ETS and ARIMA (Lecture 8)

Forecasting: principles and practice

book by Rob Hyndman and George Athanasopoulos

Review

For our course: Steps 3 to 5:

Step 3. Preliminary (exploratory) analysis. **Graphs! including graphs of time series decompositions**

Step 4. Choosing and fitting models.

Models:

-Simple models: Average models, naive model, seasonally naive model, random walk with drift (note: the series may be deseasonalized first and then re-seasonalized using stlf() or forecast())

-Regression models:

- linear regression models (with discussions of useful predictor variables including trend, seasonal variables, spike and step variables etc.)
- non-linear regression models (log-log functional form, piecewise linear trend model (change in slope), regression splines)
- choosing the best subset of regression (CV, AIC, AICc, BIC)
- ex-post forecast error vs ex-ante forecast error to compare whether the problem lies with the prediction of the prediction variable or with the fit of the regression model

-ETS models: These models specify the errors, trends and seasonality of the series.

Comparison of potential models:

Cross-validation (CV), ME, MAE, RMSE, MASE

Step 5. Using and evaluating a forecasting model.

checkresiduals(); forecasting()

```
library(fpp2)
library(tseries)
```

Taxonomy of exponential smoothing methods

Table 7.6: A two-way classification of 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)

Some of these methods we have already seen:

Each method is labelled by a pair of letters (T,S) defining the type of 'Trend' and 'Seasonal' components. For example, (A,M) is the method with an additive trend and multiplicative seasonality; $(\mathrm{A}_d, \mathrm{N})$ is the method with damped trend and no seasonality; and so on.

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

Short hand notation of methods already studied

The authors don't consider multiplicative trend methods as they produce poor forecast (Hydman et al. (2008))

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

Formula for recursive calculations and point forecast

The table gives the recursive formulae for applying the nine exponential smoothing methods. Each cell includes the forecast equation for generating h-step-ahead forecasts, and the smoothing equations for applying the method.

As discussed, ℓ_t denotes the series at time t, b_t denotes the slope at time t, s_t denotes the seasonal component of the series at time t, and m denotes the number of seasons in a year; α, β^*, γ and ϕ are smoothing parameters, $\phi_h = \phi + \phi^2 + \cdots + \phi_h$ and $h_m^+ = \lfloor (h-1)/m \rfloor + 1$

Innovations state space models for exponential smoothing

Each state space model can be labeled as ETS (Error, Trend, Seasonal). The possibilities for each component are:

Error ={A,M},

Trend ={N,A,A_d} and

Seasonal ={N,A,M},

where N =none, A = additive, M = multiplicative, and $_d$=damped.

The models can be identified using information criteria and estimated in R using the `ets()` function in the forecast package.

ETS(A,N,N): simple exponential smoothing with additive errors

Recall the component form in simple exponential smoothing:

$$\begin{aligned} \text{Forecast equation} \quad & \hat{y}_{t+1|t} = \ell_t \\ \text{Smoothing equation} \quad & \ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}, \end{aligned}$$

Error correction form

The third form of simple exponential smoothing is obtained by re-arranging the level equation in the component form to get what we refer to as the error correction form

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

where $e_t = y_t - \ell_{t-1} = y_t - \hat{y}_{t|t-1}$ for $t = 1, \ldots, T$. That is, e_t is the one-step within-sample forecast error at time t.

The "training-period" errors lead to the adjustment/correction of the estimated level throughout the smoothing process for $t = 1, \ldots, T$.

For example, if the error at time t is negative, then $\hat{y}_{t|t-1} > y_t$ and so the level at time $t-1$ has been over-estimated.

The new level ℓ_t is then the previous level ℓ_{t-1} adjusted downwards.

The closer α is to one the rougher the estimate of the level (large adjustments take place). The smaller the α the smoother the level (small adjustments take place).

We can also write $y_t = \ell_{t-1} + e_t$ (substitute into the last equation in the previous slide the smoothing equation).

To make this into an innovations state space model, all we need to do is specify the probability distribution of e_t. For a model with additive errors, we assume the one-step ahead forecast errors e_t to be normally distributed white noise with mean 0 and varianceσ^2 or

$$e_t \equiv \epsilon_t \sim NID(0, \sigma^2)$$

NID stands for "normally and independently distributed."

Hence, with additive errors:

$$\epsilon_t = y_t - \ell_{t-1}$$

ETS(A,N,N)

The equations in the model can then be rewritten as:

$$\begin{aligned} y_t &= \ell_{t-1} + \epsilon_t && \text{measurement (or observation) eq} \\ \ell_t &= \ell_{t-1} + \alpha\epsilon_t && \text{state (or transition) eq} \end{aligned}$$

These two equations, together with the statistical distribution of the errors, form a fully specified statistical model. Specifically, these constitute an innovations state space model underlying simple exponential smoothing.

The term "innovations" comes from the fact that all equations in this type of specification use the same random error process, ϵ_t.

The measurement equation shows the relationship between the observations and the unobserved states.

In this case, observation y_t is a linear function of the level ℓ_{t-1}, the predictable part of y_t, and the random error ϵ_t, the unpredictable part of y_t.

For other innovations state space models, this relationship may be nonlinear.

The transition equation shows the evolution of the state through time. The influence of the smoothing parameter α is the same as for the methods discussed earlier.

For example,α governs the degree of change in successive levels. The higher the value of α, the more rapid the changes in the level; the lower the value of α, the smoother the changes.

At the lowest extreme, where $\alpha = 0$, the level of the series does not change over time.

At the other extreme, where $\alpha = 1$, the model reduces to a random walk model, $y_t = y_{t-1} + \epsilon_t$.

i.e. $\epsilon_t = y_t - \ell_{t-1}$ so when $\alpha = 1$, $\ell_t = \ell_{t-1} + (y_t - \ell_{t-1})$ or $\ell_t = y_t$ and recursively, $\ell_{t-1} = y_{t-1}$

ETS(M,N,N): simple exponential smoothing with multiplicative errors

Similarly, we can specify models with multiplicative errors by writing one-step random errors as relative errors:

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

or

$$\ell_{t-1}\epsilon_t = y_t - \hat{y}_{t|t-1}$$

where $\epsilon_t \sim NID(0, \sigma^2)$. Substituting $\hat{y}_{t|t-1} = \ell_{t-1}$ gives the measurement or obs equation in ETS:

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

which can be rewritten as: $y_t - \ell_{t-1} = \ell_{t-1}\epsilon_t$.

Substituting into the *level equation* in component form $\ell_t = \ell_{t-1} + \alpha(y_t - \ell_{t-1})$ as before gives the state or transition equation.

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

Then the *multiplicative form* of the state space model (ETS(M,N,N)) simplifies to:

$$\begin{aligned} y_t &= \ell_{t-1}(1+\epsilon_t) && \text{measurement (or observation) eq} \\ \ell_t &= \ell_{t-1}(1+\alpha\epsilon_t) && \text{state (or transition) eq} \end{aligned}$$

ETS(A,A,N): Holt's linear method with additive errors

Holts linear method in component form:

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

For the ETS(A,A,N), we assume that the one-step forecast errors are given by

$$e_t = y_t - \ell_{t-1} - b_{t-1} \equiv \epsilon_t \sim NID(0, \sigma^2)$$

.

Substituting into the error correction equations for Holt's linear method form (level and seasonal components):

$$
\begin{aligned}
\text{Obs. or meas. eq} \quad & y_t = \ell_{t-1} + b_{t-1} + \epsilon_t \\
\text{State eq.} \quad & \ell_t = \ell_{t-1} + \alpha(y_t - \ell_{t-1} - b_{t-1}) + b_{t-1} => \ell_t = \ell_{t-1} + b_{t-1} + \alpha(\epsilon_t) \\
\text{State eq.} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} => b_t = \beta^*(\alpha\epsilon_t + b_{t-1}) + (1-\beta^*)b_{t-1}
\end{aligned}
$$

Gives:

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

where for simplicity, $\beta = \alpha\beta^*$

ETS(M,A,N): Holt's linear method with multiplicative errors

Specifying one-step forecast errors as relative errors such that

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

or

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

and following an approach similar to that used above,

Substituting this into the error correction equations for Holt's linear method form (level and seasonal components):

$$
\begin{aligned}
\text{Obs. or meas. eq} \quad & y_t = (\ell_{t-1} + b_{t-1})(1 + \epsilon_t) \\
\text{State eq.} \quad & \ell_t = \ell_{t-1} + \alpha(y_t - \ell_{t-1} - b_{t-1}) + b_{t-1} => \ell_t = \ell_{t-1} + b_{t-1} + \alpha((\ell_{t-1} + b_{t-1})\epsilon_t) \\
\text{State eq.} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} => b_t = \beta^*(b_{t-1} + \alpha((\ell_{t-1} + b_{t-1})\epsilon_t)) + (1 - \beta^*)b_{t-1}
\end{aligned}
$$

the innovations state space model underlying Holt's linear method with multiplicative errors (ETS(M,A,N)) is specified as:

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

where $\beta = \alpha\beta^*$ and $\epsilon_t \sim NID(0, \sigma^2)$

Other ETS models

The tables below show the equations for all the models in the ETS framework.

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_d	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$

State space equations for each of the models in the ETS with additive errors

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	$y_t = (\ell_{t-1}+b_{t-1})(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1}+b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}+\beta(\ell_{t-1}+b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1}+b_{t-1}+s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}+b_{t-1}+\alpha(\ell_{t-1}+b_{t-1}+s_{t-m})\varepsilon_t$ $b_t = b_{t-1}+\beta(\ell_{t-1}+b_{t-1}+s_{t-m})\varepsilon_t$ $s_t = s_{t-m}+\gamma(\ell_{t-1}+b_{t-1}+s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1}+b_{t-1})s_{t-m}(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1}+b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}+\beta(\ell_{t-1}+b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
$\mathbf{A_d}$	$y_t = (\ell_{t-1}+\phi b_{t-1})(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1}+\phi b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = \phi b_{t-1}+\beta(\ell_{t-1}+\phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1}+\phi b_{t-1}+s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}+\phi b_{t-1}+\alpha(\ell_{t-1}+\phi b_{t-1}+s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1}+\beta(\ell_{t-1}+\phi b_{t-1}+s_{t-m})\varepsilon_t$ $s_t = s_{t-m}+\gamma(\ell_{t-1}+\phi b_{t-1}+s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1}+\phi b_{t-1})s_{t-m}(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1}+\phi b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = \phi b_{t-1}+\beta(\ell_{t-1}+\phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$

State space equations for each of the models in the ETS with multiplicative errors

Estimating ETS models

ETS models are estimated in the forecasting package using the function *ets()* (more on the ets() function later.)

The ets() function estimates ETS models maximizing the "likelihood".

The likelihood is the probability of the data arising from the specified model. Thus, a large likelihood is associated with a good model.

Hence, using ets() function, we will estimate the smoothing parameters $\alpha, \beta, \gamma \text{ and } \phi$ and the initial states $\ell_0, b_0, s_0, s_{-1}, \cdots, s_{-m+1}$ by maximizing the likelihood.

The possible values that the smoothing parameters can take are restricted.

Traditionally, the parameters have been constrained to lie between 0 and 1 so that the equations can be interpreted as weighted averages.

That is, $0 < \alpha, \beta^*, \gamma^*, \phi < 1$.

For the state space models, we have set $\beta = \alpha\beta^*$.

Therefore, the traditional restrictions translate to $0 < \alpha < 1$, $0 < \beta < \alpha$.

In practice, the dampening parameter ϕ is usually constrained further to prevent numerical difficulties in estimating the model. In R, it is restricted so that $0.8 < \phi < 0.98$.

Model selection

A great advantage of the ETS statistical framework is that information criteria can calculated from the likelihood function L.

The AIC, AIC_C and BIC, introduced in Section 5.5, can be used here to determine which of the ETS models is most appropriate for a given time series.

For ETS models, Akaike's Information Criterion (AIC) is defined as

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

where L is the likelihood of the model and k is the total number of parameters and initial states that have been estimated (including the residual variance).

The AIC corrected for small sample bias (AIC_C) is defined as

$$\mathrm{AIC_c} = \mathrm{AIC} + \frac{k(k+1)}{T-k-1}.$$

and the Bayesian Information Criterion (BIC) is

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

Three of the combinations of (Error, Trend, Seasonal) can lead to numerical difficulties. Specifically, the models that can cause such instabilities are ETS(A,N,M), ETS(A,A,M), and ETS(A,A ,M). We normally do not consider these particular combinations when selecting a model.

For more information on these instabilities, please refer to the paper by Hydman and Akram (2006) available at (https://www.monash.edu/business/econometrics-and-business-statistics/research/publications/ebs/wp03-06.pdf).

The ets() function in R

The models can be estimated in R using the ets() function in the forecast package. Unlike the ses, holt and hw functions, the ets function does not produce forecasts. Rather, it estimates the model parameters and returns information about the fitted model.

The R code below shows the most important arguments that this function can take, and their default values. If only the time series is specified, and all other arguments are left at their default values, then an appropriate model will be selected automatically. The arguments are explained below. See the Rdocumentation on ets for a more complete description (https://www.rdocumentation.org/packages/forecast/versions/8.1/topics/ets).

```
ets(y, model="ZZZ", damped=NULL, alpha=NULL, beta=NULL, gamma=NULL,
phi=NULL, lambda=NULL, biasadj=FALSE, additive.only=FALSE,
restrict=TRUE, allow.multiplicative.trend=FALSE)
```

y

The time series to be forecast.

model

A three-letter code indicating the model to be estimated using the ETS classification and notation. The possible inputs are “N” for none, “A” for additive, “M” for multiplicative, or “Z” for automatic selection. If any of the inputs is left as “Z”, then this component is selected according to the information criterion chosen. The default value of ZZZ ensures that all components are selected using the information criterion.

damped

If damped=TRUE , then a damped trend will be used (either A or M). If damped=FALSE , then a nondamped trend will used. If damped=NULL (the default), then either a damped or a non-damped trend will be selected, depending on which model has the smallest value for the information criterion.

alpha , beta , gamma , phi

The values of the smoothing parameters can be specified using these arguments. If they are set to NULL (the default setting for each of them), the parameters are estimated.

lambda

Box-Cox transformation parameter. It will be ignored if lambda=NULL (the default value). Otherwise, the time series will be transformed before the model is estimated. When lambda is not NULL, additive.only is set to TRUE.

biasadj

If TRUE and lambda is not NULL, then the back-transformed fitted values and forecasts will be biasadjusted.

additive.only

Only models with additive components will be considered if additive.only=TRUE . Otherwise, all models will be considered.

restrict

If restrict=TRUE (the default), the models that cause numerical difficulties are not considered in model selection.

allow.multiplicative.trend

Multiplicative trend models are also available, but not covered in the textbook. Set this argument to TRUE to allow these models to be considered.

Working with ets objects

The ets() function will return an object of class ets. There are many R functions designed to make working with ets objects easy. A few of them are described below.

coef()

returns all fitted parameters.

accuracy()

returns accuracy measures computed on the training data.

summary()

prints some summary information about the fitted model.

autoplot() and plot()

produce time plots of the components.

residuals()

returns residuals from the estimated model.

fitted()

returns one-step forecasts for the training data.

simulate()

will simulate future sample paths from the fitted model.

forecast() computes point forecasts and prediction intervals, as described in the next section.

Example International tourist visitor nights in Australia

the ETS statistical framework is used to forecast tourist visitor nights in Australia by international arrivals over the period 2016-2019. The ets() function selects the model by minimizing the AICc.

```
aust <- window(austourists, start=2005)
fit <- ets(aust)
summary(fit)
```

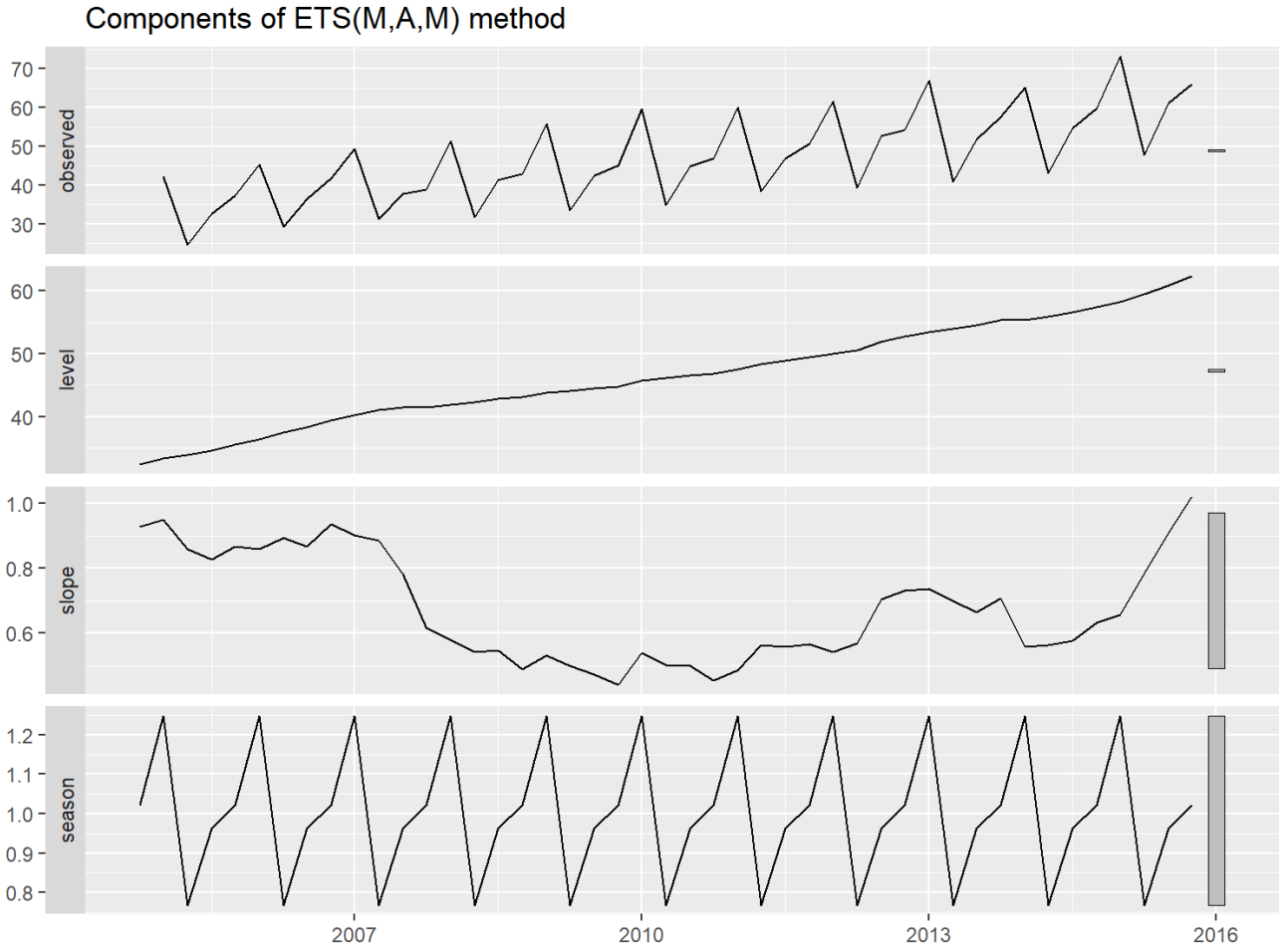
```
## ETS(M,A,M)
##
## Call:
##  ets(y = aust)
##
##   Smoothing parameters:
##     alpha = 0.1908
##     beta  = 0.0392
##     gamma = 2e-04
##
##   Initial states:
##     l = 32.3679
##     b = 0.9281
##     s = 1.0218 0.9628 0.7683 1.2471
##
##   sigma:  0.0383
##
##      AIC     AICc      BIC
## 224.8628 230.1569 240.9205
##
## Training set error measures:
##                       ME     RMSE     MAE        MPE     MAPE     MASE
## Training set 0.04836907 1.670893 1.24954 -0.1845609 2.692849 0.409454
##                    ACF1
## Training set 0.2005962
```

From the results above, ets() chooses the model: ETS(M,A,M):

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

From the output above, the parameter estimates are $\alpha = 0.198$, $\beta = 0.0319$ and $\gamma = 0.00019$. The output also returns ℓ_0, b_0, s_0, s_{-1}, s_{-2} and s_{-3}.

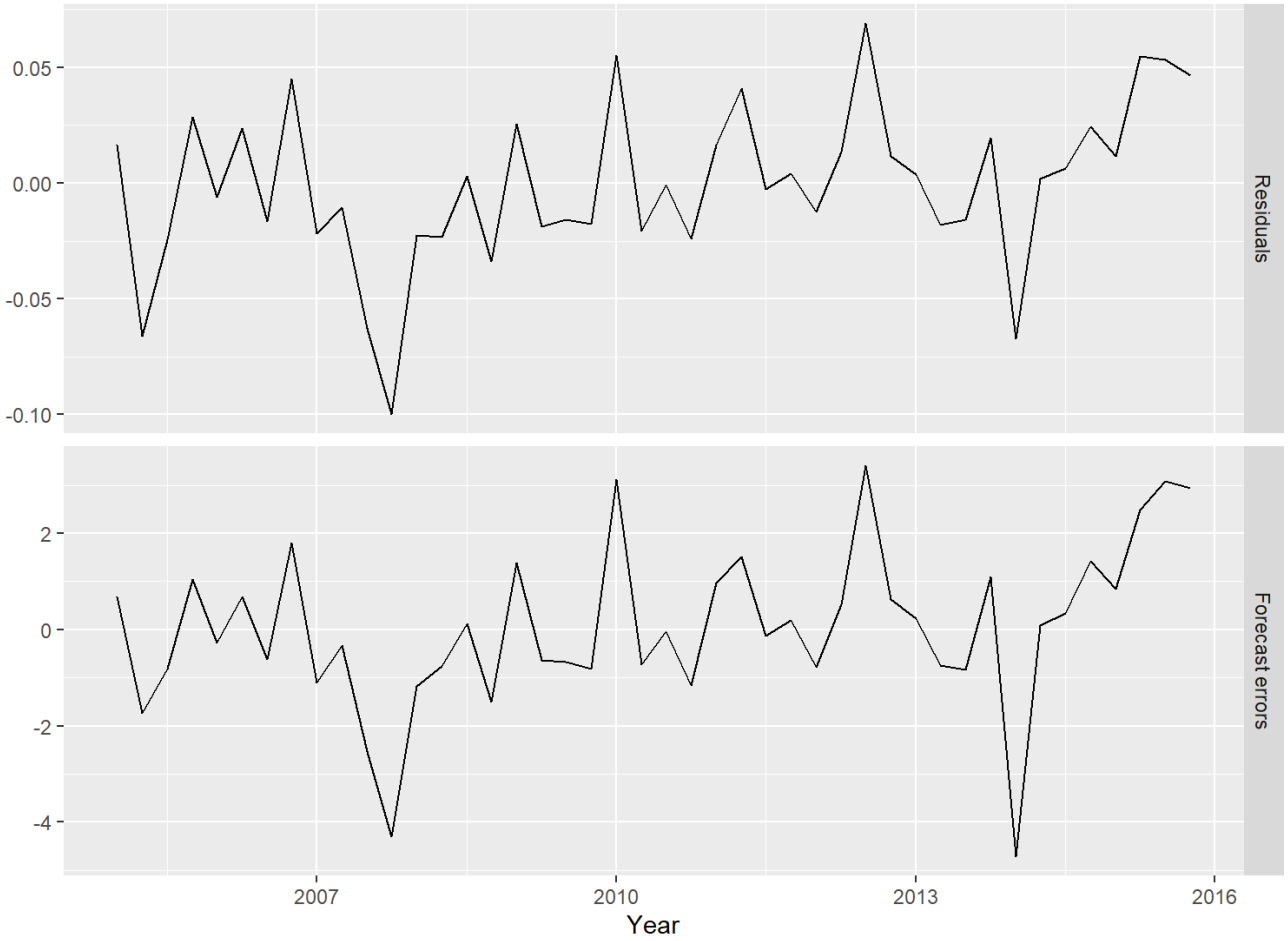
```
autoplot(fit)
```



Note that the ETS(M,A,M) has multiplicative errors. The residuals $\hat{\epsilon}_t$ and the one-step ahead forecast errors $y_t - \hat{y}_t$ could be obtained by the residuals function:

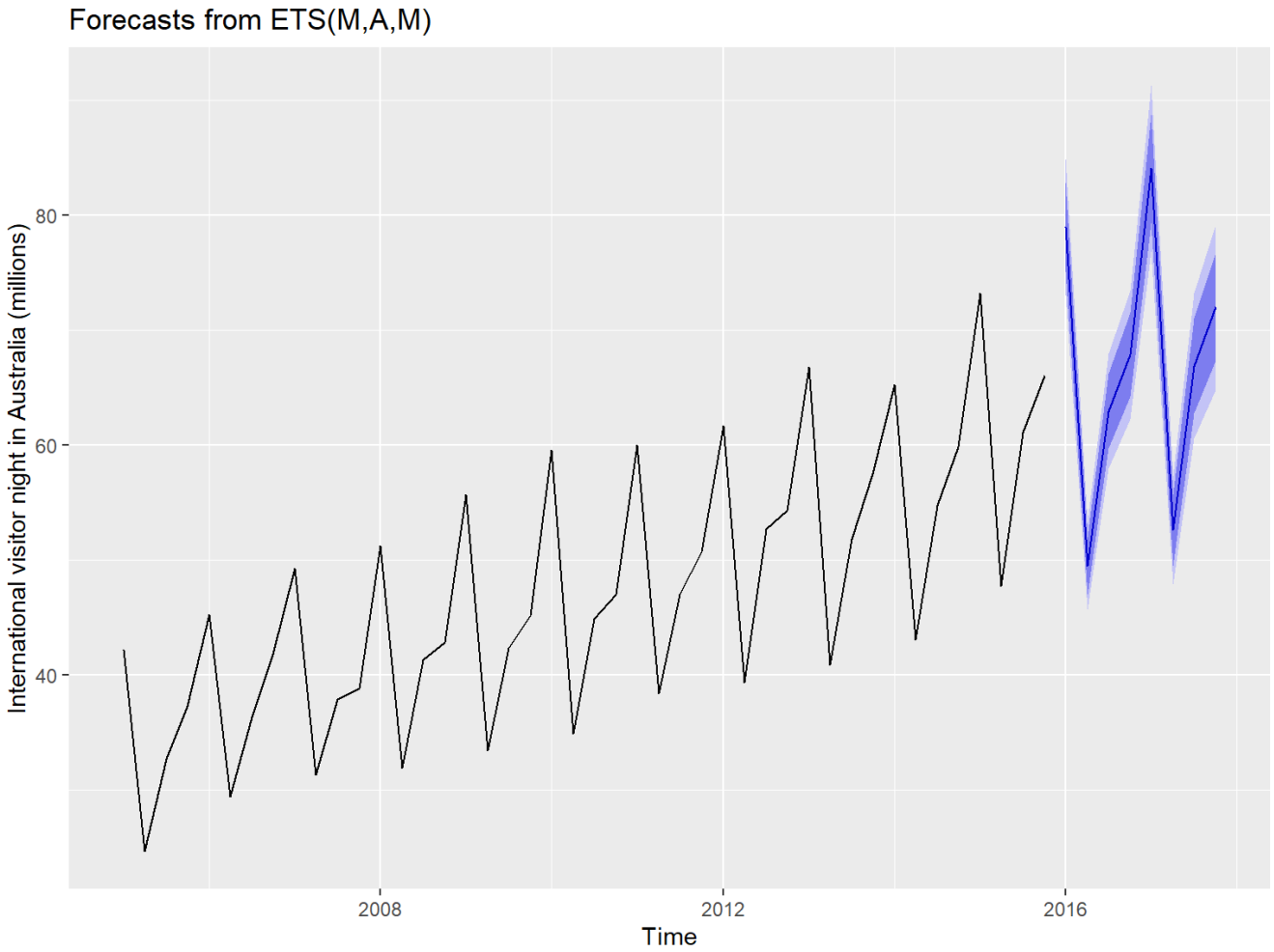
```
cbind('Residuals' = residuals(fit),
       'Forecast errors' = residuals(fit, type='response')) %>%
 autoplot(facet=TRUE) + xlab("Year") + ylab("")
```

The type argument is used in the residuals function to distinguish between residuals and forecast errors. The default is type='innovation' which gives regular residuals


0.05
0.00
-0.05
-0.10
Residuals
2
0
-2
-4
Forecast errors
2007
2010
2013
2016
Year

The next slide shows the point forecast and prediction intervals generated from the model. The small values of β and γ mean that the slope and seasonal components change very little over time. The narrow prediction intervals indicate that the series is relatively easy to forecast due to the strong trend and seasonality.

```
fit %>% forecast(h=8) %>%
autoplot() +
ylab("International visitor night in Australia (millions)")
```


Forecasts from ETS(M,A,M)
International visitor night in Australia (millions)
80
60
40
2008
2012
2016
Time

Forecasting with ETS models

For Example: ETS(M,A,N)

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

Point forecasts are obtained from the models by iterating the equations for $t = T + 1, \cdots, T + h$ and setting all $\epsilon = 0$ for $t > T$.

For model ETS(M,A,N), $y_{T+1} = (\ell_T + b_T)(1 + \epsilon_{T+1})$. Therefore $\hat{y}_{T+1|T} = \ell_T + b_T$

Similarly,

$$\begin{aligned} y_{T+2} &= (\ell_{T+1} + b_{T+1})(1 + \epsilon_{T+1}) \\ &= [(\ell_T + b_T)(1 + \alpha\epsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\epsilon_{T+1}](1 + \epsilon_{T+1}) \end{aligned}$$

Therefore, $\hat{y}_{T+2|T} = \ell_T + 2b_T$ and so on.

ETS point forecasts are equal to the medians of the forecast distributions. For models with only additive components, the forecast distributions are normal, so the medians and means are equal.

For ETS models with multiplicative errors, or with multiplicative seasonality, the point forecasts will not be equal to the means of the forecast distributions.

To obtain forecasts from an ETS model, we use the forecast function.

The R code below shows the possible arguments that this function takes when applied to an ETS model. We explain each of the arguments in what follows.

The forecast command:

```
forecast(object, h=ifelse(object$m>1, 2*object$m, 10),
level=c(80,95), fan=FALSE, simulate=FALSE, bootstrap=FALSE, npaths=5000,
PI=TRUE, lambda=object$lambda, biasadj=NULL, ...)
```

object

The object returned by the ets() function.

h

The forecast horizon - the number of periods to be forecast.

level

The confidence level for the prediction intervals.

fan

If fan=TRUE, this is suitable for fan plots.

simulate

If simulate=TRUE , prediction intervals are produced by simulation rather than using algebraic formulae. Simulation will also be used (even if simulate=FALSE) where there are no algebraic formulae available for the particular model.

bootstrap

If bootstrap=TRUE and simulate=TRUE , then the simulated prediction intervals use re-sampled errors rather than normally distributed errors.

npaths

The number of sample paths used in computing simulated prediction intervals.

PI

If PI=TRUE, then prediction intervals are produced; otherwise only point forecasts are calculated.

lambda

The Box-Cox transformation parameter. This is ignored if lambda=NULL . Otherwise, the forecasts are back-transformed via an inverse Box-Cox transformation.

biasadj

If lambda is not NULL , the backtransformed forecasts (and prediction intervals) are bias-adjusted.

Prediction intervals

The prediction intervals will differ between models with additive and multiplicative methods.

For most ETS models, a prediction interval can be written as

$$\hat{y}_{T+h|T} \pm k\sigma_h$$

where k depends on the coverage probability, and σ_h is the forecast variance. Values for k were given in Table 3.1 based on the normal distribution.

For ETS models, the formula for σ_h can be complicated; the details are given in Chapter 6 of Hyndman et al. (2008).

In the table below, the formula given are for the additive ETS models, which are the simplest. In the table, the forecast variance expressions for each additive state space model are shown, where σ^2 is the residual variance, m is the seasonal period, $h_m = \lfloor (h-1)/m \rfloor$ and $\lfloor u \rfloor$ denote the integer part of u.

Model	Forecast variance: σ_h
(A,N,N)	$\sigma_h = \sigma^2\big[1+\alpha^2(h-1)\big]$
(A,A,N)	$\sigma_h = \sigma^2\Big[1+(h-1)\big\{\alpha^2+\alpha\beta h+\frac{1}{6}\beta^2 h(2h-1)\big\}\Big]$
(A,A_d,N)	$\sigma_h = \sigma^2\Big[1+\alpha^2(h-1)+\frac{\beta\phi h}{(1-\phi)^2}\{2\alpha(1-\phi)+\beta\phi\}$
	$-\frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)}\big\{2\alpha(1-\phi^2)+\beta\phi(1+2\phi-\phi^h)\big\}\Big]$
(A,N,A)	$\sigma_h = \sigma^2\Big[1+\alpha^2(h-1)+\gamma h_m(2\alpha+\gamma)\Big]$
(A,A,A)	$\sigma_h = \sigma^2\Big[1+(h-1)\big\{\alpha^2+\alpha\beta h+\frac{1}{6}\beta^2 h(2h-1)\big\}+\gamma h_m\big\{2\alpha+\gamma+\beta m(h_m+1)\big\}\Big]$
(A,A_d,A)	$\sigma_h = \sigma^2\Big[1+\alpha^2(h-1)+\frac{\beta\phi h}{(1-\phi)^2}\{2\alpha(1-\phi)+\beta\phi\}$
	$-\frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)}\big\{2\alpha(1-\phi^2)+\beta\phi(1+2\phi-\phi^h)\big\}$
	$+\gamma h_m(2\alpha+\gamma)+\frac{2\beta\gamma\phi}{(1-\phi)(1-\phi^m)}\big\{h_m(1-\phi^m)-\phi^m(1-\phi^{mh_m})\big\}\Big]$

Forecast variance expressions for each additive state model

For a few ETS models, there are no known formula for prediction intervals. In these cases, the forecast.ets function uses simulated future sample paths and computes prediction intervals from the percentiles of these simulated future paths.

ARIMA models

What ARIMA stands for:

- A series that need to be differenced to made stationary is an "integrated" (I) series.
- Lags of stationarized series are called "autoregressive" (AR) terms
- Lags of the forecast errors are called "moving average" (MA) terms

Construction of an ARIMA model for forecasting

1. Plot the data. Identify any unusual observations.
2. If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.
3. If the data are non-stationary: take first differences of the data until the data are stationary.
4. Examine the ACF/PACF: Is an AR(p) or MA(q) model appropriate?
5. Try your chosen model(s), and use the AICc to search for a better model.
6. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
7. Once the residuals look like white noise, calculate forecasts.

ARIMA terminology

A non-seasonal ARIMA model can be (almost) completely summarized by three numbers:

p = the number of autoregressive terms

d = the number of nonseasonal differences

q = the number of moving-average terms

- This is called an "ARIMA(p,d,q)" model
- The model may also include a constant term (or not)

ARIMA models we've already met

- ARIMA(0,0,0)+c = mean (constant) model (average method)
- ARIMA(0,1,0) = Random Walk model (naive method)
- ARIMA(0,1,0)+c = Random Walk with drift model (naive drift method)
- ARIMA(1,0,0)+c = regress Y_t on Y_{t-1}

Exponential smoothing vs ARIMA models

ARIMA models provide another approach to time series forecasting. Exponential smoothing and ARIMA models are the two most widely-used approaches to time series forecasting, and provide complementary approaches to the problem.

While exponential smoothing models were based on a description of trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data.

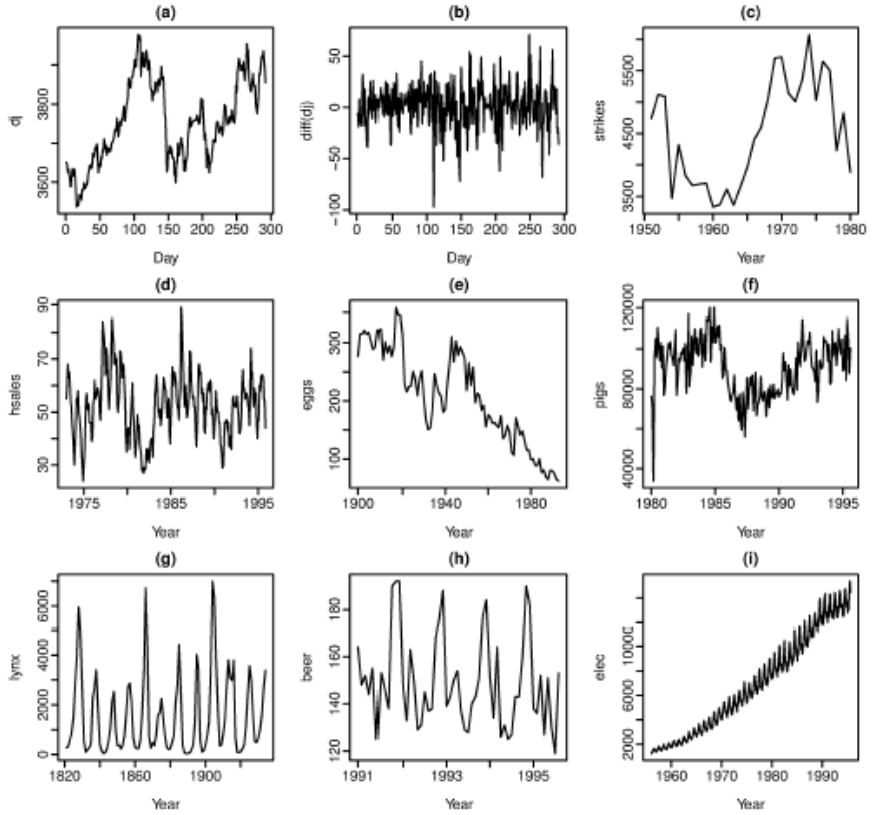
Stationarity

A stationary time series is one whose properties do not depend on the time at which the series is observed.

So time series with trends, or with seasonality, are not stationary.

Some cases can be confusing – a time series with cyclic behaviour (but not trend or seasonality) is stationary. That is because the cycles are not of fixed length, so before we observe the series we cannot be sure where the peaks and troughs of the cycles will be.

In general, a stationary time series will have no obvious observable patterns in the long-term.



Which of the series are stationary?

(a) Dow Jones index on 292 consecutive days; (b) Daily change in the Dow Jones index on 292 consecutive days; (c) Annual number of strikes in the US; (d) Monthly sales of new one-family houses sold in the US; (e) Annual price of a dozen eggs in the US (constant dollars); (f) Monthly total of pigs slaughtered in Victoria, Australia; (g) Annual total of lynx trapped in the McKenzie River district of north-west Canada; (h) Monthly Australian beer production; (i) Monthly Australian electricity production.

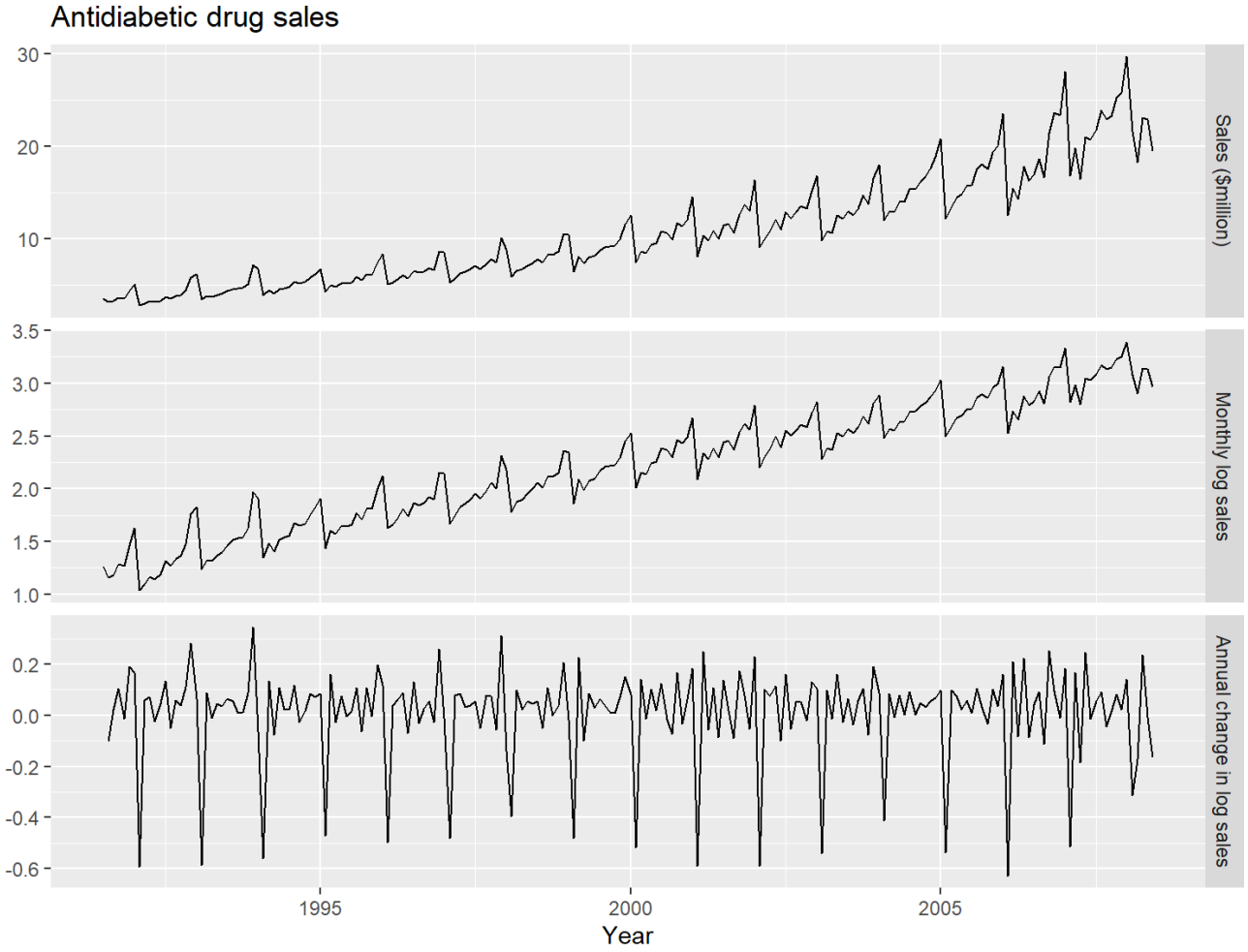
Differencing

One way to make a time series stationary is to compute the differences between consecutive observations. This is known as differencing.

Transformations such as logarithms can help to stabilize the variance of a time series.

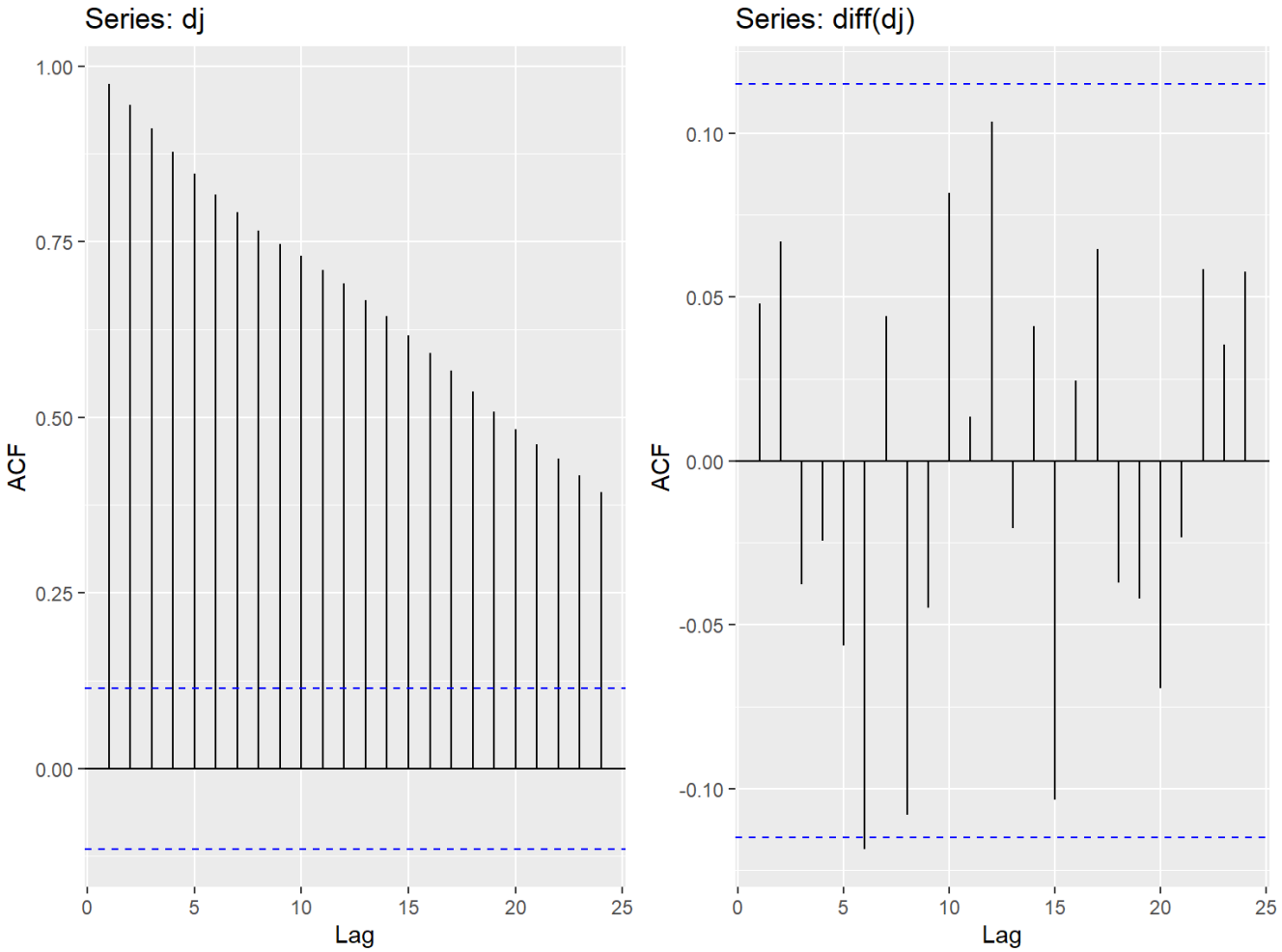
Differencing can help stabilize the mean of a time series by removing changes in the level of a time series, and eliminating or reducing the trend and the seasonality.

```
cbind("Sales ($million)" = a10,
"Monthly log sales" = log(a10),
"Annual change in log sales" = diff(log(a10),1)) %>%
autoplot(facets=TRUE) +
xlab("Year") + ylab("") +
ggtitle("Antidiabetic drug sales")
```



Besides looking at the time plot of the data, the ACF plot is also useful for identifying non-stationary time series.
ACF of Dow in levels and diffs

```
library(ggplot2)
library(gridExtra)
grid.arrange(ggAcf(dj), ggAcf(diff(dj)), ncol=2)
```



For a stationary time series, the ACF will drop to zero relatively quickly, while the ACF of non-stationary data decreases slowly. Also, for non-stationary data, the value of autocorrelation coefficient r_1 is often large and positive.

```
Box.test(diff(dj), lag=10, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  diff(dj)
## X-squared = 14.461, df = 10, p-value = 0.153
```

The ACF of the differenced Dow-Jones index looks just like that of a white noise series. There is only one autocorrelation lying just outside the 95% limits, and the Ljung-Box Q^* statistic has a p-value of 0.153 (for $h = 10$). This suggests that the daily change in the Dow-Jones index is essentially a random amount which is uncorrelated with that of previous days.

Random walk model

The differenced series is the change between consecutive observations in the original series, and can be written as

$$y'_t = y_t - y_{t-1}.$$

The differenced series will have only $T-1$ values since it is not possible to calculate a difference y'_1 for the first observation.

When the differenced series is white noise, the model for the original series can be written as

$$y_t - y_{t-1} = e_t \quad \text{or} \quad y_t = y_{t-1} + e_t \ .$$

A random walk model is very widely used for non-stationary data, particularly financial and economic data. Random walks typically have:

- long periods of apparent trends up or down
- sudden and unpredictable changes in direction.

The forecasts from a random walk model are equal to the last observation, as future movements are unpredictable, and are equally likely to be up or down. Thus, the random walk model underpins naive forecasts.

A closely related model allows the differences to have a non-zero mean. Then

$$y_t - y_{t-1} = c + e_t \quad \text{or} \quad y_t = c + y_{t-1} + e_t \ .$$

The value of c is the average of the changes between consecutive observations. If c is positive, then the average change is an increase in the value of y_t. Thus y_t will tend to drift upwards. But if c is negative, y_t will tend to drift downwards.

This is the model behind the drift method.

Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time to obtain a stationary series:

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2}. \end{aligned}$$

In this case, y''_t will have $T-2$ values. Then we would model the *change in the changes* of the original data. In practice, it is almost never necessary to go beyond second-order differences.

Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year. So

$$y'_t = y_t - y_{t-m} \qquad \text{where } m = \text{number of seasons.}$$

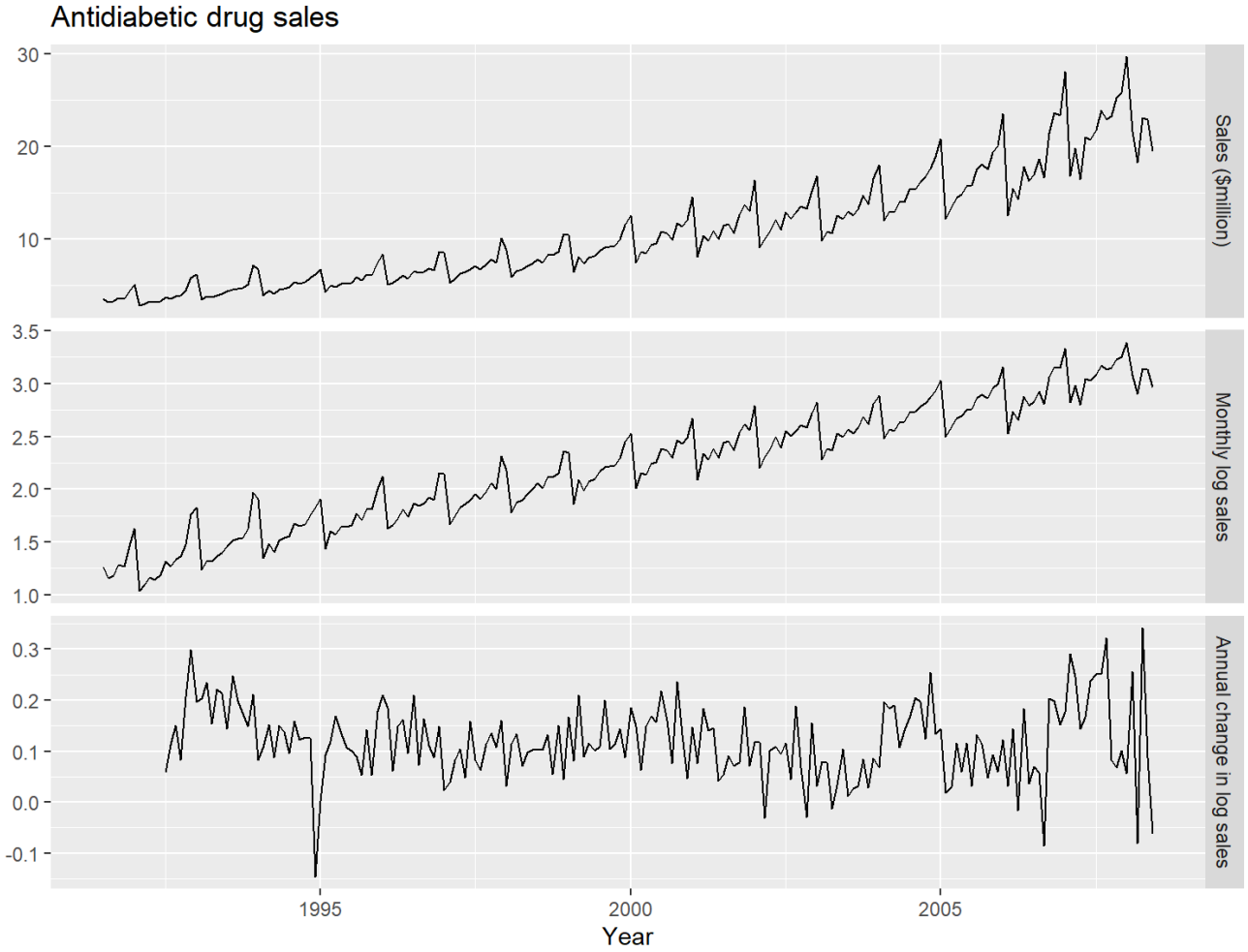
These are also called *lag-m differences* as we subtract the observation after a lag of m periods.

If seasonally differenced data appear to be white noise, then an appropriate model for the original data is

$$y_t = y_{t-m} + e_t.$$

Forecasts from this model are equal to the last observation from the relevant season. That is, this model gives seasonal naive forecasts.

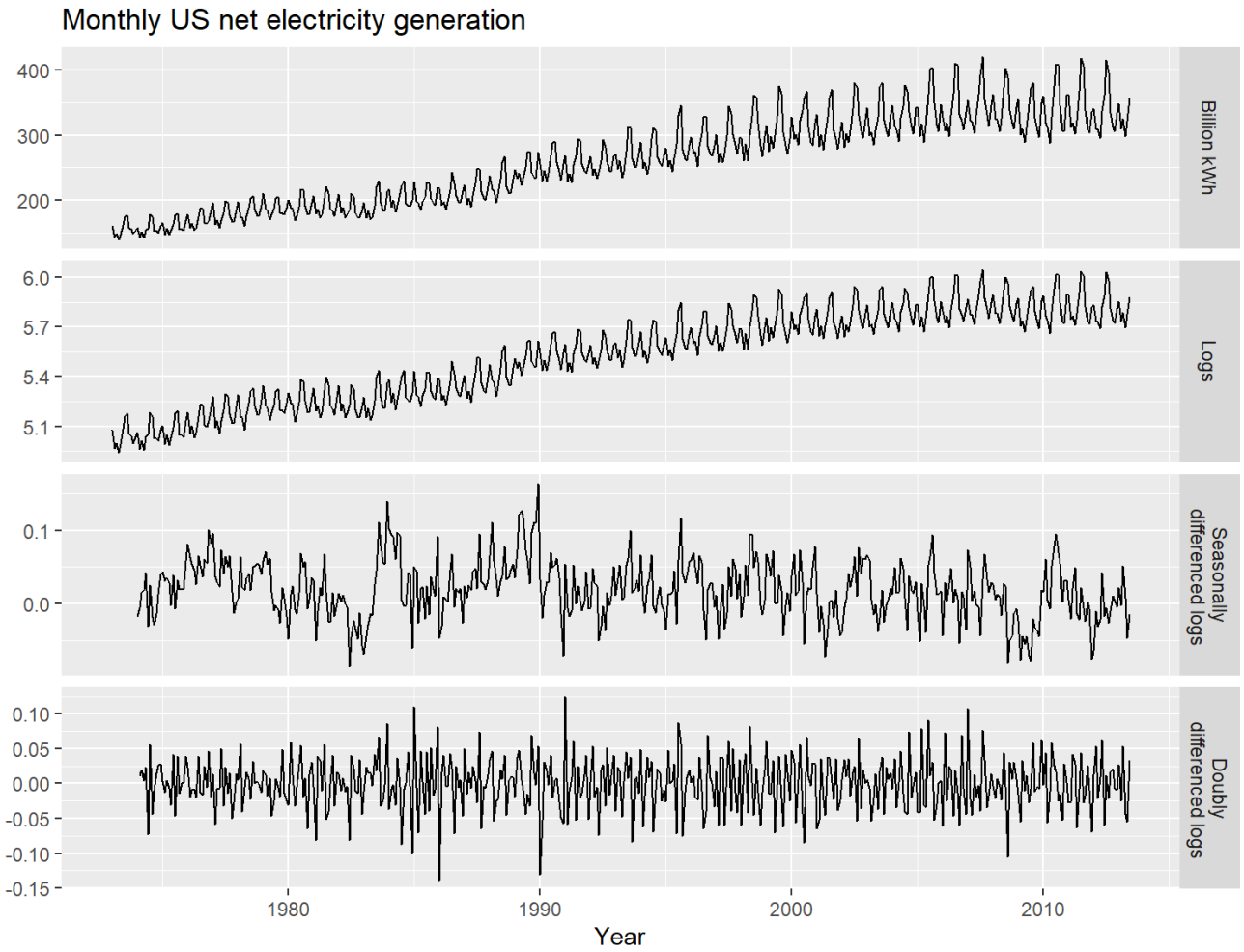
```
cbind("Sales ($million)" = a10,
"Monthly log sales" = log(a10),
"Annual change in log sales" = diff(log(a10),12)) %>%
autoplot(facets=TRUE) +
xlab("Year") + ylab("") +
ggtitle("Antidiabetic drug sales")
```



Logs and seasonal differences of the A10 (antidiabetic) sales data. The logarithms stabilize the variance, while the seasonal differences remove the seasonality and trend.

Sometimes it is necessary to do both a seasonal difference and a first difference to obtain stationary data, as is shown in figure in the next slide. Here, the data are first transformed using logarithms (second panel), then seasonal differences are calculated (third panel). The data still may seem a little non-stationary, and so further first differences are computed (bottom panel).

```
cbind("Billion kWh" = usmelec,
"Logs" = log(usmelec),
"Seasonally\n differenced logs" = diff(log(usmelec),12),
"Doubly\n differenced logs" = diff(diff(log(usmelec),12),1)) %>%
autoplot(facets=TRUE) +
xlab("Year") + ylab("") +
ggtitle("Monthly US net electricity generation")
```


Monthly US net electricity generation
400
300
200
Billion kWh
6.0
5.7
5.4
5.1
Logs
0.1
0.0
Seasonally differenced logs
0.10
0.05
0.00
-0.05
-0.10
-0.15
Doubly differenced logs
1980
1990
2000
2010
Year

There is a degree of subjectivity in selecting which differences to apply. The seasonally differenced data in Figure on "Antidiabetic drug sales" do not show substantially different behaviour from the seasonally differenced data in Figure "Monthly US net electricity generation".

In the latter case, we could have decided to stop with the seasonally differenced data, and not done an extra round of differencing. In the former case, we could have decided that the data were not sufficiently stationary and taken an extra round of differencing. Some formal tests for differencing will be discussed later, but there are always some choices to be made in the modelling process, and different analysts may make different choices.

If $y'_t = y_t - y_{t-m}$ denotes a seasonally differenced series, then the twice-differenced series is

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-m}) - (y_{t-1} - y_{t-m-1}) \\ &= y_t - y_{t-1} - y_{t-m} + y_{t-m-1}\,. \end{aligned}$$

Or

If $y'_t = y_t - y_{t-1}$ denotes first differenced series, then the twice-(seasonally) differenced series is

$$\begin{aligned} y''_t &= y'_t - y'_{t-m} \\ &= (y_t - y_{t-1}) - (y_{t-m} - y_{t-m-1}) \\ &= y_t - y_{t-1} - y_{t-m} + y_{t-m-1}\,. \end{aligned}$$

When both seasonal and first differences are applied, it makes no difference which is done first the result will be the same.

However, if the data have a strong seasonal pattern, it is recommended that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for a further first difference. If first differencing is done first, there will still be seasonality present.

It is important that if differencing is used, the differences are interpretable. First differences are the change between one observation and the next. Seasonal differences are the change between one year to the next. Other lags are unlikely to make much interpretable sense and should be avoided.

Unit root tests

One way to determine more objectively if differencing is required is to use a unit root test. These are statistical hypothesis tests of stationarity that are designed for determining whether differencing is required.

The usual hypothesis tests for regression coefficients do not work when the data are non-stationary.

A number of unit root tests are available, and they are based on different assumptions and may lead to conflicting answers.

The ADF test

One of the most popular tests is the Augmented Dickey-Fuller (ADF) test.

For this test, the following regression model is estimated:

$$y'_t = \alpha + \beta t + \phi y_{t-1} + \beta_1 y'_{t-1} + \beta_2 y'_{t-2} + \cdots + \beta_k y'_{t-k},$$

where y'_t denotes the first-differenced series, $y'_t = y_t - y_{t-1}$ and k is the number of lags to include in the regression.

If the original series, y_t, needs differencing, then the coefficient $\hat{\phi}$ should be approximately zero. If y_t is already stationary, then $\hat{\phi} < 0$.

The following R function from the tseries package carries out the ADF test

```
adf.test(x, alternative = "stationary")
```

In R, the default value of k is set to $\lfloor (T-1)^{1/3} \rfloor$, where T is set to the length of the time series $\lfloor x \rfloor$ means the largest integer not greater than x.

The null-hypothesis for an ADF test is that the data are non-stationary. So large p-values are indicative of non-stationarity, and small p-values suggest stationarity. Using the usual 5% threshold, differencing is required if the p-value is greater than 0.05.

Details of the adf.test (along with other commands in tseries package) could be found by downloading (https://cran.r-project.org/web/packages/tseries/tseries.pdf)

One of the main criticisms of the ADF test is its low power when ϕ is less than but close to 1, i.e., $\phi = 0.95$

The KPSS test

Another popular unit root test is the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. This reverses the hypotheses, so the null-hypothesis is that the data are stationary. In this case, small p-values (e.g., less than 0.05) suggest that differencing is required. The kpss.test is also from the tseries package:

```
kpss.test(x)
```

Power of ADF and KPSS tests

ADF and KPSS tests have low power with short span of data and against alternative hypothesis with time series which are stationary but exhibit non-linearities.

Backshift notation

The backward shift operator B is a useful notational device when working with time series lags:

$$By_t = y_{t-1}.$$

(Some references use L for lag instead of B for backshift.) In other words, B, operating on y_t, has the effect of shifting the data back one period. Two applications of B to y_t shifts the data back two periods:

$$B(By_t) = B^2 y_t = y_{t-2}.$$

For monthly data, if we wish to consider *the same month last year*, the notation is $B^{12}y_t$ = y_{t-12}.

The backward shift operator is convenient for describing the process of differencing. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1-B)y_t \ .$$

Note that a first difference is represented by $(1-B)$. Similarly, if second-order differences have to be computed, then:

$$y''_t = y_t - 2y_{t-1} + y_{t-2} = (1-2B+B^2)y_t = (1-B)^2y_t \ .$$

In general, a dth-order difference can be written as

$$(1-B)^d y_t.$$

Backshift notation is very useful when combining differences as the operator can be treated using ordinary algebraic rules. In particular, terms involving B can be multiplied together.

For example, a seasonal difference followed by a first difference can be written as

$$\begin{aligned}(1-B)(1-B^m)y_t &= (1-B-B^m+B^{m+1})y_t \\ &= y_t - y_{t-1} - y_{t-m} + y_{t-m-1},\end{aligned}$$

the same result we obtained earlier.

Thank you for your attention