Exponential Smoothing

Zachary Safir, Ethan Haley, Stefano Biguzzi

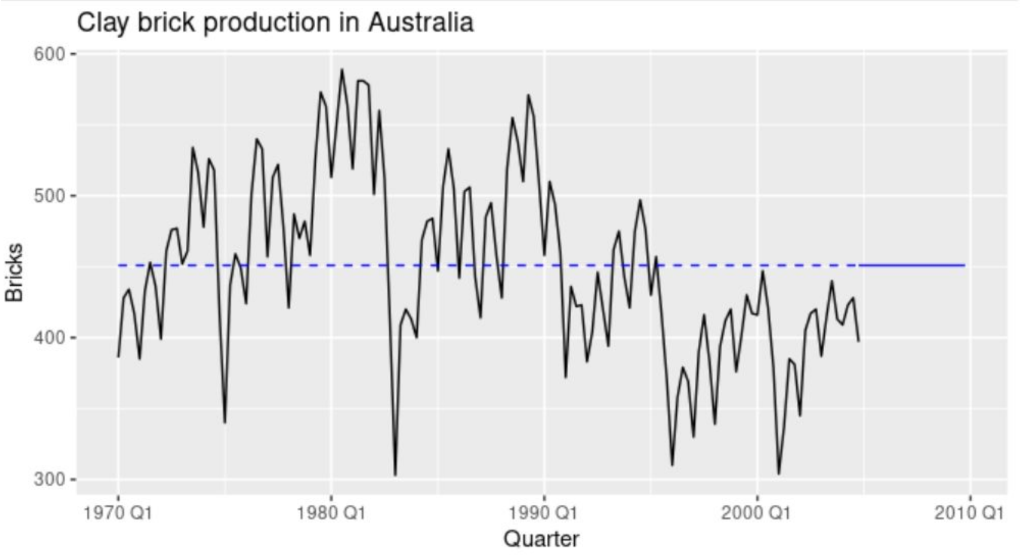
Before We Proceed: Chapter 5 Basic Methodologies

Sometimes one of these simple methods will be the best forecasting method available; but in many cases, these methods will serve as benchmarks rather than the method of choice. That is, any forecasting methods we develop will be compared to these simple methods to ensure that the new method is better than these simple alternatives. If not, the new method is not worth considering.

Mean Method

Here, the forecasts of all future values are equal to the average (or "mean") of the historical data.

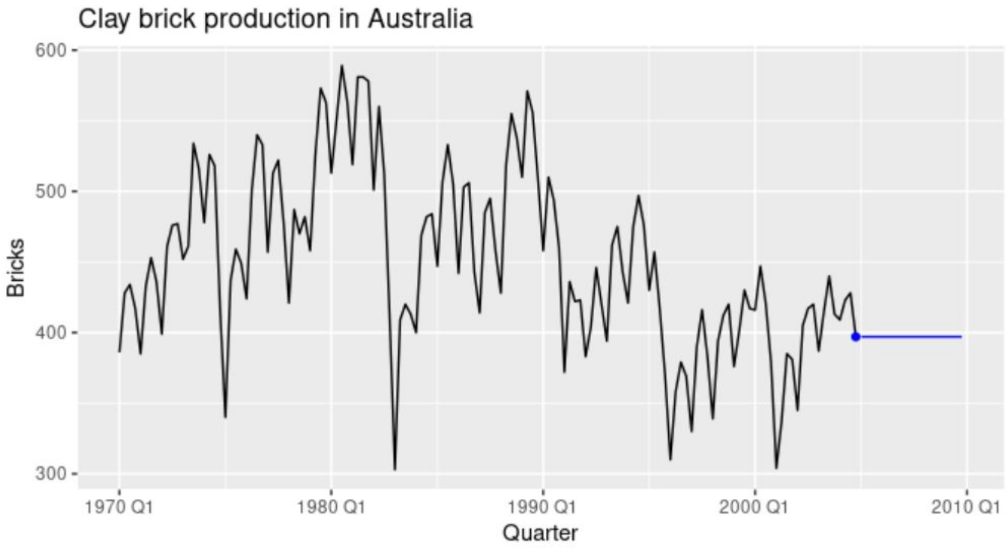
All previous values are treated equally



Naïve method

For naïve forecasts, we simply set all forecasts to be the value of the last observation.

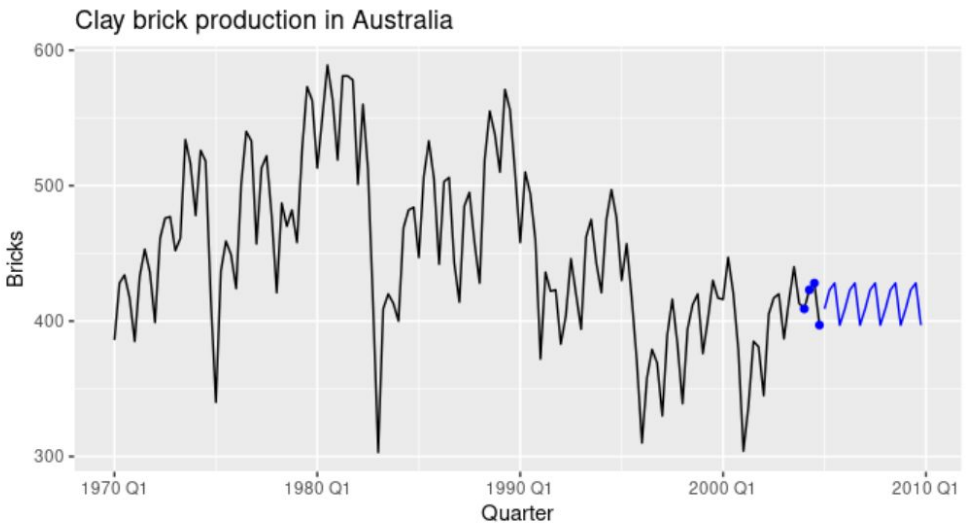
This method works remarkably well for many economic and financial time series.



Seasonal Naïve method

A similar method is useful for highly seasonal data. In this case, we set each forecast to be equal to the last observed value from the same season (e.g., the same month of the previous year).

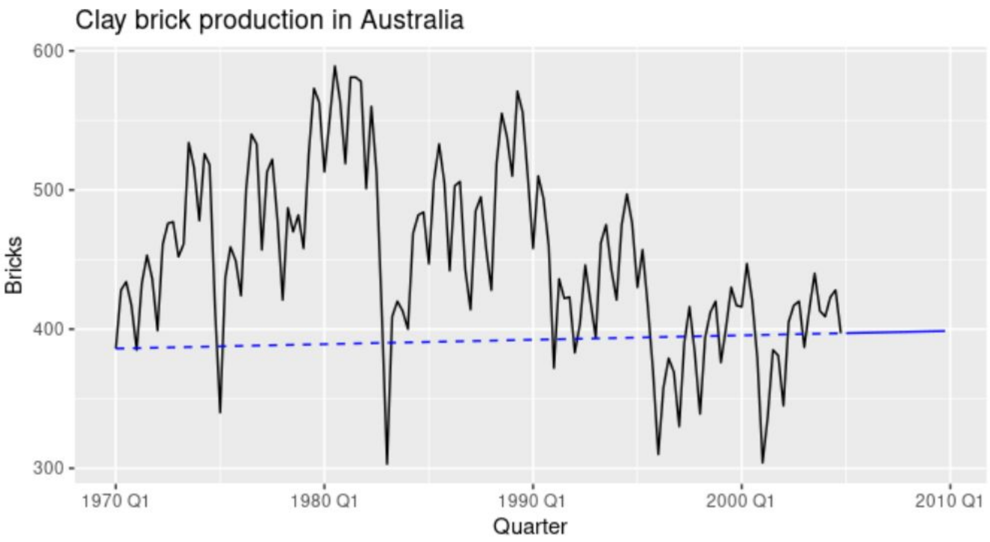
With monthly data, the forecast for all future February values is equal to the last observed February value. With quarterly data, the forecast of all future Q2 values is equal to the last observed Q2 value (where Q2 means the second quarter).



Drift method

A variation on the naïve method is to allow the forecasts to increase or decrease over time, where the amount of change over time (called the **drift**) is set to be the average change seen in the historical data.

This is equivalent to drawing a line between the first and last observations, and extrapolating it into the future.



What is Exponential Smoothing?

Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight.

Three Categories of Exponential Smoothing

1. Methods that do not assume any seasonality or trend
2. Methods that add trend but not seasonality
3. Methods that add both trend and seasonality

The Basics of Exponential Smoothing

Start with a forecasted value: ℓ_0

- *Unlike the regression case (where we have formulas which return the values of the regression coefficients that minimise the SSE), this involves a non-linear minimisation problem, and we need to use an optimisation tool to solve it.*

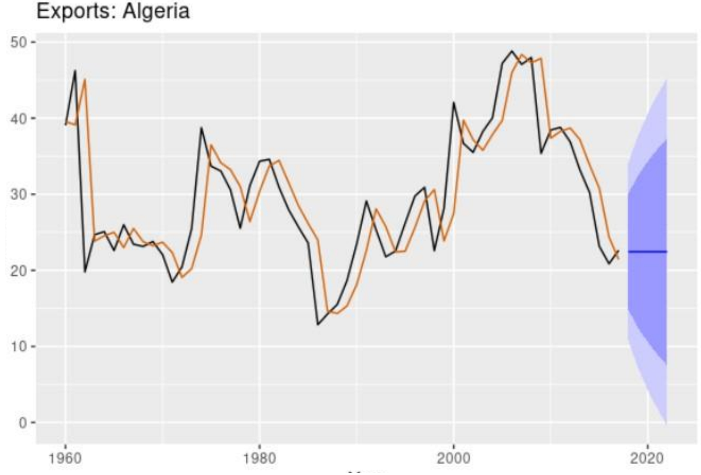
Choose a smoothing parameter: α

- This is also given by the optimisation tool.

Method 1 - No Trend, No Season

α = 0.84

Year	Time	Observation	Level	Forecast
	t	y_t	ℓ_t	$\hat{y}_{t\|t-1}$
2014	55	30.22	30.80	33.85
2015	56	23.17	24.39	30.80
2016	57	20.86	21.43	24.39
2017	58	22.64	22.44	21.43
	h			$\hat{y}_{T+h\|T}$
2018	1			22.44
2019	2			22.44
2020	3			22.44
2021	4			22.44
2022	5			22.44



The simplest of the exponential smoothing methods is naturally called simple exponential smoothing (SES). This method is suitable for forecasting data with no clear trend or seasonal pattern.

There are two ways to mathematically show the SES, the **Weighted average** and **Component forms**

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

Methods 2 - Trends, No Season

Holt's method: α = 0.9999, β^* = 0.3267

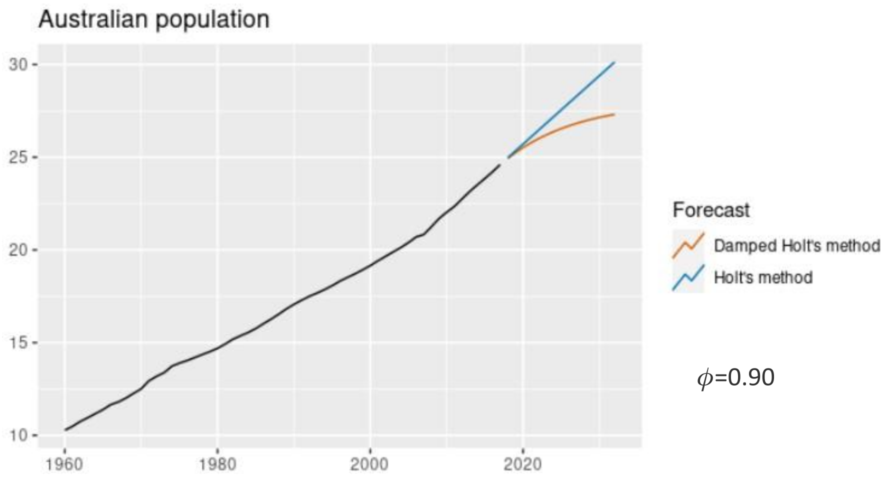
Year	Time	Observation	Level	Slope	Forecast
	t	y_t	ℓ_t		$\hat{y}_{t+1\|t}$
2014	55	23.50	23.50	0.37	23.52
2015	56	23.85	23.85	0.36	23.87
2016	57	24.21	24.21	0.36	24.21
2017	58	24.60	24.60	0.37	24.57
	h				$\hat{y}_{T+h\|T}$
2018	1				24.97
2019	2				25.34
2020	3				25.71

Holt's linear method:

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

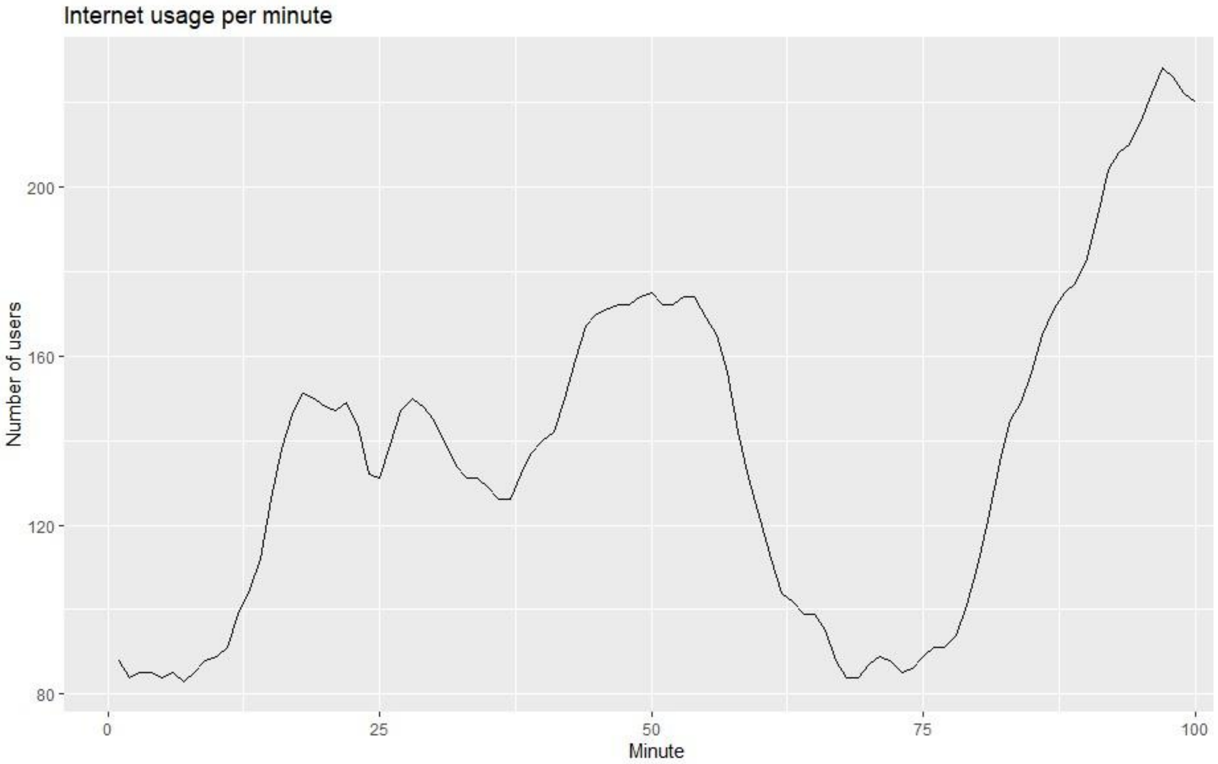
Damped trend method:

- Forecast equation: $\hat{y}_{t+h|t} = \ell_t + (\phi+\phi^2+\ldots+\phi^h)b_t$
- Level equation: $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$
- Trend equation: $b_t = \beta^*(\ell_t-\ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$
- Most common ϕ values are $0.8 < \phi < 0.98$

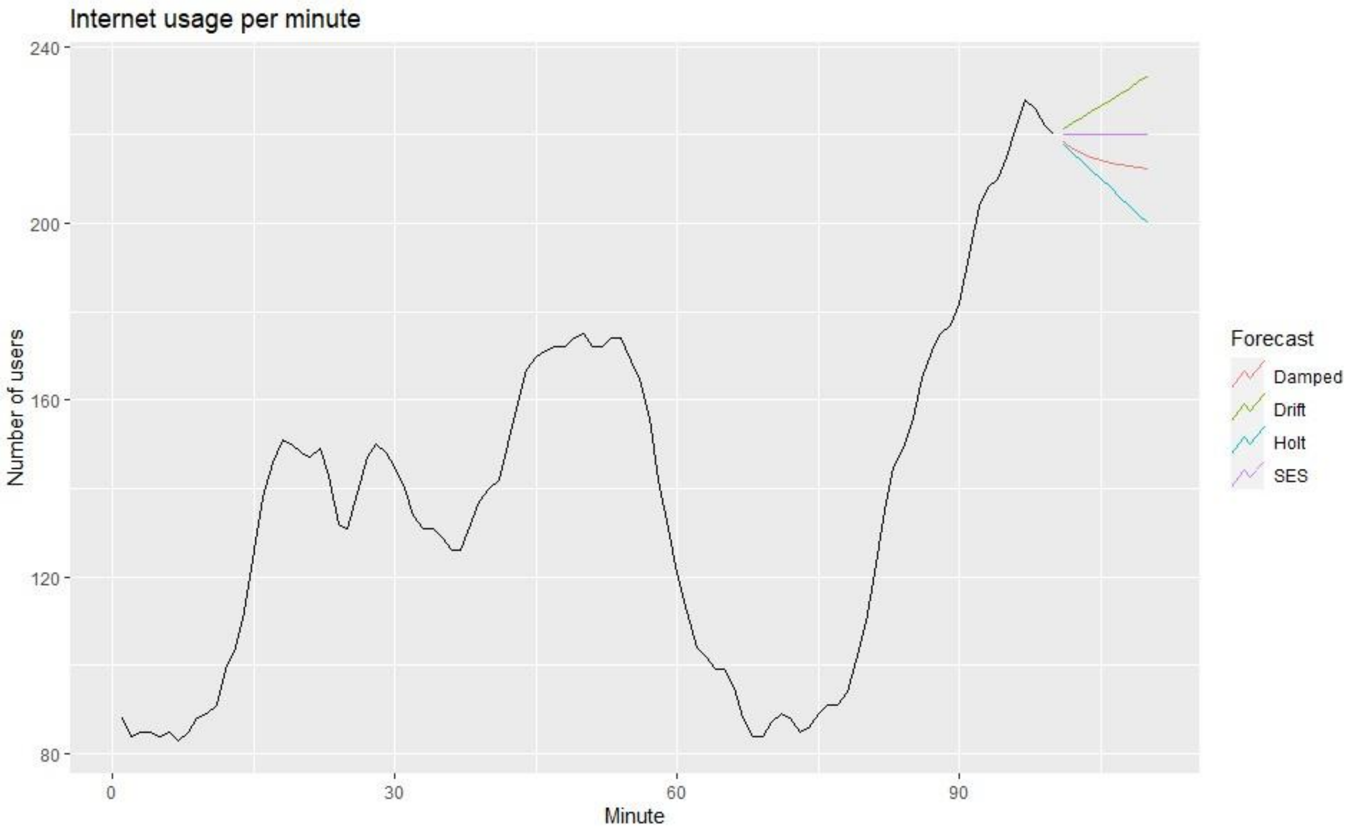


ϕ=0.90

What method would you use?



Trend Forecasting Methods Example



Methods 3 - Trends and Seasons

Holt-Winters' additive method:

- Forecast equation: $\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$
- Level equation: $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$
- Trend equation: $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$
- Season equation: $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$

Holt-Winters' multiplicative method:

- Forecast equation: $\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$
- Level equation: $\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$
- Trend equation: $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$
- Season equation: $s_t = \gamma(y_t / (\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$

Holt-Winters' damped multiplicative method:

- Forecast equation: $\hat{y}_{t+h|t} = (\ell_t + (\phi+\phi^2+\ldots+\phi^h)b_t)s_{t+h-m(k+1)}$
- Level equation: $\ell_t = \alpha(y_t / s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$
- Trend equation: $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$
- Season equation: $s_t = \gamma(y_t / (\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$

Holt-Winters' additive method:

$\alpha = 0.2620$, $\beta^* = 0.1646$, $\gamma = 0.0001$, RMSE = 0.4169

Quarter	Time	Observation	Level	Slope	Season	Forecast
1998 Q1	4	11.8	9.9	0.0	1.5	11.3
1998 Q2	5	9.3	9.9	0.0	-0.3	9.7
1998 Q3	6	8.6	9.7	-0.0	-0.7	9.2
1998 Q4	7	9.3	9.8	0.0	-0.5	9.2
	⋮	⋮	⋮	⋮	⋮	⋮
2017 Q1	80	12.4	10.9	0.1	1.5	12.3
2017 Q2	81	10.5	10.9	0.1	-0.3	10.7
2017 Q3	82	10.5	11.0	0.1	-0.7	10.3
2017 Q4	83	11.2	11.3	0.1	-0.5	10.6
	h					$\hat{y}_{T+h\|T}$
2018 Q1	1					12.9
2018 Q2	2					11.2
2018 Q3	3					11.0
2018 Q4	4					11.2
2019 Q1	5					13.4
2019 Q2	6					11.7
2019 Q3	7					11.5
2019 Q4	8					11.7

Holt-Winters' multiplicative method:

$\alpha = 0.2237$, $\beta^* = 0.1360$, $\gamma = 0.0001$, RMSE = 0.4122

Quarter	Time	Observation	Level	Slope	Season	Forecast
1998 Q1	4	11.8	10.0	-0.0	1.2	11.6
1998 Q2	5	9.3	9.9	-0.0	1.0	9.7
1998 Q3	6	8.6	9.8	-0.0	0.9	9.2
1998 Q4	7	9.3	9.8	-0.0	0.9	9.2
	⋮	⋮	⋮	⋮	⋮	⋮
2017 Q1	80	12.4	10.8	0.1	1.2	12.6
2017 Q2	81	10.5	10.9	0.1	1.0	10.6
2017 Q3	82	10.5	11.1	0.1	0.9	10.2
2017 Q4	83	11.2	11.3	0.1	0.9	10.5
	h					$\hat{y}_{T+h\|T}$
2018 Q1	1					13.3
2018 Q2	2					11.2
2018 Q3	3					10.8
2018 Q4	4					11.1
2019 Q1	5					13.8
2019 Q2	6					11.7
2019 Q3	7					11.3
2019 Q4	8					11.6

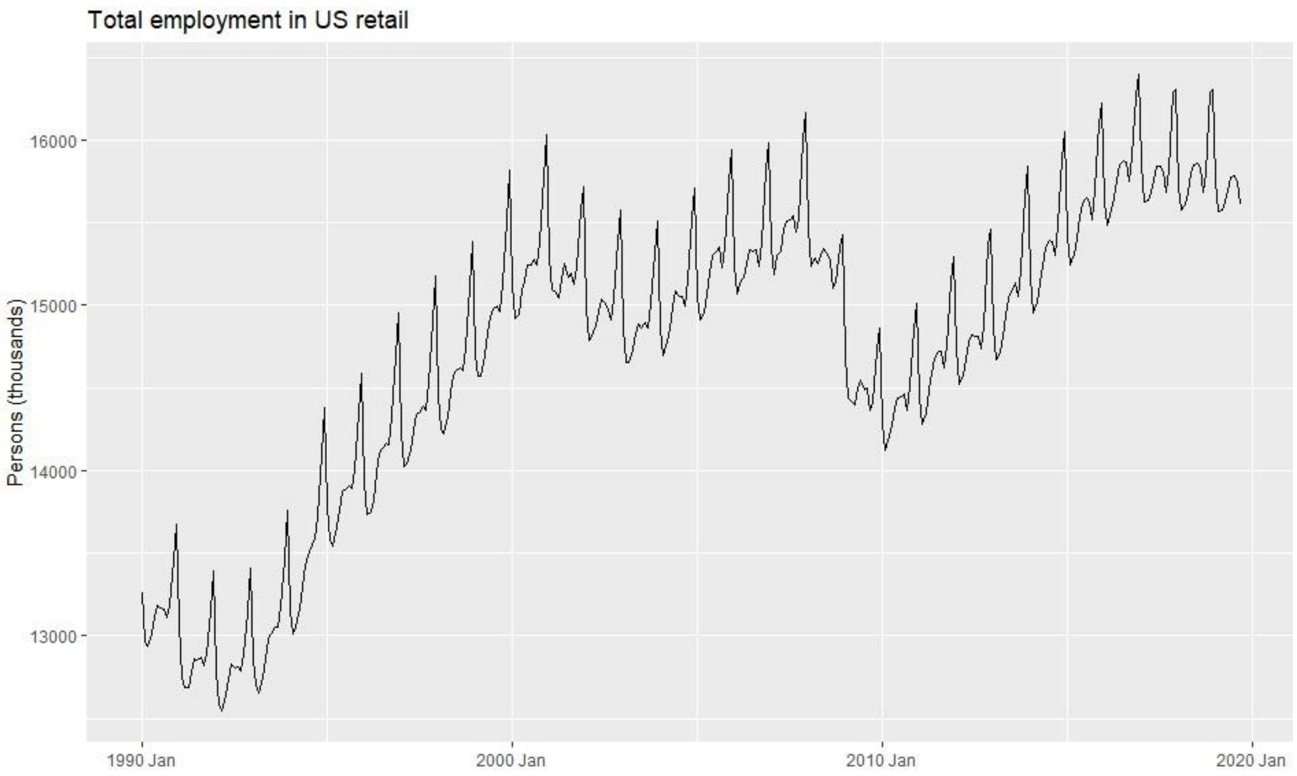
Additive or Multiplicative

From Chapter 3

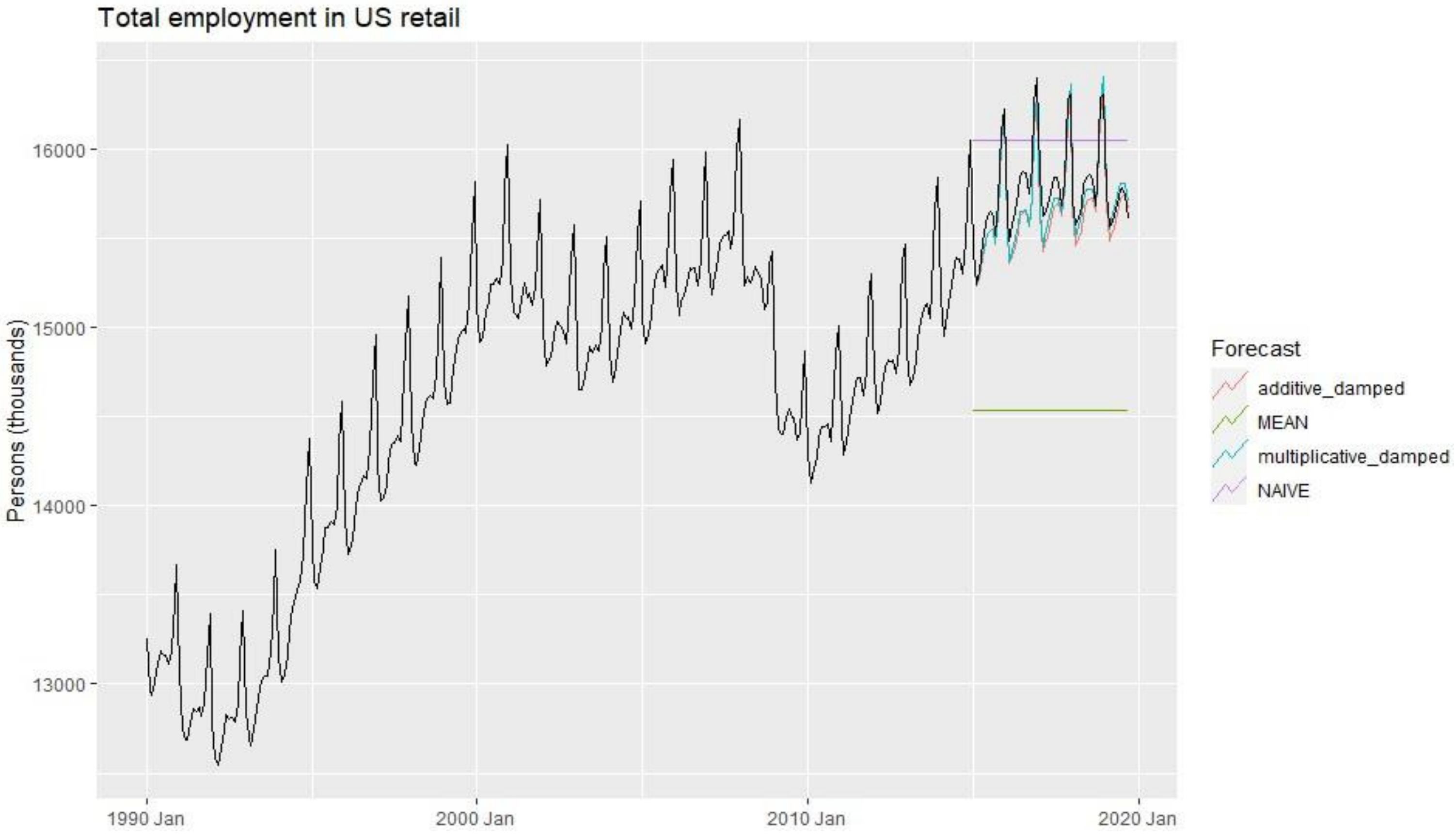
The additive decomposition is the most appropriate if the magnitude of the seasonal fluctuations, or the variation around the trend-cycle, does not vary with the level of the time series.

When the variation in the seasonal pattern, or the variation around the trend-cycle, appears to be proportional to the level of the time series, then a multiplicative decomposition is more appropriate.

What method would you use?



Trend/Season Forecasting Methods Example



Trend Season Notation

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)

Short hand	Method
(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)	Holt-Winters' damped method

We do not consider the multiplicative trend methods in this book as they tend to produce poor forecasts.

Adding (or Multiplying) Errors to the Equation

Before: We specified models such as "(**A, M**)" to indicate **A**dditive trend and **M**ultiplicative season

Now: We specify "(**M, A, M**)" to indicate **M**ultiplicative error, **A**dditive trend and **M**ultiplicative season

The new error term gives rise to the **ETS** (**E**rror, **T**rend, **S**eason) modeling approach our textbook uses.

The **ETS()** function in R's **fable** package, co-authored by Rob Hyndman and Earo Wang, implements it.

Why do we need to specify a type of error?

It allows the **ETS()** model to statistically calculate confidence intervals for its forecasts.

⟶ *The point forecasts themselves remain the same as before, because we forecast all errors to be 0*

Additive Errors: $y_t = \ell_{t-1} + \varepsilon_t$

The error term ε_t is a **measured** residual:
The difference between the observed value y_t and the forecasted value ℓ_{t-1}

Multiplicative Errors: $y_t = \ell_{t-1}(1 + \varepsilon_t)$

The 1 that's added to the error term comes from **defining** the multiplicative error to be **relative** to the forecast like this:

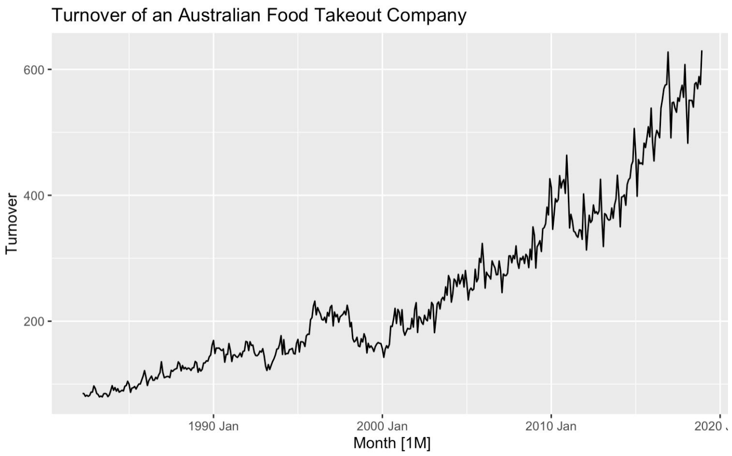
$$\varepsilon_t = \frac{y_t - \ell_{t-1}}{\ell_{t-1}}$$

If you let ETS() fit parameters to Additive and Multiplicative Error models, the parameters will be different, because the errors are different, and the model finds the MLE of the errors.

```{r}
errorMods = myseries %>%
  model("additive error model"= ETS(Turnover ~ error("A") + trend("A") + season("M")),
        "multiplicative error model" = ETS(Turnover ~ error("M") + trend("A") + season("M")))
errorMods %>%
  tidy %>%
  pivot_wider(names_from = ".model", values_from = "estimate") %>%
  select(3:5)
```

A tibble: 17 × 3

term <chr>	additive error model <dbl>	multiplicative error model <dbl>
alpha	0.615993436	0.71508960
beta	0.008336214	0.03088603
gamma	0.234395901	0.10553361
l[0]	84.551955084	82.38422818
b[0]	-0.638414542	-0.81372611
s[0]	1.001327329	1.00656816
s[-1]	0.938464035	0.95638094
s[-2]	1.116125247	1.06149360
s[-3]	1.162640785	1.08069818
s[-4]	1.073455486	0.99126157



Not all model combinations are stable.

“Some of the combinations of trend, seasonality and error can occasionally lead to numerical difficulties; specifically, any model equation that requires division by a state component could involve division by zero. This is a problem for models with additive errors and either multiplicative trend or multiplicative seasonality, as well as the model with multiplicative errors, multiplicative trend and additive seasonality. These models should therefore be used with caution.”

Source: Hyndman, Rob; Koehler, Anne B.; Ord, J. Keith; Snyder, Ralph D. (2008-06-18T23:58:59). Forecasting with Exponential Smoothing (Springer Series in Statistics) (p. 31). Springer Berlin Heidelberg.

Other notes/warnings from the ETS() docs:

- If the error is specified as **M**, the data must be non-negative.
- ETS() does not accept missing values.

How does the model choose parameters?

Smoothing Parameters

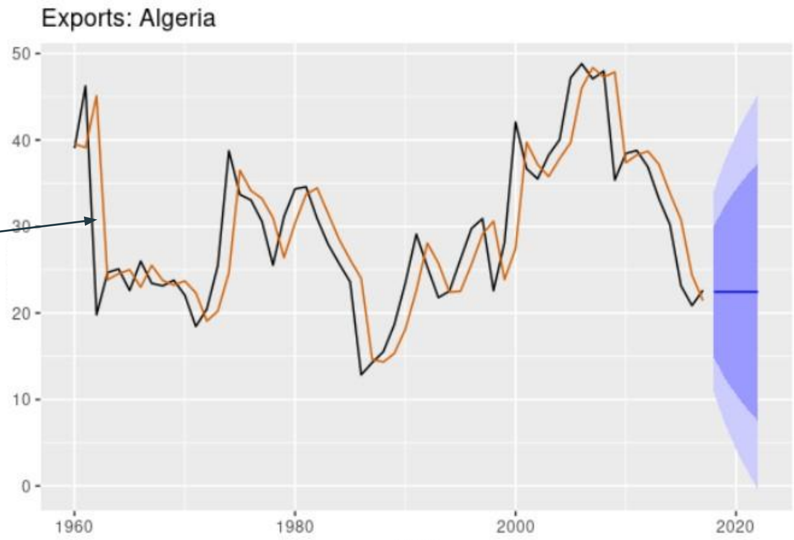
$\alpha, \beta^*, \gamma, \phi$

Initial Values

ℓ_0, b_0, s_0

Back to our Simple Exponential Smoothing example

Choose the parameters and initial values that minimize the area between the orange and black lines.
(I.e. minimize the RMSE)



Additional resources

Forecasting and Analytics with ADAM Chapters 1-5

Forecasting with Exponential Smoothing: the State Space Approach
(If you want to read further on multiplicative trend methods)

Article from last week's reading materials, using a different R package
(which has a nice example of modeling multiple seasons at once)

Code to produce participation graphs

Slide 13

```
library(fpp3)

www_usage <- as_tsibble(WWWusage)

fit <- www_usage %>%
  model(
    SES = ETS(value ~ error("A") + trend("N") + season("N")),
    Holt = ETS(value ~ error("A") + trend("A") + season("N")),
    Damped = ETS(value ~ error("A") + trend("Ad") +
                   season("N")),
    Drift = RW(value~drift())
  )

fc <- fit %>%
  forecast(h=10)

fc %>%
  autoplot(www_usage,level = NULL) +
  labs(x="Minute", y="Number of users",
       title = "Internet usage per minute") +
  guides(colour=guide_legend(title="Forecast"))
```

Code to produce participation graphs

Slide 17

```r
library(fpp3)

us_retail_employment <- us_employment %>%
  filter(year(Month) >= 1990, Title == "Retail Trade") %>%
  select(-Series_ID)

fit <- us_retail_employment %>%
  filter(year(Month) < 2015) %>%
  model(
    additive_damped = ETS(Employed ~ error("A")+trend("Ad")+season("A")),
    multiplicative_damped = ETS(Employed ~ error("M")+trend("Ad")+season("M")),
    `NAIVE` = NAIVE(Employed),
    MEAN = MEAN(Employed)
  )

fc <- fit %>%
  forecast(h=57)

fc %>%
  autoplot(us_retail_employment,level=NULL) +
  labs(y = "Persons (thousands)",
       title = "Total employment in US retail")+
  guides(colour=guide_legend(title="Forecast"))
```

ZOOM

https://us04web.zoom.us/j/53978007?pwd=cjcvZzZTbWl1Qk02TTNuaDdkUi9ZUT09

Specials

The *specials* define the methods and parameters for the components (error, trend, and seasonality) of an ETS model. If more than one method is specified, `ETS` will consider all combinations of the specified models and select the model which best fits the data (minimising `ic`). The method argument for each specials have reasonable defaults, so if a component is not specified an appropriate method will be chosen automatically.

There are a couple of limitations to note about ETS models:

- It does not support exogenous regressors.
- It does not support missing values. You can complete missing values in the data with imputed values (e.g. with `tidyr::fill()`, or by fitting a different model type and then calling `fabletools::interpolate()`) before fitting the model.

error

The `error` special is used to specify the form of the error term.

```
error(method = c("A", "M"))
```

`method`	The form of the error term: either additive ("A") or multiplicative ("M"). If the error is multiplicative, the data must be non-negative. All specified methods are tested on the data, and the one that gives the best fit (lowest `ic`) will be kept.

trend

The `trend` special is used to specify the form of the trend term and associated parameters.

```
trend(method = c("N", "A", "Ad"),
      alpha = NULL, alpha_range = c(1e-04, 0.9999),
      beta = NULL, beta_range = c(1e-04, 0.9999),
      phi = NULL, phi_range = c(0.8, 0.98))
```

`method`	The form of the trend term: either none ("N"), additive ("A"), multiplicative ("M") or damped variants ("Ad", "Md"). All specified methods are tested on the data, and the one that gives the best fit (lowest `ic`) will be kept.
`alpha`	The value of the smoothing parameter for the level. If `alpha = 0`, the level will not change over time. Conversely, if `alpha = 1` the level will update similarly to a random walk process.
`alpha_range`	If `alpha=NULL`, `alpha_range` provides bounds for the optimised value of `alpha`.
`beta`	The value of the smoothing parameter for the slope. If `beta = 0`, the slope will not change over time. Conversely, if `beta = 1` the slope will have no memory of past slopes.
`beta_range`	If `beta=NULL`, `beta_range` provides bounds for the optimised value of `beta`.
`phi`	The value of the dampening parameter for the slope. If `phi = 0`, the slope will be dampened immediately (no slope). Conversely, if `phi = 1` the slope will not be dampened.
`phi_range`	If `phi=NULL`, `phi_range` provides bounds for the optimised value of `phi`.

season

The `season` special is used to specify the form of the seasonal term and associated parameters. To specify a nonseasonal model you would include `season(method = "N")`.

```
season(method = c("N", "A", "M"), period = NULL,
       gamma = NULL, gamma_range = c(1e-04, 0.9999))
```

`method`	The form of the seasonal term: either none ("N"), additive ("A") or multiplicative ("M"). All specified methods are tested on the data, and the one that gives the best fit (lowest `ic`) will be kept.
`period`	The periodic nature of the seasonality. This can be either a number indicating the number of observations in each seasonal period, or text to indicate the duration of the seasonal window (for example, annual seasonality would be "1 year").
`gamma`	The value of the smoothing parameter for the seasonal pattern. If `gamma = 0`, the seasonal pattern will not change over time. Conversely, if `gamma = 1` the seasonality will have no memory of past seasonal periods.
`gamma_range`	If `gamma=NULL`, `gamma_range` provides bounds for the optimised value of `gamma`.