Package 'NSM3'

November 4, 2024

Type Package

Version 1.19

Date 2024-11-03

Title Functions and Datasets to Accompany Hollander, Wolfe, and Chicken - Nonparametric Statistical Methods, Third Edition

Description Designed to replace the tables which were in the back of the first two editions of Hollander and Wolfe - Nonparametric Statistical Methods. Exact procedures are performed when computationally possible. Monte Carlo and Asymptotic procedures are performed otherwise. For those procedures included in the base packages, our code simply provides a wrapper to standardize the output with the other procedures in the package.

License GPL-2

LazyLoad yes

Depends R (>= 2.10), combinat, MASS, partitions, stats, survival

Imports agricolae, ash, binom, BSDA, coin, fANCOVA, gtools, Hmisc, km.ci, metafor, nortest, np, quantreg, Rfit, SuppDists, waveslim

NeedsCompilation yes

Repository CRAN

Author Grant Schneider [aut, cre], Eric Chicken [aut], Rachel Becvarik [aut]

Maintainer Grant Schneider <gschneidz@gmail.com>

Date/Publication 2024-11-04 16:40:06 UTC

Contents

`cAnsBrad` *Function to compute a critical value for the Ansari-Bradley C distribution.*

Description

This function uses pAnsari and qAnsari from the base stats package to compute the critical value for the Ansari-Bradley C distribution at (or typically in the "Exact" case, close to) the given alpha level. The program is reasonably quick for large data, well after the asymptotic approximation suffices, so Monte Carlo methods are not included.

Usage

```
cAnsBrad(alpha, m, n, method = NA, n.mc = 10000)
```

Arguments

alpha	A numeric value between 0 and 1.
m	A numeric value indicating the size of the first data group (X).
n	A numeric value indicating the size of the second data group (Y).
method	Either "Exact" or "Asymptotic", indicating the desired distribution. When method=NA, if m+n<=200, the "Exact" method will be used to compute the C distribution. Otherwise, the "Asymptotic" method will be used.
n.mc	Not used. Only included for standardization with other critical value procedures in the NSM3 package.

Value

Returns a list with "NSM3Ch5c" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact")
cutoff.L	lower tail cutoff at or below user-specified alpha
true.alpha.L	true alpha level corresponding to cutoff.L (if method="Exact")

Author(s)

Grant Schneider

References

This function uses the source code ansari.c from the stats package by: R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

See Also

Also see `ansari.test()`

Examples

```
##Hollander, Wolfe, Chicken - NSM3 - Example 5.1 (Serum Iron Determination):
cAnsBrad(0.05,20,20,"Asymptotic")
cAnsBrad(0.05,20,20,"Exact")

##Bigger data
cAnsBrad(0.05,100,100,"Exact")
```

cBohnWolfe *Function to compute a critical value for the Bohn-Wolfe U distribution.*

Description

This function uses Monte Carlo sampling to compute the critical value for the Bohn-Wolfe U distribution at (or close to) the given alpha level. The Monte Carlo samples are simulated based on the order statistics of a uniform(0,1) distribution.

Usage

```
cBohnWolfe(alpha,k,q,c,d,method="Monte Carlo",n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
k	A numeric value indicating the set size of the first data group in the RSS (X).
q	A numeric value indicating the set size of the second data group in the RSS (Y).
c	A numeric value indicating the number of cycles for the first data group in the RSS (X).
d	A numeric value indicating the number of cycles for the second data group in the RSS (Y).
method	For this procedure, method is currently set automatically to "Monte Carlo" as the only option that is available. For standardization with other critical value procedures in the NSM3 package, "Asymptotic" and "Exact" will be supported in future versions.
n.mc	Number of Monte Carlo samples used to estimate the distribution of U.

Value

Returns a list with "NSM3Ch5c" class containing the following components:

m	number of observations in RSS for the first data group (X)
n	number of observations in RSS for the second data group (Y)
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U

Author(s)

Grant Schneider

References

Bohn, Lora L., and Douglas A. Wolfe. "Nonparametric two-sample procedures for ranked-set samples data." Journal of the American Statistical Association 87.418 (1992): 552-561.

Examples

```
cBohnWolfe(.0515,4,4,5,5)
cBohnWolfe(.0303,2,3,3,3)
```

---

cDurSkiMa *Computes a critical value for the Durbin, Skillings-Mack D distribution.*

Description

This function computes the critical value for the Durbin, Skillings-Mack D distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cDurSkiMa(alpha,obs.mat, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
obs.mat	The incidence matrix, explained below.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The incidence matrix, obs.mat, will be an n x k matrix of ones and zeroes, which indicate where the data are observed and unobserved, respectively. Methods for finding the incidence matrix for various BIBD designs are given in the literature. While the incidence matrix will not be unique for a given (k, n, s, lambda, p) combination, the distribution of D under H0 will be the same.

Value

Returns a list with "NSM3Ch7c" class containing the following components:

k	number of treatments
n	number of blocks
ss	number of treatments per block
pp	number of observations per treatment
lambda	number of times each pair of treatments occurs together within a block
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Note

The syntax of this procedure differs from the others in the NSM3 package due to the fact that creating a BIBD for a given k,n,s,p,lambda is not trivial. We therefore require obs.mat, the incidence matrix.

Author(s)

Grant Schneider

Examples

```
##Hollander, Wolfe, Chicken Chapter 7, comment 49
obs.mat<-matrix(c(1,1,0,1,0,1,0,1,1),ncol=3,byrow=TRUE)
cDurSkiMa(.75,obs.mat)
```

---

cFligPoli *Computes a critical value for the Fligner-Policello U distribution.*

Description

This function computes the critical value for the Fligner-Policello U distriburion at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cFligPoli(alpha,m,n,method=NA,n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
m	A numeric value indicating the size of the first data group (X).
n	A numeric value indicating the size of the second data group (Y).
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch5c" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Chapter 4 example Hollander-Wolfe-Chicken##
cFligPoli(.0504,8,7)
cFligPoli(.101,8,7)
```

`cFrd` *Computes a critical value for the Friedman, Kendall-Babington Smith S distribution.*

Description

This function computes the critical value for the Friedman, Kendall-Babington Smith S distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level. The method used to compute the distribution is from the reference by Van de Wiel, Bucchianico, and Van der Laan.

Usage

```
cFrd(alpha, k, n, method=NA, n.mc=10000, return.full.distribution=FALSE)
```

Arguments

`alpha`	A numeric value between 0 and 1.
`k`	A numeric value indicating the number of treatments.
`n`	A numeric value indicating the number of blocks.
`method`	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
`n.mc`	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.
`return.full.distribution`	If TRUE, and the method used is not asymptotic, the entire probability mass function of S will be returned.

Value

Returns a list with "NSM3Ch7c" class containing the following components:

`k`	number of treatments
`n`	number of blocks
`cutoff.U`	upper tail cutoff at or below user-specified alpha
`true.alpha.U`	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")
`full.distribution`	probability mass function of S

Author(s)

Grant Schneider

References

Van de Wiel, M. A., A. Di Bucchianico, and P. Van der Laan. "Symbolic computation and exact distributions of nonparametric test statistics." Journal of the Royal Statistical Society: Series D (The Statistician) 48.4 (1999): 507-516.

See Also

The `coin` package.

Examples

```
##Hollander-Wolfe-Chicken Example 7.1 Rounding First Base
#cFrd(0.01,3,22,"Exact")
cFrd(0.01,3,22,n.mc=5000)
cFrd(0.01,3,22,"Asymptotic")
```

---

ch.ro *Campbell-Hollander*

Description

Function to compute the Campbell-Hollander estimator G-hat

Usage

```
ch.ro (x,n,alpha,mu,...)
```

Arguments

`x`	a vector of data of length r
`n`	the sample size
`alpha`	the degrees of confidence in mu
`mu`	the prior guess of the unknown P (a pdf)
`...`	all of the arguments needed for mu

Value

`G.hat`	estimate of the rank order G

Author(s)

Rachel Becvarik

References

See Section 16.3 of Hollander, Wolfe, Chicken - Nonparametric Statistical Methods 3.

Examples

```
##Hollander-Wolfe-Chicken Example 16.2 Swimming in the Women's 50 yard Freestyle
freestyle<-c(22.43, 21.88, 22.39, 22.78, 22.65, 22.60)
ch.ro(freestyle,12,10,pnorm,22.52,.24)
```

`cHayStón` *Computes a critical value for the Hayter-Stone W* distribution.*

Description

This function computes the critical value for the Hayter-Stone W* distriburion at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cHayStón(alpha,n, method=NA, n.mc=10000)
```

Arguments

`alpha`	A numeric value between 0 and 1.
`n`	A vector (of length 2 or greater) indicating the sizes of the data groups.
`method`	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
`n.mc`	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The Asymptotic distribution requires that all group sizes are equal. If method="Asymptotic" and there are different group sizes in n, method="Monte Carlo" will be used.

Value

Returns a list with "NSM3Ch6MCc" class containing the following components:

`n`	data group sizes
`num.comp`	number of multiple comparisons to be made (based on the length of n)
`cutoff.U`	upper tail cutoff at or below user-specified alpha
`true.alpha.U`	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 6.7 Motivational Effect of Knowledge of Performance:
#cHayston(.0553,rep(6,3),"Monte Carlo")
cHayston(.05,c(6,6,6),"Asymptotic")
```

---

cHayStonLSA — *Computes a critical value for the Hayter-Stone W* asymptotic distribution.*

Description

This function computes the critical value for the Hayter-Stone W* asymptotic distriburion at the given alpha level.

Usage

```
cHayStonLSA(alpha,k,delta=.001)
```

Arguments

alpha	A numeric value between 0 and 1.
k	A numeric value indicating the number of the data groups (with assumed equal sizes).
delta	Increment used to create the grid on which the distribution will be approximated.

Details

The Asymptotic distribution requires that all (unspecified) group sizes are equal.

Value

Returns the cutoff (based on the specified grid) with upper tail probability nearest to alpha.

Author(s)

Grant Schneider

References

Hayter, Anthony J., and Wei Liu. "Exact calculations for the one-sided studentized range test for testing against a simple ordered alternative." Computational statistics & data analysis 22.1 (1996): 17-25.

Examples

```
##Hollander-Wolfe-Chicken Example 6.7 Motivational Effect of Knowledge of Performance:
cHayStonLSA(.0553,3,delta=0.01)

##Section preceding Example 6.7 (explaining LSA)
cHayStonLSA(.05,6,delta=0.01)
```

`cHollBivSym` *Hollander Bivariate Symmetry*

Description

Quantile function for the Hollander A distribution.

Usage

```
cHollBivSym(alpha,d.mat,method=NA, n.mc=10000)
```

Arguments

`alpha`	A numeric value between 0 and 1.
`d.mat`	The d matrix, explained below.
`method`	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used. As Kepner and Randles (1984) and Hilton and Gee (1997) have found the large sample approximation to perform poorly, method="Asymptotic" will be treated as method=NA.
`n.mc`	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The d matrix, d.mat, will be an n*n matrix of ones and zeroes, where the (i,j)th element is 1 if min(Xj,Yj)<max(Xi,Yi)<=max(Xj,Yj) and min(Xi,Yi)<=min(Xj,Yj), 0 otherwise. An illustration may be found in the example section of this document and Section 3.10 of Hollander, Wolfe, and Chicken - NSM3.

Value

Returns a list with "NSM3Ch5c" class containing the following components:

`m`	number of observations in the first data group (X)
`n`	number of observations in the second data group (Y) (equal to m, but included for standardization with other procedures)
`cutoff.U`	upper tail cutoff at or below user-specified alpha
`true.alpha.U`	true alpha level corresponding to cutoff.U

Author(s)

Grant Schneider

References

Kepner, James L., and Ronald H. Randies. "Comparison of tests for bivariate symmetry versus location and/or scale alternatives." Communications in Statistics-Theory and Methods 13.8 (1984): 915-930.

Hilton, Joan F., and Lauren Gee. "The size and power of the exact bivariate symmetry test." Computational statistics & data analysis 26.1 (1997): 53-69.

Examples

```
##Hollander-Wolfe-Chicken Example 3.11 Insulin Clearance in Kidney Transplants
x<-c(61.4,63.3,63.7,80,77.3,84,105)
y<-c(70.8,89.2,65.8,67.1,87.3,85.1,88.1)
obs.data<-cbind(x,y)
a.vec<-apply(obs.data,1,min)
b.vec<-apply(obs.data,1,max)
test<-function(r,c) {as.numeric((a.vec[c]<b.vec[r])&&(b.vec[r]<=b.vec[c])&&(a.vec[r]<=a.vec[c]))}
myVecFun <- Vectorize(test,vectorize.args = c('r','c'))

d.mat<-outer(1:length(x), 1:length(x), FUN=myVecFun)

##Cutoff based on the exact distribution
cHollBivSym(.10,d.mat)
```

---

cJCK *Computes a critical value for the Jonckheere-Terpstra J distribution.*

Description

This function computes the critical value for the Jonckheere-Terpstra J distribution at (or typically in the "Exact" case, close to) the given alpha level. The function takes advantage of Harding's (1984) algorithm to quickly generate the distribution.

Usage

```
cJCK(alpha, n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
n	A vector of numeric values indicating the size of each of the k data groups.
method	Either "Exact" or "Asymptotic", indicating the desired distribution. When method=NA, if sum(n)<=200, the "Exact" method will be used to compute the J distribution. Otherwise, the "Asymptotic" method will be used.

n.mc	Not used. Only included for standardization with other critical value procedures in the NSM3 package.

Value

Returns a list with "NSM3Ch6c" class containing the following components:

n	number of observations in the k data groups
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact")

Author(s)

Grant Schneider

References

Harding, E. F. "An efficient, minimal-storage procedure for calculating the Mann-Whitney U, generalized U and similar distributions." Applied statistics (1984): 1-6.

Examples

```
##Hollander-Wolfe-Chicken Example 6.2 Motivational Effect of Knowledge of Performance
cJCK(.0490, c(6,6,6),"Exact")
cJCK(.0490, c(6,6,6),"Monte Carlo")
cJCK(.0231, c(6,6,6),"Exact")
```

---

cKolSmirn *Computes a critical value for the Kolmogorov-Smirnov J distribution.*

Description

This function uses pSmirnov2x from the base stats package to compute the critical value for the Kolmogorov-Smirnov J distribution at (or typically in the "Exact" case, close to) the given alpha level. The program is reasonably quick for large data, well after the asymptotic approximation suffices, so Monte Carlo methods are not included.

Usage

```
cKolSmirn(alpha, m, n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
m	A numeric value indicating the size of the first data group (X).
n	A numeric value indicating the size of the second data group (Y).
method	Either "Exact" or "Asymptotic", indicating the desired distribution. When method=NA, if m+n<=200, the "Exact" method will be used to compute the J distribution. Otherwise, the "Asymptotic" method will be used.
n.mc	Not used. Only included for standardization with other critical value procedures in the NSM3 package.

Value

Returns a list with "NSM3Ch5c" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact")

Author(s)

Grant Schneider

References

This function uses the source code ks.c from the stats package by: R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

See Also

Also see `ks.test()`.

Examples

```
##Hollander-Wolfe-Chicken Example 5.4 Effect of Feedback on Salivation Rate:
cKolSmirn(0.0524,10,10,"Exact")

##or
cKolSmirn(0.06,10,10,"Exact")

##LSA
cKolSmirn(0.0551,10,10,"Asymptotic")
```

cKW *Computes a critical value for the Kruskal-Wallis H distribution.*

Description

This function computes the critical value for the Kruskal-Wallis H distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cKW(alpha,n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
n	A vector of numeric values indicating the size of each of the k data groups.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch6c" class containing the following components:

n	number of observations in the k data groups
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 6.1 Half-Time of Mucociliary Clearance
#cKW(0.0503,c(5,4,5),"Exact")
cKW(0.7147,c(5,4,5),"Asymptotic")
cKW(0.7147,c(5,4,5),"Monte Carlo",n.mc=20000)
```

cLepage *Computes a critical value for the Lepage D distribution.*

Description

This function computes the critical value for the Lepage D distriburion at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cLepage(alpha, m, n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
m	A numeric value indicating the size of the first data group (X).
n	A numeric value indicating the size of the second data group (Y).
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch5c" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 5.3 Platelet Counts of Newborn Infants
cLepage(0.02,10,6,"Exact")
cLepage(0.02,10,6,"Monte Carlo")
cLepage(0.02,10,6,"Asymptotic")
```

`cMackSkil` *Computes a critical value for the Mack-Skillings MS distribution.*

Description

This function computes the critical value for the Mack-Skillings MS distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cMackSkil(alpha,k,n,c, method=NA, n.mc=10000)
```

Arguments

`alpha`	A numeric value between 0 and 1.
`k`	A numeric value indicating the number of treatments.
`n`	A numeric value indicating the number of blocks.
`c`	A numeric value indicating the number of replications for each treatment-block combination.
`method`	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
`n.mc`	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch7c" class containing the following components:

`k`	number of treatments
`n`	number of blocks
`c`	number of replications
`cutoff.U`	upper tail cutoff at or below user-specified alpha
`true.alpha.U`	true alpha level corresponding to cutoff.U (if method="Exact")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 7.9 Determination of Niacin in Bran Flakes
cMackSkil(.0501,4,3,3)
##Hollander-Wolfe-Chicken Chapter 7 Comment 72
cMackSkil(.0502,4,4,3)
```

cMaxCorrNor *Quantile function for the maximum of k N(0,1) random variables with common correlation rho.*

Description

Uses the integrate function based on the method proposed in Gupta, Panchapakesan and Sohn (1983).

Usage

```
cMaxCorrNor(alpha,k,rho)
```

Arguments

alpha	A numeric value between 0 and 1.
k	Number of random variables.
rho	Common correlation between the random variables.

Value

Returns the upper tail cutoff at or immediately below the user-specified alpha.

Author(s)

Grant Schneider

References

Gupta, Shanti S., S. Panchapakesan, and Joong K. Sohn. "On the distribution of the studentized maximum of equally correlated normal random variables." Communications in Statistics-Simulation and Computation 14.1 (1985): 103-135.

Examples

```
##Hollander-Wolfe-Chicken Section 7.4 LSA
cMaxCorrNor(.04584,4,.5)
##Hollander-Wolfe-Chicken Section 7.14
cMaxCorrNor(.02337,5,.3)
##Hollander-Wolfe-Chicken Example 7.14
cMaxCorrNor(.10,5,.452)
```

cNDWol *Function to compute a critical value for the Nemenyi, Damico-Wolfe Y distribution.*

Description

This function computes the critical value for the Nemenyi, Damico-Wolfe Y distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cNDWol(alpha,n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
n	A vector of numeric values indicating the size of each of the k data groups, with the first element indicating the treatment group size.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch6MCc" class containing the following components:

n	number of observations in the k data groups
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 6.8 Motivational Effect of Knowledge of Performance
cNDWol(.0554, c(6, 6, 6),"Monte Carlo")
cNDWol(.0554, c(6, 6, 6),"Monte Carlo",n.mc=25000)
cNDWol(.0371, c(6, 6, 6),"Monte Carlo")
```

cNWWM *Computes a critical value for the Nemenyi, Wilcoxon-Wilcox, Miller R* distribution.*

Description

This function computes the critical value for the Nemenyi, Wilcoxon-Wilcox, Miller R* distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cNWWM(alpha, k, n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
k	A numeric value indicating the number of treatments.
n	A numeric value indicating the number of blocks.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch7c" class containing the following components:

k	number of treatments
n	number of blocks
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 7.4 Stuttering Adaptation
#cNWWM(.0492, 3, 18, "Monte Carlo")
cNWWM(.0492, 3, 18, method="Monte Carlo",n.mc=2500)
##Comment 7.35
cNWWM(.0093, 3, 3, "Exact")
#cNWWM(.0093, 3, 3, "Monte Carlo")
```

CorrUpperBound *Computes the upper bound for the null correlation between two overlapping signed rank statistics.*

Description

This function is based on the computations in Hollander (1967).

Usage

```
CorrUpperBound(n)
```

Arguments

n	number of observations

Value

Returns a numeric value indicating the upper bound.

Author(s)

Grant Schneider

References

Hollander, Myles. "Rank tests for randomized blocks when the alternatives have an a priori ordering." The Annals of Mathematical Statistics (1967): 867-877.

Examples

```
##Hollander-Wolfe-Chicken Example 7.12 Effect of Weight on Forearm Tremor Frequency
CorrUpperBound(6)
```

cPage *Function to compute a critical value for the Page L distribution.*

Description

This function computes the critical value for the Page L distriburion at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cPage(alpha, k, n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
k	A numeric value indicating the number of treatments.
n	A numeric value indicating the number of blocks.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch7c" class containing the following components:

k	number of treatments
n	number of blocks
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 7.2 Breaking Strength of Cotton Fibers
#cPage(.0097, 5, 3,"Exact")
cPage(.0097, 5, 3,"Monte Carlo")
```

---

cRangeNor *Quantile function for the range of k independent N(0,1) random variables.*

Description

Uses the integrate function based on the method proposed in Harter (1960).

Usage

```
cRangeNor(alpha,k)
```

Arguments

alpha	A numeric value between 0 and 1.
k	Number of independent Normal random variables.

Value

Returns the upper tail cutoff at or immediately below the user-specified alpha.

Author(s)

Grant Schneider

References

Harter, H. Leon. "Tables of range and studentized range." The Annals of Mathematical Statistics (1960): 1122-1147.

Examples

```
##Hollander-Wolfe-Chicken Example 7.3 Rounding First Base
cRangeNor(.01, 3)

##Hollander-Wolfe-Chicken Example 7.7 Chemical Toxicity
cRangeNor(.05, 7)
```

---

cSDCFlig — *Computes a critical value for the Dwass, Steel, Critchlow-Fligner W distribution.*

Description

This function computes the critical value for the Dwass, Steel, Critchlow-Fligner W distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cSDCFlig(alpha, n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
n	A vector of numeric values indicating the size of each of the k data groups.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch6c" class containing the following components:

n	number of observations in the k data groups
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Chapter 6 Comment 55
#cSDCFlig(.0331, c(3, 5, 7),n.mc=10000)
cSDCFlig(.0331, c(3, 5, 7),n.mc=2500)

##Another example
#cSDCFlig(alpha=0.05,n=rep(4,3),method="Exact")
cSDCFlig(alpha=0.05,n=rep(4,3),method="Monte Carlo",n.mc=2500)
#cSDCFlig(alpha=0.05,n=rep(4,3),method="Asymptotic")
```

---

cSkilMack *Computes a critical value for the Skillings-Mack SM distribution.*

Description

This function computes the critical value for the Skillings-Mack SM distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cSkilMack(alpha, obs.mat, method = NA, n.mc = 10000)
```

Arguments

alpha	A numeric value between 0 and 1.
obs.mat	The incidence matrix, explained below.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The incidence matrix, obs.mat, will be an n x k matrix of ones and zeroes, which indicate where the data are observed and unobserved, respectively.

Value

Returns a list with "NSM3Ch7c" class containing the following components:

k	number of treatments
n	number of blocks
ss	number of treatments per block
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Note

The syntax of this procedure differs from the others in the NSM3 package due to the fact that the distribution is calculated conditionally on the pattern of missingness. We therefore require obs.mat, the incidence matrix.

Author(s)

Grant Schneider

Examples

```
##Hollander, Wolfe, Chicken Example 7.8 Effect of Rhythmicity of a Metronome on Speech Fluency
obs.mat<-matrix(c(rep(1,10),0,rep(1,13)),ncol=3,byrow=TRUE)
#cSkilMack(.01,obs.mat)
cSkilMack(.01,obs.mat,n.mc=5000)
```

---

cUmbrPK *Computes a critical value for the Mack-Wolfe Peak Known A_p distribution.*

Description

This function computes the critical value for the Mack-Wolfe Peak Known A_p distribution at (or typically in the "Exact" case, close to) the given alpha level. The function generalizes Harding's (1984) algorithm to quickly generate the distribution.

Usage

```
cUmbrPK(alpha, n, peak=NA, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
n	A vector of numeric values indicating the size of each of the k data groups.
peak	An integer representing the known peak among the data groups.
method	Either "Exact" or "Asymptotic", indicating the desired distribution. When method=NA, if sum(n)<=200, the "Exact" method will be used to compute the A_p distribution. Otherwise, the "Asymptotic" method will be used.
n.mc	Not used. Only included for standardization with other critical value procedures in the NSM3 package.

Value

Returns a list with "NSM3Ch6c" class containing the following components:

n	number of observations in the k data groups
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact")

Author(s)

Grant Schneider

References

Harding, E. F. "An efficient, minimal-storage procedure for calculating the Mann-Whitney U, generalized U and similar distributions." Applied statistics (1984): 1-6.

Examples

```
##Hollander-Wolfe-Chicken Example 6.3 Fasting Metabolic Rate of White-Tailed Deer
cUmbrPK(.0101, c(7, 3, 5, 4, 4,3), peak=4)
```

---

cUmbrPU — *Computes a critical value for the Mack-Wolfe Peak Unknown A_p-hat distribution.*

Description

This function computes the critical value for the Mack-Wolfe Peak Unknown A_p-hat distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cUmbrPU(alpha, n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
n	A vector of numeric values indicating the size of each of the k data groups.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch6c" class containing the following components:

n	number of observations in the k data groups
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 6.4 Learning Comprehension and Age
#cUmbrPU(.0495, c(3, 3, 3, 3, 3))

cUmbrPU(.10, c(2, 4, 2))
```

---

cWNMT *Computes a critical value for the Wilcoxon, Nemenyi, McDonald-Thompson R distribution.*

Description

This function computes the critical value for the Wilcoxon, Nemenyi, McDonald-Thompson R distribution at (or typically in the "Exact" and "Monte Carlo" cases, close to) the given alpha level.

Usage

```
cWNMT(alpha, k, n, method=NA, n.mc=10000)
```

Arguments

alpha	A numeric value between 0 and 1.
k	A numeric value indicating the number of treatments.
n	A numeric value indicating the number of blocks.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch7c" class containing the following components:

k	number of treatments
n	number of blocks
cutoff.U	upper tail cutoff at or below user-specified alpha
true.alpha.U	true alpha level corresponding to cutoff.U (if method="Exact" or "Monte Carlo")

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 7.3 Rounding First Base
#cWNMT(.047, 3, 15)
cWNMT(.047, 3, 15,n.mc=5000)

##Chapter 7 Comment 26
#cWNMT(.083, 4, 2)
cWNMT(.083, 4, 2,n.mc=5000)
```

---

data *Dataset*

Description

These are the datasets used in the Examples of Hollander, Wolfe, and Chicken - Nonparametric Statistical Methods Third Edition. More extensive details about the data may be found there.

Usage

```
data(rhythmicity)
```

Format

The format varies depending on the dataset.

Source

Hollander, Wolfe, and Chicken - Nonparametric Statistical Methods, Third Edition

Examples

```
data(rhythmicity)
data(forearm)
```

---

dmrl.mc *Hollander-Proschan*

Description

Function to compute the Monte Carlo or asymptotic P-value for the observed Hollander-Proschan V' statistic.

Usage

```
dmrl.mc(x, alternative = "two.sided", exact=FALSE,
        min.reps = 100, max.reps = 1000, delta = 10^-3)
```

Arguments

x	a vector of data of length n
alternative	the direction of the alternative hypothesis. The choices are two.sided, dmrl, and imrl with the default value being two.sided.
exact	TRUE/FALSE value that determines whether the exact test or the large sample approximation is used if n >= 9. If n < 9 the exact test is used. The default value is FALSE, so the large sample approximation will be used unless specified not to. This is the same large sample approximation as epstein()
min.reps	the minimum number of repetitions for the Monte Carlo Approximation
max.reps	the maximum number of reps for the Monte Carlo Approximation. If the maximum number of reps has been reached, and the probability has not converged, a warning is given.
delta	the measure of accuracy for the convergence. If the probability converges to within delta, the Monte Carlo procedure stops before reaching the maximum number of reps.

Value

The function returns a list with two elements:

V	the value of the dmrl statistic
p	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
ex11.1<-c(42, 43, 51, 61, 66, 69, 71, 81, 82, 82)
dmrl.mc(ex11.1, alt="dmrl", exact=TRUE)
```

---

e.mc — *Function to compute the Monte Carlo P-value for the observed Epstein E statistic*

Description

This is the Monte Carlo approximation to the function "epstein".

Usage

```
e.mc(x, alternative = "two.sided", exact=FALSE,
     min.reps = 1000, max.reps = 10000, delta = 10^-4)
```

Arguments

x	a vector of data of length n
alternative	the direction of the alternative hypothesis. The choices are two.sided, ifr and dfr with the default value being two.sided.
exact	TRUE/FALSE value that determines whether the exact test or the large sample approximation is used if n >= 9. If n < 9 the exact test is used. The default value is FALSE, so the large sample approximation will be used unless specified not to. This is the same large sample approximation as epstein()
min.reps	the minimum number of repetitions for the Monte Carlo Approximation
max.reps	the maximum number of reps for the Monte Carlo Approximation. If the maximum number of reps has been reached, and the probability has not converged, a warning is given.
delta	the measure of accuracy for the convergence. If the probability converges to within delta, the Monte Carlo procedure stops before reaching the maximum number of reps.

Value

The function returns a list with two elements:

E	the value of the Epstein statistic
p	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
ex11.1<-c(42, 43, 51, 61, 66, 69, 71, 81, 82, 82)
Ep <- e.mc(ex11.1, alt="ifr", exact=TRUE)
Ep$E
Ep$p

#Large Sample Approximation
Ep.lsa <- e.mc(ex11.1, alt="ifr")

table11.2<-c(487, 18, 100, 7, 98, 5, 85, 91, 43, 230, 3, 130)
Ep=e.mc(table11.2,alt="i", exact=TRUE)
#Failing to converge
Ep=e.mc(table11.2,alt="i", exact=TRUE, min.reps=5, max.reps=5)
```

---

ecdf.ks.CI *Kolmogorov's Confidence Band*

Description

Function to compute and plot Kolmogorov's 95% confidence band for the distribution function F(x). This code is adapted from the code by Kjetil Halvorsen found at: https://stat.ethz.ch/pipermail/r-help/2003-July/036643.html

Usage

```
ecdf.ks.CI(x, main = NULL, sub = NULL, xlab = deparse(substitute(x)), ...)
```

Arguments

x	a vector of data of length n
main	the title of the plot. The default is ecdf(x) + 95% K.S.Bands
sub	subtitle, as used in the function plot()
xlab	the label for the x-axis of the plot. The default is x.
...	any additional plotting options

Value

The function returns a list with three elements:

lower	the values of the lower part of the confidence band
upper	the values of the upper part of the confidence band
D	the value of Kolmogorov's D statistic

Note

This function also plots the confidence bands.

Author(s)

Rachel Becvarik

Examples

```
methyl<-c(42, 43, 51, 61, 66, 69, 71, 81, 82, 82)
ecdf.ks.CI(methyl)

ecdf.ks.CI(methyl, lwd=2, main="KS Confidence Bands")
```

---

epstein *Epstein*

Description

Function to compute the P-value for the observed Epstein E statistic

Usage

```
epstein(x, alternative = "two.sided", exact=FALSE)
```

Arguments

x	a vector of data of length n
alternative	the direction of the alternative hypothesis. The choices are two.sided, ifr (for increasing failure rate) and dfr (for decreasing failure rate) with the default value being two.sided.
exact	TRUE/FALSE value that determines whether the exact test or the large sample approximation is used if n >= 9. If n < 9 the exact test is used. The default value is FALSE, so the large sample approximation will be used unless specified not to.

Value

The function returns a list with two elements:

E	the value of the Epstein statistic
p	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
ex11.1<-c(42, 43, 51, 61, 66, 69, 71, 81, 82, 82)
Ep <- epstein(ex11.1, alt="ifr", exact=TRUE)
Ep$E
Ep$p

#Large Sample Approximation
Ep.lsa <- epstein(ex11.1, alt="ifr")
```

---

ferg.df *Ferguson's Estimator*

Description

Function to compute an approximation of Ferguson's estimator mu_n.

Usage

```
ferg.df(x, alpha, mu, npoints,...)
```

Arguments

x	a vector of data of length n
alpha	the degree of confidence in mu
mu	the prior guess of the unknown P (a pdf)
npoints	the number of estimated points returned
...	all of the arguments needed for mu

Value

The function returns a vector of length num.points for Ferguson's estimator.

Author(s)

Rachel Becvarik

References

See Section 16.2 of Hollander, Wolfe, Chicken - Nonparametric Statistical Methods 3.

Examples

```
##Hollander-Wolfe-Chicken Figure 16.2
framingham<-c(2273, 2710, 141, 4725, 5010, 6224, 4991, 458, 1587, 1435, 2565, 1863)
plot.ecdf(framingham)
lines(sort(framingham),pexp(sort(framingham), 1/2922), lty=3)
temp.x = seq(min(framingham), max(framingham), length.out=100)
lines(temp.x,ferg.df(sort(framingham), 4, npoints=100,pexp,1/2922), col=2, type="s", lty=2)
legend("bottomright",  lty=c(1,3,2), legend=c("ecdf", "prior", "ferguson"), col=c(1,1,2))
```

HoeffD *Function to compute Hoeffding's D statistic for small sample sizes.*

Description

This will calculate Hoeffding's D statistic following section 8.6 of Hollander, Wolfe & Chicken, Nonparametric Statistical Methods, 3e. Uses the correction for ties given at (8.92).

Usage

```
HoeffD(x, y, example=FALSE)
```

Arguments

x	first data vector
y	second data vector
example	if true, analyzes the data from Example 8.6

Note

This function is intended for small sample sizes n only. For large n, use the asymptotic equivalence of D to the Blum-Kliefer-Rosenblatt statistic in the R package "Hmisc", command "hoeffd".

Author(s)

Eric Chicken

Examples

```
##Example 8.6 Hollander-Wolfe-Chicken##
HoeffD(example=TRUE)
```

HollBivSym *Hollander Bivariate Symmetry*

Description

Function to compute the Hollander A statistic for testing bivariate symmetry.

Usage

```
HollBivSym(x,y=NULL)
```

Arguments

x	Either a matrix containing both groups of data or a vector containing the first group of data.
y	If x is a vector, y is a required vector containing the second group of data. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the data can be entered in either of two ways. The following are equivalent:

`HollBivSym(x=matrix(c(1,2,3,4,5,6),ncol=2,byrow=T))` `HollBivSym(x=c(1,3,5),y=c(2,4,6))`

Value

Returns the observed Hollander A statistic.

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Table 3.16 example
recipient<-c(61.4,63.3,63.7,80,77.3,84,105)
donor<-c(70.8,89.2,65.8,67.1,87.3,85.1,88.1)

HollBivSym(recipient,donor)

##Or, equivalently
table3.16<-matrix(c(61.4,63.3,63.7,80,77.3,84,105,70.8,89.2,65.8,67.1,87.3,85.1,88.1),ncol=2)
HollBivSym(table3.16)
```

kendall.ci *Function to produce a confidence interval for Kendall's tau.*

Description

Based on sections 8.3 and 8.4 of Hollander, Wolfe & Chicken, Nonparametric Statistical Methods, 3e.

Usage

```
kendall.ci(x=NULL, y=NULL, alpha=0.05, type="t", bootstrap=F, B=1000, example=F)
```

Arguments

x	first data vector
y	second data vector
alpha	the significance level
type	type of confidence interval. Can be "t" (two-sided), "u" (upper) or "l" (lower).
bootstrap	if False, will find the asymptotic CI (as in section 8.3). If True, will find a bootstrap CI (as in section 8.4).
B	the number of bootstrap replicates
example	if True, will analyze data from Example 8.1

Author(s)

Eric Chicken

Examples

```
kendall.ci(example=TRUE)
```

klefsjo.ifr *Klefsjo's IFR*

Description

Function to compute the P-value for the observed Klefsjo's A* statistic.

Usage

```
klefsjo.ifr (x, alternative = "two.sided", exact=FALSE)
```

Arguments

x	a vector of data of length n
alternative	the direction of the alternative hypothesis. The choices are two.sided, ifr and dfr with the default value being two.sided.
exact	TRUE/FALSE value that determines whether the exact test or the large sample approximation is used if n >= 9. If n < 9 the exact test is used. The default value is FALSE, so the large sample approximation will be used unless specified not to.

Details

If the sample size is too large to allow for an exact value, due to duplicate coefficients, a note will be displayed and the large sample approximation will be used.

Value

The function returns a list with two elements:

A.star	the value of the Klefsjo statistic
p	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
velocity<-c(12.8, 12.9, 13.3, 13.4, 13.7, 13.8, 14.5)
klefsjo.ifr(velocity)

#Example of forced Large Sample Approximation
tb<-c(43, 45, 53, 56, 56, 57, 58, 66, 67, 73, 74, 79, 80, 80, 81, 81, 81, 82, 83, 83, 84, 88,
89,  91,  91,  92,  92,  97,  99,  99, 100, 100, 101, 102, 102, 102, 103, 104, 107, 108, 109,
113, 114, 118, 121, 123, 126, 128, 137, 138, 139, 144, 145, 147, 156, 162, 174, 178, 179, 184,
191, 198, 211, 214, 243, 249, 329, 380, 403, 511, 522, 598)
klefsjo.ifr(tb, exact=TRUE)
```

---

klefsjo.ifr.mc *Function to compute the Monte Carlo P-value for the observed Klefsjo's A* statistic.*

Description

This is the Monte Carlo approximation to the function "klefsjo.ifr".

Usage

```
klefsjo.ifr.mc(x, alternative = "two.sided", exact=FALSE,
               min.reps = 100, max.reps = 1000, delta = 10^-3)
```

Arguments

x	a vector of data of length n
alternative	the direction of the alternative hypothesis. The choices are two.sided, ifr and dfr with the default value being two.sided.
exact	TRUE/FALSE value that determines whether the exact test or the large sample approximation is used if n >= 9. If n < 9 the exact test is used. The default value is FALSE, so the large sample approximation will be used unless specified not to. This is the same large sample approximation as epstein()
min.reps	the minimum number of repetitions for the Monte Carlo Approximation
max.reps	the maximum number of reps for the Monte Carlo Approximation. If the maximum number of reps has been reached, and the probability has not converged, a warning is given.
delta	the measure of accuracy for the convergence. If the probability converges to within delta, the Monte Carlo procedure stops before reaching the maximum number of reps.

Value

The function returns a list with two elements:

A.star	the value of the Klefsjo statistic
p	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
temp.data<-c(0.33925023, 0.84005767, 0.29066189, 1.95163010, 0.74536608, 0.16714902, 0.06950791,
1.14919291, 1.93210982, 1.06006126, 0.14651009, 0.28776282, 0.72242750, 1.02227211, 1.71243334)
klefsjo.ifr.mc(temp.data, exact=TRUE)
```

`klefsjo.ifra` *Klefsjo's IFRA*

Description

Function to compute the P-value for the observed Klefsjo's B* statistic.

Usage

```
klefsjo.ifra (x, alternative = "two.sided", exact=FALSE)
```

Arguments

`x`	a vector of data of length n
`alternative`	the direction of the alternative hypothesis. The choices are two.sided, ifra and dfra with the default value being two.sided.
`exact`	TRUE/FALSE value that determines whether the exact test or the large sample approximation is used if n >= 9. If n < 9 the exact test is used. The default value is FALSE, so the large sample approximation will be used unless specified not to.

Details

If the sample size is too large to allow for an exact value, due to duplicate coefficients, a note will be displayed and the large sample approximation will be used.

Value

The function returns a list with two elements:

`B.star`	the value of the Klefsjo statistic
`p`	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
velocity<-c(12.8, 12.9, 13.3, 13.4, 13.7, 13.8, 14.5)
klefsjo.ifra(velocity)

#Example of forced Large Sample Approximation
tb<-c(43, 45, 53, 56, 56, 57, 58, 66, 67, 73, 74, 79, 80, 80, 81, 81, 81, 82, 83, 83, 84, 88,
89,  91,  91,  92,  92,  97,  99,  99, 100, 100, 101, 102, 102, 102, 103, 104, 107, 108, 109,
113, 114, 118, 121, 123, 126, 128, 137, 138, 139, 144, 145, 147, 156, 162, 174, 178, 179, 184,
191, 198, 211, 214, 243, 249, 329, 380, 403, 511, 522, 598)
klefsjo.ifra(tb, exact=TRUE)
```

`klefsjo.ifra.mc` *Function to compute the Monte Carlo P-value for the observed Klefsjo's B* statistic.*

Description

This is the Monte Carlo approximation to the function "klefsjo.ifra".

Usage

```
klefsjo.ifra.mc(x, alternative = "two.sided", exact=FALSE,
                min.reps = 100, max.reps = 1000, delta = 10^-3)
```

Arguments

`x`	a vector of data of length n
`alternative`	the direction of the alternative hypothesis. The choices are two.sided, ifra and dfra with the default value being two.sided.
`exact`	TRUE/FALSE value that determines whether the exact test or the large sample approximation is used if n >= 9. If n < 9 the exact test is used. The default value is FALSE, so the large sample approximation will be used unless specified not to. This is the same large sample approximation as epstein()
`min.reps`	the minimum number of repetitions for the Monte Carlo Approximation
`max.reps`	the maximum number of reps for the Monte Carlo Approximation. If the maximum number of reps has been reached, and the probability has not converged, a warning is given.
`delta`	the measure of accuracy for the convergence. If the probability converges to within delta, the Monte Carlo procedure stops before reaching the maximum number of reps.

Value

The function returns a list with two elements:

`B.star`	the value of the Klefsjo statistic
`p`	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
temp.data<-c(0.33925023, 0.84005767, 0.29066189, 1.95163010, 0.74536608, 0.16714902, 0.06950791,
1.14919291,1.93210982, 1.06006126, 0.14651009, 0.28776282, 0.72242750, 1.02227211, 1.71243334)
klefsjo.ifra.mc(temp.data, exact=TRUE)
```

kolmogorov *Kolmogorov*

Description

Function to compute the asymptotic P-value for the observed Kolmogorov D statistic.

Usage

```
kolmogorov(x,fnc,...)
```

Arguments

x	a vector of data of length n
fnc	the functional form of the pdf of F0. The first argument must be the data.
...	all the parameters besides the data that fnc needs to operate. (See below for an example using pnorm and pexp)

Value

The function returns a list with two elements:

D	the value of the Kolmogorov statistic
p	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
velocity<-c(12.8, 12.9, 13.3, 13.4, 13.7, 13.8, 14.5)
kolmogorov(velocity,pnorm, mean=14,sd=2)
kolmogorov(velocity,pexp,1/2)
```

mblm *Fitting Median-Based Linear Models (from 'mblm' oackage)*

Description

This function is used to fit linear models based on Theil-Sen single median, or Siegel repeated medians.

Usage

```
mblm(formula, dataframe, repeated = TRUE)
```

Arguments

formula	A formula of type y ~ x (only linear models are accepted)
dataframe	Optional dataframe
repeated	If set to true, model is computed using repeated medians. If false, a single median estimators are calculated

Details

This function is from the 'mblm' package, which is no longer available on CRAN.

Theil-Sen single median method computes slopes of lines crossing all possible pairs of points, when x coordinates differ. After calculating these n(n-1)/2 slopes (these value are true only if x is distinct), the median of them is taken as slope estimator. Next, the intercepts of n lines, crossing each point and having calculated slope are calculated. The median from them is intercept estimator.

Siegel repeated medians is more complicated. For each point, the slopes between it and the others are calcuated (resulting n-1 slopes) and the median is taken. This results in n medians and median from this medians is slope estimator. Intercept is calculated in similar way, for more information please take a look in function source.

The breakdown point of Theil-Sen method is about 29%, Siegel extended it to 50%, so these regression methods are very robust. Additionally, if the errors are normally distributed and no outliers are present, the estimators are very similar to classic least squares.

Value

An object of class c("mblm","lm"), containing minimal set of data to perform basic operations, such as in case of lm model. Additionally, the return value contains 2 fields:

slopes	The slopes (in single median), or medians of slopes (in repeated medians) between tested point pairs
intercepts	The intercepts calculated

Note

This function should have compatibility with all 'lm' methods, but it is not guaranteed that they will work or have any cognitive value (this method is nonparametric). The compatibility was only introduced to use some basic methods from 'lm' without programming new functions.

Author(s)

Lukasz Komsta, some fixes by Sven Garbade

References

Theil, H. (1950) A rank invariant method for linear and polynomial regression analysis. Nederl. Akad. Wetensch. Proc. Ser. A 53, 386-392 (Part I), 521-525 (Part II), 1397-1412 (Part III).

Sen, P.K. (1968). Estimates of Regression Coefficient Based on Kendall's tau. J. Am. Stat. Ass. 63, 324, 1379-1389.

Siegel, A.F. (1982). Robust Regression Using Repeated Medians. Biometrika, 69, 1, 242-244.

Examples

```
set.seed(1234)
x <- 1:100+rnorm(100)
y <- x+rnorm(100)
y[100] <- 200
fit <- mblm(y~x)
fit
summary(fit)
fit2 <- lm(y~x)
plot(x,y)
abline(fit)
abline(fit2,lty=2)
plot(fit)
residuals(fit)
fitted(fit)
plot(density(fit$slopes))
plot(density(fit$intercepts))
anova(fit)
anova(fit2)
anova(fit,fit2)
confint(fit)
AIC(fit,fit2)
```

MillerJack *Miller Jackknife*

Description

Function to compute the Miller Jackknife Q statistic.

Usage

```
MillerJack(x,y=NULL)
```

Arguments

x	Either a vector containing the first group of data (X) or a matrix containing both groups of data.
y	If x is a vector, y is a vector containing the second group of data (Y). Otherwise, not used.

Value

Returns the observed Q statistic.

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 5.2 Southern Armyworm and Pokeweed
kentucky.pokeweed<-c(6.2,5.9,8.9,6.5,8.6)
florida.pokeweed<-c(9.5,9.8,9.5,9.6,10.3)
MillerJack(kentucky.pokeweed,florida.pokeweed)
```

---

mrl *Mean Residual Life*

Description

Function to return the mean residual life along with Hall and Wellner's upper and lower bounds.

Usage

```
mrl(data, alpha, main=NULL, ylim=NULL, xlab=NULL,...)
```

Arguments

data	a vector of survival times
alpha	(1-alpha) is the approximate coverage probability for the confidence band.
main	title of the plot. The default is "Plot of Mean Residual Life and bounds".
ylim	the limits of the y-axis. The default is to include all points in the plotting range.
xlab	the label for the x-axis. The default is Time.
...	additional plotting options

Value

The function returns a list with three vectors:

PM	the mean residual life
PMU	upper bound for the mean residual life
PML	lower bound for the mean residual life

Author(s)

Rachel Becvarik

Examples

```
leukemia<-c(7, 429, 579, 968, 1877, 47, 440, 581, 1077, 1886, 58,
445,  650, 1109, 2045, 74, 455, 702, 1314, 2056, 177, 468,
715, 1334, 2260, 232, 495, 779, 1367, 2429, 273, 497, 881,
1534, 2509, 285, 532, 900, 1712, 317,  571, 930, 1784)
mrl(leukemia, .05)
```

multCh7 *Possible arrangements by row for a matrix*

Description

Similar to multComb, this function will generate all of the possible arrangements of the data by row within a matrix. For a given matrix of n rows and k columns, this will give (k!)^n possible arrangements

Usage

```
multCh7(our.matrix)
```

Arguments

our.matrix	The matrix containing the data which will be rearranged by row.

Details

The computations involved get very time consuming very quickly, so be careful not to use it for too large of a matrix.

Value

Returns an array, containing (k!)^n distinct matrices of the same size as our.matrix

Note

This function is used to generate the possible permutations for the Exact methods used in Chapter 7 of Hollander, Wolfe, and Chicken - Nonparametric Statistical Methods Third Edition.

Author(s)

Grant Schneider

Examples

```
some.matrix<-matrix(c(1,2,7,4,5,9),ncol=3,byrow=TRUE)
multCh7(some.matrix)
```

multCh7SM *Possible arrangements by row a matrix, where NA values are ignored*

Description

Similar to multCh7, this function will generate all of the possible arrangements of the data by row within a matrix, except for NA values, which will remain fixed. This function is used in pSkilMack and cSkilMack to generate the Exact distribution. For a given matrix of with k1,...kn non-missing values, this will give k1!*k2!*...*kn! possible arrangements

Usage

```
multCh7SM(our.matrix)
```

Arguments

our.matrix	The matrix containing the data (including NA values) which will be rearranged by row.

Details

The computations involved get very time consuming very quickly, so be careful not to use it for too large of a matrix.

Value

Returns an array, containing k1!*k2!*...*kn! distinct matrices of the same size as our.matrix

Author(s)

Grant Schneider

Examples

```
##Get a matrix with some NA's
our.matrix<-matrix(c(NA,1,2,3,5,7,NA,NA,11),ncol=3,byrow=TRUE)
##Get every possible arrangement by row, treating the NA's as fixed
multCh7SM(our.matrix)
```

multComb *Combinations of the first n integers in k groups*

Description

This is a function, used for generating the permutations used for the Exact distribution of many of the statistical procedures in Hollander, Wolfe, Chicken - Nonparametric Statistical Methods Third Edition, to generate possible combinations of the first n=n1+n2+...+nk integers within k groups.

Usage

```
multComb(n.vec)
```

Arguments

n.vec	Contains the group sizes n1,n2,...,nk

Details

The computations involved get very time consuming very quickly, so be careful not to use it for too many large groups.

Value

Returns a matrix of n!/(n1!*n2!*...*nk!) rows, where each row represents one possible combination.

Author(s)

Grant Schneider

Examples

```
##What are the ways that we can group 1,2,3,4,5 into groups of 2, 2, and 1?
multComb(c(2,2,1))

##Another example, with four groups
multComb(c(2,2,3,2))
```

nb.mc *Function to compute the Monte Carlo P-value for the observed Hollander-Proschan T statistic.*

Description

This is the Monte Carlo approximation to the newbet function.

Usage

```
nb.mc(x, alternative = "two.sided", exact=FALSE,
      min.reps = 100, max.reps = 1000, delta = 10^-3)
```

Arguments

x	a vector of data of length n
alternative	the direction of the alternative hypothesis. The choices are two.sided, nbu, and nwu with the default value being two.sided.
exact	TRUE/FALSE value that determines whether the exact test or the large sample approximation is used if n >= 9. If n < 9 the exact test is used. The default value is FALSE, so the large sample approximation will be used unless specified not to. This is the same large sample approximation as epstein()
min.reps	the minimum number of repetitions for the Monte Carlo Approximation
max.reps	the maximum number of reps for the Monte Carlo Approximation. If the maximum number of reps has been reached, and the probability has not converged, a warning is given.
delta	the measure of accuracy for the convergence. If the probability converges to within delta, the Monte Carlo procedure stops before reaching the maximum number of reps.

Value

The function returns a list with two elements:

T	the value of the Hollander-Proschan statistic
p	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
table11.4<-c(194,15,41,29,33,181)
nb.mc(table11.4, alt="nbu")
```

newbet *Hollander-Proschan T**

Description

Function to compute the asymptotic P-value for the observed Hollander-Proschan T* statistic.

Usage

```
newbet(x)
```

Arguments

x	a vector of data of length n

Value

The function returns a list with two elements:

T	the value of the Hollander-Proschan statistic
T.star	the standardized value of the Hollander-Proschan statistic
p	the corresponding probability

Author(s)

Rachel Becvarik

Examples

```
table11.4<-c(194,15,41,29,33,181)
newbet(table11.4)
```

owa *Ordered Walsh Averages*

Description

Function to compute the ordered Walsh averages and the value of the Hodges-Lehmann estimator

Usage

```
owa(x,y)
```

Arguments

x	first vector of data of length n
y	second vector of data of length n

Value

Returns a list containing:

owa	the ordered Walsh averages
h.l	the value of the Hodges-Lehmann estimator

Author(s)

Rachel Becvarik

Examples

```
##Hollander-Wolfe-Chicken Example 3.3
x<-c(1.83, 0.50, 1.62, 2.48, 1.68, 1.88, 1.55, 3.06, 1.30)
y<-c(0.878, 0.647, 0.598, 2.050, 1.060, 1.290, 1.060, 3.140, 1.290)
owa(x,y)
```

---

pAnsBrad *Function to compute the P-value for the observed Ansari-Bradley C statistic.*

Description

When there are no ties in the data, this function uses pansari and cansari from the base stats package to compute the C statistic and P-value ("Exact" or "Asymptotic"). The program is reasonably quick for large data in the absence of ties, well after the asymptotic approximation suffices, so Monte Carlo methods are not included.

When there are ties in the data, this function computes the C statistic and P-value ("Exact", "Monte Carlo", or "Asymptotic").

Usage

```
pAnsBrad(x,y=NA,g=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a list or a vector containing either all or the first group of data.
y	If x contains the first group of data, y contains the second group of data. Otherwise, not used.
g	If x contains a vector of all of the data, g is a vector of 1's and 2's corresponding to group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA and there are no ties in the data, "Exact" will be used. When method=NA and there are ties in the data, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the two groups of data can be entered in any of three ways. For data a=1,2 and b=3,4 all of the following are equivalent:

`pAnsBrad(x=c(1,2),y=c(3,4))` `pAnsBrad(x=list(c(1,2),c(3,4)))` `pAnsBrad(x=c(1,2,3,4),g=c(1,1,2,2))`

Value

Returns a list with "NSM3Ch5p" class containing the following components:

`m`	number of observations in the first data group (X)
`n`	number of observations in the second data group (Y)
`obs.stat`	the observed C statistic
`p.val`	upper tail P-value
`two.sided`	two-sided P-value

Note

If method="Monte Carlo" and there are no ties in the data, a warning is displayed and the "Exact" method is used.

Author(s)

Grant Schneider

See Also

Also see `ansari.test`.

Examples

```
##Hollander, Wolfe, Chicken Example 5.1 Serum Iron Determination:
serum<-list(ramsay = c(111, 107, 100, 99, 102, 106, 109, 108, 104, 99, 101, 96, 97, 102, 107,
113, 116, 113, 110, 98),
jung.parekh = c(107, 108, 106, 98, 105, 103, 110, 105, 104, 100, 96, 108, 103, 104, 114, 114,
113, 108, 106, 99))


pAnsBrad(serum)

##or, equivalently:
pAnsBrad(serum$ramsay, serum$jung.parekh)
```

pBohnWolfe *Function to compute the P-value for the observed Bohn-Wolfe U statistic.*

Description

This function computes the U statistic and then uses Monte Carlo sampling to compute the corresponding P-value. The Monte Carlo samples are simulated based on the order statistics of a uniform(0,1) distribution.

Usage

```
pBohnWolfe(x,y,k,q,c,d,method="Monte Carlo",n.mc=10000)
```

Arguments

x	A vector containing the data in the first group.
y	A vector containing the data in the Second group.
k	A numeric value indicating the set size of the first data group in the RSS (X).
q	A numeric value indicating the set size of the second data group in the RSS (Y).
c	A numeric value indicating the number of cycles for the first data group in the RSS (X).
d	A numeric value indicating the number of cycles for the second data group in the RSS (Y).
method	For this procedure, method is currently set automatically to "Monte Carlo" as the only option that is available. For standardization with other critical value procedures in the NSM3 package, "Asymptotic" and "Exact" will be supported in future versions.
n.mc	Number of Monte Carlo samples used to estimate the distribution of U.

Value

Returns a list with "NSM3Ch5p" class containing the following components:

m	number of observations in RSS for the first data group (X)
n	number of observations in RSS for the second data group (Y)
obs.stat	the observed U statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

References

Bohn, Lora L., and Douglas A. Wolfe. "Nonparametric two-sample procedures for ranked-set samples data." Journal of the American Statistical Association 87.418 (1992): 552-561

Examples

```
##Hollander, Wolfe, Chicken Example 15.4 Body Mass Index:
male<-c(18.0, 20.5, 21.3, 21.3, 22.3, 23.8, 23.8, 24.6, 25.0, 25.2, 25.3, 25.9, 26.1, 27.0,
27.4, 27.4, 28.4, 29.4, 29.6, 32.8)
female<-c(17.2, 17.8, 19.9, 20.0, 21.7, 22.0, 22.3, 23.1, 23.9, 25.8, 27.1, 29.6, 30.1, 30.3,
30.7, 31.1, 35.2, 35.6, 38.1, 42.5)

pBohnWolfe(male,female,4,4,5,5)
##To use more Monte Carlo samples:
#pBohnWolfe(male,female,4,4,5,5,n.mc=100000)
```

---

pDurSkiMa *Durbin, Skillings-Mack*

Description

Function to compute the P-value for the observed Durbin, Skillings-Mack D statistic.

Usage

```
pDurSkiMa(x,b=NA,trt=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a matrix or a vector containing the data.
b	If x is a vector, b is a required vector of block labels. Otherwise, not used.
trt	If x is a vector, trt is a required vector of treatment labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the data can be entered in either of two ways. The following are equivalent: `pDurSkiMa(x=matrix(c(1,2,3,4,5,6),ncol=2,byrow=T))` `pDurSkiMa(x=c(1,2,3,4,5,6),b=`

Value

Returns a list with "NSM3Ch7p" class containing the following components:

k	number of treatments in the data
n	number of blocks in the data
ss	number of treatments per block
pp	number of observations per treatment
lambda	number of times each pair of treatments occurs together within a block
obs.stat	the observed D statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

Examples

```
##Hollander, Wolfe, Chicken Example 7.6 Chemical Toxicity
table7.12<-matrix(nrow=7,ncol=7)
table7.12[1,c(1,2,4)]<-c(0.465,0.343,0.396)
table7.12[2,c(1,3,5)]<-c(0.602,0.873,0.634)
table7.12[3,c(3,4,7)]<-c(0.875,0.325,0.330)
table7.12[4,c(1,6,7)]<-c(0.423,0.987,0.426)
table7.12[5,c(2,3,6)]<-c(0.652,1.142,0.989)
table7.12[6,c(2,5,7)]<-c(0.536,0.409,0.309)
table7.12[7,c(4,5,6)]<-c(0.609,0.417,0.931)

pDurSkiMa(table7.12)

##or, equivalently:
x<-c(.465,.602,.423,.343,.652,.536,.873,.875,1.142,.396,.325,.609,.634,.409,.417,.987,.989,
.931,.330,.426,.309)
b<-c(1,2,4,1,5,6,2,3,5,1,3,7,2,6,7,4,5,7,3,4,6)
trt<-c(rep("A",3),rep("B",3),rep("C",3),rep("D",3),rep("E",3),rep("F",3),rep("g",3))

pDurSkiMa(x,b,trt)
```

---

pFligPoli *Fligner-Policello*

Description

Function to compute the P-value for the observed Fligner-Policello U statistic.

Usage

```
pFligPoli(x,y=NA,g=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a list or a vector containing either all or the first group of data.
y	If x contains the first group of data, y contains the second group of data. Otherwise, not used.
g	If x contains a vector of all of the data, g is a vector of 1's and 2's corresponding to group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the two groups of data can be entered in any of three ways. For data a=1,2 and b=3,4 all of the following are equivalent:

`pFligPoli(x=c(1,2),y=c(3,4))` `pFligPoli(x=list(c(1,2),c(3,4)))` `pFligPoli(x=c(1,2,3,4),g=c(1,1,2,2))`

Value

Returns a list with "NSM3Ch5p" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
obs.stat	the observed U statistic
p.val	upper tail P-value
two.sided	two-sided P-value

Author(s)

Grant Schneider

Examples

```
##Hollander, Wolfe, Chicken Example 4.5 Plasma Glucose in Geese
plasma.glucose<-list(healthy.geese = c(297, 340, 325, 227, 277, 337,
250, 290), poisoned.geese = c(293, 291, 289, 430, 510, 353, 318
))

pFligPoli(plasma.glucose)
```

pFrd *Function to compute the P-value for the observed Friedman, Kendall-Babington Smith S statistic.*

Description

The method used to compute the P-value is from the reference by Van de Wiel, Bucchianico, and Van der Laan.

Usage

```
pFrd(x,b=NA,trt=NA,method=NA, n.mc=10000)
```

Arguments

x	Either a matrix or a vector containing the data.
b	If x is a vector, b is a required vector of block labels. Otherwise, not used.
trt	If x is a vector, trt is a required vector of treatment labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the data can be entered in either of two ways. The following are equivalent:

```
pFrd(x=matrix(c(1,2,3,4,5,6),ncol=2,byrow=T)) pFrd(x=c(1,2,3,4,5,6),b=c(1,1,2,2,3,3),trt=c(1,2,1,2,1
```

Value

Returns a list with "NSM3Ch7p" class containing the following components:

k	number of treatments in the data
n	number of blocks in the data
obs.stat	the observed D statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

References

Van de Wiel, M. A., A. Di Bucchianico, and P. Van der Laan. "Symbolic computation and exact distributions of nonparametric test statistics." Journal of the Royal Statistical Society: Series D (The Statistician) 48.4 (1999): 507-516.

See Also

Also see the `coin` package.

Examples

```
##Hollander-Wolfe-Chicken Example 7.1 Rounding First Base
rounding.times<-matrix(c(5.40, 5.50, 5.55,
                         5.85, 5.70, 5.75,
                         5.20, 5.60, 5.50,
                         5.55, 5.50, 5.40,
                         5.90, 5.85, 5.70,
                         5.45, 5.55, 5.60,
                         5.40, 5.40, 5.35,
                         5.45, 5.50, 5.35,
                         5.25, 5.15, 5.00,
                         5.85, 5.80, 5.70,
                         5.25, 5.20, 5.10,
                         5.65, 5.55, 5.45,
                         5.60, 5.35, 5.45,
                         5.05, 5.00, 4.95,
                         5.50, 5.50, 5.40,
                         5.45, 5.55, 5.50,
                         5.55, 5.55, 5.35,
                         5.45, 5.50, 5.55,
                         5.50, 5.45, 5.25,
                         5.65, 5.60, 5.40,
                         5.70, 5.65, 5.55,
                         6.30, 6.30, 6.25),ncol=3,byrow=TRUE)
#pFrd(rounding.times,n.mc=20000)
pFrd(rounding.times,n.mc=2000)
```

`pHaySton` *Hayter-Stone*

Description

Function to compute the P-value for the observed Hayter-Stone W statistic.

Usage

```
pHaySton(x,g=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a list or a vector containing the data.
g	If x is a vector, g is a required vector of group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo", or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the groups of data can be entered in either of two ways. For data a=1,2 and b=3,4,5 the following are equivalent:

`pHaySton(x=list(c(1,2),c(3,4,5)))` `pHaySton(x=c(1,2,3,4,5),g=c(1,1,2,2,2))`

Value

Returns a list with "NSM3Ch6MCp" class containing the following components:

n	a vector containing the number of observations in each of the data groups
obs.stat	the observed W statistic for each of the k*(k-1)/2 comparisons
p.val	upper tail P-value corresponding to each W statistic

Author(s)

Grant Schneider

Examples

```
##Hollander, Wolfe, Chicken Example 6.7 Motivational Effect of Knowledge of Performance:
motivational.effect<-list(no.Info = c(40, 35, 38, 43, 44, 41), rough.Info = c(38,
40, 47, 44, 40, 42), accurate.Info = c(48, 40, 45, 43, 46, 44
))

#pHaySton(motivational.effect,method="Monte Carlo")
pHaySton(motivational.effect,method="Asymptotic")
#pHaySton(rnorm(10),rep(1:3,c(3,3,4)),method="Asymptotic")
```

---

pHaySton LSA *Hayter-Sone LSA*

Description

Function to compute the upper tail probability of the Hayter-Stone W asymptotic distribution for a given cutoff.

Usage

```
pHayStonLSA(h,k,delta=.001)
```

Arguments

h	Cutoff used to calculate the P-value.
k	Number of groups.
delta	Defines the fineness of the grid used to calculate the asymptotic distribution of W.

Value

Returns the asymptotic upper tail P-value.

Author(s)

Grant Schneider

Examples

```
pHayStonLSA(2.491,3)
pHayStonLSA(4.112,4)
```

---

pHoeff *Hoeffding's D*

Description

Function to approximate the distribution of Hoeffding's D statistic using a Monte Carlo Sample under the null hypothesis. This code follows section 8.6 of Hollander, Wolfe & Chicken, Nonparametric Statistical Methods, 3e. This calls HoeffD, a small bit of code that produces the value of D without any inference. It is intended for small sample sizes n only. For large n, use the asymptotic equivalence of D to the Blum-Kliefer-Rosenblatt statistic in the R package "Hmisc", command "hoeffd".

Usage

```
pHoeff(n=5, reps=10000, r=4)
```

Arguments

n	the sample size
reps	the number of Monte Carlo runs to produce
r	the number of digits for rounding the results

Value

Returns a matrix containing the Monte Carlo distribution of the D statistic.

Author(s)

Eric Chicken

See Also

Also see the Hmisc package.

Examples

```
pHoeff(n=5, reps=10000, r=4)
pHoeff(n=10, reps=1000, r=5)
```

---

pHollBivSym *Hollander Bivariate Symmetry*

Description

Function to compute the P-value for the observed Hollander A statistic.

Usage

```
pHollBivSym(x,y=NA,g=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a list or a vector containing either all or the first group of data.
y	If x contains the first group of data, y contains the second group of data. Otherwise, not used.
g	If x contains a vector of all of the data, g is a vector of 1's and 2's corresponding to group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used. As Kepner and Randles (1984) and Hilton and Gee (1997) have found the large sample approximation to perform poorly, method="Asymptotic" will be treated as method=NA.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the two groups of data can be entered in any of three ways. For data a=1,2 and b=3,4 all of the following are equivalent:

```
pHollBivSym(x=c(1,2),y=c(3,4)) pHollBivSym(x=list(c(1,2),c(3,4))) pHollBivSym(x=c(1,2,3,4),g=c(1,1,2
```

Value

Returns a list with "NSM3Ch5p" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
obs.stat	the observed A statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

References

Kepner, James L., and Ronald H. Randies. "Comparison of tests for bivariate symmetry versus location and/or scale alternatives." Communications in Statistics-Theory and Methods 13.8 (1984): 915-930.

Hilton, Joan F., and Lauren Gee. "The size and power of the exact bivariate symmetry test." Computational statistics & data analysis 26.1 (1997): 53-69.

Examples

```
##Hollander-Wolfe-Chicken Example 3.11 Insulin Clearance in Kidney Transplants
x<-c(61.4,63.3,63.7,80,77.3,84,105)
y<-c(70.8,89.2,65.8,67.1,87.3,85.1,88.1)

##Exact p-value
pHollBivSym(x,y)
```

---

pJCK

Function to compute the P-value for the observed Jonckheere-Terpstra J statistic.

Description

This function computes the observed J statistic for the given data and corresponding P-value. When there are no ties in the data, the function takes advantage of Harding's (1984) algorithm to quickly generate the exact distribution of J.

Usage

```
pJCK(x,g=NA,method=NA, n.mc=10000)
```

Arguments

x	Either a list or a vector containing the data.
g	If x is a vector, g is a required vector of group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo", or "Asymptotic", indicating the desired distribution. When method=NA and ties are not present, "Exact" will be used. When method=NA and ties are present, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the groups of data can be entered in either of two ways. For data a=1,2 and b=3,4,5 the following are equivalent:

`pJCK(x=list(c(1,2),c(3,4,5)))` `pJCK(x=c(1,2,3,4,5),g=c(1,1,2,2,2))`

Value

Returns a list with "NSM3Ch6p" class containing the following components:

n	a vector containing the number of observations in each of the data groups
obs.stat	the observed J statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

References

Harding, E. F. "An efficient, minimal-storage procedure for calculating the Mann-Whitney U, generalized U and similar distributions." Applied statistics (1984): 1-6.

Examples

```
##Hollander-Wolfe-Chicken Example 6.2 Motivational Effect of Knowledge of Performance
motivational.effect<-list(no.Info=c(40,35,38,43,44,41),rough.Info=c(38,40,47,44,40,42),
                          accurate.Info=c(48,40,45,43,46,44))
#pJCK(motivational.effect,method="Monte Carlo")
pJCK(motivational.effect,method="Asymptotic")
```

pKolSmirn *Function to copute the P-value for the observed Kolmogorov-Smirnov J statistic.*

Description

This function uses psmirnov2x from the base stats package to compute the J statistic and corresponding P-value. The program is reasonably quick for large data, well after the asymptotic approximation suffices, so Monte Carlo methods are not included. This function primarily serves as a wrapper to the ks.test function with the output standardized to the format of the other functions included in the NSM3 package.

Usage

```
pKolSmirn(x,y=NA,g=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a list or a vector containing either all or the first group of data.
y	If x contains the first group of data, y contains the second group of data. Otherwise, not used.
g	If x contains a vector of all of the data, g is a vector of 1's and 2's corresponding to group labels. Otherwise, not used.
method	Either "Exact" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the two groups of data can be entered in any of three ways. For data a=1,2 and b=3,4 all of the following are equivalent:

`pKolSmirn(x=c(1,2),y=c(3,4))` `pKolSmirn(x=list(c(1,2),c(3,4)))` `pKolSmirn(x=c(1,2,3,4),g=c(1,1,2,2))`

Value

Returns a list with "NSM3Ch5p" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
obs.stat	the observed C statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

See Also

Also see `ks.test()`.

Examples

```
##Hollander-Wolfe-Chicken Example 5.4 Effect of Feedback on Salivation Rate:
feedback<-c(-0.15, 8.6, 5, 3.71, 4.29, 7.74, 2.48, 3.25, -1.15, 8.38)
no.feedback<-c(2.55, 12.07, 0.46, 0.35, 2.69, -0.94, 1.73, 0.73, -0.35, -0.37)
pKolSmirn(x=feedback,y=no.feedback)
```

---

pKW *Kruskal-Wallis*

Description

Function to compute the P-value for the observed Kruskal-Wallis H statistic.

Usage

```
pKW(x,g=NA, method=NA, n.mc=10000)
```

Arguments

`x`	Either a list or a vector containing the data.
`g`	If x is a vector, g is a required vector of group labels. Otherwise, not used.
`method`	Either "Exact", "Monte Carlo", or "Asymptotic", indicating the desired distribution. When method=NA and ties are not present, "Exact" will be used. When method=NA and ties are present, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
`n.mc`	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the groups of data can be entered in either of two ways. For data a=1,2 and b=3,4,5 the following are equivalent:

`pKW(x=list(c(1,2),c(3,4,5)))` `pKW(x=c(1,2,3,4,5),g=c(1,1,2,2,2))`

Value

Returns a list with "NSM3Ch6p" class containing the following components:

`n`	a vector containing the number of observations in each of the data groups
`obs.stat`	the observed H statistic
`p.val`	upper tail P-value

Author(s)

Grant Schneider

See Also

Also see `kruskal.test()`.

Examples

```
##Hollander-Wolfe-Chicken Example 6.1 Half-Time of Mucociliary Clearance
mucociliary<-list(Normal = c(2.9, 3, 2.5, 2.6, 3.2), Obstructive = c(3.8,
2.7, 4, 2.4), Asbestosis = c(2.8, 3.4, 3.7, 2.2, 2))

pKW(mucociliary)
```

---

pLepage *Lepage*

Description

Function to compute the P-value for the observed Lepage D statistic.

Usage

```
pLepage(x,y=NA,g=NA,method=NA,n.mc=10000)
```

Arguments

`x`	Either a list or a vector containing either all or the first group of data.
`y`	If x contains the first group of data, y contains the second group of data. Otherwise, not used.
`g`	If x contains a vector of all of the data, g is a vector of 1's and 2's corresponding to group labels. Otherwise, not used.
`method`	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
`n.mc`	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the two groups of data can be entered in any of three ways. For data a=1,2 and b=3,4 all of the following are equivalent:

`pLepage(x=c(1,2),y=c(3,4))` `pLepage(x=list(c(1,2),c(3,4)))` `pLepage(x=c(1,2,3,4),g=c(1,1,2,2))`

Value

Returns a list with "NSM3Ch5p" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
obs.stat	the observed C statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 5.3 Platelet Counts of Newborn Infants
platelet.counts<-list(x = c(120000, 124000, 215000, 90000, 67000, 95000,
190000, 180000, 135000, 399000), y = c(12000, 20000, 112000,
32000, 60000, 40000))

pLepage(platelet.counts)

##or equivalently,

pLepage(platelet.counts$x,platelet.counts$y)
```

---

pMackSkil *Mack-Skillings*

Description

Function to compute the P-value for the observed Mack-Skillings MS statistic.

Usage

```
pMackSkil(x,b=NA,trt=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a 3 dimensional array or a vector containing the data.
b	If x is a vector, b is a required vector of block labels. Otherwise, not used.
trt	If x is a vector, trt is a required vector of treatment labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the data can be entered in either of two ways. The following are equivalent:

`pMackSkil(x=array(c(1,2,3,4,5,6),dim=c(1,2,3))` `pMackSkil(x=c(1,2,3,4,5,6),b=c(1,1,2,2,3,3),trt=c(1,2`

Value

Returns a list with "NSM3Ch7p" class containing the following components:

`k`	number of treatments in the data
`n`	number of blocks in the data
`c`	number of repetitions for each treatment and block combination
`obs.stat`	the observed MS statistic
`p.val`	upper tail P-value

Author(s)

Grant Schneider

Examples

```
##Hollander, Wolfe, Chicken Example 7.9 Determination of Niacin in Bran Flakes
niacin<-array(dim=c(3,4,3))
niacin[,,1]<-c(7.58,7.87,7.71,8,8.27,8,7.6,7.3,7.82,8.03,7.35,7.66)
niacin[,,2]<-c(11.63,11.87,11.4,12.2,11.7,11.8,11.04,11.5,11.49,11.5,10.10,11.7)
niacin[,,3]<-c(15,15.92,15.58,16.6,16.4,15.9,15.87,15.91,16.28,15.1,14.8,15.7)
```

`pMaxCorrNor` *Function to compute the upper tail probability of the maximum of k N(0,1) random variables with common correlation for a given cutoff.*

Description

Uses the integrate function based on the method proposed in Gupta, Panchapakesan and Sohn (1983).

Usage

```
pMaxCorrNor(x,k,rho)
```

Arguments

`x`	Cutoff at which the upper-tail P-value is to be calculated.
`k`	Number of random variables.
`rho`	Common correlation between the random variables.

Value

Returns the upper tail probability at the user-specified cutoff.

Author(s)

Grant Schneider

References

Gupta, Shanti S., S. Panchapakesan, and Joong K. Sohn. "On the distribution of the studentized maximum of equally correlated normal random variables." Communications in Statistics-Simulation and Computation 14.1 (1985): 103-135.

Examples

```
##Hollander-Wolfe-Chicken Section 7.14
pMaxCorrNor(2.575,5,.3)

##Hollander-Wolfe-Chicken Example 7.14 Effect of Weight on Forearm Tremor Frequency
pMaxCorrNor(1.93,5,.452)
```

---

pNDWol *Nemenyi, Damico-Wolfe*

Description

Function to compute the P-value for the observed Nemenyi, Damico-Wolfe Y statistic.

Usage

```
pNDWol(x,g=NA,method=NA, n.mc=10000)
```

Arguments

x	Either a list or a vector containing the data.
g	If x is a vector, g is a required vector of group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Value

Returns a list with "NSM3Ch6MCp" class containing the following components:

n	number of observations in the k data groups, with the first group representing the control
obs.stat	the observed Y statistic for each treatment vs. control comparison
p.val	upper tail P-value corresponding to each of the k-1 observed Y statistics

Note

The data group containing the treatment values should be entered as the first group.

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 6.8 Motivational Effect of Knowledge of Performance
motivational.effect<-list(no.Info = c(40, 35, 38, 43, 44, 41),
rough.Info = c(38, 40, 47, 44, 40, 42),
accurate.Info = c(48, 40, 45, 43, 46, 44))

pNDWol(motivational.effect,method="Asymptotic")
pNDWol(motivational.effect,method="Monte Carlo")
```

---

pNWWM *Nemenyi, Wilcoxon-Wilcox, Miller*

Description

Function to compute the P-value for the observed Nemenyi, Wilcoxon-Wilcox, Miller R* statistic.

Usage

```
pNWWM(x,b=NA,trt=NA,method=NA, n.mc=10000)
```

Arguments

x	Either a matrix or a vector containing the data, with control assumed to be the first group.
b	If x is a vector, b is a required vector of block labels. Otherwise, not used.
trt	If x is a vector, trt is a required vector of treatment labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the data can be entered in either of two ways. The following are equivalent:

```
pNWWM(x=matrix(c(1,2,3,4,5,6),ncol=2,byrow=T)) pNWWM(x=c(1,2,3,4,5,6),b=c(1,1,2,2,3,3),trt=c(1,2,1,2
```

Value

Returns a list with "NSM3Ch7MCp" class containing the following components:

k	number of treatments (including the control)
n	number of blocks
obs.stat	the observed R* statistic for each treatment vs. control comparison
p.val	upper tail P-value corresponding to each of the k-1 observed R* statistics

Note

The data group containing the treatment values should be entered as the first group.

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 7.4 Stuttering Adaptation
adaptation.scores<-matrix(c(57,59,44,51,43,49,48,56,44,50,44,50,70,42,58,54,38,48,38,48,50,53,53,
56,37,58,44,50,58,48,60,58,60,38,48,56,51,56,44,44,50,54,50,40,50,50,56,46,74,57,74,48,48,44),
ncol=3,dimnames = list(1 : 18,c("No Shock", "Shock Following", "Shock During")))

#pNWWM(adaptation.scores)
pNWWM(adaptation.scores,n.mc=2500)
```

---

pPage *Page*

Description

Function to compute the P-value for the observed Page L statistic.

Usage

```
pPage(x,b=NA,trt=NA,method=NA, n.mc=10000)
```

Arguments

x	Either a matrix or a vector containing the data.
b	If x is a vector, b is a required vector of block labels. Otherwise, not used.
trt	If x is a vector, trt is a required vector of treatment labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the data can be entered in either of two ways. The following are equivalent:

```
pPage(x=matrix(c(1,2,3,4,5,6),ncol=2,byrow=T)) pPage(x=c(1,2,3,4,5,6),b=c(1,1,2,2,3,3),trt=c(1,2,1,2
```

Value

Returns a list with "NSM3Ch7p" class containing the following components:

k	number of treatments in the data
n	number of blocks in the data
obs.stat	the observed L statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 7.2 Breaking Strength of Cotton Fibers
strength.index<-matrix(c(7.46, 7.68, 7.21, 7.17, 7.57, 7.80, 7.76, 7.73, 7.74, 8.14, 8.15,
7.87, 7.63, 8.00, 7.93),byrow=FALSE,ncol=5)

#pPage(strength.index,method="Exact")
pPage(strength.index,method="Monte Carlo")
```

pPairedWilcoxon *Paired Wilcoxon*

Description

Function to extend wilcox.test to compute the (exact or Monte Carlo) P-value for paired Wilcoxon data in the presence of ties.

Usage

```
pPairedWilcoxon(x,y=NA,g=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a list or a vector containing either all or the first group of data.
y	If x contains the first group of data, y contains the second group of data. Otherwise, not used.
g	If x contains a vector of all of the data, g is a vector of 1's and 2's corresponding to group labels. Otherwise, not used.
method	Either "Exact" or "Monte Carlo", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the two groups of data can be entered in any of three ways. For data a=1,2 and b=3,4 all of the following are equivalent:

`pPairedWilcoxon(x=c(1,2),y=c(3,4))` `pPairedWilcoxon(x=list(c(1,2),c(3,4)))` `pPairedWilcoxon(x=c(1,2,3,4`

Value

Returns a list with "NSM3Ch5p" class containing the following components:

m	number of observations in the first data group (X)
n	number of observations in the second data group (Y)
obs.stat	the observed T+ statistic
p.val	upper tail P-value

Note

If there are 0s in the Z values (the difference between X and Y), these will be removed and the calculations will be done based on the smaller sample size, as detailed section 3.1 of Hollander, Wolfe, and Chicken - NSM3.

Author(s)

Grant Schneider

See Also

Also see `stats::wilcox.test()`

Examples

```
##Hollander-Wolfe-Chicken Example 3.1 Hamilton Depression Scale Factor IV
x <-c(1.83, .50, 1.62, 2.48, 1.68, 1.88, 1.55, 3.06, 1.30)
y <-c(0.878, .647, .598, 2.05, 1.06, 1.29, 1.06, 3.14, 1.29)

wilcox.test(y,x,paired=TRUE,alternative="less")
pPairedWilcoxon(x,y)
```

---

pRangeNor

Function to compute the upper-tail probability of the range of k independent N(0,1) random variables for a given cutoff.

Description

Uses the integrate function based on the method proposed in Harter (1960).

Usage

```
pRangeNor(x,k)
```

Arguments

x	Cutoff at which the upper-tail P-value is to be calculated.
k	Number of independent Normal random variables.

Value

Returns the upper tail probability at the user-specified cutoff.

Author(s)

Grant Schneider

References

Harter, H. Leon. "Tables of range and studentized range." The Annals of Mathematical Statistics (1960): 1122-1147.

Examples

```
##Hollander-Wolfe-Chicken Example 7.3 Rounding First Base
pRangeNor(4.121,3)

##Hollander-Wolfe-Chicken Example 7.7 Chemical Toxicity
pRangeNor(4.171,7)
```

---

print.NSM3Ch5p *Methods to control displayed output of NSM3 tests.*

Description

These methods are used to display the list output from the functions used to perform the various nonparametric statistical procedures in the NSM3 package.

Usage

```
## S3 method for class 'NSM3Ch5p'
print(x, ...)
```

Arguments

x	The list object returned by a procedure in the NSM3 package.
...	Other options to be specified.

Value

The exact wording of the displayed output will vary depending on the setting. For example two sample procedures and k-sample procedures will be worded in a slightly different manner.

Author(s)

Grant Schneider

---

pSDCFlig *Dwass, Steel, Critchlow, Fligner*

Description

Function to compute the P-value for the observed Dwass, Steel, Critchlow, Fligner W statistic.

Usage

```
pSDCFlig(x,g=NA,method=NA,n.mc=10000)
```

Arguments

x	Either a list or a vector containing the data.
g	If x is a vector, g is a required vector of group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo", or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the groups of data can be entered in either of two ways. For data a=1,2 and b=3,4,5 the following are equivalent:

`pSDCFlig(x=list(c(1,2),c(3,4,5)))` `pSDCFlig(x=c(1,2,3,4,5),g=c(1,1,2,2,2))`

Value

Returns a list with "NSM3Ch6MCp" class containing the following components:

n	a vector containing the number of observations in each of the k data groups
obs.stat	the observed W statistic for each of the k*(k-1)/2 comparisons
p.val	upper tail P-value corresponding to each W statistic

Author(s)

Grant Schneider

Examples

```
gizzards<-list(site.I=c(46,28,46,37,32,41,42,45,38,44),
               site.II=c(42,60,32,42,45,58,27,51,42,52),
               site.III=c(38,33,26,25,28,28,26,27,27,27),
               site.IV=c(31,30,27,29,30,25,25,24,27,30))
##Takes a little while
#pSDCFlig(gizzards,method="Monte Carlo")

##Shorter version for demonstration
pSDCFlig(gizzards[1:2],method="Asymptotic")
```

pSkilMack *Skillings-Mack*

Description

Function to compute the P-value for the observed Skillings-Mack SM statistic.

Usage

```
pSkilMack(x, b = NA, trt = NA, method = NA, n.mc = 10000)
```

Arguments

x	Either a matrix or a vector containing the data.
b	If x is a vector, b is a required vector of block labels. Otherwise, not used.
trt	If x is a vector, trt is a required vector of treatment labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the data can be entered in either of two ways. The following are equivalent: `pSkilMack(x=matrix(c(1,2,3,4,5,6),ncol=2,byrow=T))` `pSkilMack(x=c(1,2,3,4,5,6),b=`

Value

Returns a list with "NSM3Ch7p" class containing the following components:

k	number of treatments in the data
n	number of blocks in the data
ss	number of treatments per block
obs.stat	the observed D statistic
p.val	upper tail P-value

Author(s)

Grant Schneider

Examples

```
##Hollander, Wolfe, Chicken Example 7.8 Effect of Rhythmicity of a Metronome on Speech Fluency
rhythmicity<-matrix(c(3, 5, 15, 1, 3, 18, 5, 4, 21, 2, NA, 6, 0, 2, 17, 0, 2, 10, 0, 3, 8,
0, 2, 13),ncol=3,byrow=TRUE)
#pSkilMack(rhythmicity)
pSkilMack(rhythmicity,n.mc=5000)
```

pUmbrPK *Function to compute the P-value for the observed Mack-Wolfe Peak Known A_p distribution.*

Description

The function generalizes Harding's (1984) algorithm to quickly generate the distribution of A_p.

Usage

```
pUmbrPK(x,peak=NA,g=NA,method=NA, n.mc=10000)
```

Arguments

x	Either a list or a vector containing the data.
peak	An integer representing the known peak among the k data groups.
g	If x is a vector, g is a required vector of group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo", or "Asymptotic", indicating the desired distribution. When method=NA, and there are ties in the data, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used. When method=NA and there are no ties in the data, if sum(n)<=200, the "Exact" method will be used to compute the A_p distribution. Otherwise, the "Asymptotic" method will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the groups of data can be entered in either of two ways. For data a=1,2 and b=3,4,5 the following are equivalent:

`pUmbrPK(x=list(c(1,2),c(3,4,5)))` `pUmbrPK(x=c(1,2,3,4,5),g=c(1,1,2,2,2))`

Value

Returns a list with "NSM3Ch6p" class containing the following components:

n	a vector containing the number of observations in each of the data groups
obs.stat	the observed A_p statistic
p.val	the upper tail P-value

Author(s)

Grant Schneider

References

Harding, E. F. "An efficient, minimal-storage procedure for calculating the Mann-Whitney U, generalized U and similar distributions." Applied statistics (1984): 1-6.

Examples

```
##Hollander-Wolfe-Chicken Example 6.3 Fasting Metabolic Rate of White-Tailed Deer
x<-c(36,33.6,26.9,35.8,30.1,31.2,35.3,39.9,29.1,43.4,44.6,54.4,48.2,55.7,50,53.8,53.9,62.5,46.6,
44.3,34.1,35.7,35.6,31.7,22.1,30.7)
g<-c(rep(1,7),rep(2,3),rep(3,5),rep(4,4),rep(5,4),rep(6,3))

pUmbrPK(x,4,g,"Exact")
pUmbrPK(x,4,g,"Asymptotic")
```

---

pUmbrPU *Mack-Wolfe Peak Unknown*

Description

Function to compute the P-value for the observed Mack-Wolfe Peak Unknown A_p-hat distribution.

Usage

```
pUmbrPU(x,g=NA,method=NA, n.mc=10000)
```

Arguments

x	Either a list or a vector containing the data.
g	If x is a vector, g is a required vector of group labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo", or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.

Details

The data entry is intended to be flexible, so that the groups of data can be entered in either of two ways. For data a=1,2 and b=3,4,5 the following are equivalent:

`pUmbrPU(x=list(c(1,2),c(3,4,5)))` `pUmbrPU(x=c(1,2,3,4,5),g=c(1,1,2,2,2))`

Value

Returns a list with "NSM3Ch6p" class containing the following components:

n	a vector containing the number of observations in each of the data groups
obs.stat	the observed A_p-hat statistic
p.val	the upper tail P-value

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 6.4 Learning Comprehension and Age
wechsler<-list("16-19"=c(8.62,9.94,10.06),"20-34"=c(9.85,10.43,11.31),"35-54"=c(9.98,10.69,11.40),
"55-69"=c(9.12,9.89,10.57),"70+"=c(4.80,9.18,9.27))

#pUmbrPU(wechsler,method="Monte Carlo",n.mc=20000)
pUmbrPU(wechsler,method="Monte Carlo",n.mc=1000)
```

---

pWNMT *Wilcoxon, Nemenyi, McDonald-Thompson*

Description

Function to compute the P-value for the observed Wilcoxon, Nemenyi, McDonald-Thompson R statistic.

Usage

```
pWNMT(x,b=NA,trt=NA,method=NA, n.mc=10000, standardized=FALSE)
```

Arguments

x	Either a matrix or a vector containing the data.
b	If x is a vector, b is a required vector of block labels. Otherwise, not used.
trt	If x is a vector, trt is a required vector of treatment labels. Otherwise, not used.
method	Either "Exact", "Monte Carlo" or "Asymptotic", indicating the desired distribution. When method=NA, "Exact" will be used if the number of permutations is 10,000 or less. Otherwise, "Monte Carlo" will be used.
n.mc	If method="Monte Carlo", the number of Monte Carlo samples used to estimate the distribution. Otherwise, not used.
standardized	If TRUE, divide the observed statistic by (nk(k+1)/12)^0.5 before returning.

Details

The data entry is intended to be flexible, so that the data can be entered in either of two ways. The following are equivalent: `pWNMT(x=matrix(c(1,2,3,4,5,6),ncol=2,byrow=T))` `pWNMT(x=c(1,2,3,4,5,6),b=c(1,1,2,`

Value

Returns a list with "NSM3Ch7MCp" class containing the following components:

k	number of treatments
n	number of blocks
obs.stat	the observed R* statistic for each of the k*(k-1)/2 comparisons
p.val	upper tail P-value corresponding to each observed R statistic

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 7.3 Rounding First Base
RoundingTimes<-matrix(c(5.40, 5.50, 5.55, 5.85, 5.70, 5.75, 5.20, 5.60, 5.50, 5.55, 5.50, 5.40,
5.90, 5.85, 5.70, 5.45, 5.55, 5.60, 5.40, 5.40, 5.35, 5.45, 5.50, 5.35, 5.25, 5.15, 5.00, 5.85,
5.80, 5.70, 5.25, 5.20, 5.10, 5.65, 5.55, 5.45, 5.60, 5.35, 5.45, 5.05, 5.00, 4.95, 5.50, 5.50,
5.40, 5.45, 5.55, 5.50, 5.55, 5.55, 5.35, 5.45, 5.50, 5.55, 5.50, 5.45, 5.25, 5.65, 5.60, 5.40,
5.70, 5.65, 5.55, 6.30, 6.30, 6.25),nrow = 22,byrow = TRUE,dimnames = list(1 : 22,
c("Round Out", "Narrow Angle", "Wide Angle")))

pWNMT(RoundingTimes,n.mc=2500)
```

---

qKolSmirnLSA *Quantile function for the asymptotic distribution of the Kolmogorov-Smirnov J* statistic.*

Description

This function computes the Q() function defined in Section 5.4 of Hollander, Wolfe, and Chicken on a grid and then searches for the cutoff based on alpha.

Usage

```
qKolSmirnLSA(alpha)
```

Arguments

alpha	A numeric value between 0 and 1.

Value

Returns the upper tail cutoff at or below user-specified alpha

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Section 5.4 LSA
qKolSmirnLSA(.05)
```

RFPW *Randles-Fligner-Policello-Wolfe*

Description

Function to compute the P-value for the observed Randles-Fligner-Policello-Wolfe V statistic.

Usage

```
RFPW(z)
```

Arguments

z	A vector containing the data.

Value

Returns a list containing:

obs.stat	the observed V statistic
p.val	the asymptotic two-sided P-value

Author(s)

Grant Schneider

Examples

```
##Hollander-Wolfe-Chicken Example 3.10 Percentage Chromium in Stainless Steel
table3.9.subset<-c(17.4,17.9,17.6,18.1,17.6)
RFPW(table3.9.subset)
```

RSS *Ranked-Set Sample*

Description

Function to obtain a ranked-set sample of given set size and number of cycles based on a specified auxiliary variable.

Usage

```
RSS(k,m,ranker)
```

Arguments

k	set size
m	number of cycles
ranker	auxiliary variable used for judgment ranking

Value

Returns a vector of the indices corresponding to the observations selected to be in the RSS.

Author(s)

Grant Schneider

Examples

```
##Simulate 100 observations of a response variable we are interested in
##and an auxiliary variable we use for ranking

set.seed(1)
response<-rnorm(100)
auxiliary<-rnorm(100)

##Get the indices for a ranked-set sample with set size 3 and 2 cycles
RSS(2,3,auxiliary) #Tells us to measure observations 2, 19, 32,..., 91

##Alternatively, get the responses for those observations.
##In practice, response will not be available ahead of time.
response[RSS(2,3,auxiliary)]
```

sen.adichie *Function to test for parallel lines.*

Description

This code tests for parallel lines based on chapter 9 of Hollander, Wolfe, & Chicken, Nonparametric Statistical Methods, 3e.

Usage

```
sen.adichie(z, example=F, r=3)
```

Arguments

z	a list of paired vectors. Each item in the list is a set of two paired vectors in the form of a matrix. The first column of each matrix is the x vector, the second in the y vector.
example	if true, analyzes the data from Example 9.5
r	determines the amount of rounding. Increase it if your P-values are coming out as 0 or 1.

Author(s)

Eric Chicken

Examples

```
##Example 9.5 Hollander-Wolfe-Chicken##
sen.adichie(example=TRUE)
```

svr.df *Susarla-van Ryzin*

Description

Function to compute the Susarla-van Ryzin estimator

Usage

```
svr.df (z, delta, lambda.hat=0.001, alpha = 3, npoints=2053)
```

Arguments

z	the vector of zi = min(Xi, Yi)
delta	the vector of indicators which is 1 when Xi<=Yi and 0 otherwise
lambda.hat	the estimate of lambda from the data
alpha	the degree of faith in F0
npoints	the number of estimated points returned

Value

Returns a list containing:

x	the x values
F.hat	the Susarla-van Ryzin estimator

Note

Requires the survival library.

Author(s)

Rachel Becvarik

Examples

```
hodgkins.affected<-matrix(c(1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
0, 1, 346, 141, 296, 1953, 1375, 822, 2052, 836, 1910, 419, 107, 570, 312,1818, 364, 401, 1645,
330, 1540, 688, 1309, 505, 1378, 1446, 86),nrow=2,byrow=TRUE)
svr.df(hodgkins.affected[2,], hodgkins.affected[1,])
```

---

tc *Guess-Hollander-Proschan*

Description

Function to compute the asymptotic P-value for the observed Guess-Hollander-Proschan T_1 statistic.

Usage

```
tc(x, tau, alternative = "two.sided")
```

Arguments

x	a vector of data of length n
tau	the known value of the turning point,T
alternative	the direction of the alternative hypothesis. The choices are two.sided, idmrl, and dimrl with the default value being two.sided.

Value

The function returns a list with four elements:

T1	the value of the idmrl statistic
T1*	the standardized value of the idmrl statistic
p	the corresponding probability for T1*
sigma.hat	the standard deviation for T1

Author(s)

Rachel Becvarik

Examples

```
tb<-c(43, 45, 53, 56, 56, 57, 58, 66, 67, 73, 74, 79, 80, 80, 81, 81, 81, 82, 83, 83, 84, 88,
89, 91, 91, 92, 92, 97, 99, 99, 100, 100, 101, 102, 102, 102, 103, 104, 107, 108, 109,
113, 114, 118, 121, 123, 126, 128, 137, 138, 139, 144, 145, 147, 156, 162, 174, 178, 179, 184,
191, 198, 211, 214, 243, 249, 329, 380, 403, 511, 522, 598)
tc(tb, tau=91.9, alt="dimrl")
tc(tb, tau=91.9, alt="idmrl")
```

---

theil *Function to estimate and perform tests on the slope and intercept of a simple linear model.*

Description

This code estimates and performs tests on the slope and intercept of a simple linear model. Based on chapter 9 of Hollander, Wolfe & Chicken, Nonparametric Statistical Methods, 3e.

Usage

```
theil(x=NULL, y=NULL, alpha=0.05, beta.0=0, type="t",
      example=FALSE, r=3, slopes=F, doplot=TRUE)
```

Arguments

x	first data vector
y	second data vector
alpha	the significance level
beta.0	the null hypothesized value
type	can be "t" (two-sided), "u" (upper) or "l" (lower). The type refers both to the test and the confidence interval.
example	if true, will analyze the data from Example 9.1
r	the number of places for rounding. Increase it if your P-values are coming out as 0 or 1.
slopes	if true, will print all n(n-1)/2 slopes
doplot	if true, will plot the data and estimated line

Value

Returns a list with "NSM3Ch9ChickFn" class containing the following components:

alpha	same as input argument
beta.0	same as input argument
type	same as input argument
r	same as input argument
slopes	same as input argument
C.stat	the observed C statistic
C.bar	the observed C.bar statistic
alpha.hat	the observed alpha.hat statistic
beta.hat	the observed beta.hat statistic
slopes.table	table containing all n(n-1)/2
p.val	the P-value corresponding to the selected type of test/confidence interval
L	the lower endpoint of the confidence interval
U	the upper endpoint of the confidence interval

Author(s)

Eric Chicken

Examples

```
##Example 9.1 Hollander-Wolfe-Chicken##
theil (x, y, example=TRUE, slopes=TRUE)
```

---

zelen.test *Function to perform Zelen's test.*

Description

Zelen's test based on section 10.4 of Hollander, Wolfe, & Chicken, Nonparametric Statistical Methods, 3e.

Usage

```
zelen.test(z, example=F, r=3)
```

Arguments

z	data as an array of k 2x2 matrices. Small data sets only!
example	if true, analyzes the data from comment 24 of Chapter 10
r	determines the amount of rounding. Increase it if your P-values are coming out as 0 or 1.

Author(s)

Eric Chicken

Examples

```
##Chapter 10 Coment 24 Hollander-Wolfe-Chicken##
zelen.test(example=TRUE)
```

Index