

# Package ‘compareGroups’

January 29, 2018

**Type** Package

**Title** Descriptive Analysis by Groups

**Version** 3.3.3

**Date** 2018-01-28

**Depends** R (>= 2.13.1), Hmisc, gdata, xtable, SNPassoc

**Imports** survival, epitools, tools, HardyWeinberg, rmarkdown, knitr

**Suggests** tcltk2, car, shiny, shinyBS, shinyjs, DT, xlsx

**Author** Isaac Subirana, Joan Vila, Héctor Sanz, Gavin Lucas, Judith Peñafiel and David Giménez

**Maintainer** Isaac Subirana <isubirana@imim.es>

**Description** Create data summaries for quality control, extensive reports for exploring data, as well as publication-ready univariate or bivariate tables in several formats (plain text, HTML, LaTeX, PDF, Word or Excel. Create figures to quickly visualise the distribution of your data (boxplots, barplots, normality-plots, etc.). Display statistics (mean, median, frequencies, incidences, etc.). Perform the appropriate tests (t-test, Analysis of variance, Kruskal-Wallis, Fisher, log-rank, ...) depending on the nature of the described variable (normal, non-normal or qualitative). Summarize genetic data (Single Nucleotide Polymorphisms) data displaying Allele Frequencies and performing Hardy-Weinberg Equilibrium tests among other typical statistics and tests for these kind of data.

**License** GPL (>= 2)

**URL** <http://www.comparegroups.eu>

**LazyLoad** yes

**Encoding** latin1

**BuildVignettes** True

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-01-29 22:44:51 UTC

## R topics documented:

compareGroups-package . . . . .	2
cGroupsGUI . . . . .	3
cGroupsWUI . . . . .	4
compareGroups . . . . .	5
compareSNPs . . . . .	11
createTable . . . . .	13
export2csv . . . . .	18
export2html . . . . .	19
export2latex . . . . .	20
export2md . . . . .	22
export2pdf . . . . .	23
export2word . . . . .	24
export2xls . . . . .	25
getResults . . . . .	26
missingTable . . . . .	28
predimed . . . . .	29
printTable . . . . .	31
radiograph . . . . .	32
regicor . . . . .	33
report . . . . .	35
varinfo . . . . .	36
<b>Index</b>	<b>38</b>

---

compareGroups-package *Descriptive analysis by groups*

---

## Description

Create data summaries for quality control, extensive reports for exploring data, as well as publication-ready univariate or bivariate tables in several formats (plain text, HTML, LaTeX, PDF, Word or Excel). Display statistics (mean, median, frequencies, incidences, etc.). Create figures to quickly visualise the distribution of your data (boxplots, barplots, normality-plots, etc.). Perform the appropriate tests (t-test, Analysis of variance, Kruskal-Wallis, Fisher, log-rank, ...) depending on the nature of the described variable (normal, non-normal or qualitative). Summarize genetic data (Single Nucleotide Polymorphisms) data displaying Allele Frequencies and performing Hardy-Weinberg Equilibrium tests among other typical statistics and tests for these kind of data.

## Details

Package: compareGroups  
 Type: Package  
 Version: 3.3.3  
 Date: 2018-01-28  
 License: GPL version 2 or newer  
 LazyLoad: yes

Main functions: `compareGroups`, `compareSNPs`, `createTable`, `missingTable`, `export2latex`, `export2html`, `export2csv`, `export2pdf`, `export2md`, `export2word`, `export2xls`, `report`, `radiograph`, `cGroupsGUI`, `cGroupsWUI`

### Author(s)

Main functions: Isaac Subirana <isubirana@imim.es>, Joan Vila <jvila@imim.es>, H'ector Sanz <hsrodenas@gmail.com>, Gavin Lucas <gavin.lucas@cleargenetics.com> and David Gim'enez <dgimenez1@imim.es>

Web User Interface: Isaac Subirana <isubirana@imim.es>, Judith Pe\~nafi'el <jpenafiel@imim.es>, Gavin Lucas <gavin.lucas@cleargenetics.com> and David Gim'enez <dgimenez1@imim.es>

Maintainer: Isaac Subirana <isubirana@imim.es>

### References

Isaac Subirana, Hector Sanz, Joan Vila (2014). Building Bivariate Tables: The `compareGroups` Package for R. Journal of Statistical Software, 57(12), 1-16. URL <http://www.jstatsoft.org/v57/i12/>.

---

cGroupsGUI

*Graphical user interface to create tables of descriptives by groups (bivariate tables)*

---

### Description

This function allows the user to build tables in an easy and intuitive way and to modify several options, using a graphical interface.

### Usage

```
cGroupsGUI(X)
```

### Arguments

`X` a matrix or a data.frame. 'X' must exist in `.GlobalEnv`.

### Details

See the vignette for more detailed examples illustrating the use of this function.

### Note

If a data.frame or a matrix is passed through 'X' argument or is loaded by the 'Load data' GUI menu, this object is placed in the `.GlobalEnv`. Manipulating this data.frame or matrix while GUI is opened may produce an error in executing the GUI operations.

**See Also**

[cGroupsWUI](#), [compareGroups](#), [createTable](#)

**Examples**

```
## Not run:  
data(regicor)  
cGroupsGUI(regicor)  
  
## End(Not run)
```

---

cGroupsWUI

*Web User Interface.*

---

**Description**

This function opens a web browser with a graphical interface based on **shiny** package.

**Usage**

```
cGroupsWUI(port = 8102L)
```

**Arguments**

port            integer. Same as 'port' argument of [runApp](#). Default value is 8102L.

**Note**

If an error occurs when launching the web browser, it may be solved by changing the port number.

**See Also**

[cGroupsGUI](#), [compareGroups](#), [createTable](#)

**Examples**

```
## Not run:  
require(compareGroups)  
cGroupsWUI()  
  
## End(Not run)
```

---

compareGroups

*Descriptives by groups*


---

## Description

This function performs descriptives by groups for several variables. Depending on the nature of these variables, different descriptive statistics are calculated (mean, median, frequencies or K-M probabilities) and different tests are computed as appropriate (t-test, ANOVA, Kruskal-Wallis, Fisher, log-rank, ...).

## Usage

```
compareGroups(X, ...)
## Default S3 method:
compareGroups(X, y = NULL, Xext = NULL, selec = NA, method = 1, timemax = NA,
alpha = 0.05, min.dis = 5, max.ylev = 5, max.xlev = 10, include.label = TRUE,
Q1 = 0.25, Q3 = 0.75, simplify = TRUE, ref = 1, ref.no = NA, fact.ratio = 1,
ref.y = 1, p.corrected = TRUE, compute.ratio = TRUE, include.miss = FALSE,
oddsratio.method = "midp", chisq.test.perm = FALSE, byrow = FALSE, ...)
## S3 method for class 'formula'
compareGroups(X, data, subset, na.action=NULL, include.label=TRUE, ...)
## S3 method for class 'compareGroups'
plot(x, file, type = "pdf", bivar = FALSE, z=1.5, n.breaks = "Sturges", ...)
```

## Arguments

- |       |  |
|-------|--|
| X     | either a data.frame or a matrix (then method 'compareGroups.default' is called), or a formula (then method 'compareGroups.formula' is called). When 'X' is a formula, it must be an object of class "formula" (or one that can be coerced to that class). Right side of ~ must have the terms in an additive way, and left side of ~ must contain the name of the grouping variable or can be left in blank (in this latter case descriptives for whole sample are calculated and no test is performed). |
| y     | a vector variable that distinguishes the groups. It must be either a numeric, character, factor or NULL. Default value is NULL which means that descriptives for whole sample are calculated and no test is performed.   |
| Xext  | a data.frame or a matrix with the same rows / individuals contained in X, and maybe with different variables / columns than X. This argument is used by compareGroups.default in the sense that the variables specified in the argument selec are searched in Xext and/or in the <code>.GlobalEnv</code> . If Xext is NULL, then Xext is created from variables of X plus y. Default value is NULL.  |
| selec | a list with as many components as row-variables. If list length is 1 it is recycled for all row-variables. Every component of 'selec' is an expression that will be evaluated to select the individuals to be analyzed for every row-variable. Otherwise, a named list specifying 'selec' row-variables is applied. '.else' is a reserved name that defines the selection for the rest of the variables; if no '.else'   |

	variable is defined, default value is applied for the rest of the variables. Default value is NA; all individuals are analyzed (no subsetting).
method	integer vector with as many components as row-variables. If its length is 1 it is recycled for all row-variables. It only applies for continuous row-variables (for factor row-variables it is ignored). Possible values are: 1 - forces analysis as "normal-distributed"; 2 - forces analysis as "continuous non-normal"; 3 - forces analysis as "categorical"; and 4 - NA, which performs a Shapiro-Wilks test to decide between normal or non-normal. Otherwise, a named vector specifying 'method' row-variables is applied. '.else' is a reserved name that defines the method for the rest of the variables; if no '.else' variable is defined, default value is applied. Default value is 1.
timemax	double vector with as many components as row-variables. If its length is 1 it is recycled for all row-variables. It only applies for 'Surv' class row-variables (for all other row-variables it is ignored). This value indicates at which time the K-M probability is to be computed. Otherwise, a named vector specifying 'timemax' row-variables is applied. '.else' is a reserved name that defines the 'timemax' for the rest of the variables; if no '.else' variable is defined, default value is applied. Default value is NA; K-M probability is then computed at the median of observed times.
alpha	double between 0 and 1. Significance threshold for the <a href="#">shapiro.test</a> normality test for continuous row-variables. Default value is 0.05.
min.dis	an integer. If a non-factor row-variable contains less than 'min.dis' different values and 'method' argument is set to NA, then it will be converted to a factor. Default value is 5.
max.ylev	an integer indicating the maximum number of levels of grouping variable ('y'). If 'y' contains more than 'max.ylev' levels, then the function 'compareGroups' produces an error. Default value is 5.
max.xlev	an integer indicating the maximum number of levels when the row-variable is a factor. If the row-variable is a factor (or converted to a factor if it is a character, for example) and contains more than 'max.xlev' levels, then it is removed from the analysis and a warning is printed. Default value is 10.
data	an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If they are not found in 'data', the variables are taken from 'environment(formula)'.
subset	an optional vector specifying a subset of individuals to be used in the computation process. It is applied to all row-variables. 'subset' and 'selec' are added in the sense of '&' to be applied in every row-variable.
na.action	a function which indicates what should happen when the data contain NAs. The default is NULL, and that is equivalent to <a href="#">na.pass</a> , which means no action. Value <a href="#">na.exclude</a> can be useful if it is desired to removed all individuals with some NA in any variable.
include.label	logical, indicating whether or not variable labels have to be shown in the results. Default value is TRUE
Q1	double between 0 and 1, indicating the quantile to be displayed as the first number inside the square brackets in the bivariate table. To compute the minimum just type 0. Default value is 0.25 which means the first quartile.

Q3	double between 0 and 1, indicating the quantile to be displayed as the second number inside the square brackets in the bivariate table. To compute the maximum just type 1. Default value is 0.75 which means the third quartile.
simplify	logical, indicating whether levels with no values must be removed for grouping variable and for row-variables. Default value is TRUE.
ref	an integer vector with as many components as row-variables. If its length is 1 it is recycled for all row-variables. It only applies for categorical row-variables. Or a named vector specifying which row-variables 'ref' is applied (a reserved name is '.else' which defines the reference category for the rest of the variables); if no '.else' variable is defined, default value is applied for the rest of the variables. Default value is 1.
ref.no	character specifying the name of the level to be the reference for Odds Ratio or Hazard Ratio. It is not case-sensitive. This is especially useful for yes/no variables. Default value is NA which means that category specified in 'ref' is the one selected to be the reference.
fact.ratio	a double vector with as many components as row-variables indicating the units for the HR / OR (note that it does not affect the descriptives). If its length is 1 it is recycled for all row-variables. Otherwise, a named vector specifying 'fact.ratio' row-variables is applied. '.else' is a reserved name that defines the reference category for the rest of the variables; if no '.else' variable is defined, default value is applied. Default value is 1.
ref.y	an integer indicating the reference category of y variable for computing the OR, when y is a binary factor. Default value is 1.
p.corrected	logical, indicating whether p-values for pairwise comparisons must be corrected. It only applies when there is a grouping variable with more than 2 categories. Default value is TRUE.
compute.ratio	logical, indicating whether Odds Ratio (for a binary response) or Hazard Ratio (for a time-to-event response) must be computed. Default value is TRUE.
include.miss	logical, indicating whether to treat missing values as a new category for categorical variables. Default value is FALSE.
oddsratio.method	Which method to compute the Odds Ratio. See 'method' argument from <a href="#">oddsratio</a> . Default value is "midp".
x	an object of class 'compareGroups'.
file	a character string giving the name of the file. A bmp, jpg, png or tif file is saved with an appendix added to 'file' corresponding to the row-variable name. If 'onefile' argument is set to TRUE through '...' argument of plot method function, a unique PDF file is saved named as [file].pdf. If it is missing, multiple devices are opened, one for each row-variable of 'x' object.
type	a character string indicating the file format where the plots are stored. Possibles formats are 'bmp', 'jpg', 'png', 'tif' and 'pdf'. Default value is 'pdf'.
bivar	logical. If bivar=TRUE, it plots a boxplot or a barplot (for a continuous or categorical row-variable, respectively) stratified by groups. If bivar=FALSE, it plots a normality plot (for continuous row-variables) or a barplot (for categorical row-variables). Default value is FALSE.

<code>z</code>	double. Indicates threshold limits to be placed in the deviation from normality plot. It is considered that too many points beyond this threshold indicates that current variable is far to be normal-distributed. Default value is 1.5.
<code>n.breaks</code>	same as argument 'breaks' of <code>hist</code> .
<code>chisq.test.perm</code>	logical. It applies a permutation chi squared test ( <code>chisq.test</code> ) instead of an exact Fisher test ( <code>fisher.test</code> ).
<code>byrow</code>	logical. Percentage of categorical variables must be reported by rows. Default value is FALSE, which means that percentages are reported by columns (withing groups).
<code>...</code>	further arguments passed to 'compareGroups.default' or other methods. For 'plot' method, '...' arguments are passed to <code>pdf</code> , <code>bmp</code> , <code>jpeg</code> , <code>png</code> or <code>tiff</code> if 'type' argument equals to 'pdf', 'bmp', 'jpg', 'png' or 'tif', respectively.

## Details

Depending whether the row-variable is considered as continuous normal-distributed (1), continuous non-normal distributed (2) or categorical (3), the following descriptives and tests are performed:

1- mean, standard deviation and t-test or ANOVA

2- median, 1st and 3rd quartiles (by default), and Kruskal-Wallis test

3- or absolute and relative frequencies and chi-squared or exact Fisher test when the expected frequencies is less than 5 in some cell

Also, a row-variable can be of class 'Surv'. Then the probability of 'event' at a fixed time (set up with 'timemax' argument) is computed and a logrank test is performed.

When there are more than 2 groups, it also performs pairwise comparisons adjusting for multiple testing (Tukey when row-variable is normal-distributed and Benjamini & Hochberg method otherwise), and computes p-value for trend. The p-value for trend is computed from the Pearson test when row-variable is normal and from the Spearman test when it is continuous non normal. If row-variable is of class 'Surv', the score test is computed from a Cox model where the grouping variable is introduced as an integer variable predictor. If the row-variable is categorical, the p-value for trend is computed as

$$1 - \text{pchisq}(\text{cor}(\text{as.integer}(x), \text{as.integer}(y))^2 * (\text{length}(x) - 1), 1)$$

where 'x' is the row-variable and 'y' is the grouping variable.

If there are two groups, the Odds Ratio is computed for each row-variable. While, if the response is of class 'Surv' (i.e. time to event) Hazard Ratios are computed.

The p-values for Hazard Ratios are computed using the logrank or Wald test under a Cox proportional hazard regression when row-variable is categorical or continuous, respectively.

See the vignette for more detailed examples illustrating the use of this function and the methods used.



**Value**

An object of class 'compareGroups'.

'print' returns a table sample size, overall p-values, type of variable ('categorical', 'normal', 'non-normal' or 'Surv') and the subset of individuals selected.

'summary' returns a much more detailed list. Every component of the list is the result for each row-variable, showing frequencies, mean, standard deviations, quartiles or K-M probabilities as appropriate. Also, it shows overall p-values as well as p-trends and pairwise p-values among the groups.

'plot' displays, for all the analyzed variables, normality plots (with the Shapiro-Wilks test), barplots or Kaplan-Meier plots depending on whether the row-variable is continuous, categorical or time-to-response, respectively. Also, bivariate plots can be displayed with stratified by groups boxplots or barplots, setting 'bivar' argument to TRUE.

An update method for 'compareGroups' objects has been implemented and works as usual to change all the arguments of previous analysis.

A subset, '[' , method has been implemented for 'compareGroups' objects. The subsetting indexes can be either integers (as usual), row-variables names or row-variable labels.

Combine by rows, 'rbind', method has been implemented for 'compareGroups' objects. It is useful to distinguish row-variable groups.

See examples for further illustration about all previous issues.

**Note**

Arguments 'X', 'y' and 'Xext' from the compareGroups.default method are not recommended to be used. Use 'X', 'data' and 'subset' arguments from the compareGroups.formula method instead.

By default, the labels of the variables (row-variables and grouping variable) are displayed in the resulting tables. These labels are taken from the "label" attribute of each variable. And if this attribute is NULL, then the name of the variable is displayed, instead. To label non-labeled variables, or to change their labels, use the function [label](#).

There may be no equivalence between the intervals of the OR / HR and p-values. For example, when the response variable is binary and the row-variable is continuous, p-value is based on Mann-Whitney U test or t-test depending on whether row-variable is normal distributed or not, respectively, while the confidence interval is build using the Wald method ( $\log(\text{OR}) \pm 1.96 \cdot \text{se}$ ). Or when the answer is of class 'Surv', p-value is computed with the logrank test, while confidence intervals are based on the Wald method ( $\log(\text{HR}) \pm 1.96 \cdot \text{se}$ ). Finally, when the response is binary and the row variable is categorical, the p-value is based on the chi-squared or Fisher test when appropriate, while confidence intervals are constructed from the median-unbiased estimation method (see

`oddsratio`).

Subjects selection criteria specified in `'selec'` and `'subset'` arguments are combined using `'&'` to be applied to every row-variable.

Through `'...'` argument of `'plot'` method, some parameters such as figure size, multiple figures in a unique file (only for `'pdf'` files), resolution, etc. are controlled. For more information about which arguments can be passed depending on the format type, see [pdf](#), [bmp](#), [jpeg](#), [png](#) or [tiff](#).

## References

Isaac Subirana, Hector Sanz, Joan Vila (2014). Building Bivariate Tables: The `compareGroups` Package for R. *Journal of Statistical Software*, 57(12), 1-16. URL <http://www.jstatsoft.org/v57/i12/>.

## See Also

[createTable](#)

## Examples

```
require(compareGroups)

# load REGICOR data
data(regicor)

# compute a time-to-cardiovascular event variable
regicor$tcv <- with(regicor, Surv(tocv, as.integer(cv=='Yes')))
Hmisc::label(regicor$tcv)<-"Cardiovascular"

# compute a time-to-overall death variable
regicor$death <- with(regicor, Surv(todeath, as.integer(death=='Yes')))
Hmisc::label(regicor$death) <- "Mortality"

# descriptives by sex
res <- compareGroups(sex ~ .-id-tocv-cv-todeath-death, data = regicor)
res

# summary of each variable
summary(res)

# univariate plots of all row-variables
## Not run:
plot(res)

## End(Not run)

# plot of all row-variables by sex
## Not run:
plot(res, bivar = TRUE)
```

```
## End(Not run)

# update changing the response: time-to-cardiovascular event.
# note that time-to-death must be removed since it is not possible
# not compute descriptives of a 'Surv' class object by another 'Surv' class object.

## Not run:
update(res, tcv ~ . + sex - tdeath - tcv)

## End(Not run)
```

---

compareSNPs

*Summarise genetic data by groups.*


---

## Description

This function provides an extensive summary range of your SNP data, allowing you to perform in-depth quality control of your genotyping results, and to explore your data before analysis. Summary measures include allele and genotype frequencies and counts, missingness rate, Hardy Weinberg equilibrium and more in the whole data set or stratified by other variables, such as case-control status. It can also test for differences in missingness between groups.

## Usage

```
compareSNPs(formula, data, subset, na.action = NULL, sep = "", verbose = FALSE, ...)
```

## Arguments

formula	an object of class "formula" (or one that can be coerced to that class). The right side of ~ must have the terms in an additive way, and these terms must refer to variables in 'data' must be of character or factor classes whose levels are the genotypes with the alleles written in their levels (e.g. A/A, A/T and T/T). The left side of ~ must contain the name of the grouping variable or can be left blank (in this case, summary data are provided for the whole sample, and no missingness test is performed).
data	an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If they are not found in 'data', the variables are taken from 'environment(formula)'.
subset	an optional vector specifying a subset of individuals to be used in the computation process (applied to all genetic variables).
na.action	a function which indicates what should happen when the data contain NAs. The default is NULL, and that is equivalent to <code>na.pass</code> , which means no action. Value <code>na.exclude</code> can be useful if it is desired to removed all individuals with some NA in any variable.

sep	character string indicating the separator between alleles (e.g. when using A/A, A/T and T/T genotype codification, 'sep' should be set to '/'. Default value is '' indicating that genotypes are coded as AA, AT and TT.
verbose	logical, print results from <a href="#">HWChisq</a> function. Default value is FALSE.
...	currently ignored.

### Value

An object of class 'compareSNPs' which is a data.frame (when no groups are specified on the left of the '~' in the 'formula' argument) or a list of data.frames, otherwise. Each data.frame contains the following fields:

- Ntotal: Total number of samples for which genotyping was attempted
- Ntyped: Number of genotypes called
- Typed.p: Percentage genotyped
- Miss.t: Number of missing genotypes
- Miss.p: Proportion of missing genotypes
- Minor: Minor Allele
- MAF: Minor allele frequency
- A1: Allele 1
- A2: Allele 2
- A1.ct: Count Allele 1
- A2.ct: Count Allele 2
- A1.p: Frequency of Allele 1
- A2.p: Frequency of Allele 2
- Hom1: Allele 1 Homozygote
- Het: Heterozygote
- Hom2: Allele 2 Homozygote
- Hom1.ct: Allele 1 Homozygote count
- Het.ct: Heterozygote Count
- Hom2.ct: Allele 2 Homozygote count
- Hom1.p: Frequency of Allele 1 Homozygote
- Het.p: Heterozygote frequency
- Hom2.p: Frequency of Allele 2 Homozygote
- HWE.p: Hardy-Weinberg equilibrium p-value

Additionally, when analysis is stratified by groups, the last component consists of a data.frame containing the p-values of missingness comparison among groups.

'print' returns a 'nice' format table for each group with the main results for each SNP (Ntotal, Ntyped, Minor, MAF, A1, A2, HWE.p), and the missingness test when group is considered.

### Note

It uses some functions taken from **SNPassoc** created by Juan Ramirez Gonzalez et al.

Hardy-Weinberg equilibrium test is performed using the [HWChisqMat](#)

**Author(s)**

Gavin Lucas (gavin.lucas<at>cleargenetics.com)

Isaac Subirana (isubirana<at>imim.es)

**See Also**

[createTable](#)

**Examples**

```
require(compareGroups)

# load example data (taken from SNPassoc)
data(SNPs)

# visualize first rows
head(SNPs)

# select casco and all SNPs
myDat <- SNPs[,c(2,6:40)]

# QC of three SNPs by groups of cases and controls
res<-compareSNPs(casco ~ .-casco, myDat)
res

# QC of three SNPs of the whole data set
res<-compareSNPs( ~ .-casco, myDat)
res
```

---

createTable

*Table of descriptives by groups: bivariate table*

---

**Description**

This functions builds a "compact" and "nice" table with the descriptives by groups.

**Usage**

```
createTable(x, hide = NA, digits = NA, type = NA, show.p.overall = TRUE,
            show.all, show.p.trend, show.p.mul = FALSE, show.n, show.ratio =
            FALSE, show.descr = TRUE, hide.no = NA, digits.ratio = NA,
            show.p.ratio = show.ratio, digits.p = 3, sd.type = 1, q.type = c(1, 1),
            extra.labels = NA)
## S3 method for class 'createTable'
```

```
print(x, which.table = "descr", nmax = TRUE, header.labels = c(), ...)
## S3 method for class 'createTable'
plot(x, ...)
```

### Arguments

x	an object of class 'compareGroups'
hide	a vector (or a list) with integers or characters with as many components as row-variables. If its length is 1 it is recycled for all row-variables. Each component specifies which category (the literal name of the category if it is a character, or the position if it is an integer) must be hidden and not shown. This argument only applies to categorical row-variables, and for continuous row-variables it is ignored. If NA, all categories are displayed. Or a named vector (or a named list) specifying which row-variables 'hide' is applied, and for the rest of row-variables default value is applied. Default value is NA.
digits	an integer vector with as many components as row-variables. If its length is 1 it is recycled for all row-variables. Each component specifies the number of significant decimals to be displayed. Or a named vector specifying which row-variables 'digits' is applied (a reserved name is '.else' which defines 'digits' for the rest of the variables); if no '.else' variable is defined, default value is applied for the rest of the variables. Default value is NA which puts the 'appropriate' number of decimals (see vignette for further details).
type	an integer that indicates whether absolute and/or relative frequencies are displayed: 1 - only relative frequencies; 2 or NA - absolute and relative frequencies in brackets; 3 - only absolute frequencies.
show.p.overall	logical indicating whether p-value of overall groups significance ('p.overall' column) is displayed or not. Default value is TRUE.
show.all	logical indicating whether the '[ALL]' column (all data without stratifying by groups) is displayed or not. Default value is FALSE if grouping variable is defined, and FALSE if there are no groups.
show.p.trend	logical indicating whether p-trend is displayed or not. It is always FALSE when there are less than 3 groups. If this argument is missing, there are more than 2 groups and the grouping variable is an ordered factor, then p-trend is displayed. By default, p-trend is not displayed, and it is displayed when there are more than 2 groups and the grouping variable is of class ordered-factor.
show.p.mul	logical indicating whether the pairwise (between groups) comparisons p-values are displayed or not. It is always FALSE when there are less than 3 groups. Default value is FALSE.
show.n	logical indicating whether number of individuals analyzed for each row-variable is displayed or not in the 'descr' table. Default value is FALSE and it is TRUE when there are no groups.
show.ratio	logical indicating whether OR / HR is displayed or not. Default value is FALSE.
show.descr	logical indicating whether descriptives (i.e. mean, proportions, ...) are displayed. Default value is TRUE.

hide.no	character specifying the name of the level to be hidden for all categorical variables with 2 categories. It is not case-sensitive. The result is one row for the variable with only the name displayed and not the category. This is especially useful for yes/no variables. It is ignored for the categorical row-variables with 'hide' argument different from NA. Default value is NA which means that no category is hidden.
digits.ratio	The same as 'digits' argument but applied for the Hazard Ratio or Odds Ratio.
show.p.ratio	logical indicating whether p-values corresponding to each Hazard Ratio / Odds Ratio are shown.
digits.p	integer indicating the number of decimals displayed for all p-values. Default value is 3.
sd.type	an integer that indicates how standard deviation is shown: 1 - mean (SD), 2 - mean ? SD.
q.type	a vector with two integer components. The first component refers to the type of brackets to be displayed for non-normal row-variables (1 - rounded and 2 - squared), while the second refers to the percentile separator (1 - ','; 2 - '.'; and 3 - '-'). Default value is c(1, 1).
extra.labels	character vector of 3 components corresponding to key legend to be appended to normal, non-normal and categorical row-variables labels. Default value is NA which appends no extra key. If it is set to c("", "", ""), "Mean (SD)", "Median [25th; 75th]" and "N (%)" are appended.
which.table	character indicating which table is printed. Possible values are 'descr', 'avail' or 'both' (partial matching allowed), printing descriptives by groups table, availability data table or both tables, respectively. Default value is 'descr'.
nmax	logical, indicating whether to show the number of subjects with at least one valid value across all row-variables. Default value is TRUE.
header.labels	a character named vector with 'all', 'p.overall', 'p.trend', 'ratio', 'p.ratio' and 'N' components indicating the label for '[ALL]', 'p.overall', 'p.trend', 'ratio', 'p.ratio' and 'N' (available data), respectively. Default is a zero length vector which makes no changes, i.e. '[ALL]', 'p.overall', 'p.trend', 'ratio', 'p.ratio' and 'N' labels appear for descriptives of entire cohort, global p-value, p-value for trend, HR/OR and p-value of each HR/OR and available data, respectively.
...	other arguments passed to <code>print.default</code> .

### Value

An object of class 'createTable', which contains a list of 2 matrix:

descr	a character matrix of descriptives for all row-variables by groups and p-values in a 'compact' format
avail	a character matrix indicating the number of available data for each group, the type of variable (categorical, continuous-normal or continuous-non-normal) and the individuals selection made (if non selection 'ALL' is displayed).

'print' prints these two tables in a 'nice' format.

'summary' prints the 'available' info table (it is a short form of `print(x, which.table = 'avail')`).

'update' modifies previous results from 'createTable'.

'plot' see the method in [compareGroups](#) function.

subsetting, '[', can also be applied to 'createTable' objects in the same way as 'compareGroups' objects.

combine by rows, 'rbind', method can be applied to 'createTable' objects, but only if all 'createTable' objects have the same columns. It is useful to distinguish row-variable groups. The resulting object is of class 'rbind.createTable' and 'createTable'.

combine by columns, 'cbind', method can be applied to 'createTable' objects, but only if all 'createTable' objects have the same rows. It may be used when combining different tables referring to different subsets of people (for example, men and women). The resulting object is of class 'cbind.createTable' and 'createTable' and has its own 'print' method.

See the vignette for more details.

### Note

The way to compute the 'N' shown in the bivariate table header, controlled by 'nmax' argument, has been changed from previous versions (<1.3). In the older versions 'N' was computed as the maximum across the cells withing each column (group) from the 'available data' table ('avail').

The p-values corresponding to the OR of a two level row-variable may not me equal to its p.overall p-value. This is because statistical tests are different: the option 'midp.exact' (see [oddsratio](#) for more details) is taken in the first case and Chi-square or Fisher exact test in the second. The same happens when OR for a continuous value is performed: the p-value corresponding to this OR is computed form a logistic regression and therefore may differ from the one computed using a Student-T test or Kruskall Wallis test. This discordance may also be present when computing the p-value corresponding to a Hazard Ratio for a categorical two level row-variable: a Wald test or a long-rank test are performed.

### References

Isaac Subirana, Hector Sanz, Joan Vila (2014). Building Bivariate Tables: The compareGroups Package for R. Journal of Statistical Software, 57(12), 1-16. URL <http://www.jstatsoft.org/v57/i12/>.

### See Also

[compareGroups](#), [export2latex](#), [export2csv](#), [export2html](#)

### Examples

```
require(compareGroups)

# load REGICOR data
data(regicor)

# compute a time-to-cardiovascular event variable
regicor$tcv <- with(regicor, Surv(tocv, as.integer(cv=='Yes')))
Hmisc::label(regicor$tcv) <- "Cardiovascular incidence"
```



```

# remove variables used to create time-to variables
regicor <- remove.vars(regicor,c("todeath","death","tocv","cv"))

# descriptives by time-to-cardiovascular event, taking 'no' category as
# the reference in computing HRs.
res <- compareGroups(tcv ~ age + sex + smoker + sbp + histhtn +
                    chol + txchol + bmi + phyact + pcs + tcv, regicor, ref.no='no')

# build table showing HR and hiding the 'no' category
restab <- createTable(res, show.ratio = TRUE, hide.no = 'no')
restab

# prints available info table
summary(restab)

# more...

## Not run:

# Adds the 'available data' column
update(restab, show.n=TRUE)

# Descriptive of the entire cohort
update(restab, x = update(res, ~ . ))

# .. changing the response variable to sex
# Odds Ratios (OR) are displayed instead of Hazard Ratios (HR).
# note that now it is possible to compute descriptives by time-to-death
# or time-to-cv but not the ORs .
# We set timemax to 5 years, to report the probability of death and CV at 5 years:
update(restab, x = update(res, sex ~ . - sex + tdeath + tcv, timemax = 5*365.25))

## Combining tables:

# a) By rows: takes the first four variables as a group and the rest as another group:
rbind("First group of variables"=restab[1:4],"Second group of variables"=
      restab[5:length(res)])

# b) By columns: puts stratified tables by sex one beside the other:
res1<-compareGroups(year ~ . - id - sex, regicor)
restab1<-createTable(res1, hide.no = 'no')
restab2<-update(restab1, x = update(res1, subset = sex == 'Male'))
restab3<-update(restab1, x = update(res1, subset = sex == 'Female'))
cbind("ALL" = restab1, "MALES" = restab2, "FEMALES" = restab3)

## End(Not run)

```

---

export2csv	<i>Exporting descriptives table to plain text (CSV) format</i>
------------	--

---

### Description

This function takes the result of `createTable` and exports the tables to plain text (CSV) format.

### Usage

```
export2csv(x, file, which.table="descr", sep=",", nmax = TRUE, header.labels = c(), ...)
```

### Arguments

x	an object of class 'createTable'.
file	file where table in CSV format will be written. Also, another file with the extension '_appendix' is written with the available data table.
which.table	character indicating which table is printed. Possible values are 'descr', 'avail' or 'both' (partial matching allowed), exporting descriptives by groups table, available data table or both tables, respectively. Default value is 'descr'.
sep	character. The variable separator, same as 'sep' argument from <code>write.table</code> . Default value is ','.
nmax	logical, indicating whether to show the number of subjects with at least one valid value across all row-variables. Default value is TRUE.
header.labels	see the 'header.labels' argument from <code>createTable</code> .
...	other arguments passed to <code>write.table</code> .

### Note

The way to compute the 'N' shown in the bivariate table header, controlled by 'nmax' argument, has been changed from previous versions (<1.3). In the older versions 'N' was computed as the maximum across the cells withing each column (group) from the 'available data' table ('avail').

### See Also

[createTable](#), [export2latex](#), [export2pdf](#), [export2html](#), [export2md](#), [export2word](#)

### Examples

```
## Not run:
require(compareGroups)
data(regicor)
res <- compareGroups(sex ~. -id-todeath-death-tocv-cv, regicor)
export2csv(createTable(res, hide.no = 'n'), file=tempfile(fileext=".csv"))

## End(Not run)
```

---

`export2html`*Exporting descriptives table to HTML format*

---

**Description**

This function takes the result of `createTable` and exports the tables to HTML format.

**Usage**

```
export2html(x, file, which.table="descr", nmax = TRUE, header.labels = c(), ...)
```

**Arguments**

<code>x</code>	an object of class 'createTable'.
<code>file</code>	file where table in HTML format will be written. Also, another file with the extension '_appendix' is written with the available data table. If missing, the HTML code is returned.
<code>which.table</code>	character indicating which table is printed. Possible values are 'descr', 'avail' or 'both' (partial matching allowed), exporting descriptives by groups table, availability data table or both tables, respectively. Default value is 'descr'.
<code>nmax</code>	logical, indicating whether to show the number of subjects with at least one valid value across all row-variables. Default value is TRUE.
<code>header.labels</code>	see the 'header.labels' argument from <a href="#">createTable</a> .
<code>...</code>	currently ignored.

**Note**

The way to compute the 'N' shown in the bivariate table header, controlled by 'nmax' argument, has been changed from previous versions (<1.3). In the older versions 'N' was computed as the maximum across the cells withing each column (group) from the 'available data' table ('avail').

**See Also**

[createTable](#), [export2latex](#), [export2pdf](#), [export2csv](#), [export2md](#), [export2word](#)

**Examples**

```
## Not run:  
require(compareGroups)  
data(regicor)  
res <- compareGroups(sex ~. -id-todeath-death-tocv-cv, regicor)  
export2html(createTable(res, hide.no = 'n'), file=tempfile(fileext=".html"))  
  
## End(Not run)
```

---

 export2latex

*Exporting descriptives table to LaTeX format*


---

## Description

This function takes the result of `createTable` and exports the tables to LaTeX format.

## Usage

```
export2latex(x, ...)
## S3 method for class 'createTable'
export2latex(x, file, which.table = 'descr', size = 'same',
             nmax = TRUE, header.labels = c(), caption = NULL, loc.caption = 'top', label = NULL,
             landscape = NA, colmax = 10, ...)
## S3 method for class 'cbind.createTable'
export2latex(x, file, which.table = 'descr', size = 'same',
             nmax = TRUE, header.labels = c(), caption = NULL, loc.caption = 'top', label = NULL,
             landscape = NA, colmax = 10, ...)
```

## Arguments

<code>x</code>	an object of class 'createTable'.
<code>file</code>	Name of file where the resulting code should be saved. If file is missing, output is displayed on screen. Also, another file with the extension '_appendix' is written with the available data table.
<code>which.table</code>	character indicating which table is exported. Possible values are 'descr', 'avail' or 'both' (partial matching allowed), exporting descriptives by groups table, availability data table or both tables, respectively. Default value is 'descr'.
<code>size</code>	character indicating the size of the table elements. Possible values are: 'tiny', 'scriptsize', 'footnotesize', 'small', 'normalsize', 'large', 'Large', 'LARGE', 'huge', 'Huge' or 'same' (partial matching allowed). Default value is 'same' which means that font size of the table is the same as specified in the main LaTeX document.
<code>nmax</code>	logical, indicating whether to show the number of subjects with at least one valid value across all row-variables. Default value is TRUE.
<code>header.labels</code>	see the 'header.labels' argument from <a href="#">createTable</a> .
<code>caption</code>	character specifying the table caption for descriptives and available data table. If <code>which.table='both'</code> the first element of 'caption' will be assigned to descriptives table and the second to available data table. If it is set to "", no caption is inserted. Default value is NULL, which writes 'Summary descriptives table by groups of 'y'' for descriptives table and 'Available data by groups of 'y'' for the available data table.
<code>label</code>	character specifying the table label for descriptives and available data table. This may be useful to cite the tables elsewhere in the LaTeX document. If <code>which.table='both'</code> the first element of 'label' will be assigned to descriptives

	table and the second to available data table. Default value is NULL, which assigns no label to the table/s.
loc.caption	character specifying the table caption location. Possible values are 'top' or 'bottom' (partial matching allowed). Default value is 'top'.
landscape	logical indicating whether the table must be placed in landscape, or NA that places the table in landscape when there are more than 'colmax' columns. Default value is NA.
colmax	integer indicating the maximum number of columns to make the table not to be placed in landscape. This argument is only applied when 'landscape' argument is NA. Default value is 10.
...	currently ignored.

### Value

List of two possible components corresponding to the code of 'descr' table and 'avail' table. Each component of the list is a character corresponding to the LaTeX code of these tables which can be helpful for post-processing.

### Note

The table is created in LaTeX language using the longtable environment. Therefore, it is necessary to type `\includepackage{longtable}` in the preamble of the LaTeX main document where the table code is inserted. Also, it is necessary to include the 'multirow' LaTeX package. \

The way to compute the 'N' shown in the bivariate table header, controlled by 'nmax' argument, has been changed from previous versions (<1.3). In the older versions 'N' was computed as the maximum across the cells within each column (group) from the 'available data' table ('avail'). \

When 'landscape' argument is TRUE or there are more than 'colmax' columns and 'landscape' is set to NA, LaTeX package 'lscap' must be loaded in the tex document.

### See Also

[createTable](#), [export2csv](#), [export2html](#), [export2pdf](#), [export2md](#), [export2word](#)

### Examples

```
## Not run:
require(compareGroups)
data(regicor)
res <- compareGroups(sex ~. -id-todeath-death-tocv-cv, regicor)
export2latex(createTable(res, hide.no = 'n'), file=tempfile(fileext=".tex"))

## End(Not run)
```

---

 export2md

*Exporting descriptives table to Markdown format*


---

### Description

This function takes the result of `createTable` and exports the tables to markdown format. It may be useful when inserting R code chunks in a Markdown file (.Rmd).

### Usage

```
export2md(x, which.table="descr", nmax = TRUE, header.labels = c(),
          caption = NULL, ...)
```

### Arguments

<code>x</code>	an object of class 'createTable'.
<code>which.table</code>	character indicating which table is printed. Possible values are 'descr' or 'avail' (partial matching allowed), exporting descriptives by groups table or availability data table, respectively. Default value is 'descr'.
<code>nmax</code>	logical, indicating whether to show the number of subjects with at least one valid value across all row-variables. Default value is TRUE.
<code>header.labels</code>	see the 'header.labels' argument from <a href="#">createTable</a> .
<code>caption</code>	character specifying the table caption for descriptives and available data table. If <code>which.table='both'</code> the first element of 'caption' will be assigned to descriptives table and the second to available data table. If it is set to "", no caption is inserted. Default value is NULL, which writes 'Summary descriptives table by groups of 'y'' for descriptives table and 'Available data by groups of 'y'' for the available data table.
<code>...</code>	currently ignored.

### Value

It does not return anything, but the Markdown code to generate the descriptive or available table is printed.

### Note

The way to compute the 'N' shown in the bivariate table header, controlled by 'nmax' argument, has been changed from previous versions (<1.3). In the older versions 'N' was computed as the maximum across the cells withing each column (group) from the 'available data' table ('avail').

### See Also

[createTable](#), [export2latex](#), [export2pdf](#), [export2csv](#), [export2html](#), [export2word](#)

## Examples

```
## Not run:
require(compareGroups)
data(regicor)
res <- compareGroups(sex ~. -id-todeath-death-tocv-cv, regicor)
export2md(createTable(res, hide.no = 'no'))

## End(Not run)
```

---

export2pdf	<i>Exports tables to PDF files.</i>
------------	-------------------------------------

---

## Description

This function creates automatically a PDF with the table. Also, the LaTeX code is stored in the specified file.

## Usage

```
export2pdf(x, file, compile = TRUE, openfile = TRUE, margin = c(2,2,1,1), ...)
```

## Arguments

x	an object of class 'createTable' or that inherits it.
file	character specifying the PDF file resulting after compiling the LaTeX code corresponding to the table specified in the 'x' argument. LaTeX code is also stored in the same folder with the same name but .tex extension. When 'compile' argument is FALSE, only .tex file is saved.
compile	logical indicating whether tex file is compiled using <a href="#">texi2pdf</a> function. Default value is TRUE.
openfile	logical indicating whether to open the compiled pdf file or not. It uses the <a href="#">sys</a> function and does not apply when 'compile' argument is FALSE. Default value is TRUE.
margin	an integer vector specifying the margins in cm. Its components refer to 'top', 'bottom', 'left' and 'right' sides. Default values are 2, 2, 1 and 1, respectively.
...	Arguments passed to <a href="#">export2latex</a> .

## Note

To make the .tex file be compiled, some LaTeX compiler such as Miktex must be installed. Also, the tex file must include the following LaTeX packages:

- babel

- longtable
- multirow
- lscape
- geometry
- inputenc

### See Also

[createTable](#), [export2latex](#), [export2csv](#), [export2html](#), [export2md](#), [export2word](#)

### Examples

```
## Not run:

require(compareGroups)
data(regicor)

# example on an ordinary table
res <- createTable(compareGroups(year ~ . -id, regicor), hide = c(sex=1), hide.no = 'no')
export2pdf(res, file=tempfile(fileext=".pdf"), size="small")

## End(Not run)
```

---

export2word

*Exports tables to Word files.*

---

### Description

This function creates automatically a Word file with the table.

### Usage

```
export2word(x, file, which.table="descr", nmax = TRUE, header.labels = c(), ...)
```



**Arguments**

x	an object of class 'createTable' or that inherits it.
file	character specifying the word file (.doc or .docx) resulting after compiling the Markdown code corresponding to the table specified in the 'x' argument.
which.table	see <a href="#">export2md</a>
nmax	see <a href="#">print.createTable</a>
header.labels	see <a href="#">print.createTable</a>
...	currently ignored

**Note**

Word file is created after compiling Markdown code created by [export2md](#). To compile it it calls [render](#) function which requires pandoc to be installed.

**See Also**

[createTable](#), [export2latex](#), [export2pdf](#), [export2csv](#), [export2html](#), [export2md](#)

**Examples**

```
## Not run:

require(compareGroups)
data(regicor)

# example on an ordinary table
res <- createTable(compareGroups(year ~ . -id, regicor), hide = c(sex=1), hide.no = 'no')
export2word(res, file = tempfile(fileext=".docx"))

## End(Not run)
```

---

export2xls

*Exporting descriptives table to Exel format (.xlsx or .xls)*

---

**Description**

This function takes the result of `createTable` and exports the tables to Excel format (.xlsx or .xls).

**Usage**

```
export2xls(x, file, which.table="descr", nmax=TRUE, header.labels=c(), ...)
```

**Arguments**

<code>x</code>	an object of class 'createTable'.
<code>file</code>	file where table in HTML format will be written. Also, another file with the extension '_appendix' is written with the available data table. If missing, the HTML code is returned.
<code>which.table</code>	character indicating which table is printed. Possible values are 'descr', 'avail' or 'both' (partial matching allowed), exporting descriptives by groups table, availability data table or both tables, respectively. Default value is 'descr'.
<code>nmax</code>	logical, indicating whether to show the number of subjects with at least one valid value across all row-variables. Default value is TRUE.
<code>header.labels</code>	see the 'header.labels' argument from <a href="#">createTable</a> .
<code>...</code>	arguments passed to <a href="#">write.xlsx</a> function.

**Note**

The way to compute the 'N' shown in the bivariate table header, controlled by 'nmax' argument, has been changed from previous versions (<1.3). In the older versions 'N' was computed as the maximum across the cells withing each column (group) from the 'available data' table ('avail').

To use this function you must have the **xlsx** package installed, which depends on **rjava** package.

**See Also**

[createTable](#), [export2latex](#), [export2pdf](#), [export2csv](#), [export2md](#), [export2word](#)

**Examples**

```
## Not run:
require(compareGroups)
data(regicor)
res <- compareGroups(sex ~. -id-todeath-death-tocv-cv, regicor)
export2xls(createTable(res, hide.no = 'n'), file=tempfile(fileext=".xlsx"))

## End(Not run)
```

---

getResults

*Easily retrieve summary data as R-objects (matrices and vectors).*

---

**Description**

This functions exccratcs specific results (descriptives, p-values, Odds-Ratios / Hazard-Ratios, ...) from a compareGroups object as matrix or vectors.

**Usage**

```
getResults(obj, what = "descr")
```

**Arguments**

**obj** an object of class 'compareGroups' or 'createTable'  
**what** character indicating which results are to be retrieved: descriptives, p-value, p-trend, pairwise p-values, or Odds-Ratios / Hazard-Ratios. Possible values are: "descr", "p.overall", "p.trend", "p.mul" and "ratio". Default value is "descr".

**Value**

**what = "descr"** An array or matrix with as many columns as variables/categories and seven columns indicating all possible descriptive statistics (mean, sd, median, Q1, Q3, absolute and relative frequencies). When different groups are analysed, the 3rd dimension of the array corresponds to the groups. Otherwise, the result will be a matrix with no 3rd dimension.

**what = "p.overall"**  
A vector whose elements are the p-value for each analysed variable.

**what = "p.trend"**  
A vector whose elements are the p-trend for each analysed variable.

**what = "p.mul"** A matrix with pairwise p-values where rows correspond to the analysed variables and columns to each pair of groups.

**what = "ratio"** A matrix with as many rows as variables/categories and 4 columns corresponding to the OR/HR, confidence interval and p-value.

**Note**

For descriptives, NA is placed for descriptives not appropriate for the variable. For example columns corresponding to frequencies for continuous variables will be NA.

**See Also**

[compareGroups](#), [createTable](#)

**Examples**

```
require(compareGroups)
data(regicor)
res<-compareGroups(sex ~ . , regicor, method=c(triglyc=2))
# retrieve descriptives
getResults(res)
# retrieve OR and their corresponding p-values
getResults(res, what="ratio")
```

---

missingTable	<i>Table of missingness counts by groups.</i>
--------------	---

---

**Description**

This functions returns a table with the non-available frequencies from a already build bivariate table.

**Usage**

```
missingTable(obj,...)
```

**Arguments**

obj	either a 'compareGroups' or 'createTable' object.
...	other arguments passed to createTable.

**Value**

An object of class 'createTable'. For further details, see 'value' section of createTable help file.

**Note**

This function returns an object of class 'createTable', and therefore all methods implemented for 'createTable' objects can be applied, except the 'update' method.

All arguments of createTable can be passed throught '...' argument, except 'hide.no' argument which is fixed inside the code and cannot be changed.

This function cannot be applied to stratified tables, i.e. 'rbind.createTable' and 'cbind.createTable'. If stratified missingness table is desired, apply this function first to each table and then use cbind.createTable or/and rbind.createTable functions to combine exactly in the same way as 'createTable' objects. See 'example' section below.

**See Also**

[createTable](#)

**Examples**

```
## Not run:  
  
require(compareGroups)  
  
# load regicor data  
data(regicor)  
  
# table of descriptives by recruitment year  
res <- compareGroups(year ~ age + sex + smoker + sbp + histhtn +  
  chol + txchol + bmi + phyact + pcs + death, regicor)
```

```

restab <- createTable(res, hide.no = "no")

# missingness table
missingTable(restab,type=1)

# also create the missing table from a compareGroups object
miss <- missingTable(res)
miss

# some methods that works for createTable objects also works for objects
# computed by missTable function.
miss[1:4]
varinfo(miss)
plot(miss)

#... but update methods cannot be applied (this returns an error).
update(miss,type=2)

## End(Not run)

```

---

predimed

*PREDIMED randomized trial*

---

## Description

The PREDIMED trial (Prevención con Dieta Mediterránea) is a randomized, parallel and multi-centric cohort with more than 7,000 participants who were randomly assigned to three diet groups (olive oil + mediterranean diet, nuts + mediterranean diet, and low-fat diet -control group-) and followed-up during more than 7 years.

## Usage

```
data(predimed)
```

## Format

A data frame with 6324 observations on the following 15 variables:

group a factor with levels Control MedDiet + Nuts MedDiet + V00. Intervention group.

sex a factor with levels Male Female.Sex.

age Age.

smoke a factor with levels Never Current Former.Smoking.

bmi Body mass index.

waist Waist circumference.

wth Waist-to-height ratio.

htn a factor with levels No Yes.Hypertension.

diab a factor with levels No Yes .Type-2 diabetes.

hyperchol a factor with levels No Yes.Dyslipidemia.

famhist a factor with levels No Yes.Family history of premature CHD.

hormo a factor with levels No Yes.Hormone-replacement therapy.

p14 MeDiet Adherence score.

toevent follow-up to main event (years).

event a factor with levels No Yes.AMI, stroke, or CV Death.

## Details

Participants of PREDIMED study were recruited in 2003 and were revisited annually until 2010. In each visit, a large variety of different information was collected: antropometric (body mass index, age, ...), blood samples (Total cholesterol, HDL cholesterol, ...), dietary pattern consumption using validated questionnaires, treatments, etc. Also, cardiovascular events were recorded during the follow-up period. At recruitment period, each participant was assigned to each of the three diet type group. Periodically (every 6-months approximately), participants were advised to follow the diet corresponding to their group. To build this example data set approx 85% of original participants were randomly selected. For more information about this data set, see the vignette.

## Source

For more information about the study these data come from, visit <http://predimed.onmedic.net/>.

## References

Estruch R, Ros E, Salas-Salvadó J, Covas MI, Corella D, Arós F, Gómez-Gracia E, Ruiz-Gutiérrez V, Fiol M, Lapetra J, Lamuela-Raventos RM, Serra-Majem L, Pintó X, Basora J, Muñoz MA, Sorlí JV, Martínez JA, Martínez-González MA; PREDIMED Study Investigators (2013). Primary prevention of cardiovascular disease with a Mediterranean diet. *New England Journal of Medicine* 368(14): 1279-1290.

## Examples

```
require(compareGroups)
data(predimed)
summary(predimed)
```

---

printTable	<i>'Nice' table format.</i>
------------	-----------------------------

---

**Description**

This functions prints a table on the console in a 'nice' format.

**Usage**

```
printTable(obj, row.names = TRUE, justify = 'right')
```

**Arguments**

obj	an object of class 'data.frame' or 'matrix'. It must be at least two columns, the first columns is considered as the 'row.names' and is left justified (if the 'row.names' argument is set to TRUE), while the rest of the columns are right justified.
row.names	logical indicating whether the first column or variable is treated as a 'row.names' column and must be left-justified. Default value is TRUE.
justify	character as 'justify' argument from <a href="#">format</a> function. It applies to all columns of the data.frame or matrix when 'row.names' argument is FALSE or all columns excluding the first one otherwise. Default value is 'right'.

**Value**

No object is returned.

**Note**

This function may be usefull when printing a table with some results with variables as the first column and a header. It adds 'nice' lines to highlight the header and also the bottom of the table.

It has been used to print 'compareSNPs' objects.

**See Also**

[compareSNPs](#)

**Examples**

```
require(compareGroups)

data(regicor)

# example of the coefficients table from a linear regression
model <- lm(chol ~ age + sex + bmi, regicor)
results <- coef(summary(model))
results <- cbind(Var = rownames(results), round(results, 4))
```

```
printTable(results)

# or visualize the first rows of the iris data frame.
# In this example, the first column is not treated as a row.names column and it is right justified.
printTable(head(iris), FALSE)

# the same example with columns centered
printTable(head(iris), FALSE, 'centre')
```

---

radiograph

*Lists the values in the data set.*

---

### Description

This function creates a report of raw data in your data set. For each variable an ordered list of the unique entries (read as strings), useful for checking for input errors.

### Usage

```
radiograph(file, header = TRUE, save=FALSE, out.file="", ...)
```

### Arguments

file	character specifying the file where the data set is located.
header	see <a href="#">read.table</a> .
save	logical indicating whether output should be stored in a file (TRUE) or printed on the console (FALSE). Default is FALSE.
out.file	character specifying the file where the results are to be output. It only applies when 'save' argument is set to TRUE.
...	Arguments passed to <a href="#">read.table</a> .

### Author(s)

Gavin Lucas (gavin.lucas<at>cleargenetics.com)

Isaac Subirana (isubirana<at>imim.es)

### See Also

[report](#)



**Examples**

```
## Not run:

require(compareGroups)

# read example data of regicor in plain text format with variables separated by '\t'.
datafile <- system.file("exdata/regicor.txt", package="compareGroups")
radiograph(datafile)

## End(Not run)
```

---

regicor	<i>REGICOR cross-sectional data</i>
---------	-------------------------------------

---

**Description**

These data come from 3 different cross-sectional surveys of individuals representative of the population from a north-west Spanish province (Girona), REGICOR study.

**Usage**

```
data(regicor)
```

**Format**

A data frame with 2294 observations on the following 21 variables:

id Individual id  
year a factor with levels 1995 2000 2005. Recruitment year  
age Patient age at recruitment date  
sex a factor with levels male female. Sex  
smoker a factor with levels Never smoker Current or former < 1y Never or former >= 1y.  
Smoking status  
sbp Systolic blood pressure  
dbp Diastolic blood pressure  
histhtn a factor with levels Yes No. History of hypertension  
txhtn a factor with levels No Yes. Hypertension (HTN) treatment  
chol Total cholesterol (mg/dl)  
hdl HDL cholesterol (mg/dl)  
triglyc Triglycerides (mg/dl)  
ldl LDL cholesterol (mg/dl)

histchol a factor with levels Yes No. History of hypercholesterolemia

txchol a factor with levels No Yes. Cholesterol treatment

height Height (cm)

weight Weight (Kg)

bmi Body mass index

phyact Physical activity (Kcal/week)

pcs Physical component summary

mcs Mental component summary

death a factor with levels No Yes. Overall death

todeath Days to overall death or end of follow-up

cv a factor with levels No Yes. Cardiovascular event

tocv Days to cardiovascular event or end of follow-up

## Details

The variables collected in the REGICOR study were mainly cardiovascular risk factors (hundreds of variables were collected in the different questionnaires and blood measurements), but the variables present in this data set are just a few of them. Also, for reasons of confidentiality, the individuals in this data set are a 30% approx. random subsample of the original one.

Each variable of this data.frame contains label describing them in the attribute "label".

For more information, see the vignette.

## Note

Variables death, todeath, cv, tocv are not real but they have been simulated at random to complete the data example with some time-to-event variables.

## Source

For reasons of confidentiality, the whole data set is not publicly available. For more information about the study these data come from, visit [www.regicor.org](http://www.regicor.org).

## Examples

```
require(compareGroups)
data(regicor)
summary(regicor)
```

---

report	<i>Report of descriptive tables and plots.</i>
--------	--

---

### Description

This function creates automatically a PDF with the descriptive table as well as availability data and all plots. This file is structured and indexed in the way that the user can navigate through all tables and figures along the document.

### Usage

```
report(x, file, fig.folder, compile = TRUE, openfile = TRUE, title = "Report",
      author, date, ...)
```

### Arguments

x	an object of class 'createTable'.
file	character specifying the PDF file resulting after compiling the LaTeX code of report. LaTeX code is also stored in the same folder with the same name but .tex extension. When 'compile' argument is FALSE, only .tex file is saved.
fig.folder	character specifying the folder where the plots corresponding to all row-variables of the table are placed. If it is left missing, a folder with the name file_figures is created in the same folder of 'file'.
compile	logical indicating whether tex file is compiled using <a href="#">texi2pdf</a> function. Default value is TRUE.
openfile	logical indicating whether to open the compiled pdf file or not. It uses the <a href="#">sys</a> function and does not apply when 'compile' argument is FALSE. Default value is TRUE.
title	character specifying the title of the report on the cover page. Default value is 'Report'.
author	character specifying the author/s name/s of the report on the cover page. When missing, no authors appear.
date	character specifying the date of the report on the cover page. When missing, the present date appears.
...	Arguments passed to <a href="#">export2latex</a> .

### Note

This functions does not work with stratified tables ('cbind.createTable' class objects). To report this class of tables you can report each of its component (see second example from 'examples' section).

In order to compile the tex file the following packages must be available:

- babel
- longtable
- hyperref

- multirow
- lscape
- geometry
- float
- inputenc
- epsfig

### See Also

[createTable](#), [export2latex](#), [export2csv](#), [export2html](#), [radiograph](#)

### Examples

```
## Not run:

require(compareGroups)
data(regicor)

# example on an ordinary table
res <- createTable(compareGroups(year ~ . -id, regicor), hide = c(sex=1), hide.no = 'no')
report(res, "report.pdf", size="small", title="\Huge \textbf{REGICOR study}",
        author="Isaac Subirana \ IMIM-Parc de Salut Mar")

# example on an stratified table by sex
res.men <- createTable(compareGroups(year ~ . -id-sex, regicor, subset=sex=='Male'),
                       hide.no = 'no')
res.wom <- createTable(compareGroups(year ~ . -id-sex, regicor, subset=sex=='Female'),
                       hide.no = 'no')
res <- cbind("Men"=res.men, "Wom"=res.wom)
report(res[[1]], "reportmen.pdf", size="small",
        title="\Huge \textbf{REGICOR study \ Men}", date="") # report for men / no date
report(res[[2]], "reportwom.pdf", size="small",
        title="\Huge \textbf{REGICOR study \ Women}", date="") # report for wom / no date

## End(Not run)
```

---

varinfo

*Variable names and labels extraction*

---

### Description

This functions builds and prints a table with the variable names and their labels.

**Usage**

```
varinfo(x, ...)  
## S3 method for class 'compareGroups'  
varinfo(x, ...)  
## S3 method for class 'createTable'  
varinfo(x, ...)
```

**Arguments**

x                    an object of class 'compareGroups' or 'createTable'  
...                  other arguments currently ignored

**Details**

By default, a compareGroup descriptives table lists variables by label (if one exists) rather than by name. If researchers have assigned detailed labels to their variables, this function is very useful to quickly locate the original variable name if some modification is required. This function simply lists all "Analyzed variable names" by "Orig varname" (i.e. variable name in the data.frame) and "Shown varname" (i.e., label).

**Value**

A 'matrix' with two columns

Orig varname    actual variable name in the 'data.frame' or in the 'parent environment'.

Shown varname   names of the variable shown in the resulting tables.

**Note**

If a variable has no "label" attribute, then the 'original varname' is the same as the 'shown varname'. The first variable in the table corresponds to the grouping variable. To label non-labeled variables or to change the label, use the function [label](#).

**See Also**

[compareGroups](#), [createTable](#)

**Examples**

```
require(compareGroups)  
data(regicor)  
res<-compareGroups(sex ~ . ,regicor)  
#createTable(res, hide.no = 'no')  
varinfo(res)
```

# Index

- \*Topic **datasets**
  - predimed, [29](#)
  - regicor, [33](#)
- \*Topic **misc**
  - compareGroups, [5](#)
  - compareSNPs, [11](#)
  - createTable, [13](#)
- \*Topic **package**
  - compareGroups-package, [2](#)
- \*Topic **utilities**
  - cGroupsGUI, [3](#)
  - cGroupsWUI, [4](#)
  - export2csv, [18](#)
  - export2html, [19](#)
  - export2latex, [20](#)
  - export2md, [22](#)
  - export2pdf, [23](#)
  - export2word, [24](#)
  - export2xls, [25](#)
  - getResults, [26](#)
  - missingTable, [28](#)
  - printTable, [31](#)
  - radiograph, [32](#)
  - report, [35](#)
  - varinfo, [36](#)
- .GlobalEnv, [5](#)
- bmp, [8](#), [10](#)
- cGroupsGUI, [3](#), [3](#), [4](#)
- cGroupsWUI, [3](#), [4](#), [4](#)
- chisq.test, [8](#)
- compareGroups, [3](#), [4](#), [5](#), [16](#), [27](#), [37](#)
- compareGroups-package, [2](#)
- compareSNPs, [3](#), [11](#), [31](#)
- createTable, [3](#), [4](#), [10](#), [13](#), [13](#), [18–22](#), [24–28](#), [36](#), [37](#)
- export2csv, [3](#), [16](#), [18](#), [19](#), [21](#), [22](#), [24–26](#), [36](#)
- export2html, [3](#), [16](#), [18](#), [19](#), [21](#), [22](#), [24](#), [25](#), [36](#)
- export2latex, [3](#), [16](#), [18](#), [19](#), [20](#), [22–26](#), [35](#), [36](#)
- export2md, [3](#), [18](#), [19](#), [21](#), [22](#), [24–26](#)
- export2pdf, [3](#), [18](#), [19](#), [21](#), [22](#), [23](#), [25](#), [26](#)
- export2word, [3](#), [18](#), [19](#), [21](#), [22](#), [24](#), [24](#), [26](#)
- export2xls, [3](#), [25](#)
- fisher.test, [8](#)
- format, [31](#)
- getResults, [26](#)
- hist, [8](#)
- HWChisq, [12](#)
- HWChisqMat, [12](#)
- jpeg, [8](#), [10](#)
- label, [9](#), [37](#)
- missingTable, [3](#), [28](#)
- na.exclude, [6](#), [11](#)
- na.pass, [6](#), [11](#)
- oddsratio, [7](#), [10](#), [16](#)
- pdf, [8](#), [10](#)
- plot.compareGroups (compareGroups), [5](#)
- plot.createTable (createTable), [13](#)
- png, [8](#), [10](#)
- predimed, [29](#)
- print.compareGroups (compareGroups), [5](#)
- print.compareSNPs (compareSNPs), [11](#)
- print.createTable, [25](#)
- print.createTable (createTable), [13](#)
- print.default, [15](#)
- print.summary.compareGroups (compareGroups), [5](#)
- print.summary.createTable (createTable), [13](#)
- printTable, [31](#)

radiograph, [3](#), [32](#), [36](#)  
read.table, [32](#)  
regicor, [33](#)  
render, [25](#)  
report, [3](#), [32](#), [35](#)  
runApp, [4](#)

shapiro.test, [6](#)  
summary.compareGroups (compareGroups), [5](#)  
summary.createTable (createTable), [13](#)  
sys, [23](#), [35](#)

texi2pdf, [23](#), [35](#)  
tiff, [8](#), [10](#)

update.compareGroups (compareGroups), [5](#)

varinfo, [36](#)

write.table, [18](#)  
write.xlsx, [26](#)