

Rob J Hyndman  
George Athanasopoulos

# FORECASTING

## PRINCIPLES AND PRACTICE

A comprehensive introduction to the latest forecasting methods using R. Learn to improve your forecast accuracy using dozens of real data examples.



3RD EDITION

 **OTexts**  
OPEN TEXTS FOR PRACTICE

## 7. Time series regression models

### 7.7 Nonlinear regression

[OTexts.org/fpp3/](https://OTexts.org/fpp3/)

# Nonlinear regression

A **log-log** functional form

$$\log y = \beta_0 + \beta_1 \log x + \varepsilon$$

where  $\beta_1$  is interpreted as an elasticity (the average percentage change in  $y$  resulting from a 1% increase in  $x$ ).

# Nonlinear regression

A log-log functional form

$$\log y = \beta_0 + \beta_1 \log x + \varepsilon$$

where  $\beta_1$  is interpreted as an elasticity (the average percentage change in  $y$  resulting from a 1% increase in  $x$ ).

- alternative specifications: log-linear, linear-log.
- use  $\log(x + 1)$  if required.

# Piecewise linear and regression splines

$$y = f(x) + \varepsilon$$

where  $f$  is a non-linear function.

# Piecewise linear and regression splines

$$y = f(x) + \varepsilon$$

where  $f$  is a non-linear function.

- For **piecewise linear** let  $x_1 = x$  and

$$x_2 = (x - c)_+ = \begin{cases} 0 & \text{if } x < c \\ x - c & \text{if } x \geq c. \end{cases}$$

# Piecewise linear and regression splines

$$y = f(x) + \varepsilon$$

where  $f$  is a non-linear function.

- For **piecewise linear** let  $x_1 = x$  and

$$x_2 = (x - c)_+ = \begin{cases} 0 & \text{if } x < c \\ x - c & \text{if } x \geq c. \end{cases}$$

- In general, **linear regression splines**

$$x_1 = x \quad x_2 = (x - c_1)_+ \quad \dots \quad x_k = (x - c_{k-1})_+$$

where  $c_1, \dots, c_{k-1}$  are knots.

# Piecewise linear and regression splines

$$y = f(x) + \varepsilon$$

where  $f$  is a non-linear function.

- For **piecewise linear** let  $x_1 = x$  and

$$x_2 = (x - c)_+ = \begin{cases} 0 & \text{if } x < c \\ x - c & \text{if } x \geq c. \end{cases}$$

- In general, **linear regression splines**

$$x_1 = x \quad x_2 = (x - c_1)_+ \quad \dots \quad x_k = (x - c_{k-1})_+$$

where  $c_1, \dots, c_{k-1}$  are knots.

- Need to select knots: can be difficult and arbitrary.
- Automatic knot selection algorithms very slow.
- Using piecewise cubics achieves a smoother result.

Warning: better fit but forecasting outside the range of the historical data is even more unreliable.

# Nonlinear trends

Piecewise linear trend with bend at  $\tau$

$$x_{1,t} = t$$

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geq \tau \end{cases}$$



# Nonlinear trends

Piecewise linear trend with bend at  $\tau$

$$x_{1,t} = t$$

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geq \tau \end{cases}$$

Quadratic or higher order trend

$$x_{1,t} = t, \quad x_{2,t} = t^2, \quad \dots$$

# Nonlinear trends

Piecewise linear trend with bend at  $\tau$

$$x_{1,t} = t$$

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geq \tau \end{cases}$$

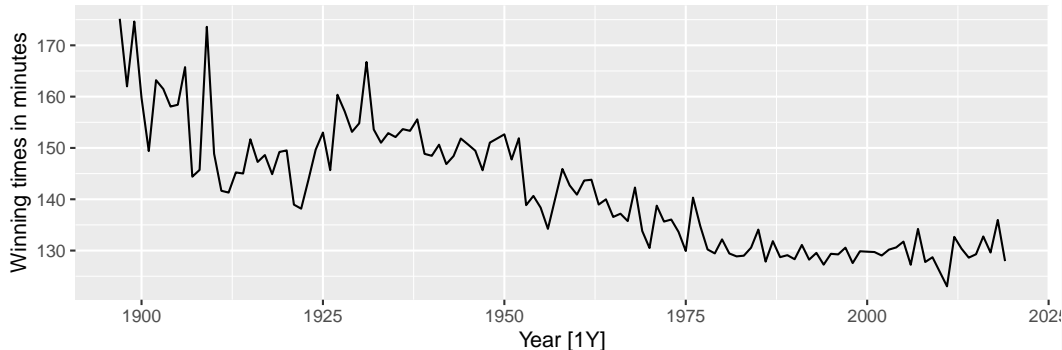
Quadratic or higher order trend

$$x_{1,t} = t, \quad x_{2,t} = t^2, \quad \dots$$

**NOT RECOMMENDED!**

# Example: Boston marathon winning times

```
marathon <- boston_marathon |>  
  filter(Event == "Men's open division") |>  
  select(-Event) |>  
  mutate(Minutes = as.numeric(Time) / 60)  
marathon |> autoplot(Minutes) + labs(y = "Winning times in minutes")
```



# Example: Boston marathon winning times

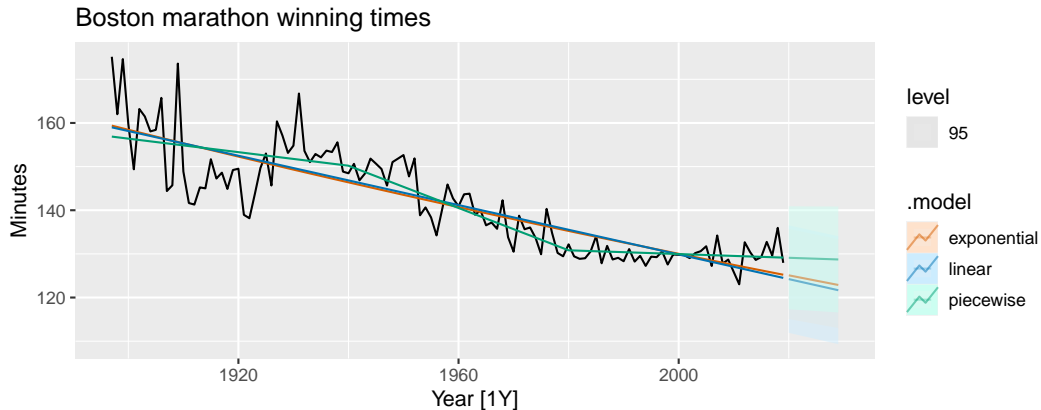
```
fit_trends <- marathon |>
  model(
    # Linear trend
    linear = TSLM(Minutes ~ trend()),
    # Exponential trend
    exponential = TSLM(log(Minutes) ~ trend()),
    # Piecewise linear trend
    piecewise = TSLM(Minutes ~ trend(knots = c(1940, 1980)))
  )
```

```
fit_trends
```

```
## # A mable: 1 x 3
##   linear exponential piecewise
##   <model>      <model>    <model>
## 1  <TSLM>      <TSLM>    <TSLM>
```

# Example: Boston marathon winning times

```
fit_trends |>  
  forecast(h = 10) |>  
  autoplot(marathon)
```



# Example: Boston marathon winning times

```
fit_trends |>  
  select(piecewise) |>  
  gg_tsresiduals()
```

