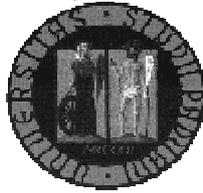


**Università degli Studi di Padova**  
**Facoltà di Scienze Statistiche**

**Corso di Laurea in Statistica e Informatica**



**Tesi di Laurea**

**COSTRUZIONE DELLE LIBRERIE  
TRV E UIT**

Relatore:  
Ch.mo Prof. Giancarlo Diana

Laureando: Michele Battocchio

Anno Accademico 2006-2007

# INDICE

<b>INTRODUZIONE</b>	5
CAPITOLO 1	
<b><u>Test Rapporto di verosimiglianza (TRV)</u></b>	7
1.1 Definizione del TRV	7
1.2 Esempio del TRV	10
1.3 Correzione di Bartlett analitica	12
1.4 Correzione di Bartlett empirica	13
1.5 Stima della distribuzione del TRV tramite simulazione	16
1.6 Analisi della bontà dei metodi	20
CAPITOLO 2	
<b><u>Test Unione-Intersezione (UIT)</u></b>	32
2.1 Definizione del UIT	32
2.2 Esempio del UIT	35
2.3 Stima della distribuzione del UIT tramite simulazione	37

## CAPITOLO 3

<b><u>La funzione test</u></b>	39
3.1 Introduzione al compilatore R	39
3.2 Algoritmo, che stima la distribuzione del TRV	42
3.3 Algoritmo, che stima la distribuzione del UIT	44

## CAPITOLO 4

<b><u>Analisi statistica delle simulazioni</u></b>	45
4.1 Introduzione	45
4.2 Generatori di numeri pseudo-casuali uniformi	46
4.3 Generatori di numeri pseudo-casuali non uniformi	49
4.3.1 Variabile normale standard	50
4.3.2 Variabile normale univariata	52
4.3.3 Variabile normale multivariata	52
4.4 Analisi delle simulazioni effettuate con il generatore mvnorm	54
4.4.1 Confronto tra la stima dell' $\alpha$ nominale con l' $\alpha$ effettivo	54
4.4.2 Test di Smirnov-Cramèr-Von Mises	57
4.4.3 Qualità dei dati uniformi	59
4.4.4 Qualità dei dati normali univariati	60
4.4.5 Qualità dei dati normali multivariati	63
4.4.6 Qualità delle simulazioni	67
4.4.7 Tempi di esecuzione	70
4.4.8 Conclusioni	72

## CAPITOLO 5

<b><u>Un'applicazione pratica</u></b>	73
5.1 I Packages TRV e UIT	73
5.2 Applicazione della libreria TRV sul data-set "iris3"	78
5.2.1 Test d'ipotesi che $\mu=\mu_0$ , con $\Sigma$ nota	"TRV1" 78
5.2.2 Test d'ipotesi che $\mu=\mu_0$ , con $\Sigma$ non nota	"TRV2" 81
5.2.3 Test d'ipotesi che $\Sigma=\Sigma_0$ , con $\mu$ non nota	"TRV3" 83
5.2.4 Test d'ipotesi che $\Sigma_1=\dots=\Sigma_g$ , test di Box	"TRV9" 86
5.3 Applicazione della libreria UIT sul data-set "iris3"	90
5.3.1 Test d'ipotesi che $\mu=\mu_0$ , con $\Sigma$ nota	"UIT1" 90
5.3.2 Test d'ipotesi che $\Sigma=\Sigma_0$ , con $\mu$ non nota	"UIT3" 95
5.3.3 Test d'ipotesi che $\mu_1=\dots=\mu_g$ , con $\Sigma_1=\dots=\Sigma_g$	"UIT8" 101
<b>CONCLUSIONI</b>	105
<b>APPENDICE</b>	107
- Package TRV	107
- Package UIT	128
- Listato funzioni del package TRV	150
- Listato funzioni del package UIT	165
- Listato funzioni per l'analisi della bontà dei metodi	179
<b>BIBLIOGRAFIA</b>	188

# INTRODUZIONE

Nell'ambito della statistica multivariata, la verifica d'ipotesi sui parametri della distribuzione normale viene abitualmente condotta con l'impiego di test basati sul rapporto di verosimiglianza (TRV), stante i risultati almeno asintotici disponibili per le distribuzioni delle statistiche test in ipotesi nulla.

Come è noto, così facendo si va incontro a due ordini di problemi:

- i) le approssimazioni che si ottengono usando la distribuzione limite non sono soddisfacenti se non per campioni di grandi dimensioni, a meno che non si utilizzi la correzione di Bartlett, quando nota;
- ii) il rifiuto dell'ipotesi nulla non consente di verificare, con un adeguato controllo dell'errore di prima specie, se il rifiuto stesso sia da imputare a uno o ad alcuni parametri di specifico interesse.

Al secondo problema si può, in linea teorica, dare soluzione ricorrendo ai test unione intersezione (UIT) introdotti da Roy (1953). Purtroppo però, fatti salvi alcuni semplici casi, non si conoscono le distribuzioni delle statistiche test così ottenute. Presubilmente è per questa ragione che gli UIT hanno avuto scarso sviluppo e impiego in questi anni.

Lo scopo del presente lavoro è mostrare come i metodi di simulazione possano costituire un valido strumento per la soluzione dei problemi descritti. L'attuale velocità di calcolo dei PC consente infatti di effettuare simulazioni ad hoc per ogni specifica situazione in tempi del tutto accettabili.

Nel caso i) si possono stimare direttamente quantità di interesse (percentile o p-value) della distribuzione del TRV o calcolare, via simulazione, una correzione di Bartlett empirica per migliorare le approssimazioni ottenibili utilizzando la distribuzione asintotica. Si vedrà come la correzione empirica possa essere addirittura migliore di quella analitica.

Nel caso ii) quando, come avviene spesso, non si conosce la distribuzione dell'UIT, la stessa può essere oggi facilmente simulata con riferimento alla specifica situazione applicativa d'interesse, consentendo l'impiego del test.

Il presente studio nasce dall'esigenza di disporre di due packages di pubblico dominio, che siano modificabili, in linea con le esigenze dettate dalla situazione che l'utente si trova ad analizzare.

In considerazione dell'utilizzo che di tale prodotto potrebbe venir fatto in futuro, l'impostazione seguita è rivolta a facilitare un suo possibile utilizzo didattico.

La tesi presenta il percorso di studio delineato. Il CAPITOLO 1 ha un carattere introduttivo, in cui si vuole illustrare le possibilità (e le problematiche) del TRV. Nel CAPITOLO 2 si presenta il test unione-intersezione, con relativa procedura che consente di calcolare i test UIT mediante simulazione. Il CAPITOLO 3 ha la funzione di collegamento tra la teoria presentata e la realizzazione al calcolatore. Si presentano cioè gli algoritmi che stanno alla base delle funzioni che simulano i test TRV e UIT. Nel CAPITOLO 4 si cerca di valutare la qualità delle simulazioni.

Il CAPITOLO 5 presenta infine qualche esempio specifico su data-set reali.

Infine nelle CONCLUSIONI, si riportano alcune riflessioni finali riguardanti il lavoro svolto.

# CAPITOLO 1

## TEST RAPPORTO DI VEROSIMIGLIANZA (TRV)

### 1.1 Definizione del TRV

I test rapporto di verosimiglianza (TRV), si fondano sulla logica dei test di significatività e portano, pertanto, all'accettazione o al rifiuto di un'ipotesi multivariata nulla ( ${}_M H_0$ ), a seconda che delle condizioni suggerite dai risultati campionari. Più in particolare, sia  $X_{n \times p}$  una matrice di dati campionari da una distribuzione con funzione di probabilità  $f(\underline{x}; \underline{\theta})$  dipendente da un parametro vettoriale  $\underline{\theta}$ . L'interesse è valutare la conformità dei dati ad una certa ipotesi sul vettore di parametri  $\underline{\theta}$ .

Si procede a saggiare il seguente sistema d'ipotesi:

$${}_M H_0: \underline{\theta} \in \Omega_0$$

$${}_M H_1: \underline{\theta} \in \Omega_1$$

dove:

${}_M H_0$  è l'ipotesi nulla

${}_M H_1$  è l'ipotesi alternativa

$\Omega_0$  e  $\Omega_1$  sono spazi parametrici in cui varia  $\underline{\theta}$  sotto  ${}_M H_0$  e  ${}_M H_1$  rispettivamente.

Un possibile modo per condurre la verifica in questione è il seguente.

Il rapporto

$$\lambda(X) = L(\underline{\theta}_0) / L(\hat{\underline{\theta}}) = L^*_0 / L^*_1$$

dove

$$L_i^* = \max_{\theta \in \Omega_i} L(\theta; X)$$

è una statistica-test per verificare  $MH_0$  contro  $MH_1$ .

Equivalentemente si può usare

$$-2\log \lambda(X) = 2(\ell_1^* - \ell_0^*)$$

$$\text{dove: } \ell_i^* = \log L_i^*$$

Sia  $\lambda(X)$  che  $-2\log\lambda(X)$  vengono denominate statistiche test basate sul rapporto di verosimiglianza.

Il rapporto può assumere valori che variano tra  $(0,1]$ . Se  $\lambda \approx 0$  l'ipotesi nulla appare del tutto insoddisfacente, viceversa se  $\lambda=1$ , significa che c'è perfetta corrispondenza dei dati all'ipotesi.

Si sceglie un valore critico  $\lambda_c$ , tale che se  $\lambda(X) < \lambda_c$  rifiuteremo l'ipotesi  $MH_0$ .

La regione di rifiuto della statistica rapporto di verosimiglianza è del tipo:

$$R1 = \{ X \mid \lambda(X) < \lambda_{c1} \}$$

$$\text{ovvero } R2 = \{ X \mid -2\log \lambda(X) > \lambda_{c2} \}$$

dove  $\lambda_{c1}$  e  $\lambda_{c2}$  sono costanti tali che

$$\sup_{\theta \in \Omega_0} \Pr(X \in R_i) = \alpha$$

per  $i= 1,2$  con  $\alpha$  prefissato.

Può verificarsi che si decida a favore di  $MH_1$  anche se è vera  $MH_0$ , ciò costituisce un errore di I tipo. Viceversa, può verificarsi che si decida a favore di  $MH_0$  anche se è vera  $MH_1$ , errore di II tipo.

Di conseguenza  $\Pr(\lambda(X) < \lambda_c \mid {}_M H_0) = \alpha$ , significa che si assegna un valore massimo  $\alpha$  della probabilità di errore di I tipo e si cerca di rendere minima la probabilità di errore di II tipo.

Comunque non approfondiamo questo argomento, per i nostri scopi è sufficiente l'aver enunciato che nel seguito effettueremo dei test di livello preassegnato, che secondo una consuetudine diffusa, viene posto abitualmente pari a 0.05 o a 0.01.

Il valore critico  $\lambda_c$ , è l'opportuno percentile  $1-\alpha$  della distribuzione di  $\lambda$ . Molto spesso però non è possibile la determinazione della distribuzione esatta di  $\lambda$  e quindi si deve ricorrere alla distribuzione asintotica. Il teorema di Wilks, dimostra che se  $n$  (numerosità campionaria) è sufficientemente elevata la verifica dell'ipotesi  ${}_M H_0$  tramite il rapporto di verosimiglianza può condursi ricorrendo alla v.c. Chi-Quadrato.

## TEOREMA

Se  $\Omega_1 \in \mathbb{R}^p$  e  $\Omega_0 \in \mathbb{R}^q \subset \mathbb{R}^p$ , sotto condizioni di regolarità,  $\forall \theta \in \Omega_0$

$$-2 \log \lambda(X) \xrightarrow{d} \chi_{p-q}^2 \quad \text{per } n \rightarrow \infty$$

Le approssimazioni che si ottengono usando la distribuzione limite non sono però soddisfacenti se non per campioni di grandi dimensioni.

## 1.2 Esempio di TRV

Sia  $X_{n \times p}$  una matrice normale di dati da  $N_p(\mu, \Sigma)$  con varianza nota e media ignota nei confronti della quale si formulano le ipotesi

$${}_M H_0: \mu = \mu_0$$

$${}_M H_1: \mu \neq \mu_0$$

che si intende verificare, a livello di significatività  $\alpha$ , ricorrendo al TRV.

Poiché la verosimiglianza risulta

$$L(\mu) = |2\pi\Sigma|^{-n/2} \exp \left\{ -1/2 (\underline{x} - \mu)' \Sigma^{-1} (\underline{x} - \mu) \right\}$$

allora sotto  ${}_M H_0: \mu = \mu_0$

$$\ell_0^* = -n/2 \log |2\pi \Sigma| - n/2 \text{tr} [\Sigma^{-1} S] - n/2 (\bar{\underline{x}} - \mu_0)' \Sigma^{-1} (\bar{\underline{x}} - \mu_0)$$

sotto  ${}_M H_1: \mu \neq \mu_0$

$$\ell_1^* = -n/2 \log |2\pi \Sigma| - n/2 \text{tr} [\Sigma^{-1} S]$$

Il test assume la forma

$$-2\log \lambda = 2 (\ell_1^* - \ell_0^*) = n (\bar{\underline{x}} - \mu_0)' \Sigma^{-1} (\bar{\underline{x}} - \mu_0)$$

La regione di rifiuto di  ${}_M H_0$  risulta

$$R_2 = \left\{ -2\log \lambda \mid -2\log \lambda = n (\bar{\underline{x}} - \mu_0)' \Sigma^{-1} (\bar{\underline{x}} - \mu_0) > \lambda_{c2} \right\}$$

Per assicurare il prefissato livello di significatività  $\alpha$  si può procedere nel modo seguente. Poiché deve essere

$$\alpha = \Pr (-2\log \lambda > \lambda_{c2} \mid {}_M H_0) = \Pr \left\{ n (\bar{\mathbf{x}} - \boldsymbol{\mu}_0)' \boldsymbol{\Sigma}^{-1} (\bar{\mathbf{x}} - \boldsymbol{\mu}_0) > \lambda_{c2} \mid {}_M H_0 \right\}$$

ricordando che in questo caso particolare, sotto  ${}_M H_0$ ,  $-2\log \lambda$  si distribuisce esattamente come un  $\chi_p^2$  allora

$$\Pr (-2\log \lambda > \lambda_{c2} \mid {}_M H_0) = \Pr \left\{ \chi_p^2 > \lambda_{c2} \right\} = \alpha$$

Il valore  $\lambda_{c2}$  viene a coincidere con il valore  $\chi_{p;1-\alpha}^2$ , cioè con il percentile di ordine  $(1 - \alpha)$  della v.c.  $\chi_p^2$  e quindi l'ipotesi  ${}_M H_0$  viene rifiutata se

$$-2\log \lambda = n (\bar{\mathbf{x}} - \boldsymbol{\mu}_0)' \boldsymbol{\Sigma}^{-1} (\bar{\mathbf{x}} - \boldsymbol{\mu}_0) \geq \chi_{p;1-\alpha}^2$$

### 1.3 Correzione di Bartlett analitica

Quando, come nella maggior parte dei casi, non si conosce la distribuzione esatta del test, si ricorre alla distribuzione asintotica. Viene naturale porsi una domanda: la distribuzione asintotica fornisce una buona approssimazione della distribuzione esatta?

Bartlett (1937) propose una correzione al TRV, per saggiare l'omogeneità di  $K$  varianze nel campionamento da popolazioni normali, noto come test di Bartlett. In termini generali, la correzione di Bartlett, che nel seguito chiameremo **correzione di Bartlett analitica (Ba)**, può essere così descritta. Posto che il TRV abbia distribuzione asintotica  $\chi_p^2$ , si cerca quella quantità  $a$  tale che  $a \rightarrow \frac{E(TRV)}{p}$ , per cui l'aggiustamento della statistica  $TRV$  è

$$TRV' = \frac{TRV}{a}$$

sperando che  $TRV'$  abbia una distribuzione più vicina al  $\chi_p^2$  che non  $TRV$ .

La quantità  $a$ , abitualmente non esprimibile analiticamente, può essere ottenuta considerando lo sviluppo di Taylor di  $E(TRV)$ , da cui risulta che

$$E(TRV) = p \left\{ 1 + \frac{b}{n} + O(n^{-2}) \right\}$$

e, quindi, prendendo  $a = 1 + \frac{b}{n}$ , si può considerare la statistica modificata

$$TRV' = \frac{TRV}{1 + \frac{b}{n}}$$

in cui  $b$  risulta essere una funzione esprimibile in termini di momenti delle derivate della log-verosimiglianza.

Lawley(1956) ha mostrato che la correzione di Bartlett, non solo migliora l'approssimazione del valore atteso, ma anche quella dell'intera distribuzione rispetto al  $\chi_p^2$ .

Inoltre Barndoff-Nielsen e Hall (1988) hanno mostrato che  $TRV'$  ha distribuzione Chi-Quadrato con  $p$  gradi di libertà con errore di ordine  $O(n^{-2})$ , anziché di ordine  $O(n^{-1})$ , come accade per  $TRV$ .

Al paragrafo 1.6 verrà valutata l'efficacia della correzione di Ba.

## 1.4 Correzione di Bartlett empirica

Accade talvolta, che sia troppo laborioso determinare la correzione di Bartlett analitica. In questi casi si può pensare di risolvere il problema tramite la simulazione.

Infatti, a livello di principio, è possibile ottenere un'approssimazione accurata quanto si vuole di  $\left\{1 + \frac{b}{n} + O(n^{-2})\right\}$ .

Questa la chiameremo **correzione di Bartlett empirica1 (Be1)**.

A differenza della correzione di Ba, l'idea è quella di stimare tramite simulazione il parametro  $\left\{1 + \frac{b}{n} + O(n^{-2})\right\}$ , che chiameremo  $a_1$ .

La logica di tale simulazione è abbastanza semplice e può essere così esposta:

Dato che

$$a_1 = \frac{E(TRV)}{p}$$

si può stimare  $a_1$  a partire da una stima di  $E(TRV)$ , la cui approssimazione migliora con l'aumentare del numero di replicazioni.

Vale la pena di sottolineare che  $a_1$  risulta preferibile ad  $a$  non solo per la facilità con cui si può ottenere, ma anche perché la bontà dell'accostamento alla distribuzione limite dipende in questo caso più dal numero  $N$  di replicazioni che si vogliono fare che dalla numerosità campionaria con cui si trova ad operare, da cui dipende invece lo sviluppo su cui si basa usualmente la correzione analitica. In compenso occorre ripetere la simulazione per stimare  $E(TRV)$  ogni qual volta che cambiano  $n$  e/o  $p$ .

Per capire meglio, serviamoci di un esempio

Considerando l'esempio del paragrafo 1.2, sappiamo che il test si distribuisce esattamente come un  $\chi_p^2$ . Questo significa che  $E(TRV) = p$  e quindi il parametro  $a$ , in questo particolare caso, è pari a 1.

Sapendo il vero valore del parametro  $a$ , possiamo valutare l'eventuale differenza tra la correzione di  $Be1 \rightarrow a$  e la correzione di  $Be1 \rightarrow a_1$ , facendo variare il numero delle replicazioni della simulazione.

**Tab.1.4.1 Stima della correzione di Be1**

Replicazioni N	Bartlett analitico	Prova1	Prova2	Prova3	Prova4	Prova5	Prova6	Prova7	Prova8	Prova9	Prova10
50.000	1	0.99493	0.99688	0.99416	1.00203	1.00784	0.99914	0.99989	1.00055	1.00728	0.99985
20.000	1	0.98975	0.99583	0.99707	1.00041	1.00436	1.00286	1.00839	0.98867	1.01912	1.00327
13.000	1	1.00243	0.99827	0.99471	0.99910	0.98511	0.99551	0.99883	0.99778	0.99845	1.00401
10.000	1	0.99617	1.00717	1.00199	1.00767	1.00109	1.01071	0.99279	0.99816	1.00079	1.01101
5.000	1	1.00019	1.00650	0.98867	1.01171	1.00174	1.00341	0.98520	0.96848	1.04180	0.99599
2.000	1	1.00036	1.04408	1.00528	0.98863	1.00364	0.97774	0.99981	1.00410	1.01317	0.98767
1.000	1	0.95687	0.98057	1.04893	0.99831	0.97807	0.97937	0.98832	0.96758	0.95861	0.92741

Per il calcolo della stima del parametro di correzione di Be1, abbiamo utilizzato matrici di dati  $X_{30 \times 2}$ .

**Tab.1.4.2 Deviazioni standard e medie delle varie prove**

	Bartlett Analitico	1.000	2.000	5.000	10.000	13.000	20.000	50.000
Medie	1	0.97840	1.00245	1.00037	1.00276	0.99742	1.00097	1.00025
Deviazione standard	1	31.706e-3	17.924e-3	19.140e-3	6.201e-3	5.140e-3	8.969e-3	4.577e-3

Si può facilmente notare che per simulazioni con numero di replicazioni  $N \geq 10.000$ , la stima della correzione di Be1 è molto buona.

Proseguendo sulla stessa linea, si è pensato di eguagliare oltre alla media, anche la varianza della statistica test con la media e la varianza della distribuzione  $\chi_p^2$ .

Questa la chiameremo **correzione di Bartlett empirica2 (Be2)**.

L'idea è quella di determinare, sempre tramite simulazione, due parametri  $a_2$  e  $c$ , in modo che  $\frac{TRV}{a_2} + c$  abbia media e varianza pari a quella della distribuzione  $\chi_p^2$ , ossia

$$E\left(\frac{TRV}{a_2} + c\right) = p \qquad V\left(\frac{TRV}{a_2} + c\right) = 2 * p$$

da queste espressioni si ricava  $a_2$  e  $c$

$$a_2 = \sqrt{\left[ \frac{V(TRV)}{2 * p} \right]} \qquad c = p - \frac{E(TRV)}{a_2}$$

Come nel caso precedente, andiamo a stimare  $V(TRV)$  e  $E(TRV)$  tramite simulazione e quindi  $a_2$  e  $c$ .

Grazie a  $a_2$  e  $c$ , si ha un "controllo" oltre sulla media, anche sulla varianza della statistica test.

L'efficienza di queste due nuove correzioni, verrà verificata nel paragrafo 1.6.

## 1.5 Stima della distribuzione del TRV tramite simulazione

Un'alternativa alle correzioni descritte nei paragrafi precedenti è la simulazione.

La simulazione permette di ottenere la stima della distribuzione del test di interesse: tramite un generatore di numeri pseudo-casuali, si simulano campioni di dati di uguale dimensione del campione originale  $X_{n_{xp}}$ , sui quali si calcola il test TRV e se ne stima la distribuzione.

L'attenzione rivolta dallo statistico, alla scelta del generatore di numeri pseudo-casuali è importante, visto che i risultati della simulazione sono fortemente condizionati dalle sue caratteristiche (vedi capitolo 4).

Il problema fondamentale consiste nel determinare il numero di simulazioni per avere una buona stima della distribuzione del test TRV nel caso non fosse nota la vera distribuzione in ipotesi nulla ( $MH_0$ ).

La verifica d'ipotesi avviene secondo due possibili approcci:

- il primo stimando il percentile  $1-\alpha$
- il secondo stimando il p-value associato al valore del test calcolato sul campione effettivamente osservato, lasciando così piena libertà all'utente nell'interpretazione dei risultati.

Il primo approccio verrà approfondito nel paragrafo 2.3, mentre di seguito si svilupperà nel dettaglio, il secondo approccio.

L'idea alla base della procedura è abbastanza semplice:

- Si fissa un numero  $N$  di simulazioni e per ciascuna di esse si calcola la statistica test TRV.
- Si utilizza come valore soglia  $t_0$ , il valore della statistica test calcolata sul campione effettivamente osservato.

- Sia  $i$  una variabile contatore che, ad ogni simulazione assume valori:

$$i = 1 \text{ se } LR > t_0$$

$$i = 0 \text{ se } LR < t_0$$

Ne segue che:

$$i \sim \text{Bi}(1, p)$$

dove

$$\hat{p} = (\# \text{volte } i=1) / N$$

è la stima del p-value.

- Supponiamo nel seguito di poter ritenere valida l'approssimazione alla normale della distribuzione di  $\hat{p}$ .

Ipotizziamo dapprima che sia pari a  $d$  l'errore assoluto accettato nella stima di  $p$  e che si sia fissata pari a 0.01 la probabilità di commettere un errore maggiore, imponiamo cioè che sia

$$\Pr (|\hat{p} - p| > d) = 0.01$$

Sapendo che

$$Z = \frac{(\hat{p} - p)}{\sqrt{V(\hat{p})}}$$

è prossima ad una normale standardizzata  $Z \sim \text{N}(0,1)$

Ne segue che

$$|Z| < \frac{d}{\sqrt{V(\hat{p})}} = Z_{1-\alpha/2}$$

$$\frac{d}{\sqrt{V(\hat{p})}} = 2.57 \rightarrow Z_{0.995}$$

dove

$$V(\hat{p}) = \frac{p(1-p)}{N}$$

Basterà dunque ricavare  $N$  dall'uguaglianza

$$d = 2.57 \cdot \sqrt{V(\hat{p})}$$

ovvero

$$N = \left( \frac{2.57}{d} \right)^2 p(1-p)$$

- Se ad esempio impostiamo un errore assoluto  $d=5e-3$ , che è anche il valore imposto di default nelle nostre funzioni e si considera la condizione di massima variabilità, cioè  $p=0.5$ , si ottiene un  $N$  pari a

$$N = 66049$$

In questo modo, compiendo 66049 replicazioni, ci si assicura che

$$\Pr(|\hat{p} - p| > d) = 0.01$$

Questo però, è il caso peggiore, cioè di massima variabilità. Nelle funzioni viene utilizzato un valore di default pari a 0.05.

Viene posto questo valore per due motivi:

- si assegna  $\Pr(\text{TRV} > \lambda_{c2} \mid M H_0) = 0.05$ , cioè l'errore di I tipo pari a 0.05
- il tempo di esecuzione dell'algoritmo, è direttamente collegato al numero di cicli che si vuole realizzare nella simulazione.  
Con il valore di  $p = 0.05$ , il numero di replicazioni diviene pari a  $N = 12549$ , approssimato poi a  $N = 13000$ , garantendo quindi una protezione

$$\Pr (|\hat{p} - p| > d) = 0.0089$$

Per ulteriori informazioni si rimanda al paragrafo 4.4.7

Ad ogni modo, l'utente ha la facoltà di modificare il valore di  $p$  e dell'errore, con piena libertà all'utente nell'interpretazione dei risultati.

## 1.6 Analisi della bontà dei metodi

Lo scopo di questo paragrafo, è valutare l'analisi della bontà dei metodi presentati nei paragrafi precedenti. A tale scopo, sono state create delle funzioni (vedi appendice), con l'obiettivo di stimare in corrispondenza di un  $\alpha$  nominale del 5% l' $\alpha$  effettivo, sulla base di  $N=10000$  replicazioni, per lo stesso test con correzione di Ba, Be1, Be2 e simulazione.

Per la stima dei parametri delle correzioni di Bartlett empiriche e per la stima dei percentili della distribuzione di TRV è stato usato  $d=5e-3$  e  $p=0.05$ , che equivale a  $N=12549$  replicazioni e approssimato a  $N=13000$ .

I test presi in considerazione sono quelli riportati di seguito.

### Test d'ipotesi $\Sigma=\Sigma_0$ , con $\mu$ non nota

Sia  $X_{n \times p}$  una matrice normale di dati da  $N_p(\mu, \Sigma)$ , si vuole verificare l'ipotesi:

- $MH_0: \quad \Sigma = \Sigma_0 \quad \rightarrow$  con  $\mu$  non nota
- $MH_1: \quad \Sigma \neq \Sigma_0$

Per la verifica dell'ipotesi in questione si utilizza il test:

$$\text{TRV3} = np (a - \log(g) - 1)$$

dove:

$g$  è la media geometrica degli autovalori di  $\Sigma_0^{-1} S$

$a$  è la media aritmetica degli autovalori di  $\Sigma_0^{-1} S$

sotto  $MH_0$ :  $\text{TRV3} \xrightarrow{d} \chi_{p(p+1)/2}^2$  per  $n \rightarrow \infty$

La correzione di Ba è:

$$a = \frac{6n}{6n - 2p - 1 + \frac{2}{p-1}}$$

Inoltre, per questa verifica d'ipotesi, esiste un'alternativa alla correzione di Ba, proposta da Korin(1968).

Nel suo lavoro l'autore mostra come la distribuzione del test  $\frac{TRV}{b}$  converga alla distribuzione di una F con  $q_1$  e  $q_2$  gradi di libertà, dove:

$$q_1 = v \qquad q_2 = \frac{v+2}{D_2 - D_1^2}$$

e

$$b = \frac{v}{1 - D_1 - \frac{v}{q_2}}$$

con

$$v = 0.5p(p+1)$$

$$D_2 - D_1^2 = \frac{2p^2 + 2p - 5 - \frac{4(p+2)}{(p+1)^2}}{36n^2}$$

$$D_1 = \frac{2p+1 - \frac{2}{p+1}}{6n}$$

Utilizzando la funzione Bontà3 dell'appendice, si è ottenuto la seguente tabella:

### Bontà dei metodi TRV3

**Tab.1.6.1 Stima degli alpha effettivi con alpha nominale del 5%**

Variabili P	Numerosità n	G.d.l.	Alpha	TRV	Ba	Korin	Be1	Be2	Simulazione
2	5	3	0.05	<b>0.183</b>	<b>0.125</b>	<b>0.122</b>	0.047	0.048	0.050
2	10	3	0.05	<b>0.096</b>	<b>0.079</b>	<b>0.079</b>	0.052	0.053	0.054
2	20	3	0.05	<b>0.075</b>	<b>0.067</b>	<b>0.067</b>	0.052	0.053	0.052
2	30	3	0.05	<b>0.063</b>	<b>0.058</b>	<b>0.058</b>	0.047	0.048	0.048
2	40	3	0.05	<b>0.059</b>	<b>0.056</b>	<b>0.056</b>	0.050	0.050	0.051
2	50	3	0.05	<b>0.059</b>	<b>0.056</b>	<b>0.056</b>	0.050	0.049	0.049
2	60	3	0.05	<b>0.059</b>	<b>0.056</b>	<b>0.056</b>	0.049	0.049	0.048
2	65	3	0.05	<b>0.058</b>	<b>0.055</b>	<b>0.055</b>	<b>0.055</b>	0.054	0.053
2	70	3	0.05	<b>0.055</b>	0.053	0.053	0.049	0.048	0.048
2	80	3	0.05	<b>0.056</b>	0.054	0.054	0.048	0.047	0.047
2	100	3	0.05	<b>0.058</b>	<b>0.056</b>	<b>0.056</b>	<b>0.055</b>	<b>0.056</b>	<b>0.058</b>
2	130	3	0.05	0.051	0.049	0.049	0.049	0.049	0.050
2	150	3	0.05	0.050	0.050	0.050	0.051	0.053	0.052
3	50	6	0.05	<b>0.063</b>	<b>0.057</b>	<b>0.057</b>	0.051	0.051	0.049
5	70	15	0.05	<b>0.062</b>	0.053	0.053	0.048	0.048	0.047
5	90	15	0.05	<b>0.062</b>	0.053	0.053	0.050	0.051	0.051
8	100	36	0.05	<b>0.073</b>	<b>0.056</b>	<b>0.055</b>	0.050	0.051	0.051
10	50	55	0.05	<b>0.141</b>	<b>0.070</b>	<b>0.068</b>	0.049	0.049	0.050
10	100	55	0.05	<b>0.086</b>	<b>0.056</b>	<b>0.056</b>	0.048	0.046	0.048
15	50	120	0.05	<b>0.434</b>	<b>0.099</b>	<b>0.087</b>	<b>0.055</b>	<b>0.055</b>	0.054
20	150	210	0.05	<b>0.149</b>	<b>0.064</b>	<b>0.063</b>	0.053	0.053	0.053

Come si può vedere, sono stati evidenziati in neretto i casi in cui approssimando alla seconda cifra decimale, il valore dell'alpha stimato risulta diverso da 0.05.

Dalla tabella si nota, come previsto, che il metodo peggiore risulta essere il test asintotico (TRV). Inoltre si può vedere che la correzione di Ba, in alcuni casi è "peggiore" del metodo di Korin. Infatti la "convergenza" della statistica test verso la distribuzione asintotica  $\chi^2_{p(p+1)/2}$ , viene raggiunta, a parità di gradi di libertà, in corrispondenza ad una numerosità campionaria più alta.

La deviazione standard e la media dei metodi considerati sono:

**Tab.1.6.2 Deviazioni standard e medie dei metodi considerati**

	TRV	Ba	Korin	Be1	Be2	Simulazione
media	0.09486	0.06295	0.06204	0.05038	0.05052	0.05062
Deviazione standard	69.47e-3	22.52e-3	20.69e-3	2.53e-3	2.83e-3	2.83e-3

Dai risultati ottenuti, si possono ricavare alcune informazioni generali molto interessanti.

Innanzitutto, vengono confermate alcune considerazioni espresse nella parte teorica: il metodo peggiore è il test TRV; la deviazione standard e la media del metodo di Korin, è inferiore, anche se non di molto, a quella di Ba.

Si osservi che, le differenze tra la deviazioni standard e le medie della correzione di Be1, Be2 e della simulazione risultano minime, e i tre metodi considerati risulterebbero, pertanto, equivalenti.

Test d'ipotesi  $\Sigma_{12}=0$ , con  $\mu$  non nota

Sia  $X_{n \times p}$  una matrice normale di dati da  $N_p(\mu, \Sigma)$ , si vuole verificare l'ipotesi:

- $MH_0: \Sigma_{12} = 0 \quad \rightarrow$  con  $\mu$  non nota
- $MH_1: \Sigma_{12} \neq 0$

Partizionando la matrice dei dati in due sottomatrici di dimensioni  $(n \times p_1)$  e  $(n \times p_2)$  rispettivamente, la verifica dell'ipotesi in questione si realizza ricorrendo al test rapporto di verosimiglianza dato da:

$$TRV72 = -n \log |I - S^{-1}_{22} S_{21} S^{-1}_{11} S_{12}|$$

sotto  $MH_0$ :  $TRV72 \xrightarrow{d} \chi^2_{p_1 p_2}$  per  $n \rightarrow \infty$

In questo caso, la correzione di Ba, è pari a:

$$a = 1 - \frac{0.5(p_1 + p_2 + 3)}{n}$$

Come nel caso precedente, prendendo in considerazione la funzione Bontà72 dell'appendice, otteniamo la seguente tabella:

## Bontà dei metodi TRV72

**Tab.1.6.3 Stima degli alpha effettivi con alpha nominale del 5%**

Variabili p1	Variabili p2	Numerosità N	G.d.l.	Alpha	TRV	Ba	Be1	Be2	Simulazione
2	2	5	4	0.05	<b>0.625</b>	<b>0.083</b>	0.053	0.048	0.049
2	2	10	4	0.05	<b>0.190</b>	0.049	0.049	0.049	0.050
2	3	10	6	0.05	<b>0.281</b>	<b>0.056</b>	0.052	0.051	0.050
3	2	15	6	0.05	<b>0.156</b>	0.046	0.047	0.046	0.046
4	2	10	8	0.05	<b>0.414</b>	<b>0.064</b>	0.054	0.050	0.050
4	2	15	8	0.05	<b>0.221</b>	<b>0.057</b>	0.054	0.053	0.053
3	3	20	9	0.05	<b>0.161</b>	0.053	0.052	0.053	0.052
4	3	15	12	0.05	<b>0.304</b>	<b>0.054</b>	0.050	0.049	0.049
4	4	10	16	0.05	<b>0.825</b>	<b>0.126</b>	<b>0.059</b>	0.049	0.049
4	4	20	16	0.05	<b>0.271</b>	0.052	0.048	0.047	0.047
4	4	30	16	0.05	<b>0.168</b>	0.051	0.049	0.048	0.046
4	4	40	16	0.05	<b>0.124</b>	0.051	0.051	0.051	0.052
4	4	50	16	0.05	<b>0.097</b>	0.049	0.049	0.048	0.048
4	4	60	16	0.05	<b>0.097</b>	0.053	0.054	0.053	0.051
4	4	70	16	0.05	<b>0.085</b>	0.049	0.049	0.048	0.046
4	4	80	16	0.05	<b>0.083</b>	0.052	0.051	0.052	0.051
4	4	90	16	0.05	<b>0.075</b>	0.050	0.049	0.049	0.051
4	4	100	16	0.05	<b>0.077</b>	0.054	0.054	0.054	0.054
4	4	150	16	0.05	<b>0.068</b>	0.054	0.053	0.053	0.054
4	4	200	16	0.05	<b>0.063</b>	0.052	0.051	0.050	0.050
5	4	40	20	0.05	<b>0.139</b>	<b>0.048</b>	<b>0.044</b>	<b>0.045</b>	<b>0.045</b>
5	5	50	25	0.05	<b>0.145</b>	0.054	0.052	0.052	0.050
6	5	50	30	0.05	<b>0.160</b>	0.051	0.050	0.049	0.050
6	6	50	36	0.05	<b>0.189</b>	0.050	0.049	0.049	0.048
7	6	50	42	0.05	<b>0.224</b>	0.052	0.049	0.049	0.049
8	7	50	56	0.05	<b>0.314</b>	<b>0.060</b>	0.054	0.054	0.053
10	8	50	80	0.05	<b>0.476</b>	<b>0.058</b>	0.053	0.051	0.054
10	10	50	100	0.05	<b>0.638</b>	<b>0.065</b>	<b>0.057</b>	<b>0.057</b>	<b>0.058</b>
12	10	50	120	0.05	<b>0.774</b>	<b>0.067</b>	0.052	0.049	0.050
14	10	50	140	0.05	<b>0.875</b>	<b>0.076</b>	<b>0.055</b>	<b>0.055</b>	0.054

Anche in questo caso sono stati evidenziati in neretto i casi in cui approssimando alla seconda cifra decimale, il valore dell'alpha stimato risulta diverso da 0.05.

Da una prima analisi della tabella si nota, come previsto, tra i metodi considerati quello peggiore risulta essere il test asintotico (TRV).

Passiamo ora ad una analisi più approfondita.

Le deviazioni standard e le medie dei metodi considerati sono:

**Tab.1.6.4 Deviazioni standard e medie dei metodi considerati**

	TRV	Ba	Be1	Be2	Simulazione
media	0.27730	0.05787	0.05143	0.05037	0.05030
Deviazione standard	23.93e-3	15.28e-3	3.08e-3	2.77e-3	2.92e-3

Si può facilmente notare come viene confermata la considerazione precedente. Infatti i valori della deviazione standard e della media del test TRV sono i più grande di tutti.

Per quanto riguarda gli altri metodi, è importante sottolineare che i metodi migliori risultano la simulazione e il Be2.

### Test d'ipotesi $\Sigma = I$ , con $\mu$ non nota

Sia  $X_{n \times p}$  una matrice normale di dati da  $N_p(\underline{\mu}, \Sigma)$ , si vuole verificare l'ipotesi:

- $MH_0: \quad \Sigma = I \rightarrow$  con  $\underline{\mu}$  non nota
- $MH_1: \quad \Sigma \neq I$

La verifica dell'ipotesi in questione si realizza ricorrendo al test:

$$TRV7 = -n \log |R|$$

dove:

R è la matrice di correlazione campionaria

sotto  $MH_0$ :  $TRV7 \xrightarrow{d} \chi^2_{p(p-1)/2}$  per  $n \rightarrow \infty$

La statistica corretta è:

$$TRV7' = \frac{TRV7}{a}$$

dove

$$a = \frac{1}{1 - \frac{2p+11}{6n}}$$

Si presentano in una tabella, alcuni esempi di utilizzo della funzione Bontà7.

## Bontà dei metodi TRV7

**Tab.1.6.5 Stima degli alpha effettivi con alpha nominale del 5%**

Variabili P	Numerosità n	G.d.l.	Alpha	TRV	Ba	Be1	Be2	Simulazione
2	4	1	0.05	<b>0.208</b>	<b>0.039</b>	0.048	0.051	0.051
2	5	1	0.05	<b>0.154</b>	<b>0.040</b>	<b>0.043</b>	<b>0.044</b>	<b>0.042</b>
2	6	1	0.05	<b>0.136</b>	0.046	0.047	0.049	0.048
2	10	1	0.05	<b>0.089</b>	0.047	0.050	0.050	0.051
3	4	3	0.05	<b>0.525</b>	<b>0.057</b>	0.053	0.050	0.051
3	5	3	0.05	<b>0.335</b>	0.051	0.050	0.051	0.053
3	10	3	0.05	<b>0.132</b>	0.049	0.051	0.051	0.048
3	15	3	0.05	<b>0.094</b>	<b>0.044</b>	<b>0.045</b>	<b>0.045</b>	0.050
3	20	3	0.05	<b>0.080</b>	0.051	0.052	0.051	0.051
4	5	6	0.05	<b>0.659</b>	<b>0.101</b>	<b>0.060</b>	0.050	0.050
4	10	6	0.05	<b>0.204</b>	0.053	0.052	0.051	0.050
4	10	6	0.05	<b>0.207</b>	0.053	0.051	0.051	0.052
4	15	6	0.05	<b>0.130</b>	0.052	0.051	0.052	0.052
4	20	6	0.05	<b>0.109</b>	<b>0.057</b>	<b>0.055</b>	<b>0.056</b>	<b>0.056</b>
4	30	6	0.05	<b>0.080</b>	0.049	0.049	0.050	0.050
4	40	6	0.05	<b>0.073</b>	0.050	0.049	0.048	0.047
4	100	6	0.05	<b>0.059</b>	0.050	0.050	0.051	0.052
4	180	6	0.05	0.053	0.050	0.049	0.048	0.047
5	40	10	0.05	<b>0.084</b>	0.052	0.052	0.054	0.054
5	50	10	0.05	<b>0.070</b>	0.048	0.047	0.047	0.049
6	50	15	0.05	<b>0.082</b>	0.050	0.049	0.049	0.049
6	50	15	0.05	<b>0.078</b>	0.048	0.049	0.050	0.049
7	50	21	0.05	<b>0.090</b>	0.048	0.047	0.048	0.047
8	50	28	0.05	<b>0.106</b>	0.048	0.046	0.046	0.046
10	20	45	0.05	<b>0.483</b>	<b>0.066</b>	0.048	<b>0.045</b>	<b>0.045</b>
10	30	45	0.05	<b>0.261</b>	<b>0.059</b>	0.054	0.052	0.052
10	50	45	0.05	<b>0.144</b>	0.053	0.052	0.051	0.051
10	50	45	0.05	<b>0.140</b>	0.050	0.050	0.051	0.052
12	50	66	0.05	<b>0.192</b>	0.048	<b>0.045</b>	<b>0.045</b>	<b>0.045</b>
14	50	91	0.05	<b>0.270</b>	0.054	0.050	0.050	0.050

Come dalle aspettative, i metodi migliori risultano quelli con correzione di Be1, Be2 e simulazione .

Passiamo ora alla deviazione standard e alla media dei metodi considerati:

**Tab.1.6.6 Deviazioni standard e medie dei metodi considerati**

	TRV	Ba	Be1	Be2	Simulazione
media	0.17757	0.05210	0.04980	0.04957	0.04967
Deviazione standard	19.63e-3	10.81e-3	3.82e-3	2.75e-3	2.96e-3

Prendendo visione di questi valori si possono confermare le considerazioni fatte precedentemente.

Nel nostro caso, comunque, non possiamo affermare che uno di questi tre metodi sia migliore dell'altro. Si noti, infatti, che per la deviazione standard risulta migliore il Be2. Invece considerando l'alpha stimato, risulta essere migliore il Be1.

### Test d'ipotesi $\Sigma_1 = \dots = \Sigma_g$ (test di Box)

Consideriamo  $g$  matrici normali di dati da  $N_p(\mu, \Sigma)$  con dimensione  $X_{n \times p}$ , si vuole verificare l'ipotesi:

- $MH_0: \Sigma_1 = \dots = \Sigma_g$
- $MH_1: \Sigma_1 \neq \dots \neq \Sigma_g \rightarrow$  almeno una sia diversa

La verifica dell'ipotesi in questione si realizza ricorrendo al test:

$$TRV9 = \sum_{i=1}^g n_i \log |S_i^{-1}S| = -n \log |S| - \sum_{i=1}^g n_i \log |S_i|$$

sotto  $MH_0$ :  $TRV9 \xrightarrow{d} \chi_{p(p+1)(g-1)/2}^2$  per  $n \rightarrow \infty$

La correzione di Ba è:

$$a = 1 - \frac{(2p^2 + 3p - 1)}{6(p+1)(g-1)} \cdot \sum_{i=1}^g \left[ \left( \frac{1}{n_i - 1} \right) \left( \frac{1}{n - g} \right) \right]$$

La statistica test corretta si ottiene facendo:

$$TRV9' = a \cdot \sum_{i=1}^g [(n_i - 1) \log |S_{ui}^{-1} - S_u|]$$

dove

$$S_{ui} = \frac{n_i}{n_i - 1} S_i \quad S_u = \frac{n}{n - g} S$$

Utilizzando la funzione Bontà9, si ottiene la seguente tabella:

## Bontà dei metodi TRV9

**Tab.1.6.7 Stima degli alpha effettivi con alpha nominale del 5%**

Campioni G	Variabili p	Numerosità Campione	G.d.l.	Alpha	TRV	Ba	Be1	Be2	Simulazione
2	2	5	3	0.05	<b>0.204</b>	0.050	0.050	0.049	0.049
3	3	5	12	0.05	<b>0.586</b>	<b>0.068</b>	0.052	0.050	0.050
3	3	25	12	0.05	<b>0.093</b>	0.053	0.052	0.053	0.054
7	2	5	18	0.05	<b>0.450</b>	<b>0.055</b>	0.052	0.053	0.054
10	2	5	27	0.05	<b>0.541</b>	<b>0.055</b>	0.051	0.051	0.051
10	2	10	27	0.05	<b>0.201</b>	0.051	0.047	0.048	0.048
2	7	10	28	0.05	<b>0.854</b>	<b>0.108</b>	<b>0.057</b>	0.054	0.054
4	4	5	30	0.05	<b>0.978</b>	<b>0.227</b>	<b>0.061</b>	0.049	0.049
4	4	25	30	0.05	<b>0.128</b>	0.050	0.050	0.050	0.050
4	4	220	30	0.05	0.054	0.051	0.052	0.049	0.048
15	2	5	42	0.05	<b>0.688</b>	<b>0.058</b>	0.050	0.051	0.051
10	3	5	54	0.05	<b>0.948</b>	<b>0.105</b>	<b>0.057</b>	0.052	0.051
10	3	30	54	0.05	<b>0.115</b>	<b>0.051</b>	0.053	0.052	0.051
5	5	6	60	0.05	<b>0.999</b>	<b>0.399</b>	<b>0.077</b>	0.053	0.051
6	5	6	75	0.05	<b>0.999</b>	<b>0.450</b>	<b>0.080</b>	0.051	0.048
6	5	25	75	0.05	<b>0.224</b>	<b>0.056</b>	0.054	<b>0.055</b>	0.054
4	7	25	84	0.05	<b>0.341</b>	0.052	0.048	0.046	0.046
10	4	5	90	0.05	<b>0.999</b>	<b>0.430</b>	<b>0.073</b>	0.052	0.050
10	4	20	90	0.05	<b>0.261</b>	<b>0.058</b>	<b>0.055</b>	<b>0.056</b>	0.054
5	7	8	112	0.05	<b>1.000</b>	<b>0.638</b>	<b>0.089</b>	0.051	0.047
5	7	15	112	0.05	<b>0.803</b>	<b>0.074</b>	0.053	0.052	0.051
6	7	20	140	0.05	<b>0.610</b>	<b>0.063</b>	0.052	0.051	0.049
7	7	20	168	0.05	<b>0.654</b>	<b>0.062</b>	0.053	0.051	0.049

Da una prima visione dei valori ottenuti, si possono trarre le stesse conclusioni dei casi precedenti.

Inoltre, val la pena di notare inoltre che la correzione di Ba fa sì che le cose migliorino decisamente non solo se vengono rispettati i vincoli usualmente posti sui parametri (g e p non superiori a 5 e n almeno 20) ma qualche volta anche quando il numero g di campioni o quello p delle variabili supera la soglia. In questi casi sembra importante che la numerosità campionaria non sia piccola in rapporto ai gradi di libertà.

Passiamo ad una ulteriore analisi, le deviazioni standard e le medie:

**Tab.1.6.8 Deviazioni standard e medie dei metodi considerati**

	TRV	Ba	Be1	Be2	Simulazione
media	0.55347	0.14191	0.05730	0.05126	0.05039
Deviazione standard	62.07e-3	19.20e-3	13.48e-3	2.60e-3	2.40e-3

Dall'ultima analisi condotta, possiamo affermare che i metodi migliori in questo caso risultano la simulazione e il Be2.

## Conclusioni

Le analisi effettuate portano ad affermare che i metodi migliori risultano essere la correzione di Be2 e la simulazione.

Questo è dovuto al fatto che la bontà dell'accostamento alla distribuzione limite dipende dal numero N di replicazioni che dalla numerosità campionaria con cui ci si trova ad operare.

Si pone allora il problema di sceglierne uno come "migliore".

Nel nostro caso, prediligiamo la correzione di Be2 quando dobbiamo replicare l'analisi su campioni di dimensione uguale. Questo perché si può stimare una sola volta la correzione e utilizzarla poi per tutti i campioni.

Invece prediligiamo la simulazione quando cambiano in tutto o in parte la numerosità o le variabili del campione d'interesse. Infatti la simulazione deve essere ripetuta ogni volta, perché stima il p-value associato al valore del test calcolato sul campione effettivamente osservato.

Nel capitolo 5 (Un'applicazione pratica), verranno considerati degli esempi pratici per evidenziare ciò che è stato detto pocanzi.

## CAPITOLO 2

### TEST UNIONE-INTERSEZIONE (UIT)

#### 2.1 Definizione del UIT

L'impiego di test basati sul rapporto di verosimiglianza per la verifica di un'ipotesi multivariata nulla ( ${}_M H_0$ ), lascia aperto un problema, in quanto, il rifiuto dell'ipotesi nulla non consente di verificare, con un adeguato controllo dell'errore di prima specie, se il rifiuto sia da imputare a uno o ad alcuni parametri di specifico interesse.

A questo problema si può, in linea teorica, dare soluzione ricorrendo ai test unione intersezione (UIT) introdotti da Roy (1953).

#### Procedura di costruzione del UIT

Data un'ipotesi multivariata del tipo di quelle considerate nel CAPITOLO 1, si scompone l'ipotesi  ${}_M H_0$  in un insieme d'ipotesi univariate  ${}_{\mu} H_{0a}$ , dove  $\underline{a}$  è un opportuno vettore di coefficienti in modo da:

- **Accettare**  ${}_M H_0$  se ogni  ${}_{\mu} H_{0a}$  **viene accettata**

$${}_M H_0 : \bigcap_a {}_{\mu} H_{0a}$$

- **Rifiutare**  ${}_M H_0$  se qualche  ${}_{\mu} H_{0a}$  **viene rifiutata**

$${}_M H_1 : \bigcup_a {}_{\mu} H_{1a}$$

Regione di rifiuto:  $\mathbf{R} = \bigcup_a \mathbf{R}_a$

Regione di accettazione:  $\mathbf{A} = \bigcap_a \mathbf{A}_a$

L'obiettivo è quello di costruire una funzione test "ragionevole" per ognuna delle ipotesi componenti, sia  $T_a$  statistica test per  $_{\mu}H_{0a}$ , ed eliminare poi la dipendenza dallo specifico  $a$ .

### Test unilaterali

Nell'ipotesi che al variare di  $a$  tutte le ipotesi alternative siano unilaterali, con regione di accettazione del tipo  $\{-\infty, T_a\}$ , oppure  $\{T_a, \infty\}$ , si ha come statistica test per  $_{M}H_0$ :

$$T = \max_a T_a \quad \text{oppure} \quad T = \min_a T_a$$

Si accetta  $_{M}H_0$  se:

$$T = \max_a \{T_a\} \leq T_{1-\alpha} \quad \text{oppure} \quad T = \min_a \{T_a\} \geq T_{1-\alpha}$$

con  $1-\alpha$  livello di significatività associato a T

Come è noto, se  $_{M}H_0$  viene accettata con probabilità  $(1 - \alpha)$ , alla singola ipotesi componente resta associato un livello di significatività  $\geq (1 - \alpha)$ .

Infatti se

$$P\{ T \leq T_{1-\alpha} \mid _{M}H_0 \} = 1 - \alpha$$

$\forall a$  si ha

$$T_a \leq T$$

dunque

$$P\{T_a \leq T_{1-\alpha} \mid M H_0\} \geq 1 - \alpha$$

Analogamente se

$$P\{T \geq T_{1-\alpha} \mid M H_0\} = 1 - \alpha$$

$\forall a, T_a \geq T$

e quindi

$$P\{T_a \geq T_{1-\alpha} \mid M H_0\} \geq 1 - \alpha$$

### Test bilaterali

Per i test bilaterali si costruiscono due regioni critiche in corrispondenza rispettivamente a:

$$T_1 = \max_a \{T_a\} \quad \text{e} \quad T_2 = \min_a \{T_a\}$$

Si accetta  $M H_0$  se:

$$T_1 \leq T_{1-\alpha/2} \quad \text{e} \quad T_2 \geq T_{\alpha/2}$$

### Il test UIT ha un vantaggio rispetto al TRV

Si può stabilire se una qualche  $\mu H_{0a}$  d'interesse abbia portato al rifiuto di  $M H_0$ .

## 2.2 Esempio di UIT

Si riprende lo stesso esempio del TRV, in modo tale da notare le differenze dei due metodi. Sia  $X_{n \times p}$  una matrice normale di dati da  $N_p(\mu, \Sigma)$  con matrice di covarianze nota e vettore media ignoto nei confronti della quale si formula l'ipotesi:

$${}_M H_0: \mu = \mu_0$$

$${}_M H_1: \mu \neq \mu_0$$

posto:  $y = Xa$

$$y = a'x \sim (a'\mu, a'\Sigma a)$$

con

$$\bar{y} = \sum_{i=1}^n y_i / n = a' \bar{x}$$

Dunque sotto  ${}_M H_{0a}$ :

$$y \sim N_p(a'\mu_0, a'\Sigma/n a)$$

Quindi al variare della configurazione di  $a$ :

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}, \dots, a_j = \begin{pmatrix} 0 \\ \dots \\ 1 \\ 0 \\ \dots \end{pmatrix}, \dots, a_p = \begin{pmatrix} 0 \\ \dots \\ 0 \\ 1 \end{pmatrix} \dots$$

si ottengono infinite **ipotesi univariate** del tipo:

$${}_M H_{0a} : a' \mu = a' \mu_0$$

**La statistica test per  $\mu H_{0a}$  è:**

$$z_a = | \bar{x}_a - \mu_0 | / ( \Sigma / n )^{1/2}$$

**Regione di rifiuto:**

$$R_a = \{ z_a : z_a^2 > C_\alpha \} \quad \forall a, \text{ ossia}$$

$H_0$  viene accettata se e solo se:  $z_a^2 \leq C_\alpha, \quad \forall a$ , ovvero se

$$T = \max_a \{ z_a^2 \} \leq C_\alpha$$

dove  $C_\alpha$ , 100(1- $\alpha$ ) percentile della distribuzione di T.

In questo caso:

$$T = n(\bar{x} - \mu_0)' \Sigma^{-1} (\bar{x} - \mu_0)$$

Ora essendo  $(\bar{x} - \mu_0)' \Sigma^{-1} (\bar{x} - \mu_0) \sim \chi_p^2$ , risulta che sotto  $H_0$ :

$$T \sim \chi_p^2 \Rightarrow \Pr\{ z_{a^*}^2 > C_\alpha | H_0 \} = \alpha$$

con  $C_\alpha$ , pari al 100(1- $\alpha$ ) percentile da  $\chi_p^2$

**N.B.**

Per le singole componenti si accetta  $\mu H_{0a}$  se:  $z_a^2 \leq C_\alpha$

Inoltre, questo è uno dei pochi casi in cui il test UIT **coincide** con il test TRV.

### In particolare:

- Se si **accetta**  $MH_0$ , questo implica l'accettazione simultanea di tutte le  $p$  ipotesi componenti.
- Se si **rifiuta**  $MH_0$ , a livello  $\alpha$ , allora si rifiuta almeno un'ipotesi  $\mu H_{0a}$  e a differenza del TRV, si può verificare se una specifica "ipotesi componente di interesse" ha portato al rifiuto.

## 2.3 Stima della distribuzione del UIT tramite simulazione

Purtroppo, fatti salvi alcuni semplici casi (come quello appena illustrato), non si conoscono le distribuzioni delle statistiche UIT, allora si è pensato, come per il TRV, di ricorrere all'uso della simulazione.

L'idea di base non è più quella di stimare il p-value della statistica test, ma il percentile di interesse di  $T$  da usare per la verifica dell'ipotesi globale e delle ipotesi componenti.

Come nel caso TRV, l'idea è semplice:

- Per prima cosa, si simula  $N$  volte il test UIT
- $N$ , è definito allo stesso modo del caso TRV, cioè:

$$N = \left( \frac{2.57}{d} \right)^2 p(1-p)$$

- Anche in questo caso si è impostato di default un errore assoluto  $d=5e-3$  e  $p=5e-2$ , che comportano un numero di simulazioni pari a 12549, approssimato a  $N=13000$ . Il valore  $p$ , è di fondamentale importanza, perché oltre a determinare il numero di simulazioni, determina anche il valore del percentile di interesse.
- Dopo aver inserito tutti i valori dei test in un vettore, li si ordina in ordine crescente e si prende come valore d'interesse il percentile  $1-p$ .
- Infine, si utilizza il percentile  $1-p$ , come valore soglia. Infatti, se il test calcolato sul campione effettivamente osservato è più piccolo del percentile di interesse, allora si accetta l'ipotesi  $MH_0$ , altrimenti la si rifiuta.

Anche in questo caso, l'utente ha la facoltà di modificare il valore di  $p$  e dell'errore  $d$ .

## CAPITOLO 3

### LA FUNZIONE SIMULAZIONE

#### 3.1 Introduzione al compilatore R

Esiste una gamma assai vasta di software specializzati nell'analisi statistica dei dati, basta ricordare SAS, SPSS, S PLUS, solo per citarne alcuni. Sono prodotti che costituiscono senz'altro un fondamentale ed insostituibile ausilio per il lavoro dello statistico, tuttavia molti di questi programmi sono anche alquanto costosi e ne è consentito l'uso solo su licenza del produttore.

Da alcuni anni a questa parte, soprattutto in ambito universitario (ma non solo), si sta sempre più diffondendo un nuovo package, il software R.

R è un ambiente statistico scaricabile gratuitamente da internet sul sito *The R Project for Statistical Computing* il cui indirizzo internet è <http://www.r-project.org><sup>1</sup>.

Esso è il frutto del lavoro di un gruppo, sempre più folto, di ricercatori in campo statistico ed informatico a livello mondiale.

R è un ambiente integrato che permette di elaborare dati, eseguire calcoli ed effettuare rappresentazioni grafiche. Si parla di ambiente e non di software perché è un linguaggio object-oriented, che può essere facilmente esteso dall'utente finale. Nasce come estensione del linguaggio di programmazione S sviluppato nei Laboratori Bell americani dell'AT&T.

A differenza di altri ambienti statistici, R è di pubblico dominio (open-source), nel rispetto dei diritti d'autore, può essere copiato, distribuito e modificato da chiunque.

R si presenta come uno strumento completo e ben si adatta a molteplici esigenze: è il software maggiormente utilizzato a livello mondiale dai ricercatori in campo statistico.

La scelta di R, per il nostro lavoro, è dovuta a due diverse considerazioni: da una parte l'immediata disponibilità a costo nullo e in secondo luogo l'utilizzo di un programma che ha già implementato diverse applicazioni statistiche, pur permettendo ampio margine di operabilità.

Esaminiamo ora i suoi punti di forza ed anche alcune sue debolezze e/o mancanze.

### Punti di forza

- 1) disponibilità di strumenti di analisi statistica (già pronti) dai più elementari ai più sofisticati
- 2) costo zero per l'utente finale
- 3) dotato di funzioni di help in linea per ciascun comando facilmente richiamabile dal programma
- 4) è un software in rapidissimo sviluppo
- 5) possibilità di programmare, creando strumenti personali di analisi statistica ad hoc necessari per le proprie ricerche
- 6) dispone di una vasta manualistica (in lingua inglese) consultabile da Internet
- 7) è possibile accedere tramite Internet, ad una vasta gamma di librerie per analisi statistiche molto dettagliate, create e messe a disposizione di tutti da parte di ricercatori di tutto il mondo
- 8) è abbastanza semplice nell'impiego a livello didattico nell'insegnamento della statistica

## Punti deboli

- 1) è un ambiente basato su un'interfaccia utente a carattere (CUI), le istruzioni vengono immesse da una linea di comando (command line interface), anche se è possibile implementare un'interfaccia (GUI), Graphical User Interface
- 2) possono mancare alcune funzioni statistiche
- 3) la sintassi del linguaggio a volte risulta alquanto inconsistente

Tuttavia, alla mancanza di alcune funzioni statistiche, si può sopperire scrivendo il codice e creando da sé questi strumenti, anche se ciò richiede maggiore padronanza di strumenti informatici avanzati.

Come può evincersi, gli aspetti positivi del software R prevalgono nettamente su quelli negativi e ne hanno decretato il successo a livello mondiale all'interno della comunità di ricercatori in ambito statistico.

## 3.2 Algoritmo, che stima la distribuzione del TRV

Dopo aver presentato, nel capitolo 1, la teoria per ottenere la stima della distribuzione dei test TRV, si passa ora alla sua realizzazione su PC.

Le funzioni, che via simulazione ci portano a costruire i test TRV, si fondano tutte sulla stessa idea di base, cioè quella di stimare il p-value, associato al test calcolato sui dati del campione effettivamente osservato, al fine di lasciare maggior libertà all'utente nel prendere una decisione.

L'algoritmo che ci permette questo è il seguente:

```
#calcola la statistica test sui dati "originali"
t<-trv(dati,...)

#determina il numero N di iterazioni della simulazione
N<-signif(((2.57/errore)^2*alpha*(1-alpha)),2)

#simula n*N dati con distribuzione normale multivariata  $N_p(\text{media},\text{sigma})$ 
dati0<-mvrnorm(n=n*N,media,sigma,tol=1e-6,empirical=FALSE)

# divide i dati simulati precedentemente in N campioni di dimensione  $X_{n \times p}$  e calcola la
statistica test sui campioni "simulati"
h<-1
k<-n
for(i in 1:N)
{
  dat<-NULL
  dat<-dati0[h:k,]
  h<-h+n
  k<-k+n
  x<-trv(dat,...)
  y<-c(y,x)
}
```

```
#calcola il p-value
pvalue<-sum(y>test)/N
```

Questo breve e semplice algoritmo, è la base che sta sotto ogni funzione che simula la distribuzione dei test T.R.V. Come si può vedere si è scelto di generare in un solo passo tutti i dati simulati evitando nel ciclo for di utilizzare il generatore di dati. Grazie a questa scelta si riesce guadagnare tempo nella simulazione rendendola più efficiente.

Purtroppo, nel dividere i dati simulati in N campioni la cosa più elegante ed efficace sarebbe stata quella di utilizzare:

```
# divide i dati simulati precedentemente in N campioni di dimensione  $X_{nxp}$ 
dat<-array(dati0,dim=c(n,p,N))
```

```
#calcola la statistica test sui campioni "simulati"
x<-apply(dat,3,trv)
```

Questo però non è possibile perché nella divisione dei campioni si perde la corrispondenza con la matrice di varianza e covarianza. Il problema è dovuto all'istruzione array che considera i dati simulati  $X_{(nxN)xp}$  come un unico vettore.

### Esempio

```
> p<-2
> n<-2
> N<-2
> dati0<-mvrnorm(n=n*N,media,sigma,tol=1e-6,empirical=FALSE)
> dati0
      [,1]      [,2]
[1,] 0.7028825 -0.3607712
[2,] 0.7848027  2.9087669
[3,] -6.6530266  1.0622148
[4,] 1.0301287  4.2870065
```

```

> dat<-array(dati0,dim=c(n,p,N))
> dat
, , 1
      [,1]      [,2]
[1,] 0.7028825 -6.653027
[2,] 0.7848027  1.030129

, , 2
      [,1]      [,2]
[1,] -0.3607712 1.062215
[2,]  2.9087669 4.287007

```

Come si può vedere non va bene.

### 3.3 Algoritmo, che stima la distribuzione del UIT

In questo caso a differenza dell'algoritmo precedente, basato sulla stima del p-value, ci si basa sul calcolo del percentile d'interesse della distribuzione del UIT. L'algoritmo è il seguente (come nel caso precedente):

```

#determina il numero N di iterazioni di della simulazione
...
#simula n*N dati con distribuzione normale multivariata  $N_p(\text{media},\text{sigma})$ 
...
# divide i dati simulati precedentemente in N campioni di dimensione  $X_{n \times p}$  e calcola la
statistica test sui campioni "simulati"
...
#ordina i valori in modo crescente
y1<-sort(y)

#determina il percentile d'interesse della statistica U.I.T.
qmax<-quantile(y1,1-alpha)

```

## CAPITOLO 4

# ANALISI STATISTICA DELLE SIMULAZIONI

### 4.1 Introduzione

Trattando di numeri casuali è d'obbligo soffermarsi su cosa si intende per casuale. Definiamo "casuale" qualcosa di accidentale e di fortuito.

Esistono diversi strumenti a disposizione dell'utente per costruire sequenze di numeri casuali: algoritmi, tavole, dispositivi fisici.

Tra questi, i primi hanno permesso una pratica diffusione di tale sequenze, consentendone la veloce riproduzione e la facile consultazione. Quanto si è detto, però fa pensare che è impossibile ritenere casuale un numero prodotto aritmicamente o ricavato consultando delle tavole, lì dove per casuale si intende un qualcosa di accidentale e di fortuito.

Casuale potrebbe essere l'estrazione a sorte di ognuna delle  $n$  cifre che compongono il numero da generare. Purtroppo questo modo di procedere comporta un enorme dispendio di tempo; se si dovesse generare una sequenza in questo modo, ci vorrebbero  $n$  estrazioni per ciascun numero.

Inoltre, la generazione di "vere" sequenze di numeri casuali non è possibile su di un computer senza uno speciale hardware. Il motivo di ciò è facilmente intuibile: quando si opera su un elaboratore elettronico i numeri che si utilizzano sono evidentemente razionali ed appartengono ad un insieme finito. Questo comporta che durante le fasi di generazione, il computer ritornerà in uno stato già occupato e comincerà a riprodurre gli stessi numeri. Questa caratteristica, detta *periodo*, identifica una delle qualità di un generatore.

Tutto ciò spiega perché è impensabile definire veramente casuali le sequenze generate tramite calcolatore e perciò esse vengono dette *pseudo-casuali*.

## 4.2 Generatori di numeri pseudo-casuali uniformi

In generale, per ottenere un valore da una variabile casuale con una specifica distribuzione, si usano uno o più valori pseudo-casuali uniformemente distribuiti nell'intervallo  $(0,1)$ .

Allora, è ovvio che è di fondamentale importanza la generazione di tali valori su di un computer.

I metodi più usati per la generazione di sequenze di numeri pseudocasuali uniformi sono quelli che sfruttano la teoria della congruenza e tra questi quello cosiddetto lineare. Questo metodo si basa su una relazione fondamentale di congruenza che può essere espressa come

$$X_{n+1} = (aX_n + b) \bmod m, \quad n \geq 0 \quad \text{“generatore lineare congruenziale”}$$

ove  $a, b, m$  sono interi non negativi e vanno scelti in maniera opportuna. Una formula di tale tipo consente di ricavare l'elemento  $n+1$ -esimo a partire da quello  $n$ -esimo; da qui l'importanza del valore iniziale  $x_0$  che caratterizza l'intera sequenza e viene perciò detto seme. La sequenza che così si costruisce è distribuita nell'intervallo  $\{0, \dots, m-1\}$ ; come conseguenza, per ottenere sequenze di numeri distribuiti nell'intervallo  $[0,1]$  basta considerare la successione  $\{y_n\}_n$  così ottenuta:

$$y_n = \frac{x_n}{(m-1)}$$

Il periodo delle sequenze costruite sfruttando un metodo di questo tipo è tale da dipendere dalla scelta dei parametri  $a, b, m$  e dal valore iniziale  $x_0$ . E' chiaro che è opportuno scegliere valori che rendono massimo il periodo e comunque esistono dei criteri di scelta per farlo. Ad esempio, poiché il

periodo della sequenza generata non può avere lunghezza maggiore di  $m$ , è preferibile scegliere per  $m$  valori piuttosto grandi.

Per capire meglio, serviamoci di un esempio:

- $m = 10^2$
- $a = 21$
- $b = 7$
- $x_0 = 63$  (il numero di partenza)

In altre parole, genereremo tutti i numeri tra 0 e 99, partendo da 63:

$$x_1 = (21 \times 63 + 7) \bmod 100 = 30$$

$$x_2 = (21 \times 30 + 7) \bmod 100 = 37$$

$$x_3 = (21 \times 37 + 7) \bmod 100 = 84$$

$$x_4 = (21 \times 84 + 7) \bmod 100 = 71$$

$$x_5 = (21 \times 71 + 7) \bmod 100 = 98$$

$$x_6 = (21 \times 98 + 7) \bmod 100 = 65$$

$$x_7 = (21 \times 65 + 7) \bmod 100 = 72$$

... e così via...

Il ciclo generato è limitato dalla macchina che sta eseguendo i calcoli. Perciò, nel caso di un computer con aritmetica a 32 bit (31 bit + il bit del segno), ciò comporta che i numeri che possono essere generati prima che la sequenza si ripeta è al massimo  $2^{31} - 1$ , cioè 2.147.483.647.

Riassumendo, le caratteristiche dei *generatori lineari congruenziali* sono:

#### *Vantaggi*

- Estrema velocità di generazione

#### *Svantaggi*

- Sequenza periodica di periodo al più pari a  $m$ .
- Ogni valore  $x_i$  è completamente determinato dai quattro parametri  $m$ ,  $a$ ,  $c$  e  $x_0$ .
- C'è correlazione fra chiamate successive del generatore

Fra le altre tecniche di generazione di numeri casuali uniformi, va citata sicuramente quella di *Mersenne Twister* che è anche la tecnica da noi utilizzata per le simulazioni.

Mersenne Twister è un algoritmo per la generazione di numeri pseudo-casuali sviluppato nel 1997 da Makoto Matsumoto e Takuji Nishimura.

Secondo gli autori, è un algoritmo che genera un ottimo insieme di numeri pseudo-casuali, supponendo a varie mancanze presenti in altri algoritmi.

Le principali caratteristiche sono:

- Periodo ben più lungo e migliore equidistribuzione: il periodo è  $2^{19937}-1$  e i numeri sono distribuiti in un iperspazio a 623 dimensioni. Questo comporta che la correlazione fra valori successivi è praticamente trascurabile.
- Generazione veloce: la velocità di generazione è sostanzialmente uguale a quella dei generatori lineari congruenziali.
- Uso efficiente della memoria.

### 4.3 Generatori di numeri pseudo-casuali non uniformi

Si è appena visto come far generare ad un computer una sequenza di numeri con distribuzione uniforme, agendo come se ogni numero fosse indipendentemente selezionato a caso tra 0 e 1. A volte le applicazioni dei numeri casuali richiedono altri tipi di distribuzioni: ad esempio come nel nostro caso quella normale. Di solito è piuttosto difficile trovare una legge di formazione originale per una sequenza di numeri pseudo-casuali con distribuzione assegnata e si cerca di ricorrere a procedimenti mediati attraverso la preventiva determinazione di una sequenza uniformemente distribuita.

Attualmente ci sono tante pubblicazioni su diversi "trucchi" che possono essere usati per eseguire tali manipolazioni in modo efficiente, ma non esistono ancora efficienti algoritmi in grado di generare sequenze di numeri con distribuzioni diversa da quella uniforme senza passare per l'uniforme. Comunque è facile, almeno teoricamente, trovare una legge che permetta di passare da una sequenza con distribuzione uniforme ad un'altra con una distribuzione assegnata  $F_X(x)$  strettamente crescente. Se  $F_X(x)$  è strettamente crescente, allora esiste la sua funzione inversa  $F_X^{-1}(x)$ . Sussiste allora la seguente doppia implicazione:

$$U = F_X(X) \Leftrightarrow X = F_X^{-1}(U) \quad (1)$$

Infatti per verificare l'implicazione  $\Rightarrow$  si noti che

$$P(X \leq x) = P(F_X^{-1}(U) \leq x) = P(U \leq F_X(x)) = F_U[F_X(x)]$$

dove  $F_U$  è la funzione di distribuzione di  $U$ .

Essendo  $F_X(x) \in [0,1]$ , dalla relazione precedente si ha che  $P(X \leq x) = F_X(x)$  e cioè quanto si era supposto in partenza.

Analogamente, si può dimostrare l'altra implicazione  $\Leftarrow$

$$P(U \leq u) = P(F_X(X) \leq u) = P(X \leq F_X^{-1}(u)) = F_X[F_X^{-1}(u)] = u$$

Sembrerebbe che questo fornisca una ovvia metodologia di costruzione per la variabile casuale  $X$ . Basterebbe, generata una sequenza  $U_n (n=1,2,..)$  di valori uniformi in  $(0,1)$ , applicare la trasformazione  $X_n = F_X^{-1}(U_n)$  per ottenere un campione della variabile aleatoria  $X$ . Purtroppo, nonostante la singolare semplicità concettuale di questo metodo (“trasformazione inversa”) che consente di passare da una distribuzione uniforme ad una distribuzione qualsiasi, le difficoltà pratiche cui si va incontro quando si tenti di utilizzarla effettivamente, ne sconsigliano quasi sempre l'adozione salvo i casi ove essa risulti particolarmente agevole.

#### **4.3.1 Variabile normale standard**

Nel caso di una variabile casuale normale, dato che la funzione di ripartizione non ha forma esplicita e quindi non è possibile il calcolo della sua inversa, è necessario ricorrere ad altri metodi. Il metodo più famoso, è quello che si basa sul *teorema del limite centrale*.

Tale teorema afferma che se le variabili casuali  $U_n$  con  $n=1,2,..$  sono indipendenti ed identicamente distribuite con valore medio  $a$  e varianza  $b$ , allora la successione di variabili casuali

$$Z_n = \frac{\sum_{i=1}^n U_i - na}{\sqrt{nb}}$$

per  $n \rightarrow \infty$  converge in distribuzione ad una variabile casuale normale standard.

Allora se si scelgono le variabili casuali  $U_n$  uniformi in  $(0,1)$  otteniamo  $a=1/2$  ed  $b=1/12$  e si ha che la successioni di variabili casuali

$$Z_n = \frac{\sum_{i=1}^n U_n - n/2}{\sqrt{n/12}}$$

convergono in distribuzione ad una gaussiana standard. Se in più si sceglie  $n=12$ , allora  $Z_n = \sum_{i=1}^{12} U_i - 6$  fornisce un metodo per generare numeri con distribuzione gaussiana standard a partire da numeri con distribuzione uniforme.

Il compilatore R, per la generazione di dati normali standard, utilizza di default il metodo noto con il nome "Inversion".

Con la tecnica Inversion, sviluppata da Micheal J. Wichurat, si cerca di ottenere, attraverso il rapporto di opportuni polinomi, dati con un'accurata approssimazione alla distribuzione normale standard.

Nel paragrafo successivo, cercheremo di valutare l'efficienza delle due tecniche introdotte.

### 4.3.2 Variabile normale univariata

Il metodo descritto nel paragrafo precedente fornisce un metodo per generare sequenze di numeri con legge gaussiana standard  $Z_n$ . Se si desiderano numeri provenienti da una distribuzione normale generica  $X \approx N(\mu, \sigma)$  è sufficiente ricordare la legge lineare che lega la variabile casuale  $X \approx N(\mu, \sigma)$  alla variabile casuale  $Z \approx N(0,1)$  data da  $X = \sigma Z + \mu$ .

### 4.3.3 Variabile normale multivariata

Quanto detto finora riguarda la generazione di numeri casuali da distribuzioni univariate. Nelle nostre simulazioni risulta indispensabile la generazione di dati da una distribuzione normale multivariata.

Se  $X$  è un qualsiasi vettore aleatorio a  $p$  componenti, per il quale esiste il vettore di medie  $\mu$  e la matrice di varianze e covarianze  $V(X)$ , e se  $Y=AX+b$  è una sua qualsiasi trasformata lineare si ha sempre:

$$E(Y) = A\mu + b \qquad \text{e} \qquad V(Y) = AV(X)A'$$

essendo  $A$  una matrice di  $k \times p$  elementi e  $b$  un vettore colonna di  $K$  elementi. Pertanto, per generare vettori casuali  $Y$  a  $p$  componenti da una normale multivariata di parametri  $\mu_Y$  e  $\Sigma_Y$  e di densità:

$$f(y; \mu_Y, \Sigma_Y) = \frac{1}{\sqrt{(2\pi)^p |\Sigma_Y|}} e^{-\frac{1}{2}[(y-\mu_Y)\Sigma_Y^{-1}(y-\mu_Y)]}$$

è sufficiente determinare prima una matrice  $A$  radice quadrata di  $\Sigma_Y$ , ossia tale che:

$$AA' = \Sigma_Y$$

e quindi generare, con un metodo qualsiasi, un vettore  $X$  costituito da  $p$  numeri pseudo-casuali normali standardizzati indipendenti e porre:

$$Y = AX + \mu_Y$$

Il calcolo della matrice  $A$  tale che:  $AA' = \Sigma_Y$ ; è possibile, data la simmetria e la positività di  $\Sigma_Y$ , mediante la decomposizione di Cholesky, che determina una opportuna matrice  $A$  triangolare inferiore.

## 4.4 Analisi delle simulazioni effettuate con il generatore mvrnorm

### 4.4.1 Confronto tra la stima dell'alpha nominale con l'alpha effettivo

Obiettivo è quello di verificare la bontà delle simulazioni effettuate con il generatore di dati pseudo-casuali mvrnorm. Come fare?

Per prima cosa abbiamo deciso di considerare un test di cui conoscevamo la *distribuzione esatta* in modo da poter valutare l'eventuale differenza della stima dell'alpha effettivo rispetto all'alpha nominale.

Nel nostro caso abbiamo utilizzato il test TRV1 (vedi paragrafo 1.2):

$$\text{TRV1} = \text{LR} = n (\underline{\mu} - \underline{\mu}_0)' \Sigma^{-1} (\underline{\mu} - \underline{\mu}_0)$$

$$\text{sotto } {}_M H_0: \text{TRV1} \sim \chi_p^2$$

Dopo di che, sono state effettuate, ovviamente mediante simulazione, alcune prove sotto le stesse condizioni, facendo variare solamente il numero delle replicazioni.

- numero variabili = 6
- numero osservazioni = 50

**Tab.1 Stima degli alpha effettivi con alpha nominale del 5%**

Replicazioni N	Alpha	Prova1	Prova2	Prova3	Prova4	Prova5	Prova6	Prova7	Prova8	Prova9	Prova10
200'000	0.05	0.05028	0.04950	0.05048	0.04934	<b>0.04878</b>	0.05025	0.05019	0.04999	0.04915	0.05050
100'000	0.05	0.05014	0.05087	0.04946	0.05114	0.04998	0.05045	0.04981	0.05072	0.04991	0.05037
80'000	0.05	0.04888	0.05135	0.05062	0.04990	0.04960	0.05068	<b>0.05161</b>	0.05037	0.04912	0.05015
60'000	0.05	0.05011	0.05156	0.04983	0.04930	0.05010	0.05051	0.04888	0.04966	0.05108	0.05023
50'000	0.05	0.04976	0.04886	0.04862	<b>0.05258</b>	0.05038	0.04942	0.04954	0.04820	0.05058	0.04982
40'000	0.05	0.05025	0.04952	0.05055	0.05062	0.05002	0.05042	0.05105	0.05027	<b>0.05260</b>	0.04985
30'000	0.05	0.05060	0.04976	0.05000	0.04890	0.05233	0.04893	<b>0.05260</b>	0.05030	0.04910	0.05190
20'000	0.05	0.05015	0.05090	0.05040	0.05295	0.04770	0.05035	<b>0.04690</b>	0.04820	0.04980	0.04915
10'000	0.05	0.04660	0.04950	0.05280	0.04860	0.04820	0.05000	0.05320	0.05250	0.05180	0.05070

Sono stati evidenziati in neretto i casi in cui l'alpha stimato non appartiene all'intervallo di confidenza asintotico con probabilità fiduciaria 0.95

Intervallo di confidenza asintotico:

$$p \in \left\{ \hat{p} \pm Z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \right\}$$

dove  $Z_{0.975} = 1.96$

**Tab.2 Intervalli di confidenza asintotici con prob. fiduciaria 0.95**

	10'000	20'000	30'000	40'000	50'000	60'000	80'000	100'000	200'000
Intervallo di confidenza	0.04573 0.05427	0.04698 0.05302	0.04753 0.05247	0.04786 0.05214	0.04809 0.05191	0.04826 0.05174	0.04849 0.05151	0.04865 0.05135	0.04904 0.05095

Da una prima analisi, si nota che aumentando il numero delle replicazioni l'approssimazione dell'alpha stimato migliora ma non sembra convergere "velocemente" al vero valore.

La deviazione standard e la media delle prove effettuate sono:

**Tab.3 Deviazioni standard e medie delle varie prove**

	10'000	20'000	30'000	40'000	50'000	60'000	80'000	100'000	200.000
Media	0.05039	0.04965	0.05044	0.05051	0.04978	0.05013	0.05023	0.05028	0.04985
Deviazione standard	2.199e-3	1.747e-3	1.395e-3	0.847e-3	1.234e-3	0.793e-3	0.888e-3	0.522e-3	0.607e-4

Viene confermata la considerazione precedente. Infatti aumentando le replicazioni il valore della deviazione standard diminuisce e la media si avvicina al vero valore, ma non sembra convergere efficacemente.

Questo però non basta per affermare che le simulazioni effettuate col generatore `mvrnorm` siano buone. Al generatore si richiedono alcune proprietà, in particolare:

- generare sequenze di numeri che seguano la distribuzione considerata
- essere molto efficiente (molto veloce); in quanto abbiamo la necessità di attivarlo molte volte

#### 4.4.2 Test Smirnov-Cramèr-Von Mises

Usualmente, per controllare se la sequenza prodotta da una simulazione è assimilabile ad una particolare distribuzione, si opera in questo modo: generato un certo numero di dati si applica a questi un test di adattamento funzionale.

Nel nostro caso consideriamo il test di Smirnov-Cramèr-Von Mises, che ha lo scopo di valutare la differenza tra la funzione di ripartizione empirica e quella teorica.

$$\omega_n^2 = \int_{-\infty}^{+\infty} (S(x) - F(x))^2 dF$$

$S(x)$  e  $F(x)$  sono rispettivamente la funzione di ripartizione empirica e teorica.

Dalla formula precedente otteniamo:

$$n \omega_n^2 = \frac{1}{12n} + \sum_{i=1}^n \left( F(x_i) - \frac{2i-1}{2n} \right)^2 \quad \text{“Test di Smirnov-Cramèr-Von Mises”}$$

dove  $x_i$  è l'i-mo valore del campione ordinato in senso crescente.

Immaginiamo poi di dimezzare la lunghezza della sequenza su cui si calcola il test. Di volta in volta una simulazione “cattiva” avrà sempre maggiori difficoltà a soddisfare il requisito di equidistribuzione.

Si perviene quindi alla conclusione che testare la distribuzione con lunghe sequenze costituisce un esame dall'esito quasi scontato, mentre il farlo con piccole sequenze può rappresentare un effettivo metodo di discriminazione delle simulazioni.

Vi è un ulteriore motivo che spinge a procedere in questo modo: nelle applicazioni pratiche molte volte si devono utilizzare i generatori propriamente in questo senso. Infatti è spesso necessario disporre di piccole sequenze ciascuna delle quali sia assimilabile ad un campione tratto da una distribuzione uniforme nell'intervallo (0,1).

Pertanto il primo stadio della procedura è quello di generare una piccola sequenza di dati e calcolare per questa un test di adattamento funzionale.

Il risultato ottenuto, espresso in funzione di ripartizione, proviene dalla distribuzione uniforme se la simulazione soddisfa i requisiti.

Naturalmente un tale risultato, preso a sé stante, non può costituire una base di giudizio. Il modo più ovvio di procedere è quello di replicare tale operazione più volte ottenendo una stima della distribuzione del test funzionale e calcolare i p-value stimati e confrontarli con quelli teorici.

Per quanto riguarda la distribuzione asintotica del test si fa uso delle tavole costruite da Anderson e Darling(1952):

**Tab.4 Valori critici della distribuzione limite del test di Smirnov-Cramèr-Von Mises**

Dimensione campione	Quantile 0.95	Quantile 0.99
$n \rightarrow \infty$	0.46136	0.74346

Vediamo ora di analizzare in dettaglio i dati pseudo-casuali generati per le nostre simulazioni.

### 4.4.3 Qualità dei dati Uniformi

Visto l'importanza dei dati pseudo-casuali uniformi, abbiamo deciso di partire analizzando il generatore runif di R.

Per testare la qualità del generatore, andiamo a stimare i p-value del test di Smirnov-Cramèr-Von Mises, sulla base di N=10000 repliche, utilizzando come valori critici i quantili della distribuzione limite, in corrispondenza a vari n.

**Tab.5 Stima degli alpha effettivi con alpha nominale del 5% e 1% con 10.000 repliche**

Dimensione campione	Alpha	Prova1	Prova2	Prova3	Prova4	Prova5	Prova6	Prova7	Prova8	Prova9	Prova10
15	0.05	0.0458	0.0463	0.0508	0.0458	0.0483	0.0522	0.0470	0.0491	0.0498	0.0488
	0.01	0.0093	0.0091	0.0090	0.0085	0.0091	0.0084	0.0103	0.0111	0.0097	0.0093
12	0.05	0.0487	0.0466	0.0483	0.0506	0.0516	0.0526	<b>0.0451</b>	0.0476	0.0463	0.0494
	0.01	0.0085	<b>0.0070</b>	0.0093	0.0101	0.0087	0.0082	<b>0.0078</b>	0.0094	0.0090	0.0106
10	0.05	<b>0.0447</b>	<b>0.0428</b>	0.0520	0.0483	0.0464	<b>0.0447</b>	0.0486	0.0458	0.0494	<b>0.0429</b>
	0.01	<b>0.0070</b>	0.0091	0.0093	0.0085	<b>0.0072</b>	0.0081	0.0085	0.0091	0.0084	<b>0.0069</b>
8	0.05	0.0491	0.0467	0.0464	0.0459	<b>0.0445</b>	0.0485	0.0509	0.0475	<b>0.0434</b>	<b>0.0441</b>
	0.01	<b>0.0072</b>	0.0086	<b>0.0072</b>	0.0085	<b>0.0074</b>	<b>0.0072</b>	0.0086	<b>0.0075</b>	<b>0.0073</b>	<b>0.0062</b>
6	0.05	<b>0.0421</b>	<b>0.0418</b>	<b>0.0454</b>	0.0467	<b>0.0429</b>	<b>0.0413</b>	0.0488	<b>0.0444</b>	0.0471	0.0496
	0.01	<b>0.0050</b>	<b>0.0057</b>	<b>0.0070</b>	<b>0.0077</b>	<b>0.0066</b>	<b>0.0069</b>	0.0083	<b>0.0076</b>	<b>0.0069</b>	0.0080

Sono stati evidenziati in neretto i casi in cui l'alpha stimato non appartiene all'intervallo di confidenza asintotico con probabilità fiduciaria 0.95

Intervallo di confidenza asintotico:

$$p \in \left\{ \hat{p} \pm Z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \right\}$$

dove  $Z_{0.975} = 1.96$

**Tab.6 Intervalli di confidenza asintotici con prob. fiduciaria 0.95**

	Replicazioni	Alpha=0.05	Alpha=0.01
Intervallo di confidenza	10.000	0.04573 0.05427	0.00805 0.01195

La deviazione standard e la media delle prove effettuate sono:

**Tab.7 Deviazioni standard e medie delle varie prove**

	Alpha	Campione n=6	Campione n=8	Campione n=10	Campione n=12	Campione n=15
Medie	0.05	0.04501	0.04670	0.04656	0.04868	0.04839
	0.01	0.00697	0.00757	0.00821	0.00886	0.00938
Deviazione standard	0.05	2.984e-3	2.368e-3	2.979e-3	2.408e-3	2.178e-3
	0.01	1.022e-3	0.773e-3	0.893e-3	1.066e-3	0.813e-3

Come si può vedere, per campioni con numerosità almeno di 15, il generatore runif va bene. Inoltre, non bisogna sottovalutare il fatto che nella stima degli alpha effettivi, abbiamo utilizzato i quantili della distribuzione limite. Questo significa che per campioni di numerosità bassa, l'approssimazione degli alpha effettivi verso gli alpha nominali, potrebbe essere ancora migliore.

Dopo questa breve analisi, possiamo confermare la bontà della tecnica Mersenne Twister utilizzata per generare numeri casuali uniformi.

#### **4.4.4 Qualità dei dati normali univariati**

Passiamo ora all'analisi del generatore rnorm, utilizzando il metodo di default "Inversion".

```
> RNGkind(kind = NULL, normal.kind = NULL)
[1] "Mersenne-Twister" "Inversion"
```

Come abbiamo fatto precedentemente, per testare la qualità del generatore rnorm, andiamo a stimare i p-value del test di Smirnov-Cramèr-Von Mises, sulla base di N=10000 replicazioni, utilizzando come valori critici i quantili della distribuzione limite.

**Tab.8 Stima degli alpha effettivi con alpha nominale del 5% e 1% con 10.000 replicazioni (metodo Inversion)**

Dimensione campione	Alpha	Prova1	Prova2	Prova3	Prova4	Prova5	Prova6	Prova7	Prova8	Prova9	Prova10
30	0.05	0.0503	0.0500	0.0479	0.0468	0.0516	0.0478	0.0541	0.0532	0.0490	0.0494
	0.01	0.0105	0.0090	0.0086	0.0097	0.0093	0.0093	0.0094	0.0102	0.0098	0.0096
25	0.05	0.0498	0.0497	0.0535	0.0499	0.0485	0.0473	0.0474	0.0473	0.0490	0.0493
	0.01	0.0098	0.0108	0.0106	0.0095	0.0086	<b>0.0076</b>	0.0089	0.0085	0.0090	0.0105
20	0.05	0.0506	0.0531	0.0474	0.0485	0.0511	0.0476	0.0488	0.0494	0.0486	0.0520
	0.01	0.0099	0.0103	<b>0.0074</b>	<b>0.0073</b>	0.0091	0.0096	0.0098	0.0111	0.0104	0.0085
15	0.05	0.0487	0.0475	0.0486	0.0491	0.0519	0.0518	<b>0.0428</b>	0.0485	<b>0.0450</b>	<b>0.0439</b>
	0.01	0.0086	0.0086	0.0094	0.0115	0.0109	0.0095	0.0092	<b>0.0069</b>	0.0082	0.0084
10	0.05	0.0474	<b>0.0447</b>	0.0475	0.0510	0.0475	<b>0.0457</b>	0.0483	<b>0.0456</b>	0.0495	<b>0.0466</b>
	0.01	0.0081	0.0083	<b>0.0068</b>	<b>0.0079</b>	<b>0.0079</b>	<b>0.0077</b>	0.0083	0.0092	<b>0.0080</b>	<b>0.0072</b>
8	0.05	0.0466	<b>0.0438</b>	0.0477	0.0508	0.0472	0.0462	0.0472	<b>0.0446</b>	<b>0.0440</b>	<b>0.0435</b>
	0.01	<b>0.0073</b>	<b>0.0064</b>	<b>0.0071</b>	0.0100	<b>0.0079</b>	<b>0.0071</b>	<b>0.0076</b>	<b>0.0059</b>	<b>0.0072</b>	0.0081
6	0.05	0.0507	<b>0.0445</b>	<b>0.0473</b>	<b>0.0423</b>	<b>0.0444</b>	<b>0.0455</b>	0.0479	<b>0.0425</b>	<b>0.0443</b>	<b>0.0443</b>
	0.01	0.0082	<b>0.0073</b>	<b>0.0076</b>	<b>0.0064</b>	<b>0.0055</b>	0.0082	0.0084	<b>0.0060</b>	<b>0.0079</b>	<b>0.0067</b>

Sono stati evidenziati in neretto i casi in cui l'alpha stimato non appartiene all'intervallo di confidenza asintotico con probabilità fiduciaria 0.95

**Tab.9 Deviazioni standard e medie delle varie prove**

	Alpha	Campione n=6	Campione n=8	Campione n=10	Campione n=15	Campione n=20	Campione n=25	Campione n=30
Medie	0.05	0.04537	0.04616	0.04738	0.04778	0.04971	0.04917	0.05001
	0.01	0.00722	0.00746	0.00794	0.00912	0.00934	0.00938	0.00954
Deviazione standard	0.05	2.586e-3	2.260e-3	1.893e-3	3.065e-3	1.911e-3	1.843e-3	2.369e-3
	0.01	1.017e-3	1.104e-3	0.648e-3	1.328e-3	1.265e-3	1.047e-3	0.554e-3

Si può facilmente notare che per campioni di dimensione maggiore di 20, il generatore rnorm funziona abbastanza bene. Come nelle aspettative, al generatore rnorm servono campioni di numerosità più grande rispetto al generatore runif, per avere un buon accostamento alla distribuzione normale. Questo deriva dal fatto che per campioni di numerosità  $n \rightarrow \infty$  i dati convergono ad una distribuzione normale.

Per nostra scrupolosità, proviamo ora a confrontare il metodo "Inversion" con il metodo spiegato precedentemente, basato sul teorema del limite centrale.

**Tab.10 Stima degli alpha effettivi con alpha nominale del 5% e 1% con 10.000 replicazioni (metodo Teorema del limite centrale)**

Dimensione campione	Alpha	Prova1	Prova2	Prova3	Prova4	Prova5	Prova6	Prova7	Prova8	Prova9	Prova10
30	0.05	0.0472	0.0498	0.052	0.0494	0.0474	0.0494	0.0530	0.0482	0.0506	0.0508
	0.01	0.0089	0.0101	0.0107	0.0088	0.0093	0.0096	0.0105	0.0096	0.0080	0.0105
25	0.05	0.0505	0.0526	0.0488	0.0502	<b>0.0445</b>	0.0521	0.0496	0.0484	0.0497	0.0468
	0.01	0.0093	0.0089	0.0097	0.0103	<b>0.0078</b>	0.0118	0.0108	0.0095	0.0090	0.0085
20	0.05	0.0507	0.0474	0.0490	0.0477	0.0502	0.0489	0.0514	0.0488	0.0491	0.0473
	0.01	0.0096	0.0097	0.0104	0.0090	0.0100	0.0080	0.0091	0.0097	0.0104	0.0090
15	0.05	0.0494	0.0479	0.0509	0.0506	0.0485	0.0467	0.0502	0.0515	0.0478	0.0457
	0.01	0.0083	<b>0.0075</b>	0.0110	0.0101	0.0089	<b>0.0074</b>	<b>0.0079</b>	0.0109	0.0091	<b>0.0077</b>
10	0.05	0.0491	0.0461	0.0462	<b>0.0446</b>	0.0528	0.0505	0.0492	<b>0.0439</b>	0.0473	0.0502
	0.01	0.0086	0.0089	<b>0.0074</b>	<b>0.0075</b>	0.0086	0.0085	0.0087	<b>0.0062</b>	0.0098	<b>0.0077</b>
8	0.05	0.0481	0.0475	0.0485	0.0459	0.0478	<b>0.0454</b>	0.0461	0.0469	0.0467	0.0478
	0.01	<b>0.0074</b>	0.0089	0.0092	0.0087	<b>0.0070</b>	<b>0.0077</b>	<b>0.0070</b>	<b>0.0076</b>	0.0094	0.0090
6	0.05	0.0481	0.0469	0.0490	0.0462	0.0509	0.0473	0.0488	<b>0.0448</b>	0.0487	<b>0.0426</b>
	0.01	0.0080	<b>0.0064</b>	0.0081	<b>0.0070</b>	<b>0.0070</b>	<b>0.0074</b>	<b>0.0070</b>	<b>0.0069</b>	<b>0.0069</b>	<b>0.0066</b>

La deviazione standard e la media delle prove effettuate sono:

**Tab.11 Deviazioni standard e medie delle varie prove**

	Alpha	Campione n=6	Campione n=8	Campione n=10	Campione n=15	Campione n=20	Campione n=25	Campione n=30
Medie	0.05	0.04733	0.04707	0.04799	0.04892	0.04905	0.04932	0.04978
	0.01	0.00713	0.00819	0.00819	0.00888	0.00949	0.00956	0.00960
Deviazione standard	0.05	2.366e-3	1.033e-3	2.838e-3	1.915e-3	1.385e-3	2.393e-3	1.889e-3
	0.01	0.551e-3	0.939e-3	1.004e-3	1.370e-3	0.735e-3	1.162e-3	0.873e-3

Come si può vedere, il metodo basato sul teorema del limite centrale, che a nostro parere è uno dei migliori, sembra andare un po' meglio rispetto al metodo Inversion per campioni di bassa numerosità. Purtroppo non sappiamo valutare con certezza questo miglioramento, perché nel calcolo degli alpha effettivi abbiamo utilizzato i quantili della distribuzione limite.

In conclusione, sono entrambi dei buoni metodi per la generazione di dati normali univariati.

Abbiamo deciso di utilizzare nelle nostre simulazioni il metodo Inversion per la sua maggiore efficienza in termini di tempo.

#### 4.4.5 Qualità dei dati normali multivariati

Per valutare la qualità dei dati normali multivariati, ottenibili col generatore `mvrnorm`, abbiamo deciso di utilizzare il test `mvrnorm.etest()`, contenuto nella libreria `energy` sviluppata da Maria L. Rizzo e Gabor J. Székely (2005).

Il test consiste nel confrontare distribuzioni multivariate basandosi sulla distanza euclidea dei dati.

Supponiamo  $X_1, \dots, X_n$  siano vettori  $d$ -variati provenienti dalla distribuzione  $F$ , e  $x_1, \dots, x_n$  i valori osservati dei vettori suddetti. Si vuole verificare l'ipotesi:

- $H_0: F = F_0$
- $H_1: F \neq F_0$

Il test per la verifica dell'ipotesi in questione è:

$$E = n \left( \frac{2}{n} \sum_{j=1}^n E \|x_j - X\| - E \|X - X'\| - \frac{1}{n^2} \sum_{j,k=1}^n \|x_j - x_k\| \right)$$

dove  $X$  e  $X'$  sono vettori  $d$ -variati indipendenti e identicamente distribuiti (iid), con distribuzione  $F_0$ . Se la distribuzione è una normale  $d$ -variata con vettore di medie  $\mu$  e matrice di varianza e covarianza  $\Sigma$ , allora si considera la seguente trasformazione:

$$y_j = \Sigma^{-1/2} (x_j - \mu)$$

Il test diventa:

$$E = n \left( \frac{2}{n} \sum_{j=1}^n E \|y_j - Z\| - E \|Z - Z'\| - \frac{1}{n^2} \sum_{j,k=1}^n \|y_j - y_k\| \right)$$

dove  $Z$  e  $Z'$  sono iid  $N_d(0, I)$ , e  $I$  è un matrice identità  $d \times d$ .

Per il calcolo del test viene utilizzata il bootstrap parametrico con R repliche (di default R=999).

Vediamo ora di analizzare in dettaglio la “qualità” del generatore mvnorm, utilizzando il test mvnorm.etest.

Per prima cosa abbiamo generato una matrice di dati di dimensione  $X_{30 \times 2}$  e calcolato il test:

```
> media
[1] 0 0
> sigma
      [,1] [,2]
[1,]    1    0
[2,]    0    1
> dati<-mvnorm(n=30,media,sigma)
> mvnorm.etest(dati)

      Energy test of multivariate normality: estimated
parameters

data:  x, sample size 30, dimension 2, replicates 999
E-statistic = 0.4942, p-value = 0.8789
```

Come si può vedere, viene ampiamente accettata l'ipotesi che la matrice di dati  $X_{30 \times 2}$  generata da mvnorm provenga da una normale bivariata.

Naturalmente un tale risultato, preso a sé stante, non può costituire una base di giudizio. Allora abbiamo deciso di replicare tale operazione N=1000, ottenendo così una stima del p-value per la generazione di matrici  $X_{30 \times 2}$ .

Utilizzando la funzione test.multivariato dell'appendice, si è ottenuto:

```
> test.multivariato(n=30,media,sigma,N=1000)
[1] 0.5026967
```

Questo conferma che i dati generati si possono considerare provenienti da una normale bivariata.

Proviamo ora a verificare per diverse numerosità l'accostamento alla distribuzione normale multivariata, utilizzando sempre la funzione test.multivariato con N=1000 replicazioni.

**Tab.12 P-value delle varie prove**

Numerosità	Variabili	p-value
30	2	0.5026967
25	2	0.4984705
20	2	0.5013201
15	2	0.4972150
10	2	0.4958326
6	2	0.5106697
3	2	0.5152452

Come dalle aspettative, il generatore mvnorm non ha problemi nel generare dati normali multivariati. Infatti utilizza il generatore rnorm per generare dati normali standardizzati e poi tramite la decomposizione di Cholesky, genera dati normali multivariati.

Per avere un'ulteriore conferma, proviamo ad utilizzare il test di Smirnov-Cramer-Von Mises, sui dati normali multivariati  $X_{30 \times 2}$ , trasformati con:

$$y_j = \Sigma^{-1/2}(x_j - \mu)$$

ottenendo così 2 normali univariate.

**Tab.13 Stima degli alpha effettivi con alpha nominale del 5% e 1%**

Dimensione campione	Alpha	Prova1	Prova2	Prova3	Prova4	Prova5	Prova6	Prova7	Prova8	Prova9	Prova10
45	0.05	0.0487	0.0490	0.0521	0.0491	0.0502	0.0496	0.0490	0.0511	0.0489	0.0498
	0.01	0.0108	0.0097	0.0107	0.0090	0.0100	0.0088	0.0093	0.0094	0.0095	0.0100
40	0.05	<b>0.0545</b>	0.0502	0.0506	0.0471	0.0499	0.0531	0.0465	0.0528	0.0479	0.0477
	0.01	0.0110	0.0101	0.0095	0.0103	0.0093	0.0108	0.0089	0.0093	0.0099	0.0090
35	0.05	0.0514	0.0493	0.0482	0.0491	0.0510	0.0470	0.0496	0.0491	0.0509	0.0503
	0.01	0.0101	<b>0.0080</b>	0.0090	0.0091	0.0099	0.0098	0.0106	0.0089	0.0088	0.0107
30	0.05	0.0521	0.0485	0.0515	0.0472	0.0509	0.0494	0.0503	0.0478	<b>0.0456</b>	0.0491
	0.01	0.0108	0.0100	0.0114	0.0084	0.0100	0.0100	0.0115	0.0102	0.0107	<b>0.0079</b>
25	0.05	<b>0.0441</b>	<b>0.0423</b>	0.0525	0.0525	0.0487	0.0507	0.0519	0.0469	0.0460	0.0488
	0.01	<b>0.0068</b>	0.0084	0.0110	0.0113	0.0091	0.0109	0.0100	0.0094	0.0086	0.0092
20	0.05	0.0487	0.0475	0.0477	0.0528	0.0519	<b>0.0546</b>	0.0463	0.0474	<b>0.0452</b>	0.0514
	0.01	0.0093	0.0103	0.0103	0.0099	0.0086	0.0094	<b>0.0075</b>	0.0082	0.0094	0.0089
15	0.05	0.0462	0.0479	0.0461	0.0478	<b>0.0452</b>	0.0472	0.0466	0.0508	0.0516	0.0476
	0.01	0.0091	0.0093	<b>0.0078</b>	0.0093	0.0085	0.0085	<b>0.0073</b>	0.0099	0.0094	<b>0.0080</b>
10	0.05	0.0476	0.0489	0.0479	0.0502	0.0486	0.0538	<b>0.0454</b>	0.0508	<b>0.0456</b>	0.0473
	0.01	<b>0.0074</b>	<b>0.0080</b>	<b>0.0075</b>	0.0085	0.0085	0.0090	0.0087	<b>0.0075</b>	<b>0.0074</b>	0.0087
8	0.05	0.0503	0.0476	0.0471	0.0492	<b>0.0448</b>	<b>0.0449</b>	0.0524	<b>0.0452</b>	<b>0.0453</b>	0.0462
	0.01	<b>0.0079</b>	<b>0.0070</b>	<b>0.0077</b>	0.0085	<b>0.0066</b>	0.0086	<b>0.0080</b>	<b>0.0080</b>	<b>0.0078</b>	<b>0.0076</b>
6	0.05	0.0472	0.0465	<b>0.0440</b>	0.0478	0.0463	0.0462	0.0479	<b>0.0432</b>	<b>0.0446</b>	0.0469
	0.01	<b>0.0071</b>	<b>0.0075</b>	<b>0.0071</b>	<b>0.0060</b>	<b>0.0068</b>	<b>0.0068</b>	<b>0.0080</b>	<b>0.0067</b>	<b>0.0075</b>	<b>0.0077</b>

Sono stati evidenziati in neretto i casi in cui l'alpha stimato non appartiene all'intervallo di confidenza asintotico con probabilità fiduciaria 0.95

**Tab.14 Deviazioni standard e medie delle varie prove**

	Alpha	Campione n=6	Campione n=8	Campione n=10	Campione n=15	Campione n=20
Medie	0.05	0.04606	0.04730	0.04861	0.04770	0.04935
	0.01	0.00712	0.00777	0.00812	0.00871	0.00918
Deviazione standard	0.05	1.607e-3	2.590e-3	2.515e-3	2.038e-3	3.110e-3
	0.01	0.581e-3	0.609e-3	0.628e-3	0.826e-3	0.903e-3
	Alpha	Campione n=25	Campione n=30	Campione n=35	Campione n=40	Campione n=45
Medie	0.05	0.04844	0.04924	0.04959	0.05003	0.04975
	0.01	0.00947	0.01009	0.00949	0.00981	0.00972
Deviazione standard	0.05	3.576e-3	2.034e-3	1.362e-3	2.758e-3	1.103e-3
	0.01	1.385e-3	1.167e-3	0.867e-3	0.732e-3	0.664e-3

Come si può vedere, per campioni di dimensione maggiore di 45, il generatore mvrnorm funziona abbastanza bene. Questo conferma le nostre attese, visto che per la normale univariata siamo nell'ordine di campioni di numerosità 30, era sensato aspettarsi un aumento trattando in questo caso di normali multivariate.

#### 4.4.6 Qualità delle simulazioni

Prendiamo in considerazione, come all'inizio del paragrafo, il test TRV1, di cui conosciamo la distribuzione esatta.

$$\text{sotto } {}_M H_0: \text{TRV1} \sim \chi_p^2$$

Dopo aver valutato la bontà del generatore mvrnorm, passiamo ad analizzare la qualità dell'accostamento del test TRV1 alla distribuzione  $\chi_p^2$ .

Come nei casi precedenti, andiamo a testare la qualità delle simulazioni, andando a stimare i p-value del test di Smirnov-Cramèr-Von Mises, sulla base di N=10000 replicazioni, utilizzando come valori critici i quantili della distribuzione limite.

**Tab.15 Stima degli alpha effettivi con alpha nominale del 5% e 1% con 10.000 repliche (matrici di dati  $X_{nx2}$ )**

Dimensione Campione	Alpha	Prova1	Prova2	Prova3	Prova4	Prova5	Prova6	Prova7	Prova8	Prova9	Prova10
60	0.05	0.0493	0.0511	0.0518	0.0540	0.0485	0.0522	0.0542	0.0512	0.0524	0.0513
	0.01	0.0100	0.0107	0.0114	0.0105	0.0090	0.0110	0.0118	0.0099	0.0108	0.0096
40	0.05	0.0543	0.0536	0.0529	<b>0.0549</b>	0.0540	0.0475	0.0507	0.0518	0.0508	0.0496
	0.01	0.0115	0.0094	0.0103	0.0117	0.0112	0.0098	0.0113	<b>0.0121</b>	0.0091	0.0088
30	0.05	0.0532	0.0525	0.0542	0.0517	0.0534	<b>0.0551</b>	0.0535	0.0496	<b>0.0606</b>	0.0512
	0.01	0.0091	0.0109	0.0100	0.0096	0.0105	<b>0.0125</b>	<b>0.0122</b>	0.0101	<b>0.0126</b>	0.0109
20	0.05	0.0533	<b>0.0548</b>	<b>0.0556</b>	<b>0.0565</b>	<b>0.0586</b>	<b>0.0566</b>	<b>0.0584</b>	<b>0.0580</b>	<b>0.0581</b>	0.0542
	0.01	0.0094	<b>0.0120</b>	<b>0.0108</b>	<b>0.0135</b>	<b>0.0130</b>	<b>0.0127</b>	0.0117	0.0118	0.0115	0.0106

Anche in questo caso, sono stati evidenziati in neretto i casi in cui l'alpha stimato non appartiene all'intervallo di confidenza asintotico con probabilità fiduciaria 0.95

**Tab.16 Deviazioni standard e medie delle varie prove**

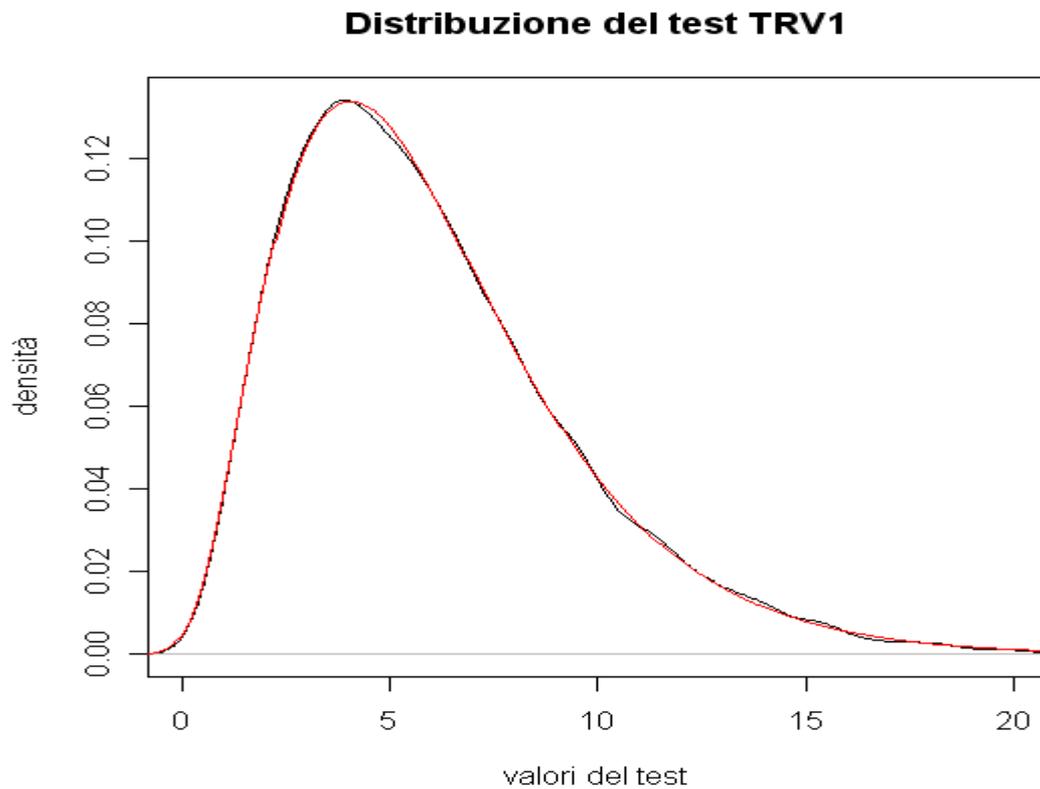
	Alpha	20	30	40	60
Medie	0.05	0.05641	0.05350	0.05201	0.05160
	0.01	0.01170	0.01084	0.01052	0.01047
Deviazione standard	0.05	1.885e-3	2.949e-3	2.363e-3	1.793e-3
	0.01	1.219e-3	1.231e-3	1.188e-3	0.852e-3

Dalla visione dei risultati, si può affermare che per il test TRV1 con numerosità campionaria di almeno 60, abbiamo un buon accostamento alla distribuzione  $\chi_p^2$ .

Visto che per normali multivariate siamo nell'ordine di campioni di numerosità 45, era sensato aspettarsi un valore numerosità campionaria  $\geq 45$ .

Dalle analisi effettuate possiamo affermare che le simulazioni con N=13.000 repliche, che è il valore di default delle nostre funzioni, sono sufficienti per avere un buon accostamento alle distribuzioni dei test considerati in questo lavoro.

Questo viene confermato anche dal grafico sottostante, in cui andiamo a simulare la distribuzione del test TRV1, utilizzando matrici di dati  $X_{30 \times 6}$ .



linea nera      ➔ distribuzione simulata  
linea rossa     ➔ distribuzione esatta

Come si può vedere, la distribuzione simulata con  $N=13.000$  replicazioni del test TRV1, si discosta di poco dalla distribuzione esatta  $\chi_6^2$ .

#### 4.4.7 Tempi di esecuzione

Lo scopo di questo lavoro consisteva nella progettazione e realizzazione di due librerie che, data una matrice normale di dati  $X_{n \times p}$ , calcolasse il test per la verifica d'ipotesi d'interesse e che quando non fosse nota la vera distribuzione del test, lasciasse la più ampia libertà all'utente nell'interpretazione dei risultati ottenuti.

Poiché le scelte fatte sono tutte opinabili, si è preferito realizzare delle librerie facili da comprendere, modificare e in grado di raggiungere dei risultati sufficientemente precisi in tempo reale.

Per noi, dunque, l'efficienza è correlata al tempo di esecuzione, che dipende dalla macchina sulla quale il programma viene eseguito, dal *particolare generatore di dati* usato e dal grado di approssimazione dei risultati.

Si sottolinea che le approssimazioni dei risultati dipendono dal numero  $N$  di repliche usate nella simulazione, che a livello di principio possono essere aumentate fino a quanto si vuole.

Per ridurre il tempo d'attesa, si è deciso di perdere nell'accuratezza dei risultati, utilizzando un numero ragionevole di repliche .

Per comprendere meglio faremo un esempio. Consideriamo una matrice normale di dati  $X_{50 \times 6}$  . Si vuole verificare l'ipotesi:

- $MH_0: \quad \Sigma = \Sigma_0 \quad \rightarrow$  con  $\mu$  non nota
- $MH_1: \quad \Sigma \neq \Sigma_0$

dove

$\Sigma_0$

	[ ,1]	[ ,2]	[ ,3]	[ ,4]	[ ,5]	[ ,6]
[1,]	0.12	0.10	0.00	0.0	0.1	0.03
[2,]	0.10	0.14	0.00	0.0	0.0	0.00
[3,]	0.00	0.00	0.26	0.1	0.0	0.00
[4,]	0.00	0.00	0.10	0.1	0.0	0.00
[5,]	0.10	0.00	0.00	0.0	0.4	0.10
[6,]	0.03	0.00	0.00	0.0	0.1	0.10

Si suppone di usare la funzione `TRV3("simulazione",dati,sigma0)`.  
 Per vedere il tempo di esecuzione della funzione, utilizziamo l'istruzione `system.time`.

```
> system.time(TRV3("simulazione",dati,sigma))
[1] "----*---*---*---*---Risultati del test*---*---*---*---*---*---*---*---"
      user  system elapsed
      7.40   0.17   7.94
```

Il tempo di esecuzione è di 7.40 secondi, eseguendo la simulazione con un pentium4 (3GHz e 512 MB di Ram).

Nell'esempio fatto, si è utilizzato un errore assoluto  $d=5e-3$  e  $p=5e-2$ , che equivale a  $N=12549$  e approssimato a  $N=13.000$  replicazioni.

Se volessimo usare  $p=0.5$ , che corrisponde alla massima variabilità, il numero di replicazioni aumenterebbe a 66.000, che comporterebbero un tempo di esecuzione abbastanza oneroso. Vale la pena di sottolineare che i risultati così ottenuti sarebbero però migliori rispetto a quelli precedenti. Proviamo a verificare i tempi di esecuzione facendo variare il numero delle replicazioni.

**Tab.17 Stima dei tempi di esecuzione, simulando con matrici di dati  $X_{50 \times 6}$**

Replicazioni	Tempo di esecuzione in sec
N=10.000	5,20
N=13.000	7,40
N=20.000	12,50
N=30.000	22,22
N=40.000	34,52
N=50.000	48,39
N=66.000	64,11

Naturalmente è lasciata piena libertà all'utente di modificare a suo piacimento i parametri (d e p).

#### 4.4.8 Conclusioni

Dopo questa breve analisi, possiamo affermare che le simulazioni effettuate con il generatore mvrnorm sono “abbastanza” efficienti anche per un numero di replicazioni relativamente basso.

Purtroppo, abbiamo verificato che per un numero elevato di replicazioni, nell'ordine delle 100.000, la simulazione mantiene delle “oscillazioni” nella stima dei veri valori della distribuzione. Queste “oscillazioni” però, a nostro avviso, non influiscono in maniera pesante sulla qualità delle nostre simulazioni, ma non vanno assolutamente trascurate.

Inoltre l'incremento della velocità, dovuto sia a macchine più avanzate che a migliori programmi, accrescerà negli anni futuri la capacità di trattare questi problemi con maggior precisione.

# CAPITOLO 5

## UN'APPLICAZIONE PRATICA

### 5.1 I Packages TRV e UIT

Procediamo, quindi, con la presentazione del package `TRV` che abbiamo sviluppato. Per poter accedere alle funzioni della libreria `TRV` è necessario, all'inizio di ogni sessione, caricarle in memoria con il comando

```
> library(TRV)
```

Al caricamento, `TRV` carica anche la libreria `MASS`, rendendo così disponibile il generatore di dati `mvrnorm`, indispensabile per le nostre simulazioni.

La libreria `TRV`, permette all'utente, di poter eseguire alcuni test rapporto verosimiglianza.

Per vedere quali test sono possibili e il nome delle funzioni che li eseguono, basta chiamare l'aiuto in linea digitando il comando:

```
> library(help=TRV)
```

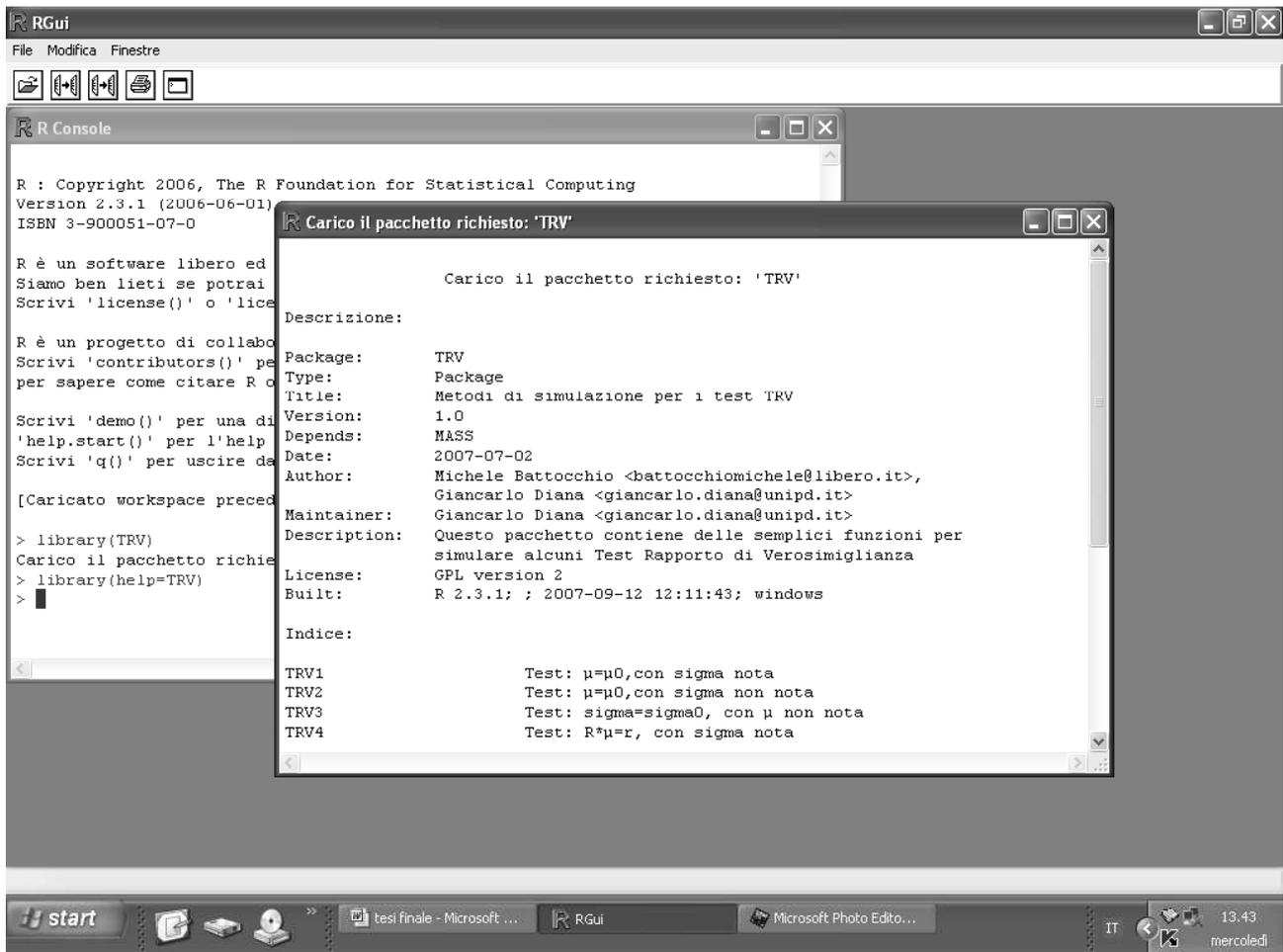


Figura 1: Help in linea del packages TRV

Come si può vedere dalla figura, con il comando `library(help=TRV)`, si richiama non solo il nome delle possibili funzioni test, ma anche molte ulteriori informazioni utili per l'utente.

In dettaglio, quello che si ottiene è:

**Descrizione:**

```
Package:      TRV
Type:        Package
Title:       Metodi di simulazione per i test TRV
Version:     1.0
Depends:     MASS
Date:        2007-07-02
Author:      Michele Battocchio <battocchiomichele@libero.it>,
             Giancarlo Diana <giancarlo.diana@unipd.it>
```

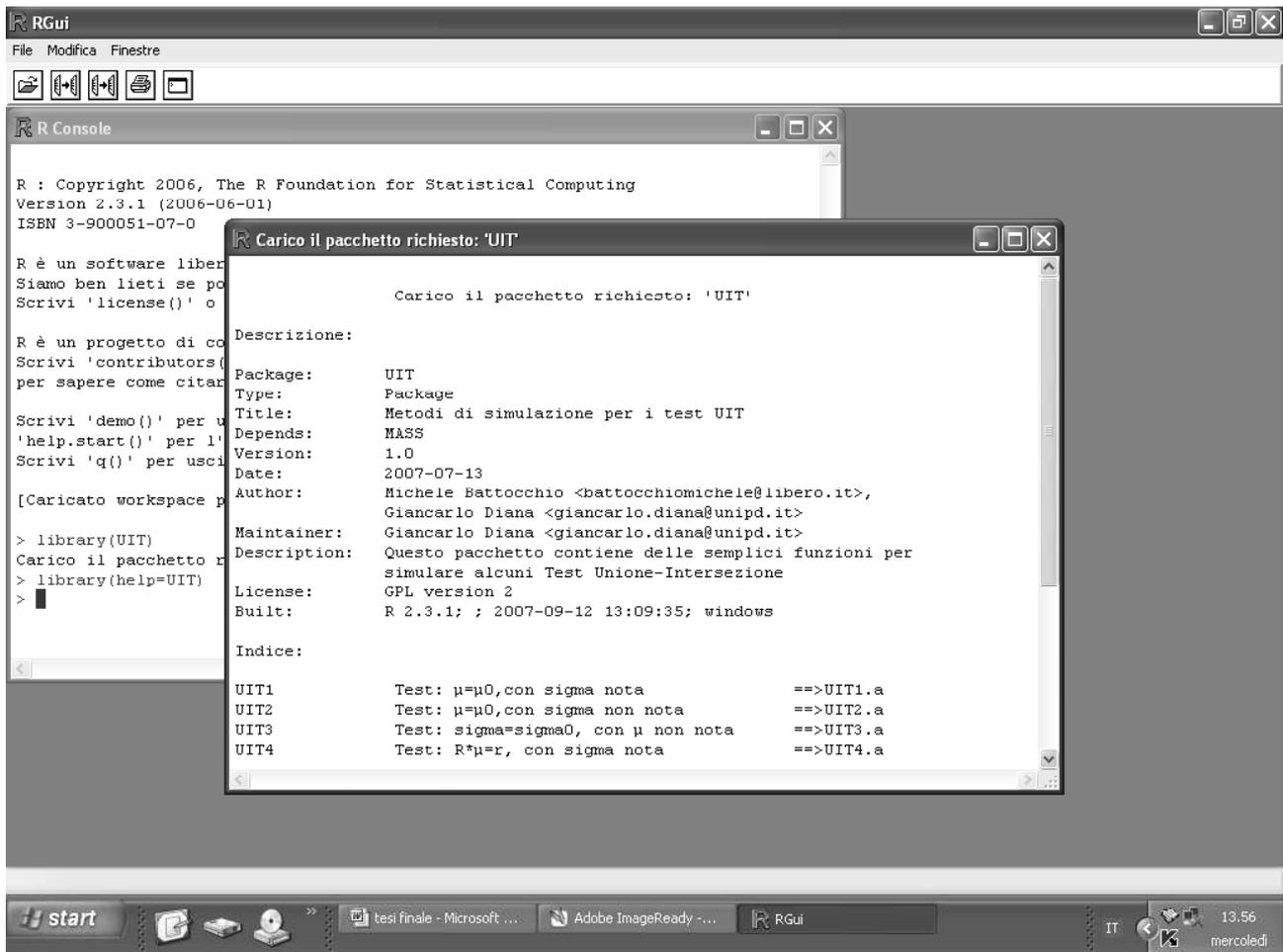
Maintainer: Giancarlo Diana <giancarlo.diana@unipd.it>  
Description: Questo pacchetto contiene delle semplici funzioni per  
simulare alcuni Test Rapporto di Verosimiglianza  
License: GPL version 2  
Built: R 2.3.1; ; 2007-09-12 12:11:43; windows

### Indice:

TRV1	Test: $\mu=\mu_0$ , con sigma nota
TRV2	Test: $\mu=\mu_0$ , con sigma non nota
TRV3	Test: $\sigma=\sigma_0$ , con $\mu$ non nota
TRV4	Test: $R*\mu=r$ , con sigma nota
TRV5	Test: $R*\mu=r$ , con sigma non nota
TRV6	Test: $\sigma=k*\sigma_0$ , con $\mu$ e $K$ non note
TRV72	Test: $\sigma^2=I$ , con $\mu$ non nota
TRV7	Test: $\sigma=I$ , con $\mu$ non nota
TRV82	Test: $\mu_1=\mu_2$ , con $\sigma_1=\sigma_2$
TRV8	Test: $\mu_1=..\mu_g$ , con $\sigma_1=..\sigma_g$
TRV9	Test: $\sigma_1=..\sigma_g$ , test di Box
TRV10	Test: $\mu_1=..\mu_g$ , e $\sigma_1=..\sigma_g$

La stessa cosa è stata pensata per i test Unione-Intersezione. Infatti, dopo aver opportunamente caricato la libreria UIT, digitando l'istruzione `library(help=UIT)`, otteniamo a video:

```
> library(help=UIT)
```



**Figura 2: Help in linea del packages UIT**

Anche in questo caso, come si può vedere, con il comando `library(help=UIT)`, non si richiama solo il nome delle possibili funzioni test, ma anche molte ulteriori informazioni utili per l'utente.

In dettaglio:

**Descrizione:**

```

Package:      UIT
Type:         Package
Title:        Metodi di simulazione per i test UIT
Depends:      MASS
Version:      1.0
Date:         2007-07-13
Author:       Michele Battocchio <battocchiomichele@libero.it>,
              Giancarlo Diana <giancarlo.diana@unipd.it>

```

Maintainer: Giancarlo Diana <giancarlo.diana@unipd.it>  
Description: Questo pacchetto contiene delle semplici funzioni per simulare alcuni Test Unione-Intersezione  
License: GPL version 2  
Built: R 2.3.1; ; 2007-09-12 13:09:35; windows

### Indice:

UIT1	Test: $\mu=\mu_0$ , con sigma nota	==>UIT1.a
UIT2	Test: $\mu=\mu_0$ , con sigma non nota	==>UIT2.a
UIT3	Test: $\sigma=\sigma_0$ , con $\mu$ non nota	==>UIT3.a
UIT4	Test: $R*\mu=r$ , con sigma nota	==>UIT4.a
UIT5	Test: $R*\mu=r$ , con sigma non nota	==>UIT5.a
UIT72	Test: $\sigma_1=I$ , con $\mu$ non nota	==>UIT72.a
UIT82	Test: $\mu_1=\mu_2$ , con $\sigma_1=\sigma_2$	==>UIT82.a
UIT8	Test: $\mu_1=..=\mu_g$ , con $\sigma_1=..=\sigma_g$	==>UIT8.a

Nei prossimi due paragrafi, faremo degli esempi di alcuni test delle due librerie, utilizzando il data-set “iris3” presente in R. Questo data-set contiene le misurazioni in centimetri delle variabili lunghezza e larghezza dei sepali e dei petali, di 150 fiori provenienti da 3 specie diverse. Le specie sono setosa, versicolor e virginica.

Dopo averlo opportunamente caricato, suddividiamo le popolazioni oggetto di studio:

```
data(iris3)
setosa<-iris3[, ,1]
versicolor<-iris3[, ,2]
virginica<-iris3[, ,3]
```

## 5.2 Applicazione della libreria TRV sul data-set “iris3”

### 5.2.1 Test d’ipotesi $\mu=\mu_0$ , con $\Sigma$ nota “TRV1”

Consideriamo solo la popolazione setosa, con tutte le sue variabili (lunghezza e larghezza dei petali e dei sepali).

Supponiamo che i dati si distribuiscano come  $N_p(\underline{\mu}, \Sigma)$  con  $\Sigma$  nota.

Dato un vettore medie  $\underline{\mu}_0=(5, 3.4, 1.4, 0.24)$ , vogliamo verificare l’ipotesi:

- ${}_M H_0: \quad \underline{\mu} = \underline{\mu}_0 \quad \rightarrow$  con  $\Sigma$  nota
- ${}_M H_1: \quad \underline{\mu} \neq \underline{\mu}_0$

Il test è:

$$\text{TRV1} = \text{LR} = n (\underline{\mu} - \underline{\mu}_0)' \Sigma^{-1} (\underline{\mu} - \underline{\mu}_0)$$

sotto  ${}_M H_0: \text{TRV1} \sim \chi_p^2$

Per effettuare tale test abbiamo costruito la funzione, “TRV1”.

Digitando `help(TRV1)`, si richiama una breve descrizione della funzione e si vede quali parametri sono necessari per il corretto funzionamento.

Funzione: **TRV1**

Parametri:

- **dati:** matrice di dati
- **$\mu_0$ :** vettore medie  $\mu_0$
- **sigma:** matrice di varianza e covarianza
- **alpha:** ampiezza del test (impostato di default pari a 0.05)
- **grafico:** valore logico. Se è 'TRUE' disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

La funzione dopo aver calcolato il test, stamperà a video:

- il valore del test
- i gradi di libertà
- il p-value del test

Applicazione:

```
setosa<-iris3[, ,1]
apply(setosa,2,mean)
Sepal L. Sepal W. Petal L. Petal W.
  5.006    3.428    1.462    0.246
```

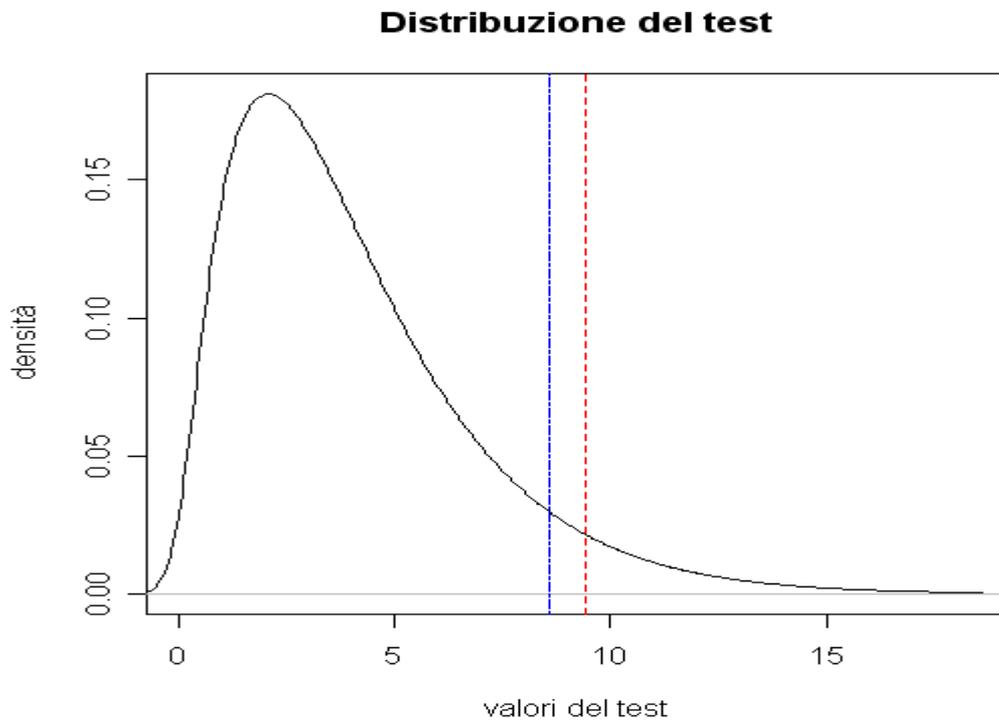
$\mu_0 <- c(5, 3.4, 1.4, 0.24)$  → vettore ipotesi  $M_{H0}$

sigma → matrice var. cov. nota

```
          Sepal L. Sepal W. Petal L. Petal W.
Sepal L. 0.12      0.10      0.02      0.01
Sepal W. 0.10      0.15      0.01      0.01
Petal L. 0.02      0.01      0.03      0.01
Petal W. 0.01      0.01      0.01      0.028
```

Adesso tramite la funzione TRV1, eseguo il test :

```
TRV1(setosa,μ0,sigma,grafico="TRUE")
```



Pause. Press <Enter> to continue...

```
[1] "-----Risultati del test-----"
```

```
$test
```

```
      [,1]  
[1,] 8.628002
```

```
$g1
```

```
[1] 4
```

```
$`p-value`
```

```
      [,1]  
[1,] 0.07110086
```

Si può accettare l'ipotesi nulla che  $\mu = \mu_0$  a livello  $\alpha=0.05$

### 5.2.2 Test d'ipotesi $\mu=\mu_0$ , con $\Sigma$ non nota

“TRV2”

Con gli stessi dati, verifichiamo l'ipotesi del caso precedente, però con  $\Sigma$  non nota.

In questo caso il test TRV è:

$$\text{TRV2} = (n-p)/p (\underline{\mu} - \underline{\mu}_0)' \mathbf{S}^{-1} (\underline{\mu} - \underline{\mu}_0)$$

sotto  $MH_0$ :  $\text{TRV2} \sim F_{p,p-n}$

Sempre con l'aiuto in linea, possiamo vedere i parametri della funzione TRV2.

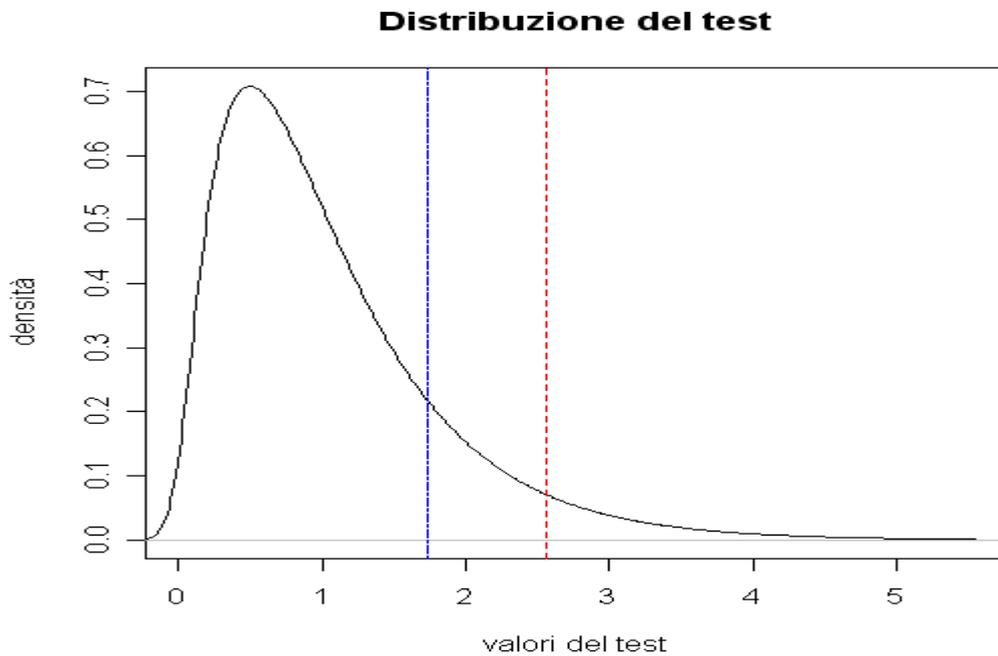
Funzione: TRV2

Parametri:

- **dati**: matrice di dati
- **$\mu_0$** : vettore medie  $\mu_0$
- **alpha**: ampiezza del test (impostato di default pari a 0.05)
- **grafico**: valore logico. Se è 'TRUE' disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

Applicazione:

```
TRV2(setosa,μ0,grafico="TRUE")
```



Pause. Press <Enter> to continue...

```
[1] "-----Risultati del test-----"
$test
```

```
      [,1]
[1,] 1.733327
```

```
$g11
[1] 4
```

```
$g12
[1] 46
```

```
$`p-value`
      [,1]
[1,] 0.1588863
```

Accettiamo l'ipotesi nulla.

### 5.2.3 Test d'ipotesi $\Sigma=\Sigma_0$ , con $\mu$ non nota

“TRV3”

Sempre con gli stessi dati, proviamo a verificare l'ipotesi che  $\Sigma=\Sigma_0$ , con  $\Sigma_0$  pari a:

$\Sigma_0$

	Sepal L.	Sepal W.	Petal L.	Petal W.
Sepal L.	0.12	0.10	0.02	0.01
Sepal W.	0.10	0.15	0.01	0.01
Petal L.	0.02	0.01	0.03	0.01
Petal W.	0.01	0.01	0.01	<b>0.028</b>

L'ipotesi risulta essere:

- $MH_0: \Sigma = \Sigma_0 \rightarrow$  con  $\mu$  non nota
- $MH_1: \Sigma \neq \Sigma_0$

Il test per la verifica d'ipotesi in questione è:

$$\text{TRV3} = np(a - \log(g) - 1)$$

dove:

$g$  è la media geometrica degli autovalori di  $\Sigma_0^{-1} S$

$a$  è la media aritmetica degli autovalori di  $\Sigma_0^{-1} S$

sotto  $MH_0$ :  $\text{TRV3} \xrightarrow{d} \chi_{p(p+1)/2}^2$  per  $n \rightarrow \infty$

Questo, è uno di quei casi in cui non si conosce la distribuzione esatta e quindi possiamo ricorrere ad uno dei cinque metodi già descritti nel capitolo 1.

La funzione “TRV3”, lascia all'utente la facoltà di sceglierne uno.

1. Analitico
2. Korin
3. Empirico1

4. Empirico2
5. Simulazione

Noi ci limitiamo a utilizzare, a titolo di esempio, la simulazione.

La funzione usata, dopo aver calcolato il test, stamperà a video il valore del test con i relativi gradi di libertà, il p-value stimato con il suo intervallo di confidenza e infine il numero di replicazioni eseguite per la simulazione.

Come nei casi precedenti tramite il comando `help(TRV3)`, possiamo vedere i parametri necessari.

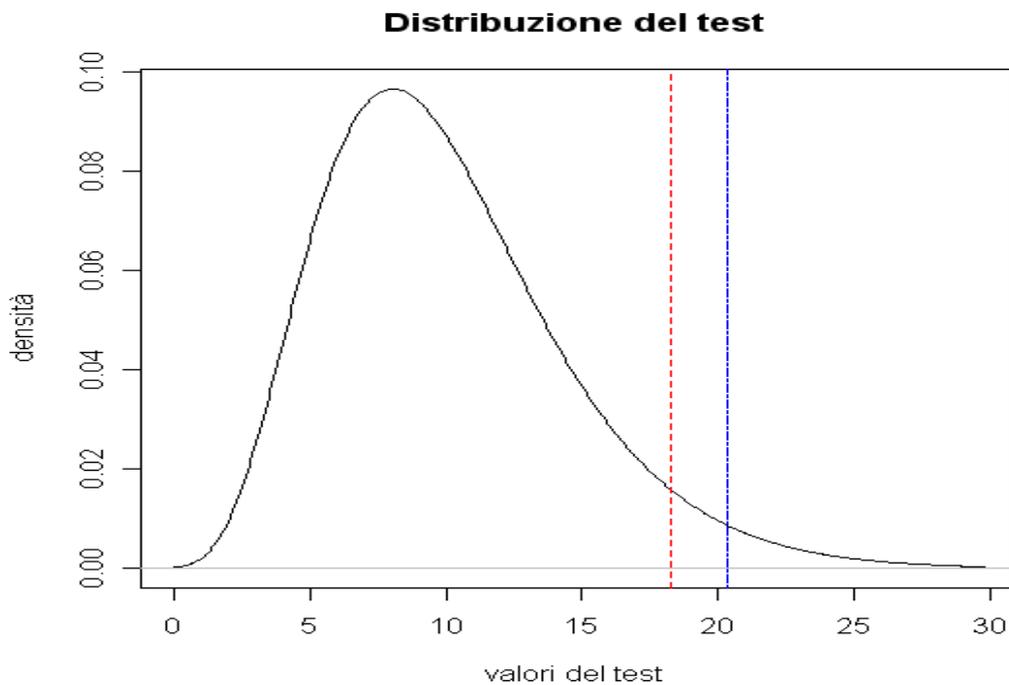
Funzione: **TRV3**

Parametri:

- **dati**: matrice di dati
- **sigma0**: matrice di varianza e covarianza dell'ipotesi  $MH_0$
- **a**: parametro di correzione1 (impostato di default pari a 0)
- **c**: parametro di correzione2 (impostato di default pari a 0)
- **alpha**: ampiezza del test (impostato di default pari a 0.05)
- **errore**: parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a  $5e-3$ )
- **grafico**: valore logico. Se è 'TRUE' disegna la distribuzione del test, due rette, una rossa che mi indica il percentile  $1-\alpha$  della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

Applicazione:

```
TRV3("simulazione",setosa,sigma0,grafico="TRUE")
```



Pause. Press <Enter> to continue...

```
[1] "-----Risultati del test-----"
```

```
$test
```

```
[1] 20.37629
```

```
$gl
```

```
[1] 10
```

```
$`p-value`
```

```
[1] 0.03892308
```

```
$`intervallo di confidenza del p-value`
```

```
[1] 0.03456350 0.04328265
```

```
$Nreplicazioni
```

```
[1] 13000
```

Sulla base del p-value stimato, lasciamo piena libertà all'utente nell'interpretazione del risultato.

## 5.2.4 Test d'ipotesi $\Sigma_1 = \dots = \Sigma_g$ (test di Box)

“TRV9”

Consideriamo le specie setosa e versicolor, si vuole verificare l'ipotesi:

- $MH_0: \Sigma_{\text{setosa}} = \Sigma_{\text{versicolor}}$
- $MH_1: \Sigma_{\text{setosa}} \neq \Sigma_{\text{versicolor}}$

La verifica dell'ipotesi in questione si realizza ricorrendo al test:

$$\text{TRV9} = \sum_{i=1}^g n_i \log |S_i^{-1}S| = -n \log |S| - \sum_{i=1}^g n_i \log |S_i|$$

sotto  $MH_0$ :  $\text{TRV9} \xrightarrow{d} \chi_{p(p+1)(g-1)/2}^2$  per  $n \rightarrow \infty$

Poiché non si conosce la distribuzione esatta, tramite la funzione “TRV9” lasciamo all'utente la facoltà di scegliere tra le 4 opzioni possibili.

1. Analitico
2. Empirico1
3. Empirico2
4. Simulazione

Utilizziamo l'opzione `Empirico2`, che dopo aver calcolato il test, stamperà a video il valore del test con i relativi gradi di libertà, il p-value stimato e infine i parametri di correzione  $a_2$  e  $c$ .

## Funzione: TRV9

### Parametri:

- **dati**: matrice di dati ottenuta mettendo una sotto l'altra le g matrici
- **num**: numerosità delle g matrici di dati
- **a**: parametro di correzione (impostato di default pari a 0)
- **c**: parametro di correzione (impostato di default pari a 0)
- **alpha**: ampiezza del test (impostato di default pari a 0.05)
- **errore**: parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a 5e-3)
- **grafico**: valore logico. Se è 'TRUE' disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Applicazione:

```
dati<-rbind(setosa,versicolor)
num<-c(50,50)
```

```
TRV9("empirico2",dati,num)
```

```
[1] "-*-*-*-*-*-*-*-*-*-*Risultati del test*-*-*-*-*-*-*-*-*-*-*-*"
$test
[1] 67.29366
```

```
$g1
[1] 10
```

```
$`p-value`
[1] 1.472522e-10
```

```
$`parametri di correzione a e c`
[1] 1.05880121 -0.04905065
```

```
$Nreplicazioni
[1] 13000
```

Come si può vedere, viene rifiutata l'ipotesi  $MH_0$ .

Adesso, supponiamo di voler verificare l'ipotesi:

- $MH_0: \Sigma_{setosa} = \Sigma_{virginica}$
- $MH_1: \Sigma_{setosa} \neq \Sigma_{virginica}$

La verifica dell'ipotesi in questione, come previsto, si realizza ricorrendo allo stesso test precedente.

In questo caso, per la verifica dell'ipotesi, conviene scegliere la stessa opzione di prima (`Empirico2`), poiché non cambiando la numerosità e le variabili dei campioni d'interesse, possiamo utilizzare gli stessi parametri di correzione, evitando così l'attesa per il calcolo della loro stima.

Applicazione:

```
dati<-rbind(setosa,virginica)
TRV9("empirico2",dati,num, 1.058,-0.049)
[1] "-*-*-*-*-*-*-*-*-*-*Risultati del test*-*-*-*-*-*-*-*-*-*-*"
$test
[1] 110.2054

$gl
[1] 10

$`p-value`
[1] 0

$`parametri di correzione a e c`
[1] 1.058 -0.049

$Nreplicazioni
[1] 0
```

Anche in questo caso rifiutiamo l'ipotesi  $MH_0$ .

Come verifica, proviamo ad eseguire la funzione senza inserire i parametri  $a_2$  e c.

```
TRV9("empirico2",dati,num)
[1] "*****-Risultati del test*****"
$test
[1] 109.9575

$gl
[1] 10

$p-value`
[1] 0

$`parametri di correzione a e c`
[1] 1.06006334 -0.08232029

$Nreplicazioni
[1] 13000
```

Come si può vedere dai risultati ottenuti, vengono confermate le considerazioni fatte prima.

## 5.3 Applicazione della libreria UIT sul data-set “iris3”

### 5.3.1 Test d’ipotesi $\mu=\mu_0$ , con $\Sigma$ nota

“UIT1”

A titolo di esempio, dato un vettore di medie  $\mu_0=(6.5, 3, 5.5, 2.15)$ , vogliamo stabilire se sono uguali a quelle delle variabili della popolazione virginica.

L’ipotesi risulta essere:

- ${}_M H_0: \mu = \mu_0 \quad \rightarrow$  con  $\Sigma$  nota
- ${}_M H_1: \mu \neq \mu_0$

Il test globale coincide in questo caso con il TRV:

$$\text{UIT1} = n (\mu - \mu_0)' \Sigma^{-1} (\mu - \mu_0)$$

sotto  ${}_M H_0: \text{UIT1} \sim \chi_p^2$

Per effettuare tale test ci siamo costruiti una funzione apposita, “UIT1”.

La funzione dopo aver calcolato il test e averlo confrontato con un adeguata distribuzione chi-quadrato, stamperà a video:

- il grafico della densità del test UIT1, due rette, una rossa che mi indica il percentile  $1-\alpha$  stimato della distribuzione del test globale e una blu che indica il valore del test globale calcolato sui dati originali
- il valore del test UIT globale.
- il p-value del test
- il valore critico della distribuzione del test, relativo all’ $\alpha$  usato
- il valore  $\alpha$  usato
- una frase che indica se l’ipotesi globale viene accettata o rifiutata

Funzione: **UIT1**

Parametri:

- **dati:** matrice di dati
- **$\mu_0$ :** vettore medie  $\mu_0$
- **sigma:** matrice di varianza e covarianza
- **alpha:** ampiezza del test (impostato di default pari a 0.05)
- **grafico:** valore logico. Se è 'TRUE' disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

Applicazione:

```
dati<-virginica
```

```
apply(virginica,2,mean)
```

```
Sepal L. Sepal W. Petal L. Petal W.  
 6.588    2.974    5.552    2.026
```

```
 $\mu_0$ <-c(6.5,3,5.5,2.15) → vettore ipotesi  $M_0$ 
```

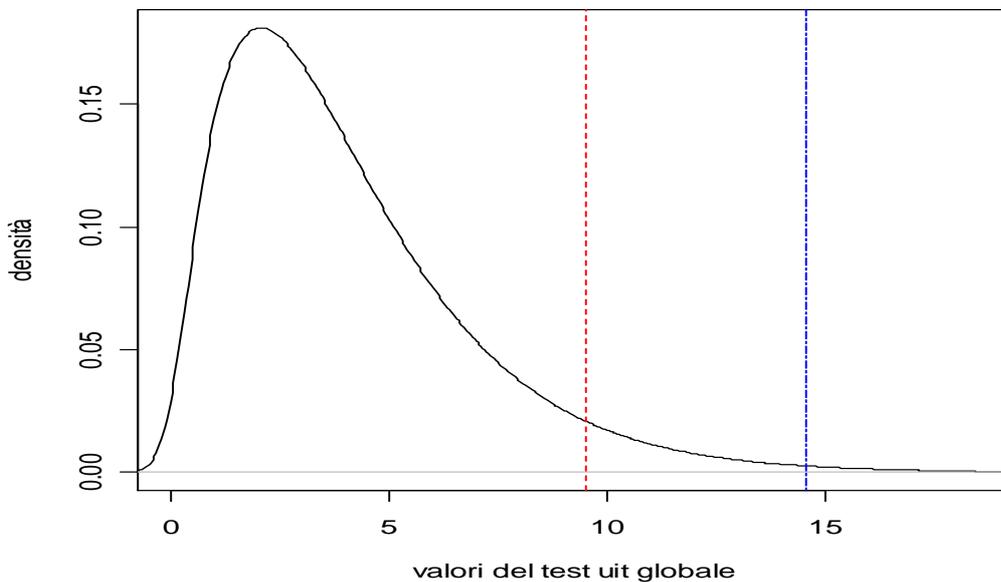
```
sigma<-var(virginica)
```

```
sigma
```

```
      Sepal L.   Sepal W.   Petal L.   Petal W.  
Sepal L. 0.40434286 0.09376327 0.30328980 0.04909388  
Sepal W. 0.09376327 0.10400408 0.07137959 0.04762857  
Petal L. 0.30328980 0.07137959 0.30458776 0.04882449  
Petal W. 0.04909388 0.04762857 0.04882449 0.07543265
```

```
UIT1(virginica,μ0,sigma,grafico="TRUE")
```

**Distribuzione del test UIT globale**



Pause. Press <Enter> to continue...

```
[1] "-----Risultati del test-----"
```

```
$`Valore del test UIT globale`
```

```
  [,1]
```

```
[1,] 14.54691
```

```
$`p-value`
```

```
  [,1]
```

```
[1,] 0.005739381
```

```
$Ipotesi
```

```
[1] "Rifiutiamo l'ipotesi globale"
```

```
[2] "Si può verificare se una qualche <ipotesi componente>"
```

```
[3] "abbia portato al rifiuto con:"
```

```
[4] "UIT1.a"
```

Se, come in questo caso l'ipotesi  $MH_0$  è rifiutata, allora, come viene riportato nell'output, si utilizza la funzione **UIT1.a**

Questa funzione calcola il test per la generica componente nella forma:

$$UIT1.a = LR = n\bar{a}' (\bar{\mu} - \mu_0)' (\bar{\mu} - \mu_0) \bar{a} / \bar{a}' \Sigma^{-1} \bar{a}$$

sotto  $MH_0$ :  $UIT1.a \sim \chi_p^2$

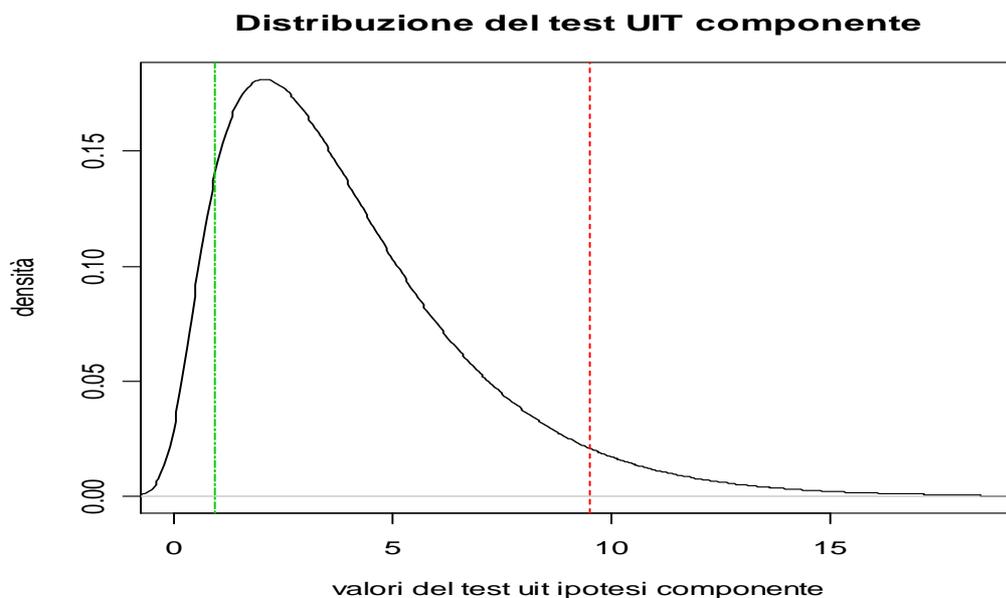
## Funzione: UIT1.a

### Parametri:

- **a**: vettore p-dimensionale che specifica un'ipotesi componente
- **grafico**: valore logico. Se è 'TRUE' disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

### Applicazione:

`a<-c(1,0,0,0)` → consideriamo solo la prima variabile  
`UIT1.a(a,grafico="TRUE")`



Pause. Press <Enter> to continue...

```
[1] "----*---*---*---*---Risultati del test*---*---*---*---*---*---"
```

```
$`Vettore a:`
```

```
[1] 1 0 0 0
```

```
$`Valore del test UIT ipotesi componente`
```

```
 [,1]
```

```
[1,] 0.9576032
```

```
$`p-value`
```

```
 [,1]
```

```
[1,] 0.9161551
```

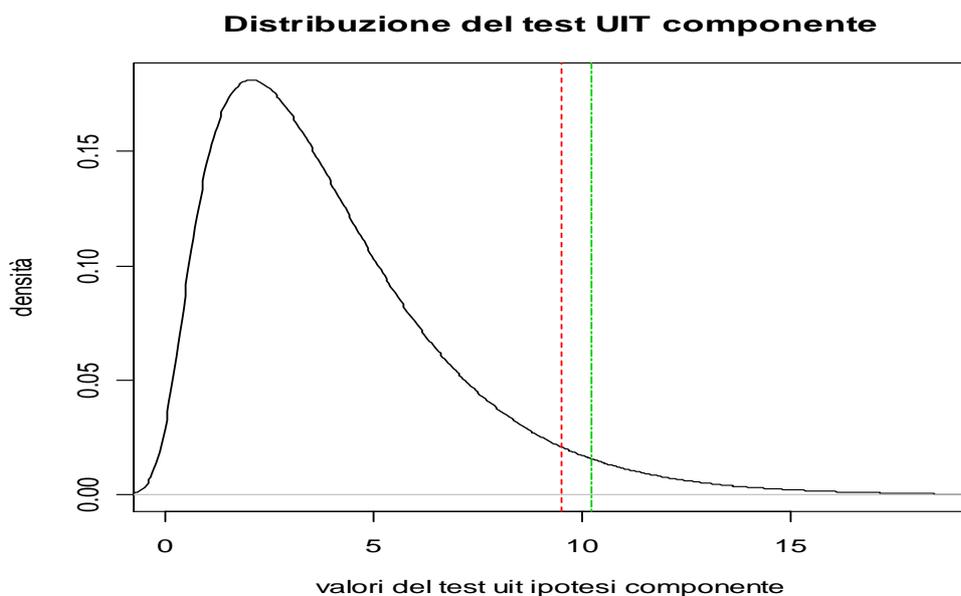
```
$Ipotesi
```

```
[1] "Accettiamo l'ipotesi componente"
```

Accettiamo l'ipotesi componente, cioè l'ipotesi che la media della variabile Sepal L. è uguale a 6.5 (primo elemento del vettore  $\mu_0$ ). Questo lo si vede anche nel grafico, nel quale la retta verde che indica il valore del test dell'ipotesi componente sui dati originali (virginica) è a sinistra, quindi inferiore al 0.95 percentile della distribuzione del test.

La stessa cosa viene eseguita anche per le altre componenti e si vedrà che un'ipotesi componente che ha portato al rifiuto il test globale è la media della variabile Petal W che è diversa da 2.2.

```
a<-c(0,0,0,1)
UIT1.a(a,grafico="TRUE")
```



```
Pause. Press <Enter> to continue...
[1] "-----Risultati del test-----"
$`Vettore a:`
[1] 0 0 0 1

$`Valore del test U.I.T. ipotesi componente`
      [,1]
[1,] 10.19187

$`p-value`
      [,1]
[1,] 0.03731671

$Ipotesi
[1] "Rifiutiamo l'ipotesi componente"
```

### 5.3.2 Test d'ipotesi $\Sigma = \Sigma_0$ , con $\mu$ non nota

“UIT3”

Consideriamo la popolazione setosa, vogliamo verificare l'ipotesi:

- $MH_0: \Sigma = \Sigma_0 \rightarrow$  con  $\mu$  non nota
- $MH_1: \Sigma \neq \Sigma_0$

dove  $\Sigma_0$  è:

	Sepal L.	Sepal W.	Petal L.	Petal W.
Sepal L.	0.12	0.10	0.02	0.01
Sepal W.	0.10	0.15	0.01	0.01
Petal L.	0.02	0.01	0.03	0.01
Petal W.	0.01	0.01	0.01	<b>0.02</b>

Il test globale (UIT3) in questo caso è la coppia di autovalori:

$$\lambda_1(\Sigma^{-1}_0 \mathbf{S}) \quad \lambda_p(\Sigma^{-1}_0 \mathbf{S})$$

dove rifiutiamo  $MH_0$  se  $\lambda_1(\Sigma^{-1}_0 \mathbf{S}) > c_{1\alpha}$  o  $\lambda_p(\Sigma^{-1}_0 \mathbf{S}) < c_{2\alpha}$

$$\Pr\{\lambda_1(\Sigma^{-1}_0 \mathbf{S}) > c_{1\alpha}, \lambda_p(\Sigma^{-1}_0 \mathbf{S}) < c_{2\alpha} \mid MH_0\} = \alpha$$

In questo caso, non conoscendo la distribuzione del test, si ricorre alla funzione “UIT3”, che ci permette di calcolare i percentili d'interesse della distribuzione del test tramite la simulazione.

La funzione dopo aver calcolato il test stamperà a video:

- il grafico della densità stimata tramite simulazione del test UIT3, quattro rette, due rosse che indicano i percentili  $\alpha/2$  e  $1-\alpha/2$  stimati dalla distribuzione del test globale e due blu che indicano i valori del test globale calcolato sui dati originali
- il valori del test UIT globale

- il valori critici della distribuzione del test, relativo all' $\alpha$  usato
- il valore  $\alpha$  usato
- una frase che indica se l'ipotesi globale viene accettata o rifiutata

Funzione: **UIT3**

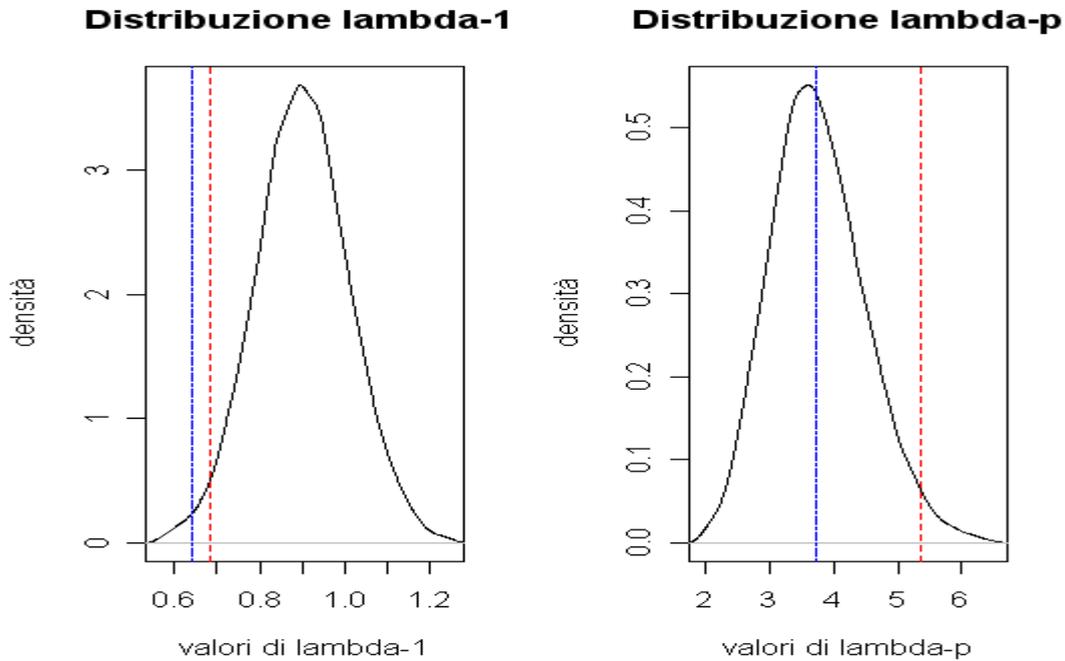
Parametri:

- **dati**: matrice di dati
- **sigma0**: matrice di varianza e covarianza dell'ipotesi  $MH_0$
- **alpha**: ampiezza del test (impostato di default pari a 0.05)
- **errore**: parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a  $5e-3$ )
- **grafico**: valore logico. Se è 'TRUE' disegna la distribuzione del test, quattro rette, due rosse che indicano i percentili  $\alpha/2$  e  $1-\alpha/2$  stimati dalla distribuzione del test globale e due blu che indicano i valori del test globale calcolato sui dati originali.

Applicazione:

```
dati<-setosa
```

```
UIT3(setosa,sigma0,errore=0.004,grafico="TRUE")
```



Pause. Press <Enter> to continue...

```
[1] "-----Risultati del test-----"
```

```
$`Valore del test U.I.T. globale`
```

```
[1] 0.6434233 3.7271792
```

```
$`Valori critici della distribuzione del Test`
```

```
      x      x  
0.6839678 5.3439971
```

```
$alpha.effettivo
```

```
[1] 0.05
```

```
$Ipotesi
```

```
[1] "Rifiutiamo l'ipotesi globale"
```

```
[2] "Si può verificare se una qualche <ipotesi componente>"
```

```
[3] "abbia portato al rifiuto con:"
```

```
[4] "UIT3.a"
```

```
$Nreplicazioni
```

```
[1] 20000
```

Il test in questione è più difficile da gestire e diversamente da quanto avviene negli altri casi, calcoliamo l'alpha.effettivo, per tener conto della eventualità (abbastanza rara) che  $\lambda_1$  e  $\lambda_p$ , escano contemporaneamente dalle zone di accettazione. Si tratta di una scelta obbligata, dovuta alla struttura del test UIT3. In questo esempio, inoltre abbiamo voluto aumentare il numero delle replicazioni a  $N=20000$ , ponendo il parametro errore=0.004

Se, come in questo caso l'ipotesi  $M_{H_0}$  è rifiutata, allora, come viene riportato nell'output, si utilizza **UIT3.a**.

Questa funzione calcola il test per la generica componente nella forma:

$$\mathbf{UIT3.a} = \mathbf{LR} = \mathbf{a' S a / a' \Sigma_0 a}$$

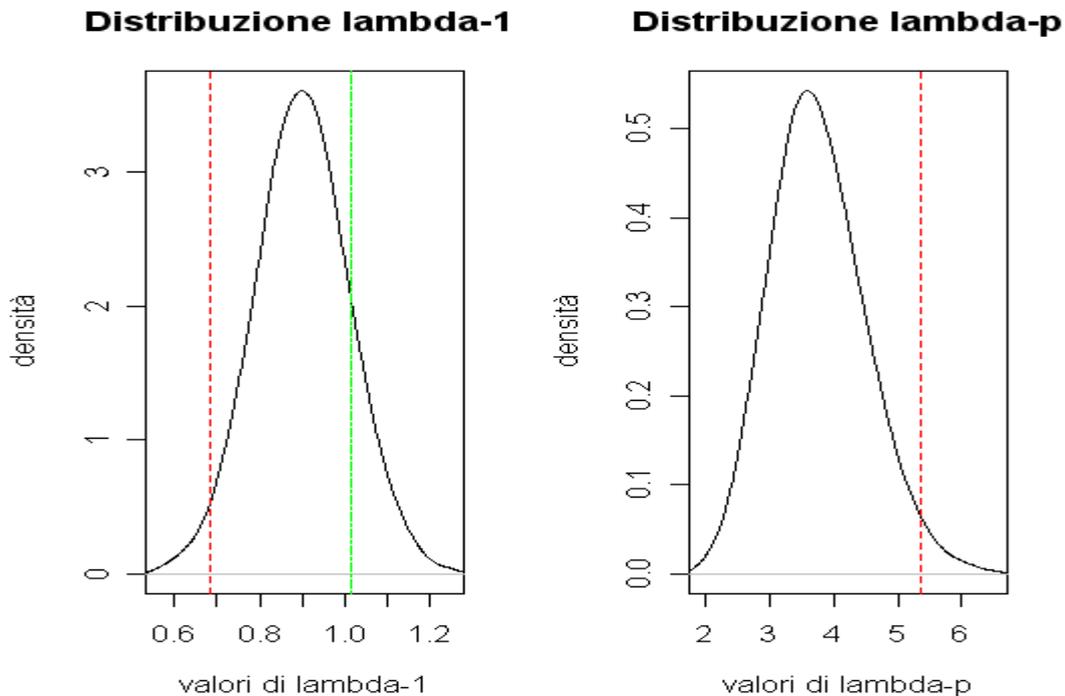
Funzione: **UIT3.a**

Parametri:

- **a**: vettore p-dimensionale che specifica un'ipotesi componente
- **grafico**: valore logico. Se è 'TRUE' disegna la distribuzione del test, quattro rette, due rosse che indicano i percentili  $\alpha/2$  e  $1-\alpha/2$  stimati dalla distribuzione del test globale e due verdi che indicano i valori del test globale calcolato sui dati originali.

Applicazione:

`a<-c(1,0,0,0)` → consideriamo solo la prima variabile  
`UIT3.a(a,grafico="TRUE")`



Pause. Press <Enter> to continue...

```
[1] "Risultati della simulazione"
[1] "*****"
$`Vettore a:`
[1] 1 0 0 0

$`Valore del test U.I.T. componente`
[1] 1.0147

$`Valori critici della distribuzione del Test`
      x      x
0.6839678 5.3439971

$alpha
[1] 0.05

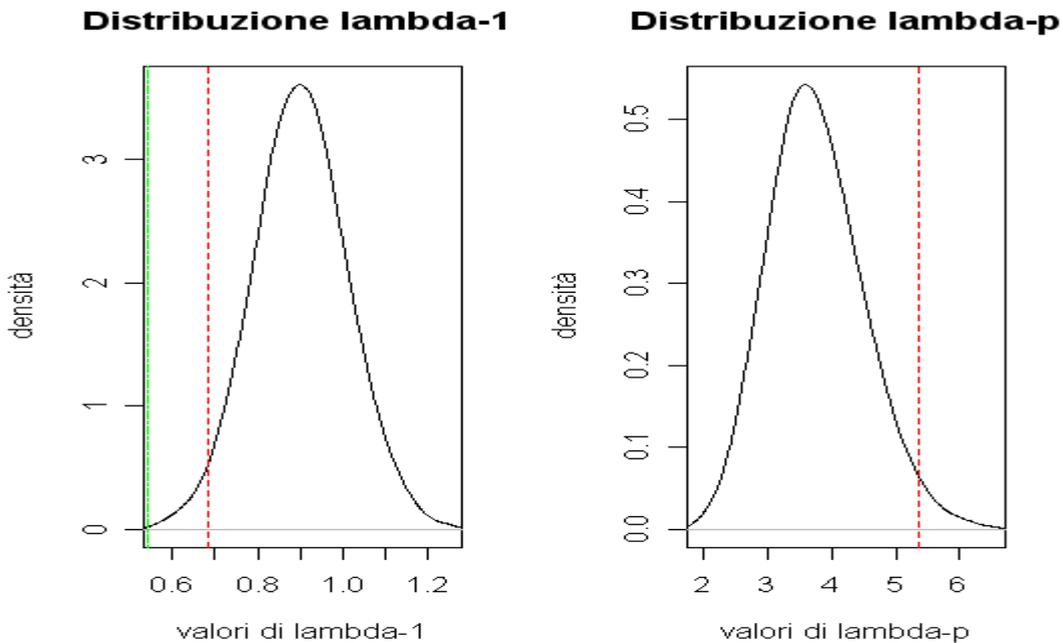
$Ipotesi
[1] "Accettiamo l'ipotesi componente"
Accettiamo l'ipotesi componente.
```

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare dal quantile 0.001 fino al quantile di livello 0.999. Può capitare,

come in questo caso, che il valore del test sia maggiore o minore dei quantili considerati e di conseguenza non essere disegnato nel grafico.

Proviamo a considerare un'altra componente:

```
a<-c(0,0,0,1)
UIT3.a(a,grafico="TRUE")
```



```
Pause. Press <Enter> to continue...
[1] "Risultati della simulazione"
[1] "-----"
$`Vettore a:`
[1] 0 0 0 1

$`Valore del test U.I.T. componente`
[1] 0.5442

$`Valori critici della distribuzione del Test`
      x      x
0.6839678 5.3439971

$alpha
[1] 0.05

$Ipotesi
[1] "Rifiutiamo l'ipotesi componente"
```

La stessa cosa può essere eseguita anche per le altre componenti.

### 5.3.3 Test d'ipotesi $\mu_1=\dots=\mu_g$ , con $\Sigma_1=\dots=\Sigma_g$

“UIT8”

Consideriamo le popolazioni versicolor e virginica, vogliamo verificare l'ipotesi:

- $MH_0: \mu_1=\dots=\mu_g \rightarrow$  con  $\Sigma_1=\dots=\Sigma_g$
- $MH_1: \mu_1 \neq \dots \neq \mu_g$

Il test globale è:

$$UIT8 = \sum_{i=1}^g \left\{ \frac{n_i(\mu_i - \mu)(\mu_i - \mu)'}{n_i \Sigma_i^2} \right\}$$

Anche in questo caso, non conoscendo la distribuzione del test, si ricorre alla stima dei percentili d'interesse della distribuzione del test tramite la funzione “UIT8”.

La funzione dopo aver calcolato il test stamperà a video:

- il grafico della densità del test UIT8, due rette, una rossa che mi indica il percentile  $1-\alpha$  stimato della distribuzione del test globale e una blu che indica il valore del test globale calcolato sui dati originali.
- il valore del test UIT globale
- il p-value del test
- il valore critico della distribuzione del test, relativo all' $\alpha$  usato
- il valore  $\alpha$  usato
- una frase che indica se l'ipotesi globale viene accettata o rifiutata

Funzione: **UIT8**

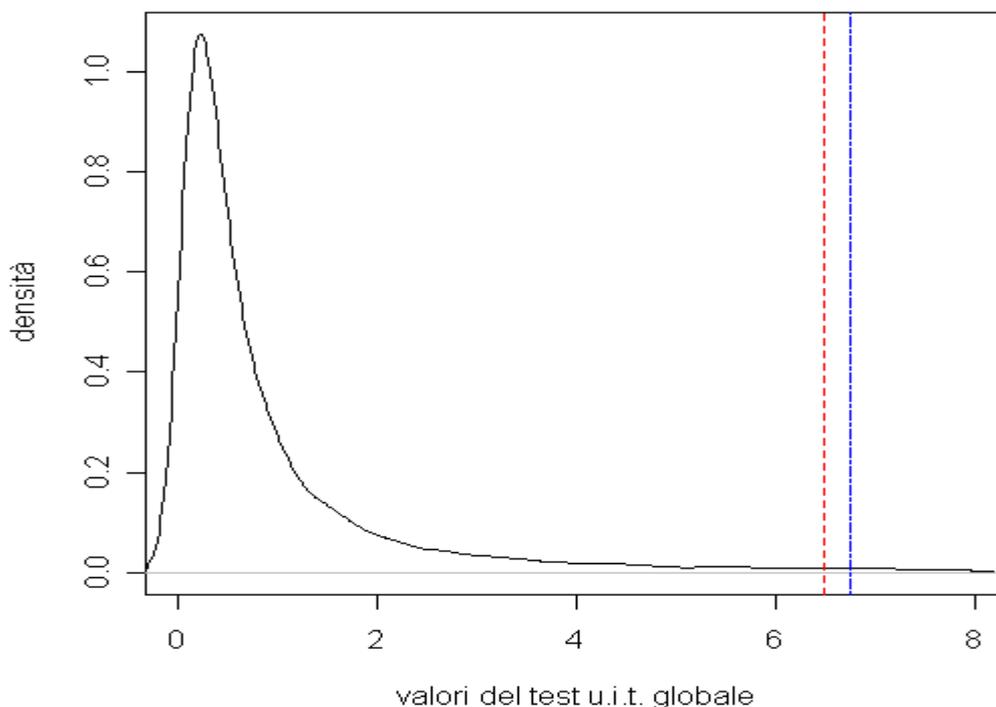
Parametri:

- **dati:** matrice di dati mettendo una sotto l'altra le g matrici
- **num:** numerosità delle g matrici di dati
- **alpha:** ampiezza del test (impostato di default pari a 0.05)
- **errore:** parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a  $5e-3$ )
- **grafico:** valore logico. Se è 'TRUE' disegna la distribuzione del test, due rette, una rossa che mi indica il percentile  $1-\alpha$  della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

Applicazione:

```
dati<-rbind(versic, virginica)  
num<-c(50,50)  
UIT8(dati,num,errore=0.003,grafico="TRUE")
```

**Distribuzione del test U.I.T. globale**



```

Pause. Press <Enter> to continue...
[1] "Risultati della simulazione"
[1] "*****"
$`Valore del test UIT globale`
[1] 6.746008

$`Valore critico della distribuzione del Test`
  95%
6.482742

$alpha
[1] 0.05

$Ipotesi
[1] "Rifiutiamo l'ipotesi globale"
[2] "Si può verificare se una qualche <ipotesi componente>"
[3] "abbia portato al rifiuto con:"
[4] "UIT8.a"

$Nreplicazioni
[1] 35000

```

L'ipotesi  $M_0$  è rifiutata, allora, come viene riportato nell'output, si utilizza **UIT8.a**.

Questa funzione calcola il test per la generica componente nella forma:

$$UIT8.a = \sum_{i=1}^g \left\{ \frac{n_i [a' (\mu_i - \mu)^2]}{n_i a' \Sigma_i^2 a} \right\}$$

Funzione: **UIT8.a**

Parametri:

- **a**: vettore p-dimensionale che specifica un'ipotesi componente
- **grafico**: valore logico. Se è 'TRUE' disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

## Applicazione:

`a<-c(0,0,1,0)` → consideriamo solo la terza variabile

**`UIT8.a(a,grafico="TRUE")`**

Pause. Press <Enter> to continue...

[1] "Risultati della simulazione"

[1] "\*\*\*\*\*"

\$`Vettore a:`

[1] 0 0 1 0

\$`Valore del test UIT componente`

[,1]

[1,] 5.32863

\$`Valore critico della distribuzione del Test`

95%

6.482742

\$alpha

[1] 0.05

\$Ipotesi

[1] "Accettiamo l'ipotesi componente"

La stessa cosa può essere eseguita anche per le altre componenti.

## Conclusioni

Come si è visto, le simulazioni in statistica possono essere utilmente impiegate per stimare delle distribuzioni campionarie.

L'errore standard nella stima di tali distribuzioni è proporzionale in generale a  $\frac{1}{\sqrt{N}}$  dove N è il numero delle replicazioni.

È evidente che le simulazioni sono dunque uno strumento importante per l'inferenza statistica, dal momento che per poter fare inferenza è fondamentale la conoscenza delle distribuzioni campionarie di test e stimatori.

I **vantaggi** sono:

- grande *flessibilità* di uso e applicabilità ad un gran numero di situazioni
- grande *semplicità*: infatti per applicazioni elementari è sufficiente saper generare particolari campioni di numeri-pseudo casuali e saper calcolare lo stimatore o il test di interesse su ciascun campione
- *rapidità* di applicazione
- facilità di *comunicazione*: essendo ormai molto comune l'impiego di tecniche di simulazione in lavori scientifici, raramente è necessario spiegare in dettaglio il particolare schema utilizzato

Gli **svantaggi**, a mio avviso, sono fondamentalmente:

- si tratta di un *approccio empirico*; una cosa è, ad esempio, far vedere tramite simulazione che facendo crescere il numero di replicazioni N l'alpha stimato si avvicina al valore nominale  $\alpha$ , e che non se ne discosta in modo significativo, un'altra è invece dimostrare

analiticamente che il livello di significatività tende ad  $\alpha$  al divergere della numerosità campionaria

- i risultati ottenuti, sono riferibili solo alle *situazioni effettivamente sperimentate*
- è poco “*elegante*” ricorrere ad una simulazione se è possibile ottenere un risultato analitico

Il lavoro svolto in questa tesi consisteva nella progettazione e realizzazione di due packages per risolvere specifici problemi di verifica d'ipotesi multidimensionali.

Credo che la cosa principale messa in luce da questa tesi è che la simulazione può facilitare molto le applicazioni inferenziali, tuttavia non può eliminare ogni difficoltà.

I metodi di simulazione presentati in questa tesi sembrano poter costituire uno strumento efficiente per alcuni problemi inferenziali, e gli esempi presentati nel CAPITOLO 5, pur con i loro limiti, sembrano complessivamente confermare tale impressione.

Per concludere, giacché i packages sono costruiti in modo modulare, possono essere ampliati nelle loro funzionalità con l'inserimento di nuove parti che realizzino ulteriori algoritmi, qualora se ne presenti la necessità.

# APPENDICE

## PACKAGE TRV

**Title:** Metodi di simulazione per i test TRV  
**Version:** 1.0  
**Depends:** MASS  
**Date:** 2007-05-02  
**Author:** Michele Battocchio <battocchiomichele@libero.it>, Giancarlo Diana <giancarlo.diana@unipd.it>  
**Maintainer:** Giancarlo Diana <giancarlo.diana@unipd.it>  
**Description:** Questo pacchetto contiene delle semplici funzioni per simulare alcuni Test Rapporto di Verosimiglianza  
**License:** GPL version 2

### FUNZIONI

TRV1.....	108
TRV2.....	109
TRV3.....	110
TRV4.....	112
TRV5.....	114
TRV6.....	115
TRV72.....	117
TRV7.....	119
TRV82.....	121
TRV8.....	122
TRV9.....	124
TRV10.....	126

---

## TRV1

Test di ipotesi  $\mu=\mu_0$ , con sigma nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu=\mu_0$ , con sigma nota

### Usage

```
TRV1(dati,  $\mu_0$ , sigma, alpha = 0.05, grafico = "False")
```

### Arguments

dati	matrice di dati
$\mu_0$	vettore medie $\mu_0$
sigma	matrice di varianza e covarianza
alpha	ampiezza del test (impostato di default pari a 0.05)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione TRV1 effettua il calcolo del test, gradi di libertà, p-value ed eventuale grafico della distribuzione del test. In questo particolare caso il test si distribuisce **ESATTAMENTE** come un Chi-Quadrato.

$H_0: \mu = \mu_0 \implies$  con S nota  
 $H_1: \mu \neq \mu_0$

$$TRV1 = n (\mu - \mu_0)' S^{-1} (\mu - \mu_0)$$

$$\text{sotto } H_0: TRV1 \sim \chi_p^2$$

### Value

test	valore del test
gl	gradi di libertà
p-value	p-value

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[TRV2](#), [TRV4](#), [TRV5](#), [TRV82](#), [TRV8](#), [TRV10](#)

## Examples

```
data(iris3)
setosa<-iris3[,1]
apply(setosa,2,mean)
μ0<-c(5,3.4,1.4,0.24)
sigma<-rbind(c(0.12,0.10,0.02,0.01),c(0.10,0.15,0.01,0.01),
             c(0.02,0.01,0.03,0.01),c(0.01,0.01,0.01,0.028))

TRV1(setosa,μ0,sigma,grafico="TRUE")
```

---

## TRV2

Test di ipotesi  $\mu=\mu_0$ , con sigma non nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu=\mu_0$ , con sigma non nota

### Usage

```
TRV2(dati, μ0, alpha = 0.05, grafico = "FALSE")
```

### Arguments

dati	matrice di dati
μ0	vettore medie μ0
alpha	ampiezza del test (impostato di default pari a 0.05)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-apha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione TRV2 effettua il calcolo del test, gradi di libertà, p-value ed eventuale grafico della distribuzione del test. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con g11 e g12 gradi di libertà.

H0:  $\mu = \mu_0$  ==>con sigma non nota  
H1:  $\mu \neq \mu_0$

$$TRV2 = (n-p)/p (\mu - \mu_0)'S^{-1} (\mu - \mu_0)$$

$$\text{sotto } H_0: TRV2 \sim F_{p,n-p}$$

## Value

test	valore del test
gl1	gradi di libertà 1
gl2	gradi di libertà 2
p-value	p-value

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[TRV1](#), [TRV4](#), [TRV5](#), [TRV82](#), [TRV8](#), [TRV10](#)

## Examples

```
data(iris3)
setosa<-iris3[,1]
apply(setosa,2,mean)
mu0<-c(5,3.4,1.4,0.24)

TRV2(setosa,mu0,grafico="TRUE")
```

---

## TRV3

Test di ipotesi  $\sigma=\sigma_0$ , con  $\mu$  non nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\sigma=\sigma_0$ , con  $\mu$  non nota.

### Usage

```
TRV3(scelta,dati,sigma0,a=0,c=0,alpha=0.05,errore=0.005,grafico="False")
```

### Arguments

scelta	inserire una delle possibili opzioni: ("analitico","korin","empirico1","empirico2","simulazione")
dati	matrice di dati
sigma0	matrice di varianza e covarianza sotto l'ipotesi nulla
a	parametro di correzione1 (impostato di default pari a 0)
c	parametro di correzione2 (impostato di default pari a 0)
alpha	ampiezza del test (impostato di default pari a 0.05)
errore	parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a 5e-3)

grafico valore logico. Se è `{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile  $1-\alpha$  della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

## Details

La funzione TRV3 effettua il calcolo del test, gradi di libertà, p-value... In questo particolare caso il test si distribuisce **ASINTOTICAMENTE** come un Chi-Quadrato.

H0:  $\sigma = \sigma_0$  ==> con  $\mu$  non nota

H1:  $\sigma \neq \sigma_0$

$$TRV3 = np (a - \log(g) - 1)$$

$$\text{sotto } H_0: TRV3 \sim \chi^2_{p(p+1)/2}$$

A seconda della scelta effettuata si hanno differenti modi di calcolare il test:

### ANALITICO

Si utilizza la correzione di Bartlett analitica.

### EMPIRICO1

Si utilizza come fattore di correzione la stima della media del test simulato in ipotesi nulla.

### EMPIRICO2

Si utilizzano come fattori di correzione la stima della media e della varianza del test simulato in ipotesi nulla.

### KORIN

Alternativa alla correzione di Bartlett analitica proposta da Korin(1968).

### SIMULAZIONE

Stima della distribuzione del test mediante simulazione.

## Value

- valore del test
- gradi di libertà
- p-value

Inoltre, a seconda della scelta effettuata si hanno in più i seguenti valori:

Empirico1:

- parametro di correzione a
- numero repliche

Empirico2:

- parametri di correzione a e c
- numero repliche

Simulazione:  
- intervallo di confidenza del p-value  
- numero replicazioni

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[TRV6](#), [TRV72](#), [TRV7](#), [TRV9](#)

## Examples

```
data(iris3)
setosa<-iris3[,1]
sigma<-var(setosa)
sigma
sigma0<-rbind(c(0.12,0.10,0.02,0.01),c(0.10,0.15,0.01,0.01),
              c(0.02,0.01,0.03,0.01),c(0.01,0.01,0.01,0.028))
sigma0

TRV3("simulazione",setosa,sigma0,errore=0.025,grafico="TRUE")

# Aumentiamo il numero delle replicazioni

TRV3("simulazione",setosa,sigma0,grafico="TRUE")
```

---

## TRV4

Test di ipotesi  $R\mu=r$ , con sigma nota

---

## Description

Questa funzione permette il calcolo del test di ipotesi  $R\mu=r$ , con sigma nota.

## Usage

```
TRV4(dati, sigma, R, r, alpha = 0.05, grafico = "FALSE")
```

## Arguments

dati	matrice di dati
sigma	matrice di varianza e covarianza
R	vincoli lineari
r	vettore contenente i valori da verificare con i vincoli lineari

alpha            ampiezza del test (impostato di default pari a 0.05)  
grafico            valore logico. Se è `\code{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$  della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

## Details

La funzione TRV4 effettua il calcolo del test, gradi di libertà, p-value ed eventuale grafico della distribuzione del test. In questo particolare caso il test si distribuisce **ESATTAMENTE** come un Chi-Quadrato.

H0:  $R\mu = r \implies$  con  $\sigma$  nota

H1:  $R\mu \neq r$

$$TRV4 = n (R\mu - r)' (R\sigma R')^{-1} (R\mu - r)$$

$$\text{sotto } H0: TRV4 \sim \chi_r^2$$

## Value

test            valore del test  
gl                gradi di libertà  
p-value        p-value

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[TRV5](#)

## Examples

```
data(iris3)
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
dati<-cbind(versicolor,virginica)
sigma<-var(dati)
R<-rbind(c(1,0,-1,0),c(0,1,0,-1))
r<-rbind(c(-0.4),c(0))

TRV4(dati,sigma,R,r,grafico="TRUE")
```

---

## TRV5

Test di ipotesi  $R\mu=r$ , con sigma non nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $R\mu=r$ , con sigma non nota.

### Usage

```
TRV5(dati, R, r, alpha = 0.05, grafico = "FALSE")
```

### Arguments

`dati` matrice di dati  
`R` vincoli lineari  
`r` vettore contenente i valori da verificare con i vincoli lineari  
`alpha` ampiezza del test (impostato di default pari a 0.05)  
`grafico` valore logico. Se è `\code{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$  della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione TRV5 effettua il calcolo del test, gradi di libertà, p-value ed eventuale grafico della distribuzione del test. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con  $g1$  e  $g2$  gradi di libertà.

$H_0: R\mu = r \implies$  con sigma non nota

$H_1: R\mu \neq r$

$$TRV5 = (n-1) (R\mu - r)'(R^*S^*R')^{-1} (R\mu - r)$$

$$\text{sotto } H_0: TRV5 \sim F_{r, n-r}$$

### Value

`test` valore del test  
`g1` gradi di libertà 1  
`g2` gradi di libertà 2  
`p-value` p-value

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[TRV4](#)

## Examples

```
data(iris3)
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
dati<-cbind(versicolor,virginica)
R<-rbind(c(1,0,-1,0),c(0,1,0,-1))
r<-rbind(c(-0.4),c(0))
```

```
TRV5(dati,R,r,grafico="TRUE")
```

---

## TRV6

Test di ipotesi  $\sigma=k*\sigma_0$ , con  $\mu$  e  $k$  non note

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\sigma=k*\sigma_0$ , con  $\mu$  e  $k$  non note.

### Usage

```
TRV6(scelta,dati,sigma0,l=0,a=0,c=0,alpha=0.05,errore=0.005,grafico="False")
```

### Arguments

scelta	inserire una delle possibili opzioni: ("empirico2","simulazione")
dati	matrice di dati
sigma0	matrice di varianza e covarianza sotto l'ipotesi nulla
l	parametro che indica che gli ultimi "variabili-l" autovalori della matrice di varianza e covarianza sono uguali (impostato di default pari a 0)}
a	parametro di correzione1 (impostato di default pari a 0)
c	parametro di correzione2 (impostato di default pari a 0)
alpha	ampiezza del test (impostato di default pari a 0.05)
errore	parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a $5e-3$ )
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile $1-\alpha$ della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione TRV6 effettua il calcolo del test, gradi di libertà, p-value....In questo particolare caso il test si distribuisce **ASINTOTICAMENTE** come un Chi-Quadrato.

H0:  $\sigma = k*\sigma_0$  ==>con  $\mu$  e  $k$  non note  
H1:  $\sigma \neq k*\sigma_0$

$$TRV6 = np \log(a/g)$$

$$\text{sotto } H_0: TRV6 \sim \chi^2_{(p-1-l)(p-l+2)/2}$$

A seconda della scelta effettuata si hanno differenti modi di calcolare il test:

#### **EMPIRICO2**

Si usano come fattori di correzione la stima della media e della varianza del test simulato in ipotesi nulla.

#### **SIMULAZIONE**

Stima della distribuzione del test mediante simulazione.

### **Value**

- valore del test
- gradi di libertà
- p-value

Inoltre, a seconda della scelta effettuata si hanno in più i seguenti valori:

Empirico2:

- parametri di correzione a e c
- numero replicazioni

Simulazione:

- intervallo di confidenza del p-value
- numero replicazioni

### **Note**

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

### **See Also**

[TRV3](#), [TRV72](#), [TRV7](#), [TRV9](#)

### **Examples**

```
data(iris)
setosa<-iris3[,1:2,1]
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
dati<-cbind(setosa,versicolor,virginica)
sigma<-
rbind(c(0.12,0.1,0,0,0.1,0.03),c(0.1,0.14,0,0,0,0),c(0,0,0.26,0.1,0,0),
      c(0,0,0.1,0.1,0,0),c(0.1,0,0,0,0.4,0.1),c(0.03,0,0,0,0.1,0.1))

TRV6("simulazione",dati,sigma,grafico="TRUE")
```

---

## TRV72

Test di ipotesi  $\sigma^2=0$ , con  $\mu$  non nota

---

### Description

Questa funzione permette di partizionare il data-set in due con dimensioni rispettivamente  $p_1$  e  $p_2$ , e calcolare il test di ipotesi  $\sigma^2=0$ , con  $\mu$  non nota.

### Usage

```
TRV72(scelta,dati,p1,p2,a=0,c=0,alpha=0.05,errore=0.005,grafico="False")
```

### Arguments

scelta	inserire una delle possibili opzioni: ("vera","analitico","empirico1","empirico2","simulazione")
dati	matrice di dati
p1	numero variabili partizione 1
p2	numero variabili partizione 2
a	parametro di correzione1 (impostato di default pari a 0)
c	parametro di correzione2 (impostato di default pari a 0)
alpha	ampiezza del test (impostato di default pari a 0.05)
errore	parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a 5e-3)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione TRV72 effettua il calcolo del test, gradi di libertà, p-value.....In questo particolare caso il test si distribuisce **ASINTOTICAMENTE** come un Chi-Quadrato.

$H_0: \sigma^2 = 0 \implies$  con  $\mu$  non nota

$H_1: \sigma^2 \neq 0$

$$TRV72 = -n \cdot \log |(I-1)|$$

$$\text{sotto } H_0: TRV72 \sim \chi^2_{p_1 \cdot p_2}$$

A seconda della scelta effettuata si hanno differenti modi di calcolare il test:

#### VERA

Nel caso  $p_1=1,2$  o  $p_2=1,2$  il test calcolato si distribuisce **ESATTAMENTE** come una F con  $g_1$  e  $g_2$  gradi di libertà.

#### ANALITICO

Si utilizza la correzione di Bartlett analitica.

### **EMPIRICO1**

Si utilizza come fattore di correzione la stima della media del test simulato in ipotesi nulla.

### **EMPIRICO2**

Si utilizzano come fattori di correzione la stima della media e della varianza del test simulato in ipotesi nulla.

### **SIMULAZIONE**

Stima della distribuzione del test mediante simulazione.

## **Value**

- valore del test
- gradi di libertà
- p-value

Inoltre, a seconda della scelta effettuata si hanno in più i seguenti valori:

Empirico1:

- parametro di correzione a
- numero replicazioni

Empirico2:

- parametri di correzione a e c
- numero replicazioni

Simulazione:

- intervallo di confidenza del p-value
- numero replicazioni

## **Note**

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## **See Also**

[TRV3](#), [TRV7](#), [TRV9](#)

## **Examples**

```
data(iris3)
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
apply(versicolor,2,var)
apply(virginica,2,var)
dati<-cbind(versicolor,virginica)

TRV72("vera",dati,2,2,grafico="TRUE")
```

---

## TRV7 Test di indipendenza totale

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\sigma=I$ , con  $\mu$  non nota.

### Usage

```
TRV7(scelta,dati,a=0,c=0,alpha=0.05,errore=0.005,grafico="False")
```

### Arguments

scelta      inserire una delle possibili opzioni:  
            ("analitico","empirico1","empirico2","simulazione")

dati        matrice di dati

a           parametro di correzione1 (impostato di default pari a 0)

c           parametro di correzione2 (impostato di default pari a 0)

alpha      ampiezza del test (impostato di default pari a 0.05)

errore     parametro che determina l'ampiezza dell'intervallo del p-value  
            stimato (impostato di default pari a  $5e-3$ )

grafico    valore logico. Se è `\code{TRUE}` disegna la distribuzione del  
            test, due rette, una rossa che mi indica il percentile  $1-\alpha$   
            della distribuzione del test e una blu che indica il valore  
            del test calcolato sulla matrice dei dati

### Details

La funzione TRV7 effettua il calcolo del test, gradi di libertà, p-value.....In questo particolare caso il test si distribuisce **ASINTOTICAMENTE** come un Chi-Quadrato.

H0:  $\sigma = I$       ==> con  $\mu$  non nota  
H1:  $\sigma \neq I$

$$TRV7 = -n \log|R|$$

$$\text{sotto } H0: TRV7 \sim \chi^2_{p(p-1)/2}$$

A seconda della scelta effettuata si hanno differenti modi di calcolare il test:

#### **ANALITICO**

Si utilizza la correzione di Bartlett analitica.

#### **EMPIRICO1**

Si utilizza come fattore di correzione la stima della media del test simulato in ipotesi nulla.

## **EMPIRICO2**

Si utilizzano come fattori di correzione la stima della media e della varianza del test simulato in ipotesi nulla.

## **SIMULAZIONE**

Stima della distribuzione del test mediante simulazione.

## **Value**

- valore del test
- gradi di libertà
- p-value

Inoltre, a seconda della scelta effettuata si hanno in più i seguenti valori:

Empirico1:

- parametro di correzione a
- numero replicazioni

Empirico2:

- parametri di correzione a e c
- numero replicazioni

Simulazione:

- intervallo di confidenza del p-value
- numero replicazioni

## **Note**

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## **See Also**

[TRV3](#), [TRV72](#), [TRV9](#)

## **Examples**

```
sigma<-rbind(c(0.12,0.01,0.01),c(0.01,0.3,0.01),c(0.01,0.01,0.55))
sigma
dati<-mvrnorm(n=100,c(rep(0,3)),sigma,tol=1e-6,empirical=FALSE)

TRV7("empirico1",dati,grafico="TRUE")
```

---

## TRV82

Test lambda di Wilks (caso particolare g=2)

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu_1 = \mu_2$ , con  $\sigma_1 = \sigma_2$ .

### Usage

```
TRV82(dati, num, alpha = 0.05, grafico = "FALSE")
```

### Arguments

`dati` matrice di dati mettendo una sotto l'altra le 2 matrici  
`num` numerosità delle 2 matrici di dati  
`alpha` ampiezza del test (impostato di default pari a 0.05)  
`grafico` valore logico. Se è `\code{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione TRV82 effettua il calcolo del test, gradi di libertà, p-value ed eventuale grafico della distribuzione del test. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con  $g_1$  e  $g_2$  gradi di libertà.

$H_0: \mu_1 = \mu_2$              $\implies$  con  $\sigma_1 = \sigma_2$   
 $H_1: \mu_1 \neq \mu_2$

$$TRV82 = [(n-p-1)/p] * [n_1 * n_2 / n] (\mu_1 - \mu_2)' (n_1 * S_1 + n_2 * S_2)^{-1} (\mu_1 - \mu_2)$$

$$\text{sotto } H_0: TRV82 \sim F_{p, n-p-1}$$

### Value

`test`            valore del test  
`g1`             gradi di libertà 1  
`g2`             gradi di libertà 2  
`p-value`        p-value

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[TRV1](#), [TRV2](#), [TRV8](#), [TRV10](#)

## Examples

```
data(iris3)
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
apply(versicolor,2,mean)
apply(virginica,2,mean)
dati<-rbind(versicolor,virginica)
num<-c(50,50)

TRV82(dati,num)
```

---

## TRV8

Test lambda di Wilks

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu_1 = \dots = \mu_g$ , con  $\sigma_1 = \dots = \sigma_g$ .

### Usage

```
TRV8(scelta,dati,num,a=0,c=0,alpha=0.05,errore=0.005,grafico="False")
```

### Arguments

**scelta**      inserire una delle possibili opzioni:  
              ("empirico2","simulazione")

**dati**        matrice di dati mettendo una sotto l'altra le g matrici

**num**         numerosità delle g matrici di dati

**a**           parametro di correzione1 (impostato di default pari a 0)

**c**           parametro di correzione2 (impostato di default pari a 0)

**alpha**      ampiezza del test (impostato di default pari a 0.05)

**errore**     parametro che determina l'ampiezza dell'intervallo del p-value  
              stimato (impostato di default pari a 5e-3)

**grafico**    valore logico. Se è `{TRUE}` disegna la distribuzione del  
              test, due rette, una rossa che mi indica il percentile 1-alpha  
              della distribuzione del test e una blu che indica il valore  
              del test calcolato sulla matrice dei dati

### Details

La funzione TRV8 effettua il calcolo del test, gradi di libertà, p-value....In questo particolare caso il test si distribuisce **ASINTOTICAMENTE** come un Chi-Quadrato.

H0:  $\mu_1 = \dots = \mu_g$       ==> con  $\sigma_1 = \dots = \sigma_g$   
H1:  $\mu_1 \neq \dots \neq \mu_g$

$$TRV8 = -[m-1/2(p-n+1)]\log(\lambda(p,m,n))$$

$$\text{sotto } H_0: TRV8 \sim \chi_{p(g-1)}^2$$

A seconda della scelta effettuata si hanno differenti modi di calcolare il test:

#### **EMPIRICO2**

Si utilizzano come fattori di correzione la stima della media e della varianza del test simulato in ipotesi nulla.

#### **SIMULAZIONE**

Stima della distribuzione del test mediante simulazione.

### **Value**

- valore del test
- gradi di libertà
- p-value

Inoltre, a seconda della scelta effettuata si hanno in più i seguenti valori:

Empirico2:

- parametri di correzione a e c
- numero replicazioni

Simulazione:

- intervallo di confidenza del p-value
- numero replicazioni

### **Note**

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

### **See Also**

[TRV1](#), [TRV2](#), [TRV82](#), [TRV10](#)

### **Examples**

```
data(iris3)
setosa<-iris3[,1]
versicolor<-iris3[,2]
virginica<-iris3[,3]

dati<-rbind(setosa,versicolor,virginica)
num<-c(50,50,50)

TRV8("empirico2",dati,num)
```

---

## TRV9

## Test di Box

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\sigma_1 = \dots = \sigma_g$ .

### Usage

```
TRV9(scelta,dati,num,a=0,c=0,alpha=0.05,errore=0.005,grafico="False")
```

### Arguments

scelta	inserire una delle possibili opzioni: ("analitico","empirico1","empirico2","simulazione")
dati	matrice di dati mettendo una sotto l'altra le g matrici
num	numero di matrici di dati
a	parametro di correzione1 (impostato di default pari a 0)
c	parametro di correzione2 (impostato di default pari a 0)
alpha	ampiezza del test (impostato di default pari a 0.05)
errore	parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a 5e-3)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione TRV9 effettua il calcolo del test, gradi di libertà, p-value....In questo particolare caso il test si distribuisce **ASINTOTICAMENTE** come un Chi-Quadrato.

H0:  $\sigma_1 = \dots = \sigma_g$

H1:  $\sigma_1 \neq \dots \neq \sigma_g$

$$TRV9 = n \log|S| - \sum (n_i \log|S_i|)$$

$$\text{sotto } H0: TRV9 \sim \chi^2_{p(p+1)(g-1)/2}$$

A seconda della scelta effettuata si hanno differenti modi di calcolare il test:

#### ANALITICO

Si utilizza la correzione di Bartlett analitica.

#### EMPIRICO1

Si utilizza come fattore di correzione la stima della media del test simulato in ipotesi nulla.

#### EMPIRICO2

Si utilizzano come fattori di correzione la stima della media e della varianza del test simulato in ipotesi nulla.

## **SIMULAZIONE**

Stima della distribuzione del test mediante simulazione.

### **Value**

- valore del test
- gradi di libertà
- p-value

Inoltre, a seconda della scelta effettuata si hanno in più i seguenti valori:

Empirico1:

- parametro di correzione a
- numero replicazioni

Empirico2:

- parametri di correzione a e c
- numero replicazioni

Simulazione:

- intervallo di confidenza del p-value
- numero replicazioni

### **Note**

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

### **See Also**

[TRV3](#), [TRV6](#), [TRV7](#), [TRV9](#)

### **Examples**

```
data(iris3)
setosa<-iris3[, ,1]
versicolor<-iris3[, ,2]
dati<-rbind(setosa,versicolor)
num<-c(50,50)

TRV9("empirico2",dati,num,grafico="TRUE")
```

---

## TRV10 Test di completa omogeneità

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu_1 = \dots = \mu_g$  e  $\sigma_1 = \dots = \sigma_g$ .

### Usage

```
TRV10(scelta,dati,num,a=0,c=0,alpha=0.05,errore=0.005,grafico="False")
```

### Arguments

scelta	inserire una delle possibili opzioni: ("analitico","empirico1","empirico2","simulazione")
dati	matrice di dati mettendo una sotto l'altra le g matrici
num	numero di osservazioni delle g matrici di dati
a	parametro di correzione1 (impostato di default pari a 0)
c	parametro di correzione2 (impostato di default pari a 0)
alpha	ampiezza del test (impostato di default pari a 0.05)
errore	parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a 5e-3)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-alpha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione TRV10 effettua il calcolo del test, gradi di libertà, p-value....In questo particolare caso il test si distribuisce **ASINTOTICAMENTE** come un Chi-Quadrato.

$H_0: \mu_1 = \dots = \mu_g$  e  $\sigma_1 = \dots = \sigma_g$

$H_1: \mu_1 \neq \dots \neq \mu_g$  e  $\sigma_1 \neq \dots \neq \sigma_g$

$$TRV10 = n \sum (|S_i|) - \sum (n_i \log |S_i|)$$

$$\text{sotto } H_0: TRV10 \sim \chi^2_{p(p+3)(g-1)/2}$$

A seconda della scelta effettuata si hanno differenti modi di calcolare il test:

#### **ANALITICO**

Si utilizza la correzione di Bartlett analitica.

#### **EMPIRICO1**

Si utilizza come fattore di correzione la stima della media del test simulato in ipotesi nulla.

## **EMPIRICO2**

Si utilizzano come fattori di correzione la stima della media e della varianza del test simulato in ipotesi nulla.

## **SIMULAZIONE**

Stima della distribuzione del test mediante simulazione.

## **Value**

- valore del test
- gradi di libertà
- p-value

Inoltre, a seconda della scelta effettuata si hanno in più i seguenti valori:

Empirico1:

- parametro di correzione a
- numero replicazioni

Empirico2:

- parametri di correzione a e c
- numero replicazioni

Simulazione:

- intervallo di confidenza del p-value
- numero replicazioni

## **Note**

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## **See Also**

[TRV8](#), [TRV9](#)

## **Examples**

```
sigma<-rbind(c(0.12,0),c(0,0.14))
dati1<-mvrnorm(n=50,c(3,2.4),sigma,tol=1e-6,empirical=FALSE)
dati2<-mvrnorm(n=50,c(3,2.3),sigma,tol=1e-6,empirical=FALSE)
dati<-rbind(dati1,dati2)
num<-c(50,50)
```

```
TRV10("simulazione",dati,num,grafico="TRUE")
```

## PACKAGE UIT

**Title:** Metodi di simulazione per i test UIT  
**Version:** 1.0  
**Depends:** MASS  
**Date:** 2007-05-13  
**Author:** Michele Battocchio <battocchiomichele@libero.it>, Giancarlo Diana <giancarlo.diana@unipd.it>  
**Maintainer:** Giancarlo Diana <giancarlo.diana@unipd.it>  
**Description:** Questo pacchetto contiene delle semplici funzioni per simulare alcuni Test Unione-Intersezione  
**License:** GPL version 2

### FUNZIONI

UIT1.....	129
UIT1.a.....	130
UIT2.....	131
UIT2.a.....	132
UIT3.....	134
UIT3.a.....	135
UIT4.....	136
UIT4.a.....	138
UIT5.....	139
UIT5.a.....	140
UIT72.....	142
UIT72.a.....	143
UIT82.....	144
UIT82.a.....	146
UIT8.....	147
UIT8.a.....	148

---

## UIT1

Test di ipotesi  $\mu=\mu_0$ , con sigma nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu=\mu_0$ , con sigma nota.

### Usage

```
UIT1(dati,  $\mu_0$ , sigma, alpha = 0.05, grafico = "False")
```

### Arguments

dati	matrice di dati
$\mu_0$	vettore medie $\mu_0$
sigma	matrice di varianza e covarianza
alpha	ampiezza del test (impostato di default pari a 0.05)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$ della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT1 effettua il calcolo del test globale. In questo particolare caso il test si distribuisce **ESATTAMENTE** come un Chi-Quadrato.

$H_0: \mu = \mu_0 \implies$  con S nota

$H_1: \mu \neq \mu_0$

$$UIT1 = n (\mu - \mu_0)' S^{-1} (\mu - \mu_0)$$

$$\text{sotto } H_0: UIT1 \sim \chi_p^2$$

### Value

test	valore del test UIT globale
p-value	p-value
ipotesi	frase che ci dice se accettiamo o non accettiamo l'ipotesi globale

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

### See Also

[UIT1.a](#), [TRV1](#)

## Examples

```
data(iris3)
virginica<-iris3[, ,3]
apply(virginica,2,mean)
μ0<-c(6.5,3,5.5,2.15)
sigma<-var(virginica)

UIT1(virginica,μ0,sigma,grafico="TRUE")
```

---

## UIT1.a

Test di ipotesi a  $\mu = a'\mu_0$ , con sigma nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi a  $\mu = a'\mu_0$ , con sigma nota.

### Usage

```
UIT1.a(a, grafico = "False")
```

### Arguments

**a** vettore p-dimensionale che specifica un'ipotesi componente  
**grafico** valore logico. Se è `{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$  della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT1.a effettua il calcolo del test per la generica componente. In questo particolare caso il test si distribuisce **ESATTAMENTE** come un Chi-Quadrato.

$H_0: a'\mu = a'\mu_0 \quad \Rightarrow$  con S nota  
 $H_1: a'\mu \neq a'\mu_0$

$$UIT1.a = n a' (\mu - \mu_0)' (\mu - \mu_0) a / (a' S^{-1} a)$$

$$\text{sotto } H_0: UIT1.a \sim \chi_p^2$$

### Value

**a** vettore p-dimensionale dell'ipotesi componente  
**test** valore del test UIT ipotesi componente  
**p-value** p-value  
**ipotesi** frase che ci dice se accettiamo o non accettiamo l'ipotesi componente

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT1](#), [TRV1](#)

## Examples

```
data(iris3)
virginica<-iris3[, ,3]
apply(virginica,2,mean)
μ0<-c(6.5,3,5.5,2.15)
sigma<-var(virginica)

UIT1(virginica,μ0,sigma,grafico="TRUE")

a<-c(1,0,0,0)
UIT1.a(a,grafico="TRUE")

a<-c(0,0,0,1)
UIT1.a(a,grafico="TRUE")
```

---

## UIT2

Test di ipotesi  $\mu=\mu_0$ , con sigma non nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu=\mu_0$ , con sigma non nota

### Usage

```
UIT2(dati, μ0, alpha = 0.05, grafico = "FALSE")
```

### Arguments

dati	matrice di dati
μ0	vettore medie μ0
alpha	ampiezza del test (impostato di default pari a 0.05)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-apha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT2 effettua il calcolo del test globale. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con  $g1$  e  $g2$  gradi di libertà.

H0:  $\mu = \mu_0$  ==> con sigma non nota  
H1:  $\mu \neq \mu_0$

$$UIT2 = (n-p)/p (\mu - \mu_0)'S^{-1} (\mu - \mu_0)$$

$$\text{sotto } H_0: UIT2 \sim F_{p, n-p}$$

## Value

test            valore del test UIT globale  
 p-value        p-value  
 ipotesi        frase che ci dice se accettiamo o non accettiamo l'ipotesi globale

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT1](#), [UIT2.a](#), [TRV2](#)

## Examples

```
data(iris3)
virginica<-iris3[, ,3]
apply(virginica,2,mean)
μ0<-c(6.5,3,5.5,2.15)

UIT2(virginica,μ0,grafico="TRUE")
```

---

## UIT2.a            Test di ipotesi a' $\mu$ =a' $\mu_0$ , con sigma non nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi a' $\mu$ =a' $\mu_0$ , con sigma non nota.

### Usage

```
UIT2.a(a, grafico = "False")
```

### Arguments

a                vettore p-dimensionale che specifica un'ipotesi componente  
 grafico        valore logico. Se è `{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$  della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

## Details

La funzione UIT2.a effettua il calcolo del test per la generica componente. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con  $g_1$  e  $g_2$  gradi di libertà.

H0:  $a'\mu = a'\mu_0$  ==>con sigma non nota

H1:  $a'\mu \neq a'\mu_0$

$$UIT2.a = (n-1) a'(\mu - \mu_0)(\mu - \mu_0)'a / (a' \text{sigma} a)$$

$$\text{sotto } H_0: UIT2.a \sim F_{p, n-p}$$

## Value

a	vettore p-dimensionale dell'ipotesi componente
test	valore del test UIT ipotesi componente
valori critici	valori critici della distribuzione del Test
alpha	alpha
ipotesi	frase che ci dice se accettiamo o non accettiamo l'ipotesi componente

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

UIT2, TRV2

## Examples

```
data(iris3)
virginica<-iris3[, ,3]
apply(virginica,2,mean)
mu0<-c(6.5,3,5.5,2.15)

UIT2(virginica,mu0,grafico="TRUE")

a<-c(1,0,0,0)
UIT2.a(a,grafico="TRUE")
```

---

## UIT3

Test di ipotesi  $\sigma = \sigma_0$ , con  $\mu$  non nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\sigma = \sigma_0$ , con  $\mu$  non nota.

### Usage

```
UIT3(dati, sigma0, alpha = 0.05, errore = 0.005, grafico = "False")
```

### Arguments

dati	matrice di dati
sigma0	matrice di varianza e covarianza sotto l'ipotesi nulla
a	parametro di correzione1 (impostato di default pari a 0)
c	parametro di correzione2 (impostato di default pari a 0)
alpha	ampiezza del test (impostato di default pari a 0.05)
errore	parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a $5e-3$ )
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile $1-\alpha$ della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT3 effettua il calcolo del test globale. In questo particolare caso **NON SI CONOSCE** la distribuzione del test.

H0:  $\sigma = \sigma_0$       ==> con  $\mu$  non nota

H1:  $\sigma \neq \sigma_0$

```
UIT3 ==> c1 = lamnda1(sigma0^-1 sigma) |||| cp = lamndap(sigma0^-1 sigma)
```

### Value

test	valore del test UIT globale
valori critici	valori critici della distribuzione del test
alpha.effettivo	alpha stimato
ipotesi	frase che ci dice se accettiamo o non accettiamo l'ipotesi globale
Nreplicazioni	numero replicazioni

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[TRV3](#), [UIT3.a](#)

## Examples

```
data(iris3)
setosa<-iris3[,1]
sigma<-var(setosa)
sigma
sigma0<-rbind(c(0.12,0.10,0.02,0.01),c(0.10,0.15,0.01,0.01),
              c(0.02,0.01,0.03,0.01),c(0.01,0.01,0.01,0.02))
sigma0

UIT3(setosa,sigma0,grafico="TRUE")
```

---

## UIT3.a

 Test di ipotesi a' $\sigma=a'\sigma_0$ , con  $\mu$  non nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi a' $\sigma=a'\sigma_0$ , con  $\mu$  non nota.

### Usage

```
UIT3.a(a, grafico = "False")
```

### Arguments

**a** vettore p-dimensionale che specifica un'ipotesi componente  
**grafico** valore logico. Se è `{TRUE}` disegna la distribuzione del test, quattro rette, due rosse che indicano i percentile  $1-\alpha/2$  dalla distribuzione del test globale e due verdi che indicano i valori del test globale calcolato sui dati originali

### Details

La funzione UIT3.a effettua il calcolo del test per la generica componente. In questo particolare caso **NON SI CONOSCE** la distribuzione del test.

H0:  $a'\sigma = a'\sigma_0$  ==> con  $\mu$  non nota  
H1:  $a'\sigma \neq a'\sigma_0$

$$UIT3.a = (a' \sigma a) / (a' \sigma_0 a)$$

### Value

**a** vettore p-dimensionale dell'ipotesi componente  
**test** valore del test UIT ipotesi componente  
**valori critici** valori critici della distribuzione del Test

alpha	alpha
ipotesi	frase che ci dice se accettiamo o non accettiamo l'ipotesi componente

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

### See Also

[UIT3](#), [TRV3](#)

### Examples

```
data(iris3)
setosa<-iris3[,1]
sigma<-var(setosa)
sigma
sigma0<-rbind(c(0.12,0.10,0.02,0.01),c(0.10,0.15,0.01,0.01),
              c(0.02,0.01,0.03,0.01),c(0.01,0.01,0.01,0.02))
sigma0

UIT3(setosa,sigma0,grafico="TRUE")

a<-c(0,0,0,1)
UIT3.a(a,grafico="TRUE")
```

---

## UIT4

Test di ipotesi  $R\mu=r$ , con sigma nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $R\mu=r$ , con sigma nota.

### Usage

```
UIT4(dati, sigma, R, r, alpha = 0.05, grafico = "False")
```

### Arguments

dati	matrice di dati
sigma	matrice di varianza e covarianza
R	vincoli lineari
r	vettore contenente i valori da verificare con vincoli lineari
alpha	ampiezza del test (impostato di default pari a 0.05)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile $1-\alpha$ della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

## Details

La funzione UIT4 effettua il calcolo del test globale. In questo particolare caso il test si distribuisce **ESATTAMENTE** come un Chi-Quadrato.

H0:  $R\mu = r$  ==>con sigma nota

H1:  $R\mu \neq r$

$$UIT4 = n (R\mu - r)' (R\sigma R')^{-1} (R\mu - r)$$

$$\text{sotto } H0: UIT4 \sim \chi_r^2$$

## Value

test            valore del test UIT globale  
p-value        p-value  
ipotesi        frase che ci dice se accettiamo o non accettiamo l'ipotesi globale

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT4.a](#), [UIT5](#)

## Examples

```
data(iris3)
setosa<-iris3[,1:2,1]
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
dati<-cbind(setosa,versicolor,virginica)
sigma<-var(dati)
R<-rbind(c(0,0,1,0,-1,0),c(0,0,0,1,0,-1))
r<-rbind(c(-0.4),c(0))

UIT4(dati,sigma,R,r,grafico="TRUE")
```

---

## UIT4.a Test di ipotesi $R'a'\mu=a'r$ , con sigma nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $R'a'\mu=a'r$ , con sigma nota.

### Usage

```
UIT4.a(a, grafico = "False")
```

### Arguments

`a` vettore p-dimensionale che specifica un'ipotesi componente  
`grafico` valore logico. Se è `\code{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$  della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT4.a effettua il calcolo del test per la generica componente. In questo particolare caso il test si distribuisce **ESATTAMENTE** come un Chi-Quadrato.

$H_0: R'a'\mu = a'r \implies$  con sigma nota

$H_1: R'a'\mu \neq a'r$

$$UIT4.a = na'(R'\mu-r)(R'\mu-r)'a / (a'R \text{ sigma } R'a)$$

$$\text{sotto } H_0: UIT4.a \sim \chi_r^2$$

### Value

`a` vettore p-dimensionale dell'ipotesi componente  
`test` valore del test UIT ipotesi componente  
`p-value` p-value  
`ipotesi` frase che ci dice se accettiamo o non accettiamo l'ipotesi componente

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

### See Also

[UIT5](#), [TRV4](#)

## Examples

```
data(iris3)
setosa<-iris3[,1:2,1]
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
dati<-cbind(setosa,versicolor,virginica)
sigma<-var(dati)
R<-rbind(c(0,0,1,0,-1,0),c(0,0,0,1,0,-1))
r<-rbind(c(-0.4),c(0))

UIT4(dati,sigma,R,r,grafico="TRUE")

a<-c(0,1)
UIT4.a(a,grafico="TRUE")
```

---

## UIT5

Test di ipotesi  $R^*\mu=r$ , con sigma non nota

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $R^*\mu=r$ , con sigma non nota.

### Usage

```
UIT5(dati, R, r, alpha = 0.05, grafico = "FALSE")
```

### Arguments

dati	matrice di dati
R	vincoli lineari
r	vettore contenente i valori da verificare con vincoli lineari
alpha	ampiezza del test (impostato di default pari a 0.05)
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1-apha della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT5 effettua il calcolo del test globale. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con g11 e g12 gradi di libertà.

H0:  $R^*\mu = r$  ==> con sigma non nota  
H1:  $R^*\mu \neq r$

$$UIT5 = (n-1) (R^*\mu - r)' (R^*S^*R')^{-1} (R^*\mu - r)$$

$$\text{sotto } H0: UIT5 \sim F_{r, n-r}$$

## Value

test            valore del test UIT globale  
p-value        p-value  
ipotesi        frase che ci dice se accettiamo o non accettiamo l'ipotesi globale

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT5.a](#), [TRV5](#)

## Examples

```
data(iris3)
setosa<-iris3[,1:2,1]
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
dati<-cbind(setosa,versicolor,virginica)
R<-rbind(c(0,0,1,0,-1,0),c(0,0,0,1,0,-1))
r<-rbind(c(-0.4),c(0))

UIT5(dati,R,r,grafico="TRUE")
```

-----  
**UIT5.a**            Test di ipotesi  $R^*a'\mu=a'r$ , con sigma non nota  
-----

## Description

Questa funzione permette il calcolo del test di ipotesi  $R^*a'\mu=a'r$ , con sigma non nota.

## Usage

```
UIT5.a(a, grafico = "False")
```

## Arguments

a                vettore p-dimensionale che specifica un'ipotesi componente  
grafico        valore logico. Se è `{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$  della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

## Details

La funzione UIT5.a effettua il calcolo del test per la generica componente. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con  $gl_1$  e  $gl_2$  gradi di libertà.

$H_0: R^*a'\mu = a'r \implies$  con sigma non nota

$H_1: R^*a'\mu \neq a'r$

$$UIT5.a = (n-q)/q a'(R^*\mu - r)(R^*\mu - r)'a / (a'R \text{ sigma } R'a)$$

$$\text{sotto } H_0: UIT5.a \sim F_{r, n-r}$$

## Value

a	vettore p-dimensionale dell'ipotesi componente
test	valore del test UIT ipotesi componente
p-value	p-value
ipotesi	frase che ci dice se accettiamo o non accettiamo l'ipotesi componente

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT5](#), [UIT4](#)

## Examples

```
data(iris3)
setosa<-iris3[,1:2,1]
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
dati<-cbind(setosa,versicolor,virginica)
R<-rbind(c(0,0,1,0,-1,0),c(0,0,0,1,0,-1))
r<-rbind(c(-0.4),c(0))

UIT5(dati,R,r,grafico="TRUE")

a<-c(0,1)
UIT5.a(a,grafico="TRUE")
```

---

## UIT72

Test di ipotesi  $\sigma_1^2=0$ , con  $\mu$  non nota

---

### Description

Questa funzione permette di partizionare il data-set in due con dimensioni rispettivamente  $p_1$  e  $p_2$ , e calcolare il test di ipotesi  $\sigma_1^2=0$ , con  $\mu$  non nota.

### Usage

```
UIT72(dati, p1, p2, alpha = 0.05, errore = 0.005, grafico = "False")
```

### Arguments

dati	matrice di dati
p1	numero variabili partizione 1
p2	numero variabili partizione 2
alpha	ampiezza del test (impostato di default pari a 0.05)
errore	parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a $5e-3$ )
grafico	valore logico. Se è <code>{TRUE}</code> disegna la distribuzione del test, due rette, una rossa che mi indica il percentile $1-\alpha$ della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT72 effettua il calcolo del test globale. In questo particolare caso **NON SI CONOSCE** la distribuzione del test.

$H_0: \sigma_1^2 = 0 \implies$  con  $\mu$  non nota

$H_1: \sigma_1^2 \neq 0$

$$UIT72 = \text{eigen}(\Sigma_{22}^{-1} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12})$$

Nel caso  $p_1=1,2$  o  $p_2=1,2$  il test calcolato si distribuisce **ESATTAMENTE** come una F-Snedecor con  $g_1$  e  $g_2$  gradi di libertà.

### Value

test	valore del test UIT globale
valore critico	valore critico della distribuzione del test
alpha	alpha
ipotesi	frase che ci dice se accettiamo o non accettiamo l'ipotesi globale
Nreplicazioni	numero replicazioni

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT72.a](#), [TRV72](#)

## Examples

```
data(iris3)
setosa<-iris3[,1:2,1]
virginica<-iris3[,1:2,3]
apply(setosa,2,var)
apply(virginica,2,var)
dati<-cbind(setosa,virginica)

UIT72(dati,2,2,grafico="TRUE")
```

---

## UIT72.a

Test di ipotesi a' $\sigma^2 \cdot d=0$ , con  $\mu$  non nota

---

### Description

Questa funzione permette di partizionare il data-set in due con dimensioni rispettivamente  $p_1$  e  $p_2$ , e calcolare il test di ipotesi a' $\sigma^2 d=0$ , con  $\mu$  non nota.

### Usage

```
UIT72.a(a, d, grafico = "False")
```

### Arguments

**a** vettore  $p_1$ -dimensionale che specifica un'ipotesi componente  
**d** vettore  $p_2$ -dimensionale che specifica un'ipotesi componente  
**grafico** valore logico. Se è `{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile  $1-\alpha$  della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione `UIT72.a` effettua il calcolo del test per la generica componente. In questo particolare caso **NON SI CONOSCE** la distribuzione del test.

$H_0: a' \sigma^2 d = 0 \quad \Rightarrow$  con  $\mu$  non nota  
 $H_1: a' \sigma^2 d \neq 0$

$$UIT72.a = (a' \Sigma_{12} d)^2 / (a' \Sigma_{11} a d' \Sigma_{22} b)$$

Nel caso  $p_1=1,2$  o  $p_2=1,2$  il test calcolato si distribuisce **ESATTAMENTE** come una F-Snedecor con  $g_1$  e  $g_2$  gradi di libertà.

## Value

a	vettore p1-dimensionale dell'ipotesi componente
d	vettore p2-dimensionale dell'ipotesi componente
test	valore del test UIT ipotesi componente
valore critico	valore critico della distribuzione del Test
alpha	alpha
ipotesi	frase che ci dice se accettiamo o non accettiamo l'ipotesi componente

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT72](#), [TRV72](#)

## Examples

```
data(iris3)
setosa<-iris3[,1:2,1]
virginica<-iris3[,1:2,3]
apply(setosa,2,var)
apply(virginica,2,var)
dati<-cbind(setosa,virginica)

UIT72(dati,2,2,grafico="TRUE")

a<-c(1,0)
d<-c(0,1)
UIT72.a(a,d,grafico="TRUE")
```

---

## UIT82

Test lambda di Wilks (caso particolare  $g=2$ )

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu_1=\mu_2$ , con  $\sigma_1=\sigma_2$ .

### Usage

```
UIT82(dati, num, alpha = 0.05, grafico = "FALSE")
```

### Arguments

dati	matrice di dati mettendo una sotto l'altra le 2 matrici
num	numerosità delle 2 matrici di dati

alpha        ampiezza del test (impostato di default pari a 0.05)  
grafico        valore logico. Se è `\code{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$  della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

## Details

La funzione UIT82 effettua il calcolo del test globale. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con  $g1$  e  $g2$  gradi di libertà.

H0:  $\mu_1 = \mu_2$                  $\implies$  con  $\sigma_1 = \sigma_2$   
H1:  $\mu_1 \neq \mu_2$

$$TRV82 = [(n-p-1)/p] * [n1*n2/n] (\mu_1 - \mu_2)' (n1*S1 + n2*S2)^{-1} (\mu_1 - \mu_2)$$

$$\text{sotto } H_0: TRV82 \sim F_{p, n-p-1}$$

## Value

test            valore del test UIT globale  
p-value        p-value  
ipotesi        frase che ci dice se accettiamo o non accettiamo l'ipotesi globale

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT82.a](#), [TRV82](#)

## Examples

```
data(iris3)
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
apply(versicolor,2,mean)
apply(virginica,2,mean)
dati<-rbind(versicolor,virginica)
num<-c(50,50)

UIT82(dati,num,grafico="TRUE")
```

-----  
**UIT82.a** Test lambda di Wilks "Unione-Intersezione" (caso particolare g=2)  
-----

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu_1 = \mu_2$ , con  $\sigma_1 = \sigma_2$ .

### Usage

```
UIT82.a(a, grafico = "False")
```

### Arguments

**a** vettore p-dimensionale che specifica un'ipotesi componente  
**grafico** valore logico. Se è `{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile 1- $\alpha$  della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT82.a effettua il calcolo del test per la generica componente. In questo particolare caso il test si distribuisce **ESATTAMENTE** come una F con  $g_1$  e  $g_2$  gradi di libertà.

$H_0: \mu_1 = \mu_2 \implies \text{con } \sigma_1 = \sigma_2$

$H_1: \mu_1 \neq \mu_2$

$$UIT82.a = [(n-p-1)/p] * [n_1 * n_2 / n] a' (\mu_1 - \mu_2)' (n_1 * a' S_1 + n_2 * a' S_2)^{-1} (\mu_1 - \mu_2) a$$

sotto  $H_0: UIT82.a \sim F_{p, n-p-1}$

### Value

**a** vettore p-dimensionale dell'ipotesi componente  
**test** valore del test UIT ipotesi componente  
**p-value** p-value  
**ipotesi** frase che ci dice se accettiamo o non accettiamo l'ipotesi componente

### Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

### See Also

[UIT82](#), [TRV82](#)

## Examples

```
data(iris3)
versicolor<-iris3[,1:2,2]
virginica<-iris3[,1:2,3]
apply(versicolor,2,mean)
apply(virginica,2,mean)
dati<-rbind(versicolor,virginica)
num<-c(50,50)

UIT82(dati,num,grafico="TRUE")

a<-c(1,0)
UIT82.a(a)
```

---

## UIT8

### Test lambda di Wilks

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $\mu_1 = \dots = \mu_g$ , con  $\sigma_1 = \dots = \sigma_g$ .

### Usage

```
UIT8(dati,num,a=0,c=0,alpha=0.05,errore=0.005,grafico="False")
```

### Arguments

**dati** matrice di dati mettendo una sotto l'altra le  $g$  matrici  
**num** numerosità delle  $g$  matrici di dati  
**alpha** ampiezza del test (impostato di default pari a 0.05)  
**errore** parametro che determina l'ampiezza dell'intervallo del p-value stimato (impostato di default pari a  $5e-3$ )  
**grafico** valore logico. Se è `{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile  $1-\alpha$  della distribuzione del test e una blu che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione UIT8 effettua il calcolo del test globale. In questo particolare caso **NON SI CONOSCE** la distribuzione del test.

$H_0: \mu_1 = \dots = \mu_g$  ==> con  $\sigma_1 = \dots = \sigma_g$   
 $H_1: \mu_1 \neq \dots \neq \mu_g$

$$UIT8 = \text{sum}( ni(\mu_i - \mu)(\mu_i - \mu)' / (ni \text{Sigma}_i^2) )$$

### Value

**test** valore del test UIT globale  
**valore critico** valore critico della distribuzione del test  
**alpha** alpha  
**ipotesi** frase che ci dice se accettiamo o non accettiamo l'ipotesi globale  
**Nreplicazioni** numero replicazioni

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT8.a](#), [TRV8](#)

## Examples

```
data(iris3)
versicolor<-iris3[, ,2]
virginica<-iris3[, ,3]
dati<-rbind(versicolor,virginica)
num<-c(50,50)
```

```
UIT8(dati,num,grafico="TRUE")
```

---

## UIT8.a Test di Wilks "Unione-Intersezione"

---

### Description

Questa funzione permette il calcolo del test di ipotesi  $a'\mu_1 = \dots = a'\mu_g$ , con  $a'\Sigma_1 = \dots = a'\Sigma_g$ .

### Usage

```
UIT8.a(a, grafico = "False")
```

### Arguments

**a** vettore p-dimensionale di un'ipotesi componente  
**grafico** valore logico. Se è `{TRUE}` disegna la distribuzione del test, due rette, una rossa che mi indica il percentile  $1-\alpha$  della distribuzione del test e una verde che indica il valore del test calcolato sulla matrice dei dati

### Details

La funzione `UIT8.a` effettua il calcolo del test per la generica componente. In questo particolare caso **NON SI CONOSCE** la distribuzione del test.

H0:  $a'\mu_1 = \dots = a'\mu_g \implies$  con  $a'\Sigma_1 = \dots = a'\Sigma_g$

H1:  $a'\mu_1 \neq \dots \neq a'\mu_g$

$$UIT8.a = \text{sum}( ni(a'(\mu_i - \mu))^2 / (ni a'\Sigma_i a) )$$

## Value

a	vettore p-dimensionale dell'ipotesi componente
test	valore del test UIT ipotesi componente
valore critico	valore critico della distribuzione del Test
alpha	alpha
ipotesi	frase che ci dice se accettiamo o non accettiamo l'ipotesi componente

## Note

Nel grafico, per evidenziare meglio la distribuzione del test, abbiamo deciso di considerare fino al quantile di livello 0.999. Può capitare che il valore del test sia maggiore del quantile 0.999 e di conseguenza non essere disegnato nel grafico.

## See Also

[UIT8](#), [TRV8](#)

## Examples

```
data(iris3)
versicolor<-iris3[, ,2]
virginica<-iris3[, ,3]
dati<-rbind(sversicolor,virginica)
num<-c(50,50)

UIT8(dati,num,grafico="TRUE" )

a<-c(0,0,1,0)
UIT8.a(a,grafico="TRUE" )
```

## APPENDICE: LISTATO FUNZIONI IN R

### Listato funzioni del package TRV

Questo appendice contiene gli algoritmi utilizzati per la realizzazione del package TRV. Parte di questi sono già stati presentati nella tesi.

Qui sotto viene data la lista degli algoritmi presenti in appendice.

Test Rapporto di verosimiglianza (T.R.V.)	Utilizzo	Pagina
TRV1	Test sull'ipotesi che $\mu=\mu_0$ , con $\Sigma$ nota	151
TRV2	Test sull'ipotesi che $\mu=\mu_0$ , con $\Sigma$ non nota	151
TRV3	Test sull'ipotesi che $\Sigma=\Sigma_0$ , con $\mu$ non nota	151
TRV4	Test sull'ipotesi che $R^*\mu=r$ , con $\mu$ non nota	153
TRV5	Test sull'ipotesi che $R^*\mu=r$ , con $\Sigma$ non nota	153
TRV6	Test sull'ipotesi che $\Sigma=k*\Sigma_0$ , con $\mu$ e $k$ non note	153
TRV72	Test sull'ipotesi che $\Sigma_{12}=0$ , con $\mu$ non nota	154
TRV7	Test sull'ipotesi che $\Sigma=I$ , con $\mu$ non nota	156
TRV82	Test sull'ipotesi che $\mu_1=\mu_2$ , con $\Sigma_1=\Sigma_2$	157
TRV8	Test sull'ipotesi che $\mu_1=\dots=\mu_g$ , con $\Sigma_1=\dots=\Sigma_g$	158
TRV9	Test sull'ipotesi che $\Sigma_1=\dots=\Sigma_g$ , test di Box	159
TRV10	Test sull'ipotesi che $\mu_1=\dots=\mu_g$ e $\Sigma_1=\dots=\Sigma_g$	160

Funzioni comuni	Pagina
Nreplicazioni	161
dati.simulati	161
simulazione	161
empirico	162
intervallo.conf	163
grafico	163
scelta.test	163
scelta.test2	163
output.A	163
output.E1	164
output.E2	164
output.S	164
output.N	164
Pause	164

```
#####
#####TEST RAPPORTO DI VEROSIMIGLIANZA#####
#####
```

```
"TRV1" <-function(dati,m0,sigma,alpha=0.05,grafico="False")
{
# mu=m0, sigma nota

n<-dim(dati)[1]
p<-dim(dati)[2]
if(length(m0)!=p) {stop("Test valido solo se length(m0)=length(medie)")}
m<-apply(dati,2,FUN=mean)
gl<-p

test<-n*t(m-m0)%*%solve(sigma)%*%(m-m0)
pvalue<-1-pchisq(test,gl)

if(grafico=="TRUE" ) {
      pvec<-seq(0,0.999,0.0001)
      f<-qchisq(pvec,gl)
      grafico(test,f,alpha)
}

output.A(test,gl,pvalue)
}
```

```
#####
```

```
"TRV2" <-function(dati,m0,alpha=0.05,grafico="FALSE" )
{
# mu=m0, sigma non nota

n<-dim(dati)[1]
p<-dim(dati)[2]
if(length(m0)!=p) {stop("Test valido solo se length(m0)=length(medie)")}
m<-apply(dati,2,FUN=mean)
s<-var(dati)*(n-1)/n
gl1<-p
gl2<-n-p

test<-(n-p)/p*t(m-m0)%*%solve(s)%*%(m-m0)
pvalue<-1-pf(test,gl1,gl2)

if(grafico=="TRUE" ) {
      pvec<-seq(0,0.999,0.0001)
      f<-qf(pvec,gl1,gl2)
      grafico(test,f,alpha)
}

output.N(test,gl1,gl2,pvalue)
}
```

```
#####
```

```
"TRV3" <-function(scelta,dati,sigma0,a=0,c=0,alpha=0.05,errore=5e-3,grafico="False")
{
# Sigma=sigma0, mu non nota

n<-dim(dati)[1]
p<-dim(dati)[2]
```

```

if(dim(sigma0)[1]!=p) {stop("Test valido solo se dim(sigma0)=dim(sigma)")}
scelta.test(scelta)
gl<-p*(p+1)/2
inv.sigma0<-solve(sigma0)
test<-trv3(dati,inv.sigma0,par2=0,par3=0,n,p)

#-----

if(scelta=="analitico")
{
a<-(1-(2*p+1-2/(p+1)))/(6*n))
test<-test*a
pvalue<-1-pchisq(test,gl)

if(grafico=="TRUE" ) {
pvec<-seq(0,0.999,0.0001)
f<-qchisq(pvec,gl)
grafico(test,f,alpha)
}

output<-output.A(test,gl,pvalue)
}

#-----

if(scelta=="korin")
{
v<-0.5*p*(p+1)
D1<-(2*p+1-2/(p+1))/(6*n)
D2<-(p-1)*(p+2)/(6*n^2)
w<-D2-D1^2
gl1<-round(v)
gl2<-round((v+2)/(w))
k<-v/(1-D1-v/gl2)
test<-test/(k)
pvalue<-1-pf(test,gl1,gl2)

if(grafico=="TRUE" ) {
pvec<-seq(0,0.999,0.001)
f<-qf(pvec,gl1,gl2)
grafico(test,f,alpha)
}

output<-output.N(test,gl1,gl2,pvalue)
}

#-----

if(scelta=="empirico1")
{
h<-empirico(scelta,c(rep(0,p)),sigma0,inv.sigma0,par2=0,par3=0,n,p,trv3,test,gl,a,c=0,
alpha,errore,grafico)

output<-output.E1(h[1],gl,h[2],h[3],h[6])
}

#-----

if(scelta=="empirico2")
{
h<-empirico(scelta,c(rep(0,p)),sigma0,inv.sigma0,par2=0,par3=0,n,p,trv3,test,gl,a,c,alpha,
,errore,grafico)

output<-output.E2(h[1],gl,h[2],h[4:5],h[6])
}

#-----

```

```

if(scelta=="simulazione")
{
h<-simulazione(c(rep(0,p)),sigma0,inv.sigma0,par2=0,par3=0,n,p,trv3,test,gl,alpha,errore,
grafico)

output<-output.S(test,gl,h[1],h[2:3],h[4])
}

output
}

#.....

"trv3" <-function(dati,inv.sigma0,par2,par3,n,p)
{
s<-var(dati)*(n-1)/n

ma<-mean(diag(inv.sigma0%%s))
mg<-det(inv.sigma0%%s)^(1/p)
test<-n*p*(ma-log(mg)-1)
}

#####

"TRV4" <-function(dati,sigma,R,r,alpha=0.05,grafico="FALSE")
{
# Rmu=r, sigma nota

dati<-as.matrix(dati)
datir<-dati%%t(R)
m0r<-r
sigmar<-R%%sigma%%t(R)
TRV1(datir,m0r,sigmar,alpha,grafico)
}

#####

"TRV5" <-function(dati,R,r,alpha=0.05,grafico="FALSE")
{
# Rmu=r, sigma non nota

dati<-as.matrix(dati)
datir<-dati%%t(R)
m0r<-r
TRV2(datir,m0r,alpha,grafico)
}

#####

"TRV6" <-function(scelta,dati,sigma0,l=0,a=0,c=0,alpha=0.05,errore=5e-3,grafico="False")
{
# Sigma=k*sigma0, mu e k non nota

n<-dim(dati)[1]
p<-dim(dati)[2]
if(dim(sigma0)[1]!=p) {stop("Test valido solo se dim(sigma0)=dim(sigma)")}
scelta.test2(scelta)
gl<-(p-1-1)*(p-1+2)/2
inv.sigma0<-solve(sigma0)
test<-trv6(dati,inv.sigma0,l,par3=0,n,p)
}

```

```

#-----

if(scelta=="empirico2")
{
h<-empirico(scelta,c(rep(0,p)),sigma0,inv.sigma0,l,par3=0,n,p,trv6,test,gl,a,c,alpha,
  errore,grafico)

output<-output.E2(h[1],gl,h[2],h[4:5],h[6])
}

#-----

if(scelta=="simulazione")
{
h<-simulazione(c(rep(0,p)),sigma0,inv.sigma0,l,par3=0,n,p,trv6,test,gl,alpha,errore,
  grafico)

output<-output.S(test,gl,h[1],h[2:3],h[4])
}

output
}

#.....

"trv6" <-function(dati,inv.sigma0,l,par3,n,p)
{
s<-var(dati)*(n-1)/n
atv<-eigen(inv.sigma0%*%s)$value
a<-mean(atv[(l+1):p])
g<-prod(atv[(l+1):p])^(1/(p-l))
test<-n*(p-l)*log(a/g)
}

#####

"TRV72" <-function(scelta,dati,p1,p2,a=0,c=0,alpha=0.05,errore=5e-3,grafico="False")
{
# sigma12=0, mu non nota

n<-dim(dati)[1]
p<-dim(dati)[2]
if(p1+p2>p) {stop("Errore p1+p2>variabili(dati)")}
scelta.test(scelta)
gl<-p1*p2
I<-diag(1,p2,p2)
test<-trv72(dati,I,p1,p2,n,p)

#-----

if(scelta=="vera")
{
if(p1>2 & p2>2) {stop("Test valido solo per p1=1,2 o p2=1,2")}

if(p1==1 | p2==1)
{
k<-max(p1,p2)
test<-((n-k+1)/k)*(1-exp(test/-n))/exp(test/-n)
gl1<-k
gl2<-n-k+1
pvalue<-1-pf(test,gl1,gl2)
}
}
}

```

```

if(p1==2 | p2==2)
{
k<-max(p1,p2)
test<-((n-k+1)/k)*(1-sqrt(exp(test/-n)))/sqrt(exp(test/-n))
gl1<-2*k
gl2<-2*(n-k+1)
pvalue<-1-pf(test,gl1,gl2)
}

if(grafico=="TRUE" ) {
      pvec<-seq(0,0.999,0.0001)
      f<-qf(pvec,gl1,gl2)
      grafico(test,f,alpha)
}

output<-output.N(test,gl1,gl2,pvalue)
}

#-----

if(scelta=="analitico")
{
a<-(1-(p1+p2+3)/(2*n))
test<-test*a
pvalue<-1-pchisq(test,gl)

if(grafico=="TRUE" ) {
      pvec<-seq(0,0.999,0.0001)
      f<-qchisq(pvec,gl)
      grafico(test,f,alpha)
}

output<-output.A(test,gl,pvalue)
}

h<-c(test,gl,n,p)

#-----

if(scelta=="empirico1")
{
h<-empirico(scelta,c(rep(0,p)),diag(100,p,p),diag(1,p2,p2),p1,p2,n,p,trv72,test,gl,a,c=0,
alpha,errore,grafico)

output<-output.E1(h[1],gl,h[2],h[3],h[6])
}

#-----

if(scelta=="empirico2")
{
h<-empirico(scelta,c(rep(0,p)),diag(100,p,p),diag(1,p2,p2),p1,p2,n,p,trv72,test,gl,a,c,
alpha,errore,grafico)

output<-output.E2(h[1],gl,h[2],h[4:5],h[6])
}

#-----

if(scelta=="simulazione")
{
h<-simulazione(c(rep(0,p)),diag(100,p,p),diag(1,p2,p2),p1,p2,n,p,trv72,test,gl,alpha,
errore,grafico)

output<-output.S(test,gl,h[1],h[2:3],h[4])
}

output
}

```

```

#.....

"trv72" <-function(dati,I,p1,p2,n,p)
{
S11<-var(dati[,1:p1])*(n-1)/n
S22<-var(dati[, (p1+1):(p1+p2)])*(n-1)/n
S12<-cov(dati[,1:p1],dati[, (p1+1):(p1+p2)])*(n-1)/n
S21<-cov(dati[, (p1+1):(p1+p2)],dati[,1:p1])*(n-1)/n
l<-solve(S22)%*%S21%*%solve(S11)%*%S12

test<--n*log(det(I-l))
}

#####

"TRV7" <-function(scelta,dati,a=0,c=0,alpha=0.05,errore=5e-3,grafico="False")
{
# sigma=I, mu non nota

n<-dim(dati)[1]
p<-dim(dati)[2]
scelta.test(scelta)
gl<-p*(p-1)/2
test<-trv7(dati,par1=0,par2=0,par3=0,n,p)

#-----

if(scelta=="analitico")
{
test<-(n-1/6*(2*p+11))*test/n
pvalue<-1-pchisq(test,gl)

if(grafico=="TRUE" ) {
pvec<-seq(0,0.999,0.0001)
f<-qchisq(pvec,gl)
grafico(test,f,alpha)
}

output<-output.A(test,gl,pvalue)
}

#-----

if(scelta=="empirico1")
{
h<-empirico(scelta,c(rep(0,p)),diag(1,p,p),par1=0,par2=0,par3=0,n,p,trv7,test,gl,a,c=0,
alpha,errore,grafico)

output<-output.E1(h[1],gl,h[2],h[3],h[6])
}

#-----

if(scelta=="empirico2")
{
h<-empirico(scelta,c(rep(0,p)),diag(1,p,p),par1=0,par2=0,par3=0,n,p,trv7,test,gl,a,c,
alpha,errore,grafico)

output<-output.E2(h[1],gl,h[2],h[4:5],h[6])
}

#-----

if(scelta=="simulazione")
{
h<-simulazione(c(rep(0,p)),diag(1,p,p),par1=0,par2=0,par3=0,n,p,trv7,test,gl,alpha,
errore,grafico)
}

```

```

output<-output.S(test,g1,h[1],h[2:3],h[4])
}

output
}

#.....

"trv7" <-function(dati,par1,par2,par3,n,p)
{
R<-cor(dati)
test<--n*log(det(R))
}

#####

"TRV82" <-function(dati,num,alpha=0.05,grafico="FALSE")
{
# mu1=mu2, con sigma1=sigma2
# T^2 a due campioni

g<-length(num)
if(g!=2) {stop("Test valido solo per due campioni")}
dati<-as.matrix(dati)
n<-dim(dati)[1]
p<-dim(dati)[2]
n1<-num[1]
n2<-num[2]
W<-matrix(rep(0,p^2),p,p)
ba<-1
m1<-apply(dati[1:n1,],2,FUN=mean)
m2<-apply(dati[(n1+1):n,],2,FUN=mean)
for(i in 1:g)
{
ni<-num[i]
sn<-var(dati[ba:(ni+ba-1),])*ni
W<-W+sn
ba<-ba+ni
}
g11<-p
g12<-n-p-1

test<-t(m1-m2)%*%solve(W)%*%(m1-m2)*n1*n2*g12/(n*g11)
pvalue<-1-pf(test,g11,g12)

if(grafico=="TRUE" ) {
pvec<-seq(0,0.999,0.0001)
f<-qf(pvec,g11,g12)
grafico(test,f,alpha)
}

output.N(test,g11,g12,pvalue)
}

```

```

#####

"TRV8" <-function(scelta,dati,num,a=0,c=0,alpha=0.05,errore=5e-3,grafico="False")
{
# mul=...=mug, con sigma1=...=sigmag

n<-dim(dati)[1]
p<-dim(dati)[2]
scelta.test2(scelta)
g<-length(num)
gl<-p*(g-1)
test<-trv8(dati,num,g,par3=0,n,p)

#-----

if(scelta=="empirico2")
{
h<-empirico(scelta,c(rep(0,p)),diag(100,p,p),num,g,par3=0,n,p,trv8,test,gl,a,c,alpha,
errore,grafico)

output<-output.E2(h[1],gl,h[2],h[4:5],h[6])
}

#-----

if(scelta=="simulazione")
{
h<-simulazione(c(rep(0,p)),diag(100,p,p),num,g,par3=0,n,p,trv8,test,gl,alpha,errore,
grafico)

output<-output.S(test,gl,h[1],h[2:3],h[4])
}

output
}

#.....

"trv8" <-function(dati,num,g,par3,n,p)
{
T<-var(dati)*(n-1)
W<-matrix(rep(0,p^2),p,p)
ba<-1
for(i in 1:g){
ni<-num[i]
sn<-var(dati[ba:(ni+ba-1),])*ni
ba<-ba+ni
W<-W+sn
}
test<-det(W)/det(T)
p1<-p
p2<-n-g
p3<-g-1
test<--(p2-(p1-p3+1)/2)*log(test)
}

```

```

#####

"TRV9" <-function(scelta,dati,num,a=0,c=0,alpha=0.05,errore=5e-3,grafico="False")
{
# sigma1=...=sigmag

n<-dim(dati)[1]
p<-dim(dati)[2]
scelta.test(scelta)
g<-length(num)
gl<-p*(p+1)*(g-1)/2
test<-trv9(dati,num,g,par3=0,n,p)

#-----

if(scelta=="analitico")
{
r<-0
for(i in 1:g)
{
ni<-num[i]
r<-r+1/(ni-1)
}
r<-r-1/(n-g)
f<-1-((2*p^2+3*p-1)/(6*(p+1)*(g-1)))*r
test<-f*test
pvalue<-1-pchisq(test,gl)

if(grafico=="TRUE" ) {
pvec<-seq(0,0.999,0.0001)
f<-qchisq(pvec,gl)
grafico(test,f,alpha)
}

output<-output.A(test,gl,pvalue)
}

#-----

if(scelta=="empirico1")
{
h<-empirico(scelta,c(rep(0,p)),diag(100,p,p),num,g,par3=0,n,p,trv9,test,gl,a,c=0,alpha,
errore,grafico)

output<-output.E1(h[1],gl,h[2],h[3],h[6])
}

#-----

if(scelta=="empirico2")
{
h<-empirico(scelta,c(rep(0,p)),diag(100,p,p),num,g,par3=0,n,p,trv9,test,gl,a,c,alpha,
errore,grafico)

output<-output.E2(h[1],gl,h[2],h[4:5],h[6])
}

#-----

if(scelta=="simulazione")
{
h<-simulazione(c(rep(0,p)),diag(100,p,p),num,g,par3=0,n,p,trv9,test,gl,alpha,errore,
grafico)

output<-output.S(test,gl,h[1],h[2:3],h[4])
}

output
}

```

```

#.....

"trv9" <-function(dati,num,g,par3,n,p)
{
W<-matrix(rep(0,p^2),p,p)
test<-0
ba<-1
for(i in 1:g)
{
ni<-num[i]
sn<-var(dati[ba:(ni+ba-1),])* (ni-1)
ba<-ba+ni
test<-test+ni*log(det(sn/ni))
W<-W+sn
}

test<-n*log(det(W/n))-test
}

#####

"TRV10" <-function(scelta,dati,num,a=0,c=0,alpha=0.05,errore=5e-3,grafico="False")
{
# mul=...=mug e sigmal=...=sigmag

n<-dim(dati)[1]
p<-dim(dati)[2]
scelta.test2(scelta)
g<-length(num)
gl<-p*(p+3)*(g-1)/2
media<-c(rep(0,p))
sigma<-diag(100,p,p)
test<-trv10(dati,num,g,par3=0,n,p)

#-----

if(scelta=="empirico2")
{
h<-empirico(scelta,c(rep(0,p)),diag(10,p,p),num,g,par3=0,n,p,trv10,test,gl,a,c,alpha,
errore,grafico)

output<-output.E2(h[1],gl,h[2],h[4:5],h[6])
}

#-----

if(scelta=="simulazione")
{
h<-simulazione(c(rep(0,p)),diag(10,p,p),num,g,par3=0,n,p,trv10,test,gl,alpha,errore,
grafico)

output<-output.S(test,gl,h[1],h[2:3],h[4])
}

output
}

#.....

"trv10" <-function(dati,num,g,par3,n,p)
{
test<-0
ba<-1
for(i in 1:g){

```

```

        ni<-num[i]
        sn<-var(dati[ba:(ni+ba-1),])*(ni-1)
        test<-test-ni*log(det(sn/ni))
        ba<-ba+ni
    }
test<-test+n*log(det(var(dati)*(n-1)/n))
}

#####FUNZIONI COMUNI#####

"Nreplicazioni" <-function(errore,alpha)
{
N<-signif(((2.57/errore)^2*alpha*(1-alpha)),2)
}

#-----

library(MASS)

"dati.simulati" <-function(media,sigma,n,p,N)
{
dati0<-mvrnorm(n=n*N,media,sigma,tol=1e-6,empirical=FALSE)
}

#-----

"simulazione" <-function(media,sigma,par1,par2,par3,n,p,t,test,gl,alpha,errore,grafico)
{
N<-Nreplicazioni(errore,alpha)
dati0<-dati.simulati(media,sigma,n,p,N)

h<-1
k<-n
y<-NULL

for(i in 1:N)
{
dat<-NULL
dat<-dati0[h:k,]
h<-h+n
k<-k+n
x<-t(dat,par1,par2,par3,n,p)
y<-c(y,x)
}

Ey<-mean(y)
a<-Ey/gl
pvalue<-sum(y>test)/N

if(grafico=="TRUE" ) {
pvec<-seq(0,0.999,0.0001)
f<-qchisq(pvec,gl)
grafico(test,f,alpha)
}

intervallo<-intervallo.conf(pvalue,N)

h<-c(pvalue,intervallo,N)
}

```

```

#-----

"empirico" <-function(scelta,media,sigma,par1,par2,par3,n,p,t,test,gl,a,c,alpha,errore,
                    grafico)
{
if(a==0 & c==0)
{
  N<-Nreplicazioni(errore,alpha)
  dati0<-dati.simulati(media,sigma,n,p,N)

  h<-1
  k<-n
  y<-NULL

  for(i in 1:N)
  {
    dat<-NULL
    dat<-dati0[h:k,]
    h<-h+n
    k<-k+n
    x<-t(dat,par1,par2,par3,n,p)
    y<-c(y,x)
  }

  Ey<-mean(y)
  v<-var(y)
  a1<-Ey/gl
  a2<-sqrt(v/(2*gl))
  c<-gl-Ey/a2
}

else
{
  N<-0
  a1<-0
  a2<-0
  if(scelta=="empirico1") {a1<-a}
  if(scelta=="empirico2") {a2<-a}
}

if(scelta=="empirico1")
{
  test<-test/a1
  pvalue<-1-pchisq(test,gl)
  pvec<-seq(0,0.999,0.0001)
  f<-qchisq(pvec,gl)
}

if(scelta=="empirico2")
{
  test<-test/a2+c
  pvalue<-1-pchisq(test,gl)
  pvec<-seq(0,0.999,0.0001)
  f<-qchisq(pvec,gl)
}

if(grafico=="TRUE") {grafico(test,f,alpha)}

h<-c(test,pvalue,a1,a2,c,N)
}

```

```

#-----

"intervallo.conf" <-function(pvalue,N)
{
errstimato<-((pvalue*(1-pvalue))/N)^(1/2)
er99<-2.57*errstimato
intervallo<-c(pvalue-er99,pvalue+er99)
}

#-----

"grafico" <-function(test,f,alpha)
{
qmax<-quantile(f,1-alpha)
w<-density(f)
plot(w,xlim=c(0,max(f)),main="Distribuzione del test",xlab="valori del
      test",ylab="densità")

abline(v=qmax,lty=2, col="red")
abline(v=test, lty=6, col="blue")
pause()
}

#-----

"scelta.test" <-function(scelta)
{

if((scelta=="analitico")|(scelta=="empirico1")|(scelta=="empirico2")|
    (scelta=="simulazione")|(scelta=="korin")|(scelta=="vera"))
{
}

else {stop("Digitare correttamente la scelta")}

}

#-----

"scelta.test2" <-function(scelta)
{

if((scelta=="empirico2")|(scelta=="simulazione"))
{
}

else {stop("Digitare correttamente la scelta")}

}

#-----

"output.A" <-function(test,gl,pvalue)
{
print("-*-*-*-*-*-*-*-*Risultati del test*-*-*-*-*-*-*-*-*")
list(test=test, gl=gl, 'p-value'=pvalue)
}

```

```

#-----

"output.E1" <-function(test,gl,pvalue,a,N)
{
print("--*--*--*--*--*--Risultati del test*--*--*--*--*--*--*--")
list(test=test, gl=gl, 'p-value'=pvalue, 'parametro di correzione a'=a,Nreplicazioni=N)
}

#-----

"output.E2" <-function(test,gl,pvalue,q,N)
{
print("--*--*--*--*--*--Risultati del test*--*--*--*--*--*--*--")
list(test=test, gl=gl, 'p-value'=pvalue, 'parametri di correzione a e
c'=q,Nreplicazioni=N)
}

#-----

"output.S" <-function(test,gl,pvalue,intervallo,N)
{
print("--*--*--*--*--*--Risultati del test*--*--*--*--*--*--*--")
list(test=test, gl=gl, 'p-value'=pvalue, 'intervallo di confidenza del
p-value'=intervallo,Nreplicazioni=N)
}

#-----

"output.N" <-function(test,gl1,gl2,pvalue)
{
print("--*--*--*--*--*--Risultati del test*--*--*--*--*--*--*--")
list(test=test, gl1=gl1, gl2=gl2, 'p-value'=pvalue)
}

#-----

"pause" <-function ()
{
cat("Pause. Press <Enter> to continue...")
readline()
invisible()
}

```

## Listato funzioni del package UIT

Presentiamo gli algoritmi utilizzati per la realizzazione del package UIT.

Qui sotto viene data la lista delle funzioni.

Test Unione-Intersezione (U.I.T.)	Utilizzo	Pagina
UIT1	Test sull'ipotesi che $\mu=\mu_0$ , con $\Sigma$ nota	166
UIT1.a	Test di ipotesi che $a'\mu=a'\mu_0$ , con $\Sigma$ nota	166
UIT2	Test sull'ipotesi che $\mu=\mu_0$ , con $\Sigma$ non nota	166
UIT2.a	Test di ipotesi che $a'\mu=a'\mu_0$ , con $\Sigma$ non nota	167
UIT3	Test sull'ipotesi che $\Sigma=\Sigma_0$ , con $\mu$ non nota	167
UIT3.a	Test di ipotesi che $a'\Sigma=a'\Sigma_0$ , con $\mu$ non nota	169
UIT4	Test sull'ipotesi che $R^*\mu=r$ , con $\Sigma$ nota	170
UIT4.a	Test di ipotesi che $R^*a'\mu=a'r$ , con $\Sigma$ nota	170
UIT5	Test sull'ipotesi che $R^*\mu=r$ , con $\Sigma$ non nota	171
UIT5.a	Test di ipotesi che $R^*a'\mu=a'r$ , con $\Sigma$ non nota	171
UIT72	Test di ipotesi che $\Sigma_{12}=0$ , con $\mu$ non nota	172
UIT72.a	Test di ipotesi che $a'\Sigma_{12}d=0$ , con $\mu$ non nota	173
UIT82	Test sull'ipotesi che $\mu_1=\mu_2$ , con $\Sigma_1=\Sigma_2$	174
UIT82.a	Test sull'ipotesi che $a'\mu_1=a'\mu_2$ , con $a'\Sigma_1=a'\Sigma_2$	174
UIT8	Test sull'ipotesi che $\mu_1=\dots=\mu_g$ , con $\Sigma_1=\dots=\Sigma_g$	175
UIT8.a	Test sull'ipotesi che $a'\mu_1=\dots=a'\mu_g$ , con $a'\Sigma_1=\dots=a'\Sigma_g$	176

Funzioni comuni	Pagina
Nreplicazioni	176
dati.simulati	176
simulazione	177
grafico1	177
grafico2	177
output.N	178
output.Na	178
output.S	178
Pause	178

```
#####
#####TEST UNIONE-INTERSEZIONE#####
#####
```

```
"UIT1" <-function(dati,m0,sigma,alpha=0.05,grafico="False")
{
# mu=m0, sigma nota

unlink("salva.Rdata")
n<-dim(dati)[1]
p<-dim(dati)[2]
if(length(m0)!=p) {stop("Test valido solo se length(m0)=length(medie)")}
m<-apply(dati,2,FUN=mean)
gl<-p

test<-n*t(m-m0)%*%solve(sigma)%*(m-m0)
pvalue<-1-pchisq(test,gl)
qmax<-qchisq(1-alpha,gl)
pvec<-seq(0,0.999,0.0001)
f<-qchisq(pvec,gl)

e<-graficol(test,qmax,f,uit="UIT1.a",grafico)
pause()
uit="UIT1"
save(uit,dati,m0,sigma,alpha,m,f,gl,qmax,n,p, file = "salva.Rdata")

output.N(test,pvalue,e)
}
```

```
#####
```

```
"UIT1.a" <-function(a,grafico="False")
{
# mu=m0, sigma nota

load("salva.Rdata")
if(uit!="UIT1") {stop("Calcolare prima il test globale")}
if(length(a)!=p) {stop("Test valido solo se length(a)=variabili dei dati")}

test.a<-n*t(a)%*(m-m0)%*%t(m-m0)%*%a/(t(a)%*%sigma%*%a)

pvalue<-1-pchisq(test.a,gl)

e<-grafico2(test.a,qmax,f,grafico)
pause()

output.Na(a,test.a,pvalue,e)
}
```

```
#####
```

```
"UIT2" <-function(dati,m0,alpha=0.05,grafico="False")
{
# mu=m0, sigma non nota

unlink("salva.Rdata")
n<-dim(dati)[1]
p<-dim(dati)[2]
if(length(m0)!=p) {stop("Test valido solo se length(m0)=length(medie)")}
m<-apply(dati,2,FUN=mean)
s<-var(dati)*(n-1)/n
gl1<-p
gl2<-n-p
```

```

test<-(n-p)/p*t(m-m0)%*%solve(s)%*%(m-m0)
pvalue<-1-pf(test,g11,g12)
qmax<-qf(1-alpha,g11,g12)
pvec<-seq(0,0.999,0.0001)
f<-qf(pvec,g11,g12)

e<-grafico1(test,qmax,f,uit="UIT2.a",grafico)
pause()
uit="UIT2"
save(uit,dati,m0,s,alpha,m,f,g11,g12,qmax,n,p, file = "salva.Rdata")

output.N(test,pvalue,e)
}

#####

"UIT2.a" <-function(a,grafico="False")
{
# mu=m0, sigma non nota

test<-0
load("salva.Rdata")
if(uit!="UIT2") {stop("Calcolare prima il test globale")}

test.a<-(n-p)/p*t(a)%*%(m-m0)%*%t(m-m0)%*%a/(t(a)%*%s%*%a)

pvalue<-1-pf(test.a,g11,g12)

e<-grafico2(test.a,qmax,f,grafico)
pause()

output.Na(a,test.a,pvalue,e)
}

#####

"UIT3" <-function(dati,sigma0,alpha=0.05,errore=5e-3,grafico="False")
{
# sigma=sigma0, con mu non nota

unlink("salva.Rdata")
n<-dim(dati)[1]
p<-dim(dati)[2]
if(dim(sigma0)[1]!=p) {stop("Test valido solo se dim(sigma0)=dim(sigma)")}
inv.sigma0<-solve(sigma0)
test<-uit3(dati,inv.sigma0,par2=0,par3=0,n,p)
t1<-test[1]
tp<-test[p]
t<-c(tp,t1)

N<-Nreplicazioni(errore,alpha)
k<-simulazione(c(rep(0,p)),sigma0,inv.sigma0,par2=0,par3=0,N,n,p,uit3)

x2<-k[,1]
y2<-k[,p]
x1<-sort(x2)
y1<-sort(y2)
d<-cbind(x1,y1)
j<-0

qmax<-quantile(x1,1-alpha/2)

```

```

qmin<-quantile(y1,0+alpha/2)

  for(i in 1:N)
    {
      if(x2[i]<qmax & y2[i]>qmin) {j<-j+1}
    }

numero1<-N*(1-alpha/2)+1
numero2<-N*(alpha/2)+1
l<-0

while(j!=N*(1-alpha))
{
j<-0
if(l==0) {numero1<-numero1-1}
if(l==1) {numero2<-numero2-1}
qmax<-x1[numero1]
qmin<-y1[numero2]

for(i in 1:N)
  {
    if(x2[i]<qmax & y2[i]>qmin) {j<-j+1}
  }

if(l==0) {l=1}
if(l!=0) {l=0}

alpha.effettivo<-j/N-1+2*alpha
}

q<-c(qmin,qmax)

uit="UIT3"
save(uit,d,qmax,qmin,t,dati,sigma0,alpha,N,n,p, file = "salva.Rdata")

if(grafico=="TRUE")
{
par(mfrow=c(1,2))
pvec<-seq(0.001,0.999,0.0001)
f2<-quantile(d[,2],pvec)
w2<-density(f2)
plot(w2,xlim=c(min(f2),max(f2)),main="Distribuzione lambda-1 ",
xlab="valori di lambda-1",ylab="densità")
abline(v=qmin,lty=2, col="red")
abline(v=t[1], lty=6, col="blue")

f1<-quantile(d[,1],pvec)
w1<-density(f1)
plot(w1,xlim=c(min(f1),max(f1)),main="Distribuzione lambda-p",
xlab="valori di lambda-p",ylab="densità")
abline(v=qmax,lty=2, col="red")
abline(v=t[2], lty=6, col="blue")

par(mfrow=c(1,1))
}

if(t[2]>qmax | t[1]<qmin)
{
e1<-c("Rifiutiamo l'ipotesi globale")
e2<-c("Si può verificare se una qualche <ipotesi componente>")
e3<-c("abbia portato al rifiuto con:")
e<-c(e1,e2,e3,"UIT3.a")
}

if(t[2]<qmax & t[1]>qmin)
{

```

```

e<-c("Accettiamo l'ipotesi globale")
}

pause()

print("--*-*-*--*--*--Risultati del test*-*-*--*--*--*--*--")
list('Valore del test U.I.T. globale'=t,
      'Valori critici della distribuzione del Test'=q,
      alpha.effettivo=alpha.effettivo,Ipotesi=e, Nreplicazioni=N)
}

#-----

"uit3" <-function(dati,inv.sigma0,par2=0,par3=0,n,p)
{
# Sigma=sigma0, mu non nota

s<-var(dati)*(n-1)/n
test<-eigen(inv.sigma0*s)$values
}

#####

"UIT3.a" <-function(a,grafico="FALSE")
{
# Sigma=sigma0, mu non nota

test<-0
load("salva.Rdata")
if(uit!="UIT3") {stop("Calcolare prima il test globale")}
if(length(a)!=p) {stop("Test valido solo se length(a)=variabili dei dati")}

test.a<-uit3.a(dati,sigma0,a)

if(grafico=="TRUE")
{
par(mfrow=c(1,2))
pvec<-seq(0.001,0.999,0.001)
f2<-quantile(d[,2],pvec)
w2<-density(f2)
plot(w2,xlim=c(min(f2),max(f2)),main="Distribuzione lambda-1 ",
xlab="valori di lambda-1",ylab="densità")
abline(v=qmin,lty=2, col="red")
abline(v=test.a, lty=6, col="green")

f1<-quantile(d[,1],pvec)
w1<-density(f1)
plot(w1,xlim=c(min(f1),max(f1)),main="Distribuzione lambda-p",
xlab="valori di lambda-p",ylab="densità")
abline(v=qmax,lty=2, col="red")
abline(v=test.a, lty=6, col="green")

par(mfrow=c(1,1))
}

if(test.a>qmax | test.a<qmin)
{
e<-c("Rifiutiamo l'ipotesi componente")
}

if(test.a<qmax & test.a>qmin)
{

```



```

test.a<-n*t(a)**(R**m-r)**t(R**m-r)**a/t(a)**(R**sigma**t(R))**a
pvalue<-1-pchisq(test.a,gl)
e<-grafico2(test.a,qmax,f,grafico)
pause()

output.Na(a,test.a,pvalue,e)
}

#####

"UIT5" <-function(dati,R,r,alpha=0.05,grafico="False")
{
# Rmu=r, sigma non nota

unlink("salva.Rdata")
n<-dim(dati)[1]
p<-dim(dati)[2]
m<-apply(dati,2,FUN=mean)
s<-var(dati)*(n-1)/n
q<-dim(R)[1]
gl1<-q
gl2<-n-q

test<-((n-q)/q)*t(R**m-r)**solve(R**s**t(R))**m-r)

pvalue<-1-pf(test,gl1,gl2)
qmax<-qf(1-alpha,gl1,gl2)
pvec<-seq(0,0.999,0.0001)
f<-qf(pvec,gl1,gl2)

e<-grafico1(test,qmax,f,uit="UIT5.a",grafico)
pause()
uit="UIT5"
save(uit,dati,R,r,s,alpha,m,f,q,gl1,gl2,qmax,n,p, file = "salva.Rdata")

output.N(test,pvalue,e)
}

#####

"UIT5.a" <-function(a,grafico="False")
{
# Rmu=r, sigma non nota

test<-0
load("salva.Rdata")
if(uit!="UIT5") {stop("Calcolare prima il test globale")}

test.a<-((n-q)/q)*t(a)**(R**m-r)**t(R**m-r)**a/t(a)**(R**s**t(R))**a

pvalue<-1-pf(test.a,gl1,gl2)

e<-grafico2(test.a,qmax,f,grafico)
pause()

output.Na(a,test.a,pvalue,e)
}

```

```
#####
```

```
"UIT72" <-function(dati,p1,p2,alpha=0.05,errore=5e-3,grafico="False")
{
# sigma12=0, mu non nota

n<-dim(dati)[1]
p<-dim(dati)[2]
if(p1+p2>p) {stop("Errore p1+p2>variabili(dati)")}
g1<-p1*p2
I<-diag(1,p2,p2)
pvec<-seq(0,0.999,0.0001)
N<-0
test<-uit72(dati,p1,p2,I,n,p)

S11<-var(dati[,1:p1])*(n-1)/n
S22<-var(dati[, (p1+1):(p1+p2)])*(n-1)/n
S12<-cov(dati[,1:p1],dati[, (p1+1):(p1+p2)])*(n-1)/n
S21<-cov(dati[, (p1+1):(p1+p2)],dati[,1:p1])*(n-1)/n

l<-solve(S22)%*%S21%*%solve(S11)%*%S12
h<-det(I-l)

if(p1==1 | p2==1)
{
k<-max(p1,p2)
test<-((n-k+1)/k)*((1-h)/h)
g11<-k
g12<-n-k+1
pvalue<-1-pf(test,g11,g12)
qmax<-qf(1-alpha,g11,g12)
f<-qf(pvec,g11,g12)
}

if(p1==2 | p2==2)
{
k<-max(p1,p2)
test<-((n-k+1)/k)*(1-sqrt(h))/sqrt(h)
g11<-2*k
g12<-2*(n-k+1)
pvalue<-1-pf(test,g11,g12)
qmax<-qf(1-alpha,g11,g12)
f<-qf(pvec,g11,g12)
}

if(!((p1==1 | p2==1)|(p1==2 | p2==2)))
{
unlink("salva.Rdata")
N<-Nreplicazioni(errore,alpha)
k<-simulazione(c(rep(0,p)),diag(100,p,p),p1,p2,I,N,n,p,uit72)
x1<-sort(k)
qmax<-quantile(x1,1-alpha)
f<-quantile(x1,pvec)
}

e<-graficol(test,qmax,f,uit="UIT72.a(a,d)",grafico)
pause()

uit="UIT72"
save(uit,f,qmax,dati,p1,p2,test,alpha,N,n,p, file = "salva.Rdata")

print("Risultati della simulazione")
print("-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*")
list('Valore del test UIT globale'=test,
      'Valore critico della distribuzione del Test'=qmax, alpha=alpha,
      Ipotesi=e, Nreplicazioni=N)
}
```



```

"UIT82" <-function(dati,num,alpha=0.05,grafico="False")
{
# mu1=mu2, con signal=sigma2
# T^2 a due campioni

unlink("salva.Rdata")

g<-length(num)
if(g!=2) {stop("Test valido solo per due campioni")}
n<-dim(dati)[1]
p<-dim(dati)[2]
n1<-num[1]
n2<-num[2]
W<-matrix(rep(0,p^2),p,p)
ba<-1
m1<-apply(dati[1:n1,],2,FUN=mean)
m2<-apply(dati[(n1+1):n,],2,FUN=mean)
for(i in 1:g){
  ni<-num[i]
  sn<-var(dati[ba:(ni+ba-1),])*ni
  W<-W+sn
  ba<-ba+ni
}
g11<-p
g12<-n-p-1

test<-t(m1-m2)%*%solve(W)%*%(m1-m2)*n1*n2*g12/(n*g11)

pvalue<-1-pf(test,g11,g12)
qmax<-qf(1-alpha,g11,g12)
pvec<-seq(0,0.999,0.0001)
f<-qf(pvec,g11,g12)

e<-graficol(test,qmax,f,uit="UIT82.a(a)",grafico)

uit="UIT82"
save(uit,dati,num,alpha,g,f,qmax,g11,g12, file= "salva.Rdata")

output.N(test,pvalue,e)
}

#####

"UIT82.a" <-function(a,grafico="False")
{
# mu1=...=mug, con signal=...=sigmag

test<-0
load("salva.Rdata")
if(uit!="UIT82") {stop("Calcolare prima il test globale")}

media<-mean(dati)
w<-0
q<-0
ba<-1
for(i in 1:g){
  ni<-num[i]
  sn<-(ni-1)%*%t(a)%*%var(dati[ba:(ni+ba-1),])%*%a
  w<-w+sn
  m<-apply(dati[ba:(ni+ba-1),],2,FUN=mean)
  q1<-ni%*%(t(a)%*%(m-media))^2
  q<-q+q1
  ba<-ba+ni
}

test.a<-q/w

```



```

#####

"UIT8.a" <-function(a,grafico="False")
{
# mul=...=mug, con signal=...=sigmag

test<-0
load("salva.Rdata")
if(uit!="UIT8") {stop("Calcolare prima il test globale")}

test.a<-uit8.a(dati,num,g,a,qmax)

e<-grafico2(test.a,qmax,f,grafico)
pause()

print("Risultati della simulazione")
print("-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*")
list('Vettore a: '=a, 'Valore del test UIT componente '=test.a,
      'Valore critico della distribuzione del Test '=qmax,
      alpha=alpha, Ipotesi=e)
}

#-----

"uit8.a" <-function(dati,num,g,a,qmax)
{
# mul=...=mug, con signal=...=sigmag

media<-mean(dati)
w<-0
q<-0
ba<-1
for(i in 1:g){
  ni<-num[i]
  sn<-((ni-1)**t(a)**var(dati[ba:(ni+ba-1),]))**a
  w<-w+sn
  m<-apply(dati[ba:(ni+ba-1),],2,FUN=mean)
  q1<-ni**a*(a**m-media)^2
  q<-q+q1
  ba<-ba+ni
}

test<-q/w
}

#####FUNZIONI COMUNI#####

"Nreplicazioni" <-function(errore,alpha)
{
N<-signif(((2.57/errore)^2*alpha*(1-alpha)),2)
}

#-----

library(MASS)

"dati.simulati" <-function(media,sigma,n,p,N)
{
dati0<-mvrnorm(n=n*N,media,sigma,tol=1e-6,empirical=FALSE)
}

```

```

#-----

"simulazione" <-function(media,sigma,par1,par2,par3,N,n,p,t)
{
dati0<-dati.simulati(media,sigma,n,p,N)

h<-1
k<-n
y<-NULL

for(i in 1:N)
  {
  dat<-NULL
  dat<-dati0[h:k,]
  h<-h+n
  k<-k+n
  x<-t(dat,par1,par2,par3,n,p)
  y<-rbind(y,x)
  }
y
}

#-----

"grafico1" <-function(test,qmax,f,uit,grafico)
{

if(grafico=="TRUE" )
{
w<-density(f)
plot(w,xlim=c(0,max(f)),main="Distribuzione del test UIT globale",
xlab="valori del test uit globale",ylab="densità")
abline(v=qmax,lty=2, col="red")
abline(v=test, lty=6, col="blue")
}

if(test>qmax)
{
e1<-c("Rifiutiamo l'ipotesi globale")
e2<-c("Si può verificare se una qualche <ipotesi componente>")
e3<-c("abbia portato al rifiuto con:")
e<-c(e1,e2,e3,uit)
}

if(test<qmax)
{
e<-c("Accettiamo l'ipotesi globale")
}
e
}

#-----

"grafico2" <-function(test.a,qmax,f,grafico)
{

if(grafico=="TRUE" )
{
w<-density(f)
plot(w,xlim=c(0,max(f)),main="Distribuzione del test UIT componente",
xlab="valori del test uit ipotesi componente",ylab="densità")
abline(v=qmax,lty=2, col="red")
abline(v=test.a, lty=6, col="green3")
}
}

```

```

}

if(test.a>qmax)
{
e<-c("Rifiutiamo l'ipotesi componente")
}

if(test.a<qmax)
{
e<-c("Accettiamo l'ipotesi componente")
}
e
}

#-----

"output.N" <-function(test,pvalue,e)
{
print("--*--*--*--*--Risultati del test*--*--*--*--*--*--*")
list('Valore del test UIT globale'=test,'p-value'=pvalue,
      'Ipotesi'=e)
}

#-----

"output.Na" <-function(a,test.a,pvalue,e)
{
print("--*--*--*--*--Risultati del test*--*--*--*--*--*--*")
list('Vettore a: '=a, 'Valore del test UIT ipotesi componente'=test.a,
      'p-value'=pvalue,'Ipotesi'=e)
}

#-----

"output.S" <-function(a,test.a,pvalue,e)
{
print("--*--*--*--*--Risultati del test*--*--*--*--*--*--*")
list('Vettore a: '=a, 'Valore del test UIT ipotesi componente'=test.a,
      'p-value'=pvalue,'Ipotesi'=e)
}

#-----

"pause" <-function ()
{
  cat("Pause. Press <Enter> to continue...")
  readline()
  invisible()
}

```

## Listato funzioni per l'analisi della bontà dei metodi

Analisi della bontà dei metodi		Pagina
Bontà3	Analisi della bontà dei metodi sul test TRV3	179
Bontà72	Analisi della bontà dei metodi sul test TRV72	181
Bontà7	Analisi della bontà dei metodi sul test TRV7	183
Bontà9	Analisi della bontà dei metodi sul test TRV9	185

```
#####
#####FUNZIONI PER L'ANALISI DELLA BONTÀ DEI METODI#####
#####
```

```
bontà3<-function(p,n,alpha=0.05,errore=5e-3)
{
nr<-10000
pvasintotico<-0
pvanalitica<-0
pvempirical<-0
pvempirica2<-0
pvkorin<-0
pvsimulazione<-0

media<-c(rep(0,p))
sigma<-diag(100,p,p)
inv.sigma<-solve(sigma)

dati<-mvrnorm(n=n,media,sigma,tol=1e-6,empirical=FALSE)

h<-p.e.trv3(n,p,alpha,errore)
a1<-h[1]
a2<-h[2]
c<-h[3]
k<-trv3.korin(dati,inv.sigma)

g1<-p*(p+1)/2
q1<-k[2]
q2<-k[3]

soglia<-qchisq(1-alpha,g1)
sogliasimulazione<-h[4]
sogliakorin<-qf(1-alpha,q1,q2)

dati0<-mvrnorm(n=n*nr,media,sigma,tol=1e-6,empirical=FALSE)
h<-1
k<-n
y<-0

for(i in 1:nr)
{
dat<-NULL
dat<-dati0[h:k,]
h<-h+n
k<-k+n
t<-trv3.n(dat,inv.sigma)
x<-trv3.bart(dat,inv.sigma)
e1<-t/a1
e2<-t/a2+c
b<-trv3.korin(dat,inv.sigma)[1]
```

```

        if(t>soglia)                pvasintotico<-pvasintotico+1
        if(x>soglia)                pvanalitica<-pvanalitica+1
        if(e1>soglia)               pvempirical<-pvempirical+1
        if(e2>soglia)               pvempirica2<-pvempirica2+1
        if(b>sogliakorin)           pvkorin<-pvkorin+1
        if(t>sogliasimulazione)    pvsimulazione<-pvsimulazione+1
    }

pvasintotico<-pvasintotico/nr
pvanalitica<-pvanalitica/nr
pvempirical<-pvempirical/nr
pvempirica2<-pvempirica2/nr
pvkorin<-pvkorin/nr
pvsimulazione<-pvsimulazione/nr

h<-c(gl,pvasintotico,pvanalitica,pvempirical,pvempirica2,pvkorin,pvsimulazione)
print(h)
}

#-----

trv3.n<-function(dati,inv.sigma)
{

n<-dim(dati)[1]
p<-dim(dati)[2]
s<-var(dati)*(n-1)/n
ma<-mean(diag(inv.sigma**s))
mg<-det(inv.sigma**s)^(1/p)
test<-n*p*(ma-log(mg)-1)
}

#-----

trv3.bart<-function(dati,inv.sigma)
{
# Sigma=sigma, mu non nota

n<-dim(dati)[1]
p<-dim(dati)[2]
s<-var(dati)*(n-1)/n
ma<-mean(diag(inv.sigma**s))
mg<-det(inv.sigma**s)^(1/p)
a<-(1-(2*p+1-2/(p+1)))/(6*n))
test<-n*p*(ma-log(mg)-1)*a

test
}

#-----

trv3.korin<-function(dati,inv.sigma)
{
# Sigma=sigma, mu non nota

n<-dim(dati)[1]
p<-dim(dati)[2]
s<-var(dati)*(n-1)/n
ma<-mean(diag(inv.sigma**s))
mg<-det(inv.sigma**s)^(1/p)
v<-0.5*p*(p+1)
D1<-(2*p+1-2/(p+1))/(6*n)
D2<-(p-1)*(p+2)/(6*n^2)
w<-D2-D1^2
q1<-v
q2<-(v+2)/(w)
b<-v/(1-D1-v/q2)

test<-n*p*(ma-log(mg)-1)/(b)
}

```

```

h<-c(test,q1,q2)
}

#-----

p.e.trv3<-function(n,p,alpha,errore)
{

y1<-NULL
Sy<-0
v<-0
media<-c(rep(0,p))
sigma<-diag(100,p,p)
gl<-p*(p+1)/2
inv.sigma<-solve(sigma)

N<-signif(((2.57/errore)^2*alpha*(1-alpha)),2)

dati0<-mvrnorm(n=n*N,media,sigma,tol=1e-6,empirical=FALSE)
h<-1
k<-n
y<-0

for(i in 1:N)
{
dat<-NULL
dat<-dati0[h:k,]
h<-h+n
k<-k+n
x<-trv3.n(dat,inv.sigma)
y<-c(y,x)
}

Ey<-mean(y)
v<-var(y)
a1<-Ey/gl
a2<-sqrt(v/(2*gl))
c<-gl-Ey/a2

y<-sort(y)
percentile<-quantile(y,1-alpha)

h<-c(a1,a2,c,percentile)
}

#####

bontà72<-function(p1,p2,n,alpha=0.05,errore=5e-3)
{

nr<-10000
pvasintotico<-0
pvanalitica<-0
pvempirical<-0
pvempirica2<-0
pvsimulazione<-0

p<-p1+p2
media<-c(rep(0,p))
sigma<-diag(100,p,p)

dati<-mvrnorm(n=n,media,sigma,tol=1e-6,empirical=FALSE)

h<-p.e.trv72(n,p1,p2,alpha,errore)
a1<-h[1]
a2<-h[2]
c<-h[3]
}

```

```

gl<-p1*p2

soglia<-qchisq(1-alpha,gl)
sogliasimulazione<-h[4]

dati0<-mvrnorm(n=n*nr,media,sigma,tol=1e-6,empirical=FALSE)
h<-1
k<-n
y<-0

for(i in 1:nr)
{
  dat<-NULL
  dat<-dati0[h:k,]
  h<-h+n
  k<-k+n
  t<-trv72.n(dat,p1,p2)
  x<-trv72.Bart(dat,p1,p2)
  e1<-t/a1
  e2<-t/a2+c

  if(t>soglia)          pvasintotico<-pvasintotico+1
  if(x>soglia)          pvanalitica<-pvanalitica+1
  if(e1>soglia)         pvempirical<-pvempirical+1
  if(e2>soglia)         pvempirical2<-pvempirical2+1
  if(t>sogliasimulazione) pvsimulazione<-pvsimulazione+1
}

pvasintotico<-pvasintotico/nr
pvanalitica<-pvanalitica/nr
pvempirical<-pvempirical/nr
pvempirical2<-pvempirical2/nr
pvsimulazione<-pvsimulazione/nr

h<-c(gl,pvasintotico,pvanalitica,pvempirical,pvempirical2,pvsimulazione)
print(h)
}

#-----

trv72.n<-function(dati,p1,p2)
{
  n<-dim(dati)[1]
  I<-diag(1,p2,p2)

  S11<-var(dati[,1:p1])*(n-1)/n
  S22<-var(dati[, (p1+1):(p1+p2)])*(n-1)/n
  S12<-cov(dati[,1:p1],dati[, (p1+1):(p1+p2)])*(n-1)/n
  S21<-cov(dati[, (p1+1):(p1+p2)],dati[,1:p1])*(n-1)/n

  l<-solve(S22)%*%S21)%*%solve(S11)%*%S12

  test<--n*log(det(I-l))
}

#-----

trv72.Bart<-function(dati,p1,p2)
{
  # sigma=I, mu non nota

  n<-dim(dati)[1]

  test<-(1-(0.5/n*(p1+p2+3)))*trv72.n(dati,p1,p2)
}

#-----

p.e.trv72<-function(n,p1,p2,alpha,errore)
{

```

```

y1<-NULL
Sy<-0
v<-0
p<-p1+p2
media<-c(rep(0,p))
sigma<-diag(100,p,p)
gl<-p1*p2

N<-signif(((2.57/errore)^2*alpha*(1-alpha)),2)

dati0<-mvrnorm(n=n*N,media,sigma,tol=1e-6,empirical=FALSE)
h<-1
k<-n
y<-0

for(i in 1:N)
{
  dat<-NULL
  dat<-dati0[h:k,]
  h<-h+n
  k<-k+n
  x<-trv72.n(dat,p1,p2)
  y<-c(y,x)
}

Ey<-mean(y)
v<-var(y)
a1<-Ey/gl
a2<-sqrt(v/(2*gl))
c<-gl-Ey/a2

y<-sort(y)
percentile<-quantile(y,1-alpha)

h<-c(a1,a2,c,percentile)
}

#####

bontà7<-function(p,n,alpha=0.05,errore=5e-3)
{

nr<-10000
pvasintotico<-0
pvanalitica<-0
pvempirica<-0
pvempirica2<-0
pvsimulazione<-0

media<-c(rep(0,p))
sigma<-diag(100,p,p)

dati<-mvrnorm(n=n,media,sigma,tol=1e-6,empirical=FALSE)

h<-p.e.trv7(n,p,alpha,errore)
a1<-h[1]
a2<-h[2]
c<-h[3]
gl<-p*(p-1)/2

soglia<-qchisq(1-alpha,gl)
sogliasimulazione<-h[4]

dati0<-mvrnorm(n=n*nr,media,sigma,tol=1e-6,empirical=FALSE)
h<-1
k<-n

```

```

y<-0
for(i in 1:nr)
  {
    dat<-NULL
    dat<-dati0[h:k,]
    h<-h+n
    k<-k+n
    t<-trv7.n(dat)
    x<-trv7.Bart(dat)
    e1<-t/a1
    e2<-t/a2+c

    if(t>soglia)          pvasintotico<-pvasintotico+1
    if(x>soglia)          pvanalitica<-pvanalitica+1
    if(e1>soglia)        pvempirical<-pvempirical+1
    if(e2>soglia)        pvempirica2<-pvempirica2+1
    if(t>sogliasimulazione) pvsimulazione<-pvsimulazione+1
  }

pvasintotico<-pvasintotico/nr
pvanalitica<-pvanalitica/nr
pvempirical<-pvempirical/nr
pvempirica2<-pvempirica2/nr
pvsimulazione<-pvsimulazione/nr

h<-c(gl,pvasintotico,pvanalitica,pvempirical,pvempirica2,pvsimulazione)
print(h)
}

#-----

trv7.n<-function(dati)
{
n<-dim(dati)[1]
R<-cor(dati)
test<--n*log(det(R))
}

#-----

trv7.Bart<-function(dati)
{
# sigma=I, mu non nota

n<-dim(dati)[1]
p<-dim(dati)[2]

test<-(n-1/6*(2*p+11))/n*(trv7.n(dati))
}

#-----

p.e.trv7<-function(n,p,alpha,errore)
{

y1<-NULL
Sy<-0
v<-0
media<-c(rep(0,p))
sigma<-diag(100,p,p)
gl<-p*(p-1)/2

N<-signif(((2.57/errore)^2*alpha*(1-alpha)),2)

dati0<-mvrnorm(n=n*N,media,sigma,tol=1e-6,empirical=FALSE)
h<-1
k<-n
y<-0

```

```

for(i in 1:N)
  {
    dat<-NULL
    dat<-dati0[h:k,]
    h<-h+n
    k<-k+n
    x<-trv7.n(dat)
    y<-c(y,x)
  }

Ey<-mean(y)
v<-var(y)
a1<-Ey/g1
a2<-sqrt(v/(2*g1))
c<-g1-Ey/a2

y<-sort(y)
percentile<-quantile(y,1-alpha)

h<-c(a1,a2,c,percentile)
}

#####

bontà9<-function(g,p,n,alpha=0.05,errore=5e-3)
{

nr<-10000
pvasintotico<-0
pvanalitica<-0
pvempirical<-0
pvempirica2<-0
pvsimulazione<-0

nc<-n*g
num<-c(rep(nc/g,g))
media<-c(rep(0,p))
sigma<-diag(100,p,p)

dati<-mvrnorm(n=nc,media,sigma,tol=1e-6,empirical=FALSE)

h<-p.e.trv9(nc,p,num,alpha,errore)
a1<-h[1]
a2<-h[2]
c<-h[3]
g1<-p*(p+1)*(g-1)/2

soglia<-qchisq(1-alpha,g1)
sogliasimulazione<-h[4]

dati0<-mvrnorm(n=nc*nr,media,sigma,tol=1e-6,empirical=FALSE)
h<-1
k<-nc
y<-0

for(i in 1:nr)
  {
    dat<-NULL
    dat<-dati0[h:k,]
    h<-h+nc
    k<-k+nc
    t<-trv9.n(dat,num)
    x<-trv9.Bart(dat,num)
    e1<-t/a1
    e2<-t/a2+c

    if(t>soglia)          pvasintotico<-pvasintotico+1
  }
}

```

```

        if(x>soglia)           pvanalitica<-pvanalitica+1
        if(e1>soglia)         pvempirical<-pvempirical+1
        if(e2>soglia)         pvempirica2<-pvempirica2+1
        if(t>sogliasimulazione) pvsimulazione<-pvsimulazione+1
    }

pvasintotico<-pvasintotico/nr
pvanalitica<-pvanalitica/nr
pvempirical<-pvempirical/nr
pvempirica2<-pvempirica2/nr
pvsimulazione<-pvsimulazione/nr

h<-c(gl,pvasintotico,pvanalitica,pvempirical,pvempirica2,pvsimulazione)
print(h)
}

#-----

trv9.n<-function(dati,num)
{

n<-dim(dati)[1]
p<-dim(dati)[2]
g<-length(num)
test<-0
W<-matrix(rep(0,p^2),p,p)
ba<-1
for(i in 1:g)
    {
        ni<-num[i]
        sn<-var(dati[ba:(ni+ba-1),])*ni
        ba<-ba+ni
        test<-test+ni*log(det(sn/ni))
        W<-W+sn
    }

test<-n*log(det(W/n))-test
}

#-----

trv9.Bart<-function(dati,num)
{
# sigma1=...=sigmag

r<-0
n<-dim(dati)[1]
p<-dim(dati)[2]
g<-length(num)
test<-0
W<-matrix(rep(0,p^2),p,p)
ba<-1
for(i in 1:g)
    {
        ni<-num[i]
        sn<-var(dati[ba:(ni+ba-1),])
        ba<-ba+ni
        test<-test+(ni-1)*log(det(sn))
        W<-W+sn*(ni-1)
        r<-r+1/(ni-1)
    }
r<-r-1/(n-g)
su<-W/(n-g)
testB<-(n-g)*log(det(su))-test

f<-1-((2*p^2+3*p-1)/(6*(p+1)*(g-1)))*r

test<-f*testB
}

```

```

#-----
p.e.trv9<-function(nc,p,num,alpha,errore)
{
y1<-NULL
Sy<-0
v<-0
media<-c(rep(0,p))
sigma<-diag(100,p,p)
g<-length(num)
gl<-p*(p+1)*(g-1)/2

N<-signif(((2.57/errore)^2*alpha*(1-alpha)),2)

dati0<-mvrnorm(n=nc*N,media,sigma,tol=1e-6,empirical=FALSE)
h<-1
k<-nc
y<-0

for(i in 1:N)
{
dat<-NULL
dat<-dati0[h:k,]
h<-h+nc
k<-k+nc
x<-trv9.n(dat,num)
y<-c(y,x)
}

Ey<-mean(y)
v<-var(y)
a1<-Ey/gl
a2<-sqrt(v/(2*gl))
c<-gl-Ey/a2

y<-sort(y)
percentile<-quantile(y,1-alpha)

h<-c(a1,a2,c,percentile)
}

```

## BIBLIOGRAFIA

- Anderson T. W. e Darling D. A. (1952). Asymptotic theory of certain “goodness of fit” criteria based on stochastic processes. *Annals of Mathematical Statistics*, **23**, 193-212.
- Anderson T. W. (1962). On the two-sample Cramèr-Von Mises criterion. *Annals of Mathematical Statistics*, **33**, 1148-1159.
- Azzalini, A. (2000). *Inferenza Statistica. Un'introduzione basata sul concetto di verosimiglianza*. Springer-Verlag Italia, II edition.
- Bartlett, M. S. (1936). The information available in small samples. *Proc. Comb. Phil. Soc.*, **32**, 560-566.
- Bartlett, M. S. (1936). Statistical information and properties of sufficiency. *Proc. Roy. Soc. A.*, **154**, 124-137.
- Bartlett, M. S. (1937). Properties of sufficiency and statistical tests. *Proceeding Royal Society, Series A*, **160**, 268-282.
- Bartlett, M. S. (1953). Approximate confidence intervals. *Biometrika*, **40**, 12-19.
- Bortot P., Salvan A. e Ventura L. (2000). *Inferenza Statistica: Applicazioni con S-Plus e R*. Cedam, Padova.
- Chiodi Marcello (2000). *Tecniche di simulazione in statistica*. RCE, Napoli.
- Cribari-Neto, F. and Cordeiro, G. M. (1996). On Bartlett-type corrections. *Econometric Reviews*, **15**, 339-367.
- DiCiccio T., Peter Hall and Joseph Romano (1991). Empirical Likelihood is Bartlett-correctable. *The annals of Statistics*, **19**, 1053-1061.

- Iacus S. M. e Masarotto G. (2003). *Laboratorio di statistica con R*. McGraw-Hill, Milano.
- Jesen, J. L. (1993). A historical sketch and some new results on the improved Likelihood statistic. *Scandinavian Journal of Statistic*, **20**, 1-15.
- Korin, B. P. (1968). On the distribution of a statistic used for testing a covariance matrix. *Biometrika*, **55**, 171-178
- Landenna G., Marasini D. e Ferrari P. (1998). *La verifica di ipotesi statistiche*. Il Mulino, Bologna.
- Lawley, D. N. (1956). A general method for approximating to the distribution of the likelihood ratio criteria. *Biometrika*, **43**, 295-303.
- Mardia, K. V., Kent J. T. e Bibby J. M. (1979). *Multivariate Analysis*. Academic Press.
- Nielsen B. (1997). Bartlett correction of the unit root test in autoregressive models. *Biometrika*, **84**, 500-504.
- Olkin, I. and Tomsy, T. L. (1975). A new class of multivariate tests based on the union-intersection principle. *Bull. Inst. Statist. Inst.*, **46**, 202-204
- Pace L. e Salvan A. (2001). *Introduzione alla statistica*. Cedam, Padova.
- Piccolo, D. (1998). *Statistica*. Il Mulino, Bologna.
- Rizzo Maria L. and Gábor J. Székely (2005). A new test for multivariate normality. *Journal of Multivariate Analysis*, **93**, 58-80.
- Wichurat Micheal J. (1988) Algorithm AS 241. The Percentage Points of the Normal Distribution. *Applied Statistics*, **37**, 477–484.

