Non-linear modeling

Linear models:

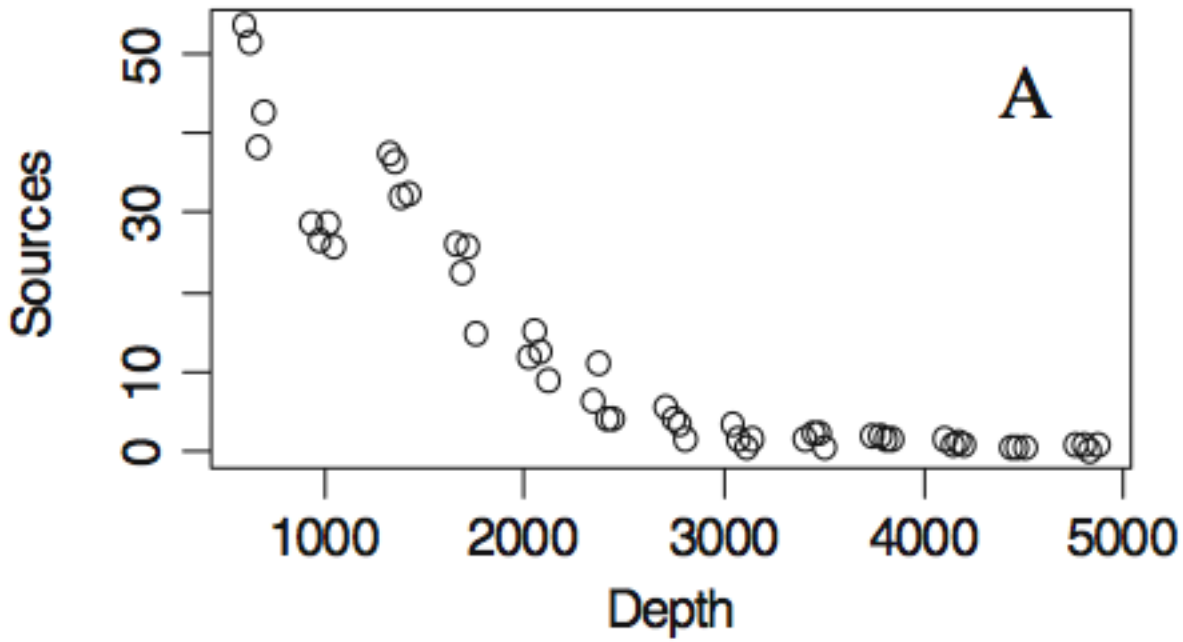
- $Y_i = \alpha + \beta_1 \times X_i + \beta_2 \times X_i^2 + \varepsilon_i$
- $Y_i = \alpha + \beta_1 \times \log(X_i) + \varepsilon_i$
- $Y_i = \alpha + \beta_1 \times (X_i \times W_i) + \varepsilon_i$
- $Y_i = \alpha + \beta_1 \times \exp(X_i) + \varepsilon_i$
- $Y_i = \alpha + \beta_1 \times \sin(X_i) + \varepsilon_i$

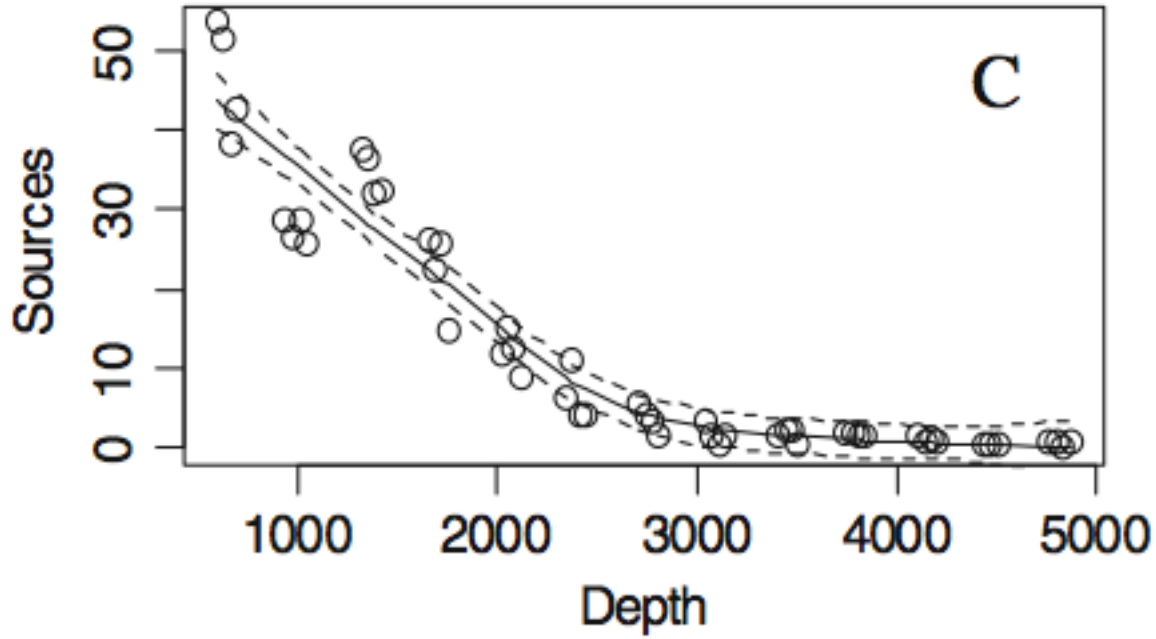
Non-linear models:

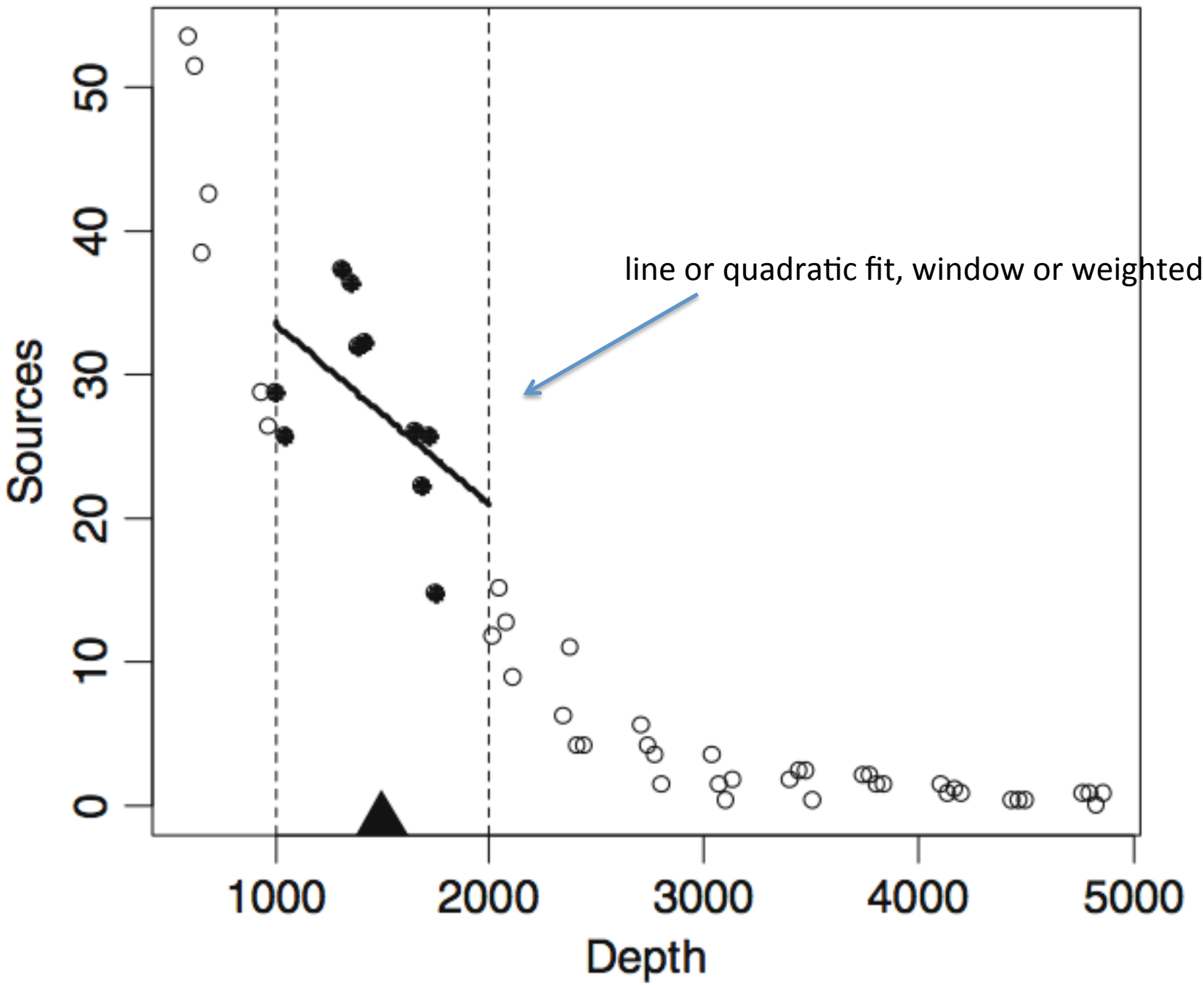
$$Y_i = \alpha + \beta_1 \times X_{1i} \times e^{\beta_2 \times X_{2i} + \beta_3 \times X_{3i}} + \varepsilon_i$$

pelagic bioluminescence along a depth gradient in the northeast Atlantic Ocean for a particular station (number 16)

```
library(AED)
data(ISIT)
Sources16 <- ISIT$Sources[ISIT$Station == 16]
Depth16 <- ISIT$SampleDepth[ISIT$Station == 16]
plot(Depth16, Sources16, type = "p")
```


A
Sources
0
10
30
50
1000
2000
3000
4000
5000
Depth


C
Sources
0
10
30
50
1000
2000
3000
4000
5000
Depth


line or quadratic fit, window or weighted
Sources
0
10
20
30
40
50
1000
2000
3000
4000
5000
Depth

Generalized Additive Model

$$Y_i = \alpha + f(X_i) + \varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim N(0, \sigma^2)$$

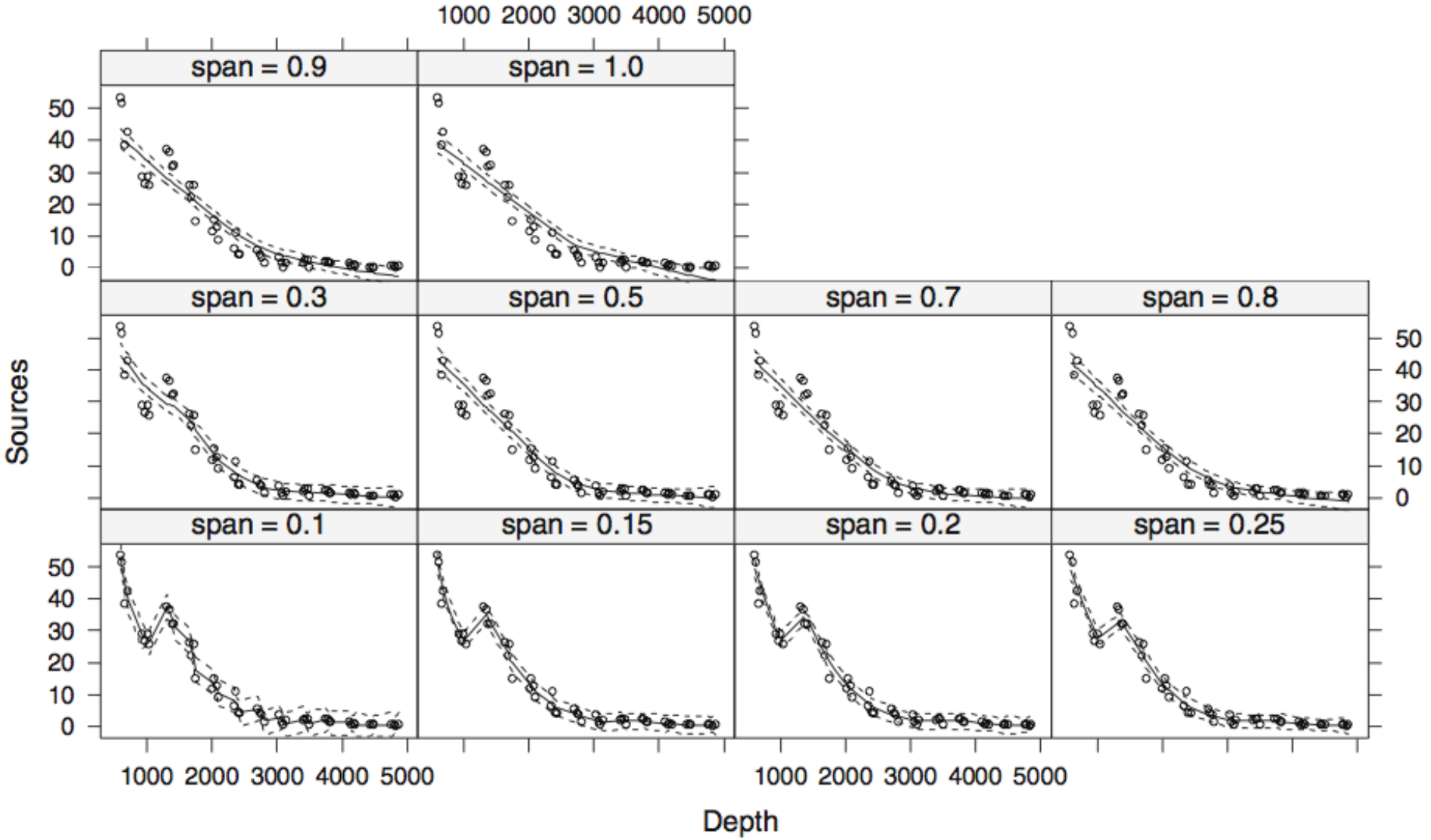
Smooth curve fit using LOESS

% of the data in included in window

```
library(gam)
M1 <- gam(Sources16~lo(Depth16, span = 0.5))
plot(M1, se = TRUE)
M11 <- gam(Sources16~lo(Depth16, span = 0.1))
plot(M11, se = TRUE)
M12 <- gam(Sources16~lo(Depth16, span = 0.95))
plot(M12, se = TRUE)
AIC(M1,M11,M12)
```

Fitting is done locally. That is, for the fit at point *x*, the fit is made using points in a neighbourhood of *x*, weighted by their distance from *x*. The size of the neighbourhood is controlled by *α* (set by span or enp.target). For *α < 1*, the neighbourhood includes proportion *α* of the points, and these have tricubic weighting (proportional to *(1 - (dist/maxdist)^3)^3*. For *α > 1*, all points are used, with the 'maximum distance' assumed to be *α^1/p* times the actual maximum distance for *p* explanatory variables.

```
M1pred <- predict(M1, se = TRUE)

plot(Depth16, Sources16, type = "p")

I1 <- order(Depth16)

lines(Depth16[I1], M1pred$fit[I1], lty = 1)
lines(Depth16[I1], M1pred$fit[I1] + 2 * M1pred$se[I1], lty = 2)
lines(Depth16[I1], M1pred$fit[I1] - 2 * M1pred$se[I1], lty = 2)
```



bias-variance tradeoff again

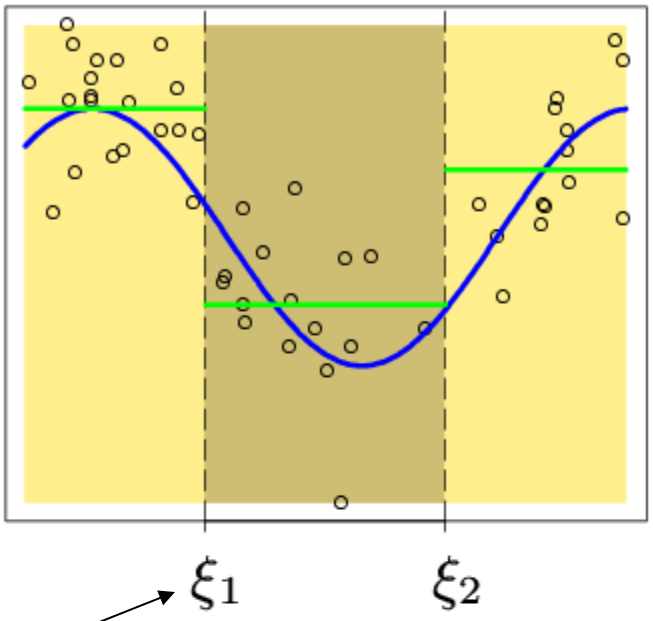
Basis Expansions for Linear Models

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X)$$

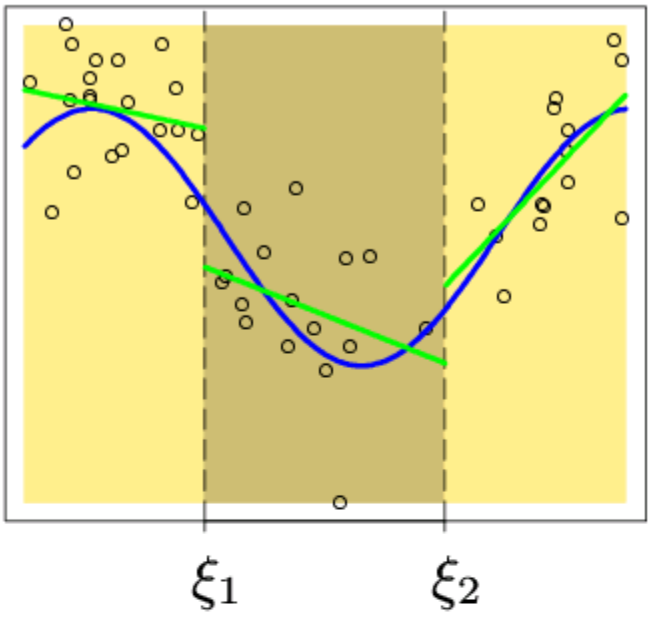
Here the h_m's might be:

- $h_m(X)=X_m$, $m=1,\ldots,p$ recovers the original model
- $h_m(X)=X_j^2$ or $h_m(X)= X_j X_k$
- $h_m(X)=\mathrm{I}(L_m \lessgtr X_k \lessgtr U_m)$,

Piecewise Constant

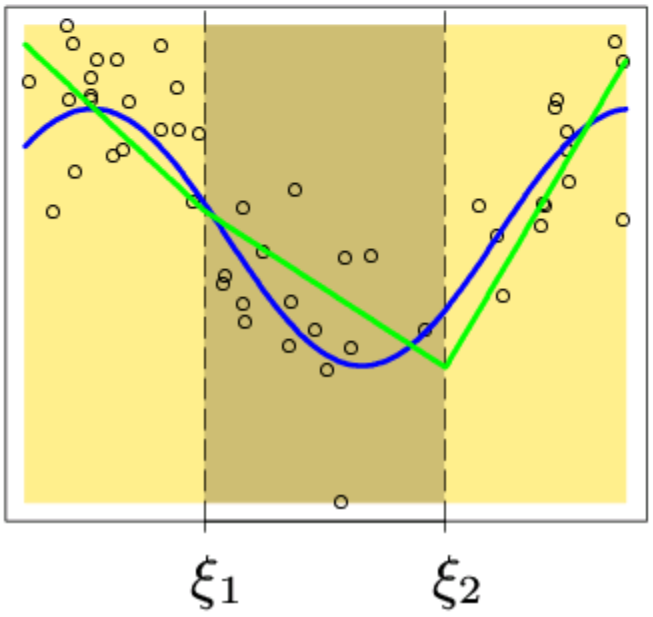


Piecewise Linear

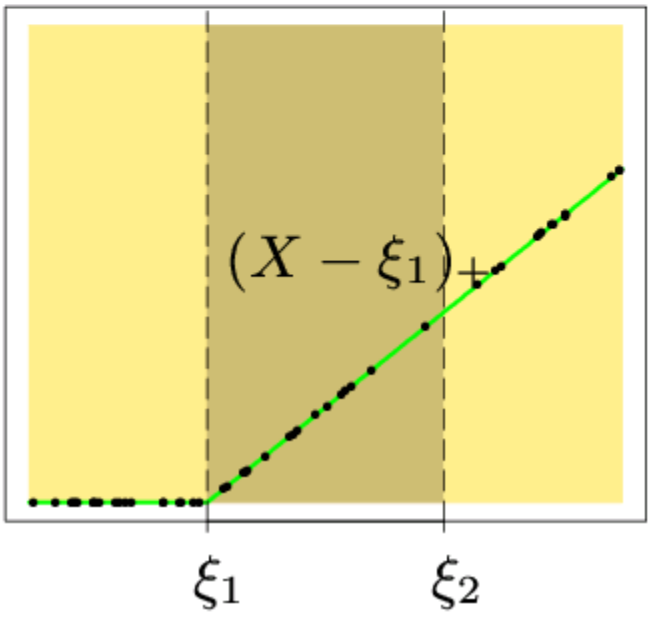


"knots"

Continuous Piecewise Linear



Piecewise-linear Basis Function

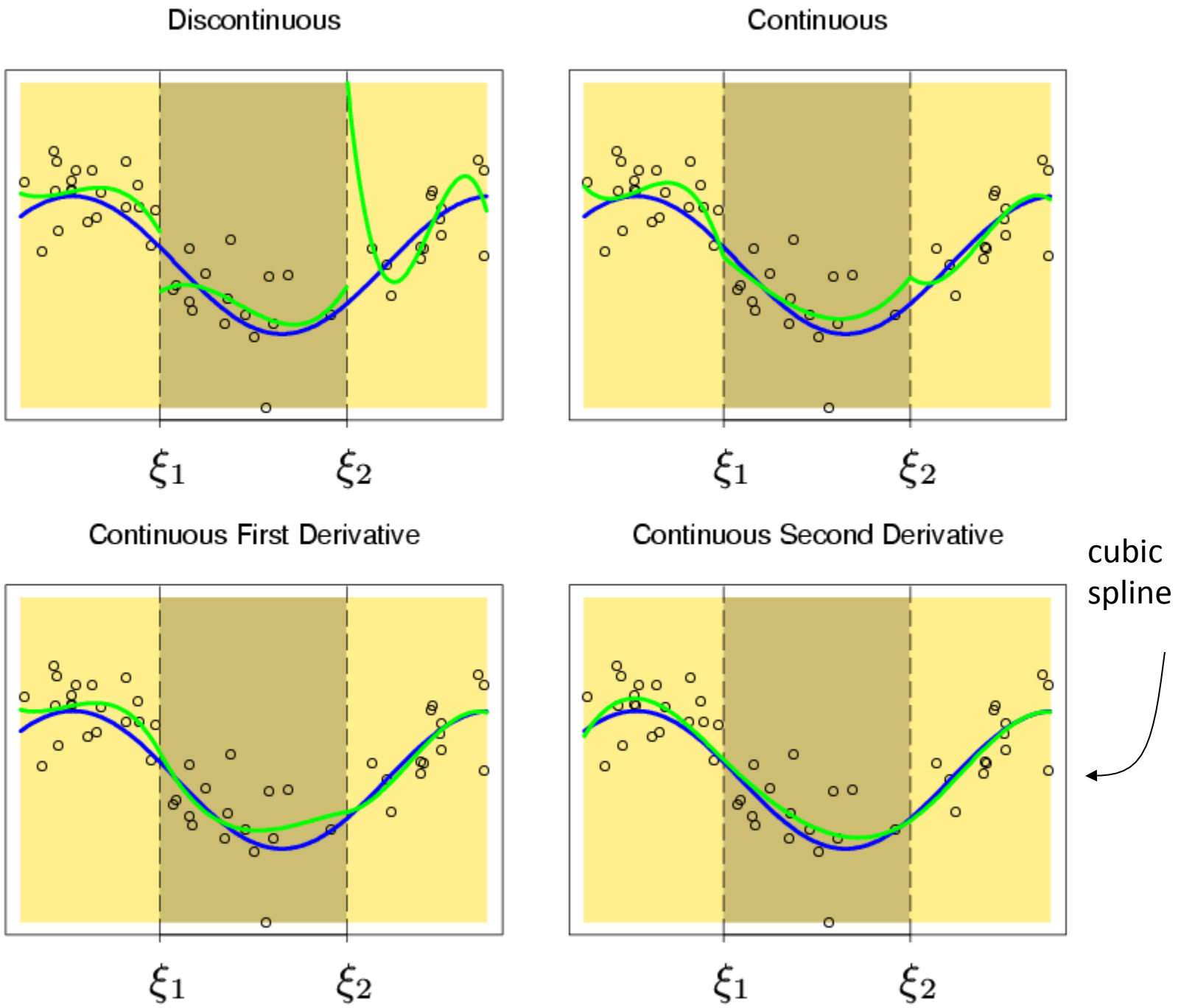


Regression Splines

Bottom left panel uses:

$$h_1(X) = 1$$
$$h_2(X) = X$$
$$h_3(X) = (X - \xi_1)_+$$
$$h_4(X) = (X - \xi_2)_+$$

Number of parameters = (3 regions) X (2 params per region)
- (2 knots X 1 constraint per knot)
= 4


Discontinuous
Continuous
Continuous First Derivative
Continuous Second Derivative
ξ_1
ξ_2
ξ_1
ξ_2
ξ_1
ξ_2
ξ_1
ξ_2
cubic
spline

Cubic Spline

continuous first and second derivatives

$$h_1(X) = 1$$
$$h_2(X) = X$$
$$h_3(X) = X^2$$
$$h_4(X) = X^3$$
$$h_5(X) = (X - \xi_1)_+^3$$
$$h_6(X) = (X - \xi_2)_+^3$$

Number of parameters = (3 regions) X (4 params per region)
- (2 knots X 3 constraints per knot)
= 6

Knot discontinuity essentially invisible to the human eye

Natural Cubic Spline

Adds a further constraint that the fitted function is linear beyond the boundary knots

A natural cubic spline model with *K* knots is represented by *K* basis functions:

$$H_1(X) = 1$$

$$H_2(X) = X$$

$$H_{k+2}(X) = d_k(X) - d_{K-1}(X), \quad \text{where}$$

$$d_k(X) = \frac{(X-\xi_k)_+^3 - (X-\xi_K)_+^3}{\xi_K - \xi_k}$$

Each of these basis functions has zero 2nd and 3rd derivative outside the boundary knots

Natural Cubic Spline Models

Can use these ideas in, for example, regression models.

For example, if you use 4 knots and hence 4 basis functions per predictor variable, then simply fit logistic regression model with four times the number of predictor variables…

Smoothing Splines

Consider this problem: among all functions $f(x)$ with two continuous derivatives, find the one that minimizes the penalized residual sum of squares:

$$RSS(f,\lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \{f''(t)\}^2 dt$$

smoothing parameter

λ=0 : f can be any function that interpolates the data

λ=infinity : least squares line

Smoothing Splines

Theorem: The unique minimizer of this penalized RSS is a natural cubic spline with knots at the unique values of x_i, i=1,…,N

Seems like there will be N features and presumably overfitting of the data. But,… the smoothing term shrinks the model towards the linear fit

$$f(x) = \sum_{i=1}^{N} H_j(x)\theta_j$$

$$RSS(\theta, \lambda) = (y - H\theta)^T (y - H\theta) + \lambda \theta^T \Omega_H \theta \quad \text{where}$$

$$\{\mathrm{H}\}_{ij} = H_j(x_i) \quad \text{and} \quad \{\Omega_H\}_{jk} = \int H_j''(t) H_k''(t) dt$$

$$\hat{\theta} = (H^T H + \lambda \Omega_H)^{-1} H^T y = S_\lambda y$$

This is a generalized ridge regression

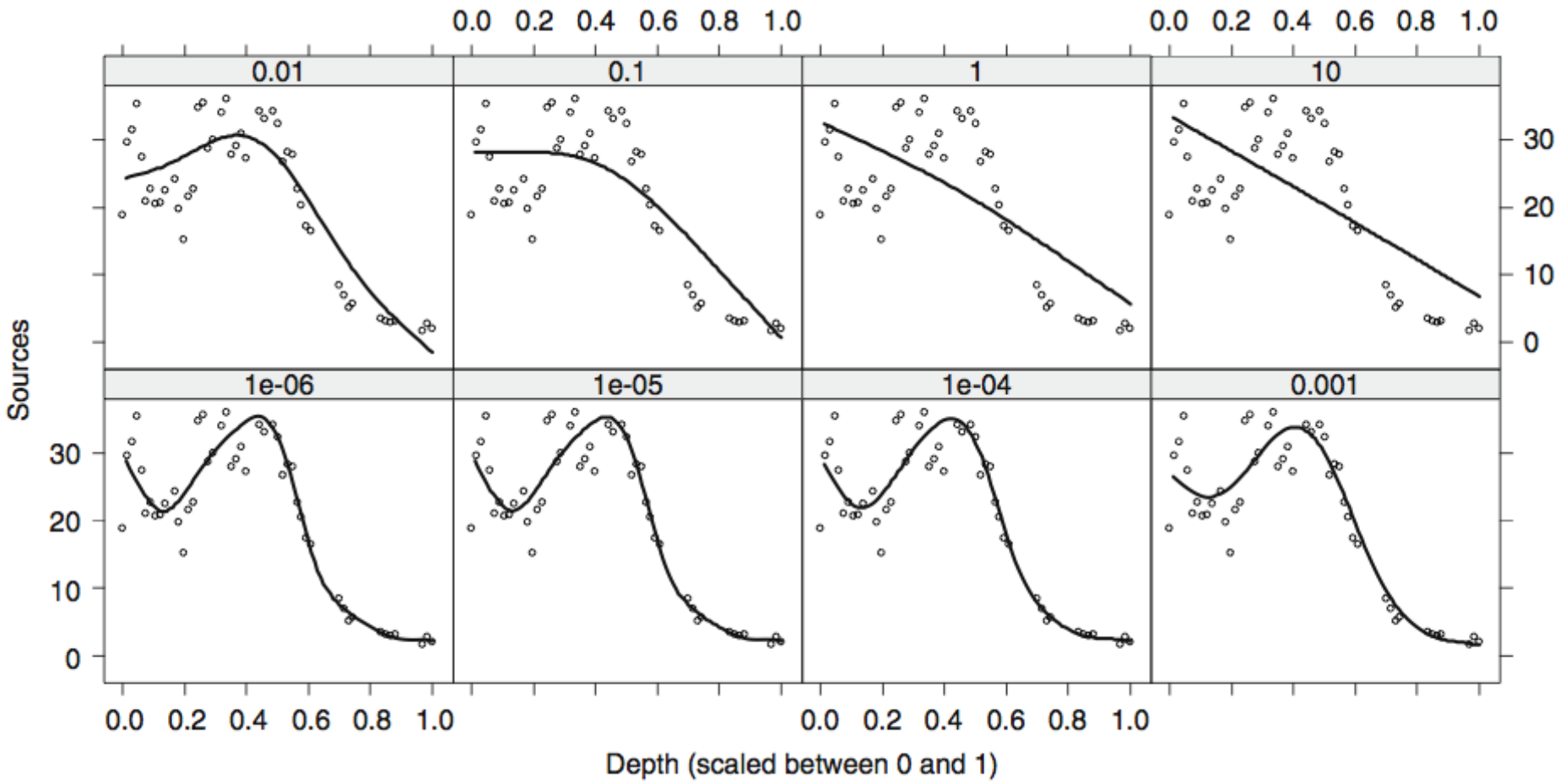
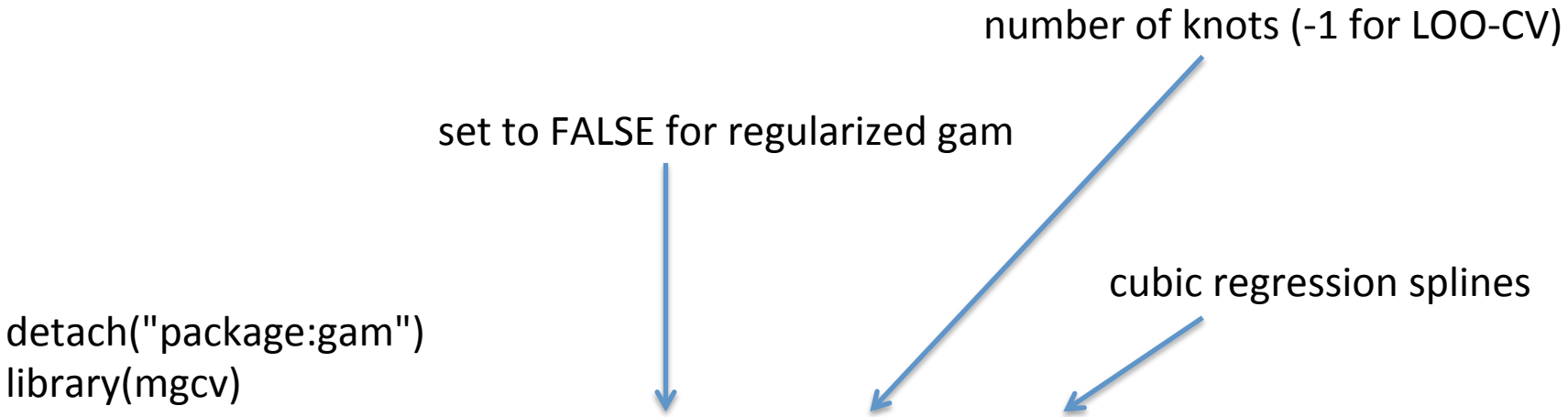


Fig. 3.10 Smoothing curve for different values of λ. The lower left panel shows the estimated smoother obtained by minimising the expression in Equation (3.9) for $\lambda = 10e{-}6$, the upper left panel for $\lambda = 10e{-}2$, and the upper right panel for $\lambda = 10$. R code to calculate the smoothing curves can be found in Wood (2006), and modified code to present the graphs in an `xyplot` from the lattice package is on the book website

cross-validation using mgcv and cubic splines



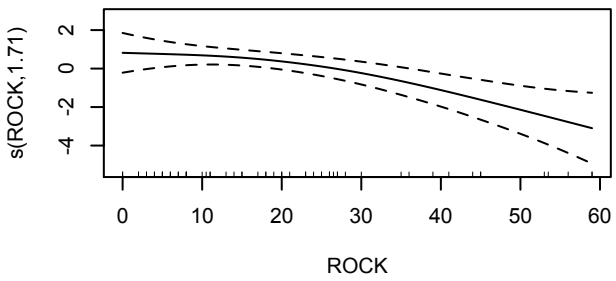

```
detach("package:gam")
library(mgcv)
M3 <- gam(Sources16~s(Depth16, fx = FALSE, k=-1, bs = "cr"))
plot(M3, se = TRUE)
M3pred <- predict(M3, se = TRUE, type = "response")

plot(Depth16, Sources16, type = "p")
I1 <- order(Depth16)
lines(Depth16[I1], M3pred$fit[I1], lty=1)
lines(Depth16[I1], M3pred$fit[I1]+2*M3pred$se[I1],lty=2)
lines(Depth16[I1], M3pred$fit[I1]-2*M3pred$se[I1],lty=2)
```

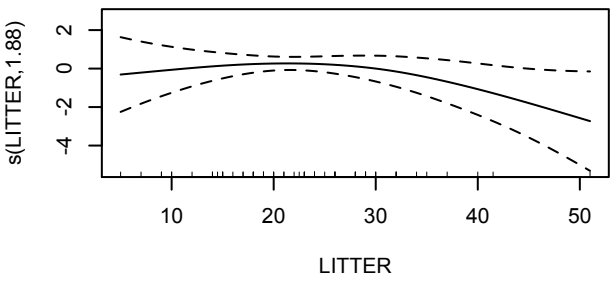
extends easily to more than one predictor…

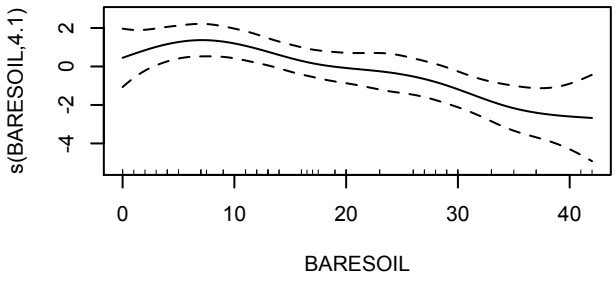
$$Y_i = \alpha + f_1(X_i) + f_2(Z_i) + \varepsilon_i \qquad \text{where} \quad \varepsilon_i \sim N(0, \sigma^2)$$

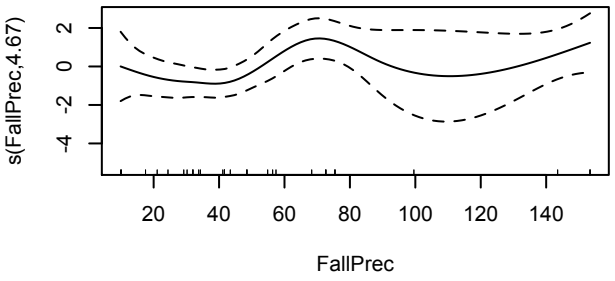
```
library(AED)
data(Vegetation)
library(mgcv)
M7 <- gam(Richness~s(ROCK, bs = "cs") +
s(LITTER, bs = "cs") + s(BARESOIL, bs = "cs") + s(FallPrec, bs = "cs") + s(SprTmax,
bs = "cs"), data = Vegetation)
par(mfrow=c(3,2))
plot(M7)
anova(M7)
```

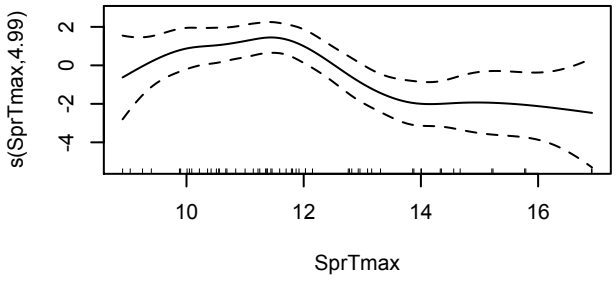
p-values approximately correct (always check residuals)
Use bootstrap if precision is required

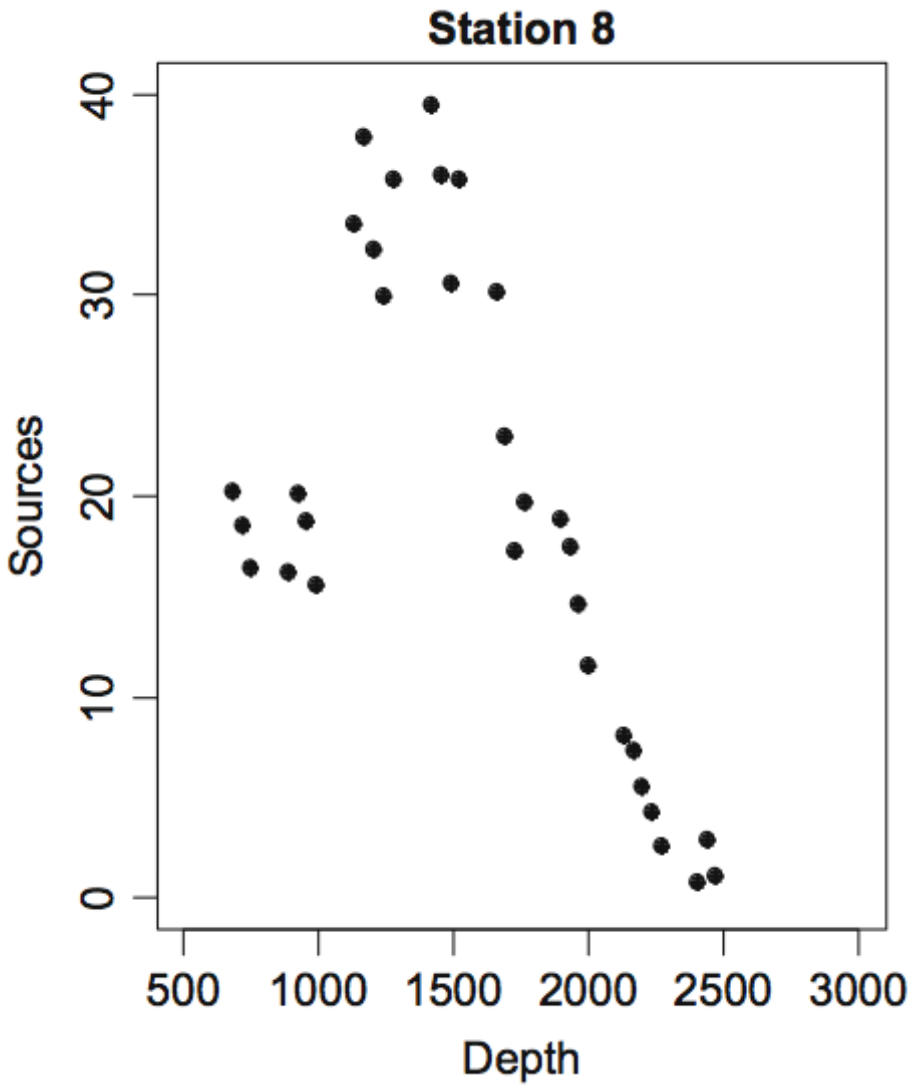

s(ROCK,1.71)
-4
-2
0
2
0
10
20
30
40
50
60
ROCK

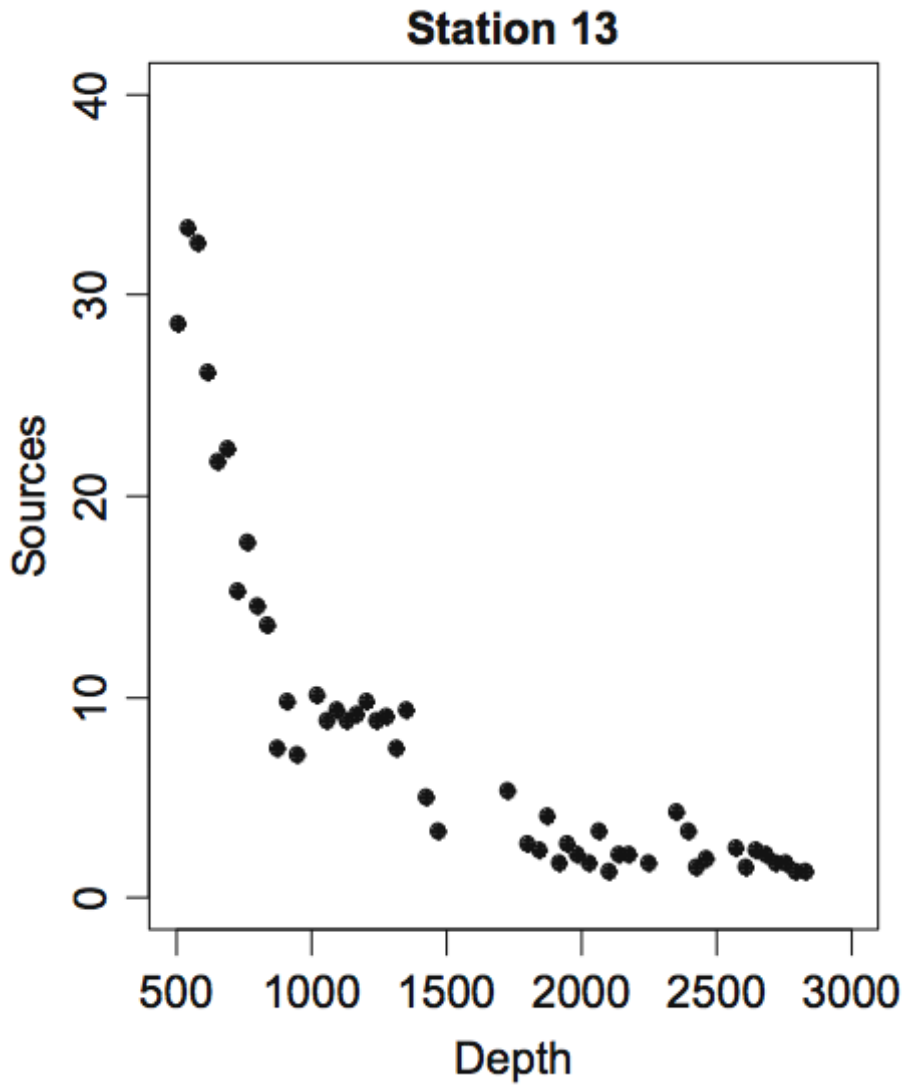

s(LITTER,1.88)
-4
-2
0
2
10
20
30
40
50
LITTER


s(BARESOIL,4.1)
-4
-2
0
2
0
10
20
30
40
BARESOIL


s(FallPrec,4.67)
-4
-2
0
2
20
40
60
80
100
120
140
FallPrec


s(SprTmax,4.99)
-4
-2
0
2
10
12
14
16
SprTmax



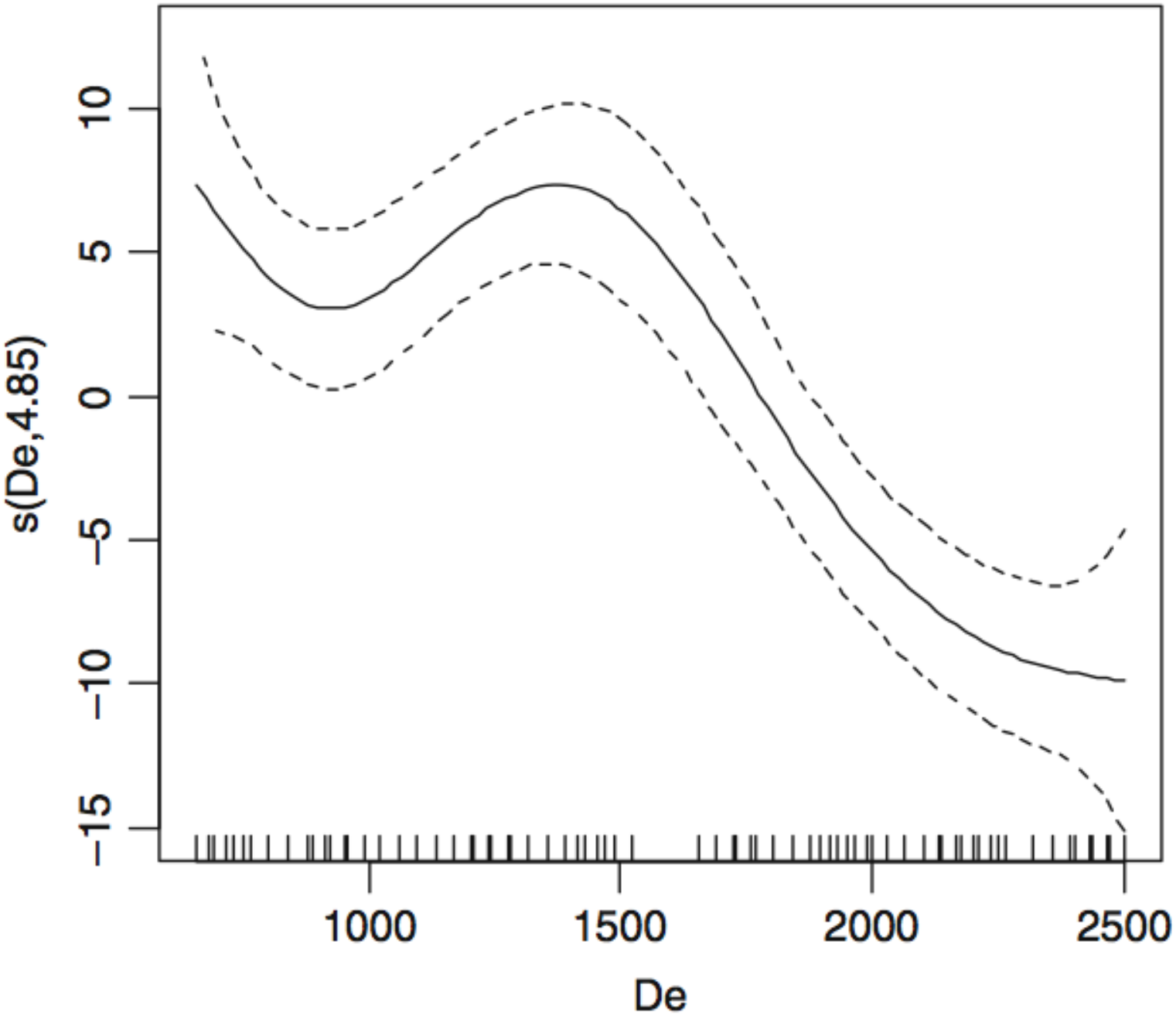


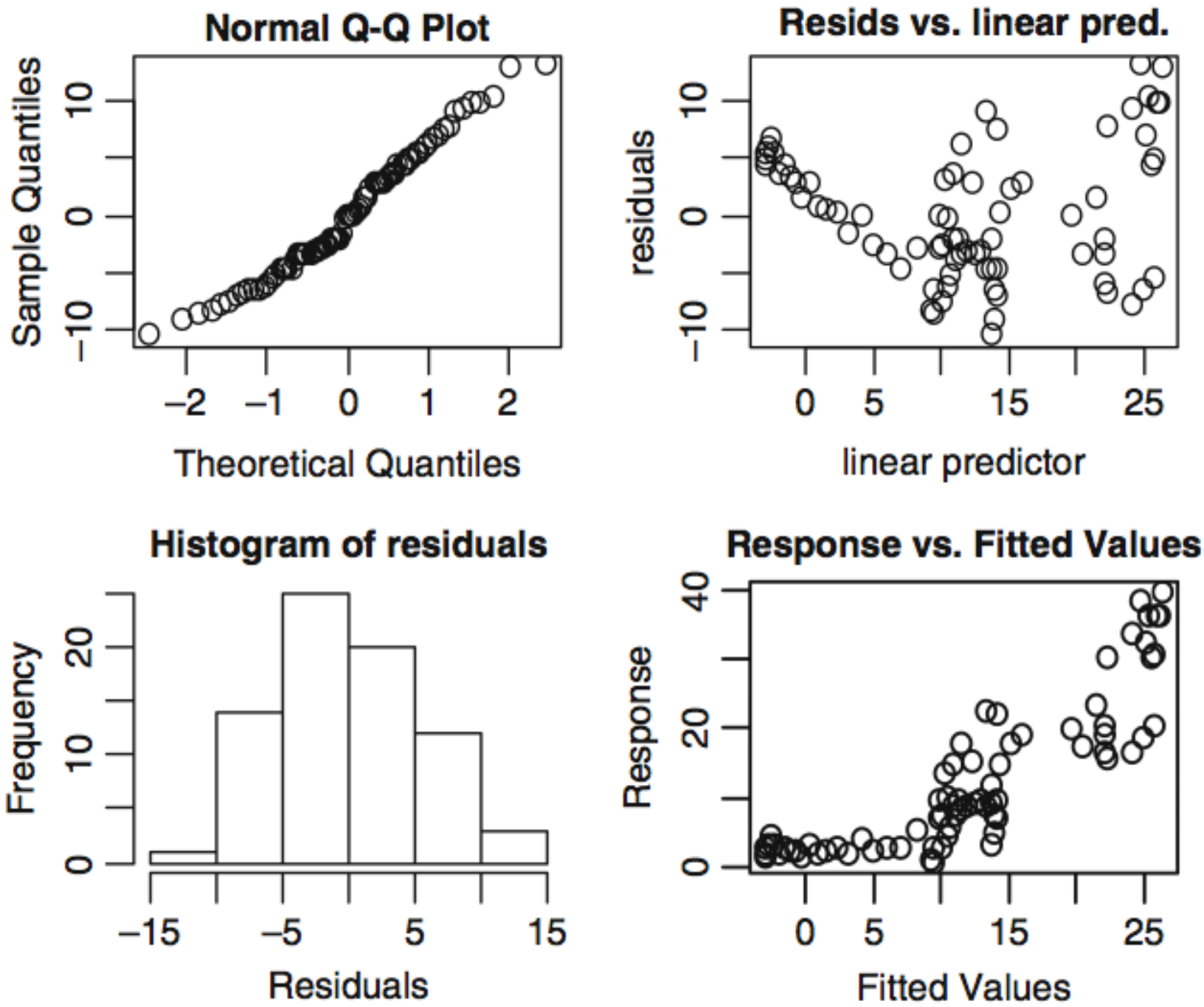
one smoother or two?

one smoother

$$Sources_i = \alpha + f(Depth_i) + factor(Station_i) + \varepsilon_i \qquad \varepsilon_i \sim N(0, \sigma^2)$$

```
library(AED)
data(ISIT)
S8 <- ISIT$Sources[ISIT$Station == 8]
D8 <- ISIT$SampleDepth[ISIT$Station == 8]
S13 <- ISIT$Sources[ISIT$Station == 13]
D13 <- ISIT$SampleDepth[ISIT$Station == 13]
ID <- rep(c(8, 13), c(length(S8), length(S13)))
So <- c(S8, S13)
De <- c(D8, D13)
library(mgcv)
M4 <- gam(So~s(De) + factor(ID))
summary(M4)
plot(M4)
gam.check(M4)
```


s(De,4.85)
10
5
0
−5
−10
−15
1000
1500
2000
2500
De


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
2
−10
0
10
Resids vs. linear pred.
residuals
linear predictor
0
5
15
25
−10
0
10
Histogram of residuals
Frequency
Residuals
−15
−5
5
15
0
10
20
Response vs. Fitted Values
Response
Fitted Values
0
5
15
25
0
20
40

two smoothers = interaction

$$Y_i = \alpha + f_1(X_{1i}) + f_2(X_{2i}) \times X_{3i} + \varepsilon_i$$

```
M5<-gam(So~s(De)+ s(De, by = as.numeric(ID == 13)) + factor(ID))
summary(M5)
gam.check(M5)
AIC(M4,M5)
```

rock dataset: predict permeability from the other measurements

```
> names(rock)
[1] "area"  "peri"  "shape" "perm"
> pairs(rock)
> rock.lm <- lm(perm~.,data=rock)
> par(mfrow=c(2,2))
> plot(rock.lm)
> rock.lm <- lm(log(perm)~.,data=rock)
> plot(rock.lm)
> rock$perm <- log(rock$perm)
> pairs(rock)
> rock.gam <- gam(log(perm)~s(area)+s(peri)+s(shape),data=rock)
> plot(rock.gam)
> gam.check(rock.gam)
```

South African heart disease: predict CHD (binary) from sbp, ldl, obesity,tobacco, family history, age

```
SAH <- read.table("http://stat.columbia.edu/~madigan/W2025/data/
SAHmissing.txt",header=TRUE,sep="\t")
```

```
SA.gam <- gam(chd~s(sbp)+s(tobacco)+s(ldl)+famhist+s(obesity)+s(age),
family=binomial(),data=SAH)
```