

# Package ‘rgrs’

January 27, 2010

**Type** Package

**Title** Functions to make R usage in social sciences easier (in french)

**Version** 0.2-15

**Date** 2009-12-14

**Author** Julien Barnier <julien.barnier@ens-lyon.fr>

**Maintainer** Julien Barnier <julien.barnier@ens-lyon.fr>

**Description** This package provides functions for beginners and social sciences students or researchers. Currently it includes functions for cross-tabulation, weighting, results export, and maps plotting. The documentation and help pages are written in french.

**License** GPL (>= 2)

**Encoding** UTF-8

**Depends**

**Suggests** tcltk, odfWeave, R2HTML, RColorBrewer, sp

**SystemRequirements** xclip (Linux)

**URL** <http://alea.fr.eu.org/j/rgrs.html>

**Repository** CRAN

**Repository/R-Forge/Project** rgrs

**Repository/R-Forge/Revision** 55

**Date/Publication** 2010-01-27 14:18:04

**R topics documented:**

Cartographie	2
Cartographie.legende	5
copie	6
copie.proptab	7
cramer.v	8
format.proptab	9
freq	10
genere.tableau	11
hdv2003	12
lyon	12
mls.export	13
mls.import	13
Pondérations	14
print.proptab	15
Profils	16
quant.cut	17
Questions multiples Modalisa	18
renomme.variable	20
rgrs.update	20
rp99	21
selectwd	21
<b>Index</b>	<b>23</b>

---

Cartographie                      *Représentations cartographiques simples*

---

**Description**

Fonctions permettant la représentation cartographique simple de données de type proportions (`carte.prop`), effectifs (`carte.eff`) ou qualitatif (`carte.qual`).

**Usage**

```
carte.prop(sp, data, varname, sp.key="id", data.key="id", diverg=FALSE,
           diverg.zero=0, nbcuts=6, at=NULL, at.lim=FALSE, main="",
           sub=NULL, posleg="topleft", palette.pos="Reds",
           palette.neg="Blues", palette=NULL, leg.options=NULL, ...)
```

```
carte.eff(sp, data, varname, sp.key="id", data.key="id", nbcuts=4,
          at=NULL, main="", sub=NULL, posleg="topleft", col.bg="red",
          col.border="white", cex=5, pch=21, plot.polygons=TRUE,
          leg.options=NULL, ...)
```

```
carte.qual(sp, data, varname, sp.key="id", data.key="id", main="", sub=NULL,
           posleg="topleft", palette.qual="Set3", palette=NULL,
```

```
leg.options=NULL, ...)
```

```
carte.labels(sp, labels, coords=NULL, cex=1, font=2, col="black",
             outline=FALSE, outline.decal=1, outline.col="white")
```

## Arguments

<code>sp</code>	objet spatial, de classe <code>SpatialPolygonsDataFrame</code> .
<code>data</code>	tableau de données contenant la variable à représenter.
<code>varname</code>	nom de la variable à représenter (sous forme de chaîne de caractère).
<code>sp.key</code>	nom de la variable de jointure de l'objet spatial.
<code>data.key</code>	nom de la variable de jointure du tableau de données.
<code>diverg</code>	si <code>TRUE</code> , les données comportent à la fois des valeurs positives et négatives, à représenter dans des schémas de couleur différents.
<code>diverg.zero</code>	si <code>diverg</code> vaut <code>TRUE</code> , valeur qui définit la limite pour laquelle les valeurs sont positives ou négatives (0 en général par défaut, mais on peut la positionner à 1 pour une carte représentant un ratio par exemple).
<code>nbcuts</code>	nombre de classes de valeurs pour la légende ( <code>carte.eff</code> ) et pour les couleurs ( <code>carte.prop</code> ).
<code>at</code>	bornes des classes de valeurs pour la légende ( <code>carte.eff</code> ) et pour les couleurs ( <code>carte.prop</code> ).
<code>at.lim</code>	si <code>TRUE</code> , les valeurs minimum et maximum sont ajoutées si besoin aux intervalles donnés via l'option <code>at</code> pour <code>carte.prop</code> .
<code>main</code>	titre de la carte.
<code>sub</code>	sous-titre de la carte.
<code>posleg</code>	position de la légende, à indiquer de la même manière que pour <code>legend</code> . Si l'argument vaut "none" ou <code>NULL</code> , aucune légende n'est affichée.
<code>col.bg</code>	couleur des symboles pour <code>carte.eff</code> .
<code>col.border</code>	couleur de la bordure des symboles pour <code>carte.eff</code> .
<code>cex</code>	facteur d'agrandissement des symboles ( <code>carte.eff</code> ) ou des labels ( <code>carte.labels</code> ).
<code>pch</code>	type de symbole pour <code>carte.eff</code> .
<code>plot.polygons</code>	si <code>FALSE</code> , le contenu de l'objet spatial (polygones) n'est pas affiché.
<code>palette.pos</code>	nom de la palette à utiliser pour les classes de valeurs positives. Chaîne de caractère transmise à <code>RColorBrewer</code> (voir <a href="#">Details</a> ).
<code>palette.neg</code>	nom de la palette à utiliser pour les classes de valeurs négatives. Chaîne de caractère transmise à <code>RColorBrewer</code> (voir <a href="#">Details</a> ).
<code>palette.qual</code>	nom de la palette à utiliser pour les catégories. Chaîne de caractère transmise à <code>RColorBrewer</code> (voir <a href="#">Details</a> ).
<code>palette</code>	palette de couleur spécifiée manuellement.
<code>labels</code>	vecteur de chaînes de caractère contenant les labels à écrire.

<code>coords</code>	coordonnées de positionnement des labels. Si NULL, les coordonnées sont calculées en fonction de la forme de chaque polygone.
<code>col</code>	couleur des labels.
<code>font</code>	type de police utilisée pour les labels. Voir <a href="#">par</a> .
<code>outline</code>	si TRUE, une bordure est affichée autour des labels.
<code>outline.decal</code>	décalage à utiliser pour les bordures de labels.
<code>outline.col</code>	couleur pour les bordures de labels.
<code>...</code>	paramètres supplémentaires passés à <code>spplot</code> .
<code>leg.options</code>	Liste de paramètres supplémentaires passée à la fonction d'affichage de la légende correspondante.

### Details

Pour la manière de spécifier des palettes à RColorBrewer, on pourra utiliser l'outil interactif à l'adresse <http://colorbrewer.org>, les noms de palette sont les mêmes. Si vous utilisez une des palettes du site, les auteurs du projet apprécient que celui-ci soit cité.

Si le nombre de classes de valeurs est trop élevé, il se peut que la palette spécifiée ne dispose pas de suffisamment de couleurs. Il faut alors soit réduire le nombre de classes, soit choisir une autre palette, soit spécifier une palette manuellement.

Les arguments supplémentaires passés à la fonction d'affichage de la légende doivent l'être sous la forme d'une liste, par exemple `leg.options=list(na.string="Manquant")`.

### Value

Affiche la carte et la légende correspondantes. Ne retourne pas de valeur particulière.

### Author(s)

Julien Barnier <[julien.barnier@ens-lyon.fr](mailto:julien.barnier@ens-lyon.fr)>, Joël Gombin <[joel.gombin@gmail.com](mailto:joel.gombin@gmail.com)>

### See Also

[carte.prop.legende](#), [carte.eff.legende](#), [carte.qual.legende](#), [spplot](#), [legend](#), [brewer.pal](#), [palette](#), [par](#)

### Examples

```
data(lyon)
data(rp99)

require(sp)
plot(lyon)

carte.prop(lyon, rp99, "tx.chom", sp.key="DepCom", data.key="code")
carte.prop(lyon, rp99, "tx.chom", sp.key="DepCom", data.key="code", main="Taux de chômage 1999")

carte.eff(lyon, rp99, "pop.act", sp.key="DepCom", data.key="code")
```

```

carte.eff(lyon, rp99, "pop.act", sp.key="DepCom", data.key="code", main="Population active e
rp99$qual <- sample(c("A","B","C","D","E"), nrow(rp99),replace=TRUE)
carte.qual(lyon, rp99, "qual", sp.key="DepCom", data.key="code", main="Types d'arrondissement

carte.prop(lyon, rp99, "tx.chom", sp.key="DepCom", data.key="code", main="Taux de chômage 19
carte.labels(lyon, lyon@data$Nom_Com, outline=TRUE)

```

---

Cartographie.legende

*Affichage des légendes pour les représentations cartographiques*

---

## Description

Fonctions permettant l'affichage de la légende pour une représentation cartographique simple de données de type proportions (`carte.prop.legende`), effectifs (`carte.eff.legende`) ou qualitatif (`carte.qual.legende`).

## Usage

```

carte.prop.legende(posleg="topleft", at, palette, rect.width=0.03,
                  rect.height=0.03, inset=c(0.02,0.02), na.leg=TRUE,
                  na.string="NA", cex=0.8)

carte.qual.legende(posleg="topleft", qual.names, palette, rect.width=0.03,
                  rect.height=0.03, inset=c(0.02,0.02), na.leg=TRUE,
                  na.string="NA", cex=0.8)

carte.eff.legende(posleg="bottomleft", at, cex.leg, inset=c(0.02,0.02),
                  cex.text=0.8, col.bg, col.border, pch)

```

## Arguments

<code>posleg</code>	position de la légende, à indiquer de la même manière que pour <code>legend</code> . Si l'argument vaut "none" ou NULL, aucune légende n'est affichée.
<code>palette</code>	Palette utilisée pour l'affichage.
<code>at</code>	Limites des classes de valeurs.
<code>qual.names</code>	Nom des classes de valeurs.
<code>rect.width</code>	Largeur des rectangles.
<code>rect.height</code>	Hauteur des rectangles.
<code>inset</code>	Position par rapport au cadre du graphique.
<code>na.leg</code>	Affichage ou non de la légende pour les valeurs manquantes.

<code>na.string</code>	Label de la légende pour les valeurs manquantes.
<code>cex</code>	Taille des caractères de la légende.
<code>cex.text</code>	Taille des caractères de la légende ( <code>carte.eff.legende</code> ).
<code>cex.leg</code>	Taille des symboles de la légende ( <code>carte.eff.legende</code> ).
<code>col.bg</code>	Couleur des symboles de la légende ( <code>carte.eff.legende</code> ).
<code>col.border</code>	Couleur de la bordure des symboles de la légende ( <code>carte.eff.legende</code> ).
<code>pch</code>	Type de symboles de la légende ( <code>carte.eff.legende</code> ).

### Details

A priori ces fonctions ne sont jamais utilisées directement, elles sont appelées par la fonction cartographique correspondante, éventuellement en leur fournissant des arguments via la liste `leg.options`.

### Value

Affiche la légende correspondante. Ne retourne pas de valeur particulière.

### Author(s)

Julien Barnier <julien.barnier@ens-lyon.fr>

### See Also

[carte.prop](#), [carte.eff](#), [carte.qual](#), [legend](#), [palette](#), [brewer.pal](#), [par](#), [points](#)

---

copie

*Export d'un objet au format HTML*

---

### Description

Cette fonction transforme l'objet passé en argument en HTML via R2HTML, puis le place dans le presse-papier ou dans un fichier.

### Usage

```
copie(obj, ...)  
## Default S3 method:  
copie(obj, append=FALSE, file=FALSE, filename="temp.html", ...)
```

**Arguments**

<code>obj</code>	nom de l'objet à exporter
<code>append</code>	si FALSE (par défaut), remplace le contenu du presse-papier ou du fichier par le résultat. Si TRUE, ajoute le résultat à la suite du contenu du presse-papier ou du fichier
<code>file</code>	si FALSE (par défaut), exporte dans le presse-papier. Si TRUE, exporte dans le fichier <code>filename</code>
<code>filename</code>	nom du fichier dans lequel exporter l'objet, si <code>file=TRUE</code>
<code>...</code>	arguments passés à la fonction <code>HTML()</code>

**Details**

ATTENTION, pour l'instant cette fonction ne fonctionne que sous Windows en ce qui concerne la copie dans le presse-papier. Sous Linux elle nécessite la présence du programme `xclip`. Elle n'a pas pu être testée sous Mac OS X.

**Value**

Après exécution, si `file=FALSE` le presse-papier contient une copie de l'objet formaté en HTML. On peut alors facilement coller le résultat directement sous Microsoft Excel, puis dans Word avec un second copier/coller.

Si on positionne l'argument `file` à TRUE, l'objet est exporté dans un fichier (par défaut nommé `temp.html` et situé dans le répertoire de travail. On peut ensuite l'intégrer directement dans Microsoft Word ou OpenOffice Writer via le menu Insertion > Fichier.

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[HTML](#), [copie.proptab](#)

**Examples**

```
data(iris)
tab <- table(cut(iris$Sepal.Length,8),cut(iris$Sepal.Width,4))
## Not run: copie(tab)
```

---

`copie.proptab`*Export d'un objet proptab au format HTML*

---

## Description

Applique la fonction générique `copie` à un tableau de classe `proptab`.

## Usage

```
## S3 method for class 'proptab':  
copie(obj, percent=NULL, digits=NULL, justify="right", ...)
```

## Arguments

<code>obj</code>	nom de l'objet à exporter
<code>percent</code>	affichage du symbole pourcentage dans les cellules du tableau
<code>digits</code>	nombre de décimales à afficher
<code>justify</code>	justification du contenu des cellules ("left", "right" ou "centre")
<code>...</code>	arguments passés à la fonction <code>copie()</code>

## Details

Pour plus d'informations sur les arguments et les résultats de la fonction, se référer à l'aide des fonctions `format.proptab` et `copie`.

## Author(s)

Julien Barnier <julien.barnier@ens-lyon.fr>

## See Also

[copie](#), [format.proptab](#)

## Examples

```
data(iris)  
tab <- table(cut(iris$Sepal.Length,8), cut(iris$Sepal.Width,4))  
ptab <- lprop(tab, percent=TRUE)  
## Not run: copie(ptab)
```

---

`cramer.v`*Calcule le V de Cramer d'un tableau croisé*

---

**Description**

Cette fonction calcule le V de Cramer pour un tableau de contingence.

**Usage**

```
cramer.v(tab)
```

**Arguments**

`tab`            Tableau croisé.

**Details**

Le tableau croisé passé en argument est un objet de type table.

**Value**

Valeur du V pour le tableau.

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**Examples**

```
v1 <- factor(round(runif(500,1,4)))
v2 <- factor(round(runif(500,1,3)))

tab <- table(v1,v2)
print(tab)
cramer.v(tab)
```

---

`format.proptab`*Formate le contenu d'un tableau contenant des proportions*

---

**Description**

Cette fonction formate un tableau contenant des pourcentages en contrôlant leur présentation. Cette fonction est prévue pour une utilisation interne, et ne devrait pas être utilisée directement.

**Usage**

```
## S3 method for class 'proptab':
format(x, digits=NULL, percent=NULL, justify="right", ...)
```

**Arguments**

<code>x</code>	tableau à formater
<code>digits</code>	indique le nombre de décimales à conserver pour l'affichage. Si <code>NULL</code> , on utilise l'attribut <code>digits</code> de <code>x</code>
<code>percent</code>	indique si on doit afficher ( <code>TRUE</code> ) ou non ( <code>FALSE</code> ) le symbole <code>%</code> dans chaque case du tableau. Si <code>NULL</code> , on utilise l'attribut <code>percent</code> de <code>x</code>
<code>justify</code>	justification du contenu des cellules (" <code>left</code> ", " <code>right</code> " ou " <code>centre</code> ")
<code>...</code>	arguments passés à la fonction <code>format()</code>

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[copie.proptab](#), [print.proptab](#)

---

freq

*Retourne le tri à plat d'une variable*

---

**Description**

Cette fonction affiche le tri à plat d'une variable (vecteur).

**Usage**

```
freq(x, digits=1, cum=FALSE, total=FALSE, exclude=NULL, sort="")
```

**Arguments**

<code>x</code>	vecteur pour lequel on souhaite obtenir le tri à plat ou tableau de dimension 1
<code>digits</code>	nombre de chiffres à conserver après la virgule
<code>cum</code>	si <code>TRUE</code> , affiche les pourcentages cumulés
<code>total</code>	si <code>TRUE</code> , affiche le total des effectifs
<code>exclude</code>	valeurs à exclure du tri à plat (aucune par défaut)
<code>sort</code>	si " <code>inc</code> ", le tableau résultat est trié par effectifs croissants. Si " <code>dec</code> ", par effectifs décroissants. Sinon, l'ordre des modalités par défaut est conservé.

**Details**

L'objet `x` est soit un vecteur, dans ce cas le tri à plat est calculé à l'aide de la fonction `table`, soit déjà un tri à plat, c'est-à-dire une table à une dimension, dans ce cas c'est cette table qui est utilisée telle quelle.

**Value**

Un data frame dont les noms de lignes sont les modalités de la variables, et dont les colonnes sont les effectifs, le pourcentage et (si demandé) le pourcentage cumulé de ces modalités.

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[table](#), [prop](#)

**Examples**

```
v <- c(round(runif(230,1,5)), NA)
freq(v)
freq(v, cum=TRUE)
freq(v, exclude=NA)
freq(v, exclude=c(1,2,NA))
freq(v, digits=3)
freq(v, total=TRUE)
freq(v, sort="inc")
tab <- table(v)
freq(tab)
```

---

genere.tableau      *Génère une représentation ODF d'un objet*

---

**Description**

Cette fonction fait appel à la fonction `odfTable()` correspondant au type d'objet passé en paramètres.

**Usage**

```
genere.tableau(x, ...)
```

**Arguments**

x	objet à exporter
...	arguments passés à la fonction <code>odfTable()</code>

**Details**

Actuellement la fonction permet de générer une version ODF des objets de type table à une ou deux dimensions, des data frames, des matrices et des vecteurs.

Actuellement que cette fonction n'est qu'une interface à `odfTable()` qui évite de devoir convertir les objets de type table en matrice ou en data frame.

**Value**

Renvoie une représentation au format ODF (XML) de l'objet.

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[odfTable](#)

**Examples**

```
## Not run:
## Not run:
data(iris)
tab <- table(iris$Species)
genere.tableau(tab)

## End(Not run)
```

---

hdv2003

*Histoire de Vie 2003*

---

**Description**

Échantillon de 2000 individus et de 20 variables issu de l'enquête *Histoire de Vie* réalisé par l'INSEE en 2003.

**Usage**

```
data(hdv2003)
```

**Format**

Data frame comportant 2000 lignes et 20 colonnes

**Source**

Fichiers détail de l'INSEE : [http://www.insee.fr/fr/themes/detail.asp?ref\\_id=fd-HDV03](http://www.insee.fr/fr/themes/detail.asp?ref_id=fd-HDV03)

---

`lyon`*Contour des arrondissements de Lyon*

---

**Description**

Contour des 9 arrondissements de Lyon pour représentation cartographique

**Usage**

```
data(lyon)
```

**Format**

Objet de classe SpatialPolygonsDataFrame

---

`mls.export`*Export de données vers Modalisa*

---

**Description**

Exporte un data frame dans un fichier texte importable ensuite sous Modalisa avec la fonction *Import ASCII*

**Usage**

```
mls.export(df, filename)
```

**Arguments**

<code>df</code>	data frame à exporter
<code>filename</code>	Nom du fichier d'export

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[mls.import](#)

**Examples**

```
## Not run:  
## Not run: mls.export(mydf, "export_modalisa.txt")
```

---

mls.import                    *Import de fichiers Modalisa*

---

## Description

Importe un fichier Modalisa enregistré sous forme d'export ASCII

## Usage

```
mls.import(filename, enc = "latin1", modif.names = TRUE)
```

## Arguments

filename	Nom du fichier à importer
enc	Encodage du fichier à importer (normalement toujours latin1)
modif.names	Correction ou non des noms de variables. Si <code>modif.names</code> vaut <code>TRUE</code> , alors les noms de variables importés sont convertis en minuscules et les espaces remplacés par des tirets bas.

## Value

Retourne un data frame contenant les données importées.

## Author(s)

Julien Barnier <julien.barnier@ens-lyon.fr>

## See Also

[mls.export](#), [mls.eclate.multi](#), [mls.eclate.ordo](#), [mls.table.multi](#)

## Examples

```
## Not run:  
## Not run: mydf <- mls.import("export_modalisa.TXT")
```

**Description**

Fonctions permettant le calcul de moyennes (`wtd.mean`), variances (`wtd.var`), tris à plat et tableaux croisés (`wtd.table`) pour des variables pondérées.

**Usage**

```
wtd.mean(x, weights = NULL, normwt = "ignored", na.rm = TRUE)
wtd.var(x, weights = NULL, normwt = FALSE, na.rm = TRUE)
wtd.table(x, y = NULL, weights = NULL, normwt = FALSE, na.rm = TRUE)
```

**Arguments**

<code>x</code> , <code>y</code>	Vecteurs de données. Doit être numérique pour <code>wtd.mean</code> et <code>wtd.var</code> .
<code>weights</code>	Vecteur des poids. Doit être de même longueur que <code>x</code>
<code>normwt</code>	Normalisation des poids pour que les effectifs totaux pondérés soient les mêmes que les effectifs initiaux
<code>na.rm</code>	Suppression des valeurs manquantes

**Details**

Si `weights` n'est pas fourni, les fonctions utilisent une pondération uniforme.

**Value**

Pour `wtd.table`, si un seul vecteur est fourni la fonction calcule le tri à plat pondéré de la variables. Si deux vecteurs sont passés en paramètres on obtient le tri croisé pondéré des deux variables.

**Author(s)**

Les fonctions `wtd.mean` et `wtd.var` sont des copies conformes des fonctions du même nom de l'extension `Hmisc`. Elles ont été développées par : Frank Harrell Department of Biostatistics Vanderbilt University School of Medicine [f.harrell@vanderbilt.edu](mailto:f.harrell@vanderbilt.edu)

Pour `wtd.table` : Julien Barnier <[julien.barnier@ens-lyon.fr](mailto:julien.barnier@ens-lyon.fr)>

**See Also**

[table](#), [mean](#), [var](#), [wtd.table](#), [wtd.quantile](#) et l'extension `survey`

## Examples

```
data(hdv2003)

mean(hdv2003$age)
wtd.mean(hdv2003$age, weights=hdv2003$poids)

table(hdv2003$sexe)
wtd.table(hdv2003$sexe, weights=hdv2003$poids)
wtd.table(hdv2003$sexe, weights=hdv2003$poids, normwt=TRUE)

table(hdv2003$sexe, hdv2003$hard.rock)
wtd.table(hdv2003$sexe, hdv2003$hard.rock, weights=hdv2003$poids)
```

---

<code>print.proptab</code>	<i>Affiche un tableau contenant des proportions</i>
----------------------------	---

---

## Description

Cette fonction affiche un tableau contenant des pourcentages en contrôlant leur présentation.

## Usage

```
## S3 method for class 'proptab':
print(x, digits=NULL, percent=NULL, justify="right", ...)
```

## Arguments

<code>x</code>	tableau à afficher
<code>digits</code>	indique le nombre de décimales à conserver pour l'affichage. Si <code>NULL</code> , on utilise l'attribut <code>digits</code> de <code>tab</code>
<code>percent</code>	indique si on doit afficher ( <code>TRUE</code> ) ou non ( <code>FALSE</code> ) le symbole <code>%</code> dans chaque case du tableau. Si <code>NULL</code> , on utilise l'attribut <code>percent</code> de <code>tab</code>
<code>justify</code>	justification du contenu des cellules (" <code>left</code> ", " <code>right</code> " ou " <code>centre</code> ")
<code>...</code>	arguments passés à la fonction <code>print.table()</code>

## Author(s)

Julien Barnier <julien.barnier@ens-lyon.fr>

## See Also

[format.proptab](#), [Profils](#), [print](#)

## Exemples

```
tab <- table(x=round(runif(100,1,3)),y=round(runif(100,1,5)))
ptab <- lprop(tab, digits=1, percent=TRUE)
print(ptab)
print(ptab, digits=2, percent=FALSE)
```

## Description

Fonctions calculant différents pourcentages d'un tableau croisé

## Usage

```
cprop(tab, digits = 1, total = TRUE, percent = FALSE)
lprop(tab, digits = 1, total = TRUE, percent = FALSE)
prop(tab, digits = 1, total = TRUE, percent = FALSE)
thprop(tab, digits = 1, percent = FALSE)
theff(tab, digits = 2)
residus(tab, digits = 2)
```

## Arguments

tab	Tableau croisé (objet de type table)
digits	Nombre de chiffres après la virgule à conserver à l'affichage
total	Ajouter des lignes/colonnes pour les marges du tableau
percent	Ajout du symbole % dans chaque case lors de l'affichage du tableau

## Details

Dans le cas des tableaux contenant des proportions (`cprop`, `lprop`, `prop`, `thprop`), les options `digits` et `percent` sont des attributs du tableau résultant qui contrôlent l'affichage du tableau avec `print` ou `copie`. On peut modifier ponctuellement ces options en les passant directement à `print.proptab` ou `copie.proptab`. Les données numériques stockées conservent l'intégralité des valeurs décimales.

## Value

`cprop` retourne un tableau contenant les pourcentages colonnes, `lprop` renvoie un tableau contenant les pourcentages lignes, `prop` renvoie un tableau contenant les pourcentages globaux, `thprop` renvoie un tableau de pourcentages théoriques sous l'hypothèse d'indépendance, `theff` renvoie un tableau d'effectifs théoriques sous l'hypothèse d'indépendance, et `residus` renvoie le tableau des résidus de Pearson.

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[table](#), [prop.table](#), [sweep](#), [chisq.test](#), [print.proptab](#), [copie.proptab](#)

**Examples**

```
v1 <- factor(round(runif(500,1,4)))
v2 <- factor(round(runif(500,1,3)))

tab <- table(v1,v2)
tab
lprop(tab,digits=5)
cprop(tab,digits=2)
prop(tab)
thprop(tab, percent=TRUE)
theff(tab)
residus(tab)
```

---

quant.cut

*Transforme une variable quantitative en variable qualitative*

---

**Description**

Cette fonction transforme une variable quantitative en une variable qualitative ayant des modalités comportant les même effectifs.

**Usage**

```
quant.cut(var, nbclass, include.lowest=TRUE, right=FALSE, dig.lab=5, ...)
```

**Arguments**

var	variable (vecteur) à transformer
nbclass	nombre de classes souhaité
include.lowest, right, dig.lab, ...	paramètres passés à la fonction cut

**Details**

Il s'agit juste d'un wrapper autour des fonctions cut et quantile

**Value**

Renvoie un vecteur de type factor généré par cut

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[cut](#), [quantile](#)

**Examples**

```
data(iris)
sepal.width3cl <- quant.cut(iris$Sepal.Width,3)
freq(sepal.width3cl)
```

---

Questions multiples Modalisa

*Traitement des questions à réponses multiples importées depuis  
Modalisa.*

---

**Description**

Ces fonctions permettent de transformer ou de traiter des questions à réponses multiples importées depuis un export Modalisa.

**Usage**

```
mls.table.multi(var)
mls.eclate.multi(var, vname = "MLS.mult.", mnames = NULL)
mls.eclate.ordo(var, vname = "MLS.ordo.", mnames = NULL, nb = 3)
```

**Arguments**

<code>var</code>	Variable correspondant à une question à réponses multiples
<code>vname</code>	Préfixe à ajouter aux noms des variables générées (si <code>mnames=NULL</code> )
<code>mnames</code>	Noms des variables générées
<code>nb</code>	Dans le cas de questions à réponses multiples ordonnées, nombre de modalités à retenir

**Details**

`mls.table.multi` génère le tri à plat des modalités d'une question à réponses multiples. `mls.eclate.multi` transforme une question à réponses multiples en autant de questions binaires qu'il y a de modalités. `mls.eclate.ordo` fait la même chose pour une question à réponses multiples ordonnées.

**Value**

`mls.table.multi` renvoie une table contenant le tri à plat des modalités de la question à réponses multiples.

`mls.table.multi` renvoie un data frame avec autant de variables que la variable initiale a de modalités. Chaque variable créée possède les modalités 0 si l'individu correspondant a choisi la modalité, et N sinon.

`mls.table.ordo` renvoie un data frame semblable à celui généré par `mls.table.multi`, mais il se limite aux `nb` premières modalités.

**Note**

De manière générale, il est préférable de toujours éviter l'usage de questions à réponses multiples dans Modalisa, et de privilégier dès la conception du questionnaire l'usage de séries de questions binaires.

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[mls.import](#)

**Examples**

```
## Not run:
## Not run:
mydf <- mls.import("export_modalisa.TXT")
mls.table.multi(mydf$couleurs)
test <- cbind(test,mls.eclate.multi(mydf$couleurs, vname="couleur"))

## End(Not run)
```

---

`renomme.variable` *Renomme une colonne d'un tableau de données*

---

**Description**

Renomme une colonne (variable) d'un tableau de données

**Usage**

```
renomme.variable(df, old, new)
```

**Arguments**

df	tableau de données (data.frame)
old	nom de la variable à renommer
new	nouveau nom

**Value**

Renvoie un tableau de données avec la colonne indiquée renommée

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**Examples**

```
data(iris)
str(iris)
iris <- renomme.variable(iris, "Species", "especies")
str(iris)
```

---

rgrs.update

*Mise à jour du paquet Rgrs*

---

**Description**

Vérifie si une nouvelle version de Rgrs est disponible et effectue la mise à jour si nécessaire.

**Usage**

```
rgrs.update()
```

**Details**

Nécessite d'avoir une connexion à Internet active pour pouvoir contacter le serveur R-forge.

**Author(s)**

Julien Barnier <julien.barnier@ens-lyon.fr>

**See Also**

[update.packages](#)

rp99

*Recensement 1999 - Communes du Rhône*

---

**Description**

Résultats tirés du recensement de la population de 1999 pour les communes du Rhône.

**Usage**

```
data(rp99)
```

**Format**

Data frame comportant 301 lignes et 21 colonnes

**Source**

Bases de données recensement de l'INSEE : <http://www.insee.fr/fr/bases-de-donnees/default.asp?page=recensements.htm>

---

selectwd

*Sélecteur de répertoire de travail*

---

**Description**

Affiche ue boîte de sélection de répertoire en Tk et modifie le répertoire de travail selon la sélection.

**Usage**

```
selectwd()
```

**Value**

La fonction renvoie le répertoire choisi sous forme de chaîne de caractères, et affiche la commande `setwd` correspondant à la sélection effectuée.

**Author(s)**

Julien Barnier <[julien.barnier@ens-lyon.fr](mailto:julien.barnier@ens-lyon.fr)>

**See Also**

[setwd](#), [getwd](#)

# Index

- \*Topic **connection**
    - copie, 6
    - copie.proptab, 7
    - genere.tableau, 10
  - \*Topic **datasets**
    - hdv2003, 11
    - lyon, 12
    - rp99, 21
  - \*Topic **file**
    - mls.export, 12
    - mls.import, 13
  - \*Topic **hplot**
    - Cartographie, 2
    - Cartographie.legende, 4
  - \*Topic **manip**
    - quant.cut, 17
    - Questions multiples
      - Modalisa, 18
    - renomme.variable, 19
  - \*Topic **print**
    - format.proptab, 8
  - \*Topic **spatial**
    - Cartographie, 2
    - Cartographie.legende, 4
  - \*Topic **univar**
    - cramer.v, 8
    - freq, 9
    - Pondérations, 14
    - print.proptab, 15
    - Profils, 16
  - \*Topic **utilities**
    - mls.export, 12
    - mls.import, 13
    - Questions multiples
      - Modalisa, 18
    - rgrs.update, 20
    - selectwd, 21
- brewer.pal, 4, 5
- carte.eff, 5
- carte.eff (Cartographie), 2
- carte.eff.legende, 4
- carte.eff.legende  
(Cartographie.legende), 4
- carte.labels (Cartographie), 2
- carte.prop, 5
- carte.prop (Cartographie), 2
- carte.prop.legende, 4
- carte.prop.legende  
(Cartographie.legende), 4
- carte.qual, 5
- carte.qual (Cartographie), 2
- carte.qual.legende, 4
- carte.qual.legende  
(Cartographie.legende), 4
- Cartographie, 2
- Cartographie.legende, 4
- chisq.test, 17
- copie, 6, 7
- copie.proptab, 6, 7, 9, 16, 17
- cprop (Profils), 16
- cramer.v, 8
- cut, 18
- format.proptab, 7, 8, 15
- freq, 9
- genere.tableau, 10
- getwd, 21
- hdv2003, 11
- HTML, 6
- legend, 4, 5
- lprop (Profils), 16
- lyon, 12
- mean, 14
- mls.eclate.multi, 13

mls.eclate.multi (*Questions multiples Modalisa*), 18  
mls.eclate.ordo, 13  
mls.eclate.ordo (*Questions multiples Modalisa*), 18  
mls.export, 12, 13  
mls.import, 12, 13, 19  
mls.table.multi, 13  
mls.table.multi (*Questions multiples Modalisa*), 18

odfTable, 11

palette, 4, 5  
par, 3–5  
points, 5  
Pondérations, 14  
print, 15  
print.proptab, 9, 15, 16, 17  
Profils, 15, 16  
prop, 10  
prop (*Profils*), 16  
prop.table, 17

quant.cut, 17  
quantile, 18  
Questions multiples Modalisa, 18

renomme.variable, 19  
residus (*Profils*), 16  
rgrs.update, 20  
rp99, 21

selectwd, 21  
setwd, 21  
spplot, 4  
sweep, 17

table, 10, 14, 17  
theff (*Profils*), 16  
thprop (*Profils*), 16

update.packages, 20

var, 14

wtd.mean (*Pondérations*), 14  
wtd.quantile, 14  
wtd.table, 14  
wtd.table (*Pondérations*), 14  
wtd.var (*Pondérations*), 14