Introduction to Forecasting – Holt-Winters for Trend

We have seen a few simple forecasting techniques at this point.

- The naïve method really just says that whatever happened yesterday will now happen today.
- When the past weights are all equal (and obviously only include a finite number of terms) we have the technique that some authors call *Simple Moving Averages*.
- A little more sophisticated, we can build a forecast by taking a weighted average of past values, with weights given corresponding to the geometric series (i.e. exponentially decreasing). This gives a forecasting method called *Simple Exponential Smoothing* or *SES*. This method gives greater weight to those values in the series closer to the forecast, and lesser weight to terms further in the past.
- Our notation shows that the forecast value for x_{n+h} made with data available at time *n* and looking *h* lags into the future is denoted x_{n+h}^{n}. The subscript always tells us where we want the forecast and the superscript always tells us what information we have available.

$$x_{n+h}^{n} = x_{\text{time for which you'd like a forecast}}^{\text{time for which you actually have data}}$$

$$\text{SES:} \quad x_{n+1}^{n} = \alpha x_n + (1-\alpha)\, x_n^{n-1}, \qquad \text{start with } \; x_2^1 = x_1$$

Many authors follow the convention that we put hats on estimated quantities, and as our formulas become more complicated, I think this handy reminder is a good idea. We'll also drop the superscripts

$$\text{SES:} \quad \hat{x}_{n+1} = \alpha x_n + (1-\alpha)\, \hat{x}_n$$

While SES this is not a bad approach, it is certainly limited and doesn't include some other factors which may be driving your system. We would like to build on this approach but also include *seasonal* and *trend* effects. The Holt-Winters method (also called, obviously enough, exponential smoothing with trend and seasonality, or even triple exponential smoothing) takes us in this direction.

In general, we will need to keep track of

- Levels, with parameter “alpha”, α
- Trend, with parameter “beta”, β
- Seasonal Component , with parameter “gamma”, γ

This method is widely known and has the virtue of being a method used by many companies for short term forecasts. As stated in the Paul Goodwin article from Foresight, available online at

https://forecasters.org/pdfs/foresight/free/Issue19_goodwin.pdf

…the method is popular because it is simple, has low data-storage requirements, and is easily automated. It also has the advantage of being able to adapt to changes in trends and seasonal patterns in sales when they occur. This means that slowdowns or speed-ups in demand, or changing consumer behavior at Christmas or in the summer, can all be accommodated. It achieves this by updating its estimates of these patterns as soon as each new sales figure arrives.

We have already seen that we can use the routine *HoltWinters()* for SES if we turn off the seasonal and trend effects.

HoltWinters(data, beta=FALSE, gamma=FALSE)

As we move forward, let’s work towards understanding and including trend in our forecasts by using one of the DataMarket datasets which describes the “volume of money” during the latter part of the 20th century (Feb 1960 – Dec 1994). The source of the data is the Australian Bureau of Statistics (G’Day mate!) Please visit the Time Series Data Library at the link below to explore other data sets, in addition to this one.

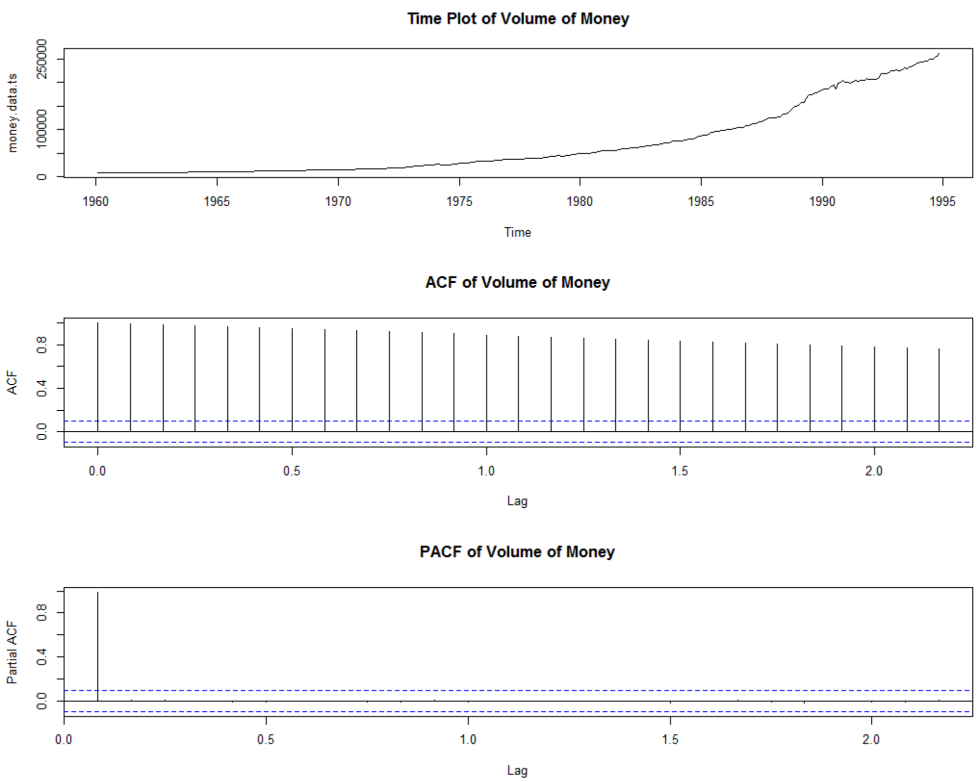
http://datamarket.com/data/list/?q=provider:tsdl

You can export the file in a convenient format, save it to your desktop, then edit a little to bring the data into R.

Once we edit the file we can run the following:

rm(list=ls(all=TRUE))

```
# use your appropriate directory!
#setwd("the directory where you have your data")
setwd("C:/Users/yourname/Desktop")                    #for example
money.data = read.table("volume-of-money-abs-definition-m.txt")
money.data.ts = ts(money.data[,2],start=c(1960,2) , frequency=12)
par(mfrow=c(3,1))
plot(money.data.ts, main="Time Plot of Volume of Money")
acf(money.data.ts, main="ACF of Volume of Money")
acf(money.data.ts, type="partial", main="PACF of Volume of Money")
```



We aren't surprised at the slow decay in the ACF, given the obvious trend. And, the PACF seems to cut off very sharply, indeed. Instead of developing a SARIMA model, at the moment we are interested in forecasting. How can we accommodate the obvious trend in these data?

To get some traction here, we need to estimate weights for level, trend, and seasonal components. Let's start the discussion with our non-seasonal data (i.e. we look for level and trend) before moving on to seasonal data. Note that we are not assuming that the trend will remain constant (that is, we are not assuming simple linear trend) but rather we allow the amount of trend to vary as we move along the series. In fact, we update the amount of trend in much the same way as we updated the level forecasts in SES, by using a weighted sum on our estimated trend components.

We've already seen for SES

$$\text{new level} = \alpha \cdot \text{new information} + (1-\alpha)\cdot \text{old level}$$

$$\hat{x}_{n+1} = \alpha\, x_n + (1-\alpha)\,\hat{x}_n$$

Think of this as getting a new level by updating the old level (forecast) and including new information. You can also think of it as yesterday's forecast plus a constant times the error in yesterday's forecast.

$$\hat{x}_{n+1} = \hat{x}_n + \alpha\,(x_n - \hat{x}_n)$$

Now a level is just a "smoothed value" of the time series. Some authors like to use *l* for level, or even *S* for smoothed and perhaps use *t* for time.

$$l_t = \alpha\, x_t + (1-\alpha)\, l_{t-1}$$

It's a little bit of overkill here, but we could think of SES as creating a forecast value at step $n+1$ as simply the smoothed value available at step n. We write

$$l_n = \alpha\, x_n + (1-\alpha)\,\hat{x}_n$$

$$\hat{x}_{n+1} = l_n$$

Double Exponential, or Exponential Smoothing with Trend

We extend SES by incorporating the "trend" T_n (i.e. the expected increase or decrease over the next lag) in order to update our forecast. The simplest way to write this is as

$$\text{forecast} = \text{level} + \text{trend}$$

We can unpack this a little bit as

$$\text{(double exponential) forecast } = \text{smoothed value} + \text{correction including trend information}$$

Be very careful comparing equations between authors since the coefficients will have a variety of names. We will follow the convention used in the *HoltWinters()* routine and use β for the trend component. The one-step-ahead equations for your forecast will look like

$$\hat{x}_{n+1} = \text{level}_n + \text{trend}_n$$

We produce the smoothed value or level as

$$\text{level}_n = \alpha \cdot \text{new information} + (1 - \alpha)\,(\text{old level} + \text{amount of trend})$$

$$\text{level}_n = \alpha x_n + (1 - \alpha)\,(\text{level}_{n-1} + \text{trend}_{n-1})$$

And we define the trend as a weighted average of the current change in level and previous trend, similarly to what we did for SES on time series values

$$\text{trend}_n = \beta \cdot \text{ new trend} + (1 - \beta) \cdot \text{old trend}$$

$$\text{trend}_n = \beta \cdot (\text{level}_n - \text{level}_{n-1}) + (1 - \beta)\text{trend}_{n-1}$$

Your software will find optimal values of α and β .

As long as we can jumpstart with initial levels and trends we can proceed inductively. There are multiple ways forward, but obvious choices for starting values would be

$$\text{level}_1 = x_1$$

$$\text{trend}_1 = x_2 - x_1$$

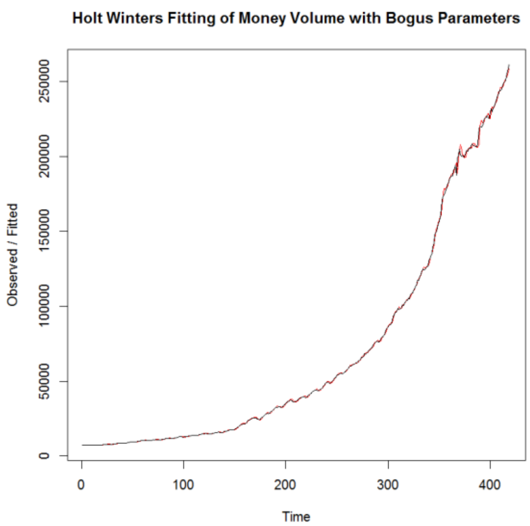
And again, for those of us who learn concepts by writing code, here is another quick, easy implementation in R. We will again compare our results to the *HoltWinters()* routine's output.

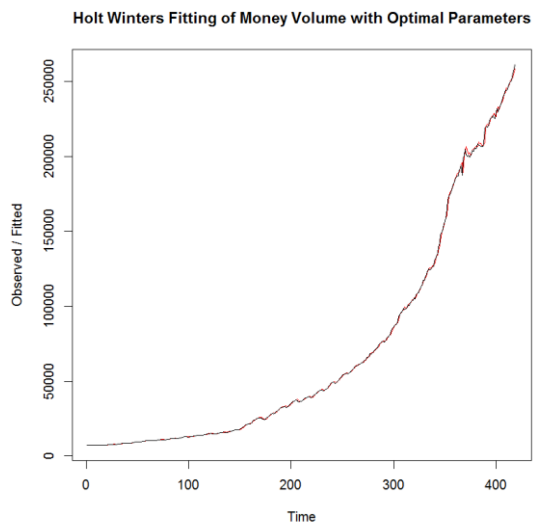
We can write code ourselves to find the best multipliers α and β , or we can use software. To make life easy as we get started, explicitly use non-optimal values taken out of a hat as α=0.7 and $\beta = 0.5$.

```
#set up our transformed data and smoothing parameters
data                    = money.data[,2]
N                       = length(data)
alpha                   = 0.7
beta                    = 0.5

##prepare empty arrays so we can store values
forecast                = NULL
level                   = NULL
trend                   = NULL

#initialize level and trend in a very simple way
level[1]                = data [1]
trend[1]                = data [2]- data [1]

#initialize forecast to get started
forecast[1]             = data [1]
forecast[2]             = data [2]

#loop to build forecasts
for( n in 2:N ) {
        level[n] = alpha* data [n] +
                              (1-alpha)*(level[n-1]+trend[n-1])

        trend[n] = beta*(level[n] - level[n-1]) +
                              (1-beta)*trend[n-1]

        forecast[n+1] =  level[n] + trend[n]
}

#display your calculated forecast values
forecast[3:N]
        7287.000,   7273.050,   7181.698,     ...,      258579.059

#verify that we have recovered HoltWinters() output
m       =       HoltWinters(data, alpha = 0.7, beta = 0.5, gamma = FALSE)

m$fitted[,1]
        7287.000   7273.050   7181.698,      ...,      258579.059
```

Note that we can plot our model with a simple call

plot(m, main="Holt Winters Fitting of Money Volume with Bogus Parameters")





It won't be a good fit with the bogus numbers we supplied. Instead, let the *HoltWinters()* routine "do its thing" and find optimal α and β (shown in the second plot).

m=HoltWinters(data, gamma = FALSE)

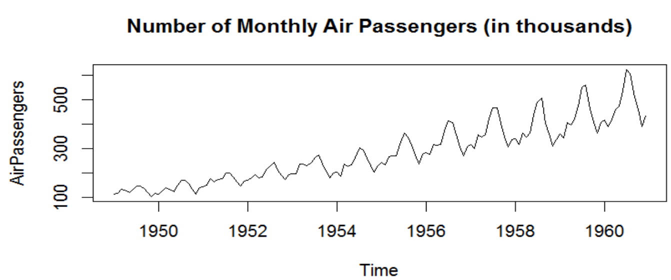
plot(m, main="Holt Winters Fitting of Money Volume with Optimal Parameters")

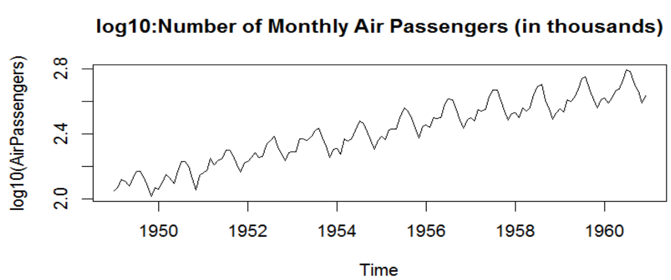
Airline Data

Here is one of the classic datasets available in R. This one concerns monthly totals of international airline passengers during the years 1949 to 1960 and is called, appropriately enough, *AirPassengers*. These data describe the total number of monthly air passengers in thousands of travelers.

> *Monthly Airline Passenger Numbers 1949-1960*
>
> *Description*: The classic Box & Jenkins airline data. Monthly totals of international airline passengers, 1949 to 1960.
>
> *Usage*: AirPassengers
>
> *Format*: A monthly time series, in thousands.
>
> *Source*: Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1976) *Time Series Analysis, Forecasting and Control*. Third Edition. Holden-Day. Series G.

A plot of the time series reveals strong seasonality as well as trend. The data set also seems to be heteroscedastic, with increasing variability in later years (see first plot below). If we transform our data set and instead consider $\log_{10}$(number of passengers) we obtain a calmer looking plot.





It is a little foolish to do this, but if we apply SES to the transformed data we obtain:

HoltWinters(x = log10(AirPassengers), beta = FALSE, gamma = FALSE)

Smoothing parameters:
alpha: 0.9999339
beta : FALSE
gamma: FALSE

Coefficients: a 2.635481

For all practical purposes, the α value is 1 and our forecast is essentially naïve forecasting.

$$\hat{x}_{n+1} = \alpha x_n$$

We are obviously "leaving money on the table" and not proceeding optimally. As a quick quality measure,

m$SSE # 0.3065102

Let's see if we can reduce the error. In the next lecture, we work to develop a forecast that can accommodate both the trend in our data, as well as the seasonal component.