**3. Holt-Winters Smoothing Models**

**Exponential Smoothing**

* Exponential smoothing is a very popular scheme to produce a smoothed time series and assigns exponentially decreasing weights as the observation get older. That is, more recent data points affect the forecast trend more heavily than older data points. This unequal weighting is accomplished by using one or more smoothing constants.



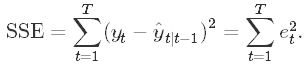
* It provides a forecasting method that is most effective when the components like trend and seasonal factors of the time series may changing over time.
* Several equivalent formulations of exponential smoothing:

1. 
2. Forecasting form: 
3. Smoothing equation: 
4. Error correction form: , where



* Estimating the smoothing parameter *α:*

The optimal value of the smoothing parameter can be estimated by maximizing the following SSE:



* Historically, the exponential smoothing methods were intuitive methods not based on any formal statistical models, but the state-space models have now provided a statistical framework for the exponential smoothing methods.
* In the R package{forecast}, function ses() can be used to fit the simple exponential smoothing models:

We use stock price data as an example to illustrate the simple exponential smoothing methods.

Three models will be fitted:

1. model 1 *α = 0.2*
2. model 2 *α = 0.6*
3. model 3 *α = 0.9333* (optimal)

stock=read.table("http://people.usm.maine.edu/cpeng/STA585/TS/stockprice.txt")

price=stock$V1[1:35]

fit1 = ses(price, alpha=0.2, initial="optimal", h=3)

fit2 = ses(price, alpha=0.6, initial="simple", h=3)

fit3 = ses(price, h=3) ## alpha is unspecified, it will be estimated

plot(fit1, plot.conf=FALSE, ylab="Stock Price",

xlab="Time", main="", fcol="white", type="o", lwd=2)

title("Comparison among exponential smoothing models")

lines(fitted(fit1), col="blue", type="o")

lines(fitted(fit2), col="red", type="o")

lines(fitted(fit3), col="darkgreen", type="o")

points(fit1$mean, col="blue", pch=16) ## plot forecast values

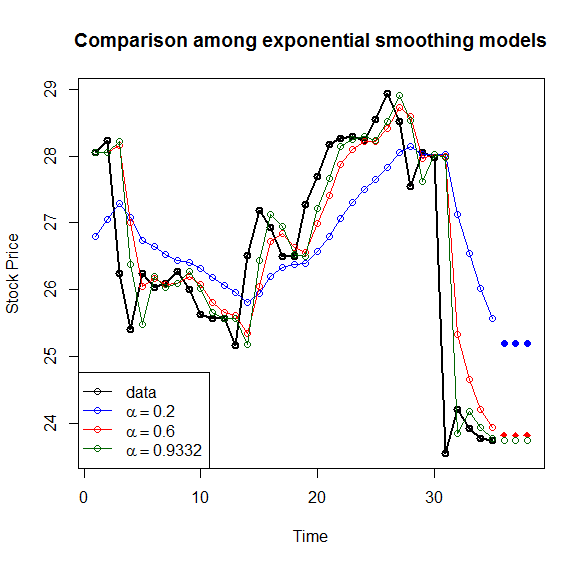
points(fit2$mean, col="red", pch=18)

points(fit3$mean, col="darkgreen", pch=21)

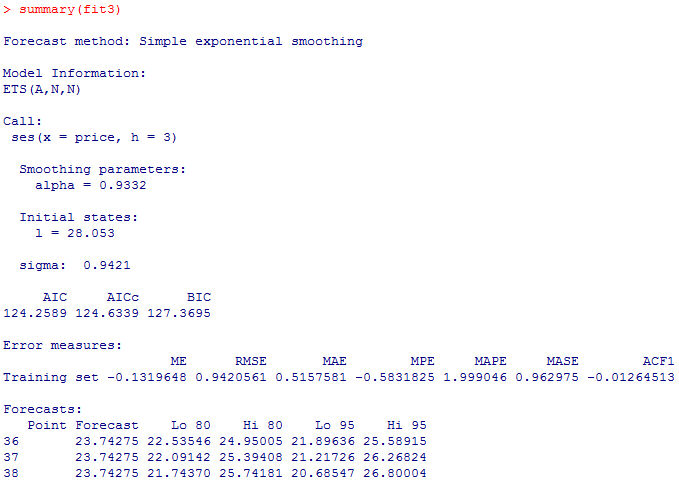
legend("bottomleft",lty=1, col=c(1,"blue","red","darkgreen"),

c("data", expression(alpha == 0.2), expression(alpha == 0.6),

expression(alpha == 0.9332)),pch=1)



summary(fit3)



**Holt's Smoothing Model: Linear Trend**

* Holt generalized simple exponential smoothing by adding a trend parameter to allow forecasting of data with a ***linear trend***.
* The Holt’s forecast model has two components: Level and Trend. To be more specific, the Holt’s model is defined to be



where level component is , trend component is  and *h* is the number of periods to be forecasted, *α* is the smoothing parameter for the level, 0≤ *α* ≤1 and *β*∗ is the smoothing parameter for the trend, 0≤ *β*∗ ≤1 (we denote this as *β*∗.

* *ℓt*−*ℓt*−1 is the difference of trends between last and present time points. The estimated level of trend of next time point is estimated as a weighted average of ℓ*t*−*ℓt*−1 and the previous estimated trend *bt*−1.
* The error correction form of the level and the trend equations are given below





where . The estimated smoothing parameters *α* and *β*∗ are the values that minimize the SSE, sum of squared errors *e*t.

* R function holt() in package {forecast} will be used to fit the smoothing model

beverage = read.table("http://people.usm.maine.edu/cpeng/STA585/TS/beverage01.txt")

bev=beverage$V1[1:35]

fit0 = holt(bev, initial="simple", h=5) ### optimal alpha and beta

fit1 = holt(bev, alpha=0.8, beta=0.2, initial="simple", h=5)

###### plot the original data

plot(fit0, lwd=2,type="o", ylab="Beverage", xlab="Time",

fcol="white", plot.conf=FALSE)

lines(fitted(fit0), col="red")

lines(fitted(fit1), col="blue")

points(fit0, col="black", pch=1)

points(fitted(fit0), col="red", pch=16)

points(fitted(fit1), col="blue", pch=22)

#######################

lines(fit0$mean, col="red", type="o")

lines(fit1$mean, col="blue", type="o")

legend("bottomright", lty=1, col=c("red","black","blue"),pch=c(16,1,22),

c("Holt's Linear(optiomal)","Data","Holt's linear trend"))



Air passengers data:

install.packages("forecast")

libray(forecast)

air = window(ausair,start=1990,end=2004)

fit0 = holt(air, initial="simple", h=5) ### optimal alpha and beta

fit1 = holt(air, alpha=0.8, beta=0.2, initial="simple", h=5)

###### plot the original data

plot(fit0, lwd=2,type="o", ylab="Air passengers(millions)", xlab="Time",

fcol="white", plot.conf=FALSE)

lines(fitted(fit0), col="red")

lines(fitted(fit1), col="blue")

points(fit0, col="black", pch=1)

points(fitted(fit0), col="red", pch=16)

points(fitted(fit1), col="blue", pch=22)

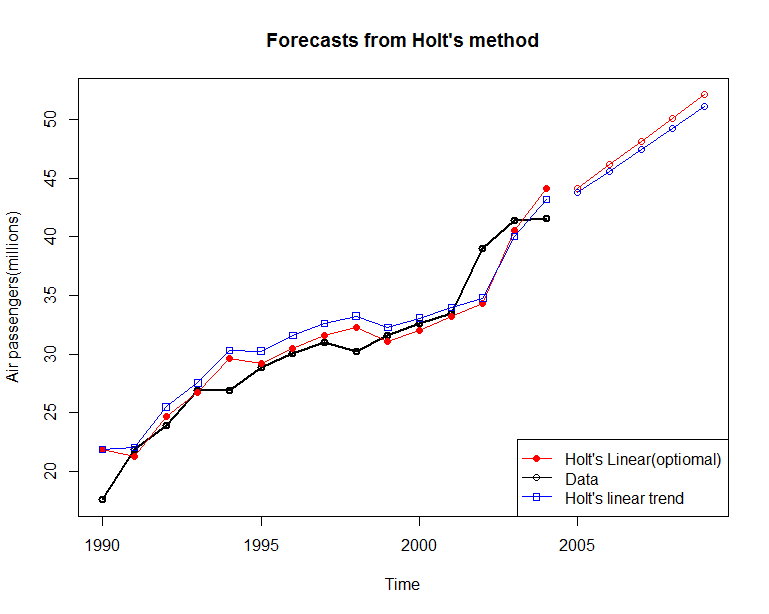
#######################

lines(fit0$mean, col="red", type="o")

lines(fit1$mean, col="blue", type="o")

legend("bottomright", lty=1, col=c("red","black","blue"),pch=c(16,1,22),

c("Holt's Linear(optiomal)","Data","Holt's linear trend"))



**Comment**: Since beverage data does not have a linear trend. Holt smoothing model does not work well. However, the air passenger data has a linear trend, the Holt’s model works very well.

# Exponential Trend Method

* A simple modification to Holt’s leaner trend smoothing model leads to an exponential version of Holt’s model as following

Forecast Model: 

Level: 

Exp Trend: 

where *bt* is the estimated growth rate and h is the number of periods to be predicted. represents the multiplicative increment of the exponential trend.

* Let , the corresponding error correction form is given by



and



* The smoothing parameters α and β\* can be estimated by minimizing the SSE define by



* R function holt() can fit Holt smoothing models with exponential trend



beverage = read.table("http://people.usm.maine.edu/cpeng/STA585/TS/beverage01.txt")

bev=beverage$V1[1:30]

### optimal alpha and beta

fit0 = holt(bev, initial="simple", exponential=TRUE,h=5) fit1 = holt(bev, alpha=0.8, beta=0.2, exponential=TRUE, initial="simple", h=5)

###### plot the original data

plot(fit0, lwd=2,type="o", ylab="Beverage", xlab="Time",

fcol="white", plot.conf=FALSE)

lines(fitted(fit0), col="red")

lines(fitted(fit1), col="blue")

points(fit0, col="black", pch=1)

points(fitted(fit0), col="red", pch=16)

points(fitted(fit1), col="blue", pch=22)

#######################

lines(fit0$mean, col="red", type="o")

lines(fit1$mean, col="blue", type="o")

legend("topleft", lty=1, col=c("red","black","blue"),pch=c(16,1,22),

c("Holt's Exp Trend(optiomal)","Data","Holt's Exp trend"))

Air passenger data

air = window(ausair,start=1990,end=2004)

fit0 = holt(air, initial="simple", exponential=TRUE,h=5) ### optimal alpha and beta

fit1 = holt(air, alpha=0.8, beta=0.2, exponential=TRUE, initial="simple", h=5)

###### plot the original data

plot(fit0, lwd=2,type="o", ylab="Air Passengers", xlab="Time",

fcol="white", plot.conf=FALSE)

lines(fitted(fit0), col="red")

lines(fitted(fit1), col="blue")

points(fit0, col="black", pch=1)

points(fitted(fit0), col="red", pch=16)

points(fitted(fit1), col="blue", pch=22)

#######################

lines(fit0$mean, col="red", type="o")

lines(fit1$mean, col="blue", type="o")

legend("topleft", lty=1, col=c("red","black","blue"),pch=c(16,1,22),

c("Holt's Exp Trend(optiomal)","Data","Holt's Exp trend"))



**Comment**: As expected,

1. the smoothing model with exponential trend works equally well as the linear trend model for air passenger data;
2. the smoothing model with exponential trend works better than the linear trend model for the beverage data;

# Damped Trend Methods

* Empirical evidence indicates that these methods tend to over-forecast, especially for longer forecast horizons. Gardner's damped trend models is shown to be effective in improving accuracy for prediction.
* In addition to the two smoothing parameters α and β\* in linear and exponential trend models, Gardener added a third parameter ϕ (0< ϕ < 1) that damps the trend as h gets bigger.
* ***Additive Model*** Specification:

Forecast Model: 

Level: 

Trend: 

where ϕ is the damping parameter. The corresponding error correction formulation is specified as follows





where . The smoothing parameters *α* and *β*∗ and damping parameter ϕ can be estimated by minimize the sum of squared errors *e*t (SSE).

* ***Multiplicative Model*** Specification:

Forecast Model: 

Level: 

Exp Trend: 

Since this is modification of the smoothing model with exponential trend, *bt* and h are interpreted in the same way as in exponential trend smoothing model and ϕ is between 0 and 1, exclusively.

Let , the corresponding error correction form is given by



and



* We use Australia GDP data as an example to illustrate the application of R function holt() for the damped trend models (both additive and multiplicative models).

ausgdp = read.table("https://sta321.s3.amazonaws.com/ausgdp.txt")

ausgdp0=ts(ausgdp$V1,frequency=4,start=1971+2/4)

ausgdp1=ausgdp$V1

fit1 = ses(ausgdp1)

fit2 = holt(ausgdp1)

fit3 = holt(ausgdp1,exponential=TRUE)

fit4 = holt(ausgdp1,damped=TRUE) ## additive damping

fit5 = holt(ausgdp1,exponential=TRUE,damped=TRUE) ## multiplicative dampling

# Results for first model:

fit1$model

accuracy(fit1) # training set

accuracy(fit1,ausgdp1) # test set

###

plot(fit3, type="o", ylab="Australia GDP",flwd=1, plot.conf=FALSE,

main="Comparison of various smoothing models")

lines(fitted(fit1),col=2)

lines(fitted(fit2),col=3)

lines(fitted(fit4),col=5)

lines(fitted(fit5),col=6)

######

lines(fit1$mean,col=2)

lines(fit2$mean,col=3)

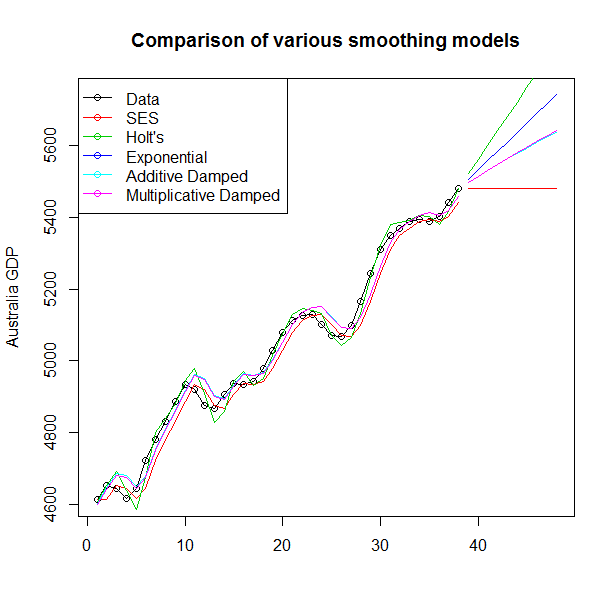
lines(fit4$mean,col=5)

lines(fit5$mean,col=6)

legend("topleft", lty=1, pch=1, col=1:6,

c("Data","SES","Holt's","Exponential",

"Additive Damped","Multiplicative Damped"))



**Holt-Winters Model**

* Next model Holt-Winters developed from exponential smoothing is used to deal with time series containing both trend and seasonal variation.
* The two main HW models are the Additive Model for time series exhibiting constant seasonality and the Multiplicative Model for time series exhibiting increasing seasonality.
* The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series.
* The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level *ℓt*, one for trend *bt*, and one for the seasonal component denoted by *st*, with smoothing parameters *α*, *β*∗ and *γ*.
* Additive Holt-Winters Model

Level: 

Trend: 

Seasonality: 

Forecast Model: 

where *s* is the length of the seasonal cycle, for ,  and . *K* is the number of periods to e predicted.

*Interpretations of individual components*:

1. ***Level***: the current level is the weighted average of the difference between the current observation and previous seasonality and the sum of the previous level and trend.
2. ***Trend***: the current trend is the weighted average of previous trend and the difference of current level and the previous level.
3. ***Seasonality***: the current seasonality is the weight average of previous seasonality and the difference between the current observation and the current level.

Let be the one-step training error, we can re-express level, trend and seasonality equation in the following form:

Level: 

Trend: 

Seasonality: 

* Multiplicative Holt-Winters Model: An alternative Holt-Winters’ model multiplies the forecast by a seasonal factor. Its equations are:

Level: 

Trend: 

Seasonality: 

Forecast Model: 

where *s* is the length of the seasonal cycle, for ,  and . *K* is the number of periods to e predicted.

Similarly, we denote the one-step error for the multiplicative model to be , the error correction representation of the model can be written as

Level: 

Trend: 

Seasonality: 

* Holt-Winters Model with a Damped Trend and Multiplicative Seasonality. This is a simple modification of the HW multiplicative model with a damped trend. The model formulation is given below

Level: 

Trend: 

Seasonality: 

Forecast Model: 

where is the damping coefficient .

* Three R functions can be used to fit Holt-Winters model: hw(),and ets() and HoltWinters() in package {forecast}.

1. The estimation of parameters used in function HoltWinters() uses optim() which requires the initial values for the parameters and . The default initial values of these parameters are 0.3, 0.1 and 0.1. You can provide your own more accurate values. HoltWinters() is using heuris­tic val­ues for the ini­tial states and then esti­mat­ing the smooth­ing para­me­ters by opti­miz­ing the MSE.
2. ets() also use optim() to estimated the parameters and the initial states by opti­miz­ing the like­li­hood func­tion (which is only equiv­a­lent to opti­miz­ing the MSE for the lin­ear addi­tive models).
3. hw() can has an option to fit a HW model with a damped trend. HoltWinters() does not have the option.

* **Examples**: We use an artificial data to illustrate how to R functions to fit various HW models

1. HW models: additive, multiplicative and damped HW models with R function hw()

dat01=read.table("http://people.usm.maine.edu/cpeng/sta585/data/ts3.txt")

datset=ts(dat01$V1+55, start=2000, frequency=12)

## model building

fit1 = hw(datset,h=12, seasonal="additive") # default h = 10

fit2 = hw(datset,h=12, seasonal="multiplicative")

fit3 = hw(datset,h=12, seasonal="additive",damped=TRUE)

### plots

plot(fit2,ylab="Artificial Series",main="Various Holt-Winters Models",

plot.conf=FALSE, type="o", fcol="white", xlab="Year")

lines(fitted(fit1), col="red", lty=2)

lines(fitted(fit2), col="green", lty=2)

lines(fitted(fit3), col="blue", lty=2)

##

lines(fit1$mean, type="o", col="red")

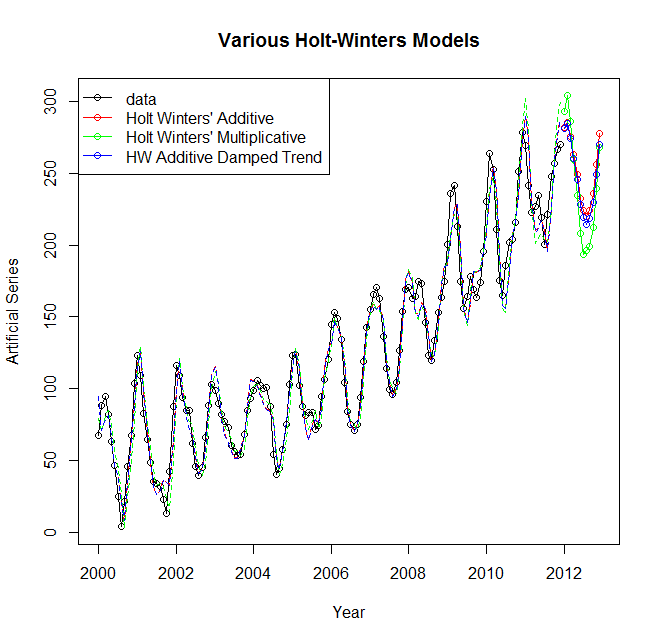
lines(fit2$mean, type="o", col="green")

lines(fit3$mean, type="o", col="blue")

###

legend("topleft",lty=1, pch=1, col=c("black", "red", "green", "blue"),

c("data","Holt Winters' Additive","Holt Winters' Multiplicative", "HW Additive Damped Trend"))



1. HW models: additive, multiplicative and damped HW models with R function HoltWinters(). Since HoltWinters() does not have an option to make predict values directly. We need to use R function [predict.HoltWinters](http://127.0.0.1:27274/library/stats/help/predict.HoltWinters)() to make prediction.

dat01=read.table("http://people.usm.maine.edu/cpeng/sta585/data/ts3.txt")

datset=ts(dat01$V1+55, start=2000, frequency=12)

## model building

fit1 = HoltWinters(datset, seasonal="additive") # default h = 10

fit2 = HoltWinters(datset, seasonal="multiplicative")

### plots

plot(fit1,ylab="Artificial Series",main="Various Holt-Winters Models", type="o", xlab="Year")

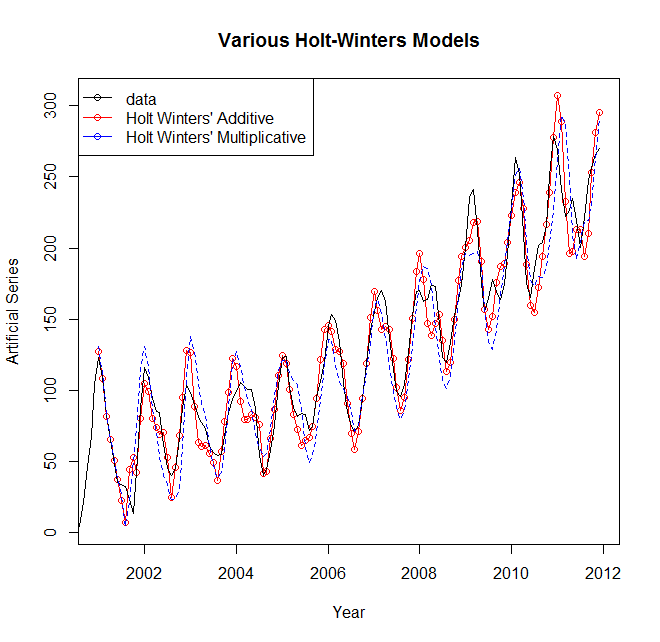
#lines(fitted(fit1)[,1], col="red", lty=2)

lines(fitted(fit2)[,1], col="blue", lty=2)

###

legend("topleft",lty=1, pch=1, col=c("black", "red", "blue"),

c("data","Holt Winters' Additive","Holt Winters' Multiplicative"))



pred = predict(fit1, 50, prediction.interval = TRUE)

pred2 = predict(fit2, 50, prediction.interval = TRUE)

plot(fit1, pred, lwd=2, main="HW Models with R Function HoltWinters()")

lines(pred[,2], col="red", lwd=2)

lines(pred[,3], col="red", lwd=2)

lines(fitted(fit2)[,1], col="blue", lty=1, lwd=2)

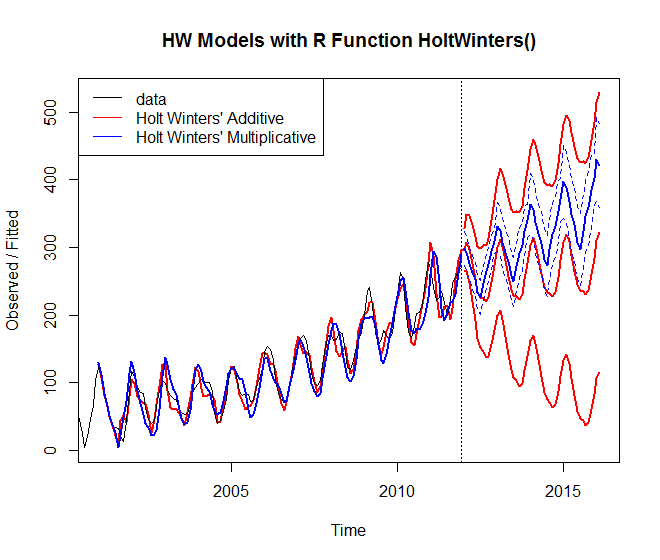
lines(pred2[,2], col="blue", lty=2)

lines(pred2[,1], col="blue", lty=1, lwd=2)

lines(pred2[,3], col="blue", lty=2)

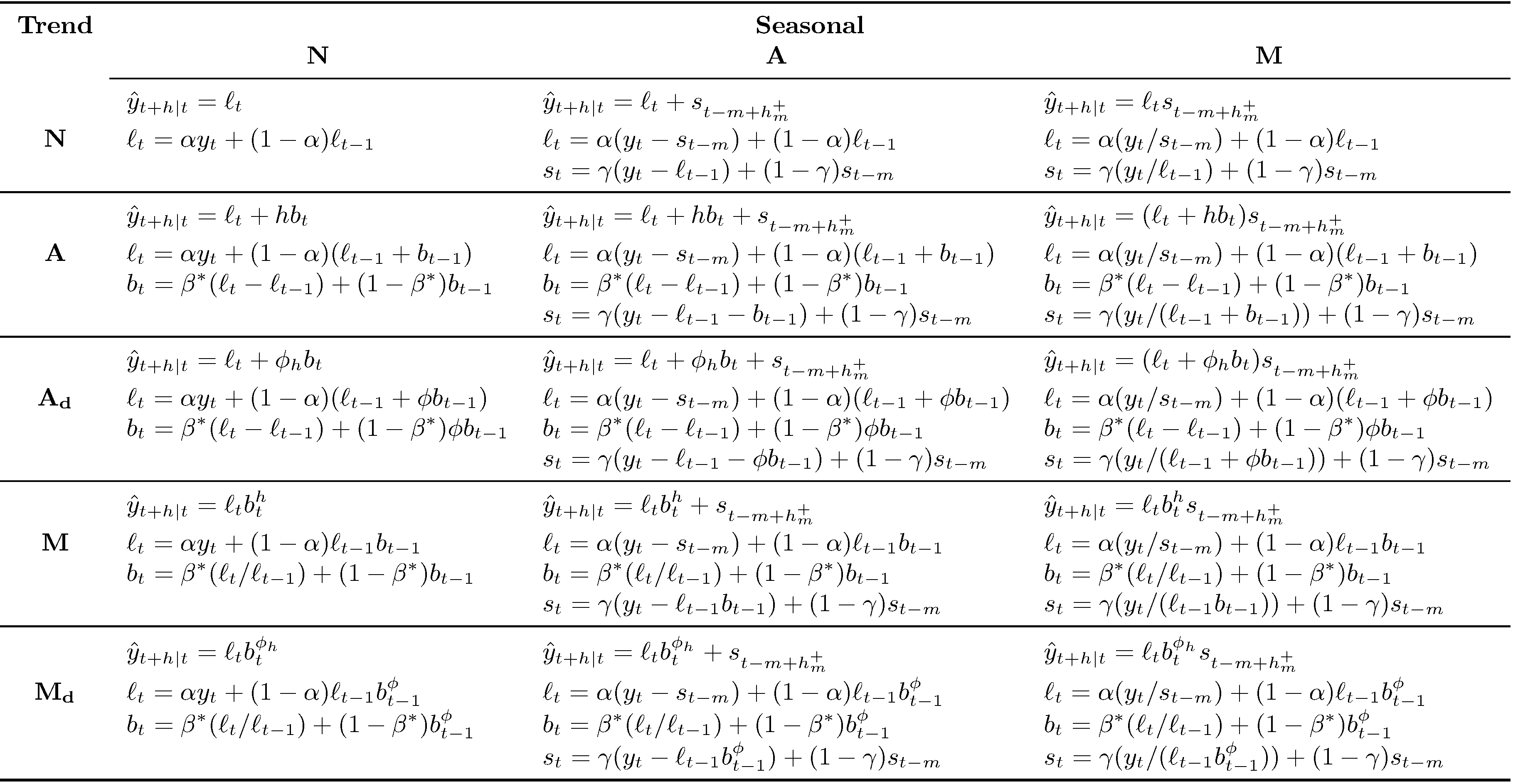
legend("topleft",lty=1, col=c("black", "red", "blue"),

c("data","Holt Winters' Additive","Holt Winters' Multiplicative"))

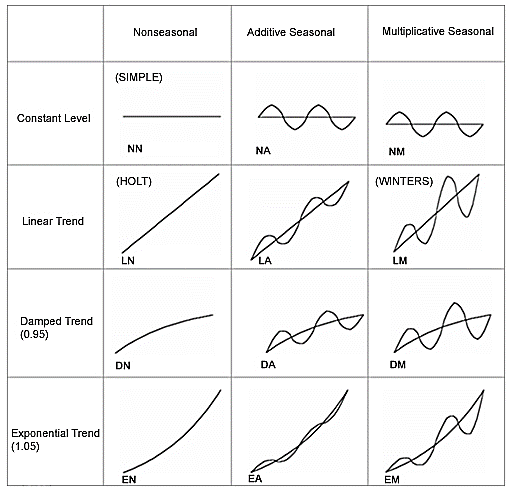


**Model Classification and Identification**

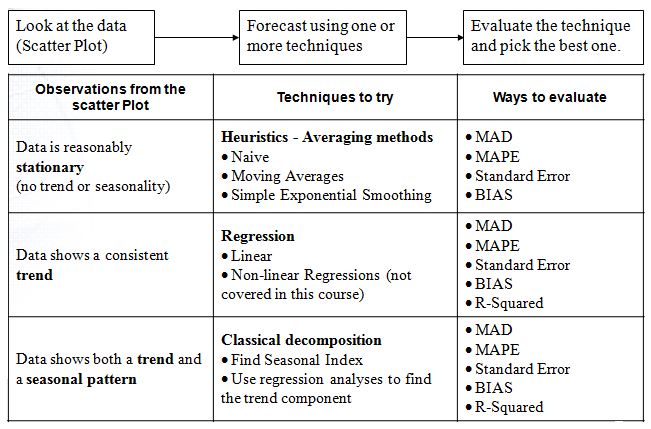
* There are three potential components in either multiplicative or additive time series model. Following table lists possible HW models



* It is easy to make a time series plot to visualize various patterns. Following figures can be used as an aid to identify appropriate type of HW models.



* Model building process



**Concluding Remarks on Assumption**

* There is no normality assumption in *fitting* an exponential smoothing model. Even if maximum likelihood estimation is used with a Gaussian likelihood, the estimates will still be good under almost all residual distributions.
* There is also no normality assumption when producing point forecasts from an exponential smoothing model.
* However, there is often a normality assumption when producing prediction intervals from an exponential smoothing model. But this assumption is easily removed by using bootstrap prediction intervals.
* For example, if you use R, then the following code produces forecasts from a multiplicative Holt-Winters model with no normality assumption for the prediction intervals:

library(forecast)

fcast = hw(x, seasonal="multiplicative", bootstrap=TRUE, simulate=TRUE)

* By the way, you don't need to worry about normality of residuals in regression either, except if you produce prediction intervals.
* As with most statistical models, the estimates are more efficient if the residuals are normally distributed, but they are still consistent and unbiased if the residuals are non-normal.

**Additional Examples**

### Case study

souvenir = scan("http://people.usm.maine.edu/cpeng/sta585/data/fancy.dat")

souvenir.ts = ts(souvenir, start=c(1987,1), freq=12)

log.souvenir.ts = log(souvenir.ts)

plot(log.souvenir.ts)

###

souvenir.forecast= HoltWinters(log.souvenir.ts)

souvenir.forecast

plot(souvenir.forecast)

###

install.packages("forecast")

library(forecast)

souvenir.forecast2 = forecast.HoltWinters(souvenir.forecast, h = 48)

souvenir.forecast2

plot(souvenir.forecast2)

#####

www = "http://people.usm.maine.edu/cpeng/sta585/data/wine.dat"

wine.dat = read.table(www, header = T)

sweetw.ts = ts(wine.dat$sweetw, start=c(1980,1), freq=12)

sweetw.hw = HoltWinters(sweetw.ts, seasonal="mult")

sweetw.hw

install.packages("forecast")

library(forecast)

sweetw.forecast2 = forecast.HoltWinters(sweetw.forecast, h = 48)

sweetw.forecast2

plot(sweetw.forecast2)

