

Forecasting

TRSM R Team

04/02/2022

Forecasting is a skill that anyone interested in analytics should learn about. With forecasting:

- Temperature
- Stock prices
- Incoming Revenue

Can be predicted.

Forecasting in R

We will be using the fpp2 library to demonstrate how to forecast in R programming language. The mathematical solutions are been excluded from this session but all the content in this file can be found here.

Properties of Forecasting

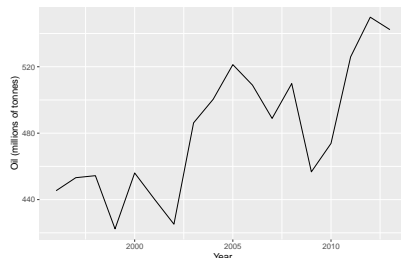
They are three main characteristics that all forecasting models try to stimulate within any given data: 1. **Seasonality**: The increasing or decreasing value in the series. 2. **Trend**: The repeating short-term cycle in the series. And given that a time series data has a Seasonality or trend (or both, or none), we can choose to use different models.

Exponentially Smoothing methods

The simplest of the exponentially smoothing methods is naturally called simple exponential smoothing (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern. For example, the data in Figure below does not display any clear trending behavior or any seasonality. (There is a rise in the last few years, which might suggest a trend. We will consider whether a trended method would be better for this series later in this chapter.)

```
# Simple Exponential Smoothing
oildata <- window(oil, start=1996)

autoplot(oildata) +
  ylab("Oil (millions of tonnes)") + xlab("Year")
```



Note: oil is a dataframe that is included with fpp2, so when fpp2 is imported, this dataset is included as well. Also this dataset is the production of oil in Saudi Arabia.

We can run a simple exponential model on this data and get its accuracy:

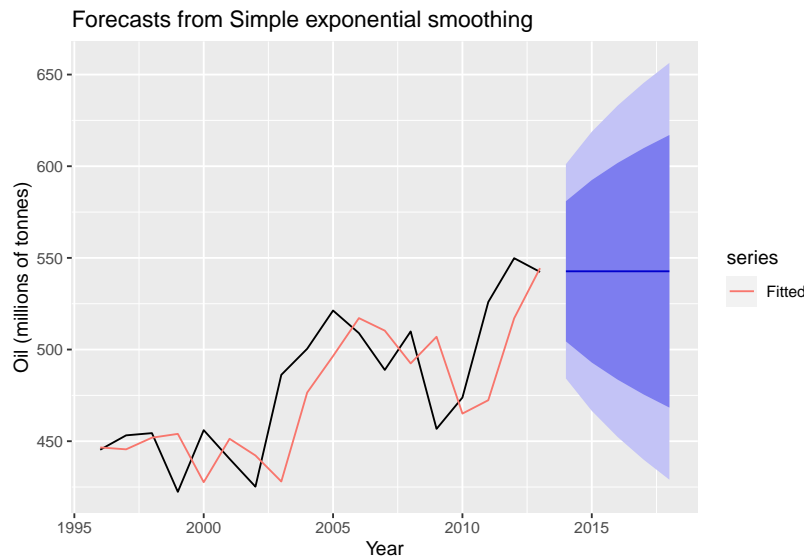
```
# Estimate parameters
fc <- ses(oildata, h=5)

# Accuracy of one-step-ahead training errors
round(accuracy(fc),2)

##              ME  RMSE  MAE  MPE  MAPE  MASE  ACF1
## Training set 6.4 28.12 22.26 1.1  4.61  0.93 -0.03
```

After fitting the model we can plot the actual data alongside our forecasts and see how close the plots are:

```
autoplot(fc) +
  autolayer(fitted(fc), series="Fitted") +
  ylab("Oil (millions of tonnes)") + xlab("Year")
```



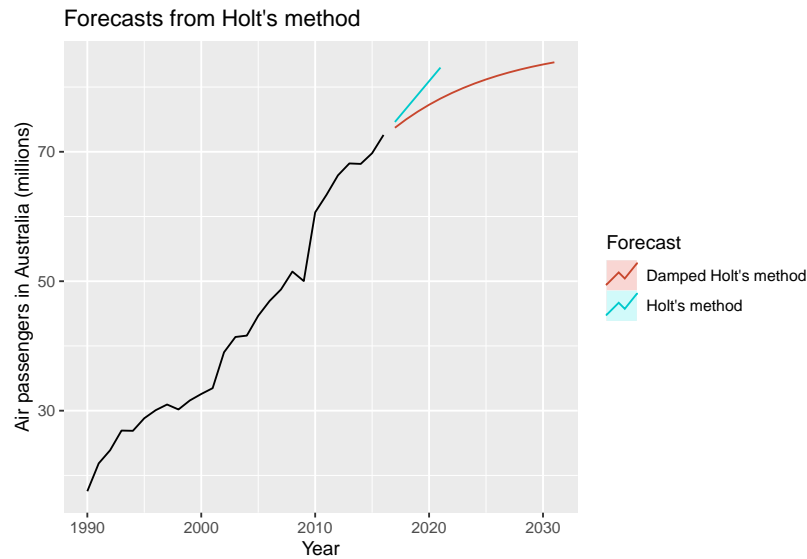
Holt's Models

Holt's models extend simple exponential smoothing to allow the forecasting of data with a trend. Two types:

1. Holt's Linear trend method: Accounts for trends. 2. Damped Holt's trend method: Same as the simple one but introduces a parameter that "dampens" the trend to a flat line some time in the future.

```
air <- window(ausair, start=1990)
fc <- holt(air, h=5)

fc2 <- holt(air, damped=TRUE, phi = 0.9, h=15)
autoplot(air) +
  autolayer(fc, series="Holt's method", PI=FALSE) +
  autolayer(fc2, series="Damped Holt's method", PI=FALSE) +
  ggtitle("Forecasts from Holt's method") + xlab("Year") +
  ylab("Air passengers in Australia (millions)") +
  guides(colour=guide_legend(title="Forecast"))
```

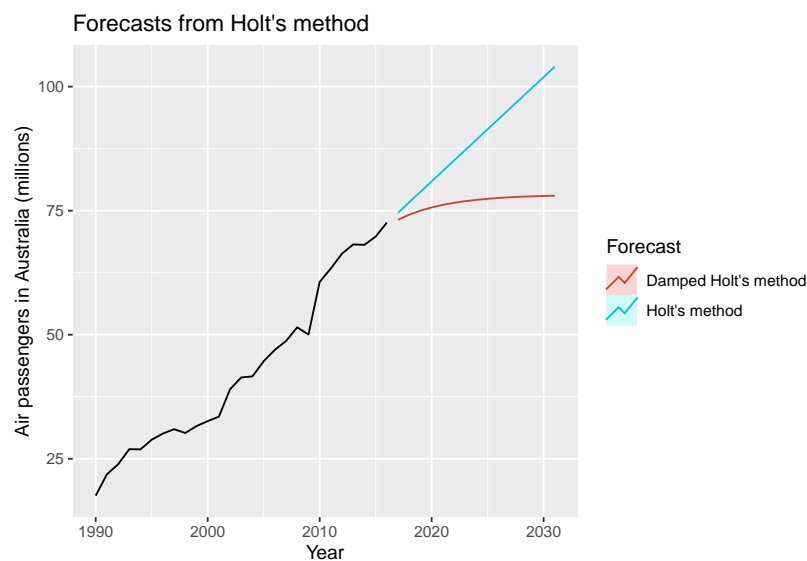


We have set the damping parameter to a relatively low number ($\phi=0.90$) to exaggerate the effect of damping for comparison. Usually, we would estimate ϕ along with the other parameters. We have also used a rather large forecast horizon ($h=15$) to highlight the difference between a damped trend and a linear trend. In practice, we would not normally want to forecast so many years ahead with only 27 years of data.

Exercise: Change the value for ϕ , what happens when this value increases or decreases?

```
air <- window(ausair, start=1990)
fc <- holt(air, h=15)

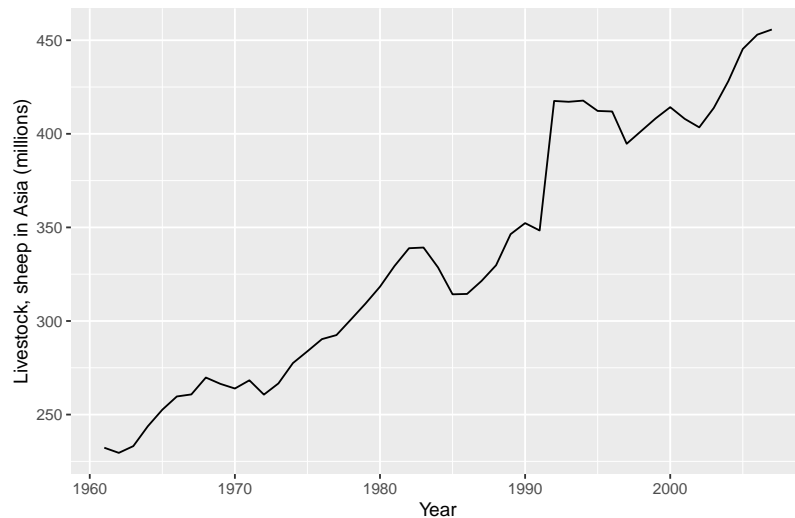
fc2 <- holt(air, damped=TRUE, phi = 0.8, h=15)
autoplot(air) +
  autolayer(fc, series="Holt's method", PI=FALSE) +
  autolayer(fc2, series="Damped Holt's method", PI=FALSE) +
  ggtitle("Forecasts from Holt's method") + xlab("Year") +
  ylab("Air passengers in Australia (millions)") +
  guides(colour=guide_legend(title="Forecast"))
```



Example: Sheep in Asia

In this example, we compare the forecasting performance of the three exponential smoothing methods that we have considered so far in forecasting the sheep livestock population in Asia. The data spans the period 1961–2007 and is shown in Figure below.

```
autoplot(livestock) +  
  xlab("Year") + ylab("Livestock, sheep in Asia (millions)")
```



Which model should we use? How do we choose which model is the best one for our estimations?

1. Run a few different models
2. Estimate their respective errors
3. Compare errors and select the best model

Note: **tsCV** (time series cross validation) is a function that gives us the best version of a model given the model type and data.

```
e1 <- tsCV(livestock, ses, h=1)  
e2 <- tsCV(livestock, holt, h=1)  
e3 <- tsCV(livestock, holt, damped=TRUE, h=1)  
# Compare MAE:  
mean(abs(e1), na.rm=TRUE)
```

```
## [1] 8.53246
```

```
mean(abs(e2), na.rm=TRUE)
```

```
## [1] 8.803058
```

```
mean(abs(e3), na.rm=TRUE)
```

```
## [1] 8.024192
```

Damped Holt's method is best whether you compare MAE values. So we will proceed with using the damped Holt's method and apply it to the whole data set to get forecasts for future years.

```
fc <- holt(livestock, damped=TRUE)  
# Estimated parameters:  
fc[["model"]]
```

```
## Damped Holt's method
```

```
##
## Call:
## holt(y = livestock, damped = TRUE)
##
## Smoothing parameters:
##   alpha = 0.9999
##   beta  = 3e-04
##   phi   = 0.9798
##
## Initial states:
##   l = 223.35
##   b = 6.9046
##
## sigma: 12.8435
##
##      AIC      AICc      BIC
## 427.6370 429.7370 438.7379
```

Holt-Winters' seasonal method

Holt-Winters' method is extended version of the Holt's method to capture seasonality. The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level, one for the trend, and one for the seasonal component, with corresponding smoothing parameters. We use m to denote the frequency of the seasonality, i.e., the number of seasons in a year. For example, for quarterly data $m=4$, and for monthly data $m=12$.

There are two variations to this method that differ in the nature of the seasonal component: 1. Additive Method: The additive method is preferred when the seasonal variations are roughly constant through the series. 2. Multiplicative Method: The multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series.

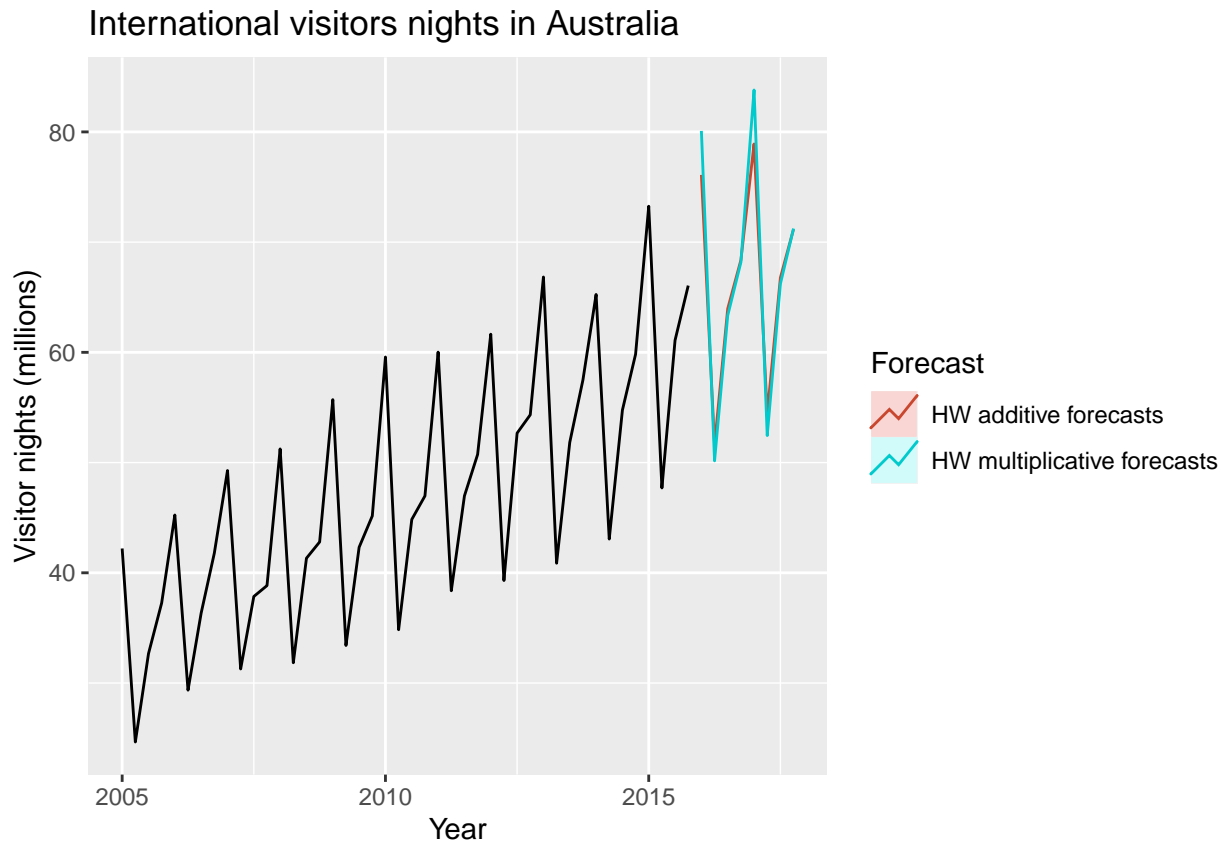
Example: International tourist visitor nights in Australia

We apply Holt-Winters' method with both additive and multiplicative seasonality to forecast quarterly visitor nights in Australia spent by international tourists. Figure below shows the data from 2005, and the forecasts for 2016–2017. The data shows an obvious seasonal pattern, with peaks observed in the March quarter of each year, corresponding to the Australian summer.

```
# Bring in the data in window format
aust <- window(austourists, start=2005)

# Running the models
fit1 <- hw(aust, seasonal="additive")
fit2 <- hw(aust, seasonal="multiplicative")

# Plotting the models side-by-side
autoplot(aust) +
  autolayer(fit1, series="HW additive forecasts", PI=FALSE) +
  autolayer(fit2, series="HW multiplicative forecasts",
    PI=FALSE) +
  xlab("Year") +
  ylab("Visitor nights (millions)") +
  ggtitle("International visitors nights in Australia") +
  guides(colour=guide_legend(title="Forecast"))
```



Exercise: In this case: Seasonal or Multiplicative?

```
print('Error for add:')
```

```
## [1] "Error for add:"
```

```
accuracy(fit1,ausair)
```

```
##
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set  0.008115785 1.763305 1.374062 -0.2860248 2.973922 0.1193523
## Test set     -3.500672471 3.500672 3.500672 -4.8220156 4.822016 0.3040718
##
##           ACF1
## Training set -0.06272507
## Test set     NA
```

```
print('Error for mult:')
```

```
## [1] "Error for mult:"
```

```
accuracy(fit2,ausair)
```

```
##
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set  0.09206228 1.575631 1.254960 -6.505533e-04 2.70539 0.1090070
## Test set     -7.49123452 7.491235 7.491235 -1.031883e+01 10.31883 0.6506958
##
##           ACF1
## Training set -0.07955726
## Test set     NA
```

Example: Holt-Winters method with daily data

Note: We can dampen the Holt-Winters model as well by setting `damped=TRUE`.

The Holt-Winters method can also be used for daily type of data, where the seasonal period is $m=7$, and the appropriate unit of time for h is in days. Here, we generate daily forecasts for the last five weeks for the `hyndsight` data, which contains the daily pageviews on the Hyndsight blog for one year starting April 30, 2014.

```
fc <- hw(subset(hyndsight,end=length(hyndsight)-35),
         damped = TRUE, seasonal="multiplicative", h=35)
autoplot(hyndsight) +
  autolayer(fc, series="HW multi damped", PI=FALSE)+
  guides(colour=guide_legend(title="Daily forecasts"))
```

