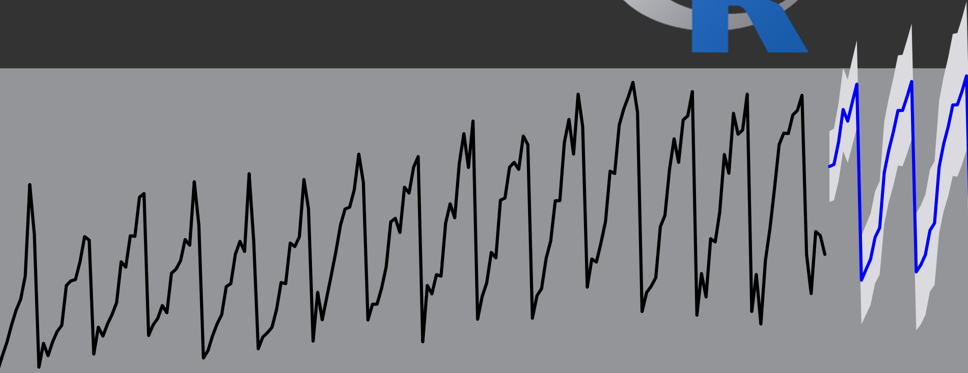
MONASH BUSINESS SCHOOL

Rob J Hyndman

Forecasting big time series data using



Outline

Motivation

Forecasting Google search traffic for the top few thousand searches by region.

Motivation

Forecasting the government PBS expenditure for the next five years.

Motivation

1. Common in business to have millions of time series that need forecasting every day.
2. Forecasts are often required by people who are untrained in time series analysis.

Specifications

Automatic forecasting algorithms must:

- ➥ determine an appropriate time series model;
- ➥ estimate the parameters;
- ➥ compute the forecasts with prediction intervals.

Motivation

1. Common in business to have millions of time series that need forecasting every day.
2. Forecasts are often required by people who are untrained in time series analysis.

Specifications

Automatic forecasting algorithms must:

- ➥ determine an appropriate time series model;
- ➥ estimate the parameters;
- ➥ compute the forecasts with prediction intervals.

M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

Spyros Makridakis, Michèle Hibon*

INSEAD, Boulevard de Constance, 77305 Fontainebleau, France

Abstract

This paper describes the M3-Competition, the latest of the M-Competitions. It explains the reasons for conducting the competition and summarizes its results and conclusions. In addition, the paper compares such results/conclusions with those of the previous two M-Competitions as well as with those of other major empirical studies. Finally, the implications of these results and conclusions are considered, their consequences for both the theory and practice of forecasting are explored and directions for future research are contemplated.

Keywords: Comparative methods — time series: univariate; Forecasting competitions; M-Competition; Forecasting methods, Forecasting accuracy

M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

www.elsevier.com/locate/ijforecast

: results, conclusion

Makridakis, Michèle Hib

ard de Constance, 77305 Fontainebleа

Abstract

This p... he latest of the M-Competitions. I... cting the competiti... clusions. In addition, the paper com... with those of the pre... with those of other major empirical ... s of these results and conclusions are considered, their consequences for both the theory and practice of forecasting are explored and directions for future research are contemplated.

Keywords: Comparative methods — time series: univariate; Forecasting competitions; M-Competition; Forecasting methods, Forecasting accuracy

M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- All data from business, demography, finance and economics.
- Series length between 14 and 126.
- Either non-seasonal, monthly or quarterly

M3 competition

ELSEVIER
International Journal of Forecasting 16 (2000) 451–476
international journal of forecasting
www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- All data from business, demography, finance and economics.
- Series length between 14 and 126.
- Either non-seasonal, monthly or quarterly.

M3 competition

ELSEVIER
International Journal of Forecasting 16 (2000) 451–476
international journal of forecasting
www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- All data from business, demography, finance and economics.
- Series length between 14 and 126.
- Either non-seasonal, monthly or quarterly.

M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- All data from business, demography, finance and economics.
- Series length between 14 and 126.
- Either non-seasonal, monthly or quarterly.

M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- All data from business, demography, finance and economics.
- Series length between 14 and 126.
- Either non-seasonal, monthly or quarterly.

M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

Winning commercial software

Method	MAPE	sMAPE	MASE
ForecastPro	18.00	13.06	1.47
ForecastX	17.35	13.09	1.42

Abstra

This … ng the
compet … those
of the p … these
results a … explored and
directions for future research are contemplated.

Keywords: Comparative methods — time series: univariate; Forecasting competitions; M-Competition; Forecasting methods, Forecasting accuracy

Outline

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	Seasonal Component A (Additive)	Seasonal Component M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

N,N: Simple exponential smoothing

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

A,N: Holt's linear method

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

A_d,N: Additive damped trend method

Exponential smoothing methods

Trend Component	Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N (None)	N,N	N,A	N,M
A (Additive)	A,N	A,A	A,M
A_d (Additive damped)	A_d,N	A_d,A	A_d,M

A,A: Additive Holt-Winters' method

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

A,M: Multiplicative Holt-Winters' method

Exponential smoothing methods

Trend Component	Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N (None)	N,N	N,A	N,M
A (Additive)	A,N	A,A	A,M
A_d (Additive damped)	A_d,N	A_d,A	A_d,M

A_d,M: Damped multiplicative Holt-Winters' method

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	Seasonal Component A (Additive)	Seasonal Component M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

Each method can have an additive or multiplicative error, giving 18 separate models.

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

↑

Trend

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

↑

Trend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

↗ ↑ ↖

Error **T**rend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

↗ ↑ ↖

Error **T**rend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Innovations state space models

- All ETS models can be written in innovations state space form (IJF, 2002).
- Additive and multiplicative versions give the same point forecasts but different prediction intervals.

General notation E T S : **E**xponen**T**ial **S**moothing

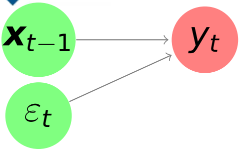
↗ ↑ ↖

Error **T**rend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

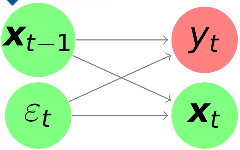
ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

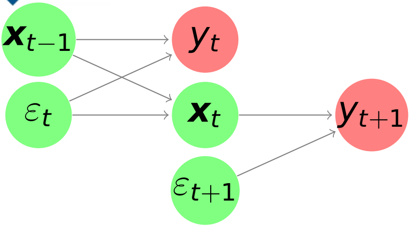
ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

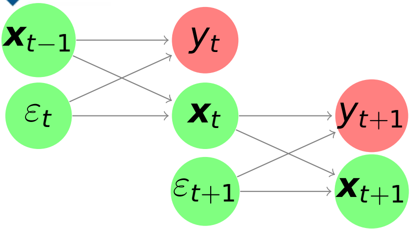
ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

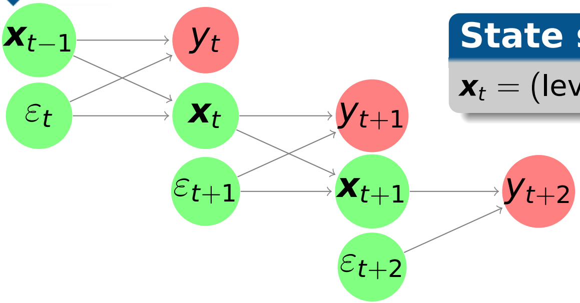
ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

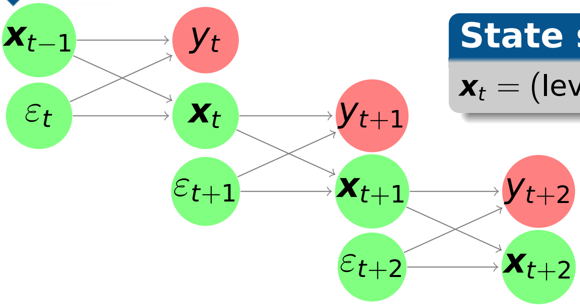
ETS state space model



State space model

$$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$$

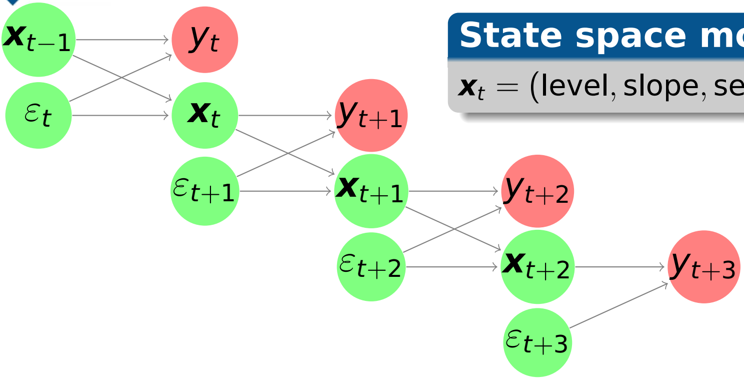
ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

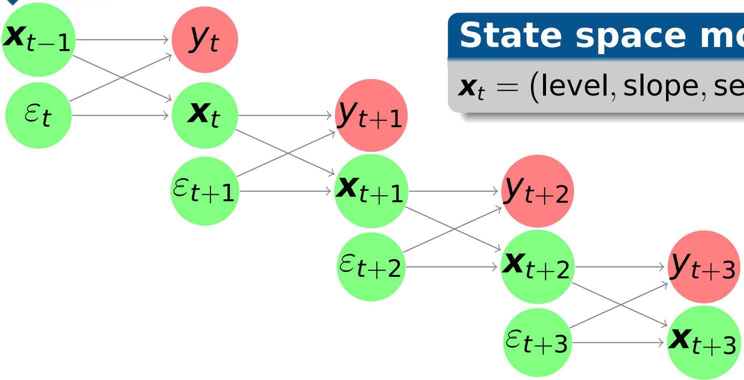
ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

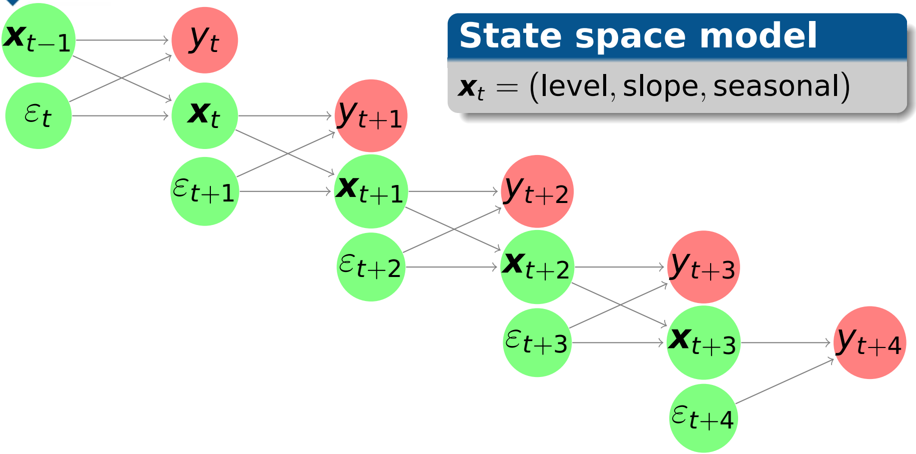
ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

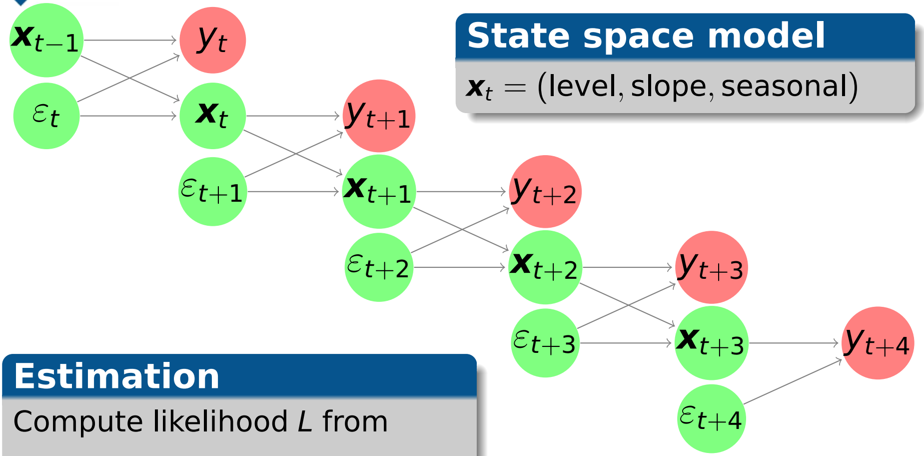
ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

ETS state space model



State space model

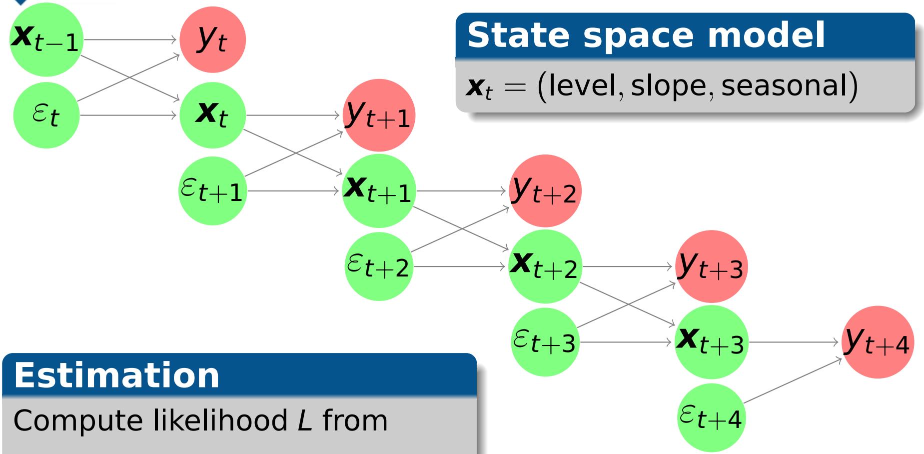
$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

Estimation

Compute likelihood L from $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$.

Optimize L wrt model parameters.

ETS state space model



State space model

$\boldsymbol{x}_t = (\text{level}, \text{slope}, \text{seasonal})$

Estimation

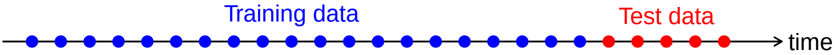
Compute likelihood L from $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T$.

Optimize L wrt model parameters.

Q: How to choose between the 18 ETS models?

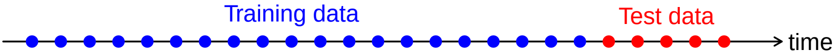
Cross-validation

Traditional evaluation

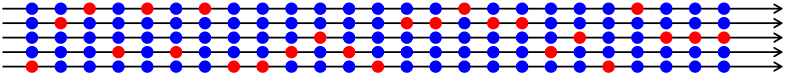


Cross-validation

Traditional evaluation

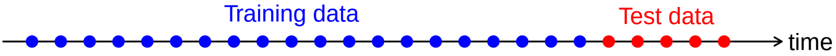


Standard cross-validation

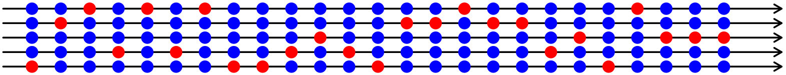


Cross-validation

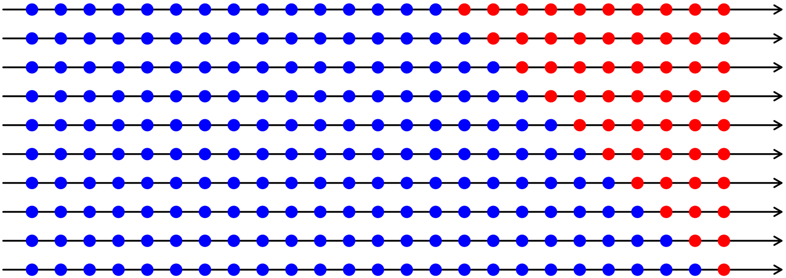
Traditional evaluation



Standard cross-validation

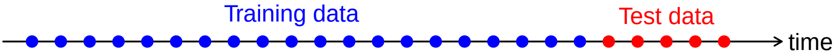


Time series cross-validation

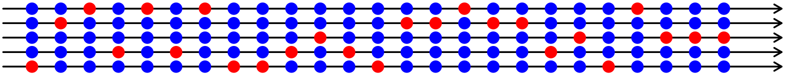


Cross-validation

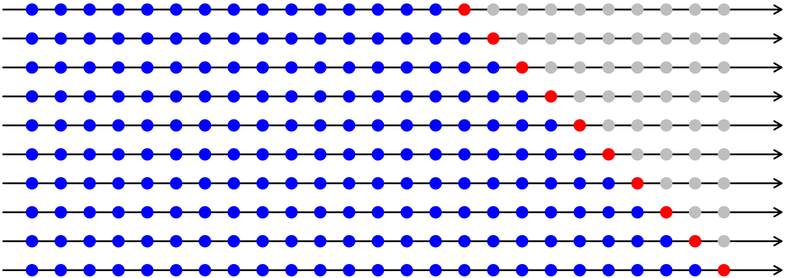
Traditional evaluation



Standard cross-validation

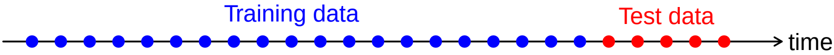


Time series cross-validation

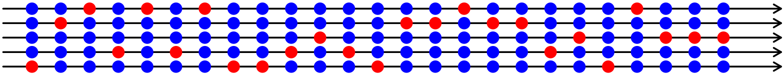


Cross-validation

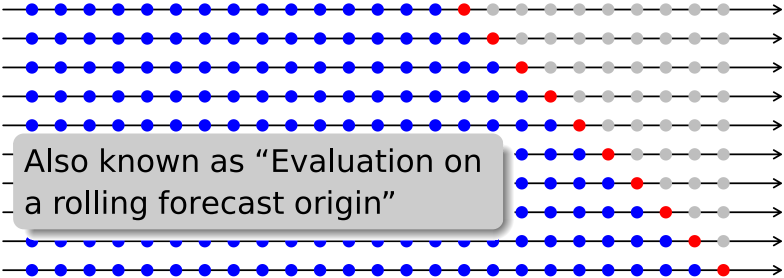
Traditional evaluation



Standard cross-validation



Time series cross-validation



Also known as "Evaluation on a rolling forecast origin"

Akaike's Information Criterion

$$\text{AIC} = -2\log(L) + 2k$$

where L = likelihood

k = number of estimated parameters in model.

- This is a *penalized likelihood* approach.

Akaike's Information Criterion

$$\text{AIC}_\text{C} = -2\log(L) + 2k + \frac{2(k+1)(k+2)}{T-k}$$

where L = likelihood

k = number of estimated parameters in model

and T = length of the series.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then

$$\text{AIC}_\text{C} \approx \text{constant} + T\log\text{MSE} + 2k$$

where MSE is on 1-step forecasts on **training set**.

Akaike's Information Criterion

$$\text{AIC}_\text{C} = -2\log(L) + 2k + \frac{2(k+1)(k+2)}{T-k}$$

where L = likelihood

k = number of estimated parameters in model

and T = length of the series.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then

$$\text{AIC}_\text{C} \approx \text{constant} + T\log\text{MSE} + 2k$$

where MSE is on 1-step forecasts on **training set**.

Akaike's Information Criterion

$$\text{AIC}_\text{C} = -2\log(L) + 2k + \frac{2(k+1)(k+2)}{T-k}$$

where $L =$ likelihood

$k =$ number of estimated parameters in model

and $T =$ length of the series.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then

$$\text{AIC}_\text{C} \approx \text{constant} + T\log\text{MSE} + 2k$$

where MSE is on 1-step forecasts on **training set**.

Akaike's Information Criterion

$$\text{AIC}_\text{C} = -2\log(L) + 2k + \frac{2(k+1)(k+2)}{T-k}$$

where L = likelihood

k = number of estimated parameters in model

and T = length of the series.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then

$$\text{AIC}_\text{C} \approx \text{constant} + T\log\text{MSE} + 2k$$

where MSE is on 1-step forecasts on **training set**.

Minimizing the Gaussian AIC_C is asymptotically equivalent (as $T \to \infty$) to minimizing MSE from 1-step forecasts via **time series cross-validation**.

ets algorithm in R

Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

- Apply each of 18 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

ets algorithm in R

Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

- Apply each of 18 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

ets algorithm in R

Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

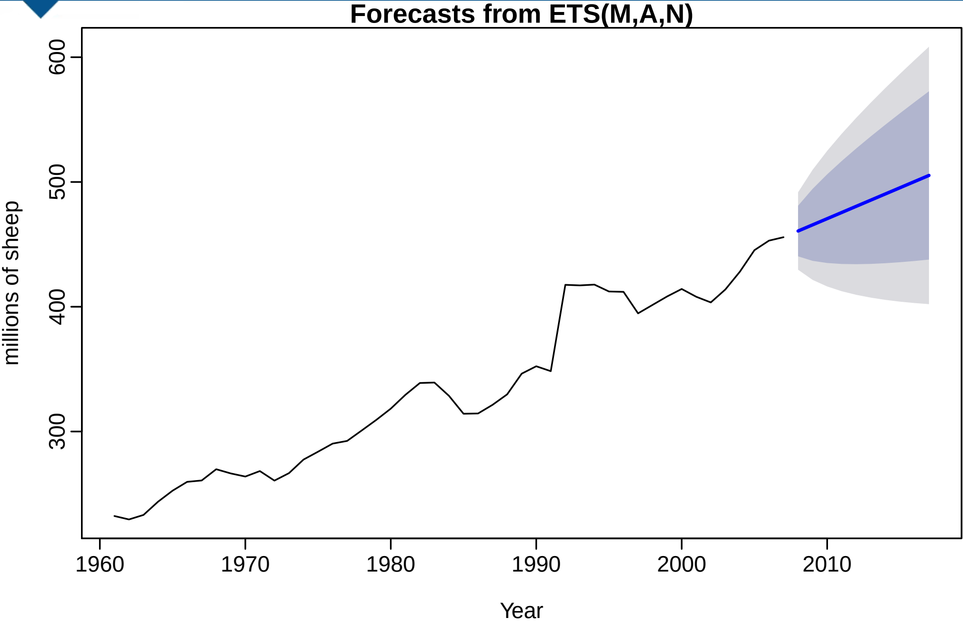
- Apply each of 18 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

ets algorithm in R

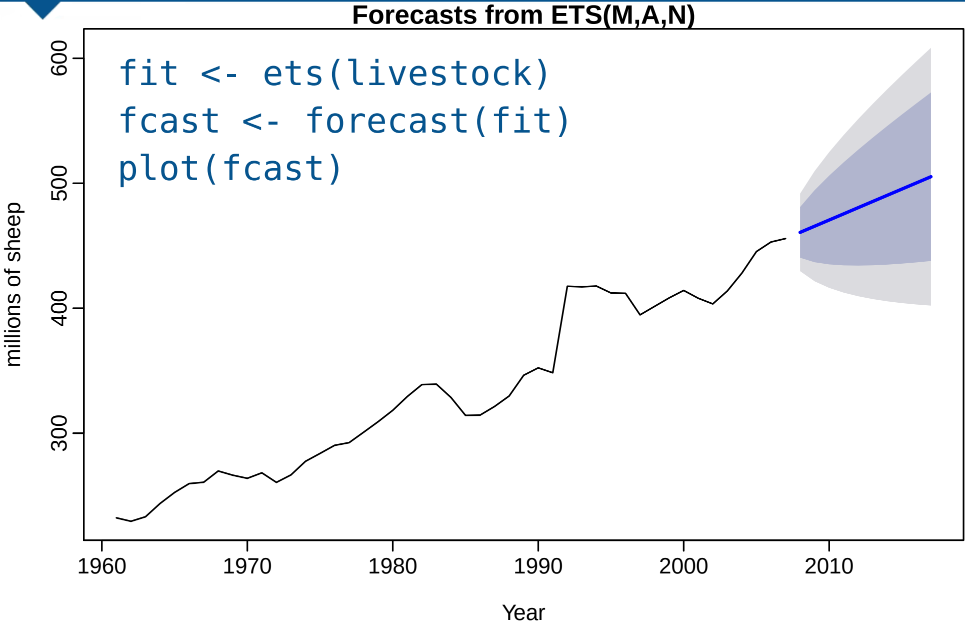
Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

- Apply each of 18 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

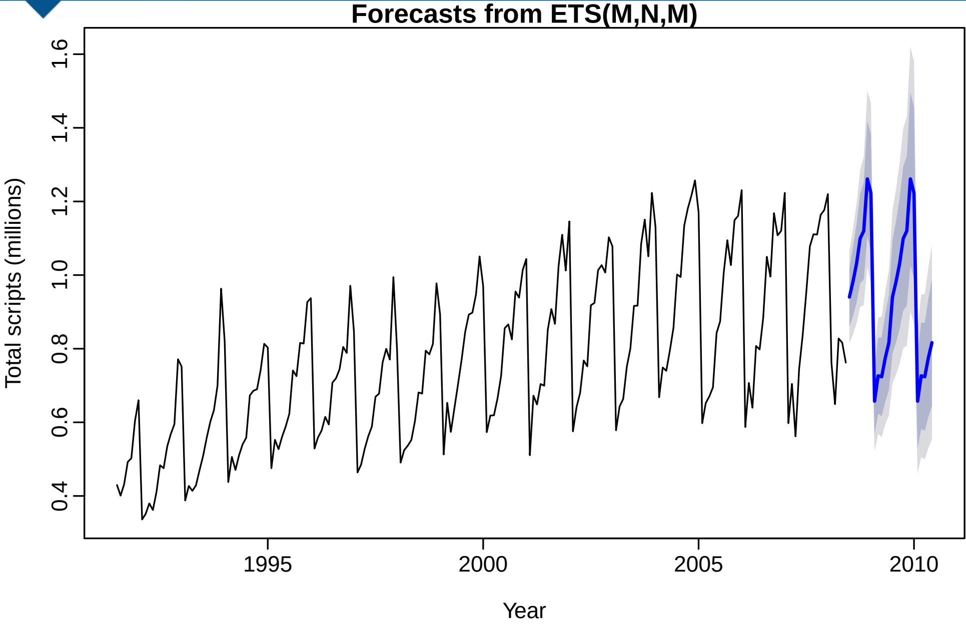
Exponential smoothing



Exponential smoothing

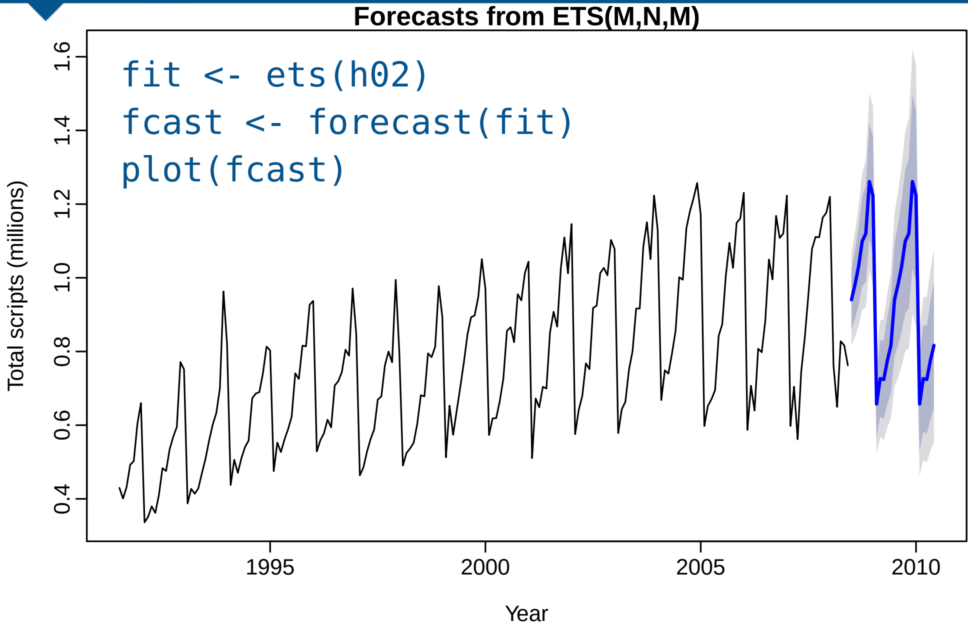


Exponential smoothing



Exponential smoothing

```
fit <- ets(h02)
fcast <- forecast(fit)
plot(fcast)
```

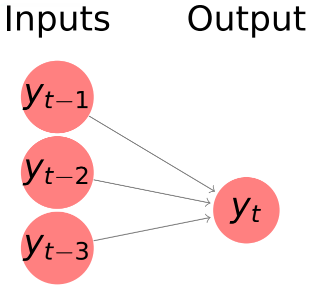


M3 comparisons

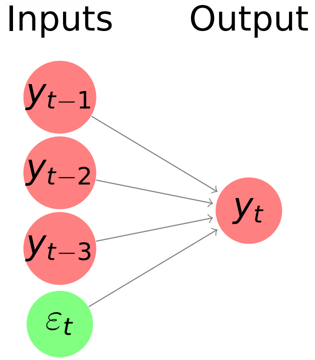
Method	MAPE	sMAPE	MASE
ForecastPro	18.00	13.06	1.47
ForecastX	17.35	13.09	1.42
ETS	17.38	13.13	1.43

Outline

ARIMA models

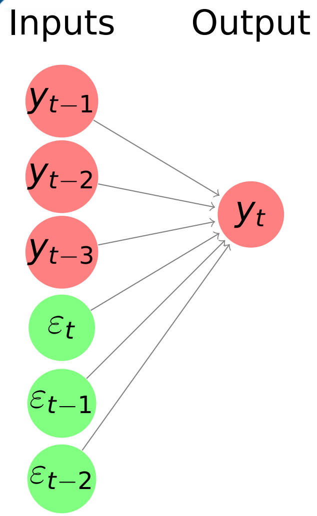


ARIMA models



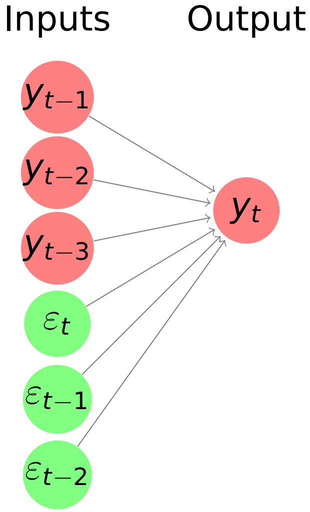
Autoregression (AR) model

ARIMA models



Autoregression moving average (ARMA) model

ARIMA models

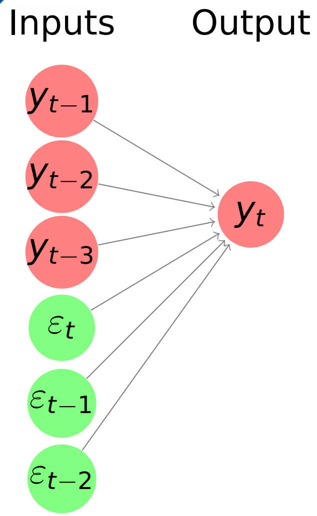


Autoregression moving average (ARMA) model

ARIMA model

Autoregression moving average (ARMA) model applied to differences.

ARIMA models



Autoregression moving average (ARMA) model

ARIMA model

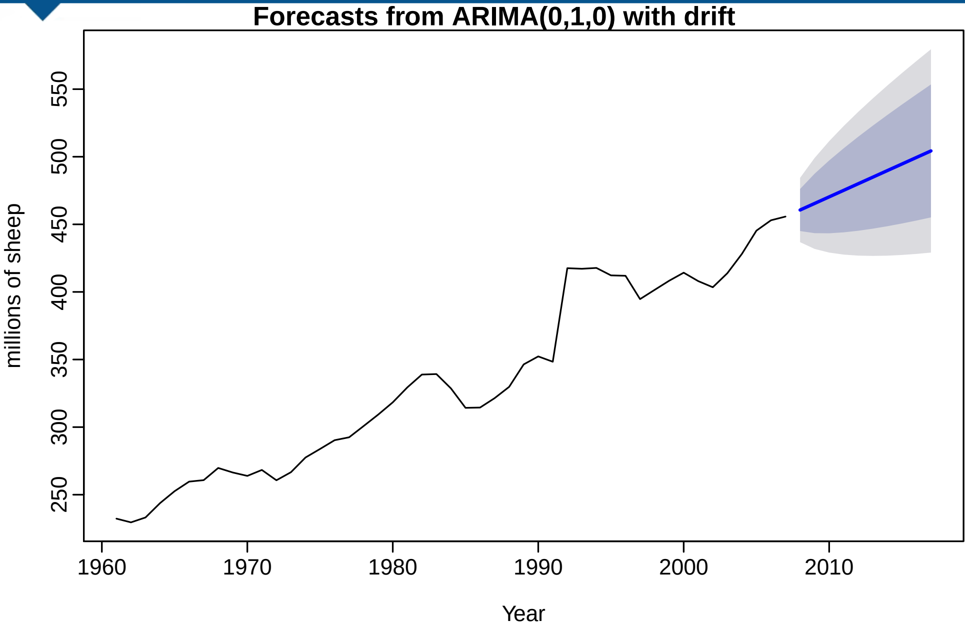
Autoregression moving average (ARMA) model applied to differences.

Estimation

Compute likelihood L from $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T$.

Use optimization algorithm to maximize L.

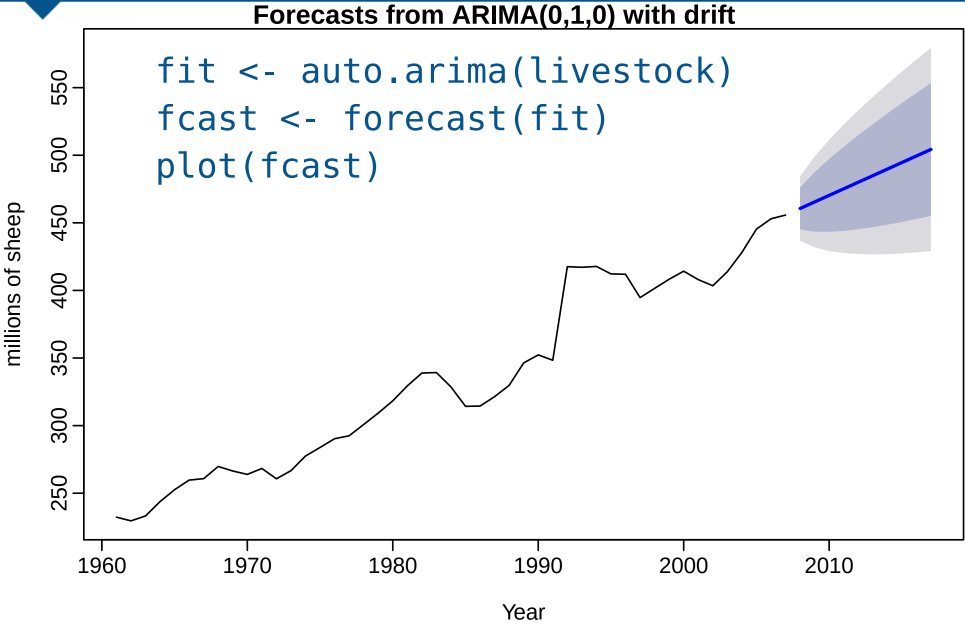
Auto ARIMA



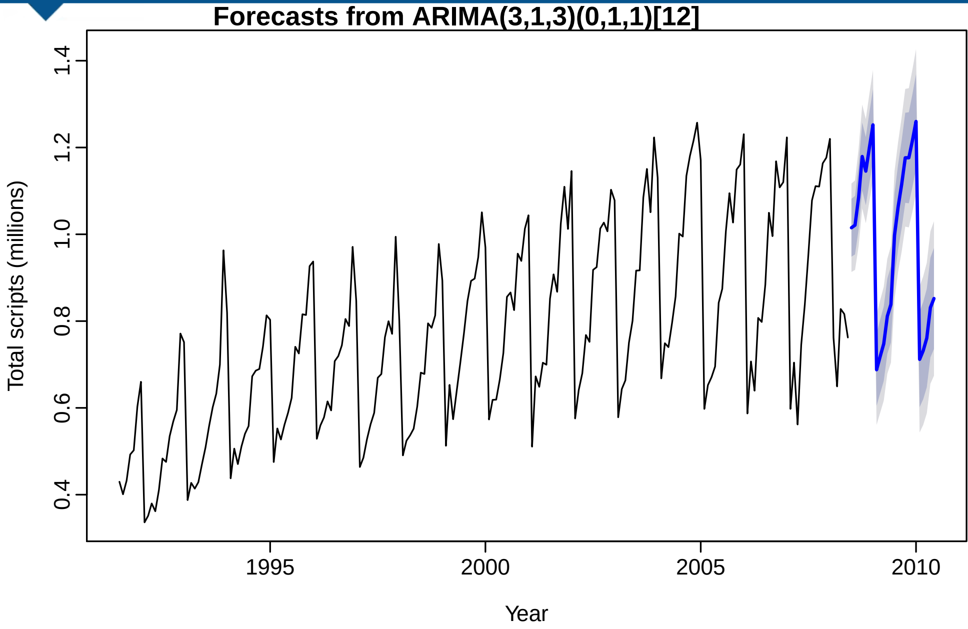
Forecasts from ARIMA(0,1,0) with drift
millions of sheep
550
500
450
400
350
300
250
1960
1970
1980
1990
2000
2010
Year

Auto ARIMA

```
fit <- auto.arima(livestock)
fcast <- forecast(fit)
plot(fcast)
```

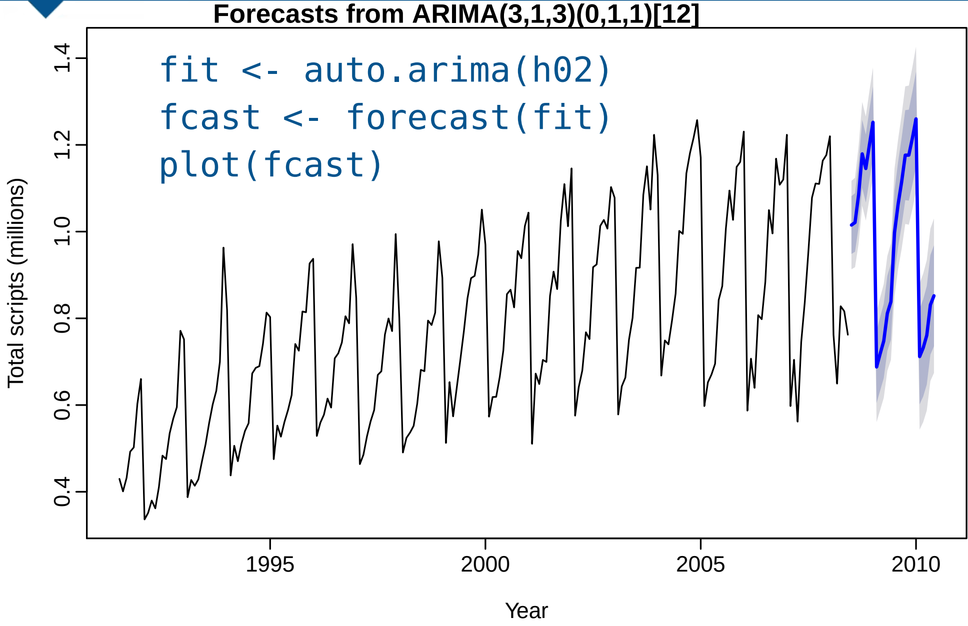


Auto ARIMA



Forecasts from ARIMA(3,1,3)(0,1,1)[12]
Total scripts (millions)
0.4
0.6
0.8
1.0
1.2
1.4
1995
2000
2005
2010
Year

Auto ARIMA




```
fit <- auto.arima(h02)
fcast <- forecast(fit)
plot(fcast)
```

How does auto.arima() work?

- Number of differences selected using unit root tests.
- Number of autoregressive and moving average terms selected by minimizing AICc.
- Inclusion of constant/drift determined by minimizing AICc.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.

How does auto.arima() work?

- Number of differences selected using unit root tests.
- Number of autoregressive and moving average terms selected by minimizing AICc.
- Inclusion of constant/drift determined by minimizing AICc.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.

How does auto.arima() work?

- Number of differences selected using unit root tests.
- Number of autoregressive and moving average terms selected by minimizing AICc.
- Inclusion of constant/drift determined by minimizing AICc.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.

How does auto.arima() work?

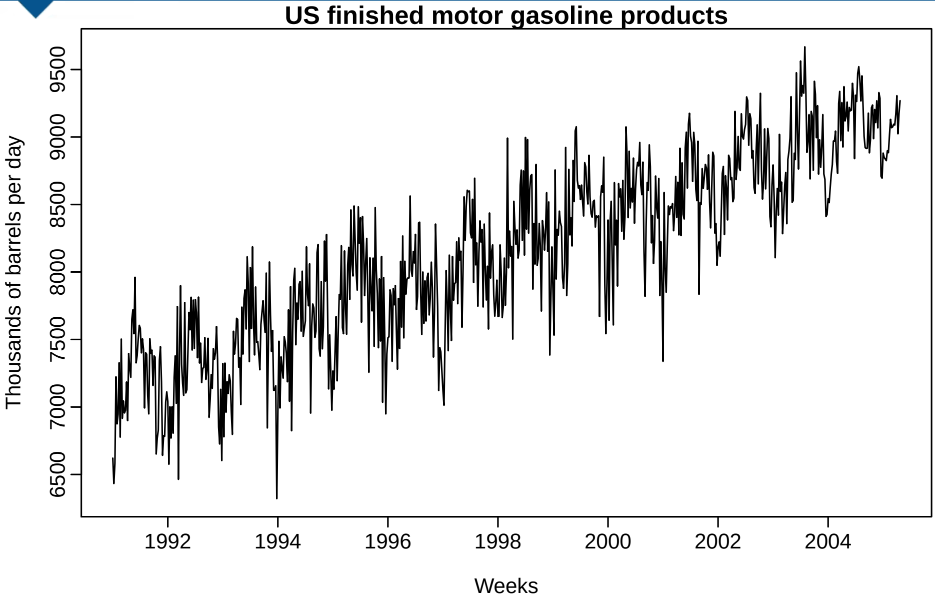
- Number of differences selected using unit root tests.
- Number of autoregressive and moving average terms selected by minimizing AICc.
- Inclusion of constant/drift determined by minimizing AICc.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.

M3 comparisons

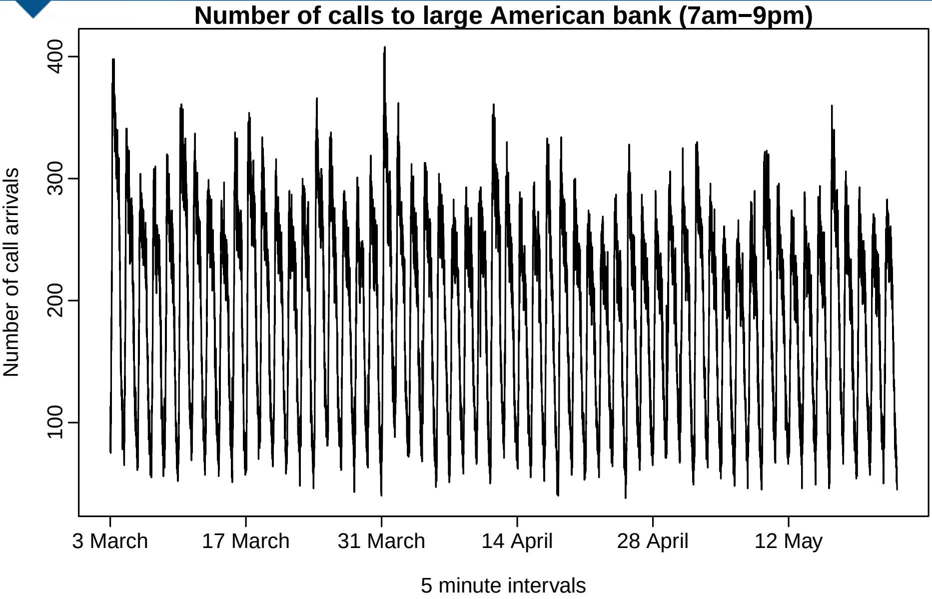
Method	MAPE	sMAPE	MASE
ForecastPro	18.00	13.06	1.47
ForecastX	17.35	13.09	1.42
ETS	17.38	13.13	1.43
AutoARIMA	19.12	13.85	1.47
ETS-ARIMA	17.92	13.02	1.44

Outline

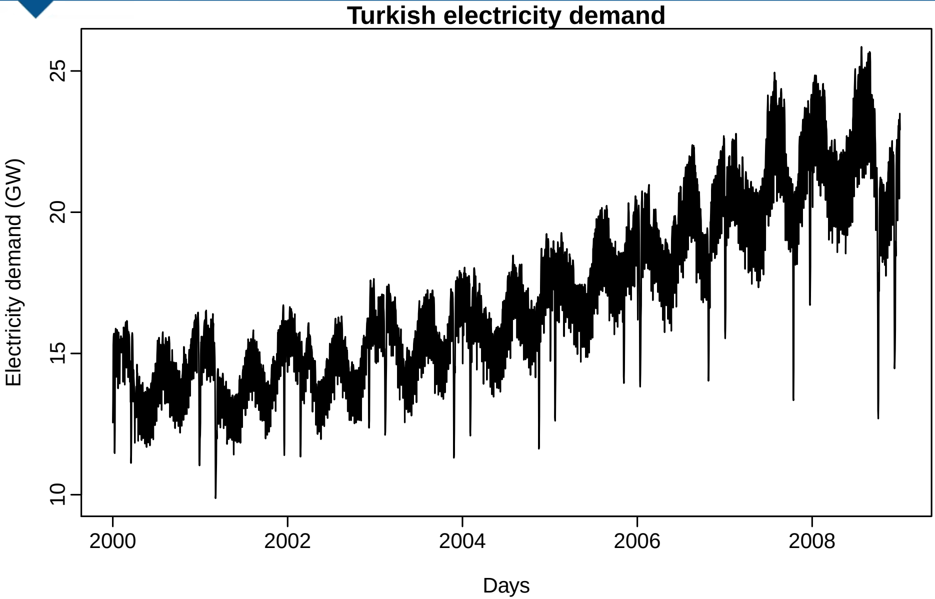
Examples


US finished motor gasoline products
Thousands of barrels per day
6500
7000
7500
8000
8500
9000
9500
1992
1994
1996
1998
2000
2002
2004
Weeks

Examples


Number of calls to large American bank (7am−9pm)
Number of call arrivals
400
300
200
100
3 March
17 March
31 March
14 April
28 April
12 May
5 minute intervals

Examples


Turkish electricity demand
Electricity demand (GW)
10
15
20
25
2000
2002
2004
2006
2008
Days

TBATS model

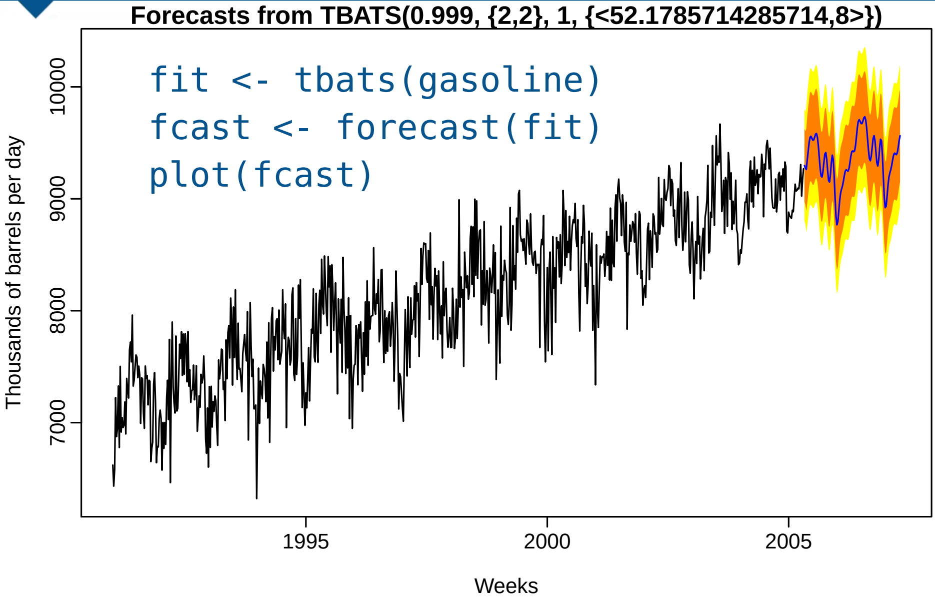
TBATS

Trigonometric terms for seasonality
Box-Cox transformations for heterogeneity
ARMA errors for short-term dynamics
Trend (possibly damped)
Seasonal (including multiple and non-integer periods)

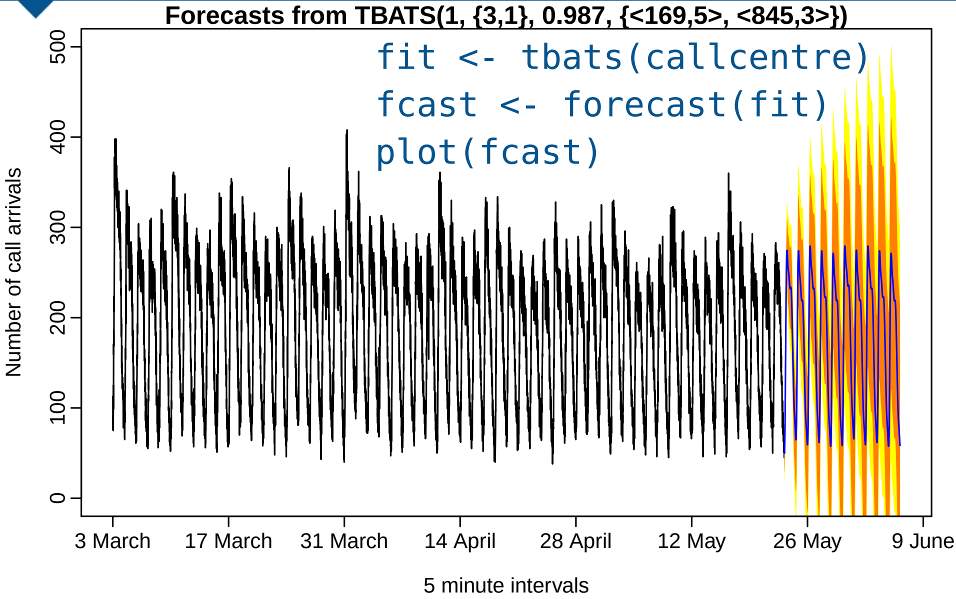
Automatic algorithm described in AM De Livera, RJ Hyndman, and RD Snyder (2011). "Forecasting time series with complex seasonal patterns using exponential smoothing". *Journal of the American Statistical Association* **106**(496), 1513–1527.

Examples

```
fit <- tbats(gasoline)
fcast <- forecast(fit)
plot(fcast)
```



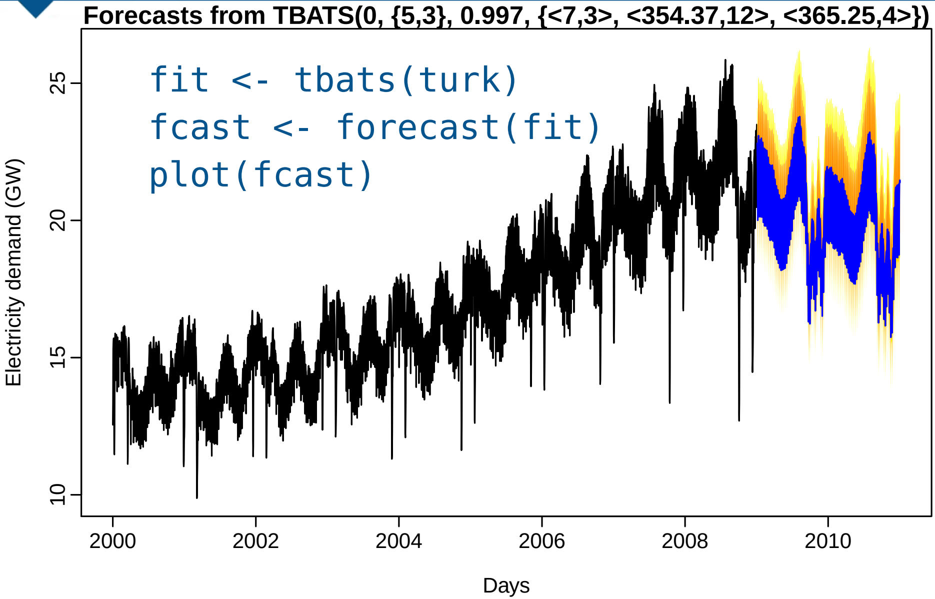
Examples




```r
fit <- tbats(callcentre)
fcast <- forecast(fit)
plot(fcast)
```

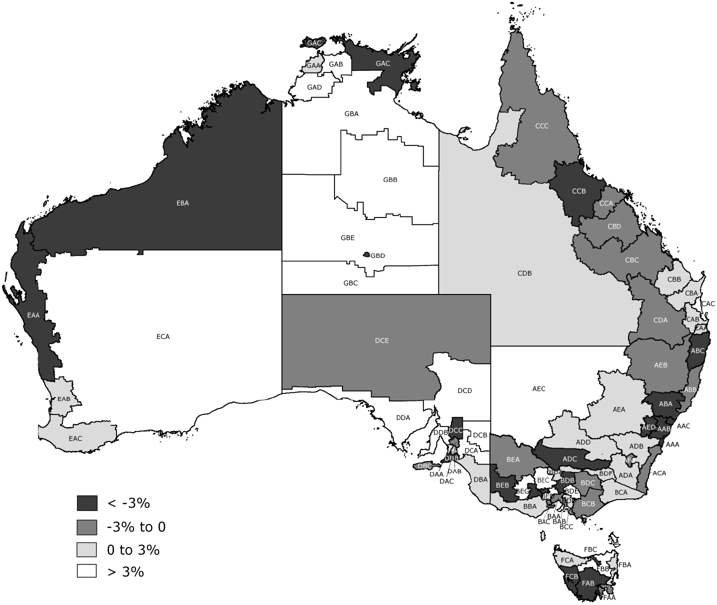
Examples

```
fit <- tbats(turk)
fcast <- forecast(fit)
plot(fcast)
```



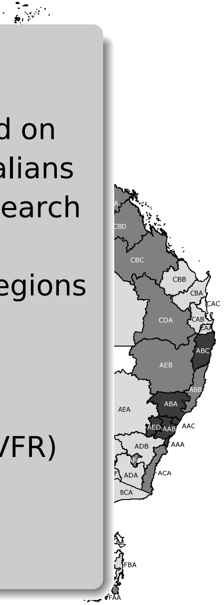
Outline

Australian tourism demand



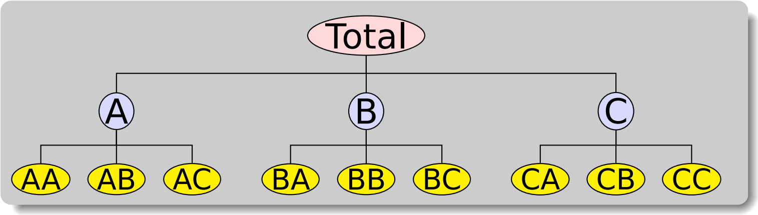
Australian tourism demand

- 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



Hierarchical time series

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

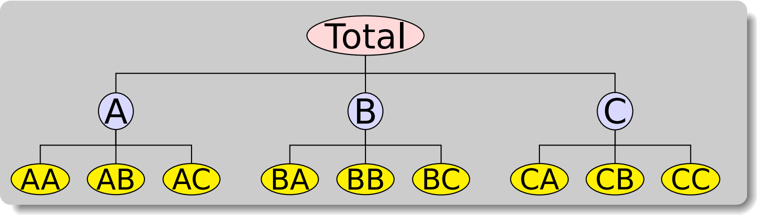


Example

- Tourism by state and region

Hierarchical time series

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

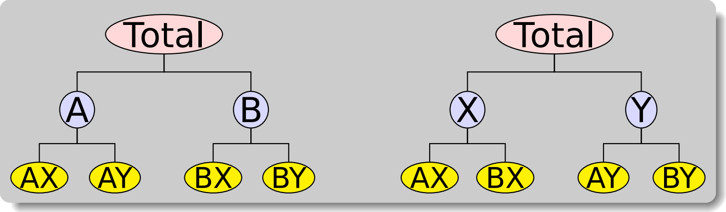


Example

- Tourism by state and region

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.

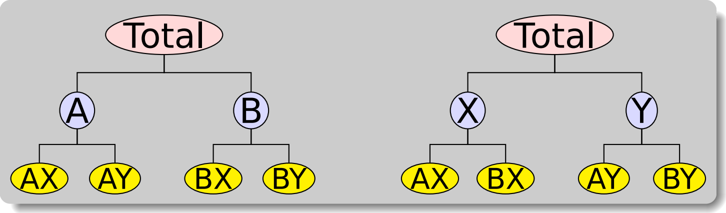


Example

- Tourism by state and purpose of travel

Grouped time series

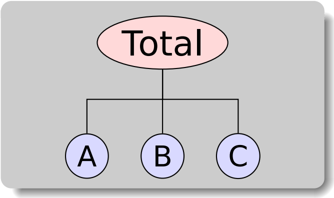
A **grouped time series** is a collection of time series that can be grouped together in a number of non-hierarchical ways.



Example

- Tourism by state and purpose of travel

Hierarchical time series

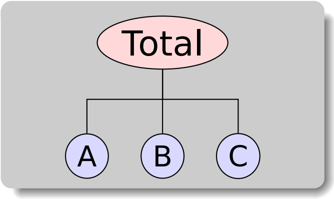


y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

Hierarchical time series

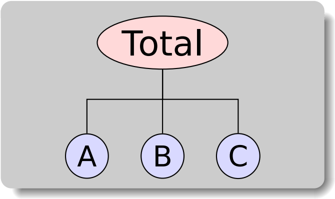


y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

Hierarchical time series



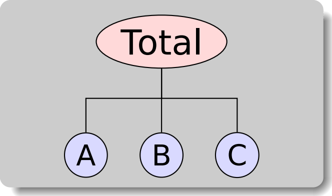
y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

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

Hierarchical time series



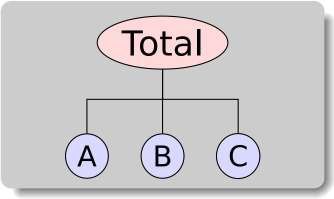
y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}$$

Hierarchical time series



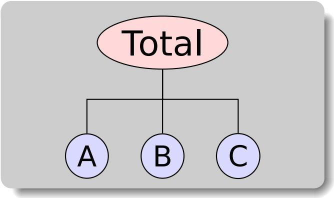
y_t : observed aggregate of all series at time t.

$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

Hierarchical time series



y_t : observed aggregate of all series at time t.

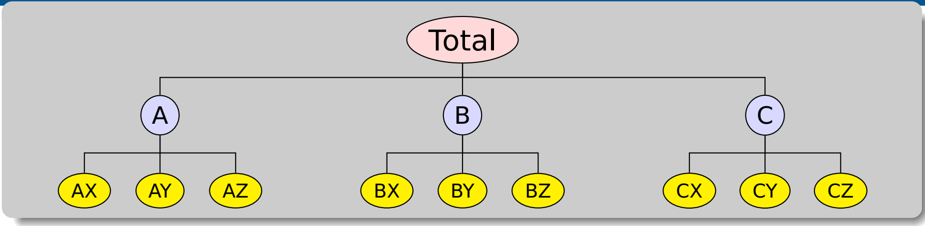
$y_{X,t}$: observation on series X at time t.

$\boldsymbol{b}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{y}_t = [y_t, y_{A,t}, y_{B,t}, y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{A,t} \\ y_{B,t} \\ y_{C,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

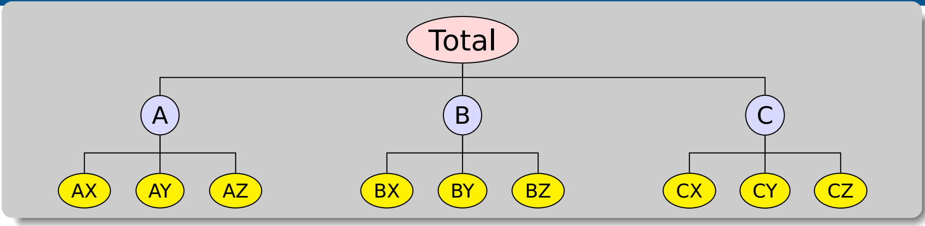
$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

Hierarchical time series



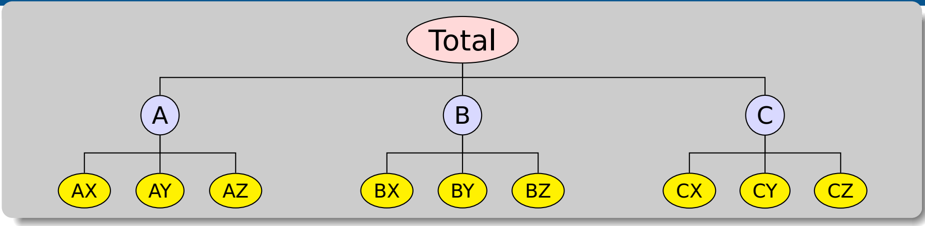
$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

Hierarchical time series



$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

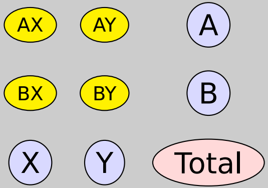
Hierarchical time series



$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{AZ,t} \\ y_{BX,t} \\ y_{BY,t} \\ y_{BZ,t} \\ y_{CX,t} \\ y_{CY,t} \\ y_{CZ,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

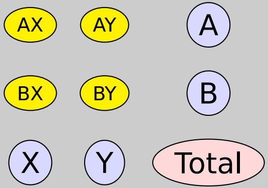
$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$$

Grouped data



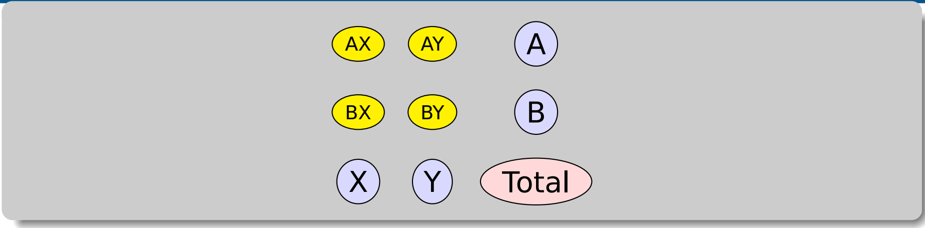
$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{X,t} \\ y_{Y,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

Grouped data



$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{X,t} \\ y_{Y,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

Grouped data



$$\boldsymbol{y}_t = \begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{X,t} \\ y_{Y,t} \\ y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} y_{AX,t} \\ y_{AY,t} \\ y_{BX,t} \\ y_{BY,t} \end{pmatrix}}_{\boldsymbol{b}_t}$$

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

Hierarchical and grouped time series

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

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

where

- $\boldsymbol{y}_t$ is a vector of all series at time t
- $\boldsymbol{b}_t$ is a vector of the most disaggregated series at time t
- $\boldsymbol{S}$ is a "summing matrix" containing the aggregation constraints.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (They may not add up.)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{P}$.

- $\boldsymbol{P}$ extracts and combines base forecasts $\hat{\boldsymbol{y}}_n(h)$ to get bottom-level forecasts.

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (They may not add up.)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{P}$.

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

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (They may not add up.)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{P}$.

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

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (They may not add up.)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{P}$.

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

Forecasting notation

Let $\hat{\boldsymbol{y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{y}_t$. (They may not add up.)

Reconciled forecasts must be of the form:

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

for some matrix $\boldsymbol{P}$.

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

General properties

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

Bias

Revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

Variance

The error variance of the revised forecasts is

$$\text{Var}[\boldsymbol{y}_{n+h} - \tilde{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \boldsymbol{SPW}_h\boldsymbol{P}'\boldsymbol{S}'$$

$$\text{where} \quad \boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{n+h} - \hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

is the error variance of the base forecasts $\hat{\boldsymbol{y}}_n(h)$

General properties

$$\tilde{\boldsymbol{y}}_n(h) = \boldsymbol{S}\boldsymbol{P}\hat{\boldsymbol{y}}_n(h)$$

Bias

Revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

Variance

The error variance of the revised forecasts is

$$\text{Var}[\boldsymbol{y}_{n+h} - \tilde{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \boldsymbol{S}\boldsymbol{P}\boldsymbol{W}_h\boldsymbol{P}'\boldsymbol{S}'$$

$$\text{where} \quad \boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{n+h} - \hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

is the error variance of the base forecasts $\hat{\boldsymbol{y}}_n(h)$

General properties

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

Bias

Revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

Variance

The error variance of the revised forecasts is

$$\text{Var}[\boldsymbol{y}_{n+h} - \tilde{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n] = \boldsymbol{SPW}_h\boldsymbol{P}'\boldsymbol{S}'$$

$$\text{where} \quad \boldsymbol{W}_h = \text{Var}[\boldsymbol{y}_{n+h} - \hat{\boldsymbol{y}}_n(h) \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$$

is the error variance of the base forecasts $\hat{\boldsymbol{y}}_n(h)$

Optimal forecast reconciliation

Theorem

For any $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, then

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{S}\boldsymbol{P}\boldsymbol{W}_h\boldsymbol{P}'\boldsymbol{S}']$$

has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}$.

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate, especially for $h > 1$.

Optimal forecast reconciliation

Theorem

For any $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, then

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{S}\boldsymbol{P}\boldsymbol{W}_h\boldsymbol{P}'\boldsymbol{S}']$$

has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}$.

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate, especially for $h > 1$.

Optimal forecast reconciliation

Theorem

For any $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, then

$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{SPW}_h\boldsymbol{P}'\boldsymbol{S}']$$

has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}$.

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate, especially for $h > 1$.

Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal combination forecasts

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

Revised forecasts — Base forecasts

Solution: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Optimal combination forecasts

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

Revised forecasts   Base forecasts

Solution: WLS

- Suppose we approximate $\boldsymbol{W}_1$ by its diagonal and assume that $\boldsymbol{W}_h \propto \boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

hts package for R

hts: Hierarchical and grouped time series

Methods for analysing and forecasting hierarchical and grouped time series

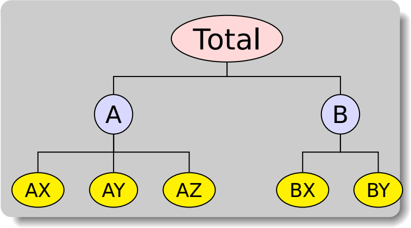
Version:	4.5
Depends:	forecast ($\geq$ 5.0), SparseM
Imports:	parallel, utils
Published:	2015-06-29
Author:	Rob J Hyndman, Earo Wang and Alan Lee with contributions from Shanika Wickramasuriya
Maintainer:	Rob J Hyndman <Rob.Hyndman at monash.edu>
BugReports:	`https://github.com/robjhyndman/hts/issues`
License:	GPL ($\geq$ 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)))
```

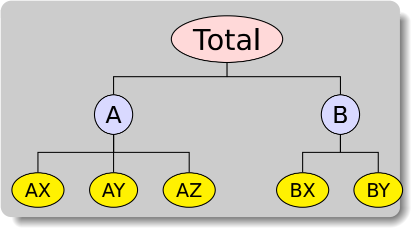
Example using R

```
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)))
```



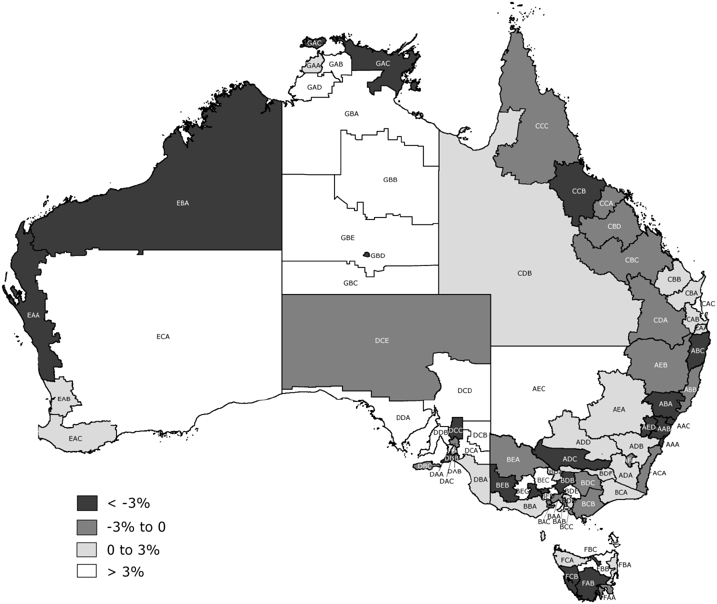
Example using R

```
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)))
```




```
summary(y)
smatrix(y)

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

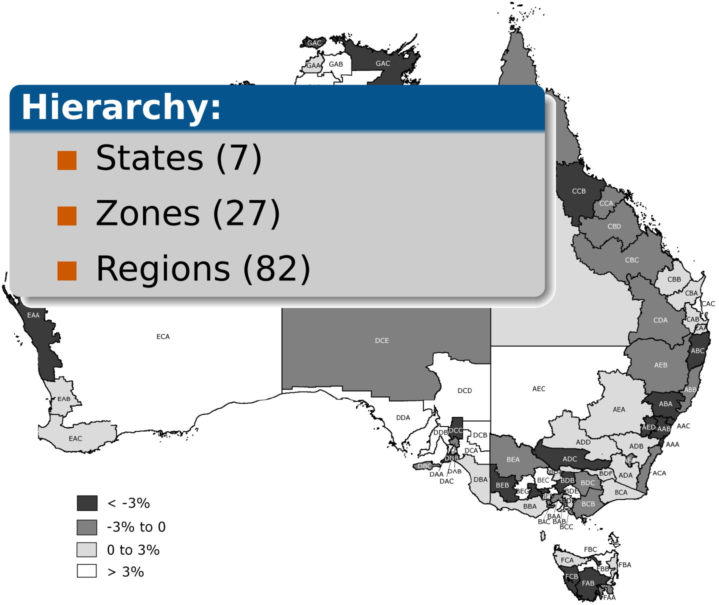
Australian tourism



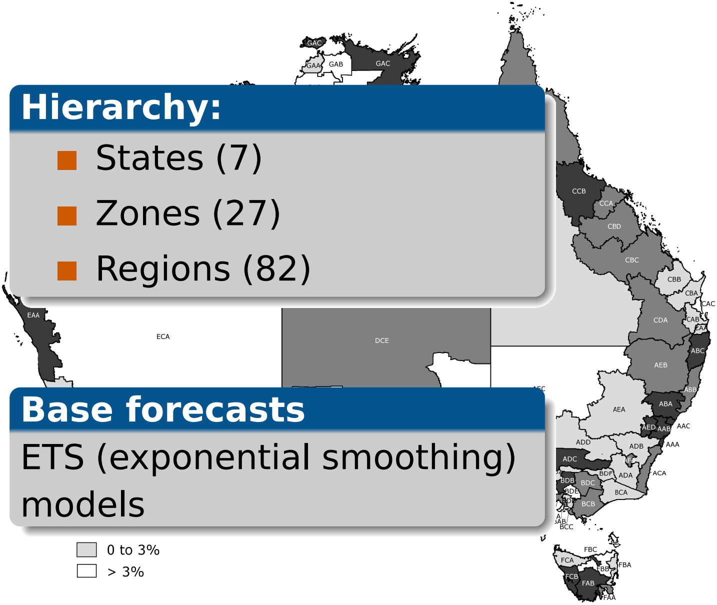
Australian tourism

Hierarchy:

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



Australian tourism



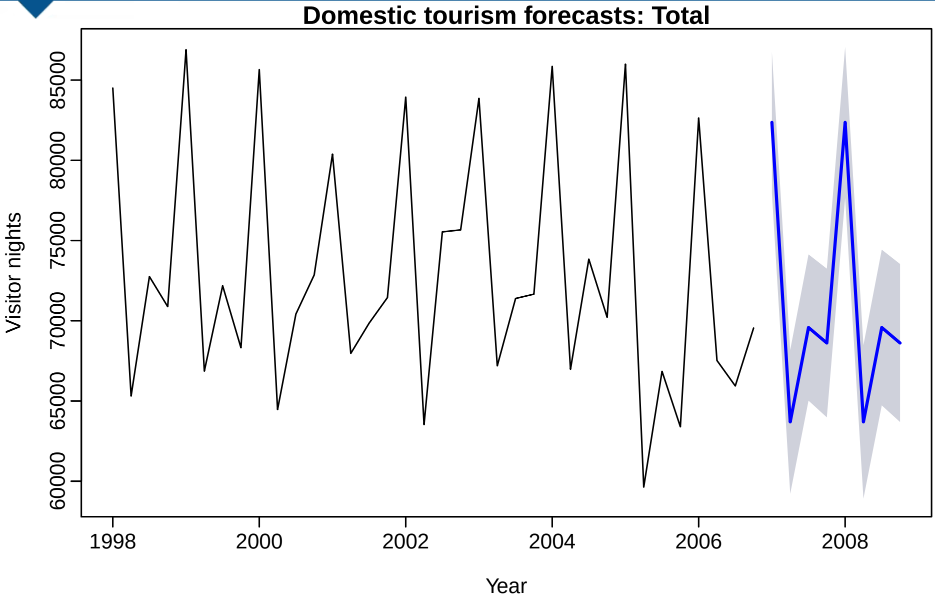
Hierarchy:

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

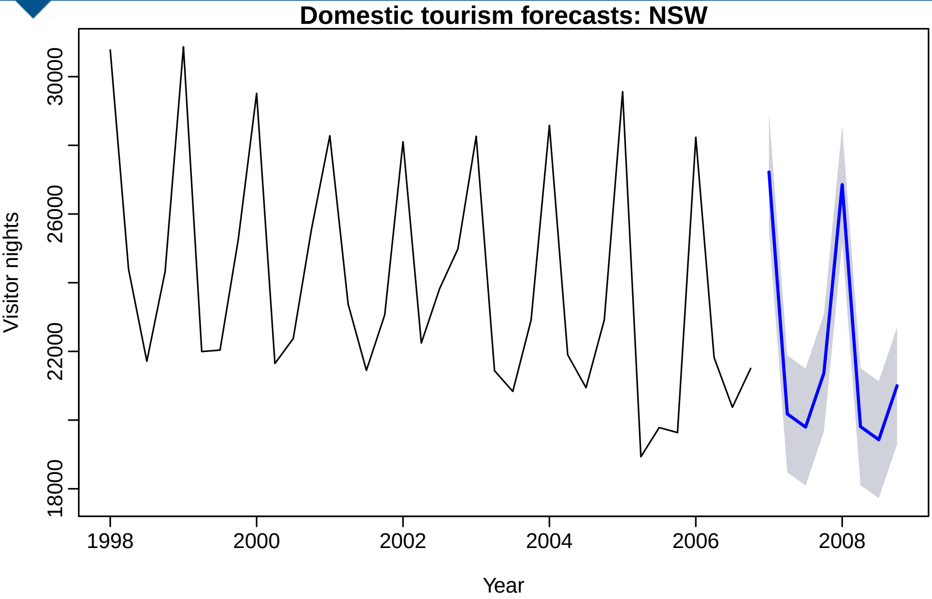
Base forecasts

ETS (exponential smoothing) models

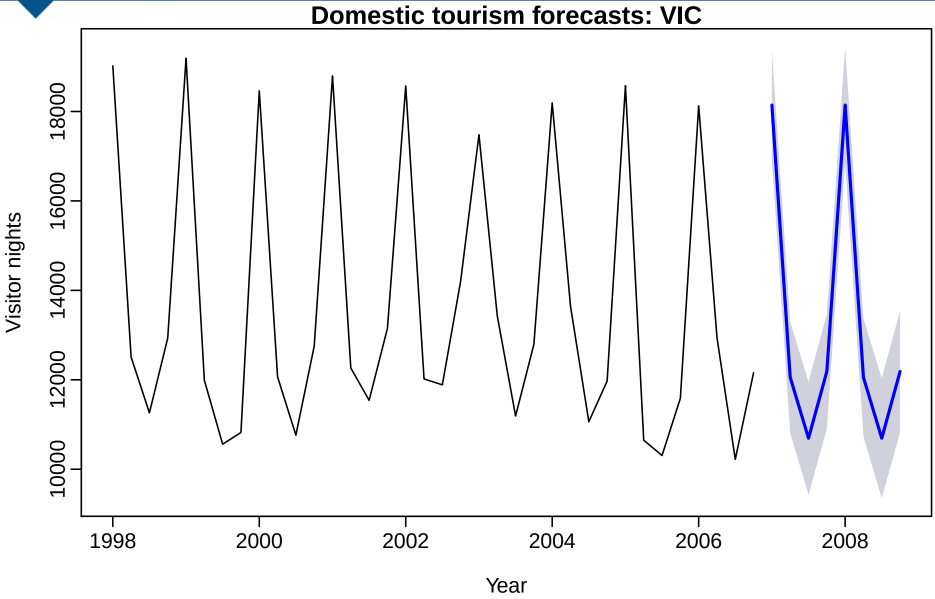
Base forecasts



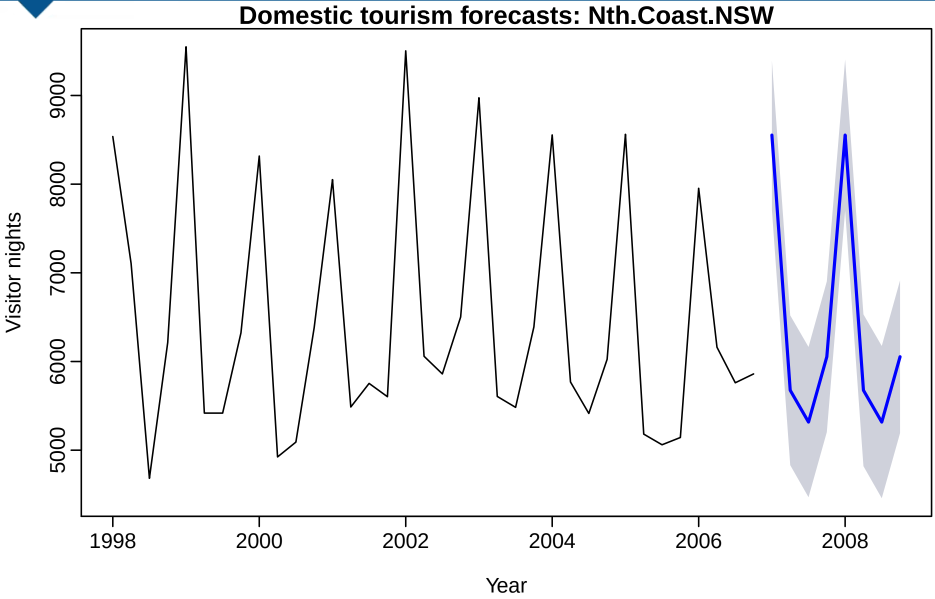
Base forecasts



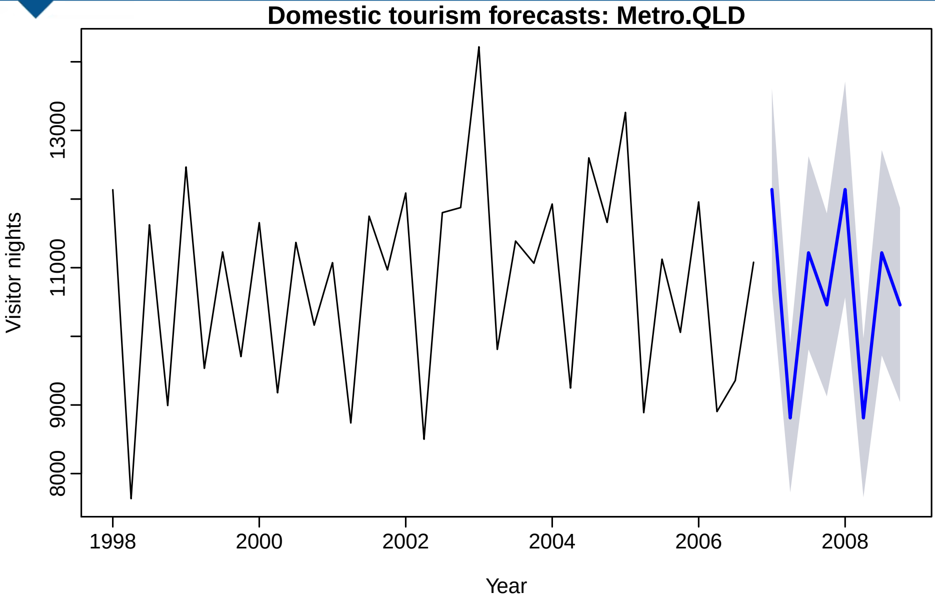
Base forecasts



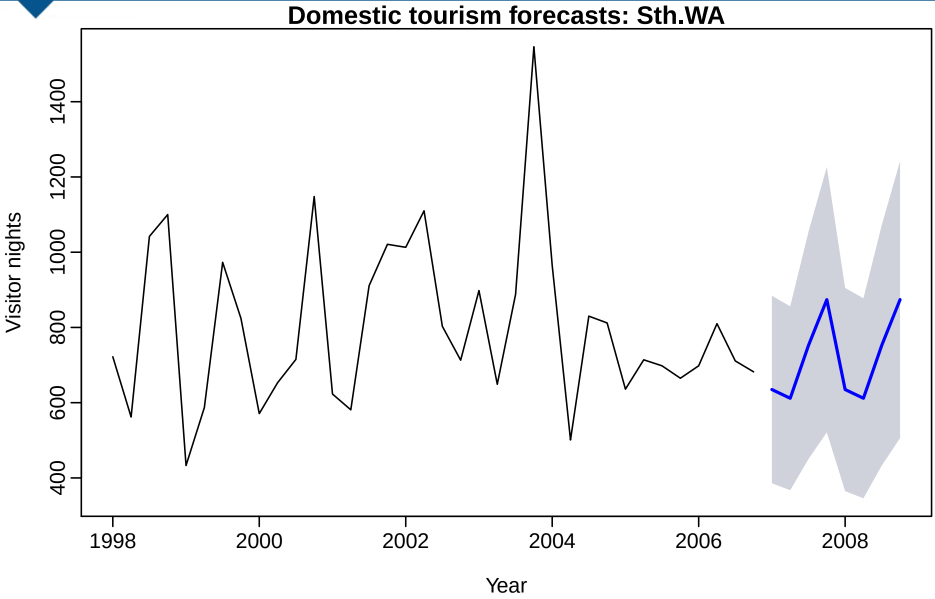
Base forecasts



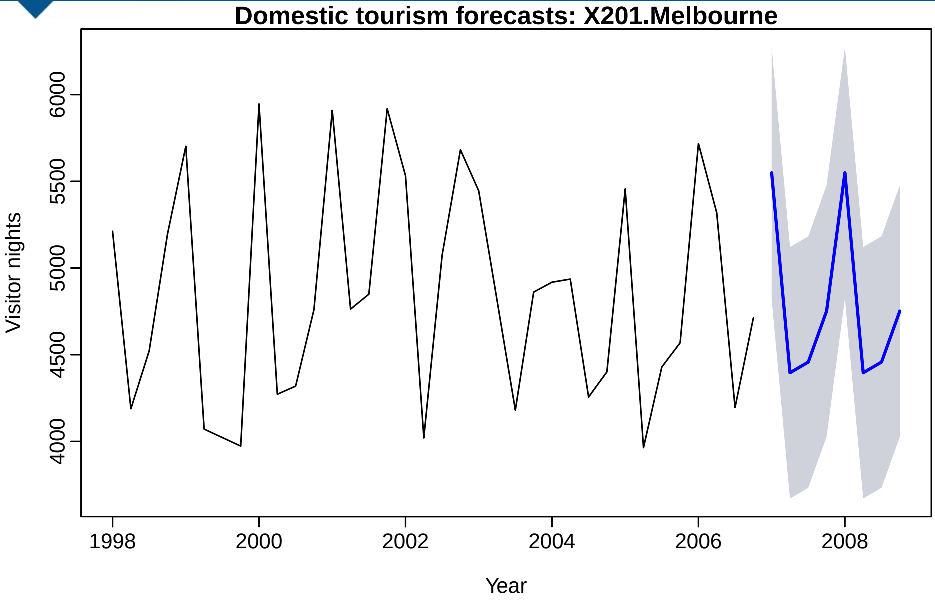
Base forecasts



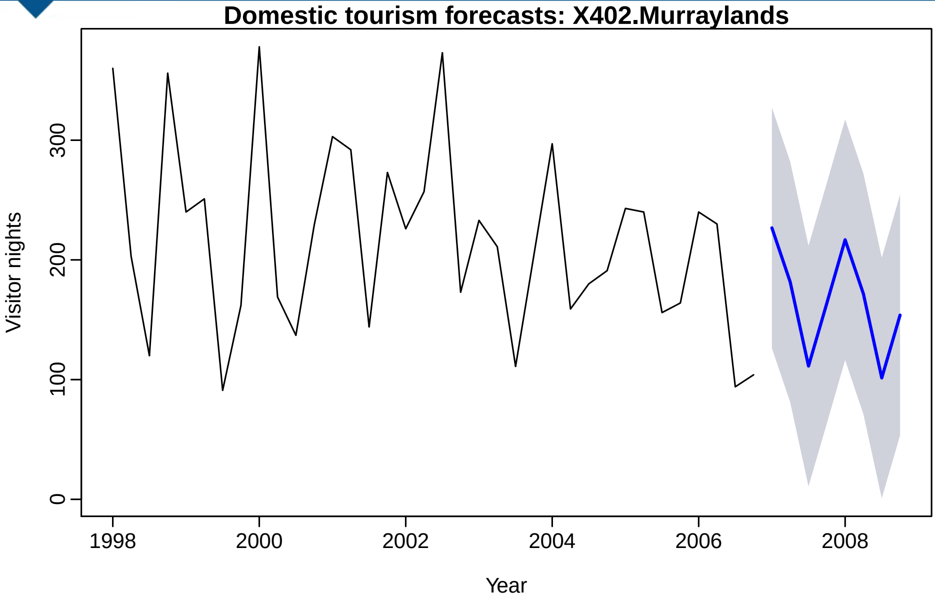
Base forecasts



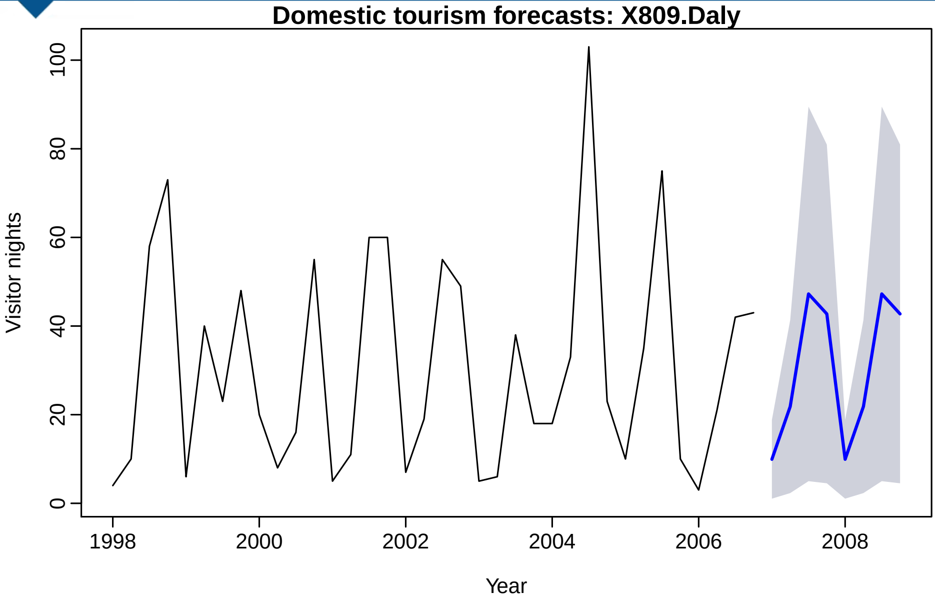
Base forecasts



Base forecasts

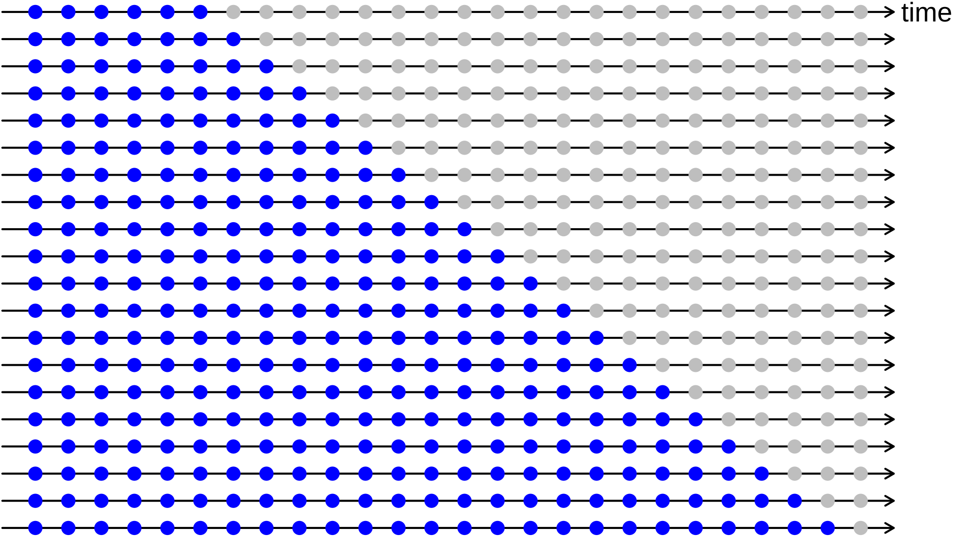


Base forecasts



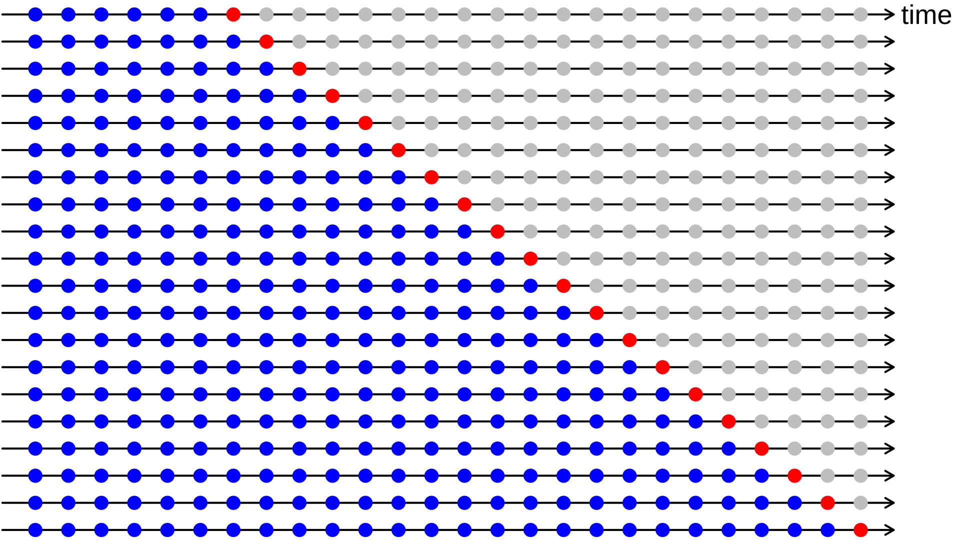
Forecast evaluation

Training sets



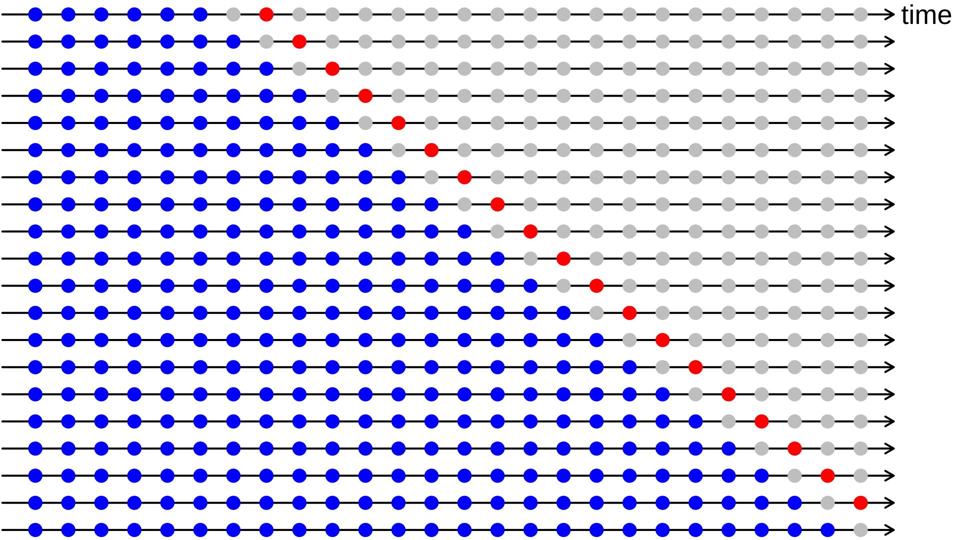
Forecast evaluation

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



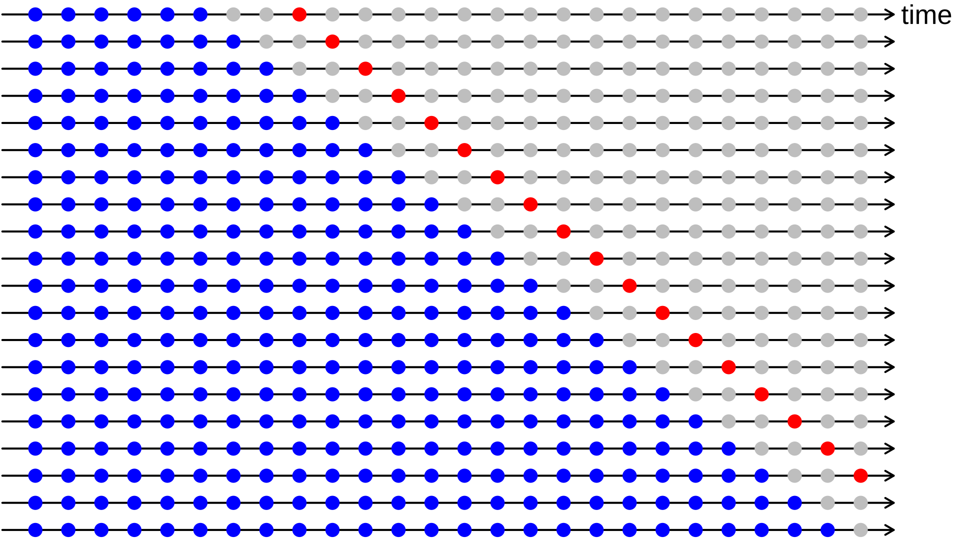
Forecast evaluation

Training sets **Test set $h = 2$**



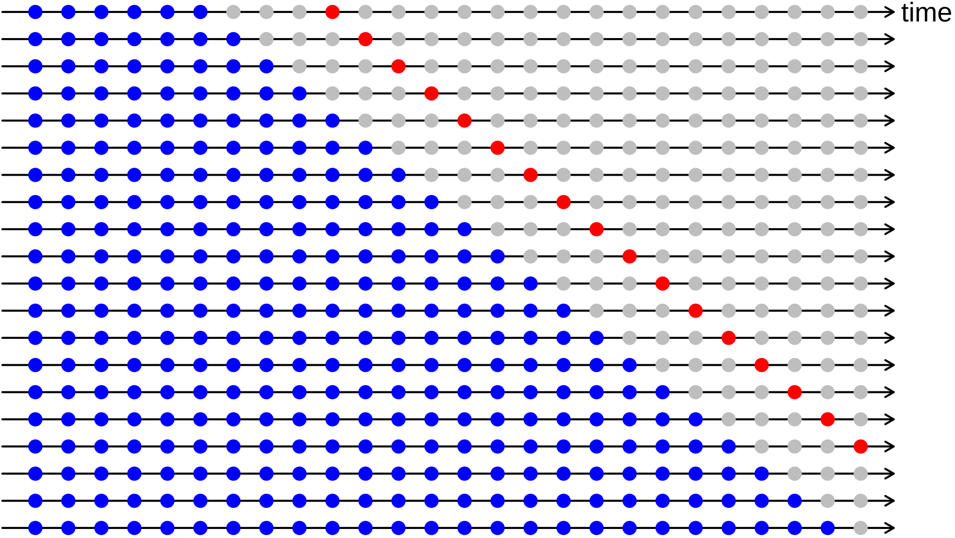
Forecast evaluation

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



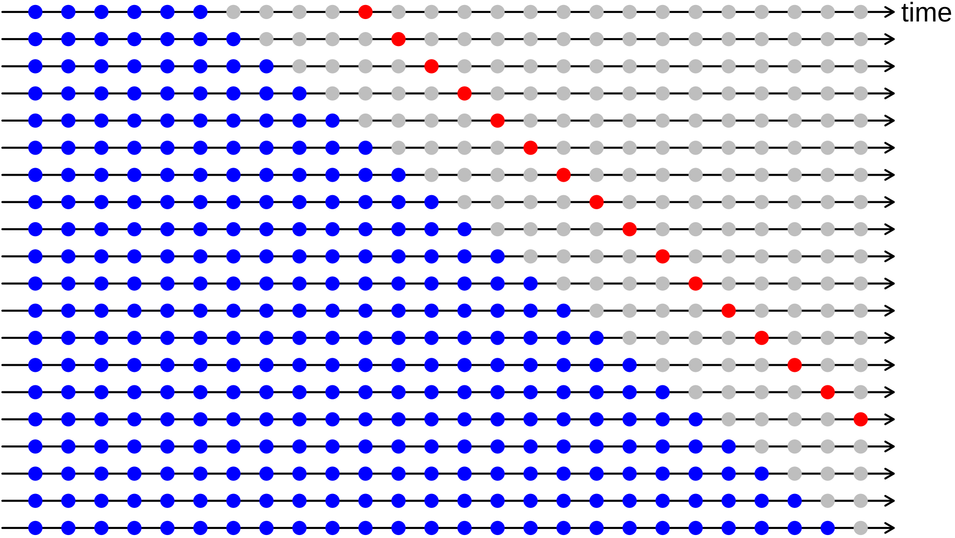
Forecast evaluation

Training sets **Test set $h = 4$**



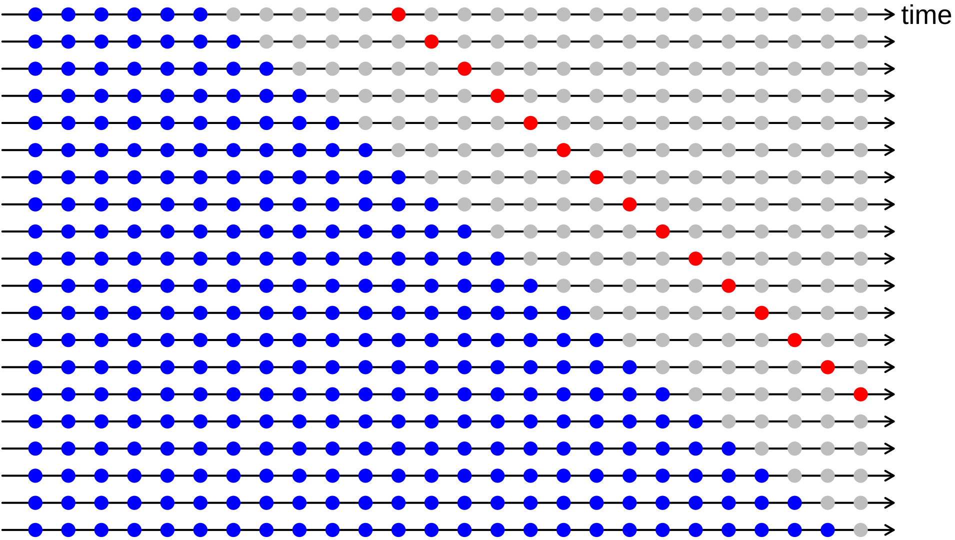
Forecast evaluation

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



Forecast evaluation

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



Hierarchy: states, zones, regions

RMSE	Forecast horizon $h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$h = 6$	**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**
WLS	1705.21	1715.87	1703.75	1729.56	1627.79	1661.24	**1690.57**
			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**
WLS	398.84	402.12	400.71	405.03	394.76	398.23	**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**
WLS	93.02	93.32	93.38	93.72	93.39	93.53	**93.39**

Outline

Forecasts about the forecast package

1. Automatic algorithms will become more general — handling a wider variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for very high dimensional time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be added.
5. Better reconciliation methods using full covariance matrix (**hts** package).

Forecasts about the forecast package

1. Automatic algorithms will become more general — handling a wider variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for very high dimensional time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be added.
5. Better reconciliation methods using full covariance matrix (**hts** package).

Forecasts about the forecast package

1. Automatic algorithms will become more general — handling a wider variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for very high dimensional time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be added.
5. Better reconciliation methods using full covariance matrix (**hts** package).

Forecasts about the forecast package

1. Automatic algorithms will become more general — handling a wider variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for very high dimensional time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be added.
5. Better reconciliation methods using full covariance matrix (**hts** package).

Forecasts about the forecast package

1. Automatic algorithms will become more general — handling a wider variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for very high dimensional time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be added.
5. Better reconciliation methods using full covariance matrix (**hts** package).

For further information

robjhyndman.com

- Slides and references for this talk.
- Links to all papers and books.
- Links to R packages.
- A blog about forecasting research.

OTexts.org/fpp

- Free online textbook on forecasting using R

For further information

robjhyndman.com

- Slides and references for this talk.
- Links to all papers and books.
- Links to R packages.
- A blog about forecasting research.

OTexts.org/fpp

- Free online textbook on forecasting using R