SAS

STATISTICAL ANALYSIS SYSTEM

Uso di SAS per le analisi statistiche

Lezione n. 3

GENERAZIONE DI NUMERI CASUALI

· Un **numero casuale** è un numero scelto da un insieme di valori egualmente probabili, cioè un numero estratto da una distribuzione uniforme;

· in una **sequenza** di numeri casuali ogni numero estratto deve essere statisticamente indipendente dagli altri.

I numeri casuali sono utili in vari casi:

· Generazione di dati cifrati e password

· Simulazione e modellazione di fenomeni complessi

· Selezione di campioni casuali

COME GENERARE NUMERI CASUALI?

È possibile generare numeri casuali con un computer?

· Sì, ci sono 2 approcci:
o Generatori di numeri *pseudo-casuali*
o Generatori di numeri casuali veri (da fenomeni fisici, p.e. gli istanti temporali in cui accade qualcosa di totalmente imprevedibile).

GENERAZIONE DI NUMERI PSEUDO-CASUALI

Gli algoritmi per la generazione di numeri pseudo-casuali utilizzano formule matematiche o tabelle pre-calcolate per
produrre sequenze di numeri che *sembrano casuali*.
Gli algoritmi oggi disponibili sono buoni e i numeri generati sono come quelli realmente casuali.

GENERAZIONE DI NUMERI PSEUDO-CASUALI IN SAS

Le funzioni e le routine di SAS per la generazione di numeri casuali producono sequenze di numeri partendo da un valore iniziale *seed.*
Il *seed* deve essere un intero non negativo minore di 2^{31}-1.
È sempre possibile riottenere la successione di numeri casuali utilizzando lo stesso DATA step.
Se si usa il valore zero come *seed* è l'orologio di sistema che inizializza la sequenza, in tal caso la sequenza di numeri casuali non è replicabile.
Il *seed* può essere una costante intera o una variabile che contiene la costante intera. La variabile *seed* deve essere inizializzata prima della prima esecuzione della funzione o della CALL routine.

FUNZIONI SAS PER LA GENERAZIONE DI NUMERI PSEUDO-CASUALI

NORMAL normale standard
RANNOR normale standard
RANBIN binomiale
RANCAU Cauchy

RANEXP esponenziale standard
RANGAM gamma standard
RANPOI Poisson
RANTBL distribuzione discreta
RANTRI distribuzione triangolare
RANUNI uniforme (0,1)
UNIFORM uniforme (0,1)

Consideriamo la Distribuzione Uniforme in (0,1).
RANUNI FUNCTION
RANUNI(seed)

RANUNI ROUTINE
CALL RANUNI(seed,x);

The RANUNI routine gives greater control of the seed and random number streams than does the RANUNI function

```
data prova;
 Seed_1 =45; Seed_2= 45; Seed_3= 45;

 /*Seed_1 =44; Seed_2= 4; Seed_3= 4;*/

   do i=1 to 10;
      call ranuni(Seed_1,X1);
      call ranuni(Seed_2,X2);
      X3=ranuni(Seed_3);
      output;
   end;
run;

proc print;
   id i;
   var Seed_1-Seed_3 X1-X3;
run;
```

Macro Variables

Taken from a seminar
http://www.ats.ucla.edu/stat/SAS/seminars/sas_macros_introduction/default.htm

The SAS macro language is a very versatile and useful tool. It is often used to reduce the amount of regular SAS code and it facilitates passing information from one procedure to another procedure. Furthermore, we can use it to write SAS programs that are "dynamic" and flexible. Generally, we can consider macro language to be composed of macro variables and macro programs. In this session we will demonstrate how to create macro variables and how to write basic macro programs.

Macro Variables

A macro variable in SAS is a string variable that allows you to dynamically modify the text in a SAS program through symbolic substitution. The following example demonstrates how to create and use a macro variable. First we set up some system options to have a more concise output style.

```
options nodate nonumber nocenter formdlim="-";
data hsb2;
  input  id female race ses prog
         read write math scinece socst;
datalines;
 70 0 4 1 1 57 52 41 47 57
121 1 4 2 3 68 59 53 63 61
 86 0 4 3 1 44 33 54 58 31
141 0 4 3 3 63 44 47 53 56
172 0 4 2 2 47 52 57 53 61
113 1 4 2 2 44 52 51 63 61
 50 0 3 2 1 50 59 42 53 61
 11 0 1 2 2 34 46 45 39 36
 84 0 4 2 1 63 57 54 51 63
 48 1 3 2 2 57 55 52 50 51
 75 1 4 2 3 60 46 51 53 61
 60 1 4 2 2 57 65 51 63 61
 95 0 4 3 2 73 60 71 61 71
104 0 4 3 2 54 63 57 55 46
 38 0 3 1 2 45 57 50 31 56
115 0 4 1 1 42 49 43 50 56
 76 0 4 3 2 47 52 51 50 56
195 0 4 2 1 57 57 60 56 52
;
run;
```

Suppose that we want to look at the means of some variables and then do a regression analysis on the same variables.

```
proc means data = hsb2;
  var write math female socst;
run;

proc reg data = hsb2;
  model read = write math female socst;
run;

quit;
```

We can simplify the program by creating a macro variable containing all the names of the independent variables. A macro variable can be created by using the **%let** statement. All the key words in statements that are related to macro variables or macro programs are preceded by percent sign %; and when we reference a macro variable it is preceded by an ampersand sign &. When we submit our program, SAS will process the macro variables first, substituting them with the text string they were defined to be and then process the program as a standard SAS program.

```
%let indvars = write math female socst;
proc means data = hsb2;
  var &indvars;
run;

proc reg data = hsb2;
  model read = &indvars;
run;
quit;
```

We can display macro variable value as text in the log window by using **%put** statement.

```
%put my first macro variable indvars is &indvars;
```