624-Hw4

Joseph Elikishvili

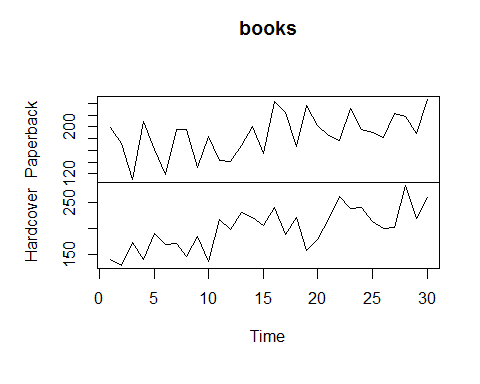
September 17, 2017

## Ex. HA7.1

### Data set books contains the daily sales of paperback and hardcover books at the same store. The task is to forecast the next four days' sales for paperback and hardcover books (data set books).

#### A. Plot the series and discuss the main features of the data.

plot(books)



We can see that there is an uptrend present in both paperback and hardcover versions. Also there is a fluctuation in salse that resembles seasonality present more in paperback but is also present in hardcover, it is likely that that are days of week like wednesday or monday where there is less traffic and potentially weekend days that attract more customers.

#### B. Use simple exponential smoothing with the ses function (setting initial="simple") and explore different values of ???? for the paperback series. Record the within-sample SSE for the one-step forecasts. Plot SSE against ???? and find which value of ???? works best. What is the effect of ???? on the forecasts?

paperback = books[,1]  
  
fit1 = ses(paperback, alpha=0.2, initial="simple", h=3)  
fit2 = ses(paperback, alpha=0.5, initial="simple", h=3)  
fit3 = ses(paperback, alpha=0.7, initial="simple", h=3)  
#fit4 = ses(paperback, initial="simple", h=3)  
fit5 = ses(paperback, alpha=0.1, initial="simple", h=3)  
fit6 = ses(paperback, alpha=0.9, initial="simple", h=3)  
sum((paperback-fitted(fit1) )^2)

## [1] 36329.34

sum((paperback-fitted(fit2) )^2)

## [1] 41383.7

sum((paperback-fitted(fit3) )^2)

## [1] 48773.56

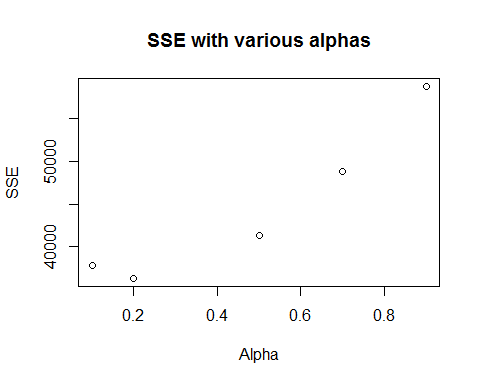
#sum((paperback-fitted(fit4) )^2)  
sum((paperback-fitted(fit5) )^2)

## [1] 37785.2

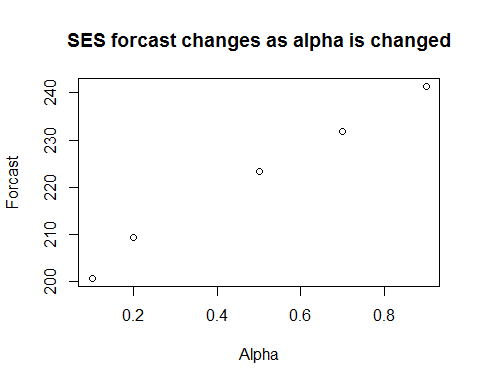
sum((paperback-fitted(fit6) )^2)

## [1] 58769.45

a = c(0.2, 0.5, 0.7, 0.1, 0.9)  
s = c(36329.34, 41383.7, 48773.56, 37785.2, 58769.45)  
f = c(209.38, 223.31, 231.74, 200.70, 241.39)  
  
plot(a,s, ylab="SSE", xlab="Alpha", main = "SSE with various alphas")



plot(a,f, ylab="Forcast", xlab="Alpha", main = "SES forcast changes as alpha is changed")



From the values I have picked alpha = 0.2 seems to have the lease SSE. SSE is dropping as it approaches 0.2 and then moves higher.

From second plot we can see that as we increase alpha, forcast also increases.

#### C. Now let ses select the optimal value of ????. Use this value to generate forecasts for the next four days. Compare your results with 2.

fit4 = ses(paperback, initial="simple", h=3)  
  
sum((paperback-fitted(fit4) )^2)

## [1] 36313.98

alpha = 0.2125, which is pretty close to 0.2 but would have been pretty tough to find without optimization. Lower SSE also confirms that we have the best value.

#### D. Repeat but with initial="optimal". How much difference does an optimal initial level make?

fit\_o = ses(paperback, initial="optimal", h=3)  
  
# OPtimized  
summary(fit\_o)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = paperback, h = 3, initial = "optimal")   
##   
## Smoothing parameters:  
## alpha = 0.1685   
##   
## Initial states:  
## l = 170.8257   
##   
## sigma: 33.6377  
##   
## AIC AICc BIC   
## 318.9747 319.8978 323.1783   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 7.176212 33.63769 27.8431 0.4737524 15.57782 0.7021303  
## ACF1  
## Training set -0.2117579  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 31 207.1098 164.0013 250.2182 141.1811 273.0384  
## 32 207.1098 163.3934 250.8261 140.2513 273.9682  
## 33 207.1098 162.7937 251.4258 139.3342 274.8853

#Simple  
summary(fit4)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = paperback, h = 3, initial = "simple")   
##   
## Smoothing parameters:  
## alpha = 0.2125   
##   
## Initial states:  
## l = 199   
##   
## sigma: 34.7918  
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 1.749509 34.79175 28.64424 -2.770157 16.56938 0.7223331  
## ACF1  
## Training set -0.1268119  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 31 210.1537 165.5663 254.7411 141.9631 278.3443  
## 32 210.1537 164.5706 255.7368 140.4404 279.8671  
## 33 210.1537 163.5962 256.7112 138.9501 281.3573

Looks like we have changed the forcst from 210.1537 to 207.1098

#### 

hardcover = books[,2]  
  
fith1 = ses(hardcover, alpha=0.2, initial="simple", h=3)  
fith2 = ses(hardcover, alpha=0.5, initial="simple", h=3)  
fith3 = ses(hardcover, alpha=0.7, initial="simple", h=3)  
fith5 = ses(hardcover, alpha=0.1, initial="simple", h=3)  
fith6 = ses(hardcover, alpha=0.9, initial="simple", h=3)  
sum((hardcover-fitted(fith1) )^2)

## [1] 33148.16

sum((hardcover-fitted(fith2) )^2)

## [1] 31702.6

sum((hardcover-fitted(fith3) )^2)

## [1] 34993.95

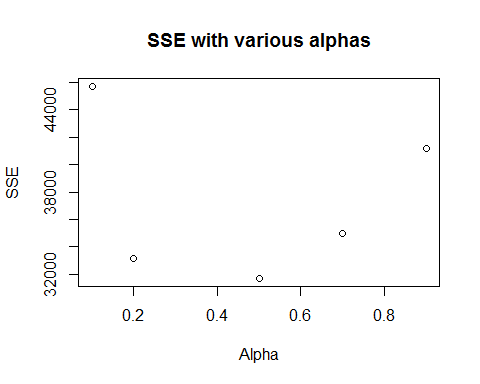
sum((hardcover-fitted(fith5) )^2)

## [1] 45714.82

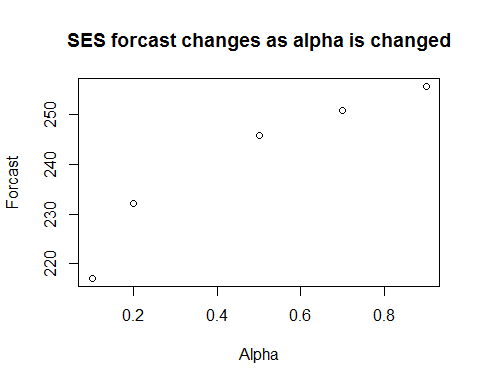
sum((hardcover-fitted(fith6) )^2)

## [1] 41209.53

a = c(0.2, 0.5, 0.7, 0.1, 0.9)  
s = c(33148.16, 31702.6, 34993.95, 45714.82, 41209.53)  
f = c(232.01, 245.73, 250.80, 217.06, 255.64)  
  
plot(a,s, ylab="SSE", xlab="Alpha", main = "SSE with various alphas")



plot(a,f, ylab="Forcast", xlab="Alpha", main = "SES forcast changes as alpha is changed")



fith4 = ses(hardcover, initial="simple", h=3)  
  
sum((hardcover-fitted(fith4) )^2)

## [1] 30758.07

fith4$model

## Simple exponential smoothing   
##   
## Call:  
## ses(y = hardcover, h = 3, initial = "simple")   
##   
## Smoothing parameters:  
## alpha = 0.3473   
##   
## Initial states:  
## l = 139   
##   
## sigma: 32.0198

#(alpha = 0.3473)  
  
  
fith\_o = ses(hardcover, initial="optimal", h=3)  
summary(fith\_o)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = hardcover, h = 3, initial = "optimal")   
##   
## Smoothing parameters:  
## alpha = 0.3283   
##   
## Initial states:  
## l = 149.2836   
##   
## sigma: 31.931  
##   
## AIC AICc BIC   
## 315.8506 316.7737 320.0542   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 9.166918 31.93101 26.7731 2.636328 13.39479 0.7987858  
## ACF1  
## Training set -0.1417817  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 31 239.5602 198.6390 280.4815 176.9766 302.1439  
## 32 239.5602 196.4905 282.6299 173.6908 305.4297  
## 33 239.5602 194.4443 284.6762 170.5613 308.5591

summary(fith4)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = hardcover, h = 3, initial = "simple")   
##   
## Smoothing parameters:  
## alpha = 0.3473   
##   
## Initial states:  
## l = 139   
##   
## sigma: 32.0198  
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 9.72952 32.01982 26.34467 3.104211 13.05063 0.7860035  
## ACF1  
## Training set -0.1629042  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 31 240.3808 199.3457 281.4158 177.6231 303.1385  
## 32 240.3808 196.9410 283.8206 173.9453 306.8162  
## 33 240.3808 194.6625 286.0990 170.4608 310.3008

b1. It appears that 0.5 is the best alpha from the values I have picked. From the second plot we see the same, alpha increase translates in increase in forcast.

c1. Optimization shows alpha = 0.3473 to be the best parameter. Again, we have a lower SSE to confirm the results.

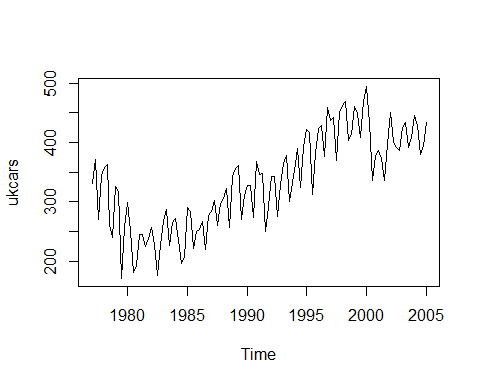
d1. We see a small change in forcast from 239.56 to 40.38

## Ex. HA7.3

### For this exercise, use the quarterly UK passenger vehicle production data from 1977:1--2005:1 (data set ukcars).

#### A. Plot the data and describe the main features of the series.

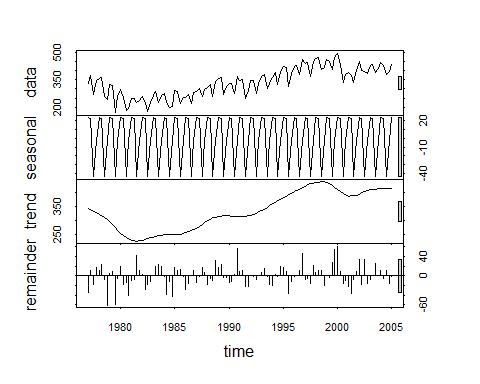
plot(ukcars)



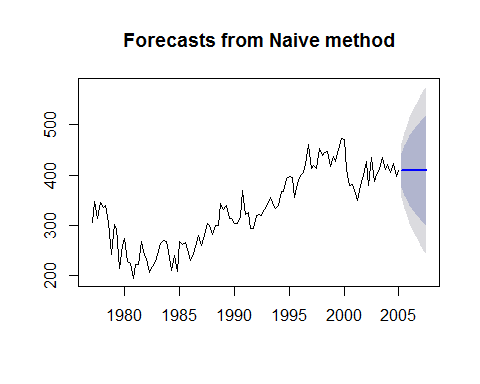
The plot shows an uptrend and clearly present seasonality

#### B. Decompose the series using STL and obtain the seasonally adjusted data.

#decompose  
fit <- stl(ukcars, t.window=15, s.window="periodic", robust=TRUE)  
plot(fit)

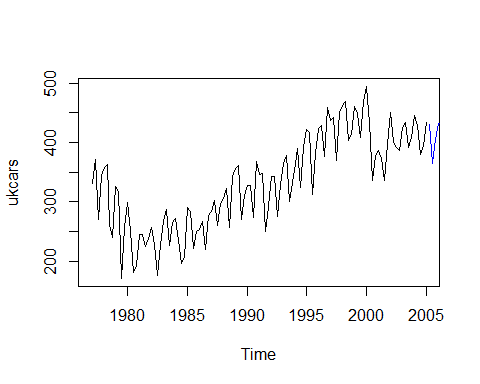


#seasonally adjust  
season = seasadj(fit)  
plot(naive(season))



#### C. Forecast the next two years of the series using an additive damped trend method applied to the seasonally adjusted data. Then reseasonalize the forecasts. Record the parameters of the method and report the RMSE of the one-step forecasts from your method.

#Add damped  
damped <- holt(season, h=8, damped=TRUE)  
#plot(damped)  
  
#reseasonalize  
#fcast <- forecast(fit, method="naive")  
#plot(fcast)  
  
reseason = fit$time.series[110:113,"seasonal"]  
reasesoned\_forcast = damped$mean + reseason  
  
plot(ukcars)  
lines(reasesoned\_forcast, col="blue")



summary(damped)

##   
## Forecast method: Damped Holt's method  
##   
## Model Information:  
## Damped Holt's method   
##   
## Call:  
## holt(y = season, h = 8, damped = TRUE)   
##   
## Smoothing parameters:  
## alpha = 0.5713   
## beta = 1e-04   
## phi = 0.9099   
##   
## Initial states:  
## l = 342.9617   
## b = -10.0627   
##   
## sigma: 25.1951  
##   
## AIC AICc BIC   
## 1275.418 1276.210 1291.782   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 2.571142 25.19512 20.40838 0.3251421 6.522226 0.6650992  
## ACF1  
## Training set 0.03704856  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2005 Q2 407.6783 375.3894 439.9671 358.2967 457.0598  
## 2005 Q3 407.6779 370.4897 444.8661 350.8035 464.5524  
## 2005 Q4 407.6776 366.1631 449.1922 344.1867 471.1686  
## 2006 Q1 407.6774 362.2457 453.1090 338.1956 477.1591  
## 2006 Q2 407.6771 358.6394 456.7149 332.6804 482.6739  
## 2006 Q3 407.6769 355.2800 460.0738 327.5427 487.8111  
## 2006 Q4 407.6767 352.1228 463.2307 322.7143 492.6392  
## 2007 Q1 407.6765 349.1350 466.2180 318.1450 497.2080

#RMSE = 25.19512

#### D. Forecast the next two years of the series using Holt's linear method applied to the seasonally adjusted data. Then reseasonalize the forecasts. Record the parameters of the method and report the RMSE of of the one-step forecasts from your method.

holt\_linear = holt(season, h=8)  
holt\_reseason = holt\_linear$mean + reseason  
  
summary(holt\_linear)

##   
## Forecast method: Holt's method  
##   
## Model Information:  
## Holt's method   
##   
## Call:  
## holt(y = season, h = 8)   
##   
## Smoothing parameters:  
## alpha = 0.5946   
## beta = 0.0017   
##   
## Initial states:  
## l = 339.9795   
## b = 0.6971   
##   
## sigma: 25.3943  
##   
## AIC AICc BIC   
## 1275.197 1275.758 1288.834   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01651047 25.39426 20.07121 -0.5261068 6.46535 0.6541111  
## ACF1  
## Training set 0.0414951  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2005 Q2 408.7673 376.2233 441.3114 358.9955 458.5391  
## 2005 Q3 409.4675 371.5772 447.3578 351.5193 467.4157  
## 2005 Q4 410.1677 367.5724 452.7630 345.0238 475.3116  
## 2006 Q1 410.8679 364.0152 457.7207 339.2129 482.5230  
## 2006 Q2 411.5682 360.7928 462.3435 333.9140 489.2223  
## 2006 Q3 412.2684 357.8327 466.7040 329.0163 495.5205  
## 2006 Q4 412.9686 355.0852 470.8520 324.4435 501.4936  
## 2007 Q1 413.6688 352.5141 474.8235 320.1408 507.1968

#RMSE = 25.39426

#### E. Now use ets() to choose a seasonal model for the data.

fit\_ets = ets(season)  
summary(fit\_ets)

## ETS(A,N,N)   
##   
## Call:  
## ets(y = season)   
##   
## Smoothing parameters:  
## alpha = 0.6143   
##   
## Initial states:  
## l = 319.208   
##   
## sigma: 25.294  
##   
## AIC AICc BIC   
## 1270.303 1270.524 1278.486   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 1.27223 25.29404 20.07755 -0.1385616 6.451837 0.6543177  
## ACF1  
## Training set 0.02840997

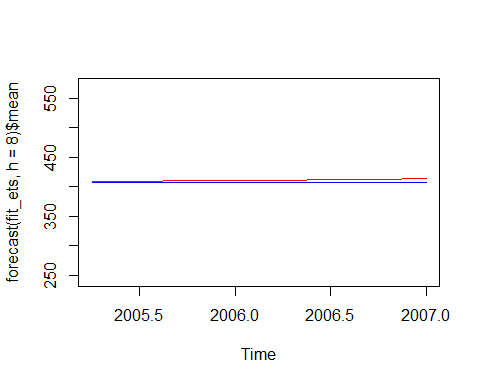
#RMSE = 25.29404

#### F. Compare the RMSE of the fitted model with the RMSE of the model you obtained using an STL decomposition with Holt's method. Which gives the better in-sample fits?

It appears that RMSEs are pretty close and do not show a significant difference.

#### G. Compare the forecasts from the two approaches? Which seems most reasonable?

plot(forecast(fit\_ets, h=8)$mean)  
lines(damped$mean, col="blue")  
lines(holt\_linear$mean, col="red")



There is very little difference in predictions to tell which one is better.