Package ‘ssym’

April 22, 2023

Type Package

Title Fitting Semi-Parametric log-Symmetric Regression Models

Version 1.5.8

Date 2023-04-21

Encoding UTF-8

Author Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

Maintainer Luis Hernando Vanegas <hvanegasp@gmail.com>

Description Set of tools to fit a semi-parametric regression model suitable for analysis of data sets in which the response variable is continuous, strictly positive, asymmetric and possibly, censored. Under this setup, both the median and the skewness of the response variable distribution are explicitly modeled by using semi-parametric functions, whose non-parametric components may be approximated by natural cubic splines or P-splines. Supported distributions for the model error include log-normal, log-Student-t, log-power-exponential, log-hyperbolic, log-contaminated-normal, log-slash, Birnbaum-Saunders and Birnbaum-Saunders-t distributions.

Depends GIGrvg, numDeriv, normalp, Formula, survival

Imports stats, grDevices, sandwich, graphics, methods, utils

Suggests NISTnls, gam, sn, MASS

NeedsCompilation no

Repository CRAN

Date/Publication 2023-04-22 00:50:12 UTC

RoxygenNote 7.2.3

R topics documented:

`ssym-package` *Fitting Semiparametric Log-symmetric Regression Models*

Description

This package allows to fit a semi-parametric regression model suitable for analysis of data sets in which the response variable is continuous, strictly positive, asymmetric and possibly, censored.

Details

Package:	ssym
Type:	Package
Version:	1.5.7
Date:	2016-10-15
License:	GPL-2 | GPL-3

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

Maintainer: Luis Hernando Vanegas

References

Vanegas, L.H. and Paula, G.A. (2015) A semiparametric approach for joint modeling of median and skewness. TEST 24, 110-135.

Vanegas, L.H. and Paula, G.A. (2016) Log-symmetric distributions: statistical properties and parameter estimation. Brazilian Journal of Probability and Statistics 30, 196-220.

Vanegas, L.H. and Paula, G.A. (2016) An extension of log-symmetric regression models: R codes and applications. Journal of Statistical Computation and Simulation 86, 1709-1735.

Examples

```
data("Snacks", package="ssym")
fit <- ssym.l(log(texture) ~ type + ncs(week) | type, data=Snacks,
              family='Student', xi=15)
summary(fit)
```

`Baboons` *What time do the baboons come down from the trees?*

Description

This data set arises in the course of analyzing data on the ecology of baboons in East Africa. The data consist on descent times of baboons (in hours since the day began) or censoring times and the (left) censoring status.

Usage

```
data(Baboons)
```

Format

A data frame with 152 observations on the following 2 variables.

`t` descent times of baboons or censoring times, in hours since the day began.

`cs` (left) censoring status.

References

Wagner, S.S. and Altmann, S.A. (1973) What time do the baboons come down from the trees? (An estimation problem). Biometrics, 29: 623-635.

Biaxial *Brown and Miller's Biaxial Fatigue*

Description

This data set describes the life of a metal piece in cycles to failure. The response is the number of cycles to failure and the explanatory variable is the work per cycle.

Usage

```
data(Biaxial)
```

Format

A data frame with 46 observations on the following 2 variables.

`Work` work per cycle.

`Life` number of cycles to failure.

References

J.R. Rieck and J.R. Nedelman (1991) A log-linear model for the Birnbaum-Saunders distribution, Technometrics 33, 51:60.

Examples

```
data("Biaxial", package="ssym")
plot(Biaxial$Work, Biaxial$Life, type="p",
     cex=0.3, lwd=3, ylab="Life", xlab="Work per cycle",
 main="Brown and Miller's Biaxial Fatigue Data")
```

BIC.ssym *BIC.ssym*

Description

BIC.ssym calculates the goodness-of-fit statistic BIC from an object of class “"ssym".

Claims *Personal Injure Insurance*

Description

This data set contains information on 540 settled personal injury insurance claims from an Australian insurance company, which is a sample of the original data set. These claims had legal representation were obtained for accidents that occurred from January 1998 to June 1999.

Usage

```
data(Claims)
```

Format

A data frame with 540 observations on the following 2 variables.

`total` amount of paid money by an insurance policy in thousands of Australian dollars.

`accmonth` month of occurrence of the accident coded 103 (January 1998) through to 120 (June 1999).

`op_time` operational time in percentage.

References

de Jong P, Heller GZ. Generalized Linear Models for Insurance Data. Cambridge University Press: Cambridge, England, 2008.

Paula, G.A., Leiva, V., Barros, M. and Liu, S. (2012) Robust statistical modeling using the Birnbaum-Saunders-t distribution applied to insurance distribution, Applied Stochastic Model in Business and Industry, 28:16-34.

Examples

```
data("Claims", package="ssym")
plot(Claims$op_time, Claims$total, type="p", cex=0.3, lwd=3,
     ylab="Amount of paid money", xlab="Operational time",
 main="Personal Injure Insurance Data")
```

coef.ssym *coef.ssym*

Description

coef.ssym extracts the parameter estimates for both submodels from an object of class “"ssym".

`envelope` *Building of normal probability plots with simulated envelope of the deviance-type residuals.*

Description

envelope is used to calculate and display normal probability plots with simulated envelope of the deviance-type residuals.

Usage

```
envelope(object, reps, conf, xlab.mu, ylab.mu, main.mu, xlab.phi, ylab.phi, main.phi)
```

Arguments

`object`	an object of the class `ssym`. This object returned from the call to `ssym.l()` or `ssym.nl()`.
`reps`	a positive integer representing the number of iterations in which the simulated envelopes are based. Default is `reps=25`.
`conf`	value within the interval (0,1) that represents the confidence level of the simulated envelopes. Default is `conf=0.95`.
`xlab.mu`	character. An optional label for the x axis for the graph of the deviance-type residuals for the median submodel.
`ylab.mu`	character. An optional label for the y axis for the graph of the deviance-type residuals for the median submodel.
`main.mu`	character. An optional overall title for the plot for the graph of the deviance-type residuals for the median submodel.
`xlab.phi`	character. An optional label for the x axis for the graph of the deviance-type residuals for the skewness submodel.
`ylab.phi`	character. An optional label for the y axis for the graph of the deviance-type residuals for the skewness submodel.
`main.phi`	character. An optional overall title for the plot for the graph of the deviance-type residuals for the skewness submodel.

Details

Objects of the class `ssym` obtained from the application of `ssym.l2()` are not supported. The smoothing parameters are assumed to be known.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Atkinson, A. C. (1985) Plots, transformations and regression: an introduction to graphical methods of diagnostic regression analysis. Oxford Science Publications, Oxford.

Examples

```
###################################################################################
################# Blood flow Data - a log-power-exponential model #################
###################################################################################
#data("la", package="gamlss.nl")
#fit <- ssym.nl(log(PET60) ~ log(bflow) + log(1+b1*exp(-b2/bflow)) | bflow,
#       data=la, start=c(b1=-0.6,b2=98), family="Powerexp", xi=-0.45)
#summary(fit)
#
################## Simulated envelopes ##################
#envelope(fit,reps=50,conf=0.99)
```

Erabbits — *Age and Eye Lens Weight of Rabbits in Australia*

Description

The dry weight of the eye lens was measured for 71 free-living wild rabbits of known age. Eye lens weight tends to vary much less with environmental conditions than does total body weight, and therefore may be a much better indicator of age.

Usage

```
data(Erabbits)
```

Format

A data frame with 71 observations on the following 2 variables.

`age` age of rabbit, in days.

`wlens` dry weight of eye lens, in milligrams.

References

Dudzinski, M.L. and Mykytowycz, R. (1961) The eye lens as an indicator of age in the wild rabbit in Australia. CSIRO Wildlife Research, 6: 156-159.

Ratkowsky, D. A. (1983). Nonlinear Regression Modelling. Marcel Dekker, New York.

Wei, B. C. (1998). Exponential Family Nonlinear Models. Springer, Singapore.

Examples

```
data("Erabbits", package="ssym")
plot(Erabbits$age, Erabbits$wlens, type="p", cex=0.3, lwd=3,
     ylab="Dry weight of eye lens (in milligrams)",
 xlab="Age of the animal (in days)")
```

---

estfun.ssym *estfun.ssym*

Description

estfun.ssym extracts the score functions evaluated at observed data and estimated parameters from an object of class `ssym`.

---

extra.parameter *Tool that supports the estimation of the extra parameter.*

Description

extra.parameter is used to plot a graph of the behaviour of the overall goodness-of-fit statistic and $-2\mathsf{L}(\hat{\theta})$ versus the extra parameter ζ in the interval/region defined by the arguments `lower` and `upper`. These graphs may be used to choosing the extra parameter value.

Usage

```
extra.parameter(object, lower, upper, grid)
```

Arguments

`object`	an object of the class `ssym`. This object is returned by the call to `ssym.l()`, `ssym.nl()` or `ssym.l2()`
`lower`	lower limit(s) of the interest interval/region for the extra parameter.
`upper`	upper limit(s) of the interest interval/region for the extra parameter.
`grid`	Number of values of the extra parameter where the overall goodness-of-fit statistic and $-2\mathsf{L}(\hat{\theta})$ are evaluated.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Vanegas, L.H. and Paula, G.A. (2015b) Log-symmetric distributions: statistical properties and parameter estimation. Brazilian Journal of Probability and Statistics (to appear)

Examples

```
#####################################################################################
############### Textures of snacks Data - a log-Student-t model  #################
#####################################################################################
#data("Snacks", package="ssym")
#fit <- extra.parameter(log(texture) ~ type + ncs(week) | type, data=Snacks,
#        family='Student', xi=10)
#summary(fit)
#
############################ Extra parameter ###########################
#extra.parameter(fit,5,50)

#####################################################################################
################## Biaxial Fatigue Data - a Birnbaum-Saunders model   #############
#####################################################################################
#data("Biaxial", package="ssym")
#fit <- ssym.nl(log(Life) ~ b1*Work^b2, start=c(b1=16, b2=-0.25),
#                data=Biaxial, family='Sinh-normal', xi=1.54)
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,1.3,1.8)
```

`fitted.ssym` *fitted.ssym*

Description

fitted.ssym extracts the fitted values for both submodels from an object of class “"ssym".

`gdp` *Gross Domestic Product (per capita)*

Description

This dataset corresponds to the per capita gross domestic product (current US$) of 190 countries during 2010.

Usage

```
data(gdp)
```

Format

A data frame with 190 observations on the following 2 variables.

`Country` Country.

`gdp2010` The per capita gross domestic product (current US$).

References

World Bank's DataBank website (http://databank.worldbank.org/data/).

Examples

```
data("gdp", package="ssym")
par(mfrow=c(1,2))
hist(gdp$gdp2010, xlim=range(gdp$gdp2010), ylim=c(0,0.00015), prob=TRUE, breaks=55,
     col="light gray",border="dark gray", xlab="GDP per capita 2010", main="Histogram")
plot(ecdf(gdp$gdp2010), xlim=range(gdp$gdp2010), ylim=c(0,1), verticals=TRUE,
     do.points=FALSE, col="dark gray", xlab="GDP per capita 2010",
 main="Empirical Cumulative Distribution Function")
```

`influence.ssym` *Tool to perform sensitivity analysis on the fitted model using local influence measures.*

Description

influence extracts from a object of class "ssym" the local influence measures and displays their graphs versus the index of the observations.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Cook, R.D. (1986). Assessment Local Influence (with discussion). Journal of the Royal Statistical Society Series B (Methodological). 48, 133-169.

Poon, W.Y. and Poon, Y.S. (1999). Conformal Normal Curvature and Assessment of Local Influence. Journal of the Royal Statistical Society Series B (Methodological). 61, 51-61.

`itpE` *itpE*

Description

itpE performs the iterative process to fit models whose error distribution can be obtained as a power mixture of the log-normal distribution..

`itpE2` *itpE2*

Description

itpE2 runs the E-step of the iterative process to fit models whose error distribution can be obtained as a shape mixture of the Birnbaum-Saunders distribution.

`itpE3` *itpE3*

Description

itpE3 performs the iterative process to fit models whose error distribution cannot be obtained as a shape mixture of log-normal or Birnbaum-Saunders distributions.

`itpEC2` *itpEC2*

Description

itpEC2 performs the iterative process to fit models under the presence of right-censored samples, wher the error distribution can be obtained as a power mixture of the log-normal distribution.

`logLik.ssym` *logLik.ssym*

Description

logLik.ssym extracts the value of the log-likelihood function avaliated at observed data and parameter estimates from an object of class “"ssym".

---

myeloma *Survival times for multiple myeloma patients*

Description

The problem is to relate survival times for multiple myeloma patients to a number of prognostic variables.

Usage

```
data("myeloma")
```

Format

A data frame with 65 observations on the following 7 variables.

`t` survival times, in months.

`event` censoring status.

`x1` logarithm of a blood urea nitrogen measurement at diagnosis.

`x2` hemoglobin measurement at diagnosis.

`x3` age at diagnosis.

`x4` sex: 0, male; 1, female.

`x5` serum calcium measurement at diagnosis.

References

J.F. Lawless (2002) Statistical Models and Methods for Lifetime Data, Wiley, New York. A.P. Li, Z.X. Chen and F.C. Xie (2012) Diagnostic analysis for heterogeneous log-Birnbaum-Saunders regression models, Statistics and Probability Letters 82, 1690:1698.

---

ncs *Tool to build the basis matrix and the penalty matrix of natural cubic splines.*

Description

ncs builds the basis matrix and the penalty matrix to approximate a smooth function using a natural cubic spline.

Usage

```
ncs(xx, lambda, nknots, all.knots)
```

Arguments

`xx`	the explanatory variable.
`lambda`	an optional positive value that represents the smoothing parameter value.
`nknots`	an optional positive integer that represents the number of knots of the natural cubic spline. Default is $m = [n^{\frac{1}{3}}] + 3$. The knots are located at the quantiles of order $0/(m-1), 1/(m-1), \ldots, (m-1)/(m-1)$ of xx.
`all.knots`	logical. If `TRUE`, the set of knots and the set of different values of xx coincide. Default is `FALSE`.

Value

`xx`	the explanatory variable xx with the following attributes: set of knots, basis matrix, penalty matrix, smoothing parameter (if it was specified), and other interest matrices.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Lancaster, P. and Salkauskas, K. (1986) Curve and Surface Fitting: an introduction. Academic Press, London. Green, P.J. and Silverman, B.W. (1994) Nonparametric Regression and Generalized Linear Models, Boca Raton: Chapman and Hall.

Examples

```
n <- 300
t <- sort(round(runif(n),digits=1))

t2 <- ncs(t,all.knots=TRUE)
N <- attr(t2, "N") ## Basis Matrix
M <- attr(t2, "K") ## Penalty Matrix
knots <- attr(t2, "knots") ## Set of knots
```

`np.graph` *Tool to plot natural cubic splines or P-splines.*

Description

np.graph displays a graph of a fitted nonparametric effect, either natural cubic spline or P-spline, from an object of class `ssym`.

Usage

```
np.graph(object, which, var, exp, simul, obs, xlab, ylab, xlim, ylim, main)
```

Arguments

- `object` — an object of the class `ssym`. This object is returned from the call to `ssym.l()`, `ssym.nl()` or `ssym.l2()`.
- `which` — an integer indicating the interest submodel. For example, 1 indicates location submodel, and 2 indicates skewness (or relative dispersion) submodel.
- `var` — character. It allows to choosing the nonparametric effect using the name of the associated explanatory variable.
- `exp` — logical. If `TRUE`, the fitted nonparametric effect is plotted in exponential scale. Default is `FALSE`.
- `simul` — logical. If `TRUE`, the fitted nonparametric effect is plotted jointly with their 95% simultaneous confidence intervals. If `TRUE`, the fitted nonparametric effect is plotted jointly with their 95% pointwise confidence intervals. Default is `TRUE`.
- `obs` — logical. If `TRUE`, the fitted nonparametric effect is plotted jointly with the observed data. Default is `FALSE`.
- `xlab` — character. An optional label for the *x* axis.
- `ylab` — character. An optional label for the *y* axis.
- `xlim` — numeric. An optional range of values for the *x* axis.
- `ylim` — numeric. An optional range of values for the *y* axis.
- `main` — character. An optional overall title for the plot.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Lancaster, P. and Salkauskas, K. (1986) Curve and Surface Fitting: an introduction. Academic Press, London. Green, P.J. and Silverman, B.W. (1994) Nonparametric Regression and Generalized Linear Models, Boca Raton: Chapman and Hall. Eilers P.H.C. and Marx B.D. (1996). Flexible smoothing with B-splines and penalties. Statistical Science. 11, 89-121.

Examples

```
#data("Ovocytes", package="ssym")
#fit <- ssym.l(fraction ~ type + psp(time) | type + psp(time), data=Ovocytes,
#              family='Powerexp', xi=-0.55)
#
#par(mfrow = c(1,2))
#np.graph(fit, which=1, xlab="Time", main="Location")
#np.graph(fit, which=2, exp=TRUE, xlab="Time", main="Dispersion")
```

Ovocytes *Fraction of cell volume*

Description

This data set comes from an experiment comparing the responses of immature and mature goat ovocytes to an hyper-osmotic test. As a compound permeates, water reenters the cell, and the cell re-expands until the system reaches an osmotic equilibrium. The results are obtained using immature and ovulated (mature) ovocytes exposed to propanediol, a permeable compound. Then, the cell volume during equilibration is recorded at each time `t`.

Usage

```
data(Ovocytes)
```

Format

A data frame with 161 observations on the following 3 variables.

`type` stage of the goat ovocyte: *Mature* or *Immature*.

`time` time since exposition to propanediol.

`fraction` fraction of initial isotonic cell volume at any given time *t* during equilibration.

References

Huet, S., Bouvier, A., Gruet, M.A. and Jolivet, E. (1996). Statistical Tools for Nonlinear Regression. Springer, New York.

Le Gal F., Gasqui P., Renard J.P. (1994) Differential Osmotic Behavior of Mammalian Oocytes before and after Maturation: A Quantitative Analysis Using Goat Oocytes as a Model. Cryobiology, 31: 154-170.

Huet S., Bouvier A., Gruet M.A., Jolivet E. (1996) Statistical Tools for Nonlinear Regression. Springer-Verlag: New York.

Examples

```
data("Ovocytes", package="ssym")
xl <- "Time"
yl <- "Fraction of Cell Volume"
mm <- "Fraction of Cell Volume for Mature and Immature Goat Ovocytes"
rx <- range(Ovocytes$time)
ry <- range(Ovocytes$fraction)
plot(Ovocytes$time[Ovocytes$type=='Mature'], Ovocytes$fraction[Ovocytes$type=='Mature'],
     xlim=rx, ylim=ry, type="p", cex=0.5, lwd=1, ylab="", xlab="")
par(new=TRUE)
plot(Ovocytes$time[Ovocytes$type=='Immature'], Ovocytes$fraction[Ovocytes$type=='Immature'],
     xlim=rx, ylim=ry, type="p", cex=0.5, lwd=2, ylab=yl, xlab=xl, main=mm)
legend(rx[1], ry[2], pt.lwd=c(1,2), bty="n", legend=c("Mature","Immature"), pt.cex=0.5, pch=1)
```

plot.ssym *plot.ssym*

Description

plot.ssym produces the graph in which the goodness-of-fit statistic Υ is based. This function also displays graphs of the deviance-type residuals versus the fitted values for the median and the skewness (or the relative dispersion) submodels. Under the presence of an uncensored sample, the function `plot()` produces a graph of the standardized individual-specific weights versus the ordinary residuals (i.e., a graph of $\rho(\hat{z}_k)$ versus $\hat{z}_k$, $k = 1, \ldots, n$), and under the presence of a right-censored sample, the function `plot()` produces a graph of the survival function of the error distribution.

print.ssym *print.ssym*

Description

print.ssym displays a summary (simpler than summary.ssym) of the fitted model including parameter estimates, (approximate) associated standard errors and goodness-of-fit statistics from an object of class `ssym`.

psp *Tool to build the basis matrix and the penalty matrix of P-splines.*

Description

psp builds the basis matrix and the penalty matrix to approximate a smooth function using a P-spline.

Usage

```
psp(xx, lambda, b.order, nknots, diff)
```

Arguments

`xx`	the explanatory variable.
`lambda`	an optional positive value that represents the smoothing parameter value.
`b.order`	an optional positive integer that specifies the degree of the B-spline basis matrix. Default is 3.
`nknots`	an optional positive integer that represents the number of internal knots of the P-spline. Default is $m = [n^{\frac{1}{3}}] + 3$. The knots are located at the quantiles of order $0/(m-1), 1/(m-1), \ldots, (m-1)/(m-1)$ of xx.
`diff`	an optional positive integer that specifies the order of the difference penalty term. Default is 2.

Value

xx	the explanatory variable xx with the following attributes: set of knots, B-spline basis matrix, penalty matrix and smoothing parameter (if it was specified).

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Eilers P.H.C. and Marx B.D. (1996). Flexible smoothing with B-splines and penalties. Statistical Science. 11, 89-121.

Examples

```
n <- 300
t <- sort(round(runif(n),digits=2))

t2 <- psp(t, diff=3)
N <- attr(t2, "N") ## B-spline basis matrix
M <- attr(t2, "K") ## Penalty Matrix
knots <- attr(t2,"knots") ## Set of knots
```

---

residuals.ssym *residuals.ssym*

Description

residuals.ssym extracts the deviance-type residuals for both submodels from an object of class “"ssym".

---

rvgs *Random generation for some symmetric continuous distributions.*

Description

rvgs is used to random generation from some standard symmetric continuous distributions.

Usage

```
rvgs(n, family, xi)
```

Arguments

n	number of observations.
family	Supported families include *Normal*, *Student*, *Contnormal*, *Powerexp*, *Hyperbolic*, *Slash*, *Sinh-normal* and *Sinh-t*, which correspond to normal, Student-t, contaminated normal, power exponential, symmetric hyperbolic, slash, sinh-normal and sinh-t distributions, respectively.
xi	a numeric value or numeric vector that represents the extra parameter value of the specified distribution.

Value

x	a vector of n observations.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

Examples

```
m1 <- "Standard Sinh-t distributions"
n <- 1000000
xi <- c(10,6,4)
plot(density(rvgs(n,"Sinh-t",xi=c(25,10))), xlim=c(-4.5,4.5), ylim=c(0,0.3), xlab="",
     ylab="", col=1, main="")
par(new=TRUE)
plot(density(rvgs(n,"Sinh-t",xi=c(25,6))), xlim=c(-4.5,4.5), ylim=c(0,0.3), xlab="",
     ylab="", col=2, main="")
par(new=TRUE)
plot(density(rvgs(n,"Sinh-t",xi=c(25,4))), xlim=c(-4.5,4.5), ylim=c(0,0.3), xlab="y",
     ylab="f(y)", main=m1, col=3)
legend(-4, 0.3, bty="n", legend=paste("xi = (",25,",",xi,")"), col=1:4, lty=1)
```

Snacks *Textures of five different types of snacks*

Description

This data set comes from an experiment developed in the School of Public Health - University of Sao Paulo, in which four different forms of light snacks (denoted by B, C, D, and E) were compared with a traditional snack (denoted by A) for 20 weeks. For the light snacks, the hydrogenated vegetable fat (hvf) was replaced by canola oil using different proportions: B (0% hvf, 22% canola oil), C (17% hvf, 5% canola oil), D (11% hvf, 11% canola oil) and E (5% hvf, 17% canola oil); A (22% hvf, 0% canola oil) contained no canola oil. The experiment was conducted such that a random sample of 15 units of each snack type was analyzed in a laboratory in each even week to measure various variables. A total of 75 units was analyzed in each even week; with 750 units being analyzed during the experiment.

Usage

```
data(Snacks)
```

Format

A data frame with 750 observations on the following 3 variables.

`texture` texture of the snack unit.

`type` a factor with levels 1-5 which correspond to A-E types of snacks.

`week` week in which the snack unit was analyzed.

References

Paula, G.A., de Moura, A.S., Yamaguchi, A.M. (2004) Sensorial stability of snacks with canola oil and hydrogenated vegetable fat. Technical Report. Center of Applied Statistics, University of Sao Paulo (in Portuguese).

Paula, G.A. (2013) On diagnostics in double generalized linear models. Computational Statistics and Data Analysis, 68: 44-51.

Examples

```
data("Snacks", package="ssym")
boxplot(log(Snacks$texture) ~ Snacks$type, xlab="Type of Snack", ylab="Log(texture)")
```

`ssym.l` *Fitting Semi-parametric Log-symmetric Regression Models*

Description

ssym.l is used to fit a semi-parametric regression model suitable for analysis of data sets in which the response variable is continuous, strictly positive, and asymmetric. Under this setup, both median and skewness of the response variable distribution are explicitly modeled through semi-parametric functions, whose nonparametric components may be approximated by natural cubic splines or P-splines.

Usage

```
ssym.l(formula, family, xi, data, epsilon, maxiter, subset, link.mu, link.phi,
       local.influence, spec, std.out)
```

Arguments

- `formula` a symbolic description of the systematic component of the model to be fitted. See details for further information.
- `family` a description of the (log) error distribution to be used in the model. Supported families include `Normal`, `Student`, `Contnormal`, `Powerexp`, `Hyperbolic`, `Slash`, `Sinh-normal` and `Sinh-t`, which correspond to normal, Student-t, contaminated normal, power exponential, symmetric hyperbolic, slash, sinh-normal and sinh-t distributions, respectively.
- `xi` a numeric value or numeric vector that represents the extra parameter of the specified error distribution.
- `data` an optional data frame, list or environment containing the variables in the model.
- `epsilon` an optional positive value, which represents the convergence criterion. Default value is 1e-07.
- `maxiter` an optional positive integer giving the maximal number of iterations for the estimating process. Default value is 1e03.
- `subset` an optional expression that specifies a subset of individuals to be used in the fitting process.
- `link.mu` an optional character that specifies the link function of the median submodel.
- `link.phi` an optional character that specifies the link function of the skewness submodel.
- `local.influence` logical. If TRUE, local influence measures under two perturbation schemes are calculated. Default is `FALSE`.
- `spec` an optional character. The smoothing parameter is estimated by minimizing a overall goodness-of-fit criterion such as AIC or BIC. `spec` is an optional string to specify the goodness-of-fit measure to be used. Default value is `AIC`.
- `std.out` logical. If FALSE, just a reduced set of attributes is returned by the model-fitting function. Default is `TRUE`.

Details

The argument *formula* comprises of three parts (separated by the symbols "~" and "|"), namely: observed response variable in log-scale, predictor of the median submodel (having logarithmic link) and predictor of the skewness (or the relative dispersion) submodel (having logarithmic link). An arbitrary number of nonparametric effects may be specified in the predictors. These effects are specified to be approximated by natural cubic splines or P-splines using the functions `ncs()` or `psp()`, respectively.

The iterative estimation process is based on the Fisher scoring and backfitting algorithms. Because some distributions such as log-Student-t, log-contaminated-normal, log-power-exponential, log-slash and log-hyperbolic may be obtained as a power mixture of the log-normal distribution, the expectation-maximization (EM) algorithm is applied in those cases to obtain a more efficient iterative process of parameter estimation. Furthermore, because the Birnbaum-Saunders-t distribution can be obtained as a scale mixture of the Birnbaum-Saunders distribution, the expectation-maximization algorithm is also applied in this case to obtain a more efficient iterative process of parameter estimation. The smoothing parameter is chosen by minimizing the AIC or BIC criteria.

The function `ssym.l()` calculates overall goodness-of-fit statistics, deviance-type residuals for both submodels, as well as local influence measures under the case-weight and response perturbation schemes.

Value

`theta.mu`	a vector of parameter estimates associated with the median submodel.
`theta.phi`	a vector of parameter estimates associated with the skewness (or the relative dispersion) submodel.
`vcov.mu`	approximate variance-covariance matrix associated with the median submodel.
`vcov.phi`	approximate variance-covariance matrix associated with the skewness (or the relative dispersion) submodel.
`weights`	final weights of the iterative process.
`lambdas.mu`	estimate of the smoothing parameter(s) associated with the nonparametric part of the median submodel.
`lambdas.phi`	estimate of the smoothing parameter(s) associated with the nonparametric part of the skewness (or the relative dispersion) submodel.
`gle.mu`	degrees of freedom associated with the nonparametric part of the median submodel.
`gle.phi`	degrees of freedom associated with the nonparametric part of the skewness (or the relative dispersion) submodel.
`deviance.mu`	a vector with the individual contributions to the deviance associated with the median submodel.
`deviance.phi`	a vector with the individual contributions to the deviance associated with the skewness (or the relative dispersion) submodel.
`mu.fitted`	a vector with the fitted values of the (in log-scale) median submodel.
`phi.fitted`	a vector with the fitted values of the skewness (or the relative dispersion) submodel.
`lpdf`	a vector of individual contributions to the log-likelihood function.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Vanegas, L.H. and Paula, G.A. (2015) A semiparametric approach for joint modeling of median and skewness. TEST 24, 110-135.

Vanegas, L.H. and Paula, G.A. (2016) Log-symmetric distributions: statistical properties and parameter estimation. Brazilian Journal of Probability and Statistics 30, 196-220.

Vanegas, L.H. and Paula, G.A. (2016) An extension of log-symmetric regression models: R codes and applications. Journal of Statistical Computation and Simulation 86, 1709-1735.

See Also

`ssym.nl`, `ssym.l2`

Examples

```
####################################################################################
######### Fraction of Cell Volume Data - a log-power-exponential model  ###########
####################################################################################
#data("Ovocytes", package="ssym")
#fit <- ssym.l(log(fraction) ~ type + psp(time) | type + psp(time),
#               data=Ovocytes, family='Powerexp', xi=-0.55, local.influence=TRUE)
#summary(fit)
#
################## Graph of the nonparametric effects ##################
#par(mfrow=c(1,2))
#np.graph(fit, which=1, exp=TRUE)
#np.graph(fit, which=2, exp=TRUE)
#
################## Graph of deviance-type residuals ##################
#plot(fit)
#
################### Simulated envelopes ##################
#envelope(fit)
#
################### Graph of local influence measures ##################
#out <- influence(fit)

####################################################################################
############### Textures of snacks Data - a log-Student-t model  ##################
####################################################################################
#data("Snacks", package="ssym")
#fit <- ssym.l(log(texture) ~ type + ncs(week) | type, data=Snacks,
#        family='Student', xi=15, local.influence=TRUE)
#summary(fit)
#
############################ Extra parameter ###########################
#extra.parameter(fit,5,50)
#
################### Graph of the nonparametric effect ##################
#np.graph(fit, which=1, exp=TRUE)
#
################### Graph of deviance-type residuals ##################
#plot(fit)
#
################### Simulated envelopes ##################
#envelope(fit)
#
################### Plot of influence measures ##################
#out <- influence(fit)

####################################################################################
################### Daphnia Data - a log-normal model #######################
####################################################################################
#data("daphnia", package="nlreg")
#fit <- ssym.l(log(time) ~ ncs(conc) | ncs(conc), data=daphnia, family="Normal")
```

```
#summary(fit)
#
################### Graph of the nonparametric effects ##################
#par(mfrow=c(1,2))
#np.graph(fit, which=1, exp=TRUE)
#np.graph(fit, which=2, exp=TRUE)
#
################### Simulated envelopes ##################
#envelope(fit)

###################################################################################
####################### gam.data - a Power-exponential model   ####################
###################################################################################
#data("gam.data", package="gam")
#
#fit <- ssym.l(y~psp(x),data=gam.data,family="Powerexp",xi=-0.5)
#summary(fit)
#
################## Graph of the nonparametric effect ##################
#np.graph(fit, which=1)
#
###################################################################################
######### Personal Injury Insurance Data - a Birnbaum-Saunders-t model   ##########
###################################################################################
#data("Claims", package="ssym")
#fit <- ssym.l(log(total) ~ op_time | op_time, data=Claims,
#        family='Sinh-t', xi=c(0.1,4), local.influence=TRUE)
#summary(fit)
#
################## Plot of deviance-type residuals ##################
#plot(fit)
#
################### Simulated envelopes ##################
#envelope(fit)
################## Plot of influence measures ##################
#out <- influence(fit)

###################################################################################
######### Body Fat Percentage Data - a Birnbaum-Saunders-t model   ##########
###################################################################################
#data("ais", package="sn")
#fit <- ssym.l(log(Bfat)~1, data=ais, family='Sinh-t', xi=c(4.5,4))
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,c(3,4),c(5,7))
#
################## Plot of the fitted model ##################
#id <- sort(ais$Bfat, index=TRUE)$ix
#par(mfrow=c(1,2))
#hist(ais$Bfat[id],xlim=range(ais$Bfat),ylim=c(0,0.1),prob=TRUE,breaks=15,
#     col="light gray",border="dark gray",xlab="",ylab="",main="")
#par(new=TRUE)
```

```
#plot(ais$Bfat[id],exp(fit$lpdf[id])/ais$Bfat[id],xlim=range(ais$Bfat),
#     ylim=c(0,0.1),type="l",xlab="",ylab="Density",main="Histogram")
#
#plot(ais$Bfat[id],fit$cdfz[id],xlim=range(ais$Bfat),ylim=c(0,1),type="l",
#     xlab="",ylab="",main="")
#par(new=TRUE)
#plot(ecdf(ais$Bfat[id]),xlim=range(ais$Bfat),ylim=c(0,1),verticals=TRUE,
#     do.points=FALSE,col="dark gray",ylab="Probability",xlab="",main="ECDF")

###################################################################################
################### ALCOA Aluminium Data - a log-slash model   ####################
###################################################################################

#data("alcoa", package="robustloggamma")
#alcoa2 <- data.frame(alcoa$dist[alcoa$label=="C"])
#colnames(alcoa2) <- "dist"
#
#fit <- ssym.l(log(dist) ~ 1, data=alcoa2, family="Slash", xi=1.212)
#
################## Plot of the fitted model ##################
#id <- sort(alcoa2$dist, index=TRUE)$ix
#par(mfrow=c(1,2))
#hist(alcoa2$dist[id],xlim=c(0,45),ylim=c(0,0.1),prob=TRUE,breaks=60,
#     col="light gray",border="dark gray",xlab="",ylab="",main="")
#par(new=TRUE)
#plot(alcoa2$dist[id],exp(fit$lpdf[id])/alcoa2$dist[id],xlim=c(0,45),
#ylim=c(0,0.1), type="l",xlab="",ylab="",main="")
#
#plot(alcoa2$dist[id],fit$cdfz[id],xlim=range(alcoa2$dist),ylim=c(0,1),type="l",
#     xlab="",ylab="",main="")
#par(new=TRUE)
#plot(ecdf(alcoa2$dist[id]),xlim=range(alcoa2$dist),ylim=c(0,1),verticals=TRUE,
#     do.points=FALSE,col="dark gray",ylab="",xlab="",main="")

##################################################################################
################### Boston Housing Data - a log-Slash model   ####################
###################################################################################
#data("Boston", package="MASS")
#fit <- ssym.l(log(medv) ~ crim + rm + tax + psp(lstat) + psp(dis) | psp(lstat),
#              data=Boston, family="Slash", xi=1.56, local.influence=TRUE)
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,1.0,2.3)
#
################## Plot of deviance-type residuals ##################
#plot(fit)
#
################## Plot of nonparametric effects ##################
#par(mfrow=c(1,3))
#np.graph(fit,which=1,exp=TRUE,"lstat")
#np.graph(fit,which=1,exp=TRUE,"dis")
#np.graph(fit,which=2,exp=TRUE,"lstat")
```

```
#
################## Plot of influence measures ##################
#out <- influence(fit)
#
################### Simulated envelopes ##################
#envelope(fit)

###################################################################################
####################### mcycle Data - a Power-exponential model   #################
###################################################################################
#data("mcycle", package="MASS")
#fit <- ssym.l(accel ~ ncs(times)|ncs(times), data=mcycle, family="Powerexp",xi=-0.6)
#summary(fit)
#
################## Plot of nonparametric effects ##################
#par(mfrow=c(1,2))
#np.graph(fit,which=1,obs=TRUE)
#np.graph(fit,which=2,exp=TRUE,obs=TRUE)
#
################### Simulated envelopes ##################
#envelope(fit)

###################################################################################
################### Steel Data - a log-hyperbolic model   ####################
###################################################################################
#data("Steel", package="ssym")
#fit <- ssym.l(log(life)~psp(stress), data=Steel, family="Hyperbolic", xi=1.25)
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,0.5,2)
#
################## Plot of nonparametric effects ##################
#np.graph(fit,which=1,exp=TRUE)
```

---

ssym.l2 *Fitting Censored Semi-parametric Log-symmetric Regression Models*

Description

ssym.l2 is used to fit a semi-parametric regression model suitable for analysis of data sets in which the response variable is continuous, strictly positive, asymmetric and right-censored. Under this setup, both median and skewness of the response variable distribution are explicitly modeled through semi-parametric functions, whose nonparametric components may be approximated by natural cubic splines or P-splines.

Usage

```
ssym.l2(formula, family, xi, data, epsilon, maxiter, subset, link.mu, link.phi,
         local.influence, spec, std.out)
```

Arguments

- `formula` — a symbolic description of the systematic component of the model to be fitted. See details for further information.
- `family` — a description of the (log) error distribution to be used in the model. Supported families include `Normal`, `Student`, `Contnormal`, `Powerexp`, `Hyperbolic`, `Slash`, `Sinh-normal` and `Sinh-t`, which correspond to normal, Student-t, contaminated normal, power exponential, symmetric hyperbolic, slash, sinh-normal and sinh-t distributions, respectively.
- `xi` — a numeric value or numeric vector that represents the extra parameter of the specified error distribution.
- `data` — an optional data frame, list or environment containing the variables in the model.
- `epsilon` — an optional positive value, which represents the convergence criterion. Default value is 1e-07.
- `maxiter` — an optional positive integer giving the maximal number of iterations for the estimating process. Default value is 1e03.
- `subset` — an optional expression specifying a subset of individuals to be used in the fitting process.
- `link.mu` — an optional character that specifies the link function of the median submodel.
- `link.phi` — an optional character that specifies the link function of the skewness submodel.
- `local.influence` — logical. If TRUE, local influence measures under two perturbation schemes are calculated. Default is `FALSE`.
- `spec` — character. The smoothing parameter is estimated by minimizing a overall goodness-of-fit criterion such as AIC or BIC. `spec` is an optional string to specify the goodness-of-fit measure to be used. Default value is `AIC`.
- `std.out` — logical. If FALSE, just a reduced set of attributes is returned by the model-fitting function. Default is `TRUE`.

Details

The argument *formula* comprises of three parts (separated by the symbols "~" and "|"), namely: event status and observed response variable (in log-scale) in a object of class *Surv*, predictor of the median submodel (having logarithmic link) and predictor of the skewness (or the relative dispersion) submodel (having logarithmic link). An arbitrary number of nonparametric effects may be specified in the predictors. These effects are specified to be approximated by natural cubic splines or P-splines using the functions `ncs()` or `psp()`, respectively.

The iterative estimation process is based on the Gauss-Seidel, Newton-Raphson and backfitting algorithms. The smoothing parameter is chosen by minimizing the AIC or BIC criteria.

The function `ssym.l2()` calculates overall goodness-of-fit statistics, deviance-type residuals for both submodels, as well as local influence measures under the case-weight and response perturbation schemes.

Value

theta.mu	a vector of parameter estimates associated with the median submodel.
theta.phi	a vector of parameter estimates associated with the skewness (or the relative dispersion) submodel.
vcov.mu	approximate variance-covariance matrix associated with the median submodel.
vcov.phi	approximate variance-covariance matrix associated with the skewness (or the relative dispersion) submodel.
lambdas.mu	estimate of the smoothing parameter(s) associated with the nonparametric part of the median submodel.
lambdas.phi	estimate of the smoothing parameter(s) associated with the nonparametric part of the skewness (or the relative dispersion) submodel.
gle.mu	degrees of freedom associated with the nonparametric part of the median submodel.
gle.phi	degrees of freedom associated with the nonparametric part of the skewness (or the relative dispersion) submodel.
deviance.mu	a vector with the individual contributions to the deviance associated with the median submodel.
deviance.phi	a vector with the individual contributions to the deviance associated with the skewness (or the relative dispersion) submodel.
mu.fitted	a vector with the fitted values of the (in log-scale) median submodel.
phi.fitted	a vector with the fitted values of the skewness (or the relative dispersion) submodel.
lpdf	a vector of individual contributions to the log-likelihood function.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Vanegas, L.H. and Paula, G.A. (2015) A semiparametric approach for joint modeling of median and skewness. TEST 24, 110-135.

Vanegas, L.H. and Paula, G.A. (2016) Log-symmetric distributions: statistical properties and parameter estimation. Brazilian Journal of Probability and Statistics 30, 196-220.

Vanegas, L.H. and Paula, G.A. (2016) An extension of log-symmetric regression models: R codes and applications. Journal of Statistical Computation and Simulation 86, 1709-1735.

See Also

ssym.nl, ssym.l

Examples

```
####################################################################################
################ Lung Cancer Trial - a log-Student model ##########################
####################################################################################
#data("veteran", package="survival")
#fit <- ssym.l2(Surv(log(time), status) ~ karno| karno, data = veteran,
#              family="Student", xi=4.5)
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,3,10)
#
################## Graph of deviance-type residuals ##################
#plot(fit)

####################################################################################
########## Primary biliary cirrhosis - a Power-exponential model ###################
####################################################################################
# data("pbc", package="survival")
# pbc2 <- data.frame(pbc[!is.na(pbc$edema) & !is.na(pbc$stage) & !is.na(pbc$bili),])
#
# fit <- ssym.l2(Surv(log(time),ifelse(status>=1,1,0) ) ~ factor(edema) +
#                 stage + ncs(bili), data = pbc2, family="Powerexp",
#                 xi=0.47, local.influence=TRUE)
# summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,c(0.6,3),c(0.9,5))
#
################## Graph of the nonparametric effect ##################
#np.graph(fit, which=1, exp=TRUE)
#
################## Graph of deviance-type residuals ##################
#plot(fit)

####################################################################################
################### Myeloma - a Birnbaum-Saunders model ###########################
####################################################################################
# data("myeloma", package="ssym")
#
#fit <- ssym.l2(Surv(log(t),1-event) ~ x1 + x2 + x5| -1 + x3, data=myeloma,
#       family="Sinh-normal", xi=1.8)
#summary(fit)
#
################## Graph of deviance-type residuals ##################
#plot(fit)

####################################################################################
################ Baboons Data - a log-power-exponential model   ####################
####################################################################################
########################## left-censored observations ##############################
```

```
#########################################################################################

#data("Baboons", package="ssym")
#fit <- ssym.l2(Surv(-log(t),1-cs) ~ 1, data=Baboons, family="Powerexp", xi=-0.35)
#
#summary(fit)
################## Graph of deviance-type residuals ##################
#plot(fit)
```

---

ssym.nl *Fitting Semi-parametric Log-symmetric Regression Models*

Description

ssym.nl is used to fit a semi-parametric regression model suitable for analysis of data sets in which the response variable is continuous, strictly positive, and asymmetric. Under this setup, both median and skewness of the response variable distribution are explicitly modeled, the median using a nonlinear function and the skewness through semi-parametric functions, whose nonparametric components may be approximated by natural cubic splines or P-splines.

Usage

```
ssym.nl(formula, start, family, xi, data, epsilon, maxiter, subset, link.phi,
        local.influence, spec, std.out)
```

Arguments

formula	a symbolic description of the systematic component of the model to be fitted. See details for further information.
start	a named numeric vector of starting estimates for the parameters in the specified nonlinear function.
family	a description of the (log) error distribution to be used in the model. Supported families include Normal, Student, Contnormal, Powerexp, Hyperbolic, Slash, Sinh-normal and Sinh-t, which correspond to normal, Student-t, contaminated normal, power exponential, symmetric hyperbolic, slash, sinh-normal and sinh-t distributions, respectively.
xi	a numeric value or numeric vector that represents the extra parameter of the specified error distribution.
data	an optional data frame, list or environment containing the variables in the model.
epsilon	an optional positive value, which represents the convergence criterion. Default value is 1e-07.
maxiter	an optional positive integer giving the maximal number of iterations for the estimating process. Default value is 1e03.
subset	an optional expression that specifies a subset of individuals to be used in the fitting process.

link.phi	an optional character that specifies the link function of the skewness submodel.
local.influence	logical. If TRUE, local influence measures under two perturbation schemes are calculated. Default is FALSE.
spec	character. The smoothing parameter is estimated by minimizing a overall goodness-of-fit criterion such as AIC or BIC. spec is an optional string to specify the goodness-of-fit measure to be used. Default value is AIC.
std.out	logical. If FALSE, just a reduced set of attributes is returned by the model-fitting function. Default is TRUE.

Details

The argument *formula* comprises of three parts (separated by the symbols "~" and "|"), namely: observed response variable in log-scale, predictor of the median submodel (having logarithmic link) and predictor of the skewness (or the relative dispersion) submodel (having logarithmic link). An arbitrary number of nonparametric effects may be specified in the predictor of the skewness submodel. These effects are specified to be approximated by natural cubic splines or P-splines using the functions `ncs()` or `psp()`, respectively.

The iterative estimation process is based on the Fisher scoring and backfitting algorithms. Because some distributions such as log-Student-t, log-contaminated-normal, log-power-exponential, log-slash and log-hyperbolic may be obtained as a power mixture of the log-normal distribution, the expectation-maximization (EM) algorithm is applied in those cases to obtain a more efficient iterative process of parameter estimation. Furthermore, because the Birnbaum-Saunders-t distribution can be obtained as a scale mixture of the Birnbaum-Saunders distribution, the expectation-maximization algorithm is also applied in this case to obtain a more efficient iterative process of parameter estimation. The smoothing parameter is chosen by minimizing the AIC or BIC criteria.

The function `ssym.nl()` calculates overall goodness-of-fit statistics, deviance-type residuals for both submodels, as well as local influence measures under the case-weight and response perturbation schemes.

Value

theta.mu	a vector of parameter estimates associated with the median submodel.
theta.phi	a vector of parameter estimates associated with the skewness (or the relative dispersion) submodel.
vcov.mu	approximate variance-covariance matrix associated with the median submodel.
vcov.phi	approximate variance-covariance matrix associated with the skewness (or the relative dispersion) submodel.
weights	final weights of the iterative process.
lambdas.phi	estimate of the smoothing parameter(s) associated with the nonparametric part of the skewness (or the relative dispersion) submodel.
gle.mu	degrees of freedom associated with the nonparametric part of the median submodel.
gle.phi	degrees of freedom associated with the nonparametric part of the skewness (or the relative dispersion) submodel.

`deviance.mu`	a vector with the individual contributions to the deviance associated with the median submodel.
`deviance.phi`	a vector with the individual contributions to the deviance associated with the skewness (or the relative dispersion) submodel.
`mu.fitted`	a vector with the fitted values of the (in log-scale) median submodel.
`phi.fitted`	a vector with the fitted values of the skewness (or the relative dispersion) submodel.
`lpdf`	a vector of individual contributions to the log-likelihood function.

Author(s)

Luis Hernando Vanegas <hvanegasp@gmail.com> and Gilberto A. Paula

References

Vanegas, L.H. and Paula, G.A. (2015) A semiparametric approach for joint modeling of median and skewness. TEST 24, 110-135.

Vanegas, L.H. and Paula, G.A. (2016) Log-symmetric distributions: statistical properties and parameter estimation. Brazilian Journal of Probability and Statistics 30, 196-220.

Vanegas, L.H. and Paula, G.A. (2016) An extension of log-symmetric regression models: R codes and applications. Journal of Statistical Computation and Simulation 86, 1709-1735.

See Also

ssym.l, ssym.l2

Examples

```
###############################################################################
######### Ultrasonic Calibration Data - a log-contaminated-normal model ###########
###############################################################################
#data("Chwirut1", package="NISTnls")
#fit<-ssym.nl(log(y) ~ -b1*x-log(b2 + b3*x)|x,start=c(b1=0.15,b2=0.005,b3=0.012),
#      data=Chwirut1, family='Contnormal', xi=c(0.68,0.1), local.influence=TRUE)
#summary(fit)
#
########################### Extra parameter ##########################
#extra.parameter(fit,c(0.4,0.08),c(0.9,0.11))
#
################## Graph of deviance-type residuals ##################
#plot(fit)
#
################## Simulated envelopes ##################
#envelope(fit)
#
################## Graph of local influence measures ##################
#out <- influence.ssym(fit)
```

```
###################################################################################
################## Biaxial Fatigue Data - a Birnbaum-Saunders model   #############
###################################################################################
#data("Biaxial", package="ssym")
#fit <- ssym.nl(log(Life) ~ b1*Work^b2, start=c(b1=16, b2=-0.25),
#                data=Biaxial, family='Sinh-normal', xi=1.54)
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,1.3,1.8)
#
################## Graph of deviance-type residuals ##################
#plot(fit)
#
################## Simulated envelopes ##################
#envelope(fit,reps=100,conf=0.95)
###################################################################################
################## European rabbits Data - a log-normal model   #############
###################################################################################
#data("Erabbits", package="ssym")
#fit <- ssym.nl(log(wlens) ~ b1 - b2/(b3 + age) | age, start=c(b1=5,
#               b2=130, b3=36), data=Erabbits, family='Normal')
#summary(fit)
#
################## Graph of deviance-type residuals ##################
#plot(fit)
#
################## Simulated envelopes ##################
#envelope(fit)
#
###################################################################################
################### Metsulfuron Data - a log-Student-t model ####################
###################################################################################
#data("M4", package="nlreg")
#fit <- ssym.nl(log(area) ~ log(b1+(b2-b1)/(1+(dose/b3)^b4))|ncs(dose), data=M4,
#       start = c(b1=4, b2=1400, b3=0.11, b4=1.23), family="Student", xi=6)
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,3,10)
#
################## Graph of deviance-type residuals ##################
#plot(fit)
#
################## Graph of the nonparametric effect ##################
#np.graph(fit,which=2,"dose")
#
################## Simulated envelopes ##################
#envelope(fit)
#
###################################################################################
################# Blood flow Data - a log-power-exponential model #################
###################################################################################
```

```
#data("la", package="gamlss.nl")
#fit <- ssym.nl(log(PET60) ~ log(bflow) + log(1+b1*exp(-b2/bflow)) | bflow,
#       data=la, start=c(b1=-0.6,b2=98), family="Powerexp", xi=-0.45)
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,-0.5,0)
################## Graph of deviance-type residuals ##################
#plot(fit)
#
################## Simulated envelopes ##################
#envelope(fit,reps=100,conf=0.99)
#
###################################################################################
######### Gross Domestic Product per capita Data - a Birnbaum-Saunders model ######
###################################################################################
#data("gdp", package="ssym")
#fit <- ssym.nl(log(gdp2010) ~ b1, start=c(b1=mean(log(gdp$gdp2010))), data=gdp,
#               family='Sinh-normal', xi=2.2)
#summary(fit)
########################### Extra parameter ###########################
#extra.parameter(fit,0.5,3)
################## Plot of the fitted model ##################
#id <- sort(gdp$gdp2010, index=TRUE)$ix
#par(mfrow=c(1,2))
#hist(gdp$gdp2010[id],xlim=range(gdp$gdp2010),ylim=c(0,0.00025),prob=TRUE,
#     breaks=200,col="light gray",border="dark gray",xlab="",ylab="",main="")
#par(new=TRUE)
#plot(gdp$gdp2010[id],exp(fit$lpdf[id])/gdp$gdp2010[id],xlim=range(gdp$gdp2010),
#     ylim=c(0,0.00025),type="l",xlab="",ylab="Density",main="Histogram")
#
#plot(gdp$gdp2010[id],fit$cdfz[id],xlim=range(gdp$gdp2010),ylim=c(0,1),type="l",
#     xlab="",ylab="",main="")
#par(new=TRUE)
#plot(ecdf(gdp$gdp2010[id]),xlim=range(gdp$gdp2010),ylim=c(0,1),verticals=TRUE,
#     do.points=FALSE,col="dark gray",ylab="Probability.",xlab="",main="ECDF")
###################################################################################
############# Australian Institute of Sport Data - a log-normal model #############
###################################################################################
#data("ais", package="sn")
#sex <- ifelse(ais$sex=="male",1,0)
#ais2 <- data.frame(BMI=ais$BMI,LBM=ais$LBM,sex)
#start = c(b1=7, b2=0.3, b3=2)
#fit <- ssym.nl(log(BMI) ~ log(b1 + b2*LBM + b3*sex) | sex + LBM,
#               data=ais2, start=start, family="Normal")
#summary(fit)
#
################## Graph of deviance-type residuals ##################
#plot(fit)
#
################## Simulated envelopes ##################
#envelope(fit)
#
```

```
#####################################################################################
################ Daphnia Data - a log-power-exponential model #####################
#####################################################################################
#data("daphnia", package="nlreg")
#fit <- ssym.nl(log(time) ~ log(b1+(b2-b1)/(1+(conc/b4)^b3)) | ncs(conc),
#       data=daphnia, start = c(b1=0, b2=50 , b3=2, b4=0.2), family="Powerexp",
# xi=-0.42)
#summary(fit)
#
########################### Extra parameter ###########################
#extra.parameter(fit,-0.5,-0.3)
#
################## Graph of deviance-type residuals ##################
#plot(fit)
#
################## Graph of the nonparametric effect ##################
#np.graph(fit,which=2,"conc")
#
################## Simulated envelopes ##################
#envelope(fit)
```

`Steel` *Hardened Steel*

Description

This dataset consists of the failure times for hardened steel specimens in a rolling contact fatigue test. Ten independent observations were taken at each of the four values of contact stress. The response is the length of the time until each specimen of the hardened steel failed.

Usage

```
data(Steel)
```

Format

A data frame with 40 observations on the following 2 variables.

`stress` values of contact stress, in pounds per square inch x 10^{-6}

`life` length of the time until the specimen of the hardened steel failed.

References

McCool, J. (1980) Confidence limits for Weibull regression with censored data. Transactions on Reliability, 29: 145-150.

`summary.ssym` *summary.ssym*

Description

summary.ssym displays the summary of the fitted model including parameter estimates, associated (approximated) standard errors and goodness-of-fit statistics from an object of class “"ssym".

`vcov.ssym` *vcov.ssym*

Description

vcov.ssym extracts the approximate variance-covariance matrix associated to the parameter estimates from an object of class “"ssym".

Index