ISYE Homework 4

2024-09-17

R Markdown

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?

Exponential smoothing would be useful in tracking and predicting personal spending over time. Data needed would be daily or weekly spending and a look at seasonal trends (spending near holidays/certain times of year/if I travel). It would smooth out random fluctuations and provide and clearer view of the overall trend. The spending pattern may shift slowly since ES would capture gradual changes without being overly influenced by one-time big purchases. The model would also account for seasonal changes in spending (ie. Christmas. Alpha would most likely be closer to 0 since spending usually does not change drastically day to day. Lower alpha values ensure the model does not overreact to a single large purchase and would focus more on longer term trends. This would aid in managing finances more efficiently and provide a smoother/more reliable prediction of future spending habits.

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).

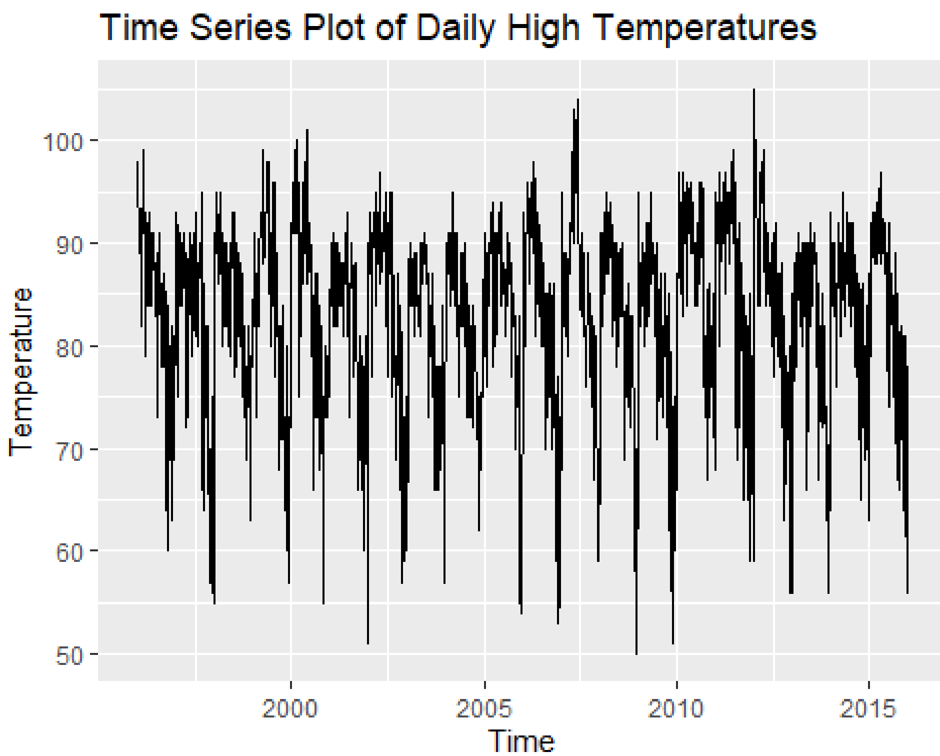
Analysis: The Holt-Winters forecasting algorithm enables users to smooth a time series and generate forecasts for key areas. It applies exponential smoothing, which assigns progressively lower weights to older data, giving more emphasis to recent data. This means that more recent observations have a greater influence on the forecast compared to older data points.The overall goal is to explore the historical temperature data and determine which Holt-Winters model (additive or multiplicative) better captures trends and seasonal fluctuations, potentially to assess whether summer is ending later over the years.

First thing, as always, is to load the libraries and import the data. I use Rio to import data easier and call in some other libraries for presentation purposes.

```
#housekeeping
library(pacman)
pacman::p_load(rio, ggplot2, ggfortify, tidyverse, reshape, survMisc,
forecast, lubridate)


temps <- import("D:/…/temps.txt")
head(temps) #preview data

##       V1  V2  V3  V4  V5  V6  V7  V8  V9  V10  V11  V12  V13  V14  V15  V16  V17  V18  V19
V20 V21
## 1 1-Jul 98 86 91 84 89 84 90 73  82  91  93  95  85  95  87  92 105  82
90  85
## 2 2-Jul 97 90 88 82 91 87 90 81  81  89  93  85  87  90  84  94  93  85
93  87
## 3 3-Jul 97 93 91 87 93 87 87 87  86  86  93  82  91  89  83  95  99  76
87  79
## 4 4-Jul 90 91 91 88 95 84 89 86  88  86  91  86  90  91  85  92  98  77
84  85
## 5 5-Jul 89 84 91 90 96 86 93 80  90  89  90  88  88  80  88  90 100  83
86  84
## 6 6-Jul 93 84 89 91 96 87 93 84  90  82  81  87  82  87  89  90  98  83
87  84
```

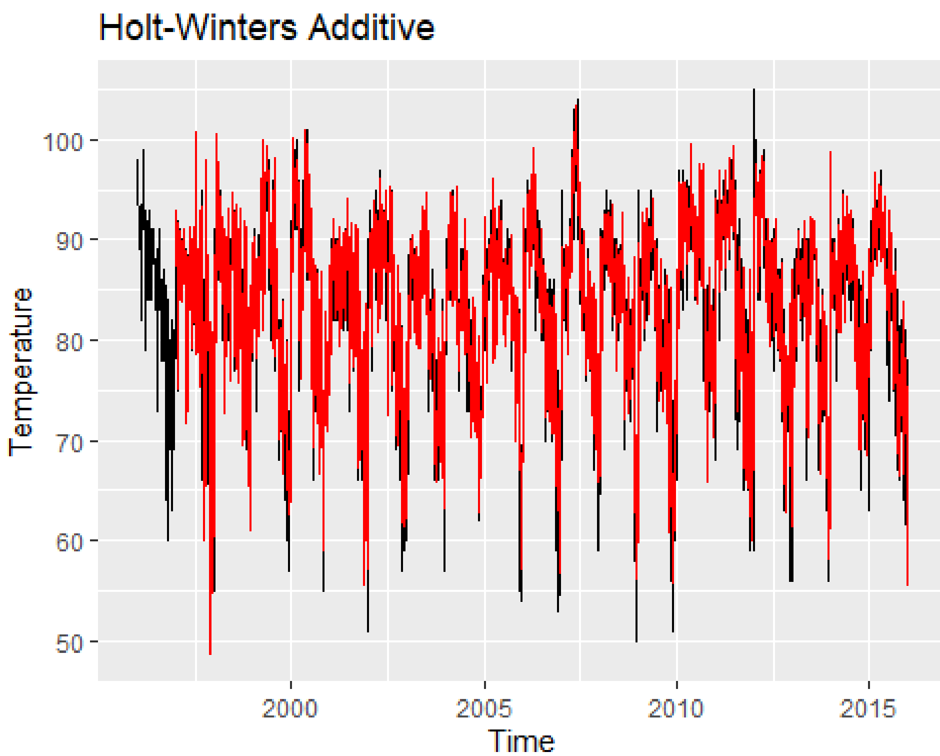
Now that the data has been imported, I will turn it into a time series object and visualize it using autoplot. This gives me a visual overview of the trends, seasonality, and any apparent shifts in the temperature over time.

```
#turn data into time series object
temp_ts<-ts(as.vector(unlist(temps[,2:21])),start=1996,frequency=123)
autoplot(temp_ts) +
  ggtitle("Time Series Plot of Daily High Temperatures") +
  xlab("Time") +
  ylab("Temperature")
```



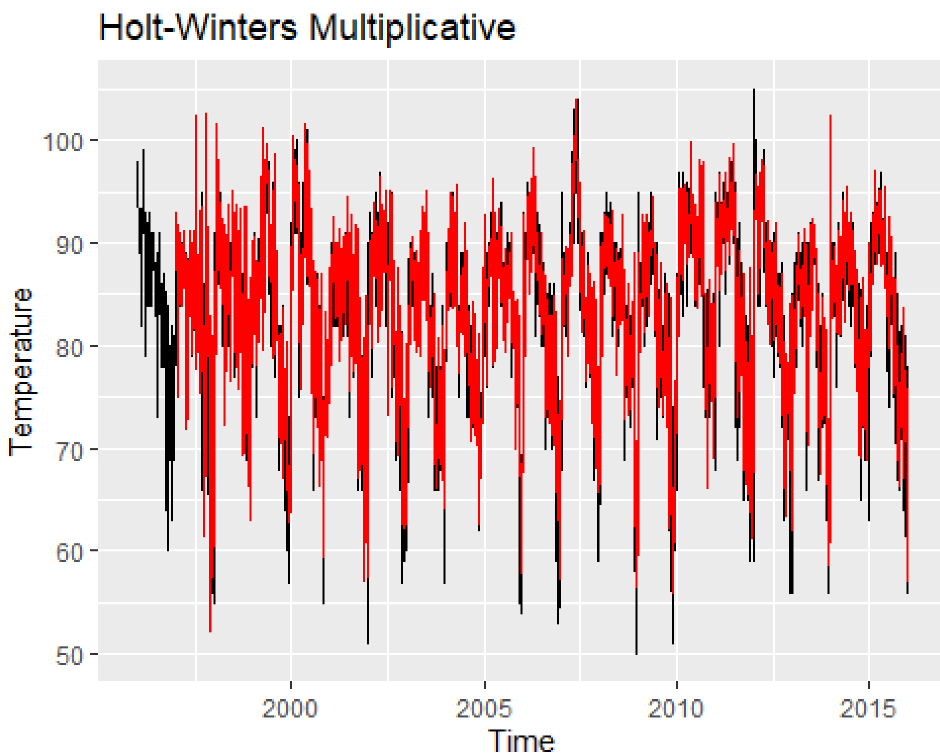
The pattern is relatively consistent and the next step would be to view the constants (level, trend, seasonality) in both an additive and multiplicative model. Below is the additive model. Additive seasonality is used when the seasonal fluctuations are roughly constant over time.

```
set.seed(123) #reproducibility

#holtwinter additive
hw_add<- HoltWinters(temp_ts,seasonal="additive")
autoplot(hw_add) +
  ggtitle("Holt-Winters Additive") +
  xlab("Time") +
  ylab("Temperature")

## Warning: Removed 123 rows containing missing values or values outside the
scale range
## (`geom_line()`).
```



Next, is the multiplicative. Multiplicative seasonality is used when the seasonal fluctuations change proportionally with the level of the time series. This is appropriate if the seasonal effect is more pronounced when the time series level is higher.

```
#holtwinter multiplicative
hw_mul<- HoltWinters(temp_ts, seasonal="multiplicative")
autoplot(hw_mul) +
  ggtitle("Holt-Winters Multiplicative") +
  xlab("Time") +
  ylab("Temperature")

## Warning: Removed 123 rows containing missing values or values outside the
scale range
## (`geom_line()`).
```



By applying both additive and multiplicative Holt-Winters models, I can compare which model better fits the data. This is crucial for understanding whether the seasonal effects are constant or proportional to the level of the time series. These models show a comprehensive view of the behavior, showing not only the fitted values but also how the model projects the future and how the actual data compares with the smoothed trends. It displays a richer set of information, including both the observed time series and how the Holt-Winters method decomposes it over time

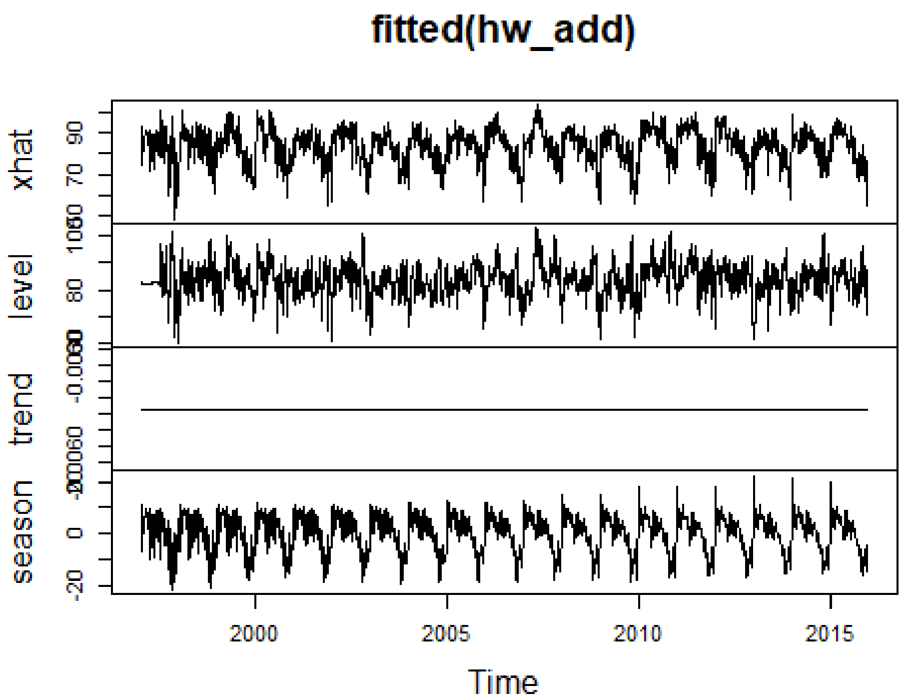
Now that have have 2 models to compare, we can look at the individual constants (the greeks: alpha, beta, gamma). These are the smoothing constants that determine the influence of past observations on the smoothed values for level, trend, and seasonality. By comparing these values for the additive and multiplicative methods, you can assess how each model is tuning these components. The Holt-Winter function also returns an SSE value (Sum of squared errors). This provides a measure of the goodness-of-fit for each model. Comparing SSE for the additive and multiplicative models helps determine which model fits the data better.

```
#hw add results (greeks)
print("Additive method:")

## [1] "Additive method:"

print(paste("\tBase factor (Alpha):", hw_add$alpha))

## [1] "\tBase factor (Alpha): 0.661061754684708"

print(paste("\tTrend factor (Beta):", hw_add$beta))
```

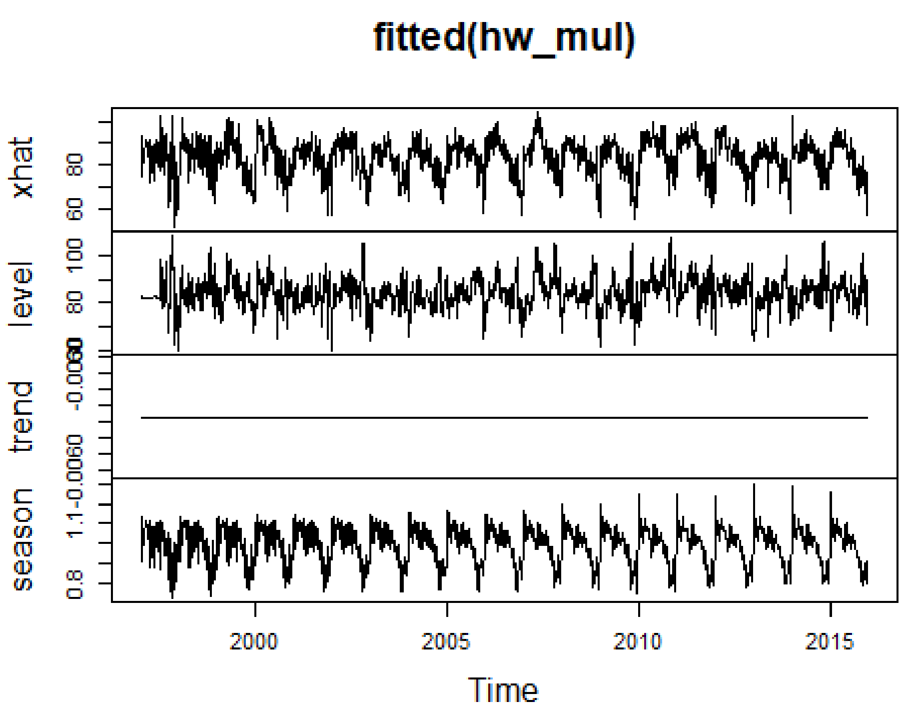
```
## [1] "\tTrend factor (Beta): 0"

print(paste("\tSeasonal factor (Gamma):", hw_add$gamma))

## [1] "\tSeasonal factor (Gamma): 0.624807621487671"

print(paste("\tSum of Squared Errors (SSE):", hw_add$SSE))

## [1] "\tSum of Squared Errors (SSE): 66244.2504058465"

#hw mult results (greeks)
print("For multiplicative method:")

## [1] "For multiplicative method:"

print(paste("\tBase factor (Alpha):", hw_mul$alpha))

## [1] "\tBase factor (Alpha): 0.615002994570597"

print(paste("\tTrend factor (Beta):", hw_mul$beta))

## [1] "\tTrend factor (Beta): 0"

print(paste("\tSeasonal factor (Gamma):", hw_mul$gamma))

## [1] "\tSeasonal factor (Gamma): 0.549525578453196"

print(paste("\tSum of Squared Errors (SSE):", hw_mul$SSE))

## [1] "\tSum of Squared Errors (SSE): 68904.5693317477"
```

I can also visualize the fitted values from the Holt-Winters models. These basically show how well the model's predictions align with the actual observed models. It also breaks down the time series into level, trend, and seasonality. This is useful because it tells us how well the model fits data over time so we can see how it captures the patterns in the historical data.

```
#visualize fitted values
par(mfrow=c(1,2)) #set up plotting area
plot(fitted(hw_add))
```




```r
plot(fitted(hw_mul))
```



Since the additive model had a smaller SSE, I will use that to draw my conclusion.

```
hw_add$fitted[1:5,] #first 5 rows of fitted values
```

```
##          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
```

The code displayed the first five rows of fitted values from the additive model. I did this just so I can see the initial part of fitted data and see how the model performed at the beginning of the time series.

x-hat is the fitted values or predicted values of the time series. These are the values that the Holt-Winters model estimates for each time point based on the historical data.

level is the level component of the model at each time point. The level is the smoothed estimate of the base value of the time series.

trend is the trend component of the model at each time point. The trend indicates the direction and rate of change in the data over time.

season is the seasonal component of the model at each time point. The seasonal component reflects the periodic fluctuations in the data.

By inspecting the visualized models as well as the data, I can see almost no trend. I can conclude that there is no evidence that summer is lasting longer. The temperatures and duration of the warm season are somewhat consistent throughout the years.