

# Generalized Additive Models

An introduction

Andreas Scharmüller

Quantitative Landscape Ecology  
University Koblenz-Landau

30.6.2019 (updated: 2019-06-30)

# Short intro: Andreas Scharmüller

- PhD student Quantitative Landscape Ecology
- Environmental Sciences + Ecotoxicology
- Research
  - Effects and distribution of pesticides in freshwaters
  - Big(ish) ecotoxicological data processing



Talk: Standartox - A tool for assessing the risk of chemicals  
Ecotoxicology and stress responses: Monday, 14:45 (Crystall ballroom 2G)

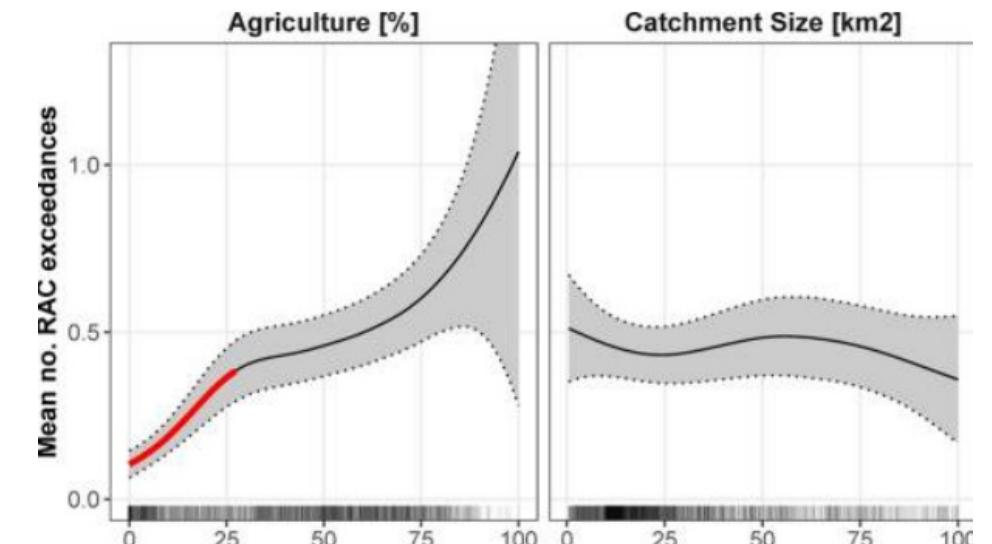
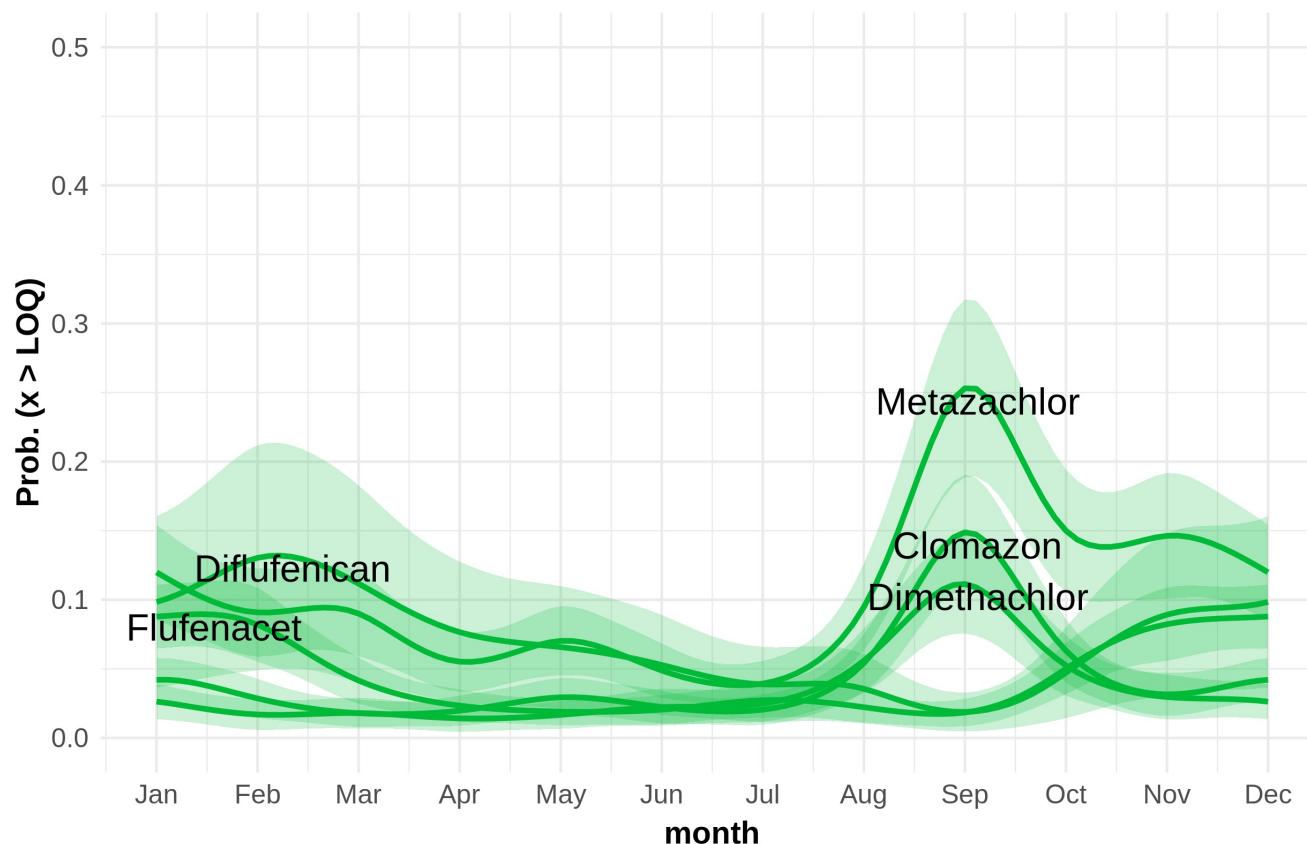
- Teaching:
  - Statistics
  - GIS
- R programming
  - Package author: standartox (in preparation)
  - Package contributions: webchem

# Research

- Seasonal occurrence patterns of **pesticides** in small waterbodies in Germany
- Modeling **benthic diatom** abundance and occurrence patterns in freshwater

Szöcs et al. 2017

Seasonal change in the occurrence of herbicides  
pre-emergence herbicides



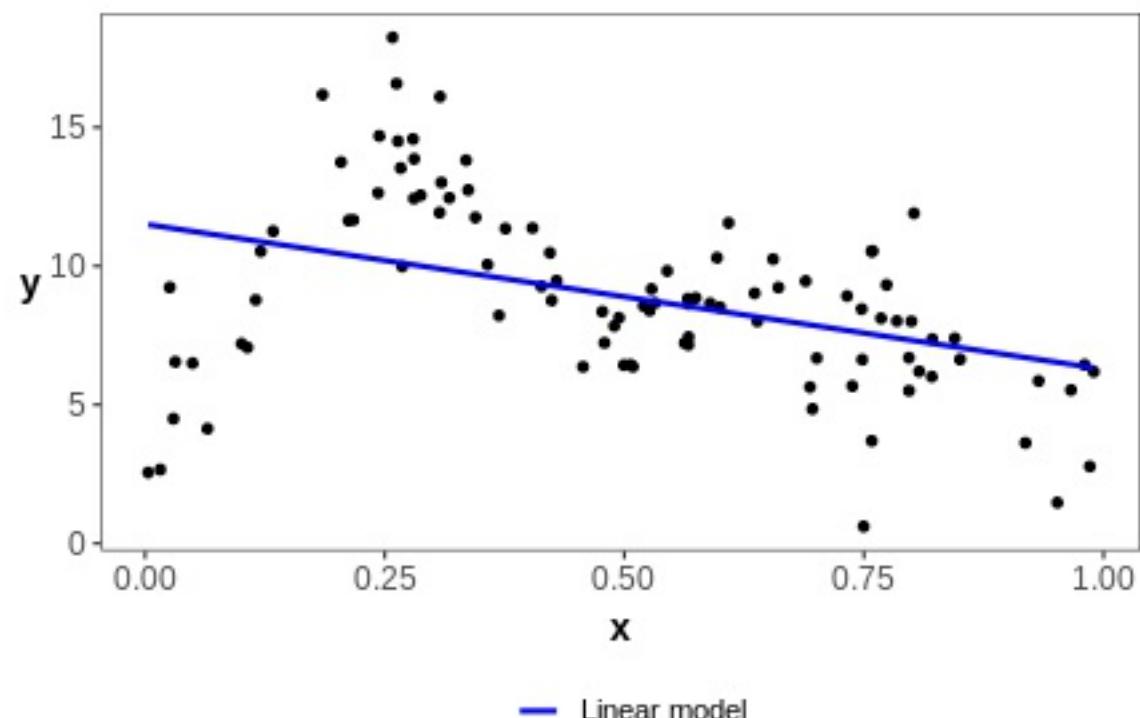
# Linear Regression

# Linear Models (LM)

- easy to interpret
- confined to linear relationships
- normally distributed responses

$$y = \beta_0 + x_1\beta_1 + \epsilon, \epsilon \sim N(0, \sigma^2)$$

```
lm(y ~ x,  
  data = data)
```

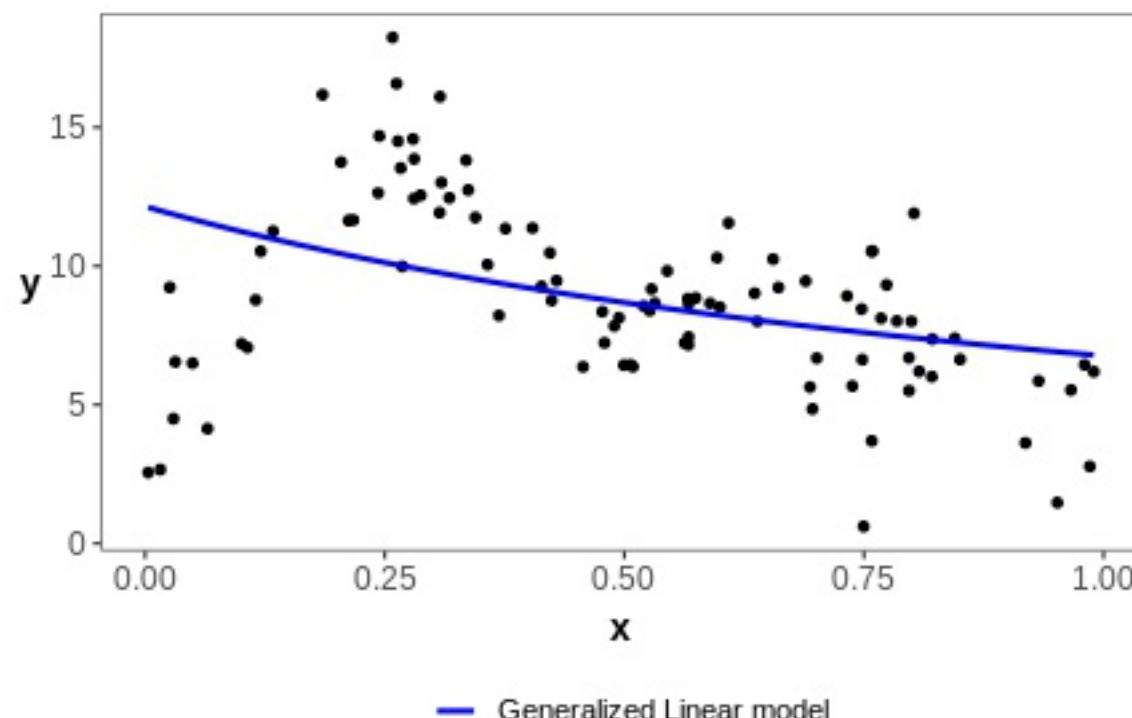


# Generalized Linear Models (GLM)

- additional distributions (Poisson, Gamma, Binomial, etc.)

$$E(y) = \beta_0 + x_1\beta_1 + x_2\beta_2 + \epsilon$$

```
glm(y ~ x,  
    data = data,  
    family = 'Gamma')
```

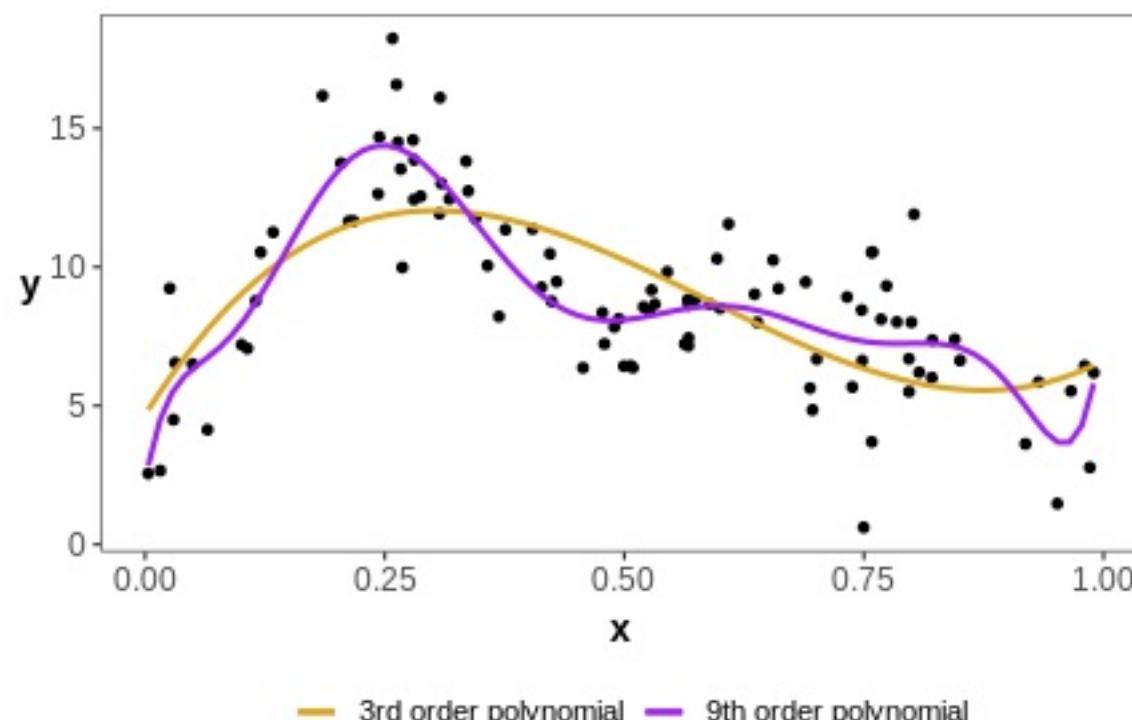


# Polynomial regression

- specific patterns, not very flexible
- might lead to poor residuals, predictions, extrapolations

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_1^3 + \epsilon, \epsilon \sim N(0, \sigma^2)$$

```
lm(y ~ poly(x, 3),  
    data = data)
```



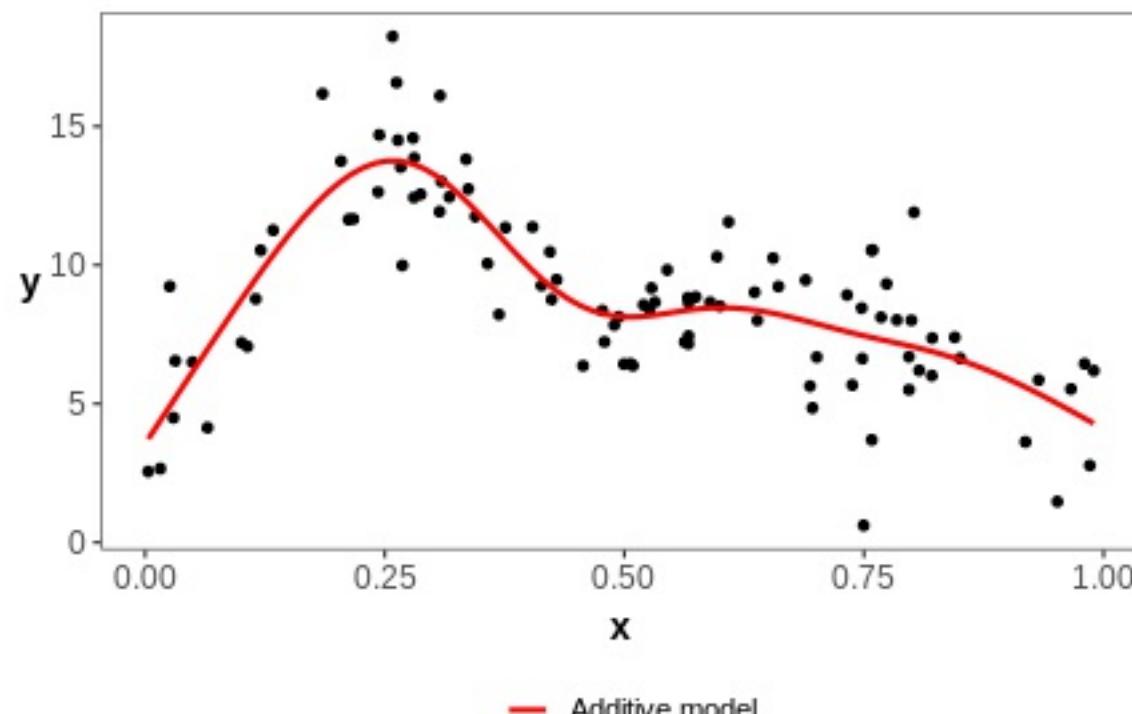
# Generalized Additive models (GAM)

# Generalized Additive Models (GAM)

- extension to GLMs
- more flexible
- can be harder to interpret

$$E(y) = \beta_0 + f_1(x_1) + x_2\beta_2 + \epsilon$$

```
mgcv:::gam(y ~ s(x1) + x2,  
            data = data,  
            family = 'gaussian')
```

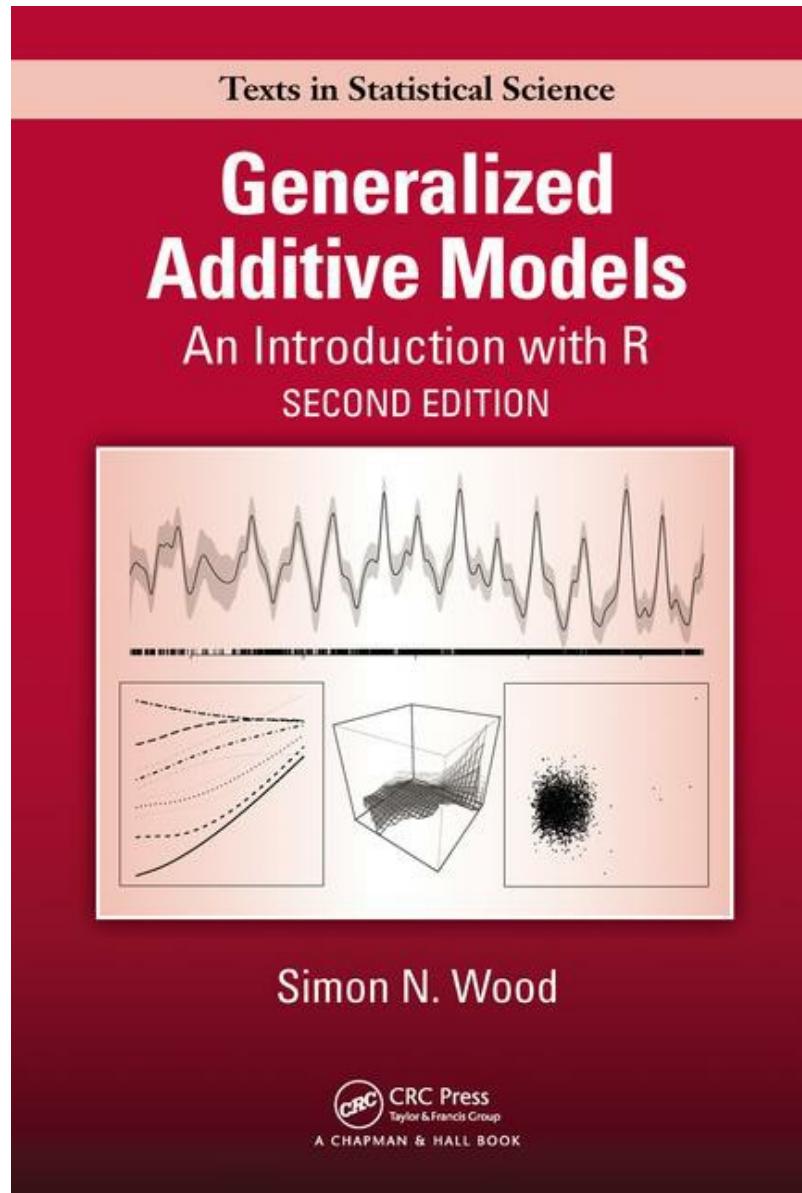


# GAMs in R

```
require(mgcv) https://cran.r-project.org/web/packages/mgcv/index.html
require(gamlss) https://cran.r-project.org/web/packages/gamlss/index.html
require(brms) https://github.com/paul-buerkner/brms
```

# GAMs in R

`require(mgcv)` <https://cran.r-project.org/web/packages/mgcv/index.html>



# GAMs in R

R-function:

```
require(mgcv)

gam(y ~ s(x1, bs = 'tp', k = -1) + x2, # formula
    data = data, # data
    family = 'gaussian', # family object
    method = 'REML', # default: 'GCV.Cp'
    sp = NULL) # smoothing parameter
```

# GAMs in R

Formula:

```
require(mgcv)

# gam(y ~
#       s(# smooth term s(), te()
#           x1, # predictor
#           bs = 'tp', # spline basis
#           k = -1, # number of basis functions
#           sp = NULL # smoothing parameter
#       )
#       + x2, # linear term
#       data = data,
#       family = 'gaussian',
#       method = 'REML' # default: 'GCV.Cp'
#       sp = NULL)
```

# Smoothing

# Smoothing

$$Fit = Likelihood - \lambda \times Wigginess$$

- Likelihood: How well a GAM captures patterns in the data
- Wigginess: Complexity of a smooth
- $\lambda$  is optimized in `gam()` and controls the trade-off between Likelihood and Wigginess

taken from: <https://noamross.github.io/gams-in-r-course>

# Smoothing

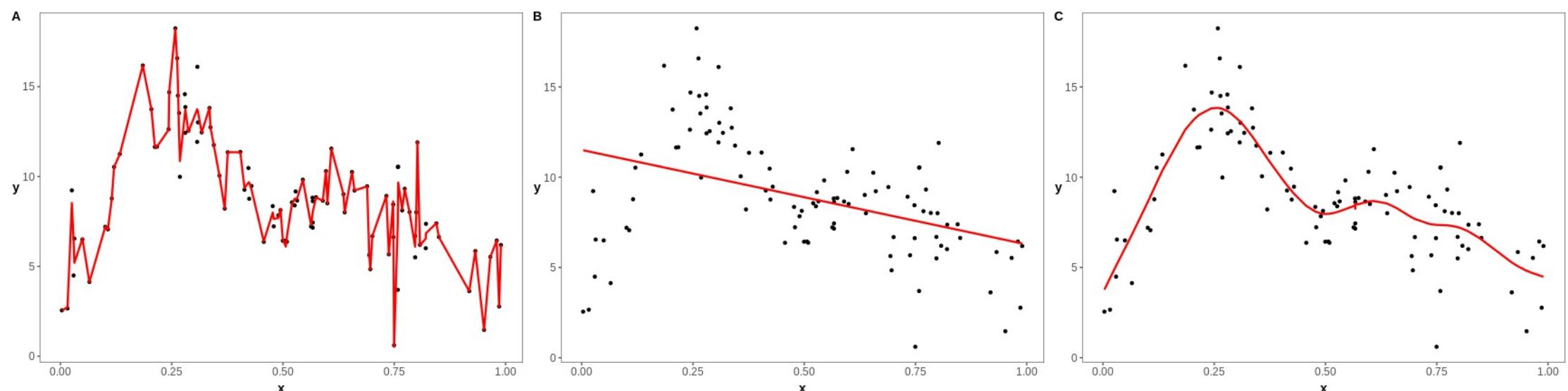
## Smoothing parameter

```
gam(y ~ s(x1, sp = NULL),  
     data = data,  
     sp = NULL,  
     method = 'REML')
```

- smootghin paramter
- estimated in REML

# Smoothing

## Smoothing parameter

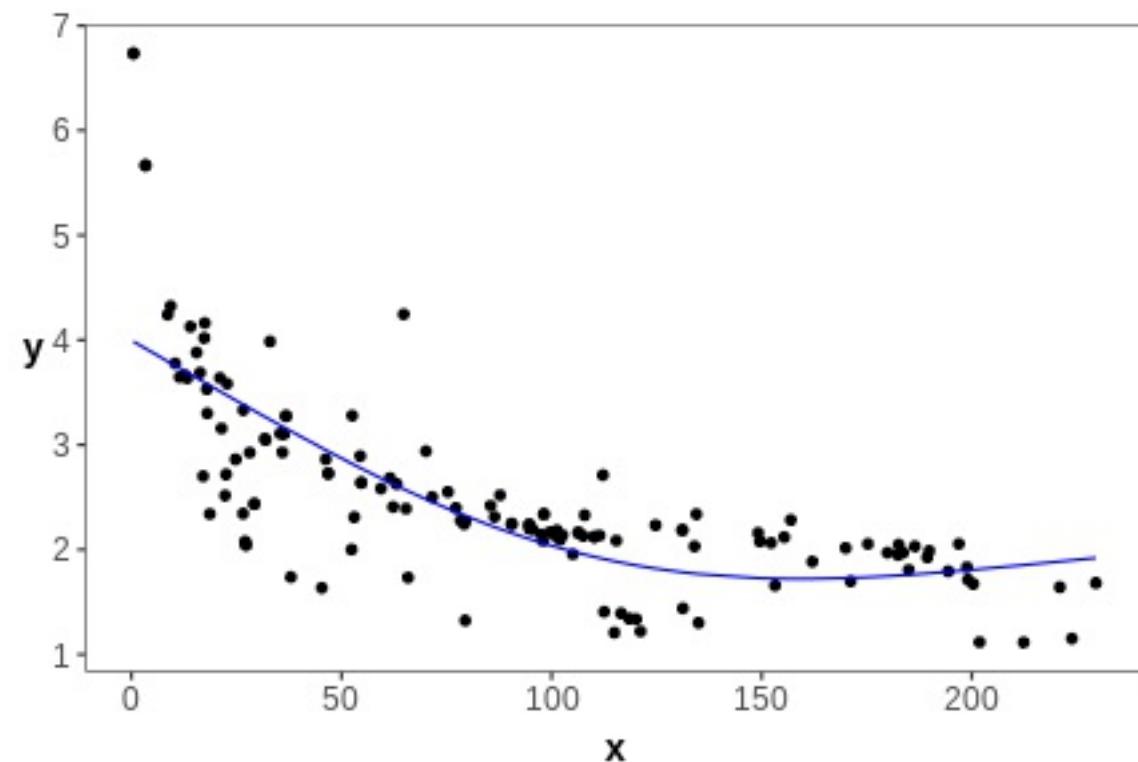


- A:  $sp = 0$
- B:  $sp = 1e5$
- C:  $method = 'REML'$

# Smoothing

## Basis Functions

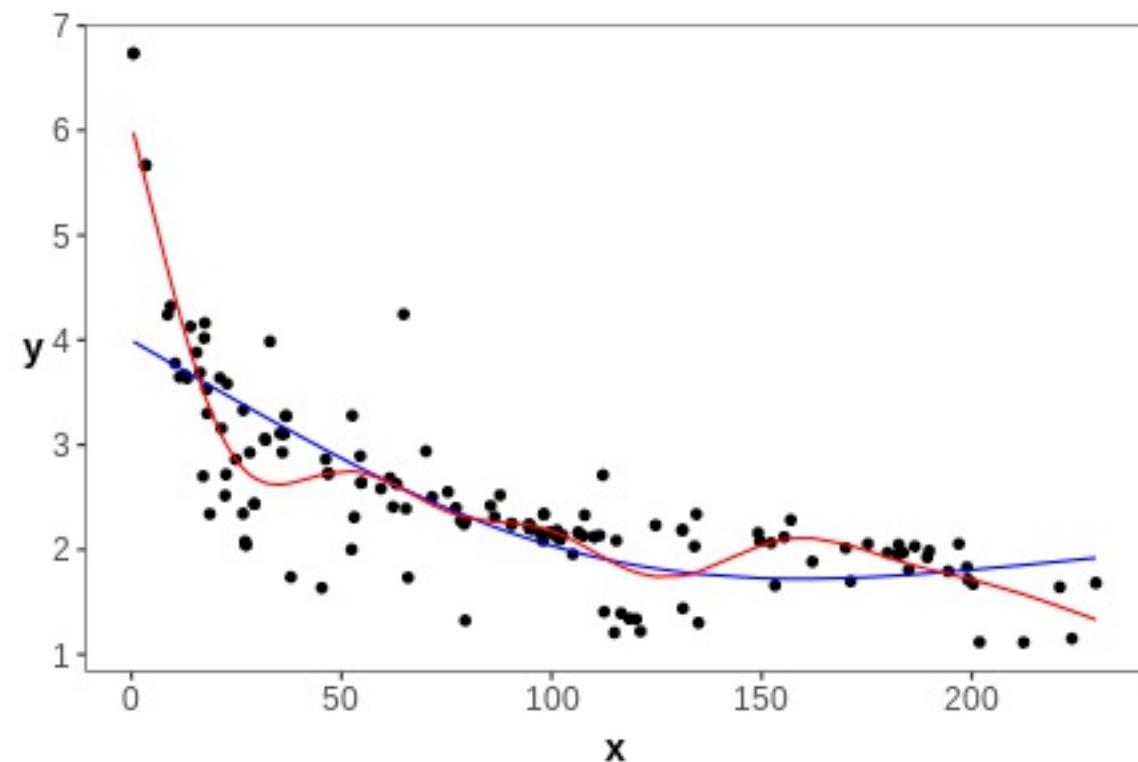
```
gam(y ~ s(x1, k = 3),  
     data = data)
```



# Smoothing

## Basis Functions

```
gam(y ~ s(x1),  
     data = data)
```



# Smoothing

## Basis Functions

Check for k:

```
gam.check(oc1)
```

```
k.check(oc1)
```

```
##          k'      edf  k-index p-value
## s(P)    2 1.972523 0.419655      0
```

If p-value is low, k might be too small

# Smooth terms

# Smooth terms

Two additive smooths

```
gam(y ~ s(x1) + s(x2))
```

Smooth-interactions

```
gam(y ~ s(x1, x2)) # common way to declare spatial data  
gam(y ~ s(x1, by = fac))
```

Tensor product smooths

```
gam(y ~ te(x1, x2, k = c(4,8))) # interaction on different scales
```

# Splines

# Splines

## Thin plate regression spline

- default

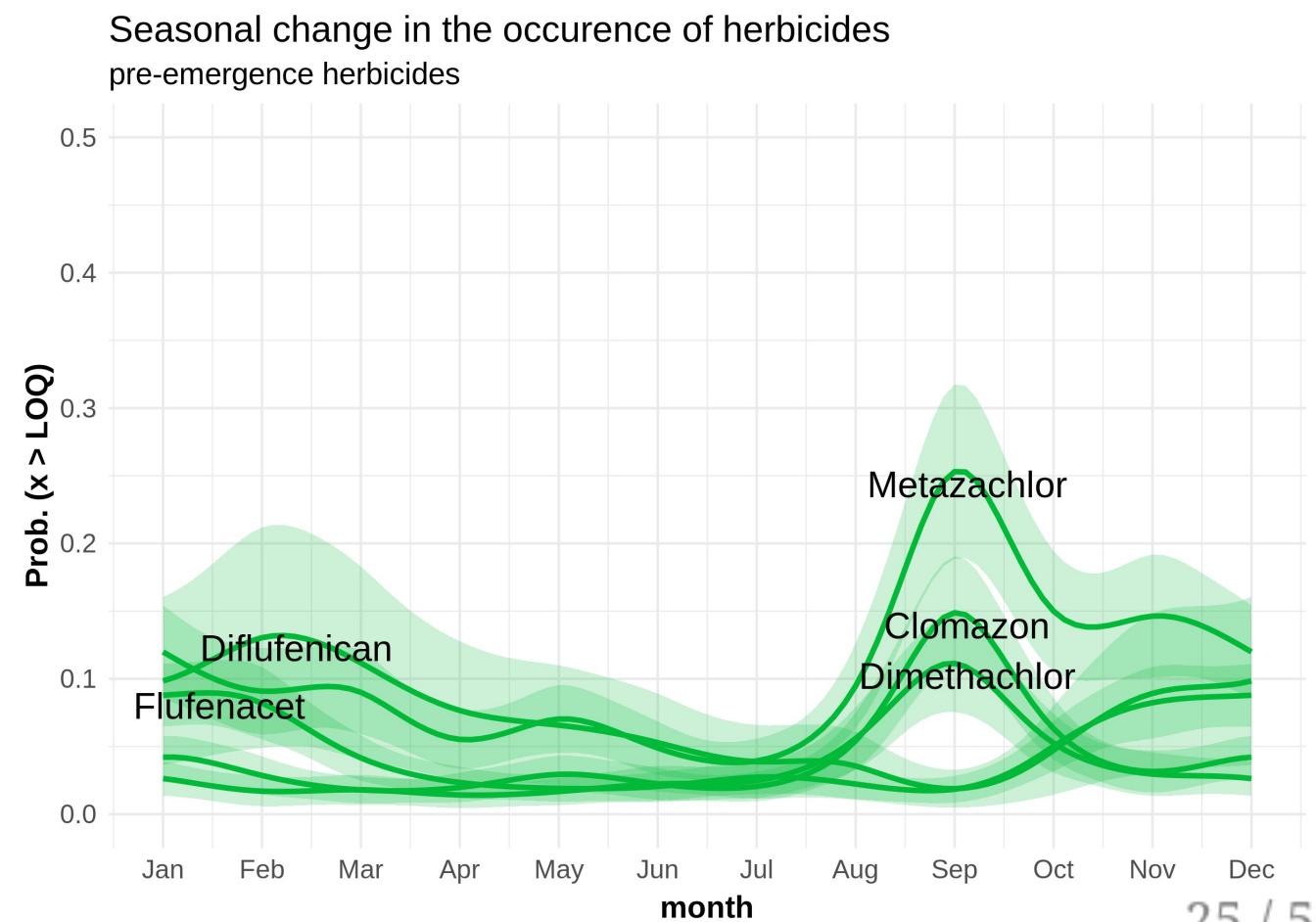
```
gam(y ~ s(x, bs = 'tp'),  
     data = data)
```

# Splines

## Cyclic cubic regression splines

- for cyclical data (e.g. seasons)

```
gam(y ~ s(x, bs = 'cc'),  
     data = data)
```



# Splines

## Soap Films

- boundary polygons can be introduced
- spatial models

```
gam(y ~ s(x, y, bs = 'so', xt = list(bnd = boundary_polygon)),  
     data = data)
```

# Splines

## Discrete random effects

- classes (e.g. age, sex)
- sites, states, rivers, lakes
- no need to set k (equals number of levels)

```
gam(y ~ s(x) + s(fac, bs = 're'),  
     data = data)
```

# What are GAMs?

<https://www.fromthebottomoftheheap.net>

140 char vrsn

- 1 GAMs are just GLMs
- 2 GAMs fit wiggly terms
- 3 use + s(foo) not foo in frmla
- 4 use method = "REML"
- 5 gam.check()

— Dr Gavin Simpson (@ucfagsl) [16 march 2017](#)

# Visualisation

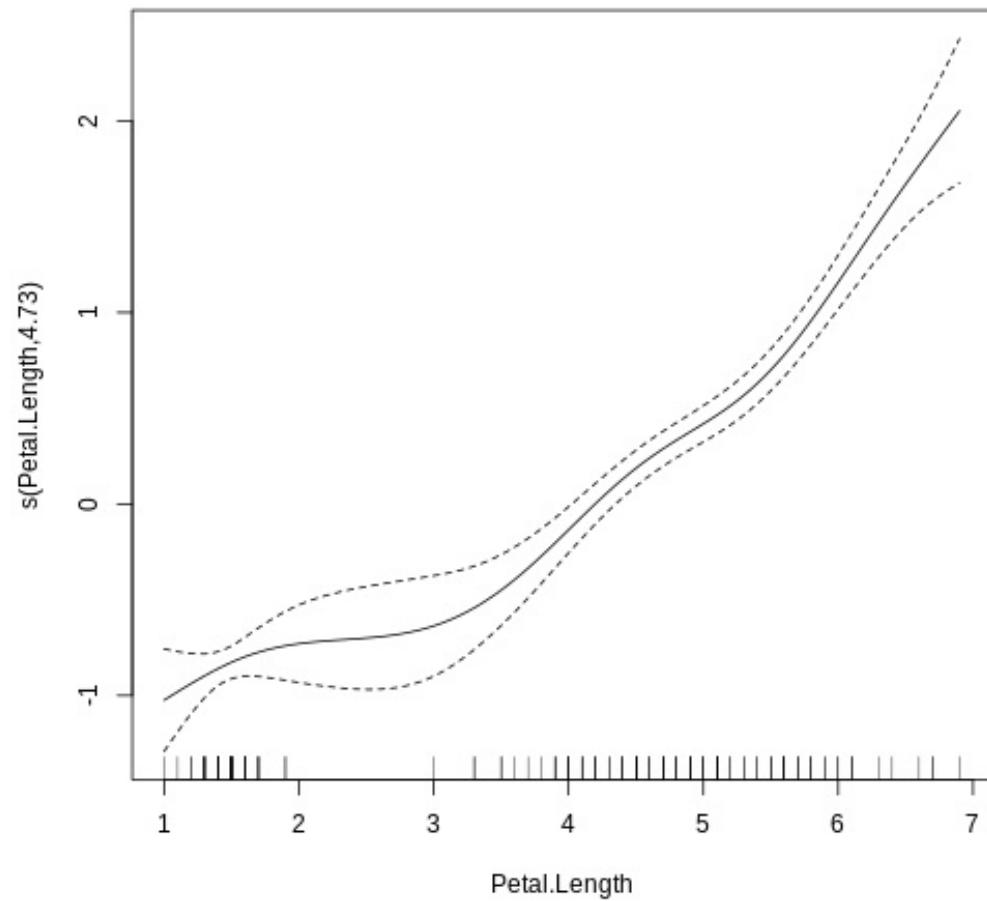
# Visualisation

```
ir1 = gam(Sepal.Length ~ s(Petal.Length),  
          data = iris,  
          family = 'gaussian',  
          method = 'REML')
```

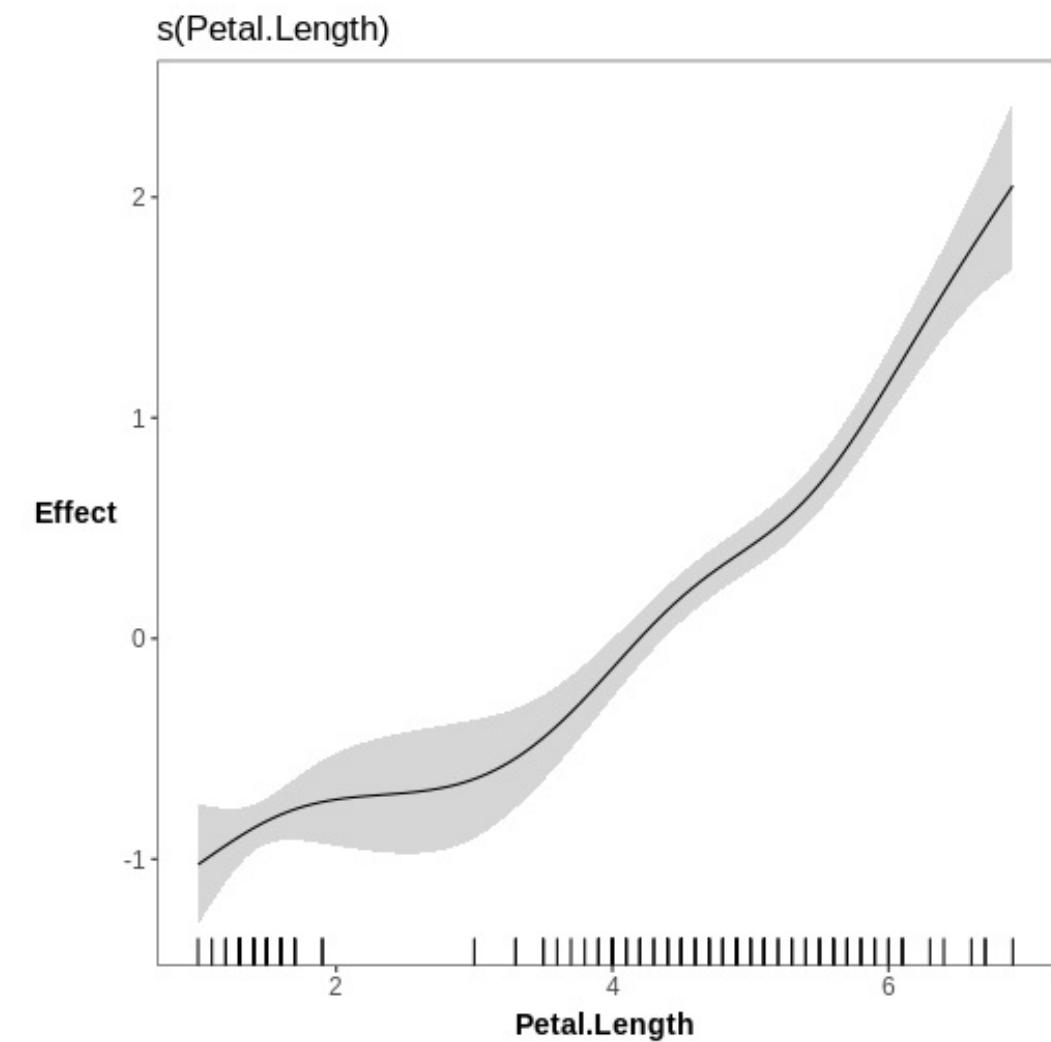
# Visualisation

## Partial effect plots

```
require(mgcv)  
plot(ir1, pages = 1)
```



```
require(gratia)  
draw(ir1)
```



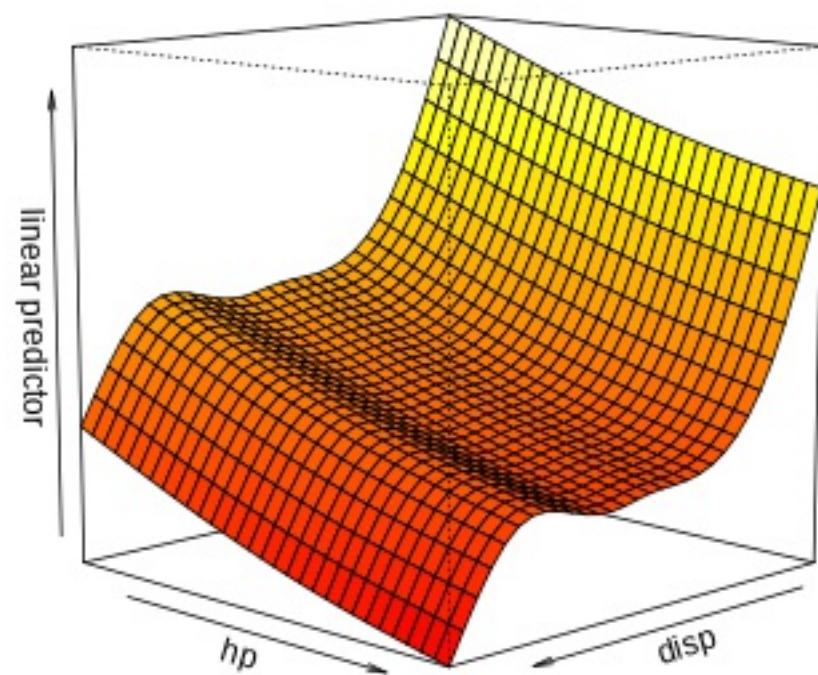
# Visualisation

## Multiple covariates

```
mt1 = gam(mpg ~ s(disp) + s(hp),  
          data = mtcars,  
          family = 'gaussian',  
          method = 'REML')
```

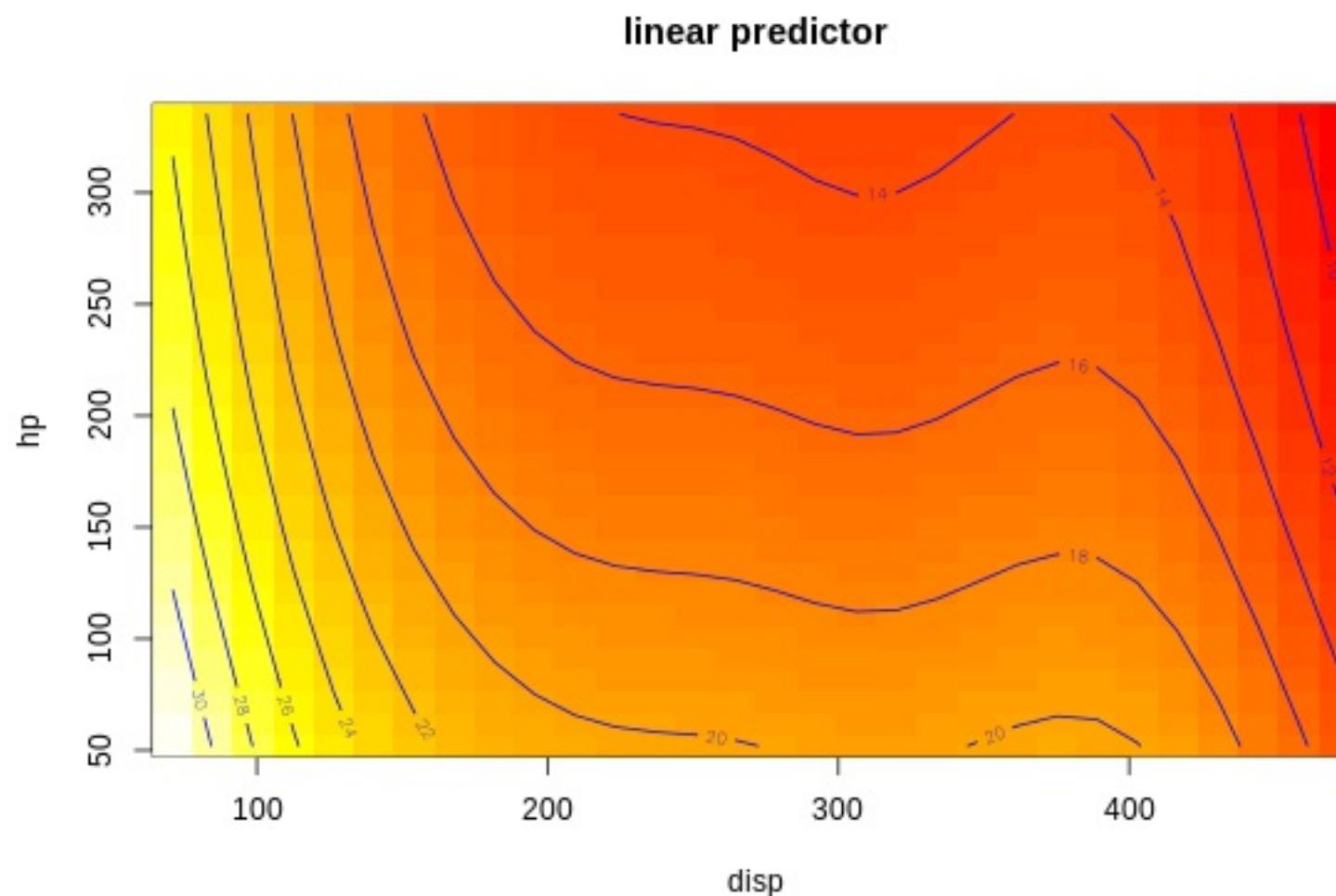
# Visualisation

```
vis.gam(mt1, # GAM object
        view = c("disp", "hp"), # variables
        plot.type = "persp", # 3D plot
        theta = 135, # horizontal rotation
        phi = 10, # phi vertical rotation
        r = 10) # zoom
```



# Visualisation

```
vis.gam(mt1, # GAM object  
        view = c("disp", "hp"), # variables  
        plot.type = "contour") # contour plot or heatmap
```



# Exercise 1

# Model output

# Model output

## Summary

```
summary(mod)
```

## Checking

```
gam.check(mod)  
k.check(mod)
```

## AIC

```
AIC(mod)
```

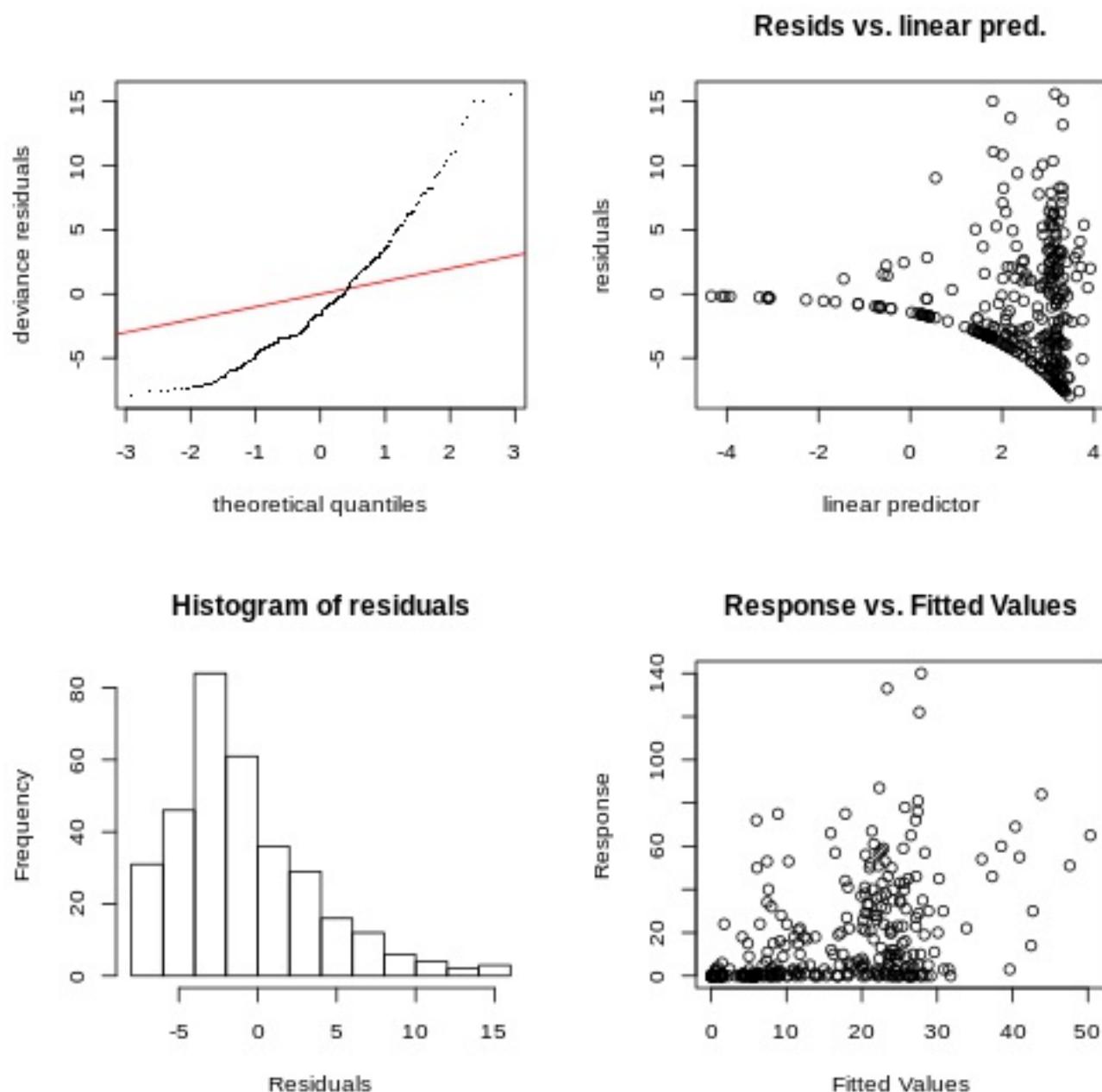
## Predict

```
predict(mod)
```

# Model summary

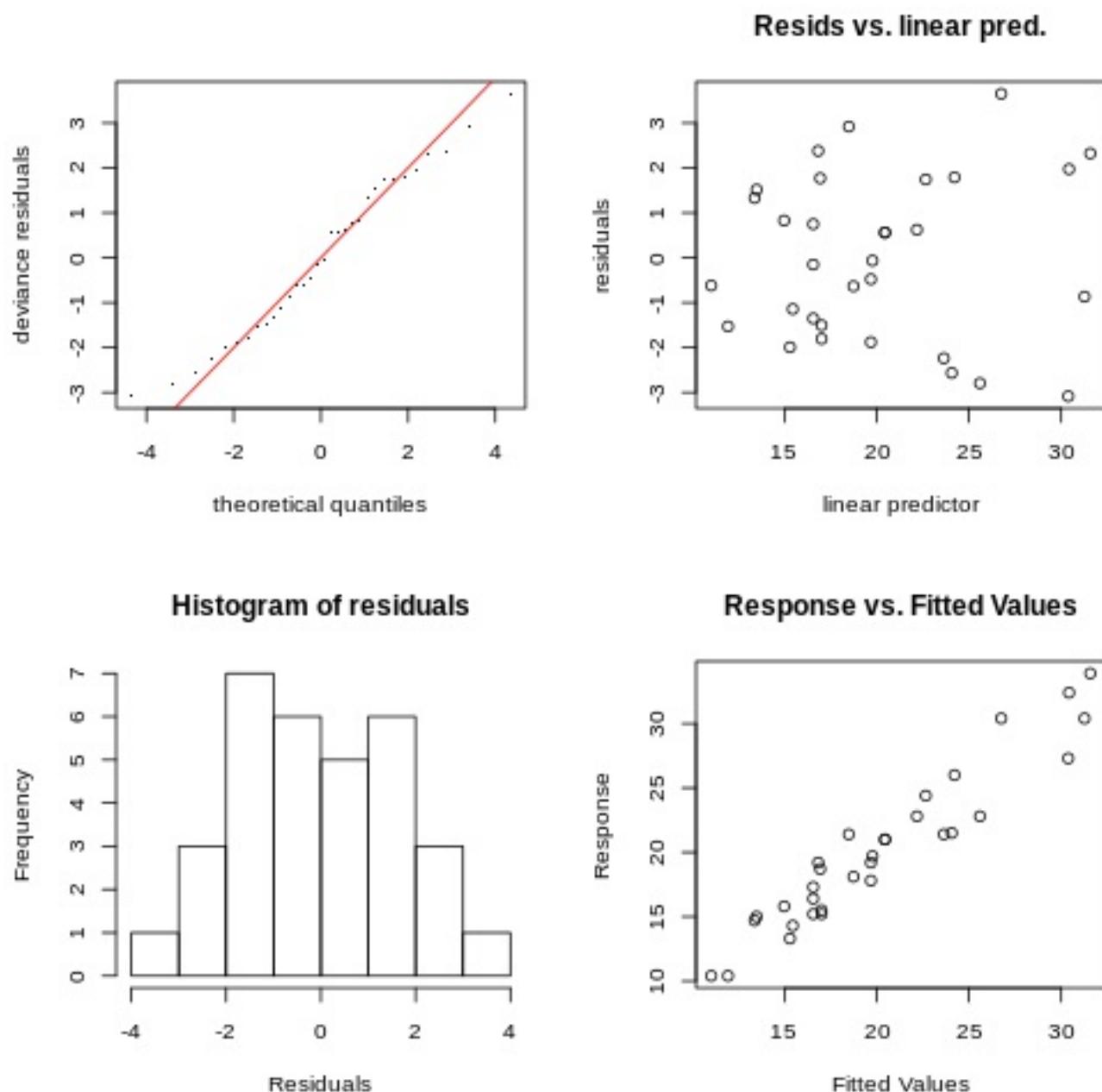
```
##  
## Family: poisson  
## Link function: log  
##  
## Formula:  
## egg.count ~ s(c.dist) + s(salinity)  
##  
## Parametric coefficients:  
##             Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 2.22458   0.04077  54.57   <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Approximate significance of smooth terms:  
##             edf Ref.df Chi.sq p-value  
## s(c.dist)    8.559  8.938 255.8  <2e-16 ***  
## s(salinity)  8.886  8.992 1204.7 <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## R-sq.(adj) =  0.165  Deviance explained = 28.9%  
## UBRE = 19.928  Scale est. = 1           n = 330
```

# Model checking



```
##  
## Method: UBRE    Optimizer: outer newton
```

# Model checking



```
##  
## Method: REML    Optimizer: outer newton
```

# Hierachical GAMs

# Hierachical GAMs

- Hierachical data: grouped data (factor!)
- non-linear relationships
- shape of function can vary between grouping/factor levels



## **Hierarchical generalized additive models in ecology: an introduction with mgcv**

Eric J. Pedersen<sup>1,2</sup>, David L. Miller<sup>3,4</sup>, Gavin L. Simpson<sup>5,6</sup> and  
Noam Ross<sup>7</sup>

<sup>1</sup> Northwest Atlantic Fisheries Center, Fisheries and Oceans Canada, St. John's, NL, Canada

<sup>2</sup> Department of Biology, Memorial University of Newfoundland, St. John's, NL, Canada

<sup>3</sup> Centre for Research into Ecological and Environmental Modelling, University of St Andrews,  
St Andrews, UK

<sup>4</sup> School of Mathematics and Statistics, University of St Andrews, St Andrews, Scotland, UK

<sup>5</sup> Institute of Environmental Change and Society, University of Regina, Regina, SK, Canada

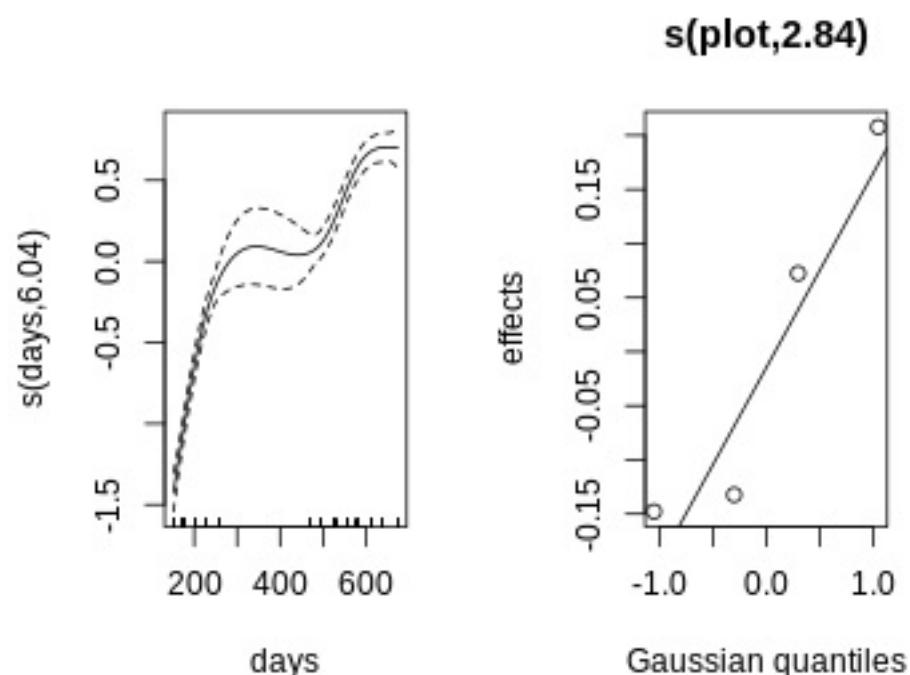
<sup>6</sup> Department of Biology, University of Regina, Regina, SK, Canada

<sup>7</sup> EcoHealth Alliance, New York, NY, USA

# Global Smoother

- single global smooth term for each variable
- level-individual **random effect intercepts** (`bs = 're'`)
- mixed models (like in lme4, nlme)

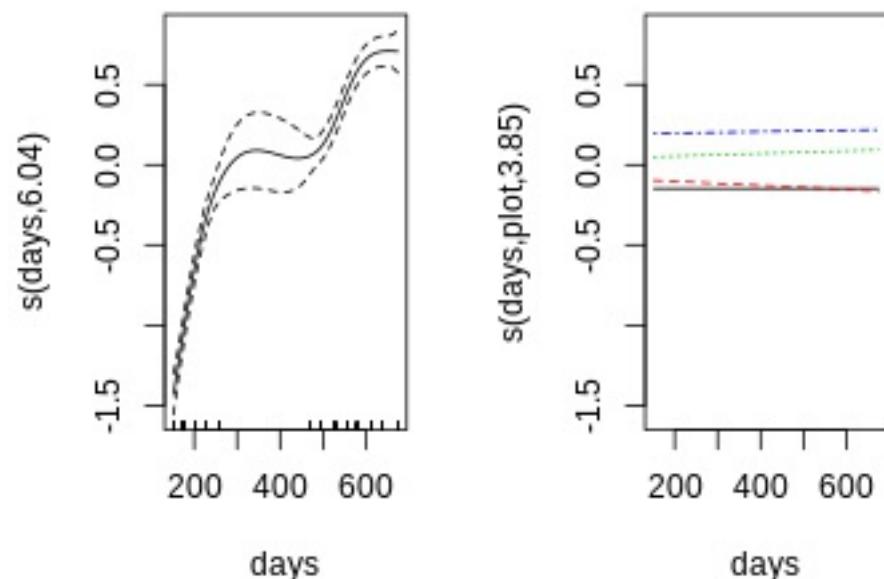
```
spr_G = gam(logSize ~ s(days) +
              s(plot, bs = 're'),
              data = Spruce,
              method = 'REML')
plot(spr_G, pages = 1)
```



# Global Smoother + Group Level Smoother

- factor-smooth interaction (`bs = 'fs'`)
- single global smooth term for each variable
- **varying slopes**
- same wiggliness (i.e. complexity) of smooths

```
spr_GS = gam(logSize ~ s(days) +
               s(days, plot, bs = 'fs'),
               data = Spruce,
               method = 'REML')
plot(spr_GS, pages = 1)
```



# Global Smoother + Group Level Smoother

- `s(x, by = fac)` and `s(fac, bs = 're')`
- similar to GS
- varying slopes
- **different wiggliness** (i.e. complexity) of smooths

```
spr_GI = gam(logSize ~ s(days) +
               s(days, by = plot) +
               s(plot, bs = 're'),
               data = Spruce,
               method = 'REML')
```

# Exercise 2

**Thank you for participating in the course!**

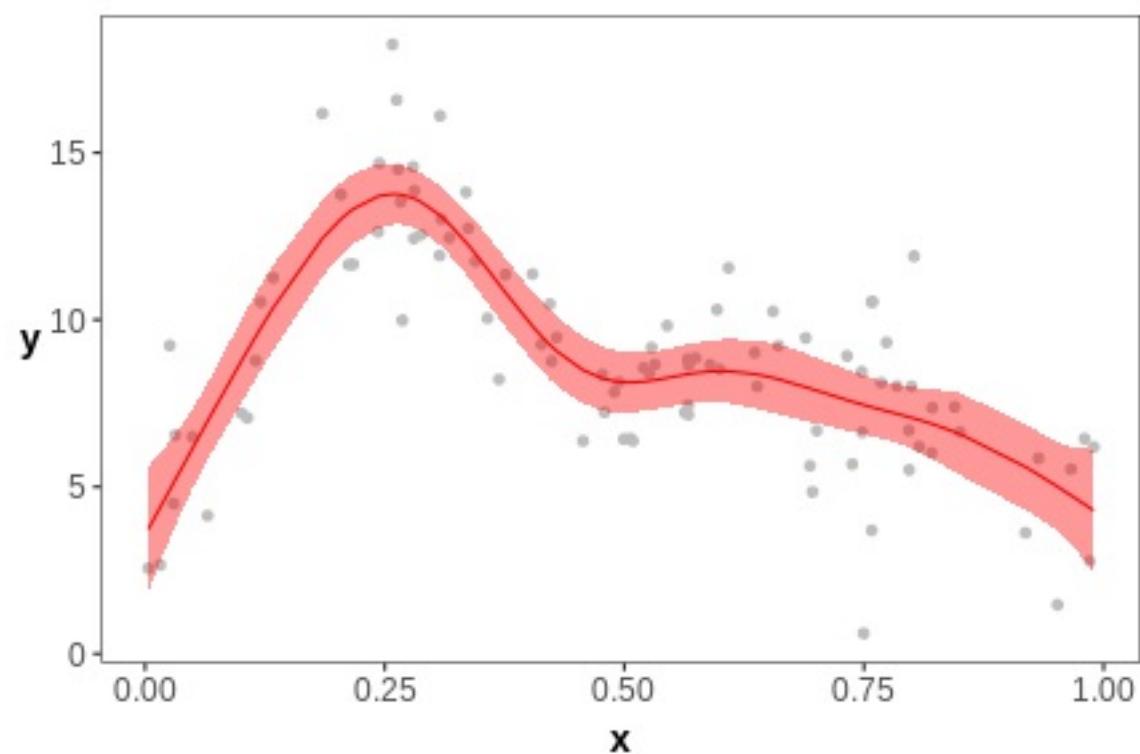
Structural Equation Modelling (Moritz Link) 15.00 - 16.30

# Tutorials, Blogs

- Generalized Additive Models in R (Noam Ross)
  - <https://noamross.github.io/gams-in-r-course>
- Doing magic and analyzing seasonal time series with GAM (Generalized Additive Model) in R
  - <https://petolau.github.io/Analyzing-double-seasonal-time-series-with-GAM-in-R/>
- From the Bottom of the Heap - Blog by Gavin Simpson
  - <https://www.fromthebottomoftheheap.net/>
- Introducing gratia
  - <https://www.fromthebottomoftheheap.net/2018/10/23/introducing-gratia/>
- Noam Ross Github
  - <https://github.com/noamross/gam-resources>
- Environmental computing
  - <http://environmentalcomputing.net/intro-to-gams>

# Prediction

```
gam_sim = gam(y ~ s(x),  
               data = sim,  
               family = 'gaussian')  
pred = predict(gam_sim, type = 'response', se.fit = TRUE)  
  
sim$fit = pred$fit  
sim$lwr = pred$fit - (2 * pred$se.fit)  
sim$upr = pred$fit + (2 * pred$se.fit)
```



# Model selection

If you are not interested in prediction, but the most parsimonious model

- penalty on overall slope (in addition to wigginess penalty)
- similar to ridge regression, lasso

```
select = TRUE
```

```
gam(y ~ s(x1) + s(x2) + s(x3),  
    select = TRUE,  
    data = data,  
    method = 'REML')
```

# Model selection

- AIC
- expert subject
- computational time
- inferential goals of the study