The geometry of forecast reconciliation

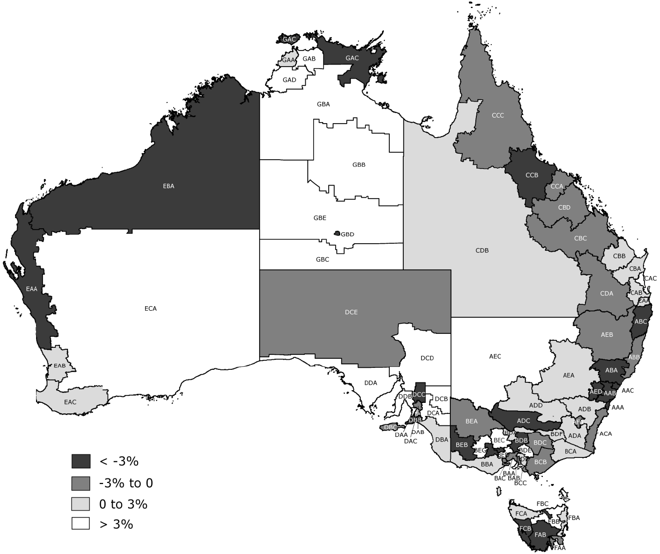
Rob J Hyndman

28 August 2020

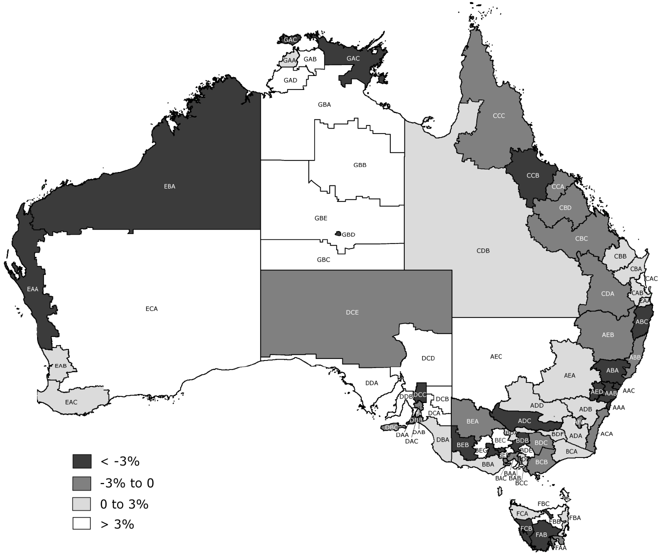
Outline

Outline

Australian tourism



Australian tourism

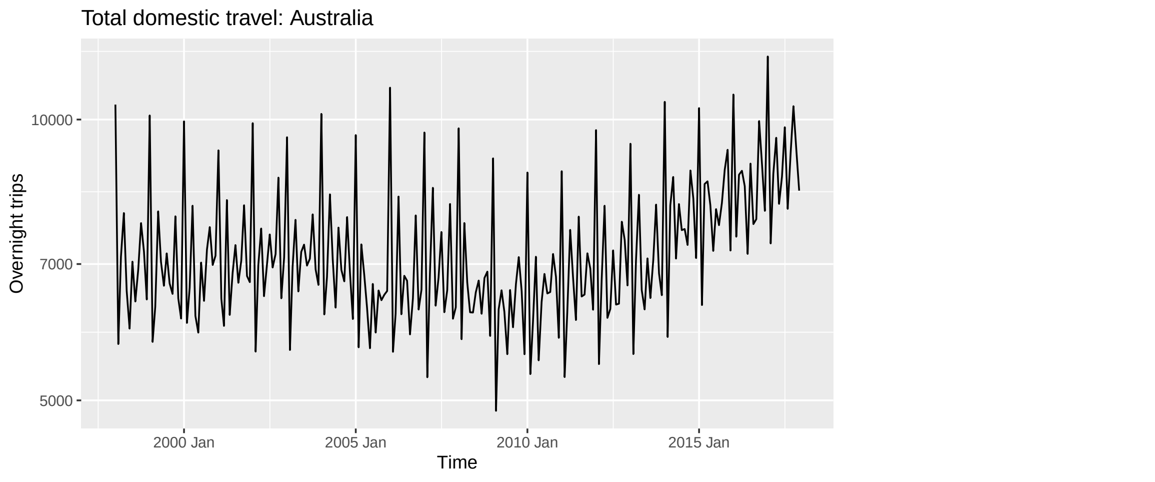


- Monthly data on visitor night from 1998 – 2017
- From *National Visitor Survey*, annual interviews of 120,000 Australians aged 15+.
- Geographical hierarchy split by
  - 7 states
  - 27 zones
  - 75 regions

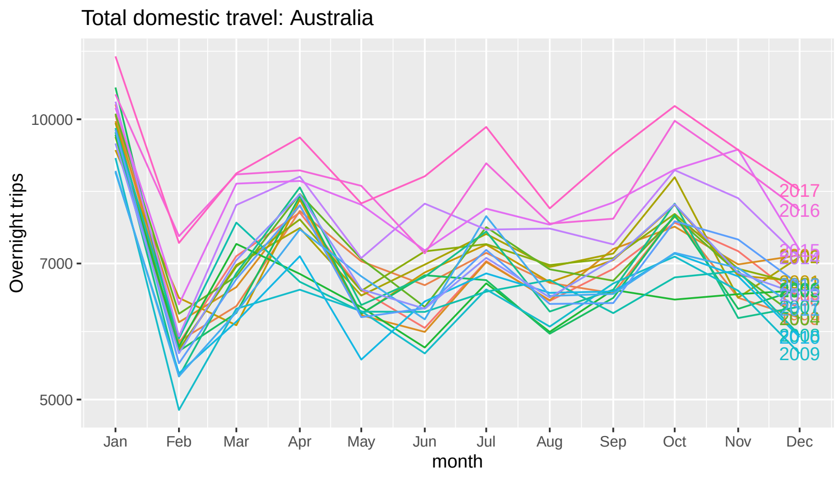
Australian tourism data

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

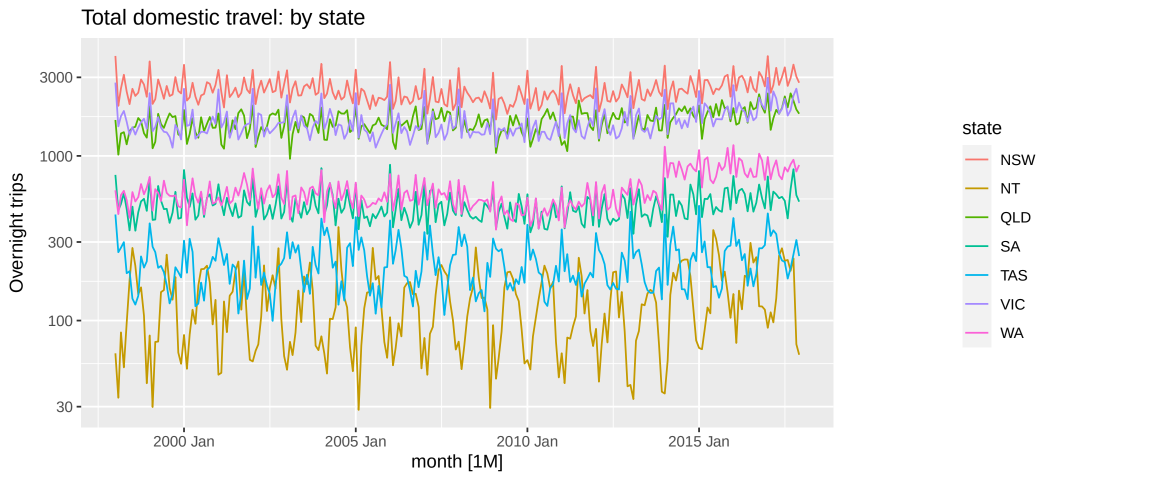
Australian tourism data



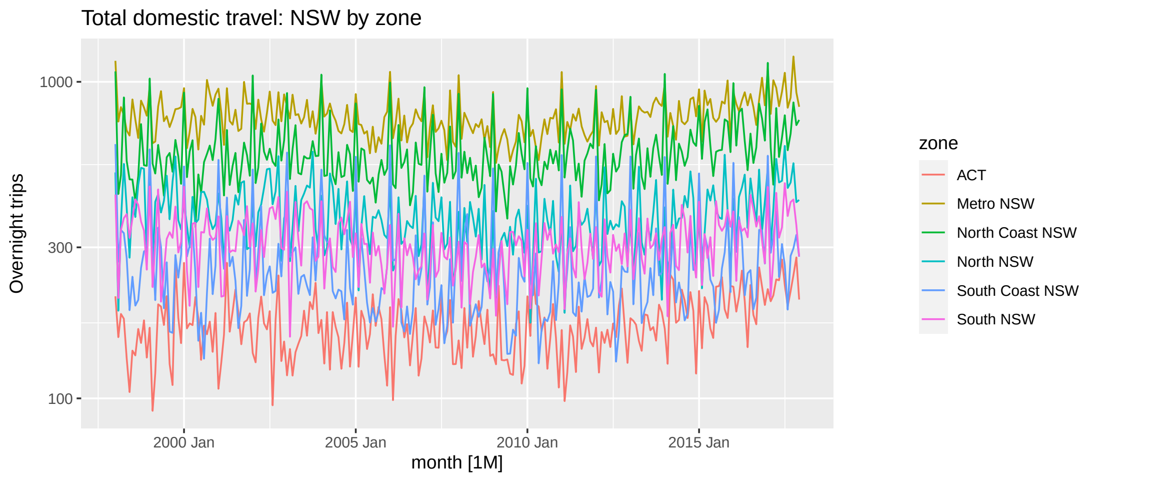
Australian tourism data



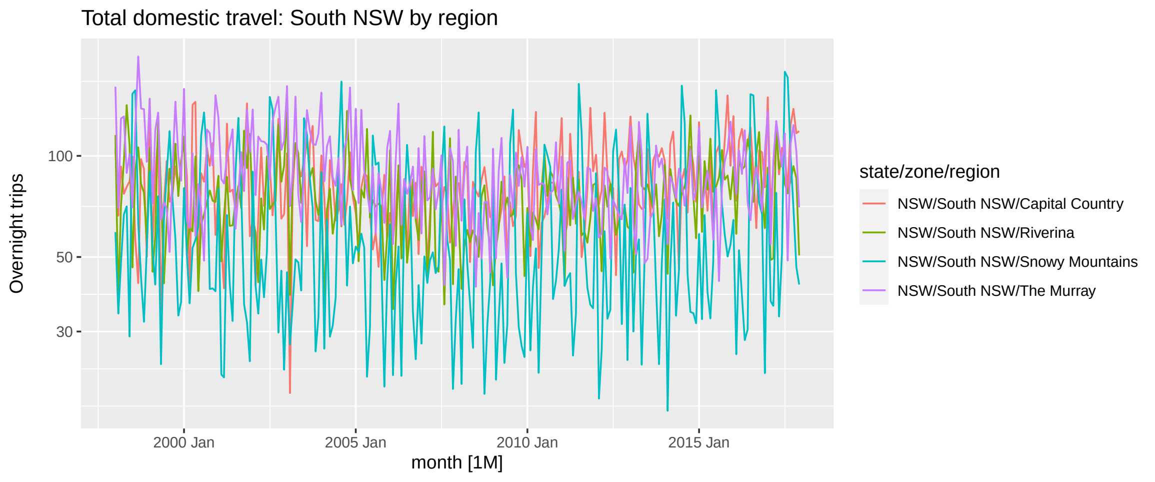
Australian tourism data



Australian tourism data

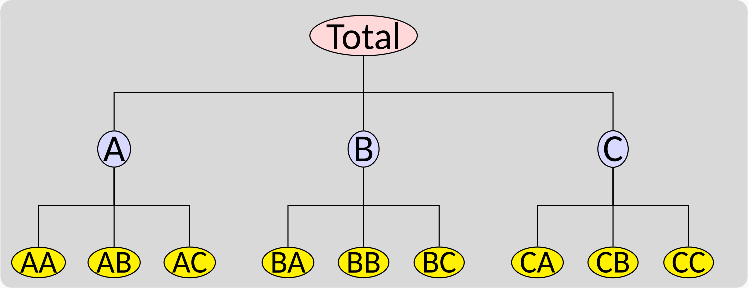


Australian tourism data



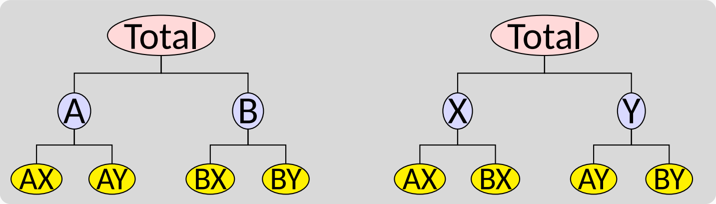
Hierarchical time series

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



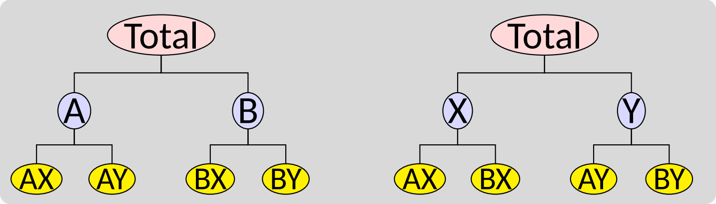
Grouped time series

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



Grouped time series

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



Examples

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

Outline

The problem

How to produce **coherent** forecasts at all nodes?

The problem

How to produce **coherent** forecasts at all nodes?

Old approaches (pre 2009)

- Bottom-up forecasting
- Top-down forecasting
- Middle-out forecasting

The problem

How to produce **coherent** forecasts at all nodes?

Old approaches (pre 2009)

- Bottom-up forecasting
- Top-down forecasting
- Middle-out forecasting

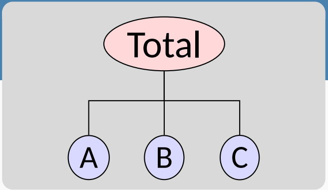
Forecast reconcilation approach

1. Forecast all series at all levels of aggregation using an automatic forecasting algorithm. (e.g., `ETS`, `ARIMA`, ...)
2. Reconcile the resulting forecasts so they are coherent using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).

Key forecast reconciliation papers

- Hyndman, Ahmed, Athanasopoulos, Shang (2011 *CSDA*) Optimal combination forecasts for hierarchical time series.
- Athanasopoulos, Ahmed, Hyndman (2009 *IJF*) Hierarchical forecasts for Australian domestic tourism.
- Hyndman, Lee, Wang (2016 *CSDA*) Fast computation of reconciled forecasts for hierarchical and grouped time series.
- Wickramasuriya, Athanasopoulos, Hyndman (2019 *JASA*) Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization.
- Panagiotelis, Gamakumara, Athanasopoulos, Hyndman (2020 *IJF*) Forecast reconciliation: A geometric view with new insights on bias correction.
- Panagiotelis, Gamakumara, Athanasopoulos, Hyndman (2020) Probabilistic forecast reconciliation: properties, evaluation and score optimisation.

Hierarchical and grouped time series



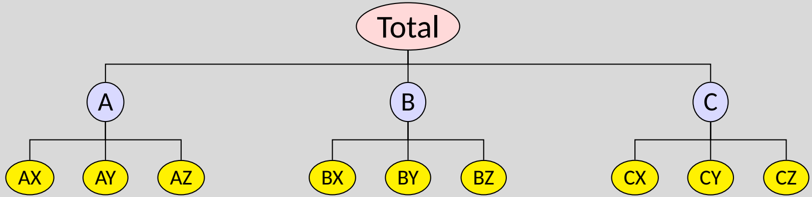
Every collection of time series with linear constraints can be written as

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

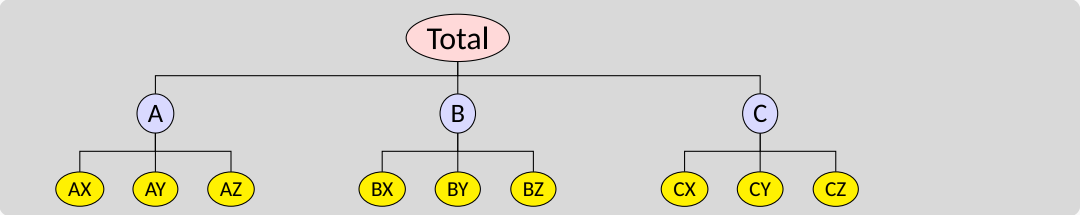
- $\boldsymbol{y}_t$ = vector of all series at time t
- y_t = aggregate of all series at time t.
- $y_{X,t}$ = value of series X at time t.
- $\boldsymbol{b}_t$ = vector of most disaggregated series at time t
- $\boldsymbol{S}$ = "summing matrix" containing the linear constraints.

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

Hierarchical time series

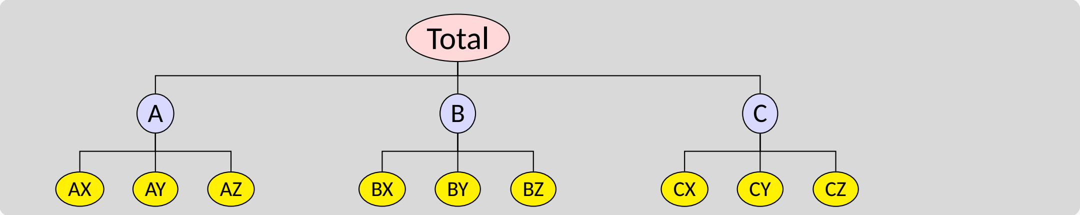


Hierarchical time series



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

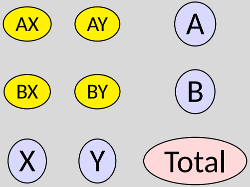
Hierarchical time series



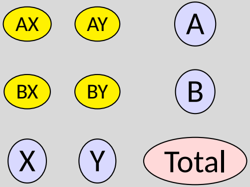
$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1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

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

Grouped data

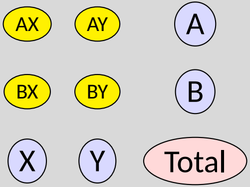


Grouped data



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

Grouped data



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

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

Definitions

Coherent subspace

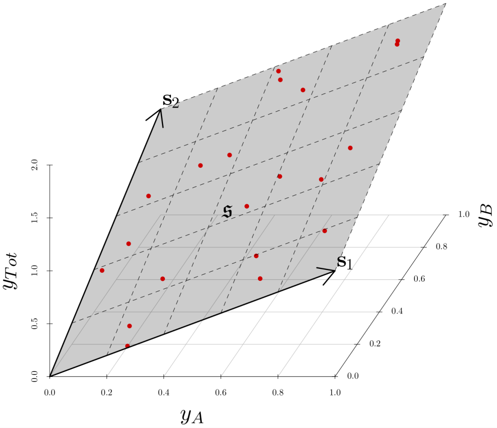
m-dimensional linear subspace $\mathfrak{s} \subset \mathbb{R}^n$ for which linear constraints hold for all $\boldsymbol{y} \in \mathfrak{s}$.

Hierarchical time series

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

Coherent point forecasts

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



$y_{Tot} = y_A + y_B$

Definitions

Coherent subspace

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

Hierarchical time series

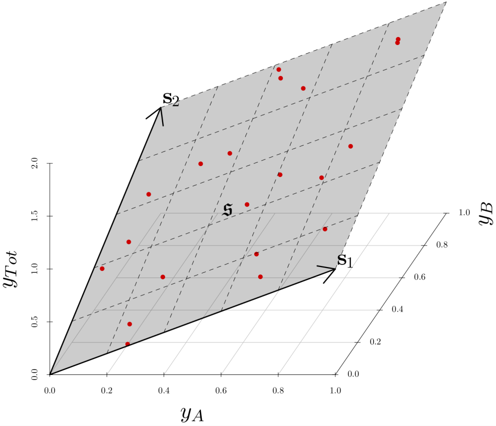
An n-dimensional multivariate time series such that $\boldsymbol{y}_t \in \mathfrak{s} \quad \forall t$.

Coherent point forecasts

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

Base forecasts

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



$y_{Tot} = y_A + y_B$

Definitions

Coherent subspace

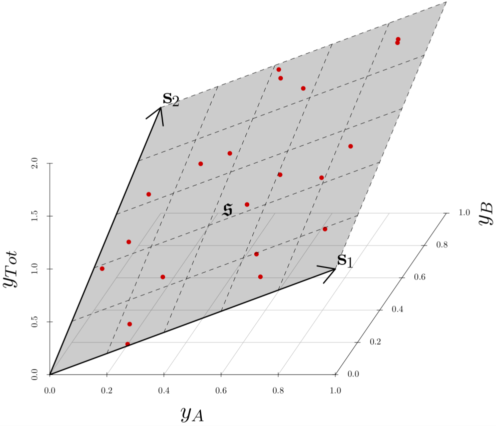
m-dimensional linear subspace $\mathfrak{s} \subset \mathbb{R}^n$ for which linear constraints hold for all $\boldsymbol{y} \in \mathfrak{s}$.

Hierarchical time series

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

Coherent point forecasts

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



$y_{Tot} = y_A + y_B$

Base forecasts

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

Reconciled forecasts

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

Linear reconciliation

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

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

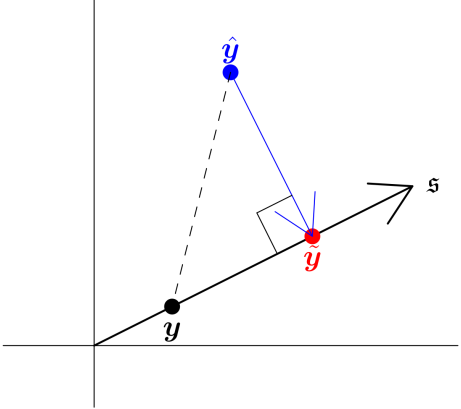
- $\boldsymbol{G}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_{T+h|T}$ to get bottom-level forecasts.
- $\boldsymbol{S}$ creates linear combinations.
- e.g., OLS reconciliation: $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$

Linear reconciliation

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

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

- **G** extracts and combines base forecasts $\hat{\boldsymbol{y}}_{T+h|T}$ to get bottom-level forecasts.
- $\boldsymbol{S}$ creates linear combinations.
- e.g., OLS reconciliation: $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$

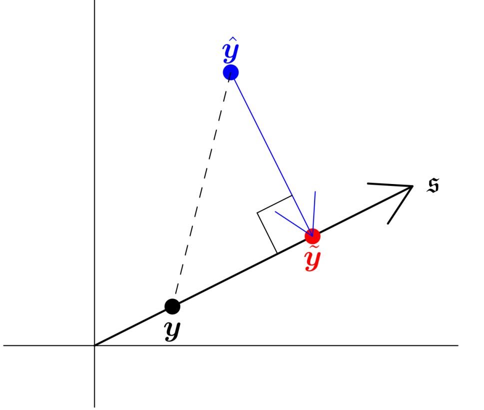


Linear reconciliation

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

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

- **G** extracts and combines base forecasts $\hat{\boldsymbol{y}}_{T+h|T}$ to get bottom-level forecasts.
- $\boldsymbol{S}$ creates linear combinations.
- e.g., OLS reconciliation: $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$



Projections

Suppose $\boldsymbol{SG}$ is a projection onto $\mathfrak{s}$, then

- Coherent base forecasts are unchanged.
- Unbiased base forecasts remain unbiased.

- Orthogonal projections lead to smallest possible adjustments of base forecasts.

Linear reconciliation

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

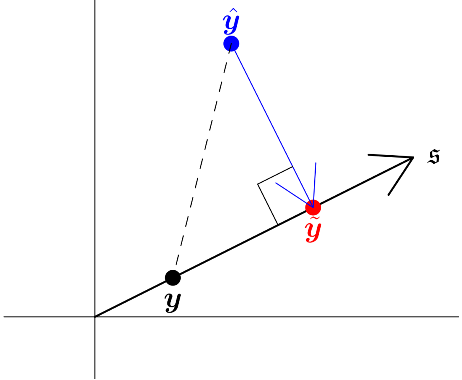
$$\tilde{\boldsymbol{y}}_{t+h|t} = \boldsymbol{S}\boldsymbol{G}\hat{\boldsymbol{y}}_{t+h|t}$$

- **G** extracts and combines base forecasts $\hat{\boldsymbol{y}}_{T+h|T}$ to get bottom-level forecasts.
- **S** creates linear combinations.
- e.g., OLS reconciliation: $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$

Distance reducing property

If $\boldsymbol{S}\boldsymbol{G}$ is an orthogonal projection onto $\mathfrak{s}$ then:

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



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

Linear projections

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

Variance

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

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

Minimum trace (MinT) reconciliation

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

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

Linear projections

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

Variance

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

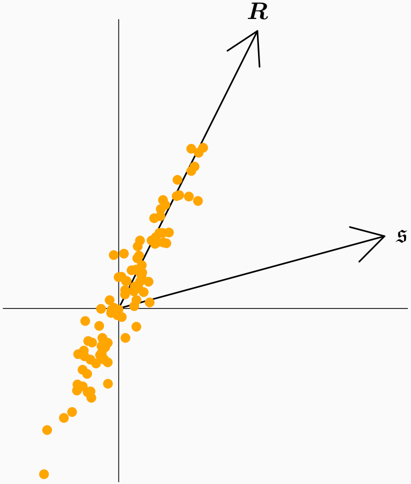
where $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.

Minimum trace (MinT) reconciliation

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

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

- R is the most likely direction of deviations from $\mathfrak{s}$.
- Orange: in-sample errors



Linear projections

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

Variance

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

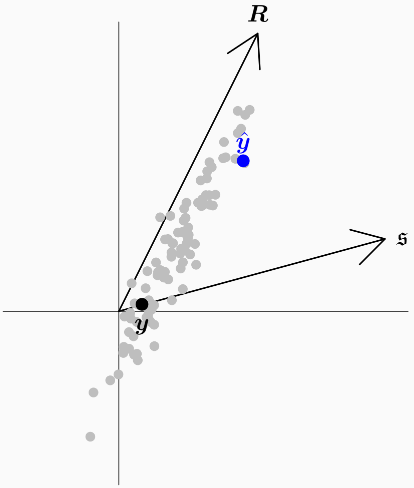
where $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.

Minimum trace (MinT) reconciliation

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

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

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



Linear projections

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

Variance

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

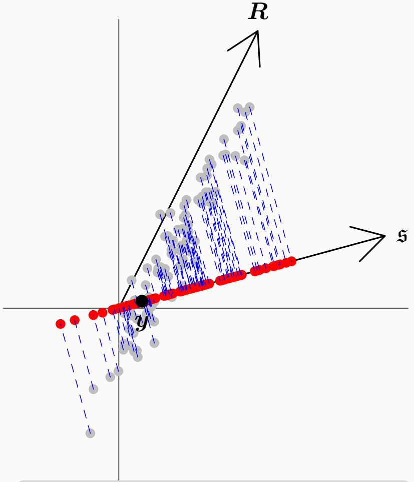
where $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.

Minimum trace (MinT) reconciliation

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

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

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



Orthogonal projection

Linear projections

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

Variance

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

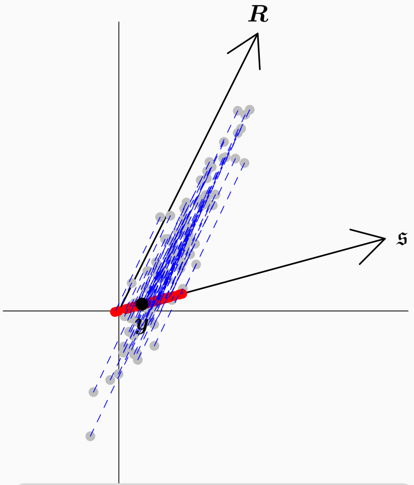
where $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$.

Minimum trace (MinT) reconciliation

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

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

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



Oblique projection

Linear projections

$$\tilde{y}_{T+h|T} = SG\hat{y}_{T+h|T}$$

Reconciliation method	G
OLS	$(S'S)^{-1}S'$
WLS	$(S'\Lambda S)^{-1}S'\Lambda$
MinT(Sample)	$(S'\hat{W}_{\text{sam}}^{-1}S)^{-1}S'\hat{W}_{\text{sam}}^{-1}$
MinT(Shrink)	$(S'\hat{W}_{\text{shr}}^{-1}S)^{-1}S'\hat{W}_{\text{shr}}^{-1}$

These approximate MinT by assuming $W_h = k_h W_1$.

- Λ is diagonal matrix
- $\hat{W}_{\text{sam}}$ is sample estimate of the residual covariance matrix
- $\hat{W}_{\text{shr}}$ is shrinkage estimator $\tau\text{diag}(\hat{W}_{\text{sam}}) + (1-\tau)\hat{W}_{\text{sam}}$ where $\tau = \dfrac{\sum_{i\neq j}\hat{\text{Var}}(\hat{\sigma}_{ij})}{\sum_{i\neq j}\hat{\sigma}_{ij}^2}$ and σ_{ij} denotes the (i,j)th element of $\hat{W}_{\text{sam}}$.

Outline

Coherent probabilistic forecasts

Coherent probabilistic forecasts

Given the triple $(\mathbb{R}^m, \mathscr{F}_{\mathbb{R}^m}, \nu)$, a coherent probability triple $(\mathfrak{s}, \mathscr{F}_{\mathfrak{s}}, \breve{\nu})$ is such that

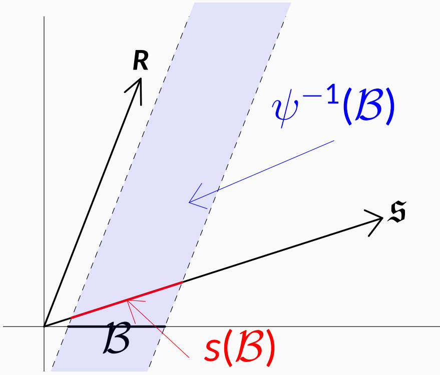
$$\breve{\nu}(s(\mathcal{B})) = \nu(\mathcal{B}) \quad \forall \mathcal{B} \in \mathscr{F}_{\mathbb{R}^m}.$$

Probabilistic forecast reconciliation

The reconciled probability measure of $\hat{\nu}$ wrt $\psi(.)$ is such that

$$\tilde{\nu}(\mathcal{B}) = \hat{\nu}(\psi^{-1}(\mathcal{B})) \qquad \forall \mathcal{B} \in \mathscr{F}_{\mathfrak{s}}\,,$$

where $\psi^{-1}(\mathcal{B}) := \{\mathbf{y} \in \mathbb{R}^n : \psi(\mathbf{y}) \in \mathcal{B}\}$ is the pre-image of $\mathcal{B}$, that is the set of all points in $\mathbb{R}^n$ that $\psi(.)$ maps to a point in $\mathcal{B}$.



Construction of reconciled distributions

Reconciled density of bottom-level

Density of bottom-level series under reconciled distribution is

$$\tilde{f}_{\boldsymbol{b}}(\boldsymbol{b}) = |\boldsymbol{G}^*| \int \hat{f}(\boldsymbol{G}^- \boldsymbol{b} + \boldsymbol{G}_\perp \boldsymbol{a}) d\boldsymbol{a}$$

- $\hat{f}$ is density of incoherent base probabilistic forecast
- $\boldsymbol{G}^-$ is $n \times m$ generalised inverse of $\boldsymbol{G}$ st $\boldsymbol{G}\boldsymbol{G}^- = \boldsymbol{I}$
- $\boldsymbol{G}_\perp$ is $n \times (n-m)$ orthogonal complement to $\boldsymbol{G}$ st $\boldsymbol{G}\boldsymbol{G}_\perp = \boldsymbol{0}$
- $\boldsymbol{G}^* = \left(\boldsymbol{G}^- \vdots \boldsymbol{G}_\perp\right)$, and $\boldsymbol{b}$ and $\boldsymbol{a}$ are obtained via

  the change of variables $\boldsymbol{y} = \boldsymbol{G}^* \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{a} \end{pmatrix}$

Construction of reconciled distributions

Reconciled density of full hierarchy

Density of full hierarchy under reconciled distribution is

$$\tilde{f}_{\boldsymbol{y}}(\boldsymbol{y}) = |\boldsymbol{S}^*|\tilde{f}_{\boldsymbol{b}}(\boldsymbol{S}^-\boldsymbol{y})\mathbb{1}\{\boldsymbol{y} \in \mathfrak{s}\}$$

- $\boldsymbol{S}^* = \begin{pmatrix} \boldsymbol{S}^- \\ \boldsymbol{S}'_\perp \end{pmatrix}$
- $\boldsymbol{S}^-$ is $m \times n$ generalised inverse of $\boldsymbol{S}$ such that $\boldsymbol{S}^-\boldsymbol{S} = \boldsymbol{I}$,
- $\boldsymbol{S}_\perp$ is $n \times (n - m)$ orthogonal complement to $\boldsymbol{S}$ such that $\boldsymbol{S}'_\perp\boldsymbol{S} = \boldsymbol{0}$.

Construction of reconciled distributions

Reconciled density of full hierarchy

Density of full hierarchy under reconciled distribution is

$$\tilde{f}_{\boldsymbol{y}}(\boldsymbol{y}) = |\boldsymbol{S}^*|\tilde{f}_{\boldsymbol{b}}(\boldsymbol{S}^-\boldsymbol{y})\mathbb{1}\{\boldsymbol{y} \in \mathfrak{s}\}$$

- $\boldsymbol{S}^* = \begin{pmatrix} \boldsymbol{S}^- \\ \boldsymbol{S}'_\perp \end{pmatrix}$
- $\boldsymbol{S}^-$ is $m \times n$ generalised inverse of $\boldsymbol{S}$ such that $\boldsymbol{S}^-\boldsymbol{S} = \boldsymbol{I}$,
- $\boldsymbol{S}_\perp$ is $n \times (n-m)$ orthogonal complement to $\boldsymbol{S}$ such that $\boldsymbol{S}'_\perp\boldsymbol{S} = \boldsymbol{0}$.

Gaussian reconciliation

If the incoherent base forecasts are $\text{N}(\hat{\mu}, \hat{\Sigma})$,
then the reconciled density is $\text{N}(\boldsymbol{S}\boldsymbol{G}\hat{\mu}, \boldsymbol{S}\boldsymbol{G}\hat{\Sigma}\boldsymbol{G}'\boldsymbol{S}')$.

Simulation from a reconciled distribution

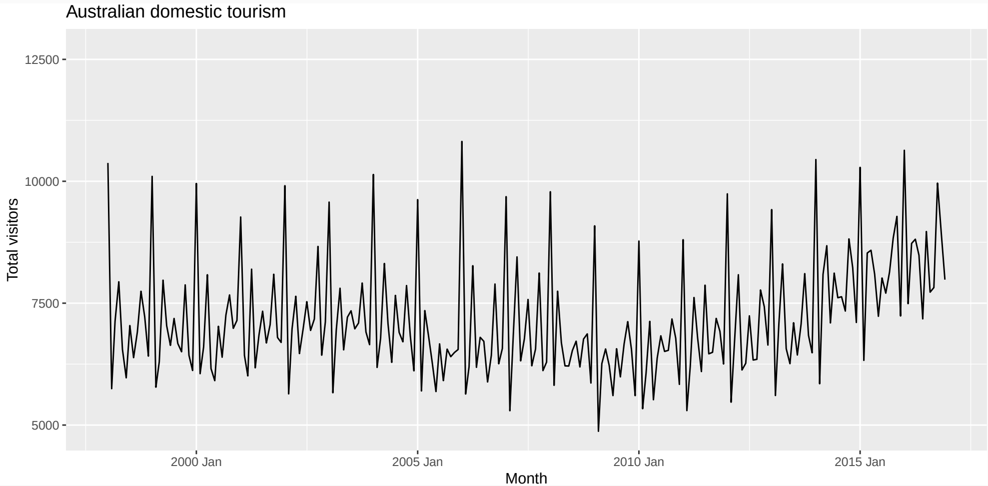
Suppose that $\left(\hat{\mathbf{y}}^{[1]}, \dots, \hat{\mathbf{y}}^{[L]}\right)$ is a sample drawn from an incoherent probability measure $\hat{\nu}$. Then $\left(\tilde{\mathbf{y}}^{[1]}, \dots, \tilde{\mathbf{y}}^{[L]}\right)$ where $\tilde{\mathbf{y}}^{[\ell]} := \psi(\hat{\mathbf{y}}^{[\ell]})$ for $\ell = 1, \dots, L$, is a sample drawn from the reconciled probability measure $\tilde{\nu}$.

- So reconciling sample paths from incoherent distributions works.

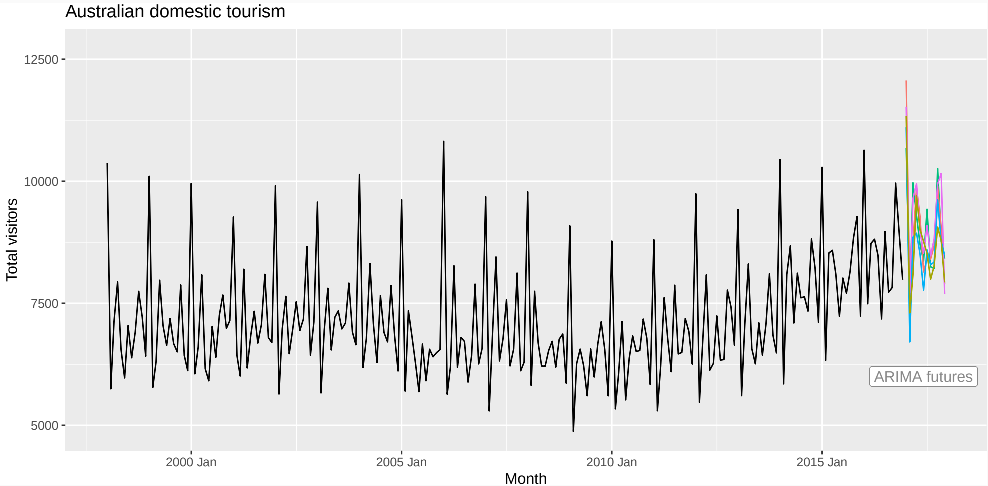
Outline

1. Hierarchical and grouped time series
2. Forecast reconciliation using projections
3. Probabilistic forecast reconciliation
4. **Evaluating probabilistic forecasts**
5. Example: Australian tourism
6. Example: Australian electricity generation

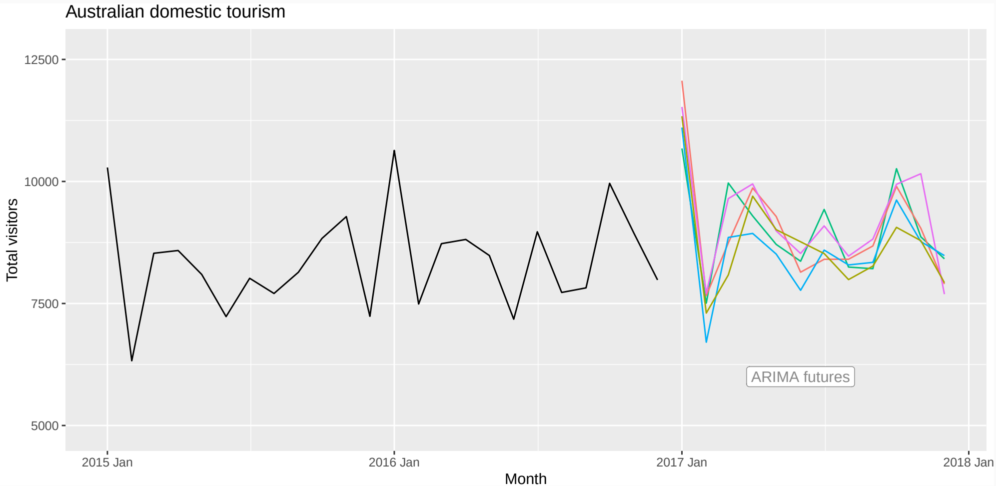
Evaluating probabilistic forecasts



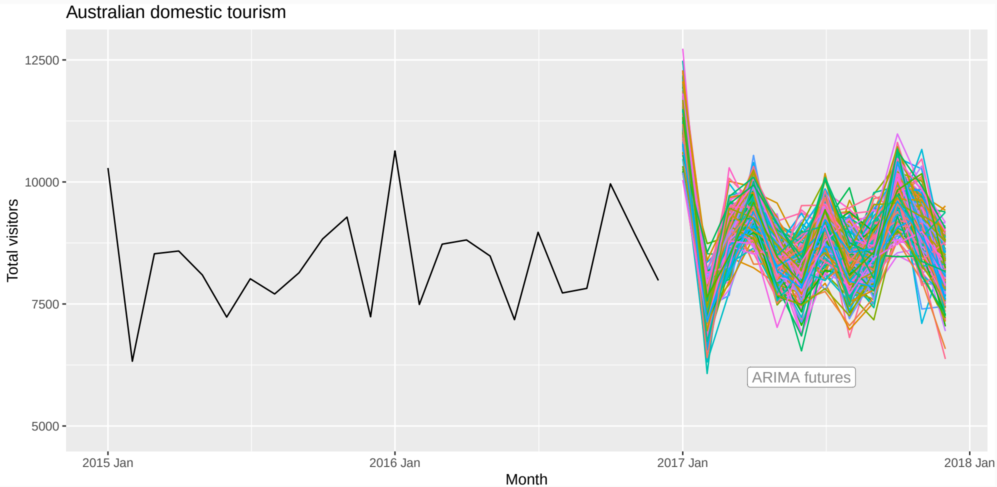
Evaluating probabilistic forecasts



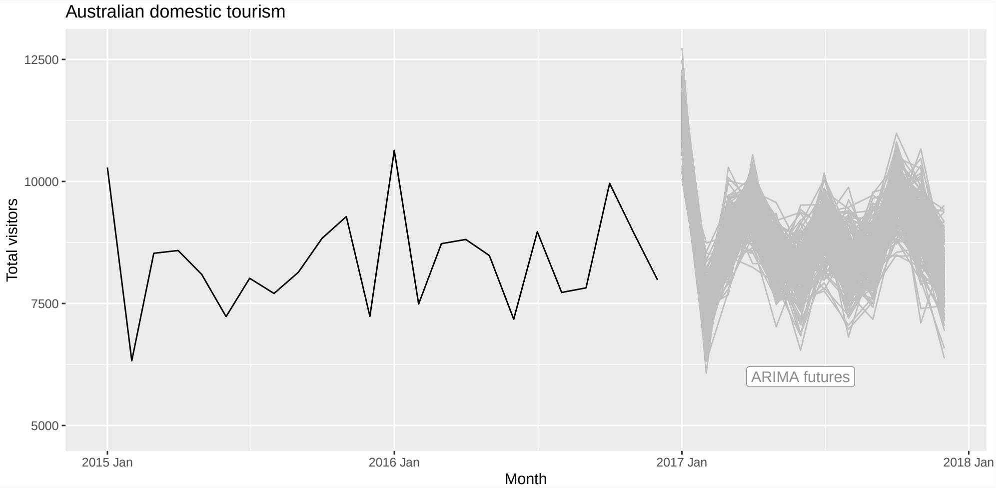
Evaluating probabilistic forecasts



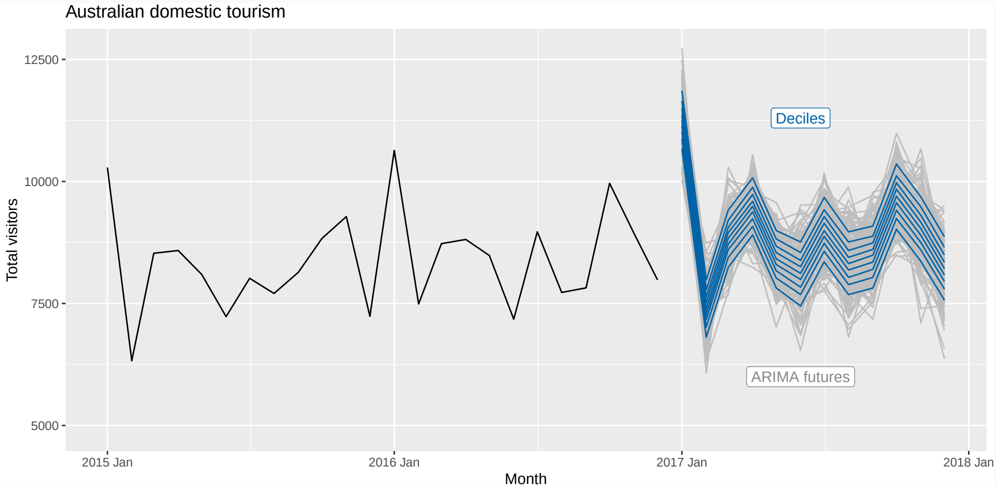
Evaluating probabilistic forecasts



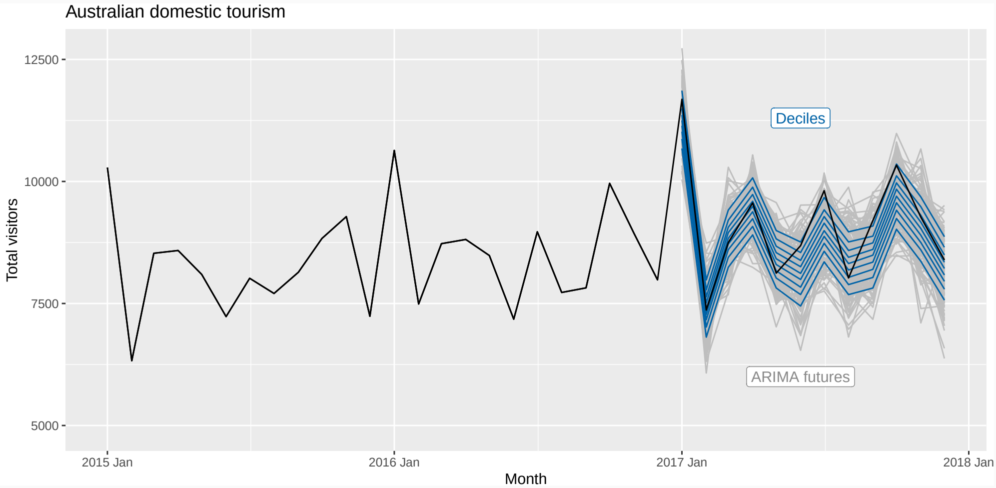
Evaluating probabilistic forecasts



Evaluating probabilistic forecasts



Evaluating probabilistic forecasts



Evaluating probabilistic forecasts

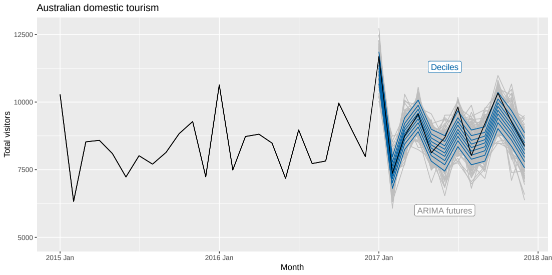
$q_{p,t}$ = quantile forecast with prob. p at time t.

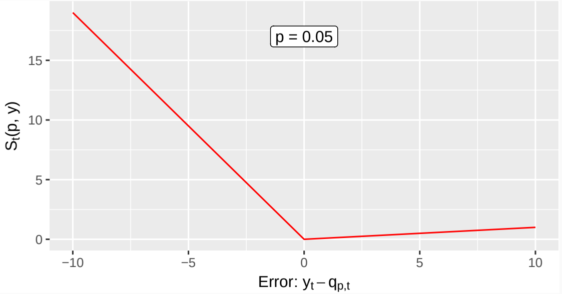
y_t = observation at time t

Quantile score

$$S_t(p, y) = \begin{cases} 2(1-p)|y_t - q_{p,t}|, & \text{if } y_t < q_{p,t} \\ 2p|y_t - q_{p,t}|, & \text{if } y_t \geq q_{p,t} \end{cases}$$

- Low S_t is good
- Multiplier of 2 often omitted, but useful for interpretation
- S_t like absolute error, weighted to account for likely exceedance
- Average $S_t(p, y)$ over p = CRPS (Continuous Rank Probability Score)





Evaluating probabilistic forecasts

Continuous Rank Probability Score (univariate forecsts)

Forecast distribution F_t and observation y_t.

$$\text{CRPS}(F_t, y_t) = \int_0^1 S_{p,t}(p, y_t)dp = \text{E}_F|Y - y_t| - \frac{1}{2}\text{E}_F|Y - Y^*|$$

- Y and Y^* are iid draws from F_t.
- Optimal when F_t is true distribution (i.e., it is a proper score)

Evaluating probabilistic forecasts

Continuous Rank Probability Score (univariate forecsts)

Forecast distribution F_t and observation y_t.

$$\text{CRPS}(F_t, y_t) = \int_0^1 S_{p,t}(p, y_t)dp = \text{E}_F|Y - y_t| - \frac{1}{2}\text{E}_F|Y - Y^*|$$

- Y and Y^* are iid draws from F_t.
- Optimal when F_t is true distribution (i.e., it is a proper score)

Energy score (multivariate forecasts)

- $\text{ES}(F_t, \boldsymbol{y}_t) = \text{E}_F||\boldsymbol{Y} - \boldsymbol{y}_t|| - \frac{1}{2}\text{E}_F||\boldsymbol{Y} - \boldsymbol{Y}^*||$

Evaluating probabilistic forecasts

Continuous Rank Probability Score (univariate forecsts)

Forecast distribution F_t and observation y_t.

$$\text{CRPS}(F_t, y_t) = \int_0^1 S_{p,t}(p, y_t)dp = \text{E}_F|Y - y_t| - \frac{1}{2}\text{E}_F|Y - Y^*|$$

- Y and Y^* are iid draws from F_t.
- Optimal when F_t is true distribution (i.e., it is a proper score)

Energy score (multivariate forecasts)

- $\text{ES}(F_t, \boldsymbol{y}_t) = \text{E}_F||\boldsymbol{Y} - \boldsymbol{y}_t|| - \frac{1}{2}\text{E}_F||\boldsymbol{Y} - \boldsymbol{Y}^*||$

Log score (multivariate forecasts)

- $\text{LS}(F_t, \boldsymbol{y}_t) = -\log f(\boldsymbol{y}_t)$

Evaluating probabilistic forecasts

Proper scoring rule

optimized when true forecast distribution is used.

Evaluating probabilistic forecasts

Proper scoring rule

optimized when true forecast distribution is used.

Scoring Rule	Coherent v Incoherent	Coherent v Coherent
Log Score	Not proper	• Ordering preserved if compared using bottom-level only
Energy Score	Proper	• Full hierarchy should be used. • Rankings may change otherwise.

Score optimal reconciliation

Algorithm proposed by Panagiotelis et al (2020) for optimizing $\boldsymbol{G}$ using stochastic gradient descent to optimize Energy Score.

1. Compute base forecasts over a test set.
2. Compute OLS reconciliation: $\boldsymbol{G} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$
3. Iteratively update $\boldsymbol{G}$ using SGD with Adam method and ES objective over a test set

Outline

Example: Australian tourism

```
tourism
```

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

Example: Australian tourism

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

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

Example: Australian tourism

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

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

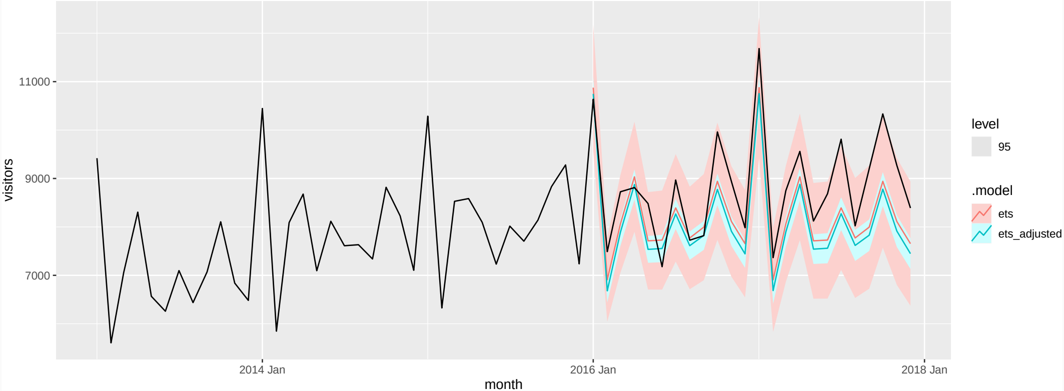
Example: Australian tourism

```r
fc <- fit %>%
  reconcile(ets_adjusted = min_trace(ets)) %>%
  forecast(h = "2 years")
```

```
## # A fable: 5,280 x 7 [1M]
## # Key:     state, zone, region, .model [220]
##    state zone         region       .model    month         visitors .mean
##    <chr> <chr>        <chr>        <chr>     <mth>           <dist> <dbl>
##  1 NSW   <aggregated> <aggregated> ets    2016 Jan N(3679, 71136) 3679.
##  2 NSW   <aggregated> <aggregated> ets    2016 Feb N(2241, 27912) 2241.
##  3 NSW   <aggregated> <aggregated> ets    2016 Mar N(2602, 37643) 2602.
##  4 NSW   <aggregated> <aggregated> ets    2016 Apr N(3027, 50976) 3027.
##  5 NSW   <aggregated> <aggregated> ets    2016 May N(2504, 36795) 2504.
##  6 NSW   <aggregated> <aggregated> ets    2016 Jun N(2447, 36005) 2447.
##  7 NSW   <aggregated> <aggregated> ets    2016 Jul N(2734, 44488) 2734.
##  8 NSW   <aggregated> <aggregated> ets    2016 Aug N(2496, 38775) 2496.
```

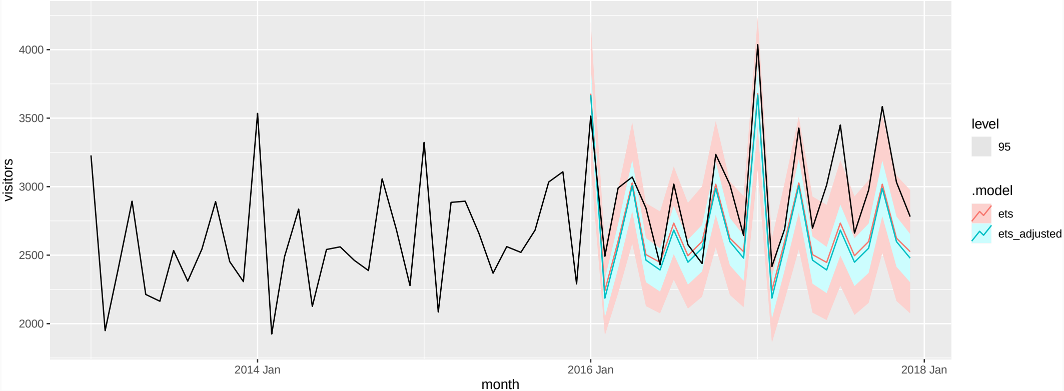
Example: Australian tourism

```r
fc %>%
  filter(is_aggregated(state)) %>%
  autoplot(filter(tourism_agg, year(month) > 2012), level = 95)
```



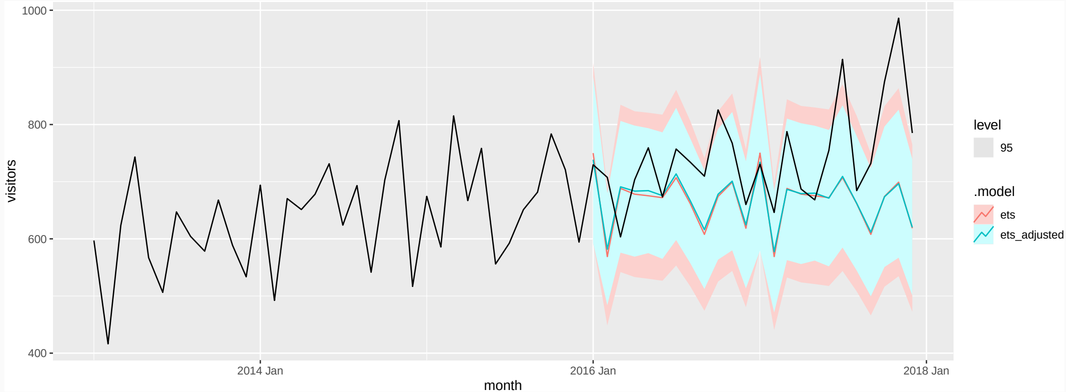
Example: Australian tourism

```
fc %>%
  filter(state == "NSW" & is_aggregated(zone)) %>%
  autoplot(filter(tourism_agg, year(month) > 2012), level = 95)
```



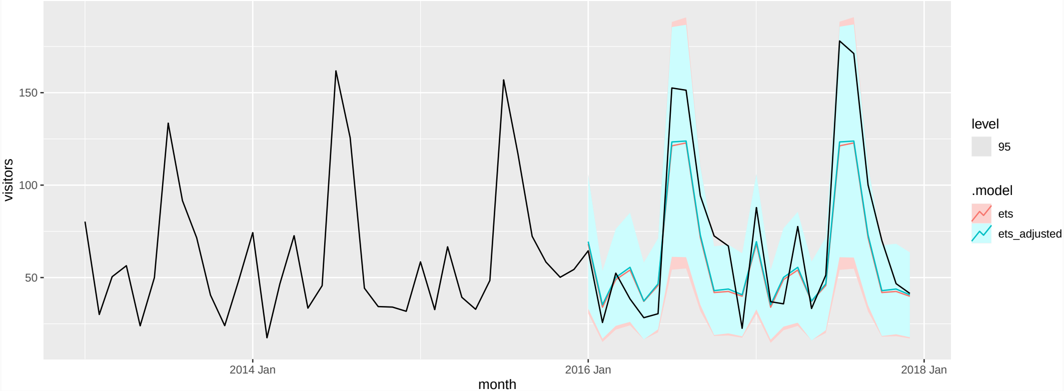
Example: Australian tourism

```
fc %>%
  filter(region == "Melbourne") %>%
  autoplot(filter(tourism_agg, year(month) > 2012), level = 95)
```



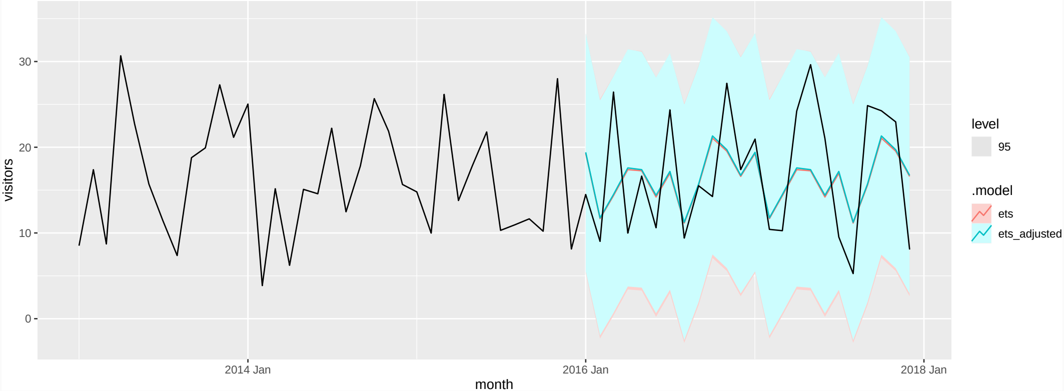
Example: Australian tourism

```
fc %>%
  filter(region == "Snowy Mountains") %>%
  autoplot(filter(tourism_agg, year(month) > 2012), level = 95)
```



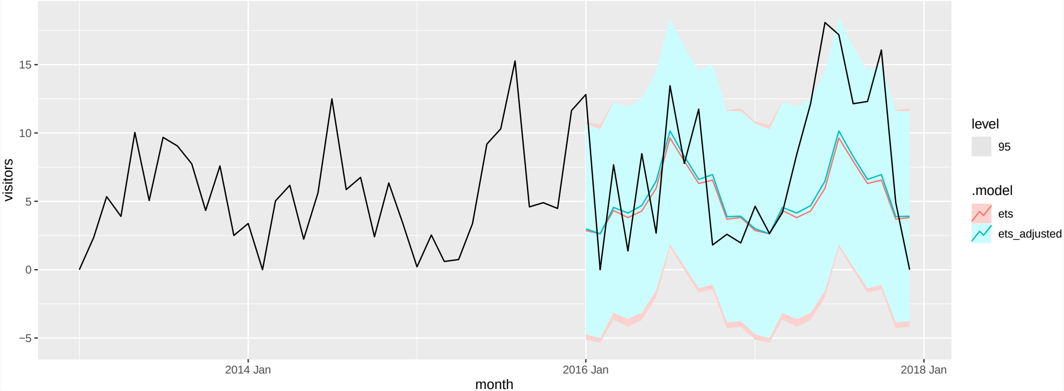
Example: Australian tourism

```r
fc %>%
  filter(region == "Barossa") %>%
  autoplot(filter(tourism_agg, year(month) > 2012), level = 95)
```



Example: Australian tourism

```
fc %>%
  filter(region == "MacDonnell") %>%
  autoplot(filter(tourism_agg, year(month) > 2012), level = 95)
```



Example: Australian tourism

```
fc <- tourism_agg %>%
  filter(year(month) <= 2015) %>%
  model(
    ets = ETS(visitors),
    arima = ARIMA(visitors)
  ) %>%
  mutate(
    comb = (ets + arima) / 2
  ) %>%
  reconcile(
    ets_adj = min_trace(ets),
    arima_adj = min_trace(arima),
    comb_adj = min_trace(comb)
  ) %>%
  forecast(h = "2 years")
```

Example: Australian tourism

```r
fc %>%
  accuracy(data = tourism_agg,
           measures = list(crps = CRPS, ss=skill_score(CRPS)))
```

```
## # A tibble: 660 x 7
##    .model state zone             region        .type  crps      ss
##    <chr>  <chr> <chr>            <chr>         <chr> <dbl>   <dbl>
##  1 arima  NSW   <aggregated>     <aggregated>  Test  158.   0.277
##  2 arima  NSW   Metro NSW        <aggregated>  Test   69.1  0.152
##  3 arima  NSW   North Coast NSW <aggregated>  Test   58.5  0.0577
##  4 arima  NSW   South Coast NSW <aggregated>  Test   24.2  0.147
##  5 arima  NSW   South NSW        <aggregated>  Test   25.4  0.277
##  6 arima  NSW   North NSW        <aggregated>  Test   57.0  0.0321
##  7 arima  NSW   ACT              <aggregated>  Test   34.5 -0.221
##  8 arima  NSW   Metro NSW        Sydney        Test   62.4  0.139
##  9 arima  NSW   Metro NSW        Central Coast Test   13.9  0.196
```

Example: Australian tourism

```
fc %>%
  accuracy(tourism_agg,
           measures = list(crps = CRPS, ss=skill_score(CRPS))) %>%
  group_by(.model) %>%
  summarise(sspc = mean(ss) * 100) %>%
  arrange(sspc)
```

```
## # A tibble: 6 x 2
##   .model      sspc
##   <chr>      <dbl>
## 1 arima_adj   11.9
## 2 arima       12.0
## 3 comb_adj    17.0
## 4 ets_adj     17.7
## 5 comb        18.2
## 6 ets         19.1
```

Outline

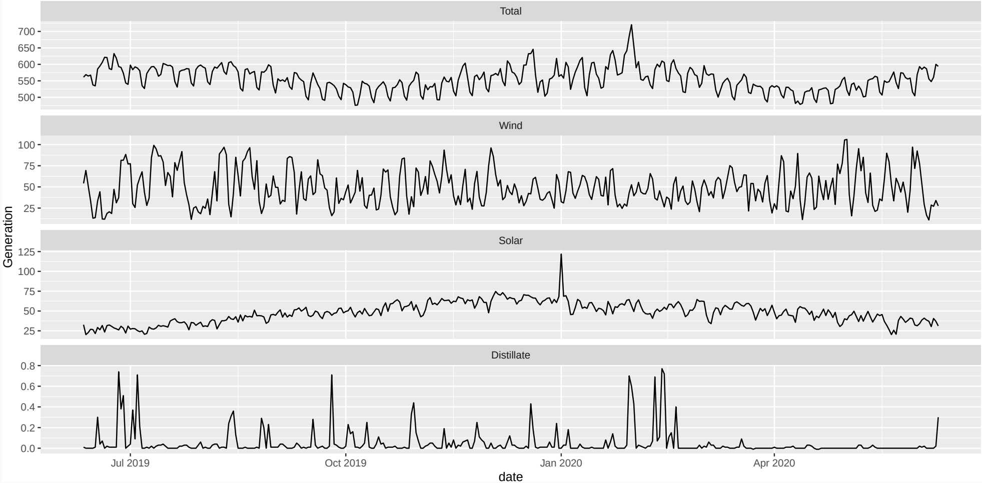
Example: Australian electricity generation

Daily time series from `opennem.org.au`

1. Total = Renewable + Non-renewable
2. Renewable = Batteries + Hydro + Solar + Wind + Biomass
   Non-Renewable = Coal + Gas + Distillate
3. Battery = Battery (Discharging) + Battery (Charging)
   Solar = Solar (Rooftop) + Solar (Utility)
   Coal = Black Coal + Brown Coal
   Gas = Gas (OCGT) + Gas (CCGT) + Gas (Steam) + Gas (Recip)

n = 23 series; m = 15 bottom-level series.

Example: Australian electricity generation

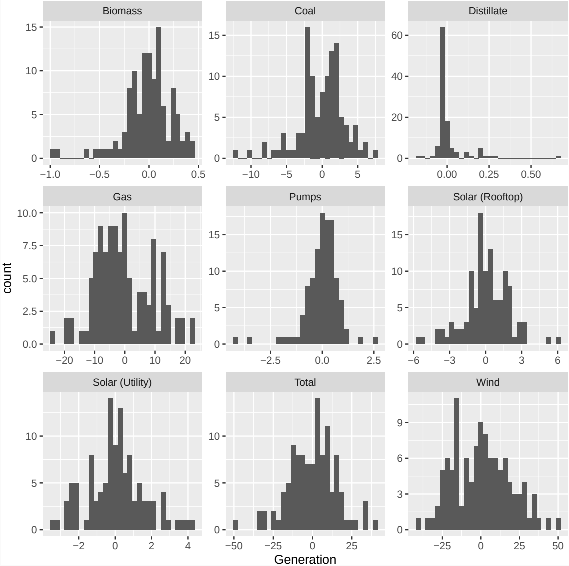


Example: Australian electricity generation

Forecast evaluation

- Rolling window of 140 days training data, and one-step-forecasts for 170 days test data.
- One-layer feed-forward neural network with up to 28 lags of target variable as inputs.
- Implemented using `NNETAR()` function in `fable` package.
- Model could be improved with temperature predictor.

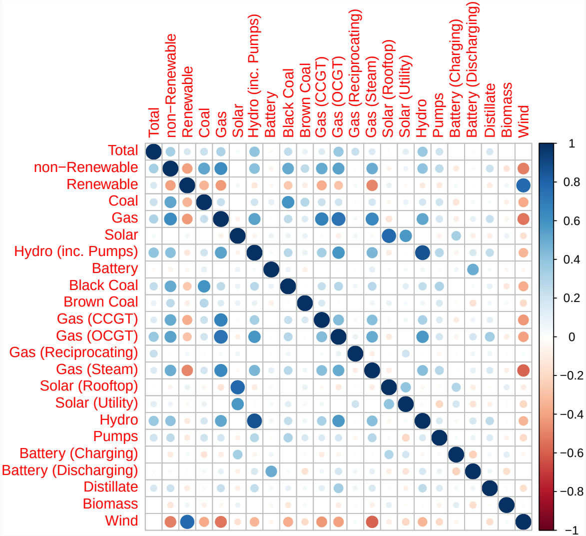
Example: Australian electricity generation



Histogram of residuals:
2 Oct 2019 – 21 Jan 2020

Clearly non-Gaussian

Example: Australian electricity generation



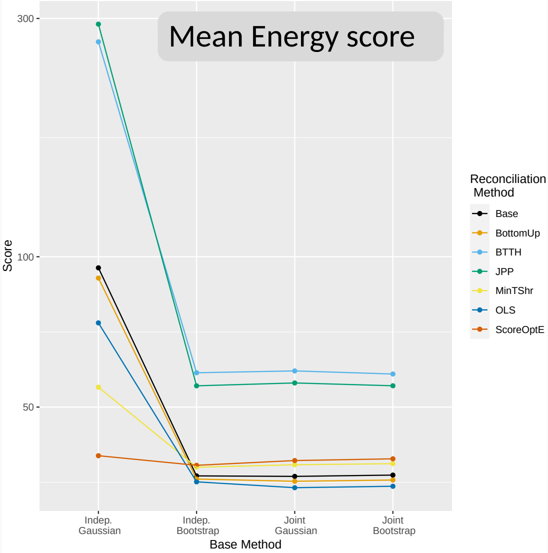
Correlations of residuals:
2 Oct 2019 – 21 Jan 2020

Blue = positive correlation.

Red = negative correlation.

Large = stronger correlations.

Example: Australian electricity generation



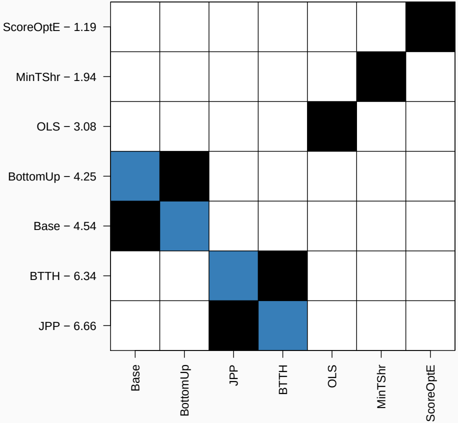
Base residual assumptions

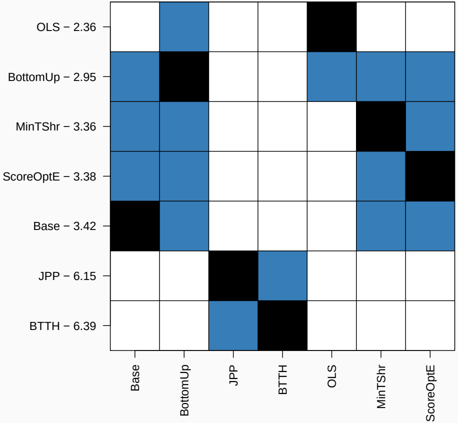
- Gaussian independent
- Gaussian dependent
- Non-Gaussian independent
- Non-Gaussian dependent

Reconciliation methods

- Base
- BottomUp
- BTTH: Ben Taieb, Taylor, Hyndman
- JPP: Jeon, Panagiotelis, Petropoulos
- OLS
- MinT(Shrink)
- Score Optimal Reconciliation

Example: Australian electricity generation





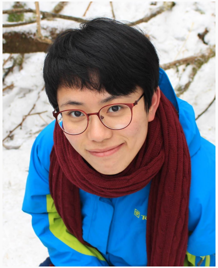
Nemenyi test for different scores

Base forecasts are independent and Gaussian.

Nemenyi test for different scores

Base forecasts are obtained by jointly bootstrapping residuals.

Thanks



More information

- Slides and papers: **robjhyndman.com**
- Packages: **tidyverts.org**
- Forecasting textbook using fable package: **OTexts.com/fpp3**

Find me at ...

@robjhyndman

@robjhyndman

robjhyndman.com

rob.hyndman@monash.edu