3.3 Hierarchical forecasting
1. Hierarchical and grouped time series
2. hts package for R
3. Application: Australian tourism
4. Optimal forecast reconciliation
5. Lab Session 22
6. Temporal hierarchies
7. Lab session 23
ATC drug classification

A  Alimentary tract and metabolism
B  Blood and blood forming organs
C  Cardiovascular system
D  Dermatologicals
G  Genito-urinary system and sex hormones
H  Systemic hormonal preparations, excluding sex hormones and insulins
J  Anti-infectives for systemic use
L  Antineoplastic and immunomodulating agents
M  Musculo-skeletal system
N  Nervous system
P  Antiparasitic products, insecticides and repellents
R  Respiratory system
S  Sensory organs
V  Various
ATC drug classification

14 classes

A

Alimentary tract and metabolism

84 classes

A10

Drugs used in diabetes

A10B

Blood glucose lowering drugs

A10BA

Biguanides

A10BA02

Metformin
Australian tourism

- Quarterly data on visitor night from 1998:Q1 – 2013:Q4
- From: *National Visitor Survey*, based on annual interviews of 120,000 Australians aged 15+, collected by Tourism Research Australia.
- Split by 7 states, 27 zones and 76 regions (a geographical hierarchy)
- Also split by purpose of travel:
  - Holiday
  - Visiting friends and relatives (VFR)
  - Business
  - Other
- 304 bottom-level series
Spectacle sales

- Monthly UK sales data from 2000 – 2014
- Provided by a large spectacle manufacturer
- Split by brand (26), gender (3), price range (6), materials (4), and stores (600)
- About 1 million bottom-level series
A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.
A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

**Examples**

- Pharmaceutical sales
- Tourism demand by state and region
A grouped time series is a collection of time series that can be grouped together in a number of non-hierarchical ways.
A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.

**Examples**

- Spectacle sales by brand, gender, stores, etc.
- Tourism by state and purpose of travel
The problem

1. How to forecast time series at all nodes such that the forecasts add up in the same way as the original data?
2. Can we exploit relationships between the series to improve the forecasts?

The solution

1. Forecast all series at all levels of aggregation using an automatic forecasting algorithm. (e.g., ets, auto.arima, ...)
2. Reconcile the resulting forecasts so they add up correctly using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).

This is available in the hts package in R.
The problem

1. How to forecast time series at all nodes such that the forecasts add up in the same way as the original data?
2. Can we exploit relationships between the series to improve the forecasts?

The solution

1. Forecast all series at all levels of aggregation using an automatic forecasting algorithm.
   (e.g., ets, auto.arima, ...)
2. Reconcile the resulting forecasts so they add up correctly using least squares optimization (i.e., find closest reconciled forecasts to the original forecasts).
3. This is available in the hts package in R.
1 Hierarchical and grouped time series
2 hts package for R
3 Application: Australian tourism
4 Optimal forecast reconciliation
5 Lab Session 22
6 Temporal hierarchies
7 Lab session 23
hts: Hierarchical and Grouped Time Series

Methods for analysing and forecasting hierarchical and grouped time series

Version: 5.1.5
Depends: R (≥ 3.2.0), forecast (≥ 8.1)
Imports: SparseM, Matrix, matrixcalc, parallel, utils, methods, graphics, grDevices
LinkingTo: Rcpp (≥ 0.11.0), RcppEigen
Suggests: testthat, knitr, rmarkdown
Published: 2018-03-26
Author: Rob J Hyndman, Alan Lee, Earo Wang, Shanika Wickramasuriya
Maintainer: Rob J Hyndman <Earo.Wang at gmail.com>
BugReports: https://github.com/earowang/hts/issues
License: GPL (≥ 2)
URL: http://pkg.earo.me/hts
Example using R

```r
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))
```
Example using R

```r
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))
```

```
Total

A

AX AY AZ

B

BX BY
```
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))

# Forecast 10-step-ahead using WLS combination method
# ETS used for each series by default
fc <- forecast(y, h=10)
forecast.gts() function

Usage

forecast(object, h,
    method = c("comb", "bu", "mo", "tdgsf", "tdgsa", "tdfp"),
    fmethod = c("ets", "rw", "arima"),
    weights = c("wls", "ols", "mint", "nseries"),
    covariance = c("shr","sam"),
    positive = TRUE,
    parallel = FALSE, num.cores = 2, ...)

Arguments

object        Hierarchical time series object of class gts.
h            Forecast horizon
method        Method for distributing forecasts within the hierarchy.
fmethod       Forecasting method to use
weights       Weights used for "optimal combination" method.
covariance   Shrinkage estimator or sample estimator for GLS covariance.
positive      If TRUE, forecasts are forced to be strictly positive
parallel      If TRUE, allow parallel processing
num.cores     If parallel = TRUE, specify how many cores to be used
Outline

1. Hierarchical and grouped time series
2. hts package for R
3. Application: Australian tourism
4. Optimal forecast reconciliation
5. Lab Session 22
6. Temporal hierarchies
7. Lab session 23
Australian tourism
Domestic visitor nights
From: *National Visitor Survey*, based on annual interviews of 120,000 Australians aged 15+, collected by Tourism Research Australia.

- < -3%
- -3% to 0
- 0 to 3%
- > 3%
Australian tourism

Hierarchy:
- States (7)
- Zones (27)
- Regions (82)
Australian tourism

Hierarchy:
- States (7)
- Zones (27)
- Regions (82)

Base forecasts
ETS (exponential smoothing) models
Base forecasts

Domestic tourism forecasts: Total

<table>
<thead>
<tr>
<th>Year</th>
<th>Visitor nights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td>60000</td>
</tr>
<tr>
<td>2000</td>
<td>65000</td>
</tr>
<tr>
<td>2002</td>
<td>70000</td>
</tr>
<tr>
<td>2004</td>
<td>75000</td>
</tr>
<tr>
<td>2006</td>
<td>80000</td>
</tr>
<tr>
<td>2008</td>
<td>85000</td>
</tr>
</tbody>
</table>
Base forecasts

**Domestic tourism forecasts: NSW**

<table>
<thead>
<tr>
<th>Year</th>
<th>Visitor nights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td>18000</td>
</tr>
<tr>
<td>2000</td>
<td>22000</td>
</tr>
<tr>
<td>2002</td>
<td>26000</td>
</tr>
<tr>
<td>2004</td>
<td>30000</td>
</tr>
<tr>
<td>2006</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
</tr>
</tbody>
</table>
Base forecasts

Domestic tourism forecasts: VIC

Visitor nights

Year


10000 12000 14000 16000 18000
Base forecasts

Domestic tourism forecasts: Nth.Coast.NSW

Visitor nights
5000 6000 7000 8000 9000
Base forecasts

Domestic tourism forecasts: Metro. QLD

<table>
<thead>
<tr>
<th>Year</th>
<th>Visitor nights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td>8000</td>
</tr>
<tr>
<td>2000</td>
<td>9000</td>
</tr>
<tr>
<td>2002</td>
<td>11000</td>
</tr>
<tr>
<td>2004</td>
<td>13000</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Year
Base forecasts

Domestic tourism forecasts: Sth.WA

<table>
<thead>
<tr>
<th>Year</th>
<th>Visitor nights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
</tr>
</tbody>
</table>

Year

Visitor nights


400 600 800 1000 1200 1400
Base forecasts

Domestic tourism forecasts: X201.Melbourne

Visitor nights

Year


4000 4500 5000 5500 6000
Base forecasts

Domestic tourism forecasts: X402.Murraylands

Visitor nights

Year

Base forecasts

Domestic tourism forecasts: X809.Daly

Visitor nights

Year


0 20 40 60 80 100
Reconciled forecasts
Reconciled forecasts
Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-step-ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.
Forecast evaluation

**Training sets**

**Test sets** $h = 1$

→ time
Forecast evaluation

Training sets

Test sets $h = 1$

$\rightarrow$ time
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets  Test sets $h = 1$

- Time
Forecast evaluation

Training sets  Test sets $h = 1$

- Time
Forecast evaluation

Training sets  Test sets $h = 1$

\[ \rightarrow \text{time} \]
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

**Training sets**

**Test sets** $h = 1$
Forecast evaluation

Training sets

Test sets $h = 1$

$\rightarrow$ time
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets  Test sets $h = 1$

$\rightarrow$ time

$\rightarrow$
Forecast evaluation

Training sets

Test sets $h = 1$

\rightarrow time
Forecast evaluation

Training sets | Test sets $h = 1$

\[ \begin{array}{c}
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\cdot \cdot \cdot \cdot \\
\end{array} \]
Forecast evaluation

Training sets  Test sets $h = 1$

$\rightarrow$ time
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 1$

→ time
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 1$

$\rightarrow$ time
Forecast evaluation

Training sets  Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 2$
Forecast evaluation

Training sets

Test sets $h = 3$
Forecast evaluation

Training sets

Test sets $h = 4$
Forecast evaluation

Training sets

Test sets $h = 5$
Forecast evaluation

Training sets

Test sets $h = 6$

time
### Hierarchy: states, zones, regions

#### Forecast horizon

<table>
<thead>
<tr>
<th>RMSE</th>
<th>$h = 1$</th>
<th>$h = 2$</th>
<th>$h = 3$</th>
<th>$h = 4$</th>
<th>$h = 5$</th>
<th>$h = 6$</th>
<th>Ave</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Australia</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>1762.04</td>
<td>1770.29</td>
<td>1766.02</td>
<td>1818.82</td>
<td>1705.35</td>
<td>1721.17</td>
<td><strong>1757.28</strong></td>
</tr>
<tr>
<td>Bottom</td>
<td>1736.92</td>
<td>1742.69</td>
<td>1722.79</td>
<td>1752.74</td>
<td>1666.73</td>
<td>1687.43</td>
<td><strong>1718.22</strong></td>
</tr>
<tr>
<td>WLS</td>
<td>1705.21</td>
<td>1715.87</td>
<td><strong>1703.75</strong></td>
<td>1729.56</td>
<td>1627.79</td>
<td>1661.24</td>
<td><strong>1690.57</strong></td>
</tr>
<tr>
<td>GLS</td>
<td>1704.64</td>
<td>1715.60</td>
<td>1705.31</td>
<td>1729.04</td>
<td>1626.36</td>
<td>1661.64</td>
<td><strong>1690.43</strong></td>
</tr>
<tr>
<td><strong>States</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>399.77</td>
<td>404.16</td>
<td>401.92</td>
<td>407.26</td>
<td>395.38</td>
<td>401.17</td>
<td><strong>401.61</strong></td>
</tr>
<tr>
<td>Bottom</td>
<td>404.29</td>
<td>406.95</td>
<td>404.96</td>
<td>409.02</td>
<td>399.80</td>
<td>401.55</td>
<td><strong>404.43</strong></td>
</tr>
<tr>
<td>WLS</td>
<td>398.84</td>
<td>402.12</td>
<td><strong>400.71</strong></td>
<td>405.03</td>
<td>394.76</td>
<td>398.23</td>
<td><strong>399.95</strong></td>
</tr>
<tr>
<td>GLS</td>
<td>398.84</td>
<td>402.16</td>
<td>400.86</td>
<td>405.03</td>
<td>394.59</td>
<td>398.22</td>
<td><strong>399.95</strong></td>
</tr>
<tr>
<td><strong>Regions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>93.15</td>
<td>93.38</td>
<td>93.45</td>
<td>93.79</td>
<td>93.50</td>
<td>93.56</td>
<td><strong>93.47</strong></td>
</tr>
<tr>
<td>Bottom</td>
<td>93.15</td>
<td>93.38</td>
<td>93.45</td>
<td>93.79</td>
<td>93.50</td>
<td>93.56</td>
<td><strong>93.47</strong></td>
</tr>
<tr>
<td>WLS</td>
<td>93.02</td>
<td>93.32</td>
<td>93.38</td>
<td>93.72</td>
<td>93.39</td>
<td>93.53</td>
<td><strong>93.39</strong></td>
</tr>
<tr>
<td>GLS</td>
<td>92.98</td>
<td>93.27</td>
<td>93.34</td>
<td>93.66</td>
<td>93.34</td>
<td>93.46</td>
<td><strong>93.34</strong></td>
</tr>
</tbody>
</table>
1 Hierarchical and grouped time series
2 hts package for R
3 Application: Australian tourism
4 Optimal forecast reconciliation
5 Lab Session 22
6 Temporal hierarchies
7 Lab session 23
Hierarchical time series
Hierarchical time series

\( y_t \): observed aggregate of all series at time \( t \).

\( y_{X,t} \): observation on series \( X \) at time \( t \).

\( b_t \): vector of all series at bottom level in time \( t \).
Hierarchical time series

\[ y_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix} \]

- \( y_t \): observed aggregate of all series at time \( t \).
- \( y_{X,t} \): observation on series \( X \) at time \( t \).
- \( b_t \): vector of all series at bottom level in time \( t \).
Hierarchical time series

\[ y_t : \text{observed aggregate of all series at time } t. \]

\[ y_{X,t} : \text{observation on series } X \text{ at time } t. \]

\[ b_t : \text{vector of all series at bottom level in time } t. \]

\[ y_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} = Sb_t \]

\[ y_t = Sb_t \]
Hierarchical time series
Hierarchical time series

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

$$= S b_t$$
Hierarchical time series

\[ \mathbf{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} \]

= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}

\[ \mathbf{b}_t = \begin{pmatrix} \mathbf{y}_{AX,t} \\ \mathbf{y}_{AY,t} \\ \mathbf{y}_{AZ,t} \\ \mathbf{y}_{BX,t} \\ \mathbf{y}_{BY,t} \\ \mathbf{y}_{BZ,t} \\ \mathbf{y}_{CX,t} \\ \mathbf{y}_{CY,t} \\ \mathbf{y}_{CZ,t} \end{pmatrix} \]

\[ \mathbf{y}_t = \mathbf{Sb}_t \]
Grouped time series

\[
\begin{bmatrix}
A_{X,t} & A_{Y,t} & A \\
B_{X,t} & B_{Y,t} & B \\
X_{t} & Y_{t} & \text{Total}
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
### Grouped time series

The relationship between the grouped time series \( y_t \) and \( b_t \) can be expressed as:

\[
y_t = \begin{pmatrix}
y_{t,1} \\
y_{A,t} \\
y_{B,t} \\
y_{X,t} \\
y_{Y,t} \\
y_{AX,t} \\
y_{AY,t} \\
y_{BX,t} \\
y_{BY,t}
\end{pmatrix} = \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
y_{AX,t} \\
y_{AY,t} \\
y_{BX,t} \\
y_{BY,t}
\end{pmatrix} = \begin{pmatrix} s \end{pmatrix} \begin{pmatrix} b_t \end{pmatrix}
\]

Here, \( X \) and \( Y \) represent the categories, and \( A \) and \( B \) are the groups.
Grouped time series

\[
y_t = \begin{pmatrix} y_t \\
y_{A,t} \\
y_{B,t} \\
y_{X,t} \\
y_{Y,t} \\
y_{AX,t} \\
y_{AY,t} \\
y_{BX,t} \\
y_{BY,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AX,t} \\
y_{AY,t} \\
y_{BX,t} \\
y_{BY,t} \end{pmatrix} = S b_t
\]

\[
y_t = S b_t
\]
Hierarchical and grouped time series

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

\[ y_t = Sb_t \]

where

- \( y_t \) is a vector of all series at time \( t \)
- \( b_t \) is a vector of the most disaggregated series at time \( t \)
- \( S \) is a “summing matrix” containing the aggregation constraints.
Let $\hat{y}_n(h)$ be vector of initial $h$-step forecasts, made at time $n$, stacked in same order as $y_t$. 
Let $\hat{y}_n(h)$ be vector of initial $h$-step forecasts, made at time $n$, stacked in same order as $y_t$. (In general, they will not “add up”.)
Forecasting notation

Let $\hat{y}_n(h)$ be vector of initial $h$-step forecasts, made at time $n$, stacked in same order as $y_t$. (In general, they will not “add up”.)

Reconciled forecasts must be of the form:

$$\tilde{y}_n(h) = SP\hat{y}_n(h)$$

for some matrix $P$. 

Forecasting notation

Let $\hat{y}_n(h)$ be vector of initial $h$-step forecasts, made at time $n$, stacked in same order as $y_t$. (In general, they will not “add up”.)

Reconciled forecasts must be of the form:

$$\tilde{y}_n(h) = SP\hat{y}_n(h)$$

for some matrix $P$.

- $P$ extracts and combines base forecasts $\hat{y}_n(h)$ to get bottom-level forecasts.
- $S$ adds them up
Main result

The best (minimum sum of variances) unbiased forecasts are obtained when

$$P = (S' \Sigma_h^{-1} S)^{-1} S' \Sigma_h^{-1},$$

where $\Sigma_h$ is the $h$-step base forecast error covariance matrix.
Optimal combination forecasts

Main result

The best (minimum sum of variances) unbiased forecasts are obtained when $P = (S' \Sigma^{-1}_h S) S' \Sigma^{-1}_h$, where $\Sigma_h$ is the $h$-step base forecast error covariance matrix.

$$\tilde{y}_n(h) = S (S' \Sigma^{-1}_h S)^{-1} S' \Sigma^{-1}_h \hat{y}_n(h)$$

Problem: $\Sigma_h$ hard to estimate, especially for $h > 1$.

Solutions:
- Ignore $\Sigma_h$ (OLS)
- Assume $\Sigma_h$ diagonal (WLS) [Default in hts]
- Try to estimate $\Sigma_h$ (GLS)
Features

- Covariates can be included in initial forecasts.
- Adjustments can be made to initial forecasts at any level.
- Very simple and flexible method. Can work with any hierarchical or grouped time series.
- Conceptually easy to implement: regression of base forecasts on structure matrix.
Lab Session 22
1. Hierarchical and grouped time series
2. hts package for R
3. Application: Australian tourism
4. Optimal forecast reconciliation
5. Lab Session 22
6. Temporal hierarchies
7. Lab session 23
Temporal hierarchies

Basic idea:
Forecast series at each available frequency.
Optimally reconcile forecasts within the same year.
Temporal hierarchies

Basic idea:
- Forecast series at each available frequency.
- Optimally reconcile forecasts within the same year.
Monthly series

- $k = 2, 4, 12$ nodes
Monthly series

- $k = 2, 4, 12$ nodes
- $k = 3, 6, 12$ nodes
Monthly series

- $k = 2, 4, 12$ nodes
- $k = 3, 6, 12$ nodes
- **Why not $k = 2, 3, 4, 6, 12$ nodes?**
**Monthly data**

\[
\begin{pmatrix}
A \\
SemiA_1 \\
SemiA_2 \\
FourM_1 \\
FourM_2 \\
FourM_3 \\
Q_1 \\
\vdots \\
Q_4 \\
BiM_1 \\
\vdots \\
BiM_6 \\
M_1 \\
\vdots \\
M_{12}
\end{pmatrix}
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
= 
\begin{pmatrix}
M_1 \\
M_2 \\
M_3 \\
M_4 \\
M_5 \\
M_6 \\
M_7 \\
M_8 \\
M_9 \\
M_{10} \\
M_{11} \\
M_{12}
\end{pmatrix}
\begin{pmatrix}
12 \\
\vdots \\
12
\end{pmatrix}
\begin{pmatrix}
b_t
\end{pmatrix}
\]
In general

For a time series \( y_1, \ldots, y_T \), observed at frequency \( m \), we generate aggregate series

\[
y_j^{[k]} = \sum_{t=1+(j-1)k}^{jk} y_t, \quad \text{for } j = 1, \ldots, \left\lfloor \frac{T}{k} \right\rfloor
\]

- \( k \in F(m) = \{ \text{factors of } m \} \).
- A single unique hierarchy is only possible when there are no coprime pairs in \( F(m) \).
- \( M_k = m/k \) is seasonal period of aggregated series.
UK Accidents and Emergency Demand

Annual (k=52)

Semi-annual (k=26)

Quarterly (k=13)

Monthly (k=4)

Bi-weekly (k=2)

Weekly (k=1)

- - - - base

--- reconciled
### UK Accidents and Emergency Demand

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Type 1 Departments — Major A&amp;E</td>
</tr>
<tr>
<td>2</td>
<td>Type 2 Departments — Single Specialty</td>
</tr>
<tr>
<td>3</td>
<td>Type 3 Departments — Other A&amp;E/Minor Injury</td>
</tr>
<tr>
<td>4</td>
<td>Total Attendances</td>
</tr>
<tr>
<td>5</td>
<td>Type 1 Departments — Major A&amp;E &gt; 4 hrs</td>
</tr>
<tr>
<td>6</td>
<td>Type 2 Departments — Single Specialty &gt; 4 hrs</td>
</tr>
<tr>
<td>7</td>
<td>Type 3 Departments — Other A&amp;E/Minor Injury &gt; 4 hrs</td>
</tr>
<tr>
<td>8</td>
<td>Total Attendances &gt; 4 hrs</td>
</tr>
<tr>
<td>9</td>
<td>Emergency Admissions via Type 1 A&amp;E</td>
</tr>
<tr>
<td>10</td>
<td>Total Emergency Admissions via A&amp;E</td>
</tr>
<tr>
<td>11</td>
<td>Other Emergency Admissions (i.e., not via A&amp;E)</td>
</tr>
<tr>
<td>12</td>
<td>Total Emergency Admissions</td>
</tr>
</tbody>
</table>
UK Accidents and Emergency Demand

- **Minimum training set**: all data except the last year
- Base forecasts using `auto.arima()`.
- Mean Absolute Scaled Errors for 1, 4 and 13 weeks ahead using a rolling origin.
UK Accidents and Emergency Demand

- **Minimum training set**: all data except the last year
- Base forecasts using `auto.arima()`.
- Mean Absolute Scaled Errors for 1, 4 and 13 weeks ahead using a rolling origin.

<table>
<thead>
<tr>
<th>Aggr. Level</th>
<th>$h$</th>
<th>Base</th>
<th>Reconciled</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly</td>
<td>1</td>
<td>1.6</td>
<td>1.3</td>
<td>$-17.2%$</td>
</tr>
<tr>
<td>Weekly</td>
<td>4</td>
<td>1.9</td>
<td>1.5</td>
<td>$-18.6%$</td>
</tr>
<tr>
<td>Weekly</td>
<td>13</td>
<td>2.3</td>
<td>1.9</td>
<td>$-16.2%$</td>
</tr>
<tr>
<td>Weekly</td>
<td>1−52</td>
<td>2.0</td>
<td>1.9</td>
<td>$-5.0%$</td>
</tr>
<tr>
<td>Annual</td>
<td>1</td>
<td>3.4</td>
<td>1.9</td>
<td>$-42.9%$</td>
</tr>
</tbody>
</table>
thief package for R

thief: Temporal HIERarchical Forecasting
thief package for R

thief: Temporal HIErarchical Forecasting

Install from CRAN

install.packages("thief")

Usage

library(thief)
thief(y)
Lab Session 23