SAS

STATISTICAL ANALYSIS SYSTEM

Uso di SAS per le analisi statistiche

A cura di

Laura Neri

Dip. di Economia Politica e Statistica

Università degli Studi di Siena

PARTE I

IL SISTEMA SAS

Sistema integrato di prodotti software

- data entry, manipolazione archivi;
- stesura di report e grafici;
- analisi statistiche e matematiche;
- previsioni e supporto alle decisioni;
- ricerca operativa e project management;
- sviluppo di applicazioni.

Il fulcro del sistema SAS è il modulo SAS Base

- linguaggio SAS;
- procedure per analisi dei dati e stesura di report;
- macro-linguaggio.

AMBIENTE DI LAVORO A FINESTRE

Le finestre base del SAS per Windows sono 5:

- **SAS Explorer** contenuto delle LIBRARIES e dei SAS datasets;
- **SAS Results** visualizzazione dei risultati di ogni procedura eseguita;

e le finestre di programmazione

- **SAS Enhanced Editor e Program Editor** scrittura istruzioni (programmi);
- **SAS Log** esito operazioni eseguite. Una volta eseguito il programma, SAS scrive dei messaggi nel SAS LOG, ignorare tali messaggi può essere pericoloso perché talvolta si ottengono dei risultati ma tali risultati potrebbero essere non corretti per qualche problema intercorso nelle istruzioni digitate;
- **SAS Output** risultati delle esecuzioni.

Inoltre:

SAS SYSTEM HELP per la consultazione della sintassi e delle opzioni del linguaggio e delle procedure

Visualizzazione dei risultati nella finestra di OUTPUT

Una volta ‘sottomesso’ il programma SAS (SAS windowing environment), i risultati appariranno nella finestra OUTPUT e nella finestra Results window.

ORGANIZZAZIONE DEL SISTEMA SAS

L’analisi dei dati si svolge seguendo due passi fondamentali:

1) L’organizzazione dei dati
2) L’analisi dei dati

Nel SAS System

Data Step

Inizia con un *data statement* e consente di leggere/creare e/o modificare archivi di dati

Proc Step

Inizia con un *proc statement*, ovvero richiama una procedura SAS, la esegue su un SAS dataset e produce dei risultati

NB la suddetta suddivisione è una semplificazione perché in realtà vedremo che anche il *proc statement* può creare un data set

Un programma SAS può essere costituito da

- Un solo Data Step
- Un solo Proc Step
- Più Data Step e/o più Proc Step

SAS DATA SET

Data Step e Proc Step si applicano ai file in formato SAS - SAS DATA SET -

I SAS DATA SET sono organizzati in forma rettangolare

V_1	V_2	....	....	V_k

- ogni riga rappresenta un'osservazione;
- ogni colonna rappresenta una variabile
- tutte le osservazioni possiedono le stesse variabili➔ le variabili non osservate per una data osservazione sono registrate come mancanti (*missing*)

Ogni SAS dataset è autodescrittivo, infatti esplicita:

- Il nome del *SAS data set*
- Il nome delle variabili, le *label*, ed il formato
- Il contenuto delle variabili per ogni osservazione

LIBRERIE: DEFINIZIONE E GESTIONE

Per definire una libreria permanente si possono seguire due strade:

- fare click sul comando New Library della barra Menu ed inserire le informazioni richieste, tra cui il nome della library ed il percorso fisico associato alla libreria stessa
- Scrivere nella finestra Program Editor l'istruzione

LIBNAME *mylibname 'path'*;

Dove:

mylibname è il nome attribuito alla libreria

path è il percorso fisico dove vengono memorizzati i dati

IL LINGUAGGIO SAS: concetti introduttivi

In SAS vengono utilizzati SAS statement per scrivere una serie di istruzioni (*statement*) che vanno a costituire il SAS PROGRAM.

Naturalmente per scrivere un programma SAS è necessario utilizzare il linguaggio appropriato il SAS LANGUAGE.

Il SAS PROGRAM è costituito da una serie di istruzioni ordinate.

La prima regola da seguire nella stesura di un programma è:

every SAS statement ends with a semicolon

Istruzioni SAS (*SAS statement)*

- Iniziano con una parola chiave
- Terminano sempre con il carattere punto e virgola
- Possono essere scritte con lettere maiuscole e minuscole, il linguaggio SAS non è *case sensitive*
- Possono iniziare in una qualsiasi colonna della riga e proseguire su più righe

- Più istruzioni possono essere scritte sulla stessa riga ma devono essere separate da punto e virgola

Per **salvare** un programma SAS si fa uso del Menu principale. Il programma viene salvato con estensione .SAS

REGOLE PER NOMI SAS DEFINITI DALL'UTENTE

- Numero massimo di caratteri dipende dal tipo di nomi (32, 8)
- Il primo carattere deve essere una lettera o un underscore "_"
- I caratteri successivi al primo possono essere lettere, cifre, "_"
- Indipendentemente da maiuscolo o minuscolo, SAS converte tutto in maiuscolo
- Nei nomi non possono comparire spazi bianchi
- Non sono ammessi caratteri speciali ($,£,*…)
- Non si possono utilizzare nomi di variabili automatiche del sistema (_N_, *ERROR*_)
- Non si possono utilizzare nomi che il SAS riserva a librerie speciali (LIBRARY, SASHELP, WORK, USER, MAPS..)

ESECUZIONE PROGRAMMI SAS

Esecuzione in modo semi-interattivo

- Stesura del programma nella finestra Enhanced/Program editor
- Visualizzazione risultati nella finestra Output
- Segnalazioni inviate dal sistema (messaggi di errore, warning, altro) nella finestra Log

Il programma SAS viene automaticamente compilato dal sistema prima di essere eseguito:

- ❖compilazione ed esecuzione avvengono a blocchi (data step, proc step);
- ❖i blocchi si chiudono con la parola chiave RUN;

L’ISTRUZIONE **DATA** e L’ISTRUZIONE **SET**

DATA *nomedataset <opzioni>;*

SET *nomedataset;*

RUN;

Dove *nomedataset* è il nome di un Sas dataset. Tale nome può essere scritto a due livelli:

nome1.nome2

- *nome1*: è il nome di primo livello ed indica la libreria in cui memorizzare il Sas data set (per default la libreria è WORK)
- *nome2*: è il nome del Sas data set che viene memorizzato nel percorso fisico associato alla libreria.

Come opera il suddetto blocco di istruzioni?

- ✓ legge il file indicato all’istruzione **SET**
- ✓ scrive sul file indicato all’istruzione **DATA**

LETTURA DI DATI DI TIPO ASCII

- Istruzione **INFILE**
- Istruzione **INPUT**
- Istruzione **DATALINES** (**CARDS)**

Istruzione **INFILE**

Indica al sistema dove leggere i dati

INFILE *'nomefile' [opzioni];*

nomefile

nome, con eventuale percorso, del file ASCII da leggere o parola chiave CARDS se i dati sono inseriti da programma

Istruzione **INPUT**

Definisce nome, tipo e modo di lettura delle variabili

I modi di lettura sono: a lista, a colonna, con formato

LETTURA A LISTA

Possibile quando i dati sono registrati in formato libero, con almeno uno spazio bianco tra un campo ed il successivo

INPUT *var1 var2 var3;*

INPUT *var1-var10;*

INPUT *var1 $ var2;*

Lettura a lista da file esterno

Input dataset: INPUT_LISTA.TXT

100 1 34
200 1 65
300 2 29
400 1 31
500 1 45
600 2 40
700 2 68
800 1 51
900 1 48

```
data pippo;
infile 'F:\written\didattica\CorsoSAS\input_lista.txt';
input codice genere eta;
run;
```

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;
```

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;
```

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 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 a 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 sesso1 eta1;
run;
```

```
/*inizia dalla osservazione 1 e stampa 10
osservazioni, inserendo la SUM*/
proc print data= consumo_toscana (obs=10);
sum bar;
var numcomp sesso1 eta1 bar;
run;
```

LINGUAGGIO SAS: GESTIONE DEI DATA SET

- **RENAME**
- **KEEP**
- **DROP**

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
titstu1 conprof1 posprof1);
set lib.consumo_toscana;
label sesso1=sesso capofamiglia eta1=eta
capofamiglia;
rename sesso1=genere1 ;
run;
```

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

priorità	*simbolo*	*descrizione*
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	^	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;
etasq1=eta1**2; *assegnazione;
statociv1r=statociv1;
if statociv1=3 or statociv1=4 or statociv1=5 then
statociv1r=3;
tit1r=titstu1;
if titstu1 in (1,2,3) then tit1r= 1;
if titstu1 =4 then tit1r= 2;
if titstu1 in(5,6) then tit1r= 3;
if titstu1 in(7,8) then tit1r= 4;

if posprof1 ge 1 and posprof1 le 9  then dip1=1;
else dip1=0;
*where dip1=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 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;
```