PharmaSUG 2024 - Paper SD-217

Semi-Automated and Modularized Approach to Generate Tables for Clinical Study – Categorical Data Report

William Wei, Shunbing Zhao, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

A table, listing and figure (TLF) package, incorporated in a clinical study report (CSR), serves to encapsulate the data from a clinical trial. Tables constitute the majority of the TLF package. The CSR tables, which primarily summarize safety and efficacy data, can be categorized into two major modules 1) categorical (gender, sex, race, etc.) and 2) continuous (age, time to event, etc.). This paper explores and discusses a design of the first major module - categorical, which could generate multiple output formats to accommodate the report requirements. The design of this module is independent, flexible, portable, informative, and parameter-driven, and can be exported or linked to different reports. This module can be used independently to produce a categorical table report in different formats. It also can produce different output formats and the reports can be combined for a more complex table. For instance, a table may consist of categorical analysis blocks, continuous blocks, and other blocks, such as Confidence Interval, P-value, etc. This approach provides users with substantial flexibility to construct any table they have in mind and present it as desired. This approach significantly reduces the time and effort required for programming CSR tables.

INTRODUCTION

Modular programming is an approach that involves breaking down a computer program into separate sub-programs known as modules. These modules are independent programs that perform specific functions and data manipulation. One of the key advantages of modular programming is that these modules can be reused in different applications, which not only enables developers to work concurrently but also reduces development time. Moreover, programs developed using modular programming are easier to read, debug, and standardize. A well-designed module should be capable of functioning as an independent unit and should be adaptable to various situations. Additionally, modules should also be portable, allowing them to be inserted into different programs when required.

By looking at the tables in the CSR reports, there are many common or similar report blocks, which makes the application of modular programming a feasible approach. These blocks are as follows:

- Categorical variable report block: The report block represents a summary of categorical variable.
- Continuous variable report block: The report block provides a summary of continuous variable.
- Shift report block: The report block represents events level shift along the trial.
- Binomial confidence interval block: The report block focuses on calculating and displaying binomial confidence intervals.
- Other blocks: This category includes additional report blocks such as P-values.

For example, in “Summary of Demographics” table (Figure 1), it mostly consists of continuous variable report blocks and categorical variable report blocks. This pattern can be seen in almost all the summary tables and other AE/CM/MH tables.

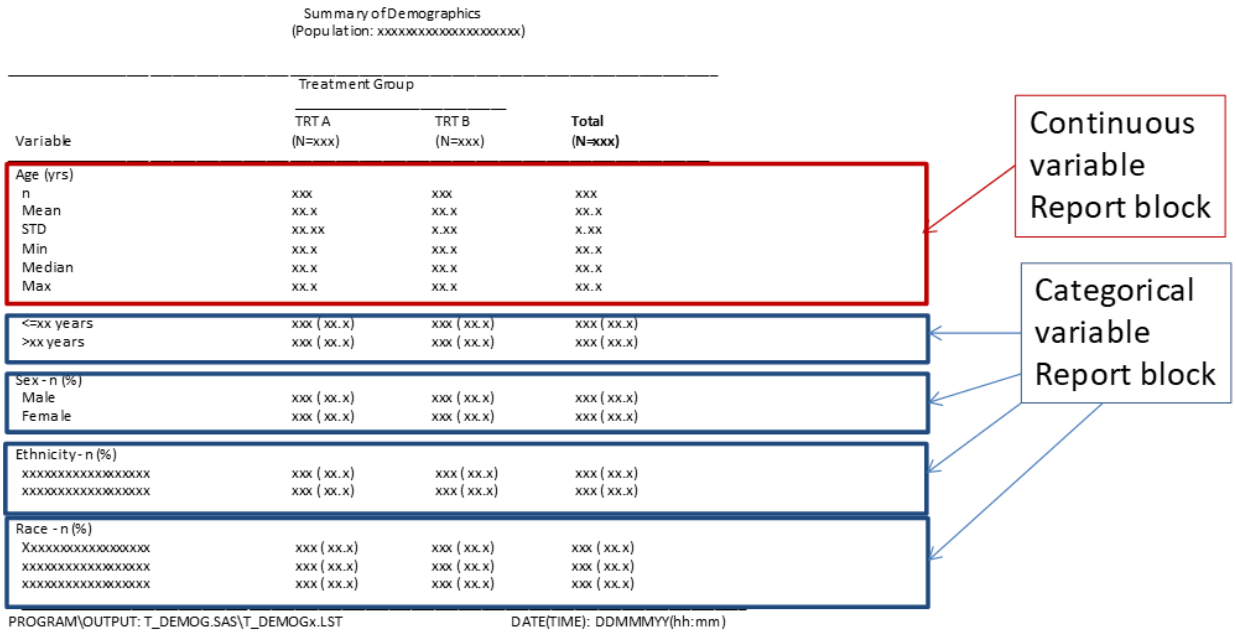

Figure 1: Summary of demographics table mockup.

In the case of categorical variable report, it could be presented in different formats within different reports. In the “Subject Disposition” table below, there are variations in how the categorical blocks are displayed, including both an "itemized" or "summarized" report block, and a regular categorical variable report block. The total column is optional to show, and this applies to all the tables. (Figure 2).

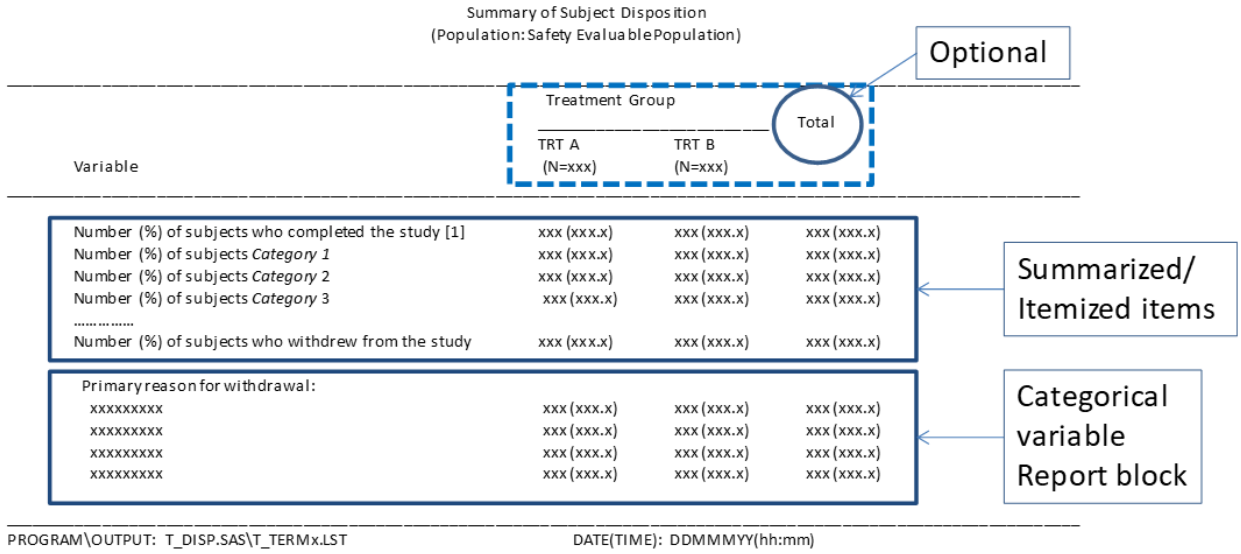

Figure 2: Screenshot of subject disposition table mockup.

Given that clinical study reports consist of different types of reports, each with their own variations in appearance, we have developed a set of report modules that can be combined to create a comprehensive safety or efficacy evaluation report. Our first designed report module is the categorical variable report module.

CATEGORICAL VARIABLE REPORT MODULE DESIGN

The categorical variable summary is a recurring element in the reports and share a similar structure and algorithm. Based on this information, different formats of the categorical report analysis blocks can be generated by a single module. The goal is to use this module to produce a table-ready data set that matches the required report format. Then a simple PROC REPORT statement in SAS can generate the final report in RTF or PDF format.

When we generate a common categorical variable report block, all necessary information for various types of categorical reports is calculated and summarized in this module (Figure 3 -1). The information includes calculated statistical values (counting numbers and percentages) in all categories and sub-categories, and the module allows for the selection of which information to be displayed. The following report blocks can be produced by this module:

1) Summarized/itemized report with options to display or not-display a percentage (Figure 3, Panel 2).
2) Categorical report of sub-category summary with options to display or not-display a percentage, as well as missing subject summary report (Figure 3, Panel 3).
3) Output format 2 without missing subject summary report (Figure 3, Panel 4).
4) Output format 2 with all categories report line (Figure 3, Panel 5). This format is used in the basket protocol design studies.
5) Categorical report with a horizontal presentation (Figure 3, Panel 6).

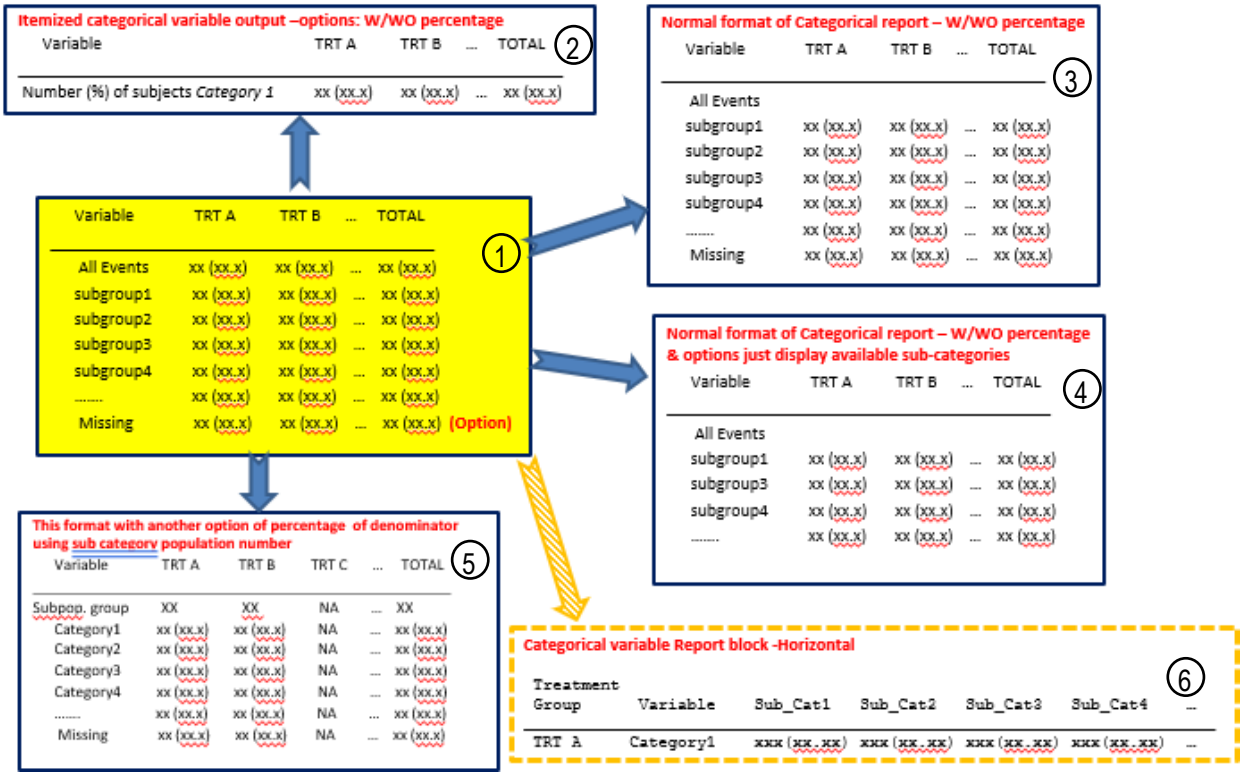

Figure 3: Categorical variable analysis module design, and variety of categorical report blocks.

The described categorical variable report module is designed to be an independent and expandable module that can be used to generate categorical tables in various formats. It can also serve as a building block to assemble more complicated clinical reports. To achieve these functionalities, the module is designed with the following features:

1. Independence: this module can be used independently and is expandible.
2. Flexibility.
3. Combination compatibility.

INDEPENDENCE

The categorical report module is designed as a robust and independent unit. It can be used to generate a single categorical variable report and can add additional reports without affecting other modules or overall project. During the program development life cycle (PDLC), additional functions can be added to this module to satisfy more requirements per new complicated reports. The module can be easily expanded while maintaining its independence.

A sample macro call codes and output are shown in Figure 4. With this macro design, it can produce all the report formats displayed in figure 3. The output column will be assigned from column C_1 to C_n+1. Column C_1 to C_n is the number of treatments, sorted by their numeric treatment value. And column C_n+1 is the total. For example, if there are five treatment groups 1, 2, 7, 8, 10, the correspondent column names are C_1, C_2, C_3, C_4, C_5. C_6 is the total column. To identify the column for treatment, the column labels with the corresponding treatment are assigned to the columns (Figure 4, showed the column names and corresponding column labels). Below shown is part of the macro parameters and a sample of macro call and outputs:

- DSIN: Input data set that has the variable to summarize.
- DSPOP: Population data set.
- TRTN: Numeric version of treatment variable.
- CATN: Numeric version of categorical variable.
- CNTFMT: Event/Subject count number format.
- PCTFMT: Percentage format
- CITMLST: The list and format of the category required to precent in the table.
- CATSUM: Show/No Show of category summary line (Figure 4-3).
- PCTYN: Show/No Show of percentage.
- GLABEL: Category label specified.
- INDENT: Indent of category labels.
- ORDERBY: List order of categories.

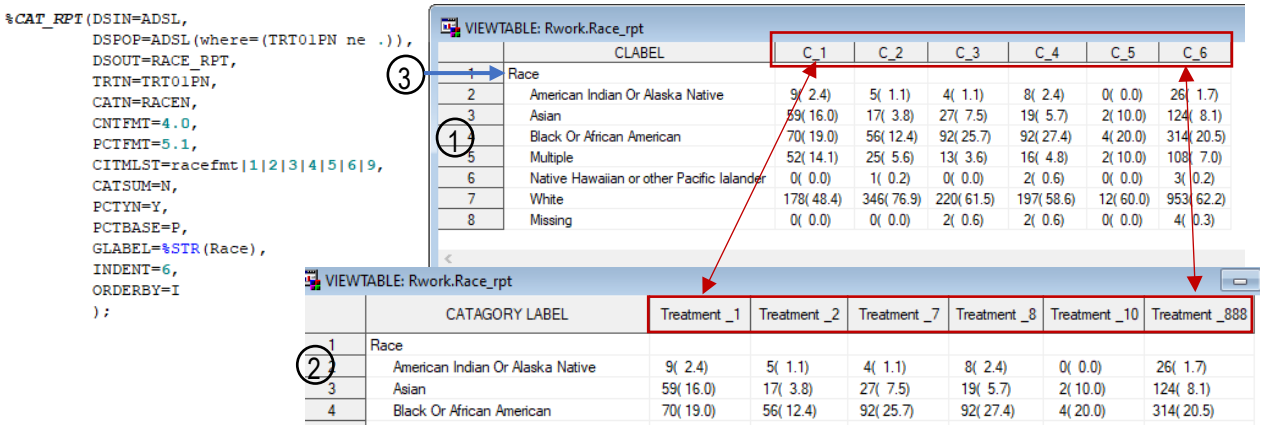

Figure 4: Example of Categorical macro input/output. Left side is a sample of macro input. Right side is the output of product data set with 1) column name displayed, 2) column label displayed. 3) Category label/summary row.

FLEXIBILITY

As a building block, this macro is flexible enough to display different format requirements. Every part of the table body is flexible to have multiple formats.

Category summary line flexibilities:

Most of the categorical variable would not show the summary line. But in certain cases, such as basket study design, there is a need to show the summary line (the total per category per treatment group showed in Figure 5). In the summary line, there is also an option to show percentage or not. In Figure 5-A: CATSUM controls option to show summary line. SUMPCT controls the option to show percentage in summary line. In Figure 5-B: if CITMLST=-1, then only show summary line. SUMINDENT is used to create label indent for summary line.

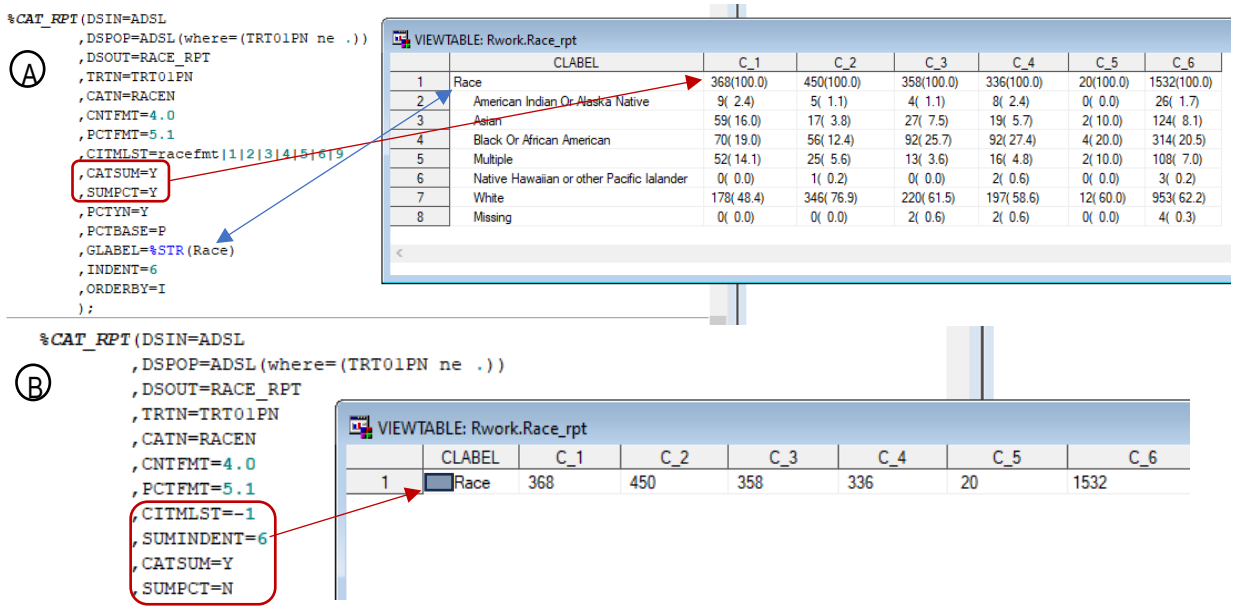

Figure 5: Macro control on category summary line. A: Control to show summary line and the percentage. B: To show only summary line without percentage.

Category report body flexibilities

All the table body parts have flexibility to show the appearances as required.

Figure 6, Panel 1: Parameter PCTYN controls if the table will show percentage or not.

Figure 6, Panel 2: The list of categories in parameter CITMLST controls which category to be shown in the table including missing category. The numbers in the list of categories are numeric represents of RACEFMT format.

Figure 6, Panel 3: User defined format RACEFMT controls the category labels. So, the category description will not be restricted to the value in data. User can define the category names according to the table mockup.

Figure 6, Panel 4: Parameter INDENT give user flexibilities to indent category labels to make summary label standing out and make tables tidy.

Figure6, Panel 5: ORDERBY parameter controls the order of categories listed in the table. I: By the order of input list in parameter CITMLST (Figure 6-2). ND: By total number descending. NA: By total number ascending. TD: By category name descending. TA: By category

name ascending.

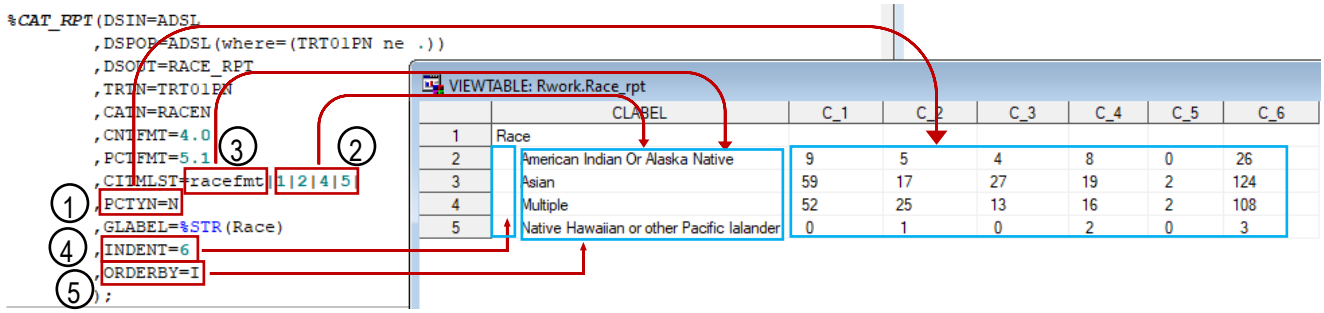

Figure 6. The flexibilities of table body parts.

COMBINATION COMPATIBILITY

Being a building block design, the report blocks produced using the categorical variable module can be easily combined with the same categorical blocks or other blocks such as numerical variable blocks. As shown in Figure 1 and Figure 2, a variety of categorical variable blocks and numerical variable blocks can be combined vertically to build intact clinical reports. While blocks need to be combined horizontally, they will be inserted into a predefined position next to the block to be combined.

A new macro tool had been developed to fulfill the task of combining multiple blocks into complicated clinical study reports. Due to the limit of paper presentation, this new macro tool will be introduced in future paper. This macro tool functions like display shelf, which can import the report blocks as required. Figure 7 described the combination of the report blocks. The macro that combines the blocks actions as a shelf which can import different report blocks into required shelf spaces. The report blocks can fill the shelf spaces.

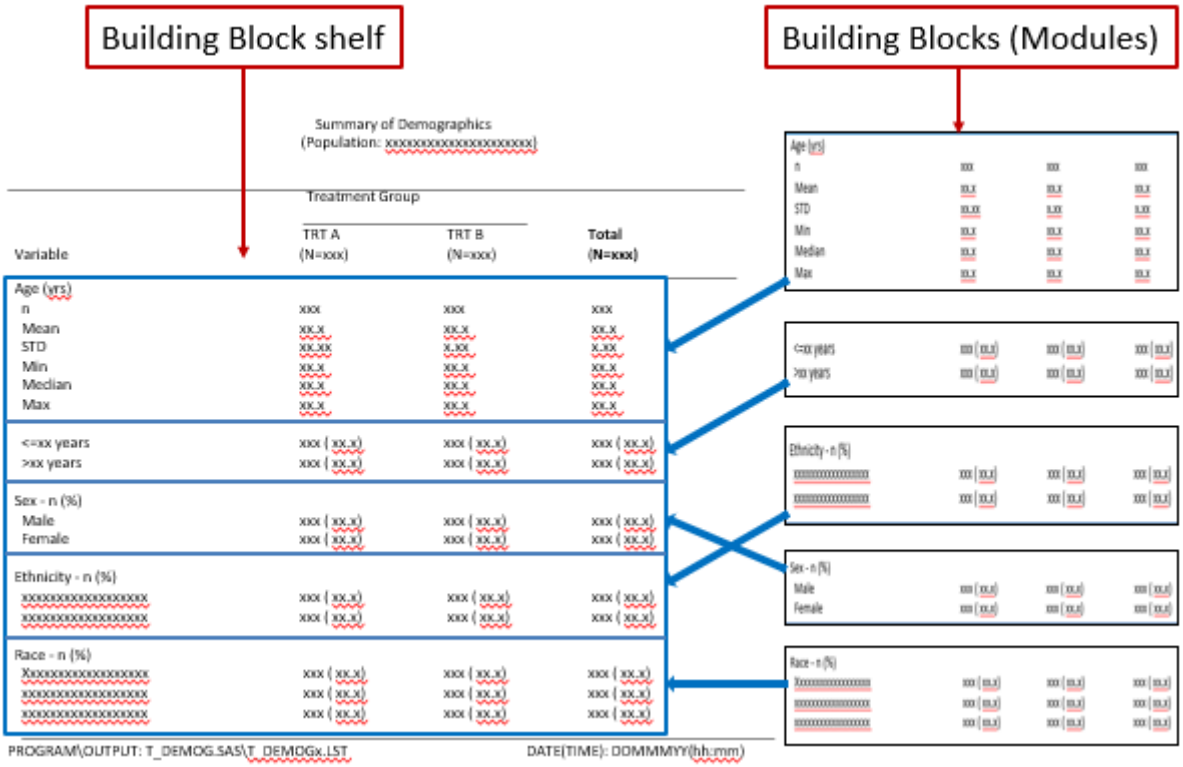

Figure 7. Example of clinical study reports generation by combining the blocks.

CONCLUSION

The modularized programming approach enables development of modules which is easy to debug, easy to maintain, allowing reusability and compatibility. This, with CDISC standardization of SDTM and ADaM data sets, allows for the development and maintenance of a few core modules, resulting in efficient programming and resource savings.

APPENDIX

A full list of parameters from %CAT_RPT macro:

```
%MACRO CAT_RPT(  DSIN=      /*MANDATORY:INPUT DATASET FOR ANALYSIS*/
                ,DSPOP=ADSL /*OPTIONAL: POPULATION DATASET, IF MISSING THEN
                              &DSIN WILL BE USED AS POPULATION DATASET;*/
                ,DSOUT=     /*OPTIONAL:OUTPUT DATASET FOR THE BLOCK*/
                ,TRTN=      /*MANDATORY:TREATMENT GROUP WHICH WILL BE SUMMRIZED ON, NUMERIC VERSION*/
                ,TRT =      /*OPTIONAL: CHARACTER VERSION OF TRTN    */
                ,CATN=      /*OPTIONAL: CATAGORICAL VARIABLE WHICH WILL BE SUMMRIZED, NO NEED FOR ONLY SUMMARY LINE*/
                ,CAT =      /*OPTIONAL: CATAGORICAL VARIABLE WHICH WILL BE SUMMRIZED, IF FORMAT NOT PROVIDED
                                        IN CITMLST, THIS WILL BE USED FOR DISPLAYING THE CATAGORY NAME IN OUTPUT*/
                ,CNTFMT=4.0  /*OPTIONAL: COUNT NUMBER FORMAT  */
                ,PCTFMT=5.1 /*OPTIONAL:FORMAT OF PERCENT%*/
                ,CITMLST=   /*MANDATORY: THE LIST OF OUTPUT CATAGORIES(NUMBER VERSION), EX:CATFMT|2|3|4|99,
                              IF MISSING AFTER FORMAT, OUTPUT ONLY EXISTING CATAGORIES
                              IF AFTER FORMAT IS -1 THEN JUST OUTPUT SUMMARY LINE
                              IF INPUT IS '-1', THEN JUST OUTPUT SUMMARY LINE
                              IF OPTION IS MISSING, THEN USE CAT FOR CATAGORY LABEL */
                ,Missing=N  /*Optional: Including missing in report. Set Missing = 99 and "Missing" for missing CAT var  */
                ,CATSUM=N   /*OPTIONAL: HAS SUMMARY COUNTING (Y) FOR ALL CATAGORIES, OR NOT */
                ,SUMPCT=N   /*OPTIONAL: ADDING PERCENT TO SUMMARY LINE (Y/N), IF PCTBASE=N THIS IS IGNORED  */
                ,PCTYN =Y   /*OPTIONAL: DEFAULT: Y=YES. PRINT THE PERCENTAGE OR NOT 'N'*/
                ,PCTBASE=P  /*MANDATORY: P/N: P-TOTAL POPULATION, N-TOTAL N IN CATAGORY, PLUS ADDITIONAL LINE FOR TOTAL NUMBER
                                                    IN THIS CATAGORY  */
                ,NPCTGAP=   /*OPTIONAL: SEPPARATE REPORT COLUMN N AND PERCENT BY THE NUMBER OF SPACES, ONLY INTEGER IS ALLOWED  */
                ,GLABEL=    /*OPTIONAL: GROUP LABEL*/
                ,SUMINDENT= /*OPTIONAL: INDENT FOR SUMMARY LINE - MOSTLY USED FOR AE SUMMARY REPORT = INTEGER>=1, MISSING=NO INDENT  */
                ,INDENT=    /*OPTIONAL: INDENT CHARACTER SPACE IN FRONT OF CATAGORY NAME, ONLY INTERGER ACCEPTED   */
                ,ORDERBY=I  /*OPTIONAL: DEFAULT=I: INPUT ORDER. ND/NA: BY TOTAL NUMBER DECENDING/ACCENDING.
                                        TD/TA: TERM DECENDING/ACCENDING;*/
                ,DEBUG = N
                );
```

CONTACT INFORMATION

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

William Wei
Merck & Co., Inc.
William.wei@merck.com

Shunbing Zhao
Merck & Co., Inc.
Shunbing.zhao@merck.com