MONASH University

MONASH BUSINESS SCHOOL

2017 Beijing Workshop on Forecasting

Hierarchical Forecasting

Rob J Hyndman

robjhyndman.com/beijing2017

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

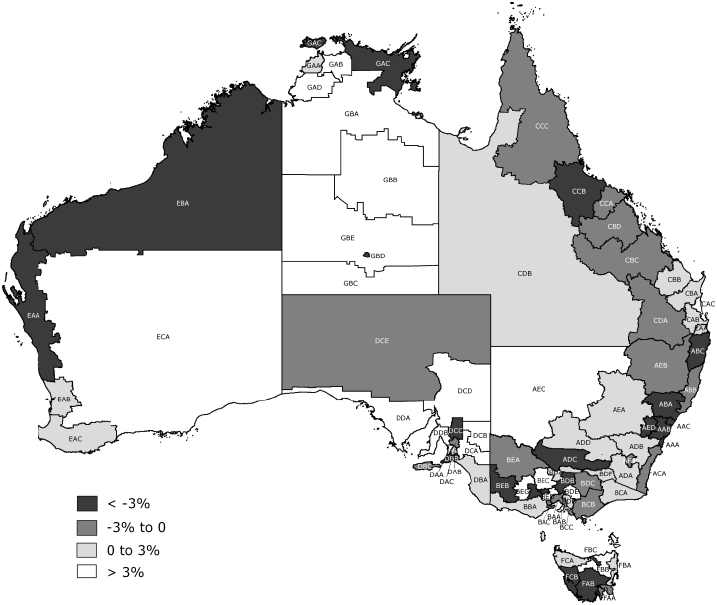
Australia and New Zealand Standard Classification of Occupations

- 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

> 3%

3. PBS sales

3. 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

3. PBS sales

ATC drug classification

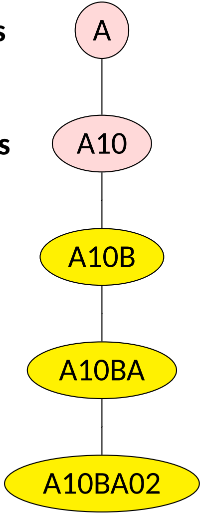
14 classes — Alimentary tract and metabolism

84 classes — Drugs used in diabetes

Blood glucose lowering drugs

Biguanides

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

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

Spectacle sales

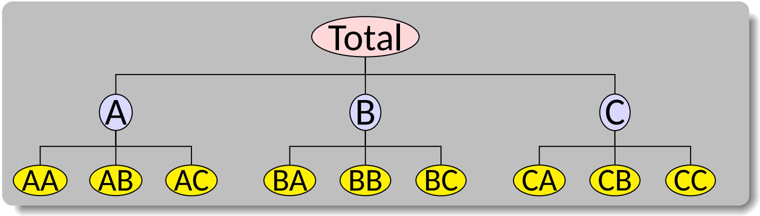
- Monthly UK sales data from 2000 – 2014
- Provided by a large spectacle manufacturer
- Split by brand (26), gender (3), price range (6), materials (4), and stores (600)
- About 1 million bottom-level series

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

Hierarchical time series

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

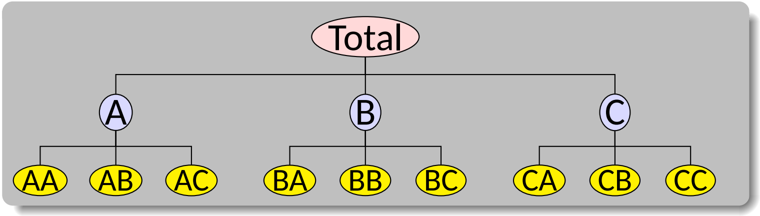


Examples

- Labour turnover by occupation
- Pharmaceutical sales
- 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.

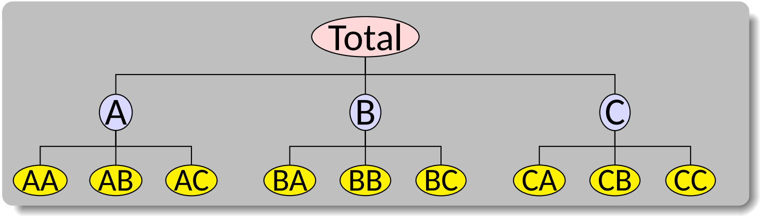


Examples

- Labour turnover by occupation
- Pharmaceutical sales
- 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.

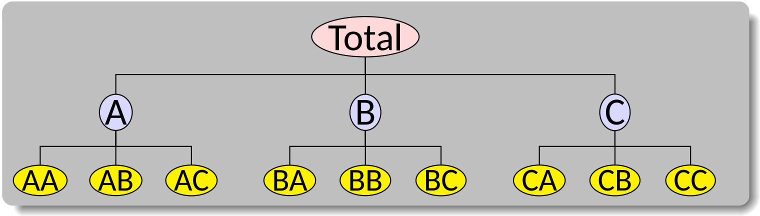


Examples

- Labour turnover by occupation
- Pharmaceutical sales
- 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.

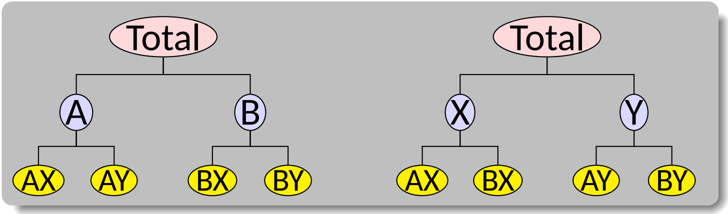


Examples

- Labour turnover by occupation
- Pharmaceutical sales
- 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.

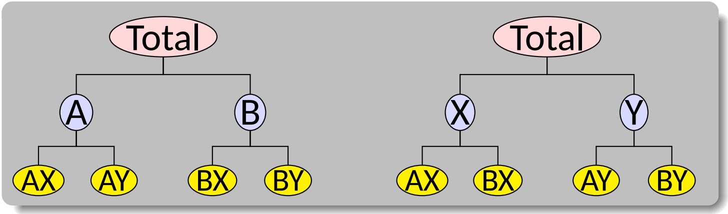


Examples

- Labour turnover by occupation and state
- Spectacle sales by brand, gender, stores, etc.
- 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.

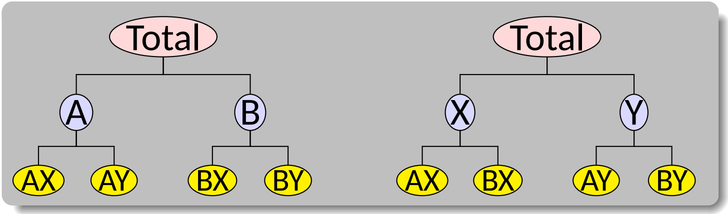


Examples

- Labour turnover by occupation and state
- Spectacle sales by brand, gender, stores, etc.
- 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.

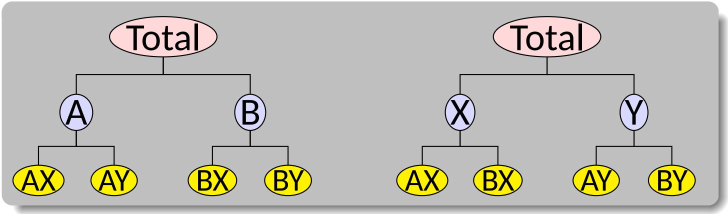


Examples

- Labour turnover by occupation and state
- Spectacle sales by brand, gender, stores, etc.
- 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
- Spectacle sales by brand, gender, stores, etc.
- Tourism by state and purpose of travel

The problem

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

The problem

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

The solution

1. Forecast all series at all levels of aggregation using an automatic forecasting algorithm (e.g., `ets`, `auto.arima`, ...)
2. Reconcile the resulting forecasts so they add up correctly using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).
3. This is all available in the **hts** package in R.

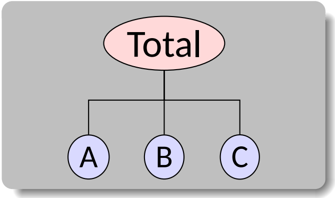
The solution

1. Forecast all series at all levels of aggregation using an automatic forecasting algorithm (e.g., `ets`, `auto.arima`, ...)
2. Reconcile the resulting forecasts so they add up correctly using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).
3. This is all available in the **hts** package in R.

The solution

1. Forecast all series at all levels of aggregation using an automatic forecasting algorithm (e.g., `ets`, `auto.arima`, `...`)
2. Reconcile the resulting forecasts so they add up correctly using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).
3. This is all available in the **hts** package in R.

Hierarchical time series

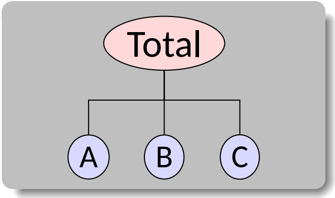


y_t : observed aggregate of all series at time t.

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

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

Hierarchical time series

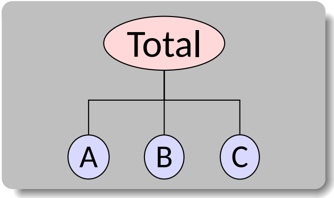


y_t : observed aggregate of all series at time t.

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

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

Hierarchical time series



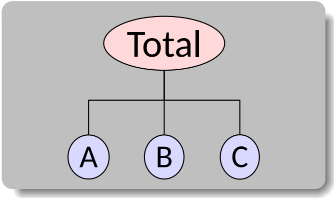
y_t : observed aggregate of all series at time t.

$y_{X,t}$: 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



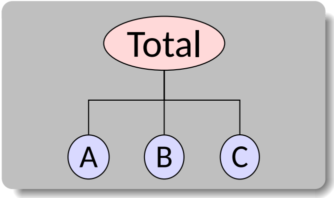
y_t : observed aggregate of all series at time t.

$y_{X,t}$: 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



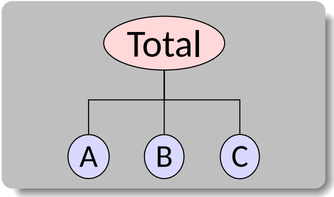
y_t : observed aggregate of all series at time t.

$y_{X,t}$: 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.

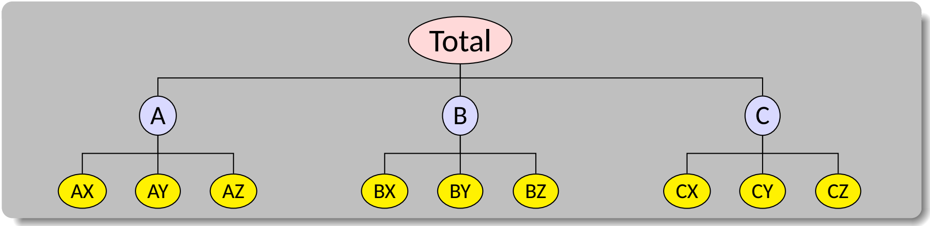
$y_{X,t}$: observation on series X at time t.

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

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

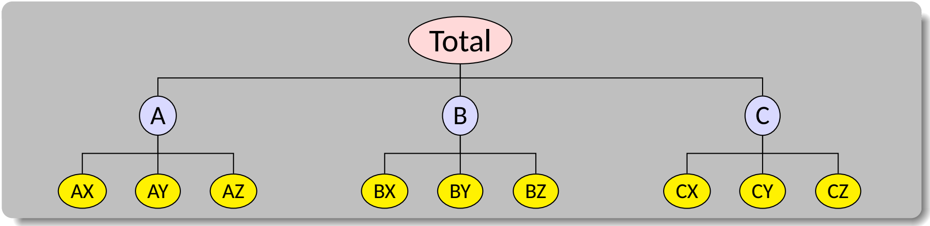
$$\boxed{\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t}$$

Hierarchical time series

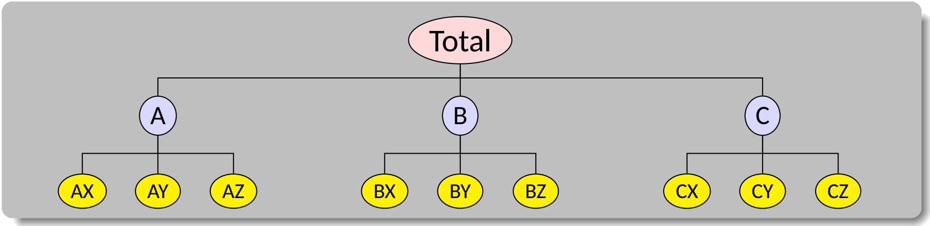


$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0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

Hierarchical time series



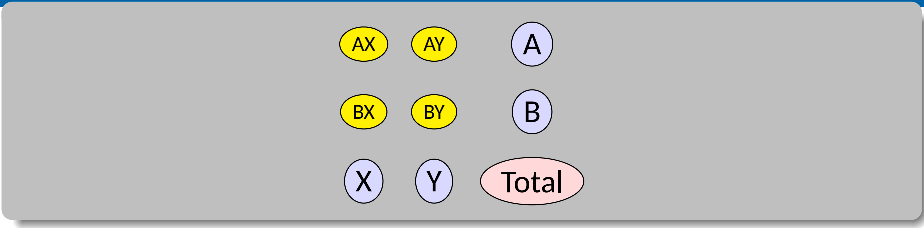
$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0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



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

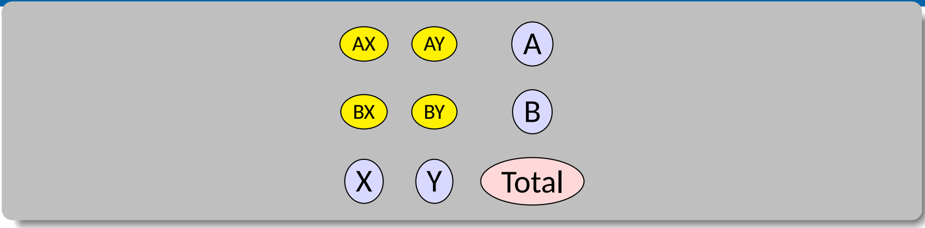
$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

Grouped data



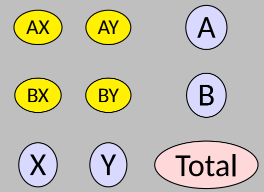
$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{X,t} \\ y_{Y,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0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

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

1 Hierarchical and grouped time series

2 **Forecast reconciliation**

3 Fast computational tricks

Hierarchical/grouped time series

- Forecasts should be “coherent”, unbiased, minimum variance.
- Existing methods:
  - Bottom-up
  - Top-down
  - Middle-out
- How to compute forecast intervals?
- Most research is concerned about relative performance of existing methods.

Hierarchical/grouped time series

- Forecasts should be "coherent", unbiased, minimum variance.
- Existing methods:
  - Bottom-up
  - Top-down
  - Middle-out
- How to compute forecast intervals?
- Most research is concerned about relative performance of existing methods.

Hierarchical/grouped time series

- Forecasts should be "coherent", unbiased, minimum variance.
- Existing methods:
  - Bottom-up
  - Top-down
  - Middle-out
- How to compute forecast intervals?
- Most research is concerned about relative performance of existing methods.

Hierarchical/grouped time series

- Forecasts should be "coherent", unbiased, minimum variance.
- Existing methods:
  - Bottom-up
  - Top-down
  - Middle-out
- How to compute forecast intervals?
- Most research is concerned about relative performance of existing methods.

Hierarchical/grouped time series

- Forecasts should be “coherent”, unbiased, minimum variance.
- Existing methods:
  - Bottom-up
  - Top-down
  - Middle-out
- How to compute forecast intervals?
- Most research is concerned about relative performance of existing methods.

Hierarchical/grouped time series

- Forecasts should be "coherent", unbiased, minimum variance.
- Existing methods:
  - Bottom-up
  - Top-down
  - Middle-out
- How to compute forecast intervals?
- Most research is concerned about relative performance of existing methods.

Hierarchical/grouped time series

- Forecasts should be "coherent", unbiased, minimum variance.
- Existing methods:
  - Bottom-up
  - Top-down
  - Middle-out
- How to compute forecast intervals?
- Most research is concerned about relative performance of existing methods.

Top-down method

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

Top-down method

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

Top-down method

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

Top-down method

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

Top-down method

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

Top-down method

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 method

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

Bottom-up method

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

Bottom-up method

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

Bottom-up method

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

Bottom-up method

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 $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$.
(In general, they will 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 $\boldsymbol{y}_t$.
(In general, they will 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 $\boldsymbol{y}_t$.
(In general, they will 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 $\boldsymbol{y}_t$.
(In general, they will 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 $\boldsymbol{y}_t$.
(In general, they will 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

Bottom-up forecasts

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

Bottom-up forecasts are obtained using

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

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

- $\boldsymbol{P}$ matrix extracts only bottom-level forecasts from $\hat{\boldsymbol{y}}_n(h)$
- $\boldsymbol{S}$ adds them up to give the bottom-up forecasts.

Bottom-up forecasts

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

Bottom-up forecasts are obtained using

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

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

- $\boldsymbol{P}$ matrix extracts only bottom-level forecasts from $\hat{\boldsymbol{y}}_n(h)$
- $\boldsymbol{S}$ adds them up to give the bottom-up forecasts.

Bottom-up forecasts

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

Bottom-up forecasts are obtained using

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

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

- $\boldsymbol{P}$ matrix extracts only bottom-level forecasts from $\hat{\boldsymbol{y}}_n(h)$
- $\boldsymbol{S}$ adds them up to give the bottom-up forecasts.

Top-down forecasts

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

Top-down forecasts are obtained using

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

where $\boldsymbol{p} = [p_1, p_2, \dots, p_{m_K}]'$ is a vector of proportions that sum to one.

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

Top-down forecasts

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

Top-down forecasts are obtained using

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

where $\boldsymbol{p} = [p_1, p_2, \dots, p_{m_K}]'$ is a vector of proportions that sum to one.

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

Top-down forecasts

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

Top-down forecasts are obtained using

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

where $\boldsymbol{p} = [p_1, p_2, \dots, p_{m_K}]'$ is a vector of proportions that sum to one.

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

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:

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

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

Reconciled forecast are unbiased if and only if $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

True for bottom-up, but *not* top-down or middle-out.

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:

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

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

Reconciled forecast are unbiased if and only if $\boldsymbol{SPS} = \boldsymbol{S}$.

True for bottom-up, but *not* top-down or middle-out.

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:

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

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

Reconciled forecast are unbiased if and only if $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

True for bottom-up, but *not* top-down or middle-out.

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:

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

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

Reconciled forecast are unbiased if and only if $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

True for bottom-up, but *not* top-down or middle-out.

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:

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

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

Reconciled forecast are unbiased if and only if $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

True for bottom-up, but *not* top-down or middle-out.

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:

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

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

Reconciled forecast are unbiased if and only if $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

True for bottom-up, but *not* top-down or middle-out.

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:

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

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

Reconciled forecast are unbiased if and only if $\boldsymbol{SPS} = \boldsymbol{S}$.

True for bottom-up, but *not* top-down or middle-out.

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:

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

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

Reconciled forecast are unbiased if and only if $\boldsymbol{S}\boldsymbol{P}\boldsymbol{S} = \boldsymbol{S}$.

True for bottom-up, but *not* top-down or middle-out.

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

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

Then the error variance of the corresponding reconciled forecasts is

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

This is a general result for all existing methods.

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

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

Then the error variance of the corresponding reconciled forecasts is

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

This is a general result for all existing methods.

Optimal forecast reconciliation

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

Theorem: MinT Reconciliation

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

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{SP}\Sigma_h \boldsymbol{P}'\boldsymbol{S}']$$

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

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

Reconciled forecasts

Base forecasts

- Assume that $\Sigma_h = k_h\Sigma_1$ to simplify computations.

Optimal forecast reconciliation

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

Theorem: MinT Reconciliation

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

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{SP}\Sigma_h \boldsymbol{P}'\boldsymbol{S}']$$

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

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

Reconciled forecasts

Base forecasts

- Assume that $\Sigma_h = k_h\Sigma_1$ to simplify computations.

Optimal forecast reconciliation

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

Theorem: MinT Reconciliation

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

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

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

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

Reconciled forecasts — Base forecasts

- Assume that $\Sigma_h = k_h\Sigma_1$ to simplify computations.

Optimal forecast reconciliation

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

Theorem: MinT Reconciliation

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

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{SP}\Sigma_h\boldsymbol{P}'\boldsymbol{S}']$$

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

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

Reconciled forecasts

Base forecasts

- Assume that $\Sigma_h = k_h\Sigma_1$ to simplify computations.

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{-1}\hat{\boldsymbol{y}}_n(h)$$

Reconciled forecasts | Base forecasts

Solution 1: OLS

- Assume $\Sigma_1 \approx kI$.

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

- Reconciliation does not depend on data
- Works surprisingly well.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{-1}\hat{\boldsymbol{y}}_n(h)$$

Reconciled forecasts

Base forecasts

Solution 1: OLS

- Assume $\Sigma_1 \approx k\boldsymbol{I}$.

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

- Reconciliation does not depend on data
- Works surprisingly well.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{-1}\hat{\boldsymbol{y}}_n(h)$$

Reconciled forecasts

Base forecasts

Solution 1: OLS

- Assume $\Sigma_1 \approx k\boldsymbol{I}$.

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

- Reconciliation does not depend on data
- Works surprisingly well.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{-1}\hat{\boldsymbol{y}}_n(h)$$

Reconciled forecasts

Base forecasts

Solution 1: OLS

- Assume $\Sigma_1 \approx k\boldsymbol{I}$.

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

- Reconciliation does not depend on data
- Works surprisingly well.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{-1}\hat{\boldsymbol{y}}_n(h)$$

Reconciled forecasts

Base forecasts

Solution 1: OLS

- Assume $\Sigma_1 \approx k\boldsymbol{I}$.

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

- Reconciliation does not depend on data
- Works surprisingly well.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{-1}\hat{\boldsymbol{y}}_n(h)$$

Reconciled forecasts

Base forecasts

Solution 1: OLS

- Assume $\Sigma_1 \approx k\boldsymbol{I}$.

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

- Reconciliation does not depend on data
- Works surprisingly well.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Solution 2: WLS

- Approximate Σ_1 by its diagonal.
- Easy to estimate, and places weight where we have best forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Solution 2: WLS

- Approximate Σ_1 by its diagonal.
- Easy to estimate, and places weight where we have best forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Solution 2: WLS

- Approximate Σ_1 by its diagonal.
- Easy to estimate, and places weight where we have best forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Solution 2: WLS

- Approximate Σ_1 by its diagonal.
- Easy to estimate, and places weight where we have best forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Solution 3: GLS

- Estimate Σ_1 using shrinkage to the diagonal.
- Allows for covariances.
- Difficult to compute for large numbers of time series.

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Solution 3: GLS

- Estimate Σ_1 using shrinkage to the diagonal.
- Allows for covariances.
- Difficult to compute for large numbers of time series.

Optimal forecast reconciliation

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

Reconciled forecasts

Base forecasts

Solution 3: GLS

- Estimate Σ_1 using shrinkage to the diagonal.
- Allows for covariances.
- Difficult to compute for large numbers of time series.

Optimal forecast reconciliation

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

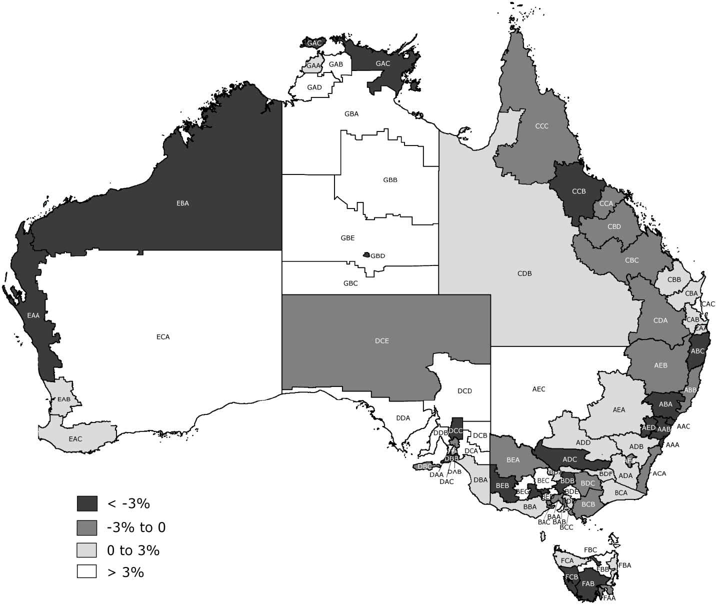
Reconciled forecasts

Base forecasts

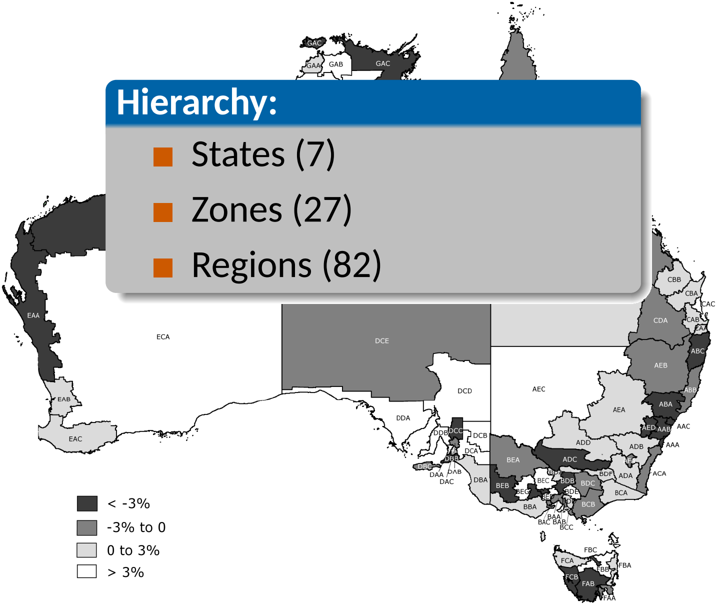
Solution 3: GLS

- Estimate Σ_1 using shrinkage to the diagonal.
- Allows for covariances.
- Difficult to compute for large numbers of time series.

Australian tourism



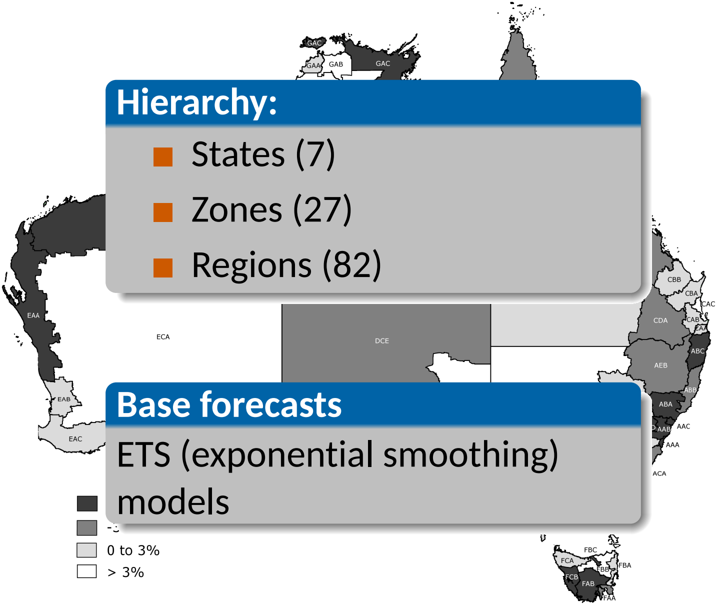
Australian tourism



Hierarchy:

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

Australian tourism



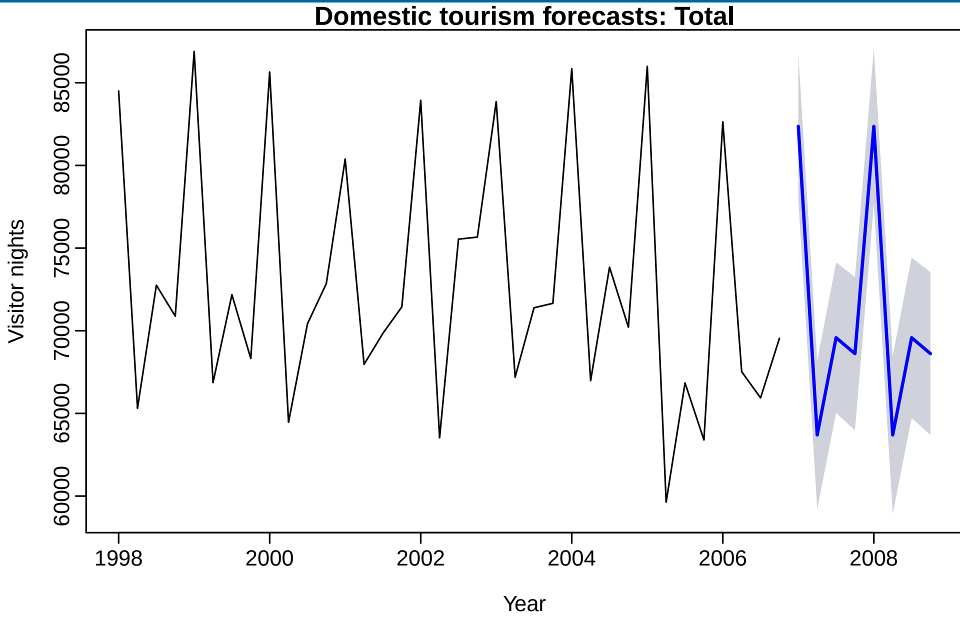
Hierarchy:

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

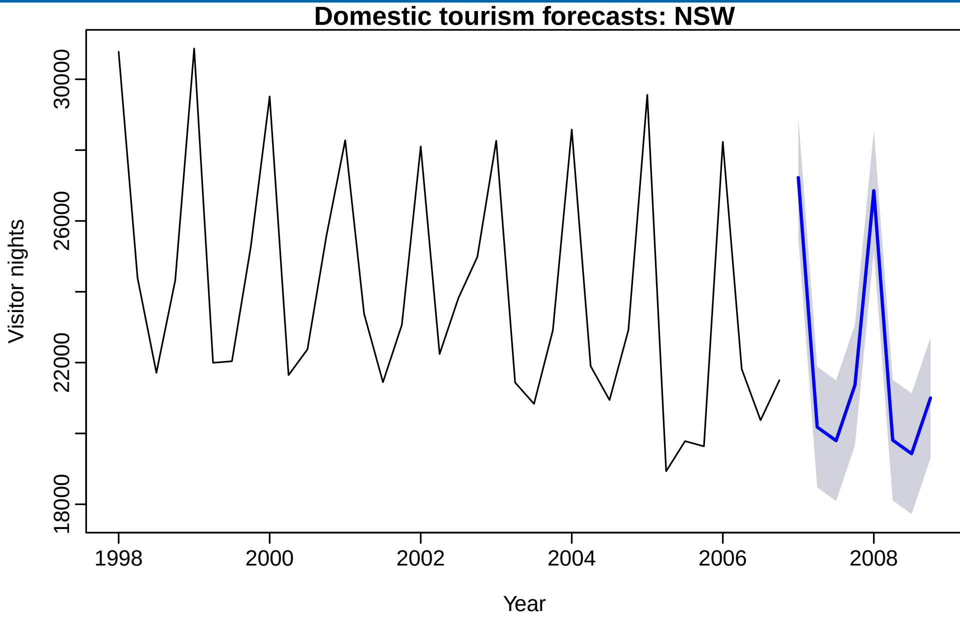
Base forecasts

ETS (exponential smoothing) models

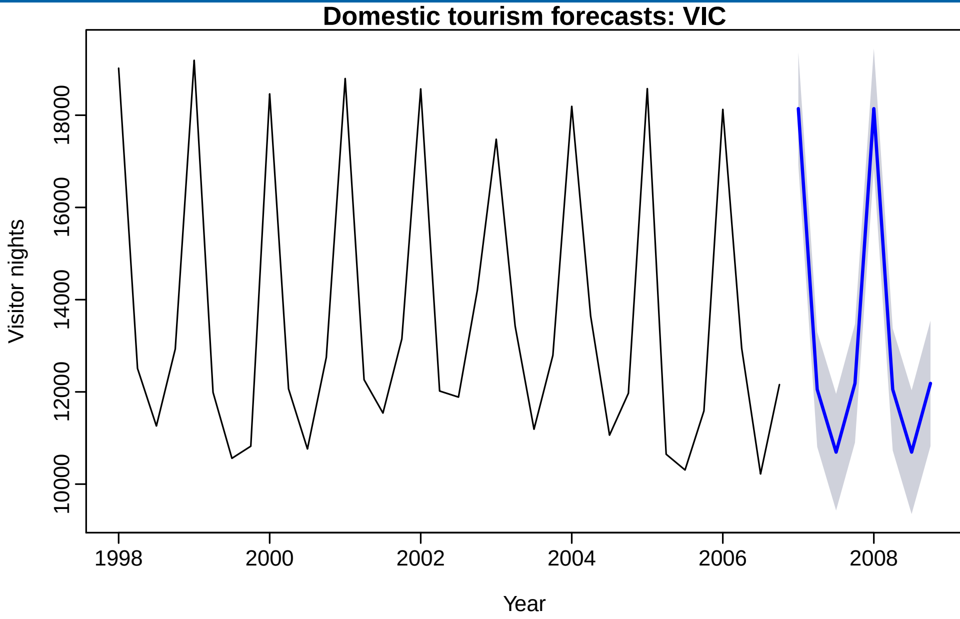
Base forecasts



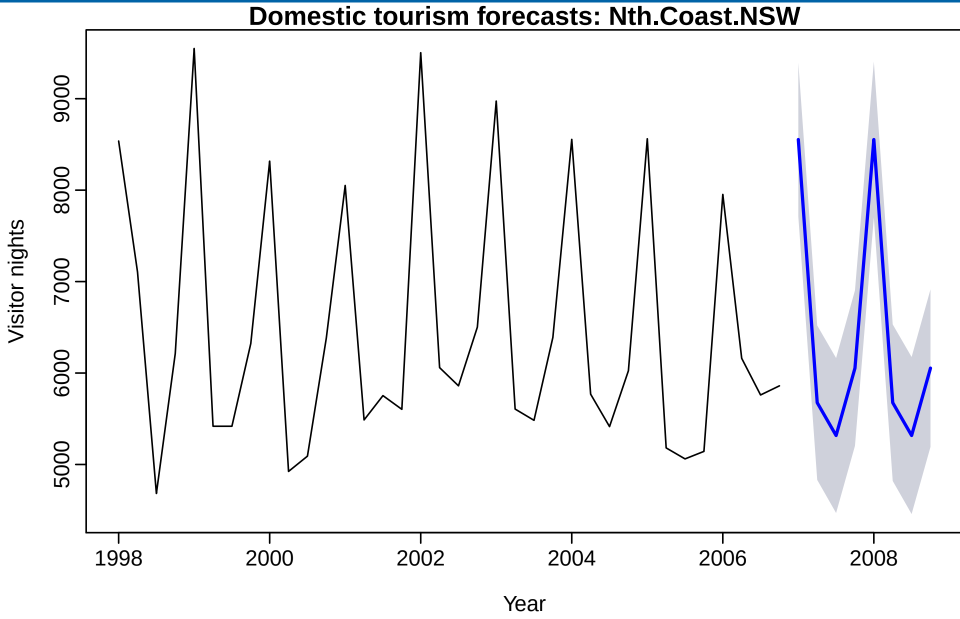
Base forecasts



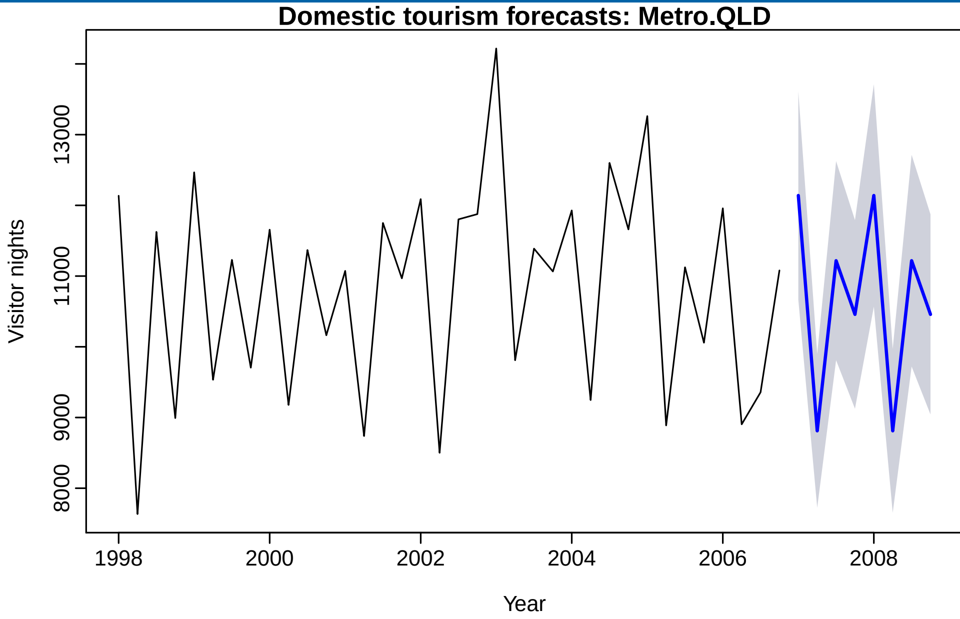
Base forecasts



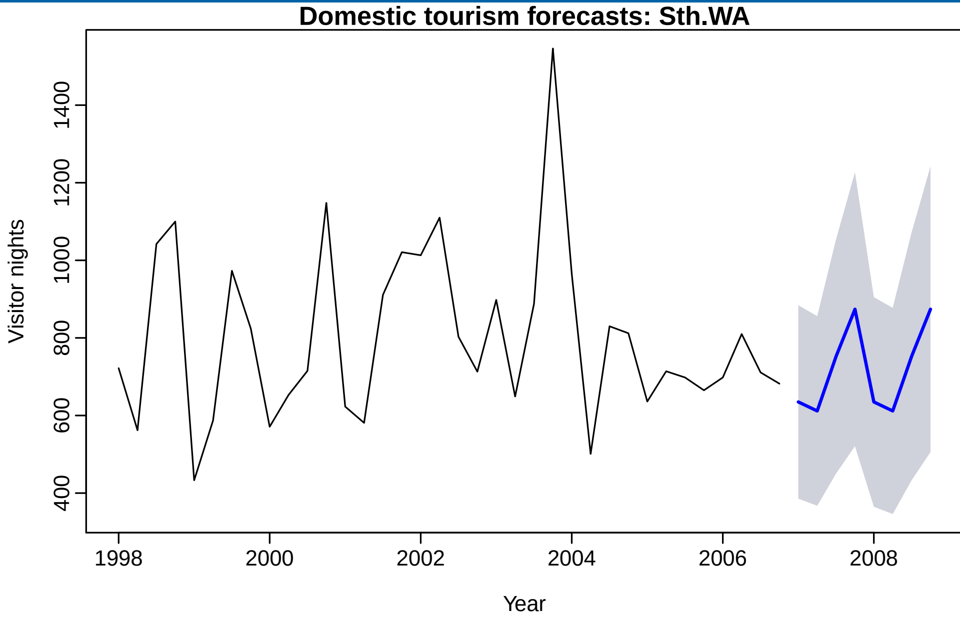
Base forecasts



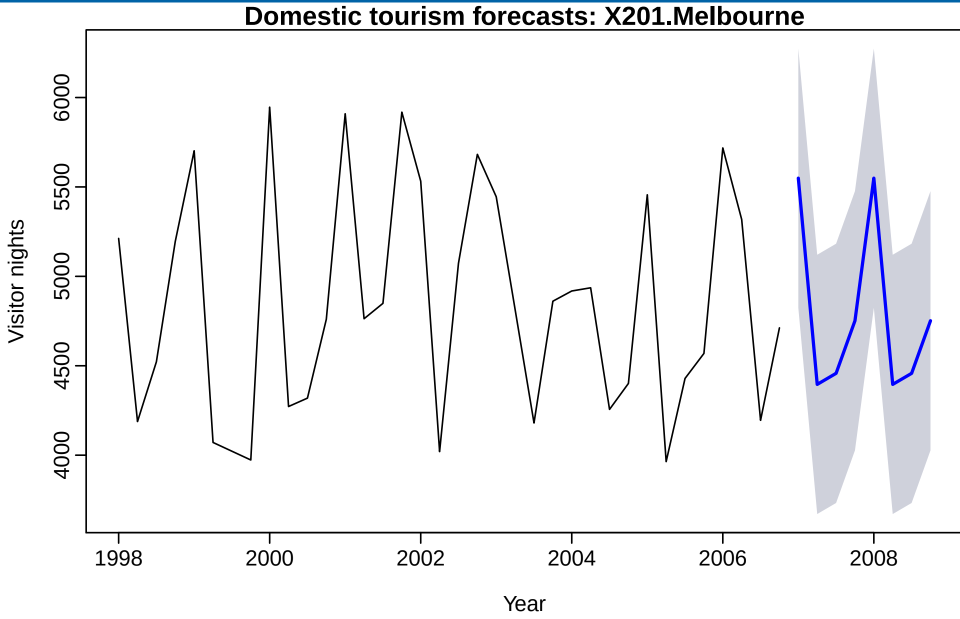
Base forecasts



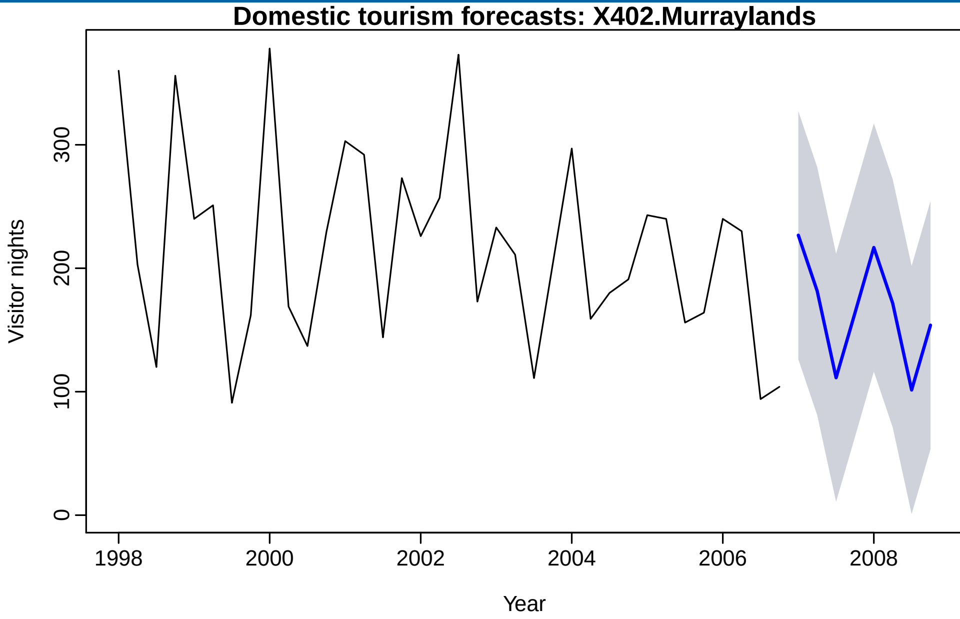
Base forecasts



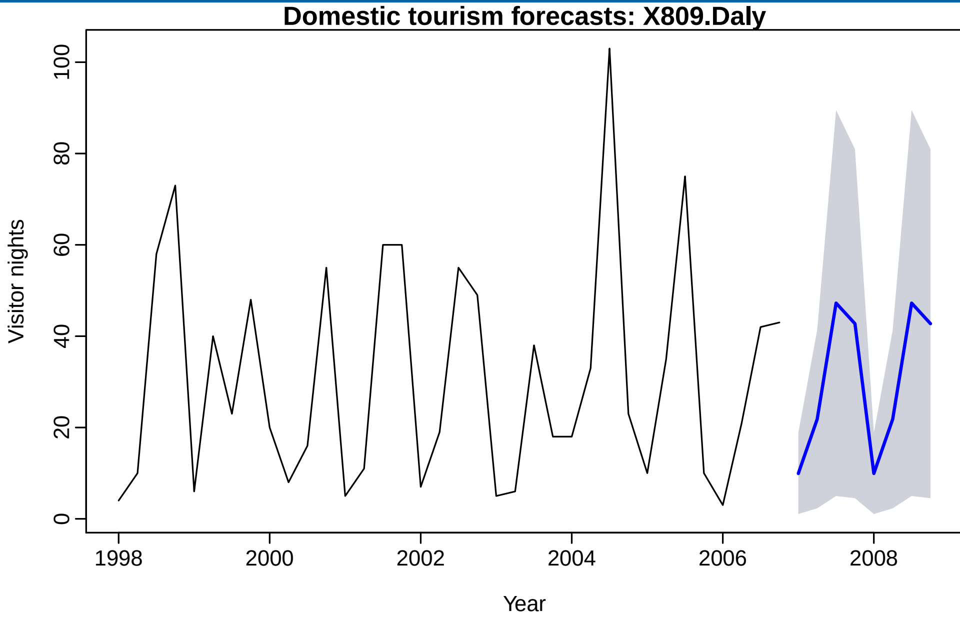
Base forecasts



Base forecasts



Base forecasts

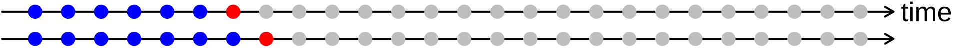


Forecast evaluation

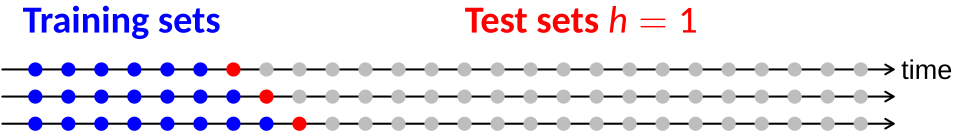
Training sets **Test sets $h = 1$**

Forecast evaluation

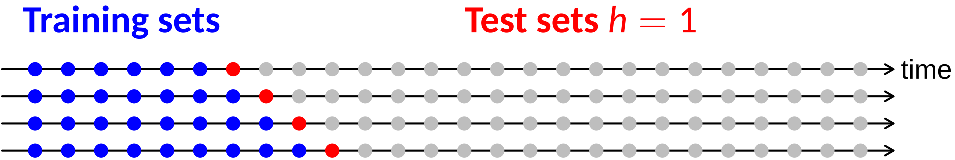
Training sets **Test sets $h = 1$**



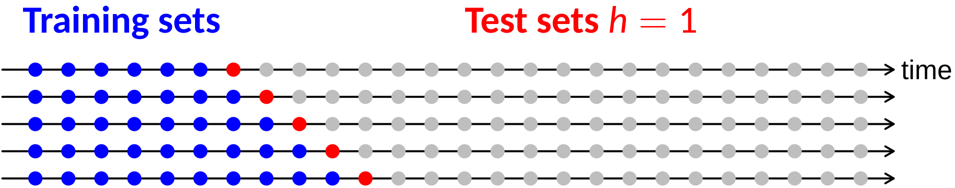
Forecast evaluation



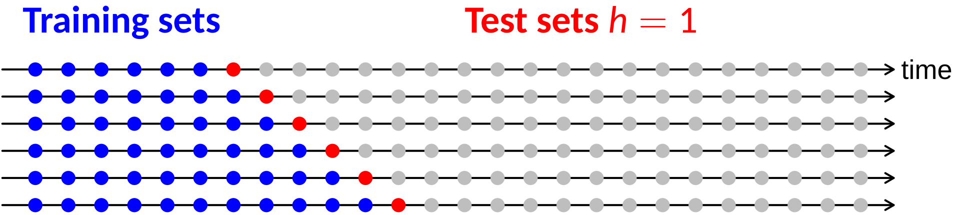
Forecast evaluation



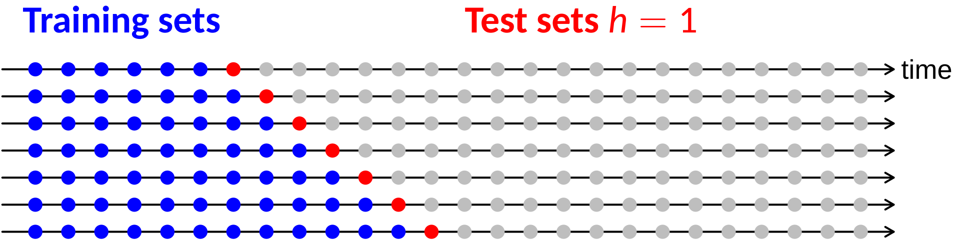
Forecast evaluation



Forecast evaluation

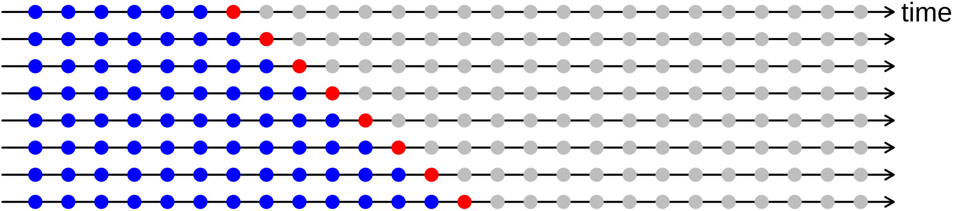


Forecast evaluation



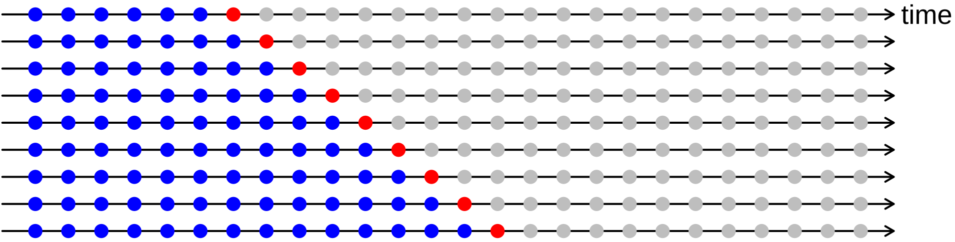
Forecast evaluation

Training sets **Test sets $h = 1$**

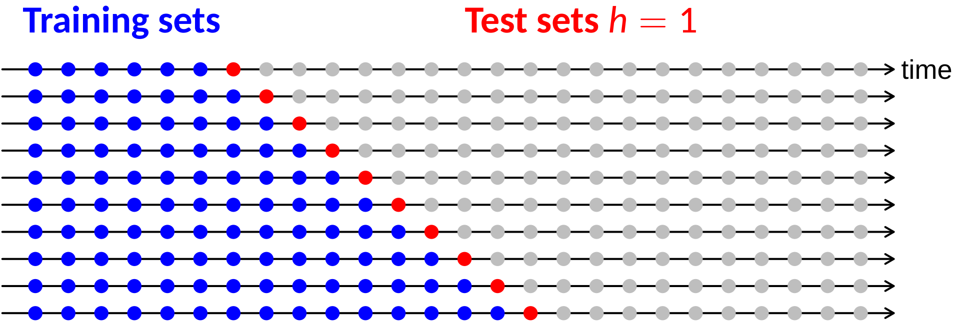


Forecast evaluation

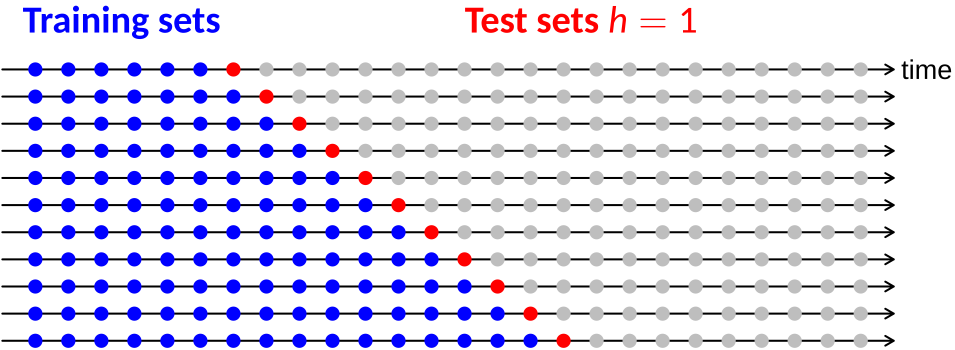
Training sets **Test sets $h = 1$**



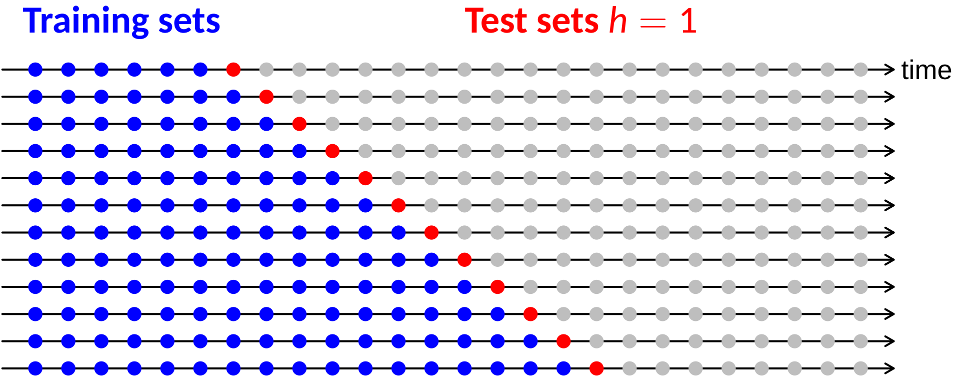
Forecast evaluation



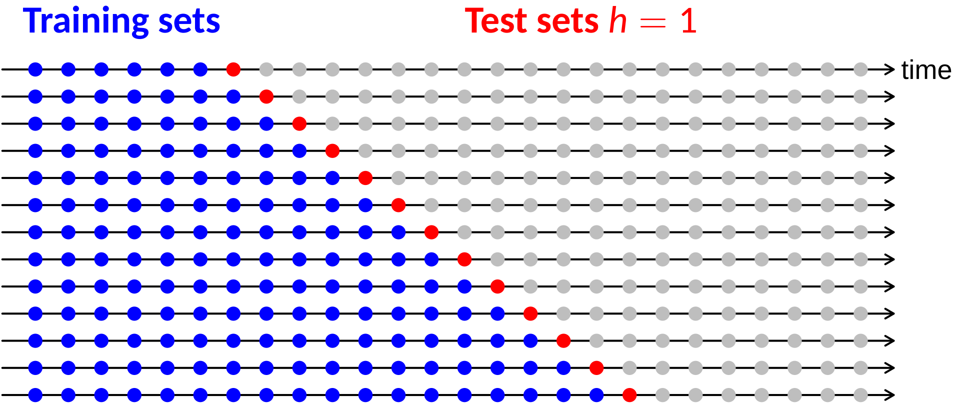
Forecast evaluation



Forecast evaluation

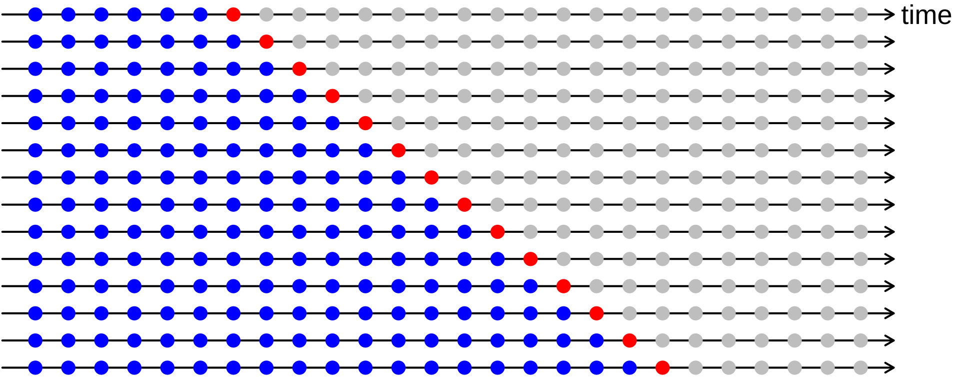


Forecast evaluation

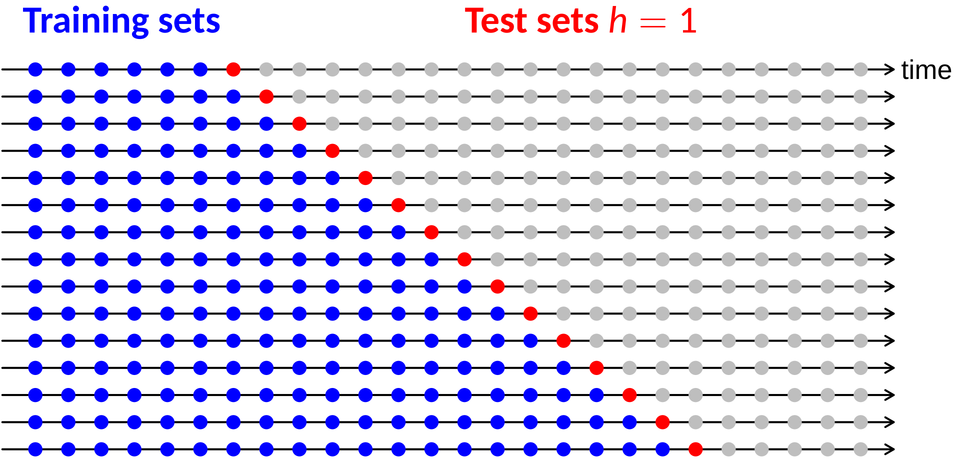


Forecast evaluation

Training sets **Test sets $h = 1$**

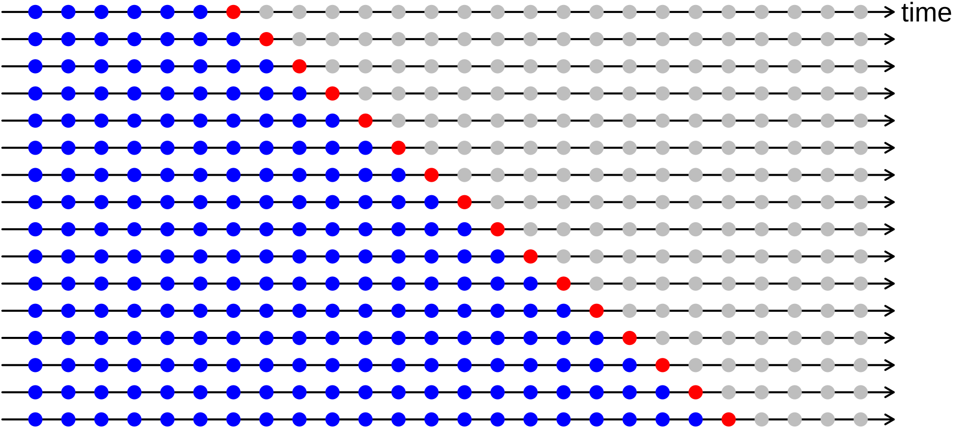


Forecast evaluation



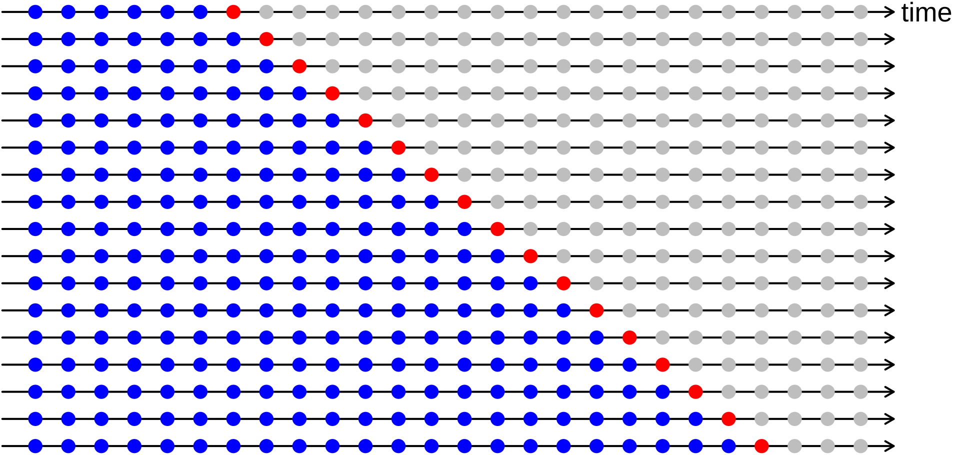
Forecast evaluation

Training sets **Test sets $h = 1$**



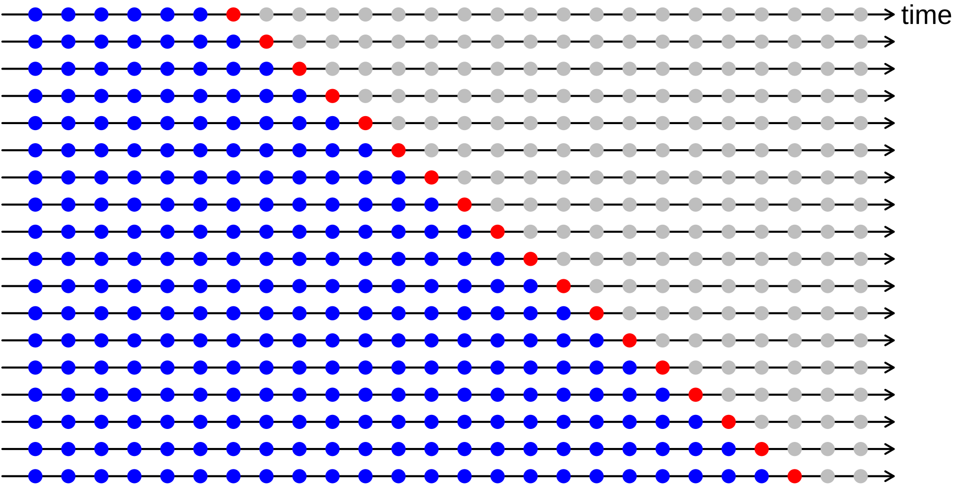
Forecast evaluation

Training sets **Test sets $h = 1$**

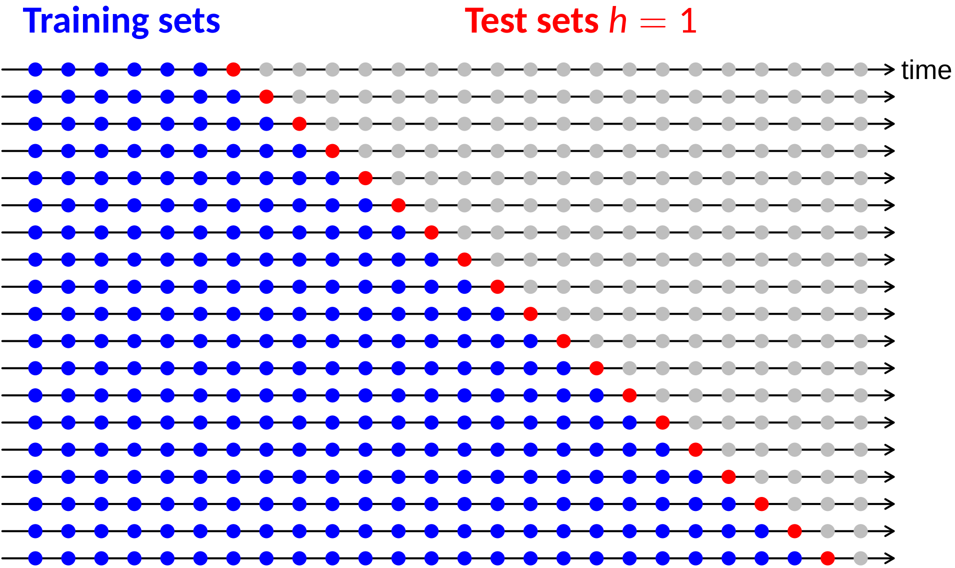


Forecast evaluation

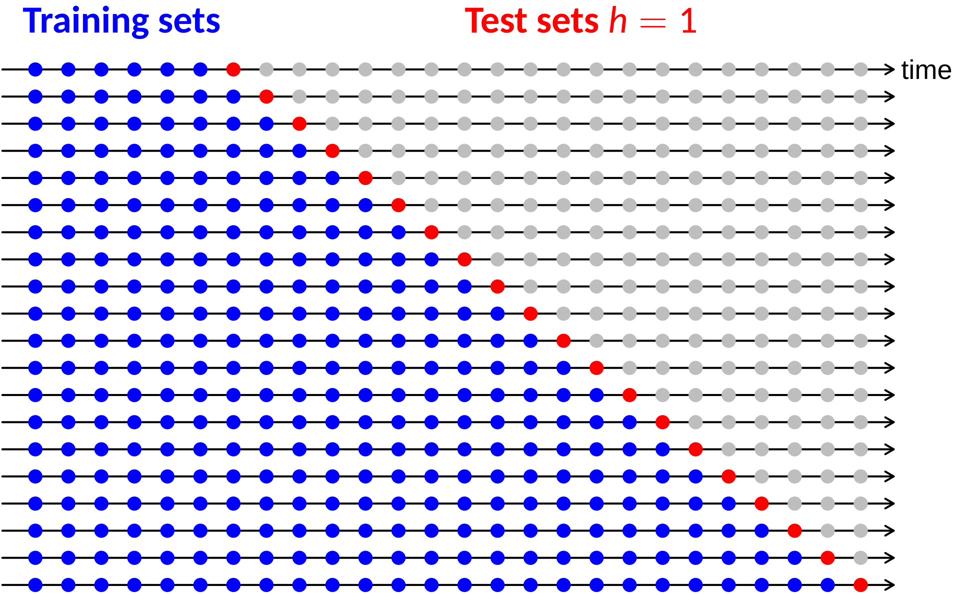
Training sets **Test sets $h = 1$**



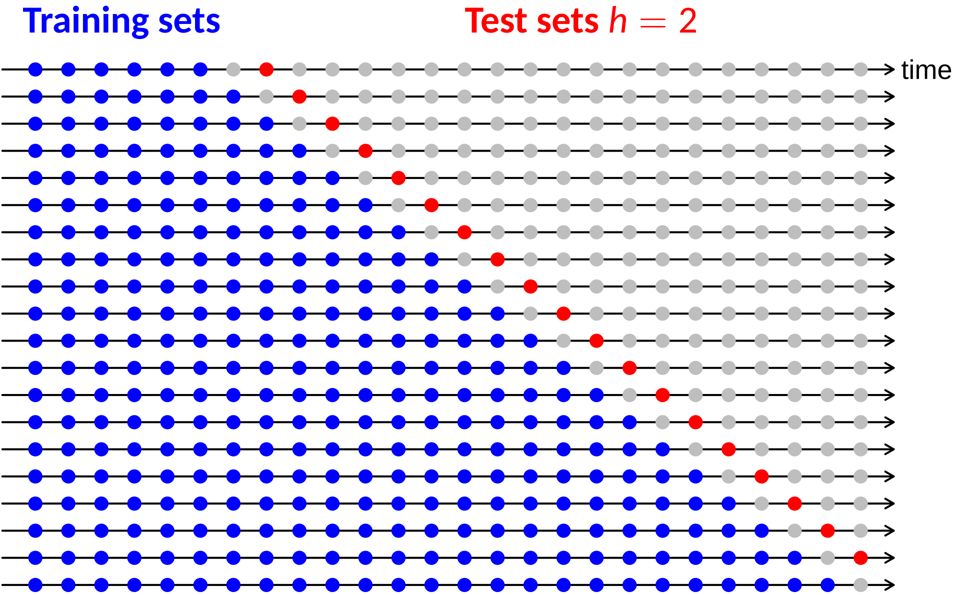
Forecast evaluation



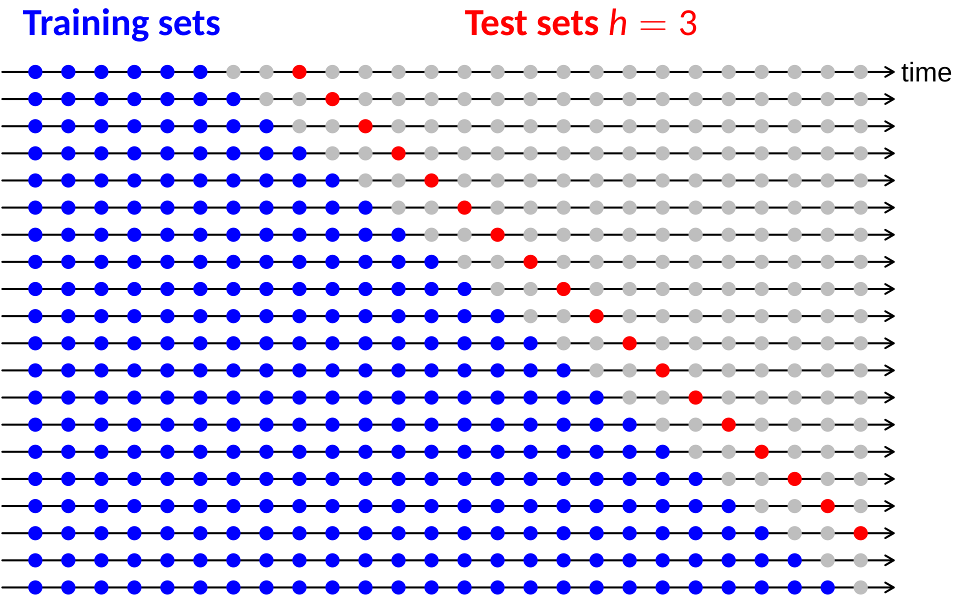
Forecast evaluation



Forecast evaluation

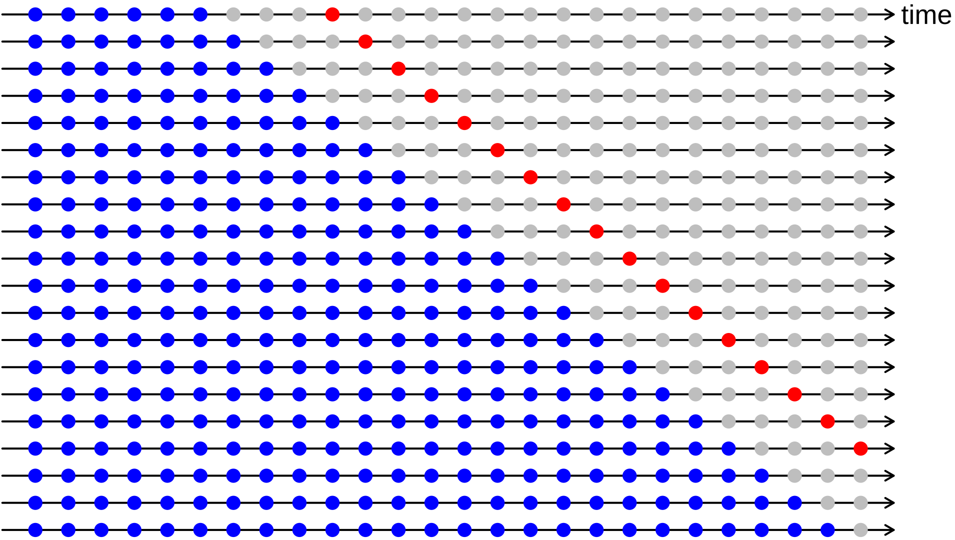


Forecast evaluation



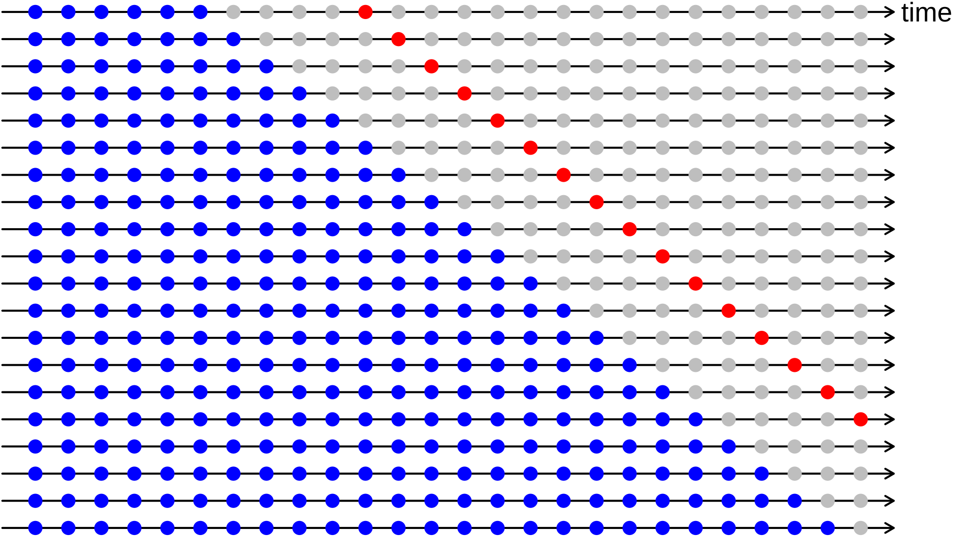
Forecast evaluation

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

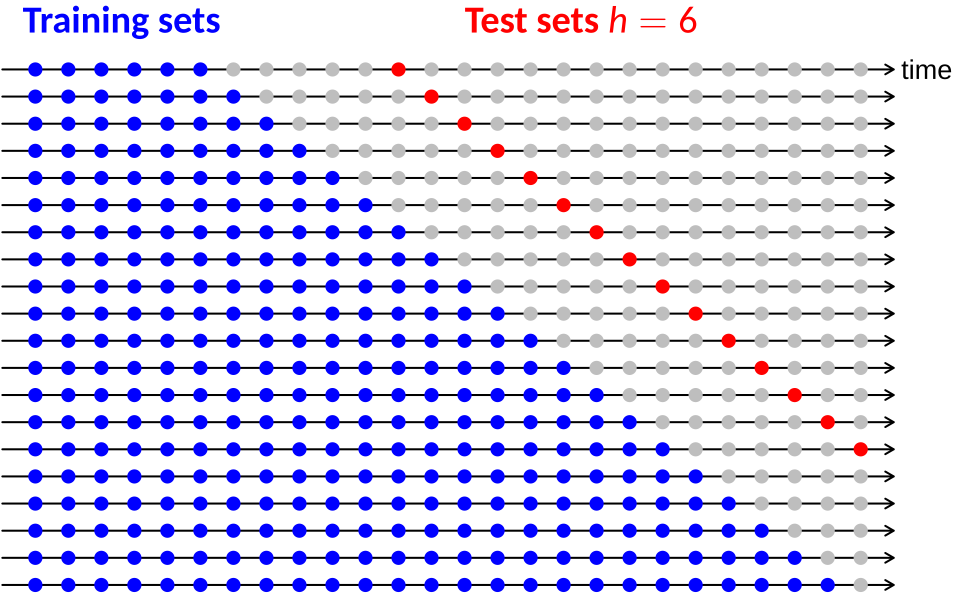


Forecast evaluation

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



Forecast evaluation



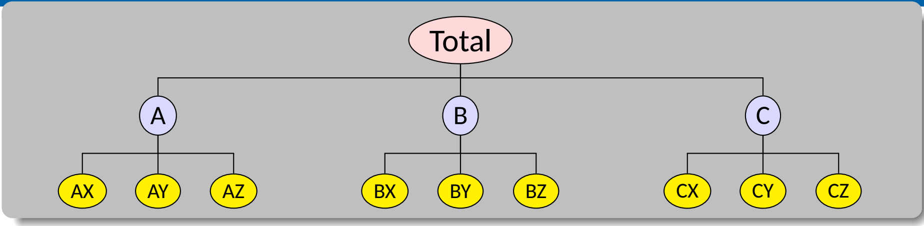
Hierarchy: states, zones, regions

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**

Forecast horizon

Outline

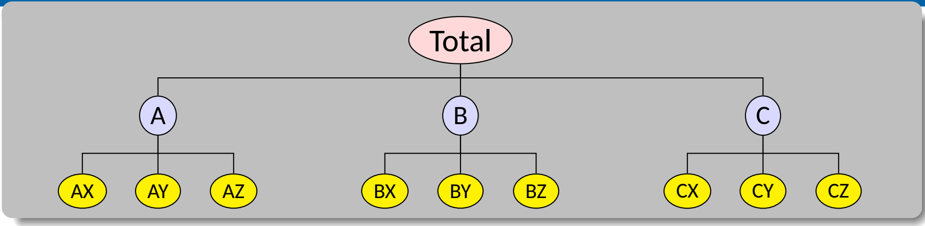
Fast computation: hierarchical data



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

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

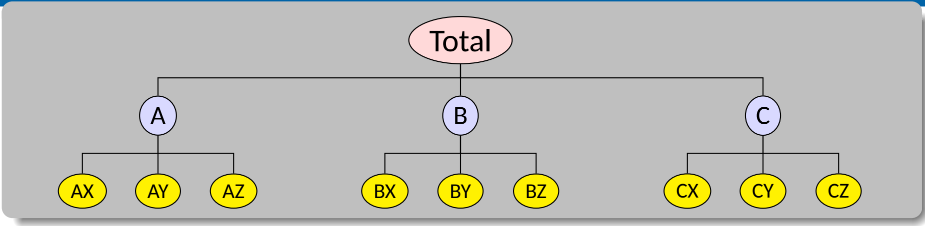
Fast computation: hierarchical data



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

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

Fast computation: hierarchical data

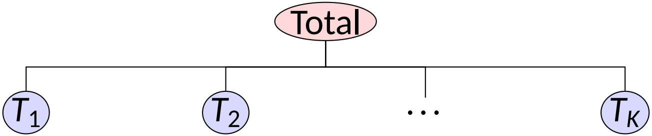


$$
\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{B,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{C,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\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

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

Fast computation: hierarchies

Think of the hierarchy as a tree of trees:



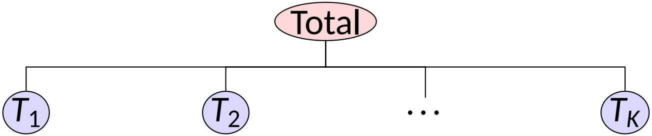
Then the summing matrix contains k smaller summing matrices:

$$S = \begin{bmatrix} \mathbf{1}'_{n_1} & \mathbf{1}'_{n_2} & \cdots & \mathbf{1}'_{n_K} \\ S_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & S_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & S_K \end{bmatrix}$$

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

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

- 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

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

- 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

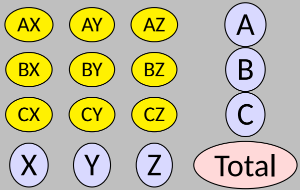
$$(S'\Lambda S)^{-1} = \begin{bmatrix} (S_1'\Lambda_1 S_1)^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (S_2'\Lambda_2 S_2)^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (S_K'\Lambda_K S_K)^{-1} \end{bmatrix} - cS_0$$

- S_0 can be partitioned into K^2 blocks, with the (k,ℓ) 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.

- J_{n_k,n_ℓ}
- $c^{-1} = \lambda_0 + \sum_k 1_{n_k}(S_k'\Lambda_k S_k)^{-1} 1_{n_k}.$
- Each $S_k'\Lambda_k S_k$ can be inverted similarly.
- $S'\Lambda y$ can also be computed recursively.

Fast computation: grouped data



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

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

Fast computation: grouped data

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

$m =$ number of rows
$n =$ number of columns

$$S'\Lambda S = \lambda_{00} J_{mn} + (\Lambda_R \otimes J_n) + (J_m \otimes \Lambda_C) + \Lambda_U$$

- 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

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

$m =$ 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

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

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

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

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

$\boldsymbol{D}$ 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

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

$\boldsymbol{M}$ is $m \times$

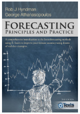
$$(\boldsymbol{M})_{ij} = \begin{cases} 1 + \lambda_{i0}\sum_k \lambda_{ik}^{-1} - \lambda_{i0}\sum_k \lambda_{ik}^{-2} d_k, & i = j, \\ -\lambda_{i0}^{1/2}\lambda_{j0}^{1/2}\sum_k \lambda_{ik}^{-1}\lambda_{jk}^{-1} d_k, & i \neq j. \end{cases}$$

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, A Lee and E Wang (2016). Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational Statistics & Data Analysis* **97**, 16–32

SL Wickramasuriya, G Athanasopoulos and RJ Hyndman (2015). *Forecasting hierarchical and grouped time series through trace minimization*. Working paper. Dept Econometrics & Business Statistics, Monash University



RJ Hyndman and G Athanasopoulos (2018). *Forecasting: principles and practice*. 2nd ed. Melbourne, Australia: OTexts. `OTexts.org/fpp2/`.

R packages

https://github.com/earowang/tsibble

http://pkg.earo.me/sugrrants

https://github.com/mitchelloharawild/fasster

http://pkg.robjhyndman.com/forecast

http://pkg.earo.me/hts