The gamlss packages

Mikis Stasinopoulos[1] Bob Rigby[1]

[1]STORM, London Metropolitan University

GAMLSS: Short course Cordoba, Argentina, Oct 2012

gamlss

gamlss

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/.

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

gamlss

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.

gamlss

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

gamlss

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

gamlss

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

gamlss

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(...), ...)
```

gamlss

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)

gamlss

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`

gamlss

Demonstration

```
library(gamlss)
data(abdom)
plot(y~x, col="blue", xlab="age", ylab="circumference",
    data=abdom)

# fitting polynomials
abd0<-gamlss(y~poly(x,3), family=NO, data=abdom)
abd00<-gamlss(y~x+I(x^2)+I(x^3), family=NO, data=abdom)
abd00
summary(abd00)

# fitting P-splines
abd1<-gamlss(y~pb(x), family=NO, data=abdom)
abd1
summary(abd1)
```

gamlss

Demonstration (continue)

```
# Fitting a model for sigma
abd2 <- gamlss(y~pb(x),sigma.formula=~pb(x),family=NO,
       data=abdom)
plot(abd2)

# term plot with partial residuals
term.plot(abd2, what="mu", se=TRUE, partial=TRUE)
term.plot(abd2, what="sigma", se=TRUE, partial=TRUE)

# worm plots
wp(abd2)
wp(abd2,ylim.all=1.5)
```

gamlss

Demonstration (continue)

```
# fitting cs()
abd3 <- gamlss(y ~ cs(x), sigma.formula = ~cs(x),
               data = abdom, family = NO)
# fitting loess
 abd4 <- gamlss(y~lo(x,span=.4),sigma.formula=~lo(x,span=.4)
               data=abdom, family=NO)

# fitting different distribution
abd5 <- gamlss(y~pb(x),sigma.formula=~pb(x), data=abdom,
                family=TF)
GAIC(abd1,abd2,abd3,abd4, abd5)
GAIC(abd1,abd2,abd3,abd4, abd5, k=3 )
```

gamlss

Demonstration (continue)

```
# plotting a continuous distribution
PPP <- par(mfrow=c(2,2))
plot(function(y) dGA(y, mu=10 ,sigma=0.3), 0.1, 25)
plot(function(y) pGA(y, mu=10 ,sigma=0.3), 0.1, 25)
plot(function(y) qGA(y, mu=10 ,sigma=0.3), 0, 1)
hist(rGA(100,mu=10,sigma=.3))
par(PPP)
```

gamlss

Demonstration (continue)

```
PPP <- par(mfrow=c(2,2))
plot(function(y) dNBI(y, mu = 10, sigma =0.5 ),
            from=0, to=40, n=40+1, type="h",
                main="pdf", ylab="pdf(x)")
cdf <- stepfun(0:39, c(0, pNBI(0:39, mu=10, sigma=0.5 ))
              , f = 0)
plot(cdf,main="cdf", ylab="cdf(x)", do.points=FALSE )
invcdf <-stepfun(seq(0.01,.99,length=39), qNBI(seq(0.01,.99
          length=40), mu=10, sigma=0.5 ), f = 0)
plot(invcdf, main="inverse cdf",ylab="inv-cdf(x)",
             do.points=FALSE )
tN <- table(Ni <- rNBI(1000,mu=5, sigma=0.5))
r <- barplot(tN, col='lightblue')
```

gamlss

END

for more information see

www.gamlss.org

gamlss