Citirea tabelelor SAS

Obiective

- crearea unui tabel SAS nou pe baza unuia existent;
- procesarea cu grupuri `by`;
- citirea observatiilor dupa numere;
- oprirea procesarii in caz de nevoie;
- scrierea observatiilor explicit in output;
- identificarea ultimei observatii din tabel;

Citirea dintr-un singur tabel

Se face intr-un pas `data` de forma:

```
DATA tabel-nou-creat;
SET tabel-din-care-copiem;
RUN;
```

Exemplu: pasul `data` creaza tabelul `drug1h` in biblioteca `lab23` copiind toate observatiile din tabelul `cltrials` din biblioteca `research`

```
libname lab23 '...';
libname research '...';
data lab23.drug1h;
set research.cltrials;
run;
```

Tabela: Actiuni

Pentru a	Se foloseste ceva de genul:
Selecta observatii	`if resthr<70 then delete;` `if tolerance='D';`
Renunta la niste variabile	`drop timemin timesec;`
Crea sau modifica o variabila	`TotalTime=(timemin*60)+timesec;`
Initializa o variabila sum	`retain SumSec 5400;`
Insuma valori acumulate	`sumsec+totaltime;`
declara lungimea	`length TestLength $ 6;`
Conditii	`if totaltime>800 then TestLength='Long';`

	`else if 750<=totaltime<=800`
	`then TestLength='Normal';`
	`else if totaltime<750`
	`then TestLength='Short';`

etichete	`label sumsec='Cumulative Total Seconds';`
Formata pentru afisare	`format sumsec comma6.;`

Exemplu

```
 data lab23.drug1h(drop=placebo uric);
set research.cltrials(drop=triglyc);
if sex='M' then delete;
if placebo='YES';
TestDate='22MAY1999'd;
retain Days 30;
days+1;
length Retest $ 5;
if cholesterol>190 then retest='YES';
else if 150<=cholesterol<=190 then retest='CHECK';
else if cholesterol<150 then retest='NO';
label retest='Perform Cholesterol Test 2?';
format enddate mmddyy10.;
run;
```

drop si keep

au efecte diferite daca se folosesc in instructiunea `data` sau `set`:

- daca nu se doreste in nici un fel procesarea unor variabile atunci `drop` se poate folosi ca optiunea a lui `set`.
- Exemplu: renuntarea completa la variabilele `triglycerides` si `uricacid`

```
data lab23.drug1h;
set research.cltrials(drop=triglycerides
uricacid);
if placebo='YES';
run;
```

drop si keep

- daca se doreste renuntarea la o variabila, dar folosirea ei in timpul procesarii pasului `data` atunci optiunea `drop` trebuie sa apara in instructiunea `data`, ca in exemplul de mai jos
- variabila `placebo` nu va aparea in `drug1h` dar e folosita pentru selectarea observatiilor in instrucitunea `if`:

```
data lab23.drug1h(drop=placebo);
set research.cltrials(drop=triglycerides
uricacid);
if placebo='YES';
run;
```

- Atunci cand `drop` e folosit cu instructiunea `data`, se renunta doar la variabilele din lista, dar acestea sunt citite din tabelul sursa.

Procesarea cu by

Instructiunea by, folosita intr-un pas data conduce la procesarea observatiilor grupate dupa o variabila. De exemplu:

```
data temp;
set usa;
by dept;
run;
```

Atunci cand by este folosita cu set:

- tabelul mentionat in set trebuie sortat dupa variabila din by;
- pasul data creeaza doua variabile temporare pentru fiecare variabila by: first.variable si last.variable care iau valori 0 sau 1 si identifica prima si ultima observatie din grupul by. first.variable are valoarea 0 pentru prima observatie din grupul by si zero pentru oricare alta. last.variable are valoarea 1 pentru ultima observatie din grupul by si 0 pentru oricare alta.

Exemplu:

Tabelul `usa` contine informatii despre salarii ale anajatilor din diferite departamente. Sunt doua categorii de salarii ("S" - lunar si "H" - plata cu ora). Pentru plata cu ora se considera 2000 de ore lucrate anual. Se cere sa se calculeze totalul anual al salariilor pentru fiecare departament intr-un tabel nou numit `budget`.

-la fiecare inceput de grup noi `payroll` se initializeaza la 0; `if last.dept` selecteaza ultima observatie din grup pentru a fi copiata in `budget` cu totalul de la `payroll`

```
proc sort data=company.usa out=work.temp;
by dept;
run;
data company.budget(keep=dept payroll);
set work.temp;
by dept;
if wagecat='S' then Yearly=wagerate*12;
else if wagecat='H' then Yearly=wagerate*2000;
if first.dept then Payroll=0;
payroll+yearly;
if last.dept;
run;
```

Pentru a vizualiza rezultatele si a calcula totalul:

```
proc print data=company.budget noobs;
sum payroll;
format payroll dollar12.2;
run;
```

Cand in `by` se specifica mai multe variable:

- `first.variable` este setata la 1 pentru fiecare valoare noua a fiecarei variabile;
- o schimbare de valoare a primei variabile `by` conduce la setarea `last.variable` la 1 pentru toate cele care ii urmeaza pe lista `by`

Exemplu: Pentru acelasi tabel `usa` dorim sa calculam totalul anual al salariilor pentru fiecare tip de pozitie si pentru fiecare manager in parte. In program vom specifica doua variabile `by`:

```
proc sort data=company.usa out=work.temp2;
by manager jobtype;
data company.budget2(keep=manager jobtype payroll);
set work.temp2;
by manager jobtype;
if wagecat='S' then Yearly=wagerate*12;
else if wagecat='H' then Yearly=wagerate*2000;
if first.jobtype then Payroll=0;
payroll+yearly;
if last.jobtype;
run;
```

Pentru afisare:

se afiseaza subtotaluri si totaluri pentru cei doi manageri

```
proc print data=company.budget2 noobs;
by manager;
var jobtype;
sum payroll;
where manager in ('Coxe','Delgado');
format payroll dollar12.2;
run;
```

Citirea observatiilor prin acces direct

Pana acum observatiile au fost citite secvential, adica in ordinea in care apareau in sursa citita. In SAS observatiile pot fi accesate si direct folosind optiunea `point` in instructiunea `set`. Forma generala este:

```
POINT=variable;
```

unde `variable`

- indica o variabila numerica temporara care contine numarul observatiei de citit;
- trebuie initializata inainte de executia instructiunii `set`.

Exemplu: Sa spunem ca vrem sa citim observatia a 5-a dintr-un tabel:
de fapt nu asa!

```
data work.getobs5;
obsnum=5;
set company.usa(keep=manager payroll) point=obsnum;
run;
```

dar acest program ar conduce la un ciclu infinit datorat faptului ca pasul `data` se executa pana se ajunge la marcajul de sfarsit de fisier.

Solutii:

- Folosirea instructiunii `stop` care opreste executarea pasului `data` si sare la urmatorul pas din program:

```
dar nici asa nu e de ajuns :)
data work.getobs5(drop=obsnum);
obsnum=5;
set company.usa(keep=manager payroll)
point=obsnum;
stop;
run;
```

- programarea unei conditii care verifica o valoare invalida pentru variabila `point`, conducand la o erroare si oprirea procesarii pasului `data`.

Pentru ca in ambele cazuri prezentate mai sus opresc executia pasului `data` abrupt, observatiile nu se vor scrie in output in aceasta varianta, deci mai e nevoie de inca o etapa.

Scrierea explicita a observatiilor

se poate face cu o instructiune `output`. Folosirea `output` modifica modul in care `data` scrie observatiile, deci toate observatiile trebuie scrise cu aceasta metoda (nu se pot amesteca). Forma generala:

```
output <tabelSAS>;
```

unde `tabelSAS` denumeste tabelul SAS in care se vor scrie observatiile, care trebuie sa apara si in instructiunea `data`. Daca nu e mentionat nici un tabel observatiile se vor scrie in toate tabelele care apar in instructiunea `data`.

Exemplul de mai sus devine:

```
data work.getobs5(drop=obsnum);
obsnum=5;
set company.usa(keep=manager payroll) point=obsnum;
output;
stop;
run;
proc print data=work.getobs5 noobs;
run;

data empty full;
set company.usa;
output full;
run;
```

Atunci cand se folosesc doua nume de table in `data` si `output` are doar un tabel parametru, pasul `data` va crea cele doua tabele, dar va copia tabelul `usa` in `full` iar `empty` ramane gol.

Gasirea sfarsitului de tabel

Optiunea `end`

putem avea nevoie sa identificam sfarsitul de tabel (ultima obsevatie) din diferite motive: sa scriem in output doar o observatie care contine totaluri/sa mai efectuam niste operatii/etc.
Pentru a crea o variabila numerica a carei valoare este folosita pentru a identifica ultima observatie se poate folosi optiunea `end=` in instructiunea `set`. Forma generala este:

```
end=variable
```

unde `variable` va contine marker-ul de sfarsit de fisier. `variable` nu este adaugata in tabel, este initializata cu 0 si va avea valoarea 1 doar cand instructiunea `set` citeste ultima observatie din tabel.

Nu se foloseste `end=` impreuna cu `point=`: `point` citeste doar observatia indicata si nu conduce la sfarsitul tabelului

Exemplu:

Dorim sa insumam numarul de secunde inregistrate la mai multe teste pe banda de alergat. Urmatorul program calculeaza variabila `TotalTime` care cumuleaza valorile convertite in secunde:

```
data work.addtoend(drop=timemin timesec);
set clinic.stress2(keep=timemin timesec);
TotalMin+timemin;
TotalSec+timesec;
TotalTime=totalmin*60+timesec;
run;
proc print data=work.addtoend noobs;
run;
```

Daca insa vrem sa aflam direct doar valoarea totala atunci putem scrie in tabelul nou creat (`addtoend`) doar ultima observatie folosind `end=` impreuna cu un `if`:

```
 data work.addtoend(drop=timemin timesec);
set clinic.stress2(keep=timemin timesec)end=last;
TotalMin+timemin;
TotalSec+timesec;
TotalTime=totalmin*60+timesec;
if last;
run;
proc print data=work.addtoend noobs;
run;
```

Modul de citire a tabelelor SAS

Pasul `data` citeste si proceseaza tabelele SAS in mod similar cu fisierele text, cu diferenta ca in cazul tabelelor SAS retine valorile variabilelor de la o observatie la alta.