The geometry of forecast reconciliation

Rob J Hyndman

28 August 2020

Outline

Outline

Australian Pharmaceutical Benefits Scheme

PBS sales

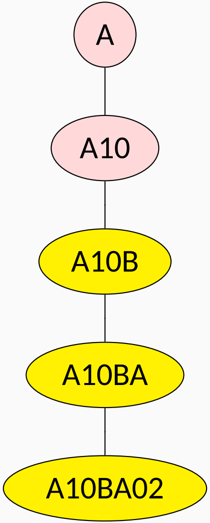
ATC drug classification

A	Alimentary tract and metabolism
B	Blood and blood forming organs
C	Cardiovascular system
D	Dermatologicals
G	Genito-urinary system and sex hormones
H	Systemic hormonal preparations, excluding sex hormones and insulins
J	Anti-infectives for systemic use
L	Antineoplastic and immunomodulating agents
M	Musculo-skeletal system
N	Nervous system
P	Antiparasitic products, insecticides and repellents
R	Respiratory system
S	Sensory organs

ATC drug classification

ATC1: 14 classes

ATC2: 84 classes



Alimentary tract and metabolism

Drugs used in diabetes

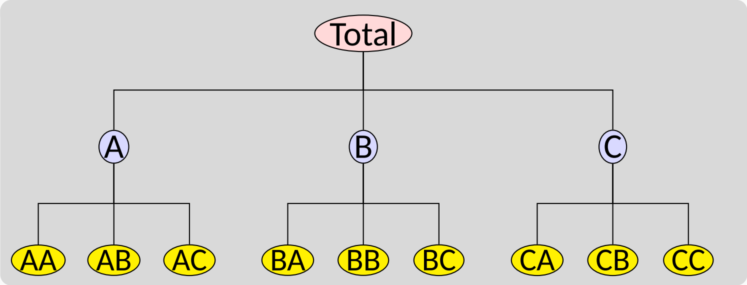
Blood glucose lowering drugs

Biguanides

Metformin

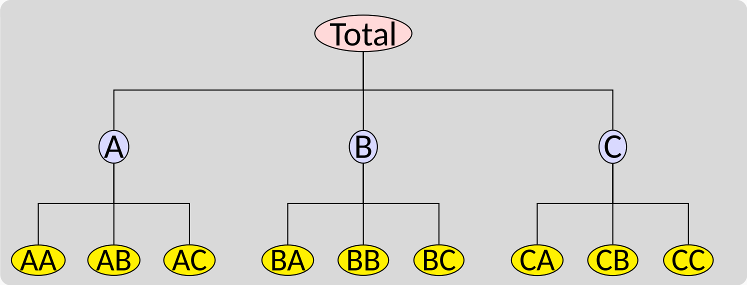
Hierarchical time series

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



Hierarchical time series

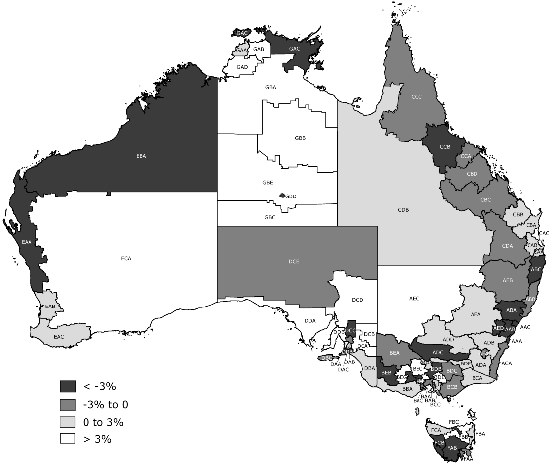
A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.



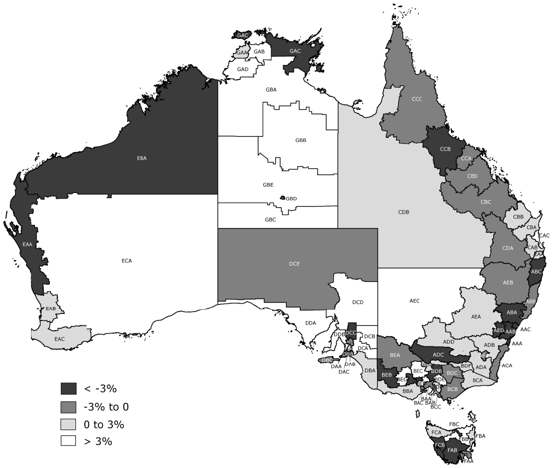
Examples

- PBS sales by ATC groups
- Tourism demand by states, zones, regions

Australian tourism



Australian tourism



- Quarterly data on visitor night from 1998:Q1 – 2013:Q4
- From *National Visitor Survey*, annual interviews of 120,000 Australians aged 15+.
- 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

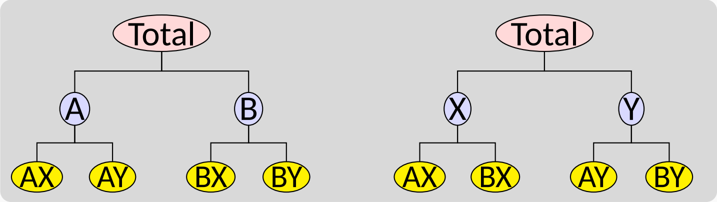
Australian tourism

```
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.
```

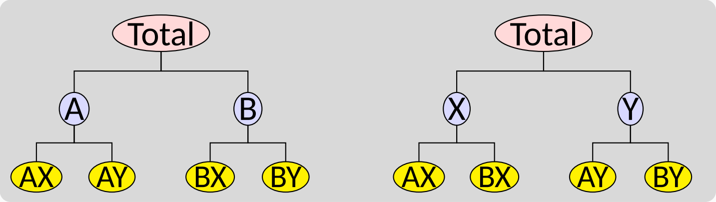
Grouped time series

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

Outline

The problem

1. How to forecast time series at all nodes such that the forecasts are **coherent**?
2. Can we exploit relationships between the series to improve the forecasts?

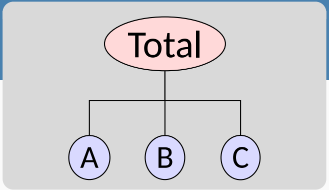
The problem

1. How to forecast time series at all nodes such that the forecasts are **coherent**?
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 are coherent using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).

Hierarchical and grouped time series



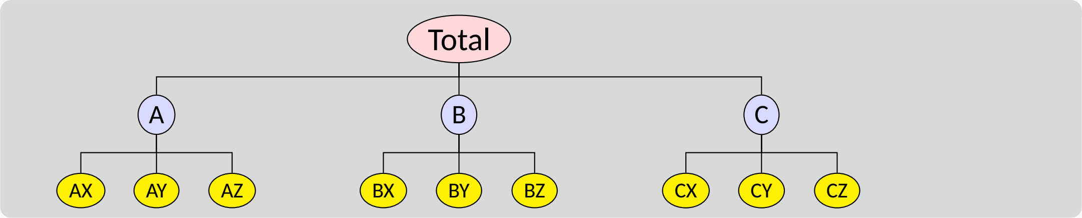
Every collection of time series with linear constraints can be written as

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

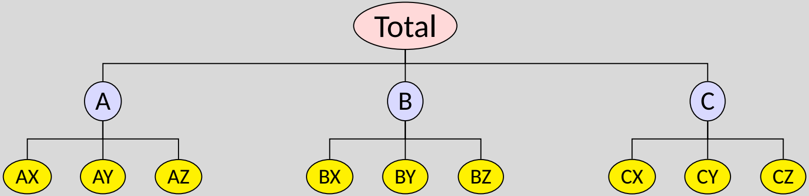
- $\boldsymbol{y}_t$ = vector of all series at time t
- y_t = aggregate of all series at time t.
- $y_{X,t}$ = value of series X at time t.
- $\boldsymbol{b}_t$ = vector of most disaggregated series at time t
- $\boldsymbol{S}$ = "summing matrix" containing the linear constraints.

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

Hierarchical time series

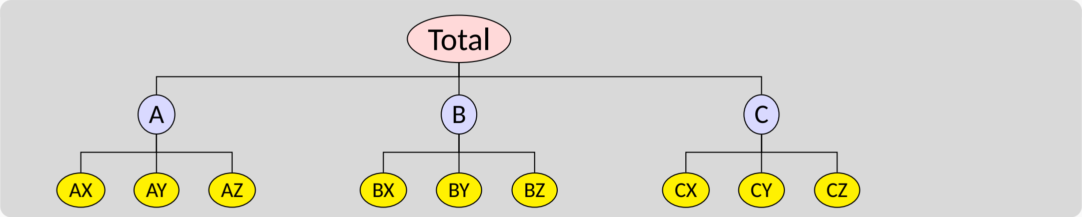


Hierarchical time series



$$
\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix}
$$

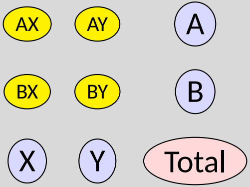
Hierarchical time series



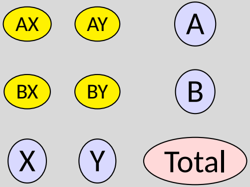
$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix}$$

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

Grouped data

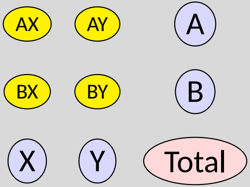


Grouped data



$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{X,t} \\ y_{Y,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix}$$

Grouped data



$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{X,t} \\ y_{Y,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix}$$

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

Forecasting notation

Base forecasts

Let $\hat{\boldsymbol{y}}_{t+h|t}$ be vector of initial h-step forecasts, made at time t, stacked in same order as $\boldsymbol{y}_t$. *(In general, they will not be coherent.)*

Forecasting notation

Base forecasts

Let $\hat{\boldsymbol{y}}_{t+h|t}$ be vector of initial h-step forecasts, made at time t, stacked in same order as $\boldsymbol{y}_t$. *(In general, they will not be coherent.)*

Reconciled forecasts

Let $\tilde{\boldsymbol{y}}_{t+h|t} = \psi(\hat{\boldsymbol{y}}_{t+h|t})$ be a vector of h-step forecasts which are coherent. *Also stacked in the same order as* $\boldsymbol{y}_t$*.)*

Geometric intuition

Coherent subspace

m-dimensional linear subspace $\mathfrak{s} \subset \mathbb{R}^n$ for which linear constraints hold for all $\boldsymbol{y} \in \mathfrak{s}$.

Hierarchical Time Series

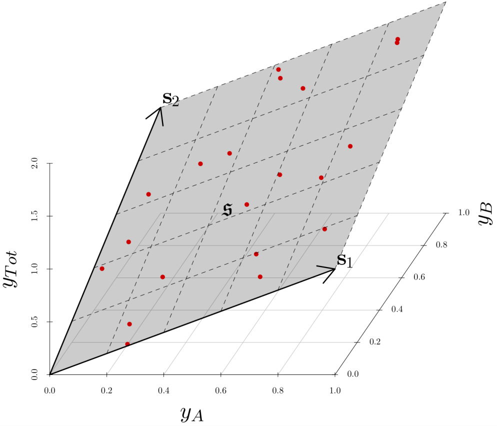
An n-dimensional multivariate time series such that $\boldsymbol{y}_t \in \mathfrak{s} \quad \forall t$.

Coherent Point Forecasts

$\tilde{\boldsymbol{y}}_{t+h|t}$ is *coherent* if $\tilde{\boldsymbol{y}}_{t+h|t} \in \mathfrak{s}$.

Reconciled forecasts

Let ψ be a mapping, $\psi : \mathbb{R}^n \to \mathfrak{s}$. $\tilde{\boldsymbol{y}}_{t+h|t} = \psi(\hat{\boldsymbol{y}}_{t+h|t})$ "reconciles" a base forecast $\hat{\boldsymbol{y}}_{t+h|t}$.



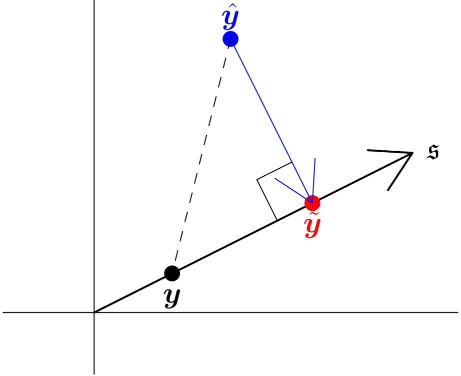
$y_{Tot} = y_A + y_B$

Linear reconciliation

If ψ is a linear function and **G** is some matrix, then

$$\tilde{\boldsymbol{y}}_{t+h|t} = \boldsymbol{S}\boldsymbol{G}\hat{\boldsymbol{y}}_{t+h|t}$$

- **G** extracts and combines base forecasts $\hat{\boldsymbol{y}}_T(h)$ to get bottom-level forecasts.
- $\boldsymbol{S}$ creates linear combinations.



Distance reducing property

If $\boldsymbol{SG}$ is an orthogonal projection (in the Euclidean sense) onto $\mathfrak{s}$ then:

$$\|\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}_{t+h|t}\| \leq \|\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_{t+h|t}\|.$$

- Distance reduction holds for any realisation and any forecast.
- Other measures of forecast accuracy may be worse.
- Not necessarily the optimal reconciliation.

General properties: bias and variance

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

General properties: bias and variance

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

Bias

Reconciled forecasts are unbiased iff $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} = \boldsymbol{S}$.

General properties: bias and variance

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

Bias

Reconciled forecasts are unbiased iff $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} = \boldsymbol{S}$.

Variance

Let error variance of h-step base forecasts $\hat{\boldsymbol{y}}_n(h)$ be

$$\Sigma_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_T(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

Then the error variance of the corresponding reconciled forecasts is

$$\text{Var}[\boldsymbol{y}_{T+h} - \tilde{\boldsymbol{y}}_T(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \boldsymbol{S}\boldsymbol{G}\Sigma_h\boldsymbol{G}'\boldsymbol{S}'$$

Optimal forecast reconciliation

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

Optimal forecast reconciliation

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

Minimum trace reconciliation

If $\boldsymbol{P}$ satisfies $\boldsymbol{SGS} = \boldsymbol{S}$, then

$\min_{\boldsymbol{G}} = \text{trace}[\boldsymbol{SG}\Sigma_h\boldsymbol{G}'\boldsymbol{S}']$ has solution $\boldsymbol{G} = (\boldsymbol{S}'\Sigma_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{-1}$.

Optimal forecast reconciliation

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

Minimum trace reconciliation

If $\boldsymbol{P}$ satisfies $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} = \boldsymbol{S}$, then

$\min_{\boldsymbol{G}} = \text{trace}[\boldsymbol{S}\boldsymbol{G}\Sigma_h\boldsymbol{G}'\boldsymbol{S}']$ has solution $\boldsymbol{G} = (\boldsymbol{S}'\Sigma_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{-1}$.

Reconciled forecasts $\quad \tilde{\boldsymbol{y}}_T(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{-1}\hat{\boldsymbol{y}}_T(h) \quad$ Base forecasts

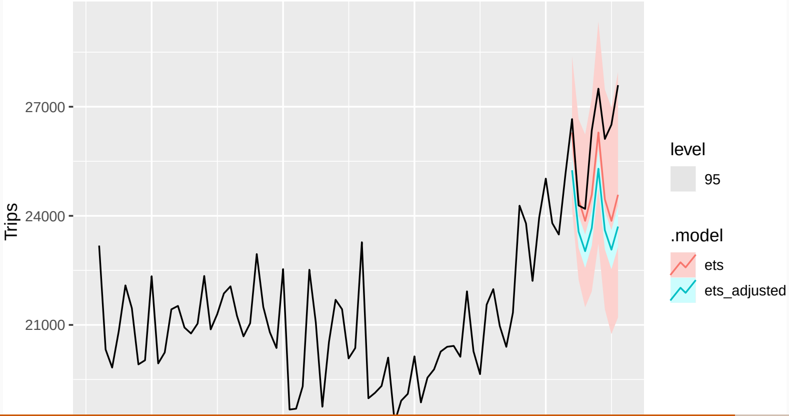
Outline

Example: Australian tourism

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

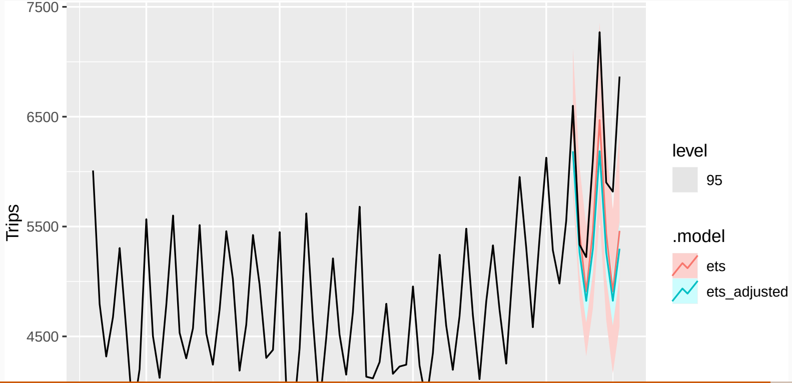
Example: Australian tourism

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



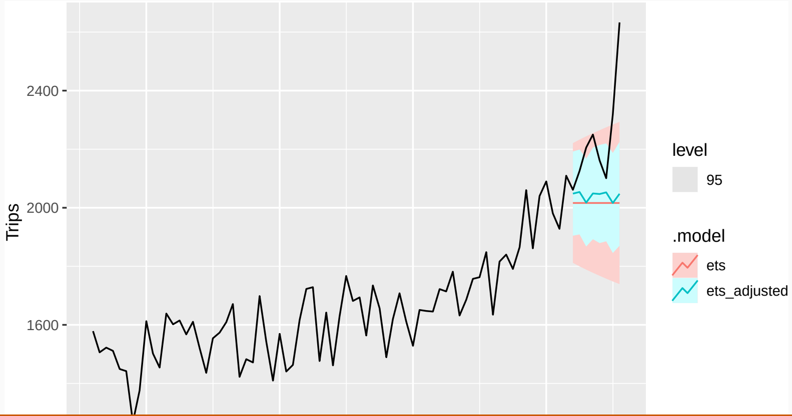
Example: Australian tourism

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



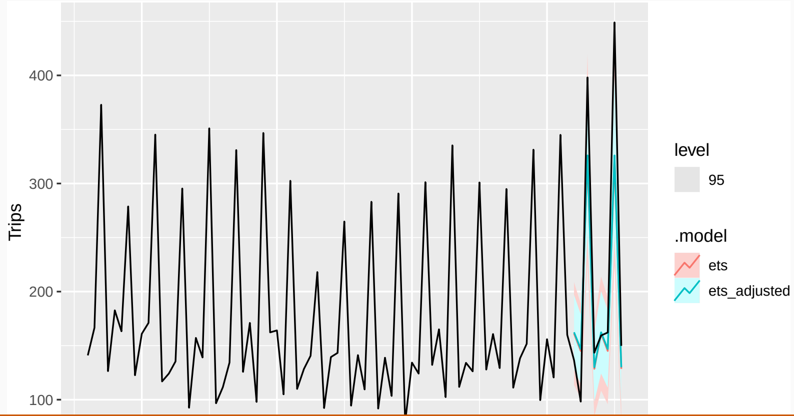
Example: Australian tourism

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



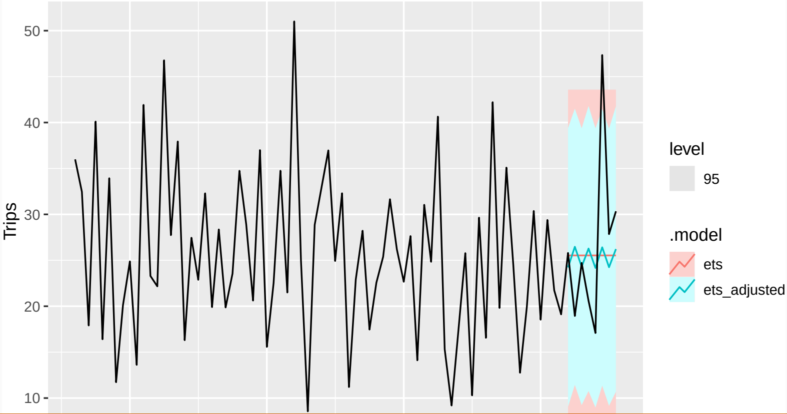
Example: Australian tourism

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



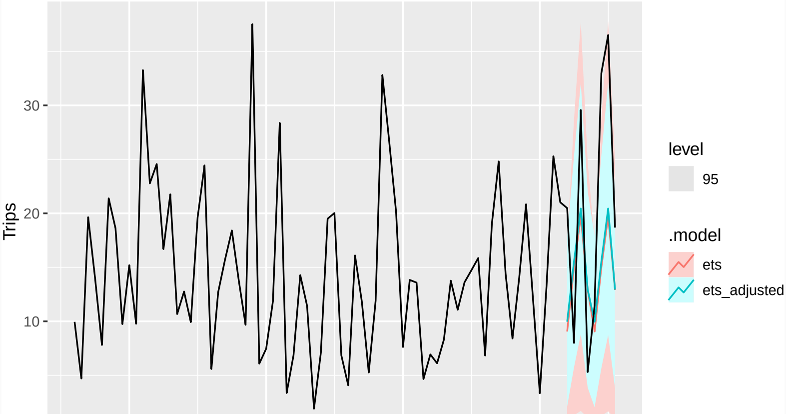
Example: Australian tourism

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



Example: Australian tourism

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



Example: Australian tourism

```r
fc <- tourism_agg %>%
  filter(Quarter <= yearquarter("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 evaluation

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

```
## # A tibble: 2,550 x 12
##    .model Purpose    State       Region      .type     ME
##    <chr>  <chr>      <chr>       <chr>       <chr>  <dbl>
##  1 arima  Business   <aggregat~ <aggregat~ Test   685.
##  2 arima  Business   South Aus~ <aggregat~ Test    49.9
##  3 arima  Business   Northern ~ <aggregat~ Test    22.2
##  4 arima  Business   Western A~ <aggregat~ Test  -138.
##  5 arima  Business   Victoria ~ <aggregat~ Test   232.
##  6 arima  Business   New South~ <aggregat~ Test   153.
##  7 arima  Business   Queenslan~ <aggregat~ Test    81.8
##  8 arima  Business   ACT       ~ <aggregat~ Test    35.9
##  9 arima  Business   Tasmania ~ <aggregat~ Test    28.8
## 10 arima  Business   South Aus~ Adelaide    Test    20.8
## # ... with 2,540 more rows, and 6 more variables:
```

Forecast evaluation

```
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
```

Creating aggregates

```r
PBS %>%
  aggregate_key(ATC1 / ATC2, Scripts = sum(Scripts)) %>%
  filter(Month == yearmonth("1991 Jul")) %>%
  print(n = 18)
```

```
## # A tsibble: 98 x 4 [1M]
## # Key:       ATC1, ATC2 [98]
##       Month ATC1         ATC2         Scripts
##       <mth> <chr>        <chr>          <dbl>
##  1 1991 Jul <aggregated> <aggregated> 8090395
##  2 1991 Jul A            <aggregated>  799025
##  3 1991 Jul B            <aggregated>  109227
##  4 1991 Jul C            <aggregated> 1794995
##  5 1991 Jul D            <aggregated>  299779
##  6 1991 Jul G            <aggregated>  300931
##  7 1991 Jul H            <aggregated>  112114
##  8 1991 Jul J            <aggregated> 1151681
##  9 1991 Jul L            <aggregated>   24580
## 10 1991 Jul M            <aggregated>  562956
## 11 1991 Jul N            <aggregated> 1546023
## 12 1991 Jul P            <aggregated>   47661
## 13 1991 Jul R            <aggregated>  859273
## 14 1991 Jul S            <aggregated>  391639
## 15 1991 Jul V            <aggregated>   38705
## 16 1991 Jul Z            <aggregated>   51806
```

Creating aggregates

```
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>        <chr>            <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> ACT             ~ <aggregated>   551.
##  7 1998 Q1 <aggregated> New South Wales~ <aggregated>  8040.
##  8 1998 Q1 <aggregated> Northern Territ~ <aggregated>   181.
##  9 1998 Q1 <aggregated> Queensland      ~ <aggregated>  4041.
## 10 1998 Q1 <aggregated> South Australia~ <aggregated>  1735.
## 11 1998 Q1 <aggregated> Tasmania        ~ <aggregated>   982.
## 12 1998 Q1 <aggregated> Victoria        ~ <aggregated>  6010.
## 13 1998 Q1 <aggregated> Western Austral~ <aggregated>  1641.
```

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.

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
##    <chr>      <chr>       <chr>       <chr>    <qtr>
##  1 Business   <aggregat~ <aggregat~ ets     2018 Q1
##  2 Business   <aggregat~ <aggregat~ ets     2018 Q2
##  3 Business   <aggregat~ <aggregat~ ets_a~ 2018 Q1
##  4 Business   <aggregat~ <aggregat~ ets_a~ 2018 Q2
##  5 Business   South Aus~ <aggregat~ ets     2018 Q1
##  6 Business   South Aus~ <aggregat~ ets     2018 Q2
##  7 Business   South Aus~ <aggregat~ ets_a~ 2018 Q1
##  8 Business   South Aus~ <aggregat~ ets_a~ 2018 Q2
```