

Time Series Analysis & Forecasting Using R

6. Introduction to forecasting



Outline

- 1 What can we forecast?
- 2 The statistical forecasting perspective
- 3 Benchmark methods
- 4 Lab Session 11
- 5 Residual diagnostics
- 6 Lab Session 12
- 7 Forecast accuracy measures
- 8 Lab Session 13

Outline

1 What can we forecast?

2 The statistical forecasting perspective

3 Benchmark methods

4 Lab Session 11

5 Residual diagnostics

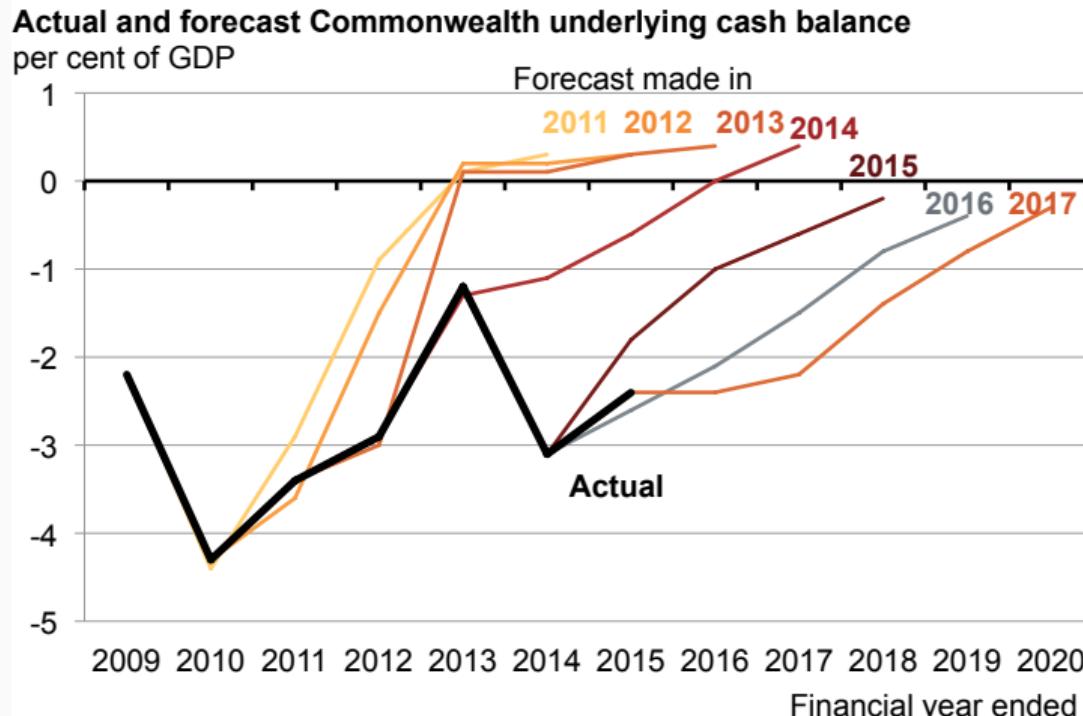
6 Lab Session 12

7 Forecast accuracy measures

8 Lab Session 13

Forecasting is difficult

Commonwealth plans to drift back to surplus **GRATTAN**
Institute



What can we forecast?



What can we forecast?



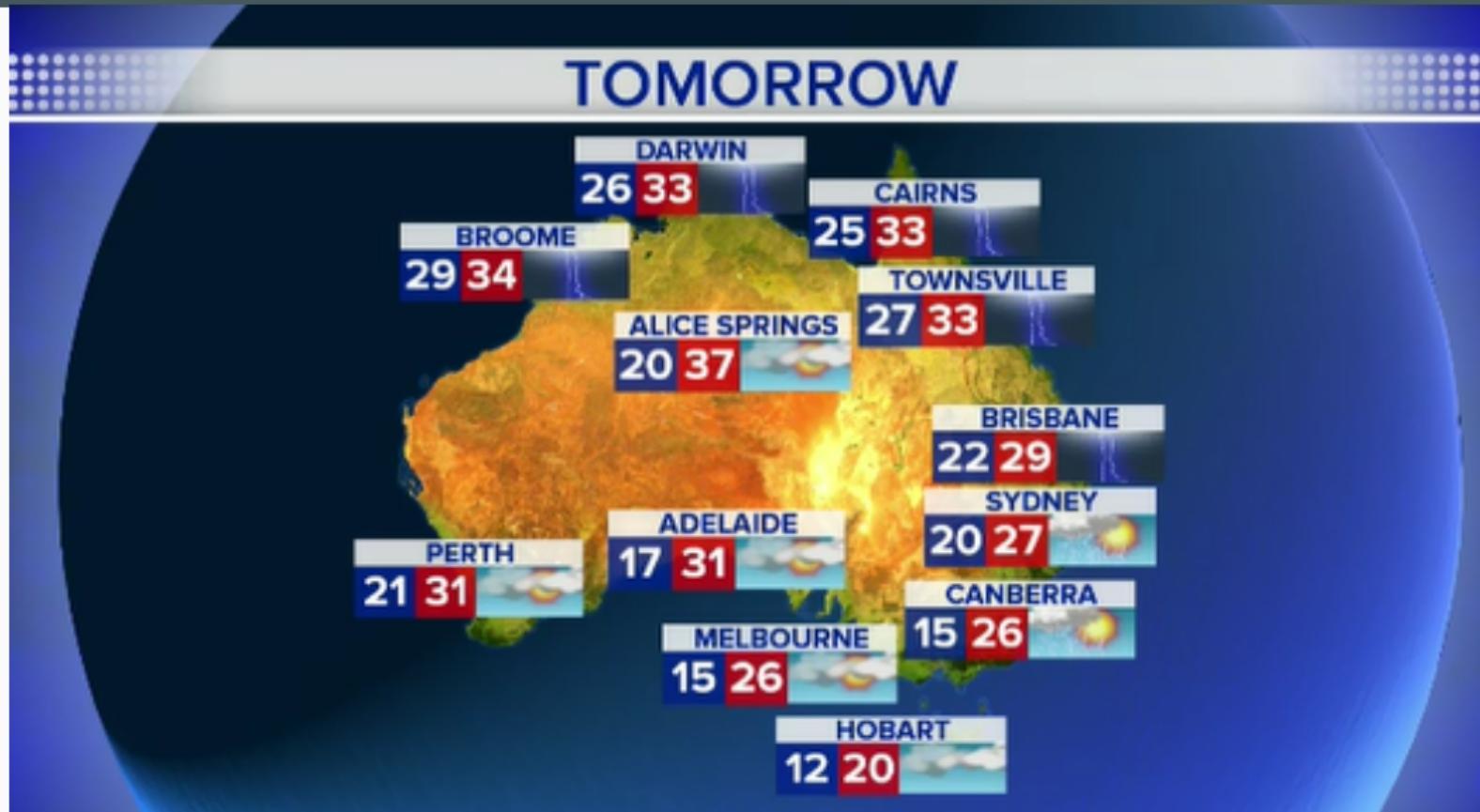
What can we forecast?



What can we forecast?



What can we forecast?



What can we forecast?



What can we forecast?



Which is easiest to forecast?

- 1 daily electricity demand in 3 days time
- 2 timing of next Halley's comet appearance
- 3 time of sunrise this day next year
- 4 Google stock price tomorrow
- 5 Google stock price in 6 months time
- 6 maximum temperature tomorrow
- 7 exchange rate of \$US/AUS next week
- 8 total sales of drugs in Australian pharmacies next month

Which is easiest to forecast?

- 1 daily electricity demand in 3 days time
 - 2 timing of next Halley's comet appearance
 - 3 time of sunrise this day next year
 - 4 Google stock price tomorrow
 - 5 Google stock price in 6 months time
 - 6 maximum temperature tomorrow
 - 7 exchange rate of \$US/AUS next week
 - 8 total sales of drugs in Australian pharmacies next month
-
- how do we measure “easiest”?
 - what makes something easy/difficult to forecast?

Factors affecting forecastability

Something is easier to forecast if:

- we have a good understanding of the factors that contribute to it
- there is lots of data available;
- the forecasts cannot affect the thing we are trying to forecast.
- there is relatively low natural/unexplainable random variation.
- the future is somewhat similar to the past

Outline

1 What can we forecast?

2 The statistical forecasting perspective

3 Benchmark methods

4 Lab Session 11

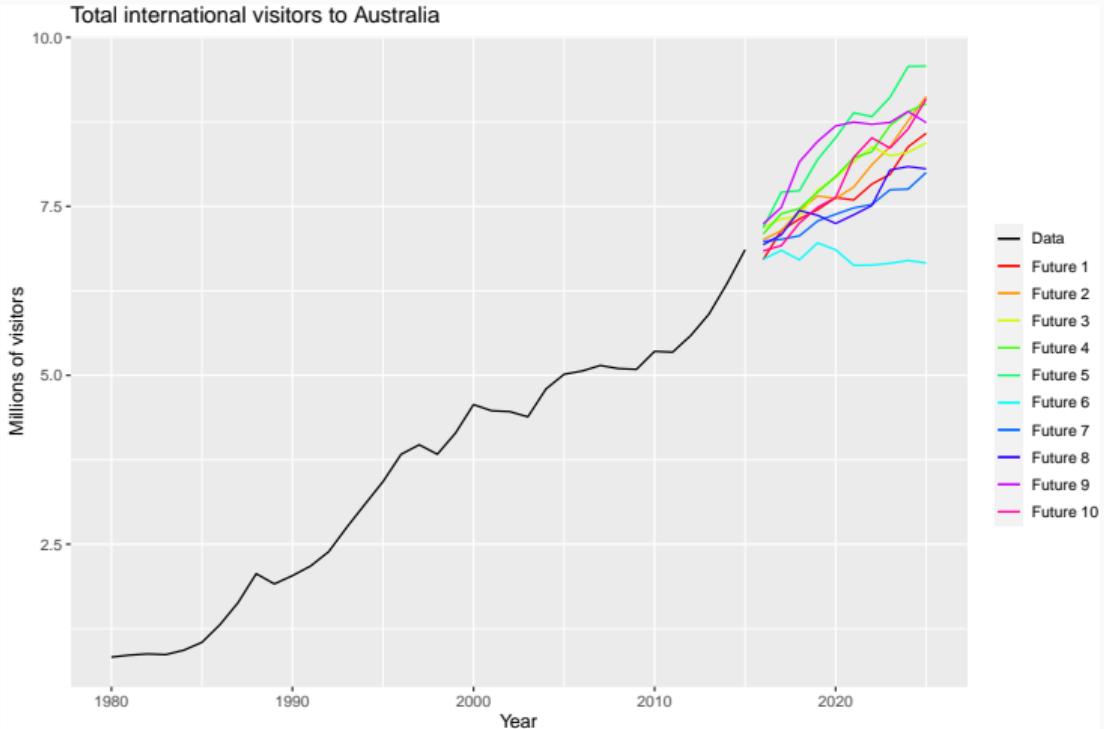
5 Residual diagnostics

6 Lab Session 12

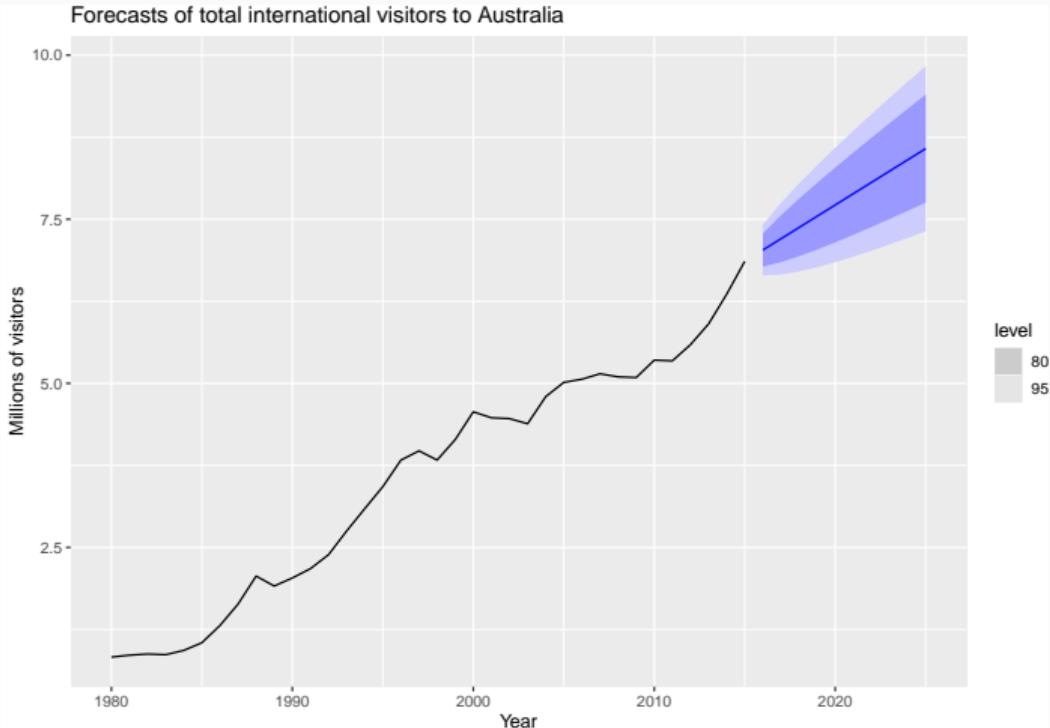
7 Forecast accuracy measures

8 Lab Session 13

Sample futures



Forecast intervals



Statistical forecasting

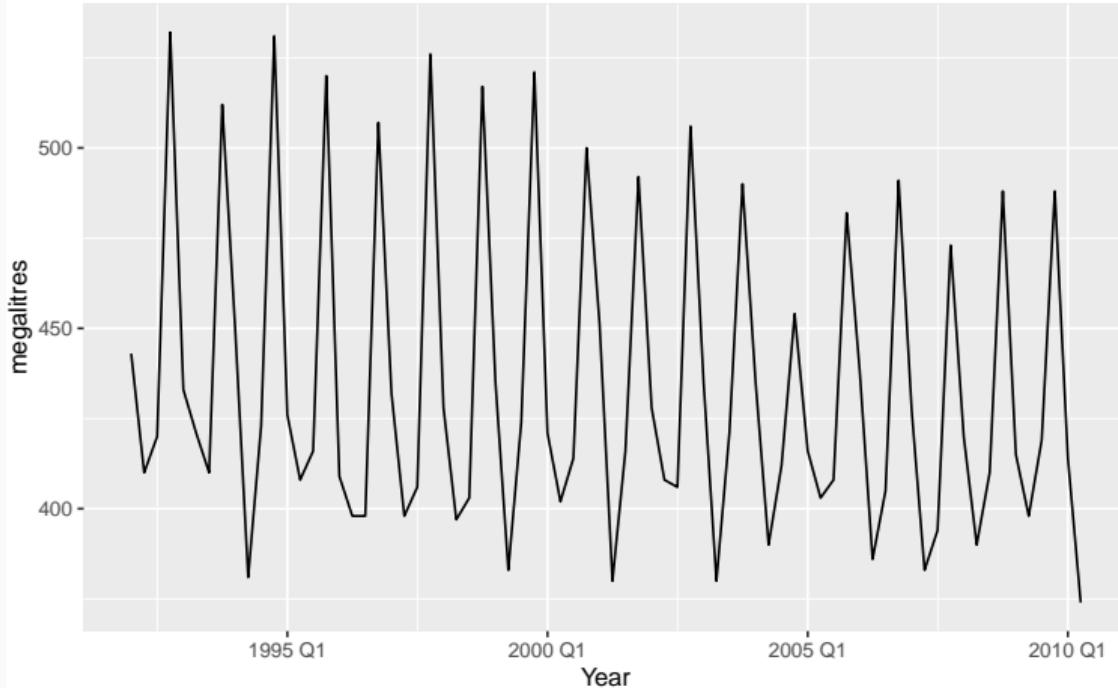
- Thing to be forecast: y_{T+h} .
- What we know: y_1, \dots, y_T .
- Forecast distribution: $y_{T+h|t} = y_{T+h} \mid \{y_1, y_2, \dots, y_T\}$.
- Point forecast: $\hat{y}_{T+h|T} = E[y_{T+h} \mid y_1, \dots, y_T]$.
- Forecast variance: $\text{Var}[y_t \mid y_1, \dots, y_T]$
- Prediction interval is a range of values of y_{T+h} with high probability.

Outline

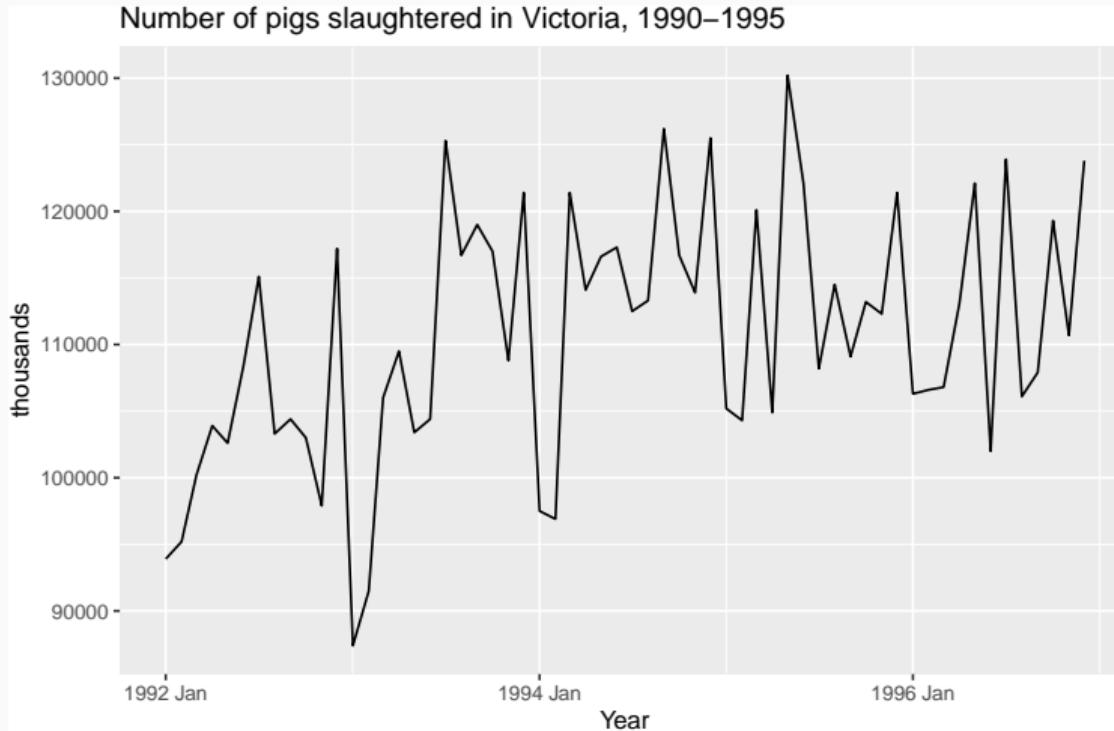
- 1 What can we forecast?
- 2 The statistical forecasting perspective
- 3 Benchmark methods
- 4 Lab Session 11
- 5 Residual diagnostics
- 6 Lab Session 12
- 7 Forecast accuracy measures
- 8 Lab Session 13

Some simple forecasting methods

Australian quarterly beer production



Some simple forecasting methods



Some simple forecasting methods

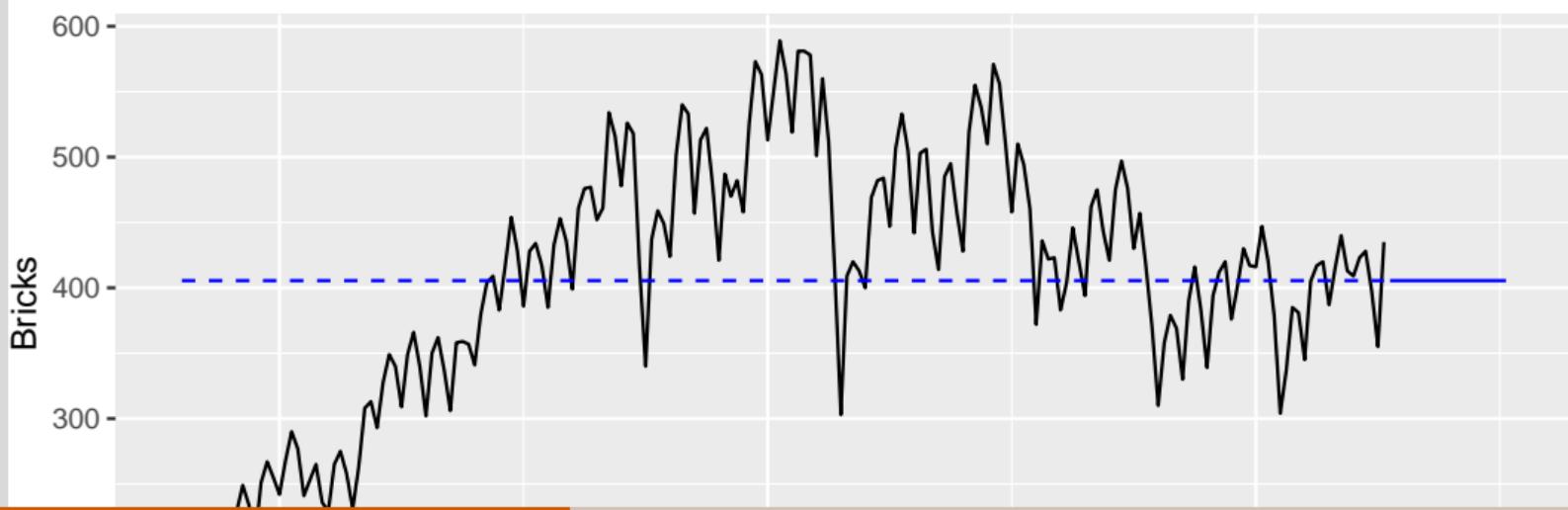


Some simple forecasting methods

MEAN(y): Average method

- Forecast of all future values is equal to mean of historical data $\{y_1, \dots, y_T\}$.
- Forecasts: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \dots + y_T)/T$

Clay brick production in Australia

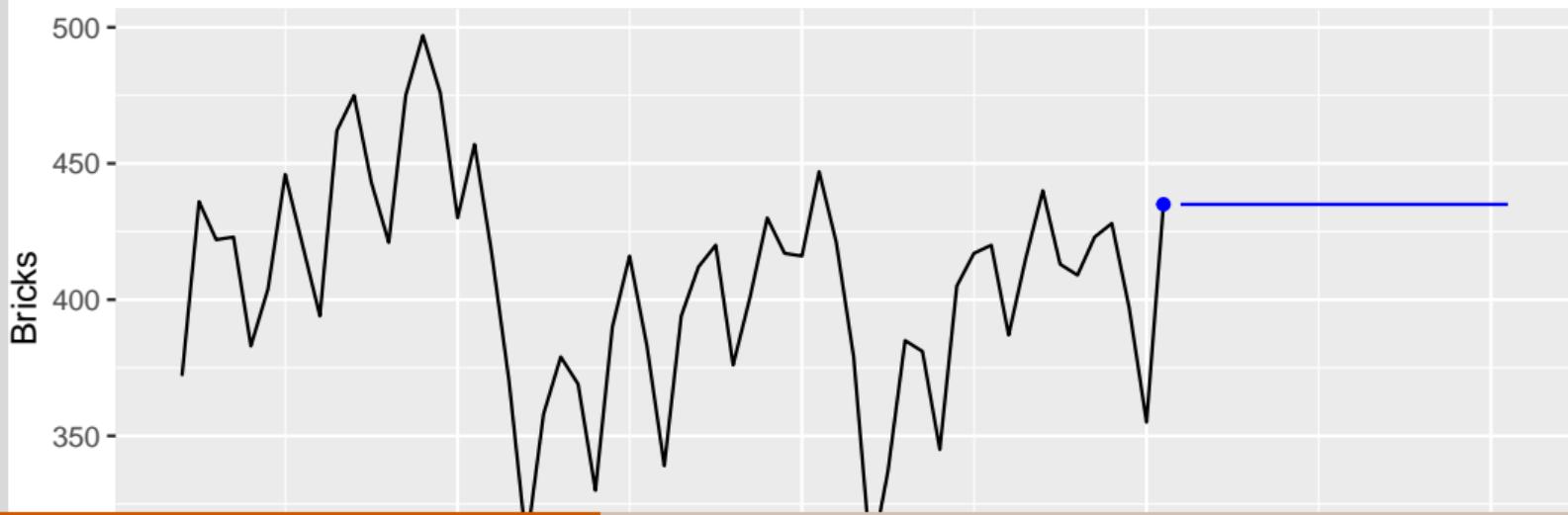


Some simple forecasting methods

NAIVE(y): Naïve method

- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.

Clay brick production in Australia

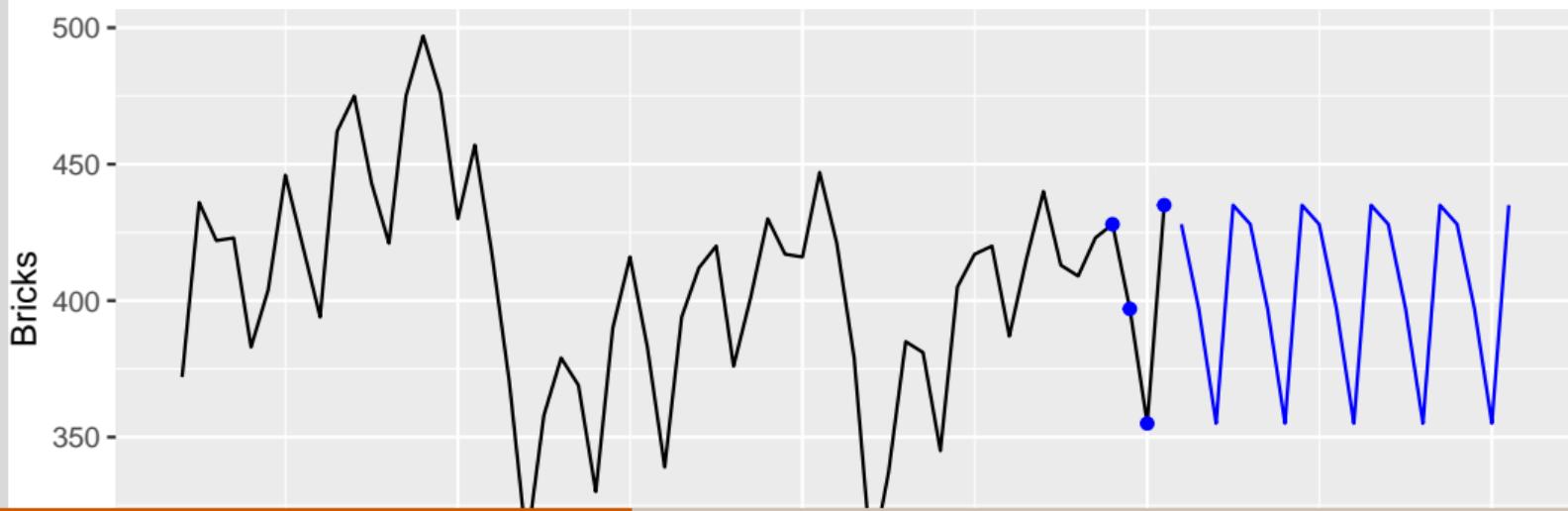


Some simple forecasting methods

SNAIVE($y \sim \text{lag}(m)$): Seasonal naïve method

- Forecasts equal to last value from same season.
- Forecasts: $\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$, where m = seasonal period and k is the integer part of $(h - 1)/m$.

Clay brick production in Australia



Some simple forecasting methods

RW(y ~ drift()): Drift method

- Forecasts equal to last value plus average change.
- Forecasts:

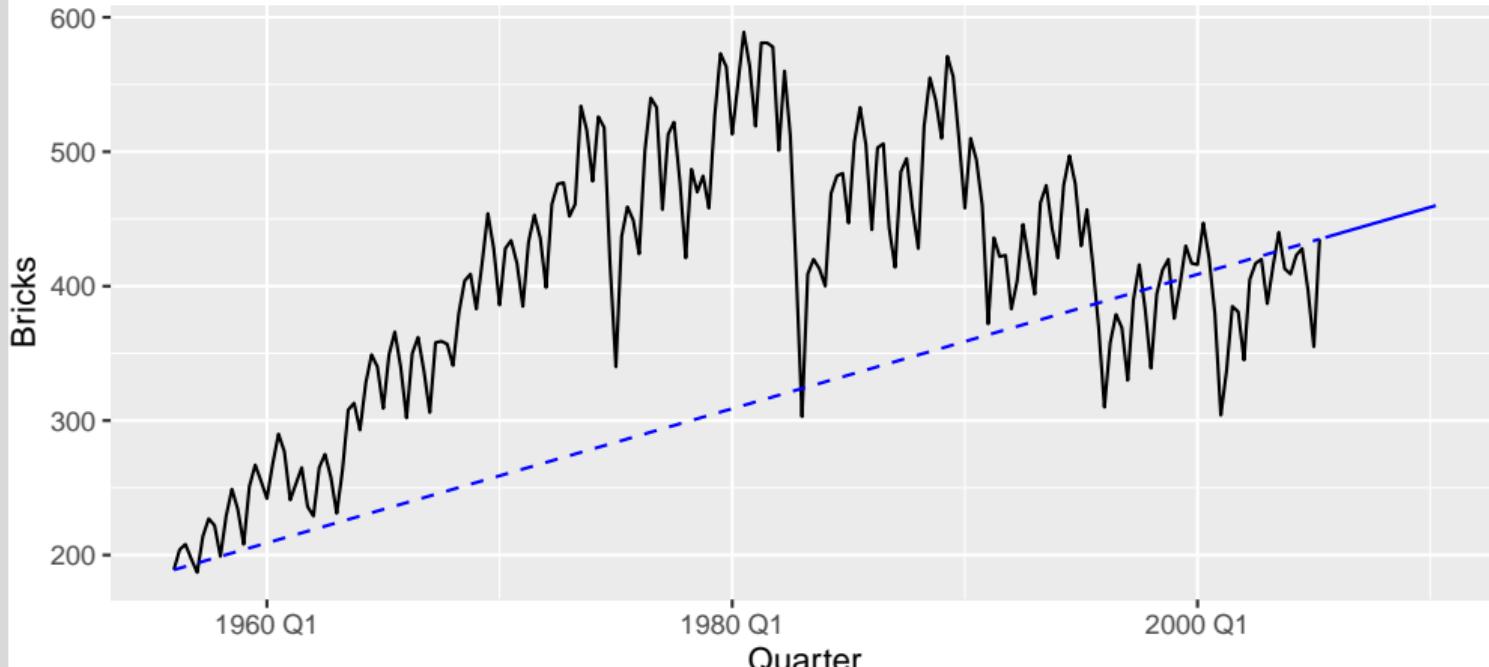
$$\begin{aligned}\hat{y}_{T+h|T} &= y_T + \frac{h}{T-1} \sum_{t=2}^T (y_t - y_{t-1}) \\ &= y_T + \frac{h}{T-1} (y_T - y_1).\end{aligned}$$

- Equivalent to extrapolating a line drawn between first and last observations.

Some simple forecasting methods

Drift method

Clay brick production in Australia



Model fitting

The `model()` function trains models to data.

```
brick_fit <- aus_production %>%
  filter(!is.na(Bricks)) %>%
  model(
    `Seasonal_naïve` = SNAIVE(Bricks),
    `Naïve` = NAIVE(Bricks),
    Drift = RW(Bricks ~ drift()),
    Mean = MEAN(Bricks)
  )
```

```
## # A mable: 1 x 4
##   Seasonal_naïve     Naïve        Drift      Mean
##           <model> <model>       <model> <model>
## 1       <SNAIVE> <NAIVE> <RW w/ drift> <MEAN>
```

A mable is a model table, each cell corresponds to a fitted model.

Producing forecasts

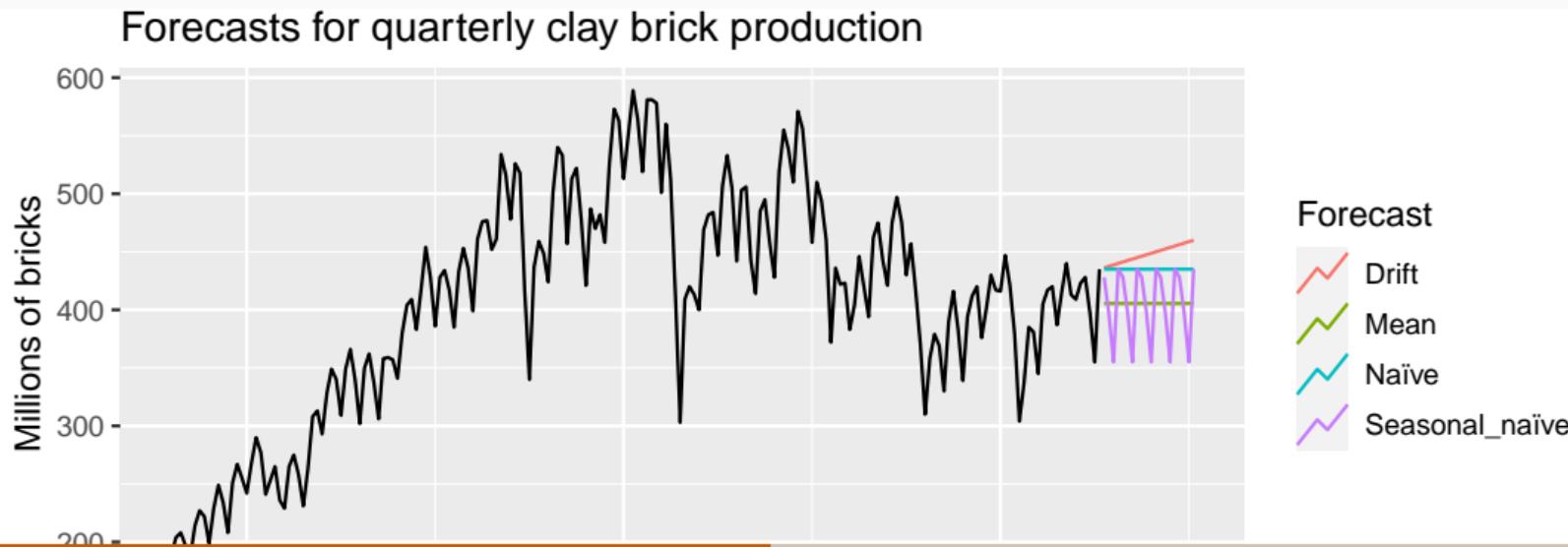
```
brick_fc <- brick_fit %>%
  forecast(h = "5 years")

## # A fable: 80 x 4 [1Q]
## # Key:     .model [4]
##   .model      Quarter      Bricks .mean
##   <chr>       <qtr>       <dist> <dbl>
## 1 Seasonal_naïve 2005 Q3 N(428, 2336) 428
## 2 Seasonal_naïve 2005 Q4 N(397, 2336) 397
## 3 Seasonal_naïve 2006 Q1 N(355, 2336) 355
## 4 Seasonal_naïve 2006 Q2 N(435, 2336) 435
## # ... with 76 more rows
```

A fable is a forecast table with point forecasts and distributions.

Visualising forecasts

```
brick_fc %>%
  autoplot(aus_production, level = NULL) +
  ggtitle("Forecasts for quarterly clay brick production") +
  xlab("Year") + ylab("Millions of bricks") +
  guides(colour = guide_legend(title = "Forecast"))
```



Prediction intervals

```
brick_fc %>% hilo(level=c(50,75))
```

```
## # A tsibble: 80 x 6 [1Q]
## # Key:      .model [4]
##   .model     Quarter     Bricks .mean    `50%`    `75%
##   <chr>      <qtr>     <dist> <dbl>    <dbl>    <dbl>
## 1 Seasonal_n~ 2005 Q3 N(428, 2336)    428 [395, 461]50 [372, 484]75
## 2 Seasonal_n~ 2005 Q4 N(397, 2336)    397 [364, 430]50 [341, 453]75
## 3 Seasonal_n~ 2006 Q1 N(355, 2336)    355 [322, 388]50 [299, 411]75
## 4 Seasonal_n~ 2006 Q2 N(435, 2336)    435 [402, 468]50 [379, 491]75
## 5 Seasonal_n~ 2006 Q3 N(428, 4672)    428 [382, 474]50 [349, 507]75
## 6 Seasonal_n~ 2006 Q4 N(397, 4672)    397 [351, 443]50 [318, 476]75
## 7 Seasonal_n~ 2007 Q1 N(355, 4672)    355 [309, 401]50 [276, 434]75
## 8 Seasonal_n~ 2007 Q2 N(435, 4672)    435 [389, 481]50 [356, 514]75
## 9 Seasonal_n~ 2007 Q3 N(428, 7008)    428 [372, 484]50 [332, 524]75
```

Prediction intervals

```
brick_fc %>% hilo(level=c(50,75)) %>% unpack_hilo(c("50%", "75%"))
```

```
## # A tsibble: 80 x 8 [1Q]
## # Key:      .model [4]
##   .model      Quarter     Bricks .mean 50%_l~1 50%_u~2 75%_l~3
##   <chr>       <qtr>      <dist> <dbl>  <dbl>    <dbl>    <dbl>
## 1 Seasonal_naï~ 2005 Q3 N(428, 2336)  428    395.    461.    372.
## 2 Seasonal_naï~ 2005 Q4 N(397, 2336)  397    364.    430.    341.
## 3 Seasonal_naï~ 2006 Q1 N(355, 2336)  355    322.    388.    299.
## 4 Seasonal_naï~ 2006 Q2 N(435, 2336)  435    402.    468.    379.
## 5 Seasonal_naï~ 2006 Q3 N(428, 4672)  428    382.    474.    349.
## 6 Seasonal_naï~ 2006 Q4 N(397, 4672)  397    351.    443.    318.
## 7 Seasonal_naï~ 2007 Q1 N(355, 4672)  355    309.    401.    276.
## 8 Seasonal_naï~ 2007 Q2 N(435, 4672)  435    389.    481.    356.
## 9 Seasonal_naï~ 2007 Q3 N(428, 7008)  428    372.    484.    332.
```

Outline

- 1 What can we forecast?
- 2 The statistical forecasting perspective
- 3 Benchmark methods
- 4 Lab Session 11
- 5 Residual diagnostics
- 6 Lab Session 12
- 7 Forecast accuracy measures
- 8 Lab Session 13

Lab Session 11

- Produce forecasts using an appropriate benchmark method for household wealth (`hh_budget`). Plot the results using `autoplot()`.
- Produce forecasts using an appropriate benchmark method for Australian takeaway food turnover (`aus_retail`). Plot the results using `autoplot()`.

Outline

- 1 What can we forecast?
- 2 The statistical forecasting perspective
- 3 Benchmark methods
- 4 Lab Session 11
- 5 Residual diagnostics
- 6 Lab Session 12
- 7 Forecast accuracy measures
- 8 Lab Session 13

Fitted values

- $\hat{y}_{t|t-1}$ is the forecast of y_t based on observations y_1, \dots, y_{t-1} .
- We call these “fitted values”.
- Sometimes drop the subscript: $\hat{y}_t \equiv \hat{y}_{t|t-1}$.
- Often not true forecasts since parameters are estimated on all data.

For example:

- $\hat{y}_t = \bar{y}$ for average method.
- $\hat{y}_t = y_{t-1} + (y_T - y_1)/(T - 1)$ for drift method.

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

Assumptions

- 1 $\{e_t\}$ uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
- 2 $\{e_t\}$ have mean zero. If they don't, then forecasts are biased.

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

Assumptions

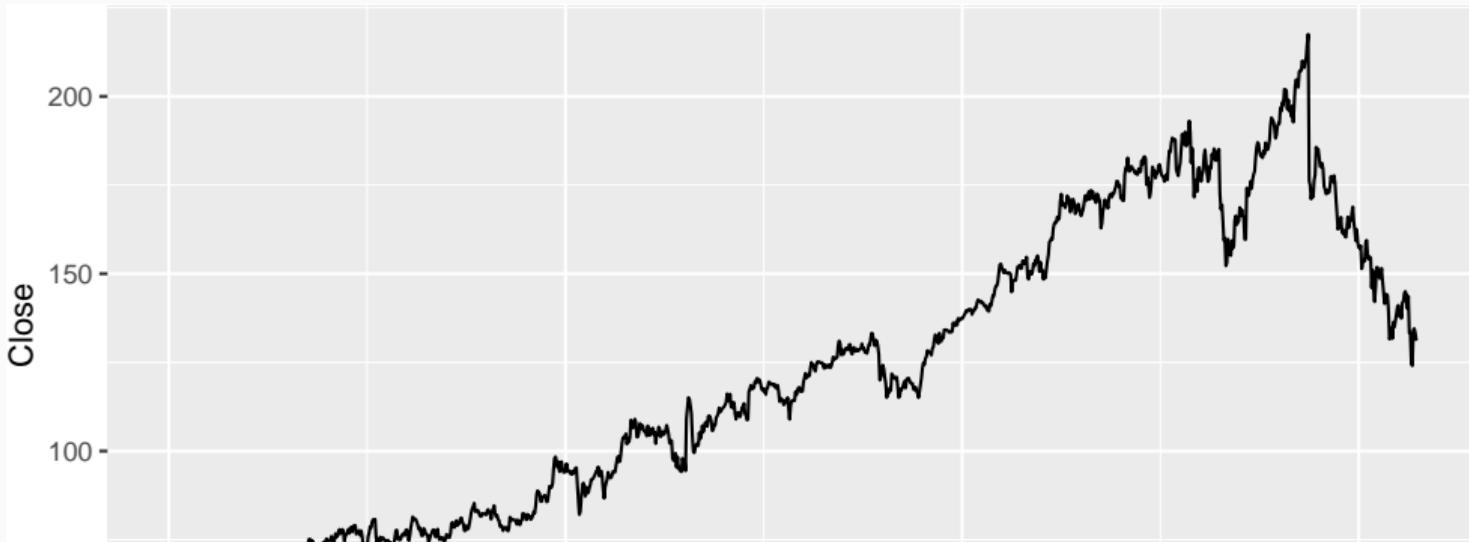
- 1 $\{e_t\}$ uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
- 2 $\{e_t\}$ have mean zero. If they don't, then forecasts are biased.

Useful properties (for prediction intervals)

- 3 $\{e_t\}$ have constant variance.
- 4 $\{e_t\}$ are normally distributed.

Facebook closing stock price

```
fb_stock <- gafa_stock %>%
  filter(Symbol == "FB") %>%
  mutate(trading_day = row_number()) %>%
  update_tsibble(index = trading_day, regular = TRUE)
fb_stock %>% autoplot(Close)
```



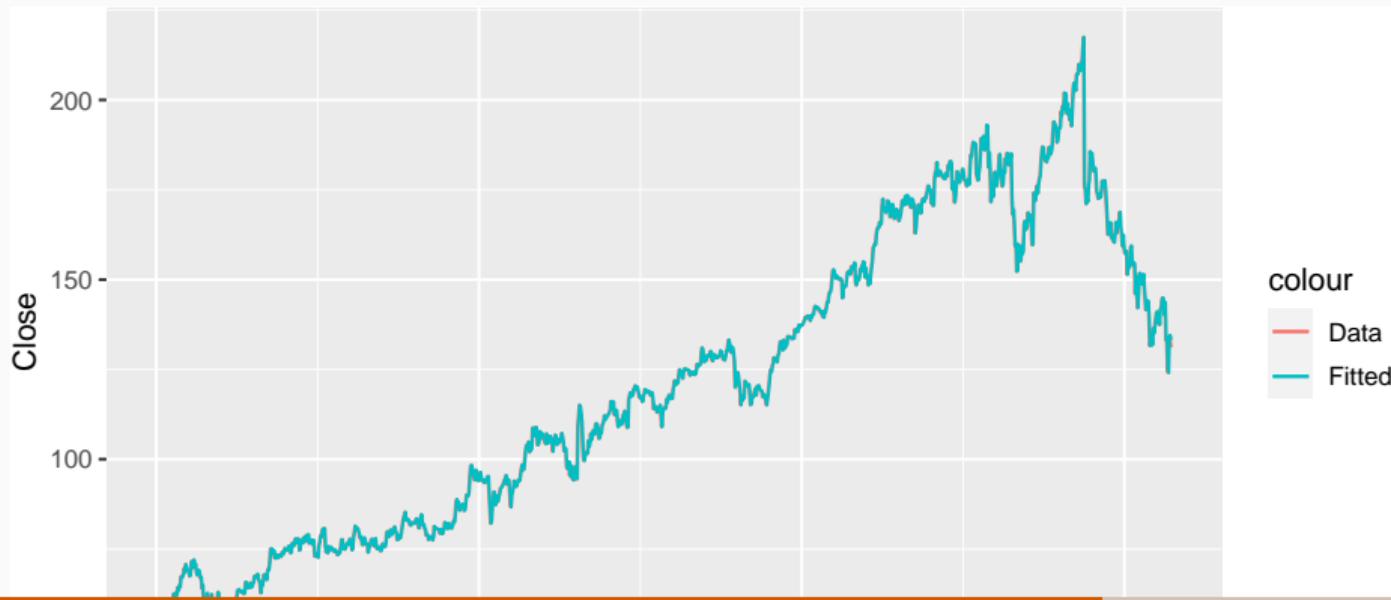
Facebook closing stock price

```
fit <- fb_stock %>% model(NAIVE(Close))  
augment(fit)
```

```
## # A tsibble: 1,258 x 7 [1]  
## # Key:      Symbol, .model [1]  
##   Symbol .model      trading_day Close .fitted .resid .innov  
##   <chr>  <chr>          <int>  <dbl>   <dbl>   <dbl>   <dbl>  
## 1 FB    NAIVE(Close)       1  54.7     NA     NA     NA  
## 2 FB    NAIVE(Close)       2  54.6   54.7 -0.150 -0.150  
## 3 FB    NAIVE(Close)       3  57.2   54.6   2.64   2.64  
## 4 FB    NAIVE(Close)       4  57.9   57.2   0.720  0.720  
## 5 FB    NAIVE(Close)       5  58.2   57.9   0.310  0.310  
## 6 FB    NAIVE(Close)       6  57.2   58.2  -1.01  -1.01  
## 7 FB    NAIVE(Close)       7  57.9   57.2   0.720  0.720  
## 8 FB    NAIVE(Close)       8  55.9   57.9  -2.03  -2.03  
## 9 FB    NAIVE(Close)       9  57.7   55.9   1.83   1.83  
## 10 FB   NAIVE(Close)      10  57.6   57.7  -0.140 -0.140  
## # ... with 1,248 more rows
```

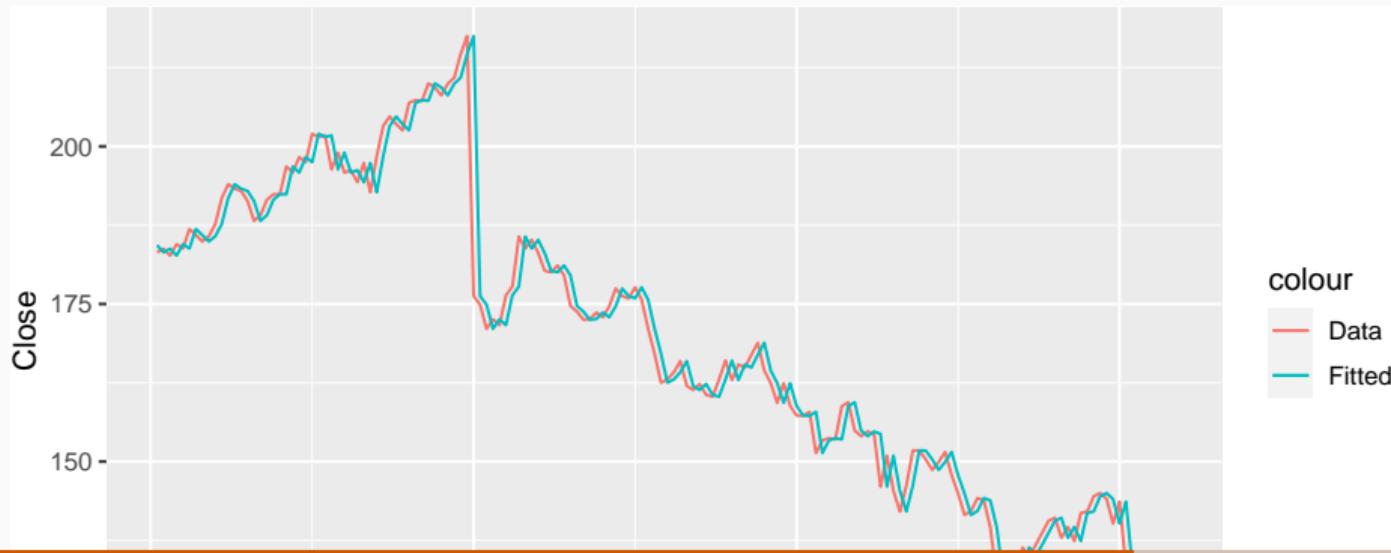
Facebook closing stock price

```
augment(fit) %>%
  ggplot(aes(x = trading_day)) +
  geom_line(aes(y = Close, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted"))
```



Facebook closing stock price

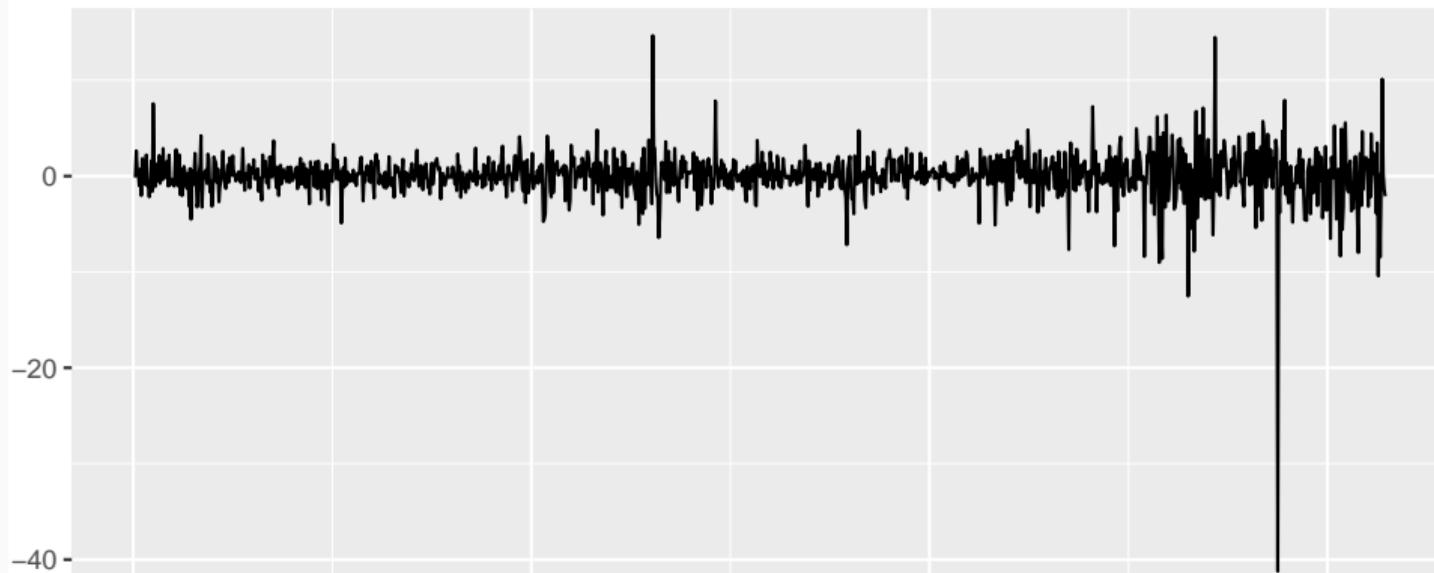
```
augment(fit) %>%
  filter(trading_day > 1100) %>%
  ggplot(aes(x = trading_day)) +
  geom_line(aes(y = Close, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted"))
```



Facebook closing stock price

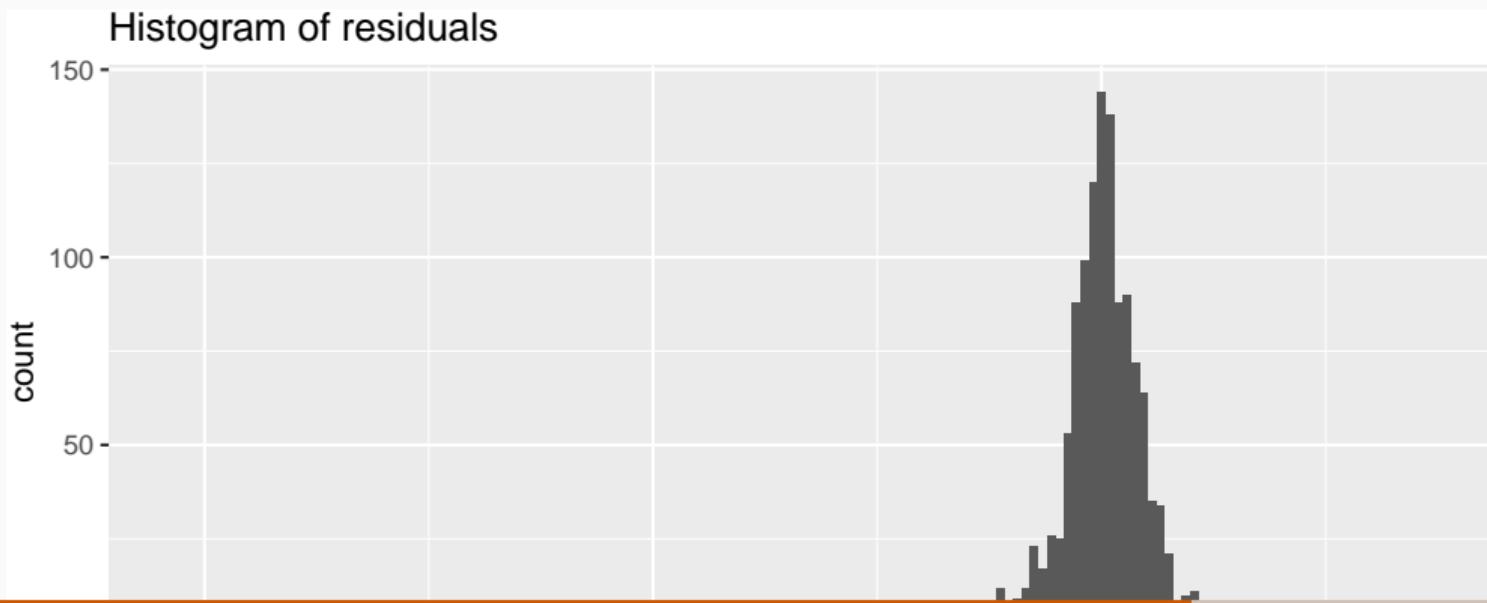
```
augment(fit) %>%
  autoplot(.resid) + xlab("Day") + ylab("") +
  ggtitle("Residuals from naïve method")
```

Residuals from naïve method



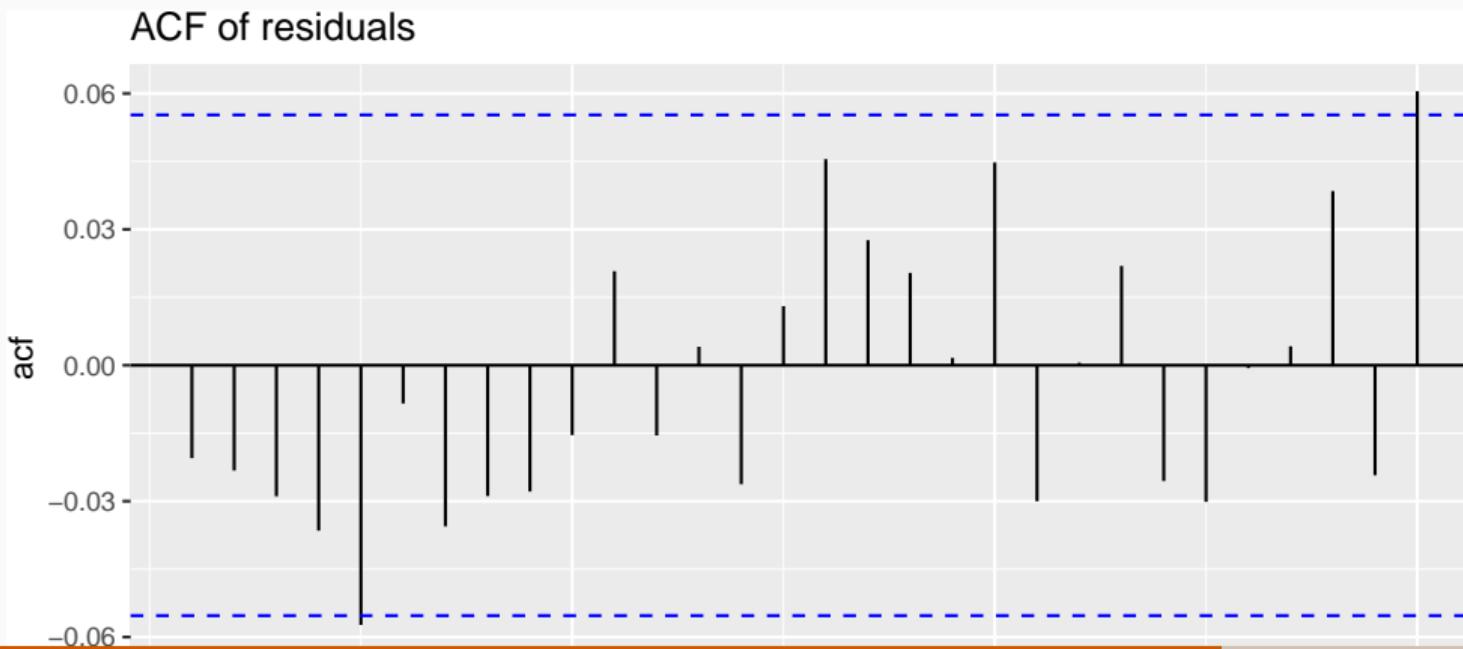
Facebook closing stock price

```
augment(fit) %>%
  ggplot(aes(x = .resid)) +
  geom_histogram(bins = 150) +
  ggtitle("Histogram of residuals")
```



Facebook closing stock price

```
augment(fit) %>%  
  ACF(.resid) %>%  
  autoplot() + ggtitle("ACF of residuals")
```

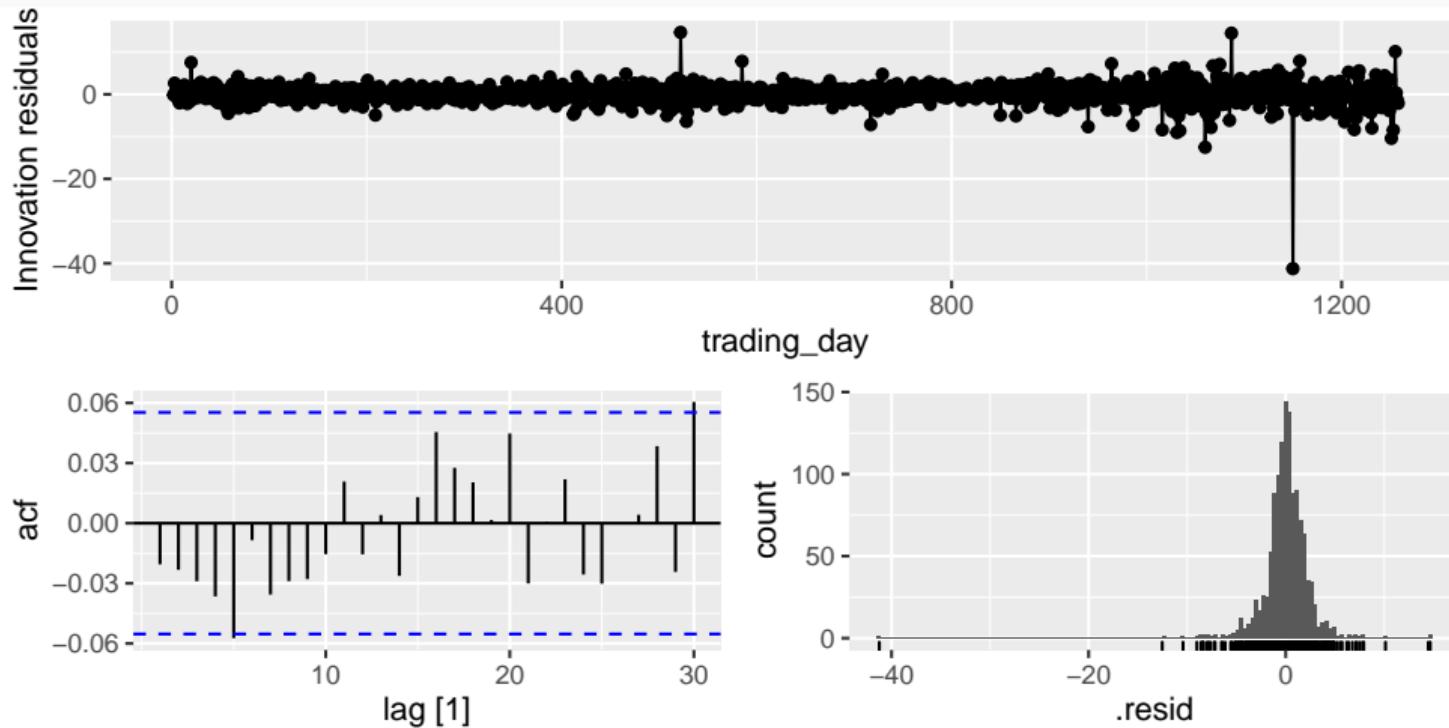


ACF of residuals

- We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren't, then there is information left in the residuals that should be used in computing forecasts.
- So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.
- We *expect* these to look like white noise.

Combined diagnostic graph

```
fit %>% gg_tsresiduals()
```



Ljung-Box test

Test whether *whole set* of r_k values is significantly different from zero set.

$$Q = T(T + 2) \sum_{k=1}^h (T - k)^{-1} r_k^2$$

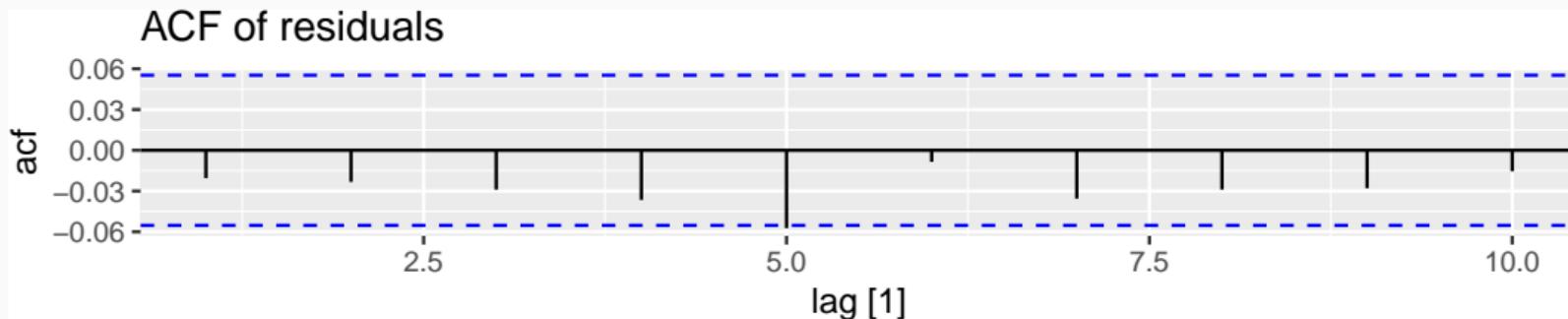
where h = max lag and T = # observations.

- If each r_k close to zero, Q will be **small**.
- If some r_k values large (+ or -), Q will be **large**.
- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- If data are WN, $Q \sim \chi^2$ with $(h - K)$ degrees of freedom where K = no. parameters in model.
- When applied to raw data, set $K = 0$.

Ljung-Box test

$$Q = T(T + 2) \sum_{k=1}^h (T - k)^{-1} r_k^2$$

where h = max lag and T = # observations.



```
# lag=h and dof=K
augment(fit) %>%
  features(.resid, ljung_box, dof = 0, lag = 10)
```

```
## # A tibble: 1 x 4
##   Symbol    model    lb     stat    pvalue
##   <fct>    <fct>    <dbl>    <dbl>    <dbl>
```

Outline

- 1 What can we forecast?
- 2 The statistical forecasting perspective
- 3 Benchmark methods
- 4 Lab Session 11
- 5 Residual diagnostics
- 6 Lab Session 12
- 7 Forecast accuracy measures
- 8 Lab Session 13

Lab Session 12

- Compute seasonal naïve forecasts for quarterly Australian beer production.
- Test if the residuals are white noise. What do you conclude?

Outline

1 What can we forecast?

2 The statistical forecasting perspective

3 Benchmark methods

4 Lab Session 11

5 Residual diagnostics

6 Lab Session 12

7 Forecast accuracy measures

8 Lab Session 13

Training and test sets



- A model which fits the training data well will not necessarily forecast well.
- Forecast accuracy is based only on the test set.

Forecast errors

Forecast “error”: the difference between an observed value and its forecast.

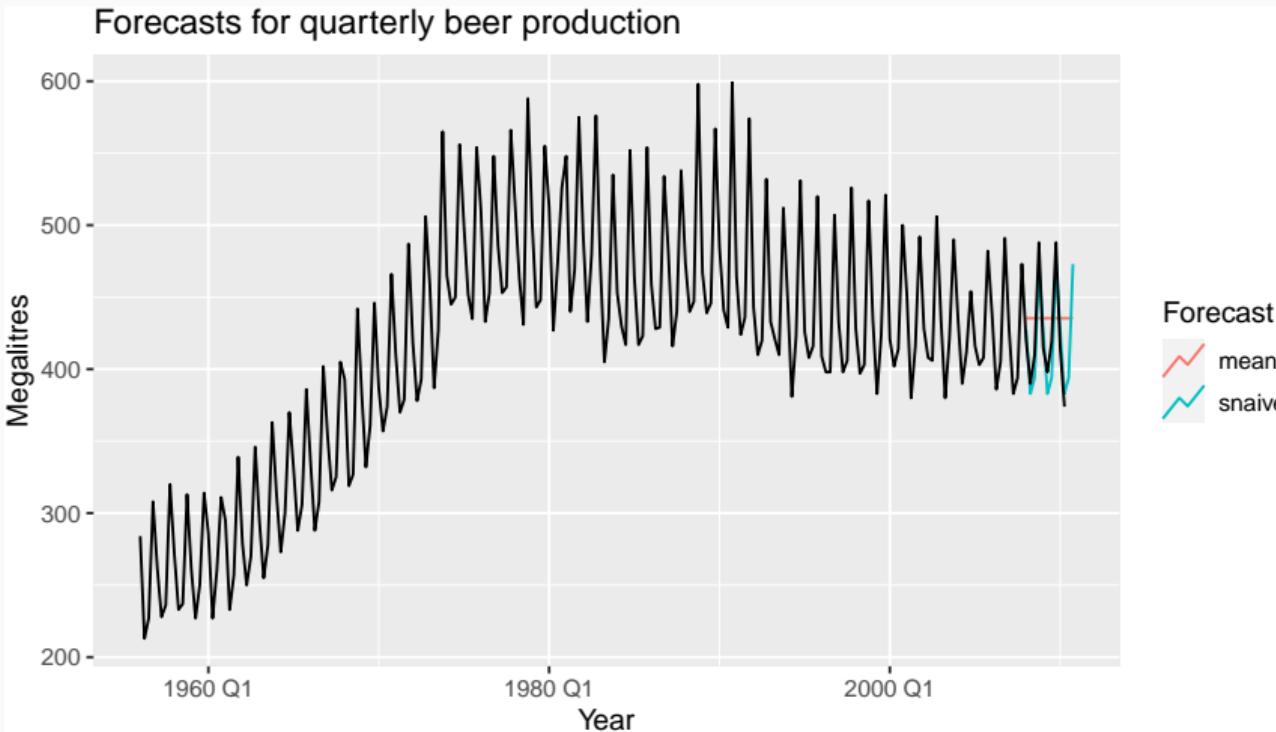
$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

where the training data is given by $\{y_1, \dots, y_T\}$

Measures of forecast accuracy

```
beer_fit <- aus_production %>%
  filter(between(year(Quarter), 1992, 2007)) %>%
  model(
    snaive = SNAIVE(Beer),
    mean = MEAN(Beer)
  )
beer_fit %>%
  forecast(h = "3 years") %>%
  autoplot(aus_production, level = NULL) +
  ggtitle("Forecasts for quarterly beer production") +
  xlab("Year") + ylab("Megalitres") +
  guides(colour = guide_legend(title = "Forecast"))
```

Measures of forecast accuracy



Measures of forecast accuracy

y_{T+h} = $(T + h)$ th observation, $h = 1, \dots, H$

$\hat{y}_{T+h|T}$ = its forecast based on data up to time T .

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

MAE = $\text{mean}(|e_{T+h}|)$

MSE = $\text{mean}(e_{T+h}^2)$

RMSE = $\sqrt{\text{mean}(e_{T+h}^2)}$

MAPE = $100\text{mean}(|e_{T+h}|/|y_{T+h}|)$

Measures of forecast accuracy

y_{T+h} = $(T + h)$ th observation, $h = 1, \dots, H$

$\hat{y}_{T+h|T}$ = its forecast based on data up to time T .

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

MAE = mean($|e_{T+h}|$)

MSE = mean(e_{T+h}^2)

RMSE = $\sqrt{\text{mean}(e_{T+h}^2)}$

MAPE = $100\text{mean}(|e_{T+h}| / |y_{T+h}|)$

- MAE, MSE, RMSE are all scale dependent.
- MAPE is scale independent but is only sensible if $y_t \gg 0$ for all t , and y has a natural zero.

Measures of forecast accuracy

Mean Absolute Scaled Error

$$\text{MASE} = \text{mean}(|e_{T+h}|)/Q$$

where Q is a stable measure of the scale of the time series $\{y_t\}$.

Proposed by Hyndman and Koehler (IJF, 2006).

For non-seasonal time series,

$$Q = (T - 1)^{-1} \sum_{t=2}^T |y_t - y_{t-1}|$$

works well. Then MASE is equivalent to MAE relative to a naïve method.

Measures of forecast accuracy

Mean Absolute Scaled Error

$$\text{MASE} = \text{mean}(|e_{T+h}|)/Q$$

where Q is a stable measure of the scale of the time series $\{y_t\}$.

Proposed by Hyndman and Koehler (IJF, 2006).

For seasonal time series,

$$Q = (T - m)^{-1} \sum_{t=m+1}^T |y_t - y_{t-m}|$$

works well. Then MASE is equivalent to MAE relative to a seasonal naïve method.

Measures of forecast accuracy

```
beer_fc <- forecast(beer_fit, h = "3 years")
accuracy(beer_fc, aus_production)
```

```
## # A tibble: 2 x 10
##   .model .type     ME   RMSE    MAE    MPE    MAPE    MASE   RMSSE     ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 mean    Test  -13.8  38.4  34.8 -3.97  8.28  2.20  1.96 -0.0691
## 2 snaive  Test    5.2  14.3  13.4  1.15  3.17  0.847 0.729  0.132
```

Outline

- 1 What can we forecast?
- 2 The statistical forecasting perspective
- 3 Benchmark methods
- 4 Lab Session 11
- 5 Residual diagnostics
- 6 Lab Session 12
- 7 Forecast accuracy measures
- 8 Lab Session 13

Lab Session 13

- Create a training set for household wealth (`hh_budget`) by withholding the last four years as a test set.
- Fit all the appropriate benchmark methods to the training set and forecast the periods covered by the test set.
- Compute the accuracy of your forecasts. Which method does best?
- Repeat the exercise using the Australian takeaway food turnover data (`aus_retail`) with a test set of four years.