Use of the package `fitdistrplus` to specify a distribution from non-censored or censored data

Marie Laure Delignette-Muller, Régis Pouillot , Jean-Baptiste Denis and Christophe Dutang

April 23, 2010

Here you will find some easy examples of use of the functions of the package `fitdistrplus`. The aim is to show you by examples how to use these functions to help you to specify a parametric distribution from data corresponding to a random sample drawn from a theoretical distribution that you want to describe. For details, see the documentation of each function, using the R help command (ex.: `?fitdist`). Do not forget to load the package using the function `library` or `require` before testing following examples.

```
> library(fitdistrplus)
```

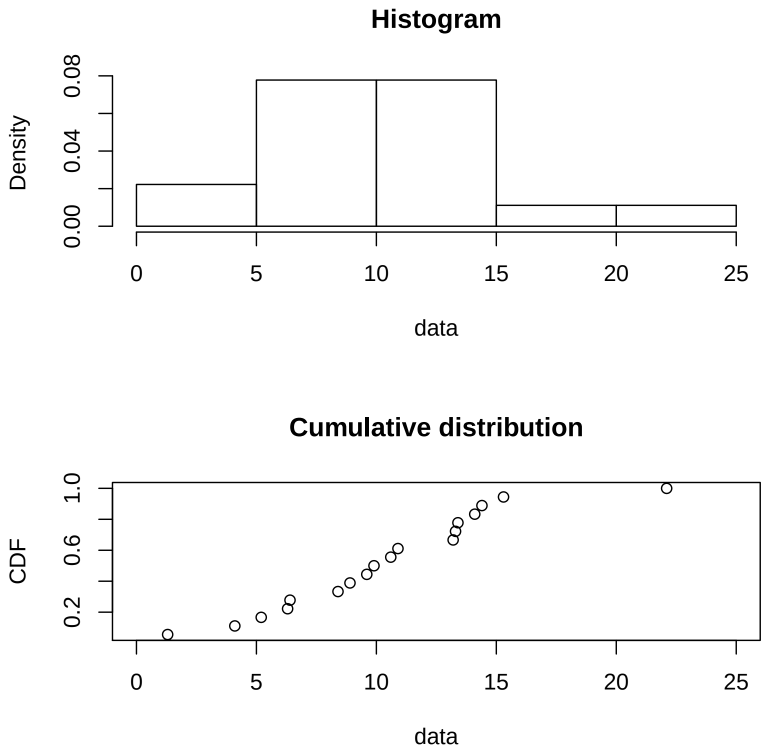
Contents

1 Specification of a distribution from non-censored continuous data

1.1 Graphical display of the observed distribution

First of all, the observed distribution may be plotted using the function `plotdist`.

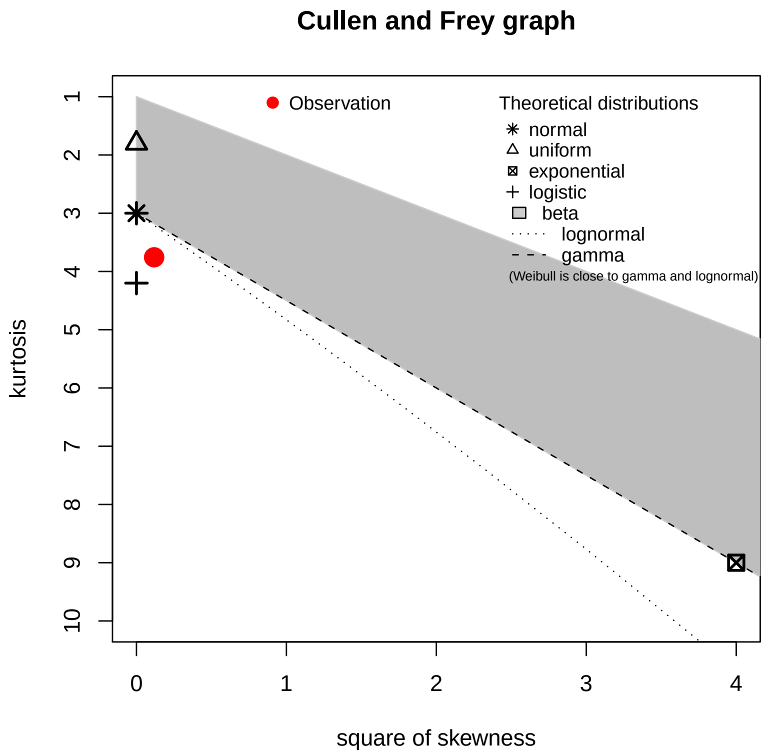
```
> x1 <- c(6.4, 13.3, 4.1, 1.3, 14.1, 10.6, 9.9, 9.6, 15.3, 22.1,
+     13.4, 13.2, 8.4, 6.3, 8.9, 5.2, 10.9, 14.4)
> plotdist(x1)
```



1.2 Characterization of the observed distribution

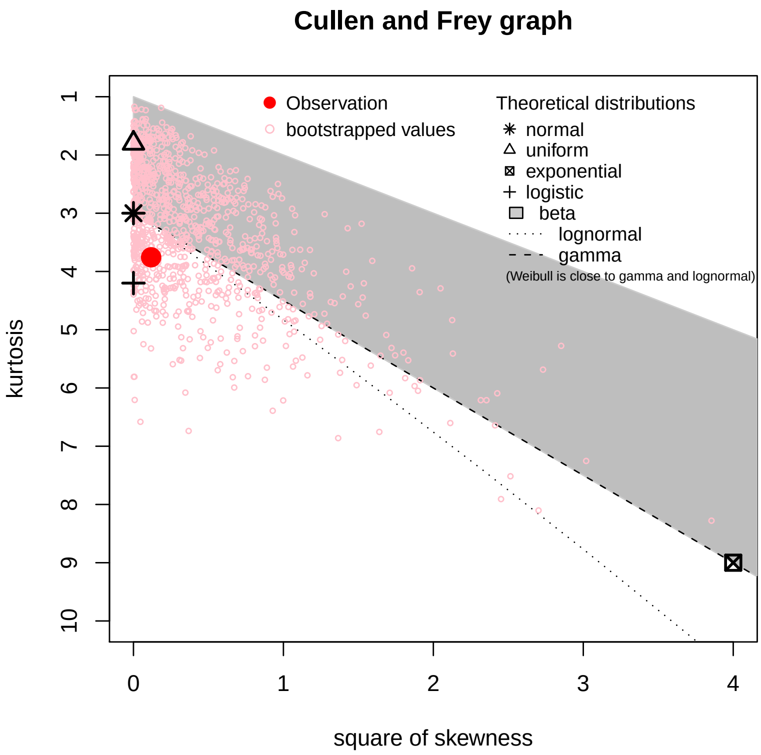
Descriptive parameters of the empirical distribution may be computed using the function `descdist`. This function will also provide by default a skewness-kurtosis plot which may help you to select which distribution(s) to fit among the potential candidates.

```
> descdist(x1)

summary statistics
------
min:  1.3   max:  22.1
median:  10.2
mean:  10.4
estimated sd:  4.88
estimated skewness:  0.343
estimated kurtosis:  3.76
```



Skewness and kurtosis are known not to be robust. In order to try to take into account the uncertainty on the estimated values of kurtosis and skewness, the data set may be boostrapped by fixing the argument `boot` to an integer above 10 in `descdist`. `boot` values of skewness and kurtosis corresponding to the boot nonparametric bootstrap samples are then computed and reported on the skewness-kurtosis plot.

```
> descdist(x1, boot = 1000)

summary statistics
------
min:  1.3   max:  22.1
median:  10.2
mean:  10.4
estimated sd:  4.88
estimated skewness:  0.343
estimated kurtosis:  3.76
```

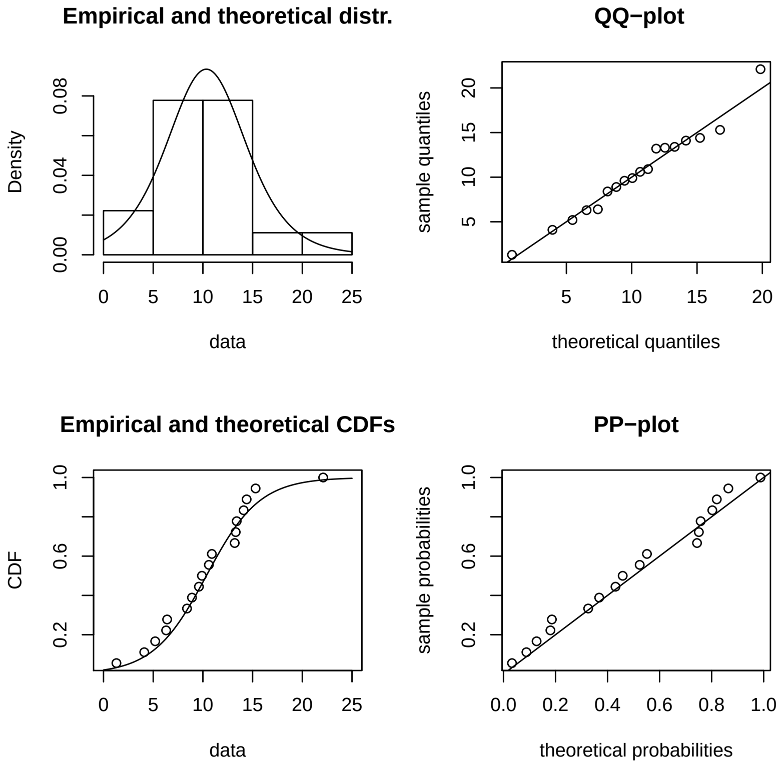


1.3 Fitting of a distribution

One or more parametric distributions may then be fitted to the data set, one at a time, using the fonction `fitdist`. This function uses the maximum likelihood method if the argument `method="mle"` (or if it is omitted) or the matching moments estimation if the argument `method="mme"`. When fitting continuous distributions, Kolmogorov-Smirnov and Anderson-Darling statistics may be computed using the function `gofstat`. Four goodness of fit plots are also provided.

Below is the result of the fit of a logistic distribution by maximum likelihood to the previous dataset.

```
> f1l <- fitdist(x1, "logis")
> summary(f1l)

Fitting of the distribution ' logis ' by maximum likelihood
Parameters :
         estimate Std. Error
location    10.35      1.099
scale        2.67      0.523
Loglikelihood:  -53.6   AIC:  111   BIC:  113
Correlation matrix:
         location    scale
location  1.00000 -0.00915
scale    -0.00915  1.00000

> gofstat(f1l)

Kolmogorov-Smirnov statistic:  0.133
Anderson-Darling statistic:  0.209

> plot(f1l)
```



In that case the Anderson-Darling test may be used to test the adequation of the distribution to data.

```
> gofstat(f1l, print.test = TRUE)

Kolmogorov-Smirnov statistic:  0.133
Kolmogorov-Smirnov test: not calculated
Anderson-Darling statistic:  0.209
Anderson-Darling test:  not rejected
```

Generally goodness-of-fit tests may be used carefully. In somme cases, especially on very big datasets, even if the null hypothesis is rejected a fitted distribution may be chosen as the best one among simple distributions to describe an empirical distribution, if the goodness-of-fit plots do not show strong differences between empirical and theoretical distributions.

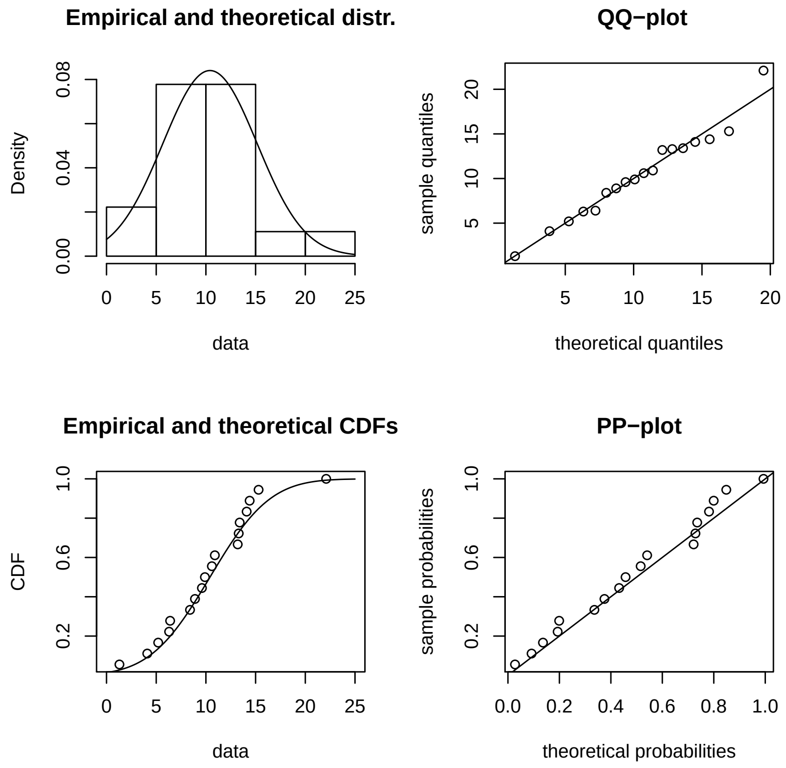
In that case, the fit seems correct, but it is easy to compare goodness-of-fit statistics for other distributions fitted on the same dataset in order to check that other distributions could not give a better fit. Below are computed the goodness-of-fit statistics for logistic, lognormal, gamma, normal and weibull distributions.

```
> gofstat(fitdist(x1, "logis"))

Kolmogorov-Smirnov statistic:  0.133
Anderson-Darling statistic:  0.209

> gofstat(fitdist(x1, "lnorm"))

Kolmogorov-Smirnov statistic:  0.178
Anderson-Darling statistic:  0.793

> gofstat(fitdist(x1, "gamma"))

Kolmogorov-Smirnov statistic:  0.138
Anderson-Darling statistic:  0.457

> gofstat(fitdist(x1, "norm"))

Kolmogorov-Smirnov statistic:  0.110
Anderson-Darling statistic:  0.226

> gofstat(fitdist(x1, "weibull"))

Kolmogorov-Smirnov statistic:  0.121
Anderson-Darling statistic:  0.282
```

Regarding the Kolmogorov-Smirnov statistic, the fit of a normal distribution seems better while regarding the Anderson-Darling statistic the fit of a logistic distribution seems better. It is recommended to base the comparison of fits on the Anderson-Darling statistic when the modelling of the tails of a distribution is important, as it is often the case in risk assessment.

In order to choose between both distributions, the goodness-of-fit plot and the summary of the fit of a normal chould be compared to the ones already obtained for the logistic distribution, but in that case the differences seem very small.

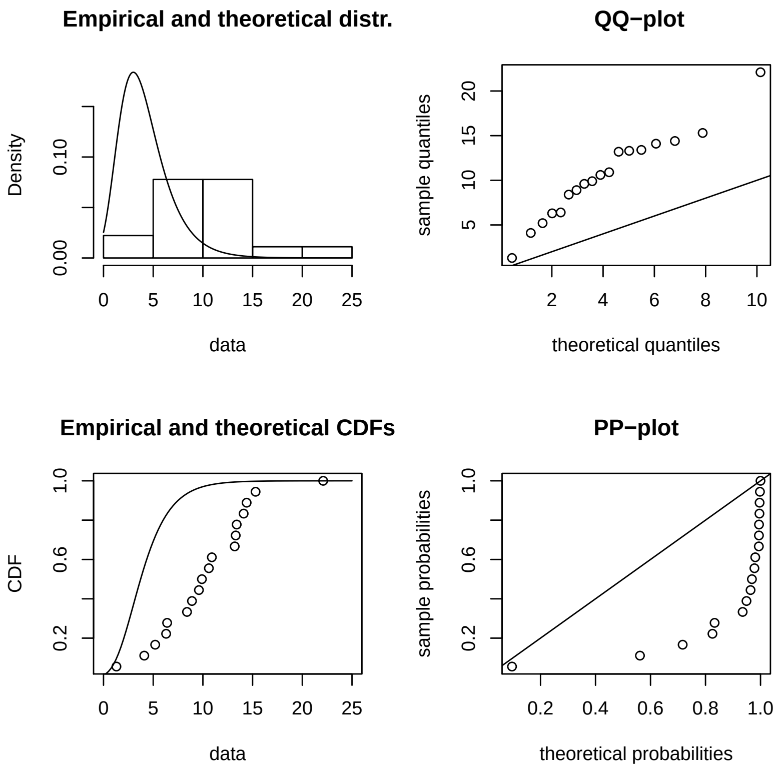
```
> f1n <- fitdist(x1, "norm")
> plot(f1n)
> summary(f1n)

Fitting of the distribution ' norm ' by maximum likelihood
Parameters :
     estimate Std. Error
mean    10.41      1.119
sd       4.75      0.791
Loglikelihood:  -53.6   AIC:  111   BIC:  113
Correlation matrix:
     mean sd
mean    1  0
sd      0  1

> plot(f1n)
```



For some distributions (see the help of `fitdist` for details), it is necessary to specify initial values for the distribution parameters in the argument `start` when using the maximum likelihood method. `start` must be a named list of parameters initial values. The names of the parameters in `start` must correspond exactly to their definition in R or to their definition in a previous R code. The function `plotdist` may help to find correct initial values for the distribution parameters in non trivial cases, by an manual iterative use if necessary.

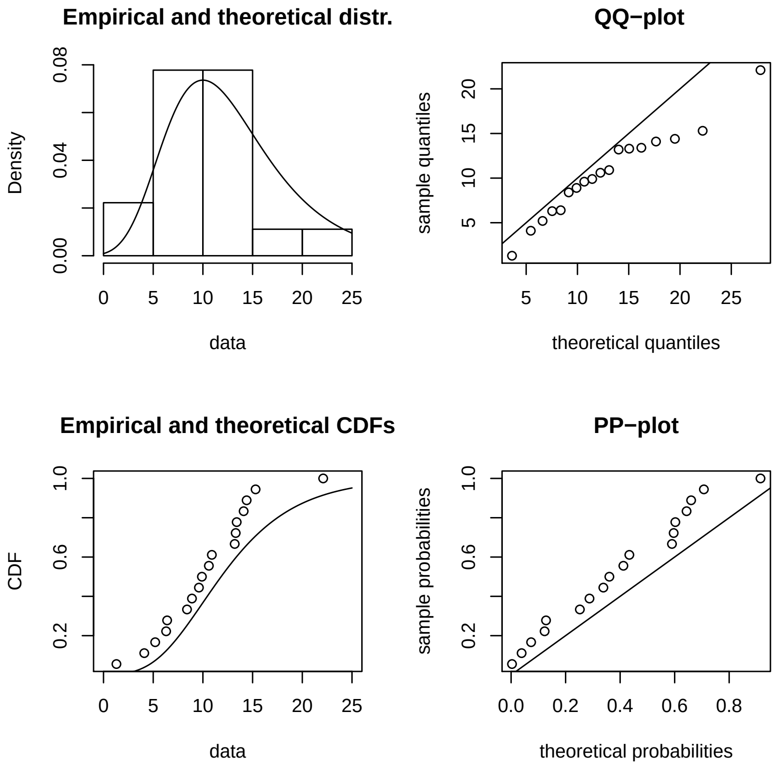
For example, below is the definition of the Gumbel distribution (also named extreme value distribution) and a first plot of the data set with the Gumbel distribution with arbitrary values for parameters.

```
> 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))
> plotdist(x1, "gumbel", para = list(a = 3, b = 2))
```



The same data set may be plotted with a Gumbel distribution with modified values for parameters.

```
> plotdist(x1, "gumbel", para = list(a = 10, b = 5))
```



And a Gumbel distribution may be fitted to data with these values for initial parameter values.

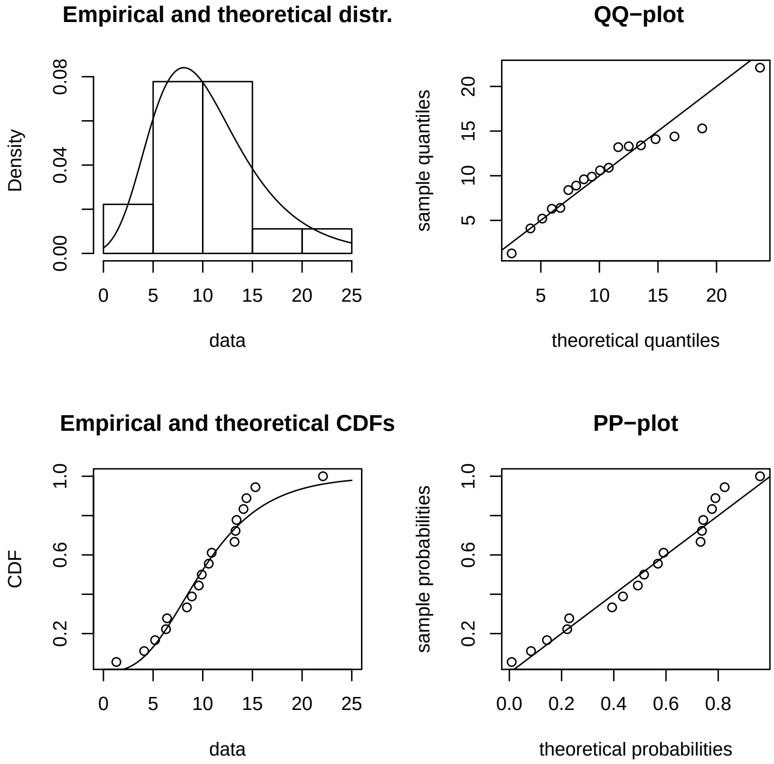
```
> fgu <- fitdist(x1, "gumbel", start = list(a = 10, b = 5))
> plot(fgu)
> summary(fgu)
```

```
Fitting of the distribution ' gumbel ' by maximum likelihood
Parameters :
  estimate Std. Error
a     8.09      1.092
```

```
b     4.38      0.766
Loglikelihood:  -54.1   AIC:  112   BIC:  114
Correlation matrix:
      a     b
a 1.000 0.330
b 0.330 1.000
```

```
> gofstat(fgu)
```

```
Kolmogorov-Smirnov statistic:  0.121
Anderson-Darling statistic:  0.34
```



1.4 Simulation of the uncertainty by boostrap

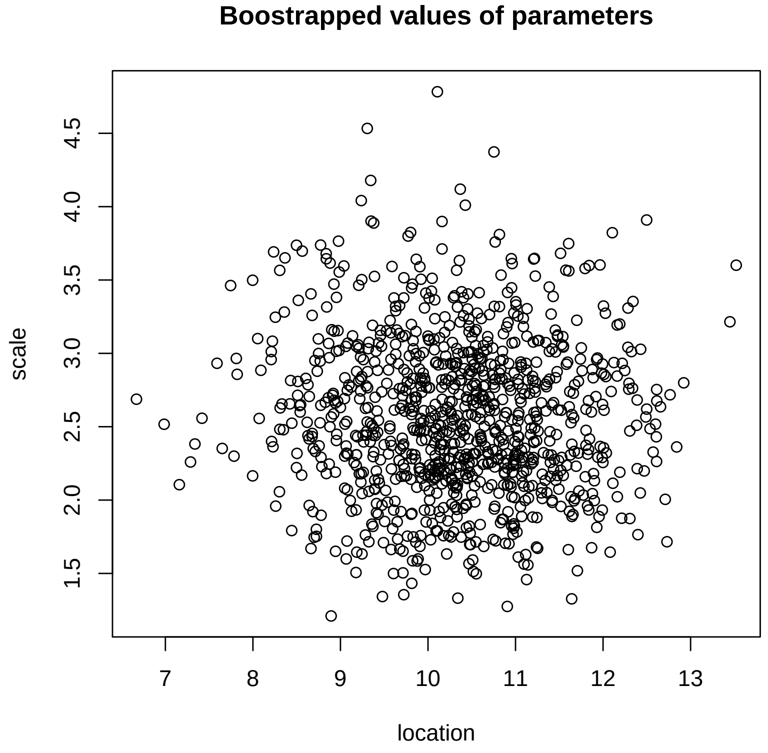
The uncertainty in the parameters of the fitted distribution may be simulated by parametric or nonparametric boostrap using the function `boodist`. This function returns the boostrapped values of parameters which may be plotted to visualize the bootstrap region. It also calculates the 95 percent confidence intervals for each parameter from the 2.5 and 97.5 percentiles of the boostrap values of each parameter (see the help of the function `bootdist` for details).

Below is an example of the use of this function with the previous fit of the logistic distribution.

```
> b1l <- bootdist(f1l)
> plot(b1l)
> summary(b1l)
```

```
Parametric bootstrap medians and 95% percentile CI
         Median 2.5% 97.5%
location  10.38 8.23 12.46
scale      2.56 1.63  3.69
```

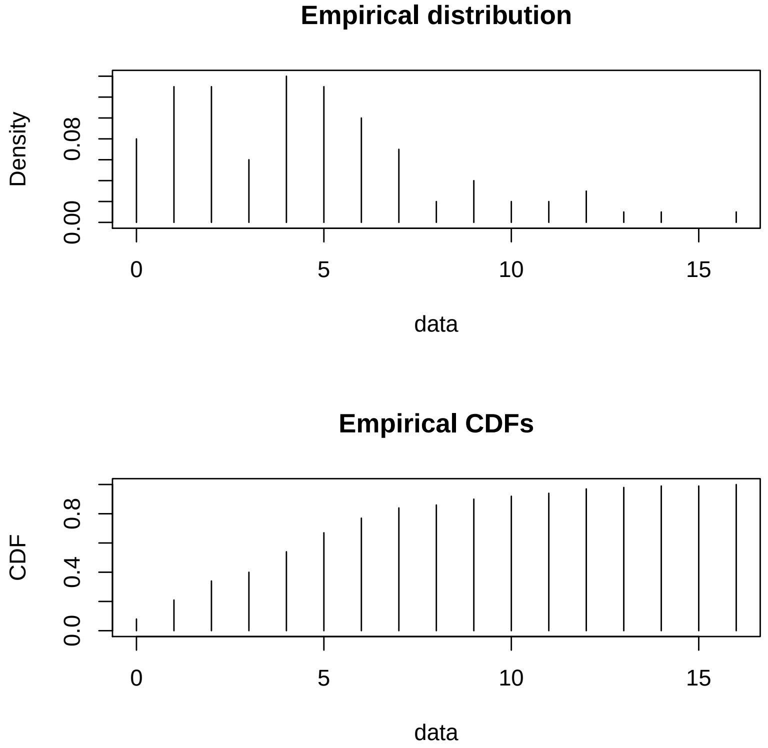
```
Maximum likelihood method converged for  1001  among  1001  iterations
```



2 Specification of a distribution from non-censored discrete data

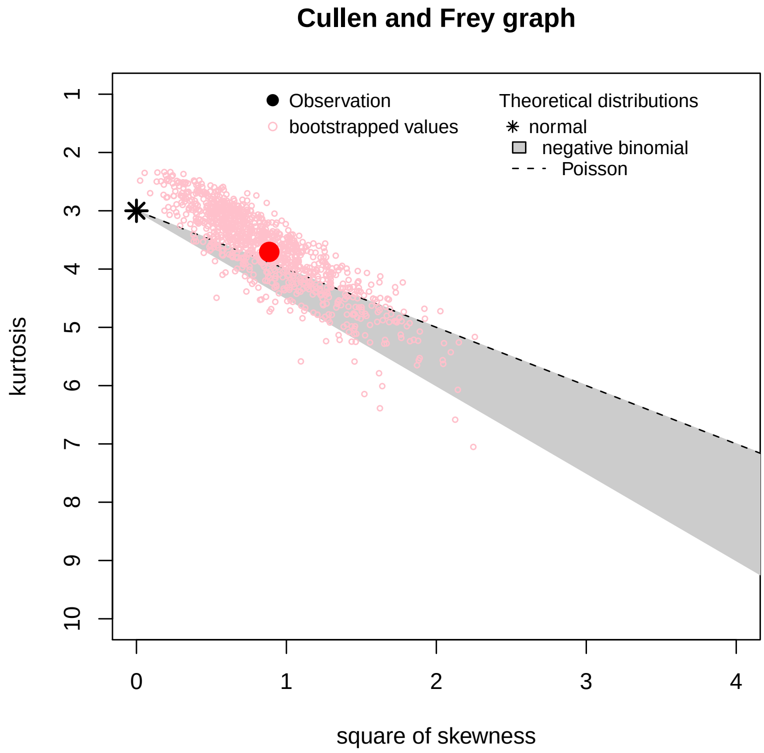
A discrete data set may be considered as a continuous one for example for a large data set from a binomial distribution converging to a normal one. A discrete plot of the distribution may also be provided, fixing the argument discrete of the function `plotdist` to `TRUE`.

```
> x2 <- rnbinom(n = 100, size = 2, prob = 0.3)
> plotdist(x2, discrete = TRUE)
```



As for continuous distributions, descriptive parameters of the empirical distribution may be computed using the function `descdist` which also provides a skewness-kurtosis plot which may help you to choose which distribution(s) to fit.

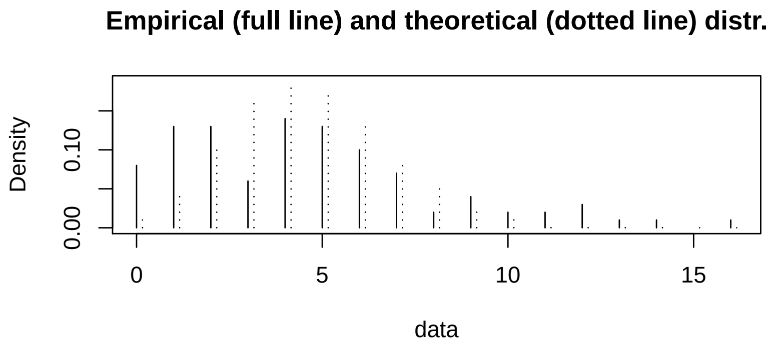
```
> descdist(x2, discrete = T, boot = 1000)

summary statistics
------
min:  0   max:  16
median:  4
mean:  4.6
estimated sd:  3.48
estimated skewness:  0.941
estimated kurtosis:  3.70
```

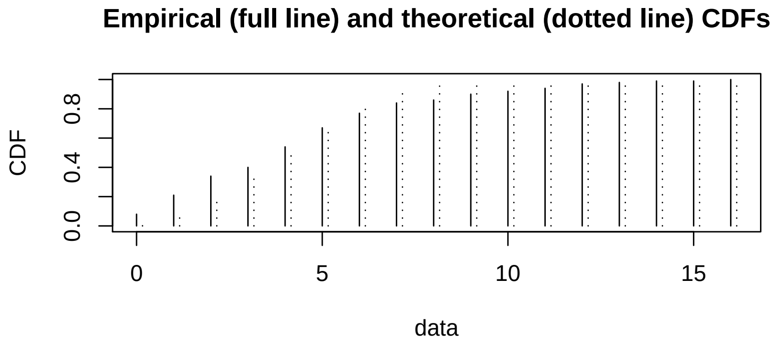


As for continuous distributions, one or more parametric distributions may then be fitted to the data set by maximum likelihood or matching moments.

Below is the result of the fit of a Poisson distribution with the bootstrap simulations.

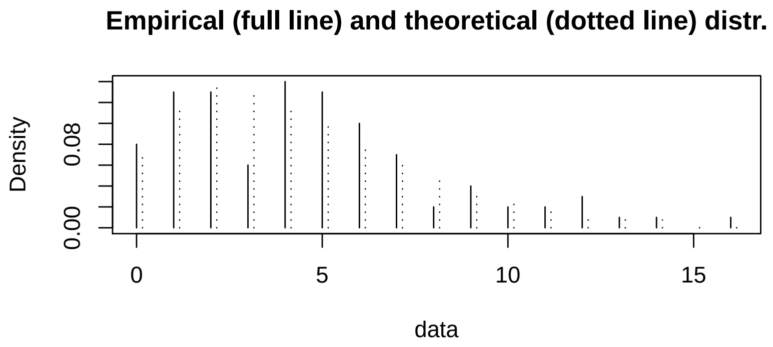
```
> f2p <- fitdist(x2, "pois")
> plot(f2p)
> summary(f2p)

Fitting of the distribution ' pois ' by maximum likelihood
Parameters :
       estimate Std. Error
lambda      4.6      0.214
Loglikelihood:  -286   AIC:  574   BIC:  577

> gofstat(f2p, print.test = TRUE)

Chi-squared statistic:  72.4
Degree of freedom of the Chi-squared distribution:  5
Chi-squared p-value:  3.24e-14
!!! the p-value may be wrong with some theoretical counts < 5 !!!

> b2p <- bootdist(f2p)
> summary(b2p)

Parametric bootstrap medians and 95% percentile CI
Median   2.5%  97.5%
  4.60   4.14   5.01

Maximum likelihood method converged for  1001  among  1001  iterations
```

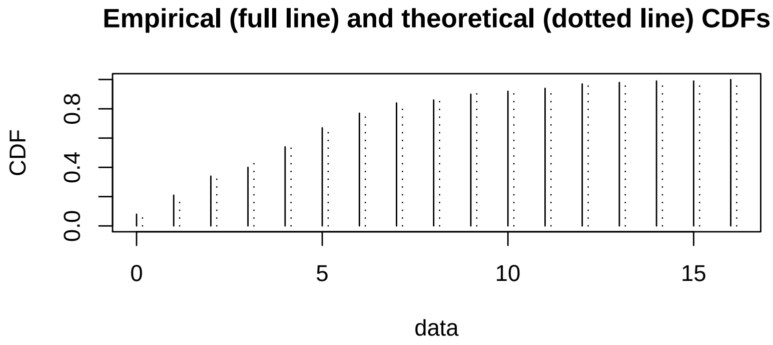




Below is the result of the fit of a negative binomial distribution with the boostrap simulations.

```
> f2n <- fitdist(x2, "nbinom")
> plot(f2n)
> summary(f2n)

Fitting of the distribution ' nbinom ' by maximum likelihood
Parameters :
     estimate Std. Error
size     2.56      0.605
mu       4.60      0.359
Loglikelihood:  -254   AIC:  511   BIC:  516
Correlation matrix:
          size        mu
size  1.000000 -0.000201
mu   -0.000201  1.000000

> gofstat(f2n, print.test = TRUE)

Chi-squared statistic:  3.1
Degree of freedom of the Chi-squared distribution:  4
Chi-squared p-value:  0.541

> b2n <- bootdist(f2n)
> summary(b2n)

Parametric bootstrap medians and 95% percentile CI
     Median 2.5% 97.5%
size   2.63 1.76  4.36
mu     4.60 3.88  5.35

Maximum likelihood method converged for  1001  among  1001  iterations
```





From goodness-of-fit graphs, Chi-squared statistics, AIC and BIC values, it seems better to choose the fit of a negative binomial distribution for this dataset even it has one more parameter than the Poisson one. This was not obvious while looking at the skewness-kurtosis graph. This graph must be used cautiously especially for continuous distributions far from the normal distribution or for discrete distributions. It is only indicative.

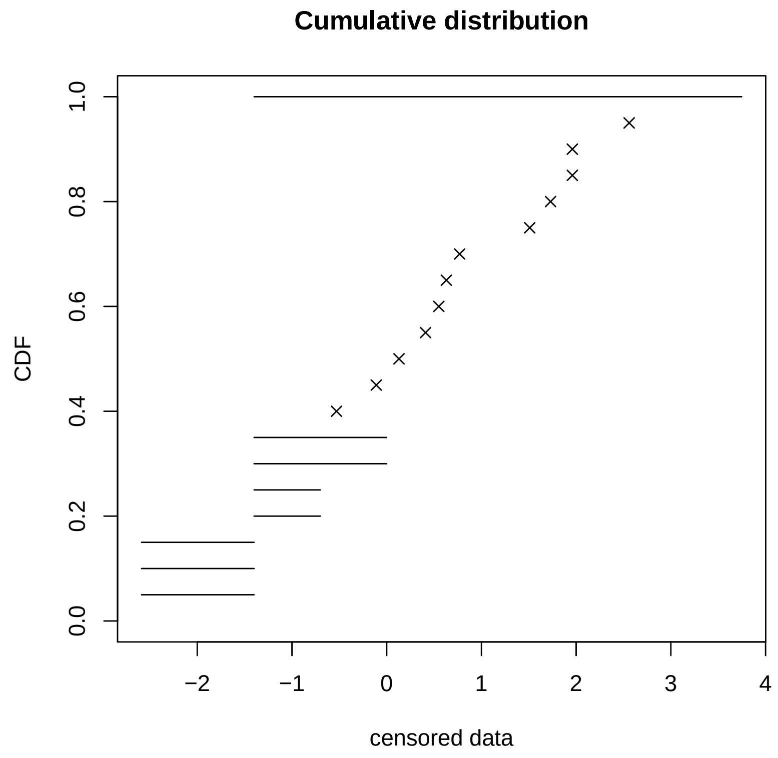
3 Specification of a distribution from censored data

Censored data may contain left censored, right censored and interval censored values, with several lower and upper bounds. Data must be coded into a dataframe with 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.

3.1 Graphical display of the observed distribution

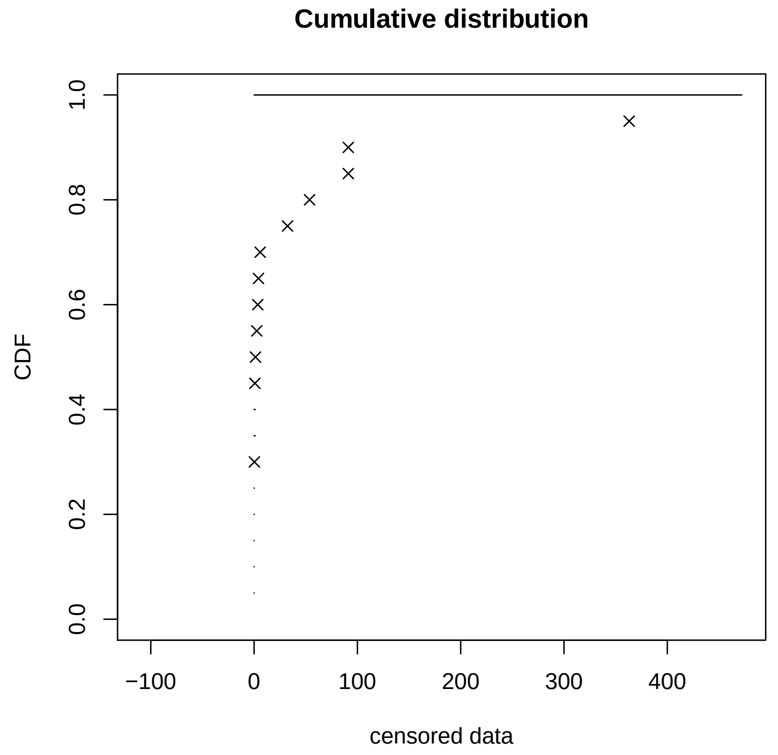
First of all, the observed distribution may be plotted using the function `plotdistcens`. Data are reported directly as segments for interval, left and right censored data, and as points for non-censored data. For more details, see the help of the function `plotdistcens`.

```
> 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)
```



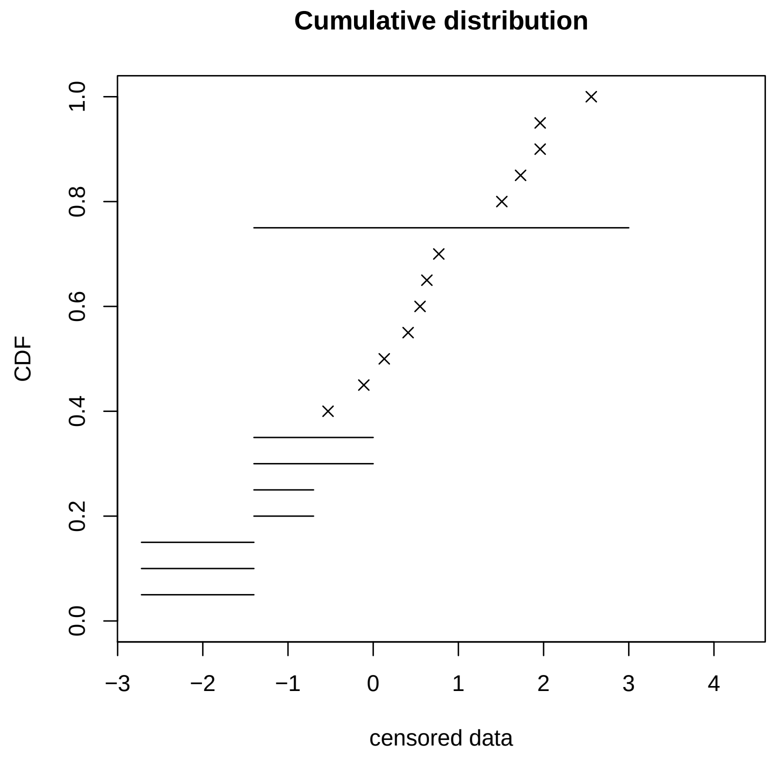
When left or right NA-values correspond to finite value (for example 0 for left NA-values of positive data), the arguments `leftNA` (or `rightNA`) must be affected to this finite value to ensure a correct plot of left (or right) censored observations, as in the example below.

```
> d2 <- data.frame(left = 10^(d1$left), right = 10^(d1$right))
> plotdistcens(d2, leftNA = 0)
```



It is also possible to fix `rightNA` or `leftNA` to a realistic extreme value, even if not exactly known, to obtain a reasonable global ranking of observations, as in the example below for the first dataset.

```
> plotdistcens(d1, rightNA = 3)
```

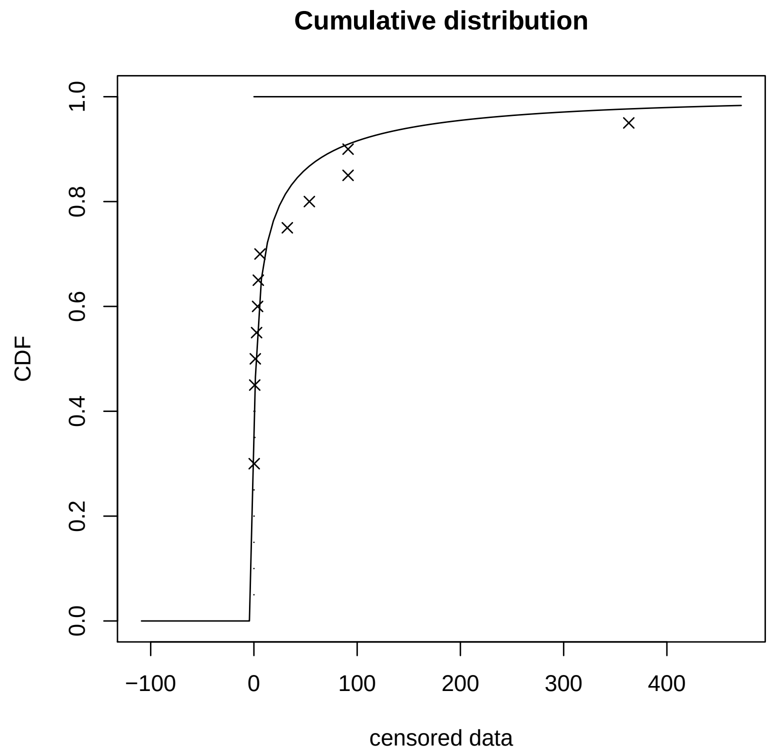


3.2 Fitting of a distribution

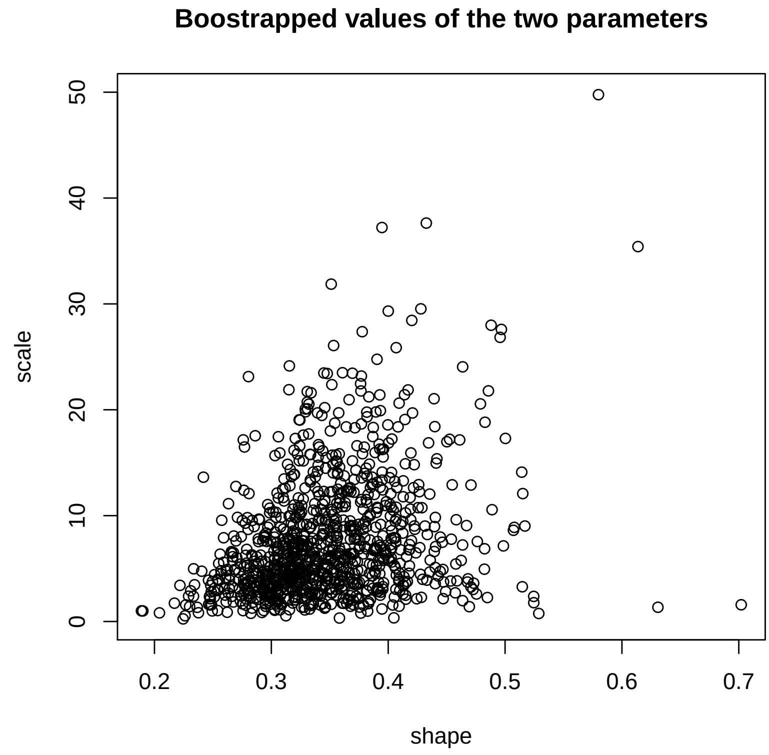
One or more parametric distributions may then be fitted to the censored data set, one at a time, using the fonction `fitdistcens`. This function always uses the maximum likelihood method. For more details, see the help of the function `fitdistcens`. Only one goodness of fit plot is provided for censored data, in cumulative frequencies. The uncertainty in the parameters of the fitted distribution may be simulated by nonparametric boostrap only, using the function `boodistcens`.

Below is the result of a fit of a Weibull distribution by maximum likelihood and the results of the corresponding boostrap simulations.

```
> f2w <- fitdistcens(d2, "weibull")
> summary(f2w)

FITTING OF THE DISTRIBUTION ' weibull ' BY MAXIMUM LIKELIHOOD ON CENSORED DATA
PARAMETERS
      estimate Std. Error
shape    0.324     0.0613
scale    6.124     4.5872
Loglikelihood:  -68.5   AIC:  141   BIC:  143
Correlation matrix:
      shape scale
shape 1.000 0.326
scale 0.326 1.000

> plot(f2w, leftNA = 0)
```



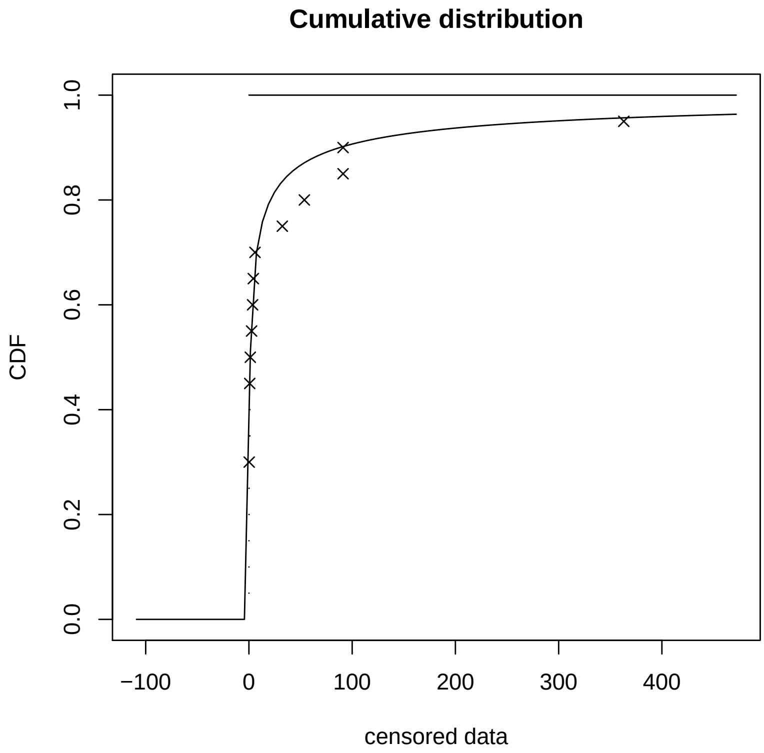
```
> b2w <- bootdistcens(f2w)
> summary(b2w)

Nonparametric bootstrap medians and 95% percentile CI
      Median  2.5%  97.5%
shape  0.339 0.247  0.474
scale  6.308 1.213 25.902

Maximum likelihood method converged for  1001  among  1001  iterations

> plot(b2w)
```



Goodness of fit statistics are not computed for fit on censored data, so the quality of fit may only be estimated from the loglikelihood and the goodness of fit plot.

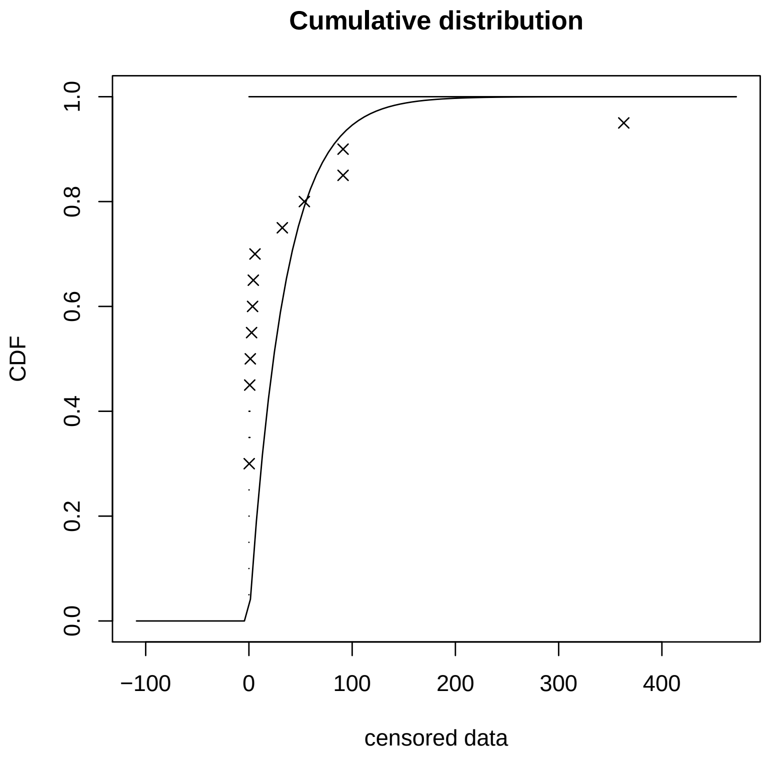
Below is the fit of a lognormal distribution to the same censored data set.

```
> f2l <- fitdistcens(d2, "lnorm")
> summary(f2l)

FITTING OF THE DISTRIBUTION ' lnorm ' BY MAXIMUM LIKELIHOOD ON CENSORED DATA
PARAMETERS
        estimate Std. Error
meanlog     0.27      0.764
sdlog       3.28      0.600
Loglikelihood:  -68.7   AIC:  141   BIC:  143
Correlation matrix:
        meanlog   sdlog
meanlog  1.0000 -0.0739
sdlog   -0.0739  1.0000

> plot(f2l, leftNA = 0)
```



Below is the fit of an exponential distribution.

```
> f2e <- fitdistcens(d2, "exp")
> summary(f2e)

FITTING OF THE DISTRIBUTION ' exp ' BY MAXIMUM LIKELIHOOD ON CENSORED DATA
PARAMETERS
     estimate Std. Error
rate   0.0292    0.00668
Loglikelihood:  -99.6   AIC:  201   BIC:  202

> plot(f2e, leftNA = 0)
```



As with `fitdist`, for some distributions (see the help of `fitdistcens` for details), it is necessary to specify initial values for the distribution parameters in the argument `start`. `start` must be a named list of parameters initial values. The names of the parameters in `start` must correspond exactly to their definition in R or to their definition in a previous R code. The function `plotdistcens` may help to find correct initial values for the distribution parameters in non trivial cases, by an manual iterative use if necessary, as explained previously for non-censored continuous data.

4 Changing the optimization algorithm used to maximize the likelihood

Sometimes the default algorithm used to maximize the likelihood fails to converge. It may then be interesting to change some options of the function `optim` or to use another optimization function than `optim` to maximize the likelihood.

4.1 Changing the arguments passed to `optim`

The argument optim.method may be used in the call to `fitdist` or `fitdistcens`. It will internally be passed to `mledist` and to `optim`. This argument may be fixed to `"Nelder-Mead"` (the robust Nelder and Mead method), `"BFGS"` (the BFGS quasi-Newton method), `"CG"` (a conjugate gradients method), `"SANN"` (a variant of simulated annealing) or `"L-BFGS-B"` (a modification of the BFGS quasi-Newton method which enables box constraints optimization). For the use of the last method the arguments `lower` and/or `upper` also have to be passed. More details on these optimization functions may be found in the help page of `optim` from the package `stats`.

Below are examples of fits of a gamma distribution to non censored data with various options of `optim`.

```
> fitdist(x1, "gamma", optim.method = "Nelder-Mead")

Fitting of the distribution ' gamma ' by maximum likelihood
Parameters:
      estimate Std. Error
shape    3.575      1.140
rate     0.343      0.118

> fitdist(x1, "gamma", optim.method = "BFGS")

Fitting of the distribution ' gamma ' by maximum likelihood
Parameters:
      estimate Std. Error
shape    3.577      1.141
rate     0.344      0.118

> fitdist(x1, "gamma", optim.method = "L-BFGS-B", lower = c(0,
+     0))
```

```
Fitting of the distribution ' gamma ' by maximum likelihood
Parameters:
      estimate Std. Error
shape    3.574      1.140
rate     0.343      0.118

> fitdist(x1, "gamma", optim.method = "SANN")

Fitting of the distribution ' gamma ' by maximum likelihood
Parameters:
      estimate Std. Error
shape    3.561      1.136
rate     0.343      0.118
```

4.2 Supplying another optimization function

You may also want to use another function than `optim` to maximize the likelihood. This optimization function has to be specified by the argument `custom.optim` in the call to `fitdist` or `fitdistcens`. But before that, it is necessary to customize this optimization function : `custom.optim` function must have (at least) the following arguments, `fn` for the function to be optimized, `par` for the initialized parameters. 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`.

Below is an example of code written to customize `genoud` function from `rgenoud` package.

```
mygenoud <- function(fn, par, ...)
{
   require(rgenoud)
   res <- genoud(fn, starting.values=par, ...)
   standardres <- c(res, convergence=0)
   return(standardres)
}
```

The customized optimization function may then be passed as the argument `custom.optim` in the call to `fitdist` or `fitdistcens`. The following code may for example be used to fit a gamma distribution to the non censored data `x1`. Note that in this example various arguments are also passed from `fitdist` to `genoud` : `nvars`, `Domains`, `boundary.enforcement`, `print.level` and `hessian`.

```
fitdist(x1, "gamma", custom.optim=mygenoud, nvars=2,
   Domains=cbind(c(0,0), c(10, 10)), boundary.enforcement=1,
   print.level=1, hessian=TRUE)
```