Package ‘sglg’

October 11, 2021

Type Package
Title Fitting Semi-Parametric Generalized log-Gamma Regression Models
Version 0.2.0
Author Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com> and G. Paula and L. Vanegas
Maintainer Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>
Description Set of tools to fit a linear multiple or semi-parametric regression models with the possibility of non-informative random right-censoring. Under this setup, the localization parameter of the response variable distribution is modeled by using linear multiple regression or semi-parametric functions, whose non-parametric components may be approximated by natural cubic spline or P-splines. The supported distribution for the model error is a generalized log-gamma distribution which includes the generalized extreme value and standard normal distributions as important special cases. Also, some numerical and graphical devices for diagnostic of the fitted models are offered.
Depends R (>= 3.1.0)
License GPL-3
Encoding UTF-8
RoxygenNote 7.1.2
Suggests testthat
Imports Formula, survival, methods, graphics, stats, AdequacyModel, ggplot2, moments, gridExtra, pracma, progress, Rcpp, plot3D, TeachingSampling
NeedsCompilation no
Repository CRAN
Date/Publication 2021-10-11 13:20:02 UTC

R topics documented:

  bootglm ................................................................. 2
**Description**

`bootglg` is used to generate bootstrap inference, such as, estimated standard errors and approximate confidence intervals for the parameters of a generalized log-gamma distribution.

**Usage**

```r
bootglg(y, R, alpha = 0.05, method, type)
```

**Arguments**

- `y`: vector, a numeric random sample.
- `R`: integer, represents the number of replications. Default value is 1000.
- `alpha`: numeric, represents a confidence level for the bootstrap intervals. Default value is 0.05.
method character, indicates the type of bootstrap 'Nonparametric' or 'Parametric'. Default value is 'Parametric'.

type character, indicates the type of bootstrap confidence interval for the estimated parameters. The options are: 'normal', 't_student' or 'bootstrap_t'. These intervals used the bootstrap estimated standard error of the ML estimates of the parameters. Other kind of bootstrap intervals are the percentile-type intervals. We offer the option 'BCa'. It is a bias-corrected and accelerated percentile interval. The default value for the 'type' argument is 'normal'.

Value

*ml_estimates* is a vector of maximum likelihood estimates associated with the location, scale, and shape parameters.

*boot_mean_estimates* is a vector of mean of the bootstrap estimates associated with the location, scale, and shape parameters.

*boot_bias_estimates* is a vector of bootstrap estimate of bias associated with the location, scale, and shape estimators.

*boot_sd_estimates* is a vector of bootstrap standard errors of the estimates associated with the location, scale, and shape estimators.

*type* indicates the type of confidence intervals.

*intervals* array of the confidence intervals of the location, scale and shape.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

```
set.seed(1)
y <- rglg(200,location=0,scale=0.5,shape=1)
## Not run:
bootglg(y,R=500,method='Parametric',type='normal',alpha=0.02)
bootglg(y,R=500,method='Nonparametric',type='t_student')
bootglg(y,R=500,method='Parametric',type='bootstrap_t')
bootglg(y,R=500,method='Nonparametric',type='BCa')
## End(Not run)
```
Build the basis matrix and the penalty matrix of cubic B-spline basis.

Description

desBoor builds the basis matrix and penalty matrix to approximate a smooth function using cubic B-spline cubic.

Usage

desBoor2(t, knots)

Arguments

t a vector of values.
knots a set of internal knot.

Value

nknot number of knots.
knots set of knots.
N basis matrix.
K penalty matrix.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

set.seed(1)
t_1 <- runif(120)
range(t_1)
t_2 <- t_1 + 2 #runif(120,2,3)
range(t_2)
knot <- 10
dB1 <- deBoor2(t_1,knot)
dB2 <- deBoor2(t_2,knot)
dB1$knots
dB2$knots
plot(0,0,xlim=c(-0.5,3.5))
points(dB1$knots,rep(0,length(dB1$knots)),pch=20)
points(dB2$knots-delta,rep(0,length(dB2$knots)),pch=2,col='blue')
dB1$K
dB2$K
zeros <- vector()
plot(\texttt{t_1},dB1$N[,1],pch=20)
for(\texttt{j} in 1:KnotS){
  points(\texttt{t_1},dB1$N[,\texttt{j}],pch=20,col=\texttt{j})
  zeros[\texttt{j}] <- sum(dB1$N[,\texttt{j}]==0)
}
zeros/120
cond_tNN <- vector()
KnotS <- 3:50
for(\texttt{j} in KnotS){
  dB1 <- deBoor2(\texttt{t_1},\texttt{j})
  print(dB1$knots[2]-dB1$knots[1])
  min_max <- range(eigen(t(dB1$N)%*%dB1$N)$values)
  cond_tNN[j-2] <- min_max[1]/min_max[2]
}
cond_tNN
plot(KnotS,cond_tNN,pch=20,ylim=c(0,0.07))

---

**deviance_residuals**

*Deviance Residuals for a Generalized Log-gamma Regression Model*

**Description**

deviance_residuals is used to generate deviance residuals for a generalized log-gamma regression model.

**Usage**

deviance_residuals(object, ...)

**Arguments**

- **object**: an object of the class sglg. This object is returned from the call to glg(), sglg(), survglg() or ssurvglg().
- **...**: other arguments.

**Author(s)**

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

**References**

Examples

```r
# Example 1
n <- 300
error <- rglg(n,0,1,1)
y <- 0.5 + error
fit <- glg(y~1,data=as.data.frame(y))
deviance_residuals(fit)
# Example 2
n <- 300
error <- rglg(n,0,1,1)
x <- runif(n,-3,3)
y <- 0.5 + 2*x + error
fit <- glg(y~x,data=as.data.frame(y,x))
deviance_residuals(fit)
```

---

**dglg**

*Density distribution function for a generalized log-gamma variable*

**Description**

dglg is used to calculate the density distribution function of a generalized log-gamma variable at x.

**Usage**

dglg(x, location, scale, shape)

**Arguments**

- **x**: numeric, a real number.
- **location**: numeric, represent the location parameter of a generalized log-gamma distribution. Default value is 0.
- **scale**: numeric, represent the scale parameter of a generalized log-gamma distribution. Default value is 1.
- **shape**: numeric, represent the shape parameter of a generalized log-gamma distribution. Default value is 1.

**Author(s)**

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

**References**

Examples

\begin{verbatim}
x <- seq(-4,4,length=100)
dglg(x,location=0,scale=1,shape=1)
plot(x,dglg(x,location=0,scale=1,shape=1),type="l",xlab="x",ylab="Density")
\end{verbatim}

entropy

Tool to calculate the entropy for a generalized log-gamma distribution.

Description

entropy is used to obtain the entropy for a generalized log-gamma distribution.

Usage

entropy(mu, sigma, lambda)

Arguments

\begin{verbatim}
mu numeric, represent the location parameter of a generalized log-gamma distribution. Default value is 0.
sigma numeric, represent the scale parameter of a generalized log-gamma distribution. Default value is 1.
lambda numeric, represent the shape parameter of a generalized log-gamma distribution. Default value is 1.
\end{verbatim}

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

\begin{verbatim}
entropy(0,1,-1) # Extreme value type I distribution, maximum case.
entropy(0,1,1)  # Extreme value type I distribution, minimum case.
entropy(0,1,0.077) # Standard normal distribution.
\end{verbatim}
Description

Build a Normal probability plot with simulated envelope for a deviance-type residuals in semiparametric or multiple linear generalized log-gamma regression models.

Usage

envelope.sglg(fit, Rep)

Arguments

fit       an object of the class sglg. This object is returned from the call to glg(), sglg().
Rep       a positive integer. This is the number of replications on which to build the simulated envelope. Default is Rep=50.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

rows <- 120
columns <- 2
t_beta <- c(0.5, 2)
t_sigma <- 0.5
t_lambda <- 1
set.seed(8142031)
x1 <- rbinom(rows, 1, 0.5)
x2 <- runif(columns, 0, 1)
X <- cbind(x1,x2)
error <- rglg(rows, 0, 1, t_lambda)
y1 <- X %*% t_beta + t_sigma * error
data.example <- data.frame(y1,X)
fit <- glg(y1 ~ x1 + x2 - 1,data=data.example)
envelope.sglg(fit,Rep=50)
Description

glg is used to fit a multiple linear regression model suitable for analysis of data sets in which the response variable is continuous, strictly positive, and asymmetric. In this setup, the location parameter of the response variable is explicitly modeled by a linear function of the parameters.

Usage

glg(
  formula,  
data,         
  shape = 0.2, 
  Tolerance = 5e-05, 
  Maxiter = 1000, 
  format,  
  envelope = FALSE
)

Arguments

formula a symbolic description of the systematic component of the model to be fitted. See details for further information.

data an optional data frame, list containing the variables in the model.

shape an optional value for the shape parameter of the error distribution of a generalized log-gamma distribution. Default value is 0.2.

Tolerance an optional positive value, which represents the convergence criterion. Default value is 1e-04.

Maxiter an optional positive integer giving the maximal number of iterations for the estimating process. Default value is 1e03.

format an optional string value that indicates if you want a simple or a complete report of the estimating process. Default value is 'complete'.

envelope an optional and internal logical value that indicates if the glg function will be employed for build an envelope plot. Default value is 'FALSE'.

Value

mu a vector of parameter estimates associated with the location parameter.

sigma estimate of the scale parameter associated with the model.

lambda estimate of the shape parameter associated with the model.

interval estimate of a 95% confidence interval for each estimate parameters associated with the model.

Deviance the deviance associated with the model.
Examples

set.seed(21)
rows <- 120
x1 <- rbinom(rows, 1, 0.5)
x2 <- runif(rows, 0, 1)
X <- cbind(x1, x2)
t_beta <- c(0.5, 2)
t_sigma <- 1

# Extreme value case #

t_lambda <- 1 # 1 -0.9
error <- rglg(rows, 0, 1, t_lambda)
y1 <- X %*% t_beta + t_sigma * error
data.example <- data.frame(y1, X)
fit <- glg(y1 ~ x1 + x2 - 1, data=data.example) # (-0.9,6.1) and (1,5.6)
logLik(fit) # -180.598 and -183.907
summary(fit)
deviance_residuals(fit)

# Normal case: A limit case #

# When the parameter lambda goes to zero the GLG tends to a standard normal distribution.
t_lambda <- 0.001
set.seed(8142031)
error <- rglg(rows, 0, 1, t_lambda)
hist(error)
y1 <- X %*% t_beta + t_sigma * error
data.example <- data.frame(y1, X)
fit0 <- glg(y1 ~ x1 + x2 - 1, data=data.example)
logLik(fit0) # -180.598 and -183.907
fit0$AIC
fit0$mu

# A comparison with a normal linear model #
# Description

This function provides some useful statistics to assess the quality of fit of generalized log-gamma probabilistic model, including the statistics Cramer-von Mises and Anderson-Darling. It can also calculate other goodness of fit such as Hannan-Quin Information Criterion and Kolmogorov-Smirnov test.

# Usage

```r
gnfit(starts, data)
```

## Arguments

- **starts**
  - numeric vector. Initial parameters to maximize the likelihood function

- **data**
  - numeric vector. A sample of a generalized log-gamma distribution.

## Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

## References


## Examples

```r
## Not run:
set.seed(1)
# The size of the sample must be median or large to obtain a good estimates
n <- 100
sample <- rglg(n, location=0, scale=0.5, shape=0.75)
# This step takes a few minutes.
result <- gnfit(starts=c(0.1,0.75,1), data=sample)
result

## End(Not run)
```
Gu

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

**Description**

Gu builds the basis matrix and penalty matrix to approximate a smooth function using natural cubic splines based on the Gu basis form.

**Usage**

`Gu(t, knot)`

**Arguments**

- `t` the covariate.
- `knot` a integer value that represent the number of knots of the natural cubic spline.

**Value**

- `n_knot` number of knots.
- `knots` set of knots.
- `N` basis matrix.
- `K` penalty matrix.

**Author(s)**

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

**References**


**Examples**

```r
  t <- runif(1000)
  knot <- 6
  N_gu <- Gu(t, knot)
```
**Description**

influence.sglg extracts from an object of class sglg the local influence measures and displays their graphs versus the index of the observations.

**Usage**

```r
## S3 method for class 'sglg'
influence(model, ...)
```

**Arguments**

- `model`: an object of the class sglg. This object is returned from the call to glg(), sglg(), survglg() or ssurvglg().
- `...`: other arguments.

**Author(s)**

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

**References**


**Examples**

```r
rows <- 100
columns <- 2
t_beta <- c(0.5, 2)
t_sigma <- 1
t_lambda <- 1
set.seed(8142031)
x1 <- rbinom(rows, 1, 0.5)
x2 <- runif(columns, 0, 1)
X <- cbind(x1, x2)
error <- rglg(rows, 0, 1, t_lambda)
y1 <- X %*% t_beta + t_sigma * error
data.example <- data.frame(y1, X)
fit1 <- glg(y1 ~ x1 + x2 - 1, data=data.example)
influence(fit1)
```
logLik.sglg  

*Extract Log-Likelihood*

**Description**

logLik.sglg extracts log-likelihood from a model from an object of class 'sglg'.

**Usage**

```r
## S3 method for class 'sglg'
logLik(object, ...)
```

**Arguments**

- `object` an object of the class sglg. This object is returned from the call to glg(), sglg(), survglg() or ssurvglg() function.
- `...` other arguments.

lss  

*Measures of location, scale and shape measures for a generalized log-gamma distribution*

**Description**

lss is used to obtain the mean, median, mode, variance, coefficient of variation, skewness and kurtosis for a generalized log-gamma distribution.

**Usage**

```r
lss(mu, sigma, lambda)
```

**Arguments**

- `mu` numeric, represents the location parameter of a generalized log-gamma distribution. Default value is 0.
- `sigma` numeric, represents the scale parameter of a generalized log-gamma distribution. Default value is 1.
- `lambda` numeric, represents the shape parameter of a generalized log-gamma distribution. Default value is 1.

**Author(s)**

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>
References

Examples
lss(0,1,-1)  # Extreme value type I distribution, maximum case.
lss(0,1,1)   # Extreme value type I distribution, minimum case.
lss(0,1,0.01) # Standard normal distribution.

order_glg

Random Sampling of K-th Order Statistics from a Generalized Log-gamma Distribution

Description
order_glg is used to obtain a random sample of the K-th order statistics from a generalized log-gamma distribution.

Usage
order_glg(size, mu, sigma, lambda, k, n, alpha = 0.05)

Arguments
size numeric, represents the size of the sample.
mu numeric, represents the location parameter. Default value is 0.
sigma numeric, represents the scale parameter. Default value is 1.
lambda numeric, represents the shape parameter. Default value is 1.
k numeric, represents the K-th smallest value from a sample.
n numeric, represents the size of the sample to compute the order statistic from.
alpha numeric, (1 - alpha) represents the confidence of an interval for the population median of the distribution of the k-th order statistic. Default value is 0.05.

Value
A list with a random sample of order statistics from a generalized log-gamma distribution, the value of its joint probability density function evaluated in the random sample and a (1 - alpha) confidence interval for the population median of the distribution of the k-th order statistic.

Author(s)
Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>.
References


Examples

# A random sample of size 10 of order statistics from a Extreme Value Distribution.
order_glg(10, 0, 1, 1, 50)
## Not run: # A small comparison between two random sampling methods of order statistics
# Method 1
m <- 10
output <- rep(0, m)
order_sample <- function(m, n, k){
  for(i in 1:m){
    sample <- rglg(n)
    order_sample <- sort(sample)
    output[i] <- order_sample[k]
  }
  return(output)
}
N <- 10000
n <- 200
k <- 100
system.time(order_sample(N, n, k))
sample_1 <- order_sample(N, n, k)
hist(sample_1)
summary(sample_1)
# Method 2
system.time(order_glg(N, 0, 1, 1, k, n))
sample_2 <- order_glg(N, 0, 1, 1, k, n)$sample
hist(sample_2)
summary(sample_2)
## End(Not run)

pglg

Cumulative distribution function for a generalized log-gamma variable

Description

pglg is used to calculate the cumulative distribution function of a generalized log-gamma variable at x.

Usage

pglg(x, location, scale, shape)
plotnpc

Arguments

x numeric, a real number.
location numeric, represent the location parameter of a generalized log-gamma distribution. Default value is 0.
scale numeric, represent the scale parameter of a generalized log-gamma distribution. Default value is 1.
shape numeric, represent the shape parameter of a generalized log-gamma distribution. Default value is 1.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

x <- runif(3,-1,1)
pglg(sort(x),location=0.5,scale=1,shape=1)

plotnpc

Plotting a natural cubic splines or P-splines.

Description

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

Usage

plotnpc(fit, conf_lev)

Arguments

fit an object of the class sglg. This object is returned from the call to glg(), sglg(), survglg() or ssurvglg().
conf_lev is the confidence level of the asymptotic confidence band. Default value is 0.05.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>
References


Examples

set.seed(1)
n <- 300
error <- rglg(n,0,0.5,1)
t <- as.matrix((2*1:n - 1)/(2*n))
colnames(t) <- "t"
f_t <- cos(4*pi*t)
y <- 0.8 + f_t + error
colnames(y) <- "y"
data <- as.data.frame(cbind(y,1,t))
fit1 <- sglg(y ~ 1,npc=t,data=data,basis = "deBoor",alpha0=0.0001)
summary(fit1)
# The adjusted (black) non-linear component
plotnpc(fit1,conf_lev=0.02)

plotsurv.sglg

Plot simultaneously the Kaplan-Meier and parametric estimators of the survival function.

Description

plotsurv.sglg is used to plot simultaneously the Kaplan-Meier and parametric estimators of the survival function.

Usage

plotsurv.sglg(fit)

Arguments

fit an object of the class sglg. This object is returned from the call to survglg() or ssurvglg().

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


**Examples**

```r
require(survival)
rows <- 240
columns <- 2
t_beta <- c(0.5, 2)
t_sigma <- 1
t_lambda <- 1
set.seed(8142031)
x1 <- rbinom(rows, 1, 0.5)
x2 <- runif(columns, 0, 1)
X <- cbind(x1, x2)
s <- t_sigma^2
a <- 1/s
t_ini1 <- exp(X %*% t_beta) * rgamma(rows, scale = s, shape = a)
cens.time <- rweibull(rows, 0.6, 14)
delta1 <- ifelse(t_ini1 > cens.time, 1, 0)
obst1 <- t_ini1
for (i in 1:rows) {
  if (delta1[i] == 1) {
    obst1[i] <- cens.time[i]
  }
}
data.example <- data.frame(obst1, delta1, X)
fit3 <- survglg(Surv(log(obst1), delta1) ~ x1 + x2 - 1, data=data.example, shape=0.9)
plotsurv.sglg(fit3)
```

---

**qglg**  
*Quantile function for a generalized log-gamma variable*

**Description**

`qglg` is used to calculate the quantile function of a generalized log-gamma variable at x.

**Usage**

`qglg(x, location, scale, shape)`

**Arguments**

- `x` numeric, a real number between 0 and 1.
- `location` numeric, represent the location parameter of a generalized log-gamma distribution. Default value is 0.
- `scale` numeric, represent the scale parameter of a generalized log-gamma distribution. Default value is 1.
- `shape` numeric, represent the shape parameter of a generalized log-gamma distribution. Default value is 1.
quantile_residuals

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

```r
x <- runif(3,0,1)
qglg(sort(x),location=0,scale=1,shape=1)
```

quantile_residuals Quantile Residuals for a Generalized Log-gamma Regression Model

Description

quantile_residuals is used to generate quantile residuals for a generalized log-gamma regression model.

Usage

```r
quantile_residuals(fit)
```

Arguments

- `fit` is an object sglg. This object is returned from the call to glg(), sglg(), survglg() or ssurvglg().

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

```r
# Example 1
n <- 400
set.seed(4)
error <- rglg(n,0,0.5,1)
y <- as.data.frame(0.5 + error)
names(y) <- "y"
fit_0 <- glg(y~1,data=y)
```

```r
fit_0$mu
```
residuals.sglg

fit_0$sigma
fit_0$lambda
quantile_residuals(fit_0)
# Example 2
n <- 500
set.seed(6)
error <- rglg(n,0,0.5,1)
x1 <- runif(n,-2,2)
beta <- c(0.5,2)
y <- cbind(1,x1)%*%beta + error
data <- data.frame(y=y,x1=x1)
fit_1 <- glg(y~x1,data=data)
fit_1$mu
fit_1$sigma
fit_1$lambda
quantile_residuals(fit_1)

residuals.sglg  Extract Model Residuals

Description
residuals.sglg extracts the deviance-type residuals for a model from an object of class 'sglg'.

Usage
## S3 method for class 'sglg'
residuals(object, ...)

Arguments
object
... an object of the class sglg. This object is returned from the call to glg(), sglg(),
survglg() or ssurvglg() function.
other arguments.

rglg
Random number generation for a generalized log-gamma distribution

Description
rglg is used to generate random numbers for a generalized log-gamma distribution.

Usage
rglg(n, location, scale, shape)
Arguments

- **n**: numeric, size of the random sample.
- **location**: numeric, represents the location parameter of a generalized log-gamma distribution. Default value is 0.
- **scale**: numeric, represents the scale parameter of a generalized log-gamma distribution. Default value is 1.
- **shape**: numeric, represents the shape parameter of a generalized log-gamma distribution. Default value is 1.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

```r
go(10, location=-1, scale=0.5, shape=1)
```

sglg

**Fitting semi-parametric generalized log-gamma regression models**

Description

sglg 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. In this setup, the location parameter of the response variable is explicitly modeled by semi-parametric functions, whose non-parametric components may be approximated by natural cubic splines or cubic P-splines.

Usage

```r
sglg(
  formula,
  npc,
  basis,
  data,
  shape = 0.2,
  method,
  alpha0,
  Knot,
  Tolerance = 5e-05,
  Maxiter = 1000,
  format = "complete"
)
```
Arguments

formula a symbolic description of the systematic component of the model to be fitted. See details for further information.

npc a matrix with the nonparametric variables of the systematic part of the model to be fitted. Must be included the names of each variables.

basis a name of the cubic spline basis to be used in the model. Supported basis include deBoor and Gu basis which are a B-spline basis and a natural cubic spline basis, respectively.

data an optional data frame, list containing the variables in the model.

shape an optional value for the shape parameter of the error distribution of a generalized log-gamma distribution. Default value is 0.2.

method There are two possibles algorithms to estimate the parameters. The default algorithm is ‘FS’ Fisher-Scoring, the other option is ‘GSFS’ an adequate combination between the block matrix version of non-linear Gauss-Seidel algorithm and Fisher-Scoring algorithm.

alpha0 is a vector of positive values for the smoothing parameters alpha. Default vector with 1 in each entry.

Knot is a vector of the number of knots in each non-linear component of the model.

Tolerance an optional positive value, which represents the convergence criterion. Default value is 5e-05.

Maxiter an optional positive integer giving the maximal number of iterations for the estimating process. Default value is 1e03.

format an optional string value that indicates if you want a simple or a complete report of the estimating process. Default value is ‘complete’.

Value

mu a vector of parameter estimates associated with the location parameter.

sigma estimate of the scale parameter associated with the model.

lambda estimate of the shape parameter associated with the model.

interval estimate of a 95% confidence interval for each estimate parameters associated with the model.

Deviance the deviance associated with the model.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References

Examples

```r
set.seed(1)
rows<- 200
t_beta <- c(0.5,2)
t_sigma <- 0.5
t_lambda <- 1
x1 <- runif(rows,-3,3)
x2 <- rbinom(rows,1,0.5)
X <- cbind(x1,x2)
t <- as.matrix((2*1:rows - 1)/(2*rows))
colnames(t) <- "t"
f_t <- cos(4*pi*t)
error <- rglg(rows,0,1,t_lambda)
y <- X %*%t_beta + f_t + t_sigma*error
colnames(y) <- "y"
data <- data.frame(y,X,t)
fit1 <- sglg(y ~ x1 + x2 - 1,npc=t,data=data,basis = "deBoor",alpha0 = 0.1)
logLik(fit1) # -195.4538
quantile_residuals(fit1)
fit2 <- sglg(y ~ x1 + x2 - 1,npc=t,data=data,basis = "Gu",alpha0=0.5)
logLik(fit2)
```

### An example with two non-parametric components ###

```r
set.seed(2)
t_2 <- as.matrix(rnorm(rows,sd=0.5))
colnames(t_2) <- "t_2"
f_t_2 <- exp(t_2)
error <- rglg(rows,0,1,t_lambda)
y_2 <- X %*%t_beta + f_t + f_t_2 + t_sigma*error
colnames(y_2) <- "y_2"
data2 <- data.frame(y_2,X,t,t_2)
npcs <- cbind(t,t_2)
fit3 <- sglg(y_2 ~ x1 + x2 - 1, npc=npcs, data=data2, alpha0 = c(0.45,0.65))
logLik(fit3)
```

Description

Tool that supports the estimation of the shape parameter in semi-parametric or multiple linear accelerated failure time model with generalized log-gamma errors under the presence of censored data. The estimation is based on the profiled likelihood function for the shape parameter of the model.

Usage

```r
shape(formula, npc, data, interval, semi, step)
```
Arguments

- **formula**: a symbolic description of the systematic component of the model to be fitted.
- **npc**: a data frame with potential nonparametric variables of the systematic part of the model to be fitted.
- **data**: a data frame which contains the variables in the model.
- **interval**: an optional numerical vector of length 2. In this interval is the maximum likelihood estimate of the shape parameter of the model. By default is [0.1, 1.5].
- **semi**: a logical value. TRUE means that the model has a non-parametric component. By default is FALSE.
- **step**: an optional positive value. This parameter represents the length of the step of the partition of the interval parameter. By default is 0.1.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

```r
rows <- 200
columns <- 2
t_beta <- c(0.5, 2)
t_sigma <- 1
t_lambda <- 1
set.seed(8142031)
x1 <- rbinom(rows, 1, 0.5)
x2 <- runif(columns, 0, 1)
X <- cbind(x1, x2)
s <- t_sigma^2
a <- 1/s
t_ini1 <- exp(X %*% t_beta) * rgamma(rows, scale = s, shape = a)
cens.time <- rweibull(rows, 0.3, 14)
delta <- ifelse(t_ini1 > cens.time, 1, 0)
obst1 = t_ini1
for (i in 1:rows) {
  if (delta[i] == 1) {
    obst1[i] = cens.time[i]
  }
}
example <- data.frame(obst1, delta, X)
lambda <- shape(Surv(log(obst1), delta) ~ x1 + x2 - 1, data = example)
lambda
# To change interval or step or both options
lambda <- shape(Surv(log(obst1), delta) ~ x1 + x2 - 1, data = example, interval = c(0.95, 1.3), step = 0.05)
lambda
```
**smoothp**

**Description**

Tool that supports the selection of the smoothing parameters in semi-parametric generalized log-gamma models. The selection is based on the AIC, BIC, or Generalized Cross Validation methods.

**Usage**

```r
smoothp(formula, npc, data, method = "PAIC", basis, interval, step)
```

**Arguments**

- **formula**: a symbolic description of the systematic component of the model to be fitted.
- **npc**: a data frame with potential nonparametric variables of the systematic part of the model to be fitted.
- **data**: a data frame which contains the variables in the model.
- **method**: There are three possible criteria to estimate the smoothing parameters: Penalized Akaike Criterion 'PAIC', Penalized Bayesian Criterion 'PBIC' and Generalized Cross Validation ‘GCV’. The default method is 'PAIC'.
- **basis**: a name of the cubic spline basis to be used in the model. Supported basis include deBoor and Gu basis.
- **interval**: an optional numerical vector of length 2. In this interval is the maximum likelihood estimate of the shape parameter of the model. By default is [0.5,2].
- **step**: an optional positive value. This parameter represents the length of the step of the partition of the interval parameter. By default is 0.5.

**Author(s)**

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

**References**


**Examples**

```r
set.seed(1)
rows <- 150
t_beta <- c(0.5,2)
t_sigma <- 0.5
t_lambda <- 1
x1 <- runif(rows,-3,3)
x2 <- rbinom(rows,1,0.5)
```
X <- cbind(x1,x2)
t <- as.matrix((2*1:rows - 1)/(2*rows))
colnames(t) <- "t"
f_t <- cos(4*pi*t)
error <- rglg(rows,0,1,t_lambda)
y <- X %*%t_beta + f_t + t_sigma*error
colnames(y) <- "y"
data <- data.frame(y,X,t)
fit1 <- sglg(y ~ x1 + x2 - 1,npc=t,data=data,basis = "deBoor",alpha0=1)
fit1$AIC
smoothp(y ~ x1 + x2 - 1,npc=t,data=data,basis = "deBoor")
fit2 <- sglg(y ~ x1 + x2 - 1,npc=t,data=data,basis = "Gu",alpha0=0.5)
fit2$BIC
smoothp(y ~ x1 + x2 - 1,npc=t,data=data,basis = "Gu")
#################################################################
# An example with two non-parametric components                    #
#################################################################
set.seed(2)
t_2 <- as.matrix(rnorm(rows,sd=0.5))
colnames(t_2) <- 't_2'
f_t_2 <- exp(t_2)
error <- rglg(rows,0,1,t_lambda)
y_2 <- X %*%t_beta + f_t + f_t_2 + t_sigma*error
colnames(y_2) <- 'y_2'
data2 <- data.frame(y_2,X,t,t_2)
npcs <- cbind(t,t_2)
smoothp(y ~ x1 + x2 - 1,npc=npcs,data=data,method='GCV')

ssurvglg

Fitting semi-parametric generalized log-gamma regression models under the presence of right censored data.

Description

ssurvglg is used to fit a semi-parametric regression model in which the response variable is continuous, strictly positive, asymmetric and there are right censored observations. In this setup, the location parameter of the logarithm of the variable is explicitly modeled by semi-parametric functions, whose nonparametric components may be approximated by natural cubic splines or cubic P-splines.

Usage

ssurvglg(formula, npc, basis, data, shape, alpha0, Maxiter, Tolerance)

Arguments

formula a symbolic description of the systematic component of the model to be fitted. See details for further information.
npc

A data frame with potential nonparametric variables of the systematic part of the model to be fitted.

basis

A name of the cubic spline basis to be used in the model. Supported basis include deBoor and Gu basis which are a B-spline basis and a natural cubic spline basis, respectively.

data

An optional data frame, list containing the variables in the model.

shape

An optional value for the shape parameter of the model.

alpha0

Is a vector of initial values for the smoothing parameter alpha.

Maxiter

An optional positive integer giving the maximal number of iterations for the estimating process. Default value is 1e03.

Tolerance

An optional positive value, which represents the convergence criterion. Default value is 1e-04.

Value

mu a vector of parameter estimates associated with the location parameter.

sigma estimate of the scale parameter associated with the model.

lambda estimate of the shape parameter associated with the model.

interval estimate of a 95% confidence interval for each estimate parameters associated with the model.

Deviance the deviance associated with the model.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

```r
require(survival)
rows <- 150
columns <- 2
t_beta <- c(0.5, 2)
t_sigma <- 0.5
t_lambda <- 1
set.seed(8142030)
x1 <- rbinom(rows, 1, 0.5)
x2 <- runif(rows, 0, 1)
X <- cbind(x1, x2)
t_knot1 <- 6
t_knot <- seq(0, 1, length = t_knot1)
t_g1 <- 0.4 * sin(pi * t_knot)
BasisN <- function(n, knot) {
  # Basis function
  # End of example code
```
N <- matrix(0, n, knot)
m <- n/knot
block <- rep(1, m)
for (i in 1:knot) {
  l <- (i - 1) * m + 1
  r <- i * m
  N[l:r, i] <- block
}
return(N)
}s_N1 <- BasisN(rows, length(ts1))
x3 <- s_N1 %*% ts1
colnames(x3) <- 'x3'
sys <- X %*% t_beta + s_N1 %*% t_g1
t_ini1 <- exp(sys) * rweibull(rows, 1/t_sigma, 1)
cens.time <- rweibull(rows, 1.5, 14)
delta <- ifelse(t_ini1 > cens.time, 1, 0)
obst1 <- t_ini1
for(i in 1:rows) {
  if (delta[i] == 1) {
    obst1[i] = cens.time[i]
  }
}
d_example <- data.frame(obst1, delta, X, x3)
fit4 <- ssurvglg(Surv(log(obst1), delta) ~ x1 + x2 - 1, npc = x3, data = d_example, shape = 0.9)
summary(fit4)

---

**Description**

`summary.sglg` extracts and displays the summary of the fitted model including parameter estimates, associated (approximated) standard errors and goodness-of-fit statistics from a model from an object of class `sglg`.

**Usage**

```r
## S3 method for class 'sglg'
summary(object, ...)
```

**Arguments**

- **object**: an object of the class `sglg`. This object is returned from the call to `glg()`, `sglg()`, `survglg()` or `ssurvglg()` function.
- **...**: other arguments.
survglg is used to fit a multiple linear regression model in which the response variable is continuous, strictly positive, asymmetric and there are right censored observations. In this setup, the location parameter of the logarithm of the response variable is modeled by a linear model of the parameters.

Usage

\texttt{survglg(formula, data, shape, Maxiter, Tolerance)}

Arguments

- \texttt{formula}: a symbolic description of the systematic component of the model to be fitted. See details for further information.
- \texttt{data}: an optional data frame, list containing the variables in the model.
- \texttt{shape}: an optional value for the shape parameter of the model.
- \texttt{Maxiter}: an optional positive integer giving the maximal number of iterations for the estimating process. Default value is 1e03.
- \texttt{Tolerance}: an optional positive value, which represents the convergence criterion. Default value is 1e-04.

Value

- \texttt{mu}: a vector of parameter estimates associated with the location parameter.
- \texttt{sigma}: estimate of the scale parameter associated with the model.
- \texttt{lambda}: estimate of the shape parameter associated with the model.
- \texttt{interval}: estimate of a 95\% confidence interval for each estimate parameters associated with the model.
- \texttt{Deviance}: the deviance associated with the model.

Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References

Examples

```r
require(survival)
rows <- 240
columns <- 2
t_beta <- c(0.5, 2)
t_sigma <- 1
set.seed(8142031)
x1 <- rbinom(rows, 1, 0.5)
x2 <- runif(columns, 0, 1)
X <- cbind(x1, x2)
s <- t_sigma^2
a <- 1/s
t_ini1 <- exp(X %*% t_beta) * rgamma(rows, scale = s, shape = a)
cens.time <- rweibull(rows, 0.3, 14)
delta1 <- ifelse(t_ini1 > cens.time, 1, 0)
obst1 <- t_ini1
for (i in 1:rows) {
  if (delta1[i] == 1) {
    obst1[i] <- cens.time[i]
  }
}
data.example <- data.frame(obst1, delta1, X)
fit3 <- survglg(Surv(log(obst1), delta1) ~ x1 + x2 - 1, data=data.example, shape=0.9)
logLik(fit3)
summary(fit3)
```

---

**survival_gg**

*Survival, Hazard and Cumulative Hazard functions for a Generalized Gamma Distribution*

**Description**

`survival_gg` is used to obtain the value of survival, hazard and cumulative hazard functions of a generalized gamma distribution at a positive value.

**Usage**

`survival_gg(x, mu, sigma, lambda)`

**Arguments**

- `x`: numeric, represent a positive value. Default value is 1.
- `mu`: numeric, represents the location parameter of a generalized gamma distribution. Default value is 0.
- `sigma`: numeric, represents the scale parameter of a generalized gamma distribution. Default value is 1.
- `lambda`: numeric, represents the shape parameter of a generalized gamma distribution. Default value is 1.
Author(s)

Carlos Alberto Cardozo Delgado <cardozorpackages@gmail.com>

References


Examples

```
survival_gg(0.0001,0,1,-1) # Extreme value type I distribution, maximum case.
times <- seq(0.05,7,by=0.05)
plot(times, survival_gg(times,0,1,-1)$survival_value,type='l')
plot(times, survival_gg(times,0,1,-1)$hazard_value,type='l')
plot(times, survival_gg(times,0,1,-1)$cumulative_hazard_value,type='l')
```
Index

bootglg, 2

deBoor2, 4
deviance_residuals, 5
dglg, 6

tenropy, 7
envelope.sglg, 8

glg, 9
gnfit, 11
Gu, 12

influence.sglg, 13

logLik.sglg, 14
lss, 14

order_glg, 15

pglg, 16
plotnpc, 17
plotsurv.sglg, 18

qglg, 19
quantile_residuals, 20

residuals.sglg, 21
rglg, 21

sglg, 22
shape, 24
smooothp, 26
ssurvglg, 27
summary.sglg, 29
survglg, 30
survival_gg, 31