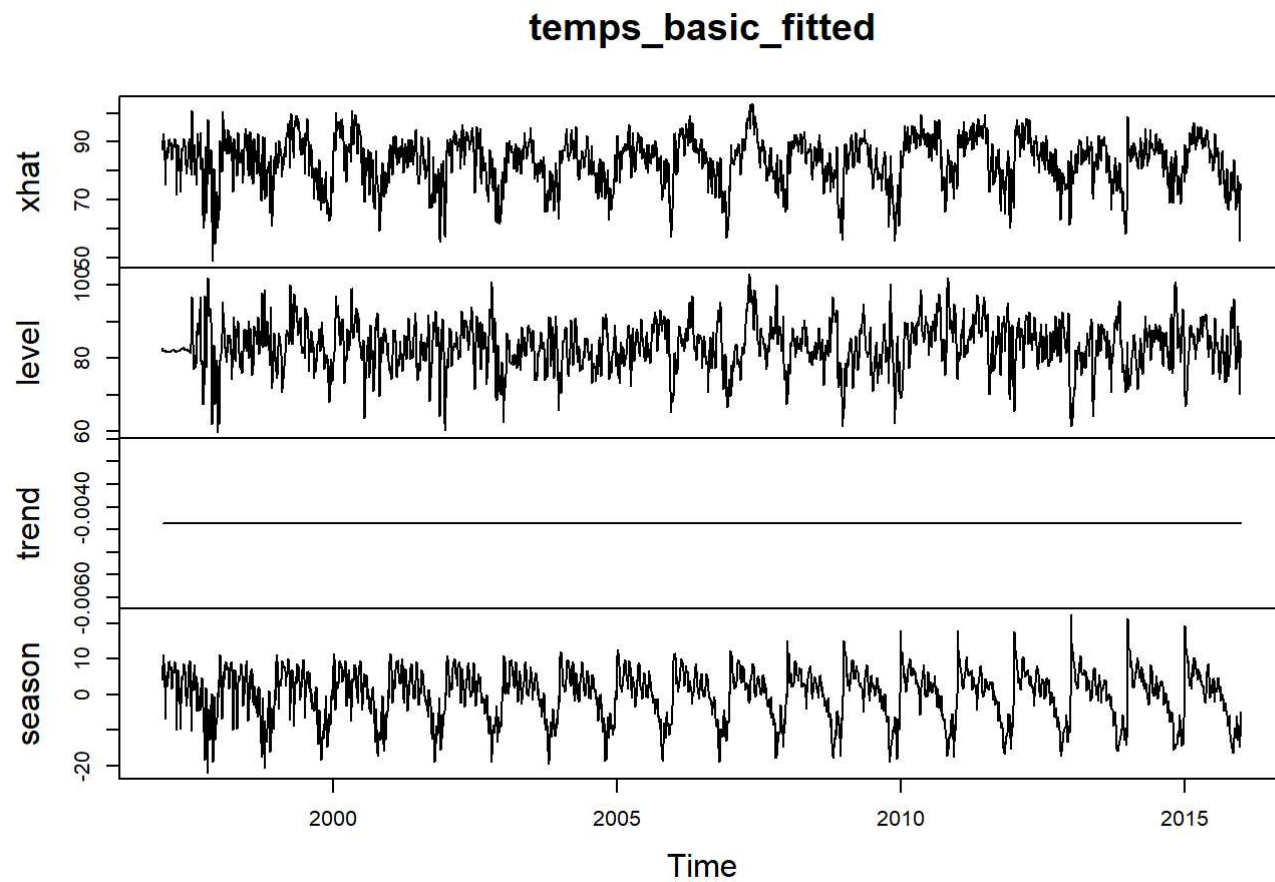
```
## [1] 68904.57
```

```
plot(temp_holt_m)
```

Holt-Winters filtering

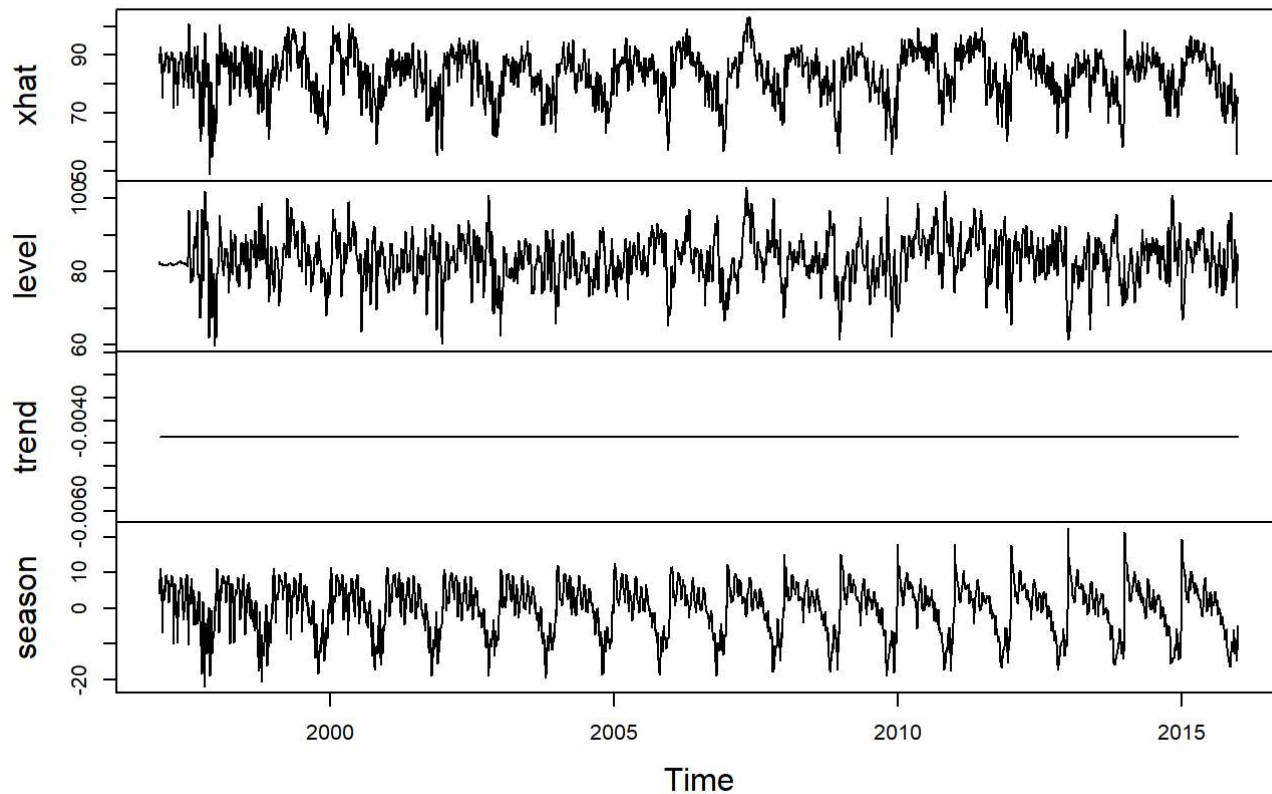


```
temps_basic_fitted = temp_holt_basic$fitted  
plot(temps_basic_fitted)
```



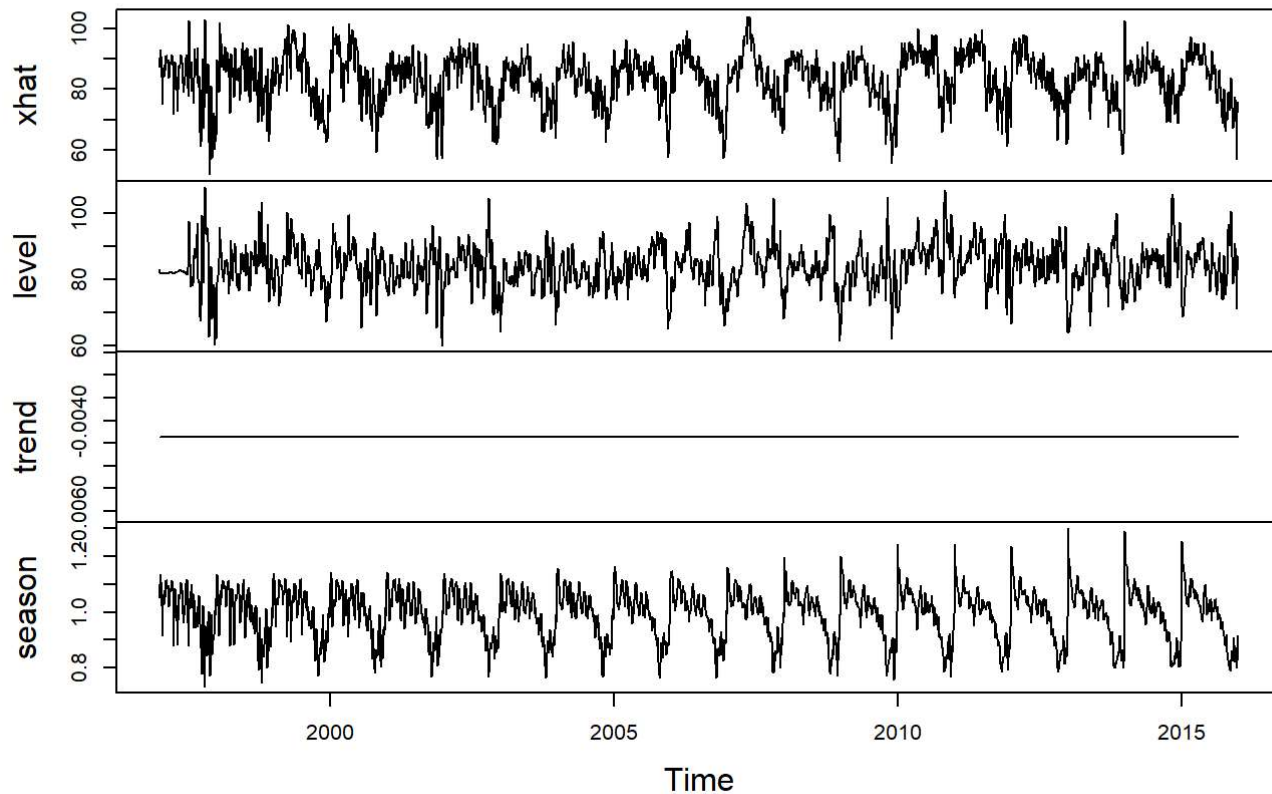
```
temps_additive_fitted = temp_holt_a$fitted  
plot(temps_additive_fitted)
```


temps_additive_fitted



```
temps_multiplicative_fitted = temp_holt_m$fitted  
plot(temps_multiplicative_fitted)
```

temps_multiplicative_fitted

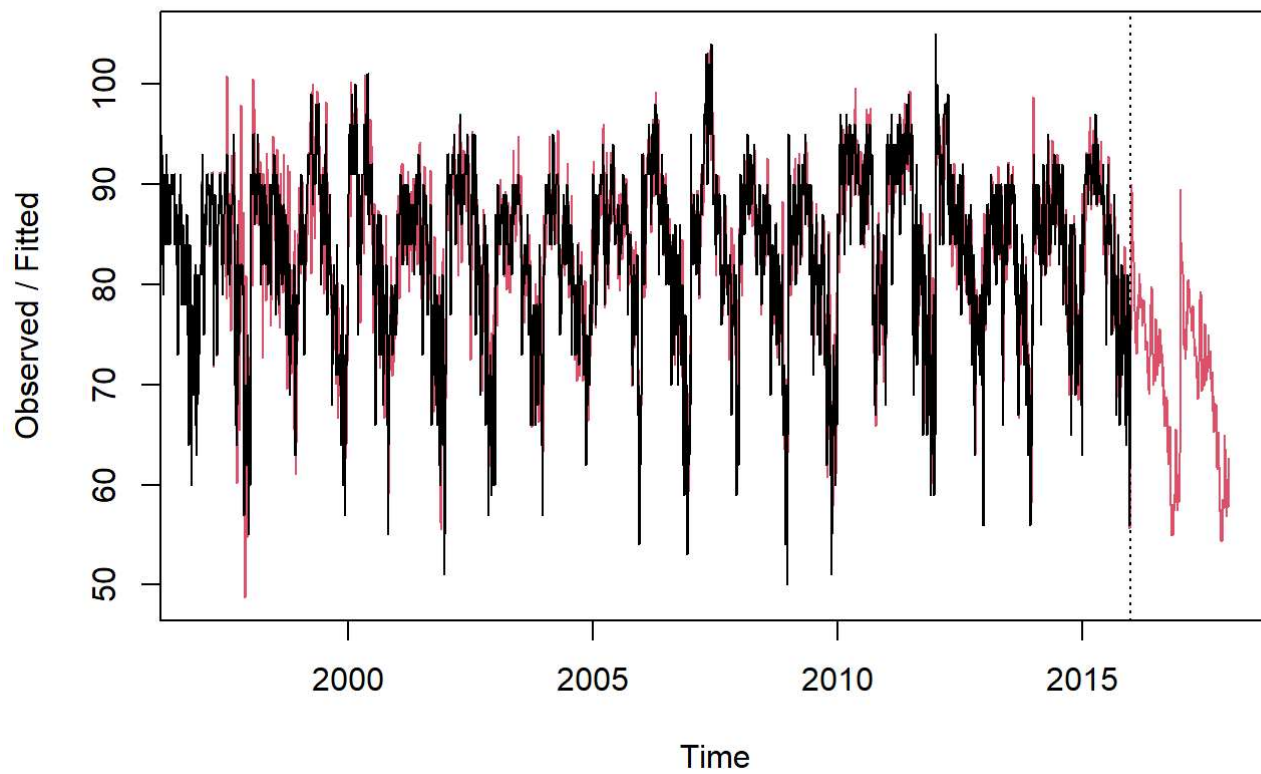


```
temp_smooth <- matrix(temp_holt_basic$fitted[,4], nrow=123)
tail(temp_smooth)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [118,] -6.4935885 -6.890605 -5.6558405 -4.4347630 -5.3063651 -7.98863484
## [119,] -6.4854584 -9.585617 -8.5954582 -6.2875108 -5.5543340 -9.16304674
## [120,] -0.5342389 -3.267917 -4.2692969 -4.8258201 -4.4623961 -5.06875912
## [121,]  0.4332408  1.208673 -0.2895929 -1.5318538 -3.1641748 -1.94952674
## [122,]  0.4088506  1.101306  1.4642322  1.3080103  0.5778192  2.10392807
## [123,] -0.6480600 -1.459240 -1.4283706 -0.8678262 -0.4418413  0.08050285
##           [,7]      [,8]      [,9]      [,10]     [,11]     [,12]
## [118,] -6.7774517 -8.4837323 -8.1417726 -7.33985781 -6.7622484 -6.67570860
## [119,] -8.2911287 -9.3950601 -8.6616976 -7.85532651 -6.7023824 -6.89657593
## [120,] -4.3687338 -7.0550231 -7.9364006 -8.02753487 -8.2346789 -7.33976472
## [121,] -2.3719010 -0.9513094 -3.1770223 -3.79134413 -2.8517916 -5.38165712
## [122,] -0.3791102  1.1636893  0.5976518  0.64984731  0.2404763 -0.21206523
## [123,] -2.0258362 -1.5768118 -0.3402956  0.08872082 -0.9891323 -0.03411151
##           [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## [118,] -6.7927592 -8.4238940 -7.9411253 -8.7304960 -9.50699113 -9.707349
## [119,] -9.2180348 -10.3152173 -11.6560776 -10.7124392 -12.67269269 -10.375104
## [120,] -9.9377581 -7.8268114 -9.2308714 -10.4825572 -14.15951727 -14.123842
## [121,] -4.7700225 -4.3009108 -6.7931879 -10.9093072 -12.06427234 -10.165486
## [122,]  0.1720216 -2.6205780 -3.1849145 -4.9196868 -6.57865384 -6.460541
## [123,]  0.4828511  0.1067451  0.1859101 -0.2679054  0.09155862 -2.127128
##           [,19]
## [118,] -8.684994
## [119,] -9.886253
## [120,] -14.645752
## [121,] -12.026808
## [122,] -8.933947
## [123,] -4.941084
```

```
plot(temp_holt_a, predict(temp_holt_a, n.ahead=246))
```

Holt-Winters filtering



```
write.csv(temp_smooth, file="temp_smoothed_export.csv", fileEncoding="UTF-16LE")
```

References:

[1] Pin Hsu TA presentation