Package 'discretefit'

December 18, 2025

Title Simulated Goodness-of-Fit Tests for Discrete Distributions

Version 0.1.3

Description Implements fast Monte Carlo simulations for goodness-of-fit (GOF) tests for discrete distributions. This includes tests based on the Chi-squared statistic, the log-likelihood-ratio (G^2) statistic, the Freeman-Tukey (Hellinger-distance) statistic, the Kolmogorov-Smirnov statistic, the Cramer-von Mises statistic as described in Choulakian, Lockhart and Stephens (1994) <doi:10.2307/3315828>, and the root-mean-square statistic, see Perkins, Tygert, and Ward (2011) <doi:10.1016/j.amc.2011.03.124>.

License MIT + file LICENSE

URL https://github.com/josh-mc/discretefit

BugReports https://github.com/josh-mc/discretefit/issues

Encoding UTF-8

RoxygenNote 7.3.2

LinkingTo Rcpp

Imports Rcpp

Suggests knitr, dgof, cvmdisc, bench, testthat (>= 3.0.0), rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Author Josh McCormick [aut, cre]

Maintainer Josh McCormick <josh.mccormick@aya.yale.edu>

Repository CRAN

Date/Publication 2025-12-18 06:10:58 UTC

Contents

---

chisq_gof	*Simulated Chi-squared goodness-of-fit test*

Description

The `chisq_gof()` function implements Monte Carlo simulations to calculate p-values based on the Chi-squared statistic for goodness-of-fit tests for discrete distributions.

Usage

```
chisq_gof(x, p, reps = 10000, tolerance = 64 * .Machine$double.eps)
```

Arguments

x	a numeric vector that contains observed counts for each bin/category.
p	a vector of probabilities of the same length of x. An error is given if any entry of p is negative or if the sum of p does not equal one.
reps	an integer specifying the number of Monte Carlo simulations. The default is set to 10,000 which may be appropriate for exploratory analysis. A higher number of simulation should be selected for more precise results.
tolerance	sets an upper bound for rounding errors when evaluating whether a statistic for a simulation is greater than or equal to the statistic for the observed data. The default is identical to the tolerance set for simulations in the `chisq.test` function from the `stats` package in base R.

Value

A list with class "htest" containing the following components:

statistic	the value of the Chi-squared test statistic
p.value	the simulated p-value for the test
method	a character string describing the test
data.name	a character string give the name of the data

Examples

```
x <- c(15, 36, 17)
p <- c(0.25, 0.5, 0.25)

chisq_gof(x, p)
```

---

cvm_gof   *Simulated Cramer-von Mises goodness-of-fit test*

Description

The `cvm_gof()` function implements Monte Carlo simulations to calculate p-values based on the Cramer-von Mises statistic (W^2) for goodness-of-fit tests for discrete distributions.

Usage

```
cvm_gof(x, p, reps = 10000, tolerance = 64 * .Machine$double.eps)
```

Arguments

x	a numeric vector that contains observed counts for each bin/category.
p	a vector of probabilities of the same length of x. An error is given if any entry of p is negative or if the sum of p does not equal one.
reps	an integer specifying the number of Monte Carlo simulations. The default is set to 10,000 which may be appropriate for exploratory analysis. A higher number of simulation should be selected for more precise results.
tolerance	sets an upper bound for rounding errors when evaluating whether a statistic for a simulation is greater than or equal to the statistic for the observed data. The default is identical to the tolerance set for simulations in the `chisq.test` function from the `stats` package in base R.

Value

A list with class "htest" containing the following components:

statistic	the value of the Cramer-von Mises test statistic (W2)
p.value	the simulated p-value for the test
method	a character string describing the test
data.name	a character string give the name of the data

Examples

```
x <- c(15, 36, 17)
p <- c(0.25, 0.5, 0.25)

cvm_gof(x, p)
```

---

ft_gof   *Simulated Freeman-Tukey (Hellinger-distance) goodness-of-fit test*

Description

The `ft_gof()` function implements Monte Carlo simulations to calculate p-values based on the Freeman-Tukey statistic for goodness-of-fit tests for discrete distributions. This statistic is also referred to as the Hellinger-distance. Asymptotically, the Freeman-Tukey GOF test is identical to the Chi-squared GOF test, but for smaller n, results may vary significantly.

Usage

```
ft_gof(x, p, reps = 10000, tolerance = 64 * .Machine$double.eps)
```

Arguments

x   a numeric vector that contains observed counts for each bin/category.

p   a vector of probabilities of the same length of x. An error is given if any entry
                of p is negative or if the sum of p does not equal one.

reps   an integer specifying the number of Monte Carlo simulations. The default is set
                to 10,000 which may be appropriate for exploratory analysis. A higher number
                of simulation should be selected for more precise results.

tolerance   sets an upper bound for rounding errors when evaluating whether a statistic for a
                simulation is greater than or equal to the statistic for the observed data. The de-
                fault is identical to the tolerance set for simulations in the `chisq.test` function
                from the `stats` package in base R.

Value

A list with class "htest" containing the following components:

statistic   the value of the Freeman-Tukey test statistic (W2)

p.value   the simulated p-value for the test

method   a character string describing the test

data.name   a character string give the name of the data

Examples

```
x <- c(15, 36, 17)
p <- c(0.25, 0.5, 0.25)

ft_gof(x, p)
```

---

g_gof   *Simulated log-likelihood-ratio (G^2) goodness-of-fit test*

Description

The g_gof() function implements Monte Carlo simulations to calculate p-values based on the log-likelihood-ratio statistic for goodness-of-fit tests for discrete distributions. In this context, the log-likelihood-ratio statistic is often referred to as the G^2 statistic. Asymptotically, the G^2 GOF test is identical to the Chi-squared GOF test, but for smaller n, results may vary significantly.

Usage

```
g_gof(x, p, reps = 10000, tolerance = 64 * .Machine$double.eps)
```

Arguments

x	a numeric vector that contains observed counts for each bin/category.
p	a vector of probabilities of the same length of x. An error is given if any entry of p is negative or if the sum of p does not equal one.
reps	an integer specifying the number of Monte Carlo simulations. The default is set to 10,000 which may be appropriate for exploratory analysis. A higher number of simulation should be selected for more precise results.
tolerance	sets an upper bound for rounding errors when evaluating whether a statistic for a simulation is greater than or equal to the statistic for the observed data. The default is identical to the tolerance set for simulations in the chisq.test function from the stats package in base R.

Value

A list with class "htest" containing the following components:

statistic	the value of the log-likelihood-ratio test statistic (G2)
p.value	the simulated p-value for the test
method	a character string describing the test
data.name	a character string give the name of the data

Examples

```
x <- c(15, 36, 17)
p <- c(0.25, 0.5, 0.25)

g_gof(x, p)
```

---

ks_gof   *Simulated Kolmogorov-Smirnov goodness-of-fit test*

Description

The `ks_gof()` function implements Monte Carlo simulations to calculate p-values based on the Kolmogorov-Smirnov statistic for goodness-of-fit tests for discrete distributions. The p-value expressed by `ks_gof()` is based on a two-sided alternative hypothesis.

Usage

```
ks_gof(x, p, reps = 10000, tolerance = 64 * .Machine$double.eps)
```

Arguments

x	a numeric vector that contains observed counts for each bin/category.
p	a vector of probabilities of the same length of x. An error is given if any entry of p is negative or if the sum of p does not equal one.
reps	an integer specifying the number of Monte Carlo simulations. The default is set to 10,000 which may be appropriate for exploratory analysis. A higher number of simulation should be selected for more precise results.
tolerance	sets an upper bound for rounding errors when evaluating whether a statistic for a simulation is greater than or equal to the statistic for the observed data. The default is identical to the tolerance set for simulations in the `chisq.test` function from the `stats` package in base R.

Value

A list with class "htest" containing the following components:

statistic	the value of the Kolmogorov-Smirnov test statistic
p.value	the simulated p-value for the test
method	a character string describing the test
data.name	a character string give the name of the data

Examples

```
x <- c(15, 36, 17)
p <- c(0.25, 0.5, 0.25)

ks_gof(x, p)
```

---

rms_gof   *Simulated root-mean-square goodness-of-fit test*

Description

The `rms_gof()` function implements Monte Carlo simulations to calculate p-values based on the root-mean-square statistic for goodness-of-fit tests for discrete distributions.

Usage

```
rms_gof(x, p, reps = 10000, tolerance = 64 * .Machine$double.eps)
```

Arguments

x   a numeric vector that contains observed counts for each bin/category.

p   a vector of probabilities of the same length of x. An error is given if any entry
             of p is negative or if the sum of p does not equal one.

reps   an integer specifying the number of Monte Carlo simulations. The default is set
             to 10,000 which may be appropriate for exploratory analysis. A higher number
             of simulation should be selected for more precise results.

tolerance   sets an upper bound for rounding errors when evaluating whether a statistic for a
             simulation is greater than or equal to the statistic for the observed data. The de-
             fault is identical to the tolerance set for simulations in the `chisq.test` function
             from the `stats` package in base R.

Value

A list with class "htest" containing the following components:

statistic   the value of the root-mean-square test statistic

p.value   the simulated p-value for the test

method   a character string describing the test

data.name   a character string give the name of the data

Examples

```
x <- c(15, 36, 17)
p <- c(0.25, 0.5, 0.25)

rms_gof(x, p)
```

Index