Package ‘fitdistrplus’

June 2, 2010

Title Help to fit of a parametric distribution to non-censored or censored data

Version 0.1-3

Date 2010-05-28

Author Marie Laure Delignette-Muller <`ml.delignette@vet-lyon.fr`>, Regis Pouillot <`rpouillot@yahoo.fr`>, Jean-Baptiste Denis <`jbdenis@jouy.inra.fr`> and Christophe Dutang <`christophe.dutang@ensimag.fr`>

Maintainer Marie Laure Delignette-Muller <`ml.delignette@vet-lyon.fr`>

Depends R (>= 2.10.0)

Description Extends the fitdistr function (of the MASS package) with several functions to help the fit of a parametric distribution to non-censored or censored data. Censored data may contain left censored, right censored and interval censored values, with several lower and upper bounds.

License GPL (>= 2)

URL `http://riskassessment.r-forge.r-project.org`

Repository CRAN

Repository/R-Forge/Project riskassessment

Repository/R-Forge/Revision 100

Date/Publication 2010-06-02 09:09:23

R topics documented:

`bootdist` *Bootstrap simulation of uncertainty for non-censored data*

Description

Uses parametric or nonparametric bootstrap resampling in order to simulate uncertainty in the parameters of the distribution fitted to non-censored data.

Usage

```
bootdist(f, bootmethod="param", niter=1001)
## S3 method for class 'bootdist':
print(x,...)
## S3 method for class 'bootdist':
plot(x,...)
## S3 method for class 'bootdist':
summary(object,...)
```

Arguments

`f`	An object of class 'fitdist' result of the function `fitdist`.
`bootmethod`	A character string coding for the type of resampling : `"param"` for a parametric resampling and `"nonparam"` for a nonparametric resampling of data.
`niter`	The number of samples drawn by bootstrap.
`x`	an object of class 'bootdist'.
`object`	an object of class 'bootdist'.
`...`	further arguments to be passed to generic methods

Details

Samples are drawn by parametric bootstrap (resampling from the distribution fitted by `fitdist`) or non parametric bootstrap (resampling with replacement from the data set). On each bootstrap sample the function `mledist` or the function `mmedist` (according to the component `f$method` of the object of class 'fitdist') is used to estimate bootstrapped values of parameters. When `mledist` fails to converge, `NA` values are returned. Medians and 2.5 and 97.5 percentiles are computed by removing `NA` values. The number of iterations for which `mledist` converges is printed in the summary with the medians and the 95 percent confidence intervals of parameters (2.5 and 97.5 percentiles).

The plot of an object of class 'bootdist' consists in a scatterplot or a matrix of scatterplots of the bootstrapped values of parameters. It uses the function `stripchart` when the fitted distribution is characterized by only one parameter, and the function `plot` in other cases. In these last cases, it provides a representation of the joint uncertainty distribution of the fitted parameters.

Value

`bootdist` returns an object of class 'bootdist', a list with 4 components,

`estim`	a data frame containing the boostrapped values of parameters.
`converg`	a vector containing the codes for convergence obtained when using `mledist` on each bootstraped data set or `NULL` if `mmedist` is used.
`method`	A character string coding for the type of resampling : `"param"` for a parametric resampling and `"nonparam"` for a nonparametric resampling of data.
`CI`	bootstrap medians and 95 percent confidence percentile intervals of parameters.

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>`

References

Cullen AC and Frey HC (1999) Probabilistic techniques in exposure assessment. Plenum Press, USA, pp. 181-241.

See Also

`fitdist`, `mledist` and `mmedist`.

Examples

```
x1<-c(6.4,13.3,4.1,1.3,14.1,10.6,9.9,9.6,15.3,22.1,13.4,
13.2,8.4,6.3,8.9,5.2,10.9,14.4)

f1<-fitdist(x1,"norm",method="mle")
b1<-bootdist(f1)
print(b1)
plot(b1)
summary(b1)

b1b<-bootdist(f1,bootmethod="nonparam")
summary(b1b)

f1b<-fitdist(x1,"gamma",method="mle")
b1b<-bootdist(f1b)
plot(b1b)

x2<-c(rep(4,1),rep(2,3),rep(1,7),rep(0,12))
f2<-fitdist(x2,"pois",method="mme")
b2<-bootdist(f2)
plot(b2,pch=16)
summary(b2)
```

`bootdistcens` *Bootstrap simulation of uncertainty for censored data*

Description

Uses nonparametric bootstrap resampling in order to simulate uncertainty in the parameters of the distribution fitted to censored data.

Usage

```
bootdistcens(f, niter=1001)
## S3 method for class 'bootdistcens':
print(x,...)
## S3 method for class 'bootdistcens':
plot(x,...)
## S3 method for class 'bootdistcens':
summary(object,...)
```

Arguments

`f`	An object of class 'fitdistcens' result of the function `fitdistcens`.
`niter`	The number of samples drawn by bootstrap.
`x`	an object of class 'bootdistcens'.
`object`	an object of class 'bootdistcens'.
`...`	further arguments to be passed to generic methods

Details

Samples are drawn by non parametric bootstrap (resampling with replacement from the data set). On each bootstrap sample the function `mledist` is used to estimate bootstrapped values of parameters. When `mledist` fails to converge, `NA` values are returned. Medians and 2.5 and 97.5 percentiles are computed by removing `NA` values. The number of iterations for which `mledist` converges is printed in the summary with the medians and the 95 percent confidence intervals of parameters (2.5 and 97.5 percentiles).

The plot of an object of class 'bootdistcens' consists in a scatterplot or a matrix of scatterplots of the bootstrapped values of parameters. It uses the function `stripchart` when the fitted distribution is characterized by only one parameter, and the function `plot` in other cases. In these last cases, it provides a representation of the joint uncertainty distribution of the fitted parameters.

Value

`bootdistcens` returns an object of class 'bootdistcens', a list with 3 components,

`estim`	a data frame containing the boostrapped values of parameters.
`converg`	a vector containing the codes for convergence obtained when using `mledist` on each bootstraped data set or `NULL` if `mmedist` is used.
`CI`	bootstrap medians and 95 percent confidence percentile intervals of parameters.

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>`

References

Cullen AC and Frey HC (1999) Probabilistic techniques in exposure assessment. Plenum Press, USA, pp. 181-241.

See Also

`fitdistcens` and `mledist`.

Examples

```
d1<-data.frame(
left=c(1.73,1.51,0.77,1.96,1.96,-1.4,-1.4,NA,-0.11,0.55,
    0.41,2.56,NA,-0.53,0.63,-1.4,-1.4,-1.4,NA,0.13),
right=c(1.73,1.51,0.77,1.96,1.96,0,-0.7,-1.4,-0.11,0.55,
    0.41,2.56,-1.4,-0.53,0.63,0,-0.7,NA,-1.4,0.13))
f1<-fitdistcens(d1, "norm")
b1<-bootdistcens(f1)
b1
summary(b1)
plot(b1)
```

`descdist` *Description of an empirical distribution for non-censored data*

Description

Computes descriptive parameters of an empirical distribution for non-censored data and provides a skewness-kurtosis plot.

Usage

```
descdist(data,discrete=FALSE,boot=NULL,method="unbiased",graph=TRUE,obs.col="red",b
```

Arguments

`data`	A numeric vector.
`discrete`	If `TRUE`, the distribution is considered as discrete.
`boot`	If not `NULL`, `boot` values of skewness and kurtosis are plotted from bootstrap samples of `data`. `boot` must be fixed in this case to an integer above 10.
`method`	"unbiased" for unbiased estimated values of statistics or "sample" for sample values.
`graph`	If `FALSE`, the skewness-kurtosis graph is not plotted.
`obs.col`	Color used for the observed point on the skewness-kurtosis graph.
`boot.col`	Color used for bootstrap sample of points on the skewness-kurtosis graph.

Details

Minimum, maximum, median, mean, sample sd, and sample (if `method=="sample"`) or by default unbiased estimations of skewness and Pearsons's kurtosis values (Fisher, 1930) are printed. Be careful, estimations of skewness and kurtosis are unbiased only for normal distributions and estimated values are thus only indicative. A skewness-kurtosis plot such as the one proposed by Cullen and Frey (1999) is given for the empirical distribution. On this plot, values for common distributions are also displayed as a tools to help the choice of distributions to fit to data. For some distributions (normal, uniform, logistic, exponential for example), there is only one possible value for the skewness and the kurtosis (for a normal distribution for example, skewness = 0 and kurtosis = 3), and the distribution is thus represented by a point on the plot. For other distributions, areas of possible values are represented, consisting in lines (gamma and lognormal distributions for example), or larger areas (beta distribution for example). The Weibull distribution is not represented on the graph but it is indicated on the legend that shapes close to lognormal and gamma distributions may be obtained with this distribution.

In order to take into account the uncertainty of the estimated values of kurtosis and skewness from data, the data set may be boostraped by fixing the argument `boot` to an integer above 10. `boot` values of skewness and kurtosis corresponding to the `boot` bootstrap samples are then computed and reported in blue color on the skewness-kurtosis plot.

If `discrete` is `TRUE`, the represented distributions are the Poisson, negative binomial and normal distributions. If `discrete` is `FALSE`, these are uniform, normal, logistic, lognormal, beta and gamma distributions.

Value

`descdist` returns a list with 7 components,

`min`	the minimum value
`max`	the maximum value
`median`	the median value
`mean`	the mean value
`sd`	the standard deviation sample or estimated value
`skewness`	the skewness sample or estimated value
`kurtosis`	the kurtosis sample or estimated value

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>`

References

Cullen AC and Frey HC (1999) Probabilistic techniques in exposure assessment. Plenum Press, USA, pp. 81-159. Evans M, Hastings N and Peacock B (2000) Statistical distributions. John Wiley and Sons Inc.

Fisher RA (1930) The moments of the distribution for normal samples of measures of departures from normality. Proc. R. Soc. London, Series A 130, 16-28.

See Also

`plotdist`

Examples

```
x1 <- rnorm(100)
descdist(x1)
descdist(x1,boot=1000)

descdist(rbeta(100,shape1=0.05,shape2=1),boot=1000)
descdist(rgamma(100,shape=2,rate=1),boot=1000)

descdist(rpois(100,lambda=2),discrete=TRUE,boot=1000)

data(groundbeef)
serving <- groundbeef$serving
descdist(serving, boot=1000)
```

`fitdist` *Fitting of univariate distributions to non-censored data*

Description

Fits a univariate distribution to non-censored data by maximum likelihood or matching moments.

Usage

```
fitdist(data, distr, method=c("mle", "mme"), start,...)
## S3 method for class 'fitdist':
print(x,...)
## S3 method for class 'fitdist':
plot(x,breaks="default",...)
## S3 method for class 'fitdist':
summary(object,...)
```

Arguments

`data`	A numeric vector.
`distr`	A character string `"name"` naming a distribution for which the corresponding density function `dname`, the corresponding distribution function `pname` and the corresponding quantile function `qname` must be defined, or directly the density function.
`method`	A character string coding for the fitting method: `"mle"` for 'maximum likelihood estimation and `"mme"` for 'matching moment estimation'.
`start`	A named list giving the initial values of parameters of the named distribution. This argument will not be taken into account if `method="mme"`, and may be omitted for some distributions for which reasonable starting values are computed if `method="mle"` (see details).

`x`	an object of class 'fitdist'.
`object`	an object of class 'fitdist'.
`breaks`	If `"default"` the histogram is plotted with the function `hist` with its default breaks definition. Else `breaks` is passed to the function `hist`. This argument is not taken into account with discrete distributions: `"binom"`, `"nbinom"`, `"geom"`, `"hyper"` and `"pois"`.
`...`	further arguments to be passed to generic functions, or to the function `"mledist"` if 'maximum likelihood' is the chosen method, in order to control the optimization method.

Details

When `method="mle"`, maximum likelihood estimations of the distribution parameters are computed using the function `mledist`. By default direct optimization of the log-likelihood is performed using `optim`, with the "Nelder-Mead" method for distributions characterized by more than one parameter and the "BFGS" method for distributions characterized by only one parameter. The method used in `optim` may be chosen or another optimization method may be chosen using ... argument (see `mledist` for details). For the following named distributions, reasonable starting values will be computed if `start` is omitted : `"norm"`, `"lnorm"`, `"exp"` and `"pois"`, `"cauchy"`, `"gamma"`, `"logis"`, `"nbinom"` (parametrized by mu and size), `"geom"`, `"beta"` and `"weibull"`. Note that these starting values may not be good enough if the fit is poor. The function is not able to fit a uniform distribution. With the parameter estimates, the function returns the log-likelihood and the standard errors of the estimates calculated from the Hessian at the solution found by `optim` or by the user-supplied function passed to mledist.

When `method="mme"`, the estimated values of the distribution parameters are provided only for the following distributions : `"norm"`, `"lnorm"`, `"pois"`, `"exp"`, `"gamma"`, `"nbinom"`, `"geom"`, `"beta"`, `"unif"` and `"logis"`. For distributions characterized by one parameter (`"geom"`, `"pois"` and `"exp"`), this parameter is simply estimated by matching theoretical and observed means, and for distributions characterized by two parameters, these parameters are estimated by matching theoretical and observed means and variances (Vose, 2000).

The plot of an object of class "fitdist" returned by `fitdist` uses the function `plotdist`.

Value

`fitdist` returns an object of class 'fitdist', a list with following components,

`estimate`	the parameter estimates
`method`	the character string coding for the fitting method : `"mle"` for 'maximum likelihood estimation' and `"mme"` for 'matching moment estimation'
`sd`	the estimated standard errors or `NULL` if `method="mme"`
`cor`	the estimated correlation matrix or `NULL` if `method="mme"`
`loglik`	the log-likelihood or `NULL` if `method="mme"`
`aic`	the Akaike information criterion or `NULL` if `method="mme"`
`bic`	the the so-called BIC or SBC (Schwarz Bayesian criterion) or `NULL` if `method="mme"`
`n`	the length of the data set
`data`	the dataset
`distname`	the name of the distribution

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>` and Christophe Dutang

References

Cullen AC and Frey HC (1999) Probabilistic techniques in exposure assessment. Plenum Press, USA, pp. 81-155.

Venables WN and Ripley BD (2002) Modern applied statistics with S. Springer, New York, pp. 435-446.

Vose D (2000) Risk analysis, a quantitative guide. John Wiley & Sons Ltd, Chischester, England, pp. 99-143.

See Also

`plotdist`, `optim`, `mledist`, `mmedist` , `gofstat` and `fitdistcens`.

Examples

```
# (1) basic fit of a normal distribution with maximum likelihood estimation
#

x1 <- c(6.4,13.3,4.1,1.3,14.1,10.6,9.9,9.6,15.3,22.1,13.4,
13.2,8.4,6.3,8.9,5.2,10.9,14.4)
f1 <- fitdist(x1,"norm")
print(f1)
plot(f1)
summary(f1)

# (2) use the moment matching estimation
#

f1b <- fitdist(x1,"norm",method="mme")
summary(f1b)

# (3) MME for log normal distribution
#

f1c <- fitdist(x1,"lnorm",method="mme")
summary(f1c)

# (4) defining your own distribution functions, here for the Gumbel distribution
# for other distributions, see the CRAN task view dedicated to probability distributions

dgumbel <- function(x,a,b) 1/b*exp((a-x)/b)*exp(-exp((a-x)/b))
pgumbel <- function(q,a,b) exp(-exp((a-q)/b))
qgumbel <- function(p,a,b) a-b*log(-log(p))

f1c <- fitdist(x1,"gumbel",start=list(a=10,b=5))
print(f1c)
plot(f1c)
```

```
# (5) fit a discrete distribution (Poisson)
#

x2<-c(rep(4,1),rep(2,3),rep(1,7),rep(0,12))
f2<-fitdist(x2,"pois")
plot(f2)
summary(f2)

# (6) how to change the optimisation method?
#

fitdist(x1,"gamma",optim.method="Nelder-Mead")
fitdist(x1,"gamma",optim.method="BFGS")
fitdist(x1,"gamma",optim.method="L-BFGS-B",lower=c(0,0))
fitdist(x1,"gamma",optim.method="SANN")

# (7) custom optimisation function
#

#create the sample
mysample <- rexp(100, 5)
mystart <- 8

res1 <- fitdist(mysample, dexp, start= mystart, optim.method="Nelder-Mead")

#show the result
summary(res1)

#the warning tell us to use optimise, because the Nelder-Mead is not adequate.

#to meet the standard 'fn' argument and specific name arguments, we wrap optimize,
myoptimize <- function(fn, par, ...)
{
    res <- optimize(f=fn, ..., maximum=FALSE)  #assume the optimization function minimize

    standardres <- c(res, convergence=0, value=res$objective, par=res$minimum, hessian=NA)

    return(standardres)
}

#call fitdist with a 'custom' optimization function
res2 <- fitdist(mysample, dexp, start=mystart, custom.optim=myoptimize, interval=c(0, 100))

#show the result
summary(res2)


# (8) custom optimisation function - another example with the genetic algorithm
#
## Not run:
    #set a sample
    x1 <- c(6.4, 13.3, 4.1, 1.3, 14.1, 10.6, 9.9, 9.6, 15.3, 22.1, 13.4, 13.2, 8.4, 6.3, 8.9
```

```
    fit1 <- fitdist(x1, "gamma")
    summary(fit1)

    #wrap genoud function rgenoud package
    mygenoud <- function(fn, par, ...)
    {
        require(rgenoud)
        res <- genoud(fn, starting.values=par, ...)
        standardres <- c(res, convergence=0)

        return(standardres)
    }

    #call fitdist with a 'custom' optimization function
    fit2 <- fitdist(x1, "gamma", custom.optim=mygenoud, nvars=2,
        Domains=cbind(c(0,0), c(10, 10)), boundary.enforcement=1,
        print.level=1, hessian=TRUE)

    summary(fit2)

## End(Not run)
```

`fitdistcens` *Fitting of univariate distributions to censored data*

Description

Fits a univariate distribution to censored data by maximum likelihood.

Usage

```
fitdistcens(censdata, distr, start,...)
## S3 method for class 'fitdistcens':
print(x,...)
## S3 method for class 'fitdistcens':
plot(x,...)
## S3 method for class 'fitdistcens':
summary(object,...)
```

Arguments

`censdata`	A dataframe of two columns respectively named `left` and `right`, describing each observed value as an interval. The `left` column contains either `NA` for left censored observations, the left bound of the interval for interval censored observations, or the observed value for non-censored observations. The `right` column contains either `NA` for right censored observations, the right bound of the interval for interval censored observations, or the observed value for non-censored observations.

`distr`	A character string `"name"` naming a distribution, for which the corresponding density function `dname` and the corresponding distribution function `pname` must be defined, or directly the density function.
`start`	A named list giving the initial values of parameters of the named distribution. This argument may be omitted for some distributions for which reasonable starting values are computed (see details).
`x`	an object of class 'fitdistcens'.
`object`	an object of class 'fitdistcens'.
`...`	further arguments to be passed to generic functions, or to the function `"mledist"` in order to control the optimization method.

Details

Maximum likelihood estimations of the distribution parameters are computed using the function `mledist`. By default direct optimization of the log-likelihood is performed using `optim`, with the "Nelder-Mead" method for distributions characterized by more than one parameter and the "BFGS" method for distributions characterized by only one parameter. The method used in `optim` may be chosen or another optimization method may be chosen using ... argument (see `mledist` for details). For the following named distributions, reasonable starting values will be computed if `start` is omitted : `"norm"`, `"lnorm"`, `"exp"` and `"pois"`, `"cauchy"`, `"gamma"`, `"logis"`, `"nbinom"` (parametrized by mu and size), `"geom"`, `"beta"` and `"weibull"`. Note that these starting values may not be good enough if the fit is poor. The function is not able to fit a uniform distribution. With the parameter estimates, the function returns the log-likelihood and the standard errors of the estimates calculated from the Hessian at the solution found by `optim` or by the user-supplied function passed to mledist.

The plot of an object of class "fitdistcens" returned by `fitdistcens` uses the function `plotdistcens`.

Value

`fitdistcens` returns an object of class 'fitdistcens', a list with following components,

`estimate`	the parameter estimates
`sd`	the estimated standard errors
`cor`	the estimated correlation matrix
`loglik`	the log-likelihood
`aic`	the Akaike information criterion
`bic`	the the so-called BIC or SBC (Schwarz Bayesian criterion)
`censdata`	the censored dataset
`distname`	the name of the distribution

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>`

References

Venables WN and Ripley BD (2002) Modern applied statistics with S. Springer, New York, pp. 435-446.

See Also

`plotdistcens`, `optim`, `mledist` and `fitdist`.

Examples

```
# (1) basic fit of a normal distribution on censored data
#

d1<-data.frame(
left=c(1.73,1.51,0.77,1.96,1.96,-1.4,-1.4,NA,-0.11,0.55,0.41,
    2.56,NA,-0.53,0.63,-1.4,-1.4,-1.4,NA,0.13),
right=c(1.73,1.51,0.77,1.96,1.96,0,-0.7,-1.4,-0.11,0.55,0.41,
    2.56,-1.4,-0.53,0.63,0,-0.7,NA,-1.4,0.13))
f1n<-fitdistcens(d1, "norm")
f1n
summary(f1n)
plot(f1n,rightNA=3)

# (2) defining your own distribution functions, here for the Gumbel distribution
# for other distributions, see the CRAN task view dedicated to probability distributions

dgumbel <- function(x,a,b) 1/b*exp((a-x)/b)*exp(-exp((a-x)/b))
pgumbel <- function(q,a,b) exp(-exp((a-q)/b))
qgumbel <- function(p,a,b) a-b*log(-log(p))
f1g<-fitdistcens(d1,"gumbel",start=list(a=0,b=2))
summary(f1g)
plot(f1g,rightNA=3)

# (3) comparison of fits of various distributions
#

d3<-data.frame(left=10^(d1$left),right=10^(d1$right))
f3w<-fitdistcens(d3,"weibull")
summary(f3w)
plot(f3w,leftNA=0)
f3l<-fitdistcens(d3,"lnorm")
summary(f3l)
plot(f3l,leftNA=0)
f3e<-fitdistcens(d3,"exp")
summary(f3e)
plot(f3e,leftNA=0)

# (4) how to change the optimisation method?
#

fitdistcens(d3,"gamma",optim.method="Nelder-Mead")
fitdistcens(d3,"gamma",optim.method="BFGS")
```

```
fitdistcens(d3,"gamma",optim.method="SANN")
fitdistcens(d3,"gamma",optim.method="L-BFGS-B",lower=c(0,0))

# (5) custom optimisation function - example with the genetic algorithm
#
## Not run:

    #wrap genoud function rgenoud package
    mygenoud <- function(fn, par, ...)
    {
        require(rgenoud)
        res <- genoud(fn, starting.values=par, ...)
        standardres <- c(res, convergence=0)

        return(standardres)
    }

    # call fitdistcens with a 'custom' optimization function
    fit.with.genoud<-fitdistcens(d3, "gamma", custom.optim=mygenoud, nvars=2,
        Domains=cbind(c(0,0), c(10, 10)), boundary.enforcement=1,
        print.level=1, hessian=TRUE)

    summary(fit.with.genoud)

## End(Not run)
```

`gofstat` *Goodness-of-fit statistics*

Description

Computes goodness-of-fit statistics for a fit of a parametric distribution on non-censored data.

Usage

```
gofstat(f, chisqbreaks, meancount, print.test = FALSE)
```

Arguments

`f`	An object of class 'fitdist' result of the function `fitdist`.
`chisqbreaks`	A numeric vector defining the breaks of the cells used to compute the chi-squared statistic. If omitted, these breaks are automatically computed from the data in order to reach roughly the same number of observations per cell, roughly equal to the argument `meancount`, or sligthly more if there are some ties.
`meancount`	The mean number of observations per cell expected for the definition of the breaks of the cells used to compute the chi-squared statistic. This argument will not be taken into account if the breaks are directly defined in the argument `chisqbreaks`. If `chisqbreaks` and `meancount` are both omitted,

	`meancount` is fixed in order to obtain roughly $(4n)^{2/5}$ cells with n the length of the dataset.
`print.test`	If `FALSE`, the results of the tests are computed but not printed

Details

Goodness-of-fit statistics are computed. The Chi-squared statistic is computed using cells defined by the argument `chisqbreaks` or cells automatically defined from the data in order to reach roughly the same number of observations per cell, roughly equal to the argument `meancount`, or sligthly more if there are some ties. If `chisqbreaks` and `meancount` are both omitted, `meancount` is fixed in order to obtain roughly $(4n)^{2/5}$ cells, with n the length of the dataset (Vose, 2000). The Chi-squared statistic is not computed if the program fails to define enough cells due to a too small dataset. When the Chi-squared statistic is computed, and if the degree of freedom (nb of cells - nb of parameters - 1) of the corresponding distribution is strictly positive, the p-value of the Chi-squared test is returned.

For the distributions assumed continuous (all but `"binom"`, `"nbinom"`, `"geom"`, `"hyper"` and `"pois"` for R base distributions), Kolmogorov-Smirnov and Anderson-Darling statistics are also computed, as defined by Cullen and Frey (1999).

An approximate Kolmogorov-Smirnov test is performed by assuming the distribution parameters known. The critical value defined by Stephens (1986) for a completely specified distribution is used to reject or not the distribution at the significance level 0.05. Because of this approximation, the result of the test (decision of rejection of the distribution or not) is returned only for datasets with more than 30 observations. Note that this approximate test may be too conservative.

For datasets with more than 5 observations and for distributions for which the test is described by Stephens (1986) (`"norm"`, `"lnorm"`, `"exp"`, `"cauchy"`, `"gamma"`, `"logis"` and `"weibull"`), the Anderson-darling test is performed as described by Stephens (1986). This test takes into account the fact that the parameters are not known but estimated from the data. The result is the decision to reject or not the distribution at the significance level 0.05.

Only recommended statistics are automatically printed, i.e. Anderson-Darling and Kolmogorov statistics for continuous distributions and Chi-squared statistics for discrete ones (`"binom"`, `"nbinom"`, `"geom"`, `"hyper"` and `"pois"`). Results of the tests are printed only if `print.test=TRUE`. Even not printed, all the available results may be found in the list returned by the function.

Value

`gof` returns a list with following components,

`chisq`	the Chi-squared statistic or `NULL` if not computed
`chisqbreaks`	breaks used to define cells in the Chi-squared statistic
`chisqpvalue`	p-value of the Chi-squared statistic or `NULL` if not computed
`chisqdf`	degree of freedom of the Chi-squared distribution or `NULL` if not computed
`chisqtable`	a table with observed and theoretical counts used for the Chi-squared calculations
`ad`	the Anderson-Darling statistic or `NULL` if not computed

`adtest`	the decision of the Anderson-Darling test or `NULL` if not computed
`ks`	the Kolmogorov-Smirnov statistic or `NULL` if not computed
`kstest`	the decision of the Kolmogorov-Smirnov test or `NULL` if not computed

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>` and Christophe Dutang

References

Cullen AC and Frey HC (1999) Probabilistic techniques in exposure assessment. Plenum Press, USA, pp. 81-155.

Stephens MA (1986) Tests based on edf statistics. In Goodness-of-fit techniques (D'Agostino RB and Stephens MA, eds), Marcel dekker, New York, pp. 97-194.

Venables WN and Ripley BD (2002) Modern applied statistics with S. Springer, New York, pp. 435-446.

Vose D (2000) Risk analysis, a quantitative guide. John Wiley & Sons Ltd, Chischester, England, pp. 99-143.

See Also

`fitdist`.

Examples

```
# (1) for a fit of a normal distribution
#

x1 <- c(6.4,13.3,4.1,1.3,14.1,10.6,9.9,9.6,15.3,22.1,13.4,
13.2,8.4,6.3,8.9,5.2,10.9,14.4)
print(f1 <- fitdist(x1,"norm"))
gofstat(f1)
gofstat(f1,print.test=TRUE)

# (2) fit a discrete distribution (Poisson)
#

x2<-c(rep(4,1),rep(2,3),rep(1,7),rep(0,12))
print(f2<-fitdist(x2,"pois"))
g2 <- gofstat(f2,chisqbreaks=c(0,1),print.test=TRUE)
g2$chisqtable


# (3) comparison of fits of various distributions
#

x3<-rweibull(n=100,shape=2,scale=1)
gofstat(f3a<-fitdist(x3,"weibull"))
gofstat(f3b<-fitdist(x3,"gamma"))
```

```
gofstat(f3c<-fitdist(x3,"exp"))

# (4) Use of Chi-squared results in addition to
#     recommended statistics for continuous distributions
#

x4<-rweibull(n=100,shape=2,scale=1)
f4<-fitdist(x4,"weibull")
g4 <-gofstat(f4,meancount=10)
print(g4)
```

`groundbeef` *Ground beef serving size data sets*

Description

Serving sizes collected in a French survey, for ground beef patties consumed by children under 5 years old.

Usage

```
data(groundbeef)
```

Format

`groundbeef` is a data frame with 1 column (serving: serving sizes in grams)

Source

Delignette-Muller, M.L., Cornu, M. 2008. Quantitative risk assessment for Escherichia coli O157:H7 in frozen ground beef patties consumed by young children in French households. *International Journal of Food Microbiology*, **128**, 158-164.

Examples

```
data(groundbeef)
descdist(groundbeef$serving)
plotdist(groundbeef$serving)
```

`mledist` *Maximum likelihood fitting of univariate distributions*

Description

Fits a univariate distribution by maximum likelihood.

Usage

```
mledist(data, distr, start, optim.method="default", lower=-Inf, upper=Inf, custom.
```

Arguments

`data`	A numeric vector for non censored data or a dataframe of two columns respectively named `left` and `right`, describing each observed value as an interval for censored data. In that case the `left` column contains either `NA` for left censored observations, the left bound of the interval for interval censored observations, or the observed value for non-censored observations. The `right` column contains either `NA` for right censored observations, the right bound of the interval for interval censored observations, or the observed value for non-censored observations.
`distr`	A character string `"name"` naming a distribution (or directly the density function) for which the corresponding density function `dname` and the corresponding distribution `pname` must be classically defined.
`start`	A named list giving the initial values of parameters of the named distribution. This argument may be omitted for some distributions for which reasonable starting values are computed (see details).
`optim.method`	`"default"` (see details) or optimization method to pass to `optim`.
`lower`	Left bounds on the parameters for the `"L-BFGS-B"` method (see `optim`).
`upper`	Right bounds on the parameters for the `"L-BFGS-B"` method (see `optim`).
`custom.optim`	a function carrying the MLE optimisation (see details).
`...`	further arguments passed to the `optim` or `custom.optim` function.

Details

When `custom.optim=NULL` (the default), maximum likelihood estimations of the distribution parameters are computed with the R base `optim`. Direct optimization of the log-likelihood is performed (using `optim`) by default with the `"Nelder-Mead"` method for distributions characterized by more than one parameter and the `"BFGS"` method for distributions characterized by only one parameter, or with the method specified in the argument `"optim.method"` if not `"default"`. Box-constrainted optimization may be used with the method `"L-BFGS-B"`, using the constraints on parameters specified in arguments `lower` and `upper`. If non-trivial bounds are supplied, this method will be automatically selected, with a warning.

For the following named distributions, reasonable starting values will be computed if `start` is omitted : `"norm"`, `"lnorm"`, `"exp"` and `"pois"`, `"cauchy"`, `"gamma"`, `"logis"`, `"nbinom"` (parametrized by mu and size), `"geom"`, `"beta"` and `"weibull"`. Note that these starting values may not be good enough if the fit is poor. The function is not able to fit a uniform distribution.

If `custom.optim` is not `NULL`, then the user-supplied function is used instead of the R base `optim`. The `custom.optim` must have (at least) the following arguments `fn` for the function to be optimized, `par` for the initialized parameters. Internally the function to be optimized will also have other arguments, such as `obs` with observations and `ddistname` with distribution name for non censored data (Beware of potential conflicts with optional arguments of `custom.optim`). It is assumed that `custom.optim` should carry out a MINIMIZATION. Finally, it should return at least the following components `par` for the estimate, `convergence` for the convergence code, `value` for `fn(par)` and `hessian`. See examples in `fitdist` and `fitdistcens`.

This function is not intended to be called directly but is internally called in `fitdist` and `bootdist` when used with the maximum likelihood method and `fitdistcens` and `bootdistcens`.

Value

`mledist` returns a list with following components,

`estimate`	the parameter estimates
`convergence`	an integer code for the convergence of `optim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`loglik`	the log-likelihood
`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function. It is used in `fitdist` to estimate standard errors.
`optim.function`	the name of the optimization function used for maximum likelihood

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>` and Christophe Dutang

References

Venables W.N. and Ripley B.D. (2002) Modern applied statistics with S. Springer, New York, pp. 435-446.

See Also

`mmedist`, `fitdist`,`fitdistcens`, `optim`, `bootdistcens` and `bootdist`.

Examples

```
# (1) basic fit of a normal distribution with maximum likelihood estimation
#

x1<-c(6.4,13.3,4.1,1.3,14.1,10.6,9.9,9.6,15.3,22.1,13.4,
13.2,8.4,6.3,8.9,5.2,10.9,14.4)
mledist(x1,"norm")

# (2) defining your own distribution functions, here for the Gumbel distribution
# for other distributions, see the CRAN task view dedicated to probability distributions

dgumbel<-function(x,a,b) 1/b*exp((a-x)/b)*exp(-exp((a-x)/b))
mledist(x1,"gumbel",start=list(a=10,b=5))

# (3) fit a discrete distribution (Poisson)
#

x2<-c(rep(4,1),rep(2,3),rep(1,7),rep(0,12))
mledist(x2,"pois")
mledist(x2,"nbinom")

# (4) fit a finite-support distribution (beta)
#

x3<-c(0.80,0.72,0.88,0.84,0.38,0.64,0.69,0.48,0.73,0.58,0.81,
0.83,0.71,0.75,0.59)
mledist(x3,"beta")


# (5) fit frequency distributions on USArrests dataset.
#

x4 <- USArrests$Assault
mledist(x4, "pois")
mledist(x4, "nbinom")

# (6) fit a continuous distribution (Gumbel) to censored data.
#

d1<-data.frame(
left=c(1.73,1.51,0.77,1.96,1.96,-1.4,-1.4,NA,-0.11,0.55,0.41,
    2.56,NA,-0.53,0.63,-1.4,-1.4,-1.4,NA,0.13),
right=c(1.73,1.51,0.77,1.96,1.96,0,-0.7,-1.4,-0.11,0.55,0.41,
    2.56,-1.4,-0.53,0.63,0,-0.7,NA,-1.4,0.13))
mledist(d1,"norm")

dgumbel<-function(x,a,b) 1/b*exp((a-x)/b)*exp(-exp((a-x)/b))
pgumbel<-function(q,a,b) exp(-exp((a-q)/b))
mledist(d1,"gumbel",start=list(a=0,b=2),optim.method="Nelder-Mead")
```

`mmedist` *Fitting of univariate distributions by matching moments*

Description

Fits a univariate distribution by matching moments.

Usage

```
mmedist(data, distr)
```

Arguments

`data`	A numeric vector.
`distr`	A character string `"name"` naming a distribution or directly the density function dname. The estimated values of the distribution parameters are provided only for the following distributions : `"norm"`, `"lnorm"`, `"exp"` and `"pois"`, `"gamma"`, `"logis"`, `"nbinom"` , `"geom"`, `"beta"` and `"unif"`.

Details

For distributions characterized by one parameter (`"geom"`, `"pois"` and `"exp"`), this parameter is simply estimated by matching theoretical and observed means, and for distributions characterized by two parameters, these parameters are estimated by matching theoretical and observed means and variances (Vose, 2000).

This function is not intended to be called directly but is internally called in `fitdist` and `bootdist` when used with the matching moments method.

Value

`mmedist` returns the named parameter or a named vector of parameters.

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>`

References

Vose D (2000) Risk analysis, a quantitative guide. John Wiley & Sons Ltd, Chischester, England, pp. 99-143. Evans M, Hastings N and Peacock B (2000) Statistical distributions. John Wiley and Sons Inc.

See Also

`mledist`, `fitdist` and `bootdist`.

Examples

```
x1<-c(6.4,13.3,4.1,1.3,14.1,10.6,9.9,9.6,15.3,22.1,13.4,
13.2,8.4,6.3,8.9,5.2,10.9,14.4)
mmedist(x1,"norm")

x2<-c(rep(4,1),rep(2,3),rep(1,7),rep(0,12))
mmedist(x2,"pois")

x3<-c(0.80,0.72,0.88,0.84,0.38,0.64,0.69,0.48,0.73,0.58,0.81,
0.83,0.71,0.75,0.59)
mmedist(x3,"beta")
```

`plotdist` *Plot of empirical and theoretical distributions for non-censored data*

Description

Plots an empirical distribution (non-censored data) with a theoretical one if specified.

Usage

```
plotdist(data,distr,para,breaks="default",discrete=FALSE,...)
```

Arguments

`data`	A numeric vector.
`distr`	A character string `"name"` naming a distribution for which the corresponding density function `dname`, the corresponding distribution function `pname` and the corresponding quantile function `qname` must be defined, or directly the density function. This argument may be omitted only if `para` is omitted.
`para`	A named list giving the parameters of the named distribution. This argument may be omitted only if `distname` is omitted.
`breaks`	If `"default"` the histogram is plotted with the function `hist` with its default breaks definition. Else `breaks` is passed to the function `hist`. This argument is not taken into account if `discrete` is `TRUE`.
`discrete`	If TRUE, the distribution is considered as discrete. This argument is not taken into account if `distname` is defined. In this last case, the distribution is automatically assumed discrete if and only if `distname` is `"binom"`, `"nbinom"`,`"geom"`, `"hyper"` or `"pois"`.
`...`	further graphical arguments passed to graphical functions used in plotdist

Details

Empirical and, if specified, theoretical distributions are plotted in density and in cdf. For continuous distributions, the function `hist` is used with its default breaks definition if `breaks` is `"default"` or passing `breaks` as an argument if it differs from `"default"`. For continuous distribution and when a theoretical distribution is specified by both arguments `distname` and `para`, Q-Q plot (plot of the quantiles of the theoretical fitted distribution (x-axis) against the empirical quantiles of the data) and P-P plot (i.e. for each value of the data set, plot of the cumulative density function of the fitted distribution (x-axis) against the empirical cumulative density function (y-axis)) are also given (Cullen and Frey, 1999). The function `ppoints` is used for the Q-Q plot, to generate the set of probabilities at which to evaluate the inverse distribution.

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>`

References

Cullen AC and Frey HC (1999) Probabilistic techniques in exposure assessment. Plenum Press, USA, pp. 81-159.

See Also

`descdist`, `hist`, `plot`, `plotdistcens` and `ppoints`.

Examples

```
x1<-c(6.4,13.3,4.1,1.3,14.1,10.6,9.9,9.6,15.3,22.1,13.4,
13.2,8.4,6.3,8.9,5.2,10.9,14.4)
plotdist(x1)
plotdist(x1,col="red",type="b",pch=4)
plotdist(x1,type="s")

x2<-c(rep(4,1),rep(2,3),rep(1,7),rep(0,12))
plotdist(x2,discrete=TRUE)
plotdist(x2,"pois",para=list(lambda=mean(x2)))
plotdist(x2,"pois",para=list(lambda=mean(x2)),col="red",lwd="2")

xn<-rnorm(n=100,mean=10,sd=5)
plotdist(xn,"norm",para=list(mean=mean(xn),sd=sd(xn)))
plotdist(xn,"norm",para=list(mean=mean(xn),sd=sd(xn)),pch=16,col="green")

data(groundbeef)
plotdist(groundbeef$serving)
```

`plotdistcens` *Plot of empirical and theoretical distributions for censored data*

Description

Plots an empirical distribution for censored data with a theoretical one if specified.

Usage

```
plotdistcens(censdata,distr,para,leftNA=-Inf,rightNA=Inf,...)
```

Arguments

`censdata`	A dataframe of two columns respectively named `left` and `right`, describing each observed value as an interval. The `left` column contains either `NA` for left censored observations, the left bound of the interval for interval censored observations, or the observed value for non-censored observations. The `right` column contains either `NA` for right censored observations, the right bound of the interval for interval censored observations, or the observed value for non-censored observations.
`distr`	A character string `"name"` naming a distribution, for which the corresponding density function `dname` and the corresponding distribution function `pname` must be defined, or directly the density function.
`para`	A named list giving the parameters of the named distribution. This argument may be omitted only if `distname` is omitted.
`leftNA`	the real value of the left bound of left censored observations : `-Inf` or a finite value such as `0` for positive data for example.
`rightNA`	the real value of the right bound of right censored observations : `Inf` or a finite value such as a realistic maximum value.
`...`	further graphical arguments passed to other methods

Details

Empirical and, if specified, theoretical distributions are plotted in cdf. Data are reported directly as segments for interval, left and right censored data, and as points for non-censored data. Before plotting, observations are ordered and a rank r is associated to each of them. Left censored observations are ordered first, by their right bounds. Interval censored and non censored observations are then ordered by their mid-points and, at last, right censored observations are ordered by their left bounds. If `leftNA` (resp. `rightNA`) is finite, left censored (resp. right censored) observations are considered as interval censored observations and ordered by mid-points with non-censored and interval censored data. It is sometimes necessary to fix `rightNA` or `leftNA` to a realistic extreme value, even if not exactly known, to obtain a reasonable global ranking of observations.

After ranking, each of the n observations is plotted as a point (one x-value) or a segment (an interval of possible x-values), with an y-value equal to r/n, r being the rank of each observation in the global ordering previously described.

Author(s)

Marie-Laure Delignette-Muller `<ml.delignette@vet-lyon.fr>`

See Also

`plotdist`.

Examples

```
d1<-data.frame(
left=c(1.73,1.51,0.77,1.96,1.96,-1.4,-1.4,NA,-0.11,0.55,
    0.41,2.56,NA,-0.53,0.63,-1.4,-1.4,-1.4,NA,0.13),
right=c(1.73,1.51,0.77,1.96,1.96,0,-0.7,-1.4,-0.11,0.55,
    0.41,2.56,-1.4,-0.53,0.63,0,-0.7,NA,-1.4,0.13))
plotdistcens(d1)
plotdistcens(d1,rightNA=3)
plotdistcens(d1,"norm",para=list(mean=0.12,sd=1.4),rightNA=3)

d3<-data.frame(left=10^(d1$left),right=10^(d1$right))
plotdistcens(d3,leftNA=0)
plotdistcens(d3,"lnorm",para=list(meanlog=0.27,sdlog=3.3),leftNA=0)
```

Index