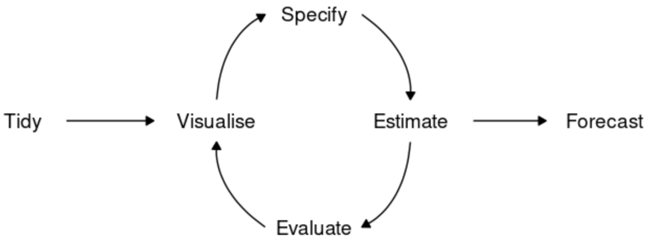
Introduction to Time Series Analyis

Dr Rafael de Andrade Moral
Associate Professor of Statistics, Maynooth University

rafael.deandrademoral@mu.ie
https://rafamoral.github.io

Forecasting Workflow

- The process of producing forecasts for time series data can be broken down into a few steps.



Data preparation (tidy)

- The first step in forecasting is to prepare data in the correct format. This process may involve loading in data, identifying missing values, filtering the time series, and other pre-processing tasks.
- The functionality provided by `tsibble` and other packages in the `tidyverse` substantially simplifies this step.
- Many models have different data requirements; some require the series to be in time order, others require no missing values. Checking your data is an essential step to understanding its features and should always be done before models are estimated.
- Example: GDP per capita.

Define a model (specify)

- There are many different time series models that can be used for forecasting.
- Specifying an appropriate model for the data is essential for producing appropriate forecasts.
- Models in `fable` are specified using model functions, which each use a formula (`y ~ x`) interface.
- The response variable(s) are specified on the left of the formula, and the structure of the model is written on the right.
- E.g. a linear trend model for GDP per capita can be specified with

```
TSLM(GDP_per_capita ~ trend())
```

- Here the model is `TSLM` (time series linear model), the response variable is `GDP_per_capita` and it is being modelled using `trend()` (a special function specifying a linear trend when used within `TSLM()`).

Train the model (estimate)

- Once an appropriate model is specified, we next train the model on some data.
- One or more model specifications can be estimated using the `model()` function.
- This fits a linear trend model to the GDP per capita data for each combination of key variables in the `tsibble`.
- In this example, it will fit a model to each of the 263 countries in the dataset. The resulting object is a model table or a `mable`.

Check model performance (evaluate)

- Once a model has been fitted, it is important to check how well it has performed on the data.
- There are several diagnostic tools available to check model behaviour, and also accuracy measures that allow one model to be compared against another.
- We will see how to evaluate point and distributional forecast accuracy.

Produce forecasts (forecast)

- With an appropriate model specified, estimated and checked, it is time to produce the forecasts using `forecast()`.
- The easiest way to use this function is by specifying the number of future observations to forecast.
- E.g. forecasts for the next 10 observations can be generated using `h = 10`.
- We can also use natural language; e.g. `h = "2 years"`.

Produce forecasts (forecast)

- This is a forecast table, or a `fable`.
- The forecasts can be plotted along with the historical data using `autoplot()` as follows.

```
fit %>%
  filter(Country == "Ireland") %>%
  forecast(h = "3 years") %>%
  autoplot(gdppc) +
  theme_bw() +
  labs(y = "$US", title = "GDP per capita for Ireland")
```

Simple forecasting methods

- Some forecasting methods are extremely simple and surprisingly effective.
- We will use four simple forecasting methods as benchmarks throughout this course.
- 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.
- To illustrate them, we will use quarterly number of house registrations in Ireland between 1994 and 2020.

Mean method

- The forecasts of all future values are equal to the average (or “mean”) of the historical data.
- Let the historical data be denoted by $y_1, \dots, y_T$.
- We can write the forecasts as

$$\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T$$

- The notation $\hat{y}_{T+h|T}$ is a short-hand for the estimate of y_{T+h} based on the data $y_1, \dots, y_T$

Naïve method

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

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

- This method works remarkably well for many economic and financial time series.
- Because a naïve forecast is optimal when data follow a random walk, these are also called random walk forecasts and the `RW()` function can be used instead of `NAIVE()`.

Seasonal naïve method

- In this case, we set each forecast to be equal to the last observed value from the same season of the year.
- Formally, the forecast for time $T + h$ is written as

$$\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$$

- m is the seasonal period and k is the integer part of $(h-1)/m$ (i.e. number of complete years in the forecast period prior to time $T+h$).

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.
- The forecast for time $T + h$ is given by

$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + h\left(\frac{y_T - y_1}{T-1}\right)$$

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

Fitted values

- Each observation in a time series can be forecast using all previous observations.
- We call these **fitted values** and they are denoted by $\hat{y}_{t|t-1}$, meaning the forecast of y_t based on observations $y_1, \dots, y_{t-1}$
- We use them so often we sometimes drop part of the subscript and just write $\hat{y}_t$ instead of $\hat{y}_{t|t-1}$.
- Fitted values are often not true forecasts because any parameters involved in the forecasting method are estimated using all available observations in the time series, including future observations.
- For example, if we use the mean method, the fitted values are given by

$$\hat{y}_t = \hat{c}$$

where $\hat{c}$ is the average computed over all available observations, including those *after* time t.

Residuals

- The “residuals” in a time series model are what is left over after fitting a model.
- The residuals are equal to the difference between the observations and the corresponding fitted values

$$e_t = y_t - \hat{y}_t$$

- If a transformation has been used in the model, then it is often useful to look at residuals on the transformed scale. These are called “innovation residuals”.
- E.g. suppose we model $w_t = \log y_t$. Then the innovation residuals are $w_t - \hat{w}_t$ whereas the regular residuals are $y_t - \hat{y}_t$.
- If no transformation has been used then the innovation residuals are identical to the regular residuals.

Residual diagnostics

- A good forecasting method will yield innovation residuals with the following properties:

1. The innovation residuals are uncorrelated. If there are correlations between innovation residuals, then there is information left in the residuals which should be used in computing forecasts.
2. The innovation residuals have zero mean. If they have a mean other than zero, then the forecasts are biased.

- Any forecasting method that does not satisfy these properties can be improved.
- That doesn't mean that forecasting methods that satisfy these properties cannot be improved.
- It is possible to have several different forecasting methods for the same dataset, all of which satisfy these properties.
- Adjusting for bias is easy: if the residuals have mean m, then simply subtract m from all forecasts and the bias problem is solved. Fixing the correlation problem is harder.

Residual diagnostics

- In addition to these properties it is also useful (but *not necessary*) if the innovation residuals

3. have constant variance;
4. are normally distributed.

Example: Forecasting Google daily closing stock prices

- For stock market prices and indexes, the best forecasting method is often the naïve method.
- Each forecast is simply equal to the last observed value, or

$$\hat{y}_t = y_{t-1}$$

- Hence, the residuals are simply equal to the difference between consecutive observations:

$$e_t = y_t - \hat{y}_t = y_t - y_{t-1}$$

Example: Forecasting Google daily closing stock prices

- These graphs show that the naïve method produces forecasts that appear to account for all available information.
- The mean of the residuals is close to zero and there is no significant correlation in the residuals series.
- The time plot of the residuals shows that the variation of the residuals stays much the same across the historical data, apart from the one outlier, and therefore the residual variance can be treated as constant.
- This can also be seen on the histogram of the residuals.
- The histogram suggests that the residuals may not be normal – the right tail seems a little too long, even when we ignore the outlier.
- Consequently, forecasts from this method will probably be quite good, but prediction intervals that are computed assuming a normal distribution may be inaccurate.

Portmanteau tests for autocorrelation

- In addition to looking at the ACF plot, we can also do a more formal test for autocorrelation by considering a whole set of autocorrelation values as a group, rather than treating each one separately.
- When we look at the ACF plot to see whether each spike is within the required limits, we are implicitly carrying out multiple hypothesis tests, each one with a small probability of giving a false positive.
- When enough of these tests are done, it is likely that at least one will give a false positive, and so we may conclude that the residuals have some remaining autocorrelation, when in fact they do not.
- In order to overcome this problem, we test whether the first l autocorrelations are significantly different from what would be expected from a white noise process.
- A test for a group of autocorrelations is called a **portmanteau test**, from a French word describing a suitcase or coat rack carrying several items of clothing.

Distributional forecasts and prediction intervals

- We express the uncertainty in our forecasts using a probability distribution.
- It describes the probability of observing possible future values using the fitted model.
- The point forecast is the mean of this distribution.
- *Most* time series models produce *normally distributed forecasts*, that is, we assume that the distribution of possible future values follows a normal distribution.
- We will look at a couple of alternatives to normal distributions later in this section.

Prediction intervals

- A prediction interval gives an interval within which we expect y_t to lie with a specified probability.
- For example, assuming that distribution of future observations is normal, a 95% prediction interval for the $h-$step forecast is

  $$\hat{y}_{T+h|T} \pm 1.96\hat{\sigma}_h$$

  where $\hat{\sigma}_h$ is an estimate of the standard deviation of the $h-$step forecast distribution.
- The value 1.96 refers to a 95% coverage probability. It will change for other coverage probabilities.
- The value of prediction intervals is that they express the **uncertainty** in the forecasts.
- Point forecasts can be of almost no value without the accompanying prediction intervals.

Prediction intervals from bootstrapped residuals

- When a normal distribution for the residuals is an unreasonable assumption, one alternative is to use **bootstrapping**, which only assumes residuals are uncorrelated with constant variance.
- We can simulate the next observation of a time series through

$$y_{T+1} = \hat{y}_{T+1} + e_{T+1}$$

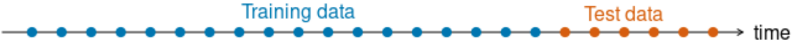
  where $\hat{y}_{T+1}$ is a one-step ahead forecast and e_{T+1} is an unknown future error.
- Assuming future errors will be similar to past errors, we can obtain e_{T+1} by sampling from the collection of errors we have seen in the past.
- We can repeat the process to obtain future values.
- Doing this repeatedly, we obtain many possible futures. Each possible future is a bootstrap realisation.
- The `generate()` function is useful to do this.

Forecasting with decomposition

- Time series decomposition can be a useful step in producing forecasts.
- Additive decomposition: $y_t = \hat{S}_t + \hat{T}_t + \hat{R}_t$
- Multiplicateive decomposition: $y_t = \hat{S}_t \hat{T}_t \hat{R}_t$
- To forecast a decomposed time series, we forecast the seasonal component $\hat{S}_t$ and the seasonally adjusted component $\hat{A}_t = \hat{T}_t + \hat{R}_t$ (or $\hat{A}_t = \hat{T}_t \hat{R}_t$) separately.
- Usually we assume $\hat{S}_t$ doesn't change over time, so we use a seasonal naïve method to forecast it.
- To forecast the seasonally adjusted component any non-seasonal forecasting method may be used

Evaluating point forecast accuracy

- We can evaluate the performance of a method via splitting the data into training vs test sets.



- We fit the model to the training set, and obtain forecasts for the test set.
- This allows us to obtain the **forecast error**

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T}$$

Evaluating point forecast accuracy

- We can use several accuracy measures, such as

$$\text{Mean absolute error: MAE} = \text{mean}(|e_t|)$$

$$\text{Root mean squared error: RMSE} = \sqrt{\text{mean}(e_t^2)}$$

- Be careful: these measures are *scale-dependent*, so they won't be comparable between time series measured in different units.
- Scale-invariant measures:

$$\text{Mean absolute percentage error: MAPE} = \text{mean}(|100e_t/y_t|)$$

$$\text{Mean absolute scaled error: MASE} = \text{mean}(|q_j|)$$

$$\text{Root mean squared scaled error: RMSSE} = \sqrt{\text{mean}(q_t^2)}$$

where

$$q_t = \frac{e_t}{\frac{1}{T-m}\sum_{j=m+1}^{T}(|y_j - y_{j-m}|)}$$

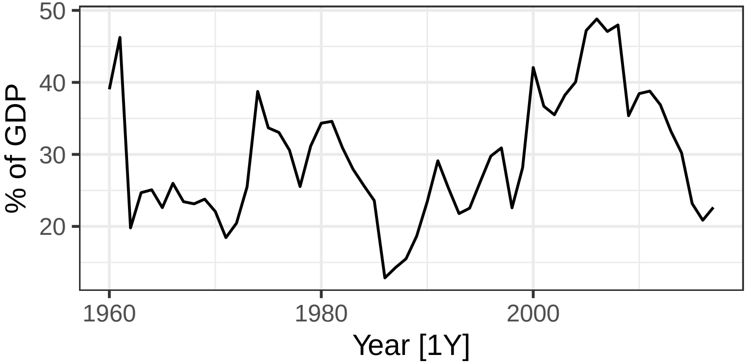
is a scaled version of the forecast error. (For a non-seasonal series simply take $m = 1$).

Exponential Smoothing

- Exponential smoothing was proposed in the late 1950s and has motivated some of the most successful forecasting methods.
- Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older.
- We will study the mechanics of the most important exponential smoothing methods and the underlying statistical models.

Simple Exponential Smoothing

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



Simple Exponential Smoothing

- The naïve method gives all weight to the last observation, whereas the mean method gives equal weight to all observations. SES is in between these two extremes.
- SES gives larger weights to more recent observations than to observations from the distant past:

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

where $0 \leq \alpha \leq 1$ is the smoothing parameter.

Simple Exponential Smoothing – Weighted Form

- The forecast at time $T+1$ is equal to a weighted average between the most recent observation y_T and the previous forecast $\hat{y}_{T|T-1}$:

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

- Let the first fitted value at time 1 be denoted by l_0 (which we will have to estimate). Then

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1-\alpha)^j y_{T-j} + (1-\alpha)^T l_0$$

Simple Exponential Smoothing – Component Form

- An alternative representation is the component form.
- For simple exponential smoothing, the only component included is the level, l_t.
- Component form representations of exponential smoothing methods comprise a forecast equation and a smoothing equation for each of the components included in the method.
- For SES we have

$$\begin{array}{lrcl} \text{Forecast equation:} & \hat{y}_{t+h|t} & = & l_t \\ \text{Smoothing equation:} & l_t & = & \alpha y_t + (1-\alpha) l_{t-1} \end{array}$$

- Here, l_t is the level (or the smoothed value) of the series at time t.
- Setting $h = 1$ gives the fitted values, while $t = T$ gives forecasts beyond the training data.

Estimation

- We estimate α and l_0 by minimising the sum of squared residuals

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

- This involves a non-linear optimisation problem, and we need an optimisation tool to solve it.

Exponential Smoothing with Trend

- Holt's linear trend method involves a forecast equation and two smoothing equations (one for level and one for trend):

$$\begin{aligned}
\text{Forecast equation:} \quad \hat{y}_{t+h|t} &= l_t + hb_t \\
\text{Level equation:} \quad l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \\
\text{Trend equation:} \quad b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}
\end{aligned}$$

- l_t is the level of the time series at time t
- b_t is the trend (slope) of the series at time t
- $0 \leq \alpha \leq 1$ is the smoothing parameter for the level
- $0 \leq \beta^* \leq 1$ is the smoothing parameter for the trend
- l_t is a weighted average between the last observartion and a one-step ahead forecast
- b_t is a weighted average between the estimated trend at time t and the previous estimate of trend b_{t-1}

Damped Trend Methods

- The forecasts generated by Holt's linear method display a constant trend (increasing or decreasing) indefinitely into the future.
- Empirical evidence indicates that these methods tend to over-forecast, especially for longer forecast horizons.
- We can introduce a parameter $0 < \phi < 1$ that "dampens" this trend to a flat line some time in the future:

$$\begin{aligned} \text{Forecast equation:} \quad \hat{y}_{t+h|t} &= l_t + (\phi + \phi^2 + \ldots + \phi^h) b_t \\ \text{Level equation:} \quad l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1}) \\ \text{Trend equation:} \quad b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1} \end{aligned}$$

- If $\phi = 1$ this is identical to Holt's method
- For $0 < \phi < 1$ the forecasts converge to $l_t + \frac{\phi b_t}{1-\phi}$ as $h \to \infty$

Methods with Seasonality

- The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations:

$$
\begin{aligned}
\text{Forecast equation:} \quad \hat{y}_{t+h|t} &= l_t + hb_t + s_{t+h-m(k+1)} \\
\text{Level equation:} \quad l_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\
\text{Trend equation:} \quad b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\
\text{Seasonality equation:} \quad s_t &= \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}
\end{aligned}
$$

- $0 \leq \gamma \leq 1 - \alpha$ is the smoothing parameter for the seasonal component s_t
- m is the period of the seasonality
- k is the integer part of $(h-1)/m$, which ensures that the estimates of the seasonal indices used for forecasting come from the final year of the sample
- This is the **Holt-Winters additive method**, and the seasonal component adds to zero

Methods with Seasonality

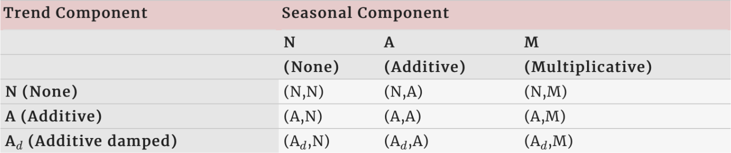
- When the seasonal variations change proportional to the level of the series, the **Holt-Winters multiplicative method** is preferred:

$$
\begin{aligned}
\text{Forecast equation:} \quad \hat{y}_{t+h|t} &= (l_t + hb_t)s_{t+h-m(k+1)} \\
\text{Level equation:} \quad l_t &= \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1}) \\
\text{Trend equation:} \quad b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\
\text{Seasonality equation:} \quad s_t &= \gamma \frac{y_t}{(l_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}
\end{aligned}
$$

- Here the seasonal component adds to m

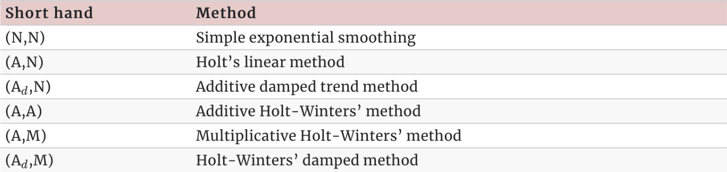
Combinations of component types

- We may combine different types of trend and seasonal components to generate nine exponential smoothing methods:



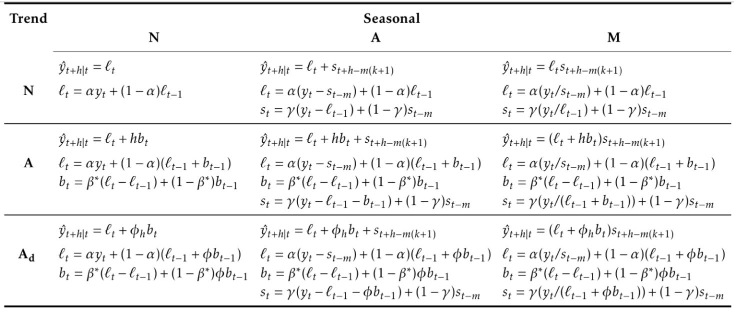
Trend Component	Seasonal Component		
	N	**A**	**M**
	(None)	**(Additive)**	**(Multiplicative)**
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
A_d (Additive damped)	(A_d,N)	(A_d,A)	(A_d,M)

- Some of those combinations have special names



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

Combinations of component types



Trend		Seasonal	
	N	**A**	**M**
N	$\hat{y}_{t+h\mid t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\mid t} = \ell_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\mid t} = \ell_t s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\mid t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\mid t} = \ell_t + hb_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\mid t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{A_d}$	$\hat{y}_{t+h\mid t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\mid t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\mid t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$

Combinations of component types

- These are examples of **state space models**
- Each model consists of a measurement equation that describes the observed data, and some state equations that describe how the unobserved components or states (level, trend, seasonal) change over time
- They can be labelled as ETS$(\cdot,\cdot,\cdot)$ for (Error, Trend, Seasonal) and a family of ETS models is available