

Data Management and Analysis of Adverse Events Data using SAS ARRAY, IF-THEN, DO LOOP, and MACRO Statements

Amita Patil, Johns Hopkins University School of Nursing; Jason E. Farley, Johns Hopkins University School of Nursing; Kelly Lowensen, Johns Hopkins University School of Nursing; and Chakra Budhathoki, Johns Hopkins University School of Nursing

ABSTRACT

The adverse events (AE) data in healthcare are essential to understand the medication side-effects. Patients experience multiple adverse effects of drugs while on medication for various diseases like tuberculosis, HIV, cancer, or other chronic illnesses. Data management, organization, and analysis are needed to provide an understandable and meaningful output for clinicians and researchers to develop intervention plans. The REDCap data collection tool captured AE or symptoms, AE grades 1 to 5, start and end dates, and outcomes. Due to the longitudinal nature of the study, the data is unformatted using the SAS PROC step to reduce processing time. The total of each grade is calculated using SAS ARRAY, IF-THEN, and DO LOOP statements. The SAS MACRO is used to categorize the symptoms in their body systems. Topics included are 1) unformatting REDCap data using the PROC step to reduce processing time; 2) generating ARRAY variables; 3) creating IF-THEN statements within DO LOOP to get total grades; 4) running MACRO to categorize symptoms into body systems; 5) running frequency statistics; and 6) selecting the ODS PDF statement to output data.

INTRODUCTION

Adverse event (AE) data reports are essential for clinicians, investigators, and pharmaceutical companies to assess the drug's side effects and analyze the safety of the drug. The adverse events (AE) report creation is a time-intensive process. This paper describes codes for the total number of each AE, the sum of each grade, and the sum of the systemic involvement of AE. The dataset contains record ID, AE terms, grades from 1 to 5, outcome, all numeric variables, and start and end dates (see Table 1).

Variable Name (Raw Data)	Type	Description
Record ID	Numeric	ID of patient
Symp_complaint_1 through Symp_complaint_55	Numeric	Adverse event terms
Sym_grade_1 through Sym_grade_55	Numeric	Grade of symptoms 1 to 5
Start Date	Date	Start Date of Adverse Event
End Date	Date	End Date of Adverse Event
Outcome	Numeric	Outcome of Adverse Event

Table 1: REDCap Variables Description

UNFORMATTING DATA

This paper is based on REDCap AE data, and unformatting is done to reduce processing time. When you download your REDCap data, map the SAS codes from the directory to run your data and unformat the dataset using the codes below (Diseker, 2006).

```
proc datasets lib=work;
    modify redcap;
    format _all_;
    informat _All_;
quit;
run;
```

SAS ARRAYS, IF-THEN, AND DO LOOP FOR TOTAL GRADE 1 TO 5

The REDCap dataset contains AE grades reported in variable name symp_grade_1 through symp_grade_55. The ARRAY named sev is created for variables symp_grade_1 through symp_grade_55 to be referenced in the DO LOOP statement. Total_Grd_1 is a new variable created to calculate the running total of grade 1 AEs reported while the patient is on treatment. Total_Grd_1 variable is initialized by giving the value 0. The DO LOOP helps to select the grade from the variable, and the iterative process increases the value of j from 1 to 55 as we have 55 variables in ARRAY. The IF-THEN statement selects grade 1 from the variables in ARRAY referenced by sev (j). Total_Grd_1+1 statement picks grade 1 from the symptom variable and adds grade 1 every time the loop runs to give the sum of grade 1 symptoms. The END statement for each DO LOOP ends the iteration and runs the statement.

We would repeat the same code for calculating total grade 2 AE by renaming variable Total_Grd_1 to Total_Grd_2 and changing if sev (j) =2 so that it calculates the total grade 2 AEs for each patient and repeat the same process for getting the total of grades 3, 4, 5 respectively.

```
data All_AE1;
set REDCAP;
array sev{55} symp_grade_1-symp_grade_55;
Total_Grd_1=0;
do j=1 to 55;
    if sev(j)=1 then do;
        Total_Grd_1+1;
    end;
end;
run;

options nodate nonumber;
title "Total of Each Grade 1-5";

ods pdf file="Pathway/outputAE.pdf";
proc print data= All_AE1;
var record_id Total_Grd_1 Total_Grd_2 Total_Grd_3 Total_Grd_4 Total_Grd_5;
run;
ods pdf close;
```

Finally, the ODS PDF FILE statement outputs the desired variables listed in the VAR statement of the PROC PRINT step (see Figure 1).

Obs	record_id	Total_Grd_1	Total_Grd_2	Total_Grd_3	Total_Grd_4	Total_Grd_5
1	CBH002	5	4	1	0	0
2	CBH003	0	2	0	0	0
3	CBH007	3	0	1	0	0
4	CBH009	5	0	0	0	0
5	CBH012	2	2	3	0	0
6	CBH013	1	0	0	0	0
7	CBH015	7	9	0	0	0
8	CBH016	1	2	1	0	0
9	CBH017	2	2	0	0	0
10	CBH018	2	0	0	0	0
11	CBH019	6	2	0	0	0
12	CBH020	5	4	0	0	0
13	CBH021	2	3	0	0	0
14	CBH022	13	8	3	0	0

Figure 1: ODS Pdf Output of Total of Each Grade 1-5

MACRO FOR FREQUENCY OF AE

If you want to count the total of each AE terms each patient experienced while on treatment, the MACRO created below calculates the total AE terms; such data are helpful to know the most frequent AE of the drug. The MACRO named 'std' is created, with MACRO parameters var1 and term1 for referencing the variable and term. The ARRAY sym_nam has 55 variables, symp_complaint_1-symp_complaint_55. The count_&term1 is a new variable initialized to 0. The DO LOOP statement iterates the process 55 times to pick the AE term referenced in the IF-THEN statement using array sym_nam (x) and &var1 parameters to specify the variable AE term (SAS Institute Inc, 2001). As we have unformatted the SAS REDCap data, the AE terms are numeric, e.g., 1 for weight loss, 2 for edema, etc. Count_&term1 creates the running total of the AE term in the dataset, and the END statement ends with the DO LOOP.

A PROC FREQ step runs the frequency of the count_&term1 to give the total frequency of the specific AE term. PROC PRINT prints the variable specified in the var statements. The %MEND ends the MACRO 'std.' You can run frequencies of the desired AEs using the MACRO call with parameters variable and term. If your dataset has any AE which repeats over interval for a patient, it will be counted that many times. Thus, running a frequency of AE terms per ID will help determine whether repetition of AE terms exists in your dataset.

```

%macro std(var1, term1);
  data name;
  set redcap;
  array sym_nam{55} symp_complaint_1-symp_complaint_55;
  count_&term1=0;
  do x=1 to 55;
    if sym_nam(x)=&var1 then do;
      count_&term1=count_&term1+1;
    end;
  end;
end;
run;

```

```

proc freq data=stdname;
tables count_&term1;
run;
proc print data= stdname;
var record_id count_&term1;
run;
%mend std;
%std(1, Weight_loss);
%std(2, Edema);
%std(3, changes_vision);
%std(4, Hearing_Loss);

```

The PROC PRINT in the MACRO prints the desired variables listed in the MACRO parameter. For example, %std (1, weight loss) gives frequencies of the AE term weight loss in the dataset (see Figure 2).

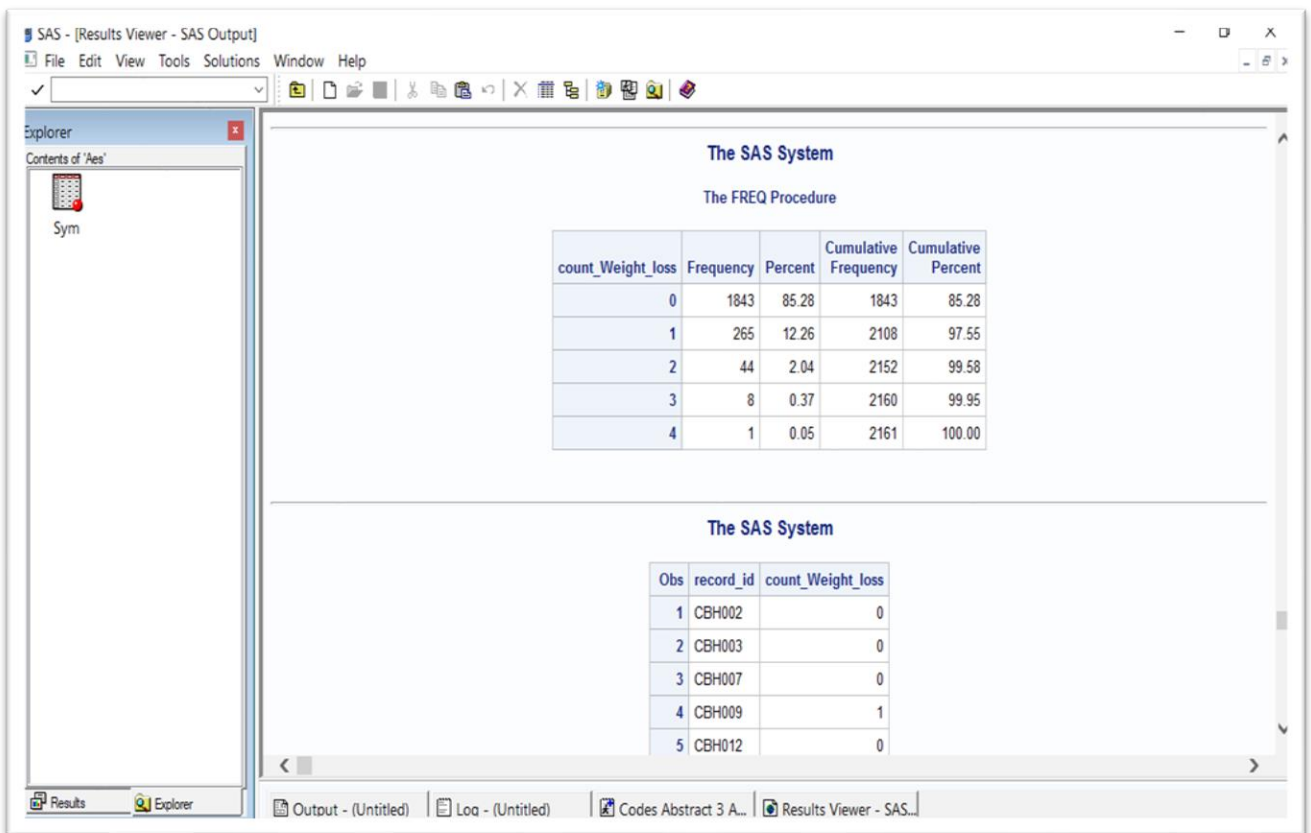


Figure 2: Output of AE term variable count_Weight_loss from the dataset

MACRO FOR CATEGORIZING AE TO BODY SYSTEMS

The MACRO `systemc` presented below categorizes the AE to the system it belongs to. The `systemc` MACRO has two parameters, `var3` and `var4`. ARRAY called `Aename` has 55 variables `symp_complaint_1-symp_complaint_55`, and the DO-END statement with IF_THEN clause creates a systemic variable `sysname_&var4` that categorizes the AE term to the body system. For example, a MACRO call `%systemc (1, General)` categorizes weight loss to the general system. PROC FREQ procedure gives the frequency of the specific term referenced by the MACRO parameter.

```

%macro systmc(var3, var4);
  data stdname_systemic;
  set REDCap;
  array AName{55} symp_complaint_1-symp_complaint_55;
  do y=1 to 55;
  if AName(y)=&var3 then sysname_&var4="&var4";
  end;
  run;

  proc freq data=stdname_systemic ;
  tables sysname_&var4/missing;
  run;

  proc print data=stdname_systemic;
  var record_id sysname_&var4;
  where sysname_&var4 ne "";
  run;
%mend systmc;
% systmc(1, General);
% systmc(2, General);
% systmc(3, ENT_HT);
% systmc(4, ENT_HT);

```

The PROC PRINT in the MACRO prints the output with the variables specified in the VAR statement by the parameter (see Figure 3).

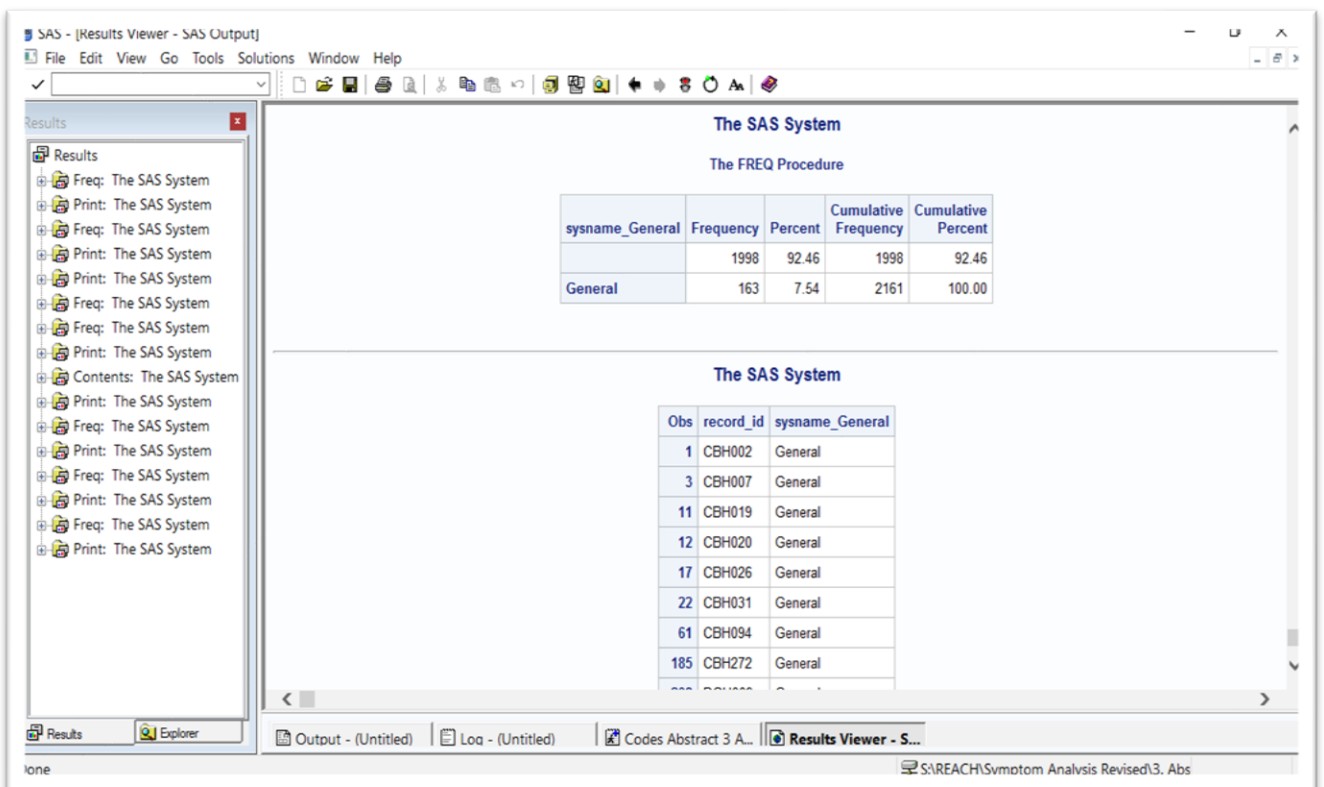


Figure 3: PROC FREQ Output of sysname_General variable from the dataset

CONCLUSION

SAS ARRAYS and MACRO programming are time-saving procedures that can be used in repetitive tasks such as processing adverse events or symptom-related clinical data, as shown in this paper.

REFERENCES

Diseker, R. (2006). *When a Shortcut is More Than a Quick Fix*. LexJansen. Retrieved September 20, 2023, from <https://www.lexjansen.com/pharmasug/2006/CodersCorner/CC08.pdf>

SAS Institute Inc. 2001. *Step-by-Step Programming with Base SAS ® Software*. Cary, NC: SAS Institute Inc.

RECOMMENDED READING

- *Base SAS® Procedures Guide*

CONTACT INFORMATION

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

Amita Patil
Johns Hopkins University School of Nursing
Amita.patil705@gmail.com