3.3 Hierarchical forecasting
1 Hierarchical and grouped time series
2 Lab session 15
3 Temporal hierarchies
4 Lab session 16
Australian tourism demand

Forecasting using R

Hierarchical and grouped time series
Australian tourism demand

- 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

> 3%
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
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
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
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.

**Examples**
- Tourism by state and region
A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

**Examples**
- Tourism 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.

### Examples

- Labour turnover by occupation and state
- Spectacle sales by brand, gender, stores, etc.
A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.

Examples
- Labour turnover by occupation and state
- Spectacle sales by brand, gender, stores, etc.
A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.

**Examples**

- Labour turnover by occupation and state
- Spectacle sales by brand, gender, stores, etc.
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 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?
Forecast all series at all levels of aggregation using an automatic forecasting algorithm (e.g., ets, auto.arima, ...)

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 all available in the hts package in R.
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 all available in the hts package in R.
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 all available in the hts package in R.
hts: Hierarchical and Grouped Time Series

Methods for analysing and forecasting hierarchical and grouped time series

Version: 5.0
Depends: R (≥ 3.0.2), forecast (≥ 5.0), SparseM, Matrix, matrixcalc
Imports: parallel, utils, methods, graphics, grDevices, stats
LinkingTo: Rcpp (≥ 0.11.0), RcppEigen
Suggests: testthat
Published: 2016-04-06
Author: Rob J Hyndman, Earo Wang, Alan Lee, Shanika Wickramasuriya
Maintainer: Rob J Hyndman <Rob.Hyndman at monash.edu>
BugReports: https://github.com/robjhyndman/hts/issues
License: GPL (≥ 2)
Example using 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)))
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)))
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)
gts function

Usage

gts(y, characters)

Arguments

y Multivariate time series containing the bottom level series

characters Vector of integers, or list of vectors, showing how column names indicate group structure.

Example

bnames <-
c("VIC1F","VIC1M","VIC2F","VIC2M","VIC3F","VIC3M", "NSW1F","NSW1M","NSW2F","NSW2M","NSW3F","NSW3M")
bts <- matrix(ts(rnorm(120)), ncol = 12)
colnames(bts) <- bnames
x <- gts(bts, characters = c(3, 1, 1))
Example 2

```r
bnames <-
c("VICMelbAA","VICMelbAB",
  "VICGeelAA","VICGeelAB",
  "VICMelbBA","VICMelbBB",
  "VICGeelBA","VICGeelBB",
  "NSWSyndAA","NSWSyndAB",
  "NSWWollAA","NSWWollAB",
  "NSWSyndBA","NSWSyndBB",
  "NSWWollBA","NSWWollBB")

bts <- matrix(ts(rnorm(160)), ncol = 16)
colnames(bts) <- bnames

x <- gts(bts, characters = list(c(3, 4), c(1, 1)))
```
forecast.gts function

Usage

```r
forecast(object, h,
    method = c("comb", "bu", "mo", "tdgsa", "tdgsf", "tdfp"),
    weights = c("wls", "ols", "mint", "nseries"),
    fmethod = c("ets", "arima", "rw"),
    algorithms = c("lu", "cg", "chol", "recursive", "slm"),
    covariance = c("shr", "sam"),
    positive = FALSE,
    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.
- **weights**: Weights used for “optimal combination” method. When weights = “sd”, it takes account of the standard deviation of forecasts.
- **fmethod**: Forecasting method to use
- **algorithm**: Method for solving regression equations
- **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 are going to be used.
Example: Australian tourism

Forecasting using R Hierarchical and grouped time series
Example: Australian tourism

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

Forecasting using R
Hierarchical and grouped time series
Example: Australian tourism

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

**Base forecasts**
ETS (exponential smoothing) models
Base forecasts

Domestic tourism forecasts: Total

Year | Visitor nights
--- | ---
1998 | 60000
2000 | 65000
2002 | 70000
2004 | 75000
2006 | 80000
2008 | 85000

Year
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: Nth.Coast.NSW

Year
Visitor nights
5000 6000 7000 8000 9000
Base forecasts

Domestic tourism forecasts: Sth.WA

Visitor nights

Year

Domestic tourism forecasts: X201.Melbourne

Year
Visitor nights
4000 4500 5000 5500 6000

Forecasting using R
Hierarchical and grouped time series
Domestic tourism forecasts: X809.Daly

Visitor nights

Year


0 20 40 60 80 100
Forecast evaluation

Training sets

Test sets $h = 1$

\[ \text{time} \]
Forecast evaluation

Training sets

Test sets $h = 1$

$\rightarrow$ time
Forecast evaluation

Training sets

Test sets \( h = 1 \)

\[ \text{time} \]
Forecast evaluation

Training sets

Test sets $h = 1$

→ time
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 1$

time
Forecast evaluation

Training sets  Test sets $h = 1$

Forecasting using R
Hierarchical and grouped time series
Forecast evaluation

Training sets |
---|
| Test sets $h = 1$ |

<table>
<thead>
<tr>
<th>Training sets</th>
<th>Test sets $h = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 1$

→ time

Forecasting using R

Hierarchical and grouped time series
Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets \( h = 1 \)

\[
\begin{array}{ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc}
Forecast evaluation

Training sets

Test sets $h = 1$

Forecasting using R

Hierarchical and grouped time series
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 1$

Forecasting using R
Hierarchical and grouped time series

17
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 1$

Forecasting using R

Hierarchical and grouped time series
Forecast evaluation

Training sets

Test sets $h = 1$
Forecast evaluation

Training sets

Test sets $h = 2$

time

Forecasting using R
Hierarchical and grouped time series
17
Forecast evaluation

Training sets

Test sets $h = 3$

Forecasting using R

Hierarchical and grouped time series
Forecast evaluation

Training sets

Test sets $h = 4$
Forecast evaluation

Training sets

Test sets $h = 5$

time
Forecast evaluation

Training sets  

Test sets $h = 6$

Forecasting using R  
Hierarchical and grouped time series
## 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>OLS</td>
<td>1747.60</td>
<td>1757.68</td>
<td>1751.77</td>
<td>1800.67</td>
<td>1686.00</td>
<td>1706.45</td>
<td><strong>1741.69</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><strong>1661.24</strong></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><strong>1729.04</strong></td>
<td><strong>1626.36</strong></td>
<td><strong>1661.64</strong></td>
<td><strong>1690.43</strong></td>
</tr>
</tbody>
</table>

| **States** |
| Base  | 399.77        | 404.16        | 401.92        | 407.26        | 395.38        | 401.17        | **401.61** |
| Bottom| 404.29        | 406.95        | 404.96        | 409.02        | 399.80        | 401.55        | **404.43** |
| OLS   | 404.47        | 407.62        | 405.43        | 413.79        | 401.10        | 404.90        | **406.22** |
| WLS   | 398.84        | 402.12        | 400.71        | 405.03        | 394.76        | 398.23        | **399.95** |
| GLS   | 398.84        | 402.16        | 400.86        | 405.03        | 394.59        | 398.22        | **399.95** |

| **Regions** |
| Base  | 93.15         | 93.38         | 93.45         | 93.79         | 93.50         | 93.56         | **93.47** |
| Bottom| 93.15         | 93.38         | 93.45         | 93.79         | 93.50         | 93.56         | **93.47** |
| OLS   | 93.28         | 93.53         | 93.64         | 94.17         | 93.78         | 93.88         | **93.71** |
| WLS   | 93.02         | 93.32         | 93.38         | 93.72         | 93.39         | 93.53         | **93.39** |
| GLS   | 92.98         | 93.27         | 93.34         | 93.66         | 93.34         | 93.46         | **93.34** |
Hierarchical time series

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

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

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

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

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

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

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

\[
y_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \begin{pmatrix}
1 & 1 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
y_{A,t} \\
y_{B,t} \\
y_{C,t}
\end{pmatrix}
\]
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 $t$.

\[
\begin{pmatrix}
y_{t} \\
y_{A,t} \\
y_{B,t} \\
y_{C,t}
\end{pmatrix}' =
\begin{pmatrix}
1 & 1 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
y_{A,t} \\
y_{B,t} \\
y_{C,t}
\end{pmatrix}
\]
Hierarchical time series

\[ y_t : \text{observed aggregate of all series at time } t. \]
\[ y_{X,t} : \text{observation on series } X \text{ at time } t. \]
\[ b_t : \text{vector of all series at bottom level in time } t. \]

\[ y_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \begin{pmatrix}
1 & 1 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
y_{A,t} \\
y_{B,t} \\
y_{C,t}
\end{pmatrix}
\]
Hierarchical time series

\[ y_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix} \]

\[ y_t = S b_t \]

- \( y_t \): 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

\[ \mathbf{y}_t = \begin{pmatrix} \mathbf{y}_t \\ \mathbf{y}_{A,t} \\ \mathbf{y}_{B,t} \\ \mathbf{y}_{C,t} \\ \mathbf{y}_{AX,t} \\ \mathbf{y}_{AY,t} \\ \mathbf{y}_{AZ,t} \\ \mathbf{y}_{BX,t} \\ \mathbf{y}_{BY,t} \\ \mathbf{y}_{BZ,t} \\ \mathbf{y}_{CX,t} \\ \mathbf{y}_{CY,t} \\ \mathbf{y}_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{b}_t \]
Hierarchical time series

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

\[ S \]
Hierarchical time series

\[ y_t = S b_t \]
Grouped data

\[
y_t = \begin{pmatrix}
    y_{AX,t} \\
    y_{AY,t} \\
    y_{BX,t} \\
    y_{BY,t} \\
    y_A,t \\
    y_B,t \\
    y_X,t \\
    y_Y,t
\end{pmatrix} = \begin{pmatrix}
    1 & 1 & 1 & 1 \\
    1 & 1 & 0 & 0 \\
    0 & 0 & 1 & 1 \\
    1 & 0 & 1 & 0 \\
    0 & 1 & 0 & 1 \\
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0
\end{pmatrix} \begin{pmatrix}
    y_A,t \\
    y_B,t \\
    y_X,t \\
    y_Y,t
\end{pmatrix}
\]

\[
S = \begin{pmatrix}
    1 & 1 & 1 & 1 \\
    1 & 1 & 0 & 0 \\
    0 & 0 & 1 & 1 \\
    1 & 0 & 1 & 0 \\
    0 & 1 & 0 & 1 \\
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0
\end{pmatrix}
\]

Forecasting using R

Hierarchical and grouped time series

21
Grouped data

\[
\begin{pmatrix}
Y_t \\
Y_{A,t} \\
Y_{B,t} \\
Y_{X,t} \\
Y_{Y,t} \\
Y_{AX,t} \\
Y_{AY,t} \\
Y_{BX,t} \\
Y_{BY,t}
\end{pmatrix}
= \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
Y_{AX,t} \\
Y_{AY,t} \\
Y_{BX,t} \\
Y_{BY,t}
\end{pmatrix}
\]

\[
S = \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
forecasting using R

Hierarchical and grouped time series

\[ y_t = S b_t \]
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.
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.
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.
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.
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.
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
Optimal combination forecasts

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

\[ P = (S' \Sigma_h^{-1} S)^{-1} S' \Sigma_h^{-1}. \]

\[ \hat{y}_n(h) = S (S' \Sigma_h^{-1} S)^{-1} S' \Sigma_h^{-1} \hat{y}_n(h) \]

Reconciled forecasts

Base forecasts

**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)
Main result

The best (minimum sum of variances) unbiased forecasts are obtained when \( P = (S'\Sigma_h^{-1}s)^{-1}s'\Sigma_h^{-1} \).

\[ \tilde{y}_n(h) = S(S'\Sigma_h^{-1}s)^{-1}s'\Sigma_h^{-1}\hat{y}_n(h) \]

Reconciled forecasts

Base forecasts

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)
Main result

The best (minimum sum of variances) unbiased forecasts are obtained when
\[ P = (S'\Sigma_h^{-1}S)^{-1}S'\Sigma_h^{-1}. \]

Reconciled forecasts

\[ \tilde{y}_n(h) = S(S'\Sigma_h^{-1}S)^{-1}S'\Sigma_h^{-1}\hat{y}_n(h) \]

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)
1 Hierarchical and grouped time series

2 Lab session 15

3 Temporal hierarchies

4 Lab session 16
Lab Session 15
1. Hierarchical and grouped time series
2. Lab session 15
3. Temporal hierarchies
4. Lab session 16
Temporal hierarchies

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

Forecasting using R
Temporal hierarchies
Temporal hierarchies

Basic idea:

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

- **Annual**
  - **Semi-Annual\(_1\)**
    - **Q\(_1\)**
      - **M\(_1\)**
      - **M\(_2\)**
      - **M\(_3\)**
    - **Q\(_2\)**
      - **M\(_4\)**
      - **M\(_5\)**
      - **M\(_6\)**
  - **Semi-Annual\(_2\)**
    - **Q\(_3\)**
      - **M\(_7\)**
      - **M\(_8\)**
      - **M\(_9\)**
    - **Q\(_4\)**
      - **M\(_10\)**
      - **M\(_11\)**
      - **M\(_12\)**

- \( k = 2, 4, 12 \) nodes
- \( k = 3, 6, 12 \) nodes
- Why not \( k = 2, 3, 4, 6, 12 \) nodes?
Monthly series

- Annual
  - FourM₁
    - BiM₁
      - M₁
    - BiM₂
      - M₂
  - FourM₂
    - BiM₃
      - M₃
    - BiM₄
      - M₄
  - FourM₃
    - BiM₅
      - M₅
    - BiM₆
      - M₆

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

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

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

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

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

- $k \in F(m) = \{\text{factors of } m\}$.
- 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.
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

Forecasting using R

Temporal hierarchies

- Annual (k=52)
- Semi-annual (k=26)
- Quarterly (k=13)
- Monthly (k=4)
- Bi-weekly (k=2)
- Weekly (k=1)

Forecasting methods: base, reconciled
UK Accidents and Emergency Demand

1. Type 1 Departments — Major A&E
2. Type 2 Departments — Single Specialty
3. Type 3 Departments — Other A&E/Minor Injury
4. Total Attendances
5. Type 1 Departments — Major A&E > 4 hrs
6. Type 2 Departments — Single Specialty > 4 hrs
7. Type 3 Departments — Other A&E/Minor Injury > 4 hrs
8. Total Attendances > 4 hrs
9. Emergency Admissions via Type 1 A&E
10. Total Emergency Admissions via A&E
11. Other Emergency Admissions (i.e., not via A&E)
12. Total Emergency Admissions
13. Number of patients spending > 4 hrs from decision to admission

Forecasting using R
Temporal hierarchies
**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>

Forecasting using R
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>

Forecasting using R

Temporal hierarchies
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>
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>
**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>

Forecasting using R Temporal hierarchies
thief package for R

**thief: Temporal HIErarchical Forecasting**

**Install from CRAN**

`install.packages("thief")`

**Install from github**

`library(devtools)`

`install_github("/robjhyndman/thief")`

**Usage**

`thief(y)`
thief package for R

**thief**: Temporal HIErarchical Forecasting

**Install from CRAN**

install.packages("thief")

**Install from github**

library(devtools)
install_github("/robjhyndman/thief")

**Usage**

thief(y)
thief package for R

**thief: Temporal HIErrarchical Forecasting**

Install from CRAN
install.packages("thief")

Install from github
library(devtools)
install_github("/robjhyndman/thief")

Usage
thief(y)
1. Hierarchical and grouped time series
2. Lab session 15
3. Temporal hierarchies
4. Lab session 16
Lab Session 16