MONASH University

MONASH BUSINESS SCHOOL

2017 Beijing Workshop on Forecasting

Automatic Forecasting Algorithms

Rob J Hyndman

robjhyndman.com/beijing2017

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

AUSTRALIA

Australian Government

Department of Health and Ageing

Motivation

FOXTEL digital

CAT

hp

Incitec Pivot

Australian Government

Department of Health and Ageing

Motivation

- Common in business to have over 1000 products that need forecasting at least monthly.
- 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

- Common in business to have over 1000 products that need forecasting at least monthly.
- 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.

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)

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

There are also multiplicative trend methods (not recommended).

Methods V Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

- Generate same point forecasts but can also generate forecast intervals.
- A stochastic (or random) data generating process that can generate an entire forecast distribution.
- Allow for “proper” model selection.
- ETS(Error,Trend,Seasonal):
  - Error = $\{A,M\}$
  - Trend = $\{N,A,A_d\}$
  - Seasonal = $\{N,A,M\}$.

Methods V Models

Exponential smoothing methods

- Algorithms that return point forecasts.

Innovations state space models

- Generate same point forecasts but can also generate forecast intervals.
- A stochastic (or random) data generating process that can generate an entire forecast distribution.
- Allow for “proper” model selection.
- ETS(Error,Trend,Seasonal):
  - Error = $\{A,M\}$
  - Trend = $\{N,A,A_d\}$
  - Seasonal = $\{N,A,M\}$.

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

Exponential smoothing models

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

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

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

Examples:

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

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

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

Exponential smoothing models

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

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

Error **Trend** **Seasonal**

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 models

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

- There are 9 separate exp. smoothing methods.
- Each can have an additive or multiplicative error, giving 18 separate models.
- Only 15 models are numerically stable.
- Additive and multiplicative error models give same point forecasts but different prediction intervals.
- All models can be written in innovations state space form.

Exponential smoothing models

		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

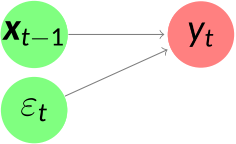
- There are 9 separate exp. smoothing methods.
- Each can have an additive or multiplicative error, giving 18 separate models.
- Only 15 models are numerically stable.
- Additive and multiplicative error models give same point forecasts but different prediction intervals.
- All models can be written in innovations state space form.

Exponential smoothing models

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

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

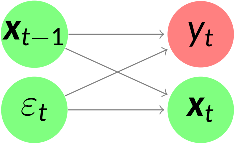
ETS state space models



State space model

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

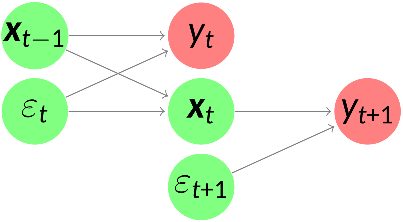
ETS state space models



State space model

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

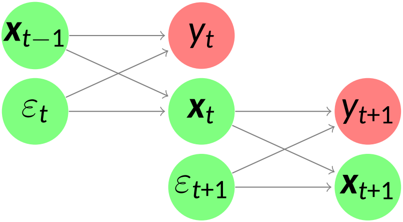
ETS state space models



State space model

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

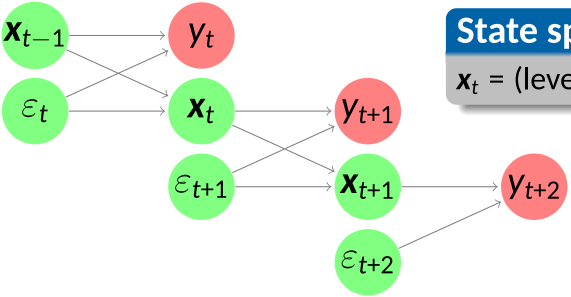
ETS state space models



State space model

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

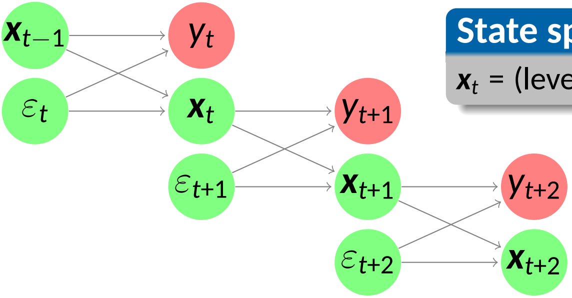
ETS state space models



State space model

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

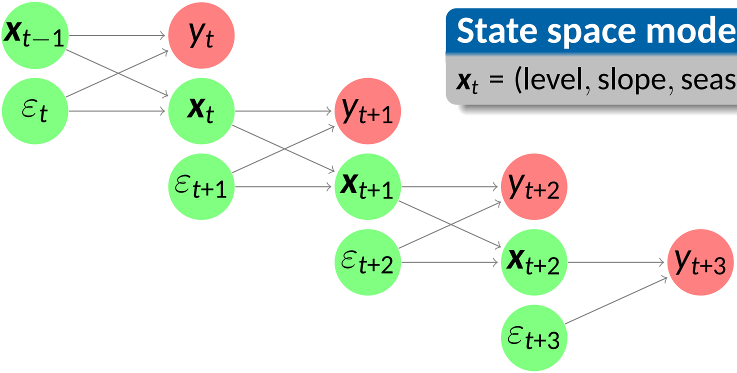
ETS state space models



State space model

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

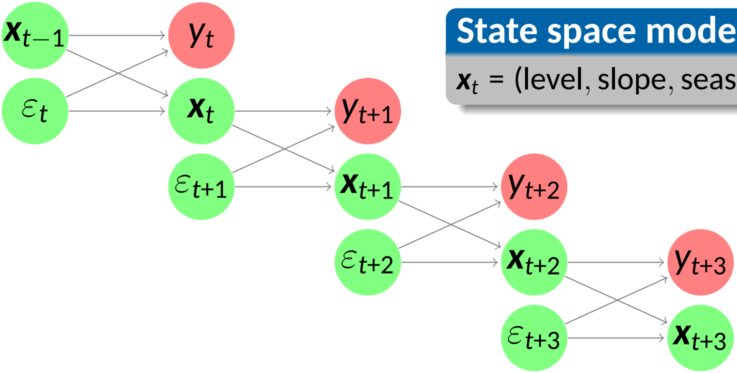
ETS state space models



State space model

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

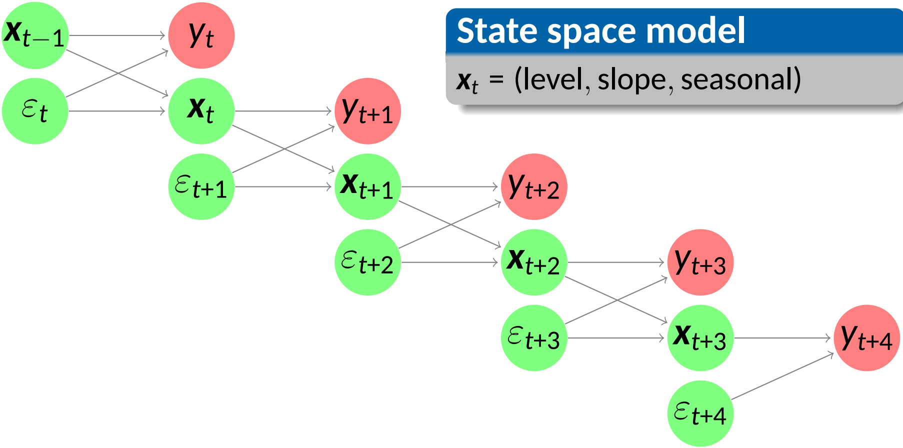
ETS state space models



State space model

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

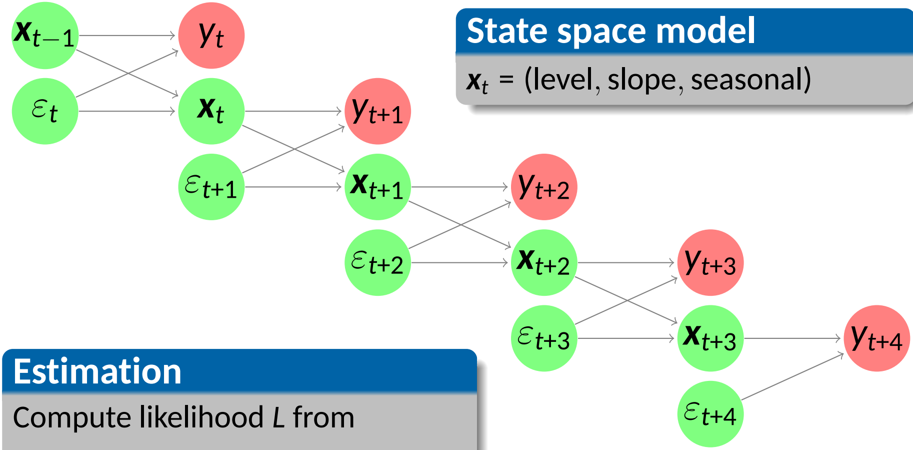
ETS state space models



State space model

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

ETS state space models



State space model

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

Estimation

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

ETS state space models

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

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

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

Estimation

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

- Estimate parameters $\boldsymbol{\theta} = (\alpha, \beta, \gamma, \phi)$ and initial states $\boldsymbol{x}_0 = (\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1})$ by minimizing L^*.

Automatic forecasting

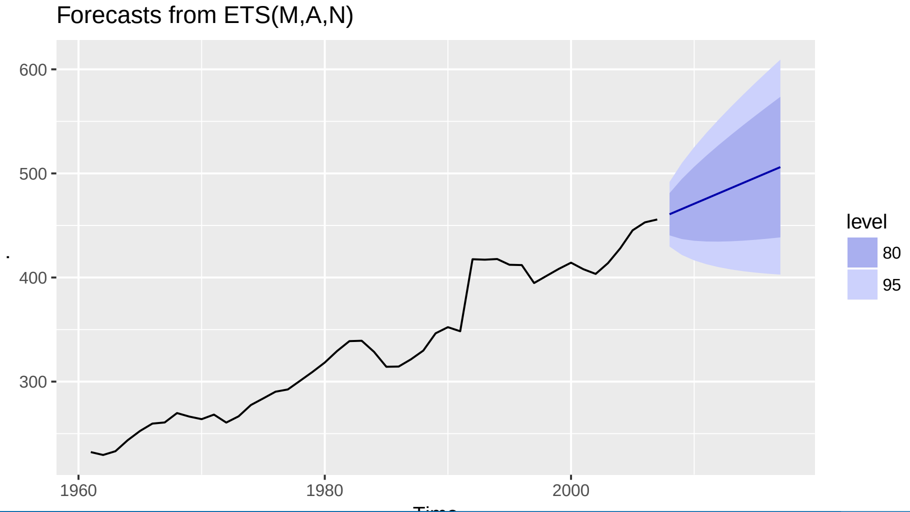
From Hyndman et al. (IJF, 2002):

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

Method performed very well in M3 competition.

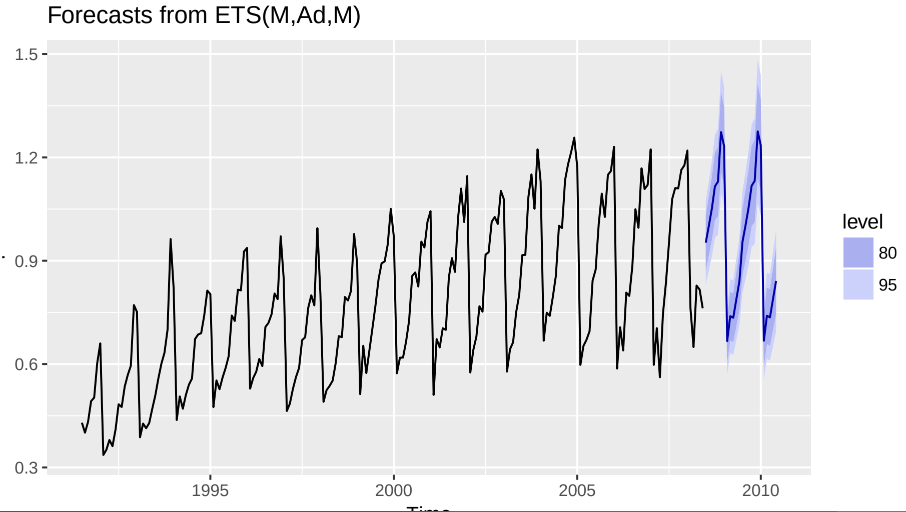
Example: Asian livestock

```
livestock %>% ets() %>% forecast() %>% autoplot
```

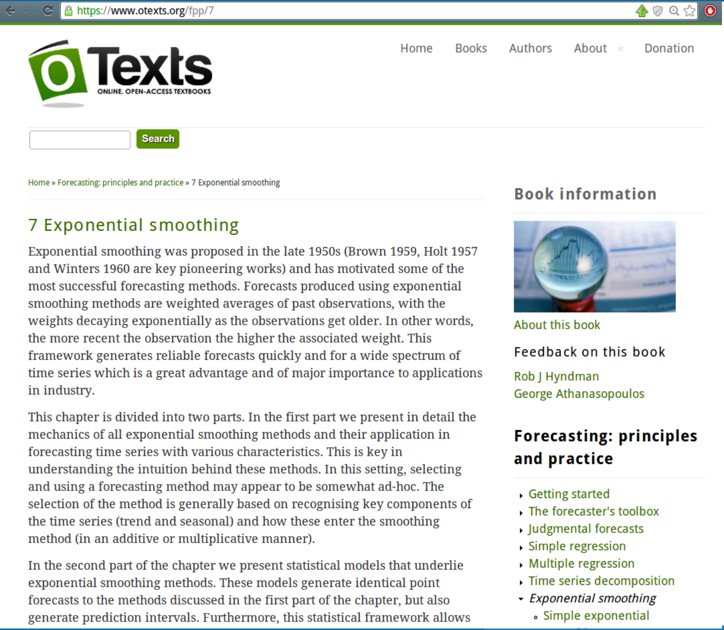


Example: drug sales

```
h02 %>% ets() %>% forecast() %>% autoplot()
```



ETS



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

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 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 selection of the method is generally based on recognising key components of the time series (trend and seasonal) and how these enter the smoothing method (in an additive or multiplicative manner).

In the second part of the chapter we present statistical models that underlie exponential smoothing methods. These models generate identical point forecasts to the methods discussed in the first part of the chapter, but also generate prediction intervals. Furthermore, this statistical framework allows

Book information

About this book

Feedback on this book

Rob J Hyndman
George Athanasopoulos

Forecasting: principles and practice

ETS

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 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 di... mechanics of all... forecasting time... understanding the ... and using a forecasting method may appear to be somewhat ad-hoc. The selection of the method is generally based on recognising key components of the time series (trend and seasonal) and how these enter the smoothing method (in an additive or multiplicative manner).

In the second part of the chapter we present statistical models that underlie exponential smoothing methods. These models generate identical point forecasts to the methods discussed in the first part of the chapter, but also generate prediction intervals. Furthermore, this statistical framework allows

www.OTexts.org/fpp2

Book information

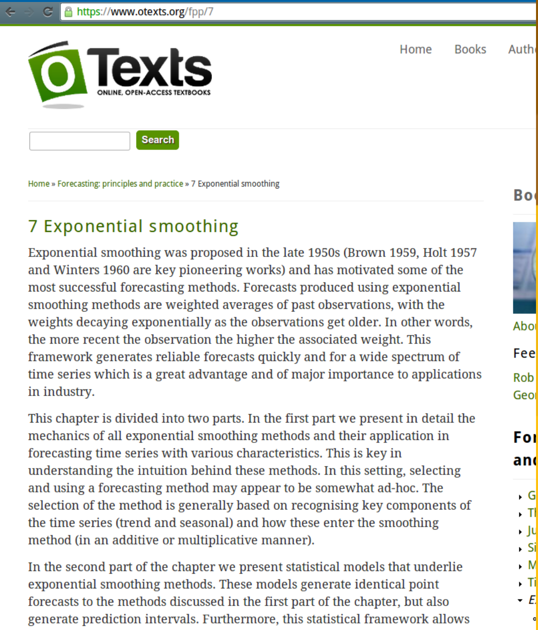
About this book

Feedback on this book

Rob J Hyndman
George Athanasopoulos

...ciples

ETS


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

OTexts ONLINE, OPEN-ACCESS TEXTBOOKS

Home Books

Search

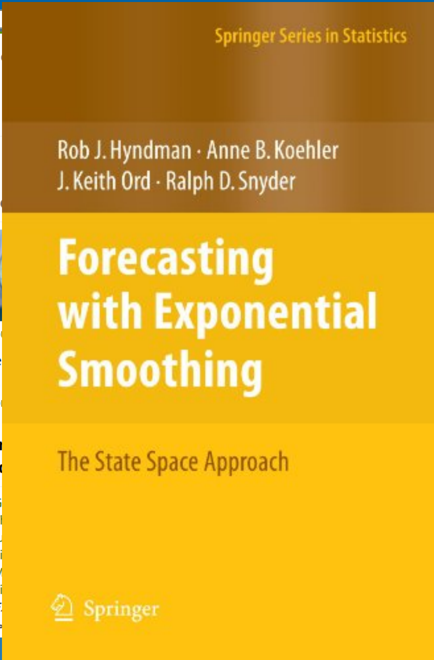
Home » Forecasting: principles and practice » 7 Exponential smoothing

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 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.

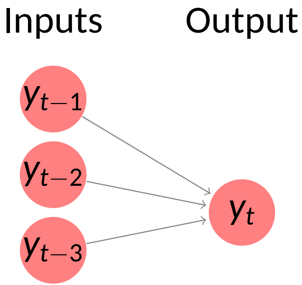
This chapter is divided into two parts. In the first part we present in detail the mechanics of all exponential smoothing methods and their application in forecasting time series with various characteristics. This is key in understanding the intuition behind these methods. In this setting, selecting and using a forecasting method may appear to be somewhat ad-hoc. The selection of the method is generally based on recognising key components of the time series (trend and seasonal) and how these enter the smoothing method (in an additive or multiplicative manner).

In the second part of the chapter we present statistical models that underlie exponential smoothing methods. These models generate identical point forecasts to the methods discussed in the first part of the chapter, but also generate prediction intervals. Furthermore, this statistical framework allows

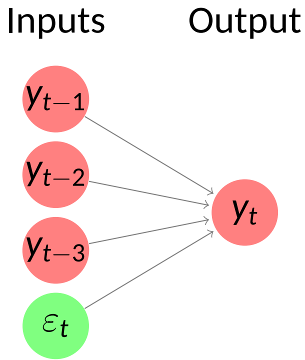


Outline

ARIMA models

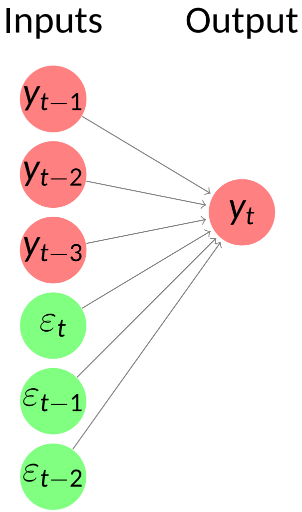


ARIMA models



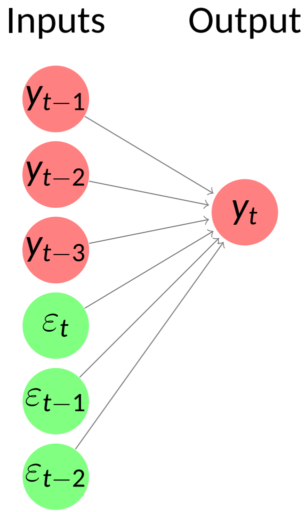
Autoregression (AR) model

ARIMA models



Autoregression moving average (ARMA) model

ARIMA models

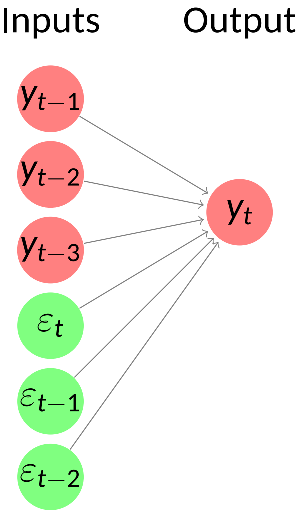


Autoregression moving average (ARMA) model

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

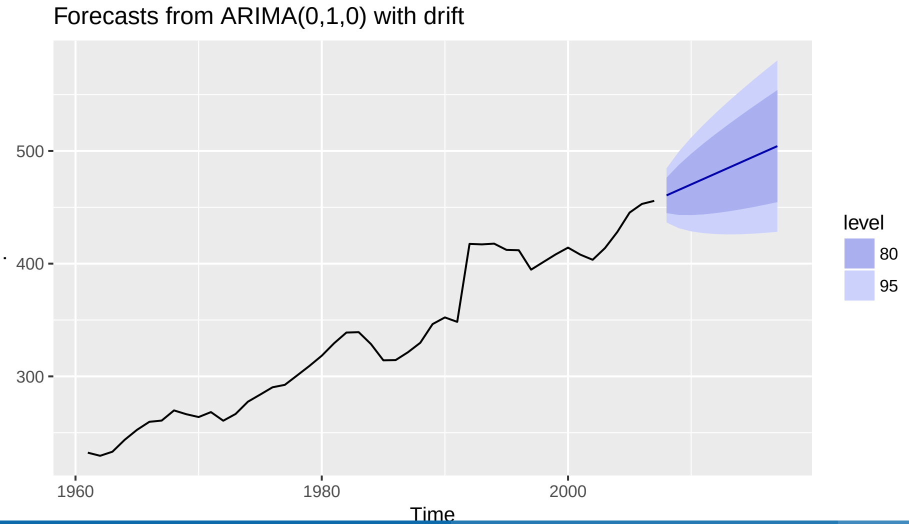
Autoregression moving average (ARMA) model applied to differences.

Estimation

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

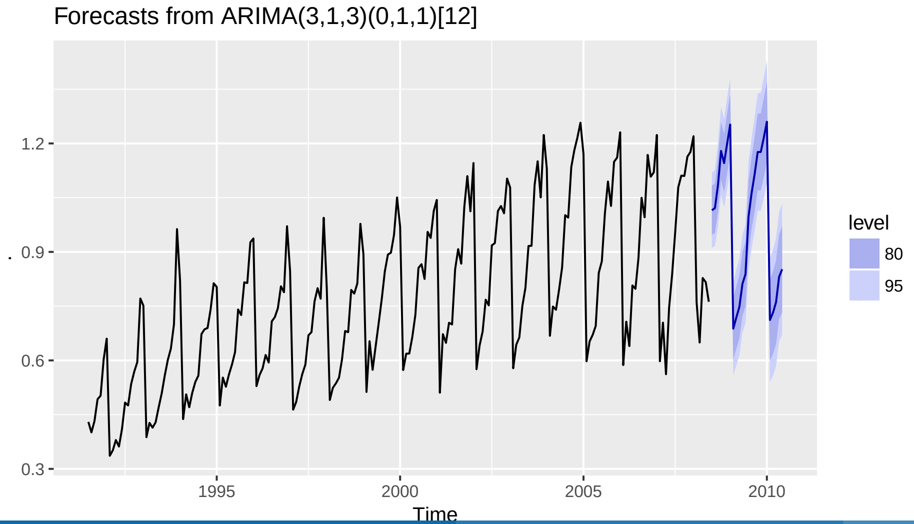
Auto ARIMA

```r
livestock %>% auto.arima() %>% forecast() %>% autopl
```



Auto ARIMA

```r
h02 %>% auto.arima() %>% forecast() %>% autoplot()
```



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.

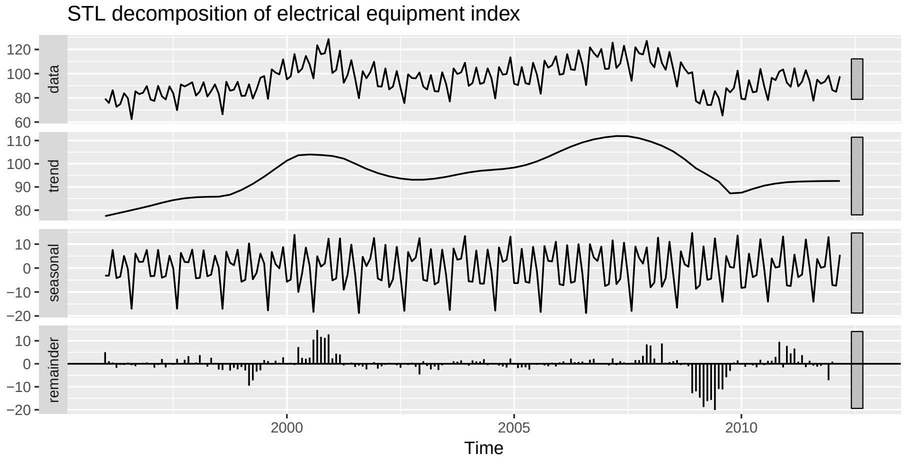
Outline

STL decomposition

- STL: “Seasonal and Trend decomposition using Loess”,
- Very versatile and robust.
- Unlike X-12-ARIMA, STL will handle any type of seasonality.
- Seasonal component allowed to change over time, and rate of change controlled by user.
- Smoothness of trend-cycle also controlled by user.
- Robust to outliers
- Not trading day or calendar adjustments.
- Only additive.
- Take logs to get multiplicative decomposition.
- Use Box-Cox transformations to get other decompositions.

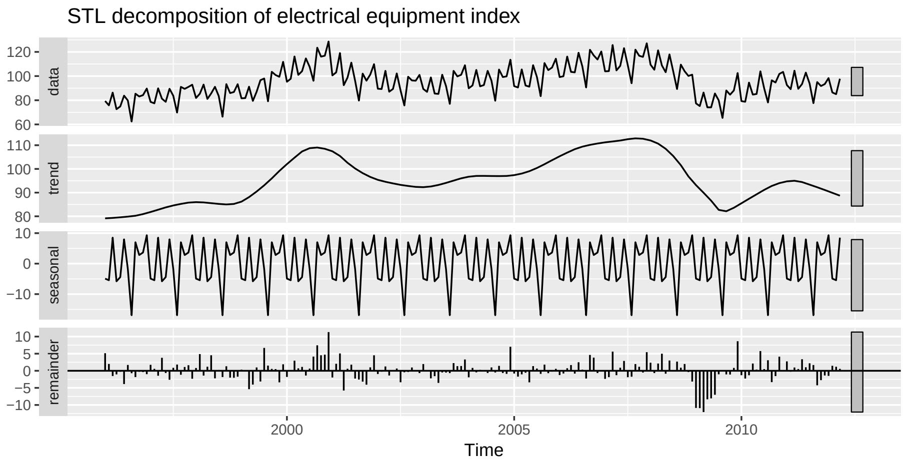
STL decomposition

```
fit <- stl(elecequip, s.window=5, robust=TRUE)
autoplot(fit) +
  ggtitle("STL decomposition of electrical equipment inde
```



STL decomposition

```
fit <- stl(elecequip, s.window="periodic", robust=TRUE)
autoplot(fit) +
  ggtitle("STL decomposition of electrical equipment inde
```

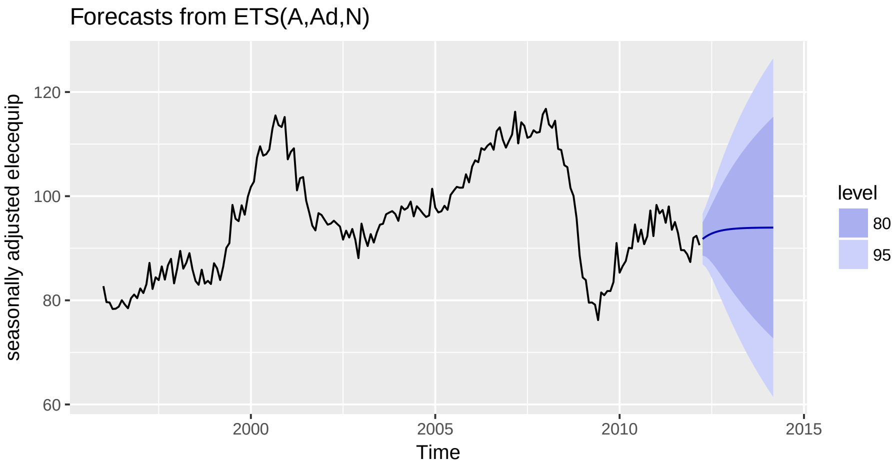


Forecasting and decomposition

- Forecast seasonal component using seasonal naive method.
- Forecast seasonally adjusted data using non-seasonal time series method. E.g., ETS or ARIMA.
- Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.

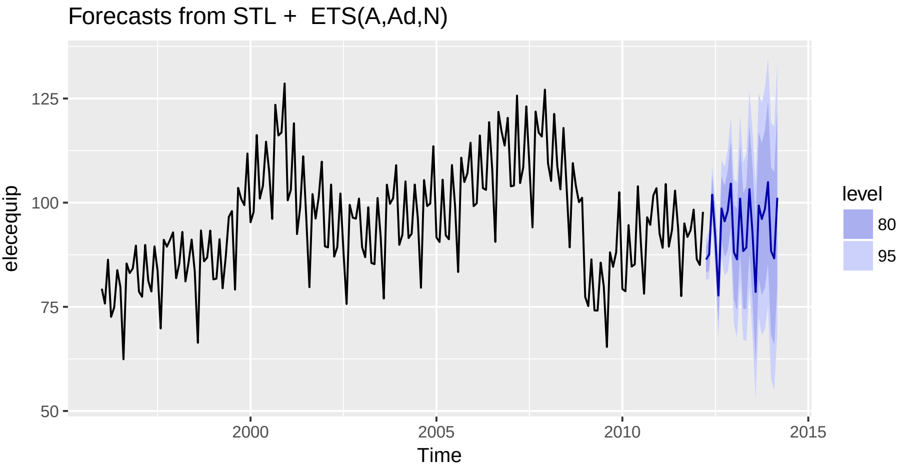
Electrical equipment

```
fit <- stl(elecequip, s.window=7)
fit %>% seasadj() %>% ets() %>% forecast() %>% autoplot(
```



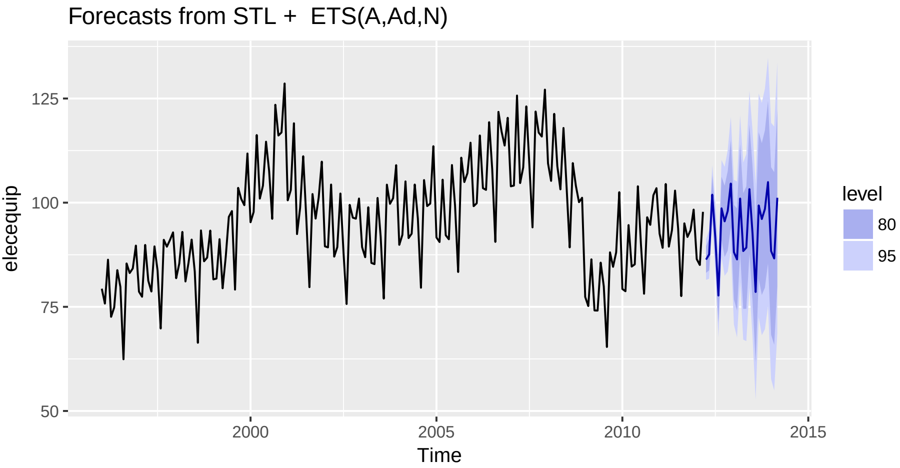
Electrical equipment

```
fit %>% forecast() %>%
  autoplot()
```



Forecasting and decomposition

```
elecequip %>% stlf() %>%
  autoplot() + ylab('elecequip')
```

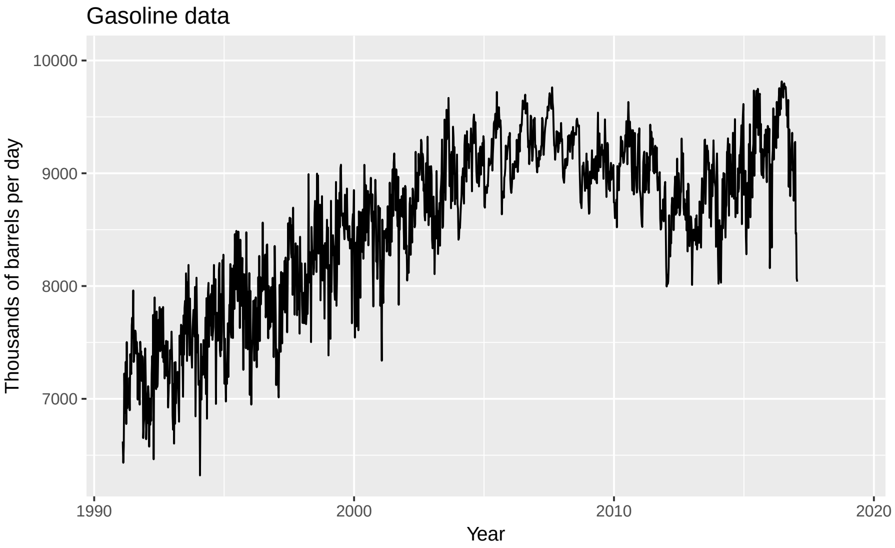


Decomposition and prediction intervals

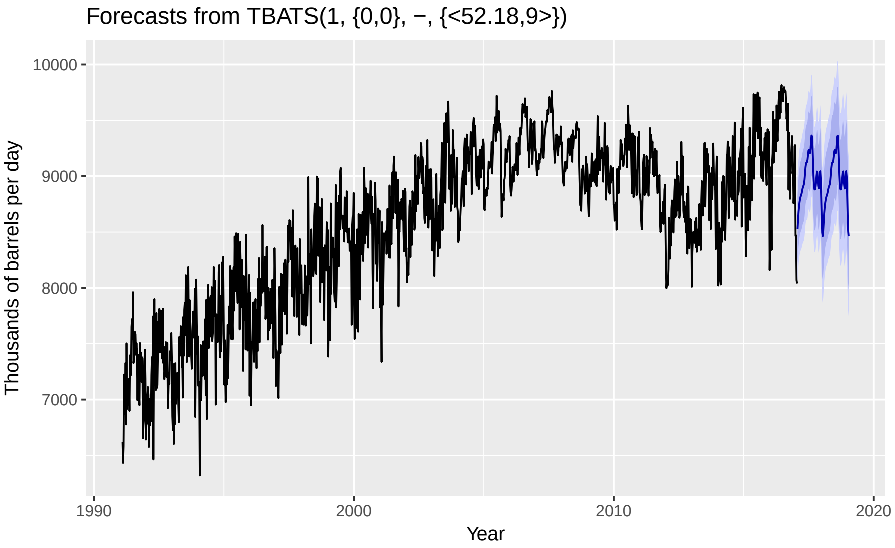
- It is common to take the prediction intervals from the seasonally adjusted forecasts and modify them with the seasonal component.
- This ignores the uncertainty in the seasonal component estimate.
- It also ignores the uncertainty in the future seasonal pattern.

Outline

Complex seasonality

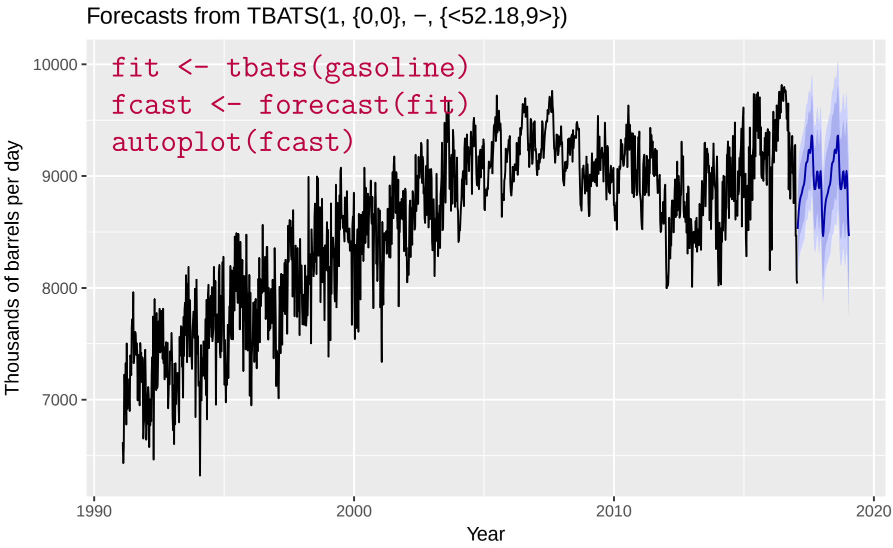


Complex seasonality

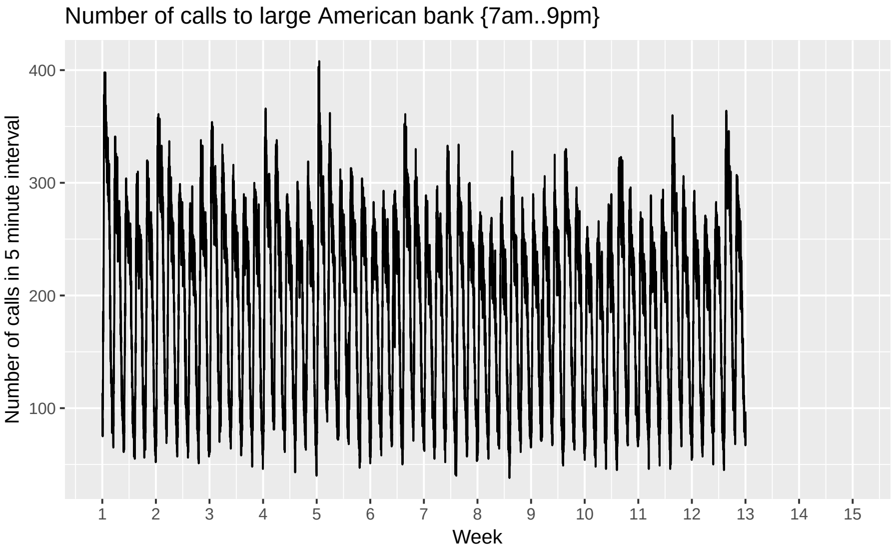


Complex seasonality

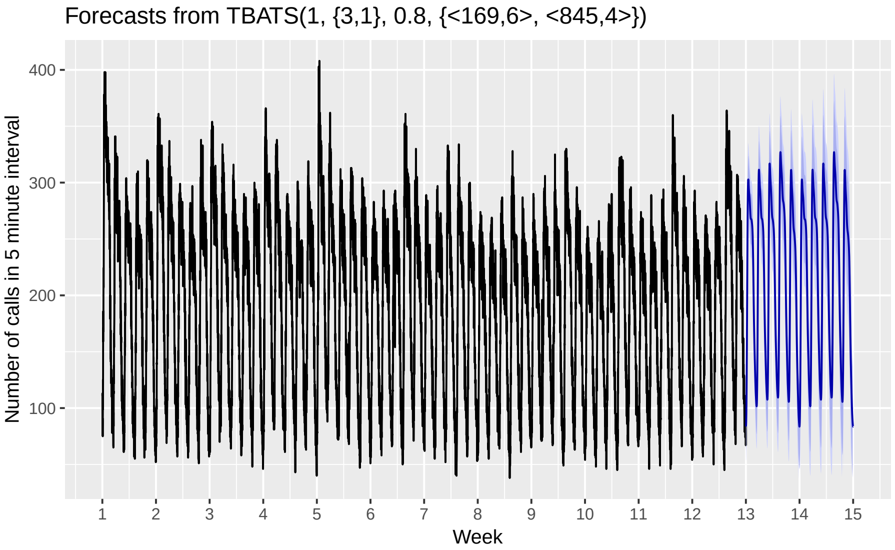
```
fit <- tbats(gasoline)
fcast <- forecast(fit)
autoplot(fcast)
```



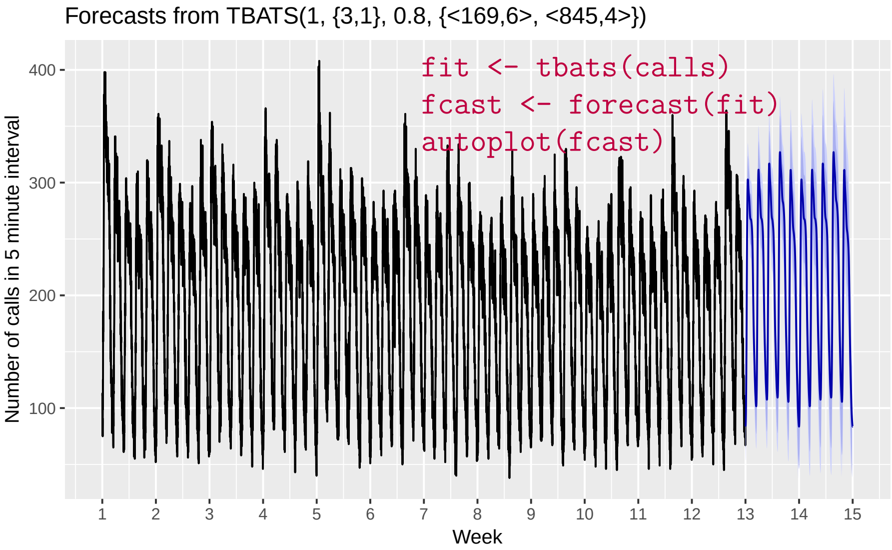
Complex seasonality



Complex seasonality



Complex seasonality



TBATS model

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

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

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

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

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

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

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

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

TBATS model

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

Box-Cox transformation

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

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

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

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

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

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

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

TBATS model

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

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

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

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

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

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

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

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

Box-Cox transformation

M seasonal periods

TBATS model

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

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

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

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

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

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

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

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

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

Box-Cox transformation

M seasonal periods

global and local trend

TBATS model

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

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

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

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

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

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

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

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

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

Box-Cox transformation

M seasonal periods

global and local trend

ARMA error

TBATS model

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

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

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

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

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

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

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

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos\lambda_j + s_{j,t-1} \sin\lambda_j + \gamma_1 d_t$$

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

Box-Cox transformation

M seasonal periods

global and local trend

ARMA error

Fourier-like seasonal terms

TBATS model

y_t = observation at time t

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

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

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

$$b_t = (1 -$$

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

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

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

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

TBATS
- Trigonometric
- Box-Cox
- ARMA
- Trend
- Seasonal

Box-Cox transformation

M seasonal periods

global and local trend

ARMA error

Fourier-like seasonal terms

TBATS model

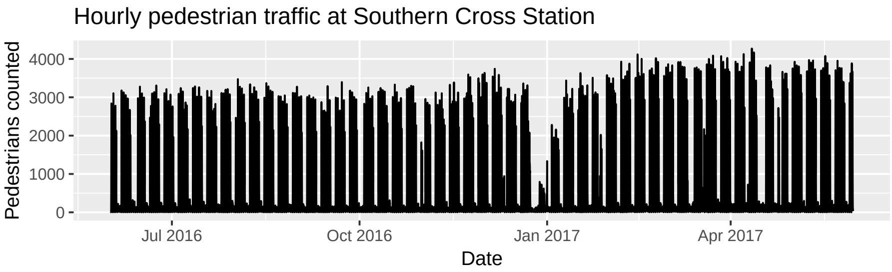
TBATS

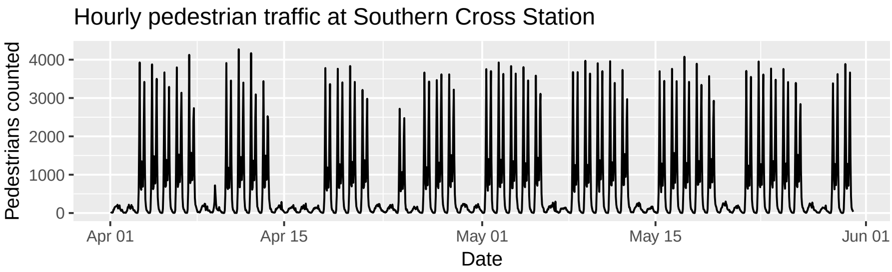
Trigonometric terms for seasonality
Box-Cox transformations for heterogeneity
ARMA errors for short-term dynamics
Trend (possibly damped)
Seasonal (including multiple and non-integer periods)

- Handles non-integer seasonality, multiple seasonal periods.
- Entirely automated
- Prediction intervals often too wide
- Very slow on long series

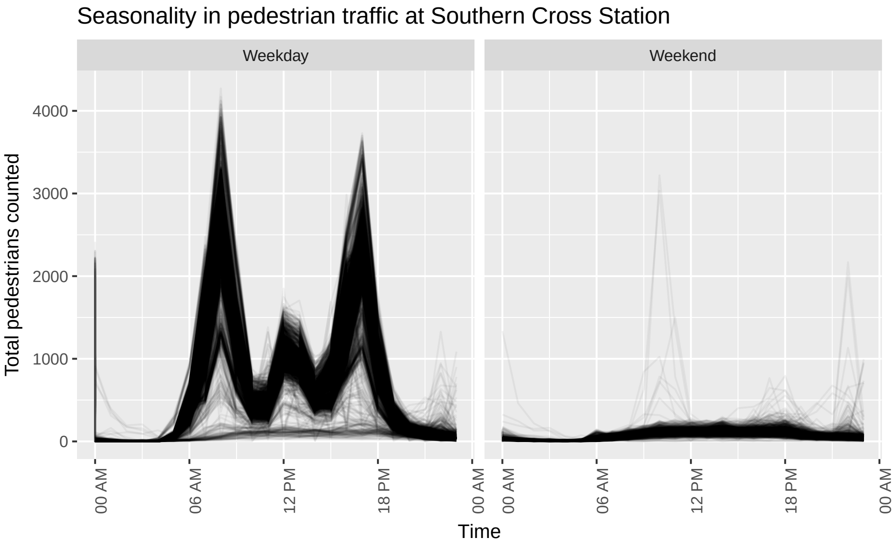
Outline

Pedestrian counts





Pedestrian counts



Switching Structure

FASSTER extends current state-space approaches by switching between states.

Dynamic linear model

$$y_t = F_t\theta_t + v_t, \qquad v_t \sim \mathcal{N}(0, V)$$
$$\theta_t = G\theta_{t-1} + w_t, \qquad w_t \sim \mathcal{N}(0, W)$$

Switch between two groups:

$$y_t = \mathbf{I}_{t\in G_1}F_t\theta_t^{(1)} + \mathbf{I}_{t\in G_2}F_t\theta_t^{(2)} + v_t \qquad v_t \sim \mathcal{N}(0, V)$$

Groups G_1 and G_2 define the switching rule (say weekdays and weekends).

Switching Structure

FASSTER extends current state-space approaches by switching between states.

Dynamic linear model

$$\begin{aligned} y_t &= F_t\theta_t + v_t, & v_t &\sim \mathcal{N}(0, V) \\ \theta_t &= G\theta_{t-1} + w_t, & w_t &\sim \mathcal{N}(0, W) \end{aligned}$$

Switch between two groups:

$$y_t = \mathbf{I}_{t\in G_1} F_t\theta_t^{(1)} + \mathbf{I}_{t\in G_2} F_t\theta_t^{(2)} + v_t \qquad v_t \sim \mathcal{N}(0, V)$$

Groups G_1 and G_2 define the switching rule (say weekdays and weekends).

Switching Structure

FASSTER extends current state-space approaches by switching between states.

Dynamic linear model

$$\begin{aligned} y_t &= F_t\theta_t + v_t, & v_t &\sim \mathcal{N}(0, V) \\ \theta_t &= G\theta_{t-1} + w_t, & w_t &\sim \mathcal{N}(0, W) \end{aligned}$$

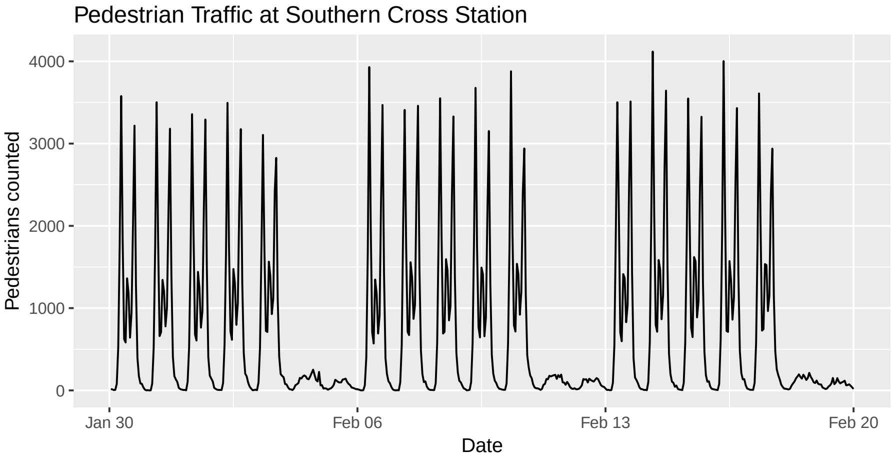
Switch between two groups:

$$y_t = \mathbf{I}_{t\in G_1}F_t\theta_t^{(1)} + \mathbf{I}_{t\in G_2}F_t\theta_t^{(2)} + v_t \qquad v_t \sim \mathcal{N}(0, V)$$

Groups G_1 and G_2 define the switching rule (say weekdays and weekends).

Application to pedestrian traffic

This series contains a switching daily pattern over weekdays and weekends. Each group can be modelled using level and hourly seasonal states.



Application to pedestrian traffic

An appropriate model can be constructed using switching states:

$$y_t = \mathbf{I}_{t \in Weekday} F_t \theta_t^{(Weekday)} + \mathbf{I}_{t \in Weekend} F_t \theta_t^{(Weekend)} + v_t$$

$$\text{where} \quad F_t \theta_t^{(i)} = \ell_t^{(i)} + f_t^{(i)} \quad \text{and} \quad v_t \sim \mathcal{N}(0, V)$$

- ℓ_t is a level component
- f_t is a seasonal component based on Fourier terms

FASSTER allows flexible use of:

- seasonal factors
- fourier seasonal terms
- polynomial trends
- BoxCox transformations
- exogenous regressors
- ARMA processes
- state switching

General measurement equation

$$y_t = F_t^{(0)}\theta_t^{(0)} + \sum_{j=1}^{k} \mathbb{I}_{t\in G_j} F_t^{(j)}\theta_t^{(j)} + v_t, \qquad v_t \sim \mathcal{N}(0, V)$$

where k is the number of switching combinations.

FASSTER allows flexible use of:

- seasonal factors
- fourier seasonal terms
- polynomial trends
- BoxCox transformations
- exogenous regressors
- ARMA processes
- state switching

General measurement equation

$$y_t = F_t^{(0)}\theta_t^{(0)} + \sum_{j=1}^{k} \mathbf{I}_{t\in G_j} F_t^{(j)}\theta_t^{(j)} + v_t, \qquad v_t \sim \mathcal{N}(0, V)$$

where k is the number of switching combinations.

FASSTER and Dynamic Linear Models

A FASSTER model can be represented as a time-varying DLM.

$$\begin{aligned} y_t &= F_t\theta_t + v_t, & v_t &\sim \mathcal{N}(0, V) \\ \theta_t &= G\theta_{t-1} + w_t, & w_t &\sim \mathcal{N}(0, W) \end{aligned}$$
where $\theta_0 \sim \mathcal{N}(m_0, C_0)$

- There are too many parameters to easily estimate.
- E.g., pedestrian model has 48 states:
  2 groups $\times$ (1 level + 23 fourier states).
- We use a "heuristic estimation" approach involving only two passes through the data.

FASSTER and Dynamic Linear Models

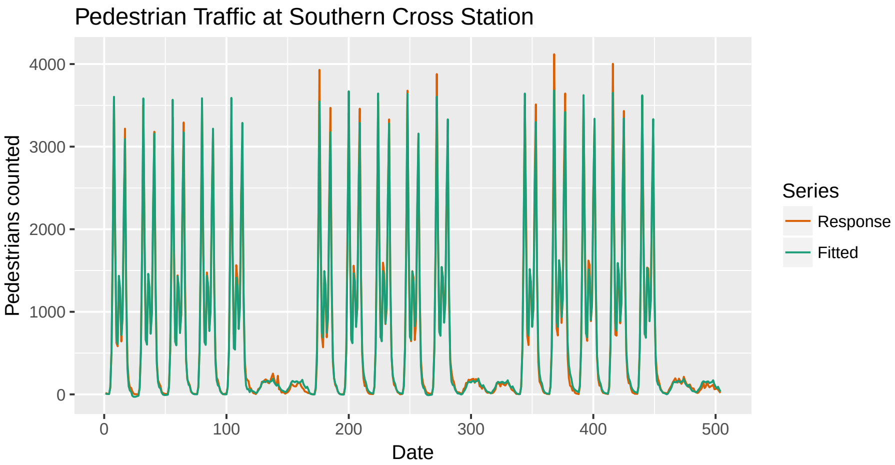
A FASSTER model can be represented as a time-varying DLM.

$$\begin{aligned} y_t &= F_t\theta_t + v_t, & v_t &\sim \mathcal{N}(0, V) \\ \theta_t &= G\theta_{t-1} + w_t, & w_t &\sim \mathcal{N}(0, W) \end{aligned}$$
$$\text{where } \theta_0 \sim \mathcal{N}(m_0, C_0)$$

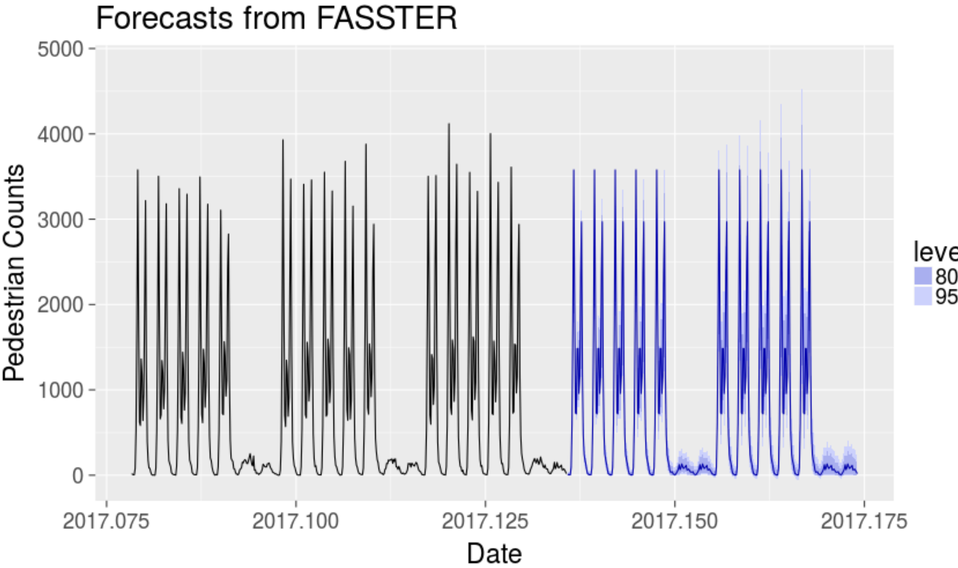
- There are too many parameters to easily estimate.
- E.g., pedestrian model has 48 states:
  2 groups $\times$ (1 level + 23 fourier states).
- We use a "heuristic estimation" approach involving only two passes through the data.

Usage (Pedestrian Counts)

```
SthCross_fasster_fit <- SthCross_Ex %>%
  fasster(Hourly_Counts ~ DayType %S% (poly(1) + trig(24)))
```

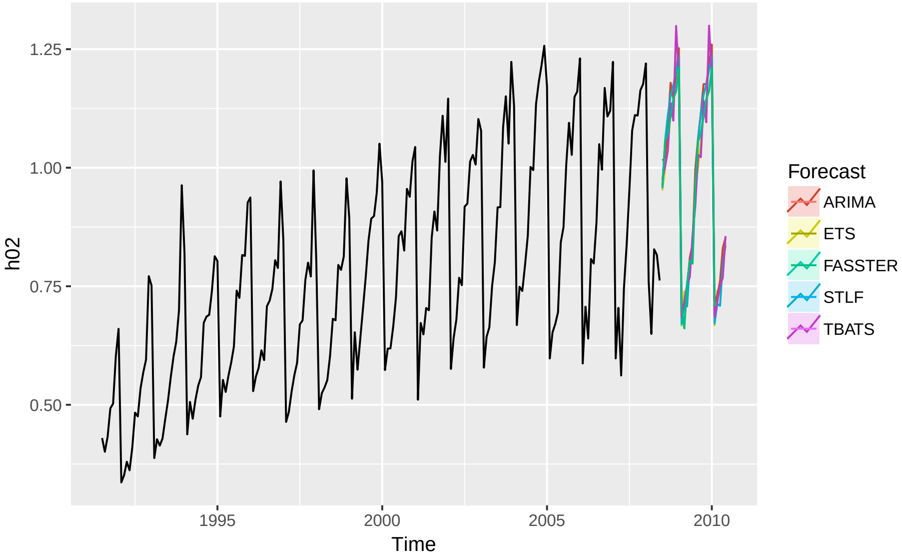


Forecasts (Pedestrian counts)



Outline

Pharmaceutical sales



R packages

https://github.com/earowang/tsibble

http://pkg.earo.me/sugrrants

https://github.com/mitchelloharawild/fasster

http://pkg.robjhyndman.com/forecast

http://pkg.earo.me/hts