Package ‘EstimationTools’

October 24, 2019

Type Package

Title Maximum Likelihood Estimation for Probability Functions from Data Sets

Version 1.2.1

Depends R (>= 3.3.0), DEoptim, boot, numDeriv, BBmisc

Imports Rdpack, utils, stats

RdMacros Rdpack

Suggests gamlss.dist, survival, knitr, rmarkdown

VignetteBuilder knitr, utils

Description A routine for parameter estimation for any probability density or mass function implemented in R via maximum likelihood (ML) given a data set. This routine is a wrapper function specifically developed for ML estimation. There are included optimization procedures such as 'nlminb' and 'optim' from base package, and 'DEoptim' Mullen (2011) <doi: 10.18637/jss.v040.i06>. Standard errors are estimated with 'numDeriv' Gilbert (2011) <http://CRAN.R-project.org/package=numDeriv> or the option 'Hessian = TRUE' of 'optim' function.

License GPL-3


Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Jaime Mosquera [aut, cre] (<https://orcid.org/0000-0002-1684-4756>), Freddy Hernandez [aut] (<https://orcid.org/0000-0001-7459-3329>)

Maintainer Jaime Mosquera <jmosquerag@una1.edu.co>

Repository CRAN

Date/Publication 2019-10-24 15:10:02 UTC
R topics documented:

- Fibers ................................................................. 2
- logit_link ............................................................ 2
- log_link .............................................................. 3
- maxlogL ............................................................. 4
- NegInv_link ......................................................... 7
- summary.maxlogL ............................................... 8

Description

Tensile strengths (in GPa) of 69 specimens of carbon fiber tested under tension at gauge lengths of 20 mm.

Usage

Fibers

Format

A data frame with 69 observations.

logit_link

Logit link function (for estimation with maxlogL)

Description

log_link object provides a way to implement logit link function that maxlogL needs to perform estimation. See documentation for maxlogL for further information on parameter estimation and implementation of link objects.

Usage

logit_link()

Details

logit_link is part of a family of generic functions with no input arguments that defines and returns a list with details of the link function:

1. name: a character string with the name of the link function.
2. g: implementation of the link function as a generic function in R.
3. g_inv: implementation of the inverse link function as a generic function in R.

There is a way to add new mapping functions. The user must specify the details aforesaid.
log_link

Value

A list with logit link function, its inverse and its name.

See Also

maxlogL

Other link functions: NegInv_link, log_link

Examples

# Estimation of proportion in binomial distribution with 'logit' function
# 10 trials, probability of success equals to 30%
N <- rbinom(n = 100, size = 10, prob = 0.3)
phat <- maxlogL(x = N, dist = 'dbinom', fixed = list(size=10),
    link = list(over = "prob", fun = "logit_link"))
summary(phat)

# Link function name
fun <- logit_link()$name
print(fun)

# Link function
g <- logit_link()$g
curve(g(x), from = 0, to = 1)

# Inverse link function
ginv <- logit_link()$g_inv
curve(ginv(x), from = -10, to = 10)
2. \( g \): implementation of the link function as a generic function in \( R \).
3. \( g_{\text{inv}} \): implementation of the inverse link function as a generic function in \( R \).

There is a way to add new mapping functions. The user must specify the details aforesaid.

**Value**

A list with logit link function, its inverse and its name.

**See Also**

maxlogL

Other link functions: NegInv_link, logit_link

**Examples**

```r
# One parameters of normal distribution mapped with logarithmic function
x <- rnorm(n = 10000, mean = 50, sd = 4)
theta_2 <- maxlogL( x = x, link = list(over = "sd",
          fun = "log_link") )
summary(theta_2)

# Link function name
fun <- log_link()$name
print(fun)

# Link function
g <- log_link()$g
curve(g(x), from = 0, to = 1)

# Inverse link function
ginv <- log_link()$g_inv
curve(ginv(x), from = -5, to = 5)
```

---

**maxlogL**

*Maximum Likelihood Estimation for parametric distributions*

**Description**

Function to compute maximum likelihood estimators (MLE) of any distribution implemented in \( R \).

**Usage**

```r
maxlogL(x, dist = "dnorm", fixed = NULL, link = NULL, start = NULL,
         lower = NULL, upper = NULL, optimizer = "nlminb", control = NULL,
         ...)```
**Arguments**

- **x**
  - A vector with data to be fitted. This argument must be a matrix with hierarchical distributions.

- **dist**
  - A length-one character vector with the name of density/mass function of interest. The default value is "dnorm", to compute maximum likelihood estimators of normal distribution.

- **fixed**
  - A list with fixed/known parameters of distribution of interest. Fixed parameters must be passed with its name.

- **link**
  - A list with names of parameters to be linked, and names of the link function object. For names of parameters, please visit documentation of density/mass function. There are three link functions available: `log_link`, `logit_link` and `NegInv_link`.

- **start**
  - A numeric vector with initial values for the parameters to be estimated.

- **lower**
  - A numeric vector with lower bounds, with the same length of argument 'start' (for box-constrained optimization).

- **upper**
  - A numeric vector with upper bounds, with the same length of argument 'start' (for box-constrained optimization).

- **optimizer**
  - A length-one character vector with the name of optimization routine. `nlminb`, `optim` and `DEoptim` are available; `nlminb` is the default routine.

- **control**
  - Control parameters of the optimization routine. Please, visit documentation of selected optimizer for further information.

- **...**
  - Further arguments to be supplied to the optimizer.

**Details**

`maxlogL` calculates computationally the likelihood function corresponding to the distribution specified in argument `dist` and maximizes it through `optim`, `nlminb` or `DEoptim`. `maxlogL` generates an S3 object of class `maxlogL`.

**Value**

A list with class "maxlogL" containing the following lists:

- **fit**
  - A list with output information about estimation and method used.

- **inputs**
  - A list with all input arguments.

- **outputs**
  - A list with some output additional information:
    - Number of parameters.
    - Sample size
    - Standard error computation method.

**Note**

The following generic functions can be used with a `maxlogL` object: `summary`
maxlogL

Author(s)
Jaime Mosquera Gutiérrez, <jmosquerag@unal.edu.co>

References


See Also
summary.maxlogL, optim, nlminb, DEoptim, DEoptim.control

Examples
# Estimation with one fixed parameter
x <- rnorm(n = 10000, mean = 160, sd = 6)
theta_1 <- maxlogL(x = x, dist = 'dnorm', control = list(trace = 1),
                   link = list(over = "sd", fun = "log_link"),
                   fixed = list(mean = 160))
summary(theta_1)

# Both parameters of normal distribution mapped with logarithmic function
theta_2 <- maxlogL(x = x, dist = "dnorm",
                   link = list(over = c("mean","sd"),
                              fun = c("log_link","log_link")))
summary(theta_2)

# Parameter estimation in ZIP distribution
library(gamlss.dist)
z <- rZIP(n=10000, mu=6, sigma=0.08)
theta_3 <- maxlogL(x = z, dist='dZIP', start = c(0, 0), lower = c(-Inf, -Inf),
                   upper = c(Inf, Inf), optimizer = 'optim',
                   link = list(over=c("mu", "sigma"),
                               fun = c("log_link", "logit_link")))
summary(theta_3)
Negative inverse link function (for estimation with maxlogL)

Description
NegInv_link object provides a way to implement negative inverse link function that maxlogL needs to perform estimation. See documentation for maxlogL for further information on parameter estimation and implementation of link objects.

Usage
NegInv_link()

Details
NegInv_link is part of a family of generic functions with no input arguments that defines and returns a list with details of the link function:

1. name: a character string with the name of the link function.
2. g: implementation of the link function as a generic function in R.
3. g_inv: implementation of the inverse link function as a generic function in R.

There is a way to add new mapping functions. The user must specify the details aforesaid.

Value
A list with negative inverse link function, its inverse and its name.

See Also
maxlogL

Other link functions: log_link, logit_link

Examples

# Estimation of rate parameter in exponential distribution
T <- rexp(n = 1000, rate = 3)
lambda <- maxlogL(x = T, dist = "dexp", start = 5,
                 link = list(over = "rate", fun = "NegInv_link"))
summary(lambda)

# Link function name
fun <- NegInv_link()$name
print(fun)
# Link function
g <- NegInv_link()$g
curve(g(x), from = 0.1, to = 1)

# Inverse link function
ginv <- NegInv_link()$g_inv
curve(ginv(x), from = 0.1, to = 1)

summary.maxlogL

## S3 method for class 'maxlogL'
summary(object, Boot_Std_Err = FALSE, ...)

Arguments

object an object class "maxlogL".

Boot_Std_Err a logical variable for standard Errors computation by bootstrapping. The default is FALSE. This computation occurs when hessian from optim and hessian fails in maxlogL routine. If this argument is TRUE, standard errors are computed, even if hessian did not fail in maxlogL routine.

... arguments passed to boot for estimation of standard error with non-parametric bootstrap.

Details

This summary method takes standard errors from maxlogL object and displays them. If hessian and Hessian from optim fails, standard errors are computed with bootstrap. However, if user sets Boot_Std_Err = TRUE in this summary function, standard errors are calculated by bootstrap, even if hessian or Hessian from optim converges.

Supose that the user creates a variable named fit that stores maxlogL object. The summary method modifies the element fit$outputs$StdE_Method object from Gobal Environment (see the Second Example). If user does not creat a variable, the summary methid it simply calculates standard errors (see the 'Third Example').

Value

An object of class 'summary.maxlogL'.
References


See Also

\texttt{maxlogL}, \texttt{boot}

Examples

```r
#--------------------------------------------------------------------------------
### First example: One known parameter

x <- rnorm(n = 10000, mean = 160, sd = 6)
theta_1 <- maxlogL(x = x, dist = 'dnorm', control = list(trace = 1),
                   link = list(over = "sd", fun = "log_link"),
                   fixed = list(mean = 160))
summary(theta_1)

#--------------------------------------------------------------------------------
# Second example: Binomial probability parameter estimation with variable
# creation

N <- rbinom(n = 100, size = 10, prob = 0.3)
phat <- maxlogL(x = N, dist = 'dbinom', fixed = list(size = 10),
                 link = list(over = "prob", fun = "logit_link"))

## Standard error calculation method
print(phat$outputs$StdE_Method)

## Standard error value (not computed yet, because is computed with 'summary')
print(phat$outputs$StdE)

## 'summary' method
summary(phat)

## Now, standard error is updated
print(phat$outputs$StdE_Method)
print(phat$outputs$StdE)

#--------------------------------------------------------------------------------
# Third example: Binomial probability parameter estimation with no variable
# creation

N <- rbinom(n = 100, size = 10, prob = 0.3)
summary(maxlogL(x = N, dist = 'dbinom', fixed = list(size = 10),
                link = list(over = "prob", fun = "logit_link")))
```

```r
summary.maxlogL
```
Index

*Topic datasets
   Fibers, 2

boot, 8, 9
DEoptim, 5, 6
DEoptim.control, 6

Fibers, 2
hessian, 8
log_link, 3, 3, 5, 7
logit_link, 2, 4, 5, 7
maxlogL, 2–4, 4, 7–9
NegInv_link, 3–5, 7
nlminb, 5, 6

optim, 5, 6, 8
summary.maxlogL, 6, 8