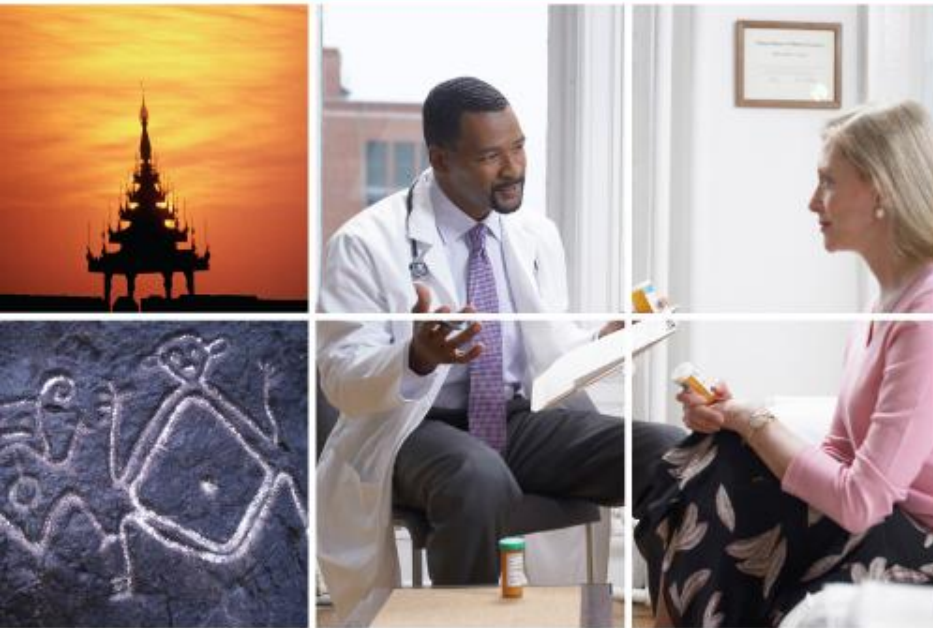




ICON Clinical Research

SAS – How to standardize solutions to recurrent issues

- Introduction to the Authors
- Objectives of the paper
- Examples
- Applications / Conclusions
- Opportunity for Questions

- **Tangi Sanséau**
 - 5+ years experience in Biostatistics
 - 3 working as a Statistician at ICON
 - 2 working as a Programmer at ICON
- **Giuseppe DiMonaco**
 - 2+ years experience in Pharmaceutical Industry
 - 5+ years experience in IT Industry



- How can CROs shorten the time from a Study Start date to its submission to the regulatory authorities while increasing quality and productivity?
- Creating re-usable programs is one of many approaches. This may be difficult because of differences in database structures, statistical requirements etc..
- Developing flexible programs which receives parameters from external files is vital to solve the problems related to these differences.
- In order to find standard solutions, we have segmented problems into sub-problems. This approach is known as modularity.

- We will show you examples on how we deal with handling of:
 - Data transfer comparison.
 - Re-usability of standard listing/tables programming across compounds/sponsors (i.e. Adverse Events, Concomitant Medications...).
 - Missing data points within datasets transfer.

RECURRENT ISSUE 1 - STEP I

SEGMENTING ISSUE INTO SUB-PROBLEMS

- Data transfer comparison
 - Background
 - Notification of data issues to DM
 - Recurrent issue = Re-Transfer
- Segmenting Issue into the following sub-problems:
 - Number of datasets
 - Names and structures of the datasets
 - Names and characteristics of the variables (IDs)

RECURRENT ISSUE 1 - STEP II

SOLVING SUB-PROBLEMS/PROBLEM

- Sub-problem 1: Number of datasets
 - Argument to the macro
- Sub-problem 2: Name and structure of datasets
 - Argument to the macro
 - Specs read directly by SAS
- Sub-problem 3: Names and characteristics of the variables (IDs)
 - Argument to the macro

RECURRENT ISSUE 1 - STEP III

STANDARDIZING THE SOLUTION

- Based on the resolution to the 3 sub-problems, the fastest solution is to read the Datasets Specifications
- However, this solution could then be also seen as a recurrent issue and split into further three sub-issues.
 - Format of the file (Excel, Word, xml...)
 - How are presented the different datasets (Tabs in Excel, different pages in word document...)
 - What is being displayed (derivation, format, length and informat, or just format and length...)
- The advantage of our method is that the resolution of each sub-issue can be worked on in parallel.

RECURRENT ISSUE 2 - STEP I

SEGMENTING ISSUE INTO SUB-PROBLEMS

- Re-usability of Listings/Table programming
 - Background
- Segmenting Issue into the following sub-problems:
 - Name and Number of treatment groups
 - Presence of the Total information
 - Name and Number of populations
- Objective: Using %include statement to point at standard macro program:
 - %let pgmname=t_aes;
 - %include './autoexec.sas';
 - %include './.../standard_macros/t_ae.sas';

RECURRENT ISSUE 2 - STEP II

SOLVING SUB-PROBLEMS/PROBLEM

- From the definition of the sub-problems, we conclude that an external file per study can be created.
- This external source needs to include the 3 information described earlier:
 - Name and number of population
 - Name and number of treatment groups
 - Is a Total column needed or not?

RECURRENT ISSUE 2 - STEP III

STANDARDIZING THE SOLUTION

Microsoft Excel - TFLs.xls

File Edit View Insert Format Tools Data Window Help

100%

Arial 10 B I U

C14

	A	B	C	D	E
1	TFL	Population Condition	Population Name	Treatment variable	Total?
2	T_AER	RAND=1	All Randomized Patients	TRTR	No
3	T_AES	SAF=1	Safety Set	TRTA	Yes
4					

Sheet1

Ready

NUM

RECURRENT ISSUE 2 - STEP III STANDARDIZING THE SOLUTION

- **Outputs:**
 - T_AER (Randomized Population):

Summary Table of Adverse Event
All Randomized Patients

	Placebo	Active Product
	(N= 200)	(N= 200)
Subjects with at least one Adverse Event	123	122
Number of events	356	355

- T_AES (Safety Population):

Summary Table of Adverse Event
Safety Set

	Placebo	Active Product	Total
	(N= 198)	(N= 199)	(N=397)
Subjects with at least one Adverse Event	121	121	242
Number of events	350	351	701

RECURRENT ISSUE 3: DELETING/STORING BLANK ROWS

- This example shows a standard solution for deleting/storing blank rows in any dataset and for any study without modifying the code.
 - Defining a blank row.
 - In general we can define a blank row as a row where the content of all database enterable variables are blank as shown in the first row in the below table, where B, E and G are DB enterable variables whereas A, C, D and F can be DB non enterable variables or other information.

A	B	C	D	E	F	G
XXX		XXX	XXX		XXX	
XXX	XXX	XXX	XXX	XXX	XXX	XXX

RECURRENT ISSUE 3 - STEP I

SEGMENTING ISSUE INTO SUB-PROBLEMS

- The following information is study/dataset dependant:
(A_001 and A_002 are two different studies)

- Sub problem 1: Number of datasets
- Sub problem 2: Datasets names
- Sub problem 3: Variable names
- Sub problem 4: Number and name of variables that participate in a blank row

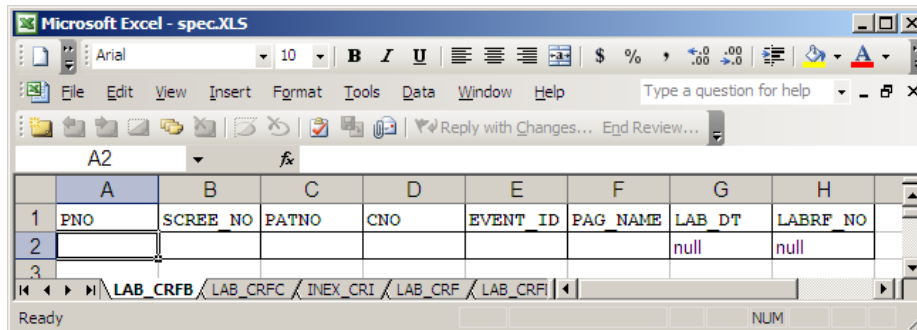
A_001	PATNO	AESEQ	AETYPE	AEV	AESDTE	AEEDT	AEONGO
	XXX		XXX			XXX	
A_002	SUBJNO	AESEQ	AETYPE	AEVT	AESDTE	AEEDT	AEONGO
	XXX		XXX	XXX		XXX	

A_001		A_002	
Name		Contents of 'libref'	
Ae		Adveve	
Cm		Antibody	
Comm		Antibrow	
Demo		Cilab	
Diary		Condis	
Disl		Conesa	
Ecq		Conmed	
Elig		Demog	
Eos1		Dialhist	
Eos2		Doseadm	
Esr		Ecq	
Ewb		Elig	
Fu		Eos	
Haha		Eq5d	
Ibdq		Excl	
Ivrs		Followup	
Lb		Formats	
Mdx		Formats	
Mh		Global	
Pe		Hemoglob	
Pga		Incl	
Pk		Ivrsraw	
Pmhx		Kdql	
Pt		Kidneydi	
Trt		Lab	
Vs		Medhist	
Xray		Phlebot	
		Phyexam	
		Range	
		Renalrpl	
		Renifail	
		Riskhist	
		Transf	
		Version	
		Vital	

RECURRENT ISSUE 3 - STEP II

SOLVING SUB-PROBLEMS/PROBLEM

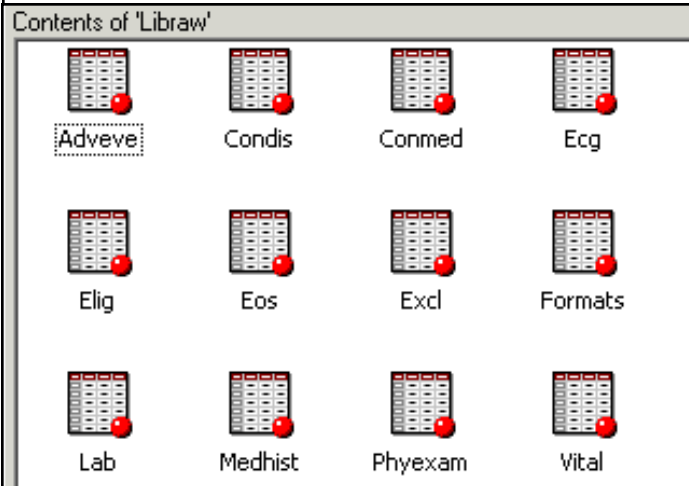
- Create an external excel file that can be read by the program as shows the below image.



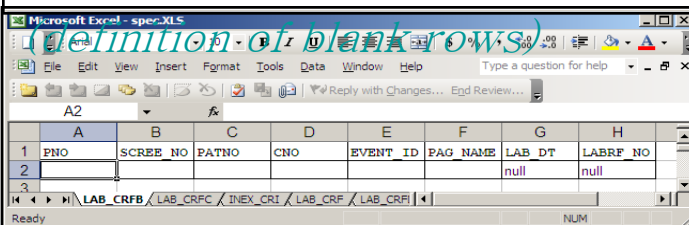
- The above file will be loaded in to SAS and it be will create a dataset for each sheet with all variables and just one row per dataset (which represents a blank row).
 - NOTE: the “null” value has been chosen for describing a blank variable but any other word can be used and of course the program should be amended accordingly.

RECURRENT ISSUE 3 - STEP III STANDARDIZING THE SOLUTION

- Names of Datasets
- Number of Datasets
(Stored in macro variables)



- Load external info into SAS.



- Check Number/Names of Datasets
- Check Variables defining a blank rows.
- Identify rows.

Delete blank row

Store the row in a Spreadsheet

	A	B	C	D	E	F
1	Clinical study	Subject ID	Visit name	Visit number	Sub event	Repeat sec
2	A_001	0160015	SUMMARY	800	1	
3	A_001	0210016	SUMMARY	800	2	

NUM

- The objective of this presentation was to demonstrate that a “global” approach to problem solving helps us attain both goals of **INCREASED QUALITY** and **TIME SAVINGS** (hence **COSTS SAVINGS**) due to the added robustness of re-using the same macro programs for several studies.
- Moreover, as shown in the first example, some sub-problems can be solved/worked on while other sub-problems are in development.
- Furthermore, some sub-problems solutions can be used on their own.
- At a time where the entire industry is adapting to standardization (i.e CDISC...), it is important to prepare the tools today that will facilitate the transition to these new standardized environments.

The authors/presenters can be contacted using the following details:

Tangi SANSEAU
Senior SAS Programmer I
Biostatistics Department

Tangi.Sanseau@iconplc.com

+353 (0)1.291.2483

and

Giuseppe DiMonaco
SAS Programmer
DIS Department

Giuseppe.DiMonaco@iconplc.com

+353 (0)1.291.2296

ICON Clinical Research
South County Business Park
Leopardstown
Dublin 18
IRELAND

Any questions?

ANY QUESTIONS?



*Get list of datasets;

```
proc sql;  
    create table dataset_list as  
    select distinct memname  
    from sashelp.vtable;  
    select count(*) into :dset_count from dataset_list;  
quit;
```

*counting the number datasets;

```
data count_dataset;  
    set dataset_list end=end_dset;  
    if end_dset then call symput('n_obs',_n_);  
run;
```

Any questions?

*creating macro variables for each dataset;

```
data _null_;  
  set count_dataset;  
  %do k=1 %to &n_obs;  
    if &k=_n_ then call symput(compress("MEMNAME" || &k),  
MEMNAME);  
  %end;  
run;
```

Any questions?

Standard macros directory structure:

