

Package ‘rgrs’ documentation

of

July 17, 2008

Type Package

Title Fonctions pour faciliter l’usage de R en sciences sociales

Version 0.1-4

Date 2008-07-17

Author Julien Barnier

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

Description Ces fonctions sont avant tout utilisées en interne par les membres du GRS (Groupe de Recherche sur la Socialisation)

License GPL-3

Encoding UTF-8

Suggests R2HTML

R topics documented:

copie	2
cramer.v	3
freq	4
mls.export	5
mls.import	6
Questions multiples Modalisa	7
Profils	8
quant.cut	9
rgrs.update	10

Index	11
--------------	-----------

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, append=FALSE, file=FALSE, filename="temp.html", ...)
```

Arguments

obj	nom de l'objet à exporter
append	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
file	si FALSE (par défaut, exporte dans le presse-papier. Si TRUE, exporte dans le fichier filename
filename	nom du fichier dans lequel exporter l'objet, si file=TRUE
...	arguments passés à la fonction HTML()

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-lsh.fr>

References

Pas pour l'instant...

See Also[HTML](#)**Examples**

```
data(iris)
tab <- table(cut(iris$Sepal.Length,8),cut(iris$Sepal.Width,4))
copie(tab)
```

`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-lsh.fr>

References

Pas pour l'instant...

Examples

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

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

`freq`*Retourne le tri à plat d'une variable*

Description

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

Usage

```
freq(var, digits=1, eff=TRUE, cum=FALSE, total=FALSE, exclude=NULL)
```

Arguments

<code>var</code>	vecteur pour lequel on souhaite obtenir le tri à plat
<code>digits</code>	nombre de chiffres à conserver après la virgule
<code>eff</code>	si TRUE, affiche les effectifs
<code>cum</code>	si TRUE, affiche les pourcentages cumulés
<code>total</code>	si TRUE, affiche le total des effectifs
<code>exclude</code>	valeurs à exclure du tri à plat (aucune par défaut)

Details

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

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-lsh.fr>

References

Pas pour l'instant...

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, eff=FALSE)
freq(v, total=TRUE)
```

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

df	data frame à exporter
filename	Nom du fichier d'export

Author(s)

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

See Also

[mls.import](#)

Examples

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

mls.import	<i>Import de fichiers Modalisa</i>
------------	------------------------------------

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-lsh.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")
```

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-lsh.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)
```

Profils

Fonctions de calcul de pourcentages à partir d'un tableau croisé

Description

Fonctions calculant différents pourcentages d'un tableau croisé

Usage

```
cprop(tab, digits = 1, add.tot = TRUE)
lprop(tab, digits = 1, add.tot = TRUE)
prop(tab, digits = 1, add.tot = TRUE)
thprop(tab, digits = 1)
theff(tab, digits = 2)
residus(tab, digits = 2)
```

Arguments

<code>tab</code>	Tableau croisé (objet de type table)
<code>digits</code>	Nombre de chiffres après la virgule à conserver
<code>add.tot</code>	Ajouter des lignes/colonnes pour les marges du tableau

Details

Ces fonctions calculent les profils lignes ou colonnes d'un tableau croisé

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-lsh.fr>

See Also

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

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)
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-lsh.fr>

References

Pas pour l'instant...

See Also

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

Examples

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

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-lsh.fr>

See Also

[update.packages](#)

Index

*Topic **connection**

copie, [1](#)

*Topic **file**

mls.export, [5](#)

mls.import, [6](#)

*Topic **manip**

quant.cut, [9](#)

Questions multiples
Modalisa, [7](#)

*Topic **univar**

cramer.v, [3](#)

freq, [4](#)

Profils, [8](#)

*Topic **utilities**

mls.export, [5](#)

mls.import, [6](#)

Questions multiples
Modalisa, [7](#)

rgrs.update, [10](#)

chisq.test, [9](#)

copie, [1](#)

cprop(Profils), [8](#)

cramer.v, [3](#)

cut, [10](#)

freq, [4](#)

HTML, [2](#)

lprop(Profils), [8](#)

mls.eclate.multi, [6](#)

mls.eclate.multi(Questions
multiples Modalisa), [7](#)

mls.eclate.ordo, [6](#)

mls.eclate.ordo(Questions
multiples Modalisa), [7](#)

mls.export, [5](#), [6](#)

mls.import, [5](#), [6](#), [8](#)

mls.table.multi, [6](#)

mls.table.multi(Questions
multiples Modalisa), [7](#)

Profils, [8](#)

prop, [4](#)

prop(Profils), [8](#)

prop.table, [9](#)

quant.cut, [9](#)

quantile, [10](#)

Questions multiples Modalisa, [7](#)

residus(Profils), [8](#)

rgrs.update, [10](#)

sweep, [9](#)

table, [4](#), [9](#)

theff(Profils), [8](#)

thprop(Profils), [8](#)

update.packages, [10](#)