Cardiff University
Prifysgol Caerdydd

Cardiff Business School
Ysgol Busnes Caerdydd

Forecasting in R

Forecast many series

Bahman Rostami-Tabar

Outline

Outline

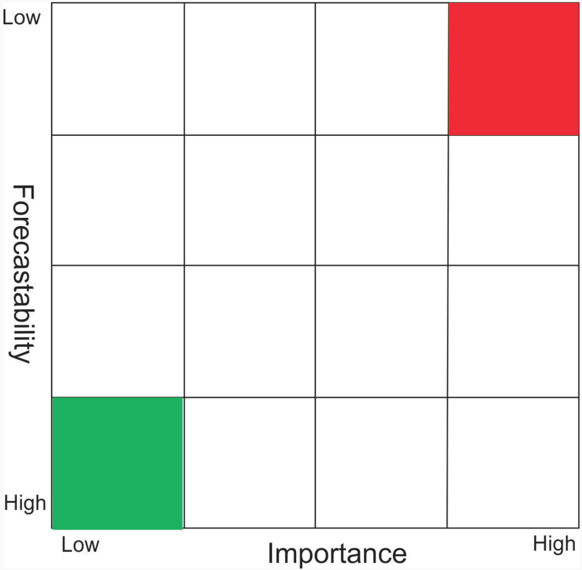
Learning outcomes

You should be able to:

1. Produce foreacsts for many time series
2. Extract infomration about specific model and series
3. Calculate forecast accuracy for many time series and summarise results

Outline

Classification: forecastability vs. importance



Using `tsibble` and `fable`

- Most time series can be naturally disaggregated using a series of factors known as keys
- These keys are used to uniquely identify separate time series, each of which can be modelled separately.
- This structure allows batch forecasting to be applied across many time series.
- Estimating multiple models is a key feature of fable.

The key to many time series

```
tourism <- imported_data %>%
  as_tsibble(
    index = Quarter,
    key = c(Region, State, Purpose)
  )
```

The key to many time series

```
tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State           Purpose  Trips
##      <qtr> <chr>    <chr>           <chr>    <dbl>
##  1 1998 Q1 Adelaide South Australia Business  135.
##  2 1998 Q2 Adelaide South Australia Business  110.
##  3 1998 Q3 Adelaide South Australia Business  166.
##  4 1998 Q4 Adelaide South Australia Business  127.
##  5 1999 Q1 Adelaide South Australia Business  137.
##  6 1999 Q2 Adelaide South Australia Business  200.
##  7 1999 Q3 Adelaide South Australia Business  169.
##  8 1999 Q4 Adelaide South Australia Business  134.
##  9 2000 Q1 Adelaide South Australia Business  154.
## 10 2000 Q2 Adelaide South Australia Business  169.
## # ... with 24,310 more rows
```

Outline

Specify model

- The mable includes models for every combination of keys in the data set.

```r
train <- tourism %>%
  filter_index(~ "2014-12")
fit <- train %>%
  model(
    snaive = SNAIVE(Trips ~ lag("year")),
    ets = ETS(Trips),
    arima = ARIMA(Trips)
  )
```

mable

```
fit
```

```
## # A mable: 304 x 6
## # Key:     Region, State, Purpose [304]
##    Region   State   Purpose  snaive ets   arima
##    <chr>    <chr>   <chr>    <mode> <mod> <model>
##  1 Adelaide South ~ Business <SNAI~ <ETS~ <ARIMA(~
##  2 Adelaide South ~ Holiday  <SNAI~ <ETS~ <ARIMA(~
##  3 Adelaide South ~ Other    <SNAI~ <ETS~ <ARIMA(~
##  4 Adelaide South ~ Visiting <SNAI~ <ETS~ <ARIMA(~
##  5 Adelaid~ South ~ Business <SNAI~ <ETS~ <ARIMA(~
##  6 Adelaid~ South ~ Holiday  <SNAI~ <ETS~ <ARIMA(~
##  7 Adelaid~ South ~ Other    <SNAI~ <ETS~ <ARIMA(~
##  8 Adelaid~ South ~ Visiting <SNAI~ <ETS~ <ARIMA(~
##  9 Alice S~ Northe~ Business <SNAI~ <ETS~ <ARIMA(~
## 10 Alice S~ Northe~ Holiday  <SNAI~ <ETS~ <ARIMA(~
## # ... with 294 more rows
```

Extract information

```r
fit %>%
  filter(Region == "Snowy Mountains", Purpose == "Holiday") %>%
  select(arima) %>%
  report()
```

```
## Series: Trips
## Model: ARIMA(0,0,0)(0,1,2)[4]
##
## Coefficients:
##          sma1     sma2
##       -0.6223  -0.2557
## s.e.   0.1947   0.1508
##
## sigma^2 estimated as 493:  log likelihood=-290.82
## AIC=587.65   AICc=588.05   BIC=594.12
```

Forecast many series

- `mable` is passed to the `forecast()` function
- Forecasts are computed for every model and every key combination

```
fc <- fit %>%
  forecast(h = "3 years")
```

`fable` object

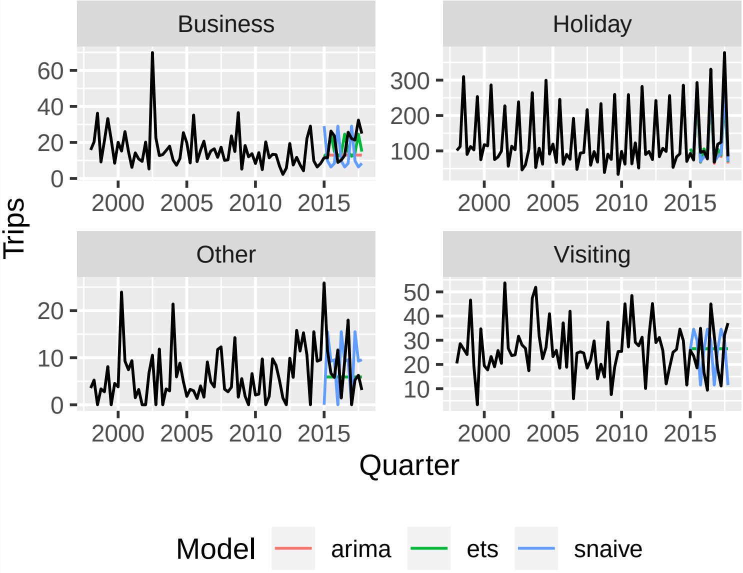
```
fc
```

```
## # A fable: 10,944 x 7 [1Q]
## # Key:     Region, State, Purpose, .model [912]
##    Region State Purpose .model    Quarter Trips
##    <chr>  <chr> <chr>   <chr>       <qtr> <dbl>
##  1 Adela~ Sout~ Busine~ snaive    2015 Q1  139.
##  2 Adela~ Sout~ Busine~ snaive    2015 Q2  155.
##  3 Adela~ Sout~ Busine~ snaive    2015 Q3  186.
##  4 Adela~ Sout~ Busine~ snaive    2015 Q4  168.
##  5 Adela~ Sout~ Busine~ snaive    2016 Q1  139.
##  6 Adela~ Sout~ Busine~ snaive    2016 Q2  155.
##  7 Adela~ Sout~ Busine~ snaive    2016 Q3  186.
##  8 Adela~ Sout~ Busine~ snaive    2016 Q4  168.
##  9 Adela~ Sout~ Busine~ snaive    2017 Q1  139.
```

Visualise forecasts

```r
snowy_mountains <- fc %>% filter(Region == "Snowy Mountains")
p1 <- snowy_mountains %>%
  ggplot(aes(x=Quarter, y=Trips, colour=.model))+
  geom_line()+
  geom_line(data = filter(tourism,Region == "Snowy Mountains"),
            aes(x=Quarter, y=Trips, group=), colour="black")+
  facet_wrap(~Purpose, scales = "free")+
  theme(legend.position = "bottom")+
  labs(colour="Model")
```

Visualise forecasts



Evaluate accuracy

```r
accuracy(fc, tourism)
```

```
## # A tibble: 912 x 12
##    .model Region State Purpose .type    ME  RMSE
##    <chr>  <chr>  <chr> <chr>   <chr> <dbl> <dbl>
##  1 arima  Adela~ Sout~ Busine~ Test  22.5  28.5 
##  2 arima  Adela~ Sout~ Holiday Test  21.9  34.8 
##  3 arima  Adela~ Sout~ Other   Test   4.71 17.5 
##  4 arima  Adela~ Sout~ Visiti~ Test  32.8  37.1 
##  5 arima  Adela~ Sout~ Busine~ Test   1.31  5.58
##  6 arima  Adela~ Sout~ Holiday Test   6.46  7.43
##  7 arima  Adela~ Sout~ Other   Test   1.35  2.79
##  8 arima  Adela~ Sout~ Visiti~ Test   8.37 12.6 
##  9 arima  Alice~ Nort~ Busine~ Test   9.85 12.2 
## 10 arima  Alice~ Nort~ Holiday Test   4.80 11.3 
## # ... with 902 more rows, and 5 more variables:
## #   MAE <dbl>, MPE <dbl>, MAPE <dbl>, MASE <dbl>,
## #   ACF1 <dbl>
```

Evaluate accuracy

```r
accuracy_model <- accuracy(fc, tourism) %>%
  group_by(.model) %>%
  summarise(
    RMSE = mean(RMSE),
    MAE = mean(MAE),
    MASE = mean(MASE)
  ) %>%
  arrange(RMSE)
```

Evaluate accuracy

```
accuracy_model
```

```
## # A tibble: 3 x 4
##   .model  RMSE   MAE  MASE
##   <chr>  <dbl> <dbl> <dbl>
## 1 ets     20.2  16.4  1.00
## 2 snaive  21.5  17.3  1.17
## 3 arima   21.9  17.8  1.07
```

Outline

Lab Session 12

```r
ae_day <- ae_tsb %>% group_by(gender, type_injury) %>%
  index_by(year_day = as_date(arrival_time)) %>%
  summarise(n_attendance=n()) %>%
  fill_gaps(n_attendance=0L) %>%
  ungroup()
```

Lab Session 12

1. Create a new tsibble with daily intervals as showed above
2. Consider the last 42 days as test and the rest as training
3. Fit ets, arima and snaive on training set
4. Extract information about ets model for `gender=="male"` and `type_injury=="major"` using the filter, select and report functions
5. Pass fitted models to the `forecast()` function, compute forecasts for every model and every key combination for the next `42 days`.
6. Calculate accuracy and summarise it for MASE and Winkler score
7. Which model is more accuracte?