Combinarea tabelelor SAS

rodica.lung@econ.ubbcluj.ro

19 decembrie 2016

Moduri de combinare:

- one to one: se creeaza observatii care contin toate variabilele in fiecare tabel folosind `set`;
- concatenare: se ataseaza tabelele unul dupa altul; se foloseste `set`;
- intercalare bazata pe variabile comune; se foloseste `set,by`;
- match-merging: potriveste observatii din doua sau mai multe tabele intr-una singura pe baza valorilor comune ale unei variabile.

Observatie

Pentru combinarea tabelelor se poate folosi si `proc sql`

Citire one-to-one

- Combinarea `one-to-one` este utila in cazurile in care observatiile sunt ordonate dupa un `ID` sau cheie unica si organizate in asa fel incat campurile celor doua tabele sa se suprapuna.
- se face prin mai multe instructiuni `set` cu care se pot citi tabele diferite sau acelasi tabel de mai multe ori;

Forma generala:

```
DATA output-SAS-data-set;
SET SAS-data-set-1;
SET SAS-data-set-2;
RUN;
```

Exemplu

Fie tabelele SAS `C` si `D`:

Tabela: Tabel C

Num	VarA
1	A1
3	A2
5	A3

Tabela: Tabel D

Num	VarA
2	B1
4	B2

si

```
data one2one;
set c;
set d;
run;
```

La executarea codului, pasul `data` va trece prin urmatorii pasi , rezultand tabelul de jos:

Tabela: Pasii parcursi `one-to-one`

Pas	Variabile in noul tabel		
noile variabile	Num	VarA	VarB
primul `set`	1	A1	
al doilea `set`	2	A1	B1
din nou primul `set`	3	A2	
din nou al doilea `set`	4	A2	B2

Tabela: Tabel `one2one`

Num	VarA	VarB
2	A1	B1
4	A2	B2

Concatenarea

- tabelele sunt asezate unul dupa altul;
- noul tabel va contine toate variabilele din toate tabelele concatenate.

- tabelele de concatenat se enumera in aceeasi instructiune `set`;
- concatenarea tabelelor se poate face si cu `proc append`.

Exemplu

```
data concat;
set C D;
run;
```

Rezultatul:

Tabela: Tabel `concat`

Num	VarA	VarB
1	A1	
3	A2	
5	A3	
2		B1
4		B2

Se observa ca:

- variabila comuna celor doua tabele, `num` are acelasi tip in ambele tabele
- este obligatoriu pentru toate variabilele care au acelasi nume in toate tabelele, altfel se emite eroare
- daca variabilele cu acelasi nume au lungimi diferite, SAS atribuie in noul tabel lungimea primei variabile intalnite (in primul tabel in care apare)
- si procedeaza la fel cu etichetele, formate sau informate, adica ia prima varianta intalnita in primul tabel.

Exemplu

```
data clinic.concat;
set clinic.therapy1999 clinic.therapy2000;
run;
```

Intercalarea

Daca in pasul `data` in care se face concatenare se foloseste o instructiune `by` atunci va rezulta o intercalare bazata pe variabilele din lista `by`. Forma generala a pasului `data` pentru intercalare este:

```
DATA output-SAS-data-set;
SET SAS-data-set-1 SAS-data-set-2;
BY variable(s);
RUN;
```

unde `variable(s)` indica variabilele dupa care sa se faca intercalarea.

!

TOATE TABELELE DIN LISTA `SET` TREBUIE SA FIE ORDONATE DUPA VARIABILELE DIN LISTA `BY`

Exemplu:

```
data interlv;
set c d;
by num;
run;
```

Tabela: Tabel C

Num	Var
1	C1
2	C2
2	C3
3	C4

Tabela: Tabel D

Num	VarA
2	D1
3	D2
3	D3

Tabela: Tabel `concat`

Num	VarA
1	C1
2	C2
2	C3
2	D1
3	C4
3	D2
3	D3

Alt exemplu

```
data clinic.interlv;
set clinic.therapy1999 clinic.therapy2000;
by month;
run;
```

Match-merging

se combina mai multe tabele pe baza unei variabile comune; se foloseste instructiunea `merge` in loc de `set`.

Forma generala:

```
DATA output-SAS-data-set;
MERGE SAS-data-set-1 SAS-data-set-2;
BY <DESCENDING> variable(s);
RUN;
```

- `output-SAS-data-set` denumeste tabelul nou creat de pasul `data`;
- `SAS-data-set-1,2` - tabelele SAS din care se citesc datele;
- `variable(s)` in `by` specifica una sau mai multe variabile dupa care se unesc observatiile
- `<descending>` indica faptul ca datele sursa sunt ordonate in ordine descrescatoare dupa variabila care urmeaza; daca in lista `by` sunt mai multe variabile, `<descending>` are efect doar asupra variabilei care urmeaza imediat dupa ea.;
- tabelele din instructiunea `by` trebuie sa fie sortate inainte de executarea pasului `data`;
- variabilele din `by` trebuie sa aiba acelasi tip in toate tabelele unite de instructiunea `merge`;
- `descending` nu se poate folosi cu tabele indexate pentru ca acestea sunt intotdeauna ordonate crescator.

Tabela: Tabel A

Num	VarA
1	A1
2	A2
3	A3

Tabela: Tabel B

Num	VarB
1	B1
2	B2
4	B3

Dupa rularea codului:

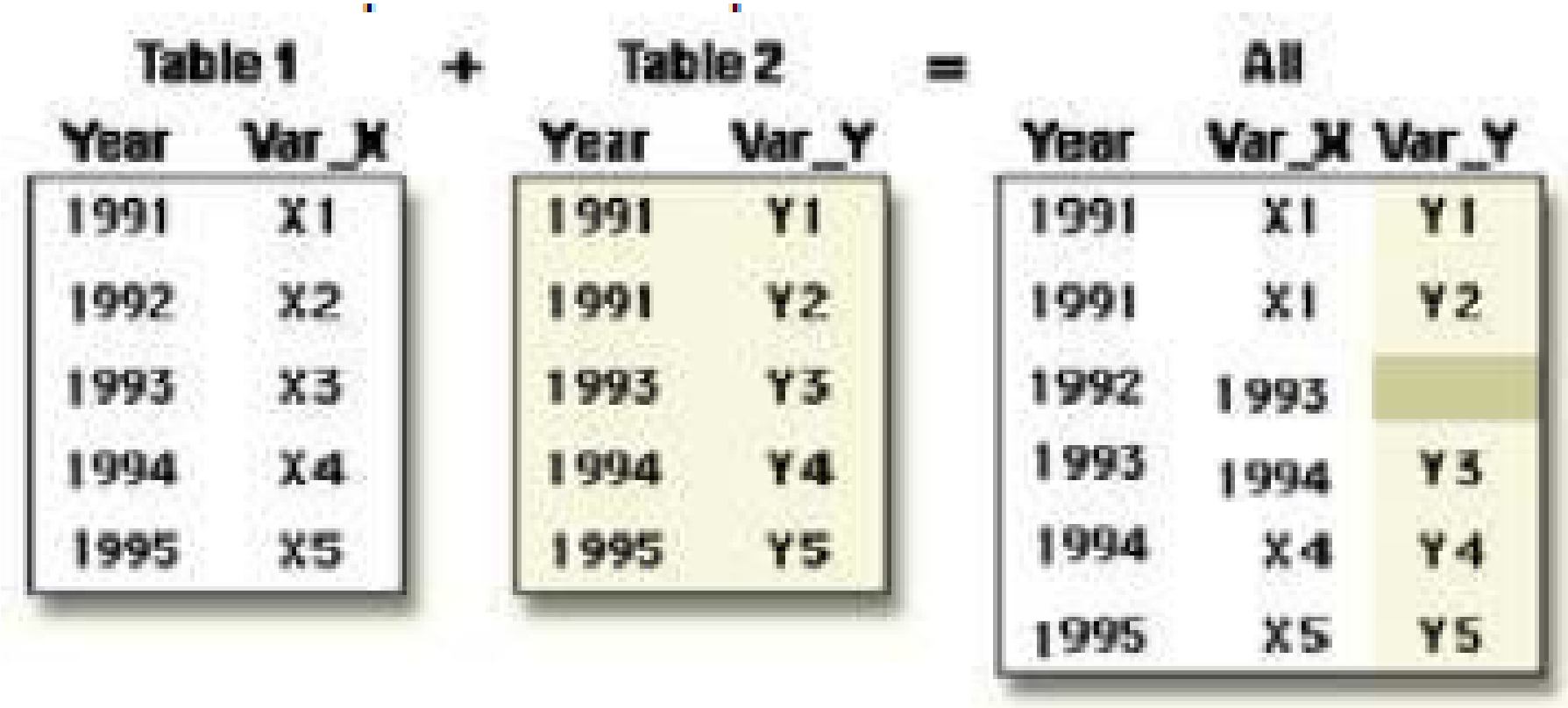
```
data merged;
merge a b;
by num;
run;
```

Tabela: Tabel `merged`

Num	VarA	VarB
1	A1	B1
2	A2	B2
3	A3	
4		B3

In tabelul rezultat vor aparea toate observatiile din toate tabelele sursa
Se pot adauga instructiuni si optiuni pentru selectarea observatiilor dorite.
Daca unul din tabelele de intrare nu contine observatii pentru o anumita valoare a variabilei `by` in tabelul rezultat acestea se completeaza cu `missing`.

Figura: Alt exemplu de match-merge



```
proc sort data=clinic.demog;
by id;
run;
proc print data=clinic.demog;
run;
```

va afisa tabelul `demog`:

Obs	ID	Age	Sex	Date
1	A001	21	m	05/22/75
2	A002	32	m	06/15/63
3	A003	24	f	08/17/72
4	A004	.		03/27/69
5	A005	44	f	02/24/52
6	A007	39	m	11/11/57

```
proc sort data=clinic.visit;
by id;
run;
proc print data=clinic.visit;
run;
```

va afisa tabelul `visit`:

Obs	ID	Visit	SysBP	DiasBP	Weight	Date
1	A001	1	140	85	195	11/05/98
2	A001	2	138	90	198	10/13/98
3	A001	3	145	95	200	07/04/98
4	A002	1	121	75	168	04/14/98
5	A003	1	118	68	125	08/12/98
6	A003	2	112	65	123	08/21/98
7	A004	1	143	86	204	03/30/98
8	A005	1	132	76	174	02/27/98
9	A005	2	132	78	175	07/11/98
10	A005	3	134	78	176	04/16/98
11	A008	1	126	80	182	05/22/98

iar unite cu `merge`:

```
data clinic.merged;
merge clinic.demog clinic.visit;
by id;
run;
proc print data=clinic.merged;
run;
```

Obs	ID	Age	Sex	Date	Visit	SysBP	DiasBP	Weight
1	A001	21	m	11/5/1998	1	140	85	195
2	A001	21	m	10/13/1998	2	138	90	198
3	A001	21	m	7/4/1998	3	145	95	200
4	A002	32	m	4/14/1998	1	121	75	168
5	A003	24	f	8/12/1998	1	118	68	125
6	A003	24	f	8/21/1998	2	112	65	123
7	A004	.		3/30/1998	1	143	86	204
8	A005	44	f	2/27/1998	1	132	76	174
9	A005	44	f	7/11/1998	2	132	78	175
10	A005	44	f	4/16/1998	3	134	78	176
11	A007	39	m	11/11/1957	.	.	.	
12	A008	.		5/22/1998	1	126	80	182

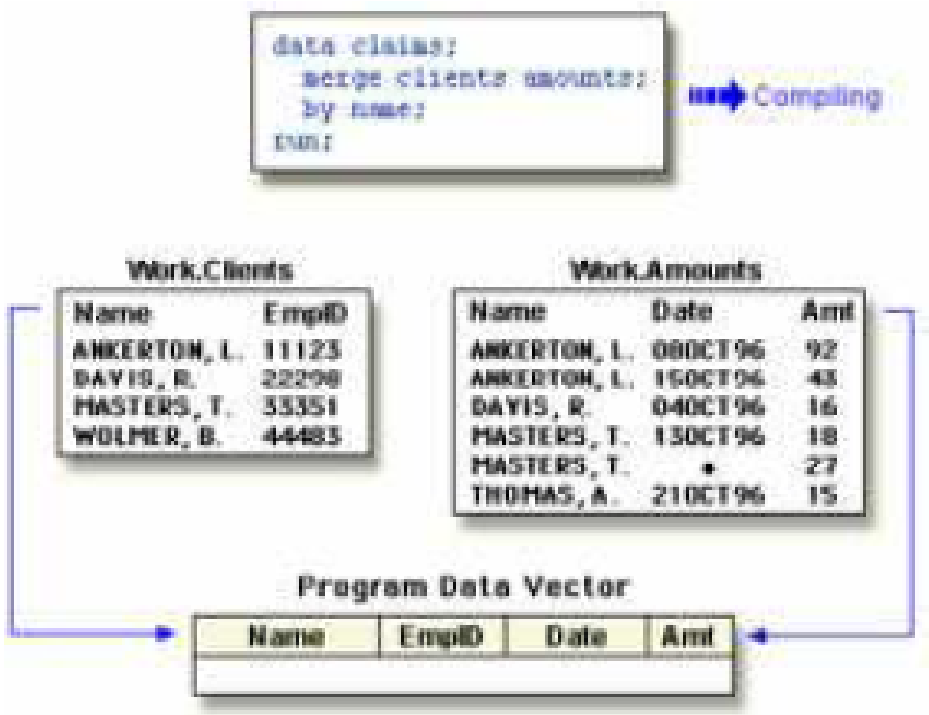
Date ordonate descrescator

```
proc sort data=clinic.demog;
by descending id;
run;
proc sort data=clinic.visit;
by descending id;
run;
data clinic.merged;
merge clinic.demog clinic.visit;
by descending id;
run;
proc print data=clinic.merged;
run;
```

Obs	ID	Age	Sex	Date	Visit	SysBP	DiasBP	Weight
1	A008	.		5/22/1998	1	126	80	182
2	A007	39	m	11/11/1957	.	.	.	
3	A005	44	f	2/27/1998	1	132	76	174
4	A005	44	f	7/11/1998	2	132	78	175
5	A005	44	f	4/16/1998	3	134	78	176
6	A004	.		3/30/1998	1	143	86	204
7	A003	24	f	8/12/1998	1	118	68	125
8	A003	24	f	8/21/1998	2	112	65	123
9	A002	32	m	4/14/1998	1	121	75	168
10	A001	21	m	11/5/1998	1	140	85	195
11	A001	21	m	10/13/1998	2	138	90	198
12	A001	21	m	7/4/1998	3	145	95	200

Procesarea `match-merge`

In faza de compilare se construieste PDV (structura noului tabel). Variabilele care au acelasi nume in tabelele sursa vor aparea o singura data in noul tabel; lungimea lor e data de prima aparitie din lista `merge`; valoarea va fi din ultimul tabel citit.



Dupa crearea PDV, in faza aceasta se atribuie cate un pointer de urmarire fiecarul tabel sursa.

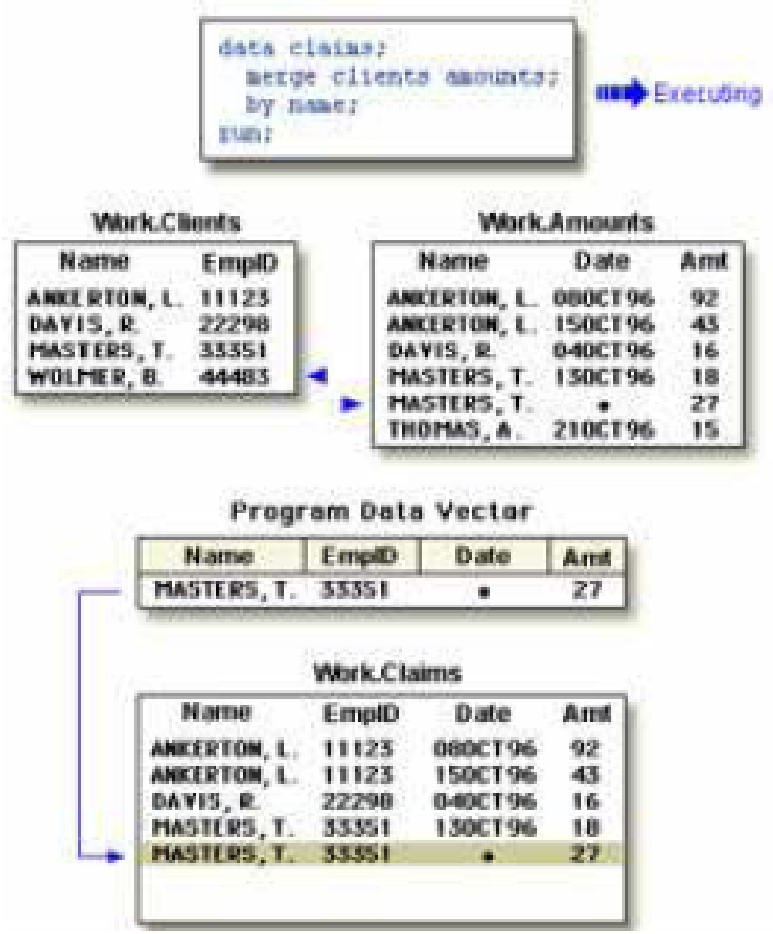
In faza de executie SAS verifica secvential daca observatiile curente din tabelele sursa au aceeasi valoare pentru variabilele din `by`:

- daca DA, atunci observatiile sunt scrise in PDV in ordinea in care apar tabelele pe lista `merge`. Valorile din variabilele cu acelasi nume sunt scrise una peste alta intr-o singura variabila. SAS scrie observatia combinata in tabelul nou si retine valoarile in PDV pana cand valoarea din `by` se schimba in toate tabelele.
- daca NU, SAS determina care valori sunt primele si scrie observatia respectiva in PDV dupa care continutul PDV se scrie in tabel.

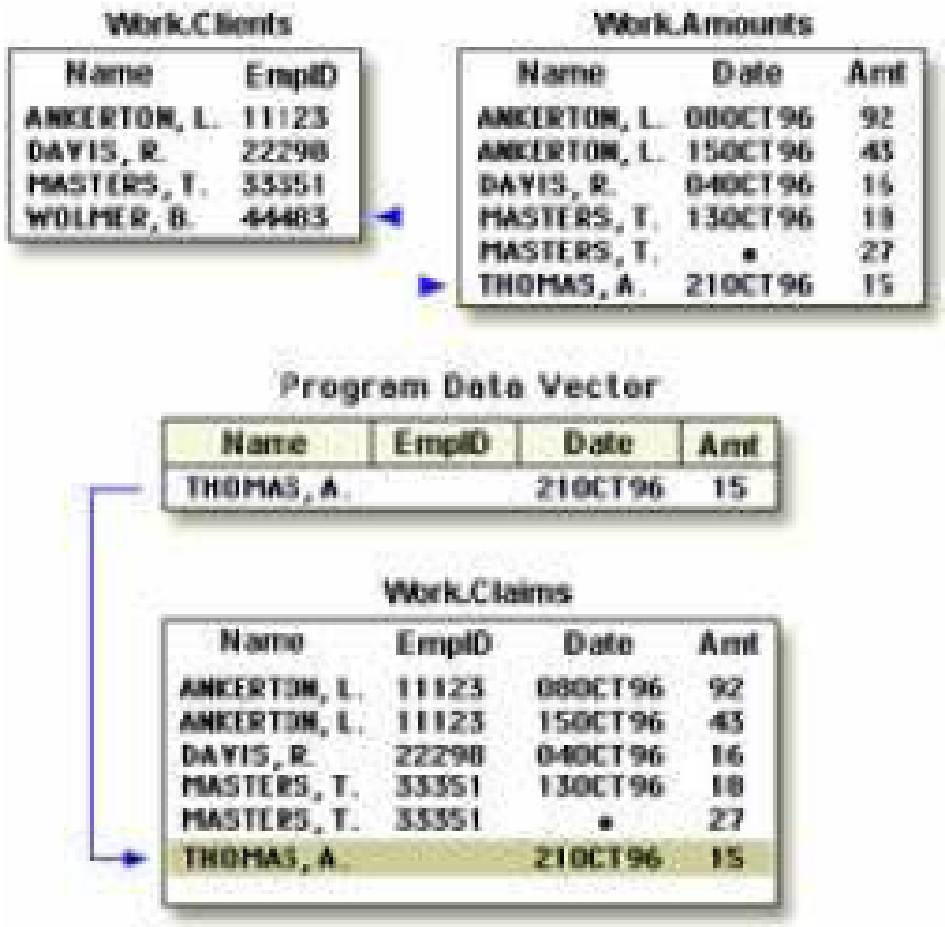
Cand se schimba valoarea din `by` in toate tabelele sursa, PDV e reinitializat cu `missing`.

Tratarea observatiilor fara pereche si a valorilor `missing`

daca o observatie are valori `missing` acestea sunt copiate in noul tabel. Observatiile care au valori `missing` pentru variabilele din lista `by` vor aparea la inceputul tabelului.



- daca o observatie nu are valoare a variabilei `by` distincta de toate celalalte, in tabelul rezultat vor fi trecute valori `missing` pentru valorile lipsa.



Re-denumirea variabilelor

- cand nu vrem sa suprapunem variabilele cu acelasi nume din tabele diferite.

Forma generala:

(RENAME=(old-variable-name=new-variable-name))

unde

- optiunea `rename=` in paranteze urmeaza numele tabelului SAS sursa care contine variabilele care trebuier re-denumite;
- `old-variable-name` specifica variabila care trebuie redenumita
- se pot redenumi mai multe variabile intr-o optiune `rename=`;
- `new-variable-name` specifica noul nume al variabilei
- `rename=` se poate folosi si in instructiunea `set`.

Exemplu:

```
data clinic.merged;
merge clinic.demog(rename=(date=BirthDate))
clinic.visit(rename=(date=VisitDate));
by id;
run;
proc print data=clinic.merged;
run;
```

Excluderea observatiilor fara pereche

-cand ne intereseaza doar observatiile care au corespondent in toate tabelele; pentru a exclude observatiile care nu au corespondent din tabelul rezultat se foloseste optiunea `in=` si `if` pentru selectare.
`in=` creeaza o variabila temporara care indica daca tabelul a contribuit la observatia curenta sau nu. `if` verifica valorile variabilelor `in=` si alege doar variabilele care apar in ambele tabelel.

Exemplu:

toate optiunile se put in aceeasi paranteza:

```
data clinic.merged;
merge clinic.demog(in=indemog
rename=(date=BirthDate))
clinic.visit(in=invisit
rename=(date=VisitDate));
by id;
if indemog=1 and invisit=1;
run;
proc print data=clinic.merged;
run;
```

Selectarea variabilelor

Se face tot cu `drop` si `keep`.

```
data clinic.merged(drop=id);
merge clinic.demog(in=indemog
rename=(date=BirthDate))
clinic.visit(drop=weight in=invisit
rename=(date=VisitDate)) ;
by id;
if indemog and invisit;
run;
proc print data=clinic.merged;
run;
```

`drop` se foloseste ca si optiune la `merge` daca nu se doreste procesarea variabilelor respective in nici un fel; daca valorile variabilelor apar in pasul `data` dar nu se doreste copierea lor in noul tabel, `drop` trebuie sa apara ca si optiune a instructiunii `data`