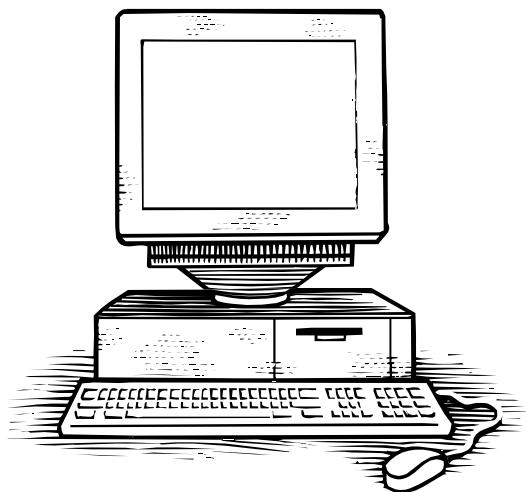


STATISTICA COMPUTAZIONALE II

Prof. T. Gastaldi

a.a.1997/98

ANALISI DELLE CORRISPONDENZE IN IML



Crea Davide
Lastrucci Gherardo
Stabellini Alessandro

Indice

	<i>Pag.</i>
COMANDI SAS/IML	1
<i>apertura del SAS</i>	<i>1</i>
<i>chiusura del SAS</i>	<i>1</i>
<i>lista comand</i>	<i>1</i>
 ANALISI DELLE CORRISPONDENZE	 32
<i>analisi dei profili riga</i>	<i>34</i>
<i>calcolo della distanza tra due punti</i>	<i>35</i>
<i>uso delle matrici per il cambiamento di scala</i>	<i>36</i>
<i>analisi in componenti principali</i>	<i>36</i>
<i>analisi dei profili colonna</i>	<i>37</i>
<i>analisi in componenti principali</i>	<i>38</i>
<i>fedeltà della rappresentazione</i>	<i>38</i>
<i>contributo di un profilo alla varianza di un asse principale</i>	<i>39</i>
<i>contributo di un asse principale all'inerzia di un profilo</i>	<i>40</i>
PROG. IN IML PER L'ANALISI DELLE CORRISPONDENZE	41
MATRICE DEI DATI	47
OUTPUT PROGRAMMA IML	48
PROG. IN SAS PER L'ANALISI DELLE CORRISPONDENZE	53
OUTPUT PROGRAMMA SAS	53

COMANDI SAS/IML

APERTURA DEL SAS

```
c:\ CD SAS↵
c:\SAS> SAS↵
```

CHIUSURA DEL SAS

```
command⇒ BYE
c:\SAS>CD..
c:\
```

LISTA COMANDI

ABORT	istruzione IML per fermare l'esecuzione: In IML causa la chiusura del file e l'uscita.
ABS	funzione scalare che calcola il valore assoluto di tutti gli elementi di una data matrice.
APPEND	legge le righe di una data matrice come schede da inserire nel data base.
ARCCOS	funzione scalare che calcola l'arcocoseno trigonometrico di ogni elemento di una data matrice. Il risultato viene dato in radianti.
ARCSIN	funzione scalare che calcola l'arcoseno trigonometrico di ogni elemento di una data matrice. Il risultato viene dato in radianti.
ATAN	funzione scalare che calcola l'arcotangente trigonometrico di ogni elemento di una data matrice. Il risultato viene dato in radianti.

BLOCK	funzione che crea, a partire da n matrici date, una nuova matrice a blocchi in cui le matrici di partenza vengono sistematate diagonalmente; le matrici da combinare possono essere al massimo 15. Date, ad esempio, le matrici A, B e C il comando:
	BLOCK(A,B,C); mi restituisce:
	$\begin{vmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{vmatrix}$
BELL	porzione del comando DISPLAY che serve a produrre un breve segnale sonoro.
BETAINV	funzione che determina i quantili di una BETA (a,b). La corretta sintassi è: BETAINV(probabilità,a,b);
BYTE	funzione che restituisce una matrice avente per elementi i caratteri corrispondenti alla cifra indicata. Scriviamo ad esempio:
	A=BETA(47); mi restituisce: A="/";
CALL CHANGE	cambia il testo di un'array. Ad esempio se scrivo: A="ieri era una bella giornata."; CALL CHANGE(A,"bella","brutta"); ottengo: A="ieri ere una brutta giornata."
CALL GDRAW	chiamata a routine grafica che permette di disegnare dei sistemi di linee.
CALL GDRAWL	chiamata a routine grafica che permette di creare sistemi di linee.
CALL GGRID	chiamata a routine grafica che permette di creare sistemi di griglie.

CALL GPIE	chiamata a routine grafica che permette di disegnare archi di cerchio o settori circolari.
CALL GPIEXY	chiamata a routine grafica che permette la conversione da coordinate polari a coordinate cartesiane.
CALL GPOINT	chiamata a routine grafica che permette di disegnare dei punti.
CALL GPOLY	chiamata a routine grafica che permette di disegnare delle poligonali (aperte o chiuse) e aree.
CALL GPORT	chiamata a routine grafica che permette di definire una viewport. La sintassi corretta è: CALL GPORT(vettore_finestra);
CALL GSHOW	chiamata a routine grafica che visualizza un grafico: La sintassi corretta è: CALL GSHOW (nome_segmento_grafico);
CALL GSTART	chiamata a routine grafica che attiva il sistema grafico. E' sempre seguita da CALL GSTOP. La sintassi corretta è: CALL GSTART (<catalogo>,<flag_rimpiazzo>);
CALL GSTOP	chiamata a routine grafica che disattiva il sistema grafico. E' sempre preceduta da CALL GSTART.
CALL GTEXT	chiamata a routine grafica che permette l'immissione di linee (orizzontali o verticali) di testo nel grafico.
CALL GXAXIS	chiamata a routine grafica che insieme a CALL GYAXIS serve per tracciare assi (o porzioni di assi) di un sistema di coordinate cartesiane.
CALL YAXIS	chiamata a routine grafica che insieme a CALL GXAXIS serve per tracciare assi (o porzioni di assi) di un sistema di coordinate cartesiane.
CALL GVTEXT	chiamata a routine grafica che permette l'immissione di linee (orizzontali o verticali) di testo nel grafico.

CALL GWINDOWS	chiamata a routine grafica che sovrappone allo schermo del monitor un sistema di coordinate definito dall'utente. Ciò svincola il programmatore dalla preoccupazione di dover trasformare linearmente i dati per ridurli alle coordinate fisiche dello schermo, le quali dipendono dal modo grafico corrente. La sintassi corretta è: CALL GWINDOWS (vattore_finestra);
CARDS	comando che precede l'immissione di una matrice di dati all'interno del programma. Dopo l'inserimento della matrice va indicato un “punto e virgola” che chiude l'istruzione CARDS.
CEIL	funzione scalare che arrotonda gli elementi della matrice in argomento al più piccolo intero maggiore o uguale all'argomento. Se x è un intero o un negativo INT(x) e CEIL(x) coincidono.
CHAR	questa funzione prende una matrice numerica di dimensione (n*m) e ne produce una della stessa dimensione dove sostituisce, ad ogni elemento numerico, il corrispondente elemento alfabetico. La sintassi corretta è: CHAR(matrice<,w,d>); CHAR(matrice<,w>); CHAR(matrice); Tale funzione, come appena indicato, può avere uno, due o tre argomenti; il primo è il nome della matrice numerica e deve essere sempre indicato; il secondo (cioè w) può non essere specificato ed indica l'ampiezza del risultato; il terzo argomento (cioè d) come w non deve essere necessariamente indicato e rappresenta il numero di decimali che vogliamo siano indicati nel risultato.
CHOOSE	istruzione che può sostituire la struttura IF... THEN, ELSE. La sintassi corretta è: risult=CHOOSE(espressione,risposta_vera,risposta_falsa);
CLOSE	si utilizza per chiudere uno o più data set del SAS (aperti

con i comandi USE, EDIT, CREATE). La sintassi corretta è:
CLOSE SASdataset(s);

CLOSEFILE

si utilizza per chiudere dei files aperti con le istruzioni FILE o INFILE. Per vedere quali files sono stati precedentemente aperti si può usare il comando SHOW FILES.

CMNDLINE

opzione del comando DISPLAY.

COLNAME

funzione che associa etichette alle colonne di una data matrice; ad esempio:

COLNAME=nome_colonna;

COMPRESS

funzione che rimuove caratteri di una data matrice; ad esempio:

mat_stringhe=COMPRESS(mat_stringhe, caratteri_rimasti);

CONCAT

funzione che produce una matrice data dalla concatenazione dei corrispondenti elementi stringa di date matrici di partenza. Se gli elemnti non sono scalari, questi vanno conformati. Vediamo un piccolo esempio:

B={"AB" "C",
"DE" "FG"};

C={"H" "IJ",
"K" "LM"};

A=CONCAT(B,C);

otteniamo:

| "ABH" "CIJ"
| "DEK" FGLM |

CONTENTS

funzione che restituisce una matrice di caratteri contenente i nomi delle variabili per i data set del SAS specificati nella libname e nella memname. L'argomento della libname è un carattere scalare contenente il nome della libreria, e memname è un carattere scalare contenente il nome dei membri della libreria. Il risultato è una matrice di caratteri con n righe, una colonna ed 8 caratteri per elemento, dove n è il numero delle variabili nel data set. La lista delle variabili è restituita in ordine alfabetico. Se viene restituito

un unico argomento, IML usa questo come una memname. Se nessun argomento è specificato, viene utilizzato il data set correntemente aperto.

COS funzione scalare che calcola il coseno trigonometrico di ogni elemento dall'argomento. Il risultato è in radianti.

COSH funzione scalare che calcola il coseno iperbolico trigonometrico di ogni elemento dall'argomento. Il risultato è in radianti.

CREATE `CREATE SASdataset <VAR operatore>;
CREATE SASdataset FROM nome <[COLNAME=nome
ROWNAME= nome]>;`
Serve per creare un nuove data set del SAS. Se il data set è già aperto, viene creato un data set nell'output corrente. Le variabili del nuovo data set SAS sono anche le variabili elencate nell'operatore VAR o variabili create dalle colonne della matrice FROM.

La matrice COLNAME da altri nomi alle variabili create nel data set. L'opzione COLNAME specifica il nome della matrice di caratteri.

L'operando ROWNAME aggiunge una variabile al data set che contiene i titoli delle righe. L'operando deve essere una matrice di caratteri esistente e avente valore.

Vediamo un esempio:

```
NAME="123456789012";  
SEX="M";  
CREATE SAVA:CLASS VAR {NAME SEX AGE  
HEIGHT WEIGHT};
```

CSHAPE funzione che crea una matrice di caratteri. E' uguale alla funzione SHAPE solo che questa ricrea a livello di caratteri. Le dimensioni della matrice creata con questa funzione sono specificate con NROW(numero righe), con NCOL(numero colonne) e con SIZE(lunghezza elementi). Vediamo un esempio:

`R=CSHAPE('abcd',2,2,1);`
ottengouna matrice R del tipo:

$$\begin{vmatrix} & a & b \end{vmatrix}$$

| c d |

CUSUM

funzione che restituisce una matrice della stessa struttura dell'argomento che contiene le somme cumulate ottenute scansionando gli argomenti, sommando in ordine di riga crescente.

A=CUSUM ({1 2 4 5});
 B=CUSUM ({5 6 , 3 4});

si ottiene una matrice A del tipo:

[1 3 7 12]

ed una matrice B del tipo:

$$\begin{vmatrix} 5 & 11 \\ 14 & 18 \end{vmatrix}$$

DATASETS

DATASETS(nome_libreria);
 DATASETS();

funzione che restituisce una matrice di caratteri contenenti i nomi dei data sets del SAS nella libreria specificata. Nome_libreria è un carattere scalare contenente il nome della libreria. Il risultato è una matrice di caratteri con n righe ed 1 colonna, dove n è il numero dei data set nella libreria. Se nessun argomento è specificato, l'IML, usa per default nome_libreria.

DELETE

PER I RECORD: comando che contrassegna dei record che verranno cancellati nell'output corrente del data set. La sintassi è:

DELETE range where;
 dove range e where non devono necessariamente essere specificati.

PER LE SUBROUTINE: comando che cancella un data set del SAS in una determinata libreria.

DESIGN

DESIGN(vettore_colonna);
 funzione che crea una matrice full-design dal vettore_colonna. Ogni singolo valore del vettore_colonna genera una colonna della matrice. La matrice contiene il valore 1 nella posizione ij (dove i è la posizione del valore all'interno del vettore e j è il numero indicato nel vettore alla

i-ma posizione) e 0 altrove. Ad esempio:

$$A=\{1, 1, 2, 2, 3, 1\};$$

si ottiene:

$$\begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{vmatrix}$$

DET

funzione che calcola il determinante di una data matrice quadrata.

DIAG

funzione che crea una matrice avente sulla diagonale principale, elementi uguali quelli della corrispondente matrice di partenza. Ad esempio:

$$\begin{aligned} A &= \{4, 3, 2, 1\}; \\ B &= \{1, 2, 3\}; \\ C &= \text{DIAG}(A); \\ D &= \text{DIAG}(B); \end{aligned}$$

ottengo una matrice C del tipo:

$$\begin{vmatrix} 4 & 0 \\ 0 & 1 \end{vmatrix}$$

ed una matrice D del tipo:

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{vmatrix}$$

DIGAMMA

funzione che calcola la derivata del logaritmo della funzione GAMMA di ciascun valore della matrice. La sintassi è:

DGAMMA (matrice);

DISPLAY

comando che serve per visualizzare ed attivare una finestra già definita.

DO

comando che fa parte di una struttura iterativa; indica che le istruzioni comprese fra il DO stesso ed il successivo comando END dovranno essere eseguite in gruppo. Usualmente questo comando appare con con il ciclo IF...

THEN, ELSE dove introduce il gruppo di comandi da eseguire quando la condizione IF è vera o falsa.

DO UNTIL

l'espressione UNTIL viene indicata all'inizio di un ciclo; i comandi che seguono tale espressione vengono eseguiti finché l'espressione indicata produce uno zero o un valore mancante; la sintassi è:

DO UNTIL (espressione);

DO WHILE

l'espressione WHILE viene indicata all'inizio di un ciclo; i comandi che seguono tale espressione vengono eseguiti finché l'espressione indicata produce uno nonzero o un valore nonmancante; la sintassi è:

DO WHILE (espressione);

EDIT

comando che apre un data set del SAS per permetterne la lettura. Se il data set è già aperto, il comando marca l'input corrente e l'output del data set.

EIGEN

crea un vettore contenente gli autovalori, ordinati in modo decrescente, di una matrice simmetrica. Crea inoltre una matrice di vettori colonna ortonormalizzati della matrice simmetrica corretti in modo da rendere le metrici corrispondenti

EIGVAL

comando che crea un vettore colonna contenente gli autovalori di una matrice simmetrica. Gli autovalori sono ordinati in modo decrescente.

EIGVEC

comando che crea una matrice contenente gli autovettori di una matrice data.

ELSE

istruzione che fa parte del ciclo IF... THEN, ELSE. E' opzionale e contiene una o un blocco di istruzioni che vengono eseguite se l'espressione risulta falsa.

EXP

funzione scalare che determina l'esponenziale di ogni elemento di una data matrice.

FILE

si utilizza per aprire un file per l'output, o se il file è già

aperto, per evidenziare l'output corrente del file così che il seguente comando PUT lo scriva.

FIND	serve per cercare osservazioni, in un dato range, che soddisfano le condizioni della clausola WHERE.
FINISH	serve per indicare la fine di un modulo.
FINV	funzione che determina i quantili dalla F di Fisher. La sintassi è: FINV(probabilità, gl1, gl2, <non_centralità>);
FLOOR	fa parte delle funzioni di troncamento ed arrotondamento; restituisce il più grande intero minore o uguale all'argomento.
GAMINV	funzione statistica che determina i quantili della distribuzione Gamma. La sintassi corretta è: GAMINV(probabilità, parametro_di_forma);
GAMMA	funzione che calcola la funzione Gamma di ciascun valore della matrice indicata nell'argomento, dove: $\text{Gamma}(x) \equiv \Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$
GOTO	comando che indica all'IML di saltare immediatamente all'istruzione con una data label ed indica allo stesso di eseguire le istruzioni che seguono.
GROUP	opzione dell'istruzione WINDOWS.
HALF	questa funzione è stata sostituita dalla funzione ROOT.
HDIR	questa funzione crea una matrice che ha sulle righe il prodotto, riga per riga, di tutti gli elementi di una matrice per quelli di un'altra matrice. Vediamo un banale esempio: A={1 2, 2 4, 3 6}; B={0 2, 1 1, 0 -1}; C=HDIR(A,B); ottengo una matrice C del tipo:

$$\begin{vmatrix} 0 & 2 & 0 & 4 \\ 2 & 2 & 4 & 4 \\ 0 & -3 & 0 & -6 \end{vmatrix}$$

I

funzione che crea una matrice identità della dimensione indicata. La dimensione indicata deve chiaramente essere un numero intero maggiore od uguale ad 1. Ad esempio:

$$A=I(3)$$

mi crea la matrice A così composta:

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

IF

istruzione che fa parte del blocco IF... THEN, ELSE. Contiene un'espressione che deve essere valutata ed è sempre seguito dall'espressione chiave THEN, mentre l'espressione ELSE non deve essere necessariamente indicata.

INDEX

si può usare questa espressione per creare un indice per le variabili nominate nel corrente input SAS data set. Viene creato un indice per ogniper ogni variabile indicata. Se il data set è molto grande, l'esecuzione di questa espressione richiede molto tempo.

INFILE

istruzione che si può utilizzare per aprire un data set. Tale istruzione può essere seguita da diverse opzioni:

RECFM=U: specifica che il file può essere letto in numerazione binaria piuttosto che come file di record con caratteri separatori.

LENGTH=L: specifica la variabile dove l'estensione del record sarà indicata e dove verrà letta dall'IML.

LRECL=operatore: specifica la misura del bufferche contiene i records. La misura di default è 512 ed è sufficiente per la maggior parte delle applicazioni.

Le seguenti opzioni controllano come l'IML agisce quando una istruzione cerca di leggere dopo la fine di un record. Il default è STOPOVER.

MISSOVER: cerca di leggere dopo la fine di un record con

l'assegnazione di dati mancanti alle variabili lette dopo la fine del record.

FLOWOVER: permette all'input di andare nel record successivo per ottenere valori per le variabili.

STOPOVER: va alla fine del record come una condizione errore.

INPUT

istruzione che consente di aprire record dal corrente file di entrata, inserendo i valori nelle variabili IML.

INSERT

INSERT(x,y,riga);
INSERT(x,y,riga,colonna);

funzione che restituisce il risultato dell'inserimento della matrice y all'interno della matrice x nel posto specificato da riga e colonna indicati nell'argomento. Ad esempio:

A={1 2, 3 4};
B={5 6, 7 8};
C=INSERT(A,B,2,0);

mi da una matrice C del tipo:

$$\begin{array}{|c|} \hline 1 & 2 \\ \hline 5 & 6 \\ \hline 7 & 8 \\ \hline 3 & 4 \\ \hline \end{array}$$

INT

funzione scalare che prende il valore intero di ogni elemento della matrice indicata nell'argomento.

INV

funzione che produce una matrice che è l'inverso della matrice indicata nell'argomento; la matrice indicata nell'argomento deve essere naturalmente quadrata e non singolare (cioè deve ammettere l'inversa).

J

funzione che crea una matrice con un numero di righe e di colonne pari alle cifre indicate nell'argomento, i cui elementi sono tutti uguali al valore indicato nell'argomento stesso. Se il numero delle colonne non è specificato, viene preso, per default, uguale al numero delle righe. Se il valore non è specificato, viene preso, per default, il numero 1.

Ad esempio:

A=J(2, 3, 8);
mi da una matrice a del tipo:

$$\begin{vmatrix} 8 & 8 & 8 \\ 8 & 8 & 8 \end{vmatrix}$$

LENGTH

funzione che prende una matrice di caratteri come argomento e produce una matrice numerica come risultato. La matrice risultato ha la stessa dimensione della matrice argomento e contiene i valori corrispondenti alle stringhe della matrice di partenza. Se una stringa è bianca verrà indicato il valore 1.

LINK

come l'istruzione GOTO dice all'IML di saltare all'istruzione con la label indicata. Diversamente da GOTO, l'IML ricorda da dove è partito il salto, e ritorna in quel punto con l'istruzione RETURN.

LIST

comando che si utilizza per mostrare dati. Tutti i valori data indicati in VAR figurano su una sola riga, i valori sono elencati in colonna seguiti dal nome delle variabili. Ogni record figura su una singola riga. Se i valori data non entrano in una singola riga, i valori di ogni record sono raggruppati in paragrafi. Ogni elemento nel paragrafo ha la forma: nome=valore.

Per esempio:

LIST ALL;	lista di tutti i data set.
LIST;	lista delle correnti osservazioni.
LIST VAR {NOME IINDIRIZZO};	lista delle variabili specificate.
LIST ALL WHERE (ANNI>30);	lista delle osservazioni dove si verifica la condizione indicata.
LIST NEXT;	lista della seguente osservazione.
LIST POINT 24;	lista l'osservazione 24.
LIST POINT (10:15);	lista le osservazioni 10,11,..., 15.

LOAD

comando che carica matrici di valori dal disco libreria sulla corrente area di lavoro. Se non viene specificato alcun nome, l'IML carica tutte le matrici presenti nel disco

libreria.

LOC

funzione che crea un vettore riga di dimensione $n*1$ dove n è il numero degli elementi non zero presenti nell'argomento. I valori mancanti vengono considerati come lo 0. I valori che troviamo nel risultante vettore riga rappresentano le posizioni dei valori non zero dell'argomento. Ad esempio:

$A=\{1 \ 0 \ 2 \ 3 \ 0\};$

$B=LOC(A);$

si ottiene una matrice B del tipo:

$$\begin{bmatrix} 1 & 3 & 4 \end{bmatrix}$$

LOG

funzione scalare che calcola il logaritmo naturale di ogni elementi della matrica in argomento.

LOG2

funzione scalare che calcola il logaritmo in base 2 di ogni elementi della matrica in argomento.

MATTRIB

istruzione che associa attributi a matrici. Ogni matrice può essere associata ad una matrice di nomi_riga e ad una matrice di nomi_colonna che vengono utilizzate ogni volta che vengono indicati i nomi di riga e di colonna, rispettivamente. Ad esempio l'istruzione:

**MATTRIB A ROWNAME=RA COLNAME=CA B
COLNAME=CB;**

da il nome_colonna ed il nome_riga alla matrice A ed il solo nome_colonna alla matrice B;

MAX

funzione che produce un singolo numero (o un singolo carattere stringa, a seconda del tipo della matrice iniziale) che è il più grande elemento tra tutti quelli presenti nella matrice di partenza. La funzione fa una ricerca dei dati mancanti, all'interno della matrice, e li esclude dal risultato. Se tutti gli argomenti sono valori mancanti, il massimo negativo, dell'elaboratore, rappresentabile è il risultato. Se vogliamo determinare i massimi elementi di due corrispondenti matrici, usiamo l'operatore $<>$. L'estensione del carattere stringa è il più lungo elemento fra gli argomenti della matrice.

MIN	funzione che produce un singolo numero (o un singolo carattere stringa, a seconda del tipo della matrice iniziale) che è il più piccolo elemento tra tutti quelli presenti nella matrice di partenza. La funzione fa una ricerca dei dati mancanti, all'interno della matrice, e li esclude dal risultato. Se tutti gli argomenti sono valori mancanti, il più largo numero rappresentabile è il risultato. Se vogliamo determinare i minimi elementi di due corrispondenti matrici, usiamo l'operatore $><$. L'estensione del carattere stringa è il più lungo elemento fra gli argomenti della matrice.
MOD	funzione scalare che restituisce il resto dalla divisione tra il primo ed il secondo elemento indicati nell'argomento. La sintassi esatta è: MOD(valore, divisore);
MSGLINE	opzione del comando WINDOWS.
NAME	istruzione che restituisce il nome degli argomenti in un vettore colonna. Ad esempio: N=NAME(A, B, C); N viene restituito come una matrice di 3 righe ed 1 colonna i cui elementi lunghi 8 caratteri contengono i valori corrispondenti ad A, B e C. NAME ha un ulteriore modo d'uso. Possiamo usare NAME quando nelle macro vogliamo utilizzare un unico argomento per valore e nome.
NCOL	funzione che restituisce un singolo numero che è il numero delle colonne dalla matrice indicata nell'argomento. Se la matrice non ha valori. La funzione NCOL restituisce il valore 0.
NLENG	funzione che restituisce un singolo valore numerico che rappresenta la lunghezza in byte di ogni singolo valore della matrice in argomento. Tutti gli elementi della matrice hanno la stessa lunghezza. Se la matrice in argomento non contiene valori, la funzione restituisce il numero 0.
NORMAL	funzione scalare che restituisce un numero pseudo-casuale

	con distribuzione normale $N(0,1)$, avente cioè media nulla e deviazione standard unitaria. Questa funzione restituisce una matrice della stessa dimensione della matrice di partenza. Il primo argomento nella prima chiamata viene usato come “seme”; se questo è nullo viene utilizzato, come “seme”, l’orario dell’elaboratore.
NROW	funzione che restituisce un singolo numero che è il numero delle righe dalla matrice indicata nell’argomento. Se la matrice non ha valori. La funzione NROW restituisce il valore 0.
NUM	funzione che ha come argomento una matrice di caratteri i cui elementi sono caratteri numerici e crea una matrice numerica della stessa dimensione della matrice di partenza i cui elementi sono delle rappresentazioni numeriche (in doppia precisione) dei corrispondenti elementi della matrice in argomento.
OPSCAL	<p>il risultato di questa funzione è l’ottimale funzione scalare dei dati qualitativi (nominali o ordinali) specificati nell’argomento (QUANTI):</p> <p style="text-align: center;">OPSCAL(MLEVEL,QUANTI); OPSCAL(MLEVEL,QUANTI,QUALI);</p> <p>L’ottimale trasformazione scalare risulta essere:</p> <ul style="list-style-type: none"> -il più piccolo quadratodei dati quantitativi in QUANTI; -preserva le misurazioni qualitative dei livelli di QUALIT. <p>OPSCAL è una funzione o chiamata. Quando è utilizzato come chiamata, il primo argomento della chiamata è la matrice che contiene i risultati restituiti. Si possono utilizzare i seguenti argomenti:</p> <p><u>MLEVEL</u>: specifica uno scalare che può avere solo due valori. Quando MLEVEL=1 la matrice QUALIT è al livello nominale di misurazione;</p> <p><u>QUANTI</u>: specifica una matrice $m*n$ di informazioni quantitative assunte per essere gli intervalli dei livelli di misurazione;</p> <p><u>QUALIT</u>: specifica una matrice $m*n$ di informazioni qualitative il cui livello di misurazione è specificato con MLEVEL. Quando QUALIT viene omesso, MLEVEL è uguale a 2. E’ importante sapere che QUALIT non può</p>

essere specificato come una matrice di caratteri. Vediamo un banale esempio:

```
QUANTI=( 5 4 6 7 4 6 2 4 8 6);  
QUALIT=( 6 6 2 12 4 10 4 10 8 6);  
OS=OPSCAL(1,QUANTI,QUALI);
```

il risultato è dato da una matrice OS del tipo:

```
[5 5 6 7 3 5 3 5 8 5]
```

ORPOL

funzione che genera una matrice ortonormale polinomiale. La sintassi corretta è:

```
ORPOL(vettore,grado_massimo,peso);
```

Se il grado massimo viene omesso, l'IML prende come default il valore:

massimo_grado(massimo grado polinomiale da costruire)
= min(n,20).

Se il peso è specificato, il massimo_grado deve assolutamente essere indicato.

PAUSE

comando che ferma l'esecuzione di un modulo, salva la catena delle chiamate così che l'esecuzione possa riprendere in seguito, scrive un messaggio di pausa che possiamo specificare, ind indica immediatamente il modo in cui si può riprendere l'esecuzione.

PRINT

comando che stampa le matrici specificate. Le opzioni che possono seguire questa istruzione sono:

FORMAT: specifica il formato da utilizzare nella stampa dei valori della matrice. Ad esempio:

```
PRINT X[FORMAT=5.3];
```

ROWNAME: specifica il nome di una matrice di caratteri che dovrà essere utilizzata come prima riga della matrice da stampare per indicare le intestazioni di colonna.

COLNAME: specifica il nome di una matrice di caratteri che dovrà essere utilizzata come prima colonna della matrice da stampare per indicare le intestazioni di riga.

Le tre opzioni possono essere utilizzate contemporaneamente; ad esempio:

```
PRINT X[ROWNAME=R COLNAME=C FORMAT=12.2]
```

PROBBETA

funzione statistica che genera il valore di ripartizione della

	funzione Beta. La sintassi corretta è: PROBBETA(matrice,parametro_a,parametro_b);
PROBBNML	funzione statistica che genera il valore di ripartizione della funzione binomiale. La sintassi corretta è: PROBBNML(probabilità,numerosità,matrice);
PROBCHI	funzione scalare che restituisce per ogni elemento dell'argomento la probabilità che una variabile casuale con una distribuzione chi-quadro cada tra i valori dati. Il secondo argomento specificato indica i gradi di libertà. Un terzo è argomento è opzionale e indica un parametro non centrale. La sintassi è: PROBCHI(matrice,gl,pn);
PROBF	funzione scalare che restituisce per ogni elemento della matrice specificata nell'argomento la probabilità che una variabile casuale con una distribuzione F cada tra i valori dati. I valori specificati in argomento rappresentano: la matrice, i gradi di libertà rispettivamente del numeratore e del denominatore. Può essere indicato un ulteriore argomento indicante un parametro non centrale.
PROBIT	funzione scalare che restituisce per ogni elemento della matrice specificata nell'argomento un valore simile ad una variabile casuale con una distribuzione normale che cade tra i valori dati.
PROBGAM	funzione statistica che genera il valore di ripartizione della funzione Gamma. La sintassi corretta è: PROBGAM (matrice,parametro_di_forma);
PROBHYP	funzione statistica che genera il valore di ripartizione della funzione ipergeometrica. La sintassi corretta è: PROBHYP(ampiezza_campione,margine_n,margine_k, rapporto);
PROBNORM	funzione scalare che restituisce per ogni elemento della matrice specificata nell'argomento la probabilità che una variabile casuale con una distribuzione normale cada tra i

	valori dati.
PROBNEG B	funzione statistica che genera il valore di ripartizione della funzione binomiale negativa. La sintassi corretta è: PROBNEG B(probabilità,numerosità,matrice);
PROBNORM	funzione statistica che genera il valore di ripartizione della funzione normale standardizzata. La sintassi corretta è: PROBNORM (matrice);
PROBT	funzione scalare che restituisce per ogni elemento della matrice specificata nell'argomento la probabilità che una variabile casuale con una distribuzione T cada tra i valori dati. Posiamo specificare i gradi di libertà ed un parametro non centrale.
PROC IML	comando di avvio dell'interprete IML all'interno del SAS.
PURGE	processo data utilizzato per cancellare record precedentemente marcati per la cancellazione. Questa operazione effettivamente rinumerà i record (osservazioni) del data set, chiudendo il comando creato per i record cancellati.
PUT	comando che scrive sul file specificato nel precedente eseguito, mettendomi velori delle variabili IML. Il comando PUT è una sequenza di posizioni e record direttivi, variabili e formati.
QUIT	comando che serve per uscire dall'IML. Il QUIT viene eseguito appena visto; non possiamo usare il quit come una istruzione eseguibile. E' parte di un modulo o clausola condizionale.
RANBIN	funzione statistica generatrice di numeri pseudo-casuali (Binomiale). La sintassi corretta è: RANBIN(seme,numerosità,probabilità);
RANCAU	funzione statistica generatrice di numeri pseudo-casuali (Cauchy). La sintassi corretta è: RANCAU(seme);

RANEXP	funzione statistica generatrice di numeri pseudo-casuali (Esponenziale). La sintassi corretta è: RANEXP(seme);
RANGAM	funzione statistica generatrice di numeri pseudo-casuali (Gamma). La sintassi corretta è: RANGAM(seme,parametro_di_forma);
RANGE	funzione che opera sugli elementi corrispondenti di una lista di matrici specificata come argomento, determinando il campo di variazione.
RANK	funzione che crea una matrice i cui elementi sono i ranghi della corrispondente matrice specificata in argomento. Ad esempio: X={2 2 1 0 5}; RANK(X); mi restituisce: [3 4 2 1 5] N.B. Vedi la funzione RANKTIE.
RANKTIE	funzione che crea una matrice i cui elementi sono i ranghi della corrispondente matrice specificata in argomento. RANKTIE differisce da RANK perché prende i ranghi dei valori vincolati; mentre RANK toglie arbitrariamente i vincoli. Infatti: X={2 2 1 0 5}; RANK(x); mi restituisce: [3.5 3.5 2 1 5]
READ	comando che serve per leggere variabili o record dal corrente data set del SAS. Vediamo dei banali esempi: READ ALL VAR{X,Y}; legge tutte le osservazioni per le due variabili. READ POINT 23; legge tutte le variabili per l'osservazione 23. READ ALL VAR{NOME,INDIR} WHERE{STATO=NY}; legge tutte le variabili dove è presente NY.

RENAME

funzione che rinomina un data set del SAS in una specifica libreria. La sintassi è:

CALL RENAME (libname,vecchio_nome,nuovo_nome);
dove:

libname è un carattere scalare contenente il nome del data set del SAS;

vecchio_nome è un carattere scalare contenente il vecchio nome del membro da rinominare;

nuovo_nome è un carattere scalare contenente il nuovo nome del membro.

REPEAT

funzione che crea una nuova matrice grazie alla ripetizione degl'amento matrice tempo n_righe*n_colonne; tempo n_righe attraversa le righe, e tempo n_colonne sotto le colonne. Per esempio: X={1 2, 3 4};

Y=REPEAT(X,2,3);

mi restituisce la seguente matrice Y:

$$\begin{array}{|c c c c c c|} \hline 1 & 2 & 1 & 2 & 1 & 2 \\ \hline 3 & 4 & 3 & 4 & 3 & 4 \\ \hline 1 & 2 & 1 & 2 & 1 & 2 \\ \hline 3 & 4 & 3 & 4 & 3 & 4 \\ \hline \end{array}$$

REPLACE

comando che sostituisce il valore di una variabile in un data set del SAS con il corrente valore valore della stessa variabile prendendolo dall'area di lavoro. REPLACE utilizza una matrice di elementi in ordine di riga sostituendo l'i-ma osservazione con l'i.mo elemento della matrice. Se ci sono più elementi di quanti sono gli elementi della matrice, sostituisce il contenuto ed utilizza gli ultimi elementi della matrice.

RESET

comando che pone delle opzioni ai processi. Possono essere implementate le seguenti opzioni:

PRINT

NOPRINT

LOG

NOLOG

ALL

NOALL

FW

DETAILS
NODETAILS
FLOW
NOFLOW
NAME
NONAME
CENTER
NOCENTER
LINESIZE
PAGESIZE
STORAGE
DEBUG
NODEBUG
DEFLIB
SPACES
SPILL
CASE
NOCASE
BREAK
NOBREAK

RESUME

comando che ci permette di continuare l'esecuzione dall'riga di programma dove è stata eseguita l'ultima istruzione di PAUSE.

RETURN

dice all'IML di tornare al punto del programma dove la LINK è stata emessa. Se non è stata emessa nessuna LINK, l'istruzione RETURN fa uscire l'IML dal modulo. Se non siamo in un modulo si ferma l'esecuzione e l'IML prosegue con l'analisi di un altro comando.

ROOT

funzione che effettua la decomposizione di Cholesky di una matrice (ad esempio A) nel seguente modo:

$$U'U=A$$

dove U è triangolare superiore. La matrice A deve essere simmetrica e definita positiva.

ROUND

funzione matriciale che arrotonda la matrice specificata in argomento. La sintassi corretta è:

ROUND(matrice<,unità_di_approssimazione>;

ROWCAT

funzione che prende una matrice di caratteri o una submatrice come argomento e crea una nuova matrice con una colonna i cui elementi sono la concatenazione degli elementi riga in una singola stringa. Se l'argomento ha n righe ed m colonne, il risultato ha n righe ed una colonna. Gli lunghezza degli elementi del risultato è pari ad m volte la lunghezza degli elementi dell'argomento. Nell'argomento possiamo indicare anche righe e colonne per indicare quali righe e colonne vogliamo concatenare. Ad esempio:

```
B={"ABC" "D" "EF",
  "GH" "I" "JK"};
A=ROWCAT(B);
```

mi da la seguente matrice A:

	ABCD	EF	
	GH	I	JK

ROWCATC

funzione che lavora come ROWCAT eccetto che per i valori mancanti che vengono spostati alla fine della concatenazione. Infatti:

```
B={"ABC" "D" "EF",
  "GH" "I" "JK"};
A=ROWCATC(B);
```

mi da la seguente matrice A:

ABCDEF
GHIJK

ROWNAME

funzione che associa etichette alle righe di una data metrice; ad esempio:

```
ROWNAME=nome_riga;
```

RUN

comando che richiede all'IML di eseguire un comando che è stato collocato all'interno di un modulo.

SETDIF

funzione che crea un vettore riga risultante da tutti gli elementi presenti (senza duplicati) all'interno del primo argomento ma non all'interno del secondo. Ad esempio:

```
A={1 2 4 5};
B={3 4};
C=SETDIF(A,B);
```

restituisce:

C=[1 2 5]

SETIN

comando che sceglie uno specificato data set tra tutti i data set già aperti con EDIT o USE. Questo data set diventa il data set che verrà utilizzato nell'istruzione successiva. La sintassi corretta è:

SETIN SASdataset <NOBS nome><POINT operatore>;
Non è obbligatorio specificare NOBS, che indica il numero di osservazioni del data set nella variabile nome. POINT fa delle specifiche osservazioni.

SETOUT

comando che sceglie uno specificato data set tra tutti i data set già aperti con EDIT. Questo data set diventa il data set che verrà utilizzato nell'istruzione successiva. La sintassi corretta è:

SETIN SASdataset <NOBS nome><POINT operatore>;
Non è obbligatorio specificare NOBS, che indica il numero di osservazioni correnti del data set nella variabile scalare nome. POINT fa delle specifiche osservazioni.

SHAPE

funzione che forma una matrice partendo da una matrice di differente dimensione. Vediamo un semplice esempio:

R=SHAPE(12,3 4);

ottengo una matrice R del tipo:

$$\begin{vmatrix} 12 & 12 & 12 & 12 \\ 12 & 12 & 12 & 12 \\ 12 & 12 & 12 & 12 \end{vmatrix}$$

SHOW

comando che indica quali informazioni sul sistema vanno scritte. Può essere seguito da:

OPTIONS: mostra il corrente ambiente di opzioni (vedi RESET);

SPACE: mostra la misura dell'area di lavoro e come si è iniziato ad utilizzarla.

MODULES: mostra i moduli che sono stati definiti.

ALL: mostra tutte le informazioni.

STORAGE: mostra le metriche del corrente data set o disco.

NAMES: mostra gli attributi di tutti i value.

ALLNAMES: è come NAMES, ma mostra anche gli

attributi delle variabili senza values.

DATASETS: mostra tutti i data set del SAS aperti.

CONTENTS: mostra i nomi e gli attributi delle variabili del corrente data set del SAS.

FILES: mostra tutti i file aperti.

WINDOWS: mostra tutte le finestre attive aperte con il comando WINDOW.

PAUSE: mostra tutti i moduli pausa.

name: mostra l'attributo nome.

SIN	funzione scalare che calcola il seno trigonometrico di ogni elemento dall'argomento. Il risultato è in radianti.
SINH	funzione scalare che calcola il seno iperbolico trigonometrico di ogni elemento dall'argomento. Il risultato è in radianti.
SKEWNESS	funzione che opera sugli elementi corrispondenti di una lista di matrici specificata in argomento, determinando lo “skewness”, misura di asimmetria, o più precisamente, una misura della tendenza delle deviazioni della media ad essere maggiori ad una direzione piuttosto che nell'altra.
SOLVE	funzione ch erisolve un'equazione lineare del tipo AX=B nell'incognita X. La matrice A deve essere quadrata e non singolare.
SORT	comando che raggruppa le osservazioni di un data set del SAS con una o più variabili, deposita i risultati ottenuti in un nuovo data set del SAS o sullo stesso data set di partenza.
SOUND	produce un suono del quale possimo specificare frequenza (in hertz) e durata (in secondi). La sintassi corretta è: CALL SOUND(freq,dur);
SQRSYM	funzione che prende una matrice come quelle generate dalla funzione SYMSQR e la ritrasforma in una matrice quadrata. Ad esempio: SQR=SQRSYM(SYMSQR({1 2, 3 4})); che sarebbe uguale a:

SQR=SQRSYM({1, 3, 4});
da una matrice SQR del tipo:

$$\begin{vmatrix} 1 & 3 \\ 3 & 4 \end{vmatrix}$$

SQRT

funzione che determina la radice quadrata positiva di ogni elemento dell'argomento.

SSQ

funzione che restituisce un singolo valore numerico che la somma (non corretta) dei quadrati di tutti gli elementi dell'argomento. Possiamo inserire al massimo 15 matrici numeriche come argomento.

START

comando che dice all'IML di aprire un modulo. E' sempre seguito da FINISH.

STD

funzione che opera sugli elementi corrispondenti di una lista di matrici specificata in argomento, determinando lo scarto quadratico medio (errore standard).

STDERR

funzione che opera sugli elementi corrispondenti di una lista di matrici specificata in argomento, determinando l'errore quadratico medio.

STOP

comando che ferma il programma IML, e nessun comando sulle matrici viene eseguito. Comunque l'IML continua l'esecuzione se sono stati inseriti molti comandi. Se l'esecuzione IML viene interrotta con il comando PAUSE, il comando STOP cancella tutte le pause e torna immediatamente nel modulo (vedi RETURN e ABORT).

STORE

comando che deposita una matrice di valori sulla libreria disco. Se non si nomina alcuna matrice, vengono depositate tutte le matrici. Se vogliamo depositare solo alcune matrici, dobbiamo fare la lista delle stesse.

SUBSTR

funzione che prende una matrice di caratteri (con posizione e lunghezza) come argomento, e produce una matrice di caratteri della stessa dimensione dell'argomento. Gli elementi della matrice ottenuta sono sub stringhe dei

corrispondenti elementi della matrice di partenza. Ogni substringa è costruita usando l'iniziale posizione fornita. Ad esempio:

```
B={abc  def  ghi,  jkl  mno  pqr};  
A=SUBSTR(B,3,2);
```

restituisce una matrice a del tipo:

$$\begin{vmatrix} c & f & i \\ l & o & r \end{vmatrix}$$

SUM

funzione di riduzione somma che ha come operatore corrispondente: +.

SUMMARY

comando IML che esegue statistiche per le variabili numeriche di un intero data set o per un subset del data set. Le opzioni di SUMMARY sono:

CLASS: specifica le variabili del corrente data set SAS sulle quali vogliamo lavorare. L'operando è un valore carattere contenente i nomi delle variabili.

VAR: richiede statistiche per variabili numeriche del corrente data set.

WEIGHT: specifica i caratteri contenenti il nome delle variabili numeriche nel corrente data set che verranno utilizzate come pesi per ogni osservazione.

STAT: esplicita alcuni requisiti statistici eseguiti. L'operando è un valore carattere contenente i requisiti richiesti.

OPT: seleziona le opzioni PRINT, NOPRINT, SAVE e NOSAVE.

WHERE: specifica i subset del corrente data set per il processo SUMMARY.

SVD CALL

comando che decomponе una matrice reale A (m*n dove m>=n) nella forma $A=U*diag(Q)*V'$ dove $U'U=V'V=VV'=I_n$; Q contiene singoli valori ed ha dimensione n*1; U ha dimensione m*n; V a dimensione n*n. Quando m>=n, U è la matrice degli autovettori ortonormalizzati di AA' . Q contiene le radici quadrate degli autovettori di $A'A$ ed AA' , eccetto per qualche zero.

SWEET

funzione che spazza la matrice indicata in argomento sul

vettore_index indicato in argomento producendo una nuova matrice. Il , valore del vettore_index indicato deve essere minore o uguale al numero di righe o di colonne della matrice.

SYMSQR

funzione che prende una matrice numerica quadrata (n*m) e compatta gli elementi del triangolo inferiore della matrice stessa in un vettore colonna (n*(n+1)/2 righe). Ad esempio:

SYMSQR({1 2, 3 4});

restituisce:

$$\begin{vmatrix} 1 \\ 3 \\ 4 \end{vmatrix}$$

TAN

funzione scalare che calcola la tangente trigonometrica di ogni elemento dall'argomento. Il risultato è in radianti.

TANH

funzione scalare che calcola la tangente iperbolica trigonometrica di ogni elemento dall'argomento. Il risultato è in radianti.

THEN

espressione che fa parte del ciclo IF... ,THEN,ELSE. E' obbligatorio indicarla e indica il comando o l'operazione da eseguire qualora la condizione risultasse vera.

TINV

funzione statistica che determina i quantili della distribuzione T di Student. La sintassi corretta è:

GAMINV(probabilità,gradi_di_libertà<,noncentralità>);

TRACE

produce un singolo valore numerico uguale alla somma degli elementi presenti sulla diagonale principale della matrice in argomento.

UNIFORM

funzione scalare che restituisce uno o più numeri pseudo-casuali con una distribuzione uniforme oltre l'intervallo 0-1. Restituisce una matrice della stessa dimensione della matrice specificata in argomento.

UNION

funzione che crea un vettore riga (senza duplicati) generato

dai valori presenti nell'argomento. L'argomento è un numero o un carattere (possiamo specificare fino a 15 argomenti). Ad esempio:

```
A={1 2 4 5};  
B={3 4};  
C=UNION(A,B);
```

restituisce:

```
C=[1 2 3 4 5]
```

UNIQUE

funzione che crea un vettore riga (senza duplicati) generato dai valori presenti nell'argomento. L'argomenti è un numero o un carattere (possiamo specificare fino a 15 argomenti). Questa funzione è identica all funzione UNION.

UPCASE

funzione che converte in maiuscolo gli elementi di una matrice stringa.

USE

istruzione che apre un data set per la lettura. Il data set non deve essere aperto.

VALSET

funzione che attende un solo carattere stringa contenente il nome della matrice. Il VALSET cerca la matrice indicata in argomento e sposta gli elementi indicati nell'argomento stesso nella matrice. Ad esempio:

```
B="A";  
CALL VALSET(B,99);
```

cerca il valore dell'argomento B che è A, quindi cerca A e

copia in A stesso il numero 99.

VALUE

funzione che attende un solo carattere stringa contenente il nome della matrice. Il VALUE cerca la matrice indicata in argomento e sposta i suoi elementi nella matrice risultato. Ad esempio:

```
A={1 2 3};  
B="A";  
C=VALUE(B);
```

restituisce:

```
C={1 2 3}
```

VAR

funzione che opera sugli elementi corrispondenti di una lista di matrici specificata in argomento, determinando varianza.

VECDIAG

funzione che crea un vettore colonna che ha come elementi gli elementi che formano la diagonale principale della matrice quadrata in argomento.

WINDOW

comando che definisce una finestra sullo schermo e può includere un numero di files. L'istruzione DISPLAY scrive valori nella finestra. Nel comando WINDOW possiamo specificare i seguenti argomenti:

nome_finestra: specifica 1-8 caratteri per il nome della finestra; le opzioni sono:

CLOSE: chiude la finestra.

opzioni_finestra: che controlla misura, posizione ed altri attributi della finestra;

ROWS: specifica il numero iniziale di righe della finestra. Il default è 23.

COLUMNS: specifica il numero iniziale di colonne della finestra. Il default è 78.

IROW: specifica dove deve essere posizionato il primo carattere della prima riga della finestra sullo schermo. Il default è 1.

ICOLUMN: specifica dove deve essere posizionato il primo carattere della prima colonna della finestra sullo schermo. Il default è 1.

COLOR: specifica il colore di fondo della finestra. Per default viene preso il colore nero. Possono essere utilizzati: nero, bianco, verde, magenta, rosso, giallo grigio, blu.

CMDLINE: specifica il nome della linea di comando che verrà riservata dall'utente.

MSGLINE: specifica il messaggio da mostrare come messaggio standard quando la finestra viene attivata.

GROUP: inizia una sequenza ripetitiva di gruppi di campi definiti per la finestra.

campi: è una sequenza di campi che specificano le posizioni alterate, campi operandi, formato ed opzioni.

ANALISI DELLE CORRISPONDENZE

Siano date due rilevazioni qualitative X (con k modalità) e Y (con h modalità) effettuate sulla stessa popolazione di numerosità n.

Si codifichino le modalità con dei numeri interi.

La distribuzione della popolazione può essere sintetizzata in una tabella a doppia entrata dove

- ♦ n_{ij} indica il numero di unità con risultato i-esimo per la prima rilevazione e con risultato j-esimo per la seconda rilevazione, ovvero frequenze assolute del risultato (i,j);
- ♦ $f_{ij} = \frac{n_{ij}}{n}$ indica la frequenza relativa di tale risultato;
- ♦ $f_{i\cdot} = \sum_{j=1}^h f_{ij}$ e $n_{i\cdot} = \sum_{j=1}^h n_{ij}$ indicano le frequenze marginali relative ed assolute delle X;
- ♦ $f_{\cdot j} = \sum_{i=1}^k f_{ij}$ e $n_{\cdot j} = \sum_{i=1}^k n_{ij}$ indicano le frequenze marginali relative e assolute delle Y.

x	y							marg.x
	1	2	3	...	j	...	h	
1								
2								
3								
...								$f_{\cdot j}$
i					f_{ij}			
...								
h								
marg. y					$f_{\cdot j}$			1

Lo scopo dell'analisi e' quello di porre in evidenza le eventuali dipendenze dei risultati delle due rilevazioni attraverso alcune rappresentazioni grafiche della tabella di contingenza.

Una tabella di contingenza può essere letta secondo le righe o secondo le colonne ottenendo due tipi di tabelle:

1) quella dei profili riga

x	y							marg.x
	1	2	3	...	j	...	h	
1								1
2								1
3								1
...								1
i					$f_{ij}/f_{i.}$			1
...								1
								1
marg. y					$f_{.j}$			1

dove si studia il carattere x rispetto ad y.

Nell'esempio ciascun profilo riga rappresenta la frazione $f_{i.}$ di individui, ossia ha un peso relativo pari a $f_{i.}$.

2) quella dei profili colonna

x	y							marg.x
	1	2	3	...	j	...	h	
1								
2								
3								
...								f_i
i					$f_{ij}/f_{i.}$			
...								
k								
marg. y	1	1	1	1	1	1	1	1

dove si studia il carattere y rispetto ad x.

Nell'esempio ciascun profilo colonna rappresenta la frazione $f_{.j}$ di individui, ossia ha un peso relativo pari a $f_{.j}$.

A questo punto per stabilire se le variabili x ed y sono indipendenti (cioè se i profili riga o colonna sono tutti simili tra loro) e' possibile utilizzare il test χ^2 di indipendenza.

Si utilizza la seguente statistica:

$$\chi_m^2 = n \sum_{i=1}^k \sum_{j=1}^h \frac{(f_{ij} - f_{i.} f_{.j})^2}{f_{i.} f_{.j}}$$

Supponendo per ipotesi x ed y indipendenti, la variabile aleatoria descritta sopra, per n sufficientemente grande, segue un distribuzione χ^2 con $m=(h-1)(k-1)$ gradi di libertà.

Fissato un livello di significatività α si trova quel valore tale che $\chi_m^2 > c_\alpha = \alpha$.

Se il valore sperimentale che si calcola supera la soglia c_α , allora si rifiuta l'ipotesi di indipendenza tra le due variabili x ed y.

ANALISI DEI PROFILI RIGA

La tabella dei profili riga può rappresentarsi come k punti di coordinate $f_{ij}/f_{i.}$ (j =1,..,h) in un sottospazio di h-1 dimensioni.

Sussiste infatti la relazione $\sum_{j=1}^h \frac{f_{ij}}{f_{i.}} = 1$. Il baricentro o profilo medio della nuvola è il

punto $(f_{1.}, f_{2.}, \dots, f_{h.})$ corrispondente al totale marginale di Y.

La matrice R dei profili riga di dimensione (k,h), il cui elemento generico è $f_{ij}/f_{i.}$ può essere scritta come

$$R = D_r^{-1} M \text{ dove}$$

- ◊ M = matrice di dimensione (k,h), il cui elemento generico è f_{ij} corrispondente alla tabella di contingenza originale;
- ◊ D_r = matrice diagonale di dimensione (k,k) contenente sulla diagonale i valori $(f_{1.}, f_{2.}, \dots, f_{k.})$ cioè i pesi delle righe, ovvero le freq. marginali ;
- ◊ D_r^{-1} = matrice diagonale ricavata da D_r il cui elemento (i,i) della diagonale è $1/f_{i.}$.

CALCOLO DELLA DISTANZA TRA DUE PUNTI

Per meglio interpretare ed evidenziare le differenze fra due punti i ed i' si prende in considerazione χ^2 che sostanzialmente è una normalizzazione della distanza euclidea.

$$\chi^2(i, i') = \sum_{i=1}^h \frac{(f_{ij}/f_{i.} - f_{i'j}/f_{i'.})^2}{f_{.j}} \quad \text{che può essere scritta}$$

$$\chi^2(i, i') = \sum_{i=1}^h \left(\frac{f_{ij}}{f_{i.} \sqrt{f_{.j}}} - \frac{f_{i'j}}{f_{i'.} \sqrt{f_{.j}}} \right)^2$$

Se il punto è il baricentro (profilo medio) si ha:

$$\chi^2(i, y_b) = \sum_{i=1}^h \frac{(f_{ij}/f_{i.} - f_{i'j}/f_{i'.})^2}{f_{.j}}$$

Nel nostro caso si effettua un cambiamento di scala, moltiplicando la matrice R dei profili riga per la radice quadrata delle coordinate del baricentro.

L' i -esimo punto diventa:

$$i^\wedge = \left(\frac{f_{i1}}{f_{i.} \sqrt{f_{.1}}}, \frac{f_{i2}}{f_{i.} \sqrt{f_{.2}}}, \dots, \frac{f_{ih}}{f_{i.} \sqrt{f_{.h}}} \right)$$

In questo caso l'usuale distanza euclidea fra due punti i ed i^\wedge sarà:

$$d^2(i^\wedge, i'^\wedge) = \sum_{j=1}^h \left(\frac{f_{ij}}{f_{i.} \sqrt{f_{.j}}} - \frac{f_{i'j}}{f_{i'.} \sqrt{f_{.j}}} \right)^2$$

che è proprio la distanza χ^2 appena vista.

USO DELLE MATRICI PER IL CAMBIAMENTO DI SCALA

la matrice R^\wedge dei profili riga trasformati sarà

$$R^\wedge = R D_c^{-1/2} = D_r^{-1} M D_c^{-1/2} \text{ dove}$$

- ◊ D_c = matrice diagonale contenete i valori $(f_{.1}, f_{.2}, \dots, f_{.h})$
- ◊ $D_c^{-1/2}$ = matrice diagonale contenente i valori $(\frac{1}{\sqrt{f_{.1}}}, \frac{1}{\sqrt{f_{.2}}}, \dots, \frac{1}{\sqrt{f_{.h}}})$

Il generico elemento della matrice è $\frac{f_{ij}}{f_{.i} \sqrt{f_{.j}}}$.

Il nuovo baricentro ha coordinate $(\sqrt{f_{.1}}, \sqrt{f_{.2}}, \dots, \sqrt{f_{.h}})$.

ANALISI IN COMPONENTI PRINCIPALI

Indichiamo con Γ^r la matrice di varanza covarianza di R^\wedge di dimensione (h, h) il cui termine generale è:

$$\beta(j, j') = \sum_{i=1}^k f_{ij} \left(\frac{f_{ij} - f_{.i} f_{.j}}{f_{.i} \sqrt{f_{.j}}} \right) \left(\frac{f_{ij'} - f_{.i} f_{.j'}}{f_{.i} \sqrt{f_{.j'}}} \right) = \sum_{i=1}^k \left(\frac{f_{ij} - f_{.i} f_{.j}}{\sqrt{f_{.i} f_{.j}}} \right) \left(\frac{f_{ij'} - f_{.i} f_{.j'}}{\sqrt{f_{.i} f_{.j'}}} \right)$$

si ha che:

$$\Gamma^r = X' X$$

dove:

- ◊ X = matrice corrispondente alla radice quadrata della matrice contenente i contributi delle celle al calcolo della statistica per il test χ^2 di indipendenza delle rilevazioni x ed y divisa per n ;

◊ X' = matrice trasposta di X .

Il rango di Γ^r sarà : $rg = \min(h, k) - 1$.

Inoltre si indica con:

◊ Λ la matrice diagonale degli autovalori di Γ^r ; gli autovalori non nulli saranno rg ;

◊ U la matrice che ha per colonne u_1, \dots, u_h autovettori normalizzati della matrice Γ^r .

Ora, se indichiamo con α un asse principale dei profili riga con $\alpha = 1, \dots, rg$, il vettore contenente le proiezioni della nuvola su tale asse sarà

$$C_\alpha^r = R^\wedge u_\alpha = D_r^{-1} M D_c^{-1/2} u_\alpha$$

Infatti per la particolare forma di R^\wedge la proiezione di punti su un asse principale può essere fatto a partire dalla matrice dei punti non centrati.

ANALISI DEI PROFILI COLONNA

Procediamo in modo analogo a quanto visto per i profili riga:

$$C = D_c^{-1} M'$$

◊ M' = matrice di dimensione h, k corrispondente alla tabella originale il cui elemento generico è $f_{j,i}$

◊ D_c^{-1} matrice diagonale di dimensione (h, h) il cui elemento (j, j) è $1/f_{j,j}$

La matrice C^\wedge dei profili colonna trasformati sarà

$$C^\wedge = C D_r^{-1/2} = D_c^{-1} M' D_r^{-1/2}$$

Il cui generico elemento è $\frac{f_{ij}}{\sqrt{f_{j,j}}}$.

Il nuovo baricentro ha coordinate $(\sqrt{f_{1,1}}, \sqrt{f_{2,2}}, \dots, \sqrt{f_{k,k}})$

ANALISI IN COMPONENTI PRINCIPALI

Analogamente a quanto fatto per i profili riga indichiamo

$$\Gamma^c = XX'$$

dove:

- ◊ Γ^c = matrice di varianza covarianza di C^\wedge
- ◊ Λ = matrice diagonale degli autovalori non nulli di Γ^c : gli autovalori non nulli di Γ^c sono uguali a quelli di Γ^r :
- ◊ V = matrice che ha per colonne v_1, \dots, v_k autovettori normalizzati della matrice Γ^c .

Il vettore contenente le proiezioni della nuvola su tale asse sarà:

$$c_\alpha^c = C^\wedge v_\alpha = D_c^{-1} M' D_r^{-1/2} v_\alpha$$

FEDELTA DELLA RAPPRESENTAZIONE DELLA NUVOLA IN UNO SPAZIO DI DIMENSIONE RIDOTTA

L'inerzia, cioè la dispersione della nuvola attorno al baricentro, ad esempio per i profili riga, è

$$I = \sum_{j=1}^h \sum_{i=1}^k f_i \left(\frac{f_{ij} - \bar{f}_{i.} \bar{f}_{.j}}{\sqrt{\bar{f}_{i.} \bar{f}_{.j}}} \right)^2 = \sum_{j=1}^h \sum_{i=1}^k \frac{(f_{ij} - \bar{f}_{i.} \bar{f}_{.j})^2}{\bar{f}_{i.} \bar{f}_{.j}}$$

che è il valore della statistica χ^2 diviso per n. Essendo simmetrica in i e j ha lo stesso valore per i profili colonna.

Come già osservato nel caso dell'analisi in componenti principali l'inerzia può essere calcolata anche a partire dalle nuove coordinate.

Ricordiamo che le nuvole dei punti stanno in uno spazio di dimensione $rg = \min(h, k) - 1$.

La varianza delle proiezioni di un profilo su un asse principale è λ_α .

Quindi $I = \sum_{\alpha=1}^{rg} \lambda_\alpha$

Se si rimpiazza la nuvola di profili con la sua proiezione nel sottospazio generato dai primi q autovettori, con $q < rg$, la dispersione della nuvola proiettata è $\lambda_1 + \dots + \lambda_q$.

La fedeltà di una proiezione è il rapporto fra la dispersione della nuvola proiettata e la dispersione della nuvola originale:

$$\frac{\sum_{\alpha=1}^q \lambda_\alpha}{\sum_{\alpha=1}^{rg} \lambda_\alpha}$$

cioè il rapporto dell'inerzia nello spazio di dimensione minore rispetto all'inerzia originale.

CONTRIBUTO DI UN PROFILO ALLA VARIANZA DI UN ASSE PRINCIPALE

La varianza delle proiezioni di un profilo riga su un asse principale è, come già osservato λ_α .

Ricordando che i punti sono centrati nel baricentro e che ciascun punto ha un peso $f_{i.}$, il contributo di ciascun punto alla varianza dell'asse è $f_{i.} (\mathcal{C}_{i\alpha}^r)^2$, avendo indicato $\mathcal{C}_{i\alpha}^r$ la coordinata dell'i-esimo profilo sull'asse α .

Quindi il rapporto $\frac{f_{i.} (\mathcal{C}_{i\alpha}^r)^2}{\lambda_\alpha}$ misura la parte dell'i-esimo profilo riga nella varianza spiegata dell'asse α ; ad esempio un alto contributo indica che il punto ha giocato un ruolo maggiore nell'orientamento finale dell'asse principale.

Analogamente per i profili colonna il rapporto $\frac{f_{.j} (\mathcal{C}_{i\alpha}^c)^2}{\lambda_\alpha}$ misura la parte dell>j-esimo profilo colonna nella varianza spiegata dell'asse α .

Questo indice dipende dal peso dei profili.

CONTRIBUTO DI UN ASSE PRINCIPALE ALL'INERZIA DI UN PROFILO

Per i profili riga se indichiamo con $d(i, G^r)$ la distanza di un profilo dal baricentro, allora il contributo dell' α - esimo asse all'inerzia di un profilo è dato da:

$$\left(\frac{C_{i\alpha}^r}{d(i, G^r)} \right)^2$$

La somma dei valori precedenti su tutti gli assi fattoriali vale 1.

Tale indice è il **quadrato del coseno** dell'angolo fra l'i-esimo profilo riga e l' α - esimo asse fattoriale.

Questo valore si interpreta anche come il quadrato del **coefficiente di correlazione** fra l'i-esimo profilo riga e l' α - esimo asse fattoriale.

Essendo indipendente dal peso dei profili il valore precedente viene anche detto **contributo relativo**.

Questo indice indica la qualità della rappresentazione del profilo sull'asse.

Se lo spazio ridotto scelto ha dimensione q , allora:

$$\sum_{\alpha=1}^q \left(\frac{C_{i\alpha}^r}{d(i, G^r)} \right)^2$$

indica la qualità della rappresentazione del punto sullo spazio scelto.

Analogamente per i profili colonna il valore

$$\left(\frac{C_{i\alpha}^c}{d(j, G^c)} \right)^2$$

si interpreta come contributo dell' α - esimo asse all'inerzia di un profilo ovvero il quadrato del **coefficiente di correlazione** fra il j-esimo profilo riga e l' α - esimo asse fattoriale.

Valori alti indicano buona qualità della rappresentazione del profilo sull'asse.

PROGRAMMA PER L'ANALISI DELLE CORRISPONDENZE

```
data dati; *nome del nuovo dataset ;
infile 'a:\provino3.dat'; *file esterno contenente i dati ;
input var1 $ var2 $; *nomi delle variabili ;
run;
proc iml; *istruzione di avvio dell' interprete IML ;

* MODULO PER IL CALCOLO DELLA TABELLA DI CONTINGENZA;
start contingenza(x,col_1,col_2,mod_1,mod_2,tab_cont); *inizio della definizione del modulo ;
run tab_disg(x,col_1,mod_1,disg_1); *esecuzione del modulo ;
run tab_disg(x,col_2,mod_2,disg_2); *esecuzione del modulo ;
tab_cont=disg_1`*disg_2; *tabella di contingenza ;
finish; *fine della definizione del modulo ;

* MODULO PER IL CALCOLO DELLE TABELLE DISGIUNTIVE;
start tab_disg(x,col,modalita,disg); *inizio della definizione del modulo ;
colonna=x[,col]; *vettore colonna del primo o secondo carattere (dipende dal valore di col) ;
num_unita=nrow(colonna); *numero di unita' del vettore colonna ;
modalita=unique(colonna); *estrae e ordina in un vettore riga (modalita) gli elementi differenti
dal vettore colonna colonna ;
num_div=ncol(modalita); * numero modalita del vettore riga (modalita) ;
disg=repeat(colonna,1,num_div)=repeat(modalita,num_unita,1); *dal confronto ottengo la
tabella disgiuntiva del primo
o secondo carattere ;
finish; *fine della definizione del modulo ;

* MODULO PER IL CALCOLO AUTOVETTORI AUTOVALORI COORDINATE RIGA E
COLONNA;
start calcolo (t_cont,cord_righe,cord_col,cont_rr,cont_rc,cont_ar,cont_ac,
fre_m_r,fre_m_c,autoval,mat_relativa,totale,ob_val,prod_fm,mat_dist);
run trasf_mat(t_cont,totale,mat_relativa,tab_freq,fre_m_r,fre_m_c,ob_val,prod_fm,mat_dist);
run autovalore(totale,tab_freq,val_cons,autoval,mat_dev_cod);
run coordinate(t_cont,totale,val_cons,autoval,mat_dev_cod,cord_righe,cord_col);
run contributi(t_cont,totale,autoval,cord_righe,cord_col,cont_rr,cont_rc,cont_ar,cont_ac);
finish;

* MODULO PER IL CALCOLO DEGLI AUTOVALORI;
start autovalore(totale,tab_freq,val_cons,autoval,mat_dev_cod);
mat_dev_cod=tab_freq`*tab_freq; *matrice devianze codevianze della tabella di contingenza ;
autoval=round(eigval(mat_dev_cod),0.00001); *vettore colonna contenente gli autovalori
arrotondati della matrice delle devianze
codevianze ;
val_cons=(loc((autoval^=1)&(autoval^=0))); * vettore colonna contenente la posizione degli
autovalori diversi da 1 e 0 ;
autoval=autoval[val_cons,]; * vettore colonna contenete gli autovalori diversi da 1 e 0 ;
finish;
```

```

* MODULO PER IL CALCOLO DELLA MATRICE;
start trasf_mat(t_cont,totale,mat_relativa,tab_freq,fre_m_r,fre_m_c,ob_val,prod_fm,mat_dist);
totale=t_cont[+,+]; * totale frequenze della matrice di contingenza ;
mat_relativa=t_cont/totale; *matrice composta dalle frequenze relative fij ;
mat_col_rel=repeat(mat_relativa[+,],nrow(t_cont),1); * matrice contenente le frequenze marginali
f.j ;
den_mat=sqrt(mat_col_rel#mat_relativa[+]); * matrice contenente sqrt(f.i.*f.j) ;
tab_freq=mat_relativa/den_mat; *matrice contenente fij/(sqrt(f.i.*f.j)) ;

* CALCOLO PER TROVARE L'INERZIA DEI PUNTI RIGA E COLONNA;
prod_fm=(t_cont[+,+]\#repeat(t_cont[+,],nrow(t_cont),1))/totale; * matrice contenente
(ni.*n.j)/totale ;
ob_val=t_cont-prod_fm; *matrice contenente nij-(ni.*n.j)/totale ;
mat_dist=(ob_val##2)/prod_fm; * matrice contenente (nij-(ni.*n.j)/totale))^(2/((ni.*n.j)/totale)) ;
fre_m_r=mat_dist[,+]; * vettore colonna contenente le frequenze marginali delle righe della
matrice mat_dist ;
fre_m_c=mat_dist[+,:]; * vettore riga contenente le frequenze marginali delle colonne della
matrice mat_dist ;
finish;

* MODULO PER IL CALCOLO DELLE COORDINATE;
start coordinate(t_cont,totale,val_cons,autoval,mat_dev_cod,cord_righe,cord_col);
autovet=eigvec(mat_dev_cod); * matrice contenente gli autovettori della matrice delle devianze
codevianze ;
autovet=autovet[val_cons,:]; * matrice contenente gli autovettori i cui autovalori sono diversi da
1 e 0 ;
cord_col=sqrt(autoval#(totale/t_cont[,]))\#autovet; *matrice contenente le coordinate colonna ;

* FORMULA ADOTTATA PER IL CALCOLO DELLE COORDINATE COLONNA:

```

$$y_{hj}^c = \sqrt{\lambda_h} \sqrt{\frac{n}{n_{ij}}} V_{hj}$$

```

cord_righe=((t_cont/repeat(t_cont[,],1,ncol(t_cont)))*cord_col)`\#(1/sqrt(autoval)); * matrice
contenente le coordinate
riga ;

```

- FORMULA ADOTTATA PER IL CALCOLO DELLE COORDINATE RIGA:

$$Y_{h.j} = \sqrt{\lambda_h} \sum_{j=1}^s \frac{N_{ij}}{N_{i.}} Y_{hj}.$$

finish;

- * MODULO PER IL CALCOLO DEI CONTRIBUTI;
start contributi(t_cont,totale,autoval,cord_righe,cord_col,cont_rr,cont_rc,cont_ar,cont_ac);

* CALCOLO DEI CONTRIBUTI RELATIVI PER I PUNTI RIGA;
 den_r=(t_cont/repeat(t_cont[+],1,ncol(t_cont)))-(repeat(t_cont[+]/totale,nrow(t_cont),1));
 cont_rr=(cord_righe##2)/repeat((totale/t_cont[+])*(den_r##2),nrow(cord_righe),1);
 * FORMULA ADOTTATA:

$$\rho_i^h = \frac{\left(y_{hi}^r \right)}{\sum_{j=1}^s \frac{n}{n_{ni.}} \left(\frac{n_{ij}}{n_{i.}} - \frac{n_{.j}}{n} \right)^2}$$

* CALCOLO DEI CONTRIBUTI REALTIVI PER I PUNTI COLONNA;
 den_c=(t_cont/repeat(t_cont[+],nrow(t_cont),1))-(repeat(t_cont[+]/totale,1,ncol(t_cont)));
 cont_rc=(cord_col##2)/repeat((totale/t_cont[+])*(den_c##2),nrow(cord_col),1);
 * FORMULA ADOTTATA : (vedere sopra)

* CALCOLO DEI CONTRIBUTI ASSOLUTI PER I PUNTI RIGA;
 or=(cord_righe##2)/repeat(autoval,1,nrow(t_cont));
 cont_ar=repeat((t_cont[+]/totale)',nrow(or),1)#or;
 * FORMULA ADOTTATA:

$$\alpha_{jh} = \frac{n_i}{n} \frac{\left(y_{hi}^r \right)^2}{\lambda_h}$$

* CALCOLO DEI CONTRIBUTI ASSOLUTI PER I PUNTI COLONNA;
 oc=(cord_col##2)/repeat(autoval,1,ncol(t_cont));
 cont_ac=repeat((t_cont[+]/totale),nrow(oc),1)#oc;
 finish;
 * FORMULA ADOTTATA : (vedere sopra)

do;
 use dati; * per la lettura del database ;
 read all var_char_ into x; * per scrivere il contenuto del dataset nella matrice x;
 run contingenza(x,1,2,mod_1,mod_2,t_cont);
 run calcolo(t_cont,cord_righe,cord_col,cont_rr,cont_rc,cont_ar,cont_ac,fre_m_r,fre_m_c,autoval,
 mat_relativa,totale,ob_val,prod_fm,mat_dist);
 end;

* STAMPA ;
 * IMPOSTAZIONE STAMPA (TRASPOSIZIONE E ESTRAPOLAZIONE DELLE 2 DIM);
 cord_righe=cord_righe';cord_righe=cord_righe[,{1 2}];
 cord_col=cord_col';cord_col=cord_col[,{1 2}];
 cont_rr=cont_rr';cont_rr=cont_rr[,{1 2}];
 cont_rc=cont_rc';cont_rc=cont_rc[,{1 2}];
 cont_ar=cont_ar';cont_ar=cont_ar[,{1 2}];
 cont_ac=cont_ac';cont_ac=cont_ac[,{1 2}];
 reset nolog noname center; * opzioni di stampa ;

```

* STAMPA TABELLA DI CONTINGENZA;
t_con=t_cont;
t_con=t_con // t_con[+]; * aggiunta margini ;
t_con=t_con || t_con[+]; * aggiunta margini ;
mo_1=mod_1 || "Tot";mo_2=mod_2 || "Tot"; * aggiunta modalita "sum" ;
print "Tabella di Contingenza",t_con[rowname=mo_1 colname=mo_2],,;

* STAMPA DELLA TABELLA DI INDIPENDENZA;
print "Chi-Square Statistic Expected Value",
prod_fm[rowname=mod_1 colname=mod_2];
print /;

* STAMPA DELLA TABELLA OBSERVED MINUS EXPECTED VALUES;
print "Observed Minus Expected Values",
ob_val[rowname=mod_1 colname=mod_2],,;

* STAMPA DELLA TABELLA CONTRIBUTIONS TO THE TOTAL CHI-SQUARE
STATISTIC;
mat_dist=mat_dist || mat_dist[+];
mat_dist=mat_dist //mat_dist[+];
print "Contributions to the total Chi-Square Statistic",
mat_dist[rowname=mo_1 colname=mo_2];
print /;

* STAMPA PROFILI RIGA E COLONNA;
prof_r=t_cont/repeat(t_cont[+],1,ncol(t_cont));
prof_c=t_cont/repeat(t_cont[+],nrow(t_cont),1);
print "Tabella Profili Riga",
prof_r[rowname=mod_1 colname=mod_2],,;
print "Tabella Profili Colonna",
prof_c[rowname=mod_1 colname=mod_2];
print /;

* STAMPA INERTIA AND CHI-SQUARE DECOMPOSITION;
sing=sqrt(autoval);m_sing={"S.Value"};
perc=autoval/autoval[+];m_perc={"Percents"};
a_val=autoval//autoval[+];m_val={"Princ.Inertia"};
chi=a_val#totale;m_chi={"Chi-Sq."};
print "Inertia and Chi-Square Decomposition",
sing[colname=m_sing] a_val[colname=m_val] chi[colname=m_chi] perc[colname=m_perc],,;

* STAMPA DELLE COORDINATE RIGA;
print "Coordinate riga",
print "Dim1 Dim2",cord_righe[rowname=mod_1];
print /;

```

```

* STAMPA STATISTICHE RIASSUNTIVE PER I PROFILI RIGA;
in_p_r=fre_m_r/(totale # autoval[+]);
marg_fre=mat_relativa[+];
mat_sum=cont_rr[+] || marg_fre || in_p_r;
mod_sum={"Quality" "Mass" "Inertia"};
print "Summary Statistic per i Punti Riga",
mat_sum[rowname=mod_1 colname=mod_sum],,,;

* STAMPA CONTRIBUTI ASSOLUTI PER I PUNTI RIGA;
print "Contributi Assoluti per Punti Riga",
print "  Dim1  Dim2",cont_ar[rowname=mod_1];
print /;

* STAMPA INDICE DELLE COORDINATE CHE CONTRIBUISCONO MAGGIORMENTE
ALL' INERZIA PER I PUNTI RIGA;
ind_co_r=choose(cont_ar[,1]>cont_ar[,2],1,2); * vettore colonna contenente il valore 1 o 2 a
                                                 seconda del risultato (vera o falsa)
                                                 dell'espressione ;
print "Indices of the Coordinate that Contribute Most to Inertia for the Row Points",
print "  Best",ind_co_r[rowname=mod_1],,,;

* STAMPA CONTRIBUTI RELATIVI PER LE RIGHE;
print "Contributi Relativi per le Righe",
print "  Dim1  Dim2",cont_rr[rowname=mod_1];
print /;

* STAMPA COORDINATE COLONNA;
print "Coordinate colonna",
print "  Dim1  Dim2",cord_col[rowname=mod_2],,,;

* STAMPA STATISTICHE RIASSUNTIVE PER I PROFILI COLONNA;
in_p_c=(fre_m_c)`/(totale # autoval[+]);
marg_f_c=(mat_relativa[+])`;
cont_rc=round(cont_rc,0.0001);
mat_sum_c=cont_rc[+] || marg_f_c || in_p_c;
print "Summary Statistics per i Punti Colonna",
mat_sum_c[rowname=mod_2 colname=mod_sum];
print /;

* STAMPA CONTRIBUTI ASSOLUTI PER I PUNTI COLONNA";
print "Contributi Assoluti per i Punti Colonna",
print "  Dim1  Dim2",cont_ac[rowname=mod_2],,,;

* STAMPA INDICE DELLE COORDINATE CHE CONTRIBUISCONO MAGGIORMENTE
ALL' INERZIA PER I PUNTI COLONNA;
ind_co_c=choose(cont_ac[,1]>cont_ac[,2],1,2);
print "Indices of the Coord.that Contribute Most to Inertia for the column points",
print "  Best",ind_co_c[rowname=mod_2];
print /;

```

```
* STAMPA CONTRIBUTI RELATIVI PER LE COLONNE;
print "Contributi Relativi per le colonne",
print "  Dim1  Dim2",cont_rc[rownname=mod_2];
quit;
```

MATRICE DEI DATI

La seguente matrice dei dati è stata ricavata da un indagine sul giudizio del fumo .I soggetti selezionati si dividono in quattro fasce di età (15-24, 25-34, 35-44,45-oltre)che hanno espresso un loro giudizio sul fumo(favorevoli, indifferenti, contrario) .La seguente matrice è stata ottenuta attraverso un campionamento.(provino3.dat).

FASCE DI ETA'	GIUDIZIO
15-24	FAVOREVOLI
25-34	FAVOREVOLI
25-34	FAVOREVOLI
25-34	INDIFFERENTI
25-34	CONTRARIO
25-34	CONTRARIO
25-34	CONTRARIO
15-24	INDIFFERENTI
15-24	INDIFFERENTI
35-44	CONTRARIO
35-44	INDIFFERENTI
35-44	INDIFFERENTI
35-44	INDIFFERENTI
35-44	CONTRARIO
35-44	CONTRARIO
35-44	CONTRARIO
35-44	FAVOREVOLI
45-OLTRE	INDIFFERENTI
45-OLTRE	CONTRARIO
45-OLTRE	INDIFFERENTI
45-OLTRE	FAVOREVOLI
45-OLTRE	CONTRARIO
45-OLTRE	CONTRARIO
45-OLTRE	CONTRARIO

OUTPUT PROGRAMMA IML

Tabella di Contingenza

	contrari	favorevo	indiffer	Tot
15-24	1	4	2	7
25-34	3	2	4	9
35-44	3	1	4	8
44-oltre	5	1	1	7
Tot	12	8	11	31

Tabella di indipendenza dei due caratteri qualitativi in esame.

Chi-Square Statistic Expected Value

	contrari	favorevo	indiffer
15-24	2.7096774	1.8064516	2.483871
25-34	3.483871	2.3225806	3.1935484
35-44	3.0967742	2.0645161	2.8387097
44-oltre	2.7096774	1.8064516	2.483871

Tabella di contingenza assoluta

Observed Minus Expected Values

	contrari	favorevo	indiffer
15-24	-1.709677	2.1935484	-0.483871
25-34	-0.483871	-0.322581	0.8064516
35-44	-0.096774	-1.064516	1.1612903
44-oltre	2.2903226	-0.806452	-1.483871

Test χ^2 di indipendenza delle due variabili

Contributions to the total Chi-Square Statistic

	contrari	favorevo	indiffer	Tot
15-24	1.078725	2.6635945	0.0942606	3.8365801
25-34	0.0672043	0.0448029	0.2036494	0.3156566
35-44	0.0030242	0.5488911	0.4750733	1.0269886
44-oltre	1.9358679	0.360023	0.8864684	3.1823593
Tot	3.0848214	3.6173115	1.6594517	8.3615846

Tabella Profili Riga

	contrari	favorevo	indiffer
15-24	0.1428571	0.5714286	0.2857143
25-34	0.3333333	0.2222222	0.4444444
35-44	0.375	0.125	0.5
44-oltre	0.7142857	0.1428571	0.1428571

Tabella Profili Colonna

	contrari	favorevo	indiffer
15-24	0.0833333	0.5	0.1818182
25-34	0.25	0.25	0.3636364
35-44	0.25	0.125	0.3636364
44-oltre	0.4166667	0.125	0.0909091

La tabella di decomposizione dell'inerzia e del χ^2 include:

i singoli valori $\sqrt{\lambda_\alpha}$, gli autovalori λ_α per ogni dimensione, l'inerzia totale $\sum \lambda_\alpha$, il $\chi^2 = n \lambda_\alpha$ per ciascuna dimensione, il totale del $\chi^2 = n \sum \lambda_\alpha$ e la percentuale di inerzia per ogni dimensione rispetto all'inerzia totale.

Inertia and Chi-Square Decomposition

S.Value	Princ.Inertias	Chi-Sq.	Percents
0.4337972	0.18818	5.83358	0.6976606
0.2855696	0.08155	2.52805	0.3023394
	0.26973	8.36163	

Per il calcolo delle coordinate riga e' stata utilizzata la formula:

$$Y_{h.j} = \sqrt{\lambda_h} \sum_{j=1}^s \frac{N_{ij}}{N_{i.}} Y_{hj.}$$

Coordinate riga

	Dim1	Dim2
15-24	0.7045364	0.2274014
25-34	0.0216393	-0.186023
35-44	-0.154267	-0.323381
44-oltre	-0.556053	0.3813498

Le statistiche sommarie per i punti riga includono:

- la qualita' della rappresentazione dei profili nello spazio di dimensione ridotta, ottenuta facendo la somma dei contributi relativi per ciascuna modalita';
- peso dei profili;
- inerzia dei profili.

Summary Statistic per i Punti Riga

	Quality	Mass	Inertia
15-24	1	0.2258065	0.4588316
25-34	1	0.2903226	0.0377506
35-44	1	0.2580645	0.1228216
44-oltre	1	0.2258065	0.3805908

Le contribuzioni assolute forniscono il contributo di ogni punto all'individuazione di un asse. La formula utilizzata e' la seguente:

$$\alpha_{jh} = \frac{n_i}{n} \frac{(y_{hi}^r)^2}{\lambda_h}$$

Contributi Assoluti per Punti Riga

	Dim1	Dim2
15-24	0.5956206	0.1431853
25-34	0.0007224	0.1231946
35-44	0.0326363	0.3309278
44-oltre	0.3710184	0.4026794

Indice delle coordinate per i punti riga che contribuiscono maggiormente nel calcolo dell'inezia dell'asse.

Indices of the Coordinate that Contribute Most to Inertia for the Row Points

Best

15-24	1
25-34	2
35-44	2
44-oltre	2

Le contribuzioni relative indicano quanto un punto e' rappresentato su ogni asse. La formula adottata e':

$$\rho_i^h = \frac{(y_{hi}^r)}{\sum_{j=1}^s \frac{n}{n_{ni.}} \left(\frac{n_{ij}}{n_{i.}} - \frac{n_{.j}}{n} \right)^2}$$

Contributi Relativi per le Righe

	Dim1	Dim2
15-24	0.9056504	0.0943496
25-34	0.013351	0.986649
35-44	0.1853833	0.8146167
44-oltre	0.6801136	0.3198864

Per il calcolo delle coordinate colonna e' stato utilizzata:

$$Y_{h.j} = \sqrt{\lambda_h} \sum_{j=1}^s \frac{N_{ij}}{N_{i.}} Y_{hj.}$$

Coordinate colonna

	Dim1	Dim2
contrari	-0.475187	0.1768235
favorevo	0.6198486	0.2606787
indiffer	0.067587	-0.382483

Le statistiche sommarie per i punti colonna includono:

- la qualita' della rappresentazione dei profili nello spazio di dimensione ridotta, ottenuta facendo la somma dei contributi relativi per ciascuna modalita';
- peso dei profili;
- inerzia dei profili.

Summary Statistics per i Punti Colonna

	Quality	Mass	Inertia
contrari	1	0.3870968	0.3689258
favorevo	1	0.2580645	0.4326084
indiffer	1	0.3548387	0.1984603

La formula utilizzata per i contributi assoluti per i punti colonna e' analoga a quella dei punti riga.

Contributi Assoluti per i Punti Colonna

	Dim1	Dim2
contrari	0.464489	0.1484142
favorevo	0.5268974	0.215038
indiffer	0.0086136	0.6365477

Determinata in maniera analoga ai punti riga

Indices of the Coord.that Contribute Most to Inertia for the column points

Best

contrari	1
favorevo	1
indiffer	2

I contributi relativi per i punti colonna sono ricavati in maniera analoga a quelli dei punti riga.

Contributi Relativi per le colonne

	Dim1	Dim2
contrari	0.8784	0.1216
favorevo	0.8497	0.1503
indiffer	0.0303	0.9697

Exiting IML.

PROGRAMMA IN SAS PER L'ANALISI DELLE CORRISPONDENZE

```

data matrix;
infile 'a:provino3.dat';
input var1 $ var2 $;
proc corresp all data=matrix;
tables var1 , var2;
run;

```

OUTPUT DEL PROGRAMMA IN SAS

The Correspondence Analysis Procedure

Contingency Table

	contrari	favorevo	indiffer	Sum
15-24	1	4	2	7
25-34	3	2	4	9
35-44	3	1	4	8
44-oltre	5	1	1	7
Sum	12	8	11	31

Chi-Square Statistic Expected Values

	contrari	favorevo	indiffer
15-24	2.70968	1.80645	2.48387
25-34	3.48387	2.32258	3.19355
35-44	3.09677	2.06452	2.83871
44-oltre	2.70968	1.80645	2.48387

Observed Minus Expected Values

	contrari	favorevo	indiffer
15-24	-1.70968	2.19355	-0.48387
25-34	-0.48387	-0.32258	0.80645
35-44	-0.09677	-1.06452	1.16129
44-oltre	2.29032	-0.80645	-1.48387

Contributions to the Total Chi-Square Statistic

	contrari	favorevo	indiffer	Sum

15-24	1.07873	2.66359	0.09426	3.83658
25-34	0.06720	0.04480	0.20365	0.31566
35-44	0.00302	0.54889	0.47507	1.02699
44-oltre	1.93587	0.36002	0.88647	3.18236
Sum	3.08482	3.61731	1.65945	8.36158

Row Profiles

	contrari	favorevo	indiffer
15-24	0.142857	0.571429	0.285714
25-34	0.333333	0.222222	0.444444
35-44	0.375000	0.125000	0.500000
44-oltre	0.714286	0.142857	0.142857

Column Profiles

	contrari	favorevo	indiffer
15-24	0.083333	0.500000	0.181818
25-34	0.250000	0.250000	0.363636
35-44	0.250000	0.125000	0.363636
44-oltre	0.416667	0.125000	0.090909

The Correspondence Analysis Procedure

Inertia and Chi-Square Decomposition

Singular Values	Principal Inertias	Chi- Squares	Percents	14	28	42	56	70			
0.43380	0.18818	5.83357	69.77%	*****	*****	*****	*****	*****			
0.28557	0.08155	2.52802	30.23%	*****	*****	*****	*****	*****			
				-----	-----	-----	-----	-----			
				0.26973 8.36158 (Degrees of Freedom = 6)							

Row Coordinates

	Dim1	Dim2
15-24	0.704536	0.227401
25-34	0.021639	-.186023
35-44	-.154267	-.323381
44-oltre	-.556053	0.381350

Summary Statistics for the Row Points

	Quality	Mass	Inertia
15-24	1.00000	0.225806	0.458834
25-34	1.00000	0.290323	0.037751
35-44	1.00000	0.258065	0.122822
44-oltre	1.00000	0.225806	0.380593

Partial Contributions to Inertia for the Row Points

	Dim1	Dim2
15-24	0.595622	0.143187
25-34	0.000722	0.123196
35-44	0.032636	0.330932
44-oltre	0.371019	0.402685

Indices of the Coordinates that Contribute Most to Inertia for the Row Points

	Dim1	Dim2	Best
15-24	1	1	1
25-34	0	0	2
35-44	0	2	2
44-oltre	2	2	2

Squared Cosines for the Row Points

	Dim1	Dim2
15-24	0.905650	0.094350
25-34	0.013351	0.986649
35-44	0.185383	0.814617
44-oltre	0.680114	0.319886

Column Coordinates

	Dim1	Dim2
contrari	-.475187	0.176822
favorevo	0.619848	0.260677

indiffer	0.067587	- .382480
----------	----------	-----------

Summary Statistics for the Column Points

	Quality	Mass	Inertia
contrari	1.00000	0.387097	0.368928
favorevo	1.00000	0.258065	0.432611
indiffer	1.00000	0.354839	0.198461

Partial Contributions to Inertia for the Column Points

	Dim1	Dim2
contrari	0.464489	0.148414
favorevo	0.526897	0.215038
indiffer	0.008614	0.636548

Indices of the Coordinates that Contribute Most to Inertia for the Column Points

	Dim1	Dim2	Best
contrari	1	0	1
favorevo	1	1	1
indiffer	0	2	2

Squared Cosines for the Column Points

	Dim1	Dim2
contrari	0.878374	0.121626
favorevo	0.849717	0.150283
indiffer	0.030280	0.969720