MWSUG 2016 - Paper BB11

Read html files and create standard tables for BISR at KUMC

Chuanwu Zhang, Department of Biostatistics, University of Kansas Medical Center, Kansas City, KS
John Keighley, Department of Biostatistics, University of Kansas Medical Center, Kansas City, KS
Byron J. Gajewski, Department of Biostatistics, University of Kansas Medical Center, Kansas City, KS

ABSTRACT

The Biostatistics and Informatics Shared Resource (BISR), an operation of University of Kansas Medical Center (KUMC) Biostatistics Department, collaborates with University of Kansas Cancer Center (KUCC) for new and revised grant applications, providing KUCC service in terms of study design, electronic Clinical Report Form (eCRF) creation, data management, statistical oversight, and analysis. Clinical trial data is entered and stored in the Velos system using eCRFs for data entry. Exporting data from tables in Velos and importing into SAS individually is time-consuming. The data entered into the eCRFs are exported into data files with an .xls file extension but they are actually html files.

A macro was developed to automatically read the data from all of the html files in a folder into SAS datasets and csv files simultaneously. This macro generates a summary report for each export/import cycle with built in error checks to ensure data quality of the conversion. Two additional macros were developed to increase efficiency. Macro 2 is used to summarize the demographics and adverse events forms. While Macro 3 creates SAS formats in a SAS catalog from the data dictionary files that are associated with the eCRFs and creates an RTF file for research team that can easily be used to review variable values and associated label. The macros increase productively by reducing the time required to move the data from storage to analysis and also gives basic reports needed for any trial automatically.

KEYWORD

KUCC, BISR, html file, data conversion, standard table, SAS formats, macros

INTRODUCTION

The Biostatistics and Informatics Shared Resource (BISR) is an operation of University of Kansas Medical Center (KUMC) Biostatistics Department, and it offers mission critical support to University of Kansas Cancer Center (KUCC) for new and revised clinical trial grant applications in terms of study design, eCRF design, data collection and management, statistical analysis, and so on.

Velos is the primary platform utilized to enter, manage and store the clinical trial data collected in KUCC clinical trials. In general, each clinical data domain, such as demographics, drug accountability, adverse event (AE), vital sign, concomitant mediation, and so on, is stored in Velos using an eCRF. A critical step for data analysis or monitoring is to export the data from Velos into file formats that can be used by statisticians within the Biostatistics Department. Among the different data file types used within the Department, extensions with .sas7dat and .csv are the two most popular since SAS and R are the primary applications used for data analysis in the Department.

Velos has the capability to export the data in a sas program with the data imbedded in the SAS program which uses the instream method to read the data into a SAS dataset. The data from each eCRF form could be exported into a separate sas program so a study with 40 forms would result in 40 separate SAS programs. However, obtaining a SAS data file via Velos default manner results in a very large file due to the length of the character variables, which are given a length of 4000 characters by default. Moreover, for the convenience of the analysis and operation, you need to repeatedly shorten the variable length. Additionally, although you can also import data with the extension of .xls from Velos, the data are actually html files so the libname statement and proc import doesn't work with these files. Finally, for those statisticians who prefer to use R to analyze data, Velos at KUMC can't export the forms in .csv file format. Facing those issues and based on the clinical research team desire to increase efficiency, the need for

Macro1 a macro which is used to batch read data files from html files to SAS files and CSV files was realized.

Macro 2 was developed to generate standard demographics and safety tables from the adverse events file for the PI and statistician request of periodical study safety monitoring. BISR has gone through the process of building a protocol toolkit which standardized many of the variables that are required for phase I trials. This allows Macro 2 to expect the presence of some variables such as gender, race, etc. Macro 3 was created to create formats to be used in the creation of tables for the clinical research team.

MACRO 1: BATCH CREATE SAS AND CSV DATA FROM VELOS HTML FILE

A general overview of macro 1 is as follows. The macro uses a data step to read all of the filenames in a folder that only contains the data files. These files have an .xls extension but they are really in html format. An example of the beginning of a file is given in figure 1.

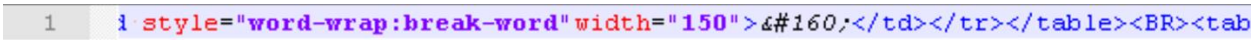

Figure 1 .xls file exists as html file

A count of the filenames are used to build a loop that reads every file. The files are read using '><' as a delimiter into a single variable. The .html file contains many formatting marks that need to be deleted this is an example '*table border="1" width="4380" cellspacing="0" cellpadding="0"*'. After removing observations that only contain the formatting marks the file will contain a variable with variable name and data. After determining the number of variables in the form the file can be transposed resulting in a file that has a typical data file layout of rows are observations and columns are variables. This file will have the variable names from Velos as data values in the first row. We use this row and PROC SQL to rename the default variable names from transpose to the names from Velos. At this time was also record values to check for problems with the PROC TRANSPOSE. We take the study number for the last observation and compare it to the study number that we hand enter. We keep the total number of observations in the file and compare to the number that Velos has recorded. We also keep a count of the number of unique study ids.

STORE THE HTML FILE NAME AND PATH INTO SAS DATABASE

We use a filename statement with a pipe to read the file names in the folder of interest as a first step. Moreover, the OPTIONS END = of SET statement combined with CALL SYMPUT statement can general a macro variable storing the total number of eCRF files.

Key Sample code:

```
Filename pipedata pipe "dir &htm_ph.  /s /b" lrecl=5000;
data _null_;
    set DataFiles end=eof;
    if eof then call symput('lastnum',_n_);
run;
```

In the following part, we will generally specify the macro development and details that plays an important role in the macro work process.

ROUGHLY BATCH READ HTML FILE TO SAS DATA BASE AND CSV FILE

This macro was not built to read every html file as the html format allows users to be very lax in their interpretation of proper html file format. When the project was started a web search was done looking for guidance on using SAS to read html files. This search wasn't encouraging to the authors with multiple blogs indicating that programmers shouldn't use SAS due to the lack of consistency in html file formats and it would be more efficient to use an existing html parser and pass those results to SAS. But, we didn't want to build a macro that would read any html file format, we only need to be able to read the files generated by Velos. Upon examination of the files we realized that we could read the file as a delimited file. We decided to use '><' as the delimiter in reading the files. The delimiter of html file is '><', but the single '>', '<' are also very commonly in the file. They are used in data values for example if a lab test is done and the result is '<6', then '<6' could be in the file as a data value. They are also used as delimiters

by the html so the macro has to check for these issues. The main functions applied here includes concrete INDEX, LENGTH, STRIP, CATS. Additionally, Velos uses some strings to represent missing data values. Html uses the strings   and   to insert white space in a document. Both of these strings contain macro triggers and need to be changed to missing values in SAS. You need to change the missing value format from "%NRSTR ( )" and "%NRSTR ( )" in html to “.”and “ ” as that in SAS data set by function of TRANWRD.

Key Sample code:

```
INFILE SOURCE  DLMSTR="><" recfm=n;

if index(web,"/td")>0 and index(web,'>')<1 and (length(strip(web)) ne 3)
  then web_p=cats('td style="wp:break-word"width="150"><',strip(web));

TEXT=TRANWRD(TEXT,"%NRSTR( )",".");
TEXT=TRANWRD(TEXT,"%NRSTR( )"," ");
```

UPDATE THE VARIABLE NAME AS STANDARD ONES FROM VELOS

By default, the file read into SAS is with the variable name as ‘Var_1’, ‘Var_2’, ‘Var_3’, and so forth after the utilization of PROC TRANSPOSE procedure. The meaningful variable name is stored as the variable value in the first row. Facing this issue, we utilize PROC SQL procedure to create n macro variables to store the meaning variable name and employ the macro %DO loop to rename the variables. Finally, delete the first row which stores the meaning variable name.

Key Sample code:

```
data _null_;
    set DataFiles end=eof;
    if eof then call symput('lastnum',_n_);
run;

proc sql noprint;
    select Distinct text_1 into: a1-:a%trim(&Vars.)
          from source4_n where group=1 order by var;
quit;

data source4t;
    set _source4t;
          %do k = 1 %to &Vars.;
          rename var_&k.= &&a&k.;
          %end;
    if _n_ = 1 then delete;
run;
```

WRITE THE SAS DATASET AND CVS FILE TO A PERMANENT LIBRARY

We then assign a permanent SAS library to store the transferred SAS dataset by DATA step and CSV file by PROC EXPORT procedure, which is quite simple straightforward.

Key Sample code:

```
libname z_test "&sas_lib.";
data z_test.&dmcmp_s.;
    set source4t;
run;

PROC EXPORT DATA= z_test.&dmcmp_s.
    OUTFILE= "&sas_lib.\%sysfunc(compress(&dmcmp_s.)).csv"
    DBMS=CSV REPLACE;
    PUTNAMES=YES;
```

```
RUN;
```

CROSS-CHECK THE STUDY ID AND OBSERVATION NUMBER FOR EACH DOMAIN

In the final step of Macro 1, we need to cross-check that the transferred data is identical as the original html file data or not. To achieve this, we make full use of the last observation from transferred data. One check condition is the number of the observation which you can get from the last row; the other condition from last observation is the study ID. Once both conditions satisfy, you can safety conclude that the transferring data is successfully done. The involved statement options and procedures includes SET NOBS= POINT =, MERGR, etc. For each cross-check of each domain, you will get one observation. Then, we apply macro %DO loop to set all those observations from all the domains together.

Key Sample code:

```
data num_sas;
  set source4t nobs=numobs point=numobs;
  nbs_sas=numobs;
  formname="%sysfunc(compbl(%sysfunc(strip(&shortfilename.))))";
  formnamelong=compbl(strip(form));
  keep nbs_sas formname;
  output;
  stop;
run;

data num_com_&j.;
  merge num_hl num_sas;
  by formname;
  length conlus  $40;
  if      nbs_hm =  nbs_sas then conlus= "observation match";
  else if nbs_hm ne nbs_sas then conlus="Warning!!! obs # does NOT match";
run;

%if &j. = 1 %then %do;
  data num_com_a;
    set num_com_&j.;
  run;
%end;

%else %if &j. ne 1 %then %do;
  data num_com_a;
    set num_com_a num_com_&j.;
  run;
%end;
```

HIGHLIGHT THE OUPUT COMPARISON RESULT

To facilitate the clinical team to review the comparison result, we also apply the proc format to highlight the background [1]. For those domain without observation, the yellow background is used. If the observation and/or study ID from last observation does not match, a warning message with red background will appear.

Key Sample code:

```
Proc format;
  value $bkgd
     "observation match" = "white"
     "study number match" = "white"
     "----" = "Yellow"
     other = "Red";
  value obscom
```

```
    . = "Red"
    0 = "Yellow";
run;

ods pdf file = "&fif_pn.";
title "observations & study number comparison";
 proc print data = final label noobs split='/';
    var com_rst uni_pid;
    var com_rst / style={background=$bkgd.};
    var uni_pid / style={background=obscom.};
 run;
ods pdf close;
```

Finally, you can be available a summary file as the figure 2 (summary screenshot) below. From the summary, you can clearly see the cross-check results of study number and observation. Moreover, we also add a column of 'Length of Longest Var Name' to avoid variable name truncation.

observations & study number comparison

form name	Number of Obs from Velos	Number of Obs from SAS	Equal # of Obs	Study # In SAS Dataset	Study # Entered manually	Study #'s Are Equal?	Length of Longest Var Name	# of unique patient study ID
ADVERSEEVENTS	20	20	observation match	ABC12	ABC12	study number match	24	7
BASELINE_SYMPTOMS	8	8	observation match	ABC12	ABC12	study number match	19	8
COMMENTS	11	11	observation match	ABC12	ABC12	study number match	19	6
COMMENTS_SHARED	0	0	observation match	No value due to NO obs	ABC12	----	.	0
CONCOMITANT_MEDICATIONS	10	10	observation match	ABC12	ABC12	study number match	22	6
DEMOGRAPHICS	19	19	observation match		ABC12	Warning!!! study # does NOT match	22	6
ECG_FORM_BASELINE	9	9	observation match	ABC12	ABC12	study number match	19	9

Figure 2 Observation & study number comparison summary

MACRO 2: GENERATE THE STANDARD DEMOGRAHICS AND ADVERSE EVENT OUPUT

The frequencies and percentage of demographics in terms of gender, ethnicity, race, education, country, etc. will be displayed in the standard output. The frequencies and percentage of AE terms, severe level as well as the AE duration days will be also summarized in the output. The key procedures involved includes PROC MEANS, PROC FREQ, PROC TRANSPOSE, PROC REPORT, and key functions involved includes PUT() and INPUT(). The final output is also partial screenshotted in Figure 2.

Key Sample code:

```
ods pdf file = "&dmg_smf." STYLE = FancyPrinter;
title "Demography Summary: Overall";
proc report data = pre_final headline headskip;
  column cate_n cate col_2 count;
    define cate_n/group noprint order = internal;
    define cate/group width = 40 "Variable";
    define col_2/display width = 40 "Category" style={background=$bkgd2_.};
    define count/display width = 6 "Frequency" center;
     break after cate_n / skip;
run;
```

Demography Summary: Overall

Variable	Category	Frequency
Gender	Female	5
	Male	5
Primary Ethnicity	Hispanic	2
	MSG	3
	NA	2
	Non-Hispanic	3

Figure 1 screenshot of sample from standard demographics output

MACRO 3: WRITE FORMAT FILE AS EXCEL TO SAS CATELOG AND RTF FILE

From statistician perspective, it is highly important to obtain the category variable value and corresponding label. Macro 3 imports the label information from the excel file into a SAS dataset, and then write this temporary SAS dataset into permanent SAS catalog and RFT file. The essential procedure involved includes PROC IMPORT, PROC FROMAT, and statements includes OPTIONS FMTSEARCH = and ODS RFT = .

Key Sample code:

```
PROC IMPORT OUT= WORK.form
  DATAFILE= "S:\Biostats\BIO-STAT\BISR\IITs\Day2_Survey.xlsx"
  DBMS=EXCELCS REPLACE;
  RANGE="Day2Survey$";
  SCANTEXT=YES;
  USEDATE=YES;
  SCANTIME=YES;
RUN;

options fmtsearch = (fmt);
ods rtf file = "C:\Users\chwzhang2015\Desktop\format_testing.rtf";
  proc format library=fmt fmtlib;
  run;
ods rtf close;
```

CONCLUSION

The elements of these 3 macro are generally not difficult to grasp. However, they are grouped smoothly to highly facilitate the routine work process of BISR KUMC, which can really be referred by other academia in case that they encounter the similar issues.

REFERENCES

1.Louise Hadden. 2006. “STOP! WAIT! GO!: See What Traffic-Lighting Can Do For You! ” *Poster of the SAS Users Group International*. San Francisco, CA. SAS Institute Inc. Available at: http://www2.sas.com/proceedings/sugi31/142-31.pdf

CONTACT INFORMATION

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

Chuanwu Zhang
Department of Biostatistics, University of Kansas Medical Center
czhang4@kumc.edu