PhUSE US Connect 2018

Paper AB12

Statistical Analysis Plan – Clinical Programming Reviewers Guide

Xiaoyin (Sherry) Zhong, GlaxoSmithKline, Collegeville, PA
Dan Schramek, GlaxoSmithKline, Collegeville, PA

ABSTRACT

A Statistical Analysis Plan (SAP) describes the planned analysis for a clinical trial. SAPs must be carefully reviewed by programmers for clarity, comprehension and to ensure sufficient detail is present to unambiguously construct analysis data sets and prepare planned Tables, Figures, and Listings (TFLs). SAP review is a challenging task that requires both statistical expertise and abstract thinking skills, and is at times completed in the absence of data. Clinical programmers must develop critical evaluation and communications skills to identify issues and seek clarity on planned analyses. Clinical programmer's education is highly valuable and is from multiple disciplines, suggesting that comprehensive training on SAP review is an essential skill to ensure consistency and quality within the project. This paper provides guidance on how a structured comprehensive SAP review can be conducted and how awareness of statistics can be increased for clinical programming, with a resulting quality improvement of statistical programming study activities.

INTRODUCTION

Clinical studies are complex scientific experiments designed to provide evidence to answer questions regarding the safety and efficacy of products. Furthermore, data generated as part of these clinical studies are used for regulatory applications and/or communications of study results in manuscripts, marketing materials, or other symposia. During the conduct of clinical studies, the programmer is presented with documents prepared to support the study design, data collection, and analysis and reporting. The SAP is one of these documents that is of critical importance. The SAP provides the clinical programmer with relevant information and detail on the scope of planned analyses, population definitions, and methodology on how prospective decisions are to be made for presenting study results. A key reference to the role of the SAP is outlined in ICH E9 Statistical Principles for Clinical Trials)[1] which states: "The reporting and analysis plan may be written as a separate document to be completed after finalizing the protocol. In this document, a more technical and detailed elaboration of the principal features stated in the protocol may be included (see section 7.1 of ICH). The plan may include detailed procedures for executing the statistical analysis of the primary and secondary variables and other data". Reporting and analysis plans may also be known as Data Analysis Plans (DAP) or Statistical Analysis Plans (SAP) in other organizations.

Clinical programmers must develop document review skills that foster reading content, comprehensive understanding, and critical review of the SAP to support programming of analysis data sets along with analysis and presentation of study results. This document provides insight on best practice and methods for clinical programmers on how to review SAPs.

PURPOSE OF A SAP

The SAP describes the planned statistical analysis of a clinical study as outlined in the protocol. In contrast to the protocol which outlines the analysis, the SAP is a technical document which describes the statistical techniques for study analysis in detail. The SAP defines all the statistical output which will be included in the Clinical Study Report (CSR). Some companies add mock example shell tables, figures and listings in the SAP, although they may be removed for disclosure purposes. The SAP and the annotated Case Report Form (CRF) are documents which are most often used by clinical programmers to create their deliverables. The SAP is a major constituent of all the documents used by clinical programmers to create their deliverables. In general, there are four different types of analysis plans in the clinical development of a compound:

Type of Analysis Plan	Description
Clinical Study SAP	• Describes the planned statistical analysis of a study. • Early drafts of the SAP can be written in parallel with protocol development that allows key input into operational/data management study setup activities.
INTERIM STUDY SAP	• Describes the planned statistical analysis of an interim analysis for a study and therefore needs to address handling of partial unblinding issues in case of

Type of Analysis Plan	Description
	blinded studies. It also describes the possible impact on the final analysis.
DATA MONITORING COMMITTEE (DMC) SAP	• Modification of the interim analysis used for DMCs and describes regular (e.g. monthly) data monitoring procedures for safety or efficacy. The DMC SAP is also supplanted with a DMC charter which clarifies names and responsibilities of the involved parties.
INTEGRATED SAP	• Describes the planned analysis for an integrated analysis which is used, for example, in submissions. It defines the details of programming output for an Integrated Summary of Safety (ISS) and Integrated Summary of Efficacy (ISE) usually in one document.

For the following section, the reader should keep the SAP for a clinical study in mind. The described methods can be easily adapted for other programming documents. Usually the SAP is written by the lead study or project statistician by using standard SAP template. A representative SAP contains the following example sections:

- Introduction
- Summary of key Protocol Information (e.g. description of changes to protocol defined analysis plan, study objective(s) & endpoint(s), study design and statistical hypotheses)
- Planned Analyses (e.g. description of interim analyses and final analyses)
- Analyses Populations (e.g. definition of analyses populations)
- Considerations for Data Analyses and Data Handling Conventions (e.g. imputation rules, algorithms and derived variables)
- Study Population Analyses (e.g. safety set, per protocol set, full analysis set, etc.)
- Safety Analyses
- Efficacy, Pharmacokinetic, Population Pharmacokinetic (PopPK), Pharmacodynamic (and / or Biomarker) and Pharmacokinetic / Pharmacodynamic (PK/PD) Analyses (i.e. driven by study objectives and endpoints)
- Complete list of data displays (TFLs) to be generated
- Example mock shells (or reference to standards) to be generated that defines layout of TFLs)

According to ICH E9 the statistical analysis is planned a priori. Therefore, the clinical trial statistician needs to ensure that the SAP is carefully reviewed and approved prior to unblinding of the study.

The SAP is a document which is submitted to regulatory authorities as part of a submission package. The SAP is also part of the appendix of a Clinical Study Report (CSR). Therefore, the SAP is critically important for documenting all the planned statistical analyses. The SAP is also stored in the trial master file and used during audits to check if statistical programming was followed exactly as described in the SAP. The SAP is meant to be a standalone document. Besides the technical statistical details, it should contain brief descriptions and summaries of the protocol and it should not only refer to the protocol.

IMPORTANCE of AN EXTENSIVE REVIEW BY THE CLINICAL PROGRAMMER

The SAP is a technical document which describes techniques used in programs by clinical programmers. The SAP is the most important document for a clinical programmer since many deliverables from the clinical programmer are defined in this document. A critical review ensures comprehensive understanding of content relevant to statistical programming and will support improved quality of statistical programming work on a clinical study. Therefore, the awareness, correctness and detail of the SAP is critically important for the clinical programmer in the following areas:

- Consistency with study protocol on study description and purpose
- Definition of analysis populations
- Data handling rules, imputation of missing data and algorithms for derived variables
- Appropriateness of described analysis for the study
- Transferability of statistical methods into statistical programming code
- Consistency between SAP text and TFL mock example shells
- Use of standard TFL mock example shells (if applicable)

HOW TO REVIEW A SAP

The SAP is a document which is governed by the organization's standard operating procedure (SOP) and is usually built from a template. The clinical programmer must read, understand, and seek comprehension for all sections in the SAP. It is important to note that the subject matter expert of the SAP is the statistician. The statistician is fully responsible and accountable for the SAP. Clinical programmers should use their statistical background knowledge to author (where applicable), review and provide critical comments back to the lead author. Nevertheless, the final decision about the implementation of the comments is at the discretion of the trial statistician.

Clinical programmers should note during the review of any deviations from project standards or similar studies and ensure understanding of why and where these deviations occurred, as it will affect programming. Any deviations could be mentioned in the Analysis Data Model (ADaM) Reviewers' Guide to aid the reviewer and document the deviations.

The SAP review should be performed in multiple stages. It depends on the expertise of the clinical programmer if these stages can be done at the same time. A less experienced programmer could do this review step by step. A more advanced programmer can do all stages within a single read of the document. Nevertheless, every clinical programmer should keep the following points to check in mind:

- **Correctness**
- **Consistency**
    - Ensure consistency with the protocol
    - Ensure consistency with any project standards
    - Ensure consistency with data standards
    - Ensure consistency with internal requirements for outsourced studies

- **Completeness:** Check if all tables, figures and listings (TFLs) specified in the SAP text are described in the list of data displays and mock example shells (at least for the non-standard TFLs) are available. Check if all TFLs mentioned in the list of data displays and mock example shell TFLs are described in the SAP text (where applicable)
- **Degree of Details:** Is all required information mentioned and described in sufficient detail to unambiguously conduct and reproduce the analysis (e.g. baseline information, algorithms for derived variables, details in statistical models needed to setup SAS code, etc.)
- **Appropriateness**: Check if the planned statistical analysis is appropriate for the purpose (e.g. tables to describe the baseline characteristics of populations, tables for comprehensive description of the primary and secondary endpoints, etc.) Are the data collection requirements sufficient to support all planned analyses?

In the next subsections, examples for these points are further explained in more detail.

Correctness

The correctness is easy to check and can be achieved with the first SAP review and without any formal statistical knowledge. Therefore, this check will also be performed by any other SAP study team members and should not be the primary focus for the clinical programmer. The checks need to be conducted to ensure statements in the SAP are correct. This could vary from typographical or grammatical errors to correctness of scientific descriptions. For example, if a range is described as "120 < DBP < 90", this is incorrect and should be commented as "90 < DBP < 120".

A check for correctness should also include checks for the correct use of SAP template, if applicable. The document should be read for general errors or incorrect content in single SAP sections. Also, due to the standardized structure the author should not be allowed to delete or add sections on the 1st level, unless specified. The author should also not be allowed to add a Table of Contents (TOC) for all the TFLs in the SAP text. These items should be checked when checking for correctness.

Consistency

The checks for consistency are checks which are more advanced than the checks for correctness. Every clinical programmer should be able to perform the check for consistency. These are checks to ensure consistency with the protocol to checking project specific standards. In other words, the checks for consistency are comparisons of existing documents (e.g. protocol, project standard definitions, company data standards or analysis programs with the SAP text). To check for consistency with

the protocol, the following can be completed:

- Is the conduct described in the SAP consistent with the protocol?
- Are the objectives (primary, secondary & exploratory) in the SAP consistent with the protocol?
- Are the endpoints (primary, secondary & exploratory) in the SAP consistent with the protocol?
- Is the study purpose specified in the SAP consistent with the protocol?

It's important to note that inconsistencies between the protocol and the SAP could potentially exist. For example, if the clinical study team becomes aware during a data review meeting that the planned analysis is not appropriate, inconsistencies will exist. For such scenarios, a decision to change the planned analysis will be documented in the SAP, including the rationale and any impacts on study results. The clinical programmer should note these inconsistencies and comment as tracked changes as part of the SAP review process. To check the consistency between the study SAP and the project, the following should be checked:

- Did the author use available standard TFLs? If not, is there a scientific justification for the change or authors personal preference?
- Where derivation rules are descried (i.e. relative days, drug relationship etc.), check if the described algorithms specified in the study SAP are consistent with the project (i.e. if specified). This may involve a comparison of two SAP documents or a comparison of an old analysis program with the new SAP text.

It is important to note that inconsistencies are also allowed between the SAP and project standards. The clinical programmer should be aware of these inconsistencies and provide comments. For example, if another study uses a different definition of drug relationship classes, the clinical programmer should comment. If this inconsistency was intentionally made, it is important for the programmer to keep in mind that programs may need to be adapted.

Another important example is a deviation from available standard TFLs. If a statistician did not use available standard TFL shells, the clinical programmer should comment. In general, the statistician should give a justification for any deviation from standards.

Note: For data disclosure purposes, the general practice is for SAPs to be posted without mock example shell TFLs included, unless there are study specific reasons to include. The programmer should question if the SAP text and TFL shells should be separate documents.

Completeness

The check for completeness is a similar comparison as the check for consistency and of similar complexity. Every statistical programmer should be able to perform the check for completeness. These are checks to ensure that the TFLs described in the SAP text are completely covered by the List of Data Displays and the mock example shell TFLs. In other words, the checks for completeness are checks to ensure the within SAP consistency. To check the completeness of the SAP, the following checks are important:

Check	Examples
Are the TFLs mentioned in the SAP text explained in the mock example TFL shells (and the list of data displays) in the same manner?	• If the SAP text describes "analysis will be completed by visit and treatment group", do the shells also present the layout "by visit and treatment group"? • If the SAP text describes "age classes will be "<65 years" and "greater equal 65 years" are these reflective in the corresponding shells?
Are the TFLs described in the mock example shell TFLs and the list of data displays also described in the SAP text?	• If the mock example TFL shells contain AE tables for intensity, maximum intensity and relationship, are all these described in words in the SAP text, if required based on template guidance?
Is the list of data displays of the mock example shell TFLs complete?	• A check of the SAP text with the list of data displays may identify missing TFLs
Are list of data display TFLs descriptions consistent with mock example shells?	• Does the TFLs titles match with the mock example shell TFLs titles. Inconsistencies could result with an incorrect title being used in programs, because mock example shell TFLs was used instead of the list of data displays

It is of major importance to compare the SAP text and the mock example shell TFLs. The only document included in the appendix of the CSR and submitted to regulatory authorities is the SAP text, unless otherwise agreed. This is the binding document for the pre-defined analyses. Often clinical programmers utilize the mock example shell TFLs to facilitate setup of programs. If mismatches between mock example shell TFLs and SAP text are observed, it's more important that the text in the SAP is correct compared to the shells but the aim is to ensure overall consistently between the SAP text and shell TFLs.

Degree of Technical Details

The check for the right degree of technical details is more complex than the previous three checks. This check requires more statistical programming expertise and some statistical knowledge. These checks ensure that the clinical programmer has sufficient technical details to perform the programming. To check for the right degree of technical details, the following should be checked:

[1] Baseline Variables

- Are the baseline variables defined for all variables with a pre-dose measurement? Examples:
    - Is it defined that the last value before dosing is taken as baseline value?
    - For cross-over studies, is it defined that the last value before dosing in every period is taken as baseline value?
    - Is a special visit identified as baseline visit and detailed specified if some measurements are not done at this visit?

- Are definitions of baseline variables adequate and sufficient for programming purposes?

    - E.g. if multiple measurements are used to define the baseline value, is there a definition of how to calculate the single value? This is often the case with ECG baseline values where the mean or median of three single measurements defines the baseline value.

[2] Analysis Visit Windows

- Are all scheduled visits included?
- Is it clear how to handle withdrawal or unscheduled visits (e.g. are they slotted to a scheduled visit?)
- Is it clear how to handle the data should multiple results occur within the same visit window?
- Do the data handling rules fit in with baseline definition (e.g. if baseline is the last pre-dose measurement but the data handling rules state to take an average of multiple results within a visit window – how does this fit in when deriving baseline?)

[3] Data imputation

- Do I understand the stated rules?

[4] Derived Variables

- Are sufficient details for a variable provided, such that any programmer will be able to program? Are there sufficient details provided to handle unexpected values, missing data, etc.?

    - Summary Scales
    - QTcF
    - Weighted Means

[5] Summary Statistics

- Are formulas for complicated statistics provided?

[6] Denominator for Percentages Defined?

- Do the denominators used for calculating percentages make sense and are they defined clearly?

[7] Statistical Models

The author of statistical analysis code should always be a statistician and another statistician should perform an independent QC; these individuals hold accountability for the analysis programs and analysis validity. Clinical programmers would therefore not write and execute statistical models. However, if there is a case where the programmer needs to support the statistical analysis programming, the following should be considered:

- Sufficient details provided to set up the analysis programming code?
- Will I find my own way from the text to the procedure?
- Structure of covariance matrix for repeated measurement
- Non-parametric vs. parametric approach
- Survival analysis: censoring mentioned?
- Stratification?

Programming “statistical models” does require knowledge of statistical methods, including the ability to understand statistical output and identify potential issues (i.e. violation or errors). Clinical programmers should not hesitate to ask questions for deeper understanding (i.e. with the clinical trial statistician, colleagues or managers with more advanced statistical expertise). For other checks specified, clinical programmer can benefit from many years of experience. The clinical programmer is aware of the challenges with programming some algorithms and therefore vigilant review of the technical details specified in the SAP should be conducted to ensure (where required) any special cases that could appear. If a clinical programmer does not have the necessary experience, close collaboration with a more experienced programmer, mentor or line manager is expected.

Appropriateness

Although it is unlikely the programmer will assist with the statistical model programming since this is the responsibility of the statistician, the following provides more details to consider if the programmer is involved. The check for appropriateness, further extends the previous checks. To check the appropriateness of a SAP, the reviewer will require some statistical and regulatory knowledge and not all programmers may possess these skills and knowledge. The programmer is required to check if the described planned analyses is appropriate to address the study objectives. The programmer should question if everything is described that needs to be described or if there is something missing in the planned analysis. This is clearly the expertise of the clinical trial statistician. Nevertheless, a careful review by a clinical programmer ensures a deeper understanding of the statistical environment and ensures the final SAP meets the reporting requirements. For example, it might be sufficient for a Phase I study to only prepare listings for medical history. Usually in Phase I studies only healthy volunteers are included in the conduct. Therefore, a table would not provide very reasonable information to describe the population characteristics. This differs from a patient study, where it is required to generate a summary table for medical history to describe the population, due to the range of pre-existing diseases. When reviewing the SAP text, the clinical programmer should check:

- Do I have listings described where I need listings?
- Do I have tables described where I need tables?

Another example for the appropriateness of statistical models, is the confusion between an “analysis of variance” (ANOVA) vs “analysis of covariance” (ANCOVA). The obvious difference between ANOVA and ANCOVA is the letter "C", which stands for 'covariance'. In short, ANOVA compares group differences on some continuous outcome variable. ANCOVA, on the other hand, assesses to what degree the groups differ after you’ve controlled for some other, continuous, predictor (the covariate). For ANCOVA, the covariate(s) need to be defined. For example, in an epilepsy study the number of seizures per day prior to first dosing might be an important variable for the later judgment of the drug effect. In this case the number of seizures at baseline and the change from baseline or the number of seizures during the conduct should be taken into the model. A clinical programmer should be able to understand the objectives of the study and the planned statistical analysis for review and comment.

Regulatory requirements are a third example for appropriateness checks. In addition to reporting study results based on protocol objectives and endpoints, other data disclosure requirements (i.e. Food and Drug Administration (FDAAA) & European Medicines Agency (EudraCT) or other country specific) also need to be considered. For example, the number of non-serious adverse events at or above 5% is required for FDAAA disclosure.

The check for appropriateness is also very important when it comes to the actual statistical analysis. There is a huge difference in the statistical analysis if a variable is:

- Dichotomous (e.g. responder yes/no)
- Categorical (e.g. response: none, mild, moderate, severe)
- Continuous (e.g. response values can range between 0 and 100)

For categorical variables, summary statistics would not be informative, instead a frequency or shift table would be more appropriate. In contrast, a frequency table would not be appropriate for a continuous variable. Programmers should check the

described analysis is appropriate for the type of data being reported.

It is important to mention that checking for appropriateness requires many years of experience. It also requires clinical programmers who are proactive and work closely with the trial statistician. The statistician will make the final decision but should always be in consultation with the study team and programmer and justify any decisions impacting the reporting of the study. The clinical programmer should be proactive with questions as there are no “dumb” questions, it’s better to ask, rather than not to ask! Close collaboration and discussions between statisticians and clinical programmers will ensure quality SAPs. There are countless examples that could be provided to describe the check for appropriateness. In summary, it is important to think outside the box and read between the lines. It should not be assumed that everything that is written in the text is sufficient. So always ask what is missing.

TRAINING OF SAP SKILLS

Programmers must read and understand the SAP SOP prior to any SAP review involvement. In addition, increasing review skills of programmers can be achieved by regular involvement in SAP development process. Programmers with limited experience reviewing SAPs should be mentored by more advanced and experienced programmers. Feedback should be provided to programmers with limited experience reviewing SAPs, to improve their SAP review skills. It is only through experience that a programmer gains knowledge and insight for potential issues with study reporting. It is of critical importance that more advanced programmers mentor the less experienced programmers to avoid these issues.

CONCLUSIONS

The review of a SAP is an essential part of the clinical programmer’s work. The detailed description in this paper provides guidance on how a clinical programmer should review a SAP. The clinical programmer need to review and ensure the correctness, consistency, completeness, sufficient degree of details, and appropriateness of a SAP. The goal is to achieve a consistent approach and common understanding across the programming group. Reviewing a SAP requires a significant investment of time to ensure issues do not occur on the critical path for a study. A high-quality SAP will make issues less likely during the actual programming and implementation stage. Therefore, a detailed SAP review is of fundamental importance to clarify any misunderstandings as early as possible and to produce high-quality programming deliverables.

REFERENCES

1. INTERNATIONAL CONFERENCE ON HARMONISATION (ICH) OF TECHINCAL REQUIREMENTS FOR REGISTRATION OF PHARMACEUTICALS FOR HUMAN USE (Statistical Principles for Clinical Trials E9). February 5,1998. Available from: http://www.ich.org/fileadmin/Public_Web_Site/ICH_Products/Guidelines/Efficacy/E9/Step4/E9_Guideline.pdf

ACKNOWLEDGMENTS

The authors would like to acknowledge GSK colleagues Prafull Mistry, Daniel Beesen, Nicola Perry and Ryan Finch who have contributed their knowledge to this work.

CONTACT INFORMATION

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

Xiaoyin (Sherry) Zhong
GlaxoSmithKline
1250 S. Collegeville Road
Collegeville, Pennsylvania, 19426-0989
610.917.7436
Xiaoyin.x.Zhong@gsk.com

Dan Schramek
GlaxoSmithKline
1250 S. Collegeville Road
Collegeville, Pennsylvania, 19426-0989
610.917.5303
Daniel.J.Schramek@gsk.com