

# Package ‘EnQuireR’

January 2, 2012

**Type** Package

**Title** A package dedicated to questionnaires

**Version** 0.10

**Date** 2009-06-10

**Author** Fournier Gwenaelle, Cadoret Marine, Fournier Olivier, Le Poder  
Francois, Bouche Jerome, Le Sebastien

**Maintainer** Sebastien Le <sebastien.le@agrocampus-ouest.fr>

**Depends** FactoMineR, SensoMineR, Rcmdr, MASS

**Description** A package dedicated to questionnaires

**License** GPL (>= 2)

**LazyLoad** yes

**Encoding** latin1

**Repository** CRAN

**Date/Publication** 2010-07-29 15:47:03

## R topics documented:

EnquireR-package . . . . .	2
chisq.desc . . . . .	2
ENbarplot . . . . .	4
ENDensity . . . . .	5
ENellipse . . . . .	6
ENlisib . . . . .	7
ENmarking . . . . .	8
ENMCA . . . . .	9
GRmarking . . . . .	10
p_inertia . . . . .	11
tea . . . . .	12
XvsYbarplot . . . . .	12

EnquireR-package

*A package dedicated to questionnaires***Description**

The EnquireR package focuses on categorical variables and provides many tools to automate the survey process. It includes both univariate and multivariate data analyses comprising Multiple Correspondence Analysis (MCA), clustering analysis and semantic marking. The package also offers an easier view of the results by the automatic generation of a .pdf report and of a Beamer type presentation via the use of Sweave. In order to generate those report and presentation, you will need MikTeX and TeXnicCenter. To facilitate the installation for the user, the .sty and .cls files needed for the generation of the reports have been integrated to the package; hence its size.

**Details**

Package: EnquireR  
 Type: Package  
 Version: 0.09  
 Date: 2009-06-10  
 License: GPL (>= 2)  
 LazyLoad: yes

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
 Jerome Bouche <jerome.bouche@yahoo.fr>  
 Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
 Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
 Olivier Fournier <olivier.fournier76@gmail.com>  
 Francois Le Poder <francois.lepoder@gmail.com>

chisq.desc

*Display the results of a chi-square test***Description**

Color the cells of the table containing the results from the Chi-square test, crossing all the selected categorical variables, according to a selected threshold.

**Usage**

```
chisq.desc(dataset, Y, X, method="proba", print=TRUE, report=FALSE, language="english")
```

**Arguments**

dataset	a dataframe
Y	first group of variable
X	second group of variable (to be crossed with the first one)
method	if "proba" : the colour depends on the p-values given by the hypergeometric law. If "mean" : the colour depends on the value of the contribution to the chi-square distance
print	a logical value. If TRUE, tables of the chi-square distance are printed with cells coloured according to the chosen threshold
report	a logical value. If TRUE, a report is automatically generated
language	english or french

**Details**

This function is useful when comparing categorical variables. If report=TRUE, a directory named "EnQuireR" is created in your working directory (which can be known thanks to the getwd() function). This directory contains the final pdf report and also the teX file used to produce it.

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
Jerome Bouche <jerome.bouche@yahoo.fr>  
Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
Olivier Fournier <olivier.fournier76@gmail.com>  
Francois Le Poder <francois.lepoder@gmail.com>

**Examples**

```
## Not run:  
data(tea)  
chisq.desc(tea, c(1, 2), c(3, 4))  
  
chisq.desc(tea, c(1, 2), c(3, 4), report=TRUE)  
  
## End(Not run)
```

---

**ENbarplot***Plots of categorical variables*

---

**Description**

This function allows to obtain barplots either sorted by frequency or by alphabetical order. The percentage of missing values is indicated at the top of the graphical device.

**Usage**

```
ENbarplot(dataset,X,spl=FALSE,numr=NULL,numc=NULL,report=FALSE)
```

**Arguments**

dataset	a data frame
X	the numbers of the variables to be plotted
spl	if TRUE, the barplots will be sorted per length
numr	the number of plots per row (by default 2)
numc	the number of plots per column (by default 2)
report	a logical value. If TRUE, a report is automatically generated

**Details**

The colour changes depending on the number of individuals of the category. If `report=TRUE`, a directory named "EnQuireR" is created in your working directory (which can be known thanks to the `getwd()` function). This directory contains the final pdf report and also the tex file used to produce it.

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
Jerome Bouche <jerome.bouche@yahoo.fr>  
Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
Olivier Fournier <olivier.fournier76@gmail.com>  
Francois Le Poder <francois.lepoder@gmail.com>

**See Also**

[XvsYbarplot](#) which gives barplots of a categorical variable depending on the categoricals of another categorical variable.

**Examples**

```
## Not run:
data(tea)
ENbarplot(tea,c(18,20,21,23))

ENbarplot(tea,c(18,20,21,23),report=TRUE)

## End(Not run)
```

---

ENdensity

*Density levels*

---

**Description**

Display density levels around clusters on the cloud of the individuals or around a cloud of individuals.

**Usage**

```
ENDensity(res,classe=NULL,nlevels=20)
```

**Arguments**

res	the result of a factorial analysis
classe	the variable of interest. Density levels will be drawn around its categories
nlevels	number of contour levels desired

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
Jerome Bouche <jerome.bouche@yahoo.fr>  
Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
Olivier Fournier <olivier.fournier76@gmail.com>  
Francois Le Poder <francois.lepoder@gmail.com>

**Examples**

```
## Not run:
data(tea)
res.enmca=ENMCA(tea[,1:18])
cluster<-res.enmca$data[,ncol(res.enmca$data)]
X<-res.enmca$MCA
ENDensity(X,cluster,10)

## End(Not run)
```

ENellipse

*Confidence ellipses***Description**

This function allows to draw ellipses around the categories of a chosen variable.

**Usage**

```
ENellipse(matP, cluster, resampling="population", iter=500, alpha = 0.05, coord = c(1, 2), eig, cex = 1,
```

**Arguments**

matP	factorial coordinates from MCA
cluster	chosen categorical variable
resampling	If "population", the resampling is done within the whole dataset. If "cluster", it is done within the clusters
iter	number of iterations for the Bootstrap method, by default iter=500
alpha	the confidence level of the ellipses. By default alpha=0.05
coord	selected axes
eig	eigenvalues of the MCA
cex	graphical option: character size
color	graphical option: ellipses' colours

**Details**

This function can be used to draw ellipses around the centers of gravity of each cluster, when the chosen categorical variable is the variable obtained after the cluster analysis.

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
 Jerome Bouche <jerome.bouche@yahoo.fr>  
 Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
 Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
 Olivier Fournier <olivier.fournier76@gmail.com>  
 Francois Le Poder <francois.lepoder@gmail.com>

**Examples**

```
## Not run:
data(tea)
res.enmca=ENMCA(tea[,1:18])
x<-res.enmca$MCA$ind$coord[,1:2]
b<-res.enmca$data[,ncol(res.enmca$data)]
ENellipse(matP=x, cluster=b, resampling="population", iter=500, alpha = 0.05, coord = c(1, 2), eig=res.enmca$MCA$eig,
```

```
## End(Not run)
```

---

ENlisib *Graph readability*

---

### Description

This function allows to improve the MCA factor map readability.

### Usage

```
ENlisib(res.mca,nbvar,nbind,axes=c(1,2))
```

### Arguments

res.mca	results from an MCA
nbvar	chosen percentage or number for categories
nbind	chosen percentage or number for individuals
axes	selected axes, by default the two first axes are selected

### Details

Categories with high v-test and individuals with high coordinates are kept. The chosen number and percentage concern individuals and categories beyond a threshold.

### Author(s)

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
Jerome Bouche <jerome.bouche@yahoo.fr>  
Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
Olivier Fournier <olivier.fournier76@gmail.com>  
Francois Le Poder <francois.lepoder@gmail.com>

### Examples

```
## Not run:  
data(tea)  
res.enmca<-ENMCA(tea[,1:18])  
ENlisib(res.enmca$MCA,0.05,50,c(1,2))  
  
res<-MCA(tea[,1:18])  
ENlisib(res,0.05,50,c(1,2))  
  
## End(Not run)
```

**Description**

Returns a semantic marking of categorical variables with 3 different levels (variables, pairs and triplets of categories).

**Usage**

```
ENmarking(dataset, var.int, proba=0.05)
```

**Arguments**

dataset	a data frame
var.int	the variable to mark
proba	Threshold to select variables, pairs and triplets. By default 0.05

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
Jerome Bouche <jerome.bouche@yahoo.fr>  
Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
Olivier Fournier <olivier.fournier76@gmail.com>  
Francois Le Poder <francois.lepoder@gmail.com>

**See Also**

[GRmarking](#) which is a graphical representation of the semantic marking.

**Examples**

```
## Not run:  
data(tea)  
ENmarking(tea[,1:21],21)  
  
res.enmca=ENMCA(tea[,1:18])  
res.enmark=ENmarking(res.enmca$data,ncol(res.enmca$data))  
GRmarking(res.enmark,1)  
  
## End(Not run)
```

**Description**

This function allows to perform a cluster analysis following a MCA.

**Usage**

```
ENMCA(dataset, ind.sup = NULL, quanti.sup = NULL, quali.sup = NULL, axes = c(1, 2), row.w = NULL, level.ventil
```

**Arguments**

dataset	current dataset
ind.sup	a vector indicating the indexes of the supplementary individuals
quanti.sup	a vector indicating the indexes of the supplementary quantitatives variables
quali.sup	a vector indicating the indexes of the supplementary qualitatives variables
axes	a length 2 vector specifying the components to plot
row.w	an optional row weights (by default, uniform row weights)
level.ventil	a number corresponding to the level under which the category is ventilated; by default 0, and no ventilation is done
signif	a logical value. If TRUE, a p-value is associated to the 2 first dimensions
proba	Threshold used for the automatic description of the clusters. By default, proba=0.05
report	a logical value. If TRUE, a pdf report is produced using the Sweave package
language	a value of c("english","french") which determines the language to use for the report

**Details**

If report=TRUE, a directory named "EnQuireR" is created in your working directory (which can be known thanks to the getwd() function). This directory contains the final pdf report and also the teX file used to produce it.

**Value**

A list containing the following elements:

data	dataset with the new categorical variable, obtained with the cluster analysis
catdes	result of the catdes of the clusters
MCA	normal results of an MCA. On the MCA factor map, each cluster is displayed in a different colour

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
 Jerome Bouche <jerome.bouche@yahoo.fr>  
 Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
 Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
 Olivier Fournier <olivier.fournier76@gmail.com>  
 Francois Le Poder <francois.lepoder@gmail.com>

**Examples**

```
## Not run:
data(tea)
res.enmca=ENMCA(tea[,1:18])

## End(Not run)
```

---

 GRmarking

*Graphical outputs for a semantic marking*


---

**Description**

This function gives graphical outputs for the variables, pairs or triplets of categories obtained with a semantic marking.

**Usage**

```
GRmarking(res,lev,col.neg="lightblue",col.pos="pink",colour="black")
```

**Arguments**

res	the result of a semantic marking
lev	level of the semantic marking (1=variables, 2=pairs and 3=triplets)
col.neg	colour for the negative v-tests
col.pos	colour for the positive v-tests
colour	colour and font to be used for the characters

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
 Jerome Bouche <jerome.bouche@yahoo.fr>  
 Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
 Olivier Fournier <olivier.fournier76@gmail.com>  
 Francois Le Poder <francois.lepoder@gmail.com>

**Examples**

```
## Not run:
data(tea)
res=ENmarking(tea[,1:21],21)
GRmarking(res,lev=2)

## End(Not run)
```

---

p_inertia	<i>Test of MCA's percentages of inertia</i>
-----------	---

---

**Description**

This function creates new data sets and performs MCA on them. It then keeps the percentages of inertia and calculates the means and the associated p-values on the first two dimensions.

**Usage**

```
p_inertia(dataset)
```

**Arguments**

dataset            a data frame

**Author(s)**

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
Jerome Bouche <jerome.bouche@yahoo.fr>  
Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
Olivier Fournier <olivier.fournier76@gmail.com>  
Francois Le Poder <francois.lepoder@gmail.com>

**Examples**

```
## Not run:
data(tea)
p_inertia(tea[,1:5])

## End(Not run)
```

---

tea	<i>tea (data)</i>
-----	-------------------

---

### Description

The data used here concern a questionnaire on tea. We asked to 300 individuals how they drink tea (19 questions), what are their product's perception (12 questions) and some personal details (4 questions).

### Usage

```
data(tea)
```

### Format

A data frame with 300 rows and 35 columns. Rows represent the individuals, columns represent the different questions.

### Examples

```
## Not run:
data(tea)

## End(Not run)
```

---

XvsYbarplot	<i>Barplot per category with respect to a categorical variable</i>
-------------	--

---

### Description

This function allows to obtain coloured barplots of a categorical variable, depending on the categories of another categorical variable.

### Usage

```
XvsYbarplot(var1, var2, dataset, width = 1, space = NULL, names.arg = NULL,
  legend.text = NULL, horiz = FALSE, density = NULL, angle = 45,
  col = NULL, border = par("fg"), main = NULL, sub = NULL,
  xlab = NULL, ylab = NULL, xlim = NULL, ylim = NULL, xpd = TRUE,
  log = "", axes = TRUE, axisnames = TRUE, cex.axis = par("cex.axis"),
  cex.names = par("cex.axis"), inside = TRUE, plot = TRUE,
  axis.lty = 0, offset = 0, add = FALSE, ...)
```

**Arguments**

var1	the name of the first variable.
var2	the name of the second variable.
dataset	the dataset to be studied.
width	optional vector of bar widths. Re-cycled to length the number of bars drawn. Specifying a single value will no visible effect unless xlim is specified.
space	the amount of space (as a fraction of the average bar width) left before each bar. May be given as a single number or one number per bar. If height is a matrix and beside is TRUE, space may be specified by two numbers, where the first is the space between bars in the same group, and the second the space between the groups. If not given explicitly, it defaults to c(0,1) if height is a matrix and beside is TRUE, and to 0.2 otherwise.
names.arg	a vector of names to be plotted below each bar or group of bars. If this argument is omitted, then the names are taken from the names attribute of height if this is a vector, or the column names if it is a matrix.
legend.text	a vector of text used to construct a legend for the plot, or a logical indicating whether a legend should be included. This is only useful when height is a matrix. In that case given legend labels should correspond to the rows of height; if legend.text is true, the row names of height will be used as labels if they are non-null.
horiz	a logical value. If FALSE, the bars are drawn vertically with the first bar to the left. If TRUE, the bars are drawn horizontally with the first at the bottom.
density	a vector giving the density of shading lines, in lines per inch, for the bars or bar components. The default value of NULL means that no shading lines are drawn. Non-positive values of density also inhibit the drawing of shading lines..
angle	the slope of shading lines, given as an angle in degrees (counter-clockwise), for the bars or bar components.
col	a vector of colors for the bars or bar components. By default, grey is used if height is a vector, and a gamma-corrected grey palette if height is a matrix.
border	the color to be used for the border of the bars. Use border = NA to omit borders. If there are shading lines, border = TRUE means use the same colour for the border as for the shading lines.
main, sub	overall and sub title for the plot.
xlab	a label for the x axis.
ylab	a label for the y axis.
xlim	limits for the x axis.
ylim	limits for the y axis.
xpd	logical. Should bars be allowed to go outside region?
log	string specifying if axis scales should be logarithmic; see <a href="#">plot.default</a> .
axes	logical. If TRUE, a vertical (or horizontal, if horiz is true) axis is drawn.
axisnames	logical. If TRUE, and if there are names.arg (see above), the other axis is drawn (with lty=0) and labeled.

<code>cex.axis</code>	expansion factor for numeric axis labels.
<code>cex.names</code>	expansion factor for axis names (bar labels).
<code>inside</code>	logical. If TRUE, the lines which divide adjacent (non-stacked!) bars will be drawn. Only applies when <code>space = 0</code> (which it partly is when <code>beside = TRUE</code> ).
<code>plot</code>	logical. If FALSE, nothing is plotted.
<code>axis.lty</code>	the graphics parameter <code>lty</code> applied to the axis and tick marks of the categorical (default horizontal) axis. Note that by default the axis is suppressed.
<code>offset</code>	a vector indicating how much the bars should be shifted relative to the x axis.
<code>add</code>	logical specifying if bars should be added to an already existing plot; defaults to FALSE.
<code>...</code>	arguments to be passed to/from other methods. For the default method these can include further arguments (such as <code>axes</code> , <code>asp</code> and <code>main</code> ) and graphical parameters (see <code>par</code> ) which are passed to <code>plot.window()</code> , <code>title()</code> and <code>axis</code> .

### Details

Each category has its own colour which is shaded inside the category according to the values of the categories of the other variable.

### Author(s)

Sebastien Le <sebastien.le@agrocampus-ouest.fr>  
 Jerome Bouche <jerome.bouche@yahoo.fr>  
 Marine Cadoret <marine.cadoret@agrocampus-ouest.fr>  
 Gwenaelle Fournier <gwenaelle.fournier@gmail.com>  
 Olivier Fournier <olivier.fournier76@gmail.com>  
 Francois Le Poder <francois.lepoder@gmail.com>

### See Also

[ENbarplot](#) which does barplots for categorical variables.

### Examples

```
## Not run:
data(tea)
XvsYbarplot(tea[,15],tea[,20],tea, legend.text=TRUE)
XvsYbarplot("socio.professional.category","sex",tea, legend.text=TRUE)

## End(Not run)
```

# Index

\*Topic **package**

EnquireR-package, [2](#)

\*Topic **univar**

ENbarplot, [4](#)

chisq.desc, [2](#)

ENbarplot, [4](#), [14](#)

ENDensity, [5](#)

ENellipse, [6](#)

ENlisib, [7](#)

ENmarking, [8](#)

ENMCA, [9](#)

EnquireR (EnquireR-package), [2](#)

EnquireR-package, [2](#)

GRmarking, [8](#), [10](#)

p\_inertia, [11](#)

plot.default, [13](#)

tea, [12](#)

XvsYbarplot, [4](#), [12](#)