Additive Terms

Flexible Regression and Smoothing

Mikis Stasinopoulos[1] Bob Rigby[1]

[1]STORM, London Metropolitan University

XXV SIMPOSIO INTERNACIONAL DE ESTADêSTICA, Armenia, Colombia, August 2015

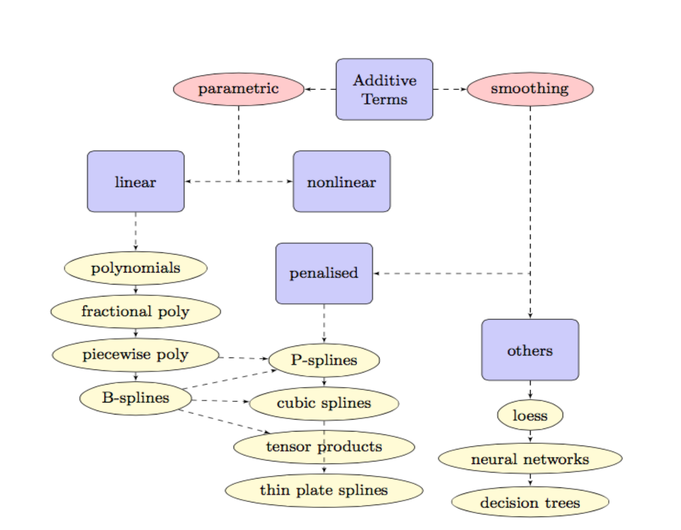
LONDON METROPOLITAN UNIVERSITY

gamlss

1. Additive Terms
   - Linear Additive
   - Smoothers
2. Centiles
   - The Dutch boys data
   - The problem
   - The methods
   - The LMS model and its extensions
   - GAMLSS functions for centile estimation
3. Conclusions

gamlss

Additive Terms



Additive Terms

model

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

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

Additive Terms: CD4 data information

Data summary: the CD4 data

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

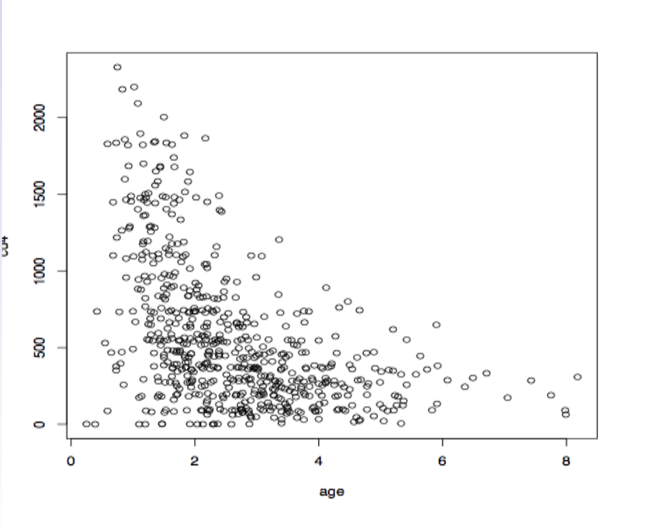
variables

- 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

gamlss

Additive Terms: CD4 data example



Additive Terms: polynomials

Model

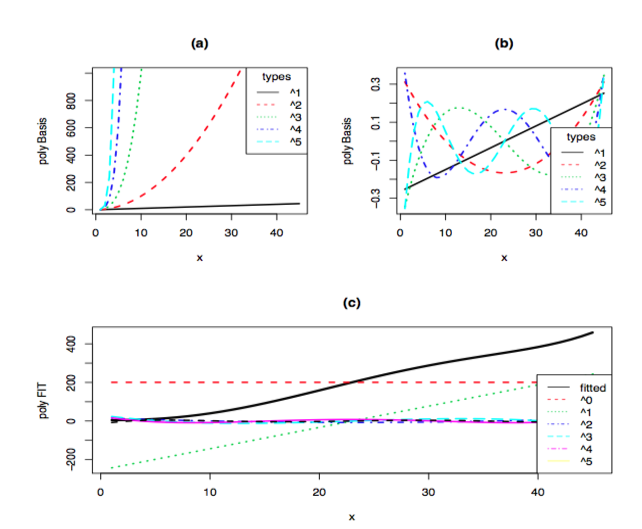
$$h(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots + \beta_p x^p$$

R code

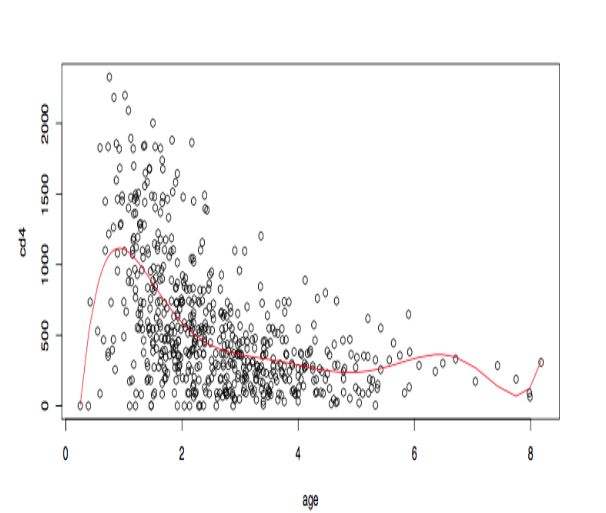
```
m0 <- gamlss(y~x+I(x^2)+I(x^3))
m1 <- gamlss(y~poly(x,3))
```

gamlss

Additive Terms: polynomials



Additive Terms: CD4 data example, polynomials degree 7



gamlss

Additive Terms: fractional polynomials

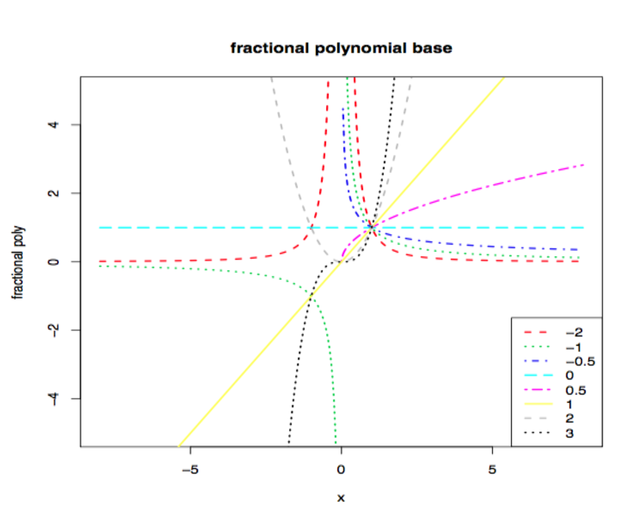
$$\beta_0 + \beta_1 x^{p_1} + \beta_2 x^{p_2} + \beta_3 x^{p_3}$$

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))
```

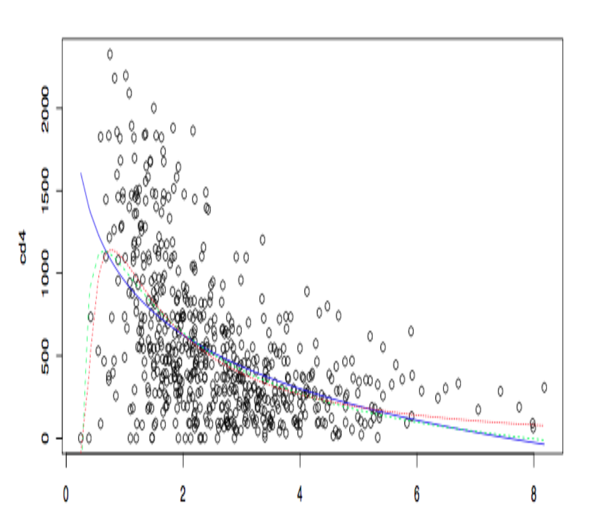
gamlss

Additive Terms: fractional polynomials



gamlss

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



gamlss

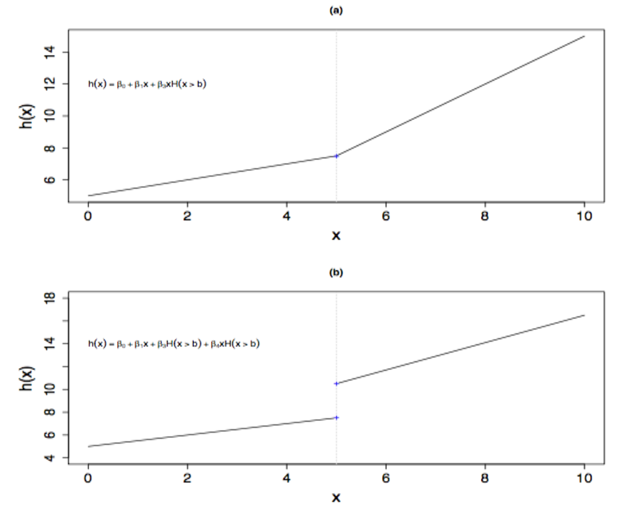
Additive Terms: Piecewise Polynomials, Linear + Linear

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

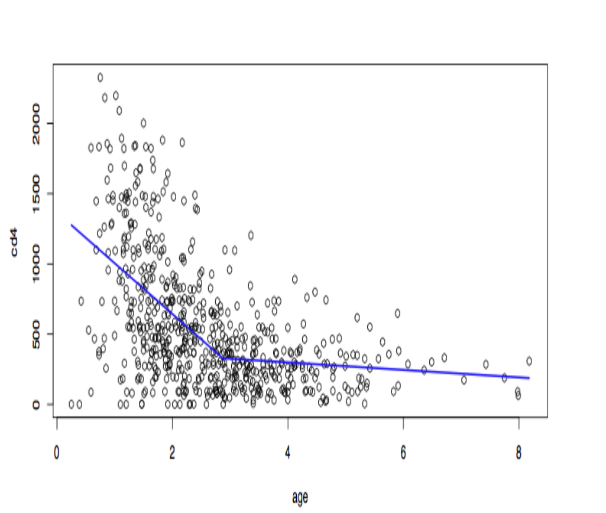
R code

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

Additive Terms: Piecewise Polynomials, Linear + Linear



Additive Terms: CD4 data example, piecewise polynomials



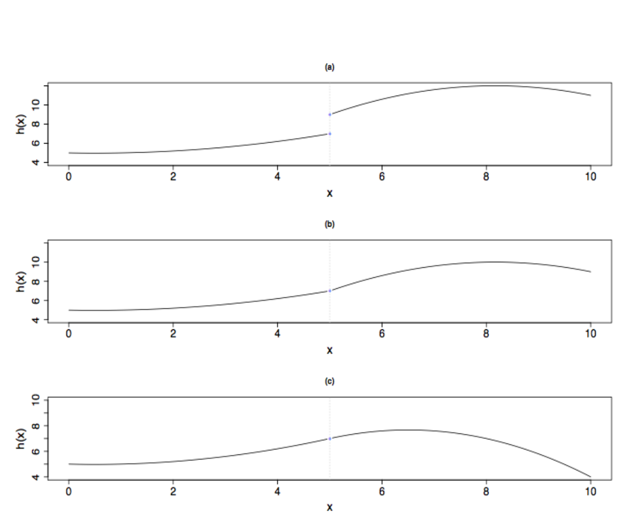
gamlss

Additive Terms: Piecewise Polynomials, Quadratic + Quadratic

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

gamlss

Additive Terms: Piecewise Polynomials, Quadratic + Quadratic



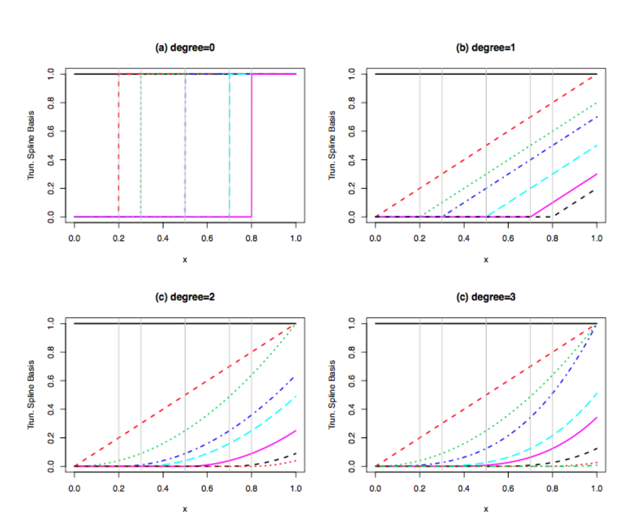
gamlss

Additive Terms: Piecewise Polynomials

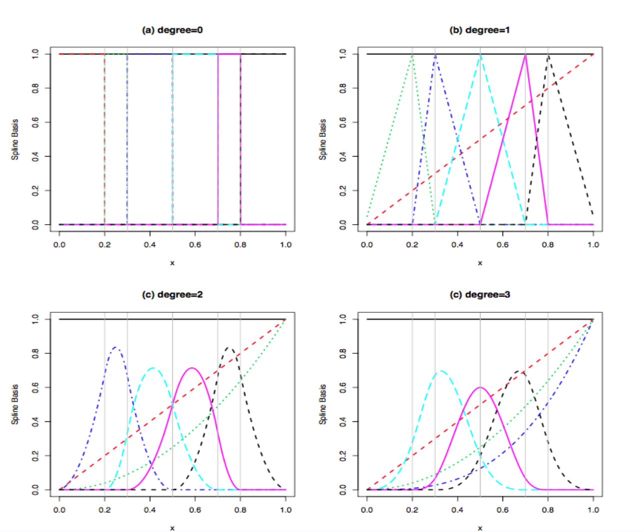
$$h(x) = \sum_{j=0}^{D} \beta_{0j} x^j + \sum_{k=1}^{K} \sum_{j=0}^{D} \beta_{kj} (x - b_k)^j H(x > b_k)$$

gamlss

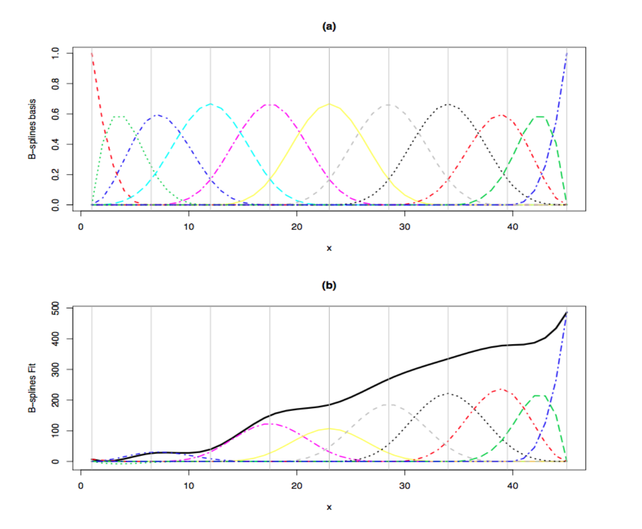
Additive Terms: Piecewise Polynomials



Additive Terms: Piecewise Polynomials, B-splines



Additive Terms: Piecewise Polynomials, B-splines



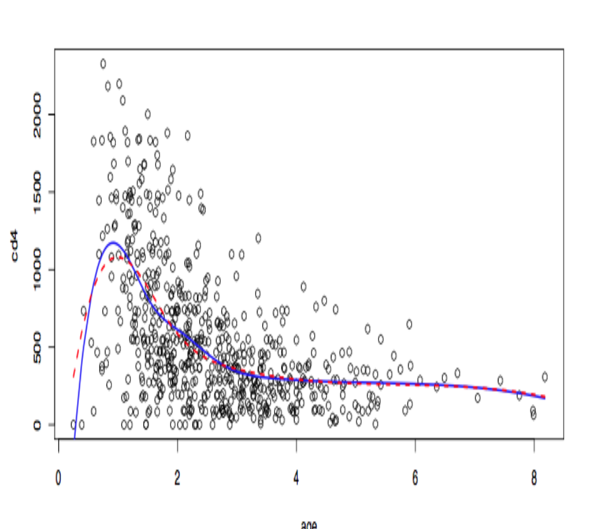
gamlss

Additive Terms: piecewise polynomial regression

R code

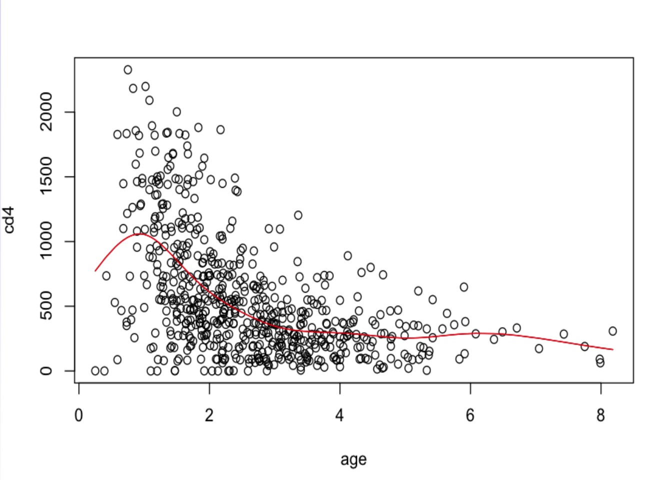
```
m2b <- gamlss(cd4 ~ bs(age), data = CD4)
m3b <- gamlss(cd4 ~ bs(age, df = 3), data = CD4)
m4b <- gamlss(cd4 ~ bs(age, df = 4), data = CD4)
m5b <- gamlss(cd4 ~ bs(age, df = 5), data = CD4)
m6b <- gamlss(cd4 ~ bs(age, df = 6), data = CD4)
m7b <- gamlss(cd4 ~ bs(age, df = 7), data = CD4)
m8b <- gamlss(cd4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

Additive Terms: CD4 data example, piecewise polynomials 5, 7 knots



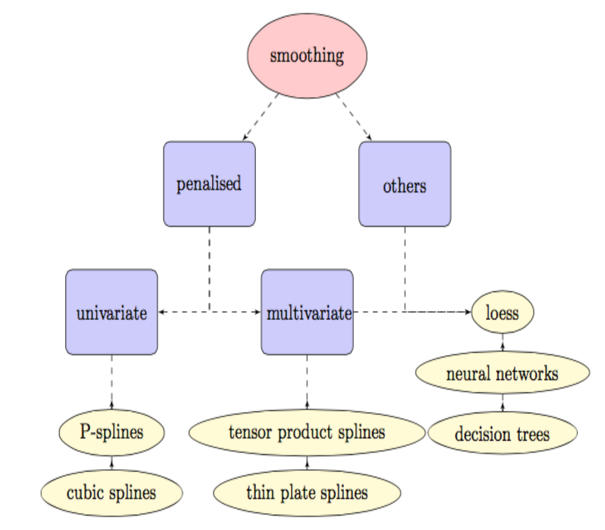
gamlss

CD4 count data: P-spline fits



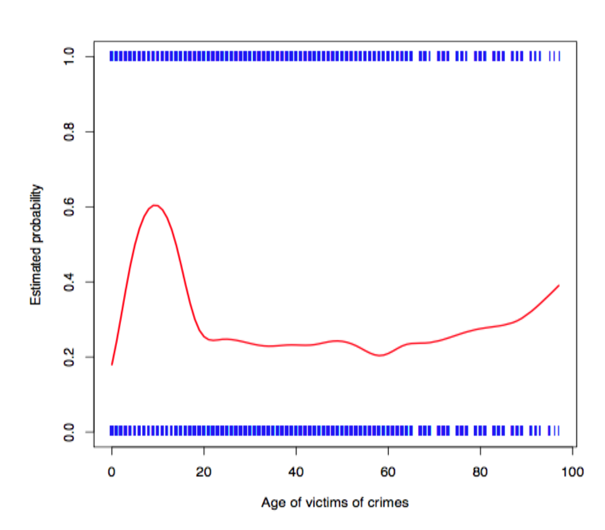
gamlss

Smoothers



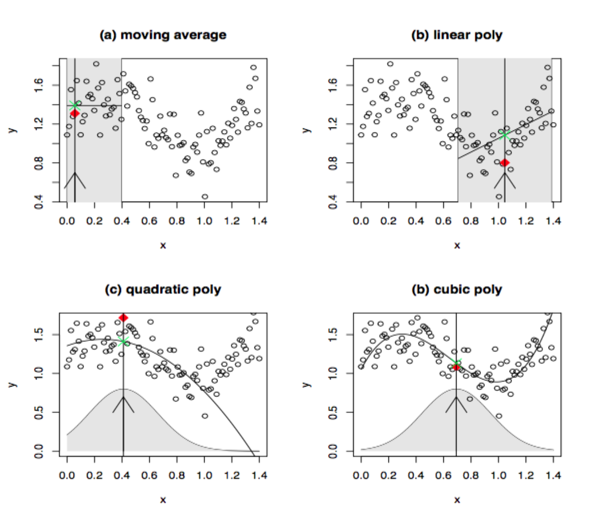
gamlss

Smoothers: reported crimes



gamlss

Smoothers: local linear



Smoothers: penalised

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

with solution

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

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

Smoothers: Demos

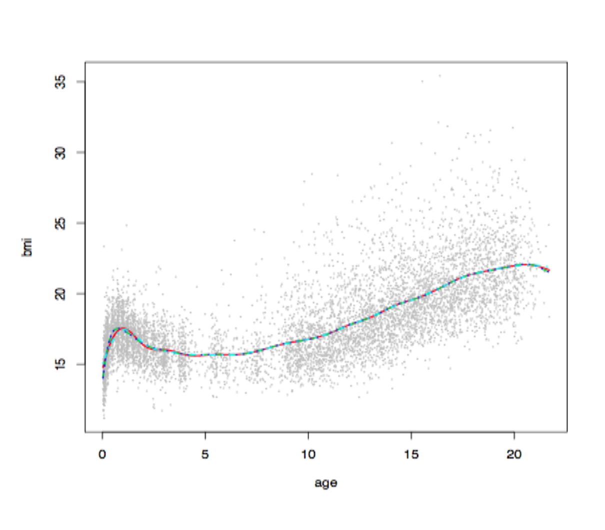
1. `demo.BSplines()`
2. `demo.RandomWalk()`
3. `demo.interpolateSmo()`
4. `demo.histSmo()`
5. `demo.PSplines()`

Additive Terms: 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)
```

Smooth functions: P-splines pb()



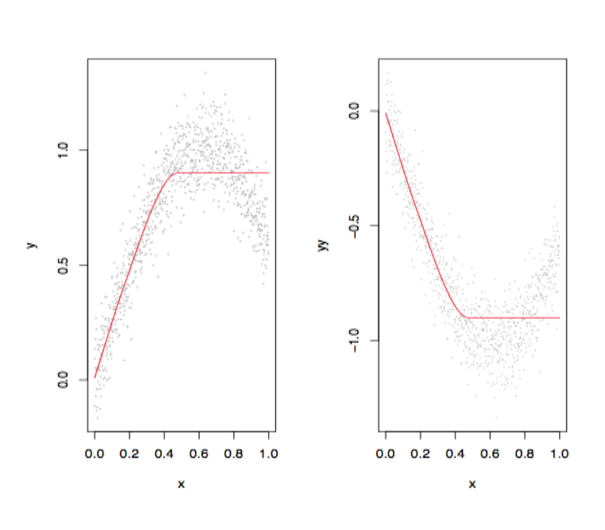
gamlss

Additive Terms: 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)
```

Smooth functions: Monotonic P-splines

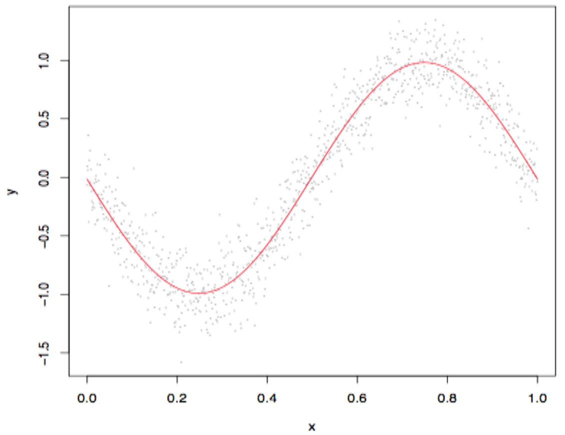


Additive Terms: cycle P-splines cy()

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)
```

Smooth functions:cycle P-splines



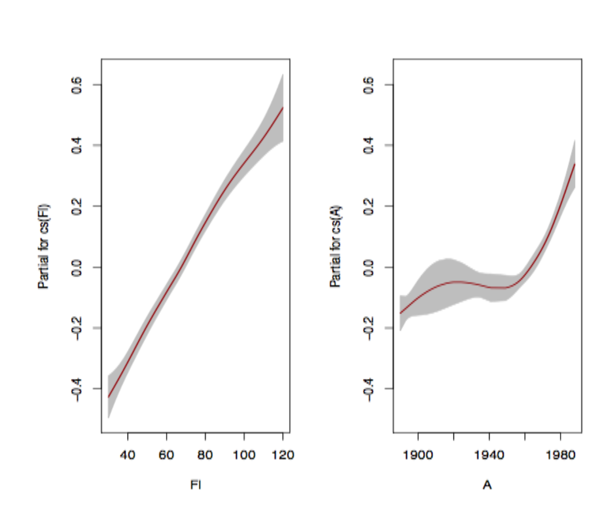
gamlss

Additive Terms: cubic 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)
```

Smooth functions: cubic splines

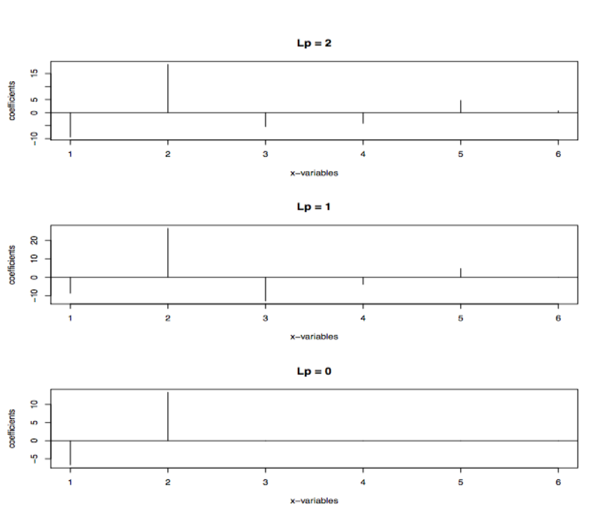


Additive Terms: ridge and lasso ri()

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
```

Smooth functions: ridge and lasso



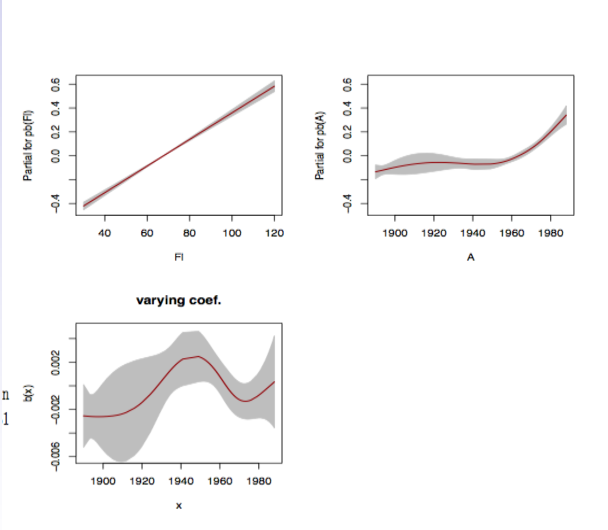
gamlss

Additive Terms: varying coefficients pvc()

R code

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

Smooth functions: varying coefficients

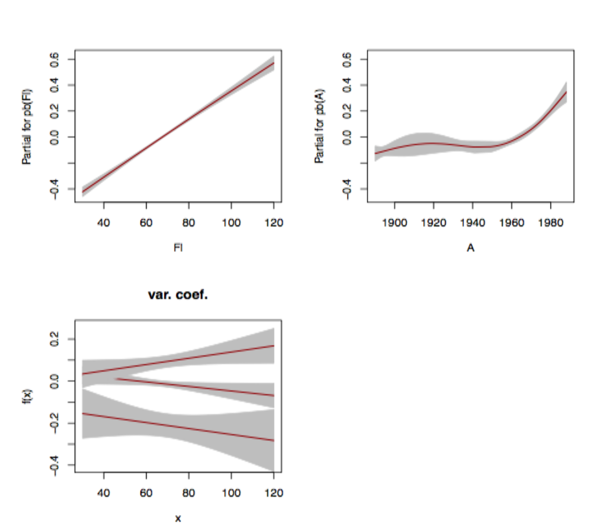


Additive Terms: varying coefficients pvc()

R code

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

Smooth functions: varying coefficients



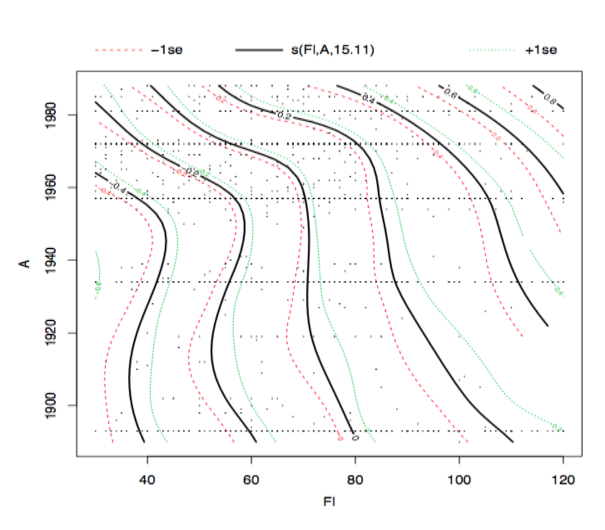
gamlss

Additive Terms: interface to gam()

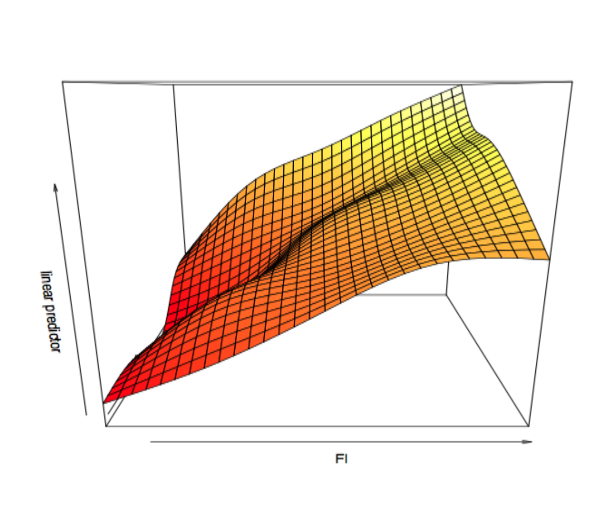
R code

```
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))
```

Smoothers: interface to gam()



Smooth functions: interface to gam()



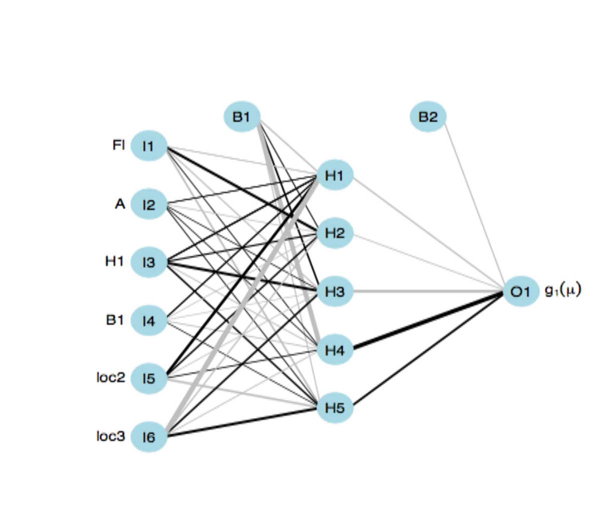
Additive Terms: neural network interface to nnet()

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)))
```

gamlss

Smooth functions: interface to nnet()



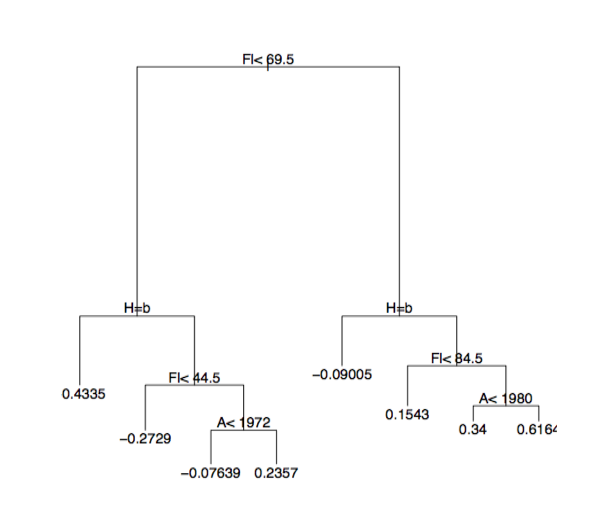
gamlss

Additive Terms: 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)
```

Smooth functions: decision tress interface to rpart()



gamlss

The Dutch boys data

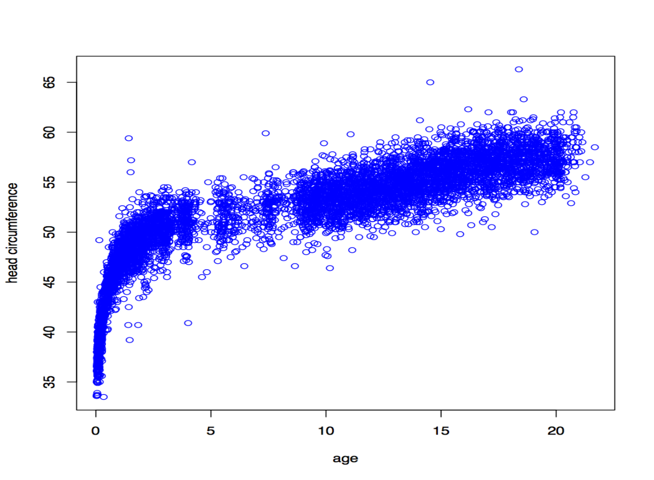
`head` : the head head circumference of 7040 boys

`age` : the age in years

Source: Buuren and Fredriks (2001)

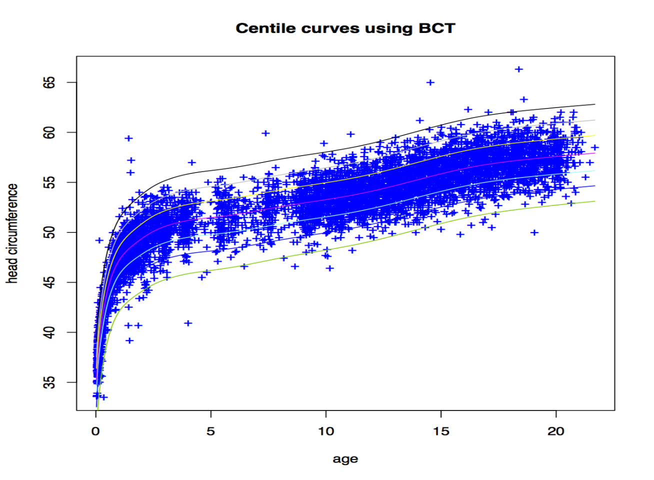
gamlss

The Dutch boys data

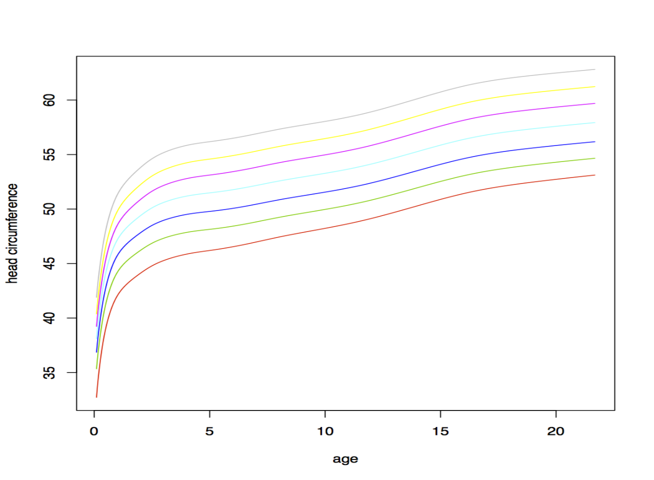


gamlss

The Dutch boys data with centiles



The Dutch boys data: centiles



The methods

i) the non parametric approach of quantile regression (Koenker, 2005; Koenker and Bassett, 1978)

ii) the parametric LMS approach of Cole (1988), Cole and Green (1992) and its extensions Rigby and Stasinopoulos (2004, 2006, 2007).

Centiles: the LMS method

$Y \sim f_Y(y|\mu, \sigma, \nu, \tau)$ where $f_Y()$ is any distribution

$Y =$head circumference and $X = AGE^{\xi}$

$$\begin{aligned} \mu &= s(x, df_{\mu}) \\ \log(\sigma) &= s(x, df_{\sigma}) \\ \nu &= s(x, df_{\nu}) \\ \log(\tau) &= s(x, df_{\tau}) \end{aligned}$$

gamlss

The LMS method and extensions

Let Y be a random variable with range $Y > 0$ defined through the transformed variable Z given by:

$$
\begin{aligned}
Z &= \frac{1}{\sigma\nu}\left[\left(\frac{Y}{\mu}\right)^{\nu} - 1\right], \quad \text{if} \quad \nu \neq 0 \\
&= \frac{1}{\sigma}\log\left(\frac{Y}{\mu}\right), \quad \text{if} \quad \nu = 0.
\end{aligned}
$$

1. if $Z \sim N(0,1)$ then $Y \sim BCCG(\mu,\sigma,\nu) =$ LMS method
2. if $Z \sim t_\tau$ then $Y \sim BCT(\mu,\sigma,\nu,\tau) =$ LMST method
3. if $Z \sim PE(0,1,\tau)$ then $Y \sim BCPE(\mu,\sigma,\nu,\tau) =$ LMSP method adopted by WHO

gamlss

Centiles: estimation of the smoothing parameters

We need to select the five values $df_\mu, df_\sigma, df_\nu, df_\tau, \xi$

- by trial and error
- minimize the generalized Akaike information criterion, GAIC($\sharp$)
- minimize the the validation global deviance VGD
- using local selection criteria, i.e. CV, ML

Diagnostics should be used in all above cases

gamlss

GAMLSS functions for centile estimation

`lms()` automatic selection of an appropriate LMS method

`centiles()` to plot centile curves against an x-variable.

`calibration()` adjust for sample quantiles to create centiles

`centiles.com()` to compare centiles curves for more than one `object`.

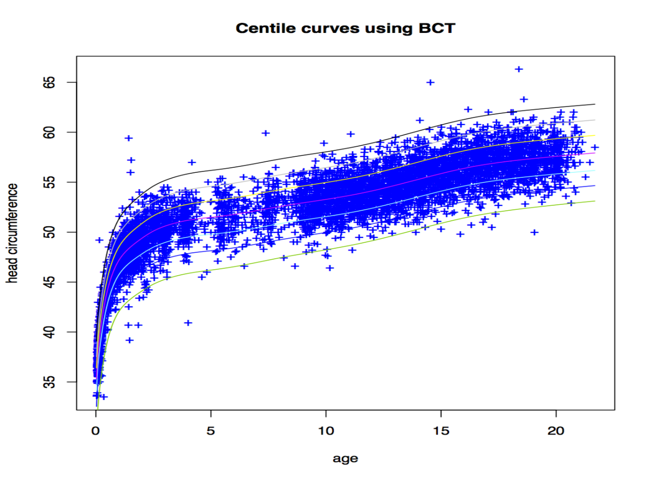
`centiles.split()` as for `centiles()`, but splits the plot at specified values of x.

`centiles.fan()` fan plot as in `centiles()`.

`centiles.pred()` to predict and plot centile curves for new x-values.

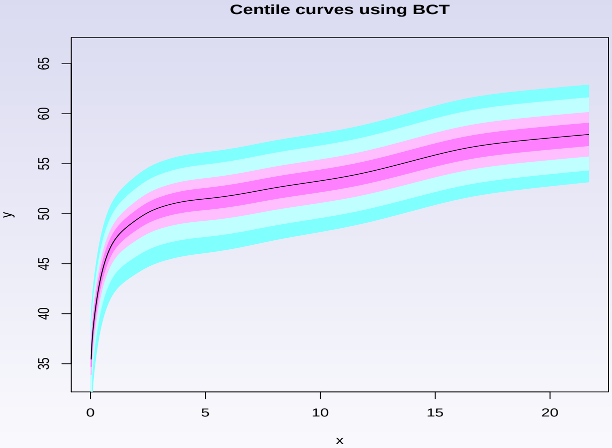
`fitted.plot()` to plot fitted values for all the parameters against an x-variable

The head circumference data: `centiles()`

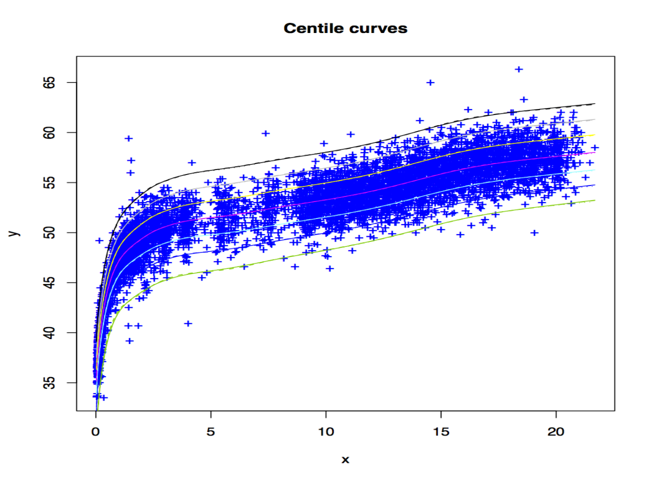


gamlss

The head circumference data: `centiles.fan()`

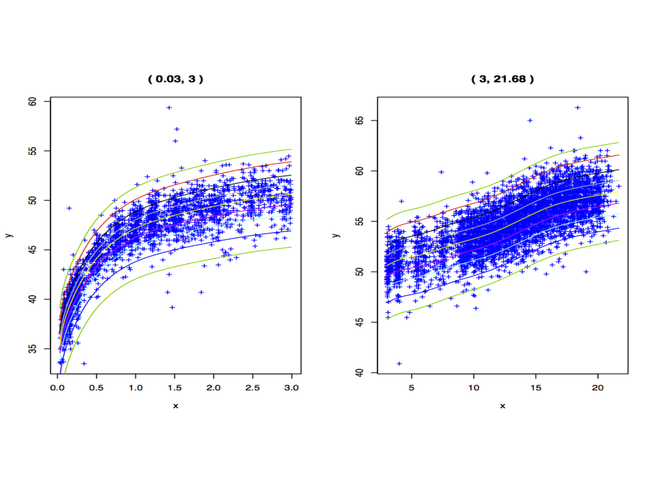


The head circumference data: comparison of the centiles, `centiles.com()`



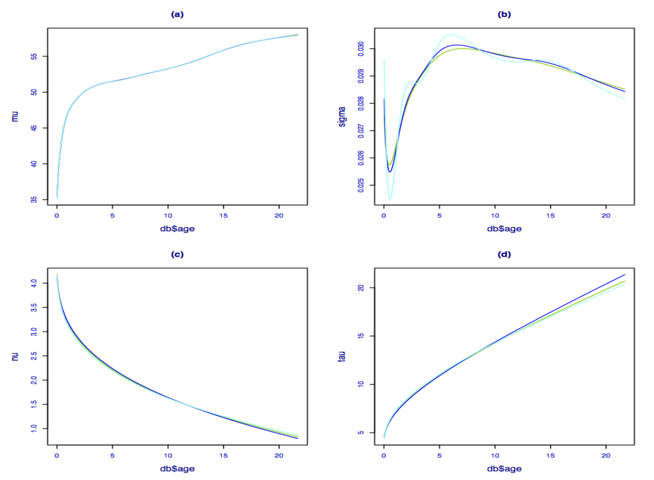
gamlss

The head circumference data: `centiles.split()`



gamlss

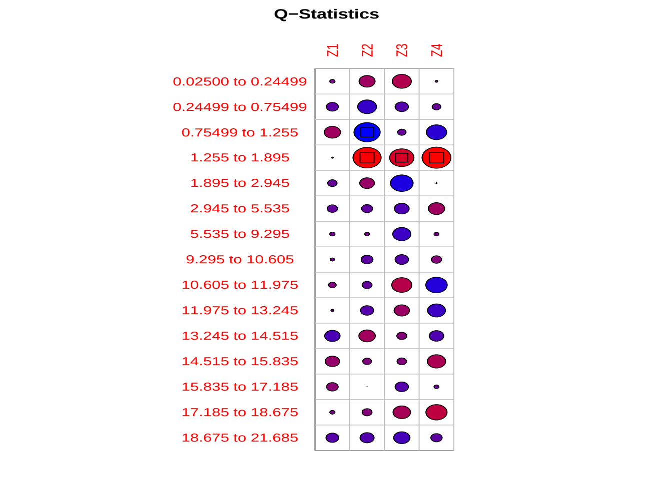
The head circumference data: `fitted.plot()`



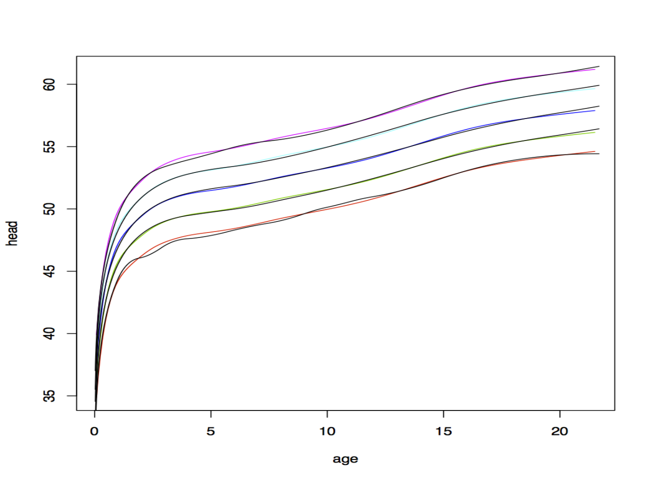
The head circumference data: The Q statistics

	Z1	Z2	Z3	Z4	Agostino	N
0.02500 to 0.24499	0.09	0.85	1.21	0.02	1.47	477
0.24499 to 0.75499	-0.48	-1.18	-0.58	-0.25	0.40	473
0.75499 to 1.255	0.88	-2.27	-0.24	-1.37	1.94	467
1.255 to 1.895	0.01	2.62	1.96	2.76	11.47	460
1.895 to 2.945	-0.29	0.72	-1.70	-0.01	2.90	473
2.945 to 5.535	-0.35	-0.40	-0.72	0.88	1.30	466
....						
15.835 to 17.185	0.45	0.00	-0.59	-0.08	0.35	466
17.185 to 18.675	0.09	0.32	1.01	1.48	3.23	472
18.675 to 21.685	-0.54	-0.65	-0.88	-0.42	0.96	467
TOTAL Q stats	2.85	16.90	15.43	18.04	33.46	7040
df for Q stats	1.95	12.01	12.35	13.00	25.35	0
p-val for Q stats	0.23	0.15	0.24	0.16	0.13	0

The head circumference data: `Q.stats()`



Comparing GAMLSS and QR: fitted centiles



gamlss

Conclusions: GAMLSS

- Unified framework for univariate regression type of models
- Allows any distribution for the response variable Y
- Models all the parameters of the distribution of Y
- Allows a variety of additive terms in the models for the distribution parameters
- The fitted algorithm is modular, where different components can be added easily
- Models can be fitted easily and fast
- Explanatory tool to find appropriate set of models
- It deals with overdispersion, skewness and kurtosis

End

END

for more information see

www.gamlss.org

gamlss