

# Package ‘LabRS’

October 12, 2022

**Type** Package

**Title** Laboratorio di “Ricerca Sociale con R”

**Version** 0.1.0

**Author** Agnese Vardanega

**Maintainer** Agnese Vardanega <avardanega@unite.it>

**URL** <https://www.agnesevardanega.eu/>

**Depends** knitr

**Suggests** Rcmdr

**Description** Dati, scripts e funzioni per il libro “Ricerca sociale con R. Concetti e funzioni base per la ricerca sociale” (Datasets, scripts and functions to support the book “Ricerca sociale con R. Concetti e funzioni base per la ricerca sociale”).

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-12-18 09:10:09 UTC

## R topics documented:

|                     |   |
|---------------------|---|
| devianza . . . . .  | 2 |
| esm . . . . .       | 2 |
| expchart . . . . .  | 3 |
| expdata . . . . .   | 4 |
| exptab . . . . .    | 4 |
| faccende . . . . .  | 5 |
| frequenze . . . . . | 6 |
| iqr . . . . .       | 6 |
| kabbit . . . . .    | 7 |
| med . . . . .       | 7 |
| MYSLID . . . . .    | 8 |

|                   |    |
|-------------------|----|
| nmiss . . . . .   | 9  |
| nval . . . . .    | 9  |
| percent . . . . . | 10 |
| tabcont . . . . . | 10 |
| tabfreq . . . . . | 11 |
| tabM . . . . .    | 12 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>13</b> |
|--------------|-----------|

---

|          |                 |
|----------|-----------------|
| devianza | <i>Devianza</i> |
|----------|-----------------|

---

### Description

Calcola la devianza come `cor * nval - 1`. Sono considerati i soli casi validi.

### Usage

```
devianza(x, ...)
```

### Arguments

|     |   |
|-----|---|
| x   | vettore                                       |
| ... | altri argomenti ereditati da <code>cor</code> |

### Value

Valore della devianza

---

|     |                                      |
|-----|--------------------------------------|
| esm | <i>Errore standard (della media)</i> |
|-----|--------------------------------------|

---

### Description

Calcola l'errore standard dalla media, e gli estremi dell'intervallo di confidenza. Di default, considera solo i casi validi con un intervallo di confidenza del 95%.

### Usage

```
esm(x, q = 1.96, digits = 6)
```

### Arguments

|        |  |
|--------|--|
| x      | vettore  |
| q      | quantile (o qualunque valore) della distribuzione normale default = 1.96 (int. conf 95%); per un intervallo di confidenza del 99% impostarlo a 2.58. |
| digits | decimali   |

**Value**

Valore della media, della deviazione standard, dell'errore standard, estremi dell'intervallo di confidenza, valore p dell'area.

**Examples**

```
esm(cars$speed)
```

---

expchart

*Esporta i grafici*

---

**Description**

Esportazione semplificata di immagini non vettoriali dei grafici con [dev.copy](#). I valori di default sono impostati in modo da esportare una immagine png con la risoluzione e le dimensioni prodotte da R di default.

**Usage**

```
expchart(device = png, file, ppi = 72, hprop = 6.7, wprop = 6.7,  
...)
```

**Arguments**

|        |   |
|--------|---|
| device | default = png; formato immagine (png, jpeg, tiff, bmp; non pdf e svg) vedi <a href="#">dev.copy</a> |
| file   | nome del file di esportazione (con estensione in base al tipo di immagine - device)                 |
| ppi    | default = 72; risoluzione dell'immagine   |
| hprop  | default = 6.7; altezza dell'immagine, in proporzione ai ppi (es: 4)                                 |
| wprop  | default = 6.7; larghezza dell'immagine, in proporzione ai ppi (es: 4)                               |
| ...    | altri argomenti ereditati <a href="#">dev.copy</a>  |

**Value**

file immagine

**Examples**

```
expchart(png, file = "chart.png", ppi = 300, hprop = 4, wprop = 4)
```

---

|         |  |
|---------|--|
| expdata | <i>Esporta un dataset in csv (standard italiano)</i> |
|---------|--|

---

### Description

Esportazione semplificata di un dataframe in formato csv, con `write.table`, e secondo lo standard italiano: `sep = ";"`, `dec = ","` (separatore di decimali = virgola), `NA = ""` (celle vuote), `row.names = FALSE` (il dataframe da esportare ha una colonna / variabile per gli identificativi di caso).

### Usage

```
expdata(x, file, ...)
```

### Arguments

|      |  |
|------|--|
| x    | dataset  |
| file | nome del file di esportazione (con estensione csv) |
| ...  | argomenti ereditati da <code>write.table</code>    |

### Value

file in formato csv

### Examples

```
data("MYSLID")
expdata(MYSLID, file = "MYSLID.csv")
```

---

|        |  |
|--------|--|
| exptab | <i>Esporta una o piu' tabelle in un unico file csv (standard italiano)</i> |
|--------|--|

---

### Description

Esportazione semplificata di più tabelle in formato csv in uno stesso file con `write.table`, e secondo lo standard Excel in lingua italiana: `sep = ";"`, `dec = ","` (separatore di decimali = virgola), `NA = ""` (celle vuote), `row.names = TRUE`, `col.names = NA` (per le colonne nel file csv).

### Usage

```
exptab(tab, file, dids = names(tab), aggiungi = FALSE, ...)
```

**Arguments**

|          |  |
|----------|--|
| tab      | lista degli oggetti (table); se la lista è con nomi, questi saranno utilizzati come didascalie                   |
| file     | nome del file di esportazione (con estensione csv)   |
| dids     | vettore carattere con le didascalie delle tabelle; per default, è costituito dal nome degli elementi della lista |
| aggiungi | aggiunge le tabelle ad un file esistente (come append), logico   |
| ...      | altri argomenti ereditati <a href="#">write.table</a>  |

**Value**

file in formato csv con le tabelle

**Examples**

```
data("MYSLID")
tab1 = tabfreq(MYSLID$Genere)
tab2 = tabcont(MYSLID$Lingua, MYSLID$Genere)
# una tabella
exptab(list(tab1), file = "tabelle.csv")
# tabelle diverse con didascalie
exptab(list(tab1, tab2), file = "tabelle.csv",
        dids = c("Genere", "Lingua parlata per Genere"),
        aggiungi = TRUE)
```

---

faccende

*Faccende domestiche (dati)*

---

**Description**

Tabella di contingenza, tratta da [housetasks](#). Tradotto in italiano

**Usage**

```
data("faccende")
```

**Format**

Tabella di contingenza con 13 righe e 4 colonne:

Moglie vettore numerico

Alternati vettore numerico

Marito vettore numerico

Insieme vettore numerico

**Source**

dataset tratto da factoextra, e prima da FactoMineR

---

|           |   |
|-----------|---|
| frequenze | <i>Tabelle di frequenza di un dataframe (formato csv)</i> |
|-----------|---|

---

### Description

Produce le tabelle delle frequenze per tutte le variabili categoriali (fattori) di un file, con `tabfreq`, e le esporta in formato csv con `exptab`.

### Usage

```
frequenze(x, write = TRUE, file, aggiungi = FALSE, ...)
```

### Arguments

|          |  |
|----------|--|
| x        | dataframe  |
| write    | logico; esporta o no le frequenze in un file csv                                 |
| file     | (se write = TRUE) carattere; nome del file di esportazione (con estensione csv)  |
| aggiungi | (se write = TRUE) logico; aggiunge le tabelle ad un file esistente (come append) |
| ...      | argomenti ereditati da <code>tabfreq</code>                                      |

### Value

file in formato csv con le tabelle, o lista con le tabelle

### Examples

```
data("MYSLID")
frequenze(MYSLID, "frequenze.csv")
```

---

|     |  |
|-----|--|
| iqr | <i>Differenza interquartile (fattori ordinati)</i> |
|-----|--|

---

### Description

Calcola la differenza interquartile anche per i fattori ordinati. Per default, considera solo i casi validi.

### Usage

```
iqr(x, na.rm = TRUE, ...)
```

### Arguments

|       |   |
|-------|---|
| x     | vettore o fattore ordinato                            |
| na.rm | valore logico (TRUE o FALSE) (vedi <code>IQR</code> ) |
| ...   | altri argomenti ereditati da <code>IQR</code>         |

**Value**

quantili (valori o livelli)

**Examples**

```
iqr(Orange$Tree) # fattore ordinato
iqr(cars$speed) # vettore numerico
```

---

kabbit

*Tabella 'kable' con formato dei numeri in italiano*

---

**Description**

Genera tabelle con standard dei numeri in lingua italiana, con la funzione `kable` di `knitr`: separatore di decimali = virgola, separatore di migliaia = punto

**Usage**

```
kabbit(x, ...)
```

**Arguments**

`x`                    oggetto table o xtabs, matrice, o dataframe  
`...`                altri argomenti ereditati da [kable](#)

**Value**

vettore carattere del codice della tabella

**Examples**

```
data("MYSLID")
kabbit(table(MYSLID$Lingua))
```

---

med

*Mediana (fattori ordinati)*

---

**Description**

Calcola la mediana anche per i fattori ordinati. Tratta dalla funzione [Me](#) del pacchetto `labstatR`. Per default, considera solo i casi validi.

**Usage**

```
med(x, na.rm = TRUE, ...)
```

**Arguments**

x                    vettore o fattore ordinato  
na.rm                valore logico (TRUE o FALSE) (vedi [median](#))  
...                    altri argomenti ereditati da [median](#)

**Value**

mediana (valore o livello)

**References**

Stefano M.Iacus and Guido Masarotto (2018). *labstatR: Libreria Del Laboratorio Di Statistica Con R*. R package version 1.0.9.

**Examples**

```
med(Orange$Tree) # fattore ordinato  
med(cars$speed) # vettore numerico
```

---

MYSLID

*MYSLID (dati)*

---

**Description**

Dataset adattato da [SLID](#). Tradotto e ricodificato

**Usage**

```
data("MYSLID")
```

**Format**

Data frame con 7425 casi e 7 variabili.

ID    Identificativo di caso

Retribuzione    Retribuzione oraria (indice composito)

Istruzione    Numero di anni di scuola

Eta    In anni

Genere    Fattore con i livelli Donna Uomo

Lingua    Fattore con i livelli Inglese Francese Altro

Eta.classi    Fattore ordinato con i livelli Fino a 25 < 26-35 < 36-45 < 46-55 < 55-65 < Oltre  
65

**Source**

Dataset tratto da carData



**Examples**

```
data(MYSLID)
```

---

|       |                      |
|-------|----------------------|
| nmiss | <i>Casi mancanti</i> |
|-------|----------------------|

---

**Description**

Numero di casi mancanti in un oggetto.

**Usage**

```
nmiss(x)
```

**Arguments**

x                    oggetto (vettore, matrice, dataframe ...)

**Value**

Numero di casi mancanti

**Examples**

```
nmiss(cars)  
nmiss(ChickWeight$Diet)
```

---

|      |                    |
|------|--------------------|
| nval | <i>Casi validi</i> |
|------|--------------------|

---

**Description**

Numero di casi validi in un oggetto.

**Usage**

```
nval(x)
```

**Arguments**

x                    oggetto (vettore, matrice, dataframe ...)

**Value**

Numero di casi validi

**Examples**

```
nval(cars)
nval(ChickWeight$Diet)
```

---

percent

*Percentuali*

---

**Description**

Calcola le percentuali, con arrotondamento (se x = dataframe -> as.matrix, se x = double -> as.array).

**Usage**

```
percent(x, digits = 8, margin = NULL)
```

**Arguments**

x                   array (dataframe, numeric).  
 digits             numero dei decimali (vedi [round](#))  
 margin             marginale (vedi [prop.table](#))

**Value**

matrice dei valori percentuali

**Examples**

```
percent(table(ChickWeight$Diet), 2)
percent(HairEyeColor, 1, margin = c(3,2))
```

---

tabcont

*Tabelle di contingenza*

---

**Description**

Produce una tabella di contingenza per la presentazione dei dati, in valori percentuali di colonna (un decimale di default), totali di colonna in VA opzionali

**Usage**

```
tabcont(x, y, digits = 1, margin = 2, totali = TRUE)
```

**Arguments**

|        |  |
|--------|--|
| x      | fattore: variabile di riga                           |
| y      | fattore: variabile di colonna                        |
| digits | numero dei decimali (vedi <a href="#">round</a> )    |
| margin | margin marginale (vedi <a href="#">prop. table</a> ) |
| totali | logico: TRUE = con totali di colonna in VA           |

**Value**

una matrice con la tabella

**Examples**

```
data("MYSLID")
tabcont(MYSLID$Lingua, MYSLID$Genere)
tabcont(MYSLID$Lingua, MYSLID$Genere, totali = FALSE)
```

---

|         |                             |
|---------|-----------------------------|
| tabfreq | <i>Tabelle di frequenza</i> |
|---------|-----------------------------|

---

**Description**

Produce una tabella delle frequenze per la presentazione dei dati, in valori assoluti e percentuali (due decimali di default)

**Usage**

```
tabfreq(x, digits = 2, totali = TRUE)
```

**Arguments**

|        |   |
|--------|---|
| x      | fattore   |
| digits | numero dei decimali (vedi <a href="#">round</a> ) |
| totali | logico: TRUE = con totale e casi mancanti         |

**Value**

una matrice con due colonne

**Examples**

```
data("MYSLID")
tabfreq(MYSLID$Lingua)
```

---

|      |                           |
|------|---------------------------|
| tabM | <i>Tabella "M" (dati)</i> |
|------|---------------------------|

---

**Description**

Tabella di contingenza (tipo "matrice"), adattata da un esempio nel file [chisq.test](#).

**Usage**

```
data("tabM")
```

**Format**

Frequenze condizionate delle variabili:

x identificazione di partito: vettore numerico

y genere: vettore numerico

**Source**

Tabella tratta da [chisq.test](#)

**References**

Agresti, Alan. 2007. An introduction to categorical data analysis. 2nd ed. Hoboken, NJ: Wiley-Interscience.

# Index

## \* datasets

faccende, 5  
MYSLID, 8  
tabM, 12

chisq.test, 12  
cor, 2

dev.copy, 3  
devianza, 2

esm, 2  
expchart, 3  
expdata, 4  
exptab, 4, 6

faccende, 5  
frequenze, 6

housetasks, 5

IQR, 6  
iqr, 6

kabbit, 7  
kable, 7

Me, 7  
med, 7  
median, 8  
MYSLID, 8

nmiss, 9  
nval, 2, 9

percent, 10  
prop.table, 10, 11

round, 10, 11

SLID, 8

tabcont, 10  
tabfreq, 6, 11  
tabM, 12

write.table, 4, 5