Forecast reconciliation

1. Hierarchical time series & forecast reconciliation

Rob J Hyndman

MONASH University

robjhyndman.com/fr2023

Photo by Edvard Alexander Rølvaag on

Outline

1. Hierarchical time series data
2. Hierarchical forecasting using single-level approaches
3. Linear forecast reconciliation
4. Example: Australian tourism
5. Fast computational tricks

Outline

1 Hierarchical time series data

2 Hierarchical forecasting using single-level approaches

3 Linear forecast reconciliation

4 Example: Australian tourism

5 Fast computational tricks

Labour market participation

Australia & New Zealand Standard Classification of Occupations

- 8 major groups
  - 43 sub-major groups
    - 97 minor groups
      - 359 unit groups
        - 1023 occupations

Labour market participation

Australia & New Zealand Standard Classification of Occupations

- 8 major groups
  - 43 sub-major groups
    - 97 minor groups
      - 359 unit groups
        - 1023 occupations

Example: statistician

2 Professionals
22 Business, Human Resource and Marketing Professionals
224 Information and Organisation Professionals
2241 Actuaries, Mathematicians and Statisticians
224113 Statistician

PBS sales

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
V	Various

PBS sales

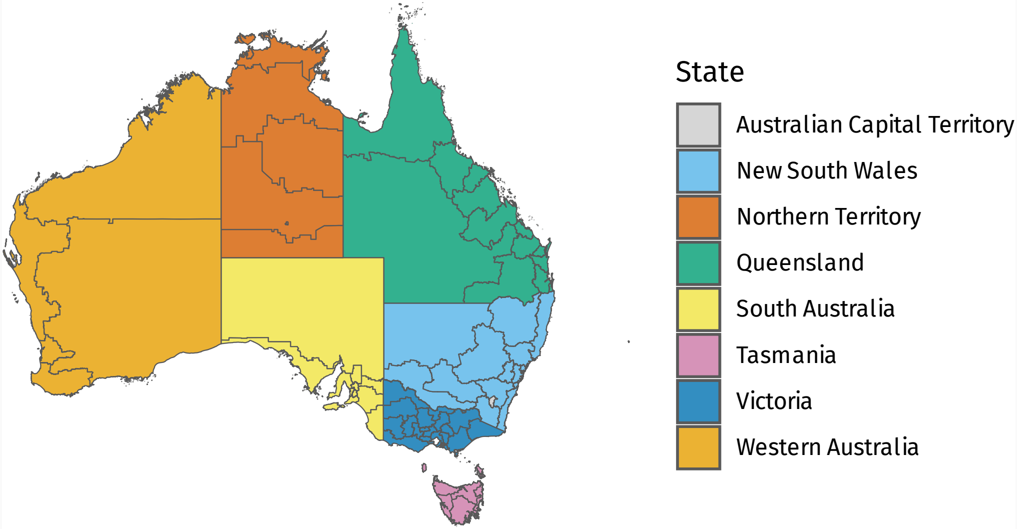
ATC drug classification

14 classes	A	Alimentary tract and metabolism
84 classes	A10	Drugs used in diabetes
	A10B	Blood glucose lowering drugs
	A10BA	Biguanides
	A10BA02	Metformin

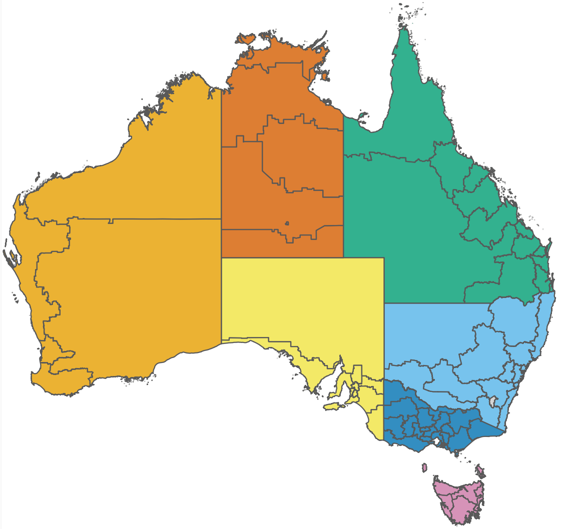
Spectacle sales

- Monthly UK sales data from 2000 – 2014
- Provided by a large spectacle manufacturer
- Split by brand (26), gender (3), price range (6), materials (4), and stores (600)
- About 1 million bottom-level series

Australian tourism regions

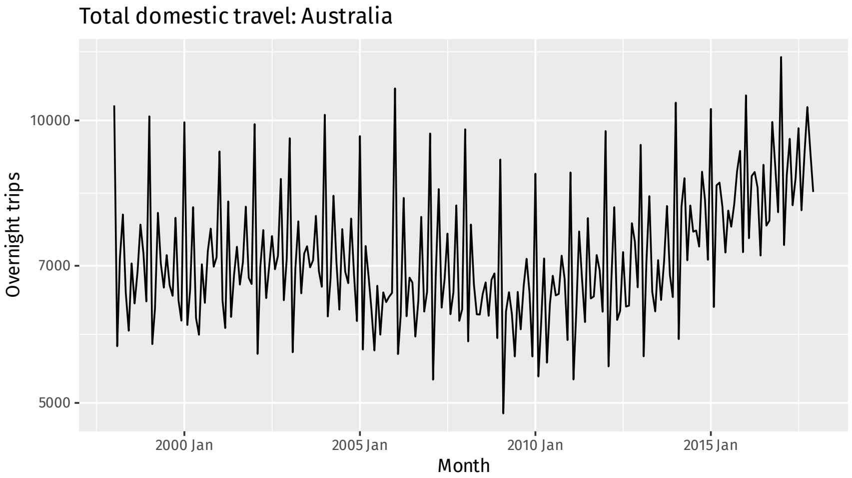


Australian tourism regions

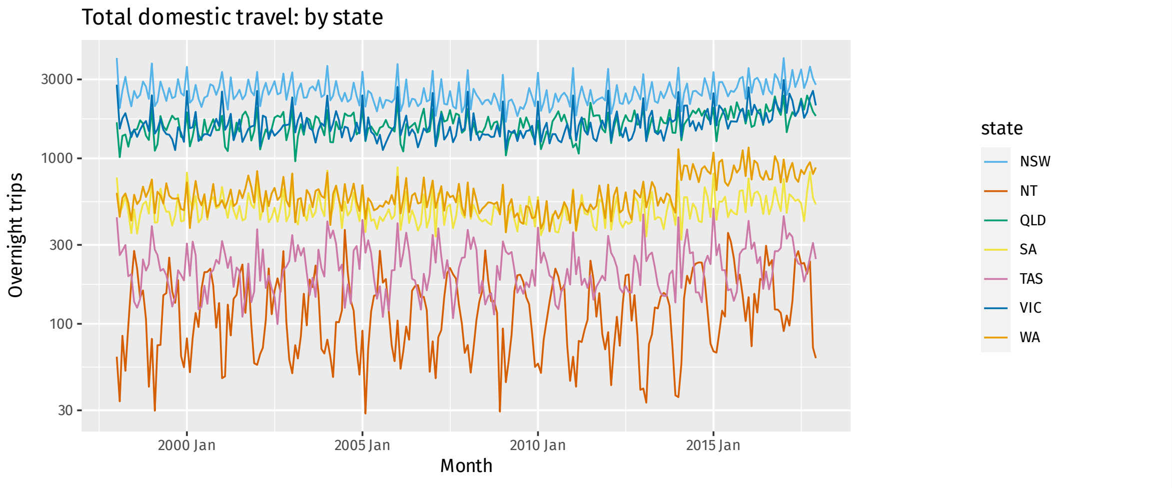


- Monthly data on visitor night from 1998 – 2017
- From *National Visitor Survey*, annual interviews of 120,000 Australians aged 15+.
- Geographical hierarchy split by
  - 7 states
  - 27 zones
  - 75 regions

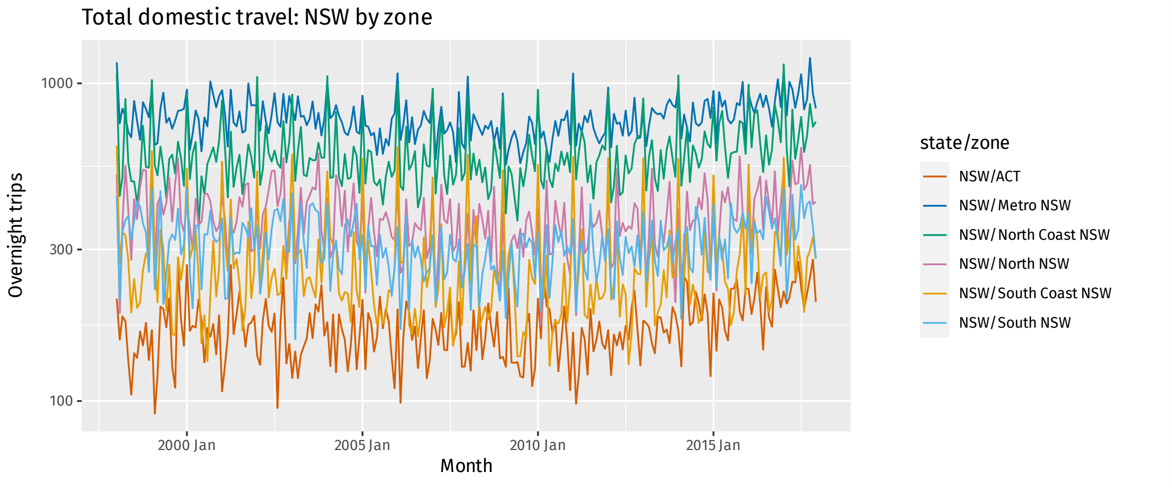
Australian tourism data



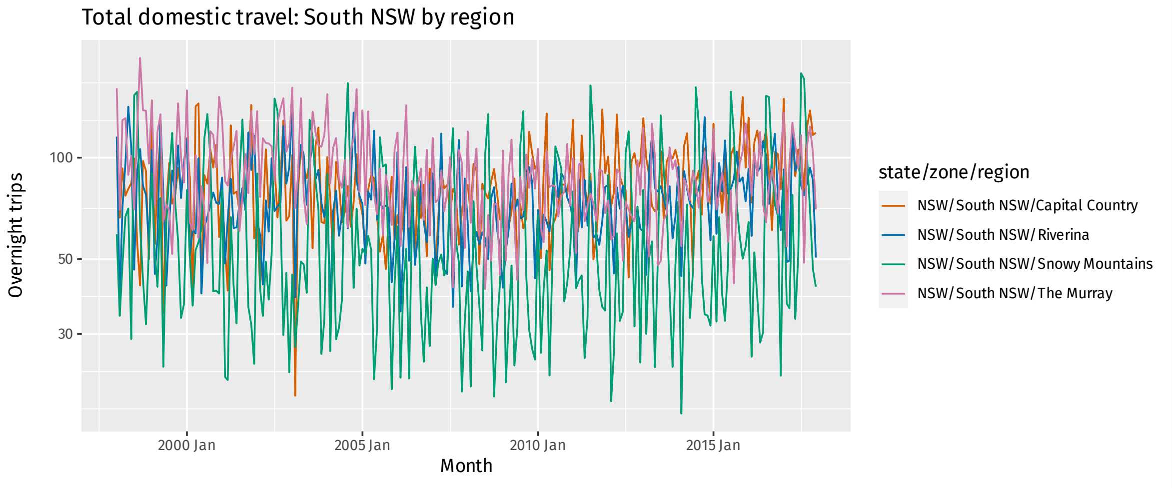
Australian tourism data



Australian tourism data

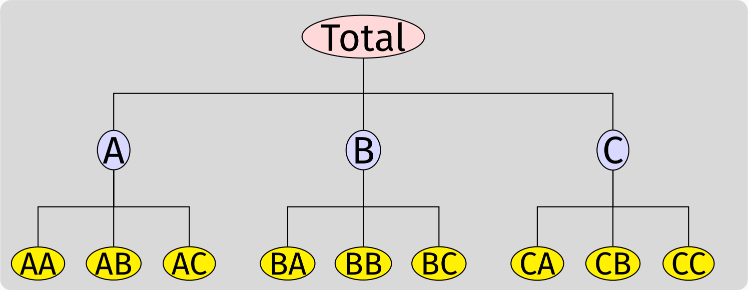


Australian tourism data



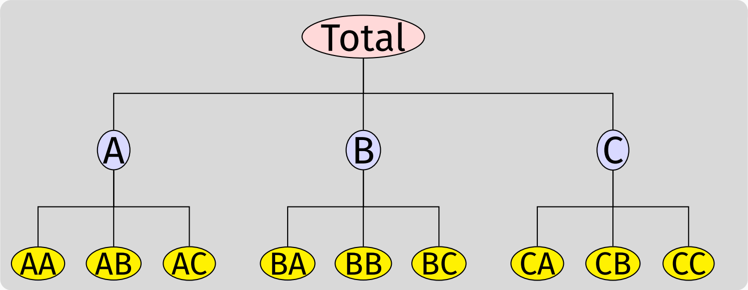
Hierarchical time series

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.

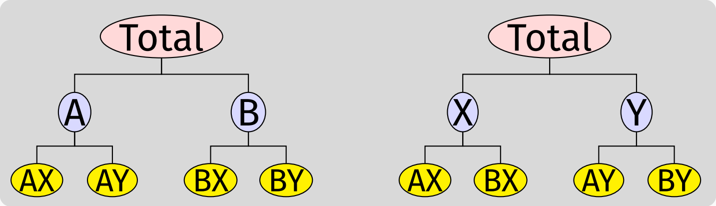


Examples

- Tourism by state and region
- Retail sales by product groups, sub groups, and SKUs

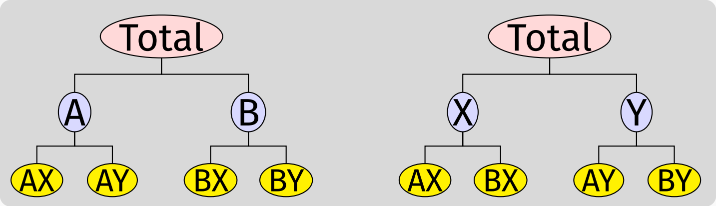
Grouped time series

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

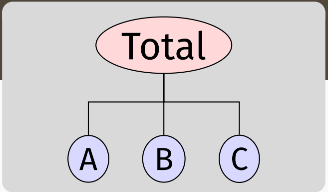
A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.



Examples

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

Hierarchical and grouped time series



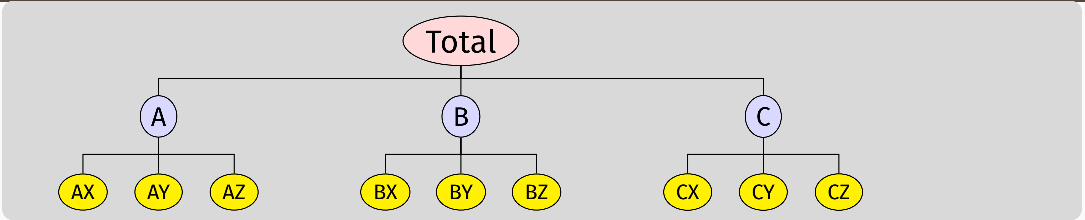
Every collection of time series with linear constraints can be written as

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

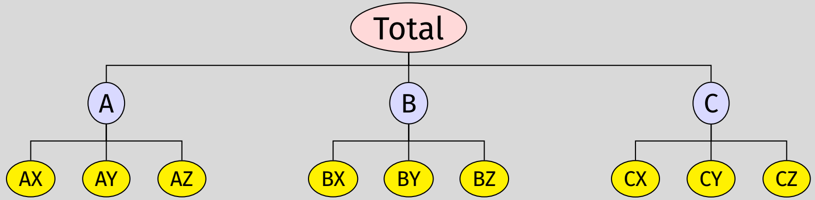
- $\boldsymbol{y}_t$ = vector of all series at time t
- y_t = aggregate of all series at time t.
- $y_{X,t}$ = value of series X at time t.
- $\boldsymbol{b}_t$ = vector of most disaggregated series at time t
- $\boldsymbol{S}$ = "summing matrix" containing the linear constraints.

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

Hierarchical time series

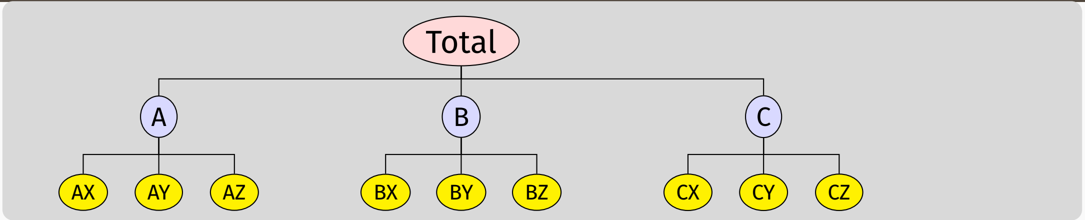


Hierarchical time series



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

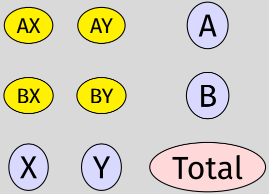
Hierarchical time series



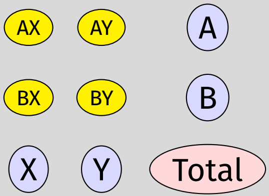
$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1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

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

Grouped data

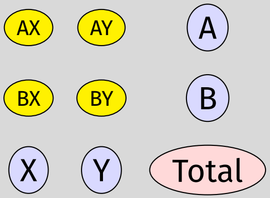


Grouped data



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

Grouped data



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

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

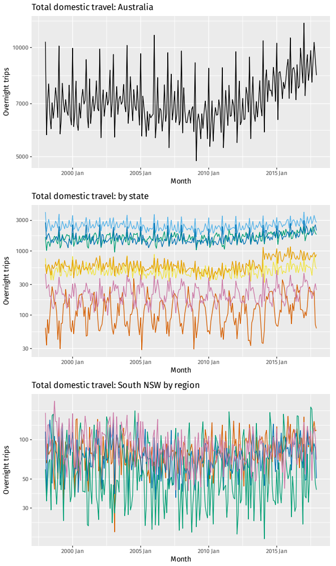
The hierarchical forecasting problem

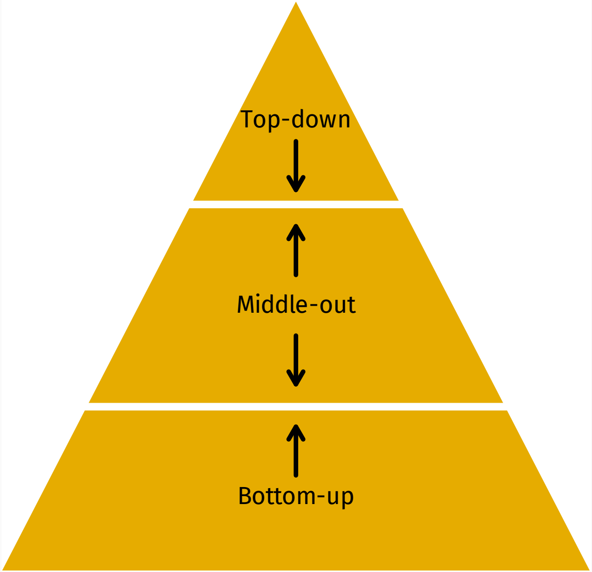
- We want forecasts at all levels of aggregation.
- If we model and forecast each series independently, the forecasts will almost certainly not add up.
- We need to impose constraints on the forecasts to ensure they are “coherent”.
- We need to do this in a way that is computationally efficient.

Outline

1. Hierarchical time series data
2. Hierarchical forecasting using single-level approaches
3. Linear forecast reconciliation
4. Example: Australian tourism
5. Fast computational tricks

Hierarchical forecasting 20 years ago





Top-down forecasting

Advantages

- Works well in presence of low counts.
- Single forecasting model easy to build
- Provides reliable forecasts for aggregate levels.

Disadvantages

- Loss of information, especially individual series dynamics.
- Distribution of forecasts to lower levels can be difficult
- No prediction intervals

Bottom-up forecasting

Advantages

- No loss of information.
- Better captures dynamics of individual series.

Disadvantages

- Large number of series to be forecast.
- Constructing forecasting models is harder because of noisy data at bottom level.
- No prediction intervals

Forecasting notation

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 “add up”.)

Forecasting notation

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 "add up".)

Coherent linear forecasts are of the form:

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

for some matrix $\boldsymbol{G}$.

Forecasting notation

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 "add up".)

Coherent linear forecasts are of the form:

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

for some matrix $\boldsymbol{G}$.

- $\boldsymbol{G}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_{T+h|T}$ to get bottom-level forecasts.
- $\boldsymbol{S}$ adds them up

Bottom-up forecasting

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

Bottom-up forecasts are obtained using

$$\boldsymbol{G} = [\boldsymbol{0} \mid \boldsymbol{I}],$$

where $\boldsymbol{0}$ is null matrix and $\boldsymbol{I}$ is identity matrix.

- $\boldsymbol{G}$ matrix extracts only bottom-level forecasts from $\hat{\boldsymbol{y}}_{T+h|T}$
- $\boldsymbol{S}$ adds them up to give the bottom-up forecasts.

Top-down forecasting

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

Top-down forecasts are obtained using

$$\boldsymbol{G} = [\boldsymbol{p} \mid \boldsymbol{0}]$$

where $\boldsymbol{p} = [p_1, p_2, \dots, p_{n_b}]'$ and $\sum_{k=1}^{n_b} p_k = 1$.

- $\boldsymbol{G}$ distributes forecasts of aggregate to lowest level series.
- Different methods of top-down forecasting lead to different proportionality vectors $\boldsymbol{p}$.

Properties of single-level methods

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

Mean

$$\begin{aligned}
\mathrm{E}[\tilde{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T] &= \boldsymbol{S}\boldsymbol{G}\mathrm{E}[\hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T] \\
&= \boldsymbol{S}\mathrm{E}[\boldsymbol{b}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]
\end{aligned}$$

provided $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} = \boldsymbol{S}$ and $\mathrm{E}[\hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T] = \boldsymbol{S}\mathrm{E}[\boldsymbol{b}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]$. i.e., forecasts $\tilde{\boldsymbol{y}}_{T+h|T}$ are unbiased iff base forecasts $\hat{\boldsymbol{y}}_{T+h|T}$ are unbiased and $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} = \boldsymbol{S}$.

Properties of single-level methods

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

Mean

$$\begin{aligned}
\mathrm{E}[\tilde{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T] &= \boldsymbol{S}\boldsymbol{G}\mathrm{E}[\hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T] \\
&= \boldsymbol{S}\mathrm{E}[\boldsymbol{b}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]
\end{aligned}$$

provided $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} = \boldsymbol{S}$ and $\mathrm{E}[\hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T] = \boldsymbol{S}\mathrm{E}[\boldsymbol{b}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]$. i.e., forecasts $\tilde{\boldsymbol{y}}_{T+h|T}$ are unbiased iff base forecasts $\hat{\boldsymbol{y}}_{T+h|T}$ are unbiased and $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} = \boldsymbol{S}$.

- $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} = \boldsymbol{S}$ for bottom-up method
- $\boldsymbol{S}\boldsymbol{G}\boldsymbol{S} \neq \boldsymbol{S}$ for any top-down or middle-out method.

Properties of single-level methods

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

Variance

$$\begin{aligned}\boldsymbol{V}_h &= \text{Var}[\boldsymbol{y}_{T+h} - \tilde{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T] \\ &= \boldsymbol{S}\boldsymbol{G}\boldsymbol{W}_h\boldsymbol{G}'\boldsymbol{S}'\end{aligned}$$

where $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]$

Properties of single-level methods

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

Variance

$$\begin{aligned}\boldsymbol{V}_h &= \text{Var}[\boldsymbol{y}_{T+h} - \tilde{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1,\dots,\boldsymbol{y}_T] \\ &= \boldsymbol{S}\boldsymbol{G}\boldsymbol{W}_h\boldsymbol{G}'\boldsymbol{S}'\end{aligned}$$

where $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1,\dots,\boldsymbol{y}_T]$

- $\boldsymbol{W}_h$ is hard to estimate for $h > 1$.
- This suggests we should choose $\boldsymbol{G}$ to minimise $\boldsymbol{V}_h$.

Outline

1. Hierarchical time series data
2. Hierarchical forecasting using single-level approaches
3. Linear forecast reconciliation
4. Example: Australian tourism
5. Fast computational tricks

Early history of forecast reconciliation

History

2001: Idea to use all available series to forecast Australia's labour market by occupation.

2004: PhD student Roman Ahmed begins, co-supervised with George Athanasopoulos.

2006: Presentation at ISF, Santander.

2007: Pre-print of "Optimal combination forecasts for hierarchical time series".

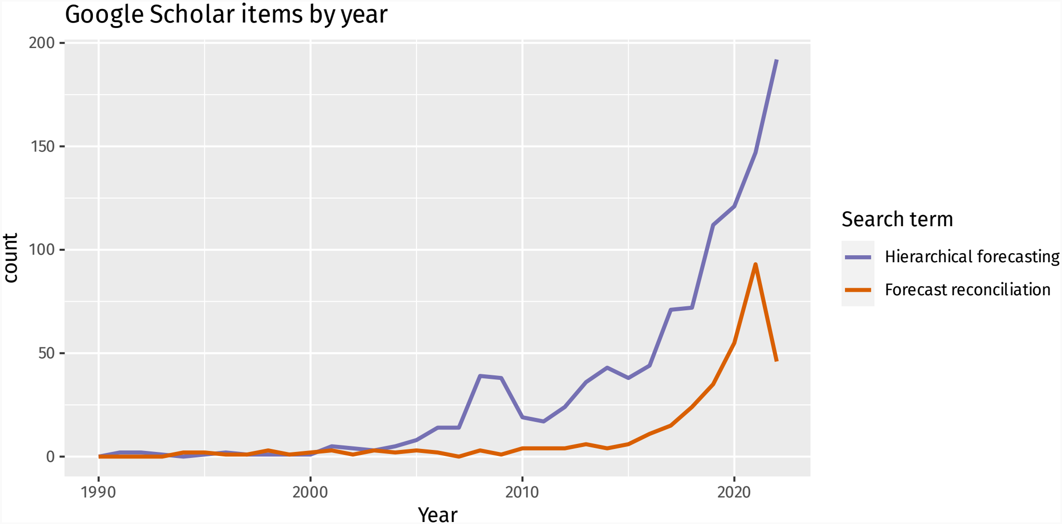
2009: Application to Australian tourism published in IJF.

2010: First version of hts package on CRAN.

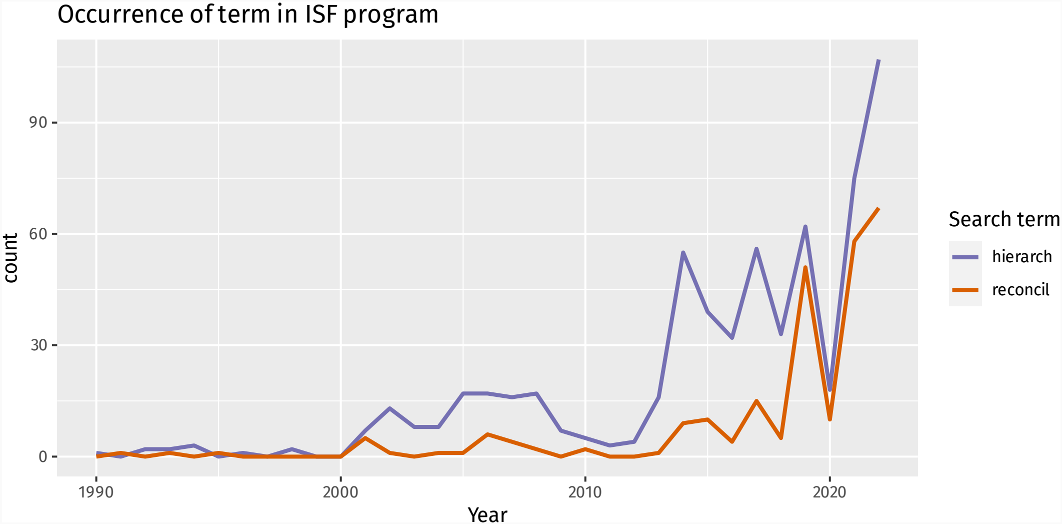
2011: "Optimal combination forecasts for hierarchical time series" appears in CSDA.

2019: "Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization"' appears in JASA.

Forecast reconciliation research



Forecast reconciliation research



Linear forecast reconciliation

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

- $\boldsymbol{G}$ combines base forecasts $\hat{\boldsymbol{y}}_{T+h|T}$ to get bottom-level forecasts.
- $\boldsymbol{S}$ creates linear combinations.
- Bottom-up and top-down can be seen as special cases.
- $\boldsymbol{SG}$ is a projection matrix iff $\boldsymbol{SGS}' = \boldsymbol{S}$.
- Reconciled forecasts are unbiased iff base forecasts are unbiased and $\boldsymbol{SG}$ is a projection.
- $\boldsymbol{V}_h = \text{Var}[\boldsymbol{y}_{T+h} - \tilde{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T] = \boldsymbol{SGW}_h\boldsymbol{G}'\boldsymbol{S}'$
- How to choose $\boldsymbol{G}$ to create optimal forecasts?

Minimum trace reconciliation

Minimum trace (MinT) reconciliation

If $\boldsymbol{SG}$ is a projection, then the trace of $\boldsymbol{V}_h$ is minimized when

$$\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}$$

$$\tilde{\boldsymbol{y}}_{T+h|T} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}\hat{\boldsymbol{y}}_{T+h|T}$$

Reconciled forecasts

Base forecasts

- Trace of $\boldsymbol{V}_h$ is sum of forecast variances.
- MinT solution is L_2 optimal amongst linear unbiased forecasts.
- How to estimate $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]$?

Linear projections

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

Reconciliation method	$\boldsymbol{G}$
OLS	$(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$
WLS(var)	$(\boldsymbol{S}'\Lambda_v\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda_v$
WLS(struct)	$(\boldsymbol{S}'\Lambda_s\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda_s$
MinT(sample)	$(\boldsymbol{S}'\hat{\boldsymbol{W}}_{\text{sam}}^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{W}}_{\text{sam}}^{-1}$
MinT(shrink)	$(\boldsymbol{S}'\hat{\boldsymbol{W}}_{\text{shr}}^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{W}}_{\text{shr}}^{-1}$

These approximate MinT by assuming $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$.

- $\Lambda_v = \text{diag}(\boldsymbol{W}_1)^{-1}$
- $\Lambda_s = \text{diag}(\boldsymbol{S}\boldsymbol{1})^{-1}$
- $\hat{\boldsymbol{W}}_{\text{sam}}$ is sample estimate of the residual covariance matrix
- $\hat{\boldsymbol{W}}_{\text{shr}}$ is shrinkage estimator $\tau\text{diag}(\hat{\boldsymbol{W}}_{\text{sam}}) + (1-\tau)\hat{\boldsymbol{W}}_{\text{sam}}$ where τ selected optimally.
- Still need a good estimate of $\boldsymbol{W}_h$ for forecast variance.

OLS is not as bad as you might think

- If all base forecasts come from the same model, then forecast errors are coherent.
- So $\hat{\boldsymbol{y}}_{T+h|T} = \boldsymbol{S}\hat{\boldsymbol{b}}_{T+h|T} + \boldsymbol{S}\hat{\varepsilon}_{T+h}$ and

$$\boldsymbol{W}_h = \text{Var}(\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T}) = \boldsymbol{S}\Sigma_h\boldsymbol{S}'$$

where $\Sigma_h = \text{Var}(\hat{\varepsilon}_{T+h})$. So

$$\begin{aligned}\boldsymbol{G} &= \left[\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S}\right]^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}\\ &= \left[\boldsymbol{S}'(\boldsymbol{S}\Sigma_h\boldsymbol{S}')^{-1}\boldsymbol{S}\right]^{-1}\boldsymbol{S}'(\boldsymbol{S}\Sigma_h\boldsymbol{S}')^{-1}\\ &= (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\end{aligned}$$

(Proof: Hyndman et al, *CSDA* 2011)

Outline

1 Hierarchical time series data

2 Hierarchical forecasting using single-level approaches

3 Linear forecast reconciliation

4 Example: Australian tourism

5 Fast computational tricks

Example: Australian tourism

```
tourism
```

```
# A tsibble: 18,000 x 5 [1M]
# Key:       state, zone, region [75]
      month state zone      region visitors
      <mth> <chr> <chr>     <chr>     <dbl>
 1 1998 Jan NSW   Metro NSW Sydney     926.
 2 1998 Feb NSW   Metro NSW Sydney     647.
 3 1998 Mar NSW   Metro NSW Sydney     716.
 4 1998 Apr NSW   Metro NSW Sydney     621.
 5 1998 May NSW   Metro NSW Sydney     598.
 6 1998 Jun NSW   Metro NSW Sydney     601.
 7 1998 Jul NSW   Metro NSW Sydney     720.
 8 1998 Aug NSW   Metro NSW Sydney     645.
 9 1998 Sep NSW   Metro NSW Sydney     633.
10 1998 Oct NSW   Metro NSW Sydney     771.
# i 17,990 more rows
```

Example: Australian tourism

```r
tourism_agg <- tourism |>
  aggregate_key(state / zone / region, visitors = sum(visitors))
```

```
# A tsibble: 26,400 x 5 [1M]
# Key:       state, zone, region [110]
      month state        zone         region       visitors
      <mth> <chr*>       <chr*>       <chr*>          <dbl>
 1 1998 Jan <aggregated> <aggregated> <aggregated>   10376.
 2 1998 Feb <aggregated> <aggregated> <aggregated>    5746.
 3 1998 Mar <aggregated> <aggregated> <aggregated>    7129.
 4 1998 Apr <aggregated> <aggregated> <aggregated>    7939.
 5 1998 May <aggregated> <aggregated> <aggregated>    6552.
 6 1998 Jun <aggregated> <aggregated> <aggregated>    5969.
 7 1998 Jul <aggregated> <aggregated> <aggregated>    7041.
 8 1998 Aug <aggregated> <aggregated> <aggregated>    6382.
 9 1998 Sep <aggregated> <aggregated> <aggregated>    6907.
10 1998 Oct <aggregated> <aggregated> <aggregated>    7744.
```

Example: Australian tourism

```r
fit <- tourism_agg |>
  filter(year(month) <= 2015) |>
  model(ets = ETS(visitors))
```

```
# A mable: 110 x 4
# Key:     state, zone, region [110]
   state  zone             region                     ets
   <chr*> <chr*>           <chr*>                 <model>
 1 NSW    ACT              Canberra          <ETS(M,N,A)>
 2 NSW    ACT              <aggregated>      <ETS(M,N,A)>
 3 NSW    Metro NSW        Central Coast     <ETS(M,N,M)>
 4 NSW    Metro NSW        Sydney            <ETS(M,N,A)>
 5 NSW    Metro NSW        <aggregated>      <ETS(M,N,A)>
 6 NSW    North Coast NSW  Hunter            <ETS(M,N,M)>
 7 NSW    North Coast NSW  North Coast NSW   <ETS(M,N,M)>
 8 NSW    North Coast NSW  <aggregated>      <ETS(M,N,M)>
 9 NSW    North NSW        Blue Mountains    <ETS(M,N,A)>
```

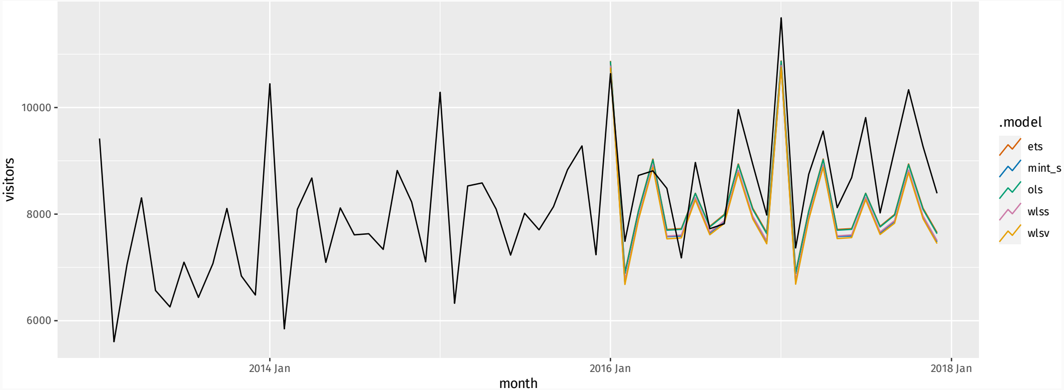
Example: Australian tourism

```r
fc <- fit |>
  reconcile(
    ols = min_trace(ets, method = "ols"),
    wlsv = min_trace(ets, method = "wls_var"),
    wlss = min_trace(ets, method = "wls_struct"),
    # mint_c = min_trace(ets, method="mint_cov"),
    mint_s = min_trace(ets, method = "mint_shrink"),
  ) |>
  forecast(h = "2 years")
```

```
# A fable: 13,200 x 7 [1M]
# Key:     state, zone, region, .model [550]
   state  zone   region   .model     month     visitors .mean
   <chr*> <chr*> <chr*>   <chr>      <mth>       <dist> <dbl>
 1 NSW    ACT    Canberra ets     2016 Jan N(202, 1437)  202.
 2 NSW    ACT    Canberra ets     2016 Feb  N(160, 912)  160.
 3 NSW    ACT    Canberra ets     2016 Mar N(204, 1489)  204.
```

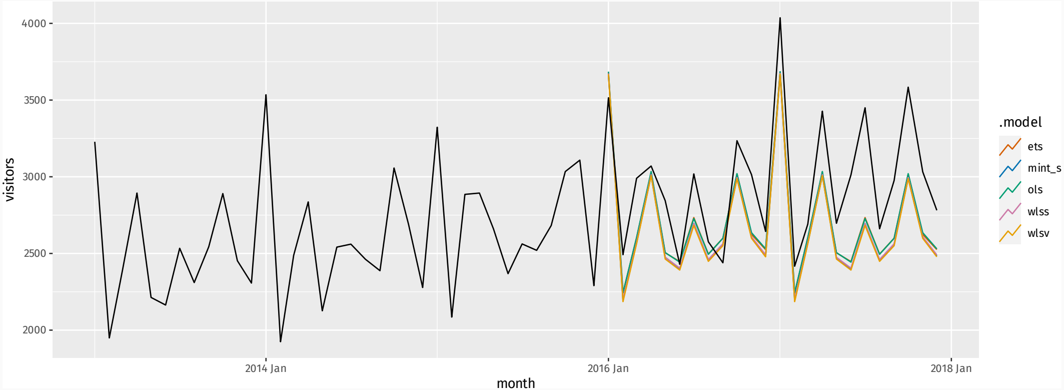
Example: Australian tourism

```r
fc |>
  filter(is_aggregated(state)) |>
  autoplot(filter(tourism_agg, year(month) > 2012), level = NULL)
```



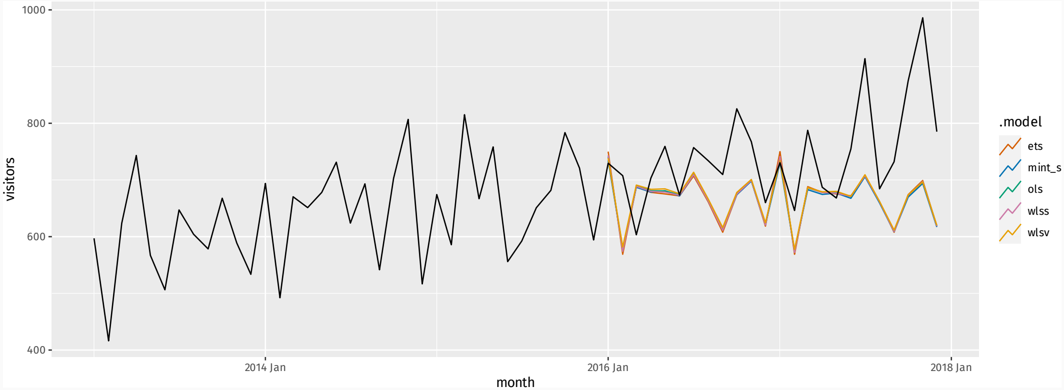
Example: Australian tourism

```
fc |>
  filter(state == "NSW" & is_aggregated(zone)) |>
  autoplot(filter(tourism_agg, year(month) > 2012), level = NULL)
```



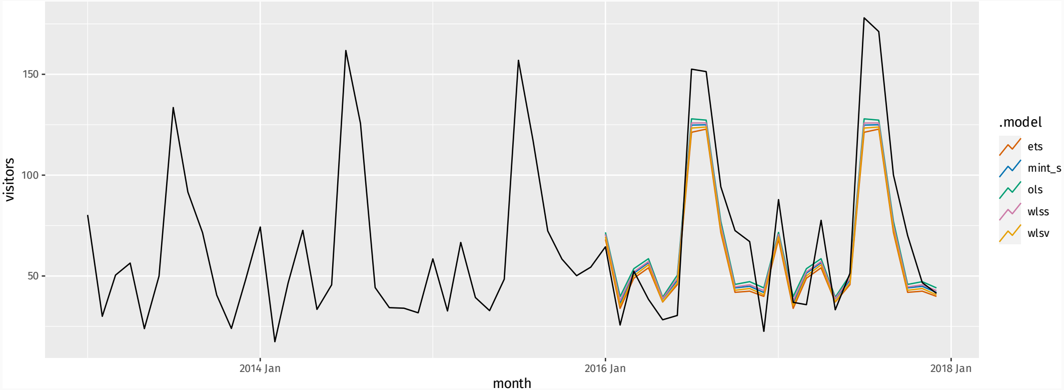
Example: Australian tourism

```r
fc |>
  filter(region == "Melbourne") |>
  autoplot(filter(tourism_agg, year(month) > 2012), level = NULL)
```



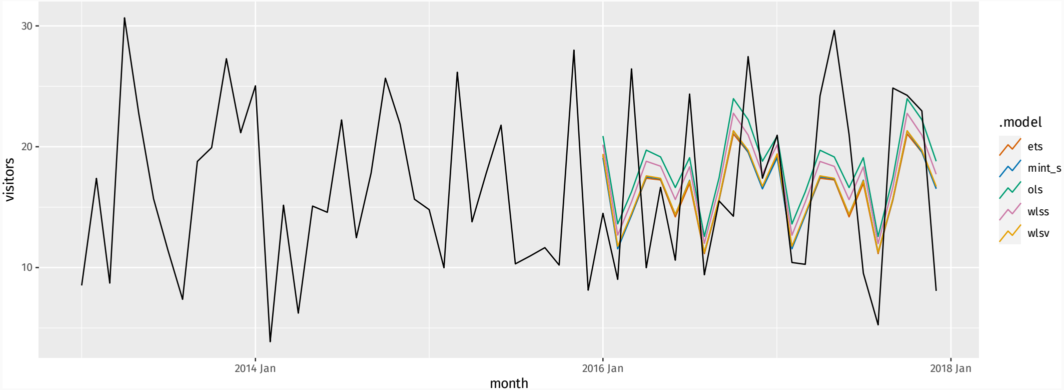
Example: Australian tourism

```
fc |>
  filter(region == "Snowy Mountains") |>
  autoplot(filter(tourism_agg, year(month) > 2012), level = NULL)
```



Example: Australian tourism

```r
fc |>
  filter(region == "Barossa") |>
  autoplot(filter(tourism_agg, year(month) > 2012), level = NULL)
```



Example: Australian tourism

```
fc |>
  accuracy(
    data = tourism_agg,
    measures = list(rmsse = RMSSE)
  )
```

```
# A tibble: 550 x 6
   .model state  zone             region           .type rmsse
   <chr>  <chr*> <chr*>           <chr*>           <chr> <dbl>
 1 ets    NSW    ACT              Canberra         Test  0.835
 2 ets    NSW    ACT              <aggregated>     Test  0.835
 3 ets    NSW    Metro NSW        Central Coast    Test  0.747
 4 ets    NSW    Metro NSW        Sydney           Test  1.16
 5 ets    NSW    Metro NSW        <aggregated>     Test  1.18
 6 ets    NSW    North Coast NSW  Hunter           Test  1.21
 7 ets    NSW    North Coast NSW  North Coast NSW  Test  0.884
 8 ets    NSW    North Coast NSW  <aggregated>     Test  1.02
```

Example: Australian tourism

```r
fc |>
  accuracy(tourism_agg,
    measures = list(mase = MASE, rmsse = RMSSE)
  ) |>
  group_by(.model) |>
  summarise(mase = mean(mase), rmsse = sqrt(mean(rmsse^2))) |>
  arrange(rmsse)
```

```
# A tibble: 5 x 3
  .model  mase rmsse
  <chr>  <dbl> <dbl>
1 ols    0.930 0.926
2 wlss   0.949 0.948
3 mint_s 0.953 0.954
4 wlsv   0.964 0.965
5 ets    0.968 0.968
```

Example: Australian tourism

```
fc |>
  accuracy(tourism_agg,
    measures = list(mase = MASE, rmsse = RMSSE)
  ) |>
  group_by(.model) |>
  summarise(mase = mean(mase), rmsse = sqrt(mean(rmsse^2))) |>
  arrange(rmsse)
```

```
# A tibble: 5 x 3
  .model   mase rmsse
  <chr>   <dbl> <dbl>
1 ols     0.930 0.926
2 wlss    0.949 0.948
3 mint_s  0.953 0.954
4 wlsv    0.964 0.965
5 ets     0.968 0.968
```

- Overall, every reconciliation method is better than the base ETS forecasts.

Example: Australian tourism

```
# A tibble: 20 x 4
# Groups:   .model [5]
   .model level     mase rmsse
   <chr>  <fct>    <dbl> <dbl>
 1 ets    National 1.44  1.27
 2 ols    National 1.46  1.29
 3 wlss   National 1.61  1.43
 4 mint_s National 1.64  1.45
 5 wlsv   National 1.69  1.49
 6 ols    State    1.07  1.08
 7 ets    State    1.10  1.11
 8 wlss   State    1.13  1.14
 9 mint_s State    1.15  1.15
10 wlsv   State    1.18  1.17
11 ols    Zone     0.954 0.948
12 wlss   Zone     0.987 0.980
13 mint_s Zone     0.995 0.988
14 ets    Zone     1.01  0.999
15 wlsv   Zone     1.01  1.00
16 ols    Region   0.901 0.895
17 wlss   Region   0.910 0.907
18 mint_s Region   0.911 0.911
19 wlsv   Region   0.917 0.919
20 ets    Region   0.935 0.938
```

Example: Australian tourism

```
# A tibble: 20 x 4
# Groups:   .model [5]
   .model level     mase rmsse
   <chr>  <fct>    <dbl> <dbl>
 1 ets    National 1.44  1.27
 2 ols    National 1.46  1.29
 3 wlss   National 1.61  1.43
 4 mint_s National 1.64  1.45
 5 wlsv   National 1.69  1.49
 6 ols    State    1.07  1.08
 7 ets    State    1.10  1.11
 8 wlss   State    1.13  1.14
 9 mint_s State    1.15  1.15
10 wlsv   State    1.18  1.17
11 ols    Zone     0.954 0.948
12 wlss   Zone     0.987 0.980
13 mint_s Zone     0.995 0.988
14 ets    Zone     1.01  0.999
15 wlsv   Zone     1.01  1.00
16 ols    Region   0.901 0.895
17 wlss   Region   0.910 0.907
18 mint_s Region   0.911 0.911
19 wlsv   Region   0.917 0.919
20 ets    Region   0.935 0.938
```

- OLS is best for all levels except national.
- Improvements due to reconciliation are greater at lower levels.

Outline

1 Hierarchical time series data

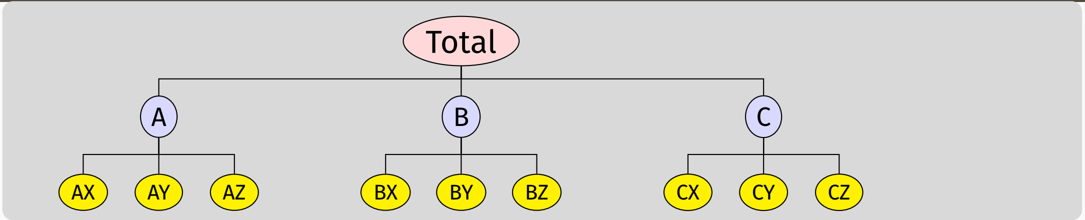
2 Hierarchical forecasting using single-level approaches

3 Linear forecast reconciliation

4 Example: Australian tourism

5 Fast computational tricks

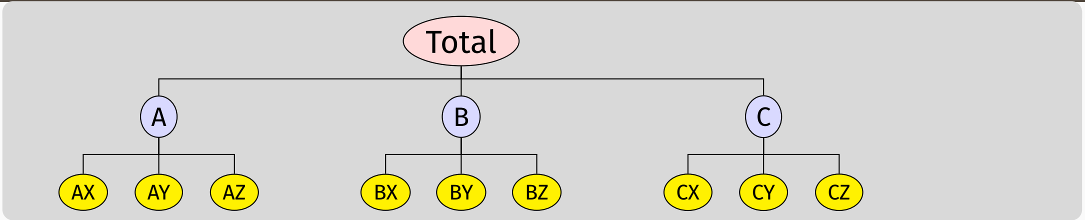
Fast computation: hierarchical data



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

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

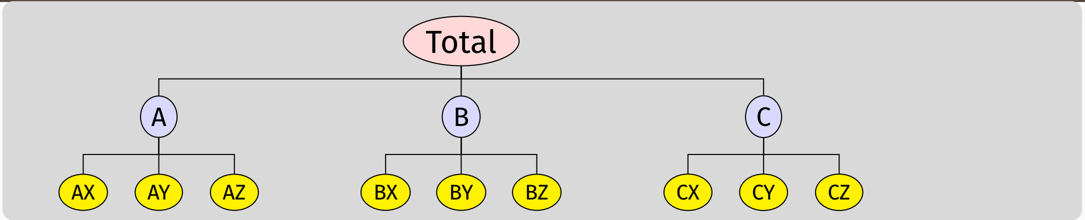
Fast computation: hierarchical data



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

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

Fast computation: hierarchical data

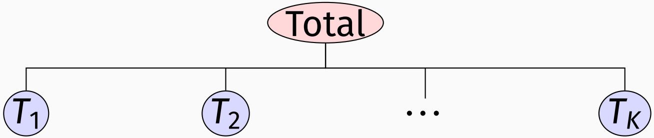


$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{B,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{C,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0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

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

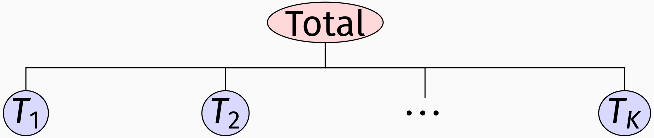
Fast computation: hierarchies

Think of the hierarchy as a tree of trees:



Fast computation: hierarchies

Think of the hierarchy as a tree of trees:



Then the summing matrix contains k smaller summing matrices:

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{1}'_{n_1} & \mathbf{1}'_{n_2} & \cdots & \mathbf{1}'_{n_K} \\ \boldsymbol{S}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{S}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{S}_K \end{bmatrix}$$

where $\mathbf{1}_n$ is an n-vector of ones and tree T_i has n_i terminal nodes.

Fast computation: hierarchies

$$\boldsymbol{S}'\Lambda\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_1'\Lambda_1\boldsymbol{S}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_2'\Lambda_2\boldsymbol{S}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{S}_K'\Lambda_K\boldsymbol{S}_K \end{bmatrix} + \lambda_0 \boldsymbol{J}_n$$

- λ_0 is the top left element of Λ
- Λ_k is a block of Λ, corresponding to tree T_k
- $\boldsymbol{J}_n$ is a matrix of ones
- $n = \sum_k n_k$

Fast computation: hierarchies

$$\boldsymbol{S}'\Lambda\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_1'\Lambda_1\boldsymbol{S}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_2'\Lambda_2\boldsymbol{S}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{S}_K'\Lambda_K\boldsymbol{S}_K \end{bmatrix} + \lambda_0 \boldsymbol{J}_n$$

- λ_0 is the top left element of Λ
- Λ_k is a block of Λ, corresponding to tree T_k
- $\boldsymbol{J}_n$ is a matrix of ones
- $n = \sum_k n_k$

Now apply the Sherman-Morrison formula . . .

Fast computation: hierarchies

$$(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1} = \begin{bmatrix} (\boldsymbol{S}_1'\Lambda_1\boldsymbol{S}_1)^{-1} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{S}_2'\Lambda_2\boldsymbol{S}_2)^{-1} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & (\boldsymbol{S}_K'\Lambda_K\boldsymbol{S}_K)^{-1} \end{bmatrix} - c\boldsymbol{S}_0$$

- $\boldsymbol{S}_0$ can be partitioned into K^2 blocks, with the (k,ℓ) block (of dimension $n_k \times n_\ell$) being $(\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k)^{-1}\boldsymbol{J}_{n_k,n_\ell}(\boldsymbol{S}_\ell'\Lambda_\ell\boldsymbol{S}_\ell)^{-1}$
- $\boldsymbol{J}_{n_k,n_\ell}$ is a $n_k \times n_\ell$ matrix of ones
- $c^{-1} = \lambda_0^{-1} + \sum_k \boldsymbol{1}_{n_k}'(\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k)^{-1}\boldsymbol{1}_{n_k}$
- Each $\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k$ can be inverted similarly
- $\boldsymbol{S}'\Lambda\boldsymbol{y}$ can also be computed recursively

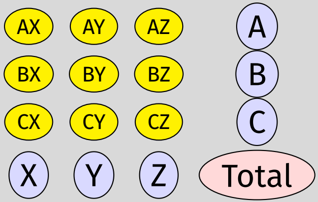
Fast computation: hierarchies

$$(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1} = \begin{bmatrix} (\boldsymbol{S}_1'\Lambda_1\boldsymbol{S}_1)^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{S}_2'\Lambda_2\boldsymbol{S}_2)^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (\boldsymbol{S}_K'\Lambda_K\boldsymbol{S}_K)^{-1} \end{bmatrix} - c\boldsymbol{S}_0$$

- $\boldsymbol{S}_0$ can be partitioned into K^2 blocks, with the (k,ℓ) block (of dimension $n_k \times n_\ell$) being $(\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k)^{-1}\boldsymbol{J}_{n_k,n_\ell}(\boldsymbol{S}_\ell'\Lambda_\ell\boldsymbol{S}_\ell)^{-1}$
- $\boldsymbol{J}_{n_k,n_\ell}$ is a $n_k \times n_\ell$ matrix of ones
- $c^{-1} = \lambda_0^{-1} + \sum_k \mathbf{1}_{n_k}'(\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k)^{-1}\mathbf{1}_{n_k}$
- Each $\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k$ can be inverted similarly
- $\boldsymbol{S}'\Lambda\boldsymbol{y}$ can also be computed recursivel

The recursive calculations can be done in such a way that we never store any of the large matrices involved.

Fast computation: grouped data



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

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

Fast computation: grouped data

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{1}'_m \otimes \mathbf{1}'_n \\ \mathbf{1}'_m \otimes \boldsymbol{I}_n \\ \boldsymbol{I}_m \otimes \mathbf{1}'_n \\ \boldsymbol{I}_m \otimes \boldsymbol{I}_n \end{bmatrix}$$

m = number of rows
n = number of columns

Fast computation: grouped data

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{1}'_m \otimes \mathbf{1}'_n \\ \mathbf{1}'_m \otimes \boldsymbol{I}_n \\ \boldsymbol{I}_m \otimes \mathbf{1}'_n \\ \boldsymbol{I}_m \otimes \boldsymbol{I}_n \end{bmatrix}$$

m = number of rows
n = number of columns

$$\boldsymbol{S}'\Lambda\boldsymbol{S} = \lambda_{00}\boldsymbol{J}_{mn} + (\Lambda_R \otimes \boldsymbol{J}_n) + (\boldsymbol{J}_m \otimes \Lambda_C) + \Lambda_U$$

- Λ_R, Λ_C and Λ_U are diagonal matrices corresponding to rows, columns and unaggregated series;
- λ_{00} corresponds to aggregate.

Fast computation: grouped data

$$(\boldsymbol{S}\Lambda\boldsymbol{S})^{-1} = \boldsymbol{A} - \frac{\boldsymbol{A}\mathbf{1}_{mn}\mathbf{1}'_{mn}\boldsymbol{A}}{1/\lambda_{00} + \mathbf{1}'_{mn}\boldsymbol{A}\mathbf{1}_{mn}}$$

$\boldsymbol{A} = \Lambda_U^{-1} - \Lambda_U^{-1}(\boldsymbol{J}_m \otimes \boldsymbol{D})\Lambda_U^{-1} - \boldsymbol{E}\boldsymbol{M}^{-1}\boldsymbol{E}'$.
$\boldsymbol{D}$ is diagonal with elements $d_j = \lambda_{0j}/(1 + \lambda_{0j}\sum_i \lambda_{ij}^{-1})$.
$\boldsymbol{E}$ has $m \times m$ blocks where $\boldsymbol{e}_{ij}$ has kth element

$$(\boldsymbol{e}_{ij})_k = \begin{cases} \lambda_{i0}^{1/2}\lambda_{ik}^{-1} - \lambda_{i0}^{1/2}\lambda_{ik}^{-2}d_k, & i = j, \\ -\lambda_{j0}^{1/2}\lambda_{ik}^{-1}\lambda_{jk}^{-1}d_k, & i \neq j. \end{cases}$$

$\boldsymbol{M}$ is $m \times m$ with (i, j) element

$$(\boldsymbol{M})_{ij} = \begin{cases} 1 + \lambda_{i0}\sum_k \lambda_{ik}^{-1} - \lambda_{i0}\sum_k \lambda_{ik}^{-2}d_k, & i = j, \\ -\lambda_{i0}^{1/2}\lambda_{j0}^{1/2}\sum_k \lambda_{ik}^{-1}\lambda_{jk}^{-1}d_k, & i \neq j. \end{cases}$$

Fast computation: grouped data

$$(\boldsymbol{S}\Lambda\boldsymbol{S})^{-1} = \boldsymbol{A} - \frac{\boldsymbol{A}\boldsymbol{1}_{mn}\boldsymbol{1}'_{mn}\boldsymbol{A}}{1/\lambda_{00} + \boldsymbol{1}'_{mn}\boldsymbol{A}\boldsymbol{1}_{mn}}$$

$\boldsymbol{A} = \Lambda_U^{-1} - \Lambda_U^{-1}(\boldsymbol{J}_m \otimes \boldsymbol{D})\Lambda_U^{-1} - \boldsymbol{E}\boldsymbol{M}^{-1}\boldsymbol{E}'$.

$\boldsymbol{D}$ is diagonal with elements $d_j = \lambda_{0j}/(1 + \lambda_{0j}\sum_i \lambda_{ij}^{-1})$.

$\boldsymbol{E}$ has $m \times m$ blocks where $\boldsymbol{e}_{ij}$ has kth element

$$(\boldsymbol{e}_{ij})_k = \begin{cases} \lambda_{i0}^{1/2}\lambda_{ik}^{-1} - \lambda_{i0}^{1/2}\lambda_{ik}^{-2}d_k, & i = j, \\ -\lambda_{j0}^{1/2}\lambda_{ik}^{-1}\lambda_{jk}^{-1}d_k, & i \neq j. \end{cases}$$

$\boldsymbol{M}$ is $m \times m$ with (i, j) element

$$(\boldsymbol{M})_{ij} = \begin{cases} 1 + \lambda_{i0}\sum_k \lambda_{ik}^{-1} - \lambda_{i0}\sum_k \lambda_{ik}^{-2}d_k, & i = j, \\ -\lambda_{i0}^{1/2}\lambda_{j0}^{1/2}\sum_k \lambda_{ik}^{-1}\lambda_{jk}^{-1}d_k, & i \neq j. \end{cases}$$

Again, the calculations can be done in such a way that we never store any of the large matrices involved

Fast forecast reconciliation using linear models

Suppose $\hat{y}_{t,i} = \hat{\beta}_i' \boldsymbol{x}_{t,i}$ with $\boldsymbol{x}_{t,i} = (1, x_{t,1,i}, \dots, x_{t,p,i})$ & $\hat{\boldsymbol{y}}_i = (\hat{y}_{1,i}, \dots, \hat{y}_{T,i})$.

Fast forecast reconciliation using linear models

Suppose $\hat{y}_{t,i} = \hat{\beta}_i' \boldsymbol{x}_{t,i}$ with $\boldsymbol{x}_{t,i} = (1, x_{t,1,i}, \dots, x_{t,p,i})$ & $\hat{\boldsymbol{y}}_i = (\hat{y}_{1,i}, \dots, \hat{y}_{T,i})$.

$$\begin{pmatrix} \hat{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_2 \\ \vdots \\ \hat{\boldsymbol{y}}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{X}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \boldsymbol{X}_n \end{pmatrix} \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_n \end{pmatrix}, \qquad \boldsymbol{X}_i = \begin{pmatrix} 1 & x_{1,i,1} & x_{1,i,2} & \dots & x_{1,i,p} \\ 1 & x_{2,i,1} & x_{2,i,2} & \dots & x_{2,i,p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{T,i,1} & x_{T,i,2} & \dots & x_{T,i,p} \end{pmatrix}$$

Fast forecast reconciliation using linear models

Suppose $\hat{y}_{t,i} = \hat{\beta}_i' \boldsymbol{x}_{t,i}$ with $\boldsymbol{x}_{t,i} = (1, x_{t,1,i}, \dots, x_{t,p,i})$ & $\hat{\boldsymbol{y}}_i = (\hat{y}_{1,i}, \dots, \hat{y}_{T,i})$.

$$\begin{pmatrix} \hat{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_2 \\ \vdots \\ \hat{\boldsymbol{y}}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{X}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \boldsymbol{X}_n \end{pmatrix} \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_n \end{pmatrix}, \qquad \boldsymbol{X}_i = \begin{pmatrix} 1 & x_{1,i,1} & x_{1,i,2} & \dots & x_{1,i,p} \\ 1 & x_{2,i,1} & x_{2,i,2} & \dots & x_{2,i,p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{T,i,1} & x_{T,i,2} & \dots & x_{T,i,p} \end{pmatrix}$$

Then $\hat{\boldsymbol{B}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$,

Fast forecast reconciliation using linear models

Suppose $\hat{y}_{t,i} = \hat{\beta}_i' \boldsymbol{x}_{t,i}$ with $\boldsymbol{x}_{t,i} = (1, x_{t,1,i}, \dots, x_{t,p,i})$ & $\hat{\boldsymbol{y}}_i = (\hat{y}_{1,i}, \dots, \hat{y}_{T,i})$.

$$\begin{pmatrix} \hat{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_2 \\ \vdots \\ \hat{\boldsymbol{y}}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{X}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \boldsymbol{X}_n \end{pmatrix} \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_n \end{pmatrix}, \qquad \boldsymbol{X}_i = \begin{pmatrix} 1 & x_{1,i,1} & x_{1,i,2} & \dots & x_{1,i,p} \\ 1 & x_{2,i,1} & x_{2,i,2} & \dots & x_{2,i,p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{T,i,1} & x_{T,i,2} & \dots & x_{T,i,p} \end{pmatrix}$$

Then $\hat{\boldsymbol{B}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$,and

$\hat{\boldsymbol{y}}_{t+h} = \boldsymbol{X}^*_{t+h}\hat{\boldsymbol{B}}$ with $\boldsymbol{X}^*_{t+h} = \text{diag}(\boldsymbol{x}'_{t+h,i}, \dots, \boldsymbol{x}'_{t+h,n})$.

Fast forecast reconciliation using linear models

Suppose $\hat{y}_{t,i} = \hat{\beta}_i'\boldsymbol{x}_{t,i}$ with $\boldsymbol{x}_{t,i} = (1, x_{t,1,i}, \dots, x_{t,p,i})$ & $\hat{\boldsymbol{y}}_i = (\hat{y}_{1,i}, \dots, \hat{y}_{T,i})$.

$$\begin{pmatrix} \hat{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_2 \\ \vdots \\ \hat{\boldsymbol{y}}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{X}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \boldsymbol{X}_n \end{pmatrix} \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_n \end{pmatrix}, \qquad \boldsymbol{X}_i = \begin{pmatrix} 1 & x_{1,i,1} & x_{1,i,2} & \dots & x_{1,i,p} \\ 1 & x_{2,i,1} & x_{2,i,2} & \dots & x_{2,i,p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{T,i,1} & x_{T,i,2} & \dots & x_{T,i,p} \end{pmatrix}$$

Then $\hat{\boldsymbol{B}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$,and

$\hat{\boldsymbol{y}}_{t+h} = \boldsymbol{X}^*_{t+h}\hat{\boldsymbol{B}}$ with $\boldsymbol{X}^*_{t+h} = \text{diag}(\boldsymbol{x}'_{t+h,i}, \dots, \boldsymbol{x}'_{t+h,n})$.So

$$\tilde{\boldsymbol{y}}_{t+h} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h\hat{\boldsymbol{y}}_{t+h} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h\boldsymbol{X}^*_{t+h}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

References

- Ashouri, M, RJ Hyndman, and G Shmueli (2022). Fast forecast reconciliation using linear models. *J Computational & Graphical Statistics* **31**(1), 263–282.
- Hyndman, RJ, RA Ahmed, G Athanasopoulos, and HL Shang (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis* **55**(9), 2579–2589.
- Hyndman, RJ, A Lee, and E Wang (2016). Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational Statistics & Data Analysis* **97**, 16–32.
- Wickramasuriya, SL, G Athanasopoulos, and RJ Hyndman (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *J American Statistical Association* **114**(526), 804–819.