MONASH University

Rob J Hyndman

State space models

1: Exponential smoothing

Outline

Outline

State space perspective

- Observed data: $y_1, \dots, y_T$.
- Unobserved state: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T$.

- Forecast $\hat{y}_{T+h|T} = \mathrm{E}(y_{T+h}|\boldsymbol{x}_T)$.
- The "forecast variance" is $\mathrm{Var}(y_{T+h}|\boldsymbol{x}_T)$.
- A prediction interval or "interval forecast" is a range of values of y_{T+h} with high probability.

State space perspective

- Observed data: $y_1, \dots, y_T$.
- Unobserved state: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T$.

- Forecast $\hat{y}_{T+h|T} = \mathrm{E}(y_{T+h}|\boldsymbol{x}_T)$.
- The "forecast variance" is $\mathrm{Var}(y_{T+h}|\boldsymbol{x}_T)$.
- A prediction interval or "interval forecast" is a range of values of y_{T+h} with high probability.

State space perspective

- Observed data: $y_1, \dots, y_T$.
- Unobserved state: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T$.

- Forecast $\hat{y}_{T+h|T} = \text{E}(y_{T+h}|\boldsymbol{x}_T)$.
- The "forecast variance" is $\text{Var}(y_{T+h}|\boldsymbol{x}_T)$.
- A prediction interval or "interval forecast" is a range of values of y_{T+h} with high probability.

State space perspective

- Observed data: $y_1, \dots, y_T$.
- Unobserved state: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T$.

- Forecast $\hat{y}_{T+h|T} = \mathrm{E}(y_{T+h}|\boldsymbol{x}_T)$.
- The “forecast variance” is $\mathrm{Var}(y_{T+h}|\boldsymbol{x}_T)$.
- A prediction interval or “interval forecast” is a range of values of y_{T+h} with high probability.

Outline

Simple Exponential Smoothing

Component form

Forecast equation $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

$\ell_1 = \alpha y_1 + (1-\alpha)\ell_0$

Simple Exponential Smoothing

Component form

Forecast equation	$\hat{y}_{t+h|t} = \ell_t$
Smoothing equation	$\ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1}$

$\ell_1 = \alpha y_1 + (1 - \alpha)\ell_0$

Simple Exponential Smoothing

Component form

Forecast equation	$\hat{y}_{t+h\|t} = \ell_t$
Smoothing equation	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

$$\ell_1 = \alpha y_1 + (1-\alpha)\ell_0$$

$$\ell_2 = \alpha y_2 + (1-\alpha)\ell_1 = \alpha y_2 + \alpha(1-\alpha)y_1 + (1-\alpha)^2\ell_0$$

Simple Exponential Smoothing

Component form

Forecast equation	$\hat{y}_{t+h\|t} = \ell_t$
Smoothing equation	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

$$\ell_1 = \alpha y_1 + (1-\alpha)\ell_0$$

$$\ell_2 = \alpha y_2 + (1-\alpha)\ell_1 = \alpha y_2 + \alpha(1-\alpha)y_1 + (1-\alpha)^2\ell_0$$

$$\ell_3 = \alpha y_3 + (1-\alpha)\ell_2 = \sum_{j=0}^{2} \alpha(1-\alpha)^j y_{3-j} + (1-\alpha)^3\ell_0$$

Simple Exponential Smoothing

Component form

Forecast equation $\qquad \hat{y}_{t+h|t} = \ell_t$

Smoothing equation $\qquad \ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

$$\ell_1 = \alpha y_1 + (1-\alpha)\ell_0$$

$$\ell_2 = \alpha y_2 + (1-\alpha)\ell_1 = \alpha y_2 + \alpha(1-\alpha)y_1 + (1-\alpha)^2\ell_0$$

$$\ell_3 = \alpha y_3 + (1-\alpha)\ell_2 = \sum_{j=0}^{2} \alpha(1-\alpha)^j y_{3-j} + (1-\alpha)^3\ell_0$$

$$\vdots$$

$$\ell_t = \sum_{j=0}^{t-1} \alpha(1-\alpha)^j y_{t-j} + (1-\alpha)^t\ell_0$$

Simple Exponential Smoothing

Forecast equation

$$\hat{y}_{t+h|t} = \sum_{j=1}^{t} \alpha(1-\alpha)^{t-j} y_j + (1-\alpha)^t \ell_0, \qquad (0 \le \alpha \le 1)$$

	Weights assigned to observations for:			
Observation	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_t	0.2	0.4	0.6	0.8
y_{t-1}	0.16	0.24	0.24	0.16
y_{t-2}	0.128	0.144	0.096	0.032
y_{t-3}	0.1024	0.0864	0.0384	0.0064
y_{t-4}	$(0.2)(0.8)^4$	$(0.4)(0.6)^4$	$(0.6)(0.4)^4$	$(0.8)(0.2)^4$
y_{t-5}	$(0.2)(0.8)^5$	$(0.4)(0.6)^5$	$(0.6)(0.4)^5$	$(0.8)(0.2)^5$

- Limiting cases: $\alpha \to 1$, $\alpha \to 0$.

Simple Exponential Smoothing

Forecast equation

$$\hat{y}_{t+h|t} = \sum_{j=1}^{t} \alpha(1-\alpha)^{t-j} y_j + (1-\alpha)^t \ell_0, \qquad (0 \le \alpha \le 1)$$

Observation	Weights assigned to observations for: $\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_t	0.2	0.4	0.6	0.8
y_{t-1}	0.16	0.24	0.24	0.16
y_{t-2}	0.128	0.144	0.096	0.032
y_{t-3}	0.1024	0.0864	0.0384	0.0064
y_{t-4}	$(0.2)(0.8)^4$	$(0.4)(0.6)^4$	$(0.6)(0.4)^4$	$(0.8)(0.2)^4$
y_{t-5}	$(0.2)(0.8)^5$	$(0.4)(0.6)^5$	$(0.6)(0.4)^5$	$(0.8)(0.2)^5$

- Limiting cases: $\alpha \to 1$, $\quad \alpha \to 0$.

Simple Exponential Smoothing

Forecast equation

$$\hat{y}_{t+h|t} = \sum_{j=1}^{t} \alpha(1-\alpha)^{t-j} y_j + (1-\alpha)^t \ell_0, \qquad (0 \le \alpha \le 1)$$

	Weights assigned to observations for:			
Observation	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_t	0.2	0.4	0.6	0.8
y_{t-1}	0.16	0.24	0.24	0.16
y_{t-2}	0.128	0.144	0.096	0.032
y_{t-3}	0.1024	0.0864	0.0384	0.0064
y_{t-4}	$(0.2)(0.8)^4$	$(0.4)(0.6)^4$	$(0.6)(0.4)^4$	$(0.8)(0.2)^4$
y_{t-5}	$(0.2)(0.8)^5$	$(0.4)(0.6)^5$	$(0.6)(0.4)^5$	$(0.8)(0.2)^5$

- Limiting cases: $\alpha \to 1$, $\quad \alpha \to 0$.

Simple Exponential Smoothing

Component form

Forecast equation $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

State space form

Observation equation $y_t = \ell_{t-1} + e_t$

State equation $\ell_t = \ell_{t-1} + \alpha e_t$

- $e_t = y_t - \ell_{t-1} = y_t - \hat{y}_{t|t-1}$ for $t = 1, \dots, T$, the one-step within-sample forecast error at time t.
- ℓ_t is an unobserved “state”.
- Need to estimate α and ℓ_0.

Simple Exponential Smoothing

Component form

Forecast equation $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

State space form

Observation equation $y_t = \ell_{t-1} + e_t$

State equation $\ell_t = \ell_{t-1} + \alpha e_t$

- $e_t = y_t - \ell_{t-1} = y_t - \hat{y}_{t|t-1}$ for $t = 1, \dots, T$, the one-step within-sample forecast error at time t.
- ℓ_t is an unobserved "state".
- Need to estimate α and ℓ_0.

Simple Exponential Smoothing

Component form

Forecast equation	$\hat{y}_{t+h\|t} = \ell_t$
Smoothing equation	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

State space form

Observation equation	$y_t = \ell_{t-1} + e_t$
State equation	$\ell_t = \ell_{t-1} + \alpha e_t$

- $e_t = y_t - \ell_{t-1} = y_t - \hat{y}_{t|t-1}$ for $t = 1, \dots, T$, the one-step within-sample forecast error at time t.
- ℓ_t is an unobserved "state".
- Need to estimate α and ℓ_0.

Simple Exponential Smoothing

Component form

Forecast equation	$\hat{y}_{t+h\|t} = \ell_t$
Smoothing equation	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$

State space form

Observation equation	$y_t = \ell_{t-1} + e_t$
State equation	$\ell_t = \ell_{t-1} + \alpha e_t$

- $e_t = y_t - \ell_{t-1} = y_t - \hat{y}_{t|t-1}$ for $t = 1, \dots, T$, the one-step within-sample forecast error at time t.
- ℓ_t is an unobserved "state".
- Need to estimate α and ℓ_0.

SES in R

```
library(fpp)

fit <- ses(oil, h=3)

plot(fit)

summary(fit)
```

Outline

Holt's linear trend

Component form

$$
\begin{aligned}
\text{Forecast} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\
\text{Level} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{Trend} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1},
\end{aligned}
$$

- Two smoothing parameters α and β^* $(0 \le \alpha, \beta^* \le 1)$.
- ℓ_t level: weighted average between y_t one-step ahead forecast for time t, $(\ell_{t-1} + b_{t-1} = \hat{y}_{t|t-1})$
- b_t trend (slope): weighted average of $(\ell_t - \ell_{t-1})$ and b_{t-1}, current and previous estimate of the trend.

Holt's linear trend

Component form

$$
\begin{aligned}
\text{Forecast} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\
\text{Level} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{Trend} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1},
\end{aligned}
$$

- Two smoothing parameters α and β^* $(0 \leq \alpha, \beta^* \leq 1)$.
- ℓ_t level: weighted average between y_t one-step ahead forecast for time t, $(\ell_{t-1} + b_{t-1} = \hat{y}_{t|t-1})$
- b_t trend (slope): weighted average of $(\ell_t - \ell_{t-1})$ and b_{t-1}, current and previous estimate of the trend.

Holt's linear trend

Component form

$$\begin{aligned} \text{Forecast} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\ \text{Level} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}, \end{aligned}$$

- Two smoothing parameters α and β^* $(0 \leq \alpha, \beta^* \leq 1)$.
- ℓ_t level: weighted average between y_t one-step ahead forecast for time t, $(\ell_{t-1} + b_{t-1} = \hat{y}_{t|t-1})$
- b_t trend (slope): weighted average of $(\ell_t - \ell_{t-1})$ and b_{t-1}, current and previous estimate of the trend.

Holt's linear trend

Component form

$$\begin{aligned} \text{Forecast} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\ \text{Level} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}, \end{aligned}$$

- Two smoothing parameters α and β^* $(0 \leq \alpha, \beta^* \leq 1)$.
- ℓ_t level: weighted average between y_t one-step ahead forecast for time t, $(\ell_{t-1} + b_{t-1} = \hat{y}_{t|t-1})$
- b_t trend (slope): weighted average of $(\ell_t - \ell_{t-1})$ and b_{t-1}, current and previous estimate of the trend.

Holt's linear trend

Component form

$$
\begin{aligned}
\text{Forecast} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\
\text{Level} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{Trend} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1},
\end{aligned}
$$

State space form

$$
\begin{aligned}
\text{Observation equation} \quad & y_t = \ell_{t-1} + b_{t-1} + e_t \\
\text{State equations} \quad & \ell_t = \ell_{t-1} + b_{t-1} + \alpha e_t \\
& b_t = b_{t-1} + \beta e_t
\end{aligned}
$$

- $\beta = \alpha\beta^*$
- $e_t = y_t - (\ell_{t-1} + b_{t-1}) = y_t - \hat{y}_{t|t-1}$
- Need to estimate $\alpha, \beta, \ell_0, b_0$.

Holt's linear trend

Component form

$$\begin{aligned} \text{Forecast} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\ \text{Level} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}, \end{aligned}$$

State space form

$$\begin{aligned} \text{Observation equation} \quad & y_t = \ell_{t-1} + b_{t-1} + e_t \\ \text{State equations} \quad & \ell_t = \ell_{t-1} + b_{t-1} + \alpha e_t \\ & b_t = b_{t-1} + \beta e_t \end{aligned}$$

- $\beta = \alpha\beta^*$
- $e_t = y_t - (\ell_{t-1} + b_{t-1}) = y_t - \hat{y}_{t|t-1}$
- Need to estimate $\alpha, \beta, \ell_0, b_0$.

Holt's linear trend

Component form

$$\begin{aligned} \text{Forecast} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\ \text{Level} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}, \end{aligned}$$

State space form

$$\begin{aligned} \text{Observation equation} \quad & y_t = \ell_{t-1} + b_{t-1} + e_t \\ \text{State equations} \quad & \ell_t = \ell_{t-1} + b_{t-1} + \alpha e_t \\ & b_t = b_{t-1} + \beta e_t \end{aligned}$$

- $\beta = \alpha\beta^*$
- $e_t = y_t - (\ell_{t-1} + b_{t-1}) = y_t - \hat{y}_{t|t-1}$
- Need to estimate $\alpha, \beta, \ell_0, b_0$.

Holt's linear trend

Component form

Forecast	$\hat{y}_{t+h\|t} = \ell_t + hb_t$
Level	$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$
Trend	$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1},$

State space form

Observation equation	$y_t = \ell_{t-1} + b_{t-1} + e_t$
State equations	$\ell_t = \ell_{t-1} + b_{t-1} + \alpha e_t$
	$b_t = b_{t-1} + \beta e_t$

- $\beta = \alpha\beta^*$
- $e_t = y_t - (\ell_{t-1} + b_{t-1}) = y_t - \hat{y}_{t|t-1}$
- Need to estimate $\alpha, \beta, \ell_0, b_0$.

Holt’s method in R

```
fit2 <- holt(ausair, h=5)

plot(fit2)

summary(fit2)
```

Exponential trend

Level and trend are multiplied rather than added:

Component form

$$\hat{y}_{t+h|t} = \ell_t b_t^h$$
$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} b_{t-1})$$
$$b_t = \beta^* \frac{\ell_t}{\ell_{t-1}} + (1-\beta^*) b_{t-1}$$

State space form

Observation equation	$y_t = (\ell_{t-1} b_{t-1}) + e_t$
State equations	$\ell_t = \ell_{t-1} b_{t-1} + \alpha e_t$
	$b_t = b_{t-1} + \beta e_t / \ell_{t-1}$

Trend methods in R

```
fit3 <- holt(air, h=5, exponential=TRUE)
plot(fit3)
summary(fit3)
```

Additive damped trend

Component form

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \cdots + \phi^h) b_t$$
$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}.$$

State space form

Observation equation	$y_t = \ell_{t-1} + \phi b_{t-1} + e_t$
State equations	$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha e_t$
	$b_t = \phi b_{t-1} + \beta e_t$

- Damping parameter $0 < \phi < 1$.
- If $\phi = 1$, identical to Holt's linear trend.
- As $h \to \infty$, $\hat{y}_{T+h|T} \to \ell_T + \phi b_T/(1-\phi)$.
- Short-run forecasts trended, long-run forecasts constant.

Additive damped trend

Component form

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t$$
$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}.$$

State space form

Observation equation	$y_t = \ell_{t-1} + \phi b_{t-1} + e_t$
State equations	$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha e_t$
	$b_t = \phi b_{t-1} + \beta e_t$

- Damping parameter $0 < \phi < 1$.
- If $\phi = 1$, identical to Holt's linear trend.
- As $h \to \infty$, $\hat{y}_{T+h|T} \to \ell_T + \phi b_T/(1-\phi)$.
- Short-run forecasts trended, long-run forecasts constant.

Additive damped trend

Component form

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t$$
$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}.$$

State space form

Observation equation $\quad y_t = \ell_{t-1} + \phi b_{t-1} + e_t$

State equations $\quad \ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha e_t$

$\quad b_t = \phi b_{t-1} + \beta e_t$

- Damping parameter $0 < \phi < 1$.
- If $\phi = 1$, identical to Holt's linear trend.
- As $h \to \infty$, $\hat{y}_{T+h|T} \to \ell_T + \phi b_T/(1-\phi)$.
- Short-run forecasts trended, long-run forecasts constant.

Additive damped trend

Component form

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t$$
$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}.$$

State space form

Observation equation $\quad y_t = \ell_{t-1} + \phi b_{t-1} + e_t$

State equations $\quad \ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha e_t$

$$b_t = \phi b_{t-1} + \beta e_t$$

- Damping parameter $0 < \phi < 1$.
- If $\phi = 1$, identical to Holt's linear trend.
- As $h \to \infty$, $\hat{y}_{T+h|T} \to \ell_T + \phi b_T/(1-\phi)$.
- Short-run forecasts trended, long-run forecasts constant.

Additive damped trend

Component form

$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + (\phi + \phi^2 + \cdots + \phi^h) b_t \\
\ell_t &= \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}.
\end{aligned}$$

State space form

Observation equation	$y_t = \ell_{t-1} + \phi b_{t-1} + e_t$
State equations	$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha e_t$
	$b_t = \phi b_{t-1} + \beta e_t$

- Damping parameter $0 < \phi < 1$.
- If $\phi = 1$, identical to Holt's linear trend.
- As $h \to \infty$, $\hat{y}_{T+h|T} \to \ell_T + \phi b_T/(1-\phi)$.
- Short-run forecasts trended, long-run forecasts constant.

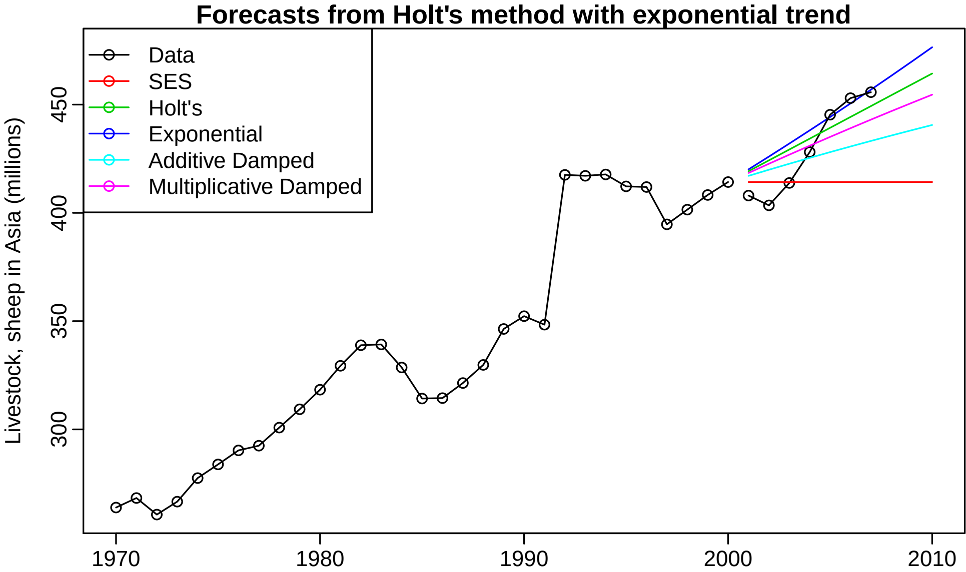
Trend methods in R

```
fit4 <- holt(air, h=5, damped=TRUE)

plot(fit4)

summary(fit4)
```

Example: Sheep in Asia



Outline

Holt-Winters additive method

Holt and Winters extended Holt's method to capture seasonality.

Component form

$$
\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t-m+h_m^+} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},
\end{aligned}
$$

- $h_m^+ = \lfloor (h-1) \mod m \rfloor + 1$ - the largest integer not greater than $(h-1) \mod m$. Ensures estimates from the final year are used for forecasting.
- Parameters: $0 \le \alpha \le 1$, $0 \le \beta^* \le 1$, $0 \le \gamma \le 1-\alpha$ and $m =$ period of seasonality (e.g. m=4 for quarterly data).

Holt-Winters additive method

Holt and Winters extended Holt's method to capture seasonality.

Component form

$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t-m+h_m^+} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},
\end{aligned}$$

- $h_m^+ = \lfloor (h-1) \mod m \rfloor + 1$ - the largest integer not greater than $(h-1) \mod m$. Ensures estimates from the final year are used for forecasting.
- Parameters: $0 \le \alpha \le 1$, $0 \le \beta^* \le 1$, $0 \le \gamma \le 1-\alpha$ and $m =$ period of seasonality (e.g. m=4 for quarterly data).

Holt-Winters additive method

Holt and Winters extended Holt's method to capture seasonality.

Component form

$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t-m+h_m^+} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},
\end{aligned}$$

- $h_m^+ = \lfloor (h-1) \mod m \rfloor + 1$ - the largest integer not greater than $(h-1) \mod m$. Ensures estimates from the final year are used for forecasting.
- Parameters: $0 \le \alpha \le 1$, $0 \le \beta^* \le 1$, $0 \le \gamma \le 1-\alpha$ and $m =$ period of seasonality (e.g. m=4 for quarterly data).

Holt-Winters additive method

Component form

$$
\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t-m+h_m^+} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},
\end{aligned}
$$

State space form

$$
\begin{aligned}
y_t &= \ell_{t-1} + b_{t-1} + s_{t-m} + e_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \alpha e_t \\
b_t &= b_{t-1} + \beta e_t \\
s_t &= s_{t-m} + \gamma e_t.
\end{aligned}
$$

Holt-Winters multiplicative

Component form

$$\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t-m+h_m^+}.$$
$$\ell_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$$
$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}$$

State space form

$$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + e_t$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha e_t/s_{t-m}$$
$$b_t = b_{t-1} + \beta e_t/s_{t-m}$$
$$s_t = s_{t-m} + \gamma e_t/(\ell_{t-1} + b_{t-1}).$$

Seasonal methods in R

```r
aus1 <- hw(austourists)
aus2 <- hw(austourists, seasonal="mult")

plot(aus1)
plot(aus2)

summary(aus1)
summary(aus2)
```

Holt-Winters damped method

Often the single most accurate forecasting method for seasonal data:

$$
\begin{aligned}
\hat{y}_{t+h|t} &= [\ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t]s_{t-m+h_m^+} \\
\ell_t &= \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1} \\
s_t &= \gamma \frac{y_t}{(\ell_{t-1} + \phi b_{t-1})} + (1-\gamma)s_{t-m}
\end{aligned}
$$

State space form

$$
\begin{aligned}
y_t &= (\ell_{t-1} + \phi b_{t-1})s_{t-m} + e_t \\
\ell_t &= \ell_{t-1} + \phi b_{t-1} + \alpha e_t/s_{t-m} \\
b_t &= \phi b_{t-1} + \beta e_t/s_{t-m} \\
s_t &= s_{t-m} + \gamma e_t/(\ell_{t-1} + \phi b_{t-1}).
\end{aligned}
$$

Seasonal methods in R

```
aus3 <- hw(austourists, seasonal="mult",
    damped=TRUE)

summary(aus3)

plot(aus3)
```

Outline

1. The state space perspective
2. Simple exponential smoothing
3. Trend methods
4. Seasonal methods
5. **Taxonomy of exponential smoothing methods**
6. Innovations state space models
7. ETS in R

Exponential smoothing 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

Exponential smoothing methods

Trend Component		Seasonal 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

Exponential smoothing methods

Trend Component		Seasonal 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

Exponential smoothing methods

Trend Component		Seasonal 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
A_d,N: Additive damped trend method

Exponential smoothing methods

Trend Component		Seasonal 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
A_d,N: Additive damped trend method
M,N: Exponential trend method

Exponential smoothing methods

Trend Component		Seasonal 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
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative damped trend method

Exponential smoothing methods

Trend Component		Seasonal 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
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative damped trend method
A,A: Additive Holt-Winters' method

Exponential smoothing methods

Trend Component		Seasonal 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
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative damped trend method
A,A: Additive Holt-Winters' method
A,M: Multiplicative Holt-Winters' method

Exponential smoothing methods

Trend Component		Seasonal 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

There are 15 separate exponential smoothing methods.

Component form

Trend	Seasonal N	Seasonal A	Seasonal M
N	$\hat{y}_{t+h\|t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} - b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{A_d}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} - \phi b_{t-1})) + (1-\gamma)s_{t-m}$
M	$\hat{y}_{t+h\|t} = \ell_t b_t^h$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t b_t^h + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t b_t^h s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1}b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{M_d}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h}$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}^{\phi}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t/(\ell_{t-1}b_{t-1}^{\phi})) + (1-\gamma)s_{t-m}$

Outline

Methods V Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

- Generate same point forecasts but can also generate forecast intervals.
- A stochastic (or random) data generating process that can generate an entire forecast distribution.
- Allow for "proper" model selection.

Methods V Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

- Generate same point forecasts but can also generate forecast intervals.
- A stochastic (or random) data generating process that can generate an entire forecast distribution.
- Allow for "proper" model selection.

Methods V Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

- Generate same point forecasts but can also generate forecast intervals.
- A stochastic (or random) data generating process that can generate an entire forecast distribution.
- Allow for “proper” model selection.

Methods V Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

- Generate same point forecasts but can also generate forecast intervals.
- A stochastic (or random) data generating process that can generate an entire forecast distribution.
- Allow for “proper” model selection.

Methods V Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

- Generate same point forecasts but can also generate forecast intervals.
- A stochastic (or random) data generating process that can generate an entire forecast distribution.
- Allow for “proper” model selection.

ETS models

- Each model has an *observation* equation and *transition* equations, one for each state (level, trend, seasonal), i.e., state space models.
- Two models for each method: one with additive and one with multiplicative errors, i.e., in total 30 models.
- ETS(Error,Trend,Seasonal):
  - Error $= \{A, M\}$
  - Trend $= \{N, A, A_d, M, M_d\}$
  - Seasonal $= \{N, A, M\}$

ETS models

- Each model has an *observation* equation and *transition* equations, one for each state (level, trend, seasonal), i.e., state space models.
- Two models for each method: one with additive and one with multiplicative errors, i.e., in total 30 models.
- ETS(Error,Trend,Seasonal):
  - Error$= \{A, M\}$
  - Trend $= \{N, A, A_d, M, M_d\}$
  - Seasonal $= \{N, A, M\}$

ETS models

- Each model has an *observation* equation and *transition* equations, one for each state (level, trend, seasonal), i.e., state space models.
- Two models for each method: one with additive and one with multiplicative errors, i.e., in total 30 models.
- ETS(Error,Trend,Seasonal):
  - Error $= \{A, M\}$
  - Trend $= \{N, A, A_d, M, M_d\}$
  - Seasonal $= \{N, A, M\}$.

ETS models

- Each model has an *observation* equation and *transition* equations, one for each state (level, trend, seasonal), i.e., state space models.
- Two models for each method: one with additive and one with multiplicative errors, i.e., in total 30 models.
- ETS(Error,Trend,Seasonal):
  - Error$= \{A, M\}$
  - Trend $= \{N, A, A_d, M, M_d\}$
  - Seasonal $= \{N, A, M\}$.

ETS models

- Each model has an *observation* equation and *transition* equations, one for each state (level, trend, seasonal), i.e., state space models.
- Two models for each method: one with additive and one with multiplicative errors, i.e., in total 30 models.
- ETS(Error,Trend,Seasonal):
  - Error$= \{A, M\}$
  - Trend $= \{N, A, A_d, M, M_d\}$
  - Seasonal $= \{N, A, M\}$.

ETS models

- Each model has an *observation* equation and *transition* equations, one for each state (level, trend, seasonal), i.e., state space models.
- Two models for each method: one with additive and one with multiplicative errors, i.e., in total 30 models.
- ETS(Error,Trend,Seasonal):
  - Error$= \{\mathrm{A}, \mathrm{M}\}$
  - Trend $= \{\mathrm{N}, \mathrm{A}, \mathrm{A_d}, \mathrm{M}, \mathrm{M_d}\}$
  - Seasonal $= \{\mathrm{N}, \mathrm{A}, \mathrm{M}\}$.

Exponential smoothing methods

Trend Component		Seasonal 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

General notation E T S : **E**xponen**T**ial **S**moothing

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal 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

General notation E T S : **E**xponen**T**ial **S**moothing

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal 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

General notation

E T S : **E**xponen**T**ial **S**moothing

Trend

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal 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

General notation E T S : **E**xponen**T**ial **S**moothing

Trend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal 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

General notation E T S : **E**xponen**T**ial **S**moothing

Error **T**rend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal 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

General notation   E T S : **E**xponen**T**ial **S**moothing

Error **T**rend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Innovations state space models

- All ETS models can be written in innovations state space form.
- Additive and multiplicative versions give the same point forecasts but different prediction intervals.

General notation E T S : **E**xponen**T**ial **S**moothing

Error **T**rend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

ETS(A,N,N)

Observation equation	$y_t = \ell_{t-1} + \varepsilon_t,$
State equation	$\ell_t = \ell_{t-1} + \alpha\varepsilon_t$

- $e_t = y_t - \hat{y}_{t|t-1} = \varepsilon_t$
- Assume $\varepsilon_t \sim \text{NID}(0, \sigma^2)$
- "innovations" or "single source of error" because same error process, ε_t.

ETS(A,N,N)

Observation equation	$y_t = \ell_{t-1} + \varepsilon_t,$
State equation	$\ell_t = \ell_{t-1} + \alpha\varepsilon_t$

- $e_t = y_t - \hat{y}_{t|t-1} = \varepsilon_t$
- Assume $\varepsilon_t \sim \text{NID}(0, \sigma^2)$
- "innovations" or "single source of error" because same error process, ε_t.

ETS(A,N,N)

Observation equation	$y_t = \ell_{t-1} + \varepsilon_t,$
State equation	$\ell_t = \ell_{t-1} + \alpha\varepsilon_t$

- $e_t = y_t - \hat{y}_{t|t-1} = \varepsilon_t$
- Assume $\varepsilon_t \sim \text{NID}(0, \sigma^2)$
- "innovations" or "single source of error" because same error process, ε_t.

ETS(M,N,N)

SES with multiplicative errors.

- Specify relative errors $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting $\hat{y}_{t|t-1} = \ell_{t-1}$ gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Observation equation	$y_t = \ell_{t-1}(1 + \varepsilon_t)$
State equation	$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$

ETS(M,N,N)

SES with multiplicative errors.

- Specify relative errors $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting $\hat{y}_{t|t-1} = \ell_{t-1}$ gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Observation equation	$y_t = \ell_{t-1}(1 + \varepsilon_t)$
State equation	$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$

ETS(M,N,N)

SES with multiplicative errors.

- Specify relative errors $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting $\hat{y}_{t|t-1} = \ell_{t-1}$ gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Observation equation	$y_t = \ell_{t-1}(1 + \varepsilon_t)$
State equation	$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$

ETS(M,N,N)

SES with multiplicative errors.

- Specify relative errors $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting $\hat{y}_{t|t-1} = \ell_{t-1}$ gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Observation equation	$y_t = \ell_{t-1}(1 + \varepsilon_t)$
State equation	$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$

- Models with additive and multiplicative errors with the same parameters generate the same point forecasts but different prediction intervals

ETS(M,N,N)

SES with multiplicative errors.

- Specify relative errors $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting $\hat{y}_{t|t-1} = \ell_{t-1}$ gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Observation equation	$y_t = \ell_{t-1}(1 + \varepsilon_t)$
State equation	$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$

- Models with additive and multiplicative errors with the same parameters generate the same point forecasts but different prediction intervals.

ETS(M,N,N)

SES with multiplicative errors.

- Specify relative errors $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting $\hat{y}_{t|t-1} = \ell_{t-1}$ gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Observation equation	$y_t = \ell_{t-1}(1 + \varepsilon_t)$
State equation	$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$

- Models with additive and multiplicative errors with the same parameters generate the same point forecasts but different prediction intervals.

ETS(M,N,N)

SES with multiplicative errors.

- Specify relative errors $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting $\hat{y}_{t|t-1} = \ell_{t-1}$ gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Observation equation	$y_t = \ell_{t-1}(1 + \varepsilon_t)$
State equation	$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$

- Models with additive and multiplicative errors with the same parameters generate the same point forecasts but different prediction intervals.

Holt's linear method

ETS(A,A,N)

$$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$$
$$b_t = b_{t-1} + \beta\varepsilon_t$$

ETS(M,A,N)

$$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$$
$$\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$$
$$b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$$

Holt's linear method

ETS(A,A,N)

$$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$$
$$b_t = b_{t-1} + \beta\varepsilon_t$$

ETS(M,A,N)

$$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$$
$$\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$$
$$b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$$

ETS(A,A,A)

Holt-Winters additive method with additive errors.

Forecast equation	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t-m+h_m^+}$
Observation equation	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$
State equations	$\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$
	$b_t = b_{t-1} + \beta\varepsilon_t$
	$s_t = s_{t-m} + \gamma\varepsilon_t$

- Forecast errors: $\varepsilon_t = y_t - \hat{y}_{t|t-1}$
- $h_m^+ = \lfloor (h-1) \mod m \rfloor + 1$.

Additive error models

Trend	Seasonal N	Seasonal A	Seasonal M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/\ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
A$_\mathbf{d}$	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$
M	$y_t = \ell_{t-1}b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1}b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t/\ell_{t-1}$	$y_t = \ell_{t-1}b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1}b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t/\ell_{t-1}$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}b_{t-1}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1}b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/(s_{t-m}\ell_{t-1})$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1}b_{t-1})$
M$_\mathbf{d}$	$y_t = \ell_{t-1}b_{t-1}^{\phi} + \varepsilon_t$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi} + \alpha\varepsilon_t$ $b_t = b_{t-1}^{\phi} + \beta\varepsilon_t/\ell_{t-1}$	$y_t = \ell_{t-1}b_{t-1}^{\phi} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi} + \alpha\varepsilon_t$ $b_t = b_{t-1}^{\phi} + \beta\varepsilon_t/\ell_{t-1}$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}b_{t-1}^{\phi}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1}^{\phi} + \beta\varepsilon_t/(s_{t-m}\ell_{t-1})$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1}b_{t-1}^{\phi})$

Multiplicative error models

Trend	Seasonal N	Seasonal A	Seasonal M
N	$y_t = \ell_{t-1}(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}(1+\alpha\varepsilon_t)$	$y_t = (\ell_{t-1}+s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}+\alpha(\ell_{t-1}+s_{t-m})\varepsilon_t$ $s_t = s_{t-m}+\gamma(\ell_{t-1}+s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m}(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}(1+\alpha\varepsilon_t)$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
A	$y_t = (\ell_{t-1}+b_{t-1})(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1}+b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}+\beta(\ell_{t-1}+b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1}+b_{t-1}+s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}+b_{t-1}+\alpha(\ell_{t-1}+b_{t-1}+s_{t-m})\varepsilon_t$ $b_t = b_{t-1}+\beta(\ell_{t-1}+b_{t-1}+s_{t-m})\varepsilon_t$ $s_t = s_{t-m}+\gamma(\ell_{t-1}+b_{t-1}+s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1}+b_{t-1})s_{t-m}(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1}+b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}+\beta(\ell_{t-1}+b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
A$_\mathbf{d}$	$y_t = (\ell_{t-1}+\phi b_{t-1})(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1}+\phi b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = \phi b_{t-1}+\beta(\ell_{t-1}+\phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1}+\phi b_{t-1}+s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}+\phi b_{t-1}+\alpha(\ell_{t-1}+\phi b_{t-1}+s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1}+\beta(\ell_{t-1}+\phi b_{t-1}+s_{t-m})\varepsilon_t$ $s_t = s_{t-m}+\gamma(\ell_{t-1}+\phi b_{t-1}+s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1}+\phi b_{t-1})s_{t-m}(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1}+\phi b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = \phi b_{t-1}+\beta(\ell_{t-1}+\phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
M	$y_t = \ell_{t-1}b_{t-1}(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}b_{t-1}(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}(1+\beta\varepsilon_t)$	$y_t = (\ell_{t-1}b_{t-1}+s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}b_{t-1}+\alpha(\ell_{t-1}b_{t-1}+s_{t-m})\varepsilon_t$ $b_t = b_{t-1}+\beta(\ell_{t-1}b_{t-1}+s_{t-m})\varepsilon_t/\ell_{t-1}$ $s_t = s_{t-m}+\gamma(\ell_{t-1}b_{t-1}+s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}b_{t-1}s_{t-m}(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}b_{t-1}(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}(1+\beta\varepsilon_t)$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
M$_\mathbf{d}$	$y_t = \ell_{t-1}b_{t-1}^{\phi}(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi}(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}^{\phi}(1+\beta\varepsilon_t)$	$y_t = (\ell_{t-1}b_{t-1}^{\phi}+s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi}+\alpha(\ell_{t-1}b_{t-1}^{\phi}+s_{t-m})\varepsilon_t$ $b_t = b_{t-1}^{\phi}+\beta(\ell_{t-1}b_{t-1}^{\phi}+s_{t-m})\varepsilon_t/\ell_{t-1}$ $s_t = s_{t-m}+\gamma(\ell_{t-1}b_{t-1}^{\phi}+s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}b_{t-1}^{\phi}s_{t-m}(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi}(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}^{\phi}(1+\beta\varepsilon_t)$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$

Innovations state space models

Let $\boldsymbol{x}_t = (\ell_t, b_t, s_t, s_{t-1}, \dots, s_{t-m+1})$ and $\varepsilon_t \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$.

$$\begin{aligned} y_t &= \underbrace{h(\boldsymbol{x}_{t-1})}_{\mu_t} + \underbrace{k(\boldsymbol{x}_{t-1})\varepsilon_t}_{e_t} \\ \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}) + g(\boldsymbol{x}_{t-1})\varepsilon_t \end{aligned}$$

Additive errors:

$$k(x) = 1. \qquad y_t = \mu_t + \varepsilon_t.$$

Multiplicative errors:

$$k(\boldsymbol{x}_{t-1}) = \mu_t. \qquad y_t = \mu_t(1 + \varepsilon_t).$$

$\varepsilon_t = (y_t - \mu_t)/\mu_t$ is relative error.

Innovations state space models

- All the methods can be written in this state space form.
- The only difference between the additive error and multiplicative error models is in the observation equation.
- Additive and multiplicative versions give the same point forecasts.

Innovations state space models

- All the methods can be written in this state space form.
- The only difference between the additive error and multiplicative error models is in the observation equation.
- Additive and multiplicative versions give the same point forecasts.

Innovations state space models

- All the methods can be written in this state space form.
- The only difference between the additive error and multiplicative error models is in the observation equation.
- Additive and multiplicative versions give the same point forecasts.

Some unstable models

- Some of the combinations of (Error, Trend, Seasonal) can lead to numerical difficulties; see equations with division by a state.
- These are: ETS(M,M,A), ETS(M,M_d,A), ETS(A,N,M), ETS(A,A,M), ETS(A,A_d,M), ETS(A,M,N), ETS(A,M,A), ETS(A,M,M), ETS(A,M_d,N), ETS(A,M_d,A), and ETS(A,M_d,M).
- Models with multiplicative errors are useful for strictly positive data – but are not numerically stable with data containing zeros or negative values. In that case only the six fully additive models will be applied.

Some unstable models

- Some of the combinations of (Error, Trend, Seasonal) can lead to numerical difficulties; see equations with division by a state.
- These are: ETS(M,M,A), ETS(M,M_d,A), ETS(A,N,M), ETS(A,A,M), ETS(A,A_d,M), ETS(A,M,N), ETS(A,M,A), ETS(A,M,M), ETS(A,M_d,N), ETS(A,M_d,A), and ETS(A,M_d,M).
- Models with multiplicative errors are useful for strictly positive data – but are not numerically stable with data containing zeros or negative values. In that case only the six fully additive models will be applied.

Some unstable models

- Some of the combinations of (Error, Trend, Seasonal) can lead to numerical difficulties; see equations with division by a state.
- These are: ETS(M,M,A), ETS(M,M_d,A), ETS(A,N,M), ETS(A,A,M), ETS(A,A_d,M), ETS(A,M,N), ETS(A,M,A), ETS(A,M,M), ETS(A,M_d,N), ETS(A,M_d,A), and ETS(A,M_d,M).
- Models with multiplicative errors are useful for strictly positive data – but are not numerically stable with data containing zeros or negative values. In that case only the six fully additive models will be applied.

Exponential smoothing models

Additive Error		**Seasonal Component**		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	A,N,N	A,N,A	~~A,N,M~~
A	(Additive)	A,A,N	A,A,A	~~A,A,M~~
A_d	(Additive damped)	A,A_d,N	A,A_d,A	~~A,A_d,M~~
M	(Multiplicative)	~~A,M,N~~	~~A,M,A~~	~~A,M,M~~
M_d	(Multiplicative damped)	~~A,M_d,N~~	~~A,M_d,A~~	~~A,M_d,M~~

Multiplicative Error		**Seasonal Component**		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	M,N,N	M,N,A	M,N,M
A	(Additive)	M,A,N	M,A,A	M,A,M
A_d	(Additive damped)	M,A_d,N	M,A_d,A	M,A_d,M
M	(Multiplicative)	M,M,N	~~M,M,A~~	M,M,M
M_d	(Multiplicative damped)	M,M_d,N	~~M,M_d,A~~	M,M_d,M

Estimation

$$L^*(\theta, \boldsymbol{x}_0) = n \log\left(\sum_{t=1}^{n} \varepsilon_t^2 / k^2(\boldsymbol{x}_{t-1}) \right) + 2 \sum_{t=1}^{n} \log |k(\boldsymbol{x}_{t-1})|$$
$$= -2 \log(\text{Likelihood}) + \text{constant}$$

- Estimate parameters $\theta = (\alpha, \beta, \gamma, \phi)$ and initial states $\boldsymbol{x}_0 = (\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1})$ by minimizing L^*.

Innovations state space models

Estimation

$$L^*(\boldsymbol{\theta}, \boldsymbol{x}_0) = n \log\left(\sum_{t=1}^{n} \varepsilon_t^2 / k^2(\boldsymbol{x}_{t-1})\right) + 2\sum_{t=1}^{n} \log|k(\boldsymbol{x}_{t-1})|$$
$$= -2\log(\text{Likelihood}) + \text{constant}$$

- Estimate parameters $\boldsymbol{\theta} = (\alpha, \beta, \gamma, \phi)$ and initial states $\boldsymbol{x}_0 = (\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1})$ by minimizing L^*.

Parameter restrictions

Usual region

- Traditional restrictions in the methods $0 < \alpha, \beta^*, \gamma^*, \phi < 1$ — equations interpreted as weighted averages.
- In models we set $\beta = \alpha\beta^*$ and $\gamma = (1 - \alpha)\gamma^*$ therefore $0 < \alpha < 1$, $0 < \beta < \alpha$ and $0 < \gamma < 1 - \alpha$.
- $0.8 < \phi < 0.98$ — to prevent numerical difficulties.

Admissible region

Parameter restrictions

Usual region

- Traditional restrictions in the methods $0 < \alpha, \beta^*, \gamma^*, \phi < 1$ — equations interpreted as weighted averages.
- In models we set $\beta = \alpha\beta^*$ and $\gamma = (1-\alpha)\gamma^*$ therefore $0 < \alpha < 1$, $0 < \beta < \alpha$ and $0 < \gamma < 1-\alpha$.
- $0.8 < \phi < 0.98$ — to prevent numerical difficulties.

Admissible region

- To prevent observations in the distant past having a continuing effect on current forecasts.

Parameter restrictions

Usual region

- Traditional restrictions in the methods $0 < \alpha, \beta^*, \gamma^*, \phi < 1$ — equations interpreted as weighted averages.
- In models we set $\beta = \alpha\beta^*$ and $\gamma = (1-\alpha)\gamma^*$ therefore $0 < \alpha < 1$, $0 < \beta < \alpha$ and $0 < \gamma < 1 - \alpha$.
- $0.8 < \phi < 0.98$ — to prevent numerical difficulties.

Admissible region

- To prevent observations in the distant past having a continuing effect on current forecasts.

Parameter restrictions

Usual region

- Traditional restrictions in the methods $0 < \alpha, \beta^*, \gamma^*, \phi < 1$ — equations interpreted as weighted averages.
- In models we set $\beta = \alpha\beta^*$ and $\gamma = (1-\alpha)\gamma^*$ therefore $0 < \alpha < 1$, $0 < \beta < \alpha$ and $0 < \gamma < 1-\alpha$.
- $0.8 < \phi < 0.98$ — to prevent numerical difficulties.

Admissible region

- To prevent observations in the distant past having a continuing effect on current forecasts.
- Usually (but not always) less restrictive than the *usual* region.
- For example for ETS(A,N,N): *usual* $0 < \alpha < 1$ — *admissible* is $0 < \alpha < 2$.

Parameter restrictions

Usual region

- Traditional restrictions in the methods $0 < \alpha, \beta^*, \gamma^*, \phi < 1$ — equations interpreted as weighted averages.
- In models we set $\beta = \alpha\beta^*$ and $\gamma = (1 - \alpha)\gamma^*$ therefore $0 < \alpha < 1$, $0 < \beta < \alpha$ and $0 < \gamma < 1 - \alpha$.
- $0.8 < \phi < 0.98$ — to prevent numerical difficulties.

Admissible region

- To prevent observations in the distant past having a continuing effect on current forecasts.
- Usually (but not always) less restrictive than the *usual* region.
- For example for ETS(A,N,N): *usual* $0 < \alpha < 1$ — *admissible* is $0 < \alpha < 2$.

Parameter restrictions

Usual region

- Traditional restrictions in the methods $0 < \alpha, \beta^*, \gamma^*, \phi < 1$ — equations interpreted as weighted averages.
- In models we set $\beta = \alpha\beta^*$ and $\gamma = (1 - \alpha)\gamma^*$ therefore $0 < \alpha < 1$, $0 < \beta < \alpha$ and $0 < \gamma < 1 - \alpha$.
- $0.8 < \phi < 0.98$ — to prevent numerical difficulties.

Admissible region

- To prevent observations in the distant past having a continuing effect on current forecasts.
- Usually (but not always) less restrictive than the *usual* region.
- For example for ETS(A,N,N): *usual* $0 < \alpha < 1$ — *admissible* is $0 < \alpha < 2$.

Parameter restrictions

Usual region

- Traditional restrictions in the methods $0 < \alpha, \beta^*, \gamma^*, \phi < 1$ — equations interpreted as weighted averages.
- In models we set $\beta = \alpha\beta^*$ and $\gamma = (1 - \alpha)\gamma^*$ therefore $0 < \alpha < 1$, $0 < \beta < \alpha$ and $0 < \gamma < 1 - \alpha$.
- $0.8 < \phi < 0.98$ — to prevent numerical difficulties.

Admissible region

- To prevent observations in the distant past having a continuing effect on current forecasts.
- Usually (but not always) less restrictive than the *usual* region.
- For example for ETS(A,N,N): *usual* $0 < \alpha < 1$ — *admissible* is $0 < \alpha < 2$.

Model selection

Akaike's Information Criterion

$$\text{AIC} = -2\log(\text{L}) + 2k$$

where L is the likelihood and k is the number of parameters initial states estimated in the model.

Corrected AIC

$$\text{AIC}_{\text{c}} = \text{AIC} + \frac{2(k+1)(k+2)}{T-k}$$

which is the AIC corrected (for small sample bias).

Bayesian Information Criterion

$$\text{BIC} = \text{AIC} + k(\log(T) - 2).$$

Model selection

Akaike's Information Criterion

$$\text{AIC} = -2\log(\text{L}) + 2k$$

where L is the likelihood and k is the number of parameters initial states estimated in the model.

Corrected AIC

$$\text{AIC}_\text{c} = \text{AIC} + \frac{2(k+1)(k+2)}{T-k}$$

which is the AIC corrected (for small sample bias).

Bayesian Information Criterion

$$\text{BIC} = \text{AIC} + k(\log(T) - 2).$$

Model selection

Akaike's Information Criterion

$$\text{AIC} = -2\log(\text{L}) + 2k$$

where L is the likelihood and k is the number of parameters initial states estimated in the model.

Corrected AIC

$$\text{AIC}_\text{c} = \text{AIC} + \frac{2(k+1)(k+2)}{T-k}$$

which is the AIC corrected (for small sample bias).

Bayesian Information Criterion

$$\text{BIC} = \text{AIC} + k(\log(T) - 2).$$

Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

Method performed very well in M3 competition.

Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

Method performed very well in M3 competition.

Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

Method performed very well in M3 competition.

Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

Method performed very well in M3 competition.

Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

Method performed very well in M3 competition.

Forecasting with ETS models

- Point forecasts obtained by iterating equations for $t = T+1, \dots, T+h$, setting $\varepsilon_t = 0$ for $t > T$.
- Not the same as $\text{E}(y_{t+h} | \boldsymbol{x}_t)$ unless trend and seasonality are both additive.
- Point forecasts for ETS(A,x,y) are identical to ETS(M,x,y) if the parameters are the same.
- Prediction intervals will differ between models with additive and multiplicative methods.
- Exact PI available for many models.
- Otherwise, simulate future sample paths, conditional on last estimate of states, and obtain PI from percentiles of simulated paths.

Forecasting with ETS models

- Point forecasts obtained by iterating equations for $t = T + 1, \dots, T + h$, setting $\varepsilon_t = 0$ for $t > T$.
- Not the same as $\mathrm{E}(y_{t+h}|\boldsymbol{x}_t)$ unless trend and seasonality are both additive.
- Point forecasts for ETS(A,x,y) are identical to ETS(M,x,y) if the parameters are the same.
- Prediction intervals will differ between models with additive and multiplicative methods.
- Exact PI available for many models.
- Otherwise, simulate future sample paths, conditional on last estimate of states, and obtain PI from percentiles of simulated paths.

Forecasting with ETS models

- Point forecasts obtained by iterating equations for $t = T + 1, \dots, T + h$, setting $\varepsilon_t = 0$ for $t > T$.
- Not the same as $\text{E}(y_{t+h}|\boldsymbol{x}_t)$ unless trend and seasonality are both additive.
- Point forecasts for ETS(A,x,y) are identical to ETS(M,x,y) if the parameters are the same.
- Prediction intervals will differ between models with additive and multiplicative methods.
- Exact PI available for many models.
- Otherwise, simulate future sample paths, conditional on last estimate of states, and obtain PI from percentiles of simulated paths.

Forecasting with ETS models

- Point forecasts obtained by iterating equations for $t = T+1, \dots, T+h$, setting $\varepsilon_t = 0$ for $t > T$.
- Not the same as $\mathrm{E}(y_{t+h} | \boldsymbol{x}_t)$ unless trend and seasonality are both additive.
- Point forecasts for ETS(A,x,y) are identical to ETS(M,x,y) if the parameters are the same.
- Prediction intervals will differ between models with additive and multiplicative methods.
- Exact PI available for many models.
- Otherwise, simulate future sample paths, conditional on last estimate of states, and obtain PI from percentiles of simulated paths.

Forecasting with ETS models

- Point forecasts obtained by iterating equations for $t = T+1,\dots,T+h$, setting $\varepsilon_t = 0$ for $t > T$.
- Not the same as $\mathrm{E}(y_{t+h}|\boldsymbol{x}_t)$ unless trend and seasonality are both additive.
- Point forecasts for ETS(A,x,y) are identical to ETS(M,x,y) if the parameters are the same.
- Prediction intervals will differ between models with additive and multiplicative methods.
- Exact PI available for many models.
- Otherwise, simulate future sample paths, conditional on last estimate of states, and obtain PI from percentiles of simulated paths.

Forecasting with ETS models

- Point forecasts obtained by iterating equations for $t = T + 1, \ldots, T + h$, setting $\varepsilon_t = 0$ for $t > T$.
- Not the same as $\mathrm{E}(y_{t+h}|\boldsymbol{x}_t)$ unless trend and seasonality are both additive.
- Point forecasts for ETS(A,x,y) are identical to ETS(M,x,y) if the parameters are the same.
- Prediction intervals will differ between models with additive and multiplicative methods.
- Exact PI available for many models.
- Otherwise, simulate future sample paths, conditional on last estimate of states, and obtain PI from percentiles of simulated paths.

Forecasting with ETS models

Point forecasts: iterate the equations for $t = T+1, T+2, \dots, T+h$ and set all $\varepsilon_t = 0$ for $t > T$.

For example, for ETS(M,A,N):

- $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$
- $y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+1}) = [(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}](1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+2|T} = \ell_T + 2b_T$ and so on.

Identical forecast with Holt's linear method and ETS(A,A,N). So the point forecasts obtained from the method and from the two models that underly the method are identical (assuming the same parameter values are used).

Forecasting with ETS models

Point forecasts: iterate the equations for $t = T+1, T+2, \dots, T+h$ and set all $\varepsilon_t = 0$ for $t > T$.

For example, for ETS(M,A,N):

- $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$
- $y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+1}) = \left[(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}\right](1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+2|T} = \ell_T + 2b_T$ and so on.

Identical forecast with Holt's linear method and ETS(A,A,N). So the point forecasts obtained from the method and from the two models that underly the method are identical (assuming the same parameter values are used).

Forecasting with ETS models

Point forecasts: iterate the equations for $t = T+1, T+2, \dots, T+h$ and set all $\varepsilon_t = 0$ for $t > T$.

For example, for ETS(M,A,N):

- $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$
- $y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+1}) =$ $[(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}]\,(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+2|T} = \ell_T + 2b_T$ and so on.

Identical forecast with Holt's linear method and ETS(A,A,N). So the point forecasts obtained from the method and from the two models that underly the method are identical (assuming the same parameter values are used).

Forecasting with ETS models

Point forecasts: iterate the equations for $t = T+1, T+2, \dots, T+h$ and set all $\varepsilon_t = 0$ for $t > T$.

For example, for ETS(M,A,N):

- $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$
- $y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+1}) = [(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}]\,(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+2|T} = \ell_T + 2b_T$ and so on.

Identical forecast with Holt's linear method and ETS(A,A,N). So the point forecasts obtained from the method and from the two models that underly the method are identical (assuming the same parameter values are used).

Forecasting with ETS models

Point forecasts: iterate the equations for $t = T+1, T+2, \dots, T+h$ and set all $\varepsilon_t = 0$ for $t > T$.
For example, for ETS(M,A,N):

- $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$
- $y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+1}) = \left[(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}\right](1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+2|T} = \ell_T + 2b_T$ and so on.

Identical forecast with Holt's linear method and ETS(A,A,N). So the point forecasts obtained from the method and from the two models that underly the method are identical (assuming the same parameter values are used).

Forecasting with ETS models

Point forecasts: iterate the equations for $t = T+1, T+2, \dots, T+h$ and set all $\varepsilon_t = 0$ for $t > T$.
For example, for ETS(M,A,N):

- $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$
- $y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+1}) =$ $[(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}]\,(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+2|T} = \ell_T + 2b_T$ and so on.

Identical forecast with Holt's linear method and ETS(A,A,N). So the point forecasts obtained from the method and from the two models that underly the method are identical (assuming the same parameter values are used).

Forecasting with ETS models

Point forecasts: iterate the equations for $t = T+1, T+2, \ldots, T+h$ and set all $\varepsilon_t = 0$ for $t > T$.

For example, for ETS(M,A,N):

- $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$
- $y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+1}) = [(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}]\,(1 + \varepsilon_{T+1})$
- Therefore $\hat{y}_{T+2|T} = \ell_T + 2b_T$ and so on.

Identical forecast with Holt's linear method and ETS(A,A,N). So the point forecasts obtained from the method and from the two models that underly the method are identical (assuming the same parameter values are used).

Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

- The prediction intervals will differ between models with additive and multiplicative methods.
- Exact formulae for some models.
- More general to simulate future sample paths, conditional on the last estimate of the states, and to obtain prediction intervals from the percentiles of these simulated future paths.
- Options are available in R using the `forecast` function in the forecast package.

Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

- The prediction intervals will differ between models with additive and multiplicative methods.
- Exact formulae for some models.
- More general to simulate future sample paths, conditional on the last estimate of the states, and to obtain prediction intervals from the percentiles of these simulated future paths.
- Options are available in R using the `forecast` function in the forecast package.

Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

- The prediction intervals will differ between models with additive and multiplicative methods.
- Exact formulae for some models.
- More general to simulate future sample paths, conditional on the last estimate of the states, and to obtain prediction intervals from the percentiles of these simulated future paths.
- Options are available in R using the `forecast` function in the forecast package.

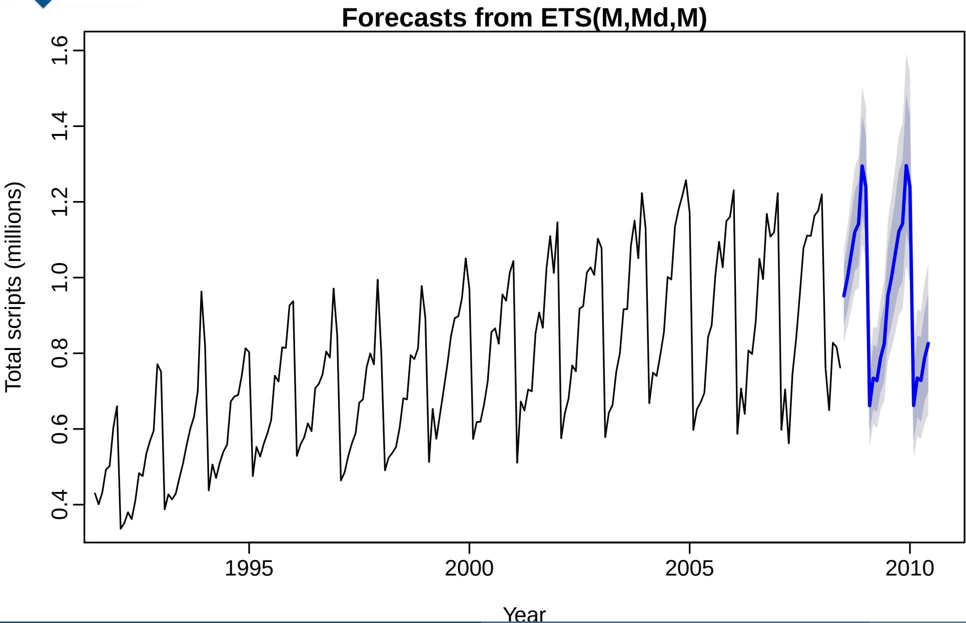
Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

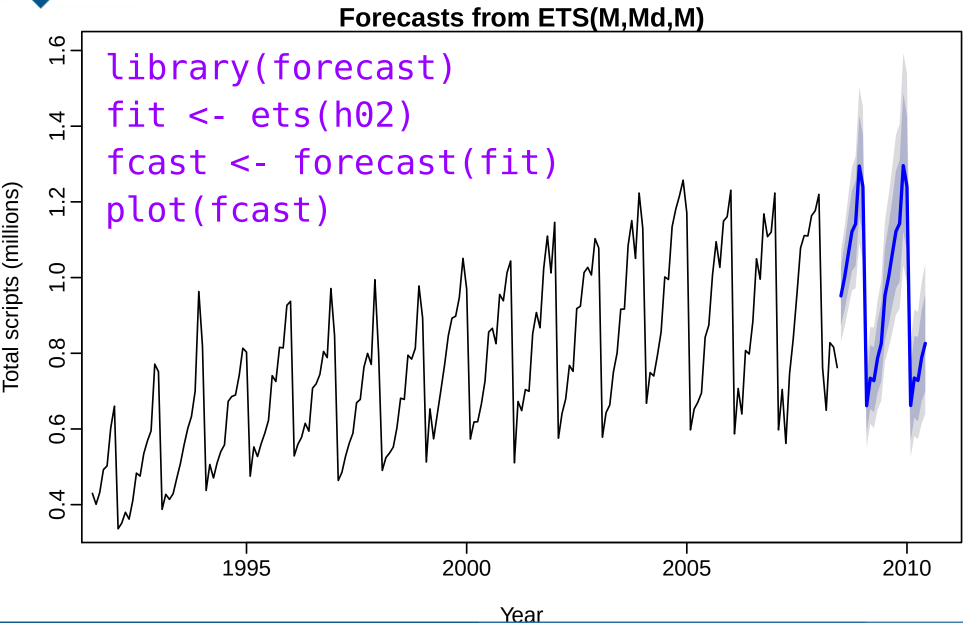
- The prediction intervals will differ between models with additive and multiplicative methods.
- Exact formulae for some models.
- More general to simulate future sample paths, conditional on the last estimate of the states, and to obtain prediction intervals from the percentiles of these simulated future paths.
- Options are available in R using the `forecast` function in the forecast package.

Outline

Exponential smoothing



Exponential smoothing



Exponential smoothing

```
> fit
ETS(M,Md,M)

  Smoothing parameters:
    alpha = 0.3318
    beta  = 4e-04
    gamma = 1e-04
    phi   = 0.9695
  Initial states:
    l = 0.4003
    b = 1.0233
    s = 0.8575 0.8183 0.7559 0.7627 0.6873 1.2884
        1.3456 1.1867 1.1653 1.1033 1.0398 0.9893

  sigma:  0.0651

       AIC       AICc        BIC
-121.97999 -118.68967  -65.57195
```

The `ets()` function in R

```
ets(y, model="ZZZ", damped=NULL,
    alpha=NULL, beta=NULL,
    gamma=NULL, phi=NULL,
    additive.only=FALSE,
    lambda=NULL
    lower=c(rep(0.0001,3),0.80),
    upper=c(rep(0.9999,3),0.98),
    opt.crit=c("lik","amse","mse","sigma"),
    nmse=3,
    bounds=c("both","usual","admissible"),
    ic=c("aic","aicc","bic"), restrict=TRUE)
```

The `ets()` function in R

- `y`
  The time series to be forecast.

- `model`
  use the ETS classification and notation: “N” for none, “A” for additive, “M” for multiplicative, or “Z” for automatic selection. Default `ZZZ` all components are selected using the information criterion.

- `damped`
  - If damped=TRUE, then a damped trend will be used (either A_d or M_d).
  - damped=FALSE, then a non-damped trend will used.
  - If damped=NULL (the default), then either a damped or a non-damped trend will be selected according to the information criterion chosen.

The `ets()` function in R

- `y`
  The time series to be forecast.
- `model`
  use the ETS classification and notation: “N” for none, “A” for additive, “M” for multiplicative, or “Z” for automatic selection. Default `ZZZ` all components are selected using the information criterion.
- `damped`
  - If `damped=TRUE`, then a damped trend will be used (either A_d or M_d).
  - `damped=FALSE`, then a non-damped trend will used.

The `ets()` function in R

- `y`
  The time series to be forecast.
- `model`
  use the ETS classification and notation: "N" for none, "A" for additive, "M" for multiplicative, or "Z" for automatic selection. Default `ZZZ` all components are selected using the information criterion.
- `damped`
  - If `damped=TRUE`, then a damped trend will be used (either A_d or M_d).
  - `damped=FALSE`, then a non-damped trend will used.
  - If `damped=NULL` (the default), then either a damped or a non-damped trend will be selected according to the information criterion chosen.

The `ets()` function in R

- `y`
  The time series to be forecast.
- `model`
  use the ETS classification and notation: “N” for none, “A” for additive, “M” for multiplicative, or “Z” for automatic selection. Default `ZZZ` all components are selected using the information criterion.
- `damped`
  - If `damped=TRUE`, then a damped trend will be used (either A_d or M_d).
  - `damped=FALSE`, then a non-damped trend will used.
  - If `damped=NULL` (the default), then either a damped or a non-damped trend will be selected according to the information criterion chosen.

The `ets()` function in R

- `y`
  The time series to be forecast.
- `model`
  use the ETS classification and notation: “N” for none, “A” for additive, “M” for multiplicative, or “Z” for automatic selection. Default `ZZZ` all components are selected using the information criterion.
- `damped`
  - If `damped=TRUE`, then a damped trend will be used (either A_d or M_d).
  - `damped=FALSE`, then a non-damped trend will used.
  - If `damped=NULL` (the default), then either a damped or a non-damped trend will be selected according to the information criterion chosen.

The `ets()` function in R

- `y`
  The time series to be forecast.
- `model`
  use the ETS classification and notation: "N" for none, "A" for additive, "M" for multiplicative, or "Z" for automatic selection. Default `ZZZ` all components are selected using the information criterion.
- `damped`
  - If `damped=TRUE`, then a damped trend will be used (either A_d or M_d).
  - `damped=FALSE`, then a non-damped trend will used.
  - If `damped=NULL` (the default), then either a damped or a non-damped trend will be selected according to the information criterion chosen.

The `ets()` function in R

- `alpha, beta, gamma, phi`
  The values of the smoothing parameters can be specified using these arguments. If they are set to `NULL` (the default value for each of them), the parameters are estimated.
- `additive.only`
  Only models with additive components will be considered if `additive.only=TRUE`. Otherwise all models will be considered.
- `lambda`
  Box-Cox transformation parameter. It will be ignored if `lambda=NULL` (the default value). Otherwise, the time series will be transformed before the model is estimated. When `lambda` is not `NULL`, `additive.only` is set to `TRUE`.

The `ets()` function in R

- `alpha, beta, gamma, phi`
  The values of the smoothing parameters can be specified using these arguments. If they are set to `NULL` (the default value for each of them), the parameters are estimated.
- `additive.only`
  Only models with additive components will be considered if `additive.only=TRUE`. Otherwise all models will be considered.
- `lambda`
  Box-Cox transformation parameter. It will be ignored if `lambda=NULL` (the default value). Otherwise, the time series will be transformed before the model is estimated. When `lambda` is not `NULL`, `additive.only` is set to `TRUE`.

The `ets()` function in R

- `alpha, beta, gamma, phi`
  The values of the smoothing parameters can be specified using these arguments. If they are set to `NULL` (the default value for each of them), the parameters are estimated.
- `additive.only`
  Only models with additive components will be considered if `additive.only=TRUE`. Otherwise all models will be considered.
- `lambda`
  Box-Cox transformation parameter. It will be ignored if `lambda=NULL` (the default value). Otherwise, the time series will be transformed before the model is estimated. When `lambda` is not `NULL`, `additive.only` is set to `TRUE`.

The `ets()` function in R

- `lower,upper` bounds for the parameter estimates of α, β, γ and ϕ.
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- `bounds` Constraints on the parameters.
  - usual region – "bounds=usual";
  - admissible region – "bounds=admissible";
  - "bounds=both" (the default) requires the parameters to satisfy both sets of constraints.
- `ic=aic` (the default) information criterion to be used in selecting models.
- `restrict=TRUE` (the default) models that cause numerical difficulties are not considered in model selection.

The `ets()` function in R

- `lower,upper` bounds for the parameter estimates of α, β, γ and ϕ.
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- `bounds` Constraints on the parameters.
  - *usual* region – "`bounds=usual`";
  - *admissible* region – "`bounds=admissible`";
  - "`bounds=both`" (the default) requires the parameters to satisfy both sets of constraints.
- `ic=aic` (the default) information criterion to be used in selecting models.
- `restrict=TRUE` (the default) models that cause numerical difficulties are not considered in model selection.

The `ets()` function in R

- `lower,upper` bounds for the parameter estimates of α, β, γ and ϕ.
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- `bounds` Constraints on the parameters.
  - *usual* region – `"bounds=usual"`;
  - *admissible* region – `"bounds=admissible"`;
  - `"bounds=both"` (the default) requires the parameters to satisfy both sets of constraints.
- `ic=aic` (the default) information criterion to be used in selecting models.
- `restrict=TRUE` (the default) models that cause numerical difficulties are not considered in model selection.

The `ets()` function in R

- `lower,upper` bounds for the parameter estimates of α, β, γ and ϕ.
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- `bounds` Constraints on the parameters.
  - *usual* region – `"bounds=usual"`;
  - *admissible* region – `"bounds=admissible"`;
  - `"bounds=both"` (the default) requires the parameters to satisfy both sets of constraints.
- `ic=aic` (the default) information criterion to be used in selecting models.
- `restrict=TRUE` (the default) models that cause numerical difficulties are not considered in model selection.

The `ets()` function in R

- `lower,upper` bounds for the parameter estimates of α, β, γ and ϕ.
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- `bounds` Constraints on the parameters.
  - *usual* region – `"bounds=usual"`;
  - *admissible* region – `"bounds=admissible"`;
  - `"bounds=both"` (the default) requires the parameters to satisfy both sets of constraints.
- `ic=aic` (the default) information criterion to be used in selecting models.
- `restrict=TRUE` (the default) models that cause numerical difficulties are not considered in model selection.

The `ets()` function in R

- `lower,upper` bounds for the parameter estimates of α, β, γ and ϕ.
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- `bounds` Constraints on the parameters.
  - *usual* region – `"bounds=usual"`;
  - *admissible* region – `"bounds=admissible"`;
  - `"bounds=both"` (the default) requires the parameters to satisfy both sets of constraints.
- `ic=aic` (the default) information criterion to be used in selecting models.
- `restrict=TRUE` (the default) models that cause numerical difficulties are not considered in model selection.

The `ets()` function in R

- `lower,upper` bounds for the parameter estimates of α, β, γ and ϕ.
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- `bounds` Constraints on the parameters.
  - *usual* region – `"bounds=usual"`;
  - *admissible* region – `"bounds=admissible"`;
  - `"bounds=both"` (the default) requires the parameters to satisfy both sets of constraints.
- `ic=aic` (the default) information criterion to be used in selecting models.
- `restrict=TRUE` (the default) models that cause numerical difficulties are not considered in model selection.

The `ets()` function in R

- `lower,upper` bounds for the parameter estimates of α, β, γ and ϕ.
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- `bounds` Constraints on the parameters.
  - *usual* region – `"bounds=usual"`;
  - *admissible* region – `"bounds=admissible"`;
  - `"bounds=both"` (the default) requires the parameters to satisfy both sets of constraints.
- `ic=aic` (the default) information criterion to be used in selecting models.
- `restrict=TRUE` (the default) models that cause numerical difficulties are not considered in model selection.

The `forecast()` function in R

```
forecast(object,
  h=ifelse(object$m>1, 2*object$m, 10),
  level=c(80,95), fan=FALSE,
  simulate=FALSE, bootstrap=FALSE,
  npaths=5000, PI=TRUE, lambda=object$lambda, ...)
```

- `object`: the object returned by the `ets()` function.
- `h`: the number of periods to be forecast.
- `level`: the confidence level for the prediction intervals.
- `fan`: if `fan=TRUE`, suitable for fan plots.
- `simulate`

The `forecast()` function in R

```
forecast(object,
  h=ifelse(object$m>1, 2*object$m, 10),
  level=c(80,95), fan=FALSE,
  simulate=FALSE, bootstrap=FALSE,
  npaths=5000, PI=TRUE, lambda=object$lambda, ...)
```

- `object`: the object returned by the `ets()` function.
- `h`: the number of periods to be forecast.
- `level`: the confidence level for the prediction intervals.
- `fan`: if `fan=TRUE`, suitable for fan plots.
- `simulate`

The `forecast()` function in R

```
forecast(object,
  h=ifelse(object$m>1, 2*object$m, 10),
  level=c(80,95), fan=FALSE,
  simulate=FALSE, bootstrap=FALSE,
  npaths=5000, PI=TRUE, lambda=object$lambda, ...)
```

- `object`: the object returned by the `ets()` function.
- `h`: the number of periods to be forecast.
- `level`: the confidence level for the prediction intervals.
- `fan`: if `fan=TRUE`, suitable for fan plots.
- `simulate`
  - If `simulate=TRUE`, prediction intervals generated via simulation rather than analytic formulae.
  - Even if `simulate=FALSE` simulation will be used if there are no algebraic formulae exist.

The `forecast()` function in R

```
forecast(object,
  h=ifelse(object$m>1, 2*object$m, 10),
  level=c(80,95), fan=FALSE,
  simulate=FALSE, bootstrap=FALSE,
  npaths=5000, PI=TRUE, lambda=object$lambda, ...)
```

- `object`: the object returned by the `ets()` function.
- `h`: the number of periods to be forecast.
- `level`: the confidence level for the prediction intervals.
- `fan`: if `fan=TRUE`, suitable for fan plots.
- `simulate`
  - If `simulate=TRUE`, prediction intervals generated via simulation rather than analytic formulae.
  - Even if `simulate=FALSE` simulation will be used if there are no algebraic formulae exist.

The `forecast()` function in R

```
forecast(object,
  h=ifelse(object$m>1, 2*object$m, 10),
  level=c(80,95), fan=FALSE,
  simulate=FALSE, bootstrap=FALSE,
  npaths=5000, PI=TRUE, lambda=object$lambda, ...
```

- `object`: the object returned by the `ets()` function.
- `h`: the number of periods to be forecast.
- `level`: the confidence level for the prediction intervals.
- `fan`: if `fan=TRUE`, suitable for fan plots.
- `simulate`
  - If `simulate=TRUE`, prediction intervals generated via simulation rather than analytic formulae.
  - Even if `simulate=FALSE` simulation will be used if there are no algebraic formulae exist.

The `forecast()` function in R

```
forecast(object,
  h=ifelse(object$m>1, 2*object$m, 10),
  level=c(80,95), fan=FALSE,
  simulate=FALSE, bootstrap=FALSE,
  npaths=5000, PI=TRUE, lambda=object$lambda, ...
```

- `object`: the object returned by the `ets()` function.
- `h`: the number of periods to be forecast.
- `level`: the confidence level for the prediction intervals.
- `fan`: if `fan=TRUE`, suitable for fan plots.
- `simulate`
  - If `simulate=TRUE`, prediction intervals generated via simulation rather than analytic formulae.
  - Even if `simulate=FALSE` simulation will be used if there are no algebraic formulae exist.

The `forecast()` function in R

```
forecast(object,
  h=ifelse(object$m>1, 2*object$m, 10),
  level=c(80,95), fan=FALSE,
  simulate=FALSE, bootstrap=FALSE,
  npaths=5000, PI=TRUE, lambda=object$lambda, ...
```

- `object`: the object returned by the `ets()` function.
- `h`: the number of periods to be forecast.
- `level`: the confidence level for the prediction intervals.
- `fan`: if `fan=TRUE`, suitable for fan plots.
- `simulate`
  - If `simulate=TRUE`, prediction intervals generated via simulation rather than analytic formulae.
  - Even if `simulate=FALSE` simulation will be used if there are no algebraic formulae exist.

The `forecast()` function in R

- `bootstrap`: If `bootstrap=TRUE` and `simulate=TRUE`, then the simulated prediction intervals use re-sampled errors rather than normally distributed errors.
- `npaths`: The number of sample paths used in computing simulated prediction intervals.
- `PI`: If `PI=TRUE`, then prediction intervals are produced; otherwise only point forecasts are calculated. If `PI=FALSE`, then `level`, `fan`, `simulate`, `bootstrap` and `npaths` are all ignored.
- `lambda`: The Box-Cox transformation parameter. This is ignored if `lambda=NULL`. Otherwise, forecasts are back-transformed via an inverse Box-Cox transformation.

The `forecast()` function in R

- `bootstrap`: If `bootstrap=TRUE` and `simulate=TRUE`, then the simulated prediction intervals use re-sampled errors rather than normally distributed errors.
- `npaths`: The number of sample paths used in computing simulated prediction intervals.
- `PI`: If `PI=TRUE`, then prediction intervals are produced; otherwise only point forecasts are calculated. If `PI=FALSE`, then `level`, `fan`, `simulate`, `bootstrap` and `npaths` are all ignored.
- `lambda`: The Box-Cox transformation parameter. This is ignored if `lambda=NULL`. Otherwise, forecasts are back-transformed via an inverse Box-Cox transformation.

The `forecast()` function in R

- `bootstrap`: If `bootstrap=TRUE` and `simulate=TRUE`, then the simulated prediction intervals use re-sampled errors rather than normally distributed errors.
- `npaths`: The number of sample paths used in computing simulated prediction intervals.
- `PI`: If `PI=TRUE`, then prediction intervals are produced; otherwise only point forecasts are calculated. If `PI=FALSE`, then `level`, `fan`, `simulate`, `bootstrap` and `npaths` are all ignored.
- `lambda`: The Box-Cox transformation parameter. This is ignored if `lambda=NULL`. Otherwise, forecasts are back-transformed via an inverse Box-Cox transformation.

The `forecast()` function in R

- `bootstrap`: If `bootstrap=TRUE` and `simulate=TRUE`, then the simulated prediction intervals use re-sampled errors rather than normally distributed errors.
- `npaths`: The number of sample paths used in computing simulated prediction intervals.
- `PI`: If `PI=TRUE`, then prediction intervals are produced; otherwise only point forecasts are calculated. If `PI=FALSE`, then `level`, `fan`, `simulate`, `bootstrap` and `npaths` are all ignored.
- `lambda`: The Box-Cox transformation parameter. This is ignored if `lambda=NULL`. Otherwise, forecasts are back-transformed via an inverse Box-Cox transformation.