

An introduction to GAM(M)s

Stefano Coretta

12/07/2018

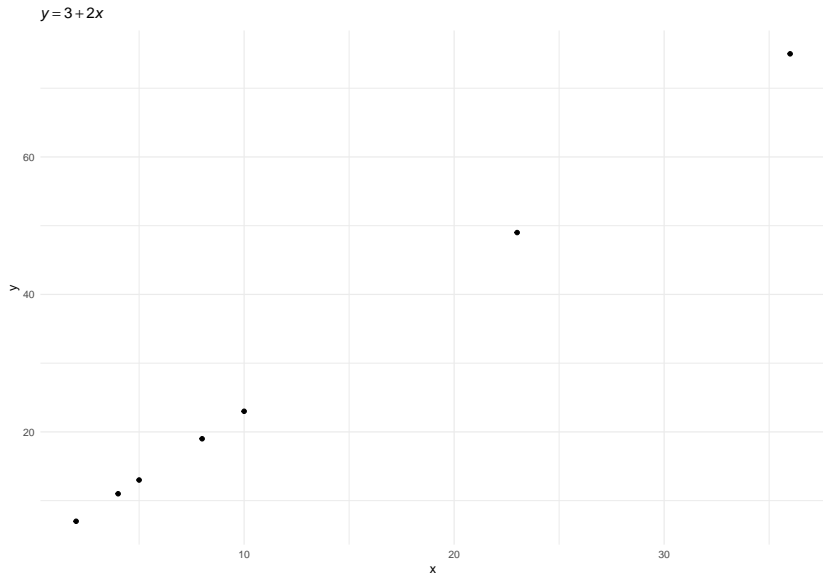
Time travel...

Linear models

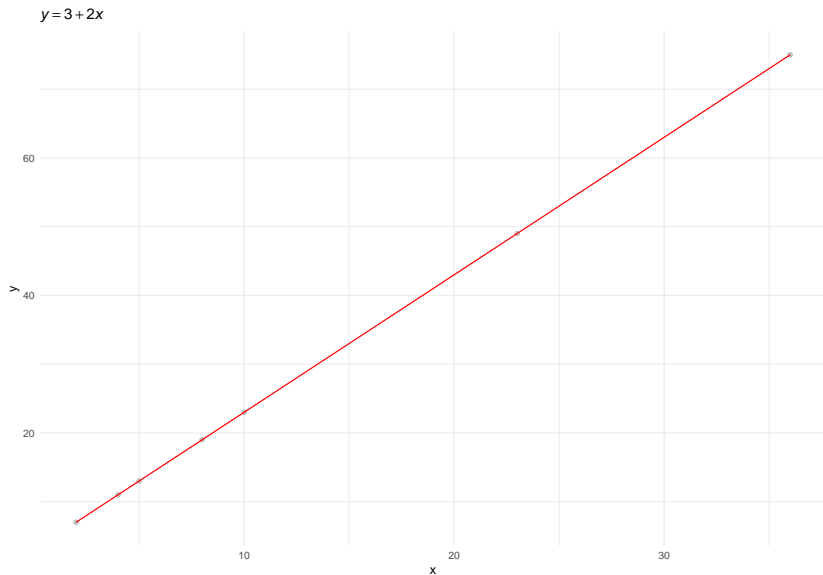
$$y = 3 + 2x$$

where $x = (2, 4, 5, 8, 10, 23, 36)$

Linear models



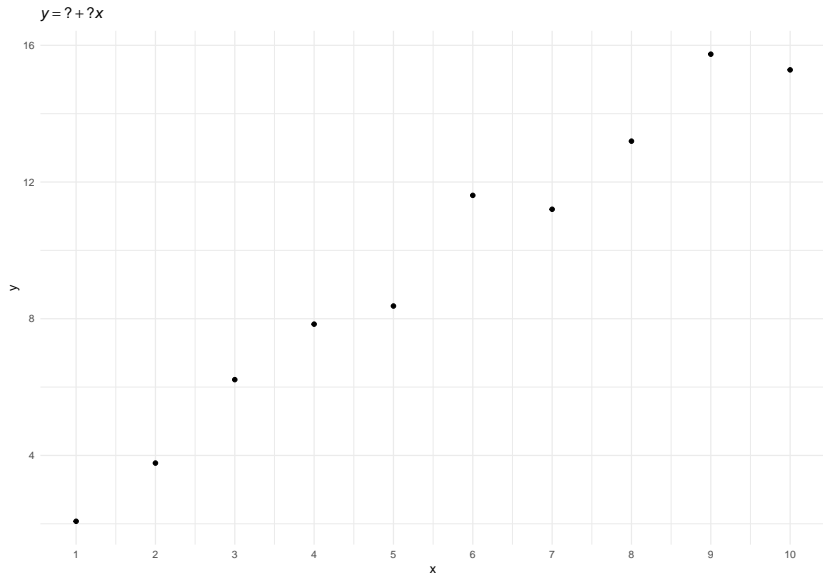
Linear models



Linear models

- ▶ In science, we have x and y ...
- ▶ for example, vowel duration and VOT

Linear models

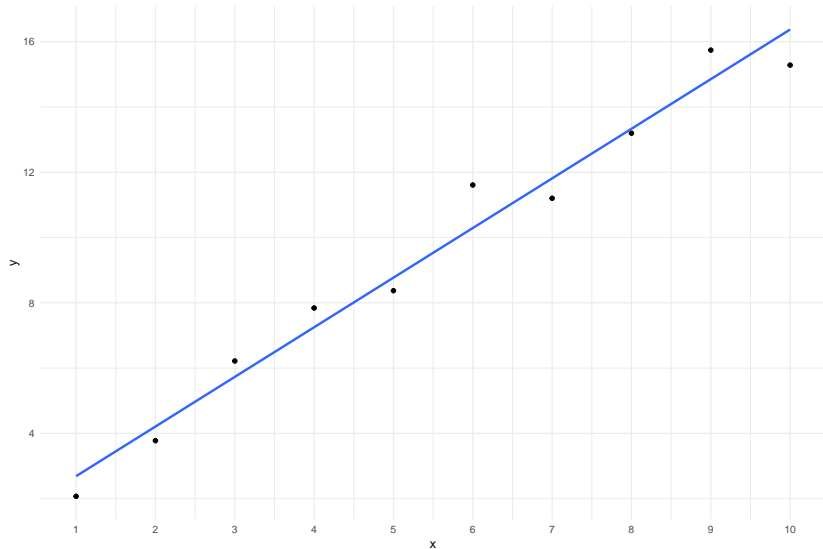


Linear models

- ▶ The formula: $y = \beta_0 + \beta_1 x$
 - ▶ β_0 is the **intercept**
 - ▶ β_1 is the **slope**
- ▶ We know x and y
 - ▶ we need to estimate $\beta_0, \beta_1 = \hat{\beta}_0, \hat{\beta}_1$
- ▶ We can add more predictors
 - ▶ $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$
- ▶ `lm(y ~ x, data)` ('y as a function of x')

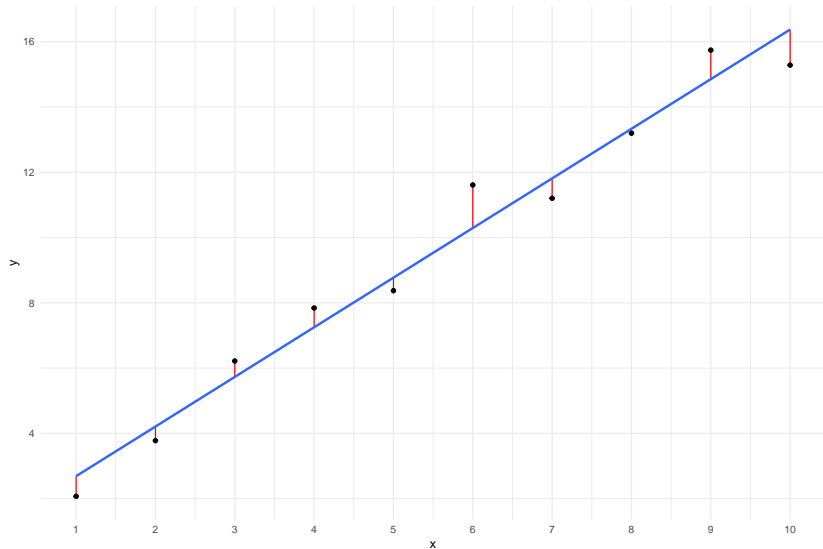
Linear models

$$y = \beta_0 + \beta_1 x = 1 + 1.5x$$



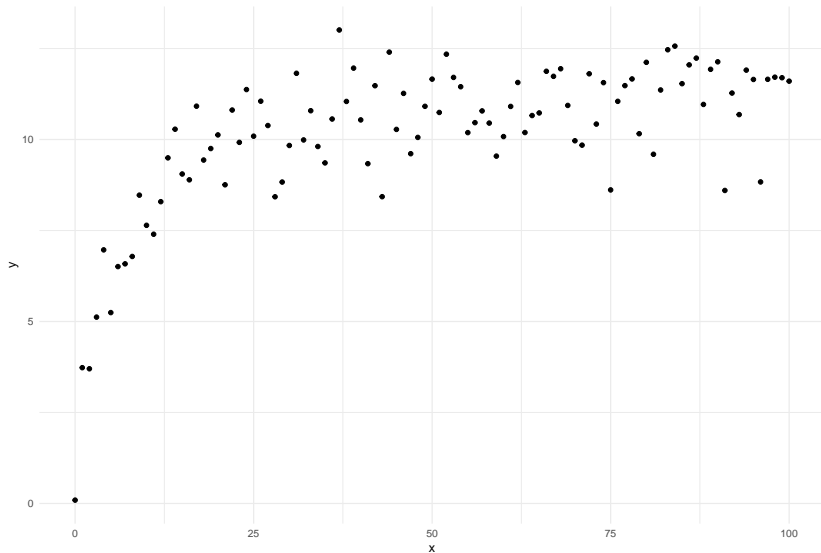
Linear models

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



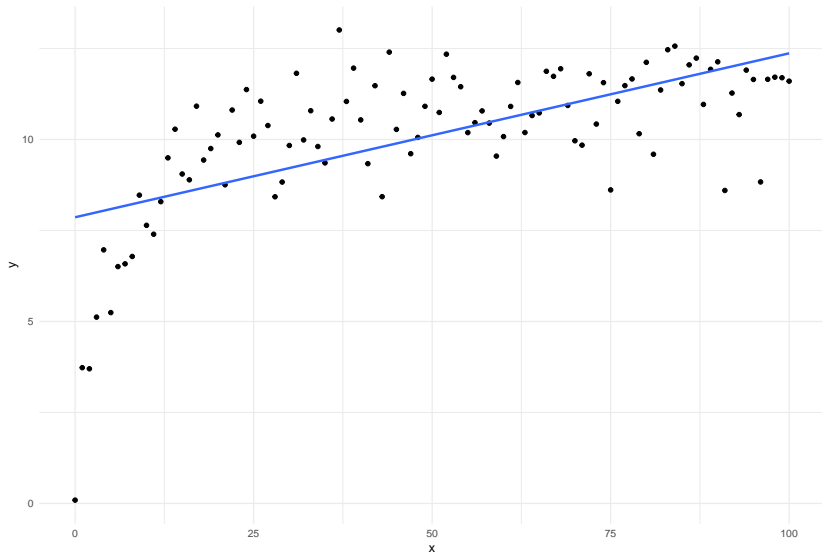
LM with non-linear data

Some non-linear data



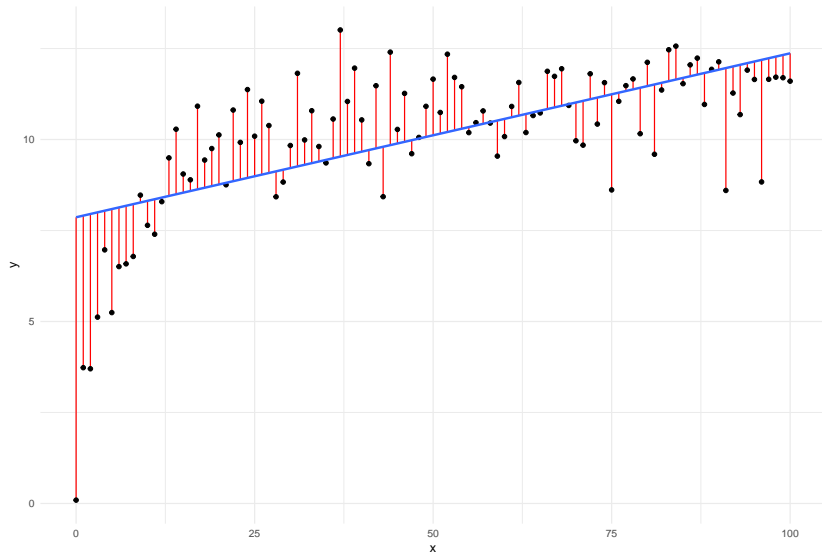
LM with non-linear data

Some non-linear data



LM with non-linear data

Some non-linear data



LM with non-linear data

How to account for non-linearity in a linear model?

- ▶ Use **higher-degree polynomials**

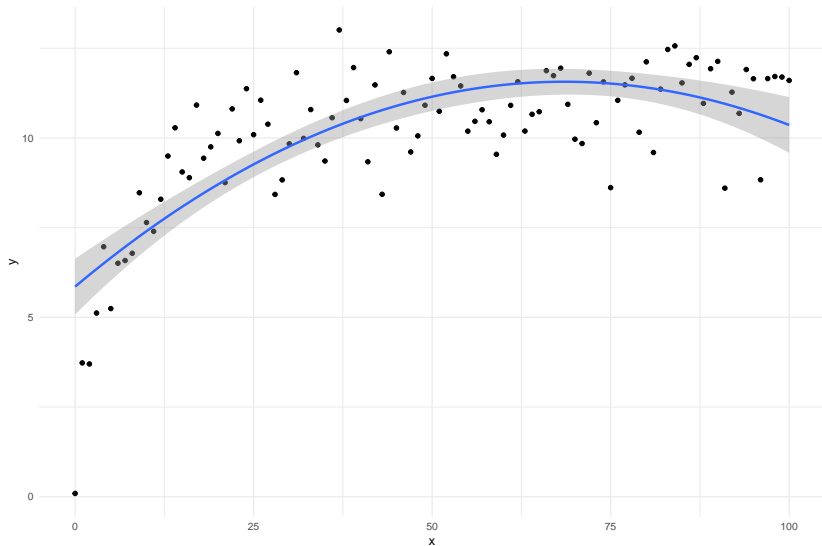
- ▶ quadratic: $y = \beta_0 + \beta_1x + \beta_2x^2$

- ▶ cubic: $y = \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3$

- ▶ n th: $y = \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3 + \dots + \beta_nx^n$

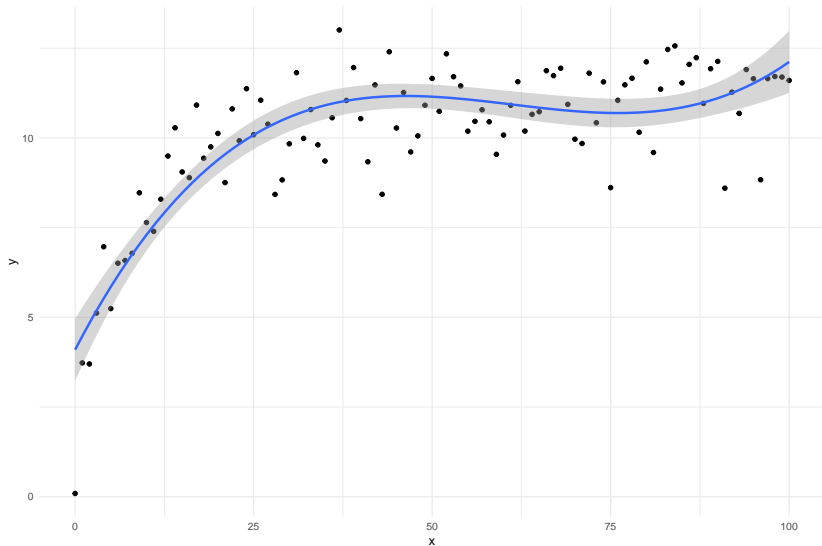
LM with non-linear data

$$y = \beta_0 + \beta_1 x + \beta_2 x^2$$



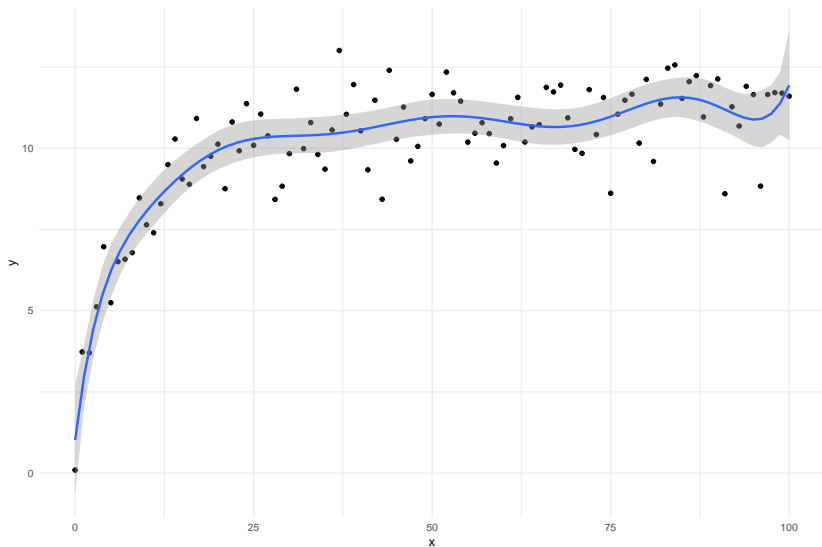
LM with non-linear data

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$



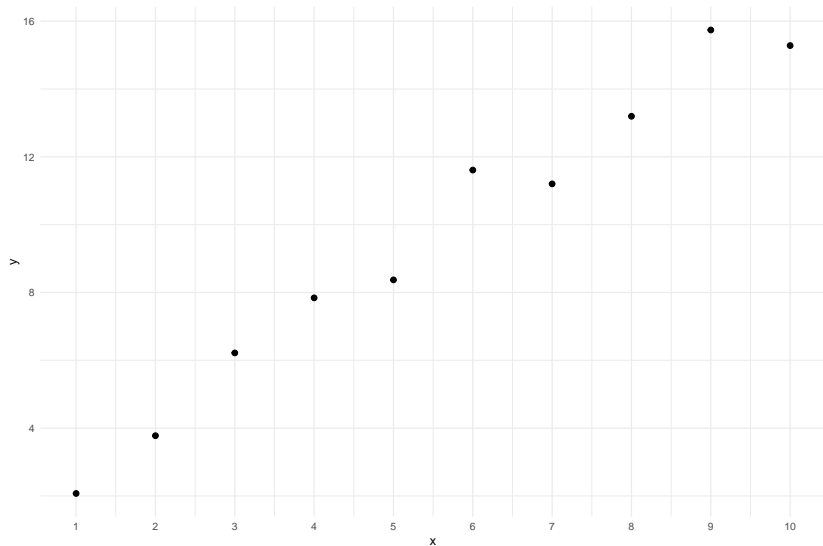
LM with non-linear data

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5 + \beta_6 x^6 + \beta_7 x^7 + \beta_8 x^8 + \beta_9 x^9 + \beta_{10} x^{10}$$



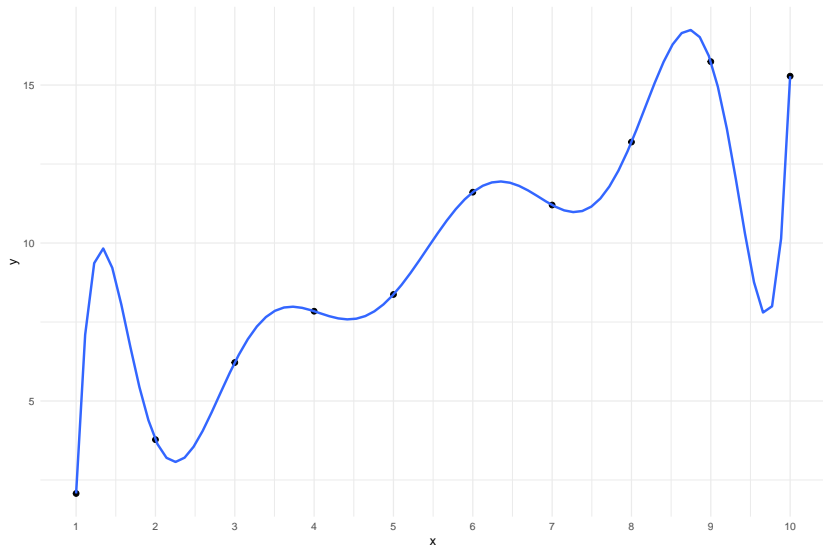
LM with non-linear data

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5 + \beta_6 x^6 + \beta_7 x^7 + \beta_8 x^8 + \beta_9 x^9 + \beta_{10} x^{10}$$



LM with non-linear data

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5 + \beta_6 x^6 + \beta_7 x^7 + \beta_8 x^8 + \beta_9 x^9 + \beta_{10} x^{10}$$



Generalised additive models

- ▶ **Generalised Additive Models**

- ▶ $y = f(x) + \epsilon$

- ▶ $f(x)$ = 'some function of x ' (or *smooth function*)

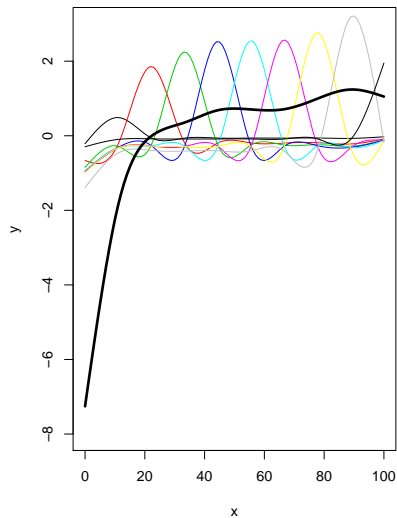
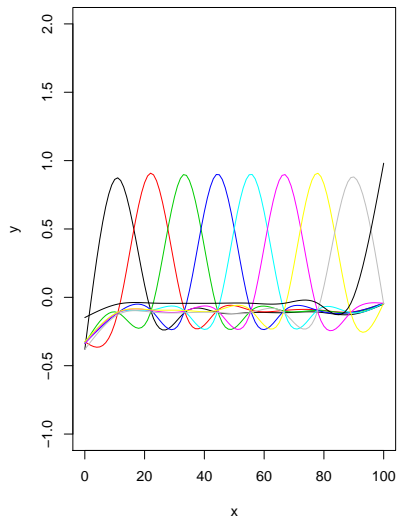
Smooth terms

- ▶ LMs have **parametric terms**
 - ▶ $\beta_n x_n$ (x in \mathbb{R})
 - ▶ linear effects
- ▶ GAMs add (non-parametric) **smooth terms** (or simply smooths, also smoothers)
 - ▶ $f(x)$, $s(x)$ in \mathbb{R}
 - ▶ non-linear effects
- ▶ `gam(y ~ s(x), data)`, 'y as *some* function of x'

Smoothing splines, basis, basis functions

- ▶ smooths in GAMs are **smoothing splines**
 - ▶ splines are defined piecewise with a set of polynomials
- ▶ the set of polynomials is called a **basis**
 - ▶ the basis is composed of **basis functions** (the polynomials)
- ▶ a spline is the sum of the products of each basis function and its coefficient

Basis functions



Smoothing splines

- ▶ there are **several kinds** of splines
 - ▶ each with their own basis functions
 - ▶ thin plate regression splines
 - ▶ cubic regression splines
- ▶ for more info, run `?smooth.terms`

A simple GAM

```
simple <- gam(y ~ s(x, bs = "cr", k = 10), data = sim_nl_a)
summary(simple)
```

```
##
```

```
## Family: gaussian
```

```
## Link function: identity
```

```
##
```

```
## Formula:
```

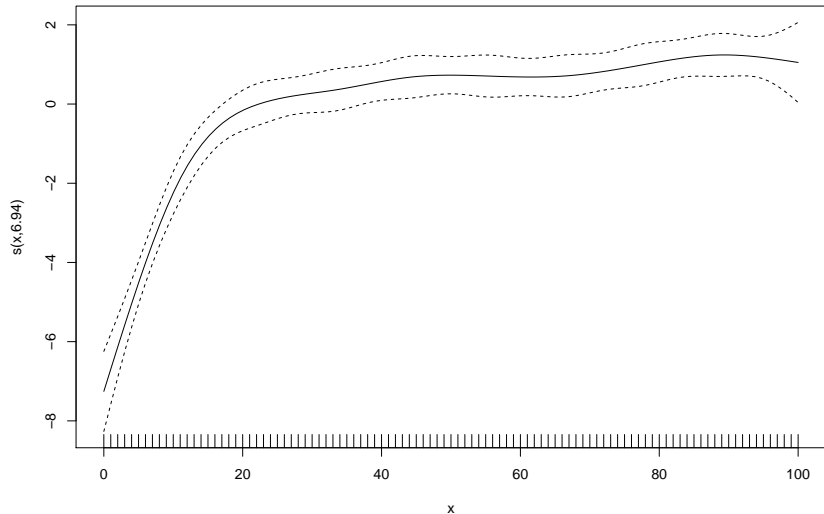
```
## y ~ s(x, bs = "cr", k = 10)
```

```
##
```

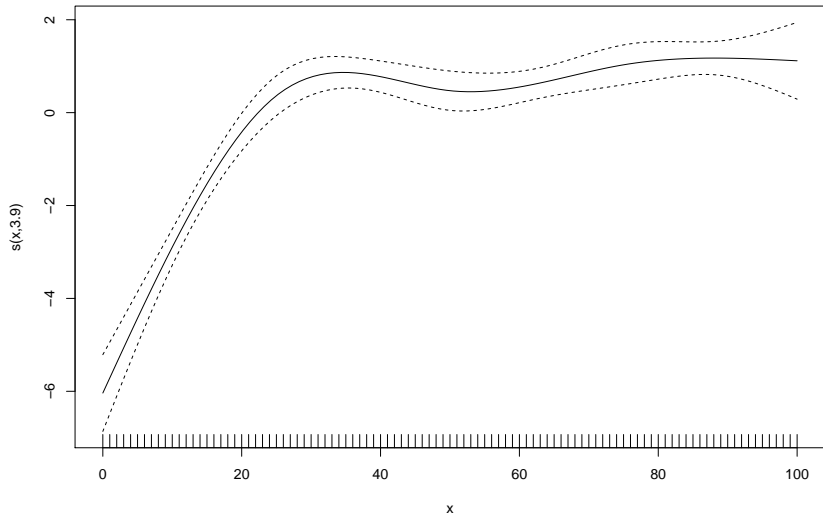
```
## Parametric coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
```

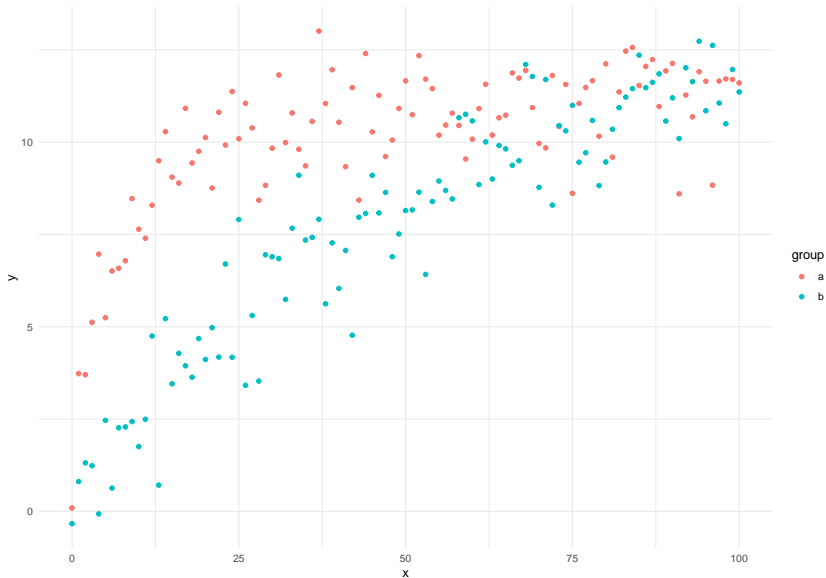
A simple GAM



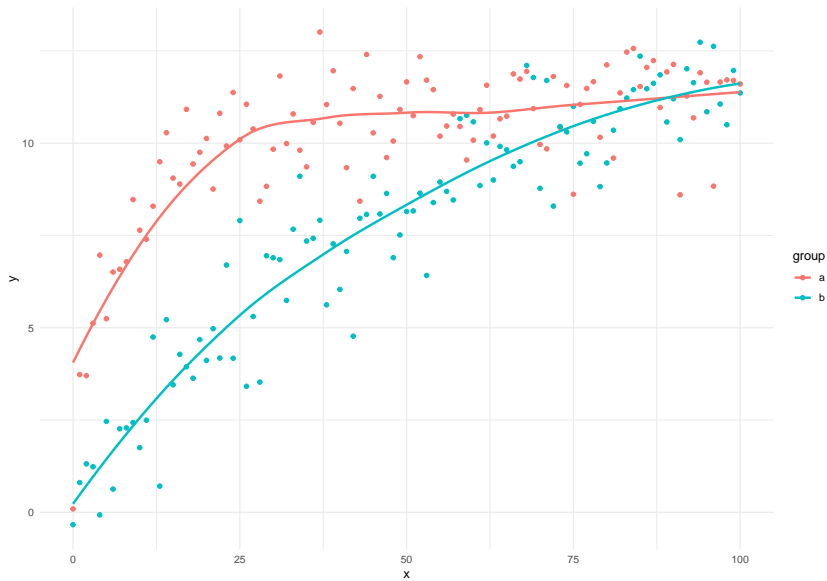
A simple GAM



Comparing levels



Comparing levels



Comparing levels

- ▶ ordered by-variables

```
compare <- gam(  
  y ~  
    group +  
    s(x, bs = "cr", k = 5) +  
    s(x, bs = "cr", k = 5, by = group),  
  data = sim_n1  
)
```

Comparing levels

- ▶ to use ordered by-variables
 - ▶ change factor to **ordered factor**
 - ▶ change factor contrast to **treatment contrast**
(`contr.treatment`)
 - ▶ the default in ordered factors is `contr.poly`, this won't work
 - ▶ include factor as **parametric term**
 - ▶ include a **reference smooth** and a **difference smooth** with the by-variable