MONASH University

MONASH BUSINESS SCHOOL

ETC3550/ETC5550 Applied forecasting

Ch8. Exponential smoothing

OTexts.org/fpp3/

Outline

Outline

Historical perspective

- Developed in the 1950s and 1960s as methods (algorithms) to produce point forecasts.
- Combine a “level”, “trend” (slope) and “seasonal” component to describe a time series.
- The rate of change of the components are controlled by “smoothing parameters”: α, β and γ respectively.
- Need to choose best values for the smoothing parameters (and initial states).
- Equivalent ETS state space models developed in the 1990s and 2000s.

Big idea: control the rate of change

α controls the flexibility of the **level**

- If α = 0, the level never updates (mean)
- If α = 1, the level updates completely (naive)

β controls the flexibility of the **trend**

- If β = 0, the trend is linear
- If β = 1, the trend changes suddenly every observation

γ controls the flexibility of the **seasonality**

- If γ = 0, the seasonality is fixed (seasonal means)
- If γ = 1, the seasonality updates completely (seasonal naive)

A model for levels, trends, and seasonalities

We want a model that captures the level (ℓ_t), trend (b_t) and seasonality (s_t).

How do we combine these elements?

A model for levels, trends, and seasonalities

We want a model that captures the level (ℓ_t), trend (b_t) and seasonality (s_t).

How do we combine these elements?

Additively?

$$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$$

Multiplicatively?

$$y_t = \ell_{t-1} b_{t-1} s_{t-m} (1 + \varepsilon_t)$$

Perhaps a mix of both?

$$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m} + \varepsilon_t$$

A model for levels, trends, and seasonalities

We want a model that captures the level (ℓ_t), trend (b_t) and seasonality (s_t).

How do we combine these elements?

Additively?

$$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$$

Multiplicatively?

$$y_t = \ell_{t-1} b_{t-1} s_{t-m} (1 + \varepsilon_t)$$

Perhaps a mix of both?

$$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m} + \varepsilon_t$$

How do the level, trend and seasonal components evolve over time?

ETS models

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

↗ ↑ ↖

Error **T**rend **S**eason

Error: Additive (`"A"`) or multiplicative (`"M"`)

ETS models

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

↗ ↑ ↖

Error **T**rend **S**eason

Error: Additive (`"A"`) or multiplicative (`"M"`)

Trend: None (`"N"`), additive (`"A"`), multiplicative (`"M"`), or damped (`"Ad"` or `"Md"`).

ETS models

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

↗ ↑ ↖

Error **T**rend **S**eason

Error: Additive (`"A"`) or multiplicative (`"M"`)

Trend: None (`"N"`), additive (`"A"`), multiplicative (`"M"`), or damped (`"Ad"` or `"Md"`).

Seasonality: None (`"N"`), additive (`"A"`) or multiplicative (`"M"`)

Outline

Simple methods

Time series $y_1, y_2, \ldots, y_T$.

Random walk forecasts

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

Simple methods

Time series $y_1, y_2, \ldots, y_T$.

Random walk forecasts

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

Average forecasts

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

Simple methods

Time series $y_1, y_2, \ldots, y_T$.

Random walk forecasts

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

Average forecasts

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

- Want something in between these methods.
- Most recent data should have more weight.

Simple Exponential Smoothing

Forecast equation

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha) y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots,$$

where $0 \le \alpha \le 1$.

Simple Exponential Smoothing

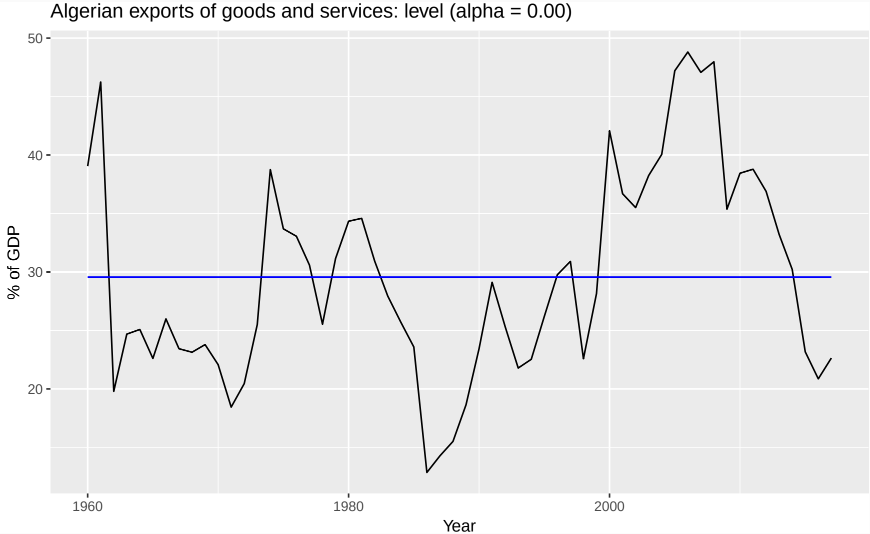
Forecast equation

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots,$$

where $0 \leq \alpha \leq 1$.

	Weights assigned to observations for:			
Observation	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_T	0.2	0.4	0.6	0.8
y_{T-1}	0.16	0.24	0.24	0.16
y_{T-2}	0.128	0.144	0.096	0.032
y_{T-3}	0.1024	0.0864	0.0384	0.0064
y_{T-4}	$(0.2)(0.8)^4$	$(0.4)(0.6)^4$	$(0.6)(0.4)^4$	$(0.8)(0.2)^4$
y_{T-5}	$(0.2)(0.8)^5$	$(0.4)(0.6)^5$	$(0.6)(0.4)^5$	$(0.8)(0.2)^5$

Simple Exponential Smoothing



Simple Exponential Smoothing

Component form

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

- ℓ_t is the level (or the smoothed value) of the series at time t.
- $\hat{y}_{t+1|t} = \alpha y_t + (1 - \alpha)\hat{y}_{t|t-1}$

Simple Exponential Smoothing

Component form

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

- ℓ_t is the level (or the smoothed value) of the series at time t.
- $\hat{y}_{t+1|t} = \alpha y_t + (1 - \alpha)\hat{y}_{t|t-1}$

Iterate to get exponentially weighted moving average form.

Weighted average form

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1 - \alpha)^j y_{T-j} + (1 - \alpha)^T \ell_0$$

Optimising smoothing parameters

- Need to choose best values for α and ℓ_0.
- Similarly to regression, choose optimal parameters by minimising SSE:

$$\text{SSE} = \sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2.$$

- Unlike regression there is no closed form solution — use numerical optimization.

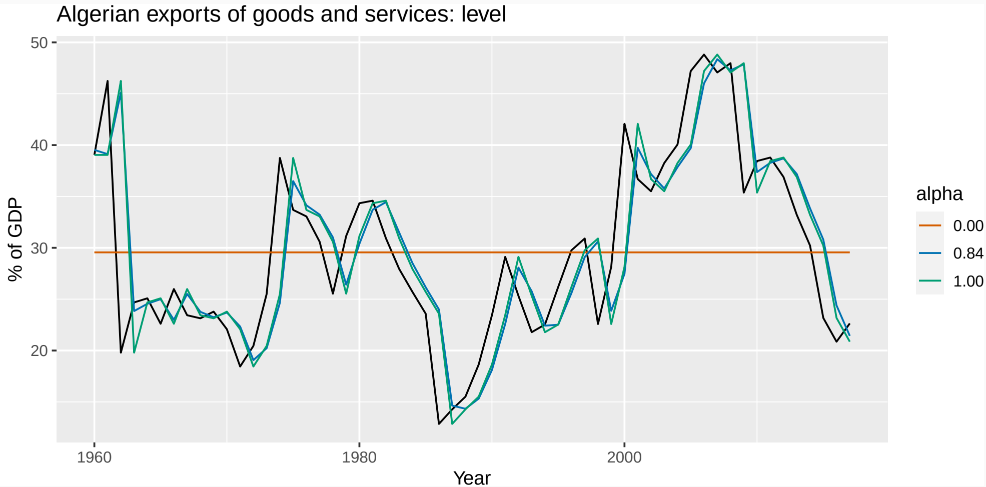
Optimising smoothing parameters

- Need to choose best values for α and ℓ_0.
- Similarly to regression, choose optimal parameters by minimising SSE:

$$\text{SSE} = \sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2.$$

- Unlike regression there is no closed form solution — use numerical optimization.
- For Algerian Exports example:
  - $\hat{\alpha} = 0.8400$
  - $\hat{\ell}_0 = 39.54$

Simple Exponential Smoothing



Models and methods

Methods

- Algorithms that return point forecasts.

Models

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

ETS(A,N,N): SES with additive errors

Component form

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

ETS(A,N,N): SES with additive errors

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

ETS(A,N,N): SES with additive errors

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

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

ETS(A,N,N): SES with additive errors

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

$$y_t = \ell_{t-1} + e_t$$

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

$$= \ell_{t-1} + \alpha e_t$$

Specify probability distribution for e_t, we assume $e_t = \varepsilon_t \sim \text{NID}(0, \sigma^2)$.

ETS(A,N,N): SES with additive errors

Measurement 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 equations have the same error process, ε_t.
- Measurement equation: relationship between observations and states.
- State equation(s): evolution of the state(s) through time.

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:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

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:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Measurement 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:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Measurement 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(A,N,N): Specifying the model

```
ETS(y ~ error("A") + trend("N") + season("N"))
```

By default, an optimal value for α and ℓ_0 is used.

α can be chosen manually in `trend()`.

```
trend("N", alpha = 0.5)
trend("N", alpha_range = c(0.2, 0.8))
```

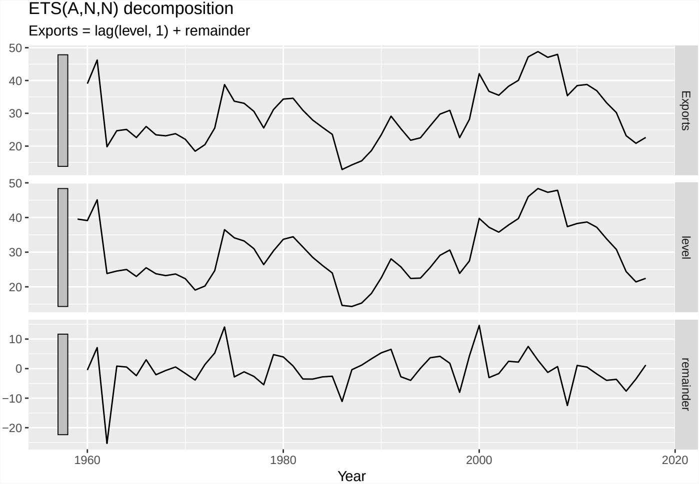
Example: Algerian Exports

```r
algeria_economy <- global_economy %>%
  filter(Country == "Algeria")
fit <- algeria_economy %>%
  model(ANN = ETS(Exports ~ error("A") + trend("N") + season("N")))
report(fit)
```

```
## Series: Exports
## Model: ETS(A,N,N)
##   Smoothing parameters:
##     alpha = 0.84
##
##   Initial states:
##  l[0]
##  39.5
##
##   sigma^2:  35.6
##
##  AIC AICc  BIC
```

Example: Algerian Exports

```
components(fit) %>% autoplot()
```



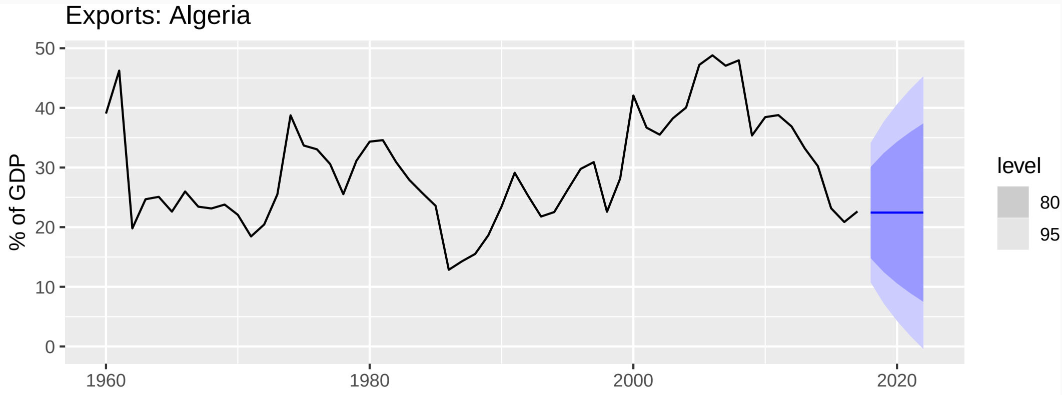
Example: Algerian Exports

```r
components(fit) %>%
  left_join(fitted(fit), by = c("Country", ".model", "Year"))
```

```
## # A dable: 59 x 7 [1Y]
## # Key:     Country, .model [1]
## # :        Exports = lag(level, 1) + remainder
##    Country .model  Year Exports level remainder .fitted
##    <fct>   <chr>  <dbl>   <dbl> <dbl>     <dbl>   <dbl>
##  1 Algeria ANN     1959    NA    39.5    NA        NA
##  2 Algeria ANN     1960    39.0  39.1    -0.496    39.5
##  3 Algeria ANN     1961    46.2  45.1     7.12     39.1
##  4 Algeria ANN     1962    19.8  23.8   -25.3      45.1
##  5 Algeria ANN     1963    24.7  24.6     0.841    23.8
##  6 Algeria ANN     1964    25.1  25.0     0.534    24.6
##  7 Algeria ANN     1965    22.6  23.0    -2.39     25.0
##  8 Algeria ANN     1966    26.0  25.5     3.00     23.0
##  9 Algeria ANN     1967    23.4  23.8    -2.07     25.5
## 10 Algeria ANN     1968    23.1  23.2    -0.630    23.8
```

Example: Algerian Exports

```
fit %>%
  forecast(h = 5) %>%
  autoplot(algeria_economy) +
  labs(y = "% of GDP", title = "Exports: Algeria")
```



Outline

Holt's linear trend

Component form

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

Holt's linear trend

Component form

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

- Two smoothing parameters α and β^* ($0 \leq \alpha, \beta^* \leq 1$).
- ℓ_t level: weighted average between y_t and one-step ahead forecast for time t, $(\ell_{t-1} + b_{t-1} = \hat{y}_{t|t-1})$
- b_t slope: weighted average of $(\ell_t - \ell_{t-1})$ and b_{t-1}, current and previous estimate of slope.
- Choose $\alpha, \beta^*, \ell_0, b_0$ to minimise SSE.

ETS(A,A,N)

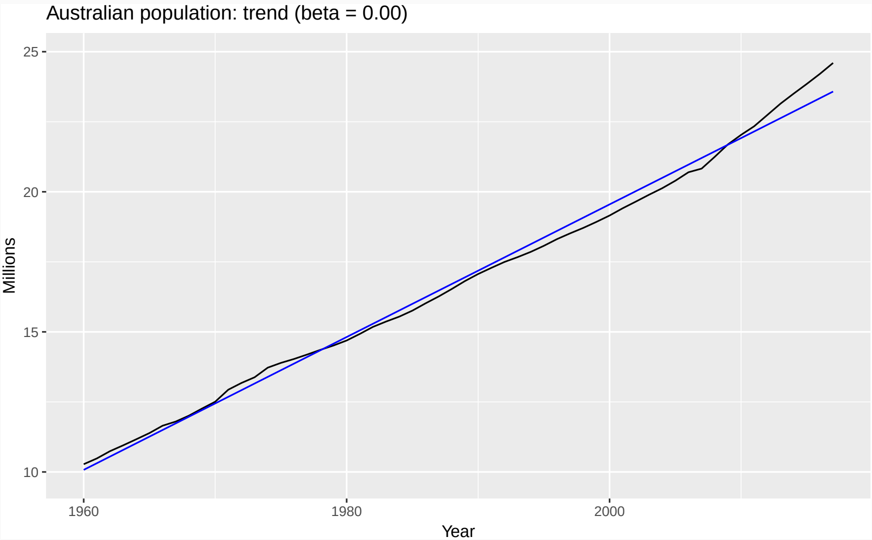
Holt's linear method with additive errors.

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

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

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

Exponential smoothing: trend/slope



ETS(M,A,N)

Holt's linear method with multiplicative errors.

- Assume $\varepsilon_t = \frac{y_t-(\ell_{t-1}+b_{t-1})}{(\ell_{t-1}+b_{t-1})}$
- Following a similar approach as above, the innovations state space model underlying Holt's linear method with multiplicative errors is specified as

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

where again $\beta = \alpha\beta^*$ and $\varepsilon_t \sim \text{NID}(0, \sigma^2)$.

ETS(A,A,N): Specifying the model

```
ETS(y ~ error("A") + trend("A") + season("N"))
```

By default, optimal values for β and b_0 are used.

β can be chosen manually in `trend()`.

```
trend("A", beta = 0.004)
trend("A", beta_range = c(0, 0.1))
```

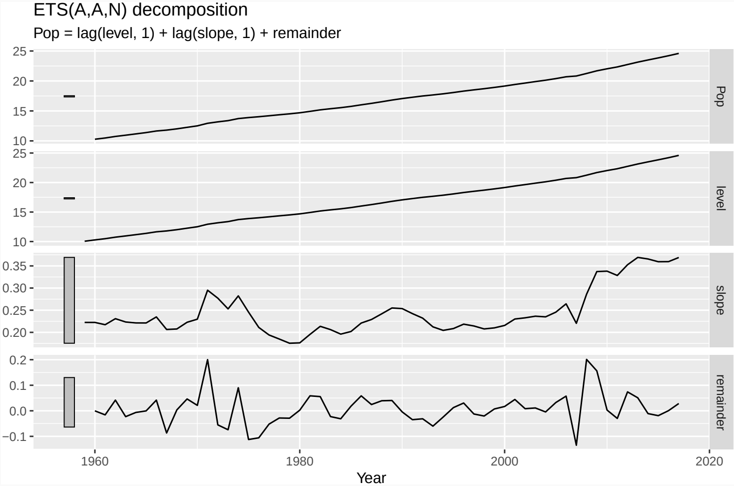
Example: Australian population

```r
aus_economy <- global_economy %>% filter(Code == "AUS") %>%
  mutate(Pop = Population / 1e6)
fit <- aus_economy %>%
  model(AAN = ETS(Pop ~ error("A") + trend("A") + season("N")))
report(fit)
```

```
## Series: Pop
## Model: ETS(A,A,N)
##   Smoothing parameters:
##     alpha = 1
##     beta  = 0.327
##
##   Initial states:
##  l[0]  b[0]
##  10.1 0.222
##
##   sigma^2:  0.0041
##
```

Example: Australian population

```
components(fit) %>% autoplot()
```



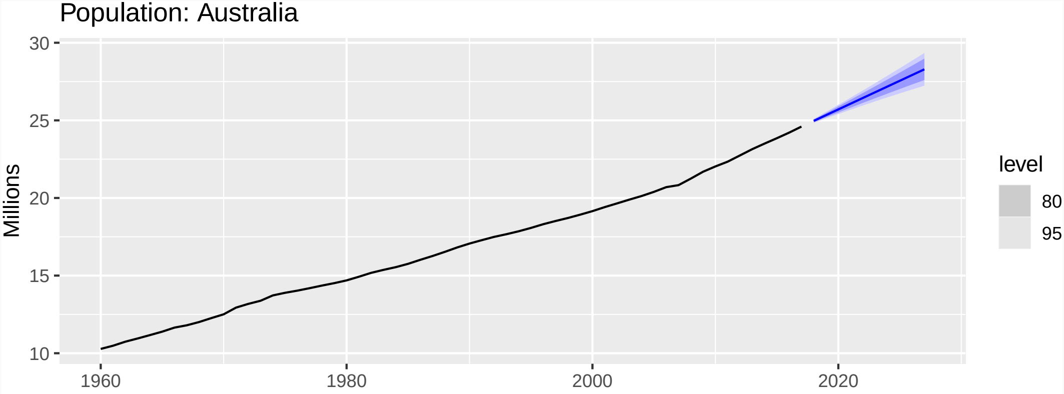
Example: Australian population

```
components(fit) %>%
  left_join(fitted(fit), by = c("Country", ".model", "Year"))
```

```
## # A dable: 59 x 8 [1Y]
## # Key:     Country, .model [1]
## # :        Pop = lag(level, 1) + lag(slope, 1) + remainder
##    Country   .model  Year   Pop level slope remainder .fitted
##    <fct>     <chr>  <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl>
##  1 Australia AAN     1959  NA    10.1 0.222 NA          NA
##  2 Australia AAN     1960  10.3  10.3 0.222 -0.000145   10.3
##  3 Australia AAN     1961  10.5  10.5 0.217 -0.0159     10.5
##  4 Australia AAN     1962  10.7  10.7 0.231  0.0418     10.7
##  5 Australia AAN     1963  11.0  11.0 0.223 -0.0229     11.0
##  6 Australia AAN     1964  11.2  11.2 0.221 -0.00641    11.2
##  7 Australia AAN     1965  11.4  11.4 0.221 -0.000314   11.4
##  8 Australia AAN     1966  11.7  11.7 0.235  0.0418     11.6
##  9 Australia AAN     1967  11.8  11.8 0.206 -0.0869     11.9
## 10 Australia AAN     1968  12.0  12.0 0.208  0.00350    12.0
```

Example: Australian population

```
fit %>%
  forecast(h = 10) %>%
  autoplot(aus_economy) +
  labs(y = "Millions", title = "Population: Australia")
```



Damped trend method

Component form

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

Damped trend method

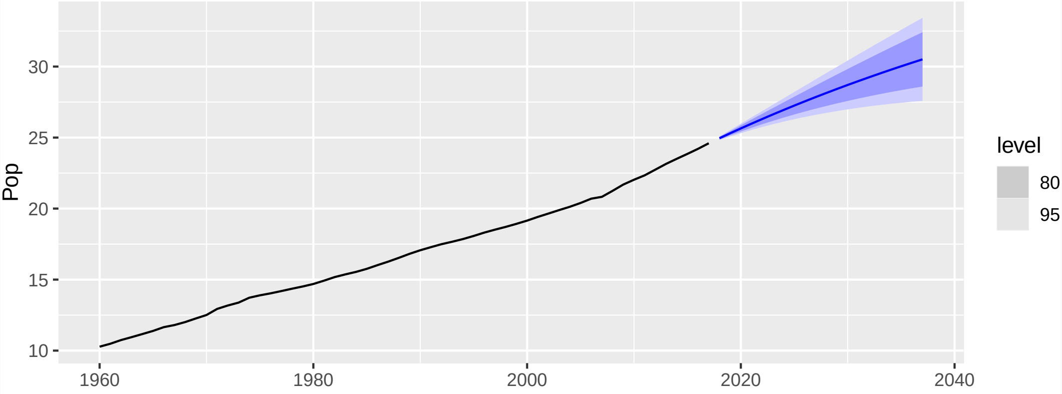
Component form

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

- Damping parameter $0 < \phi < 1$.
- If $\phi = 1$, identical to Holt's linear trend.
- As $h \to \infty$, $\hat{y}_{T+h|T} \to \ell_T + \phi b_T/(1 - \phi)$.
- Short-run forecasts trended, long-run forecasts constant.

Example: Australian population

```
aus_economy %>%
  model(holt = ETS(Pop ~ error("A") + trend("Ad") + season("N"))) %>%
  forecast(h = 20) %>%
  autoplot(aus_economy)
```



Example: Australian population

```r
fit <- aus_economy %>%
  filter(Year <= 2010) %>%
  model(
    ses = ETS(Pop ~ error("A") + trend("N") + season("N")),
    holt = ETS(Pop ~ error("A") + trend("A") + season("N")),
    damped = ETS(Pop ~ error("A") + trend("Ad") + season("N"))
  )
```

```r
tidy(fit)
accuracy(fit)
```

Example: Australian population

term	SES	Linear trend	Damped trend
α	1.00	1.00	1.00
β^*		0.30	0.40
ϕ			0.98
NA		0.22	0.25
NA	10.28	10.05	10.04
Training RMSE	0.24	0.06	0.07
Test RMSE	1.63	0.15	0.21
Test MASE	6.18	0.55	0.75
Test MAPE	6.09	0.55	0.74
Test MAE	1.45	0.13	0.18

Outline

Holt-Winters additive method

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

Component form

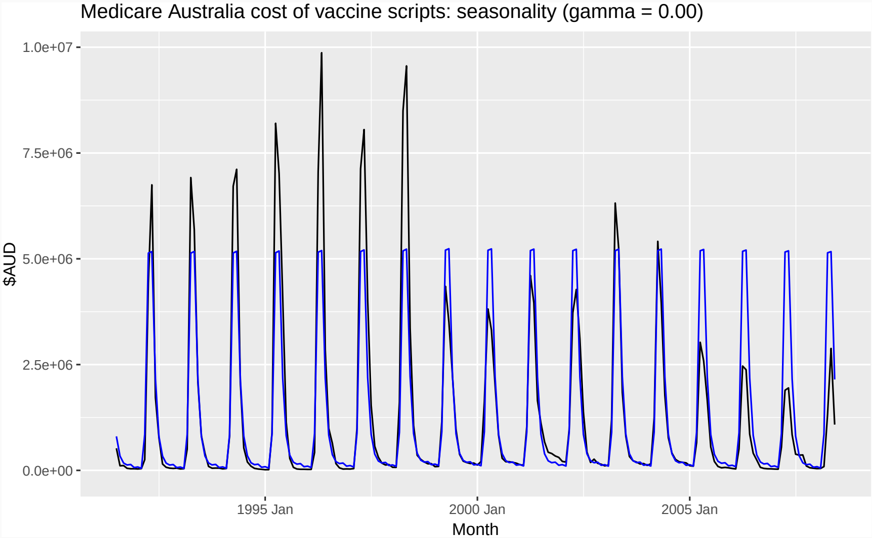
$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}
\end{aligned}$$

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

Holt-Winters additive method

- Seasonal component is usually expressed as
  $s_t = \gamma^*(y_t - \ell_t) + (1 - \gamma^*)s_{t-m}.$
- Substitute in for ℓ_t:
  $s_t = \gamma^*(1 - \alpha)(y_t - \ell_{t-1} - b_{t-1}) + [1 - \gamma^*(1 - \alpha)]s_{t-m}$
- We set $\gamma = \gamma^*(1 - \alpha)$.
- The usual parameter restriction is $0 \le \gamma^* \le 1$, which translates to $0 \le \gamma \le (1 - \alpha)$.

Exponential smoothing: seasonality



ETS(A,A,A)

Holt-Winters additive method with additive errors.

Forecast equation	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$
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: $\varepsilon_t = y_t - \hat{y}_{t|t-1}$
- k is integer part of $(h-1)/m$.

Holt-Winters multiplicative method

Seasonal variations change in proportion to the level of the series.

Component form

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

- k is integer part of $(h-1)/m$.
- Additive method: s_t in absolute terms — within each year $\sum_i s_i \approx 0$.
- Multiplicative method: s_t in relative terms — within each year $\sum_i s_i \approx m$.

ETS(M,A,M)

Holt-Winters multiplicative method with multiplicative errors.

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

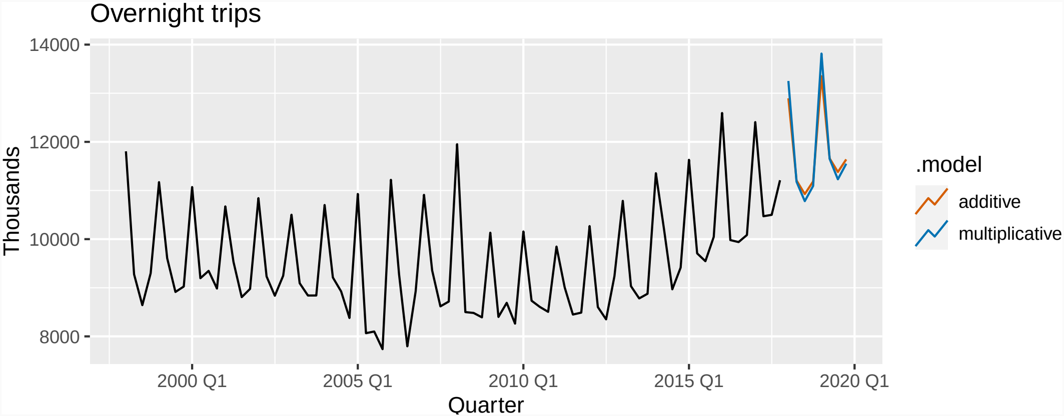
- Forecast errors: $\varepsilon_t = (y_t - \hat{y}_{t|t-1})/\hat{y}_{t|t-1}$
- k is integer part of $(h-1)/m$.

Example: Australian holiday tourism

```r
aus_holidays <- tourism %>%
  filter(Purpose == "Holiday") %>%
  summarise(Trips = sum(Trips))
fit <- aus_holidays %>%
  model(
    additive = ETS(Trips ~ error("A") + trend("A") + season("A")),
    multiplicative = ETS(Trips ~ error("M") + trend("A") + season("M"))
  )
fc <- fit %>% forecast()
```

Example: Australian holiday tourism

```r
fc %>%
  autoplot(aus_holidays, level = NULL) +
  labs(y = "Thousands", title = "Overnight trips")
```

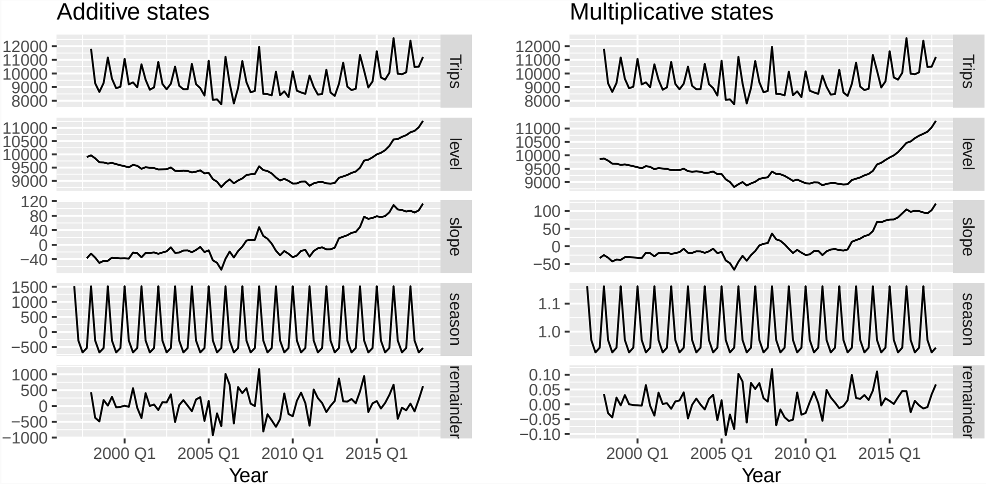


Estimated components

```
components(fit)
```

```
## # A dable: 168 x 7 [1Q]
## # Key:      .model [2]
## # :         Trips = lag(level, 1) + lag(slope, 1) + lag(season, 4) +
## #   remainder
##    .model   Quarter  Trips level slope season remainder
##    <chr>      <qtr>  <dbl> <dbl> <dbl>  <dbl>     <dbl>
##  1 additive 1997 Q1    NA    NA    NA    1512.       NA
##  2 additive 1997 Q2    NA    NA    NA    -290.       NA
##  3 additive 1997 Q3    NA    NA    NA    -684.       NA
##  4 additive 1997 Q4    NA  9899. -37.4  -538.       NA
##  5 additive 1998 Q1 11806. 9964. -24.5  1512.      433.
##  6 additive 1998 Q2  9276. 9851. -35.6  -290.     -374.
##  7 additive 1998 Q3  8642. 9700. -50.2  -684.     -489.
##  8 additive 1998 Q4  9300. 9694. -44.6  -538.      188.
```

Estimated components



Holt-Winters damped method

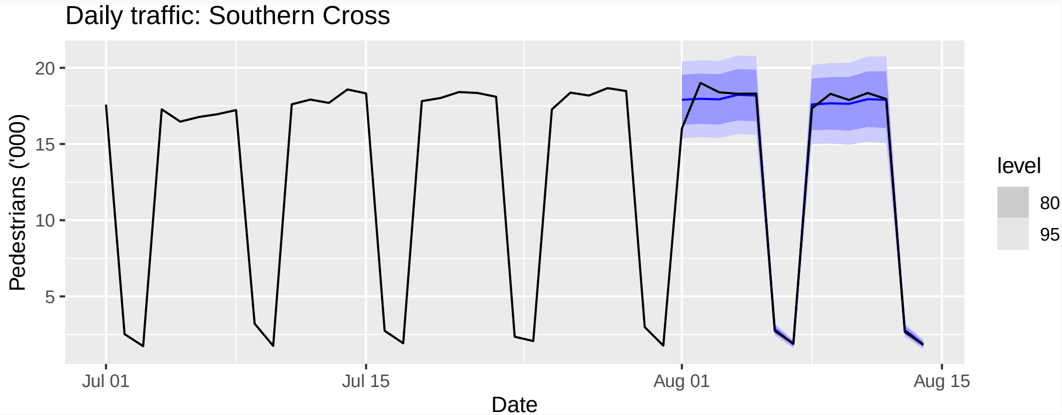
Often the single most accurate forecasting method for seasonal data:

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

Holt-Winters with daily data

```
sth_cross_ped <- pedestrian %>%
  filter(
    Date >= "2016-07-01",
    Sensor == "Southern Cross Station"
  ) %>%
  index_by(Date) %>%
  summarise(Count = sum(Count) / 1000)
sth_cross_ped %>%
  filter(Date <= "2016-07-31") %>%
  model(
    hw = ETS(Count ~ error("M") + trend("Ad") + season("M"))
  ) %>%
  forecast(h = "2 weeks") %>%
  autoplot(sth_cross_ped %>% filter(Date <= "2016-08-14")) +
  labs(
    title = "Daily traffic: Southern Cross",
    y = "Pedestrians ('000)"
  )
```

Holt-Winters with daily data



Outline

Exponential smoothing methods

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

(N,N): Simple exponential smoothing
(A,N): Holt's linear method
(A_d,N): Additive damped trend method
(A,A): Additive Holt-Winters' method
(A,M): Multiplicative Holt-Winters' method
(A_d,M): Damped multiplicative Holt-Winters' method

Exponential smoothing methods

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

(N,N): Simple exponential smoothing

(A,N): Holt's linear method

(A_d,N): Additive damped trend method

(A,A): Additive Holt-Winters' method

(A,M): Multiplicative Holt-Winters' method

(A_d,M): Damped multiplicative Holt-Winters' method

There are also multiplicative trend methods (not recommended).

ETS 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

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	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t / s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t / (\ell_{t-1} + b_{t-1})$
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 ℓ_0, b_0, $s_0, s_{-1}, \dots, s_{-m+1}$ are estimated by maximising the "likelihood" = the probability of the data arising from the specified model.
- For models with additive errors equivalent to minimising SSE.
- For models with multiplicative errors, **not** equivalent to minimising SSE.

Innovations state space models

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

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

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

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

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 $0 < \alpha < 1, \quad 0 < \beta < \alpha \quad$ 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 $0 < \alpha < 1, \quad 0 < \beta < \alpha \quad$ 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

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

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

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

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

Model selection

Akaike's Information Criterion

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

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

Corrected AIC

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

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.

Example: National populations

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

```
## # A mable: 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)>
```

Example: National populations

```r
fit %>%
  forecast(h = 5)
```

```
## # A fable: 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

```r
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)>
```

Example: Australian holiday tourism

```r
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
```

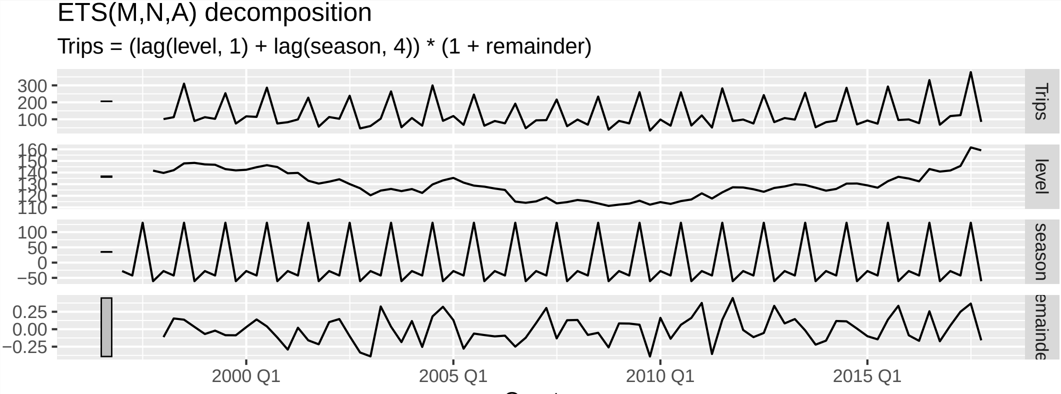
Example: Australian holiday tourism

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

```
## # A dable: 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 remainder
##    <chr>           <chr> <chr>   <chr>    <qtr> <dbl> <dbl>  <dbl>     <dbl>
##  1 Snowy Mountai~ New ~ Holiday ets     1997 Q1   NA     NA  -27.7    NA
##  2 Snowy Mountai~ New ~ Holiday ets     1997 Q2   NA     NA  -42.2    NA
##  3 Snowy Mountai~ New ~ Holiday ets     1997 Q3   NA     NA  131.     NA
##  4 Snowy Mountai~ New ~ Holiday ets     1997 Q4   NA   142.  -61.0    NA
##  5 Snowy Mountai~ New ~ Holiday ets     1998 Q1 101.   140.  -27.7    -0.113
##  6 Snowy Mountai~ New ~ Holiday ets     1998 Q2 112.   142.  -42.2     0.154
##  7 Snowy Mountai~ New ~ Holiday ets     1998 Q3 310.   148.  131.      0.137
##  8 Snowy Mountai~ New ~ Holiday ets     1998 Q4  89.8  148.  -61.0     0.0335
##  9 Snowy Mountai~ New ~ Holiday ets     1999 Q1 112.   147.  -27.7    -0.0687
```

Example: Australian holiday tourism

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



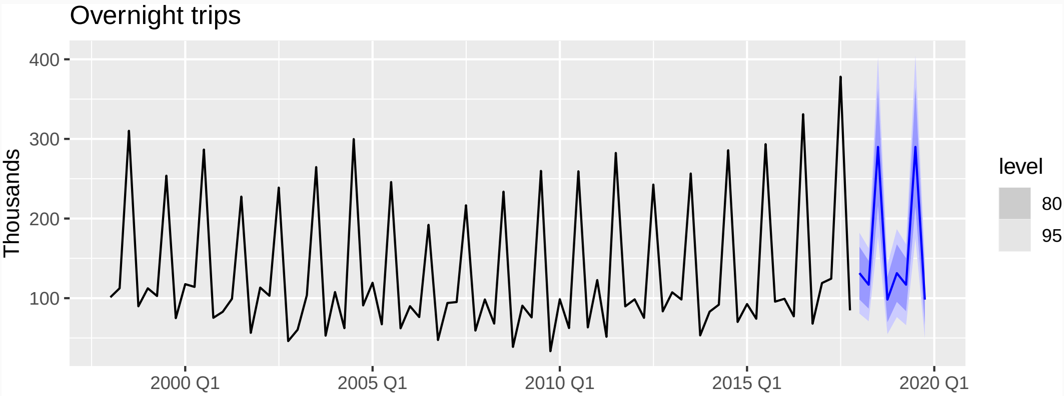
Example: Australian holiday tourism

```
fit %>% forecast()
```

```
## # A fable: 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:

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

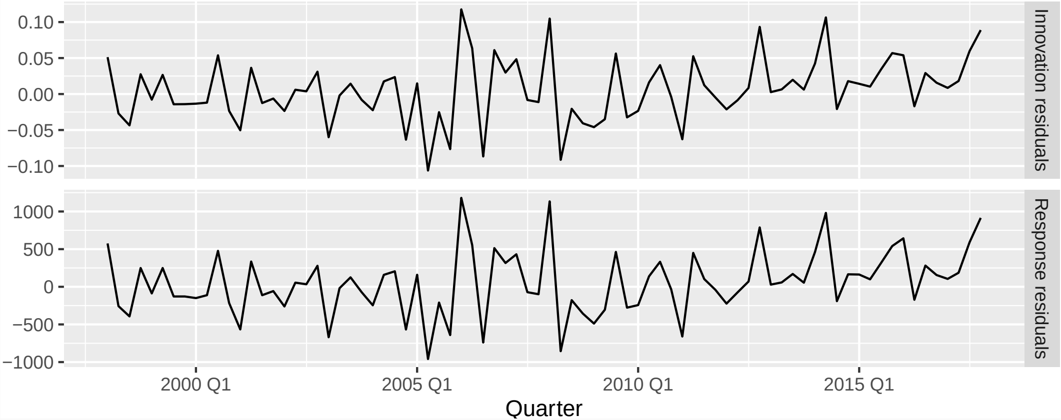
Example: Australian holiday tourism

```
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
```

Example: Australian holiday tourism

```r
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
```

Some unstable models

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

Exponential smoothing models

Additive Error

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

Multiplicative Error

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

Outline

Forecasting with ETS models

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

Forecasting with ETS models

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

- Not the same as $\text{E}(y_{t+h}|\boldsymbol{x}_t)$ unless seasonality is additive.
- `fable` uses $\text{E}(y_{t+h}|\boldsymbol{x}_t)$.
- Point forecasts for ETS(A,*,*) are identical to ETS(M,*,*) if the parameters are the same.

Example: ETS(A,A,N)

$$
\begin{aligned}
y_{T+1} &= \ell_T + b_T + \varepsilon_{T+1} \\
\hat{y}_{T+1|T} &= \ell_T + b_T \\
y_{T+2} &= \ell_{T+1} + b_{T+1} + \varepsilon_{T+2} \\
&= (\ell_T + b_T + \alpha\varepsilon_{T+1}) + (b_T + \beta\varepsilon_{T+1}) + \varepsilon_{T+2} \\
\hat{y}_{T+2|T} &= \ell_T + 2b_T
\end{aligned}
$$

etc.

Example: ETS(M,A,N)

$$\begin{aligned}
y_{T+1} &= (\ell_T + b_T)(1 + \varepsilon_{T+1}) \\
\hat{y}_{T+1|T} &= \ell_T + b_T. \\
y_{T+2} &= (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+2}) \\
&= \left\{(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + [b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}]\right\}(1 + \varepsilon_{T+2}) \\
\hat{y}_{T+2|T} &= \ell_T + 2b_T
\end{aligned}$$

etc.

Forecasting with ETS models

Prediction intervals: can only be generated using 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) $\quad \sigma_h = \sigma^2\Big[1 + \alpha^2(h-1)\Big]$

(A,A,N) $\quad \sigma_h = \sigma^2\Big[1 + (h-1)\{\alpha^2 + \alpha\beta h + \frac{1}{6}\beta^2 h(2h-1)\}\Big]$

(A,A$_d$,N) $\quad \sigma_h = \sigma^2\Big[1 + \alpha^2(h-1) + \frac{\beta\phi h}{(1-\phi)^2}\{2\alpha(1-\phi) + \beta\phi\} - \frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)}\Big\{2\alpha(1-\phi^2) + \beta\phi(1+2\phi-\phi^h)\Big\}\Big]$

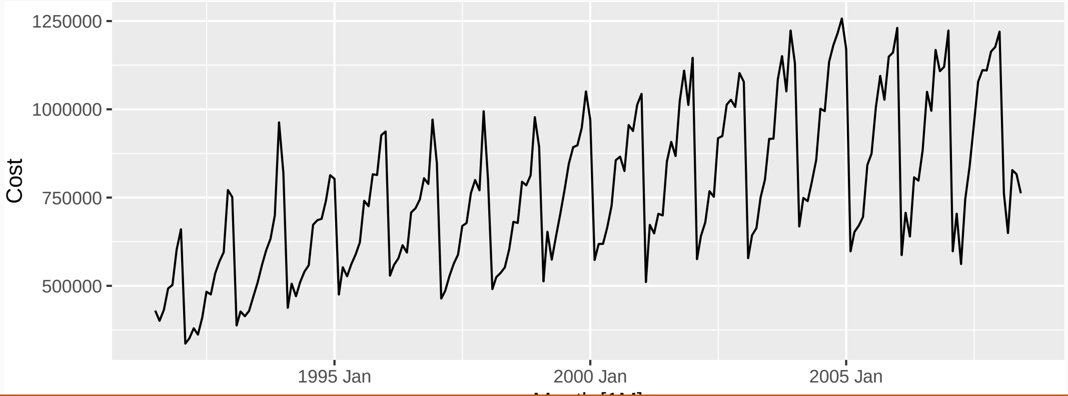
(A,N,A) $\quad \sigma_h = \sigma^2\Big[1 + \alpha^2(h-1) + \gamma k(2\alpha + \gamma)\Big]$

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

(A,A$_d$,A) $\quad \sigma_h = \sigma^2\Big[1 + \alpha^2(h-1) + \frac{\beta\phi h}{(1-\phi)^2}\{2\alpha(1-\phi) + \beta\phi\} - \frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)}\Big\{2\alpha(1-\phi^2) + \beta\phi(1+2\phi-\phi^h)\Big\}$
$\quad + \gamma k(2\alpha + \gamma) + \frac{2\beta\gamma\phi}{(1-\phi)(1-\phi^m)}\Big\{k(1-\phi^m) - \phi^m(1-\phi^{mk})\Big\}\Big]$

Example: Corticosteroid drug sales

```r
h02 <- PBS %>%
  filter(ATC2 == "H02") %>%
  summarise(Cost = sum(Cost))
h02 %>% autoplot(Cost)
```



Example: Corticosteroid drug sales

```
h02 %>%
  model(ETS(Cost)) %>%
  report()
```

```
## Series: Cost
## Model: ETS(M,Ad,M)
##   Smoothing parameters:
##     alpha = 0.307
##     beta  = 0.000101
##     gamma = 0.000101
##     phi   = 0.978
##
##   Initial states:
##    l[0] b[0]  s[0] s[-1] s[-2] s[-3] s[-4] s[-5] s[-6] s[-7] s[-8] s[-9]
##  417269 8206 0.872 0.826 0.756 0.773 0.687  1.28  1.32  1.18  1.16   1.1
##  s[-10] s[-11]
##    1.05  0.981
##
##   sigma^2:  0.0046
##
##  AIC AICc  BIC
## 5515 5519 5575
```

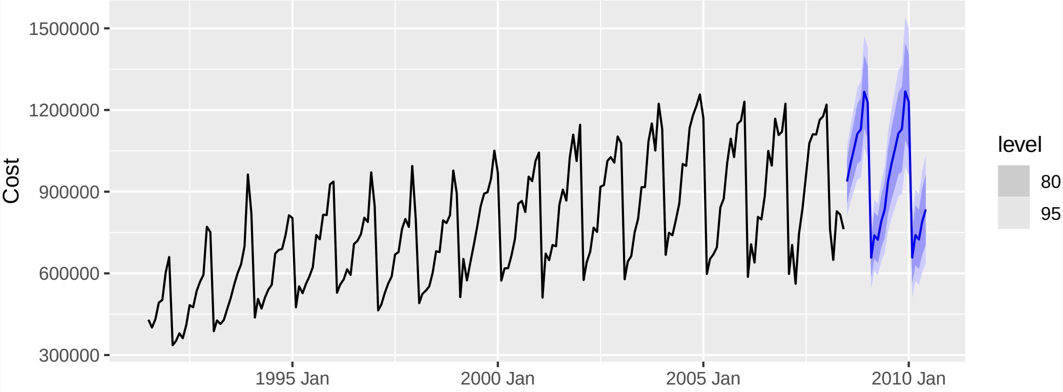
Example: Corticosteroid drug sales

```
h02 %>%
  model(ETS(Cost ~ error("A") + trend("A") + season("A"))) %>%
  report()
```

```
## Series: Cost
## Model: ETS(A,A,A)
##   Smoothing parameters:
##     alpha = 0.17
##     beta  = 0.00631
##     gamma = 0.455
##
##   Initial states:
##     l[0] b[0]   s[0]   s[-1]   s[-2]   s[-3]   s[-4]  s[-5]  s[-6]  s[-7]
##   409706 9097 -99075 -136602 -191496 -174531 -241437 210644 244644 145368
##   s[-8] s[-9] s[-10] s[-11]
##   130570 84458  39132 -11674
##
##   sigma^2:  3.5e+09
##
##   AIC AICc  BIC
## 5585 5589 5642
```

Example: Corticosteroid drug sales

```
h02 %>%
  model(ETS(Cost)) %>%
  forecast() %>%
  autoplot(h02)
```



Example: Corticosteroid drug sales

```
h02 %>%
  model(
    auto = ETS(Cost),
    AAA = ETS(Cost ~ error("A") + trend("A") + season("A"))
  ) %>%
  accuracy()
```

Model	MAE	RMSE	MAPE	MASE	RMSSE
auto	38649	51102	4.99	0.638	0.689
AAA	43378	56784	6.05	0.716	0.766