FC2018: Exponential smoothing

Samuel Orso

2018-09-25

Slides generously provided by Rob J Hyndman
Based on Chapter 7 of *Forecasting: Principles and Practice* by Rob J Hyndman and George Athanasopoulos

Outline

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

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 $y_1, y_2, \dots, y_T$.

Random walk forecasts

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

Average forecasts

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

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 $0 \le \alpha \le 1$.

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 $0 \le \alpha \le 1$.

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

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

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

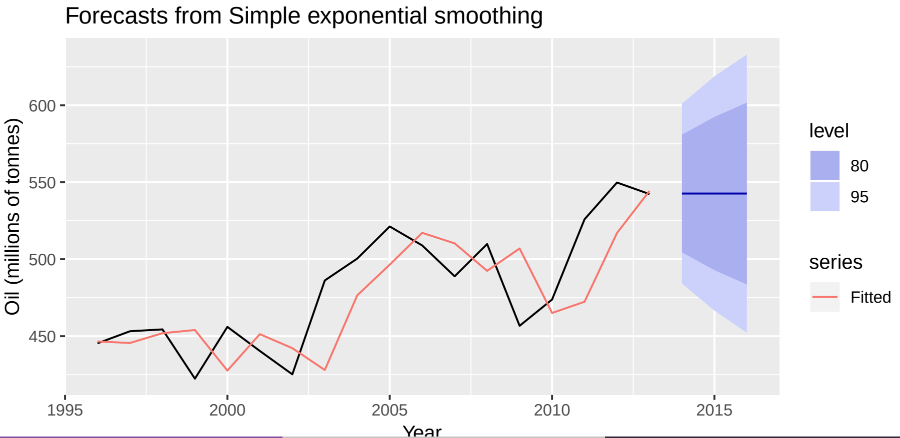
```
## Simple exponential smoothing
##
## Call:
##  ses(y = oildata, h = 5)
##
##   Smoothing parameters:
##     alpha = 0.8339
##
##   Initial states:
##     l = 446.5868
##
##   sigma:  29.83
##
##   AIC   AICc    BIC
```

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
2016	3			542.68

Example: Oil production

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



Outline

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

Holt's linear trend

Component form

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

Holt's linear trend

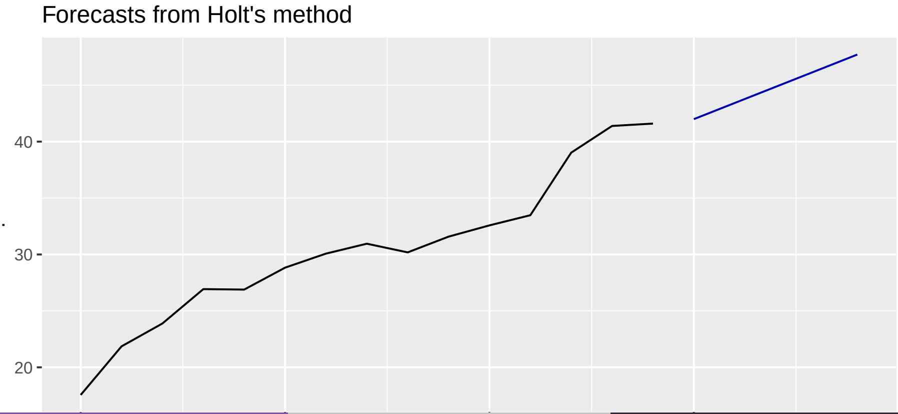
Component form

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

- Two smoothing parameters α and β^* $(0 \le \alpha, \beta^* \le 1)$.
- ℓ_t level: weighted average between y_t 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.
- Choose $\alpha, \beta^*, \ell_0, b_0$ to minimise SSE.

Holt's method in R

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



Damped trend method

Component form

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

Damped trend method

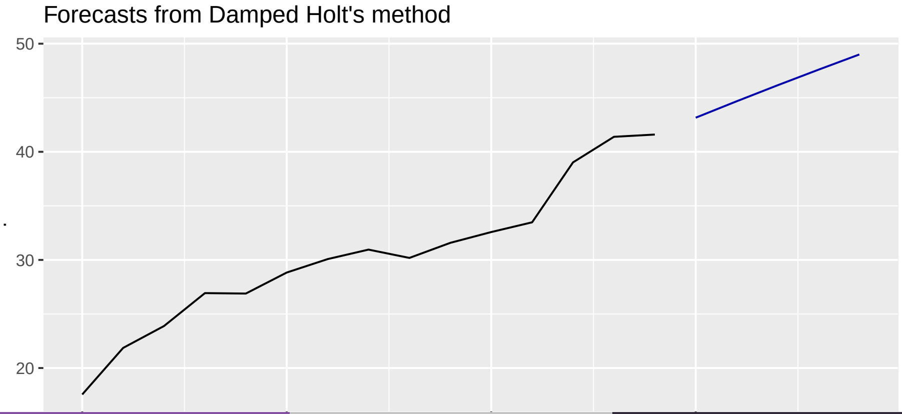
Component form

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

- 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

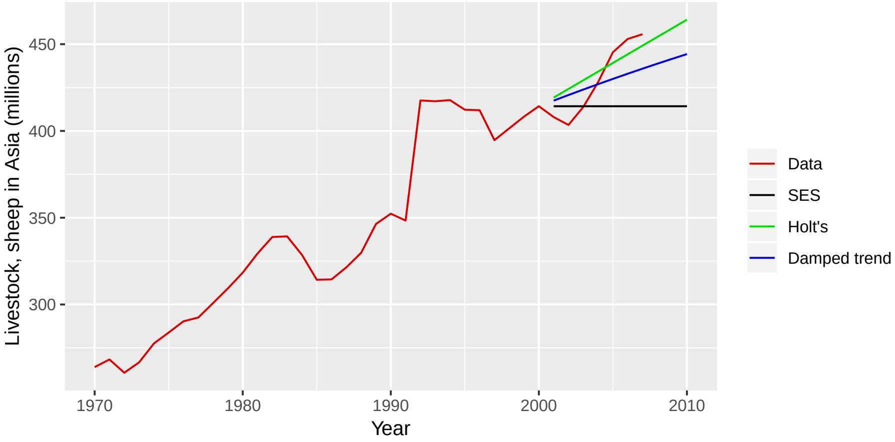
```r
livestock2 <- window(livestock, start=1970,
                                end=2000)
fit1 <- ses(livestock2)
fit2 <- holt(livestock2)
fit3 <- holt(livestock2, damped = TRUE)
```

```r
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

Example: Sheep in Asia (con't)



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

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

Holt-Winters additive method

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

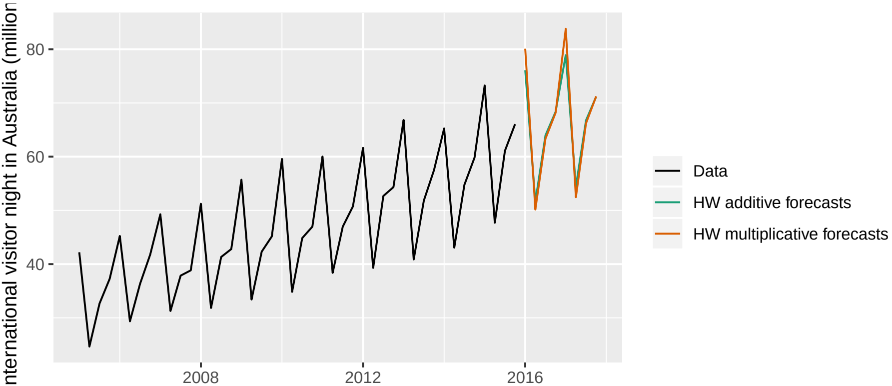
Component form

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

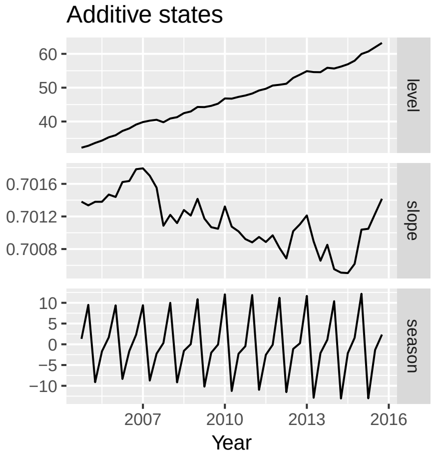
- $k =$ integer part of $(h-1)/m$. Ensures estimates from the final year are used for forecasting.
- Parameters: $0 \le \alpha \le 1$, $0 \le \beta^* \le 1$, $0 \le \gamma \le 1-\alpha$ and $m =$ period of seasonality (e.g. $m = 4$ for quarterly data).

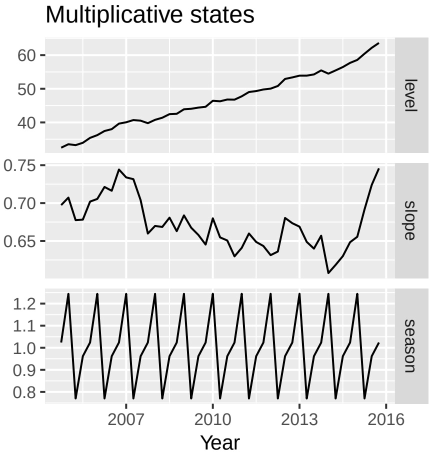
Example: Visitor Nights

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



Estimated components





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

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

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)

(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

There are also multiplicative trend methods (not recommended).

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

1 Simple exponential smoothing

2 Trend methods

3 Seasonal methods

4 Taxonomy of exponential smoothing methods

5 **Innovations state space models**

6 ETS in R

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)	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

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)	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

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

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

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

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

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

Method performed very well in M3 competition.

Outline

1 Simple exponential smoothing

2 Trend methods

3 Seasonal methods

4 Taxonomy of exponential smoothing methods

5 Innovations state space models

6 ETS in R

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

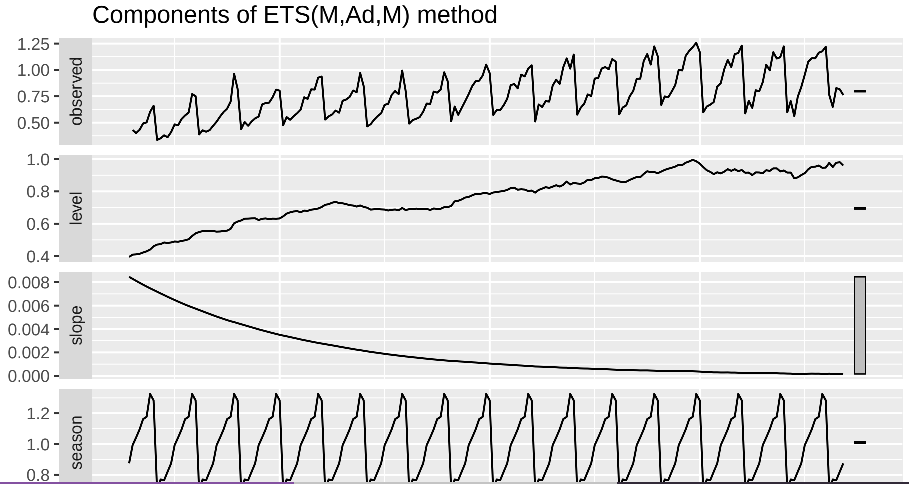
- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping
- Ensures the parameters are admissible (equivalent to invertible)
- Produces an object of class "ets".

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).

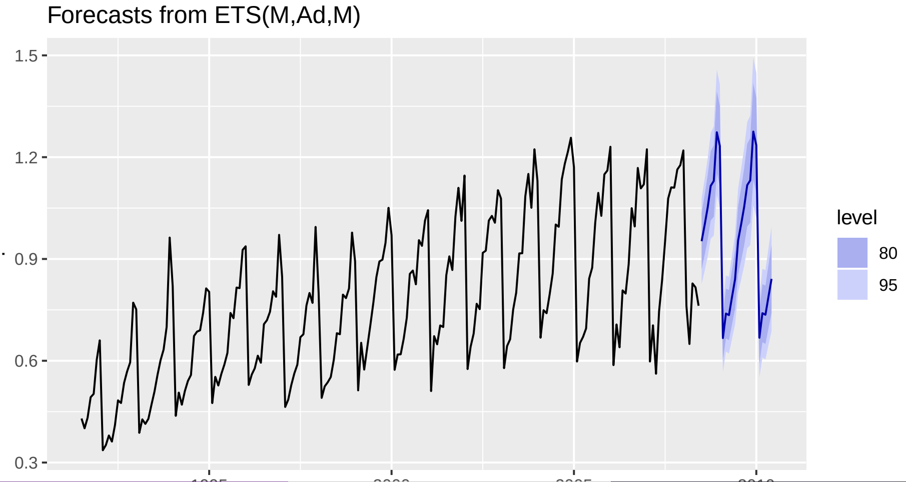
Example: drug sales

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



Example: drug sales

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



Example: drug sales

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

```
##                     ME    RMSE     MAE    MPE  MAPE  MASE
## Training set 0.003873 0.05097 0.03904 0.1125 5.046 0.644 0.0
```

```
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 0.
```

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

```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?