PharmaSUG 2013 - Paper PO09

Efficient Statistical Review Using the ExcelXP Tagset

Bradford J. Danner, inVentiv Health Clinical, Tennessee

ABSTRACT

Working with clinical trial data, and assisting in the preparation of programming deliveries needed for clinical study reports, as statisticians we are often confronted with the review of many outputs. Part of such a review often requires spot checking of numbers. Programmers are frequently required to produce series of individual outputs, created in an iterative manner from one or few centralized programs, so that all are cosmetically consistent. Alternatively, as a statistician, our goal is to efficiently review several related outputs from one or few comparative outputs. We have found that the use of the REPORT and PRINT procedures presented using the ExcelXP tagset, and some of the built-in options, provides an effective tool to centralize and accelerate review of many outputs produced from a SAS program or SAS dataset.

INTRODUCTION

Several levels of quality control are often required to produce an output for inclusion in a clinical study report. Whether a given output is a table, listing, or figure for example, can dictate how to best proceed with verification and QC. Across different organizations, similar processes are required to complete a thorough QC of outputs. However, a generalized approach which seems common across the industry seems to encourage peer programmer validation or double programming, and a subsequent step, or two, of review performed by a statistician, or project leader. Since there usually are fewer statisticians assigned to a work on a given project, relative to SAS programmers, they can often be inundated with many outputs, and have less time in which to review, as compared to programming resources.

To complicate the issue further, as work progresses from initial planning stages to final reporting, data can change. A flexible manner in which to confirm high quality over time can be a very important tool, especially as data changes. Similarly, as changes are requested, a manner in which to perform quality control iteratively becomes crucial, in order to be efficient with time. Furthermore, since one or few statisticians may be tasked to review the possibly many outputs going into a report, many of which are likely very similar in nature, a tool to centralize review of the many files in one place or destination also fosters efficiency and effectiveness. As both a programmer and statistician, often a goal is to simplify the task of review and quality control by centralizing comparative outputs in fewer files or locations, as opposed to many.

Often reviewers like to visually inspect the entire dataset, either through simple prints (PROC PRINT) to the output window, or using a viewer of SAS datasets. While both the output window available during a SAS session and a data viewer utility are powerful tools, interface and manipulation of the information often seems awkward. Alternatively, many are accustomed to looking at numeric information in spreadsheets, such as Excel files, where interface and manipulation of numbers is less awkward and a more hands-on approach is possible. The proposed quality control methodology illustrated here demonstrates a fast manner in which to review numbers of several report outputs by centralizing comparative output created by SAS procedures and/or Data steps into one, or relatively few, Excel worksheets using the ExcelXP tagset.

METHODS AND RESULTS

Several aspects of an output, or collection of similar outputs, must be considered when performing a quality review. The focus of this approach is verification of summaries (counts, percentages, test statistics, and p-values, for example), and efficient organization of comparative output for various subgroups, populations, or study phases. The following is a proposed straightforward process for assessing a collection of outputs and quickly compiling comparative output in a centralized manner within a spreadsheet:

1. Determine number of outputs and distinguishing characteristic(s)
2. Pre-process data as necessary, to include merges, data steps, and manipulation to aid in review
3. Process data with appropriate statistical procedures
4. Collect output datasets and perform any merging or cosmetic manipulations to facilitate review more efficiently
5. Direct final datasets to centralized review file using ODS TAGSET.EXCELXP

This process is illustrated briefly using an adverse event incidence table type that reflects a scenario that commonly can occur during the course of analysis of clinical study safety data.

EXAMPLE – ADVERSE EVENT INCIDENCE

Clinical study reports commonly require summarization of adverse events, and will often call for separate tables for different subgroups or type of events, or for different phase of the study itself. Consider the following mock table shell example:

<*Subset*> Adverse Events– <*Phase or Period*>

	Treatment Group (as Administered)								
	Group 1 Na=nnn			Group 2 Na=nnn			Overall Na=nnn		
System Organ Class\ Preferred Term	No. of Subjectsb	%	No. of Eventsc	No. of Subjectsb	%	No. of Eventsc	No. of Subjectsb	%	No. of Eventsc
Any event	nnn	pp.p	nnn	nnn	pp.p	nnn	nnn	pp.p	nnn
System organ class	nnn	pp.p	nnn	nnn	pp.p	nnn	nnn	pp.p	nnn
Preferred term	nnn	pp.p	nnn	nnn	pp.p	nnn	nnn	pp.p	nnn
Preferred term	nnn	pp.p	nnn	nnn	pp.p	nnn	nnn	pp.p	nnn
Preferred term	nnn	pp.p	nnn	nnn	pp.p	nnn	nnn	pp.p	nnn
Preferred term	nnn	pp.p	nnn	nnn	pp.p	nnn	nnn	pp.p	nnn
Preferred term	nnn	pp.p	nnn	nnn	pp.p	nnn	nnn	pp.p	nnn

a. The values in this row are used as the denominators for percentages.
b. Number of subjects reporting at least 1 event.
c. The total number of events. Multiple events may be reported by 1 subject.

Suppose the reviewers are interested in reviewing subsets of adverse events separately according to a level of relatedness, severity or seriousness. The following 8 levels of subset are examples I recently encountered on a relatively simple study: all, related, severe, serious, serious with death as outcome, both related and severe, both related and serious, and not serious. Furthermore, this same study had 4 definitive periods of interest. Therefore, using this simplified example required statistical review of 32 outputs.

The two crucial steps necessary for pre-processing the adverse event information, once the actual events were combined with subject level data, was establishment of an identifier for the phase or period of interest, and combination of specifically targeted datasets to address the groupings of interest. To generalize establishment of a period identifier, in a simplified manner:

```
data one;
set two;
  if date1 <= AESTDT < date2 then period=1;
  else if date2 <= AESTDT < date3 then period=2;
  else if date3 <= AESTDT < date4 then period=3;
  else if date4 <= AESTDT < date5 then period=4;
run;
```

To group the data as needed, several steps are required to count events at the individual preferred term level, preferred term collapsed into the system organ class level, again collapsed into the Any Event level, and further segmented into the groupings of interest. To generalize the steps taken for this aspect of pre-processing:

1. Start with all the AE data, where SOC and PT are populated, and overwrite the coded information in order to collapse events into the Any Event level:

```
data all; set aeprep;
  output;
  if soc not in ('NONE',' ')  then do;
    soc="Any event"; pt='Any event'; output;
  end;
run;
```

2. Create subgroup datasets as needed, from the first dataset (illustrated with related and severe adverse events only), by overwriting coded information of events that do not satisfy the condition of interest:

```
data related; set all;
  output;
  if soc ne ' ' and upcase(related) ne 'YES' then do;
    soc=' '; pt=' ';
  end; output;
run;

data severe; set all;
  output;
  if soc ne ' ' and upcase(severe) ne 'SEVERE' then do;
    soc=' '; pt=' ';
  end; output;
run;
```

3. Combine the subgroup datasets into one analysis dataset, while simultaneously creating an indicator category to denote the subgroup:

```
data pre_stat1;
  length category $20.;
  set   all (in=f1) sev (in=f2) ser (in=f3) rel (in=f4) ns (in=f5)
        relser (in=f6) serd (in=f7) relsev (in=f8)
  if f1 then category='AE General';
  else if f2 then category='AE Severe';
  else if f3 then category='AE Serious';
  else if f4 then category='AE Related';
  else if f5 then category='AE NS';
  else if f6 then category='AE RelSer';
  else if f7 then category='AE Serious Death';
  else if f8 then category='AE RelSev';
run;
```

4. Overwrite the coded information again in order to collapse preferred term coding into the system organ class level, and establish an Overall group:

```
data pre_stat2;
  set pre_stat1;
  output;
  pt=strip(soc); output;
run;

data pre_stat3;
  set pre_stat1;
  output;
  group='Overall'; output;
run;
```

After performing these pre-processing steps, the number of events, and subjects with events, can be counted as needed, using an appropriate SAS procedure. For tables of this nature, a SQL or FREQ procedure(s) can certainly obtain the necessary statistics. A simplified example using SQL is as follows:

```
proc sql;
  create table denoms as
  select tgroup, count(distinct(subjid)) as denom
  from work.pre_stat3
  group by tgroup;

  create table nsubjects as
  select tgroup, category, period, soc, pt, count(distinct(subjid)) as nsubjects
  from work.pre_stat3
  where pt ne ' '
  group by tgroup, category, period, soc, pt;
```

```
    create table nevents as
    select tgroup, category, period, soc, pt, count(pt) as nevents
    from work.pre_stat3
    where pt ne ' '
    group by tgroup, category, period, soc, pt;
  quit;
```

The final steps prior to presenting in a centralized spreadsheet are essentially cosmetic in nature – merging of datasets, calculations of percentages dependant upon study guidelines, sorting to accommodate transposition of datasets to cross-reference with actual outputs, and will not be illustrated here. However, the categories that were established during pre-processing steps were designed such that output could be contained in one Excel worksheet within a spreadsheet, and quickly reviewed when a simple AutoFilter is in place:

```
ods tagsets.Excelxp file="psug2013_po09.xls" style=normal
options(frozen_headers='Yes' autofilter="All" sheet_name="AE Incidence"
absolute_column_width="20" );

  proc print data=final noobs;
   var category period soc pt group_1 group_2 total;
  run;

ods tagsets.Excelxp close;
```

	A	B	C	D	E	F	G
1	category	period	soc	pt	Group_1	Group_2	Total
2	Sort Ascending / Sort Descending / (All) / (Top 10...) / (Custom...) / AE General / AE NS / AE Related / AE Serious / AE Severe	1	Any event	Any event	123 (77.4%) 746	126 (79.7%) 700	249 (78.5%) 1446
3		1	Blood and lymphatic system disorders	Blood and lymphatic system disorders	1 (0.6%) 1	1 (0.6%) 1	2 (0.6%) 2
4		1	Blood and lymphatic system disorders	Iron deficiency anaemia	1 (0.6%) 1	1 (0.6%) 1	2 (0.6%) 2
5		1	Eye disorders	Eye disorders	9 (5.7%) 9	18 (11.4%) 19	27 (8.5%) 28
6		1	Eye disorders	Conjunctivitis	6 (3.8%) 6	14 (8.9%) 14	20 (6.3%) 20
7		1	Eye disorders	Eye discharge	3 (1.9%) 3	3 (1.9%) 3	6 (1.9%) 6
8		1	Eye disorders	Keratitis		1 (0.6%) 1	1 (0.3%) 1
9	AE General	1	Eye disorders	Ocular hyperaemia		1 (0.6%) 1	1 (0.3%) 1
10	AE General	1	Gastrointestinal disorders	Gastrointestinal disorders	18 (11.3%) 23	17 (10.8%) 20	35 (11.0%) 43
11	AE General	1	Gastrointestinal disorders	Constipation	8 (5.0%) 8	6 (3.8%) 6	14 (4.4%) 14
12	AE General	1	Gastrointestinal disorders	Diarrhoea	7 (4.4%) 9	9 (5.7%) 11	16 (5.0%) 20
13	AE General	1	Gastrointestinal disorders	Dyspepsia	1 (0.6%) 1		1 (0.3%) 1
14	AE General	1	Gastrointestinal disorders	Haematochezia	1 (0.6%) 1	1 (0.6%) 1	2 (0.6%) 2
15	AE General	1	Gastrointestinal disorders	Ileus	1 (0.6%) 1		1 (0.3%) 1
16	AE General	1	Gastrointestinal disorders	Inguinal hernia		1 (0.6%) 1	1 (0.3%) 1
17	AE General	1	Gastrointestinal disorders	Intussusception	1 (0.6%) 1	1 (0.6%) 1	2 (0.6%) 2
18	AE General	1	Gastrointestinal disorders	Upper gastrointestinal haemorrhage	1 (0.6%) 1		1 (0.3%) 1
19	AE General	1	Gastrointestinal disorders	Vomiting	1 (0.6%) 1		1 (0.3%) 1
20	AE General	1	General disorders and administration site conditions	General disorders and administration site conditions	13 (8.2%) 18	17 (10.8%) 22	30 (9.5%) 40
21	AE General	1	General disorders and administration site conditions	Application site erythema		1 (0.6%) 2	1 (0.3%) 2

AE Incidence

Select destination and press ENTER or choose Paste NUM

Figure 1. Screen Print of Centralized AE Incidence Review Output

Using the autofilter option in the TAGSET.EXCELP output destination allows the reviewer to quickly toggle through many combinations of categories and periods, up to 32 possibilities in this example, for which individual AE tables were produced.

This approach can easily be generalized to work with all types of data summarized in the course of compiling a clinical study report where essentially multiple versions of the same table are required. By centralizing in one or few files, the statistical reviewer can spot check the numbers of programmer reviewed and validated outputs relatively quickly. Reviewers could also consider supplemental customization of review output through the use of proc REPORT, rather then the simpler PRINT output illustrated here.

CONCLUSION

The technique illustrated here offers users the benefit of centralizing the review of many outputs in one place, and with a little pre-planning during preliminary SAS coding, information can be laid out in a manner which allows one to

efficiently review from different perspectives through simple autofilters. While this method will certainly not provide a replacement for full programmer validation and direct file comparison, or PROC COMPARE at the SAS dataset level, it does give final reviewers the ability to quickly spot check numbers across many outputs, all in one localized file.

A concern with using this technique is that the format of the comparative output, the spreadsheet, is obviously modifiable and less secure in that manner, as compared to a SAS dataset. However, assuming peer validation with aspects of double programming are already established, the flexibility allowed is actually preferred and more efficient at the secondary/tertiary level of quality control statisticians often conduct. Admittedly, the example presented in this paper is simplistic in nature, but the wealth of options available in the ExcelXP tagset, coupled with the ability to quickly increase functionality within Excel offers reviewers many options for possible enhancements to meet their quality control needs.

AKNOWLEDGEMENTS

I would like to thank my colleagues in the Biostatistics and Statistical Programming groups at inVentiv Health Clinical who provided comments and feedback. Their insights and guidance are appreciated.

CONTACT INFORMATION

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

Name: Bradford J. Danner
Enterprise: inVentiv Health Clinical
Address: 1787 Sentry Parkway West, Suite 300, Building 16
City, State ZIP: Blue Bell, PA 19422 USA
Work Phone: (615) 302-3608
E-mail: brad.danner@inventivhealth.com