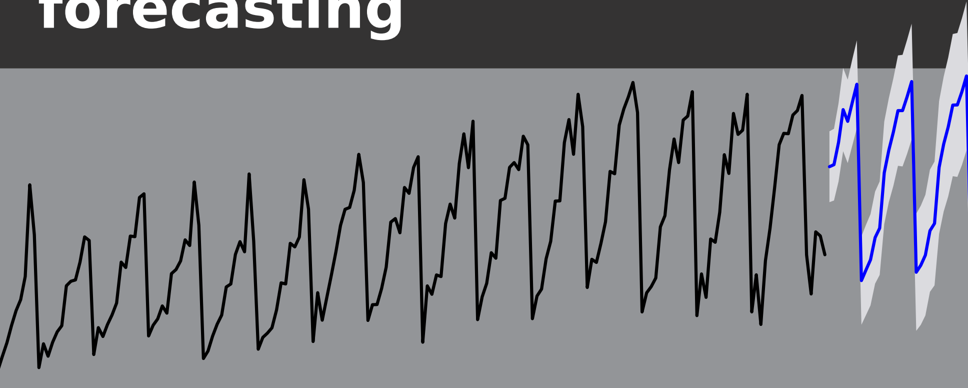
MONASH BUSINESS SCHOOL

Rob J Hyndman

Automatic algorithms for time series forecasting



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

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.

Specifications

Automatic forecasting algorithms must:

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

Motivation

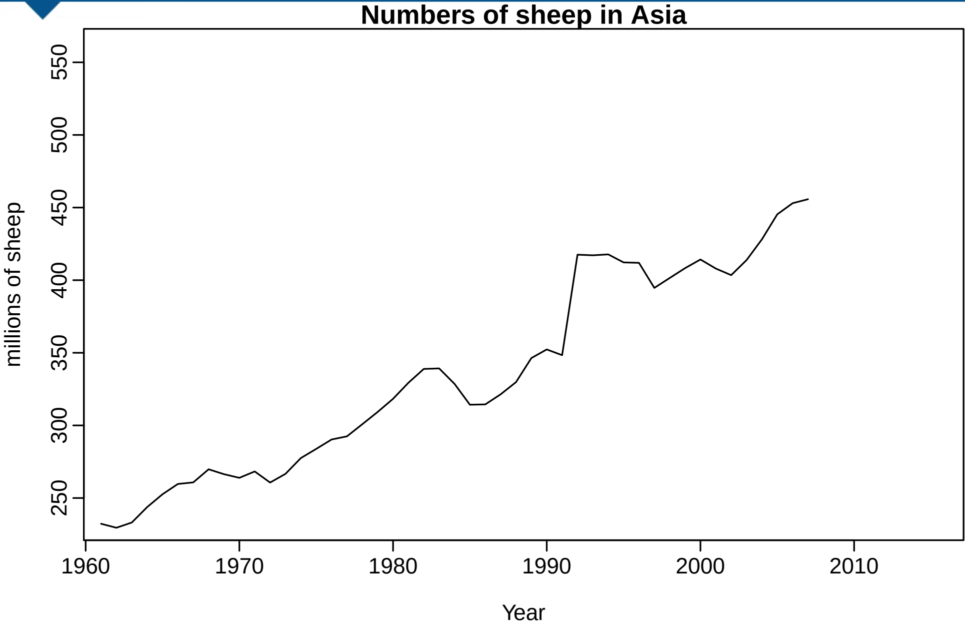
1. Common in business to have over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.

Specifications

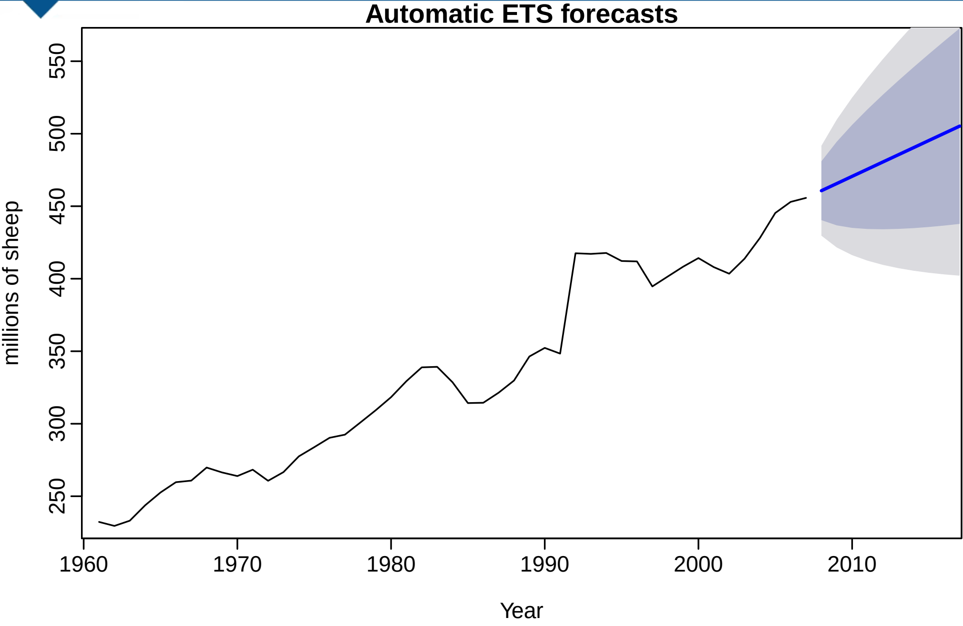
Automatic forecasting algorithms must:

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

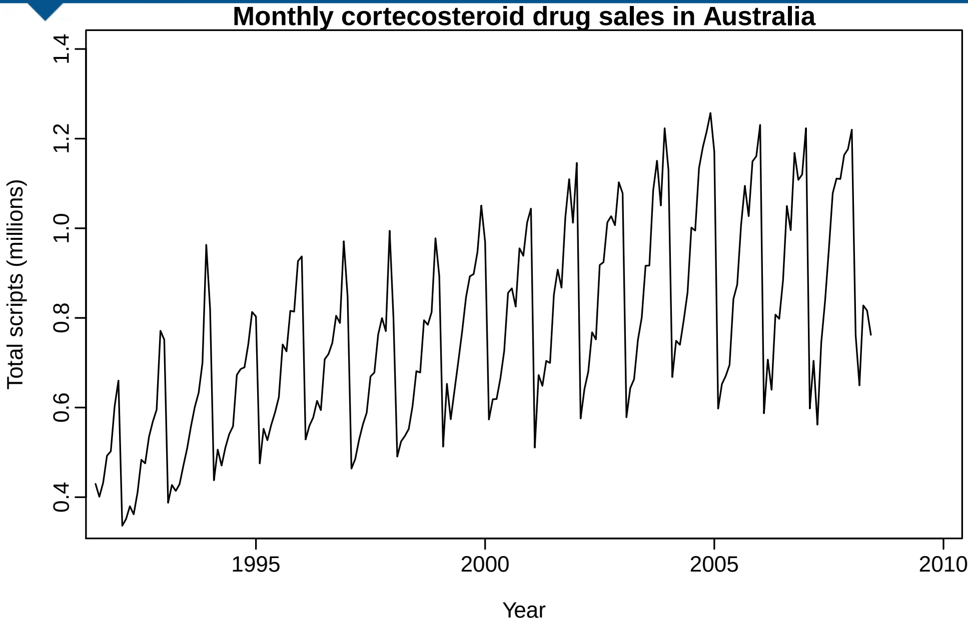
Example: Asian sheep



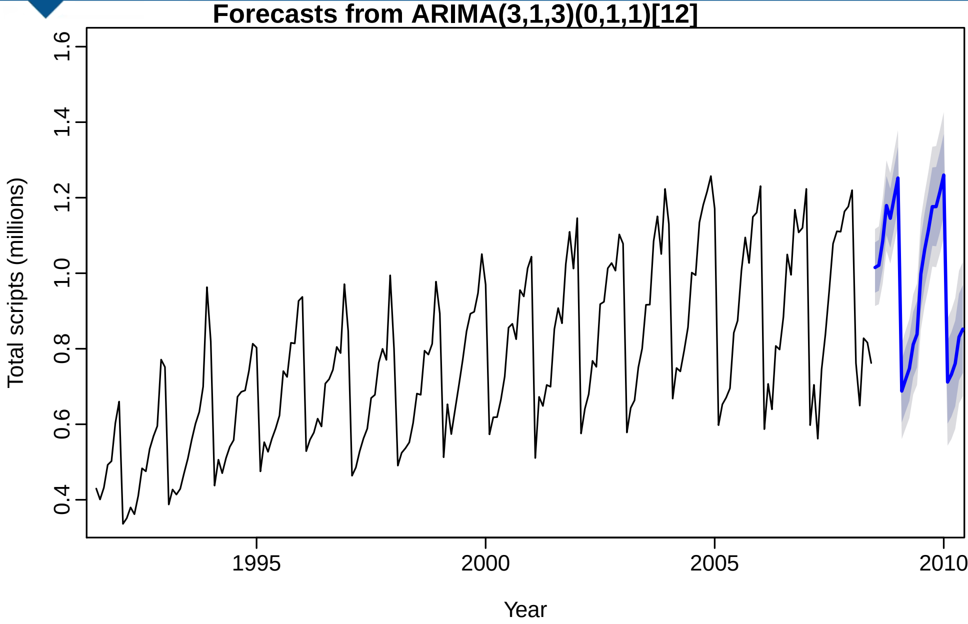
Example: Asian sheep



Example: Cortecosteroid sales



Example: Cortecosteroid sales



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

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

N,N: Simple exponential smoothing

Exponential smoothing methods

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

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

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)	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

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 exp. 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 exp. 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

- There are 15 separate exp. smoothing methods.
- Each can have an additive or multiplicative error, giving 30 separate models.
- Only 19 models are numerically stable.

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 exp. smoothing methods.
- Each can have an additive or multiplicative error, giving 30 separate models.
- Only 19 models are numerically stable.
- Multiplicative trend models give poor forecasts leaving 15 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

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

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

↑

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)	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

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

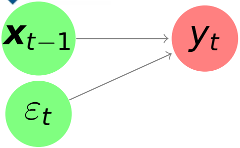
Examples:

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

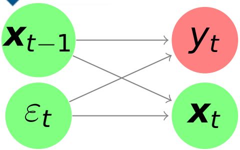
ETS state space model



State space model

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

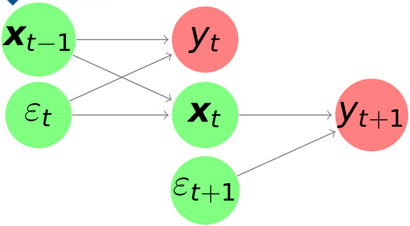
ETS state space model



State space model

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

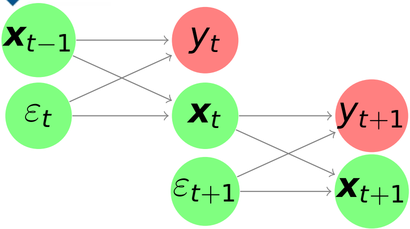
ETS state space model



State space model

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

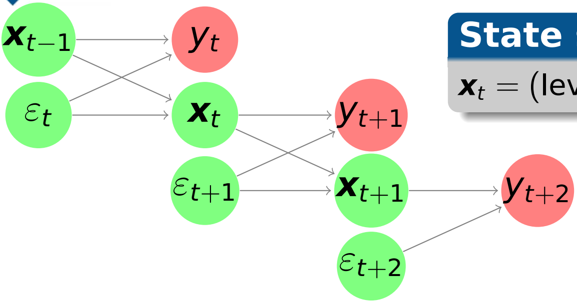
ETS state space model



State space model

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

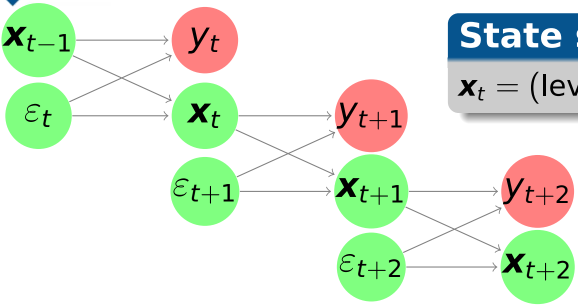
ETS state space model



State space model

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

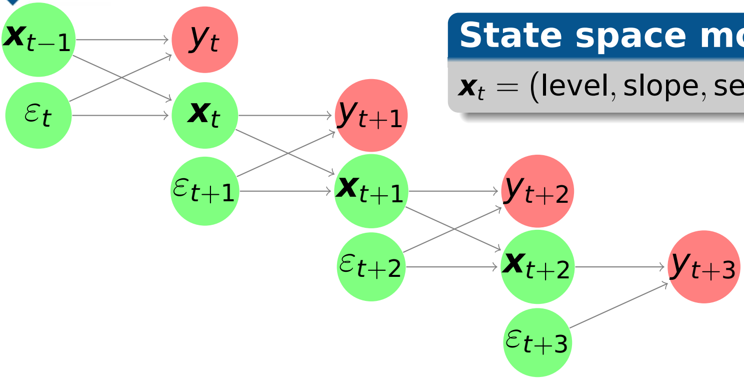
ETS state space model



State space model

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

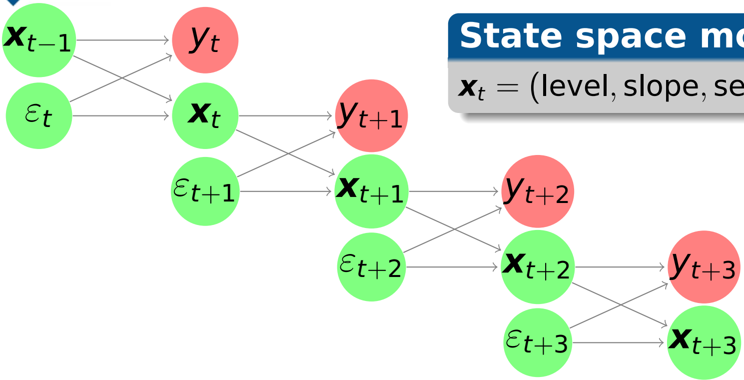
ETS state space model



State space model

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

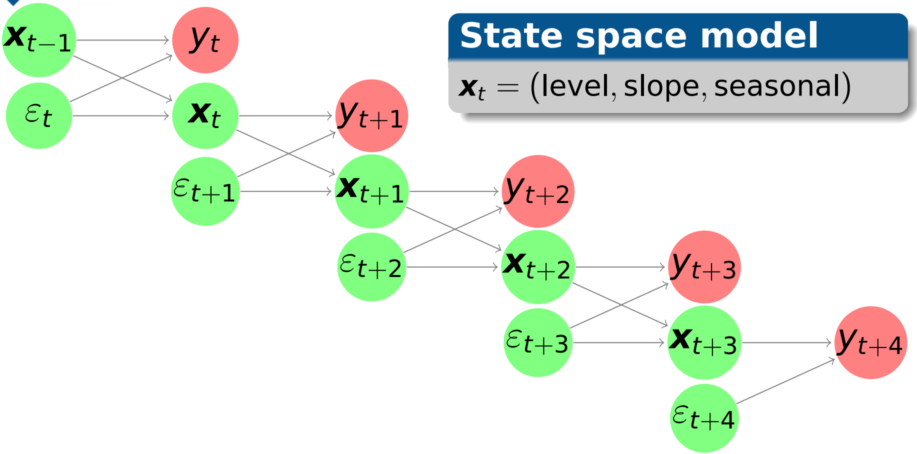
ETS state space model



State space model

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

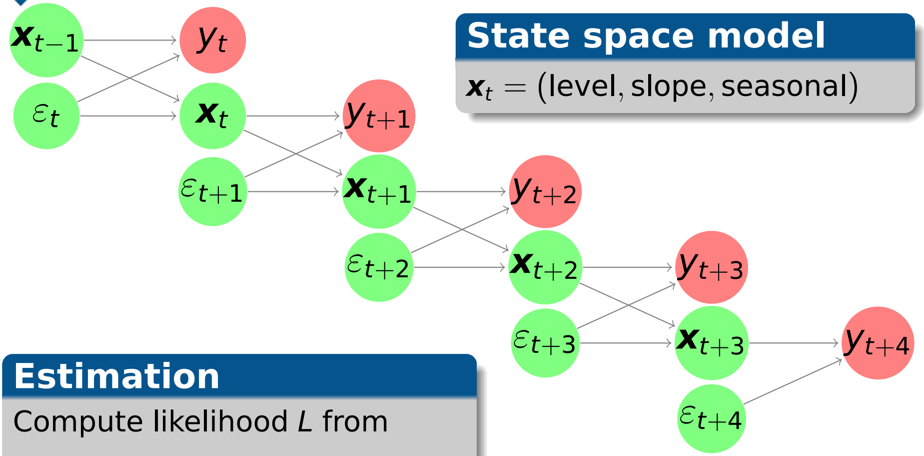
ETS state space model



State space model

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

ETS state space model



State space model

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

Estimation

Compute likelihood L from $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$.
Optimize L wrt model parameters.

Innovations state space models

Let $\boldsymbol{x}_t = (\ell_t, b_t, s_t, s_{t-1}, \dots, s_{t-m+1})$ and $\varepsilon_t \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$.

$$y_t = \underbrace{h(\boldsymbol{x}_{t-1})}_{\mu_t} + \underbrace{k(\boldsymbol{x}_{t-1})\varepsilon_t}_{e_t} \qquad \text{Observation equation}$$

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}) + g(\boldsymbol{x}_{t-1})\varepsilon_t \qquad \text{State equation}$$

Additive errors:

$$k(\boldsymbol{x}_{t-1}) = 1. \qquad y_t = \mu_t + \varepsilon_t.$$

Multiplicative errors:

$$k(\boldsymbol{x}_{t-1}) = \mu_t. \qquad y_t = \mu_t(1 + \varepsilon_t).$$

$\varepsilon_t = (y_t - \mu_t)/\mu_t$ is relative error.

Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

Estimation

$$L^*(\theta, \boldsymbol{x}_0) = n \log\left(\sum_{t=1}^{n} \varepsilon_t^2 / k^2(\boldsymbol{x}_{t-1})\right) + 2\sum_{t=1}^{n} \log|k(\boldsymbol{x}_{t-1})|$$

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

Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

Estimation

$$L^*(\theta, \boldsymbol{x}_0) = n \log\left(\sum_{t=1}^{n} \varepsilon_t^2 / k^2(\boldsymbol{x}_{t-1})\right) + 2\sum_{t=1}^{n} \log|k(\boldsymbol{x}_{t-1})|$$

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

Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

Estimation

$$L^*(\theta, \boldsymbol{x}_0) = n \log\left(\sum_{t=1}^{n} \varepsilon_t^2 / k^2(\boldsymbol{x}_{t-1})\right) + 2\sum_{t=1}^{n} \log|k(\boldsymbol{x}_{t-1})|$$
$$= -2\log(\text{Likelihood}) + \text{constant}$$

- Minimize wrt $\theta = (\alpha, \beta, \gamma, \phi)$ and initial states $\boldsymbol{x}_0 = (\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1})$.

Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

Estimation

$$
\begin{aligned}
L^*(\boldsymbol{\theta}, \boldsymbol{x}_0) &= n \log\left(\sum_{t=1}^{n} \varepsilon_t^2 / k^2(\boldsymbol{x}_{t-1}) \right) + 2 \sum_{t=1}^{n} \log |k(\boldsymbol{x}_{t-1})| \\
&= -2 \log(\text{Likelihood}) + \text{constant}
\end{aligned}
$$

- Minimize wrt $\boldsymbol{\theta} = (\alpha, \beta, \gamma, \phi)$ and initial states $\boldsymbol{x}_0 = (\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1})$.

Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

Estimation

$$L^*(\boldsymbol{\theta}, \boldsymbol{x}_0) = n \log\left(\sum_{t=1}^{n} \varepsilon_t^2 / k^2(\boldsymbol{x}_{t-1})\right) + 2\sum_{t=1}^{n} \log|k(\boldsymbol{x}_{t-1})|$$
$$= -2\log(\text{Likelihood}) + \text{constant}$$

- Minimize wrt $\boldsymbol{\theta} = (\alpha, \beta, \gamma, \phi)$ and initial states $\boldsymbol{x}_0 = (\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1})$.

Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

Estimation

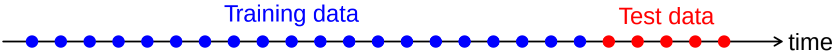
$$L^*(\boldsymbol{\theta}, \boldsymbol{x}_0) = n \log\left(\sum_{t=1}^{n} \varepsilon_t^2 / k^2(\boldsymbol{x}_{t-1})\right) + 2\sum_{t=1}^{n} \log|k(\boldsymbol{x}_{t-1})|$$
$$= -2\log(\text{Likelihood}) + \text{constant}$$

- Minimize wrt $\boldsymbol{\theta} = (\alpha, \beta, \gamma$
$\boldsymbol{x}_0 = (\ell_0, b_0, s_0, s_{-1}, \dots, s$

Q: How to choose between the 15 useful ETS models?

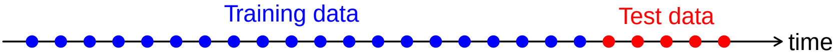
Cross-validation

Traditional evaluation

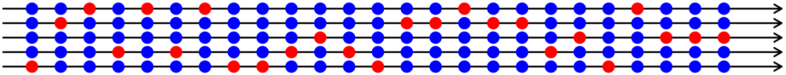


Cross-validation

Traditional evaluation

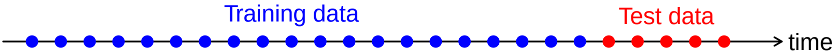


Standard cross-validation

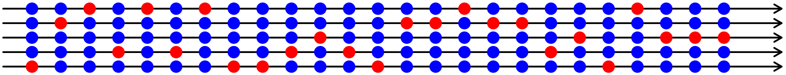


Cross-validation

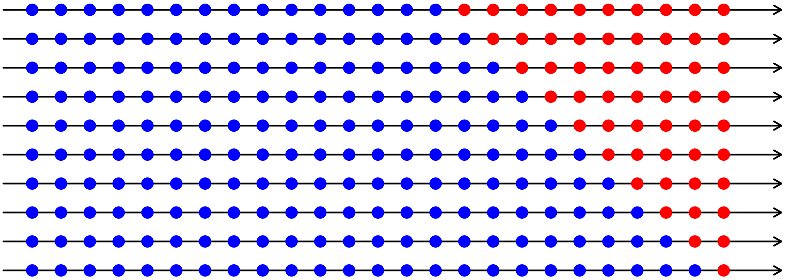
Traditional evaluation



Standard cross-validation

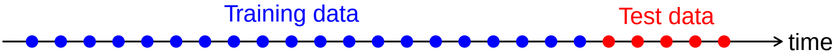


Time series cross-validation

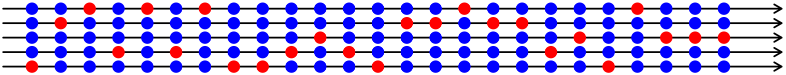


Cross-validation

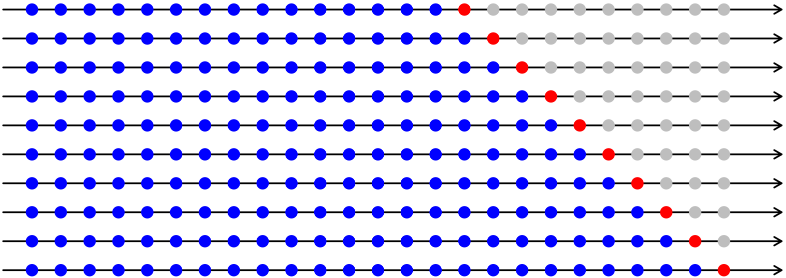
Traditional evaluation



Standard cross-validation

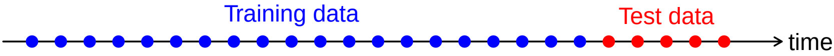


Time series cross-validation

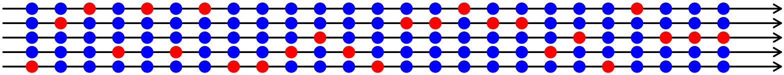


Cross-validation

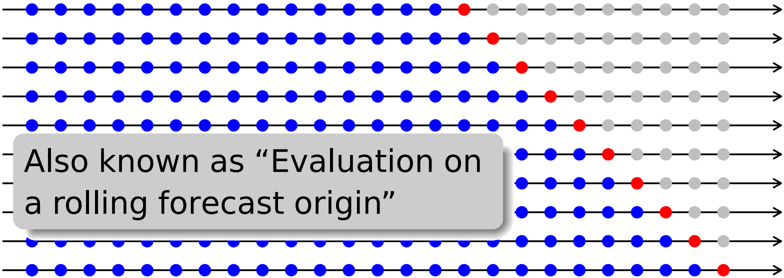
Traditional evaluation



Standard cross-validation



Time series cross-validation



Also known as "Evaluation on a rolling forecast origin"

Akaike's Information Criterion

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

where L is the likelihood and k is the number of estimated parameters in the model.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then $\text{AIC} \approx c + T\log\text{MSE} + 2k$ where c is a constant, MSE is from one-step forecasts on **training set**, and T is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as $T \to \infty$) to minimizing MSE from one-step forecasts on **test set** via time series cross-validation.

Akaike's Information Criterion

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

where L is the likelihood and k is the number of estimated parameters in the model.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then $\text{AIC} \approx c + T \log \text{MSE} + 2k$ where c is a constant, MSE is from one-step forecasts on **training set**, and T is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as $T \to \infty$) to minimizing MSE from one-step forecasts on **test set** via time series cross-validation.

Akaike's Information Criterion

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

where L is the likelihood and k is the number of estimated parameters in the model.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then $\text{AIC} \approx c + T \log \text{MSE} + 2k$ where c is a constant, MSE is from one-step forecasts on **training set**, and T is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as $T \to \infty$) to minimizing MSE from one-step forecasts on **test set** via time series cross-validation.

Akaike's Information Criterion

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

where L is the likelihood and k is the number of estimated parameters in the model.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then $\text{AIC} \approx c + T\log\text{MSE} + 2k$ where c is a constant, MSE is from one-step forecasts on **training set**, and T is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as $T \to \infty$) to minimizing MSE from one-step forecasts on **test set** via time series cross-validation.

Akaike's Information Criterion

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

where L is the likelihood and k is the number of estimated parameters in the model.

- This is a *penalized likelihood* approach.
- If L is Gaussian, then $\text{AIC} \approx c + T\log \text{MSE} + 2k$ where c is a constant, MSE is from one-step forecasts on **training set**, and T is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as $T \to \infty$) to minimizing MSE from one-step forecasts on **test set** via time series cross-validation.

Akaike's Information Criterion

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

Corrected AIC

For small T, AIC tends to over-fit. Bias-corrected version:

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

Bayesian Information Criterion

$$\text{BIC} = \text{AIC} + k[\log(T) - 2]$$

- BIC penalizes terms more heavily than AIC
- Minimizing BIC is consistent **if there is a true model**.

Akaike's Information Criterion

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

Corrected AIC

For small T, AIC tends to over-fit. Bias-corrected version:

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

Bayesian Information Criterion

$$\text{BIC} = \text{AIC} + k[\log(T) - 2]$$

- BIC penalizes terms more heavily than AIC
- Minimizing BIC is consistent **if there is a true model**.

Akaike's Information Criterion

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

Corrected AIC

For small T, AIC tends to over-fit. Bias-corrected version:

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

Bayesian Information Criterion

$$\text{BIC} = \text{AIC} + k[\log(T) - 2]$$

- BIC penalizes terms more heavily than AIC
- Minimizing BIC is consistent **if there is a true model**.

What to use?

Choice: AIC, AICc, BIC, CV-MSE

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large T.
- As $T \to \infty$, BIC selects *true* model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.
- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.

What to use?

Choice: AIC, AICc, BIC, CV-MSE

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large T.
- As $T \to \infty$, BIC selects *true* model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.
- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.

What to use?

Choice: AIC, AICc, BIC, CV-MSE

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large T.
- As $T \to \infty$, BIC selects *true* model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.
- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.

What to use?

Choice: AIC, AICc, BIC, CV-MSE

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large T.
- As $T \to \infty$, BIC selects *true* model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.
- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.

What to use?

Choice: AIC, AICc, BIC, CV-MSE

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large T.
- As $T \to \infty$, BIC selects *true* model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.
- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.

ets algorithm in R

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

- Apply each of 15 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

ets algorithm in R

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

- Apply each of 15 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

ets algorithm in R

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

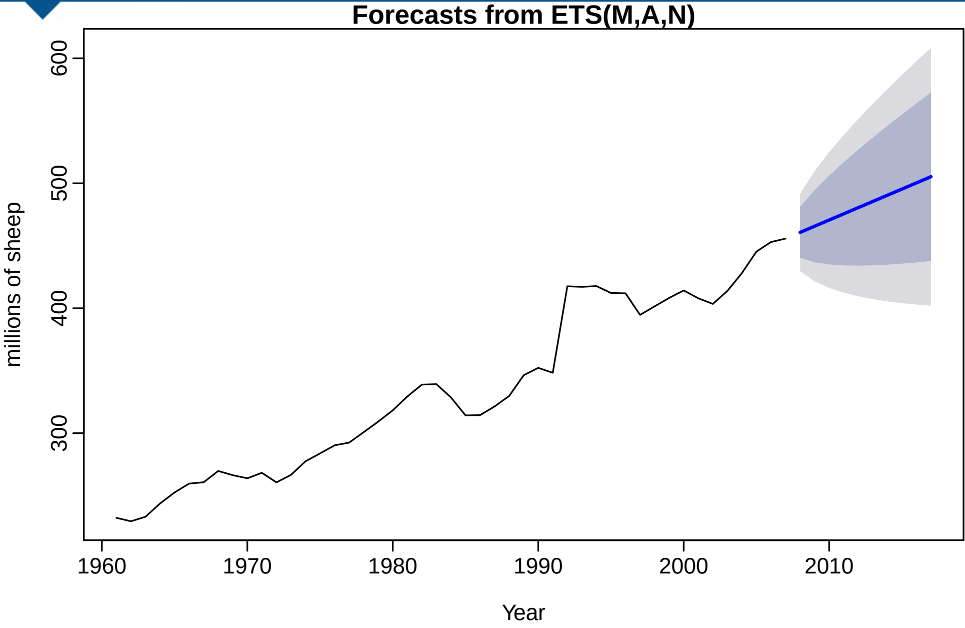
- Apply each of 15 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

ets algorithm in R

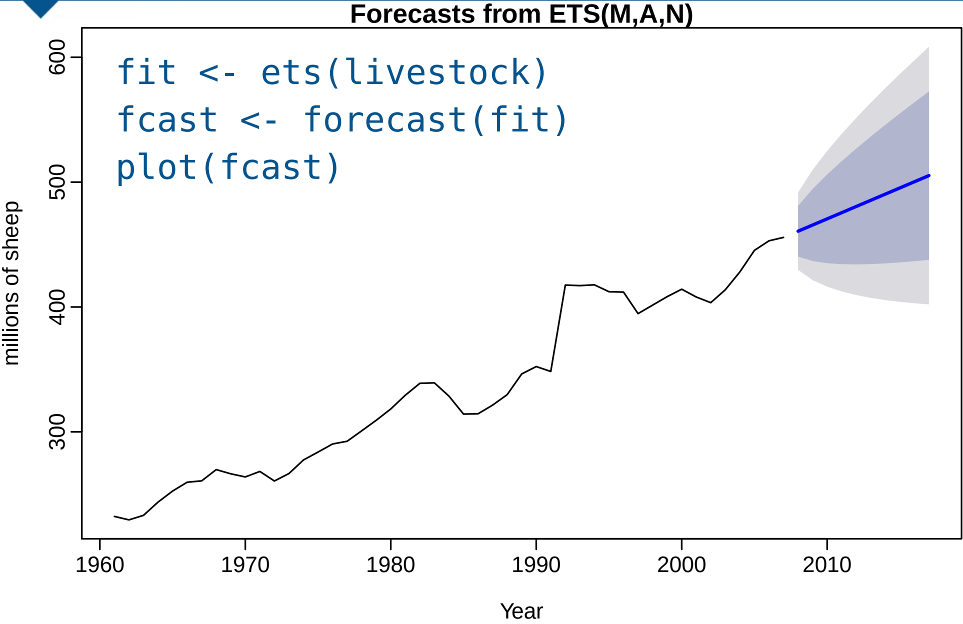
Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

- Apply each of 15 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.

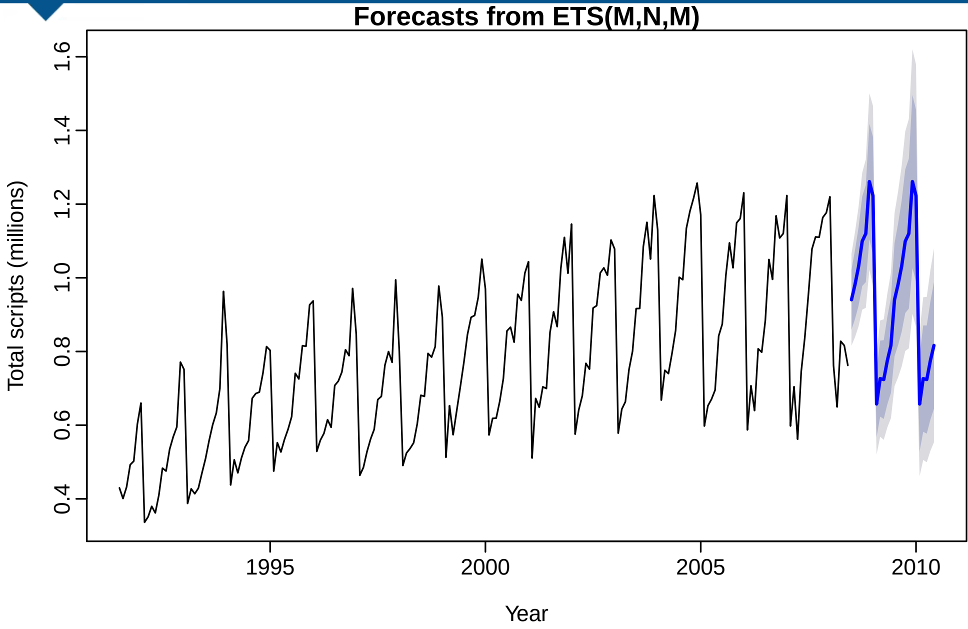
Exponential smoothing



Exponential smoothing

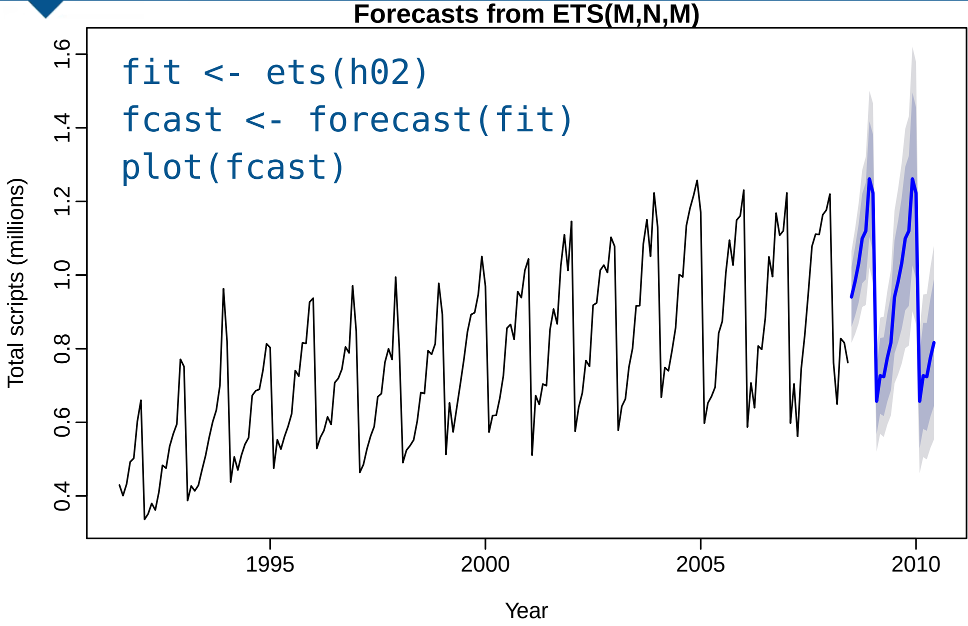


Exponential smoothing



Exponential smoothing

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



Exponential smoothing

```
> fit
ETS(M,N,M)

  Smoothing parameters:
    alpha = 0.4597
    gamma = 1e-04

  Initial states:
    l = 0.4501
    s = 0.8628 0.8193 0.7648 0.7675 0.6946 1.2921
        1.3327 1.1833 1.1617 1.0899 1.0377 0.9937

  sigma:  0.0675

       AIC       AICc        BIC
-115.69960 -113.47738  -69.24592
```

M3 comparisons

Method	MAPE	sMAPE	MASE
Theta	17.42	12.76	1.39
ForecastPro	18.00	13.06	1.47
ForecastX	17.35	13.09	1.42
Automatic ANN	17.18	13.98	1.53
B-J automatic	19.13	13.72	1.54
ETS	17.38	13.13	1.43

Exponential smoothing

7 Exponential smoothing

Exponential smoothing was proposed in the late 1950s (Brown 1959, Holt 1957 and Winters 1960 are key pioneering works) and has motivated some of the most successful forecasting methods. Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight. This framework generates reliable forecasts quickly and for a wide spectrum of time series which is a great advantage and of major importance to applications in industry.

This chapter is divided into two parts. In the first part we present in detail the mechanics of all exponential smoothing methods and their application in forecasting time series with various characteristics. This is key in understanding the intuition behind these methods. In this setting, selecting and using a forecasting method may appear to be somewhat ad-hoc. The

Book information

About this book

Feedback on this book

Rob J Hyndman
George Athanasopoulos

Forecasting: principles and practice

Exponential smoothing

https://www.otexts.org/fpp/7

OTexts ONLINE, OPEN-ACCESS TEXTBOOKS

Home Books Authors About Donation

Search

Home » Forecasting: principles and practice » 7 Exponential smoothing

7 Exponential smoothing

Exponential smoothing was proposed in the late 1950s (Brown 1959, Holt 1957 and Winters 1960 are key pioneering works) and has motivated some of the most successful forecasting methods. Forecasts produced using exponential smoothing r… weights de… the more re… framework ge… time series which is a great advantage and of major importance to applications in industry.

www.OTexts.org/fpp

This chapter is divided into two parts. In the first part we present in detail the mechanics of all exponential smoothing methods and their application in forecasting time series with various characteristics. This is key in understanding the intuition behind these methods. In this setting, selecting and using a forecasting method may appear to be somewhat ad-hoc. The

Book information

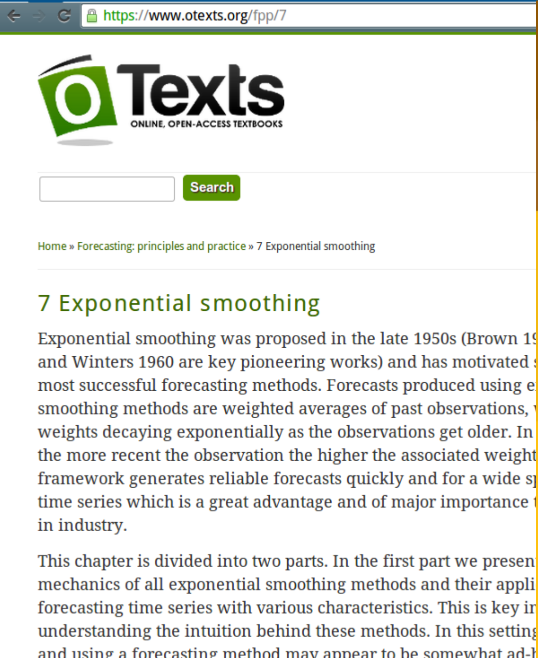
About this book

Feedback on this book

Rob J Hyndman
George Athanasopoulos

Forecasting: principles and practice

Exponential smoothing


https://www.otexts.org/fpp/7

OTexts
ONLINE, OPEN-ACCESS TEXTBOOKS

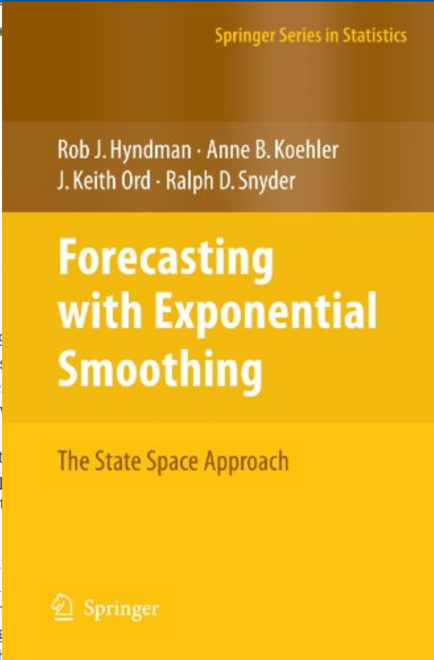
Search

Home » Forecasting: principles and practice » 7 Exponential smoothing

7 Exponential smoothing

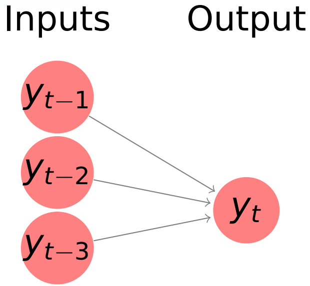
Exponential smoothing was proposed in the late 1950s (Brown 19
and Winters 1960 are key pioneering works) and has motivated s
most successful forecasting methods. Forecasts produced using e
smoothing methods are weighted averages of past observations, w
weights decaying exponentially as the observations get older. In
the more recent the observation the higher the associated weight
framework generates reliable forecasts quickly and for a wide sp
time series which is a great advantage and of major importance t
in industry.

This chapter is divided into two parts. In the first part we presen
mechanics of all exponential smoothing methods and their appli
forecasting time series with various characteristics. This is key ir
understanding the intuition behind these methods. In this setting
and using a forecasting method may appear to be somewhat ad-h

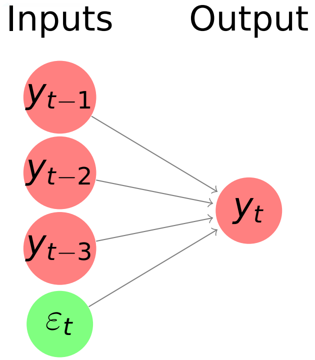


Outline

ARIMA models

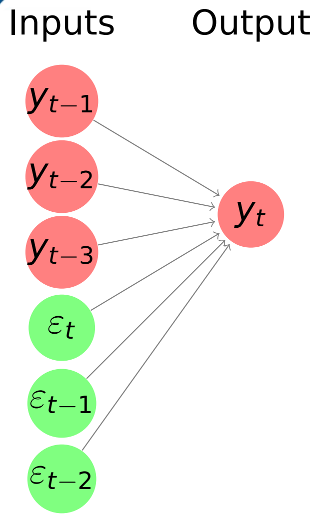


ARIMA models



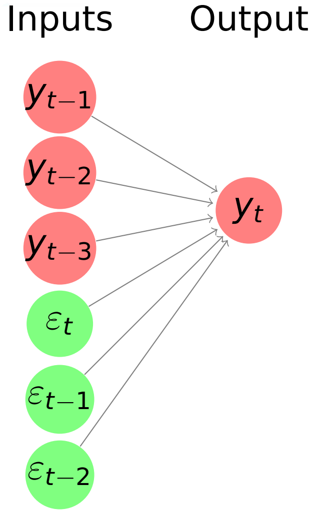
Autoregression (AR) model

ARIMA models



Autoregression moving average (ARMA) model

ARIMA models

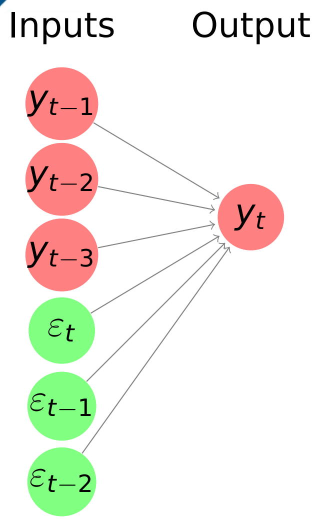


Autoregression moving average (ARMA) model

Estimation

Compute likelihood L from $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T$.
Use optimization algorithm to maximize L.

ARIMA models



Autoregression moving average (ARMA) model

ARIMA model

Autoregression moving average (ARMA) model applied to differences.

Estimation

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

Use optimization algorithm to maximize L.

ARIMA modelling

ELSEVIER

International Journal of Forecasting 16 (2000) 497–508

international journal of forecasting

www.elsevier.com/locate/ijforecast

Automatic ARIMA modeling including interventions, using time series expert software

G. Mélard*, J.-M. Pasteels

ISRO CP 210 (bldg NO room 2.O.9.300), Campus Plaine, Université Libre de Bruxelles, Bd du Triomphe, B-1050 Bruxelles, Belgium

Abstract

This article has three objectives: (a) to describe the method of automatic ARIMA modeling (AAM), with and without intervention analysis, that has been used in the analysis; (b) to comment on the results; and (c) to comment on the M3 Competition in general. Starting with a computer program for fitting an ARIMA model and a methodology for building univariate ARIMA models, an expert system has been built, while trying to avoid the pitfalls of most existing software packages. A software package called Time Series Expert TSE-AX is used to build a univariate ARIMA model with or without an intervention analysis. The characteristics of TSE-AX are summarized and, more especially, its automatic ARIMA modeling method. The motivation to take part in the M3-Competition is also outlined. The methodology is described mainly

A Course in Time Series Analysis
Edited by Daniel Peña, George C. Tiao and Ruey S. Tsay

CHAPTER 7

Automatic Modeling Methods for Univariate Series

Víctor Gómez
Ministerio de Hacienda

Agustín Maravall
Banco de España

ARIMA modelling

Journal of Statistical Software

July 2008, Volume 26, Issue 3. http://www.jstatsoft.org/

Automatic Time Series Forecasting: The forecast Package for R

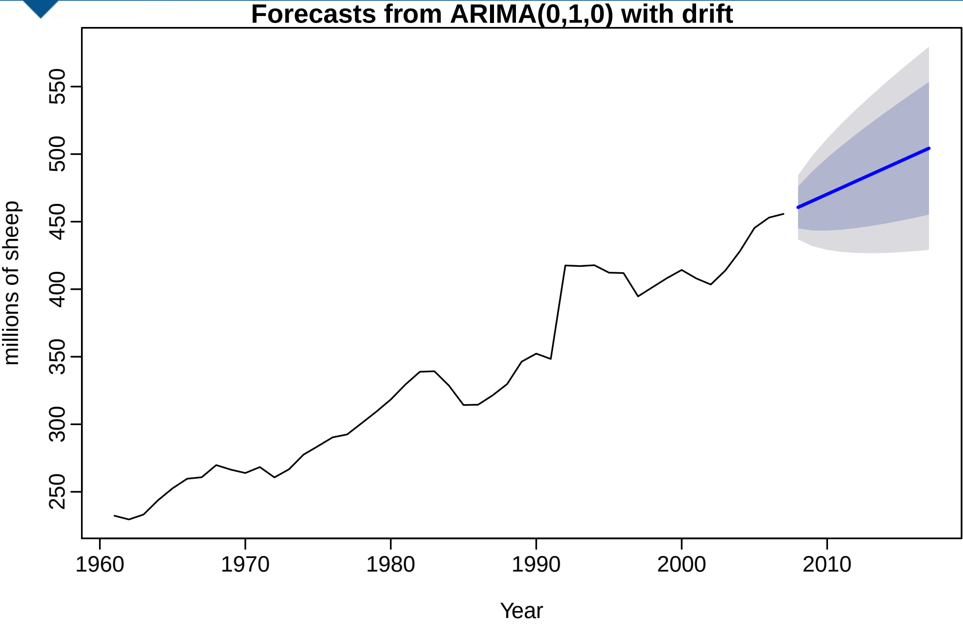
Rob J. Hyndman
Monash University

Yeasmin Khandakar
Monash University

Abstract

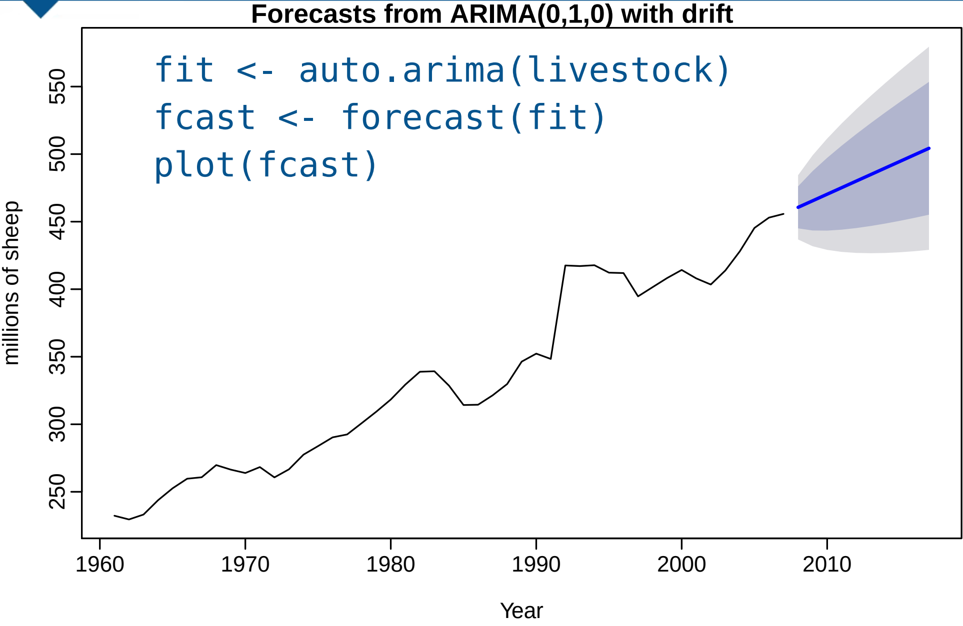
Automatic forecasts of large numbers of univariate time series are often needed in

Auto ARIMA

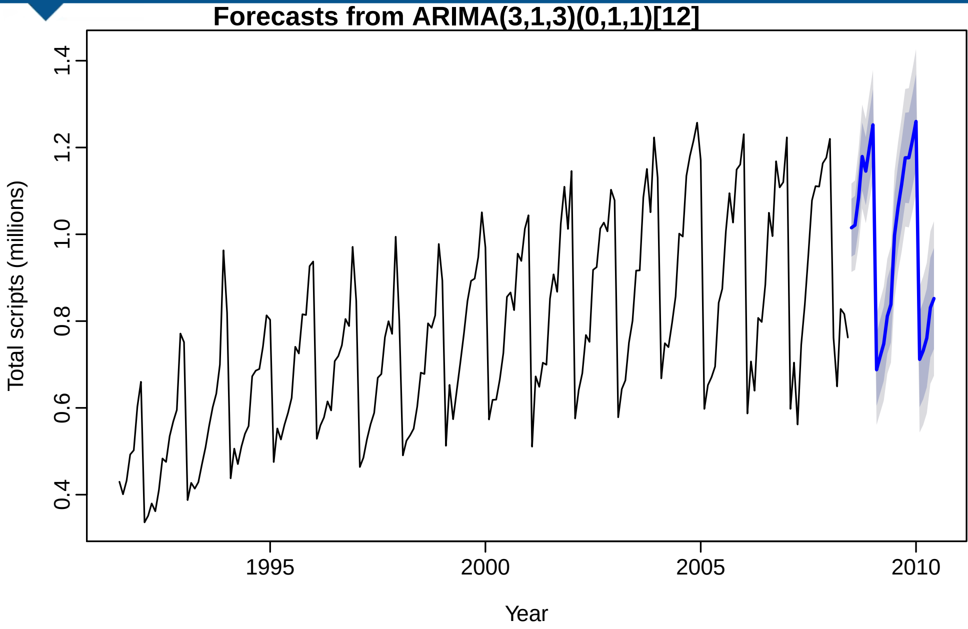

Forecasts from ARIMA(0,1,0) with drift
millions of sheep
550
500
450
400
350
300
250
1960
1970
1980
1990
2000
2010
Year

Auto ARIMA

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

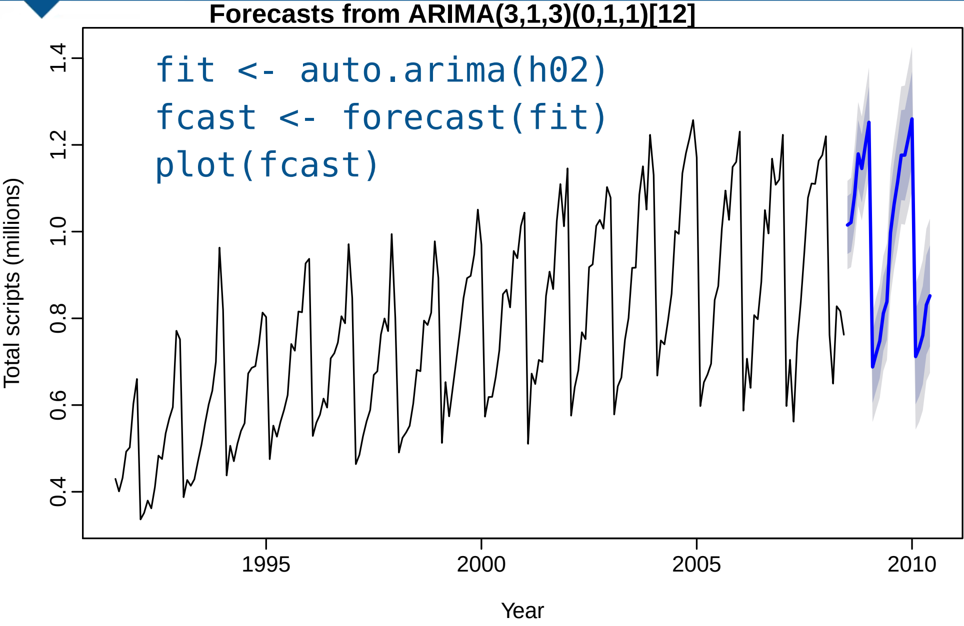


Auto ARIMA


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

Auto ARIMA

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



Auto ARIMA

```
> fit
Series: h02
ARIMA(3,1,3)(0,1,1)[12]

Coefficients:
          ar1      ar2     ar3      ma1     ma2     ma3     sma1
      -0.3648  -0.0636  0.3568  -0.4850  0.0479  -0.353  -0.5931
s.e.   0.2198   0.3293  0.1268   0.2227  0.2755   0.212   0.0651

sigma^2 estimated as 0.002706:  log likelihood=290.25
AIC=-564.5   AICc=-563.71   BIC=-538.48
```

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 AICc.
- 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 AICc.
- 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 AICc.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.

M3 comparisons

Method	MAPE	sMAPE	MASE
Theta	17.42	12.76	1.39
ForecastPro	18.00	13.06	1.47
B-J automatic	19.13	13.72	1.54
ETS	17.38	13.13	1.43
AutoARIMA	19.12	13.85	1.47

Outline

Automatic nonlinear forecasting

- Automatic ANN in M3 competition did poorly.
- Linear methods did best in the NN3 competition!
- Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.
- Some good recent work by Kourentzes and Crone on automated ANN for time series.
- **Watch this space!**

Automatic nonlinear forecasting

- Automatic ANN in M3 competition did poorly.
- Linear methods did best in the NN3 competition!
- Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.
- Some good recent work by Kourentzes and Crone on automated ANN for time series.
- **Watch this space!**

Automatic nonlinear forecasting

- Automatic ANN in M3 competition did poorly.
- Linear methods did best in the NN3 competition!
- Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.
- Some good recent work by Kourentzes and Crone on automated ANN for time series.
- **Watch this space!**

Automatic nonlinear forecasting

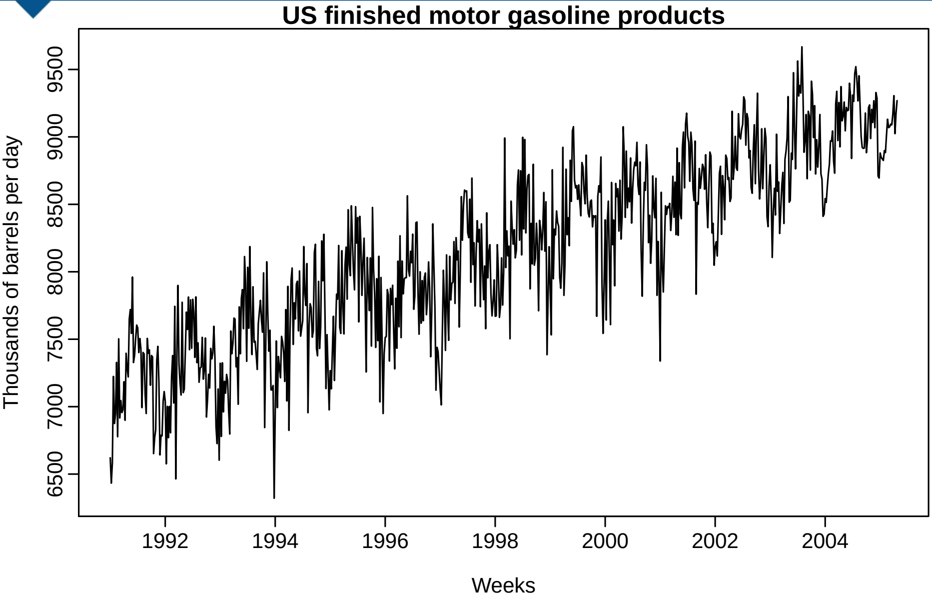
- Automatic ANN in M3 competition did poorly.
- Linear methods did best in the NN3 competition!
- Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.
- Some good recent work by Kourentzes and Crone on automated ANN for time series.
- **Watch this space!**

Automatic nonlinear forecasting

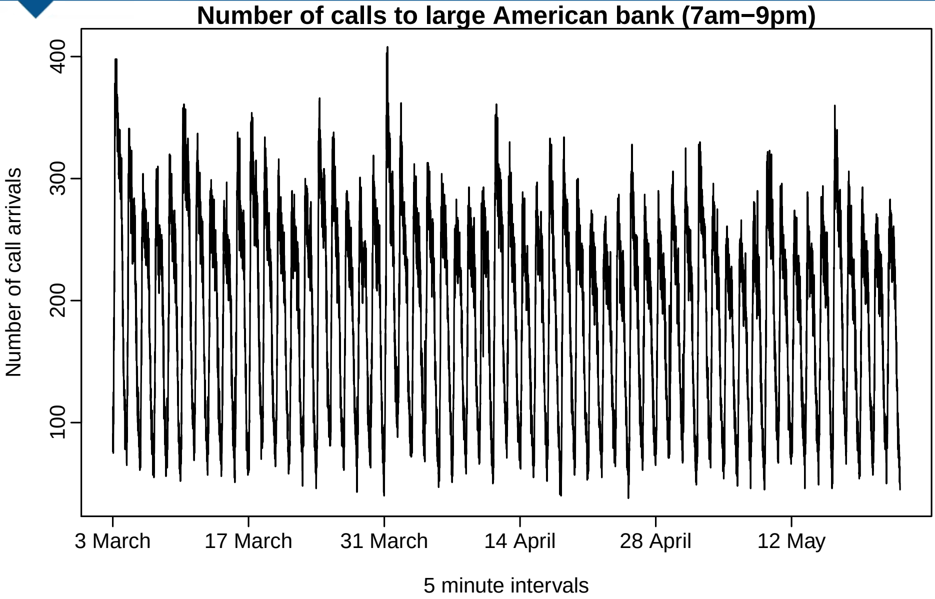
- Automatic ANN in M3 competition did poorly.
- Linear methods did best in the NN3 competition!
- Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.
- Some good recent work by Kourentzes and Crone on automated ANN for time series.
- **Watch this space!**

Outline

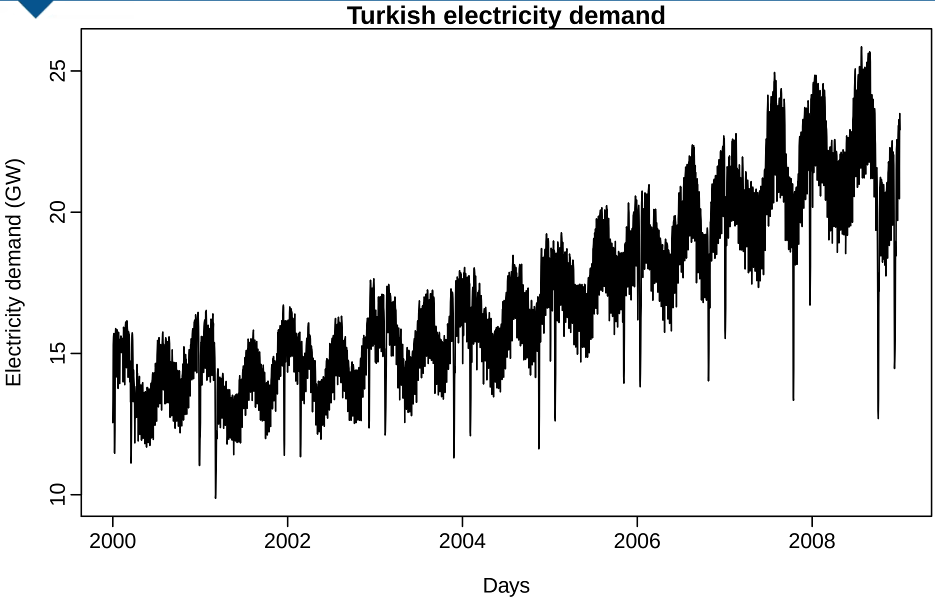
Examples


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

TBATS

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

Automatic algorithm described in AM De Livera, RJ Hyndman, and RD Snyder (2011). "Forecasting time series with complex seasonal patterns using exponential smoothing". In: *Journal of the American Statistical Association* 106.496, pp. 1513–1527. URL: `http://pubs.amstat.org/doi/abs/10.1198/jasa.2011.tm09771`

TBATS model

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Box-Cox transformation

TBATS

Trigonometric

Box-Cox

ARMA

Trend

Seasonal

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

M seasonal periods

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

$$b_t = (1 -$$

global and local trend

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

ARMA error

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

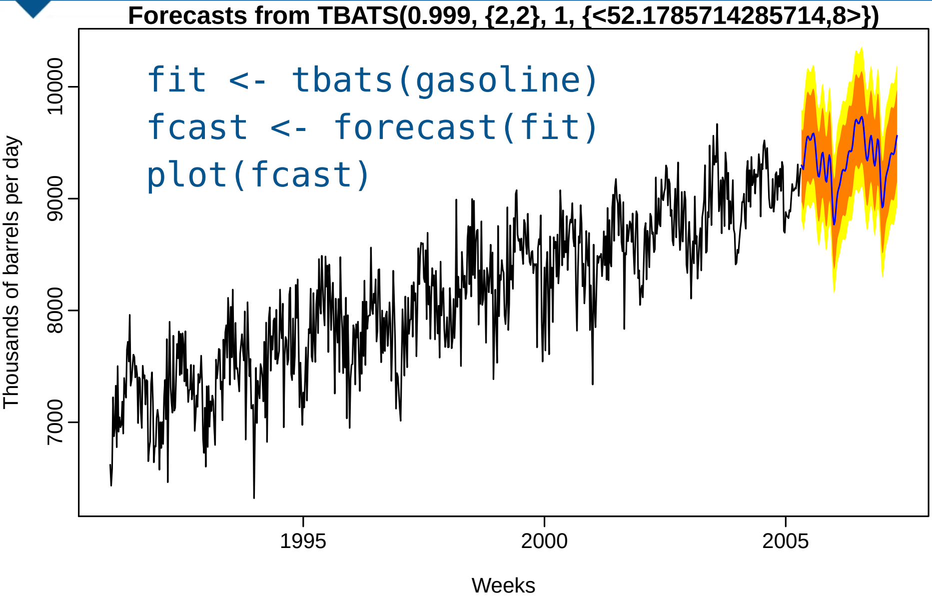
$$s_{j,t}^{(i)} = \quad s_{j,t-1}^{(i)} \, \text{c}$$

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

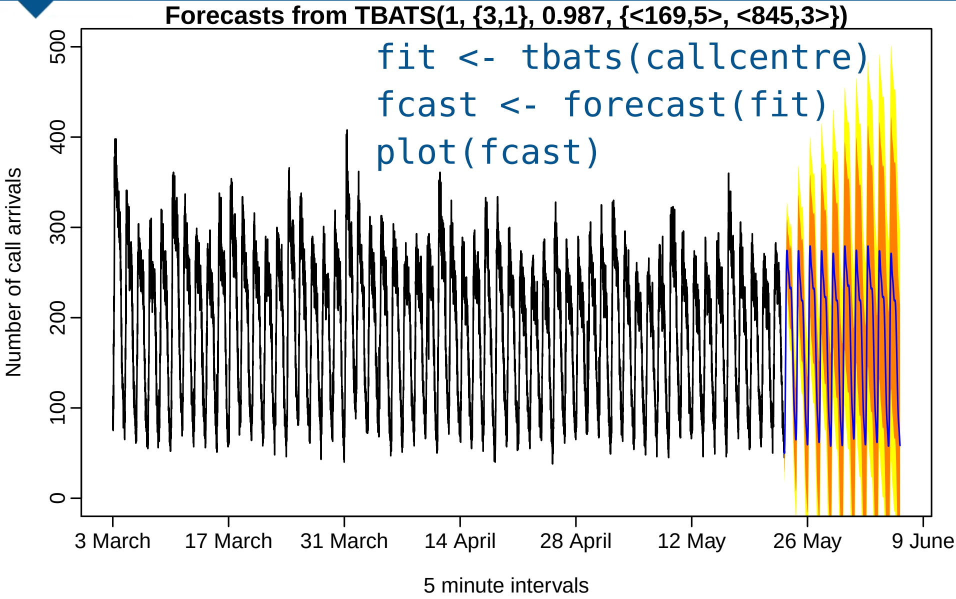
Fourier-like seasonal terms

Examples

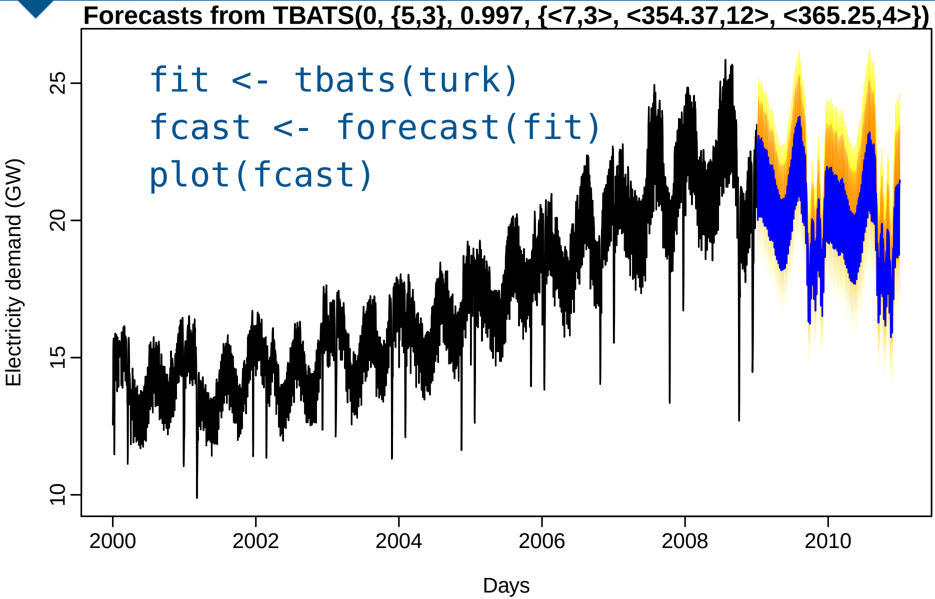
```
fit <- tbats(gasoline)
fcast <- forecast(fit)
plot(fcast)
```



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



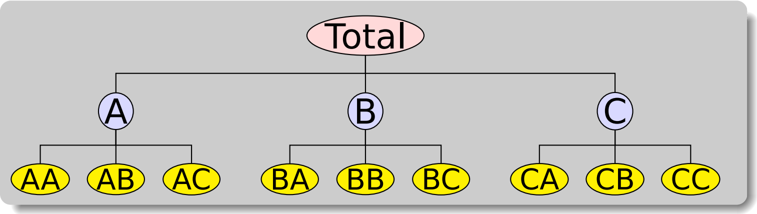
Forecasts from TBATS(0, {5,3}, 0.997, {<7,3>, <354.37,12>, <365.25,4>})
fit <- tbats(turk)
fcast <- forecast(fit)
plot(fcast)
Electricity demand (GW)
10
15
20
25
2000
2002
2004
2006
2008
2010
Days

Outline

1. Motivation
2. Exponential smoothing
3. ARIMA modelling
4. Automatic nonlinear forecasting?
5. Time series with complex seasonality
6. **Hierarchical and grouped time series**
7. The future of forecasting

Hierarchical time series

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

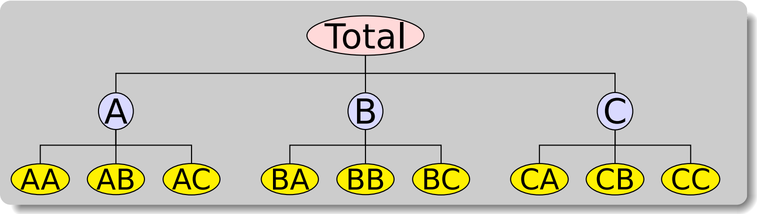


Examples

- Net labour turnover
- Tourism by state and region

Hierarchical time series

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

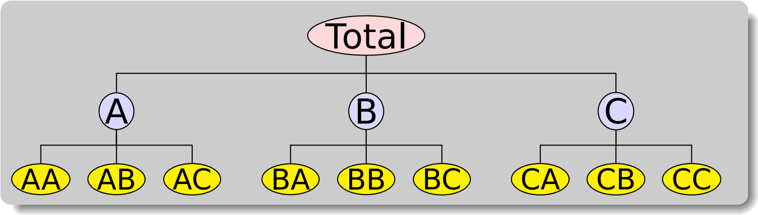


Examples

- Net labour turnover
- Tourism by state and region

Hierarchical time series

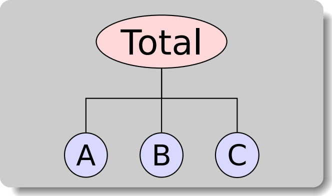
A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.



Examples

- Net labour turnover
- Tourism by state and region

Hierarchical time series

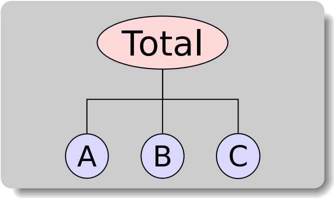


Y_t : observed aggregate of all series at time t.

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

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

Hierarchical time series

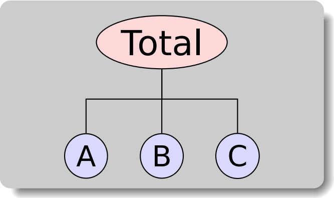


Y_t : observed aggregate of all series at time t.

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

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

Hierarchical time series



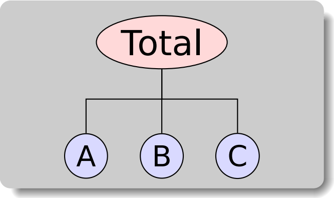
Y_t : observed aggregate of all series at time t.

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

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

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

Hierarchical time series



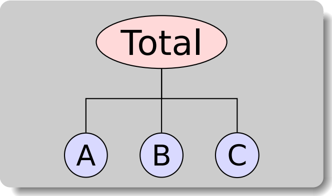
Y_t : observed aggregate of all series at time t.

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

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

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

Hierarchical time series



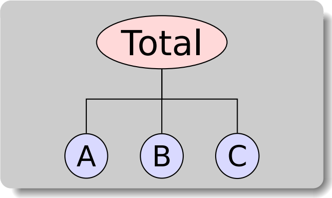
Y_t : observed aggregate of all series at time t.

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

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

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

Hierarchical time series



Y_t : observed aggregate of all series at time t.

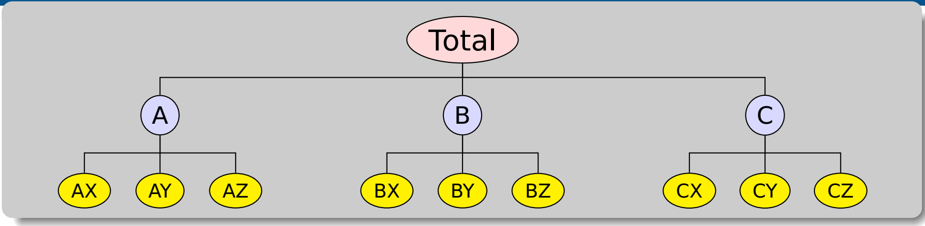
$Y_{X,t}$: observation on series X at time t.

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

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

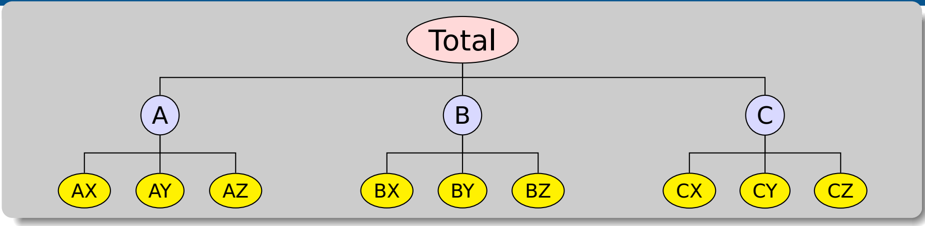
$$\boxed{\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t}$$

Hierarchical time series



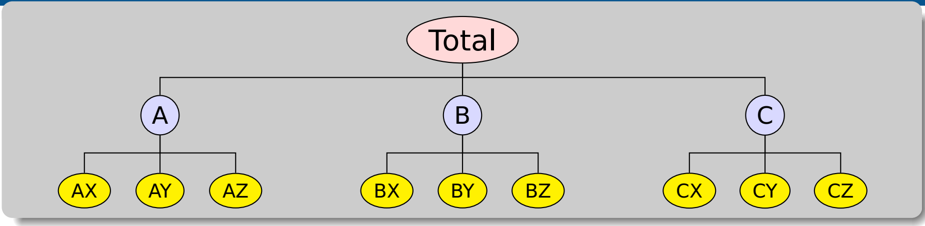
$$
\boldsymbol{y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\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

Hierarchical time series



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

Hierarchical time series



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

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

Forecasting notation

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

Reconciled forecasts are of the form:

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

for some matrix $\boldsymbol{P}$.

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

Forecasting notation

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

Reconciled forecasts are of the form:

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

for some matrix $\boldsymbol{P}$.

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

Forecasting notation

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

Reconciled forecasts are of the form:

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

for some matrix $\boldsymbol{P}$.

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

Forecasting notation

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

Reconciled forecasts are of the form:

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

for some matrix $\boldsymbol{P}$.

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

Forecasting notation

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

Reconciled forecasts are of the form:

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

for some matrix $\boldsymbol{P}$.

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

General properties

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

Forecast bias

Assuming the base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased, then the revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

Forecast variance

For any given $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, the covariance matrix of the h-step ahead reconciled forecast errors is given by

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

where $\boldsymbol{W}_h$ is the covariance matrix of the h-step ahead base forecast errors.

General properties

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

Forecast bias

Assuming the base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased, then the revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

Forecast variance

For any given $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, the covariance matrix of the h-step ahead reconciled forecast errors is given by

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

where $\boldsymbol{W}_h$ is the covariance matrix of the h-step ahead base forecast errors.

General properties

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

Forecast bias

Assuming the base forecasts $\hat{\boldsymbol{y}}_n(h)$ are unbiased, then the revised forecasts are unbiased iff $\boldsymbol{SPS} = \boldsymbol{S}$.

Forecast variance

For any given $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, the covariance matrix of the h-step ahead reconciled forecast errors is given by

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

where $\boldsymbol{W}_h$ is the covariance matrix of the h-step ahead base forecast errors.

BLUF via trace minimization

Theorem

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

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

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

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.

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

Revised forecasts　　Base forecasts

BLUF via trace minimization

Theorem

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

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

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

- $\boldsymbol{W}_h^\dagger$ is generalized inverse of $\boldsymbol{W}_h$.

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

Revised forecasts | Base forecasts

- Equivalent to GLS estimate of regression $\hat{\boldsymbol{y}}_n(h) = \boldsymbol{S}\beta_n(h) + \varepsilon_h$ where $\varepsilon \sim N(\boldsymbol{0}, \boldsymbol{W}_h)$.

BLUF via trace minimization

Theorem

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

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

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

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.

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

Revised forecasts Base forecasts

- Equivalent to GLS estimate of regression $\hat{\boldsymbol{y}}_n(h) = \boldsymbol{S}\beta_n(h) + \varepsilon_h$ where $\varepsilon \sim N(\boldsymbol{0}, \boldsymbol{W}_h)$.
- Problem: $\boldsymbol{W}_h$ hard to estimate.

BLUF via trace minimization

Theorem

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

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

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

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.

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

Revised forecasts

Base forecasts

- Equivalent to GLS estimate of regression $\hat{\boldsymbol{y}}_n(h) = \boldsymbol{S}\beta_n(h) + \varepsilon_h$ where $\varepsilon \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{W}_h)$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate.

BLUF via trace minimization

Theorem

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

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

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

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.

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

Revised forecasts

Base forecasts

- Equivalent to GLS estimate of regression $\hat{\boldsymbol{y}}_n(h) = \boldsymbol{S}\beta_n(h) + \varepsilon_h$ where $\varepsilon \sim \text{N}(\boldsymbol{0}, \boldsymbol{W}_h)$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate.

BLUF via trace minimization

Theorem

For any $\boldsymbol{P}$ satisfying $\boldsymbol{SPS} = \boldsymbol{S}$, then
$$\min_{\boldsymbol{P}} = \text{trace}[\boldsymbol{SPW}_h\boldsymbol{P}'\boldsymbol{S}']$$
has solution $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{\dagger}$.

- $\boldsymbol{W}_h^{\dagger}$ is generalized inverse of $\boldsymbol{W}_h$.

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

Revised forecasts   Base forecasts

- Equivalent to GLS estimate of regression $\hat{\boldsymbol{y}}_n(h) = \boldsymbol{S}\beta_n(h) + \varepsilon_h$ where $\varepsilon \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{W}_h)$.
- **Problem:** $\boldsymbol{W}_h$ hard to estimate.

Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate $\boldsymbol{W}_1$ by its diagonal.
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate $\boldsymbol{W}_1$ by its diagonal.
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Optimal combination forecasts

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

Revised forecasts   Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate $\boldsymbol{W}_1$ by its diagonal.
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Optimal combination forecasts

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

Revised forecasts Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate $\boldsymbol{W}_1$ by its diagonal.
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution 1: OLS

$$\color{red}{\tilde{\boldsymbol{y}}_n(h)} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\color{blue}{\hat{\boldsymbol{y}}_n(h)}$$

Solution 2: WLS

- Approximate $\boldsymbol{W}_1$ by its diagonal.
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best one-step forecasts.

$$\color{red}{\tilde{\boldsymbol{y}}_n(h)} = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\color{blue}{\hat{\boldsymbol{y}}_n(h)}$$

Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate $\boldsymbol{W}_1$ by its diagonal.
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate $\boldsymbol{W}_1$ by its diagonal.
- Assume $\boldsymbol{W}_h = k_h\boldsymbol{W}_1$.
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Challenges

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

- Computational difficulties in big hierarchies due to size of the $\boldsymbol{S}$ matrix and singular behavior of $(\boldsymbol{S}'\Lambda\boldsymbol{S})$.
- Loss of information in ignoring covariance matrix in computing point forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Challenges

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

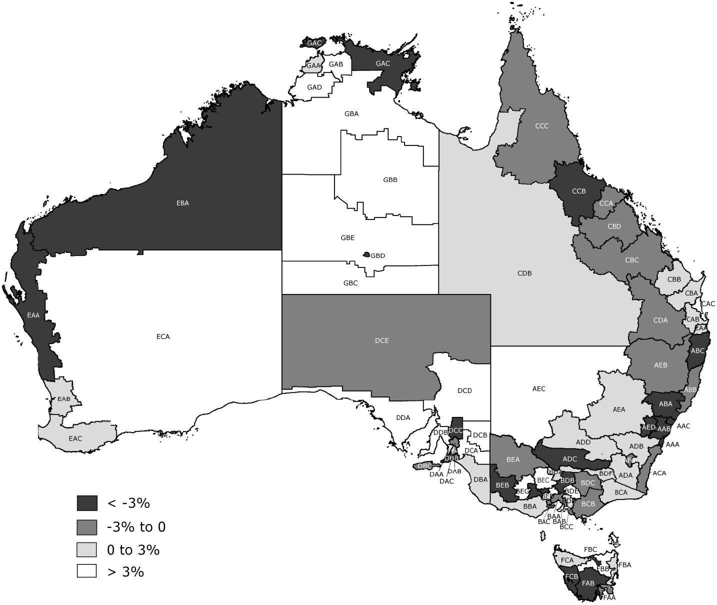
- Computational difficulties in big hierarchies due to size of the $\boldsymbol{S}$ matrix and singular behavior of $(\boldsymbol{S}'\Lambda\boldsymbol{S})$.
- Loss of information in ignoring covariance matrix in computing point forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

Challenges

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

- Computational difficulties in big hierarchies due to size of the $\boldsymbol{S}$ matrix and singular behavior of $(\boldsymbol{S}'\Lambda\boldsymbol{S})$.
- Loss of information in ignoring covariance matrix in computing point forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

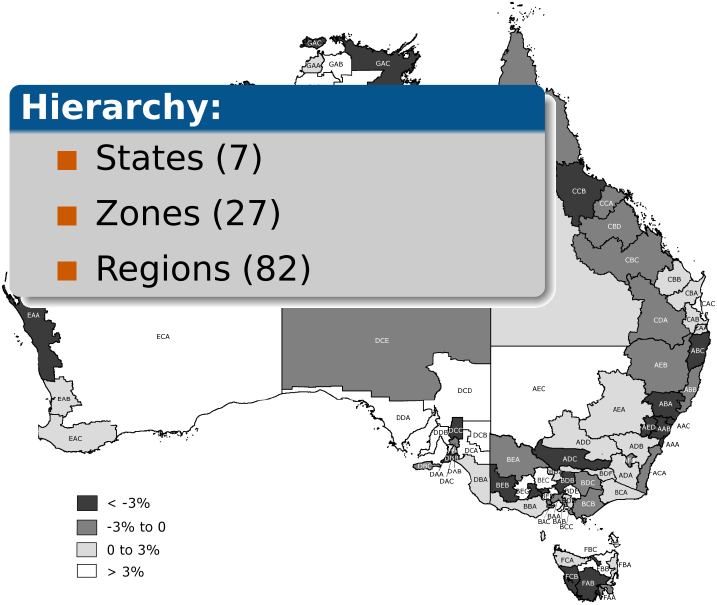
Australian tourism



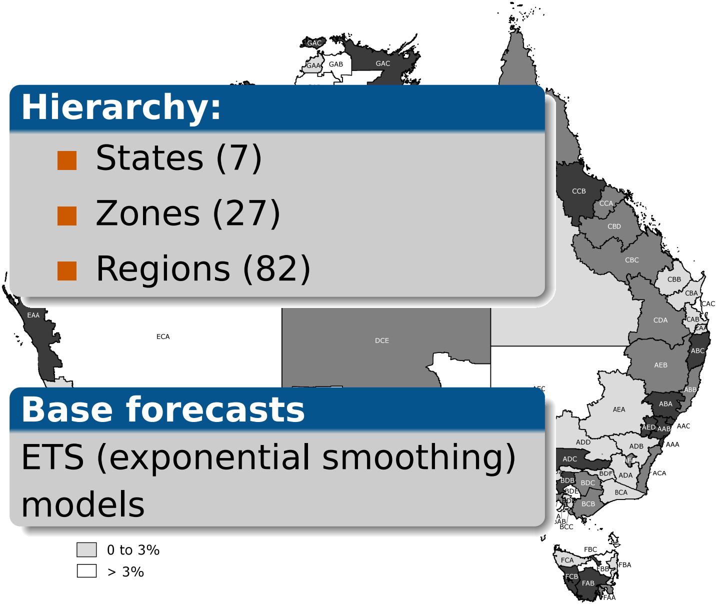
Australian tourism

Hierarchy:

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



Australian tourism



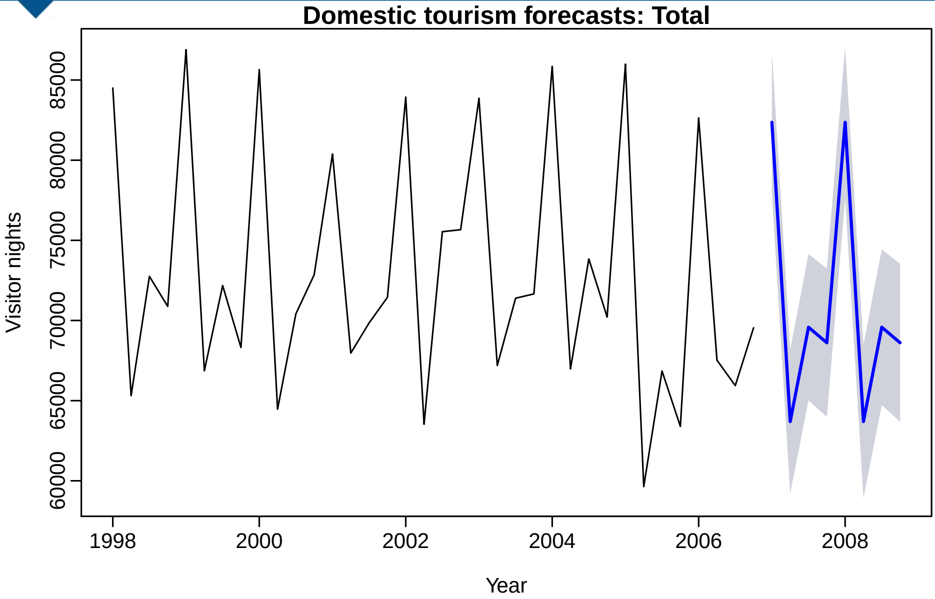
Hierarchy:

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

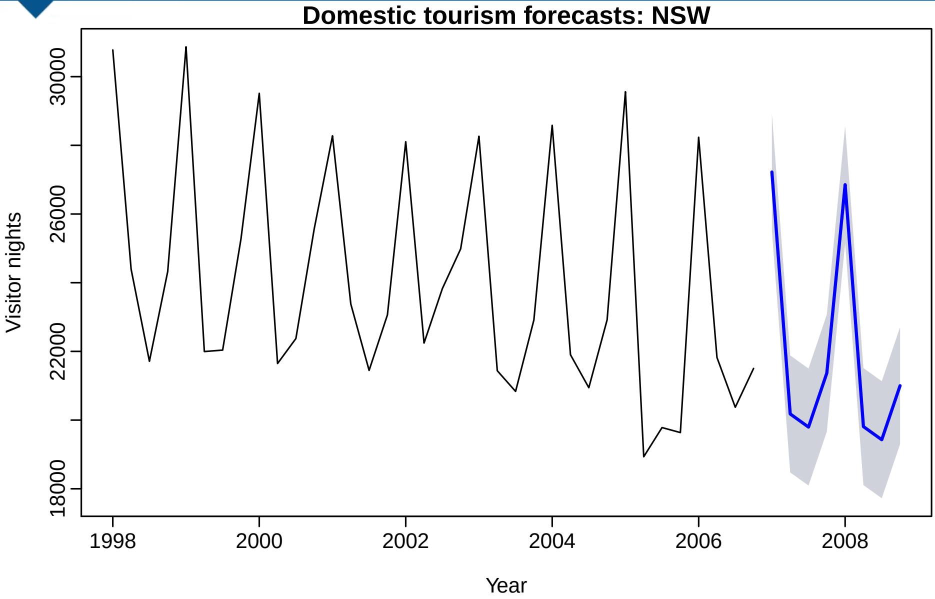
Base forecasts

ETS (exponential smoothing) models

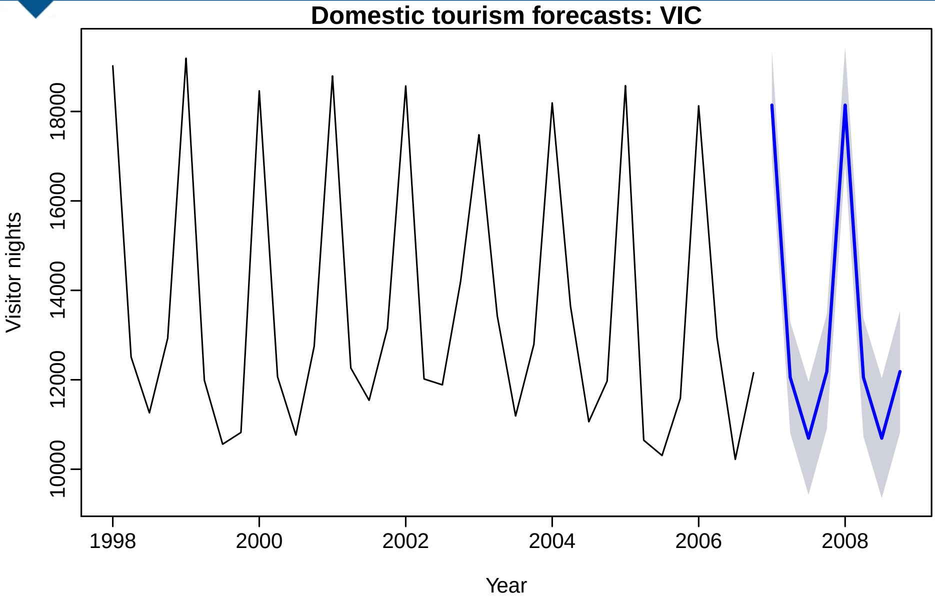
Base forecasts



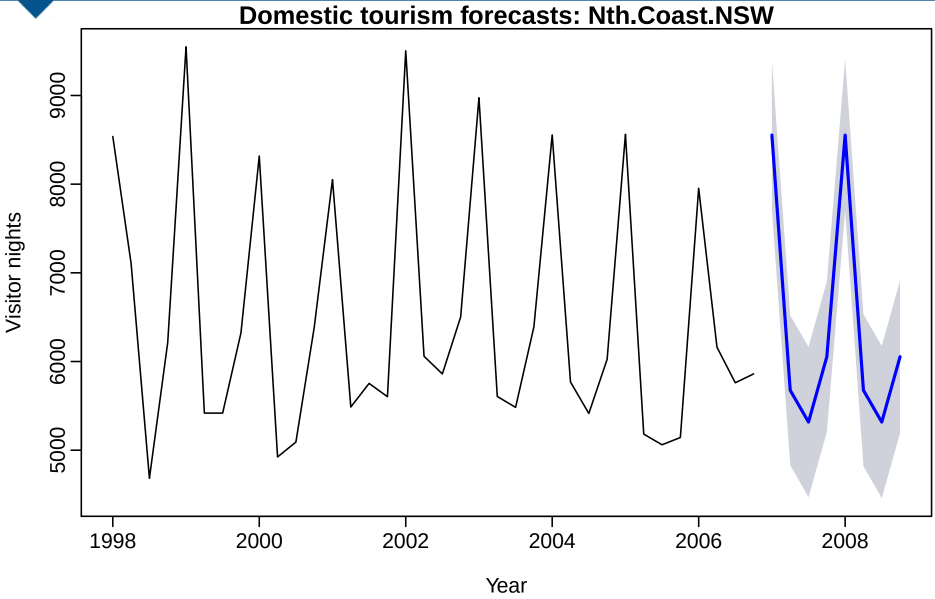
Base forecasts



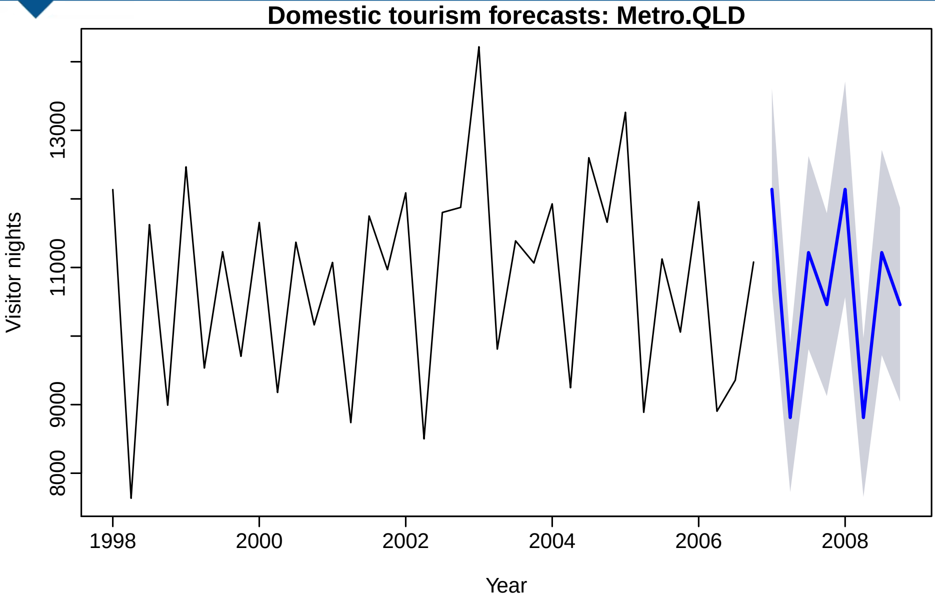
Base forecasts



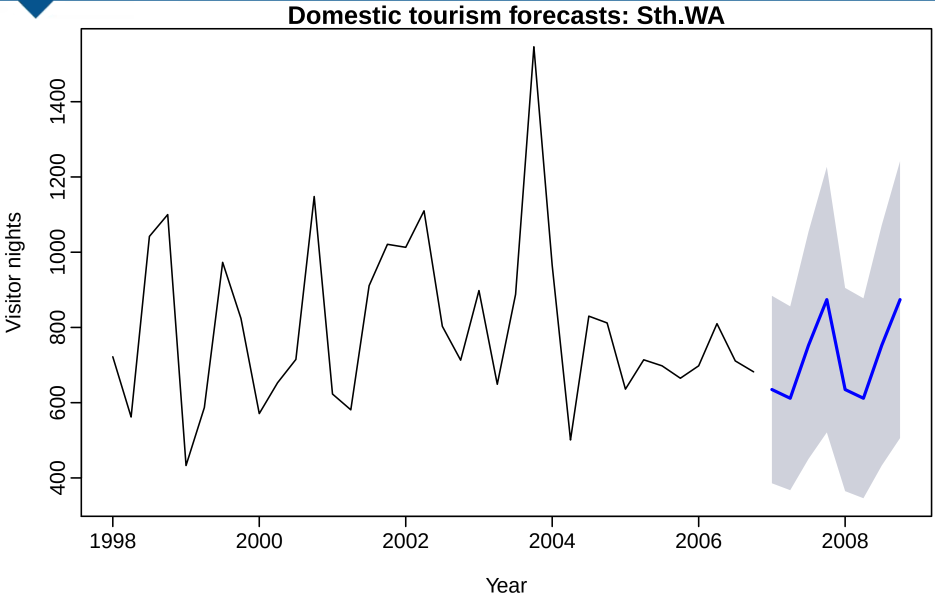
Base forecasts



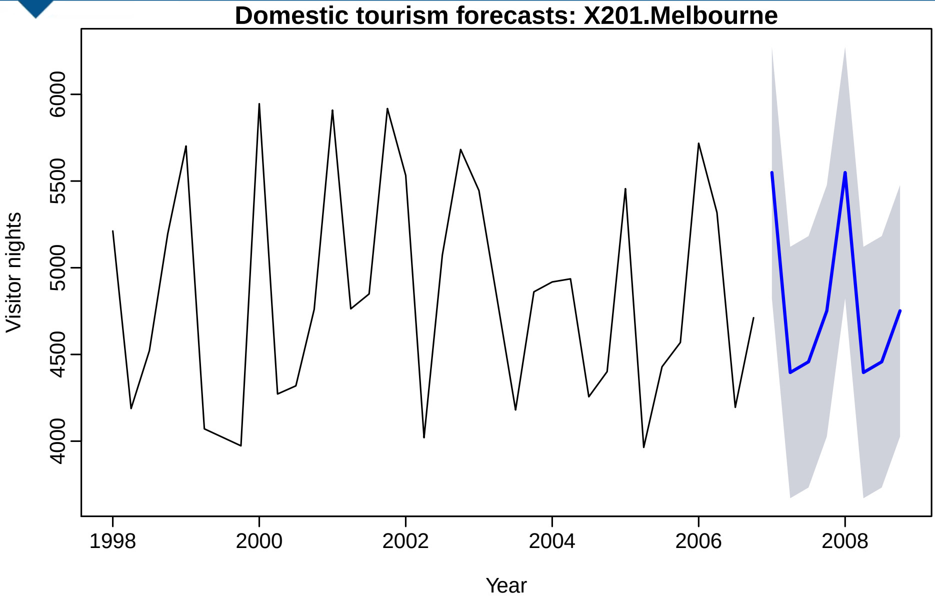
Base forecasts



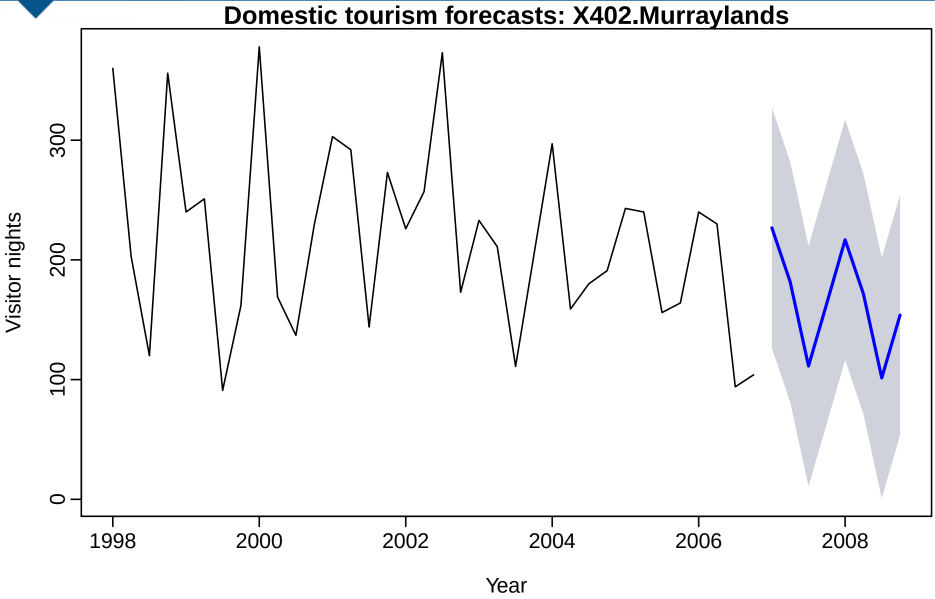
Base forecasts



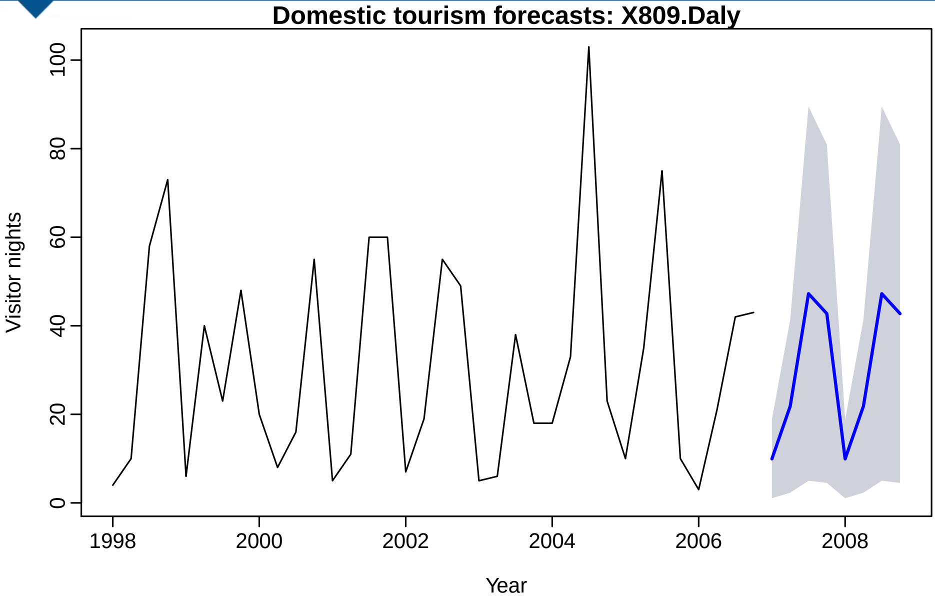
Base forecasts



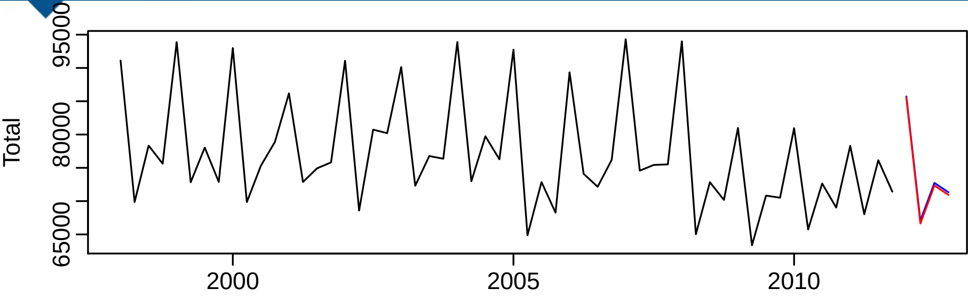
Base forecasts



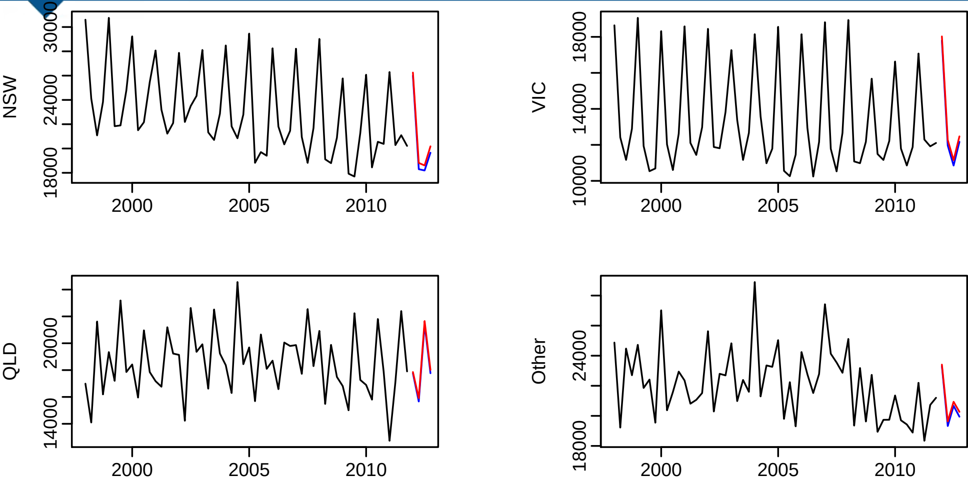
Base forecasts



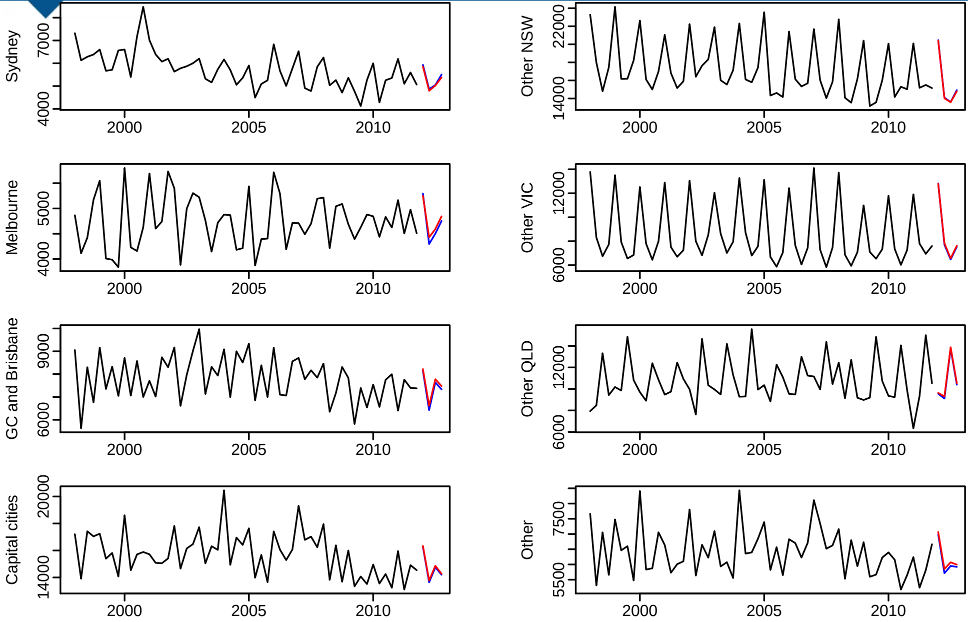
Reconciled forecasts



Reconciled forecasts



Reconciled forecasts



Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-step-ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.

Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-step-ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.

Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-step-ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.

Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-step-ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.

Hierarchy: states, zones, regions

MAPE	$h=1$	$h=2$	$h=4$	$h=6$	$h=8$	Average
Top Level: Australia						
Bottom-up	3.79	3.58	4.01	4.55	4.24	4.06
OLS	3.83	3.66	**3.88**	**4.19**	4.25	**3.94**
WLS	**3.68**	**3.56**	3.97	4.57	4.25	4.04
Level: States						
Bottom-up	10.70	10.52	10.85	11.46	11.27	11.03
OLS	11.07	10.58	11.13	11.62	12.21	11.35
WLS	**10.44**	**10.17**	**10.47**	**10.97**	**10.98**	**10.67**
Level: Zones						
Bottom-up	14.99	14.97	14.98	15.69	15.65	15.32
OLS	15.16	15.06	15.27	15.74	16.15	15.48
WLS	**14.63**	**14.62**	**14.68**	**15.17**	**15.25**	**14.94**
Bottom Level: Regions						
Bottom-up	33.12	32.54	32.26	33.74	33.96	33.18
OLS	35.89	33.86	34.26	36.06	37.49	35.43
WLS	**31.68**	**31.22**	**31.08**	**32.41**	**32.77**	**31.89**

hts package for R

hts: Hierarchical and grouped time series

Methods for analysing and forecasting hierarchical and grouped time series

Version:	4.5
Depends:	forecast ($\geq$ 5.0), SparseM
Imports:	parallel, utils
Published:	2014-12-09
Author:	Rob J Hyndman, Earo Wang and Alan Lee
Maintainer:	Rob J Hyndman $<$Rob.Hyndman at monash.edu$>$
BugReports:	`https://github.com/robjhyndman/hts/issues`
License:	GPL ($\geq$ 2)

Outline

Forecasts about forecasting

1. Automatic algorithms will become more general — handling a wide variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for multivariate time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be developed.

Forecasts about forecasting

1. Automatic algorithms will become more general — handling a wide variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for multivariate time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be developed.

Forecasts about forecasting

1. Automatic algorithms will become more general — handling a wide variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for multivariate time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be developed.

Forecasts about forecasting

1. Automatic algorithms will become more general — handling a wide variety of time series.
2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.
3. Automatic forecasting algorithms for multivariate time series will be developed.
4. Automatic forecasting algorithms that include covariate information will be developed.

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.