PharmaSUG2012 – Paper CC14

Quick Data Definitions Using SQL, REPORT and PRINT Procedures

Bradford J. Danner, PharmaNet/i3, Tennessee

ABSTRACT

Prior to undertaking analysis of clinical trial data, in addition to review of the Protocol, Case Report Forms, and Statistical Analysis Plan, a basic understanding of the SAS data available is crucial for statistical programmers and biostatisticians. Such an understanding can be accessed readily using the DATASETS and CONTENTS procedures. To centralize in one place and increase efficiency by which the information may be reviewed, a quick method using the SAS DICTIONARY library with the SQL, REPORT and PRINT procedures is presented. Through dynamic, data-driven, generation of macro control variables, a hyper linked data definitions document, in multiple formats, may be produced quickly, and with minimal user input.

INTRODUCTION AND BACKGROUND

Successful programming and analysis of a clinical trial requires a good understanding of both the protocol and statistical analysis plan, and most importantly, the data collected and available for the study. With regards to the SAS datasets, the size, number, and structure of datasets available are all crucial bits of information needed to get started with programming an analysis. Where information is coming from and resides is also important. Specifications can change within a study, or vary greatly between studies and/or therapeutic areas. Therefore, in the absence of already established documentation, a tool to gather all this information together into one or two files quickly at the initial stages of project work is very useful for a programmer or statistician to familiarize them with the SAS datasets.

Informally, the DATASETS and CONTENTS procedures in SAS are great for quickly presenting this information throughout the programming process. Alternatively, much of this information also resides in the DICTIONARY library, accessible using the SQL procedure. Furthermore, the flexibility that SQL provides for dynamically creating and storing macro variables offers a powerful method to access and present the information, with little or no user input. Thus, after a few interim steps, the REPORT procedure is used to present the information such that individual users can navigate as needed using hyperlinks. A subsequent optional feature uses the PRINT procedure to present a subset of the data itself.

METHODS AND RESULTS

The process by which the quick data definitions are built was intentionally designed as simply as possible, to maximize ease of use and minimize the time necessary to produce a reviewable document, and may be summarized as follows:

1. User provides location of SAS datasets needed for review.
2. SQL procedure to create temporary datasets from Tables and Columns for use by REPORT and PRINT procedures.
3. SQL procedure to dynamically generate macro variables.
4. Create data definitions within one hyperlinked PDF file.
5. Optionally create a hyperlinked spreadsheet with example data.

The code necessary to perform these 5 steps may be found in the Appendix. The next few paragraphs will go into further detail regarding specific aspects of Steps 2-5.

Once the location of relevant data is provided by the user, temporary datasets are created to contain the information presented in the data definitions. For this simplified example, two datasets are created from the Table and Column dictionaries:

```
create table TOC as select libname, memname,nobs,nvar,crdate from dictionary.tables
where upcase(libname) in ("&libname.") and  nobs gt 0 ;

create table ALLVARS as select memname,name, type, length, varnum, label
from  dictionary.columns where upcase(libname) in ("&libname.") and memname in
(select memname from dictionary.tables where upcase(libname) in ("&libname.") and
nobs gt 0);
```

The temporary dataset TOC created from the Table dictionary will be used later in the process as a table of contents for the data definitions. Given its purpose, the dataset is limited to only a subset of the variables included in the Table dictionary. Another temporary dataset ALLVARS is created as a subset of the Columns dictionary, which will act as a repository for dataset level information of interest in the data definitions. Again, only a subset of the variables available is kept for use in the document.

For the next step in the process, still within the SQL procedure, several macro variables are dynamically created to iteratively present information within the definitions. Then main information needed is the number of datasets present in the library location, and the names of those datasets. Both types of information may be quickly found and stored as macro variables using the SQL procedure and the temporary datasets already created, as follows:

```
select n(memname) into :nset from work.TOC;

select memname into :domain1-:domain%left(&nset) from work.TOC;
```

In this example, only the datasets with more than zero observations are considered, by adding the appropriate qualifier to the WHERE clause within the SELECT statement. Additional qualifiers to filter, or narrow, the scope of data considered may be quickly added as needed.

The next step of the process utilizes the macro variables and temporary datasets of TOC and ALLVARS, all created in one SQL procedure, in combination with the REPORT procedure, to quickly and efficiently produce a one-stop document of data definitions. In this example, a PDF format is chosen for illustration, but other formats could be appropriate. The first portion of the document is a table of contents produced from the temporary dataset TOC. Since the MEMNAME variable, or dataset name, is common to both the TOC and ALLVARS datasets, it will be used to automatically provide navigable linkage to subsequent dataset level portions of the document, and to subsequently provide linkage to return to the table of contents. This is accomplished using the ANCHOR option in the ODS PDS statement and the COMPUTE and CALL DEFINE statements with the REPORT procedure as follows:

```
ods pdf anchor="page1";

compute memname;
    rtag="#"||strip(lowcase(memname));
    call define(_col_,'url',rtag);
endcomp;
```

Following the first REPORT procedure, the macro variables created previously in SQL are used to individually present the datasets within the library, with a target link from the table of contents, and a link to return to the table of contents at the bottom of each page. Again, this is accomplished using the ANCHOR option in the ODS PDF statement, coupled with a URL tag in the FOOTNOTE statement of the REPORT procedure, as such:

```
%do i=1 %to &nset.;
  ods pdf anchor="%lowcase(%cmpres(&&domain&i))" startpage=now;
  ods proclabel="&&domain&i";
  title1 "contents of &&domain&i";
  proc report data=work.ALLVARS (where=(memname = "&&domain&i" ))
                                contents="&&domain&i" nowindows;
    column varnum name label type length;

    define <...>

  footnote1 "^S={URL='#page1'}Table of Contents";
  run;
%end;
```

Therefore, with one SQL procedure to create macro variables and a small number of temporary datasets, one REPORT procedure to create a broad table of contents for a location of interest, and one REPORT procedure within an iterative loop dependant upon the number of datasets of interest, a quick overview of the dataset structure is obtained, with navigability embedded automatically. The collection of procedures runs easily with very little input, specifically location of data and location desired for the document, necessary from the user becoming familiar with project data.

In addition to quickly being able to see the datasets and variables available for a given project, a closer look at the contents may also be necessary. Therefore, an option to present the data itself was also included in the data definitions macro, which allows the user to create a linked document, in this example a spreadsheet with generic

auto-filters in place, where each worksheet within the file represents either the table of contents consistent to the definitions document, all the variables and associated dataset in which they are located, or the dataset itself with a subset of the data presented to limit the size of the file. Using the same macro variables and temporary datasets created before, a spreadsheet file is created using the ODS EXCELXP tagset, linkage between the table of contents and individual worksheets facilitated through COMPUTE and CALL DEFINE in the REPORT procedure, and the SHEET_NAME option of the tagset associated with each PRINT procedure, illustrated briefly here:

```
%do i=1 %to &nset.;
  ods tagsets.excelxp options(sheet_name="&&domain&i");
  proc print noobs data=&libname..&&domain&i (obs=10);
  run;
%end;
```

Using the same general process utilized for creating the data definitions in PDF format the user now has a navigable spreadsheet with a subset of the data itself. When used in conjunction with a study protocol and analysis plan, the statistician or programmer will quickly be able to familiarize themselves with the data available.

CONCLUSIONS

The technique presented here offers statisticians and programmers a quick and easy to navigate tool for familiarizing oneself with data available when beginning a project. Using the technique or macro presented in the Appendix requires very little user input and can be rapidly deployed. An adjustment to the code for selection of information from the Table and Column dictionaries requires only minor adjustments to the SQL selection statements. Adding navigability through embedded hyperlinks allows reviewers to get a quick overview of the data available with which to work with.

REFERENCES AND RECOMMENDED READING

- Carpenter, Arthur L, 2007, *Carpenter's Complete Guide to the SAS REPORT Procedure*, SAS Institute Inc., Cary NC.
- SAS 9.2 Online Documentation

ACKNOWLEDGMENTS

I would like to thank my colleagues in the Biostatistics and Statistical Programming groups at PharmaNet/i3 who provided comments while drafting the process, especially Fernando Enriquez and Karl Miller. Their insights and guidance are appreciated.

CONTACT INFORMATION

Comments and questions are valued and encouraged. Please contact author at:

Name: Bradford J. Danner
Enterprise: PharmaNet/i3
Address: 1787 Sentry Parkway West, Suite 300, Building 16
City, State ZIP: Blue Bell, PA 19422 USA
Work Phone: (615) 302-3608
E-mail: BDanner@pharmanet-i3.com

APPENDIX

```
/*  user needs to assign a path to SAS datasets */
libname <user supplied> '<user supplied>';

%macro definitions(libname=,destination=,example=);

proc sql noprint;
  create table TOC as select libname, memname,nobs,nvar,crdate
  from dictionary.tables
  where upcase(libname) in ("&libname.") and  nobs gt 0 ;

  create table ALLVARS as select memname,name, type, length, varnum, label
  from  dictionary.columns where upcase(libname) in ("&libname.")
  and memname in (select memname from dictionary.tables where upcase(libname) in
  ("&libname.") and nobs gt 0);

  select n(memname) into :nset from work.TOC;

  select memname into :domain1-:domain%left(&nset) from work.TOC;
quit;

ods listing close;
ods escapechar="^";
ods pdf anchor="page1";
ods pdf style=printer file="&destination.quick_dde.pdf" pdftoc=1;
ods proclabel='list of datasets';
title1 "list of datasets";
proc report data=TOC nowindows contents="dataset definitions";
  column memname nobs nvar  crdate;

  define memname/display      style={cellwidth=3 cm just=l};
  define nobs/display style={cellwidth=3 cm just=c};
  define nvar/display style={cellwidth=3 cm just=c};
  define crdate/display style={cellwidth=4 cm just=c};

  compute memname;
    rtag="#"||strip(lowcase(memname));
    call define(_col_,'url',rtag);
  endcomp;
run;

%do i=1 %to &nset.;
ods pdf anchor="%lowcase(%cmpres(&&domain&i))" startpage=now;
ods proclabel="&&domain&i";
title1 "contents of &&domain&i";
proc report data=work.ALLVARS (where=(memname = "&&domain&i" ))
                              contents="&&domain&i" nowindows;
  column varnum name label type length;

  define varnum/order noprint order=internal;
  define name/display   style={cellwidth=4 cm just=l};
  define label/display  style={cellwidth=8 cm just=l};
  define type/display   style={cellwidth=4 cm just=c};
  define length/display style={cellwidth=4 cm just=c};
footnote1 "^S={URL='#page1'}Table of Contents";
run;
%end;

ods pdf close; ods listing;

%if &example=YES %then %do;
ods listing close;
```

```
ods tagsets.excelxp
file="&destination.quick_data_view.xls" style=normal
options(zoom='75' frozen_headers='yes' autofilter="all" sheet_interval='proc'
absolute_column_width="10" sheet_name="datasets");

proc report data=TOC nowindows contents="dataset definitions";
  column memname   nobs nvar  crdate;

  define memname/display      style={color=blue};
  define nobs/display   ;
  define nvar/display;
  define crdate/display ;

  compute memname;
    rtag="#"||strip(upcase(memname))||"!a1";
    call define(_col_,'url',rtag);
  endcomp;
run;

ods tagsets.excelxp style=normal options(sheet_name="variables");
proc print label data=ALLVARS noobs;
  var  memname varnum name type length label;
run;

%do i=1 %to &nset.;
  ods tagsets.excelxp style=normal options(sheet_name="&&domain&i");
  proc print noobs data=&libname..&&domain&i (obs=10);
  run;
%end;

ods tagsets.excelxp close;
ods listing;
%end;
%mend definitions;
```