Week2\_7.1\_7.2\_7.3\_Chunjie\_Nan

Chunjie Nan

6/12/2021

7.1. Consider the pigs series — the number of pigs slaughtered in Victoria each month.

library(forecast)

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

library(fpp2)

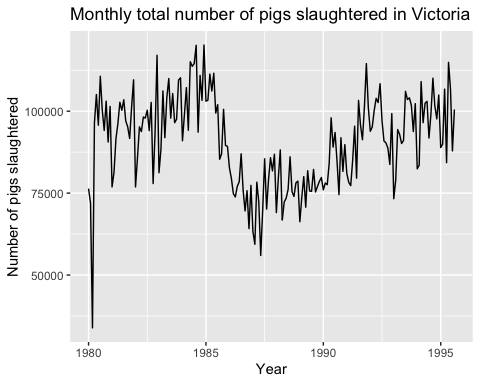
## ── Attaching packages ────────────────────────────────────────────── fpp2 2.4 ──

## ✓ ggplot2 3.3.3 ✓ expsmooth 2.3   
## ✓ fma 2.4

##

This is Monthly total number of pigs slaughtered in Victoria, Australia from Jan 1980 to Aug 1995.

autoplot(pigs)+  
 xlab("Year")+  
 ylab("Number of pigs slaughtered")+  
 ggtitle("Monthly total number of pigs slaughtered in Victoria")



1. Use the ses() function in R to find the optimal values of α and ℓ0, and generate forecasts for the next four months.

fc\_pigs<-ses(pigs, h=4)  
summary(fc\_pigs)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = pigs, h = 4)   
##   
## Smoothing parameters:  
## alpha = 0.2971   
##   
## Initial states:  
## l = 77260.0561   
##   
## sigma: 10308.58  
##   
## AIC AICc BIC   
## 4462.955 4463.086 4472.665   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 385.8721 10253.6 7961.383 -0.922652 9.274016 0.7966249 0.01282239  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 1995 98816.41 85605.43 112027.4 78611.97 119020.8  
## Oct 1995 98816.41 85034.52 112598.3 77738.83 119894.0  
## Nov 1995 98816.41 84486.34 113146.5 76900.46 120732.4  
## Dec 1995 98816.41 83958.37 113674.4 76092.99 121539.8

round(accuracy(fc\_pigs),3)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 385.872 10253.6 7961.383 -0.923 9.274 0.797 0.013

With summary of simple smoothing exponential smoothing, alpha = 0.2971, and initial states =77260.0561 The forecast from September 1995 to December 1995, the forecast gives flat number of 98816.41 pigs.

1. Compute a 95% prediction interval for the first forecast using y ± 1.96s where is the standard deviation of the residuals. Compare your interval with the interval produced by R.

fc\_pigs

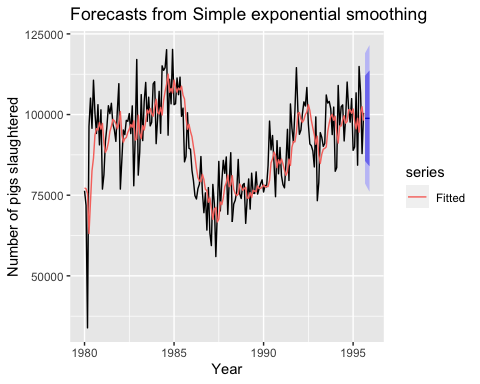
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 1995 98816.41 85605.43 112027.4 78611.97 119020.8  
## Oct 1995 98816.41 85034.52 112598.3 77738.83 119894.0  
## Nov 1995 98816.41 84486.34 113146.5 76900.46 120732.4  
## Dec 1995 98816.41 83958.37 113674.4 76092.99 121539.8

sdr<-sd(fc\_pigs$residuals)\*1.96  
manual\_lo95<-fc\_pigs$mean[1] - sdr  
manual\_hi95<-fc\_pigs$mean[1] + sdr  
lo\_hi95<-c(manual\_lo95, manual\_hi95)  
lo\_hi95

## [1] 78679.97 118952.84

autoplot(fc\_pigs)+   
 autolayer(fc\_pigs$fitted, series = "Fitted",PI=F)+  
 ylab("Number of pigs slaughtered")+  
 xlab("Year")

## Warning: Ignoring unknown parameters: PI

 The manual computation and produced by R produced very similar results.

7.2. Write your own function to implement simple exponential smoothing. The function should take arguments y (the time series), alpha (the smoothing parameter α) and level (the initial levelℓ0 ). It should return the forecast of the next observation in the series. Does it give the same forecast as ses() ?

manual\_ses <- function(y, alpha, initiate){  
 y\_hat <- initiate  
 for(index in 1:length(y)){  
 y\_hat <- alpha\*y[index] + (1 - alpha)\*y\_hat   
 }  
 cat("manual calculation: ",  
 as.character(y\_hat),  
 sep = "\n")  
}  
  
  
  
summary(fc\_pigs)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = pigs, h = 4)   
##   
## Smoothing parameters:  
## alpha = 0.2971   
##   
## Initial states:  
## l = 77260.0561   
##   
## sigma: 10308.58  
##   
## AIC AICc BIC   
## 4462.955 4463.086 4472.665   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 385.8721 10253.6 7961.383 -0.922652 9.274016 0.7966249 0.01282239  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 1995 98816.41 85605.43 112027.4 78611.97 119020.8  
## Oct 1995 98816.41 85034.52 112598.3 77738.83 119894.0  
## Nov 1995 98816.41 84486.34 113146.5 76900.46 120732.4  
## Dec 1995 98816.41 83958.37 113674.4 76092.99 121539.8

manual\_ses(pigs,0.2971,77260.0561)

## manual calculation:   
## 98816.4543875089

after making function, use summary(pigs) find out alpha and initial level, and plug into the manual function. The calculation result is almost the same with the result from ses().

7.3. Modify your function from the previous exercise to return the sum of squared errors rather than the forecast of the next observation. Then use the optim() function to find the optimal values of α and ℓ0 . Do you get the same values as the ses() function?

fc\_ausbeer<-ses(ausbeer)  
  
SES <- function(pars = c(alpha, l0), y){  
 error <- 0  
 SSE <- 0  
 alpha <- pars[1]  
 l0 <- pars[2]  
 y\_hat <- l0  
   
 for(index in 1:length(y)){  
 error <- y[index] - y\_hat  
 SSE <- SSE + error^2  
   
 y\_hat <- alpha\*y[index] + (1 - alpha)\*y\_hat   
 }  
   
 return(SSE)  
}  
  
opt\_pigs <- optim(par = c(0.5, pigs[1]), y = pigs, fn = SES)  
as.character(opt\_pigs$par[1]) #alpha = 0.299

## [1] "0.299008094014243"

as.character(opt\_pigs$par[2]) # initial = 76379.365

## [1] "76379.2653476235"

summary(fc\_pigs) # alpha = 0.297, initial = 77260.056

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = pigs, h = 4)   
##   
## Smoothing parameters:  
## alpha = 0.2971   
##   
## Initial states:  
## l = 77260.0561   
##   
## sigma: 10308.58  
##   
## AIC AICc BIC   
## 4462.955 4463.086 4472.665   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 385.8721 10253.6 7961.383 -0.922652 9.274016 0.7966249 0.01282239  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 1995 98816.41 85605.43 112027.4 78611.97 119020.8  
## Oct 1995 98816.41 85034.52 112598.3 77738.83 119894.0  
## Nov 1995 98816.41 84486.34 113146.5 76900.46 120732.4  
## Dec 1995 98816.41 83958.37 113674.4 76092.99 121539.8

opt\_ausbeer <- optim(par = c(0.5, ausbeer[1]), y = ausbeer, fn = SES)  
as.character(opt\_ausbeer$par[1]) # alpha = 0.148

## [1] "0.148803623284766"

as.character(opt\_ausbeer$par[2]) # initial = 259.658

## [1] "259.658459712619"

summary(fc\_ausbeer) # alpha = 0.149, initial = 259.6592

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = ausbeer)   
##   
## Smoothing parameters:  
## alpha = 0.1489   
##   
## Initial states:  
## l = 259.6592   
##   
## sigma: 50.8804  
##   
## AIC AICc BIC   
## 2891.063 2891.175 2901.216   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 4.980069 50.64645 40.86583 0.1793346 9.618061 2.634917 -0.04443204  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2010 Q3 421.3136 356.1078 486.5195 321.5899 521.0374  
## 2010 Q4 421.3136 355.3889 487.2384 320.4905 522.1368  
## 2011 Q1 421.3136 354.6778 487.9495 319.4029 523.2244  
## 2011 Q2 421.3136 353.9742 488.6531 318.3268 524.3005  
## 2011 Q3 421.3136 353.2778 489.3495 317.2618 525.3655  
## 2011 Q4 421.3136 352.5885 490.0388 316.2077 526.4196  
## 2012 Q1 421.3136 351.9061 490.7212 315.1639 527.4634  
## 2012 Q2 421.3136 351.2303 491.3970 314.1304 528.4969  
## 2012 Q3 421.3136 350.5609 492.0664 313.1067 529.5206  
## 2012 Q4 421.3136 349.8979 492.7294 312.0926 530.5347

According to the alpha and initial as above, the alpha and initial are almost the same to the one from ses function except for the ℓ0 from pigs data, but they are still closed to each other.