Paper 104-2010

Finding Your Subjects in a Haystack: The Many Ways of Implementing Table Look-Ups

Alla Sapozhnikova, Olga Galin
The George Washington University Biostatistics Center, Rockville, MD

ABSTRACT

This presentation concentrates on two SAS programming methods that enable the user to print various data files by ID when the list of IDs selected for reporting is stored as a separate data set. In order to avoid the use of multiple MERGE statements and WHERE clauses for each ID selection, we have developed a SAS macro that allows this task to be completed in a simple and elegant manner. Implementation of these applications made possible full automation of the process.

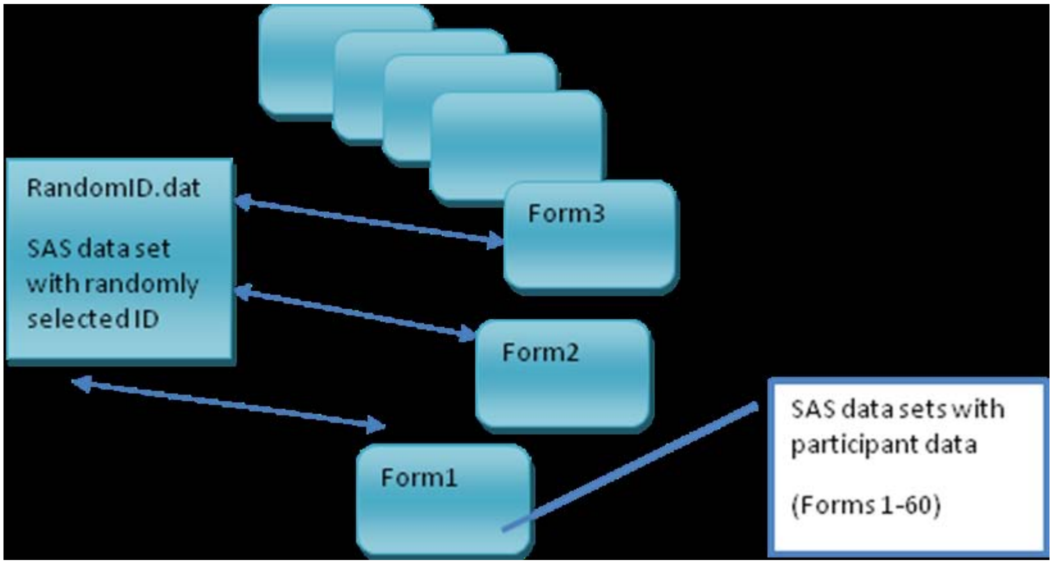
INTRODUCTION AND SUMMARY OF PROGRAMMING TASKS.

Ensuring data quality and integrity is key to the success of clinical trials. One of the methods of data QC for multi-center clinical trials is source document verification of recorded data during Clinical Center Site Visit. CoC representatives compare random samples of data from the central data base with source documents (participant's charts, paper questionnaires/forms, etc.).
The main goals of the data review:

- Data integrity assurance
- Form completion and data entry process assessment.

In order to perform this task, listings are created in SAS® that match the paper forms and are collated by subject and form. This presentation demonstrates several methods that enable the user to print various data files by specific ID order when the list of IDs selected for reporting is stored as a separate data set.

DATA STRUCTURE AND DESIRABLE REPORT STRUCTURE.


RandomID.dat
SAS data set with randomly selected ID
Form3
Form2
Form1
SAS data sets with participant data
(Forms 1-60)

Data for each form is stored as a separate SAS data set and contains data for all study participants; however only reports for those IDs that are located in RandomID.dat need to be printed.

Data audit reports also have to be printed in certain ID order. All reports are produced for the first participant ID, and so on in chronological order by ID number.

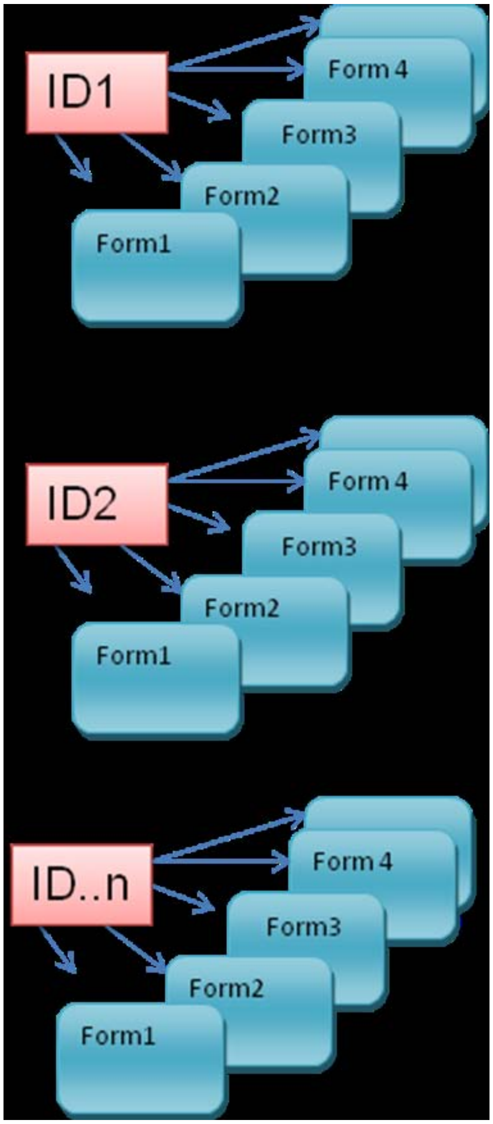


Figure 2.
Data Report structure

PROGRAMMING APPROACHES.

1. APPLICATION BASED ON CALL SYMPUT ROUTINE APPROACH.

- Create a lookup table as an array of macro variables to hold ID values and the last observation number.
- Create template sub-macros for printing/ reporting

- Call templates for each printing/reporting pattern in do loop from the first until the last observation of the IDs lookup table.

EXAMPLE.

Array of macro variables to hold ID values and the last observation number.

```
data _null_;
set metd_s end=last;
call symput ('patid'||trim(left(_n_)),ptid);
if last then call symput('nobs',_n_);
run;
```

reporting templates.

```
%macro print1 (id=);
proc print data = metd_s (where = (ptid=%str(%')&id%str(%'))) noobs label;
title 'Possible metabolic decompensation';
run;
%mend;
%macro print2 (id=);
proc print data = inslog (where =(ptid=%str(%')&id%str(%'))) noobs label ;
title 'Related Form1 records ';
run;
%mend;
%macro print3 (id=);
proc print data = decomp (where =(ptid=%str(%')&id%str(%'))) noobs label;
title 'Related records of confirmed metabolic decompensation';
run;
%mend;
```

Call reporting templates in a do loop first-last obs. of data set that contains ID values.

```
%macro block;
%do i = 1 %to &nobs;
%put &&patid&i;
%print1(id=&&patid&i);
%print2(id=&&patid&i);
%print3(id=&&patid&i);
%end;
%mend;
%block;
```

2. APPLICATION BASED ON USE OF PROC SQL APPROACH.

- Generate a string macro variable to hold * delimited list of ID selected for reporting.
- Scan macro variable to create an IDs lookup table.
- Process printing/reporting templates using the IDs lookup table.

EXAMPLE.

```
proc sql;
select ptid into :patids separated by '*' from metd_s;
quit;
%put &patids;



%Macro printit ;
%let i =1 ;
%do %until (%scan(&patids,&i,'*')= ) ;
%let id = %scan(&patids,&i,'*');
%put &id;

%print1(id=&id);
%print2(id=&id);
```

```
        %print3(id=&id);

        %let i = %eval(&i + 1);
        %end;
        %mend;
        options macrogen;
        %printit;
```

CONCLUSION.

This paper presented simple programming methods that can be used by inexperienced SAS programmers on different platforms. Implementation of these applications allows full automation of the data reporting process.

For more information, contact

Alla Sapozhnikova
The George Washington University Biostatistics Center, Rockville, MD
Rockville, MD 20852
(301)881-9260