

Package ‘wPerm’

October 12, 2022

Type Package

Title Permutation Tests

Version 1.0.1

Date 2015-11-03

Author Neil A. Weiss

Maintainer Neil A. Weiss <neil.a.weiss@asu.edu>

Description Supplies permutation-test alternatives to traditional hypothesis-test procedures such as two-sample tests for means, medians, and standard deviations; correlation tests; tests for homogeneity and independence; and more. Suitable for general audiences, including individual and group users, introductory statistics courses, and more advanced statistics courses that desire an introduction to permutation tests.

License GPL (>= 2)

Depends R (>= 3.1.0)

NeedsCompilation no

Repository CRAN

Date/Publication 2015-11-04 02:08:45

R topics documented:

wPerm-package	2
bmi	2
control	3
elmendorf	4
energy	5
experimental	6
learning	6
perm.hom.test	7
perm.ind.loc	9
perm.ind.spread	11
perm.ind.test	13
perm.oneway.anova	15

perm.paired.loc	16
perm.relation	18
print.perm.cs.hom	20
print.perm.oneway.anova	20
print.perm.paired.loc	21
print.perm.ts.ind	22
print.perm.two.var	23
relig.and.ed	23
salary	24
self	25
spouse.ages	26
water	26

Index	28
--------------	-----------

wPerm-package	<i>Permutation Tests</i>
---------------	--------------------------

Description

Supplies permutation-test alternatives to traditional hypothesis-test procedures such as two-sample tests for means, medians, and standard deviations; correlation tests; tests for homogeneity and independence; and more. Suitable for general audiences, including individual and group users, introductory statistics courses, and more advanced statistics courses that desire an introduction to permutation tests.

Details

Package:	wPerm
Type:	Package
Version:	1.0.1
Date:	2015-11-03
License:	GPL (>= 2)

Author(s)

Neil A. Weiss
 Maintainer: Neil A. Weiss <neil.a.weiss@asu.edu>

bmi	<i>Denosumab and Osteoporosis</i>
-----	-----------------------------------

Description

Body mass index for postmenopausal women with osteoporosis given six different treatments of Denosumab.

Usage

```
data("bmi")
```

Format

A data frame with 236 observations on the following 2 variables.

BMI a numeric vector

TREATMENT a factor with levels 100-mg 14-mg 210-mg 60-mg Placebo

Details

A clinical study was conducted to see whether an antibody called denosumab is effective in treatment of osteoporosis of postmenopausal women. In the article cited below, researchers reported on a study where women with osteoporosis were randomly assigned into groups that received either a placebo, or a six-month regimen of Denosumab at doses of 14 mg, 60 mg, 100 mg, or 210 mg. The bmi dataset provides data on the body-mass indexes (BMI) of the women in each treatment group.

Source

McClung, M., et al. (2006) Denosumab in Postmenopausal Women with Low Bone Mineral Density. *New England Journal of Medicine*, **354**, pp. 821-831.

Examples

```
data(bmi)
str(bmi)
attach(bmi)
plot(BMI ~ TREATMENT)
detach(bmi)
```

control

Exam Scores for Algebra Control Group

Description

The final-exam scores (out of 40 possible) for a control group of 41 algebra students.

Usage

```
data("control")
```

Format

The format is: num [1:41] 36 35 35 33 32 32 31 29 29 28 ...

Details

One year at Arizona State University, the algebra course director decided to experiment with a new teaching method that might reduce variability in final-exam scores by eliminating lower scores. The director randomly divided the algebra students who were registered for class at 9:40 A.M. into two groups. One of the groups, called the *control group*, was taught the usual algebra course; the other group, called the *experimental group*, was taught by the new teaching method. Both classes covered the same material, took the same unit quizzes, and took the same final exam at the same time. The final-exam scores (out of 40 possible) for the students in the control group are provided in the control dataset.

Examples

```
data(control)
str(control)
boxplot(control)
qqnorm(control)
```

elmendorf

Elmendorf Tear-Strength Data

Description

Elmendorf tear strengths, in grams, for independent samples of Brand A and Brand B vinyl floor coverings.

Usage

```
data("elmendorf")
```

Format

A data frame with 20 observations on the following 2 variables.

BRAND a factor with levels BRAND.A BRAND.B

STRENGTH a numeric vector

Details

Variation within a method used for testing a product is an essential factor in deciding whether the method should be employed. Indeed, when the variation of such a test is high, ascertaining the true quality of a product is difficult. The Elmendorf tear test is used to evaluate material strength for various manufactured products. In the article cited below, researchers investigated the variation of that test. For one aspect of the study, they randomly and independently obtained the data in "elmendorf" on Elmendorf tear strength, in grams, of two different brands of vinyl floor coverings.

Source

Phillips, A., Jeffries, R., Schneider, J., and Frankoski, S. (1997) Using Repeatability and Reproducibility Studies to Evaluate a Destructive Test Method. *Quality Engineering*, **10**, pp. 283-290.

Examples

```
data(elmendorf)
str(elmendorf)
plot(elmendorf)
attach(elmendorf)
detach(elmendorf)
```

energy

Energy Consumption

Description

Last year's energy consumptions for independent random samples of households in the four U.S. regions.

Usage

```
data("energy")
```

Format

A data frame with 20 observations on the following 2 variables.

ENERGY a numeric vector

REGION a factor with levels Midwest Northeast South West

Details

The Energy Information Administration gathers data on residential energy consumption and expenditures and publishes its findings in *Residential Energy Consumption Survey*. Independent random samples of households in the four U.S. regions yielded the data on last year's energy consumptions presented in the energy dataset. The data are displayed to the nearest 10 million BTU.

Examples

```
data(energy)
str(energy)
attach(energy)
plot(ENERGY ~ REGION)
detach(energy)
```

experimental

Exam Scores for Algebra Experimental Group

Description

The final-exam scores (out of 40 possible) for an experimental group of 20 algebra students.

Usage

```
data("experimental")
```

Format

The format is: num [1:20] 36 35 35 31 30 29 27 27 26 23 ...

Details

One year at Arizona State University, the algebra course director decided to experiment with a new teaching method that might reduce variability in final-exam scores by eliminating lower scores. The director randomly divided the algebra students who were registered for class at 9:40 A.M. into two groups. One of the groups, called the *control group*, was taught the usual algebra course; the other group, called the *experimental group*, was taught by the new teaching method. Both classes covered the same material, took the same unit quizzes, and took the same final exam at the same time. The final-exam scores (out of 40 possible) for the students in the experimental group are provided in the experimental dataset.

Examples

```
data(experimental)
str(experimental)
boxplot(experimental)
qqnorm(experimental)
```

learning

Social Class and Nursery-Rhyme Knowledge

Description

Contingency table of social class and nursery-rhyme knowledge for 66 children in kindergarten through second grade.

Usage

```
data("learning")
```

Format

A data frame with 2 observations on the following 4 variables.

SOCIAL_CLASS a factor with levels Middle Working

A_few a numeric vector

Some a numeric vector

Lots a numeric vector

Details

M. Stuart et al. studied various aspects of grade-school children and their mothers. The researchers gave a questionnaire to parents of 66 children in kindergarten through second grade. Two social-class groups, middle and working, were identified based on the mother's occupation. One aspect of the study cross-classified social class (of the mother) and nursery-rhyme knowledge (of the child).

Source

Stuart, M., Dixon, M., Masterson, J., and Quinlan, P. (1998) Learning to Read at Home and at School. *British Journal of Educational Psychology*, **68**, pp. 3-14.

Examples

```
data(learning)
str(learning)
learning
```

perm.hom.test	<i>Permutation Homogeneity Test</i>
---------------	-------------------------------------

Description

Performs a permutation (randomization) test for homogeneity of one variable on two or more populations, using chi-square as the test statistic.

Usage

```
perm.hom.test(x, type = c("cont", "flat", "raw"),
             variable = NULL, R = 9999)
```

Arguments

x	a data frame (see details below).
type	a character string indicating the type of data frame; must be one of "cont" (default), "flat", or "raw".
variable	an optional character string that gives the name of the variable whose distributions are to be compared.
R	number of replications (default = 9999).

Details

The null hypothesis is that the populations are homogeneous with respect to the variable under consideration. The alternative hypothesis is that the populations are nonhomogeneous with respect to the variable under consideration.

Types of data frames permitted:

cont: In this type of data frame, the first variable gives either the possible values of the variable under consideration or the populations. The remaining variables give the observed frequencies.

flat: This type of data frame consists of three variables. The first two variables give the pairs of possible values of the variable under consideration and the populations; the third variable gives the frequencies of the pairs.

raw: This type of data frame consists of two variables, which give the raw data of the variable-values and populations.

Value

A list with class "perm.cs.hom" containing the following components:

Perm.values	the values of chi-square obtained from the permutations.
Header	the main title for the output.
Variable	the name of the variable whose distributions are to be compared or NULL.
Statistic	the statistic used for the permutation test; here, always chi.square.
Observed	the value of the chi-square statistic for the observed data.
n	the (total) sample size.
Null	the null hypothesis; here, always homogeneous.
Alternative	the alternative hypothesis; here, always nonhomogeneous.
P.value	the P-value or a statement like $P < 0.001$.
p.value	the P-value.

Author(s)

Neil A. Weiss

Examples

```
# Self-concept for independent random samples of sighted and blind
# Indian adolescents.
data("self")
str(self)
self
# Note that self is in the form of a contingency table ("cont").

# Permutation homogeneity test to decide whether a difference exists in
# self-concept distributions between sighted and blind Indian adolescents,
# using 999 replications.
perm.hom.test(self, "cont", "Self-concept", 999)
# Or, equivalently, since "cont" is the default "type":
perm.hom.test(self, variable = "Self-concept", R = 999)
```

perm.ind.loc

Independent-Samples Permutation Test for Difference in Location

Description

Performs a permutation (randomization) test for difference in location based on independent samples from two populations.

Usage

```
perm.ind.loc(x, y, parameter, stacked = TRUE, variable = NULL,
            alternative = c("two.sided", "less", "greater"),
            R = 9999)
```

Arguments

x	a numeric vector of observations of the variable (stacked case) or a numeric vector of data values representing the first of the two samples (unstacked case).
y	a vector of corresponding population identifiers (stacked case) or a numeric vector of data values representing the second of the two samples (unstacked case).
parameter	the location parameter under consideration (e.g., mean, trimmed mean).
stacked	a logical value (default TRUE) indicating whether the data are stacked.
variable	an optional character string that gives the name of the variable under consideration; ignored if stacked is TRUE.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "less", or "greater".
R	number of replications (default = 9999)

Details

The null hypothesis is that the distributions of the variable on the two populations are identical—"identical".

The possible alternative hypotheses are:

Two tailed ("two.sided"): The distribution of the variable on the first population has either systematically smaller values or systematically larger values than that of the variable on the second population—"shifted".

Left tailed ("less"): The distribution of the variable on the first population has systematically smaller values than that of the variable on the second population—"shifted.left".

Right tailed ("greater"): The distribution of the variable on the first population has systematically larger values than that of the variable on the second population—"shifted.right".

Value

A list with class "perm.ts.ind" containing the following components:

Stacked	TRUE if the data are stacked, FALSE otherwise.
Perm.values	the values of the test statistic obtained from the permutations.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL.
Pop.1	the first population.
Pop.2	the second population.
n.1	the sample size for the first population.
n.2	the sample size for the second population.
Statistic	the test statistic.
Observed	the observed value of the test statistic.
Null	the null hypothesis; here, always identical.
Alternative	the alternative hypothesis.
P.value	the P-value or a statement like $P < 0.001$.
p.value	the P-value.

Note

For the permutation test, we need to assume that, under the null hypothesis, the two distributions are identical (i.e., the variable under consideration has the same distribution on both populations). If the two distributions have the same shape and spread, then a null hypothesis of equal population means or equal population medians implies that the two distributions are identical.

Author(s)

Neil A. Weiss

Examples

```
# Annual salaries, in thousands of dollars, for independent samples of
# faculty in private and public institutions.
data("salary")
str(salary)
attach(salary)
# Note that the data are stacked.

# Independent-samples permutation test to decide whether there is a
# difference in location for salaries of faculty in private and public
# institutions, using the mean as the location parameter.
perm.ind.loc(SALARY, TYPE, mean)

# Independent-samples permutation test to decide whether faculty in private
# institutions have systematically larger salaries than those in public
# institutions, using the 20% trimmed mean as the location parameter.
```

```
tr20.mean <- function(x) mean(x, trim = 0.20)
perm.ind.loc(SALARY, TYPE, tr20.mean, alternative = "greater")

detach(salary) # clean up.
```

perm.ind.spread *Permutation Test for Difference in Spread*

Description

Performs a permutation (randomization) test for difference in spread (variation) based on independent samples from two populations.

Usage

```
perm.ind.spread(x, y, parameter, stacked = TRUE, variable = NULL,
               alternative = c("two.sided", "less", "greater"),
               R = 9999)
```

Arguments

x	a numeric vector of observations of the variable (stacked case) or a numeric vector of data values representing the first of the two samples (unstacked case).
y	a vector of corresponding population identifiers (stacked case) or a numeric vector of data values representing the second of the two samples (unstacked case).
parameter	the spread parameter under consideration (e.g., sd, var).
stacked	a logical value (default TRUE) indicating whether the data are stacked.
variable	an optional character string that gives the name of the variable under consideration; ignored if stacked is TRUE.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "less", or "greater".
R	number of replications (default = 9999).

Details

The null hypothesis is that the distributions of the variable on the two populations are identical—"identical".

The possible alternative hypotheses are:

Two tailed ("two.sided"): The distribution of the variable on the first population has a different spread than that of the variable on the second population—"different.spread".

Left tailed ("less"): The distribution of the variable on the first population has a smaller spread than that of the variable on the second population—"smaller.spread".

Right tailed ("greater"): The distribution of the variable on the first population has a larger spread than that of the variable on the second population—"larger.spread".

Value

A list with class "perm.ts.ind" containing the following components:

Stacked	TRUE if the data are stacked, FALSE otherwise.
Perm.values	the values of the test statistic obtained from the permutations.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL.
Pop.1	the first population.
Pop.2	the second population.
n.1	the sample size for the first population.
n.2	the sample size for the second population.
Statistic	the test statistic.
Observed	the observed value of the test statistic.
Null	the null hypothesis; here, always identical.
Alternative	the alternative hypothesis.
P.value	the P-value or a statement like $P < 0.001$.
p.value	the P-value.

Author(s)

Neil A. Weiss

Examples

```
# Manufacturers use the Elmendorf tear test to evaluate material
# strength for various manufactured products.
#
# Elmendorf tear strength, in grams, of two different vinyl floor
# coverings, Brand A and Brand B.
data("elmendorf")
str(elmendorf)
# Note that the data are stacked.

# Permutation test to decide whether there is a difference in spread of
# tear strength for Brand A and Brand B vinyl floor coverings, using the
# standard deviation as the spread parameter.
attach(elmendorf)
perm.ind.spread(STRENGTH, BRAND, sd)

detach(elmendorf) # clean up

# Final-exam scores (out of 40 possible) for two groups of algebra
# students. One group, called the control group, was taught the usual
# algebra course; the other group, called the experimental group, was
# taught by a new teaching method.
data("control")
```

```

str(control)
data("experimental")
str(experimental)

# Permutation test to decide whether the new teaching method reduces
# variation in final-exam scores, using the variance as the spread
# parameter.
perm.ind.spread(control, experimental, var, stacked = FALSE,
variable = "Score", alternative = "greater")

```

perm.ind.test	<i>Permutation Independence Test</i>
---------------	--------------------------------------

Description

Performs a permutation (randomization) test for independence of two variables, using chi-square as the test statistic.

Usage

```

perm.ind.test(x, type = c("cont", "flat", "raw"),
var.names = NULL, R = 9999)

```

Arguments

x	a data frame (see details below).
type	a character string indicating the type of data frame; must be one of "cont" (default), "flat", or "raw".
var.names	an optional character string of length two that gives the names of the variables under consideration; if omitted Var.1 and Var.2 are used.
R	number of replications (default = 9999).

Details

The null hypothesis is that the two variables are not associated (i.e., are independent). The alternative hypothesis is that the two variables are associated (i.e., are dependent).

Types of data frames permitted:

cont: In this type of data frame, the first variable gives the possible values of one of the two variables under consideration. The remaining variables of the data frame give the observed frequencies.

flat: This type of data frame consists of three variables. The first two variables give the pairs of possible values of the two variables under consideration; the third variable of the data frame gives the frequencies of the pairs.

raw: This type of data frame consists of two variables, which give the raw data of the two variables under consideration.

Value

A list with class "perm.two.var" containing the following components:

Perm.values	the values of chi-square obtained from the permutations.
Header	the main title for the output.
Variable.1	the name of the first variable or Var.1
Variable.2	the name of the second variable or Var.2
Statistic	the statistic used for the permutation test; here, always chi.square.
Observed	the value of the chi-square statistic for the observed data.
n	the sample size.
Null	the null hypothesis; here, always nonassociated.
Alternative	the alternative hypothesis; here, always associated.
P.value	the P-value or a statement like $P < 0.001$.
p.value	the P-value.

Author(s)

Neil A. Weiss

Examples

```
# Religious belief vs education for a sample of 509 people.
data("relig.and.ed")
str(relig.and.ed)
relig.and.ed
# Note that relig.and.ed is in the form of a flat contingency table ("flat").

# Permutation independence test to decide whether an association exists
# between religiosity and education, using 999 replications.
perm.ind.test(relig.and.ed, "flat", c("Religiosity", "Education"), 999)

# Social class vs nursery-rhyme knowledge for a sample of 66 grade-school
# children.
data("learning")
str(learning)
learning
# Note that the learning data is in the form of a contingency table ("cont").

# Permutation independence test to decide whether an association exists
# between social class and nursery-rhyme knowledge, using 999 replications.
perm.ind.test(learning, "cont", c("Social class", "Nursery-rhyme knowledge"), 999)
# Or, equivalently, since "cont" is the default "type":
perm.ind.test(learning, var.names = c("Social class", "Nursery-rhyme knowledge"), R = 999)
```

perm.oneway.anova *Permutation One-Way ANOVA Test*

Description

Performs a permutation (randomization) test for location, using trimmed data (trim = 0 gives untrimmed data) on several independent samples.

Usage

```
perm.oneway.anova(x, y, trim = 0, ford = NULL, R = 9999)
```

Arguments

x	a (non-empty) vector of observations of the (response) variable.
y	a vector of the corresponding populations (levels of the factor).
trim	the fraction (0 to 0.5) of observations to be trimmed from each sample; default is 0.
ford	an optional integer vector giving the change from alphabetical order of the populations to some other desired order.
R	number of replications (default = 9999).

Details

The null hypothesis is that the distributions of the variable are identical on all the populations. The alternative hypothesis is that the distributions of the variable have systematically larger values on some of the populations than on others.

Value

A list with class "perm.oneway.anova" containing the following components:

Perm.values	the values of the test statistic obtained from the permutations.
Header	the main title for the output.
Response	the name of the (response) variable.
Factor	the name of the factor.
Levels	the populations (levels of the factor).
n	the sample sizes.
Mean	the sample means.
SD	the sample standard deviations.
Statistic	the test statistic; here, always F.trim.
Observed	the observed value of the test statistic.
P.value	the P-value or a statement like $P < 0.001$.
p.value	the P-value.
Trim	the trim value.

Author(s)

Neil A. Weiss

Examples

```
# Last year's energy consumptions, to the nearest 10 million BTU, for
# independent random samples of households in the four U.S. regions.
data("energy")
str(energy)
attach(energy)

# Permutation one-way ANOVA to decide whether the energy distributions
# have systematically larger values in some U.S. regions than in others.
# Regions ordered to Northeast, Midwest, South, and West; 999 replications.
perm.oneway.anova(ENERGY, REGION, ford = c(2,1,3,4), R = 999)

detach(energy) # clean up
```

perm.paired.loc

Paired-Sample Permutation Test for Difference in Location

Description

Performs a permutation (randomization) test for difference in location based on a paired sample.

Usage

```
perm.paired.loc(x, y, parameter, variable = NULL,
               alternative = c("two.sided", "less", "greater"),
               R = 9999)
```

Arguments

x	a numeric vector of data values representing the first components of the pairs.
y	a numeric vector of data values representing the second components of the pairs.
parameter	the location parameter under consideration (e.g., mean, trimmed mean).
variable	an optional character string that gives the name of the variable under consideration.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "less", or "greater".
R	number of replications (default = 9999).

Details

The null hypothesis is that the distributions of the variable on the two populations are identical—"identical".

The possible alternative hypotheses are:

Two tailed ("two.sided"): The distribution of the variable on the first population has either systematically smaller values or systematically larger values than that of the variable on the second population—"shifted".

Left tailed ("less"): The distribution of the variable on the first population has systematically smaller values than that of the variable on the second population—"shifted.left".

Right tailed ("greater"): The distribution of the variable on the first population has systematically larger values than that of the variable on the second population—"shifted.right".

Value

A list with class "perm.paired.loc" containing the following components:

Perm.values	the values of the test statistic obtained from the permutations.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL.
Pop.1	the first population.
Pop.2	the second population.
n	the sample size.
Statistic	the test statistic.
Observed	the observed value of the test statistic.
Null	the null hypothesis; here, always identical.
Alternative	the alternative hypothesis.
P.value	the P-value or a statement like $P < 0.001$.
p.value	the P-value.

Author(s)

Neil A. Weiss

Examples

```
# Ages of a sample of 10 heterosexual spouses.
data("spouse.ages")
str(spouse.ages)
attach(spouse.ages)

# Paired-sample permutation test to decide whether there is a difference
# in location for age distributions of married men and married women,
# using the mean as the location parameter. Variable named "Age".
perm.paired.loc(HUSBAND, WIFE, mean, "Age")
```

```
# Paired-sample permutation test to decide whether married men have
# systematically greater ages than married women, using the 10% trimmed
# mean as the location parameter.
tr10.mean <- function(x) mean(x, trim = 0.10)
perm.paired.loc(HUSBAND, WIFE, tr10.mean, alternative = "greater")

detach(spouse.ages) # clean up.
```

perm.relation

Permutation Test for a Relationship

Description

Performs a permutation (randomization) test for a relationship (correlation, association) for two quantitative variables, using Pearson's r (product moment correlation coefficient), Spearman's ρ (rank correlation coefficient), or Kendall's tau as the test statistic.

Usage

```
perm.relation(x, y, method = c("pearson", "kendall", "spearman"),
             alternative = c("two.sided", "less", "greater"),
             R = 9999)
```

Arguments

x	a numeric vector of data values representing the first variable.
y	a numeric vector of data values representing the second variable.
method	a character string indicating which method is to be used for the test; one of "pearson" (default), "kendall", or "spearman".
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "less", or "greater".
R	number of replications (default = 9999).

Details

The null hypothesis is that there is no relationship between the variables.

The possible alternative hypotheses are:

Two tailed ("two.sided"): There is a relationship between the variables—"relation".

Left tailed ("less"): There is a negative relationship between the variables—"neg.relation".

Right tailed ("greater"): There is a positive relationship between the variables—"pos.relation".

Value

A list with class "perm.two.var" containing the following components:

Perm.values	the values of the test statistic obtained from the permutations.
Header	the main title for the output.
Variable.1	the name of the first variable.
Variable.2	the name of the second variable.
n	the sample size.
Statistic	the test statistic.
Observed	the observed value of the test statistic.
Null	the null hypothesis; here, always no relation.
Alternative	the alternative hypothesis.
P.value	the P-value or a statement like $P < 0.001$.
p.value	the P-value.

Author(s)

Neil A. Weiss

Examples

```
# Prices, in euros, of a 50cl bottle of water and distances, in meters,  
# of convenience stores from the Contemporary Art Museum in El Raval,  
# Barcelona.  
data("water")  
str(water)  
attach(water)  
  
# Permutation test to decide whether a negative relationship exists  
# between price and distance, using Pearson's r as the test statistic.  
perm.relation(PRICE, DISTANCE, alternative = "less")  
  
# Permutation test to decide whether a negative relationship exists  
# between price and distance, using Kendall's tau as the test statistic.  
perm.relation(PRICE, DISTANCE, "kendall", "less")  
  
# Permutation test to decide whether a negative relationship exists  
# between price and distance, using Spearman's rho as the test statistic.  
perm.relation(PRICE, DISTANCE, "spearman", "less")  
  
detach(water) # clean up.
```

print.perm.cs.hom *Print Output for Class "perm.cs.hom"*

Description

This is a method for the function `print()` to print objects of class "perm.cs.hom".

Usage

```
## S3 method for class 'perm.cs.hom'  
print(x, ...)
```

Arguments

`x` an object of class "perm.cs.hom".
`...` further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a permutation function with output list of class "perm.cs.hom".

Value

The `perm.cs.hom` object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[perm.hom.test](#)

print.perm.oneway.anova
 Print Output for Class "perm.oneway.anova"

Description

This is a method for the function `print()` to print objects of class "perm.oneway.anova".

Usage

```
## S3 method for class 'perm.oneway.anova'  
print(x, ...)
```

Arguments

- x an object of class "perm.oneway.anova".
- ... further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a permutation function with output list of class "perm.oneway.anova".

Value

The perm.oneway.anova object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[perm.oneway.anova](#)

print.perm.paired.loc *Print Output for Class "perm.paired.loc"*

Description

This is a method for the function print() to print objects of class "perm.paired.loc".

Usage

```
## S3 method for class 'perm.paired.loc'  
print(x, ...)
```

Arguments

- x an object of class "perm.paired.loc".
- ... further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a permutation function with output list of class "perm.paired.loc".

Value

The perm.paired.loc object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[perm.paired.loc](#)

print.perm.ts.ind *Print Output for Class "perm.ts.ind"*

Description

This is a method for the function `print()` to print objects of class "perm.ts.ind".

Usage

```
## S3 method for class 'perm.ts.ind'  
print(x, ...)
```

Arguments

x	an object of class "perm.ts.ind".
...	further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a permutation function with output list of class "perm.ts.ind".

Value

The perm.ts.ind object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[perm.ind.loc](#), [perm.ind.spread](#)

print.perm.two.var *Print Output for Class "perm.two.var"*

Description

This is a method for the function `print()` to print objects of class "perm.two.var".

Usage

```
## S3 method for class 'perm.two.var'  
print(x, ...)
```

Arguments

x an object of class "perm.two.var".
... further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a permutation function with output list of class "perm.two.var".

Value

The perm.two.var object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[perm.relation](#), [perm.ind.test](#)

relig.and.ed *Religiosity and Education*

Description

Flat contingency table for religiosity and educational attainment for a sample of 509 people world-wide.

Usage

```
data("relig.and.ed")
```

Format

A data frame with 12 observations on the following 3 variables.

RELIGIOUSITY a factor with levels Religious Not religious Atheist Do not know

EDUCATION a factor with levels Basic Secondary Advanced

COUNT a numeric vector

Details

A worldwide poll on religion was conducted by WIN-Gallup International and published as the document *Global Index of Religiosity and Atheism*. One question involved religious belief and educational attainment. The data in the relig.and.ed dataset are based on the answers to that question.

Examples

```
data(relig.and.ed)
str(relig.and.ed)
relig.and.ed
```

salary

Faculty Salaries

Description

Salaries, in thousands of dollars rounded to the nearest hundred, for independent random samples of 35 faculty members from private institutions and 30 faculty members from public institutions.

Usage

```
data("salary")
```

Format

A data frame with 65 observations on the following 2 variables.

TYPE a factor with levels PRIVATE PUBLIC

SALARY a numeric vector

Details

The American Association of University Professors (AAUP) conducts salary studies of college professors and publishes its findings in *AAUP Annual Report on the Economic Status of the Profession*. Independent random samples of 35 faculty members in private institutions and 30 faculty members in public institutions yielded the salaries, in thousands of dollars rounded to the nearest hundred, provided in the salary dataset.

Examples

```
data(salary)
str(salary)
plot(salary)
attach(salary)
detach(salary)
```

self

Self-Concept and Sightedness

Description

Contingency table on self-concept for independent random samples of sighted and blind Indian adolescents.

Usage

```
data("self")
```

Format

A data frame with 2 observations on the following 4 variables.

SIGHTEDNESS a factor with levels Blind Sighted

High a numeric vector

Moderate a numeric vector

Low a numeric vector

Details

Self-concept can be defined as the general view of oneself in terms of personal value and capabilities. A study of whether visual impairment affects self-concept was reported in the article cited below. The researchers classified self-concept as high, moderate, or low. Independent random samples of sighted and blind Indian adolescents gave the data on self-concept presented in the self dataset.

Source

Halder, S. and Datta, P. (2014) An Exploration into Self Concept: A Comparative Analysis between the Adolescents Who Are Sighted and Blind in India. *British Journal of Visual Impairment*, **30**, pp. 31-41.

Examples

```
data(self)
str(self)
self
```

`spouse.ages`*Ages of Married People*

Description

Ages, in years, of a random sample of 10 heterosexual married couples.

Usage

```
data("spouse.ages")
```

Format

A data frame with 10 observations on the following 2 variables.

HUSBAND a numeric vector

WIFE a numeric vector

Details

The U.S. Census Bureau publishes information on the ages of married people in *Current Population Reports*. A random sample of 10 heterosexual married couples gave the data on ages, in years, shown in the `spouse.ages` dataset.

Examples

```
data(spouse.ages)
str(spouse.ages)
attach(spouse.ages)
detach(spouse.ages)
```

`water`*Water Price vs Distance from Museum*

Description

Prices, in euros, of a 50cl bottle of water and distances, in meters, of convenience stores from the Contemporary Art Museum in El Raval, Barcelona.

Usage

```
data("water")
```

Format

A data frame with 10 observations on the following 2 variables.

DISTANCE a numeric vector

PRICE a numeric vector

Details

Does the price of a convenience-store item, such as a bottle of water, decrease as distance from the Contemporary Art Museum in El Raval, Barcelona, increases? A sample of 10 convenience stores yielded the data presented in the water dataset on price, in euros, of a 50cl bottle of water and distance, in meters, of the convenience store from the Contemporary Art Museum.

Source

Barcelona Field Studies Centre, <http://geographyfieldwork.com/SpearmansRank.htm>.

Examples

```
data(water)
str(water)
plot(water)
attach(water)
detach(water)
```

Index

- * **Kendall's tau**
 - perm.relation, 18
- * **Pearson's r**
 - perm.relation, 18
- * **Spearman's rho**
 - perm.relation, 18
- * **association test**
 - perm.relation, 18
- * **correlation test**
 - perm.relation, 18
- * **datasets**
 - bmi, 2
 - control, 3
 - elmendorf, 4
 - energy, 5
 - experimental, 6
 - learning, 6
 - relig.and.ed, 23
 - salary, 24
 - self, 25
 - spouse.ages, 26
 - water, 26
- * **distributions comparison**
 - perm.hom.test, 7
- * **homogeneity test**
 - perm.hom.test, 7
- * **independence test**
 - perm.ind.test, 13
- * **independent-samples inferences**
 - perm.ind.loc, 9
 - perm.ind.spread, 11
- * **inferences for several independent samples**
 - perm.oneway.anova, 15
- * **inference**
 - wPerm-package, 2
- * **one-way ANOVA**
 - perm.oneway.anova, 15
- * **package, datasets, graphs**
 - wPerm-package, 2
- * **paired-sample inferences**
 - perm.paired.loc, 16
- * **permutation one-way ANOVA test**
 - perm.oneway.anova, 15
- * **permutation tests**
 - perm.hom.test, 7
 - perm.ind.loc, 9
 - perm.ind.spread, 11
 - perm.ind.test, 13
 - perm.oneway.anova, 15
 - perm.paired.loc, 16
 - perm.relation, 18
 - wPerm-package, 2
- * **randomization tests**
 - perm.hom.test, 7
 - perm.ind.loc, 9
 - perm.ind.spread, 11
 - perm.ind.test, 13
 - perm.oneway.anova, 15
 - perm.paired.loc, 16
 - perm.relation, 18
 - wPerm-package, 2
- * **simulation**
 - wPerm-package, 2
- * **test for association**
 - perm.ind.test, 13
- * **tests for a relationship**
 - perm.relation, 18
- * **tests for difference in location**
 - perm.ind.loc, 9
 - perm.oneway.anova, 15
 - perm.paired.loc, 16
- * **tests for difference in spread**
 - perm.ind.spread, 11
- bmi, 2
- control, 3
- elmendorf, 4

energy, [5](#)
experimental, [6](#)

learning, [6](#)

perm.hom.test, [7](#), [20](#)
perm.ind.loc, [9](#), [22](#)
perm.ind.spread, [11](#), [22](#)
perm.ind.test, [13](#), [23](#)
perm.oneway.anova, [15](#), [21](#)
perm.paired.loc, [16](#), [22](#)
perm.relation, [18](#), [23](#)
print.perm.cs.hom, [20](#)
print.perm.oneway.anova, [20](#)
print.perm.paired.loc, [21](#)
print.perm.ts.ind, [22](#)
print.perm.two.var, [23](#)

relig.and.ed, [23](#)

salary, [24](#)
self, [25](#)
spouse.ages, [26](#)

water, [26](#)
wPerm (wPerm-package), [2](#)
wPerm-package, [2](#)