

# **ETC3550/ETC5550**

## **Applied forecasting**

Ch8. Exponential smoothing

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

# Outline

- 1 Exponential smoothing
- 2 Simple exponential smoothing
- 3 Models with trend
- 4 Models with seasonality
- 5 Innovations state space models
- 6 Forecasting with exponential smoothing

# Outline

- 1 Exponential smoothing
- 2 Simple exponential smoothing
- 3 Models with trend
- 4 Models with seasonality
- 5 Innovations state space models
- 6 Forecasting with exponential smoothing

# 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”:  
 $\alpha$ ,  $\beta$  and  $\gamma$  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

$\alpha$  controls the flexibility of the **level**

- If  $\alpha = 0$ , the level never updates (mean)
- If  $\alpha = 1$ , the level updates completely (naive)

$\beta$  controls the flexibility of the **trend**

- If  $\beta = 0$ , the trend is linear (regression trend)
- If  $\beta = 1$ , the trend updates every observation

$\gamma$  controls the flexibility of the **seasonality**

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

# A model for levels, trends, and seasonalities

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

How do we combine these elements?

**Additively?**

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

# A model for levels, trends, and seasonalities

We want a model that captures the level ( $\ell_t$ ), 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?**

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



# A model for levels, trends, and seasonalities

We want a model that captures the level ( $\ell_t$ ), 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 ( $\ell_t$ ), 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

ETS : ExponenTial Smoothing




Error Trend Season

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

# ETS models

General notation

ETS : Exponential Smoothing  
Error Trend Season




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

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

# ETS models

General notation

ETS : ExponenTial Smoothing



Error Trend Season

The diagram shows three arrows pointing from the words 'Error', 'Trend', and 'Season' below to the letters 'E', 'T', and 'S' respectively in the 'ETS' part of the text above.

**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

- 1 Exponential smoothing
- 2 Simple exponential smoothing
- 3 Models with trend
- 4 Models with seasonality
- 5 Innovations state space models
- 6 Forecasting with exponential smoothing

# Simple methods

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

## Random walk forecasts

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

# Simple methods

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

## Random walk forecasts

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

## Average forecasts

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



# Simple methods

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

## Random walk forecasts

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

## Average forecasts

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

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

# Simple Exponential Smoothing

## Forecast equation

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

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

# Simple Exponential Smoothing

## Forecast equation

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

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

Observation	Weights assigned to observations for:			
	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
$y_T$	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  $\ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1}$

- $\ell_t$  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  $\alpha$  and  $\ell_0$ .
- 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  $\alpha$  and  $\ell_0$ .
- 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

```
alpha_static <- map_dfr(list(0, as.numeric(tidy(
  fit <- algeria_economy %>%
    model(ETS(Exports ~ error("A") + trend("N",
      beta_range = c(-1,1)) + season("N",
fit %>%
  augment() %>%
  mutate(alpha = tidy(fit)$estimate[tidy(fit)
    as_tibble()
})) %>%
  mutate(alpha = factor(format(alpha)))
algeria_economy %>%
```



# 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): A model for SES

## Component form

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

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

# ETS(A,N,N): A model for SES

## Component form

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

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

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

# ETS(A,N,N): A model for SES

## Component form

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

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

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

## Error correction form

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

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

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

# ETS(A,N,N): A model for SES

## Component form

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

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

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

## Error correction form

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

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

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

Specify probability distribution for  $e_t$ , we assume

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

# ETS(A,N,N)

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,  $\varepsilon_t$ .
- Measurement equation: relationship between observations and states.
- State equation(s): evolution of the state(s) through time.

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

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

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



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. 18

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

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

By default, an optimal value for  $\alpha$  and  $\ell_0$  is used.

$\alpha$  can be chosen manually in `trend()`.

```
trend("N", alpha = 0.5)
```

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

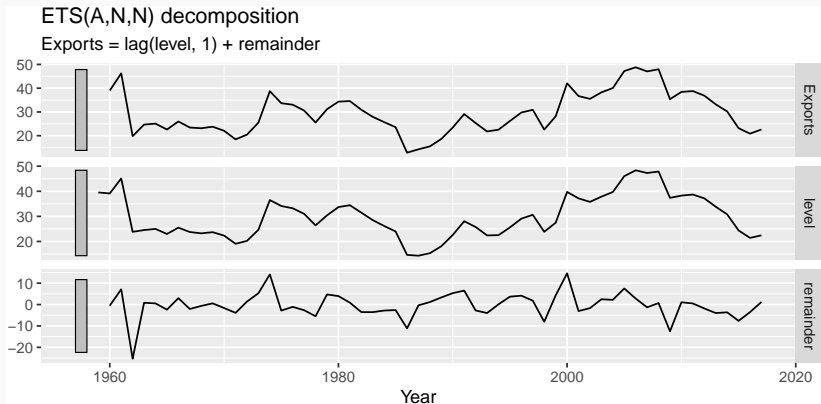
# Example: Algerian Exports

```
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  
## 39.5  
##  
## sigma^2: 35.6  
##  
## AIC AICc BIC  
## 447 447 453
```

# Example: Algerian Exports

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



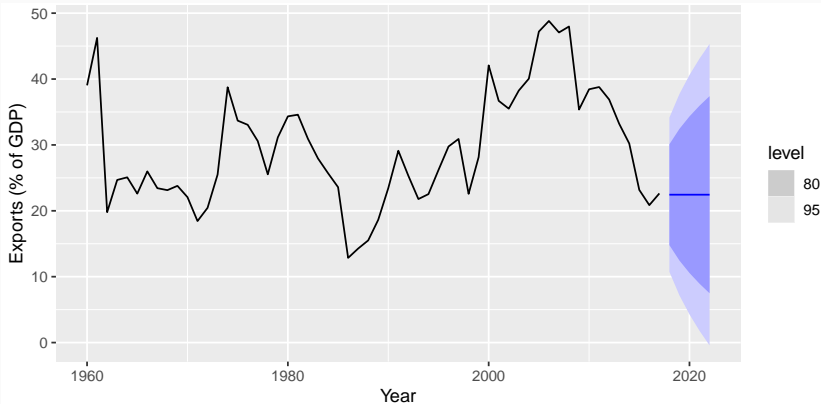
# Example: Algerian Exports

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

```
## # A dable:                59 x 7 [1Y]  
## # Key:                  Country, .model [1]  
## # ETS(A,N,N) Decomposition: 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
```

# Example: Algerian Exports

```
fit %>%  
  forecast(h = 5) %>%  
  autoplot(algeria_economy) +  
  ylab("Exports (% of GDP)") + xlab("Year")
```



# Outline

- 1 Exponential smoothing
- 2 Simple exponential smoothing
- 3 Models with trend**
- 4 Models with seasonality
- 5 Innovations state space models
- 6 Forecasting with exponential smoothing

# 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  $\alpha$  and  $\beta^*$  ( $0 \leq \alpha, \beta^* \leq 1$ ).
- $\ell_t$  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.

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

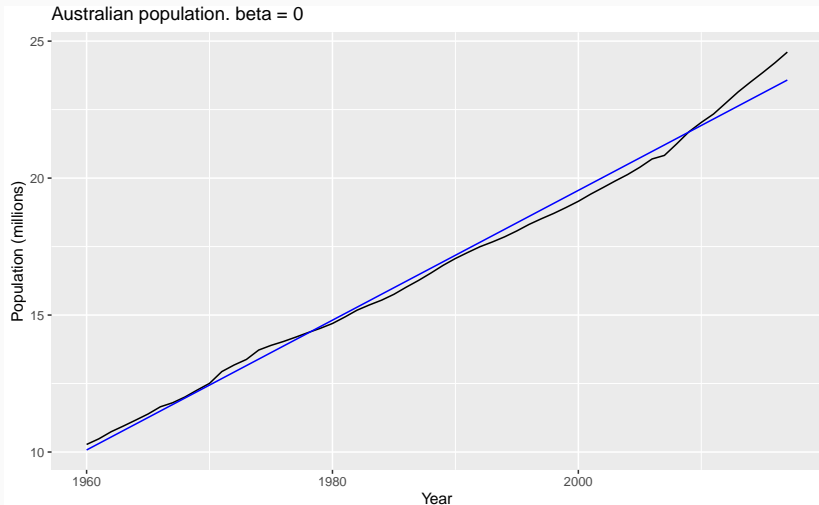
$$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$$

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

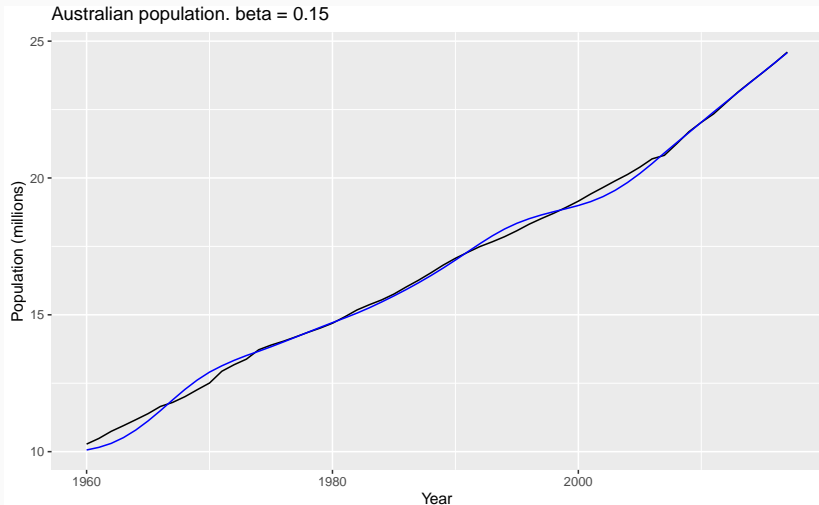
$$b_t = b_{t-1} + \alpha \beta^* \varepsilon_t$$

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

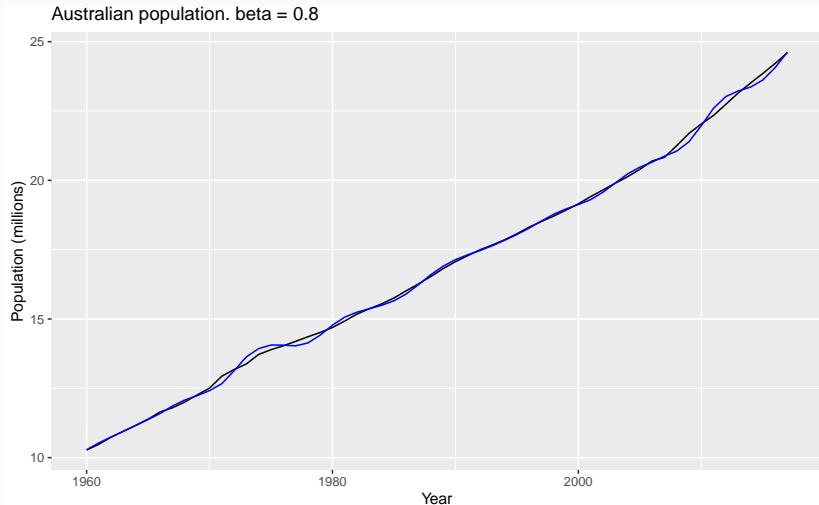
# Exponential smoothing: trend/slope



# Exponential smoothing: trend/slope



# Exponential smoothing: trend/slope



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

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

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

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

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  $\beta$  and  $b_0$  are used.

$\beta$  can be chosen manually in `trend()`.

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

# Example: Australian population

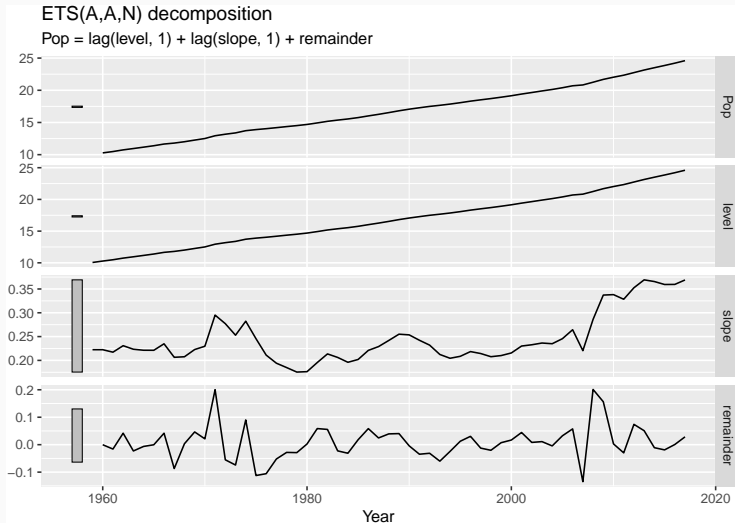
```
aus_economy <- global_economy %>% filter(Code == "AUS") %>%  
  mutate(Pop = Population/1e6)  
fit <- aus_economy %>%  
  model(AAN = ETS(Pop ~ error("A") + trend("A") + season("N")))  
report(fit)
```

```
## Series: Pop  
## Model: ETS(A,A,N)  
## Smoothing parameters:  
##   alpha = 1  
##   beta  = 0.327  
##  
## Initial states:  
##   l      b  
## 10.1 0.222  
##  
## sigma^2: 0.0041  
##  
## AIC AICc BIC  
## -77.0 -75.8 -66.7
```



# Example: Australian population

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



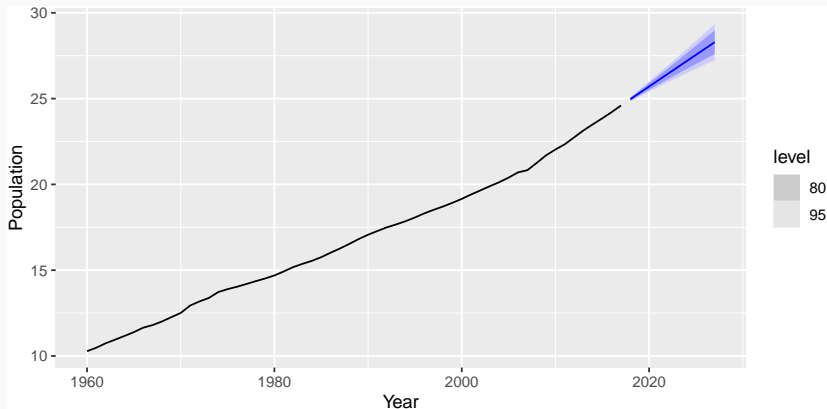
# Example: Australian population

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

```
## # A dable:                59 x 8 [1Y]  
## # Key:                    Country, .model [1]  
## # ETS(A,A,N) Decomposition: 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 Austral~ AAN    1959  NA    10.1 0.222 NA         NA  
## 2 Austral~ AAN    1960  10.3  10.3 0.222 -0.000145  10.3  
## 3 Austral~ AAN    1961  10.5  10.5 0.217 -0.0159    10.5  
## 4 Austral~ AAN    1962  10.7  10.7 0.231  0.0418    10.7  
## 5 Austral~ AAN    1963  11.0  11.0 0.223 -0.0229    11.0  
## 6 Austral~ AAN    1964  11.2  11.2 0.221 -0.00641   11.2  
## 7 Austral~ AAN    1965  11.4  11.4 0.221 -0.000314  11.4  
## 8 Austral~ AAN    1966  11.7  11.7 0.235  0.0418    11.6  
## 9 Austral~ AAN    1967  11.8  11.8 0.206 -0.0869    11.9  
## 10 Austral~ AAN   1968  12.0  12.0 0.208  0.00350   12.0
```

# Example: Australian population

```
fit %>%  
  forecast(h = 10) %>%  
  autoplot(aus_economy) +  
  ylab("Population") + xlab("Year")
```



# Damped trend method

## Component form

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \dots + \phi^h)b_t$$

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

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}.$$

# Damped trend method

## Component form

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \dots + \phi^h)b_t$$

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

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}.$$

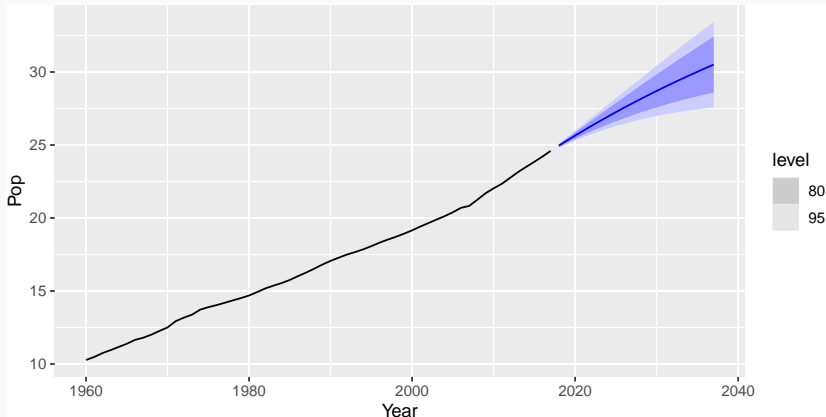
- Damping parameter  $0 < \phi < 1$ .
- If  $\phi = 1$ , identical to Holt's linear trend.
- As  $h \rightarrow \infty$ ,  $\hat{y}_{T+h|T} \rightarrow \ell_T + \phi b_T / (1 - \phi)$ .
- Short-run forecasts trended, long-run forecasts constant.

# Your turn

- Write down the model for  $ETS(A, A_d, N)$

# Example: Australian population

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



# Example: Australian population

```
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"))  
  )
```

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



## Example: Australian population

term	SES	Linear trend	Damped trend
$\alpha$	1.00	1.00	1.00
$\beta^*$		0.30	0.40
$\phi$			0.98
$\ell_0$	10.28	10.05	10.04
$b_0$		0.22	0.25
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

## Your turn

`fma::eggs` contains the price of a dozen eggs in the United States from 1900–1993

- 1 Use SES and Holt's method (with and without damping) to forecast "future" data.  
[Hint: use  $h=100$  so you can clearly see the differences between the options when plotting the forecasts.]
- 2 Which method gives the best training RMSE?
- 3 Are these RMSE values comparable?
- 4 Do the residuals from the best fitting method look like white noise?

# Outline

- 1 Exponential smoothing
- 2 Simple exponential smoothing
- 3 Models with trend
- 4 Models with seasonality
- 5 Innovations state space models
- 6 Forecasting with exponential smoothing

# Holt-Winters additive method

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

## Component form

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$$

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

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$$

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

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

# 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  $\ell_t$ :

$$s_t = \gamma^*(1 - \alpha)(y_t - \ell_{t-1} - b_{t-1}) + [1 - \gamma^*(1 - \alpha)]s_{t-m}$$

- We set  $\gamma = \gamma^*(1 - \alpha)$ .

- The usual parameter restriction is  $0 \leq \gamma^* \leq 1$ ,  
which translates to  $0 \leq \gamma \leq (1 - \alpha)$ .

# Exponential smoothing: seasonality

# Exponential smoothing: seasonality

```
gamma_static <- map_dfr(list(0, NULL, 1), func
  fit <- j07 %>%
    model(ETS(Cost ~ error("A") + trend("N", a
      beta_range = c(-1,1)) + season("
  fit %>%
    augment() %>%
    mutate(gamma = tidy(fit)$estimate[tidy(fit
    as_tibble()
}) %>%
  mutate(gamma = factor(format(gamma)))
j07 %>%
```

# 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  $(h - 1)/m$ .



# Your turn

- Write down the model for ETS(A,N,A)

# Holt-Winters multiplicative method

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

## Component form

$$\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$$

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

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$$

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

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

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

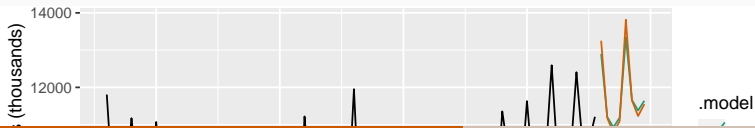
$$\begin{aligned}\ell_t &= (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t) \\ b_t &= b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t \\ s_t &= s_{t-m}(1 + \gamma\varepsilon_t)\end{aligned}$$

- 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

```
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()
```

```
fc %>%  
  autoplot(aus_holidays, level = NULL) + xlab("Year") +  
  ylab("Overnight trips (thousands)") +  
  scale_color_brewer(type = "qual", palette = "Dark2")
```



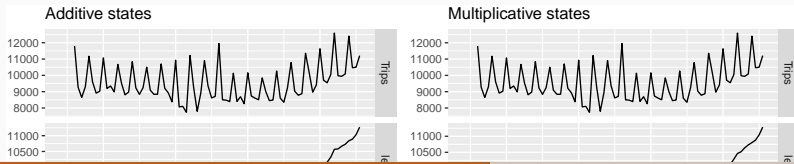
# Estimated components

```
components(fit)
```

```
## # A dable:                                168 x 7 [1Q]
## # Key:                                     .model [2]
## # ETS(A,A,A) & ETS(M,A,M) Decomposition: 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.
## 9 additive 1999 Q1 11172. 9652. -44.3  1512.    10.7
## 10 additive 1999 Q2  8608. 9676. -35.6  -290.   -374.
```

# Estimated components

```
components(fit) %>%  
  gather("state", "value", -.model, -Quarter, factor_key = TRUE) %>%  
  group_by(.model) %>%  
  group_split() %>%  
  purrr::map(  
    ~ ggplot(., aes(x = Quarter, y = value)) +  
      geom_line() +  
      facet_grid(state ~ ., scales = "free") +  
      xlab("Year") + ylab(NULL) +  
      ggtitle(stringr::str_to_title(unique(.$.model)) %>% paste("s"  
  ) %>%  
  wrap_plots(ncol=2)
```



# Holt-Winters damped method

Often the single most accurate forecasting method for seasonal data:

$$\hat{y}_{t+h|t} = [\ell_t + (\phi + \phi^2 + \dots + \phi^h)b_t]s_{t+h-m(k+1)}$$

$$\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$$

$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + \phi b_{t-1})} + (1 - \gamma)s_{t-m}$$

# Your turn

Apply Holt-Winters' multiplicative method to the Gas data from `aus_production`.

- 1 Why is multiplicative seasonality necessary here?
- 2 Experiment with making the trend damped.
- 3 Check that the residuals from the best method look like white noise.



# Outline

- 1 Exponential smoothing
- 2 Simple exponential smoothing
- 3 Models with trend
- 4 Models with seasonality
- 5 Innovations state space models
- 6 Forecasting with exponential smoothing

# Exponential smoothing methods

		Seasonal Component		
		N	A	M
Trend Component		(None)	(Additive)	(Multiplicative)
N	(None)	(N,N)	(N,A)	(N,M)
A	(Additive)	(A,N)	(A,A)	(A,M)
A <sub>d</sub>	(Additive damped)	(A <sub>d</sub> ,N)	(A <sub>d</sub> ,A)	(A <sub>d</sub> ,M)

(N,N): Simple exponential smoothing

(A,N): Holt's linear method

(A<sub>d</sub>,N): Additive damped trend method

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

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

(A<sub>d</sub>,M): Damped multiplicative Holt-Winters' method

# ETS models

## Additive Error

		Seasonal Component		
Trend Component		N	A	M
		(None)	(Additive)	(Multiplicative)
N	(None)	A,N,N	A,N,A	A,N,M
A	(Additive)	A,A,N	A,A,A	A,A,M
A <sub>d</sub>	(Additive damped)	A,A <sub>d</sub> ,N	A,A <sub>d</sub> ,A	A,A <sub>d</sub> ,M

## Multiplicative Error

		Seasonal Component		
Trend Component		N	A	M
		(None)	(Additive)	(Multiplicative)
N	(None)	M,N,N	M,N,A	M,N,M
A	(Additive)	M,A,N	M,A,A	M,A,M
A <sub>d</sub>	(Additive damped)	M,A <sub>d</sub> ,N	M,A <sub>d</sub> ,A	M,A <sub>d</sub> ,M

# Additive error models

Trend	Seasonal		
	N	A	M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = \ell_{t-1} s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / \ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$ $b_t = b_{t-1} + \beta \varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$ $b_t = b_{t-1} + \beta \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $b_t = b_{t-1} + \beta \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / (\ell_{t-1} + b_{t-1})$
A <sub>d</sub>	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t$ $b_t = \phi b_{t-1} + \beta \varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t$ $b_t = \phi b_{t-1} + \beta \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $b_t = \phi b_{t-1} + \beta \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / (\ell_{t-1} + \phi b_{t-1})$

# Multiplicative error models

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

# Estimating ETS models

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

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

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

**Additive errors**  $k(\mathbf{x}) = 1$ .  $y_t = \mu_t + \varepsilon_t$ .

**Multiplicative errors**  $k(\mathbf{x}_{t-1}) = \mu_t$ .  $y_t = \mu_t(1 + \varepsilon_t)$ .  
 $\varepsilon_t = (y_t - \mu_t)/\mu_t$  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 the *traditional* region.
- For example for ETS(A,N,N):  
*traditional*  $0 < \alpha < 1$  — *admissible* is  $0 < \alpha < 2$ .

# 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

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

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

# Model selection

## Akaike's Information Criterion

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

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

## Corrected AIC

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

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

## Bayesian Information Criterion

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

# 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)>  
## 10 Armenia    <ETS(M,A,N)>  
## # ... with 253 more rows
```



# 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) 2.8769
```

# 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 Austral~	Holiday	<ETS(A,N,A)>
## 2	Adelaide Hills	South Austral~	Holiday	<ETS(A,A,N)>
## 3	Alice Springs	Northern Terr~	Holiday	<ETS(M,N,A)>
## 4	Australia's Coral Co~	Western Austr~	Holiday	<ETS(M,N,A)>
## 5	Australia's Golden O~	Western Austr~	Holiday	<ETS(M,N,M)>
## 6	Australia's North We~	Western Austr~	Holiday	<ETS(A,N,A)>
## 7	Australia's South We~	Western Austr~	Holiday	<ETS(M,N,M)>
## 8	Ballarat	Victoria	Holiday	<ETS(M,N,A)>
## 9	Barkly	Northern Terr~	Holiday	<ETS(A,N,A)>
## 10	Barossa	South Austral~	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   s1  s2   s3   s4  
## 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)
```

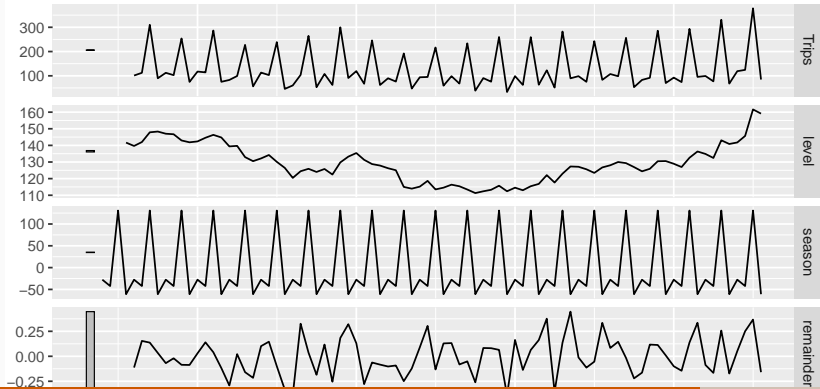
```
## # A dable:                84 x 9 [1Q]  
## # Key:                    Region, State, Purpose, .model  
## #   [1]  
## # ETS(M,N,A) Decomposition: Trips = (lag(level, 1) +  
## #   lag(season, 4)) * (1 + remainder)  
##   Region State Purpose .model Quarter Trips level season  
##   <chr>  <chr> <chr>   <chr>    <qtr> <dbl>  <dbl>  <dbl>  
## 1 Snowy~ New ~ Holiday ets    1997 Q1  NA      NA    -27.7  
## 2 Snowy~ New ~ Holiday ets    1997 Q2  NA      NA    -42.2  
## 3 Snowy~ New ~ Holiday ets    1997 Q3  NA      NA    131.  
## 4 Snowy~ New ~ Holiday ets    1997 Q4  NA     142.   -61.0  
## 5 Snowy~ New ~ Holiday ets    1998 Q1 101.    140.   -27.7  
## 6 Snowy~ New ~ Holiday ets    1998 Q2 112.    142.   -42.2  
## 7 Snowy~ New ~ Holiday ets    1998 Q3 310.    148.    131.  
## 8 Snowy~ New ~ Holiday ets    1998 Q4  89.8   148.   -61.0
```

# Example: Australian holiday tourism

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

ETS(M,N,A) decomposition

$\text{Trips} = (\text{lag}(\text{level}, 1) + \text{lag}(\text{season}, 4)) * (1 + \text{remainder})$



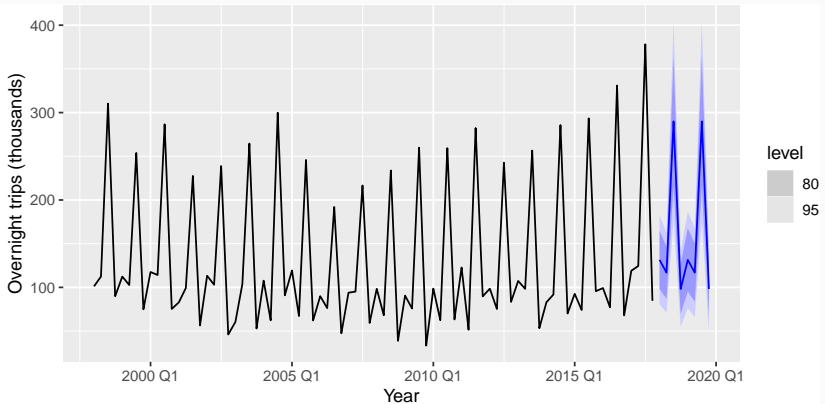
# Example: Australian holiday tourism

```
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 ~ Holiday ets    2018 Q1 N(210, 457) 210.
## 2 Adelaide South ~ Holiday ets    2018 Q2 N(173, 473) 173.
## 3 Adelaide South ~ Holiday ets    2018 Q3 N(169, 489) 169.
## 4 Adelaide South ~ Holiday ets    2018 Q4 N(186, 505) 186.
## 5 Adelaide South ~ Holiday ets    2019 Q1 N(210, 521) 210.
## 6 Adelaide South ~ Holiday ets    2019 Q2 N(173, 537) 173.
## 7 Adelaide South ~ Holiday ets    2019 Q3 N(169, 553) 169.
## 8 Adelaide South ~ Holiday ets    2019 Q4 N(186, 569) 186.
## 9 Adelaide~ South ~ Holiday ets    2018 Q1   N(19, 36)  19.4
##10 Adelaide~ South ~ 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) +  
  xlab("Year") + ylab("Overnight trips (thousands)")
```



## 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

		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	A,N,N	A,N,A	<del>A,N,M</del>
A	(Additive)	A,A,N	A,A,A	<del>A,A,M</del>
A <sub>d</sub>	(Additive damped)	A,A <sub>d</sub> ,N	A,A <sub>d</sub> ,A	<del>A,A<sub>d</sub>,M</del>

## 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 <sub>d</sub>	(Additive damped)	M,A <sub>d</sub> ,N	M,A <sub>d</sub> ,A	M,A <sub>d</sub> ,M

# 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:

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

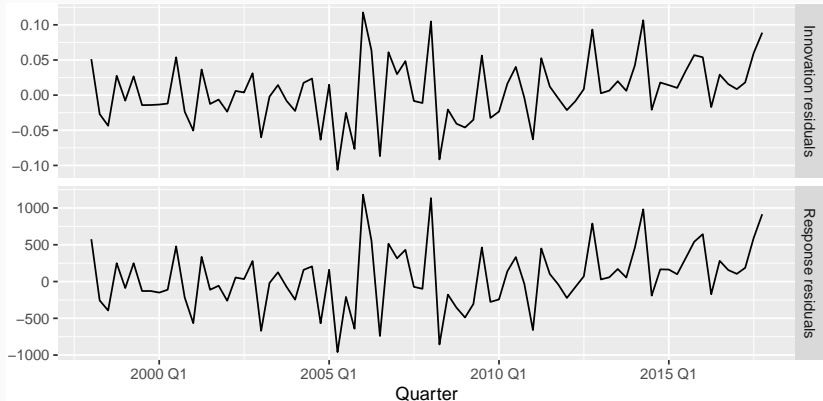
# Example: Australian holiday tourism

```
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    s1    s2    s3    s4  
## 9667 0.943 0.927 0.968 1.16
```

# Example: Australian holiday tourism

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



# Outline

- 1 Exponential smoothing
- 2 Simple exponential smoothing
- 3 Models with trend
- 4 Models with seasonality
- 5 Innovations state space models
- 6 Forecasting with exponential smoothing

# 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  $t > T$ .

# 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  $t > T$ .

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

## Example: ETS(A,A,N)

$$y_{T+1} = \ell_T + b_T + \varepsilon_{T+1}$$

$$\hat{y}_{T+1|T} = \ell_T + b_T$$

$$y_{T+2} = \ell_{T+1} + b_{T+1} + \varepsilon_{T+2}$$

$$= (\ell_T + b_T + \alpha\varepsilon_{T+1}) + (b_T + \beta\varepsilon_{T+1}) + \varepsilon_{T+2}$$

$$\hat{y}_{T+2|T} = \ell_T + 2b_T$$

etc.



## Example: ETS(M,A,N)

$$y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$$

$$\hat{y}_{T+1|T} = \ell_T + b_T.$$

$$y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+2})$$

$$= \{(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + [b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}]\} (1 + \varepsilon_{T+2})$$

$$\hat{y}_{T+2|T} = \ell_T + 2b_T$$

etc.

# Forecasting with ETS models

**Prediction intervals:** can only 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  $\sigma_h$  is forecast standard deviation.

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

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

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

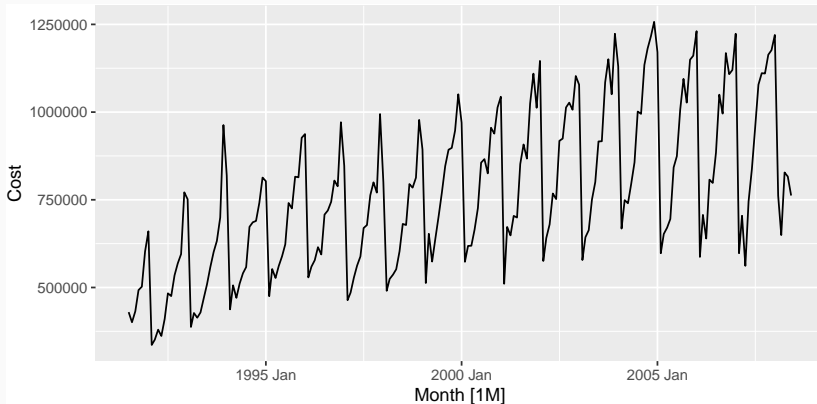
$$(A,N,A) \quad \sigma_h = \sigma^2 \left[ 1 + \alpha^2(h-1) + \gamma k(2\alpha + \gamma) \right]$$

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

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

# Example: Corticosteroid drug sales

```
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      b      s1      s2      s3      s4      s5      s6      s7      s8  
## 417269 8206 0.872 0.826 0.756 0.773 0.687 1.28 1.32 1.18  
##      s9 s10 s11 s12  
## 1.16 1.1 1.05 0.981  
##  
## sigma^2: 0.0046  
##  
## AIC AICc BIC  
## 5515 5519 5575
```

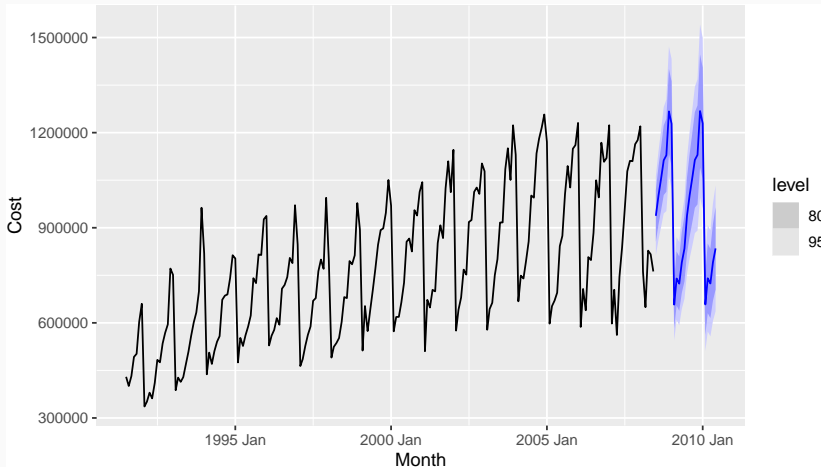
# Example: Corticosteroid drug sales

```
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      b      s1      s2      s3      s4      s5      s6  
## 409706 9097 -99075 -136602 -191496 -174531 -241437 210644  
##      s7      s8      s9      s10     s11     s12  
## 244644 145368 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()
```

```
h02 %>%  
  model(  
    auto = ETS(Cost),  
    AAA = ETS(Cost ~ error("A") + trend("A") + season("A"))  
  ) %>%  
  accuracy() %>%  
  transmute(Model = .model, ME, MAE, RMSE, MAPE, MASE) %>%  
  knitr::kable(booktabs = TRUE)
```



## Your turn

- Use `ETS()` on some of these series:  
*tourism, gafa\_stock, pelt*
- Does it always give good forecasts?
- Find an example where it does not work well.  
Can you figure out why?