https://workshop.f4sg.org/africast/

Africast-Time Series Analysis & Forecasting Using R

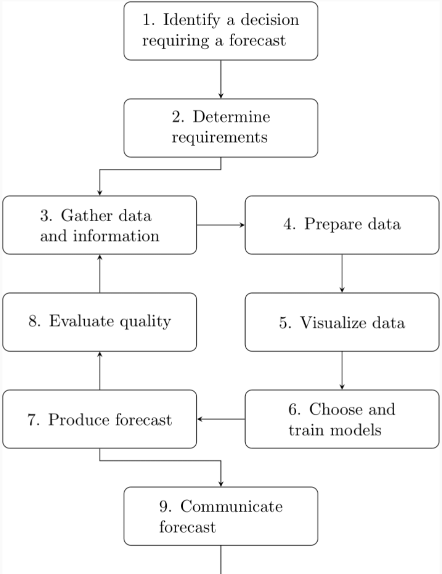
5. Basic modeling and forecasting

tsibble feasts fable

Outline

Outline

Forecasting workflow



Statistical forecasting steps

- Prepare data.
- Visualise data.
- Choosing and fitting models (specify and train models).
- Produce forecast.
- Evaluate quality.

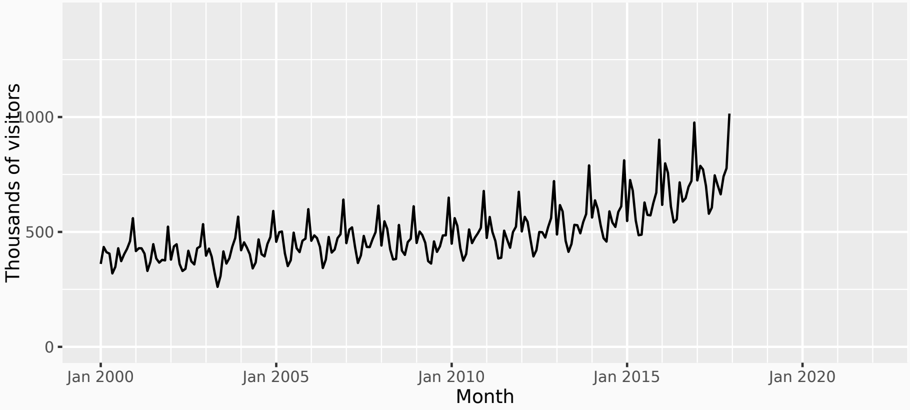
What is a forecast?

A forecast is an estimate of the probability distribution of a variable to be observed in the future.

What is a forecast?

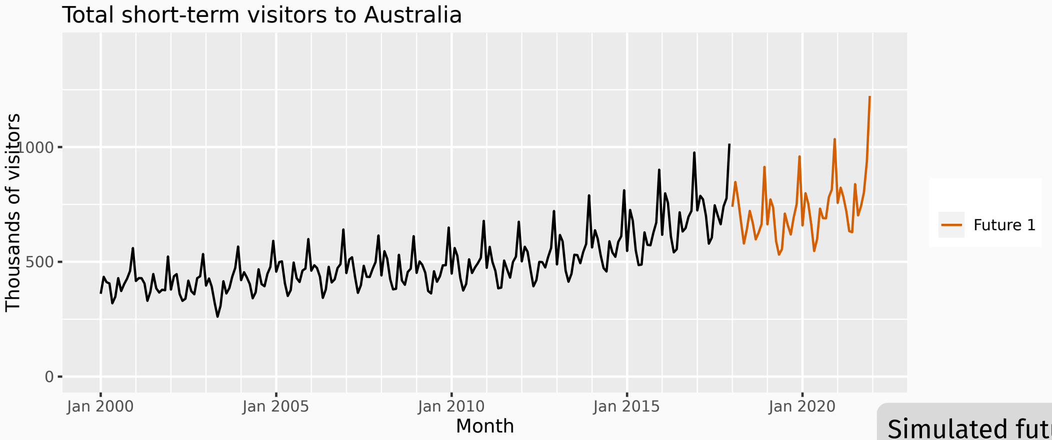
A forecast is an estimate of the probability distribution of a variable to be observed in the future.

Total short-term visitors to Australia



What is a forecast?

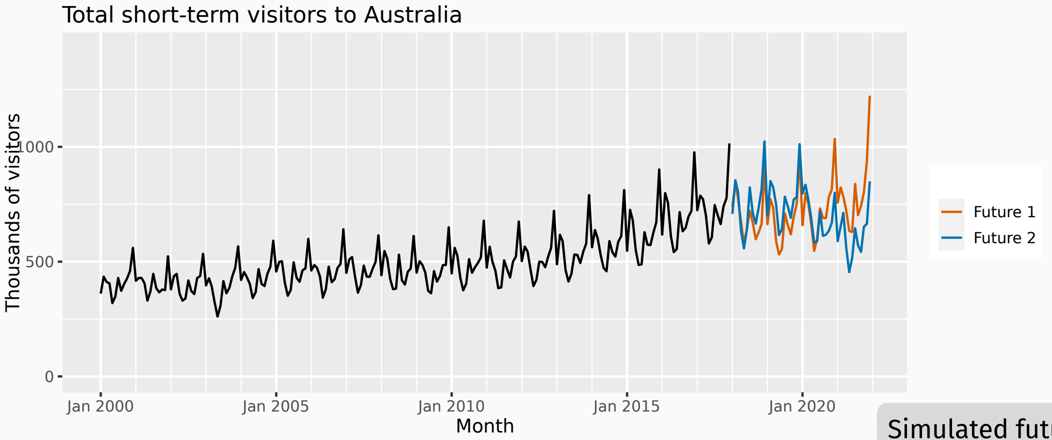
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS

What is a forecast?

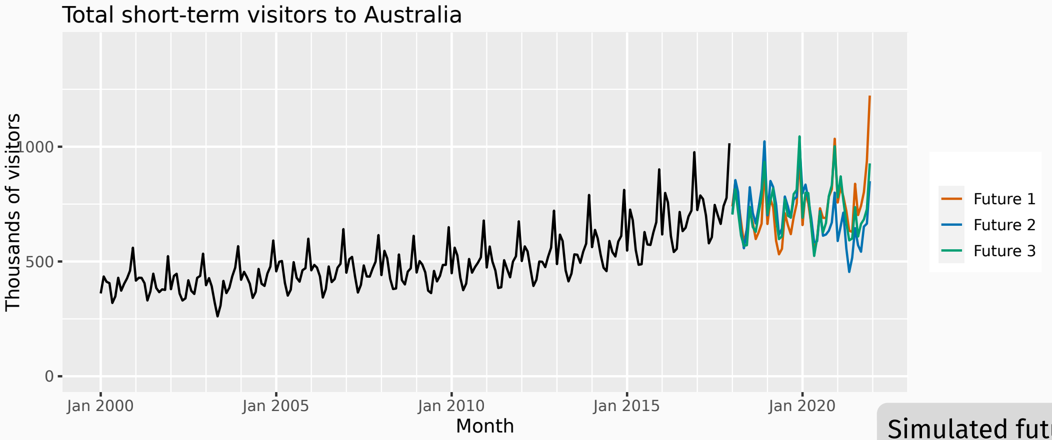
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS

What is a forecast?

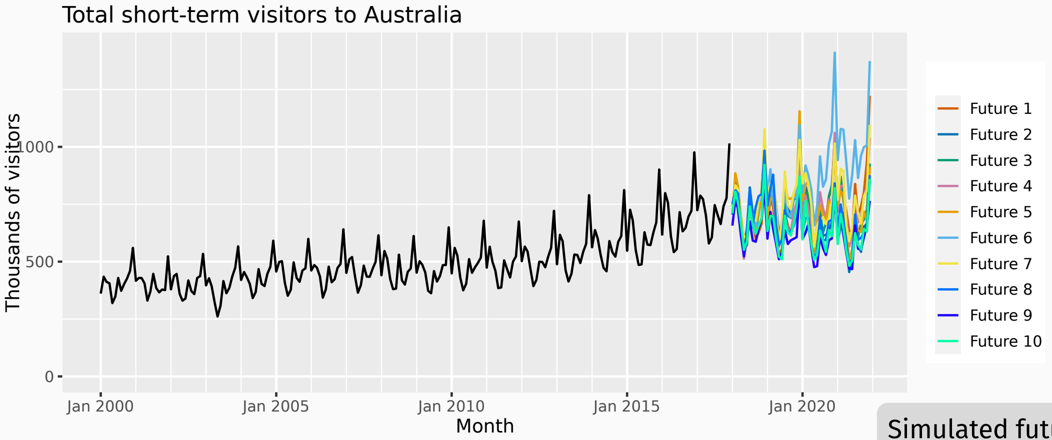
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS

What is a forecast?

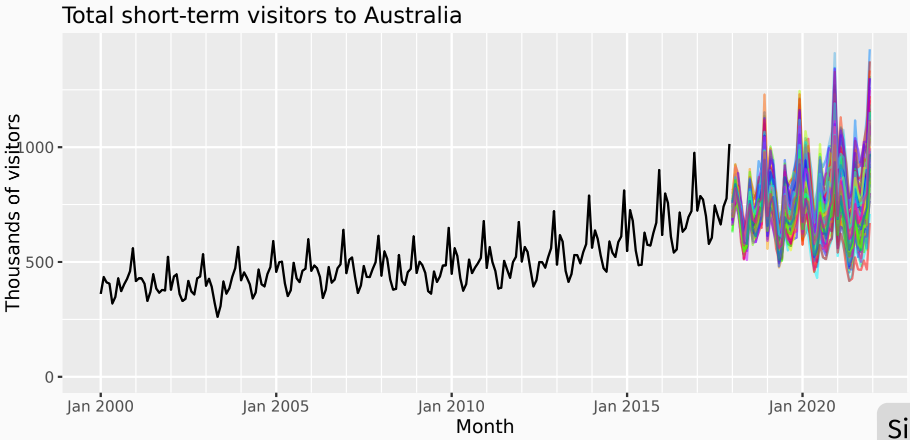
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS

What is a forecast?

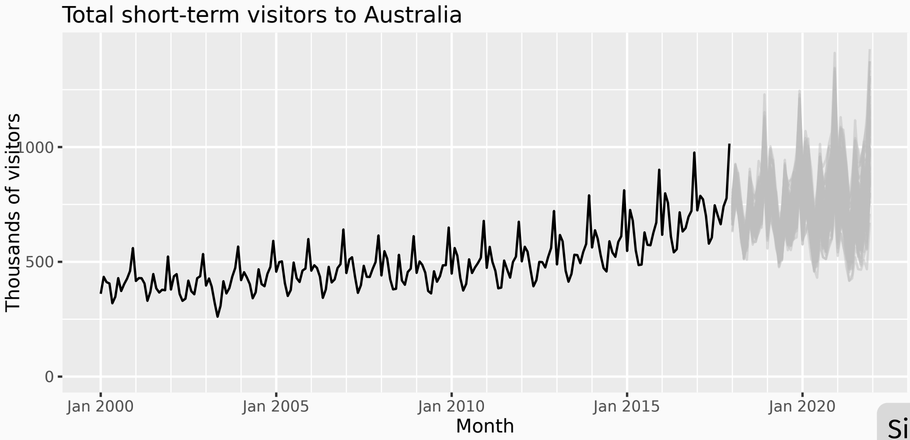
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS

What is a forecast?

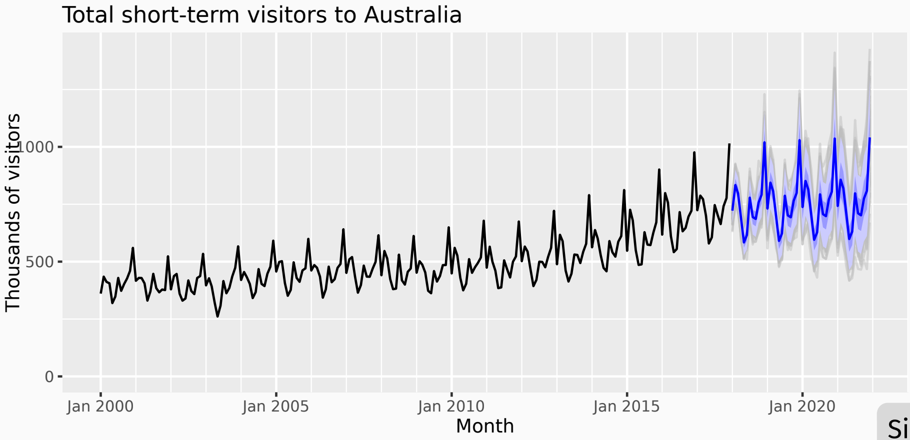
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS

What is a forecast?

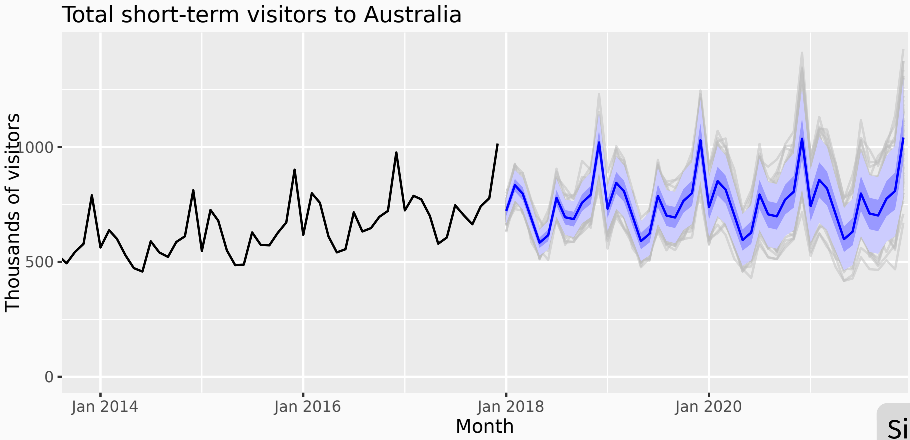
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS

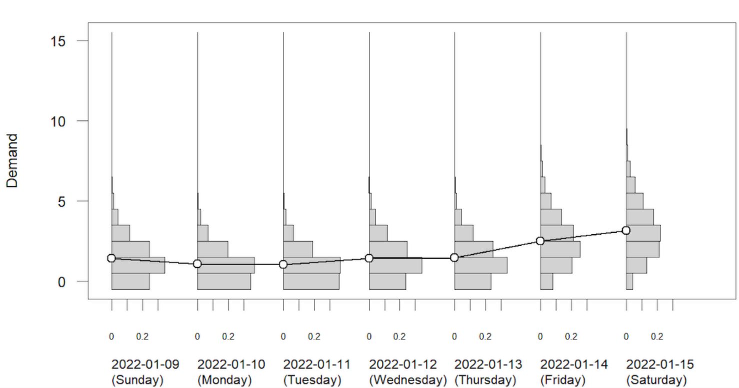
Prediction interval

A forecast is an estimate of the probability distribution of a variable to be observed in the future.

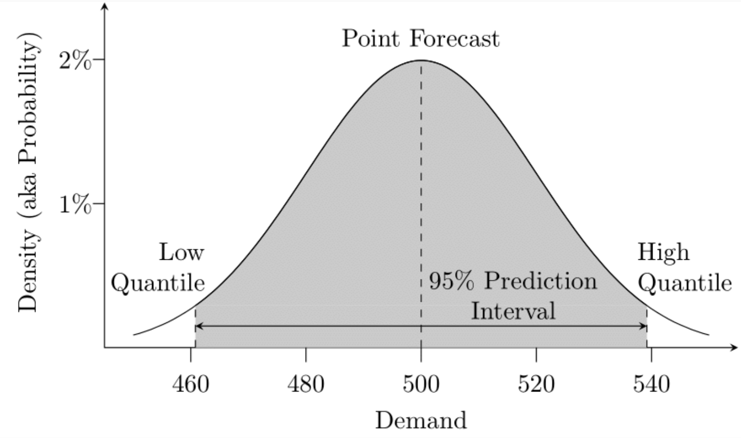


Simulated futures from an ETS

Visualising forecast distributions



Forecast distribution



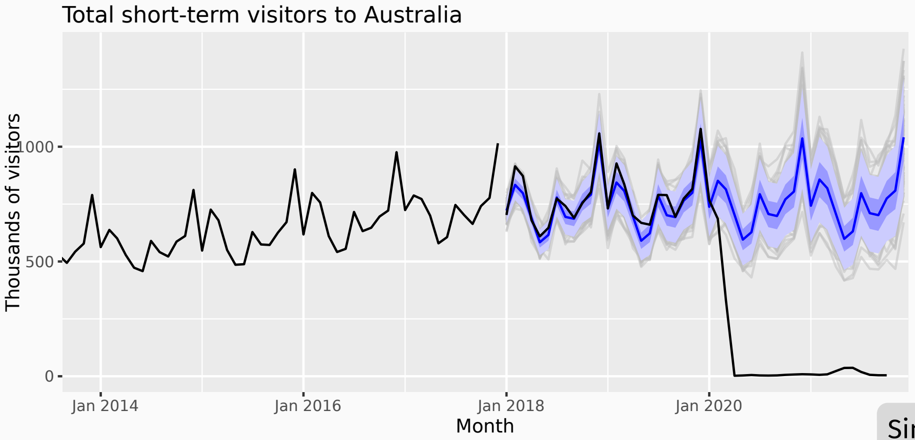
Statistical forecasting

- 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: $\hat{y}_{T+h|T} = \text{E}[y_{T+h} \mid y_1, \dots, y_T]$.
- 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

What can we forecast?

A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS

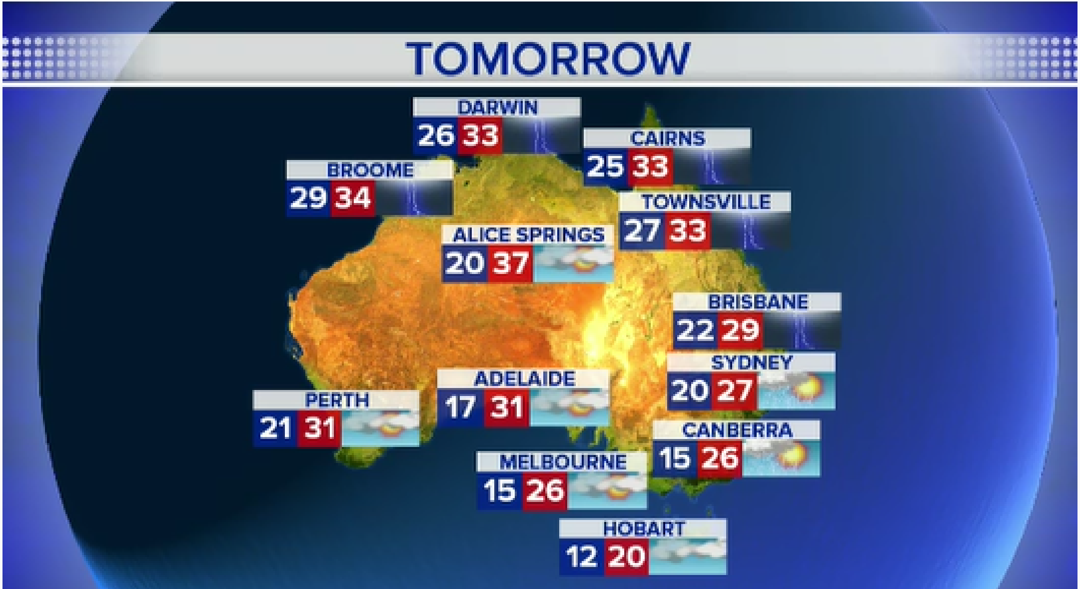
What can we forecast?

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?

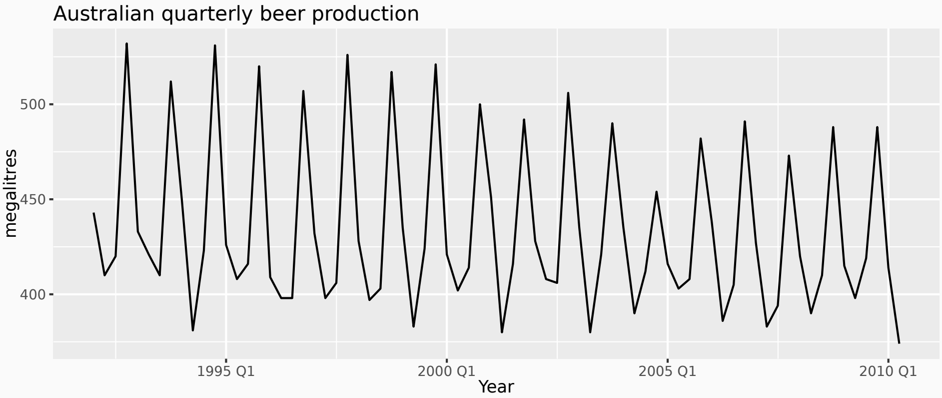
Factors affecting forecastability

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

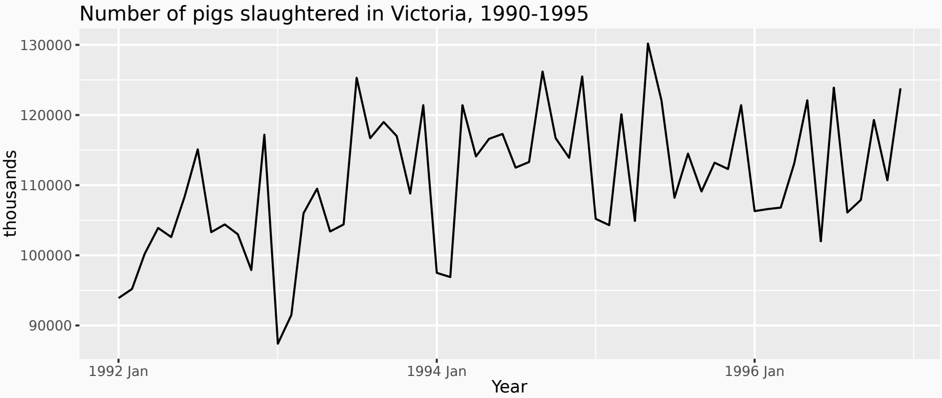
Outline

Some simple forecasting methods



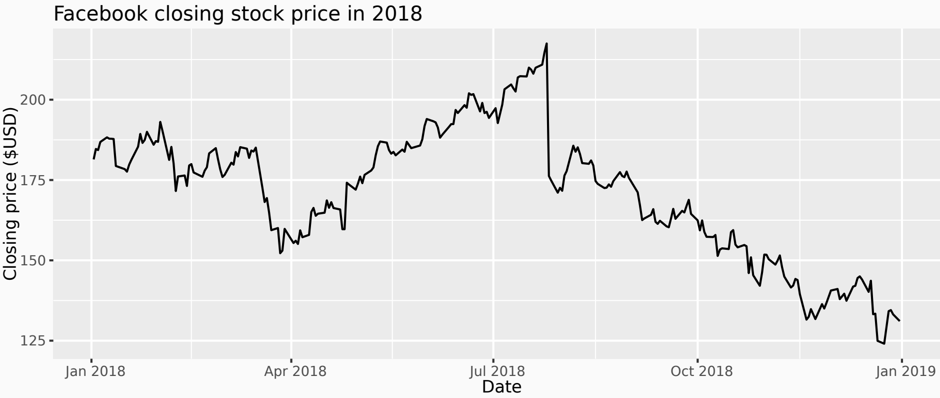
How would you forecast these series?

Some simple forecasting methods



How would you forecast these series?

Some simple forecasting methods

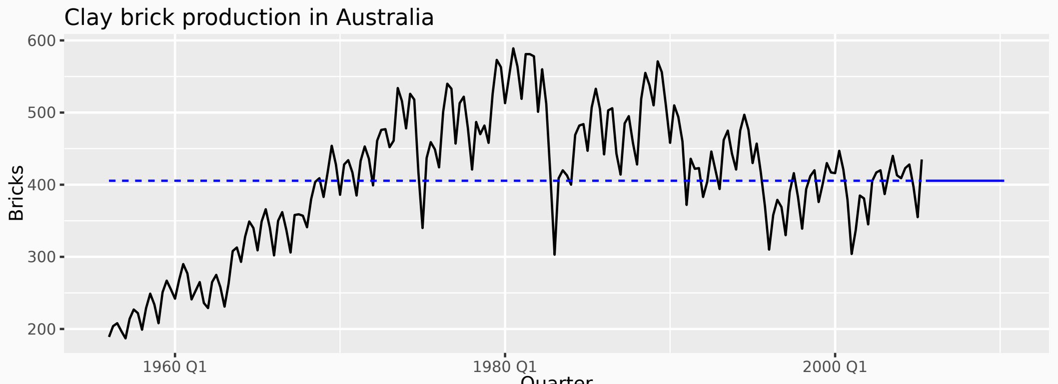


How would you forecast these series?

Some simple forecasting methods

`MEAN(y)`: Average method

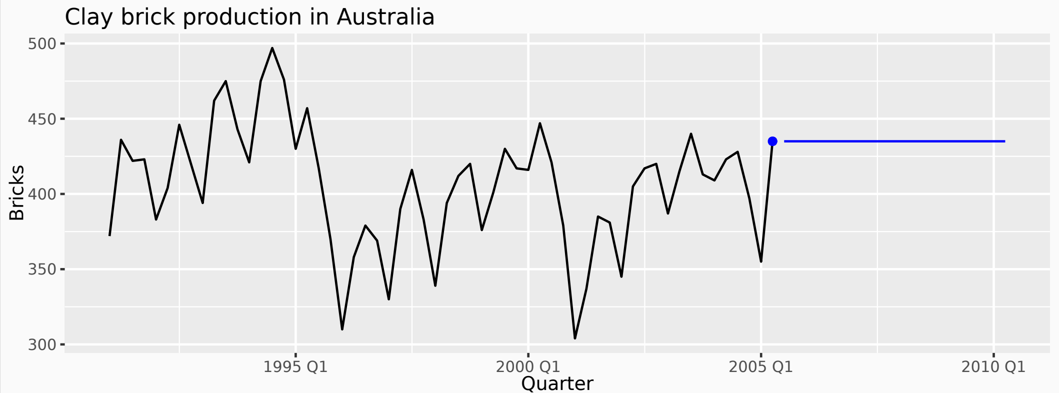
- Forecast of all future values is equal to mean of historical data $\{y_1, \dots, y_T\}$.
- Forecasts: $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T$



Some simple forecasting methods

`NAIVE(y)`: Naïve method

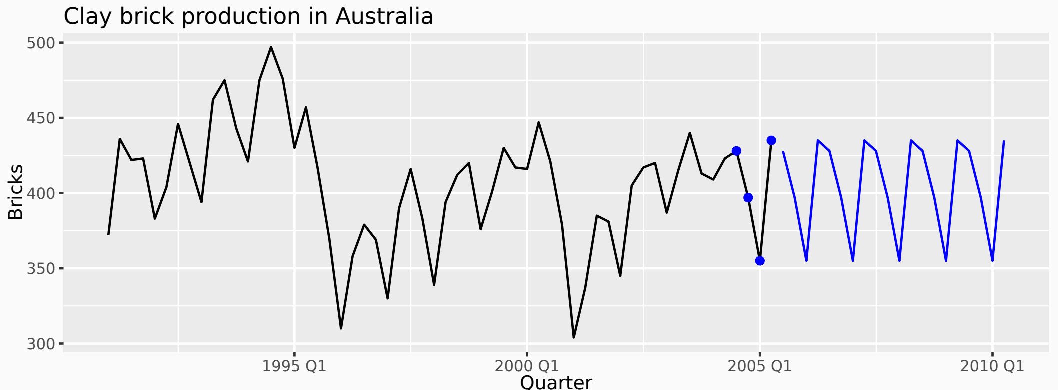
- Forecasts equal to last observed value.
- Forecasts: $\hat{y}_{T+h|T} = y_T$.
- Consequence of efficient market hypothesis.



Some simple forecasting methods

`SNAIVE(y ~ 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:

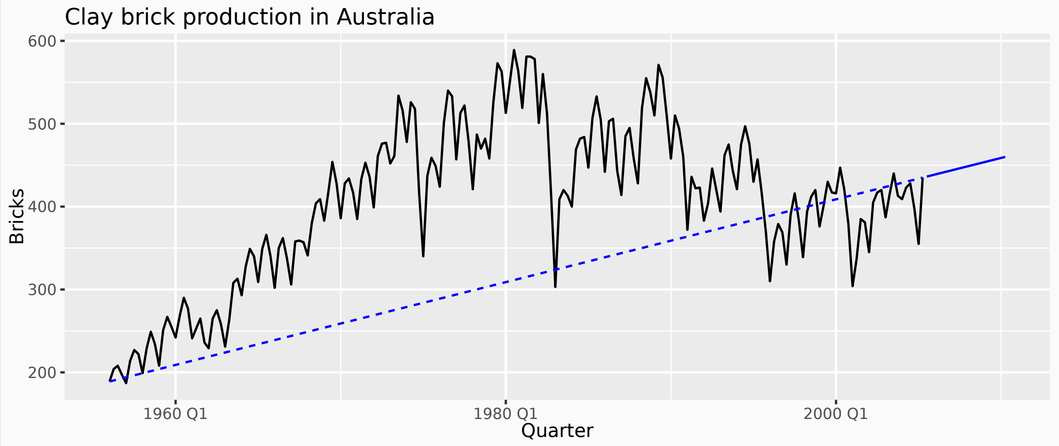
$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1})$$

$$= y_T + \frac{h}{T-1}(y_T - y_1).$$

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

Some simple forecasting methods

Drift method



Outline

Model specification

- Model specification in fable supports a formula based interface
- A model formula in R is expressed using `response ~ terms`
  - the formula's left side describes the response
  - the right describes terms used to model the response.
- `Attention`: MODEL name is in capital letters, e.g. `SNAIVE`

Model estimation

The `model()` function trains models on data. - It returns a `mable` object.

```
# Fit the models
my_mable <- my_data %>%
  model(
    choose_name1 = MODEL1(response ~ term1+...),
    choose_name2 = MODEL2(response ~ term1+...),
    choose_name3 = MODEL3(response ~ term1+...),
    choose_name4 = MODEL4(response ~ term1+...)
  )
```

Model fitting- example

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

```
beer_fit <- aus_production |>
  model(
    `Seasonal_naïve` = SNAIVE(Beer),
    `Naïve` = NAIVE(Beer),
    Drift = RW(Beer ~ drift()),
    Mean = MEAN(Beer)
  )
```

```
# 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.

Extract information from `mable`

```
beer_fit %>% select(snaive) %>% report()
beer_fit %>% tidy()
beer_fit %>% glance()
```

- The `report()` function gives a formatted model-specific display.
- The `tidy()` function is used to extract the coefficients from the models.
- We can extract information about some specific model using the `filter()` and `select()`functions.

Check model performance

Once a model has been fitted, it is important to check how well it has performed on the data. I come back to this latter.

Outline

Producing forecasts

- The `forecast()` function is used to produce forecasts from estimated models.
- **h** can be specified with:
  - a number (the number of future observations)
  - natural language (the length of time to predict)
  - provide a dataset of future time periods

Producing forecasts

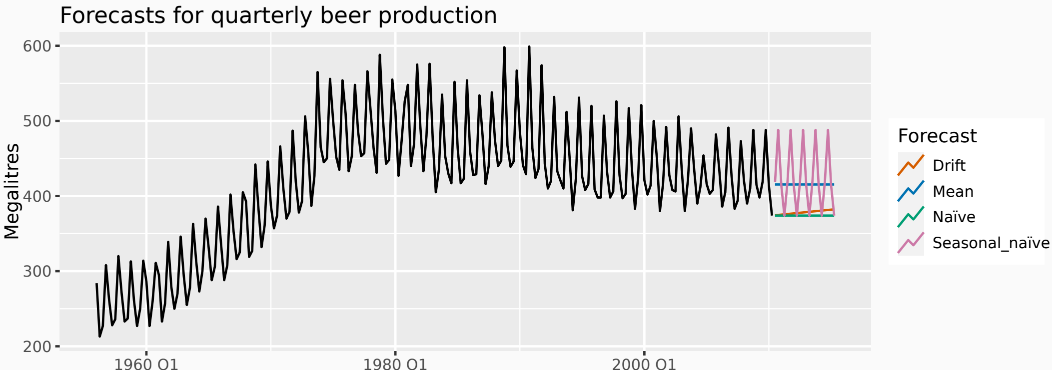
```r
beer_fc <- beer_fit |>
  forecast(h = "5 years")
```

```
# A fable: 80 x 4 [1Q]
# Key:     .model [4]
  .model         Quarter        Beer .mean
  <chr>            <qtr>      <dist> <dbl>
1 Seasonal_naïve 2010 Q3 N(419, 373)   419
2 Seasonal_naïve 2010 Q4 N(488, 373)   488
3 Seasonal_naïve 2011 Q1 N(414, 373)   414
4 Seasonal_naïve 2011 Q2 N(374, 373)   374
# i 76 more rows
```

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

Visualising forecasts

```
beer_fc |>
  autoplot(aus_production, level = NULL) +
  labs(title = "Forecasts for quarterly beer production",
       x = "Quarter", y = "Megalitres") +
  guides(colour = guide_legend(title = "Forecast"))
```



Forecast distributions

- A forecast $\hat{y}_{T+h|T}$ is (usually) the mean of the conditional distribution $y_{T+h} \mid y_1, \dots, y_T$.
- Most time series models produce normally distributed forecasts.
- The forecast distribution describes the probability of observing any future value.

Forecast distributions - normal distribution

Assuming residuals are normal, uncorrelated, sd = $\hat{\sigma}$:

Mean:	$\hat{y}_{T+h\|T} \sim N(\bar{y}, (1 + 1/T)\hat{\sigma}^2)$
Naïve:	$\hat{y}_{T+h\|T} \sim N(y_T, h\hat{\sigma}^2)$
Seasonal naïve:	$\hat{y}_{T+h\|T} \sim N(y_{T+h-m(k+1)}, (k + 1)\hat{\sigma}^2)$
Drift:	$\hat{y}_{T+h\|T} \sim N(y_T + \frac{h}{T-1}(y_T - y_1), h\frac{T+h}{T}\hat{\sigma}^2)$

where k is the integer part of $(h - 1)/m$.

Note that when $h = 1$ and T is large, these all give the same approximate forecast variance: $\hat{\sigma}^2$.

Forecast distributions from bootstrapping

When a normal distribution for the residuals is an unreasonable assumption, one alternative is to use bootstrapping, which only assumes that the residuals are uncorrelated with constant variance.

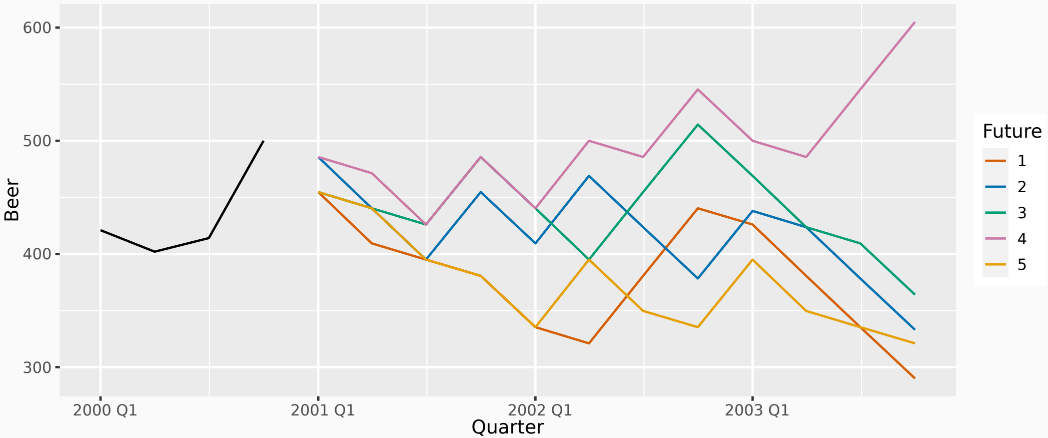
- A one-step forecast error is defined as $e_t = y_t - \hat{y}_{t|t-1}$, $y_t = \hat{y}_{t|t-1} + e_t$.
- So we can simulate the next observation of a time series using $y_{T+1} = \hat{y}_{T+1|T} + e_{T+1}$
- Adding the new simulated observation to our data set, we can repeat the process to obtain $y_{T+2} = \hat{y}_{T+2|T+1} + e_{T+2}$

Generate many possible future using generate()

```r
beer_2000 <- aus_production |> filter(year(Quarter) == 2000) |> select(Beer)
fit <- beer_2000 |>
  model(NAIVE(Beer))
sim <- fit |> generate(h = 12, times = 5, bootstrap = TRUE)
sim
```

```
# A tsibble: 60 x 5 [1Q]
# Key:       .model, .rep [5]
   .model      .rep  Quarter .innov  .sim
   <chr>       <chr>   <qtr>  <dbl> <dbl>
 1 NAIVE(Beer) 1     2001 Q1  -45.3  455.
 2 NAIVE(Beer) 1     2001 Q2  -45.3  409.
 3 NAIVE(Beer) 1     2001 Q3  -14.3  395
 4 NAIVE(Beer) 1     2001 Q4  -14.3  381.
 5 NAIVE(Beer) 1     2002 Q1  -45.3  335.
```

Generate 5 different futures



casts

```r
fc <- fit |> forecast(h = 12, bootstrap = TRUE)
fc
```

```
# A fable: 12 x 4 [1Q]
# Key:     .model [1]
   .model      Quarter         Beer .mean
   <chr>         <qtr>       <dist> <dbl>
 1 NAIVE(Beer) 2001 Q1 sample[5000]  500.
 2 NAIVE(Beer) 2001 Q2 sample[5000]  500.
 3 NAIVE(Beer) 2001 Q3 sample[5000]  500.
 4 NAIVE(Beer) 2001 Q4 sample[5000]  499.
 5 NAIVE(Beer) 2002 Q1 sample[5000]  500.
 6 NAIVE(Beer) 2002 Q2 sample[5000]  500.
 7 NAIVE(Beer) 2002 Q3 sample[5000]  500.
 8 NAIVE(Beer) 2002 Q4 sample[5000]  500.
```

Prediction intervals

- Forecast intervals can be extracted using the `hilo()` function.

```r
fit <- aus_production |> select(Beer) %>% model(NAIVE(Beer))
forecast(fit) %>% hilo(level = 95) %>% unpack_hilo("95%")
```

```
# A tsibble: 8 x 6 [1Q]
# Key:       .model [1]
  .model      Quarter         Beer .mean `95%_lower` `95%_upper`
  <chr>         <qtr>       <dist> <dbl>       <dbl>       <dbl>
1 NAIVE(Beer) 2010 Q3  N(374, 4580)   374       241.        507.
2 NAIVE(Beer) 2010 Q4  N(374, 9159)   374       186.        562.
3 NAIVE(Beer) 2011 Q1 N(374, 13739)   374       144.        604.
4 NAIVE(Beer) 2011 Q2 N(374, 18319)   374       109.        639.
5 NAIVE(Beer) 2011 Q3 N(374, 22898)   374        77.4       671.
```

Prediction intervals

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

```
# A tsibble: 80 x 6 [1Q]
# Key:       .model [4]
   .model         Quarter         Beer .mean         `50%`         `75%`
   <chr>            <qtr>       <dist> <dbl>        <hilo>        <hilo>
 1 Seasonal_naïve 2010 Q3  N(419, 373)   419 [406, 432]50 [397, 441]75
 2 Seasonal_naïve 2010 Q4  N(488, 373)   488 [475, 501]50 [466, 510]75
 3 Seasonal_naïve 2011 Q1  N(414, 373)   414 [401, 427]50 [392, 436]75
 4 Seasonal_naïve 2011 Q2  N(374, 373)   374 [361, 387]50 [352, 396]75
 5 Seasonal_naïve 2011 Q3  N(419, 747)   419 [401, 437]50 [388, 450]75
 6 Seasonal_naïve 2011 Q4  N(488, 747)   488 [470, 506]50 [457, 519]75
 7 Seasonal_naïve 2012 Q1  N(414, 747)   414 [396, 432]50 [383, 445]75
 8 Seasonal_naïve 2012 Q2  N(374, 747)   374 [356, 392]50 [343, 405]75
 9 Seasonal_naïve 2012 Q3 N(419, 1120)   419 [396, 442]50 [380, 458]75
```

Prediction intervals

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

```
# A tsibble: 80 x 8 [1Q]
# Key:       .model [4]
   .model          Quarter         Beer .mean          `50%`          `75%` lower upper
   <chr>             <qtr>       <dist> <dbl>         <hilo>         <hilo> <dbl> <dbl>
 1 Seasonal_naïve 2010 Q3  N(419, 373)   419 [406, 432]50 [397, 441]75  406.  432.
 2 Seasonal_naïve 2010 Q4  N(488, 373)   488 [475, 501]50 [466, 510]75  475.  501.
 3 Seasonal_naïve 2011 Q1  N(414, 373)   414 [401, 427]50 [392, 436]75  401.  427.
 4 Seasonal_naïve 2011 Q2  N(374, 373)   374 [361, 387]50 [352, 396]75  361.  387.
 5 Seasonal_naïve 2011 Q3  N(419, 747)   419 [401, 437]50 [388, 450]75  401.  437.
 6 Seasonal_naïve 2011 Q4  N(488, 747)   488 [470, 506]50 [457, 519]75  470.  506.
 7 Seasonal_naïve 2012 Q1  N(414, 747)   414 [396, 432]50 [383, 445]75  396.  432.
 8 Seasonal_naïve 2012 Q2  N(374, 747)   374 [356, 392]50 [343, 405]75  356.  392.
```

Outline

Fitted values

- $\hat{y}_{t|t-1}$ is the forecast of y_t based on observations $y_1, \dots, y_t$.
- 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: $e_t = y_t - \hat{y}_{t|t-1}$.

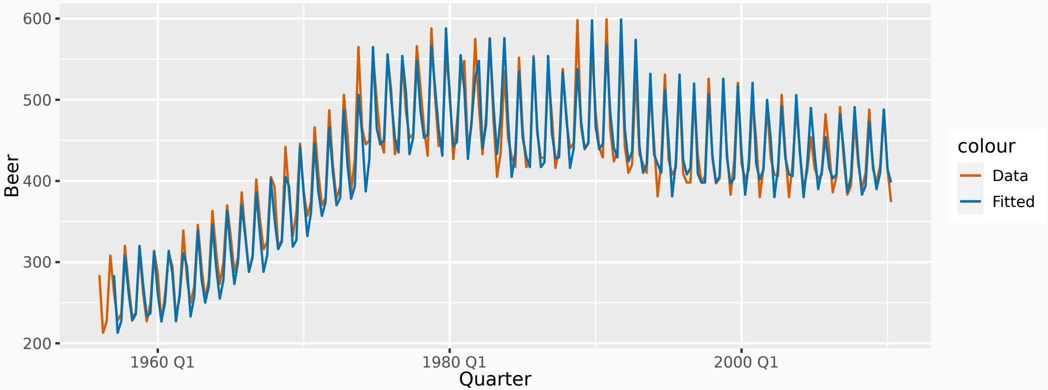
Beer production - augment

```r
fit <- aus_production |> select(Beer) %>% model(SNAIVE(Beer))
augment(fit)
```

```
# A tsibble: 218 x 6 [1Q]
# Key:       .model [1]
   .model       Quarter  Beer .fitted .resid .innov
   <chr>          <qtr> <dbl>   <dbl>  <dbl>  <dbl>
 1 SNAIVE(Beer) 1956 Q1   284      NA     NA     NA
 2 SNAIVE(Beer) 1956 Q2   213      NA     NA     NA
 3 SNAIVE(Beer) 1956 Q3   227      NA     NA     NA
 4 SNAIVE(Beer) 1956 Q4   308      NA     NA     NA
 5 SNAIVE(Beer) 1957 Q1   262     284    -22    -22
 6 SNAIVE(Beer) 1957 Q2   228     213     15     15
 7 SNAIVE(Beer) 1957 Q3   236     227      9      9
 8 SNAIVE(Beer) 1957 Q4   320     308     12     12
 9 SNAIVE(Beer) 1958 Q1   272     262     10     10
10 SNAIVE(Beer) 1958 Q2   233     228      5      5
# i 208 more rows
```

Beer production - fitted values

```
augment(fit) |>
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Beer, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted"))
```



Beer production - residuals

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