2017 Beijing Workshop on Forecasting

Automatic Forecasting Algorithms

Rob J Hyndman

robjhyndman.com/beijing2017
Motivation
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.
Automatic Forecasting Algorithms

ETS

1 Motivation
2 ETS
3 ARIMA models
4 STLM
5 TBATS
6 FASSTER
7 Comparisons
## Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N (None)</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>(N,N)</td>
</tr>
<tr>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
<td>(A,N)</td>
</tr>
<tr>
<td></td>
<td>(A&lt;sub&gt;d&lt;/sub&gt;,N)</td>
</tr>
</tbody>
</table>

- **(N,N):** Simple exponential smoothing
- **(A,N):** Holt’s linear method
- **(A<sub>d</sub>,N):** Additive damped trend method
- **(A,A):** Additive Holt-Winters’ method
- **(A,M):** Multiplicative Holt-Winters’ method
- **(A<sub>d</sub>,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\}.
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

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N (None)</td>
</tr>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>N,A</td>
</tr>
<tr>
<td>A (d) (Additive damped)</td>
<td>N,M</td>
</tr>
<tr>
<td>A (d) (Additive damped)</td>
<td>A,N</td>
</tr>
<tr>
<td>A (d) (Additive damped)</td>
<td>A,A</td>
</tr>
<tr>
<td>A (d) (Additive damped)</td>
<td>A,M</td>
</tr>
<tr>
<td>A (d) (Additive damped)</td>
<td>A (d),N</td>
</tr>
<tr>
<td>A (d) (Additive damped)</td>
<td>A (d),A</td>
</tr>
<tr>
<td>A (d) (Additive damped)</td>
<td>A (d),M</td>
</tr>
</tbody>
</table>

**ETS 8**

**Automatic Forecasting Algorithms**
### Exponential smoothing models

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N (None)</td>
</tr>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,N</td>
</tr>
</tbody>
</table>

#### General notation

**ETS**: **Exponential Smoothing**

- **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

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
<th>Seasonal Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
</tr>
<tr>
<td>Ad (Additive damped)</td>
<td>Ad,N</td>
<td>Ad,A</td>
<td>Ad,M</td>
</tr>
</tbody>
</table>

#### General notation
E T S : Exponential Smoothing

#### 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

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component (N)</th>
<th>Seasonal Component (A)</th>
<th>Seasonal Component (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
</tr>
<tr>
<td>Ad (Additive damped)</td>
<td>Ad,N</td>
<td>Ad,A</td>
<td>Ad,M</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component N (None)</th>
<th>Seasonal Component A (Additive)</th>
<th>Seasonal Component M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
</tr>
<tr>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,N</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,A</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,M</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Additive Error Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N (None)</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A (Additive)</td>
</tr>
<tr>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
</tr>
<tr>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
</tr>
</tbody>
</table>

### Additive Error

- **Trend Component:**
  - N (None)
  - A (Additive)
  - A<sub>d</sub> (Additive damped)

### Seasonal Component

- **N (None):**
  - A,N,N
  - A,A,N
  - A,A<sub>d</sub>,N

- **A (Additive):**
  - A,N,A
  - A,A,A
  - A,A<sub>d</sub>,A

- **M (Multiplicative):**
  - A,N,M
  - A,A,M
  - A,A<sub>d</sub>,M

### Multiplicative Error

- **Trend Component:**
  - N (None)
  - A (Additive)
  - A<sub>d</sub> (Additive damped)

### Seasonal Component

- **N (None):**
  - M,N,N
  - M,A,N
  - M,A<sub>d</sub>,N

- **A (Additive):**
  - M,N,A
  - M,A,A
  - M,A<sub>d</sub>,A

- **M (Multiplicative):**
  - M,N,M
  - M,A,M
  - M,A<sub>d</sub>,M
ETS state space models

\[ x_{t-1} \rightarrow y_t \]

\[ \varepsilon_t \]

State space model

\[ x_t = (\text{level, slope, seasonal}) \]
ETS state space models

State space model

\[ x_t = (\text{level}, \text{slope}, \text{seasonal}) \]
ETS state space models

\[ x_{t-1}, \epsilon_t, x_t, \epsilon_{t+1} \rightarrow y_t \rightarrow y_{t+1} \]

State space model

\[ x_t = \text{(level, slope, seasonal)} \]
ETS state space models

\[ x_{t-1} \rightarrow y_t \rightarrow x_t \rightarrow y_{t+1} \]

\[ \varepsilon_t \rightarrow x_t \rightarrow \varepsilon_{t+1} \rightarrow x_{t+1} \]

State space model

\[ x_t = \text{(level, slope, seasonal)} \]
ETS state space models

State space model

\[ x_t = (\text{level, slope, seasonal}) \]
ETS state space models

\[ x_{t-1} \rightarrow y_t \rightarrow x_t \rightarrow y_{t+1} \rightarrow x_{t+1} \rightarrow y_{t+2} \rightarrow x_{t+2} \]

State space model

\[ x_t = (\text{level, slope, seasonal}) \]
ETS state space models

\[ \begin{align*}
\varepsilon_{t} & \quad \rightarrow \quad x_{t} \quad \rightarrow \quad y_{t} \\
\varepsilon_{t+1} & \quad \rightarrow \quad x_{t+1} \quad \rightarrow \quad y_{t+1} \\
\varepsilon_{t+2} & \quad \rightarrow \quad x_{t+2} \quad \rightarrow \quad y_{t+2} \\
\varepsilon_{t+3} & \quad \rightarrow \quad x_{t+3} \quad \rightarrow \quad y_{t+3}
\end{align*} \]

State space model
\[ x_{t} = (\text{level, slope, seasonal}) \]
ETS state space models

State space model
\[ x_t = (\text{level}, \text{slope}, \text{seasonal}) \]
ETS state space models

\[ x_{t-1} \rightarrow y_t \]
\[ \varepsilon_t \rightarrow x_t \rightarrow y_{t+1} \]
\[ \varepsilon_{t+1} \rightarrow x_{t+1} \rightarrow y_{t+2} \]
\[ \varepsilon_{t+2} \rightarrow x_{t+2} \rightarrow y_{t+3} \]
\[ \varepsilon_{t+3} \rightarrow x_{t+3} \rightarrow y_{t+4} \]

State space model

\[ x_t = \text{(level, slope, seasonal)} \]
ETS state space models

State space model
\( x_t = \text{(level, slope, seasonal)} \)

Estimation
Compute likelihood \( L \) from \( \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T \).
Optimize \( L \) wrt model parameters.
ETS state space models

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

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

\[ \mathbf{x}_t = f(\mathbf{x}_{t-1}) + g(\mathbf{x}_{t-1})\varepsilon_t \]  
State equation

Additive errors:

\[ k(\mathbf{x}_{t-1}) = 1. \quad y_t = \mu_t + \varepsilon_t. \]

Multiplicative errors:

\[ k(\mathbf{x}_{t-1}) = \mu_t. \quad y_t = \mu_t(1 + \varepsilon_t). \]

\[ \varepsilon_t = (y_t - \mu_t)/\mu_t \]  
is relative error.
Innovations state space models

Estimation

\[
L^*(\theta, x_0) = n \log \left( \sum_{t=1}^{n} \frac{\varepsilon_t^2}{k^2(x_{t-1})} \right) + 2 \sum_{t=1}^{n} \log |k(x_{t-1})| \\
= -2 \log(\text{Likelihood}) + \text{constant}
\]

- Estimate parameters \( \theta = (\alpha, \beta, \gamma, \phi) \) and initial states \( x_0 = (\ell_0, b_0, s_0, s_{-1}, \ldots, s_{-m+1}) \) 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

Forecasts from ETS(M,A,N)

Automatic Forecasting Algorithms

ETS
Example: drug sales

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

Forecasts from ETS(M,Ad,M)
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
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 designed to explain the mechanics of all the exponential smoothing methods and their role in forecasting time series. It should help you understand the intuition behind each method and how 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
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...
Automated Forecasting Algorithms

ARIMA models

Inputs

\[ y_{t-1} \]

\[ y_{t-2} \]

\[ y_{t-3} \]

Output

\[ y_t \]
ARIMA models

Inputs

$y_{t-1}$
$y_{t-2}$
$y_{t-3}$

Output

$\varepsilon_t$

Autoregression (AR) model

$y_t = y_{t-1} + y_{t-2} + y_{t-3} + \varepsilon_t$
ARIMA models

Inputs

\( y_{t-1} \)
\( y_{t-2} \)
\( y_{t-3} \)
\( \epsilon_t \)
\( \epsilon_{t-1} \)
\( \epsilon_{t-2} \)

Output

\( y_t \)

Autoregression moving average (ARMA) model
ARIMA models

Autoregression moving average (ARMA) model

Inputs

$Y_{t-1}$

$Y_{t-2}$

$Y_{t-3}$

$\varepsilon_t$

$\varepsilon_{t-1}$

$\varepsilon_{t-2}$

Output

$Y_t$

Estimation

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

Use optimization algorithm to maximize $L$. 

Automatic Forecasting Algorithms

ARIMA models

20
ARIMA models

Inputs

$y_{t-1}$

$y_{t-2}$

$y_{t-3}$

$\epsilon_t$

$\epsilon_{t-1}$

$\epsilon_{t-2}$

Output

$y_t$

Autoregression moving average (ARMA) model applied to differences.

Estimation

Compute likelihood $L$ from $\epsilon_1, \epsilon_2, \ldots, \epsilon_T$.

Use optimization algorithm to maximize $L$. 
livestock %>% auto.arima() %>% forecast() %>% autoplot

Forecasts from ARIMA(0,1,0) with drift

Automatic Forecasting Algorithms
ARIMA models
Auto ARIMA

\texttt{h02} \texttt{\%\%>\% auto.arima()} \texttt{\%\%>\% forecast()} \texttt{\%\%>\% autoplot()}

Forecasts from ARIMA(3,1,3)(0,1,1)[12]
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

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

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

1. Motivation
2. ETS
3. ARIMA models
4. STLM
5. TBATS
6. FASSTER
7. Comparisons
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.
```r
fit <- stl(elecequip, s.window=5, robust=TRUE)
autoplot(fit) +
ggtitle("STL decomposition of electrical equipment index")
```
**STL decomposition**

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

---

**STL decomposition of electrical equipment index**

- **Data**
- **Trend**
- **Seasonal**
- **Remainder**

---

**Automatic Forecasting Algorithms**

STLM
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

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

Forecasts from ETS(A,Ad,N)

![Graph showing seasonal adjustment and forecasts for electrical equipment.](image)
fit %>% forecast() %>% autoplot()

Forecasts from STL + ETS(A,Ad,N)

Automatic Forecasting Algorithms  STLM
Forecasting and decomposition

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

Forecasts from STL + ETS(A,Ad,N)

### Automatic Forecasting Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>STLM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>32</td>
</tr>
</tbody>
</table>
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.
Complex seasonality

Gasoline data

Thousands of barrels per day

Year

Automatic Forecasting Algorithms

TBATS
Complex seasonality

Forecasts from TBATS(1, {0,0}, −, {<52.18,9>})

Thousands of barrels per day

Year

Automatic Forecasting Algorithms

TBATS
Complex seasonality

Forecasts from TBATS(1, {0,0}, -, {<52.18,9>})

```r
fit <- tbats(gasoline)
fcast <- forecast(fit)
autoplot(fcast)
```
Complex seasonality

Number of calls to large American bank {7am..9pm}

Week
Number of calls in 5 minute interval

Number of calls in 5 minute interval

Week

Automatic Forecasting Algorithms

TBATS
Complex seasonality

Forecasts from TBATS$(1, \{3,1\}, 0.8, \{<169,6>, <845,4>\})$
Complex seasonality

Forecasts from TBATS(1, \{3,1\}, 0.8, \{<169,6>, <845,4>\})

```r
fit <- tbats(calls)
fcast <- forecast(fit)
autoplot(fcast)
```
TBATS model

\[ y_t = \text{observation at time } t \]

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

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

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

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

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

\[
s_i^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}
\]

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

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

\[ y_t = \text{observation at time } t \]

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

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

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

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

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

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

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

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

Box-Cox transformation
TBATS model

\[ y_t = \text{observation at time } t \]

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

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

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

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

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

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

Box-Cox transformation

\[ M \text{ seasonal periods} \]
TBATS model

\[ y_t = \text{observation at time } t \]

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

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

\[ l_t = l_{t-1} + \phi b_{t-1} + \alpha d_t \]

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

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

\[ s_{t}^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)} \]

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

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

\[ y_t = \text{observation at time } t \]

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

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

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

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

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

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

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

\[ s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \]
The TBATS model is given by:

\[ y_t = \text{observation at time } t \]

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

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

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

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

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

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

- **Box-Cox transformation**
- **M seasonal periods**
- **global and local trend**
- **ARMA error**
- **Fourier-like seasonal terms**
TBATS model

\[ y_t = \text{observation at time } t \]

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

\[ y_t^{(\omega)} = \ell_{t-1} + b_t \]

\[ \ell_t = \ell_{t-1} + \phi b_{t-1} + M \sum_{i=1}^{\omega} s(i)_{j,t} - m_i + \alpha d_t \]

\[ b_t = (1 - \phi) b_{t-1} + \beta d_t \]

\[ d_t = \sum_{i=1}^{p} \phi_i d_{t-i} \]

\[ s(i)_{j,t} = \sum_{j=1}^{k_i} s(j)_{j,t} \]

\[ s(j)_{j,t} = s(j)_{j,t-1} \cos \lambda_j + s(j)_{j,t-1} \sin \lambda_j + \gamma_1 d_t \]

\[ s(j)_{j,t} = -s(j)_{j,t-1} \sin \lambda_j + s(j)_{j,t-1} \cos \lambda_j + \gamma_2 d_t \]

- **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
Pedestrian counts

Hourly pedestrian traffic at Southern Cross Station

Pedestrians counted

Hourly pedestrian traffic at Southern Cross Station

Pedestrians counted

Automatic Forecasting Algorithms

FASSTER
Pedestrian counts

Seasonality in pedestrian traffic at Southern Cross Station

<table>
<thead>
<tr>
<th>Time</th>
<th>Weekday</th>
<th>Weekend</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 AM</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>06 AM</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>12 PM</td>
<td>2000</td>
<td>0</td>
</tr>
<tr>
<td>18 PM</td>
<td>3000</td>
<td>0</td>
</tr>
<tr>
<td>00 AM</td>
<td>4000</td>
<td>0</td>
</tr>
</tbody>
</table>

Total pedestrians counted

Automatic Forecasting Algorithms

FASSTER
Switching Structure

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

**Dynamic linear model**

\[
y_t = F_t \theta_t + v_t, \quad v_t \sim \mathcal{N}(0, V)
\]

\[
\theta_t = G\theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, W)
\]

Switch between two groups:

\[
y_t = I_{t \in G_1} F_t \theta_t^{(1)} + I_{t \in G_2} F_t \theta_t^{(2)} + v_t, \quad v_t \sim \mathcal{N}(0, V)
\]

Groups \( G_1 \) and \( G_2 \) define the switching rule (say weekdays and weekends).
FASSTER extends current state-space approaches by switching between states.

**Dynamic linear model**

\[
y_t = F_t \theta_t + v_t, \quad v_t \sim \mathcal{N}(0, V)
\]

\[
\theta_t = G \theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, W)
\]

Switch between two groups:

\[
y_t = I_{t \in G_1} F_t \theta_t^{(1)} + I_{t \in G_2} F_t \theta_t^{(2)} + v_t \quad v_t \sim \mathcal{N}(0, V)
\]

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

\[
y_t = F_t \theta_t + v_t, \quad v_t \sim \mathcal{N}(0, V)
\]

\[
\theta_t = G \theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, W)
\]

Switch between two groups:

\[
y_t = I_{t \in G_1} F_t \theta_t^{(1)} + I_{t \in G_2} F_t \theta_t^{(2)} + v_t, \quad v_t \sim \mathcal{N}(0, V)
\]

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.

Pedestrian Traffic at Southern Cross Station

Pedestrians counted

Date

Automatic Forecasting Algorithms

FASSTER
An appropriate model can be constructed using switching states:

\[ y_t = \begin{cases} 
  I_{t \in \text{Weekday}} F_t \theta_t^{(\text{Weekday})} + \begin{cases} 
    v_t 
  \end{cases} & \text{if } t \in \text{Weekday} \\
  I_{t \in \text{Weekend}} F_t \theta_t^{(\text{Weekend})} + \begin{cases} 
    v_t 
  \end{cases} & \text{if } t \in \text{Weekend}
\end{cases} \]

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

- \( \ell_t \) 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

\[ y_t = F_t^{(0)} \theta_t^{(0)} + \sum_{j=1}^{k} I_{t \in G_j} F_t^{(j)} \theta_t^{(j)} + \nu_t, \quad \nu_t \sim \mathcal{N}(0, V) \]

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

\[ y_t = F_t^{(0)} \theta_t^{(0)} + \sum_{j=1}^{k} I_{t \in G_j} F_t^{(j)} \theta_t^{(j)} + v_t, \quad v_t \sim \mathcal{N}(0, V) \]

where \( k \) is the number of switching combinations.
A FASSTER model can be represented as a time-varying DLM.

\[
\begin{align*}
y_t &= F_t \theta_t + v_t, \\
\theta_t &= G \theta_{t-1} + w_t, \\
\text{where } \theta_0 &\sim \mathcal{N}(m_0, C_0)
\end{align*}
\]

\[v_t \sim \mathcal{N}(0, V)\]
\[w_t \sim \mathcal{N}(0, W)\]

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

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

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)))

Pedestrian Traffic at Southern Cross Station

Series
- Response
- Fitted

Automatic Forecasting Algorithms
FASSTER 52
Forecasts (Pedestrian counts)

Forecasts from FASSTER

Pedestrian Counts

Date


Automatic Forecasting Algorithms  FASSTER  53
R packages

- tsibble
  - https://github.com/earowang/tsibble

- sugrrants
  - http://pkg.earo.me/sugrrants

- fasster
  - https://github.com/mitchelloharawild/fasster

- forecast
  - http://pkg.robjhyndman.com/forecast

- hts
  - http://pkg.earo.me/hts