

Time Series Analysis & Forecasting Using R

6. Introduction to forecasting



Outline

- 1 Statistical forecasting
- 2 Benchmark methods
- 3 Lab Session 11
- 4 Residual diagnostics
- 5 Lab Session 12
- 6 Forecast accuracy measures
- 7 Lab Session 13

Outline

1 Statistical forecasting

2 Benchmark methods

3 Lab Session 11

4 Residual diagnostics

5 Lab Session 12

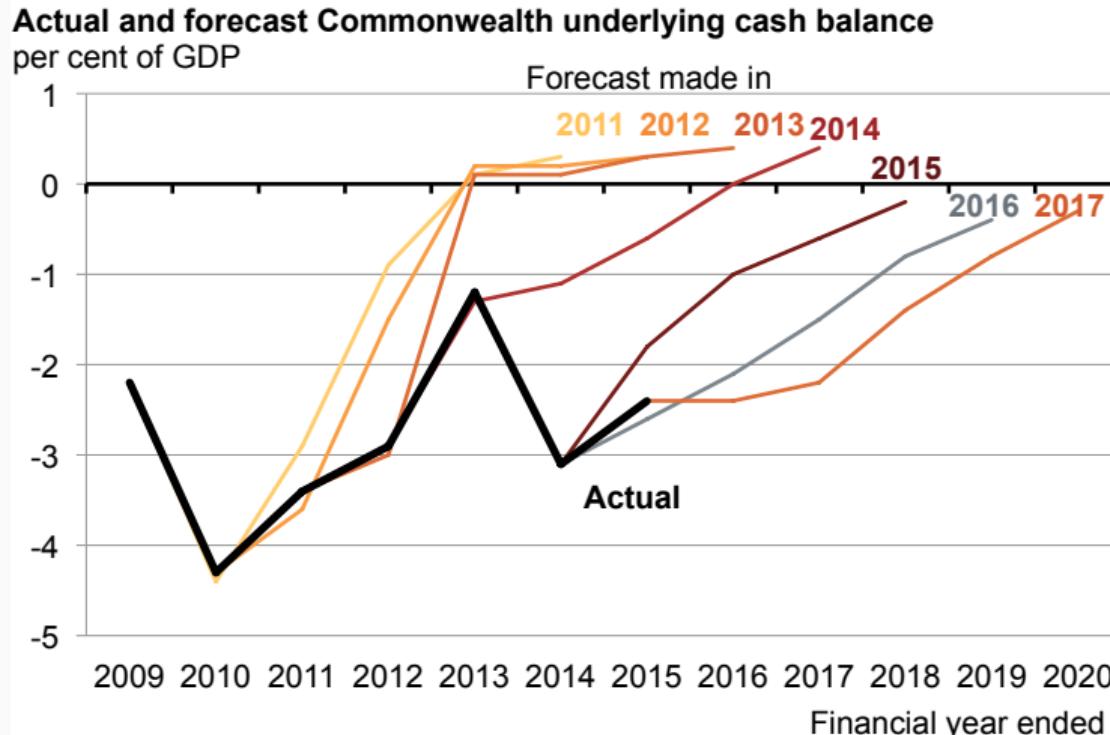
6 Forecast accuracy measures

7 Lab Session 13

Forecasting is difficult

Commonwealth plans to drift back to surplus
show the triumph of experience over hope

GRATTAN
Institute



What can we forecast?



What can we forecast?



What can we forecast?

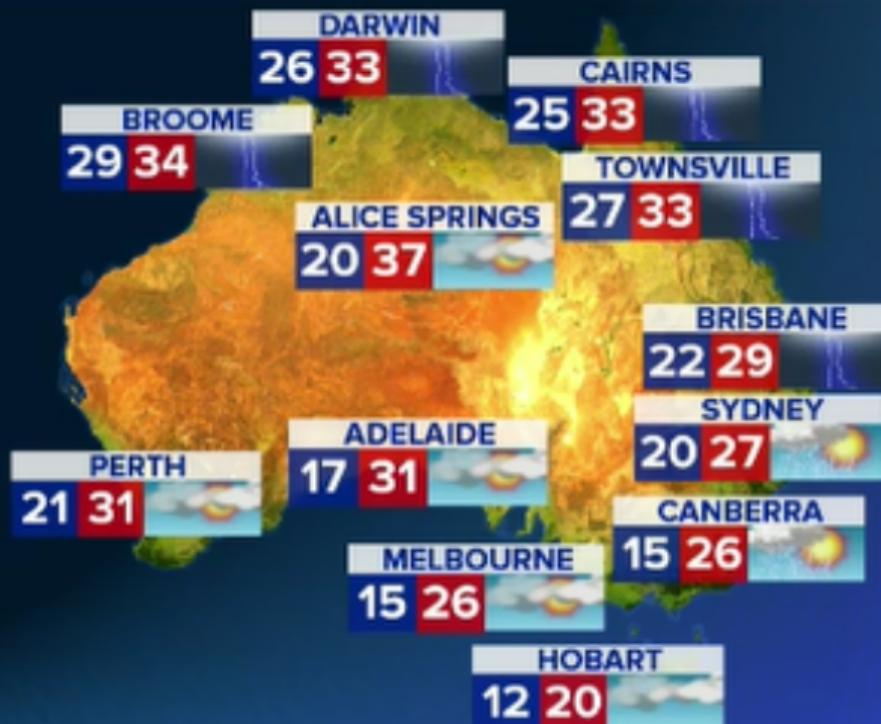


What can we forecast?



What can we forecast?

TOMORROW



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:

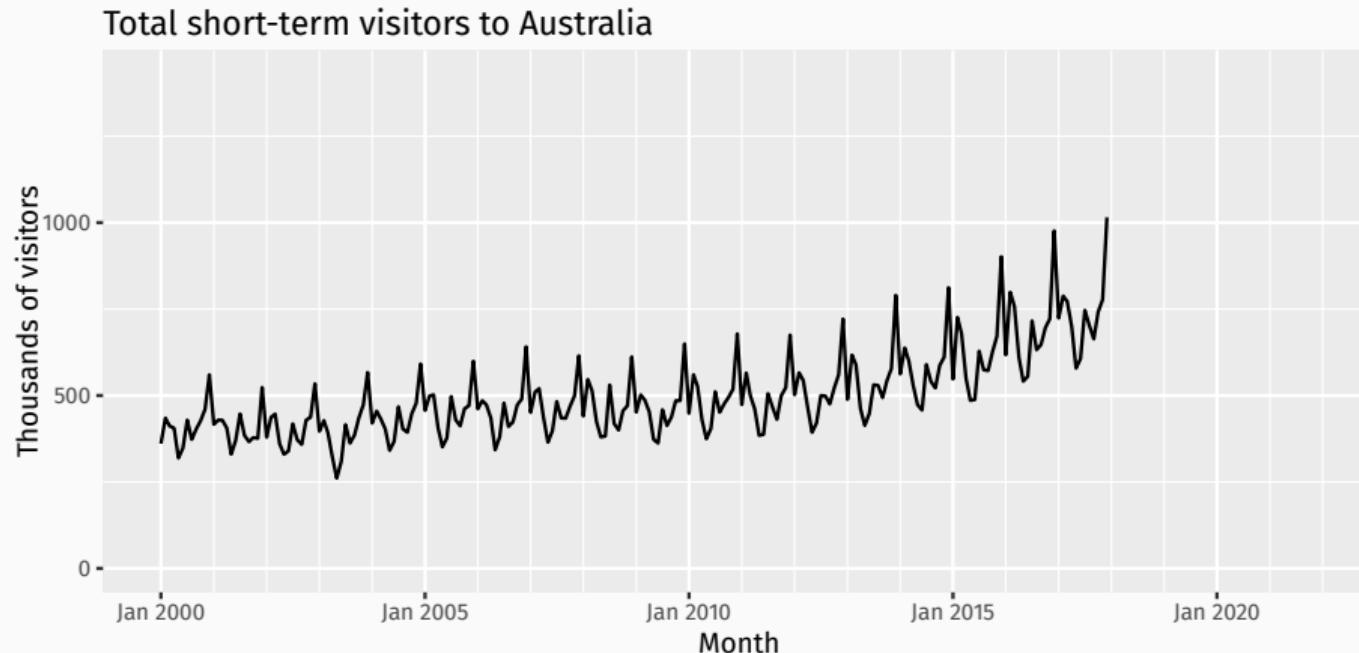
- we have a good understanding of the factors that contribute to it
- there is lots of data available;
- the forecasts cannot affect the thing we are trying to forecast.
- there is relatively low natural/unexplainable random variation.
- the future is somewhat similar to the past

Random futures

A forecast is an estimate of the probabilities of possible futures.

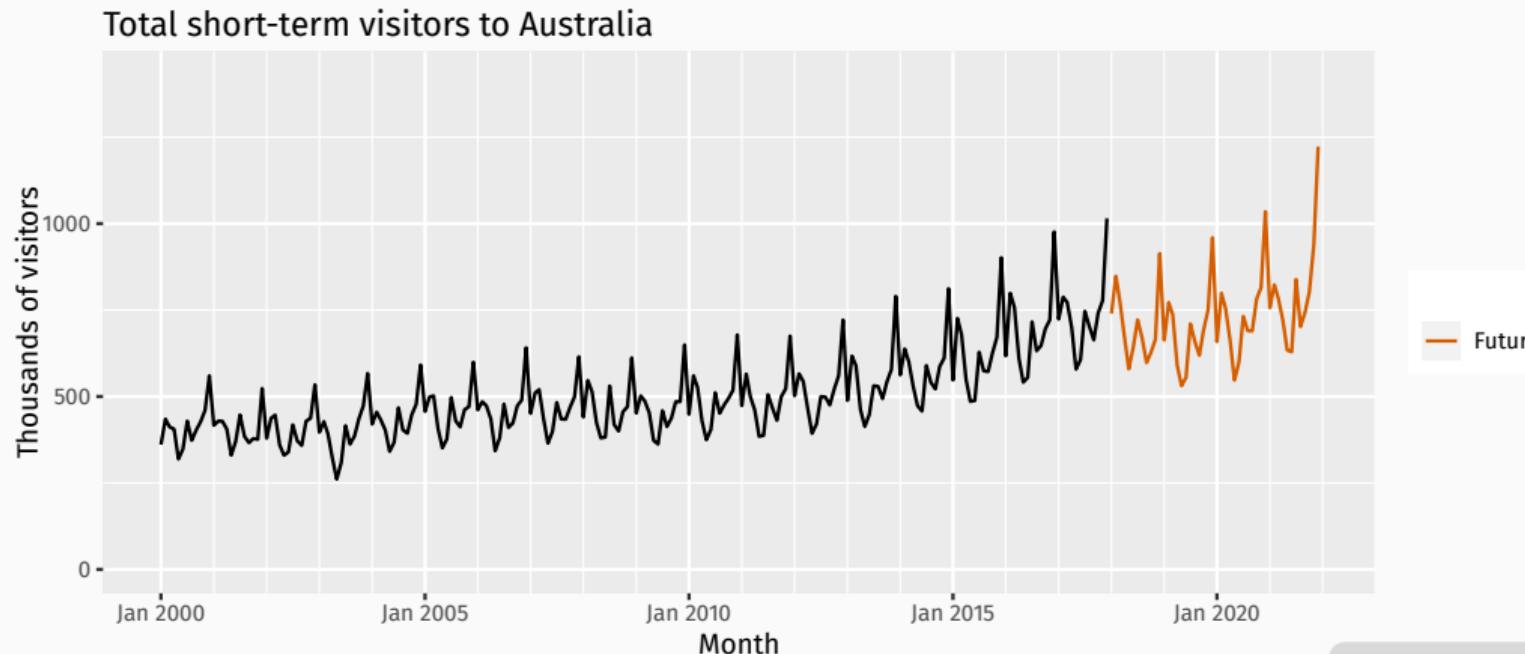
Random futures

A forecast is an estimate of the probabilities of possible futures.



Random futures

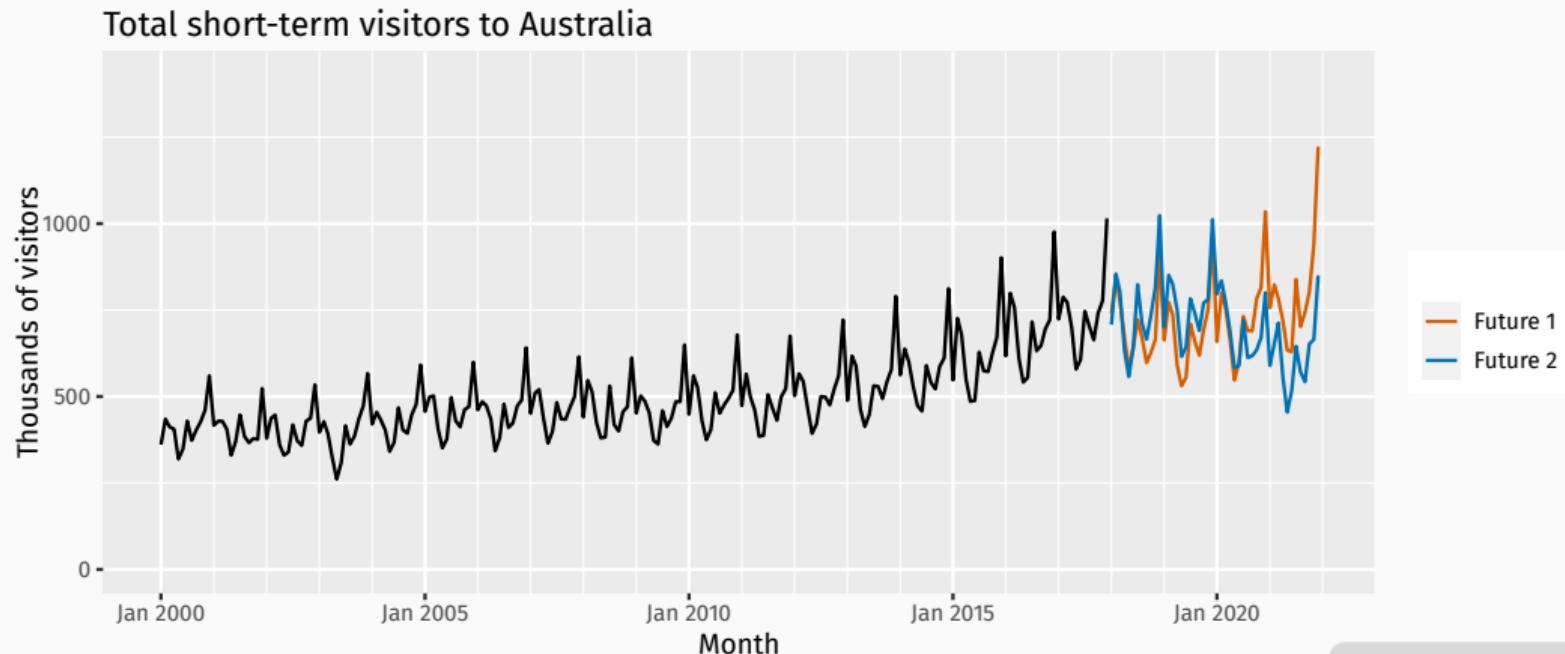
A forecast is an estimate of the probabilities of possible futures.



Simulated futures
from an ETS model

Random futures

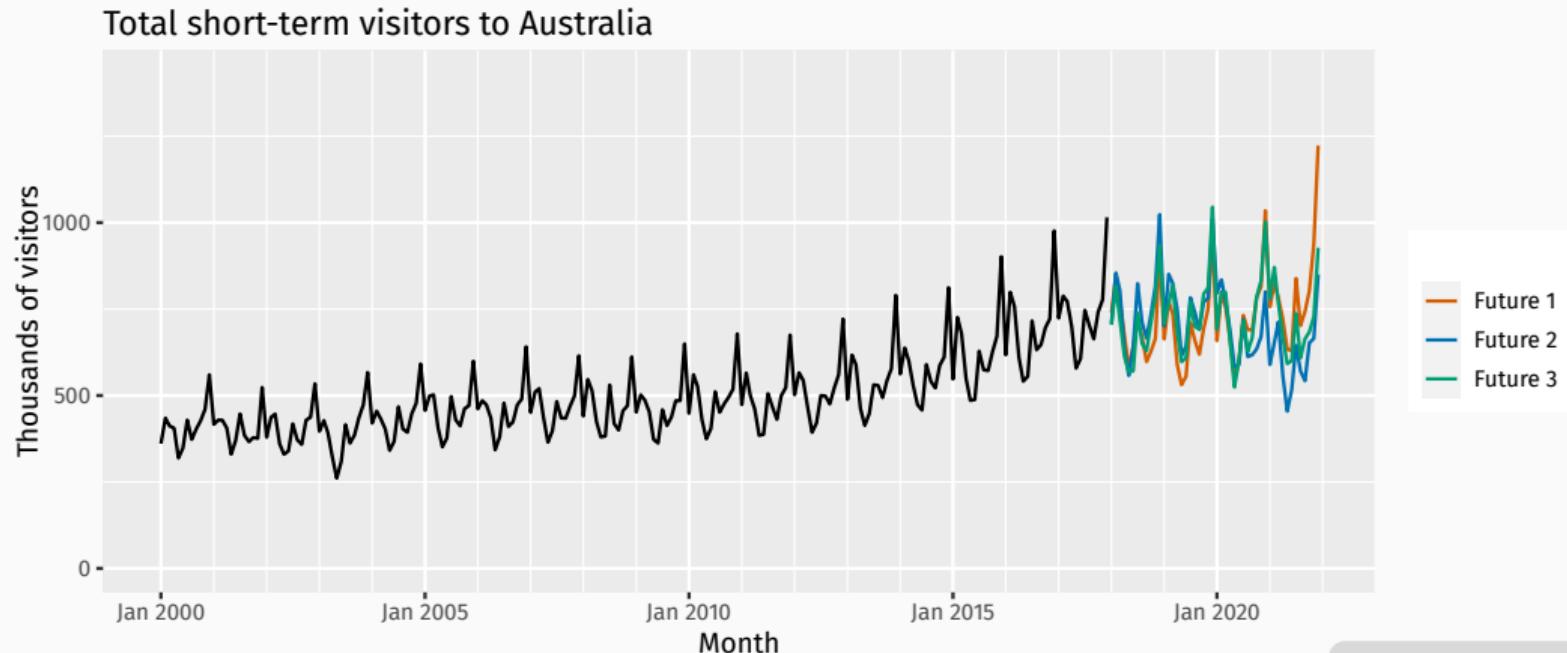
A forecast is an estimate of the probabilities of possible futures.



Simulated futures
from an ETS model

Random futures

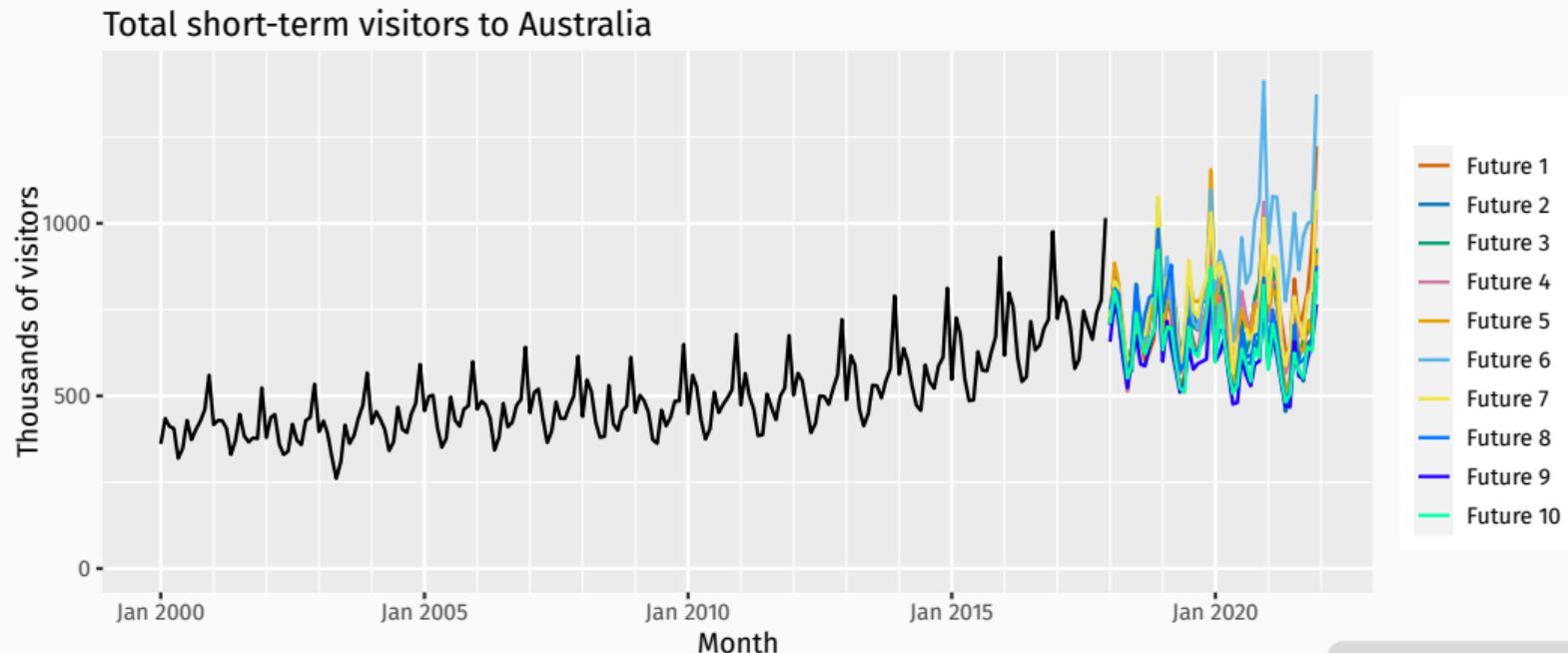
A forecast is an estimate of the probabilities of possible futures.



Simulated futures
from an ETS model

Random futures

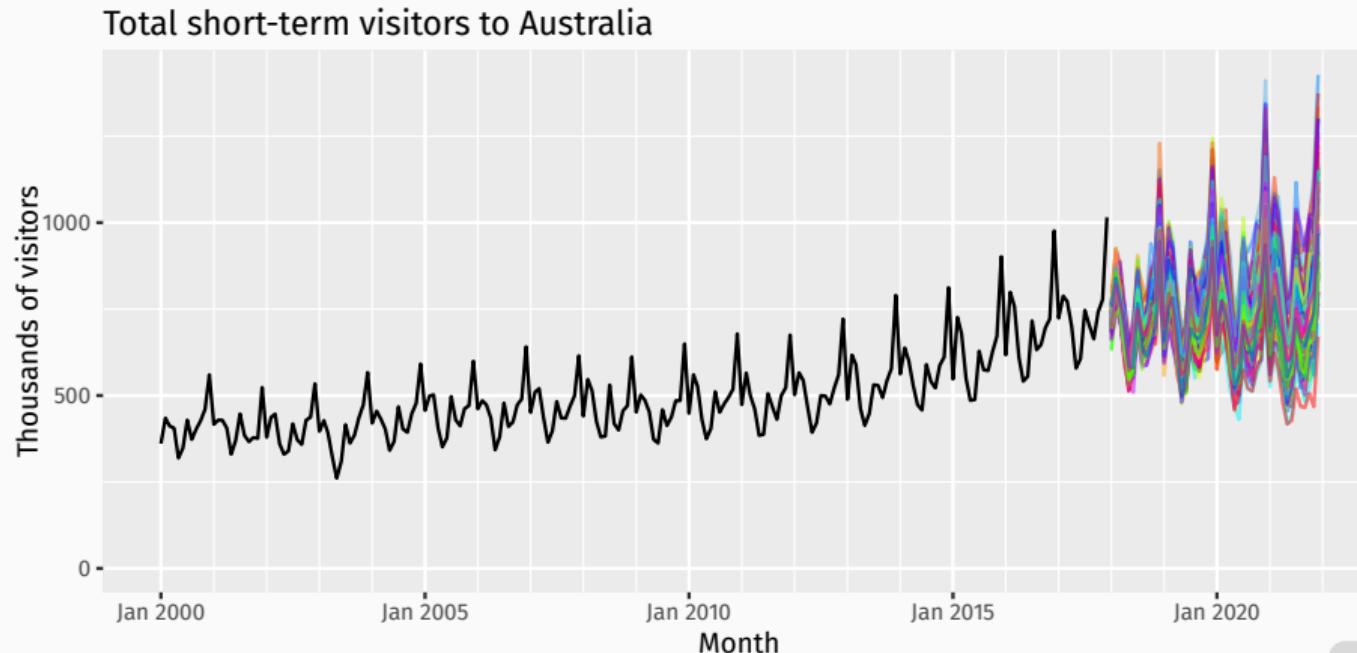
A forecast is an estimate of the probabilities of possible futures.



Simulated futures
from an ETS model

Random futures

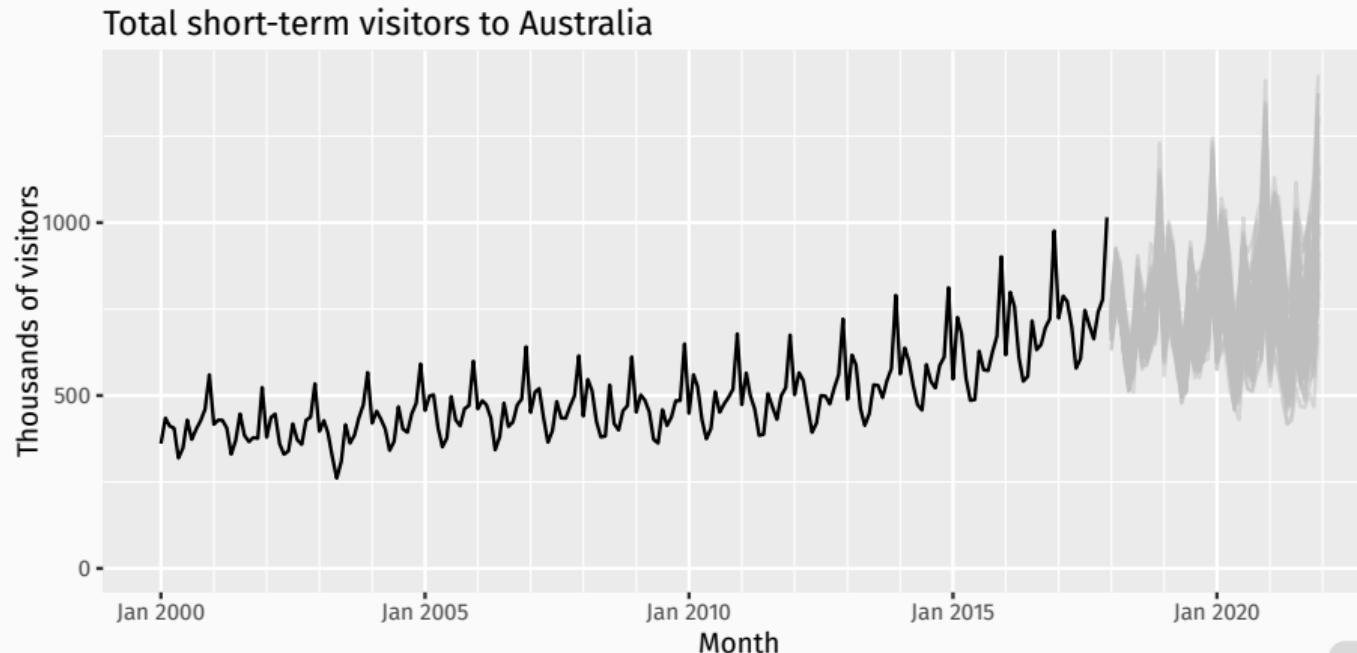
A forecast is an estimate of the probabilities of possible futures.



Simulated futures
from an ETS model

Random futures

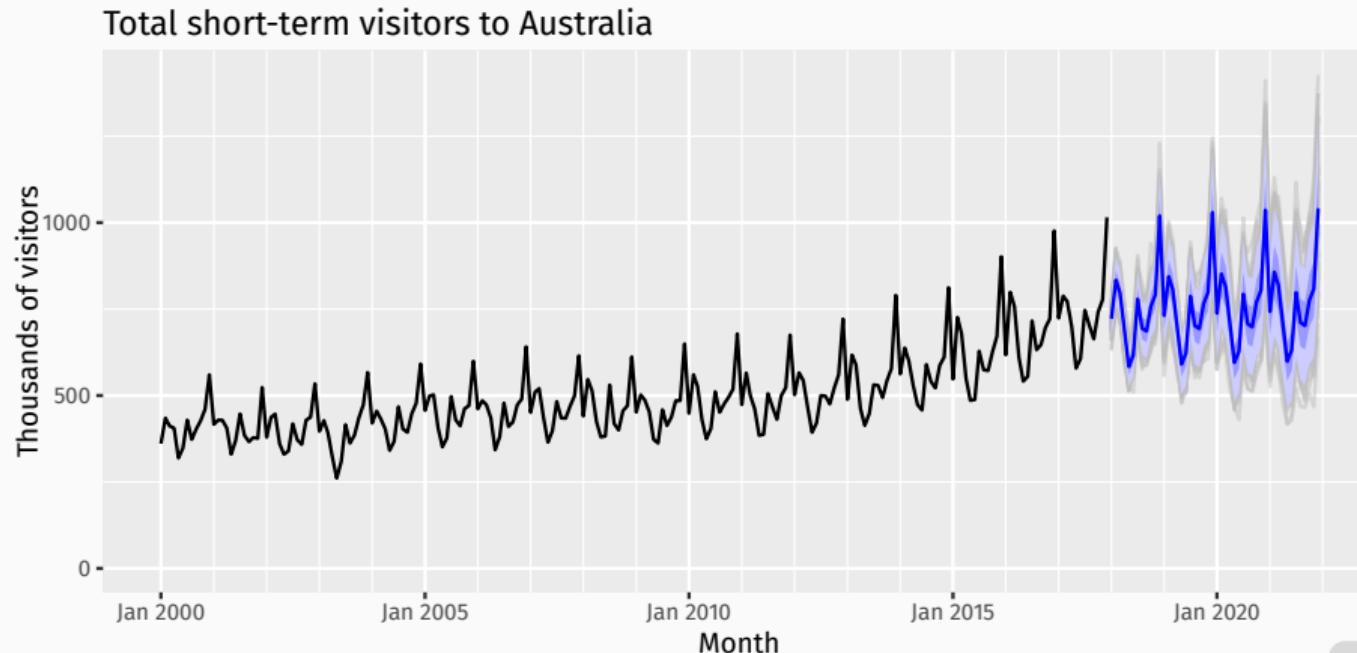
A forecast is an estimate of the probabilities of possible futures.



Simulated futures
from an ETS model

Random futures

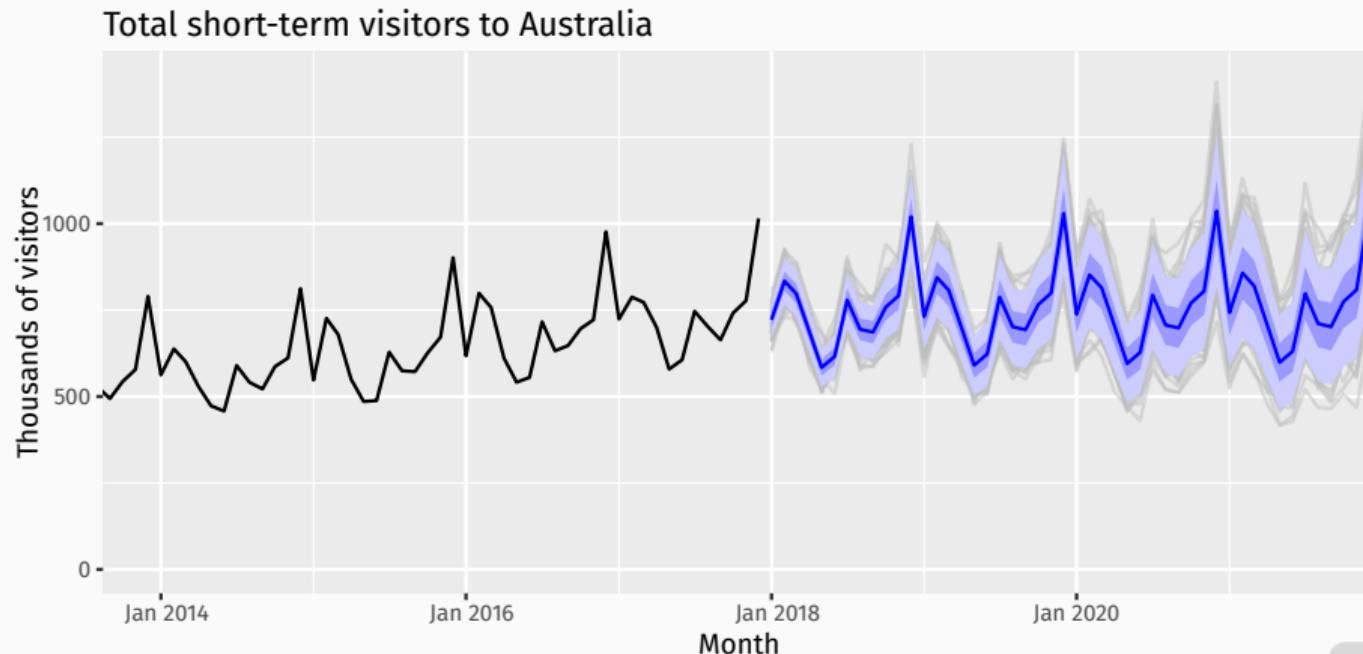
A forecast is an estimate of the probabilities of possible futures.



Simulated futures
from an ETS model

Random futures

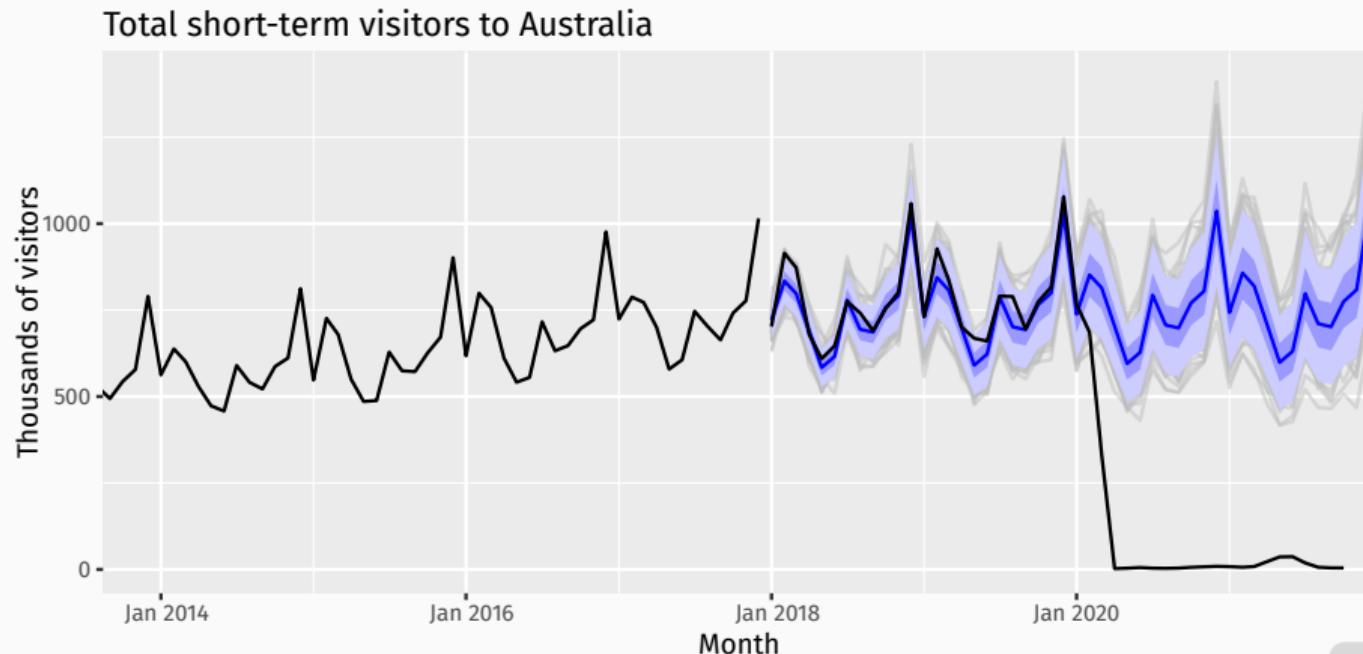
A forecast is an estimate of the probabilities of possible futures.



Simulated futures
from an ETS model

Random futures

A forecast is an estimate of the probabilities of possible futures.

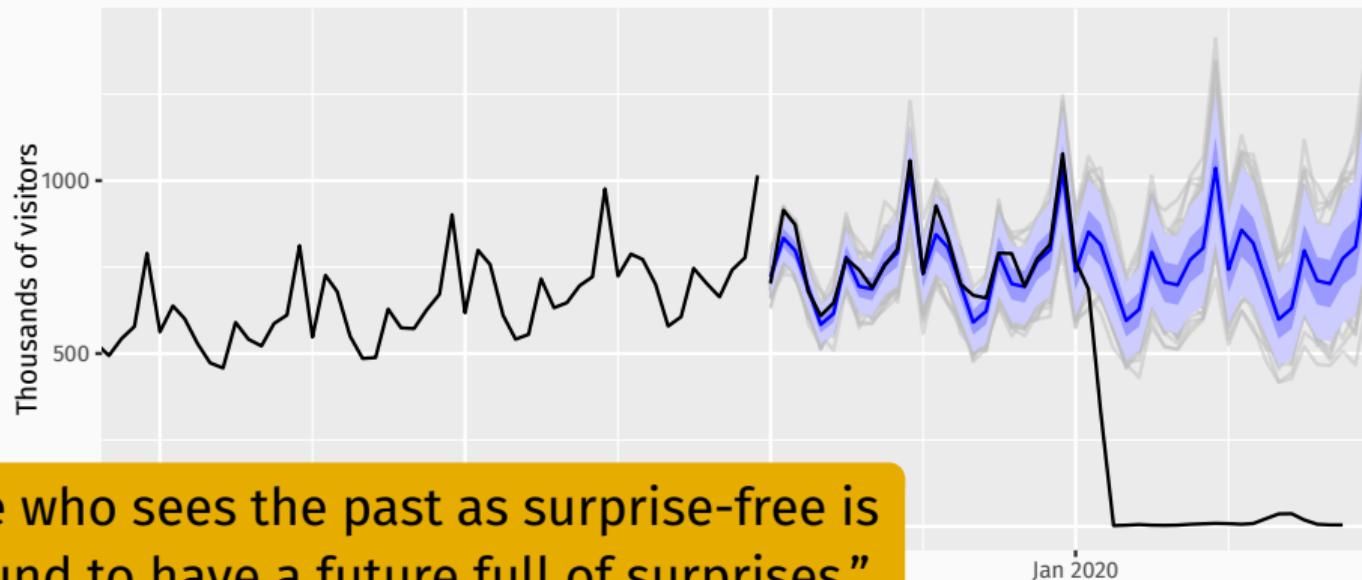


Simulated futures
from an ETS model

Random futures

A forecast is an estimate of the probabilities of possible futures.

Total short-term visitors to Australia



“He who sees the past as surprise-free is bound to have a future full of surprises.”

(Amos Tversky)

Simulated futures
from an ETS model

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 Statistical forecasting

2 Benchmark methods

3 Lab Session 11

4 Residual diagnostics

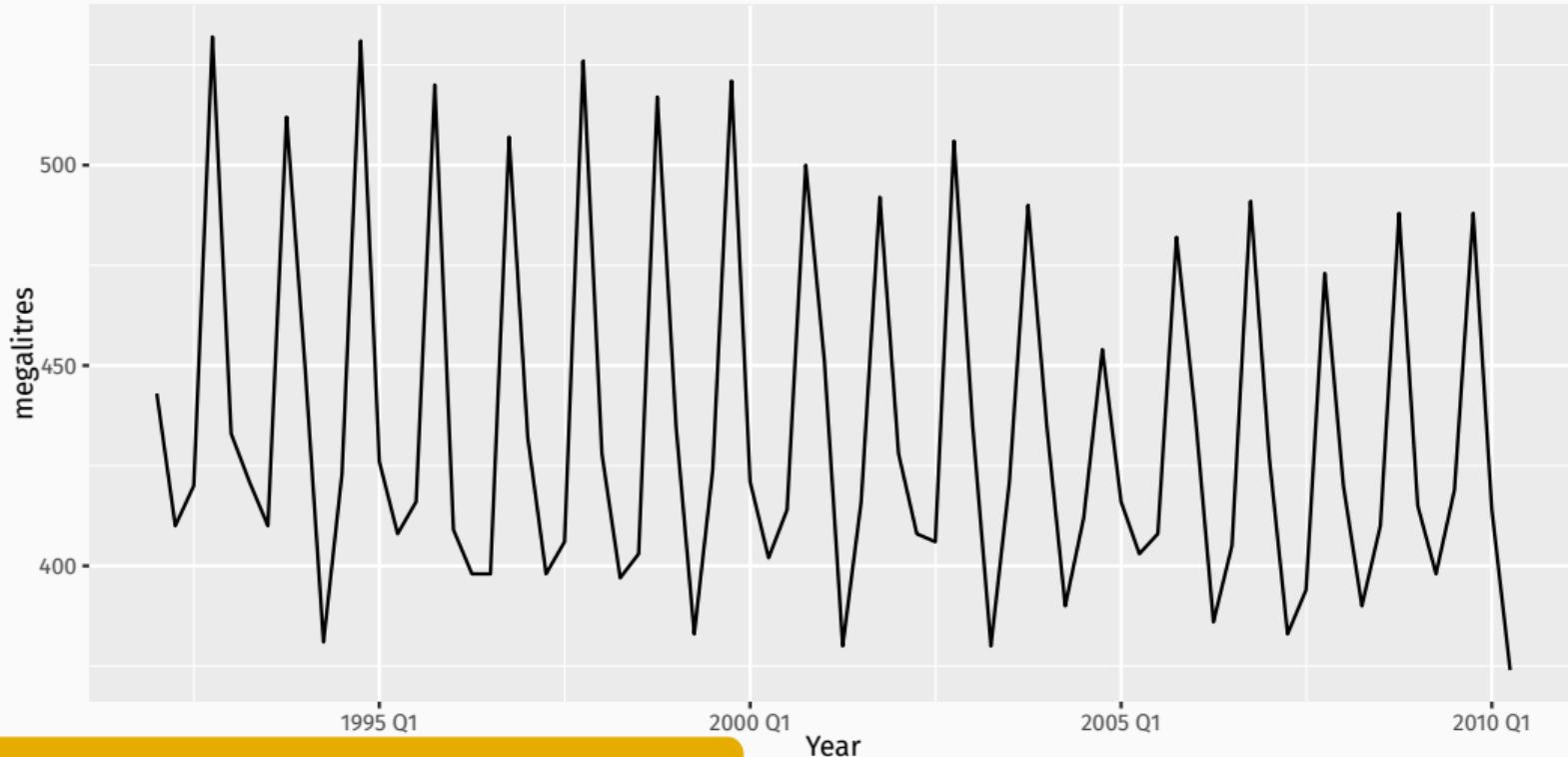
5 Lab Session 12

6 Forecast accuracy measures

7 Lab Session 13

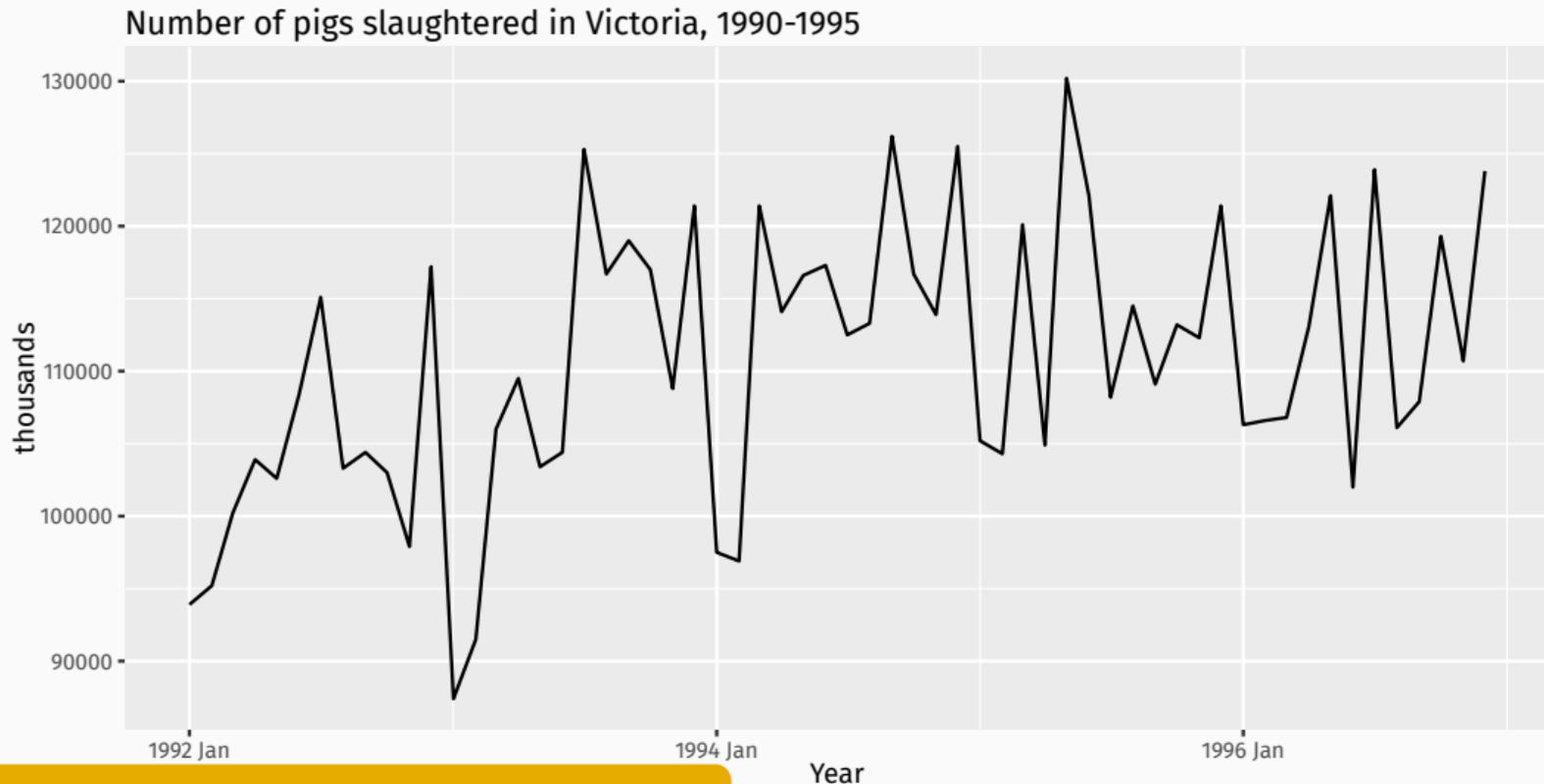
Some simple forecasting methods

Australian quarterly beer production



How would you forecast these series?

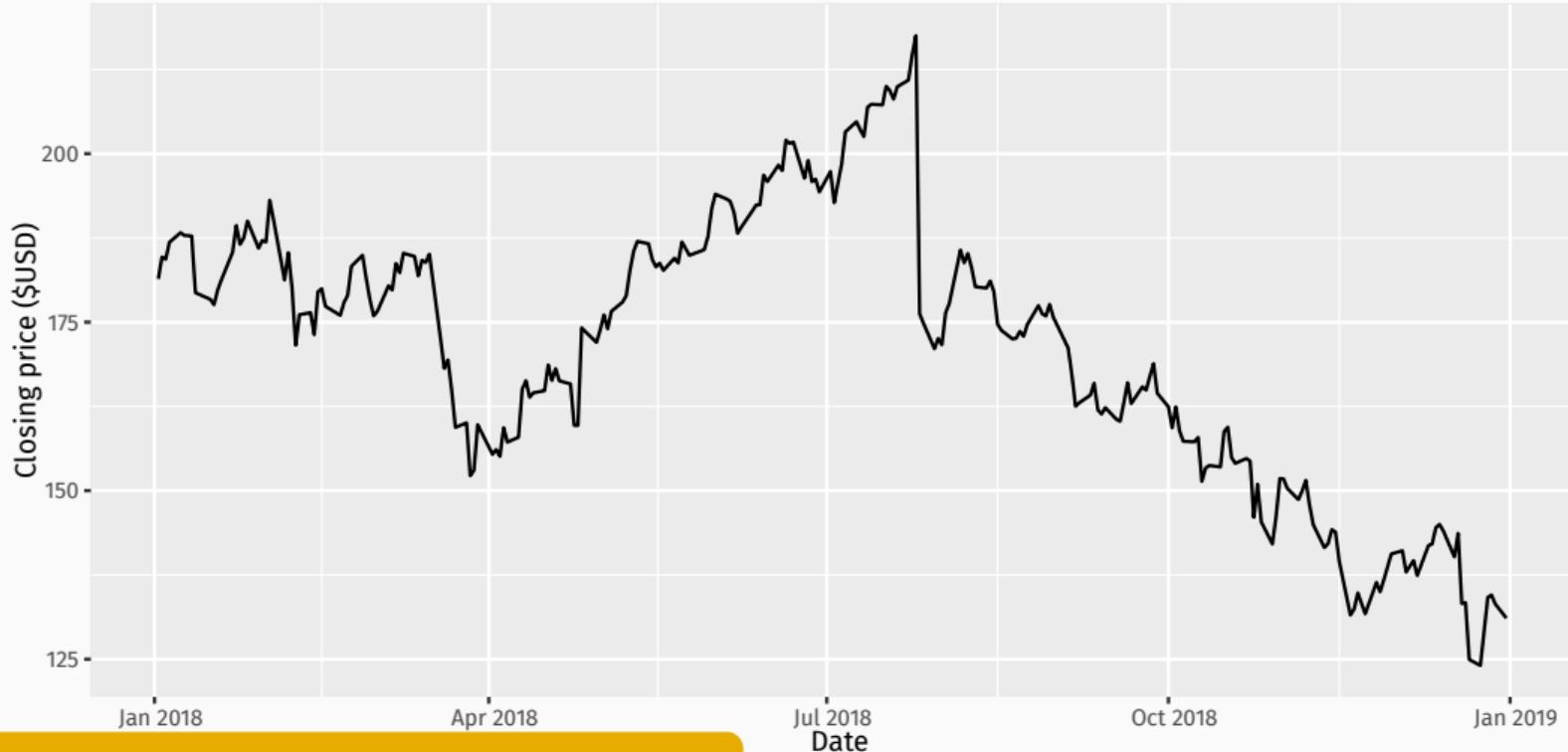
Some simple forecasting methods



How would you forecast these series?

Some simple forecasting methods

Facebook closing stock price in 2018

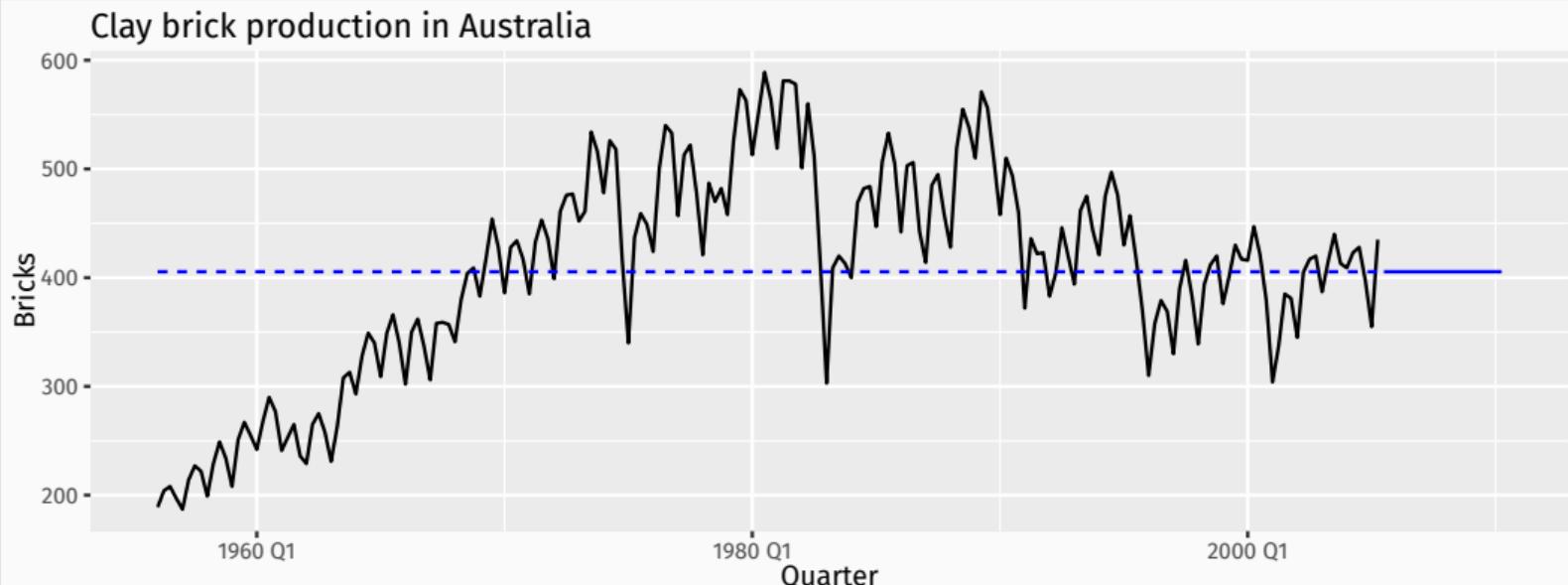


How would you forecast these series?

Some simple forecasting methods

MEAN(y): Average method

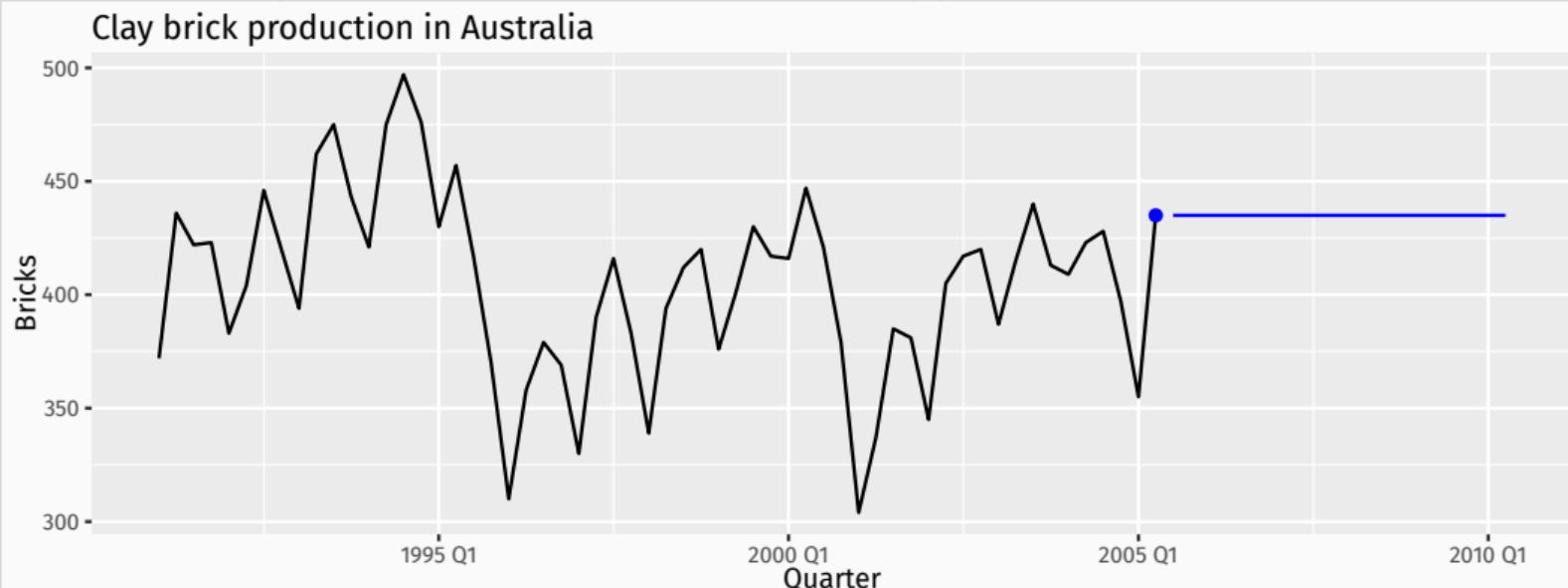
- Forecast of all future values is equal to mean of historical data $\{y_1, \dots, y_T\}$.
- Forecasts: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \dots + y_T)/T$



Some simple forecasting methods

NAIVE(y): Naïve method

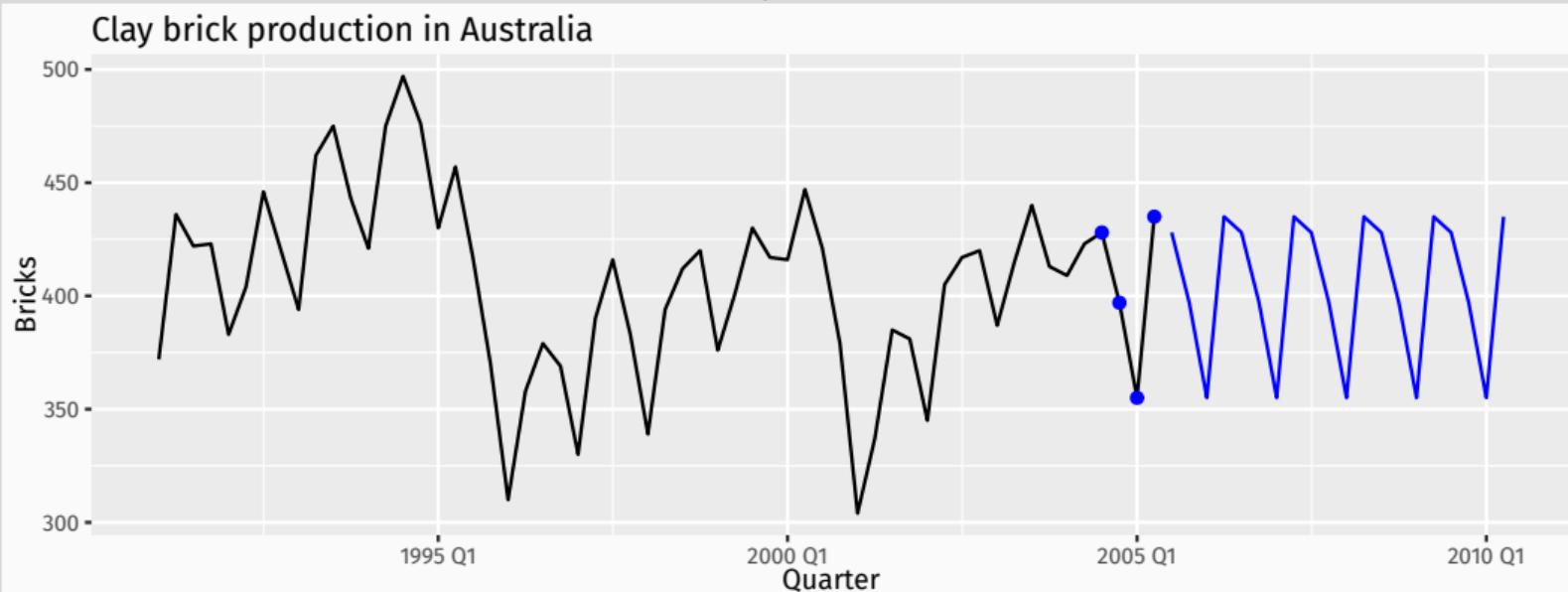
- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.



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 $(h - 1)/m$.



Some simple forecasting methods

RW(y ~ drift()): Drift method

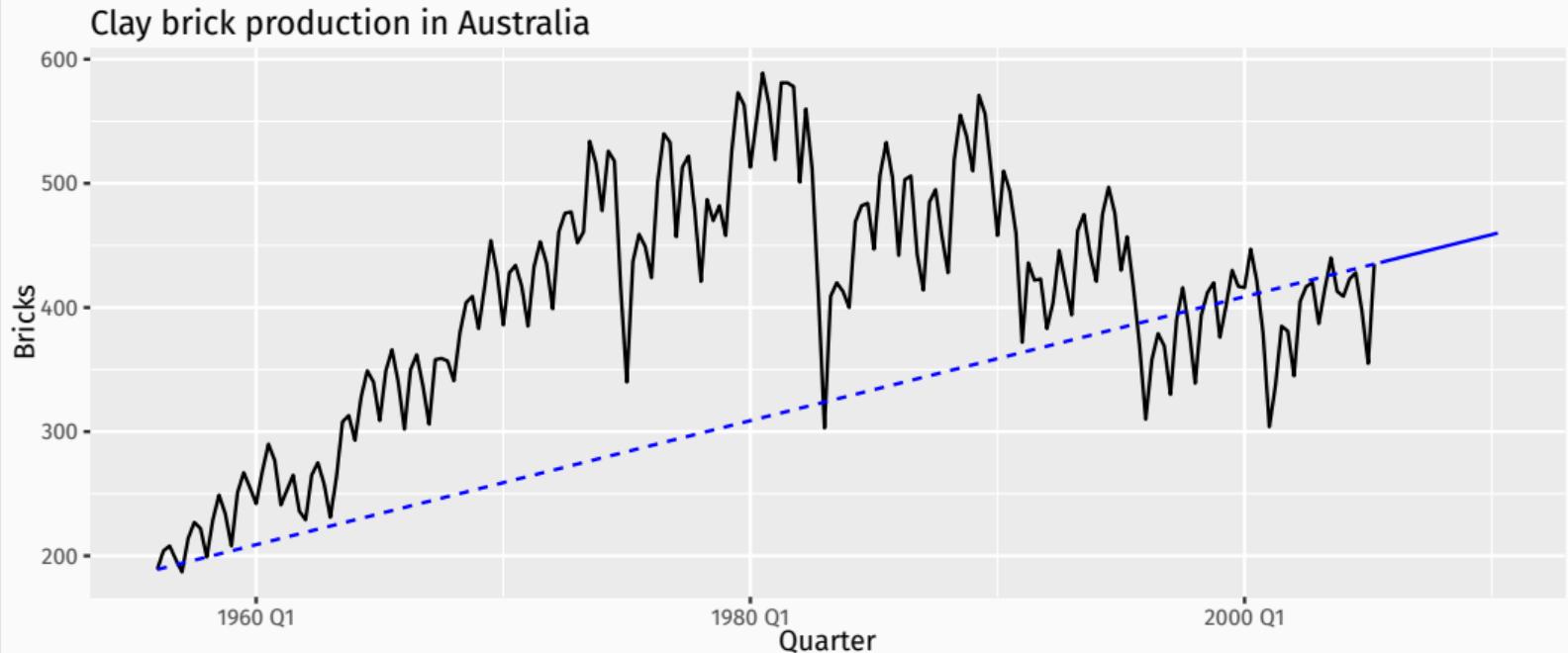
- Forecasts equal to last value plus average change.
- Forecasts:

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

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

Some simple forecasting methods

Drift method



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

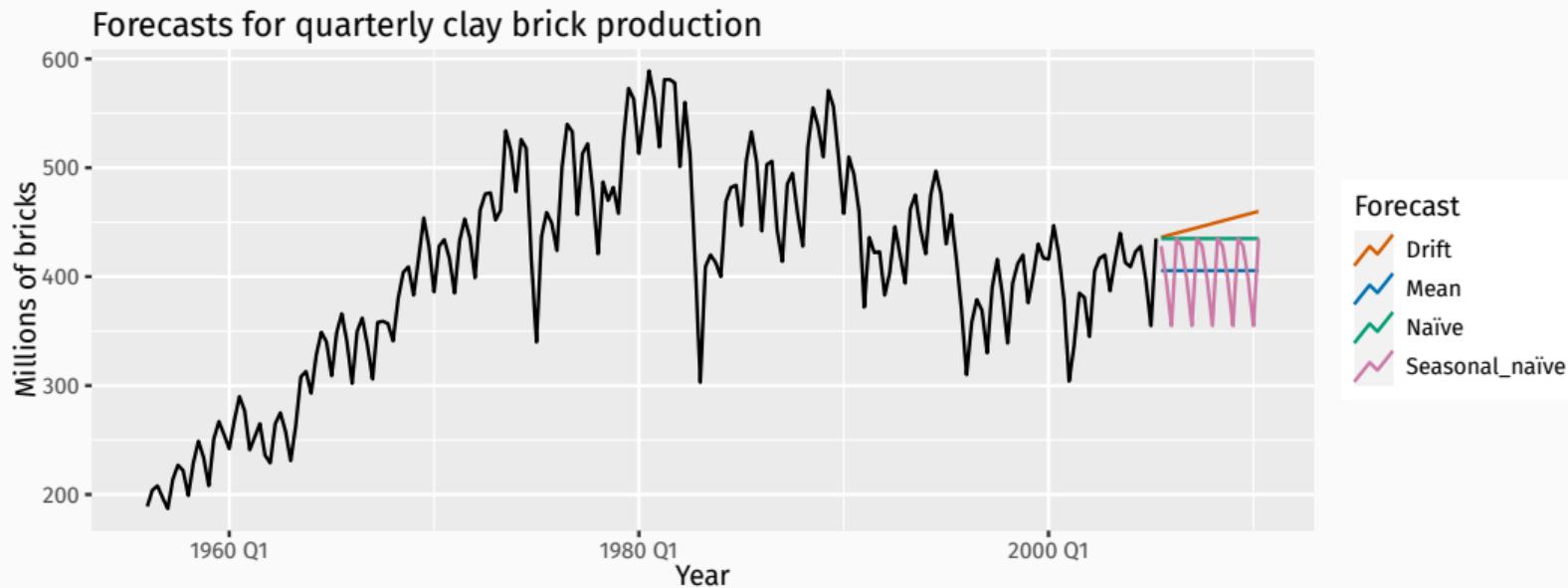
```
brick_fc <- brick_fit |>  
  forecast(h = "5 years")
```

```
# 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  
# i 76 more rows
```

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

Visualising forecasts

```
brick_fc |>  
  autoplot(aus_production, level = NULL) +  
  labs(title = "Forecasts for quarterly clay brick production",  
       x = "Year", y = "Millions of bricks") +  
  guides(colour = guide_legend(title = "Forecast"))
```



Prediction intervals

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

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

Prediction intervals

```
brick_fc |>  
  hilo(level = c(50, 75)) |>  
  mutate(lower = `50%`$lower, upper = `50%`$upper)
```

# A tsibble: 80 x 8 [1Q]	# Key: .model [4]	.model	Quarter	Bricks	.mean	`50%`	`75%`	lower	upper
		<chr>	<qtr>	<dist>	<dbl>	<hilo>	<hilo>	<dbl>	<dbl>
1	Seasonal_naïve	2005 Q3	N(428, 2336)	428	[395, 461]	50	[372, 484]	75	395.
2	Seasonal_naïve	2005 Q4	N(397, 2336)	397	[364, 430]	50	[341, 453]	75	364.
3	Seasonal_naïve	2006 Q1	N(355, 2336)	355	[322, 388]	50	[299, 411]	75	322.
4	Seasonal_naïve	2006 Q2	N(435, 2336)	435	[402, 468]	50	[379, 491]	75	402.
5	Seasonal_naïve	2006 Q3	N(428, 4672)	428	[382, 474]	50	[349, 507]	75	382.
6	Seasonal_naïve	2006 Q4	N(397, 4672)	397	[351, 443]	50	[318, 476]	75	351.
7	Seasonal_naïve	2007 Q1	N(355, 4672)	355	[309, 401]	50	[276, 434]	75	309.
8	Seasonal_naïve	2007 Q2	N(435, 4672)	435	[389, 481]	50	[356, 514]	75	389.
9	Seasonal_naïve	2007 Q3	N(428, 5600)	428	[378, 478]	50	[328, 524]	75	378.
10	Seasonal_naïve	2007 Q4	N(397, 5600)	397	[351, 443]	50	[318, 476]	75	351.
11	Seasonal_naïve	2008 Q1	N(355, 5600)	355	[309, 401]	50	[276, 434]	75	309.
12	Seasonal_naïve	2008 Q2	N(435, 5600)	435	[389, 481]	50	[356, 514]	75	389.
13	Seasonal_naïve	2008 Q3	N(428, 6560)	428	[378, 478]	50	[328, 524]	75	378.
14	Seasonal_naïve	2008 Q4	N(397, 6560)	397	[351, 443]	50	[318, 476]	75	351.
15	Seasonal_naïve	2009 Q1	N(355, 6560)	355	[309, 401]	50	[276, 434]	75	309.
16	Seasonal_naïve	2009 Q2	N(435, 6560)	435	[389, 481]	50	[356, 514]	75	389.
17	Seasonal_naïve	2009 Q3	N(428, 7520)	428	[378, 478]	50	[328, 524]	75	378.
18	Seasonal_naïve	2009 Q4	N(397, 7520)	397	[351, 443]	50	[318, 476]	75	351.
19	Seasonal_naïve	2010 Q1	N(355, 7520)	355	[309, 401]	50	[276, 434]	75	309.
20	Seasonal_naïve	2010 Q2	N(435, 7520)	435	[389, 481]	50	[356, 514]	75	389.
21	Seasonal_naïve	2010 Q3	N(428, 8480)	428	[378, 478]	50	[328, 524]	75	378.
22	Seasonal_naïve	2010 Q4	N(397, 8480)	397	[351, 443]	50	[318, 476]	75	351.
23	Seasonal_naïve	2011 Q1	N(355, 8480)	355	[309, 401]	50	[276, 434]	75	309.
24	Seasonal_naïve	2011 Q2	N(435, 8480)	435	[389, 481]	50	[356, 514]	75	389.
25	Seasonal_naïve	2011 Q3	N(428, 9440)	428	[378, 478]	50	[328, 524]	75	378.
26	Seasonal_naïve	2011 Q4	N(397, 9440)	397	[351, 443]	50	[318, 476]	75	351.
27	Seasonal_naïve	2012 Q1	N(355, 9440)	355	[309, 401]	50	[276, 434]	75	309.
28	Seasonal_naïve	2012 Q2	N(435, 9440)	435	[389, 481]	50	[356, 514]	75	389.
29	Seasonal_naïve	2012 Q3	N(428, 10400)	428	[378, 478]	50	[328, 524]	75	378.
30	Seasonal_naïve	2012 Q4	N(397, 10400)	397	[351, 443]	50	[318, 476]	75	351.
31	Seasonal_naïve	2013 Q1	N(355, 10400)	355	[309, 401]	50	[276, 434]	75	309.
32	Seasonal_naïve	2013 Q2	N(435, 10400)	435	[389, 481]	50	[356, 514]	75	389.
33	Seasonal_naïve	2013 Q3	N(428, 11360)	428	[378, 478]	50	[328, 524]	75	378.
34	Seasonal_naïve	2013 Q4	N(397, 11360)	397	[351, 443]	50	[318, 476]	75	351.
35	Seasonal_naïve	2014 Q1	N(355, 11360)	355	[309, 401]	50	[276, 434]	75	309.
36	Seasonal_naïve	2014 Q2	N(435, 11360)	435	[389, 481]	50	[356, 514]	75	389.
37	Seasonal_naïve	2014 Q3	N(428, 12320)	428	[378, 478]	50	[328, 524]	75	378.
38	Seasonal_naïve	2014 Q4	N(397, 12320)	397	[351, 443]	50	[318, 476]	75	351.
39	Seasonal_naïve	2015 Q1	N(355, 12320)	355	[309, 401]	50	[276, 434]	75	309.
40	Seasonal_naïve	2015 Q2	N(435, 12320)	435	[389, 481]	50	[356, 514]	75	389.
41	Seasonal_naïve	2015 Q3	N(428, 13280)	428	[378, 478]	50	[328, 524]	75	378.
42	Seasonal_naïve	2015 Q4	N(397, 13280)	397	[351, 443]	50	[318, 476]	75	351.
43	Seasonal_naïve	2016 Q1	N(355, 13280)	355	[309, 401]	50	[276, 434]	75	309.
44	Seasonal_naïve	2016 Q2	N(435, 13280)	435	[389, 481]	50	[356, 514]	75	389.
45	Seasonal_naïve	2016 Q3	N(428, 14240)	428	[378, 478]	50	[328, 524]	75	378.
46	Seasonal_naïve	2016 Q4	N(397, 14240)	397	[351, 443]	50	[318, 476]	75	351.
47	Seasonal_naïve	2017 Q1	N(355, 14240)	355	[309, 401]	50	[276, 434]	75	309.
48	Seasonal_naïve	2017 Q2	N(435, 14240)	435	[389, 481]	50	[356, 514]	75	389.
49	Seasonal_naïve	2017 Q3	N(428, 15200)	428	[378, 478]	50	[328, 524]	75	378.
50	Seasonal_naïve	2017 Q4	N(397, 15200)	397	[351, 443]	50	[318, 476]	75	351.
51	Seasonal_naïve	2018 Q1	N(355, 15200)	355	[309, 401]	50	[276, 434]	75	309.
52	Seasonal_naïve	2018 Q2	N(435, 15200)	435	[389, 481]	50	[356, 514]	75	389.
53	Seasonal_naïve	2018 Q3	N(428, 16160)	428	[378, 478]	50	[328, 524]	75	378.
54	Seasonal_naïve	2018 Q4	N(397, 16160)	397	[351, 443]	50	[318, 476]	75	351.
55	Seasonal_naïve	2019 Q1	N(355, 16160)	355	[309, 401]	50	[276, 434]	75	309.
56	Seasonal_naïve	2019 Q2	N(435, 16160)	435	[389, 481]	50	[356, 514]	75	389.
57	Seasonal_naïve	2019 Q3	N(428, 17120)	428	[378, 478]	50	[328, 524]	75	378.
58	Seasonal_naïve	2019 Q4	N(397, 17120)	397	[351, 443]	50	[318, 476]	75	351.
59	Seasonal_naïve	2020 Q1	N(355, 17120)	355	[309, 401]	50	[276, 434]	75	309.
60	Seasonal_naïve	2020 Q2	N(435, 17120)	435	[389, 481]	50	[356, 514]	75	389.
61	Seasonal_naïve	2020 Q3	N(428, 18080)	428	[378, 478]	50	[328, 524]	75	378.
62	Seasonal_naïve	2020 Q4	N(397, 18080)	397	[351, 443]	50	[318, 476]	75	351.
63	Seasonal_naïve	2021 Q1	N(355, 18080)	355	[309, 401]	50	[276, 434]	75	309.
64	Seasonal_naïve	2021 Q2	N(435, 18080)	435	[389, 481]	50	[356, 514]	75	389.
65	Seasonal_naïve	2021 Q3	N(428, 19040)	428	[378, 478]	50	[328, 524]	75	378.
66	Seasonal_naïve	2021 Q4	N(397, 19040)	397	[351, 443]	50	[318, 476]	75	351.
67	Seasonal_naïve	2022 Q1	N(355, 19040)	355	[309, 401]	50	[276, 434]	75	309.
68	Seasonal_naïve	2022 Q2	N(435, 19040)	435	[389, 481]	50	[356, 514]	75	389.
69	Seasonal_naïve	2022 Q3	N(428, 19920)	428	[378, 478]	50	[328, 524]	75	378.
70	Seasonal_naïve	2022 Q4	N(397, 19920)	397	[351, 443]	50	[318, 476]	75	351.
71	Seasonal_naïve	2023 Q1	N(355, 19920)	355	[309, 401]	50	[276, 434]	75	309.
72	Seasonal_naïve	2023 Q2	N(435, 19920)	435	[389, 481]	50	[356, 514]	75	389.
73	Seasonal_naïve	2023 Q3	N(428, 20800)	428	[378, 478]	50	[328, 524]	75	378.
74	Seasonal_naïve	2023 Q4	N(397, 20800)	397	[351, 443]	50	[318, 476]	75	351.
75	Seasonal_naïve	2024 Q1	N(355, 20800)	355	[309, 401]	50	[276, 434]	75	309.
76	Seasonal_naïve	2024 Q2	N(435, 20800)	435	[389, 481]	50	[356, 514]	75	389.
77	Seasonal_naïve	2024 Q3	N(428, 21680)	428	[378, 478]	50	[328, 524]	75	378.
78	Seasonal_naïve	2024 Q4	N(397, 21680)	397	[351, 443]	50	[318, 476]	75	351.
79	Seasonal_naïve	2025 Q1	N(355, 21680)	355	[309, 401]	50	[276, 434]	75	309.
80	Seasonal_naïve	2025 Q2	N(435, 21680)	435	[389, 481]	50	[356, 514]	75	389.

Outline

1 Statistical forecasting

2 Benchmark methods

3 Lab Session 11

4 Residual diagnostics

5 Lab Session 12

6 Forecast accuracy measures

7 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 Statistical forecasting
- 2 Benchmark methods
- 3 Lab Session 11
- 4 Residual diagnostics
- 5 Lab Session 12
- 6 Forecast accuracy measures
- 7 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")  
fb_stock |> autoplot(Close)
```



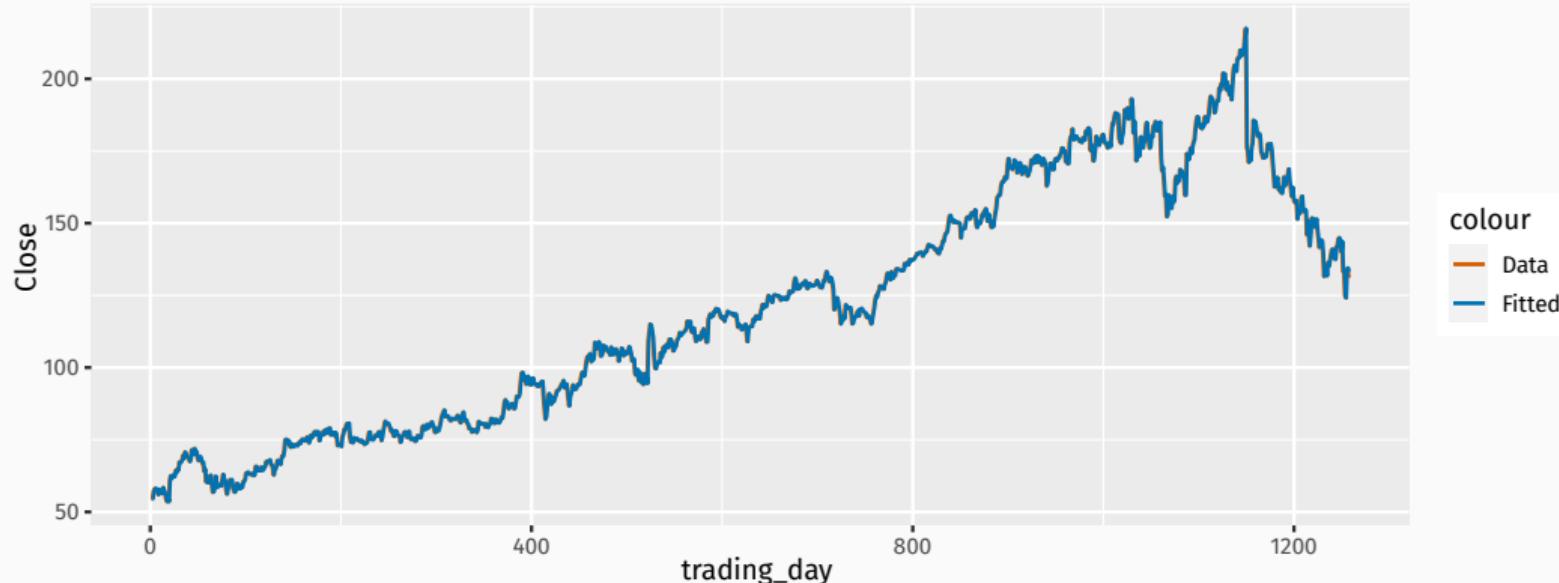
Facebook closing stock price

```
fb_stock <- fb_stock |>  
  mutate(trading_day = row_number()) |>  
  update_tsibble(index = trading_day, regular = TRUE)  
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
```

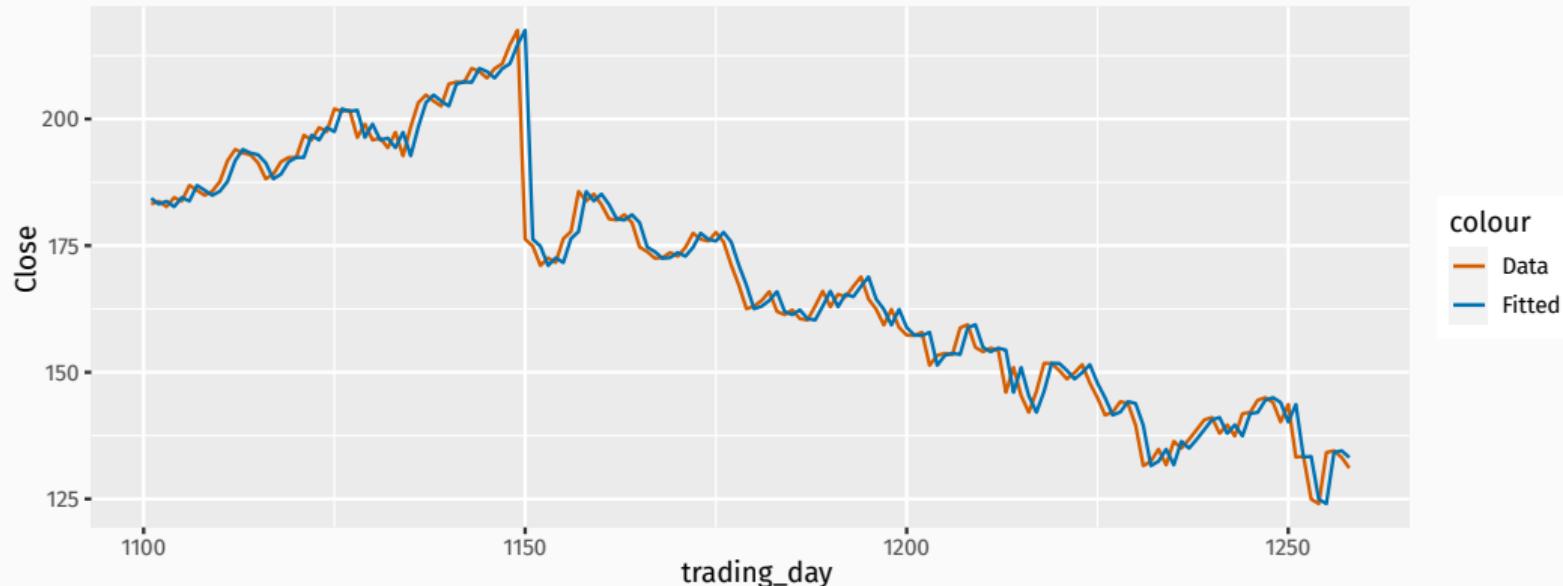
Facebook closing stock price

```
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

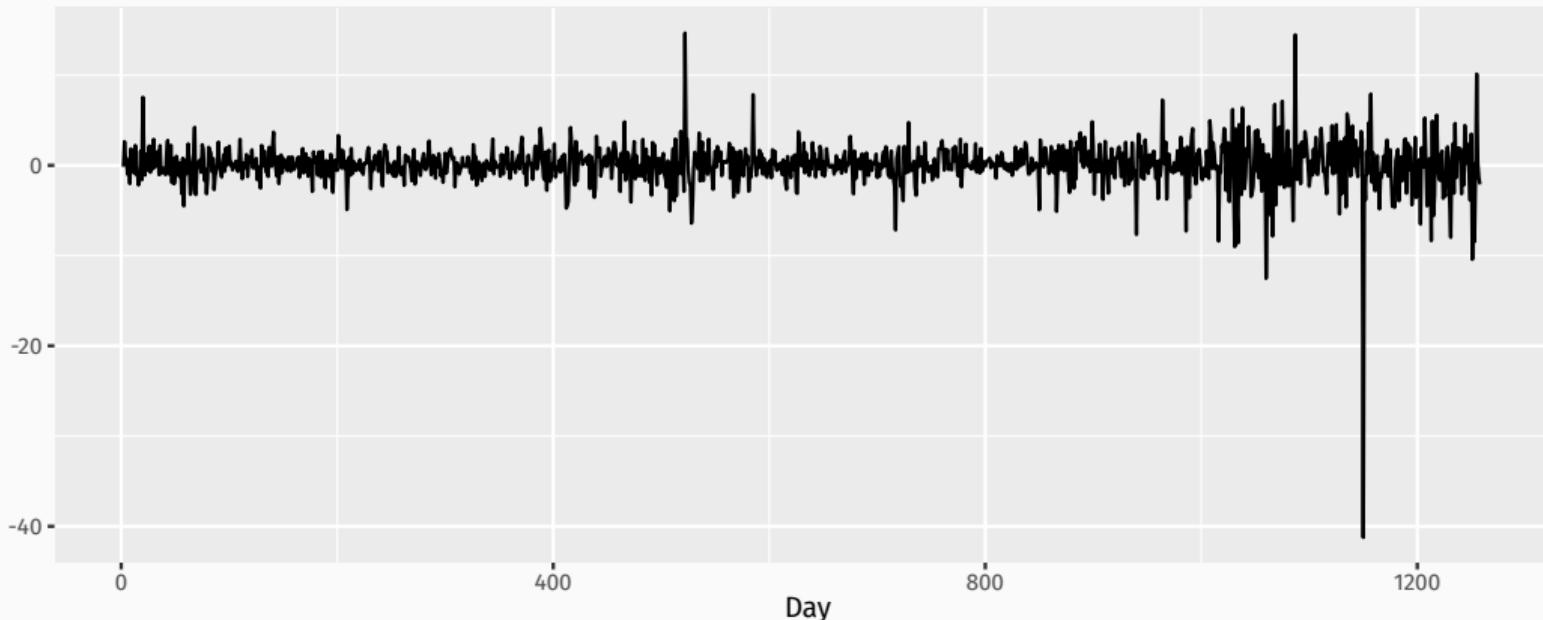
```
augment(fit) |>  
  filter(trading_day > 1100) |>  
  ggplot(aes(x = trading_day)) +  
  geom_line(aes(y = Close, colour = "Data")) +  
  geom_line(aes(y = .fitted, colour = "Fitted"))
```



Facebook closing stock price

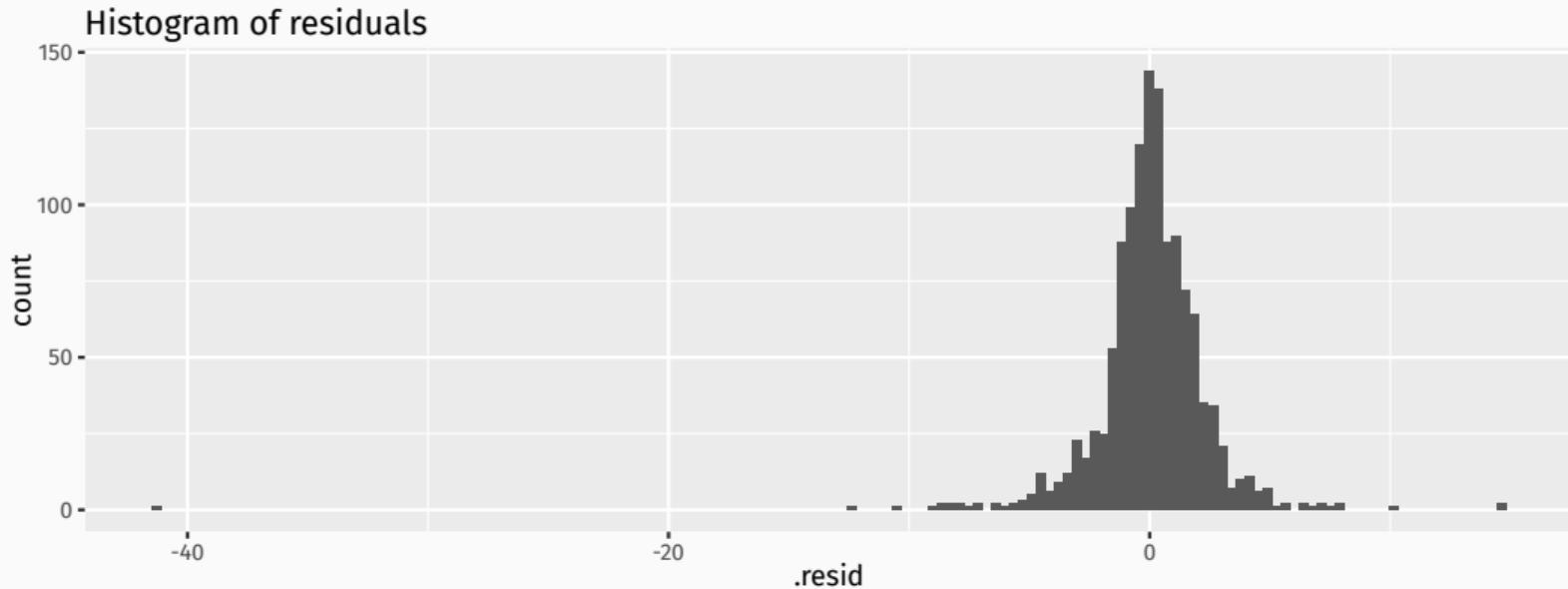
```
augment(fit) |>  
  autoplot(.resid) +  
  labs(x = "Day", y = "", title = "Residuals from naïve method")
```

Residuals from naïve method



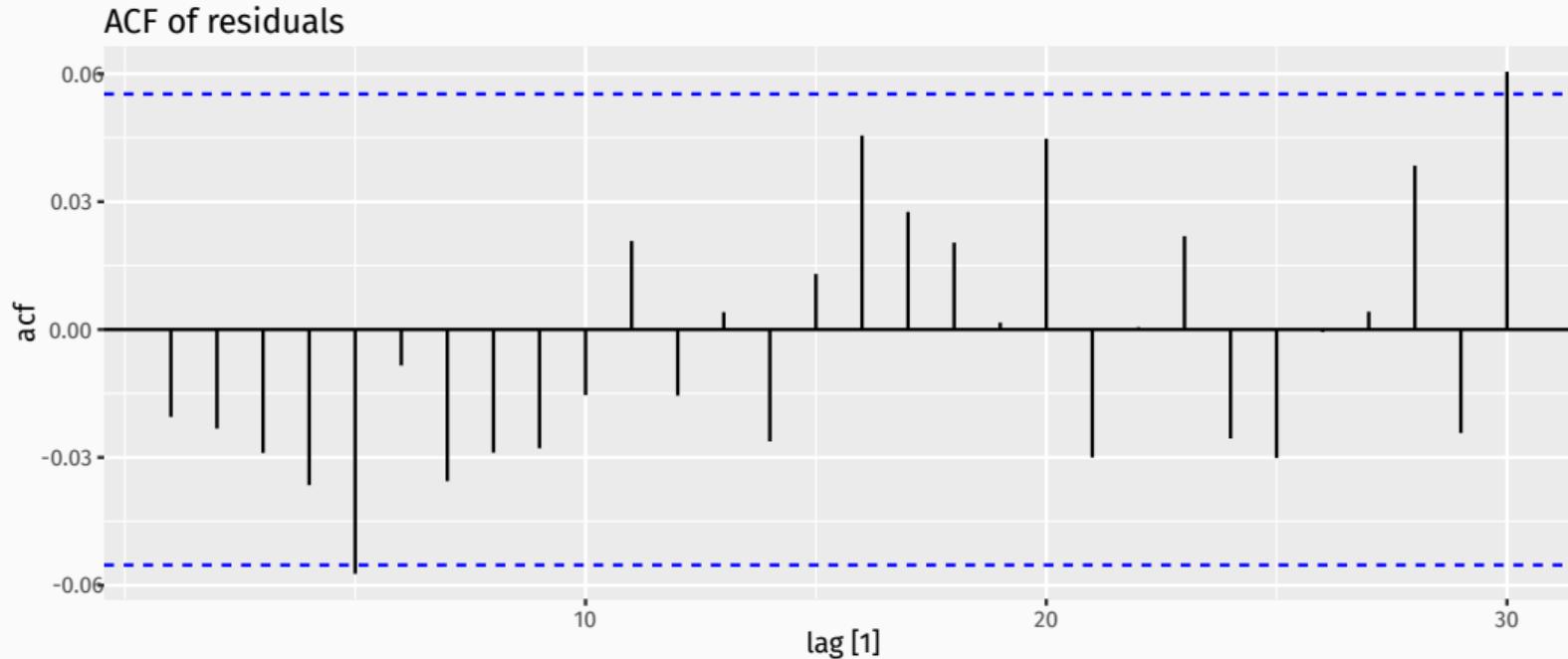
Facebook closing stock price

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



Facebook closing stock price

```
augment(fit) |>  
ACF(.resid) |>  
autoplot() + labs(title = "ACF of residuals")
```

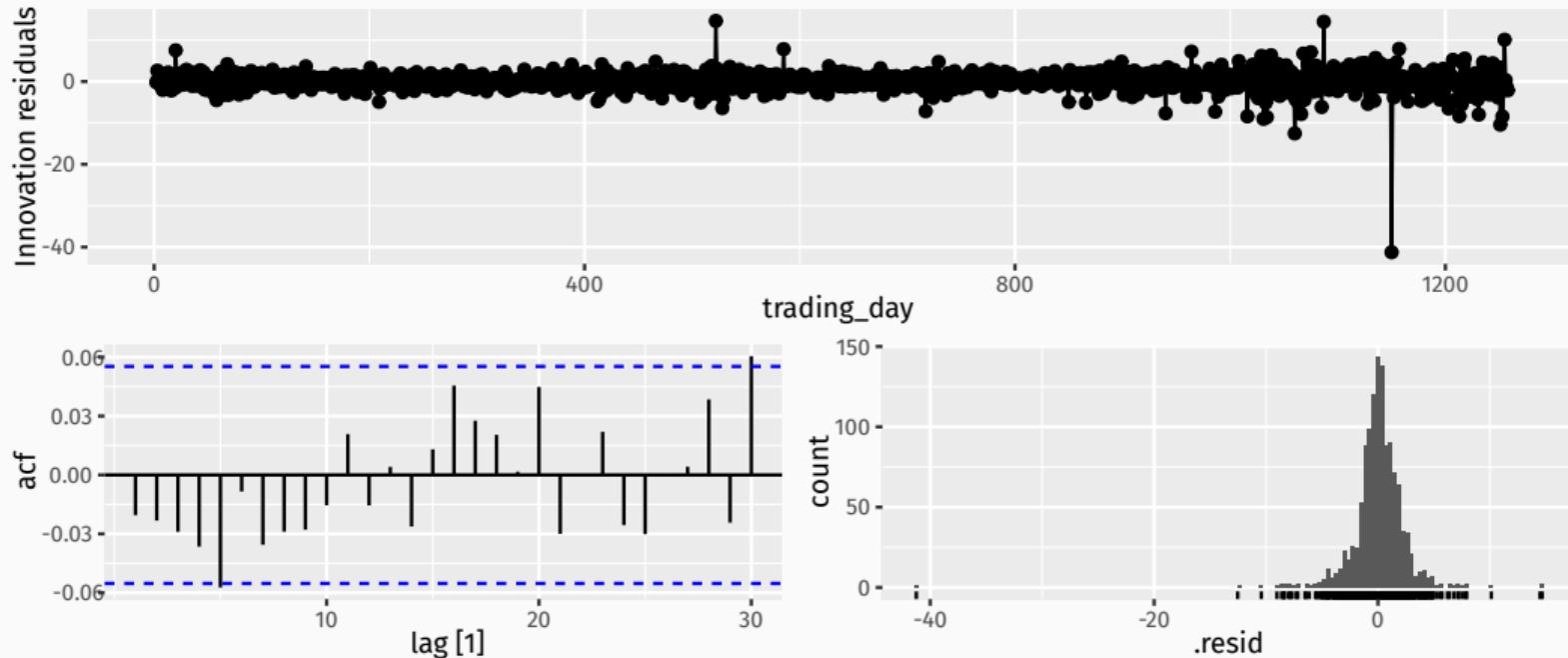


ACF of residuals

- 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

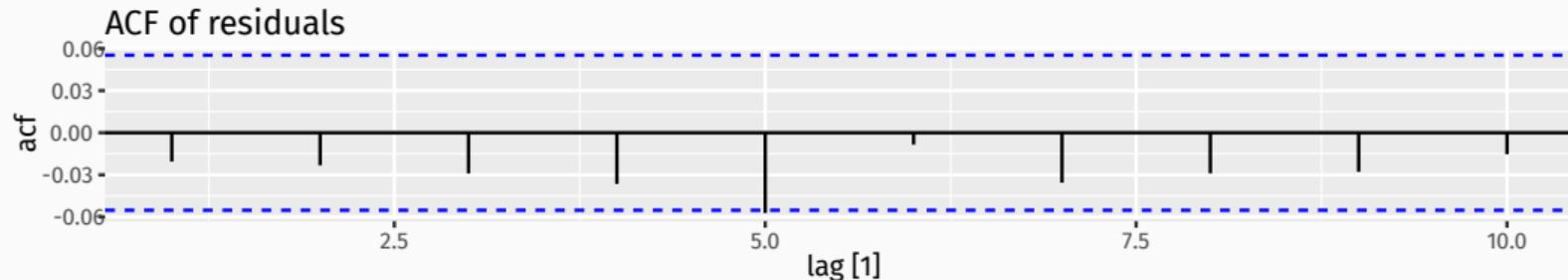
Test whether *whole set* of r_k values is significantly different from zero set.

$$Q = T(T + 2) \sum_{k=1}^{\ell} (T - k)^{-1} r_k^2 \quad \text{where } \ell = \text{max lag and } T = \# \text{ observations}$$

- 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 and T large, $Q \sim \chi^2$ with ℓ degrees of freedom.

Ljung-Box test

$$Q = T(T + 2) \sum_{k=1}^{\ell} (T - k)^{-1} r_k^2 \quad \text{where } \ell = \text{max lag and } T = \# \text{ observations.}$$



```
# lag = h
augment(fit) |> features(.resid, ljung_box, lag = 10)
```

```
# A tibble: 1 x 4
  Symbol .model      lb_stat lb_pvalue
  <chr>  <chr>       <dbl>     <dbl>
1 FB    NAIVE(Close) 12.1      0.276
```

Outline

1 Statistical forecasting

2 Benchmark methods

3 Lab Session 11

4 Residual diagnostics

5 Lab Session 12

6 Forecast accuracy measures

7 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 Statistical forecasting
- 2 Benchmark methods
- 3 Lab Session 11
- 4 Residual diagnostics
- 5 Lab Session 12
- 6 Forecast accuracy measures
- 7 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.

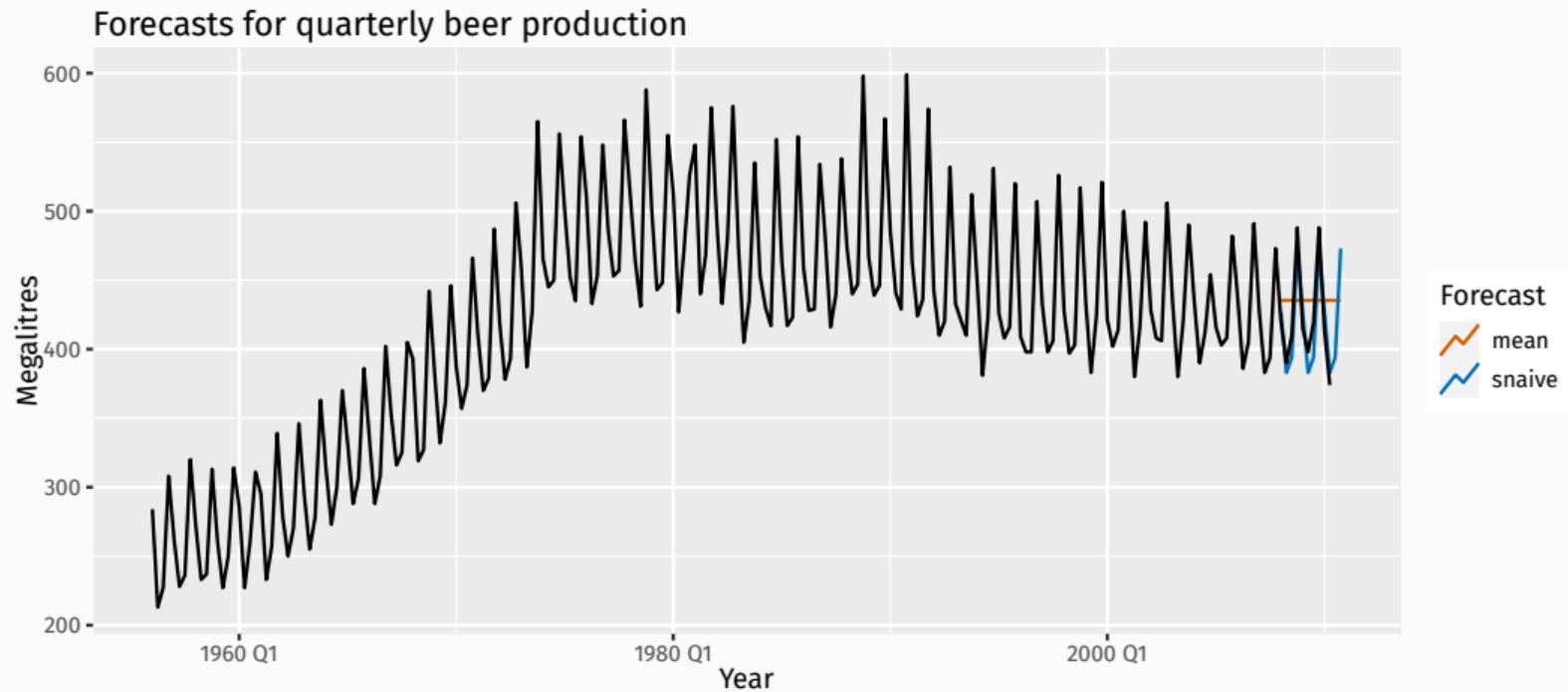
$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

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) +
  labs(title ="Forecasts for quarterly beer production",
       x ="Year", y ="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 = $\text{mean}(|e_{T+h}|)$

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

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

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

- 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

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

Measures of forecast accuracy

Mean Absolute Scaled Error

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

- For non-seasonal series, scale uses naïve forecasts:

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

- For seasonal series, scale uses seasonal naïve forecasts:

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

where m is the seasonal frequency

Measures of forecast accuracy

Mean Absolute Scaled Error

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

- For non-seasonal series, scale uses naïve forecasts:

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

- For seasonal series, scale uses seasonal naïve forecasts:

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

where m is the seasonal frequency

Proposed by Hyndman and Koehler (IJF, 2006).

Measures of forecast accuracy

Root Mean Squared Scaled Error

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

- For non-seasonal series, scale uses naïve forecasts:

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

- For seasonal series, scale uses seasonal naïve forecasts:

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

where m is the seasonal frequency

Proposed by Hyndman and Koehler (IJF, 2006).

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 Statistical forecasting

2 Benchmark methods

3 Lab Session 11

4 Residual diagnostics

5 Lab Session 12

6 Forecast accuracy measures

7 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.