

# Introduzione a SAS

Gianluca Della Vedova

Univ. Milano–Bicocca  
<http://gianluca.dellavedova.org>

16 maggio 2016, revisione 0189fde

# Gianluca Della Vedova

- Ufficio U14-2041
- <http://gianluca.dellavedova.org>
- [gianluca.dellavedova@unimib.it](mailto:gianluca.dellavedova@unimib.it)

# Finestre

- Editor (F5)
- Log (F6)
- Output (F7)
- Icona Esegui (F8)

# Altre finestre

- Explorer
- Viewtable
- Risultati

# Data set

- righe = osservazioni
- colonne = variabili

# Data Step

```
1 DATA nome;  
2 /*  
3     elenco variabili (con formato)  
4     come inserire i dati  
5 */  
6 RUN;  
7
```

# Data Step: dati nel programma

```
1 DATA nomipersone;  
2     INPUT nome$ cognome$ altezza peso;  
3     DATALINES;  
4         Mario Rossi 178 69  
5         Lucia Bianchi 170 57  
6         Andrea Verdi 169 69  
7         ;  
8 RUN;
```

# Valori mancanti

- Rappresentati da un punto .
- Inseriti come spazi o punti



# Data Step: Importazione

- Da file di dati grezzi (.txt .csv .dat)
- Da wizard di importazione
- Da viewtable

# Data set

- nome di data set  $\leq 32$  caratteri
- lettere e cifre
- cifre in fondo
- nome di variabili  $\leq 32$  caratteri

# Filesystem

- Ogni data set corrisponde ad un file
- Una directory (cartella) contenente data set = **libreria**
- `LIBNAME nomelibreria directory;`
- data set permanente = in una libreria
- data set temporaneo = libreria `WORK` = no libreria
- nome libreria  $\leq 8$  caratteri

# Filesystem

- Ogni data set corrisponde ad un file
- Dati: data set oppure file di dati grezzi

# Data Step: da dati grezzi

```
1 DATA nomipersone;  
2   INFILE 'E:\anagrafe.txt';  
3   INPUT nome$ cognome$ altezza peso;  
4 RUN;
```

# Data Step: da dati grezzi 2

```
1 LIBNAME esercizi "E:\libreriaSAS";  
2  
3 DATA esercizi.nomipersone;  
4   INFILE 'E:\anagrafe.txt';  
5   INPUT nome$ cognome$ altezza peso;  
6 RUN;
```

Dati separati da spazi

## Data Step: da dati grezzi 2

```
1 LIBNAME esercizi "E:\libreriaSAS";  
2  
3 DATA esercizi.nomipersone;  
4   INFILE 'E:\anagrafe.txt' LRECL=2000;  
5   INPUT nome$ cognome$ altezza peso;  
6 RUN;
```

Se la riga contiene > 256 caratteri.

# Data Step - lettura da file

```
1 DATA nome;  
2   INFILE nomefile DLM=', ' DSD;  
3   INPUT nome$ cognome$ altezza peso;  
4 RUN;
```

Dati separati da virgola

DSD: campi alfanumerici racchiusi da virgolette



# Data Step - lettura da file

```
1 DATA nome;  
2   INFILE nomefile DLM='09'x;  
3   INPUT nome$ cognome$ altezza peso;  
4 RUN;
```

Dati separati da tabulazione

# Data Step - lettura da file

- **FIRSTOBS:** da quale riga iniziare
- **OBS:** quante righe leggere
- **MISCOVER:** variabili non assegnate=MISSING
- **TRUNCOVER:** dati a fondo riga = scartati

# Formato a colonne

```
1 DATA nome;  
2   INFILE nomefile;  
3   INPUT nome$ 1-10 cognome$ 11-20  
4       altezza 21-25 peso 28-30;  
5 RUN;
```

- Il cognome inizia alla colonna n. 11 e termina alla colonna n. 20.
- Righello nel log
- Gli spazi non separano.

# Formati Alfanumerici

: \$20.

Rappresenta il numero massimo di caratteri di un campo (in questo caso 20). Il dato termina con il primo spazio.

\$20.

Esattamente 20 caratteri.

# Formati Numerici

10.3

Numero totale di caratteri/cifre

.

Numero di cifre decimali

# Formati Date

- DDMMYY8.
- DATE7.
- Le date sono rappresentate in un formato interno, come numero di giorni dall'1/1/1960.
- Quindi una data è un numero
- Differenza fra date

# Funzioni su numeri

- `SUM(var1 var2 var3)`, `MIN(var1 var2 var3)`,  
`MAX(var1 var2 var3)`, `MEAN(var1 var2 var3)`,
- `SUM(of var1-var3)`
- funzioni su variabili numeriche.
- Attenzione ai valori mancanti.

# Funzioni su numeri

- concatena due stringhe: `CAT(var1, var2)`
- estrae sottostringa: `SUBSTR(var, inizio, lunghezza)`
- converte in formato numerico: `INPUT(var, informat)`
- trova una sottostringa: `INDEX(var, sottostringa)`
- calcola la lunghezza: `LENGTH(var)`
- trasforma in maiuscolo: `UPCASE(var)`



# Altre funzioni

- MONTH(data)
- YEAR(data), DAY(data)
- Sono funzioni che agiscono sulle date
- '01Mar2000'd

# Formati

```
1 DATA nome;  
2   INFILE nomefile;  
3   INPUT nome$ +1 cognome$ @30  
4     (altezza peso) COMMA7.;  
5 RUN;
```

- +1 salta un carattere
- le parentesi raggruppano variabili
- COMMA7. legge numeri con virgole come separatore migliaia
- DOLLAR7.2 come COMMA, ma con un dollaro all'inizio
- @30 va a colonna 30

# Formati

- @'code:' vai alla parte di dato grezzo che è uguale a code:
- / vai alla riga successiva dei dati
- @@ per avere più di una osservazione in una riga di dati

# Stampa

```
1 PROC PRINT;  
2 RUN;
```

- Stampa il contenuto dell'ultimo data set creato.

# Stampa

```
1 PROC PRINT DATA=prova;  
2     VAR nome;  
3 RUN;
```

- Stampa solo le variabili specificate

# Stampa

```
1 PROC PRINT;  
2     VAR nome;  
3     ID cognome;  
4     TITLE 'elenco delle persone';  
5 RUN;
```

- Usa la variabile `cognome` al posto di `obs`.
- Definisce il titolo.

# Stampa

```
1 PROC PRINT;  
2     VAR nome;  
3     ID cognome;  
4     TITLE 'elenco delle persone';  
5     WHERE peso > 80;  
6 RUN;
```

- WHERE definisce su quali osservazioni agire

# Etichette di stampa

```
1 LABEL variabile='Etichetta';
```

L'etichetta della variabile deve essere dentro il DATA step



# Formati di stampa

- `FORMAT variabile 16.;`
- `FORMAT variabile DATE7.;`
- `FORMAT variabile 8.2;`
- La `FORMAT` deve essere dentro la `PROC PRINT`

# Formati di stampa

```
1 PROC PRINT;  
2     VAR nome;  
3     ID cognome;  
4     TITLE 'elenco delle persone';  
5     WHERE peso > 80;  
6     FORMAT (cognome nome) $30.;  
7 RUN;
```

# Riepilogo data set

```
1 PROC CONTENTS DATA=data set;  
2 RUN;
```

# Nuovi formati

```
1 PROC FORMAT;  
2   VALUE formato  0='Rosso'  
3                 1='Giallo'  
4                 2='Blu';  
5 RUN;
```

- Adesso `formato.` è utilizzabile in una istruzione `format.`

# Nuovi formati

```
1 PROC FORMAT;  
2   VALUE $formato2  'RED'='Rosso'  
3                   'YELLOW'='Giallo'  
4                   'BLUE'='Blu';  
5 RUN;
```

# Nuovi formati

```
1 PROC PRINT DATA=vario;  
2   FORMAT data DATE7.;  
3   FORMAT colore formato.;  
4 RUN;
```

# Data Step - copia

```
1 DATA nome;  
2   SET data set originale;  
3   KEEP variabili da tenere;  
4 RUN;  
5  
6 DATA nome;  
7   SET data set originale;  
8   DROP variabili da eliminare;  
9 RUN;  
10  
11 DATA nome;  
12   SET data set originale;  
13   KEEP variabili da tenere;  
14   RENAME=(vecchia=nuova);  
15 RUN;
```

# Data Step - copia

```
1 DATA nomipersone;  
2   SET persone;  
3   KEEP nome cognome;  
4 RUN;  
5
```



# Creazione variabili

```
1 DATA persone2;  
2   SET persone;  
3   rapporto=altezza/peso;  
4 RUN;
```

- Le operazioni fra variabili creano nuove variabili
- Operazioni usuali, + - / \* \*\*

# Selezione di osservazioni

- IF condizione;

```
1 DATA personealte;  
2   SET persone;  
3   IF altezza GT 180;  
4 RUN;
```

- Operatori confronto, EQ =, GT >, GE ≥, LT <, LE ≤, NE ≠
- Valori non numerici devono essere racchiusi da apici

# Selezione di osservazioni 2

```
1 DATA personealte;  
2   SET persone;  
3   IF altezza LE 180 THEN DELETE;  
4 RUN;
```

- Rimozione di osservazione

# Istruzioni condizionali

```
1 IF condizione THEN DO;  
2     istruzione 1;  
3     istruzione 2;  
4     istruzione 3;  
5 END;
```

- Più semplice se una sola istruzione
- IF condizione THEN istruzione;

# Condizioni complesse

```
1 DATA personealteemagre;  
2   SET persone;  
3   IF altezza GE 180 AND peso LE 70;  
4 RUN;
```

- Operatori logici: AND, OR, NOT

## Condizioni complesse 2

```
1 DATA personealteemagre;  
2 SET persone;  
3 IF altezza GE 180 AND peso LE 70 THEN  
4     TIPO = 'A';  
5 ELSE IF altezza LT 170 THEN  
6     TIPO = 'B';  
7 ELSE IF peso GT 90 THEN  
8     TIPO = 'C';  
9 RUN;
```

- Cosa succede se un'osservazione non soddisfa alcuna condizione?
- E se ne soddisfa più di una?

# Esercizio: BMI

$$BMI = \frac{\text{peso}}{\text{altezza}^2}$$

- $BMI < 18.5 \Rightarrow$  sottopeso
- $18.5 \leq BMI < 25 \Rightarrow$  OK
- $25 \leq BMI < 30 \Rightarrow$  sovrappeso
- $BMI \geq 30 \Rightarrow$  obesità
- Modificare il data set attribuendo ad ogni persona la sua classificazione

# Lettura vs. Scrittura

- INFILE vs. FILE
- INPUT vs. PUT



# Data Step - copia

```
1 DATA nomipersone;  
2   SET persone;  
3   ????  
4   KEEP nome cognome;  
5 RUN;  
6
```

- Parte di data step eseguita per ogni osservazione del data set di origine
- Ciclo implicito
- variabili inizialmente mancanti

# Retain

- RETAIN mantiene il valore di una variabile per osservazioni diverse
- senza RETAIN: variabile MISSING
- con RETAIN: mantiene valore precedente
- Somma  $\Rightarrow$  RETAIN
- $a + (5 * b)$
- $a+1;$

# Sequenza di variabili

- temp1 temp2 temp3 temp4 temp5
- temp1-temp5
- Sono variabili individuali

# Array

- Elemento array = nome alternativo variabile
- `temp[1]`
- nome collettivo dell'array + indice
- `array temp[5];`
- Gli array devono essere **dichiarati** con il numero di elementi.

# Array

- `temp[1]` singolo elemento dell'array
- `temp` nome array
- `array temp[5];`
- Associa le variabili `temp1-temp5` agli elementi dell'array `temp`.
- `array temp[5] a b c d e;`
- Associa le variabili `a b c d e` agli elementi dell'array `temp`.

# Ciclo

- Ripetere più volte delle istruzioni.
- numero di volte = contatore = variabile dedicata
- Ciclo DO

```
1 array temp[5];  
2 do i=1 to 5;  
3     if temp[i]=. then temp[i]=0;  
4 end;
```

# Gestione variabili

- Il corpo di un data set viene ripetuto per ogni osservazione del data set originario.
- Il contenuto delle variabili vengono distrutte all'inizio dell'esecuzione di ogni osservazione.
- `i+1;`
- `RETAIN variabile;`

# Ordinare un data set

```
1 PROC SORT DATA=data set;  
2     BY variabile;  
3     BY DESCENDING variabile2;  
4     BY variabile1 DESCENDING variabile2;  
5 RUN;
```



# Ordinare = Partizionare

- Stesso valore = osservazioni consecutive = insieme nella partizione
- `FIRST.variabile` = prima osservazione dell'insieme
- `LAST.variabile` = ultima osservazione dell'insieme

# Uso di first e last

- Sono predicati
- Vengono utilizzati in IF

# Esempio

```
1 PROC SORT DATA=nuovometeo;  
2     BY provincia;  
3 RUN;  
4 DATA primo;  
5     SET nuovometeo;  
6     BY provincia;  
7     IF FIRST.provincia;  
8 RUN;
```

# Raggruppare osservazioni 1

- Una variabile **chiave**
- Ordinare il data set con `BY chiave`
- Gestire un contatore `i` per contare la posizione dell'osservazione corrente all'interno del gruppo.

# Raggruppare osservazioni 2

- `FIRST.chiave`: azzerare `i`
- Ogni osservazione: scrivere il dato letto nella `i`-esima posizione di un array, incrementare `i`
- `LAST.chiave`: OUTPUT

# Raggruppare osservazioni 3

Provincia	temp	i	FIRST.provincia	LAST.provincia
Milano	23			
Milano,	20			
Milano,	22			
Milano,	12			
Milano,	.			
Pavia,	24			
Pavia,	21			
Pavia,	19			
Pavia,	24			
Pavia,	21			

# Raggruppare osservazioni 3

Provincia	temp	i	FIRST.provincia	LAST.provincia
Milano	23	1	V	
Milano	20	2		
Milano	22	3		
Milano	12	4		
Milano	.	5		V
Pavia	24	1	V	
Pavia	21	2		
Pavia	19	3		
Pavia	24	4		
Pavia	21	5		V

# Calcolare statistiche

```
1 PROC MEANS DATA=data set N MEAN;  
2     VAR variabile1 variabile2;  
3 run;
```

- Seleziona variabili
- Altrimenti su tutte le variabili numeriche



# Calcolare statistiche

```
1 PROC MEANS DATA=dataset N MEAN;  
2     VAR variabile1 variabile2;  
3     CLASS variabile3;  
4 run;
```

- Stratifica le statistiche

# Calcolare statistiche

```
1 PROC MEANS DATA=dataset N MEAN;  
2     VAR variabile1 variabile2;  
3     CLASS variabile3;  
4     ID variabile3;  
5     OUTPUT OUT=nuovodat MEAN=media  
6     MAXID=massimo;  
7 run;
```

- Risultati in un dataset
- `_TYPE_` e `_FREQ_`

# Calcolare statistiche

```
1 PROC MEANS DATA=dataset NWAY N MEAN;  
2   VAR variabile1 variabile2;  
3   CLASS variabile3;  
4   ID variabile3;  
5   OUTPUT OUT=nuovodat  
6     MAXID(variabile1)=massimo  
7     MEAN(variabile1 variabile2)=media1 m2;  
8 run;
```

- NWAY: Stratificazione massima

# Statistiche

N: numero osservazioni con valore non mancante

NMISS: numero osservazioni con valore mancanti

NONOBS: numero osservazioni

MEAN: media

MEDIAN: mediana

STDDEV: deviazione standard

MAX: massimo

SUM: somma

ALPHA= .05 CLM: intervallo di confidenza

# Fondere due data set

```
1 DATA nuovods;  
2     SET vecchiods1 vecchiods2;  
3 run;
```

- Le osservazioni vengono aggiunte sequenzialmente

# Fondere due data set

```
1 DATA nuovods;  
2     MERGE vecchiods1 vecchiods2;  
3     BY comune;  
4 run;
```

- Le osservazioni sono mantenute
- Il campo comune in entrambi i data set guida la fusione.
- BY richiede un ordinamento.

# Analisi delle frequenze

```
1 PROC FREQ DATA=gare;  
2     TABLES posizioneg;  
3 RUN;
```

# Analisi delle frequenze

```
1 PROC FREQ DATA=gare;  
2     TABLES posizioneg*partenzag;  
3 RUN;
```



# Analisi delle frequenze

```
1 PROC FREQ DATA=votazioni;  
2     TABLES candidato*seggio;  
3     WEIGHT voti;  
4 RUN;
```

- WEIGHT: variabile che indica un peso per ogni osservazione
- WEIGHT: variabile quantitativa
- TABLES: variabili qualitative

# Opzioni

```
1 PROC FREQ DATA=votazioni /CHISQ;  
2     TABLES candidato*seggio;  
3     WEIGHT voti;  
4 RUN;
```

- CHISQ:  $\chi^2$
- CL: Intervalli di confidenza
- MEASURES: misure di associazione

# Output Delivery System: ODS

- Le procedure inviano dati all'ODS
- Destinazioni: LISTING (standard output), HTML, PDF, OUTPUT (data set), MARKUP (csv, xml, . . . ), DOCUMENT
- Template: tabella, stile

# Output Delivery System

```
1 ODS TRACE ON;  
2 PROC FREQ DATA=dataset;  
3   TABLES variabile;  
4 run;  
5 ODS TRACE OFF;
```

- Per avere nel log come ODS interpreta il programma

# Output Delivery System

```
1 PROC FREQ DATA=dataset;  
2   TABLES variabile;  
3   ODS SELECT CrossFreqsTab;  
4 run;
```

- Seleziona elemento
- Evitare nome duplicati (Path)
- ODS EXCLUDE

# Output Delivery System

```
1 PROC FREQ DATA=dataset;  
2   TABLES variabile;  
3   ODS OUTPUT CrossFreqsTab = dataset2;  
4 run;
```

# Output Delivery System

```
1 ODS PDF FILE = 'c:\Documents...\file.pdf';  
2 PROC FREQ DATA=dataset;  
3   TABLES variabile;  
4 run;  
5 ODS PDF CLOSE;
```

# Output Delivery System

```
1 ODS HTML FILE = 'e:\file.html';  
2 PROC FREQ DATA=dataset;  
3     TABLES variabile;  
4 RUN;  
5 ODS HTML CLOSE;
```



# Licenza d'uso

Quest'opera è distribuita con Licenza Creative Commons Attribuzione  
- Condividi allo stesso modo 4.0 Internazionale

<http://creativecommons.org/licenses/by-sa/4.0/>.

La versione più recente, con i sorgenti per modificare l'opera si trova a

<http://gianluca.dellavedova.org> e

[https://github.com/gdv/lab\\_statistico-informatico](https://github.com/gdv/lab_statistico-informatico).