PHUSE EU CONNECT 2023 – SA12

Unleashing the role of a statistical programmer in expediting a submission

Sowmya Singuluri, Novartis Healthcare Pvt. Ltd., Hyderabad, India

ABSTRACT:

Being an integral part of pharma world, we know the magnitude of impact created when the drug is made available to patients with critical ailments even by a day ahead. As statistical programmers, we strive hard in all possible ways to plan study activities and deliver the analysis reports well ahead of planned timelines. There are a few key areas to focus on as a statistical programmer in each phase of the study conduct which paves way to deliver the study package to health authority with high quality and double the speed. To mention a few

a) Providing correct annotations especially for study specific case report forms ensuring CDISC standards are rightly implemented which will avoid unnecessary post-production changes.

b) Staying a step ahead with lean statistical analysis plan to ensure all the reports generated would be a part of your CSR and using R shiny apps for exploratory analysis.

c) Drafting a clear and clean define compliant programming data specifications.

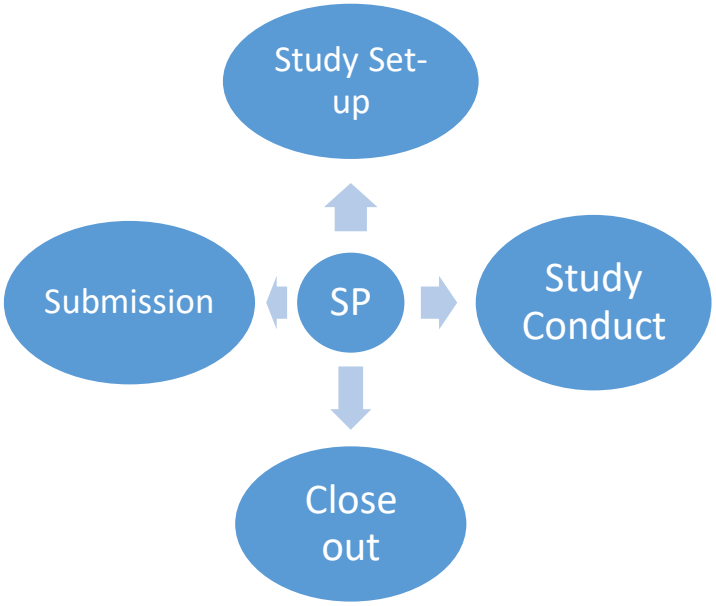
d) Programming dynamic algorithms and using SQL over data step aiming for less program execution time and performing systematic data review linking annotated case report forms, SDTMs, ADaMs and TFLs which has a huge impact on identifying any outlier data.

In this paper, I would like to discuss in detail on all the key focus areas showing how as statistical programmers we can contribute to these factors resulting in expediting submission timelines.

INTRODUCTION

Many often think that statistical programmers just implement the analysis methods on collected data and create tables, figures and listings and share it with their stakeholders like statisticians, clinicians and medical writers for writing the clinical study report. But, there is more we do than just creating TFLs. A statistical programmer plays a key role from drafting stages of a protocol until the final stages of submitting data package to health authorities.

To have a more precise discussion throughout the paper I have classified clinical trial process into below categories and would be discussing what a programmer can contribute to expedite the final goal of submission.



Study Set-up:

Protocol and SAP Review: A clinical trial starts with drafting the protocol. We are deemed to collect and analyze all the data points specified in the protocol. Hence, the core clinical team must be very clear and precise about the data to be collected and analyzed which is required to derive the primary and secondary endpoints in question. Often it is observed that along with primary and secondary end point analysis extensive exploratory analysis will also be specified in protocol. Here a statistical programmer must make every effort make team understand how data exploration and visualization tools like R can be used to explore the data instead of having everything mentioned in protocol to understand the other characteristics of data where programming team ends up in creating bunch of tables for the same and out of which only few would actually be utilized for reporting in study report. By doing so, the team would be able to utilize time and resources for primary analysis which will increase the quality of the reports and help team delivering them ahead of planned timelines.

Once protocol is available statistical analysis plan is the next key document to review. A programmer must be very clear with SAP and understand each and every analysis derivation as this will be translated to a programming specification document. With close collaboration with statisticians a programmer must make every effort to have only sub-group analysis which is having a direct effect on the population. If the team likes to see any other major sub-group effects, this can be well achieved using R shiny apps instead of creating huge number of by sub group tables.

Example:

- In one of my studies during the SAP discussion, we were able to make team understand and remove few duplication of analysis which were a replica based on different population sets like shown below and also quite a few listings. This was not as easy as said but took many negotiations with regulatory, clinical and writing colleagues.
    - Adverse events by preferred term: X analysis
    - Adverse events by preferred term: Y analysis
    - Adverse events by preferred term: Z analysis

- Involving regulatory colleagues at the very beginning stage helps quite a lot as they are the one who are very much aware of what the health authorities' expectations are. This resulted in a

significant reduction in number of outputs which gave us a lot of room to work on the required analysis.

- This subsequently resulted in planning the clinical summary report writing as the shells and writing the appendix was drafted before the final lock.
- Programming team had enough time to work on the submission package well in advance which is a big positive catalyst to expedite the submission.

This has a very high impact on all planned deliverables. We are well aware that TFLs are not generated just with a simple click of buttons and always involves two programmers as developer and validator. There are lot of processes involved in executing and creating these reports. By reducing the number of reports the team can be very prepared by the time of data base lock and can deliver the required output for writing as soon as possible and utilize the resources for package preparation and can negotiate timelines to submit ahead of actual planned time.

Annotated CRF: Annotated CRFs are effective in extracting structured information from unstructured data points. In the context of submission, they can be used to identify and extract relevant data or any specific information that needs to be collected and organized. By accurately annotating CRFs we are directing the data to their respective SDTM domains. aCRF should be created very early in the process as soon as the protocol is ready. Once we have a blank unique CRFs in place a programmer annotates each CRF page and creates a structure for the data as per CDISC guidelines. Hence this is a critical step to get the data in required datasets and in required format. aCRF is one of the submission document in SDTM define package.

How is this helping in expediting the submission?

Before discussing about how this will help expediting submission, one has to clearly understand how annotation of CRF defines data collection in required CDISC format. Below is an example of an annotated CRF.

Figure 1.1

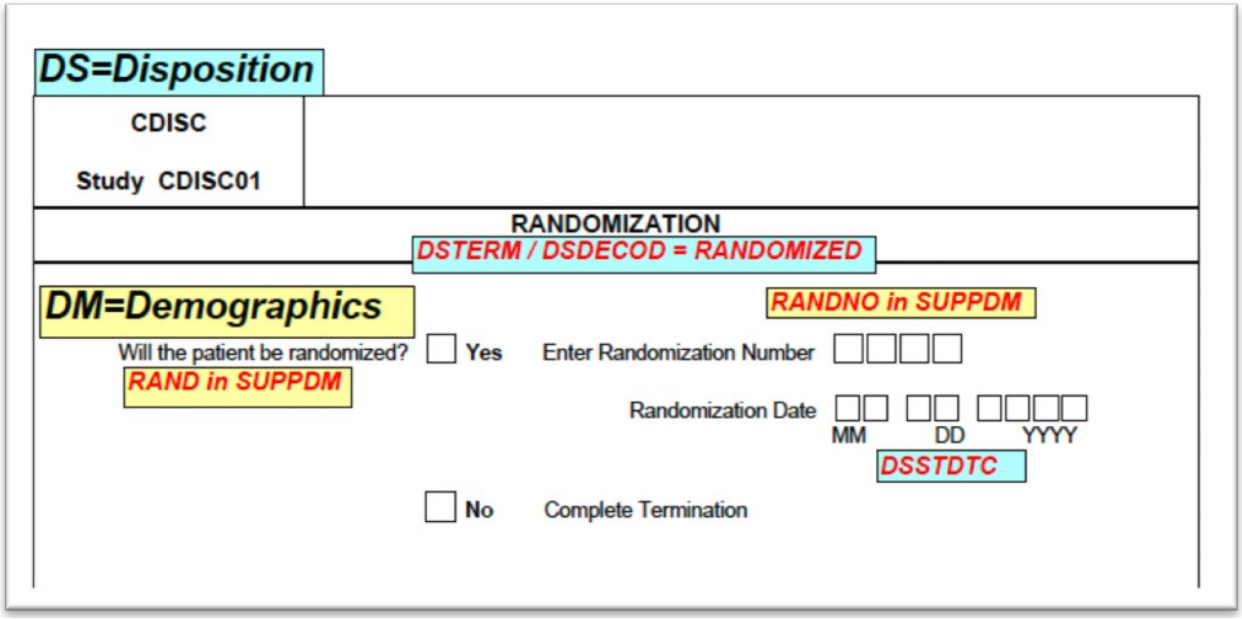
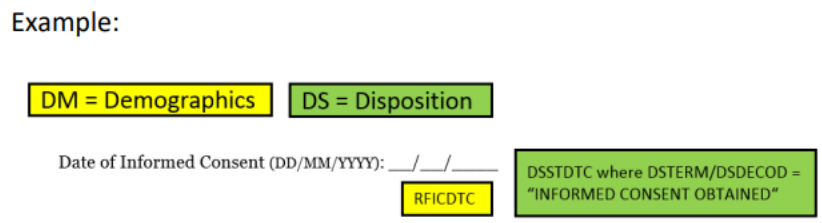


Figure 1.2



- As shown in Fig. 1.1 the data from this page is expected to go to both disposition and demographic domains. Data received from the sites to source systems from this page comes into sas domain. Based on the annotations provided, the data is mapped to the respective domains and is available for analysis to build SDTMs, ADaMs and TFLs.
- A special care has to be taken when data from one page is expected in more than one domain. Like shown in Fig. 1.2 informed consent date is mapped to both RFICDTC in DM and DSSTDTC in DS. If there is more than one informed consent page ex: STUDY INFORMED CONCENT, PHARMACOGENTIC INFORMED CONSENT programmer has to be very careful to annotate only STUDY INFORMED CONCENT date (or the consent date required) to RFICDTC but not other consent dates which might lead to wrong mapping of dates.
- In cases like mentioned in above or any other mistakes in annotations thus can lead to wrong mapping of data and thus which can affect analysis.
- If these are identified just before data base lock there will be unexpected consequences which will lead to delay in planned timelines for all the deliverables eventually.
- A thorough validation of annotations must be performed to avoid such last minute updates. As we know this is a part of submission package each and every variable mentioned on CRF must be present in the SDTM data. When data are recorded on CRF but are not submitted the respective field should be annotated as “NOT SUBMITTED”. This must be checked with biostatisticians that the respective information is not required for analysis. These checks are very crucial at the very beginning stage of the study.

Thus by keeping above pointers in mind a programmer plays a key role here to avoid any hiccups during the study closure with data re-structuring and which will eventually help all line functions to plan other vital deliverables.

Study Conduct and closeout:

Once source data is available to programming team starts analyzing the data and builds data derivations required to create TFLs based on SAP definitions. The process of analysis is:

- SDTM specification
- SDTM datasets
- ADaM specification
- ADaM datasets
- Tables, figures and listings

How does a programmer contribute in this phase to expedite a submission?

- **Writing define compliant SDTM/ADaM specifications is the first step in this phase.**

- Programmer must be very clear about the versions of implementation guides (SDTM IG/ADAM IG) used for the study which is decided at the time of set up phase of the trial.
- Algorithms in specification must not be written as program coding language but as a detailed and meaningful text in plain English should be provided. This will save a lot of time during package preparation. Specification is the key input for define xml.
- A thorough review is required in this step and programmer must have repeated discussions and alignment on clarifications with statisticians is a must as the coding and analysis is programmed based on the logics provided in the specification.

- **Dynamic/defensive coding must be followed in all levels of programming.**

  - Example1: In double blind studies if dummy data is being used from a different library other than source, coding must be done in such a way that once after the concerned data is available in source location that data will be read instead of dummy. This will avoid updating codes after data base lock during which we cannot afford any extra amount of time for these level of updates rather spend time on refreshing the data and doing final reviews before releasing it to stakeholders.
  - Eaxmple2: The execution time of a programmer is one of tough nut to crack while dealing with huge data where you have thousands of subjects randomized. This will in turn save time to execute data post lock and deliver the reports as early as possible. A simple example of reading a huge lab data from a study with millions of records shows a significance difference in run time. Proc sql was used for one with less time and data step was used for one with more time. We were able to bring the execution time of one questionnaire dataset from 1.5hrs to 40 minutes. This will eventually reduce the execution time of the output files and will give team more space for reviewing and delivering the reports.

```
NOTE: Remote submit to SERVER commencing.
NOTE: Table WORK.LB created, with 4234560 rows and 70 columns.

NOTE: PROCEDURE SQL used (Total process time):
      real time           2:28.07
      cpu time            36.55 seconds
```

```
NOTE: There were 4234560 observations read from the data set PAR.ADLB.
NOTE: The data set WORK.QS has 4234560 observations and 70 variables.
NOTE: DATA statement used (Total process time):
      real time           4:06.10
      cpu time            35.77 seconds
```

- **Planning retrospectively**
  - Data availability is just not enough to start planning for analysis. We must know what the final destination is and the rules and regulations to reach. Hence we can plan for a hassle free submission only when a programmer knows complete in and out of the submission components and strategies. Few simple examples:
    - If a programmer do not know that there is a size restriction on datasets for eCTD submission to health authorities. This will lead to a lot of re-work where dataset has to be split based on some meaningful criteria with clear instruction as in how to combine it during the time of review.

- If they are not aware in what format dataset programs are submitted, if there are too many internal macros how this has to be handled?
- If team is using any other programming language is this clearly mentioned in briefing book and there is a consensus with respective health authority or not.
- If they are not aware of concept of CSPD/SCS/SCS/CO and many other eCTD components where programmers are actively involved, this will lead to a huge problem at the time of planning and delivering the components.
- Checking of hyperlinks in all the relevant define documents like define xml, sas pointers of usage etc. enough time has to be planned and this is possible only when we know this is a very important step before uploading the package in eCTD component.

In a nut shell it is key to plan from backwards by understanding all the eCTD modules where programming team is involved. By doing so the programming team will be always a step ahead of delivering the required deliverables making the journey to destination i.e. submission more peaceful with high quality.

Submission:

This phase would be as smooth as it can be only when all the programming components were completed in a define complaint process as discussed above. It is always suggested to initiate the process of package preparation as soon as possible and not to sit back and wait for all the analysis to be completed. Below are some key points to keep in mind which are often missed and might lead to resources spending more than required time on updating them again and again.

- SDTM component:
  - Pages must be rightly hyperlinked in define xml and all the variables annotated in CRF must have their origin as CRF in derivation section of define.
  - Checking the key variables specified in the dataset definition section if uniqueness in data after using those key variables exists or not.
  - Code list values especially the non-extensible code list dependent variables must not contain any values that are not part of the code list. These points must be taken care at the CRF set stage.
  - If legacy studies are part of submission a clear planning at what level conversions are planned? SDTM or ADaM. Thorough traceability checks performed on converted data versus the old format data.
    - Example: If the CSR for that legacy study is previously submitted to health authority the converted data should be in line with that analysis reports otherwise programmer must discuss with all relevant line functions as to how this must be explained in reviewers guide.
- ADaM Component:
  - Analysis results metadata is one key part where we must know which reports derivations would be a part of it.
  - Any complex derivations must be clearly drafted and must be a part of appendix in reviewers guide.
  - It is advisable to have Conformance report section ready by the database lock, this means that each and every issue in p21 should be addressed and the entire clinical trial team must come to consensus on that so that there is no unforeseen updates to data post analysis.
- Pooling Component:

- Programmer must be in regular touch with the submission leads for the project and must be clearly aware if there is a pooling planned as a part of submission. A simple yet powerful key for a successful pooling is creating a project level standard in common derivation logics so that while combining the data re-derivations can be avoided.
- With a prior knowledge and planning of the submission strategy a programmer will be able to plan for SCS/SCE/CO more efficiently rather than starting everything at the end which creates a lot of pressure which in turn reduces the quality of the delivery further complicating the plan.
- Often there would be common type of analysis included as part of RMP (Risk management plan) along with SCS but programming team must have complete knowledge about any specific flagging conditions to be used for them separately.
- ADRs (Adverse Drug Reaction) and labelling is a very key process as a part of safety of the compound and programming team needs to provide a table and a thorough knowledge about the process helps in preparing the same.

Many often think that there is nothing much left in programming team's plate after delivering reports for CRS but this takes as much time and effort like planning and conduct phase of a trial. The crux here is this will be as smooth as it can be only when the programmers are very well aware of these various components so that planning begins well ahead which ultimately results in expediting a submission plan.

Conclusion:

As discussed above many of the components might not seem to have a direct impact but it will make a huge difference with all the timelines. This also results in a very structured process of submission. It is always said that too much of anything is good for nothing, likewise giving too much of information about the study is not at all a requirement and health authorities would not expect too. Analysis targeting primary and key secondary objectives and Safety profiling, knowing the complete submission strategy, including regulatory team in discussions and collaboration with all the required line functions will help in everyone to clearly see what huge impact programming team is capable of creating in submission of a project/study.

References-

1. Electronic Common Technical Document (eCTD) | FDA
2. The Do's and Don'ts of Define.xml | Pinnacle 21
3. Pedersen, R., & Both, N. (2016). QC of SDTM and aCRF using SAS. In Proceedings of the PhUSE Annual European Conference
4. Lyma Faroz, Seagen Inc., Bothell WA. First Time Creating a Submission Package? Don't Worry, We Got You Covered!
   First Time Creating a Submission Package? Don't Worry, We Got You Covered! (pharmasug.org)

Contact information-

Your comments and questions are valued and encouraged.

Contact the author at:

Sowmya Singuluri

Novartis Healthcare Pvt. Ltd

sowmya.singuluri@novartis.com