

# Package ‘userfriendlyscience’

July 2, 2014

**Type** Package

**Title** Quantitative analysis made accessible

**Version** 0.1-2

**Date** 2014-02-25

**Author** Gjalt-Jorn Peters

**Maintainer** Gjalt-Jorn Peters <gjalt-jorn@userfriendlyscience.com>

**License** GPL (>= 2)

**Description** This package contains a number of functions that serve two goals: first, make R more accessible to people migrating from SPSS by adding a number of functions that behave roughly like their SPSS equivalents; and second, make a number of slightly more advanced functions more userfriendly to relatively novice users.

**URL** <http://userfriendlyscience.com>

**Imports** plyr, psych, pwr, fBasics, e1071, ltm, MBESS, foreign, knitr, xtable, GGally, grid, lavaan

**Depends** ggplot2

**Suggests** XLConnect

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-04-22 07:26:56

**R topics documented:**

basicSPSStranslationFunctions . . . . .	2
convert . . . . .	4
dlvPlot . . . . .	5
freq . . . . .	7
itemInspection . . . . .	8
meanDiff . . . . .	9
meanDiff.multi . . . . .	12
normalityAssessment . . . . .	13
oddsratio . . . . .	15
powerHist . . . . .	16
rMatrix . . . . .	17
rnwString . . . . .	19
scaleDiagnosis . . . . .	20
scaleInspection . . . . .	22
scaleReliability . . . . .	24
sysrev . . . . .	26
testRetestAlpha . . . . .	27
testRetestCES . . . . .	28
testRetestReliability . . . . .	30
testRetestSimData . . . . .	32
userfriendlyscience . . . . .	33
userfriendlyscienceBasics . . . . .	34
userfriendlysciencePrintMethods . . . . .	36
<b>Index</b>	<b>39</b>

---

basicSPSStranslationFunctions  
*Basic SPSS translation functions*

---

**Description**

Basic functions to make working with R easier for SPSS users: `getData` provides an easy way to load SPSS datafiles; `filterBy` and `useAll` allow easy temporary filtering of rows from the dataframe; `mediaan` and `modus` compute the median and mode of ordinal or numeric data.

**Usage**

```
getData(filename=NULL, errorMessage = "[defaultErrorMessage]",
         use.value.labels=TRUE, to.data.frame=TRUE, stringsAsFactors=TRUE, ...)
filterBy(dat, expression, replaceOriginalDataframe = TRUE, envir = parent.frame())
useAll(dat, replaceFilteredDataframe = TRUE)
mediaan(vector)
modus(vector)
```

**Arguments**

<code>filename</code>	It is possible to specify a path and filename to load here. If not specified, the default R file selection dialogue is shown
<code>errorMessage</code>	The error message that is shown if the file does not exist or does not have the right extension; "[defaultErrorMessage]" is replaced with a default error message (and can be included in longer messages).
<code>use.value.labels</code>	Only useful when reading from SPSS files: whether to read variables with value labels as factors (TRUE) or numeric vectors (FALSE).
<code>to.data.frame</code>	Only useful when reading from SPSS files: whether to return a dataframe or not.
<code>stringsAsFactors</code>	Whether to read strings as strings (FALSE) or factors (TRUE).
<code>...</code>	Additional options, passed on to the function used to import the data (which depends on the extension of the file).
<code>dat</code>	Dataframe to process: for <code>filterBy</code> , dataframe to filter rows from; for <code>useAll</code> , dataframe to restore ('unfilter').
<code>expression</code>	Logical expression determining which rows to keep and which to drop. Can be either a logical vector or a string which is then evaluated. If it's a string, it's evaluated using 'with' to evaluate the expression using the variable names.
<code>replaceOriginalDataframe</code>	Whether to also replace the original dataframe in the parent environment. Very messy, but for maximum compatibility with the 'SPSS way of doing things', by default, this is true. After all, people who care about the messiness/inappropriateness of this function wouldn't be using it in the first place :-)
<code>envir</code>	The environment where to create the 'backup' of the unfiltered dataframe, for when <code>useAll</code> is called and the filter is deactivated again.
<code>replaceFilteredDataframe</code>	Whether to replace the filtered dataframe passed in the 'dat' argument (see <code>replaceOriginalDataframe</code> ).
<code>vector</code>	For <code>mediaan</code> and <code>modus</code> , the vector for which to find the median or mode.

**Value**

`getData` returns the imported dataframe, with the filename from which it was read stored in the 'filename' attribute.

`mediaan` returns the median, or, in the case of a factor where the median is in between two categories, both categories.

`modus` returns the mode.

**Note**

`getData()` currently can't read from LibreOffice or OpenOffice files. There doesn't seem to be a platform-independent package that allows this. Non-CRAN package `ROpenOffice` from Omega-Hat should be able to do the trick, but fails to install (manual download and installation using

<http://www.omegahat.org> produces "ERROR: dependency 'Rcompression' is not available for package 'ROpenOffice'" - and manual download and installation of RCompression produces "Please define LIB\_ZLIB; ERROR: configuration failed for package 'Rcompression'"). If you have any suggestions, please let me know!

## Examples

```
### Open a dialogue to read an SPSS file
### (wrapped in 'if (interactive())' to prevent execution during
### the testing of the examples when building the package)
if (interactive()) {
  getData();
}

### Get a median and a mode
mediaan(c(1,2,2,3,4,4,5,6,6,6,7));
modus(c(1,2,2,3,4,4,5,6,6,6,7));

### Create an example dataframe
(exampleDat <- data.frame(x=rep(8, 8), y=rep(c(0,1), each=4)));
### Filter it, replacing the original dataframe
(filterBy(exampleDat, "y=0"));
### Restore the old dataframe
(useAll(exampleDat));
```

---

convert

*convert functions*

---

## Description

These are a number of functions to convert statistics and effect size measures from/to each other.

## Usage

```
convert.r.to.t(r, n)
convert.t.to.r(t, n)
convert.t.to.p(t, df)
convert.f.to.p(f, df1, df2, lower.tail=FALSE)
convert.chisq.to.p(chisq, df, lower.tail=FALSE)
```

## Arguments

<code>r</code> , <code>t</code> , <code>f</code> , <code>chisq</code>	The value of the relevant statistic
<code>n</code>	The number of observations that the <code>r</code> or <code>t</code> value is based on.
<code>df</code> , <code>df1</code> , <code>df2</code>	The degrees of freedom for that statistic (for <code>F</code> , the first one is the numerator (i.e. the effect), and the second one the denominator (i.e. the error term).
<code>lower.tail</code>	For the <code>F</code> and chisquare distributions, whether to get the probability of the lower or upper tail.

**Value**

The converted value as a numeric value.

**Examples**

```
convert.t.to.r(t=-6.46, n=200);
convert.r.to.t(r=-.41, n=200);

### Compute some p-values
convert.t.to.p(4.2, 197);
convert.chisq.to.p(5.2, 3);
convert.f.to.p(8.93, 3, 644);
```

---

dlvPlot

*dlvPlot*


---

**Description**

The `dlvPlot` function produces a dot-violin-line plot, and `dlvTheme` is the default theme.

**Usage**

```
dlvPlot(dat, x = NULL, y, z = NULL, conf.level = .95,
        jitter = "FALSE", binnedDots = TRUE, binwidth=NULL,
        error="lines", dotsize="density", densityDotBaseSize=3,
        normalDotBaseSize=3, violinAlpha = .2, dotAlpha = .4,
        lineAlpha = 1, meanDotSize=5, posDodge=0)
dlvTheme(base_size = 14, base_family = "")
```

**Arguments**

<code>dat</code>	The dataframe containing <code>x</code> , <code>y</code> and <code>z</code> .
<code>x</code>	Character value with the name of the predictor ('independent') variable, must refer to a categorical variable (i.e. a factor).
<code>y</code>	Character value with the name of the criterion ('dependent') variable, must refer to a continuous variable (i.e. a numeric vector).
<code>z</code>	Character value with the name of the moderator variable, must refer to a categorical variable (i.e. a factor).
<code>conf.level</code>	Confidence of confidence intervals.
<code>jitter</code>	Logical value (i.e. <code>TRUE</code> or <code>FALSE</code> ) whether or not to jitter individual data-points. Note that jitter cannot be combined with <code>posDodge</code> (see below).
<code>binnedDots</code>	Logical value indicating whether to use binning to display the dots. Overrides jitter and dotsize.

binwidth	Numeric value indicating how broadly to bin (larger values is more binning, i.e. combining more dots into one big dot).
error	Character value: "none", "lines" or "whiskers"; indicates whether to show the confidence interval as lines with (whiskers) or without (lines) horizontal whiskers or not at all (none)
dotsize	Character value: "density" or "normal"; when "density", the size of each dot corresponds to the density of the distribution at that point.
densityDotBaseSize	Numeric value indicating base size of dots when their size corresponds to the density (bigger = larger dots).
normalDotBaseSize	Numeric value indicating base size of dots when their size is fixed (bigger = larger dots).
violinAlpha	Numeric value indicating alpha value of violin layer (0 = completely transparent, 1 = completely opaque).
dotAlpha	Numeric value indicating alpha value of dot layer (0 = completely transparent, 1 = completely opaque).
lineAlpha	Numeric value indicating alpha value of line layer (0 = completely transparent, 1 = completely opaque).
meanDotSize	Numeric value indicating the size of the dot used to indicate the mean in the line layer.
posDodge	Numeric value indicating the distance to dodge positions (0 for complete overlap).
base_size, base_family	Passed on to the ggplot theme_grey() function.

## Details

This function creates Dot Violin Line plots. One image says more than a thousand words; I suggest you run the example :-)

## Value

The behavior of this function depends on the arguments.

If no x and z are provided and y is a character value, dlvPlot produces a univariate plot for the numerical y variable.

If no x and z are provided, and y is c character vector, dlvPlot produces multiple Univariate plots, with variable names determining categories on x-axis and with numerical y variables on y-axis

If both x and y are a character value, and no z is provided, dlvPlot produces a bivariate plot where factor x determines categories on x-axis with numerical variable y on the y-axis (roughly a line plot with a single line)

Finally, if x, y and z are each a character value, dlvPlot produces multivariate plot where factor x determines categories on x-axis, factor z determines the different lines, and with the numerical y variable on the y-axis

An object is returned with the following elements:

dat.raw	Raw datafile provided when calling dlvPlot
dat	Transformed (long) datafile dlvPlot uses
descr	Dataframe with extracted descriptives used to plot the mean and confidence intervals
yRange	The range of the Y variable used to construct the plot
plot	The plot itself

### Examples

```
### Create simple dataset
dat <- data.frame(x1 = factor(rep(c(0,1), 20)),
                 x2 = factor(c(rep(0, 20), rep(1, 20))),
                 y=rep(c(4,5), 20) + rnorm(40));
### Generate a simple dlvPlot of y
dlvPlot(dat, y='y');
### Now add a predictor
dlvPlot(dat, x='x1', y='y');
### And finally also a moderator:
dlvPlot(dat, x='x1', y='y', z='x2');
### The number of datapoints might be a bit clearer if we jitter
dlvPlot(dat, x='x1', y='y', z='x2', jitter=TRUE);
### Although just dodging the density-sized dots might work better
dlvPlot(dat, x='x1', y='y', z='x2', posDodge=.3);
```

---

freq	<i>Freq</i>
------	-------------

---

### Description

Function to show frequencies in a manner similar to what SPSS' "FREQUENCIES" command does.

### Usage

```
freq(vector, digits = 1, nsmall=1, transposed=FALSE, round=1)
```

### Arguments

vector	A vector of values to compute frequencies for.
digits	Minimum number of significant digits to show in result.
nsmall	Minimum number of digits after the decimal point to show in the result.
transposed	Whether to transpose the results when printing them (this can be useful for blind users).
round	Number of digits to round the results to (can be used in conjunction with digits to determine format of results).

**Value**

An object with several elements, the most notable of which is:

`dat`                    A dataframe with the frequencies

**Examples**

```
### Create factor vector
ourFactor <- factor(mtcars$gear, levels=c(3,4,5),
                   labels=c("three", "four", "five"));
### Add some missing values
factorWithMissings <- ourFactor;
factorWithMissings[10] <- factorWithMissings[20] <- NA;

### Show frequencies
freq(ourFactor);
freq(factorWithMissings);
```

---

itemInspection	<i>itemInspection</i>
----------------	-----------------------

---

**Description**

Function to generate a PDF with four panels per page, showing some basic item characteristics.

**Usage**

```
itemInspection(dat, items,
              docTitle = "Scale inspection", docAuthor = "Author",
              pdfLaTeXPath, rnwPath, filename="itemInspection",
              convertFactors = TRUE, digits=4)
```

**Arguments**

<code>dat</code>	Dataframe containing the items of the relevant scale
<code>items</code>	Either a character vector with the itemnames, or, if the items are organised in scales, a list of character vectors with the items in each scale.
<code>docTitle</code>	Title to use when generating the PDF.
<code>docAuthor</code>	Author(s) to include when generating the PDF.
<code>pdfLaTeXPath</code>	The path to PdfLaTeX. This file is part of a LaTeX installation that creates a pdf out of a .tex file. In Windows, you can download (portable) MikTeX from <a href="http://miktex.org/portable">http://miktex.org/portable</a> . You then decide yourself where to install MikTeX; pdflatex will end up in a sub-folder 'miktex\bin', so if you installed MikTeX in, for example, 'C:\Program Files\MikTeX', the total path becomes 'C:\Program Files\MikTeX\miktex\bin'.



Note that R uses slashes instead of backslashes to separate folders, so in this example, pdfLaTeXPath should be 'C:/Program Files/MikTeX/miktex/bin'

In MacOS, you can install MacTeX from <http://tug.org/mactex/> By default, pdflatex ends up in folder '/user/texbin', which is what pdfLaTeXPath should be in that default case.

In Ubuntu, you can install TexLive base by using your package manager to install texlive-latex-base, or using the terminal: 'sudo apt-get install texlive-latex-base' In ubuntu, by default pdflatex ends un in folder '/usr/bin', which is what pdfLaTeXPath should be in that default case.

rnwPath	The path where the temporary files and the resulting PDF should be stored.
filename	The filename to use to save the pdf.
convertFactors	Whether to convert factors to numeric vectors for the analysis.
digits	The number of digits to use in the tables.

### Value

This function returns nothing; it just generates a PDF.

### Examples

```
## Not run:
itemInspection(mtcars, items=c('disp', 'hp', 'drat'), pdfLaTeXPath="valid/path/here");

## End(Not run)
```

---

meanDiff

*meanDiff*

---

### Description

The meanDiff function compares the means between two groups. It computes Cohen's d, the unbiased estimate of Cohen's d (Hedges' g), and performs a t-test. It also shows the achieved power, and, more usefully, the power to detect small, medium, and large effects.

### Usage

```
meanDiff(x, y=NULL, paired = FALSE, var.equal = "test",
         conf.level = .95, digits = 2, envir = parent.frame())
```

**Arguments**

x	Dichotomous factor: variable 1; can also be a formula of the form $y \sim x$ , where x must be a factor with two levels (i.e. dichotomous).
y	Numeric vector: variable 2; can be empty if x is a formula.
paired	Boolean; are x & y independent or dependent? Note that if x & y are dependent, they need to have the same length.
var.equal	String; only relevant if x & y are independent; can be "test" (default; test whether x & y have different variances), "no" (assume x & y have different variances; see the Warning below!), or "yes" (assume x & y have the same variance)
conf.level	Confidence of confidence intervals you want.
digits	With what precision you want the results to print.
envir	The environment where to search for the variables (useful when calling meanDiff from a function where the vectors are defined in that functions environment).

**Details**

This function uses the formulae from Borenstein, Hedges, Higgins & Rothstein (2009) (pages 25-32).

**Value**

An object is returned with the following elements:

variables	Input variables
groups	Levels of the x variable, the dichotomous factor
ci.confidence	Confidence of confidence intervals
digits	Number of digits for output
x	Values of dependent variable in first group
y	Values of dependent variable in second group
type	Type of t-test (independent or dependent, equal variances or not)
n	Sample sizes of the two groups
mean	Means of the two groups
sd	Standard deviations of the two groups
objects	Objects used; the t-test and optionally the test for equal variances
variance	Variance of the difference score
meanDiff	Difference between the means
meanDiff.d	Cohen's d
meanDiff.d.var	Variance of Cohen's d
meanDiff.d.se	Standard error of Cohen's d
meanDiff.J	Correction for Cohen's d to get to the unbiased Hedges g
power	Achieved power with current effect size and sample size

power.small	Power to detect small effects with current sample size
power.medium	Power to detect medium effects with current sample size
power.large1	Power to detect large effects with current sample size
meanDiff.g	Hedges' g
meanDiff.g.var	Variance of Hedges' g
meanDiff.g.se	Standard error of Hedges' g
ci.usedZ	Z value used to compute confidence intervals
meanDiff.d.ci.lower	Lower bound of confidence interval around Cohen's d
meanDiff.d.ci.upper	Upper bound of confidence interval around Cohen's d
meanDiff.g.ci.lower	Lower bound of confidence interval around Hedges' g
meanDiff.g.ci.upper	Upper bound of confidence interval around Hedges' g
meanDiff.ci.lower	Lower bound of confidence interval around raw mean
meanDiff.ci.upper	Upper bound of confidence interval around raw mean
t	Student t value for Null Hypothesis Significance Testing
df	Degrees of freedom for t value
p	p-value corresponding to t value

### Warning

Note that when different variances are assumed for the t-test (i.e. the null-hypothesis test), the values of Cohen's d are still based on the assumption that the variance is equal. In this case, the confidence interval might, for example, not contain zero even though the NHST has a non-significant p-value (the reverse can probably happen, too).

### References

Borenstein, M., Hedges, L. V., Higgins, J. P., & Rothstein, H. R. (2011). Introduction to meta-analysis. John Wiley & Sons.

### Examples

```
### Create simple dataset
dat <- PlantGrowth[1:20,];
### Remove third level from group factor
dat$group <- factor(dat$group);
### Compute mean difference and show it
meanDiff(dat$weight ~ dat$group);

### Look at second treatment
dat <- rbind(PlantGrowth[1:10,], PlantGrowth[21:30,]);
```

```
### Remove third level from group factor
dat$group <- factor(dat$group);
### Compute mean difference and show it
meanDiff(x=dat$group, y=dat$weight);
```

---

 meanDiff.multi

*meanDiff.multi*


---

## Description

The meanDiff.multi function compares many means for many groups. It presents the results in a dataframe summarizing all relevant information, and produces plot showing the confidence intervals for the effect sizes for each predictor (i.e. dichotomous variable). Like meanDiff, it computes Cohen's d, the unbiased estimate of Cohen's d (Hedges' g), and performs a t-test. It also shows the achieved power, and, more usefully, the power to detect small, medium, and large effects.

## Usage

```
meanDiff.multi(dat, y, x=NULL, var.equal = "yes", conf.level = .95,
              digits = 2, orientation = "vertical",
              zeroLineColor = "grey", zeroLineSize = 1.2,
              envir = parent.frame())
```

## Arguments

dat	The dataframe containing the variables involved in the mean tests.
y	Character vector containing the list of interval variables to include in the tests.
x	Character vector containing the list of the dichotomous variables to include in the tests. If x is empty, paired samples t-tests will be conducted.
var.equal	String; only relevant if x & y are independent; can be "test" (default; test whether x & y have different variances), "no" (assume x & y have different variances; see the Warning below!), or "yes" (assume x & y have the same variance)
conf.level	Confidence of confidence intervals you want.
digits	With what precision you want the results to print.
orientation	Whether to plot the effect size confidence intervals vertically (like a forest plot, the default) or horizontally.
zeroLineColor	Color of the horizontal line at an effect size of 0 (set to 'white' to not display the line; also adjust the size to 0 then).
zeroLineSize	Size of the horizontal line at an effect size of 0 (set to 0 to not display the line; also adjust the color to 'white' then).
envir	The environment where to search for the variables (useful when calling meanDiff from a function where the vectors are defined in that functions environment).

**Details**

This function uses the meanDiff function, which uses the formulae from Borenstein, Hedges, Higgins & Rothstein (2009) (pages 25-32).

**Value**

An object is returned with the following elements:

results.raw	Objects returned by the calls to meanDiff.
plots	For every comparison, a plot with the datapoints, means, and confidence intervals in the two groups.
results.compiled	Dataframe with the most important results from each comparison.
plots.compiled	For every dichotomous (x) variable, a plot with the confidence interval for the effect size of each dependent (y) variable.
input	The arguments with which the function was called.

**Warning**

Note that when different variances are assumed for the t-test (i.e. the null-hypothesis test), the values of Cohen's d are still based on the assumption that the variance is equal. In this case, the confidence interval might, for example, not contain zero even though the NHST has a non-significant p-value (the reverse can probably happen, too).

**References**

Borenstein, M., Hedges, L. V., Higgins, J. P., & Rothstein, H. R. (2011). Introduction to meta-analysis. John Wiley & Sons.

**Examples**

```
### Create simple dataset
dat <- data.frame(x1 = factor(rep(c(0,1), 20)),
                 x2 = factor(c(rep(0, 20), rep(1, 20))),
                 y=rep(c(4,5), 20) + rnorm(40));
### Compute mean difference and show it
meanDiff.multi(dat, x=c('x1', 'x2'), y='y', var.equal="yes");
```

---

normalityAssessment    *normalityAssessment*

---

**Description**

This function can be used to assess whether a variable and the sampling distribution of its mean have an approximately normal distribution.

**Usage**

```
normalityAssessment(sampleVector, samples = 5000, digits=3,
                    samplingDistColor = "#2222CC",
                    normalColor = "#00CC00",
                    samplingDistLineSize = 2,
                    normalLineSize = 1,
                    xlabel.sampleDist = NULL,
                    ylabel.sampleDist = NULL,
                    xlabel.samplingDist = NULL,
                    ylabel.samplingDist = NULL)
```

**Arguments**

sampleVector	Numeric vector containing the sample data.
samples	Number of samples to use when constructing sampling distribution.
digits	Number of digits to use when printing results.
samplingDistColor	Color to use when drawing the sampling distribution.
normalColor	Color to use when drawing the standard normal curve.
samplingDistLineSize	Size of the line used to draw the sampling distribution.
normalLineSize	Size of the line used to draw the standard normal distribution.
xlabel.sampleDist	Label of x axis of the distribution of the sample.
ylabel.sampleDist	Label of y axis of the distribution of the sample.
xlabel.samplingDist	Label of x axis of the sampling distribution.
ylabel.samplingDist	Label of y axis of the sampling distribution.

**Details**

This function provides a number of normality tests and draws histograms of the sample data and the sampling distribution of the mean (most statistical tests assume the latter is normal, rather than the first; normality of the sample data guarantees normality of the sampling distribution of the mean, but if the sample size is sufficiently large, the sampling distribution of the mean is approximately normal even when the sample data are not normally distributed). Note that for the sampling distribution, the degrees of freedom are usually so huge that the normality tests, negligible deviations from normality will already result in very small p-values.

**Value**

An object with several results, the most notably of which are:

plot.sampleDist	Histogram of sample distribution
-----------------	----------------------------------

```

sw.sampleDist  Shapiro-Wilk normality test of sample distribution
ad.sampleDist  Anderson-Darling normality test of sample distribution
ks.sampleDist  Kolmogorov-Smirnof normality test of sample distribution
kurtosis.sampleDist
                Kurtosis for sample distribution
skewness.sampleDist
                Skewness for sample distribution
plot.samplingDist
                Histogram of sampling distribution
sw.samplingDist
                Shapiro-Wilk normality test of sampling distribution
ad.samplingDist
                Anderson-Darling normality test of sampling distribution
ks.samplingDist
                Kolmogorov-Smirnof normality test of sampling distribution
kurtosis.samplingDist
                Kurtosis for sampling distribution
skewness.samplingDist
                Skewness for sampling distribution

```

### Examples

```
normalityAssessment(rnorm(35));
```

---

oddsratio

*oddsratio*

---

### Description

The `oddsratio` function simply computes a point estimate and confidence interval for an odds ratio.

### Usage

```
oddsratio(x, y = NULL, conf.level = .95, digits=2)
```

### Arguments

<code>x</code>	<code>x</code> can be either a table (then <code>y</code> can be <code>NULL</code> ) or a factor.
<code>y</code>	If <code>x</code> is a factor, <code>y</code> also has to be a factor; <code>x</code> and <code>y</code> are then used to create the crosstable.
<code>conf.level</code>	The confidence level of the confidence interval.
<code>digits</code>	Number of digits to round output to

**Value**

The oddsratio function returns an object with the input and output.

input	List with input arguments
or	Point estimate for odds ratio
or.ci	Confidence interval for odds ratio

**Examples**

```
### Generate two factor vectors
treatment <- factor(c(rep(0, 33), rep(1, 45), rep(0, 63), rep(1, 21)),
  levels=c(0,1), labels=c("no", "yes"));
survival <- factor(c(rep(0, 78), rep(1, 84)),
  levels=c(0, 1), labels=c("no", "yes"));

### Compute and display odds ratio
oddsratio(treatment, survival);

### Or present a table
oddsratio(table(treatment, survival));
```

---

powerHist

*powerHist*

---

**Description**

powerHist generates a histogram with a density curve and a normal density curve.

**Usage**

```
powerHist(vector, distributionColor = "#2222CC",
  normalColor = "#00CC00", distributionLineSize = 2,
  normalLineSize = 1, xLabel = NULL,
  yLabel = NULL, density=TRUE)
```

**Arguments**

vector	A numeric vector.
distributionColor	The colour to use for the distribution (the histogram and the density curve).
normalColor	The colour to use for the normal curve.
distributionLineSize	The line size to use for the distribution density curve.
normalLineSize	The line size to use for the normal curve.
xLabel	Label to use on x axis.
yLabel	Label to use on y axis.
density	This argument is not in use yet.



**Value**

An object, with the following elements:

input	The input when the function was called.
intermediate	The intermediate numbers and distributions.
dat	The dataframe used to generate the plot.
plot	The histogram.

**Examples**

```
powerHist(mtcars$mpg)
```

---

rMatrix

*rMatrix*


---

**Description**

rMatrix provides a correlation matrix with confidence intervals and a p-value adjusted for multiple testing.

**Usage**

```
rMatrix(dat, x, y=NULL, conf.level = .95, correction = "fdr",
        digits = 2, pval=FALSE, colspace=2, rowspace=0,
        colNames = "numbers",
        output="R",
        env.LaTeX = 'tabular',
        pboxWidthMultiplier = 1)
```

**Arguments**

dat	A dataframe containing the relevant variables.
x	Vector of 1+ variable names.
y	Vector of 1+ variable names; if this is left empty, a symmetric matrix is created; if this is filled, the matrix will have the x variables defining the rows and the y variables defining the columns.
conf.level	The confidence of the confidence intervals.
correction	Correction for multiple testing: an element out of the vector c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"). NOTE: the p-values are corrected for multiple testing; The confidence intervals are not (yet :-)).
digits	With what precision do you want the results to print.

pval	Determines whether format.pval is used to display the p-value. This will add three characters to the width of columns in case p-values require scientific notation.
colspace	Number of spaces between columns (only for R output, ignored for LaTeX output)
rowspace	Number of rows between table rows (note: one table row is 2 rows; only for R output, ignored for LaTeX output).
colNames	colNames can be "numbers" or "names". "Names" cause variables names to be printed in the heading; "numbers" causes the rows to become numbered and the numbers to be printed in the heading.
output	Can be "R" or "LaTeX"; if output is set to "LaTeX", the result is a LaTeX table (e.g. for use in knitr).
env.LaTeX	For LaTeX output, the environment can be set with env.LaTeX.
pboxWidthMultiplier	When using LaTeX, pboxWidthMultiplier can be used to make the cells narrower or wider (1 works for anything up until 4 or 5 digits).

### Details

rMatrix provides a symmetric or asymmetric matrix of correlations their confidence intervals, and p-values. The p-values can be corrected for multiple testing.

### Value

An object with the input and several output variables. Most notably a number of matrices:

r	Pearson r values.
parameter	Degrees of freedom.
ci.lo	Lower bound of Pearson r confidence interval.
ci.hi	Upper bound of Pearson r confidence interval.
p.raw	Original p-values.
p.adj	p-values adjusted for multiple testing.

### Examples

```
rMatrix(mtcars, x=c('disp', 'hp', 'drat'))
```

---

rnwString	<i>rnwString functions</i>
-----------	----------------------------

---

### Description

The `rnwString` functions make knitting PFDs a bit more userfriendly.

The `sanitizeLatexString` function sanitizes a LaTeX string by escaping special characters. It is strongly based on the function described on <http://stackoverflow.com/questions/5406071/r-sweave-latex-escape-variables-to-be-printed-in-latex> by Aaron Rendahl.

### Usage

```
rnwString.initiate(studyName, authorName,
                  docClassArgs = 'a4paper,portrait,11pt',
                  newPage = TRUE, pageMargins=15)
rnwString.terminate(rnwString)
rnwString.generate(rnwString, rnwPath, fileName, pdfLatexPath,
                  envir = parent.frame())
sanitizeLatexString(str)
hasLaTeX(pdfLatexPath)
```

### Arguments

<code>studyName</code>	The name of the study - used as the title of the PDF.
<code>authorName</code>	The name of the author(s) - also inserted on title page of the PDF.
<code>docClassArgs</code>	Default arguments for the document class in LaTeX. For example, to use landscape pages, this should be <code>'a4paper,landscape,11pt'</code> .
<code>newPage</code>	Whether to end the initiation string with a <code>newpage</code> command. This can be set to false if you want to add more information on the first page(s).
<code>pageMargins</code>	Margin of the pages in millimeters.
<code>rnwString</code>	The <code>rnwString</code> to terminate or (after termination) generate.
<code>rnwPath</code>	The path where the temporary files ( <code>.rnw</code> , <code>.tex</code> , etc) should be created. Use forward slashes. Note: the last character should not be a slash!
<code>fileName</code>	The filename to use for the temporary files. Omit the extension!
<code>pdfLatexPath</code>	The path to PdfLaTeX. This file is part of a LaTeX installation that creates a pdf out of a <code>.tex</code> file. In Windows, you can download (portable) MikTeX from <a href="http://miktex.org/portable">http://miktex.org/portable</a> . You then decide yourself where to install MikTeX; <code>pdflatex</code> will end up in a subfolder <code>'miktex\bin'</code> , so if you installed MikTeX in, for example, <code>'C:\Program Files\MikTeX'</code> , the total path becomes <code>'C:\Program Files\MikTeX\miktex\bin'</code> . Note that R uses slashes instead of backslashes to separate folders, so in this example, <code>pdfLatexPath</code> should be <code>'C:/Program Files/MikTeX/miktex/bin'</code> In MacOS, you can install MacTeX from <a href="http://tug.org/mactex/">http://tug.org/mactex/</a> By default, <code>pdflatex</code> ends up in folder <code>'/user/texbin'</code> , which is what <code>pdfLatexPath</code> should be in that default case.

In Ubuntu, you can install TexLive base by using your package manager to install texlive-latex-base, or using the terminal: 'sudo apt-get install texlive-latex-base' In ubuntu, by default pdflatex ends un in folder '/usr/bin', which is what pdfLatexPath should be in that default case.

envir	The environment where to evaluate the expressions (normally the environment where the function is called).
str	The character string to sanitize.

### Value

rnwString.initiate starts an rnwString; rnwString.terminate closes it; and rnwString.generate takes an rnwString and creates a pdf.

sanitizeLatexString returns the sanitized string.

hasLaTeX checks pdfLatexPath to make sure pdflatex or pdflatex.exe exists.

### Examples

```
### sanitize a string
newString <- sanitizeLatexString('this is a tilde: ~. ');
newString;
### newString is now: "this is a tilde: ~."
```

---

scaleDiagnosis	<i>scaleDiagnosis</i>
----------------	-----------------------

---

### Description

scaleDiagnosis provides a number of diagnostics for a scale (an aggregative measure consisting of several items).

### Usage

```
scaleDiagnosis(dat=NULL, items=NULL, plotSize=180, sizeMultiplier = 1,
               axisLabels = "none", scaleReliability.ci=FALSE, conf.level=.95)
```

### Arguments

dat	A dataframe containing the items in the scale. All variables in this dataframe will be used if items is NULL.
items	If not NULL, this should be a character vector with the names of the variables in the dataframe that represent items in the scale.
plotSize	Size of the final plot in millimeters.
sizeMultiplier	Allows more flexible control over the size of the plot elements
axisLabels	Passed to ggpairs function to set axisLabels.

scaleReliability.ci	TRUE or FALSE: whether to compute confidence intervals for Cronbach's Alpha and Omega (uses bootstrapping function in MBESS, takes a while).
conf.level	Confidence of confidence intervals for reliability estimates (if requested with scaleReliability.ci).

## Details

Function to generate an object with several useful statistics and a plot to assess how the elements (usually items) in a scale relate to each other, such as Cronbach's Alpha, omega, the Greatest Lower Bound, a factor analysis, and a correlation matrix.

## Value

An object with the input and several output variables. Most notably:

scaleReliability	The results of scaleReliability.
pca	A Principal Components Analysis
fa	A Factor Analysis
describe	Decriptive statistics about the items
ggpairs.combined	A scattermatrix with histograms on the diagonal and correlation coefficients in the upper right half.

## Examples

```
## Not run:
### This will prompt the user to select an SPSS file
scaleDiagnosis();

## End(Not run)

### Generate a datafile to use
exampleData <- data.frame(item1=rnorm(100));
exampleData$item2 <- exampleData$item1+rnorm(100);
exampleData$item3 <- exampleData$item1+rnorm(100);
exampleData$item4 <- exampleData$item2+rnorm(100);
exampleData$item5 <- exampleData$item2+rnorm(100);

### Use all items
scaleDiagnosis(dat=exampleData);

### Use a selection of two variables
scaleDiagnosis(dat=exampleData, items=c('item2', 'item4'));
```

---

scaleInspection      *scaleInspection and a number of useful helper functions*

---

## Description

scaleInspection is a function to generate a PDF with information to diagnose and inspect scales (aggregate measures); makeScales actually generates the scales; and meanConfInt and sdConfInt provide confidence intervals for means and standard deviations.

## Usage

```
scaleInspection(dat, items = NULL,
               docTitle = "Scale inspection", docAuthor = "Author",
               pdfLaTeXPath, rnwPath=getwd(),
               filename = "scaleInspection", convertFactors=TRUE,
               scaleReliability.ci=FALSE, conf.level=.95, digits=2,
               rMatrixColsLandscape = 6,
               pboxWidthMultiplier = 1,
               scatterPlotBaseSize = 4,
               pageMargins=15,
               pval=TRUE)
makeScales(dat, scales)
meanConfInt(vector=NULL, mean=NULL, sd=NULL, n=NULL, se=NULL, conf.level=.95)
sdConfInt(vector=NULL, sd=NULL, n=NULL, conf.level=.95)
```

## Arguments

dat	Dataframe containing the items of the relevant scale
items	Either a character vector with the itemnames, or, if the items are organised in scales, a list of character vectors with the items in each scale.
scales	A list of character vectors with the items in each scale, where each vectors' name is the name of the scale.
docTitle	Title to use when generating the PDF.
docAuthor	Author(s) to include when generating the PDF.
pdfLaTeXPath	The path to PdfLaTeX. This file is part of a LaTeX installation that creates a pdf out of a .tex file. In Windows, you can download (portable) MikTeX from <a href="http://miktex.org/portable">http://miktex.org/portable</a> . You then decide yourself where to install MikTeX; pdflatex will end up in a subfolder 'miktex\bin', so if you installed MikTeX in, for example, 'C:\Program Files\MikTeX', the total path becomes 'C:\Program Files\MikTeX\miktex\bin'. Note that R uses slashes instead of backslashes to separate folders, so in this example, pdfLaTeXPath should be 'C:/Program Files/MikTeX/miktex/bin' In MacOS, you can install MacTex from <a href="http://tug.org/mactex/">http://tug.org/mactex/</a> By default, pdflatex ends up in folder '/user/texbin', which is what pdfLaTeXPath should be in that default case.

In Ubuntu, you can install TexLive base by using your package manager to install texlive-latex-base, or using the terminal: 'sudo apt-get install texlive-latex-base' In ubuntu, by default pdflatex ends un in folder '/usr/bin', which is what pdfLaTeXPath should be in that default case.

rnwPath	The path where the temporary files and the resulting PDF should be stored.
filename	The filename to use to save the pdf.
convertFactors	Whether to convert factors to numeric vectors for the analysis.
scaleReliability.ci	TRUE or FALSE: whether to compute confidence intervals for Cronbach's Alpha and Omega (uses bootstrapping function in MBESS, takes a while).
conf.level	Confidence of confidence intervals (for reliability estimates (if requested with scaleReliability.ci), meand, and sd, for respectively scaleInspection, meanConfInt and sdConfInt).
digits	The number of digits to use in the tables.
rMatrixColsLandscape	At how many columns (or rather, variables) or more should rMatrices be printed landscape?
pboxWidthMultiplier	Used for print.rMatrix; used to tweak the width of columns in the correlation matrix.
scatterPlotBaseSize	Size of one scatterplot in the scattermatrix in centimeters. If the total scattermatrix becomes larger than 18 cm, it's scaled down to 18 cm.
pageMargins	Margins of the page in millimeters.
pval	Whether to print p-values as p-values in correlation matrix.
vector	Numeric vector to use when computing confidence intervals.
mean	Mean to use when computing confidence intervals (when no vector is provided).
sd	Standard deviaton to use when computing confidence intervals (when no vector is provided).
n	Number of datapoints to base confidence intervals on.\
se	Standard error to use when computing confidence intervals (when no standard deviation or vector is provided).

## Details

scaleInspection generates a PDF with useful diagnostics to assess a scale; those from scaleDiagnosis and an rMatrix.

makeScales generates the scales and stores them in the dataframe.

meanConfInt and sdConfInt just compute and return a confidence interval for a mean or standard deviation.

**Value**

scaleInspection returns nothing; it just generates a PDF.

makeScales returns the provided dataframe, now including the new scale variables.

meanConfInt and sdConfInt return an object, with in its 'output' list, the confidence interval for a mean or standard deviation.

**Examples**

```
## Not run:
scaleInspection(mtcars, items=c('disp', 'hp', 'drat'), pdfLaTeXPath="valid/path/here");

## End(Not run)

newDataframe <- makeScales(mtcars, list(senselessScale = c('disp', 'hp', 'drat')));

sdConfInt(sd=4, n=30);

meanConfInt(mean=5, sd=4, n=30)
```

---

scaleReliability      *scale.ic*

---

**Description**

The scale.ic function computes a number of measures to assess scale reliability and internal consistency.

**Usage**

```
scaleReliability(dat=NULL, items = 'all', digits = 2, ci = TRUE,
                 conf.level=.95, silent=FALSE, samples=1000, bootstrapSeed = NULL,
                 omega.psych = FALSE)
```

**Arguments**

dat	A dataframe containing the items in the scale. All variables in this dataframe will be used if items = 'all'.
items	If not 'all', this should be a character vector with the names of the variables in the dataframe that represent items in the scale.
digits	Number of digits to use in the presentation of the results.
ci	Whether to compute confidence intervals as well (this can take a while!).
conf.level	The confidence of the confidence intervals.
silent	If computing confidence intervals, the user is warned that it may take a while, unless silent=TRUE.



samples	The number of samples to compute for the bootstrapping of the confidence intervals.
bootstrapSeed	The seed to use for the bootstrapping - setting this seed makes it possible to replicate the exact same intervals, which is useful for publications.
omega.psych	Whether to also compute omega using the function 'omega' in the 'psych' package. This method differs from the one used by the MBESS package, and usually (perhaps always) results in higher estimates for omega.

### Details

This function is basically a wrapper for functions from the psych, GPArotation, ltm, and MBESS packages that compute measures of reliability and internal consistency. For backwards compatibility, in addition to scaleReliability, scale.ic can still be used (but this is obsolete and therefore strongly discouraged).

### Value

An object with the input and several output variables. Most notably:

input	Input specified when calling the function
intermediate	Intermediate values and objects computed to get to the final results
output	Values of reliability / internal consistency measures, with as most notable elements:
output\$dat	A dataframe with the most important outcomes
output\$omega	Point estimate for omega
output\$glb	Point estimate for the Greatest Lower Bound
output\$alpha	Point estimate for Cronbach's alpha
output\$omega.ci	Confidence interval for omega
output\$alpha.ci	Confidence interval for Cronbach's alpha

### Examples

```
## Not run:
### This will prompt the user to select an SPSS file
scaleReliability();

## End(Not run)

### Load data from simulated dataset testRetestSimData (which
### satisfies essential tau-equivalence).
data(testRetestSimData);

### Select some items in the first measurement
exampleData <- testRetestSimData[2:6];
```

```

### Use all items
scaleReliability(dat=exampleData, ci=FALSE);

### Use a selection of two variables
scaleReliability(dat=exampleData, items=c('t0_item2', 't0_item4'), ci=FALSE);

### Note that 'ci = FALSE' can be omitted to compute confidence intervals
### for scales of three items or more.

```

---

sysrev

*sysrev functions*


---

## Description

These functions are meant for use in systematic reviews.

## Usage

```

sysrev.read.bibtex(filename, encoding="unknown")
sysrev.export(libraryObject, filename, drop=NULL, keep=NULL,
              sep="\t", row.names = FALSE, ...)

```

## Arguments

filename	The path and filename to read from or write to
encoding	When reading bibtex files, you might want to specify the encoding (passed on to 'readLines').
libraryObject	An object, created with sysrev.read.bibtex, with references.
drop	Character vector with names of fields to drop when exporting (rest is kept). Cannot be used at the same time as 'keep'.
keep	Character vector with names of fields to keep when exporting (rest is dropped). Cannot be used at the same time as 'drop'.
sep	Separator to use when writing exported file (defaults to tab).
row.names	Whether to write rownames in the export (passed on to write.table).
...	Additional arguments passed on to write.table().

## Details

sysrev.read.bibtex and sysrev.export and for importing bibtex files, and after manipulation, exporting them to a 'spreadsheet format'. t.to.p, f.to.p, and chisq.to.p make it slightly easier to compute p-values from reported statistics.

**Value**

sysrev.read.bibtex returns a BibTeX Library object (a collection of references).

sysrev.export returns nothing.

t.to.p, f.to.p and chisq.to.p return a p-value.

**Examples**

```
## Not run:
bibTexObject <- sysrev.read.bibtex("BibTeX library.bib");
sysrev.export(bibTexObject, "references in table.csv");

## End(Not run)
```

---

testRetestAlpha	<i>testRetestAlpha</i>
-----------------	------------------------

---

**Description**

The testRetestAlpha function computes the test-retest alpha coefficient (Green, 2003).

**Usage**

```
testRetestAlpha(dat = NULL, moments = NULL,
                testDat = NULL, retestDat = NULL,
                sortItems = FALSE, convertToNumeric = TRUE)
```

**Arguments**

dat	A dataframe containing the items in the scale at both measurement moments. If no dataframe is specified, a dialogue will be launched to allow the user to select an SPSS datafile. If only one dataframe is specified, either the items have to be ordered chronologically (i.e. first all items for the first measurement, then all items for the second measurement), or the vector 'moments' has to be used to indicate, for each item, to which measurement moment it belongs.
moments	Used to indicate to which measurement moment each item in 'dat' belongs; should be a vector with the same length as dat has columns, and with two possible values (e.g. 1 and 2).
testDat, retestDat	Dataframes with the items for each measurement moment: note that the items have to be in the same order (unless sortItems is TRUE).
sortItems	If true, the columns (items) in each dataframe are ordered alphabetically before starting. This can be convenient to ensure that the order of the items at each measurement moment is the same.
convertToNumeric	When TRUE, the function will attempt to convert all vectors in the dataframes to numeric.

**Details**

This function computes the test-retest alpha coefficient as described in Green (2003).

**Value**

An object with the input and several output variables. Most notably:

input	Input specified when calling the function
intermediate	Intermediate values and objects computed to get to the final results
output\$testRetestAlpha	The value of the test-retest alpha coefficient.

**References**

Green, S. N. (2003). A Coefficient Alpha for Test-Retest Data. *Psychological Methods*, 8(1), 88-101. doi:10.1037/1082-989X.8.1.88

**Examples**

```
## Not run:
### This will prompt the user to select an SPSS file
testRetestAlpha();

## End(Not run)

### Load data from simulated dataset testRetestSimData (which
### satisfies essential tau-equivalence).
data(testRetestSimData);

### The first column is the true score, so it's excluded in this example.
exampleData <- testRetestSimData[, 2:ncol(testRetestSimData)];

### Compute test-retest alpha coefficient
testRetestAlpha(exampleData);
```

---

testRetestCES

*testRetestCES*


---

**Description**

The testRetestCES function computes the test-retest Coefficient of Equivalence and Stability (Schmidt, Le & Ilies, 2003).

**Usage**

```
testRetestCES(dat = NULL, moments = NULL,
              testDat = NULL, retestDat = NULL,
              parallelTests = 'means',
              sortItems = FALSE, convertToNumeric = TRUE,
              digits=4)
parallelSubscales(dat, convertToNumeric = TRUE)
```

**Arguments**

dat	A dataframe. For testRetestCES, this dataframe must contain the items in the scale at both measurement moments. If no dataframe is specified, a dialogue will be launched to allow the user to select an SPSS datafile. If only one dataframe is specified, either the items have to be ordered chronologically (i.e. first all items for the first measurement, then all items for the second measurement), or the vector 'moments' has to be used to indicate, for each item, to which measurement moment it belongs. The number of columns in this dataframe <b>MUST</b> be even! Note that instead of providing this dataframe, the items of each measurement moment can be provided separately in testDat and retestDat as well.
moments	Used to indicate to which measurement moment each item in 'dat' belongs; should be a vector with the same length as dat has columns, and with two possible values (e.g. 1 and 2).
testDat, retestDat	Dataframes with the items for each measurement moment: note that the items have to be in the same order (unless sortItems is TRUE).
parallelTests	A vector indicating which items belong to which parallel test; like the moments vector, this should have two possible values (e.g. 1 and 2). Alternatively, it can be character value with 'means' or 'variances'; in this case, parallelSubscales will be used to create roughly parallel halves.
sortItems	If true, the columns (items) in each dataframe are ordered alphabetically before starting. This can be convenient to ensure that the order of the items at each measurement moment is the same.
convertToNumeric	When TRUE, the function will attempt to convert all vectors in the dataframes to numeric.
digits	Number of digits to print.

**Details**

This function computes the test-retest Coefficient of Equivalence and Stability (CES) as described in Schmidt, Le & Ilies (2003). Note that this function only computes the test-retest CES for a scale that is administered twice and split into two parallel halves post-hoc (this procedure is explained on page 210, and the equations that are used, 16 and 17a are explained on page 212).

**Value**

An object with the input and several output variables. Most notably:

input	Input specified when calling the function
intermediate	Intermediate values and objects computed to get to the final results
output\$testRetestCES	The value of the test-retest Coefficient of Equivalence and Stability.

**Note**

This function uses equations 16 and 17 on page 212 of Schmidt, Le & Ilies (2003): in other words, this function assumes that one scale is administered twice. If you'd like the computation for two different but parallel scales/measures to be implemented, please contact me.

**References**

Schmidt, F. L., Le, H., & Ilies, R. (2003) Beyond Alpha: An Empirical Examination of the Effects of Different Sources of Measurement Error on Reliability Estimates for Measures of Individual-differences Constructs. *Psychological Methods*, 8(2), 206-224. doi:10.1037/1082-989X.8.x.206

**Examples**

```
## Not run:
### This will prompt the user to select an SPSS file
testRetestCES();

## End(Not run)

### Load data from simulated dataset testRetestSimData (which
### satisfies essential tau-equivalence).
data(testRetestSimData);

### The first column is the true score, so it's excluded in this example.
exampleData <- testRetestSimData[, 2:ncol(testRetestSimData)];

### Compute test-retest alpha coefficient
testRetestCES(exampleData);
```

---

testRetestReliability *testRetestReliability*

---

**Description**

The testRetestReliability function is a convenient interface to testRetestAlpha and testRetestCES.

**Usage**

```
testRetestReliability(dat = NULL, moments = NULL,
  testDat = NULL, retestDat = NULL,
  parallelTests = 'means',
  sortItems = FALSE, convertToNumeric = TRUE,
  digits=2)
```

**Arguments**

<code>dat</code>	A dataframe. This dataframe must contain the items in the scale at both measurement moments. If no dataframe is specified, a dialogue will be launched to allow the user to select an SPSS datafile. If only one dataframe is specified, either the items have to be ordered chronologically (i.e. first all items for the first measurement, then all items for the second measurement), or the vector 'moments' has to be used to indicate, for each item, to which measurement moment it belongs. The number of columns in this dataframe <b>MUST</b> be even! Note that instead of providing this dataframe, the items of each measurement moment can be provided separately in <code>testDat</code> and <code>retestDat</code> as well.
<code>moments</code>	Used to indicate to which measurement moment each item in 'dat' belongs; should be a vector with the same length as <code>dat</code> has columns, and with two possible values (e.g. 1 and 2).
<code>testDat, retestDat</code>	Dataframes with the items for each measurement moment: note that the items have to be in the same order (unless <code>sortItems</code> is TRUE).
<code>parallelTests</code>	A vector indicating which items belong to which parallel test; like the <code>moments</code> vector, this should have two possible values (e.g. 1 and 2). Alternatively, it can be character value with 'means' or 'variances'; in this case, <code>parallelSubscales</code> will be used to create roughly parallel halves.
<code>sortItems</code>	If true, the columns (items) in each dataframe are ordered alphabetically before starting. This can be convenient to ensure that the order of the items at each measurement moment is the same.
<code>convertToNumeric</code>	When TRUE, the function will attempt to convert all vectors in the dataframes to numeric.
<code>digits</code>	Number of digits to show when printing the output

**Details**

This function calls both `testRetestAlpha` and `testRetestCES` to compute and print measures of the test-retest reliability.

**Value**

An object with the input and several output variables. Most notably:

<code>input</code>	Input specified when calling the function
<code>intermediate</code>	Intermediate values and objects computed to get to the final results

```
output$testRetestAlpha
    The value of the test-retest alpha coefficient.
output$testRetestCES
    The value of the test-retest Coefficient of Equivalence and Stability.
```

### Examples

```
## Not run:
### This will prompt the user to select an SPSS file
testRetestReliability();

## End(Not run)

### Load data from simulated dataset testRetestSimData (which
### satisfies essential tau-equivalence).
data(testRetestSimData);

### The first column is the true score, so it's excluded in this example.
exampleData <- testRetestSimData[, 2:ncol(testRetestSimData)];

### Compute test-retest alpha coefficient
testRetestReliability(exampleData);
```

---

testRetestSimData	<i>testRetestSimData is a simulated dataframe used to demonstrate the testRetestAlpha coefficient function.</i>
-------------------	---

---

### Description

This dataset contains the true scores of 250 participants on some variable, and 10 items of a scale administered twice (at t0 and at t1).

### Usage

```
data(testRetestSimData)
```

### Format

A data frame with 250 observations on the following 21 variables.

```
trueScore The true scores
t0_item1 Score on item 1 at test
t0_item2 Score on item 2 at test
t0_item3 Score on item 3 at test
t0_item4 Score on item 4 at test
t0_item5 Score on item 5 at test
```



```
t0_item6 Score on item 6 at test
t0_item7 Score on item 7 at test
t0_item8 Score on item 8 at test
t0_item9 Score on item 9 at test
t0_item10 Score on item 10 at test
t1_item1 Score on item 1 at retest
t1_item2 Score on item 2 at retest
t1_item3 Score on item 3 at retest
t1_item4 Score on item 4 at retest
t1_item5 Score on item 5 at retest
t1_item6 Score on item 6 at retest
t1_item7 Score on item 7 at retest
t1_item8 Score on item 8 at retest
t1_item9 Score on item 9 at retest
t1_item10 Score on item 10 at retest
```

### Details

This dataset was generated with the code in the `reliabilityTest.r` test script.

### Examples

```
data(testRetestSimData);
head(testRetestSimData);
hist(testRetestSimData$t0_item1);
cor(testRetestSimData);
```

---

userfriendlyscience    *Userfriendlyscience*

---

### Description

This package contains a number of functions that serve two goals: first, make R more accessible to people migrating from SPSS by adding a number of functions that behave roughly like their SPSS equivalents; and second, make a number of slightly more advanced functions more userfriendly to relatively novice users.

### Details

Package: userfriendlyscience  
Type: Package  
Version: 0.1.1  
Date: 2014-02-25  
License: GPL (>= 2)

The package contains a variety of functions designed to make life easier. These functions are geared towards researchers in psychology.

### Author(s)

Gjalt-Jorn Peters

Maintainer: Gjalt-Jorn Peters <gjalt-jorn@userfriendlyscience.com>

### Examples

```
### Generate two factor vectors
treatment <- factor(c(rep(0, 33), rep(1, 45), rep(0, 63), rep(1, 21)),
                   levels=c(0,1), labels=c("no", "yes"));
survival <- factor(c(rep(0, 78), rep(1, 84)),
                  levels=c(0, 1), labels=c("no", "yes"));

### Compute and display odds ratio
oddsratio(treatment, survival);

### Or present a table
oddsratio(table(treatment, survival));
```

---

userfriendlyscienceBasics

*userfriendlyscience basics*

---

### Description

The userfriendlyscience basics functions are some very basic functions to make life that little bit easier.

### Usage

```
safeRequire(packageName)
trim(str)
noZero(str)
formatR(r, digits)
repeatStr(str = " ", n = 1)
ifelseObj(condition, ifTrue, ifFalse)
invertItem(item, range=NULL, ignorePreviousInversion = FALSE)
is.odd(vector)
is.even(vector)
convertToNumeric(vector, byFactorLabel = FALSE)
massConvertToNumeric(dat, byFactorLabel = FALSE, ignoreCharacter = TRUE)
```

**Arguments**

packageName	The name of the package, as character string.
str	The character string to process.
r	The Pearson correlation to format.
digits	The number of digits to use when formatting the Pearson correlation.
n	The number of times to repeat the string.
condition	Condition to evaluate.
ifTrue	Object to return if the condition is true.
ifFalse	Object to return if the condition is false.
item	Item to invert
range	If provided it must be a numeric vector with the minimum and the maximum of the scale. If not provided, the range function is used (so, use this range argument if the scale minimum and/or maximum do not occur in the data).
ignorePreviousInversion	If this item has already been inverted, the function will halt with an error unless it's told to ignore previous inversions with this boolean.
dat, vector	The dataframe of vector to process.
byFactorLabel	If TRUE, convertToNumeric and massConvertToNumeric use the factor labels, interpreted as character vectors, to determine the numeric value, instead of the level's indices (which is what as.numeric() does).
ignoreCharacter	If TRUE, character vectors are ignored. If FALSE, character vectors are converted (or, an attempt is made :-)).

**Details**

The `safeRequire` function checks whether a package is already installed. If so, it loads the package (using `require/library`). If not, it first installs it, and then loads it.

The `trim` function removes whitespaces from the start and end of a text string.

The `noZero` function removes the first zero from a string that was originally a number.

The `formatR` function format a Pearson correlation for pretty printing (using `noZero`).

The `repeatStr` function repeats a string a given number of times.

The `ifelseObj` function just evaluates a condition, returning one object if it's true, and another if it's false.

The `invertItem` function 'unmirrors' an inverted item (i.e. for a 1-3 item, 1 becomes 3, 2 stays 2, and 3 becomes 1).

`is.odd` and `is.even` check whether a number is, or numbers in a vector are, odd or even.

**Value**

safeRequire returns nothing.

trim, noZero, formatR, and repeatStr return a string.

ifelseObj return an object.

The invertItem function returns the inverted item vector, with an attribute "inverted" set to TRUE.

is.odd and is.even return a logical vector.

**Examples**

```
### load a package
safeRequire('ggplot2');

### trim a string
trim(' this is a string with whitespace in front and at the end ');
### Returns "this is a string with whitespace in front and at the end"

repeatStr("-", 8);
### Returns "-----" (incredibly useful, no? :-))

tempVector <- c(1,2,3,3,2,4,3,2,1,1,3,4,5,4,3,2,2,1,1,2);
invertedTempVector <- invertItem(tempVector);

### We can also invert it back, but then we have to override the security
### that prevents accidentally inverting items back.
invertItem(tempVector, ignorePreviousInversion=TRUE);
```

---

userfriendlysciencePrintMethods  
*userfriendlyscience print methods*

---

**Description**

These methods print the userfriendlyscience objects.

**Usage**

```
## S3 method for class 'dlvPlot'
print(x, ...)
## S3 method for class 'freq'
print(x, digits=x$input$digits, nsmall=x$input$nsmall,
      transposed=x$input$transposed, ...)
## S3 method for class 'meanConfInt'
print(x, digits=2, ...)
## S3 method for class 'meanDiff'
print(x, digits=x$digits, powerDigits=x$digits + 2, ...)
```

```

## S3 method for class 'meanDiff.multi'
print(x, digits=x$digits,
powerDigits=x$digits + 2, ...)
## S3 method for class 'normalityAssessment'
print(x, ...)
## S3 method for class 'oddsratio'
print(x, digits=x$input$digits, ...)
## S3 method for class 'powerHist'
print(x, ...)
## S3 method for class 'rMatrix'
print(x, digits=x$digits, output=x$output,
env.LaTeX = x$env.LaTeX,
pboxWidthMultiplier = x$pboxWidthMultiplier,
colNames = x$colNames, pval=x$pval, ...)
## S3 method for class 'scaleDiagnosis'
print(x, ...)
## S3 method for class 'scaleReliability'
print(x, digits=x$input$digits, ...)
## S3 method for class 'sdConfInt'
print(x, digits=2, ...)
## S3 method for class 'testRetestAlpha'
print(x, ...)
## S3 method for class 'testRetestCES'
print(x, digits=x$input$digits, ...)
## S3 method for class 'testRetestReliability'
print(x, digits=x$input$digits, ...)
## S3 method for class 'parallelSubscales'
print(x, nsmall=2, ...)

```

### Arguments

<code>x</code>	The object to print
<code>digits</code>	The number of significant digits to print.
<code>nsmall</code>	The minimum number of digits to the right of the decimal point in formatting real/complex numbers in non-scientific formats. Allowed values are $0 \leq nsmall \leq 20$ .
<code>transposed</code>	Whether the frequency object should be printed transposed (this can be useful for blind users).
<code>powerDigits</code>	Number of digits to use when printing the power.
<code>output, env.LaTeX, pboxWidthMultiplier, colNames, pval</code>	Output to produce; see <code>?rMatrix</code> for details.
<code>...</code>	Addition arguments that are passed on to the print functions when it's called.

### Value

These printing methods return nothing, but print stuff.

**Examples**

```
print.sdConfInt(sdConfInt(sd=4, n=20));
```

# Index

- \*Topic **datasets**
  - testRetestSimData, 32
- \*Topic **file**
  - basicSPSStranslationFunctions, 2
- \*Topic **package**
  - userfriendlyscience, 33
- \*Topic **univar**
  - basicSPSStranslationFunctions, 2
  - freq, 7
  - scaleDiagnosis, 20
  - scaleReliability, 24
  - testRetestAlpha, 27
  - testRetestCES, 28
  - testRetestReliability, 30
- \*Topic **utilities**
  - basicSPSStranslationFunctions, 2
  - convert, 4
  - dlvPlot, 5
  - itemInspection, 8
  - meanDiff, 9
  - meanDiff.multi, 12
  - normalityAssessment, 13
  - oddsratio, 15
  - powerHist, 16
  - rMatrix, 17
  - rnwString, 19
  - scaleDiagnosis, 20
  - scaleInspection, 22
  - scaleReliability, 24
  - sysrev, 26
  - testRetestAlpha, 27
  - testRetestCES, 28
  - testRetestReliability, 30
  - userfriendlyscienceBasics, 34
  - userfriendlysciencePrintMethods, 36
- basicSPSStranslationFunctions, 2
- convert, 4
- convertToNumeric
  - (userfriendlyscienceBasics), 34
- dlvPlot, 5
- dlvTheme (dlvPlot), 5
- filterBy
  - (basicSPSStranslationFunctions), 2
- formatR (userfriendlyscienceBasics), 34
- freq, 7
- getData
  - (basicSPSStranslationFunctions), 2
- hasLaTeX (rnwString), 19
- ifelseObj (userfriendlyscienceBasics), 34
- invertItem (userfriendlyscienceBasics), 34
- is.even (userfriendlyscienceBasics), 34
- is.odd (userfriendlyscienceBasics), 34
- itemInspection, 8
- makeScales (scaleInspection), 22
- massConvertToNumeric
  - (userfriendlyscienceBasics), 34
- meanConfInt (scaleInspection), 22
- meanDiff, 9
- meanDiff.multi, 12
- mediaan
  - (basicSPSStranslationFunctions), 2
- modus (basicSPSStranslationFunctions), 2
- normalityAssessment, 13
- noZero (userfriendlyscienceBasics), 34
- oddsratio, 15

parallelSubscales (testRetestCES), 28  
 powerHist, 16  
 print.dlvPlot  
     (userfriendlysciencePrintMethods),  
     36  
 print.freq  
     (userfriendlysciencePrintMethods),  
     36  
 print.meanConfInt  
     (userfriendlysciencePrintMethods),  
     36  
 print.meanDiff  
     (userfriendlysciencePrintMethods),  
     36  
 print.normalityAssessment  
     (userfriendlysciencePrintMethods),  
     36  
 print.oddsratio  
     (userfriendlysciencePrintMethods),  
     36  
 print.parallelSubscales  
     (userfriendlysciencePrintMethods),  
     36  
 print.powerHist  
     (userfriendlysciencePrintMethods),  
     36  
 print.rMatrix  
     (userfriendlysciencePrintMethods),  
     36  
 print.scaleDiagnosis  
     (userfriendlysciencePrintMethods),  
     36  
 print.scaleReliability  
     (userfriendlysciencePrintMethods),  
     36  
 print.sdConfInt  
     (userfriendlysciencePrintMethods),  
     36  
 print.testRetestAlpha  
     (userfriendlysciencePrintMethods),  
     36  
 print.testRetestCES  
     (userfriendlysciencePrintMethods),  
     36  
 print.testRetestReliability  
     (userfriendlysciencePrintMethods),  
     36  
 repeatStr (userfriendlyscienceBasics),  
     34  
 rMatrix, 17  
 rnwString, 19  
 safeRequire  
     (userfriendlyscienceBasics), 34  
 sanitizeLatexString (rnwString), 19  
 scaleDiagnosis, 20  
 scaleInspection, 22  
 scaleReliability, 24  
 sdConfInt (scaleInspection), 22  
 sysrev, 26  
 testRetestAlpha, 27  
 testRetestCES, 28  
 testRetestReliability, 30  
 testRetestSimData, 32  
 trim (userfriendlyscienceBasics), 34  
 useAll (basicSPSStranslationFunctions),  
     2  
 userfriendlyscience, 33  
 userfriendlyscience-package  
     (userfriendlyscience), 33  
 userfriendlyscienceBasics, 34  
 userfriendlysciencePrintMethods, 36