Module 4: Smoothing, Part 2

Xuan Pham

9/10/2019

# Simple Exponential Smoothing (SES)

First, let’s look at the data set for the week. Oil production in Saudi Arabia

library(forecast)

## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':  
## method from   
## fitted.fracdiff fracdiff  
## residuals.fracdiff fracdiff

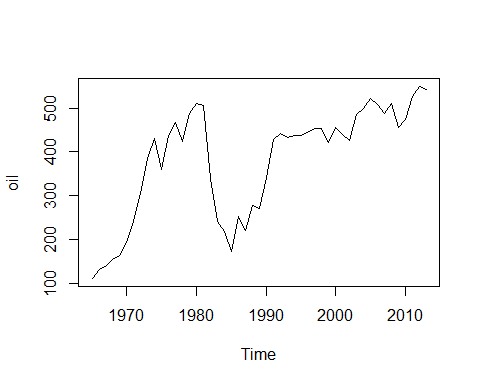
library(fpp2)

## Loading required package: ggplot2

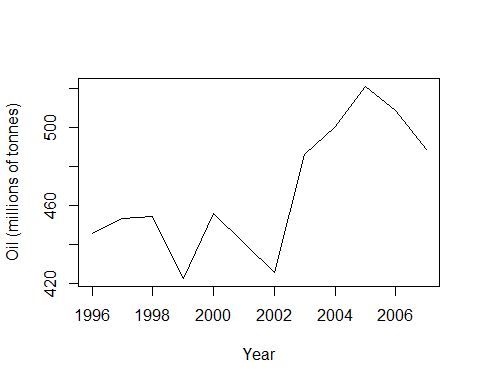
## Loading required package: fma

## Loading required package: expsmooth

plot(oil) #entire series



oildata <- window(oil, start = 1996, end = 2007) #subset of series  
plot(oildata, ylab = "Oil (millions of tonnes)", xlab = "Year")



We know that we can deploy these simple forecasting methods:  
\* mean \* naive  
\* moving average

mean(oildata)

## [1] 466.8927

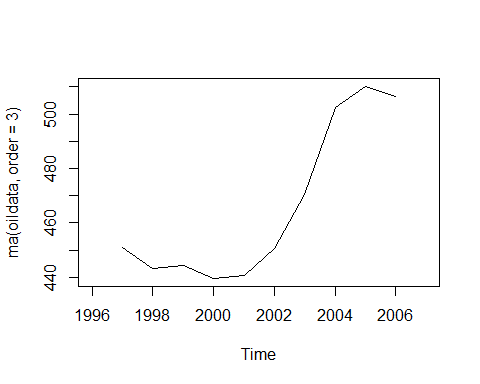
naive(oildata)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2008 488.8889 455.1788 522.5989 437.3338 540.4439  
## 2009 488.8889 441.2157 536.5621 415.9790 561.7987  
## 2010 488.8889 430.5014 547.2764 399.5929 578.1848  
## 2011 488.8889 421.4688 556.3089 385.7787 591.9990  
## 2012 488.8889 413.5109 564.2668 373.6082 604.1695  
## 2013 488.8889 406.3165 571.4613 362.6053 615.1724  
## 2014 488.8889 399.7005 578.0772 352.4870 625.2907  
## 2015 488.8889 393.5425 584.2352 343.0691 634.7086  
## 2016 488.8889 387.7587 590.0190 334.2237 643.5540  
## 2017 488.8889 382.2884 595.4894 325.8575 651.9203

ma(oildata, order=3)

## Time Series:  
## Start = 1996   
## End = 2007   
## Frequency = 1   
## [1] NA 450.9896 443.3279 444.2752 439.6009 440.5394 450.5954  
## [8] 470.6095 502.6367 510.2175 506.3708 NA

plot(ma(oildata, order=3))



What if we do not give equal weights to all observations? What if weights are assigned based on whether we think the more recent history or distant past is more important?

fit1 <- ses(oildata, alpha = 0.2, initial = "simple", h = 3)  
fit2 <- ses(oildata, alpha = 0.6, initial = "simple", h = 3)  
fit3 <- ses(oildata, h = 3) #note no value for alpha and no initial value. The algorithm will calculate alpha as part of the optimization.   
  
#what kind of objects are these?  
class(fit3)

## [1] "forecast"

#what is in them  
summary(fit3)

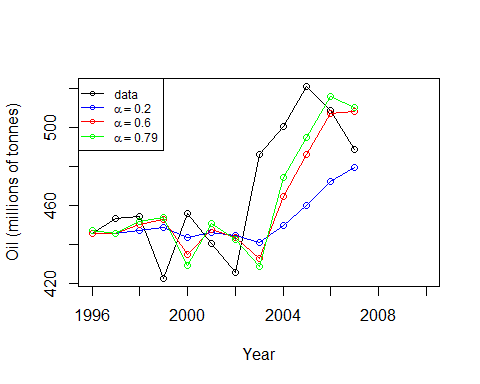
##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = oildata, h = 3)   
##   
## Smoothing parameters:  
## alpha = 0.7956   
##   
## Initial states:  
## l = 446.7948   
##   
## sigma: 27.1495  
##   
## AIC AICc BIC   
## 112.8636 115.8636 114.3183   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 4.86894 24.78395 19.65328 0.8653916 4.179747 0.923691  
## ACF1  
## Training set -0.0431624  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2008 493.2803 458.4869 528.0737 440.0684 546.4923  
## 2009 493.2803 448.8182 537.7424 425.2814 561.2792  
## 2010 493.2803 440.9050 545.6557 413.1791 573.3815

summary(fit2)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = oildata, h = 3, initial = "simple", alpha = 0.6)   
##   
## Smoothing parameters:  
## alpha = 0.6   
##   
## Initial states:  
## l = 445.3641   
##   
## sigma: 25.2026  
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 7.121036 25.2026 19.57078 1.311303 4.138815 0.9198136  
## ACF1  
## Training set 0.09497862  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2008 496.6356 464.3371 528.9340 447.2394 546.0317  
## 2009 496.6356 458.9694 534.3017 439.0302 554.2409  
## 2010 496.6356 454.2766 538.9945 431.8531 561.4180

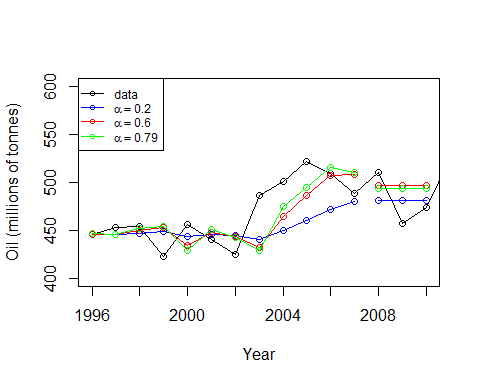
Now let’s look at some plots.

plot(fit1, PI=FALSE, ylab="Oil (millions of tonnes)",  
 xlab="Year", main="", fcol="white", type="o")  
lines(fitted(fit1), col="blue", type="o")  
lines(fitted(fit2), col="red", type="o")  
lines(fitted(fit3), col="green", type="o")  
  
legend("topleft",lty=1, col=c(1,"blue","red","green"), cex = 0.75,   
 c("data", expression(alpha == 0.2), expression(alpha == 0.6),  
 expression(alpha == 0.79)),pch=1)



What if we want to use our SES methods to forecast three years ahead? 2008-2010

fit1 <- ses(oildata, alpha = 0.2, initial = "simple", h = 3)  
fit2 <- ses(oildata, alpha = 0.6, initial = "simple", h = 3)  
fit3 <- ses(oildata, h = 3)  
  
plot(fit1, PI=FALSE, ylim = c(400, 600), ylab="Oil (millions of tonnes)",  
xlab="Year", main="", fcol="white", type="o")  
#this just plots the data through 2007 because that's what we ran the model over  
#we have more data but have to plot it seperately if we want to compare the forecast with the actual   
# add these lines for putting the actual data past 2007  
test.oil <- window(oil, start = 2007)  
lines(test.oil, col = "black", type = "o")  
lines(fitted(fit1), col="blue", type="o")  
lines(fitted(fit2), col="red", type="o")  
lines(fitted(fit3), col="green", type="o")  
lines(fit1$mean, col="blue", type="o")  
lines(fit2$mean, col="red", type="o")  
lines(fit3$mean, col="green", type="o")  
legend("topleft",lty=1, cex =0.75, col=c(1,"blue","red","green"),  
c("data", expression(alpha == 0.2), expression(alpha == 0.6),  
expression(alpha == 0.79)),pch=1)



# Holt’s Linear Trend

The problem with SES is that it does not forecast anything else but a flat line. What if we expect forecasts to have a trend?

fith <- holt(oildata, alpha=0.8, beta=0.2, initial="simple", h=3)   
class(fith)

## [1] "forecast"

summary(fith)

##   
## Forecast method: Holt's method  
##   
## Model Information:  
## Holt's method   
##   
## Call:  
## holt(y = oildata, h = 3, initial = "simple", alpha = 0.8, beta = 0.2)   
##   
## Smoothing parameters:  
## alpha = 0.8   
## beta = 0.2   
##   
## Initial states:  
## l = 445.3641   
## b = 7.8309   
##   
## sigma: 26.6089  
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -2.062825 26.60893 21.6139 -0.5912386 4.622665 1.015839  
## ACF1  
## Training set -0.02963702  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2008 499.5248 465.4241 533.6255 447.3722 551.6773  
## 2009 503.3951 455.1694 551.6208 429.6402 577.1499  
## 2010 507.2653 444.0179 570.5128 410.5368 603.9939

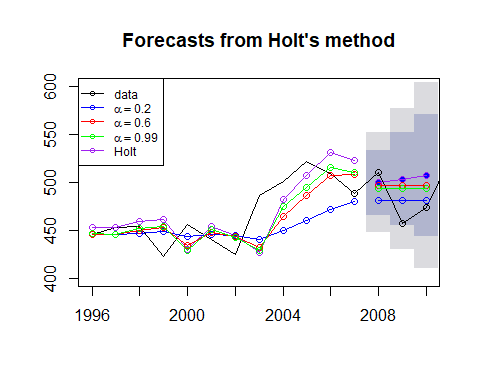
oildata[1]

## [1] 445.3641

Notice that it shows the parameter settings we gave it. We now have a setting for the effect of the trend. What’s the initial value? It’s the first value of the oildata series - because we set the initialization routine = “simple” fith$mean.

Adding these lines to our chart. What is the effect of adding trend?

plot(fith, PI=TRUE, ylim = c(400, 600))  
lines(fitted(fith), type = "o", col="purple")   
lines(fith$mean, col="purple", type="o")   
lines(test.oil, col = "black", type = "o")  
lines(fitted(fit1), col="blue", type="o")  
lines(fitted(fit2), col="red", type="o")  
lines(fitted(fit3), col="green", type="o")  
lines(fit1$mean, col="blue", type="o")  
lines(fit2$mean, col="red", type="o")  
lines(fit3$mean, col="green", type="o")  
legend("topleft",lty=1, cex = 0.75, col=c(1,"blue","red","green", "purple"),  
 c("data", expression(alpha == 0.2), expression(alpha == 0.6),  
 expression(alpha == 0.99),"Holt"),pch=1)



That was the additive trend Holt’s method - let’s make it muliplicative.

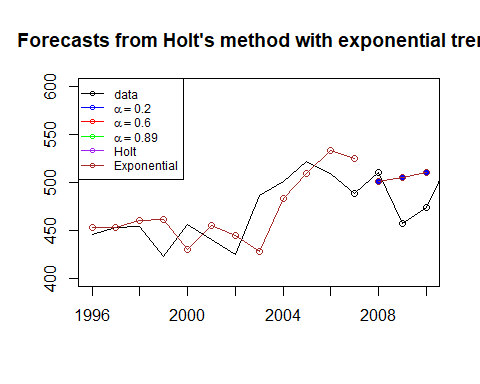
fith2 <- holt(oildata,alpha=0.8,beta=0.2,initial="simple",exponential=TRUE,h=3)   
summary(fith2)

##   
## Forecast method: Holt's method with exponential trend  
##   
## Model Information:  
## Holt's method with exponential trend   
##   
## Call:  
## holt(y = oildata, h = 3, initial = "simple", exponential = TRUE,   
##   
## Call:  
## alpha = 0.8, beta = 0.2)   
##   
## Smoothing parameters:  
## alpha = 0.8   
## beta = 0.2   
##   
## Initial states:  
## l = 445.3641   
## b = 1.0176   
##   
## sigma: 0.0588  
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -2.704003 26.81176 21.7455 -0.7209538 4.651922 1.022024  
## ACF1  
## Training set -0.02855242  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2008 500.7304 461.7457 537.0310 442.4334 557.2575  
## 2009 505.5078 452.5839 557.6012 426.4087 588.3002  
## 2010 510.3308 441.2425 580.9792 403.9723 619.1051

We specified alpha and beta and told it to start with the initial value and the initial trend.

Now add that line to the chart:

plot(fith2, PI=FALSE, ylim = c(400, 600))  
lines(test.oil, col = "black", type = "o")  
lines(fitted(fith2), col="brown", type = "o")  
lines(fith2$mean, col="brown", type="o")  
legend("topleft", cex =0.75, lty=1, col=c(1,"blue","red","green", "purple", "brown"),  
 c("data", expression(alpha == 0.2), expression(alpha == 0.6),  
 expression(alpha == 0.89),"Holt","Exponential"),pch=1)



Let’s look closer at the components of each forecast output.

fith2$model

## Holt's method with exponential trend   
##   
## Call:  
## holt(y = oildata, h = 3, initial = "simple", exponential = TRUE,   
##   
## Call:  
## alpha = 0.8, beta = 0.2)   
##   
## Smoothing parameters:  
## alpha = 0.8   
## beta = 0.2   
##   
## Initial states:  
## l = 445.3641   
## b = 1.0176   
##   
## sigma: 0.0588

fith2$x

## Time Series:  
## Start = 1996   
## End = 2007   
## Frequency = 1   
## [1] 445.3641 453.1950 454.4096 422.3789 456.0371 440.3866 425.1944  
## [8] 486.2052 500.4291 521.2759 508.9476 488.8889

fith2$fitted

## Time Series:  
## Start = 1996   
## End = 2007   
## Frequency = 1   
## [1] 453.1950 453.5314 459.9023 461.2979 429.7498 454.7545 444.9082  
## [8] 427.6789 483.2416 509.0280 533.4373 524.4355

fith2$mean

## Time Series:  
## Start = 2008   
## End = 2010   
## Frequency = 1   
## [1] 500.7304 505.5078 510.3308

What about Holt’s trend with a damped factor?

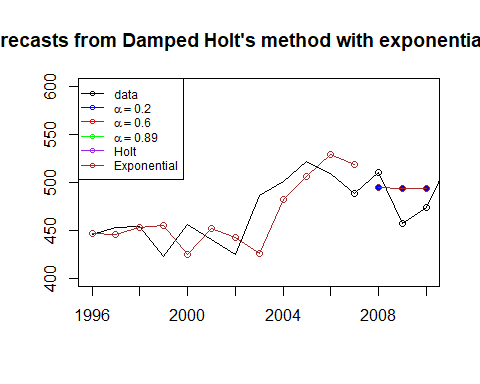
fith3 <- holt(oildata,alpha=0.8,beta=0.2,initial="simple",exponential=TRUE,  
 damped=TRUE, phi=NULL, h=3) #setting phi=NULL allows R to automatically come up with the best damped factor.

## Warning in holt(oildata, alpha = 0.8, beta = 0.2, initial = "simple",  
## exponential = TRUE, : Damped Holt's method requires optimal initialization

summary(fith3)

##   
## Forecast method: Damped Holt's method with exponential trend  
##   
## Model Information:  
## Damped Holt's method with exponential trend   
##   
## Call:  
## holt(y = oildata, h = 3, damped = TRUE, initial = "simple", exponential = TRUE,   
##   
## Call:  
## alpha = 0.8, beta = 0.2, phi = NULL)   
##   
## Smoothing parameters:  
## alpha = 0.8   
## beta = 0.2   
## phi = 0.8002   
##   
## Initial states:  
## l = 446.3   
## b = 1.0018   
##   
## sigma: 0.0757  
##   
## AIC AICc BIC   
## 116.7636 122.4779 118.7032   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 1.712068 25.82345 20.44594 0.2398576 4.357572 0.9609457  
## ACF1  
## Training set -0.08718763  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2008 494.2701 446.8503 541.6940 422.2369 568.4969  
## 2009 493.7910 430.2403 561.3054 397.4882 599.6116  
## 2010 493.4080 411.7195 579.3850 371.6049 626.4850

plot(fith3, PI=FALSE, ylim = c(400, 600))  
lines(test.oil, col = "black", type = "o")  
lines(fitted(fith3), col="brown", type = "o")  
lines(fith3$mean, col="brown", type="o")  
legend("topleft", cex =0.75, lty=1, col=c(1,"blue","red","green", "purple", "brown"),  
 c("data", expression(alpha == 0.2), expression(alpha == 0.6),  
 expression(alpha == 0.89),"Holt","Exponential"),pch=1)

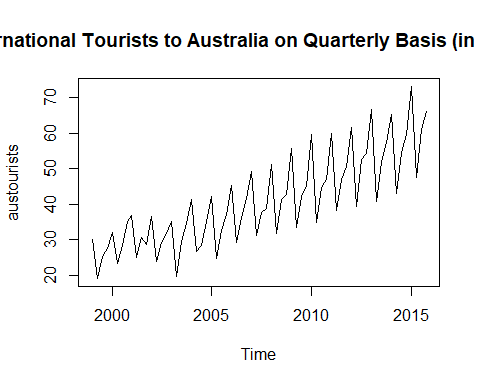


#Holt-Winters’ Method

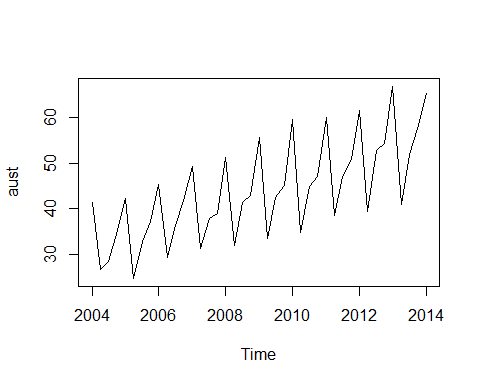
But what if there is seasonality in the time series?

First, we need to find a different time series (with seasonality) to examine. Here’s a time series of tourists coming to Australia.

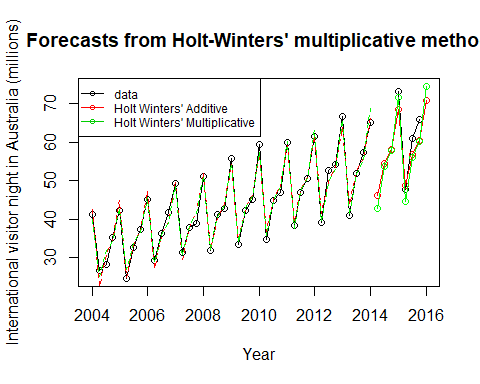
#?austourists  
plot(austourists, main = "International Tourists to Australia on Quarterly Basis (in Millions)")



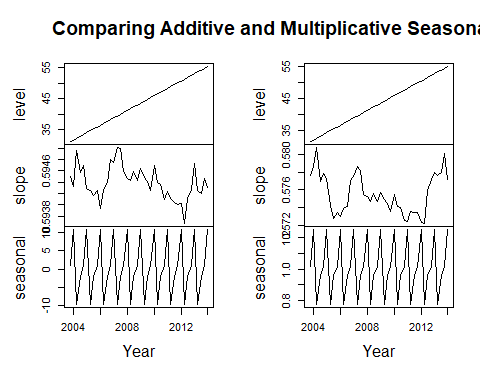
aust <- window(austourists,start=2004, end=2014)  
plot(aust)



#let's make the series a bit shorter for manageability purpose.   
fit1hw <- hw(aust,seasonal="additive")  
fit2hw <- hw(aust,seasonal="multiplicative")  
  
#Can you take a look at Holt Winters method with damped factor for trend?  
  
test.aust <- window(austourists, start=2015)  
plot(fit2hw,ylab="International visitor night in Australia (millions)",  
 PI=FALSE, type="o", fcol="white", xlab="Year")  
lines(test.aust, col = "black", type = "o")  
lines(fitted(fit1hw), col="red", lty=2)  
lines(fitted(fit2hw), col="green", lty=2)  
lines(fit1hw$mean, type="o", col="red")  
lines(fit2hw$mean, type="o", col="green")  
legend("topleft",lty=1, cex = 0.75, pch=1, col=1:3,   
 c("data","Holt Winters' Additive","Holt Winters' Multiplicative"))

 We can pull the different pieces out of the forecast object and compare them across specifications. What’s the difference between additive seasonal and multiplicative seasonal?

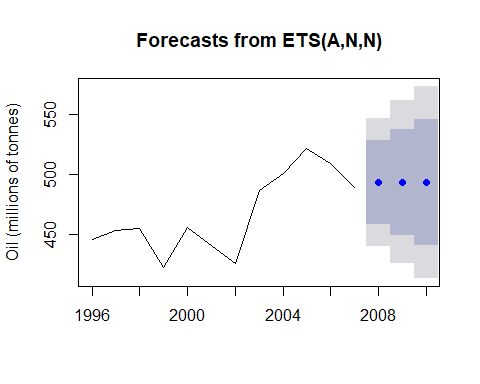
states <- cbind(fit1hw$model$states[,1:3],fit2hw$model$states[,1:3])  
colnames(states) <- c("level","slope","seasonal","level","slope","seasonal")  
plot(states, xlab="Year", main = "Comparing Additive and Multiplicative Seasonals")



# ETS (Error-Trend-Seasonal) Statistical Models

Let’s revisit the oil data - we originally used the SES method on it which is equivalent to ETS(A,N,N)

oildata <- window(oil, start = 1996, end = 2007)  
fita <- ets(oildata, model = "ANN")  
plot(forecast(fita, h=3), ylab="Oil (millions of tonnes)")



summary(fita)

## ETS(A,N,N)   
##   
## Call:  
## ets(y = oildata, model = "ANN")   
##   
## Smoothing parameters:  
## alpha = 0.7958   
##   
## Initial states:  
## l = 446.7849   
##   
## sigma: 27.1495  
##   
## AIC AICc BIC   
## 112.8636 115.8636 114.3183   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 4.868242 24.78395 19.65274 0.8652851 4.179647 0.923666  
## ACF1  
## Training set -0.04331639

ls(fita) #list names of the objects in the specified environment

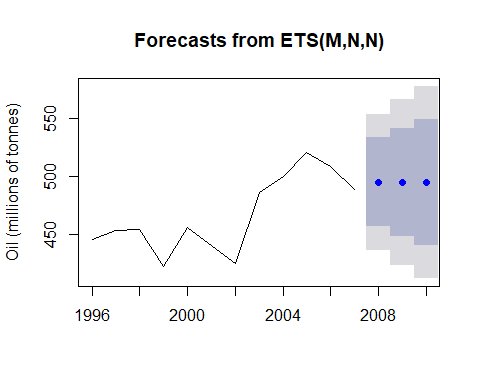
## [1] "aic" "aicc" "amse" "bic" "call"   
## [6] "components" "fit" "fitted" "initstate" "loglik"   
## [11] "m" "method" "mse" "par" "residuals"   
## [16] "series" "sigma2" "states" "x"

fita$par

## alpha l   
## 0.7958197 446.7849382

What if we used the ETS(M,N,N) model?

oildata <- window(oil, start = 1996, end = 2007)  
fitm <- ets(oildata, model = "MNN")  
plot(forecast(fitm, h=3), ylab="Oil (millions of tonnes)")



summary(fitm)

## ETS(M,N,N)   
##   
## Call:  
## ets(y = oildata, model = "MNN")   
##   
## Smoothing parameters:  
## alpha = 0.7022   
##   
## Initial states:  
## l = 445.538   
##   
## sigma: 0.0606  
##   
## AIC AICc BIC   
## 113.5169 116.5169 114.9716   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 5.889318 24.87807 19.5004 1.068927 4.137105 0.9165061  
## ACF1  
## Training set 0.02210889

ls(fitm) #list names of the objects in the specified environment

## [1] "aic" "aicc" "amse" "bic" "call"   
## [6] "components" "fit" "fitted" "initstate" "loglik"   
## [11] "m" "method" "mse" "par" "residuals"   
## [16] "series" "sigma2" "states" "x"

fitm$par

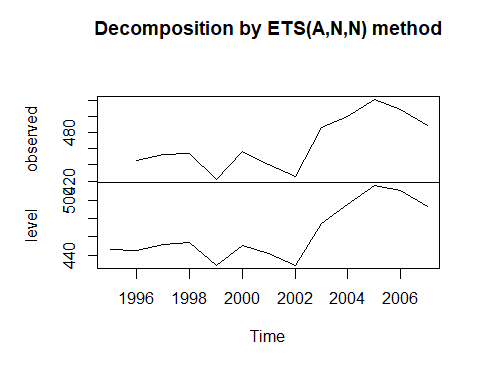
## alpha l   
## 0.7021876 445.5380329

But are either of these the right model for these data? What if we let the ETS code decide?

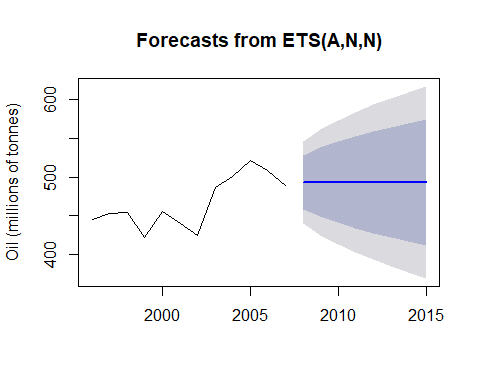
oildata <- window(oil, start = 1996, end = 2007)  
fit <- ets(oildata)  
summary(fit)

## ETS(A,N,N)   
##   
## Call:  
## ets(y = oildata)   
##   
## Smoothing parameters:  
## alpha = 0.7958   
##   
## Initial states:  
## l = 446.7849   
##   
## sigma: 27.1495  
##   
## AIC AICc BIC   
## 112.8636 115.8636 114.3183   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 4.868242 24.78395 19.65274 0.8652851 4.179647 0.923666  
## ACF1  
## Training set -0.04331639

plot(fit)



plot(forecast(fit, h = 8), ylab = "Oil (millions of tonnes)")



The model chose additive. Compare the AIC for the two we examined.

fita$aic

## [1] 112.8636

fitm$aic

## [1] 113.5169

Yup - the AIC for the additive model is lower so it is the “better” model–but not by much.

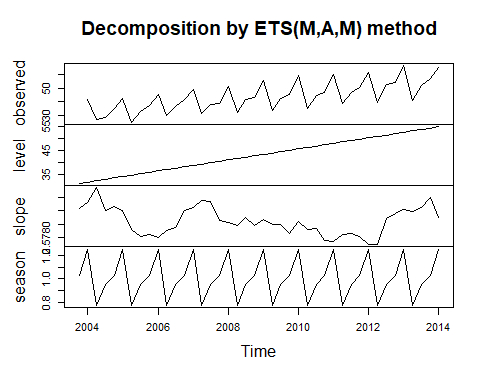
How about a more interesting example? Remember the Australian tourism time series?

Very strongly seasonal. Probably not a good candidate for the ETS(A,N,N) model. Let’s see:

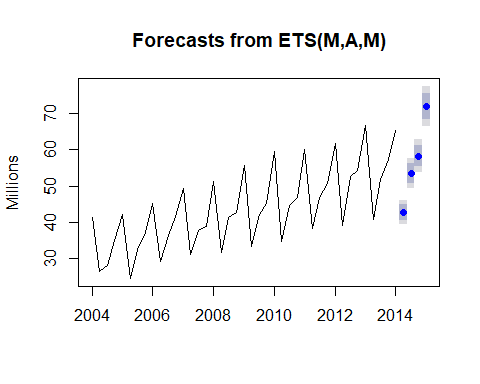
fit <- ets(aust)  
summary(fit)

## ETS(M,A,M)   
##   
## Call:  
## ets(y = aust)   
##   
## Smoothing parameters:  
## alpha = 1e-04   
## beta = 1e-04   
## gamma = 2e-04   
##   
## Initial states:  
## l = 31.3602   
## b = 0.5784   
## s = 1.0267 0.9504 0.77 1.2528  
##   
## sigma: 0.0396  
##   
## AIC AICc BIC   
## 204.1782 209.9846 219.6003   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.04042055 1.460647 1.109073 -0.238826 2.686734 0.4153527  
## ACF1  
## Training set -0.08350248

plot(fit)



plot(forecast(fit, h = 4), ylab = "Millions")

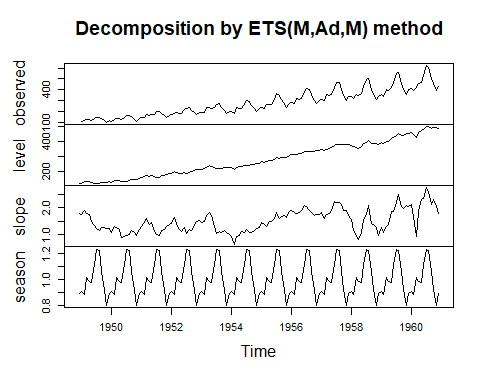


What about our ubiquitous air passengers data?

ap <- window(AirPassengers)  
fit <- ets(ap)  
summary(fit)

## ETS(M,Ad,M)   
##   
## Call:  
## ets(y = ap)   
##   
## Smoothing parameters:  
## alpha = 0.7096   
## beta = 0.0204   
## gamma = 1e-04   
## phi = 0.98   
##   
## Initial states:  
## l = 120.9939   
## b = 1.7705   
## s = 0.8944 0.7993 0.9217 1.0592 1.2203 1.2318  
## 1.1105 0.9786 0.9804 1.011 0.8869 0.9059  
##   
## sigma: 0.0392  
##   
## AIC AICc BIC   
## 1395.166 1400.638 1448.623   
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 1.567359 10.74726 7.791605 0.4357799 2.857917 0.2432573  
## ACF1  
## Training set 0.03945056

plot(fit)



plot(forecast(fit, h = 8), ylab = "Millions")

