

# Package ‘discretefit’

January 26, 2022

**Title** Simulated Goodness-of-Fit Tests for Discrete Distributions

**Version** 0.1.2

**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** 7.1.1

**LinkingTo** Rcpp

**Imports** Rcpp

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

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**SystemRequirements** C++11

**NeedsCompilation** yes

**Author** Josh McCormick [aut, cre]

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

**Repository** CRAN

**Date/Publication** 2022-01-25 23:52:50 UTC

**R topics documented:**

chisq_gof . . . . .	2
cvm_gof . . . . .	3
ft_gof . . . . .	4
g_gof . . . . .	5
ks_gof . . . . .	6
rms_gof . . . . .	7
<b>Index</b>	<b>8</b>

---

chisq_gof	<i>Simulated Chi-squared goodness-of-fit test</i>
-----------	---

---

**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**

<code>x</code>	a numeric vector that contains observed counts for each bin/category.
<code>p</code>	a vector of probabilities of the same length of <code>x</code> . An error is given if any entry of <code>p</code> is negative or if the sum of <code>p</code> does not equal one.
<code>reps</code>	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.
<code>tolerance</code>	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 <code>chisq.test</code> function from the <code>stats</code> package in base R.

**Value**

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

<code>statistic</code>	the value of the Chi-squared test statistic
<code>p.value</code>	the simulated p-value for the test
<code>method</code>	a character string describing the test
<code>data.name</code>	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**

<code>x</code>	a numeric vector that contains observed counts for each bin/category.
<code>p</code>	a vector of probabilities of the same length of <code>x</code> . An error is given if any entry of <code>p</code> is negative or if the sum of <code>p</code> does not equal one.
<code>reps</code>	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.
<code>tolerance</code>	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 <code>chisq.test</code> function from the <code>stats</code> package in base R.

**Value**

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

<code>statistic</code>	the value of the Cramer-von Mises test statistic ( $W^2$ )
<code>p.value</code>	the simulated p-value for the test
<code>method</code>	a character string describing the test
<code>data.name</code>	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**

<code>x</code>	a numeric vector that contains observed counts for each bin/category.
<code>p</code>	a vector of probabilities of the same length of <code>x</code> . An error is given if any entry of <code>p</code> is negative or if the sum of <code>p</code> does not equal one.
<code>reps</code>	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.
<code>tolerance</code>	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 <code>chisq.test</code> function from the <code>stats</code> package in base R.

**Value**

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

<code>statistic</code>	the value of the Freeman-Tukey test statistic ( $W^2$ )
<code>p.value</code>	the simulated p-value for the test
<code>method</code>	a character string describing the test
<code>data.name</code>	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**

<code>x</code>	a numeric vector that contains observed counts for each bin/category.
<code>p</code>	a vector of probabilities of the same length of <code>x</code> . An error is given if any entry of <code>p</code> is negative or if the sum of <code>p</code> does not equal one.
<code>reps</code>	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.
<code>tolerance</code>	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 <code>chisq.test</code> function from the <code>stats</code> package in base R.

**Value**

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

<code>statistic</code>	the value of the log-likelihood-ratio test statistic ( $G^2$ )
<code>p.value</code>	the simulated p-value for the test
<code>method</code>	a character string describing the test
<code>data.name</code>	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**

<code>x</code>	a numeric vector that contains observed counts for each bin/category.
<code>p</code>	a vector of probabilities of the same length of <code>x</code> . An error is given if any entry of <code>p</code> is negative or if the sum of <code>p</code> does not equal one.
<code>reps</code>	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.
<code>tolerance</code>	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 <code>chisq.test</code> function from the <code>stats</code> package in base R.

**Value**

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

<code>statistic</code>	the value of the Kolmogorov-Smirnov test statistic
<code>p.value</code>	the simulated p-value for the test
<code>method</code>	a character string describing the test
<code>data.name</code>	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**

<code>x</code>	a numeric vector that contains observed counts for each bin/category.
<code>p</code>	a vector of probabilities of the same length of <code>x</code> . An error is given if any entry of <code>p</code> is negative or if the sum of <code>p</code> does not equal one.
<code>reps</code>	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.
<code>tolerance</code>	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 <code>chi_sq.test</code> function from the <code>stats</code> package in base R.

**Value**

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

<code>statistic</code>	the value of the root-mean-square test statistic
<code>p.value</code>	the simulated p-value for the test
<code>method</code>	a character string describing the test
<code>data.name</code>	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

chisq\_gof, 2

cvm\_gof, 3

ft\_gof, 4

g\_gof, 5

ks\_gof, 6

rms\_gof, 7