3.5 Prediction intervals

As discussed in Section 1.7, a prediction interval gives an interval within which we expect y_t to lie with a specified probability. For example, assuming that the forecast errors are normally distributed, 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.

More generally, a prediction interval can be written as

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

where the multiplier c depends on the coverage probability. In this book we usually calculate 80% intervals and 95% intervals, although any percentage may be used. The following table gives the value of c for a range of coverage probabilities assuming normally distributed forecast errors.

Table 3.1: Multipliers to be used for prediction intervals.

Percentage	Multiplier
50	0.67
55	0.76
60	0.84
65	0.93
70	1.04
75	1.15
80	1.28
85	1.44
90	1.64
95	1.96
96	2.05
97	2.17
98	2.33
99	2.58

The value of prediction intervals is that they express the uncertainty in the forecasts. If we only produce point forecasts, there is no way of telling how accurate the forecasts are. However, if we also produce prediction intervals, then it is clear how much uncertainty is associated with each forecast. For this reason, point forecasts can be of almost no value without the accompanying prediction intervals.

One-step prediction intervals

When forecasting one step ahead, the standard deviation of the forecast distribution is almost the same as the standard deviation of the residuals. (In fact, the two standard deviations are identical if there are no parameters to be estimated, as is the case with the naïve method. For forecasting methods involving parameters to be estimated, the standard deviation of the forecast distribution is slightly larger than the residual standard deviation, although this difference is often ignored.)

For example, consider a naïve forecast for the Google stock price data `goog200` (shown in Figure 3.5). The last value of the observed series is 531.48, so the forecast of the next value of the GSP is 531.48. The standard deviation of the residuals from the naïve method is 6.21. Hence, a 95% prediction interval for the next value of the GSP is

$$531.48 \pm 1.96(6.21) = [519.3, 543.6].$$

Similarly, an 80% prediction interval is given by

$$531.48 \pm 1.28(6.21) = [523.5, 539.4].$$

The value of the multiplier (1.96 or 1.28) is taken from Table 3.1.

Multi-step prediction intervals

A common feature of prediction intervals is that they increase in length as the forecast horizon increases. The further ahead we forecast, the more uncertainty is associated with the forecast, and thus the wider the prediction intervals. That is, σ_h usually increases with h (although there are some non-linear forecasting methods that do not have this property).

To produce a prediction interval, it is necessary to have an estimate of σ_h. As already noted, for one-step forecasts ($h = 1$), the residual standard deviation provides a good estimate of the forecast standard deviation σ_1. For multi-step forecasts, a more complicated method of calculation is required. These calculations assume that the residuals are uncorrelated.

Benchmark methods

For the four benchmark methods, it is possible to mathematically derive the forecast standard deviation under the assumption of uncorrelated residuals. If $\hat{\sigma}_h$ denotes the standard deviation of the h-step forecast distribution, and $\hat{\sigma}$ is the residual standard deviation, then we can use the following expressions.

Mean forecasts: $\hat{\sigma}_h = \hat{\sigma}\sqrt{1 + 1/T}$

Naïve forecasts: $\hat{\sigma}_h = \hat{\sigma}\sqrt{h}$

Seasonal naïve forecasts $\hat{\sigma}_h = \hat{\sigma}\sqrt{k + 1}$, where k is the integer part of $(h - 1)/m$ and m is the seasonal period.

Drift forecasts: $\hat{\sigma}_h = \hat{\sigma}\sqrt{h(1 + h/T)}$.

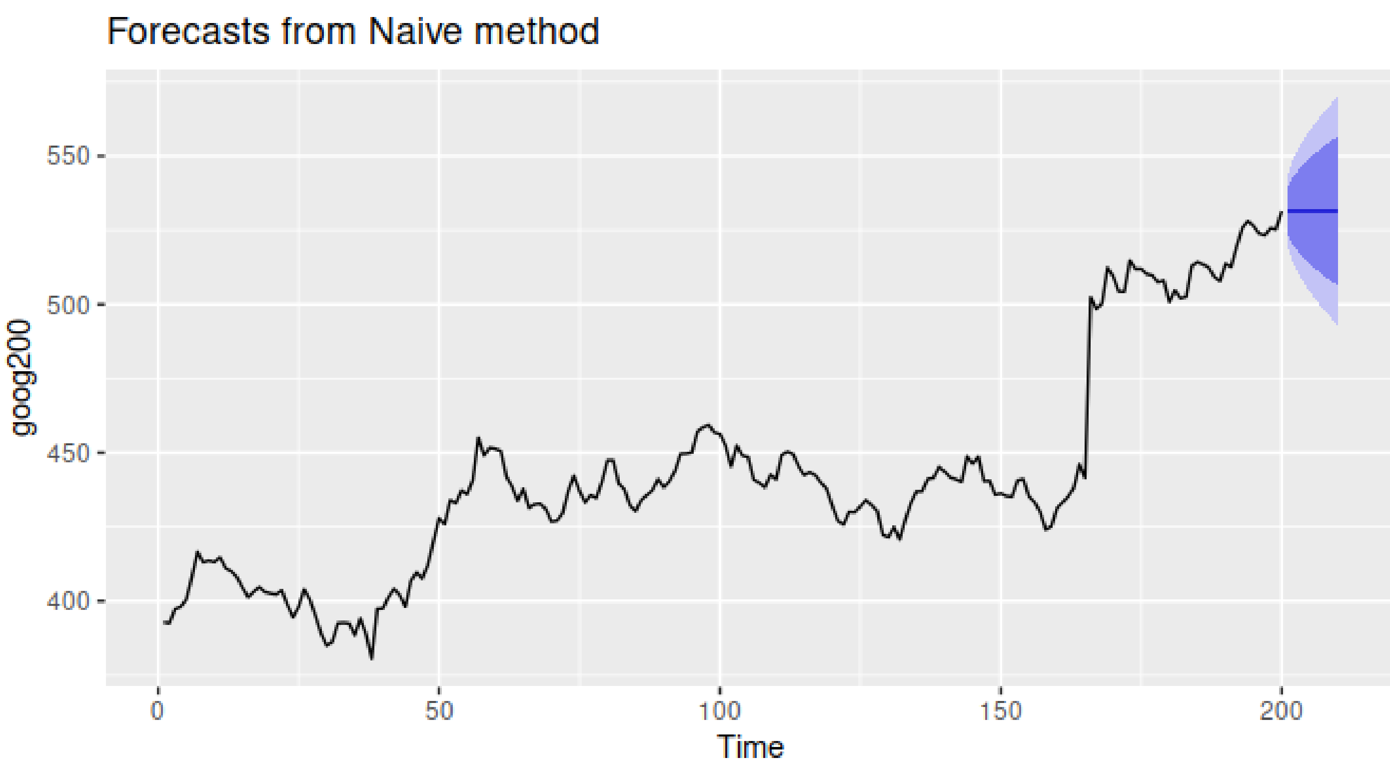
Note that when $h = 1$ and T is large, these all give the same approximate value $\hat{\sigma}$.

Prediction intervals will be computed for you when using any of the benchmark forecasting methods. For example, here is the output when using the naïve method for the Google stock price.

```
naive(goog200)
#>     Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
#> 201          531.5 523.5 539.4 519.3 543.6
#> 202          531.5 520.2 542.7 514.3 548.7
#> 203          531.5 517.7 545.3 510.4 552.6
#> 204          531.5 515.6 547.4 507.1 555.8
#> 205          531.5 513.7 549.3 504.3 558.7
#> 206          531.5 512.0 551.0 501.7 561.3
#> 207          531.5 510.4 552.5 499.3 563.7
#> 208          531.5 509.0 554.0 497.1 565.9
#> 209          531.5 507.6 555.3 495.0 568.0
#> 210          531.5 506.3 556.6 493.0 570.0
```

When plotted, the prediction intervals are shown as shaded region, with the strength of colour indicating the probability associated with the interval.

```
autoplot(naive(goog200))
```



Prediction intervals from bootstrapped residuals

When a normal distribution for the forecast errors is an unreasonable assumption, one alternative is to use bootstrapping, which only assumes that the forecast errors are uncorrelated.

A forecast error is defined as $e_t = y_t - \hat{y}_{t|t-1}$. We can re-write this as

$$y_t = \hat{y}_{t|t-1} + e_t.$$

So we can simulate the next observation of a time series using

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

where $\hat{y}_{T+1|T}$ is the one-step forecast and e_{T+1} is the unknown future error. Assuming future errors will be similar to past errors, we can replace e_{T+1} by sampling from the collection of errors we have seen in the past (i.e., the residuals). Adding the new simulated observation to our data set, we can repeat the process to obtain

$$y_{T+2} = \hat{y}_{T+2|T+1} + e_{T+2}$$

where e_{T+2} is another draw from the collection of residuals. Continuing in this way, we can simulate an entire set of future values for our time series.

Doing this repeatedly, we obtain many possible futures. Then we can compute prediction intervals by calculating percentiles for each forecast horizon. The result is called a **bootstrapped** prediction interval. The name "bootstrap" is a reference to pulling ourselves up by our bootstraps, because the process allows us to measure future uncertainty by only using the historical data.

To generate such intervals, we can simply add the `bootstrap` argument to our forecasting functions. For example:

```
naive(goog200, bootstrap=TRUE)
#>     Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
#> 201          531.5 525.7 537.8 522.9 542.9
#> 202          531.5 523.2 539.5 519.4 547.0
#> 203          531.5 520.9 541.2 516.7 552.3
#> 204          531.5 519.0 543.0 514.0 560.3
#> 205          531.5 517.5 544.6 511.8 582.1
#> 206          531.5 516.1 545.9 509.5 582.4
#> 207          531.5 514.8 547.3 508.0 583.5
#> 208          531.5 513.5 548.9 505.8 584.9
#> 209          531.5 512.3 549.8 503.9 586.6
#> 210          531.5 510.7 551.4 502.1 587.5
```

In this case, they are similar (but not identical) to the prediction intervals based on the normal distribution.

Prediction intervals with transformations

If a transformation has been used, then the prediction interval should be computed on the transformed scale, and the end points back-transformed to give a prediction interval on the original scale. This approach preserves the probability coverage of the prediction interval, although it will no longer be symmetric around the point forecast.

The back-transformation of prediction intervals is done automatically using the functions in the **forecast** package in R, provided you have used the `lambda` argument when computing the forecasts.