THE ROLE OF SAS PROGRAMMERS IN CLINICAL TRIAL DATA ANALYSIS

Ming Wang / Independent SAS Consultant

Abstract

This article shows in-depth the role of SAS programmers in clinical trial data analysis. It outlines the task flow of clinical trial data analysis, discusses project programming planning and documenting procedures, describes SAS programmers' tasks and programming skills, and provides insight on how to work with people in your team: clinical data managers, statisticians, programmer manager and project manager.

This paper is a valuable overview for programmers who are starting out or plan on being involved with clinical trials, and for data managers and statisticians who work with SAS programmers in a clinical trial team. It is also a valuable management guideline for experienced SAS programmers.

Introduction

SAS is widely used in clinical trial data analysis in pharmaceutical, biotech and clinical research companies. SAS programmers play an important role in clinical trial data analysis. In addition to doctors and clinicians who collect clinical trial data, the group conducting data analysis includes statisticians, clinical data managers (CDMs) and SAS programmers. Statisticians provide the ideas and methods of the data analysis, clinical data managers manage the collected data and control the data quality. In between, SAS programmers implement the analysis methods on the collected data and provide the study summary tables, data listing and graphs to the statisticians and/or clinicians to write study report. SAS programmers work closely with statisticians and data managers. They provide the link between raw data and the analysis. This paper discusses the SAS programmers' roles in the clinical trial data analysis task flow, describes the SAS programmers' tasks and skills, and provides insight on how to work with people in the team.

Task Flow of Clinical Trial Data Analysis

A Case Report Form (CRF) designed for a study is used to collect clinical trial data. The collected data are stored in a corresponding database. These data will be analyzed and the results will be included in the study report.

An example task flow of clinical data analysis is shown in Figure 1. The detail steps may vary from company to company.

Final Blank CRF

After the CRF is designed by clinicians, the data analysis group should review the CRF and make sure that all the fields for analysis can be computerized. The final CRF will be used to design the database and distributed to the site to collect data. In the real world it happens sometimes that different versions of CRFs are used to collect data. This will cause extra work in data analysis.

Annotate CRF

For each section of the CRF, for instance demographic section, adverse event(AE) section, etc, a database table is usually designed. For each question and/or answer on the CRF, a field in the database table is designed. The attributes of the fields, such as, field type (numeric or character), length, format, etc. are also considered. To annotate the CRF is to design the database on paper by writing down the name of the table, single or multiple records per patient and/or visit, field name, type, length, and associated format. The database tables will be converted to SAS data sets eventually and an annotated CRF with the data set name, variable name, type, length, and SAS format is very helpful for programming and data analysis. Though CDMs may write an annotation for the purpose of database design, it is suggested that SAS programmers write another one for the purpose of SAS programming.

Database Design and Testing

In some companies SAS programmers also design the database, while in other companies only CDMs design the database. A well designed database provides convenience in SAS programming. Before entering real data into a database, the designed database needs to be tested using test data and real data to make sure it works the way you expected. The test result should be documented and approved by managers.

A data entry instruction may be written along with the designed database to specify general data entry rules, e.g. all character fields are capitalized, etc. and specific rules for each file if needed. This will help data entry people to have a better understanding of the database and will result in higher quality data entry.

Pre-entry Review

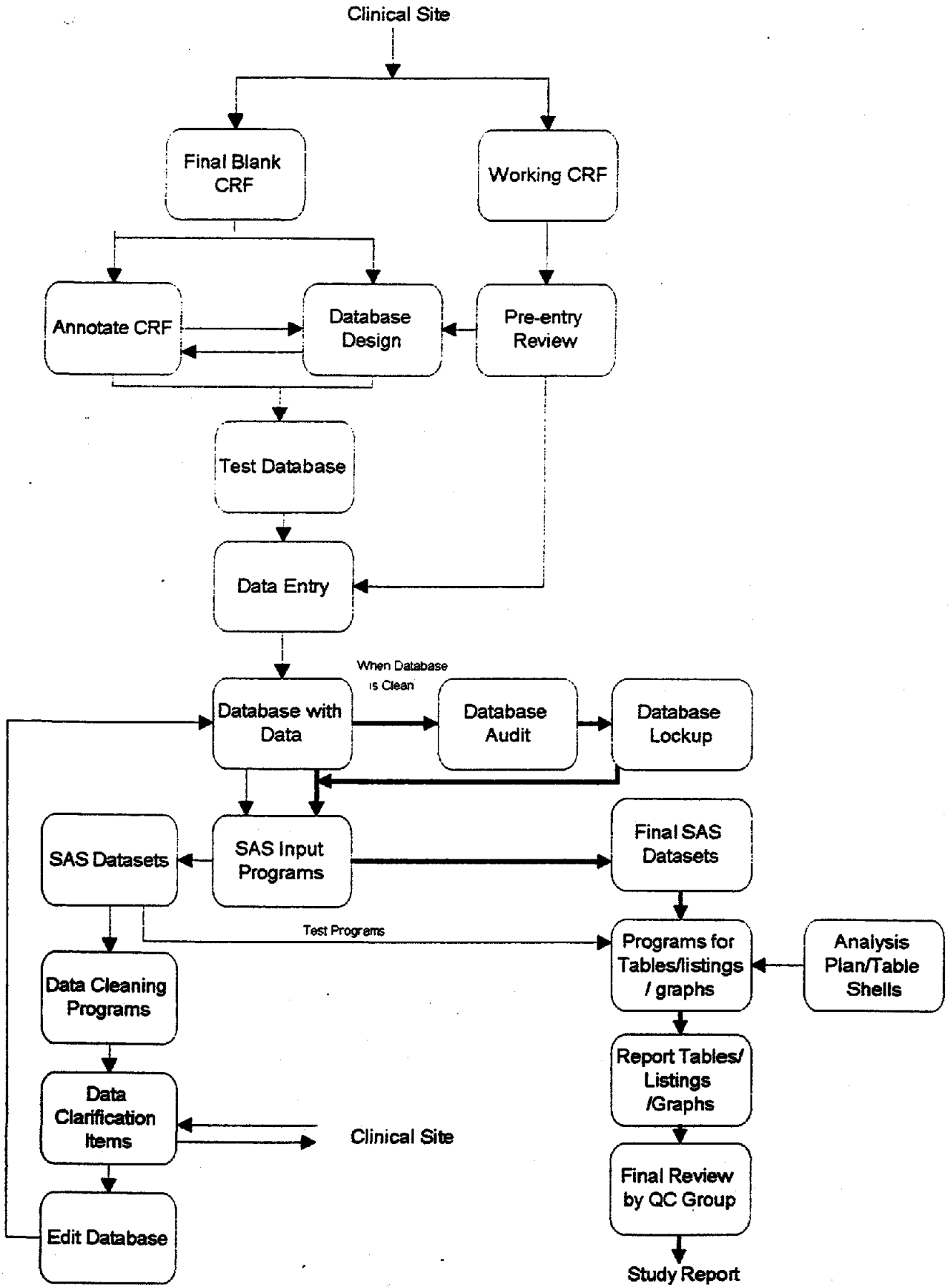
Clinical Site
Final Blank CRF
Working CRF
Annotate CRF
Database Design
Pre-entry Review
Test Database
Data Entry
When Database is Clean
Database with Data
Database Audit
Database Lockup
SAS Datasets
SAS Input Programs
Final SAS Datasets
Test Programs
Programs for Tables/listings / graphs
Analysis Plan/Table Shells
Data Cleaning Programs
Report Tables/ Listings /Graphs
Data Clarification Items
Clinical Site
Final Review by QC Group
Edit Database
Study Report

Figure 1

When CDMs get the working CRFs from clinicians they need to make sure that the data from the CRF can be entered into database. This procedure may provide input to database modification and generate manual data clarification items.

Data Entry
Usually the data entry group has a standard operating procedure to ensure the data entry quality.

SAS Input Programs
SAS programmers need to write programs to read database data to SAS data sets and also generate a SAS format library. A program to print out data contents and data book is also useful.

Data Cleaning System
A SAS programmer will write SAS programs to clean the database data according to the edit specifications provided by CDMs. This data cleaning system can be designed to run at night. Data clarification items will be generated by this data cleaning system and will be sent to the clinical site for a resolution.

Edit Database
CDMs will edit the database according to the resolution of the data clarification items. The data cleaning cycle will repeat until all of the data are entered and all of the queries are resolved.

Report Table, Listing, and Graph Programs
In the mean time, while data is being re-entered and cleaned, SAS programmers may start to write SAS programs to generate report tables, listings, and graphs according to the statisticians' analysis plan and tables shells. Drafts of tables, listings and graphs will be reviewed by the statisticians and CDMs and more data clarification items may also be generated in this process.

Database Lockup and Final Report Tables, Listings, and graphs
When the database is clean, it will be locked and final SAS data sets will be generated. An audit for the entire database or random selected points may be performed to ensure the quality of the database. If the audit is passed, the SAS programs will be rerun to generate tables, listings, and graphs, which will be reviewed by the quality control group and final tables, listings and graphs will be generated and included in the study report.

In some companies there is a SOP (Standard Operation Procedure) for each box in the diagram. It describes each of the detailed steps in the procedures.

SAS Programmers' Tasks in Clinical Trial Data Analysis

Project Directory Setup
For each project create a project directory. In each project directory, the basic subdirectories are the following, though you may not need all of them depending on the system design in your company:

- raw data (database data or ASCII files)
- SAS input programs (SAS programs to convert database data to SAS data sets)
- SAS format libraries
- SAS data sets
- data cleaning programs
- SAS macro library
- listing programs/output
- table programs/output
- graph programs/output
- miscellaneous programs
- documentation/memos

In each directory, it is convenient to create an index file to indicate program names, author and table titles, etc.

You may put the above directories in a development directory and a production directory. Programs are developed and tested under the development directory. programs can be copied to the production directory after they are fully tested and reviewed by a programming manager, and can be run under the production directory to produce final tables.

There may also be a group wide macro library and SAS format libraries. You can specify your macro libraries, format libraries, system options, etc in you autoexec.sas file.

Writing Programs to Convert or Read in Data from Database to SAS Data sets
There are many ways to convert data from a database to SAS data sets depending on the system setup in your company. For example,

- Use SAS/ACCESS to access database data directly and save to temporary or permanent SAS data sets.

- Output database data to ASCII files, and write SAS programs to read data from ASCII file to SAS data sets.

- Use DBMSCOPY, software which can convert database data to SAS data sets, or vice versa.

In addition to data sets, you also need to create format program to generate format library. You can create more than one SAS format libraries if necessary.

Writing Programs to Conduct Data Cleaning

In many companies, an automated data cleaning system is created. It still needs to be updated or revised for new projects. A data cleaning specification usually is written by CDMs to specify what to check on which data set and which item.

The checking is usually performed within a data set or across data sets. It includes range checking, missing value checking, logic checking, protocol violation checking, etc. If a variable fails to pass the checking rules, the name and value of this variable will be put into a data set along with the key variables in the same record, such as patient number, visit data, etc. CDMs will review and edit this data set and check whether this "error" is caused by data entry or originally recorded on the CRF. If it is a data entry error, it needs to be corrected in the database. If it is originally recorded on the CRF, CDMs need to send the data clarification form to the clinical site. CDMs also need to enter an identifier to the records he/she reviewed so that the system won't generate the same data clarification item again in the next run.

Write Programs for Study Tables, Listings, and Graphs

Programming report tables/listings/graphs (T/L/G) is the major task of SAS programmers in a clinical trial study. Appendix listings are lists of variables by patient; report tables are the tables containing summary information across patients, and usually are counts and statistical results; graphs can present information for single patient or summary information across patients. In T/L/G programs, there are usually two parts. The first part is to prepare the data set through SAS data steps and SAS procedures to get the data set and variables for the output. The second part in T/L programs is usually data _NULL_ to put the value of output variables to each column and row along with table titles, column titles, etc. The second part in graph programs is SAS graph procedures.

– Prepare Data set

Common programming strategies in this step include subsetting data sets; combining data sets; reshaping data sets, such as, changing multiple records per patient to single record per patient or vice versa; select statistical parameters from statistical procedures, such as mean, median, minimum, maximum, standard deviation, p-value, and save them in a SAS data set, etc. It requires programming and data analysis skills. You need to:

- Understand the data structure, such as single or multiple records per patient, single or multiple records per visit, etc.
- Understand the rules in the study. For instance, the algorithm to collapse AE records. The algorithms for collapsing AE records in AE listings and summary tables can be very complex depending on how the data were collected. In the listings the variation of the severity, study drug relatedness, etc. of a continuous AE may be shown. In the summary tables a patient may be counted only once for his most severe AE.
- Understand the methods of calculation. For instance, to calculate the mean of drug dose over all the patients, where each patient was dosed at many study periods, the mean dose for each patient may be calculated first, and then the mean of these individual means is calculated. Another example is to calculate the percentage of the counts in each cell in the table. The denominator could be the overall total, or the row total, or the column total. You need to confirm every detail of the calculation methods with statisticians.
- Understand how SAS works.

– Output Listings and Tables Using Data _NULL_

Using Data _NULL_ and PUT statements, you can output variable values along with table titles and column titles to an output file. The following items are the common issues in this step.

- Formatted output. Character variables may have right or left justified output; numerical variables need to be lined up by decimal points; use formatted values for coded variables.
- If output data set is sorted by patient and visit date (vis_date), output patient number only at first.patient, output visit date only at first.vis_date.
- Page Break Point. Page break point can be designed for each new patient number, or visit number, or at any point when the page is full. For a continuation of a patient from previous page, print the patient number following with (Cont'd).
- Calculate the position of each column. Some SAS programmers wrote macros and tools to enhance the flexibility in this step. These methods make it easier even when you delete or add columns later on.
- Print program names, generation date, database lockup date, variable values, etc, in the header or footnote. Make the above variables into macro variables for the portability.
- Wrap long text fields. For instance a 200

character comment field needs to be wrapped to 30 characters and put under the comment column. Macros have been developed by SAS programmers to handle this issue. They also work for wrapping more than one long text fields.

- Print page 1 of n.

-- Common SAS Graph Procedures
PROC GPLOT, PROC GCHART and SAS/GRAPH statements are commonly used. PROC GREPLAY can present more than one graphs on one page.

Miscellaneous
Database Auditing, Spell checking, data book programming, etc

Create Tools
Write SAS macros or Programs to create tools to enhance efficiency and standardization for your project and/or the entire programming group.

Study Documentation
The following items could be included in the study documentation.

- Backup and retrieval information on archived files.
- Study directories.
- Study data sets, programs, output files, and report table/listing/graph titles.
- Rules and algorithms used in the study data analysis.

Backup Data sets and Programs
At the end of each study, archive all study related data sets, programs, output files, memos, etc. on tapes or diskettes. They will be put to storage along with other paper documents, such as CRFs, etc.

Basic Skills SAS Programmers Should Have

- Base SAS, SAS procedures, common Statistical procedures, SAS/Graph, SAS Macro Processing.
- Operating Systems and editors
- File transfer between different operating systems, such as between Unix and PC, including text file and SAS data sets, etc.
- Convert data set from spread sheets or other database to SAS or vice versa.
- Experiences in programming using C, Fortran, etc.
- Database skills.

Work with People in Your Team

SAS programmers are the link between the data and the report tables. So you are also the link between statisticians and CDMs. It is your responsibility to identify inconsistencies between the report table specification and the data in the database, report these inconsistencies and find a solution through team discussion. A regular team meeting is very helpful to summarize, discuss and solve these kinds of problems.

Through team meetings you also can confirm the algorithms or rules you use to subset data sets with both CDMs and statisticians. These rules may be related to protocols, CRF design and the way that the data are recorded. You also need to make sure that the database is consistent for you to apply these rules.

Working with Clinical Data Managers (CDMs)
You are going to work with CDMs on the data and the database. You may or may not get involved in database design and the data cleaning process depending on each company's setup. CDMs will provide you a database with data. During the process of generating report listings, tables and graphs, report any data problems to CDMs (such as duplicate records, missing code for AEs, invalid data, inconsistent coding methods, etc.) through e-mail or memo and keep a copy for yourself for tracking and documentation purpose.

When you print records with data problems to CDMs, print the records with key variables, such as patient number, visit date, etc, so that CDMs can locate the records easily from the database. Print a title or write a memo in the printout to indicate the problems in the printout.

Working with Statisticians
You work with statisticians on report listings, tables and graphs.

First you'll have a discussion with the statistician about the specifications the statistician provides to you. The specification may be very detail oriented with variable names and algorithms, or just a table shell. A very efficient way I found is to study the table shell, write down the data sets, variables, and possible algorithms you need for each listing and table. Discuss your understanding and plan with statisticians before you really start to program. This process gives you an opportunity to understand the database and the tables. Often you may find an inconsistency between database and the table specifications. Statisticians may change the specifications

based on your input. Keep copies of confirmed specifications. After you've finished a table or listing, the statistician will review it to make sure that it is right and accurate. Before you give the tables and listings to the statistician, some QC steps are necessary.

- Check the SAS log to see whether there are any errors, warnings, uninitialized, etc, messages. Make sure that there are no bugs in you program.

- Check your programming logic and accuracy. Check listing results against database printout (data book). Check tables against listings. Check statistical numbers against statistical output.

- Check format and layout. Make sure variable formats are used correctly. Numbers are lined up by decimal points; calculated numbers, such as means, are rounded first then formatted to the output, etc. if calculated p-value is 0.0000, then present it as <0.0001. Also check spelling, justification, etc.

The fewer errors that a statistician finds from your listings and tables, the higher grade you may get from him/her at the annual evaluation.

Working with The Project Manager

The Project manager can be the statistician, or clinical scientist. Discussing with the programming manager, the project manager will design time lines and deadlines. So following time lines and meeting deadlines are the most important things that the project manager expects from you. In some companies, you also need to be aware of the budget. If you follow time lines and meet deadlines, you usually help managers to control the budget. Writing down your own plan for each section of work is very helpful for you to meet the "big deadline". Keep records that explain any difficulties in meeting the scheduled deadline. Maybe it is because the database is not clean, specifications changed, or a tough SAS programming issue. Time lines are usually developed by experienced programmers. In addition to planing the time for studying specs, writing programs, checking results, you also need to make sure that you have time for revisions. Designing a precise time line is difficult. Having good communications in the group is the most important thing to adjust internal deadlines in order to meet the big deadline.

Work with The Programming Manager

The programming manager is usually your supervisor. He/she may or may not be in the project that you are working on. The programming manager will help you solve your SAS programming or computer problems. Keep the programming manager informed of communications within your project team, your deadlines, and your difficulties in meeting these deadlines. The programming manager also expects you to develop macros, tools or new programming methods for the whole group to enhance efficiency and standardization. In some companies, the programming manager is the primary reviewer of you performance in your annual evaluation. It is important to show that you can independently work in a project as a programmer, and in the mean time constantly provide the programming manager your performance information.

Summary

Enhancing SAS programming skills, accumulating clinical trial data analysis experiences, and keeping positive attitude in a team work environment are the keys to success.