Package 'kSamples'

July 22, 2025

Type Package

Title K-Sample Rank Tests and their Combinations

Version 1.2-10

Date 2023-10-07

Author Fritz Scholz [aut, cre],
Angie Zhu [aut]

Maintainer Fritz Scholz <fscholz@u.washington.edu>

Depends SuppDists

Imports methods, graphics, stats

Description Compares k samples using the Anderson-Darling test, Kruskal-Wallis type tests with different rank score criteria, Steel's multiple comparison test, and the Jonckheere-Terpstra (JT) test. It computes asymptotic, simulated or (limited) exact P-values, all valid under randomization, with or without ties, or conditionally under random sampling from populations, given the observed tie pattern. Except for Steel's test and the JT test it also combines these tests across several blocks of samples. Also analyzed are 2 x t contingency tables and their blocked combinations using the Kruskal-Wallis criterion. Steel's test is inverted to provide simultaneous confidence bounds for shift parameters. A plotting function compares tail probabilities obtained under asymptotic approximation with those obtained via simulation or exact calculations.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-10-07 23:10:05 UTC

Contents

`kSamples-package` *The Package kSamples Contains Several Nonparametric K-Sample Tests and their Combinations over Blocks*

Description

The k-sample Anderson-Darling, Kruskal-Wallis, normal score and van der Waerden score tests are used to test the hypothesis that k samples of sizes $n_1, \ldots, n_k$ come from a common continuous distribution F that is otherwise unspecified. They are rank tests. Average rank scores are used in case of ties. While `ad.test` is consistent against all alternatives, `qn.test` tends to be sensitive mainly to shifts between samples. The combined versions of these tests, `ad.test.combined` and `qn.test.combined`, are used to simultaneously test such hypotheses across several blocks of samples. The hypothesized common distributions and the number k of samples for each block of samples may vary from block to block.

The Jonckheere-Terpstra test addresses the same hypothesis as above but is sensitive to increasing alternatives (stochastic ordering).

Also treated is the analysis of 2 x t contingency tables using the Kruskal-Wallis criterion and its extension to blocks.

Steel's simultaneous comparison test of a common control sample with $s = k-1$ treatment samples using pairwise Wilcoxon tests for each control/treatment pair is provided, and also the simultaneous confidence bounds of treatment shift effects resulting from the inversion of these tests when sampling from continuous populations.

Distributional aspects are handled asymptotically in all cases, and by choice also via simulation or exact enumeration. While simulation is always an option, exact calculations are only possible for small sample sizes and only when few samples are involved. These exact calculations can be done with or without ties in the pooled samples, based on a recursively extended version of Algorithm C (Chase's sequence) in Knuth (2011), which allows the enumeration of all possible splits of the pooled data into samples of sizes of $n_1, \ldots, n_k$, as appropriate under treatment randomization or random sampling, when viewing tests conditionally given the observed tie pattern.

Author(s)

Fritz Scholz and Angie Zhu

Maintainer: Fritz Scholz <fscholz@u.washington.edu>

References

Hajek, J., Sidak, Z., and Sen, P.K. (1999), *Theory of Rank Tests (Second Edition)*, Academic Press.

Knuth, D.E. (2011), *The Art of Computer Programming, Volume 4A Combinatorial Algorithms Part 1*, Addison-Wesley

Kruskal, W.H. (1952), A Nonparametric Test for the Several Sample Problem, *The Annals of Mathematical Statistics*, **Vol 23, No. 4**, 525-540

Kruskal, W.H. and Wallis, W.A. (1952), Use of Ranks in One-Criterion Variance Analysis, *Journal of the American Statistical Association*, **Vol 47, No. 260**, 583–621.

Lehmann, E.L. (2006), *Nonparametrics, Statistical Methods Based on Ranks*, Revised First Edition, Springer, New York.

Scholz, F.W. (2023), "On Steel's Test with Ties", https://arxiv.org/abs/2308.05873

Scholz, F. W. and Stephens, M. A. (1987), K-sample Anderson-Darling Tests, *Journal of the American Statistical Association*, **Vol 82, No. 399**, 918–924.

`ad.pval` *P-Value for the Asymptotic Anderson-Darling Test Distribution*

Description

This function computes upper tail probabilities for the limiting distribution of the standardized Anderson-Darling test statistic.

Usage

```
ad.pval(tx,m,version=1)
```

Arguments

`tx`	a vector of desired thresholds ≥ 0
`m`	The degrees of freedom for the asymptotic standardized Anderson-Darling test statistic
`version`	= 1 (default) if P-value for version 1 of the test statistic is desired, otherwise the version 2 P-value is calculated.

Details

Extensive simulations (sampling from a common continuous distribution) were used to extend the range of the asymptotic P-value calculation from the original $[.01, .25]$ in Table 1 of the reference paper to 36 quantiles corresponding to P = .00001, .00005, .0001, .0005, .001, .005, .01, .025, .05, .075, .1, .2, .3, .4, .5, .6, .7, .8, .9, .925, .95, .975, .99, .9925, .995, .9975, .999, .99925, .9995, .99975, .9999, .999925, .99995, .999975, .99999. Note that the entries of the original Table 1 were obtained by using the first 4 moments of the asymptotic distribution and a Pearson curve approximation.

Using `ad.test`, 1 million replications of the standardized AD statistics with sample sizes $n_i = 500$, $i = 1, \ldots, k$ were run for $k = 2, 3, 4, 5, 7$ ($k = 2$ was done twice). These values of k correspond to degrees of freedom $m = k - 1 = 1, 2, 3, 4, 6$ in the asymptotic distribution. The random variable described by this distribution is denoted by T_m. The actual variances (for $n_i = 500$) agreed fairly well with the asymptotic variances.

Using the convolution nature of the asymptotic distribution, the performed simulations were exploited to result in an effective simulation of 2 million cases, except for $k = 11$, i.e., $m = k - 1 = 10$, for which the asymptotic distribution of T_{10} was approximated by the sum of the AD statistics for $k = 7$ and $k = 5$, for just the 1 million cases run for each k.

The interpolation of tail probabilities P for any desired k is done in two stages. First, a spline in $1/\sqrt{m}$ is fitted to each of the 36 quantiles obtained for $m = 1, 2, 3, 4, 6, 8, 10, \infty$ to obtain the corresponding interpolated quantiles for the m in question.

Then a spline is fitted to the $\log((1 - P)/P)$ as a function of these 36 interpolated quantiles. This latter spline is used to determine the tail probabilities P for the specified threshold `tx`, corresponding to either AD statistic version. The above procedure is based on simulations for either version of the test statistic, appealing to the same limiting distribution.

Value

a vector of upper tail probabilities corresponding to `tx`

References

Scholz, F. W. and Stephens, M. A. (1987), K-sample Anderson-Darling Tests, *Journal of the American Statistical Association*, **Vol 82, No. 399**, 918–924.

See Also

`ad.test`, `ad.test.combined`

Examples

```
ad.pval(tx=c(3.124,5.65),m=2,version=1)
ad.pval(tx=c(3.124,5.65),m=2,version=2)
```

`ad.test` *Anderson-Darling k-Sample Test*

Description

This function uses the Anderson-Darling criterion to test the hypothesis that k independent samples with sample sizes $n_1, \ldots, n_k$ arose from a common unspecified distribution function $F(x)$ and testing is done conditionally given the observed tie pattern. Thus this is a permutation test. Both versions of the AD statistic are computed.

Usage

```
ad.test(..., data = NULL, method = c("asymptotic", "simulated", "exact"),
dist = FALSE, Nsim = 10000)
```

Arguments

`...`	Either several sample vectors, say $x_1, \ldots, x_k$, with x_i containing n_i sample values. $n_i > 4$ is recommended for reasonable asymptotic P-value calculation. The pooled sample size is denoted by $N = n_1 + \ldots + n_k$, or a list of such sample vectors, or a formula y ~ g, where y contains the pooled sample values and g is a factor (of same length as y) with levels identifying the samples to which the elements of y belong.
`data`	= an optional data frame providing the variables in formula y ~ g.
`method`	= `c("asymptotic","simulated","exact")`, where `"asymptotic"` uses only an asymptotic P-value approximation, reasonable for P in [.00001, .99999] when all $n_i > 4$. Linear extrapolation via $\log(P/(1-P))$ is used outside [.00001, .99999]. This calculation is always done. See `ad.pval` for details. The adequacy of the asymptotic P-value calculation may be checked using `pp.kSamples`. `"simulated"` uses `Nsim` simulated AD statistics, based on random splits of the pooled samples into samples of sizes $n_1, \ldots, n_k$, to estimate the exact conditional P-value. `"exact"` uses full enumeration of all sample splits with resulting AD statistics to obtain the exact conditional P-values. It is used only when `Nsim` is at least as large as the number $$ncomb = \frac{N!}{n_1! \ldots n_k!}$$ of full enumerations. Otherwise, `method` reverts to `"simulated"` using the given `Nsim`. It also reverts to `"simulated"` when $ncomb > 1e8$ and `dist = TRUE`.
`dist`	= `FALSE` (default) or `TRUE`. If `TRUE`, the simulated or fully enumerated distribution vectors `null.dist1` and `null.dist2` are returned for the respective test statistic versions. Otherwise, `NULL` is returned. When `dist = TRUE` then `Nsim <- min(Nsim, 1e8)`, to limit object size.

`Nsim` = `10000` (default), number of simulation sample splits to use. It is only used when `method` = `"simulated"`, or when `method` = `"exact"` reverts to `method` = `"simulated"`, as previously explained.

Details

If AD is the Anderson-Darling criterion for the k samples, its standardized test statistic is $T.AD = (AD - \mu)/\sigma$, with $\mu = k - 1$ and σ representing mean and standard deviation of AD. This statistic is used to test the hypothesis that the samples all come from the same but unspecified continuous distribution function $F(x)$.

According to the reference article, two versions of the AD test statistic are provided. The above mean and standard deviation are strictly valid only for version 1 in the continuous distribution case.

NA values are removed and the user is alerted with the total NA count. It is up to the user to judge whether the removal of NA's is appropriate.

The continuity assumption can be dispensed with, if we deal with independent random samples, or if randomization was used in allocating subjects to samples or treatments, and if we view the simulated or exact P-values conditionally, given the tie pattern in the pooled samples. Of course, under such randomization any conclusions are valid only with respect to the group of subjects that were randomly allocated to their respective samples. The asymptotic P-value calculation assumes distribution continuity. No adjustment for lack thereof is known at this point. For details on the asymptotic P-value calculation see `ad.pval`.

Value

A list of class `kSamples` with components

`test.name`	`"Anderson-Darling"`
`k`	number of samples being compared
`ns`	vector of the k sample sizes $(n_1, \ldots, n_k)$
`N`	size of the pooled sample $= n_1 + \ldots + n_k$
`n.ties`	number of ties in the pooled samples
`sig`	standard deviations σ of version 1 of AD under the continuity assumption
`ad`	2 x 3 (2 x 4) matrix containing $AD, T.AD$, asymptotic P-value, (simulated or exact P-value), for each version of the standardized test statistic T, version 1 in row 1, version 2 in row 2.
`warning`	logical indicator, warning = TRUE when at least one $n_i < 5$
`null.dist1`	simulated or enumerated null distribution of version 1 of the test statistic, given as vector of all generated AD statistics.
`null.dist2`	simulated or enumerated null distribution of version 2 of the test statistic, given as vector of all generated AD statistics.
`method`	The `method` used.
`Nsim`	The number of simulations.

warning

`method = "exact"` should only be used with caution. Computation time is proportional to the number of enumerations. In most cases `dist = TRUE` should not be used, i.e., when the returned distribution vectors `null.dist1` and `null.dist2` become too large for the R work space. These vectors are limited in length by 1e8.

Note

For small sample sizes and small k exact null distribution calculations are possible (with or without ties), based on a recursively extended version of Algorithm C (Chase's sequence) in Knuth (2011), Ch. 7.2.1.3, which allows the enumeration of all possible splits of the pooled data into samples of sizes of $n_1, \ldots, n_k$, as appropriate under treatment randomization. The enumeration and simulation are both done in C.

Note

It has recently come to our attention that the Anderson-Darling test, originally proposed by Pettitt (1976) in the 2-sample case and generalized to k samples by Scholz and Stephens, has a close relative created by Baumgartner et al (1998) in the 2 sample case and populatized by Neuhaeuser (2012) with at least 6 papers among his cited references and generalized by Murakami (2006) to k samples.

References

Baumgartner, W., Weiss, P. and Schindler, H. (1998), A nonparametric test for the general two-sample problem, *Bionetrics*, **54**, 1129-1135.

Knuth, D.E. (2011), *The Art of Computer Programming, Volume 4A Combinatorial Algorithms Part 1*, Addison-Wesley

Neuhaeuser, M. (2012), *Nonparametric Statistical Tests, A Computational Approach*, CRC Press.

Murakami, H. (2006), A k-sample rank test based on modified Baumgartner statistic and it power comparison, *Jpn. Soc. Comp. Statist.*, **19**, 1-13.

Murakami, H. (2012), Modified Baumgartner statistic for the two-sample and multisample problems: a numerical comparison. *J. of Statistical Comput. and Simul.*, **82:5**, 711-728.

Pettitt, A.N. (1976), A two-sample Anderson_Darling rank statistic, *Biometrika*, **63**, 161-168.

Scholz, F. W. and Stephens, M. A. (1987), K-sample Anderson-Darling Tests, *Journal of the American Statistical Association*, **Vol 82, No. 399**, 918–924.

See Also

`ad.test.combined`, `ad.pval`

Examples

```
u1 <- c(1.0066, -0.9587,  0.3462, -0.2653, -1.3872)
u2 <- c(0.1005, 0.2252, 0.4810, 0.6992, 1.9289)
u3 <- c(-0.7019, -0.4083, -0.9936, -0.5439, -0.3921)
y <- c(u1, u2, u3)
```

```
g <- as.factor(c(rep(1, 5), rep(2, 5), rep(3, 5)))
set.seed(2627)
ad.test(u1, u2, u3, method = "exact", dist = FALSE, Nsim = 1000)
# or with same seed
# ad.test(list(u1, u2, u3), method = "exact", dist = FALSE, Nsim = 1000)
# or with same seed
# ad.test(y ~ g, method = "exact", dist = FALSE, Nsim = 1000)
```

`ad.test.combined` *Combined Anderson-Darling k-Sample Tests*

Description

This function combines several independent Anderson-Darling k-sample tests into one overall test of the hypothesis that the independent samples within each block come from a common unspecified distribution, while the common distributions may vary from block to block. Both versions of the Anderson-Darling test statistic are provided.

Usage

```
ad.test.combined(..., data = NULL,
method = c("asymptotic", "simulated", "exact"),
dist = FALSE, Nsim = 10000)
```

Arguments

`...`	Either a sequence of several lists, say $L_1, \ldots, L_M$ $(M > 1)$ where list L_i contains $k_i > 1$ sample vectors of respective sizes $n_{i1}, \ldots, n_{ik_i}$, where $n_{ij} > 4$ is recommended for reasonable asymptotic P-value calculation. $N_i = n_{i1} + \ldots + n_{ik_i}$ is the pooled sample size for block i, or a list of such lists, or a formula, like y ~ g | b, where y is a numeric response vector, g is a factor with levels indicating different treatments and b is a factor indicating different blocks; y, g, b are or equal length. y is split separately for each block level into separate samples according to the g levels. The same g level may occur in different blocks. The variable names may correspond to variables in an optionally supplied data frame via the data = argument,
`data`	= an optional data frame providing the variables in formula input
`method`	= `c("asymptotic","simulated","exact")`, where `"asymptotic"` uses only an asymptotic P-value approximation, reasonable for P in [0.00001, .99999], linearly extrapolated via $\log(P/(1-P))$ outside that range. See `ad.pval` for details. The adequacy of the asymptotic P-value calculation may be checked using `pp.kSamples`. `"simulated"` uses simulation to get `Nsim` simulated AD statistics for each block of samples, adding them across blocks component wise to get `Nsim` combined values. These are compared with the observed combined value to obtain the estimated P-value.

"exact" uses full enumeration of the test statistic values for all sample splits of the pooled samples within each block. The test statistic vectors for the first 2 blocks are added (each component against each component, as in the R outer(x,y, "+") command) to get the convolution enumeration for the combined test statistic. The resulting vector is convoluted against the next block vector in the same fashion, and so on. It is possible only for small problems, and is attempted only when Nsim is at least the (conservatively maximal) length

$$\frac{N_1!}{n_{11}!\ldots n_{1k_1}!} \times \ldots \times \frac{N_M!}{n_{M1}!\ldots n_{Mk_M}!}$$

of the final distribution vector. Otherwise, it reverts to the simulation method using the provided Nsim.

dist — FALSE (default) or TRUE. If TRUE, the simulated or fully enumerated convolution vectors null.dist1 and null.dist2 are returned for the respective test statistic versions. Otherwise, NULL is returned for each.

Nsim — = 10000 (default), number of simulation splits to use within each block of samples. It is only used when method = "simulated" or when method = "exact" reverts to method = "simulated", as previously explained. Simulations are independent across blocks, using Nsim for each block. Nsim is limited by 1e7.

Details

If AD_i is the Anderson-Darling criterion for the i-th block of k_i samples, its standardized test statistic is $T_i = (AD_i - \mu_i)/\sigma_i$, with μ_i and σ_i representing mean and standard deviation of AD_i. This statistic is used to test the hypothesis that the samples in the i-th block all come from the same but unspecified continuous distribution function $F_i(x)$.

The combined Anderson-Darling criterion is $AD_{comb} = AD_1 + \ldots + AD_M$ and $T_{comb} = (AD_{comb} - \mu_c)/\sigma_c$ is the standardized form, where $\mu_c = \mu_1 + \ldots + \mu_M$ and $\sigma_c = \sqrt{\sigma_1^2 + \ldots + \sigma_M^2}$ represent the mean and standard deviation of AD_{comb}. The statistic T_{comb} is used to simultaneously test whether the samples in each block come from the same continuous distribution function $F_i(x), i = 1, \ldots, M$. The unspecified common distribution function $F_i(x)$ may change from block to block. According to the reference article, two versions of the test statistic and its corresponding combinations are provided.

The k_i for each block of k_i independent samples may change from block to block.

NA values are removed and the user is alerted with the total NA count. It is up to the user to judge whether the removal of NA's is appropriate.

The continuity assumption can be dispensed with if we deal with independent random samples, or if randomization was used in allocating subjects to samples or treatments, independently from block to block, and if we view the simulated or exact P-values conditionally, given the tie patterns within each block. Of course, under such randomization any conclusions are valid only with respect to the blocks of subjects that were randomly allocated. The asymptotic P-value calculation assumes distribution continuity. No adjustment for lack thereof is known at this point. The same comment holds for the means and standard deviations of respective statistics.

Value

A list of class kSamples with components

test.name	= "Anderson-Darling"
M	number of blocks of samples being compared
n.samples	list of M vectors, each vector giving the sample sizes for each block of samples being compared
nt	$= (N_1, \ldots, N_M)$
n.ties	vector giving the number of ties in each the M comparison blocks
ad.list	list of M matrices giving the ad results for ad.test applied to the samples in each of the M blocks
mu	vector of means of the AD statistic for the M blocks
sig	vector of standard deviations of the AD statistic for the M blocks
ad.c	2 x 3 (2 x 4) matrix containing AD_{comb}, T_{comb}, asymptotic P-value, (simulated or exact P-value), for each version of the combined test statistic, version 1 in row 1 and version 2 in row 2
mu.c	mean of AD_{comb}
sig.c	standard deviation of AD_{comb}
warning	logical indicator, warning = TRUE when at least one $n_{ij} < 5$
null.dist1	simulated or enumerated null distribution of version 1 of AD_{comb}
null.dist2	simulated or enumerated null distribution of version 2 of AD_{comb}
method	the method used.
Nsim	the number of simulations used for each block of samples.

Note

This test is useful in analyzing treatment effects in randomized (incomplete) block experiments and in examining performance equivalence of several laboratories when presented with different test materials for comparison.

References

Scholz, F. W. and Stephens, M. A. (1987), K-sample Anderson-Darling Tests, *Journal of the American Statistical Association*, **Vol 82, No. 399**, 918–924.

See Also

ad.test, ad.pval

Examples

```
## Create two lists of sample vectors.
x1 <- list( c(1, 3, 2, 5, 7), c(2, 8, 1, 6, 9, 4), c(12, 5, 7, 9, 11) )
x2 <- list( c(51, 43, 31, 53, 21, 75), c(23, 45, 61, 17, 60) )
# and a corresponding data frame datx1x2
x1x2 <- c(unlist(x1),unlist(x2))
gx1x2 <- as.factor(c(rep(1,5),rep(2,6),rep(3,5),rep(1,6),rep(2,5)))
bx1x2 <- as.factor(c(rep(1,16),rep(2,11)))
```

```
datx1x2 <- data.frame(A = x1x2, G = gx1x2, B = bx1x2)

## Run ad.test.combined.
set.seed(2627)
ad.test.combined(x1, x2, method = "simulated", Nsim = 1000)
# or with same seed
# ad.test.combined(list(x1, x2), method = "simulated", Nsim = 1000)
# ad.test.combined(A~G|B,data=datx1x2,method="simulated",Nsim=1000)
```

`contingency2xt` *Kruskal-Wallis Test for the 2 x t Contingency Table*

Description

This function uses the Kruskal-Wallis criterion to test the hypothesis of no association between the counts for two responses "A" and "B" across t categories.

Usage

```
contingency2xt(Avec, Bvec,
method = c("asymptotic", "simulated", "exact"),
dist = FALSE, tab0 = TRUE, Nsim = 1e+06)
```

Arguments

`Avec`	vector of length t giving the counts $A_1, \ldots, A_t$ for response "A" according to t categories. $m = A_1 + \ldots + A_t$.
`Bvec`	vector of length t giving the counts $B_1, \ldots, B_t$ for response "B" according to t categories. $n = B_1 + \ldots + B_t = N - m$.
`method`	= `c("asymptotic","simulated","exact")`, where `"asymptotic"` uses only an asymptotic chi-square approximation with $t - 1$ degrees of freedom to approximate the P-value. This calculation is always done. `"simulated"` uses `Nsim` simulated counts for `Avec` and `Bvec` with the observed marginal totals, `m, n, d = Avec+Bvec`, to estimate the P-value. `"exact"` enumerates all counts for `Avec` and `Bvec` with the observed marginal totals to get an exact P-value. It is used only when `Nsim` is at least as large as the number `choose(m+t-1,t-1)` of full enumerations. Otherwise, `method` reverts to `"simulated"` using the given `Nsim`.
`dist`	`FALSE` (default) or `TRUE`. If `dist = TRUE`, the distribution of the simulated or fully enumerated Kruskal-Wallis test statistics is returned as `null.dist`, if `dist = FALSE` the value of `null.dist` is `NULL`. The coice `dist = TRUE` also limits `Nsim <- min(Nsim,1e8)`.

`tab0` TRUE (default) or FALSE. If `tab0 = TRUE`, the null distribution is returned in 2 column matrix form when `method = "simulated"`. When `tab0 = FALSE` the simulated null distribution is returned as a vector of all simulated values of the test statistic.

`Nsim` `=10000` (default), number of simulated Avec splits to use. It is only used when `method = "simulated"`, or when `method = "exact"` reverts to `method = "simulated"`, as previously explained.

Details

For this data scenario the Kruskal-Wallis criterion is

$$K.star = \frac{N(N-1)}{mn}(\sum \frac{A_i^2}{d_i} - \frac{m^2}{N})$$

with $d_i = A_i + B_i$, treating "A" responses as 1 and "B" responses as 2, and using midranks as explained in Lehmann (2006), Chapter 5.3.

For small sample sizes exact null distribution calculations are possible, based on Algorithm C (Chase's sequence) in Knuth (2011), which allows the enumeration of all possible splits of m into counts $A_1, \ldots, A_t$ such that $m = A_1 + \ldots + A_t$, followed by the calculation of the statistic $K.star$ for each such split. Simulation of $A_1, \ldots, A_t$ uses the probability model (5.35) in Lehmann (2006) to successively generate hypergeometric counts $A_1, \ldots, A_t$. Both these processes, enumeration and simulation, are done in C.

Value

A list of class `kSamples` with components

`test.name` `"2 x t Contingency Table"`

`t` number of classification categories

`KW.cont` 2 (3) vector giving the observed KW statistic, its asymptotic P-value (and simulated or exact P-value)

`null.dist` simulated or enumerated null distribution of the test statistic. It is given as an `M` by 2 matrix, where the first column (named `KW`) gives the `M` unique ordered values of the Kruskal-Wallis statistic and the second column (named `prob`) gives the corresponding (simulated or exact) probabilities.

This format of `null.dist` is returned when `method = "exact"` and `dist = TRUE` or when `method = "simulated"` and `dist = TRUE` and `tab0 = TRUE` are specified.

For `method = "simulated"`, `dist = TRUE`, and `tab0 = FALSE` the null distribution `null.dist` is returned as the vector of all simulated test statistic values. This is used in `contingency2xt.comb` in the simulation mode.

`null.dist = NULL` is returned when `dist = FALSE` or when `method = "asymptotic"`.

`method` the `method` used.

`Nsim` the number of simulations.

warning

`method = "exact"` should only be used with caution. Computation time is proportional to the number of enumerations. In most cases `dist = TRUE` should not be used, i.e., when the returned distribution objects become too large for R's work space.

References

Knuth, D.E. (2011), *The Art of Computer Programming, Volume 4A Combinatorial Algorithms Part 1*, Addison-Wesley

Kruskal, W.H. (1952), A Nonparametric Test for the Several Sample Problem, *The Annals of Mathematical Statistics*, **Vol 23, No. 4**, 525-540

Kruskal, W.H. and Wallis, W.A. (1952), Use of Ranks in One-Criterion Variance Analysis, *Journal of the American Statistical Association*, **Vol 47, No. 260**, 583–621.

Lehmann, E.L. (2006), *Nonparametrics, Statistical Methods Based on Ranks*, Revised First Edition, Springer, New York.

Examples

```
contingency2xt(c(25,15,20),c(16,6,18),method="exact",dist=FALSE,
tab0=TRUE,Nsim=1e3)
```

---

contingency2xt.comb *Combined Kruskal-Wallis Tests for the 2 x t Contingency Tables*

Description

This function uses the Kruskal-Wallis criterion to test the hypothesis of no association between the counts for two responses "A" and "B" across t categories and across M blocks.

Usage

```
contingency2xt.comb(...,
method = c("asymptotic", "simulated", "exact"),
dist = FALSE, Nsim = 10000)
```

Arguments

`...`	Either several lists $L_1, \ldots, L_M$, each of two equal length vectors A_i and B_i, $i = 1, \ldots, M$, of counts ≥ 0, where the common length t_i of A_i and B_i may vary from list to list or a list of `M` such lists
`method`	`= c("asymptotic","simulated","exact")`, where `"asymptotic"` uses only an asymptotic chi-square approximation with $(t_1-1)+\ldots+(t_M-1)$ degrees of freedom to approximate the P-value, This calculation is always done.

"simulated" uses Nsim simulated counts for the two vectors A_i and B_i in list L_i, with the observed marginal totals, $m_i = \sum A_i$, $n_i = \sum B_i$, $d_i = A_i + B_i$. It does this independently from list to list using the same Nsim each time, adding the resulting Kruskal-Wallis criteria across lists to get Nsim such summed values to estimate the P-value.

"exact" enumerates all counts for A_i and B_i with the respective observed marginal totals to get an exact distribution for each list. These distributions are then convolved to obtain the P-value. It is used only when Nsim is at least as large as the product across blocks of the number choose(m+t-1,t-1) of full enumerations per block, where $t = t_1, \ldots, t_M$. Otherwise, method reverts to "simulated" using the given Nsim.

dist — FALSE (default) or TRUE. If TRUE, the simulated or fully enumerated null distribution null.dist is returned for the Kruskal-Wallis test statistic. Otherwise null.dist = NULL is returned.

Nsim — =10000 (default), number of simulated A_i splits to use per block. It is only used when method = "simulated", or when method = "exact" reverts to method = "simulated", as previously explained.

Details

For details on the calculation of the Kruskal-Wallis criterion and its exact or simulated distribution for each block see contingency2xt.

Value

A list of class kSamples with components

test.name — "Combined 2 x t Contingency Tables"

t — vector giving the number of classification categories per block

M — number of blocked tables

kw.list — a list of the KW.cont output componenents from contingency2xt for each of the blocks

null.dist — simulated or enumerated null distribution of the combined test statistic. It is given as an L by 2 matrix, where the first column (named KW) gives the L unique ordered values of the combined Kruskal-Wallis statistic and the second column (named prob) gives the corresponding (simulated or exact) probabilities.

null.dist = NULL is returned when dist = FALSE or when method = "asymptotic".

method — the method used.

Nsim — the number of simulations.

warning

method = "exact" should only be used with caution. Computation time is proportional to the number of enumerations. In most cases dist = TRUE should not be used, i.e., when the returned distribution objects become too large for R's work space.

Note

The required level for `Nsim` in order for `method = "exact"` to be carried out, is conservative, but there is no transparent way to get a better estimate. The actual dimension `L` of the realized `null.dist` will typically be much smaller, since the distribution is compacted to its unique support values.

Examples

```
out <- contingency2xt.comb(list(c(25,15,20),c(16,6,18)),
list(c(12,4,5),c(13,8,9)),method = "simulated", dist=FALSE, Nsim=1e3)
```

---

conv *Convolution of Two Discrete Distributions*

Description

This function convolutes two discrete distribution, each given by strictly increasing support vectors and corresponding probability vectors.

Usage

```
conv(x1,p1,x2,p2)
```

Arguments

`x1`	support vector of the first distribution, with strictly increasing elements.
`p1`	vector of probabilities corresponding to `x1`.
`x2`	support vector of the second distribution, with strictly increasing elements.
`p2`	vector of probabilities corresponding to `x2`.

Details

The convolution is performed in C, looping through all paired sums, augmenting existing values or inserting them with an update of the corresponding probabilities.

Value

A matrix with first column the new support vector and the second column the corresponding probability vector.

Examples

```
x1 <- c(1,2,3.5)
p1 <- c(.2,.3,.5)
x2 <- c(0,2.3,3,4)
p2 <- c(.1,.3,.3,.3)


conv(x1,p1,x2,p2)
```

JT.dist *Null Distribution of the Jonckheere-Terpstra k-Sample Test Statistic*

Description

The Jonckheere-Terpstra k-sample test statistic JT is defined as $JT = \sum_{i<j} W_{ij}$ where W_{ij} is the Mann-Whitney statistic comparing samples i and j, indexed in the order of the stipulated increasing alternative. It is assumed that there are no ties in the pooled samples.

This function uses Harding's algorithm as far as computations are possible without becoming unstable.

Usage

```
djt(x, nn)

pjt(x, nn)

qjt(p, nn)
```

Arguments

x	a numeric vector, typically integers
nn	a vector of integers, representing the sample sizes in the order stipulated by the alternative
p	a vector of probabilities

Details

While Harding's algorithm is mathematically correct, it is problematic in its computing implementation. The counts become very large and normalizing them by combinatorials leads to significance loss. When that happens the functions return an error message: can't compute due to numerical instability. This tends to happen when the total number of sample values becomes too large. That depends also on the way the sample sizes are allocated.

Value

For djt it is a vector $p = (p_1, \ldots, p_n)$ giving the values of $p_i = P(JT = x_i)$, where n is the length of the input x.

For pjt it is a vector $P = (P_1, \ldots, P_n)$ giving the values of $P_i = P(JT \leq x_i)$.

For qjt is a vecto r $x = (x_1, \ldots, x_n)$,where x_i is the smallest x such that $P(JT \leq x) \geq p_i$.

References

Harding, E.F. (1984), An Efficient, Minimal-storage Procedure for Calculating the Mann-Whitney U, Generalized U and Similar Distributions, *Appl. Statist.* **33** No. 1, 1-6.

Jonckheere, A.R. (1954), A Distribution Free *k*-sample Test against Ordered Alternatives, *Biometrika*, **41**, 133-145.

Lehmann, E.L. (2006), *Nonparametrics, Statistical Methods Based on Ranks, Revised First Edition*, Springer Verlag.

Terpstra, T.J. (1952), The Asymptotic Normality and Consistency of Kendall's Test against Trend, when Ties are Present in One Ranking, *Indagationes Math.* **14**, 327-333.

Examples

```
djt(c(-1.5,1.2,3), 2:4)
pjt(c(2,3.4,7), 3:5)
qjt(c(0,.2,.5), 2:4)
```

---

jt.test *Jonckheere-Terpstra k-Sample Test for Increasing Alternatives*

Description

The Jonckheere-Terpstra k-sample test statistic JT is defined as $JT = \sum_{i<j} W_{ij}$ where W_{ij} is the Mann-Whitney statistic comparing samples i and j, indexed in the order of the stipulated increasing alternative. There may be ties in the pooled samples.

Usage

```
jt.test(..., data = NULL, method=c("asymptotic","simulated","exact"),
dist = FALSE, Nsim = 10000)
```

Arguments

...	Either several sample vectors, say $x_1, \ldots, x_k$, with x_i containing n_i sample values. $n_i > 4$ is recommended for reasonable asymptotic P-value calculation. The pooled sample size is denoted by $N = n_1 + \ldots + n_k$. The order of samples should be as stipulated under the alternative or a list of such sample vectors, or a formula y ~ g, where y contains the pooled sample values and g (same length as y) is a factor with levels identifying the samples to which the elements of y belong, the factor levels reflecting the order under the stipulated alternative,
data	= an optional data frame providing the variables in formula y ~ g.
method	= c("asymptotic","simulated","exact"), where "asymptotic" uses only an asymptotic normal P-value approximation. "simulated" uses Nsim simulated JT statistics based on random splits of the pooled samples into samples of sizes $n_1, \ldots, n_k$, to estimate the P-value.

"exact" uses full enumeration of all sample splits with resulting JT statistics to obtain the exact P-value. It is used only when Nsim is at least as large as the number

$$ncomb = \frac{N!}{n_1! \ldots n_k!}$$

of full enumerations. Otherwise, method reverts to "simulated" using the given Nsim. It also reverts to "simulated" when $ncomb > 1e8$ and dist = TRUE.

dist = FALSE (default) or TRUE. If TRUE, the simulated or fully enumerated distribution vector null.dist is returned for the JT test statistic. Otherwise, NULL is returned. When dist = TRUE then Nsim <- min(Nsim, 1e8), to limit object size.

Nsim = 10000 (default), number of simulation sample splits to use. It is only used when method = "simulated", or when method = "exact" reverts to method = "simulated", as previously explained.

Details

The JT statistic is used to test the hypothesis that the samples all come from the same but unspecified continuous distribution function $F(x)$. It is specifically aimed at alternatives where the sampled distributions are stochastically increasing.

NA values are removed and the user is alerted with the total NA count. It is up to the user to judge whether the removal of NA's is appropriate.

The continuity assumption can be dispensed with, if we deal with independent random samples, or if randomization was used in allocating subjects to samples or treatments, and if we view the simulated or exact P-values conditionally, given the tie pattern in the pooled samples. Of course, under such randomization any conclusions are valid only with respect to the group of subjects that were randomly allocated to their respective samples. The asymptotic P-value calculation is valid provided all sample sizes become large.

Value

A list of class kSamples with components

test.name	"Jonckheere-Terpstra"
k	number of samples being compared
ns	vector $(n_1, \ldots, n_k)$ of the k sample sizes
N	size of the pooled sample $= n_1 + \ldots + n_k$
n.ties	number of ties in the pooled sample
qn	4 (or 5) vector containing the observed JT, its mean and standard deviation and its asymptotic P-value, (and its simulated or exact P-value)
warning	logical indicator, warning = TRUE when at least one $n_i < 5$
null.dist	simulated or enumerated null distribution of the test statistic. It is NULL when dist = FALSE or when method = "asymptotic".
method	the method used.
Nsim	the number of simulations used.

References

Harding, E.F. (1984), An Efficient, Minimal-storage Procedure for Calculating the Mann-Whitney U, Generalized U and Similar Distributions, *Appl. Statist.* **33** No. 1, 1-6.

Jonckheere, A.R. (1954), A Distribution Free *k*-sample Test against Ordered Alternatives, *Biometrika*, **41**, 133-145.

Lehmann, E.L. (2006), *Nonparametrics, Statistical Methods Based on Ranks, Revised First Edition*, Springer Verlag.

Terpstra, T.J. (1952), The Asymptotic Normality and Consistency of Kendall's Test against Trend, when Ties are Present in One Ranking, *Indagationes Math.* **14**, 327-333.

Examples

```
x1 <- c(1,2)
x2 <- c(1.5,2.1)
x3 <- c(1.9,3.1)
yy <- c(x1,x2,x3)
gg <- as.factor(c(1,1,2,2,3,3))
jt.test(x1, x2, x3,method="exact",Nsim=90)
# or
# jt.test(list(x1, x2, x3), method = "exact", Nsim = 90)
# or
# jt.test(yy ~ gg, method = "exact", Nsim = 90)
```

---

pp.kSamples *Upper Tail Probability Plots for Objects of Class kSamples*

Description

This function plots upper tail probabilities of the limiting distribution against the corresponding exact or simulated probabilities, both on a log-scale.

Usage

```
pp.kSamples(x)
```

Arguments

x	an object of class kSamples

Details

Objects of class kSamples are produced by any of the following functions

ad.test Anderson-Darling k-sample test.

ad.test.combined Combined Anderson-Darling k-sample tests.

qn.test QN rank scores test.

qn.test.combined Combined QN rank scores tests.

contingency2xt test for $2 * t$ contingency table.

contingency2xt.comb test for the combination of $2 * t$ contingency tables.

jt.test Jonckheere-Terpstra test.

Steel.test Steel test. This will work only for alternative = "greater" or "two-sided". The approximation quality for "less" is the same as for "greater".

The command pp.kSamples(x) for an object of class kSamples will only produce a plot when the object x contains non-NULL entries for the null distribution. The purpose of this function is to give the user a sense of the asymptotic distribution accuracy.

See Also

ad.test, ad.test.combined, qn.test, qn.test.combined,

contingency2xt, contingency2xt.comb jt.test Steel.test

Examples

```
qn.out <- qn.test(c(1,3,7,2,9),c(1,4,6,11,2),test="KW",
method="simulated",dist=TRUE,Nsim=1000)
pp.kSamples(qn.out)
```

---

qn.test *Rank Score k-Sample Tests*

Description

This function uses the QN criterion (Kruskal-Wallis, van der Waerden scores, normal scores) to test the hypothesis that k independent samples arise from a common unspecified distribution.

Usage

```
qn.test(..., data = NULL, test = c("KW", "vdW", "NS"),
method = c("asymptotic", "simulated", "exact"),
dist = FALSE, Nsim = 10000)
```

Arguments

...	Either several sample vectors, say $x_1, \ldots, x_k$, with x_i containing n_i sample values. $n_i > 4$ is recommended for reasonable asymptotic P-value calculation. The pooled sample size is denoted by $N = n_1 + \ldots + n_k$, or a list of such sample vectors, or a formula y ~ g, where y contains the pooled sample values and g (same length as y) is a factor with levels identifying the samples to which the elements of y belong.
data	= an optional data frame providing the variables in formula y ~ g.

`test` = `c("KW", "vdW", "NS")`, where

`"KW"` uses scores `1:N` (Kruskal-Wallis test)

`"vdW"` uses van der Waerden scores, `qnorm( (1:N) / (N+1) )`

`"NS"` uses normal scores, i.e., expected standard normal order statistics, invoking function `normOrder` of package `SuppDists (>=1.1-9.4)`

`method` = `c("asymptotic","simulated","exact")`, where

`"asymptotic"` uses only an asymptotic chi-square approximation with `k-1` degrees of freedom to approximate the P-value. This calculation is always done.

`"simulated"` uses `Nsim` simulated QN statistics based on random splits of the pooled samples into samples of sizes $n_1, \ldots, n_k$, to estimate the P-value.

`"exact"` uses full enumeration of all sample splits with resulting QN statistics to obtain the exact P-value. It is used only when `Nsim` is at least as large as the number

$$ncomb = \frac{N!}{n_1! \ldots n_k!}$$

of full enumerations. Otherwise, `method` reverts to `"simulated"` using the given `Nsim`. It also reverts to `"simulated"` when $ncomb > 1e8$ and `dist = TRUE`.

`dist` `FALSE` (default) or `TRUE`. If `TRUE`, the simulated or fully enumerated null distribution vector `null.dist` is returned for the QN test statistic. Otherwise, `NULL` is returned. When `dist = TRUE` then `Nsim <- min(Nsim, 1e8)`, to limit object size.

`Nsim` = `10000` (default), number of simulation sample splits to use. It is only used when `method = "simulated"`, or when `method = "exact"` reverts to `method = "simulated"`, as previously explained.

Details

The QN criterion based on rank scores $v_1, \ldots, v_N$ is

$$QN = \frac{1}{s_v^2} \left(\sum_{i=1}^{k} \frac{(S_{iN} - n_i \bar{v}_N)^2}{n_i} \right)$$

where S_{iN} is the sum of rank scores for the i-th sample and $\bar{v}_N$ and s_v^2 are sample mean and sample variance (denominator $N-1$) of all scores.

The statistic QN is used to test the hypothesis that the samples all come from the same but unspecified continuous distribution function $F(x)$. QN is always adjusted for ties by averaging the scores of tied observations.

Conditions for the asymptotic approximation (chi-square with $k-1$ degrees of freedom) can be found in Lehmann, E.L. (2006), Appendix Corollary 10, or in Hajek, Sidak, and Sen (1999), Ch. 6, problems 13 and 14.

For small sample sizes exact null distribution calculations are possible (with or without ties), based on a recursively extended version of Algorithm C (Chase's sequence) in Knuth (2011), which allows the enumeration of all possible splits of the pooled data into samples of sizes of $n_1, \ldots, n_k$, as appropriate under treatment randomization. This is done in C, as is the simulation.

NA values are removed and the user is alerted with the total NA count. It is up to the user to judge whether the removal of NA's is appropriate.

The continuity assumption can be dispensed with, if we deal with independent random samples from any common distribution, or if randomization was used in allocating subjects to samples or treatments, and if the asymptotic, simulated or exact P-values are viewed conditionally, given the tie pattern in the pooled sample. Under such randomization any conclusions are valid only with respect to the subjects that were randomly allocated to their respective treatment samples.

Value

A list of class `kSamples` with components

`test.name`	`"Kruskal-Wallis"`, `"van der Waerden scores"`, or `"normal scores"`
`k`	number of samples being compared
`ns`	vector $(n_1, \ldots, n_k)$ of the k sample sizes
`N`	size of the pooled samples $= n_1 + \ldots + n_k$
`n.ties`	number of ties in the pooled sample
`qn`	2 (or 3) vector containing the observed QN, its asymptotic P-value, (its simulated or exact P-value)
`warning`	logical indicator, `warning = TRUE` when at least one $n_i < 5$
`null.dist`	simulated or enumerated null distribution of the test statistic. It is `NULL` when `dist = FALSE` or when `method = "asymptotic"`.
`method`	the `method` used.
`Nsim`	the number of simulations used.

warning

`method = "exact"` should only be used with caution. Computation time is proportional to the number of enumerations. Experiment with `system.time` and trial values for `Nsim` to get a sense of the required computing time. In most cases `dist = TRUE` should not be used, i.e., when the returned distribution objects become too large for R's work space.

References

Hajek, J., Sidak, Z., and Sen, P.K. (1999), *Theory of Rank Tests (Second Edition)*, Academic Press.

Knuth, D.E. (2011), *The Art of Computer Programming, Volume 4A Combinatorial Algorithms Part 1*, Addison-Wesley

Kruskal, W.H. (1952), A Nonparametric Test for the Several Sample Problem, *The Annals of Mathematical Statistics*, **Vol 23, No. 4**, 525-540

Kruskal, W.H. and Wallis, W.A. (1952), Use of Ranks in One-Criterion Variance Analysis, *Journal of the American Statistical Association*, **Vol 47, No. 260**, 583–621.

Lehmann, E.L. (2006), *Nonparametrics, Statistical Methods Based on Ranks, Revised First Edition*, Springer Verlag.

See Also

`qn.test.combined`

Examples

```
u1 <- c(1.0066, -0.9587,  0.3462, -0.2653, -1.3872)
u2 <- c(0.1005, 0.2252, 0.4810, 0.6992, 1.9289)
u3 <- c(-0.7019, -0.4083, -0.9936, -0.5439, -0.3921)
yy <- c(u1, u2, u3)
gy <- as.factor(c(rep(1,5), rep(2,5), rep(3,5)))
set.seed(2627)
qn.test(u1, u2, u3, test="KW", method = "simulated",
  dist = FALSE, Nsim = 1000)
# or with same seed
# qn.test(list(u1, u2, u3),test = "KW", method = "simulated",
#  dist = FALSE, Nsim = 1000)
# or with same seed
# qn.test(yy ~ gy, test = "KW", method = "simulated",
#  dist = FALSE, Nsim = 1000)
```

`qn.test.combined` *Combined Rank Score k-Sample Tests*

Description

This function combines several independent rank score k-sample tests into one overall test of the hypothesis that the independent samples within each block come from a common unspecified distribution, while the common distributions may vary from block to block.

Usage

```
qn.test.combined(..., data = NULL, test = c("KW", "vdW", "NS"),
method = c("asymptotic", "simulated", "exact"),
dist = FALSE, Nsim = 10000)
```

Arguments

`...`	Either a sequence of several lists, say $L_1, \ldots, L_M$ ($M > 1$) where list L_i contains $k_i > 1$ sample vectors of respective sizes $n_{i1}, \ldots, n_{ik_i}$, where $n_{ij} > 4$ is recommended for reasonable asymptotic P-value calculation. $N_i = n_{i1} + \ldots + n_{ik_i}$ is the pooled sample size for block i, or a list of such lists, or a formula, like y ~ g | b, where y is a numeric response vector, g is a factor with levels indicating different treatments and b is a factor indicating different blocks; y, g, b have same length. y is split separately for each block level into separate samples according to the g levels. The same g level may occur in different blocks. The variable names may correspond to variables in an optionally supplied data frame via the data = argument.
`data`	= an optional data frame providing the variables in formula input

`test` = `c("KW", "vdW", "NS")`,

where `"KW"` uses scores `1:N` (Kruskal-Wallis test)

`"vdW"` uses van der Waerden scores, `qnorm( (1:N) / (N+1) )`

`"NS"` uses normal scores, i.e., expected values of standard normal order statistics, invoking function `normOrder` of package `SuppDists (>=1.1-9.4)`

For the above scores N changes from block to block and represents the total pooled sample size N_i for block i.

`method` =`c("asymptotic","simulated","exact")`, where

`"asymptotic"` uses only an asymptotic chi-square approximation for the P-value. The adequacy of asymptotic P-values for use with moderate sample sizes may be checked with `method = "simulated"`.

`"simulated"` uses simulation to get `Nsim` simulated QN statistics for each block of samples, adding them component wise across blocks to get `Nsim` combined values, and compares these with the observed combined value to get the estimated P-value.

`"exact"` uses full enumeration of the test statistic value for all sample splits of the pooled samples within each block. The test statistic vectors for each block are added (each component against each component, as in the R `outer(x,y, "+")` command) to get the convolution enumeration for the combined test statistic. This "addition" is done one block at a time. It is possible only for small problems, and is attempted only when `Nsim` is at least the (conservatively maximal) length

$$\frac{N_1!}{n_{11}!\ldots n_{1k_1}!} \times \ldots \times \frac{N_M!}{n_{M1}!\ldots n_{Mk_M}!}$$

of the final distribution vector, were $N_i = n_{i1} + \ldots + n_{ik_i}$ is the sample size of the pooled samples for the i-th block. Otherwise, it reverts to the simulation method using the provided `Nsim`.

`dist` `FALSE` (default) or `TRUE`. If `TRUE`, the simulated or fully enumerated convolution vector `null.dist` is returned for the QN test statistic.

Otherwise, `NULL` is returned.

`Nsim` = `10000` (default), number of simulation splits to use within each block of samples. It is only used when `method = "simulated"` or when `method = "exact"` reverts to `method = "simulated"`, as previously explained. Simulations are independent across blocks, using `Nsim` for each block.

Details

The rank score QN criterion QN_i for the i-th block of k_i samples, is used to test the hypothesis that the samples in the i-th block all come from the same but unspecified continuous distribution function $F_i(x)$. See `qn.test` for the definition of the QN criterion and the exact calculation of its null distribution.

The combined QN criterion $QN_{\text{comb}} = QN_1 + \ldots + QN_M$ is used to simultaneously test whether the samples in block i come from the same continuous distribution function $F_i(x)$. However, the unspecified common distribution function $F_i(x)$ may change from block to block.

The k for each block of k independent samples may change from block to block.

The asymptotic approximating chi-square distribution has $f = (k_1 - 1) + \ldots + (k_M - 1)$ degrees of freedom.

NA values are removed and the user is alerted with the total NA count. It is up to the user to judge whether the removal of NA's is appropriate.

The continuity assumption can be dispensed with if we deal with independent random samples, or if randomization was used in allocating subjects to samples or treatments, independently from block to block, and if the asymptotic, simulated or exact P-values are viewed conditionally, given the tie patterns within each block. Under such randomization any conclusions are valid only with respect to the blocks of subjects that were randomly allocated. In case of ties the average rank scores are used across tied observations within each block.

Value

A list of class `kSamples` with components

`test.name`	`"Kruskal-Wallis"`, `"van der Waerden scores"`, or `"normal scores"`
`M`	number of blocks of samples being compared
`n.samples`	list of `M` vectors, each vector giving the sample sizes for each block of samples being compared
`nt`	vector of length `M` of total sample sizes involved in each of the `M` comparisons of k_i samples, respectively
`n.ties`	vector giving the number of ties in each the `M` comparison blocks
`qn.list`	list of `M` matrices giving the `qn` results from `qn.test`, applied to the samples in each of the `M` blocks
`qn.c`	2 (or 3) vector containing the observed QN_{comb}, asymptotic P-value, (simulated or exact P-value).
`warning`	logical indicator, `warning = TRUE` when at least one $n_{ij} < 5$.
`null.dist`	simulated or enumerated null distribution of the QN_{comb} statistic. It is `NULL` when `dist = FALSE` or when `method = "asymptotic"`.
`method`	The `method` used.
`Nsim`	The number of simulations used for each block of samples.

Note

These tests are useful in analyzing treatment effects of shift nature in randomized (incomplete) block experiments.

References

Lehmann, E.L. (2006), *Nonparametric, Statistical Methods Based on Ranks*, Springer Verlag, New York. Ch. 6, Sec. 5D.

See Also

`qn.test`

Examples

```
## Create two lists of sample vectors.
x1 <- list( c(1, 3, 2, 5, 7), c(2, 8, 1, 6, 9, 4), c(12, 5, 7, 9, 11) )
x2 <- list( c(51, 43, 31, 53, 21, 75), c(23, 45, 61, 17, 60) )
# and a corresponding data frame datx1x2
x1x2 <- c(unlist(x1),unlist(x2))
gx1x2 <- as.factor(c(rep(1,5),rep(2,6),rep(3,5),rep(1,6),rep(2,5)))
bx1x2 <- as.factor(c(rep(1,16),rep(2,11)))
datx1x2 <- data.frame(A = x1x2, G = gx1x2, B = bx1x2)

## Run qn.test.combined.
set.seed(2627)
qn.test.combined(x1, x2, method = "simulated", Nsim = 1000)
# or with same seed
# qn.test.combined(list(x1, x2), method = "simulated", Nsim = 1000)
# or qn.test.combined(A~G|B,data=datx1x2,method="simulated",Nsim=1000)
```

`ShorelineFireEMS` *Shoreline Fire and EMS Turnout Times*

Description

This data set gives turnout response times for Fire and EMS (Emergency Medical Services) dispatch calls to the Shoreline, WA, Fire Department in 2006. The turnout time refers to time elapsed between the emergency call dispatch and the crew leaving the fire station, or signaling that they are on their way while being on route already. The latter scenario may explain the bimodal distribution character.

Usage

```
data(ShorelineFireEMS)
```

Format

A list of two sublists `$EMSTOT` and `$FireTOT`, each with 4 vector components `$ST57`, `$ST63`, `$ST64`, and `$ST65` respectively, giving the turnout times (in seconds) (for EMS and Fire) at fire stations ST57, ST63, ST64, and ST65.

Note

These data sets are provided to illustrate usage of `ad.test` and `qn.test` and their combined versions in testing for performance equivalence across fire stations.

Source

Thanks to Michael Henderson and the Fire Fighters and Paramedics of the Shoreline Fire Department in Washington State.

Examples

```
data(ShorelineFireEMS)
boxplot(ShorelineFireEMS$EMSTOT,xlab="Station", ylab="seconds",
main="EMS Turnout Time")
boxplot(ShorelineFireEMS$FireTOT,xlab="Station", ylab="seconds",
main="Fire Turnout Time")
```

`Steel.test` *Steel's Multiple Comparison Wilcoxon Tests*

Description

This function uses pairwise Wilcoxon tests, comparing a common control sample with each of several treatment samples, in a multiple comparison fashion. The experiment wise significance probabity is calculated, estimated, or approximated, when testing the hypothesis that all independent samples arise from a common unspecified distribution, or that treatments have no effect when assigned randomly to the given subjects.

Usage

```
Steel.test(..., data = NULL,
method = c("asymptotic", "simulated", "exact"),
alternative = c("greater","less","two-sided"),
dist = FALSE, Nsim = 10000)
```

Arguments

`...`	Either several sample vectors, say $x_1, \ldots, x_k$, with x_i containing n_i sample values. $n_i > 4$ is recommended for reasonable asymptotic P-value calculation. The pooled sample size is denoted by $N = n_1 + \ldots + n_k$. The first vector serves as control sample, the others as treatment samples. or a list of such sample vectors. or a formula y ~ g, where y contains the pooled sample values and g (same length as y) is a factor with levels identifying the samples to which the elements of y belong. The lowest factor level corresponds to the control sample, the other levels to treatment samples.
`data`	= an optional data frame providing the variables in formula y ~ g or y, g input
`method`	= `c("asymptotic","simulated","exact")`, where `"asymptotic"` uses only an asymptotic normal approximation to approximate the P-value, This calculation is always done. `"simulated"` uses `Nsim` simulated standardized Steel statistics based on random splits of the pooled samples into samples of sizes $n_1, \ldots, n_k$, to estimate the P-value.

"exact" uses full enumeration of all sample splits with resulting standardized Steel statistics to obtain the exact P-value. It is used only when Nsim is at least as large as the number

$$ncomb = \frac{N!}{n_1! \ldots n_k!}$$

of full enumerations. Otherwise, method reverts to "simulated" using the given Nsim. It also reverts to "simulated" when $ncomb > 1e8$ and dist = TRUE.

alternative = c("greater","less","two-sided"), where for "greater" the maximum of the pairwise standardized Wilcoxon test statistics is used and a large maximum value is judged significant. For "less" the minimum of the pairwise standardized Wilcoxon test statistics is used and a low minimum value is judged significant. For "two-sided" the maximum of the absolute pairwise standardized Wilcoxon test statistics is used and a large maximum value is judged significant.

dist = FALSE (default) or TRUE. If TRUE, the simulated or fully enumerated null distribution vector null.dist is returned for the Steel test statistic, as chosen via alternative. Otherwise, NULL is returned. When dist = TRUE then Nsim <- min(Nsim, 1e8), to limit object size.

Nsim = 10000 (default), number of simulation sample splits to use. It is only used when method = "simulated", or when method = "exact" reverts to method = "simulated", as previously explained.

Details

The Steel criterion uses the Wilcoxon test statistic in the pairwise comparisons of the common control sample with each of the treatment samples. These statistics are used in standardized form, using the means and standard deviations as they apply conditionally given the tie pattern in the pooled data, see Scholz (2016). This conditional treatment allows for correct usage in the presence of ties and is appropriate either when the samples are independent and come from the same distribution (continuous or not) or when treatments are assigned randomly among the total of N subjects. However, in the case of ties the significance probability has to be viewed conditionally given the tie pattern.

The Steel statistic is used to test the hypothesis that the samples all come from the same but unspecified distribution function $F(x)$, or, under random treatment assigment, that the treatments have no effect. The significance probability is the probability of obtaining test results as extreme or more extreme than the observed test statistic, when testing for the possibility of a treatment effect under any of the treatments.

For small sample sizes exact (conditional) null distribution calculations are possible (with or without ties), based on a recursively extended version of Algorithm C (Chase's sequence) in Knuth (2011), which allows the enumeration of all possible splits of the pooled data into samples of sizes of $n_1, \ldots, n_k$, as appropriate under treatment randomization. This is done in C, as is the simulation of such splits.

NA values are removed and the user is alerted with the total NA count. It is up to the user to judge whether the removal of NA's is appropriate.

Value

A list of class kSamples with components

test.name	"Steel"
alternative	"greater", "less", or "two-sided"
k	number of samples being compared, including the control sample as the first one
ns	vector $(n_1, \ldots, n_k)$ of the k sample sizes
N	size of the pooled sample $= n_1 + \ldots + n_k$
n.ties	number of ties in the pooled sample
st	2 (or 3) vector containing the observed standardized Steel statistic, its asymptotic P-value, (its simulated or exact P-value)
warning	logical indicator, warning = TRUE when at least one $n_i < 5$
null.dist	simulated or enumerated null distribution vector of the test statistic. It is NULL when dist = FALSE or when method = "asymptotic".
method	the method used.
Nsim	the number of simulations used.
W	vector $(W_{12}, \ldots, W_{1k})$ of Mann-Whitney statistics comparing each observation under treatment $i(> 1)$ against each observation of the control sample.
mu	mean vector $(n_1 n_2/2, \ldots, n_1 n_k/2)$ of W.
tau	vector of standard deviations of W.
sig0	standard deviation used in calculating the significance probability of the maximum (minimum) of (absolute) standardized Mann-Whitney statistics, see Scholz (2016).
sig	vector $(\sigma_1, \ldots, \sigma_k)$ of standard deviations used in calculating the significance probability of the maximum (minimum) of (absolute) standardized Mann-Whitney statistics, see Scholz (2016).

warning

method = "exact" should only be used with caution. Computation time is proportional to the number of enumerations. Experiment with system.time and trial values for Nsim to get a sense of the required computing time. In most cases dist = TRUE should not be used, i.e., when the returned distribution objects become too large for R's work space.

References

Knuth, D.E. (2011), *The Art of Computer Programming, Volume 4A Combinatorial Algorithms Part 1*, Addison-Wesley

Lehmann, E.L. (2006), *Nonparametrics, Statistical Methods Based on Ranks, Revised First Edition*, Springer Verlag.

Scholz, F.W. (2023), "On Steel's Test with Ties", https://arxiv.org/abs/2308.05873

Examples

```
z1 <- c(103, 111, 136, 106, 122, 114)
z2 <- c(119, 100,  97,  89, 112,  86)
z3 <- c( 89, 132,  86, 114, 114, 125)
z4 <- c( 92, 114,  86, 119, 131,  94)
y <- c(z1, z2, z3, z4)
g <- as.factor(c(rep(1, 6), rep(2, 6), rep(3, 6), rep(4, 6)))
set.seed(2627)
Steel.test(list(z1, z2, z3, z4), method = "simulated",
  alternative = "less", Nsim = 1000)
# or with same seed
# Steel.test(z1, z2, z3, z4,method = "simulated",
#   alternative = "less", Nsim = 1000)
# or with same seed
# Steel.test(y ~ g, method = "simulated",
#   alternative = "less", Nsim=1000)
```

`SteelConfInt` *Simultaneous Confidence Bounds Based on Steel's Multiple Comparison Wilcoxon Tests*

Description

This function inverts pairwise Wilcoxon tests, comparing a common control sample with each of several treatment samples to provide simultaneous confidence bounds for the respective shift parameters by which the sampled treatment populations may differ from the control population. It is assumed that all samples are independent and that the sampled distributions are continuous to avoid ties. The joint coverage probability for all bounds/intervals is calculated, estimated, or approximated, see Details. For treatment of ties also see Details.

Usage

```
SteelConfInt(..., data = NULL, conf.level = 0.95,
alternative = c("less", "greater", "two.sided"),
      method = c("asymptotic", "exact", "simulated"), Nsim = 10000)
```

Arguments

`...`	Either several sample vectors, say $x_1, \ldots, x_k$, with x_i containing n_i sample values. $n_i > 4$ is recommended for reasonable asymptotic P-value calculation. The pooled sample size is denoted by $N = n_1 + \ldots + n_k$. The first vector serves as control sample, the others as treatment samples. or a list of such sample vectors. or a formula y ~ g, where y contains the pooled sample values and g (same length as y) is a factor with levels identifying the samples to which the elements of y belong. The lowest factor level corresponds to the control sample, the other levels to treatment samples.

`data` = an optional data frame providing the variables in formula y ~ g.

`conf.level` = `0.95` (default) the target joint confidence level for all bounds/intervals. `0 < conf.level < 1`.

`alternative` = `c("less", "greater", "two.sided")`, where `"less"` results in simultaneous upper confidence bounds for all shift parameters and any negative upper bound should lead to the rejection of the null hypothesis of all shift parameters being zero or positive in favor of at least one being less than zero.

`"greater"` results in simultaneous lower confidence bounds for all shift parameters and any positive lower bound should lead to the rejection of the null hypothesis of all shift parameters being zero or negative in favor of at least one being greater than zero.

`"two.sided"` results in simultaneous confidence intervals for all shift parameters and any interval not straddling zero should lead to the rejection of the null hypothesis of all shift parameters being zero in favor of at least one being different from zero.

`method` = `c("asymptotic", "exact", "simulated")`, where

`"asymptotic"` uses only an asymptotic normal approximation to approximate the achieved joint coverage probability. This calculation is always done.

`"exact"` uses full enumeration of all sample splits to obtain the exact achieved joint coverage probability (see Details). It is used only when `Nsim` is at least as large as the number of full enumerations. Otherwise, `method` reverts to `"simulated"` using the given `Nsim`.

`"simulated"` uses `Nsim` simulated random splits of the pooled samples into samples of sizes $n_1, \ldots, n_k$, to estimate the achieved joint coverage probability.

`Nsim` = `10000` (default), number of simulated sample splits to use. It is only used when `method = "simulated"`, or when `method = "exact"` reverts to `method = "simulated"`, as previously explained.

Details

The first sample is treated as control sample with sample size n_1. The remaining $s = k - 1$ samples are treatment samples. Let $W_{1i}, i = 2, \ldots, k$ denote the respective Wilcoxon statistics comparing the common control sample (index 1) with each of the s treatment samples (indexed by i). For each comparison of control and treatment i sample only the observations of the two samples involved are ranked. By $W_i = W_{1i} - n_i(n_i + 1)/2$ we denote the corresponding Mann-Whitney test statistic. Furthermore, let $D_{i(j)}$ denote the j-th ordered value (ascending order) of the $n_1 n_i$ paired differences between the observations in treatment sample i and those of the control sample. By simple extension of results in Lehmann (2006), pages 87 and 92, the following equations hold, relating the null distribution of the Mann-Whitney statistics and the joint coverage probabilities of the $D_{i(j_i)}$ for any set of $j_1, \ldots, j_s$ with $1 \le j_i \le n_1 n_i$.

$$P_\Delta(\Delta_i \le D_{i(j_i)}, i = 2, \ldots, k) = P_0(W_i \le j_i - 1, i = 2, \ldots, k)$$

and

$$P_\Delta(\Delta_i \ge D_{i(j_i)}, i = 2, \ldots, s) = P_0(W_i \le n_1 n_i - j_i, i = 2, \ldots, k)$$

where P_Δ refers to the distribution under $\Delta = (\Delta_2, \ldots, \Delta_k)$ and P_0 refers to the joint null distribution of the W_i when all sampled distributions are the same and continuous. There are $k - 1$

indices j_i that can be manipulated to affect the achieved confidence level. To limit the computational complexity standardized versions of the W_i, i.e., $(W_i - \mu_i)/\tau_i$ with μ_i and τ_i representing mean and standard deviation of W_i, are used to choose a common value for $(j_i - 1 - \mu_i)/\tau_i$ (satisfying the γ level) from the multivariate normal approximation for the W_i (see Miller (1981) and Scholz (2016)), and reduce that to integer values for j_i, rounding up, rounding down, and rounding to the nearest integer. These integers j_i are then used in approximating the actual joint probabilities $P_0(W_i \leq j_i - 1, i = 2, \ldots, k)$, and from these three coverage probabilities the one that is closest to the nominal confidence level γ and $\geq \gamma$ and also also the one that is closest without the restriction $\geq \gamma$ are chosen.

When `method = "exact"` or `= "simulated"` is specified, the same process is used, using either the fully enumerated exact distribution of $W_i, i = 2, \ldots, k$ (based on a recursive version of Chase's sequence as presented in Knuth (2011)) for all sample splits, or the simulated distribution of $W_i, i = 2, \ldots, k$. However, since these distributions are discrete the starting point before rounding up is the smallest quantile such that the proportion of distribution values less or equal to it is at least γ. The starting point before rounding down is the highest quantile such that the proportion of distribution values less or equal to it is at most γ. The third option of rounding to the closest integer is performed using the average of the first two.

Confidence intervals are constructed by using upper and lower confidence bounds, each with same confidence level of $(1 + \gamma)/2$.

When the original sample data appear to be rounded, and especially when there are ties, one should widen the computed intervals or bounds by the rounding ϵ, as illustrated in Lehmann (2006), pages 85 and 94. For example, when all sample values appear to end in one of .0, .2, .4, .6, .8, the rounding ϵ would be .2. Ultimately, this is a judgment call for the user. Such widening of intervals will make the actually achieved confidence level $\geq$ the stated achieved level.

Value

A list of class `kSamples` with components

`test.name`	`"Steel.bounds"`
`n1`	the control sample size $= n_1$
`ns`	vector $(n_2, \ldots, n_k)$ of the $s = k - 1$ treatment sample sizes
`N`	size of the pooled sample $= n_1 + \ldots + n_k$
`n.ties`	number of ties in the pooled sample
`bounds`	a list of data frames. When `method = "asymptotic"` is specified, the list consists of two data frames named `conservative.bounds.asymptotic` and `closest.bounds.asymptotic`. Each data frame consists of s rows corresponding to the s shift parameters Δ_i and three columns, the first column giving the lower bound, the second column the upper bound, while the first row of the third column states the computed confidence level by asymptotic approximation, applying jointly to all s sets of bounds. For one-sided bounds the corresponding other bound is set to `Inf` or `-Inf`, respectively. In case of `conservative.bounds.asymptotic` the achieved asymptotic confidence level is targeted to be $\geq$ `conf.level`, but closest to it among three possible choices (see Details). In the case of `closest.bounds.asymptotic` the achieved level is targeted to be closest to `conf.level`.

When `method = "exact"` or `method = "simulated"` is specified the list consists in addition of two further data frames named either

`conservative.bounds.exact` and `closest.bounds.exact` or

`conservative.bounds.simulated` and `closest.bounds.simulated`.

In either case the structure and meaning of these data frames parallels that of the `"asymptotic"` case.

`method` the `method` used.

`Nsim` the number of simulations used.

`j.LU` an s by 4 matrix giving the indices j_i used for computing the bounds $D_{i(j_i)}$ for $\Delta_i, i = 1, \ldots, s$.

warning

`method = "exact"` should only be used with caution. Computation time is proportional to the number of enumerations. Experiment with `system.time` and trial values for `Nsim` to get a sense of the required computing time.

References

Knuth, D.E. (2011), *The Art of Computer Programming, Volume 4A Combinatorial Algorithms Part 1*, Addison-Wesley

Lehmann, E.L. (2006), *Nonparametrics, Statistical Methods Based on Ranks, Revised First Edition*, Springer Verlag.

Miller, Rupert G., Jr. (1981), *Simultaneous Statistical Inference, Second Edition*, Springer Verlag, New York.

Scholz, F.W. (2023), "On Steel's Test with Ties", https://arxiv.org/abs/2308.05873

Examples

```
z1 <- c(103, 111, 136, 106, 122, 114)
z2 <- c(119, 100,  97,  89, 112,  86)
z3 <- c( 89, 132,  86, 114, 114, 125)
z4 <- c( 92, 114,  86, 119, 131,  94)
set.seed(2627)
SteelConfInt(list(z1,z2,z3,z4),conf.level=0.95,alternative="two.sided",
   method="simulated",Nsim=10000)
# or with same seed
# SteelConfInt(z1,z2,z3,z4,conf.level=0.95,alternative="two.sided",
#   method="simulated",Nsim=10000)
```

Index