Study on Good Programming Practices in Health and Life Sciences Results

File Generated: 30AUG2013

Table of Contents

Study on Good Programming Practices in Health and Life Sciences
Results Analysis

Results from the first twenty respondents to the Study on Good Programming Practices in Health and Life Sciences are now available.

The GPP Board would like to thank all those who have responded so far and invite those who have not responded to take part by following this link to the survey (https://www.surveymonkey.com/s/GPPSURVEY).

The respondents represent a range of organizations, including pharmaceutical companies, biotechnology companies, CROs and consultancies. 68% of respondents had a GPP document to follow such as a guideline or SOP and of these, 80% were mandatory.

Here are some highlights from the results of good programming practices which respondents indicated that they used and felt were either important or very important:

- 82% of respondents said consistently ending steps with run and quit is important or very important
- 78% consistently explicitly referencing input dataset in each step or procedure
- 77% use of descriptive comments in each section of the program
- Others include breaking lines in a neat and logical way and placing only one statement per line

Number of participants	22 (100)
Organization type	
Academic Research Institution	1 (5)
Biotechnology Company	1 (5)
Consultancy	4 (18)
Other (please specify)	1 (5)
Pharmaceutical Company	9 (41)
Pharmaceutical/Biotechnology CRO	6 (27)
Region	
Europe	12 (55)
Global	3 (14)
U.S.	7 (32)
How many programmers are in the department/group/unit that you are responding for?	
0 - 10	8 (36)
11 - 30	8 (36)
31 - 50	2 (9)
51 - 100	3 (14)
>100	1 (5)

NOTE: Numbers in parentheses are percentages

SOURCE: DRUG/STUDY/SMART1/GPP-SUM.SAS DATE: 30AUG2013

Have written documentation detailing the use of good programming practices	15 (100)
Guideline	13 (87)
SOP	3 (20)
Work Instruction	6 (40)
Other	2 (13)
	7 (47)
If there is documentation, is it mandatory to follow?	
No	3 (20)
Yes	12 (80)

NOTE: Numbers in parentheses are percentages

SOURCE: DRUG/STUDY/SMART1/GPP-SUM.SAS DATE: 30AUG2013

Number of participants	22 (100)
Program Layout - Use of standard header	
Not used (0)	1 (5)
Used but not important (1)	1 (5)
Used but only somewhat important (2)	1 (5)
Used and important (3)	4 (18)
Used and very important (4)	15 (68)
Program Layout - Descriptive comments in each section of the program	
Not used (0)	0
Used but not important (1)	1 (5)
Used but only somewhat important (2)	4 (18)
Used and important (3)	10 (45)
Used and very important (4)	7 (32)
Program Layout - Consistent use of appropriate spacing between statements and sections	
Not used (0)	3 (14)
Used but not important (1)	3 (14)
Used but only somewhat important (2)	5 (23)
Used and important (3)	6 (27)
Used and very important (4)	5 (23)

NOTE: Numbers in parentheses are percentages.

SOURCE: DRUG/STUDY/SMART1/GPP-SUM2.SAS DATE: 30AUG2013

Program Layout - Consistent use of appropriate indentation on each line	
Not used (0)	2 (9)
Used but not important (1)	5 (23)
Used but only somewhat important (2)	5 (23)
Used and important (3)	3 (14)
Used and very important (4)	7 (32)
Program Flow - Consistently placing only one statement per line	
Not used (0)	1 (5)
Used but not important (1)	4 (18)
Used but only somewhat important (2)	4 (18)
Used and important (3)	9 (41)
Used and very important (4)	4 (18)
Program Flow - Breaking lines in a neat and logical way	
Not used (0)	3 (14)
Used but not important (1)	2 (9)
Used but only somewhat important (2)	3 (14)
Used and important (3)	12 (55)
Used and very important (4)	2 (9)

NOTE: Numbers in parentheses are percentages.

SOURCE: DRUG/STUDY/SMART1/GPP-SUM2.SAS DATE: 30AUG2013

Program Flow - Consistently ending steps with run and quit where appropriate	
Not used (0)	0
Used but not important (1)	2 (9)
Used but only somewhat important (2)	2 (9)
Used and important (3)	7 (32)
Used and very important (4)	11 (50)
Program Flow - Consistently explicitly referencing input dataset in each step or procedure	
Not used (0)	2 (9)
Used but not important (1)	1 (5)
Used but only somewhat important (2)	2 (9)
Used and important (3)	5 (23)
Used and very important (4)	12 (55)
Program Flow - Program is logically organized	
Not used (0)	1 (5)
Used but not important (1)	1 (5)
Used but only somewhat important (2)	2 (9)
Used and important (3)	7 (32)
Used and very important (4)	11 (50)

NOTE: Numbers in parentheses are percentages.

Standard Naming Conventions - Standard naming convention for temporary datasets	
Not used (0)	10 (45)
Used but not important (1)	3 (14)
Used but only somewhat important (2)	3 (14)
Used and important (3)	4 (18)
Used and very important (4)	2 (9)
Standard Naming Conventions - Standard naming convention for variables in temporary datasets	
Not used (0)	12 (55)
Used but not important (1)	4 (18)
Used but only somewhat important (2)	1 (5)
Used and important (3)	3 (14)
Used and very important (4)	2 (9)
Standard Naming Conventions - Standard naming convention for formats	
Not used (0)	9 (41)
Used but not important (1)	4 (18)
Used but only somewhat important (2)	2 (9)
Used and important (3)	5 (23)
Used and very important (4)	2 (9)

NOTE: Numbers in parentheses are percentages.

SOURCE: DRUG/STUDY/SMART1/GPP-SUM2.SAS DATE: 30AUG2013

Standard Naming Conventions - Standard naming convention for macros	
Not used (0)	7 (32)
Used but not important (1)	2 (9)
Used but only somewhat important (2)	3 (14)
Used and important (3)	5 (23)
Used and very important (4)	5 (23)
Coding Practices - Explicitly code character to numeric and numeric to character conversions	
Not used (0)	3 (14)
Used but not important (1)	2 (9)
Used but only somewhat important (2)	2 (9)
Used and important (3)	8 (36)
Used and very important (4)	6 (27)
Coding Practices - Explicitly code to handle missing data	
Not used (0)	5 (23)
Used but not important (1)	0
Used but only somewhat important (2)	4 (18)
Used and important (3)	9 (41)
Used and very important (4)	3 (14)

NOTE: Numbers in parentheses are percentages.

Coding Practices - No overwriting temporary datasets within a program	
Not used (0)	5 (23)
Used but not important (1)	2 (9)
Used but only somewhat important (2)	5 (23)
Used and important (3)	4 (18)
Used and very important (4)	5 (23)
Coding Practices - Check log for errors and warnings	
Not used (0)	0
Used but not important (1)	0
Used but only somewhat important (2)	1 (5)
Used and important (3)	2 (9)
Used and very important (4)	18 (82)
Coding Practices - Clean up work area after each program when running interactively	
Not used (0)	7 (32)
Used but not important (1)	2 (9)
Used but only somewhat important (2)	2 (9)
Used and important (3)	4 (18)
Used and very important (4)	6 (27)

NOTE: Numbers in parentheses are percentages.

SOURCE: DRUG/STUDY/SMART1/GPP-SUM2.SAS DATE: 30AUG2013

Coding Practices - Clean up work area before each program when running interactively	
Not used (0)	5 (23)
Used but not important (1)	3 (14)
Used but only somewhat important (2)	3 (14)
Used and important (3)	3 (14)
Used and very important (4)	7 (32)

NOTE: Numbers in parentheses are percentages.

SOURCE: DRUG/STUDY/SMART1/GPP-SUM2.SAS DATE: 30AUG2013

Study on Good Programming Practices in Health and Life Sciences
Summary Part 3 - Programming Practices Ranked by Mean Importance

Programming practice	Mean Importance (0=Not Used - 4=Very Important)
Program Layout - Use of standard header	3.4
Program Flow - Consistently ending steps with run and quit where appropriate	3.2
Program Flow - Program is logically organized	3.2
Program Flow - Consistently explicitly referencing input dataset in each step or procedure	3.1
Program Layout - Descriptive comments in each section of the program	3.0
Program Flow - Consistently placing only one statement per line	2.5
Program Layout - Consistent use of appropriate indentation on each line	2.4
Program Flow - Breaking lines in a neat and logical way	2.4
Program Layout - Consistent use of appropriate spacing between statements and sections	2.3
Standard Naming Conventions - Standard naming convention for macros	2.0
Standard Naming Conventions - Standard naming convention for formats	1.4

Programming practice	Mean Importance (0=Not Used - 4=Very Important)
Standard Naming Conventions - Standard naming convention for temporary datasets	1.3
Standard Naming Conventions - Standard naming convention for variables in temporary datasets	1.0
Coding Practices - Check log for errors and warnings	3.81
Coding Practices - Explicitly code character to numeric and numeric to character conversions	2.57
Coding Practices - Explicitly code to handle missing data	2.24
Coding Practices - Clean up work area before each program when running interactively	2.19
Coding Practices - No overwriting temporary datasets within a program	2.10
Coding Practices - Clean up work area after each program when running interactively	2.00

1) A standard templates library should be created for each project that will be applied to all ODS outputs (RTF, PDF, etc.), ensuring consistency across all outputs. 2) A centralized titles spreadsheet should be created, allowing easy modification of titles. This titles spreadsheet can then be converted to a SAS´fi data set and accessed by output programs

1) Macro parameters are keywords, not positional - Janssen 2) SAS code in lower case - Janssen 3) Options and formats created at beginning of program 4) No overwriting variables with new content

1) Program not just a call to a macro but actually having code in the program. 2) Standard macro variable initialization and setup file call at top of program.

1) Standard main derivation (e.g. Date imputation for AEs) 2) folders/files organisation 3) folders/files naming conventions 4) always save log and lst

1) Use of standard/shared code to set up data access 2) Set all data access to read-only except when necessary in a particular program

1) Use upper case for SAS keywords 2) Read in each external data only once into a program 3) Read in all external data at the top of the program 4) Use a standard program where all libnames and formats are defined 5) If creating a permanent dataset then always do a proc contents after the dataset is created

1) for Sql statement, alignments

1) stand alone programs, evrey program should be submitable in batch mode

1) terminate a macro with %mend followed by the name of the macro, not only %mend. 2) include the option NOWD in each proc report for those who like to run SAS interactively 3) a macro should never change OPTIONS 4) a macro should never modify a global macro variable or a dataset, unless it was specified in a parameter fasion 5) Use SELECT statement to replace 3 or more IF statements on a same variable/condition 6) In the program header and where appropriate, an author and modifier of a program should be identified by a full name, not just initials. 7) Clear the content of the WORK at the beginning rather than the end of a program - leaving datasets present at the end makes it easier to debug, if necessary 8) Isoteric code should be avoided, but when necessary, it should always be accompagnied with descriptive comments 9) Avoid writing the words ERROR and WARNING. If necessary, split them by use of "ER" "ROR: ..." (e.g. PUT), or

1)Code review 2)Bug resolution provide dedicated testing 3)tracing relationship between source code, test code and requirement

1. Save logs 2. Save lst files 3. Save comparison output

NA

Standard macros go through rigid SDLC and have propper user documentation. Standard macros use standard return codes and must be designed to be as robust as possible. We also use standard directory structure and have programming to set these up.

We have naming conventions for permanent data sets and variables.