

Package ‘labstatR’

April 12, 2018

Version 1.0.9

Date 2018-04-12

Title Libreria Del Laboratorio Di Statistica Con R

Author Stefano M.Iacus <stefano.iacus@unimi.it> and Guido Masarotto
<guido@sirio.stat.unipd.it>

Maintainer Stefano M.Iacus <stefano.iacus@unimi.it>

Description Insieme di funzioni di supporto al volume ``Laboratorio di Statistica con R'', Iacus-Masarotto, MacGraw-Hill Italia, 2006.
This package contains sets of functions defined in ``Laboratorio di Statistica con R'', Iacus-Masarotto, MacGraw-Hill Italia, 2006. Function names and docs are in italian as well.

License GPL (>= 2)

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-12 14:00:43 UTC

R topics documented:

birthday	2
bubbleplot	3
chi2	3
COV	4
cv	5
E	5
eta	6
gen.vc	6
gini	7
gioco1	8
gioco2	8
histpf	9
ic.var	10

interinale	10
kurt	11
lewis	12
Markov	12
Me	13
meana	14
meang	15
Rp	15
Rpa	16
sigma2	17
skew	18
test.var	18
trajectory	19
Index	21

birthday	<i>Calcola la probabilita' di compleanni coincidenti</i>
----------	--

Description

Questa funzione risolve il problema del calcolo della probabilita' di trovare due persone in un gruppo di n nate lo stesso giorno.

Usage

```
birthday(n)
```

Arguments

n numero di persone nel gruppo

See Also

[pbirthday.](#)

Examples

```
n <- c(5,10,15,20,21,22,23,24,25,30,50,60,
      70,80,90,100,200,300,365)
for(i in n)
  cat("\n n=",i,"P(A)=",birthday(i))
```

bubbleplot*Disegna un grafico a bolle*

Description

Questa funzione disegna un grafico a bolle (bubbleplot) a partire da una tabella a doppia entrata.

Usage

```
bubbleplot(tab, joint = TRUE, magnify = 1,  
           filled = TRUE, main = "bubble plot")
```

Arguments

tab	tabella di contingenza a due vie
joint	valore logico. Se TRUE disegna la distribuzione di frequenza congiunta altrimenti la distribuzione condizionata per riga
magnify	parametro per il controllo dell'ampiezza delle bolle
filled	valore logico. Se TRUE riempie di colore le bolle
main	titolo del grafico

Examples

```
x <- c("O", "O", "S", "B", "S", "O", "B", "B", "S",  
      "B", "O", "B", "B", "O", "S")  
y <- c("O", "B", "B", "B", "S", "S", "O", "O", "B",  
      "B", "O", "S", "B", "S", "B")  
x <- ordered(x, levels=c("S", "B", "O"))  
y <- ordered(y, levels=c("S", "B", "O"))  
table(x,y)  
bubbleplot(table(x,y),main="Musica versus Pittura")
```

chi2*Calcola l'indice di connessione*

Description

Questa funzione permette il calcolo dell'indice di connessione.

Usage

```
chi2(x,y)
```

Arguments

x	vettore di dati
y	vettore di dati

Examples

```
x <- rbinom(8,5,0.5)
y <- c("A", "A", "B", "A", "B", "B", "C", "B")
chi2(x,y)
```

COV

Calcola la covarianza non corretta

Description

Questa funzione permette il calcolo della covarianza non corretta.

Usage

```
COV(x,y)
```

Arguments

x	vettore di dati
y	vettore di dati

Details

La funzione cov di R effettua il calcolo della varianza campionaria, ovvero divide la codevianza per il numero di dati meno uno. Questa funzione invece divide la codevianza per il numero di dati a disposizione.

See Also

[cov](#).

Examples

```
x <- c(1,3,2,4,6,7)
y <- c(7,3,2,1,-1,-3)
cov(x,y)
COV(x,y)
```

`cv`*Calcola il coefficiente di variazione*

Description

Questa funzione permette il calcolo del coefficiente di variazione.

Usage`cv(x)`**Arguments**

`x` vettore di dati

Examples

```
x <- c(1, 3, 2, 4, 6, 7)
cv(x)
```

`E`*Calcola l'indice di eterogeneita'*

Description

Questa funzione permette il calcolo dell'indice di eterogeneita' di Gini.

Usage`E(x)`**Arguments**

`x` vettore di dati

See Also

[var.](#)

Examples

```
x <- c("A", "A", "B", "A", "C", "A")
E(x)
```

`eta`*Calcola l'indice di dipendenza in media*

Description

Questa funzione permette il calcolo dell'indice di dipendenza in media e traccia il grafico della funzione di regressione.

Usage`eta(x,y)`**Arguments**

`x` vettore di dati eventualmente qualitativo
`y` vettore di dati numerico

Details

Questa funzione considera la dipendenza in media di `y` da `x`.

Examples

```
x <- c(rep(1,10),rep(0,23), rep(2,15))
y <- c(rnorm(10,mean=7),rnorm(23,mean=19),rnorm(15,mean=17))
eta(x,y)
y <- c(rnorm(10,mean=8),rnorm(23,mean=7),rnorm(15,mean=6.5))
eta(x,y)
```

`gen.vc`*Simula una variabile casuale discreta*

Description

Questa funzione permette di simulare un valore da una variabile casuale discreta con distribuzione di probabilita' assegnata.

Usage`gen.vc(x,p)`**Arguments**

`x` valori assumibili dalla variabile casuale
`p` distribuzione di probabilita'

Details

La funzione restituisce un numero casuale.

Examples

```
x <- c(-2,3,7,10,12)
p <- c(0.2, 0.1, 0.4, 0.2, 0.1)
y <- NULL
for(i in 1:1000) y <- c(y,gen.vc(x,p))
table(y)/length(y)
```

gini	<i>Calcola l'indice di concentrazione</i>
------	---

Description

Questa funzione permette il calcolo l'indice di concentrazione e il rapporto di concentrazione di Gini. Inoltre disegna la curva di Lorenz.

Usage

```
gini(x, plot=TRUE, add=FALSE, col="black")
```

Arguments

x	vettore di dati
plot	valore logico. Se TRUE disegna la curva di Lorenz
add	valore logico. Se TRUE disegna una nuova curva di Lorenz sul precedente grafico della curva di concentrazione
col	colore con cui disegnare l'area di concentrazione

Examples

```
x <- c(1,3,4,30,100)
gini(x)
y <- c(10,10,10,10)
gini(y, add=TRUE,col="red")
```

gioco1 *Simula la scommessa di De Mere*

Description

Questa funzione simula la scommessa di de Mere calcolando la probabilita' di fare almeno un 6 in 4 lanci di un dado regolare.

Usage

```
gioco1(prove=10000)
gioco1a(prove=10000)
```

Arguments

prove numero di prove da utilizzare nella simulazione

Details

La versione `gioco1` della funzione non e' efficiente in termini di velocita' in quanto vengono impiegati cicli `for`. Si noti la differenza in termini di velocita' con la version `gioco1a`.

See Also

[gioco2](#).

Examples

```
ptm <- proc.time()
gioco1a(10000)
proc.time() - ptm
ptm <- proc.time()
gioco1(10000)
proc.time() - ptm
```

gioco2 *Simula la scommessa di De Mere*

Description

Questa funzione simula la scommessa di de Mere calcolando la probabilita' di fare almeno un doppio 6 in 24 lanci di un dado regolare.

Usage

```
gioco2(prove=10000)
gioco2a(prove=10000)
```


Arguments

prove numero di prove da utilizzare nella simulazione

Details

La versione `gioco2` della funzione non e' efficiente in termini di velocita' in quanto vengono impiegati cicli `for`. Si noti la differenza in termini di velocita' con la version `gioco2a`.

See Also

[gioco1](#).

Examples

```
ptm <- proc.time()
gioco2a(10000)
proc.time() - ptm
ptm <- proc.time()
gioco2(10000)
proc.time() - ptm
```

histpf

Disegna il poligono di frequenza

Description

Questa funzione disegna l'istogramma e vi sovrappone il corrispondente poligono di frequenza.

Usage

```
histpf(x, br, ...)
```

Arguments

x vettore di dati
br numero di intervalli, metodo di scelta degli intervalli o vettore di estremi degli intervalli
... argomenti da passare alla funzione `hist`

Details

Il parametro `br` si comporta esattamente come il parametro `breaks` della funzione `hist`.

See Also

[hist](#).

Examples

```
x <- rnorm(50)
histpf(x,br=5)
```

`ic.var`*Calcola intervallo di confidenza per la varianza*

Description

Questa funzione effettua il calcolo dell'intervallo di confidenza per la varianza di campione gaussiano.

Usage

```
ic.var(x, twosides = TRUE, conf.level = 0.95)
```

Arguments

<code>x</code>	vettore di dati
<code>twosides</code>	logico. Se FALSE l'estremo inferiore e' posto pari a 0
<code>conf.level</code>	livello confidenza

Examples

```
x <- c(0.39, 0.68, 0.82, 1.35, 1.38, 1.62, 1.70,
      1.71, 1.85, 2.14, 2.89, 3.69)
ic.var(x)
ic.var(x,FALSE)
```

`interinale`*Dati sul lavoro interinale*

Description

Si tratta di un campione di dati relativi agli iscritti ed avviati alle missioni di una societa' di fornitura di lavoro interinale.

Usage

```
data(interinale)
```

Format

Dataset tratti dall'archivio di una societa' interinale italiana.

Source

Iacus, S.M., Porro, G. (2001)

References

Iacus, S.M., Porro, G. (2001) Occupazione interinale e terzo settore. Analisi dei microdati di una società "no profit" di fornitura di lavoro interinale, IRES Quaderno n.2, *IRES-Lombardia*, www.ireslombardia.it.

Examples

```
data(interinale)
glm(avviato~., binomial, data=interinale) -> model
model
pr <- predict(model, type="response")
plot(density(pr),xlim=c(0,0.2),main="")
```

kurt

Calcola l'indice di curtosi

Description

Questa funzione permette il calcolo dell'indice di curtosi.

Usage

```
kurt(x)
```

Arguments

x vettore di dati

See Also

[skew](#).

Examples

```
x <- rnorm(50)
kurt(x)
y <- rt(50,df=1)
kurt(y)
```

`lewis`*Simulatore di processi di Poisson*

Description

Questa funzione simula un processo di Poisson non omogeneo.

Usage

```
lewis(T, lambda, plot.int = TRUE)
```

Arguments

<code>T</code>	orizzonte temporale
<code>lambda</code>	funzione di intensita'
<code>plot.int</code>	se TRUE traccia il grafico della funzione di intensita' oltre alla traiettoria del processo

Details

Disegna una traiettoria di un processo di Poisson non omogeneo con funzione di intensita' `lambda` (che deve essere una funzione di una variabile) nell'intervallo $(0,T)$.

See Also

[gen.vc](#), [Markov](#), [trajectory](#).

Examples

```
lewis(20, sin)
```

`Markov`*Simulatore di catene di Markov*

Description

Questa funzione simula una catena di Markov a stati finiti.

Usage

```
Markov(x0, n, x, P)
```

Arguments

x0	stato iniziale
n	lunghezza della traiettoria
x	insieme degli stati
P	matrice di probabilita' di transizione

Details

La funzione Markov2 e' basata sulla funzione [sample](#).

Value

Una lista contenente la traiettoria della catena di Markov:

X	valori assunti dalla catena di Markov
t	tempi

See Also

[gen.vc](#), [trajectory](#), [lewis](#).

Examples

```
x <- c("P", "S", "N")
P <- matrix(c(0.5, 0.5, 0.25, 0.25, 0, 0.25, 0.25, 0.5, 0.5), 3, 3)
Markov("S", 15, x, P) -> traj
traj
plot(traj$t, unclass(factor(traj$X)), type="s", axes=FALSE,
     xlab="t", ylab="Che tempo fa'")
axis(1)
axis(2, c(1, 2, 3), levels(factor(traj$X)))
box()
```

Description

Questa funzione permette il calcolo della mediana anche nel caso di fenomeni qualitativi ordinabili.

Usage

```
Me(x)
```

Arguments

x	vettore di dati
---	-----------------

Details

La funzione `median` di R contenuta nel pacchetto base funziona solo per dati quantitativi. La funzione `Me` restituisce un messaggio d'errore se la mediana risulta indeterminata.

See Also

[median](#).

Examples

```
x <- factor(c("A", "B", "A", "C", "A"))
Me(x)
```

meana

Calcola la media armonica

Description

Questa funzione permette il calcolo della media armonica.

Usage

```
meana(x, ...)
```

Arguments

<code>x</code>	vettore di dati
<code>...</code>	parametri aggiuntivi (ignorati in questa versione)

See Also

[meang](#).

Examples

```
x <- c(1,3,2,4,6,7)
meana(x)
```

meang	<i>Calcola la media geometrica</i>
-------	------------------------------------

Description

Questa funzione permette il calcolo della media geometrica.

Usage

```
meang(x, ...)
```

Arguments

x	vettore di dati
...	parametri aggiuntivi (ignorati in questa versione)

See Also

[meana](#).

Examples

```
x <- c(1, 3, 2, 4, 6, 7)
meang(x)
```

Rp	<i>Calcola l'allocazione ottimale di un portafoglio</i>
----	---

Description

Questa funzione permette il calcolo dell'allocazione ottimale di due titoli di un portafoglio.

Usage

```
Rp(x, y, pxy)
```

Arguments

x	rendimenti del primo titolo
y	rendimenti del secondo titolo
pxy	distribuzione doppia dei due titoli

Details

La funzione restituisce rendimento medio e varianza attesa del portafoglio allocato in modo ottimo. Restituisce inoltre il valore ottimo di capitale da allocare nel primo titolo.

Value

Una lista contenente media e varianza del rendimento del portafoglio:

a quota ottimale da allocare nel primo titolo
Rm rendimento medio.
VR varianza del portafoglio.

See Also

[Rpa.](#)

Examples

```
x <- c(11,9,25,7,-2)/100
y <- c(-3,15,2,20,6)/100
pxy <- matrix(rep(0,25),5,5)
pxy[1,1] <- 0.2
pxy[2,2] <- 0.2
pxy[3,3] <- 0.2
pxy[4,4] <- 0.2
pxy[5,5] <- 0.2
Rp(x,y,pxy)
```

Rpa

Calcola il rendimento di un portafoglio

Description

Questa funzione permette il calcolo del rendimento atteso di un portafoglio di due titoli al variare della quantità allocata nei due titoli.

Usage

```
Rpa(a, x, y, pxy)
```

Arguments

a percentuale allocata al primo titolo
x rendimenti del primo titolo
y rendimenti del secondo titolo
pxy distribuzione doppia dei due titoli

Details

La funzione restituisce rendimento medio e varianza attesa del portafoglio.

Value

Una lista contenente media e varianza del rendimento del portafoglio:

Rm rendimento medio.
VR varianza del portafolio.

See Also

[Rp.](#)

Examples

```
x <- c(11,9,25,7,-2)/100
y <- c(-3,15,2,20,6)/100
pxy <- matrix(rep(0,25),5,5)
pxy[1,1] <- 0.2
pxy[2,2] <- 0.2
pxy[3,3] <- 0.2
pxy[4,4] <- 0.2
pxy[5,5] <- 0.2
Rpa(0.1,x,y,pxy)
Rpa(0.5,x,y,pxy)
```

sigma2

Calcola la varianza non corretta

Description

Questa funzione calcola la varianza non corretta.

Usage

```
sigma2(x)
```

Arguments

x vettore di dati

Details

La funzione var di R calcola la varianza campionaria corretta, ovvero dividendo la devianza per il numero di elementi del campione meno uno. Questa funzione calcola la varianza dividendo la stessa quantità per il numero totale di osservazioni.

Examples

```
x <- rnorm(10)
var(x)
sigma2(x)
```

skew

Calcola l'indice di asimmetria

Description

Questa funzione permette il calcolo dell'indice di asimmetria.

Usage

```
skew(x)
```

Arguments

x vettore di dati

See Also

[kurt.](#)

Examples

```
x <- rnorm(50)
skew(x)
y <- rchisq(50,df=1)
skew(y)
```

test.var

Calcola intervallo di confidenza per la varianza

Description

Questa funzione effettua il calcolo dell'intervallo di confidenza per la varianza di campione gaussiano.

Usage

```
test.var(x, var0, alternative = "greater", alpha = 0.05)
```

Arguments

x vettore di dati
var0 valore della varianza sotto l'ipotesi nulla
alternative direzione del test "greater" o "less". Default = "greater"
alpha ampiezza del test

Examples

```
x <- rnorm(100, sd=5)
var(x)
test.var(x,20)
test.var(x,20,alternative="less")
```

trajectory

Simulatore di processi di diffusione

Description

Questa funzione simula un processo di diffusione.

Usage

```
trajectory(x0=1, t0=0, T=1, a, b, n=100)
```

Arguments

x0	stato iniziale
t0	istante iniziale
T	istante finale
a	coefficiente di deriva
b	coefficiente di diffusione
n	numero di valori in cui suddividere l'intervallo (t0,T)

Details

I due coefficienti di deriva e diffusione devono essere funzioni di due variabili x e t. La funzione utilizza lo schema di Eulero quindi il processo da simulare deve rispettare le opportune ipotesi sui coefficienti dell'equazione differenziale stocastica.

Value

Una lista contenente la traiettoria del processo di diffusione:

t	vettore dei tempi
y	valori assunti dalla traiettoria

See Also

[gen.vc](#), [Markov](#), [lewis](#).

Examples

```
n <- 100
T <- 1
x0 <- 1
mu <- function(x,t) {-x*t}
sigma <- function(x,t) {x*t}
diff <- trajectory(1,0,1,mu,sigma,100)
plot(diff$t,diff$y,type="l")
acf(diff$y, main="Processo di diffusione")
```

Index

*Topic **datasets**

interinale, [10](#)

*Topic **distribution**

birthday, [2](#)

gen.vc, [6](#)

gioco1, [8](#)

gioco2, [8](#)

lewis, [12](#)

Markov, [12](#)

Rp, [15](#)

Rpa, [16](#)

trajectory, [19](#)

*Topic **multivariate**

bubbleplot, [3](#)

chi2, [3](#)

COV, [4](#)

eta, [6](#)

*Topic **univar**

cv, [5](#)

E, [5](#)

gini, [7](#)

histpf, [9](#)

ic.var, [10](#)

kurt, [11](#)

Me, [13](#)

meana, [14](#)

meang, [15](#)

sigma2, [17](#)

skew, [18](#)

test.var, [18](#)

birthday, [2](#)

bubbleplot, [3](#)

chi2, [3](#)

COV, [4](#)

cov, [4](#)

cv, [5](#)

E, [5](#)

eta, [6](#)

gen.vc, [6](#), [12](#), [13](#), [19](#)

gini, [7](#)

gioco1, [8](#), [9](#)

gioco1a (gioco1), [8](#)

gioco2, [8](#), [8](#)

gioco2a (gioco2), [8](#)

hist, [9](#)

histpf, [9](#)

ic.var, [10](#)

interinale, [10](#)

kurt, [11](#), [18](#)

lewis, [12](#), [13](#), [19](#)

Markov, [12](#), [12](#), [19](#)

Markov2 (Markov), [12](#)

Me, [13](#)

meana, [14](#), [15](#)

meang, [14](#), [15](#)

median, [14](#)

pbirthday, [2](#)

Rp, [15](#), [17](#)

Rpa, [16](#), [16](#)

sample, [13](#)

sigma2, [17](#)

skew, [11](#), [18](#)

test.var, [18](#)

trajectory, [12](#), [13](#), [19](#)

var, [5](#)