Package 'terminaldigits'

October 14, 2022

Title Tests of Uniformity and Independence for Terminal Digits

Version 0.1.0

Description Implements simulated tests for the hypothesis that terminal digits are uniformly distributed (chi-squared goodness-of-fit) and the hypothesis that terminal digits are independent from preceding digits (several tests of independence for r x c contingency tables). Also, for a number of distributions, implements Monte Carlo simulations for type I errors and power for the test of independence.

License MIT + file LICENSE

Depends R (>= 2.10)

Imports discretefit, Rcpp

Suggests dplyr, gt, ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

LinkingTo Rcpp

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

VignetteBuilder knitr

NeedsCompilation yes

Author Josh McCormick [aut, cre] (<https://orcid.org/0000-0002-2920-1119>)

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

Repository CRAN

Date/Publication 2022-05-13 09:10:12 UTC

R topics documented:

decoy *3,320 observations from a decoy experiment*

Description

A data frame containing 3,320 observations (with NA's) from the third round of a decoy experiment involving hand-washing purportedly carried out in a number of factories in China.

Usage

```
decoy
```

Format

A data frame with 3320 rows and 3 variables:

- subject
- workroom: The room for which the sanitizer weight is recorded.
- value: The weight in grams for the sanitizer.

Details

This series of experiments was published in an article in Psychological Science in 2018. Subsequently, Frank Yu, Leif Nelson, and Uri Simonsohn argued that the data for the experiments could not be trusted,and Simonsohn developed number-bunching in relation to his analysis of the data. The article was eventually retracted. This data frame consists of the data contained in the tab named "Study3-sanitizer usage(grams)".

Source

https://osf.io/wqp7y

td_independence *Test of independence of terminal digits*

Description

The `td_independence` function tests the independence of terminal digits from preceding digits by constructing a contingency table of counts where rows constitute unique preceding digits and columns constitute unique terminal digits. A test of independence for a contingency tables is then implemented via Monte Carlo simulation.

Usage

```
td_independence(
  x,
  decimals,
  reps = 10000,
  test = "Chisq",
  tolerance = 64 * .Machine$double.eps
)
```

Arguments

x	a numeric vector
decimals	an integer specifying the number of decimals. This can be zero if the terminal digit is not a decimal.
reps	a positive 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.
test	a string specifying the test of independence. The default is Pearson's chi-squared statistic ("Chisq"). Also available is the log-likelihood ratio statistic ("G2"), the Freeman-Tukey statistic ("FT"), and the Root-mean-square statistic ("RMS").
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 R.

Details

Monte Carlo simulations are implemented for contingency tables with fixed margins using algorithm ASA 144 (Agresti, Wackerly, and Boyett, 1979; Boyett 1979).

Value

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

statistic	the value of the 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

References

Agresti, A., Wackerly, D., & Boyett, J. M. (1979). Exact conditional tests for cross-classifications: approximation of attained significance levels. Psychometrika, 44(1), 75-83.

Boyett, J. M. (1979). Algorithm AS 144: Random r × c tables with given row and column totals. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(3), 329-332.

Examples

```
td_independence(decoy$weight, decimals = 2, reps = 2000)
```

`td_simulate` *Monte Carlo simulations for independence of terminal digits*

Description

The `td_simulate` function performs Monte Carlo simulations to assess type I errors and power for tests of independence of terminal digits for various truncated continuous distributions.

Usage

```
td_simulate(
  distribution,
  duplicates = 0,
  n,
  parameter_1,
  parameter_2 = NULL,
  decimals,
  significance = 0.05,
  reps = 500,
  simulations = 300,
  tolerance = 64 * .Machine$double.eps
)
```

Arguments

`distribution`	A string specifying the distribution from which to draw data for simulations. Options include "normal", "uniform", and "exponential".
`duplicates`	A number between 0 and 1 specifying the proportion of data to be comprised by duplicates. The default value is 0. This is appropriate for testing type I errors. For testing power, a value greater than 0 should be entered. For example, entering '0.05' would ensure that for each simulation, 5% of the data would be comprised by duplicates.
`n`	An integer specifying the number of observes to draw from the distribution.
`parameter_1`	A numeric value specifying the mean for the normal distribution, the lower bound of interval for the uniform distribution, or the rate for the exponential distribution.
`parameter_2`	A numeric value specifying the standard deviation for the normal distribution or the upper bound of the interval for the uniform distribution.
`decimals`	an integer specifying the number of decimals (including 0) to which the values drawn from the distribution should be truncated.

`significance`	a number between 0 and 1 defining the level for statistical significance. The default is set to 0.05.
`reps`	an integer specifying the number of Monte Carlo simulations to implement under the null for each draw. The default is set to 500 but this is only appropriate for initial exploration.
`simulations`	an integer specifying the number of Monte Carlo simulations to perform, i.e. the number of draws from the specified distribution to be tested. The default is set to 300 but this is only appropriate for initial exploration.
`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 R.

Details

Monte Carlo simulations for the null hypothesis are implemented for contingency tables with fixed margins using algorithm ASA 144 (Agresti, Wackerly, and Boyett, 1979; Boyett 1979).

Value

A list containing the following components:

`method`	method employed
`distribution`	the distribution
`Chisq`	proportion of p-values less than or equal to defined significance level for Pearson's chi-squared test of independence
`G2`	proportion of p-values less than or equal to defined significance level for log-likelihood ratio test of independence
`FT`	proportion of p-values less than or equal to defined significance level for Freeman-Tukey test of independence
`RMS`	proportion of p-values less than or equal to defined significance level for root-mean-squared test of independence
`O`	proportion of p-values less than or equal to defined significance level for occupancy test of independence
`AF`	proportion of p-values less than or equal to defined significance level for average frequency test of independence

References

Agresti, A., Wackerly, D., & Boyett, J. M. (1979). Exact conditional tests for cross-classifications: approximation of attained significance levels. Psychometrika, 44(1), 75-83.

Boyett, J. M. (1979). Algorithm AS 144: Random r × c tables with given row and column totals. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(3), 329-332.

Examples

```
td_simulate(distribution = "normal",
n = 50,
parameter_1 = 100,
parameter_2 = 1,
decimals = 1,
reps = 100,
simulations = 100)
```

`td_tests` *Tests of independence and uniformity for terminal digits in a data frame*

Description

The function `td_tests()` is a wrapper which applies the functions`td_independence()` and `td_uniformity` to a data frame. When a `group` is specified, tests are conducted separated for each group. P-values and p-values adjusted by the false discovery rate (Benjamini and Hochberg, 1995) are reported.

Usage

```
td_tests(
  data,
  variable,
  decimals,
  group = NULL,
  reps = 10000,
  test = "Chisq",
  tolerance = 64 * .Machine$double.eps
)
```

Arguments

`data`	A data frame
`variable`	A numeric variable. Tests for terminal digits are performed on this variable.
`decimals`	an integer specifying the number of decimals. This can be zero if the terminal digit is not a decimal.
`group`	A variable used to group the primary variable such that p-values are calculated separately for each group. The default is set to NULL in which case p-values are simply calculated for the whole data set.
`reps`	an integer specifying the number of Monte Carlo simulations. The default is set to 10,000.
`test`	a string specifying the test of independence. The default is Pearson's chi-squared statistic ("Chisq"). Also available is the log-likelihood ratio statistic ("G2"), the Freeman-Tukey statistic ("FT"), and the Root-mean-square statistic ("RMS").

`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 R.

Value

A data frame containing the following components:

`statistic` the value of the test statistic

`p_value_independence`
the simulated p-value for the test of independence

`P_value_uniformity`
the simulated p-value for the test of uniformity (chi-squared GOF)

`p_value_independence_fdr`
the simulated p-value for the test of independence adjusted via the false discovery rate (if the `group` variable is specified)

`P_value_uniformity`
the simulated p-value for the test of uniformity (chi-squared GOF) adjusted via the false discovery rate (if the `group` variable is specified)

References

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B, 57, 289–300. doi: 10.1111/j.2517-6161.1995.tb02031.x. https://www.jstor.org/stable/2346101.

Examples

```
td_tests(decoy, weight, decimals = 2, group = subject, reps = 1000)
```

`td_uniformity` *Test of uniformity of terminal digits*

Description

The `td_uniformity` function tests the uniformity of terminal digits via Pearson's chi-squared test of goodness-of-fit. Rather than relying on the asymptotic approximation to the chi-squared distribution, `td_unformity` uses the `chisq_gof` function from the `discretefit` package to simulate the distribution under the null.

Usage

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

Arguments

x	a numeric vector
decimals	an integer specifying the number of decimals. This can be zero if the terminal digit is not a decimal.
reps	a positive integer specifying the number of Monte Carlo simulations. The default is set to 10,000.
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 R.

Value

A list containing the following components:

statistic	the value of the 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

```
td_uniformity(decoy$weight, decimals = 2, reps = 2000)
```

Index