Time Series Analysis

Exponential Smoothing in R

Dean Marchiori
Senior Analyst- Wealth Analytics

Introduction

This pack walks through an example of time series analysis using an exponential smoothing method known as 'Holt-Winters Filtering' in R

This type of analysis uses historical or 'time series' data to provide a prediction of future events, such as:

- Monthly sales
- Inbound call centre data
- New account applications
- Daily trading activity

In a simple model, future events can be predicted by looking at the moving average. However exponential smoothing adds smoothing factors to assign exponentially decreasing weights over time, in addition it can handle the impact of long term trends and even seasonal spikes.

Air Passenger data example

Lets look at a sample data set of Air Passengers from 1949 - 1960

```
data(AirPassengers)
x <- as.data.frame(AirPassengers)
head(x)
```

```
##     x
## 1 112
## 2 118
## 3 132
## 4 129
## 5 121
## 6 135
```

Create time series data

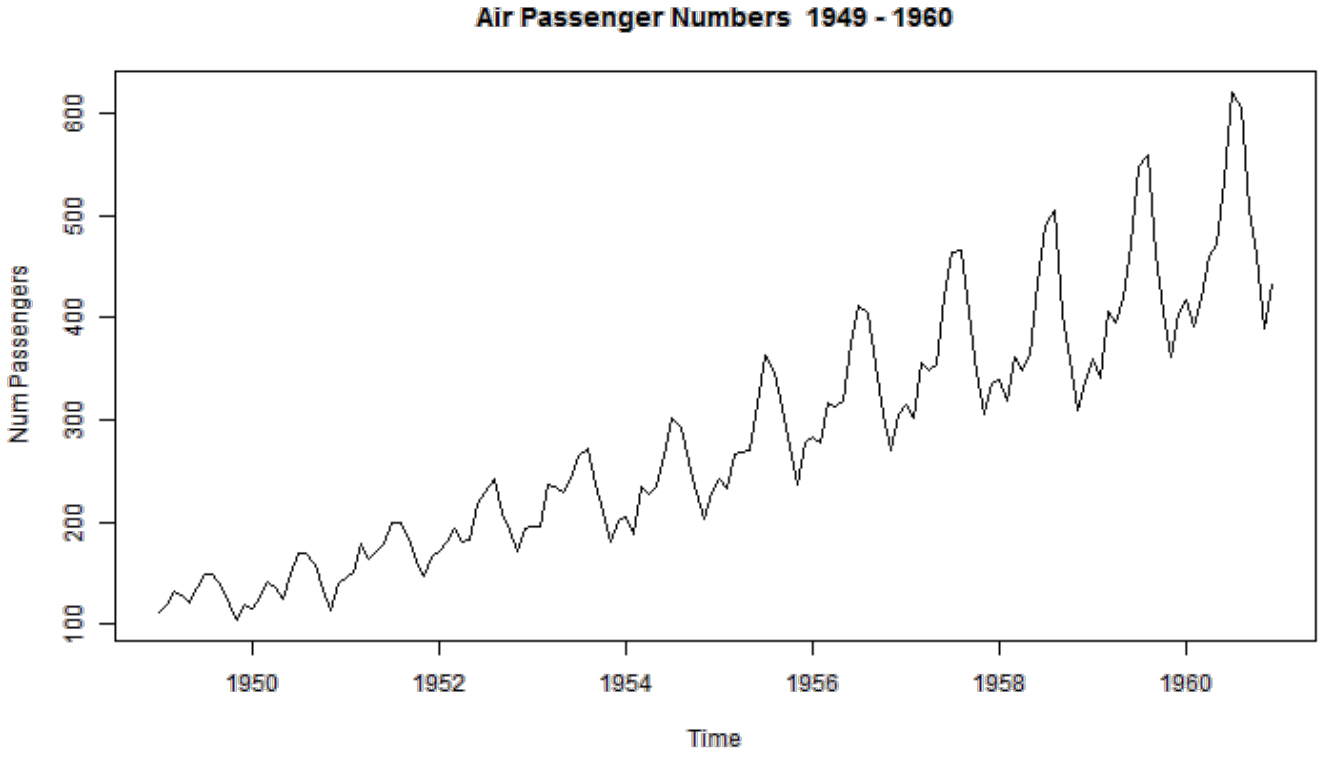
We first need to convert this data frame to an R 'time series' object

```
AirPsgr <- ts(data = AirPassengers, start = c(1949,1),frequency = 12)
AirPsgr
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

Plot of the data

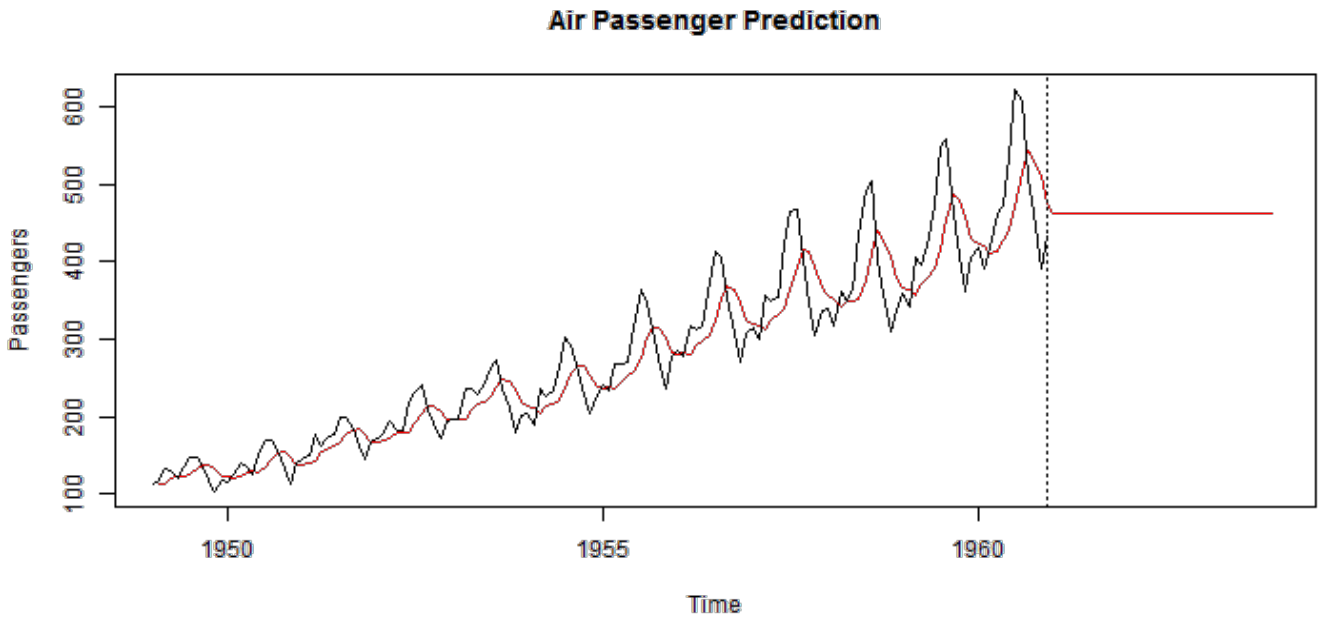
```
plot(AirPsgr, main = "Air Passenger Numbers  1949 - 1960", ylab = "Num Passengers")
```



There is clearly a trend and seaonal peaks and troughs.

Single Exponential Smoothing

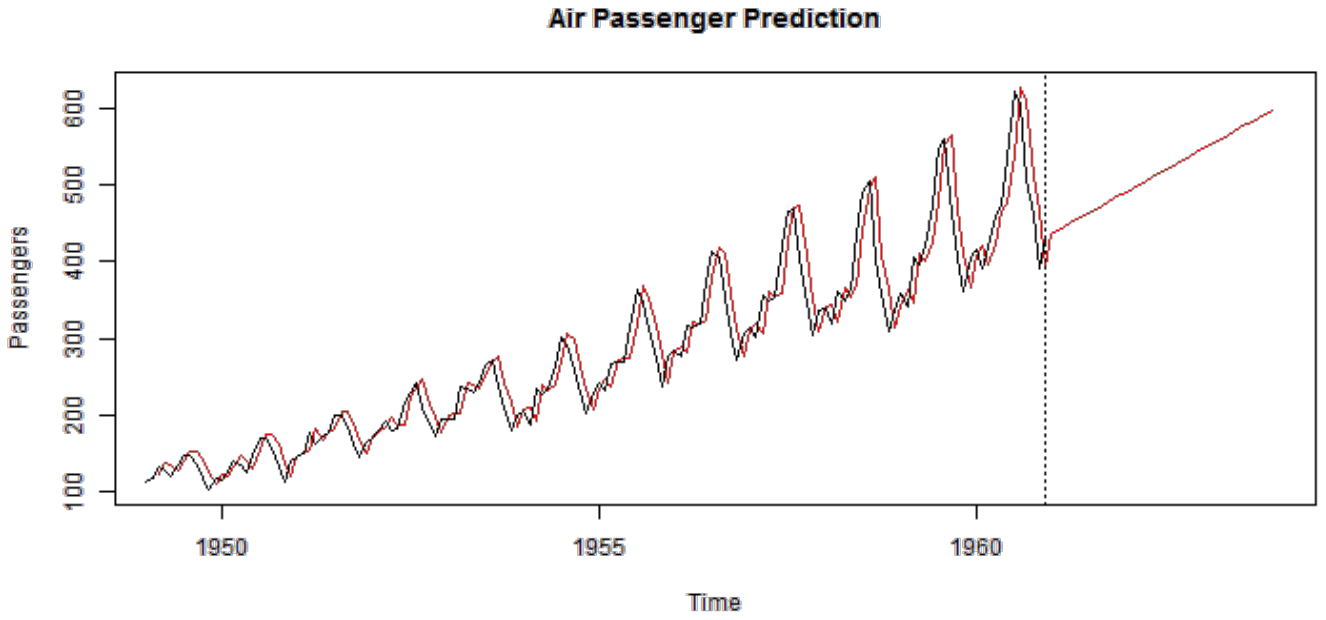
```
x <- HoltWinters(x = AirPsgr, alpha = 0.3, beta = FALSE, gamma = FALSE)
prediction <- predict(object = x, n.ahead = 36)
plot(x,prediction, main = "Air Passenger Prediction", ylab = "Passengers")
```



This is a poor prediction as the use of a straight line prediction is limiting for trending data and doesnt account for seasonal spikes.

Exponential smoothing using a trend factor

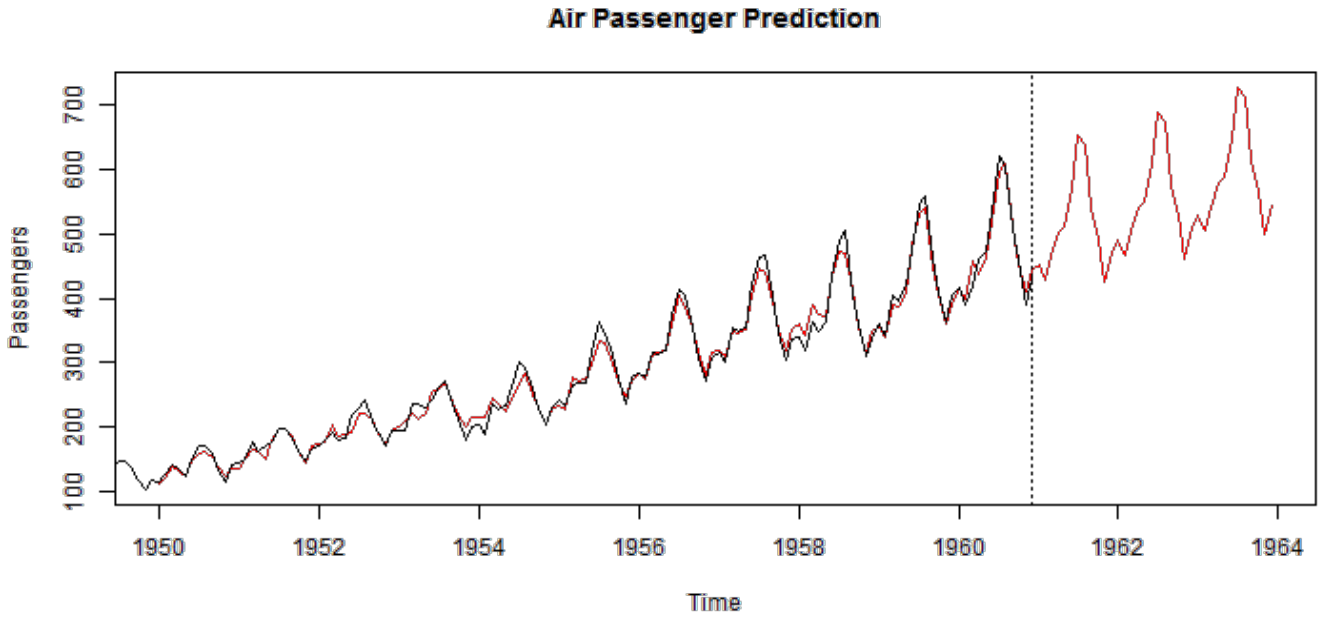
```
x <- HoltWinters(x = AirPsgr, gamma = FALSE)
prediction <- predict(object = x, n.ahead = 36)
plot(x,prediction, main = "Air Passenger Prediction", ylab = "Passengers")
```



Leaving out the 'Alpha' & 'Beta' the package now optimises the trend smoothing factor. This provides a more realistic prediction of the trending data.

Exponential smoothing using trend and seasonality factors

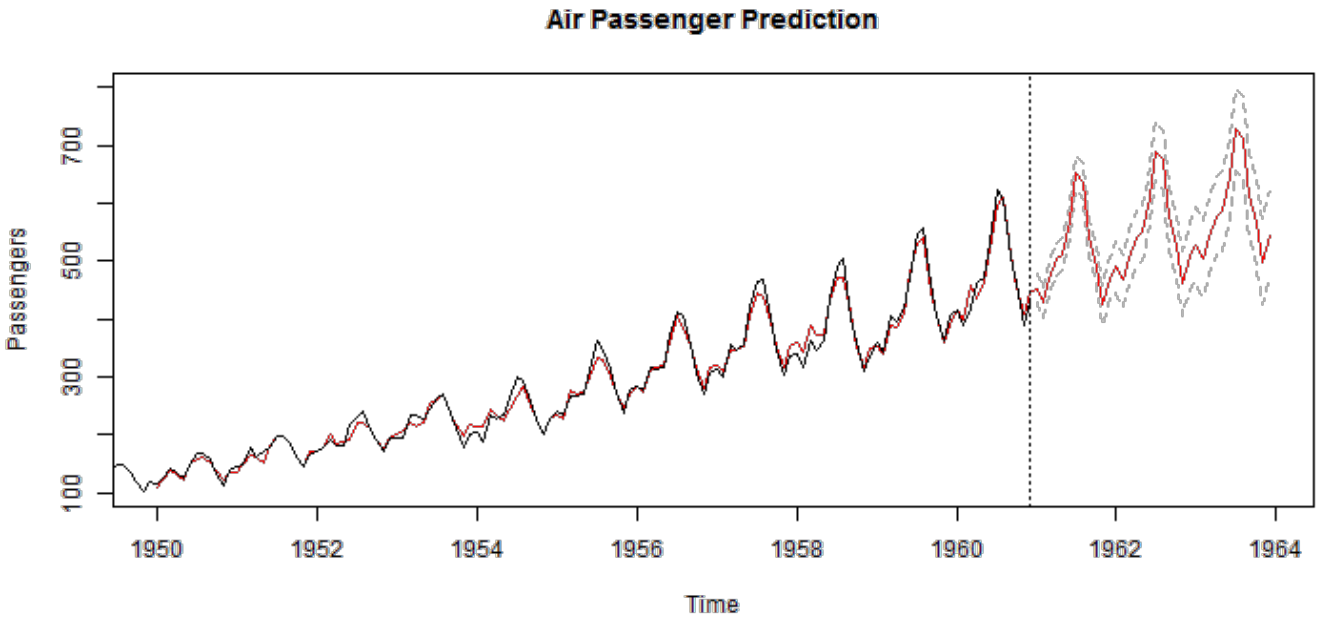
```
x <- HoltWinters(x = AirPsgr)
prediction <- predict(object = x, n.ahead = 36)
plot(x,prediction, main = "Air Passenger Prediction", ylab = "Passengers")
```



Now the package optimises all smoothing factors, creating a prediction including seasonal impacts. This provides a much more reliable prediction including the impact of seaonality.

Adding prediction interval bands

```
x <- HoltWinters(x = AirPsgr)
prediction <- predict(object = x, n.ahead = 36, prediction.interval = TRUE)
plot(x,prediction, main = "Air Passenger Prediction", ylab = "Passengers",
                   lty.intervals = "dashed",
                   col.intervals = "grey")
```



This overlays an error band around the prediction level.

Full R code

```
#loading dataset
data(AirPassengers)
x<- as.data.frame(AirPassengers)

#Convert to R 'time series' object and plot
AirPsgr <-ts(data = AirPassengers, start = c(1949,1),frequency = 12)

#Holt Winters Exp Smoothing - Optimised Alpha, Beta and Gamma - Includes Trend & Seasonality
x <- HoltWinters(x = AirPsgr)

#Running prediction for 36 months
prediction <- predict(object = x, n.ahead = 36)

#plot output including fitting diagnostics
plot(x,prediction, main = "Air Passenger Prediction", ylab = "Passengers")
plot(fitted(x))
```