Package 'simukde'

October 14, 2022

Type Package

Version 1.3.0

Date 2021-05-20

Title Simulation with Kernel Density Estimation

Description
Generates random values from a univariate and multivariate continuous distribution by using kernel density estimation based on a sample. Duong (2017) <doi:10.18637/jss.v021.i07>, Christian P. Robert and George Casella (2010 ISBN:978-1-4419-1575-7) <doi:10.1007/978-1-4419-1576-4>.

Author MAKHGAL Ganbold [aut, cre],
BAYARBAATAR Amgalan [aut]

Maintainer MAKHGAL Ganbold <makhgal@seas.num.edu.mn>

URL https://github.com/galaamn/simukde

BugReports https://github.com/galaamn/simukde/issues

Depends R (>= 2.14.0)

Imports ks, mvtnorm, parallel, stats, MASS, methods

Suggests testthat, datasets

License GPL (>= 3) | file LICENSE

Encoding UTF-8

ByteCompile true

NeedsCompilation no

RoxygenNote 7.1.1

Repository CRAN

Date/Publication 2021-05-20 11:00:02 UTC

R topics documented:

`find_best_fit` *Find The Best Fitting Distribution*

Description

It finds the best fitting distribution from supported univariate continuous distributions for given data.

Usage

```
find_best_fit(
  x,
  positive = FALSE,
  plot = TRUE,
  legend.pos = "topright",
  dlc = NULL,
  dlw = 1,
  ...
)
```

Arguments

`x`	a numeric vector; data.
`positive`	a logical constant; distribution type.
`plot`	a logical constant. If `TRUE` (default), a histogram and density lines are drawn.
`legend.pos`	a character string. Indicates the legend position and must be one of "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright" (default), "right" and "center".
`dlc`	a vector; probability density line colors for supported (up to 7) distributions. If unspecified, the rainbow color palette will be used.
`dlw`	a numerical constant; probability density line width.
`...`	Further arguments and parameters for the function `hist`, particularly, main title and axis labels. However, the parameter `freq` is not able to override.

Details

This function is supported following univariate distributions:

- for positive random variables: Log normal, Exponential, Gamma and Weibull.
- for all random variables: Normal, Cauchy, Log normal, Exponential, Gamma, Weibull and Uniform.

Legends of the plot are ordered by p-values of the test.

Value

A list containing the following items:

distribution the name of the best fitting distribution.

ks.statistic the Kolmogorov-Smirnov test statistic for the distribution.

p.value the p-value of the test.

summary results similar to above for other distributions.

x given data.

n the sample size.

References

1. William J. Conover (1971). Practical Nonparametric Statistics. New York: John Wiley & Sons. Pages 295–301.
2. Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

See Also

`ks.test`, `fitdistr`, `hist`

Examples

```
petal.length <- datasets::iris$Petal.Length[datasets::iris$Species == "setosa"]
simukde::find_best_fit(x = petal.length, positive = TRUE)
```

`simukde` *Simulation with Kernel Density Estimation*

Description

The simukde package provides a function which generates random values from a univariate and multivariate continuous distribution by using kernel density estimation based on a sample. The function uses the Accept-Reject method.

Note

Funding: This package has been done within the framework of the project Statistics and Optimization Based Methods for Identification of Cancer-Activated Biological Processes (P2017-2519) supported by the Asia Research Center, Mongolia and Korea Foundation for Advanced Studies, Korea.

The funders had no role in study design, analysis, decision to publish, or preparation of the package.

Author(s)

MAKHGAL Ganbold and BAYARBAATAR Amgalan, National University of Mongolia

References

Duong (2017) <doi:10.18637/jss.v021.i07>, Christian P. Robert and George Casella (2010 ISBN:978-1-4419-1575-7) <doi:10.1007/978-1-4419-1576-4>.

`simulate_kde` *Simulation with Kernel Density Estimation*

Description

Generates random values from a univariate and multivariate continuous distribution by using kernel density estimation based on a sample. The function uses the Accept-Reject method.

Usage

```
simulate_kde(
  x,
  n = 100,
  distr = "norm",
  const.only = FALSE,
  seed = NULL,
  parallel = FALSE,
  ...
)
```

Arguments

`x`	a numeric vector, matrix or data frame; data.
`n`	integer; the number of random values will be generated.
`distr`	character; instrumental or candidate distribution name. See details.
`const.only`	logical; if `TRUE`, the constant of the Accept-Reject method will be returned.
`seed`	a single value, interpreted as an integer, or `NULL` (default).
`parallel`	logical; if `TRUE` parallel generator will be worked. `FALSE` is default.
`...`	other parameters for functions `kde`.

Details

Such function uses the function `kde` as kernel density estimator.

The Accept-Reject method is used to simulate random variables. Following code named distributions can be used as a value of the argument `distr` and an instrumental or candidate distribution of the simulation method. For univariate distributions:

norm normal distribution (default), $(-\infty, +\infty)$

cauchy Cauchy distribution, $(-\infty, +\infty)$

lnorm log-normal distribution, $(0, +\infty)$

exp exponential distribution, $(0, +\infty)$

gamma gamma distribution, $(0, +\infty)$

weibull Weibull distribution, $(0, +\infty)$

unif uniform distribution, (a, b)

And you can choose the best fitting instrumental distribution to simulate random variables more effectively by using `find_best_fit`. See examples.

For multivariate distributions, "norm" (multivariate normal distribution) is used.

Value

list of given data, simulated values, kernel density estimation and the constant of the Accept-Reject method when `const.only` is `FALSE` (default).

References

- Tarn Duong (2018). ks: Kernel Smoothing. R package version 1.11.2. https://CRAN.R-project.org/package=ks
- Christian P. Robert and George Casella (2010) Introducing Monte Carlo Methods with R. Springer. Pages 51-57.

See Also

`find_best_fit`, `kde`

Examples

```
## 1-dimensional data
data(faithful)
hist(faithful$eruptions)
res <- simukde::simulate_kde(x = faithful$eruptions, n = 100, parallel = FALSE)
hist(res$random.values)

## Simulation with the best fitting instrumental distribution
data(faithful)
par(mfrow = c(1, 3))
hist(faithful$eruptions)
fit <- simukde::find_best_fit(x = faithful$eruptions, positive = TRUE)
res <- simukde::simulate_kde(
  x = faithful$eruptions, n = 100,
  distr = fit$distribution, parallel = FALSE
)
hist(res$random.values)
par(mfrow = c(1, 1))

## 2-dimensional data
data(faithful)
res <- simukde::simulate_kde(x = faithful, n = 100)
plot(res$kde, display = "filled.contour")
points(x = res$random.values, cex = 0.25, pch = 16, col = "green")
```

```
points(x = faithful, cex = 0.25, pch = 16, col = "black")
```

Index