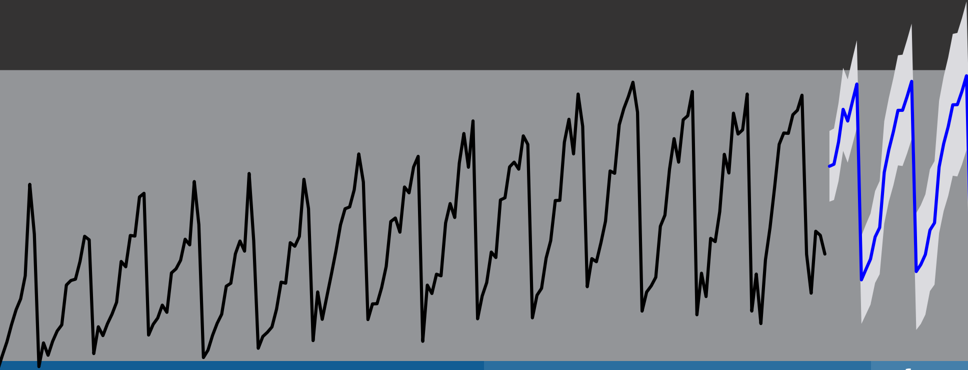
Rob J Hyndman

Forecasting without forecasters



Outline

Motivation

Australian Government

Department of Health and Ageing

Motivation

FOXTEL digital

Australian Government

Department of Health and Ageing

Motivation

FOXTEL digital

Incitec Pivot

Australian Government

Department of Health and Ageing

Motivation

FOXTEL digital

CAT

Incitec Pivot

Australian Government

Department of Health and Ageing

Motivation

FOXTEL digital

CAT

hp

Incitec Pivot

Australian Government

Department of Health and Ageing

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

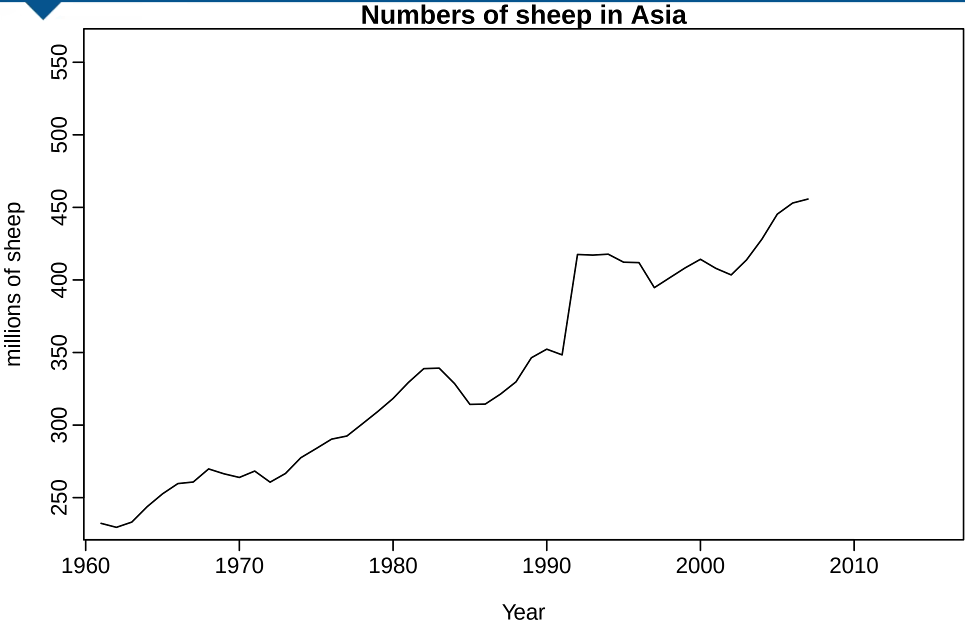
1. Common in business to have over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.
3. Some types of data can be decomposed into a large number of univariate time series that need to be forecast.

Specifications

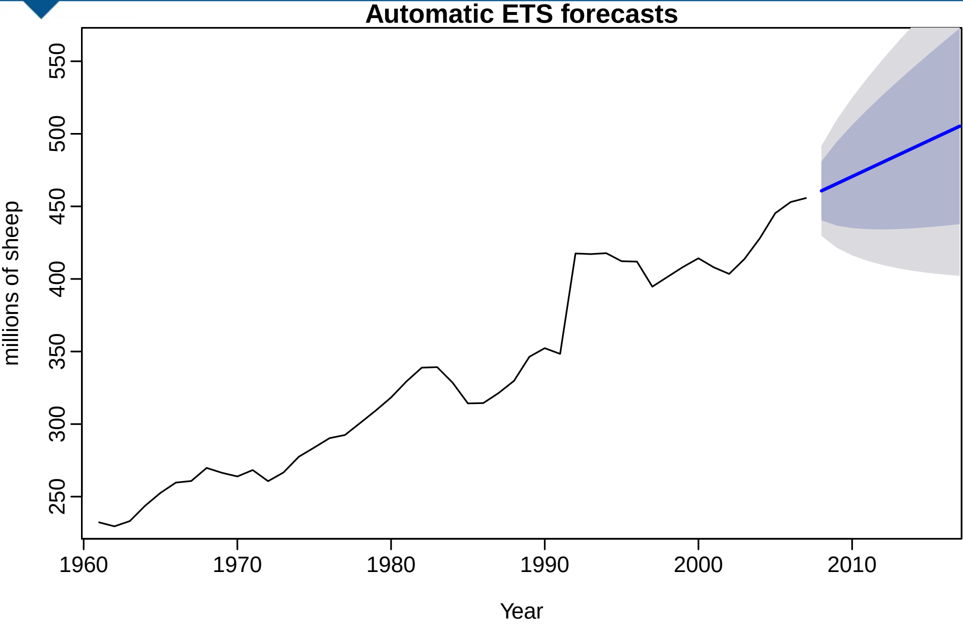
Automatic forecasting algorithms must:

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

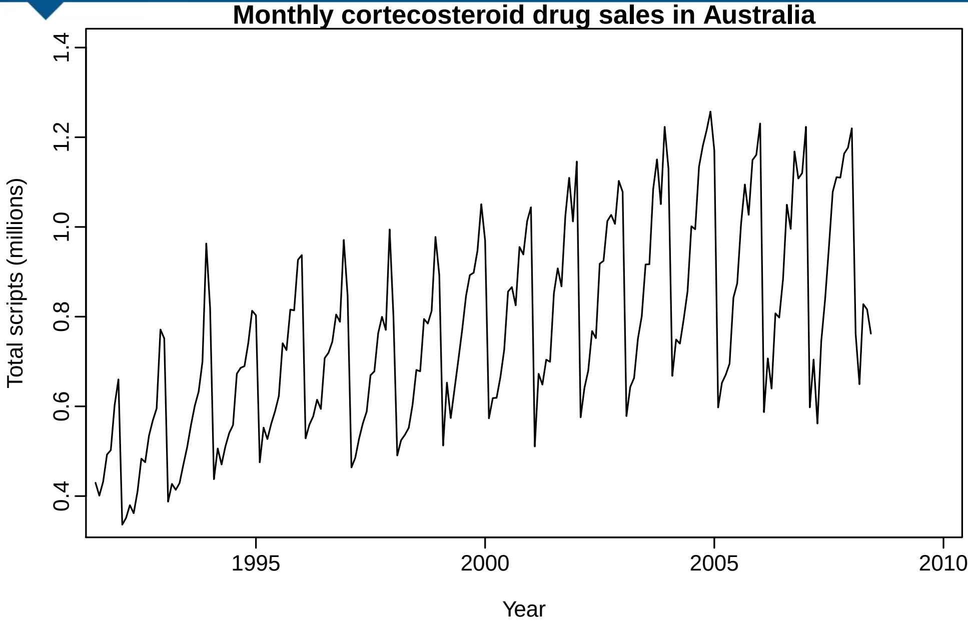
Example: Asian sheep



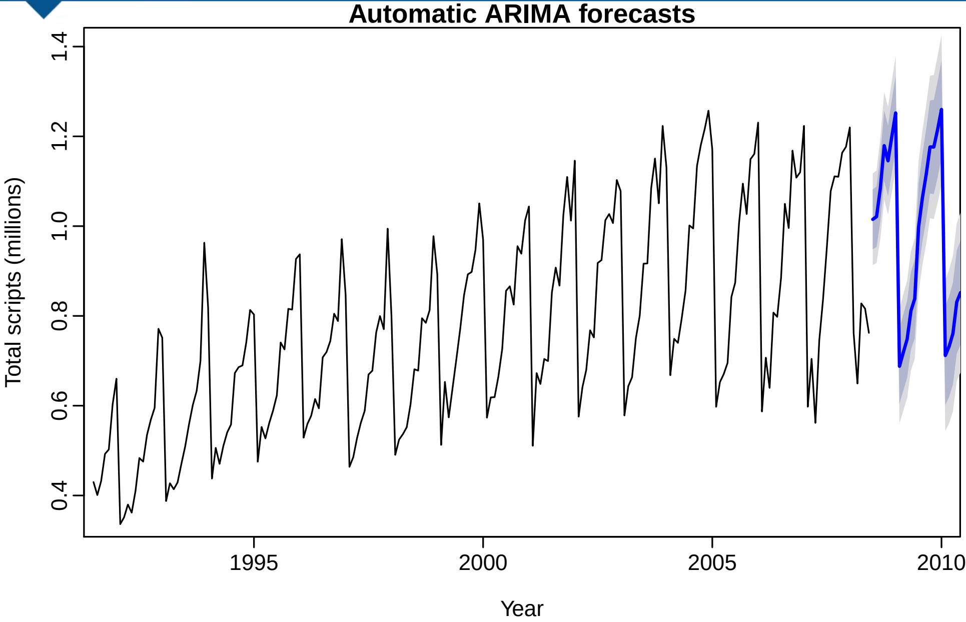
Example: Asian sheep



Example: Cortecosteroid sales



Example: Cortecosteroid sales



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

The M3-Competition: results, conclusions and implications

Spyros Makridakis, Michèle Hibon*

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- Best-performing methods undocumented.
- Limited subsequent research on general automatic forecasting algorithms.

M3 competition

International Journal of Forecasting 16 (2000) 451–476

www.elsevier.com/locate/ijforecast

The M3-Competition: results, conclusions and implications

Spyros Makridakis, Michèle Hibon*

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- Best-performing methods undocumented.
- Limited subsequent research on general automatic forecasting algorithms.

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*

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- Best-performing methods undocumented.
- Limited subsequent research on general automatic forecasting algorithms.

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

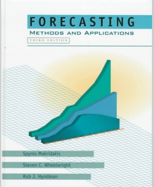
Spyros Makridakis, Michèle Hibon*

- 3003 time series.
- Early comparison of automatic forecasting algorithms.
- Best-performing methods undocumented.
- Limited subsequent research on general automatic forecasting algorithms.

Outline

Exponential smoothing

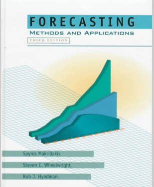
Classic Reference



Makridakis, Wheelwright and Hyndman (1998) *Forecasting: methods and applications*, 3rd ed., Wiley: NY.

Exponential smoothing

Classic Reference

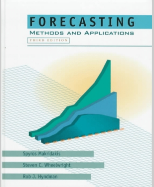


Makridakis, Wheelwright and Hyndman (1998) *Forecasting: methods and applications*, 3rd ed., Wiley: NY.

- “Unfortunately, exponential smoothing methods do not allow the easy calculation of prediction intervals.” (MWH, p.177)
- No satisfactory way to select an exponential smoothing method.

Exponential smoothing

Classic Reference

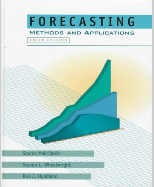


Makridakis, Wheelwright and Hyndman (1998) *Forecasting: methods and applications*, 3rd ed., Wiley: NY.

- "Unfortunately, exponential smoothing methods do not allow the easy calculation of prediction intervals." (MWH, p.177)
- No satisfactory way to select an exponential smoothing method.

Exponential smoothing

Classic Reference



Makridakis, Wheelwright and Hyndman (1998) *Forecasting: methods and applications*, 3rd ed., Wiley: NY.

Current Reference

Hyndman and Athanasopoulos (2013) *Forecasting: principles and practice*, OTexts: Australia. `OTexts.com/fpp`.

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

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

N,N: Simple exponential smoothing

Exponential smoothing methods

		Seasonal Component		
Trend 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

N,N: Simple exponential smoothing

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

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method

Exponential smoothing methods

		Seasonal Component		
Trend 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

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
M,N: Exponential 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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative 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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative damped trend method
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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

N,N: Simple exponential smoothing
A,N: Holt's linear method
A_d,N: Additive damped trend method
M,N: Exponential trend method
M_d,N: Multiplicative damped trend method
A,A: Additive Holt-Winters' method
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
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

- There are 15 separate exponential smoothing methods.

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

- There are 15 separate exponential smoothing methods.
- Each can have an additive or multiplicative error, giving 30 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
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

		Seasonal Component		
Trend 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

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

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 (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

Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each of 30 models that are appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each of 30 models that are appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each of 30 models that are appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

Automatic forecasting

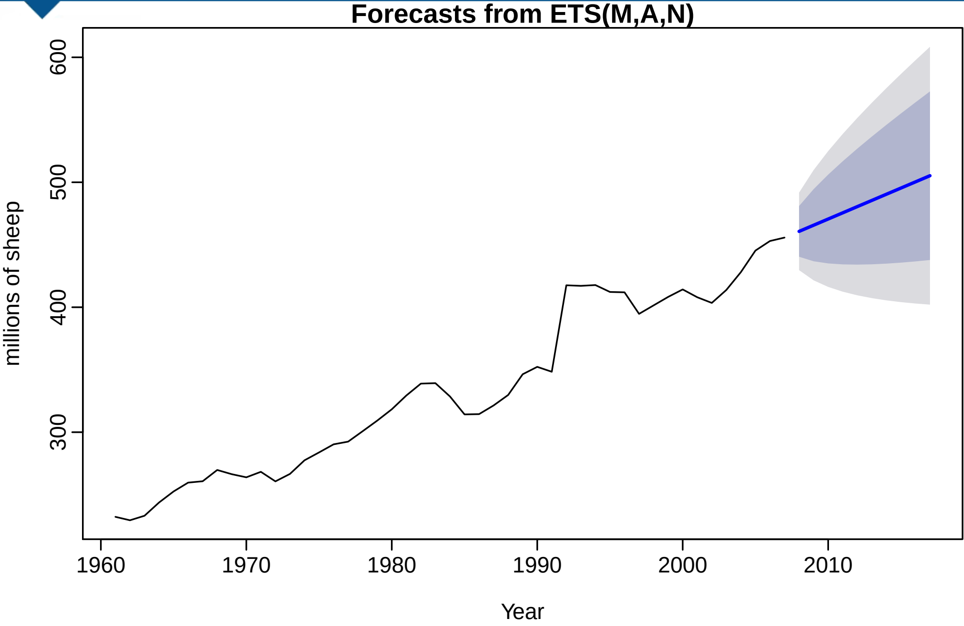
From Hyndman et al. (IJF, 2002):

- Apply each of 30 models that are appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AIC:

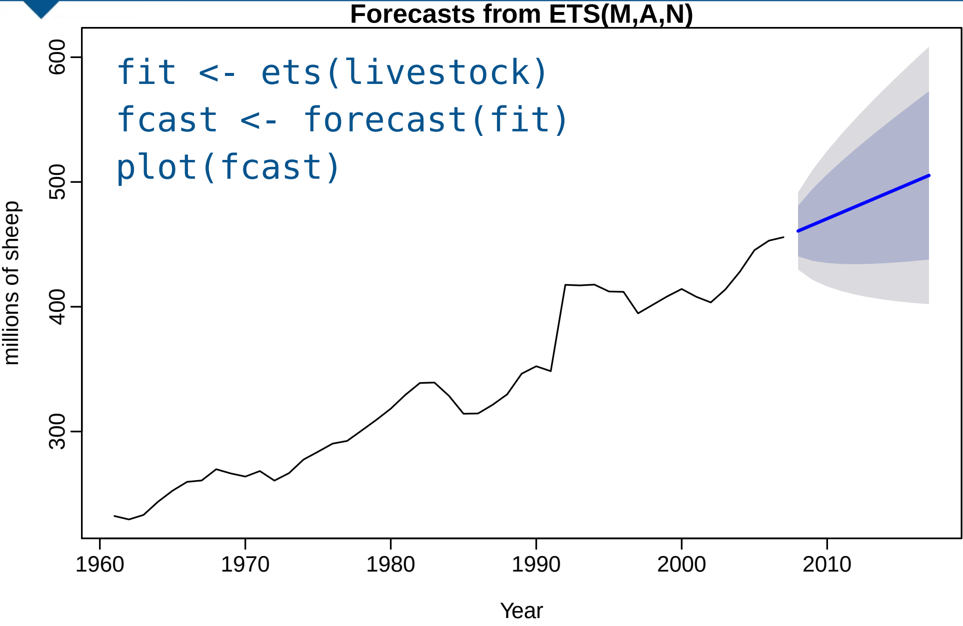
$$\text{AIC} = -2\log(\text{Likelihood}) + 2p$$

  where $p = \#$ parameters.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

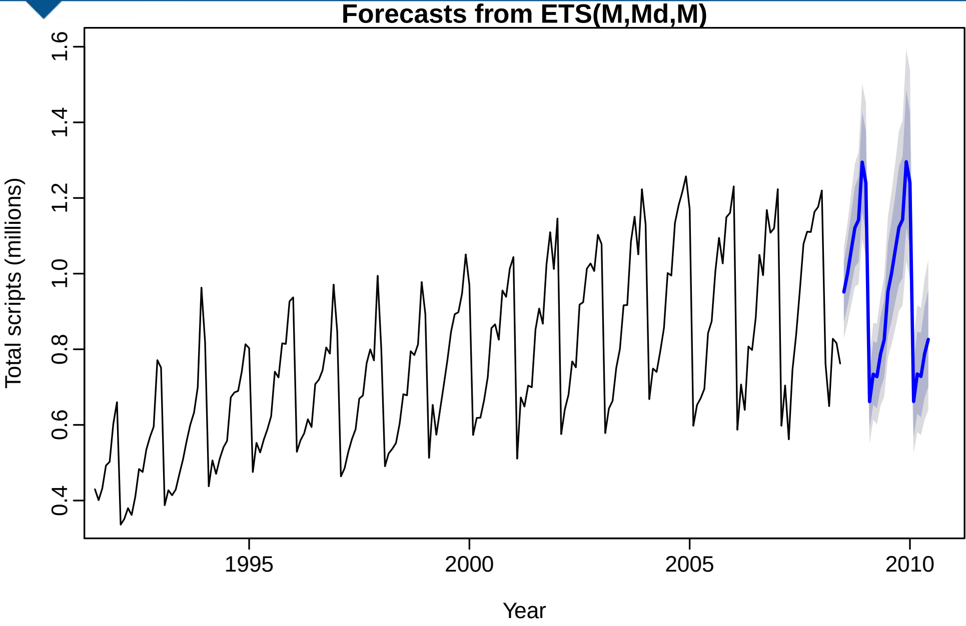
Exponential smoothing



Exponential smoothing

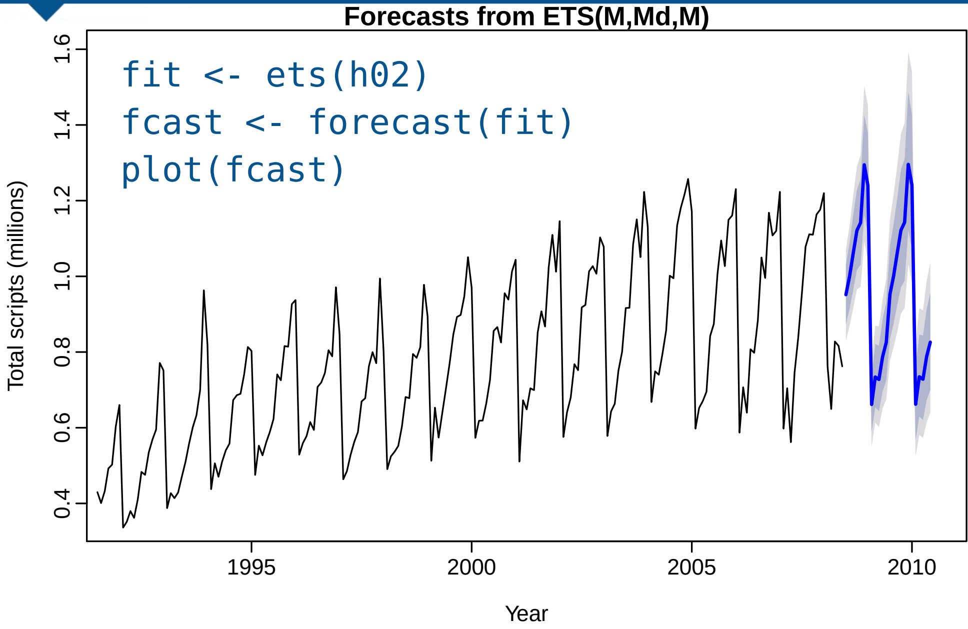


Exponential smoothing



Exponential smoothing

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

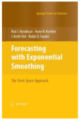


M3 comparisons

Method	MAPE	sMAPE	MASE
Theta	17.83	12.86	1.40
ForecastPro	18.00	13.06	1.47
ETS additive	18.58	13.69	1.48
ETS	19.33	13.57	1.59

References

RJ Hyndman, AB Koehler, RD Snyder, and S Grose (2002). “A state space framework for automatic forecasting using exponential smoothing methods”. *International Journal of Forecasting* **18**(3), 439–454.



RJ Hyndman, AB Koehler, JK Ord, and RD Snyder (2008). *Forecasting with exponential smoothing: the state space approach*. Springer-Verlag.

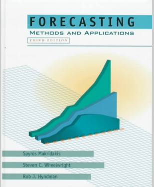
RJ Hyndman and G Athanasopoulos (2013). *Forecasting: principles and practice*. OTexts. `OTexts.com/fpp/`.

Outline

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

ARIMA modelling

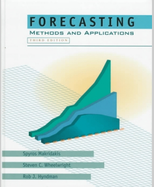
Classic Reference



Makridakis, Wheelwright and Hyndman (1998) *Forecasting: methods and applications*, 3rd ed., Wiley: NY.

ARIMA modelling

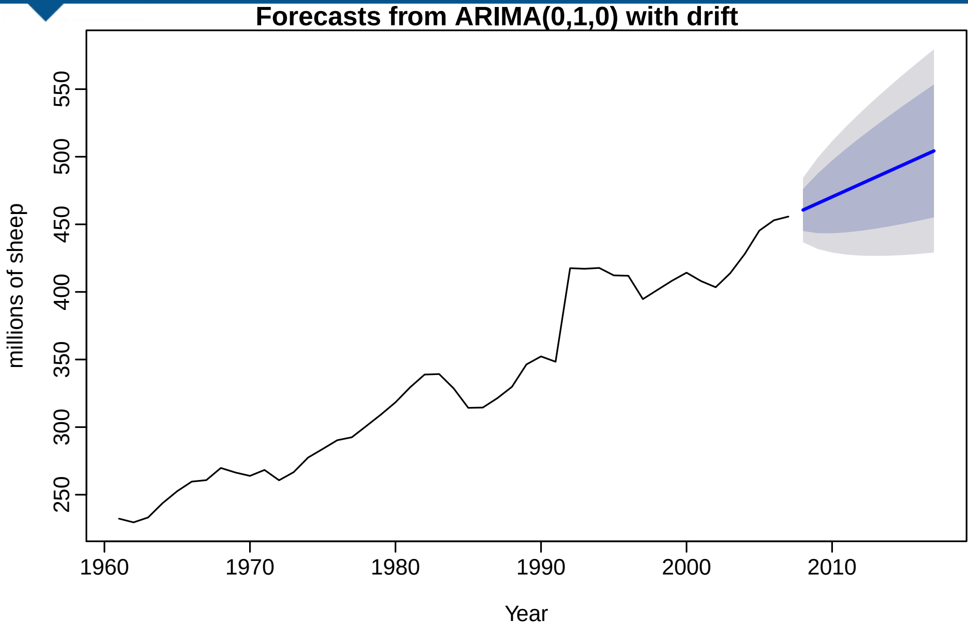
Classic Reference



Makridakis, Wheelwright and Hyndman (1998) *Forecasting: methods and applications*, 3rd ed., Wiley: NY.

- “There is such a bewildering variety of ARIMA models, it can be difficult to decide which model is most appropriate for a given set of data.” (MWH, p.347)

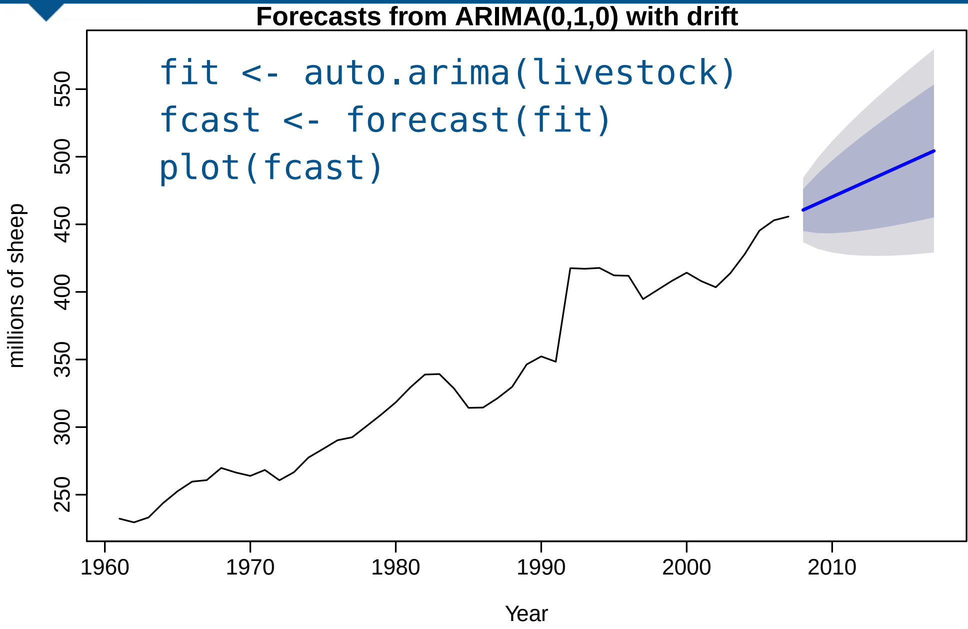
Auto ARIMA



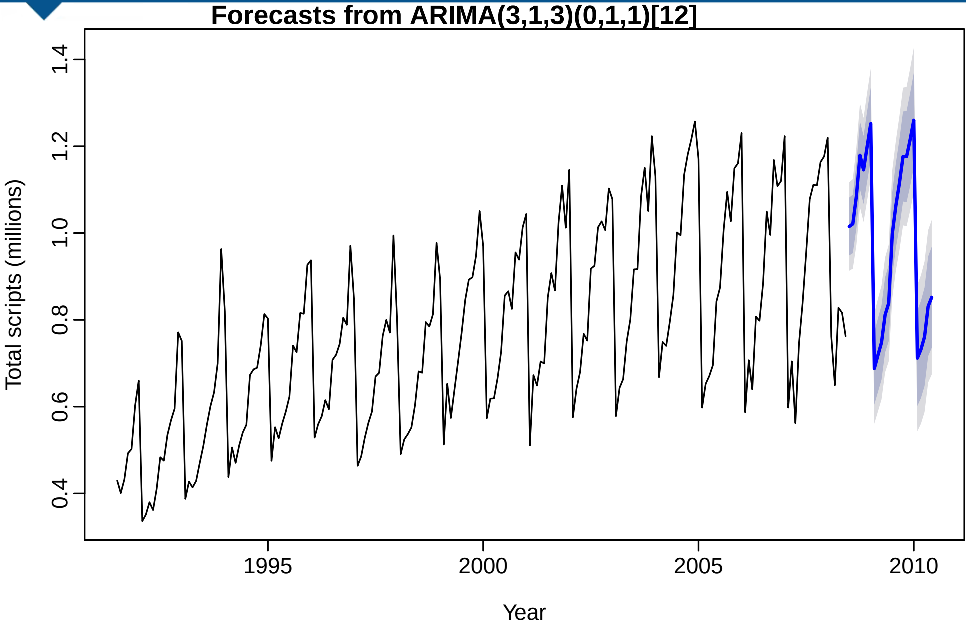
Forecasts from ARIMA(0,1,0) with drift
millions of sheep
250
300
350
400
450
500
550
1960
1970
1980
1990
2000
2010
Year

Auto ARIMA

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

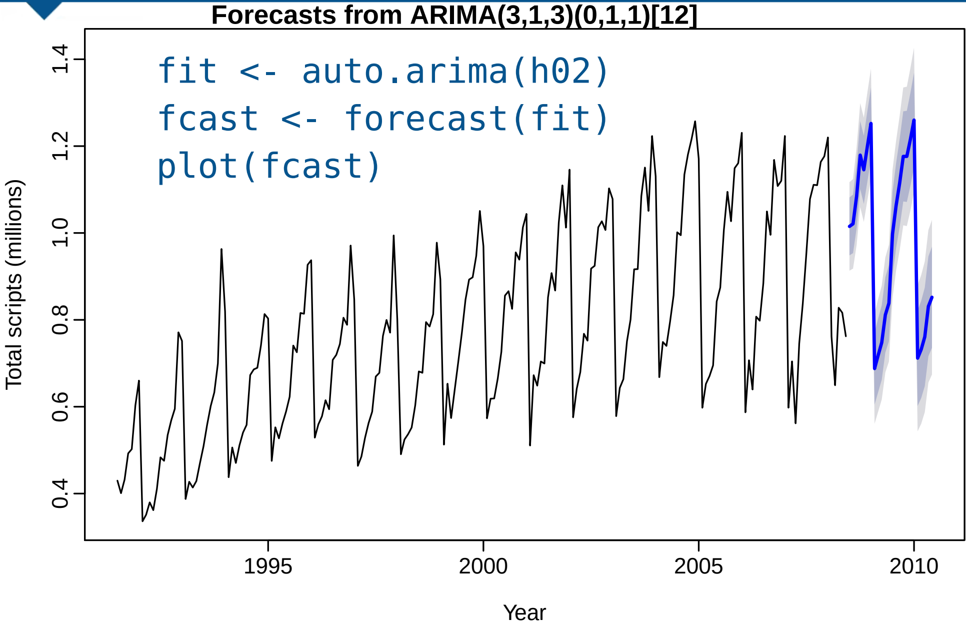


Auto ARIMA



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?

A non-seasonal ARIMA process

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

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

Algorithm choices driven by forecast accuracy.

How does auto.arima() work?

A non-seasonal ARIMA process

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

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

Hyndman & Khandakar (JSS, 2008) algorithm:

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

Algorithm choices driven by forecast accuracy.

How does auto.arima() work?

A non-seasonal ARIMA process

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

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

Hyndman & Khandakar (JSS, 2008) algorithm:

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

Algorithm choices driven by forecast accuracy.

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.

M3 comparisons

Method	MAPE	sMAPE	MASE
Theta	17.83	12.86	1.40
ForecastPro	18.00	13.06	1.47
BJauto	19.14	13.73	1.55
AutoARIMA	18.98	13.75	1.47
ETS-additive	18.58	13.69	1.48
ETS	19.33	13.57	1.59
ETS-ARIMA	18.17	13.11	1.44

M3 conclusions

MYTHS

- Simple methods do better.
- Exponential smoothing is better than ARIMA.

FACTS

- The best methods are hybrid approaches.

M3 conclusions

MYTHS

- Simple methods do better.
- Exponential smoothing is better than ARIMA.

FACTS

- The best methods are hybrid approaches.
- ETS-ARIMA (the simple average of ETS-additive and AutoARIMA) is the only fully documented method that is comparable to the M3 competition winners.
- I have an algorithm that does better than all of these, but it takes too long to be practical.

M3 conclusions

MYTHS

- Simple methods do better.
- Exponential smoothing is better than ARIMA.

FACTS

- The best methods are hybrid approaches.
- ETS-ARIMA (the simple average of ETS-additive and AutoARIMA) is the only fully documented method that is comparable to the M3 competition winners.
- I have an algorithm that does better than all of these, but it takes too long to be practical.

M3 conclusions

MYTHS

- Simple methods do better.
- Exponential smoothing is better than ARIMA.

FACTS

- The best methods are hybrid approaches.
- ETS-ARIMA (the simple average of ETS-additive and AutoARIMA) is the only fully documented method that is comparable to the M3 competition winners.
- I have an algorithm that does better than all of these, but it takes too long to be practical.

M3 conclusions

MYTHS

- Simple methods do better.
- Exponential smoothing is better than ARIMA.

FACTS

- The best methods are hybrid approaches.
- ETS-ARIMA (the simple average of ETS-additive and AutoARIMA) is the only fully documented method that is comparable to the M3 competition winners.
- I have an algorithm that does better than all of these, but it takes too long to be practical.

M3 conclusions

MYTHS

- Simple methods do better.
- Exponential smoothing is better than ARIMA.

FACTS

- The best methods are hybrid approaches.
- ETS-ARIMA (the simple average of ETS-additive and AutoARIMA) is the only fully documented method that is comparable to the M3 competition winners.
- I have an algorithm that does better than all of these, but it takes too long to be practical.

References

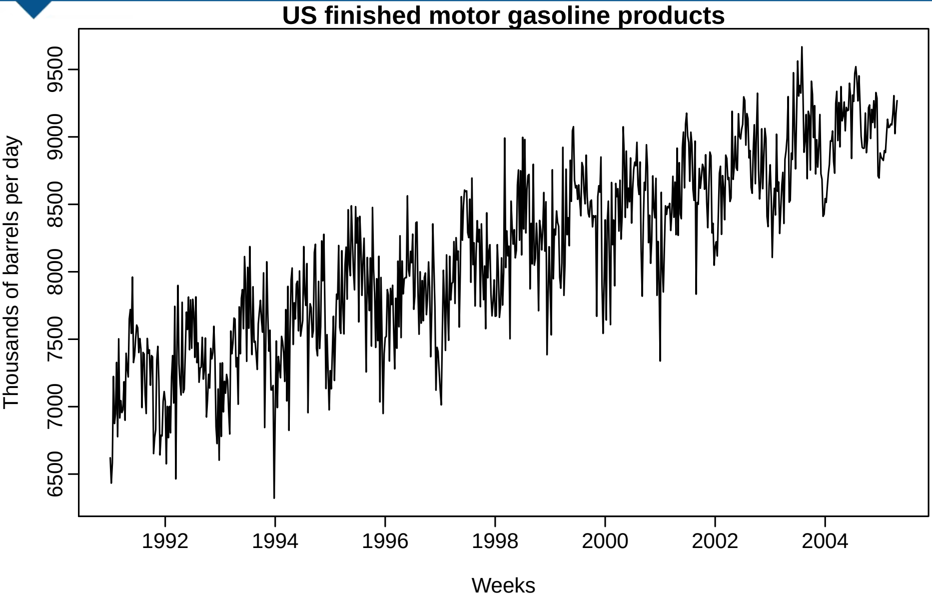
RJ Hyndman and Y Khandakar (2008). "Automatic time series forecasting : the forecast package for R". *Journal of Statistical Software* **26**(3)

RJ Hyndman (2011). "Major changes to the forecast package". `robjhyndman.com/hyndsight/forecast3/`.

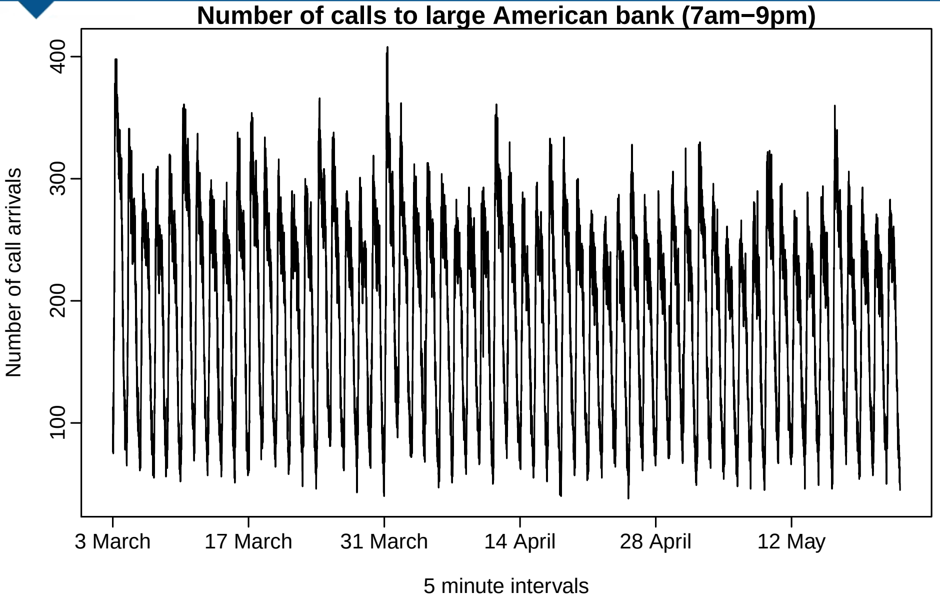
RJ Hyndman and G Athanasopoulos (2013). *Forecasting: principles and practice*. OTexts. `OTexts.com/fpp/`.

Outline

Examples

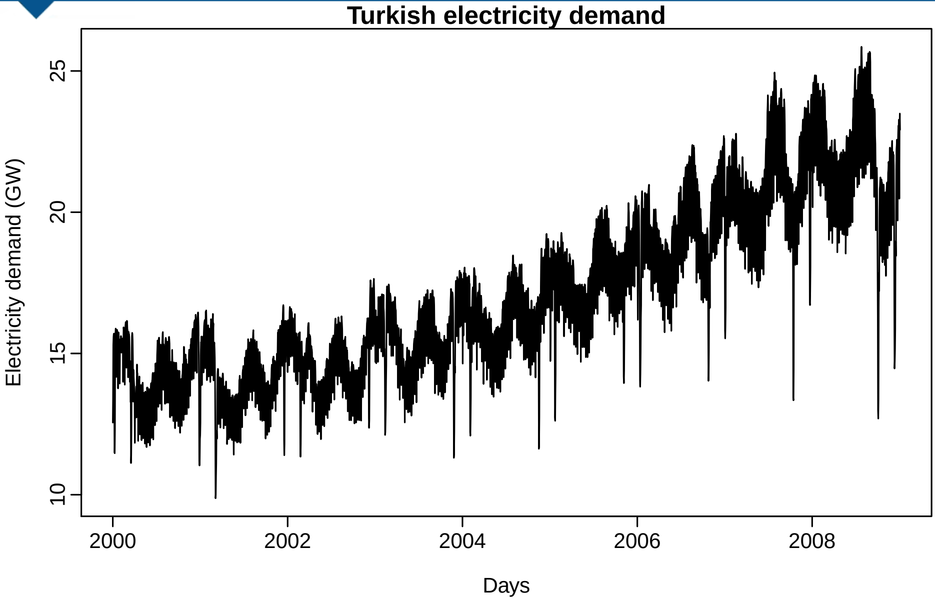

US finished motor gasoline products
Thousands of barrels per day
6500
7000
7500
8000
8500
9000
9500
1992
1994
1996
1998
2000
2002
2004
Weeks

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)
25
20
15
10
2000
2002
2004
2006
2008
Days

TBATS model

TBATS

Trigonometric terms for seasonality

Box-Cox transformations for heterogeneity

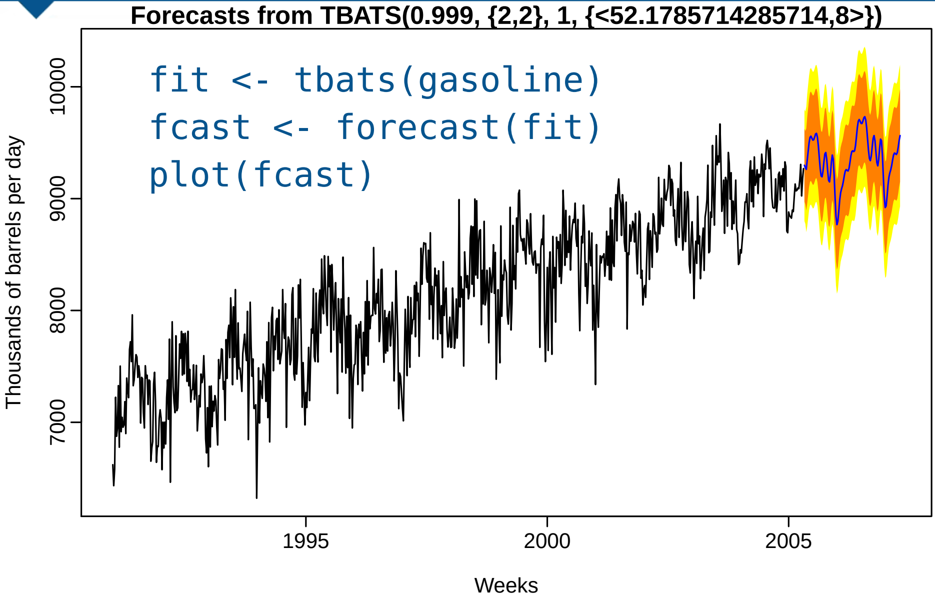
ARMA errors for short-term dynamics

Trend (possibly damped)

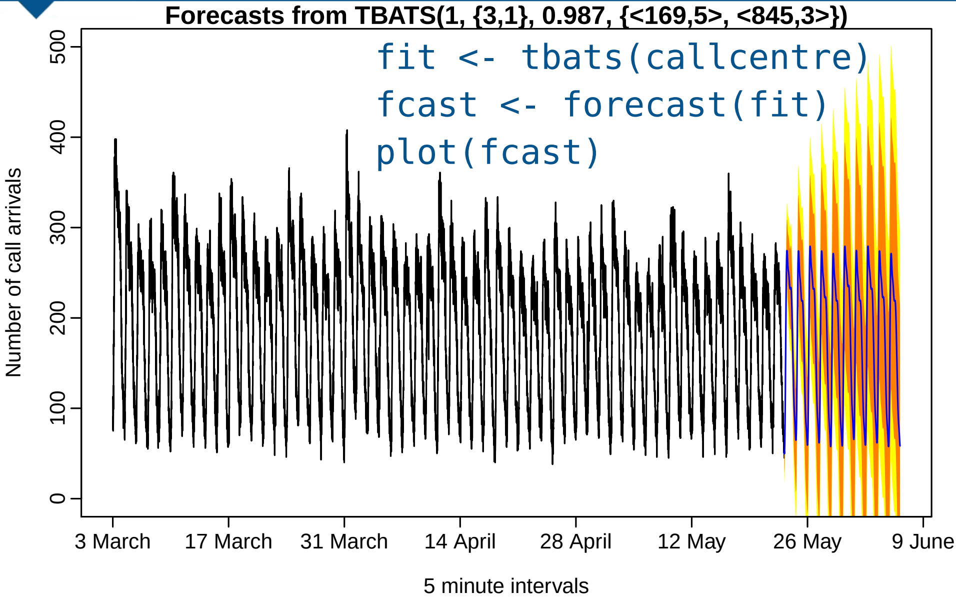
Seasonal (including multiple and non-integer periods)

Examples

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



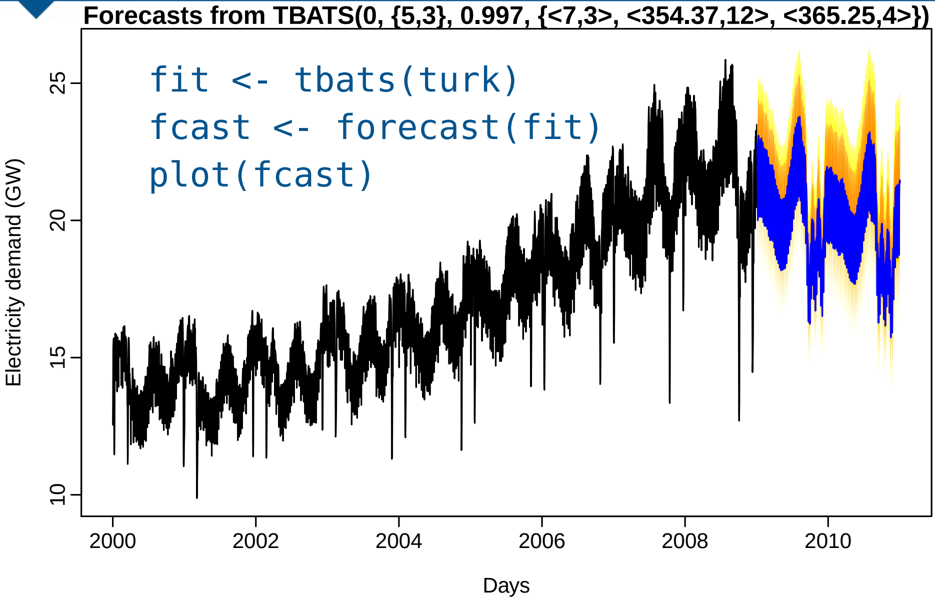
Examples



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)
```



References

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.

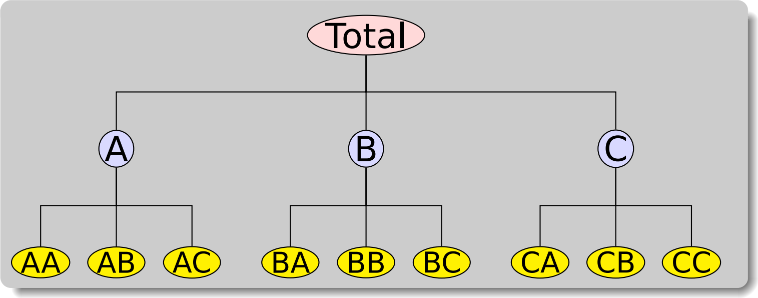
Slightly improved algorithm implemented in
RJ Hyndman (2012). *forecast: Forecasting functions for time series*.
`cran.r-project.org/package=forecast`.

More work required!

Outline

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

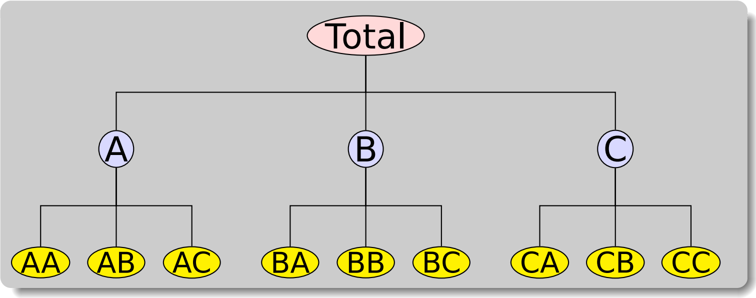
Introduction



Examples

- Manufacturing product hierarchies
- Pharmaceutical sales
- Net labour turnover

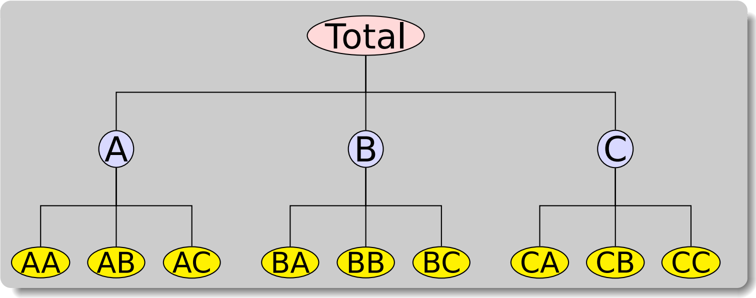
Introduction



Examples

- Manufacturing product hierarchies
- Pharmaceutical sales
- Net labour turnover

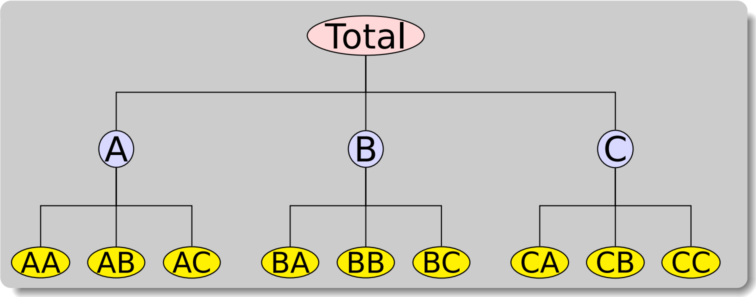
Introduction



Examples

- Manufacturing product hierarchies
- Pharmaceutical sales
- Net labour turnover

Introduction



Examples

- Manufacturing product hierarchies
- Pharmaceutical sales
- Net labour turnover

Hierarchical/grouped time series

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

Example: Pharmaceutical products are organized in a hierarchy under the Anatomical Therapeutic Chemical (ATC) Classification System.

- A **grouped time series** is a collection of time series that are aggregated in a number of non-hierarchical ways.

Example: daily numbers of calls to HP call centres are grouped by product type and location of call centre.

Hierarchical/grouped time series

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

Example: Pharmaceutical products are organized in a hierarchy under the Anatomical Therapeutic Chemical (ATC) Classification System.

- A **grouped time series** is a collection of time series that are aggregated in a number of non-hierarchical ways.

Example: daily numbers of calls to HP call centres are grouped by product type and location of call centre.

Hierarchical/grouped time series

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

Example: Pharmaceutical products are organized in a hierarchy under the Anatomical Therapeutic Chemical (ATC) Classification System.

- A **grouped time series** is a collection of time series that are aggregated in a number of non-hierarchical ways.

Example: daily numbers of calls to HP call centres are grouped by product type and location of call centre.

Hierarchical/grouped time series

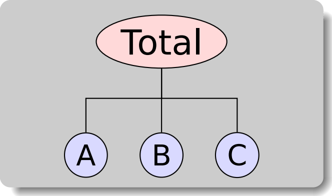
- A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

Example: Pharmaceutical products are organized in a hierarchy under the Anatomical Therapeutic Chemical (ATC) Classification System.

- A **grouped time series** is a collection of time series that are aggregated in a number of non-hierarchical ways.

Example: daily numbers of calls to HP call centres are grouped by product type and location of call centre.

Hierarchical data

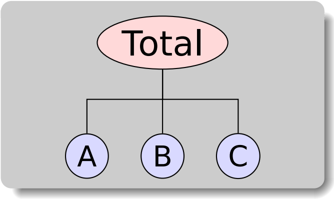


Y_t : observed aggregate of all series at time t.

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

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

Hierarchical data

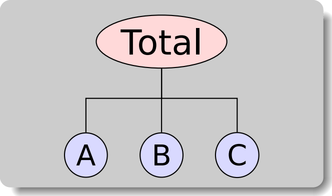


Y_t : observed aggregate of all series at time t.

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

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

Hierarchical data



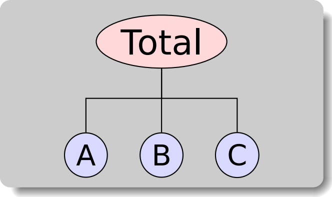
Y_t : observed aggregate of all series at time t.

$Y_{X,t}$: 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 data



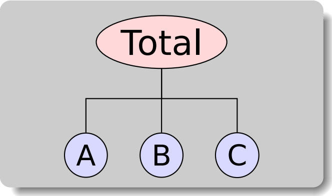
Y_t : observed aggregate of all series at time t.

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

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

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

Hierarchical data



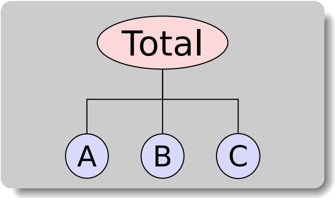
Y_t : observed aggregate of all series at time t.

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

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

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

Hierarchical data



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.

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

$$\boxed{\boldsymbol{Y}_t = S\boldsymbol{B}_t}$$

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

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

Grouped data

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

$$\boldsymbol{Y}_t = S\boldsymbol{B}_t$$

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 $\boldsymbol{Y}_t$.

Optimal reconciled forecasts:

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

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 $\boldsymbol{Y}_t$.

Optimal reconciled forecasts:

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

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 $\boldsymbol{Y}_t$.

Optimal reconciled forecasts:

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

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 $\boldsymbol{Y}_t$.

Optimal reconciled forecasts:

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

- Independent of covariance structure of hierarchy!
- Optimal reconciliation weights are $S(S'S)^{-1}S'$, independent of data.

Features

- Forget "bottom up" or "top down". This approach combines all forecasts optimally.
- Method outperforms bottom-up and top-down, especially for middle levels.
- Covariates can be included in base forecasts.
- Adjustments can be made to base forecasts at any level.
- Point forecasts are always aggregate consistent.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Forget “bottom up” or “top down”. This approach combines all forecasts optimally.
- Method outperforms bottom-up and top-down, especially for middle levels.
- Covariates can be included in base forecasts.
- Adjustments can be made to base forecasts at any level.
- Point forecasts are always aggregate consistent.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Forget “bottom up” or “top down”. This approach combines all forecasts optimally.
- Method outperforms bottom-up and top-down, especially for middle levels.
- Covariates can be included in base forecasts.
- Adjustments can be made to base forecasts at any level.
- Point forecasts are always aggregate consistent.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Forget “bottom up” or “top down”. This approach combines all forecasts optimally.
- Method outperforms bottom-up and top-down, especially for middle levels.
- Covariates can be included in base forecasts.
- Adjustments can be made to base forecasts at any level.
- Point forecasts are always aggregate consistent.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Forget "bottom up" or "top down". This approach combines all forecasts optimally.
- Method outperforms bottom-up and top-down, especially for middle levels.
- Covariates can be included in base forecasts.
- Adjustments can be made to base forecasts at any level.
- Point forecasts are always aggregate consistent.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Forget “bottom up” or “top down”. This approach combines all forecasts optimally.
- Method outperforms bottom-up and top-down, especially for middle levels.
- Covariates can be included in base forecasts.
- Adjustments can be made to base forecasts at any level.
- Point forecasts are always aggregate consistent.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Forget “bottom up” or “top down”. This approach combines all forecasts optimally.
- Method outperforms bottom-up and top-down, especially for middle levels.
- Covariates can be included in base forecasts.
- Adjustments can be made to base forecasts at any level.
- Point forecasts are always aggregate consistent.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Challenges

- Computational difficulties in big hierarchies due to size of the S matrix and non-singular behavior of $(S'S)$.
- Need to estimate covariance matrix to produce prediction intervals.

Challenges

- Computational difficulties in big hierarchies due to size of the S matrix and non-singular behavior of $(S'S)$.
- Need to estimate covariance matrix to produce prediction intervals.

Example using R

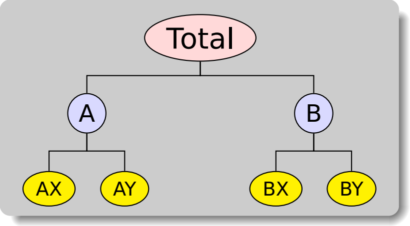
```
library(hts)

# bts is a matrix containing the bottom level time series
# g describes the grouping/hierarchical structure
y <- hts(bts, g=c(1,1,2,2))
```

Example using R

```
library(hts)

# bts is a matrix containing the bottom level time series
# g describes the grouping/hierarchical structure
y <- hts(bts, g=c(1,1,2,2))
```



Example using R

```
library(hts)

# bts is a matrix containing the bottom level time series
# g describes the grouping/hierarchical structure
y <- hts(bts, g=c(1,1,2,2))

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

Example using R

```
library(hts)

# bts is a matrix containing the bottom level time series
# g describes the grouping/hierarchical structure
y <- hts(bts, g=c(1,1,2,2))

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

# Select your own methods
ally <- allts(y)
allf <- matrix(, nrow=10, ncol=ncol(ally))
for(i in 1:ncol(ally))
  allf[,i] <- mymethod(ally[,i], h=10)
allf <- ts(allf, start=2004)
# Reconcile forecasts so they add up
fc2 <- combinef(allf, Smatrix(y))
```

References

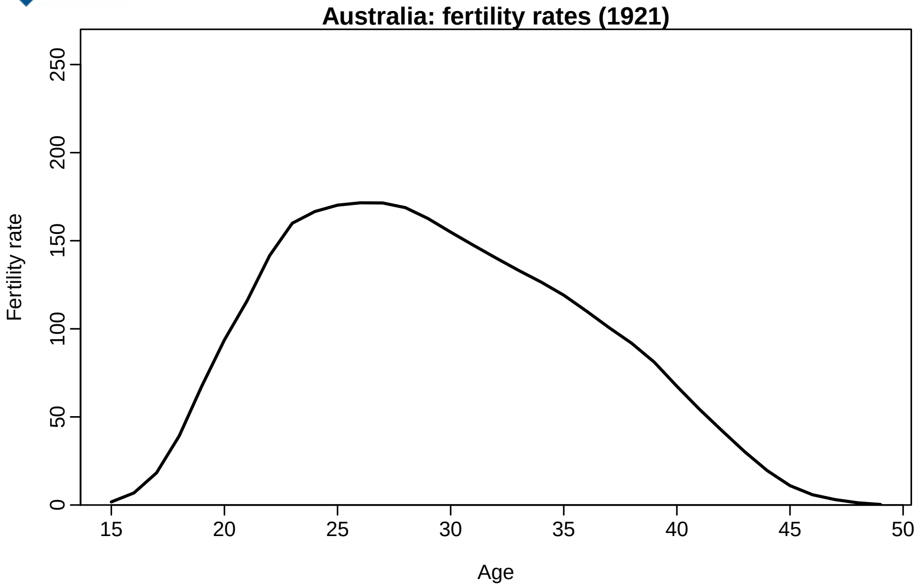
RJ Hyndman, RA Ahmed, G Athanasopoulos, and HL Shang (2011). “Optimal combination forecasts for hierarchical time series”. *Computational Statistics and Data Analysis* **55**(9), 2579–2589

RJ Hyndman, RA Ahmed, and HL Shang (2013). *hts: Hierarchical time series*. `cran.r-project.org/package=hts`.

RJ Hyndman and G Athanasopoulos (2013). *Forecasting: principles and practice*. OTexts. `OTexts.com/fpp/`.

Outline

Fertility rates



Functional data model

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

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

- Decomposition separates time and age to allow forecasting.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- Univariate models used for automatic forecasting of scores $\{\beta_{t,k}\}$.

Functional data model

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

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

- Decomposition separates time and age to allow forecasting.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- Univariate models used for automatic forecasting of scores $\{\beta_{t,k}\}$.

Functional data model

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

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

- Decomposition separates time and age to allow forecasting.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- Univariate models used for automatic forecasting of scores $\{\beta_{t,k}\}$.

Functional data model

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

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

- Decomposition separates time and age to allow forecasting.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- Univariate models used for automatic forecasting of scores $\{\beta_{t,k}\}$.

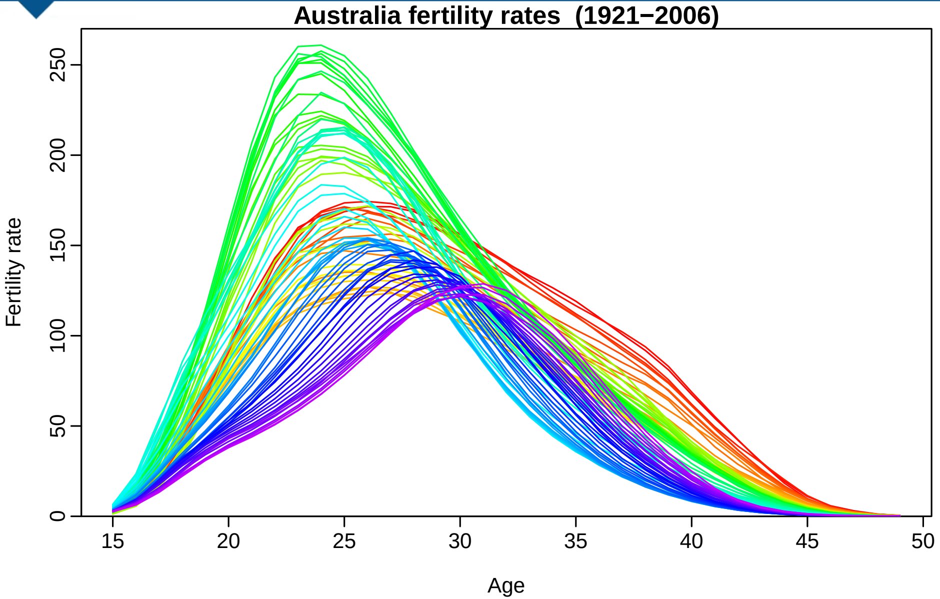
Functional data model

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

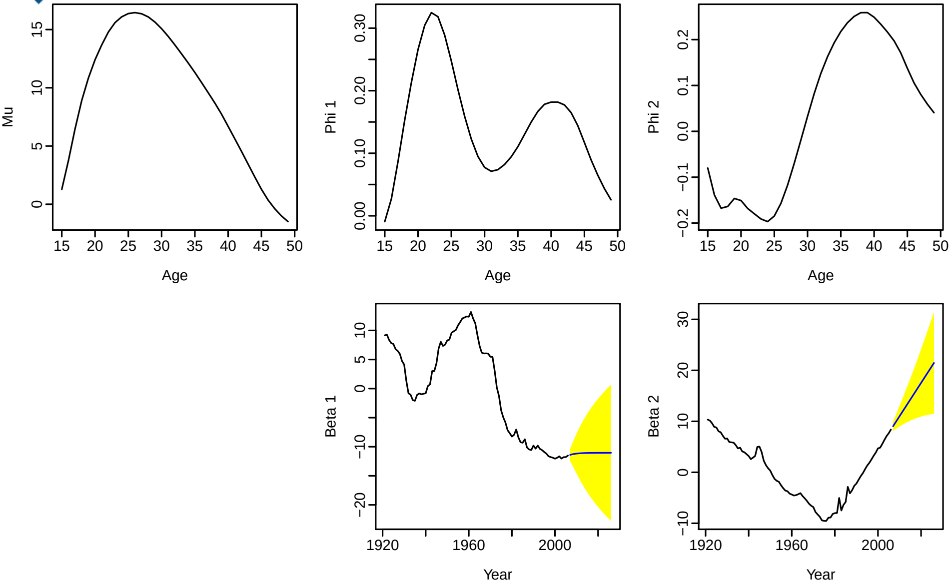
$$f_t(x) = \mu(x) + \sum_{k=1}^{K} \beta_{t,k}\, \phi_k(x) + e_t(x)$$

- Decomposition separates time and age to allow forecasting.
- Estimate $\mu(x)$ as mean $f_t(x)$ across years.
- Estimate $\beta_{t,k}$ and $\phi_k(x)$ using functional (weighted) principal components.
- Univariate models used for automatic forecasting of scores $\{\beta_{t,k}\}$.

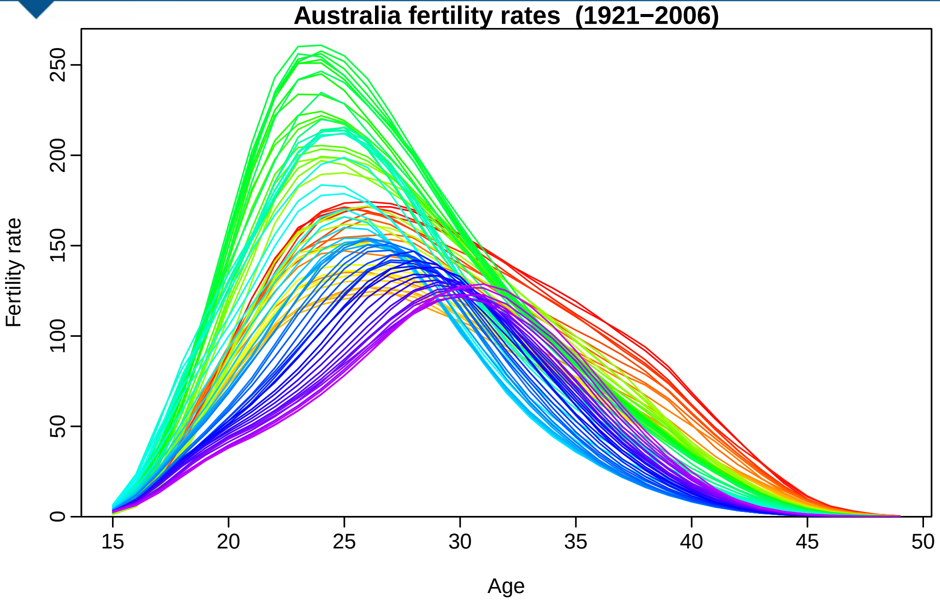
Fertility application



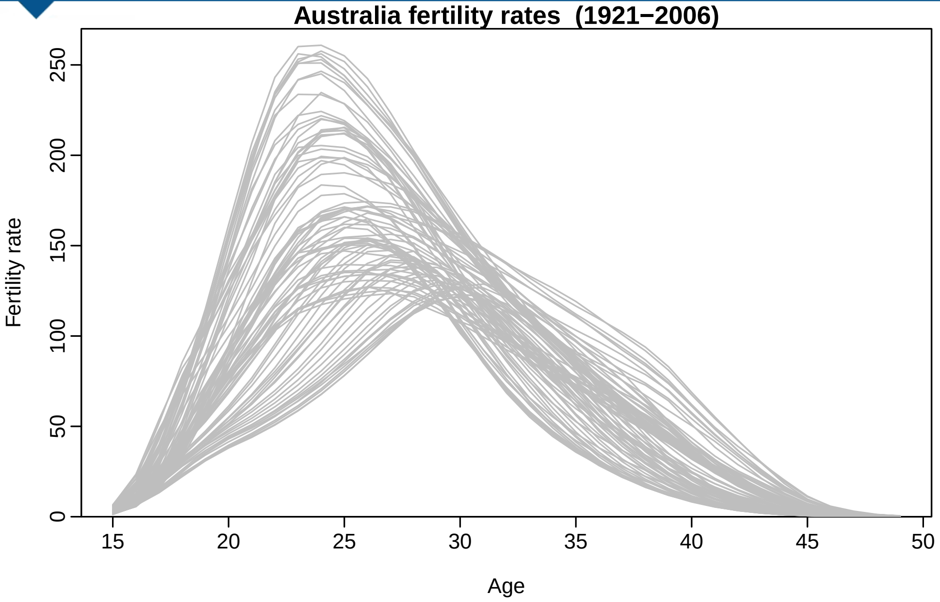
Fertility model



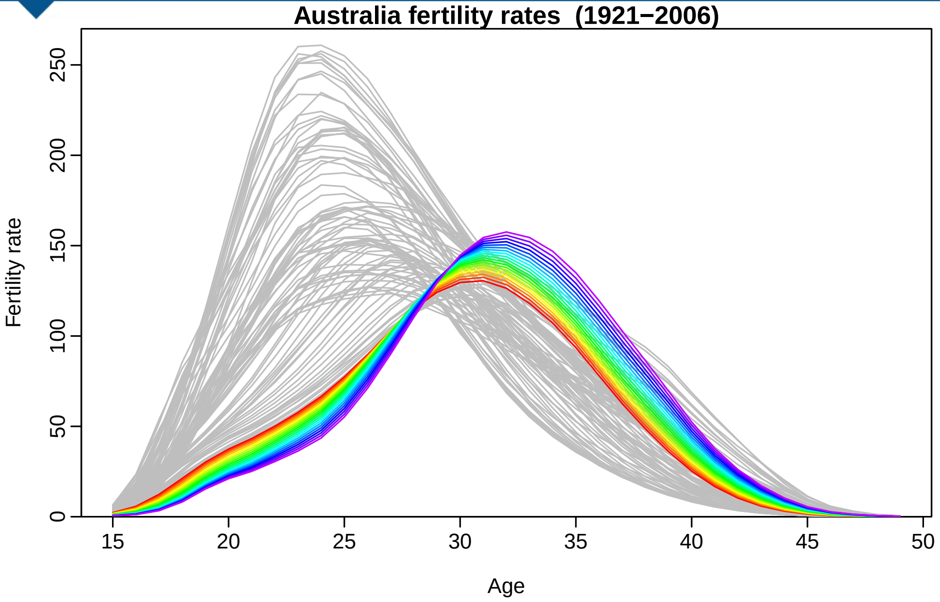
Forecasts of $f_t(x)$



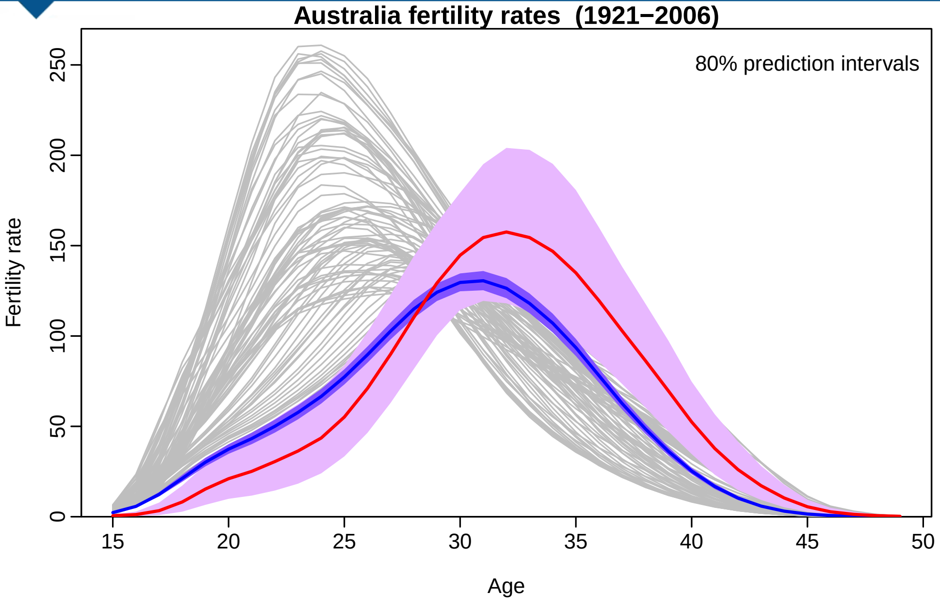
Forecasts of $f_t(x)$



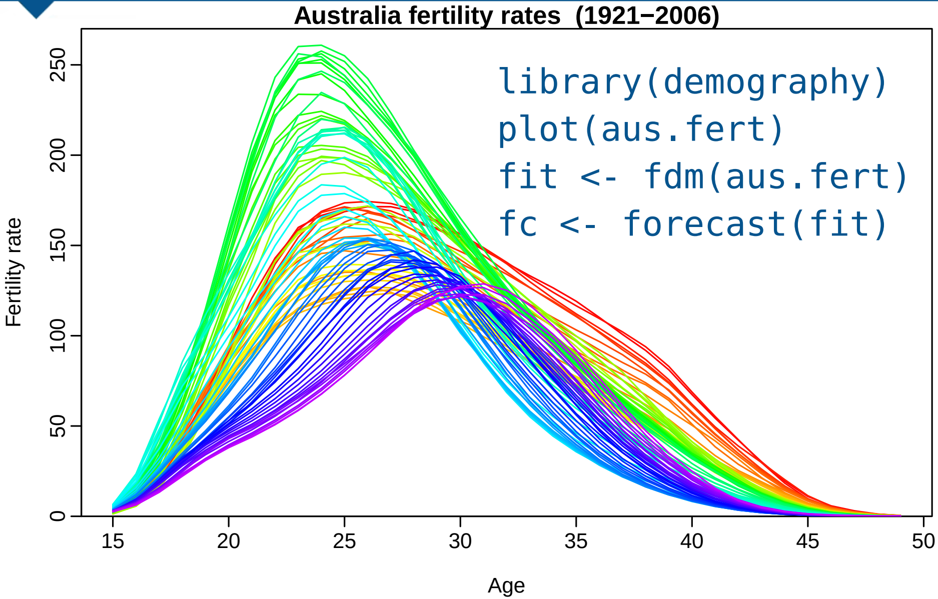
Forecasts of $f_t(x)$



Forecasts of $f_t(x)$



R code




```
library(demography)
plot(aus.fert)
fit <- fdm(aus.fert)
fc <- forecast(fit)
```

References

RJ Hyndman and S Ullah (2007). "Robust forecasting of mortality and fertility rates: A functional data approach". *Computational Statistics and Data Analysis* **51**(10), 4942–4956

RJ Hyndman and HL Shang (2009). "Forecasting functional time series (with discussion)". *Journal of the Korean Statistical Society* **38**(3), 199–221

RJ 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.