

Package: PBGoF (via r-universe)

October 13, 2026

Type Package

Title Goodness-of-Fit Tests for Skew-Normal and Skew-t Models

Version 0.2.0

Date 2026-10-11

Author Hongxiang Li [aut, cre], Chenglin Xu [aut], Tsung Fei Khang [aut]

Maintainer Hongxiang Li <hxli@ynnu.edu.cn>

Depends R (>= 4.1)

Description Provides precomputed-quantile goodness-of-fit tests for skew-normal distributions, parametric-bootstrap tests for skew-normal and skew-t distributions, robust fitting, and graphical diagnostics. Package methods and documentation are described by Li, Xu, and Khang (2026)
<<https://github.com/Divo-Lee/PBGoF>>.

Imports methods, sn

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL (>= 2)

Encoding UTF-8

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

NeedsCompilation no

Repository <https://cran.r-universe.dev>

Date/Publication 2026-10-12 13:31:07 UTC

RemoteUrl <https://github.com/cran/PBGoF>

RemoteRef HEAD

RemoteSha a30f2c68b52cb142e632329d08b460132703d982

Contents

cdf.plot.check	2
para.bootstrap.test	3
PBGoF-precomputed-test	4
QQ.plot	5
sn-parametric-bootstrap	6
sn.fit.robust	7
sn.plot.check	8
st-parametric-bootstrap	9
st.fit.robust	10
st.plot.check	10

Index	12
--------------	-----------

cdf.plot.check	<i>Graphical Check Using Empirical and Fitted Distribution Functions</i>
----------------	--

Description

Plots the empirical cumulative distribution function together with the CDF of a fitted skew-normal or skew-t distribution.

Usage

```
cdf.plot.check(data, model = "SN", curve_col = NULL, lwd = 2,
  main = NULL, xlab = "Values", ylab = "Distribution function",
  legend = FALSE, ...)
```

Arguments

data	A numeric vector containing finite observations.
model	Either "SN" or "ST".
curve_col	Color of the fitted CDF. The default is "#6CC6C6" for both SN and ST.
lwd	Line width of the fitted CDF. The ECDF uses lwd = 0.5.
main	Main plot title. The default is NULL.
xlab	Label for the horizontal axis.
ylab	Label for the vertical axis.
legend	Logical; if TRUE, add a legend for the ECDF and fitted CDF.
...	Additional graphical arguments passed to <code>graphics::plot()</code> .

Details

The fitted CDF is overlaid on the ECDF with the standard horizontal reference lines at $y = 0$ and $y = 1$.

Value

Invisibly returns a list containing the model, fitted parameters, grid values, fitted CDF values, and ECDF object.

Examples

```
set.seed(123)
x <- sn::rsn(200, xi = 0, omega = 1, alpha = 5)
cdf.plot.check(x, model = "SN")
```

para.bootstrap.test	<i>Unified parametric-bootstrap goodness-of-fit test</i>
---------------------	--

Description

Perform a parametric-bootstrap goodness-of-fit test for either a skew-normal or a skew-t distribution. The fitted model is re-estimated in every bootstrap sample.

Usage

```
para.bootstrap.test(
  data = NULL,
  model = c("SN", "ST"),
  statistic = c("KS", "CvM"),
  B = 1000L,
  seed = 103L,
  verbose = FALSE
)
```

Arguments

data	A numeric vector with at least 10 finite observations.
model	The fitted model, either "SN" or "ST".
statistic	The goodness-of-fit statistic: "KS" or "CvM".
B	A positive integer giving the number of bootstrap samples.
seed	A single integer used to initialize the bootstrap random-number stream, or NULL to use the current stream.
verbose	Logical; if TRUE, print a short summary.

Details

This function dispatches to the existing model- and statistic-specific implementations, so the estimation and bootstrap calibration are unchanged. The legacy functions `sn.param.bootstrap.ks.test()`, `sn.param.bootstrap.cvm.test()`, `st.param.bootstrap.ks.test()`, and `st.param.bootstrap.cvm.test()` remain available for compatibility.

Value

A numeric bootstrap p-value with attributes containing the observed statistic and bootstrap diagnostics. For the skew-t model, the fitted parameters are also returned in the `fit` attribute.

Examples

```
set.seed(123)
x_sn <- sn::rsn(100, alpha = 3)
para.bootstrap.test(x_sn, model = "SN", statistic = "KS", B = 100)

x_st <- sn::rst(100, alpha = 3, nu = 8)
para.bootstrap.test(x_st, model = "ST", statistic = "CvM", B = 100)
```

PBGof-precomputed-test

Skew-Normal Goodness-of-Fit Tests Using Precomputed Quantiles

Description

Computes a skew-normal goodness-of-fit statistic and obtains an approximate p-value from a pre-computed 100,000-replicate quantile table.

Usage

```
PBGof_ks_test(data, ks_table = NULL)

PBGof_cvm_test(data, cvm_table = NULL)
```

Arguments

<code>data</code>	A numeric vector whose length is at least the smallest sample size represented in the table.
<code>ks_table, cvm_table</code>	Optional custom quantile tables. If <code>NULL</code> , the corresponding table bundled with PBGoF is used.

Details

The fitted absolute CP skewness is rounded to two decimal places and truncated to $[0.01, 0.99]$. P-values are conservative step-function approximations between 0.01 and 0.99. No interpolation is performed.

If the fitted `gamma1` is negative, table lookup uses `abs(gamma1)`. Reflection of a skew-normal variable changes the signs of the DP shape parameter and CP skewness, but the null distributions of the EDF statistics are invariant to this sign change (Mateu-Figueras et al., 2007). The bundled tables therefore require only non-negative skewness values.

For samples larger than 500, all observations remain in the fit and empirical distribution function, while the $n = 500$ row is used for table lookup. The KS comparison uses $\sqrt{n}D_n$. For CvM, the stored $n = 500$ quantiles are divided by $\sqrt{500}$ and compared with W_n^2 .

Value

A list containing statistic, actual sample size n , γ_1 _hat, γ_1 _used, and p.value.

References

Mateu-Figueras, G., Puig, P., and Pewsey, A. (2007). Goodness-of-fit tests for the skew-normal distribution when the parameters are estimated from the data. *Communications in Statistics—Theory and Methods*, 36(9), 1735–1755. doi:[10.1080/03610920601126217](https://doi.org/10.1080/03610920601126217)

Examples

```
set.seed(123)
x <- sn::rsn(50, xi = 0, omega = 1, alpha = 3)
PBGof_ks_test(x)
PBGof_cvm_test(x)
```

QQ.plot	<i>QQ Plot for a Fitted Skew-Normal or Skew-t Distribution</i>
---------	--

Description

Draws a QQ plot for a fitted skew-normal or skew-t distribution.

Usage

```
QQ.plot(data, model = "SN", main = NULL,
        ylab = "Observed quantiles", pch = 16, cex = 0.65, lwd = 2)
```

Arguments

data	A numeric vector containing finite observations.
model	Either "SN" or "ST". Other models are not supported.
main	Main plot title. The default is NULL.
ylab	Label for the vertical axis.
pch	Plotting character for the observed quantiles.
cex	Point size.
lwd	Line width of the reference line.

Details

This function is intended only for checking skew-normal and skew-t models. The reference line uses color "#D55E00". The horizontal-axis label is automatically set to Theoretical SN quantiles or Theoretical ST quantiles.

Value

Invisibly returns a list containing the model, fitted parameters, theoretical quantiles, observed quantiles, and reference-line color.

Examples

```
set.seed(123)
x <- sn::rsn(200, xi = 0, omega = 1, alpha = 5)
QQ.plot(x, model = "SN")
```

sn-parametric-bootstrap

Parametric Bootstrap Goodness-of-Fit Tests for a Skew-Normal Distribution

Description

Tests skew-normal goodness of fit using a parametric bootstrap, re-estimating the model in each bootstrap sample with `sn.fit.robust()`.

Usage

```
sn.param.bootstrap.ks.test(data = NULL, B = 1000L, seed = 103L,
  verbose = FALSE)

sn.param.bootstrap.cvm.test(data = NULL, B = 1000L, seed = 103L,
  verbose = FALSE)
```

Arguments

<code>data</code>	A numeric vector containing at least 10 finite observations.
<code>B</code>	A positive integer giving the number of bootstrap samples.
<code>seed</code>	A single integer, or NULL to use the current random-number stream. The caller's random-number state is restored on exit.
<code>verbose</code>	Logical; whether to print a short summary.

Details

The KS function uses $\sqrt{n}D$; the CvM function uses the Cramér–von Mises statistic. Failed fits are excluded and the p-value uses the finite-simulation correction $(1 + \sum I(T_b \geq T_0)) / (B_{valid} + 1)$. The names ending in `.1` are compatibility aliases.

Value

A single numeric bootstrap p-value with attributes statistic, B, valid, and failed.

Examples

```
set.seed(123)
x <- sn::rsn(50, xi = 0, omega = 1, alpha = 3)
sn.para.bootstrap.ks.test(x, B = 99)
sn.para.bootstrap.cvm.test(x, B = 99)
```

sn.fit.robust

Numerically Robust Fitting of the Skew-Normal Distribution

Description

Fits a skew-normal distribution by MLE, then falls back to default penalized MLE and matching-prior penalized MLE when necessary.

Usage

```
sn.fit.robust(data = NULL, para_form = c("DP", "CP"))
```

Arguments

data	A numeric vector containing at least 10 finite observations.
para_form	Parameterization of the result: "DP" for direct parameters or "CP" for centered parameters.

Details

Robustness here concerns numerical fitting failures, not resistance to outliers or model contamination.

Value

A named numeric vector containing parameter estimates and standard errors. If all fitting procedures fail, all entries are NA.

Examples

```
set.seed(123)
x <- sn::rsn(100, xi = 0, omega = 1, alpha = 5)
sn.fit.robust(x, "DP")
sn.fit.robust(x, "CP")
```

sn.plot.check

*Graphical Check of a Fitted Skew-Normal Distribution***Description**

Draws a density-scale histogram of the observed data and overlays a skew-normal density fitted by `sn.fit.robust()`.

Usage

```
sn.plot.check(
  data,
  breaks = "FD",
  col = "grey83",
  border = "grey83",
  curve_col = "#6CC6C6",
  lwd = 2,
  main = NULL,
  xlab = "Values",
  add_rug = TRUE,
  ...
)
```

Arguments

<code>data</code>	A numeric vector containing at least 10 finite observations.
<code>breaks</code>	Histogram breaks passed to <code>graphics::hist()</code> .
<code>col</code>	Fill color for the histogram.
<code>border</code>	Border color for histogram bars.
<code>curve_col</code>	Color of the fitted skew-normal density curve.
<code>lwd</code>	Line width of the fitted density curve.
<code>main</code>	Main plot title.
<code>xlab</code>	Label for the horizontal axis.
<code>add_rug</code>	Logical; whether to add observations as a rug plot.
<code>...</code>	Additional graphical arguments passed to <code>graphics::hist()</code> .

Details

This visual model check complements, but does not replace, the formal goodness-of-fit tests provided by `PBGoF`.

Value

Invisibly returns a list containing the fitted DP parameters, histogram object, and coordinates of the fitted density curve.

Examples

```
set.seed(123)
x <- sn::rsn(100, xi = 0, omega = 1, alpha = 4)
sn.plot.check(x)
```

st-parametric-bootstrap

Parametric Bootstrap Goodness-of-Fit Tests for a Skew-t Distribution

Description

Tests a fitted univariate skew-t distribution using KS or Cramer-von Mises statistics. All four skew-t parameters are re-estimated in each bootstrap sample with `st.fit.robust()`.

Usage

```
st.param.bootstrap.ks.test(data = NULL, B = 1000L, seed = 103L, verbose = FALSE)
```

```
st.param.bootstrap.cvm.test(data = NULL, B = 1000L, seed = 103L, verbose = FALSE)
```

Arguments

data	A numeric vector with at least 10 finite observations.
B	Positive integer number of bootstrap samples.
seed	Non-negative integer or NULL.
verbose	Logical; print a short summary.

Details

The KS statistic is $\sqrt{n} D$. The CvM statistic is $1/(12n) + \sum_i [U_i - (2i - 1)/(2n)]^2$. Failed fits are excluded, and the finite-simulation correction is used.

Value

A numeric bootstrap p-value with attributes `statistic`, `B`, `valid`, `failed`, and `fit`.

Examples

```
set.seed(123)
x <- sn::rst(100, alpha=3, nu=8)
st.param.bootstrap.ks.test(x, B=100)
st.param.bootstrap.cvm.test(x, B=100)
```

st.fit.robust

Numerically Robust Fitting of a Univariate Skew-t Distribution

Description

Fits a univariate skew-t distribution using ordinary maximum likelihood, followed by Q-penalized maximum likelihood with alternative optimizers when needed.

Usage

```
st.fit.robust(data = NULL, para_form = c("DP", "CP"))
```

Arguments

data A numeric vector with at least 10 finite observations.
para_form Parameterization to return: "DP" or "CP".

Details

DP contains xi, omega, alpha, and nu. CP contains the mean, standard deviation, skewness, and kurtosis when these moments exist. Robust refers to numerical fallback, not resistance to outliers.

Value

A named numeric vector containing four estimates and their standard errors.

Examples

```
set.seed(123)
x <- sn::rst(100, alpha=3, nu=8)
st.fit.robust(x, "DP")
```

st.plot.check

Graphical Check of a Fitted Skew-t Distribution

Description

Draws a density-scale histogram and overlays the skew-t density fitted by st.fit.robust().

Usage

```
st.plot.check(data, breaks = "FD", col = "grey83", border = "grey83",
  curve_col = "#6CC6C6", lwd = 2, main = NULL, xlab = "Values",
  add_rug = TRUE, ...)
```

Arguments

<code>data</code>	A numeric vector with at least 10 finite observations.
<code>breaks</code>	Histogram breaks passed to <code>hist</code> .
<code>col</code>	Histogram fill color.
<code>border</code>	Histogram border color.
<code>curve_col</code>	Fitted density color. The default is "#6CC6C6".
<code>lwd</code>	Fitted density line width.
<code>main</code>	Main title.
<code>xlab</code>	Horizontal-axis label.
<code>add_rug</code>	Logical; add observed values as a rug plot.
<code>...</code>	Additional graphical arguments passed to <code>hist</code> .

Value

Invisibly returns fitted DP parameters, histogram information, and fitted curve coordinates.

Examples

```
set.seed(123)
x <- sn::rst(100, alpha=3, nu=8)
st.plot.check(x)
```

Index

`cdf.plot.check`, [2](#)

`para.bootstrap.test`, [3](#)

`PBGoF-precomputed-test`, [4](#)

`PBGoF_cvm_test`
 (`PBGoF-precomputed-test`), [4](#)

`PBGoF_ks_test` (`PBGoF-precomputed-test`),
 [4](#)

`QQ.plot`, [5](#)

`sn-parametric-bootstrap`, [6](#)

`sn.fit.robust`, [7](#)

`sn.para.bootstrap.cvm.test`
 (`sn-parametric-bootstrap`), [6](#)

`sn.para.bootstrap.ks.test`
 (`sn-parametric-bootstrap`), [6](#)

`sn.plot.check`, [8](#)

`st-parametric-bootstrap`, [9](#)

`st.fit.robust`, [10](#)

`st.para.bootstrap.cvm.test`
 (`st-parametric-bootstrap`), [9](#)

`st.para.bootstrap.ks.test`
 (`st-parametric-bootstrap`), [9](#)

`st.plot.check`, [10](#)