Research Report

KSTS/RR-90/003
Feb.7, 1990

New Environment for Statistical Inference (NESI)

by

R. Shibata and M. Takagiwa

R. Shibata and M. Takagiwa
Department of Mathematics
Faculty of Science and Technology
Keio University
Hiyoshi 3-14-1, Kohoku-ku
Yokohama, 223 Japan

Department of Mathematics
Faculty of Science and Technology
Keio University
© 1990 KSTS
Hiyoshi 3-14-1, Kohoku-ku, Yokohama, 223 Japan
New Environment for Statistical Inference (NESI)

by Ritei Shibata and Mutsumi Takagiwa

Department of Mathematics, Keio University, Japan

1. Introduction

NESI is a new environment for statistical inference on top of the S language (Becker, Chambers and Wilks (1988), The new S language, Wadsworth & Brooks/Cole). All routines are written by the S language. A few exceptions are m-sequence generation and calculation of Noncentral F-distribution, which require highly efficient computation. S is an interactive object oriented language in which basic unit is an expression, combination of operators and functions. An advantage of using such a language is not only that the any time the algorithm can be seen simply by typing the name of function but also that list is a basic data structure. It makes possible to combine the results of each statistical inference into one and specify it only by the name. There are three primary reasons behind our decision to produce the NESI. The first is to provide a vehicle for doing classical testings or inferences in the frame of modern data analysis. Such vehicle is missing in the currently available S environment. The second is to be able to experiment with such environment in instruction. Finally, we were interested in exploring a good organization of classical inference, particularly statistical testings, in such an advanced environment for data analysis.

2. Design Policy

There is no definite way to place classical inference in modern data analysis. Usually it is placed on the final stage, confirmative stage, but, even in a stage of exploring data, formal and objective inference will help user to make a decision. More importantly, classical testings are still main objectives in many field of statistical data analysis. For example, in pharmaceutical company or clinical trials, formal testing of effect or side effect of drugs are main concern. Even in such field of application, interactive environment like S will help user to analyze his/her data.
Basic idea of the design of NESI is, dividing routines into parts, that is, S functions, and combining those into one to do an inference. Dividing into parts will make system much simpler. For example, chisquare type testing is common in various testing procedures. As is seen in Section 7 a simple chisquare test is explicitly used in three places, testing for the equivalence of two categorical data (diff.test), oneway analysis of categorical data (oneway.test), independence test of two samples (indep.test). Basic function like Chisq.test is designed very general way. User can invoke it alone and also can use it indirectly through one of parent functions as above. Then the parent function will invoke it in an appropriate form. To systematically organize such basic functions we had to establish a standard of communications between functions, what is necessary and sufficient information which should be passed through. We believe that such organization will not only make system simpler but also clarify the relation of various statistical inferences. This consideration is also applied for communication between functions for analysis and graphics. As a side effect of such standardization, it becomes able to have each function a polymorphism. Each function understands any result of previous application of a function.

Another important point is how to design a good interface to the user, particularly to novice. A problem of novice in statistics is that it is hard to find an appropriate procedure to analyze his/her data. Interactive functions diff.test, oneway.test and indep.test in NESI will lead user to an appropriate procedure implied by the features of the given data i.e., for example, categorical or not, or by asking user if mathematical assumptions like equal scale or normality is acceptable or not. Also, to assist user’s judge, a graphical or numerical display is provided.

```r
> nesi(X,Y)
1: diff.test
2: indep.test
3: oneway.test
Selection: 1
1: paired
2: unpaired
Selection: 1
1: I know if x-y is normal or not
2: Judge by graphics
3: Judge by p-values
```
Selection: 2

In this stage a graphics like Fig.1.1 is shown.

```
X - Y
```

![Figure 1.1 Assistance for judgement of normality for paired data](image)

If the user selects "unpaired" at the previous stage, then he/she will see a graphics like Fig.1.2.

```
X       Y
```

![Figure 1.2 Assistance for judgement of normality for unpaired data](image)

For the case of `indep.test`, a graphical display of a bivariate normal test will be
3. Functions for statistical inference

In NESI, S function names for statistical inference basically begin with an upper case character. Names for testing functions further have a suffix test. Interactive functions and utilities have all lower case names. Interactive function will prompt user to choose one from possible inference functions. In due course, a graphical or numerical assistance is available for user to check mathematical assumptions as is described in the previous section.

3.1. Two sample problem

diff.test          interactive test for the difference of two samples
Bn.test           binomial test, signed test
F.test            F test for the equality of variances
Chisq.test        chisquare test for contingency table
Mc.test           McNemar test for the symmetry of contingency table
T.test            paired t test
Two.t.test        two sample t test
Welch.test        Aspin-Welch test
Wil.test          Wilcoxon rank sum test (U test)
3.2. One way analysis

one way test interactive one way analysis of variance with or without covariate

Chi.sq.test chi square test
Kruskal-Wallis rank test
One way test one way analysis (ANOVA)
One way.cov.test one way analysis with covariate (ANOCOVA)
Q test generalized Cochran Q test, Friedman test

3.3. Repeated t test

rep.t.test interactive repeated t test for the difference of locations
Rep.t.test repeated t test for the difference of locations

3.4. Association test for contingency table

table.assoc.test interactive association test for contingency table

Fisher exact test
McNemar test
Mantel-Haenszel test
Stuart-Maxwell test

3.5. Independence of two samples

indep.test interactive independence test
Chi.sq.test chi square test
t test for zero correlation
Kendall's tau and nonparametric test for zero correlation
Spearman's rank correlation and nonparametric test for zero correlation

3.6. Survival time objects

create a survival time object
survive
survival time object or not
is.survive
make a survival time object
as.survive
3.6.1. Estimation of survival function

- `survive.fun` interactive estimation of survival function
- `KpMeier` Kaplan-Meier estimate of survival function
- `Lifetable` survival function estimate by lifetable method

3.6.2. Test for the equivalence of two survival time objects

- `life.test` interactive equivalence test of two survival time objects
- `Cox.test` Cox-Mantel test
- `Logrank.test` Logrank test (Peto & Peto test)
- `Wil.test` generalized Wilcoxon test (Gehan's generalized test)
- `Z.test` Z test

3.7. Normality test

- `Bnormal.test` bivariate normality test
- `Normal.test` univariate normality test

3.8. M-sequence random number generation

- `Msequence` m-sequence random number generator for various distributions

3.9. Distributions

- `pf` central or noncentral F distribution
- `Binomial` binomial distribution
- `Hypergeometric` hypergeometric distribution
- `Wilcoxon` Wilcoxon rank sum distribution

3.10. Utilities

- `autocov` autocovariances
- `Cat` concatenation of two sets of categorical data
- `Conf` upper and lower confidence bounds
- `Show` multiple graphical representation of the result.
4. Graphical Representation of the result

The function `Show` is a powerful tool for graphical representation of the result of a function of inference in NESI. This function understands any result of inferences in NESI and draw an appropriate graphics, confidence bounds, rejection region(s), or a curve of survival function. The use of this function is simple. Invoke a graphic driver first and give as the first argument to `Show` the object returned by functions described in Section 3.

5. How to use (an example)

One of the easiest way of using various functions in NESI is through the function `nesi`. This is a tree based menu system. This function covers almost all testing functions,

```
% S
> library(nesi, first=T)
> printer() invoke a device driver
> nesi(X, Y)
> ... interactive use ...
> Show( d < - diff.test(X,Y) ) # graphical representation of the result of the testing
> .... interaction with user ...
> d
    # look at numerical result
> d$prob > 0.05  # test with significance level 0.05
> Show( d, tol=pretty(c(0.01)) )  # draw confidence bounds
```

6. Miscellaneous

6.1. Related functions

- `Rej` draw a rejection region for continuous distribution
- `Rej.d` draw a rejection region for discrete distribution
- `Show.r` show multiple plot of rejection regions

6.2. Objects

A specific object "survival time" is defined in NESI, which has an attribute "status" indicating death, censor(alive) or censored(dropout). Also time series, matrix, and categorical data are recognized.
Functions for testing basically returns a list with following components:

- statistic: the value of test statistic. Name of the component is that of the null distribution.
- p-value: p value under null hypothesis
- parameter(s): parameter(s) of null distribution
- alt: alternative hypothesis, "less", "two", "greater" (-1, 0, 1 for rep.t.test), or "one" (for chisquare type testings only).
- method: name of the method applied

Function for discrete distribution will return a list with following components:

- x: vector of integers
- prob: probability or cumulative probability on x.

6.3. Availability

Source code is available free of charge for non-commercial use. Easiest way to get the code is to send an electronic mail to "statlib@temper.stat.cmu.edu" or to "statlib@math.keio.ac.jp". The body of the e-mail should be "send nesi from s".

NESI is an experimental program. It has been tested extensively, but the authors take no responsibility for losses or damages resulting directly or indirectly from the use of this program. The authors appreciate of receiving any bug reports or comments from the user. NESI is an evolving system. Over time new features will be introduced, and existing features may be changed. NESI is an enhanced version of NEST on old S system. It took half a year to produce NEST. In due course, various people in our statistics group contributed to this NESI, including Hiroe Tsubaki, Toshiya Takahashi and Rina Fujiwara.

7. NESI function manuals

The followings are on-line manuals for functions in NESI.
Binomial

<table>
<thead>
<tr>
<th>Function</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbn</td>
<td>a, p, xmin=0, xmax=a</td>
</tr>
<tr>
<td>pbin</td>
<td>x, a, p</td>
</tr>
<tr>
<td>qbin</td>
<td>prob, a, p</td>
</tr>
<tr>
<td>rbin</td>
<td>nn, a, p</td>
</tr>
</tbody>
</table>

ARGUMENTS
- a: number of trials.
- p: probability of success.
- x: number of success.
- prob: vector of probabilities. Missing values (#s) are allowed.
- xmin: minimum of x, from where the probabilities should be computed.
- xmax: maximum of x, to where the probabilities should be computed.
- nn: sample size. If length(nn) is larger than 1, then length(nn) random values are returned.

VALUE
- The function dbn returns a list with components:
  - x: vector of integers; max(0, xmin):min(a, xmax)
  - prob: vector of probabilities on x.

Other functions return vector of cumulative probability (pbin), quantile (qbin), or random sample (rbin), respectively for Binomial distribution. The probability function is given by

\[ P_n(x) = \binom{a}{x} p^x (1-p)^{a-x} \]

EXAMPLES
- dbn(6, 0.4)

---

Binomial test

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x, n, p=0.5, alt=&quot;two&quot;</td>
<td>_BIN.test(x, n, p=0.5, alt=&quot;two&quot;)</td>
</tr>
</tbody>
</table>

ARGUMENTS
- x: number of success.
- n: number of trials.
- p: probability of success to be tested.
- alt: alternative hypothesis, "two" (not equal to p), "less" (less than p) or "greater" (greater than p).

VALUE
- a list representing the result of binomial test:
  - x: number of trials.
  - p: null hypothesis.
  - na: number of success.
  - prob: p-value.
  - method: the name of method applied.
alt alternative hypothesis.

EXAMPLES

x = rnorm(100)
y = runif(20)
Bn.test(y, 100) # median == 0 ?
y = rnorm(100)
d = x - y
Bn.test(sum(d > 0), length(d)) # signed test

\begin{tabular}{|c|c|}
\hline
**Bn.test** & **Test for Bivariate Normality** \\
\hline
Bn.test(x, y) & \\
\hline
\end{tabular}

ARGUMENTS

x,y numeric vectors. The x and y must have the same length. A two column matrix can be given to x. \texttt{NA}s are allowed but ignored at calculation.

VALUE

a list representing the result of bivariate normality test with the following components:

- \( z \) vector of differences \( Z_1, Z_2 \) and \( Z_3 \) among the number of pair of observations which fall in each four quadrants of the \( x \)-\( y \) plane.

\[
Z_1 = \frac{N_1 - N_3}{\sqrt{n \left( \frac{1}{2} + \frac{1}{\pi} \sin^{-1} R \right) - \frac{1 + R}{\pi}}},
\]

\[
Z_2 = \frac{N_2 - N_4}{\sqrt{n \left( \frac{1}{2} - \frac{1}{\pi} \sin^{-1} R \right) - \frac{1 - R}{\pi}}},
\]

and

\[
Z_3 = \frac{N_1 + N_3 - \frac{n}{2} - \frac{n}{\pi} \sin^{-1} R}{\sqrt{n \left( \frac{1}{4} \frac{1}{\pi^2} \left( \sin^{-1} R \right)^2 - \frac{1 - R^2}{\pi^2}\right)}},
\]

where \( N_i \) is the number of observations which fall in the \( i \)-th quadrant. The \( n \) is \texttt{length}(x)=\texttt{length}(y), and \( R \) is the sample correlation coefficient between \( x \) and \( y \). \( Z_1, Z_2 \) and \( Z_3 \) are asymptotically independent and normally distributed under the hypothesis.

- \texttt{prob} vector of p-values (asymptotic).
- alt alternative hypothesis. Always "two".
- \texttt{method} the name of method applied.
- \texttt{sigma} vector giving standard deviations of each value of \( z \).

REFERENCE

S-Function Documentation

Cat

Concatenation of two sets of categorical data

Cat

Cat(x, y)

ARGUMENTS

x, y categories.

VALUE

A category which is a concatenation of x and y.

EXAMPLES

cat(Cat(x), Cat(y))

Chiq.test

Chisquare test for contingency table

Chiq.test

Chiq.test(x, y, correct=TRUE, difference=FALSE)

ARGUMENTS

x, y categories or vectors. A contingency table (matrix) may be given for x. Table values must be non-negative. NA's are allowed but ignored at calculation.

correct If TRUE, Yates's continuity correction is applied. This correction is valid only for dichotomous categories.

difference If TRUE, chisquare type difference test is applied. If both x and y are vectors, a goodness of fit test is applied, regarding x as expected frequencies and y as observed frequencies, otherwise, frequencies in x and y with the same level are compared.

VALUE

A list representing the result of chisquare test:

- chiq chisquare statistic. Table x.
- degree of freedom. $(\text{levels}(x)-1) \times (\text{levels}(y)-1)$ or $(\text{length}(x)-1) \times (\text{length}(y)-1)$ when difference=TRUE.
- prob p-value (asymptotic).
- method the name of method applied.
- alt alternative hypothesis. Always "one".
- goodness.stat This component is TRUE if the test is a goodness of fit test. Otherwise this component is omitted.
- difference.test This component is TRUE if the test is a chisquare type difference test other than goodness of fit test.
- dimnames This component gives dimnames attribute of contingency table.

The definition of chisquare statistic when difference=FALSE is

$$\chi^2 = \sum_{ij} \frac{(n_{ij} - n_i \cdot n_j / n)^2}{n_i \cdot n_j / n}$$

where $n_{ij}$ is the $(i,j)$ element of table(x,y). If difference=TRUE and x and y are categories, then $n_{ij}$ and $n_{ij}$ are frequencies in x and y with the level $j$. If a goodness of fit test is applied, $\chi^2 = \sum (x - y)^2 / x$.
Chisq.test

SEE
diff.test, indep.test, oneway.test

EXAMPLES
Chisq.test(Cat.X, Cat.Y)

Conf

Conf(x, tol=pretty(c(0.0,0.1)), alt="two")

ARGUMENTS
x a list like that returned by Paired.t.test, Cor.test, Cox.test, Sp.test, Kd.test,
Two.t.test, Welch.test, Wil.s.test or Wil.test.
tol vector of tolerances.
alt If "two", lower and upper bounds are calculated. If "greater" or "less", only upper or
lower bounds are calculated, respectively.

VALUE
a list with following components:
tol vector of tolerances.
conf.bd matrix of confidence bounds. Each row is a vector consisting of lower and upper bounds for
each value of tol.
stat value of the statistic. The actual name is the same as that of the component of statistic in x.
parameters parameters of the distribution. The actual names are those of the components in x,
representing distribution parameters.
method the name of method applied.
alt alternative hypothesis.

EXAMPLES
x = Two.t.test(x,Y)
bd = Conf(x,tol=seq(0.0,1,lem=100))
Show(bd) # plot both confidence bounds for the difference
# of locations of X and Y

bd = Conf(Cor.test(x, y), tol=seq(0.0,1,lem=100))
matplot(bd$tol, conf(bd$conf,bd$t), type="l") # confidence bounds and
# point estimate of correlation of x and y are plotted.

Cor.test

T test for zero correlation

Cor.test(x, y, alt="two")

ARGUMENTS
x,y numeric vectors. x and y must have the same length. NAs are allowed but ignored at
calculation.
alt alternative hypothesis for zero correlation, "greater" (greater than 0), "less" (less than 0)
or "two" (nonzero).
5 Function Documentation

VALUE

a list representing the result of Student’s t test:

- rho: correlation between x and y.
- t: normalized rho.
- df: degree of freedom.
- prob: p-value (asymptotic).
- method: the name of method applied.
- alt: alternative hypothesis.

The statistic is given by

\[ t = \frac{\sqrt{n-2} \rho}{\sqrt{1-\rho^2}} \]

where \( n = \text{length}(x) = \text{length}(y) \).

EXAMPLES

\[
\text{Cox.test( rnorm(10), rnorm(10) )}
\]

\[
\text{Cox.test}(x, y, \text{alt="two")}
\]

ARGUMENTS

- x, y: survival time object like that returned by the function `survive`.
- alt: alternative hypothesis, "greater" (survival function for x dominates that for y), "less" (survival function for x is dominated by that for y) or "two" (survival functions for x and y differ).

VALUE

a list representing the result of Cox Mantel test:

- z: Cox Mantel statistic.
- sigma: standard deviation. z is normalized by this amount.
- prob: p-value (asymptotic).
- method: the name of method applied.
- alt: alternative hypothesis.

The statistic \( z = u / \sigma \) is calculated from

\[ u = d = \sum_{d=1}^m m[d] s[d] \]

and

\[ \sigma^2 = \sum_{d=1}^m \frac{m[d](s[d]-m[d])s[d](1-s[d])}{s[d]-1} \]

where \( d \) is the number of death cases in \( y \). Survival time objects \( x \) and \( y \) are classified into several classes by levels, \( \text{rev(sort(unique(c(x,y))))} \). The \( m[d] \) is the total number of death cases in the \( i \)-th class. The \( s[d] \) is the total number of cases with survival time being longer or equal to the \( i \)-th level. The \( a[d] \) is the ratio to the number \( s[d] \) of the number of cases in \( y \) with survival time being longer or equal to the \( i \)-th level.
NULL HYPOTHESIS
Two survival functions are equivalent.

EXAMPLES
Cox.test(Surv1,Surv2)

F.test(x, y)
ARGUMENTS
x, y numeric vectors. NAs are allowed but ignored at calculation.

VALUE
A list representing the result of F test with following components:
F value, that is, var(x)/var(y).
df degree of freedoms, length(x)-1 and length(y)-1.
prob p-value.
alt alternative hypothesis. Always "one".
method the name of method applied.

NULL HYPOTHESIS
Equal variances of x and y under normality assumption.

Fis.test(x, y, alt="two")
ARGUMENTS
x, y categories. A 2 by 2 contingency table may be given for x. Categories should be
dichotomous. The x and y must have the same length. Table values must be non-negative.
NAs are allowed but ignored at calculation.
alt alternative hypothesis with respect to the variable, the value of the [1, 1] cell, "less"
(negative dependency), "greater" (positive dependency) or "two" (any dependency).

VALUE
A list representing the result of Fisher exact test:
hyp frequency of the [1,1] cell.
x marginal totals with respect to the first column.
N marginal totals with respect to the second column.
x marginal totals with respect to the first row.
prob p-value. If x and y are given, the p-value is calculated for table(x,y).
method the name of method applied.
alt alternative hypothesis.

Under the following null hypothesis the statistic hyp is distributed as a hypergeometric
distribution with parameters n, n and k.

NULL HYPOTHESIS
x and y are conditionally independent when all marginal totals are given.
S Function Documentation

EXEMPLARY

Hypergeometric Distribution

\[
\begin{align*}
\text{dhyyp}(m, n, k, x_{\min}=0, x_{\max}=\min(m,k)) \\
\text{phyyp}(x, m, n, k) \\
\text{qhyyp}(p, m, n, k) \\
\text{rhyyp}(n, m, a, k)
\end{align*}
\]

ARGUMENTS

- \(n\) number of red balls.
- \(m\) number of black balls.
- \(k\) number of balls drawn without replication from an urn with \(m\) red and \(n\) black balls.
- \(x_{\min}\) minimum of \(x\), from where hypergeometric probabilities should be computed.
- \(x_{\max}\) maximum of \(x\), to where hypergeometric probabilities should be computed.
- \(x\) vector of values of random variable \(x\), the number of red balls drawn.
- \(p\) vector of probabilities. Missing values (NAs) are allowed.
- \(n\) sample size. If \text{length}(n) is larger than 1, then \text{length}(n) random numbers are returned.

VALUE

The function \text{dhyyp} returns a list with components:
- \(x\) vector of integers; \(\max(0, k-n, x_{\min})\leq\min(n, k, x_{\max})\).
- \(p\) vector of probabilities on \(x\).

Other functions return vector of cumulative probability(\text{phyyp}), quantile (\text{qhyyp}), or random sample(\text{rhyyp}) for the Hypergeometric distribution, which is explained by Urn Model with \(m\) red and \(n\) black balls. Any sequence of \(k\) drawings resulting in \(k-x\) black and \(x\) red balls has the same probability. The probability function is given by

\[
P_{m,n}(x) = \binom{m}{x} \binom{n}{k-x} / \binom{m+n}{k}
\]

SIDE EFFECTS

The function \text{rhyyp} causes creation of the dataset \_Random.seed if it does not exist in working directory, otherwise the value is updated.

EXEMPLARY

\[
\begin{align*}
dhyyp(4, 6, 7) \\
rhyyp(10, 4, 7, 6) \# sample of 10
\end{align*}
\]
**Kd.test**

Kendall's tau and nonparametric test for zero correlation

Kd.test(x, y, alt="two")

**ARGUMENTS**

x, y numeric vectors. The x and y must have the same length. Na's are allowed but ignored at calculation.

alt: alternative hypothesis for the correlation, "greater" (greater than 0) or "less" (less than 0) or "two" (nonzero).

**VALUE**

a list representing the result of t test for zero correlation:

- tau: Kendall's tau.
- z: normalized tau.
- sigma: standard deviation. z is normalized by this amount.
- prob: p-value (asymptotic).
- method: the name of method applied.
- alt: alternative hypothesis.

The statistic tau is given by

\[
\tau = \frac{\sum_{i<j} \text{sgn}(r_i - r_j) \text{sgn}(s_i - s_j)}{\sqrt{\frac{n(n-1)}{2}}}
\]

and z is given by \(z = \tau / \sigma\), where \(\sigma^2 = \frac{2(2n+5)}{(9n(n-1))}\). Where \(\text{sgn}(x) = 1, 0\) or -1 according to \(x > 0, =0\) or \(< 0\). The n is \(\text{length}(x) = \text{length}(y)\) and \(r_i\) is the rank of \(x[i]\) in \(x\) and \(s_i\) is the rank of \(y[i]\) in \(y\).

**NULL HYPOTHESIS**

x and y are independent.

**EXAMPLES**

Kd.test(x, y)

**KpMeier**

Kaplan-Meier estimate of survival function

KpMeier

KpMeier(x)

**ARGUMENTS**

x survival time object like that returned by the function `survival`.

**VALUE**

survival function object, that is, a list with following components:

- time: time vector.
- surv: vector giving the survival rate for each value of `time`.
- stderr: vector giving the standard error for each value of `time`.
- method: the name of method applied.

The survival function changes its value at time when a death case occurs. For later convenience, both previous and updated rates are in the component `surv.rate`. The
S Function Documentation

S FUNCTION: Kr.test

The Kruskal-Wallis rank test

ARGUMENTS

data data vector. A data matrix can be given. In this case, the second axis is regarded as
the corresponding the level, so that the following level should be omitted. NAs are allowed but
ignored at calculation.

level integer vector giving the level for each value of data. NAs are allowed but ignored at
calculation.

VALUE

a list representing the result of Kruskal-Wallis rank test:

chisq Kruskal-Wallis statistic.

df degree of freedom, the number of levels k - 1.

prob p-value (asymptotic).

method the name of method applied.

alt alternative hypothesis.

The test is a oneway analysis of variance for the case when normality assumption might be
violated.

The statistic \( \text{chisq} \) is currently defined as

\[
\text{chisq} = \frac{12}{n(n+1)} \sum_{i=1}^{k} \frac{r_i^2}{n_i} - 3(n+1)
\]

where \( r_i \) is the rank sum of data with level \( i \), the \( n_i \) is the number of such observations, and
the \( n \) is the total number of observations.

NULL HYPOTHESIS

No factor effects.

EXAMPLES

Kr.test(data, level)
Lifetable

**Estimation of survival function by lifetable method**

| Lifetable(x, breaks=6) |

**ARGUMENTS**
- `x` survival time object like that returned by the function `survive`.
- `breaks` either a vector of breakpoints or the number of equal-width intervals into which the values of `x` should be cut. The first interval always starts from 0.

**VALUE**
- `survival` function object, that is, a list with following components:
  - `time` time vector
  - `survive.rate` vector of survival rates at each intervals `time[i-1]` to `time[i]` for `i=1, 2, ...,` where `time[0]` is assumed to be 0.
  - `stderr` vector of standard errors at each intervals.
  - `method` the name of method applied

`survive.rate` is cumulative product of instantaneous survival rates on each time intervals. Instantaneous survival rate is defined by $1 - d/n$, where $d$ is the number of death cases and the $n$ is the effective number of cases; (the number of alive at the beginning of the time interval) - (the number of censored cases during the time interval)/2.

**NOTE**
If breakpoints are given for `breaks`, the right end of intervals should be given, since the values of `x` are cut into left open and right closed intervals. For example, if `x` takes only integers from 0 to 7 and which should be cut into intervals [0, 3] and [4, 7], then a simplest way is to give c(3,7) to `breaks`.

**EXAMPLES**

```
Lifetable(surv1)
```

**Logrank.test**

Logrank test for the equivalence of two survival time objects

| Logrank.test(x, y) |

**ARGUMENTS**
- `x`, `y` survival time objects like that returned by the function `survive`.

**VALUE**
- A list representing the result of Logrank test:
  - `chiq` Logrank statistic.
  - `df` degree of freedom 1.
  - `prob` p-value (asymptotic).
  - `method` the name of method applied.
  - `alt` alternative hypothesis.

The test is also called "Peto & Peto" test. The chi-square statistic is defined as
S Function Documentation

\[
\text{chisq} = \frac{(n_x - e_x)^2}{e_x} + \frac{(n_y - e_y)^2}{e_y}
\]

where \( n_x \) and \( n_y \) are the number of observed death cases respectively in \( x \) and \( y \) and \( e_x \) and \( e_y \) are expected number of death cases under the hypothesis.

NULL HYPOTHESIS
Two survival functions are equivalent.

EXAMPLES
Logrank.test(Surv1,Surv2)

S-Plus documentation for the Mantel-Haenszel test function.

ARGUMENTS
\( x,y \) dichotomous categories. A 2 by 2 by \( s \) table (array) may be given for \( x \). The \( s \) is the number of different strata.

\text{strata} vector giving the number which identifies stratum to which each value of \( x \) and \( y \) belong. This argument can be omitted when a 2 by 2 by \( s \) table is given for \( x \).

correct If \text{TRUE}, continuity correction is applied.

VALUE
A list representing the result of Mantel-Haenszel test:

\text{chisq} Mantel-Haenszel statistic.

\text{df} degrees of freedom 1.

\text{prob} p-value (asymptotic).

\text{method} the name of method applied.

\text{alt} alternative hypothesis. Always "one".

If two categories \( x \), \( y \) and \( \text{strata} \) are given, a 2 by 2 by \( s \) array is first created. The chisquare statistic chisq is calculated based on the array as

\[
\text{chisq} = \left( \frac{\sum n_{11s} - \sum n_{14s} n_{41s}}{n_{11s}} \right)^2 / \sum n_{14s} n_{24s} n_{21s} n_{42s}
\]

where \( n_{ab} \) is the \((i,j)\) element of the 2 by 2 by \( s \) array.

NULL HYPOTHESIS
\( x \) and \( y \) are conditionally independent when all marginal totals are given.

EXAMPLES
Mantel Haenszel.test(array) # 2 x 2 x s table (array)
Mantel Haenszel.test(F1,X,F2,Y, Strata)
Mc.test
McNemar test for the symmetry of contingency table

Mc.test(x, y, correct=T)

ARGUMENTS
x, y categories. A square table may be given for x. The x and y must have the same length and the same number m of levels. Table value must be non-negative. NA's are allowed but ignored at calculation.
correct If TRUE, continuity correction is applied if x and y are dichotomous or if a 2 by 2 table is given for x.

VALUE
a list representing the result of McNemar test:
chisq McNemar statistic. If both x and y are given, the statistic is calculated based on the table table(x,y). Otherwise, the statistic is calculated based on the given table x.
df degree of freedom m(m-1)/2.
pval p-value (asymptotic).
method the name of method applied.
alt alternative hypothesis. Always 'one'.

McNemar statistic is given by

\[
\text{chisq} = \sum_{i<j} \left(\frac{(|n_{ij}-n_{ji}| - \text{correct})^2}{n_{ij}+n_{ji}}\right)
\]

where \(n_{ij}\) is the \([i,j]\) element of table(x,y).

NULL HYPOTHESIS
Matrix of cell probabilities is symmetric. This means that variables x and y are exchangeable. If the table is 2 by 2 and one of variables x or y can be considered as a response variable, then the test is regarded as a significance test of effects of explanatory variable in logit model.

EXAMPLES
Mc.test(Mc.X, Mc.Y)

<table>
<thead>
<tr>
<th>Msequence</th>
<th>M-Sequence Random Numbers</th>
<th>Msequence</th>
</tr>
</thead>
</table>
mrbeta(n, shape1=0, shape2=1, refresh=F) | mseqchisq(n, location=0, refresh=F) | mrf(n, df1, df2, refresh=F) |
mrchisq(n, df, refresh=F) | mrgamma(n, shape, refresh=F) | mrgnorm(n, mean=0, scale=1, refresh=F) |
mrexparet(n, refresh=F) | mrlnorm(n, meanlog=0, sdlog=1, refresh=F) | mxlogis(n, location=0, scale=1, refresh=F) |
mrnorm(n, mean=0, sd=1, refresh=F) | mxr(n, df, refresh=F) | mrunif(n, min = 0, max = 1, int = F, refresh = F) |
S Function Documentation

ARGUMENTS

- **n**: sample size. If `length(a)` is larger than 1, then `length(a)` random numbers are returned.
- **ist**: If TRUE, `mrand` returns vector of integer random numbers between 0 and 2**32-1. Else returns uniform random numbers between `min` and `max`.
- **refresh**: If TRUE, the seed `.MRDseed` for M-sequence generation on working directory is updated and a new M-sequence generation is started from that seed.

See documentation for the corresponding standard random number generation function for other arguments.

VALUE

vector of random numbers.

Each M-sequence can be identified by two values of `.MRDseed`. The first value is the initial value for an internal congruence routine which derives a table for M-sequence generation. The last value is the number of M-sequence random numbers generated after creation of the new table. Successive random numbers are equivalent to a long sample from the underlying uniform distribution unless `refresh = FALSE`. It allows the long-term properties of the generator to be maintained. If `refresh = TRUE`, the first value of `.MRDseed` is updated by invoking once the congruence routine and the last value is set as 0. There is a useful technique for reproducing random sample in later work. Just copy `.MRDseed` before generating the sample for the first time, and then restore it when the sample is to be reproduced. To save computation time, it is recommended to refresh `.MRDseed` before copying `.MRDseed` for later use.

SIDE EFFECT

The functions cause creation of the dataset `.MRDseed` if it does not exist on the working directory, otherwise the value is updated.

NOTE

This function will load the C object code "mrand.o" through `library.dynam("mrand", "mrand.o")`.

SOURCE

The original C code was written by K. Kano and M. Fushimi, Tokyo University.

REFERENCE


---

**Normal.test**

**Test for Normality**

**Normal.test**

ARGUMENTS

- **x**: numeric vector. Missing value (NA) are allowed but ignored at calculation.

VALUE

- a list representing the result of chisquare type goodness of fit of normality test with following components:
  - `chisq`: chi-square statistic.
  - `df`: degree of freedom.
Normal.test

prob p-value (asymptotic).
method the name of method applied.
alt alternative hypothesis.
dimnames a list representing dimnames of the contingency table. dimnames[[2]] shows the definition of classes.

Observed frequencies are calculated from x and expected frequencies are from normal distribution with the mean mean(x) and the standard deviation sqrt(var(x)), and Chi^2 test called with the option difference=T. The number k of the classes is chosen as ceiling( log(length(x), base=2) + 1) and each class is chosen so as to almost have the same probability under the normal distribution.

NULL HYPOTHESIS

x is normally distributed.

**One way cov.test**

One way analysis of variance with covariate (ANOCOVA) One way cov.test

One way cov.test(data, level, block, cov, tol=0.05)

ARGUMENTS

data data vector. NAs are allowed but ignored at calculation.
cov vector giving the value of covariate for each value of data. NAs are allowed but ignored at calculation.
level vector giving the level for each value of data. NAs are allowed but ignored at calculation.
block vector giving the block number for each value of data. This argument may be given only when the design is a blocked design. NAs are allowed but ignored at calculation.
tol tolerance for the significance of covariate.

VALUE

t a list representing the result of ANOCOVA:
SS vector giving decomposed values of total sum of squares.
cov the result of t-test for the significance of covariate. This component further has following subcomponents:
df degree of freedom.
t t statistic.
prob p-value.
alt alternative hypothesis. Always "two", that is, two sided.
between the result of F-test for the significance of difference of levels, that is, the significance of "between variance". This component further has following subcomponents:
df degree of freedom.
F F statistic.
prob p-value.
alt alternative hypothesis. Always "one".
block the result of F-test for the significance of block effect. This component exists only when the argument block is given. This component further has following subcomponents:
df degree of freedom.
F F statistic.
prob p-value.
alt alternative hypothesis. Always "one".
method the name of method applied
S Function Documentation

EXAMPLES

Oneway.cov.test(Data, Level, Block, Cov)

Oneway.test

Oneway.test(data, level, block)

ARGUMENTS

data data vector. A data matrix can be given by omitting other arguments. In this case, the first axis stands for repetition and the second for levels. #s are allowed but ignored at calculation.

level vector giving the level of controllable factor corresponding to each value of data. #s are allowed but ignored at calculation.

block vector giving the block number to which each value of data belongs. This argument may be given only for the case where the design is a blocked design. #s are allowed but ignored at calculation.

VALUE

a list representing the result of ANOVA:

$SS$ vector giving decomposed values of total sum of squares.

between the result of F-test for the significance of difference of levels, that is, significance of "between variance". This component further has following subcomponents:

df degree of freedom.

f F statistic.

prob p-value.

alt alternative hypothesis. Always "one", that is, one sided.

block result of F-test for the significance of block effects. This component exists only when the argument block is given. This component further has following subcomponents:

df degree of freedom.

f F statistic.

prob p-value.

alt alternative hypothesis. Always "two", that is, two sided.

method the name of method applied

EXAMPLES

Oneway.test(Data, Level, Block)

Q.test

Q.test(data, level, block)

ARGUMENTS

data data vector. #s are allowed but ignored at calculation.

level vector giving the level of each value of data. #s are allowed but ignored at calculation.

block vector giving the block number to which each value of data belongs. The design should be a completely randomized block design, that is, the number of observations in data is the same for every combination of level and block. #s are allowed but ignored at calculation.
VALUE

a list representing the result of generalized Cochran’s Q test:

- chisq: generalized Cochran’s Q statistic.
- df: degree of freedom, the number of levels k - 1.
- prob: p-value (asymptotic).
- method: the name of method applied.
- alt: alternative hypothesis.

The test is equivalent to the Friedman test if no ties exist in any block. If data is dichotomous category, then the test is equivalent to Cochran’s Q.

The chisquare statistic $\chi^2$ is defined as

$$\chi^2 = \frac{12}{bk(k+1)} \sum q_i^2 - 34(k+1)$$

$$1 - \frac{\sum (d_j - d_i)^2}{bk(k^2 - 1)}$$

where $d_i$ is the number of ties with the i-th smallest value in the block j, $r_i$ is the rank sum of data with the i-th level. The b is the number of blocks.

NULL HYPOTHESIS

No factor effects in any block under normality assumption.

EXAMPLES

```r
g.test(category(Data), Level, Block)
```

**ARGUMENTS**

- name: character string giving the name of the distribution. Currently "t", "norm", "f" or "chisq" can be specified.
- s: vector of values of the statistic distributed as the name distribution under the null hypothesis. Missing values (NA) are allowed but ignored.
- ...: parameters to the distribution name if necessary. These are passed through `name` or `pars` functions.
- main: character string for the main title of graphical representation.
- sub: character string for the sub title of graphical representation. If missing, significance level and the value of distribution parameters are drawn.
- xlab: x axis label
- ylab: y axis label. If missing the name of the distribution is used.
- level: significance level between 0 and 1.
- alt: alternative hypothesis. If "two", two sided rejection region is shaded. If "greater" or "less", right or left sided rejection region is respectively shaded.
- exact: If TRUE, exact proportion given by level of the barplot of the (discrete) distribution will be shaded.

**SIDE EFFECTS**

Density function is drawn on a plot with shaded rejection region. The value of statistic is
S Function Documentation

plotted by "=" below the curve. If length(a) > 1, values of the statistic are plotted with sequence numbers. Values outside of the range are plotted with "o". Rej is for continuous distribution and Rej.d is for discrete distribution.

Graphical parameters may also be supplied as arguments to this function (see par).

NOTE
This function is usually invoked by the function Show.

EXAMPLES
Rej("t",1.2,df=4)  # Draw rejection region of t test with degree of freedom 4.
Rej("F",1.6,df1=4,df2=8)
Rej("norm",1.0,alt="two")

Rej.d
See Rej

Rep.t.test
Repeate t tests based on sample mean and variance

Rep.t.test

Rep.t.test(mean, n, var, df, alt)

ARGUMENTS
mean vector of sample means. Has are allowed but ignored at calculation.
n vector giving the sample size for each value of mean. If the length(n) is 1, then all sizes are assumed to be the same.
var common variance.
df degree of freedom. Default is sum(n) - length(mean).
alt square matrix with the same number of columns or rows as the length of mean. The lower triangle part of this matrix specifies alternative hypothesis for every combination of mean: 1(greater sided), 0(two sided), -1(less sided) or NA(no testing). Default is a 0 matrix.

VALUE
a list representing the result of repeated t test with the common variance var:
prob matrix. The [i,j] element of the lower triangular of this matrix gives p-value of the t test for the pair mean[i] and mean[j] as specified by the given for argument alt. Any other elements are NAs.
t matrix. The lower triangular element gives the value of t statistic for each test. Any other elements are NAs.
alt the same matrix as the argument alt.
df common degree of freedom.
signs common standard deviation. The component t is normalized by this amount.
method the name of method applied
alt alternative hypothesis

See rep.t.test.

EXAMPLES
Rep.t.test(1:3, 5, 4)
Show rejection region, confidence bounds or survival function

Show(x, level=0.05, main, alt, tol, exact=T, child=F)

ARGUMENTS
- `x`: a list like that returned by one of various testing functions including `rep.t.test`, survival analysis functions `survive.fun`, `SpMeier` and `Lifetable` and the function `Conf`. Atomic components `t.s, chisq or f` are recognized as values of such test statistics. Components `cov, between` and `block` are recognized as results of `Oneway.cov.test` (ANOVA) or `Oneway.cov.test` (ANOVA).
- `level`: significance level, the size of rejection region.
- `main`: character string for the main title. This is valid only for the case of single plot. If the argument is omitted or multiple plot is required, an appropriate title is generated from the component method of `x`.
- `alt`: alternative hypothesis. If "two", rejection region is two sided, and right sided or left sided according to "greater" or "less". This is valid only for the case of single plot. For components `chisq or f, alt` is always "greater", otherwise the value of components `alt` of `x` is used.
- `tol`: if a vector of tolerances is given, confidence bounds will be calculated and plotted from the result of tests.
- `exact`: If `TRUE`, exact proportion of the barplot is shaded according to `level`.
- `child`: If `FALSE` multiple plot is turned off. This flag is used for recursive call of `Show`.

SIDE EFFECTS
- If the component `time` is found in `x`, survival function will be drawn. If the component `confbd` is found, or the argument `tol` is given and `x` the result of a testing, then confidence bound will be drawn. If one of atomic components `t.s, chisq or f` is found in `x`, corresponding rejection region is shaded. Otherwise, components `cov, between or block` will be searched. If found, corresponding rejection regions are plotted.

Graphical parameters may also be supplied as arguments to this function (see par).

EXAMPLES

```
Show(survive.fun(Surv1 )) # survival function
Show(Paired.t.test(3,7, tol=pretty(c(0,0.1))) ) # confidence bounds
Show(ttt--Paired.t.test(x,y)) # save numerical result and have a
# graphical representation

tt # check the numerical result

ex--Oneway.cov.test(Data,Level,Block,Cov) # oneway covariance test
Show(ex) # graphical representation
```

### Sp.test

Spearman's rank correlation and nonparametric test for zero correlation

```
Sp.test(x, y, alt="two")
```

ARGUMENTS
- `x, y`: vectors. The `x` and `y` must have the same length `n`. `NA`s are allowed but ignored at calculation.
S Function Documentation

null alternative hypothesis for the correlation, "greater" (greater than 0) or "less" (less than 0) or "two" (nonzero).

VALUE

a list representing the result of Spearman's rank correlation test:

tau correlation between x and y.

n normalized rho.

m number of observation.

p-value (asymptotic).

method the name of method applied.

null alternative hypothesis.

The Spearman's rank correlation rho is defined as

\[ \rho = 1 - \frac{6}{n^2 - n} \sum (r_i - s_i)^2 \]

where \( r_i \) is the rank of \( x[i] \) in \( x \) and \( s_i \) is the rank of \( y[i] \) in \( y \). The normalized rho is

\[ n = \sqrt{(n-1)} \left[ \rho = \frac{6}{n^3 - n} \right] \]

NULL HYPOTHESIS

\( x \) and \( y \) are independent.

EXAMPLES

Sp.test(x, y)

StNw.test Stuart-Maxwell test of the marginal homogeneity of a contingency table

StNw.test

ARGUMENTS

\( x, y \) categories. A square contingency table may be given for \( x \). The length and the level of \( x \) and \( y \) should be the same. Table values must be non-negative. Missing are allowed but ignored at calculation.

\( w \) a weight matrix for the generalized Stuart-Maxwell statistics. Default is diag(1, m-1, m) where \( m \) is the number of levels.

VALUE

a list representing the result of Stuart Maxwell test:

\( \chi^2 \) Stuart Maxwell statistic. If \( x \) and \( y \) are given, the statistic is calculated for the table \( x, y \).

\( \nu \) degree of freedom, \( m-1 \).

p-value (asymptotic).

method the name of method applied.

null alternative hypothesis. Always "one".

The test is for the homogeneity of row and column marginal distributions of the contingency table.
The chi-square statistic \( \text{chisq} \) is defined as

\[
\text{chisq} = (Ad)(AVA')^{-1}(Ad)
\]

where \( V = (v_{ij}) \) is the \( m \) by \( m \) matrix with \( v_{ii} = n_i + n_j - 2n_{ij} \) and \( v_{ij} = -(n_i + n_j) \) for \( i \neq j \), and the \( d \) is the vector with \( d_i = n_i - n \). The weight matrix \( A \) is a \( m \times 1 \) by \( m \) matrix of full rank.

**NULL HYPOTHESIS**
Marginal distributions of \( x \) and \( y \) are the same.

**REFERENCE**

**EXAMPLES**
StNW.test(Vision)

<table>
<thead>
<tr>
<th>T.test</th>
<th>T test for location</th>
<th>T.test</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.test(x, y, alt=&quot;two&quot;)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ARGUMENTS**

- \( x, y \) numeric vectors. The \( x \) and \( y \) must have the same length. If \( y \) is missing, one sample t test is applied. \( \mathbb{N} \)s are allowed but ignored at calculation.
- \( \text{alt} \): alternative hypothesis, "greater" (location of \( x \) is greater than that of \( y \)) or "less" (location of \( x \) is less than \( y \)) or "two" (locations of \( x \) and \( y \) are different).

**VALUE**

- a list representing the result of paired t test:
  - \( t \) paired t statistic.
  - \( \text{sigma} \) standard deviation. \( t \) is normalized by this amount.
  - \( d \) degree of freedom.
  - \( \text{prob} \) p-value.
  - \( \text{method} \) the name of method applied.
  - \( \text{alt} \) alternative hypothesis.

Paired t statistic is given by

\[
t = \frac{\sqrt{n} \bar{x}}{\text{sigma}}
\]

where \( \bar{x} \) and \( \text{sigma} \) are mean and standard deviation of the difference \( x-y \).

**NULL HYPOTHESIS**
Locations of \( x \) and \( y \) are equal under normality assumption.

**EXAMPLES**

T.test(x, y)
**S Function Documentation**

<table>
<thead>
<tr>
<th>Two.t.test</th>
<th>Two sample t test</th>
<th>Two.t.test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two.t.test(x, y, alt=&quot;two&quot;)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ARGUMENTS**

- `x, y` numeric vectors. NAs are allowed but ignored at calculation.
- `alt` alternative hypothesis, "greater" (location of `x` is greater than that of `y`) or "less" (location of `x` is less than that of `y`) or "two" (two locations of `x` and `y` are different).

**VALUE**

- a list representing with the following components:
  - `t`: two sample t test statistic.
  - `df`: degree of freedom.
  - `prob`: p-value.
  - `method`: the name of method applied.
  - `alt`: alternative hypothesis.

Unpaired two sample t statistic is given by

\[
t = \frac{\bar{x} - \bar{y}}{\text{sigmas}}
\]

where

\[
\text{sigmas}^2 = \left(\frac{1}{n_x} + \frac{1}{n_y}\right) \left(\frac{(n_x-1)s_x^2 + (n_y-1)s_y^2}{n_x + n_y - 2}\right)
\]

and \(s_x^2\) and \(s_y^2\) are unbiased estimate of variances of `x` and `y`, respectively.

**NULL HYPOThESIS**

There is no difference between locations of `x` and `y`.

See `T.test`

**EXAMPLES**

Two.t.test(x, y)

<table>
<thead>
<tr>
<th>Welch.test</th>
<th>Aspin-Welch test</th>
<th>Welch.test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welch.test(x, y, alt=&quot;two&quot;)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ARGUMENTS**

- `x, y` numeric vectors. NAs are allowed but ignored at calculation.
- `alt` alternative hypothesis, "greater" (location of `x` is greater than that of `y`) or "less" (location of `x` is less than that of `y`) or "two" (two locations of `x` and `y` are different).

**VALUE**

- a list representing the result of Aspin-Welch test for the difference of locations:
22 Welch.test

S Function Documentation

- Aspin-Welch statistic.
- `sigma` standard deviation. The `t` is normalized by this amount.
- `df` (continuous) approximate degree of freedom.
- `prob` p-value (approximate).
- `method` the name of method applied.
- `alt` alternative hypothesis.

This test is a `t` test of the equality of location of two samples for the case when equal variance may be violated.

The `t` statistic is defined as

\[ t = \frac{\overline{x} - \overline{y}}{\sigma} \]

where

\[ \sigma^2 = \frac{s_x^2}{n_x} + \frac{s_y^2}{n_y} \]

The approximate degree of freedom `df` is

\[ df = \frac{1}{\frac{\sigma^2}{n_x-1} + \frac{(1-c)^2}{n_y-1}} \]

where

\[ c = \frac{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}{n_x + n_y} \]

**NULL HYPOTHESIS**

There is no difference between locations of `x` and `y` under normality assumption.

**EXAMPLES**

`Welch.test(x, y)`

`Wil.s.test(x, y, alt="two", correct=T)`

**ARGUMENTS**

- `x`, `y` numeric vectors. The `x` and `y` must have the same length. `NA`'s are allowed but ignored at calculation.
- `alt` alternative hypothesis, "greater" (location of `x` is greater than that of `y`) or "less" (location of `x` is less than that of `y`) or "two" (two locations of `x` and `y` are different).
- `correct` If `TRUE`, continuity correction is applied.

**VALUE**

- a list representing the result of Wilcoxon signed rank sum test:
  - `z` normalized signed rank sum.
  - `sigma` standard deviation. `z` is normalized by this amount.
  - `prob` p-value (asymptotic).
S Function Documentation

method the name of method applied.
alt alternative hypothesis. The Wilcoxon signed test statistic is currently defined as

\[ z = \frac{r - \frac{n(n+1)}{4}}{\sigma} \]

where \( \sigma^2 = \frac{n(n+1)(2n+1)}{24} \) and \( r \) is the rank sum of abs(x-y) with x-y>0.

NULL HYPOTHESIS
The median of x-y is 0.

EXAMPLES
Wil.test(x, y)

<table>
<thead>
<tr>
<th>Wil.test</th>
<th>Wil.test(x, y, alt='two', exact=T)</th>
</tr>
</thead>
</table>

ARGUMENTS
x,y numeric vectors, or survival time objects like that returned by the function survive.
alt alternative hypothesis, "greater" (location of x is greater than that of y, survival function for x dominates that for y) or "less" (location of x is less than that of y, survival function for x is dominated by that for y) or "two" (two locations are different, two survival functions for x and y are different).
exact If exact is TRUE and x and y are numeric vectors, the exact p-value will be calculated as far as both length(x) and length(y) are less than 50 and no ties exist in ranks of x and y. Otherwise normal approximations is applied to obtain the p-value.

If numeric vectors are given, Wilcoxon U statistic is calculated. Otherwise, generalized Wilcoxon test statistic for the equivalence of two survival time data is calculated. It is also known as "Gehan's Generalized test statistic".

VALUE
z rank sum test statistic based on x and y. When exact p-value is calculated, the name of this component is wil and the value is a simple rank sum of x.
sigma standard deviation. z is normalized by this amount.
prob p-value.
method the name of method applied.
alt alternative hypothesis.

The Wilcoxon U statistic is defined as

\[ z = \frac{U - \frac{n_x(n_x+n_y+1)}{2} + \text{correction}}{\sigma} \]

where sigma is given by

\[ \sigma^2 = \frac{n_x n_y (n_x+n_y+1)}{12} \]

and U is the sum of rank(c(x,y)) along with longer vector x or y.
Generalized Wilcoxon test statistic is calculated as

\[ z = \frac{\sum u_j}{\sigma} \]

where \( \sigma^2 \) is

\[ \sigma^2 = \frac{n_x n_y}{(n_x + n_y)(n_x + n_y - 1)} \sum_j (\sum y_j)^3 \]

and \( u_j \) is -1, 1 or 0 according to
1. \( x(i) < y(j) \) or \( x(i) = y(j) \) and \( y(j) \) has status 2 or 3, i.e. censored.
2. \( x(i) > y(j) \) or \( x(i) = y(j) \) and \( x(i) \) has status 2 or 3, i.e. censored.
3. otherwise.

The \( u_j \) is defined as similarly as \( u_i \) by putting \( x \rightarrow c(x, y) \).

**NULL HYPOTHESIS**

Two locations are equal (U test) or Survival functions for \( x \) and \( y \) are equivalent (Generalized Wilcoxon).

**EXAMPLES**

`wil.test(Survi, Surv2)`

<table>
<thead>
<tr>
<th>Wilcoxon</th>
<th>Distribution of Wilcoxon rank sum statistic</th>
<th>Wilcoxon</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dwil(m, x, xmin, xmax, freq=F)</code></td>
<td><code>pwil(x, m, n)</code></td>
<td><code>qwil(prob, m, x)</code></td>
</tr>
<tr>
<td><code>rwil(xm, m, n)</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ARGUMENTS**

- \( m \): number of observations from treatment group.
- \( n \): number of observations from contrast group.
- \( x \): vector of cumulative probability on each value of \( x \) will be calculated.
- \( xmin \): minimum of \( x \), from where the probabilities should be computed.
- \( xmax \): maximum of \( x \), to where the probabilities should be computed.
- \( freq \): if \( freq \) is \( TRUE \), the theoretical frequencies are returned, instead of probabilities.
- \( prob \): vector of probabilities. Missing values(\( NA \)) are allowed.
- \( n \): sample size. If \( length(n) \) is greater than 1, then \( length(n) \) random numbers are returned.

**VALUE**

The function `dwil` returns a list with the following components:
- \( x \): vector of integers, \( max(m*(m+1)/2, xmin) : min( m*(2*m*(2*m+1)/2, xmax) \).
- \( prob \): vector of probabilities on \( x \).

Other functions return vector of cumulative probability(`pwil`), quantile(`qwil`), or random sample(`rwil`) for the rank sum distribution.

**NOTE**

This function will load the C object "mw.o" through `library.dynam("mwi", "mw.o")`.

**SOURCE**

The original C code is written by A.R. Wilks at Bell Labs.
S Function Documentation

EXAMPLES

dwil(4,6)  # all probabilities of Wilcoxon rank sum statistics with
# n=4 and m=6

pwil(24,4,6)  # the probability of x<24

Z.test  Z test for the equivalence of pair of survival time objects

Z.test(x, y, alt="two", breaks=6)

ARGUMENTS

x,y survival time objects, like that returned by the function surv
alt alternative hypothesis. "greater" (survival function for x dominates that for y) or "less" (survival function for x is dominated by that for y) or "two" (two survival functions for x and y differ).
breaks either a vector of breakpoints or the number of equal-width intervals into which x and y should be cut.

VALUE

This function calls Lifetable to obtain survival functions for x and y, and Z test statistic is calculated based on those two functions. A list representing the result of Z test:
time time vector.
x vector giving Z statistic for each value of time.
prob vector giving p-value for each value of time.
method the name of method applied.
alt alternative hypothesis.

The Z statistic is defined as

\[ z[i] = \frac{S_x[i] - S_y[i]}{(s_x[i]^2 + s_y[i]^2)^{1/2}} \]

where \( S_x \) and \( S_y \) are survival rates and \( s_x \) and \( s_y \) are standard errors of \( S_x \) and \( S_y \), with respect to x and y.

NULL HYPOTHESIS

Two survival functions are equivalent.

EXAMPLES

Z.test(Surv1,Surv2)
autocov

Autocovariances

autocov(..., maxlag=6)

ARGUMENTS
any number of arguments. Each argument is interpreted as a time-series except when a time
series matrix like that returned by tsmatrix is given. In this case, each column is interpreted
as a time-series.

maxlag maximum lag of autocovariances

VALUE
a list with following components:

autocov a \text{maxlag+1} \times \text{maxlag+1} array, where \text{k} is the number of arguments given for \ldots. If only a
time series given, the value is a simple vector.
tap time series parameter, which is intersections of time windows for the arguments \ldots.

The \([ij]\) elements of the component \text{autocov} give a sequence of autocovariances,

\[ \frac{1}{n} \sum_{t=1}^{n} x_i(t) x_j(t+h), \quad h=0, \ldots, \text{maxlag}, \]

where \(x_i(t)\) is the \(i\)-th time series given for \ldots and the \(n\) is the length of intersection of given
time series.

EXAMPLES

autocov(spin, hstart)


cnorm

Coverage probability of normal distribution

cnorm

cnorm(x, mean=0, sd=1, sides=T)

ARGUMENTS
\(x\) vector of break points. Real line is divided into length(x)+1 intervals, where the first
interval covers from minus infinity to x[1] and the last interval covers from x[length(x)]
to infinity.

mean mean of normal distribution

sd standard deviation of normal distribution

sides If FALSE, coverage probabilities of both sides are not returned.

VALUE
vector of coverage probabilities calculated from the normal distribution with mean \text{mean} and
standard deviation \text{sd}, of intervals specified by the break points \(x\) and the option \text{sides}. 
S Function Documentation

\[ \text{diff.test}(x, y, \text{alt}="two", \text{correct}=\text{TRUE}) \]

**ARGUMENTS**
- \( x \): vector or category
- \( y \): vector or category
- \( \text{alt} \): alternative hypothesis. Choices are "two" (two sided), "greater" (x is greater sided than y) or "less" (x is less sided than y). This argument is not used when \( x \) and \( y \) are categories.
- \( \text{correct} \): If TRUE, continuity correction is applied as far as possible.

**VALUE**
A list representing the result of one of tests for the difference of two samples described in the following tree. At each stage, graphical or numeric assistance is available for user to choose one of options. If \text{Chisq.test} is selected, vector of frequencies of \( x \) and \( y \) are combined into a 2 by (the number of levels) contingency table and passed through \text{Chisq.test}.

**EXAMPLES**
\[ \text{diff.test}(x, y) \]

**FUNCTIONS CALLED**

\begin{align*}
\text{diff.test} & \\
\text{paired} & = \text{F} & \text{T} \\
\text{category} & & \text{vector} \\
\text{Chisq.test} & & \text{Mc.test} \\
\text{Normal} & = \text{F} & \text{T} & \text{Normal} & = \text{F} & \text{T} \\
\text{Wils.test} & & \\
\text{T.test} & & \\
\text{eq.scale} & = \text{F} & \text{T} 
\end{align*}
28 diff.test

| Wil. test | Welch.test | Two.t.test |


```
dwil
See: wilcox.test
dwil
```

```
indep.test Interactive test for the independence of two samples indep.test
```

```
indep.test(x, y, correct=TRUE, alt="two")
```

ARGUMENTS

- `x` vector or category
- `y` vector or category
- `correct` If TRUE, continuity correction is applied.
- `alt` alternative hypothesis. Choices are "two", "greater" or "less". This is not used when Chisq.test is selected.

VALUE

A list representing the result of test for the independence of two samples, Cor.test, Chisq.test, Ed.test or Sp.test.

If `x` and `y` are categories, always Chisq.test is called. Otherwise the user will be asked if normality can be assumed. At this stage, a graphical or numeric assistance is available. If normality can be assumed, Cor.test is called, otherwise the user has to choose either Ed.test or Sp.test.

EXAMPLES

```
indep.test(1,1)
```

FUNCTIONS CALLED

```
<table>
<thead>
<tr>
<th>indep.test</th>
</tr>
</thead>
<tbody>
<tr>
<td>different length</td>
</tr>
</tbody>
</table>
```

ERROR

```
<table>
<thead>
<tr>
<th>category</th>
<th>vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal=F</td>
<td>T</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>Chisq.test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kd.test</td>
</tr>
<tr>
<td>Sp.test</td>
</tr>
<tr>
<td>Cor.test</td>
</tr>
</tbody>
</table>
```
life.test Interactive test for the equivalence of two survival time objects

life.test(x, y, alt="two", breaks=5)

ARGUMENTS
- x: survival time object, like that returned by the function `survive`.
- y: survival time object, like that returned by the function `survive`.
- breaks: either a vector of breakpoints or the number of equal-width intervals into which the value of x and y should be cut.
- alt: alternative hypothesis, which is one of "two" (survival functions differ), "greater" (survival function for x dominates that for y) or "less" (survival function for x is dominated by that for y).

VALUE
- a list representing the result of test for the equality of two survival functions, `Cox.test`, `Logrank.test`, `Wil.test` or `Z.test`.

If `breaks` is given, always `Z.test` is invoked. Otherwise, the user may choose one from `Cox.test`, `Logrank.test`, `Wil.test` or `Z.test`.

EXAMPLES

```r
life.test(Surv1, Surv2)
```
### mrgamma

See `msequence`  

### mrlnorm

See `msequence`  

### mrlogis

See `msequence`  

### mrmnorm

See `msequence`  

### mrt

See `msequence`  

### mrunif

See `msequence`  

### nesi

Tree based menu for NESI functions  

nesi(..., tree=test)

**ARGUMENTS**

... S objects to be analyzed  

tree a tree which derives the function `mrmn`.

**EXAMPLES**

nesi(x,y)  # analyze x and y

![](oneway.png)

**ARGUMENTS**

- `data` vector or category. NAs are allowed but ignored at calculation.  
- `level` vector giving the level of each value of `data`. NAs are allowed but ignored at calculation.  
- `block` vector giving the block number to which each value of `data` belongs. This argument is required only when the design is a blocked design. NAs are allowed but ignored at calculation.  
- `cov` vector giving the value of covariate for each value of `data`. NAs are allowed but ignored at calculation. This argument is required only for the case when `oneway.cov.test` is chosen.  
- `correct` if `TRUE`, continuity correction is applied. This option is valid only for dichotomous categories. This argument is required only for the case when `Chisq.test` is chosen.  
- `tol` tolerance for the significance of covariate `cov`. This argument is required only for the case when `oneway.cov.test` is chosen.

**VALUE**

a list representing the result of oneway variance analysis by one of methods, `Chisq.test`,
S Function Documentation

**Kr.test, Q.test, Oneway.test or Oneway.cov.test.**

**EXAMPLES**

`oneway.test(Data, Level, Block)`

**FUNCTIONS CALLED**

```
oneway.test
  cov:given  |  cov:omitted
    Oneway.cov.test
      block:given  |  block:omitted
        category  |  vector
          normal=F  |  T
```

```
Q.test  Oneway.test  Chisq.test  Kr.test  Oneway.test
```

**pf**

*Probability for Central or Noncentral F-distribution*

```
pf(q, df1, df2, delta=0)
```

**ARGUMENTS**

- `q` vector of (positive) quantiles.
- `df1` vector of degrees of freedom for numerator.
- `df2` vector of degrees of freedom for denominator.
- `delta` vector of noncentrality parameters, that is, the sum of squares of mean of normal random variables when the denominator of the definition of F-statistic is decomposed into the sum of squares of such normal variables.

**VALUE**

Lower probability for central or noncentral F-distribution with degree of freedoms `df1` and `df2`, and noncentrality `delta`.

**NOTE**

To use this function, the library should be attached at position 2 to mask the system function `pf`, for example, by `library(nesi, first=T)`. This function will load the FORTRAN object "pf.o" through `library.dynam("nesi", "pf.o")".
SOURCE
The original FORTRAN code is written by J. Takeuchi at Keio Univ.

EXAMPLES
pf(1,6,5,6,1) # Cumulative probability of noncentral F-distribution at 1
# With degree of freedom 5 and 6 and noncentrality 0.1.
pf(0.4,2,3) # Cumulative probability of F-distribution at 0.4 with
# Degree of freedom 2 and 3.

phyp
See Hypergeometric

pwil
See Wilcoxon

qbn
See Binomial

qhyp
See Hypergeometric

qwil
See Wilcoxon

rbn
See Binomial

rep.t.test
Repeated t tests

data, level, alt, method

ARGUMENTS

data data vector. NA s are allowed but ignored at calculation.
level vector giving the level to which each value of data belongs.
alts square matrix which has the same number of columns and rows as the number of levels. The
lower triangle part of this matrix specifies alternative hypothesis for each combination of
levels, 1 (greater sided), 0 (two sided), -1 (less sided) or NA (no testing). Default is 0 matrix.
method choice of methods to evaluate p-values of repeated t tests. Currently three methods are
implemented, Students, Bonferoni, or Dunn.

VALUE

A list with following components:
prob matrix. The [i,j] element of lower triangular of this matrix gives the transformed p-value of t
test for the pair mean[i] and mean[j] as specified by the argument alt. Any other
elements are all NAs. The transformations are, no transformation, k*(p-value), 1 - (1 - (p-
value))**k respectively for the method, Student, Bonferoni and Dunn, where k is the number
of combinations of t tests.
S Function Documentation

survive 33

t matrix. The lower triangular element gives the value of t statistic for each t test. Any other
elements are all #s.
altn the same as the matrix given for altn.
df common degree of freedom.
sigma common standard deviation.
method the name of method applied.
altn alternative hypothesis.

If method is omitted, the user will be asked to choose one of methods St, Be or Da.

EXAMPLES

t.test(Data, Level)

<table>
<thead>
<tr>
<th>rhyp</th>
<th>See Hypergeometric</th>
</tr>
</thead>
<tbody>
<tr>
<td>rwil</td>
<td>See Wilcoxon</td>
</tr>
<tr>
<td>survive</td>
<td>Survival Time Objects</td>
</tr>
</tbody>
</table>

\[
survive(data, status=1, start=0, end=0) \\
is.survive(x) \\
as.survive(x, status=1)
\]

ARGUMENTS

data vector of survival time. If omitted, this is calculated by data = end - start.
status vector giving the status of each case, 1 is for "death", 2 is for "censored\(\text{alive}\)", and 3 is for "censored\(\text{dropout}\)".
start vector giving the time at which observation starts.
end vector giving the time at which observation ends.
x any S object.

VALUE

survive returns a survival time object with the same mode as data and with an attribute status.
is.survive returns TRUE if x is a survival time object, FALSE otherwise.
as.survive returns x if x is a survival time object, otherwise returns a survival time object
with an attribute status given by the argument status.

EXAMPLES

\[
survive(c(3,4,6,7,3),c(1,3,2,1,2)) \quad \# \text{ creates a survival time object}
is.survive( Surv1 ) \quad \# \text{ check if survival time object or not}
\]
survive.fun

Estimation of survival function

survive.fun(x, breaks)

ARGUMENTS

x survival time objects, like that returned by the function survive.
breaks either a vector of breakpoints or the number of equal-width intervals into which the values of
x should be cut. If breaks is 0 or omitted, KpMeier is called, otherwise, Lifetable is called.

VALUE

a survival function object as described in KpMeier or Lifetable.

NOTE

If break points are given to breaks the right end of intervals may be given. The values of x
are cut into left open and right closed intervals. For example, if x takes only integers from 0
to 7 and the desired intervals are [0, 3] and [4, 7] then a simplest way is to give c(3,7) for
breaks.

EXAMPLES

survive.fun(tsurv1)

table.assoc.test

Interactive test of association for contingency table

table.assoc.test

table.assoc.test(x, y, strata, correct=T, alt="two")

ARGUMENTS

x, y categories, matrix or array. NAs are allowed but ignored at calculation
strata vector giving sequence number of stratum to which each value of x and y belongs. If given,
Mahoe.test is always called.
correct continuity correction is applied if possible.
alt alternative hypothesis. This argument is required only for Fis.test.

VALUE

a list representing the result of a test of the association for a contingency table, Mahoe.test,
Mc.test, StMc.test, or Fis.test or Mahoe.test.

If x and y are dichotomous categories and strata is missing, the user will be asked to
choose one from Fis.test, StMc.test or Mc.test. Otherwise, choices are only StMc.test
or Mc.test.

EXAMPLES

table.assoc.test(Nc.1, Nc.7)
table.assoc.test(Visual)