Advanced Exponential Forecasting

Series with No Trend and Seasonality

- Moving Average and Simple Exponential Smoothing should be used for forecasting the series with no trend and seasonality

Series with Additive Trend

- For series with trend, we can use Holt's method, also known as double exponential smoothing
- Similar to Simple Exponential Smoothing, the level of the series is estimated from the data and is updated as more data would become available
- Level is estimated using maximum likelihood method

Holt's Linear Trend Method

- The k-step ahead forecast is given by combining the level estimate at time t (L_t) and trend estimate at time t (T_t):

$$F_{t+k} = L_t + kT_t$$

- The level and trend are updated by the equations:

$$L_t = \alpha y_t + (1-\alpha)(L_{t-1} + T_{t-1})$$
$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$

  – Where α and β are smoothing constants whose values range from 0 to 1 and are set by the user or chosen iteratively by R function holt()

- Level equation shows L_t, Level at time t as weighted average of the observation at time t y_t and within sample one step ahead forecast at time t , $(L_{t-1}+T_{t-1})$
- Trend Equation shows T_t , trend estimate at time t as weighted average of $(L_t - L_{t-1})$ and T_{t-1}, the previous trend estimate

Holt’s Method in R

- Holt’s Method can be implemented using function holt() in R

Syntax : holt(ts, h, initial, damped=FALSE, exponential=FALSE, alpha=NULL, beta=NULL,…)

where

ts : a numeric vector or time series object

h : Number of periods for forecasting

initial : If “optimal”, (default) initial values are optimized with smoothing parameters using ets() which use likelihood method. If “simple”, initial values are set to values on first few observations

damped : default=FALSE, if it is set to TRUE then damped trend method is used

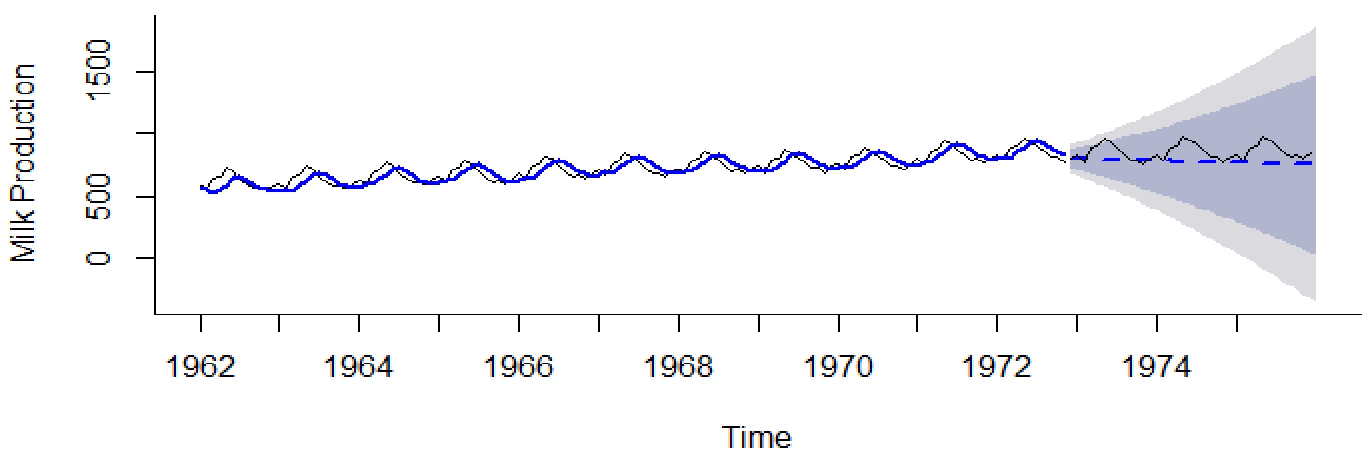
exponential : If TRUE then exponential trend is fitted, otherwise (default=FALSE) linear trend is fitted

alpha : smoothing constant for level, if NULL then it is estimated

beta : smoothing constant for trend , if NULL then it is estimated

Example: setting alpha and beta

```
# Setting Aplha and Beta
holtMilk <- holt(train.ts,alpha = 0.5,beta = 0.05,initial = "simple",h = nValid)
plot(holtMilk,   ylab = "Milk Production", xlab = "Time", bty = "l",
     xaxt = "n",  main = "", flty = 2)
axis(1, at = seq(1962, 1975, 1), labels = format(seq(1962, 1975, 1)))
lines(holtMilk$fitted, lwd = 2, col = "blue")
lines(valid.ts)
```

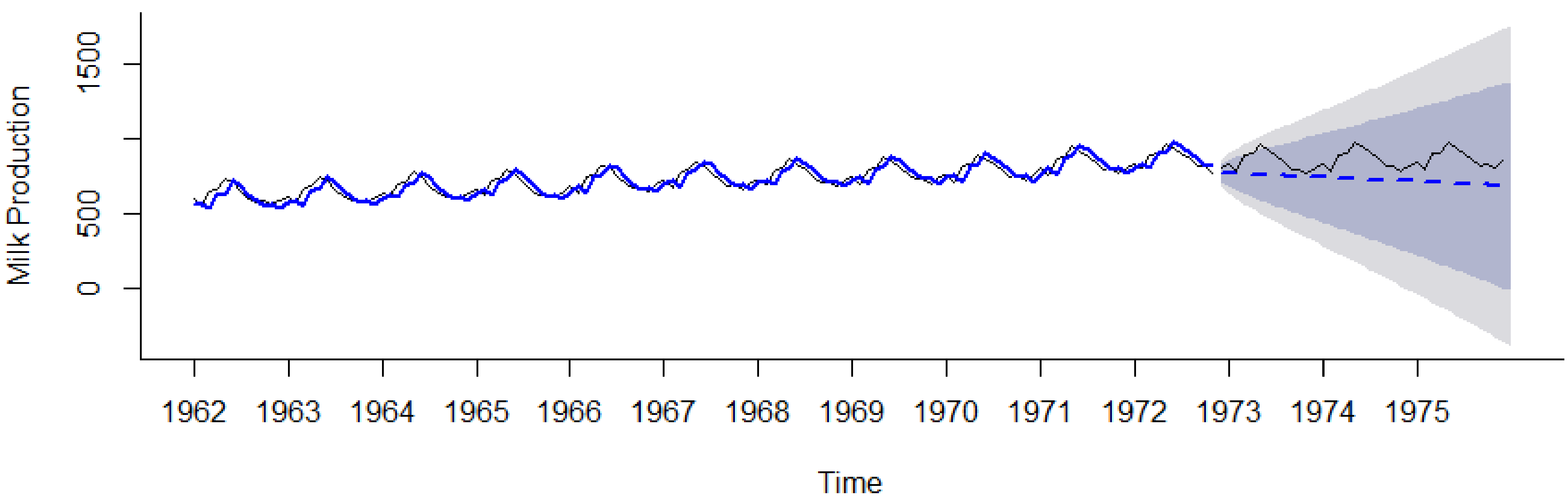


Example: setting alpha and beta

```
> holtMilk$model
Holt's method

Call:
 holt(x = train.ts, h = nValid, initial = "simple", alpha = 0.5,

 Call:
     beta = 0.05)

  Smoothing parameters:
    alpha = 0.5
    beta  = 0.05

  Initial states:
    l = 589
    b = -28

  sigma:  57.7039
> accuracy(holtMilk , valid.ts)
                   ME      RMSE      MAE       MPE     MAPE     MASE      ACF1 Theil's U
Training set  8.15813  57.70387 49.92364 0.9300842 6.838588 2.071448 0.5999152        NA
Test set     79.04433 100.58977 81.63049 8.8166707 9.153775 3.387039 0.6787697  1.918983
```

Example: Without setting alpha and beta

```
#Without setting Alpha and Beta
holtMilkLin <- holt(train.ts,initial = "simple",h = nValid)
plot(holtMilkLin,   ylab = "Milk Production", xlab = "Time",
     bty = "l", xaxt = "n",  main = "", flty = 2)
axis(1, at = seq(1962, 1975, 1), labels = format(seq(1962, 1975, 1)))
lines(holtMilkLin$fitted, lwd = 2, col = "blue")
lines(valid.ts)
```



Example: Without setting alpha and beta

```
> holtMilkLin$model
Holt's method

Call:
 holt(x = train.ts, h = nValid, initial = "simple")

  Smoothing parameters:
    alpha = 1
    beta  = 0.0443

  Initial states:
    l = 589
    b = -28

  sigma:  45.8938
> accuracy(holtMilkLin , valid.ts)
                     ME      RMSE       MAE        MPE      MAPE     MASE       ACF1 Theil's U
Training set   4.437321  45.89378  39.12207  0.5560376  5.350448 1.623266 0.08574719        NA
Test set     130.394211 146.13942 130.39421 14.8578433 14.857843 5.410360 0.70254950  2.821901
```

Example: setting the initial=optimal

```
> holtMilkLin$model
Holt's method

Call:
 holt(x = train.ts, h = nValid, initial = "optimal")

  Smoothing parameters:
    alpha = 0.9999
    beta  = 1e-04

  Initial states:
    l = 637.5537
    b = 1.3343

  sigma:  44.3091

     AIC     AICc      BIC
1639.943 1640.260 1651.443
> accuracy(holtMilkLin , valid.ts)
                     ME     RMSE      MAE        MPE     MAPE     MASE       ACF1 Theil's U
Training set -0.3539962 44.30911 38.43737 -0.2325406 5.284276 1.594856 0.06750352        NA
Test set     61.9474847 86.62871 66.89926  6.8199315 7.457638 2.775806 0.67455013  1.643056
```

Exponential Trend Method

- The k-step ahead forecast is given by combining the level estimate at time t (L_t) and trend estimate at time t (T_t):

$$F_{t+k} = L_t \times {T_t}^k$$

- The level and trend are updated by the equations:

$$L_t = \propto y_t + (1 - \propto)(L_{t-1} \times T_{t-1})$$

$$T_t = \beta\left(\frac{L_t}{L_{t-1}}\right) + (1-\beta)T_{t-1}$$

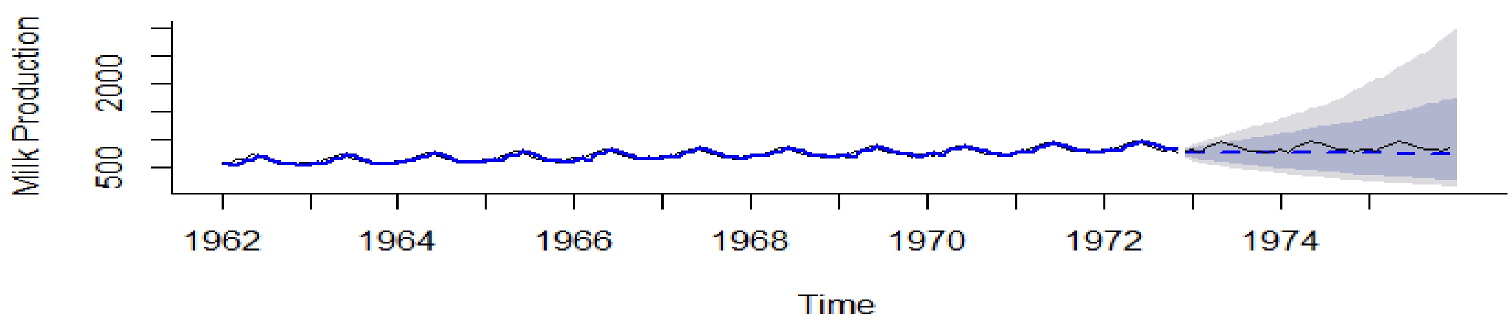
  – Where α and β are smoothing constants whose values range from 0 to 1 and are set by the user or chosen iteratively by R function holt()

- Level equation shows L_t, Level at time t as weighted average of the observation at time t y_t and within sample one step ahead forecast at time t , $(L_{t-1} \times T_{t-1})$
- Trend Equation shows T_t , trend estimate at time t as weighted average of (L_t/L_{t-1}) and T_{t-1}, the previous trend estimate

Exponential Trend in R

- For implementing exponential trend method in holt() we set the argument exponential=TRUE

```r
holtMilkExp <- holt(train.ts,initial = "simple",exponential = TRUE ,h = nValid)
plot(holtMilkExp,   ylab = "Milk Production", xlab = "Time", bty = "l",
     xaxt = "n",  main = "", flty = 2)
axis(1, at = seq(1962, 1975, 1), labels = format(seq(1962, 1975, 1)))
lines(holtMilkExp$fitted, lwd = 2, col = "blue")
lines(valid.ts)
```



Model and Accuracy

```
> holtMilkExp$model
Holt's method with exponential trend

Call:
 holt(x = train.ts, h = nValid, initial = "simple", exponential = TRUE)

  Smoothing parameters:
    alpha = 1
    beta  = 0.039

  Initial states:
    l = 589
    b = 0.9525

  sigma:  0.0664
> accuracy(holtMilkExp , valid.ts)
                 ME      RMSE      MAE        MPE      MAPE     MASE       ACF1 Theil's U
Training set  4.225  45.88032 38.98685  0.5368849  5.328352 1.617655 0.09351393        NA
Test set     96.527 113.84889 96.53213 10.8807155 10.881390 4.005343 0.66663233  2.177755
```

Damped Trend Methods

- It has been observed that Holt's Linear Trend and Exponential Trend tend to over-forecast for longer forecast horizons
- Gardner and McKenzie (1985) suggested a parameter that dampens the trend line to a flat line some time in the future
- Methods with damped trend have been proven to be more successful when forecasts are to be predicted by automatic process
- There are two types of damped trend methods:
  - Additive Damped Trend
  - Multiple Damped Trend

Additive Damped Trend

- In association with the smoothing parameters α and β, damped methods also include a damping parameter φ; 0 <φ< 1 as:

$$F_{t+k} = L_t + (\varphi + \varphi^2 + \cdots + \varphi^k)T_t$$
$$L_t = \propto y_t + (1 - \propto)(L_{t-1} - \varphi T_{t-1})$$
$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)\varphi T_{t-1}$$

- If φ=1 then the method is Holt's Linear Method

Multiplicative Damped Trend

- Taylor(2003) introduced a damping parameter to the exponential trend

$$F_{t+k} = L_t \times {T_t}^{(\varphi+\varphi^2+\cdots+\varphi^k)}$$

$$L_t = \alpha y_t + (1-\alpha)L_{t-1} \times {T_{t-1}}^{\varphi}$$

$$T_t = \beta\left(\frac{L_t}{L_{t-1}}\right) + (1-\beta){T_{t-1}}^{\varphi}$$

Example: Additive Damped Trend

```
> holtMilkDamp$model
Damped Holt's method

Call:
 holt(x = train.ts, h = nValid, damped = TRUE, initial = "optimal")

  Smoothing parameters:
    alpha = 0.9999
    beta  = 1e-04
    phi   = 0.98

  Initial states:
    l = 636.8221
    b = 1.8639

  sigma:  44.3054

     AIC     AICc      BIC
1641.921 1642.401 1656.297
> accuracy(holtMilkDamp , valid.ts)
                     ME      RMSE      MAE        MPE     MAPE     MASE       ACF1 Theil's U
Training set  0.3351106  44.30542 38.40045 -0.1422836 5.277879 1.593324 0.06737316        NA
Test set     85.2917080 104.13195 85.69006  9.5612885 9.613703 3.555480 0.66346402  1.985189
```

Example: Multiplicative Damped Trend

```
> holtMilkDamp$model
Damped Holt's method with exponential trend

Call:
 holt(x = train.ts, h = nValid, damped = TRUE, exponential = TRUE)

  Smoothing parameters:
    alpha = 0.9999
    beta  = 1e-04
    phi   = 0.9067

  Initial states:
    l = 636.7437
    b = 0.9917

  sigma:  0.0629

     AIC     AICc      BIC
1647.589 1648.069 1661.965
> accuracy(holtMilkDamp , valid.ts)
                    ME      RMSE      MAE        MPE     MAPE     MASE       ACF1 Theil's U
Training set  1.369702  44.30205 38.32190 0.01641978 5.261043 1.590065 0.06791672        NA
Test set     87.260090 105.78596 87.58223 9.79247286 9.834860 3.633991 0.66359471   2.01787
```

Holt-Winters Seasonal Method

- This method comprises of the forecast equation and three smoothing equations each for level, trend and seasonal component
- We use m to denote the period of season
- The additive method of Holt-Winters can be preferred when the seasonal variations are roughly constant through the series
- The multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series.

Holt-Winters Additive Method

- The component form of the model:

$$F_{t+k} = L_t + kT_t + S_{t-m+k_m{}^+}$$

$$L_t = \propto (y_t - S_{t-m}) + (1 - \propto)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

$$S_t = \gamma(y_t - L_t) + (1 - \gamma)S_{t-m}$$

Where

S_t : Seasonal Estimate at time t

${k_m}^+$: [(k-1) mod m)]+1 which ensures that the estimates of the seasonal indices used for forecasting come from the final year

Holt-Winters Multiplicative Method

- The component form of the model: (Additive Trend)

$$F_{t+k} = (L_t + kT_t)S_{t-m+k_m^+}$$

$$L_t = \propto \left(\frac{y_t}{S_{t-m}}\right) + (1 - \propto)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

$$S_t = \gamma\left(\frac{y_t}{L_t}\right) + (1 - \gamma)S_{t-m}$$

Where

S_t : Seasonal Estimate at time t

${k_m}^+$: [(k-1) mod m)]+1 which ensures that the estimates of the seasonal indices used for forecasting come from the final year

Holt-Winters in R

- Holt-Winters method can be implemented in R with function hw()

Syntax : hw(ts, h, initial, seasonal, exponential, alpha, beta, gamma, …)

where

ts : a numeric vector or time series object

h : Number of periods for forecasting

initial : If “optimal”, initial values are optimized with smoothing parameters using ets(). If “simple”, initial values are set to values on first few observations

seasonal : Type of seasonality in hw model. "additive" or "multiplicative"

exponential : If TRUE then exponential trend is fitted, otherwise (default=FALSE) linear trend is fitted

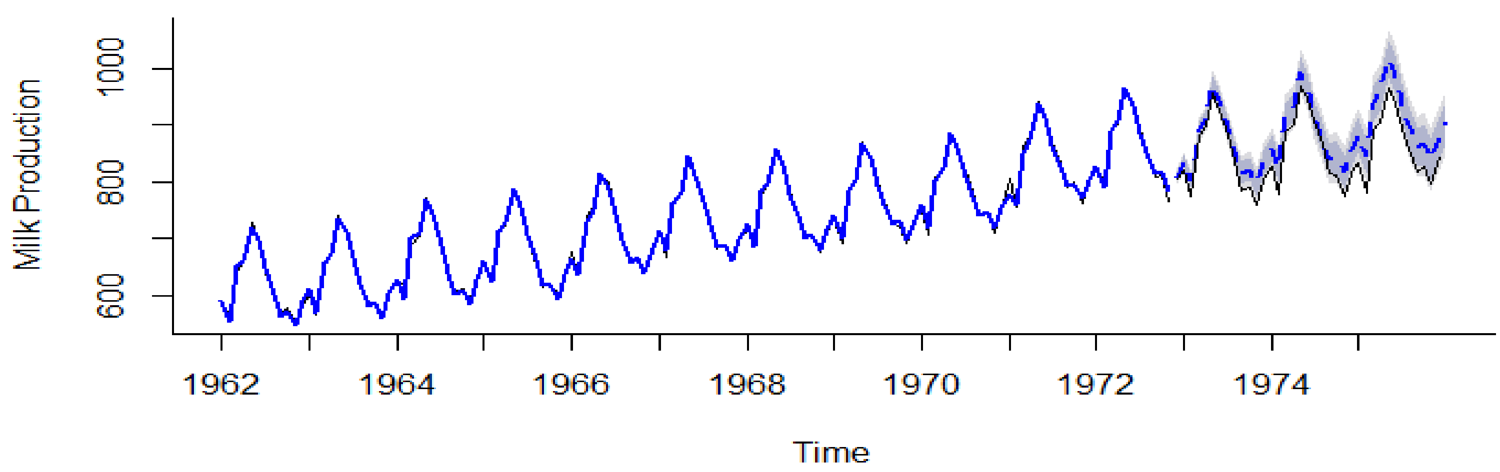
alpha : smoothing constant for level, if NULL then it is estimated

beta : smoothing constant for trend , if NULL then it is estimated

gamma : smoothing constant for seasonal component, if NULL then it is estimated

Example: Holt-Winters Additive

```
HWMilkAdd <- hw(train.ts,h = nValid ,seasonal = "additive")
plot(HWMilkAdd,   ylab = "Milk Production", xlab = "Time", bty = "l",
     xaxt = "n",  main = "", flty = 2)
axis(1, at = seq(1962, 1975, 1), labels = format(seq(1962, 1975, 1)))
lines(HWMilkAdd$fitted, lwd = 2, col = "blue")
lines(valid.ts)
```

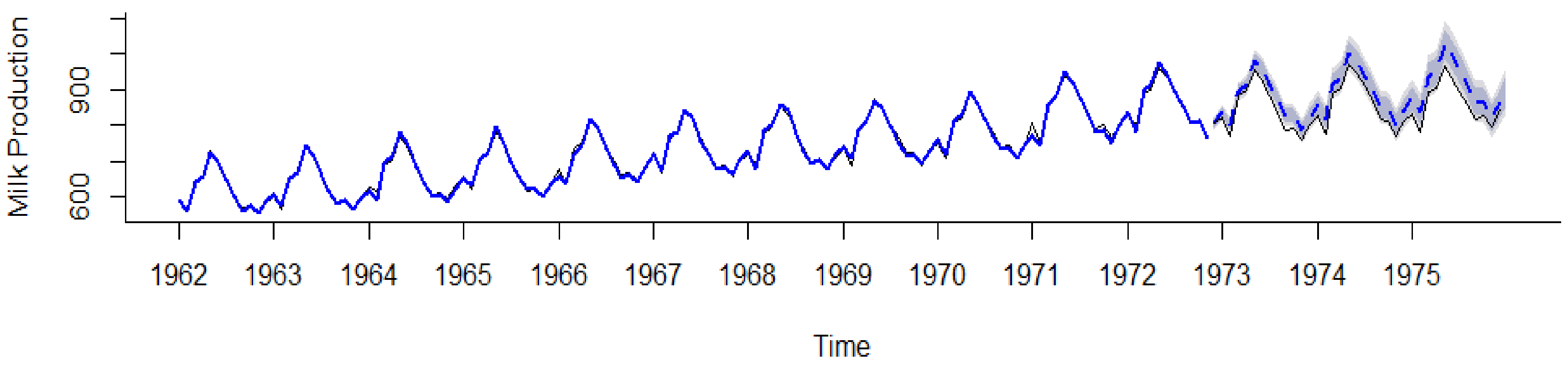


Model and Accuracy

```
> HWMilkAdd$model
Holt-Winters' additive method

Call:
 hw(x = train.ts, h = nValid, seasonal = "additive")

  Smoothing parameters:
    alpha = 0.6799
    beta  = 1e-04
    gamma = 1e-04

  Initial states:
    l = 605.2517
    b = 1.8674
    s=-42.4299 -78.1707 -49.0858 -52.9431 -12.6427 30.1153
           81.8793 110.4519 50.4173 34.5289 -54.7018 -17.4186

  sigma:  6.7803

     AIC     AICc      BIC
1172.125 1176.897 1218.128
> accuracy(HWMilkAdd , valid.ts)
                       ME      RMSE       MAE          MPE      MAPE      MASE        ACF1 Theil's U
Training set  0.004482957  6.780315  5.113118 -0.004536466 0.7172879 0.2121552 -0.00143546        NA
Test set    -31.344975577 34.516856 31.344976 -3.714358918 3.7143589 1.3005760  0.75248253 0.6954029
```

Example: Holt-Winters Multiplicative Method

```
HWMilkMult <- hw(train.ts,h = nValid ,seasonal = "multiplicative")
plot(HWMilkMult,   ylab = "Milk Production", xlab = "Time", bty = "l",
     xaxt = "n",  main = "", flty = 2)
axis(1, at = seq(1962, 1975, 1), labels = format(seq(1962, 1975, 1)))
lines(HWMilkMult$fitted, lwd = 2, col = "blue")
lines(valid.ts)
```



Model and Accuracy

```
> HWMilkMult$model
Holt-Winters' multiplicative method

Call:
 hw(x = train.ts, h = nValid, seasonal = "multiplicative")

  Smoothing parameters:
    alpha = 0.4545
    beta  = 1e-04
    gamma = 0.4984

  Initial states:
    l = 607.4071
    b = 1.6494
    s=0.9349 0.8864 0.9193 0.9055 0.9746 1.0478
           1.1337 1.1832 1.0747 1.0507 0.9208 0.9683

  sigma:  0.0114

     AIC     AICc      BIC
1222.410 1227.182 1268.413
> accuracy(HWMilkMult , valid.ts)
                     ME      RMSE       MAE         MPE      MAPE      MASE      ACF1 Theil's U
Training set   0.292404  8.285546  6.318835  0.05940374 0.8635815 0.2621832 0.2478558        NA
Test set     -34.286119 37.181644 34.299906 -4.01202864 4.0137413 1.4231830 0.7322070 0.7299709
```

Taxonomy of Exponential Methods

	Seasonal Component		
Trend Component	N (None)	A (Additive)	M (Multiplicative)
N (None)	**(N,N)**	(N,A)	(N,M)
A (Additive)	**(A,N)**	**(A,A)**	**(A,M)**
A_d (Additive Damped)	(A_d,N)	(A_d,A)	**(A_d,M)**
M (Multiplicative)	**(M,N)**	(M,A)	(M,M)
M_d (Multiplicative Damped)	**(M_d,N)**	(M_d,A)	(M_d,M)

- (N,N) : Simple exponential smoothing
- (A,N) : Holt's linear method
- (M,N) : Exponential trend method
- (A_d,N) : Additive damped trend method
- (M_d,N) : Multiplicative damped trend method
- (A,A) : Additive Holt-Winters method
- (A,M) : Multiplicative Holt-Winters method
- (A_d,M) : Holt-Winters damped method