Package ‘truncgof’

February 20, 2015

Type Package

Title GoF tests allowing for left truncated data

Version 0.6-0

Date 2012-12-24

Author Thomas Wolter <thwolter@gmail.com>

Maintainer Renato Vitolo <renato.vitolo@banca.mps.it>

Depends R (>= 2.4.0),MASS

Description Goodness-of-fit tests and some adjusted exploratory tools allowing for left truncated data

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-12-25 08:11:28

NeedsCompilation yes

R topics documented:

ad.test *Supremum Class Anderson-Darling test*

Description

Supremum class version of the Anderson-Darling test providing a comparison of a fitted distribution with the empirical distribution.

Usage

```
ad.test(x, distn, fit, H = NA,
        alternative = c("two.sided", "less", "greater"),
        sim = 100, tol = 1e-04, estfun = NA)
```

Arguments

`x`	a numeric vector of data values
`distn`	character string naming the null distribution function
`fit`	list of null distribution parameters
`H`	a treshold value
`alternative`	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater". Initial letter must be specified only.
`sim`	maximum number of szenarios in the Monte-Carlo simulation
`tol`	if the difference of two subsequent p-value calculations is lower than `tol` the Monte-Carlo simulation stops
`estfun`	an function as character string or `NA` (default). See `mctest`.

Details

The supremum class Anderson-Darling test compares the null distribution with the empirical distribution of the observed data. The test statistic is given by

$$AD^{+} = \sqrt{n} \sup_j \{ \frac{z_H + \frac{j}{n}(1 - z_H) - z_j}{\sqrt{(z_j - z_H)(1 - z_j)}} \}$$

$$AD^{-} = \sqrt{n} \sup_j \{ \frac{z_j - (z_H + \frac{j-1}{n}(1 - z_H))}{\sqrt{(z_j - z_H)(1 - z_j)}} \}$$

$$AD = \max\{AD^{+}, AD^{-}\}.$$

with $z_H = F_\theta(H)$ and $z_j = F_\theta(x_j)$, where $x_1, \ldots, x_n$ are the ordered data values. Here, F_θ is the null distribution.

Value

A list with class "mchtest" containing the following components

`statistic`	the value of the Supremum Class Anderson-Darling statistic
`treshold`	the treshold value
`p.value`	the p-value of the test
`data.name`	a character string giving the name of the data
`method`	the character string "Supremum Class Anderson-Darling test"
`alternative`	the alternative
`sim.no`	number of simulated szenarios in the Monte-Carlo simulation

References

Chernobay, A., Rachev, S., Fabozzi, F. (2005), *Composites goodness-of-fit tests for left-truncated loss samples*, Tech. rep., University of Calivornia Santa Barbara

See Also

`ks.test`, `v.test`, `adup.test` for other supremum class tests and `ad2.test`, `ad2up.test`, `w2.test` for quadratic class tests. For more details see `mctest`.

Examples

```
set.seed(123)
treshold <- 10
xc  <- rlnorm(100, 2, 2)    # complete sample
xt <- xc[xc >= treshold]    # left truncated sample
ad.test(xt, "plnorm", list(meanlog = 2, sdlog = 2), H = 10)
```

`ad2.test` *Quadratic Class Anderson-Darling test*

Description

Quadratic class Anderson-Darling test providing a comparison of a fitted distribution with the empirical distribution.

Usage

```
ad2.test(x, distn, fit, H = NA, sim = 100, tol = 1e-04, estfun = NA)
```

Arguments

`x`	a numeric vector of data values
`distn`	character string naming the null distribution
`fit`	list of null distribution parameters
`H`	a treshold value
`sim`	maximum number of szenarios in the Monte-Carlo simulation
`tol`	if the difference of two subsequent p-value calculations is lower than `tol` the Monte-Carlo simulation is discontinued
`estfun`	an function as character string or `NA` (default). See `mctest`.

Details

The Anderson-Darling test compares the null distribution with the empirical distribution function of the observed data, where left truncated data samples are allowed. The test statistic is given by

$$AD^2 = -n + 2n\log(1 - z_H) - \frac{1}{n}\sum_{j=1}^{n}(1 + 2(n-j))\log(1 - z_j) + \frac{1}{n}\sum_{j=1}^{n}(1 - 2j)\log(z_j - z_H)$$

with $z_H = F_\theta(H)$ and $z_j = F_\theta(x_j)$, where $x_1, \ldots, x_n$ are the ordered data values. Here, F_θ is the null distribution.

Value

A list with class "mchtest" containing the following components

`statistic`	the value of the Quadratic Class Anderson-Darling statistic
`treshold`	the treshold value
`p.value`	the p-value of the test
`data.name`	a character string giving the name of the data
`method`	the character string "Quadratic Class Anderson-Darling test"
`sim.no`	number of simulated szenarios in the Monte-Carlo simulation

References

Chernobay, A., Rachev, S., Fabozzi, F. (2005), *Composites goodness-of-fit tests for left-truncated loss samples*, Tech. rep., University of Calivornia Santa Barbara

See Also

`ad2up.test`, `w2.test` for other quadratic class tests and `ks.test`, `v.test`, `adup.test`, `ad.test` for supremum class tests. For more details see `mctest`.

Examples

```
set.seed(123)
treshold <- 10
xc  <- rlnorm(1000, 2, 2)     # complete sample
xt <- xc[xc >= treshold]      # left truncated sample
ad2.test(xt, "plnorm", list(meanlog = 2, sdlog = 2), H = 10)
```

`ad2up.test` *Quadratic Class Upper Tail Anderson-Darling test*

Description

Quadratic Class Upper Tail Anderson-Darling test providing a comparison of a fitted distribution with the empirical distribution.

Usage

```
ad2up.test(x, distn, fit, H = NA, sim = 100, tol = 1e-04, estfun = NA)
```

Arguments

`x`	a numeric vector of data values
`distn`	character string naming the null distribution
`fit`	list of null distribution parameters
`H`	a treshold value
`sim`	maximum number of szenarios in the Monte-Carlo simulation
`tol`	if the difference of two subsequent p-value calculations is lower than `tol` the Monte-Carlo simulation is discontinued
`estfun`	an function as character string or `NA` (default). See `mctest`.

Details

The Anderson-Darling test compares the null distribution with the empirical distribution function of the observed data, where left truncated data samples are allowed. The test statistic is given by

$$AD_{up}^2 = -2n\log(1-z_H) + 2\sum_{j=1}^{n}\log(1-z_j) + \frac{1-z_H}{n}\sum_{j=1}^{n}(1+2(n-j))\frac{1}{1-z_j}$$

with $z_H = F_\theta(H)$ and $z_j = F_\theta(x_j)$, where $x_1, \ldots, x_n$ are the ordered data values. Here, F_θ is the null distribution.

Value

A list with class "mchtest" containing the following components

`statistic`	the value of the Quadratic Class Upper Tail Anderson-Darling statistic
`treshold`	the treshold value
`p.value`	the p-value of the test
`data.name`	a character string giving the name of the data
`method`	the character string "Quadratic Class Upper Tail Anderson-Darling Test"
`sim.no`	number of simulated szenarios in the Monte-Carlo simulation

References

Chernobay, A., Rachev, S., Fabozzi, F. (2005), *Composites goodness-of-fit tests for left-truncated loss samples*, Tech. rep., University of Calivornia Santa Barbara

See Also

`ad2.test`, `w2.test` for other quadratic class tests and `ks.test`, `v.test`, `adup.test`, `ad.test` for supremum class tests. For more details see `mctest`.

Examples

```
set.seed(123)
treshold <- 10
xc  <- rlnorm(100, 2, 2)     # complete sample
xt <- xc[xc >= treshold]     # left truncated sample
ad2up.test(xt, "plnorm", list(meanlog = 2, sdlog = 2), H = 10)
```

---

adup.test *Supremum Class Upper Tail Anderson-Darling test*

Description

Supremum class version of the Upper Tail Anderson-Darling test providing a comparison of a fitted distribution with the empirical distribution.

Usage

```
adup.test(x, distn, fit, H = NA,
          alternative = c("two.sided", "less", "greater"),
          sim = 100, tol = 1e-04, estfun = NA)
```

Arguments

`x`	a numeric vector of data values
`distn`	character string naming the null distribution
`fit`	list of null distribution parameters
`H`	a treshold value
`alternative`	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater". Initial letter must be specified only.
`sim`	maximum number of szenarios in the Monte-Carlo simulation
`tol`	if the difference of two subsequent p-value calculations is lower than `tol` the Monte-Carlo simulation is discontinued
`estfun`	an function as character string or `NA` (default). See `mctest`.

Details

The supremum class Upper Tail Anderson-Darling test compares the null distribution with the empirical distribution function of the observed data. The test statistic is given by

$$ADup^{+} = \sqrt{n}\sup_{j}\{\frac{\frac{j}{n} - z_j}{1 - z_j}\}$$

$$ADup^{-} = \sqrt{n}\sup_{j}\{\frac{z_j - \frac{j-1}{n}}{1 - z_j}\}$$

$$ADup = \max\{ADup^{+}, ADup^{-}\},$$

with $z_H = F_\theta(H)$ and $z_j = F_\theta(x_j)$, where $x_1, \ldots, x_n$ are the ordered data values. Here, F_θ is the null distribution.

Value

A list with class "mchtest" containing the following components

`statistic`	the value of the Supremum Class Upper Tail Anderson-Darling statistic
`treshold`	the treshold value
`p.value`	the p-value of the test
`data.name`	a character string giving the name of the data
`method`	the character string "Supremum Class Upper Tail Anderson-Darling test"
`alternative`	the alternative
`sim.no`	number of simulated szenarios in the Monte-Carlo simulation

References

Chernobay, A., Rachev, S., Fabozzi, F. (2005), *Composites goodness-of-fit tests for left-truncated loss samples*, Tech. rep., University of Calivornia Santa Barbara

See Also

ks.test, v.test, ad.test for supremum class tests and ad2.test, w2.test for other quadratic class tests. For more details see mctest.

Examples

```
set.seed(123)
treshold <- 10
xc  <- rlnorm(100, 2, 2)     # complete sample
xt <- xc[xc >= treshold]     # left truncated sample
adup.test(xt, "plnorm", list(meanlog = 2, sdlog = 2), H = 10)
```

---

cdens *Build a conditional density function*

Description

For a given distribution function cdens builds a conditional density function with respect to a relevant treshold.

Usage

```
cdens(distn, H)
```

Arguments

distn	character string naming the distribution function for which the conditional density is to be built
H	a treshold value

Details

For $x \geq H$ the conditional density f^* of a density f is given by

$$f^*_\theta(x) = f(x|x \geq H) = \frac{f_\theta(x)}{1 - F_\theta(H)},$$

with θ the parameters of the distribution, F the cumulative distribution function and H the treshhold value. For $x < H$, f^* disappear.

Value

The conditional density of the specified density function with arguments x, the relevant parameters and the treshold H predefined as the value of cdens' argument H. x can be a numeric value or numeric vector, but must be greater or equal to H.

See Also

density functions, e.g. dlnorm, dgamma, etc.

Examples

```
require(MASS)
set.seed(123)
treshold <- 10
xc  <- rlnorm(100, 2, 2)   # complete sample
xt <- xc[xc >= treshold]   # left truncated sample

clnorm <- cdens("plnorm", H = treshold)
args(clnorm)

# mle fitting based on the complete sample
start <- list(meanlog = 2, sdlog = 1)
fitdistr(xc, dlnorm, start = start)

# mle fitting based on the truncated sample
fitdistr(xt, clnorm, start = start)

# in contrast
fitdistr(xt, dlnorm, start = start)
```

---

dplot *Plot of the distribution functions*

Description

Plot the empirical against the theoretical distibution function.

Usage

```
dplot(x, distn, parm, H = NA, verticals = FALSE, ...)
```

Arguments

`x`	a numeric vector of data samples
`distn`	character string naming the theoretical distribution function
`parm`	list of theoretical distribution parameters
`H`	a treshold value
`verticals`	see 'plot.stepfun'
`...`	graphical parameters can be given as arguments to `plot`

Details

The empirical and the theoretical distribution function specified by the arguments `distn` and `parm` are plotted in one single graphic. For truncated data values it is important to assign the treshold value `H`.

See Also

ecdf, plot.ecdf, plot.stepfun

Examples

```
xc <- rnorm(25)     # complete sample
xt <- xc[xc >= 0]   # left truncated sample

# df of the complete sample
dplot(xc, "pnorm", list(0,1), vertical = TRUE)

# df of the left truncated sample
dplot(xt, "pnorm", list(0,1), H = 0, vertical = TRUE)
```

---

edf *Empirical distribution function*

Description

Empircal distribution function of left truncated data with known distribution.

Usage

```
edf(x, distn = NA, parm = NA, H = NA)
```

Arguments

x	a numerical vector of data values
distn	character string naming the distribution function
parm	list of distribution parameters
H	a treshold value

Details

edf is a version of ecdf allowing left truncated data. If distn is not assigned all other arguments except x are ignored and the result is exactly the same as of ecdf.

Value

A function of class "stepfun".

See Also

ecdf, dplot

Examples

```
set.seed(123)
treshold <- 10
xc <- rlnorm(30, meanlog = 2, sdlog = 1)    # complete sample
xt <- xc[xc >= treshold]                    # truncated sample

# the results are identical:
plot(edf(xc))
plot(ecdf(xc))

# considering truncated samples:
plot(edf(xt))   # wrong plot
plot(edf(xt, "plnorm", list(meanlog = 2, sdlog = 1), H = 10))
```

---

ks.test *Kolmogorov-Smirnov test*

Description

Kolmogorov-Smirnov test providing a comparison of a fitted distribution with the empirical distribution.

Usage

```
ks.test(x, distn, fit, H = NA,
        alternative = c("two.sided", "less", "greater"),
        sim = 100, tol = 1e-04, estfun = NA)
```

Arguments

`x`	a numeric vector of data values
`distn`	character string naming the null distribution
`fit`	list of null distribution parameters
`H`	a treshold value
`alternative`	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater". Initial letter must be specified only.
`sim`	maximum number of szenarios in the Monte-Carlo simulation
`tol`	if the difference of two subsequent p-value calculations is lower than `tol` the Monte-Carlo simulation is discontinued
`estfun`	an function as character string or `NA` (default). See `mctest`.

Details

The Kolmogorov-Smirnov test compares the null distribution with the empirical distribution function of the observed data, where left truncated data samples are allowed. The test statistic is given by

$$KS^+ = \frac{\sqrt{n}}{1 - z_H} \sup_j \{z_H + \frac{j}{n}(1 - z_H) - z_j\}$$

$$KS^- = \frac{\sqrt{n}}{1 - z_H} \sup_j \{z_j - (z_H + \frac{j-1}{n}(1 - z_H))\}$$

$$KS = \max\{KS^+, KS^-\},$$

with $z_H = F_\theta(H)$ and $z_j = F_\theta(x_j)$, where $x_1, \ldots, x_n$ are the ordered data values. Here, F_θ is the null distribution.

Value

A list with class "mchtest" containing the following components

`statistic`	the value of the Kolmogorov-Smirnov statistic
`treshold`	the treshold value
`p.value`	the p-value of the test
`data.name`	a character string giving the name of the data
`method`	the character string "Kolmorov-Smirnov test"
`alternative`	the alternative
`sim.no`	number of simulated szenarios in the Monte-Carlo simulation

References

Chernobay, A., Rachev, S., Fabozzi, F. (2005), *Composites goodness-of-fit tests for left-truncated loss samples*, Tech. rep., University of Calivornia Santa Barbara

See Also

`ad.test`, `v.test`, `adup.test` for other supremum class tests and `ad2.test`, `ad2up.test`, `w2.test` for quadratic class tests. For more details see `mctest`.

Examples

```
set.seed(123)
treshold <- 10
xc  <- rlnorm(100, 2, 2)     # complete sample
xt <- xc[xc >= treshold]     # left truncated sample
ks.test(xt, "plnorm", list(meanlog = 2, sdlog = 2), H = 10)
```

mctest *Monte-Carlo simulation based GoF test*

Description

Performs Monte-Carlo based Goodness-of-Fit tests. `mctest` is called by the GoF tests defined in this package. For internal use only.

Usage

```
mctest(x, distn, parm, H, sim, tol, STATISTIC, estfun)
```

Arguments

`x`	numerical vector of data values
`distn`	character string specifying the null distribution
`parm`	parameters of the null distribution
`H`	a treshold value
`sim`	maximum number of szenarios within the Monte-Carlo-Simulation
`tol`	if the difference of two subsequent p-value calculations is lower than `tol` the Monte-Carlo simulation stops
`STATISTIC`	function of the test statistic
`estfun`	an function as character string or `NA`, see details.

Details

From the fitted null distribution `mctest` draws samples each with length of the observed sample `x` and with treshold `H`. The random numbers are taken from the conditional distribution with support $[H, \infty)$. The maximum number of samples is specified by `sim`. For each of these samples the conditional distribution is fitted and the statistic given in `STATISTIC` is calculated. The p-value is the proportion of times the sample statistics values exceed the statistic value of the observed sample.

For each szenario sample `mctest` uses a Maximum-likelihood fitting of the distribution `distn` as default. This is done by direct optimization of the log-likelihood function using `optim`.

Alternativly the fitting parameters for the szenario samples might be estimated with a user-specified function assigned in `estfun`. It must be a function with argument `x` (and `H` if desired) which can be parsed. The return value of the evaluated function must be a list with the parameters which should be fitted. Inside `mctest` the evaluation of `estfun` is performed with `H` as assigned in the call of `mctest` and `x` the szenario sample.

By assigning a function to `estfun`, the fitting procedure can be done faster and more appropriate to a given problem. The 'evir' package for example defines a function `gpd` to fit the Generalized Pareto Model. To start a test it is more reasonable to set `estfun = gpd(x, y)`, where `y` must be a defined numeric value.

Value

named list of

TS	value of the test statistic for x
p.value	Monte-Carlo simulation based p-value of the statistic
sim	number of simulated szenarios in the Monte-Carlo simulation

References

Chernobay, A., Rachev, S., Fabozzi, F. (2005), *Composites goodness-of-fit tests for left-truncated loss samples*, Tech. rep., University of Calivornia Santa Barbara

Ross, S. M. (2002), *Simulation*, 3rd Edition, Academic Press. Pages 205-208.

See Also

ad2.test, ad2up.test, w2.test for quadratic class GoF tests and ks.test, v.test, adup.test, ad.test for supremum class GoF tests.

Examples

```
set.seed(123)
treshold <- 10
xc <- rgamma(100, 20, 2)    # complete sample
xt <- xc[xc > treshold]     # left truncated sample

## function for parameter fitting
estimate <- function(x, H){
    cgamma <- cdens("pgamma", H)
    ll <- function(p, y) {
        res <- -sum(do.call("cgamma", list(c(y), p[1], p[2], log = TRUE)))
        if (!is.finite(res)) return(-log(.Machine$double.xmin)*length(x))
        return(res)
    }
    est <- optim(c(1,1), ll, y = x, lower = c(.Machine$double.eps, 0),
                 method = "L-BFGS-B")
    as.list(est$par)
}

fit <- estimate(xt, treshold)
cat("fitting parameters:", unlist(fit), "\n")

## calculate p-value with fitting algorithm defined in 'mctest' ...
ad2up.test(xt, "pgamma", fit, H = treshold,  estfun = NA, tol = 1e-02)

## ... or with the function 'estimate'
ad2up.test(xt, "pgamma", fit, H = treshold, estfun = "estimate(x, H)",
           tol = 1e-02)

## not run:
## if the 'evir' package is loaded:
## ad.test(xt, "pgpd", list(2,3), H = treshold,
```

```
##        estfun = "as.list(gpd(x, 0)$par.ests)", tol = 1e-02)
```

qplot *QQ-Plot*

Description

Adjusted QQ-Plot allowing for left-truncated data values.

Usage

```
qplot(x, distn, parm, H = NA, plot.it = TRUE, main = "QQ-Plot",
      xlab = "empirical quantiles", ylab = "theoretical quantiles", ...)
```

Arguments

`x`	a numeric vector of data values
`distn`	character string of the distribution
`parm`	list of distribution parameters
`H`	a treshold value
`plot.it`	logical. `TRUE` (default) if the result is to be plotted.
`xlab, ylab, main`	plot labels
`...`	further graphical parameters

See Also

`dplot`, `cdens`

Examples

```
set.seed(123)
treshold <- 10
xc  <- rlnorm(100, 2, 2)     # complete sample
xt <- xc[xc >= treshold]     # left truncated sample

# for not assigned treshold the folliwing
# graphics are identical but not usefull
par(mfrow = c(2,1))
y <- qlnorm(ppoints(length(xt)), 2, 2)
qplot(xt, "plnorm", list(2,2))
qqplot(xt, y); abline(0,1)

# fot trucated data rather use
qplot(xt, "plnorm", list(2,2), H = 10)
```

`v.test` *Kuiper test*

Description

Kuiper test providing a comparison of a fitted distribution with the empirical distribution.

Usage

```
v.test(x, distn, fit, H = NA, sim = 100, tol = 1e-04, estfun = NA)
```

Arguments

`x`	a numeric vector of data values
`distn`	character string naming the null distribution
`fit`	list of distribution parameters
`H`	a treshold value
`sim`	maximum number of szenarios in the Monte-Carlo simulation
`tol`	if the difference of two subsequent p-value calculations is lower than `tol` the Monte-Carlo simulation stops
`estfun`	an function as character string or `NA` (default). See `mctest`.

Details

The Kolmogorov-Smirnov test compares the null distribution with the empirical distribution of the observed data, where left truncated data samples are allowed. The test statistic (see `ks.test`) is given by $V = \max\{KS^+, KS^-\}$.

Value

A list with class "mchtest" containing the following components

`statistic`	the value of the Kuiper statistic
`treshold`	the treshold value
`p.value`	the p-value of the test
`data.name`	a character string giving the name of the data
`method`	the character string "Kuiper test"
`sim.no`	number of simulated szenarios in the Monte-Carlo simulation

References

Chernobay, A., Rachev, S., Fabozzi, F. (2005), *Composites goodness-of-fit tests for left-truncated loss samples*, Tech. rep., University of Calivornia Santa Barbara

See Also

ks.test, ad.test, adup.test for other supremum class tests and ad2.test, ad2up.test, w2.test for quadratic class tests. For more details see mctest.

Examples

```
set.seed(123)
treshold <- 10
xc  <- rlnorm(100, 2, 2)     # complete sample
xt <- xc[xc >= treshold]     # left truncated sample
v.test(xt, "plnorm", list(meanlog = 2, sdlog = 2), H = 10)
```

---

w2.test *Cram/'er-von Mises test*

Description

Cram/'er-von Mises test providing a comparison of a fitted distribution with the empirical distribution.

Usage

```
w2.test(x, distn, fit, H = NA, sim = 100, tol = 1e-04, estfun = NA)
```

Arguments

x	a numeric vector of data values
distn	character string naming the null distribution
fit	list of null distribution parameters
H	a treshold value
sim	maximum number of szenarios in the Monte-Carlo simulation
tol	if the difference of two subsequent p-value calculations is lower than tol the Monte-Carlo simulation is discontinued
estfun	an function as character string or NA (default). See mctest.

Details

The Cram/'er-von Mies test compares the null distribution with the empirical distribution function of the observed data, where left truncated data samples are allowed. The test statistic is given by

$$W^2 = \frac{n}{3} + \frac{n z_H}{1 - z_H} + \frac{1}{n(1 - z_H)} \sum_{j=1}^{n} (1 - 2j) z_j + \frac{1}{(1 - z_H)^2} \sum_{j=1}^{n} (z_j - z_H)^2$$

with $z_H = F_\theta(H)$ and $z_j = F_\theta(x_j)$, where $x_1, \ldots, x_n$ are the ordered data values. Here, F_θ is the null distribution.

Value

A list with class "mchtest" containing the following components

`statistic`	the value of the Cram\'er-von Mies statistic
`treshold`	the treshold value
`p.value`	the p-value of the test
`data.name`	a character string giving the name of the data
`method`	the character string "Cramer-von Mises test"
`sim.no`	number of simulated szenarios in the Monte-Carlo simulation

References

Chernobay, A., Rachev, S., Fabozzi, F. (2005), *Composites goodness-of-fit tests for left-truncated loss samples*, Tech. rep., University of Calivornia Santa Barbara

See Also

`ad2up.test`, `ad2.test` for other quadratic class tests and `ks.test`, `v.test`, `adup.test`, `ad.test` for supremum class tests. For more details see `mctest`.

Examples

```
set.seed(123)
treshold <- 10
xc  <- rlnorm(100, 2, 2)     # complete sample
xt <- xc[xc >= treshold]     # left truncated sample
w2.test(xt, "plnorm", list(meanlog = 2, sdlog = 2), H = 10)
```

Index