# Improving forecasts via subspace projections

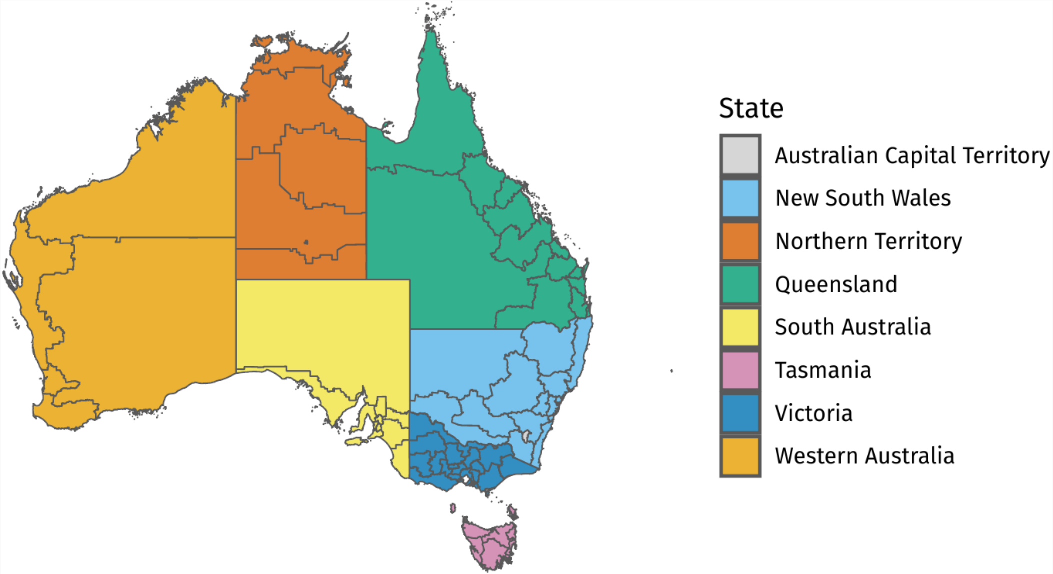
**Rob J Hyndman**

Photo by Stanley Cheung on Unsplash

# Outline

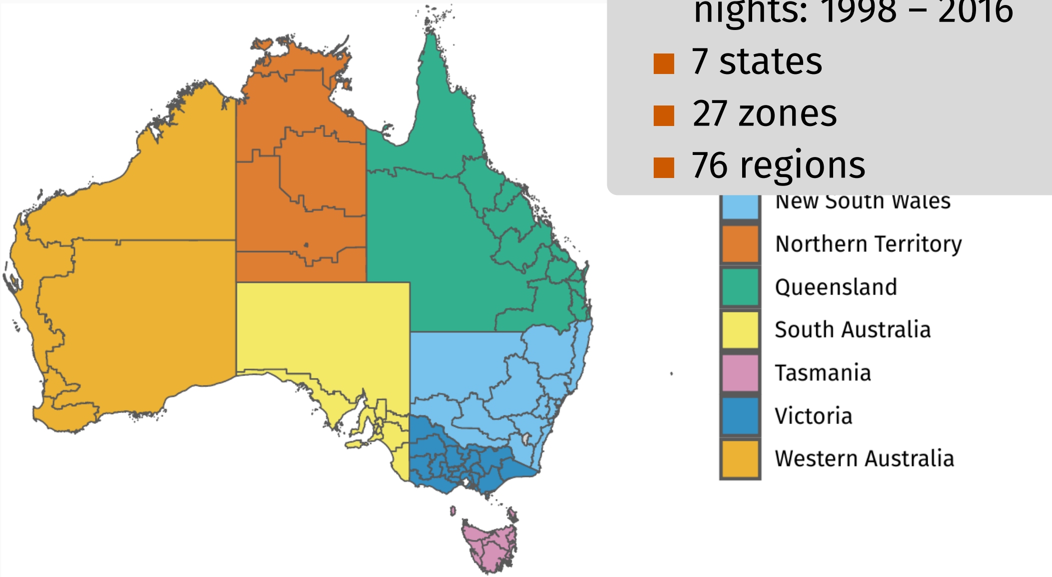
# Outline

# Australian tourism regions

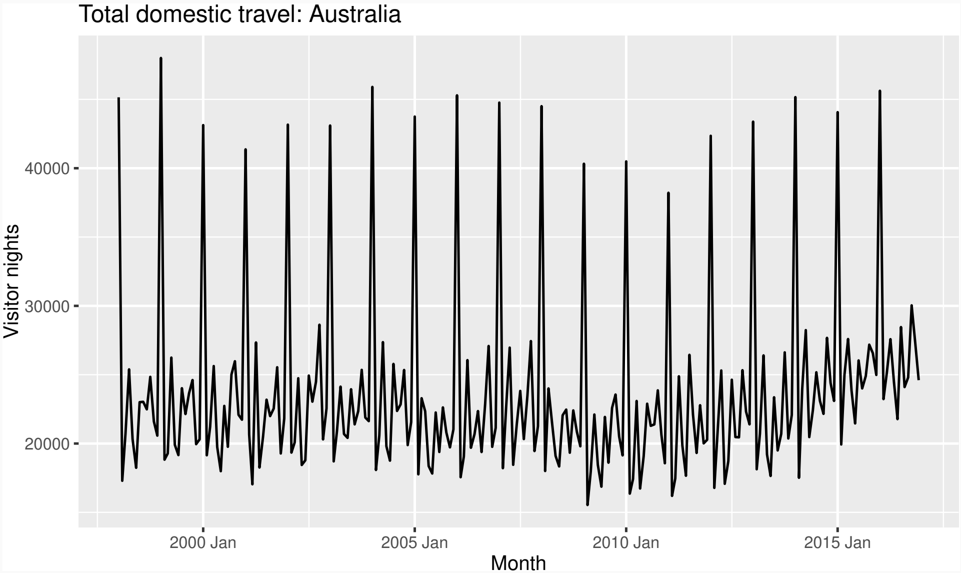


# Australian tourism regions

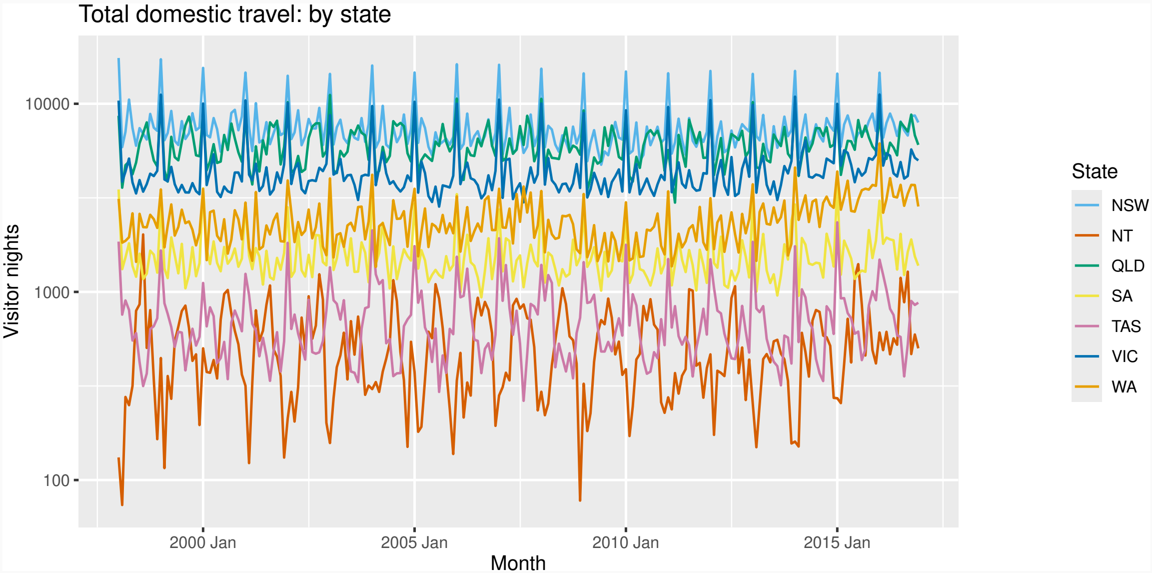
- Monthly data on visitor nights: 1998 – 2016
- 7 states
- 27 zones
- 76 regions



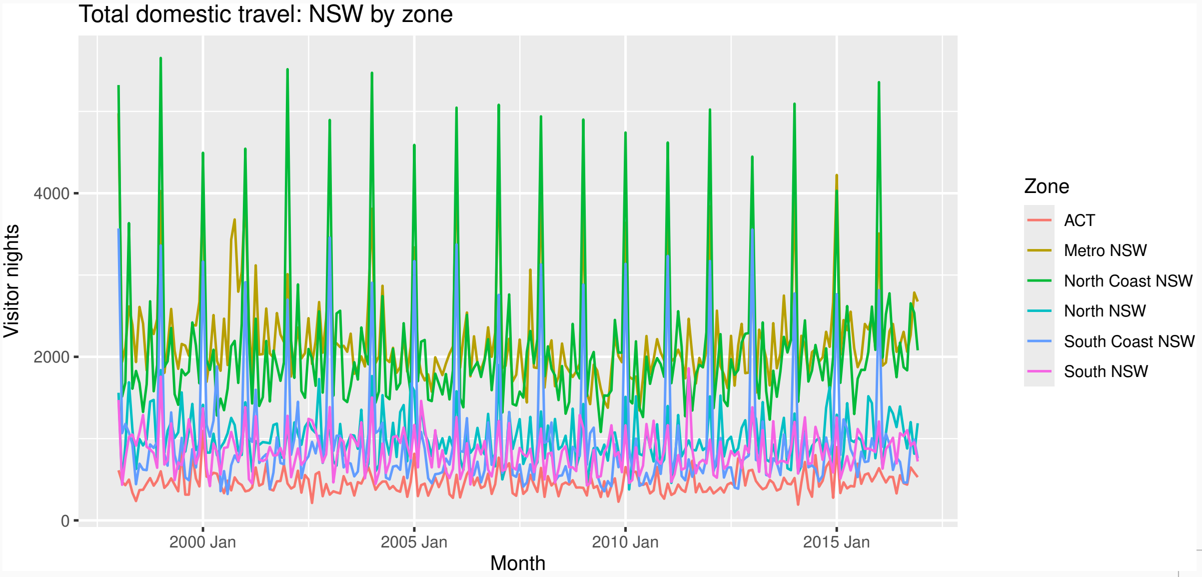
# Australian tourism data



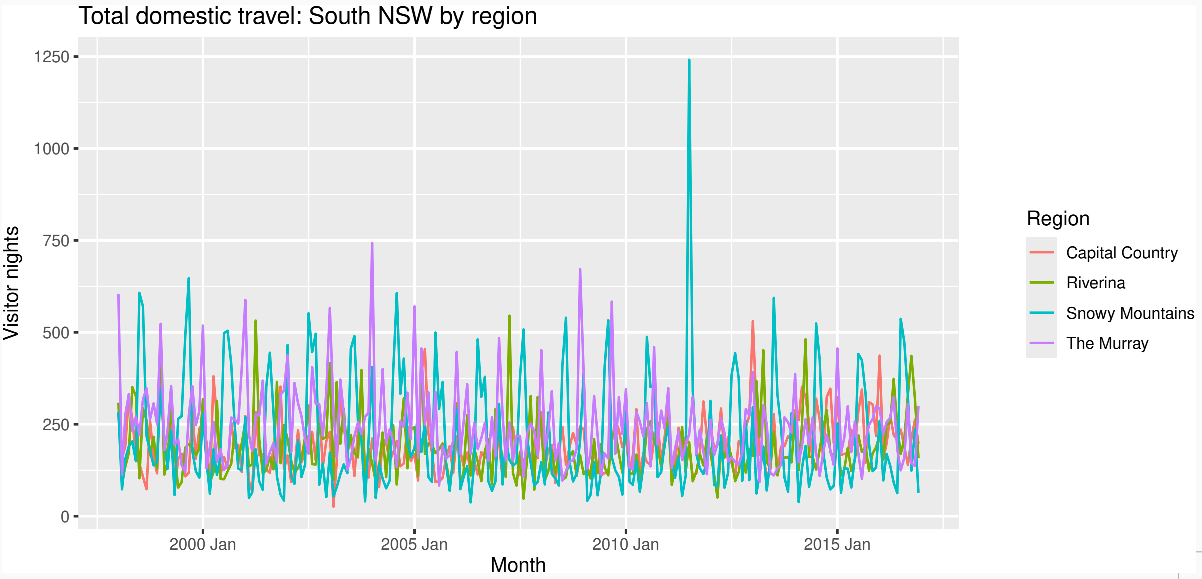
# Australian tourism data



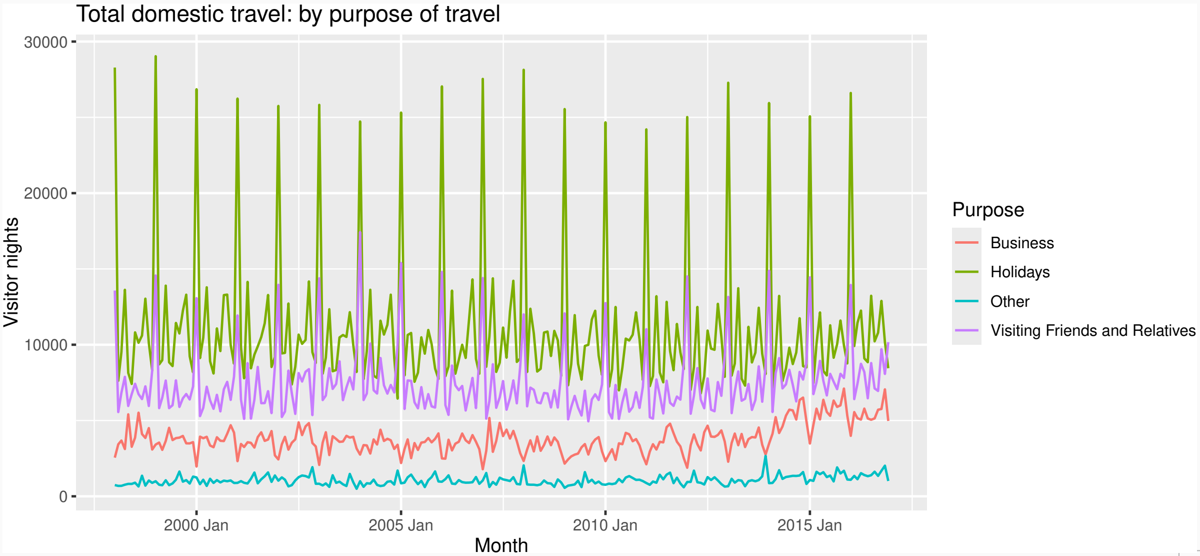
# Australian tourism data



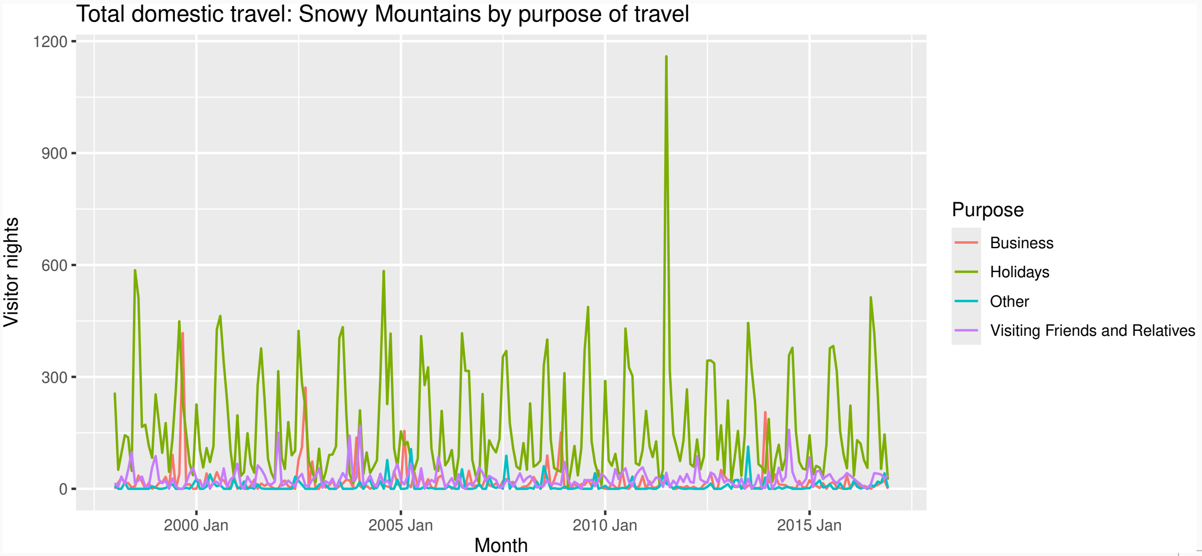
# Australian tourism data



# Australian tourism data

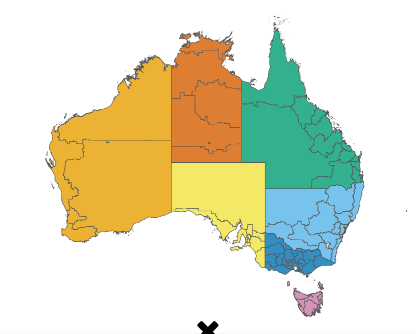


# Australian tourism data



# Australian tourism data

**Geographical division**



×

**Purpose of travel**

Holiday, Visiting friends & relatives, Business, Other

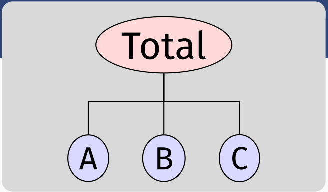
- **Grouped ts**
  (geographical divisions × purpose of travel)

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

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

- 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

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

- $\boldsymbol{y}_t$ = 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.

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

# The coherent subspace

## Coherent subspace

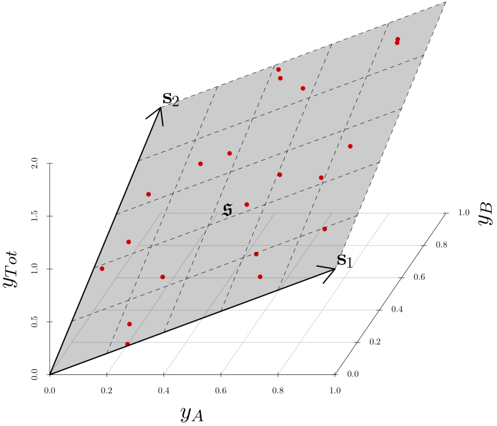
$n_b$-dimensional linear subspace $\mathfrak{s} \subset \chi^n$ for which linear constraints hold for all $\boldsymbol{y} \in \mathfrak{s}$.

## 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

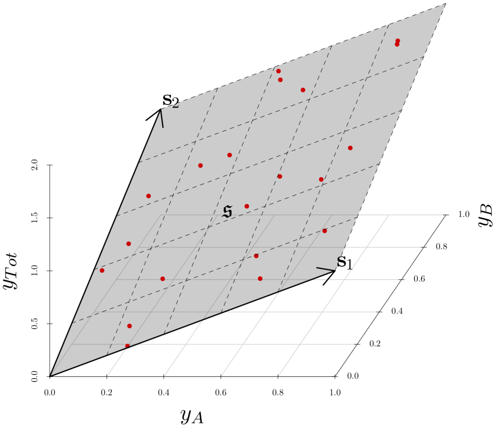
An $n$-dimensional multivariate time series such that $\boldsymbol{y}_t \in \mathfrak{s} \quad \forall t$.

## 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

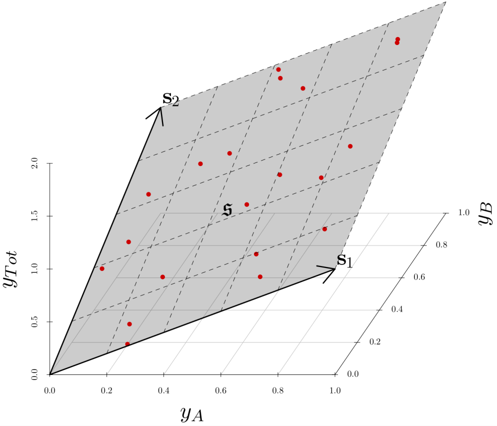
$n_b$-dimensional linear subspace $\mathfrak{s} \subset \chi^n$ for which linear constraints hold for all $\boldsymbol{y} \in \mathfrak{s}$.

## 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 $\boldsymbol{M}$ be a projection matrix. $\tilde{\boldsymbol{y}}_{t+h|t} = \boldsymbol{M}\hat{\boldsymbol{y}}_{t+h|t}$ "reconciles" $\hat{\boldsymbol{y}}_{t+h|t}$.

# Linear projection reconciliation

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

- 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 $\Psi$ is a positive definite matrix.
- 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.
- $\boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{T+h} - \hat{\boldsymbol{y}}_{T+h|T} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_T]$ 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.
- How to choose the best $\Psi$?

# Minimum trace reconciliation

Wickramasuriya et al (2019)

**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

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

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

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

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

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

```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)>
# i 101 more rows
```

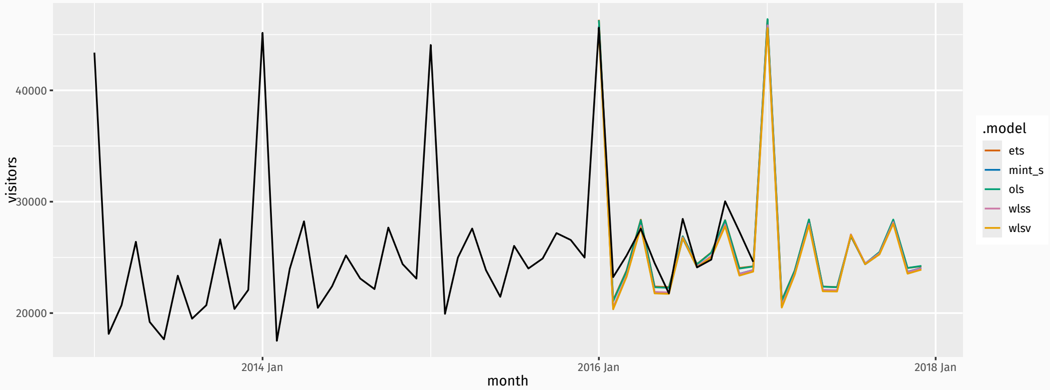
# Example: Australian tourism

```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
   <chr*> <chr*> <chr*>   <chr>      <mth>
 1 NSW    ACT    Canberra ets     2016 Jan
 2 NSW    ACT    Canberra ets     2016 Feb
 3 NSW    ACT    Canberra ets     2016 Mar
 4 NSW    ACT    Canberra ets     2016 Apr
 5 NSW    ACT    Canberra ets     2016 May
```

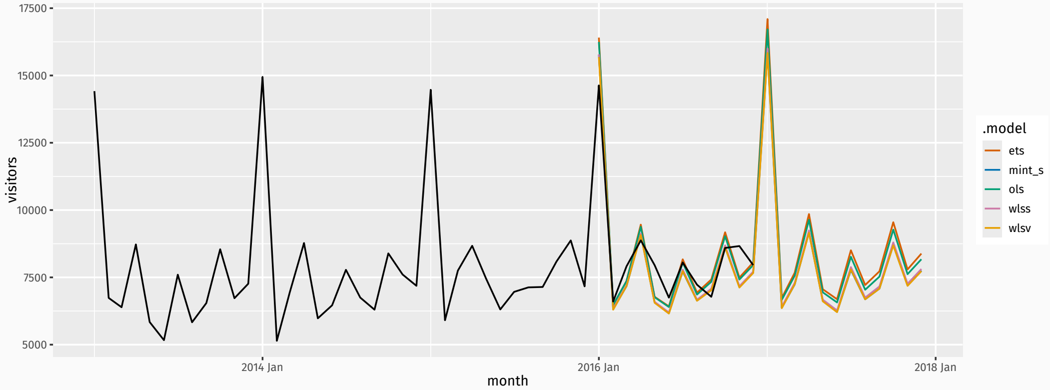
# Example: Australian tourism

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



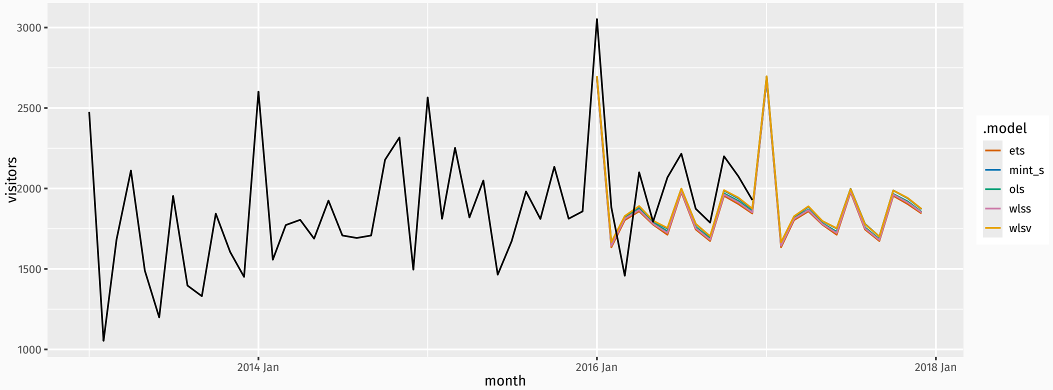
# Example: Australian tourism

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



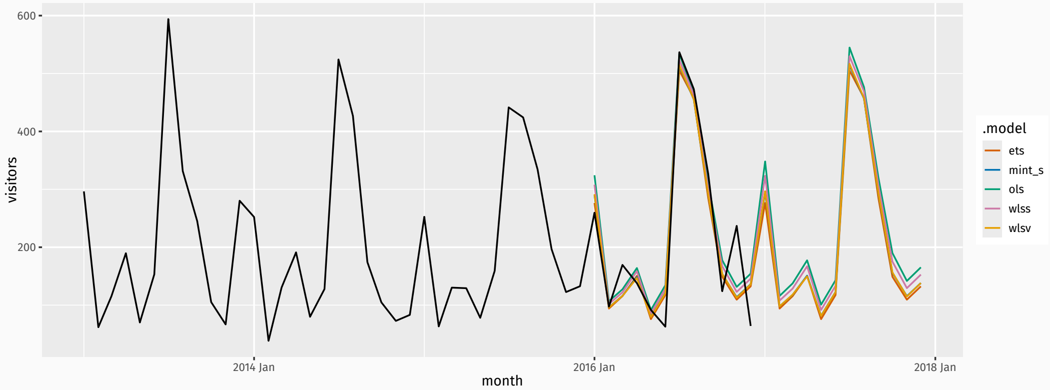
# Example: Australian tourism

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



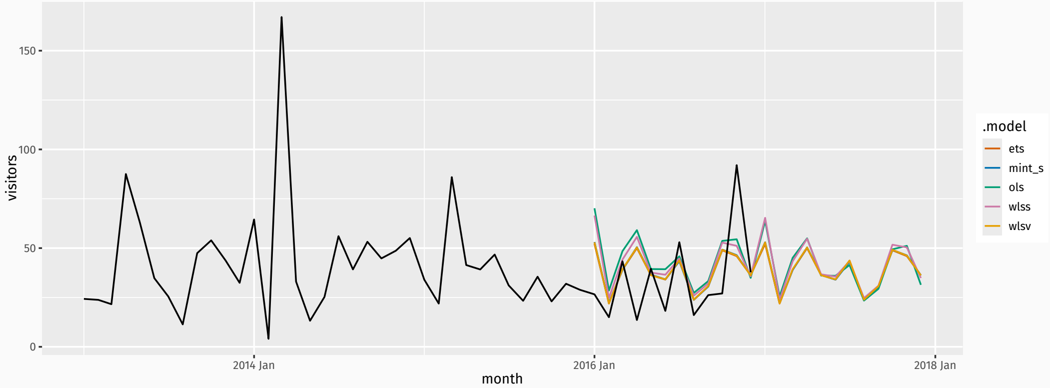
# Example: Australian tourism

```r
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

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

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

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

# Example: Australian tourism

```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

```
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
```

# 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
```

- 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

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

- $\Psi$-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)

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

- $\sigma_{\text{max}}$ 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)

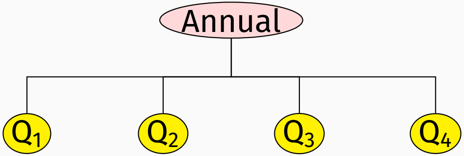
$$\begin{aligned}
&\text{tr}\Big(\text{E}[\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}^{\text{MinT}}_{t+h|t}]'[\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}^{\text{MinT}}_{t+h|t}]\Big)\\
\le\ &\text{tr}\Big(\text{E}[\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}^{\text{OLS}}_{t+h|t}]'[\boldsymbol{y}_{t+h}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

Using sums of variances:

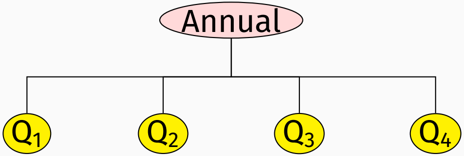
- MinT reconciliation is better than OLS reconciliation
- OLS reconciliation is better than base forecasts

# Outline

# Temporal reconciliation: quarterly data

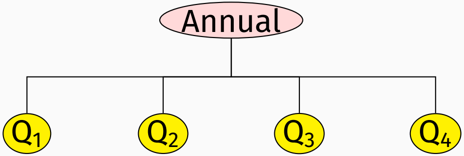


# Temporal reconciliation: quarterly data




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

# Temporal reconciliation: quarterly data

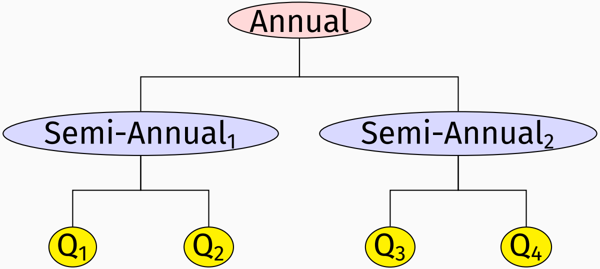



$$\boldsymbol{y}_\tau = \begin{bmatrix} x_\tau^{[4]} \\ x_{\tau,1}^{[1]} \\ x_{\tau,2}^{[1]} \\ x_{\tau,3}^{[1]} \\ x_{\tau,4}^{[1]} \end{bmatrix} \qquad \boldsymbol{S} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$\tau$ = index of largest temporal aggregation level.

# Temporal reconciliation: quarterly data

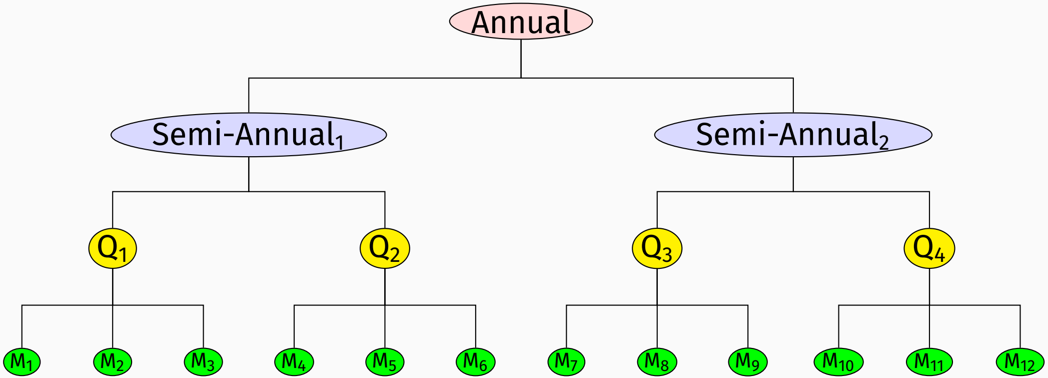



- Forecast series at each available frequency.
- Optimally combine forecasts within the same year.

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

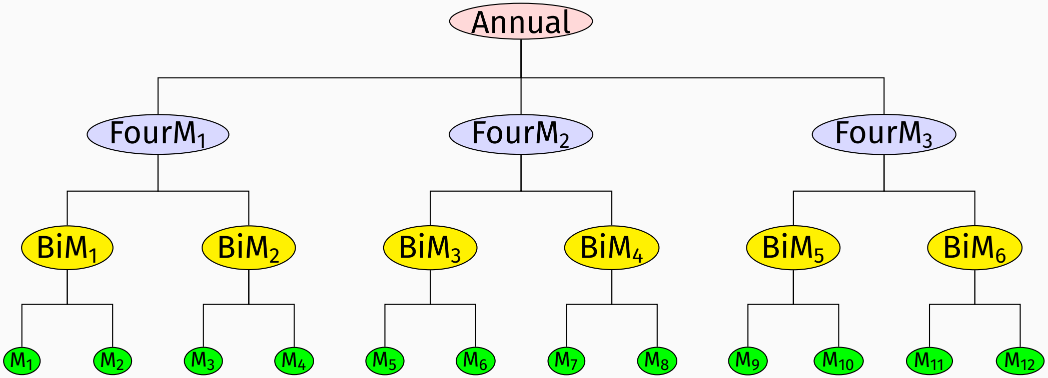
$\tau$ = index of largest temporal aggregation level.

# 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

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

$\tau$ 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

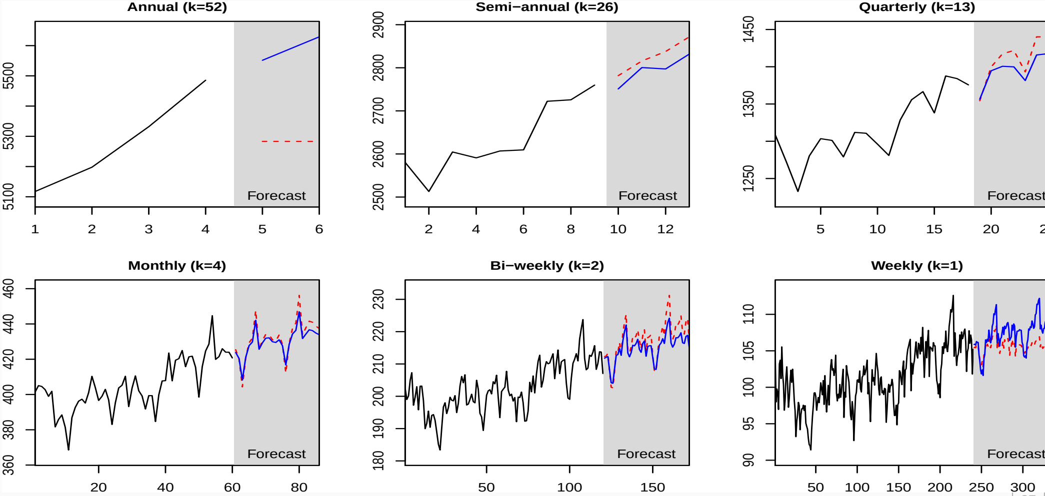
Weekly A&E demand data: 7 November 2010 to 7 June 2015.

---

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)

| Aggregation 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:
  - $\text{WLS}_H$ (diagonal)
  - $\text{WLS}_V$ (diagonal with common variances for same frequency)
  - $\text{WLS}_S$ (diagonal/structural)

# Temporal reconciliation: M3 monthly series

## Improvement in MASE relative to base forecasts

| | | ETS | | | | ARIMA | | | |
|---|---|---|---|---|---|---|---|---|---|
| Aggregation level | *h* | BU | $\text{WLS}_H$ | $\text{WLS}_V$ | $\text{WLS}_S$ | BU | $\text{WLS}_H$ | $\text{WLS}_V$ | $\text{WLS}_S$ |
| Annual | 1 | -12.1 | -17.9 | -17.8 | **-18.5** | -25.4 | -29.9 | -29.9 | **-30.2** |
| Semi-annual | 3 | 0.0 | -6.3 | -6.0 | **-6.9** | -2.9 | -8.1 | -8.2 | **-9.4** |
| Four-monthly | 4 | 3.1 | -3.2 | -3.0 | **-3.4** | -1.8 | -6.2 | -6.5 | **-7.1** |
| Quarterly | 6 | 3.2 | -2.8 | -2.7 | **-3.4** | -2.6 | -6.9 | -7.4 | **-8.1** |
| Bi-monthly | 9 | 2.7 | -2.9 | -3.0 | **-3.7** | -1.3 | -5.0 | -5.5 | **-6.3** |
| Monthly | 18 | 0.0 | -3.7 | -4.6 | **-5.0** | 0.0 | -1.9 | -3.2 | **-3.7** |
| Average | NA | -0.5 | -6.1 | -6.2 | **-6.8** | -5.7 | -9.7 | -10.1 | **-10.8** |

# Outline

# Forecast Linear Augmented Projection (FLAP)

- We want to forecast a multivariate series $\boldsymbol{y}_t$.
- Construct many linear combinations $\boldsymbol{c}_t = \Phi\boldsymbol{y}_t$ of the multivariate series (e.g., principal components or random combinations)
- Produce univariate forecasts of all series $\hat{\boldsymbol{y}}_t$ and all linear combinations $\hat{\boldsymbol{c}}_t$.
- Reconcile forecasts so they are coherent ($\tilde{\boldsymbol{c}}_t = \Phi\tilde{\boldsymbol{y}}_t$)

# Forecast Linear Augmented Projection (FLAP)

- We want to forecast a multivariate series $\boldsymbol{y}_t$.
- Construct many linear combinations $\boldsymbol{c}_t = \Phi\boldsymbol{y}_t$ of the multivariate series (e.g., principal components or random combinations)
- Produce univariate forecasts of all series $\hat{\boldsymbol{y}}_t$ and all linear combinations $\hat{\boldsymbol{c}}_t$.
- Reconcile forecasts so they are coherent ($\tilde{\boldsymbol{c}}_t = \Phi\tilde{\boldsymbol{y}}_t$)

$$\boldsymbol{z}_t = \begin{bmatrix}\boldsymbol{y}_t \\ \boldsymbol{c}_t\end{bmatrix} \qquad \tilde{\boldsymbol{z}}_{t+h} = \boldsymbol{M}\hat{\boldsymbol{z}}_{t+h}$$

where $\boldsymbol{M}$ is a projection matrix onto the coherent subspace.

# Forecast error variance reduction

- The variance reduction $\text{Var}(\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_{t+h}) - \text{Var}(\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}_{t+h})$ is **positive semi-definite**.
- The diagonal elements of $\text{Var}(\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_{t+h}) - \text{Var}(\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}_{t+h})$ are non-decreasing as the number of components increases.

# Minimum variance of individual series

The projection is equivalent to the mapping

$$\tilde{\boldsymbol{y}}_{t+h} = \boldsymbol{G}\hat{\boldsymbol{z}}_{t+h},$$

where $\boldsymbol{G} = \begin{bmatrix}\boldsymbol{g}_1 & \boldsymbol{g}_2 & \dots & \boldsymbol{g}_m\end{bmatrix}' \in \mathbb{R}^{m\times(m+p)}$ is the solution to

$$\arg\min_{\boldsymbol{G}} \boldsymbol{G}\boldsymbol{W}_h\boldsymbol{G}' \qquad \text{s.t. } \boldsymbol{G}\boldsymbol{S} = \boldsymbol{I}$$

or

$$\arg\min_{\boldsymbol{g}_i} \boldsymbol{g}_i'\boldsymbol{W}_h\boldsymbol{g}_i \qquad \text{s.t. } \boldsymbol{g}_i'\boldsymbol{s}_j = \boldsymbol{1}(i = j),$$

where $\boldsymbol{S} = \begin{bmatrix}\boldsymbol{I}_m \\ \boldsymbol{\Phi}\end{bmatrix} = \begin{bmatrix}\boldsymbol{s}_1 \cdots \boldsymbol{s}_m\end{bmatrix}$.

# Key results

1. The forecast error variance is **reduced** with FLAP
2. The forecast error variance **monotonically** decreases with increasing number of components
3. The forecast projection is **optimal** to achieve minimum forecast error variance of each series

# Key results

1. The forecast error variance is **reduced** with FLAP
2. The forecast error variance **monotonically** decreases with increasing number of components
3. The forecast projection is **optimal** to achieve minimum forecast error variance of each series

In practice, we need to:

- Estimate $\boldsymbol{W}_h = \mathrm{Var}(\boldsymbol{z}_{t+h} - \hat{\boldsymbol{z}}_{t+h})$. (We can use the MinT shrinkage estimator.)
- Construct the components, $\boldsymbol{\Phi}$.

# Construction of $\Phi$

## Principal component analysis (PCA)

Finding the weights matrix $\Phi$ so that the resulting components **maximise variance**

## Simulation

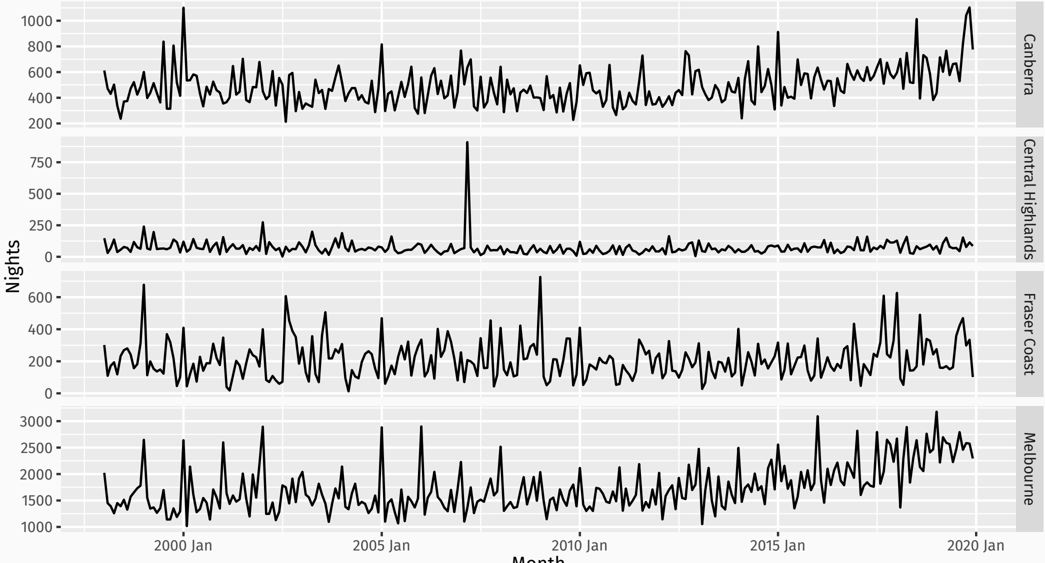
Generating values of $\Phi$ from a random distribution and normalising them to unit vectors

- Normal distribution
- Uniform distribution
- Orthonormal matrix

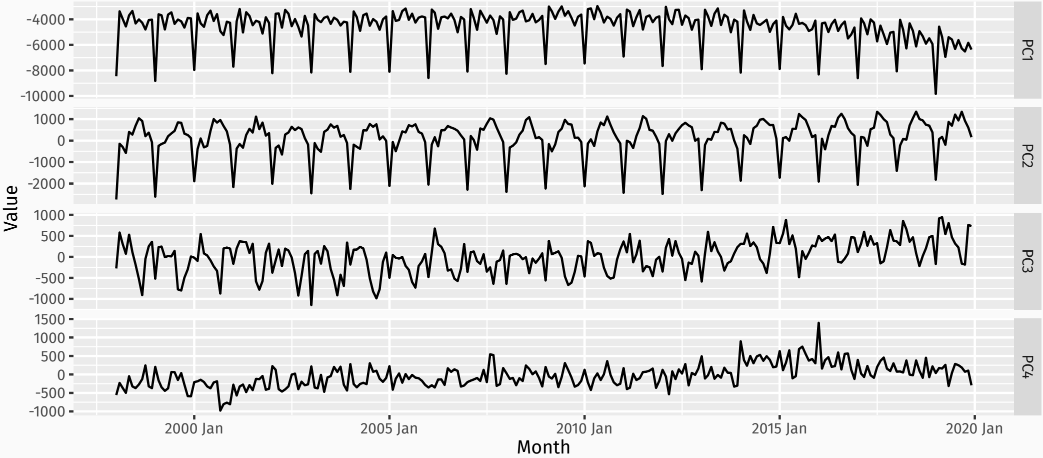
# Monthly Australian regional tourism

- Monthly Australian tourism data set aggregated by region giving 77 series, from Jan 1998 to Dec 2019
- Use expanding window time series cross-validation with $T = 84$ observations in first training set, and forecast horizons $h = 1, 2, \ldots, 12$.

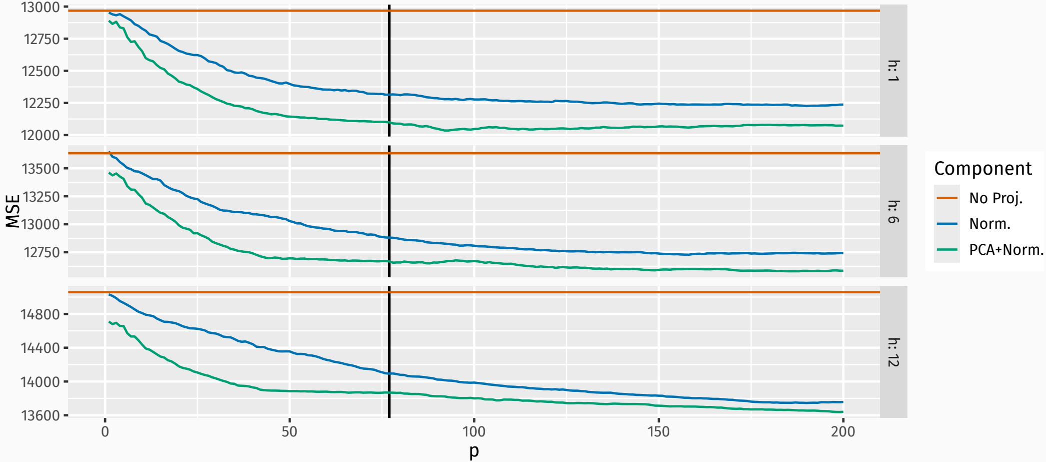
# Monthly Australian regional tourism



# Monthly Australian regional tourism



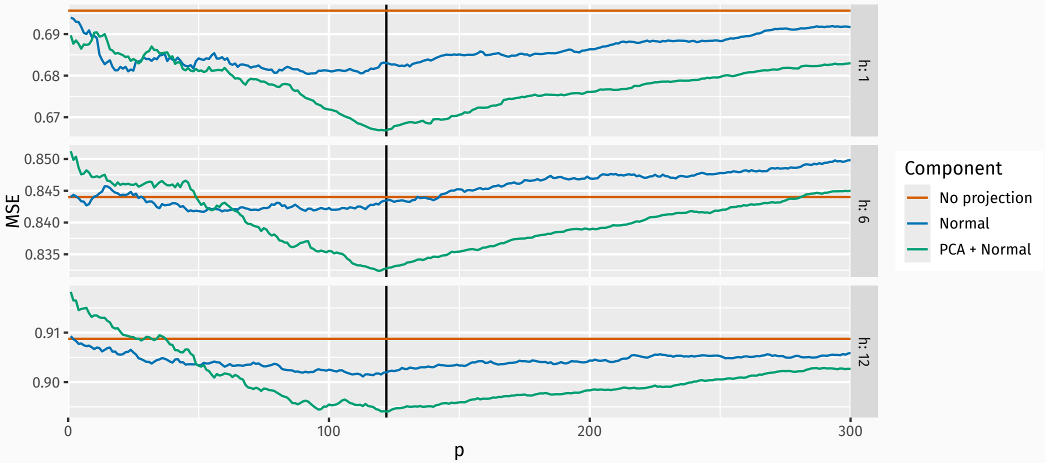
# Monthly Australian regional tourism



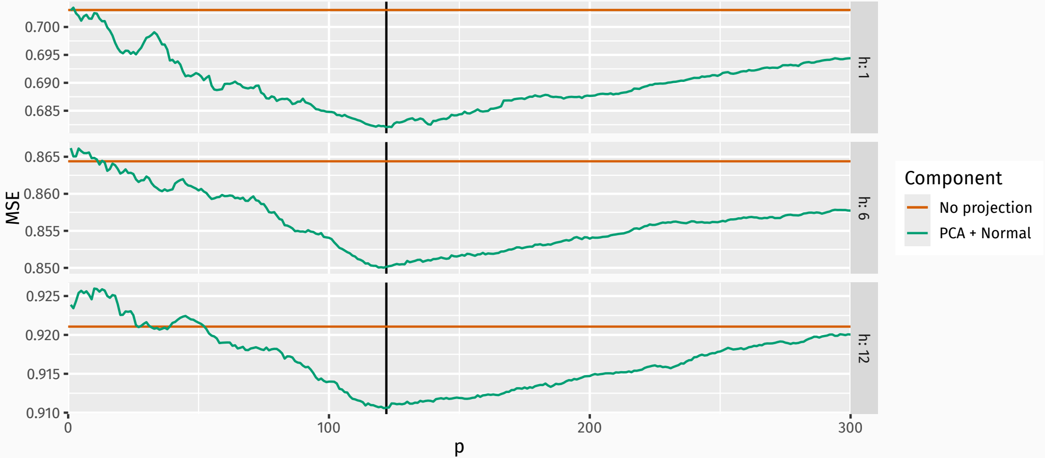
# FRED-MD

- Monthly data of macroeconomic variables (McCracken and Ng, 2016).
- Data from Jan 1959 – Sep 2023. 777 observations on 122 series.
- Same cleaning process as per McCracken and Ng (2016).
- All series scaled to have mean 0 and variance 1.
- Expanding time series cross-validation with initial size of 25 years and forecast horizon 12 months.

# FRED-MD (ARIMA)



# FRED-MD (DFM)

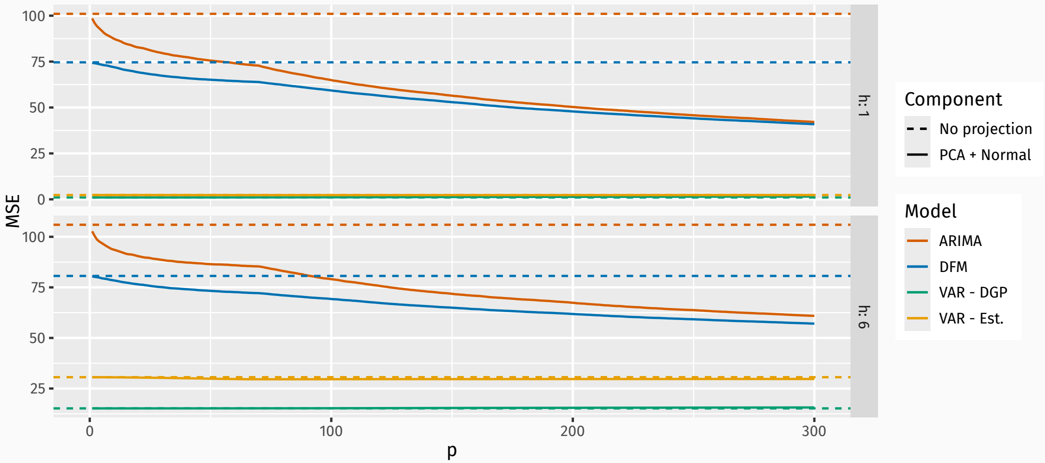


MSE
0.700
0.695
0.690
0.685
h: 1
0.865
0.860
0.855
0.850
h: 6
0.925
0.920
0.915
0.910
h: 12
0
100
200
300
p
Component
No projection
PCA + Normal

# Simulation

- Data generating process: VAR(3) with 70 variables
- Sample size: $T = 400$
- Number of repeated samples: 220
- Base models:
  - automatic ARIMA (based on AICc)
  - DFM (structure chosen using BIC, different model for each horizon)

# Simulation



MSE
100
75
50
25
0
100
75
50
25
0
100
200
300
p
h: 1
h: 6
Component
No projection
PCA + Normal
Model
ARIMA
DFM
VAR - DGP
VAR - Est.

# Future research directions

- Investigate why PCA performs better than random weights
- Find other components that are better than PCA
- Find optimal components by minimising forecast error variance with respect to $\Phi$
- Use forecast projection and forecast reconciliation together

# Outline

# Software

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

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

# Thanks!

# More information

robjhyndman.com

aus.social/@robjhyndman

@robjhyndman

rob.hyndman@monash.edu

robjhyndman.com/curtin2024

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: methodological and practical issues”.** 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.

- Panagiotelis, A, P Gamakumara, G Athanasopoulos, and RJ Hyndman (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.
- Yang, YF, G Athanasopoulos, RJ Hyndman, and A Panagiotelis (2024). **“Forecast Linear Augmented Projection (FLAP): A free lunch to reduce forecast error variance”.** Working paper. URL: https://robjhyndman.com/publications/flap.html.