

Validate the validated: A python script for study leads to review clinical outputs

*Varaprasad Ilapogu, Ephicity Consulting Group
Inc.; Janet J. Li, Pfizer Inc.; Masaki Mihaila,
Pfizer Inc.; Ernesto Gutierrez, Pfizer Inc*

Introduction

- ▶ Validation of clinical SAS® programming datasets and statistical outputs is important in ensuring the accuracy of the submission packet in the pharmaceutical industry
- ▶ Table generation can be summarized as
 1. Sub-setting the population into specific population and treatment sub-groups.
 2. Calculating appropriate statistics for the tables.
 3. Presenting the information in RTF or PDF format usually using PROC REPORT

Validation

Study Table				
Demographic and Baseline Characteristics				
Visual Check	Baseline Characteristic	Cohort 1 (N=XX)	Cohort 2 (N=XX)	Total (N=XX)
	Category			
	n	XX	XX	XX
	Mean (SD)	XX.X (XX.XX)	XX.XX (XX.XX)	XX.X (XX.XX)
	Median	XX.XX	XX.X	XX.X
Programmatic check	Min, max	XX.X, XX.X	XX.X, XX.X	XX.X, XX.X

- ▶ **Independent programming**
 - ▶ Production programmer outputs the final dataset used in PROC REPORT to a permanent library on the server
 - ▶ Validation programmer uses PROC COMPARE to detect if any differences exist - mostly aimed at checking body of output (e.g. frequency or proportion of an event)
- ▶ **Visual checking**
 - ▶ Look for spelling errors, alignment issues, population denominator inconsistencies
 - ▶ Can be subjective

Need for programmatic validation

- ▶ Current practice may give false impression that production table has passed validation
 - ▶ Only values (frequencies and percentages) in the final SAS® dataset get validated programmatically
 - ▶ Population denominators (defined in the PROC REPORT DEFINE statement) are indirectly validated
 - ▶ If production and validation denominators differ by a small value, the percentage calculations in the two datasets may still match if the denominator is a large enough number and if the percentages are rounded

```
proc report data = final split = '~' missing nowindows;  
  by pageno;  
  
  column (pageno order text value1 value2 value3);  
  
  define pageno    / order order=internal noprint;  
  define order     / order order=internal noprint;  
  define text      / display "Generic Name" ;  
  define value1    / display "Cohort 1 ~ (N=&C1.)";  
  define value2    / display "Cohort 2 ~ (N=&C2.)";  
  define value3    / display "Total ~ (N=&C3.)";  
  
run;
```

Need for programmatic validation

- ▶ Individual programmers do not routinely cross check one output against a different output
 - ▶ Table denominators sometimes end up differing across the study.
 - ▶ Different outputs have some underlying commonalities such as analysis population (Safety, Intent to Treat etc.), a specific overall statistic (Number of subjects who had at least 1 Adverse Event), Number of subjects in each treatment group etc
 - ▶ Some of these numbers are not presented in the body of the table and instead presented in the headers and checked visually if at all.

Python Script

- ▶ We have developed a python script that addresses gaps that may exist in checking information across statistical outputs
- ▶ The script extracts commonalities from each statistical output and presents the relevant information in a single, easily accessible document

Python

- ▶ Python is a high-level, general-purpose programming language. For the script to run:
 1. Install *Python 3* on your PC
 2. *Anaconda 3* should be installed - most of the utilized libraries are included
 3. Obtain *StyleFrame* package - call "pip install StyleFrame"
 4. Obtain *pyth3* library - call "pip install pyth3"
- ▶ Input = location where the tables are stored
 - ▶ Python Graphical User Interface (GUI) app will open and ask the user to point to the location by navigating the Windows File Explorer.
- ▶ Each RTF table in the source location is converted to plaintext string using XHTMLWriter from *pyth3's* "pyth.plugins.xhtml.writer"
- ▶ Relevant information (['Protocol name', 'Table number', 'Total pages', 'Table title', 'Header', 'Date created']) is stripped and outputted to an external Excel file using *Pandas DataFrame* and *StyleFrame* utilities.

Table 1

Protocol 001

(Page 1 of 1)

Table 1
Patient Populations

Patient Population	Cohort 1 (N=49)	Cohort 2 (N=35)	Total (N=84)
ITT population	49 (100.0%)	35 (100.0%)	84 (100.0%)
Safety population	48 (98.0%)	35 (100.0%)	83 (98.8%)
Tumor-evaluable population --- IRF	48 (98.0%)	35 (100.0%)	83 (98.8%)
Tumor-evaluable population --- investigator	48 (98.0%)	35 (100.0%)	83 (98.8%)
PK population	48 (98.0%)	35 (100.0%)	83 (98.8%)

Table 2

Protocol 001

(Page 1 of 3)

Table 2
Demographic and Baseline Characteristics
(ITT Population)

Baseline Characteristic	Cohort 1 (N=49)	Cohort 2 (N=34)	Total (N=83)
Age			
n	49	34	83
Mean (SD)	50.1 (11.48)	53.4 (11.05)	51.5 (11.35)
Median	50.0	52.0	50.0
Min, max	31.0, 74.0	33.0, 75.0	31.0, 75.0

Table 3

Protocol 001

(Page 1 of 1)

Table 3
Dose Modifications Due to Adverse Events
(Safety Population)

	Cohort 1 (N=49)	Cohort 2 (N=35)	Total (N=84)
Patients with no dose reduction due to adverse event	22 (45.8%)	12 (34.3%)	34 (41.0%)
Patients with at least 1 dose reduction due to adverse event	26 (54.2%)	23 (65.7%)	49 (59.0%)

Table 4

Protocol 001

(Page 1 of 1)

Table 4
Study Drug-Related Treatment-Emergent Adverse Events by System Organ Class, and Preferred Term
(Safety Population)

System Organ Class Preferred Term	Cohort 1 (N=50)	Cohort 2 (N=35)	Total (N=85)
Number of patients with at least 1 study drug-related TEAE	46 (95.8%)	33 (94.3%)	79 (95.2%)

Table 5

Protocol 001

(Page 1 of 1)

Table 5
Treatment-Emergent Adverse Events Leading to Study Drug Dose Reduction by System Organ Class and Preferred Term
(Safety Population)

System Organ Class Preferred Term	Cohort 1 (N=48)	Cohort 2 (N=35)	Total (N=83)
Number of patients with at least 1 study drug-related TEAE	45 (93.8%)	33 (94.3%)	78 (94.0%)

Excel Output Generated by Python Script

Project name	Table number	Total pages	Table title	Header
Protocol 001	Table 1	1	Table 1 Patient Populations	Cohort 1 (N=49) Cohort 2 (N=35) Total (N=84)
Protocol 001	Table 2	1	Table 2 Demographic and Baseline Characteristics (ITT Population)	Cohort 1 (N=49) Cohort 2 (N=34) Total (N=83)
Protocol 001	Table 3	1	Table 3 Dose Modifications Due to Adverse Events (Safety Population)	Cohort 1 (N=49) Cohort 2 (N=35) Total (N=84)
Protocol 001	Table 4	1	Table 4 Study Drug-Related Treatment-Emergent Adverse Events by System Organ Class, and Preferred Term (Safety Population)	Cohort 1 (N=50) Cohort 2 (N=35) Total (N=85)
Protocol 001	Table 5	2	Table 5 Treatment-Emergent Adverse Events Leading to Study Drug Dose Reduction by System Organ Class and Preferred Term (Safety Population)	Cohort 1 (N=48) Cohort 2 (N=35) Total (N=83)

Conclusion

- ▶ In this paper, we focus on the current practice of table validation.
 - ▶ While the body of a table is validated programmatically, there are some parts of the table that are validated indirectly or through visual checking.
 - ▶ There is no formal programmatic procedure in place to check for accuracy of information that is similar across tables.
- ▶ The python script we have created provides a more efficient and accurate programmatic solution for the gaps in the table validation process
- ▶ We hope that this additional step can help enhance the quality and increase the efficiency of the study package review process.

Contact Information

- ▶ Name: Prasad Ilapogu
- ▶ Company: Pfizer Inc.
- ▶ City/State: San Francisco, CA
- ▶ Email: prasad.ilapogu@pfizer.com