

Improving forecasts via subspace projections

Rob J Hyndman



Outline

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

Outline

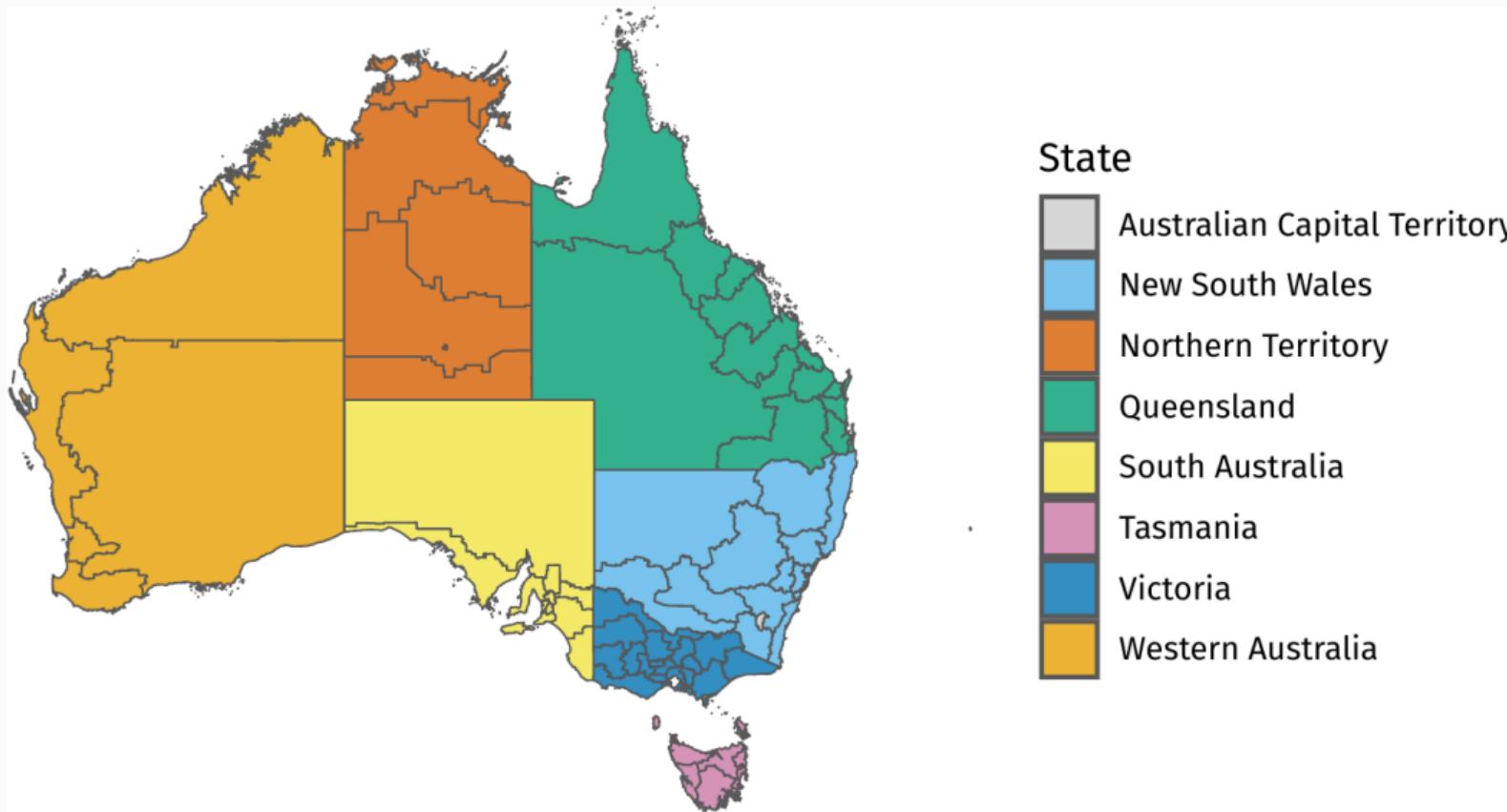
1 Improving hierarchical forecasts

2 Improving univariate forecasts

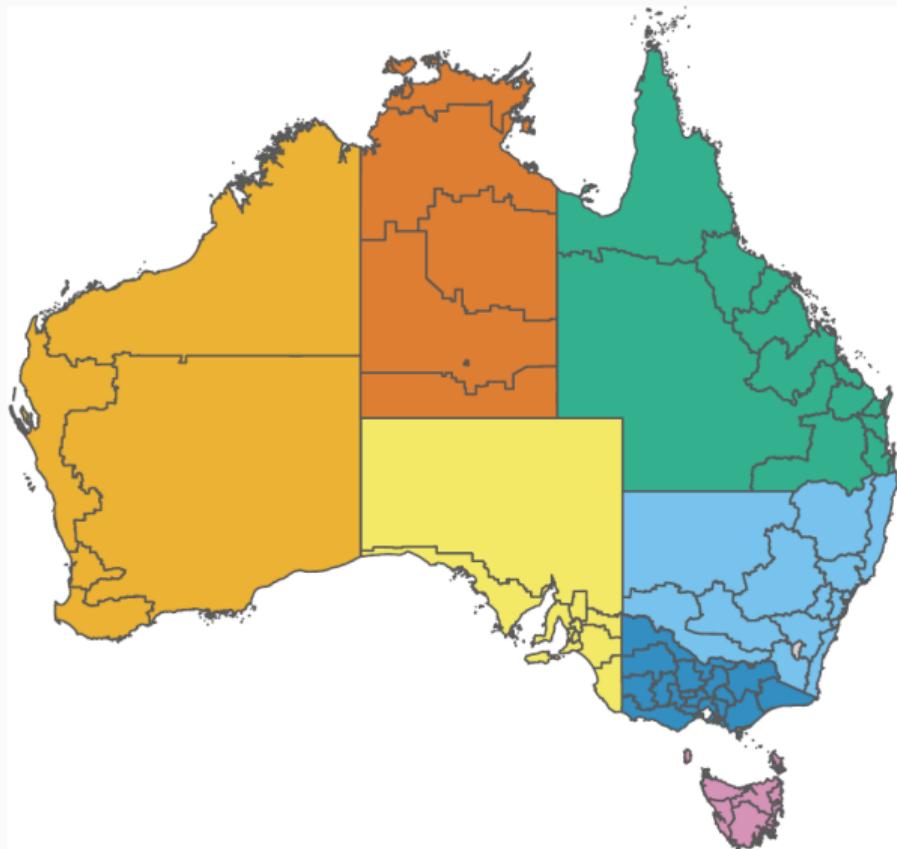
3 Improving multivariate forecasts

4 Final comments

Australian tourism regions



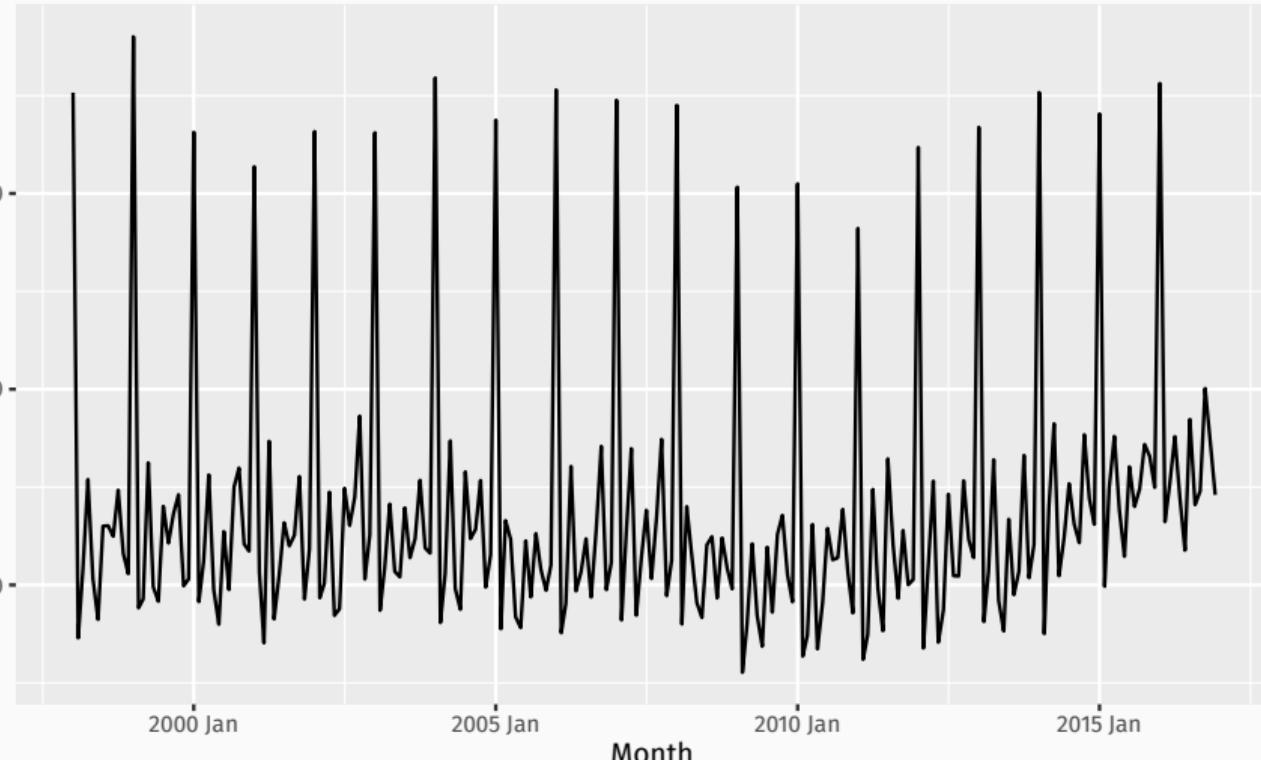
Australian tourism regions



- 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

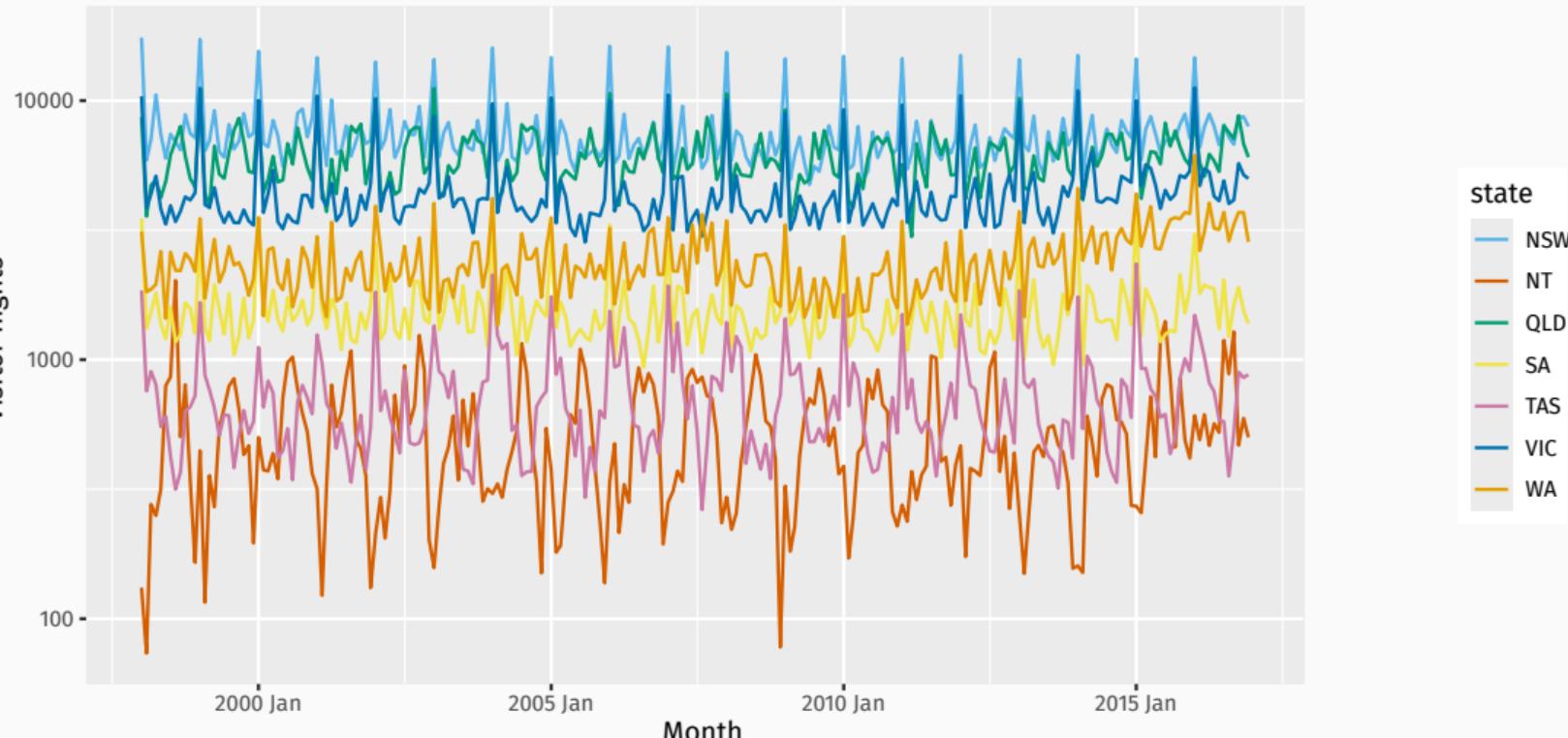
Australian tourism data

Total domestic travel: Australia



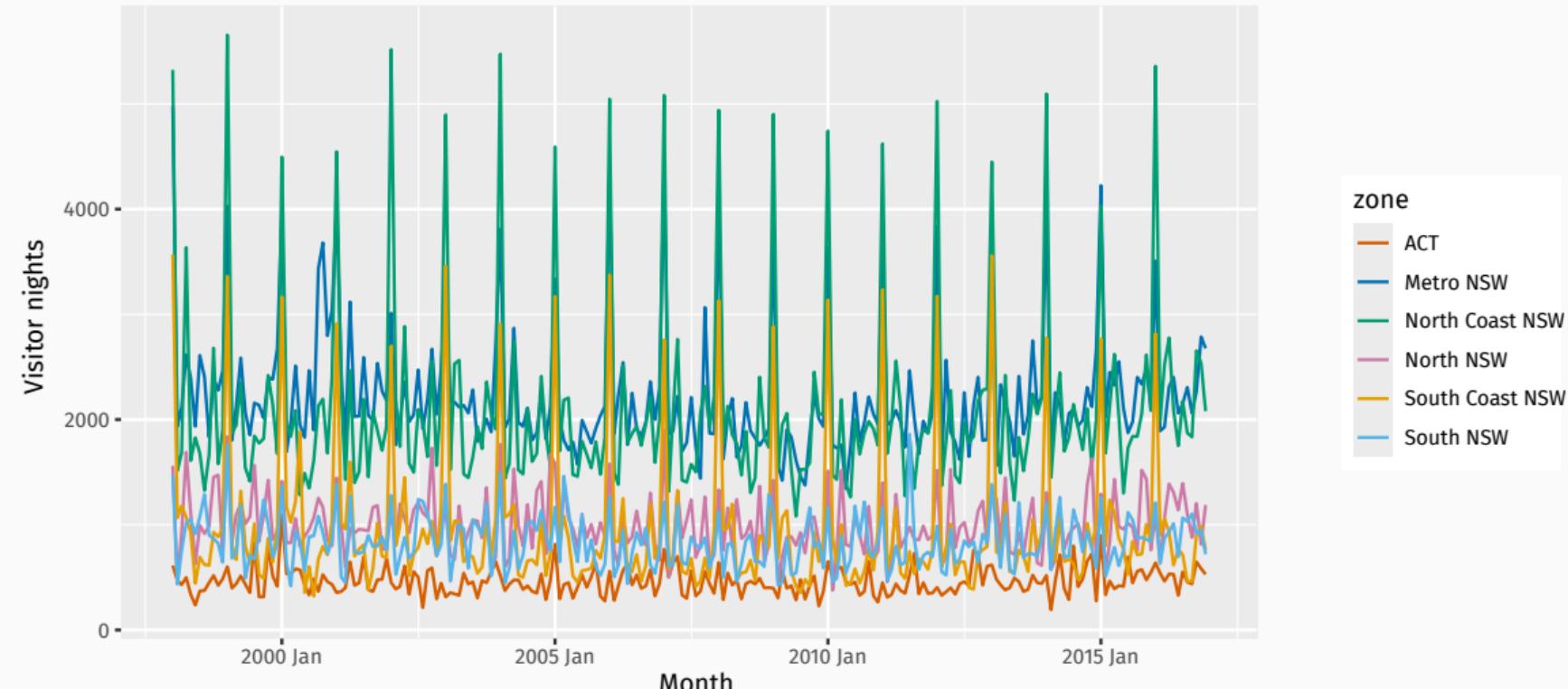
Australian tourism data

Total domestic travel: by state



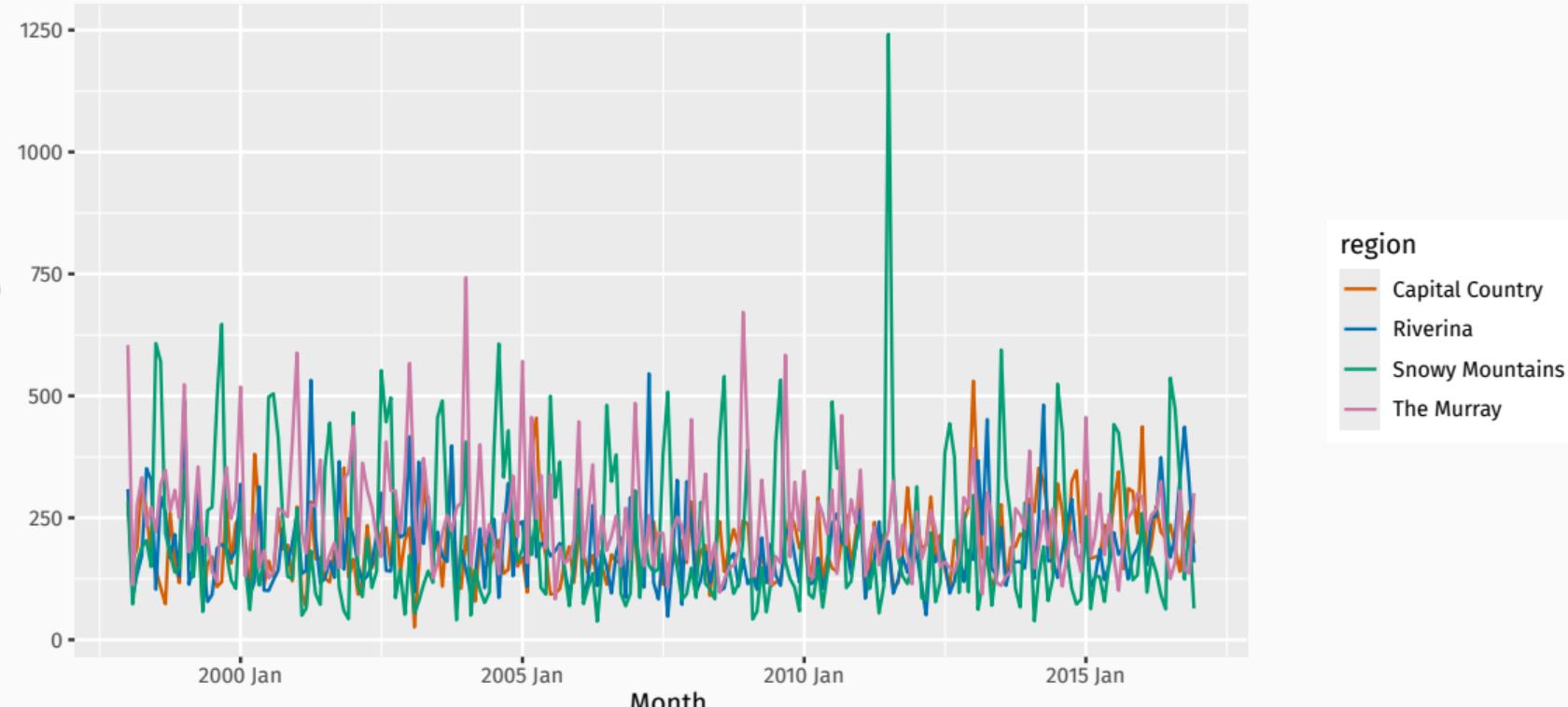
Australian tourism data

Total domestic travel: NSW by zone



Australian tourism data

Total domestic travel: South NSW by region



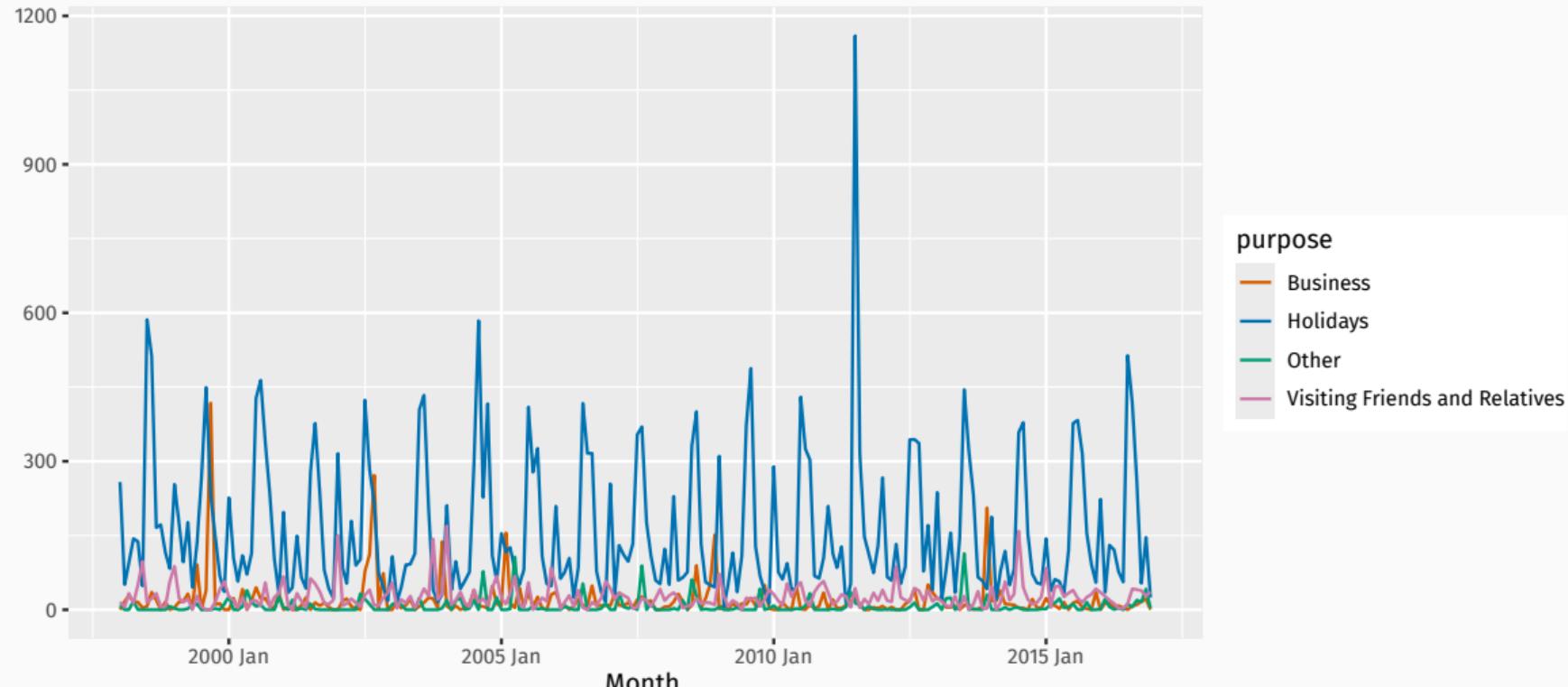
Australian tourism data

Total domestic travel: by purpose of travel



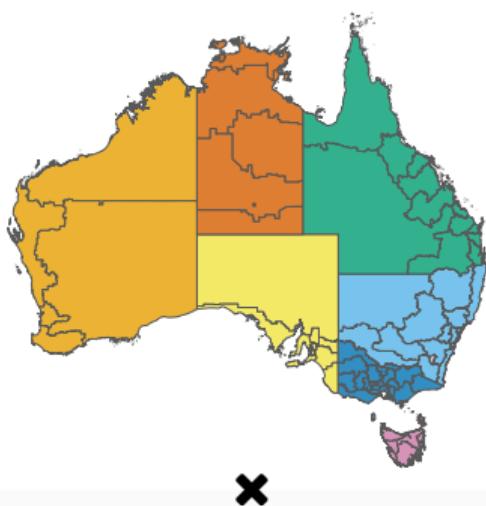
Australian tourism data

Total domestic travel: Snowy Mountains by purpose of travel



Australian tourism data

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

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

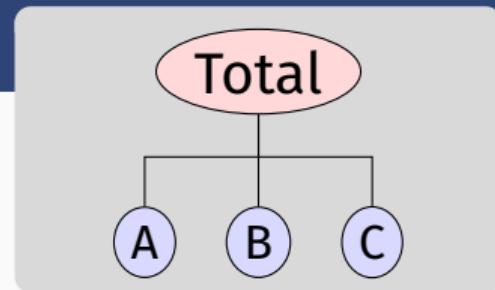
- Need forecasts at all levels of aggregation.
- Independent forecasts will not add up.
- Impose constraints on the forecasts to ensure they are "coherent".

Hierarchical and grouped time series

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

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

- \mathbf{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 .
- \mathbf{b}_t = vector of most disaggregated series at time t
- \mathbf{S} = “summing matrix” containing the linear constraints.



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

The coherent subspace

Coherent subspace

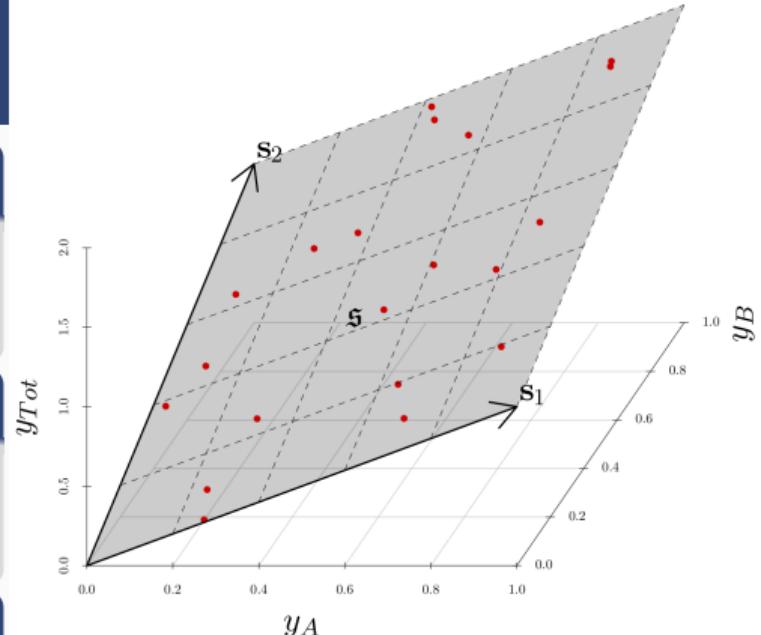
n_b -dimensional linear subspace $\mathfrak{s} \subset \chi^n$ for which linear constraints hold for all $\mathbf{y} \in \mathfrak{s}$.

Hierarchical time series

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

Coherent point forecasts

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



$$y_{Tot} = y_A + y_B$$

The coherent subspace

Coherent subspace

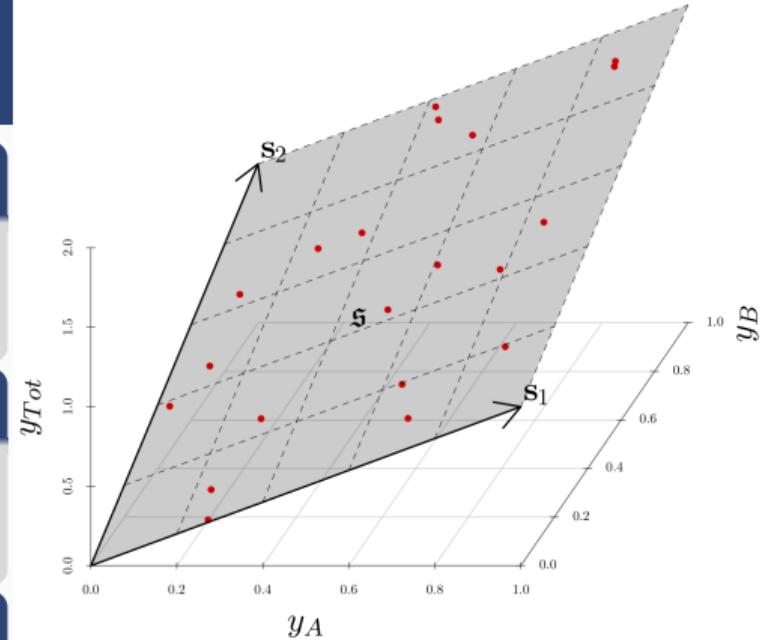
n_b -dimensional linear subspace $\mathfrak{s} \subset \chi^n$ for which linear constraints hold for all $\mathbf{y} \in \mathfrak{s}$.

Hierarchical time series

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

Coherent point forecasts

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



$$y_{Tot} = y_A + y_B$$

Base forecasts

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

The coherent subspace

Coherent subspace

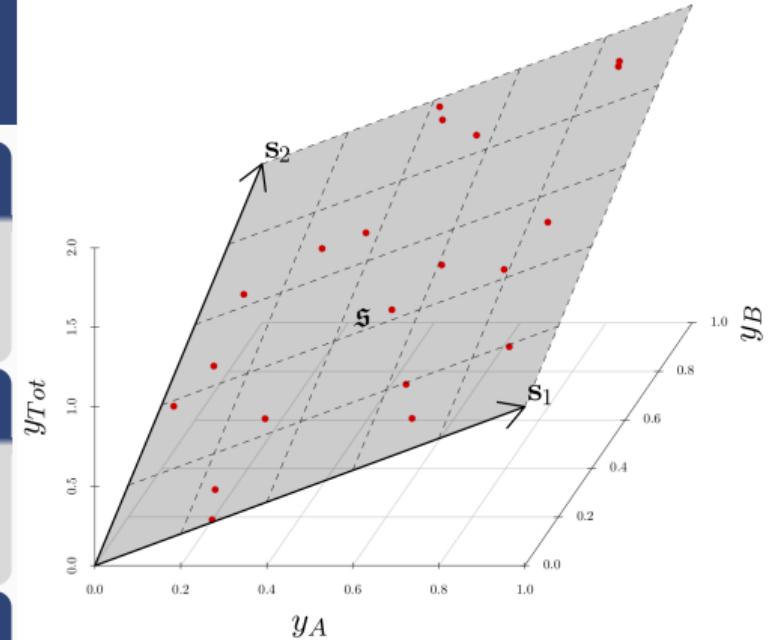
n_b -dimensional linear subspace $\mathfrak{s} \subset \chi^n$ for which linear constraints hold for all $\mathbf{y} \in \mathfrak{s}$.

Hierarchical time series

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

Coherent point forecasts

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



$$y_{Tot} = y_A + y_B$$

Base forecasts

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

Reconciled forecasts

Let \mathbf{M} be a projection matrix. $\tilde{\mathbf{y}}_{t+h|t} = \mathbf{M}\hat{\mathbf{y}}_{t+h|t}$ “reconciles” $\hat{\mathbf{y}}_{t+h|t}$.

Linear projection reconciliation

$$\tilde{\mathbf{y}}_{t+h|t} = \mathbf{M}\hat{\mathbf{y}}_{t+h|t}$$

- If \mathbf{S} forms a basis set for \mathfrak{s} , then projections are of the form $\mathbf{M} = \mathbf{S}(\mathbf{S}'\Psi\mathbf{S})^{-1}\mathbf{S}'\Psi$ where Ψ is a positive definite matrix.
- Coherent base forecasts are unchanged since $\mathbf{M}\hat{\mathbf{y}} = \hat{\mathbf{y}}$
- If $\hat{\mathbf{y}}$ is unbiased, then $\tilde{\mathbf{y}}$ is also unbiased.
- $\mathbf{W}_h = \text{Var}[\mathbf{y}_{T+h} - \hat{\mathbf{y}}_{T+h|T} \mid \mathbf{y}_1, \dots, \mathbf{y}_T]$ is the covariance matrix of the base forecast errors.
- $\mathbf{V}_h = \text{Var}[\mathbf{y}_{T+h} - \tilde{\mathbf{y}}_{T+h|T} \mid \mathbf{y}_1, \dots, \mathbf{y}_T] = \mathbf{M}\mathbf{W}_h\mathbf{M}'$ is the covariance matrix of the reconciled forecast errors.
- How to choose the best Ψ ?

Minimum trace (MinT) reconciliation

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

$$\mathbf{M} = \mathbf{S}(\mathbf{S}'\mathbf{W}_h^{-1}\mathbf{S})^{-1}\mathbf{S}'\mathbf{W}_h^{-1}$$

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

Reconciled forecasts

Base forecasts

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

Reconciliation method M

OLS $\mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'$

WLS(var) $\mathbf{S}(\mathbf{S}'\Lambda_v\mathbf{S})^{-1}\mathbf{S}'\Lambda_v$

WLS(struct) $\mathbf{S}(\mathbf{S}'\Lambda_s\mathbf{S})^{-1}\mathbf{S}'\Lambda_s$

MinT(sample) $\mathbf{S}(\mathbf{S}'\hat{\mathbf{W}}_{\text{sam}}^{-1}\mathbf{S})^{-1}\mathbf{S}'\hat{\mathbf{W}}_{\text{sam}}^{-1}$

MinT(shrink) $(\mathbf{S}'\hat{\mathbf{W}}_{\text{shr}}^{-1}\mathbf{S})^{-1}\mathbf{S}'\hat{\mathbf{W}}_{\text{shr}}^{-1}$

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

- $\Lambda_v = \text{diag}(\mathbf{W}_1)^{-1}$
- $\Lambda_s = \text{diag}(\mathbf{S}\mathbf{1})^{-1}$
- $\hat{\mathbf{W}}_{\text{sam}}$ is sample estimate of the residual covariance matrix
- $\hat{\mathbf{W}}_{\text{shr}}$ is shrinkage estimator $\tau \text{diag}(\hat{\mathbf{W}}_{\text{sam}}) + (1 - \tau)\hat{\mathbf{W}}_{\text{sam}}$
where τ selected optimally.

Example: Australian tourism

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

```
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

```
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)>
	# i 101 more rows			

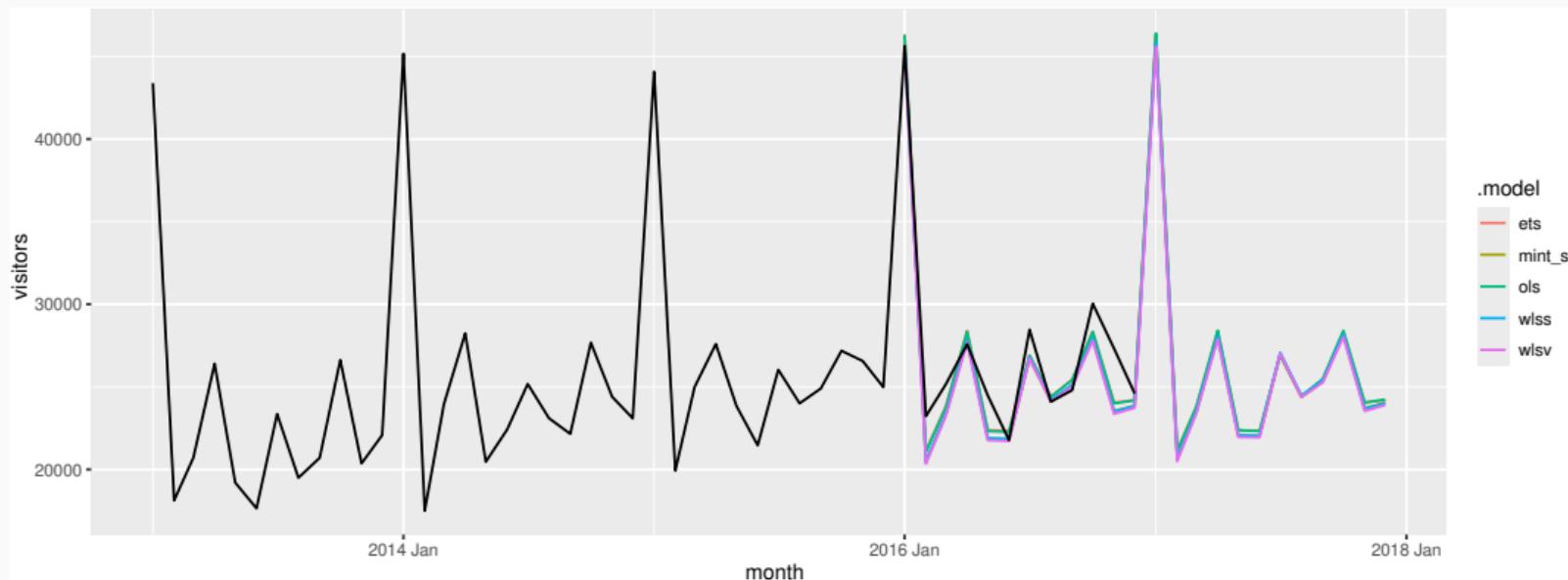
Example: Australian tourism

```
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.
5 NSW     ACT     Canberra  ets     2016 May N(430, 11509)  430.
```

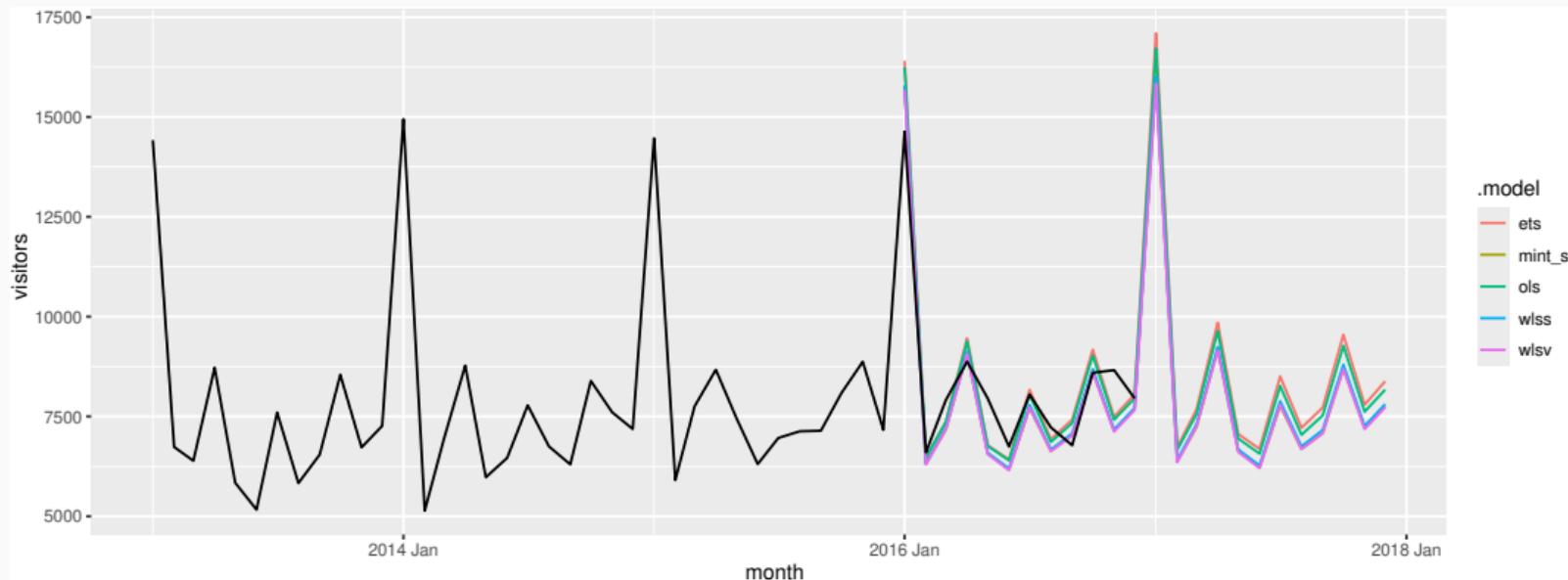
Example: Australian tourism

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



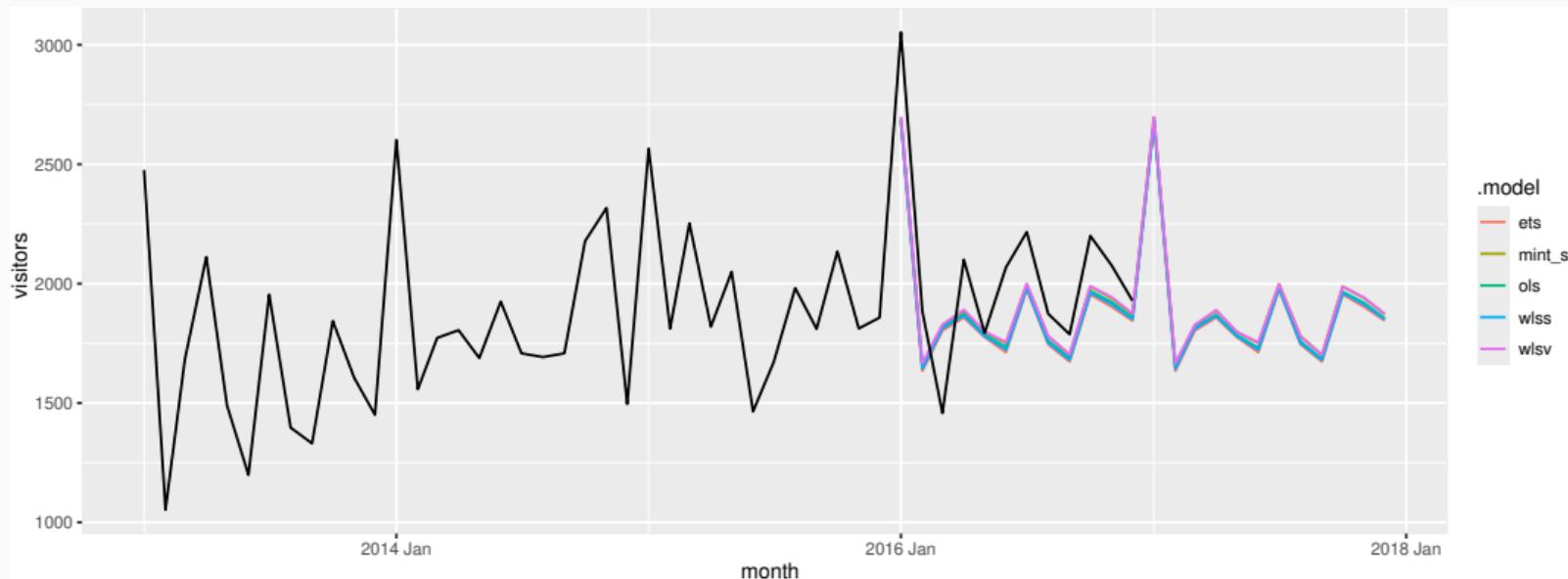
Example: Australian tourism

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



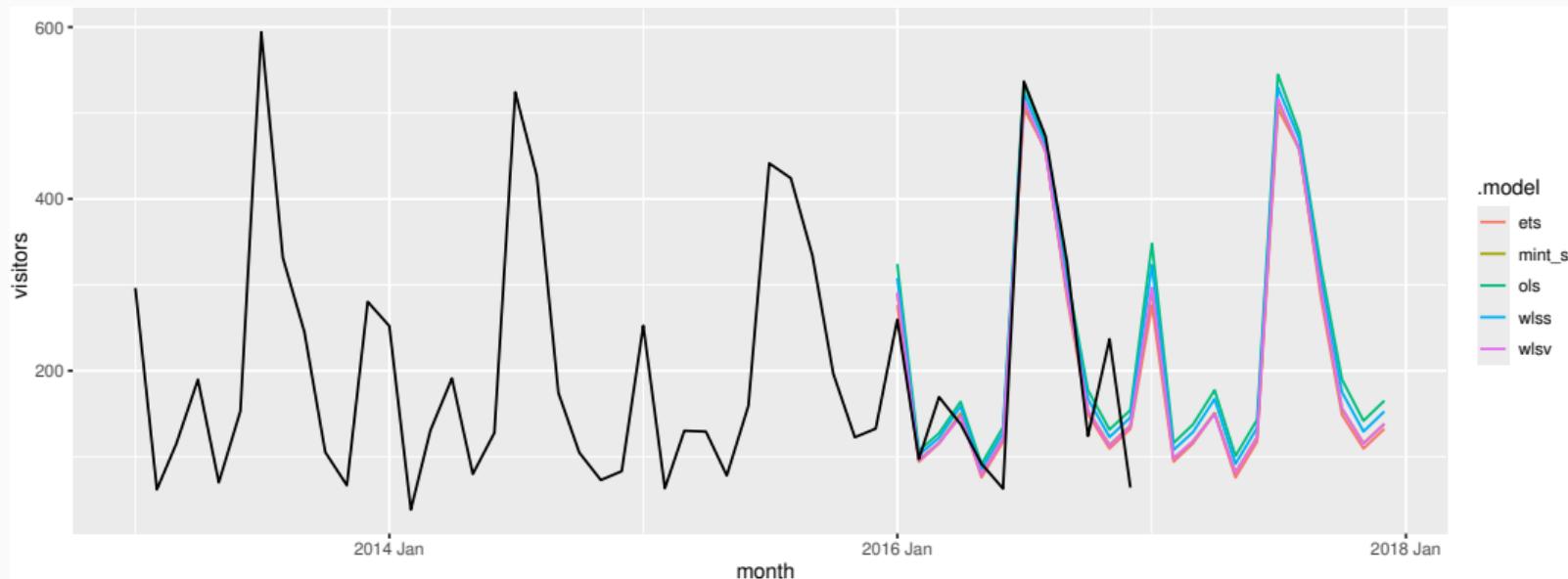
Example: Australian tourism

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



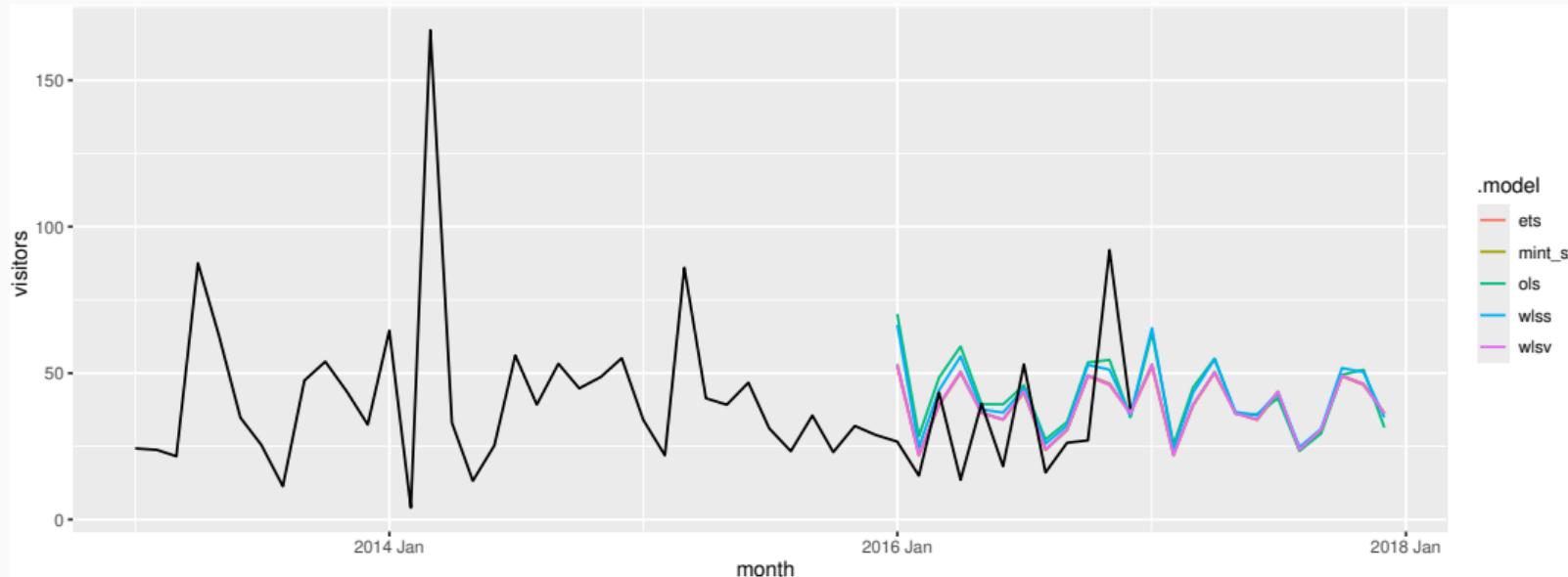
Example: Australian tourism

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



Example: Australian tourism

```
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

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

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

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

Example: Australian tourism

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

```
# A tibble: 555 x 7  
#> # ... with 545 more rows  
#> # ... with 7 variables:  
#> #   .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
```

Example: Australian tourism

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

```
# A tibble: 5 x 3
  .model    mase   rmsse
  <chr>    <dbl>  <dbl>
1 ols      0.890  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

```
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  0.925
```

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  0.925
```

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

Distance reducing property

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

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

- 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.

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

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

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

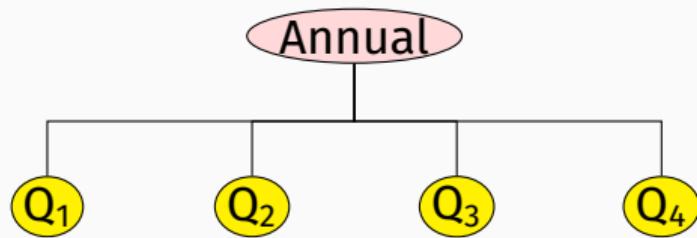
Using sums of variances:

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

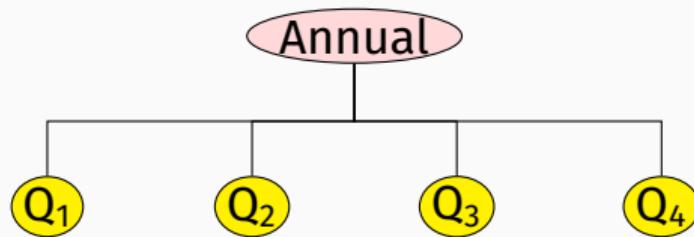
Outline

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

Temporal reconciliation: quarterly data

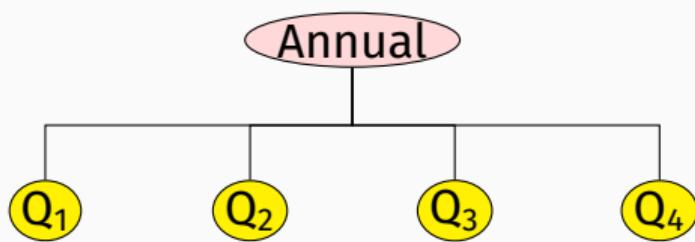


Temporal reconciliation: quarterly data



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

Temporal reconciliation: quarterly data

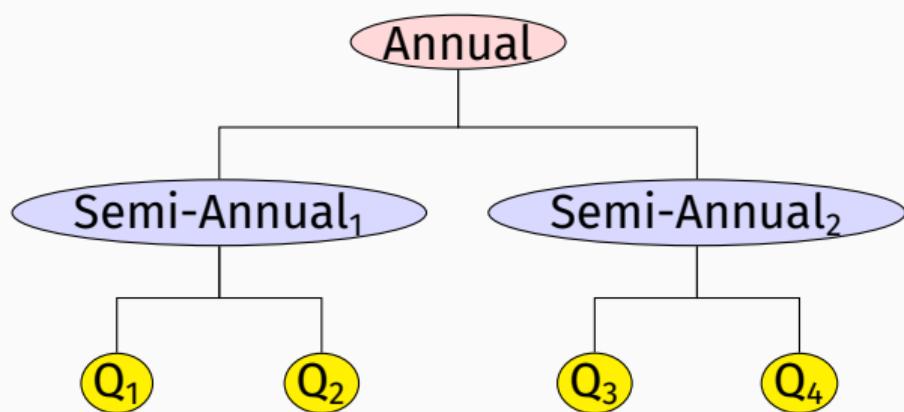


$$\mathbf{y}_\tau = \begin{bmatrix} x_\tau^{[4]} \\ x_{\tau,1}^{[1]} \\ x_{\tau,2}^{[1]} \\ x_{\tau,3}^{[1]} \\ x_{\tau,4}^{[1]} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

τ = index of largest temporal aggregation level.

Temporal reconciliation: quarterly data

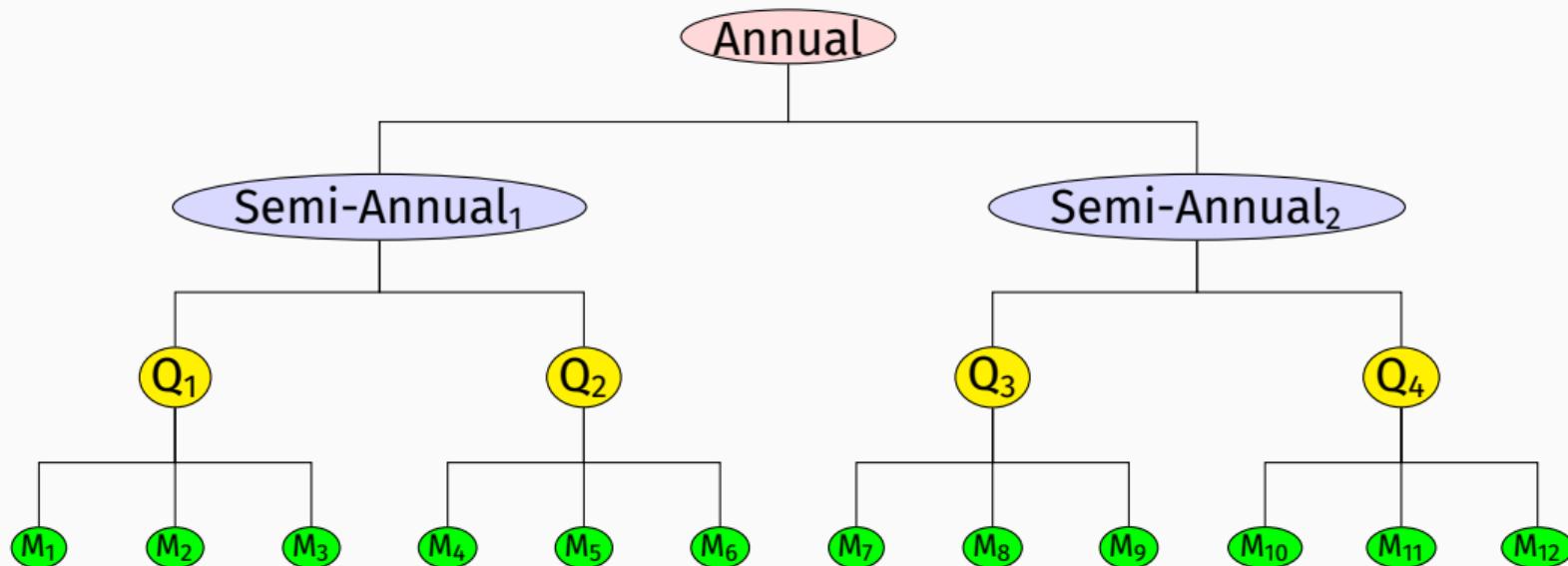


- Forecast series at each available frequency.
- Optimally combine forecasts within the same year.

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

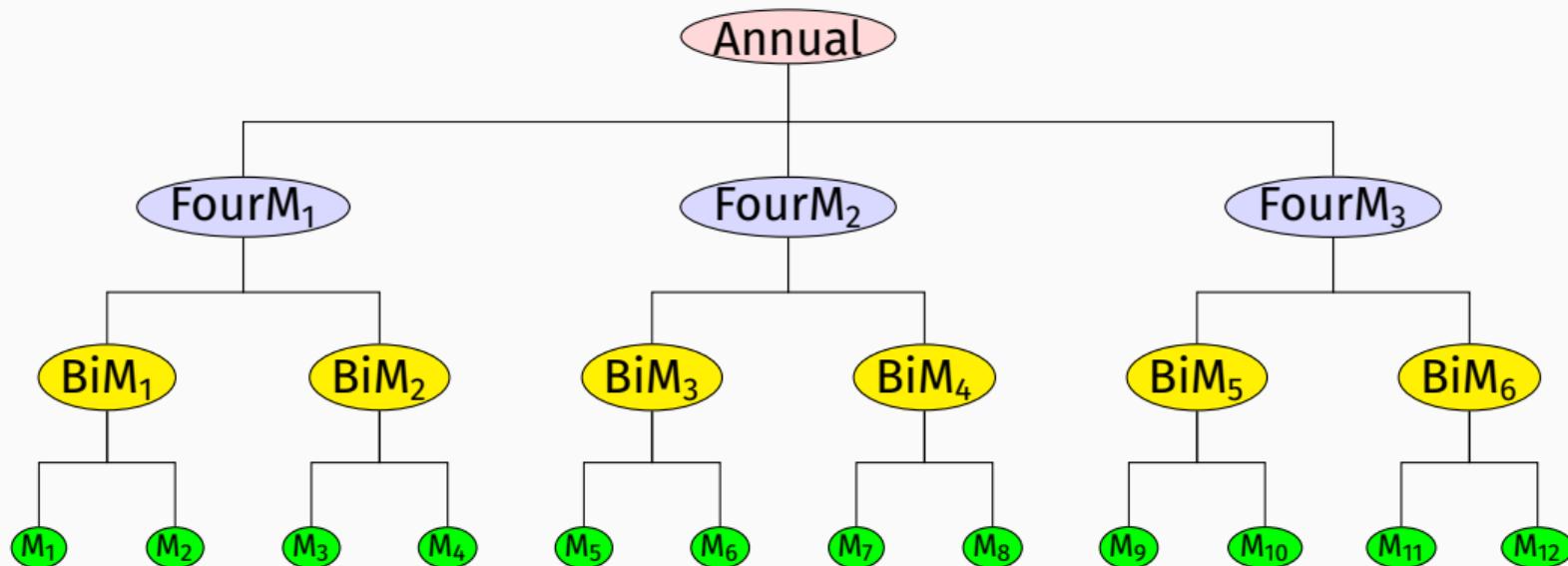
τ = index of largest temporal aggregation level.

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

$$\mathbf{y}_\tau = \begin{bmatrix} \mathbf{x}_\tau^{[12]} \\ \mathbf{x}_\tau^{[6]} \\ \mathbf{x}_\tau^{[4]} \\ \mathbf{x}_\tau^{[3]} \\ \mathbf{x}_\tau^{[2]} \\ \mathbf{x}_\tau^{[1]} \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \mathbf{I}_{12}$$

Temporal reconciliation

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

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

- $k \in \mathcal{K} = \{k_1, \dots, k_p\}$ 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

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

where

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

τ is time index for most aggregated series,

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

Example: Accident & emergency services demand

Weekly A&E demand data: 7 November 2010 to 7 June 2015.

Type 1 Departments – Major A&E

Type 2 Departments – Single Specialty

Type 3 Departments – Other A&E/Minor Injury Unit

Total Attendances

Type 1 Departments – Major A&E > 2 hours

Type 2 Departments – Single Specialty > 2 hours

Type 3 Departments – Other A&E/Minor Injury Unit > 2 hours

Total Attendances > 2 hours

Emergency Admissions via Type 1 A&E

Total Emergency Admissions via A&E

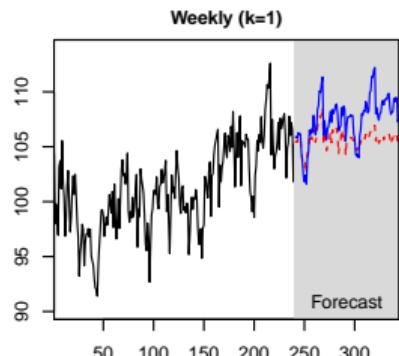
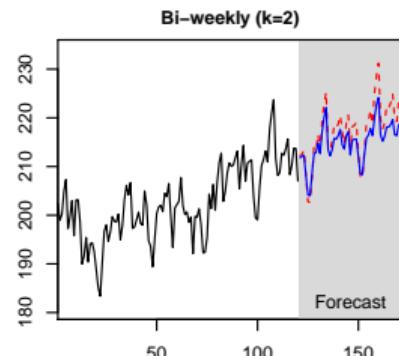
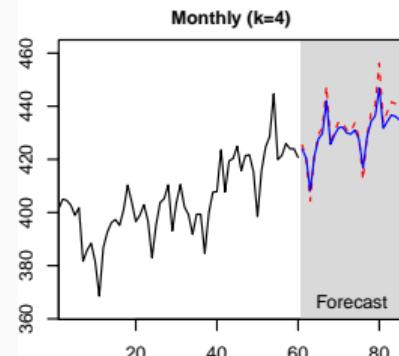
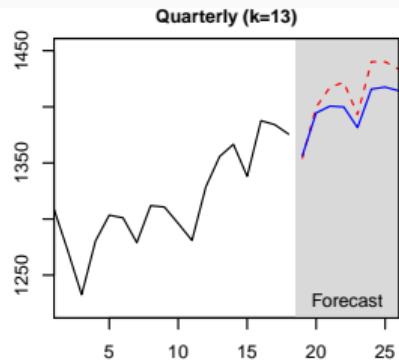
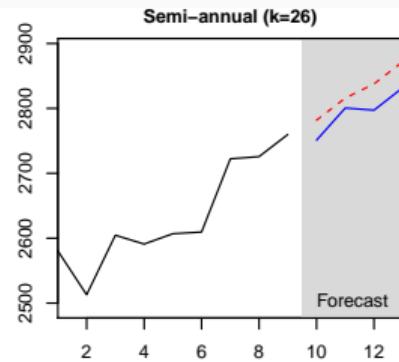
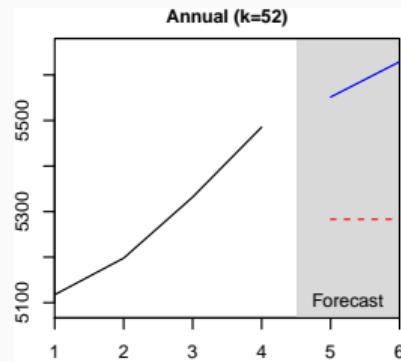
Other Emergency Admissions (i.e not via A&E)

Total Emergency Admissions

Number of patients spending > 2 hours from decision to admit to admission

Example: Accident & emergency services demand

Total emergency admissions via A&E



Example: Accident & emergency services demand

Test set: last 52 weeks

MASE comparison (ARIMA models)

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

Temporal reconciliation: M3 monthly series

- Apply temporal reconciliation to all 1428 monthly series from M3 competition
- Forecast horizon $h = 18$ months
- ETS and ARIMA models
- Measure percentage difference to base forecasts
- Reconciliation methods:
 - ▶ WLS_H (diagonal)
 - ▶ WLS_V (diagonal with common variances for same frequency)
 - ▶ WLS_S (diagonal/structural)

Temporal reconciliation: M3 monthly series

Improvement in MASE

Aggregation		ETS					ARIMA				
level	h	Base	BU	WLS_H	WLS_V	WLS_S	Base	BU	WLS_H	WLS_V	WLS_S
Annual	1	1.1	-12.1	-17.9	-17.8	-18.5	1.3	-25.4	-29.9	-29.9	-30.2
Semi-annual	3	1.0	0.0	-6.3	-6.0	-6.9	1.1	-2.9	-8.1	-8.2	-9.4
Four-monthly	4	0.9	3.1	-3.2	-3.0	-3.4	0.9	-1.8	-6.2	-6.5	-7.1
Quarterly	6	0.9	3.2	-2.8	-2.7	-3.4	1.0	-2.6	-6.9	-7.4	-8.1
Bi-monthly	9	0.9	2.7	-2.9	-3.0	-3.7	0.9	-1.3	-5.0	-5.5	-6.3
Monthly	18	0.9	0.0	-3.7	-4.6	-5.0	0.9	0.0	-1.9	-3.2	-3.7
Average			-0.5	-6.1	-6.2	-6.8		-5.7	-9.7	-10.1	-10.8

Outline

1 Improving hierarchical forecasts

2 Improving univariate forecasts

3 Improving multivariate forecasts

4 Final comments

Outline

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

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](https://github.com/robjhyndman)

 rob.hyndman@monash.edu

-  Athanasopoulos, G, RJ Hyndman, N Kourentzes, and A Panagiotelis (2024). "Forecast reconciliation: a review". In: *International J Forecasting* 40.2, pp. 430–456.
-  Athanasopoulos, G, RJ Hyndman, N Kourentzes, and F Petropoulos (2017). "Forecasting with temporal hierarchies". In: *European J Operational Research* 262.1, pp. 60–74.
-  Di Fonzo, T and D Girolimetto (2023). "Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives". In: *International J Forecasting* 39.1, pp. 39–57.
-  Girolimetto, D, G Athanasopoulos, T Di Fonzo, and RJ Hyndman (2024). "Cross-temporal probabilistic forecast reconciliation". In: *International J Forecasting* 40.3, pp. 1134–1151.

-  Hyndman, RJ, RA Ahmed, G Athanasopoulos, and HL Shang (2011). “Optimal combination forecasts for hierarchical time series”. In: *Computational Statistics & Data Analysis* 55.9, pp. 2579–2589.
-  Panagiotelis, A, P Gamakumara, G Athanasopoulos, and RJ Hyndman (2021). “Forecast reconciliation: A geometric view with new insights on bias correction”. In: *International J Forecasting* 37.1, pp. 343–359.
-  — (2023). “Probabilistic forecast reconciliation: properties, evaluation and score optimisation”. In: *European J Operational Research* 306.2, pp. 693–706.
-  Wickramasuriya, SL (2021). “Properties of point forecast reconciliation approaches”. In: *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”. In: *J American Statistical Association* 114.526, pp. 804–819.