MONASH University

Rob J Hyndman

Forecasting:

Principles and Practice

4. Exponential smoothing II

OTexts.com/fpp/7/

A confusing array of methods?

- All these methods can be confusing!
- How to choose between them?
- The ETS framework provides an automatic way of selecting the best method.
- It was developed to solve the problem of automatically forecasting pharmaceutical sales across thousands of products.

A confusing array of methods?

- All these methods can be confusing!
- How to choose between them?
- The ETS framework provides an automatic way of selecting the best method.
- It was developed to solve the problem of automatically forecasting pharmaceutical sales across thousands of products.

A confusing array of methods?

- All these methods can be confusing!
- How to choose between them?
- The ETS framework provides an automatic way of selecting the best method.
- It was developed to solve the problem of automatically forecasting pharmaceutical sales across thousands of products.

A confusing array of methods?

- All these methods can be confusing!
- How to choose between them?
- The ETS framework provides an automatic way of selecting the best method.
- It was developed to solve the problem of automatically forecasting pharmaceutical sales across thousands of products.

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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

Exponential smoothing methods

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

N,N: Simple exponential smoothing

Exponential smoothing methods

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

N,N: Simple exponential smoothing

A,N: Holt's linear method

Exponential smoothing methods

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

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method

Exponential smoothing methods

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

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
M,N: Exponential trend method

Exponential smoothing methods

Trend Component		Seasonal Component: N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative damped trend method

Exponential smoothing methods

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

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative damped trend method
A,A: Additive Holt-Winters' method

Exponential smoothing methods

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

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative damped trend method
A,A: Additive Holt-Winters' method
A,M: Multiplicative Holt-Winters' method

Exponential smoothing methods

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

There are 15 separate exponential smoothing methods.

State space form

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

Outline

1 Taxonomy of exponential smoothing methods

2 Innovations state space models

3 ETS in R

4 Forecasting with ETS models

Methods v Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

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

Methods v Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

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

Methods v Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

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

Methods v Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

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

Methods v Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

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

ETS models

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

ETS models

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

ETS models

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

ETS models

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

ETS models

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

ETS models

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

Exponential smoothing methods

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

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

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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

General notation

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

Exponential smoothing methods

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

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

↑

Trend

Examples:

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

Exponential smoothing methods

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

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

Trend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

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

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

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

Examples:

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

Exponential smoothing methods

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

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

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

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Innovations state space models

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

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

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

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

ETS(A,N,N)

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

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

ETS(A,N,N)

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

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

ETS(A,N,N)

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

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

ETS(M,N,N)

SES with multiplicative errors.

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

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

ETS(M,N,N)

SES with multiplicative errors.

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

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

ETS(M,N,N)

SES with multiplicative errors.

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

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

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

ETS(M,N,N)

SES with multiplicative errors.

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

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

ETS(M,N,N)

SES with multiplicative errors.

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

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

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

ETS(M,N,N)

SES with multiplicative errors.

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

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

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

ETS(M,N,N)

SES with multiplicative errors.

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

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

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

Holt's linear method

ETS(A,A,N)

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

ETS(M,A,N)

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

Holt's linear method

ETS(A,A,N)

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

ETS(M,A,N)

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

ETS(A,A,A)

Holt-Winters additive method with additive errors.

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

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

Additive error models

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

Multiplicative error models

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

Innovations state space models

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

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

Additive errors:

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

Multiplicative errors:

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

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

Innovations state space models

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

Innovations state space models

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

Innovations state space models

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

Some unstable models

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

Some unstable models

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

Some unstable models

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

Exponential smoothing models

Additive Error

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~~
M	(Multiplicative)	~~A,M,N~~	~~A,M,A~~	~~A,M,M~~
M_d	(Multiplicative damped)	~~A,M_d,N~~	~~A,M_d,A~~	~~A,M_d,M~~

Multiplicative Error

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
M	(Multiplicative)	M,M,N	~~M,M,A~~	M,M,M
M_d	(Multiplicative damped)	M,M_d,N	~~M,M_d,A~~	M,M_d,M

Innovations state space models

Estimation

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

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

Innovations state space models

Estimation

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

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

Parameter restrictions

Usual region

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

Admissible region

Parameter restrictions

Usual region

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

Admissible region

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

Parameter restrictions

Usual region

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

Admissible region

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

Parameter restrictions

Usual region

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

Admissible region

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

Parameter restrictions

Usual region

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

Admissible region

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

Parameter restrictions

Usual region

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

Admissible region

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

Parameter restrictions

Usual region

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

Admissible region

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

Model selection

Akaike's Information Criterion

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

where p is the number of estimated parameters in the model.

- *Minimizing* the AIC gives the best model for prediction.

AIC corrected (for small sample bias)

$$\text{AIC}_\text{C} = \text{AIC} + \frac{2(p+1)(p+2)}{n-p}$$

Schwartz' Bayesian IC

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

Model selection

Akaike's Information Criterion

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

where p is the number of estimated parameters in the model.

- *Minimizing* the AIC gives the best model for prediction.

AIC corrected (for small sample bias)

$$\text{AIC}_{\text{C}} = \text{AIC} + \frac{2(p+1)(p+2)}{n-p}$$

Schwartz' Bayesian IC

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

Model selection

Akaike's Information Criterion

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

where p is the number of estimated parameters in the model.

- *Minimizing* the AIC gives the best model for prediction.

AIC corrected (for small sample bias)

$$\text{AIC}_\text{C} = \text{AIC} + \frac{2(p+1)(p+2)}{n-p}$$

Schwartz' Bayesian IC

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

Model selection

Akaike's Information Criterion

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

where p is the number of estimated parameters in the model.

- *Minimizing* the AIC gives the best model for prediction.

AIC corrected (for small sample bias)

$$\text{AIC}_{\text{C}} = \text{AIC} + \frac{2(p+1)(p+2)}{n-p}$$

Schwartz' Bayesian IC

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

Model selection

Akaike's Information Criterion

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

where p is the number of estimated parameters in the model.

- *Minimizing* the AIC gives the best model for prediction.

AIC corrected (for small sample bias)

$$\text{AIC}_\text{C} = \text{AIC} + \frac{2(p+1)(p+2)}{n-p}$$

Schwartz' Bayesian IC

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

Akaike's Information Criterion

- Value of AIC/AICc/BIC given in the R output.
- AIC does not have much meaning by itself. Only useful in comparison to AIC value for another model fitted to *same data set.*
- Consider several models with AIC values close to the minimum.
- A difference in AIC values of 2 or less is not regarded as substantial and you may choose the simpler but non-optimal model.
- AIC can be negative.

Akaike's Information Criterion

- Value of AIC/AICc/BIC given in the R output.
- AIC does not have much meaning by itself. Only useful in comparison to AIC value for another model fitted to *same data set.*
- Consider several models with AIC values close to the minimum.
- A difference in AIC values of 2 or less is not regarded as substantial and you may choose the simpler but non-optimal model.
- AIC can be negative.

Akaike's Information Criterion

- Value of AIC/AICc/BIC given in the R output.
- AIC does not have much meaning by itself. Only useful in comparison to AIC value for another model fitted to *same data set.*
- Consider several models with AIC values close to the minimum.
- A difference in AIC values of 2 or less is not regarded as substantial and you may choose the simpler but non-optimal model.
- AIC can be negative.

Akaike's Information Criterion

- Value of AIC/AICc/BIC given in the R output.
- AIC does not have much meaning by itself. Only useful in comparison to AIC value for another model fitted to *same data set.*
- Consider several models with AIC values close to the minimum.
- A difference in AIC values of 2 or less is not regarded as substantial and you may choose the simpler but non-optimal model.
- AIC can be negative.

Akaike's Information Criterion

- Value of AIC/AICc/BIC given in the R output.
- AIC does not have much meaning by itself. Only useful in comparison to AIC value for another model fitted to *same data set.*
- Consider several models with AIC values close to the minimum.
- A difference in AIC values of 2 or less is not regarded as substantial and you may choose the simpler but non-optimal model.
- AIC can be negative.

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.

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.

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.

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.

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.

Outline

Exponential smoothing

```
fit <- ets(ausbeer)
fit2 <- ets(ausbeer,model="AAA",damped=FALSE)
fcast1 <- forecast(fit, h=20)
fcast2 <- forecast(fit2, h=20)
```

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

Exponential smoothing

```
fit <- ets(ausbeer)
fit2 <- ets(ausbeer,model="AAA",damped=FALSE)
fcast1 <- forecast(fit, h=20)
fcast2 <- forecast(fit2, h=20)
```

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

Exponential smoothing

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

  Smoothing parameters:
    alpha = 0.1776
    beta  = 0.0454
    gamma = 0.1947
    phi   = 0.9549

  Initial states:
    l = 263.8531
    b = 0.9997
    s = 1.1856 0.9109 0.8612 1.0423

  sigma:  0.0356

     AIC     AICc      BIC
2272.549 2273.444 2302.715
```

Exponential smoothing

```
> fit2
ETS(A,A,A)

  Smoothing parameters:
    alpha = 0.2079
    beta  = 0.0304
    gamma = 0.2483

  Initial states:
    l = 255.6559
    b = 0.5687
    s = 52.3841 -27.1061 -37.6758 12.3978

  sigma:  15.9053

     AIC     AICc      BIC
2312.768 2313.481 2339.583
```

Exponential smoothing

`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`.

Exponential smoothing

`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`.

Exponential smoothing

`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`.

Exponential smoothing

`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`.

Exponential smoothing

`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`.

Exponential smoothing

`ets` objects

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

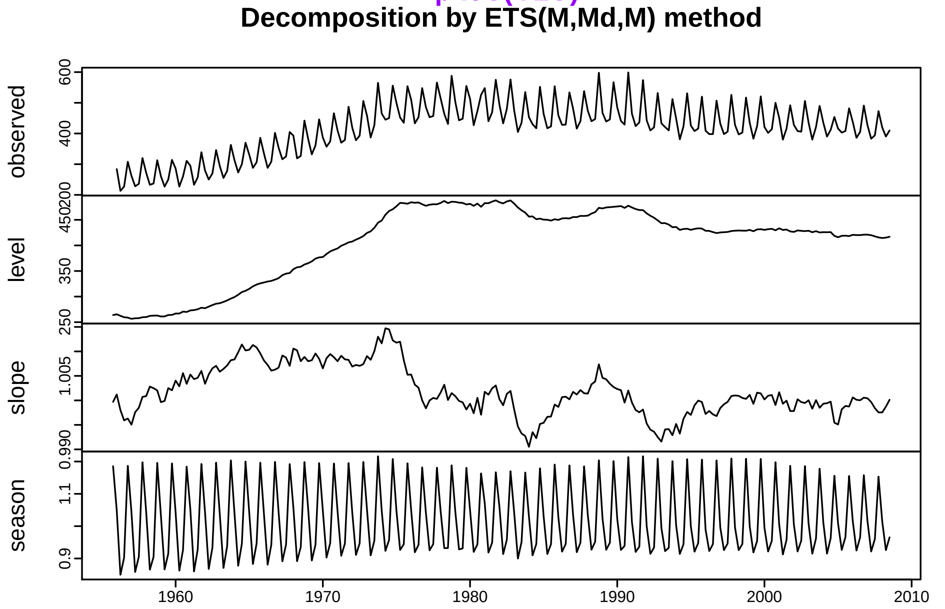
Exponential smoothing

`ets` objects

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

Exponential smoothing

`plot(fit)`

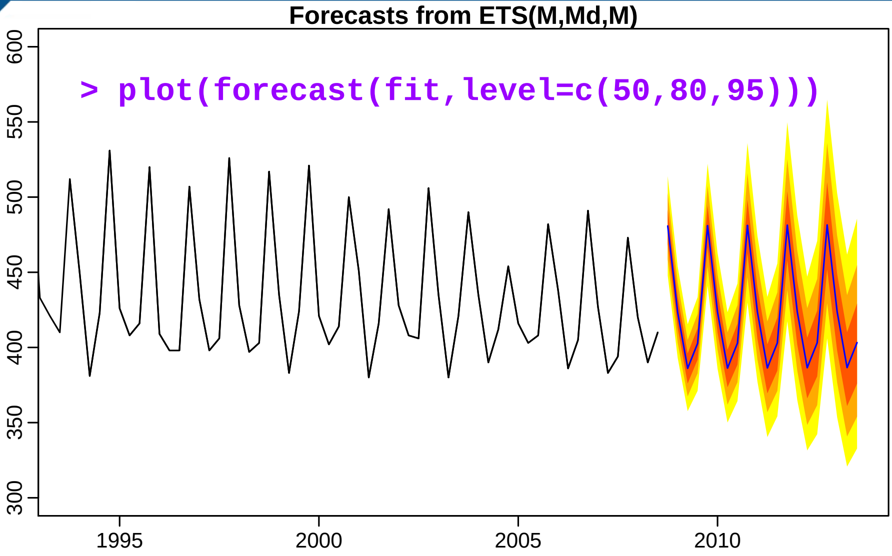


Goodness-of-fit

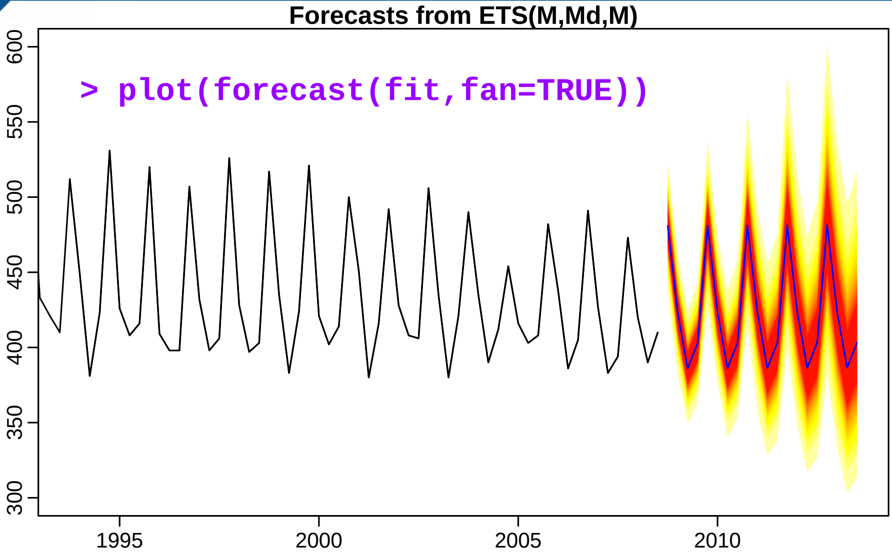
```
> accuracy(fit)
      ME     RMSE      MAE      MPE     MAPE     MASE
 0.17847 15.48781 11.77800  0.07204  2.81921  0.20705

> accuracy(fit2)
      ME     RMSE      MAE      MPE     MAPE     MASE
-0.11711 15.90526 12.18930 -0.03765  2.91255  0.21428
```

Forecast intervals



Forecast intervals



Exponential smoothing

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

```
> usfit <- ets(usnetelec[1:45])
> test <- ets(usnetelec[46:55], model = usfit)

> accuracy(test)
     ME     RMSE      MAE      MPE     MAPE     MASE
-3.35419 58.02763 43.85545 -0.07624  1.18483  0.52452

> accuracy(forecast(usfit,10), usnetelec[46:55])
      ME       RMSE        MAE        MPE       MAPE       MASE
  40.7034   61.2075   46.3246    1.0980    1.2620    0.6776
```

The `ets()` function in R

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

The `ets()` function in R

- `y`
  The time series to be forecast.

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

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

The `ets()` function in R

- `y`

  The time series to be forecast.

- `model`

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

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

The `ets()` function in R

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

Outline

Forecasting with ETS models

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

Forecasting with ETS models

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

Forecasting with ETS models

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

Forecasting with ETS models

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

Forecasting with ETS models

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

Forecasting with ETS models

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

Forecasting with ETS models

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

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

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

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

Forecasting with ETS models

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

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

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

Forecasting with ETS models

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

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

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

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

Forecasting with ETS models

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

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

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

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

Forecasting with ETS models

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

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

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

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

Forecasting with ETS models

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

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

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

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

Forecasting with ETS models

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

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

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

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

Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

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

Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

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

Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

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

Forecasting with ETS models

Prediction intervals: cannot be generated using the methods.

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