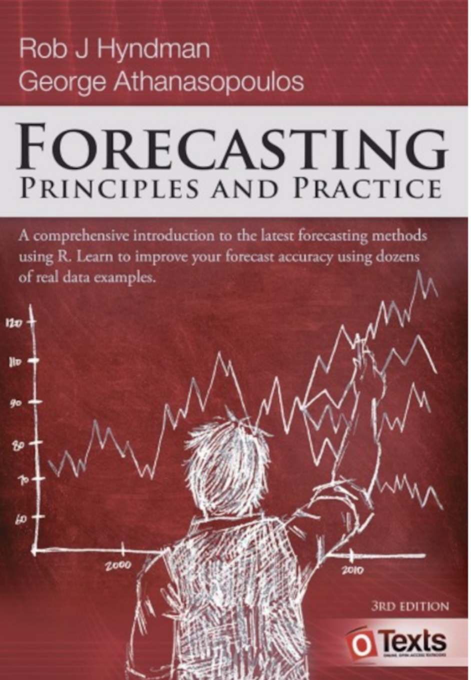


5. The forecaster's toolbox

5.9 Evaluating distributional accuracy

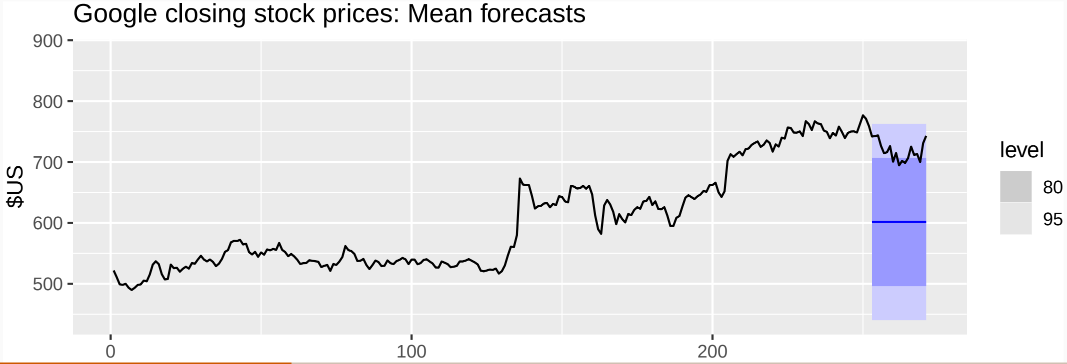
OTexts.org/fpp3/

Google closing stock prices

```
google_stock <- gafa_stock |>
  filter(Symbol == "GOOG", year(Date) >= 2015) |>
  mutate(day = row_number()) |>
  update_tsibble(index = day, regular = TRUE)
google_2015 <- google_stock |>
  filter(Symbol == "GOOG", year(Date) == 2015)
google_jan_2016 <- google_stock |>
  filter(Symbol == "GOOG", yearmonth(Date) == yearmonth("2016 Jan"))
google_fit <- google_2015 |>
  model(
    Mean = MEAN(Close),
    `Naïve` = NAIVE(Close),
    Drift = RW(Close ~ drift())
  )
google_fc <- google_fit |>
  forecast(google_jan_2016)
```

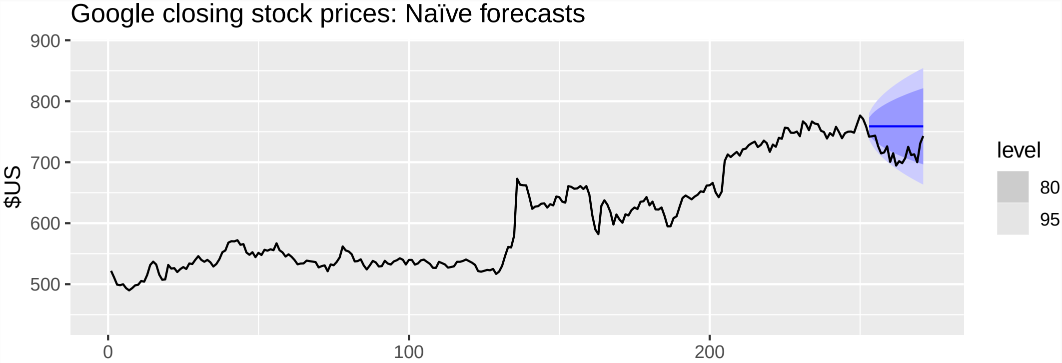
Google closing stock prices

```
google_fc |>
  filter(.model == "Mean") |>
  autoplot(bind_rows(google_2015, google_jan_2016)) +
  labs(y = "$US", title = "Google closing stock prices: Mean forecasts") +
  guides(colour = guide_legend(title = "Forecast")) + ylim(439,880)
```



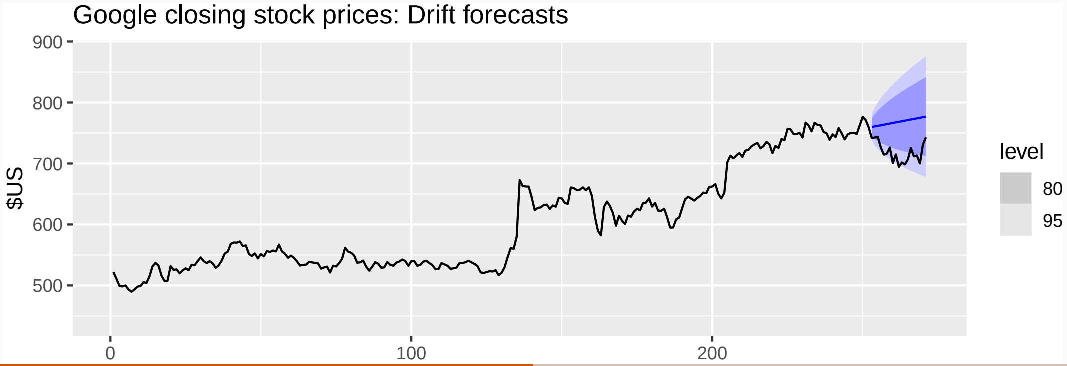
Google closing stock prices

```r
google_fc |>
  filter(.model == "Naïve") |>
  autoplot(bind_rows(google_2015, google_jan_2016)) +
  labs(y = "$US", title = "Google closing stock prices: Naïve forecasts") +
  guides(colour = guide_legend(title = "Forecast")) + ylim(439,880)
```



Google closing stock prices

```r
google_fc |>
  filter(.model == "Drift") |>
  autoplot(bind_rows(google_2015, google_jan_2016)) +
  labs(y = "$US", title = "Google closing stock prices: Drift forecasts") +
  guides(colour = guide_legend(title = "Forecast")) + ylim(439,880)
```



Evaluating quantile forecasts

$f_{p,t}$ = quantile forecast with prob. p at time t.

y_t = observation at time t

Expect probability($y_t < f_{p,t}$) = p

Evaluating quantile forecasts

$f_{p,t}$ = quantile forecast with prob. p at time t.

y_t = observation at time t

Expect probability$(y_t < f_{p,t}) = p$

Quantile score

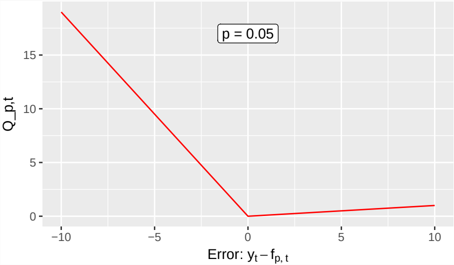
$$Q_{p,t} = \begin{cases} 2(1-p)|y_t - f_{p,t}|, & \text{if } y_t < f_{p,t} \\ 2p|y_t - f_{p,t}|, & \text{if } y_t \geq f_{p,t} \end{cases}$$

Evaluating quantile forecasts

$f_{p,t}$ = quantile forecast with prob. p at time t.

y_t = observation at time t

Expect probability$(y_t < f_{p,t}) = p$



Quantile score

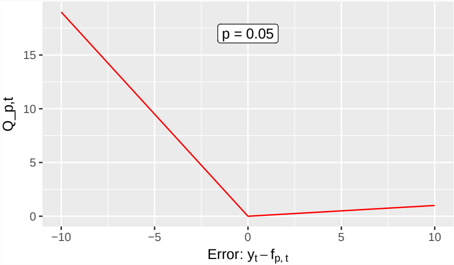
$$Q_{p,t} = \begin{cases} 2(1-p)|y_t - f_{p,t}|, & \text{if } y_t < f_{p,t} \\ 2p|y_t - f_{p,t}|, & \text{if } y_t \geq f_{p,t} \end{cases}$$

Evaluating quantile forecasts

$f_{p,t}$ = quantile forecast with prob. p at time t.

y_t = observation at time t

Expect probability$(y_t < f_{p,t}) = p$



Quantile score

$$Q_{p,t} = \begin{cases} 2(1-p)\big|y_t - f_{p,t}\big|, & \text{if } y_t < f_{p,t} \\ 2p\big|y_t - f_{p,t}\big|, & \text{if } y_t \ge f_{p,t} \end{cases}$$

- Low $Q_{p,t}$ is good
- Multiplier of 2 often omitted, but useful for interpretation
- $Q_{p,t}$ like absolute error (weighted to account for likely exceedance)

Quantile scores

```
google_fc |>
  filter(.model == "Naïve", Date == "2016-01-04") |>
  accuracy(google_stock, list(qs=quantile_score), probs=0.1)
```

```
## # A tibble: 1 x 4
##   .model Symbol .type    qs
##   <chr>  <chr>  <chr> <dbl>
## 1 Naïve  GOOG   Test   4.86
```

```
google_fc |>
  filter(.model == "Naïve", Date == "2016-01-04") |>
  accuracy(google_stock, list(qs=quantile_score), probs=0.9)
```

```
## # A tibble: 1 x 4
##   .model Symbol .type    qs
##   <chr>  <chr>  <chr> <dbl>
## 1 Naïve  GOOG   Test   6.28
```

Winkler Score

For $100(1-\alpha)\%$ prediction interval: $[\ell_{\alpha,t}, u_{\alpha,t}]$.

$$W_{\alpha,t} = \frac{Q_{\alpha/2,t} + Q_{1-\alpha/2,t}}{\alpha} = \begin{cases} (u_{\alpha,t} - \ell_{\alpha,t}) + \frac{2}{\alpha}(\ell_{\alpha,t} - y_t) & \text{if } y_t < \ell_{\alpha,t} \\ (u_{\alpha,t} - \ell_{\alpha,t}) & \text{if } \ell_{\alpha,t} \le y_t \le u_{\alpha,t} \\ (u_{\alpha,t} - \ell_{\alpha,t}) + \frac{2}{\alpha}(y_t - u_{\alpha,t}) & \text{if } y_t > u_{\alpha,t}. \end{cases}$$

```r
google_fc |>
  filter(.model == "Naïve", Date == "2016-01-04") |>
  accuracy(google_stock, list(winkler = winkler_score), level = 80)
```

```
## # A tibble: 1 x 4
##   .model Symbol .type winkler
##   <chr>  <chr>  <chr>   <dbl>
## 1 Naïve  GOOG   Test     55.7
```

Continuous Ranked Probability Score

Average quantile scores over all values of p to obtain the

Continuous Ranked Probability Score or CRPS.

```
google_fc |>
  accuracy(google_stock, list(crps = CRPS))
```

```
## # A tibble: 3 x 4
##   .model Symbol .type  crps
##   <chr>  <chr>  <chr> <dbl>
## 1 Drift  GOOG   Test   33.5
## 2 Mean   GOOG   Test   76.7
## 3 Naïve  GOOG   Test   26.5
```

Scale-free comparisons using skill scores

Skill scores provide a forecast accuracy measure relative to some benchmark method (often the naïve method).

$$\text{CRPS\_SS}_{\text{Method}} = \frac{\text{CRPS}_{\text{Naïve}} - \text{CRPS}_{\text{Method}}}{\text{CRPS}_{\text{Naïve}}}.$$

```r
google_fc |>
  accuracy(google_stock, list(skill = skill_score(CRPS)))
```

```
## # A tibble: 3 x 4
##   .model Symbol .type  skill
##   <chr>  <chr>  <chr>  <dbl>
## 1 Drift  GOOG   Test  -0.266
## 2 Mean   GOOG   Test  -1.90
## 3 Naïve  GOOG   Test   0
```