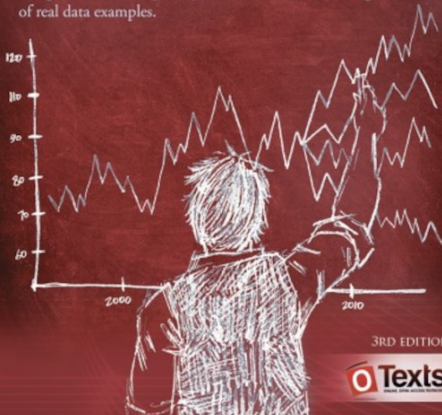


Rob J Hyndman  
George Athanasopoulos

# FORECASTING

## PRINCIPLES AND PRACTICE

A comprehensive introduction to the latest forecasting methods using R. Learn to improve your forecast accuracy using dozens of real data examples.



3RD EDITION

 **OTexts**  
OPEN TEXTS FOR PRACTICE

## 8. Exponential smoothing

### 8.6 Estimation and model selection

[OTexts.org/fpp3/](https://OTexts.org/fpp3/)

# Estimating ETS models

- Smoothing parameters  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\phi$ , 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.

# Innovations state space models

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

$$y_t = \underbrace{h(\mathbf{x}_{t-1})}_{\mu_t} + \underbrace{k(\mathbf{x}_{t-1})\varepsilon_t}_{e_t}$$

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + g(\mathbf{x}_{t-1})\varepsilon_t$$

## Additive errors

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

## Multiplicative errors

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

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

# Innovations state space models

## Estimation

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

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

# 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 *traditional* region.
- For example for ETS(A,N,N):  
*traditional*  $0 < \alpha < 1$  while *admissible*  $0 < \alpha < 2$ .

# Model selection

## Akaike's Information Criterion

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

where  $L$  is the likelihood and  $k$  is the number of parameters & initial states estimated in the model.

# Model selection

## Akaike's Information Criterion

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

where  $L$  is the likelihood and  $k$  is the number of parameters & initial states estimated in the model.

## Corrected AIC

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

which is the AIC corrected (for small sample bias).



# Model selection

## Akaike's Information Criterion

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

where  $L$  is the likelihood and  $k$  is the number of parameters & initial states estimated in the model.

## Corrected AIC

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

which is the AIC corrected (for small sample bias).

## Bayesian Information Criterion

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

## AIC and cross-validation

Minimizing the AIC assuming Gaussian residuals is asymptotically equivalent to minimizing one-step time series cross validation MSE.

# Automatic forecasting

**From Hyndman et al. (IJF, 2002):**

- Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AICc:
- Produce forecasts using best method.
- Obtain forecast intervals using underlying state space model.

Method performed very well in M3 competition.

## Some unstable models

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

# Example: National populations

```
fit <- global_economy |>
  mutate(Pop = Population / 1e6) |>
  model(ets = ETS(Pop))
fit
```

```
## # A tibble: 263 x 2
## # Key:   Country [263]
##   Country          ets
##   <fct>          <model>
## 1 Afghanistan    <ETS(A,A,N)>
## 2 Albania         <ETS(M,A,N)>
## 3 Algeria         <ETS(M,A,N)>
## 4 American Samoa  <ETS(M,A,N)>
## 5 Andorra         <ETS(M,A,N)>
## 6 Angola          <ETS(M,A,N)>
## 7 Antigua and Barbuda <ETS(M,A,N)>
## 8 Arab World      <ETS(M,A,N)>
## 9 Argentina       <ETS(A,A,N)>
## 10 Armenia        <ETS(M,A,N)>
```

# Example: National populations

```
fit |>  
  forecast(h = 5)
```

```
## # A tibble: 1,315 x 5 [1Y]  
## # Key:      Country, .model [263]  
##   Country      .model Year      Pop .mean  
##   <fct>        <chr>  <dbl>      <dist> <dbl>  
## 1 Afghanistan ets     2018    N(36, 0.012) 36.4  
## 2 Afghanistan ets     2019    N(37, 0.059) 37.3  
## 3 Afghanistan ets     2020    N(38, 0.16) 38.2  
## 4 Afghanistan ets     2021    N(39, 0.35) 39.0  
## 5 Afghanistan ets     2022    N(40, 0.64) 39.9  
## 6 Albania      ets     2018    N(2.9, 0.00012) 2.87  
## 7 Albania      ets     2019    N(2.9, 6e-04) 2.87  
## 8 Albania      ets     2020    N(2.9, 0.0017) 2.87  
## 9 Albania      ets     2021    N(2.9, 0.0036) 2.86  
## 10 Albania     ets     2022    N(2.9, 0.0066) 2.86  
## # ... with 1,305 more rows
```

# Example: Australian holiday tourism

```
holidays <- tourism |>
  filter(Purpose == "Holiday")
fit <- holidays |> model(ets = ETS(Trips))
fit
```

```
## # A mable: 76 x 4
```

```
## # Key:      Region, State, Purpose [76]
```

##	Region	State	Purpose	ets
##	<chr>	<chr>	<chr>	<model>
## 1	Adelaide	South Australia	Holiday	<ETS(A,N,A)>
## 2	Adelaide Hills	South Australia	Holiday	<ETS(A,A,N)>
## 3	Alice Springs	Northern Territory	Holiday	<ETS(M,N,A)>
## 4	Australia's Coral Coast	Western Australia	Holiday	<ETS(M,N,A)>
## 5	Australia's Golden Outback	Western Australia	Holiday	<ETS(M,N,M)>
## 6	Australia's North West	Western Australia	Holiday	<ETS(A,N,A)>
## 7	Australia's South West	Western Australia	Holiday	<ETS(M,N,M)>
## 8	Ballarat	Victoria	Holiday	<ETS(M,N,A)>
## 9	Barkly	Northern Territory	Holiday	<ETS(A,N,A)>
## 10	Barossa	South Australia	Holiday	<ETS(A,N,N)>

# Example: Australian holiday tourism

```
fit |>  
  filter(Region == "Snowy Mountains") |>  
  report()
```

```
## Series: Trips  
## Model: ETS(M,N,A)  
##   Smoothing parameters:  
##     alpha = 0.157  
##     gamma = 1e-04  
##  
##   Initial states:  
##   l[0] s[0] s[-1] s[-2] s[-3]  
##   142  -61   131 -42.2 -27.7  
##  
##   sigma^2:  0.0388  
##  
##   AIC AICc  BIC  
##   852  854  869
```



# Example: Australian holiday tourism

```
fit |>
  filter(Region == "Snowy Mountains") |>
  components(fit)
```

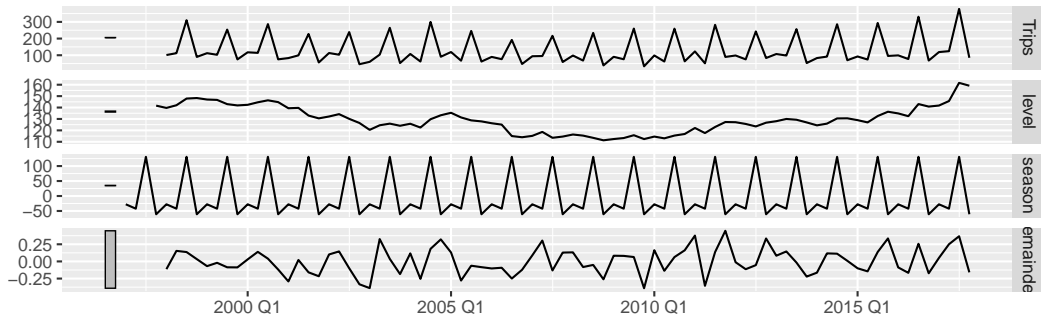
```
## # A dtable: 84 x 9 [1Q]
## # Key:      Region, State, Purpose, .model [1]
## # :        Trips = (lag(level, 1) + lag(season, 4)) * (1 + remainder)
##   Region      State Purpose .model Quarter Trips level season remain~1
##   <chr>        <chr>  <chr>  <chr>    <qtr>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Snowy Mountains New S~ Holiday ets     1997 Q1  NA      NA    -27.7  NA
## 2 Snowy Mountains New S~ Holiday ets     1997 Q2  NA      NA    -42.2  NA
## 3 Snowy Mountains New S~ Holiday ets     1997 Q3  NA      NA    131.   NA
## 4 Snowy Mountains New S~ Holiday ets     1997 Q4  NA     142.  -61.0  NA
## 5 Snowy Mountains New S~ Holiday ets     1998 Q1 101.    140.  -27.7  -0.113
## 6 Snowy Mountains New S~ Holiday ets     1998 Q2 112.    142.  -42.2   0.154
## 7 Snowy Mountains New S~ Holiday ets     1998 Q3 310.    148.  131.   0.137
## 8 Snowy Mountains New S~ Holiday ets     1998 Q4  89.8   148.  -61.0   0.0335
## 9 Snowy Mountains New S~ Holiday ets     1999 Q1 112.    147.  -27.7  -0.0687
## 10 Snowy Mountains New S~ Holiday ets     1999 Q2 103.    147.  -42.2  -0.0199
```

# Example: Australian holiday tourism

```
fit |>  
  filter(Region == "Snowy Mountains") |>  
  components(fit) |>  
  autoplot()
```

ETS(M,N,A) decomposition

Trips = (lag(level, 1) + lag(season, 4)) \* (1 + remainder)



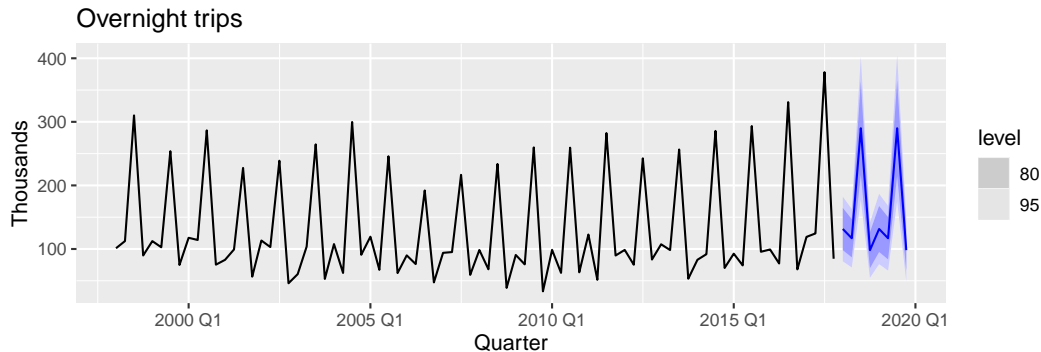
# Example: Australian holiday tourism

```
fit |> forecast()
```

```
## # A tibble: 608 x 7 [1Q]
## # Key:   Region, State, Purpose, .model [76]
##   Region      State      Purpose .model Quarter      Trips .mean
##   <chr>      <chr>      <chr>  <chr>    <qtr>      <dist> <dbl>
## 1 Adelaide    South Australia Holiday ets    2018 Q1 N(210, 457) 210.
## 2 Adelaide    South Australia Holiday ets    2018 Q2 N(173, 473) 173.
## 3 Adelaide    South Australia Holiday ets    2018 Q3 N(169, 489) 169.
## 4 Adelaide    South Australia Holiday ets    2018 Q4 N(186, 505) 186.
## 5 Adelaide    South Australia Holiday ets    2019 Q1 N(210, 521) 210.
## 6 Adelaide    South Australia Holiday ets    2019 Q2 N(173, 537) 173.
## 7 Adelaide    South Australia Holiday ets    2019 Q3 N(169, 553) 169.
## 8 Adelaide    South Australia Holiday ets    2019 Q4 N(186, 569) 186.
## 9 Adelaide Hills South Australia Holiday ets    2018 Q1   N(19, 36)  19.4
## 10 Adelaide Hills South Australia Holiday ets    2018 Q2   N(20, 36)  19.6
## # ... with 598 more rows
```

# Example: Australian holiday tourism

```
fit |>  
  forecast() |>  
  filter(Region == "Snowy Mountains") |>  
  autoplot(holidays) +  
  labs(y = "Thousands", title = "Overnight trips")
```



# Residuals

## Response residuals

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

## Innovation residuals

Additive error model:

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

Multiplicative error model:

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

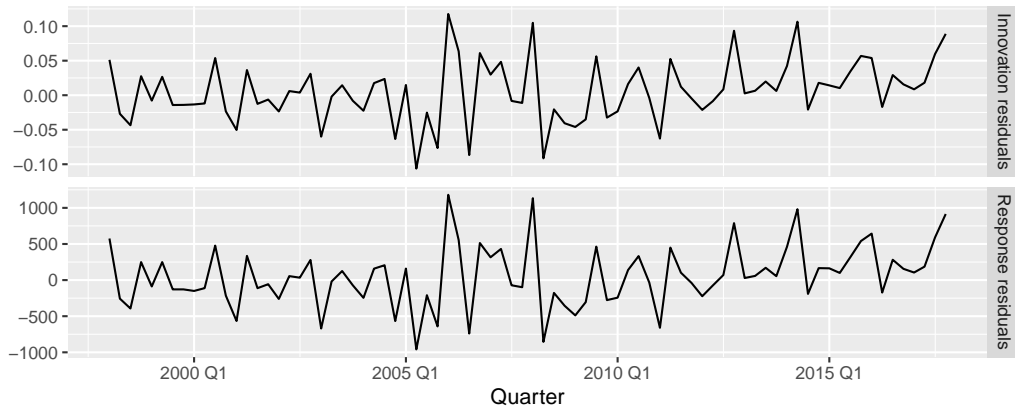
# Example: Australian holiday tourism

```
aus_holidays <- tourism |>
  filter(Purpose == "Holiday") |>
  summarise(Trips = sum(Trips))
fit <- aus_holidays |>
  model(ets = ETS(Trips)) |>
  report()
```

```
## Series: Trips
## Model: ETS(M,N,M)
## Smoothing parameters:
##   alpha = 0.358
##   gamma = 0.000969
##
## Initial states:
## l[0] s[0] s[-1] s[-2] s[-3]
## 9667 0.943 0.927 0.968 1.16
##
## sigma^2: 0.0022
##
## AIC AICc BIC
## 1331 1333 1348
```

# Example: Australian holiday tourism

```
residuals(fit)  
residuals(fit, type = "response")
```



# Example: Australian holiday tourism

```
fit |>  
  augment()
```

```
## # A tsibble: 80 x 6 [1Q]  
## # Key:           .model [1]  
##   .model Quarter Trips .fitted .resid .innov  
##   <chr>      <qtr>  <dbl>  <dbl>  <dbl>  <dbl>  
## 1 ets      1998 Q1 11806. 11230. 576.   0.0513  
## 2 ets      1998 Q2 9276.  9532. -257.  -0.0269  
## 3 ets      1998 Q3 8642.  9036. -393.  -0.0435  
## 4 ets      1998 Q4 9300.  9050. 249.   0.0275  
## 5 ets      1999 Q1 11172. 11260. -88.0  -0.00781  
## 6 ets      1999 Q2 9608.  9358. 249.   0.0266  
## 7 ets      1999 Q3 8914.  9042. -129.  -0.0142  
## 8 ets      1999 Q4 9026.  9154. -129.  -0.0140  
## 9 ets      2000 Q1 11071. 11221. -150.  -0.0134  
## 10 ets     2000 Q2 9196.  9308. -111.  -0.0120  
## # ... with 70 more rows
```



# Example: Australian holiday tourism

```
fit |>
  augment()
```

Innovation residuals (`.innov`) are given by  $\hat{\varepsilon}_t$  while regular residuals (`.resid`) are  $y_t - \hat{y}_{t-1}$ . They are different when the model has multiplicative errors.

```
## # A tsibble: 80 x 6 [1Q]
## # Key:           .model [1]
##   .model Quarter  Trips .fitted .resid  .innov
##   <chr>      <qtr>  <dbl>   <dbl>  <dbl>   <dbl>
## 1 ets       1998 Q1 11806.  11230.  576.    0.0513
## 2 ets       1998 Q2  9276.   9532. -257.   -0.0269
## 3 ets       1998 Q3  8642.   9036. -393.   -0.0435
## 4 ets       1998 Q4  9300.   9050.  249.    0.0275
## 5 ets       1999 Q1 11172.  11260. -88.0  -0.00781
## 6 ets       1999 Q2  9608.   9358.  249.    0.0266
## 7 ets       1999 Q3  8914.   9042. -129.   -0.0142
## 8 ets       1999 Q4  9026.   9154. -129.   -0.0140
## 9 ets       2000 Q1 11071.  11221. -150.   -0.0134
## 10 ets      2000 Q2  9196.   9308. -111.   -0.0120
## # ... with 70 more rows
```