

Forecast reconciliation

4. Temporal & cross-temporal forecast reconciliation

Rob J Hyndman



MONASH University

Photo by Edvard Alexander Rølvaag on Unsplash

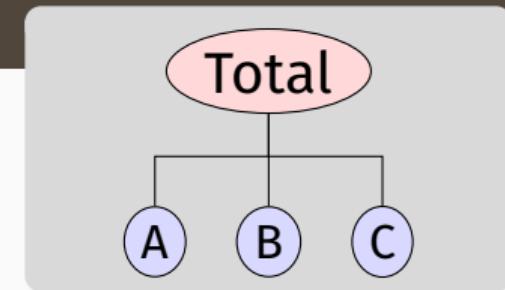
robjhyndman.com/fr2023

Notation reminder

Every collection 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.



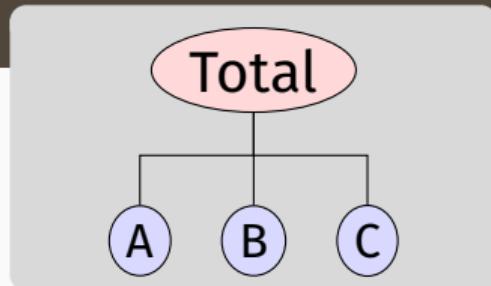
$$\begin{aligned} \mathbf{y}_t &= \begin{pmatrix} y_{\text{Total},t} \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_S \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\mathbf{b}_t}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

Notation reminder

Every collection 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.

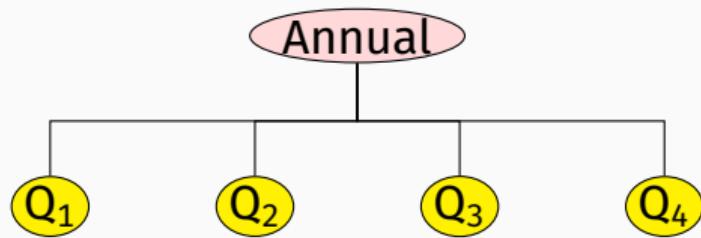


- Base forecasts: $\hat{\mathbf{y}}_{T+h|T}$
- Reconciled forecasts:
 $\tilde{\mathbf{y}}_{T+h|T} = \mathbf{S}\mathbf{G}\hat{\mathbf{y}}_{T+h|T}$
- MinT:
 $\mathbf{G} = (\mathbf{S}'\mathbf{W}_h^{-1}\mathbf{S})^{-1}\mathbf{S}'\mathbf{W}_h^{-1}$
where \mathbf{W}_h is covariance matrix of base forecast errors.

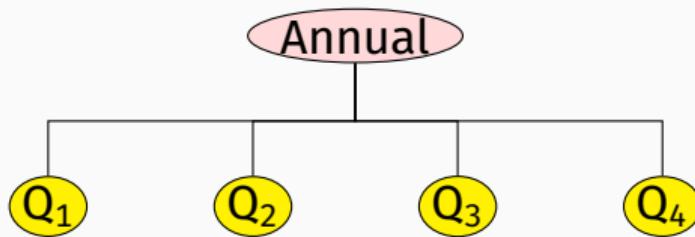
Outline

- 1 Temporal reconciliation
- 2 Temporal reconciliation: thief package
- 3 Temporal reconciliation: Examples
- 4 Cross-temporal forecast reconciliation
- 5 Cross-temporal probabilistic forecast reconciliation
- 6 Cross-temporal probabilistic reconciliation: Examples
- 7 Software

Temporal reconciliation: quarterly data

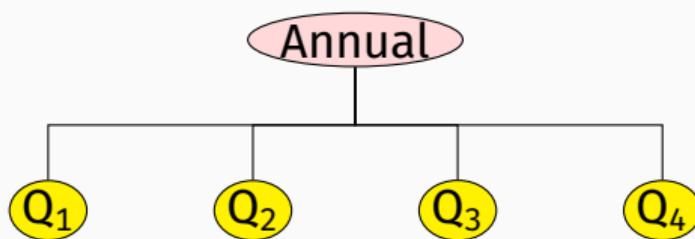


Temporal reconciliation: quarterly data



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

Temporal reconciliation: quarterly data

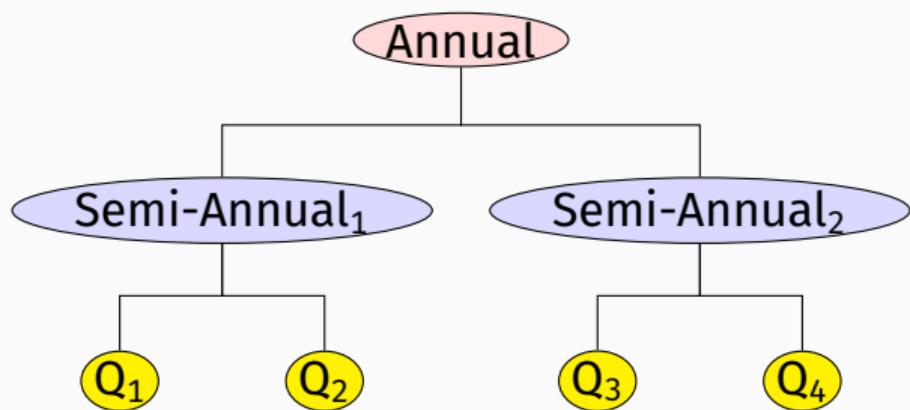


$$\mathbf{y}_\tau = \begin{bmatrix} x_\tau^{[4]} \\ x_{\tau,1}^{[1]} \\ x_{\tau,2}^{[1]} \\ x_{\tau,3}^{[1]} \\ x_{\tau,4}^{[1]} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

τ = index of largest temporal aggregation level.

Temporal reconciliation: quarterly data

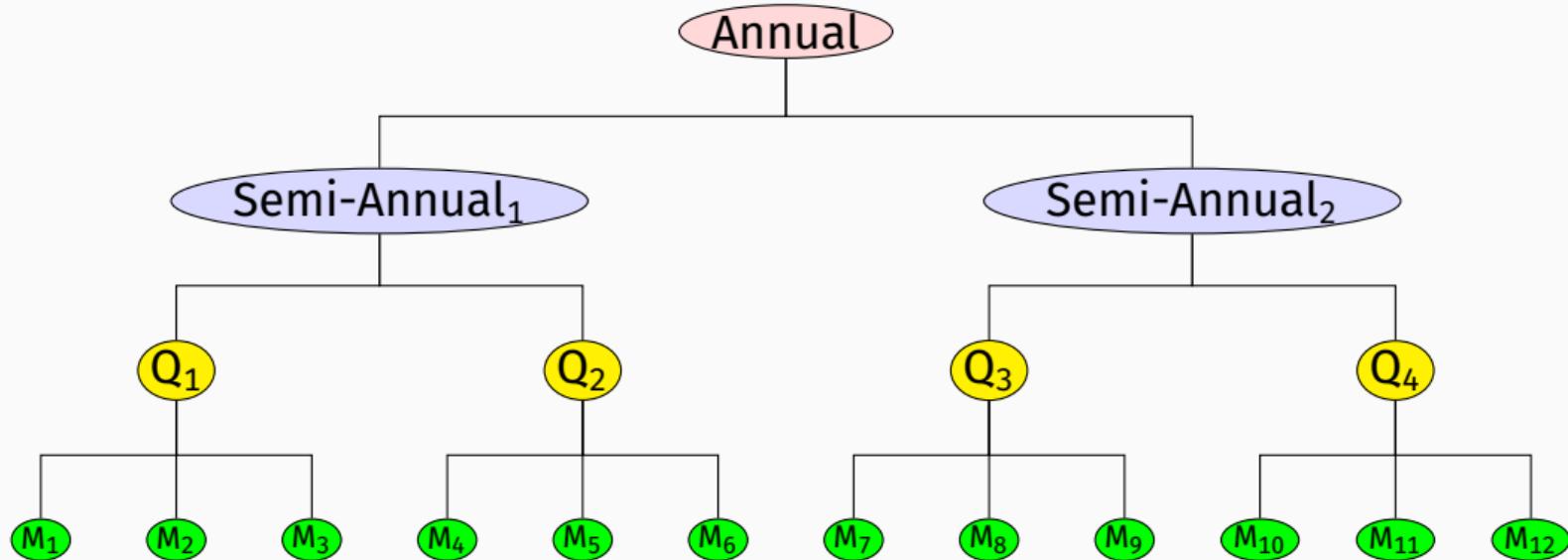


- Forecast series at each available frequency.
- Optimally combine forecasts within the same year.

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

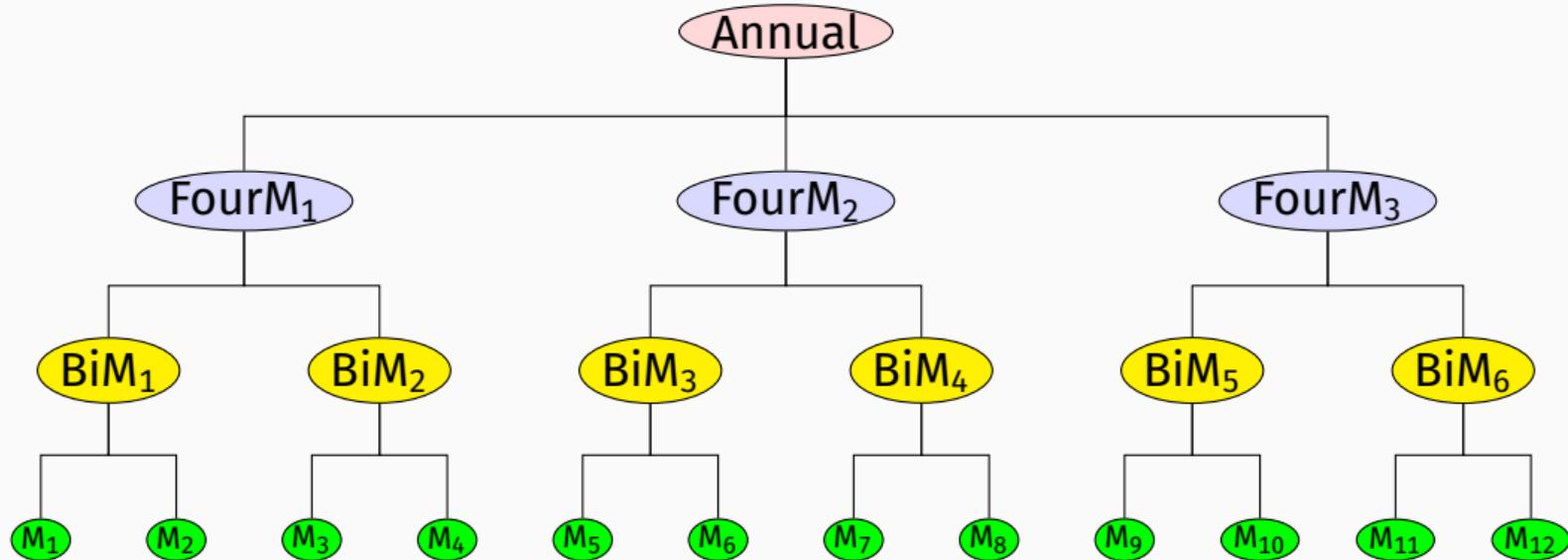
τ = index of largest temporal aggregation level.

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 :

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

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

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

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,

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

Outline

- 1 Temporal reconciliation
- 2 Temporal reconciliation: thief package
- 3 Temporal reconciliation: Examples
- 4 Cross-temporal forecast reconciliation
- 5 Cross-temporal probabilistic forecast reconciliation
- 6 Cross-temporal probabilistic reconciliation: Examples
- 7 Software

Temporal reconciliation: thief package

```
library(thief)
```

```
USAccDeaths
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1973	9007	8106	8928	9137	10017	10826	11317	10744	9713	9938	9161	8927
1974	7750	6981	8038	8422	8714	9512	10120	9823	8743	9129	8710	8680
1975	8162	7306	8124	7870	9387	9556	10093	9620	8285	8466	8160	8034
1976	7717	7461	7767	7925	8623	8945	10078	9179	8037	8488	7874	8647
1977	7792	6957	7726	8106	8890	9299	10625	9302	8314	8850	8265	8796
1978	7836	6892	7791	8192	9115	9434	10484	9827	9110	9070	8633	9240

```
thief(USAccDeaths, usemodel = "arima", comb = "struc")
```

	Jan	Feb	Mar	Apr	May	Jun	Jul
1979	8185.240	7381.008	8131.674	8516.070	9393.247	9764.091	10880.835
1980	8405.850	7601.618	8352.284	8736.680	9613.857	9984.701	11101.445
	Aug	Sep	Oct	Nov	Dec		
1979	10059.873	9182.403	9405.069	8869.173	9360.772		
1980	10280.483	9403.013	9625.679	9089.783	9581.382		

Temporal reconciliation: thief package

```
# Construct aggregates  
aggts <- tsaggregates(USAccDeaths)  
aggts[[1]]
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1973	9007	8106	8928	9137	10017	10826	11317	10744	9713	9938	9161	8927
1974	7750	6981	8038	8422	8714	9512	10120	9823	8743	9129	8710	8680
1975	8162	7306	8124	7870	9387	9556	10093	9620	8285	8466	8160	8034
1976	7717	7461	7767	7925	8623	8945	10078	9179	8037	8488	7874	8647
1977	7792	6957	7726	8106	8890	9299	10625	9302	8314	8850	8265	8796
1978	7836	6892	7791	8192	9115	9434	10484	9827	9110	9070	8633	9240

Temporal reconciliation: thief package

```
aggs[[2]]
```

Time Series:

Start = c(1973, 1)

End = c(1978, 6)

Frequency = 6

```
[1] 17113 18065 20843 22061 19651 18088 14731 16460 18226 19943 17872 17390  
[13] 15468 15994 18943 19713 16751 16194 15178 15692 17568 19257 16525 16521  
[25] 14749 15832 18189 19927 17164 17061 14728 15983 18549 20311 18180 17873
```

Temporal reconciliation: thief package

```
aggts[[3]]
```

	Qtr1	Qtr2	Qtr3	Qtr4
1973	26041	29980	31774	28026
1974	22769	26648	28686	26519
1975	23592	26813	27998	24660
1976	22945	25493	27294	25009
1977	22475	26295	28241	25911
1978	22519	26741	29421	26943

Temporal reconciliation: thief package

```
aggs[[4]]
```

Time Series:

Start = c(1973, 1)

End = c(1978, 3)

Frequency = 3

```
[1] 35178 42904 37739 31191 38169 35262 31462 38656 32945 30870 36825 33046  
[13] 30581 38116 34225 30711 38860 36053
```

Temporal reconciliation: thief package

```
aggs[[5]]
```

Time Series:

Start = c(1973, 1)

End = c(1978, 2)

Frequency = 2

```
[1] 56021 59800 49417 55205 50405 52658 48438 52303 48770 54152 49260 56364
```

Temporal reconciliation: thief package

```
aggs[[6]]
```

Time Series:

Start = 1973

End = 1978

Frequency = 1

```
[1] 115821 104622 103063 100741 102922 105624
```

Temporal reconciliation: thief package

```
# Compute forecasts
fc <- list()
for(i in seq_along(aggrts)) {
  fc[[i]] <- forecast(auto.arima(aggrts[[i]]), h=2*frequency(aggrts[[i]])))
}
reconciled <- reconcilethief(fc)
reconciled[[1]]
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 1979	8185.240	7773.891	8596.589	7556.136	8814.344
Feb 1979	7381.008	6907.643	7854.373	6657.059	8104.957
Mar 1979	8131.674	7603.525	8659.822	7323.941	8939.407
Apr 1979	8516.070	7938.310	9093.830	7632.462	9399.678
May 1979	9393.247	8769.810	10016.683	8439.783	10346.710
Jun 1979	9764.091	9098.104	10430.079	8745.552	10782.631
Jul 1979	10880.835	10174.856	11586.813	9801.134	11960.535
Aug 1979	10059.873	9316.050	10803.695	8922.295	11197.451
Sep 1979	9182.403	8402.572	9962.235	7989.754	10375.053

Temporal reconciliation: thief package

```
reconciled[[2]]
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1979.000	15566.25	14788.53	16343.97	14376.83	16755.67
1979.167	16647.74	15634.48	17661.00	15098.10	18197.39
1979.333	19157.34	17953.79	20360.89	17316.66	20998.01
1979.500	20940.71	19573.09	22308.33	18849.11	23032.30
1979.667	18587.47	17073.46	20101.48	16271.99	20902.95
1979.833	18229.94	16582.50	19877.39	15710.40	20749.49
1980.000	16007.47	14159.37	17855.57	13181.05	18833.89
1980.167	17088.96	15077.40	19100.53	14012.54	20165.39
1980.333	19598.56	17435.85	21761.27	16290.98	22906.14
1980.500	21381.93	19077.96	23685.89	17858.32	24905.53
1980.667	19028.69	16591.65	21465.73	15301.56	22755.83
1980.833	18671.16	16107.94	21234.39	14751.06	22591.27

Temporal reconciliation: thief package

```
reconciled[[3]]
```

		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1979	Q1	23697.92	22460.67	24935.17	21805.71	25590.13
1979	Q2	27673.41	25923.67	29423.15	24997.41	30349.40
1979	Q3	30123.11	27980.13	32266.09	26845.70	33400.52
1979	Q4	27635.01	25160.51	30109.52	23850.59	31419.44
1980	Q1	24359.75	20860.28	27859.23	19007.77	29711.74
1980	Q2	28335.24	24049.27	32621.20	21780.42	34890.06
1980	Q3	30784.94	25835.93	35733.95	23216.09	38353.79
1980	Q4	28296.84	22763.69	33830.00	19834.61	36759.08

Temporal reconciliation: thief package

```
reconciled[[4]]
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1979.000	32213.99	30196.88	34231.11	29129.08	35298.90
1979.333	40098.05	37245.41	42950.68	35735.32	44460.77
1979.667	36817.42	33323.67	40311.16	31474.19	42160.64
1980.000	33096.43	27759.64	38433.22	24934.52	41258.34
1980.333	40980.49	34290.47	47670.50	30748.99	51211.98
1980.667	37699.86	29887.60	45512.11	25752.04	49647.67

Temporal reconciliation: thief package

```
reconciled[[5]]
```

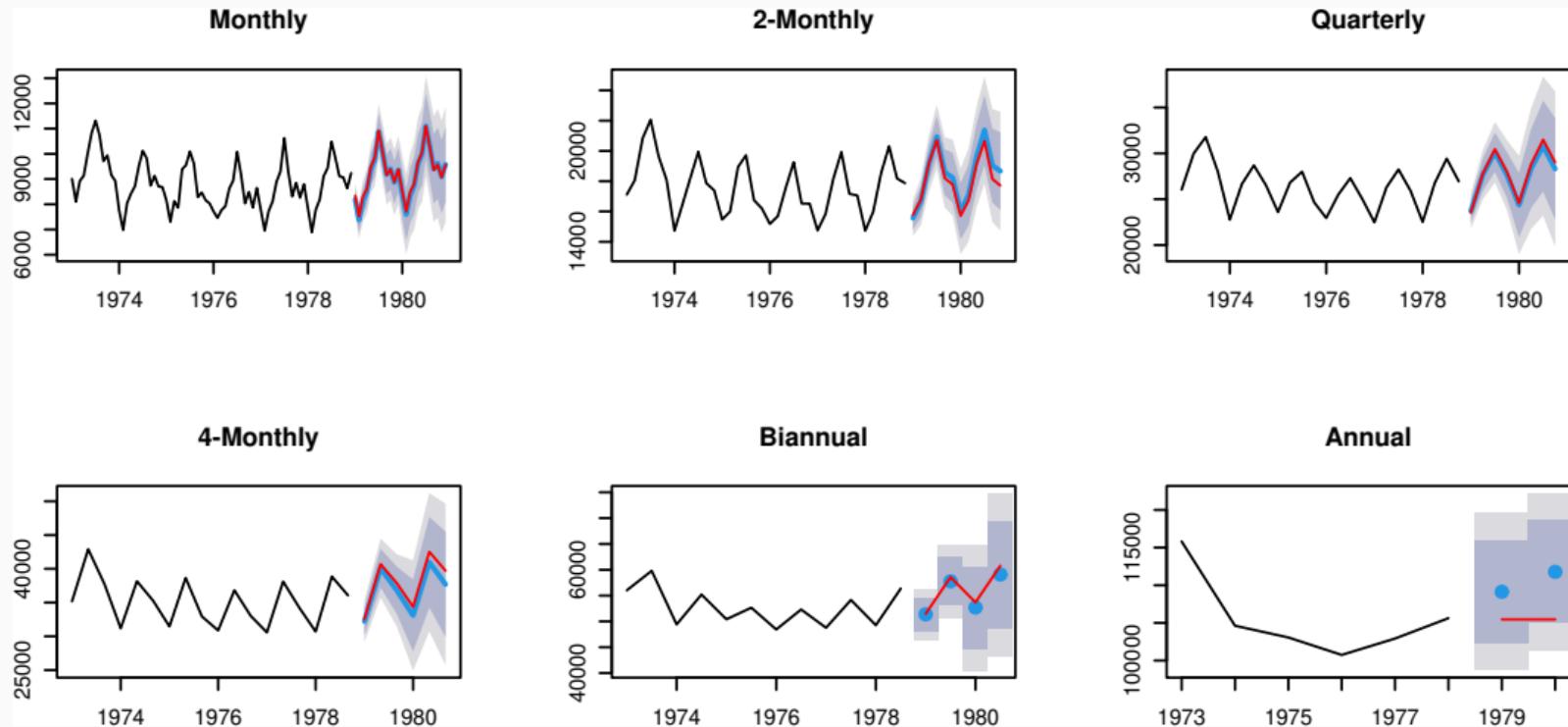
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1979.00	51371.33	48111.38	54631.28	46385.66	56357.00
1979.50	57758.12	53147.85	62368.40	50707.32	64808.93
1980.00	52694.99	44709.77	60680.21	40482.64	64907.33
1980.50	59081.78	48772.91	69390.66	43315.72	74847.85

Temporal reconciliation: thief package

```
reconciled[[6]]
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1979	109129.5	102287.0	115971.9	98664.81	119594.1
1980	111776.8	104934.3	118619.2	101312.13	122241.4

Temporal reconciliation: thief package



Outline

- 1 Temporal reconciliation
- 2 Temporal reconciliation: thief package
- 3 Temporal reconciliation: Examples
- 4 Cross-temporal forecast reconciliation
- 5 Cross-temporal probabilistic forecast reconciliation
- 6 Cross-temporal probabilistic reconciliation: Examples
- 7 Software

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

Aggregation		ETS				ARIMA					
level	<i>h</i>	Base	BU	WLS _H	WLS _V	WLS _S	Base	BU	WLS _H	WLS _V	WLS _S
RMAE											
Annual	1	1.0	-19.6	-22.0	-22.0	-25.1	1.0	-28.6	-33.1	-32.8	-33.4
Semi-annual	3	1.0	0.6	-4.0	-3.6	-5.4	1.0	-3.4	-8.2	-8.3	-9.9
Four-monthly	4	1.0	2.0	-2.4	-2.2	-3.0	1.0	-1.7	-5.5	-5.9	-6.7
Quarterly	6	1.0	2.4	-1.6	-1.7	-2.8	1.0	-3.6	-7.2	-8.1	-9.1
Bi-monthly	9	1.0	0.7	-2.9	-3.3	-4.3	1.0	-1.5	-4.4	-5.3	-6.3
Monthly	18	1.0	0.0	-2.2	-3.2	-3.9	1.0	0.0	-0.9	-2.9	-3.4
Average			-2.3	-5.9	-6.0	-7.4		-6.5	-9.9	-10.5	-11.5
MASE											
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

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

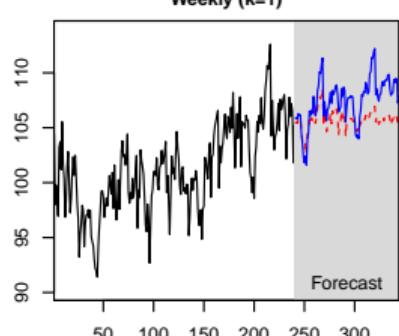
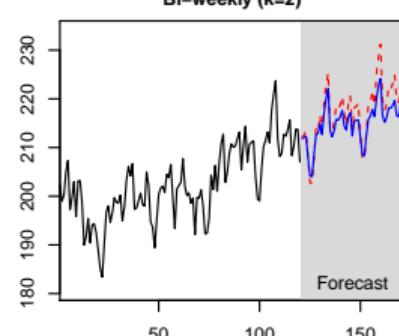
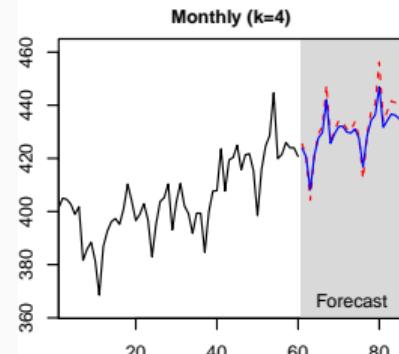
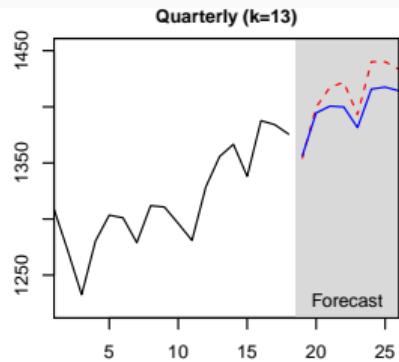
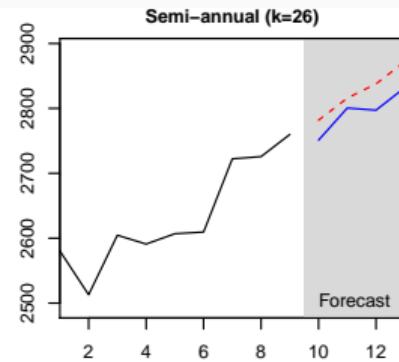
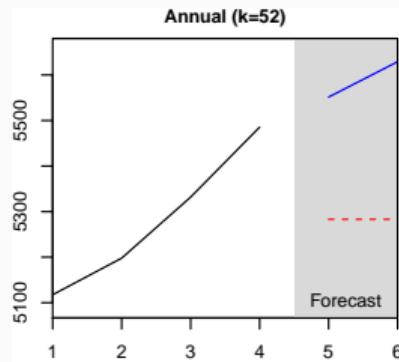
Other Emergency Admissions (i.e not via A&E)

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

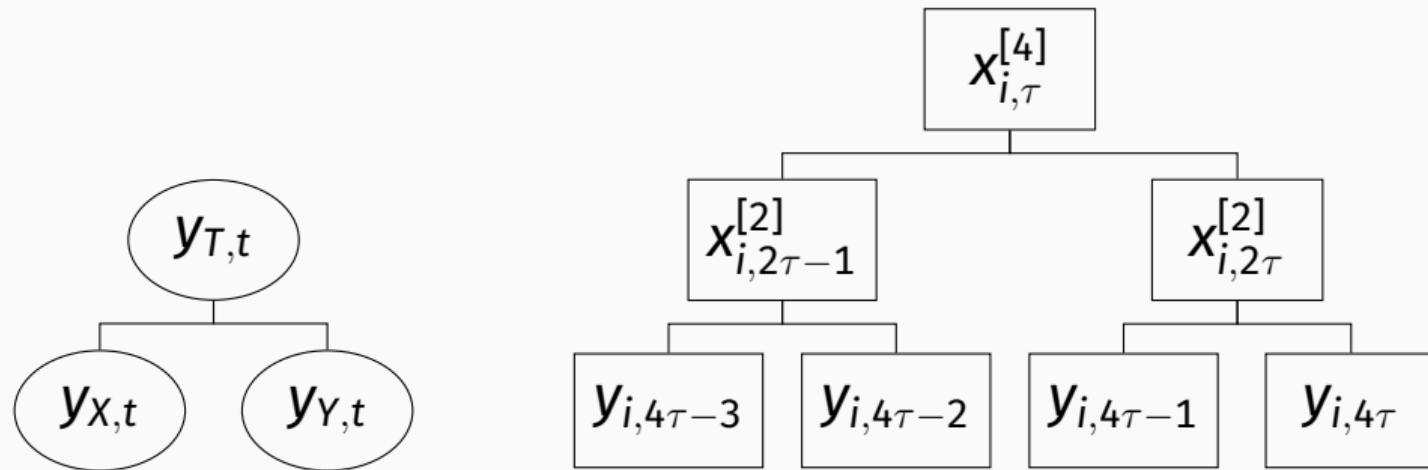
MASE comparison (ARIMA models)

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%

Outline

- 1 Temporal reconciliation
- 2 Temporal reconciliation: thief package
- 3 Temporal reconciliation: Examples
- 4 Cross-temporal forecast reconciliation
- 5 Cross-temporal probabilistic forecast reconciliation
- 6 Cross-temporal probabilistic reconciliation: Examples
- 7 Software

Cross-temporal forecast reconciliation



- $n = 3, n_a = 1, n_b = 2$
- Quarterly series: $m = 2, \mathcal{K} = \{1, 2, 4\}$

Cross-temporal forecast reconciliation

- \mathbf{y}_t = series at most temporally disaggregated level, including all cross-sectionally disaggregated and aggregated series.
- $y_{i,t}$ = i th element of \mathbf{y}_t , $i = 1, \dots, n$.
- For each i , we can expand $y_{i,t}$ to include all temporally aggregated variants:

$$\mathbf{x}_{i,\tau} = \begin{bmatrix} \mathbf{x}_{i,\tau}^{[k_p]} \\ \vdots \\ \mathbf{x}_{i,\tau}^{[k_1]} \end{bmatrix} \quad \mathbf{x}_\tau = \begin{bmatrix} \mathbf{x}_{1,\tau} \\ \vdots \\ \mathbf{x}_{n,\tau} \end{bmatrix}.$$

Cross-temporal forecast reconciliation

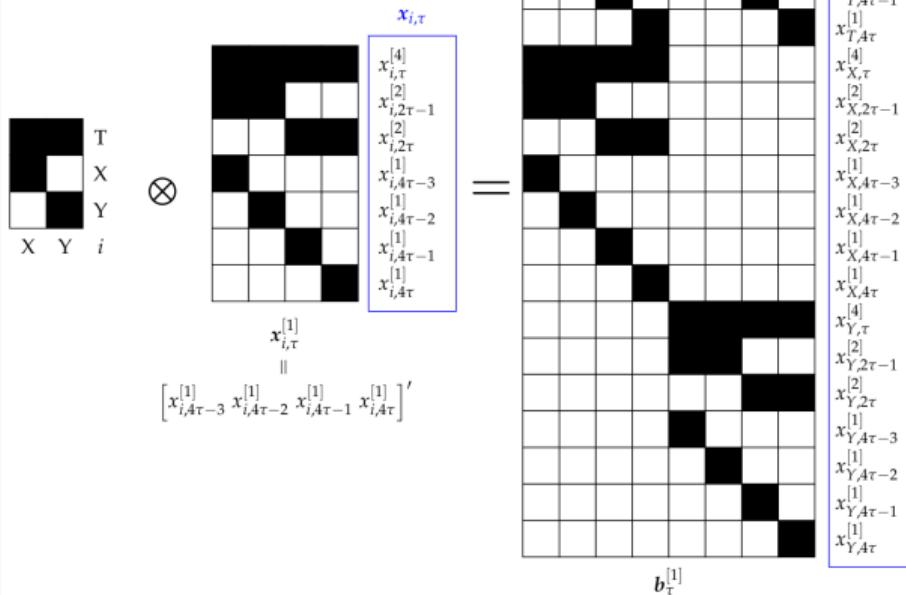
Structural matrix approach

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

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

Cross-temporal forecast reconciliation

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



Cross-temporal forecast reconciliation

Constraint matrix approach

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

$$\mathbf{C}_{ct} \mathbf{x}_\tau = \mathbf{0} \quad \text{where} \quad \mathbf{C}_{ct} = \begin{bmatrix} (\mathbf{0}_{(n_a m \times nk^*)} \ I_m \otimes \mathbf{C}_{cs}) \mathbf{P}' \\ \ I_n \otimes \mathbf{C}_{te} \end{bmatrix}$$

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

Outline

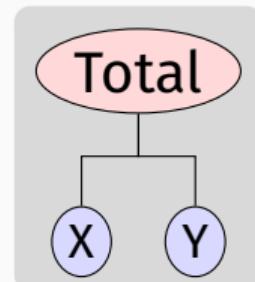
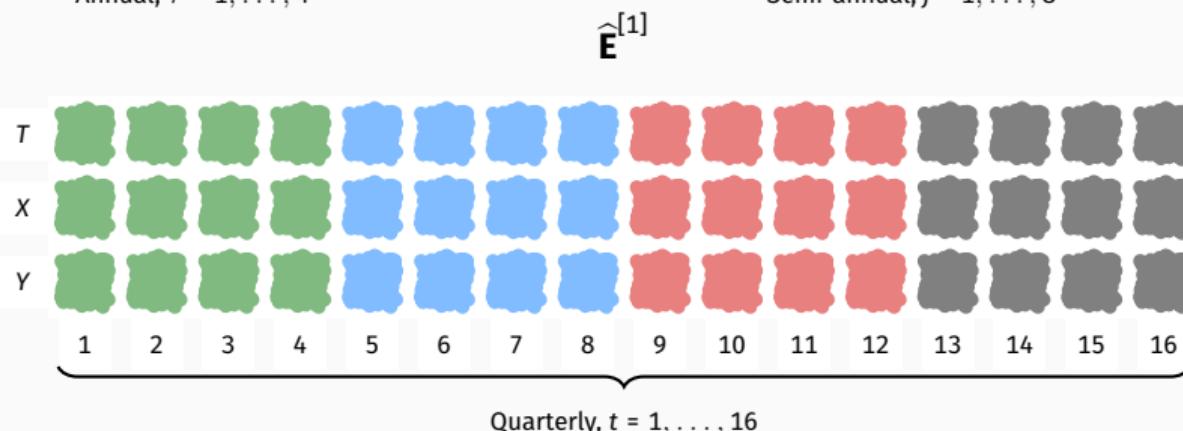
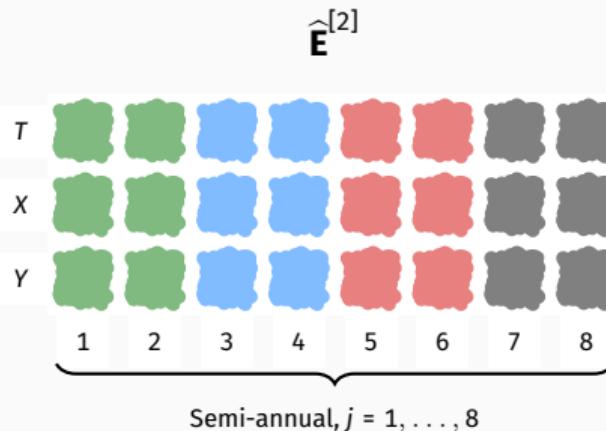
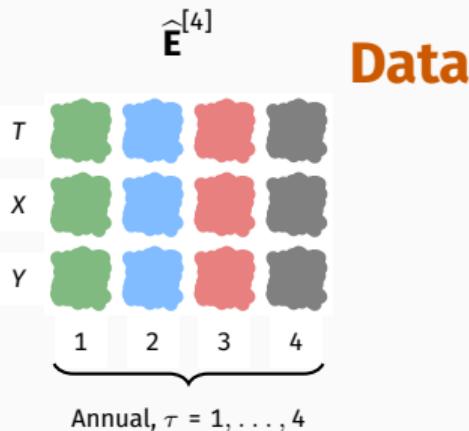
- 1 Temporal reconciliation
- 2 Temporal reconciliation: thief package
- 3 Temporal reconciliation: Examples
- 4 Cross-temporal forecast reconciliation
- 5 Cross-temporal probabilistic forecast reconciliation
- 6 Cross-temporal probabilistic reconciliation: Examples
- 7 Software

Cross-temporal probabilistic forecast reconciliation

Nonparametric bootstrap

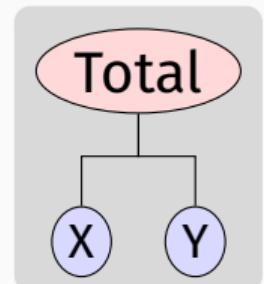
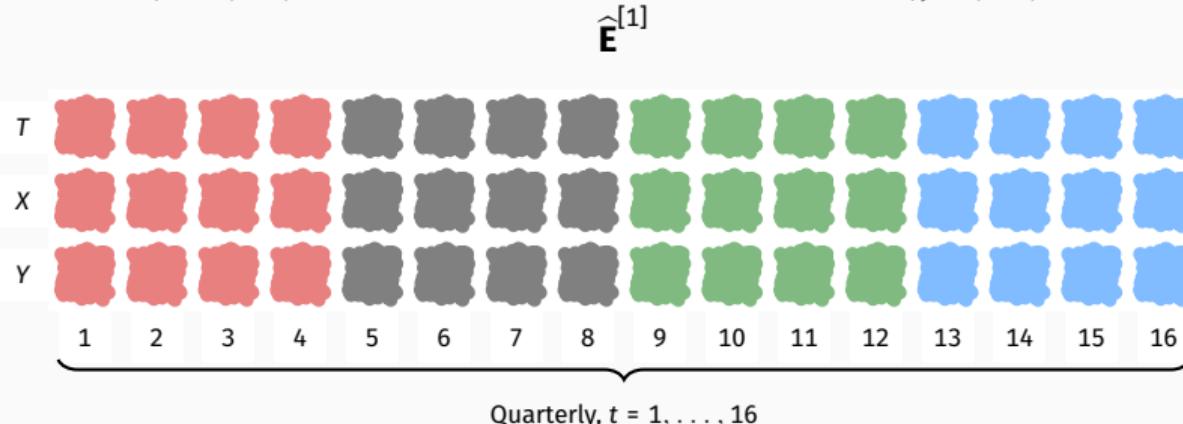
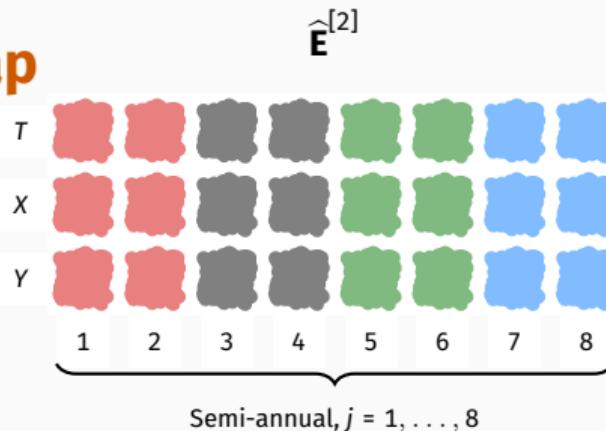
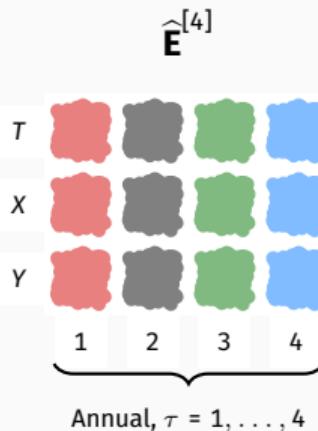
- Simulate future sample paths from all models using bootstrapped residuals, then reconcile them to obtain coherent sample paths.
- Need to generate samples that preserve cross-temporal relationships.
- Draw residual samples of all series at same time from most temporally aggregated level.
- Residuals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

Cross-temporal probabilistic forecast reconciliation



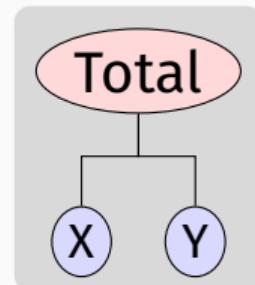
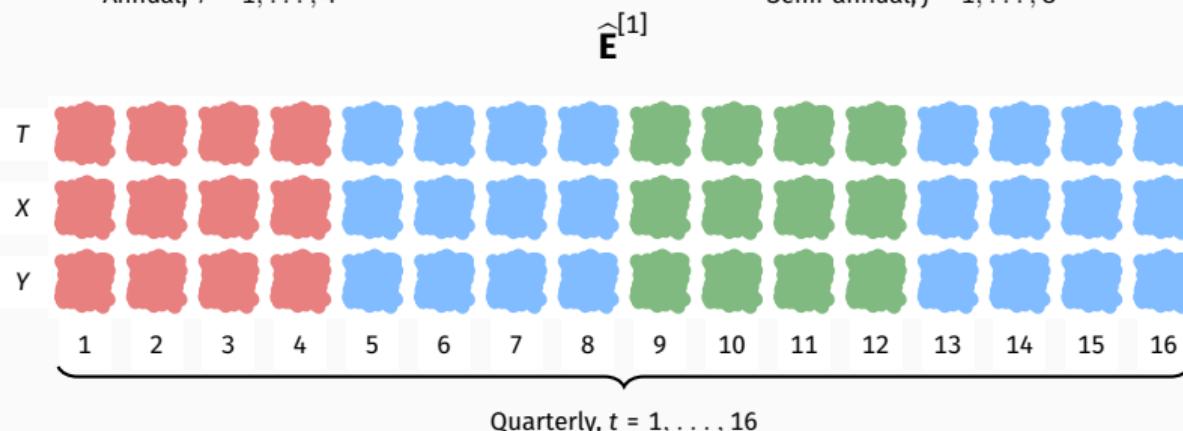
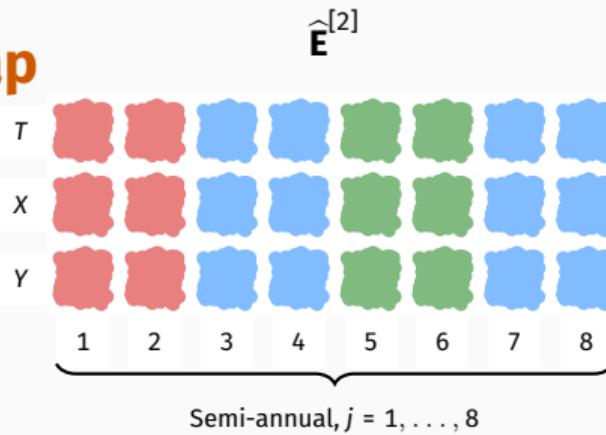
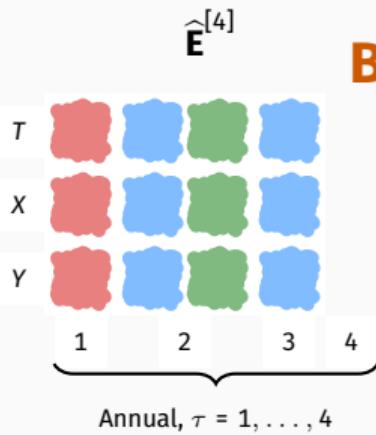
Year 1
Year 2
Year 3
Year 4

Cross-temporal probabilistic forecast reconciliation



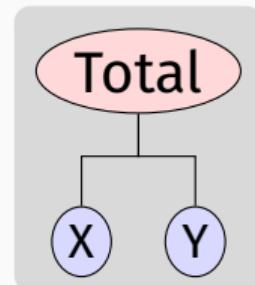
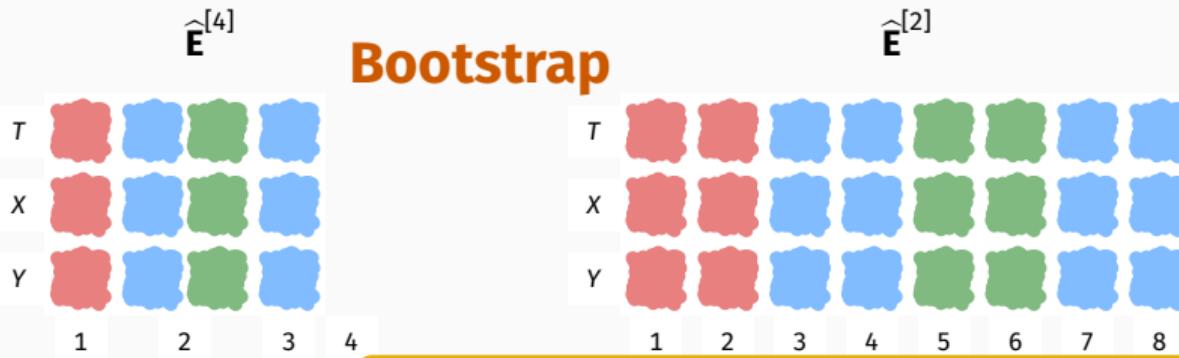
Year 1
Year 2
Year 3
Year 4

Cross-temporal probabilistic forecast reconciliation

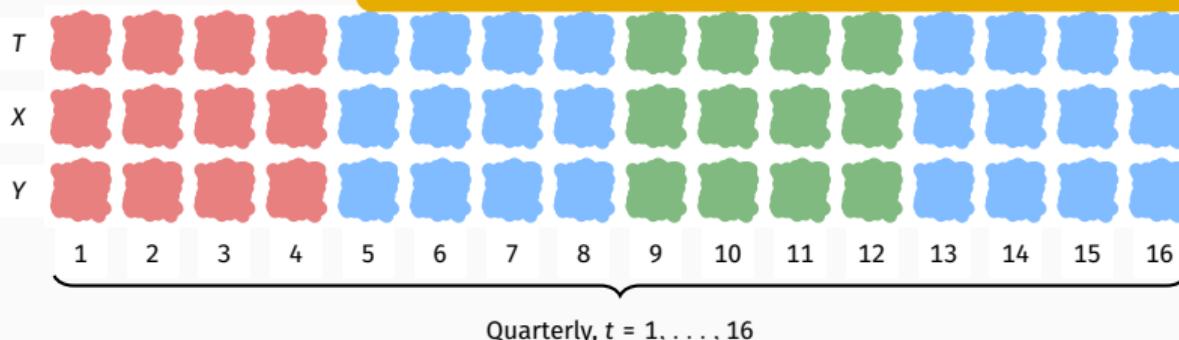


Year 1
Year 2
Year 3
Year 4

Cross-temporal probabilistic forecast reconciliation



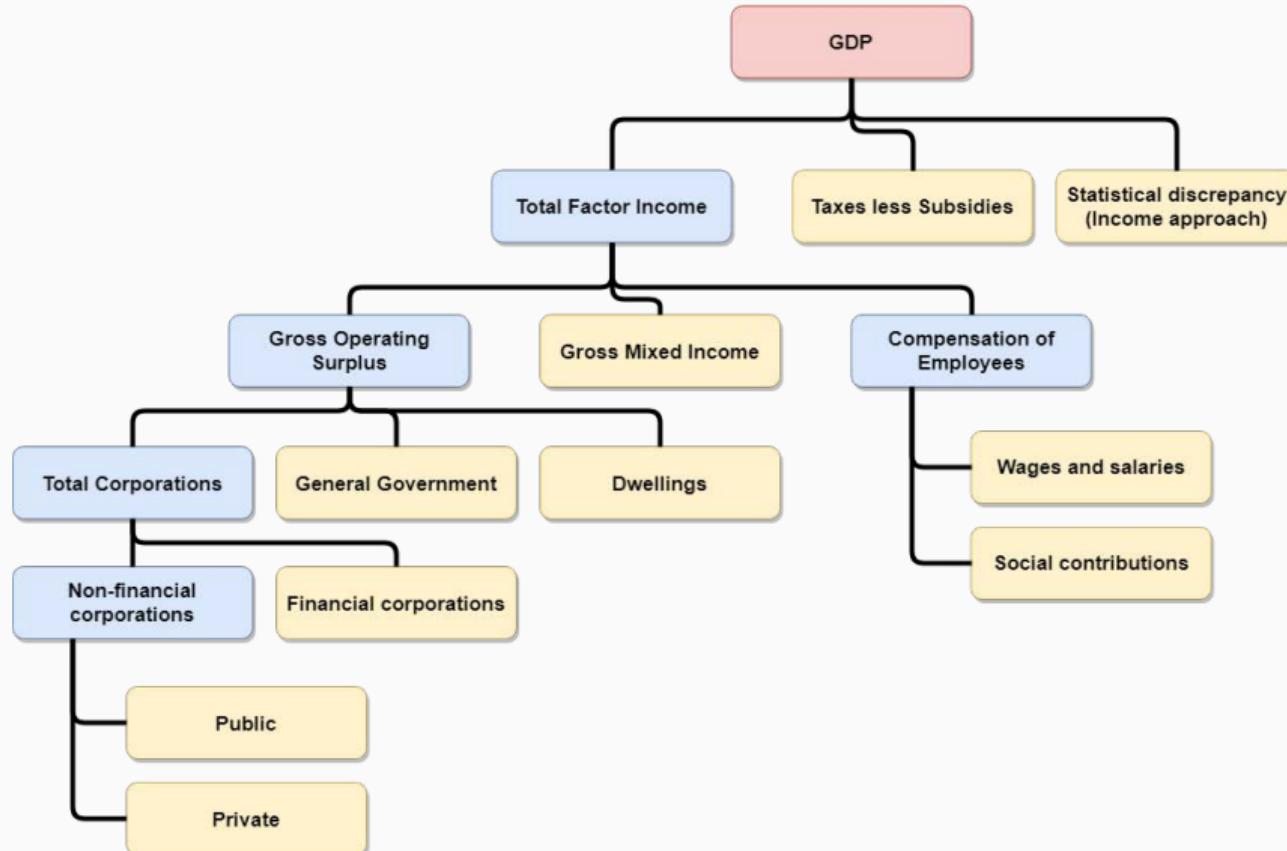
The “year” can start in any quarter,
giving overlapping blocks.



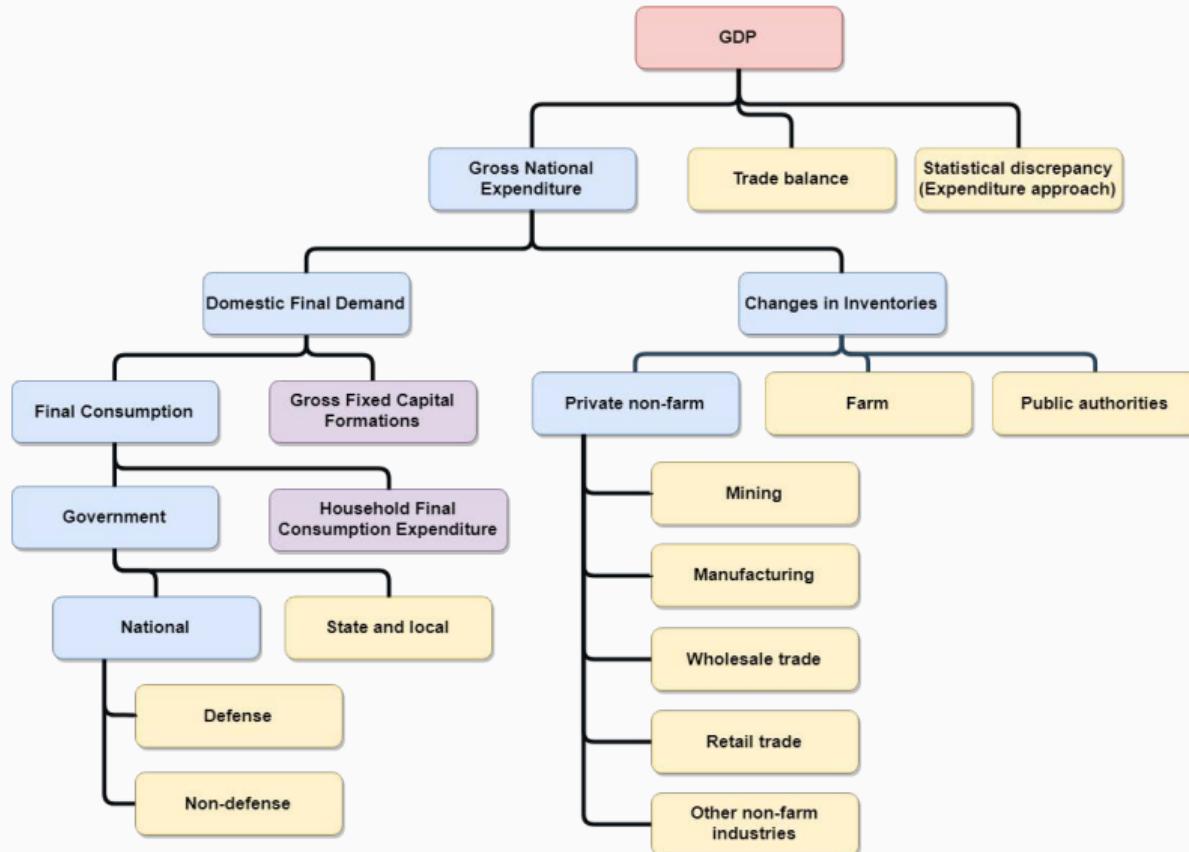
Outline

- 1 Temporal reconciliation
- 2 Temporal reconciliation: thief package
- 3 Temporal reconciliation: Examples
- 4 Cross-temporal forecast reconciliation
- 5 Cross-temporal probabilistic forecast reconciliation
- 6 Cross-temporal probabilistic reconciliation: Examples
- 7 Software

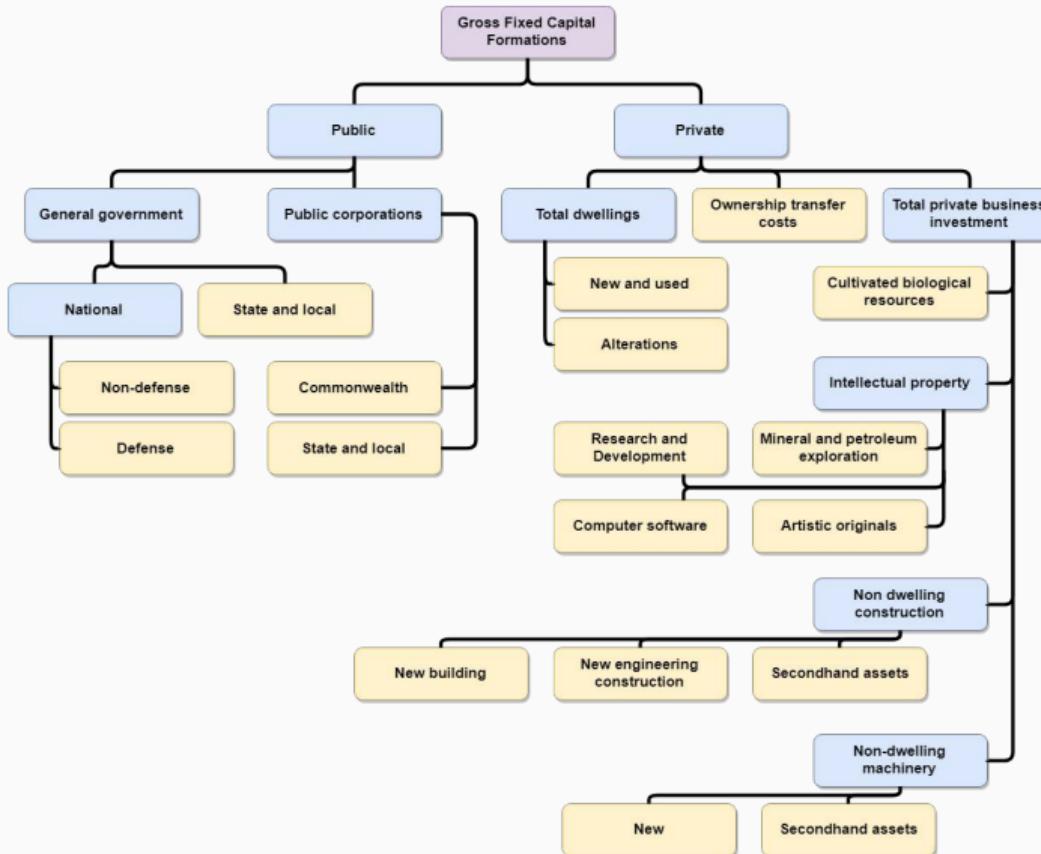
Quarterly Australian GDP data



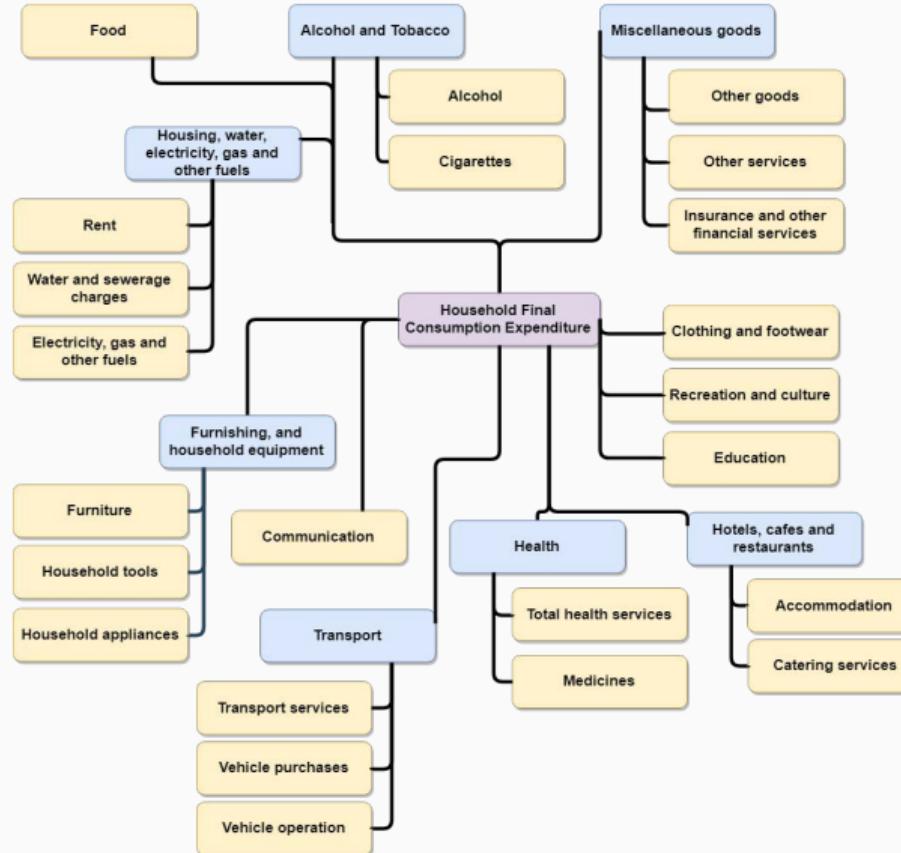
Quarterly Australian GDP data



Quarterly Australian GDP data



Quarterly Australian GDP data



Quarterly Australian GDP data

- Quarterly data: 1984:Q4 – 2018:Q1
- 95 series
- Time series cross-validation; initial training set of 10 years.
- One-quarter increase in each training set
- For each training set, compute temporally aggregated series for $k \in \{1, 2, 4\}$, and produce forecasts up to $h_1 = 4$, $h_2 = 2$, and $h_4 = 1$ steps ahead.
- Automatic ARIMA forecasts for all series.

Quarterly Australian GDP data – CRPS skill scores

Reconciliation approaches

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

$ct(shr_{cs}, bu_{te}) \quad | \quad ct(wlsv_{te}, bu_{cs}) \quad |$

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

$oct_o(wlsv) \quad | \quad oct_o(bdshr) \quad |$

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

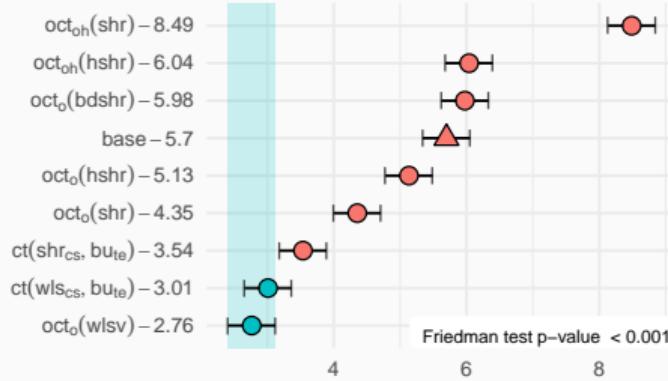
$oct_{oh}(hshr) \quad | \quad oct_{oh}(shr) \quad |$

Quarterly Australian GDP data – CRPS skill scores

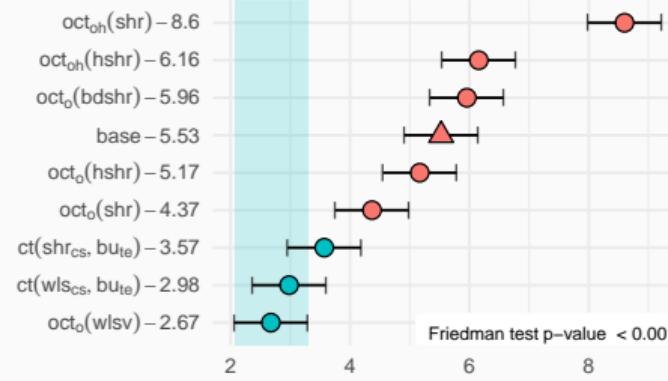
	Worse than benchmark	Best			
Reconciliation approach	$\forall k \in \{4, 2, 1\}$		$k = 1$	$k = 2$	$k = 4$
base		1.000	1.000	1.000	1.000
$ct(shr_{cs}, bu_{te})$		0.937	0.992	0.949	0.874
$ct(wls_{cs}, bu_{te})$		0.930	0.986	0.942	0.866
$oct_o(wlsv)$		0.926	0.984	0.938	0.860
$oct_o(bdshr)$		0.978	1.034	0.991	0.914
$oct_{oh}(shr)$		1.102	1.172	1.120	1.020
$oct_{oh}(hshr)$		1.006	1.068	1.021	0.934

Quarterly Australian GDP data: Nemenyi test

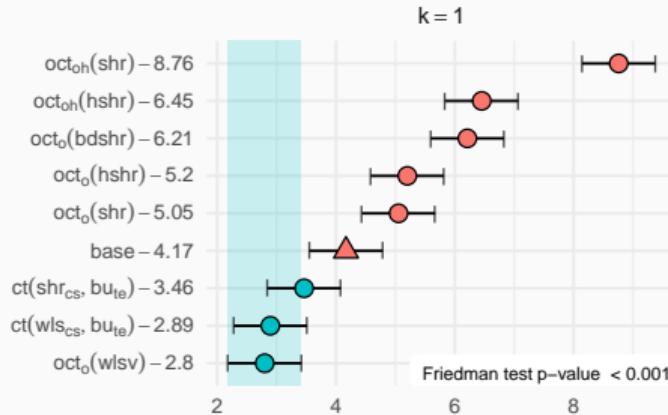
$k \in \{4, 2, 1\}$



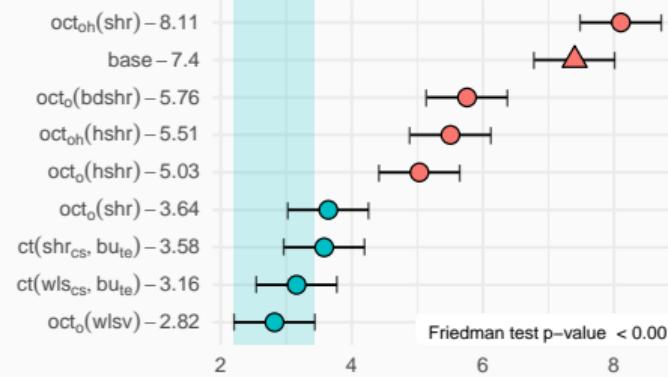
$k = 2$



$k = 1$

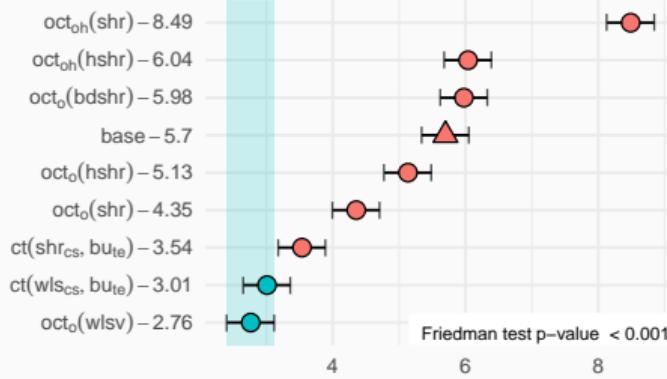


$k = 4$

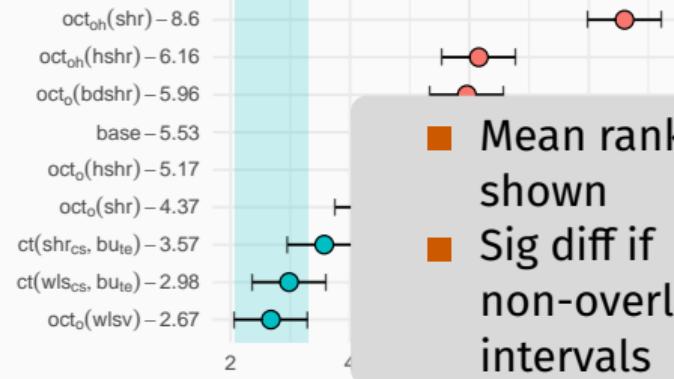


Quarterly Australian GDP data: Nemenyi test

$$k \in \{4, 2, 1\}$$

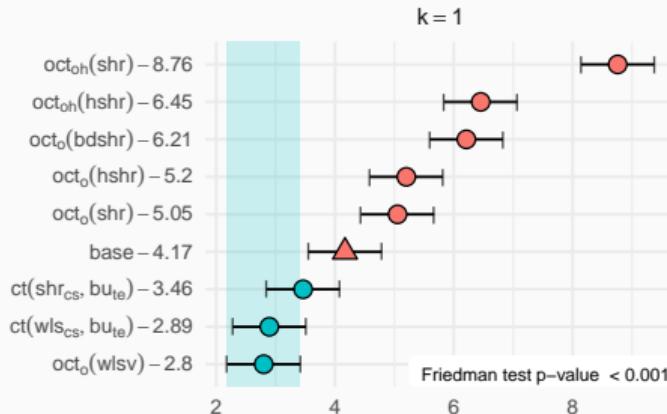


$$k = 2$$

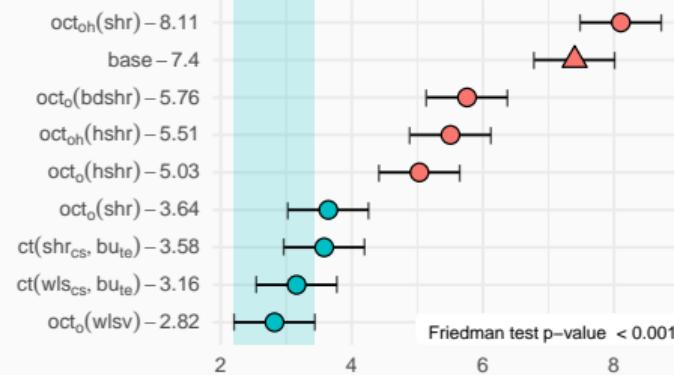


- Mean ranks shown
- Sig diff if non-overlapping intervals

$$k = 1$$

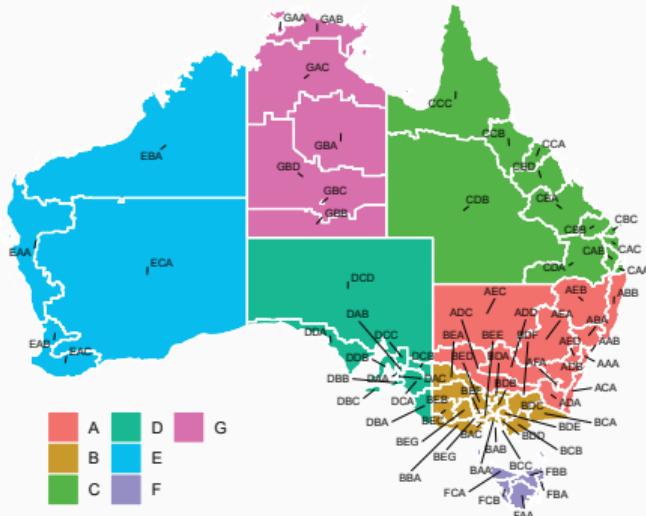


$$k = 4$$



Monthly Australian Tourism Demand

Geographical division



Grouped ts

(geographical divisions \times purpose of travel)

	AUS	States	Zones*	Regions	Tot
g.d.	1	7	21	76	105
p.o.t.	4	28	84	304	420
Tot	5	35	105	380	525

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



Purpose of travel

Holiday, Visiting friends & relatives, Business, Other

Temporal framework, frequencies:

- ▶ Monthly
- ▶ Bi-Monthly
- ▶ Quarterly
- ▶ Four-Monthly
- ▶ Semi-Annual
- ▶ Annual

Example: Monthly Australian Tourism Demand

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

Example: Australian GDP data – CRPS skill scores

Reconciliation approaches

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

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

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

$oct(ols)$ | $oct(struc)$ | $oct(wlsv)$ | $oct(bdshr)$

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

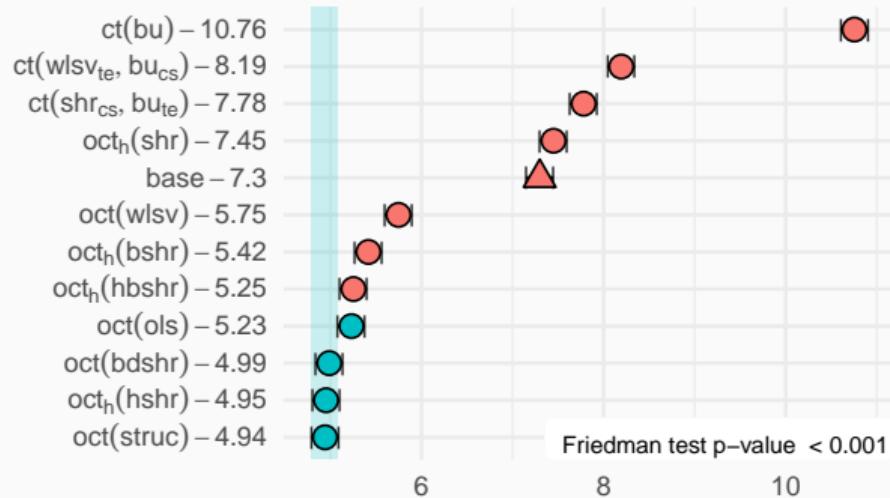
$oct_h(hbshr)$ | $oct_h(bshr)$ | $oct_h(hshr)$ | $oct_h(shr)$

Example: Australian tourism data – CRPS skill scores

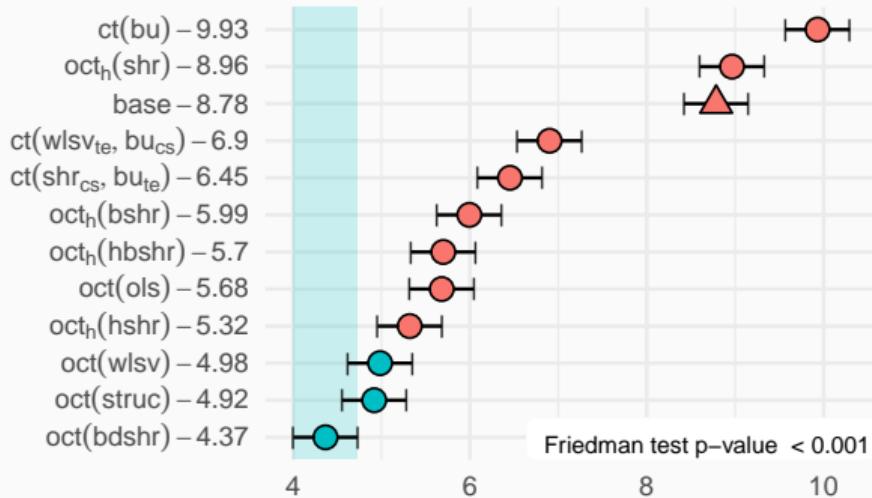
	Worse than benchmark	Best
	$\forall k \in \{12, 6, 4, 3, 2, 1\}$	$k = 1$
base	1.000	1.000
ct(bu)	1.321	1.077
ct(shr _{cs} , bu _{te})	1.057	0.976
ct(wlsv _{te} , bu _{cs})	1.062	0.976
oct(ols)	0.989	0.982
oct(struc)	0.982	0.970
oct(wlsv)	0.987	0.952
oct(bdshr)	0.975	0.949
oct _h (hbshr)	0.989	0.982
oct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

MCB Nemenyi test

$$k \in \{12, 6, 4, 3, 2, 1\}$$



$$k = 1$$



- mean ranks shown
- significant differences if non-overlapping intervals

Outline

- 1 Temporal reconciliation
- 2 Temporal reconciliation: thief package
- 3 Temporal reconciliation: Examples
- 4 Cross-temporal forecast reconciliation
- 5 Cross-temporal probabilistic forecast reconciliation
- 6 Cross-temporal probabilistic reconciliation: Examples
- 7 Software

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/fr2023

 robjhyndman.com

 @robjhyndman

 aus.social/@robjhyndman

 @robjhyndman

 rob.hyndman@monash.edu

References

-  Athanasopoulos, G, RJ Hyndman, N Kourentzes, and F Petropoulos (2017). Forecasting with temporal hierarchies. *European J Operational Research* 262(1), 60–74.
-  Di Fonzo, T and D Girolimetto (2023). Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives. *International Journal of Forecasting* 39(1), 39–57.
-  Girolimetto, D, G Athanasopoulos, T Di Fonzo, and RJ Hyndman (2023). “Cross-temporal probabilistic forecast reconciliation”. robjhyndman.com/publications/ctprob.html.