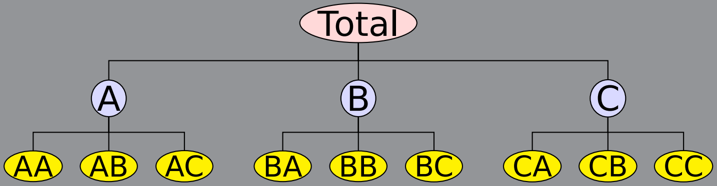
MONASH BUSINESS SCHOOL

Rob J Hyndman

# Optimal forecast reconciliation for big time series data



# Outline

# Labour market participation

## Australia and 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

# Labour market participation

## Australia and 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

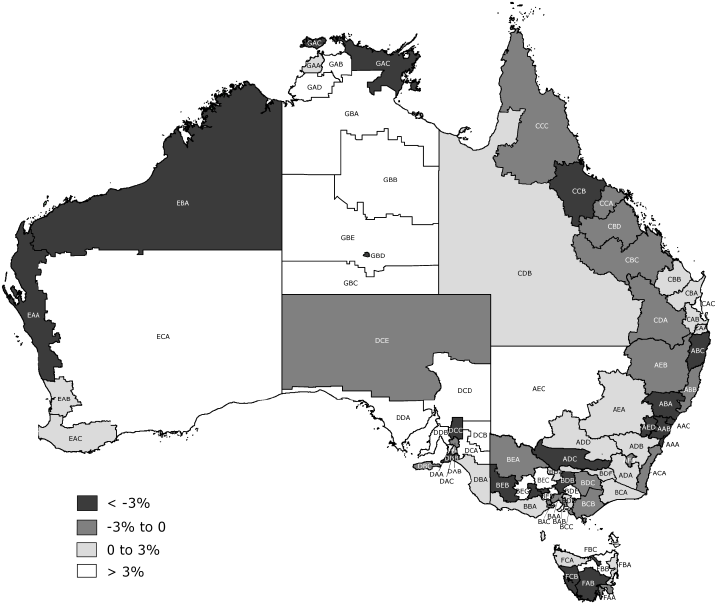
**22** Business, Human Resource and Marketing Professionals

**224** Information and Organisation Professionals

2241 Actuaries, Mathematicians and Statisticians

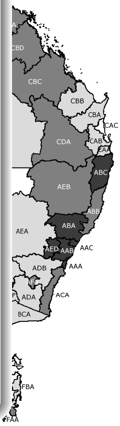
224113 Statistician

# Australian tourism demand



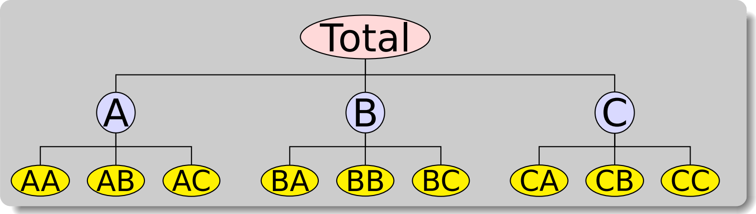
# Australian tourism demand

- 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



# Hierarchical time series

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

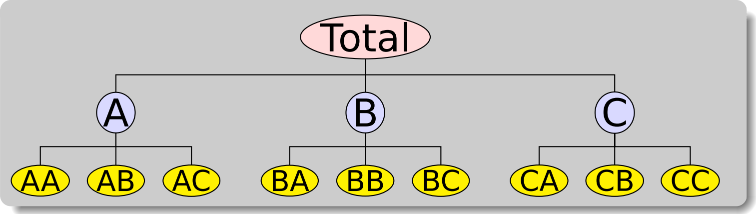



## Examples

- Labour turnover by occupation
- Tourism by state and region

# Hierarchical time series

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

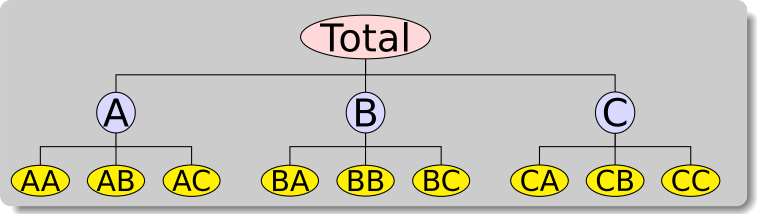



## Examples

- Labour turnover by occupation
- Tourism by state and region

# Hierarchical time series

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

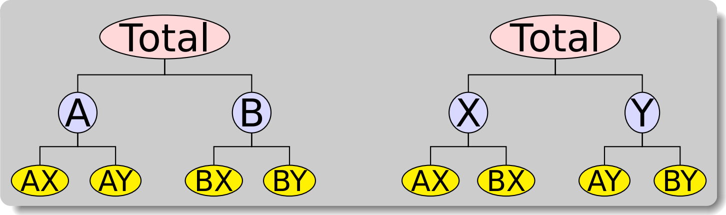



## Examples

- Labour turnover by occupation
- Tourism by state and region

# 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.

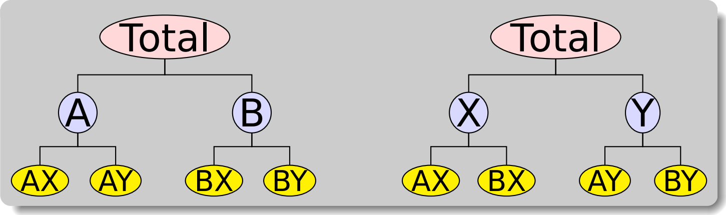



## Examples

- Labour turnover by occupation and state
- Tourism by state and purpose of travel

# 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.

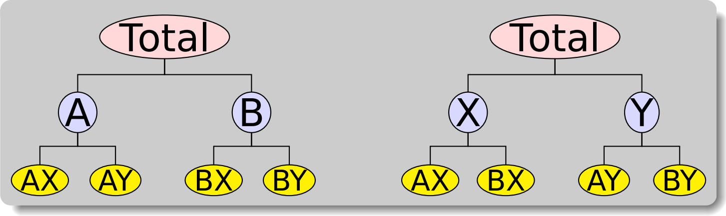



## Examples

- Labour turnover by occupation and state
- Tourism by state and purpose of travel

# Grouped time series

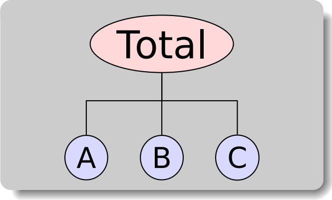
A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.




## Examples

- Labour turnover by occupation and state
- Tourism by state and purpose of travel

# Hierarchical time series

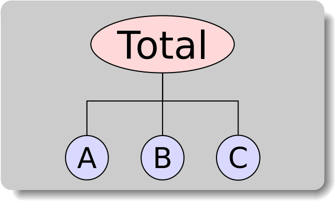



$y_t$ : observed aggregate of all series at time $t$.

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

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

# Hierarchical time series

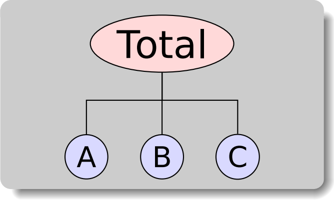



$y_t$ : observed aggregate of all series at time $t$.

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

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

# Hierarchical time series



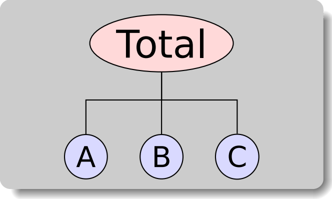

$y_t$ : observed aggregate of all series at time $t$.

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

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

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}$$

# Hierarchical time series



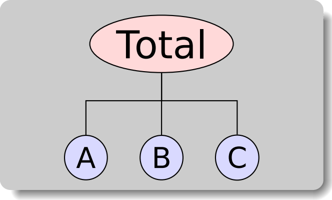

$y_t$ : observed aggregate of all series at time $t$.

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

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

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}$$

# Hierarchical time series



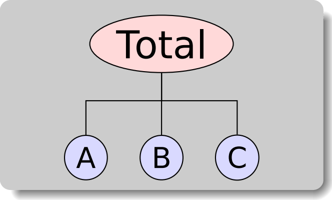

$y_t$ : observed aggregate of all series at time $t$.

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

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

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

# Hierarchical time series




$y_t$ : observed aggregate of all series at time $t$.

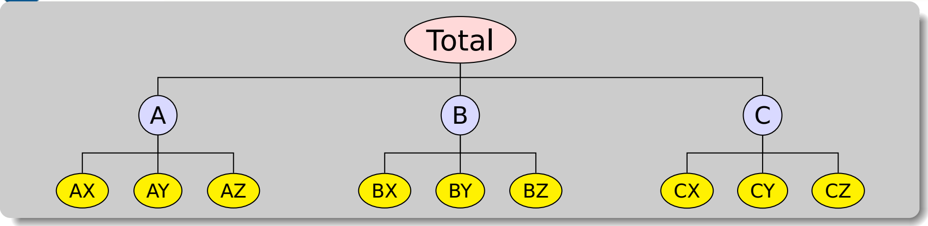
$y_{X,t}$ : observation on series $X$ at time $t$.

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

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

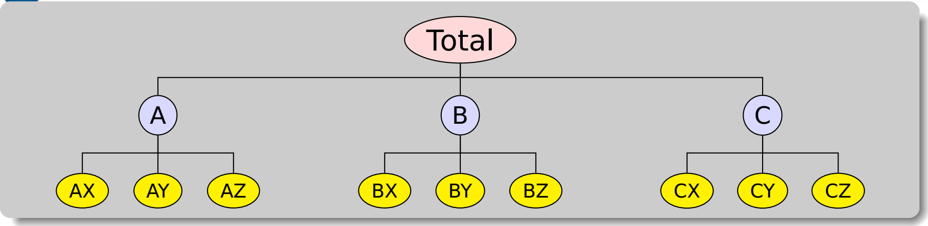
$$\boxed{\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t}$$

# Hierarchical time series



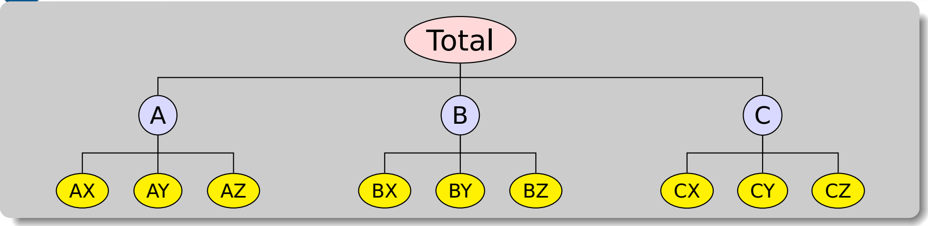

$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0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

# Hierarchical time series




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

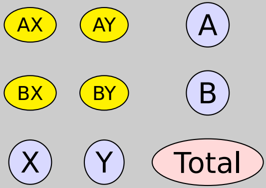
# Hierarchical time series




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

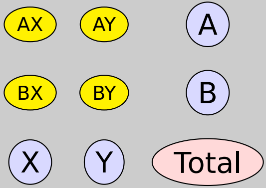
$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

# Grouped data



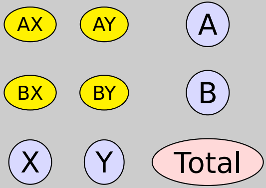

$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{X,t} \\ y_{Y,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0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

# Grouped data




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

# Grouped data




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

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

# 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.

# Outline

# Forecasting notation

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

Reconciled forecasts must be of the form:
$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$
for some matrix $\boldsymbol{P}$.

- $\boldsymbol{P}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_n(h)$ to get bottom-level forecasts.

# Forecasting notation

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

Reconciled forecasts must be of the form:

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

for some matrix $\boldsymbol{P}$.

- $\boldsymbol{P}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_n(h)$ to get bottom-level forecasts.
- $\boldsymbol{S}$ adds them 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 $\boldsymbol{y}_t$. (They may not add up.)

Reconciled forecasts must be of the form:

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

for some matrix $\boldsymbol{P}$.

- $\boldsymbol{P}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_n(h)$ to get bottom-level forecasts.
- $\boldsymbol{S}$ adds them 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 $\boldsymbol{y}_t$. (They may not add up.)

Reconciled forecasts must be of the form:

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

for some matrix $\boldsymbol{P}$.

- $\boldsymbol{P}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_n(h)$ to get bottom-level forecasts.
- $\boldsymbol{S}$ adds them 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 $\boldsymbol{y}_t$. (They may not add up.)

Reconciled forecasts must be of the form:

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

for some matrix $\boldsymbol{P}$.

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

# General properties: bias

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

**Assume:** base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased:

$$\mathrm{E}[\hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \mathrm{E}[\boldsymbol{y}_{n+h} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

- Let $\hat{\boldsymbol{b}}_n(h)$ be bottom level base forecasts with $\beta_n(h) = \mathrm{E}[\hat{\boldsymbol{b}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.
- Then $\mathrm{E}[\hat{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\beta_n(h)$.
- We want the revised forecasts to be unbiased: $\mathrm{E}[\tilde{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\boldsymbol{P}\boldsymbol{S}\beta_n(h) = \boldsymbol{S}\beta_n(h)$.

Revised forecasts are unbiased iff $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

# General properties: bias

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

**Assume:** base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased:

$$\mathrm{E}[\hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \mathrm{E}[\boldsymbol{y}_{n+h} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

- Let $\hat{\boldsymbol{b}}_n(h)$ be bottom level base forecasts with $\beta_n(h) = \mathrm{E}[\hat{\boldsymbol{b}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.
- Then $\mathrm{E}[\hat{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\beta_n(h)$.
- We want the revised forecasts to be unbiased: $\mathrm{E}[\tilde{\boldsymbol{y}}_n(h)] = \boldsymbol{SPS}\beta_n(h) = \boldsymbol{S}\beta_n(h)$.

Revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

# General properties: bias

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

**Assume:** base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased:

$$\mathrm{E}[\hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1,\dots,\boldsymbol{y}_n] = \mathrm{E}[\boldsymbol{y}_{n+h} \mid \boldsymbol{y}_1,\dots,\boldsymbol{y}_n]$$

- Let $\hat{\boldsymbol{b}}_n(h)$ be bottom level base forecasts with $\boldsymbol{\beta}_n(h) = \mathrm{E}[\hat{\boldsymbol{b}}_n(h) \mid \boldsymbol{y}_1,\dots,\boldsymbol{y}_n]$.
- Then $\mathrm{E}[\hat{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\boldsymbol{\beta}_n(h)$.
- We want the revised forecasts to be unbiased: $\mathrm{E}[\tilde{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\boldsymbol{P}\boldsymbol{S}\boldsymbol{\beta}_n(h) = \boldsymbol{S}\boldsymbol{\beta}_n(h)$.

Revised forecasts are unbiased iff $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

# General properties: bias

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

**Assume:** base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased:

$$\mathrm{E}[\hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \mathrm{E}[\boldsymbol{y}_{n+h} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

- Let $\hat{\boldsymbol{b}}_n(h)$ be bottom level base forecasts with $\boldsymbol{\beta}_n(h) = \mathrm{E}[\hat{\boldsymbol{b}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.
- Then $\mathrm{E}[\hat{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\boldsymbol{\beta}_n(h)$.
- We want the revised forecasts to be unbiased: $\mathrm{E}[\tilde{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\boldsymbol{P}\boldsymbol{S}\boldsymbol{\beta}_n(h) = \boldsymbol{S}\boldsymbol{\beta}_n(h)$.

Revised forecasts are unbiased iff $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

# General properties: bias

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

**Assume:** base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased:

$$\mathrm{E}[\hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \mathrm{E}[\boldsymbol{y}_{n+h} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

- Let $\hat{\boldsymbol{b}}_n(h)$ be bottom level base forecasts with $\boldsymbol{\beta}_n(h) = \mathrm{E}[\hat{\boldsymbol{b}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.
- Then $\mathrm{E}[\hat{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\boldsymbol{\beta}_n(h)$.
- We want the revised forecasts to be unbiased: $\mathrm{E}[\tilde{\boldsymbol{y}}_n(h)] = \boldsymbol{SPS}\boldsymbol{\beta}_n(h) = \boldsymbol{S}\boldsymbol{\beta}_n(h)$.

Revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

# General properties: bias

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

**Assume:** base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased:

$$\mathrm{E}[\hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \mathrm{E}[\boldsymbol{y}_{n+h} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

- Let $\hat{\boldsymbol{b}}_n(h)$ be bottom level base forecasts with $\beta_n(h) = \mathrm{E}[\hat{\boldsymbol{b}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.
- Then $\mathrm{E}[\hat{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\beta_n(h)$.
- We want the revised forecasts to be unbiased: $\mathrm{E}[\tilde{\boldsymbol{y}}_n(h)] = \boldsymbol{SPS}\beta_n(h) = \boldsymbol{S}\beta_n(h)$.

Revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

# General properties: bias

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

**Assume:** base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased:

$$\mathrm{E}[\hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \mathrm{E}[\boldsymbol{y}_{n+h} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

- Let $\hat{\boldsymbol{b}}_n(h)$ be bottom level base forecasts with $\beta_n(h) = \mathrm{E}[\hat{\boldsymbol{b}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.
- Then $\mathrm{E}[\hat{\boldsymbol{y}}_n(h)] = \boldsymbol{S}\beta_n(h)$.
- We want the revised forecasts to be unbiased: $\mathrm{E}[\tilde{\boldsymbol{y}}_n(h)] = \boldsymbol{SPS}\beta_n(h) = \boldsymbol{S}\beta_n(h)$.

Revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

# General properties: variance

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

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

$$\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{n+h} - \hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

Then the error variance of the corresponding revised forecasts is

$$\text{Var}[\boldsymbol{y}_{n+h} - \tilde{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \boldsymbol{S}\boldsymbol{P}\boldsymbol{W}_h\boldsymbol{P}'\boldsymbol{S}'$$

# General properties: variance

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

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

$$\boldsymbol{W}_h = \mathrm{Var}[\boldsymbol{y}_{n+h} - \hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

Then the error variance of the corresponding revised forecasts is

$$\mathrm{Var}[\boldsymbol{y}_{n+h} - \tilde{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \boldsymbol{S}\boldsymbol{P}\boldsymbol{W}_h\boldsymbol{P}'\boldsymbol{S}'$$

# BLUF via trace minimization

**Theorem**

For any $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, then

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{SPW}_h\boldsymbol{P}'\boldsymbol{S}']$$

has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}$.

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate, especially for $h > 1$.

# BLUF via trace minimization

**Theorem**

For any $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, then

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{SPW}_h\boldsymbol{P}'\boldsymbol{S}']$$

has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}$.

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate, especially for $h > 1$.

# BLUF via trace minimization

**Theorem**

For any $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, then

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{SPW}_h\boldsymbol{P}'\boldsymbol{S}']$$

has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}$.

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate, especially for $h > 1$.

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 1: OLS

- Assume forecast errors have the same aggregation constraints as the data.
- Then $\boldsymbol{W}_h = \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h$ is covariance matrix of bottom level errors.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

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

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 1: OLS

- Assume forecast errors have the same aggregation constraints as the data.
- Then $\boldsymbol{W}_h = \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h$ is covariance matrix of bottom level errors.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

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

# Optimal combination forecasts

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

Revised forecasts Base forecasts

## Solution 1: OLS

- Assume forecast errors have the same aggregation constraints as the data.
- Then $\boldsymbol{W}_h = \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h$ is covariance matrix of bottom level errors.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

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

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 1: OLS

- Assume forecast errors have the same aggregation constraints as the data.
- Then $\boldsymbol{W}_h = \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h$ is covariance matrix of bottom level errors.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

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

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 1: OLS

- Assume forecast errors have the same aggregation constraints as the data.
- Then $\boldsymbol{W}_h = \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h$ is covariance matrix of bottom level errors.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

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

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 1: OLS

- Assume forecast errors have the same aggregation constraints as the data.
- Then $\boldsymbol{W}_h = \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h$ is covariance matrix of bottom level errors.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

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

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 2: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Empirically, WLS gives better forecasts than OLS.
- Still need to estimate covariance matrix to produce prediction intervals.

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 2: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Empirically, WLS gives better forecasts than OLS.
- Still need to estimate covariance matrix to produce prediction intervals.

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 2: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Empirically, WLS gives better forecasts than OLS.
- Still need to estimate covariance matrix to produce prediction intervals.

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 2: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Empirically, WLS gives better forecasts than OLS.
- Still need to estimate covariance matrix to produce prediction intervals.

# Optimal combination forecasts

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

Revised forecasts Base forecasts

## Solution 2: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Empirically, WLS gives better forecasts than OLS.
- Still need to estimate covariance matrix to produce prediction intervals.

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 3: GLS

- Estimate $\boldsymbol{W}_1$ using shrinkage to the diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Allows for covariances.
- Empirically, GLS gives better forecasts than WLS or OLS.
- Difficult to compute for large numbers of time series.

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 3: GLS

- Estimate $\boldsymbol{W}_1$ using shrinkage to the diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Allows for covariances.
- Empirically, GLS gives better forecasts than WLS or OLS.
- Difficult to compute for large numbers of time series.

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 3: GLS

- Estimate $\boldsymbol{W}_1$ using shrinkage to the diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Allows for covariances.
- Empirically, GLS gives better forecasts than WLS or OLS.
- Difficult to compute for large numbers of time series.

# Optimal combination forecasts

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

Revised forecasts

Base forecasts

## Solution 3: GLS

- Estimate $\boldsymbol{W}_1$ using shrinkage to the diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Allows for covariances.
- Empirically, GLS gives better forecasts than WLS or OLS.
- Difficult to compute for large numbers of time series.

# Optimal combination forecasts

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

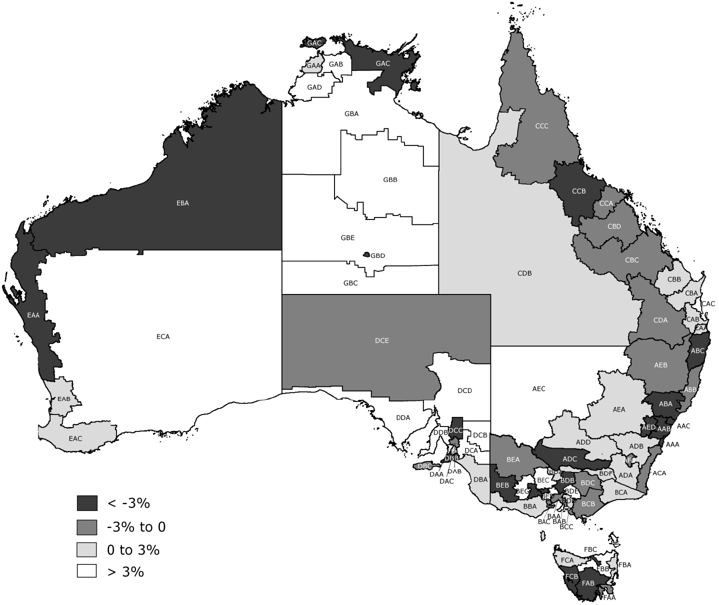
Revised forecasts | Base forecasts

## Solution 3: GLS

- Estimate $\boldsymbol{W}_1$ using shrinkage to the diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Allows for covariances.
- Empirically, GLS gives better forecasts than WLS or OLS.
- Difficult to compute for large numbers of time series.

# Outline

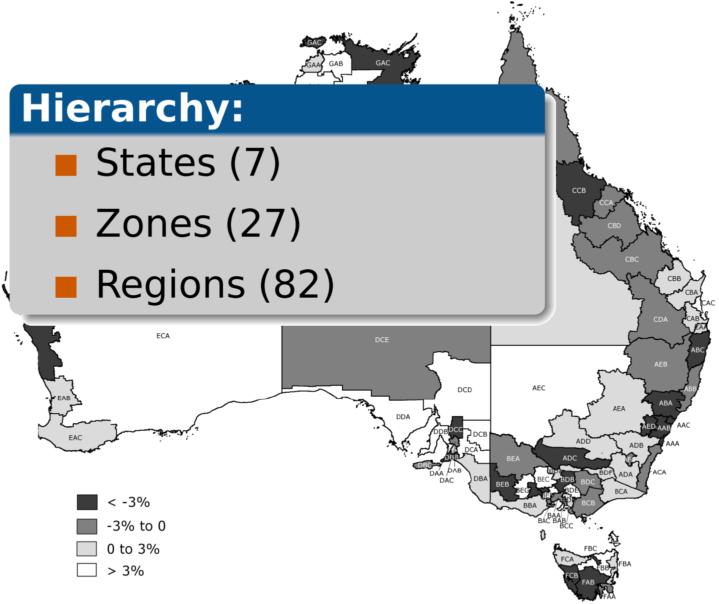
# Australian tourism



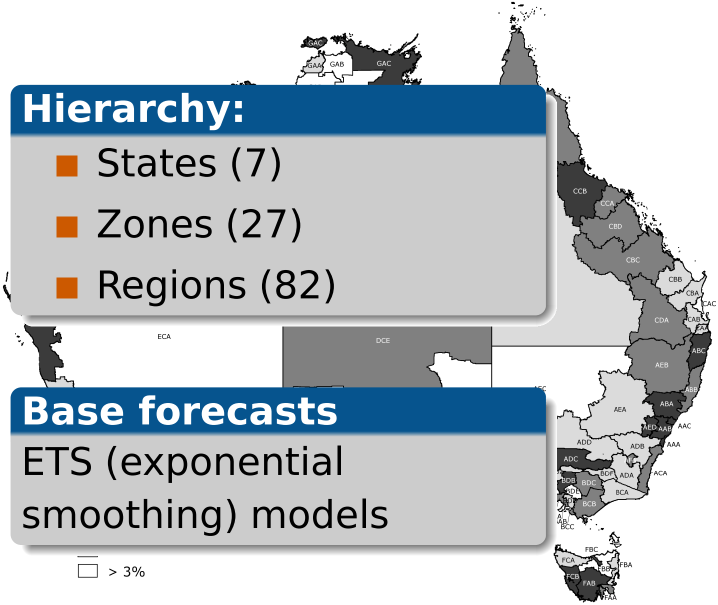
# Australian tourism

## Hierarchy:

- States (7)
- Zones (27)
- Regions (82)



# Australian tourism



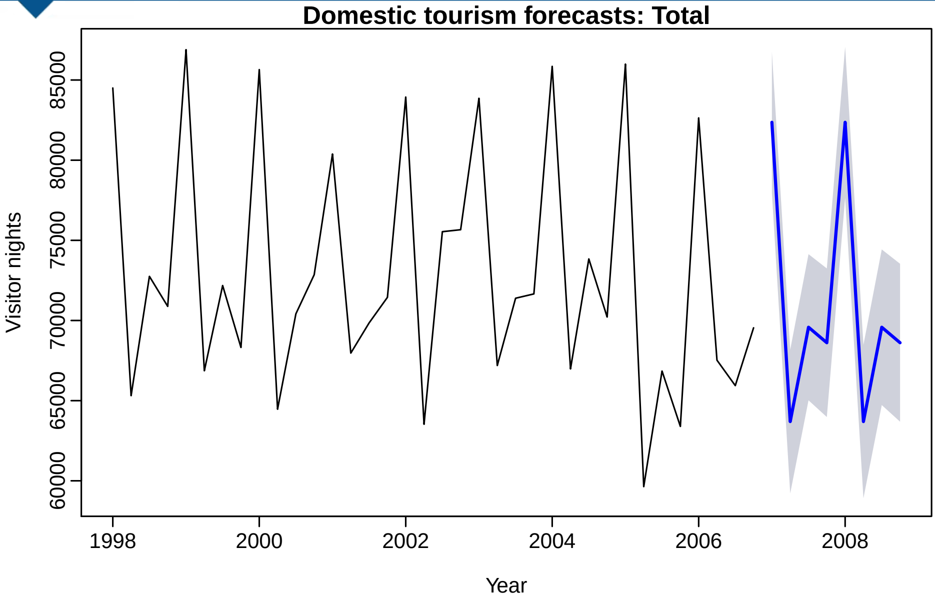

## Hierarchy:

- States (7)
- Zones (27)
- Regions (82)

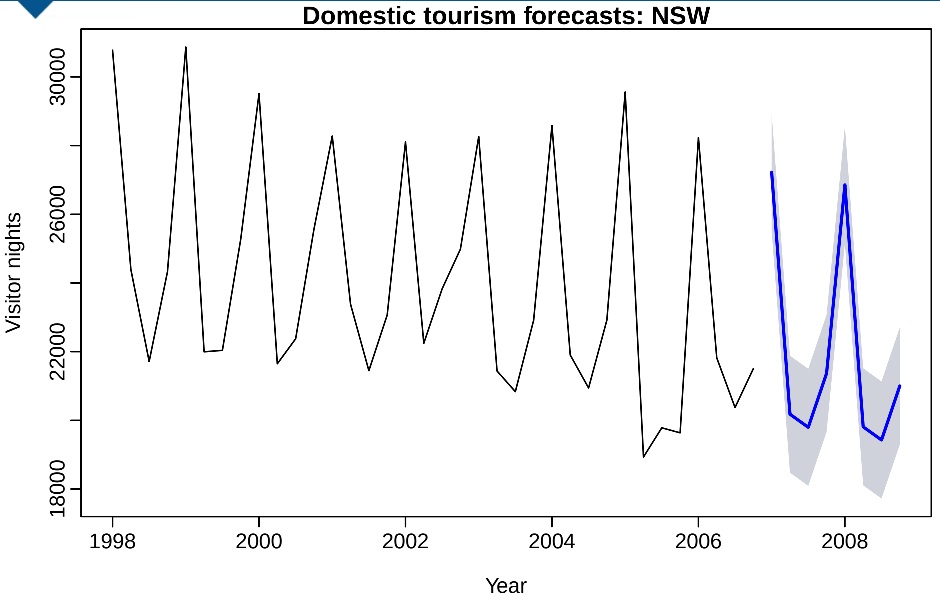
## Base forecasts

ETS (exponential smoothing) models

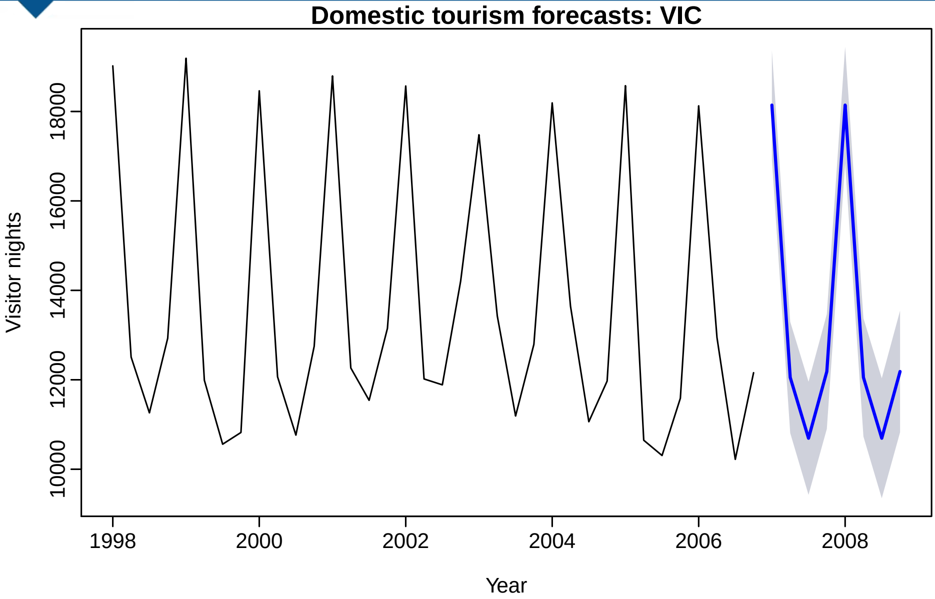
# Base forecasts



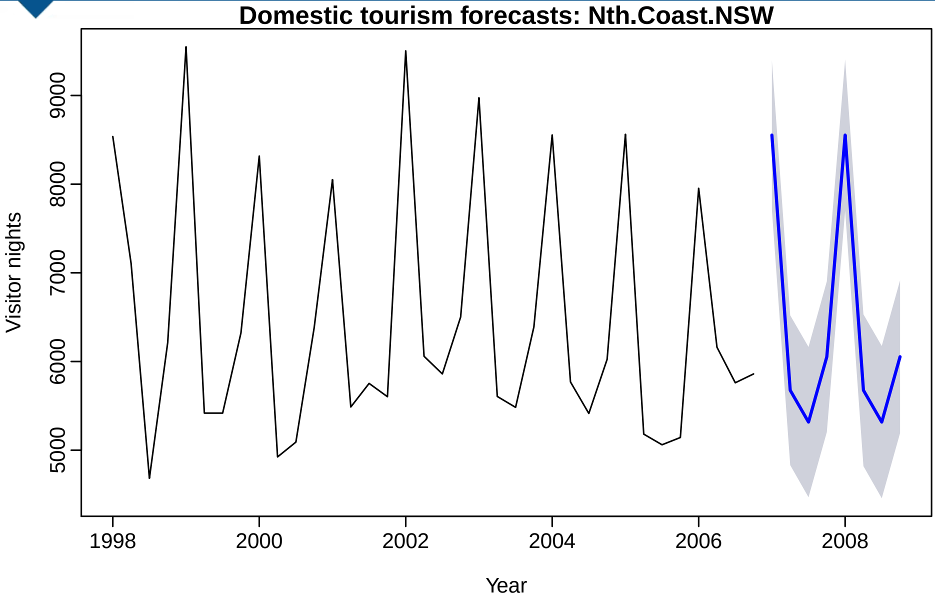
# Base forecasts



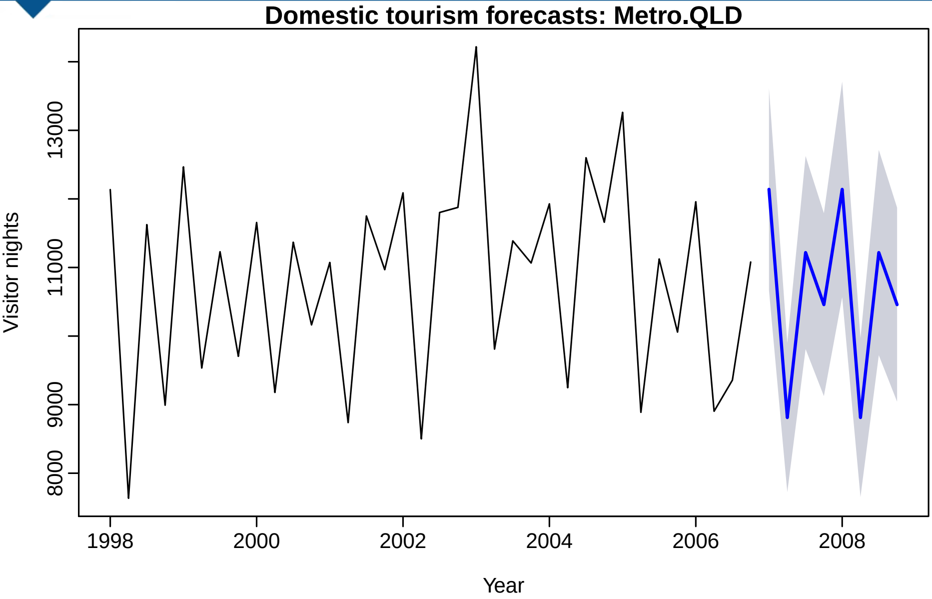
# Base forecasts



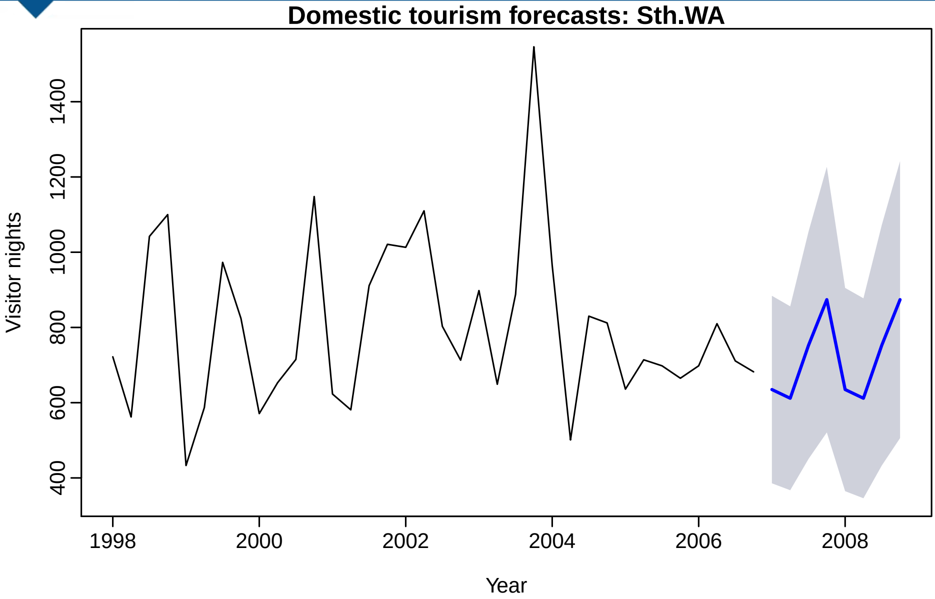
# Base forecasts



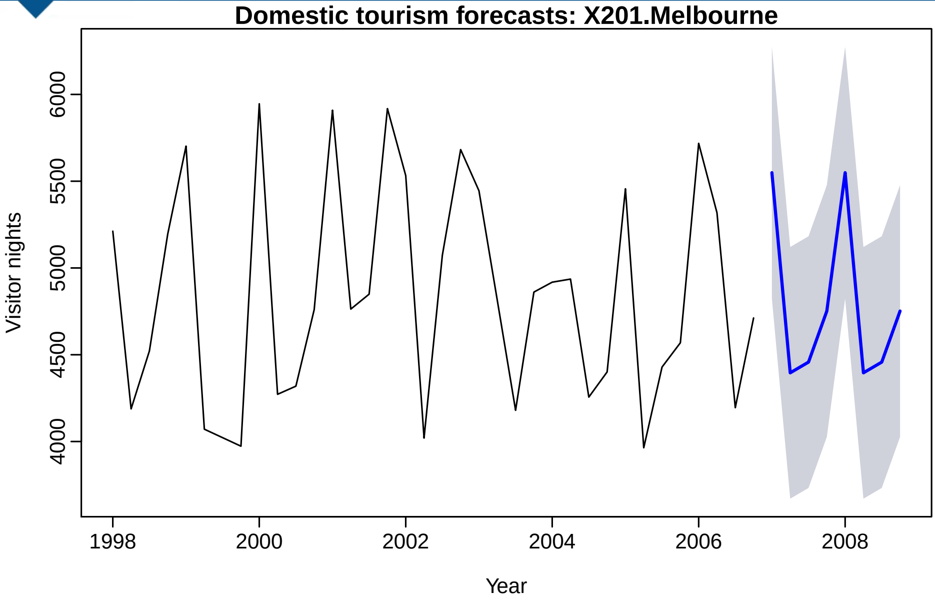
# Base forecasts



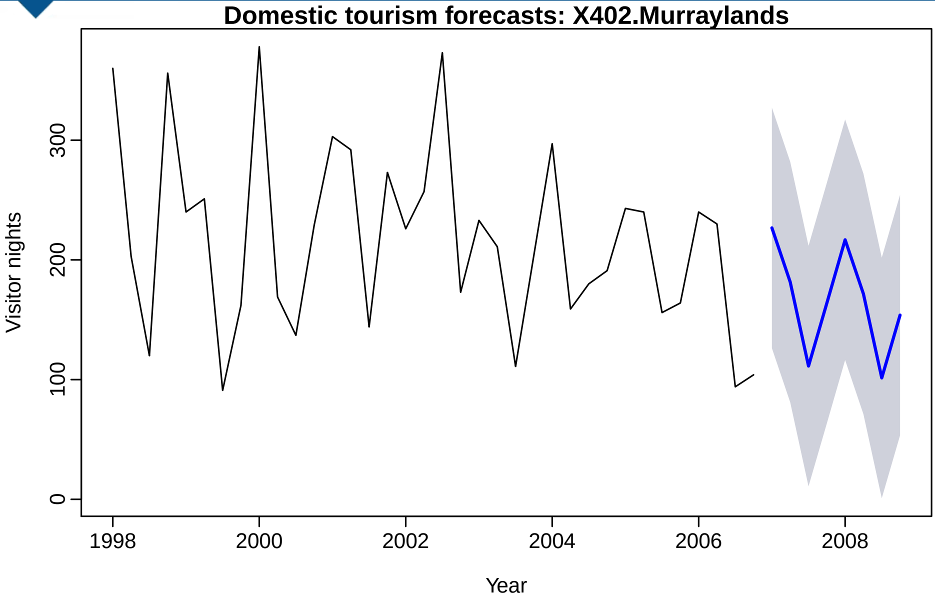
# Base forecasts



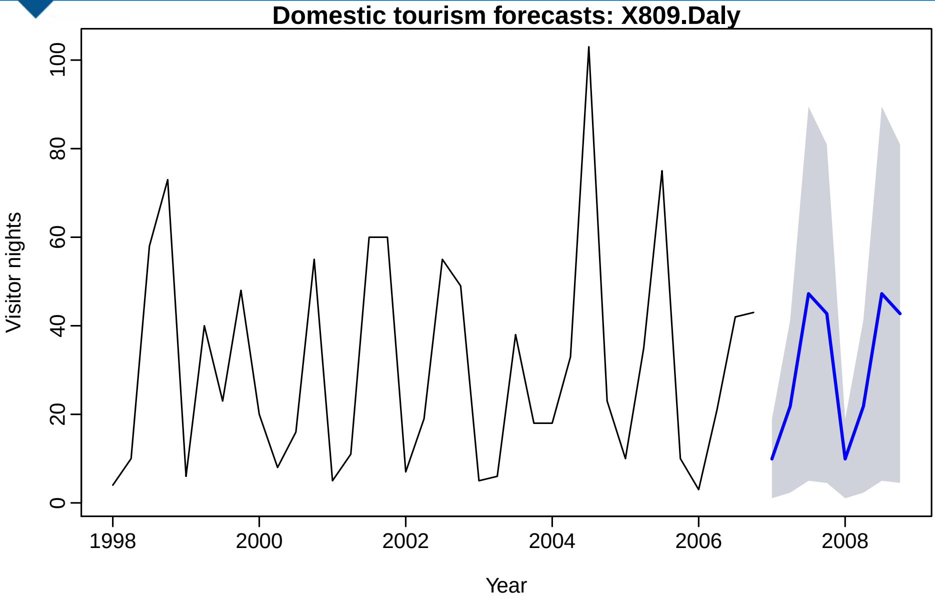
# Base forecasts



# Base forecasts

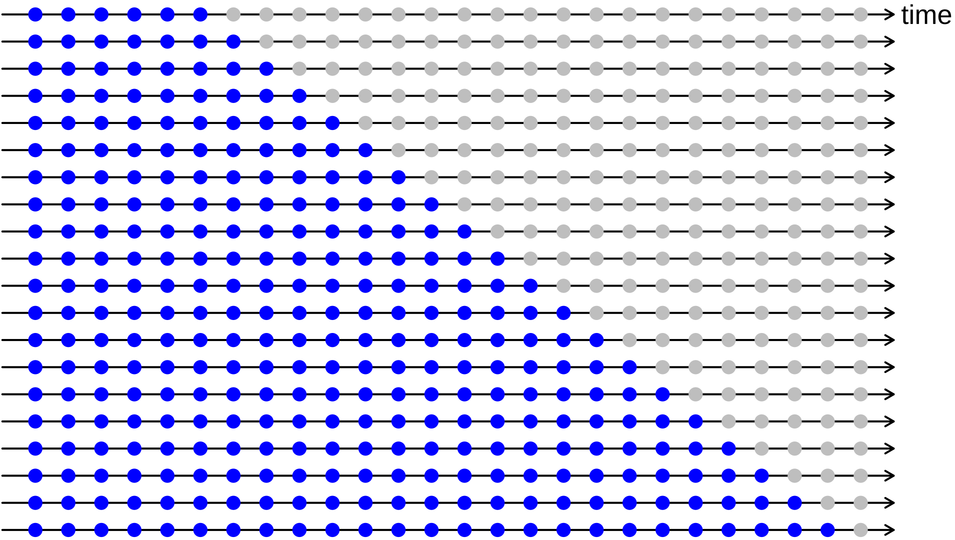


# Base forecasts

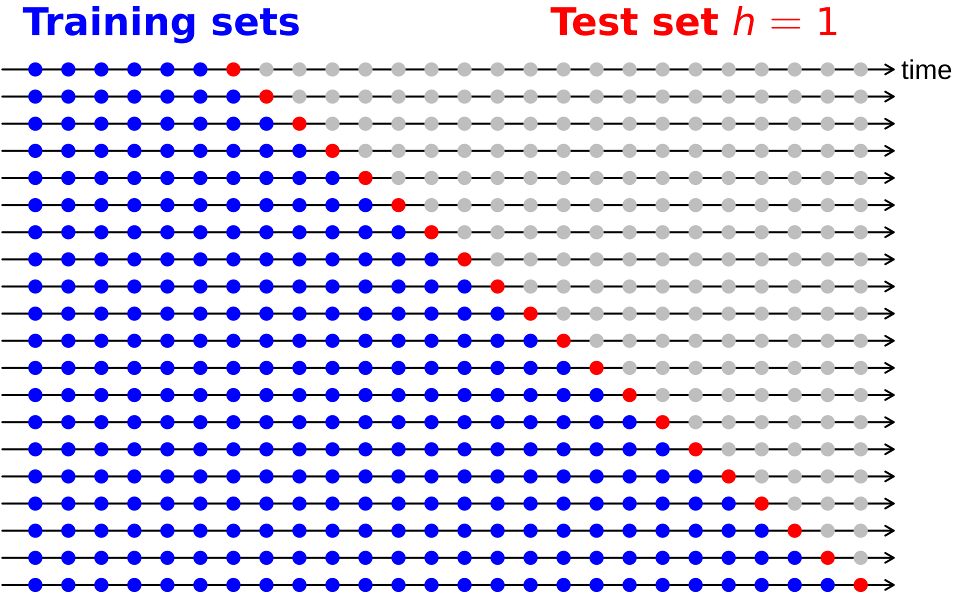


# Forecast evaluation

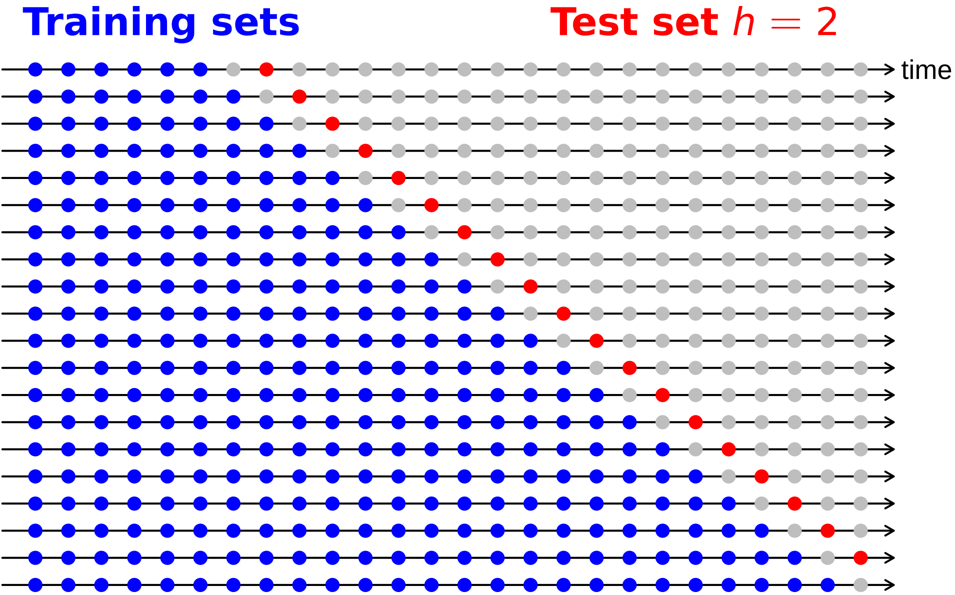
## Training sets



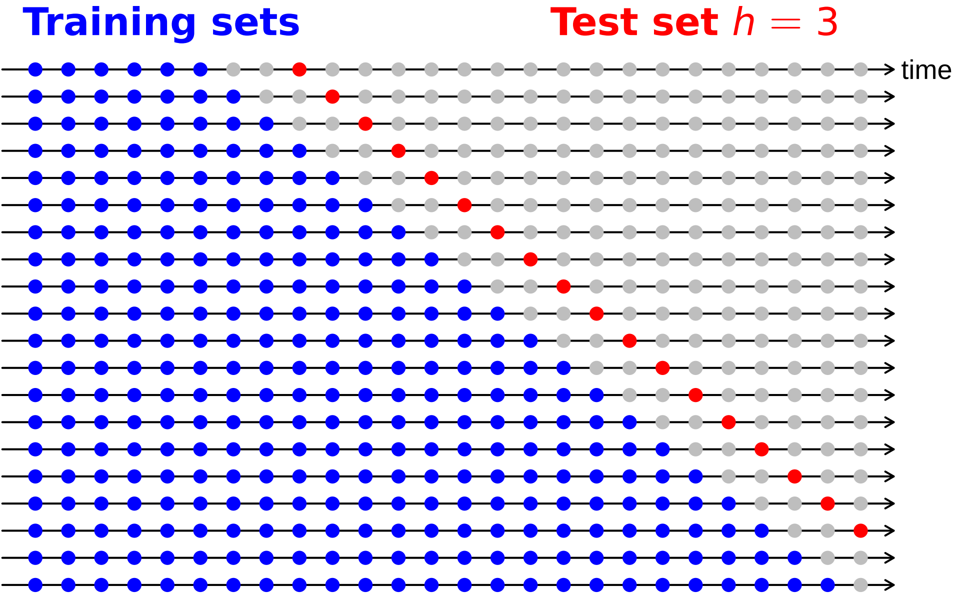
# Forecast evaluation



# Forecast evaluation

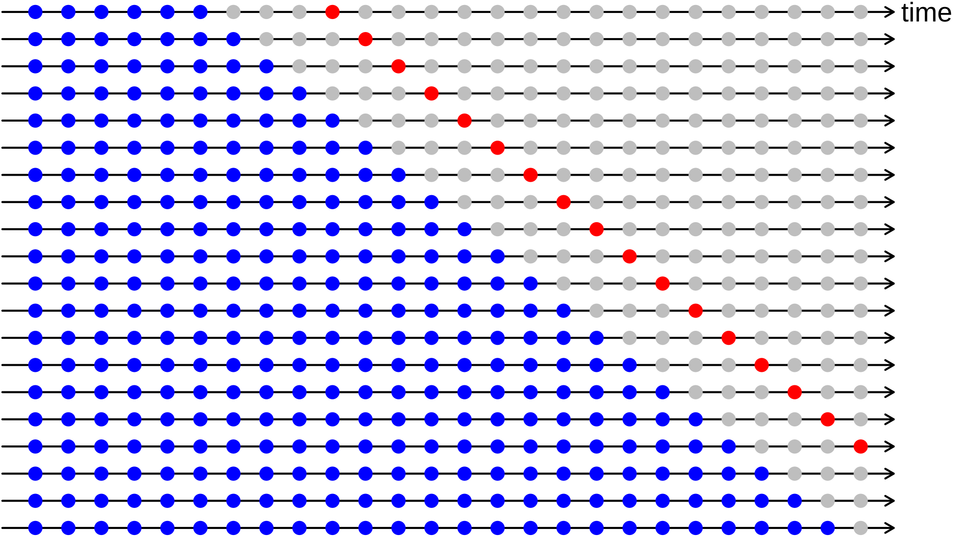


# Forecast evaluation



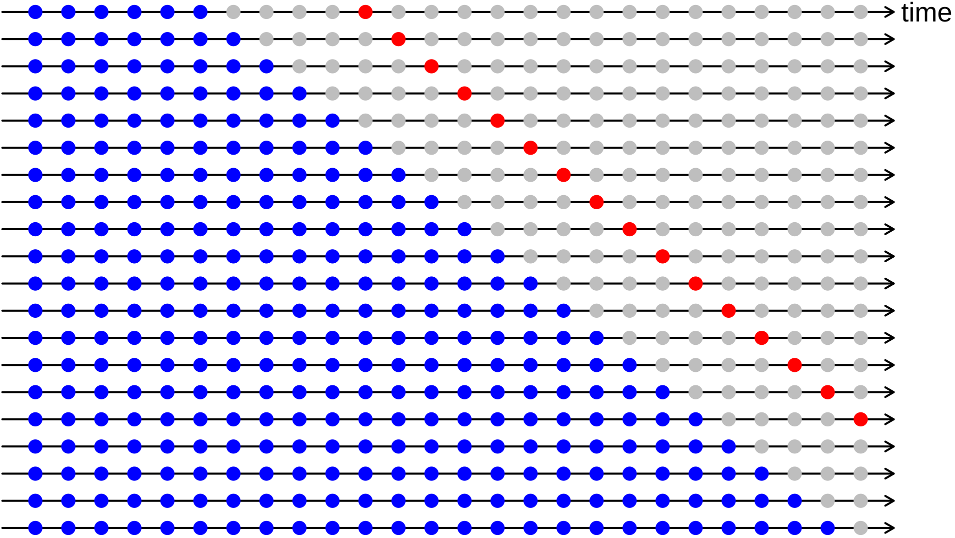
# Forecast evaluation

**Training sets** **Test set $h = 4$**



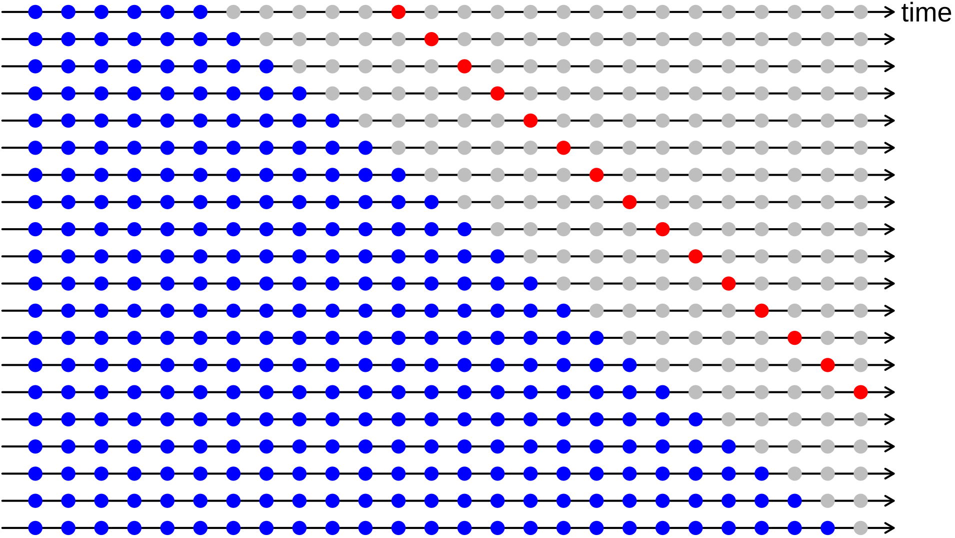
# Forecast evaluation

**Training sets** **Test set $h = 5$**



# Forecast evaluation

**Training sets** **Test set $h = 6$**

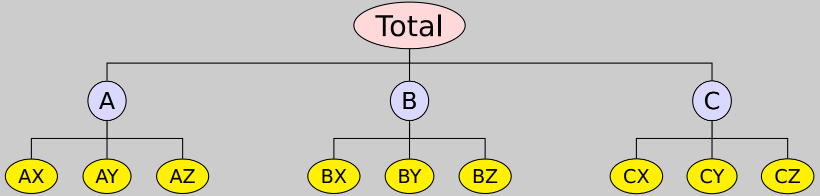


# Hierarchy: states, zones, regions

| RMSE | $h=1$ | $h=2$ | $h=3$ | $h=4$ | $h=5$ | $h=6$ | **Ave** |
|---|---|---|---|---|---|---|---|
| | | | **Australia** | | | | |
| Base | 1762.04 | 1770.29 | 1766.02 | 1818.82 | 1705.35 | 1721.17 | **1757.28** |
| Bottom | 1736.92 | 1742.69 | 1722.79 | 1752.74 | 1666.73 | 1687.43 | **1718.22** |
| OLS | 1747.60 | 1757.68 | 1751.77 | 1800.67 | 1686.00 | 1706.45 | **1741.69** |
| WLS | 1705.21 | 1715.87 | 1703.75 | 1729.56 | 1627.79 | 1661.24 | **1690.57** |
| GLS | 1704.64 | 1715.60 | 1705.31 | 1729.04 | 1626.36 | 1661.64 | **1690.43** |
| | | | **States** | | | | |
| Base | 399.77 | 404.16 | 401.92 | 407.26 | 395.38 | 401.17 | **401.61** |
| Bottom | 404.29 | 406.95 | 404.96 | 409.02 | 399.80 | 401.55 | **404.43** |
| OLS | 404.47 | 407.62 | 405.43 | 413.79 | 401.10 | 404.90 | **406.22** |
| WLS | 398.84 | 402.12 | 400.71 | 405.03 | 394.76 | 398.23 | **399.95** |
| GLS | 398.84 | 402.16 | 400.86 | 405.03 | 394.59 | 398.22 | **399.95** |
| | | | **Regions** | | | | |
| Base | 93.15 | 93.38 | 93.45 | 93.79 | 93.50 | 93.56 | **93.47** |
| Bottom | 93.15 | 93.38 | 93.45 | 93.79 | 93.50 | 93.56 | **93.47** |
| OLS | 93.28 | 93.53 | 93.64 | 94.17 | 93.78 | 93.88 | **93.71** |
| WLS | 93.02 | 93.32 | 93.38 | 93.72 | 93.39 | 93.53 | **93.39** |
| GLS | 92.98 | 93.27 | 93.34 | 93.66 | 93.34 | 93.46 | **93.34** |

# Outline

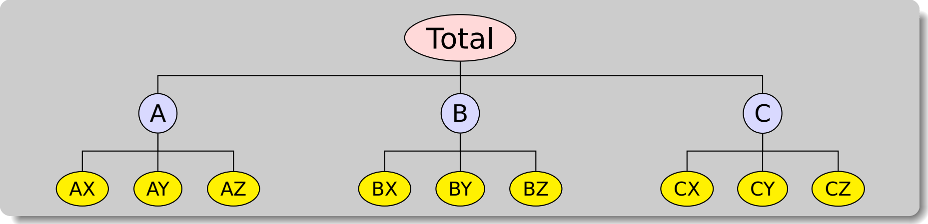
# Fast computation: hierarchical data




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

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

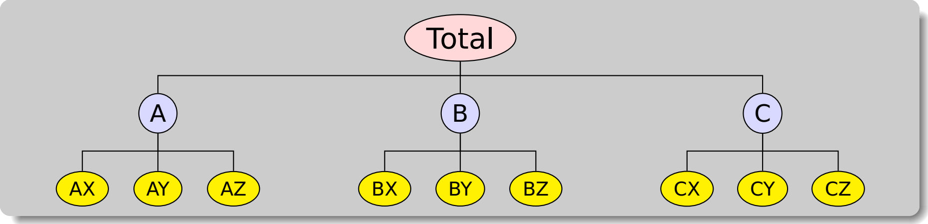
# Fast computation: hierarchical data




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

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

# Fast computation: hierarchical data

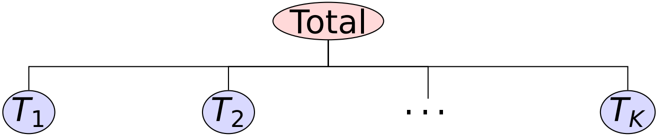



$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{B,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{C,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1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

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

# Fast computation: hierarchies

Think of the hierarchy as a tree of trees:



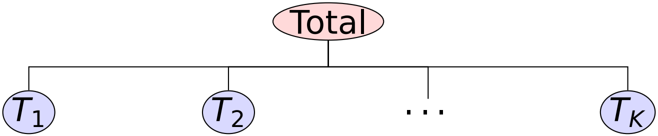

Then the summing matrix contains $k$ smaller summing matrices:

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

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

# Fast computation: hierarchies

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

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

- $\lambda_0$ is the top left element of $\Lambda$;
- $\Lambda_k$ is a block of $\Lambda$, 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

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

- $\lambda_0$ is the top left element of $\Lambda$;
- $\Lambda_k$ is a block of $\Lambda$, 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

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

- $\boldsymbol{S}_0$ can be partitioned into $K^2$ blocks, with the $(k,\ell)$ 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 recursively.

# 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 $(k,\ell)$ block (of dimension $n_k \times n_\ell$) being

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

- $\boldsymbol{J}_{n_k,n_\ell}$
- $c^{-1}$

$k$

- 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

A similar algorithm has been developed for grouped time series with two groups.

When the time series are not strictly hierarchical and have more than two grouping variables:

- Use sparse matrix storage and arithmetic.
- Use iterative approximation for inverting large sparse matrices.

Paige & Saunders (1982)
*ACM Trans. Math. Software*

# Fast computation

A similar algorithm has been developed for grouped time series with two groups.
When the time series are not strictly hierarchical and have more than two grouping variables:

- Use sparse matrix storage and arithmetic.
- Use iterative approximation for inverting large sparse matrices.

Paige & Saunders (1982)
*ACM Trans. Math. Software*

# Fast computation

A similar algorithm has been developed for grouped time series with two groups.

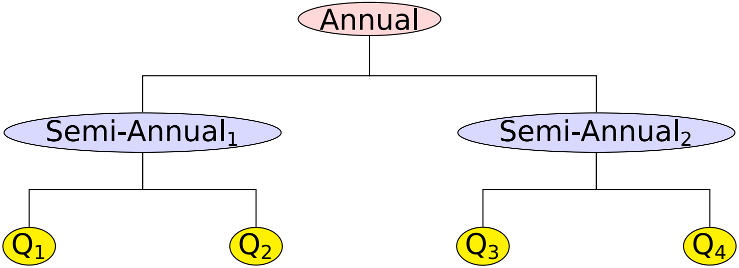
When the time series are not strictly hierarchical and have more than two grouping variables:

- Use sparse matrix storage and arithmetic.
- Use iterative approximation for inverting large sparse matrices.

Paige & Saunders (1982)
*ACM Trans. Math. Software*

# Outline

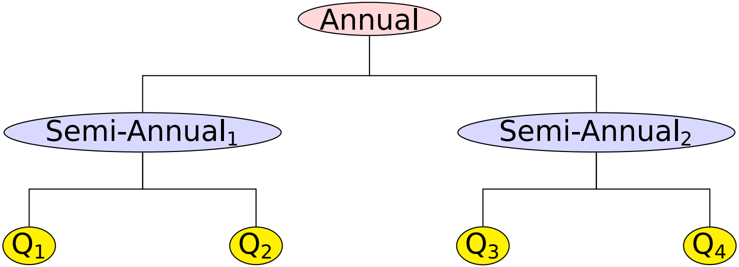
# Temporal hierarchies




**Basic idea:**

- Forecast series at each available frequency.
- Optimally combine forecasts within the same year.

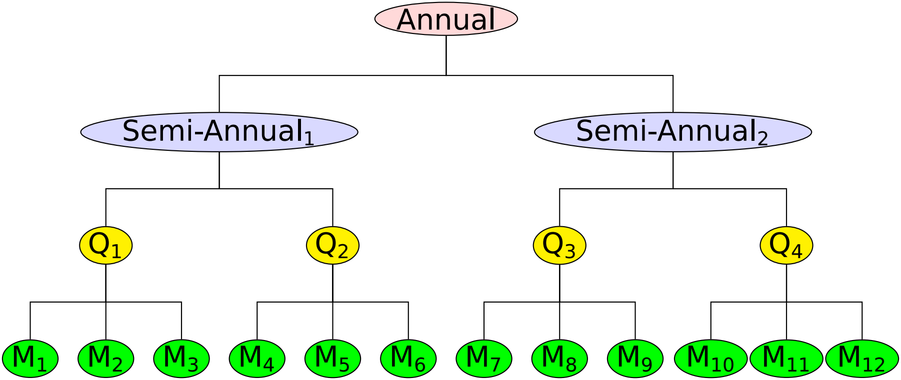
# Temporal hierarchies




**Basic idea:**

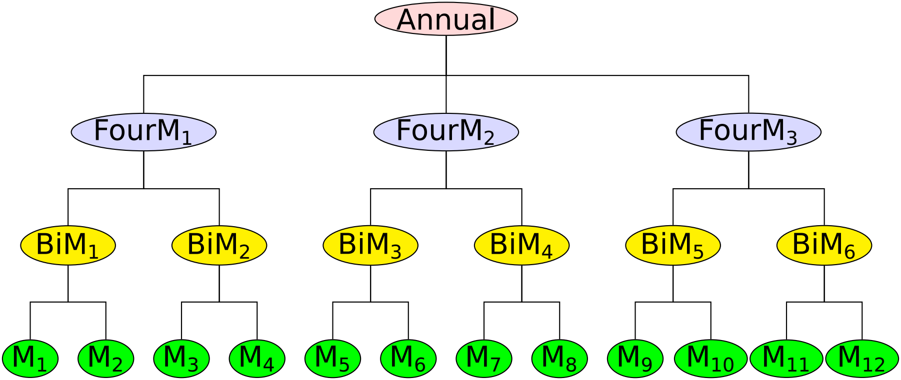
- Forecast series at each available frequency.
- Optimally combine forecasts within the same year.

# Monthly series



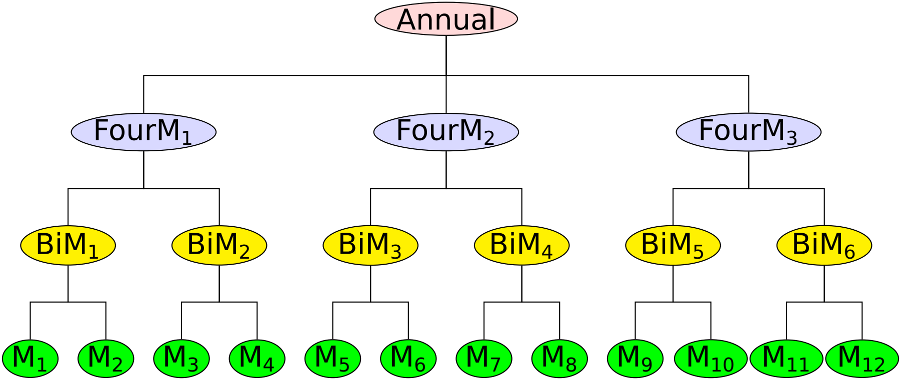

- $k = 2, 4, 12$ nodes
- $k = 3, 6, 12$ nodes
- Why not $k = 2, 3, 4, 6, 12$ nodes?

# Monthly series




- $k = 2, 4, 12$ nodes
- $k = 3, 6, 12$ nodes
- Why not $k = 2, 3, 4, 6, 12$ nodes?

# Monthly series




- $k = 2, 4, 12$ nodes
- $k = 3, 6, 12$ nodes
- Why not $k = 2, 3, 4, 6, 12$ nodes?

# Monthly data

$$
\underbrace{\begin{pmatrix} A \\ SemiA_1 \\ SemiA_2 \\ FourM_1 \\ FourM_2 \\ FourM_3 \\ Q_1 \\ \vdots \\ Q_4 \\ BiM_1 \\ \vdots \\ BiM_6 \\ M_1 \\ \vdots \\ M_{12} \end{pmatrix}}_{(28\times1)} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ & & & & & & \boldsymbol{I}_{12} & & & & & \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9 \\ M_{10} \\ M_{11} \\ M_{12} \end{pmatrix}}_{\boldsymbol{B}_t}
$$

# In general

For a time series $y_1, \dots, y_T$, observed at frequency $m$, we generate aggregate series

$$y_j^{[k]} = \sum_{t=1+(j-1)k}^{jk} y_t, \qquad \text{for } j = 1, \dots, \lfloor T/k \rfloor$$

- $k \in F(m) = \{\text{factors of } m\}$.
- A single unique hierarchy is only possible when there are no coprime pairs in $F(m)$.
- $M_k = m/k$ is seasonal period of aggregated series.

# In general

For a time series $y_1, \dots, y_T$, observed at frequency $m$, we generate aggregate series

$$y_j^{[k]} = \sum_{t=1+(j-1)k}^{jk} y_t, \qquad \text{for } j = 1, \dots, \lfloor T/k \rfloor$$

- $k \in F(m) = \{\text{factors of } m\}$.
- A single unique hierarchy is only possible when there are no coprime pairs in $F(m)$.
- $M_k = m/k$ is seasonal period of aggregated series.

# In general

For a time series $y_1, \dots, y_T$, observed at frequency $m$, we generate aggregate series

$$y_j^{[k]} = \sum_{t=1+(j-1)k}^{jk} y_t, \qquad \text{for } j = 1, \dots, \lfloor T/k \rfloor$$

- $k \in F(m) = \{\text{factors of } m\}$.
- A single unique hierarchy is only possible when there are no coprime pairs in $F(m)$.
- $M_k = m/k$ is seasonal period of aggregated series.

# WLS weights

**Hierarchy variance scaling** $\Lambda_H$: diagonal.

**Series variance scaling** $\Lambda_V$: elements equal within aggregation level.

**Structural scaling** $\Lambda_S = \text{diag}(\boldsymbol{S1})$: elements equal to # nodes at each level.

- Depends only on seasonal period $m$.
- Independent of data and model.
- Allows forecasts where no errors available.

**Quarterly example**

$$\Lambda_H = \text{diag}\left(\hat{\sigma}^2_A,\ \hat{\sigma}^2_{S_1},\ \hat{\sigma}^2_{S_2},\ \hat{\sigma}^2_{Q_1},\ \hat{\sigma}^2_{Q_2},\ \hat{\sigma}^2_{Q_3},\ \hat{\sigma}^2_{Q_4}\right)$$

$$\Lambda_V = \text{diag}\left(\hat{\sigma}^2_A,\ \hat{\sigma}^2_S,\ \hat{\sigma}^2_S,\ \hat{\sigma}^2_Q,\ \hat{\sigma}^2_Q,\ \hat{\sigma}^2_Q,\ \hat{\sigma}^2_Q\right)$$

$$\Lambda_S = \text{diag}\left(4, 2, 2, 1, 1, 1, 1\right)$$

# WLS weights

**Hierarchy variance scaling** $\Lambda_H$: diagonal.

**Series variance scaling** $\Lambda_V$: elements equal within aggregation level.

**Structural scaling** $\Lambda_S = \text{diag}(\boldsymbol{S}\mathbf{1})$: elements equal to # nodes at each level.

- Depends only on seasonal period $m$.
- Independent of data and model.
- Allows forecasts where no errors available.

**Quarterly example**

$$\Lambda_H = \text{diag}\big(\hat\sigma^2_A,\ \hat\sigma^2_{S_1},\ \hat\sigma^2_{S_2},\ \hat\sigma^2_{Q_1},\ \hat\sigma^2_{Q_2},\ \hat\sigma^2_{Q_3},\ \hat\sigma^2_{Q_4}\big)$$

$$\Lambda_V = \text{diag}\big(\hat\sigma^2_A,\ \hat\sigma^2_S,\ \hat\sigma^2_S,\ \hat\sigma^2_Q,\ \hat\sigma^2_Q,\ \hat\sigma^2_Q,\ \hat\sigma^2_Q\big)$$

$$\Lambda_S = \text{diag}\big(4, 2, 2, 1, 1, 1, 1\big)$$

# WLS weights

**Hierarchy variance scaling** $\Lambda_H$: diagonal.

**Series variance scaling** $\Lambda_V$: elements equal within aggregation level.

**Structural scaling** $\Lambda_S = \text{diag}(\boldsymbol{S1})$: elements equal to # nodes at each level.

- Depends only on seasonal period $m$.
- Independent of data and model.
- Allows forecasts where no errors available.

**Quarterly example**

$$\Lambda_H = \text{diag}\big(\hat\sigma^2_A,\ \hat\sigma^2_{S_1},\ \hat\sigma^2_{S_2},\ \hat\sigma^2_{Q_1},\ \hat\sigma^2_{Q_2},\ \hat\sigma^2_{Q_3},\ \hat\sigma^2_{Q_4}\big)$$

$$\Lambda_V = \text{diag}\big(\hat\sigma^2_A,\ \hat\sigma^2_S,\ \hat\sigma^2_S,\ \hat\sigma^2_Q,\ \hat\sigma^2_Q,\ \hat\sigma^2_Q,\ \hat\sigma^2_Q\big)$$

$$\Lambda_S = \text{diag}\big(4, 2, 2, 1, 1, 1, 1\big)$$

# WLS weights

**Hierarchy variance scaling** $\Lambda_H$: diagonal.

**Series variance scaling** $\Lambda_V$: elements equal within aggregation level.

**Structural scaling** $\Lambda_S = \text{diag}(\boldsymbol{S1})$: elements equal to # nodes at each level.

- Depends only on seasonal period $m$.
- Independent of data and model.
- Allows forecasts where no errors available.

**Quarterly example**

$$\Lambda_H = \text{diag}\big(\hat{\sigma}_A^2,\ \hat{\sigma}_{S_1}^2,\ \hat{\sigma}_{S_2}^2,\ \hat{\sigma}_{Q_1}^2,\ \hat{\sigma}_{Q_2}^2,\ \hat{\sigma}_{Q_3}^2,\ \hat{\sigma}_{Q_4}^2\big)$$

$$\Lambda_V = \text{diag}\big(\hat{\sigma}_A^2,\ \hat{\sigma}_S^2,\ \hat{\sigma}_S^2,\ \hat{\sigma}_Q^2,\ \hat{\sigma}_Q^2,\ \hat{\sigma}_Q^2,\ \hat{\sigma}_Q^2\big)$$

$$\Lambda_S = \text{diag}\big(4, 2, 2, 1, 1, 1, 1\big)$$

# WLS weights

**Hierarchy variance scaling** $\Lambda_H$: diagonal.

**Series variance scaling** $\Lambda_V$: elements equal within aggregation level.

**Structural scaling** $\Lambda_S = \text{diag}(\boldsymbol{S1})$: elements equal to # nodes at each level.

- Depends only on seasonal period $m$.
- Independent of data and model.
- Allows forecasts where no errors available.

## Quarterly example

$$\Lambda_H = \text{diag}\left(\hat{\sigma}_A^2,\ \hat{\sigma}_{S_1}^2,\ \hat{\sigma}_{S_2}^2,\ \hat{\sigma}_{Q_1}^2,\ \hat{\sigma}_{Q_2}^2,\ \hat{\sigma}_{Q_3}^2,\ \hat{\sigma}_{Q_4}^2\right)$$

$$\Lambda_V = \text{diag}\left(\hat{\sigma}_A^2,\ \hat{\sigma}_S^2,\ \hat{\sigma}_S^2,\ \hat{\sigma}_Q^2,\ \hat{\sigma}_Q^2,\ \hat{\sigma}_Q^2,\ \hat{\sigma}_Q^2\right)$$

$$\Lambda_S = \text{diag}\left(4, 2, 2, 1, 1, 1, 1\right)$$

# Experimental setup:

- M3 forecasting competition (Makridakis and Hibon, 2000, *IJF*). In total 3003 series.
- 1,428 monthly series with a test sample of 12 observations each.
- 756 quarterly series with a test sample of 8 observations each.
- Forecast each series with ETS models.

# Experimental setup:

- M3 forecasting competition (Makridakis and Hibon, 2000, *IJF*). In total 3003 series.
- 1,428 monthly series with a test sample of 12 observations each.
- 756 quarterly series with a test sample of 8 observations each.
- Forecast each series with ETS models.

# Experimental setup:

- M3 forecasting competition (Makridakis and Hibon, 2000, *IJF*). In total 3003 series.
- 1,428 monthly series with a test sample of 12 observations each.
- 756 quarterly series with a test sample of 8 observations each.
- Forecast each series with ETS models.

# Experimental setup:

- M3 forecasting competition (Makridakis and Hibon, 2000, *IJF*). In total 3003 series.
- 1,428 monthly series with a test sample of 12 observations each.
- 756 quarterly series with a test sample of 8 observations each.
- Forecast each series with ETS models.

# Results: Monthly

## MAE percent difference relative to base

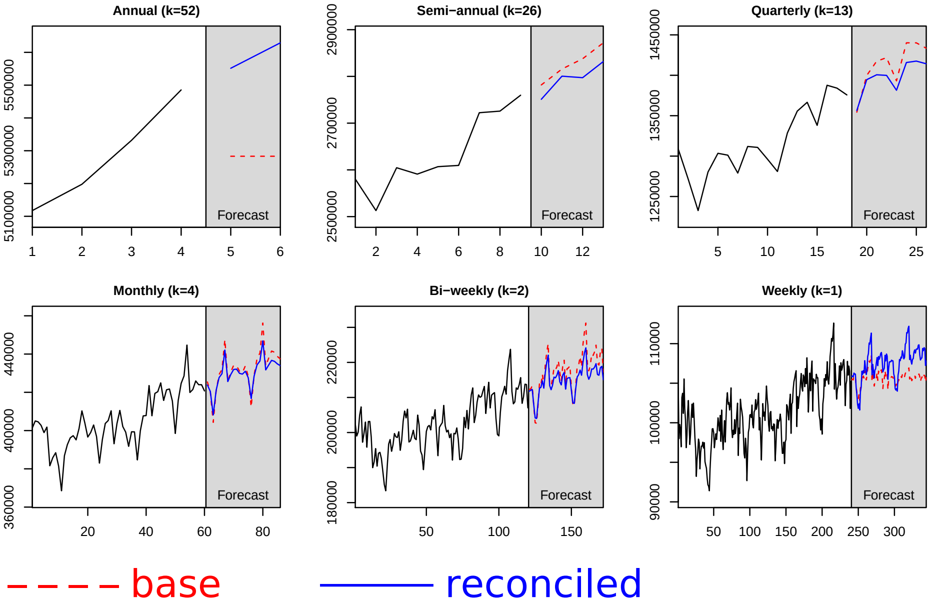
| | max $h$ | BU | $\mathrm{WLS}_H$ | $\mathrm{WLS}_V$ | $\mathrm{WLS}_S$ |
|---|---|---|---|---|---|
| Annual | 1 | −19.6 | −22.0 | −22.0 | −25.1 |
| Semi-annual | 3 | 0.6 | −4.0 | −3.6 | −5.4 |
| Four-monthly | 4 | 2.0 | −2.4 | −2.2 | −3.0 |
| Quarterly | 6 | 2.4 | −1.6 | −1.7 | −2.8 |
| Bi-monthly | 9 | 0.7 | −2.9 | −3.3 | −4.3 |
| Monthly | 18 | 0.0 | −2.2 | −3.2 | −3.9 |

# Results: Quarterly

## MAE percent difference relative to base

| | max $h$ | BU | $\text{WLS}_H$ | $\text{WLS}_V$ | $\text{WLS}_S$ |
|---|---|---|---|---|---|
| Annual | 1 | $-20.9$ | -22.7 | $-22.8$ | -22.7 |
| Semi-annual | 3 | $-4.5$ | $-6.0$ | $-6.2$ | -4.8 |
| Quarterly | 6 | 0.0 | $-0.2$ | $-1.1$ | -0.3 |

# UK Accidents and Emergency Demand



# UK Accidents and Emergency Demand

1. Type 1 Departments — Major A&E
2. Type 2 Departments — Single Specialty
3. Type 3 Departments — Other A&E/Minor Injury
4. Total Attendances
5. Type 1 Departments — Major A&E $>$ 4 hrs
6. Type 2 Departments — Single Specialty $>$ 4 hrs
7. Type 3 Departments — Other A&E/Minor Injury $>$ 4 hrs
8. Total Attendances $>$ 4 hrs
9. Emergency Admissions via Type 1 A&E
10. Total Emergency Admissions via A&E
11. Other Emergency Admissions (i.e., not via A&E)
12. Total Emergency Admissions
13. Number of patients spending $>$ 4 hrs from decision to admission

# UK Accidents and Emergency Demand

- **Minimum training set**: all data except the last year
- Base forecasts using `auto.arima()`.
- Reconciled using $\text{WLS}_V$.
- Mean Absolute Scaled Errors for 1, 4 and 13 weeks ahead using a rolling origin.

| Aggr. Level | $h$ | Base | Reconciled | Change |
|---|---|---|---|---|
| Weekly | 1 | 1.6 | 1.3 | −17.2% |
| Weekly | 4 | 1.9 | 1.5 | −18.6% |
| Weekly | 13 | 2.3 | 1.9 | −16.2% |
| Weekly | 1–52 | 2.0 | 1.9 | −5.0% |
| Annual | 1 | 3.4 | 1.9 | −42.9% |

# UK Accidents and Emergency Demand

- **Minimum training set**: all data except the last year
- Base forecasts using `auto.arima()`.
- Reconciled using $\text{WLS}_V$.
- Mean Absolute Scaled Errors for 1, 4 and 13 weeks ahead using a rolling origin.

| Aggr. Level | $h$ | Base | Reconciled | Change |
|---|---|---|---|---|
| Weekly | 1 | 1.6 | 1.3 | −17.2% |
| Weekly | 4 | 1.9 | 1.5 | −18.6% |
| Weekly | 13 | 2.3 | 1.9 | −16.2% |
| Weekly | 1–52 | 2.0 | 1.9 | −5.0% |
| Annual | 1 | 3.4 | 1.9 | −42.9% |

# UK Accidents and Emergency Demand

- **Minimum training set**: all data except the last year
- Base forecasts using `auto.arima()`.
- Reconciled using WLS$_V$.
- Mean Absolute Scaled Errors for 1, 4 and 13 weeks ahead using a rolling origin.

| Aggr. Level | $h$ | Base | Reconciled | Change |
|---|---|---|---|---|
| Weekly | 1 | 1.6 | 1.3 | −17.2% |
| Weekly | 4 | 1.9 | 1.5 | −18.6% |
| Weekly | 13 | 2.3 | 1.9 | −16.2% |
| Weekly | 1–52 | 2.0 | 1.9 | −5.0% |
| Annual | 1 | 3.4 | 1.9 | −42.9% |

# UK Accidents and Emergency Demand

- **Minimum training set**: all data except the last year
- Base forecasts using `auto.arima()`.
- Reconciled using $\text{WLS}_V$.
- Mean Absolute Scaled Errors for 1, 4 and 13 weeks ahead using a rolling origin.

| Aggr. Level | $h$ | Base | Reconciled | Change |
|---|---|---|---|---|
| Weekly | 1 | 1.6 | 1.3 | −17.2% |
| Weekly | 4 | 1.9 | 1.5 | −18.6% |
| Weekly | 13 | 2.3 | 1.9 | −16.2% |
| Weekly | 1–52 | 2.0 | 1.9 | −5.0% |
| Annual | 1 | 3.4 | 1.9 | −42.9% |

# UK Accidents and Emergency Demand

- **Minimum training set**: all data except the last year
- Base forecasts using `auto.arima()`.
- Reconciled using $\text{WLS}_V$.
- Mean Absolute Scaled Errors for 1, 4 and 13 weeks ahead using a rolling origin.

| Aggr. Level | $h$ | Base | Reconciled | Change |
|---|---|---|---|---|
| Weekly | 1 | 1.6 | 1.3 | −17.2% |
| Weekly | 4 | 1.9 | 1.5 | −18.6% |
| Weekly | 13 | 2.3 | 1.9 | −16.2% |
| Weekly | 1–52 | 2.0 | 1.9 | −5.0% |
| Annual | 1 | 3.4 | 1.9 | −42.9% |

# UK Accidents and Emergency Demand

- **Minimum training set**: all data except the last year
- Base forecasts using `auto.arima()`.
- Reconciled using $\text{WLS}_V$.
- Mean Absolute Scaled Errors for 1, 4 and 13 weeks ahead using a rolling origin.

| Aggr. Level | $h$ | Base | Reconciled | Change |
|---|---|---|---|---|
| Weekly | 1 | 1.6 | 1.3 | −17.2% |
| Weekly | 4 | 1.9 | 1.5 | −18.6% |
| Weekly | 13 | 2.3 | 1.9 | −16.2% |
| Weekly | 1–52 | 2.0 | 1.9 | −5.0% |
| Annual | 1 | 3.4 | 1.9 | −42.9% |

# Outline

# hts package for R

**hts: Hierarchical and grouped time series**

Methods for analysing and forecasting hierarchical and grouped time series

| | |
|---|---|
| Version: | 4.5 |
| Depends: | forecast ($\geq$ 5.0), SparseM |
| Imports: | parallel, utils |
| Published: | 2014-12-09 |
| Author: | Rob J Hyndman, Earo Wang and Alan Lee |
| Maintainer: | Rob J Hyndman <Rob.Hyndman at monash.edu> |
| BugReports: | `https://github.com/robjhyndman/hts/issues` |
| License: | GPL ($\geq$ 2) |

# Example using R

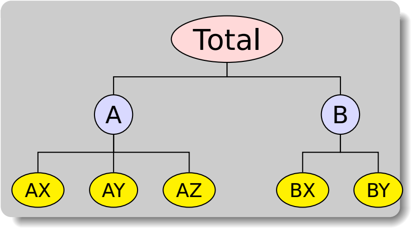
```
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))
```

# Example using R

```
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))
```



# Example using R

```
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))

# Forecast 10-step-ahead using WLS combination method
# ETS used for each series by default
fc <- forecast(y, h=10)
```

# forecast.gts function

**Usage**

```
forecast(object, h,
  method = c("comb", "bu", "mo", "tdgsf", "tdgsa", "tdfp"),
  fmethod = c("ets", "rw", "arima"),
  weights = c("sd", "none", "nseries"),
  positive = FALSE,
  parallel = FALSE, num.cores = 2, ...)
```

**Arguments**

| | |
|---|---|
| `object` | Hierarchical time series object of class `gts`. |
| `h` | Forecast horizon |
| `method` | Method for distributing forecasts within the hierarchy. |
| `fmethod` | Forecasting method to use |
| `positive` | If TRUE, forecasts are forced to be strictly positive |
| `weights` | Weights used for "optimal combination" method. When weights = "sd", it takes account of the standard deviation of forecasts. |
| `parallel` | If TRUE, allow parallel processing |
| `num.cores` | If parallel = TRUE, specify how many cores are going to be used |

# Outline

# References



RJ Hyndman, RA Ahmed, G Athanasopoulos, and HL Shang (2011). "Optimal combination forecasts for hierarchical time series". *Computational statistics & data analysis* **55**(9), 2579–2589.


RJ Hyndman, AJ Lee, and E Wang (2014). *Fast computation of reconciled forecasts for hierarchical and grouped time series*. Working paper 17/14. Monash University


SL Wickramasuriya, G Athanasopoulos, and RJ Hyndman (2015). *Forecasting hierarchical and grouped time series through trace minimization*. Working paper 15/15. Monash University


G Athanasopoulos, RJ Hyndman, N Kourentzes, and F Petropoulos (2015). *Forecasting with temporal hierarchies*. Working paper. Monash University

RJ Hyndman, AJ Lee, and E Wang (2015). *hts: Hierarchical and grouped time series*. R package v4.5 on CRAN.

# References


RJ Hyndman, RA Ahmed, G Athanasopoulos, and HL Shang (2011). “Optimal combination forecasts for hierarchical time series”. *Computational statistics & data analysis* **55**(9), 2579–2589.

RJ Hyndman, AJ Lee, and E Wang (2014). *Fast computation of reconciled forecasts for hierarchical and grouped time series*. Working paper 17/14. Monash University

SL Wickramasuriya, G Athanasopoulos, and RJ Hyndman (2015). *Forecasting hierarchical and grouped time series through trace minimization*. Working paper 15/15. Monash University

*grouped time series*. R package v4.5 on CRAN.


- Papers and R code: **robjhyndman.com**
- Email: **Rob.Hyndman@monash.edu**