MONASH University

MONASH BUSINESS SCHOOL

Time Series and Forecasting

Exponential smoothing methods

Outline

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

Component form

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

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

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

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

$$\vdots$$

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

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

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" that follows a random walk.
- Need to estimate α and ℓ_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.

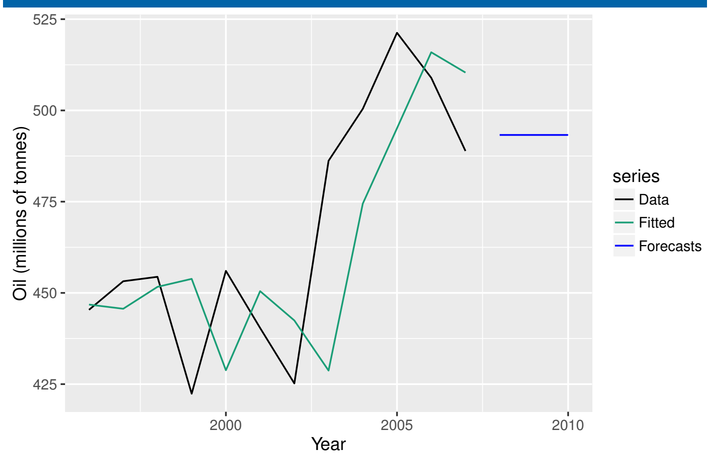
Example: Oil production

Year	Time t	Observation y_t	Level ℓ_t	Forecast $\hat{y}_{t\|t-1}$	Error $\hat{e}_t$
1995	0		446.80		
1996	1	445.36	445.66	446.80	-1.43
1997	2	453.20	451.65	445.66	7.54
1998	3	454.41	453.85	451.65	2.75
1999	4	422.38	428.81	453.85	-31.47
2000	5	456.04	450.47	428.81	27.23
2001	6	440.39	442.45	450.47	-10.09
2002	7	425.19	428.72	442.45	-17.25
2003	8	486.21	474.46	428.72	57.49
2004	9	500.43	495.12	474.46	25.97
2005	10	521.28	515.93	495.12	26.15
2006	11	508.95	510.37	515.93	-6.98
2007	12	488.89	493.28	510.37	-21.49
	h			$\hat{y}_{T+h\|T}$	
2008	1			493.28	
2009	2			493.28	
2010	3			493.28	

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

$\alpha = 0.796$ and $\ell_0 = 446.80$ are obtained by minimising SSE over periods $t = 1, 2, \dots, 12$.

Example: Oil production



SES in R

```
fit <- ses(oildata, h=3)
autoplot(fit)

accuracy(fit)
```


Simple exponential smoothing

Multi-step forecasts

$$\hat{y}_{T+h|T} = \hat{y}_{T+1|T}, \qquad h = 2, 3, \dots$$

- A "flat" forecast function.
- Remember, a forecast is an estimated mean of a future value.
- So with no trend, no seasonality, and no other patterns, the forecasts are constant.

Outline

Holt's local trend method

- Holt (1957) extended SES to allow forecasting of data with trends.
- Two smoothing parameters: α and β^* (with values between 0 and 1).

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

- ℓ_t denotes an estimate of the level of the series at time t
- b_t denotes an estimate of the slope of the series at time t.

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

```
air <- window(ausair, start=1990, end=2004)
fit2 <- holt(air, h=5)
```


Comparing Holt and SES

- Holt's method will almost always have better in-sample RMSE because it is optimized over one additional parameter.
- It may not be better on other measures.
- You need to compare out-of-sample RMSE (using a test set) for the comparison to be useful.
- But we don't have enough data.
- A better method for comparison will be coming up!

Additive damped trend

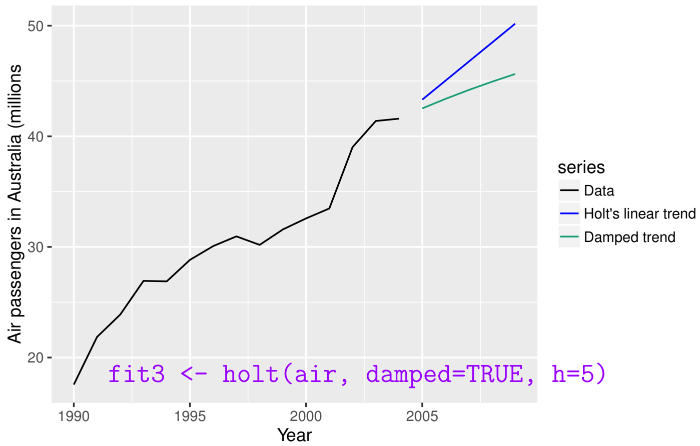
- Gardner and McKenzie (1985) suggested that the trends should be "damped" to be more conservative for longer forecast horizons.
- Damping parameter $0 < \phi < 1$.

State space form

Forecast equation	$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t$
Observation equation	$y_t = \ell_{t-1} + \phi b_{t-1} + e_t$
State equations	$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha e_t$
	$b_t = \phi b_{t-1} + \beta e_t$

- 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




```
fit3 <- holt(air, damped=TRUE, h=5)
```

Outline

Holt-Winters additive method

- Holt and Winters extended Holt's method to capture seasonality.
- Three smoothing equations—one for the level, one for trend, and one for seasonality.
- Parameters: $0 \le \alpha \le 1$, $0 \le \beta^* \le 1$, $0 \le \gamma \le 1-\alpha$ and $m =$ period of seasonality.

State space form

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

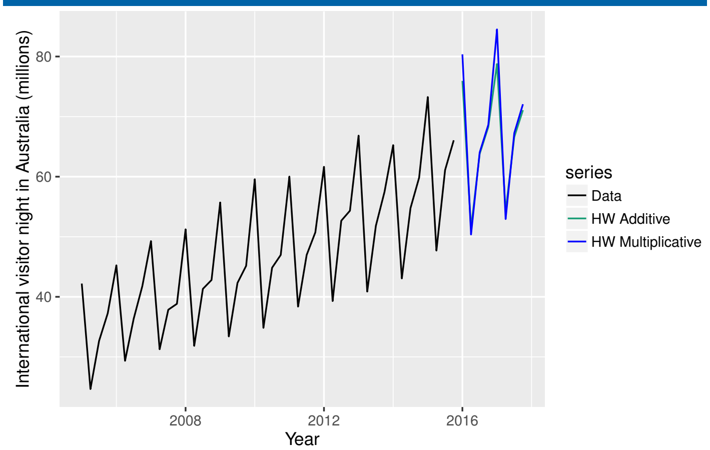
Holt-Winters multiplicative

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

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

- 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 $\sum_i s_i \approx m$.
- We optimize for α, β^*, γ, ℓ_0, b_0, s_0, $s_{-1}, \dots, s_{1-m}$.

Example 7.4: Visitor Nights



Holt-Winters damped method

Often the single most accurate forecasting method for seasonal data:

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

Exponential smoothing methods

- Simple exponential smoothing: no trend.
  `ses(x)`
- Holt's method: linear trend.
  `holt(x)`
- Damped trend method.
  `holt(x, damped=TRUE)`
- Holt-Winters methods
  `hw(x, damped=TRUE, seasonal="additive")`

Outline

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 9 separate exponential smoothing methods.

Recursive formulae

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

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.

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

Innovations state space models

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

Examples: Error Trend Seasonal

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

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 **ETS**(Error,Trend,Seasonal)
Exponen**T**ial **S**moothing

ETS(A,N,N):	Simple exponential smoothing with additive errors
ETS(M,N,N):	Simple exponential smoothing with multiplicative errors
ETS(A,A,N):	Holt's linear method with additive errors
ETS(A,A,A):	Additive Holt-Winters' method with additive errors

Estimating ETS models

- Smoothing parameters α, β, γ and ϕ, and the initial states ℓ_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.

Model selection

Akaike's Information Criterion

$$\text{AIC} = -2\log(\text{L}) + 2(k+1)$$

where L is the likelihood and k is the number of parameters and 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

$$\text{BIC} = \text{AIC} + (k+1)(\log(T)-2).$$

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 Trend Component		Seasonal 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 Trend Component		Seasonal 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

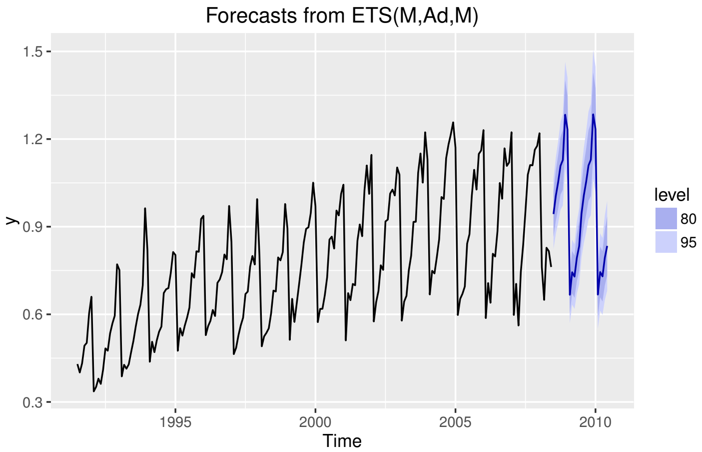
Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

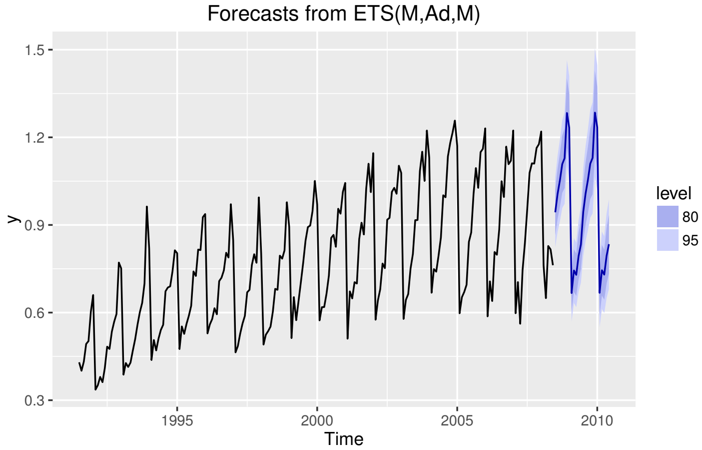
- The prediction intervals will differ between models with additive and multiplicative methods.
- Exact formulae for some models.
- More general to simulate future sample paths, conditional on the last estimate of the states, and to obtain prediction intervals from the percentiles of these simulated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

Outline

Exponential smoothing



Exponential smoothing



Exponential smoothing

```
ETS(M,Ad,M)

  Smoothing parameters:
    alpha = 0.2481
    beta  = 2e-04
    gamma = 2e-04
    phi   = 0.9722

  Initial states:
    l = 0.3948
    b = 0.0101
    s = 0.8688 0.8191 0.7579 0.7897 0.7041 1.2889
        1.3253 1.1718 1.1597 1.1018 1.0385 0.9744

  sigma:  0.0654

     AIC     AICc      BIC
-120.929 -117.638  -64.521
```


The `ets()` function in R

```
ets(y, model="ZZZ", damped=NULL,
   additive.only=FALSE,
   lambda=NULL, biasadj=FALSE,
   lower=c(rep(0.0001,3),0.80),
   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("aic","aicc","bic"), restrict=TRUE,
   allow.multiplicative.trend=FALSE,
   use.initial.values=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` (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

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

The `forecast()` function in R

- `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.
- `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?