CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Cardiff Business School
Ysgol Busnes Caerdydd

Forecasting in R

Forecasting by aggregation

Bahman Rostami-Tabar

Outline

Outline

Learning outcomes

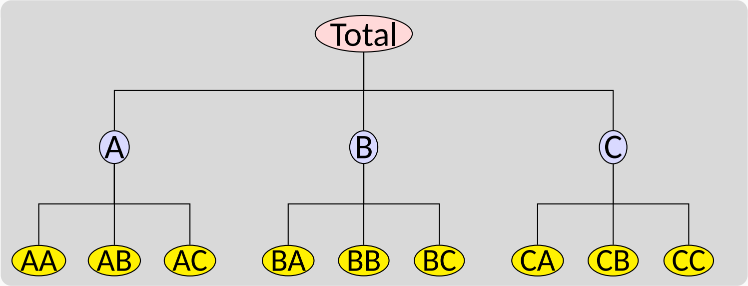
You should be able to:

1. Create a hierarchical/ group time series structure
2. Produce forecasts for any desired level of hierarchy or any group
3. Calculate forecast accuracy

Outline

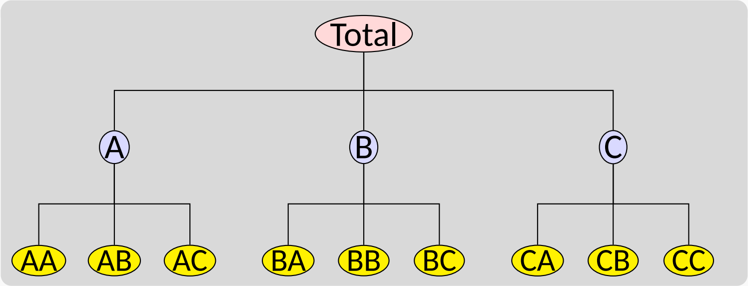
Hierarchical time series

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



Hierarchical time series

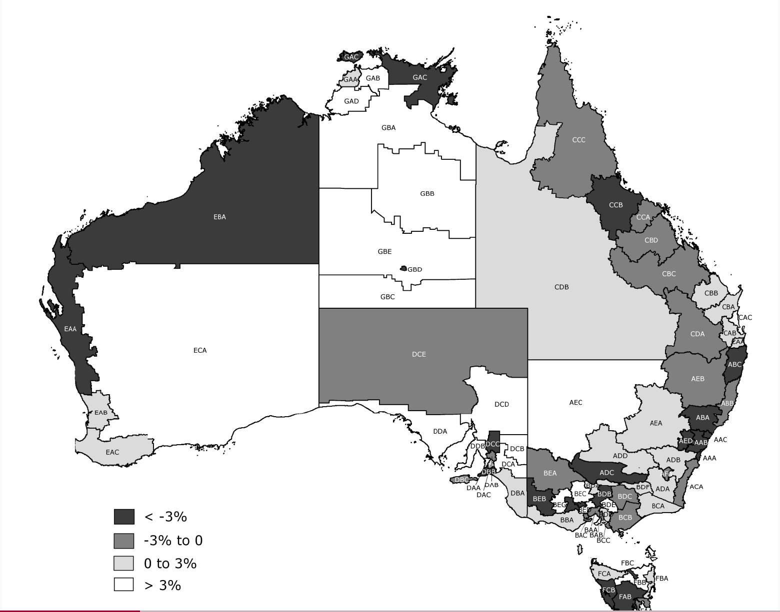
A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.



Examples

- Tourism demand by states, zones, regions

Australian tourism



Australian tourism

```r
tourism
```

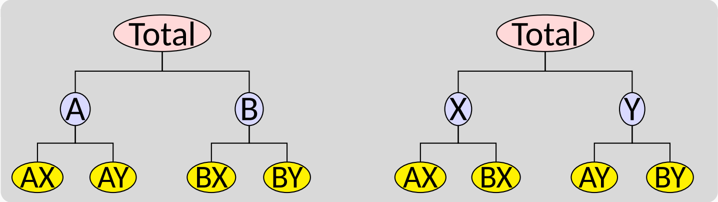
```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State           Purpose  Trips
##      <qtr> <chr>    <chr>           <chr>    <dbl>
##  1 1998 Q1 Adelaide South Australia Business  135.
##  2 1998 Q2 Adelaide South Australia Business  110.
##  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.
```

Australian tourism

- Quarterly data on visitor night from 1998:Q1 – 2013:Q4
- From: *National Visitor Survey*, based on annual interviews of 120,000 Australians aged 15+, collected by Tourism Research Australia.
- Split by 7 states, 27 zones and 76 regions (a geographical hierarchy)
- Also split by purpose of travel
  - Holiday
  - Visiting friends and relatives (VFR)
  - Business
  - Other
- 304 bottom-level series

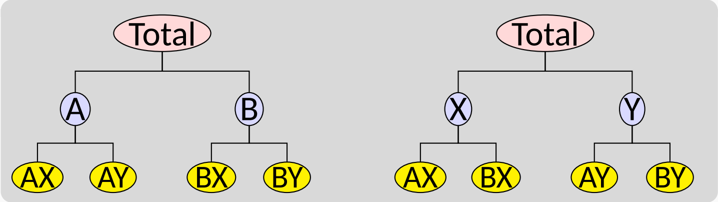
Grouped time series

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

key in tsibble

- Keys are used within tsibble to uniquely identify related time series in a tidy structure
- Useful for identifying relational structures between each time series
- Useful where a hierarchical or grouped structure is imposed on a set of forecasts to impose relational constraints (typically aggregation).
- Keys within tsibble can be either nested (hierarchical) or crossed (grouped), and can be directly used to reconcile forecasts.

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 [?]
## # Key:       Purpose, State, Region [425]
##    Purpose      State            Region        Quarter  Trips
##    <chr>        <chr>            <chr>           <qtr>  <dbl>
##  1 <aggregated> <aggregated>     <aggregated>  1998 Q1 23182.
##  2 Business     <aggregated>     <aggregated>  1998 Q1  3599.
##  3 Holiday      <aggregated>     <aggregated>  1998 Q1 11806.
##  4 Other        <aggregated>     <aggregated>  1998 Q1   680.
##  5 Visiting     <aggregated>     <aggregated>  1998 Q1  7098.
##  6 <aggregated> ACT            ~ <aggregated>  1998 Q1   551.
##  7 <aggregated> New South Wale~ <aggregated>  1998 Q1  8040.
##  8 <aggregated> Northern Terri~ <aggregated>  1998 Q1   181.
##  9 <aggregated> Queensland     ~ <aggregated>  1998 Q1  4041.
## 10 <aggregated> South Australi~ <aggregated>  1998 Q1  1735.
## 11 <aggregated> Tasmania       ~ <aggregated>  1998 Q1   982.
## 12 <aggregated> Victoria       ~ <aggregated>  1998 Q1  6010.
## 13 <aggregated> Western Austra~ <aggregated>  1998 Q1  1641.
## 14 <aggregated> ACT            ~ Canberra    ~ 1998 Q1   551.
## 15 <aggregated> New South Wale~ Blue Mounta~ 1998 Q1   196.
```

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.

Outline

The problem

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

The problem

- How to forecast time series at all nodes such that the forecasts add up in the same way as the original data?
- 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

```r
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
##    <chr>      <chr>      <chr>      <chr>       <qtr> <dbl>
##  1 Business   ACT      ~ Canberra ~ ets       2018 Q1 144.
##  2 Business   ACT      ~ Canberra ~ ets       2018 Q2 203.
##  3 Business   ACT      ~ <aggregat~ ets       2018 Q1 144.
##  4 Business   ACT      ~ <aggregat~ ets       2018 Q2 203.
##  5 Business   New South~ Blue Moun~ ets       2018 Q1  19.7
##  6 Business   New South~ Blue Moun~ ets       2018 Q2  19.7
##  7 Business   New South~ Capital C~ ets       2018 Q1  36.1
##  8 Business   New South~ Capital C~ ets       2018 Q2  36.1
##  9 Business   New South~ Central C~ ets       2018 Q1  25.7
```

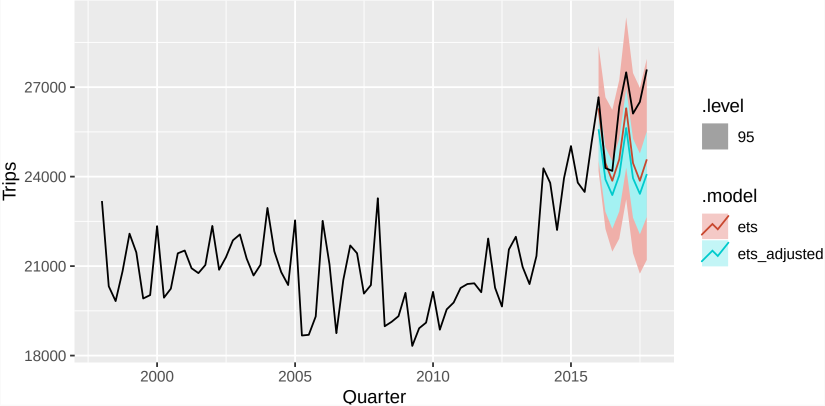
Outline

Example: Australian tourism

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

Example: Australian tourism

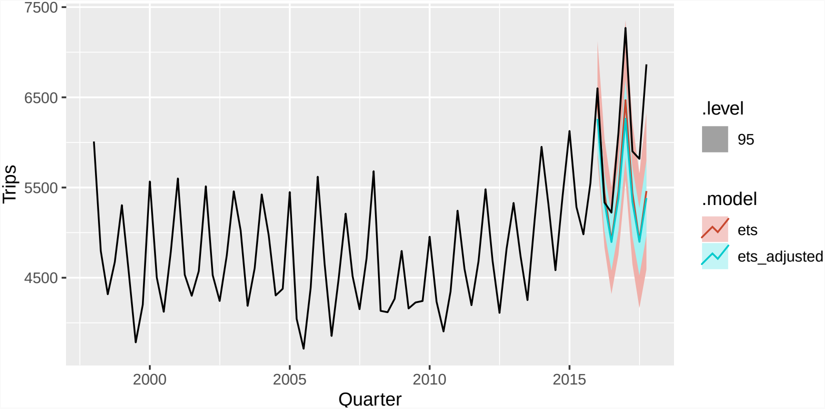
```r
fc %>%
  filter(is_aggregated(Purpose) & is_aggregated(State)) %>%
  autoplot(tourism_agg, level = 95)
```




```r
unique(fc$State)
```

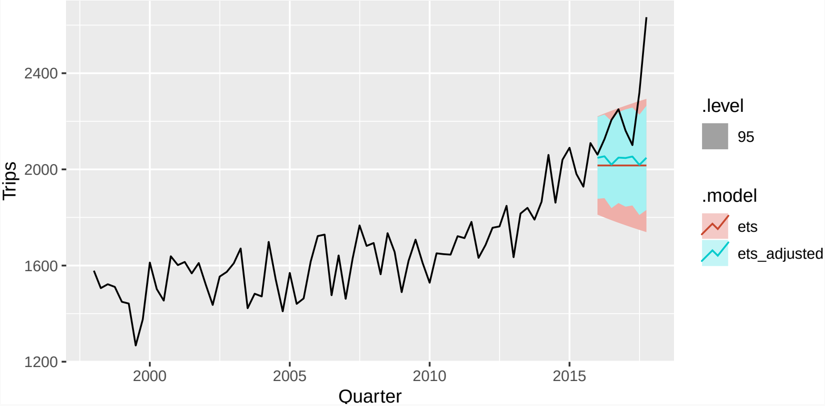
Example: Australian tourism

```
fc %>%
  filter(is_aggregated(Purpose) & State == "Victoria" &
    is_aggregated(Region)) %>%
  autoplot(tourism_agg, level = 95)
```



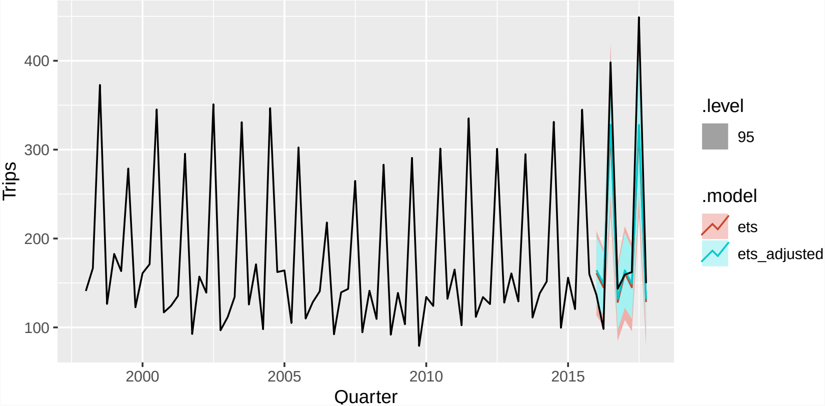
Example: Australian tourism

```
fc %>%
  filter(is_aggregated(Purpose) & Region == "Melbourne") %>%
  autoplot(tourism_agg, level = 95)
```



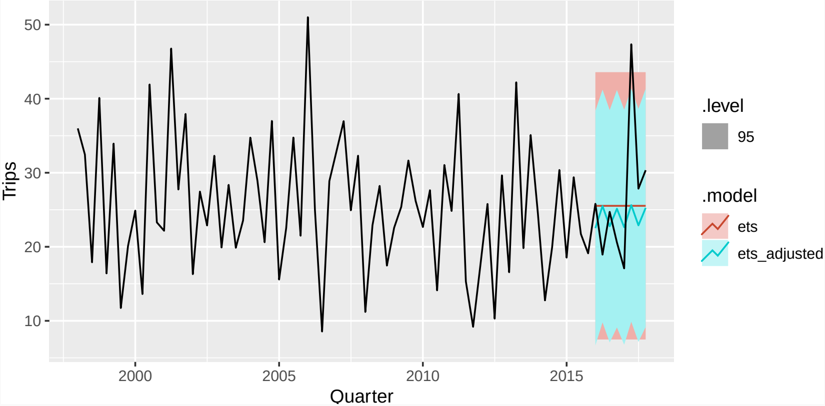
Example: Australian tourism

```r
fc %>%
  filter(is_aggregated(Purpose) & Region == "Snowy Mountains") %>%
  autoplot(tourism_agg, level = 95)
```



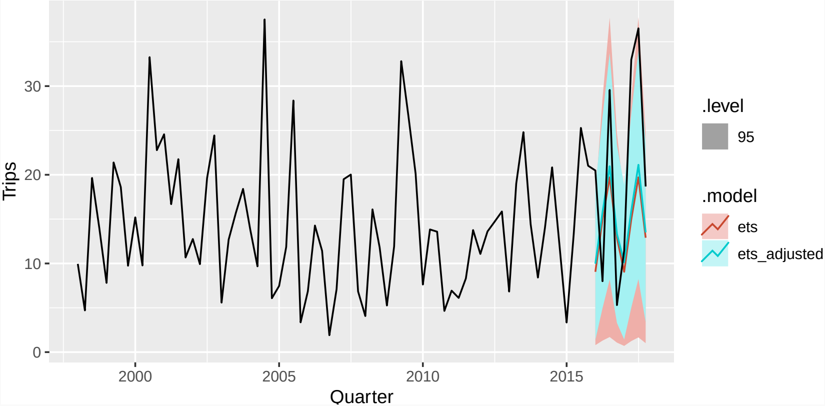
Example: Australian tourism

```
fc %>%
  filter(Purpose == "Holiday" & Region == "Barossa") %>%
  autoplot(tourism_agg, level = 95)
```



Example: Australian tourism

```
fc %>%
  filter(is_aggregated(Purpose) & Region == "MacDonnell") %>%
  autoplot(tourism_agg, level = 95)
```



Example: Australian tourism

```
tourism_agg <- tourism %>%
  aggregate_key(Purpose * (State / Region),
    Trips = sum(Trips))

fc <- tourism_agg %>%
  filter_index(. ~ yearquarter("2015 Q4")) %>%
  model(
    ets = ETS(Trips),
    arima = ARIMA(Trips)
  ) %>%
  reconcile(
    ets_adj = min_trace(ets),
    arima_adj = min_trace(arima),
  ) %>%
  forecast(h = "2 years")
```

Forecast evaluation

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

```
## # A tibble: 1,700 x 12
##    .model Purpose   State      Region     .type    ME  RMSE
##    <chr>  <chr>     <chr>      <chr>      <chr> <dbl> <dbl>
##  1 arima  Business  ACT      ~ Canberra ~ Test  35.9   45.7
##  2 arima  Business  ACT      ~ <aggregat~ Test  35.9   45.7
##  3 arima  Business  New South~ Blue Moun~ Test   1.93  10.6
##  4 arima  Business  New South~ Capital C~ Test   8.08  15.6
##  5 arima  Business  New South~ Central C~ Test  10.0   14.5
##  6 arima  Business  New South~ Central N~ Test  17.7   31.9
##  7 arima  Business  New South~ Hunter   ~ Test  35.3   43.9
##  8 arima  Business  New South~ New Engla~ Test  23.1   31.8
##  9 arima  Business  New South~ North Coa~ Test  24.8   40.1
## 10 arima  Business  New South~ Outback N~ Test   6.87  11.0
## # ... with 1,690 more rows, and 5 more variables:
## #   MAE <dbl>, MPE <dbl>, MAPE <dbl>, MASE <dbl>,
```

Forecast evaluation

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

```
## # A tibble: 4 x 2
##   .model     MASE
##   <chr>     <dbl>
## 1 ets_adj   0.984
## 2 arima_adj 1.01
## 3 ets       1.04
## 4 arima     1.09
```

Outline

Lab Session 10

- Create the hierarchal/grouped series of the daily A&E data by gender and type of injury
- Use forecast reconciliation with using ETS
- Does the reconciliation improve the forecast accuracy?