

Lettura a lista e scrittura del file da programma

```
data pluto;  
input x1-x3;  
datalines;  
1 5 7  
9 3  
2  
6 9 8  
13 5 8  
;  
run;
```

equivalentemente

```
input x1-x3;  
cards;  
1 5 7  
9 3  
2  
6 9 8  
13 5 8  
;  
run;
```

Output SAS data set PLUTO

x1	x2	x3
1	5	7
9	3	2
6	9	8
13	5	8

LETTURA A COLONNA

Input dataset: INPUT_COLONNA.TXT

100m34
200m65
300f29
400m31
500m45
600f40
700f68
800m51
900m48
999m36

```
data pippo;  
infile  
'F:\written\didattica\CorsoSAS\input_colonna.txt';  
input codice 1-3 genere $ 4 eta 5-6;  
run;
```

LETTURA CON FORMATO

Non si specificano le colonne ma la lunghezza di ogni campo

INPUT *var informat.* ;

INPUT (*varlist*) (*informat list*) ;

INPUT (*varlist*) (*[n*] informat.*) ;

Sintassi generale *informat*

w. legge numeri interi o decimali con punto decimale codificato nel campo

\$w. Legge stringhe di caratteri ASCII

```
*LETTURA A FORMATO: i campi non sono separati;  
data pippo;  
infile  
'F:\written\didattica\CorsoSAS\input_colonna.txt';  
input codice 3.  genere $1.  eta 2.;  
run;
```

In questo caso i dati sono separati da uno spazio quindi devo incrementare il puntatore

```
*LETTURA A FORMATO: i campi sono separati da uno
spazio;
data pippo;
infile
'F:\written\didattica\CorsoSAS\input_lista.txt';
input codice 3. +1 genere $1. +1 eta 2.;
run;
```

alternativamente il simbolo @i indica al puntatore di spostarsi alla colonna i

```
data pippo;
infile
'F:\written\didattica\CorsoSAS\input_lista.txt';
input codice 3. @5 genere $1. @7 eta 2.;
run;
```

Si tenga presente che la lettura con formato è indispensabile nel caso in cui ci siano da leggere dei campi contenenti una data.

Esempio: Supponiamo di avere dei dati provenienti dalla degustazione di 4 vini di marche diverse. I degustatori sono 3. I dati inseriti nel file sono: la marca del vino, l'anno di produzione, la data di degustazione, i punteggi dei tre degustatori.

```

*Esempio di lettura con formato ;
data punti_vino;
infile cards;
input marca $1 +1 anno 2. +1 data ddmmyy10. +1
(punt11-punt13) (3*4.);
cards;
A 93 20-11-1994 7.8 7.1 6.5
B 95 23-12-1998 7.9 7.6 7.5
C 99 10-11-2000 6.5 7.0 6.8
D 00 30-10-2002 5.9 6.4 7.2
;
run;

```

```

proc print data=punti_vino;
var marca anno punt11-punt13 data;
format data date8.;
run;

```

```

proc print data=punti_vino;
var marca anno punt11-punt13 data;
format data MONyy.;
run;

```

```

proc print data=punti_vino;
var marca anno punt11-punt13 data;
format data DDMMyy.;
run;

```

LETTURA DI PIU' OSSERVAZIONI DA UNO STESSO RECORD

@@ indica la reale fine del record, quindi nell'esempio che segue i dati vengono letti a coppie

```
/*LETTURA DI PIU' OSSERVAZIONI DA UNO STESSO  
RECORD*/  
data unrecord;  
input genere $ peso @@;  
cards;  
m 60 f 50 m 68  
m 82 f 55 f 56  
;  
run;
```

LETTURA DI UNA OSSERVAZIONE SU PIU' RECORDS

/ indica di andare al record successivo

#n indica di andare al record n

Nell'esempio l'osservazione relativa a ciascun vino è registrata su 3 record:

- nel 1° la marca del vino, l'anno di produzione,
- nel 2° la zona di provenienza del vino,
- nel 3° il punteggio attribuito al vino.

```
/*LETTURA DI una OSSERVAZIONE SU PIU' RECORD*/  
*ESEMPIO VINO, ELIMINATA LA DATA;  
data punti_vino_zona;  
infile cards;  
input marca $ anno / ZONA $ #3 punti;  
cards;  
A 1993  
ZONA1  
7  
B 1995  
ZONA2  
7.9  
C 1999  
ZONA3  
6.5  
D 2000  
ZONA4  
5.9  
;  
RUN;
```

IMPORTAZIONE/ESPORTAZIONE AUTOMATICA DI DATI

Dal Menu principale:

- ❖ File
- ❖ Import Data/Export Data
- ❖

*** tracciato record indagine consumi ISTAT**

*esempio di importazione da EXCEL dati Istat “indagine sui Consumi delle famiglie italiane”, il file **CONSUMO_TOSCANA.XLS** è una selezione di variabili per la sola Regione Toscana

Se l’operazione di importazione ha funzionato nella finestra di Log apparirà:

NOTE: WORK.CONSUMO_TOSCANA was
successfully created

IL LINGUAGGIO SAS: PROCEDURE GENERALI RELATIVE ALLA VISUALIZZAZIONE DEI DATA SET

❖PROC CONTENTS

❖PROC PRINT

❖PROC FORMAT

**PROGRAM_PROCEDURE_GENERALI.SAS;*

PROC CONTENTS: permette di visualizzare molte informazioni sul Sas data set, tra cui la directory della libreria SAS e l'elenco di tutte le variabili in ordine alfabetico.

PROC CONTENTS *<options>;*

Con l'opzione VARNUM la lista delle variabili rispetta l'ordinamento fisico del Sas data set

PROC PRINT: stampa le osservazioni del Sas data set relative a tutte o ad una selezione di variabili. Si possono creare report semplicissimi (elenco osservazioni) o anche complessi utilizzando le varie opzioni.

PROC PRINT < *option(s)*>;

BY < DESCENDING> *variable-1*

<...< DESCENDING> *variable-n*>< NOTSORTED>;

PAGEBY *BY-variable*;

SUMBY *BY-variable*;

ID *variable(s)*;

SUM *variable(s)*;

VAR *variable(s)*;

```
/*inizia dalla osservazione 5 e stampa 10
osservazioni*/
proc print data= consumo_toscana (firstobs=5
obs=10) ;
var numcomp sessol etal;
run;
```

```
/*inizia dalla osservazione 1 e stampa 10  
osservazioni, inserendo la SUM*/  
proc print data= consumo_toscana (obs=10);  
sum bar;  
var numcomp sessol etal bar;  
run;
```

PROC FORMAT: consente di creare una libreria di format permanenti da utilizzare nei data step o nelle procedure

PROC FORMAT <option(s)>;

.....

VALUE <\$>name <(format-option(s))>

value-range-set(s);

```

*****PROC FORMAT*****;
proc format;
value $genere 'm'='maschio'
              'f'='femmina';
value si_no    1='sì'
              2='no';

run;

data segreteria;
input      id 1. @3 genere $1. @5 ricovero 1. @7
analisi 1.;
          format genere $genere.
              ricovero si_no.
              analisi si_no.;

cards;
1 m 1 1
2 f 2 1
3 f 1 2
4 m 2 1
;

run;

proc print;run;

```

LINGUAGGIO SAS: GESTIONE DEI DATA SET

❖ **LABEL**

❖ **RENAME**

❖ **KEEP**

❖ **DROP**

❖ **RETAIN**

***PROGRAM_GESTIONE1 ;**

LABEL: consente di assegnare nomi estesi alle variabili la *label* non può superare 40 caratteri;

LABEL *variable* = "*label*" ... ;

RENAME: consente di modificare il nome della variabile;

RENAME *vecchio_nome* = *nuovo_nome*. ;

DROP: elenca le variabili da eliminare nel SAS data set;

DROP *variabili* ;

KEEP: elenca le variabili da scrivere nel SAS data set;

KEEP *variabili* ;

Esempio: seleziono solo le variabili che riguardano il capofamiglia

```
data lib.consumo_toscana;set consumo_toscana;  
run;
```

```
data info_capof (keep=rela1 genere1 eta1 statociv1  
titstul conprof1 posprof1);  
set lib.consumo_toscana;  
label sessol1=sesso capofamiglia eta1=eta  
capofamiglia;  
rename sessol1=genere1 ;  
run;
```

N.B. Nel data set `info_capof` le variabili non sono nell'ordine specificato, se voglio che il data set rispetti l'ordinamento delle variabili che io specifico devo usare l'istruzione `retain`

```
data info_capof (keep=rela1 genere1 eta1 statociv1  
titstul conprof1 posprof1);  
retain rela1 genere1 eta1 statociv1 titstul  
conprof1 posprof1;  
set lib.consumo_toscana;  
label sessol=sesso capofamiglia eta1=eta  
capofamiglia;  
rename sessol=genere1 ;  
run;
```

PARTE III

LINGUAGGIO SAS: ISTRUZIONI DI ASSEGNAZIONE

Variabile=Espressione

L'*espressione* è una sequenza di:

- *operandi* (variabili, costanti);
- *operatori* (caratteri speciali, funzioni, parentesi)

Tipo di operatori:

Aritmetici, di comparazione, logici, carattere

Regole ordine di esecuzione

- I. Espressioni entro parentesi
- II. 7 livelli di priorità (1=massima)

OPERATORI ARITMETICI

<i>priorità</i>	<i>simbolo</i>	<i>descrizione</i>
1	**	Elevamento a potenza
2	*	moltiplicazione
2	/	Divisione
3	+	addizione
3	-	sottrazione

Gli operatori aritmetici

- Agiscono su variabili di tipo numerico;
- Conversione automatica da carattere a numerico;
- Gli operatori con uguale priorità vengono eseguite da sin a dx; l'elevamento a potenza da dx a sin;
- le parentesi possono modificare la priorità degli operatori;
- tutte le operazioni aritmetiche vengono eseguite in doppia precisione;

OPERATORI DI COMPARAZIONE

<	LT	Minore di
<=	LE	Minore o uguale di
>	GT	Maggiore di
>=	GE	Maggiore o uguale di
^=	NE	Non uguale
	IN	Uguale a uno degli elementi della lista

Gli operatori di comparazione:

- effettuano un confronto tra due operandi, tale confronto genera un valore numerico (1 confronto vero, 0 confronto falso)
- hanno tutti lo stesso livello gerarchico
- possono operare su variabili/costanti numeriche e/o carattere
- la variabile carattere viene trasformata in numerica se il confronto è tra numerica e carattere;
- il valore missing è sempre considerato il più piccolo

OPERATORI LOGICI

Priorità		
1	\wedge	NOT
2	$\&$	AND
3	$ $	OR

Gli operatori logici consentono di mettere in relazione:

- più variabili
- due o più espressioni operando sul loro risultato

LINGUAGGIO SAS: ISTRUZIONI “WHERE” e “IF..THEN..ELSE”

❖ Istruzione WHERE

❖ Istruzione IF..THEN <ELSE>

WHERE: seleziona le osservazioni in fase di esecuzione della procedura, lasciando inalterato l’archivio da cui legge i dati. Da ricordare che tale istruzione viene eseguita dopo che hanno avuto effetto le opzioni relative al Sas data set.

WHERE espressione;

IF: valuta un’espressione e condizionatamente al risultato esegue i comandi che seguono

IF *expression* **THEN** *clause* <; **ELSE** *clause*>

```

/*uso di operatori aritmetici, comparazione,
logici, WHERE, IF THEN ELSE*/
data info_capof1;set info_capof;
etasql=eta1**2; *assegnazione;
statociv1r=statociv1;
if statociv1=3 or statociv1=4 or statociv1=5 then
statociv1r=3;
tit1r=titstul;
if titstul in (1,2,3) then tit1r= 1;
if titstul =4 then tit1r= 2;
if titstul in(5,6) then tit1r= 3;
if titstul in(7,8) then tit1r= 4;

if posprof1 ge 1 and posprof1 le 9 then dipl=1;
else dipl=0;
*where dipl=1;
run;

```

```

/*uso di IF THEN per la creazione di nuovi data
set*/
data donne uomini;set info_capof;
if genere1=2 then output donne;else output uomini;
run;

```

Il FORMAT può essere definito anche per la visualizzazione di tabelle

```
*****PROC  FORMAT*****;  
  
PROC FORMAT ;  
value genere 1='maschio'  
              2='femmina';  
value titstu 1='laurea o laurea breve'  
              2='diploma'  
              3='licenza media o qualifica'  
              4='lic. elementare,  
analfabeta';  
  
proc freq data=info_capof1;  
tables genere1 tit1r;  
format genere1 genere. tit1r titstu.;  
run;
```