# Reconciling forecasts: the hts and thief packages

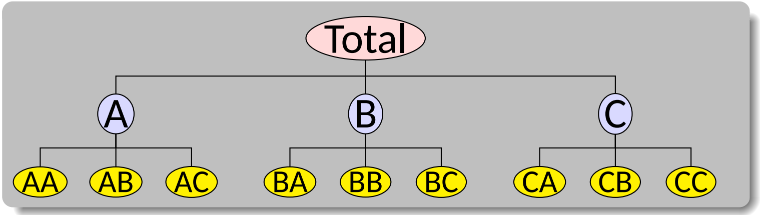
Rob J Hyndman

# 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).
- Split by region, city and stores (600)
- About 1 million bottom-level series

# Hierarchical time series

A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

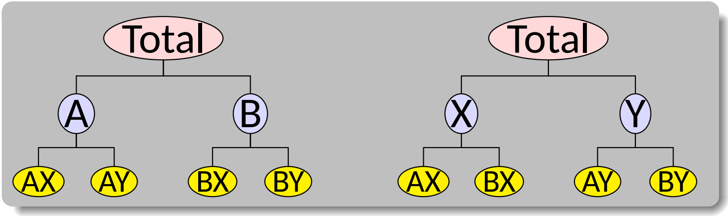



## Examples

- Sales by region, city, store

# Grouped time series

A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.




## Examples

- Sales by brand, gender, material, 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?

# 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.

# 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.

# 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 package for R

**hts: Hierarchical and Grouped Time Series**

Methods for analysing and forecasting hierarchical and grouped time series

| | |
|---|---|
| Version: | 5.0 |
| Depends: | R ($\geqslant$ 3.0.2), forecast ($\geqslant$ 5.0), SparseM, Matrix, matrixcalc |
| Imports: | parallel, utils, methods, graphics, grDevices, stats |
| LinkingTo: | Rcpp ($\geqslant$ 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 ($\geqslant$ 2) |

# Example using R

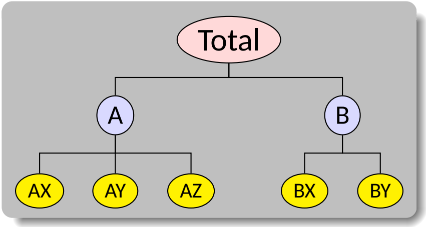
```
library(hts)

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

# 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)))
```

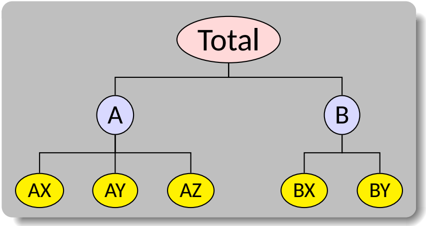


# Example using R

```
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)))

# 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)
```

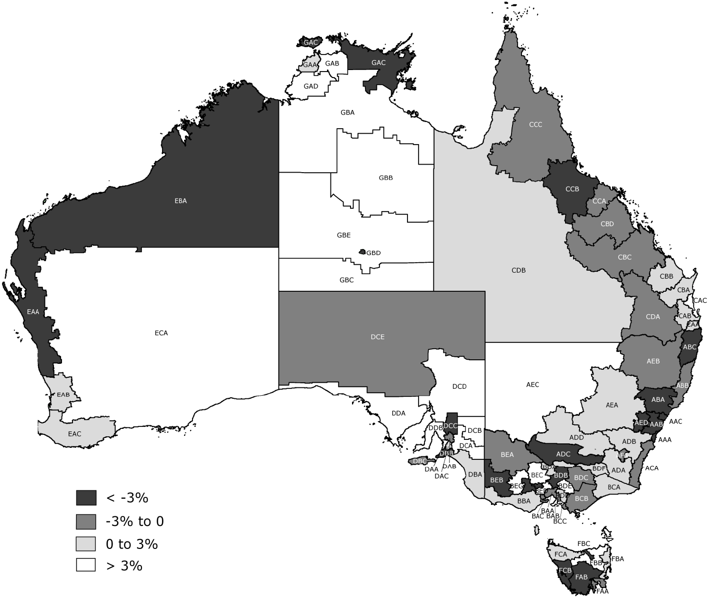
### Arguments

| | |
|---|---|
| `y` | Multivariate time series containing the bottom level series |
| `characters` | Vector of integers, or list of vectors, showing how column names indicate group structure. |

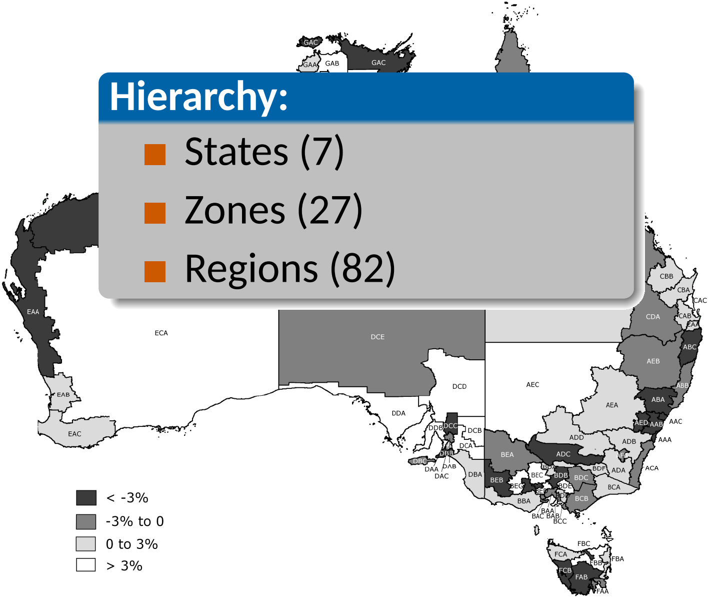
### Example

```
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))
```

# Australian tourism



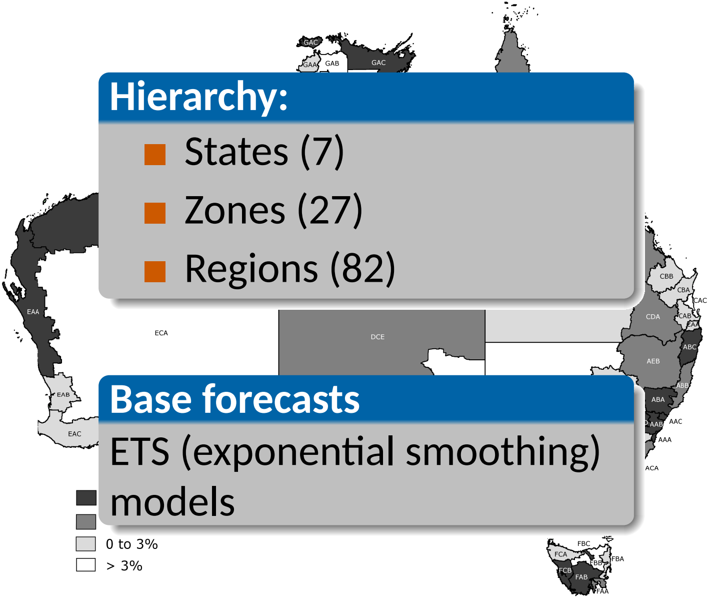
# Australian tourism




## Hierarchy:

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

# Australian tourism



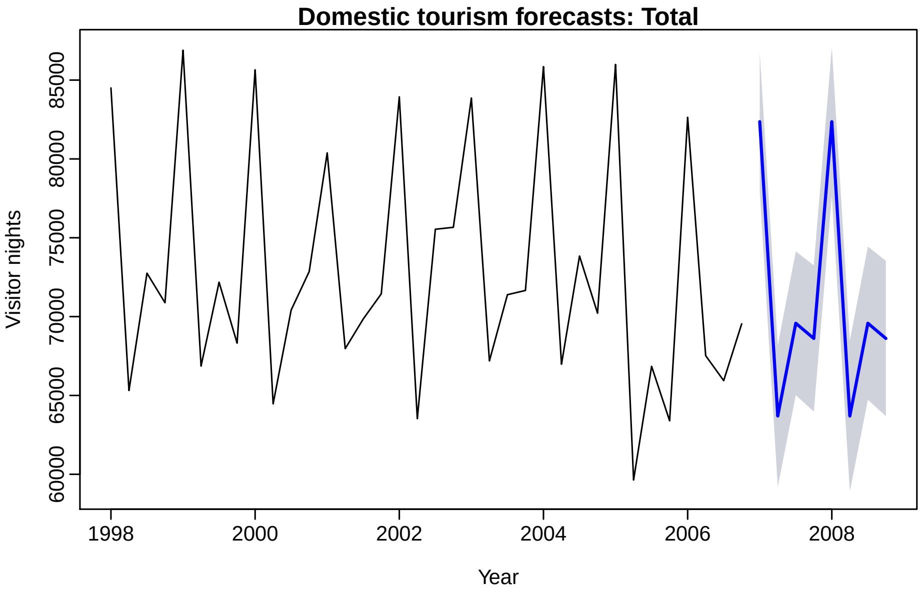

## Hierarchy:

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

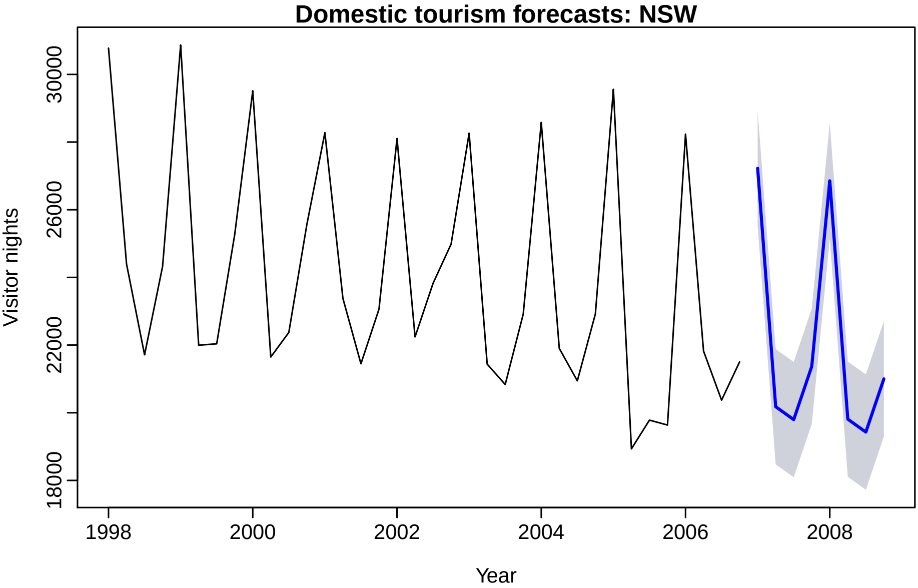
## Base forecasts

ETS (exponential smoothing) models

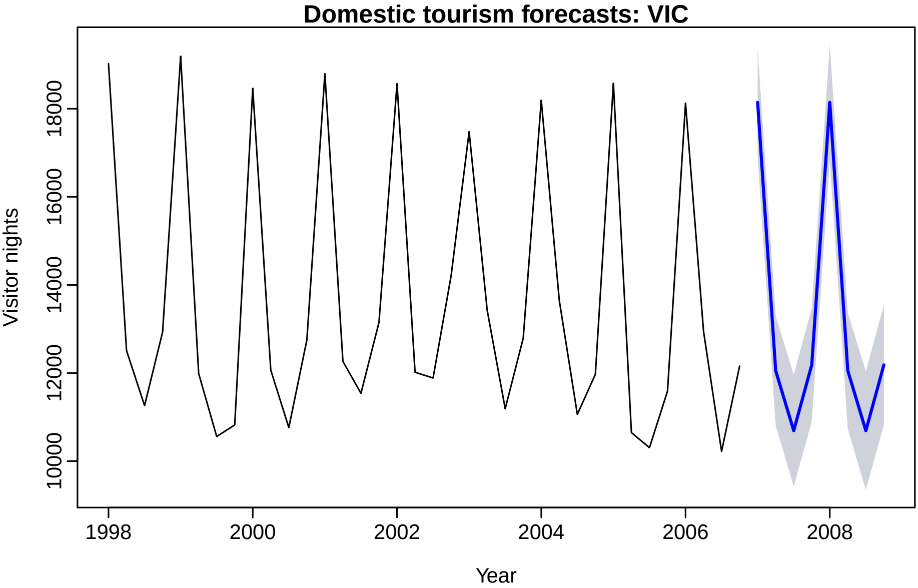
# Base forecasts



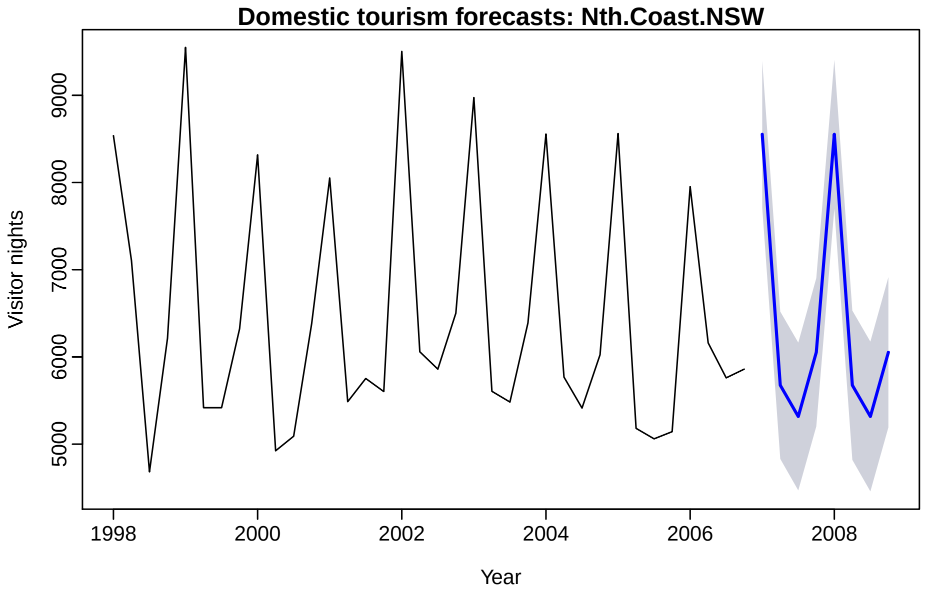
# Base forecasts



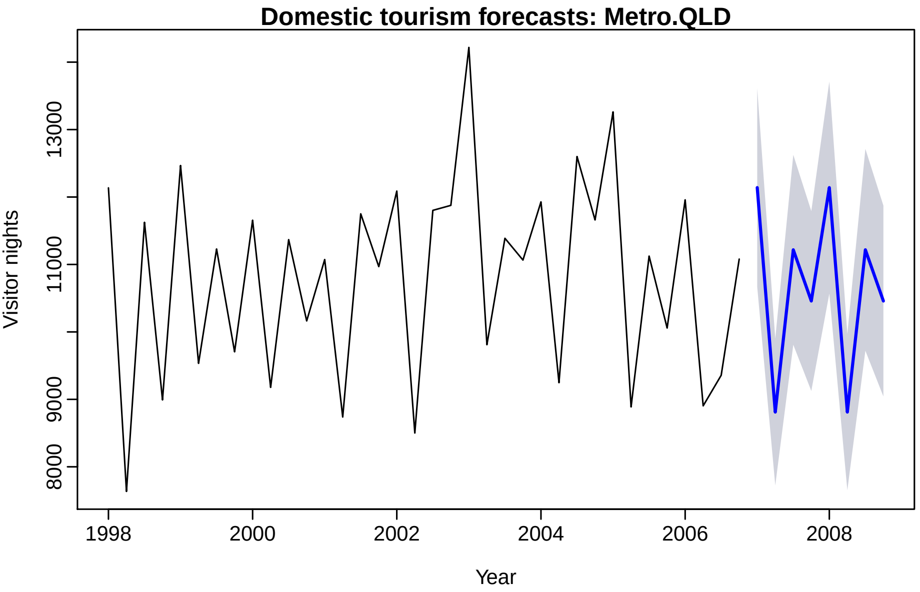
# Base forecasts



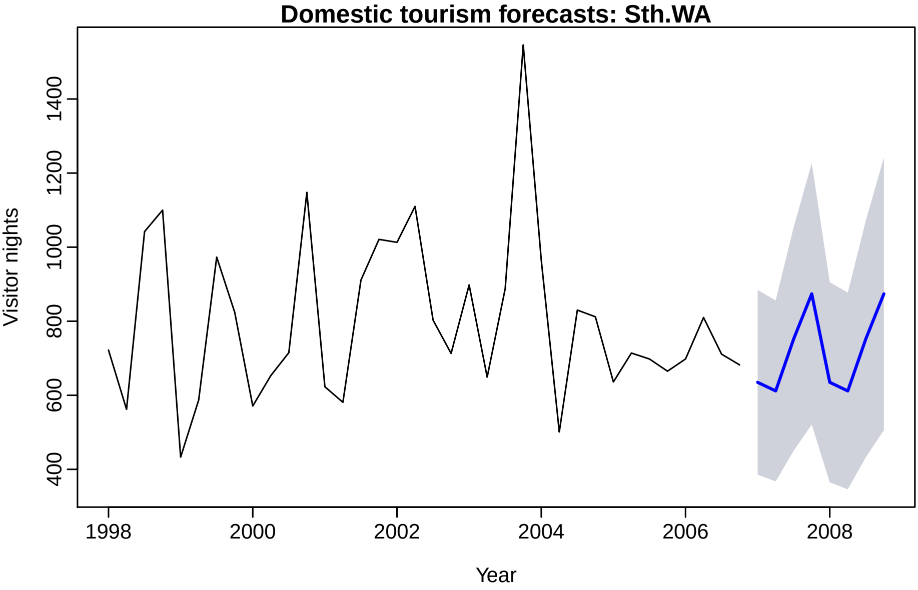
# Base forecasts



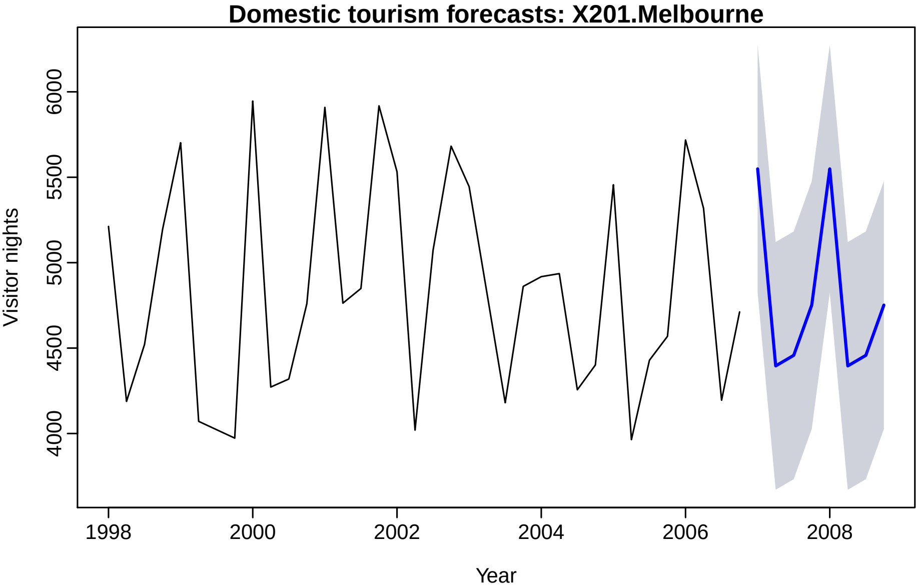
# Base forecasts



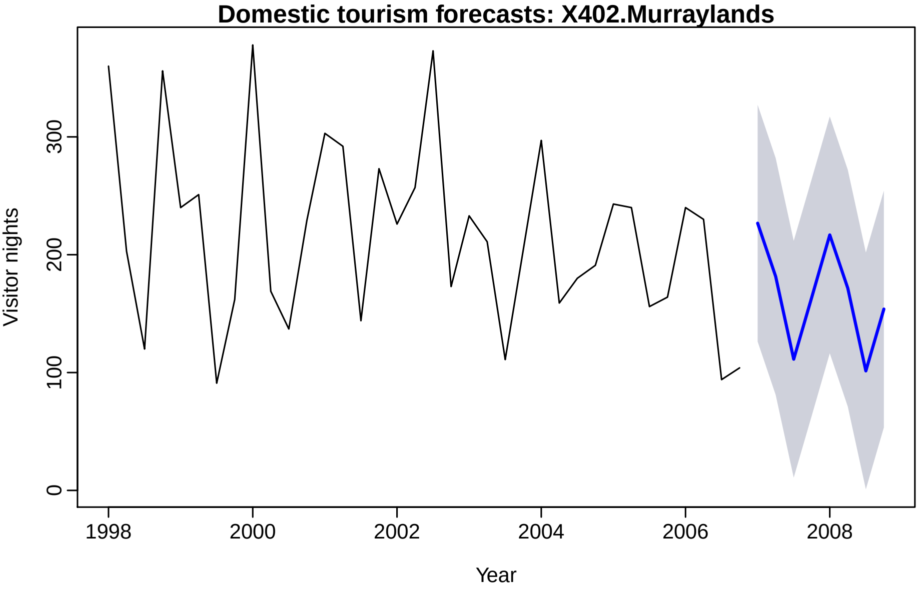
# Base forecasts



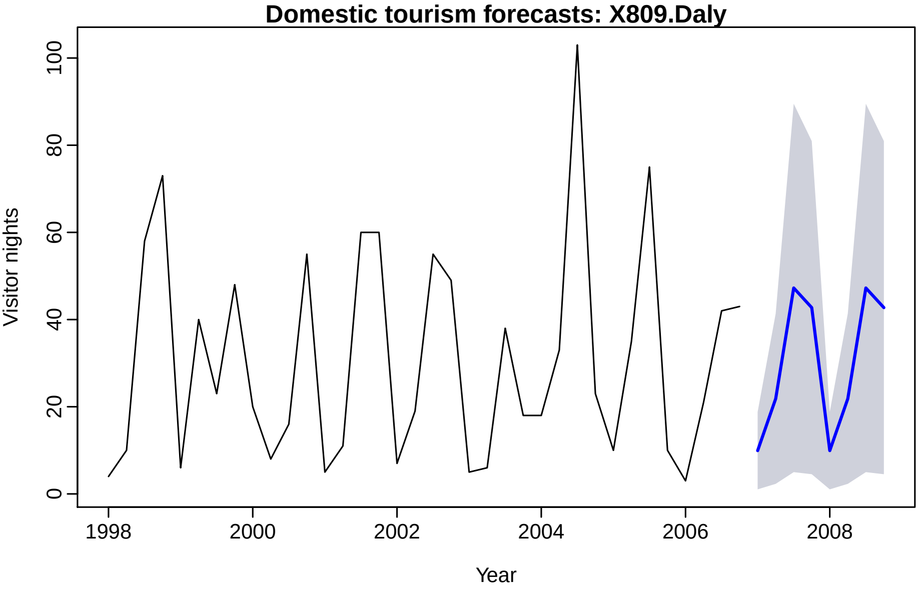
# Base forecasts



# Base forecasts



# Base forecasts

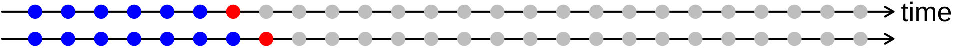


# Forecast evaluation

**Training sets** **Test sets $h = 1$**

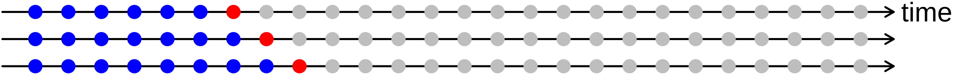
# Forecast evaluation

**Training sets** **Test sets $h = 1$**



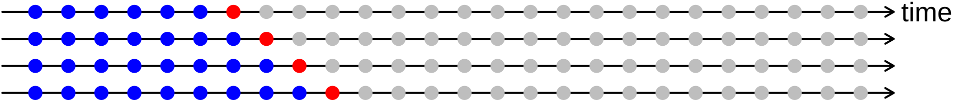
# Forecast evaluation

**Training sets** **Test sets $h = 1$**

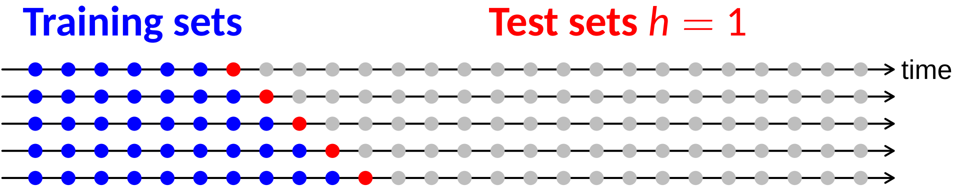


# Forecast evaluation

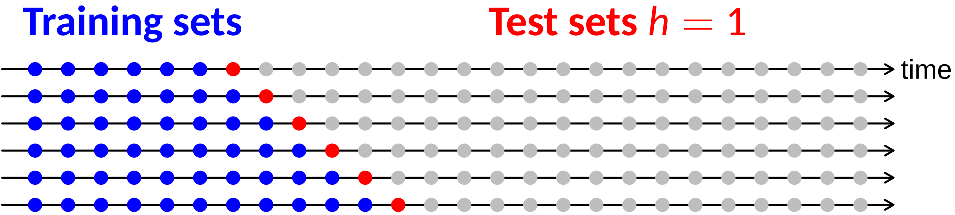
**Training sets** **Test sets $h = 1$**



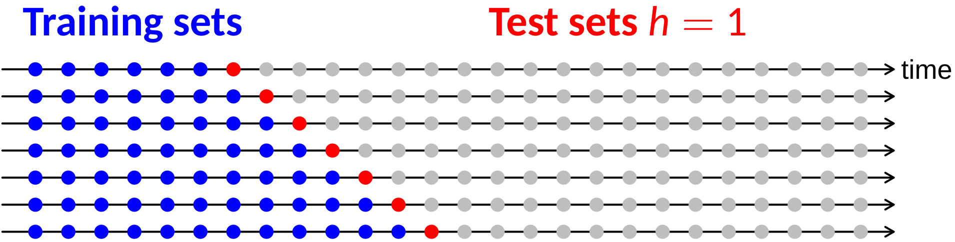
# Forecast evaluation



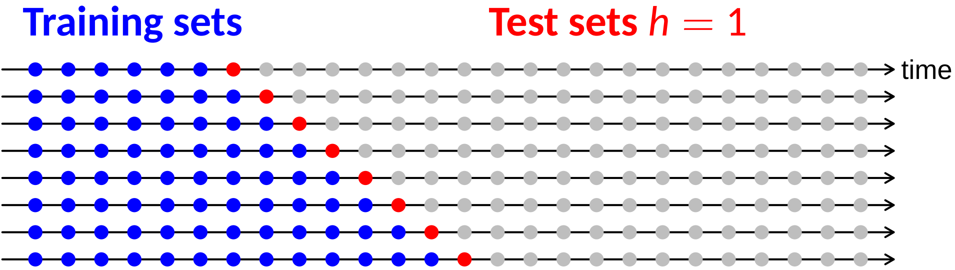
# Forecast evaluation



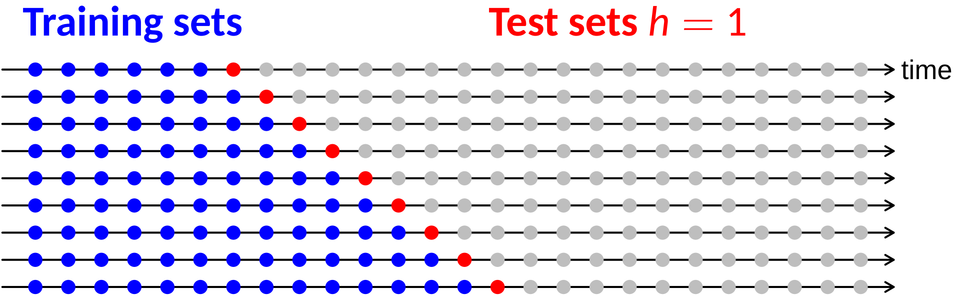
# Forecast evaluation



# Forecast evaluation

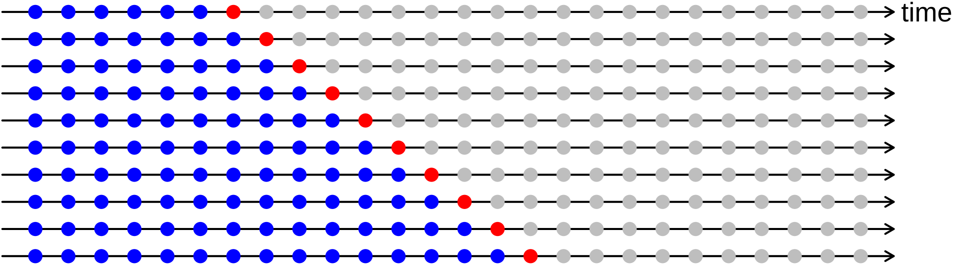


# Forecast evaluation

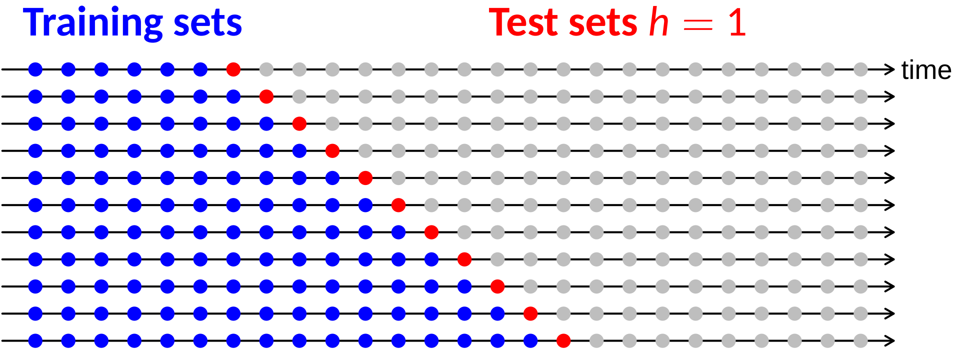


# Forecast evaluation

**Training sets** **Test sets $h = 1$**

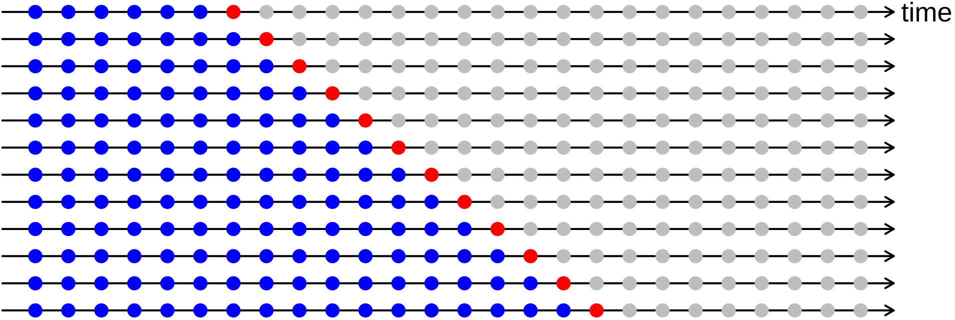


# Forecast evaluation

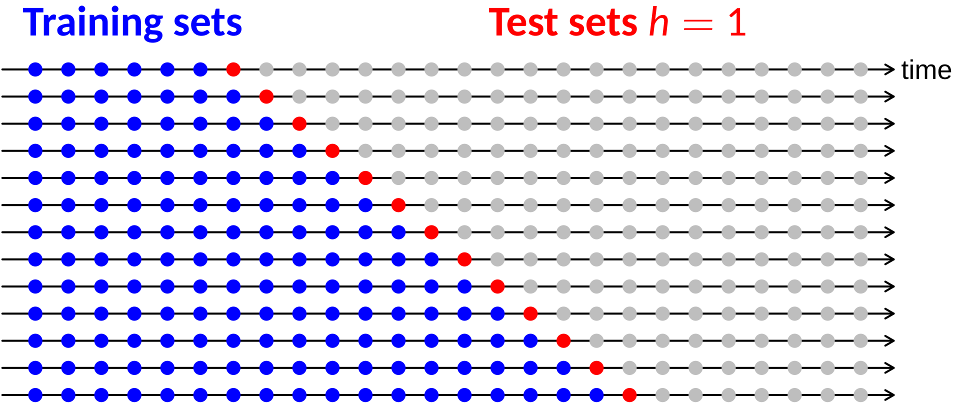


# Forecast evaluation

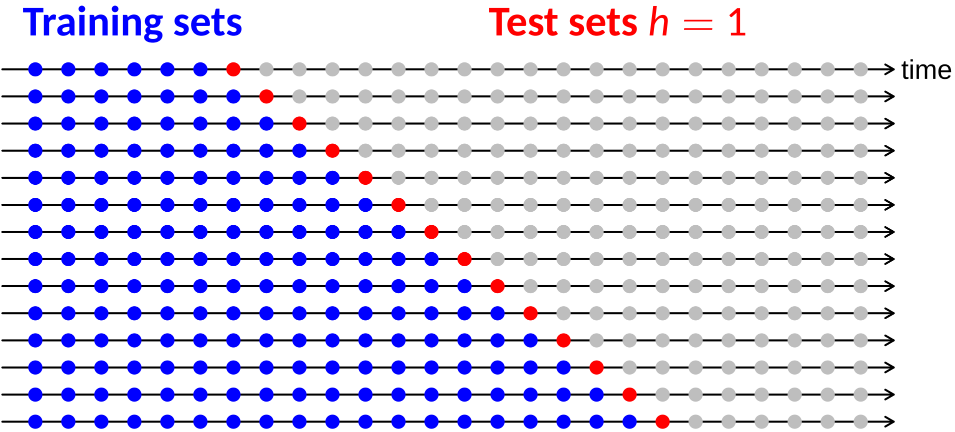
**Training sets** **Test sets $h = 1$**



# Forecast evaluation

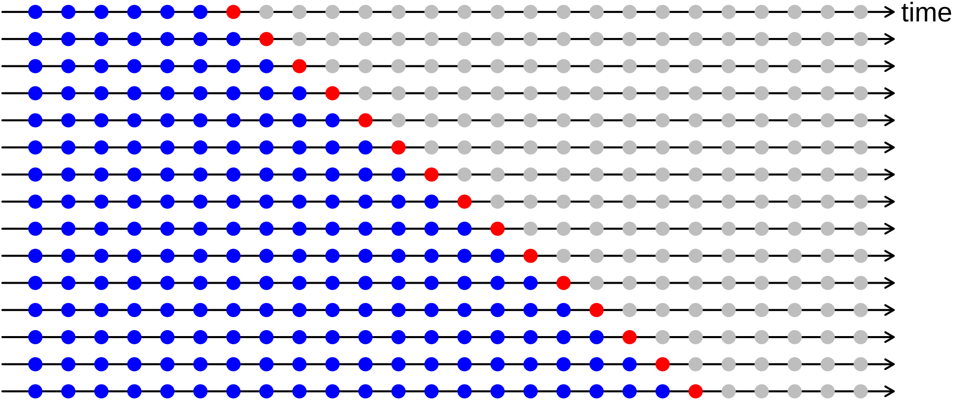


# Forecast evaluation



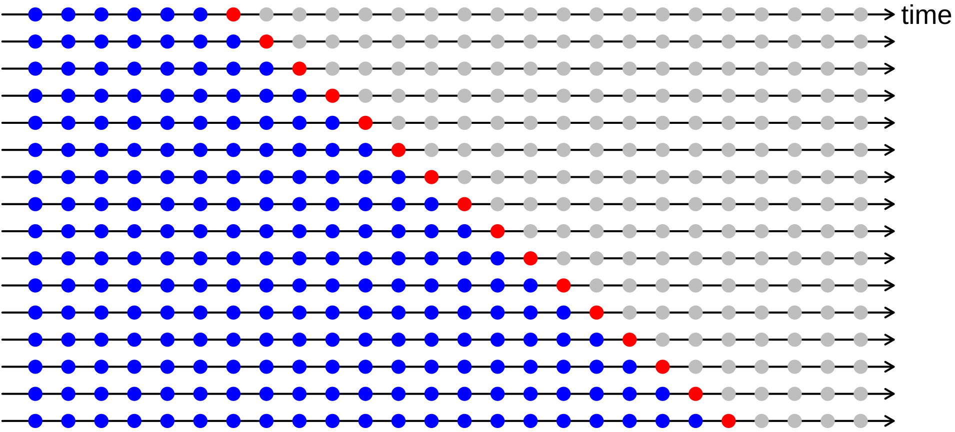
# Forecast evaluation

**Training sets** **Test sets $h = 1$**

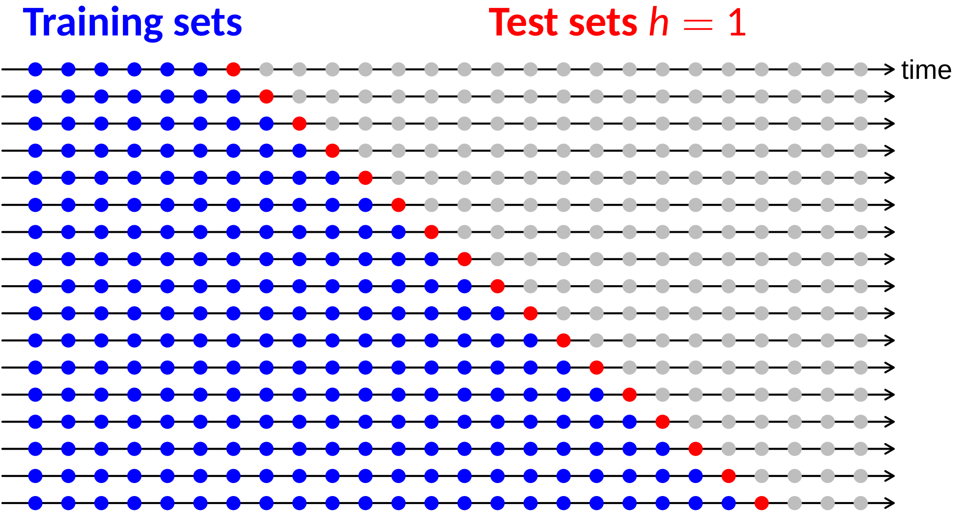


# Forecast evaluation

**Training sets** **Test sets $h = 1$**

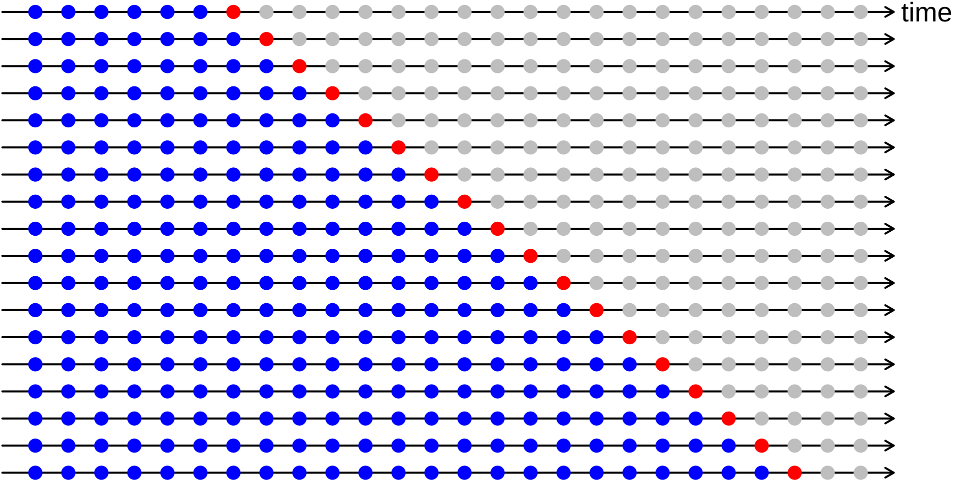


# Forecast evaluation

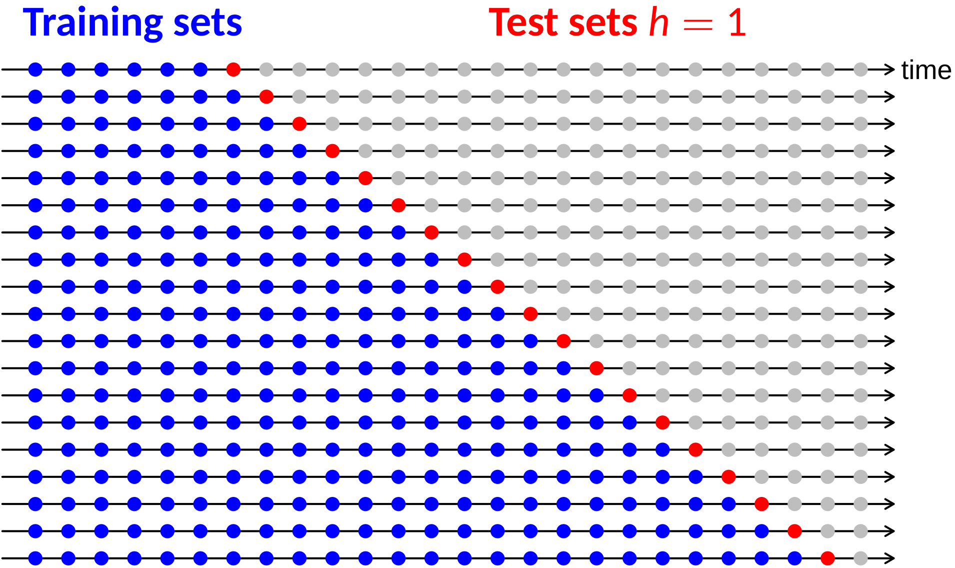


# Forecast evaluation

**Training sets** **Test sets $h = 1$**

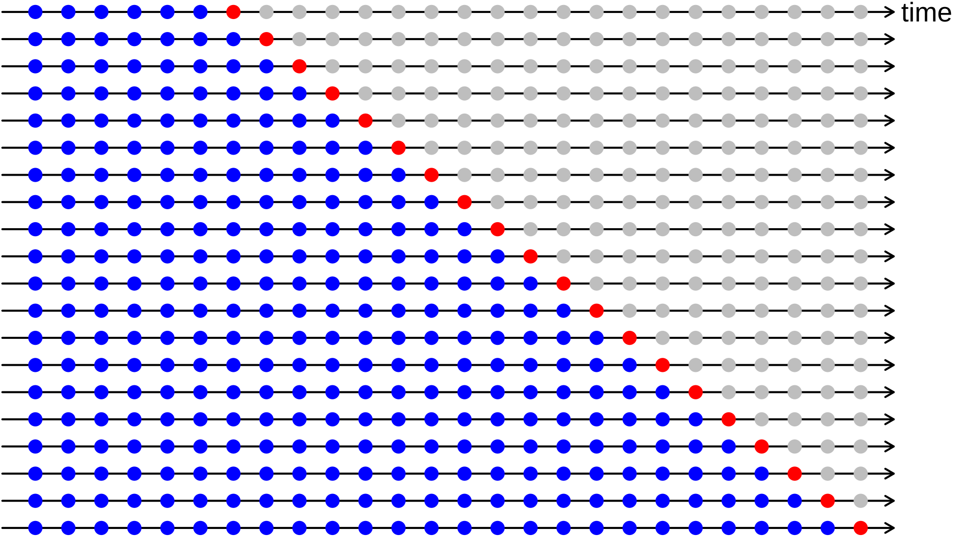


# Forecast evaluation

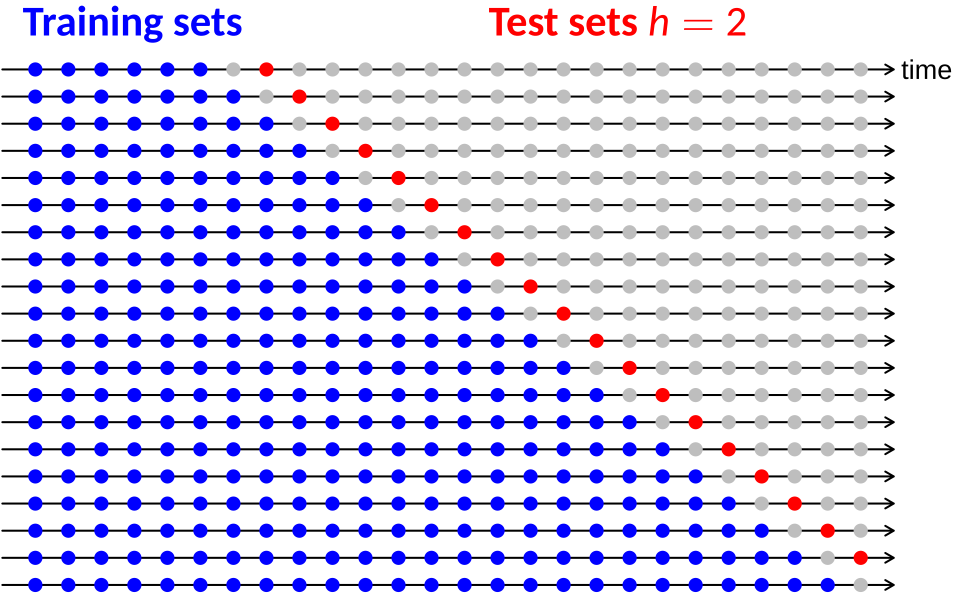


# Forecast evaluation

**Training sets** **Test sets $h = 1$**

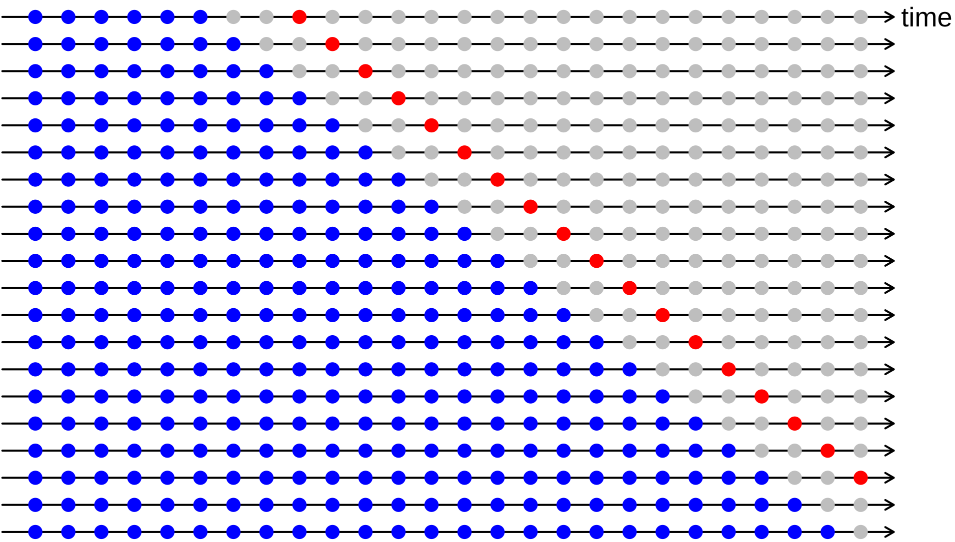


# Forecast evaluation

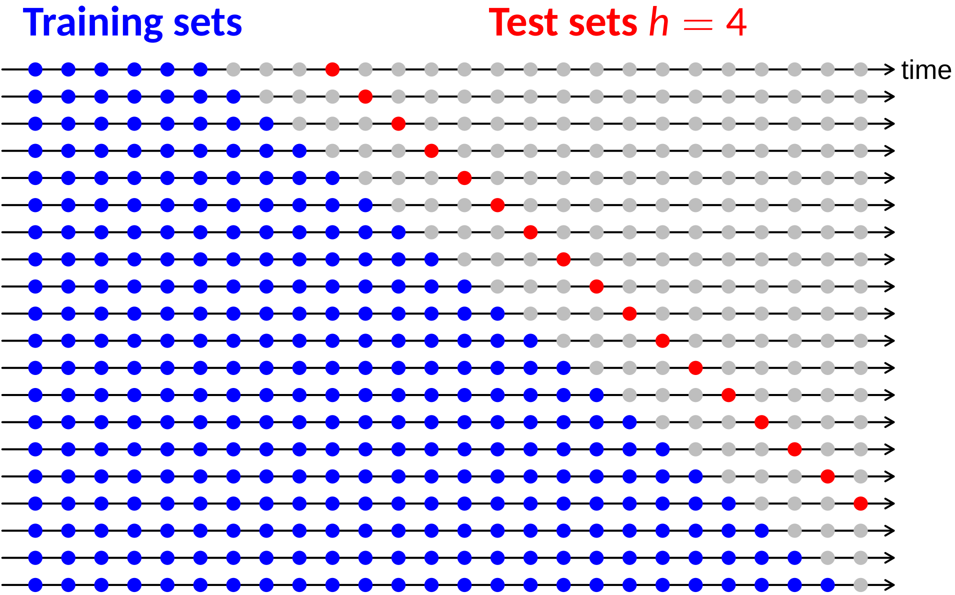


# Forecast evaluation

**Training sets** **Test sets $h = 3$**

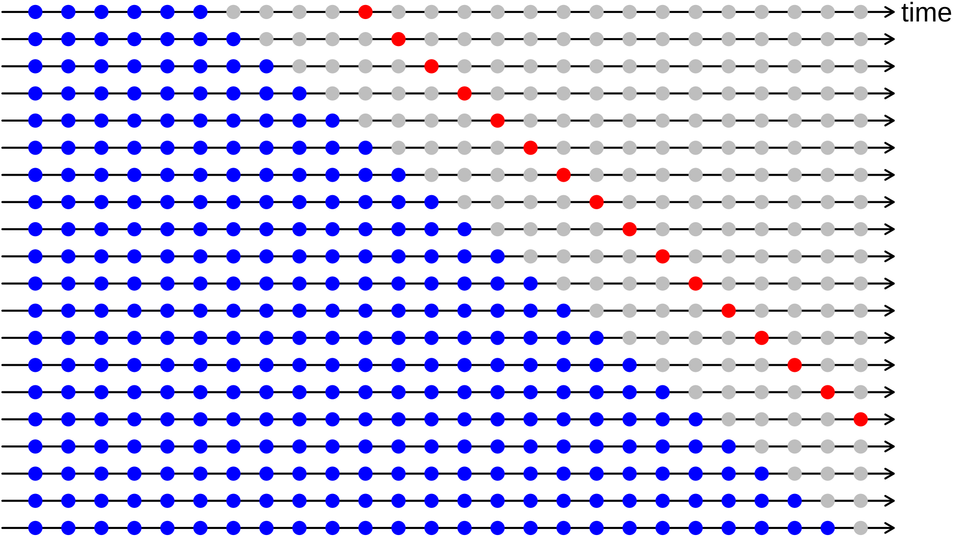


# Forecast evaluation



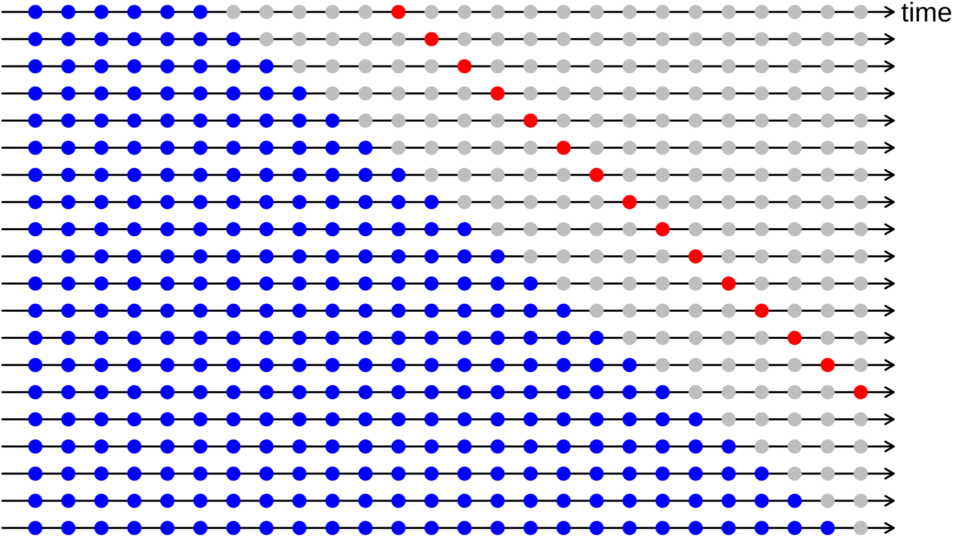
# Forecast evaluation

**Training sets** **Test sets $h = 5$**



# Forecast evaluation

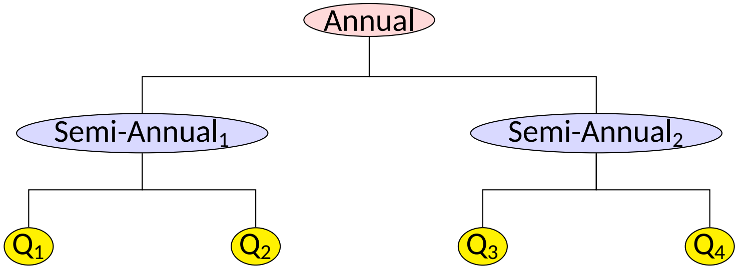
**Training sets** **Test sets $h = 6$**



# Hierarchy: states, zones, regions

| | Forecast horizon | | | | | | |
|---|---|---|---|---|---|---|---|
| **RMSE** | $h = 1$ | $h = 2$ | $h = 3$ | $h = 4$ | $h = 5$ | $h = 6$ | **Ave** |
| | | | **Australia** | | | | |
| Base | 1762.04 | 1770.29 | 1766.02 | 1818.82 | 1705.35 | 1721.17 | **1757.28** |
| Bottom | 1736.92 | 1742.69 | 1722.79 | 1752.74 | 1666.73 | 1687.43 | **1718.22** |
| OLS | 1747.60 | 1757.68 | 1751.77 | 1800.67 | 1686.00 | 1706.45 | **1741.69** |
| WLS | 1705.21 | 1715.87 | 1703.75 | 1729.56 | 1627.79 | 1661.24 | **1690.57** |
| GLS | 1704.64 | 1715.60 | 1705.31 | 1729.04 | 1626.36 | 1661.64 | **1690.43** |
| | | | **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** |

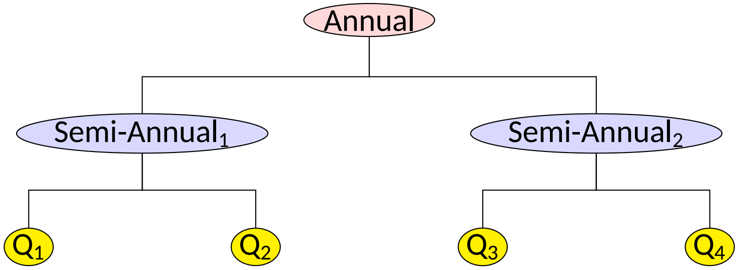
# Temporal hierarchies




**Basic idea:**

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

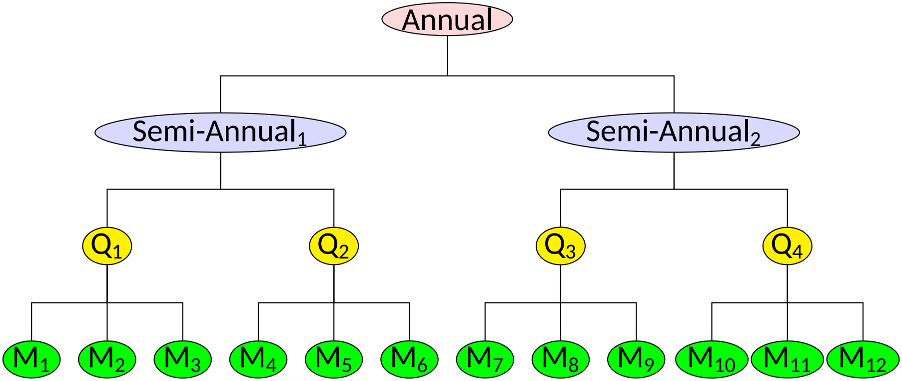
# Temporal hierarchies




**Basic idea:**

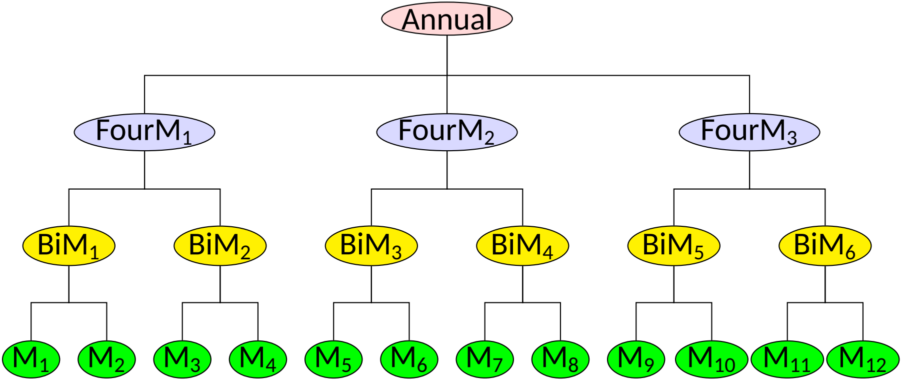
- Forecast series at each available frequency.
- Optimally reconcile forecasts within the same year.

# Monthly series



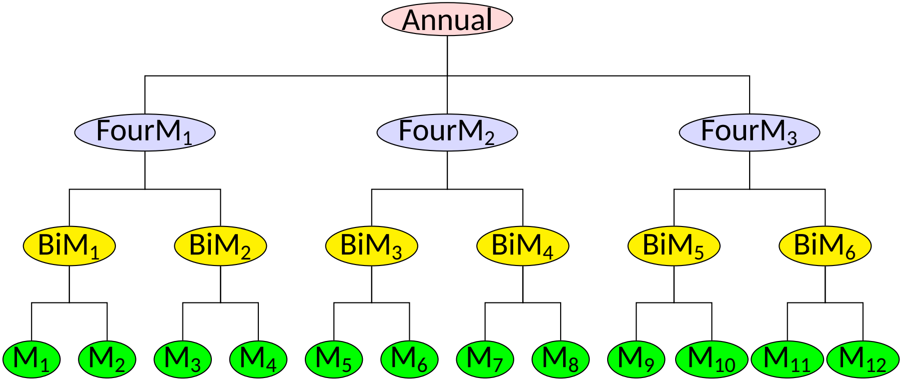

- $k = 2, 4, 12$ nodes
- $k = 3, 6, 12$ nodes
- Why not $k = 2, 3, 4, 6, 12$ nodes?

# Monthly series




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

# Monthly series




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

# thief package for R

## thief: Temporal HIErarchical Forecasting

**Usage**

```
thief(y)
```

```
thief(y, m = frequency(y), h = m*2,
  comb = c("struc","mse","ols","bu","shr","sam"),
  usemodel = c("ets","arima","theta","naive","snaive"),
  forecastfunction = NULL, ...)
```

# More information



Rob J Hyndman, Roman A Ahmed, George Athanasopoulos, and Han Lin Shang (2011). "Optimal combination forecasts for hierarchical time series". *Computational Statistics & Data Analysis* **55**(9), 2579–2589.


Rob J Hyndman, Alan J Lee, and Earo Wang (2016). "Fast computation of reconciled forecasts for hierarchical and grouped time series". *Computational Statistics & Data Analysis* **97**, 16–32.

Shanika L Wickramasuriya, George Athanasopoulos, and Rob J Hyndman (2015). *Forecasting hierarchical and grouped time series through trace minimization*. Working paper 15/15. Monash University

George Athanasopoulos, Rob J Hyndman, Nikolaos Kourentzes, and Fotios Petropoulos (2015). *Forecasting with temporal hierarchies*. Working paper. Monash University

Rob J Hyndman, Alan J Lee, Earo Wang, and Shanika Wickramasuriya (2016). *hts: Hierarchical and Grouped Time Series*. R package v5.0 on CRAN.

Rob J Hyndman and Nikolaos Kourentzes (2016). *thief: Temporal Hierarchical Forecasting*. R package v0.2 on CRAN.

# More information



Rob J Hyndman, Roman A Ahmed, George Athanasopoulos, and Han Lin Shang (2011). "Optimal combination forecasts for hierarchical time series". *Computational Statistics & Data Analysis* **55**(9), 2579–2589.


Rob J Hyndman, Alan J Lee, and Earo Wang (2016). "Fast computation of reconciled forecasts for hierarchical and grouped time series". *Computational Statistics & Data Analysis* **97**, 16–32.


Shanika L Wickramasuriya, George Athanasopoulos, and Rob J Hyndman (2015). *Forecasting hierarchical and grouped time series through trace minimization*. Working paper 15/15. Monash University


George Athanasopoulos, Rob J Hyndman, Nikolaos Kourentzes, and Fotios Petropoulos (2015). *Forecasting with temporal hierarchies*. Working paper. Monash University

Rob J Hyndman, Alan J Lee, Earo Wang, and Shanika Wickramasuriya (2016). *hts: Hierarchical and Grouped Time Series*. R package v5.0 on CRAN.

Rob J Hyndman and Nikolaos Kourentzes (2016). *thief: Temporal Hierarchical Forecasting*. R package v0.2 on CRAN.


➡ More information: **robjhyndman.com**