Package ‘Upsilon’

January 6, 2026

Type Package

Title Another Test of Association for Count Data

Version 0.1.0

Date 2025-12-20

Description The Upsilon test assesses association among categorical variables against the null hypothesis of independence (Luo 2021 MS thesis; ProQuest Publication No. 28649813). While promoting dominant function patterns, it demotes non-dominant function patterns. It is robust to low expected count---continuity correction like Yates's seems unnecessary. Using a common null population following a uniform distribution, contingency tables are comparable by statistical significance---not the case for most association tests defining a varying null population by tensor product of observed marginals. Although Pearson's chi-squared test, Fisher's exact test, and Woolf's G-test (related to mutual information) are useful in some contexts, the Upsilon test appeals to ranking association patterns not necessarily following same marginal distributions, such as in count data from DNA sequencing---an important modern scientific domain.

Encoding UTF-8

License LGPL (>= 3)

Imports Rcpp (>= 1.0.8), Rdpack, ggplot2 (>= 3.4.0), reshape2, scales

RdMacros Rdpack

LinkingTo Rcpp

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0), DescTools, USP, metan, FunChisq, patchwork

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Xuye Luo [aut],
Joe Song [aut, cre] (ORCID: <https://orcid.org/0000-0002-6883-6547>)

Maintainer Joe Song <joemsong@nmsu.edu>

Repository CRAN

Date/Publication 2026-01-06 11:40:07 UTC

Contents

`fast.chisq.test` *Fast Zero-Tolerant Pearson's Chi-squared Test of Association*

Description

Performs a fast zero-tolerant Pearson's chi-squared test (Pearson 1900) to evaluate association between observations from two categorical variables.

Usage

```
fast.chisq.test(x, y, log.p = FALSE)
```

Arguments

`x`	a vector to specify observations of the first categorical variable. The vector can be of numeric, character, or logical type. `NA` values must be removed or replaced before calling the function.
`y`	a vector to specify observations of the second categorical variable. Must not contain `NA` values and must be of the same length as `x`.
`log.p`	a logical. If `TRUE`, the *p*-value is calculated in closed form to **natural logarithm** of *p*-value to improve numerical precision when *p*-value approaches zero. Defaults to `FALSE`.

Value

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

`statistic`	the value of chi-squared test statistic.
`parameter`	the degrees of freedom.
`p.value`	the p-value of the test.
`estimate`	Cramér's V statistic representing the effect size.
`method`	a character string indicating the method used.
`data.name`	a character string giving the names of input data.

Note

The test uses an internal hash table, instead of matrix, to store the contingency table. Savings in both runtime and memory saving can be substantial if the contingency table is sparse and large. The test is implemented in C++, to give an additional layer of speedup over an R implementation.

References

Pearson K (1900). "X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **50**(302), 157–175. doi:10.1080/14786440009463897.

Examples

```
library("Upsilon")
weather <- c(
  "rainy", "sunny", "rainy", "sunny", "rainy"
)
mood <- c(
  "wistful", "upbeat", "upbeat", "upbeat", "wistful"
)

fast.chisq.test(weather, mood)

# The result is equivalent to:
modified.chisq.test(table(weather, mood))
```

`fast.gtest` *Fast Zero-Tolerant G-Test of Association*

Description

Performs a fast zero-tolerant G-test (Woolf 1957) to evaluate association between observations from two categorical variables.

Usage

```
fast.gtest(x, y, log.p = FALSE)
```

Arguments

`x`	a vector to specify observations of the first categorical variable. The vector can be of numeric, character, or logical type. `NA` values must be removed or replaced before calling the function.
`y`	a vector to specify observations of the second categorical variable. Must not contain `NA` values and must be of the same length as `x`.
`log.p`	a logical. If `TRUE`, the p-value is calculated in closed form to **natural logarithm** of p-value to improve numerical precision when p-value approaches zero. Defaults to `FALSE`.

Value

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

`statistic`	the G-test statistic (Likelihood Ratio Chi-squared statistic).
`parameter`	the degrees of freedom.
`p.value`	the p-value of the test.
`estimate`	the mutual information between the two variables.
`method`	a character string indicating the method used.
`data.name`	a character string giving the names of the data.

Note

The test uses an internal hash table, instead of matrix, to store the contingency table. Savings in both runtime and memory saving can be substantial if the contingency table is sparse and large. The test is implemented in C++, to give an additional layer of speedup over an R implementation.

References

Woolf B (1957). “The log likelihood ratio test (the G-test); methods and tables for tests of heterogeneity in contingency tables.” *Annals of Human Genetics*, **21**(4), 397–409. doi:10.1111/j.1469-1809.1972.tb00293.x.

Examples

```
library("Upsilon")
weather <- c(
  "rainy", "sunny", "rainy", "sunny", "rainy"
)
mood <- c(
  "wistful", "upbeat", "upbeat", "upbeat", "wistful"
)

fast.gtest(weather, mood)
```

```
# The result is equivalent to:
modified.gtest(table(weather, mood))
```

`fast.upsilon.test` *Fast Upsilon Test of Association between Two Categorical Variables*

Description

Performs a fast Upsilon test (Luo 2021) to evaluate association between observations from two categorical variables.

Usage

```
fast.upsilon.test(x, y, log.p = FALSE)
```

Arguments

`x`	a vector to specify observations of the first categorical variable. The vector can be of numeric, character, or logical type. `NA` values must be removed or replaced before calling the function.
`y`	a vector to specify observations of the second categorical variable. Must not contain `NA` values and must be of the same length as `x`.
`log.p`	a logical. If `TRUE`, the *p*-value is calculated in closed form to **natural logarithm** of *p*-value to improve numerical precision when *p*-value approaches zero. Defaults to `FALSE`.

Details

The Upsilon test is designed to promote dominant function patterns. In contrast to other tests of association to favor all function patterns, it is unique in demoting non-dominant function patterns.

Null hypothesis (H_0): Row and column variables are statistically independent.

Null population: A discrete uniform distribution, where each entry in the table has the same probability.

Null distribution: The Upsilon test statistic asymptotically follows a chi-squared distribution with `(nrow(x) - 1)(ncol(x) - 1)` degrees of freedom, under the null hypothesis on the null population.

See (Luo 2021) for full details of the Upsilon test.

Value

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

`statistic`	the Upsilon test statistic.
`parameter`	the degrees of freedom.
`p.value`	the *p*-value of the test.
`estimate`	the effect size derived from the Upsilon statistic.
`method`	a character string indicating the method used.
`data.name`	a character string giving the name of input data.

Note

The test uses an internal hash table, instead of matrix, to store the contingency table. Savings in both runtime and memory saving can be substantial if the contingency table is sparse and large. The test is implemented in C++, to give an additional layer of speedup over an R implementation.

References

Luo X (2021). *The Upsilon Test for Association Between Categorical Variables*. Master's thesis, Department of Computer Science, New Mexico State University, Las Cruces, NM, United States. Publication No. 28649813. ProQuest Dissertations and Theses Global.

Examples

```
library("Upsilon")

weather <- c(
  "rainy", "sunny", "rainy", "sunny", "rainy"
)
mood <- c(
  "wistful", "upbeat", "upbeat", "upbeat", "wistful"
)

fast.upsilon.test(weather, mood)

# The result is equivalent to:
upsilon.test(table(weather, mood))
```

`modified.chisq.test` *Zero-Tolerant Pearson's Chi-squared Test for Contingency Tables*

Description

Performs Pearson's chi-squared test (Pearson 1900) on contingency tables, slightly modified to handle rows or columns of all zeros.

Usage

```
modified.chisq.test(x, log.p = FALSE)
```

Arguments

`x`	a matrix or data frame of floating or integer numbers to specify a contingency table. Entries must be non-negative.
`log.p`	a logical. If `TRUE`, the *p*-value is calculated in closed form to **natural logarithm** of *p*-value to improve numerical precision when *p*-value approaches zero. Defaults to `FALSE`.

Details

This test is useful if p-value must be returned on a contingency table with valid non-negative counts, where the build-in R implementation of `chisq.test` could return `NA` as p-value, regardless of a pattern being strong or weak. See Examples.

Unlike `chisq.test`, this function handles tables with empty rows or columns (where expected values are 0) by calculating the test statistic over non-zero entries only. This prevents the result from becoming `NA`, while giving meaningful p-values.

Value

A list with class `"htest"` containing:

`statistic`	the chi-squared test statistic (calculated ignoring entries of 0-expected count).
`parameter`	the degrees of freedom.
`p.value`	the p-value by the test.
`estimate`	Cramér's V statistic.
`observed`	the observed counts.
`expected`	the expected counts under the null hypothesis.

Note

This function only takes contingency table as input. It does not support goodness-of-fit test on vectors. It does **not** offer an option to apply Yates's continuity correction on 2×2 tables.

References

Pearson K (1900). "X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **50**(302), 157–175. doi:10.1080/14786440009463897.

Examples

```
library("Upsilon")

# A table with a dominant function and an empty column
x <- matrix(
  c(0, 3, 0,
    3, 0, 0),
   nrow = 2, byrow = TRUE)
print(x)

# Standard chisq.test fails or returns NA warning
chisq.test(x)

# Modified chi-squared test is significant:
modified.chisq.test(x)
```

`modified.gtest` *Zero-Tolerant G-Test for Contingency Tables*

Description

Performs *G*-test (Woolf 1957) on contingency tables, slightly modified to handle rows or columns of all zeros.

Usage

```
modified.gtest(x, log.p = FALSE)
```

Arguments

`x`	a matrix or data frame of floating or integer numbers to specify a contingency table. Entries must be non-negative.
`log.p`	a logical. If `TRUE`, the *p*-value is calculated in closed form to **natural logarithm** of *p*-value to improve numerical precision when *p*-value approaches zero. Defaults to `FALSE`.

Details

This test is useful if a *p*-value must be returned on a contingency table with valid non-negative counts, where other implementations of *G*-test could return `NA` as the *p*-value, regardless of a pattern being strong or weak.

This function handles tables with empty rows or columns (where expected values are 0) by calculating the test statistic over non-zero entries only. This prevents the result from becoming `NA`, while giving meaningful *p*-values.

Value

A list with class `"htest"` containing:

`statistic`	the *G* statistic (log-likelihood ratio).
`parameter`	the degrees of freedom.
`p.value`	the *p*-value of the test.
`estimate`	the value of mutual information.
`method`	a character string indicating the method used.
`data.name`	a character string, name of the input data.
`observed`	the observed counts.
`expected`	the expected counts under the null hypothesis.

References

Woolf B (1957). "The log likelihood ratio test (the G-test); methods and tables for tests of heterogeneity in contingency tables." *Annals of Human Genetics*, **21**(4), 397–409. doi:10.1111/j.1469-1809.1972.tb00293.x.

Examples

```
library("Upsilon")

# Create a sparse table with empty rows/cols
x <- matrix(
  c(0, 3, 0,
    3, 0, 0),
  nrow = 2, byrow = TRUE
)
print(x)
# Perform the modified G-test
modified.gtest(x)
```

`upsilon.test` *Upsilon Test of Association for Count Data*

Description

Performs the Upsilon test to evaluate association among categorical variables represented by a contingency table.

Usage

```
upsilon.test(x, log.p = FALSE)
```

Arguments

`x`	a matrix or data frame of floating or integer numbers to specify a contingency table. Entries must be non-negative.
`log.p`	a logical. If `TRUE`, the *p*-value is calculated in closed form to **natural logarithm** of *p*-value to improve numerical precision when *p*-value approaches zero. Defaults to `FALSE`.

Details

The Upsilon test is designed to promote dominant function patterns. In contrast to other tests of association to favor all function patterns, it is unique in demoting non-dominant function patterns.

Null hypothesis (H_0): Row and column variables are statistically independent.

Null population: A discrete uniform distribution, where each entry in the table has the same probability.

Null distribution: The Upsilon test statistic asymptotically follows a chi-squared distribution with `(nrow(x) - 1)(ncol(x) - 1)` degrees of freedom, under the null hypothesis on the null population.

See (Luo 2021) for full details of the Upsilon test.

Value

A list with class "htest" containing:

statistic	the value of the Upsilon statistic.
parameter	the degrees of freedom.
p.value	the *p*-value.
estimate	the effect size.
method	a character string giving the test name.
data.name	a character string giving the name of input data.
observed	the observed counts, a matrix copy of the input data.
expected	the expected counts under the null hypothesis using the observed marginals.

References

Luo X (2021). *The Upsilon Test for Association Between Categorical Variables*. Master's thesis, Department of Computer Science, New Mexico State University, Las Cruces, NM, United States. Publication No. 28649813. ProQuest Dissertations and Theses Global.

Examples

```
library("Upsilon")

# A contingency table with independent row and column variables
x <- matrix(
  c(1, 1, 0,
    1, 1, 0,
    1, 1, 0),
  nrow = 3, byrow = TRUE
 )

print(x)

upsilon.test(x)

# A contingency table with a non-dominant function
x <- matrix(
  c(4, 0, 0,
    0, 1, 0,
    0, 0, 1),
  nrow = 3, byrow = TRUE
 )

print(x)

upsilon.test(x)

# A contingency table with a dominant function
x <- matrix(
  c(2, 0, 0,
    0, 2, 0,
    0, 0, 2),
  nrow = 3, byrow = TRUE)

print(x)

upsilon.test(x)

# Another contingency table with a dominant function
x <- matrix(
  c(3, 0, 0,
    0, 3, 0,
    0, 0, 0),
  nrow = 3, byrow = TRUE)

print(x)

upsilon.test(x)
```

Index