ABSTRACT

Rules are one of the cornerstones for analysing clinical trials, either defined through regulatory guidance, analysis plans or specifications. As standards for both the capture, analysis and reporting of data prevail in a business process; the flexibility is further constrained or more precisely defined. We can use this to our advantage by creating an evolving book of rules that can be implemented to verify that the capture data follows standards and study expectations as well as deriving the more standard variables for analysis and reports. We consider how a rule book can influence standard programming and the two SAS macros that are the cornerstone for rule based programming.

INTRODUCTION

Two popular myths are that all analysis is unique or all analysis is standard analysis and consistent. Both are valid but we are fast approaching consistency as CDISC and other industry standards have become mainstream and de-facto standards.

Most organisations have a set of standards of varying detail, official and enforced through templates or unofficial propagated through copying the last or a similar analysis. As data standards mature, evolve and are refined, additional elements are incorporated into the standards as part of the natural process. More organisations are progressing to develop a book of standard derivations that drive consistency across all analysis.

It is not uncommon for data set standards, and associated standard derivations, to be implemented through large all-inclusive and ever complex macros. Another approach is to create many smaller functional and less flexible macros to execute a single defined derivation efficiently. Or simply eliminate macros entirely and focus on template code that efficiently derives one or a set of related rules.

Although each approach has its own merits and drawbacks, the functional approach, many smaller macros and more efficient in meta-data driven environments as one macro would implement one derivation rule. Because the simplicity of the functional rule-based macros, creating test cases for validation is a straightforward exercise.

PROGRAMMING BY RULES

Programming by rules implements a modular approach and set pattern that combines the derivation of related variables, much like the structure of specifications.

A good example is Age versus Body Mass Index (BMI), where the variables AGE and BMI are derived in two separate rules, simply because they are not related.

Programming by rules implements a modular approach and set pattern that combines the derivation of related variables, much like the structure of specifications. The macro is a utility that manages a list of intermediate data sets and associated variables so that the final output data set can be a simple affair, consisting of one-line, or span multiple DATA steps and SAS procedures.

RULE MACRO APPROACH

The rule macro approach will result in many small data sets, conceivably one for each variable in extreme cases, that will need to be combined into a final data set. After each rule, we register the intermediate data set, the derived variables, and any necessary merge keys (%rules_register%) and at the end use that information to create the final output data set (%rules_make%).

Regardless of our coding approach, we need to consider that a derivation can be a simple affair, consisting of one line, or span multiple DATA steps and SAS procedures.

RULE MACRO APPROACH

The rule macro approach will result in many small data sets, conceivably one for each variable in extreme cases, that will need to be combined into a final data set. After each rule, we register the intermediate data set, the derived variables, and any necessary merge keys (%rules_register%) and at the end use that information to create the final output data set (%rules_make%).

Regardless of our coding approach, we need to consider that a derivation can be a simple affair, consisting of one line, or span multiple DATA steps and SAS procedures.

META-DATA DRIVEN

Changes to the specification (Figure 1) are subtle if we want to control derivation from meta-data.

- A rule reference
- Rule options
- Inclusion flag

The inclusion flag uses

- Required
- Impractical
- Not Applicable

In this case, Permissible means use it if it is derived. We could also simply rely on Yes and No.

The concept of having meta data generate programs (%rules_make_program%) is not far off given that we have a pattern for rule programming, conversion for standard rule macros and the documentation to drive the process.

A natural extension would be to wrap this entire process, from a book of rules to meta-data creating the programs, with a suitable interface and the promise of point-and-click reporting is a reality. The added benefit is that the approach retains a high degree of flexibility with regards to both analysis and the associated programming

The Macros

The macro is a utility that manages a list of intermediate data sets and associated variables so that the final output data set can be easily created.

%rules_register()

The macro creates an output data set based on the intermediate data sets and variables as defined in the registry, starting with the core variables.

%rules_make()

Macro that creates one or more standard programs using the meta data in the specification.

%rules_make_program()

Macro that creates one or more standard programs using the meta data in the specification.

Conclusion

We all advocate, practice and adhere to programming by the rules set out in study protocols, Statistical Analysis Plans and associated specifications. In practice, we are practicing meta-data driven programming, or programming by rules, as we progress from source data sets to final report deliverables, but the path from one-off programs to SAS programs generated entirely from meta-data will require shifts in programming conventions, small new utilies and the predictable updates to input specifications.

The programming approach discussed highlights the results of taking a programming process from one-off programs to entirely meta-data driven with the flexibility you would expect. Add a simple meta-data interface and you have the ability to create study report deliverables with the click of a button while retaining a flexible analysis and programming process.