Flexible Regression and Smoothing

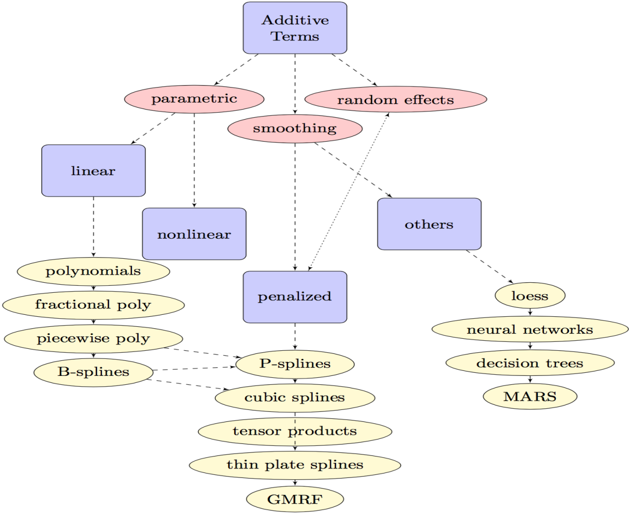
Linear and Smoothing Additive Terms

Mikis Stasinopoulos   Bob Rigby

Graz University of Technology, Austria, November 2016

gamlss

Additive Terms



Additive Terms

model

$$\begin{aligned}\eta &= b_0 + b_1 x_1 + b_2 x_2 + b_3 \texttt{if}(f=2) + b_4 \texttt{if}(f=3) + b_5 (x_1 \times x_2) + \\ &\quad b_6 x_1 \texttt{if}(f=2) + b_7 x_1 \texttt{if}(f=3) + s_1(x_3) + s_2(x_4) + \\ &\quad s_3 b_8(x_1) x_4 + s_4(x_1)\texttt{if}(f=2) + s_5(x_1)\texttt{if}(f=3) + s_6(x_3, x_5)\end{aligned}$$

R code

```
m1 <- gamlss(y~x1+x2+f+x1:x2+f:x1+pb(x3)+pb(x4)+
                      pvc(x1, by=f)+ga(~s(x3,x5))
```

gamlss

The CD4 data

Data summary: the CD4 data

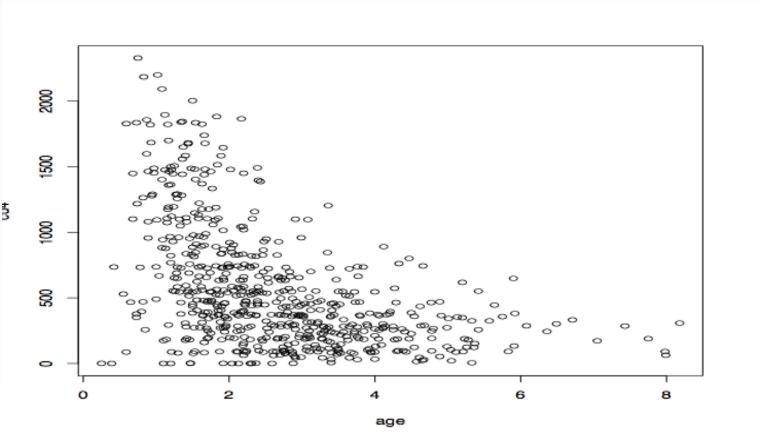
R data file: CD4 in package **MASS** of dimensions 609 × 2

variables

- cd4 : CD4 counts from uninfected children born to HIV-1 mothers.
- age : The age of child in in years

purpose: to demonstrate the use of linear parametric terms

The CD4 data: plot



Polynomials

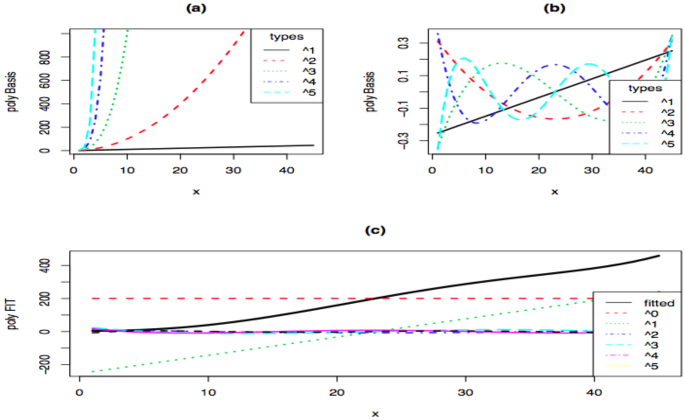
Model

$$h(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots + \beta_p x^p$$

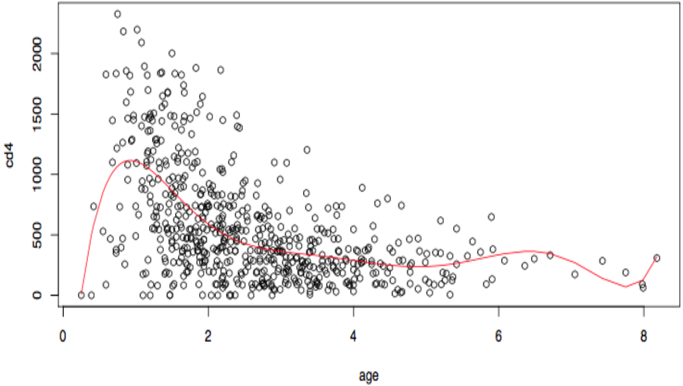
R code

```
m0 <- gamlss(y~x+I(x^2)+I(x^3))
m1 <- gamlss(y~poly(x,3))
```

Polynomials: basis



CD4 data example, polynomials degree 7



gamlss

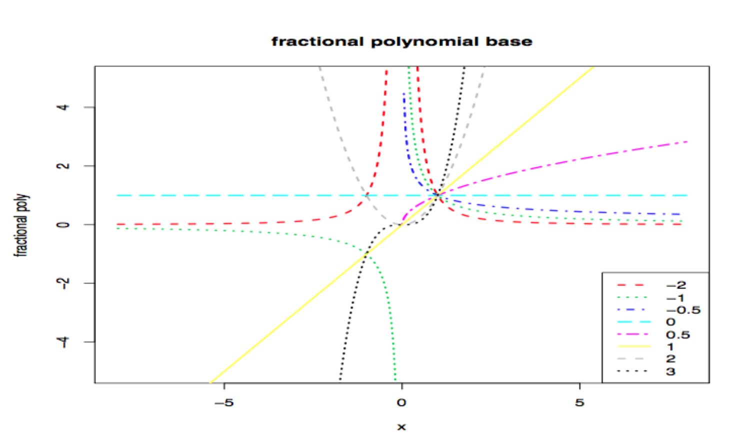
Fractional Polynomials

$$\beta_0 + \beta_1 x^{p_1} + \beta_2 x^{p_2} + \beta_3 x^{p_3}$$

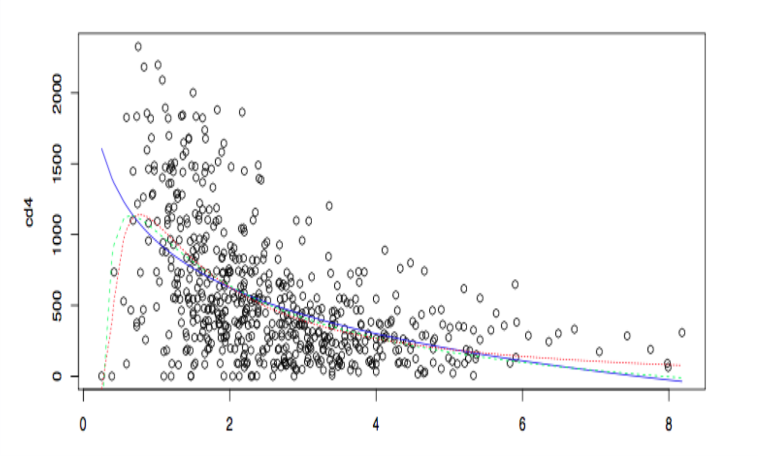
where p_j, for $j = 1, 2, 3$ takes values $(-2, -1, -0.5, 0, 0.5, 1, 2, 3)$ with the value 0 interpreted as function $\log(x)$ (rather than x^0).
R code

```
m0 <- gamlss(y~fp(x, 3))
```

Fractional Polynomials



CD4 data example, fractional polynomials with 1, 2 and 3 degrees



gamlss

Piecewise Polynomials, Linear + Linear

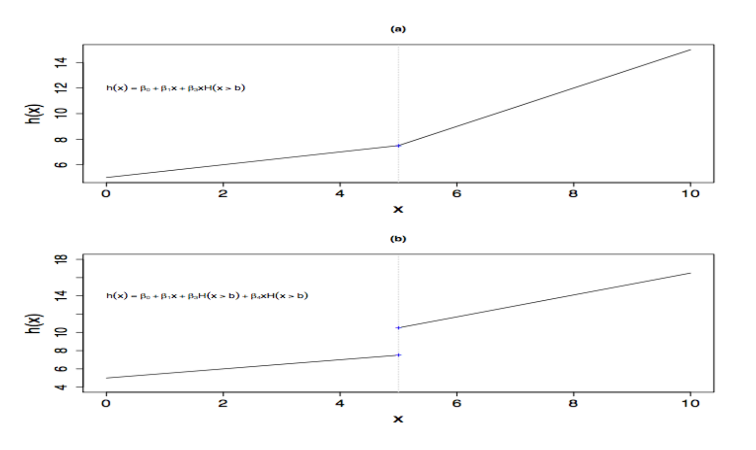
$$h(x) = \beta_{00} + \beta_{01}x + \beta_{11}(x - b)H(x > b)$$

R code

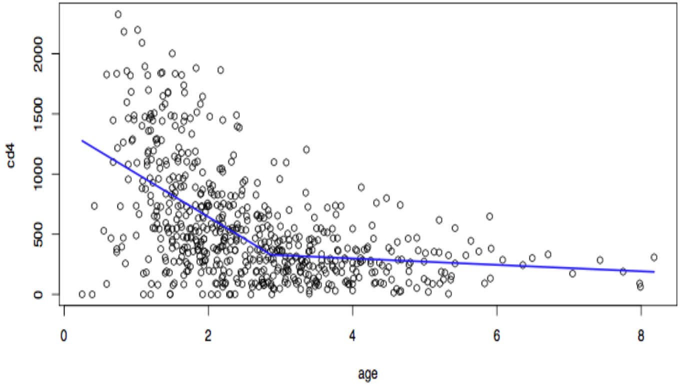
```
f1<-gamlss(cd4~fk(age, degree=1, start=2), data=CD4)
```

gamlss

Piecewise Polynomials, Linear + Linear



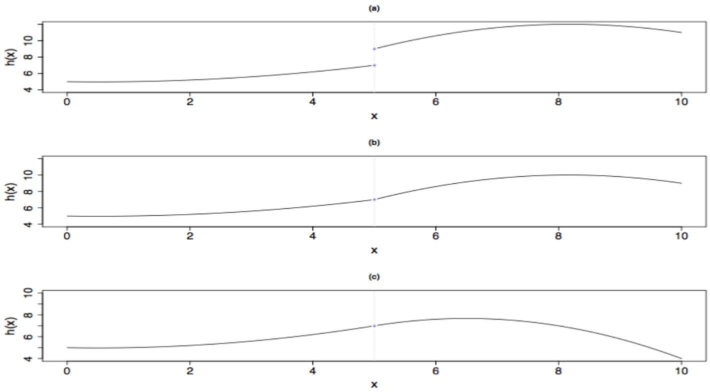
CD4 data example, piecewise polynomials



Piecewise Polynomials, Quadratic + Quadratic

$$h(x) = \beta_{00} + \beta_{01}x + \beta_{02}x^2 + \left[\beta_{10} + \beta_{11}(x-b) + \beta_{12}(x-b)^2\right] H(x > b). \quad (1$$

Piecewise Polynomials, Quadratic + Quadratic



gamlss

Piecewise Polynomials: splines

$$h(x) = \sum_{j=0}^{D} \beta_{0j} x^j + \sum_{k=1}^{K} \sum_{j=0}^{D} \beta_{kj} (x - b_k)^j H(x > b_k)$$

Splines

$$h(x) = \sum_{j=0}^{D} \beta_{0j} x^j + \sum_{k=1}^{K} \beta_k (x - b_k)^D H(x > b_k)$$

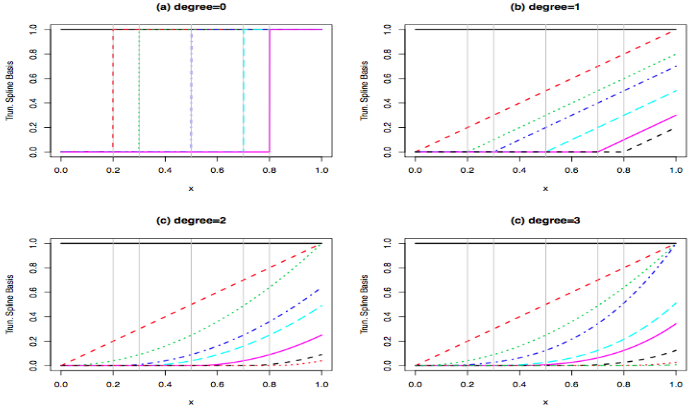
cubic splines

$$h(x) = \beta_{00} + \beta_{01} x + \beta_{02} x^2 + \beta_{03} x^3 + \sum_{k=1}^{K} \beta_k (x - b_k)^3 H(x > b_k)$$

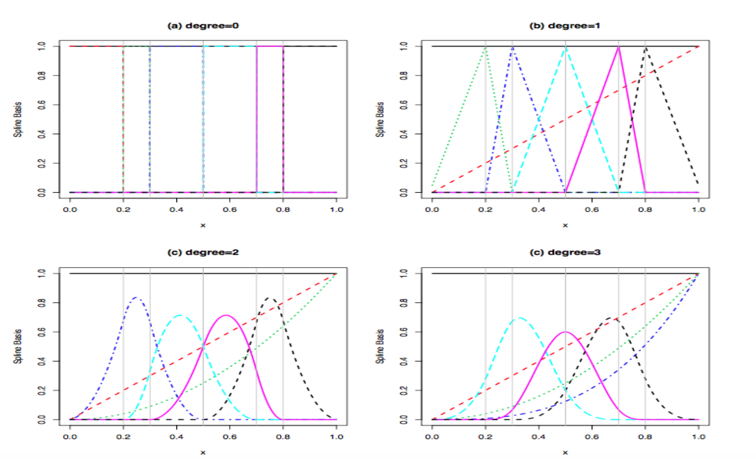
natural cubic splines

$$h(x) = \beta_{00} + \beta_{01} x + \sum_{k=1}^{K} \beta_k (x - b_k)^3 H(x > b_k)$$

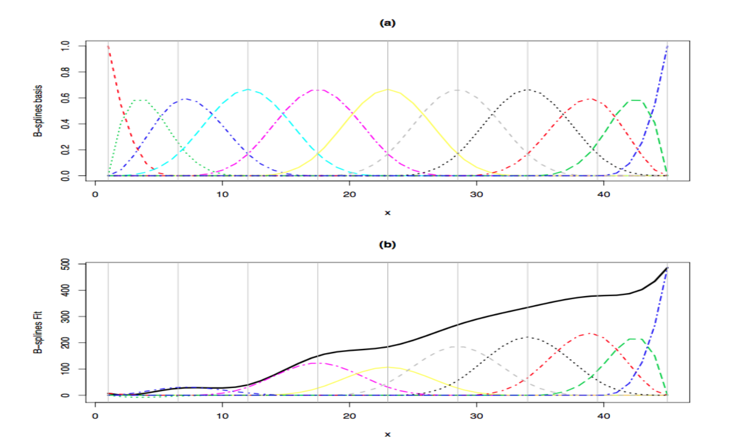
Splines basis: truncated



Splines basis: B-splines



Splines basis, linear function of a B-splines basis



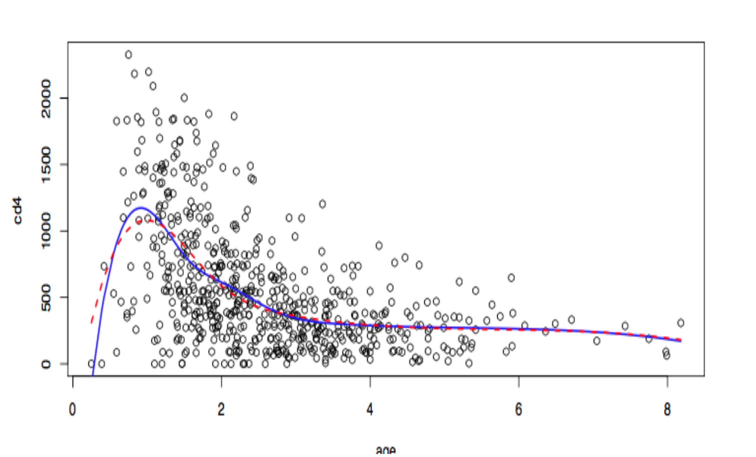
gamlss

Piecewise polynomial regression

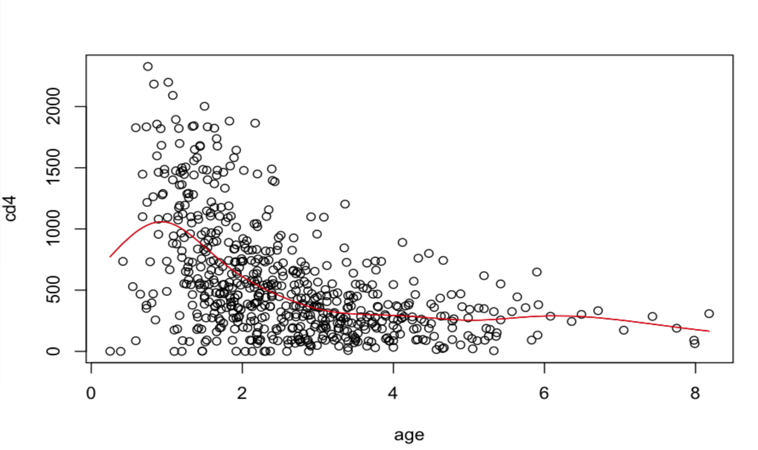
R code

```
m2b <- gamlss(cd4 ~ bs(age), data = CD4)
m3b <- gamlss(cd4 ~ bs(age, df = 3), data = CD4)
m4b <- gamlss(cd4 ~ bs(age, df = 4), data = CD4)
m5b <- gamlss(cd4 ~ bs(age, df = 5), data = CD4)
m6b <- gamlss(cd4 ~ bs(age, df = 6), data = CD4)
m7b <- gamlss(cd4 ~ bs(age, df = 7), data = CD4)
m8b <- gamlss(cd4 ~ bs(age, df = 8), data = CD4)
```

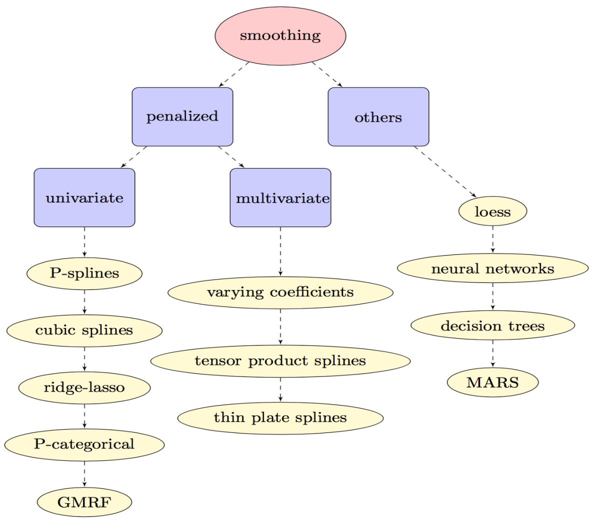
CD4 data example, piecewise polynomials 5, 7 knots



CD4 count data: P-spline fits

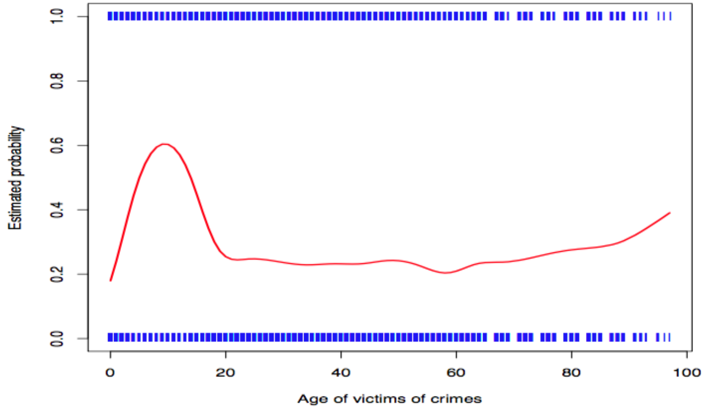


Additive Smoothing Terms: Smoothers

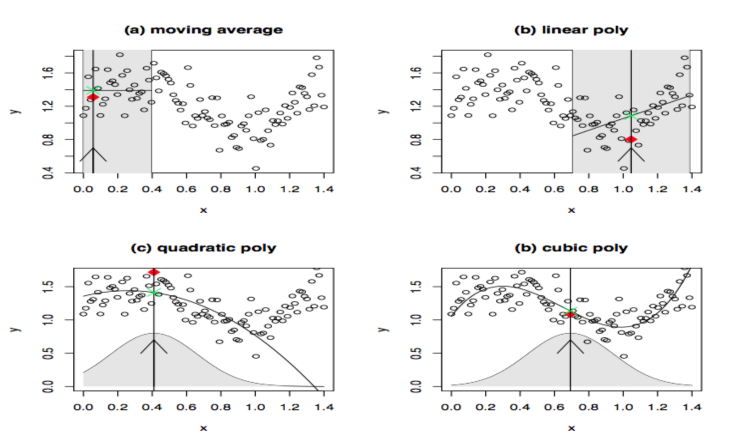


gamlss

Reported crimes



Local Polynomial Smoothers



gamlss

Penalised Smoothers

$$Q = (\mathbf{y} - \mathbf{Z}\gamma)^\top \mathbf{W}(\mathbf{y} - \mathbf{Z}\gamma) + \lambda\gamma^\top \mathbf{G}\gamma.$$

with solution

$$\hat{\gamma} = \left(\mathbf{Z}^\top \mathbf{W}\mathbf{Z} + \lambda \mathbf{G}\right)^{-1} \mathbf{Z}^\top \mathbf{W}\mathbf{y}.$$

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{Z}\left(\mathbf{Z}^\top \mathbf{W}\mathbf{Z} + \lambda \mathbf{G}\right)^{-1} \mathbf{Z}^\top \mathbf{W}\mathbf{y} \\ &= \mathbf{S}\mathbf{y} \end{aligned}$$

gamlss

Demos

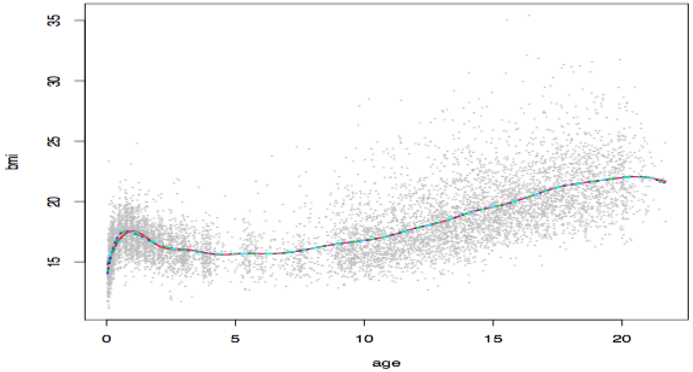
1. `demo.BSplines()`
2. `demo.RandomWalk()`
3. `demo.interpolateSmo()`
4. `demo.histSmo()`
5. `demo.PSplines()`

P-splines: pb()

R code

```
p1 <- gamlss(bmi~pb(age, method="ML"), data=dbbmi)
p2 <- gamlss(bmi~pb(age, method="GCV") , data=dbbmi)
p3 <- gamlss(bmi~pb(age, method="GAIC", k=2) , data=dbbmi)
p4 <- gamlss(bmi~pb(age, method="GAIC",
                      k=log(length(dbbmi$bmi))),
                      data=dbbmi)
```

P-splines: pb()



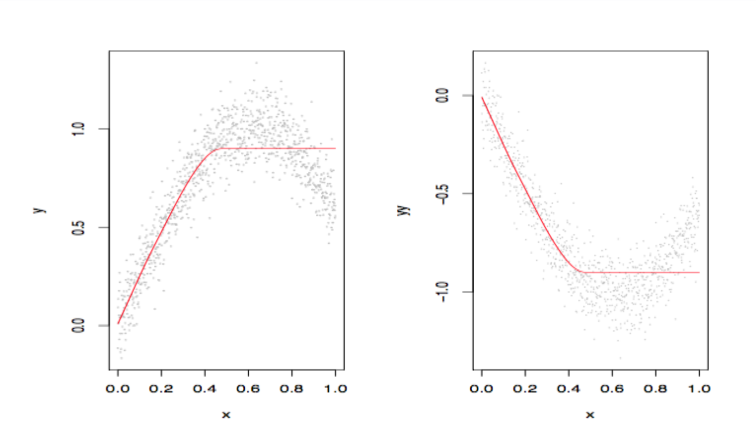
gamlss

Monotonic P-splines: pbm()

R code

```
set.seed(1334)
x = seq(0, 1, length = 1000)
p = 0.4
y = sin(2 * pi * p * x) + rnorm(1000) * 0.1
m1 <- gamlss(y~pbm(x), trace=FALSE)
yy <- -y
m2 <- gamlss(yy~pbm(x, mono="down"), trace=FALSE)
```

Monotonic P-splines

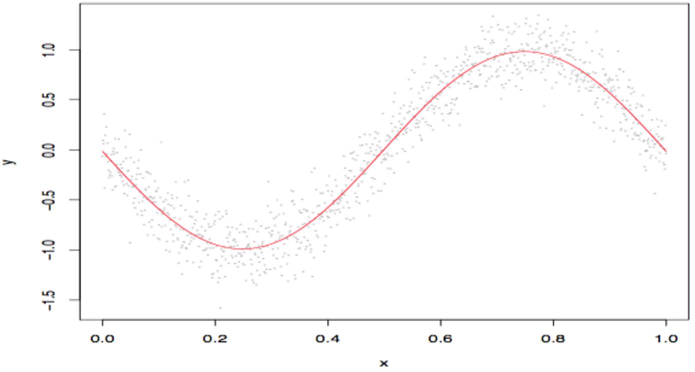


Cycle P-splines: pbc(), cy()

R code

```
 set.seed(555)
x = seq(0, 1, length = 1000)
y<-cos(1 * x * 2 * pi + pi / 2)+rnorm(length(x)) * 0.2
m1<-gamlss(y~cy(x), trace=FALSE)
```

Cycle P-splines



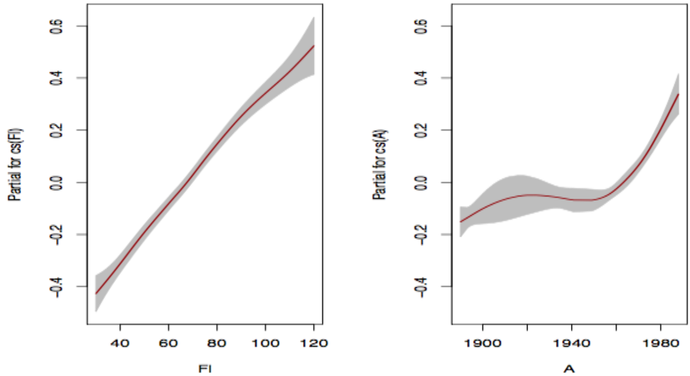
gamlss

Cubic smoothing splines cs()

R code

```
# fitting cubic splines with fixed degrees of freedom
rcs1<-gamlss(R~cs(Fl)+cs(A), data=rent, family=GA)
term.plot(rcs1, pages=1)
```

Cubic smoothing splines



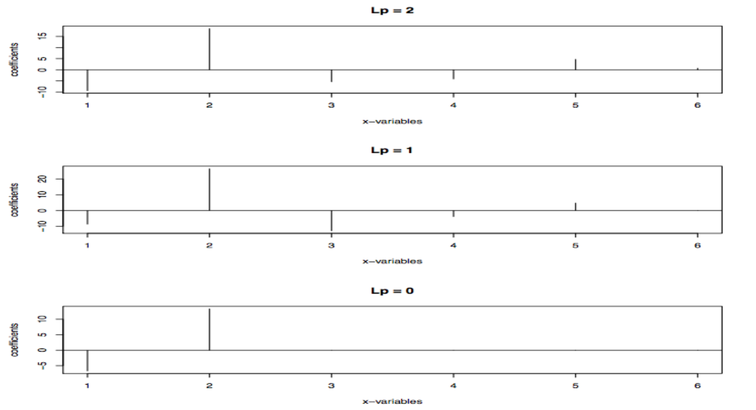
gamlss

Ridge and Lasso Regression : ri()

R code

```
# standardise the data
X<-with(usair, cbind(x1,x2,x3,x4,x5,x6))
sX<-scale(X)
m0<- gamlss(y~sX, data=usair) # least squares
m1<- gamlss(y~ri(sX), data=usair) # ridge
m2<- gamlss(y~ri(sX, Lp=1), data=usair) # lasso
m3<- gamlss(y~ri(sX, Lp=0), data=usair) # best subset
AIC(m0,m1,m2,m3)
##          df      AIC
## m2 5.336309 341.2492
## m0 8.000000 344.7232
## m1 5.884452 345.6097
## m3 2.838310 350.1807
```

Ridge and Lasso Regression

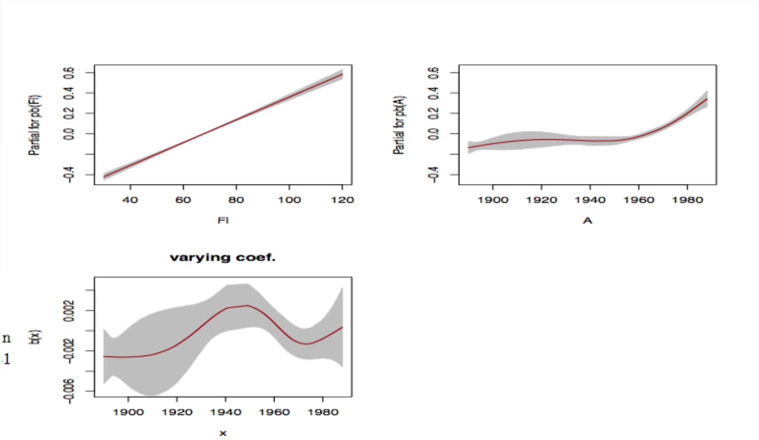


Varying Coefficients Models : pvc()

R code

```
m2<-gamlss(R~pb(Fl)+pb(A)+pvc(A, by=Fl), data=rent,
                        family=GA)
term.plot(m2, pages=1)
```

Varying Coefficients Models

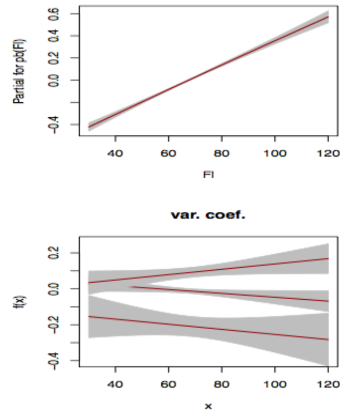


Varying Coefficients Models

R code

```
g1<-gamlss(R~pb(Fl)+pb(A)+pvc(Fl,by=loc),  data=rent,
                                         family=GA)
term.plot(g1, pages=1)
```

Varying Coefficients Models

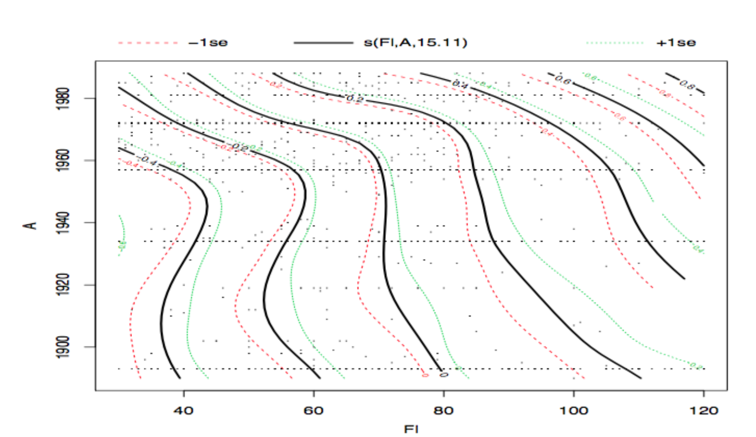


Interface to gam()

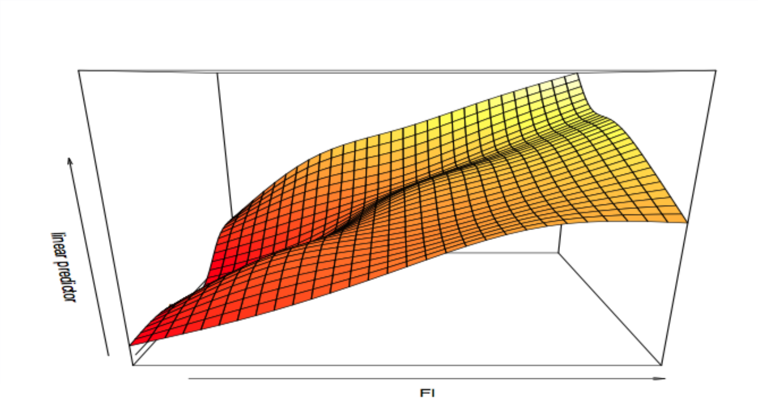
R code

```
ga4 <-gam(R~s(Fl,A), method="REML", data=rent,
                            family=Gamma(log))
gn4 <- gamlss(R~ga(~s(Fl,A), method="REML"), data=rent,
                            family=GA)
term.plot(gn4)
vis.gam(getSmo(gn5))
```

Interface to gam()



Interface to gam()

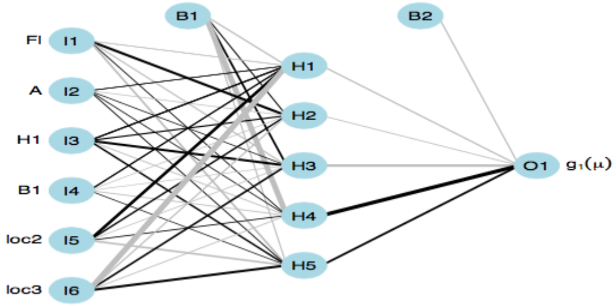


Neural Network: Interface to nnet()

R code

```
mr3 <- gamlss(R~nn(~Fl+A+H+B+loc, size=5, decay=0.01),
            data=rent, family=GA)
summary(getSmo(mr3))
## a 6-5-1 network with 41 weights
## options were - linear output units  decay=0.01
##  b->h1 i1->h1 i2->h1 i3->h1 i4->h1 i5->h1 i6->h1
......
##  -0.13  -0.86   0.05   0.84   0.00  -1.43   2.07
##  b->o h1->o h2->o h3->o h4->o h5->o
## -0.28 -0.68 -0.24 -2.39  3.64  1.33
plot(getSmo(mr3), y.lab=expression(g[1](mu)))
```

Interface to nnet()

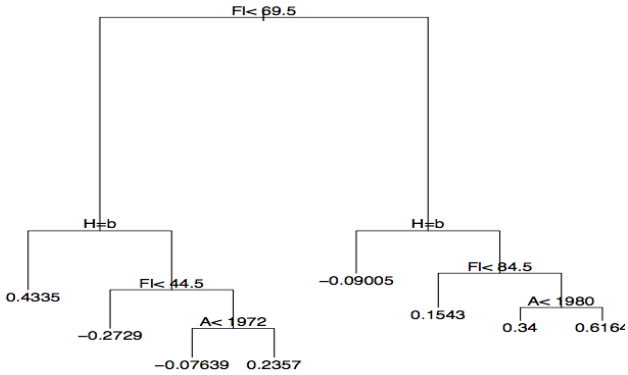


Decision Trees: interface to rpart()

R code

```
r2 <- gamlss(R ~ tr(~Fl+A+H+B+loc),
                sigma.fo=~tr(~Fl+A+H+B+loc), data=rent,
                family=GA, gd.tol=100, c.crit=0.1)
term.plot(r2, parameter="mu", pages=1)
```

Decision Tress interface to rpart()



END

for more information see

www.gamlss.org