Package: exactcond (via r-universe)

October 22, 2026

Type Package

Title Exact Conditional Reference Distributions for Tables and Selected Linear Models

Version 0.2.0

Description Reference distributions that are exact at every sample size, obtained by conditioning on a statistic sufficient for the nuisance parameter rather than by appealing to an asymptotic approximation. Two forms of conditioning are supplied. For a two-way table the row and column totals are sufficient for the marginal parameters, so conditioning on both margins leaves the Fisher-Yates law, sampled here exactly by random permutation of the column labels; statistics include the linear-by-linear association of Goodman (1979) <doi:10.1080/01621459.1979.10481650> and, for nominal rows against an ordinal column, the dispersion of the row mean scores, whose components are the group contrasts themselves. A two-table form conditions further on the total score to cancel a common association parameter, giving a nuisance-free test that two tables share it. For a linear model under a null in which the tested structure is absent, a statistic invariant to the coefficients and the error scale depends on the data only through the residual direction, which is uniform on the sphere of the residual space; the reference stays exact when the statistic is maximised over candidate structures, and so remains valid after a spatial weight matrix has been chosen by fit. Permutation tests for a battery of categorical items and for differences between correlation networks follow the nonparametric combination of Pesarin and Salmaso (2010) <doi:10.1002/9780470689516>, with exchangeability blocks as in Winkler et al. (2014) <doi:10.1016/j.neuroimage.2014.01.060>. Every replicate draws from its own stream, so results do not depend on the thread count. The same C sources back the 'Python' package 'exactcond'.

URL https://github.com/mtloots/exactcond

BugReports https://github.com/mtloots/exactcond/issues

License GPL-3

Encoding UTF-8

Imports stats

NeedsCompilation yes

Config/testthat/edition 3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Repository https://mtloots.r-universe.dev

Date/Publication 2026-10-12 09:03:37 UTC

RemoteUrl https://github.com/mtloots/exactcond

RemoteRef HEAD

RemoteSha 4f13e1ad2622783ebcb68a577a57b1e8240dfddf

Contents

ec_diffnet_test 2
ec_multi_test 3
ec_pvalue 5
ec_rc_null 5
ec_rc_test 6
ec_rc_two 7
ec_scores 8
ec_sphere_maxquad 9

Index **11**

`ec_diffnet_test` *A differential network without a Gaussian assumption*

Description

Which pairs of variables are associated differently in two or more groups, tested without assuming the data are Gaussian. The edge statistic is the heterogeneity of the Fisher-transformed correlations across groups, $\sum_a (n_a - 3)(z_a - \bar{z})^2$, the same quantity the Gaussian theory uses; what changes is the reference. Instead of a chi-square that holds only under normality, the statistic is referred to the law obtained by permuting the group labels, which conditions on the observed data and the group sizes and is therefore exact whatever the marginals are. With `rank = TRUE` the correlations are Spearman's, so the whole procedure is distribution-free.

Usage

```
ec_diffnet_test(x, group, cluster = NULL, rank = TRUE, B = 20000L, seed = 1L)
```

Arguments

x	a matrix or data frame, variables in columns, respondents in rows.
group	a factor or vector giving each respondent's group.
cluster	an optional cluster per respondent, every cluster inside one group; when given the labels are permuted across clusters, each keeping its members.
rank	replace each variable by its ranks, average ties, before correlating. The default, and what makes the test distribution-free; set FALSE to use the values as scores, which is what an ordinal coding already is.
B	number of permutations.
seed	integer seed; each permutation has its own stream.

Details

A network has $p(p-1)/2$ edges, so testing them one at a time needs a correction that costs most of the power. The same permutations give ONE statistic for the whole network, the sum of the edge statistics when many edges differ a little and the maximum when one edge differs a lot, hence ONE p-value for "is the network different at all"; the per-edge p-values are then a closed follow-up, valid without further correction once the omnibus has rejected. Nothing is corrected because nothing is repeated.

Value

A list with sum and max over the $p(p-1)/2$ edges, each holding the observed statistic, the exact p-value, its mid-p companion, a Wilson interval and the largest null draw; and edges, a data frame of every edge with its statistic and the unadjusted per-edge p-value from the same permutations.

Examples

```
set.seed(1)
n <- 120; p <- 5
g <- rep(c("a", "b"), each = n / 2)
z <- matrix(rnorm(n * p), n)
z[g == "b", 2] <- 0.8 * z[g == "b", 1] + 0.6 * z[g == "b", 2]   # one edge differs
ec_diffnet_test(z, g, B = 999, seed = 1)
```

---

ec_multi_test *One exact test for a battery of items across groups*

Description

Several categorical items are recorded on the same respondents, who fall into groups, and the question is whether ANY item differs across the groups. Rather than test each item and pay a multiplicity correction, one statistic is formed, the sum or the maximum over items of a per-item statistic, and referred to the law obtained by permuting the group labels. That law conditions on every item's margin and on the group sizes, so it is exact conditional in the sense of ec_rc_test, and it gives ONE p-value for the whole battery. The per-item statistics are referred to the same permutations, which supplies the unadjusted per-item p-values a closed procedure uses once the omnibus has rejected.

Usage

```
ec_multi_test(
  x,
  group,
  cluster = NULL,
  stat = c("pearson", "dispersion"),
  B = 20000L,
  seed = 1L
)
```

Arguments

`x`	a matrix or data frame of items, one column per item, each a factor or integer coded from zero (or from one, which is shifted), with respondents in rows.
`group`	a factor or vector giving each respondent's group.
`cluster`	an optional factor or vector giving each respondent's cluster, every cluster lying inside one group. When supplied the labels are permuted across clusters, each cluster keeping its members, which is the exact treatment of intra-cluster correlation such as an interviewer or fieldworker effect.
`stat`	`"pearson"`, the Pearson statistic of each item's group-by-level table, which standardises each item by its own variance and is the natural choice for the sum; or `"dispersion"`, the dispersion of the group mean scores with scores `0:(k-1)`, the Kruskal-Wallis analogue on the score scale.
`B`	number of permutations.
`seed`	integer seed; each permutation uses its own stream, so the result does not depend on the thread count.

Details

The sum has power against many items moving a little; the maximum against one item moving a lot, and it is the one that matches or beats a Bonferroni-corrected battery while holding its size. With one binary item and no clusters, `"pearson"` is the exact test of the two-way table and agrees with `ec_rc_test` in distribution.

Value

A list with `sum` and `max`, each holding the observed statistic, the exact p-value, its mid-p companion, a Wilson interval for the Monte Carlo error and the largest null draw; `items`, a data frame of the per-item statistics and unadjusted per-item p-values from the same permutations; and the sizes of the groups and, if given, the number of clusters.

Examples

```
set.seed(1)
g <- rep(c("a", "b", "c"), c(30, 40, 30))
x <- cbind(rbinom(100, 1, 0.9), rbinom(100, 1, 0.95), rbinom(100, 1, ifelse(g == "c", 0.7, 0.9)))
ec_multi_test(x, g, B = 2000, seed = 1)
```

ec_pvalue — *Tail probabilities from an observed value and a null sample*

Description

Tail probabilities from an observed value and a null sample

Usage

```
ec_pvalue(obs, null)
```

Arguments

obs	the observed statistic.
null	a vector of draws from the null.

Value

A list with the p-value, its mid-p companion, and a Wilson interval reflecting the Monte Carlo error in the number of draws supplied.

Examples

```
ec_pvalue(2, rnorm(1000))
```

ec_rc_null — *Null draws of a table statistic given both margins*

Description

Null draws of a table statistic given both margins

Usage

```
ec_rc_null(
  rows,
  cols,
  stat = c("dispersion", "linlin", "pearson", "lr"),
  u = NULL,
  v = NULL,
  B = 20000L,
  seed = 1L
)
```

Arguments

`rows`, `cols`	the fixed row and column totals.
`stat`	one of `"dispersion"`, `"linlin"`, `"pearson"`, `"lr"`.
`u`, `v`	scores for the rows and columns; defaults `0:(r-1)` and `0:(c-1)`. `"dispersion"` ignores `u`, the row variable being treated as nominal.
`B`	number of draws from the conditional law.
`seed`	integer seed. Each replicate uses its own stream, so the result does not depend on the number of threads.

Value

A numeric vector of `B` draws.

Examples

```
ec_rc_null(c(30, 40), c(35, 35), B = 100, seed = 2)
```

---

ec_rc_test *Exact conditional test in a two-way table*

Description

Conditions on BOTH margins, which are sufficient for the marginal parameters, and so leaves a null distribution carrying no nuisance parameter at all. That distribution is sampled exactly, by random permutation of the column labels; nothing here is asymptotic.

Usage

```
ec_rc_test(
  x,
  stat = c("dispersion", "linlin", "pearson", "lr"),
  u = NULL,
  v = NULL,
  B = 20000L,
  seed = 1L
)
```

Arguments

`x`	an r by c table or integer matrix of counts.
`stat`	one of `"dispersion"`, `"linlin"`, `"pearson"`, `"lr"`.
`u`, `v`	scores for the rows and columns; defaults `0:(r-1)` and `0:(c-1)`. `"dispersion"` ignores `u`, the row variable being treated as nominal.
`B`	number of draws from the conditional law.
`seed`	integer seed. Each replicate uses its own stream, so the result does not depend on the number of threads.

Details

"dispersion" is for NOMINAL rows against an ORDINAL column and is $\sum_i n_i(\bar{v}_i - \bar{v})^2$. Its components are the row mean scores themselves, so in an application where the column score is a signed difference the statistic is built out of the very contrasts being reported, rather than out of ranks. "linlin" is Goodman's $\sum_{ij} u_i v_j x_{ij}$, sufficient for the linear-by-linear association parameter given both margins.

When p.value is at 1/(B+1) the observed statistic exceeded every draw, so the test has hit its resolution floor and the honest report is an upper bound together with max_null, not a p-value.

Value

A list with the observed statistic, the exact conditional p-value and its mid-p companion, a Wilson interval for the p-value reflecting the Monte Carlo error in B draws, the largest null draw seen, and the components of the statistic.

Examples

```
tab <- matrix(c(3, 228, 72, 10, 548, 67, 7, 276, 131), nrow = 3, byrow = TRUE)
ec_rc_test(tab, v = c(-1, 0, 1), B = 2000, seed = 1)
```

---

ec_rc_two — *Exact conditional test that two tables share an association parameter*

Description

Conditions on both margins WITHIN each table and then on the total score across them. Under the null of a common linear-by-linear parameter the joint density carries that parameter only through the total score, which the conditioning has fixed, so it leaves the density altogether: what remains carries no unknown parameter at all. Nothing is exchanged between the groups, so the test stays valid when the two tables have very different margins, which is exactly where permuting a group label fails.

Usage

```
ec_rc_two(
  t1,
  t2,
  u = NULL,
  v = NULL,
  B = 20000L,
  burn = 20000L,
  thin = 20L,
  alpha = 0.05,
  seed = 1L
)
```

Arguments

`t1`, `t2`	the two tables, of the same dimension.
`u`, `v`	integer scores; required to be integer valued so the total score lies on a lattice.
`B`	number of recorded draws.
`burn`	moves discarded before recording begins.
`thin`	moves between recorded draws.
`alpha`	level at which the attainable size is reported.
`seed`	integer seed.

Details

The law is sampled by moves that make a two by two swap in EACH table with equal and opposite score changes, so both sets of margins and the total score are preserved, and the chain starts at the observed pair.

That last point is what makes this usable. The obvious construction instead estimates each table's null distribution separately and convolves them, which requires the observed score to be reachable under NO association; for a strongly associated table it is not, and no increase in `B` repairs it. Worse, the failure arrives precisely when BOTH tables are strongly associated, which is the case where the correct answer is a large p-value, so the convolution fails in the direction a reader would take for evidence of a difference.

Successive draws are dependent, so accuracy is governed by the chain rather than by independent sampling. Check `accept` and raise `burn` or `thin` if it is low.

Value

A list with the conditional mid-p value, the attainable size at `alpha`, the two conditioning statistics, and the acceptance rate of the sampler.

Examples

```
a <- matrix(c(20, 8, 3, 9, 22, 7, 4, 8, 19), 3, byrow = TRUE)
b <- matrix(c(18, 9, 4, 8, 20, 8, 5, 9, 18), 3, byrow = TRUE)
ec_rc_two(a, b, B = 2000, seed = 3)
```

`ec_scores` *Integer scores for an ordinal margin*

Description

Integer scores for an ordinal margin

Usage

```
ec_scores(k, centre = FALSE)
```

Arguments

`k`	number of categories.
`centre`	if TRUE, centre the scores on zero.

Details

The dispersion and linear-by-linear statistics are unchanged in distribution by an affine transformation of the scores, so `0:(k-1)` and a centred version give identical p-values; `centre` exists only to make output easier to read.

Value

A numeric vector of length `k`.

Examples

```
ec_scores(3)
```

`ec_sphere_maxquad` *Exact null for a statistic maximised over candidate structures*

Description

Under a null in which the tested structure does not enter the data-generating process, a statistic invariant to the regression coefficients and to the error scale depends on the data only through the residual direction, which is uniform on the unit sphere of the residual space whatever those nuisance parameters are. Maximising over candidates inside each draw gives the exact null of the statistic a practitioner actually reports after selecting a structure by fit, and it is valid for ANY selection rule that is a function of that direction.

Usage

```
ec_sphere_maxquad(M, A, B = 20000L, seed = 1L)
```

Arguments

`M`	the residual maker, a symmetric idempotent n by n matrix.
`A`	a single n by n symmetric matrix, or a list of them, one per candidate.
`B`	number of draws from the conditional law.
`seed`	integer seed. Each replicate uses its own stream, so the result does not depend on the number of threads.

Value

A numeric vector of `B` draws of the maximised quadratic form.

Examples

```
n <- 12; X <- cbind(1, seq_len(n))
M <- diag(n) - X %*% solve(crossprod(X), t(X))
A <- diag(n)[, c(n, seq_len(n - 1))]; A <- (A + t(A)) / 2
quantile(ec_sphere_maxquad(M, A, B = 500, seed = 4), 0.95)
```

Index

ec_diffnet_test, 2
ec_multi_test, 3
ec_pvalue, 5
ec_rc_null, 5
ec_rc_test, *3*, *4*, 6
ec_rc_two, 7
ec_scores, 8
ec_sphere_maxquad, 9