

3.6 The forecast package in R

This book uses the facilities in the **forecast** package in R (which is loaded automatically whenever you load the **fpp2** package). This appendix briefly summarises some of the features of the package. Please refer to the help files for individual functions to learn more, and to see some examples of their use.

Functions that output a forecast object:

Many functions, including `meanf()`, `naive()`, `snaive()` and `rwf()`, produce output in the form of a `forecast` object (i.e., an object of class `forecast`). This allows other functions (such as `autoplot()`) to work consistently across a range of forecasting models.

Objects of class `forecast` contain information about the forecasting method, the data used, the point forecasts obtained, prediction intervals, residuals and fitted values. There are several functions designed to work with these objects including `autoplot()`, `summary()` and `print()`.

The following list shows all the functions that produce `forecast` objects.

- `meanf()`
- `naive()`, `snaive()`
- `rwf()`
- `croston()`
- `stlf()`
- `ses()`
- `holt()`, `hw()`
- `splinef()`
- `thetaf()`
- `forecast()`

`forecast()` function

So far we have used functions which produce a `forecast` object directly. But a more common approach, which we will focus on in the rest of the book, will be to fit a model to the data, and then use the `forecast()` function to produce forecasts from that model.

The `forecast()` function works with many different types of inputs. It generally takes a time series or time series model as its main argument, and produces forecasts appropriately. It always returns objects of class `forecast`.

If the first argument is of class `ts`, it returns forecasts from the automatic ETS algorithm discussed in Chapter 7.

Here is a simple example, applying `forecast()` to the `ausbeer` data:

```
forecast(ausbeer, h=4)
#>           Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
#> 2010 Q3           404.6 385.9 423.3 376.0 433.3
#> 2010 Q4           480.4 457.5 503.3 445.4 515.4
#> 2011 Q1           417.0 396.5 437.6 385.6 448.4
#> 2011 Q2           383.1 363.5 402.7 353.1 413.1
```

That works quite well if you have no idea what sort of model to use. But by the end of this book, you should not need to use `forecast()` in this “blind” fashion. Instead, you will fit a model appropriate to the data, and then use `forecast()` to produce forecasts from that model.