A Simple Solution for Generating Two Sets of Similar Reports

Lion Li, Merck Serono, Beijing, China
Zheng Wang, Merck Serono, Beijing, China
Yanhong Li, Merck Serono, Beijing, China

ABSTRACT

In preparing regulatory submissions for multi-regional clinical trials, we are often required to generate two sets of reports for one trial with one set of report including the whole ITT population (i.e., subjects from all countries involved) and another partial ITT population (i.e., subjects in the country with regulatory authority). The two reports contain all contents in the same layout, and the only difference is the population selection, which should be specified in titles to differentiate one from the other. In this paper a simple solution for generating two sets of similar reports is presented with the aim of reducing workload and increasing efficiency.

INTRODUCTION

To create two sets of reports, generally speaking, programmers need to prepare two sets of programs in a habitual manner. Meanwhile doing reproduction of programming and making electronic comparison of outputs are two critical steps to insure the quality of the programming performance. That will be time consuming if we handle this task using traditional method, i.e., develop two sets of programs, especially for a task with more than 300 TLFs to be generated in a month. The process shown here has been demonstrated as an efficient way to produce double outputs in one program, with different population selection defined.

Flow chart in Figure 1 is the basic idea showing how we handle this process in a simple way.

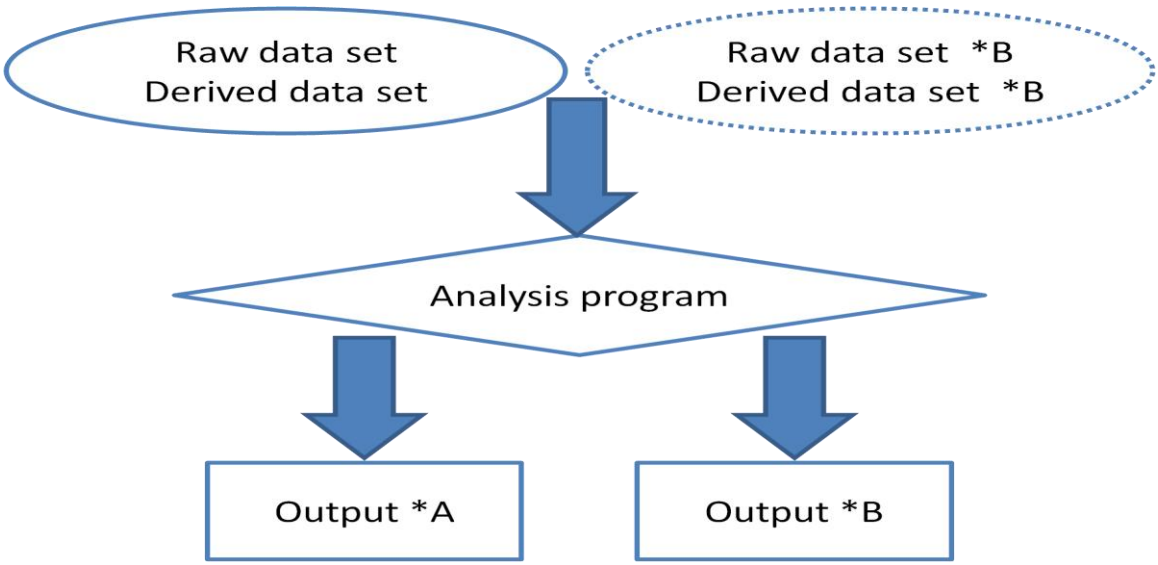

Figure 1. Basic idea

IMPLEMENTATION

First of all, we prepare all necessary datasets. Following within each analysis program, we perform additional data manipulating if needed and then follow with analysis procedure step by step in a general way. It sounds nothing

charming here until now. However, several macros have been developed and implemented to help us achieve our goal in an easy way.

1. %GETDDB:
   This macro reads raw data and/or derived data, changes the population set, and saves raw data *B or derived data *B in a temporary folder.
2. %RUNB:
   This macro is the critical part in the whole process. It creates a program *B from former developing program, and saves program *B provisionally. Then in batch mode, original program and program *B are submitted in succession. Therefore, there are two sets of outputs (tables, listings, or figures) will be generated successfully.
3. %TITLESB:
   This macro is mainly used for creating appropriate titles in regard to displaying appropriate population selection.

Let's start reading more details about these macros to see how they work to reach our target.

%GETDDB

Macro %getddb works for generating new dataset *B based on existing dataset. It is applied for both raw datasets and derived datasets. All dataset *Bs are saved impermanently after generation, are then called into analysis programs, and will not be moved into production eventually. Hence, it does not occupy extra space to save data resource redundantly.

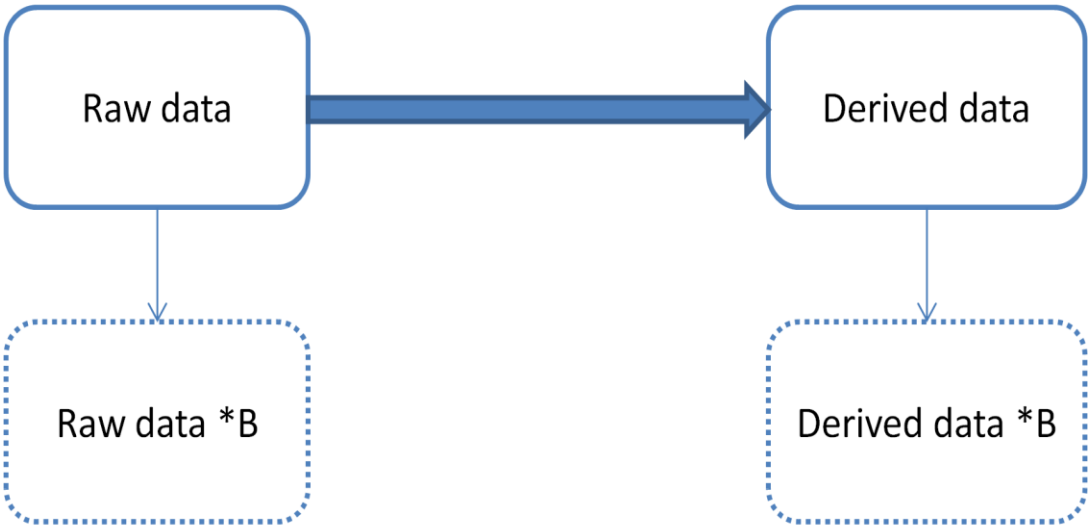

Figure 2. Get data

This macro starts with copying format catalog to applied datasets. To accomplish this, PROC CATALOG is utilized.

Secondly, every piece of information from the existing derived dataset is reproduced and will be copied into derived dataset *B with different population selection. Similarly, it creates raw datasets *B from original locked raw dataset (Figure 2).

At the end of this macro's execution we then have all required data source beforehand.

SAS codes below shows how this macro creates raw dataset *B and derived dataset *B.

```
proc sql noprint;
    create table detail as
    select distinct libname, memname
    from dictionary.columns
    where upcase(libname)='DD';
    select compress(put(count(*),best.)) into :cnt from detail;
    select compress(memname) into :name1-:name&cnt from detail;
quit;
%macro doit;
    %do i=1 %to &cnt;
        proc sql noprint;
            create table ddb.&&name&i as
            select * from dd.&&name&i
            where pt in (select distinct pt from dd.ad_subj where pop=2);
        quit;
    %end;
%mend;
```

%RUNB

Original idea on how to perform this task is to create a virtual program *B which will be deleted eventually.

Macro %runb is such a macro that helps us to accomplish the task for the entire process as described above. Points to be noted that macro %runb should be submitted in a batch mode to streamline the process and reduces the chance of errors.

Figure 3 below shows a flow chart that describes how macro %runb performs its operation.

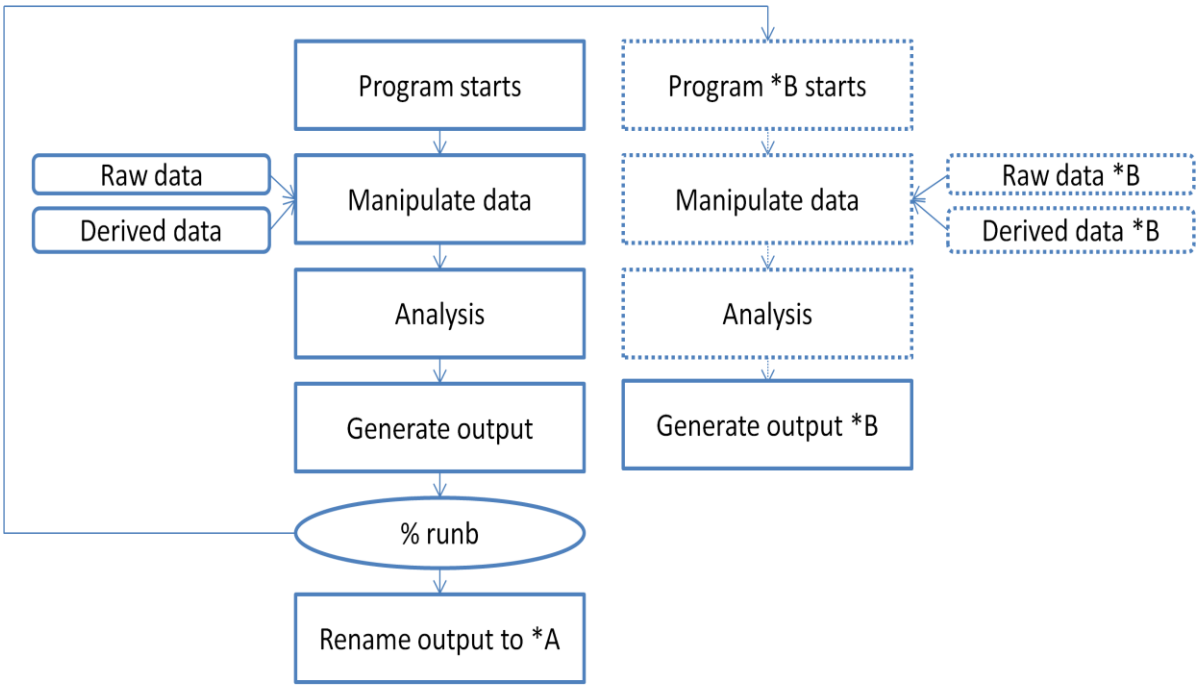

Figure 3. Flow Chart

1. Defines output set *A. To achieve this goal easily, it copies the output which has the exact same name as the original program's name (without A or B as a suffix), renames it as *A, and deletes the original one. It works for independent validation program in the same way.
2. Macro %runb then creates program *B in a temporary folder.
3. Macro %runb searches all library names, program names, and related information in *B program and annex them a suffix 'b' until finding the word '%runb ' exists.
4. Now we have built up program *B, brick by brick, and it is fully ready to run.
5. When the analysis program is submitted in batch mode, output *A will be created first. Next, program*B will be called and output*B will then be generated automatically.

The piece of SAS code below shows how %runb works with producing two sets of outputs.

```
     x "copy &tabdir.\&se_pgmname..%SYSFUNC(scan(&suf.,&i,' '))
&tabdir.\&se_pgmname.a.%SYSFUNC(scan(&suf.,&i,' '))";
     x "del &tabdir.\&se_pgmname..%SYSFUNC(scan(&suf.,&i,' '))";
data pgma;
   infile pgma;
   if index(pgma,'rd055.') then pgma=tranwrd(pgma,"rd055.","rd055b.");
   if index(lowcase(pgma),lowcase(%str('%runb;'))) then stop;
run;

%include pgmb;
```

After running program, all contents within solid lines in Figure 3 will be saved into production and submitted to regulatory. Other than that, things are saved in a temporary place and will be deleted at the end.

%TITLESB

This macro would be invoked within macros %printto_sb to call an Excel spread sheet which contains all titles appeared in TLFs. By running this macro, for each output *B, a 'b' will be concatenated to the TLF number, and a subtitle which specifies the population selection will be added as well. In other words, through calling macro %titlesb, we will have **Table 14.1-1a** appeared in output *A header part, and **Table 14.1-1b** appeared on the same location in output *B. Also, a subtitle like 'Including Korean' or 'Excluding Korean' is added to each TLF. Here below is the example table header of Table 14.1-1a and Table 14.1-1b.

Table 14.1-1a:

Table 14.1-1a: Subject Disposition
Including Korean

	All screened subjects[1]	ITT/Safety population[2]	PP population[3]

Table 14.1-1b:

Table 14.1-1b: Subject Disposition
Excluding Korean

	All screened subjects[1]	ITT/Safety population[2]	PP population[3]

CONCLUSION

The original plan is to timely and effectively deliver 'duplicated' outputs for a regulatory submission. The whole process described above is demonstrated to be a simple and efficient way to achieve the goal. This simple method only requires minimal numbers of program and reduces retention, and the efficiencies of programming work are scaled up very well.

REFERENCES

[1]Macro %printto_s and %titles from macro libraries in Merck Serono.

[2]Sanbonmatsu, Lisa (2000), "Batch Processing with SAS®: Beyond Running Programs Overnight", NESUG 2000 Conference Proceedings (Coders' Corner).

ACKNOWLEDGMENTS

I would like to thank Dr. Jie Chen for reviewing this paper and providing valuable comments.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Lion Li
Merck Serono (Beijing) Pharmaceutical R&D Co., Ltd
10F, Tower B, Gemdale Plaza
91 Jianguo Road, Chaoyang District
Beijing 100022 China
Work Phone: +86 10 5903 1499
Fax: +86 10 5907 2699
Email: lion.li@merckgroup.com

Zheng Wang
Merck Serono (Beijing) Pharmaceutical R&D Co., Ltd
10F, Tower B, Gemdale Plaza
91 Jianguo Road, Chaoyang District
Beijing 100022 China
Work Phone: +86 10 5903 1524
Fax: +86 10 5907 2699
Email: zheng.wang@merckgroup.com

Yanhong Li
Merck Serono (Beijing) Pharmaceutical R&D Co., Ltd

10F, Tower B, Gemdale Plaza
91 Jianguo Road, Chaoyang District
Beijing 100022 China
Work Phone: +86 10 5903 1494
Fax: +86 10 5907 2699
Email: yanhong.li@merckgroup.com