ECON 427: ECONOMIC FORECASTING

Ch7. Exponential smoothing

OTexts.org/fpp2/

Outline

Simple methods

Time series $y_1, y_2, \dots, y_T$.

Random walk forecasts

$$\hat{y}_{T+h|T} = y_T$$

Simple methods

Time series $y_1, y_2, \ldots, y_T$.

Random walk forecasts

$$\hat{y}_{T+h|T} = y_T$$

Average forecasts

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

Simple methods

Time series $y_1, y_2, \dots, y_T$.

Random walk forecasts

$$\hat{y}_{T+h|T} = y_T$$

Average forecasts

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

- Want something in between that weights most recent data more highly.
- Simple exponential smoothing uses a weighted moving average with weights that decrease exponentially.

Simple Exponential Smoothing

Forecast equation

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

where $0 \leq \alpha \leq 1$.

Simple Exponential Smoothing

Forecast equation

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

where $0 \leq \alpha \leq 1$.

	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}	$(0.2)(0.8)^5$	$(0.4)(0.6)^5$	$(0.6)(0.4)^5$	$(0.8)(0.2)^5$

Simple Exponential Smoothing

Component form

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

- ℓ_t is the level (or the smoothed value) of the series at time t.
- $\hat{y}_{t+1|t} = \alpha y_t + (1-\alpha)\hat{y}_{t|t-1}$
  Iterate to get exponentially weighted moving average form.

Weighted average form

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1-\alpha)^j y_{T-j} + (1-\alpha)^T \ell_0$$

Optimisation

- Need to choose value for α and ℓ_0
- Similarly to regression — we choose α and ℓ_0 by minimising SSE:

$$\text{SSE} = \sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2.$$

- Unlike regression there is no closed form solution — use numerical optimization.

Example: Oil production

```r
oildata <- window(oil, start=1996)
# Estimate parameters
fc <- ses(oildata, h=5)
summary(fc[["model"]])
```

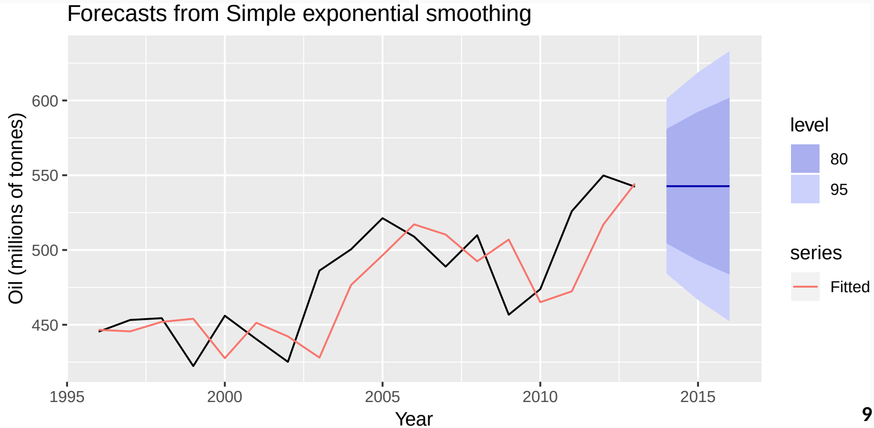
```
## Simple exponential smoothing
##
## Call:
##  ses(y = oildata, h = 5)
##
##   Smoothing parameters:
##     alpha = 0.8339
##
##   Initial states:
##     l = 446.5868
##
##   sigma:  29.83
```

Example: Oil production

Year	Time	Observation	Level	Forecast
	t	y_t	ℓ_t	$\hat{y}_{t+1\|t}$
1995	0		446.59	
1996	1	445.36	445.57	446.59
1997	2	453.20	451.93	445.57
1998	3	454.41	454.00	451.93
1999	4	422.38	427.63	454.00
2000	5	456.04	451.32	427.63
2001	6	440.39	442.20	451.32
2002	7	425.19	428.02	442.20
2003	8	486.21	476.54	428.02
2004	9	500.43	496.46	476.54
2005	10	521.28	517.15	496.46
2006	11	508.95	510.31	517.15
2007	12	488.89	492.45	510.31
2008	13	509.87	506.98	492.45
2009	14	456.72	465.07	506.98
2010	15	473.82	472.36	465.07
2011	16	525.95	517.05	472.36
2012	17	549.83	544.39	517.05
2013	18	542.34	542.68	544.39
	h			$\hat{y}_{T+h\|T}$
2014	1			542.68
2015	2			542.68

Example: Oil production

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



Outline

Holt's linear trend

Component form

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

Holt's linear trend

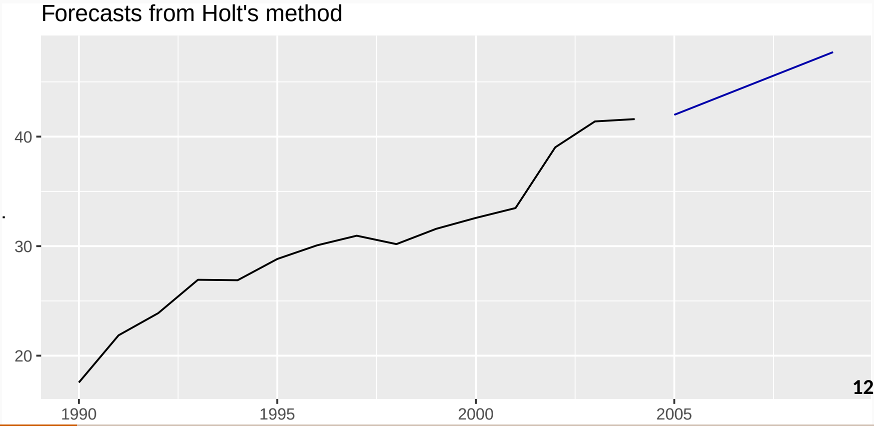
Component form

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

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

Holt's method in R

```r
window(ausair, start=1990, end=2004) %>%
  holt(h=5, PI=FALSE) %>%
  autoplot()
```



Damped trend method

Component form

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

Damped trend method

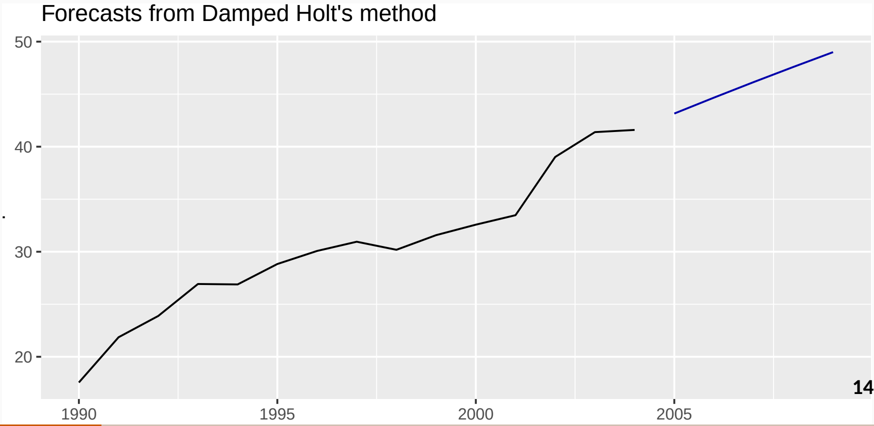
Component form

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

- 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.

Example: Air passengers

```
window(ausair, start=1990, end=2004) %>%
  holt(damped=TRUE, h=5, PI=FALSE) %>%
  autoplot()
```



Example: Sheep in Asia

```
livestock2 <- window(livestock, start=1970,
                                end=2000)
fit1 <- ses(livestock2)
fit2 <- holt(livestock2)
fit3 <- holt(livestock2, damped = TRUE)
```

```
accuracy(fit1, livestock)
accuracy(fit2, livestock)
accuracy(fit3, livestock)
```

Example: Sheep in Asia

	SES	Linear trend	Damped trend
α	1.00	0.98	0.97
β^*		0.00	0.00
ϕ			0.98
ℓ_0	263.90	251.46	251.89
b_0		4.99	6.29
Training RMSE	14.77	13.98	14.00
Test RMSE	25.46	11.88	14.73
Test MAE	20.38	10.71	13.30
Test MAPE	4.60	2.54	3.07
Test MASE	2.26	1.19	1.48

Your turn

`eggs` contains the price of a dozen eggs in the United States from 1900–1993

1. Use SES and Holt's method (with and without damping) to forecast "future" data.
   [Hint: use h=100 so you can clearly see the differences between the options when plotting the forecasts.]
2. Which method gives the best training RMSE?
3. Are these RMSE values comparable?
4. Do the residuals from the best fitting method look like white noise?

Outline

Holt-Winters additive method

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

Component form

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

- k = integer part of $(h-1)/m$. 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

- Seasonal component is usually expressed as $s_t = \gamma^*(y_t - \ell_t) + (1 - \gamma^*)s_{t-m}.$
- Substitute in for ℓ_t: $s_t = \gamma^*(1-\alpha)(y_t-\ell_{t-1}-b_{t-1})+[1-\gamma^*(1-\alpha)]s_{t-m}$
- We set $\gamma = \gamma^*(1 - \alpha)$.
- The usual parameter restriction is $0 \le \gamma^* \le 1$, which translates to $0 \le \gamma \le (1 - \alpha)$.

Holt-Winters multiplicative method

For when seasonal variations are changing proportional to the level of the series.

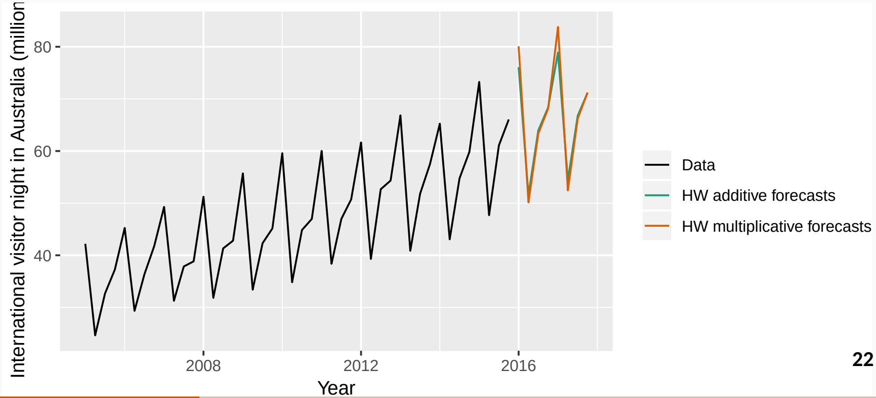
Component form

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

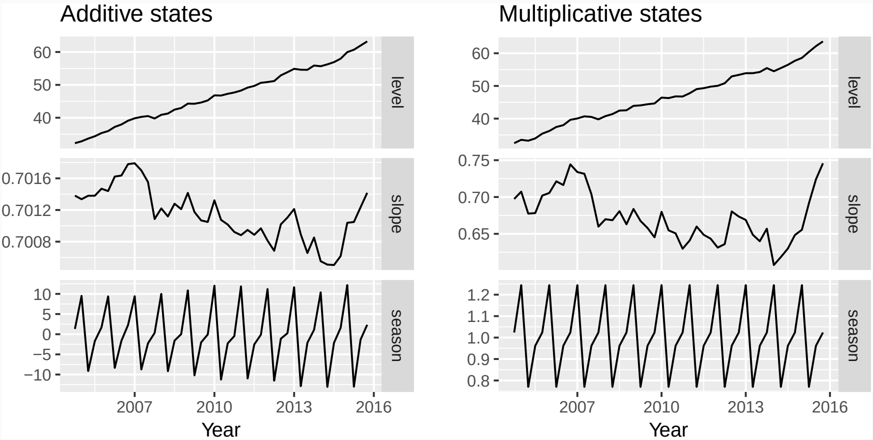
- k is integer part of $(h-1)/m$.
- With additive method s_t is in absolute terms: within each year $\sum_i s_i \approx 0$.
- With multiplicative method s_t is in relative terms: within each year $\sum_i s_i \approx m$.

Example: Visitor Nights

```
aust <- window(austourists,start=2005)
fit1 <- hw(aust,seasonal="additive")
fit2 <- hw(aust,seasonal="multiplicative")
```



Estimated components



Holt-Winters damped method

Often the single most accurate forecasting method for seasonal data:

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

Your turn

Apply Holt-Winters' multiplicative method to the `gas` data.

1. Why is multiplicative seasonality necessary here?
2. Experiment with making the trend damped.
3. Check that the residuals from the best method look like white noise.

Outline

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	Seasonal Component A (Additive)	Seasonal Component 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)

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

Recursive formulae

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

R functions

- Simple exponential smoothing: no trend.

  ```
  ses(y)
  ```
- Holt's method: linear trend.

  ```
  holt(y)
  ```
- Damped trend method.

  ```
  holt(y, damped=TRUE)
  ```
- Holt-Winters methods

  ```
  hw(y, damped=TRUE, seasonal="additive")
  hw(y, damped=FALSE, seasonal="additive")
  hw(y, damped=TRUE, seasonal="multiplicative")
  hw(y, damped=FALSE, seasonal="multiplicative")
  ```
- Combination of no trend with seasonality not possible using these functions.

Outline

Methods v Models

Exponential smoothing methods

- Algorithms that return point forecasts.

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 18 models.
- ETS(Error,Trend,Seasonal):
  - Error = $\{A, M\}$
  - Trend = $\{N, A, A_d\}$
  - Seasonal = $\{N, A, M\}$.

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

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

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

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

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

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

$\nearrow$ $\uparrow$ $\nwarrow$

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

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

There are 18 separate models in the ETS framework

A model for SES

Component form

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

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

A model for SES

Component form

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

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

Forecast error: $e_t = y_t - \hat{y}_{t|t-1} = y_t - \ell_{t-1}$.

A model for SES

Component form

$$\begin{aligned} \text{Forecast equation} \quad & \hat{y}_{t+h|t} = \ell_t \\ \text{Smoothing equation} \quad & \ell_t = \alpha y_t + (1-\alpha)\ell_{t-1} \end{aligned}$$

Forecast error: $e_t = y_t - \hat{y}_{t|t-1} = y_t - \ell_{t-1}$.

Error correction form

$$\begin{aligned} y_t &= \ell_{t-1} + e_t \\ \ell_t &= \ell_{t-1} + \alpha(y_t - \ell_{t-1}) \\ &= \ell_{t-1} + \alpha e_t \end{aligned}$$

A model for SES

Component form

$$
\begin{aligned}
\text{Forecast equation} \quad & \hat{y}_{t+h|t} = \ell_t \\
\text{Smoothing equation} \quad & \ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}
\end{aligned}
$$

Forecast error: $e_t = y_t - \hat{y}_{t|t-1} = y_t - \ell_{t-1}$.

Error correction form

$$
\begin{aligned}
y_t &= \ell_{t-1} + e_t \\
\ell_t &= \ell_{t-1} + \alpha(y_t - \ell_{t-1}) \\
&= \ell_{t-1} + \alpha e_t
\end{aligned}
$$

Specify probability distribution for e_t, we assume $e_t = \varepsilon_t \sim \text{NID}(0, \sigma^2)$.

ETS(A,N,N)

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

where $\varepsilon_t \sim \text{NID}(0, \sigma^2)$.

- "innovations" or "single source of error" because same error process, ε_t.
- Measurement equation: relationship between observations and states.
- Transition equation(s): evolution of the state(s) through time.

ETS(A,A,N)

Holt's linear method with additive errors.

- Assume $\varepsilon_t = y_t - \ell_{t-1} - b_{t-1} \sim \text{NID}(0, \sigma^2)$.
- Substituting into the error correction equations for Holt's linear method

$$
\begin{aligned}
y_t &= \ell_{t-1} + b_{t-1} + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t \\
b_t &= b_{t-1} + \alpha\beta^*\varepsilon_t
\end{aligned}
$$

- For simplicity, set $\beta = \alpha\beta^*$.

Your turn

- Write down the model for ETS(A,Ad,N)

ETS(A,A,A)

Holt-Winters additive method with additive errors.

$$
\begin{aligned}
\text{Forecast equation} \quad \hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\
\text{Observation equation} \quad y_t &= \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t \\
\text{State equations} \quad \ell_t &= \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t \\
b_t &= b_{t-1} + \beta\varepsilon_t \\
s_t &= s_{t-m} + \gamma\varepsilon_t
\end{aligned}
$$

- Forecast errors: $\varepsilon_t = y_t - \hat{y}_{t|t-1}$
- k is integer part of $(h-1)/m$.

Your turn

- Write down the model for ETS(A,N,A)

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:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

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:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Measurement 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:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Measurement 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,A,N)

Holt's linear method with multiplicative errors.

- Assume $\varepsilon_t = \frac{y_t - (\ell_{t-1} + b_{t-1})}{(\ell_{t-1} + b_{t-1})}$
- Following a similar approach as above, the innovations state space model underlying Holt's linear method with multiplicative errors is specified as

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

where again $\beta = \alpha\beta^*$ and $\varepsilon_t \sim \text{NID}(0, \sigma^2)$.

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	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
$\mathbf{A_d}$	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$

Multiplicative error models

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

Estimating ETS models

- Smoothing parameters α, β, γ and ϕ, and the initial states $\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1}$ are estimated by maximising the "likelihood" = the probability of the data arising from the specified model.
- For models with additive errors equivalent to minimising SSE.
- For models with multiplicative errors, **not** equivalent to minimising SSE.
- We will estimate models with the `ets()` function in the forecast package.

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

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 $0 < \alpha < 1, \quad 0 < \beta < \alpha \quad$ and $0 < \gamma < 1 - \alpha$.
- $0.8 < \phi < 0.98$ — to prevent numerical difficulties.

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 $0 < \alpha < 1, \quad 0 < \beta < \alpha \quad$ 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 *traditional* region.
- For example for ETS(A,N,N): *traditional* $0 < \alpha < 1$ — *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.

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).

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

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

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 AICc:
- Produce forecasts using best method.
- Obtain forecast intervals using underlying state space model.

Method performed very well in M3 competition.

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(A,N,M), ETS(A,A,M), ETS(A,A_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~~

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

Example: International tourists

```r
aust <- window(austourists, start=2005)
fit <- ets(aust)
summary(fit)
```

```
## ETS(M,A,M)
##
## Call:
##  ets(y = aust)
##
##   Smoothing parameters:
##     alpha = 0.1908
##     beta  = 0.0392
##     gamma = 2e-04
##
##   Initial states:
##     l = 32.3679
##     b = 0.9281
##     s = 1.022 0.9628 0.7683 1.247
##
##   sigma:  0.0383
##
##   AIC  AICc   BIC
```

Example: International tourists

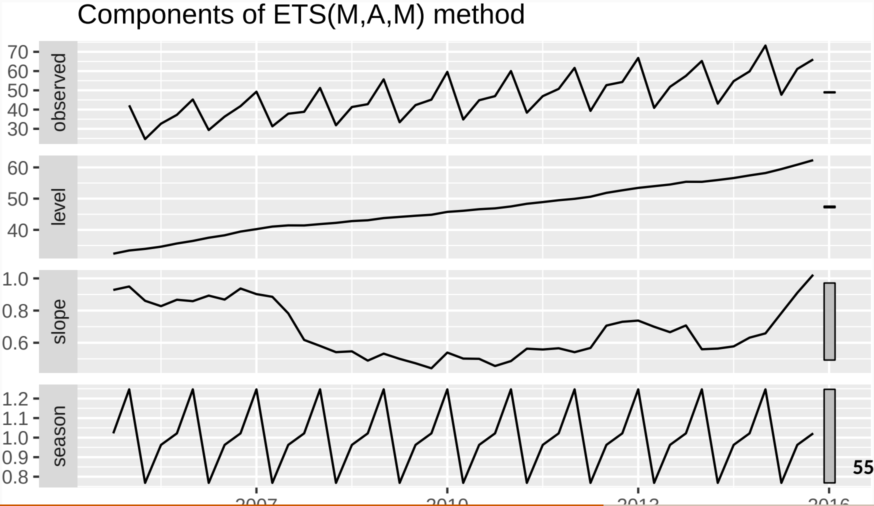
Model selected: ETS(M,A,M)

$$
\begin{aligned}
y_t &= (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)\\
\ell_t &= (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)\\
b_t &= b_{t-1} + \beta(\ell_{t-1} + b_{t_1})\varepsilon_t\\
s_t &= s_{t-m}(1 + \gamma\varepsilon_t).
\end{aligned}
$$

$\hat{\alpha} = 0.1908$, $\hat{\beta} = 0.0392$, and $\hat{\gamma} = 0.00019$.

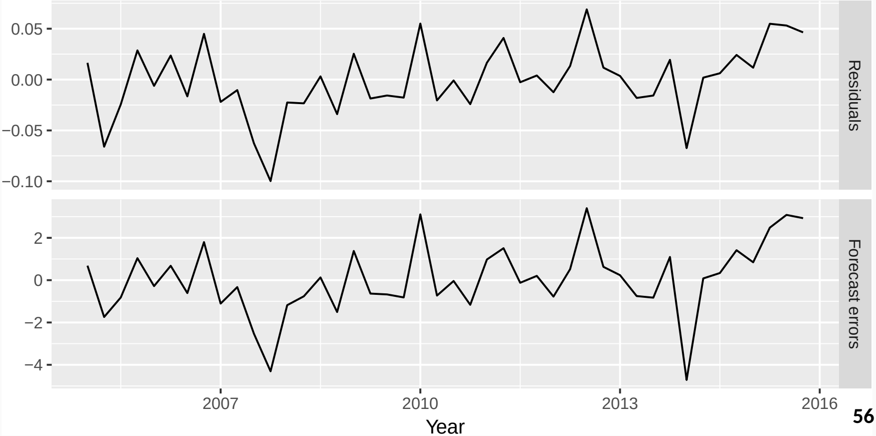
Example: International tourists

```
autoplot(fit)
```



Example: International tourists

```r
cbind('Residuals' = residuals(fit),
      'Forecast errors' = residuals(fit, type='response')) %>%
  autoplot(facet=TRUE) + xlab("Year") + ylab("")
```



Residuals

Response residuals

$$\hat{e}_t = y_t - \hat{y}_{t|t-1}$$

Innovation residuals

Additive error model:

$$\hat{\varepsilon}_t = y_t - \hat{y}_{t|t-1}$$

Multiplicative error model:

$$\hat{\varepsilon}_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}}$$

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 $t > T$.

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 $t > T$.

- 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.

Example: ETS(A,A,N)

$$
\begin{aligned}
y_{T+1} &= \ell_T + b_T + \varepsilon_{T+1} \\
\hat{y}_{T+1|T} &= \ell_T + b_T \\
y_{T+2} &= \ell_{T+1} + b_{T+1} + \varepsilon_{T+2} \\
&= (\ell_T + b_T + \alpha\varepsilon_{T+1}) + (b_T + \beta\varepsilon_{T+1}) + \varepsilon_{T+2} \\
\hat{y}_{T+2|T} &= \ell_T + 2b_T
\end{aligned}
$$

etc.

Example: ETS(M,A,N)

$$
\begin{aligned}
y_{T+1} &= (\ell_T + b_T)(1 + \varepsilon_{T+1}) \\
\hat{y}_{T+1|T} &= \ell_T + b_T. \\
y_{T+2} &= (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+2}) \\
&= \left\{(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + [b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}]\right\}(1 + \varepsilon_{T+2}) \\
\hat{y}_{T+2|T} &= \ell_T + 2b_T
\end{aligned}
$$

etc.

Forecasting with ETS models

Prediction intervals: cannot be generated using the methods, only the models.

- The prediction intervals will differ between models with additive and multiplicative errors.
- 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.

Prediction intervals

PI for most ETS models: $\hat{y}_{T+h|T} \pm c\sigma_h$, where c depends on coverage probability and σ_h is forecast standard deviation.

(A,N,N)	$\sigma_h = \sigma^2\Big[1 + \alpha^2(h-1)\Big]$
(A,A,N)	$\sigma_h = \sigma^2\Big[1 + (h-1)\{\alpha^2 + \alpha\beta h + \frac{1}{6}\beta^2 h(2h-1)\}\Big]$
(A,A$_d$,N)	$\sigma_h = \sigma^2\Big[1 + \alpha^2(h-1) + \frac{\beta\phi h}{(1-\phi)^2}\left\{2\alpha(1-\phi) + \beta\phi\right\} - \frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)}\left\{2\alpha(1-\phi^2) + \beta\phi(1+2\phi-\phi^h)\right\}\Big]$
(A,N,A)	$\sigma_h = \sigma^2\Big[1 + \alpha^2(h-1) + \gamma k(2\alpha+\gamma)\Big]$
(A,A,A)	$\sigma_h = \sigma^2\Big[1 + (h-1)\{\alpha^2 + \alpha\beta h + \frac{1}{6}\beta^2 h(2h-1)\} + \gamma k\{2\alpha + \gamma + \beta m(k + 1$
(A,A$_d$,A)	$\sigma_h = \sigma^2\Big[1 + \alpha^2(h-1) + \frac{\beta\phi h}{(1-\phi)^2}\left\{2\alpha(1-\phi) + \beta\phi\right\} - \frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)}\left\{2\alpha(1-\phi^2) + \beta\phi(1+2\phi-\phi^h)\right\} + \gamma k(2\alpha+\gamma) + \frac{2\beta\gamma\phi}{(1-\phi)(1-\phi^m)}\left\{k(1-\phi^m) - \phi^m(1-\phi^{mk})\right\}\Big]$

Outline

Example: drug sales

```r
ets(h02)
```

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = h02)
##
##   Smoothing parameters:
##     alpha = 0.1953
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9798
##
##   Initial states:
##     l = 0.3945
##     b = 0.0085
##     s = 0.874 0.8197 0.7644 0.7693 0.6941 1.284
##            1.326 1.177 1.162 1.095 1.042 0.9924
##
##   sigma:  0.0676
##
##     AIC    AICc     BIC
## -122.91 -119.21  -63.18
```

Example: drug sales

```r
ets(h02, model="AAA", damped=FALSE)
```

```
## ETS(A,A,A)
##
## Call:
##  ets(y = h02, model = "AAA", damped = FALSE)
##
##   Smoothing parameters:
##     alpha = 0.1672
##     beta  = 0.0084
##     gamma = 1e-04
##
##   Initial states:
##     l = 0.3895
##     b = 0.0116
##     s = -0.1058 -0.1359 -0.1875 -0.1803 -0.2414 0.2097
##            0.2493 0.1426 0.1411 0.0823 0.0293 -0.0033
##
##   sigma:  0.0642
##
##     AIC    AICc     BIC
## -18.26 -14.97   38.14
```

The `ets()` function

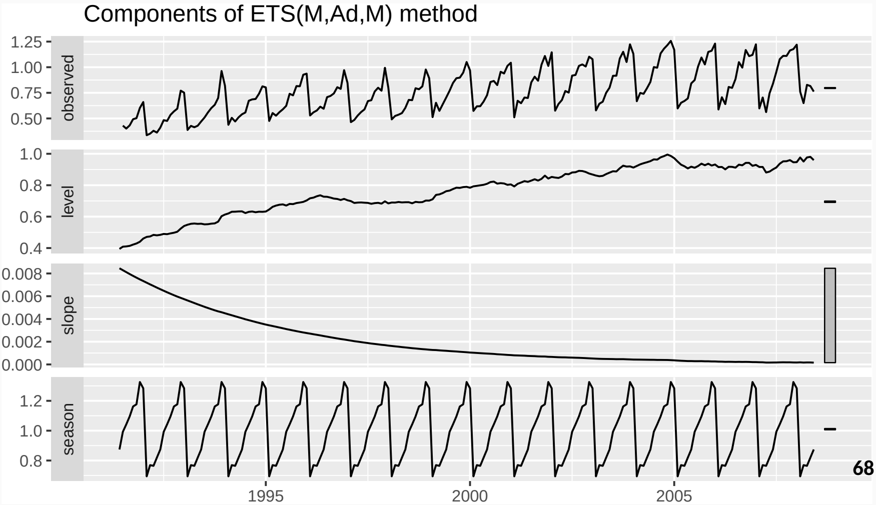
- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping
- Ensures the parameters are admissible (equivalent to invertible)
- Produces an object of class “ets”.

`ets` objects

- **Methods:** `coef()`, `autoplot()`, `plot()`, `summary()`, `residuals()`, `fitted()`, `simulate()` and `forecast()`
- `autoplot()` shows time plots of the original time series along with the extracted components (level, growth and seasonal).

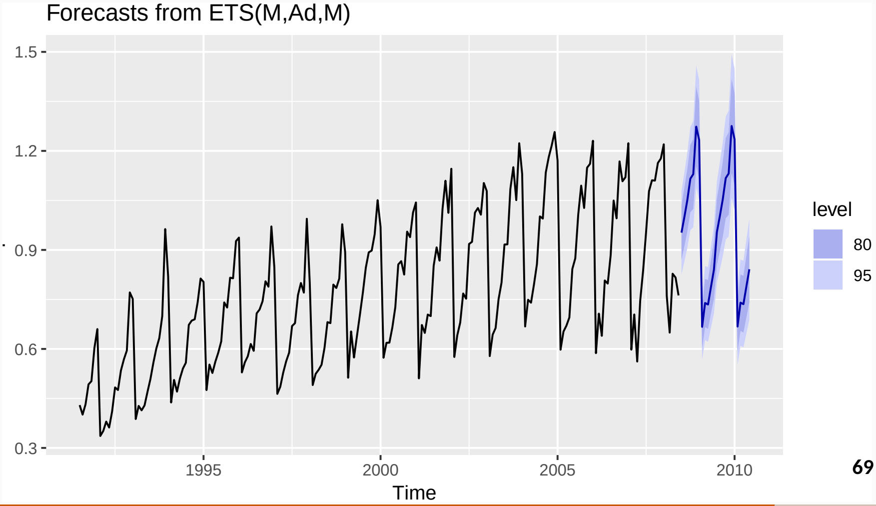
Example: drug sales

```
h02 %>% ets() %>% autoplot()
```



Example: drug sales

```r
h02 %>% ets() %>% forecast() %>% autoplot()
```



Example: drug sales

```r
h02 %>% ets() %>% accuracy()
```

```
##                    ME    RMSE     MAE    MPE  MAPE  MASE
## Training set 0.003873 0.05097 0.03904 0.1125 5.046 0.644
```

```r
h02 %>% ets(model="AAA", damped=FALSE) %>% accuracy()
```

```
##                     ME   RMSE     MAE    MPE  MAPE   MASE
## Training set -0.006447 0.0616 0.04949 -1.258 7.142 0.8164
```

The `ets()` function

`ets()` function also allows refitting model to new data set.

```
train <- window(h02, end=c(2004,12))
test <- window(h02, start=2005)
fit1 <- ets(train)
fit2 <- ets(test, model = fit1)
accuracy(fit2)
```

```
##                   ME    RMSE     MAE     MPE  MAPE   MASE    ACF1
## Training set 0.00144 0.05406 0.04314 -0.4332 5.218 0.6785 -0.4121
```

```
accuracy(forecast(fit1,10), test)
```

```
##                     ME    RMSE     MAE      MPE   MAPE  MASE     ACF1
## Training set  0.003427 0.04453 0.03290   0.1589  4.364 0.558  0.02236
## Test set     -0.077245 0.09158 0.07955 -10.0413 10.252 1.349 -0.04361
##              Theil's U
## Training set        NA
## Test set        0.6333
```

The `ets()` function in R

```
ets(y, model = "ZZZ", damped = NULL,
  additive.only = FALSE,
  lambda = NULL, biasadj = FALSE,
  lower = c(rep(1e-04, 3), 0.8),
  upper = c(rep(0.9999, 3), 0.98),
  opt.crit = c("lik","amse","mse","sigma","mae"),
  nmse = 3,
  bounds = c("both", "usual", "admissible"),
  ic = c("aicc", "aic", "bic"),
  restrict = TRUE,
  allow.multiplicative.trend = FALSE, ...)
```

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` (default), then either a damped or a non-damped trend will be selected according to the information criterion chosen.

The `ets()` function in R

- `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` (default). Otherwise, the time series will be transformed before the model is estimated. When `lambda` is not `NULL`, `additive.only` is set to `TRUE`.
- `biadadj`
  Uses bias-adjustment when undoing Box-Cox transformation for fitted values.

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"` (default) requires the parameters to satisfy both sets of constraints.
- `ic=aicc` (default) information criterion to be used in selecting models.
- `restrict=TRUE` (default) models that cause numerical problems not considered in model selection.
- `allow.multiplicative.trend` allows models with a multiplicative trend.

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, biasadj=FALSE,...)
```

- `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.

The `forecast()` function in R

- `simulate`: If `TRUE`, prediction intervals generated via simulation rather than analytic formulae. Even if `FALSE` simulation will be used if no algebraic formulae exist.
- `bootstrap`: If `bootstrap=TRUE` and `simulate=TRUE`, then 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.

The `forecast()` function in R

- `lambda`: The Box-Cox transformation parameter. Ignored if `lambda=NULL`. Otherwise, forecasts are back-transformed via inverse Box-Cox transformation.
- `biasadj`: Apply bias adjustment after Box-Cox?

Your turn

- Use `ets()` on some of these series:

  bicoal, *chicken*, *dole*, *usdeaths*, *bricksq*, *lynx*, *ibmclose*, *eggs*, *bricksq*, *ausbeer*

- Does it always give good forecasts?
- Find an example where it does not work well. Can you figure out why?