Generalized Additive Models

David L Miller

Overview

- What is a GAM?
- What is smoothing?
- How do GAMs work? (*Roughly*)

From GAMs to GLMs and LMs

(Generalized) Linear Models

Models that look like:

$$ y_i = \beta_0 + x_{1i}\beta_1 + x_{2i}\beta_2 + \ldots + \epsilon_i $$

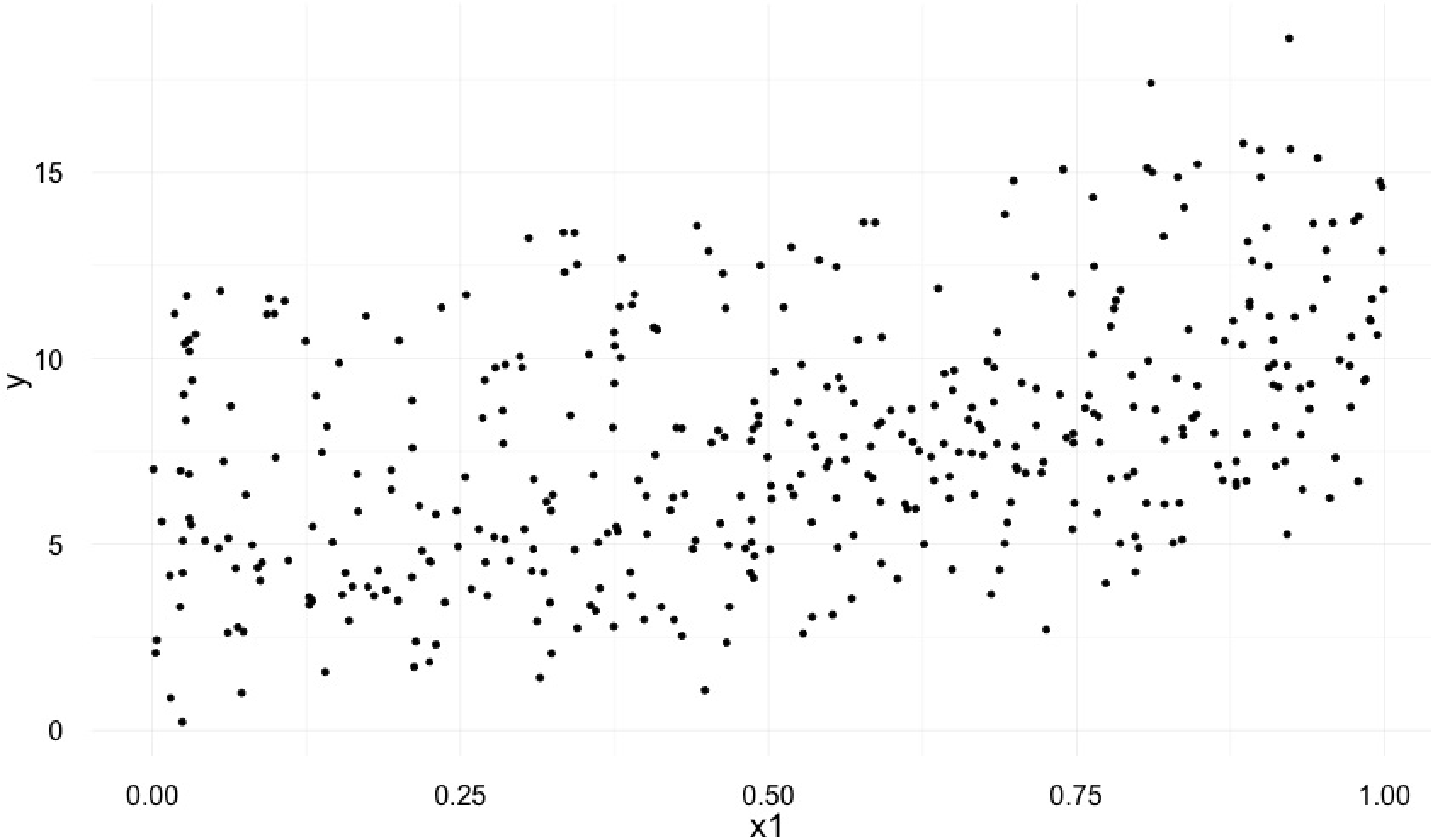
(describe the response, y_i, as linear combination of the covariates, x_{ji}, with an offset)

We can make $y_i\sim$ any exponential family distribution (Normal, Poisson, etc).

Error term ϵ_i is normally distributed (usually).

Why bother with anything more complicated?!

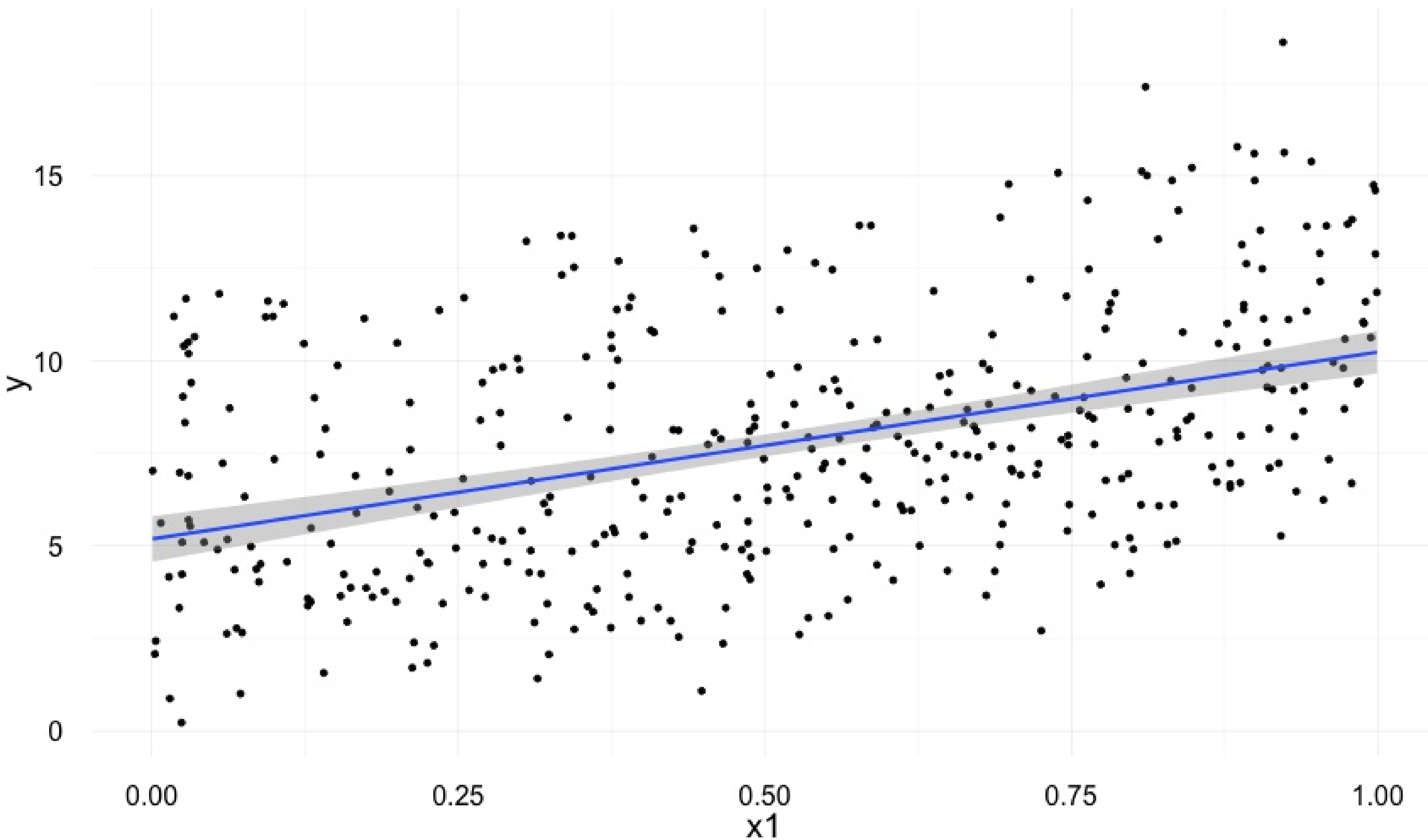
Is this relationship linear?



A linear model...

```r
lm(y ~ x1 + poly(x1, 2), data=dat)
```

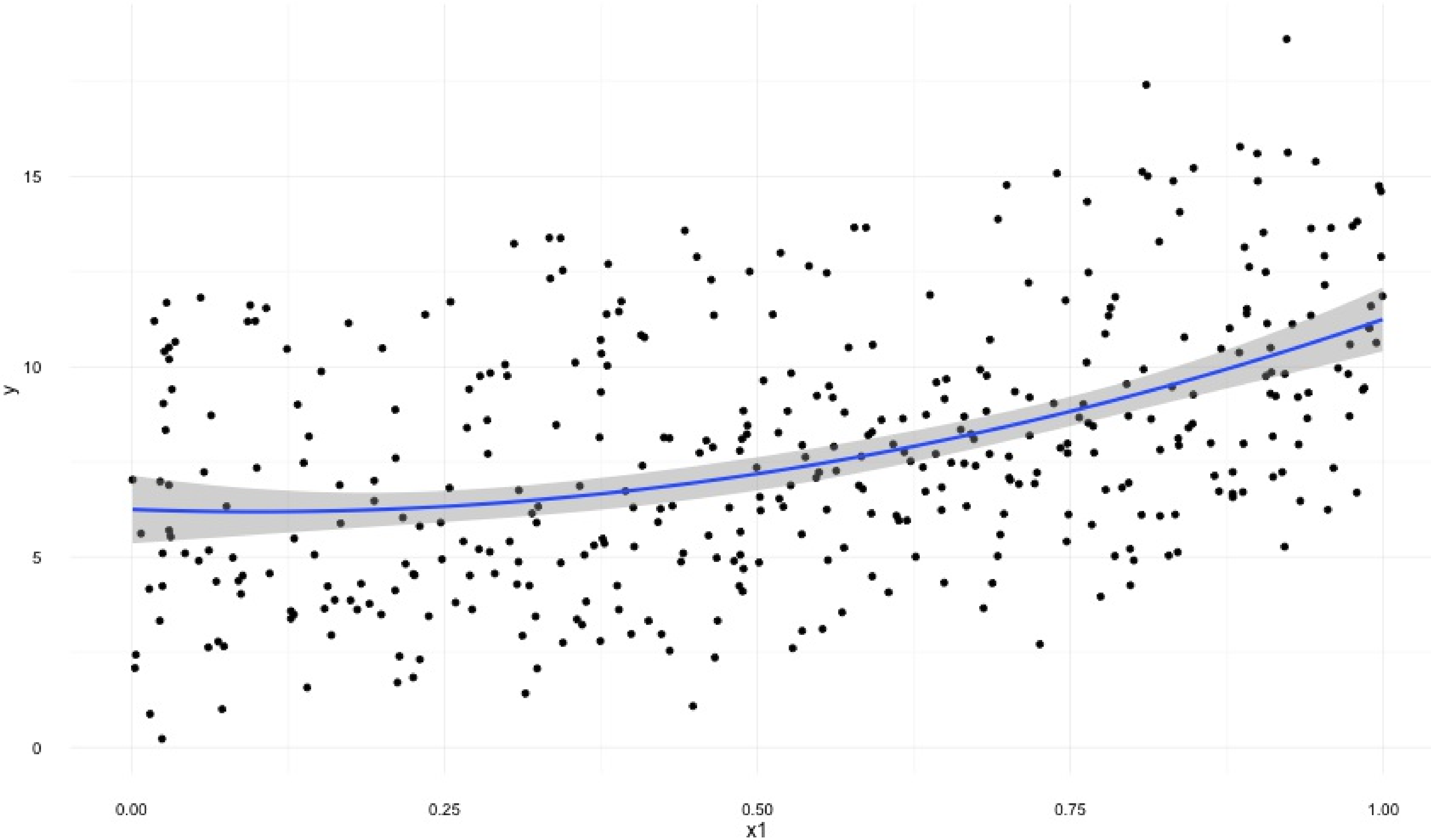
Is this relationship linear? Maybe?



What can we do?

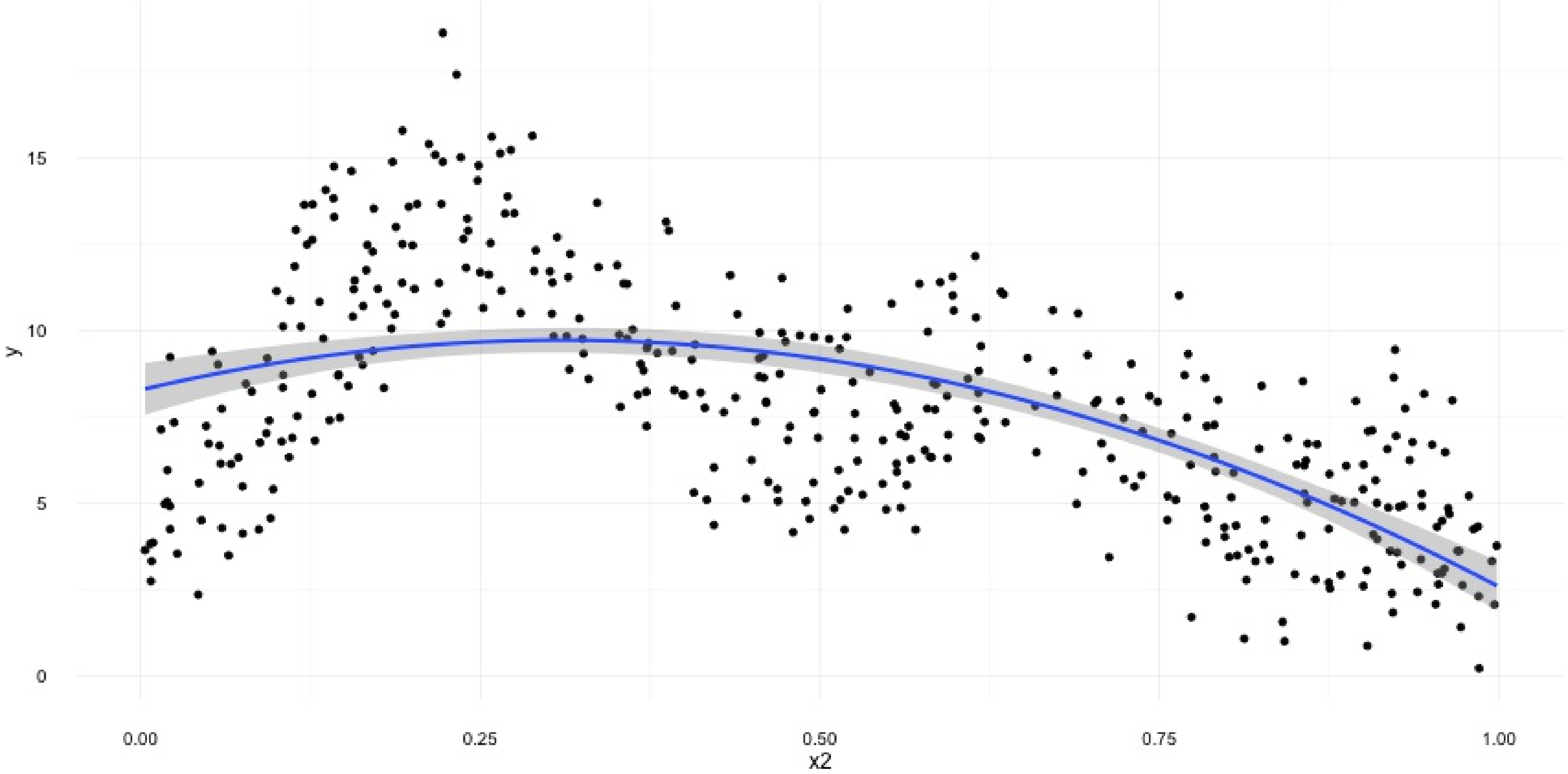
```
lm(y ~ x1 + poly(x1, 2), data=dat)
```

Adding a quadratic term?



Is this sustainable?

- Adding in quadratic (and higher terms) *can* make sense
- This feels a bit *ad hoc*
- Better if we had a **framework** to deal with these issues?



[drumroll]

Generalized Additive Models

- Generalized: many response distributions
- Additive: terms **add** together
- Models: well, it's a model…

What does a model look like?

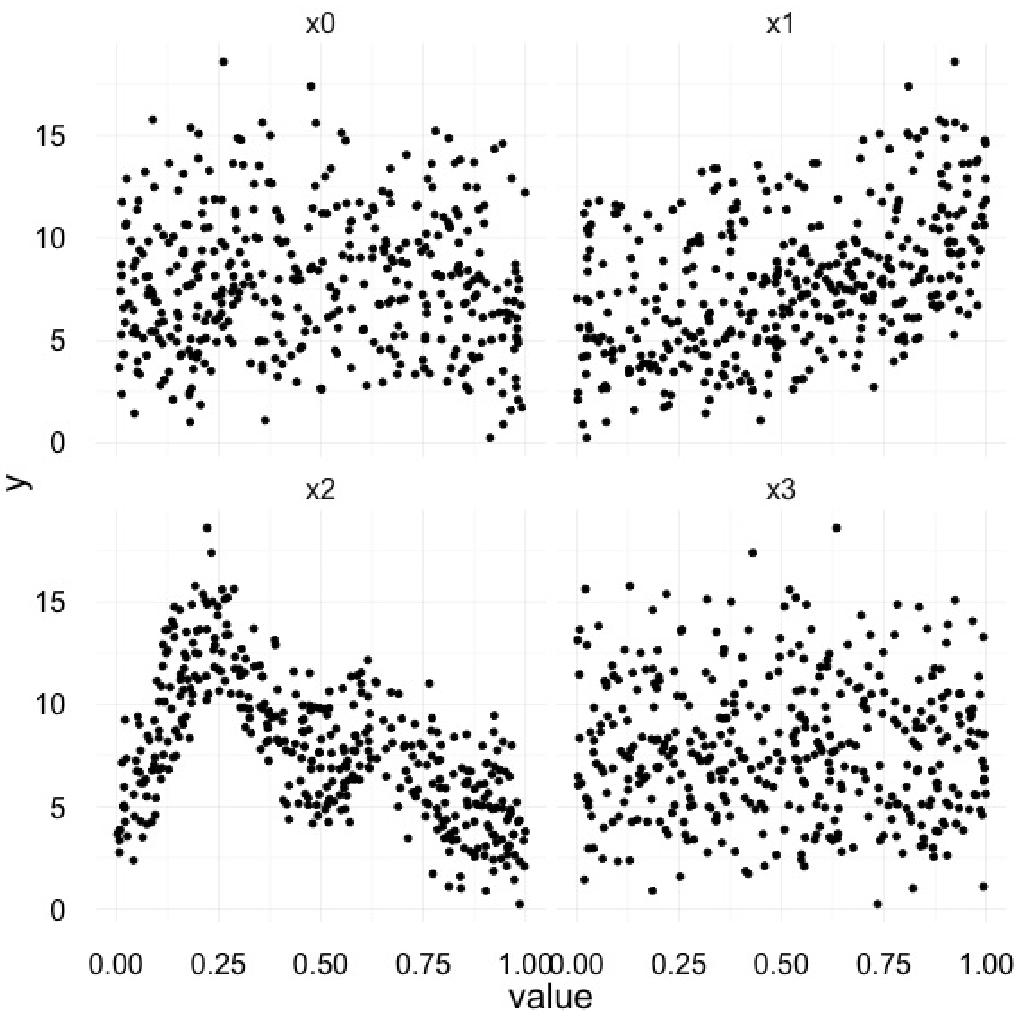
$$ y_i = \beta_0 + \sum_j s_j(x_{ji}) + \epsilon_i $$

where $\epsilon_i \sim N(0, \sigma^2)$, $y_i \sim \text{Normal}$ (for now)

Remember that we're modelling the **mean** of this distribution!

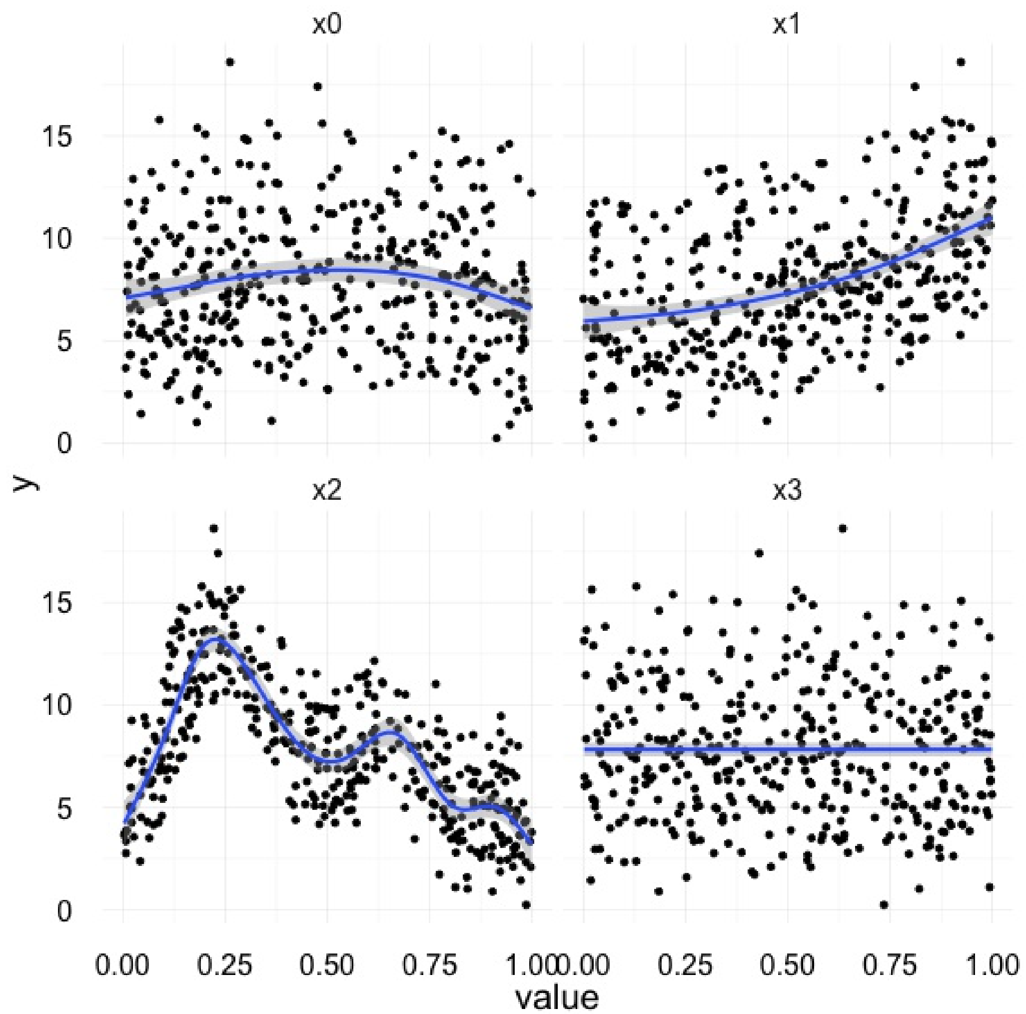
Call the above equation the **linear predictor**

Okay, but what about these ”s” things?



- Think \(s \)=**smooth**
- Want to model the covariates flexibly
- Covariates and response not necessarily linearly related!
- Want some “wiggles”

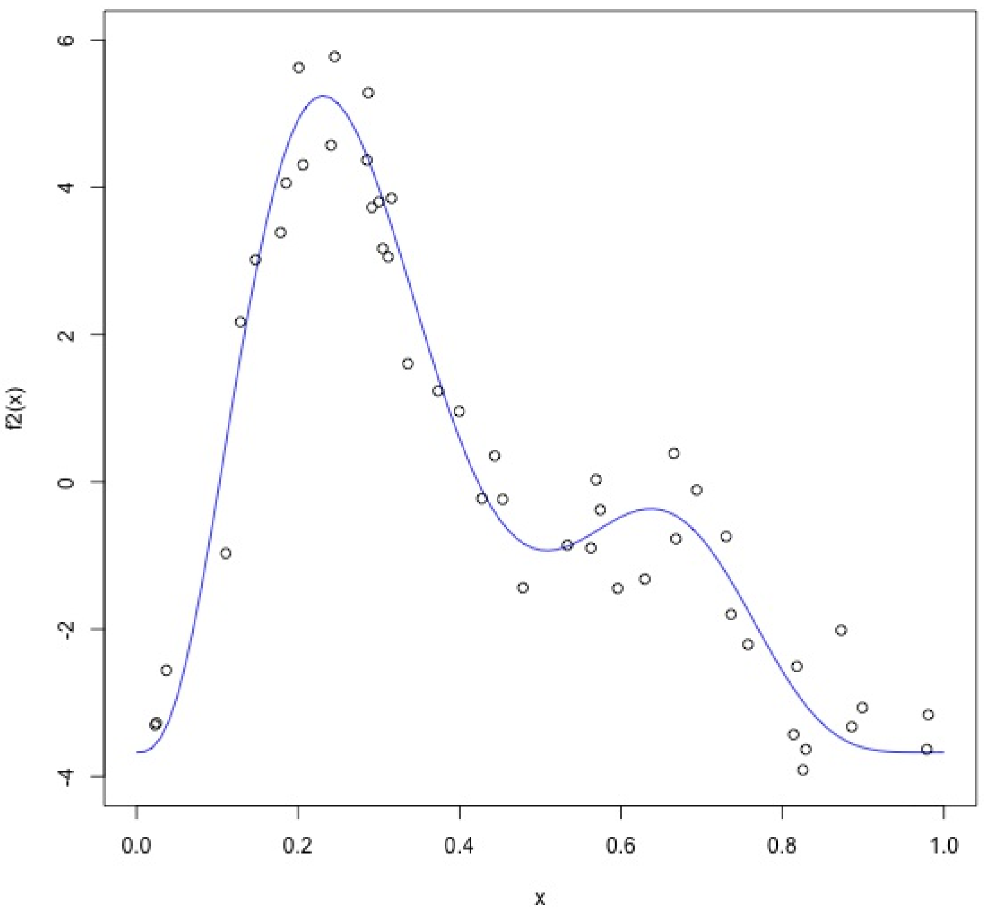
Okay, but what about these ”s” things?



- Think \(s \)=**smooth**
- Want to model the covariates flexibly
- Covariates and response not necessarily linearly related!
- Want some “wiggles”

What is smoothing?

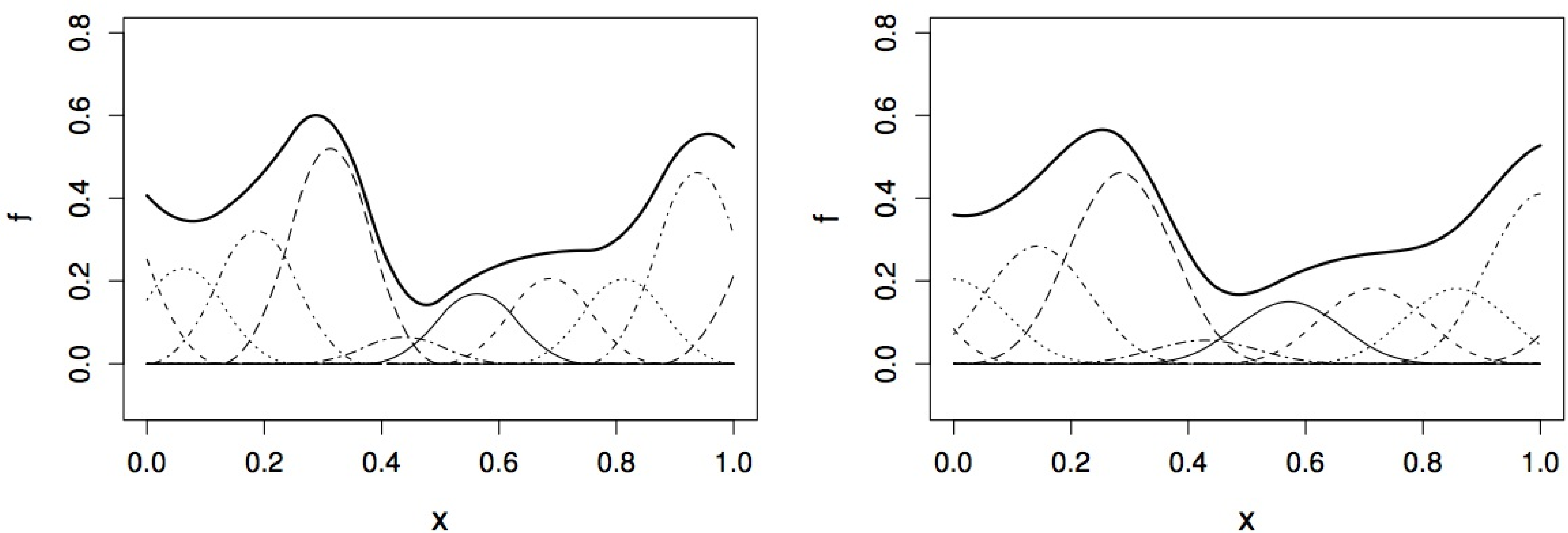
Straight lines vs. interpolation



- Want a line that is “close” to all the data
- Don't want interpolation – we know there is “error”
- Balance between interpolation and “fit”

Splines

- Functions made of other, simpler functions
- **Basis functions** $b_k(x)$, estimate β_k
- $s(x) = \sum_{k=1}^K \beta_k b_k(x)$
- Makes the math(s) much easier



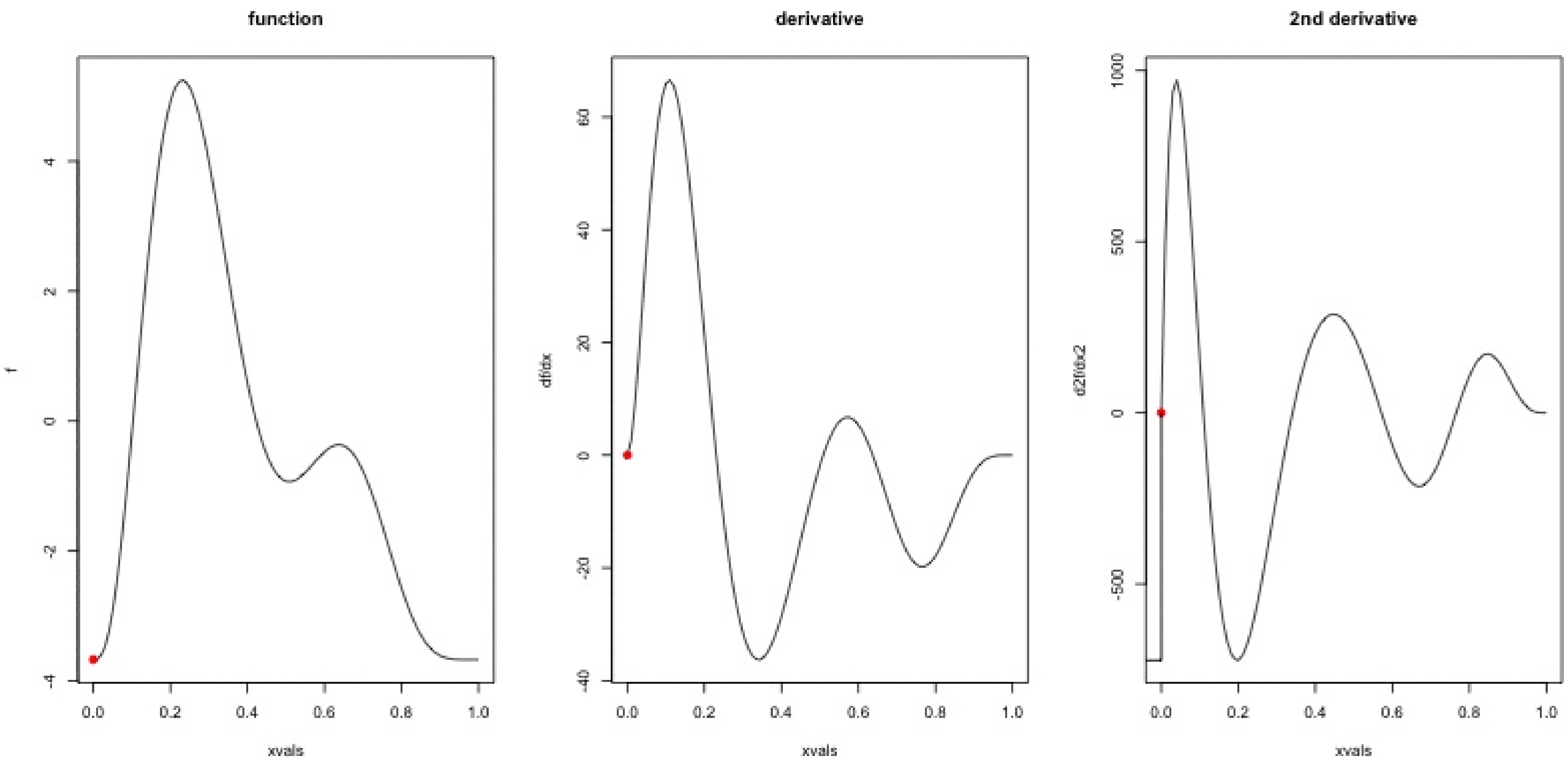
Design matrices

- We often write models as $X\boldsymbol{\beta}$
  - X is our data
  - $\boldsymbol{\beta}$ are parameters we need to estimate
- For a GAM it's the same
  - X has columns for each basis, evaluated at each observation (row)
  - again, this is the linear predictor

Measuring wigglyness

- Visually:
  - Lots of wiggles == NOT SMOOTH
  - Straight line == VERY SMOOTH
- How do we do this mathematically?
  - Derivatives!
  - (Calculus *was* a useful class afterall!)

Wigglyness by derivatives



What was that grey bit?

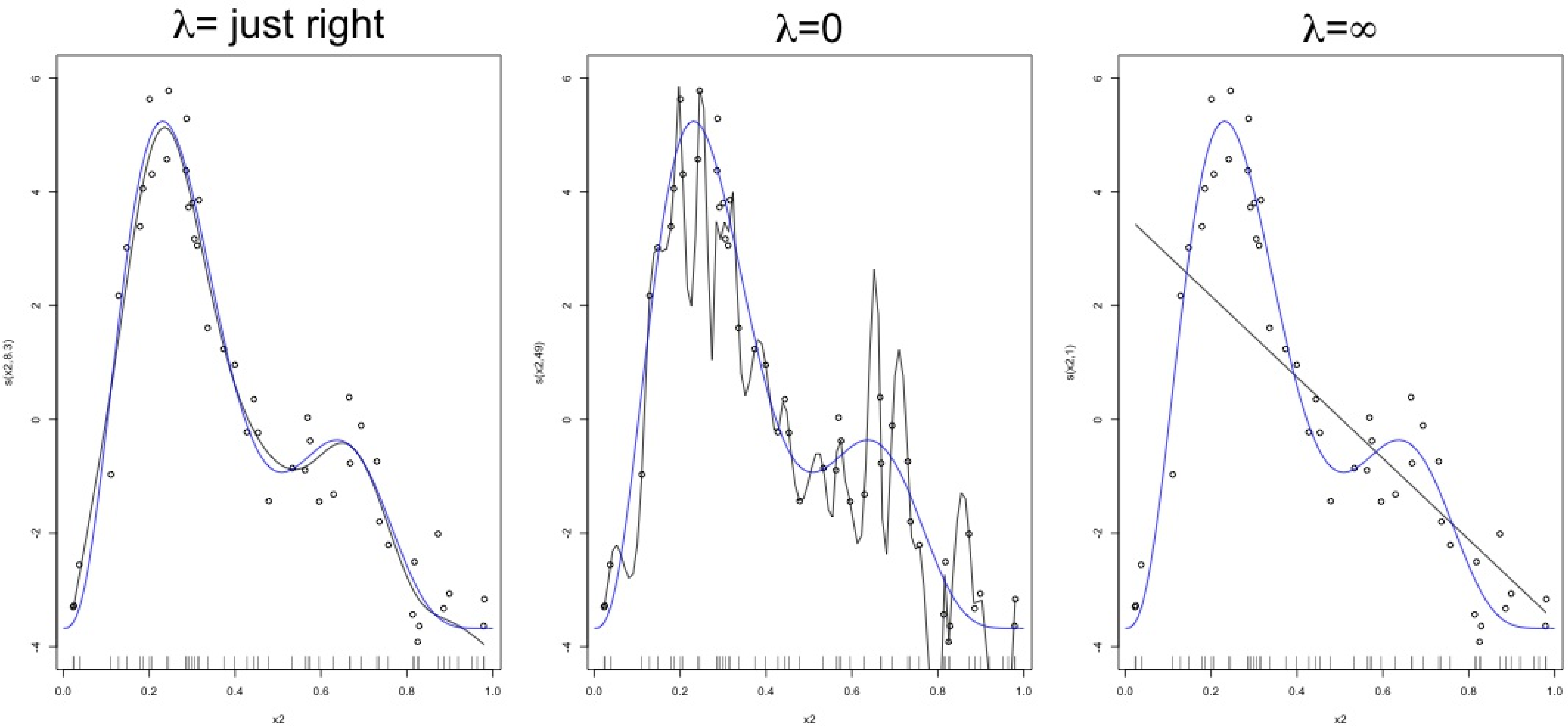
$$ \int_\mathbb{R} \left(\frac{\partial^2 f(x)}{\partial^2 x}\right)^2 \text{d}x\\ $$

- *Turns out* we can always write this as $\boldsymbol{\beta}^\text{T}S\boldsymbol{\beta}$, so the $\boldsymbol{\beta}$ is separate from the derivatives
- Call S the **penalty matrix**
- Different penalties lead to difference f s $\Rightarrow$ different $b_k(x)$ s

Making wigglyness matter

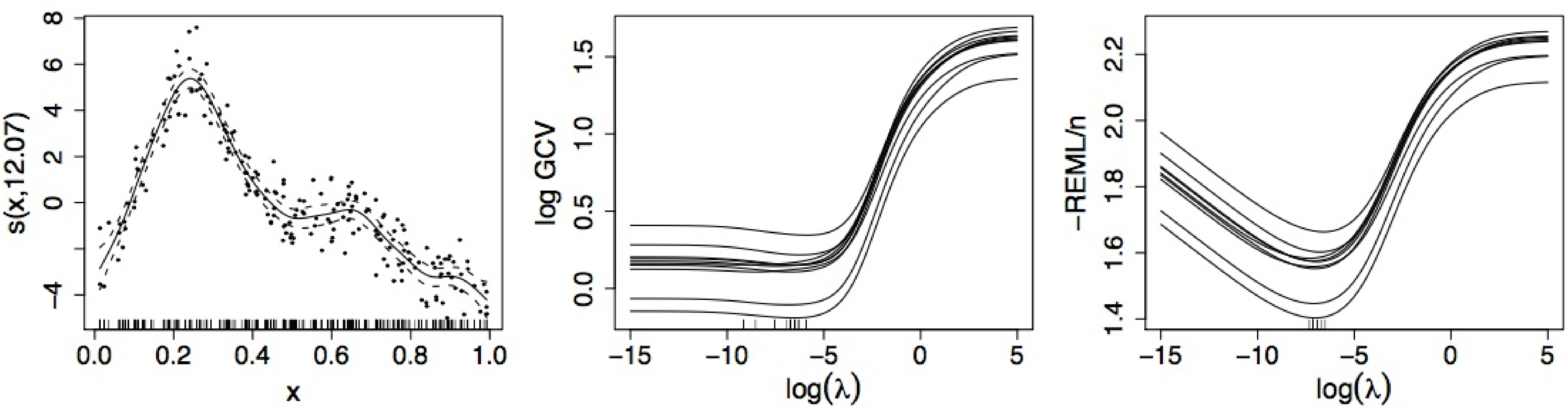
- $\boldsymbol{\beta}^\text{T}S\boldsymbol{\beta}$ measures wigglyness
- "Likelihood" measures closeness to the data
- Penalise closeness to the data…
- Use a **smoothing parameter** to decide on that trade-off…
  - $\lambda \boldsymbol{\beta}^\text{T}S\boldsymbol{\beta}$
- Estimate the β_k terms but penalise objective
  - "closeness to data" + penalty

Smoothing parameter



Smoothing parameter selection

- Many methods: AIC, Mallow's \(C_p \), GCV, ML, REML
- Recommendation, based on simulation and practice:
  - Use REML or ML
  - Reiss & Ogden (2009), Wood (2011)



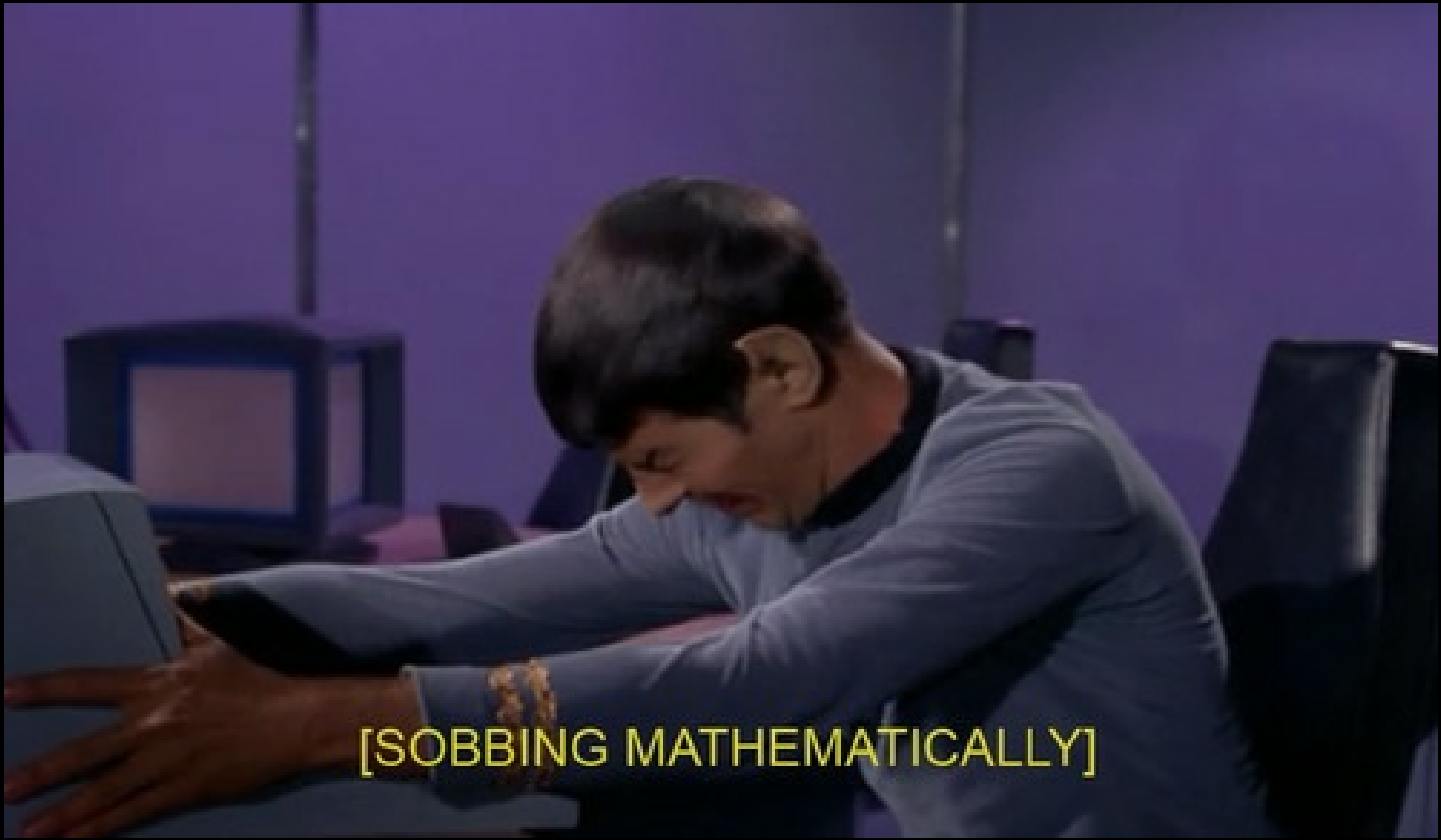
Maximum wiggliness

- We can set **basis complexity** or “size” (k)
  - Maximum wigglyness
- Smooths have **effective degrees of freedom** (EDF)
- EDF < k
- Set k “large enough”
  - Penalty does the rest

More on this in a bit…

Response distributions

- Exponential family distributions are available
- Normal, Poisson, binomial, gamma, quasi etc (`?family`)
- Tweedie and negative binomial
- Plus more! (More on that in a bit)


[SOBBING MATHEMATICALLY]

GAM summary

- Straight lines suck — we want **wiggles**
- Use little functions (**basis functions**) to make big functions (**smooths**)
- Need to make sure your smooths are **wiggly enough**
- Use a **penalty** to trade off wiggliness/generality