

Esercizi di MERGE

Esercizio 1

Si considerino i seguenti data set ELEZIONE e CANDIDAT. Il primo data set contiene il nome del vincitore e del perdente per ogni elezione presidenziale negli USA dal 1960 al 1984, mentre il secondo contiene i nomi dei candidati dei maggiori partiti, il loro partito politico e lo Stato di origine di ciascuno. Rispondete alle seguenti domande utilizzando i due data set:

1. per ogni anno di elezione, trovare il partito e lo Stato di appartenenza del candidato perdente;
2. per ogni candidato, trovare l'ultima elezione cui ha partecipato e indicare se ha vinto o perso;
3. per ogni candidato, contare il numero di volte in cui ha vinto una elezione;
4. trovare i candidati che sono stati sia perdenti che vincenti.

DATA SET ELEZIONE

| OBS | ANNO | VINTO | PERSO |
|-----|------|---------|----------|
| 1 | 1960 | KENNEDY | NIXON |
| 2 | 1964 | JOHNSON | GOLDWATE |
| 3 | 1968 | NIXON | HUMPHREY |
| 4 | 1972 | NIXON | MCGOVERN |
| 5 | 1976 | CARTER | FORD |
| 6 | 1980 | REAGAN | CARTER |
| 7 | 1984 | REAGAN | MONDALE |

DATA SET CANDIDAT

| OBS | NOME | PARTITO | STATO |
|-----|----------|---------|-------|
| 1 | NIXON | R | CALIF |
| 2 | GOLDWATE | R | ARIZ |
| 3 | KENNEDY | D | MASS |
| 4 | JOHNSON | D | TEXAS |
| 5 | HUMPHREY | D | MINN |
| 6 | MCGOVERN | D | S_DAK |
| 7 | FORD | R | MICH |
| 8 | CARTER | D | GA |
| 9 | MONDALE | D | MINN |
| 10 | REAGAN | R | CALIF |

Esercizio 2

Un file contiene i record relativi ad un certo insieme di individui per i quali sono stati registrati i seguenti dati: nome, titolo di studio, età, sesso, nome del padre e della madre. Il file è riportato in Tab. 1.

1. Si costruisca una tabella a doppia entrata relativa all'incrocio tra titolo di studio del padre e della madre;
2. Si costruisca una tabella a doppia entrata relativa all'incrocio tra età della madre all'ultimo figlio e sesso dell'ultimo figlio.

Tabella 1

| <i>Nome</i> | <i>Padre</i> | <i>Madre</i> | <i>Titolo</i> | <i>Età</i> | <i>sesso</i> |
|-------------|--------------|--------------|---------------|------------|--------------|
| Mary | | | C | 45 | F |
| Joe | | | HS | 50 | M |
| Alice | Joe | Mary | C | 33 | F |
| Jim | Joe | Mary | HS | 32 | M |
| Bruce | Joe | Mary | EL | 12 | M |
| John | | | EL | 58 | M |
| Carla | | | HS | 56 | F |
| Will | Jhon | Carla | HS | 38 | M |
| Beth | Will | Alice | EL | 12 | F |
| Al | Will | Alice | EL | 13 | M |

Utilizzo della opzione IN=

Una ditta produttrice di articoli sportivi vuole inviare un depliant pubblicitario a tutti i clienti che non hanno effettuato ordini durante l'anno in corso. L'azienda dispone di due archivi di dati, uno contenente l'elenco dei clienti e l'altro l'elenco degli ordini.

Al fine di ottenere l'elenco dei clienti che non hanno effettuato ordini è necessario unire i due archivi utilizzando l'opzione IN=, e selezionare i clienti che non hanno osservazioni nell'archivio ordini.

L'archivio clienti contiene le seguenti variabili:

CODICE CLIENTE

NOME

INDIRIZZO

L'archivio ordini contiene una osservazione per ogni ordine effettuato nell'anno in corso e, per ciascun ordine, le variabili:

CODICE CLIENTE

IMPORTO ORDINE

SOLUZIONI

Esercizio 1

(merge.sas)

```
*eser1_13_5_05;
*soluzione esercizi di merge 13 maggio 2005;
titlel 'Esercizio 1';
DATA ELEZIONE;
input anno vinto $ perso $;
cards;
1960      KENNEDY      NIXON
1964      JOHNSON      GOLDWATE
1968      NIXON        HUMPHREY
1972      NIXON        MCGOVERN
1976      CARTER       FORD
1980      REAGAN       CARTER
1984      REAGAN       MONDALE
;
run;
DATA CANDIDATI;
input nome $ partito $ stato $;
cards;
NIXON      R      CALIF
GOLDWATE   R      ARIZ
KENNEDY    D      MASS
JOHNSON    D      TEXAS
HUMPHREY   D      MINN
MCGOVERN   D      S_DAK
FORD       R      MICH
CARTER     D      GA
MONDALE    D      MINN
REAGAN     R      CALIF
;
run;
```

*1. per ogni anno di elezione, trovare il partito e lo Stato di appartenenza del candidato

perdente;

```
proc sort data=elezione;
```

```
by perso;
```

```
run;
```

```
proc sort data=candidati;
```

```
by nome;
```

```
run;
```

```
data perdenti;
```

```
merge elezione(keep=anno perso
```

```
rename=(perso=nome) in=p) candidati;
```

```
by nome;
```

```
if p;
```

```
run;
```

```
title2 'Partito e stato di appartenenza  
del candidato perdente';
```

```
proc sort data=perdenti;
```

```
by anno;
```

```
proc print data=perdenti;
```

```
run;
```

```

*2. per ogni candidato, trovare
l'ultima elezione cui ha partecipato
e indicare se ha vinto o perso;
*crea archivio con un record per
candidato vincente (vinto=1) o perdente
(vinto=0);
data espandi(keep=nome anno vince);
set elezione;
nome=vinto; vince=1; output;
nome=perso; vince=0; output;
label vince='1 vincente';
run;
*ordina archivio per nome e anno
elezione;
proc sort data=espandi;
by nome anno;
run;
title2 'stampa archivio ordinato per
candidato: un record per ogni elezione
cui ha partecipato';
proc print data=espandi;
run;
*per ogni candidato prende solo ultima
osservazione;
data ultima;
set espandi;
by nome;
if last.nome then output;
run;

```

```
title2 'Candidati per ultima elezione  
cui hanno partecipato (vinta se  
vince=1)';  
proc print data=ultima;  
id nome;  
run;
```

*3. per ogni candidato, contare il numero di volte in cui ha vinto una elezione;

```
data vittorie;
```

```
set expandi;
```

```
by nome;
```

```
if first.nome then do;
```

```
    vittorie=0; elezioni=0;
```

```
end;
```

```
elezioni+1;
```

```
vittorie+vince;
```

```
label vittorie='numero di vittorie'
```

```
elezioni='numero candidature';
```

```
if last.nome then output;
```

```
drop vince anno;
```

```
run;
```

```
title2 'Numero di vittorie e candidature  
per candidato';
```

```
proc print data=vittorie;
```

```
id nome;
```

```
run;
```

*4. trovare i candidati che sono stati sia perdenti che vincenti;

```
proc print data=vittorie;  
where vittorie ne 0 and vittorie  
ne elezioni;  
run;
```

*modo alternativo se non si dispone del data set vittorie;

```
data doppi;  
set expandi;  
by nome;  
if first.nome then do;  
    vittorie=0; perdite=0;  
end;  
perdite+(1-vince);  
vittorie+vince;  
label vittorie='numero ddi vittorie'  
perdite='numero candidature';  
if last.nome and perdite ne 0 and  
vittorie ne 0 then output;  
drop vince anno;  
run;  
title2 'Candidati che hanno sia vinto  
che perso';  
proc print data=doppi;  
id nome;
```

`run ;`

Esercizio 2

```
*eser2_13_5_05;
*soluzione esercizi di merge dati a
lezione 13 maggio 2005;
options nocenter nodate nonumber;
title1 'Esercizio 2 (testo in merge.doc
dato a lezione)';
data dati;
input nome $7. +3 padre $7.  madre $7.
titolo $  eta sesso $;
cards;
Mary                C  45  F
Joe                 HS  50  M
Alice               Joe   Mary  C  33  F
Jim                 Joe   Mary  HS 32  M
Bruce               Joe   Mary  EL 12  M
John                EL  58  M
Carla               HS  56  F
Will                John   Carla HS 38  M
Beth                Will   Alice EL 12  F
Al                  Will   Alice EL 13  M
;
run;
*ordina archivio dati per nome;
proc sort data=dati;
by nome;
title2 'stampa archivio dati ordinato
per nome';
proc print data=dati;
run;
```

*1. Si costruisca una tabella a doppia entrata relativa all'incrocio tra titolo di studio del padre e della madre;

*associa a padri e madri il loro titolo di studio;

```
proc sort data=dati out=padri;
by padre;
run;
data uniti;
merge padri(in=p) dati(keep=nome titolo
rename=(nome=padre titolo=tit_padre));
by padre;
if p;
run;
proc sort data=uniti out=madri;
by madre;
run;
title2 'stampa archivio con titolo
studio padre e madre';
data uniti;
merge madri(in=p) dati(keep=nome titolo
rename=(nome=madre titolo=tit_madre) );
by madre;
if p;
run;
proc print data=uniti;
run;
```

```
title2 'tabella titolo padre per titolo  
madre';  
proc freq data=uniti;  
tables tit_padre*tit_madre;  
run;
```

*2. Si costruisca una tabella a doppia entrata relativa all'incrocio tra età della madre all'ultimo figlio e sesso dell'ultimo figlio.;

```
proc sort data=dati out=madri;
by madre;
*prende ultimo figlio per ogni madre e
associa info eta madre;
data ultimo(keep=madre eta_ultimo
eta_madre eta_nascita sesso_ultimo);
merge madri (where=(madre ne ' ') in=p)
dati(keep=nome eta rename=(nome=madre
eta=eta_madre));
by madre;
*calcola età della madre alla nascita
dell'ultimo figlio;
eta_nascita=eta_madre-eta;
if p;
if last.madre then output;
rename eta=eta_ultimo
sesso=sesso_ultimo;
label eta_ultimo='età ultimo figlio'
eta_madre='età attuale madre'
eta_nascita='età madre alla nascita
figlio';
run;
title2 'madri per età e sesso
dell''ultimo figlio';
proc print data=ultimo;
run;
```

```
*tabella;  
proc freq data=ultimo;  
table eta_nascita*sexo_ultimo;  
run;
```

Esercizio 3

```
DATA CLIENTI;  
    INFILE 'C:\CLIENTI.DAT';  
    INPUT CODICE NOME $ 5-21 INDIRIZZO $23-42;  
RUN;  
DATA ORDINI;  
    INFILE 'C:\ORDINI.DAT';  
    INPUT CODICE IMPORTO;  
RUN;  
PROC SORT DATA=CLIENTI;  
BY CODICE;  
RUN;  
PROC SORT DATA=ORDINI;  
BY CODICE;  
RUN;
```

*Combina i data set per individuare clienti senza ordini;

```
DATA NOORD;  
    MERGE CLIENTI ORDINI(IN=recenti);  
    BY CODICE;  
    IF RECENTI=0;  
RUN;  
TITLE 'CLIENTI SENZA ORDINI RECENTI';  
PROC PRINT DATA=NOORD;  
RUN;
```