PharmaSUG2013 – Paper BB04

Enhancements to Basic Patient Profiles

Scott Burroughs, GlaxoSmithKline, Research Triangle Park, NC

ABSTRACT

Patient Data Viewers are becoming more prevalent in the pharmaceutical industry, but not all companies use them nor need them for all situations. Old-fashioned patient profiles still have use in today's industry, but how can they be enhanced?

Missing data, bad data, and outliers can affect the output and/or the running of the program. Also, relying on analysis data sets that need to be run first by others can affect timing (vacations, out-of-office, busy, etc.). As always, there are things you can do to make them look prettier in general. This paper will show how to solve these issues and make the program more robust.

INTRODUCTION

A couple of years ago, I was asked to do the patient profiles for the Safety Review Team monthly updates of a phase IIb trial. To save work, another team in our therapeutic area gave me a patient profile program they used for another study. While we had just started use of a more automatic Patient Data Viewer software around the company, not everyone has training and/or license for it, so we still use basic patient profiles for periodic review of demographic and safety data.

Upon looking at the program, it was basically one PROC REPORT for each data module, output using ODS PDF, all inside a macro that was run for each subject separately. Upon trying to adapt this program to my studies, I found several things I felt could be improved about the program, to account for several factors.

1. The macro output each subject to a separate PDF file, meaning there would be potentially several hundred different files, depending on the size of the study. The statistician on the other study was zipping all the files 'together' in one package to send out to the team, but they were still separate files. Talk about a lot of opening and closing of files! I wanted the option to keep this format *and* output all the subjects into one PDF file. Browsing through the subjects would be much easier.

2. The program was using ODS PDF to output to PDF, but wasn't using the available functionality to create bookmarks to easily go from page to page, even within the individual subjects. One could go from Demographics to Labs 5 pages down with one easy click, instead of paging or scrolling down. Also, within the one large PDF file I wanted, custom bookmarks could include subject number and even investigator name, for easier navigation.

3. The macro call per subject was wrapped around everything except the PROC FORMAT call, meaning that data was being read in from the shared drive for every subject and every data module….that's a lot of I/O! Granted each data read had a WHERE clause subsetting on just one subject at a time, it still was (# of subjects x # of data modules) calls to the shared drive. I wanted to move all of the reading in of the data (and subsequent data manipulation) to outside the macro calls so it would only be done once for the entire study. This would mean the number of calls to the shared drive would only equal the number of data modules.

4. The biggest undertaking could potentially have been the most important. The original program used mostly analysis data sets, instead of the 'raw' (CRF) data sets, so all of the calculated variables and decodes were included already. The drawback was that someone else was most likely responsible for keeping those analysis data sets updated each time the patient profiles were needed (which was once a month). There were definite possibilities that at least one of those responsible for keeping the analysis data sets updated could have been unavailable (vacation, meetings, out-of-office, too busy, etc.) to do so. It's also possible that the programs that created the analysis data sets might not have even be created yet!

   I wanted to get rid of this dependency on others, because it could potentially be a show-stopper. The reason it would be a big undertaking is that I had to find the code for every data module that

created each analysis data set. Some of these were 'buried' inside standard macros, though some were simple enough that it was just a simple decode creation (a 'Put' and a format).

5. There was also the potential of data modules having no data for a subject (AE, CONMEDS, etc.), which would either print PROC REPORT variable titles only with no data or not print anything at all, with either result potentially confusing the reader. My solution was to make sure that something descriptive was printed out for each data module with no observations.

6. The original program printed out all labs in one PROC REPORT call, in the alphanumeric order of the coded variable of each lab test parameter. This means that all Chemistry, Haematology, Urinalysis, etc. parameters were intermixed in the same listing. I wanted each section to be printed separately (and also use the same lab macro call!).

7. Unfortunately, ODS PDF overrides some SAS formatting statements and does its own thing. Some variable labels get wrapped unintentionally, even when the specified format is longer and there is plenty of room. This uses up more space unnecessarily (prints more lines per module) and adds pages to the file. I found a special command to prevent this from happening.

METHODS

1. In order to also have one PDF file that included all subjects, I needed to move a few lines around and comment out some commands in the ODS PDF statement.

Original code:

```
%macro pat(num)/parmbuff;

ods PDF file="~/XYZPDQ123_prof_&num..pdf"
     startpage=no COMPRESS=9 NOTOC BOOKMARKGEN=NO BOOKMARKLIST=NONE;

[data calls + PROC REPORTS]

ODS PDF close;

%MEND;

%MACRO loop;
    %DO i = 1 %TO &numsubj;
        %pat(&&subj&i);
    %END;
%MEND loop;

%loop;
```

New code:

```
%macro pat(num=,star=)/parmbuff;

&star ods PDF file="&g_outfile._&num..pdf"
     startpage=no COMPRESS=9 NOTOC BOOKMARKGEN=NO BOOKMARKLIST=NONE;

[data calls + PROC REPORTS]

&star ODS PDF CLOSE;

%mend pat;

%MACRO loop;
    %DO i = 1 %TO &numsubj;
        %pat(&&subj&I,star=);
```

```
    %END;
%MEND loop;

%loop;

ods PDF file="&g_outfile..pdf"
     startpage=no COMPRESS=9;*NOTOC BOOKMARKGEN=NO BOOKMARKLIST=NONE;

%MACRO loop_;
    %DO i = 1 %TO &numsubj;
        %pat(num=&&subj&i,star=*);
    %END;

    ODS PDF CLOSE;
%MEND loop_;

%loop_;
```

Now the ODS PDF CLOSE is outside the %pat call, so the same (only) PDF is open and written to for each &&subj&i. Also, the bookmarks are created for each subject and data module (discussed in more details in next step).

If all subjects in one file is too big (say, you have several hundred subjects), you could break it up into smaller files, say, by investigator/site. Or you could subset for only those subjects of a special interest.

2. Discussed in the previous step is the code needed to have the bookmarks turned on (though that code alone would not have any custom bookmarks).

When bookmarks are turned on, each PROC run will provide a default bookmark. In order to provide a custom bookmark, we must first turn off the default: `ODS NOPTITLE;  *** no PROC title ***;`

The command ODS PROCLABEL “title”; prior to any PROC provides a custom bookmark for that PROC.

In order to add this to the original %pat macro, where the individual subject files did NOT have bookmarks, but the overall big file does, we must make sure previous commands with &star are turned ‘off’ (&star=*) and turn on new statements with &star2=;

New code:

```
%macro pat(num=,star=,star2=) / parmbuff;

&star2 ods noptitle;
&star2 ods pdf startpage=now;  *** make sure each new subject starts on new
page ***;

&star2 ods proclabel "Subject #&num Demographics and Background
Characteristics";

proc report data=demo2(where = (subjid=&num)).....
%mend pat;

%MACRO loop;
    %DO i = 1 %TO &numsubj;
        %pat(&&subj&I,star=,star2=*);
    %END;
%MEND loop;

%loop;

ods PDF file="&g_outfile..pdf" /* does not have _&num in file name */
     startpage=no COMPRESS=9;*NOTOC BOOKMARKGEN=NO BOOKMARKLIST=NONE;
```

```
%MACRO loop_;
    %DO i = 1 %TO &numsubj;
        %pat(num=&&subj&i,star=*,star2=);
    %END;

    ODS PDF CLOSE;
%MEND loop_;

%loop_;
```

3. Moving the shared area data reads to outside the main macro %pat is simple enough.

New code:

```
[data calls]

%macro pat(num=,star=)/parmbuff;

[PROC REPORTS]

&star ODS PDF CLOSE;

%mend pat;
```

4. And now, the hard part, changing the code from using analysis data sets to CRF data sets. This involved going into each analysis data set creation program and finding the correct code needed. Some of the data sets were simply adding code with format statements to create decodes. Maybe a study day variable based on (module date-treatment start date)+1. There were a few modules that needed delving into standard macro code (AEs and CONMEDs, for example) and finding just the specific code needed for the variables needed for printing. I wanted to take out the extraneous code to save in processing time. I will not show any of this code, as it is rather lengthy and much of it is proprietary.

5. In order to fix the problem of missing/no data in a specific module (say, if subject has no AEs or CONMEDs, for example), I'd need code to check for missing data and then code to print out a note saying that there was no data/observations in that data module for that subject. Also, depending of which module it was, the note printed out might need to vary a bit. Sometimes a subject may not have had something collected, but sometimes an event may not have occurred.

   If trying to print directly from a data set where no observations are selected, nothing will print at all, which is not informative.

   If printed as an empty data set (no observations selected from one data set merged with one that does) this situation is printed:

Adverse Events									
Verbatim / Preferred Term	Start Date	Start Day	End Date	Duration	Toxicity Grade	Outcome	Relationship	Serious?	Withdraw?
/	.	.	.						

   With my new code below, I get a more informative output:

Adverse Events
No Adverse Events Reported

   Another example:

Labs - Urinalysis
No Urinalysis Data Collected

New code:

```
%MACRO empty(ds);

%if %std_nobs(&ds)=0 %then %do;  *** %std_nobs is standard macro we have that
tests if data set is empty or not ***;
**** I believe %if %sysfunc(exist(&ds)) %then %do works for when no data set is
present ***;

data mt;
  length x $132;
  %if "&subtyt"="Adverse Events" or "&subtyt"="Concomitant Medications" or
[others] %then %do;
    x="No &subtyt Reported" ;
  %end;
  %else %if "&subtyt"="Disposition" or "&subtyt"="Clinical Chemistry" or
[others] %then %do;
    x="No &subtyt Data Collected" ;
  %end;
  %else %if "&subtyt"="Mutations" %then %do;
    x="No &subtyt Found" ;
  %end;
  %else %do;
    x="No &subtyt Events Reported" ;
  %end;
  output;
run;

proc report
    data = mt nowd
    contents=''
    ;
  %if "&subtyt"="Immunology" or "&subtyt"="Clinical Chemistry" or [others]
%then %do;
    column ("Labs - &subtyt" x);
  %end;
  %else %do;
    column ("&subtyt" x);
  %end;
  define x / width=132 "";
run;

%end;

%MEND empty;
```

6. The program I inherited had the lab category sections (Chem/Haem/Urin/Immun/etc) all in one section, with no designations for the sections. They were sort in lab category/lab test code order, but nothing distinguished them.

   Here's a pic of the labs, with Chemistry parameters immediately followed by Haemotology parameters:

Labs								
Lab Test	Lab Date	Study Day	Visit	Lab Result	Lab Unit	Normal range lower limit	Normal range upper limit	Toxicity Grade
Triglycerides	11FEB2011	1	DAY 1	2.47	MMOL/L	0	2.24	
	03MAY2011	82	WEEK 12	2.7	MMOL/L	0	2.24	
	26JUL2011	166	WEEK 24	1.43	MMOL/L	.	.	
Urea/BUN	07JAN2011	-35	SCREENING	4.8	MMOL/L	2.5	9	
	11FEB2011	1	DAY 1	4.1	MMOL/L	2.5	9	
	11MAR2011	29	WEEK 4	5.2	MMOL/L	2.5	9	
	08APR2011	57	WEEK 8	5.1	MMOL/L	2.5	9	
	03MAY2011	82	WEEK 12	3.4	MMOL/L	2.5	9	
	31MAY2011	110	WEEK 16	4.4	MMOL/L	2.5	9	
	26JUL2011	166	WEEK 24	5.3	MMOL/L	2.5	9	
International Normalized Ratio (PT)	07JAN2011	-35	SCREENING	0.9		0.9	1.1	
Prothrombin Time	07JAN2011	-35	SCREENING	9.7	SEC	9	11.5	
Basophils	07JAN2011	-35	SCREENING	0.01	GI/L	0	0.2	
	11FEB2011	1	DAY 1	0.01	GI/L	0	0.2	
	11MAR2011	29	WEEK 4	0.03	GI/L	0	0.2	
	08APR2011	57	WEEK 8	0.01	GI/L	0	0.2	
	03MAY2011	82	WEEK 12	0.01	GI/L	0	0.2	
	31MAY2011	110	WEEK 16	0.01	GI/L	0	0.2	
	26JUL2011	166	WEEK 24	0.01	GI/L	0	0.2	
Basophils (percentage)	07JAN2011	-35	SCREENING	0.3	%	0	2	
	11FEB2011	1	DAY 1	0.3	%	0	2	
	11MAR2011	29	WEEK 4	0.7	%	0	2	

I wanted to do separated lab sections (Chem/Haem, etc), each with their own titles. After sorting by subject, lab section, lab test code, and visit, I started the %labb macro.

New Code:

```
%MACRO labb(sub=,typ=or);  *** sub=sectin, typ=unit type (orig or SI) ***;

&star2 ods proclabel "Subject #&num Labs - &subtyt";

data lab3;
  set lab2(where = (lbcat="&sub")) ;   *** subset each section ***;
run;

%empty(lab3);  *** check for empty/missing for each section ***;

proc report
    data = lab3 nowd spacing=2 headline headskip split="~" missing
    contents=''
    ;
  column ("Labs - &subtyt" [lab vars] lb&typ.unit lb&typ.nrlo lb&typ.nrhi [lab var]);

  [lab var define calls]
```

```
  define lb&typ.nrlo / group order=internal format=best. &asis width=13 "Normal
range~lower limit";
  define lb&typ.nrhi / group order=internal format=best. &asis width=13 "Normal
range~upper limit";
  define lb&typ.unit / group order=internal format=$10.   "Lab~Unit"                ;

  [lab var define call]

run;
%MEND labb;


%let subtyt=Immunology;
%labb(sub=IMMU,typ=st);

%let subtyt=Clinical Chemistry;
%labb(sub=CHEM);

%let subtyt=Haemotology;
%labb(sub=HAEM);

%let subtyt=Urinalysis;
%labb(sub=URIN);
```

And, as shown in the previous section, I had the empty data set macro run on each section.

7. I like to use up the entire width of the page, as each wrapped word loses a line per page and the potential to wrap increases. ODS PDF overrides some SAS settings/formats and wraps some variable labels when there is still plenty of room for the given linesize. As usual, I did an internet search, and eventually found a command that will keep the widths as originally stated during an ODS PDF/PROC REPORT call (in the DEFINE statement). That command is the STYLE= option. Not surprising, but the command needed in this case was **STYLE=[ASIS=ON].** Yes, that means you want to output 'as is'. I guess they had to come up with this statement for those statements that override defaults (even when they likely shouldn't). Another option I wanted for this command was the ability to center the results (or at least *keep* the centering originally asked for). Like other uses, it is **JUSTIFY=CENTER.** So the final command is **STYLE=[ASIS=ON JUST=CENTER].** I set these two as macro variables to use easy throughout the program.

Here is what it looks like with space-losing wrapping:

Vital Signs								
Date	Study Day	Visit	Height (cm)	Weight (kg)	Systolic Blood Pressure (mm/Hg)	Diastolic Blood Pressure (mm/Hg)	Heart Rate (beats/min)	BMI
10NOV2010	-33	SCREENING	169.0	56.5	130.00	90.00	72.00	19.8
13DEC2010	1	DAY 1	.	55.8	116.00	78.00	72.00	19.5
24JAN2011	43	WEEK 4	.	.	116.00	80.00	88.00	.

When using **STYLE=[ASIS=ON ON JUST=CENTER],** (and put a Split value in), it saves to rows per page as such:

Vital Signs								
Date	Study Day	Visit	Height (cm)	Weight (kg)	Systolic Blood Pressure (mm/Hg)	Diastolic Blood Pressure (mm/Hg)	Heart Rate (beats/min)	BMI
13SEP2012	-21	SCREENING	177.0	63.0	130.00	82.00	62.00	20.1
04OCT2012	1	BASELINE - DAY 1		63.0	128.00	78.00	65.00	20.1
05OCT2012	2	DAY 2		63.0	120.00	78.00	65.00	20.1
11OCT2012	8	WEEK 1		63.0	120.00	78.00	58.00	20.1
16OCT2012	13	WEEK 2		62.0	120.00	78.00	63.00	19.8

Additional Enhancements

There were a few other things I did to 'improve' the originals. The original title started as such:

Subject 9973 : 39 year old

Where there was no description of the profiles as profiles and there was no mention of the protocol #. I added an additional title line as such:

Individual Patient Profiles for Subjects in

Also, at the right of the titles, it looked as such, with a default SAS page counter and date/time stamp, which wrapped:

00mg BID) **22**
22:39 Thursday, March 28, 2013

I added a 'Page 1 of' counter on the same line as the new title above:

Page 16 of 3525

And took the SAS date/time stamp and page counter out with `options nodate nonumber`. The date/time stamp was included in a footnote at the bottom, which has the program name/USERID in it, as well.

In the Medical History section, the conditions were sorted by Classification, alphabetically, regardless of Current/Pass status:

Medical History		
Classification	Specific Conditon	Current/Past
Blood and lymphatic system disorders		Current
Cardiac disorders	Hypertension	Past
Eye disorders		Current
Gastrointestinal disorders	Gastrointestinal disorder NOS	Current

I sorted by Current/Past (it's a time variable, in a way) and did a 'first.dot' grouping on it:

Medical History	
Current/Past	**Condition**
Current	Anxiety
	Depression
	Hypertension
	Insomnia
	Psychiatric symptom
Past	Headache
	Hypercholesterolaemia
	Suicidal ideation

A recent study had several personal history sections, a few of which only would be asked if a qualifying liver event occurred during the study. The default sections were these 4:

Surgery, Family Cardiac, and Smoking History			
Coronary Artery Bypass	**Coronary Angioplasty**	**Family history of premature coronary artery disease in women <65 or men <55**	**History of Tobacco Use**
No	No	No	Never smoked

But some subjects were going to be asked about Current Tobacco Use during the study, so this section was a logical place to put it, instead of being its own new section. I just added it in:

Surgery, Family Cardiac, and Smoking History				
Coronary Artery Bypass	**Coronary Angioplasty**	**Family history of premature coronary artery disease in women <65 or men <55**	**History of Tobacco Use**	**During-study Current Tobacco Use**
No	No	No	Current smoker	Former smoker

But that meant the 3rd column title had to wrap, so I had options for it, depending on if the During-study Current Tobacco Use question was present.

Another liver event-dependent question was asking about current alcohol use and this (again) seemed like the place to put it. That meant options for the overall section title and adding new columns:

Surgery, Family Cardiac, and Smoking History and Current Alcohol Use					
				Liver Event-Spurred Current Alcohol Use Questions	
Coronary Artery Bypass	**Coronary Angioplasty**	**Family history of premature coronary artery disease in women <65 or men <55**	**History of Tobacco Use**	**Does subject consume alcohol?**	**Avg.units of alcohol consumed weekly**
No	No	No	Never smoked	Yes	1

Code for all of this here:

```
/* Surgery and Family Cardiac History */

%let smkc=;
%let alc=;
%let sual_=;
%let sualunwk_=;

data famsurg2(drop=visit:);
  merge famsurg(
                where = (subjid=&num)
                in    = infs
               )
        smoke_f(where = (subjid=&num) in=insmkf)
```

```
        smoke_c(where = (subjid=&num) in=insmkc)
        alc_y  (where = (subjid=&num) in=inalc)
        onset  (keep  = subjid in=inonset)
        ;
  by subjid;
  if inonset and (infs or insmkf or insmkc or inalc);
  if insmkc then call symput('smkc','smoke4');
  if inalc then do;
    call symput('alc','alc_y2');
    call symput('sual_','sual');
    call symput('sualunwk_','sualunwk');
  end;
run;

%let fho=Family history of premature coronary artery disease in women <65 or
men <55;

%let smkc_star=*;
%if "&smkc"^="" %then %do;
  %let smkc_star=;
  %let fho=Family history of premature coronary~artery disease in women <65 or
men <55;
%end;
%let alc_star=*;
%if "&alc"^="" %then %do;
  %let alc_star=;
%end;


%let subtyt=%str(Surgery, Family Cardiac, and Smoking History);

%let col_star=;
%let alc_star=*;
%if "&alc"^="" %then %do;
  %let alc_star=;
  %let col_star=*;
  %let subtyt=%str(Surgery, Family Cardiac, and Smoking History and Current
Alcohol Use);
  %let fho=Family history of premature coronary~artery disease in women <65 or
men <55;
%end;

&star2 ods proclabel "Subject #&num &subtyt";

PROC REPORT
    DATA=famsurg2
    NOWD COLWIDTH=40 SPACING=3 HEADLINE HEADSKIP SPLIT='~' MISSING
    contents=''
    ;
&col_star COLUMN ("&subtyt" surg1 surg2 fhoccur smoke1 &smkc );
&alc_star COLUMN ("&subtyt" surg1 surg2 fhoccur smoke1 &smkc ("Liver Event-
Spurred~Current Alcohol~Use Questions" &sual_ &sualunwk_) );
...
[define statements]
...
&smkc_star DEFINE smoke4  / GROUP ORDER=INTERNAL format=$susscds. &asis_c
width=30 "During-study~Current Tobacco Use";
&alc_star  DEFINE sual     / GROUP ORDER=INTERNAL format=$zyesnos. &asis_c
width=30 "Does subject~consume alcohol?";
&alc_star  DEFINE sualunwk / GROUP ORDER=INTERNAL format=2.        &asis_c
width=30 "Avg.units of~alcohol consumed~weekly";
RUN;
```

Final side notes

Despite the fact that I originally changed the program to print both individual patient profiles (separate files per subject) and one overall version, I eventually dropped the individual profiles, as they weren't being used! Also, when the studies were prematurely cancelled, we ended up deciding to use the patient profiles as the subject data listings. With that, I needed to create our standard set of titles and footnotes (which I do standard now).

CONCLUSION

In the end, I created a much more robust program. I didn't need to depend on others to complete the monthly profiles. I also made it more efficient, organized, and user-friendly.

ADDENDUM

All along, I knew that a primary issue with these profiles was that there is no easy way to QC them. The output is a PDF file and there are no QC data sets to match to. Also, even if there was a way to output the data directly (as presented), the data would be very 'wide', often with no merge variables (such as date) between modules across a subject. However, it is possible to have the data separated by module, sorted by subject.

I discovered that PROC REPORT has an OUT= output data set command in the PROC statement….I never knew this, and other programmers I asked also did not know as well. In order to keep the ordering that is output, I added a SEQ variable to all data sets within subject.

After using `an "out=dem&num"` in the PROC REPORT statement to get one data set per module per subject, I need to loop it through the subjects and through the modules.

```
**** macro to help create QC data set ****;
%macro kyoosee(ds);
data _null_;
  %if %sysfunc(exist(&ds)) %then %do ;
    call symput("&ds","&ds");
  %end;
  %else %do;
    call symput("&ds",''); *** for first pass when data set ***;
  %end;                    *** hasn't been created yet ***;
run;

data &ds;
  retain seq 0;
  set &&&ds &ds.&num(in=insubj) ;  *** append data sets by subjid ***;
  if insubj then do;
    subjid=&num;  *** add subjid to data set ***;
    seq+1;        *** counter to keep obs in output order ***;
  end;
run;

proc delete data=&ds.&num;run ;
%mend kyoosee ;


%macro deedee(ds) ;   *** last macro to add labels and do PROC CONTENTS ***;
  data dddata.&ds;    *** and output QC data sets ***;
    set &ds (drop=_break_);
    label subjid='Subject ID' seq='Sequence/order of observations for QC
purposes';
  run ;
  proc contents data=dddata.&ds;run ;
%mend  ;
```

```
*** Run through all of the modules ***;

%deedee(trt);
%deedee(dem);
%deedee(mh);
```

etc.

REFERENCES

SAS® is a registered trademark of SAS Institute, Inc. in the USA and other countries.

CONTACT INFORMATION

BIOGRAPHY

Scott was a statistician for GlaxoSmithKline for almost 12 years until switching to a full-time programmer role in 2006. He has worked in Research Triangle Park, NC since 1994. He has programmed in SAS extensively since 1992 while at a previous pharmaceutical company. He has a B.S. and an M.S. in Statistics from Virginia Tech.

CONTACT

Scott Burroughs
GlaxoSmithKline
5.3743
5 Moore Drive
Research Triangle Park, NC 27709
scott.m.burroughs@gsk.com