MONASH University

MONASH BUSINESS SCHOOL

Forecasting: principles and practice

Rob J Hyndman

1.5 State space models

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 *state* 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 Trend Seasonal

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 Trend Seasonal

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 **Trend** **Seasonal**

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

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

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

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

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

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

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

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

Error correction form

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

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

ETS(A,N,N)

SES with additive errors

Observation 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.
- Observation equation: relationship between observations and states.
- State equation(s): evolution of the state(s) through time.

ETS(A,A,N)

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

Holt's linear method with additive errors

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

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

ETS(A,A,A)

Holt-Winters additive method with additive errors

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

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

ETS(M,N,N)

SES with multiplicative errors:

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

ETS(M,N,N)

SES with multiplicative errors:

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

SES with multiplicative errors

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 $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$

SES with multiplicative errors

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,A,N)

Holt's linear method with multiplicative errors

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

- $\varepsilon_t = \frac{y_t - (\ell_{t-1} + b_{t-1})}{(\ell_{t-1} + b_{t-1})}$
- $\beta = \alpha\beta^*$ and $\varepsilon_t \sim \text{NID}(0, \sigma^2)$.

Additive error models

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

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.

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 prediction intervals using underlying state space model.

Method performed very well in M3 competition.

Some unstable models

- 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 of the combinations of (Error, Trend, Seasonal) can lead to numerical difficulties.

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

Prediction intervals

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.

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)	$\sigma_h = \sigma^2\Big[1+(h-1)\{\alpha^2+\alpha\beta h+\frac{1}{6}\beta^2 h(2h-1)\}+\gamma k\{2\alpha+\gamma+\beta m(k+1)\}\Big]$
(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

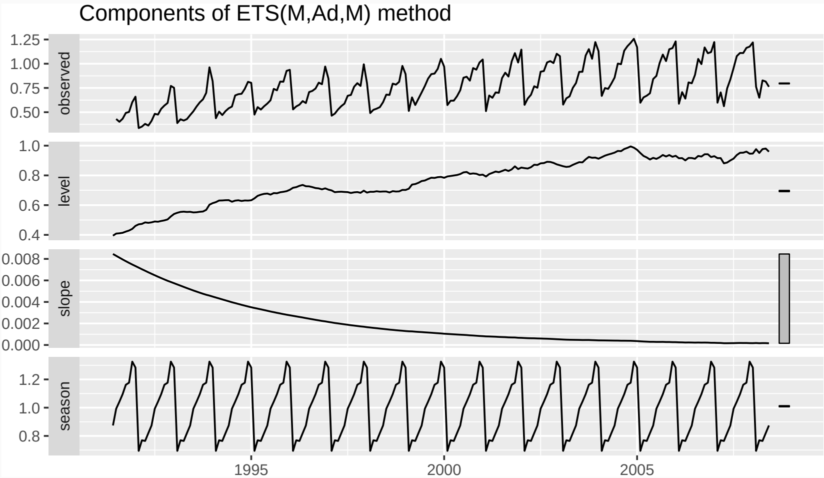
- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping
- Produces prediction intervals for every model
- Ensures the parameters are admissible (equivalent to invertible)
- Produces an object of class "ets".

ets objects

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

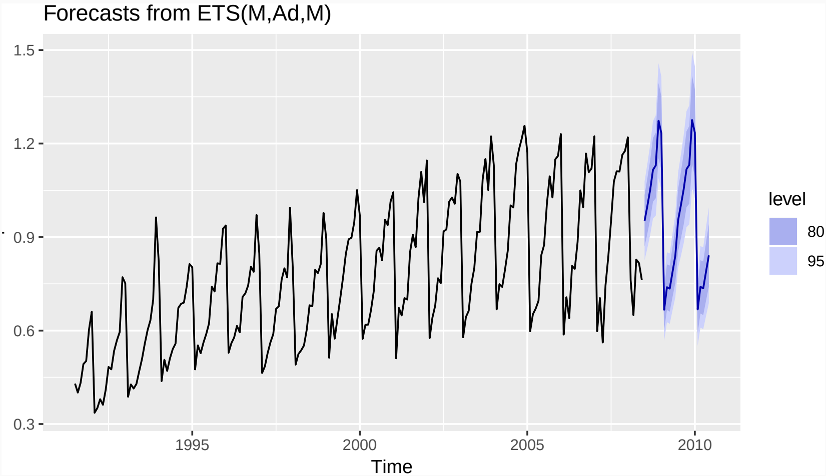
Example: drug sales

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



Example: drug sales

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



Example: drug sales

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

```
##                     ME    RMSE     MAE    MPE  MAPE
## Training set 0.003873 0.05097 0.03904 0.1125 5.046
##               MASE     ACF1
## Training set 0.644 0.006125
```

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

```
##                      ME   RMSE     MAE    MPE  MAPE
## Training set -0.006447 0.0616 0.04949 -1.258 7.142
##                MASE   ACF1
## Training set 0.8164 0.2612
```

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
## Training set 0.00144 0.05406 0.04314 -0.4332 5.218
##                MASE    ACF1
## Training set 0.6785 -0.4121
```

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

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

Outline

Lab Session 9

Outline

Lab Session 10