HW4 - Time Series

Kristiyan Dimitrov

03/14/2020 - Happy Pi Day!

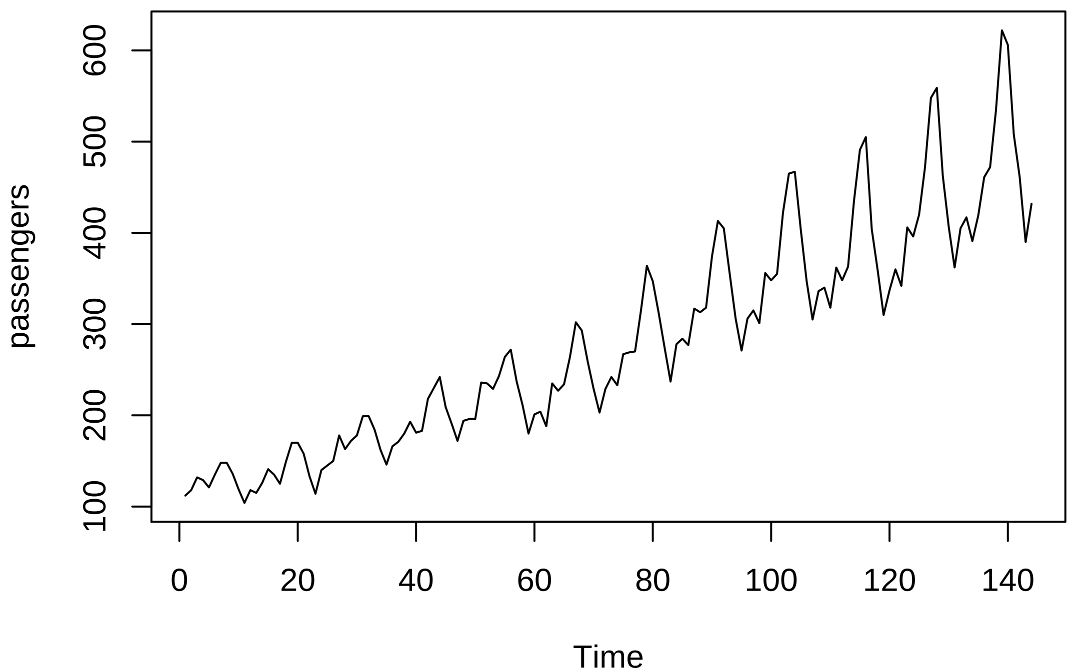
Question 1

Loading the data

```
New names:
* `` -> ...1
```

```
'data.frame':    144 obs. of  1 variable:
 $ passengers: num  112 118 132 129 121 135 148 148 136 119 ...
```

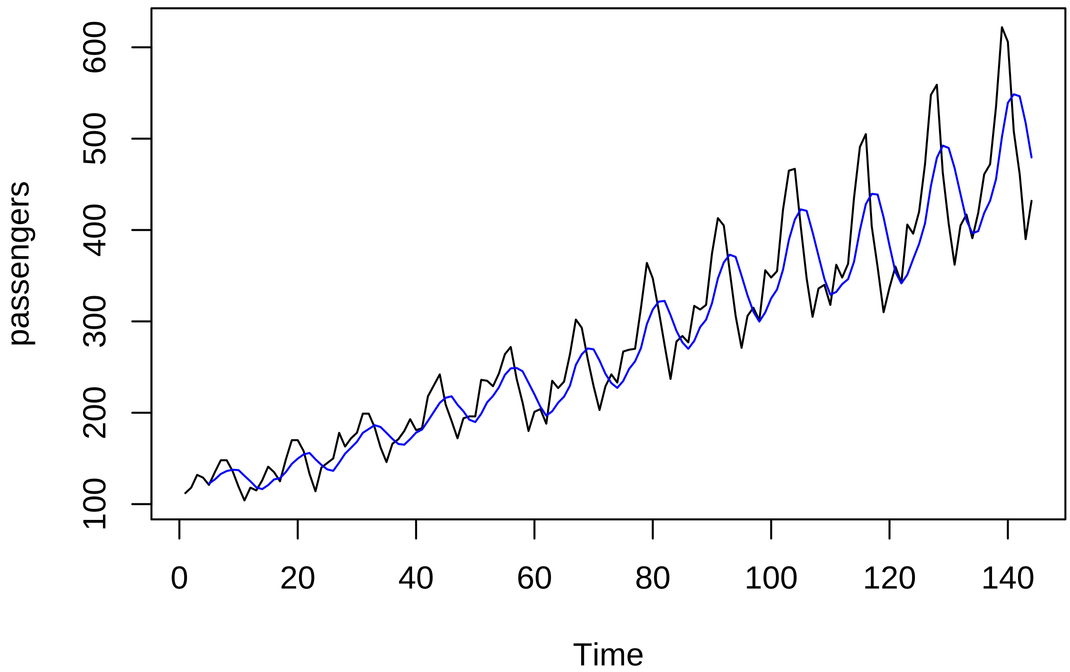
Let's convert to timeseries object and plot the data



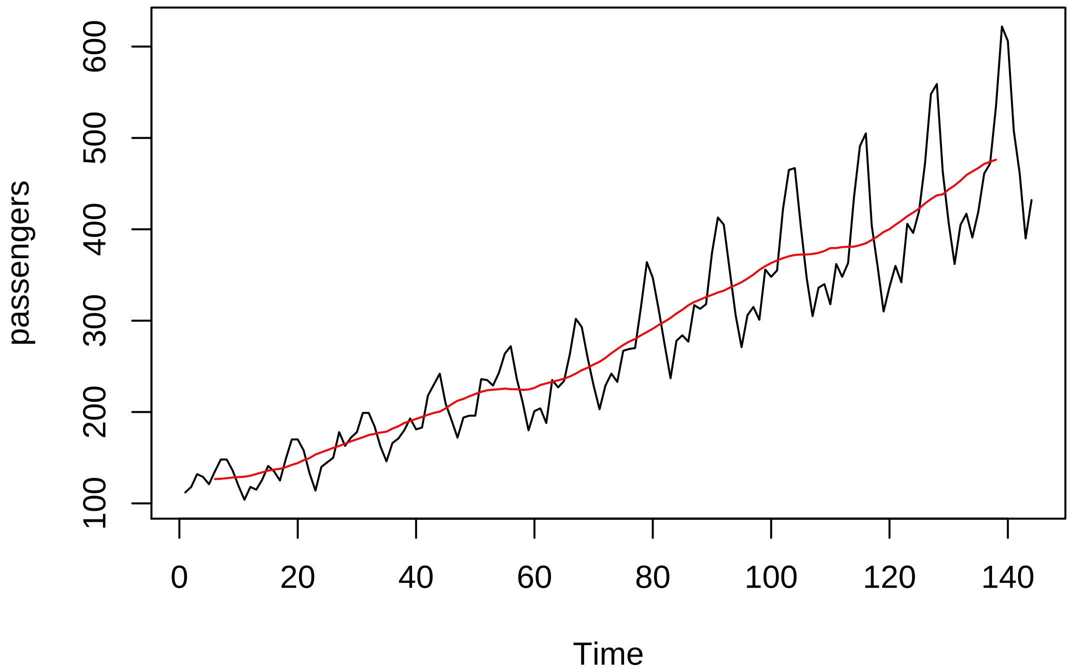
I observe that there is definitely an (upward) Trend. Furthermore, there is definite seasonality, which apparently is related to the trend. Therefore, a multiplicative model would probably be better: passengers = T x S x R (R is always there, because we assume there's always noise)

(a) - A moving average filter to smooth out the seasonality

I'll try a few different values of m.

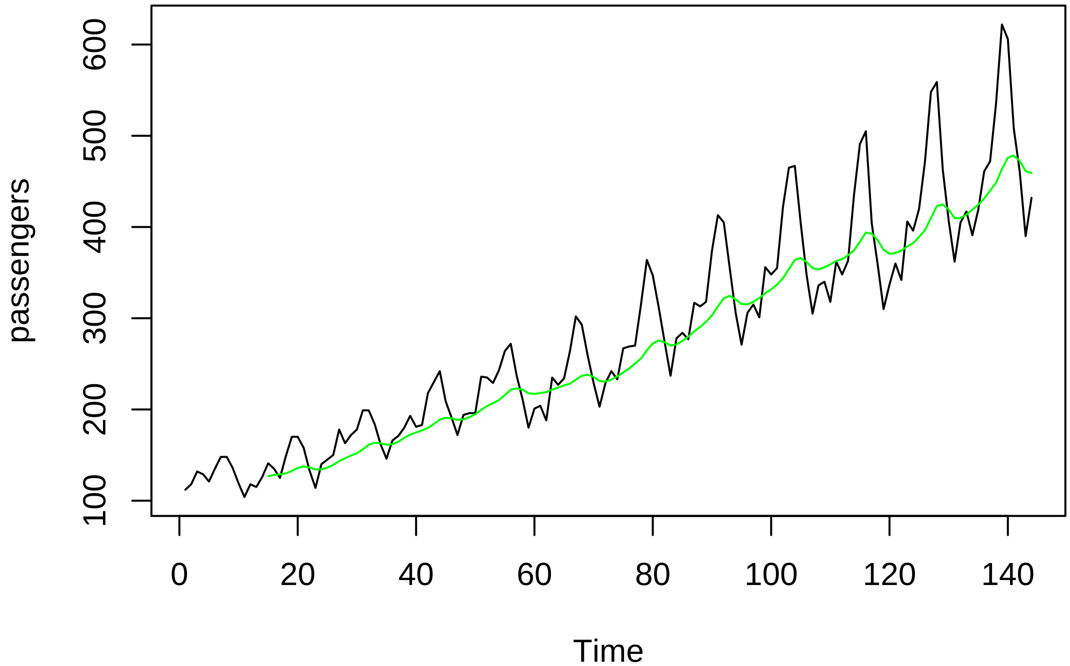


I don't think this m is high enough, because it's still reading in the seasonality. Let's repeat with m=12 (yearly)



This looks much more reasonable for the Trend (T) part of the graph. Note that I've also specified sides = 2, to do a centered moving average. Prof. Apley suggested that we use this option only for smoothing and never for predictions. In this case we are doing smoothing and I think it's a more appropriate option. It shifts the red line a bit to the left and so it fits the trend a bit better. Without this adjustment, the line is a bit to the right and looks like it's underestimating the trend. In fact, the professor says that 'you should always center the MA" for retrospective smoothing. Therefore, this is indeed the best option. He also says, on the same slide, "If smoothing out strong seasonality, use an MA with m exactly equal the period of seasonality". The seasonality is definitely annual, so m=12 is a good choice.

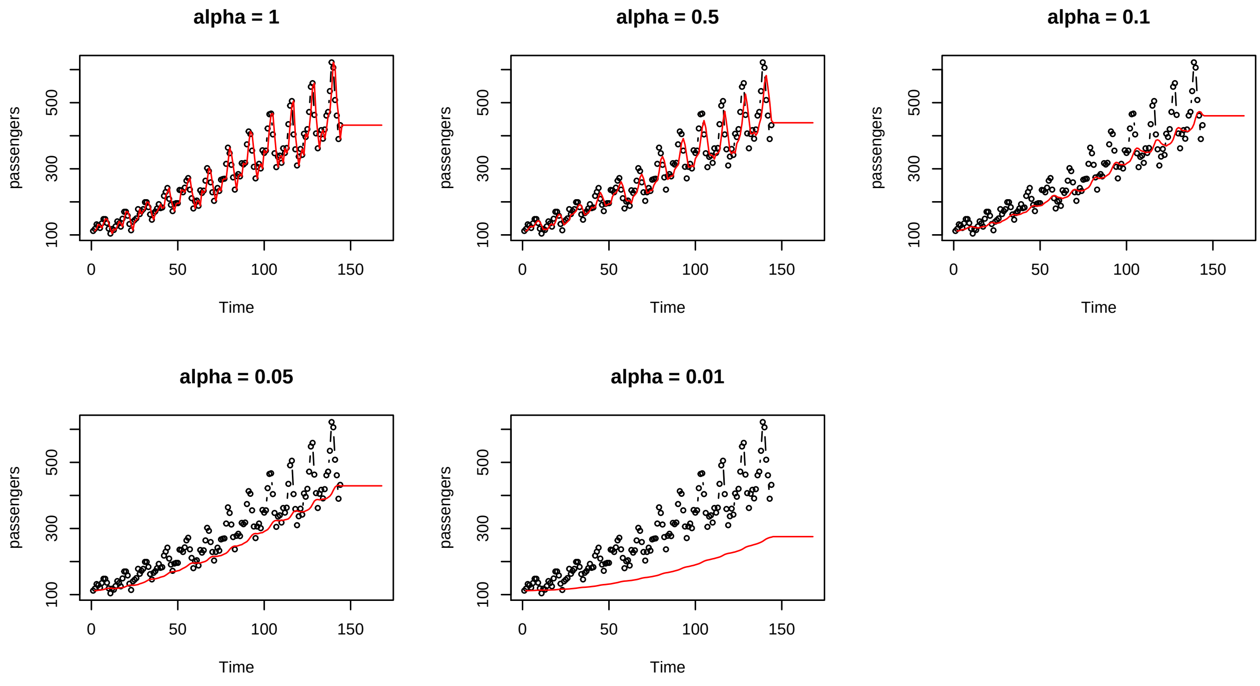
What happens if I use m=15 for example?



Not good; Certain estimates are based on two seasonalities and so are once again creating new seasonalities (represented by the wavy line).

(b) - Calculating an Exponentially Weighted Moving Average (EWMA) for the next 2 yrs

I specify k=24, because we want to make predictions for the next 2 yrs (24 months)



I think an alpha somewhere between 0.1 and 0.05 fits the data most reasonably. Specifically looking at the 0.1 graph, I think that the 24-step-ahead prediction will dramatically understimate the peaks in the data during high seasonality. More importantly, it completely ignores any sort of trend or seasonality when making a prediction; that's why the professor says in his notes that MA & EWMA are not used when there is seasonality or a trend; Better to use them just for smoothing out R & S (leaving T & C) or when making predictions, which require only R (noise) and C (cyclical)

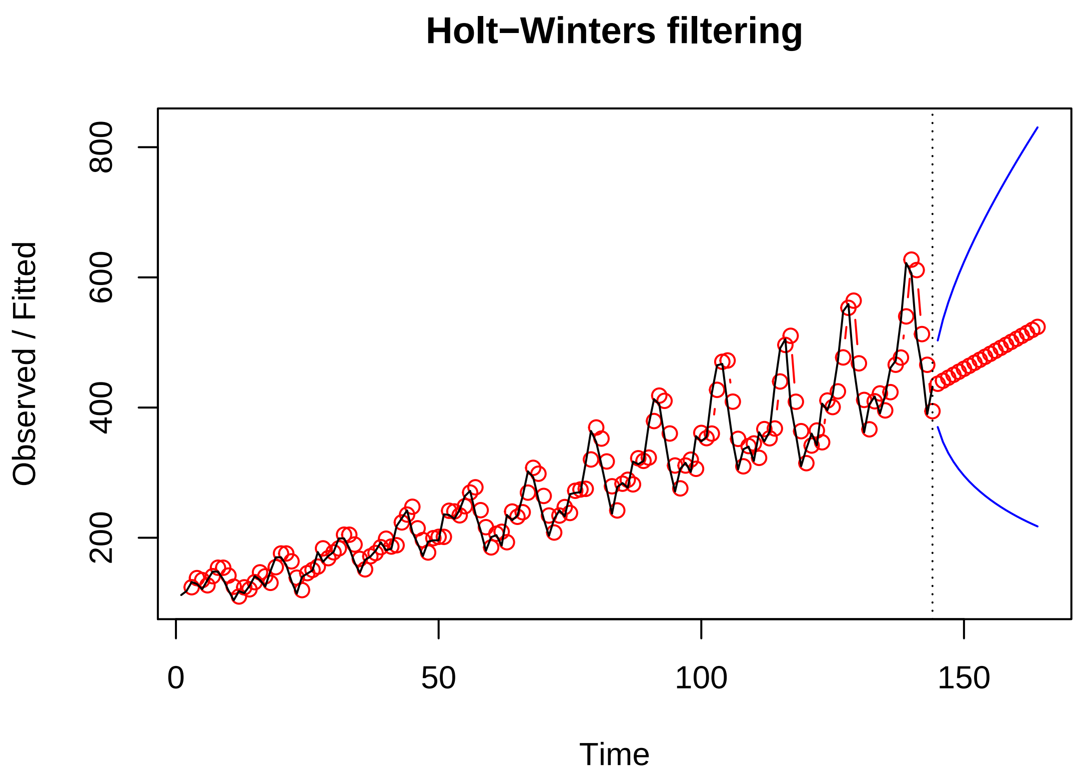
Below I've calculated the weight that EWMA will give to each of the last 24 observations when calculating the level (with alpha = 0.1)

```
 [1] 0.100000000 0.090000000 0.081000000 0.072900000 0.065610000
 [6] 0.059049000 0.053144100 0.047829690 0.043046721 0.038742049
[11] 0.034867844 0.031381060 0.028242954 0.025418658 0.022876792
[16] 0.020589113 0.018530202 0.016677182 0.015009464 0.013508517
[21] 0.012157665 0.010941899 0.009847709 0.008862938
```

c) Calculate & Plot the Holt method forecasts for the next 2 years; what are the optimal alpha & beta?

```
Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = flights, alpha = NULL, beta = NULL, gamma = FALSE,     seasonal = "additive")

Smoothing parameters:
 alpha: 1
 beta : 0.003218516
 gamma: FALSE

Coefficients:
        [,1]
a 432.000000
b   4.597605
```

The estimated best alpha is 1. I think this means that we wouldn't be doing EWMA and just MA, because all weights for observations other than the 'current' one will be 0. The optimal beta is = .0032. I shouldn't confuse it with the b coefficient (=4.597), which is the slope of the prediction based on the trend. In short, the smoothing parameters are used to calculate the coefficients. Now I will make predictions for the next 24 months and plot the estimates



The upside of this new prediction is that it takes into consideration the trend (that's why we included the beta parameter to be estimated)

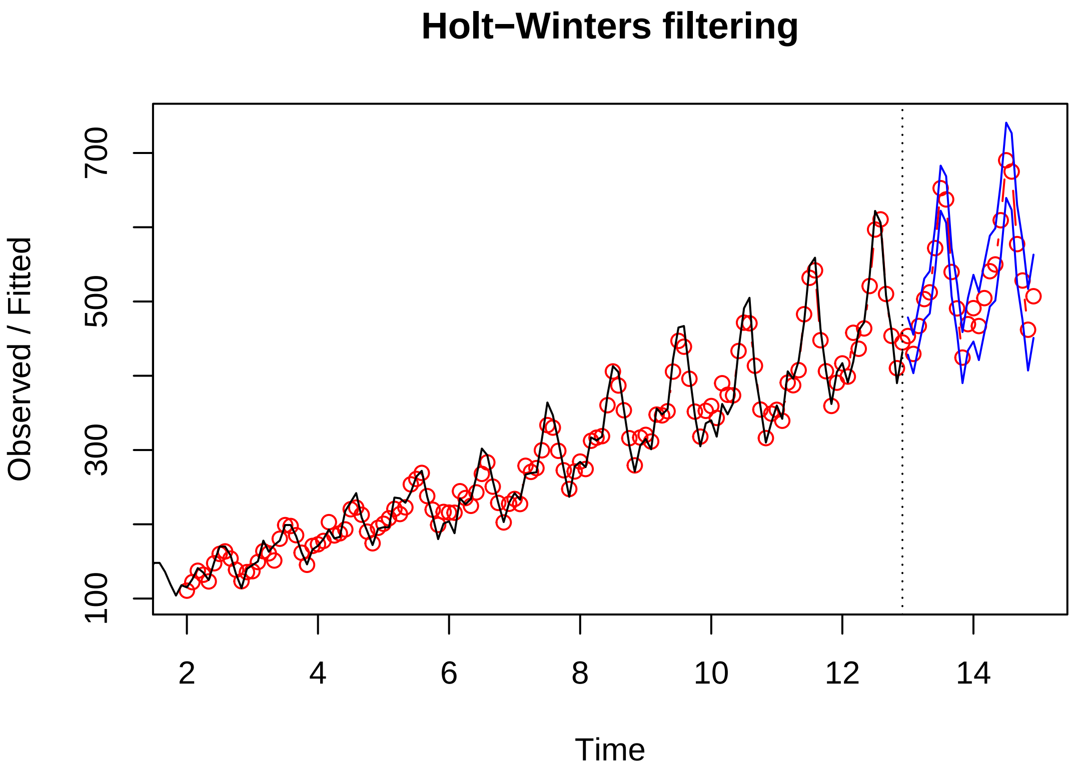
The downside is that by specifying gamma = FALSE, we are still ignoring the seasonality; Therefore, just like our EWMA estimate we will be doing a poor job of predicting those seasonalities.

d) = Using HoltWinters for an additive model with seasonality & trend included

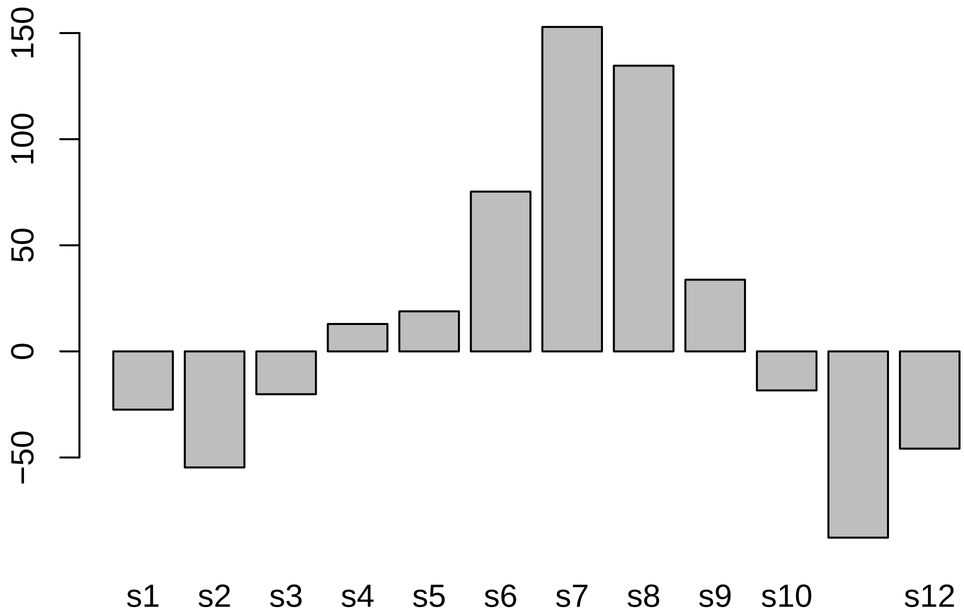
```
   Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1  112 118 132 129 121 135 148 148 136 119 104 118
2  115 126 141 135 125 149 170 170 158 133 114 140
3  145 150 178 163 172 178 199 199 184 162 146 166
4  171 180 193 181 183 218 230 242 209 191 172 194
5  196 196 236 235 229 243 264 272 237 211 180 201
6  204 188 235 227 234 264 302 293 259 229 203 229
7  242 233 267 269 270 315 364 347 312 274 237 278
8  284 277 317 313 318 374 413 405 355 306 271 306
9  315 301 356 348 355 422 465 467 404 347 305 336
10 340 318 362 348 363 435 491 505 404 359 310 337
11 360 342 406 396 420 472 548 559 463 407 362 405
12 417 391 419 461 472 535 622 606 508 461 390 432

Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = flights, alpha = NULL, beta = NULL, gamma = NULL,     seasonal = "additive")

Smoothing parameters:
 alpha: 0.2479595
```

```
 beta : 0.03453373
 gamma: 1

Coefficients:
            [,1]
a    477.827781
b      3.127627
s1   -27.457685
s2   -54.692464
s3   -20.174608
s4    12.919120
s5    18.873607
s6    75.294426
s7   152.888368
s8   134.613464
s9    33.778349
s10  -18.379060
s11  -87.772408
s12  -45.827781
```

The optimal smoothing parameters are alpha = .248, beta = .0345, gamma = 1. The fact that gamma = 1 means we consider only the latest seasonality value is taken into consideration when calculating a prediction. Now I make predictions and plot them.



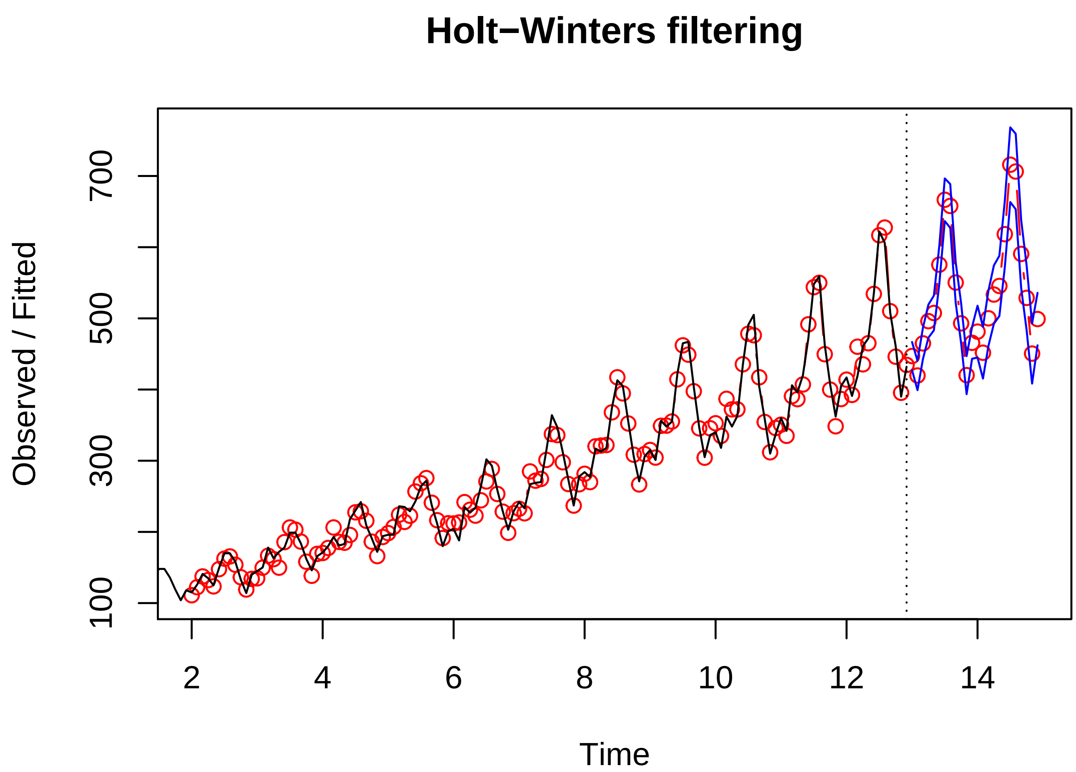
Let's look at the seasonality effects month by month. As expected, there are many more passengers during the summer (June-August, s6-s8)



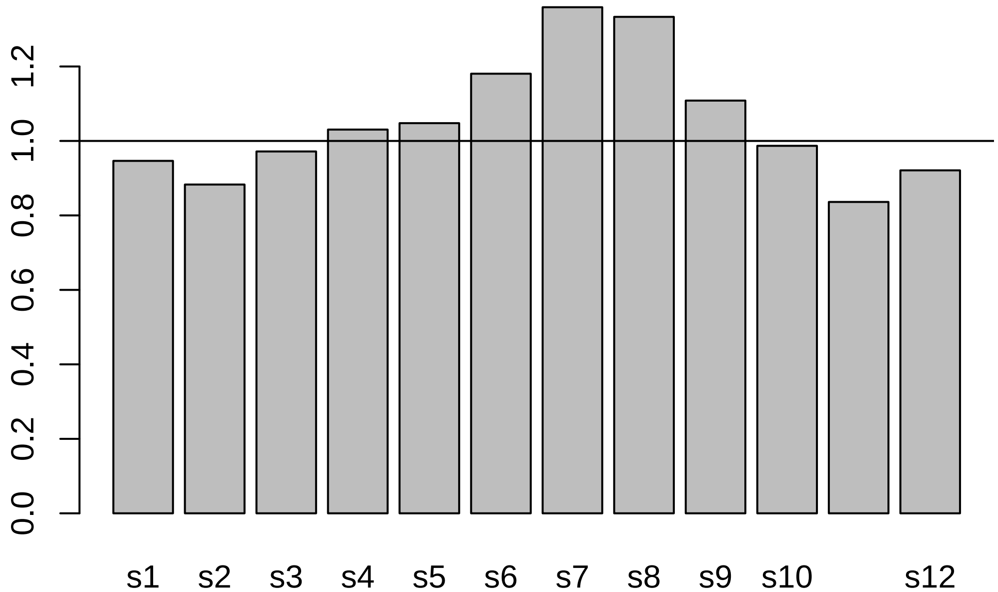
e) Multiplicative Model

```
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.

Call:
HoltWinters(x = flights, alpha = NULL, beta = NULL, gamma = NULL,     seasonal = "multiplicative")

Smoothing parameters:
 alpha: 0.2755925
 beta : 0.03269295
 gamma: 0.8707292

Coefficients:
           [,1]
a   469.3232206
b     3.0215391
s1    0.9464611
s2    0.8829239
s3    0.9717369
s4    1.0304825
s5    1.0476884
s6    1.1805272
s7    1.3590778
s8    1.3331706
s9    1.1083381
s10   0.9868813
s11   0.8361333
s12   0.9209877
```

The best parameters are alpha = .0275; beta = .0327; gamma = .8707



The seasonalities can be interpreted in the same way (s_i < 1, means lower than average seasonality i.e. 'negative' seasonality) i.e. months 6-8 are high seasonality, months 2 & 11 (feb & nov) are lowest.



f) Which method produces better forecasts?

As I mentioned at the very beginning, a multiplicative model is more appropriate for predictions, because it appears that the trend influences the amplitude of the seasonality i.e. higher trend value -> larger seasonality spike.

We can confirm this numerically by looking at the SSE of the multiplicative & additive models:

```
[1] 16570.78
```

```
[1] 21860.18
```

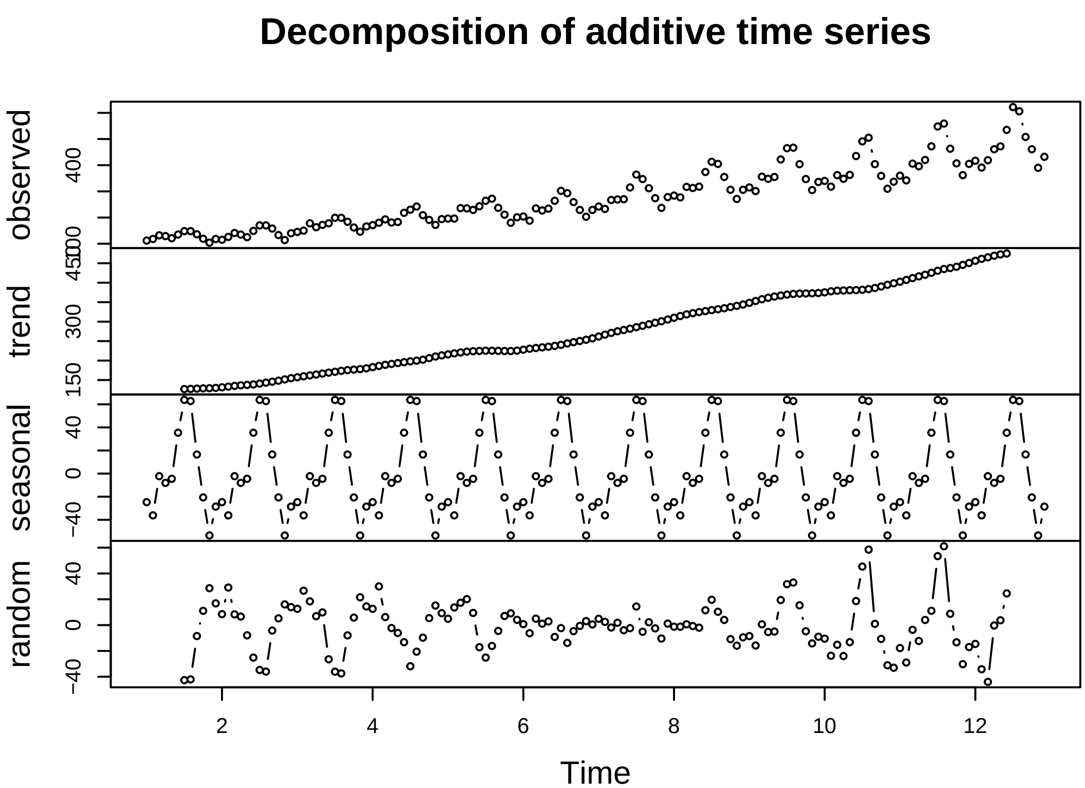
We see that the Multiplicative one has a much lower SSE, which means it fits the data better. The documentation describes this SSE as: "The final sum of squared errors achieved in optimizing"

Question 2

```
New names:
* `` -> ...1
```

```
'data.frame':	144 obs. of  1 variable:
 $ passengers: num  112 118 132 129 121 135 148 148 136 119 ...
```

a) Additive Decomposition model

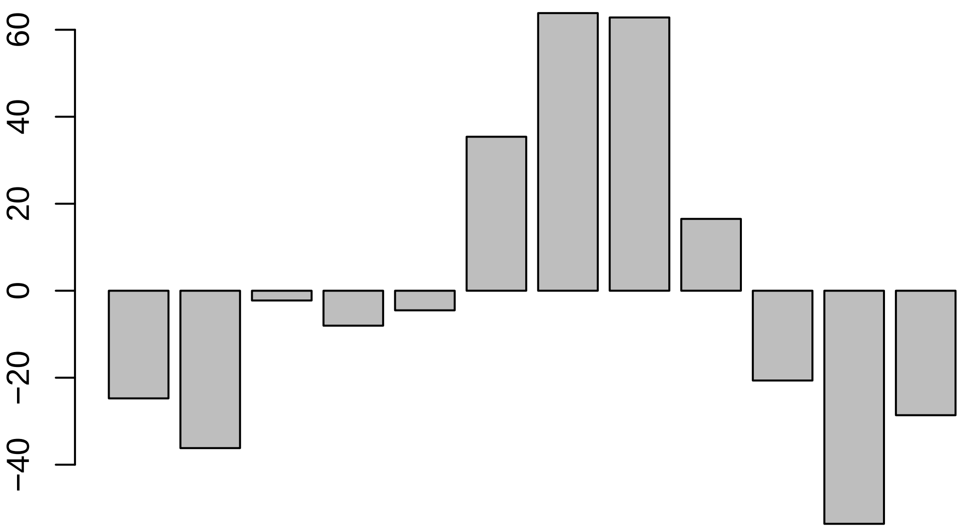


Below we see the trend indices. It is expected behavior that the first 6 and last 6 are NAs, because by default the decompose function uses a centered MA approach to extract the trend. The number is 6, because that is half the frequency of the data (which is 12 i.e. monthly data with annual seasonality)

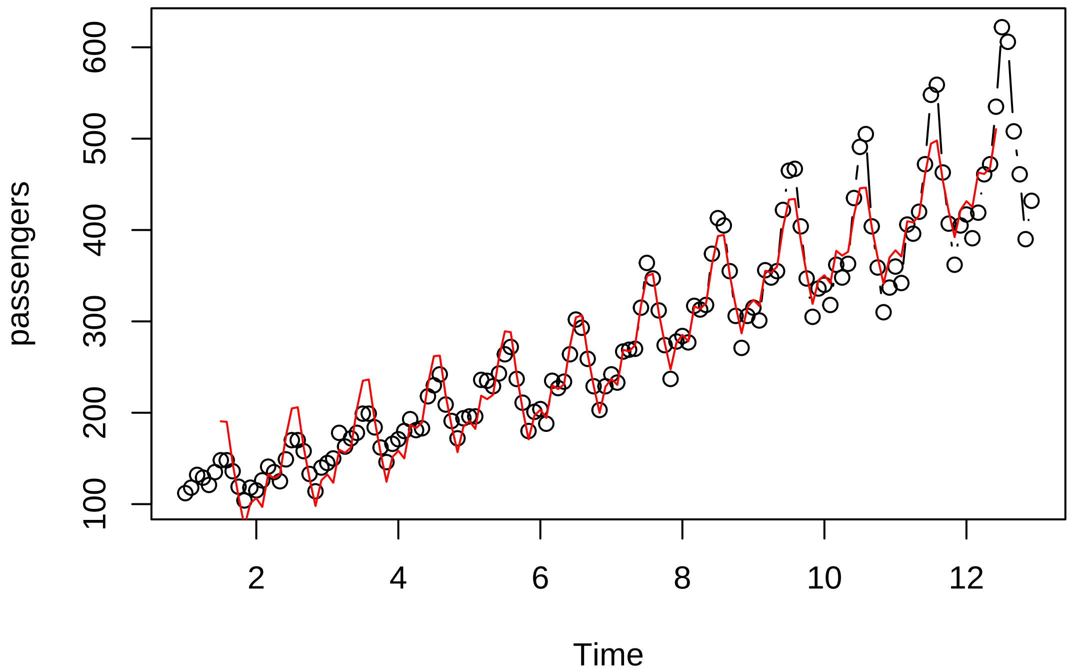
```
       Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
1       NA       NA       NA       NA       NA       NA 126.7917 127.2500
2 131.2500 133.0833 134.9167 136.4167 137.4167 138.7500 140.9167 143.1667
3 157.1250 159.5417 161.8333 164.1250 166.6667 169.0833 171.2500 173.5833
4 183.1250 186.2083 189.0417 191.2917 193.5833 195.8333 198.0417 199.7500
```

```
5  215.8333 218.5000 220.9167 222.9167 224.0833 224.7083 225.3333 225.3333
6  228.0000 230.4583 232.2500 233.9167 235.6250 237.7500 240.5000 243.9583
7  261.8333 266.6667 271.1250 275.2083 278.5000 281.9583 285.7500 289.3333
8  309.9583 314.4167 318.6250 321.7500 324.5000 327.0833 329.5417 331.8333
9  348.2500 353.0000 357.6250 361.3750 364.5000 367.1667 369.4583 371.2083
10 375.2500 377.9167 379.5000 380.0000 380.7083 380.9583 381.8333 383.6667
11 402.5417 407.1667 411.8750 416.3333 420.5000 425.5000 430.7083 435.1250
12 456.3333 461.3750 465.2083 469.3333 472.7500 475.0417       NA       NA
        Sep      Oct      Nov      Dec
1  127.9583 128.5833 129.0000 129.7500
2  145.7083 148.4167 151.5417 154.7083
3  175.4583 176.8333 178.0417 180.1667
4  202.2083 206.2500 210.4167 213.3750
5  224.9583 224.5833 224.4583 225.5417
6  247.1667 250.2500 253.5000 257.1250
7  293.2500 297.1667 301.0000 305.4583
8  334.4583 337.5417 340.5417 344.0833
9  372.1667 372.4167 372.7500 373.6250
10 386.5000 390.3333 394.7083 398.6250
11 437.7083 440.9583 445.8333 450.6250
12       NA       NA       NA       NA
```

Here are the seasonality indices; they look pretty similar to the ones from the HoltWinters model.

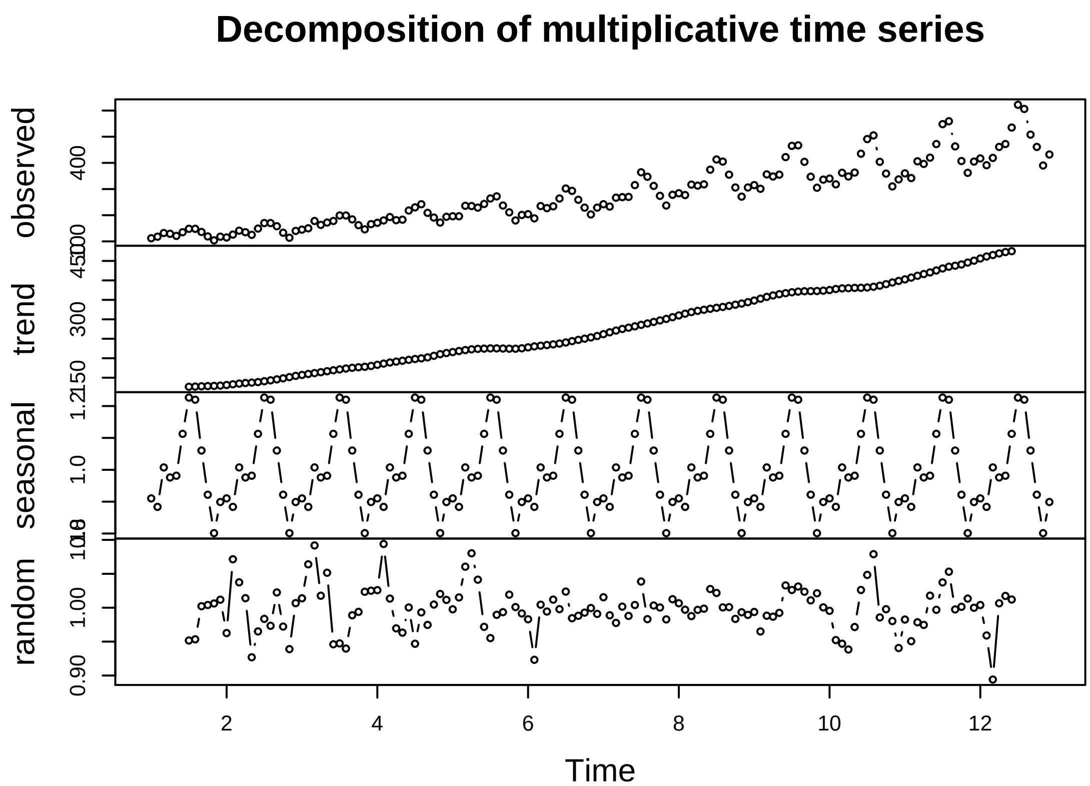


Now I will construct a plot of the original time series and the fitted values.

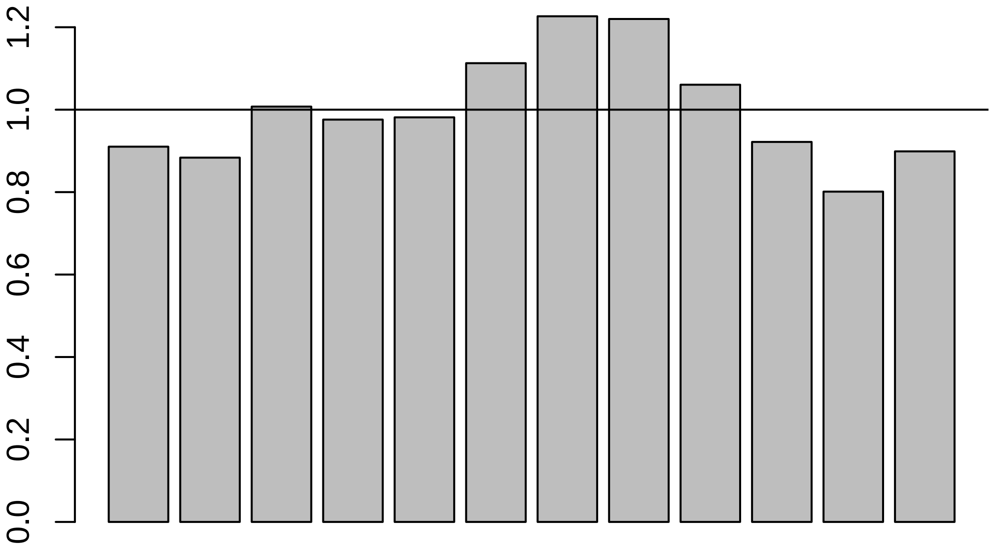


As in question 1, it is apparent that the predicted values overestimate the spikes in the beginning of the data and underestimate them in the end. Again, I suspect a multiplicative model will address this issue well.

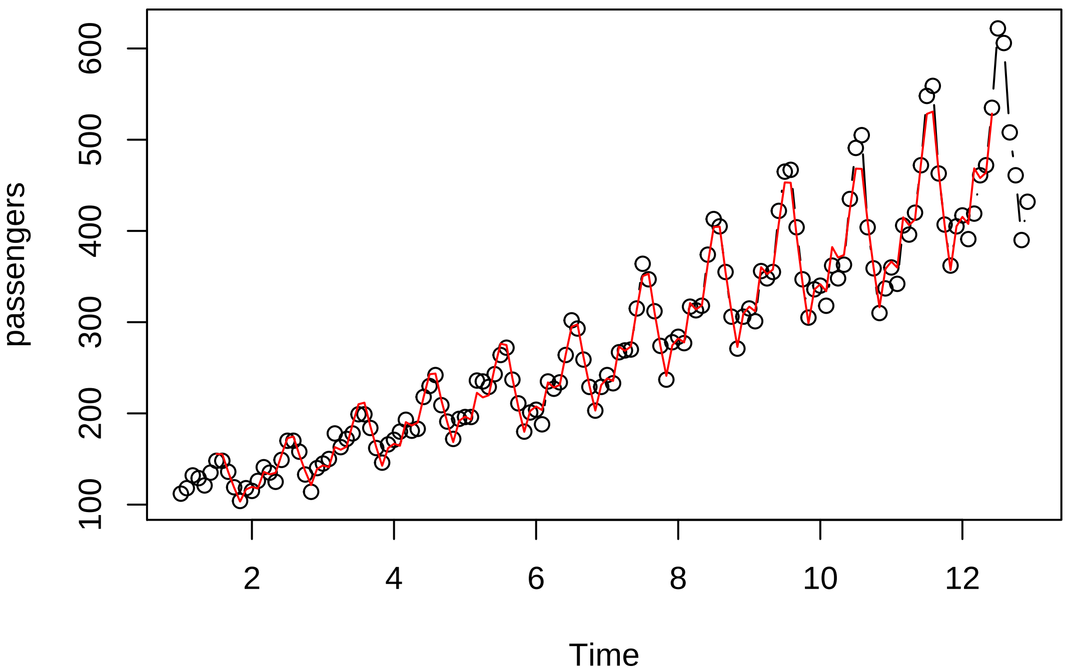
b) Multiplicative Decomposition model



Here are the seasonality indices



Here is the fitted and observed data on the same plot



I think the multiplicative model, once again represents the data better.

If we look at the random part of the additive decomposition, we see there are negative spikes in the beginning and positive spikes towards the end of the data. In other words, we are interpreting the lower than average seasonal spikes in the beginning of the data AS WELL AS the higher than average spikes towards the end of the data as noise.

In fact, they are not noise, but a sign that the amplitude of the seasonality is dependent on the trend and therefore a multiplicative model is more suitable to decompose this time series.

This is even more apparent when we look at the fitted & observed values on the same plot. The seasonality is definitely better captured by the multiplicative model.

Appendix

This section is to be used for including your R code. The following lines of code will take care of it. Please make sure to comment your code appropriately - in particular, demarcating codes belonging to different questions. Among other things, it will be easier for you to debug your own code.

```
# define functions used globally which are not dependent on any question specifics
# instead of reloading them each time
# you can also use this space to load R packages
# or custom scripts sourced from a R file
# for example, I have included the CV_ind function

library(readxl)

CVInd <- function(n,K) {
  # n is sample size; K is number of parts;
  # returns K-length list of indices for each part
  m<-floor(n/K) #approximate size of each part
  r<-n-m*K
  I<-sample(n,n) #random reordering of the indices
  Ind<-list() #will be list of indices for all K parts
  length(Ind)<-K
  for (k in 1:K) {
    if (k <= r) kpart <- ((m+1)*(k-1)+1):((m+1)*k)
    else kpart<-((m+1)*r+m*(k-r-1)+1):((m+1)*r+m*(k-r))
    Ind[[k]] <- I[kpart] #indices for kth part of data
  }
  Ind
}
########### QUESTION 1 begins here ##################
flights = read_excel('HW4_data.xls',col_names = F)
flights = as.data.frame(flights)
colnames(flights) = 'passengers'
str(flights)
# flights = ts(flights, frequency = 12)
flights = ts(flights) # deltat = 1/12
plot(flights)
m = 5
n = length(flights)
weights = rep(1/m, m)
flightsMovingAverage = stats::filter(flights, filter = weights, method = 'convolution', sides = 1) # Sp
plot(flights)
smooth=c(flightsMovingAverage) # I did not include the NA in the vector (like it's done in the Notes co
lines(smooth, col='blue')
m = 12
weights = rep(1/m, m)
flightsMovingAverage = stats::filter(flights, filter = weights, method = 'convolution', sides = 2)
plot(flights)
smooth=c(flightsMovingAverage) # I did not include the NA in teh vector (like it's done in the Notes co
lines(smooth, col='red')
m = 15
weights = rep(1/m, m)
flightsMovingAverage = stats::filter(flights, filter = weights, method = 'convolution', sides = 1) # Sp
plot(flights)
```

```
smooth=c(flightsMovingAverage) # I did not include the NA in teh vector (like it's done in the Notes co
lines(smooth, col='green')
### SURAJ'S EWMA FUNCTION WITH MULTIPLE ALPHA VALUES
plot_ewma <- function(alpha,k=24){ # Specify k=24, because we want to make predictions for the next 2 y
  n <- length(flights)
  EWMA <- filter(alpha*flights,filter=1-alpha,method='recursive',sides=1,init=flights[1]) # Since this
  yhat <- c(NA,EWMA,rep(EWMA[n],k-1)) # one-step ahead predictions and mean
  plot(flights,type='b',xlim=c(0,n+k),main=paste('alpha =',alpha))
  lines(yhat,col='red')
}

par(mfrow=c(2,3))
for(alpha in c(1,0.5,0.1,0.05, 0.01)){
  plot_ewma(alpha)
}
weights = c()
for (j in seq(0,23,1)) {
  weights = c(weights, .1*((.9)^j))
}; weights
# estimating optimal alpha & beta hrough Holt method, which does CV and compares based on MSD (MSE)
Holtflights <- HoltWinters(
  flights,
  alpha = NULL, # specify NULL (default) for estimation or specify a fixed value
  beta = NULL, # trend parameter; set FALSE to ignore trends; I will set it to NULL, because I want it
  gamma = FALSE, # seasonality parameter; set FALSE to ingore seasonalities
  seasonal = 'additive' # additive or multiplicative model (Only takes effect if gamma is non-zero).
)
Holtflights
Holtpred <- predict(Holtflights,n.ahead=20,prediction.interval = TRUE,level=0.95)
plot(Holtflights,Holtpred,type='b')
flights = ts(flights, frequency = 12); flights # I had to specify freq = 12, because otherwise I was ge
# estimating optimal alpha & beta hrough Holt method, which does CV and compares based on MSD (MSE)
HoltAdd <- HoltWinters(
  flights,
  alpha = NULL, # specify NULL (default) for estimation or specify a fixed value
  beta = NULL, # trend parameter; set FALSE to ignore trends; I will set it to NULL, because I want it
  gamma = NULL, # seasonality parameter; set FALSE to ingore seasonalities
  seasonal = 'additive' # additive or multiplicative model (Only takes effect if gamma is non-zero).
)
HoltAdd
k = 24
Holtpred<-predict(HoltAdd, n.ahead=k, prediction.interval = T, level = 0.95)
plot(HoltAdd,Holtpred,type="b")
barplot(HoltAdd$coefficients[-c(1,2)])
HoltMult <- HoltWinters(
  flights,
  alpha = NULL, # specify NULL (default) for estimation or specify a fixed value
  beta = NULL, # trend parameter; set FALSE to ignore trends; I will set it to NULL, because I want it
  gamma = NULL, # seasonality parameter; set FALSE to ingore seasonalities
  seasonal = 'multiplicative' # additive or multiplicative model (Only takes effect if gamma is non-zer
)
HoltMult
Holtpred<-predict(HoltMult, n.ahead=k, prediction.interval = T, level = 0.95)
```

```r
plot(HoltMult,Holtpred,type="b")
barplot(HoltMult$coefficients[-c(1,2)])
abline(a=1, b=0)
HoltMult$SSE
HoltAdd$SSE
########### QUESTION 2 begins here ###################
# Improt data
flights = read_excel('HW4_data.xls',col_names = F)
flights = as.data.frame(flights)
colnames(flights) = 'passengers'
str(flights)
flights = ts(flights, deltat = 1/12)
decomposeFlightsAdd = decompose(flights, type = 'additive')
plot(decomposeFlightsAdd,type='b')
decomposeFlightsAdd$trend
barplot(decomposeFlightsAdd$figure)
predictionsAdd = decomposeFlightsAdd$trend + decomposeFlightsAdd$seasonal
plot(flights,type="b")
lines(predictionsAdd, col='red')
decomposeFlightsMult = decompose(flights, type = 'multiplicative')
plot(decomposeFlightsMult,type='b')
barplot(decomposeFlightsMult$figure)
abline(a=1,b=0)
predictionsMult = decomposeFlightsMult$trend * decomposeFlightsMult$seasonal
plot(flights,type="b")
lines(predictionsMult, col='red')
```