Feasts and fables: Time series analysis using R

Rob J Hyndman

robjhyndman.com/rba2023

Outline

1. tsibble: Time series data
2. feasts: Data visualization
3. feasts: Time series features
4. fable: Forecasting
5. fable: Evaluating forecast accuracy
6. fable: Forecast reconciliation

Outline

Tidyverts packages

tidyverts.org

tsibble
tsibbledata
feasts
fable

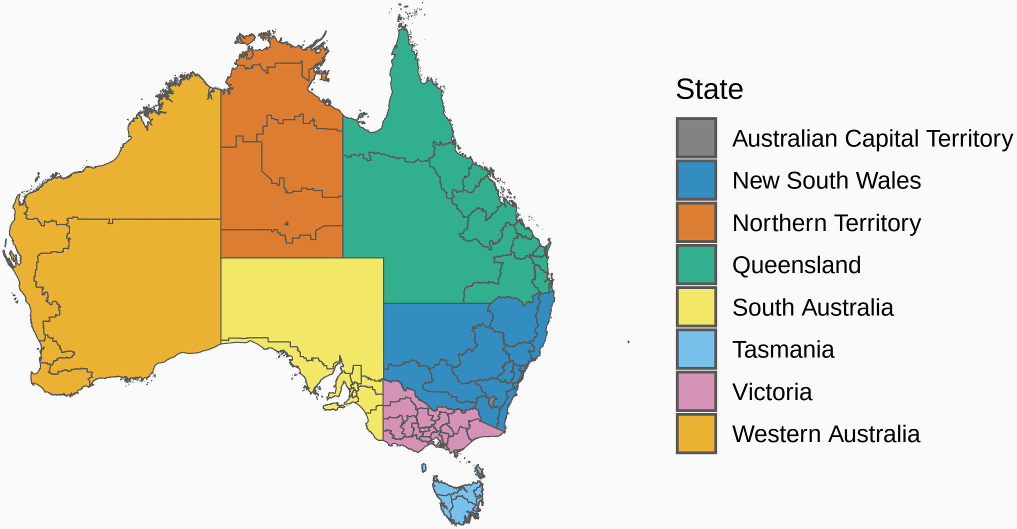
Tidyverts packages

tidyverts.org

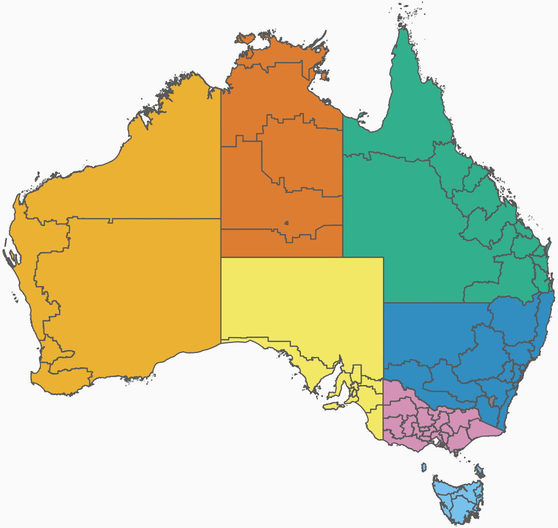
```r
library(fpp3)
```

```
-- Attaching packages -------------- fpp3 0.5.0 --
✓ tibble      3.1.8     ✓ tsibble     1.1.2
✓ dplyr       1.1.0     ✓ tsibbledata 0.4.1
✓ tidyr       1.3.0     ✓ feasts      0.3.0
✓ lubridate   1.9.2     ✓ fable       0.3.2
✓ ggplot2     3.4.1     ✓ fabletools  0.3.2
-- Conflicts ------------------- fpp3_conflicts --
X lubridate::date()     masks base::date()
X dplyr::filter()       masks stats::filter()
X tsibble::intersect()  masks base::intersect()
X tsibble::interval()   masks lubridate::interval()
X dplyr::lag()          masks stats::lag()
```

Australian tourism regions



Australian tourism regions



- Quarterly data on visitor nights: 1998 – 2017
- From *National Visitor Survey*, interviews of 120,000 Australians aged 15+.
- Geographical hierarchy split by
  - 8 states and territories
  - 76 regions
- Purpose:
  - Holidays
  - Business
  - Visiting friends & relatives
  - Other

tsibble objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##      <qtr> <chr>    <fct> <chr>    <dbl>
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
```

Domestic visitor nights in thousands by state/region and purpose.

`tsibble` objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##      Index <chr>    <fct> <chr>    <dbl>
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
```

Domestic visitor nights in thousands by state/region and purpose.

tsibble objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##     Index  Keys                    <dbl>
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
```

Domestic visitor nights in thousands by state/region and purpose.

tsibble objects

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State Purpose  Trips
##    Index   Keys                    Measure
##  1 1998 Q1 Adelaide SA    Business  135.
##  2 1998 Q2 Adelaide SA    Business  110.
##  3 1998 Q3 Adelaide SA    Business  166.
##  4 1998 Q4 Adelaide SA    Business  127.
##  5 1999 Q1 Adelaide SA    Business  137.
##  6 1999 Q2 Adelaide SA    Business  200.
##  7 1999 Q3 Adelaide SA    Business  169.
##  8 1999 Q4 Adelaide SA    Business  134.
##  9 2000 Q1 Adelaide SA    Business  154.
## 10 2000 Q2 Adelaide SA    Business  169.
```

Domestic visitor nights in thousands by state/region and purpose.

`tsibble` objects

- A `tsibble` allows storage and manipulation of multiple time series in R.
- It contains:
  - An index: time information about the observation
  - Measured variable(s): numbers of interest
  - Key variable(s): optional unique identifiers for each series
- It works with tidyverse functions.

Outline

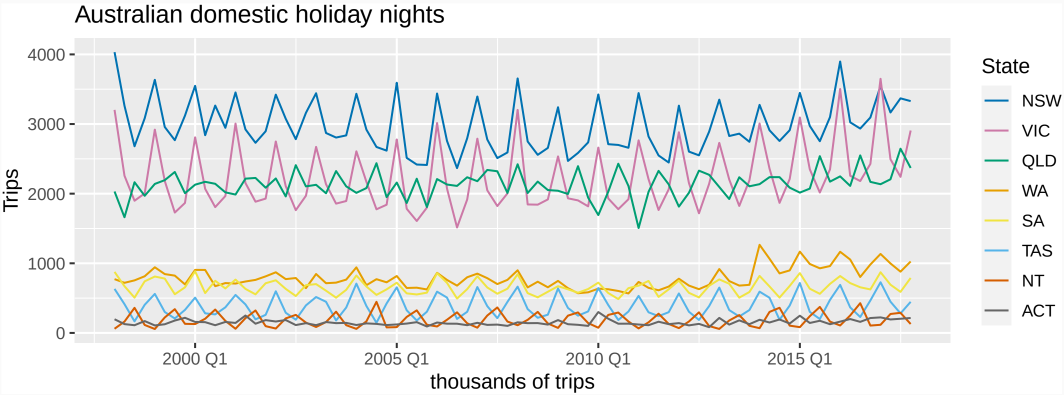
Australian holidays

```r
holidays <- tourism |>
  filter(Purpose == "Holiday") |>
  group_by(State) |>
  summarise(Trips = sum(Trips))
```

```
## # A tsibble: 640 x 3 [1Q]
## # Key:       State [8]
##    State Quarter Trips
##    <fct>   <qtr> <dbl>
##  1 NSW   1998 Q1 4033.
##  2 NSW   1998 Q2 3262.
##  3 NSW   1998 Q3 2681.
##  4 NSW   1998 Q4 3083.
##  5 NSW   1999 Q1 3635.
##  6 NSW   1999 Q2 2958.
##  7 NSW   1999 Q3 2768.
##  8 NSW   1999 Q4 3121.
##  9 NSW   2000 Q1 3548.
```

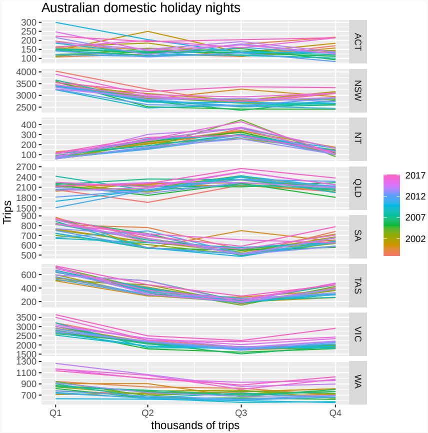
Australian holidays

```r
holidays |> autoplot(Trips) +
  labs(x = "thousands of trips", title = "Australian domestic holiday nights")
```



Seasonal plots

```r
holidays |> gg_season(Trips) +
  labs(x = "thousands of trips", title = "Australian domestic holiday nights")
```



Seasonal subseries plots

```r
holidays |> gg_subseries(Trips) +
  labs(x = "thousands of trips", title = "Australian domestic holiday nights")
```

Outline

Strength of seasonality and trend

STL decomposition

$$y_t = T_t + S_t + R_t$$

Seasonal strength

$$\max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t + R_t)}\right)$$

Trend strength

$$\max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(T_t + R_t)}\right)$$

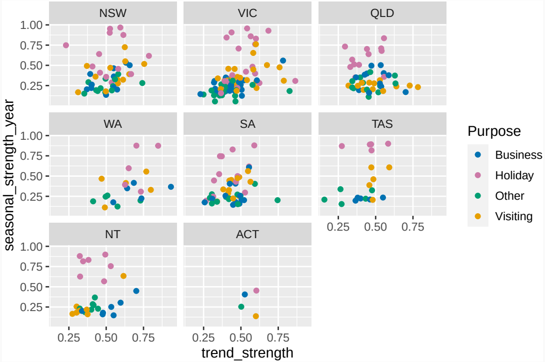
Feature extraction and statistics

```r
tourism |> features(Trips, feat_stl)
```

```
## # A tibble: 304 x 12
##    Region     State Purpose trend~1 seaso~2 seaso~3 seaso~4 spiki~5 linea~6
##    <chr>      <fct> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Adelaide   SA    Busine~   0.464   0.407       3       1 1.58e+2  -5.31
##  2 Adelaide   SA    Holiday   0.554   0.619       1       2 9.17e+0  49.0
##  3 Adelaide   SA    Other     0.746   0.202       2       1 2.10e+0  95.1
##  4 Adelaide   SA    Visiti~   0.435   0.452       1       3 5.61e+1  34.6
##  5 Adelaide ~ SA    Busine~   0.464   0.179       3       0 1.03e-1   0.968
##  6 Adelaide ~ SA    Holiday   0.528   0.296       2       1 1.77e-1  10.5
##  7 Adelaide ~ SA    Other     0.593   0.404       2       2 4.44e-4   4.28
##  8 Adelaide ~ SA    Visiti~   0.488   0.254       0       3 6.50e+0  34.2
##  9 Alice Spr~ NT    Busine~   0.534   0.251       0       1 1.69e-1  23.8
## 10 Alice Spr~ NT    Holiday   0.381   0.832       3       1 7.39e-1 -19.6
## # ... with 294 more rows, 3 more variables: curvature <dbl>,
## #   stl_e_acf1 <dbl>, stl_e_acf10 <dbl>, and abbreviated variable names
## #   1: trend_strength, 2: seasonal_strength_year, 3: seasonal_peak_year,
## #   4: seasonal_trough_year, 5: spikiness, 6: linearity
```

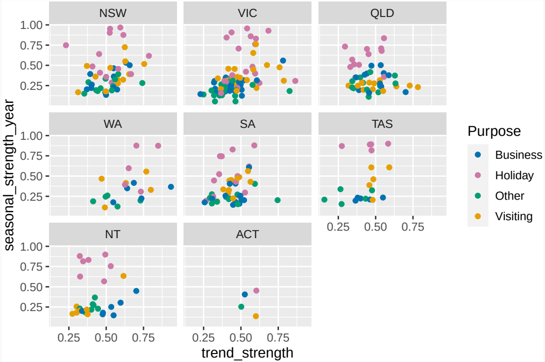
Feature extraction and statistics

```r
tourism |>
  features(Trips, feat_stl) |>
  ggplot(aes(x = trend_strength, y = seasonal_strength_year, col = Purpose)) +
  geom_point() +
  facet_wrap(vars(State))
```



Feature extraction and statistics

```r
tourism |>
  features(Trips, feat_stl) |>
  ggplot(aes(x = trend_strength, y = seasonal_strength_year, col = Purpose)) +
  geom_point() +
  facet_wrap(vars(State))
```



- Holidays more seasonal than other travel.
- WA has strongest trends.

Feature extraction and statistics

Find the most seasonal time series:

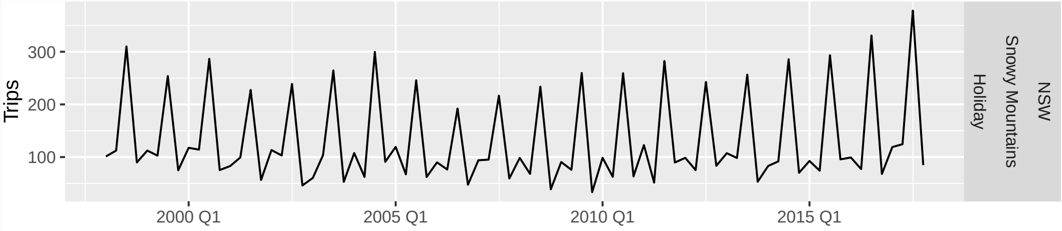
```
most_seasonal <- tourism |>
  features(Trips, feat_stl) |>
  filter(seasonal_strength_year == max(seasonal_strength_year))
```

Feature extraction and statistics

Find the most seasonal time series:

```
most_seasonal <- tourism |>
  features(Trips, feat_stl) |>
  filter(seasonal_strength_year == max(seasonal_strength_year))
```

```
tourism |>
  right_join(most_seasonal, by = c("State", "Region", "Purpose")) |>
  ggplot(aes(x = Quarter, y = Trips)) +
  geom_line() +
  facet_grid(vars(State, Region, Purpose))
```



Feature extraction and statistics

```r
tourism_features <- tourism |>
  features(Trips, feature_set(pkgs = "feasts"))
```

All features from the feasts package

```
## # A tibble: 304 x 51
##    Region     State Purpose trend~1 seaso~2 seaso~3 seaso~4 spiki~5 linea~6
##    <chr>      <fct> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Adelaide   SA    Busine~   0.464   0.407       3       1 1.58e+2  -5.31
##  2 Adelaide   SA    Holiday   0.554   0.619       1       2 9.17e+0  49.0
##  3 Adelaide   SA    Other     0.746   0.202       2       1 2.10e+0  95.1
##  4 Adelaide   SA    Visiti~   0.435   0.452       1       3 5.61e+1  34.6
##  5 Adelaide ~ SA    Busine~   0.464   0.179       3       0 1.03e-1   0.968
##  6 Adelaide ~ SA    Holiday   0.528   0.296       2       1 1.77e-1  10.5
##  7 Adelaide ~ SA    Other     0.593   0.404       2       2 4.44e-4   4.28
##  8 Adelaide ~ SA    Visiti~   0.488   0.254       0       3 6.50e+0  34.2
##  9 Alice Spr~ NT    Busine~   0.534   0.251       0       1 1.69e-1  23.8
## 10 Alice Spr~ NT    Holiday   0.381   0.832       3       1 7.39e-1 -19.6
## # ... with 294 more rows, 42 more variables: curvature <dbl>,
## #   stl_e_acf1 <dbl>, stl_e_acf10 <dbl>, acf1 <dbl>, acf10 <dbl>,
## #   diff1_acf1 <dbl>, diff1_acf10 <dbl>, diff2_acf1 <dbl>,
## #   diff2_acf10 <dbl>, season_acf1 <dbl>, pacf5 <dbl>, diff1_pacf5 <dbl>,
```

Feature extraction and statistics

```r
pcs <- tourism_features |>
  select(-State, -Region, -Purpose) |>
  prcomp(scale = TRUE) |>
  broom::augment(tourism_features)
```

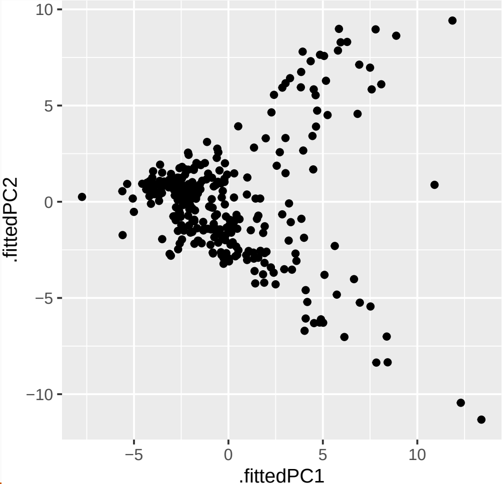
Principal components based on all features from the feasts package

```
## # A tibble: 304 x 100
##    .rownames Region   State Purpose trend~1 seaso~2 seaso~3 seaso~4 spiki~5
##    <chr>     <chr>    <fct> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 1         Adelaide SA    Busine~   0.464   0.407       3       1 1.58e+2
##  2 2         Adelaide SA    Holiday   0.554   0.619       1       2 9.17e+0
##  3 3         Adelaide SA    Other     0.746   0.202       2       1 2.10e+0
##  4 4         Adelaide SA    Visiti~   0.435   0.452       1       3 5.61e+1
##  5 5         Adelaid~ SA    Busine~   0.464   0.179       3       0 1.03e-1
##  6 6         Adelaid~ SA    Holiday   0.528   0.296       2       1 1.77e-1
##  7 7         Adelaid~ SA    Other     0.593   0.404       2       2 4.44e-4
##  8 8         Adelaid~ SA    Visiti~   0.488   0.254       0       3 6.50e+0
##  9 9         Alice S~ NT    Busine~   0.534   0.251       0       1 1.69e-1
## 10 10        Alice S~ NT    Holiday   0.381   0.832       3       1 7.39e-1
## # ... with 294 more rows, 91 more variables: linearity <dbl>,
## #   curvature <dbl>, stl_e_acf1 <dbl>, stl_e_acf10 <dbl>, acf1 <dbl>,
## #   acf10 <dbl>, diff1_acf1 <dbl>, diff1_acf10 <dbl>, diff2_acf1 <dbl>,
## #   diff2_acf10 <dbl>, season_acf1 <dbl>, pacf5 <dbl>, diff1_pacf5 <dbl>,
```

Feature extraction and statistics

```r
pcs |> ggplot(aes(x=.fittedPC1, y=.fittedPC2)) +
  geom_point() + theme(aspect.ratio=1)
```

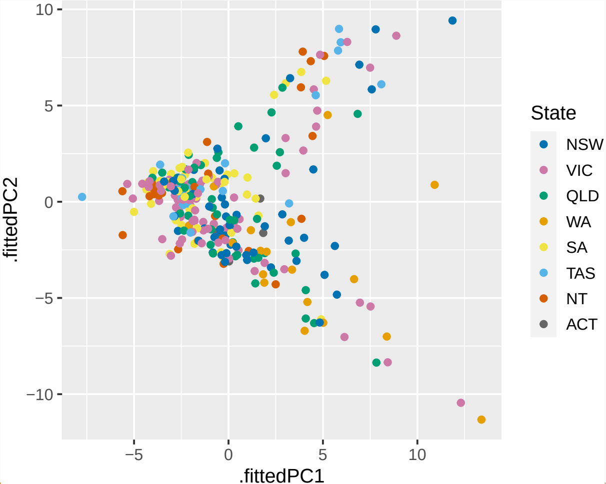
Principal components based on all features from the feasts package



Feature extraction and statistics

```r
pcs |> ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=State)) +
  geom_point() + theme(aspect.ratio=1)
```

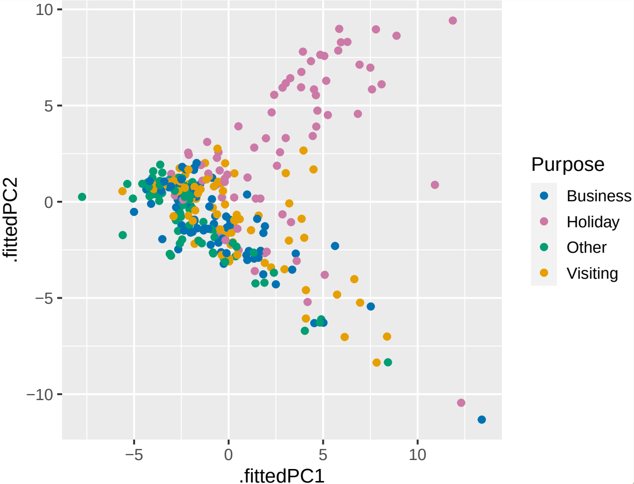
Principal components based on all features from the feasts package



Feature extraction and statistics

```
pcs |> ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=Purpose)) +
  geom_point() + theme(aspect.ratio=1)
```

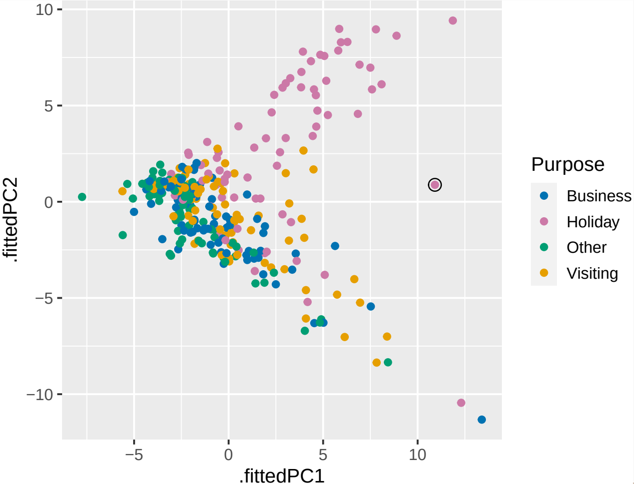
Principal components based on all features from the feasts package



Feature extraction and statistics

```r
pcs |> ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=Purpose)) +
  geom_point() + theme(aspect.ratio=1)
```

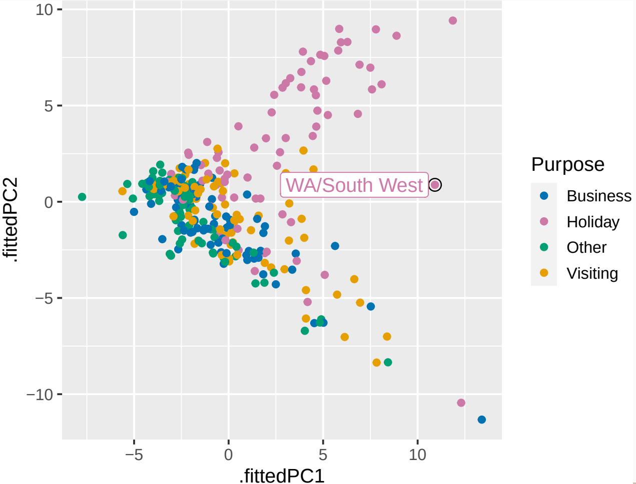
Principal components based on all features from the feasts package



Feature extraction and statistics

```
pcs |> ggplot(aes(x=.fittedPC1, y=.fittedPC2, col=Purpose)) +
  geom_point() + theme(aspect.ratio=1)
```

Principal components based on all features from the feasts package



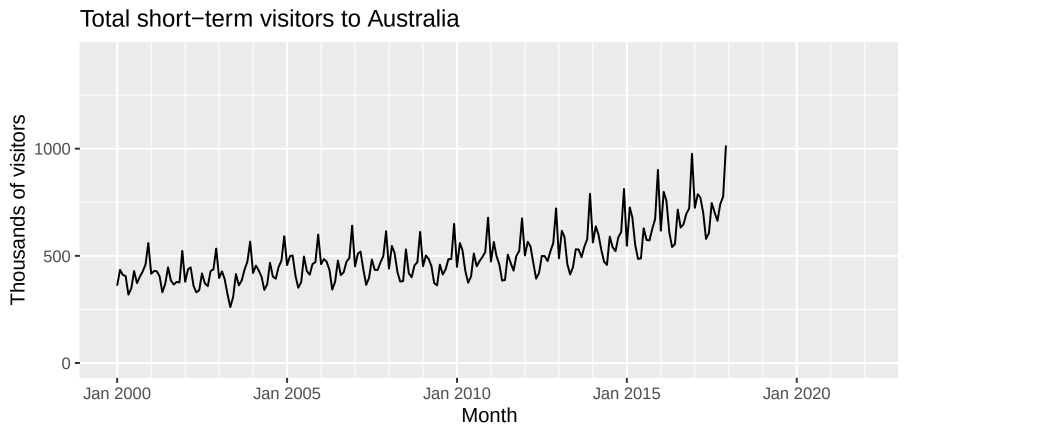
Outline

Random futures

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

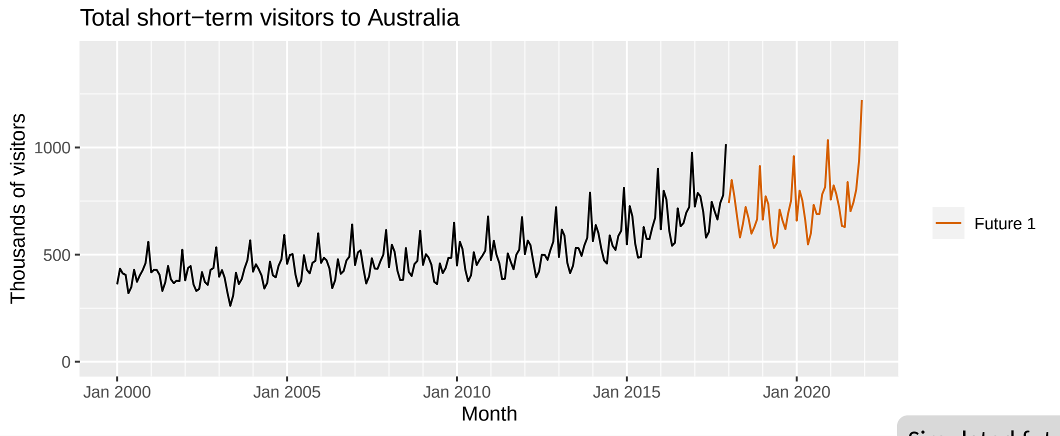
Random futures

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



Random futures

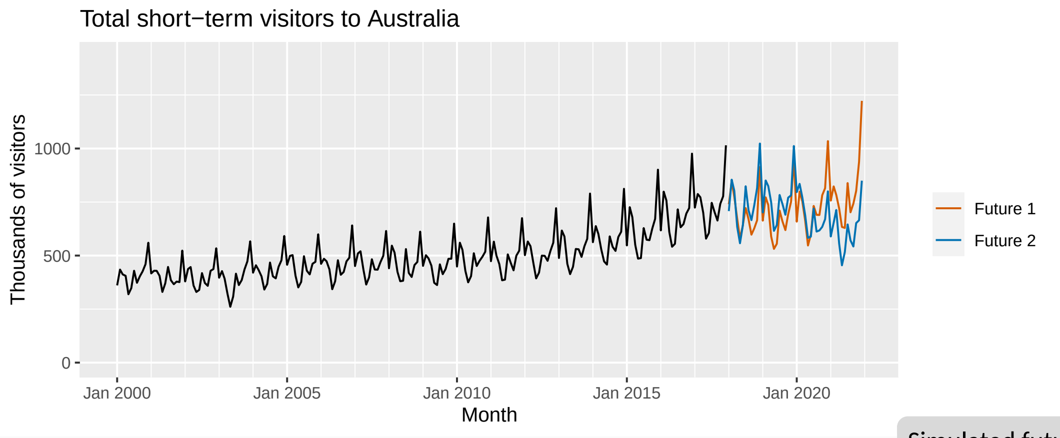
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

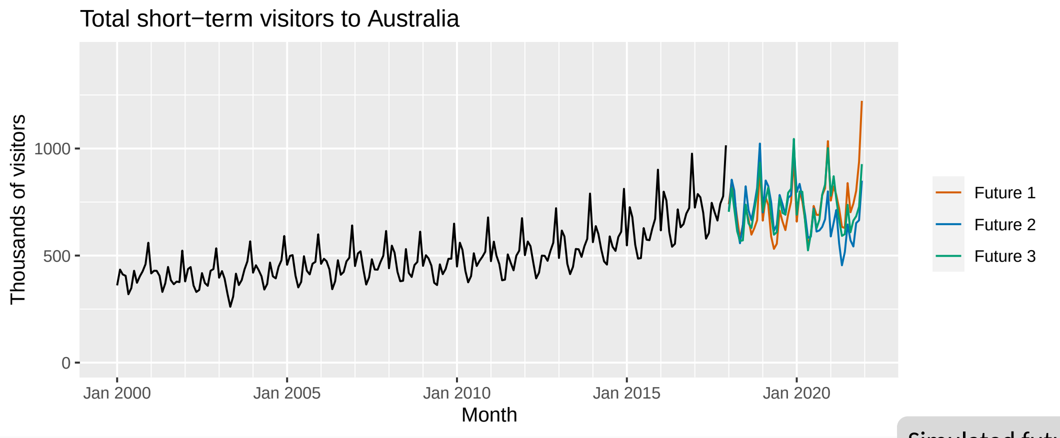
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

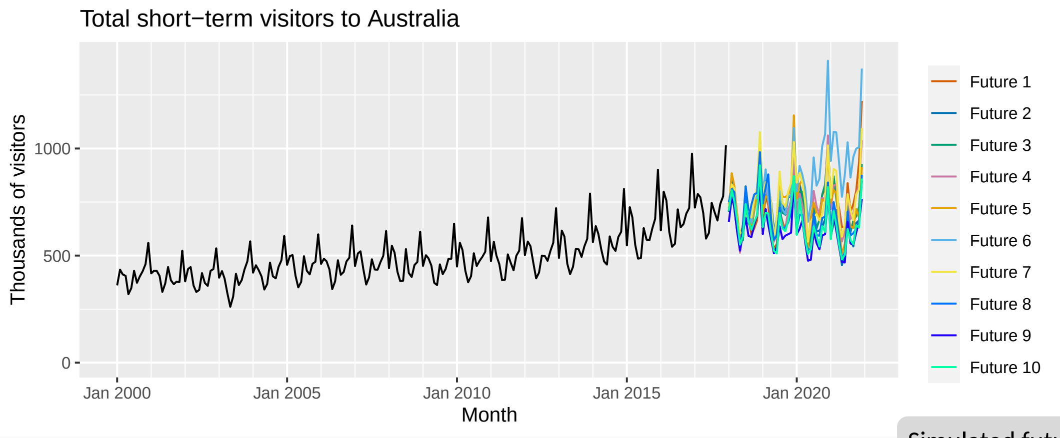
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

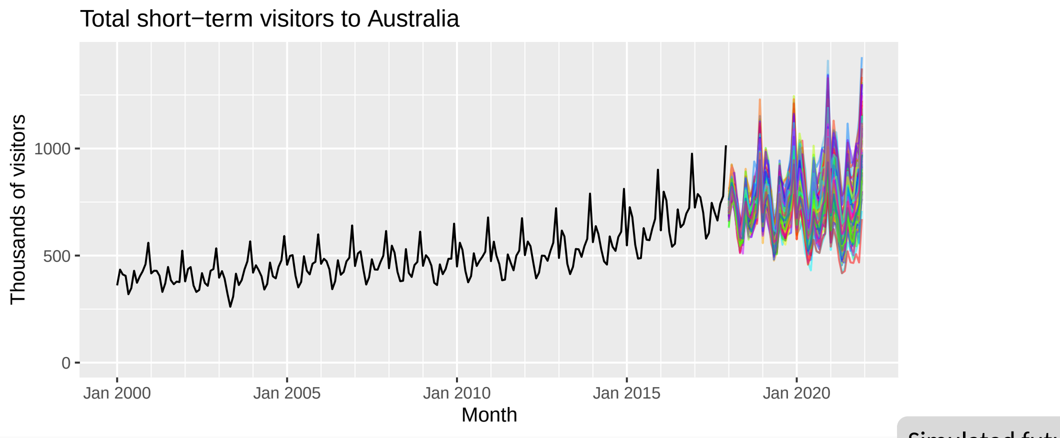
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

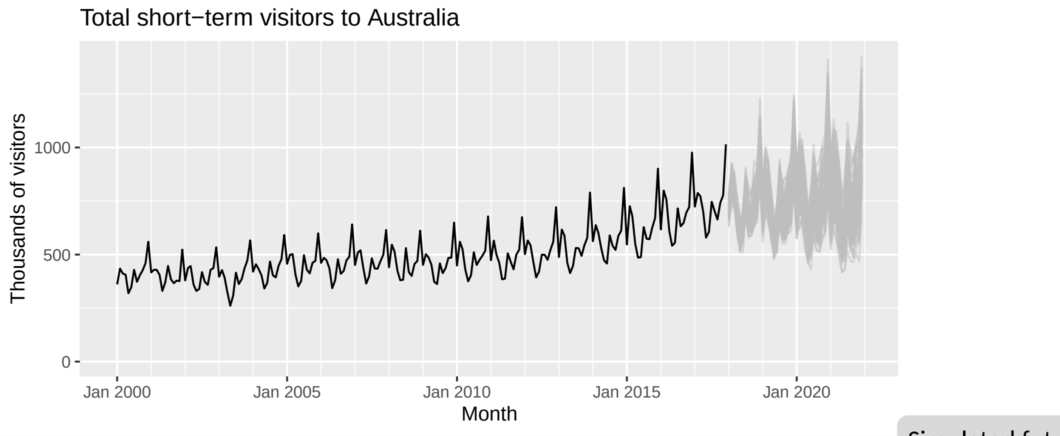
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

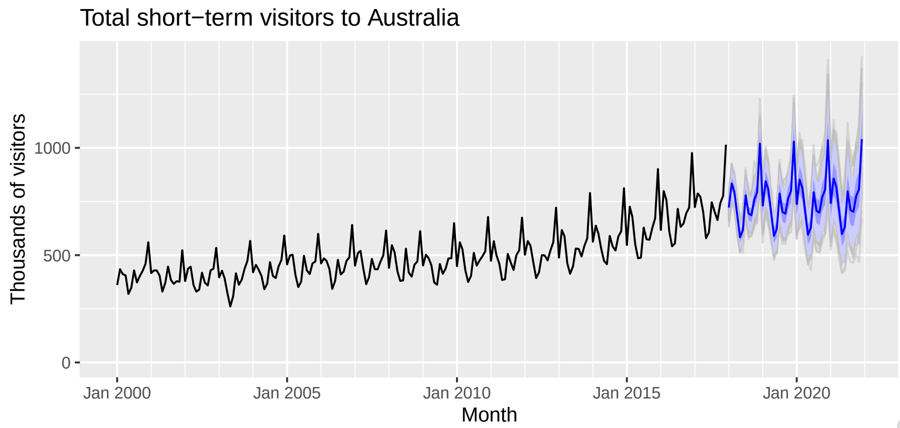
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

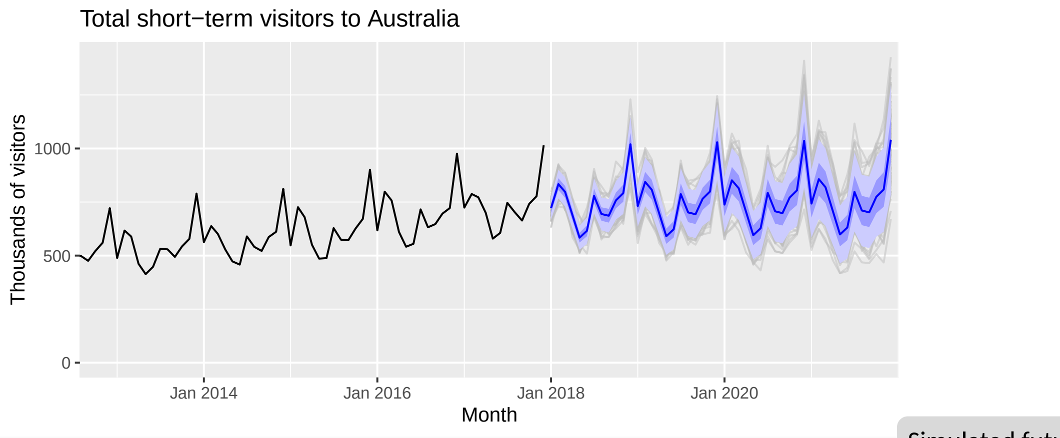
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

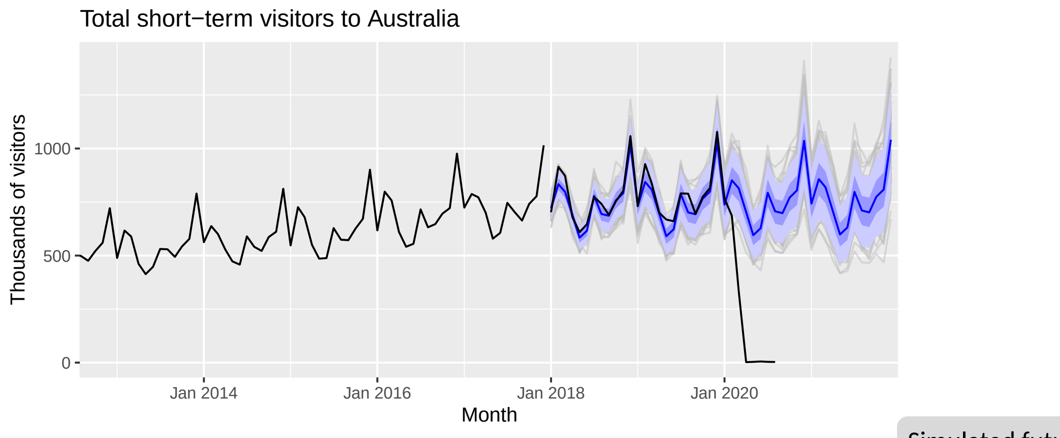
A forecast is an estimate of the probability distribution of a variable to be observed in the future.



Simulated futures from an ETS model

Random futures

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



Simulated futures from an ETS model

Model fitting

```r
holiday_fit <- holidays |>
  model(
    snaive = SNAIVE(Trips),
    naive = NAIVE(Trips),
    ets = ETS(Trips),
    arima = ARIMA(Trips)
  )
```

```
## # A mable: 8 x 5
## # Key:     State [8]
##   State  snaive   naive         ets                               arima
##   <fct> <model> <model>     <model>                             <model>
## 1 NSW   <SNAIVE> <NAIVE> <ETS(M,N,A)>         <ARIMA(0,1,1)(0,1,1)[4]>
## 2 VIC   <SNAIVE> <NAIVE> <ETS(M,A,M)>         <ARIMA(0,1,1)(0,1,1)[4]>
## 3 QLD   <SNAIVE> <NAIVE> <ETS(A,N,A)> <ARIMA(0,0,0)(1,0,0)[4] w/ mean>
## 4 WA    <SNAIVE> <NAIVE> <ETS(M,N,M)>         <ARIMA(1,0,1)(0,1,1)[4]>
## 5 SA    <SNAIVE> <NAIVE> <ETS(M,N,A)>         <ARIMA(0,0,0)(1,1,1)[4]>
## 6 TAS   <SNAIVE> <NAIVE> <ETS(M,N,A)>         <ARIMA(0,0,0)(0,1,1)[4]>
```

Model fitting

```r
holiday_fit |>
  filter(State == "NSW") |>
  select(arima) |>
  report()
```

```
## Series: Trips
## Model: ARIMA(0,1,1)(0,1,1)[4]
##
## Coefficients:
##           ma1    sma1
##       -0.7424  -0.805
## s.e.   0.0963   0.113
##
## sigma^2 estimated as 33036:  log likelihood=-498
## AIC=1003   AICc=1003   BIC=1010
```

Model fitting

```r
augment(holiday_fit)
```

```
## # A tsibble: 2,560 x 7 [1Q]
## # Key:       State, .model [32]
##    State .model Quarter Trips .fitted .resid .innov
##    <fct> <chr>    <qtr> <dbl>   <dbl>  <dbl>  <dbl>
##  1 NSW   snaive 1998 Q1 4033.     NA     NA      NA
##  2 NSW   snaive 1998 Q2 3262.     NA     NA      NA
##  3 NSW   snaive 1998 Q3 2681.     NA     NA      NA
##  4 NSW   snaive 1998 Q4 3083.     NA     NA      NA
##  5 NSW   snaive 1999 Q1 3635.   4033. -398.   -398.
##  6 NSW   snaive 1999 Q2 2958.   3262. -305.   -305.
##  7 NSW   snaive 1999 Q3 2768.   2681.   87.2   87.2
##  8 NSW   snaive 1999 Q4 3121.   3083.   38.1   38.1
##  9 NSW   snaive 2000 Q1 3548.   3635.  -86.4  -86.4
## 10 NSW   snaive 2000 Q2 2840.   2958. -118.   -118.
## # ... with 2,550 more rows
```

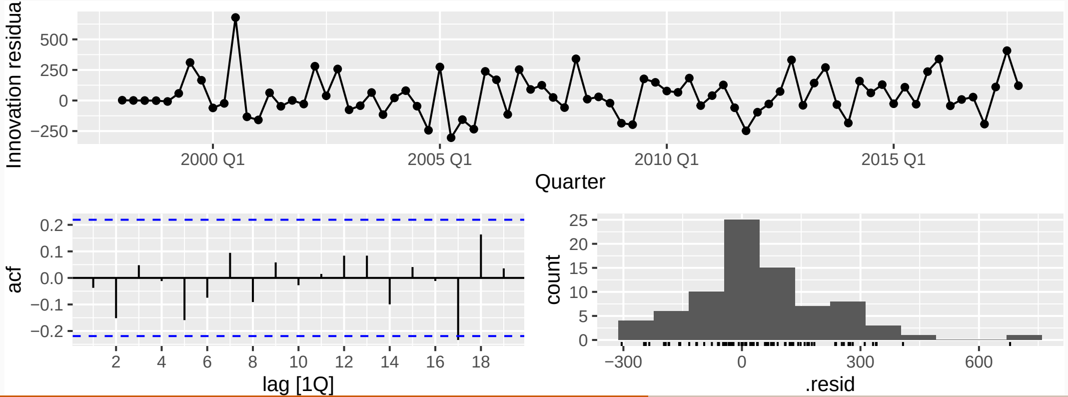
Ljung-Box test

```r
augment(holiday_fit) |>
  filter(State == "NSW", .model == "arima") |>
  features(.resid, ljung_box, dof = 2, lag = 8)
```

```
## # A tibble: 1 x 4
##   State .model lb_stat lb_pvalue
##   <fct> <chr>    <dbl>     <dbl>
## 1 NSW   arima     6.52     0.368
```

gg_tsresiduals() function

```r
holiday_fit |>
  filter(State == "NSW") |>
  select(arima) |>
  gg_tsresiduals()
```



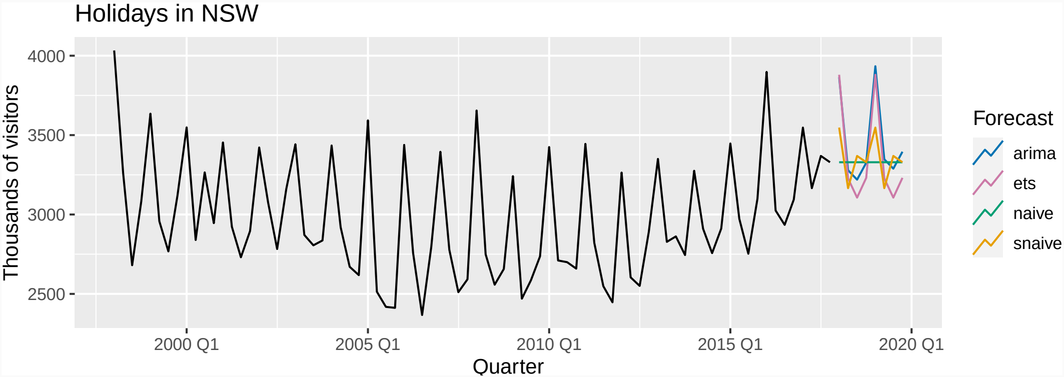
Producing forecasts

```r
holiday_fc <- holiday_fit |>
  forecast(h = "2 years")
```

```
## # A fable: 256 x 5 [1Q]
## # Key:     State, .model [32]
##    State .model Quarter            Trips .mean
##    <fct> <chr>    <qtr>           <dist> <dbl>
##  1 NSW   snaive 2018 Q1  N(3547, 45906) 3547.
##  2 NSW   snaive 2018 Q2  N(3166, 45906) 3166.
##  3 NSW   snaive 2018 Q3  N(3369, 45906) 3369.
##  4 NSW   snaive 2018 Q4  N(3329, 45906) 3329.
##  5 NSW   snaive 2019 Q1  N(3547, 91812) 3547.
##  6 NSW   snaive 2019 Q2  N(3166, 91812) 3166.
##  7 NSW   snaive 2019 Q3  N(3369, 91812) 3369.
##  8 NSW   snaive 2019 Q4  N(3329, 91812) 3329.
##  9 NSW   naive  2018 Q1 N(3329, 251179) 3329.
## 10 NSW   naive  2018 Q2  N(3329, 5e+05) 3329.
## # ... with 246 more rows
```

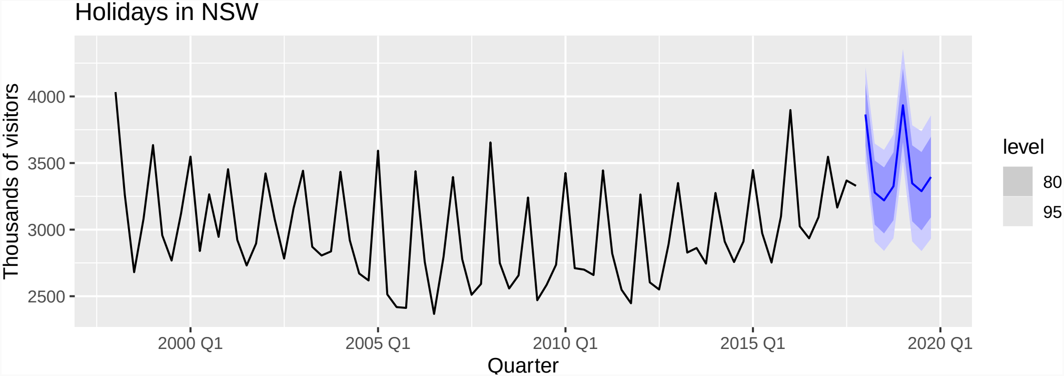
Visualising forecasts

```r
holiday_fc |>
  filter(State == "NSW") |>
  autoplot(holidays, level = NULL) +
  labs(title = "Holidays in NSW", y = "Thousands of visitors") +
  guides(color = guide_legend(title = "Forecast"))
```



Visualising forecasts

```
holiday_fc |>
  filter(State == "NSW", .model == "arima") |>
  autoplot(holidays) +
  labs(title = "Holidays in NSW", y = "Thousands of visitors") +
  guides(color = guide_legend(title = "Forecast"))
```



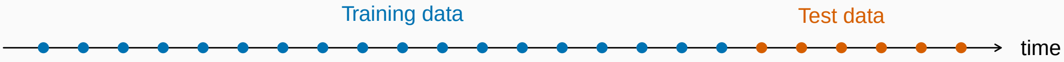
Prediction intervals

```r
holiday_fc |> hilo(level = 95)
```

```
## # A tsibble: 256 x 6 [1Q]
## # Key:       State, .model [32]
##    State .model Quarter          Trips .mean          `95%`
##    <fct> <chr>    <qtr>         <dist> <dbl>         <hilo>
##  1 NSW   snaive 2018 Q1  N(3547, 45906) 3547. [3127, 3967]95
##  2 NSW   snaive 2018 Q2  N(3166, 45906) 3166. [2746, 3586]95
##  3 NSW   snaive 2018 Q3  N(3369, 45906) 3369. [2949, 3789]95
##  4 NSW   snaive 2018 Q4  N(3329, 45906) 3329. [2909, 3749]95
##  5 NSW   snaive 2019 Q1  N(3547, 91812) 3547. [2954, 4141]95
##  6 NSW   snaive 2019 Q2  N(3166, 91812) 3166. [2572, 3760]95
##  7 NSW   snaive 2019 Q3  N(3369, 91812) 3369. [2775, 3963]95
##  8 NSW   snaive 2019 Q4  N(3329, 91812) 3329. [2735, 3923]95
##  9 NSW   naive  2018 Q1 N(3329, 251179) 3329. [2347, 4311]95
## 10 NSW   naive  2018 Q2  N(3329, 5e+05) 3329. [1940, 4718]95
## # ... with 246 more rows
```

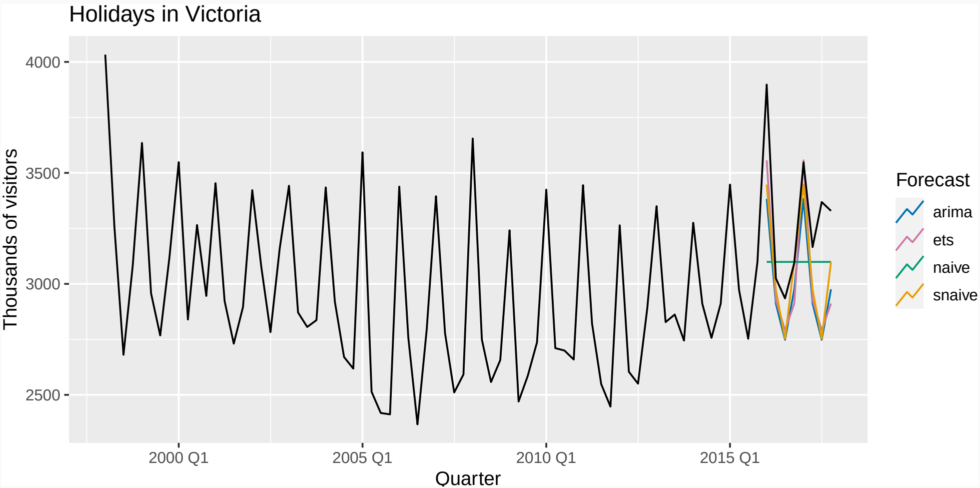
Outline

Training and test sets



- A model which fits the training data well will not necessarily forecast well.
- A perfect fit can always be obtained by using a model with enough parameters.
- Over-fitting a model to data is just as bad as failing to identify a systematic pattern in the data.
- The test set must not be used for *any* aspect of model development or calculation of forecasts.
- Forecast accuracy is based only on the test set.

Measures of forecast accuracy



Measures of forecast accuracy

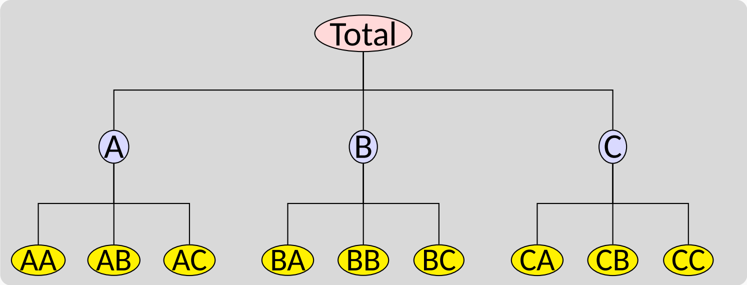
```r
accuracy(nsw_fc, holidays)
```

```
## # A tibble: 4 x 11
##   .model State .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
##   <chr>  <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 arima  NSW   Test   291.  341.  291.  8.61  8.61  1.80  1.66  0.126
## 2 ets    NSW   Test   249.  305.  251.  7.47  7.54  1.55  1.49  0.323
## 3 naive  NSW   Test   196.  354.  257.  5.24  7.30  1.59  1.72 -0.120
## 4 snaive NSW   Test   227.  300.  228.  6.68  6.72  1.41  1.46  0.00301
```

Outline

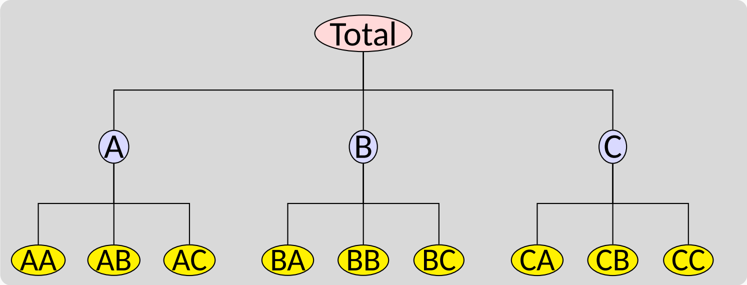
Hierarchical time series

A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.



Hierarchical time series

A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

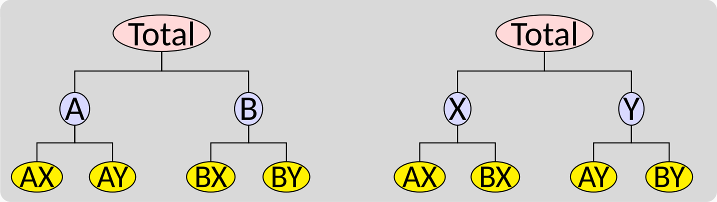


Examples

- Tourism demand by states, zones, regions

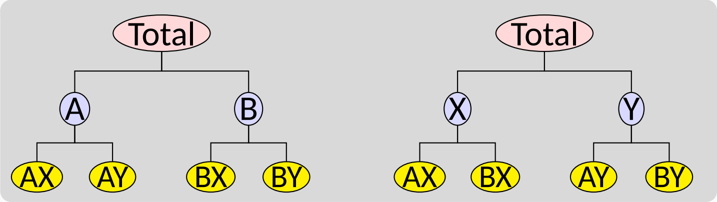
Grouped time series

A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.



Grouped time series

A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.



Examples

- Tourism by state and purpose of travel
- Retail sales by product groups/sub groups, and by countries/regions

Creating aggregates

```r
tourism |>
  aggregate_key(Purpose * (State / Region), Trips = sum(Trips)) |>
  filter(Quarter == yearquarter("1998 Q1")) |>
  print(n = 15)
```

```
## # A tsibble: 425 x 5 [1Q]
## # Key:       Purpose, State, Region [425]
##    Quarter Purpose      State        Region        Trips
##      <qtr> <chr*>       <fct*>       <chr*>        <dbl>
##  1 1998 Q1 <aggregated> <aggregated> <aggregated>  23182.
##  2 1998 Q1 Business     <aggregated> <aggregated>   3599.
##  3 1998 Q1 Holiday      <aggregated> <aggregated>  11806.
##  4 1998 Q1 Other        <aggregated> <aggregated>    680.
##  5 1998 Q1 Visiting     <aggregated> <aggregated>   7098.
##  6 1998 Q1 <aggregated> NSW          <aggregated>   8040.
##  7 1998 Q1 <aggregated> VIC          <aggregated>   6010.
##  8 1998 Q1 <aggregated> QLD          <aggregated>   4041.
##  9 1998 Q1 <aggregated> WA           <aggregated>   1641.
## 10 1998 Q1 <aggregated> SA           <aggregated>   1735.
## 11 1998 Q1 <aggregated> TAS          <aggregated>    982.
## 12 1998 Q1 <aggregated> NT           <aggregated>    181.
## 13 1998 Q1 <aggregated> ACT          <aggregated>    551.
```

Creating aggregates

- Similar to `summarise()` but using the key structure
- A grouped structure is specified using `grp1 * grp2`
- A nested structure is specified via `parent / child`.
- Groups and nesting can be mixed:

```
(country/region/city) * (brand/product)
```

- All possible aggregates are produced.
- These are useful when forecasting at different levels of aggregation.

The problem

1. How to forecast time series at all nodes such that the forecasts add up in the same way as the original data?
2. Can we exploit relationships between the series to improve the forecasts?

The problem

1. How to forecast time series at all nodes such that the forecasts add up in the same way as the original data?
2. Can we exploit relationships between the series to improve the forecasts?

The solution

1. Forecast all series at all levels of aggregation using an automatic forecasting algorithm.
   (e.g., `ETS`, `ARIMA`, ...)
2. Reconcile the resulting forecasts so they add up correctly using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).
3. This is available using `reconcile()`.

Forecast reconciliation

```
tourism |>
  aggregate_key(Purpose * (State / Region), Trips = sum(Trips)) |>
  model(ets = ETS(Trips)) |>
  reconcile(ets_adjusted = min_trace(ets)) |>
  forecast(h = 2)
```

```
## # A fable: 1,700 x 7 [1Q]
## # Key:     Purpose, State, Region, .model [850]
##    Purpose  State  Region          .model       Quarter       Trips .mean
##    <chr*>   <fct*> <chr*>          <chr>          <qtr>      <dist> <dbl>
##  1 Business NSW    Blue Mountains  ets          2018 Q1 N(20, 140)  19.7
##  2 Business NSW    Blue Mountains  ets          2018 Q2 N(20, 140)  19.7
##  3 Business NSW    Blue Mountains  ets_adjusted 2018 Q1 N(20, 133)  20.2
##  4 Business NSW    Blue Mountains  ets_adjusted 2018 Q2 N(21, 143)  20.5
##  5 Business NSW    Capital Country ets          2018 Q1 N(36, 202)  36.1
##  6 Business NSW    Capital Country ets          2018 Q2 N(36, 202)  36.1
##  7 Business NSW    Capital Country ets_adjusted 2018 Q1 N(37, 190)  37.5
##  8 Business NSW    Capital Country ets_adjusted 2018 Q2 N(38, 194)  38.2
```

Hierarchical and grouped time series

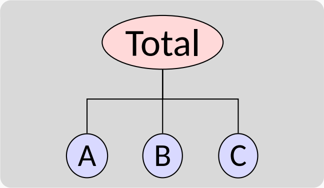
Every collection of time series with aggregation constraints can be written as

$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

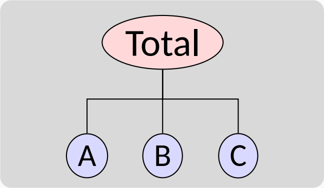
where

- $\boldsymbol{y}_t$ is a vector of all series at time t
- $\boldsymbol{b}_t$ is a vector of the most disaggregated series at time t
- $\boldsymbol{S}$ is a “summing matrix’ ’ containing the aggregation constraints.

Hierarchical time series



Hierarchical time series

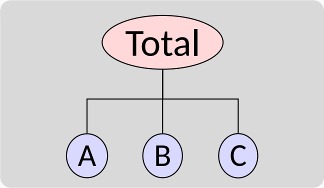


y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

Hierarchical time series



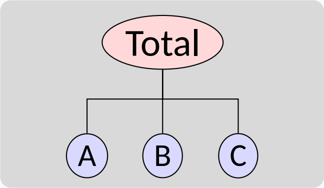
y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}$$

Hierarchical time series



y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.
(In general, they will not "add up' '.)

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.
(In general, they will not "add up' '.)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{G}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{G}$.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.
(In general, they will not "add up".)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{G}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{G}$.

- $\boldsymbol{G}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_n(h)$ to get bottom-level forecasts.
- $\boldsymbol{S}$ adds them up

Optimal combination forecasts

Main result

The best (minimum sum of variances) unbiased forecasts are obtained when $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}$, where $\boldsymbol{W}_h$ is the h-step base forecast error covariance matrix.

Optimal combination forecasts

Main result

The best (minimum sum of variances) unbiased forecasts are obtained when $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}$, where $\boldsymbol{W}_h$ is the h-step base forecast error covariance matrix.

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}\hat{\boldsymbol{y}}_n(h)$$

Problem: $\boldsymbol{W}_h$ hard to estimate, especially for $h > 1$.

Solutions:

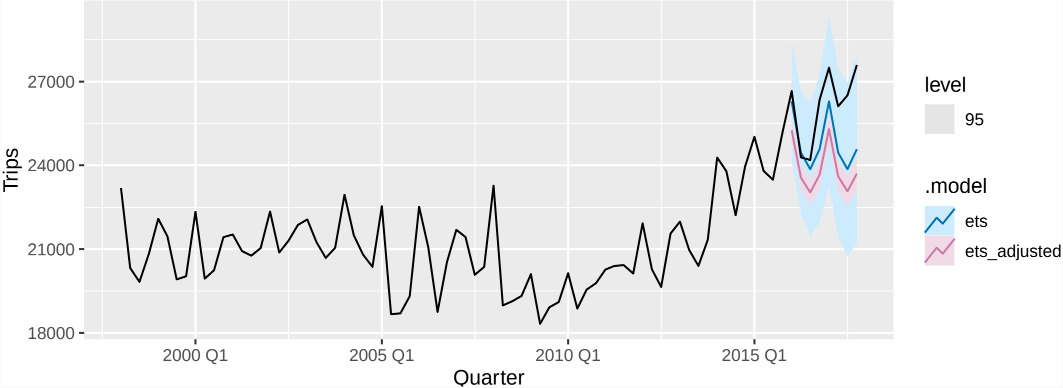
- Ignore $\boldsymbol{W}_h$ (OLS) [`min_trace(method='ols')`]
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$ is diagonal (WLS) [`min_trace(method='wls')`]
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$ and estimate it (GLS)

Example: Australian tourism

```
tourism_agg <- tourism |>
  aggregate_key(Purpose * (State / Region),
    Trips = sum(Trips)
  )
fc <- tourism_agg |>
  filter_index(. ~ "2015 Q4") |>
  model(ets = ETS(Trips)) |>
  reconcile(ets_adjusted = min_trace(ets)) |>
  forecast(h = "2 years")
```

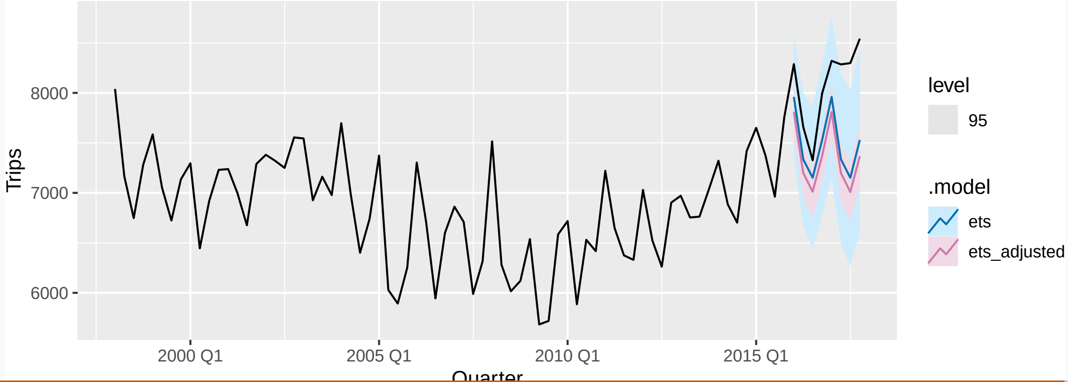
Example: Australian tourism

```r
fc |>
  filter(is_aggregated(Purpose) & is_aggregated(State)) |>
  autoplot(tourism_agg, level = 95)
```



Example: Australian tourism

```r
fc |>
  filter(is_aggregated(Purpose) & State == "NSW" &
    is_aggregated(Region)) |>
  autoplot(tourism_agg, level = 95)
```



Example: Australian tourism

```r
fc <- tourism_agg |>
  filter_index(. ~ "2015 Q4") |>
  model(
    ets = ETS(Trips),
    arima = ARIMA(Trips)
  ) |>
  mutate(
    comb = (ets + arima) / 2
  ) |>
  reconcile(
    ets_adj = min_trace(ets),
    arima_adj = min_trace(arima),
    comb_adj = min_trace(comb)
  ) |>
  forecast(h = "2 years")
```

Forecast evaluation

```r
fc |> accuracy(tourism_agg)
```

```
## # A tibble: 2,550 x 13
##    .model Purpose  State  Region                .type    ME  RMSE   MAE    MPE
##    <chr>  <chr*>   <fct*> <chr*>                <chr> <dbl> <dbl> <dbl>  <dbl>
##  1 arima  Business NSW    Blue Mountains      ~ Test   1.93 10.6   8.52 -18.0
##  2 arima  Business NSW    Capital Country     ~ Test   8.08 15.6  10.4   11.8
##  3 arima  Business NSW    Central Coast       ~ Test  10.0  14.5  10.8   26.9
##  4 arima  Business NSW    Central NSW         ~ Test  17.7  31.9  28.2   12.0
##  5 arima  Business NSW    Hunter              ~ Test  35.3  43.9  35.3   24.2
##  6 arima  Business NSW    New England North~ Test  23.1  31.8  26.8   19.5
##  7 arima  Business NSW    North Coast NSW     ~ Test  24.8  40.1  36.8   11.5
##  8 arima  Business NSW    Outback NSW         ~ Test   6.87 11.0   7.76  13.7
##  9 arima  Business NSW    Riverina            ~ Test   5.84 20.4  16.5   -2.48
## 10 arima  Business NSW    Snowy Mountains     ~ Test   5.48  9.54  8.24  12.3
## # ... with 2,540 more rows, and 4 more variables: MAPE <dbl>, MASE <dbl>,
```

Forecast evaluation

```r
fc |>
  accuracy(tourism_agg) |>
  group_by(.model) |>
  summarise(MASE = mean(MASE)) |>
  arrange(MASE)
```

```
## # A tibble: 6 x 2
##   .model     MASE
##   <chr>     <dbl>
## 1 ets_adj    1.02
## 2 comb_adj   1.02
## 3 ets        1.04
## 4 comb       1.04
## 5 arima_adj  1.07
## 6 arima      1.09
```

More information

- Slides and papers: **robjhyndman.com**
- Packages: **tidyverts.org**
- Forecasting textbook using fable package: **OTexts.com/fpp3**

Find me at . . .

@robjhyndman
@robjhyndman
robjhyndman.com
rob.hyndman@monash.edu