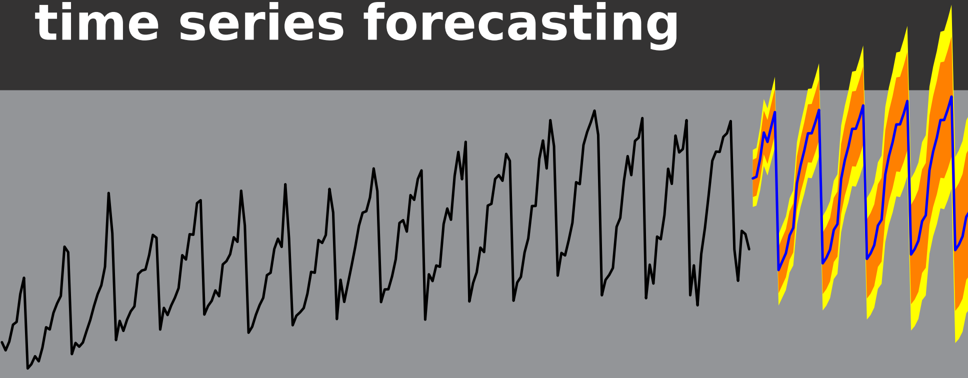
Rob J Hyndman

Advances in automatic time series forecasting



Outline

1. Motivation
2. Exponential smoothing
3. ARIMA modelling
4. Time series with complex seasonality
5. Hierarchical time series
6. Functional time series

Motivation

1. Common in business to have over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.
3. Some types of data can be decomposed into a large number of univariate time series that need to be forecast.

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 over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.
3. Some types of data can be decomposed into a large number of univariate time series that need to be forecast.

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 over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.
3. Some types of data can be decomposed into a large number of univariate time series that need to be forecast.

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 over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.
3. Some types of data can be decomposed into a large number of univariate time series that need to be forecast.

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 over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.
3. Some types of data can be decomposed into a large number of univariate time series that need to be forecast.

Specifications

Automatic forecasting algorithms must:

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

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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

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

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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

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

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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_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.
- Additive and multiplicative versions give the same point forecasts but different prediction intervals.
- Use AIC to select best model.

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.
- Additive and multiplicative versions give the same point forecasts but different prediction intervals.
- Use AIC to select best model.

General notation E T S : **E**xpone

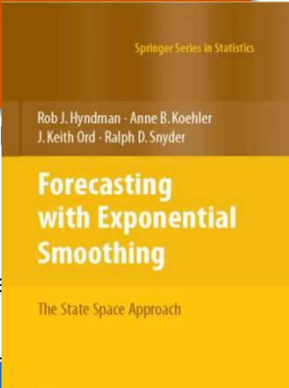
Error **T**rend **S**easonal

Examples:

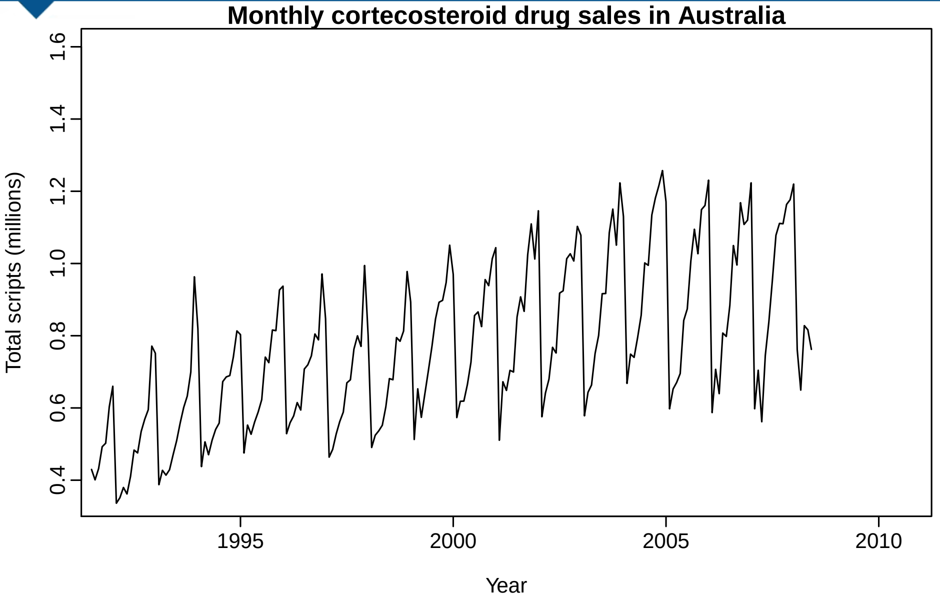
A,N,N: Simple exponential smoothing with additive

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

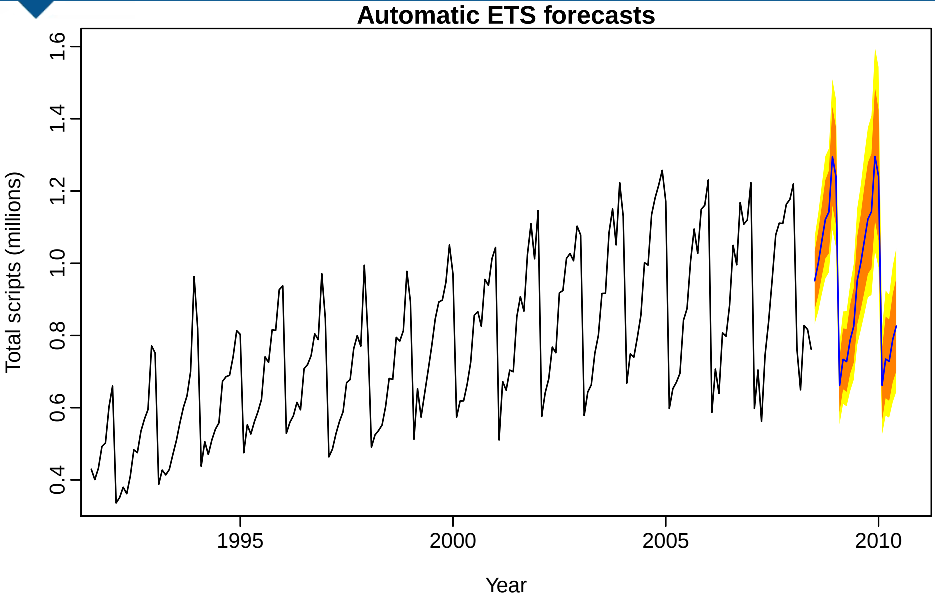
M,A,M: Multiplicative Holt-Winters' method with mu



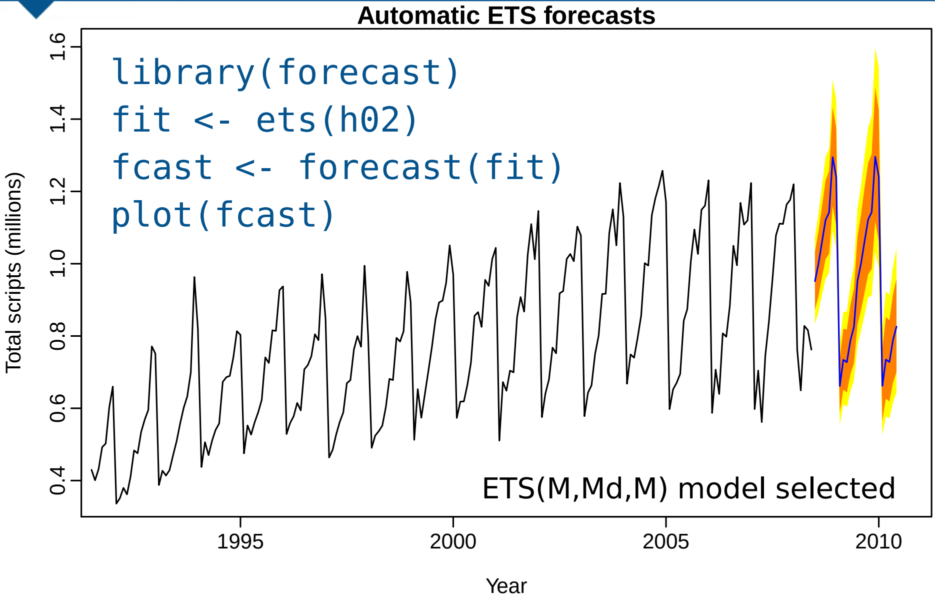
Exponential smoothing



Exponential smoothing



Exponential smoothing



Outline

How does auto.arima() work?

A seasonal ARIMA process

$$\Phi(B^m)\phi(B)(1-B)^d(1-B^m)^D y_t = c + \Theta(B^m)\theta(B)\varepsilon_t$$

Need to select appropriate orders p, q, d, P, Q, D, and whether to include c.

How does auto.arima() work?

A seasonal ARIMA process

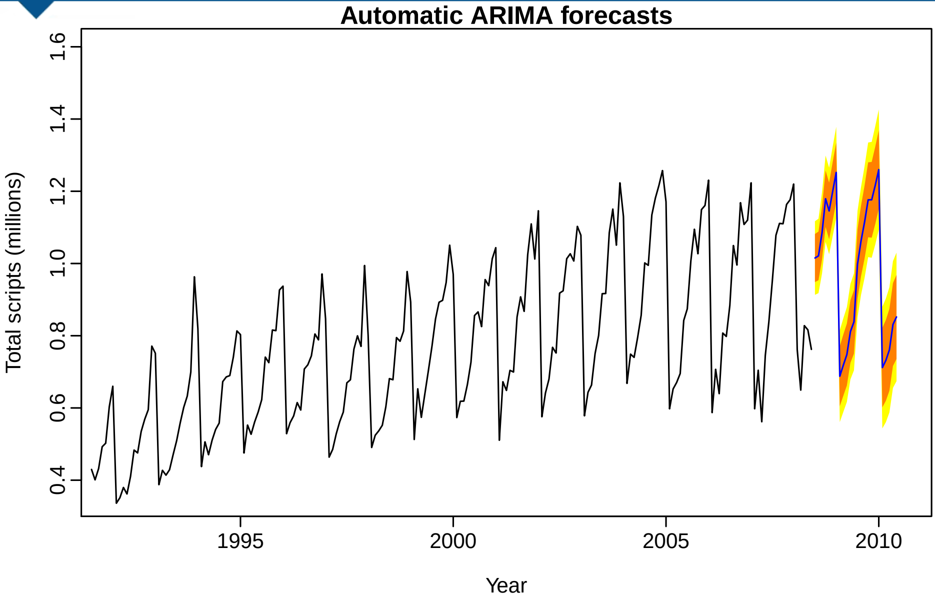
$$\Phi(B^m)\phi(B)(1-B)^d(1-B^m)^D y_t = c + \Theta(B^m)\theta(B)\varepsilon_t$$

Need to select appropriate orders p, q, d, P, Q, D, and whether to include c.

Hyndman & Khandakar (JSS, 2008) algorithm:

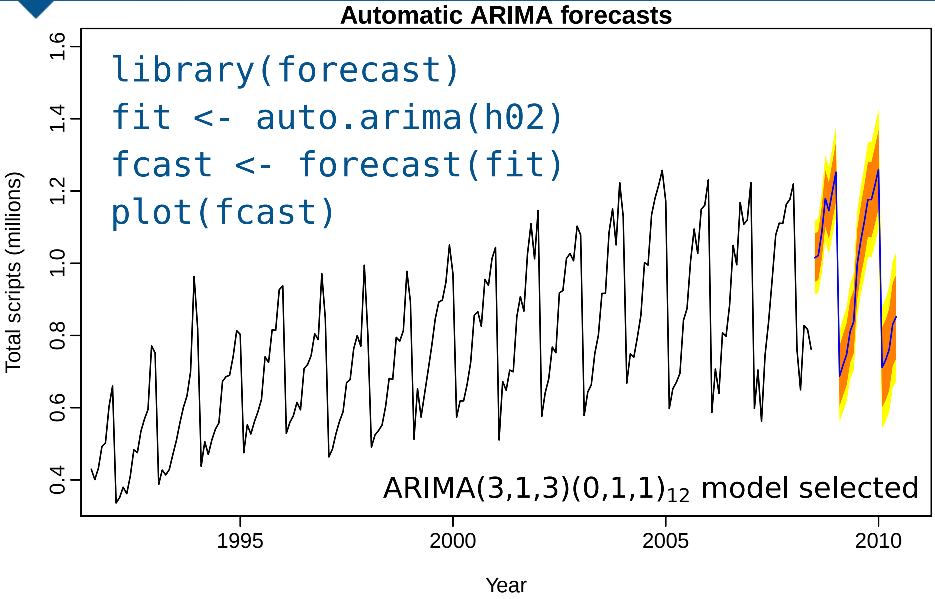
- Select no. differences d via KPSS unit root test.
- Select D using OCSB unit root test.
- Select p, q, P, Q, c by minimising AIC.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.

Auto ARIMA


Automatic ARIMA forecasts
Total scripts (millions)
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1995
2000
2005
2010
Year

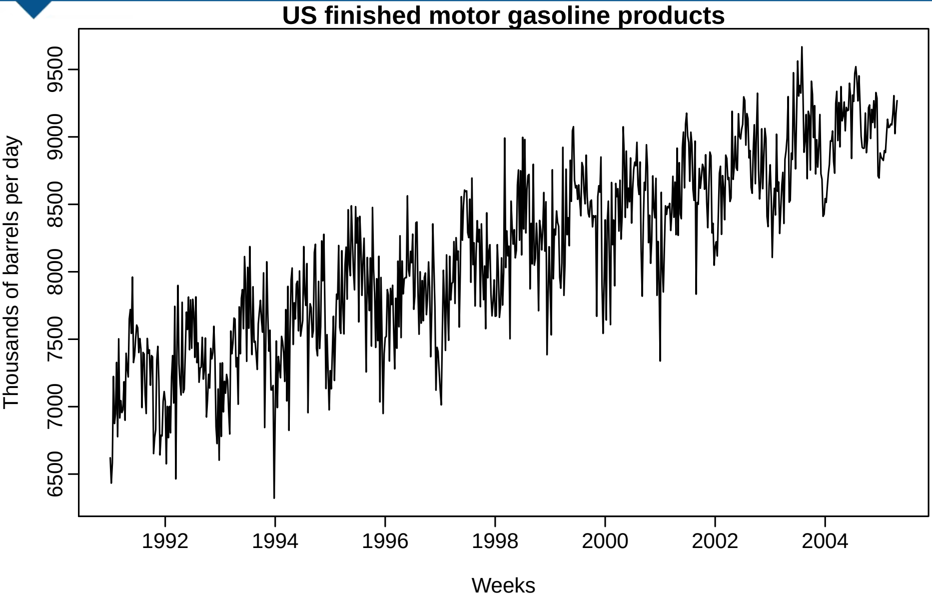
Auto ARIMA

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



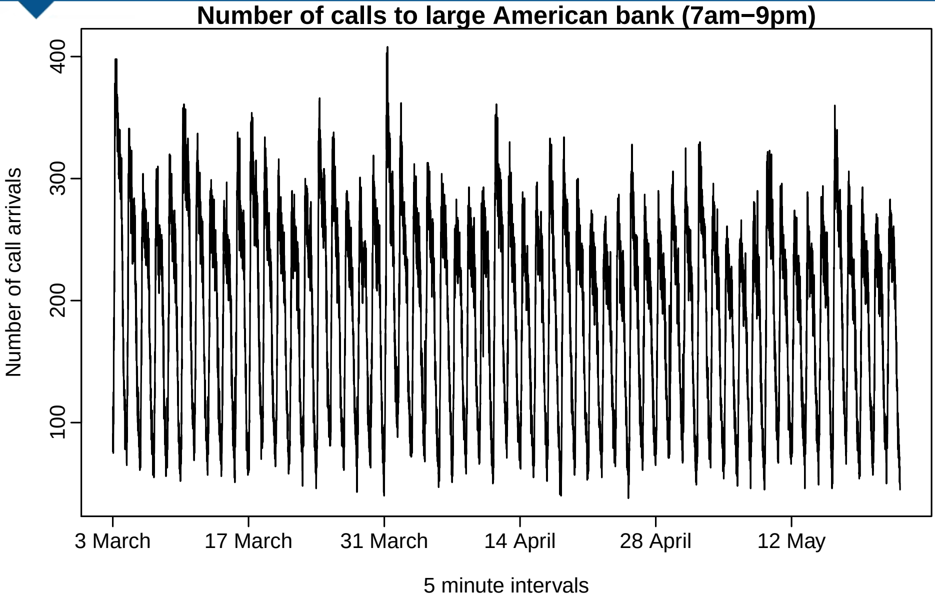
Outline

Examples

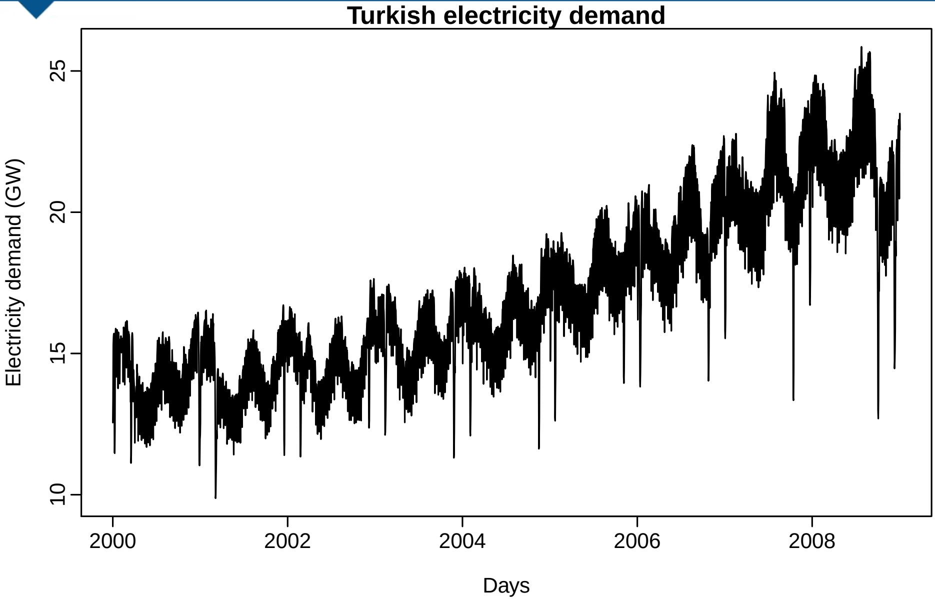


US finished motor gasoline products
Thousands of barrels per day
6500
7000
7500
8000
8500
9000
9500
1992
1994
1996
1998
2000
2002
2004
Weeks

Examples


Number of calls to large American bank (7am−9pm)
Number of call arrivals
100
200
300
400
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

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

De Livera, Hyndman, Snyder (2011). "Forecasting time series with complex seasonal patterns using exponential smoothing". *JASA*, **106**, 1513-1527.

TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

M seasonal periods

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= \quad s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

M seasonal periods

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

global and local trend

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

M seasonal periods

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

global and local trend

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

ARMA error

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$\begin{aligned} s_{j,t}^{(i)} &= s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\ s_{j,t}^{(i)} &= -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \end{aligned}$$

TBATS model

$$y_t = \text{observation at time } t$$

$$y_t^{(\omega)} = \begin{cases} (y_t^{\omega} - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$$

Box-Cox transformation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$$

M seasonal periods

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t$$

global and local trend

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$$

ARMA error

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \mathrm{c}$$

$$s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t$$

Fourier-like seasonal terms

TBATS model

$y_t =$ observation at time t

$$y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \text{lo} \end{cases}$$

TBATS

Trigonometric

Box-Cox

ARMA

Trend

Seasonal

$y_t^{(\omega)} = \ell_{t-1}$

$\ell_t = \ell_{t-1}$

$b_t = (1 -$

$d_t = \sum_{i=1}^{p}$

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$$

$s_{j,t}^{(i)} = \quad s_{j,t-1}^{(i)}$ c

$$s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t$$

Box-Cox transformation

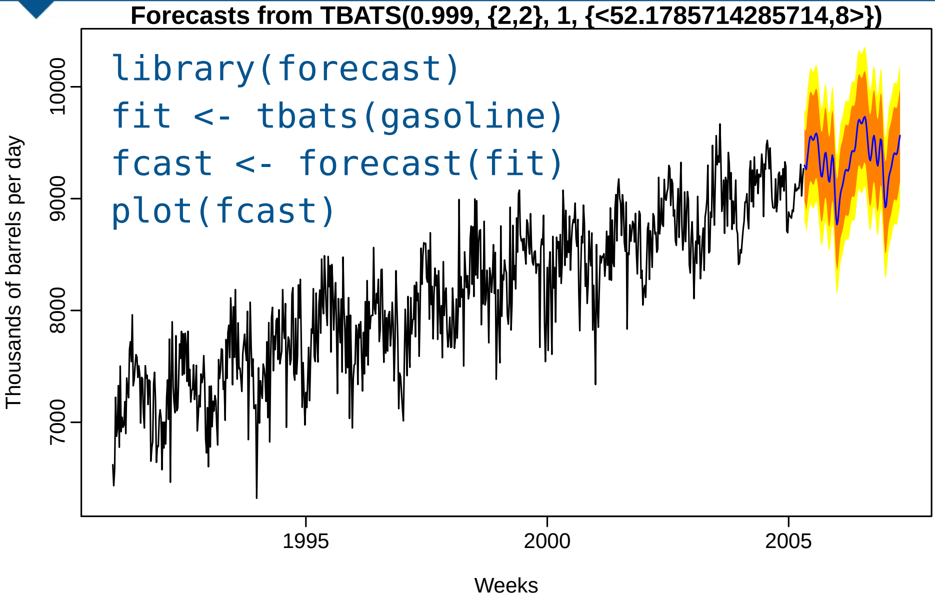
M seasonal periods

global and local trend

ARMA error

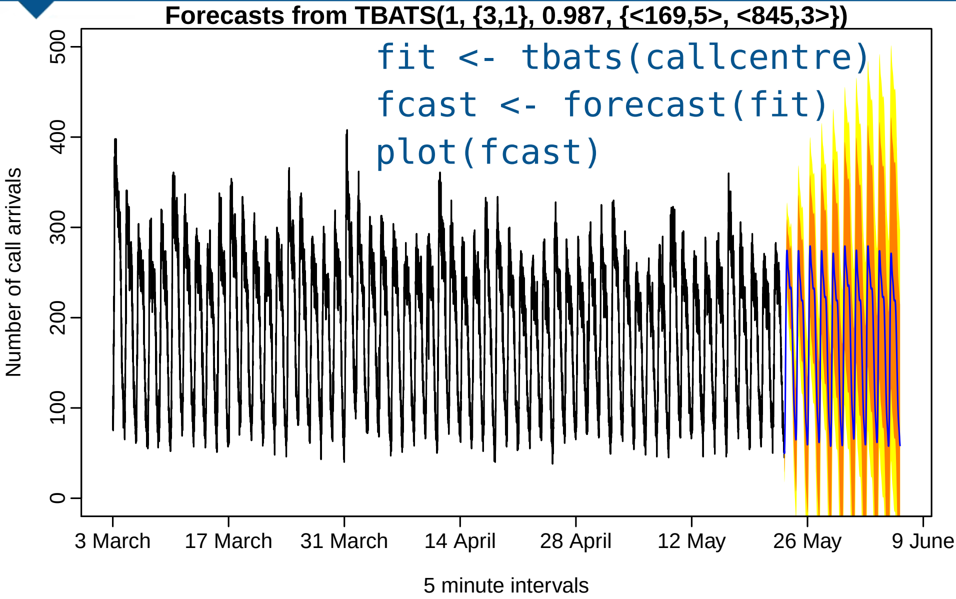
Fourier-like seasonal terms

Examples

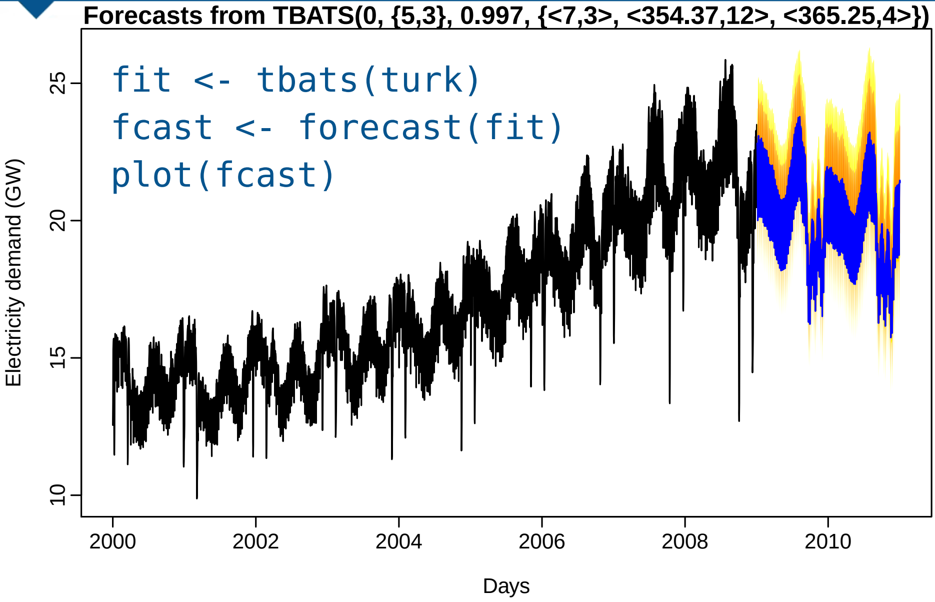



```r
library(forecast)
fit <- tbats(gasoline)
fcast <- forecast(fit)
plot(fcast)
```

Examples


Forecasts from TBATS(1, {3,1}, 0.987, {<169,5>, <845,3>})
fit <- tbats(callcentre)
fcast <- forecast(fit)
plot(fcast)
Number of call arrivals
0
100
200
300
400
500
3 March
17 March
31 March
14 April
28 April
12 May
26 May
9 June
5 minute intervals

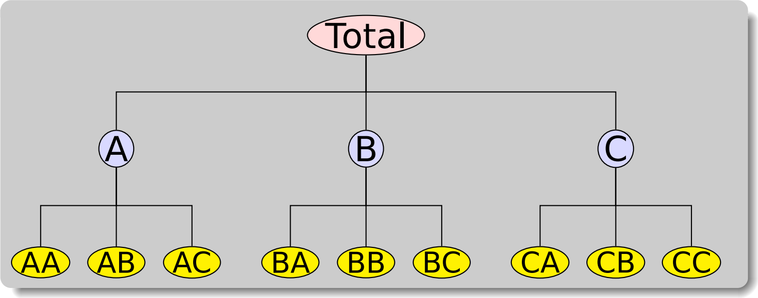
Examples




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

Outline

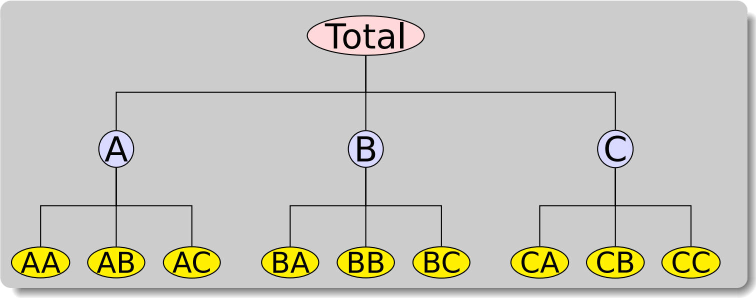
Introduction



Examples

- Manufacturing product hierarchies
- Pharmaceutical sales
- Net labour turnover

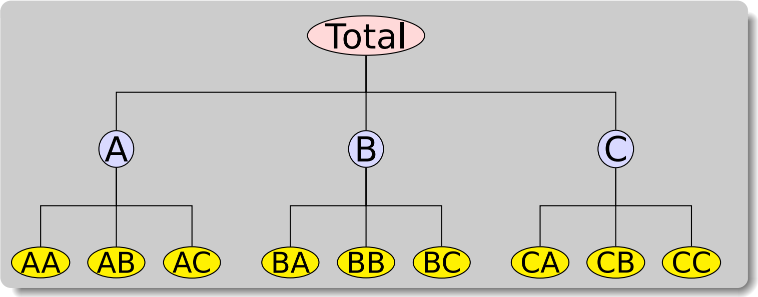
Introduction



Examples

- Manufacturing product hierarchies
- Pharmaceutical sales
- Net labour turnover

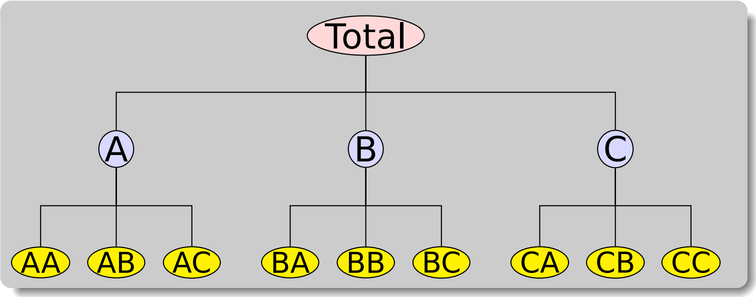
Introduction



Examples

- Manufacturing product hierarchies
- Pharmaceutical sales
- Net labour turnover

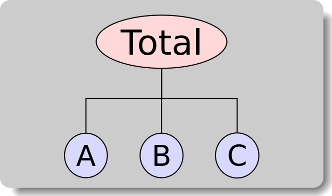
Introduction



Examples

- Manufacturing product hierarchies
- Pharmaceutical sales
- Net labour turnover

Notation



K : number of levels in hierarchy (excl. Total).

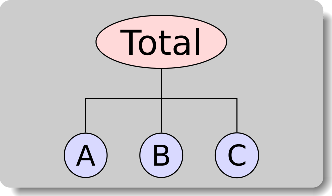
Y_t : observed aggregate of all series at time t.

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

$\boldsymbol{Y}_{i,t}$: vector of all series at level i in time t.

$\boldsymbol{Y}_t = [Y_t, \boldsymbol{Y}_{1,t}, \dots, \boldsymbol{Y}_{K,t}]'$

Notation



K : number of levels in hierarchy (excl. Total).

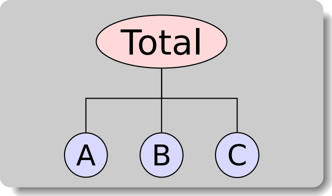
Y_t : observed aggregate of all series at time t.

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

$\boldsymbol{Y}_{i,t}$: vector of all series at level i in time t.

$\boldsymbol{Y}_t = [Y_t, \boldsymbol{Y}_{1,t}, \dots, \boldsymbol{Y}_{K,t}]'$

Notation



K : number of levels in hierarchy (excl. Total).

Y_t : observed aggregate of all series at time t.

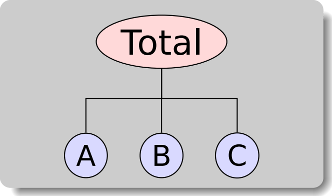
$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{Y}_{i,t}$: vector of all series at level i in time t.

$\boldsymbol{Y}_t = [Y_t, \boldsymbol{Y}_{1,t}, \dots, \boldsymbol{Y}_{K,t}]'$

$$\boldsymbol{Y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix}$$

Notation



K : number of levels in hierarchy (excl. Total).

Y_t : observed aggregate of all series at time t.

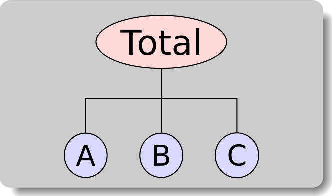
$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{Y}_{i,t}$: vector of all series at level i in time t.

$\boldsymbol{Y}_t = [Y_t, \boldsymbol{Y}_{1,t}, \ldots, \boldsymbol{Y}_{K,t}]'$

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

Notation



K : number of levels in hierarchy (excl. Total).

Y_t : observed aggregate of all series at time t.

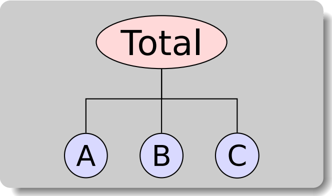
$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{Y}_{i,t}$: vector of all series at level i in time t.

$\boldsymbol{Y}_t = [Y_t, \boldsymbol{Y}_{1,t}, \dots, \boldsymbol{Y}_{K,t}]'$

$$\boldsymbol{Y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{S} \underbrace{\begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix}}_{\boldsymbol{Y}_{K,t}}$$

Notation



K : number of levels in hierarchy (excl. Total).

Y_t : observed aggregate of all series at time t.

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

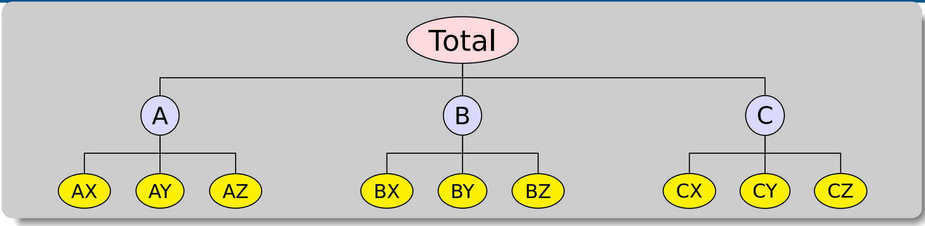
$\boldsymbol{Y}_{i,t}$: vector of all series at level i in time t.

$\boldsymbol{Y}_t = [Y_t, \boldsymbol{Y}_{1,t}, \dots, \boldsymbol{Y}_{K,t}]'$

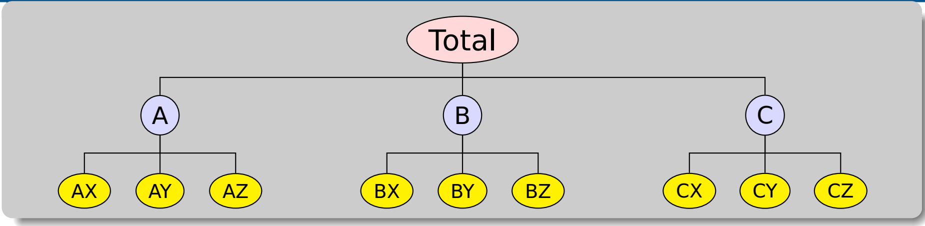
$$\boldsymbol{Y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{S} \underbrace{\begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix}}_{\boldsymbol{Y}_{K,t}}$$

$$\boxed{\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}}$$

Hierarchical data



$$\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{S} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix}}_{\boldsymbol{Y}_{K,t}}$$



$$
\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{S} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix}}_{\boldsymbol{Y}_{K,t}}
$$

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial forecasts for horizon h, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial forecasts for horizon h, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = E[\boldsymbol{Y}_{K,n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial forecasts for horizon h, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boxed{\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}}$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = E[\boldsymbol{Y}_{K,n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- ε_h has zero mean and covariance matrix Σ_h.
- Estimate $\beta_n(h)$ using GLS?
- Revised forecasts: $\tilde{\boldsymbol{Y}}_n(h) = S\hat{\beta}_n(h)$

Forecasts

Key idea: forecast reconciliation

- ➥ Ignore structural constraints and forecast every series of interest independently.
- ➥ Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial forecasts for horizon h, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}$.   So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = E[\boldsymbol{Y}_{K,n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- ε_h has zero mean and covariance matrix Σ_h.
- Estimate $\beta_n(h)$ using GLS?
- Revised forecasts: $\tilde{\boldsymbol{Y}}_n(h) = S\hat{\beta}_n(h)$

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial forecasts for horizon h, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = E[\boldsymbol{Y}_{K,n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- ε_h has zero mean and covariance matrix Σ_h.
- Estimate $\beta_n(h)$ using GLS?
- Revised forecasts: $\tilde{\boldsymbol{Y}}_n(h) = S\hat{\beta}_n(h)$

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial forecasts for horizon h, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = E[\boldsymbol{Y}_{K,n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- ε_h has zero mean and covariance matrix Σ_h.
- Estimate $\beta_n(h)$ using GLS?
- Revised forecasts: $\tilde{\boldsymbol{Y}}_n(h) = S\hat{\beta}_n(h)$

Forecasts

Key idea: forecast reconciliation

- ➥ Ignore structural constraints and forecast every series of interest independently.
- ➥ Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial forecasts for horizon h, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = E[\boldsymbol{Y}_{K,n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- ε_h has zero mean and covariance matrix Σ_h.
- Estimate $\beta_n(h)$ using GLS?
- Revised forecasts: $\tilde{\boldsymbol{Y}}_n(h) = S\hat{\beta}_n(h)$

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial forecasts for horizon h, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{Y}_{K,t}$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = E[\boldsymbol{Y}_{K,n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- ε_h has zero mean and covariance matrix Σ_h.
- Estimate $\beta_n(h)$ using GLS?
- Revised forecasts: $\tilde{\boldsymbol{Y}}_n(h) = S\hat{\beta}_n(h)$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^{\dagger}S)^{-1}S'\Sigma_h^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

- $\Sigma_h^{\dagger}$ is generalized inverse of Σ_h.

$$\tilde{\boldsymbol{Y}}_n(h) = S(S'S)^{-1}S'\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^\dagger S)^{-1}S'\Sigma_h^\dagger \hat{\mathbf{Y}}_n(h)$$

Base forecasts

- $\Sigma_h^\dagger$ is generalized inverse of Σ_h.
- Problem: Don't know Σ_h and hard to estimate.
- Solution: Assume $\varepsilon_h \approx S\varepsilon_{K,h}$ where $\varepsilon_{K,h}$ is the forecast error at bottom level.
  Then $\Sigma_h \approx S\Omega_h S'$ where $\Omega_h = \mathrm{Var}(\varepsilon_{K,h})$.
  If Moore-Penrose generalized inverse used, then

$$(S'\Sigma^\dagger S)^{-1}S'\Sigma^\dagger = (S'S)^{-1}S'.$$

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^\dagger S)^{-1}S'\Sigma_h^\dagger\hat{\boldsymbol{Y}}_n(h)$$

Revised forecasts

Base forecasts

- $\Sigma_h^\dagger$ is generalized inverse of Σ_h.
- **Problem:** Don't know Σ_h and hard to estimate.
- **Solution:** Assume $e_h \approx S e_{K,h}$ where $e_{K,h}$ is the forecast error at bottom level.
  Then $\Sigma_h \approx S\Omega_h S'$ where $\Omega_h = \text{Var}(e_{K,h})$.
  If Moore-Penrose generalized inverse used, then

$$(S'\Sigma^\dagger S)^{-1}S'\Sigma^\dagger = (S'S)^{-1}S'.$$

$$\tilde{\boldsymbol{Y}}_n(h) = S(S'S)^{-1}S'\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^\dagger S)^{-1}S'\Sigma_h^\dagger \hat{\mathbf{Y}}_n(h)$$

Revised forecasts   Base forecasts

- $\Sigma_h^\dagger$ is generalized inverse of Σ_h.
- **Problem:** Don't know Σ_h and hard to estimate.
- **Solution:** Assume $\varepsilon_h \approx S\varepsilon_{K,h}$ where $\varepsilon_{K,h}$ is the forecast error at bottom level.
  Then $\Sigma_h \approx S\Omega_h S'$ where $\Omega_h = \text{Var}(\varepsilon_{K,h})$.
  If Moore-Penrose generalized inverse used, then

$$(S'\Sigma^\dagger S)^{-1}S'\Sigma^\dagger = (S'S)^{-1}S'.$$

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^\dagger S)^{-1}S'\Sigma_h^\dagger \hat{\mathbf{Y}}_n(h)$$

Revised forecasts   Base forecasts

- $\Sigma_h^\dagger$ is generalized inverse of Σ_h.
- **Problem:** Don't know Σ_h and hard to estimate.
- **Solution:** Assume $\varepsilon_h \approx S\varepsilon_{K,h}$ where $\varepsilon_{K,h}$ is the forecast error at bottom level.
  Then $\Sigma_h \approx S\Omega_h S'$ where $\Omega_h = \mathrm{Var}(\varepsilon_{K,h})$.
  If Moore-Penrose generalized inverse used, then

$$(S'\Sigma^\dagger S)^{-1}S'\Sigma^\dagger = (S'S)^{-1}S'.$$

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^{\dagger}S)^{-1}S'\Sigma_h^{\dagger}\hat{\mathbf{Y}}_n(h)$$

Revised forecasts | Base forecasts

- $\Sigma_h^{\dagger}$ is generalized inverse of Σ_h.
- **Problem:** Don't know Σ_h and hard to estimate.
- **Solution:** Assume $\varepsilon_h \approx S\varepsilon_{K,h}$ where $\varepsilon_{K,h}$ is the forecast error at bottom level.
  Then $\Sigma_h \approx S\Omega_h S'$ where $\Omega_h = \text{Var}(\varepsilon_{K,h})$.
  If Moore-Penrose generalized inverse used, then

$$(S'\Sigma^{\dagger}S)^{-1}S'\Sigma^{\dagger} = (S'S)^{-1}S'.$$

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^{\dagger}S)^{-1}S'\Sigma_h^{\dagger}\hat{\mathbf{Y}}_n(h)$$

Revised forecasts Base forecasts

- $\Sigma_h^{\dagger}$ is generalized inverse of Σ_h.
- **Problem:** Don't know Σ_h and hard to estimate.
- **Solution:** Assume $\varepsilon_h \approx S\varepsilon_{K,h}$ where $\varepsilon_{K,h}$ is the forecast error at bottom level.
  Then $\Sigma_h \approx S\Omega_h S'$ where $\Omega_h = \mathrm{Var}(\varepsilon_{K,h})$.
  If Moore-Penrose generalized inverse used, then

$$(S'\Sigma^{\dagger}S)^{-1}S'\Sigma^{\dagger} = (S'S)^{-1}S'.$$

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^\dagger S)^{-1}S'\Sigma_h^\dagger \hat{\mathbf{Y}}_n(h)$$

Revised forecasts

Base forecasts

- $\Sigma_h^\dagger$ is generalized inverse of Σ_h.
- **Problem:** Don't know Σ_h and hard to estimate.
- **Solution:** Assume $\varepsilon_h \approx S\varepsilon_{K,h}$ where $\varepsilon_{K,h}$ is the forecast error at bottom level.
  Then $\Sigma_h \approx S\Omega_h S'$ where $\Omega_h = \text{Var}(\varepsilon_{K,h})$.
  If Moore-Penrose generalized inverse used, then

$$(S'\Sigma^\dagger S)^{-1}S'\Sigma^\dagger = (S'S)^{-1}S'.$$

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h^\dagger S)^{-1}S'\Sigma_h^\dagger \hat{\mathbf{Y}}_n(h)$$

Revised forecasts

Base forecasts

- $\Sigma_h^\dagger$ is generalized inverse of Σ_h.
- **Problem:** Don't know Σ_h and hard to estimate.
- **Solution:** Assume $\varepsilon_h \approx S\varepsilon_{K,h}$ where $\varepsilon_{K,h}$ is the forecast error at bottom level.
  Then $\Sigma_h \approx S\Omega_h S'$ where $\Omega_h = \text{Var}(\varepsilon_{K,h})$.
  If Moore-Penrose generalized inverse used, then

$$(S'\Sigma^\dagger S)^{-1}S'\Sigma^\dagger = (S'S)^{-1}S'.$$

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

GLS = OLS

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

- Optimal weighted average of base forecasts.
- Optimal weights are $S(S'S)^{-1}S'$ (independent of the data!)
- Covariates can be included in base forecasts.
- Computational difficulties in big hierarchies due to size of S matrix.

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

- Optimal weighted average of base forecasts.
- Optimal weights are $S(S'S)^{-1}S'$ (independent of the data!)
- Covariates can be included in base forecasts.
- Computational difficulties in big hierarchies due to size of S matrix.

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

- Optimal weighted average of base forecasts.
- Optimal weights are $S(S'S)^{-1}S'$ (independent of the data!)
- Covariates can be included in base forecasts.
- Computational difficulties in big hierarchies due to size of S matrix.

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

- Optimal weighted average of base forecasts.
- Optimal weights are $S(S'S)^{-1}S'$ (independent of the data!)
- Covariates can be included in base forecasts.
- Computational difficulties in big hierarchies due to size of S matrix.

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = S(S'S)^{-1}S'\hat{\mathbf{Y}}_n(h)$$

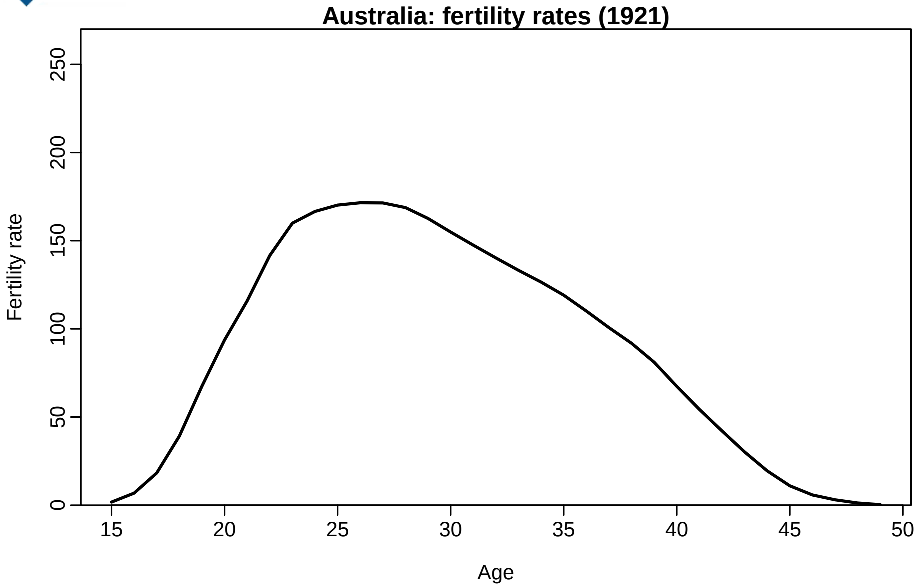
- Optimal weighted average of base forecasts.
- Optimal weights are $S(S'S)^{-1}S'$ (independent of the data!)
- Covariates can be included in base forecasts.
- Computational difficulties in big hierarchies due to size of S matrix.

Hyndman, Ahmed, Athanasopoulos, Shang (2011). “Optimal combination forecasts for hierarchical time series”. *Computational Statistics and Data Analysis* **55**(9), 2579–2589

Hyndman, Ahmed, Shang (2011). *hts: Hierarchical time series*. `cran.r-project.org/package=hts`

Outline

Fertility rates



A functional time series model

Let $y_{t,x}$ be the observed data in period t at age x, $t = 1, \dots, n$.

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- Estimate $f_t(x)$ using penalized regression splines.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- $\varepsilon_{t,x} \overset{iid}{\sim} N(0,1)$ and $e_t(x) \overset{iid}{\sim} N(0, v(x))$.

A functional time series model

Let $y_{t,x}$ be the observed data in period t at age x, $t = 1, \dots, n$.

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- Estimate $f_t(x)$ using penalized regression splines.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- $\varepsilon_{t,x} \overset{\text{iid}}{\sim} \text{N}(0,1)$ and $e_t(x) \overset{\text{iid}}{\sim} \text{N}(0, v(x))$.

A functional time series model

Let $y_{t,x}$ be the observed data in period t at age x, $t = 1, \dots, n$.

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- Estimate $f_t(x)$ using penalized regression splines.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- $\varepsilon_{t,x} \overset{\text{iid}}{\sim} \mathrm{N}(0,1)$ and $e_t(x) \overset{\text{iid}}{\sim} \mathrm{N}(0, v(x))$.

A functional time series model

Let $y_{t,x}$ be the observed data in period t at age x, $t = 1, \dots, n$.

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- Estimate $f_t(x)$ using penalized regression splines.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- $\varepsilon_{t,x} \overset{\text{iid}}{\sim} \text{N}(0,1)$ and $e_t(x) \overset{\text{iid}}{\sim} \text{N}(0, v(x))$.

A functional time series model

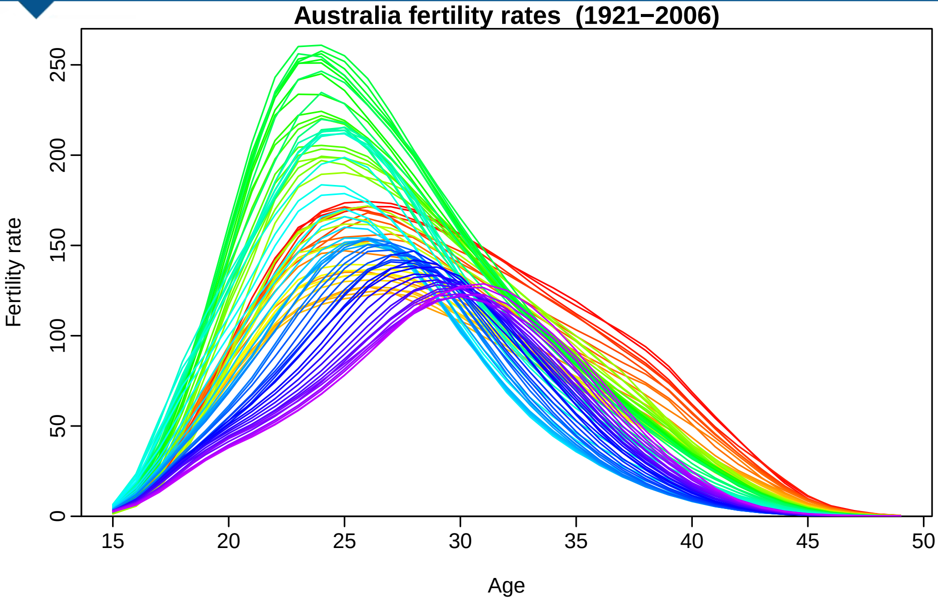
Let $y_{t,x}$ be the observed data in period t at age x, $t = 1, \dots, n$.

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

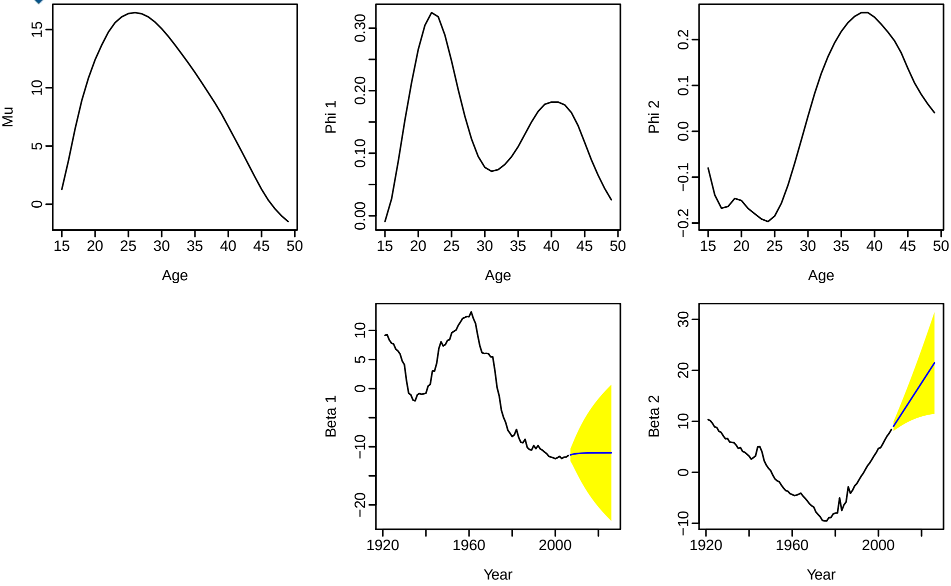
$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- Estimate $f_t(x)$ using penalized regression splines.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- $\varepsilon_{t,x} \overset{\text{iid}}{\sim} \mathrm{N}(0,1)$ and $e_t(x) \overset{\text{iid}}{\sim} \mathrm{N}(0, v(x))$.

Fertility application



Fertility model



Functional time series model

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- The eigenfunctions $\phi_k(x)$ show the main regions of variation.
- The scores $\{\beta_{t,k}\}$ are uncorrelated by construction. So we can forecast each $\beta_{t,k}$ using a univariate time series model.
- Univariate ARIMA models used for automatic forecasting of scores.

Functional time series model

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- The eigenfunctions $\phi_k(x)$ show the main regions of variation.
- The scores $\{\beta_{t,k}\}$ are uncorrelated by construction. So we can forecast each $\beta_{t,k}$ using a univariate time series model.
- Univariate ARIMA models used for automatic forecasting of scores.

Functional time series model

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- The eigenfunctions $\phi_k(x)$ show the main regions of variation.
- The scores $\{\beta_{t,k}\}$ are uncorrelated by construction. So we can forecast each $\beta_{t,k}$ using a univariate time series model.
- Univariate ARIMA models used for automatic forecasting of scores.

Functional time series model

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

- The eigenfunctions $\phi_k(x)$ show the main regions of variation.
- The scores $\{\beta_{t,k}\}$ are uncorrelated by construction. So we can forecast each $\beta_{t,k}$ using a univariate time series model.
- Univariate ARIMA models used for automatic forecasting of scores.

Forecasts

$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$

$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

Forecasts

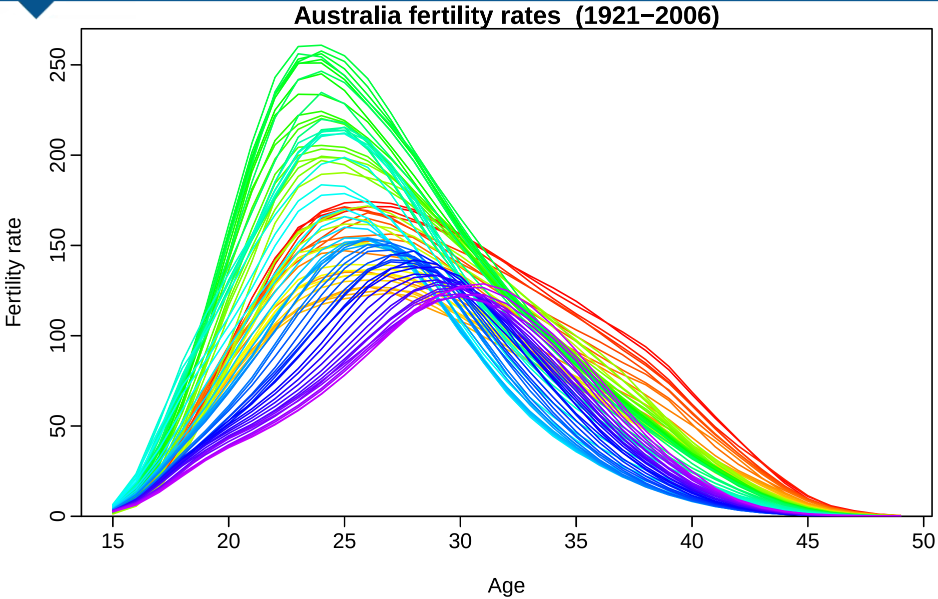
$$y_{t,x} = f_t(x) + \sigma_t(x)\varepsilon_{t,x}$$
$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\,\phi_k(x) + e_t(x)$$

$$\mathrm{E}[y_{n+h,x} \mid \boldsymbol{y}] = \hat{\mu}(x) + \sum_{k=1}^{K} \hat{\beta}_{n+h,k}\,\hat{\phi}_k(x)$$

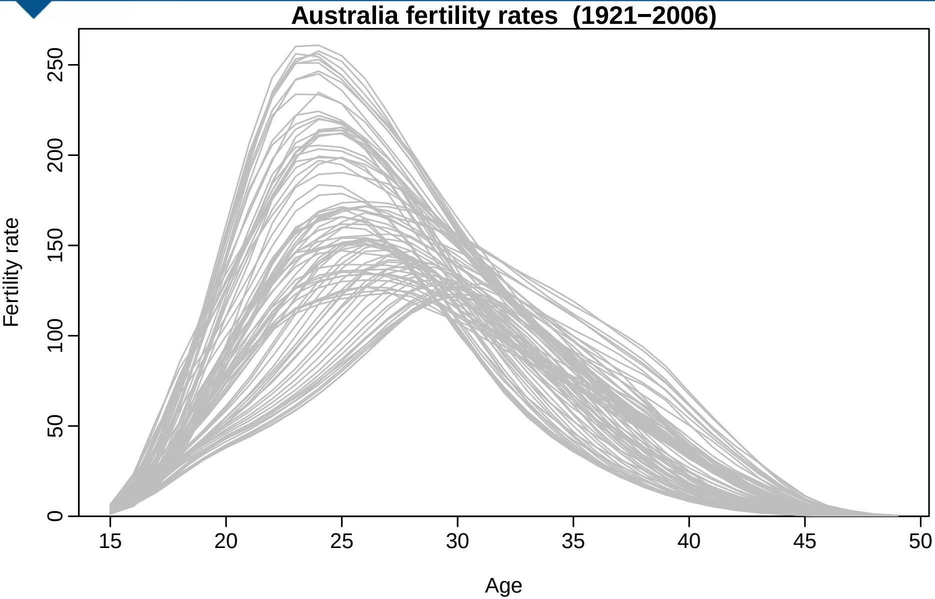
$$\mathrm{Var}[y_{n+h,x} \mid \boldsymbol{y}] = \hat{\sigma}_{\mu}^2(x) + \sum_{k=1}^{K} v_{n+h,k}\,\hat{\phi}_k^2(x) + \sigma_t^2(x) + v(x)$$

where $v_{n+h,k} = \mathrm{Var}(\beta_{n+h,k} \mid \beta_{1,k}, \dots, \beta_{n,k})$
and $\boldsymbol{y} = [y_{1,x}, \dots, y_{n,x}]$.

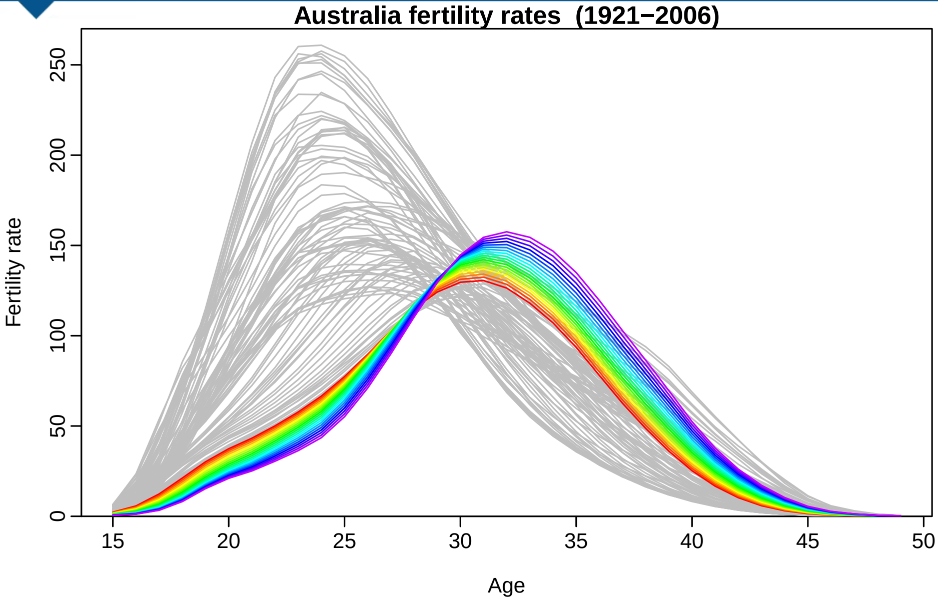
Forecasts of $f_t(x)$



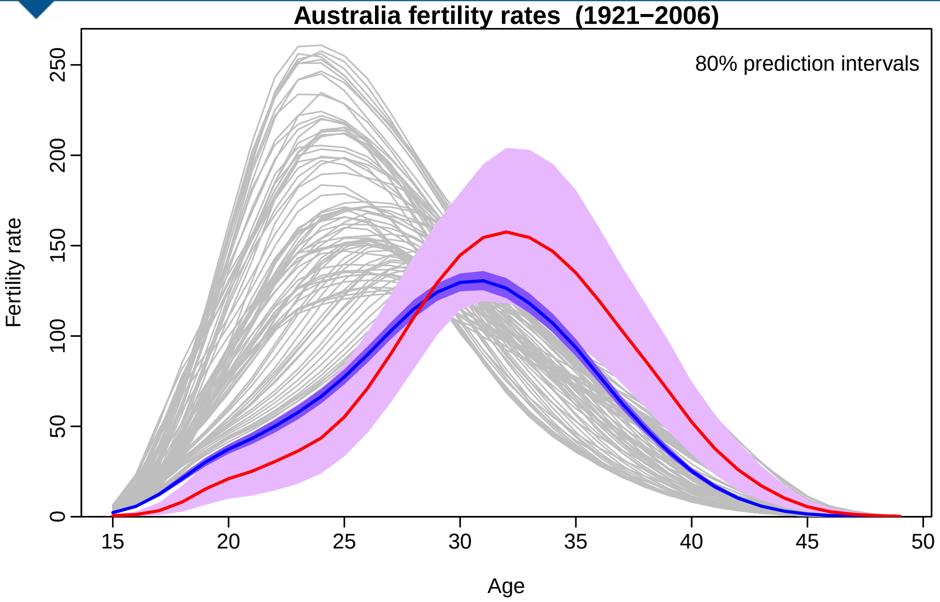
Forecasts of $f_t(x)$



Forecasts of $f_t(x)$



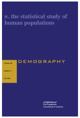
Forecasts of $f_t(x)$



References

Hyndman, Ullah (2007). Robust forecasting of mortality and fertility rates: A functional data approach. *CSDA*, **51**, 4942–4956

Hyndman, Shang (2009). Forecasting functional time series (with discussion). *JKSS* **38**(3), 199–221



Hyndman, Booth, Yasmeen (2012). Coherent mortality forecasting: the product-ratio method with functional time series models. *Demography*, to appear.

Hyndman (2012). demography: Forecasting mortality, fertility, migration and population data. `cran.r-project.org/package=demography`

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.