The `gamlss` packages

Mikis Stasinopoulos[1] Bob Rigby[1]

[1]STORM, London Metropolitan University

GAMLSS: Short course Campina Grande, Brazil, Jul 2013

1. The R packages
2. Demonstration
3. The gamlss package
    - Fitting or Updating a Model
    - Extracting Information
    - Selecting a Model
    - Diagnostics
    - Centile estimation
    - Other useful functions
4. The gamlss() function
5. End

The R packages

gamlss	the original package
gamlss.dist	all `gamlss.family` distributions
gamlss.data	different sets of data
gamlss.demo	demos for distributions and smoothing
gamlss.nl	non-linear term fitting
gamlss.tr	generating truncated distributions
gamlss.cens	for censored (left, right or interval) response variables
gamlss.mx	finite mixtures distributions and random effects
gamlss.add	for extra additive terms
gamlss.util	for extra utilities

The GAMLSS framework packages can be downloaded and installed from CRAN, the R library at `http://www.r-project.org/`. Test versions may be found at the GAMLSS web site at `http://www.gamlss.com/`.

Demonstration

```
library(gamlss)
data(abdom)
plot(y~x, col="blue", xlab="age", ylab="circumference",
    data=abdom)
............................................................
SEE  FILE chaper-3-rscript.r
............................................................
AIC(abd1,abd2,abd3, abd5, abd6,abd7)
GAIC(abd1,abd2,abd3, abd5, abd6,abd7)
GAIC(abd1,abd2,abd3, abd5, abd6,abd7, k=log(length(abdom$y)
```

The gamlss package

Classification of the available functions

- Fitting or Updating a Model
- Extracting Information from the Fitted Model
- Selecting a Model
- Plotting and Diagnostics
- Centile Estimation

gamlss

Fitting or Updating a Model

`gamlss()`	for fitting and creating a `gamlss` object
`refit()`	to refit a `gamlss` object (i.e. continue iterations)
`update()`	to update a given `gamlss` model object
`gamlssML()`	fitting a parametric distribution to a single (response) variable
`histDist()`	to fit and plot a parametric distribution
`fitDist()`	select a parametric distribution from an appropriate list

Extracting Information from the Fitted Model

`GAIC()`	generalised Akaike information criterion (or AIC)
`coef()`	the linear coefficients
`deviance()`	the global deviance $-2 \log L$
`fitted()`	the fitted values for a distribution parameter
`predict()`	to predict from new data individual distribution parameter values
`predictAll()`	to predict from new data all the distribution parameter values
`print()`	: to print a `gamlss` object
`residuals()`	to extract the normalised (randomised) quantile residuals
`summary()`	to summarise the fit in a `gamlss` object
`vcov()`	to extract the variance-covariance matrix of the beta estimates.

Selecting a Model

`add1()` `drop1()`	to add or drop a single term
`find.hyper()`	to find the hyper-parameters
`stepGAIC()`	to select explanatory terms in one parameter
`stepGAICAll.A()`	to select explanatory terms in all the parameters (strategy A)
`stepGAICAll.B()`	to select explanatory terms in all the parameters (strategy B)
`stepTGD()`	selecting variables using a test set the global deviance for new (test) data set given a fitted gamlss model.

Diagnostics

`plot()`	a plot of four graphs for the normalized (randomized) quantile residuals
`pdf.plot()`	for plotting the pdf functions for a given fitted `gamlss` object or a given `gamlss.family` distribution
`Q.stats()`	for printing and plotting the Q statistics of Royston and Wright (2000).
`rqres.plot()`	for plotting QQ-plots of different realisations of randomised residuals (for discrete distributions)
`wp()`	worm plot of the residuals from a fitted `gamlss` object
`dtop()`	detrended Own's plot of the residuals

Centile estimation

`centiles()`	to plot centile curves against an x-variable.
`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.pred()`	to predict and plot centile curves for new x-values.
`centiles.fan()`	fan plot od centile curves
`fitted.plot()`	to plot fitted values for all the parameters against an x-variable
`lms()`	a function trying to automate the process of fitting growth curves

Other useful functions

`prof.dev()`	the profile global deviance of one of the distribution parameters
`prof.term()`	for plotting the profile global deviance of one of the model (beta) parameters
`show.link()`	for showing available link functions
`term.plot()`	for plotting additive (smoothing) terms
`gen.likelihood()`	generates the likelihood from a GAMLSS fitted model [used in vcov()]

The gamlss() function

```
gamlss( formula = ~1, sigma.formula}= ~1,
      nu.formula = ~1, tau.formula = ~1,
      family = NO(),
      data = sys.parent(), weights = NULL,
      contrasts = NULL, method = RS(),  start.from = NULL,
      mu.start = NULL, sigma.start = NULL,
      nu.start = NULL, tau.start = NULL,
      mu.fix = FALSE, sigma.fix = FALSE, nu.fix = FALSE,
      tau.fix = FALSE, control = gamlss.control(...),
        i.control = glim.control(...), ...)
```

Arguments of the gamlss() function

formula $= y \sim x1 + x3$
sigma.fo $= \sim x1$
nu.fo $= \sim x2$
tau.fo $= \sim 1$
data = abdom
family = LO
weights = freq
method = mixed(10,50)
control = gamlss.control(trace=FALSE)

Starting values

Generally are not needed:

`start.mu= 2` or `start.mu=fitted(m1, "mu")`

The same applies for other parameters
`start.sigma` , `start.nu`, `start.tau`

Starting from a previous model

`start.from= model1`

END

for more information see

www.gamlss.org