Improving forecasts via subspace projections

Rob J Hyndman

MONASH University

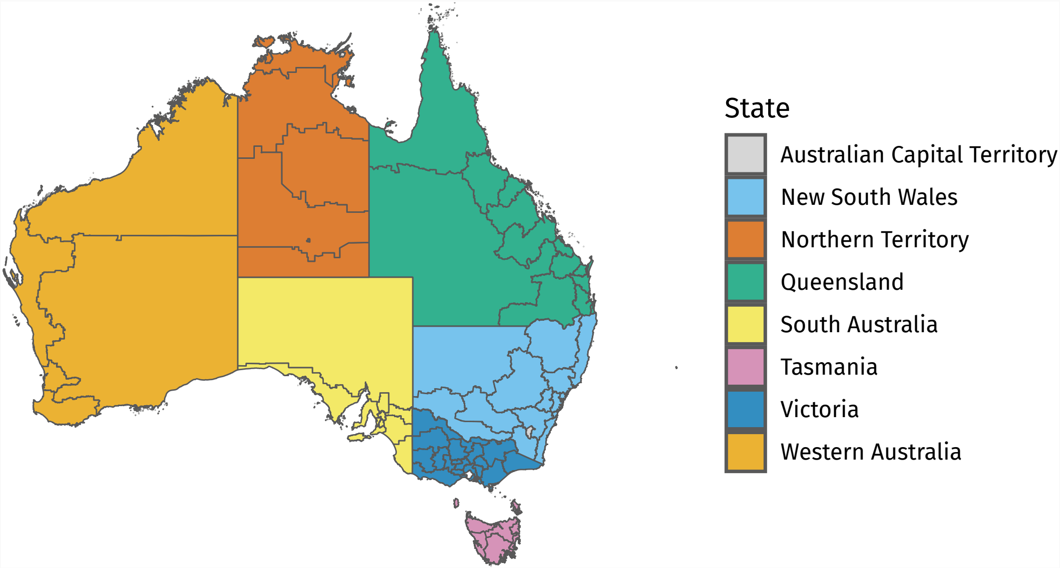
OPTiMA

Photo by Stanley Cheung on Unsplash

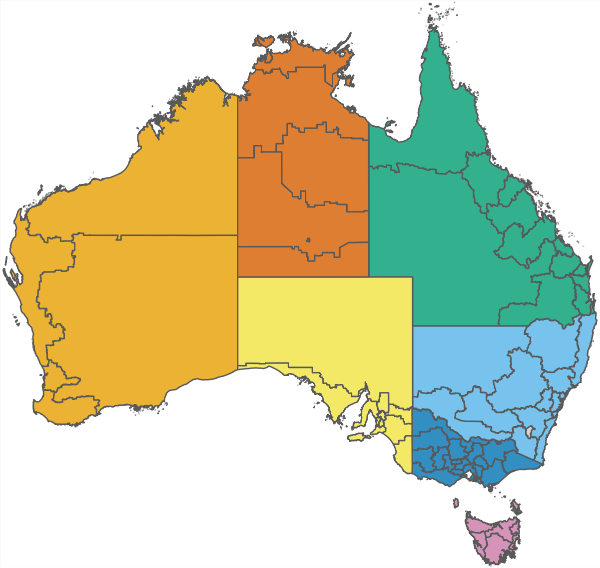
Outline

Outline

Australian tourism regions

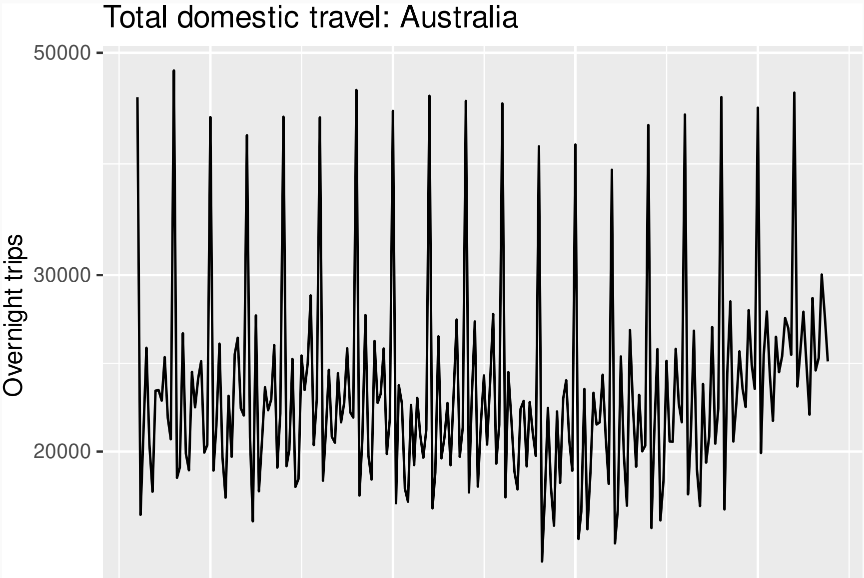


Australian tourism regions

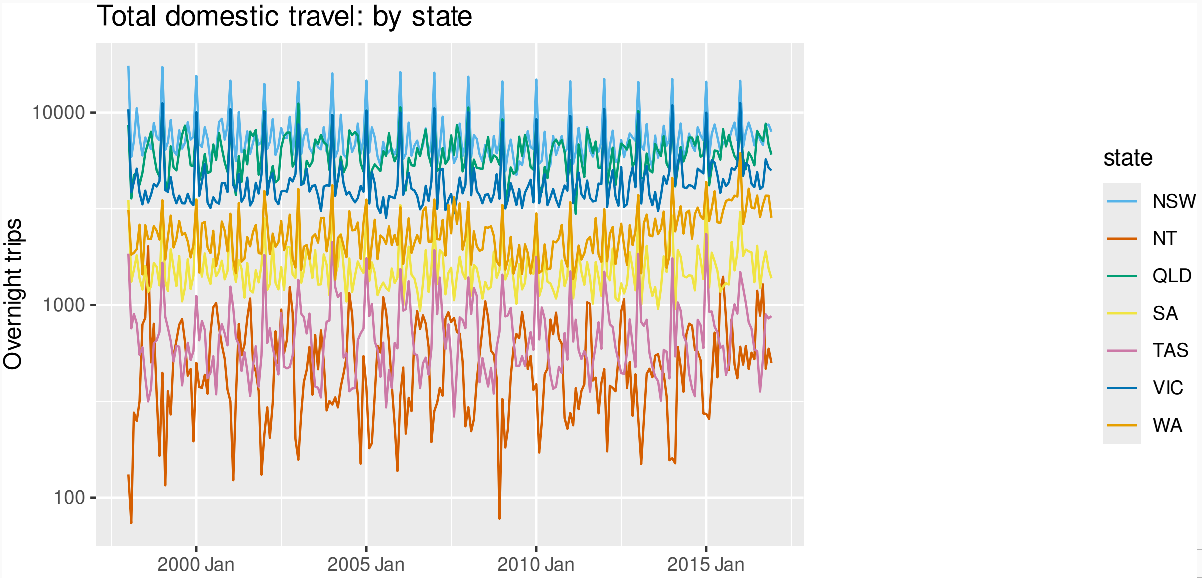


- Monthly data on visitor night from 1998 – 2017
- From *National Visitor Survey*, annual interviews of 120,000 Australians aged 15+.
- Geographical hierarchy split by
  - 7 states
  - 27 zones
  - 75 regions

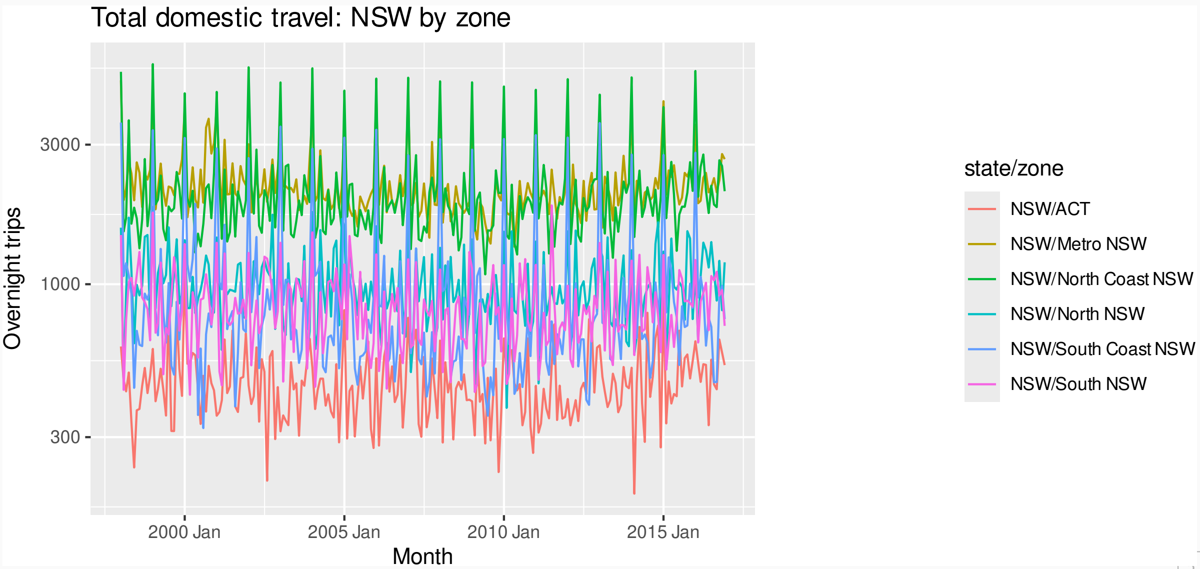
Australian tourism data



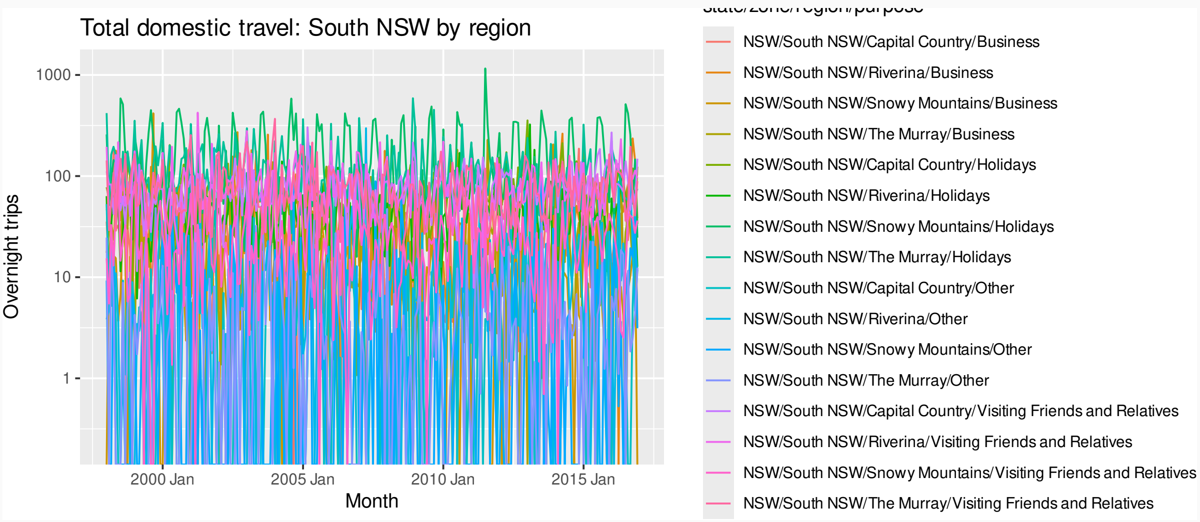
Australian tourism data



Australian tourism data



Australian tourism data

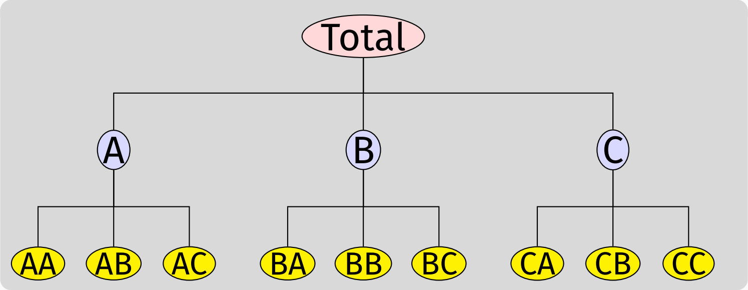


Spectacle sales

- 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

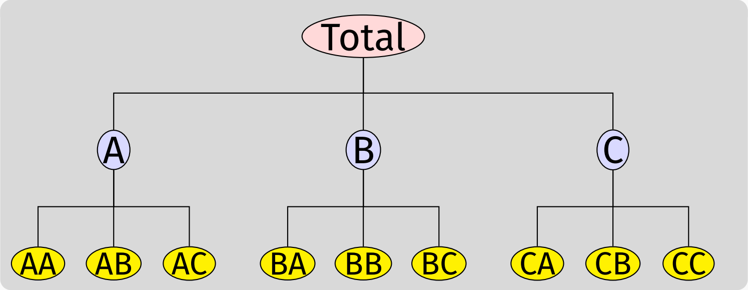
Hierarchical time series

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



Hierarchical time series

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

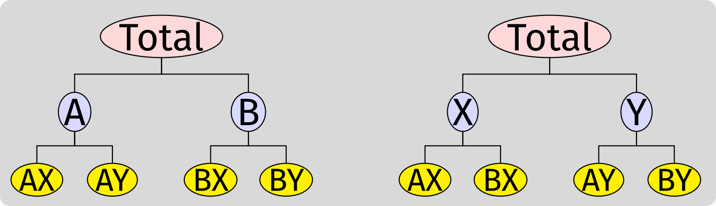


Examples

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

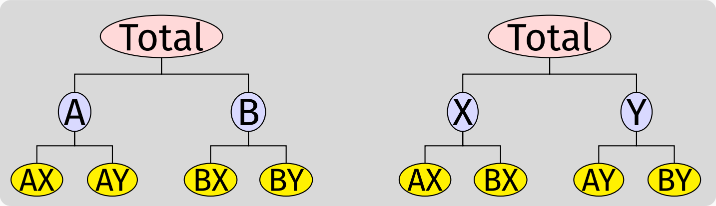
Grouped time series

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



Grouped time series

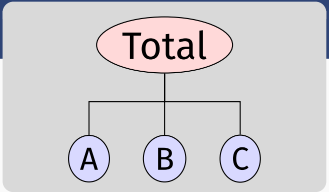
A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.



Examples

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

Hierarchical and grouped time series



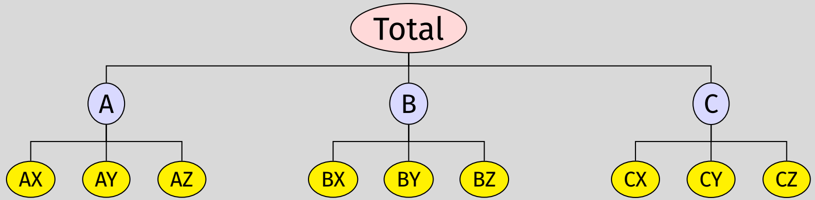
Almost all collections of time series with linear constraints can be written as

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

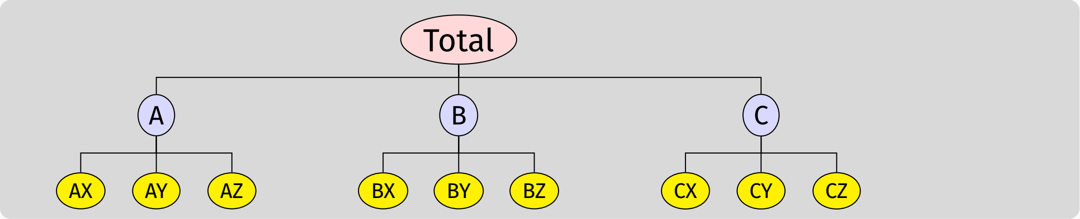
- $\boldsymbol{y}_t$ = vector of all series at time t
- $y_{\text{Total},t}$ = aggregate of all series at time t.
- $y_{X,t}$ = value of series X at time t.
- $\boldsymbol{b}_t$ = vector of most disaggregated series at time t
- $\boldsymbol{S}$ = "summing matrix" containing the linear constraints.

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

Hierarchical time series

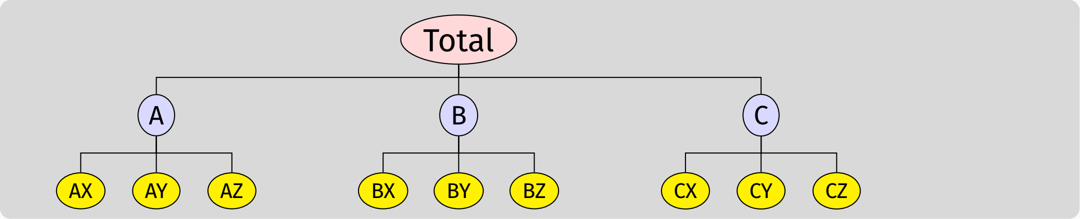


Hierarchical time series



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

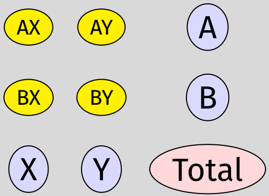
Hierarchical time series



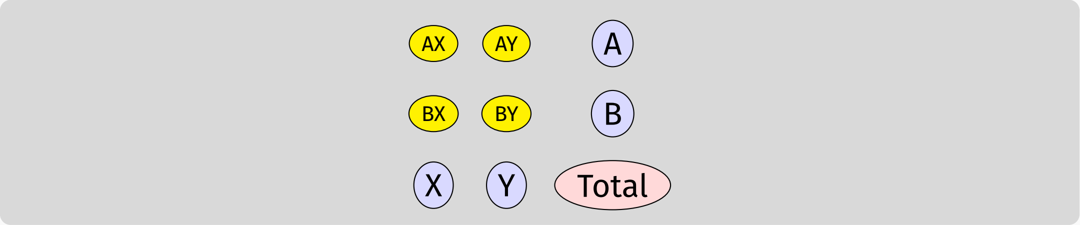
$$\boldsymbol{y}_t = \begin{pmatrix} y_{\text{Total},t} \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\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

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

Grouped data

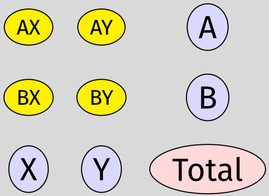


Grouped data



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

Grouped data



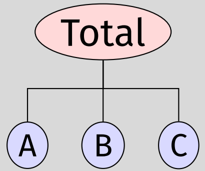
$$\boldsymbol{y}_t = \begin{pmatrix} y_{\text{Total},t} \\ y_{A,t} \\ y_{B,t} \\ y_{X,t} \\ y_{Y,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0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

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

The hierarchical forecasting problem

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

Constraint notation



Aggregation matrix

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

$$\begin{pmatrix} y_{\text{Total},t} \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}$$

$$\begin{pmatrix} \boldsymbol{a}_t \\ \boldsymbol{b}_t \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{I}_{n_b} \end{pmatrix} \boldsymbol{b}_t$$

Constraint matrix

$$\boldsymbol{C}\boldsymbol{y}_t = \boldsymbol{0}$$

where

$$\begin{aligned} \boldsymbol{C} &= \begin{bmatrix} 1 & -1 & -1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I}_{n_a} & -\boldsymbol{A} \end{bmatrix} \end{aligned}$$

Constraint notation

Aggregation matrix A

$$\boldsymbol{y}_t = \begin{bmatrix} \boldsymbol{a}_t \\ \boldsymbol{b}_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{I}_{n_b} \end{bmatrix} \boldsymbol{b}_t = \boldsymbol{S}\boldsymbol{b}_t$$

Constraint notation

Aggregation matrix $\boldsymbol{A}$

$$\boldsymbol{y}_t = \begin{bmatrix} \boldsymbol{a}_t \\ \boldsymbol{b}_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{I}_{n_b} \end{bmatrix} \boldsymbol{b}_t = \boldsymbol{S}\boldsymbol{b}_t$$

Constraint matrix $\boldsymbol{C}$

$$\boldsymbol{C}\boldsymbol{y}_t = \boldsymbol{0}$$

- Constraint matrix approach more general & more parsimonious.
- $\boldsymbol{C} = [\boldsymbol{I}_{n_a} \quad -\boldsymbol{A}]$.
- $\boldsymbol{S}$, $\boldsymbol{A}$ and $\boldsymbol{C}$ may contain any real values (not just 0s and 1s).

The coherent subspace

Coherent subspace

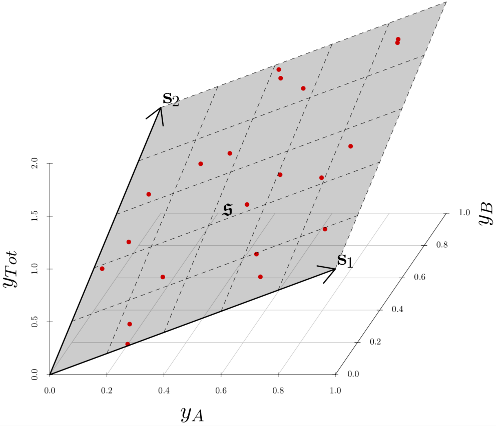
n_b-dimensional linear subspace $\mathfrak{s} \subset \chi^n$ for which linear constraints hold for all $\boldsymbol{y} \in \mathfrak{s}$.

Hierarchical time series

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

Coherent point forecasts

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



$y_{Tot} = y_A + y_B$

The coherent subspace

Coherent subspace

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

Hierarchical time series

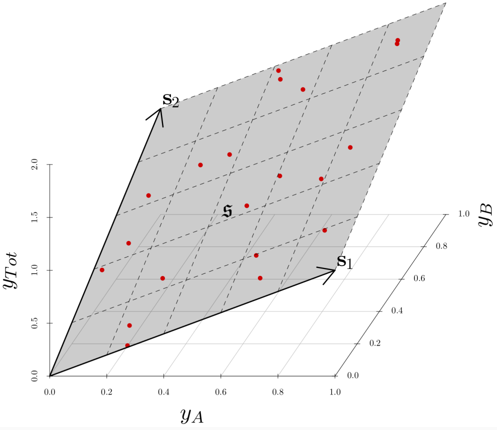
An n-dimensional multivariate time series such that $\boldsymbol{y}_t \in \mathfrak{s} \quad \forall t$.

Coherent point forecasts

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

Base forecasts

Let $\hat{\boldsymbol{y}}_{t+h|t}$ be vector of *incoherent* initial h-step forecasts.



$y_{Tot} = y_A + y_B$

The coherent subspace

Coherent subspace

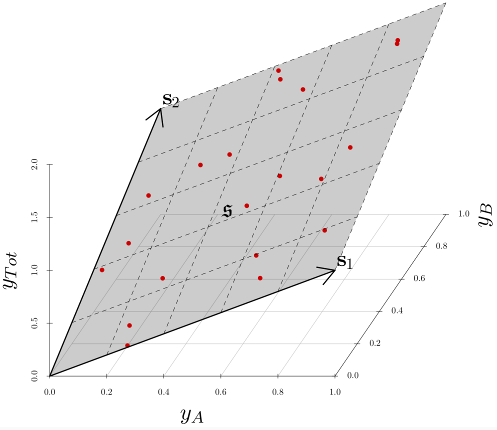
n_b-dimensional linear subspace $\mathfrak{s} \subset \chi^n$ for which linear constraints hold for all $\boldsymbol{y} \in \mathfrak{s}$.

Hierarchical time series

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

Coherent point forecasts

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



$y_{Tot} = y_A + y_B$

Base forecasts

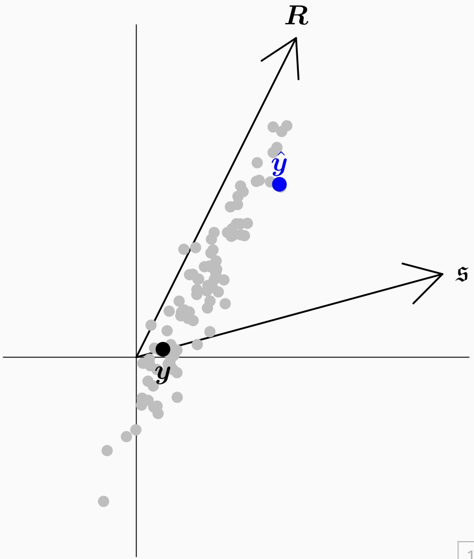
Let $\hat{\boldsymbol{y}}_{t+h|t}$ be vector of *incoherent* initial h-step forecasts.

Reconciled forecasts

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

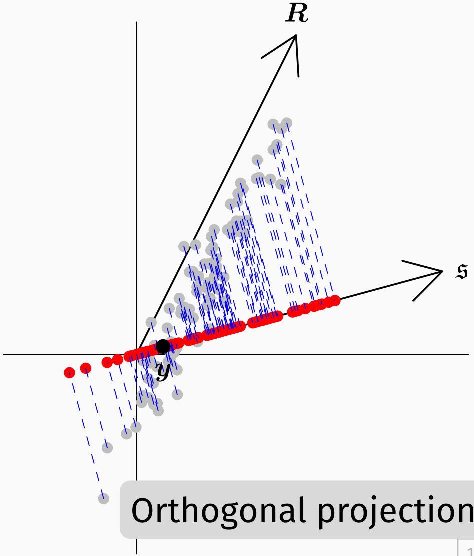
Linear projection reconciliation

- R is the most likely direction of deviations from $\mathfrak{s}$.
- Grey: potential base forecasts



Linear projection reconciliation

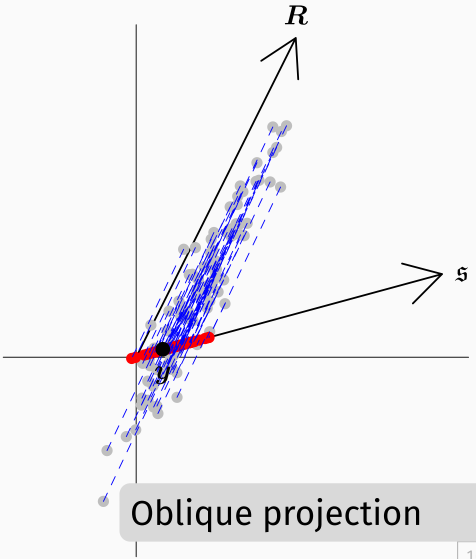
- R is the most likely direction of deviations from $\mathfrak{s}$.
- Grey: potential base forecasts
- Red: reconciled forecasts
- Orthogonal projections (i.e., OLS) lead to smallest possible adjustments of base forecasts.



Orthogonal projection

Linear projection reconciliation

- R is the most likely direction of deviations from $\mathfrak{s}$.
- Grey: potential base forecasts
- Red: reconciled forecasts
- Orthogonal projections (i.e., OLS) lead to smallest possible adjustments of base forecasts.
- Oblique projections (i.e., MinT) give reconciled forecasts with smallest variance.



Oblique projection

Linear projection reconciliation

$$\tilde{\boldsymbol{y}}_{t+h|t} = \psi(\hat{\boldsymbol{y}}_{t+h|t}) = \boldsymbol{M}\hat{\boldsymbol{y}}_{t+h|t}$$

- $\boldsymbol{M}$ is a projection onto $\mathfrak{s}$ if and only if $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{y}$ for all $\boldsymbol{y} \in \mathfrak{s}$.
- Coherent base forecasts are unchanged since $\boldsymbol{M}\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}$
- If $\hat{\boldsymbol{y}}$ is unbiased, then $\tilde{\boldsymbol{y}}$ is also unbiased since

  $$\mathrm{E}(\tilde{\boldsymbol{y}}_{t+h|t}) = \mathrm{E}(\boldsymbol{M}\hat{\boldsymbol{y}}_{t+h|t}) = \boldsymbol{M}\mathrm{E}(\hat{\boldsymbol{y}}_{t+h|t}) = \mathrm{E}(\hat{\boldsymbol{y}}_{t+h|t}),$$

  and unbiased estimates must lie on $\mathfrak{s}$.
- If $\boldsymbol{S}$ forms a basis set for $\mathfrak{s}$, then projections are of the form $\boldsymbol{M} = \boldsymbol{S}(\boldsymbol{S}'\Psi\boldsymbol{S})^{-1}\boldsymbol{S}'\Psi$ where Ψ is a positive definite matrix.
- The projection is orthogonal if and only if $\boldsymbol{M}' = \boldsymbol{M}$. Equivalently, $\Psi = \boldsymbol{I}$.

Linear projection reconciliation

$$\tilde{\boldsymbol{y}}_{t+h|t} = \psi(\hat{\boldsymbol{y}}_{t+h|t}) = \boldsymbol{M}\hat{\boldsymbol{y}}_{t+h|t}$$

$$\text{where} \quad \boldsymbol{M} = \boldsymbol{S}(\boldsymbol{S}'\Psi\boldsymbol{S})^{-1}\boldsymbol{S}'\Psi = \boldsymbol{I} - \Psi\boldsymbol{C}'(\boldsymbol{C}\Psi\boldsymbol{C}')^{-1}\boldsymbol{C}$$

OLS:	$\Psi = \boldsymbol{I}$
WLS:	$\Psi = \text{diagonal}$
MinT:	$\Psi = \boldsymbol{W}_h$

- $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]$ is the covariance matrix of the base forecast errors.
- $\boldsymbol{V}_h = \text{Var}[\boldsymbol{y}_{T+h} - \tilde{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]) = \boldsymbol{M}\boldsymbol{W}_h\boldsymbol{M}'$ is the covariance matrix of the reconciled forecast errors.

Minimum trace reconciliation

Minimum trace (MinT) reconciliation

If $\boldsymbol{M}$ is a projection, then trace of $\boldsymbol{V}_h$ is minimized when $\Psi = \boldsymbol{W}_h$, so that

$$\boldsymbol{M} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1} = \boldsymbol{I} - \boldsymbol{W}_h\boldsymbol{C}'(\boldsymbol{C}\boldsymbol{W}_h\boldsymbol{C}')^{-1}\boldsymbol{C}$$

$$\tilde{\boldsymbol{y}}_{T+h|T} = \boldsymbol{M}\,\hat{\boldsymbol{y}}_{T+h|T}$$

Reconciled forecasts   Base forecasts

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

Linear projections

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

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

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

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

Example: Australian tourism

```r
tourism
```

```
# A tsibble: 69,312 x 6 [1M]
# Key:       state, zone, region, purpose [304]
      month state zone  region   purpose  visitors
      <mth> <chr> <chr> <chr>    <chr>       <dbl>
 1 1998 Jan NSW   ACT   Canberra Business     25.0
 2 1998 Feb NSW   ACT   Canberra Business    148.
 3 1998 Mar NSW   ACT   Canberra Business    111.
 4 1998 Apr NSW   ACT   Canberra Business     93.1
 5 1998 May NSW   ACT   Canberra Business     78.1
 6 1998 Jun NSW   ACT   Canberra Business     44.3
 7 1998 Jul NSW   ACT   Canberra Business    129.
 8 1998 Aug NSW   ACT   Canberra Business     71.3
 9 1998 Sep NSW   ACT   Canberra Business     77.7
10 1998 Oct NSW   ACT   Canberra Business    145.
# i 69,302 more rows
```

Example: Australian tourism

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

```
# A tsibble: 25,308 x 5 [1M]
# Key:       state, zone, region [111]
      month state        zone         region       visitors
      <mth> <chr*>       <chr*>       <chr*>          <dbl>
 1 1998 Jan <aggregated> <aggregated> <aggregated>   45151.
 2 1998 Feb <aggregated> <aggregated> <aggregated>   17295.
 3 1998 Mar <aggregated> <aggregated> <aggregated>   20725.
 4 1998 Apr <aggregated> <aggregated> <aggregated>   25389.
 5 1998 May <aggregated> <aggregated> <aggregated>   20330.
 6 1998 Jun <aggregated> <aggregated> <aggregated>   18238.
 7 1998 Jul <aggregated> <aggregated> <aggregated>   23005.
 8 1998 Aug <aggregated> <aggregated> <aggregated>   23033.
 9 1998 Sep <aggregated> <aggregated> <aggregated>   22483.
10 1998 Oct <aggregated> <aggregated> <aggregated>   24845.
# i 25,298 more rows
```

Example: Australian tourism

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

```
# A mable: 111 x 4
# Key:     state, zone, region [111]
   state  zone            region                   ets
   <chr*> <chr*>          <chr*>               <model>
 1 NSW    ACT             Canberra        <ETS(M,N,A)>
 2 NSW    ACT             <aggregated>    <ETS(M,N,A)>
 3 NSW    Metro NSW       Central Coast   <ETS(M,N,A)>
 4 NSW    Metro NSW       Sydney          <ETS(M,N,A)>
 5 NSW    Metro NSW       <aggregated>    <ETS(M,N,A)>
 6 NSW    North Coast NSW Hunter          <ETS(M,N,M)>
 7 NSW    North Coast NSW North Coast NSW <ETS(M,N,M)>
 8 NSW    North Coast NSW <aggregated>    <ETS(M,N,M)>
 9 NSW    North NSW       Blue Mountains  <ETS(M,N,M)>
10 NSW    North NSW       Central NSW     <ETS(A,N,A)>
```

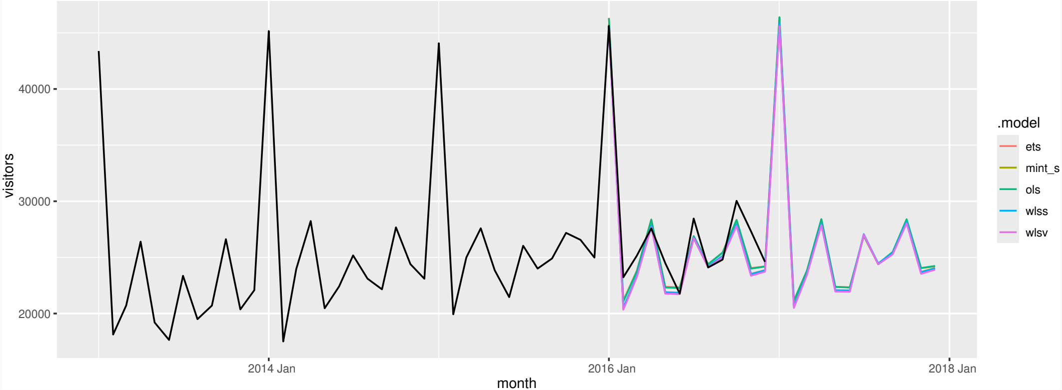
Example: Australian tourism

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

```
# A fable: 13,320 x 7 [1M]
# Key:     state, zone, region, .model [555]
   state  zone   region   .model    month        visitors .mean
   <chr*> <chr*> <chr*>   <chr>     <mth>          <dist> <dbl>
 1 NSW    ACT    Canberra ets    2016 Jan N(593, 21917)   593.
 2 NSW    ACT    Canberra ets    2016 Feb  N(367, 8384)   367.
 3 NSW    ACT    Canberra ets    2016 Mar N(461, 13247)   461.
 4 NSW    ACT    Canberra ets    2016 Apr N(506, 15968)   506.
```

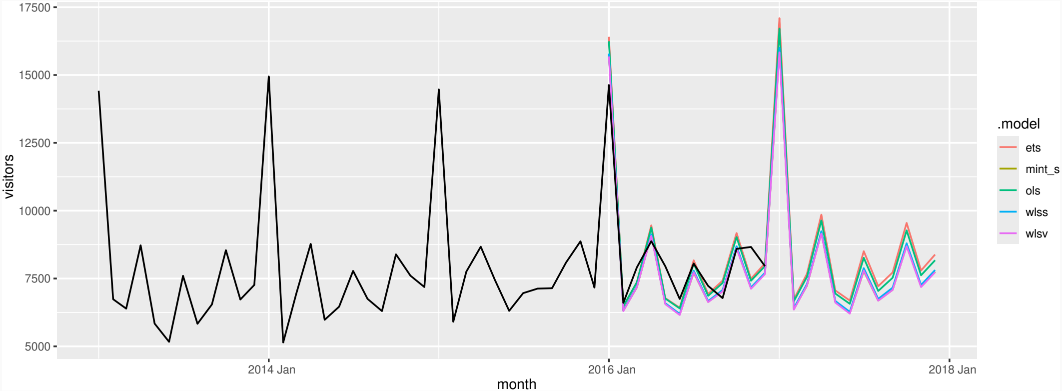
Example: Australian tourism

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



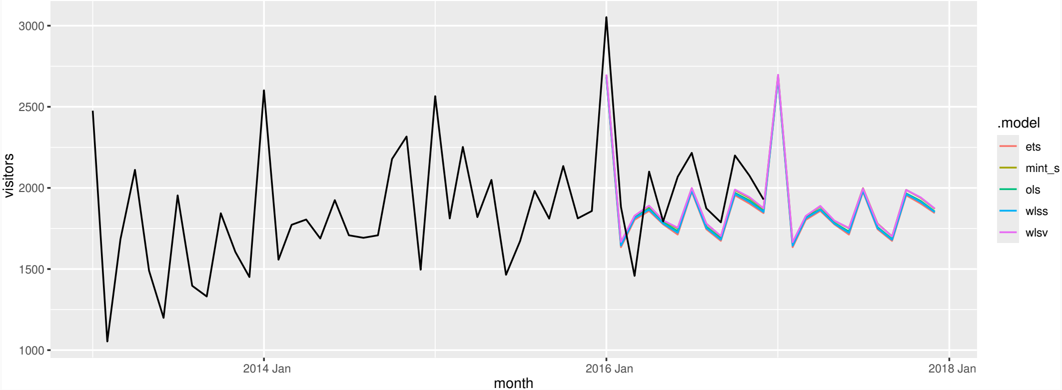
Example: Australian tourism

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



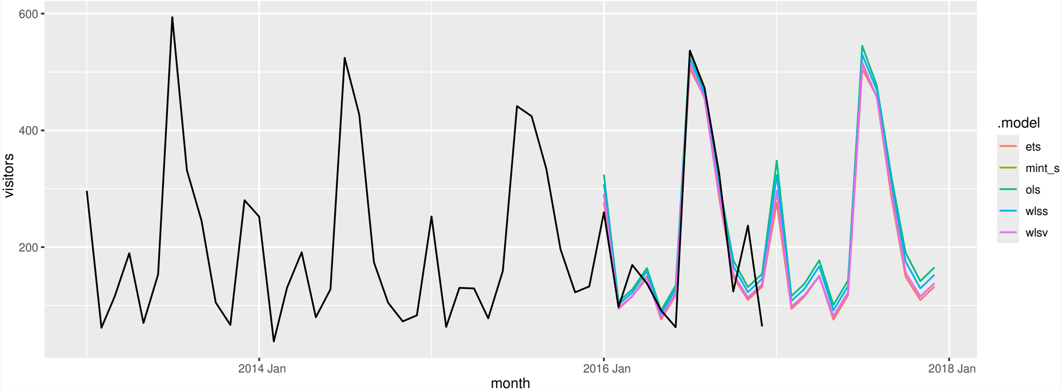
Example: Australian tourism

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



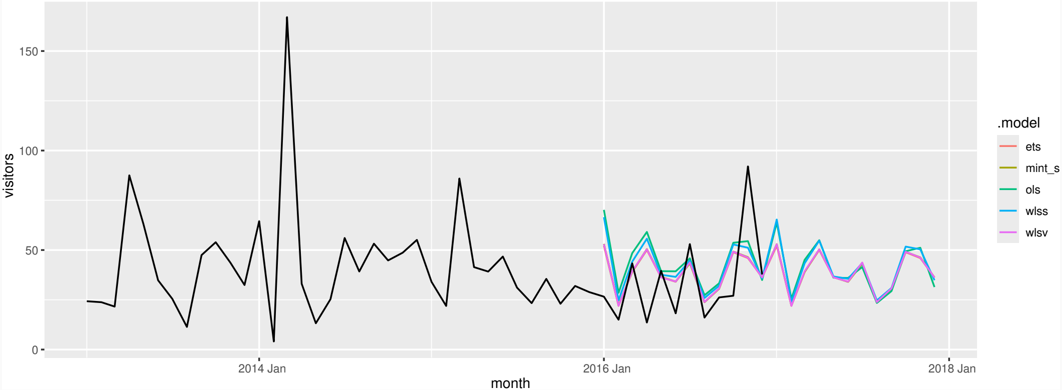
Example: Australian tourism

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



Example: Australian tourism

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



Performance evaluation

$$\text{MASE} = \text{mean}(|q_j|)$$

$$q_j = \frac{e_j}{\frac{1}{T-m}\sum_{t=m+1}^{T}|y_t - y_{t-m}|}$$

- y_t = observation for period t
- e_j = forecast error for forecast horizon j
- T = size of training set
- $m = 12$

Performance evaluation

$$\text{RMSSE} = \sqrt{\text{mean}(q_j^2)}$$

$$q_j^2 = \frac{e_j^2}{\dfrac{1}{T-m}\displaystyle\sum_{t=m+1}^{T}(y_t - y_{t-m})^2}$$

- y_t = observation for period t
- e_j = forecast error for forecast horizon j
- T = size of training set
- $m = 12$

Example: Australian tourism

```r
fc |>
  accuracy(tourism_agg, measures = list(mase = MASE, rmsse = RMSSE))
```

```
# A tibble: 555 x 7
   .model state  zone            region           .type  mase rmsse
   <chr>  <chr*> <chr*>          <chr*>           <chr> <dbl> <dbl>
 1 ets    NSW    ACT             Canberra         Test  0.546 0.513
 2 ets    NSW    ACT             <aggregated>     Test  0.546 0.513
 3 ets    NSW    Metro NSW       Central Coast    Test  0.909 0.829
 4 ets    NSW    Metro NSW       Sydney           Test  0.891 0.764
 5 ets    NSW    Metro NSW       <aggregated>     Test  0.848 0.715
 6 ets    NSW    North Coast NSW Hunter           Test  0.804 0.696
 7 ets    NSW    North Coast NSW North Coast NSW Test  1.21  1.17
 8 ets    NSW    North Coast NSW <aggregated>     Test  1.10  0.986
 9 ets    NSW    North NSW       Blue Mountains   Test  0.932 1.13
10 ets    NSW    North NSW       Central NSW      Test  1.02  0.805
# i 545 more rows
```

Example: Australian tourism

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

```
# A tibble: 5 x 3
  .model  mase rmsse
  <chr>  <dbl> <dbl>
1 ols    0.890 0.863
2 mint_s 0.878 0.866
3 wlss   0.886 0.871
4 wlsv   0.882 0.873
5 ets    0.886 0.880
```

Example: Australian tourism

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

```
# A tibble: 5 x 3
  .model   mase rmsse
  <chr>   <dbl> <dbl>
1 ols     0.890 0.863
2 mint_s  0.878 0.866
3 wlss    0.886 0.871
4 wlsv    0.882 0.873
5 ets     0.886 0.880
```

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

Example: Australian tourism

```
# A tibble: 20 x 4
# Groups:   .model [5]
   .model level     mase rmsse
   <chr>  <fct>    <dbl> <dbl>
 1 ets    National 0.806 0.755
 2 ols    National 0.812 0.768
 3 wlss   National 0.846 0.889
 4 mint_s National 0.853 0.896
 5 wlsv   National 0.883 0.934
 6 ols    State    0.902 0.905
 7 ets    State    0.921 0.919
 8 mint_s State    0.956 0.953
 9 wlss   State    0.950 0.954
10 wlsv   State    0.966 0.971
11 ols    Zone     0.932 0.912
12 mint_s Zone     0.924 0.914
13 wlss   Zone     0.931 0.924
14 wlsv   Zone     0.933 0.925
15 ets    Zone     0.936 0.935
16 mint_s Region   0.855 0.839
17 ols    Region   0.875 0.843
18 wlsv   Region   0.856 0.843
19 wlss   Region   0.864 0.844
20 ets    Region   0.866 0.858
```

Example: Australian tourism

```
# A tibble: 20 x 4
# Groups:   .model [5]
   .model level     mase rmsse
   <chr>  <fct>    <dbl> <dbl>
 1 ets    National 0.806 0.755
 2 ols    National 0.812 0.768
 3 wlss   National 0.846 0.889
 4 mint_s National 0.853 0.896
 5 wlsv   National 0.883 0.934
 6 ols    State    0.902 0.905
 7 ets    State    0.921 0.919
 8 mint_s State    0.956 0.953
 9 wlss   State    0.950 0.954
10 wlsv   State    0.966 0.971
11 ols    Zone     0.932 0.912
12 mint_s Zone     0.924 0.914
13 wlss   Zone     0.931 0.924
14 wlsv   Zone     0.933 0.925
15 ets    Zone     0.936 0.935
16 mint_s Region   0.855 0.839
17 ols    Region   0.875 0.843
18 wlsv   Region   0.856 0.843
19 wlss   Region   0.864 0.844
20 ets    Region   0.866 0.858
```

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

Mean square error bounds

Panagiotelis, Gamakumara, Athanasopoulos, and Hyndman (2021)

Distance reducing property

Let $\|\boldsymbol{u}\|_\Psi = \boldsymbol{u}'\Psi\boldsymbol{u}$. Then

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

- Ψ-projection is guaranteed to improve forecast accuracy over base forecasts *using this distance measure*.
- Distance reduction holds for any realisation and any forecast.
- OLS reconciliation minimizes Euclidean distance.

Mean square error bounds

Wickramasuriya (2021)

$$\begin{aligned}
\|\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}_{t+h}\|_2^2 &= \|\boldsymbol{M}(\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_{t+h})\|_2^2 \\
&\le \|\boldsymbol{M}\|_2^2 \|\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_{t+h}\|_2^2 \\
&= \sigma_{\text{max}}^2 \|\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_{t+h}\|_2^2
\end{aligned}$$

- σ_{max} is the largest eigenvalue of $\boldsymbol{M}$
- $\sigma_{\text{max}} \ge 1$ as $\boldsymbol{M}$ is a projection matrix.
- Every projection reconciliation is better than base forecasts using Euclidean distance.

Mean square error bounds

Wickramasuriya (2021)

$$
\begin{aligned}
&\text{tr}\Big(\text{E}[\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}^{\text{MinT}}_{t+h|t}]'[\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}^{\text{MinT}}_{t+h|t}]\Big) \\
\leq\ &\text{tr}\Big(\text{E}[\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}^{\text{OLS}}_{t+h|t}]'[\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}^{\text{OLS}}_{t+h|t}]\Big) \\
\leq\ &\text{tr}\Big(\text{E}[\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_{t+h|t}]'[\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_{t+h|t}]\Big)
\end{aligned}
$$

Using sums of variances:

- MinT reconciliation is better than OLS reconciliation
- OLS reconciliation is better than base forecasts

Non-negative forecasts

$$\min_{\boldsymbol{G}_h} \operatorname{tr}\Big(\mathrm{E}[\boldsymbol{y}_{t+h} - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t}]'[\boldsymbol{y}_{t+h} - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t}]\Big)$$
$$\text{such that } \boldsymbol{b}_{t+h|t} = \boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t} \geq 0$$

Non-negative forecasts

$$\min_{\boldsymbol{G}_h} \operatorname{tr}\Big(\mathrm{E}[\boldsymbol{y}_{t+h} - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t}]'[\boldsymbol{y}_{t+h} - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t}]\Big)$$
$$\text{such that } \boldsymbol{b}_{t+h|t} = \boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t} \geq 0$$

Solve via quadratic programming:

$$\min_{\boldsymbol{b}}\Big[\boldsymbol{b}'\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S}\boldsymbol{b} - 2\boldsymbol{b}'\boldsymbol{S}'\boldsymbol{W}_h^{-1}\hat{\boldsymbol{y}}_{T+h|T}\Big] \quad \text{s.t. } \boldsymbol{b} \geq 0$$

(Wickramasuriya, Turlach, and Hyndman, 2020)

Non-negative forecasts

$$\min_{\boldsymbol{G}_h} \text{tr}\Big(\text{E}[\boldsymbol{y}_{t+h} - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t}]'[\boldsymbol{y}_{t+h} - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t}]\Big)$$
$$\text{such that } \boldsymbol{b}_{t+h|t} = \boldsymbol{G}_h\hat{\boldsymbol{y}}_{t+h|t} \geq 0$$

Solve via quadratic programming:

$$\min_{\boldsymbol{b}}\left[\boldsymbol{b}'\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S}\boldsymbol{b} - 2\boldsymbol{b}'\boldsymbol{S}'\boldsymbol{W}_h^{-1}\hat{\boldsymbol{y}}_{T+h|T}\right] \quad \text{s.t. } \boldsymbol{b} \geq 0$$

(Wickramasuriya, Turlach, and Hyndman, 2020)

Set-negative-to-zero heuristic solution

- Negative reconciled forecasts at bottom level set to zero
- Remaining forecasts computed via aggregation

(Di Fonzo and Girolimetto, 2023b)

Reconciliation and regularization

Mishchenko, Montgomery, and Vaggi (2019):
Optimize all forecasts with an incoherence penalty

$$\min_{\hat{\boldsymbol{y}}_t} \sum_{t=1}^{T} \|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\|_2 + \lambda \sum_{t=1}^{T} \|\hat{\boldsymbol{y}}_t - \boldsymbol{S}_t \hat{\boldsymbol{b}}_t\|_2$$

Reconciliation and regularization

Mishchenko, Montgomery, and Vaggi (2019):
Optimize all forecasts with an incoherence penalty

$$\min_{\hat{\boldsymbol{y}}_t} \sum_{t=1}^{T} \|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\|_2 + \lambda \sum_{t=1}^{T} \|\hat{\boldsymbol{y}}_t - \boldsymbol{S}_t \hat{\boldsymbol{b}}_t\|_2$$

Shiratori, Kobayashi, and Takano (2020):
Optimize bottom level forecasts with an incoherence penalty

$$\min_{\hat{\boldsymbol{b}}_t} \sum_{t=1}^{T} \|\boldsymbol{b}_t - \hat{\boldsymbol{b}}_t\|_2 + \sum_{t=1}^{T} \|\Lambda(\boldsymbol{a}_t - \boldsymbol{A}_t \hat{\boldsymbol{b}}_t)\|_2$$

Outline

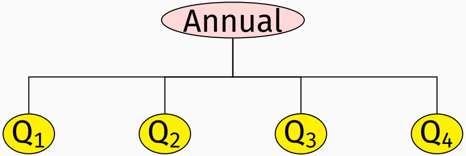
1 Improving hierarchical forecasts

2 Improving univariate forecasts

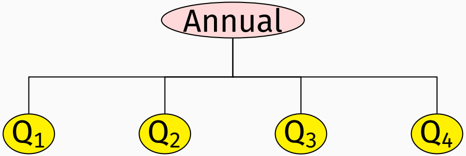
3 Improving multivariate forecasts

4 Final comments

Temporal reconciliation: quarterly data

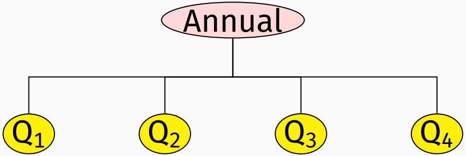


Temporal reconciliation: quarterly data



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

Temporal reconciliation: quarterly data

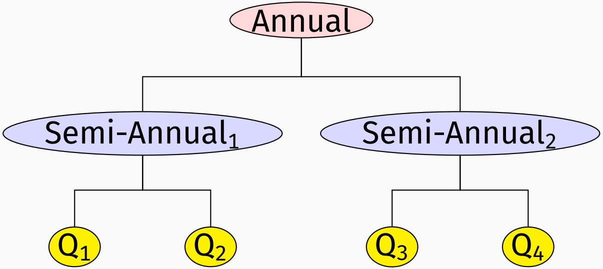


$$\boldsymbol{y}_\tau = \begin{bmatrix} x_\tau^{[4]} \\ x_{\tau,1}^{[1]} \\ x_{\tau,2}^{[1]} \\ x_{\tau,3}^{[1]} \\ x_{\tau,4}^{[1]} \end{bmatrix} \qquad \boldsymbol{S} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

τ = index of largest temporal aggregation level.

Temporal reconciliation: quarterly data

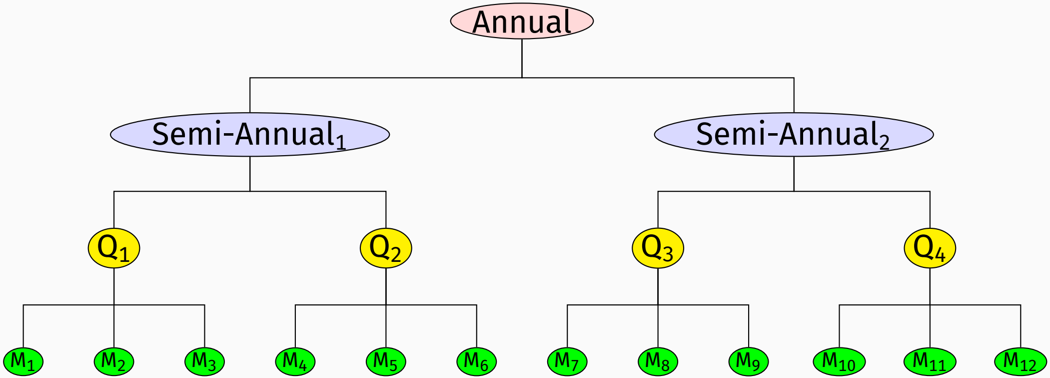


- Forecast series at each available frequency.
- Optimally combine forecasts within the same year.

$$\boldsymbol{y}_\tau = \begin{bmatrix} x_\tau^{[4]} \\ x_{\tau,1}^{[2]} \\ x_{\tau,2}^{[2]} \\ x_{\tau,1}^{[1]} \\ x_{\tau,2}^{[1]} \\ x_{\tau,3}^{[1]} \\ x_{\tau,4}^{[1]} \end{bmatrix} \qquad \boldsymbol{S} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

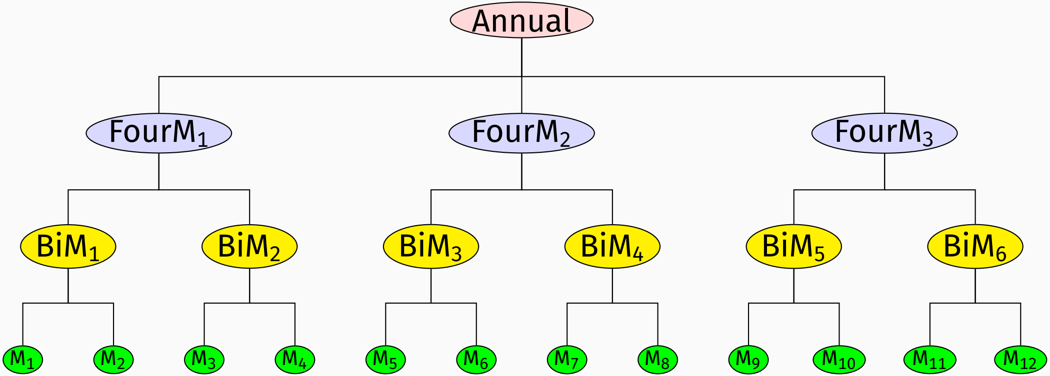
τ = index of largest temporal aggregation level.

Temporal reconciliation: monthly data



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

Temporal reconciliation: monthly data



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

Temporal reconciliation: monthly data

$$\boldsymbol{y}_\tau = \begin{bmatrix} x_\tau^{[12]} \\ \boldsymbol{x}_\tau^{[6]} \\ \boldsymbol{x}_\tau^{[4]} \\ \boldsymbol{x}_\tau^{[3]} \\ \boldsymbol{x}_\tau^{[2]} \\ \boldsymbol{x}_\tau^{[1]} \end{bmatrix} \qquad \boldsymbol{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & \vdots & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & \vdots & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ & & & & & \boldsymbol{I}_{12} & & & & & & \end{bmatrix}$$

Temporal reconciliation

For a time series $y_1, \dots, y_T$, observed at frequency m:

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

- $k \in \mathcal{K} = \{k_1, \dots, k_p\}$ denote the p factors of m in ascending order, where $k_1 = 1$ and $k_p = m$
- $x_j^{[1]} = y_t$
- A single unique hierarchy is only possible when there are no coprime pairs in $\mathcal{K}$.
- $M_k = m/k$ is seasonal period of aggregated series.

Temporal reconciliation

$$\boldsymbol{x}_\tau = \boldsymbol{S}\boldsymbol{x}_\tau^{[1]}, \qquad \boldsymbol{S} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{I} \end{bmatrix}$$

where

$$\boldsymbol{x}_\tau = \begin{bmatrix} x_\tau^{[k_p]} \\ \boldsymbol{x}_\tau^{[k_{p-1}]} \\ \vdots \\ \boldsymbol{x}_\tau^{[k_1]} \end{bmatrix} \qquad \boldsymbol{x}_\tau^{[k]} = \begin{bmatrix} x_{M_k(\tau-1)+1}^{[k]} \\ x_{M_k(\tau-1)+2}^{[k]} \\ \vdots \\ x_{M_k\tau}^{[k]} \end{bmatrix} \qquad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{1}'_m \\ \boldsymbol{I}_{m/k_{p-1}} \otimes \boldsymbol{1}'_{k_{p-1}} \\ \vdots \\ \boldsymbol{I}_{m/k_2} \otimes \boldsymbol{1}'_{k_2} \end{bmatrix}$$

τ is time index for most aggregated series,

$k \in \mathcal{K} = \{k_1, \dots, k_p\}$, $k_1 = 1$, $k_p = m$, $\tau = 1, \dots, T/m$.

Cross-temporal forecast reconciliation

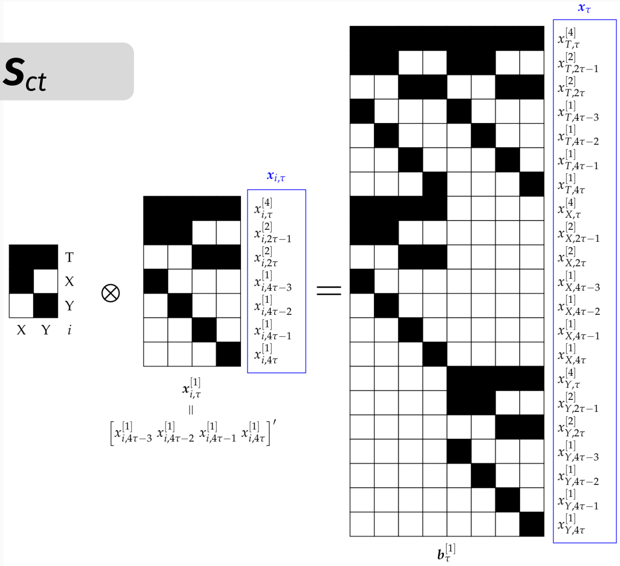
Structural matrix approach

- $\boldsymbol{S}_{cs}$ = structural cross-sectional matrix
- $\boldsymbol{S}_{te}$ = structural temporal matrix
- $\boldsymbol{S}_{ct} = \boldsymbol{S}_{cs} \otimes \boldsymbol{S}_{te}$

$$\boldsymbol{x}_\tau = \boldsymbol{S}_{ct}\boldsymbol{b}_\tau, \qquad \text{where} \qquad \boldsymbol{b}_\tau = \begin{bmatrix} \boldsymbol{x}_{1,\tau}^{[1]} \\ \vdots \\ \boldsymbol{x}_{n,\tau}^{[1]} \end{bmatrix}.$$

Cross-temporal forecast reconciliation

$$\boldsymbol{S}_{cs} \otimes \boldsymbol{S}_{te} = \boldsymbol{S}_{ct}$$



Cross-temporal forecast reconciliation

Constraint matrix approach

- $\boldsymbol{C}_{cs}$ = cross-sectional constraint matrix
- $\boldsymbol{C}_{te}$ = temporal constraint matrix

$$\boldsymbol{C}_{ct}\boldsymbol{x}_\tau = \boldsymbol{0} \qquad \text{where} \qquad \boldsymbol{C}_{ct} = \begin{bmatrix} (\boldsymbol{0}_{(n_a m \times n k^*)} \quad \boldsymbol{I}_m \otimes \boldsymbol{C}_{cs})\boldsymbol{P}' \\ \boldsymbol{I}_n \otimes \boldsymbol{C}_{te} \end{bmatrix}$$

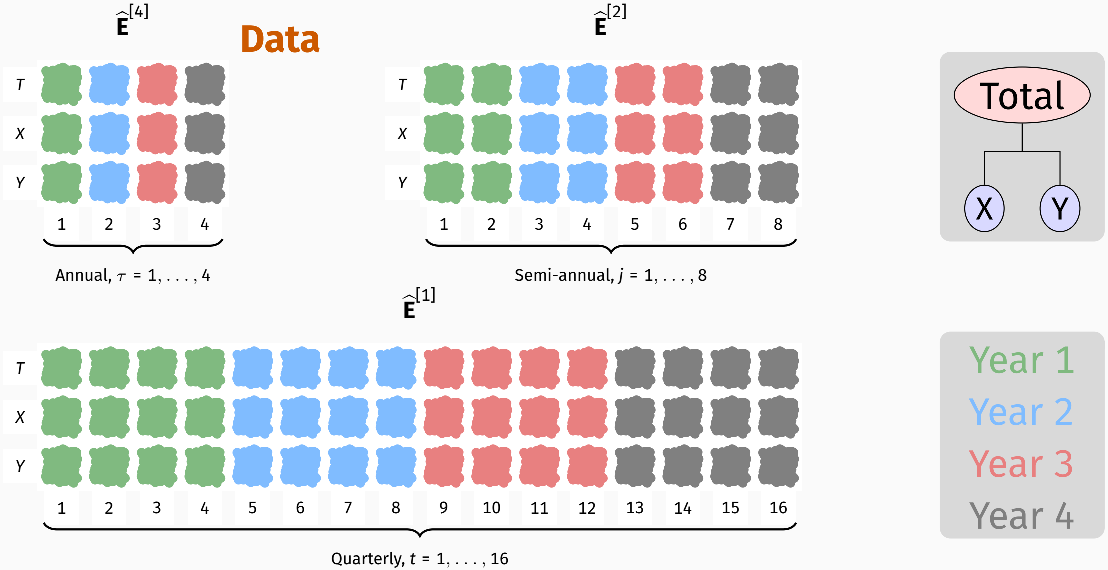
- $k^* = \displaystyle\sum_{k \in \mathcal{K} \setminus \{k_1\}} \frac{m}{k}$
- $\boldsymbol{P}$ = the permutation matrix such that $\boldsymbol{P}\text{vec}(\boldsymbol{Y}_\tau) = \text{vec}(\boldsymbol{Y}_\tau')$.

Cross-temporal probabilistic forecast reconciliation

Nonparametric bootstrap

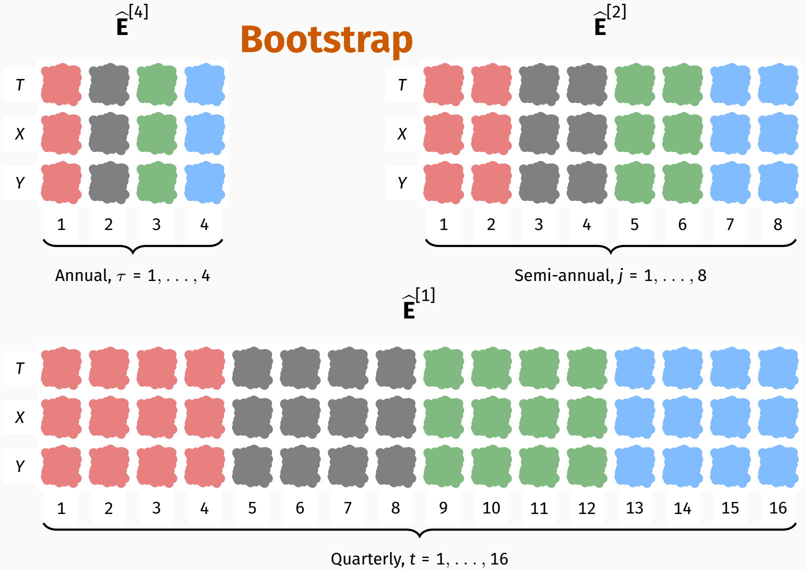
- Simulate future sample paths from all models using bootstrapped residuals, then reconcile them to obtain coherent sample paths.
- Need to generate samples that preserve cross-temporal relationships.
- Draw residual samples of all series at same time from most temporally aggregated level.
- Residuals for other levels obtained using the corresponding time indices.

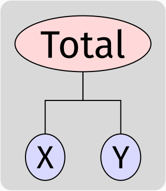
Cross-temporal probabilistic forecast reconciliation



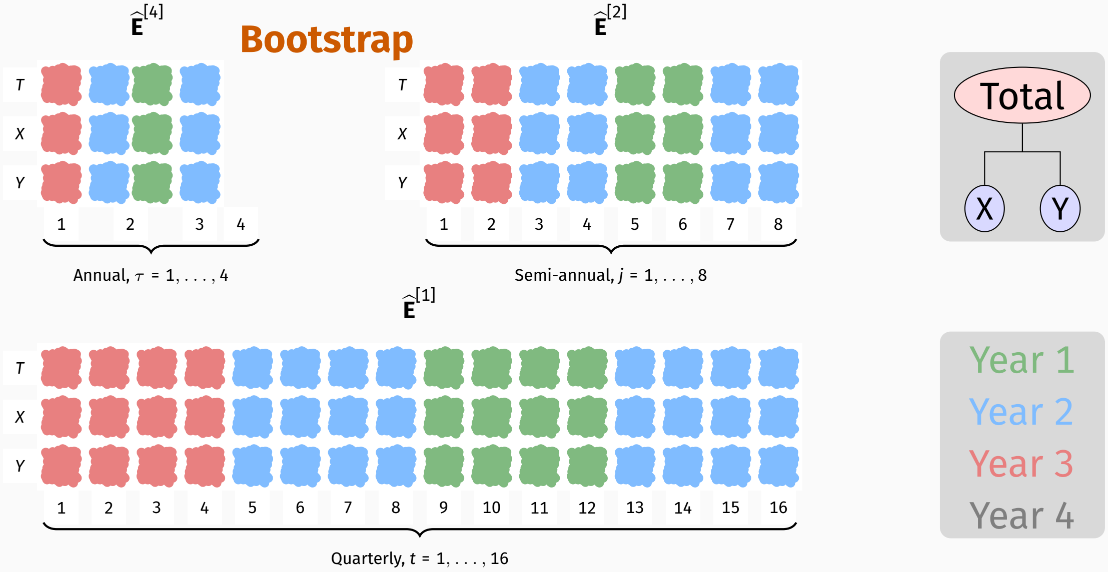
Cross-temporal probabilistic forecast reconciliation

Bootstrap



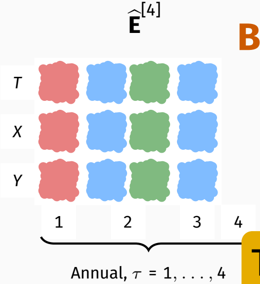


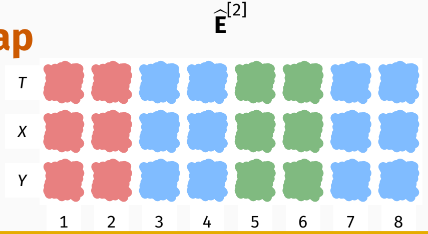
Cross-temporal probabilistic forecast reconciliation



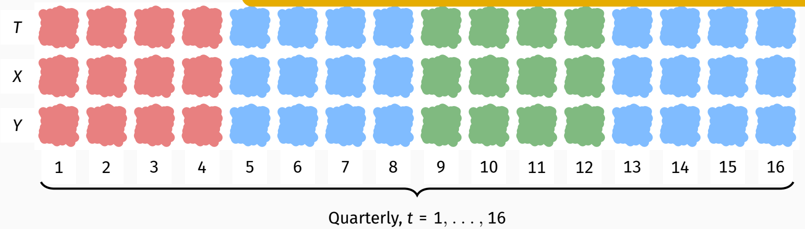
Cross-temporal probabilistic forecast reconciliation

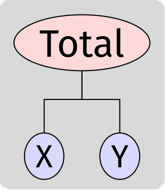
Bootstrap





The "year" can start in any quarter, giving overlapping blocks.

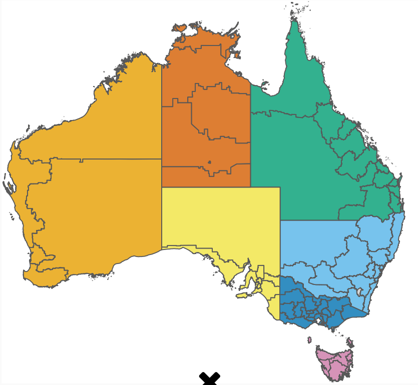




Year 1
Year 2
Year 3
Year 4

Monthly Australian Tourism Demand

Geographical division



×

Purpose of travel

Holiday, Visiting friends & relatives, Business, Other

- **Grouped ts**
  (geographical divisions × purpose of travel)

	AUS	States	Zones*	Regions	Tot
geographical	1	7	21	76	105
purpose	4	28	84	304	420
total	5	35	105	380	**525**

$n_a = 221$, $n_b = 304$, and $\mathbf{n = 525}$

- **Temporal framework**, frequencies:
  - Monthly
  - Bi-Monthly
  - Quarterly
  - Four-Monthly
  - Semi-Annual
  - Annual

Monthly Australian Tourism Demand

- Monthly data: January 1998 to December 2016
- Time series cross-validation; initial training set 10 years.
- One-month increase in each training set
- For each training set, compute temporally aggregated series for $k \in \{1, 2, 3, 4, 6, 12\}$, and produce forecasts up to $h_2 = 6$, $h_3 = 4$, $h_4 = 3$, $h_6 = 2$ and $h_{12} = 1$ steps ahead.
- Automatic ETS forecasts on log-transformed data

Monthly Australian Tourism Demand

Reconciliation approaches

- Cross-temporal **bottom-up** and **partly bottom-up**

$\text{ct}(bu)$	$\text{ct}(shr_{cs}, bu_{te})$	$\text{ct}(wlsv_{te}, bu_{cs})$

- Optimal forecast reconciliation with **one-step residuals**

$\text{oct}(ols)$	$\text{oct}(struc)$	$\text{oct}(wlsv)$	$\text{oct}(bdshr)$

- Optimal forecast reconciliation with **multi-step residuals**

$\text{oct}_h(hbshr)$	$\text{oct}_h(bshr)$	$\text{oct}_h(hshr)$	$\text{oct}_h(shr)$

Monthly Australian tourism data – CRPS skill scores

Worse than benchmark **Best**

	$\forall k \in \{12, 6, 4, 3, 2, 1\}$	$k = 1$
base	1.000	1.000
ct(bu)	1.321	1.077
ct(shr_{cs}, bu_{te})	1.057	0.976
ct($wlsv_{te}, bu_{cs}$)	1.062	0.976
oct(ols)	0.989	0.982
oct($struc$)	0.982	0.970
oct($wlsv$)	0.987	0.952
oct($bdshr$)	0.975	**0.949**
oct$_h$($hbshr$)	0.989	0.982
oct$_h$($bshr$)	0.994	0.988
oct$_h$($hshr$)	**0.969**	0.953
oct$_h$(shr)	1.007	1.000

Outline

1. Improving hierarchical forecasts
2. Improving univariate forecasts
3. Improving multivariate forecasts
4. Final comments

Outline

Software

Package	Language	Cross-sectional	Temporal	Cross-temporal	Probabilistic
`hts`	R	✓			
`thief`	R		✓		
`fable`	R	✓			✓
`FoReco`	R	✓	✓	✓	✓
`pyhts`	Python	✓	✓		
`hierarchicalforecast`	Python	✓			✓

- `hts`, `thief`, and `FoReco` use `ts` objects
- `fable` uses `tsibble` objects
- `fable` has plans to implement temporal and cross-temporal reconciliation

Thanks!

More information

- robjhyndman.com
- aus.social/@robjhyndman
- @robjhyndman
- rob.hyndman@monash.edu

References

- Ashouri, M, RJ Hyndman, and G Shmueli (2022). Fast forecast reconciliation using linear models. *J Computational & Graphical Statistics* **31**(1), 263–282.
- Athanasopoulos, G, RJ Hyndman, N Kourentzes, and A Panagiotelis (2024). "Forecast reconciliation: a review". in press.
- Athanasopoulos, G, RJ Hyndman, N Kourentzes, and F Petropoulos (2017). Forecasting with temporal hierarchies. *European J Operational Research* **262**(1), 60–74.
- Di Fonzo, T and D Girolimetto (2023a). Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives. *International J Forecasting* **39**(1), 39–57.
- Di Fonzo, T and D Girolimetto (2023b). Spatio-temporal reconciliation of solar forecasts. *Solar Energy* **251**, 13–29.
- Di Fonzo, T and D Girolimetto (2024). Forecast combination-based forecast reconciliation: Insights and extensions. *International J Forecasting*. in press.

References

- Girolimetto, D, G Athanasopoulos, T Di Fonzo, and RJ Hyndman (2024). Cross-temporal probabilistic forecast reconciliation. *International J Forecasting*. in press.
- Hyndman, RJ, RA Ahmed, G Athanasopoulos, and HL Shang (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis* **55**(9), 2579–2589.
- Hyndman, RJ, A Lee, and E Wang (2016). Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational Statistics & Data Analysis* **97**, 16–32.
- Mishchenko, K, M Montgomery, and F Vaggi (2019). A self-supervised approach to hierarchical forecasting with applications to groupwise synthetic controls. In: *Proceedings of the Time Series Workshop at the 36th International Conference on Machine Learning, Long Beach, California*. PMLR 97.

References

- Panagiotelis, A, P Gamakumara, G Athanasopoulos, and RJ Hyndman (2021). Forecast reconciliation: A geometric view with new insights on bias correction. *International J Forecasting* **37**(1), 343–359.
- Shiratori, T, K Kobayashi, and Y Takano (2020). Prediction of hierarchical time series using structured regularization and its application to artificial neural networks. *PLOS ONE* **15**(11), e0242099.
- Wickramasuriya, SL (2021). Properties of point forecast reconciliation approaches. *arXiv preprint arXiv:2103.11129*.
- Wickramasuriya, SL, G Athanasopoulos, and RJ Hyndman (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *J American Statistical Association* **114**(526), 804–819.
- Wickramasuriya, SL, BA Turlach, and RJ Hyndman (2020). Optimal non-negative forecast reconciliation. *Statistics & Computing* **30**(5), 1167–1182.