SAS dataset utility with excel output

Rob Horton

Quanticate, Canterbury, UK

There may be a requirement to check that dataset creation dates are later than a certain point in time. A quick utility that allows for checking of file creation dates or to check that previously empty datasets are now populated may be of use in many situations. One such utility can be run by the programmer: an XML file is generated and a non-programmer can then check output without having to use SAS®. There are options to run over multiple study areas, to exclude a particular list of files, and to change cell background based on size or date. This presentation will go through the set-up of the macro to produce output files which can be tailored to study specifications.

Keywords: SAS, Excel, ODS, PROC report

Introduction

This paper will cover the utility code to produce and output an Excel spread sheet containing a list of dataset-associated creation dates, number of observations, and variables within.

The utility has been used across multiple protocols to allow a non-programmer to quickly check that all data were extracted post database release.

Section I — setting up macro variables

The macro requires the setting of parameters to state what is to be included. This includes specifying: a project area listing all protocols to be included or excluded; datasets to be excluded, for example, we may want to exclude metadata that will not change between different periodic runs of tables; data path, whether we are looking at raw or derived datasets; cut-off date of the reporting event this will be used to colour code any files older than the date time; and finally an output filename and destination (Fig. 1).

Section II — list files

The macro builds a list of protocols to be checked based on the macro variable settings, and then using SAS Help files, reads in the dataset details for those specified files, excluding the 'ignored' datasets. The datasets containing this metadata for each file are then appended into a single dataset named 'Matrix' (Fig. 2).

Section III — output to Excel

The final section of the code uses ODS to output two sheets in Microsoft Excel format:

- sheet 1: a 'matrix' or horizontal list of dataset names with modification date number of records and number of variables as one row per study;
- sheet 2: a vertical list of all datasets listed by study and dataset name.

ODS options are used to define column widths, add auto filter and name the sheet and output the file to the predefined destination (Fig. 3).

In this section we create formats and use hex colour codes that are applied to cells based on the define statements or compute block for 'RND_ERR' below (Fig. 4).

Hex code	Colour
#FFCC00	Yellow
#FF6600	Orange
#FFFFFF	White
#FF0000	Red
#FF00FF	Pink
#FFFF99	Pale yellow
#CCFFFF	Pale blue

PROC report is used to format the data going into the ODS output. There are two PROC reports firstly to create the horizontal list of files (Fig. 5).

This creates a list as shown in Figure 6.

In the second output, which is used to create a vertical list of files, we have a new ODS section, a second PROC report we also use a compute block to apply a background colour to a cell if the dataset 'RND_ERR' has any observations. There were some issues with the same randomization number being given to more than one subject; if this occurs, then RND_ERR is populated and therefore, we wanted to highlight where that dataset is populated (Fig. 7).

This creates a vertical list (Fig. 8).

Conclusion

The macro has been used to summarize all datasets in a piece of work involving multiple studies. It allows a

Correspondence to: R Horton, Floor 2, Exchange House, Lakesview Business Park, Canterbury, Kent CT3 4NH, UK. Email: Rob.Horton@Quanticate.com

Published by Maney
DOI 10.1179/175709311X13166801334154

```
*** Set up path for project, protocol and data sets to use ***;
%let prot_path = /app/bin/project/;  ** Unix path to the project **;
%let protocols = A1201;              ** Protocols to check, if blank all in directory **;
%let ignore_protocols = ;            ** Protocols to skip **;
%let ignore_datasets = MT1 MT2;      ** Datasets to skip **;
%let data_path = data;               ** Dataset library to use: data or derived_data **;
%let cutdt = "06MAY2011:07:06:00"dt; ** Flag modification dates older than **;
%let outxml = ~/matrix_&prot_path._&data_path..xml;   ** Output file name and path **;
```

Figure 1 Define macro variables

user to quickly view multiple project areas without having to navigate different study directories and uses colour coding to allow for fast visualization of errors.

Similar tools could be developed to create a suite of visual QC aides which enable non-technical staff to assist with basic but time-consuming QC tasks such as null-table checks or checking for differences between current and previous runs.

Acknowledgements

The author is grateful to Tomas Demcenko for the development of code to loose specifications.

```
   %if &protocols eq %str() %then %do;
       ** Get all protocols **;
       filename dir pipe "ls -f &prot_path/protocols |tail +3";

       data avail_prots;
          length prot $8192;
          length filename $200;
          infile dir lrecl=200 end=eof;
          input  filename $;
          prot = strip(filename);
          if indexw("&ignore_protocols", prot) > 0 then delete;
       run;
       filename dir;
   %end;
   %else %do;
       data avail_prots;
          length prot $8192;
          i = 1;
          drop i;
          do while(scan("&protocols", i) ne "");
            prot = scan("&protocols", i);
            output;
            i = i + 1;
          end;
       run;
   %end;

   ** Work on each protocol found not in exclusion list**;
   %local i j dsid2 dsid rc nrows nobs lstmoddt owner prot;
   %let dsid2 = %sysfunc(open(avail_prots));

   %do %while(%sysfunc(fetch(&dsid2)) eq 0);
       %let prot = %sysfunc(getvarc(&dsid2, %sysfunc(varnum(&dsid2, PROT))));
       libname dsmat "&prot_path/&data_path" access=readonly;

       proc sql noprint;
         create table ds_info as
           select *,
                  "&prot" as protocol length=128 label = "Protocol"
           from sashelp.vtable where libname eq 'DSMAT' and indexw("&ignore_datasets",
strip(memname)) eq 0
         ;
       quit;

       proc datasets nolist library=work;
          append
              base = matrix
              data = ds_info
              force;
         run;
       quit;


   %end;
```

Figure 2 Compile a list of datasets

```
ods tagsets.excelxp options(
    absolute_column_width="15, %sysfunc(repeat(%str(16, 10, 10,), 50)) 10"
    embedded_titles='yes'
    print_header="&sysdate9.: dataset (/&data_path) matrix"
    autofilter='yes'
    frozen_headers='yes'
    frozen_rowheaders='1'
    sheet_label="Matrix"
    sheet_name="Matrix"
)
file="&outxml." style=statistical

;
```

Figure 3 ODS set-up

```
proc format;
     value clrs  0 = '#FFCC00'
                 . = '#FF6600'
                0< - high = '#FFFFFF'
               ;
     value rnderr 0 = '#FFFFFF'
                  . = '#FF6600'
                 0< - high = '#FF00FF'
     ;

     value cutdt low - <&cutdt = '#FF0000';

run;
```

Figure 4 Format

```
options linesize=250 pagesize=60;
proc report data=rep nowd missing headline headskip split='|';
   columns protocol memname, ( modate nlobs nvar dummy) ;

   define protocol/group order=data width=32 id ;
   define memname/across ;
   define modate/display style=[background=cutdt.];
   define nlobs/display "Number Records" style=[background=clrs.];
   define nvar/display "Number Variables" style=[background=clrs.];
   define dummy/noprint;

run;
```

Figure 5 Create output using PROC report

	AE			AUA		
Protocol	Date Modified	Number Records	Number Variables	Date Modified	Number Records	Number Variables
A1201	06MAY11:06:46:38	628	68	06MAY11:06:46:38	2959	62

Figure 6 Horizontal output

```
ods tagsets.excelxp options(
     absolute_column_width="12, 10, 16, 10, 10, 16, 20, 10, 10, 10, 10, 10, 10, 10"
     embedded_titles='yes'
     print_header="&sysdate9.: dataset (/&data_path.) matrix (&sysdate.)"
     autofilter='yes'
     frozen_headers='yes'
     frozen_rowheaders='2'
     sheet_name="List"
     sheet_label="List"
 )
 ;
 options pagesize=32000;
 proc report data=rep nowd headline headskip missing;
   column protocol memname modate nlobs nvar crdate filesize
          maxvar maxlabel datarep num_character num_numeric z;

   define protocol/display "Study" style=[background=#CCFFFF];
   define memname/display "Dataset" style=[background=#FFFF99];
   define modate/display style=[background=cutdt.];
   define nlobs/display "Number Records" style=[background=clrs.];
   define nvar/display "Number Variables" style=[background=clrs.];
   define z/display noprint computed;

 compute z;
       if memname eq 'RAND_ERR' then do;
         call define('NLOBS', 'style', 'style=[background='||put(nlobs, rnderr.)||']');
       end;
   endcomp;
 run;

  options pagesize=60;
 ods tagsets.excelxp close;
```

Figure 7 Vertical output code

Study	Dataset	Date Modified	Number Records	Number Variables
A1201	AE	06MAY11:06:46:38	628	68
A1201	AUA	06MAY11:06:46:38	2959	62
A1201	BLD	06MAY11:06:46:39	6203	64
A1201	BP	06MAY11:07:15:39	5070	33
A1201	CD	06MAY11:07:05:19	1097	58
A1201	COM	06MAY11:06:46:40	0	0
A1201	CT	06MAY11:07:05:19	470	43
A1201	DE	06MAY11:06:39:07	609	76
A1201	ECG	01AUG08:04:22:13	503	37
A1201	EFF	06MAY11:07:15:38	2133	44
A1201	EFF2	06MAY11:07:15:39	418	43
A1201	LB	06MAY11:07:21:34	32020	50
A1201	MH	06MAY11:07:10:45	1168	30
A1201	PD	06MAY11:06:46:40	1168	30
A1201	PE	06MAY11:07:10:44	7372	42
A1201	PK	06MAY11:07:05:19	4022	46
A1201	QE	06MAY11:07:10:45	1229	49
A1201	RANDO	06MAY11:07:15:39	418	15
A1201	RAND_ERR	06MAY11:09:11:56	1	6
A1201	SB	06MAY11:06:39:30	609	36
A1201	SCDG	01AUG08:04:30:48	522	38
A1201	TD	06MAY11:07:15:39	839	111

Figure 8 Vertical output