Paper PD17

Beyond Code: Elevating NextGen Statistical Programmers through Cross-Functional Learnings

Saumilkumar Tripathi, Ephicacy Lifescience Analytics, Bengaluru, India

ABSTRACT

As clinical research landscape evolves, the role of statistical programmers is expanding beyond traditional coding. Insufficient understanding of cross-functional teams can hinder effective collaboration, resulting in delays, inaccurate results, and suboptimal outcomes.

This paper discusses about real-time case-studies on how cross-functional training in areas like clinical trial operations, data management, and regulatory affairs can help statistical programmers to transform their role into clinical data scientists. Programmers with knowledge of study phases, data collection process, and regulatory submission requirements (e.g., FDA/EMA) can better anticipate stakeholder needs and deliver reliable outcomes ensuring deliverables are both compliant and submission ready. Cross-functional skills also foster greater collaboration, increase data accuracy, reduce rework and accelerate timelines. Such foundational trainings avoid downstream data discrepancies and prevent major protocol deviation during analysis.

This knowledge not only enhances drug development efficiency but redefines the career trajectory of statistical programmers, transforming them into role of clinical data scientists.

INTRODUCTION

The role of statistical programmers in clinical research is undergoing a significant transformation. It is no longer confined to writing code behind the scenes. Today's programmers are increasingly recognized as strategic contributors to clinical development. As clinical trials become more complex with diverse data sources such as real-world evidence, wearables technologies, electronic health records and as regulatory expectations grow more stringent, the need for seamless cross-functional collaboration is very important. A limited understanding of cross-functional teams like clinical operations, data management, and regulatory affairs can hinder communication, delay timelines, and compromise data quality and compliance. For instance, misalignment between programming deliverables and regulatory requirements can lead to agency queries and rework, while insufficient awareness of data collection and cleaning processes may result in late-stage discrepancies or protocol deviations during statistical analysis.

To address these challenges, cross-functional training is becoming an important way to bridge gaps and improve collaboration. By expanding their expertise beyond programming into clinical operations, data management and regulatory domains, next-generation statistical programmers are better equipped to anticipate stakeholder needs, ensure data integrity, and deliver submission-ready outputs. This evolution marks the emergence of the "clinical data scientist", a professional who integrates technical acumen with scientific and operational insight to drive clinical trial success. This paper offers a practical, industry-focused exploration of this shift. It outlines the methodological approach, to demonstrate the value of cross-functional learning, and discusses the tangible benefits observed, including enhanced collaboration, improved data quality, regulatory compliance, and accelerated timelines. The paper concludes by examining the broader implications for both individual career growth and the efficiency of drug development programs.

Clinical Research Workflow

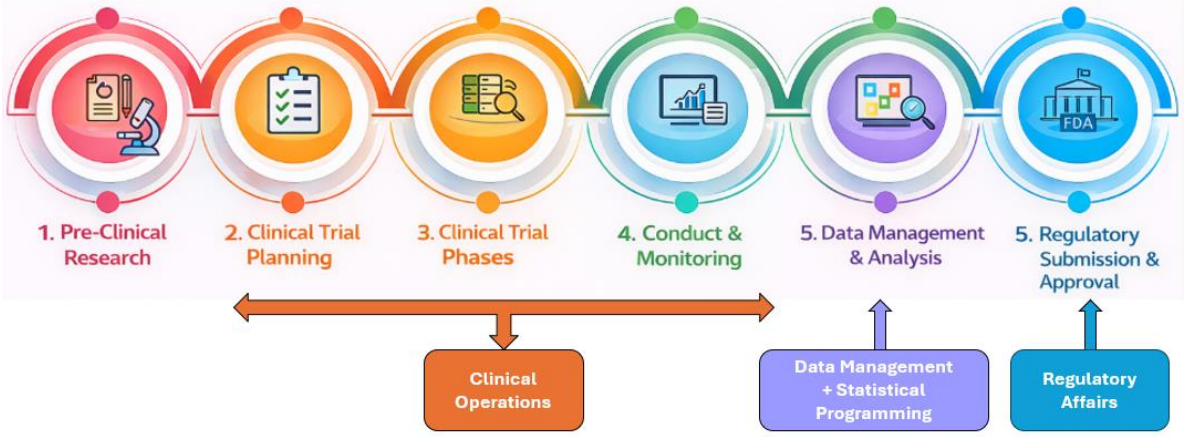

Figure 1

CASE STUDY 1 :

Embedding Statistical Programmers in Clinical Operations: Turning Early Insight into Better Outcomes

Background:
The clinical operations team plays a central role in the execution of clinical trials, overseeing activities such as protocol development, patient recruitment, and site monitoring. Traditionally, this function operates separately from statistical programming. However, decisions made during trial design and conduct such as endpoint definitions, visit schedules, and protocol amendments have a direct and often significant impact on data collection and downstream analysis. Misalignment or lack of early collaboration can lead to data inconsistencies, rework, or delays in analysis. This case study demonstrates how embedding statistical programmers into clinical operations through early protocol review and active participation in study team discussions can enhance data quality, reduce rework, and improve overall trial outcomes.

Approach:
As a part of a growing industry trend, many organizations are now integrating lead statistical programmers into cross-functional study teams from the earliest stages of trial design. In one such example, a pharmaceutical company assigned a programming lead to participate in protocol development meetings alongside clinicians and statisticians. To support this role, the programmer received targeted training in key aspects of clinical trial operations, including study phase (Phase I-IV), patient and data flow logistics, endpoint definitions, and Good Clinical Practice (GCP) principles. This foundational knowledge enabled the programmer to effectively engage in operational discussions, interpret protocol requirements through a data-centric lens, and ensure that data collection strategies aligned with downstream analytical needs.

Outcome:
The integration of statistical programmers into clinical operations from the early stages of trial design and execution yielded several measurable benefits. Following are the key outcomes.

- Statistical Programmer have identified and corrected a missing biomarker field in the CRF before study launch, preventing downstream data issues.
- Enabled proactive adjustments to dataset specifications during protocol amendments, maintaining timelines and reducing rework.
- Improved collaboration between programming and clinical operations, fostering mutual understanding of data dependencies and operational constraints.
- Detected a protocol deviation (out-of-window dosing) early and adjusted analysis accordingly, avoiding bias and re-analysis during CSR review.
- Strengthened data quality and regulatory compliance through early involvement in protocol and CRF review.
- Demonstrated the value of cross-functional training in enabling programmers to anticipate and respond to operational changes effectively.
- Reinforced the role of programmers as strategic partners in trial execution, not just as technical contributors.

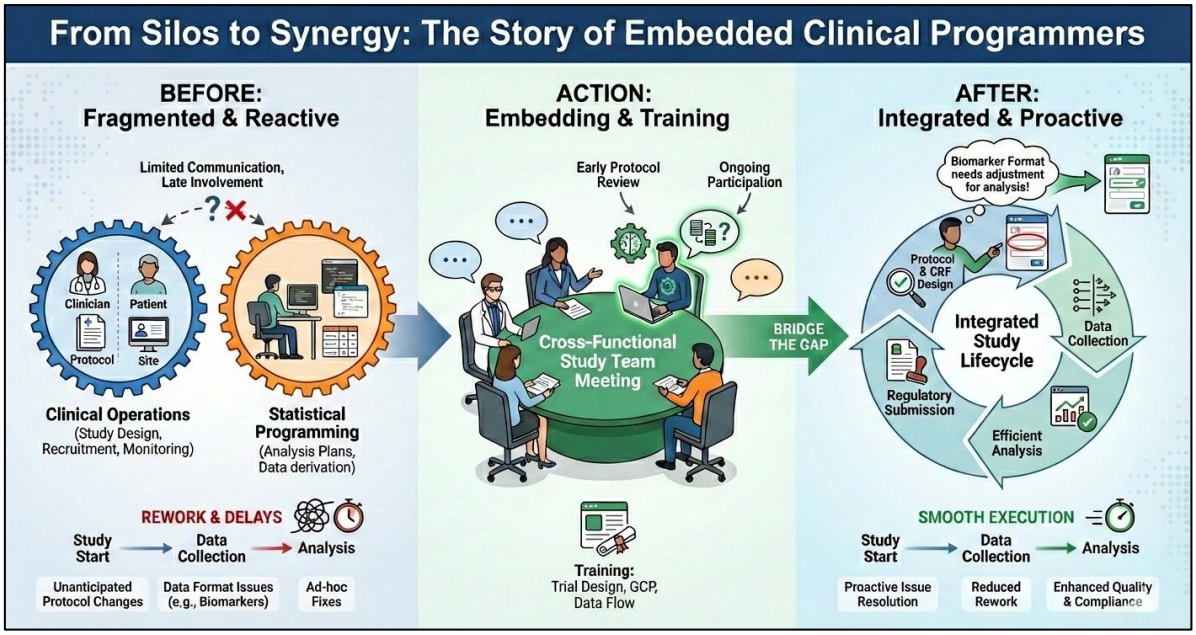

Figure 2*

CASE STUDY 2 :

Shift Left to Move Faster: Enabling Faster Statistical Analysis Through Data Management Insights

Background:
In a typical clinical trial, data management and statistical programming operate as sequential but interdependent functions: data managers design and configure EDC databases, implement CDASH-aligned eCRFs, execute edit checks, manage medical coding, and resolve queries to deliver a clean database at lock, while statistical programmers transform the locked data into CDISC-compliant SDTM and ADaM datasets and generate analysis outputs for regulatory submission. Despite this technical dependency, these functions have traditionally worked in silos, with limited collaboration prior to database lock. Programmers often lack visibility into upstream data capture, validation logic, and protocol-driven data handling decisions, while data managers may not fully understand how CRF design, unit standardization, controlled terminology, or query prioritization affect downstream derivations, analysis populations, and regulatory outputs. This disconnect frequently results in late discovery of data issues, extended query cycles, delayed database locks, Adhoc programming workarounds, and increased risk of non-compliance with CDISC and submission requirements, highlighting the need for earlier technical alignment and cross-functional understanding between data management and statistical programming.

Approach:
Upfront Alignment Meetings:

- Statistical programmers participated in study start up activities, including protocol and CRF reviews, to provide early analysis focused input. This ensured that key endpoints, derived variables, and analysis critical attributes were captured at data collection and that CDASH fields aligned with downstream SDTM and ADaM requirements, reducing post lock transformations and rework.

Data Review Metrics:

- The data management team implemented shared, real time data cleaning metrics informed by analysis priorities. These metrics highlighted critical data elements such as primary endpoints, key safety variables, and timing data allowing high impact queries to be identified and resolved early.

Regular Cross Functional Check Ins:

- Weekly data readiness meetings involving data management, statistical programming, and biostatisticians enabled joint review of query status, data trends, and analysis risks. These forums supported early identification of protocol deviations, coding issues, and data inconsistencies, reinforcing shared accountability for timely database lock.

Protocol Driven eCRF Controls:

- Implemented online edit checks during eCRF design aligned with protocol inclusion and exclusion criteria, enabling early identification of potential screen failures. This prevented major protocol deviations and reduced the need for retrospective population adjustments and rework during statistical analysis.

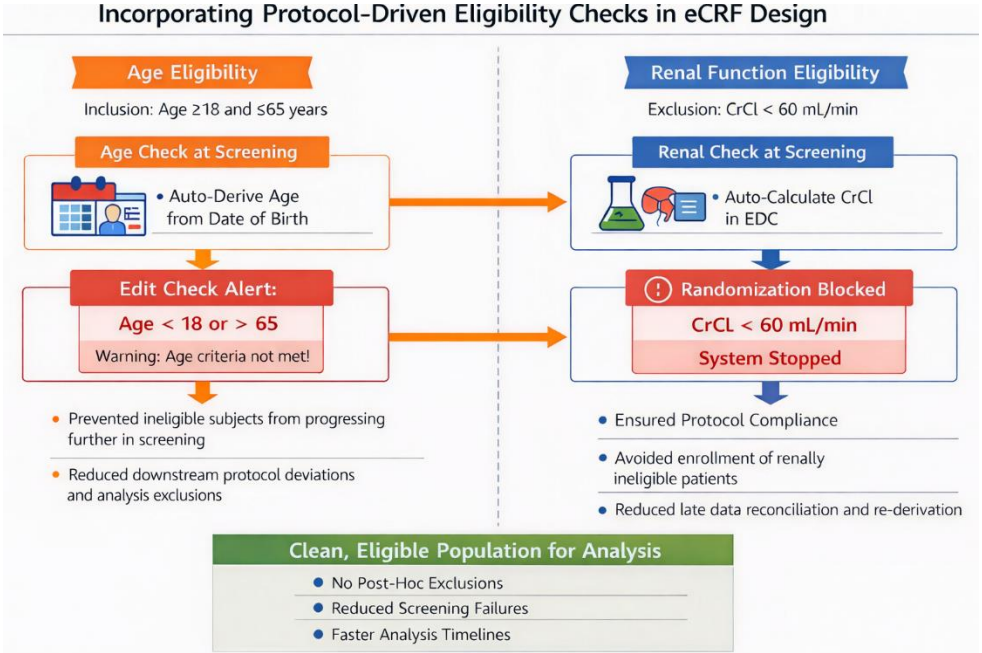

Figure 3*

Early Lab Data Standardization:

- In multi-center trials involving both local and central laboratories, different conventional units are often used for laboratory parameters. The Data Management team receives data from local laboratory vendors according to pre-determined data transfer guidelines, which is then provided to the statistical programming team along with CRF raw data extracted from the EDC system. Subsequently, the statistical programming team is responsible for converting results from conventional units into SI units for the creation of the SDTM.LB domain. This process requires the development of an intermediate unit conversion file containing the appropriate conversion factors for each conventional unit. A lack of prior knowledge about potential units can lead to delays in SDTM programming. In the example below, the statistical programmer anticipated this requirement and proactively converted external laboratory data to the required SI units prior to analysis, thereby ensuring alignment with CDISC standards. This approach reduced SDTM generation turnaround time, minimized programming rework, and eliminated repeated clarification cycles with local laboratory vendors.

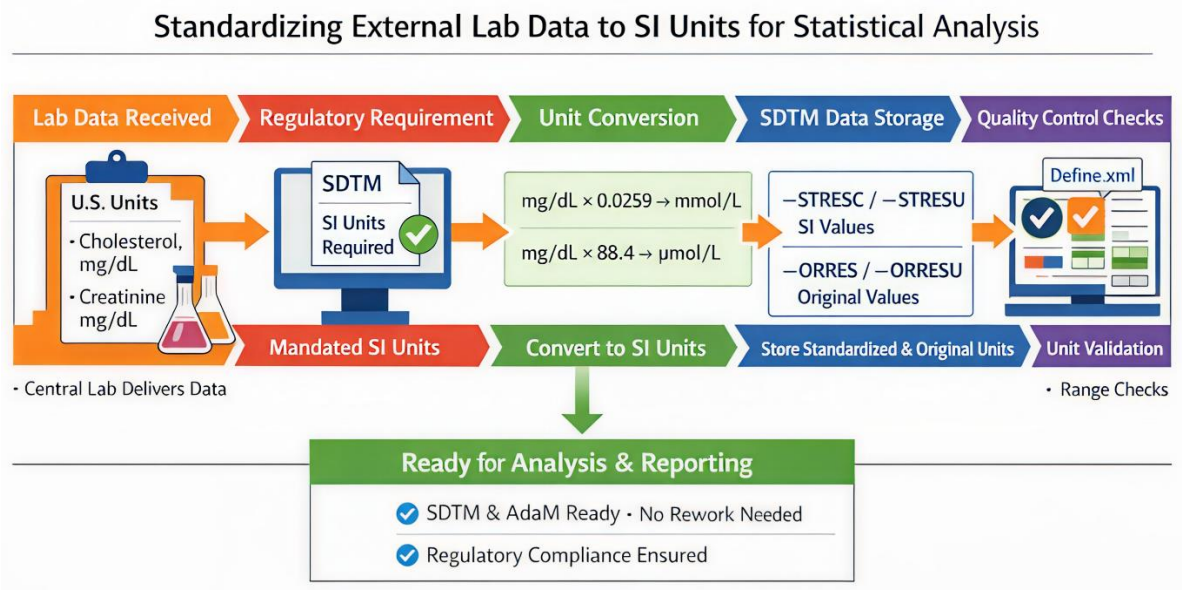

Figure 4*

Outcome: **:**
Cross-functional training between Data Management and Statistical Programming delivered measurable improvements in both efficiency and data quality.

Key Metric	Before Cross-Functional Training	After Cross-Functional Training
Outstanding Queries at Database Lock	High backlog of unresolved queries at time of DB lock (high number of open queries, risking lock delays)	Minimal queries remaining at lock (critical data issues identified and resolved earlier)
Query Turnaround Time (site query resolution speed)	Slow query resolution: separate teams and unclear priorities led to longer times to resolve data queries	Faster, prioritized query handling; cross-trained teams focused on quick resolution of critical queries, reducing average query turnaround time
Time from Last Patient Last Visit (LPLV) to Database Lock	Often prolonged (several weeks) due to extended cleaning and query reconciliation post-LPLV	Accelerated DB lock achieved on or ahead of schedule.
Post-Lock Programming Rework	Frequent post-lock data updates requiring re-programming or additional QC cycles (e.g., unplanned derivations, multiple reruns of analysis datasets and outputs)	Little to no rework needed after lock. Clean data and statistical programmers could focus on analysis, saving ~1-2 weeks on analysis timelines

CASE STUDY 3 :

Building a Submission-Ready Mindset: Cross-Functional Regulatory Training for Statistical Programmers

Background:
Regulatory affairs team plays a critical role in ensuring that clinical trial data meets the stringent requirements of global health authorities. Errors or misalignment in this domain such as non-compliance with CDISC standards or insufficient documentation of trial results in CSR or Reviewer's guide can lead to regulatory queries, submission delays, or even rejection of marketing applications. Traditionally, regulatory affairs professionals have been responsible for compiling submission packages, while statistical programmers have focused primarily on generating analysis datasets, outputs, define packages. However, as agencies like the FDA and EMEA increasingly emphasize data traceability, metadata integrity, and standardization, it has become essential for statistical programmers to develop a deeper understanding of regulatory expectations. Cross-functional training in regulatory affairs equips programmers with knowledge of submission guidelines, documentation standards (e.g., define.xml, reviewer's guides), eCTD package requirements and common regulatory review practices, fostering a mindset that prioritizes compliance, audit readiness, and regulatory reviewer's needs from the outset of programming activities.

Approach:
A structured regulatory training program was delivered by the Regulatory Affairs (RA) team to statistical programmers during preparation for a late-stage FDA submission. The training combined regulatory guidance review with hands-on examples drawn from real submission artifacts.

1. File Format and Transport Requirements

RA team educated programmers on FDA-mandated submission formats, emphasizing that SDTM and ADaM datasets must be provided as SAS® Version 5 XPT files, not compressed and free of custom formats. A real submission example was reviewed where datasets created using an incorrect transport method triggered validation errors. Programmers subsequently updated their dataset creation macros to enforce compliant XPT generation by default.

2. Dataset Size and Splitting Rules

Using examples from large Phase III studies, RA team explained FDA dataset size limits (e.g., maximum dataset size thresholds) and acceptable splitting strategies. Programmers were shown how oversized adverse event datasets had previously required urgent splitting late in submission. Post-training, programmers proactively assessed dataset sizes during SDTM creation and implemented standardized split logic (with proper sequence variables and documentation in the SDRG), avoiding last-minute remediation.

3. Traceability from Analysis Results to Source Data

RA team walked programmers through how FDA reviewers navigate submissions, starting from key results, moving to ADaM, then SDTM, and finally to annotated CRFs. Using an efficacy endpoint example, programmers were required to demonstrate traceability from a CSR table value back to ADaM variables, SDTM domains, and CRF annotations. This exercise reinforced the importance of clearly documented derivations in define.xml and ADRG, leading programmers to enhance variable origin descriptions and derivation algorithms.

4. eCTD Module 5 Folder Structure and Content

RA team provided a detailed overview of eCTD Module 5, focusing on the placement and purpose of study data tabulations, analysis datasets, define.xml, reviewer's guides, and annotated CRFs. Programmers reviewed a prior submission where misplacement of define.xml and reviewer's guides resulted in additional validation cycles. As a result, programmers adopted a checklist-based approach to ensure all required components were present, correctly named, and placed in the appropriate Module 5 subfolders.

5. Reviewer's Guides and Metadata Quality

RA team demonstrated how reviewers rely on SDRG and ADRG documents to understand study-specific decisions, deviations from standards, and known data limitations. Programmers practiced drafting reviewer-friendly explanations for complex derivations and protocol deviations in analysis reviewer's guide. This shifted documentation from a compliance afterthought to an integral part of programming deliverables.

Outcome :

- Regulatory training resulted in a clear improvement in submission readiness and overall efficiency.
- Statistical programmers consistently delivered SDTM and ADAM datasets that complied with required file formats and size limits, leading to fewer technical validation findings.
- End-to-end traceability between analysis results, datasets, define.xml, and annotated CRF was strengthened, enabling regulatory reviewers to navigate the data with minimal clarification requests.
- Regulatory affairs reported a reduction in late-stage issues related to dataset splitting, metadata inconsistencies, and missing documentation.
- During FDA review, an unplanned subgroup analysis request was addressed rapidly due to well-structured and well-documented datasets that supported flexible re-analysis.
- The submission passed technical validation without rejection-criteria findings and progressed through regulatory review faster than comparable historical submissions.

CROSS-CASE INSIGHTS AND BEST PRACTICES

The case studies demonstrate that statistical programmers are increasingly serving as proactive collaborators through enhanced cross-functional integration. Early engagement with diverse teams enables programmers to expand their contributions beyond traditional responsibilities, playing a strategic role in trial execution and regulatory outcomes. The following table summarizes these findings and presents best practices for promoting cross-functional learnings, enhancing data quality and compliance, and positioning programmers as essential clinical data scientists.

Insight Area	Key Observations	Impact on Statistical Programming	Best Practice
Early Cross-Functional Integration	Early exposure to upstream and downstream processes improved context and decision-making.	Reduced late-stage rework and better stakeholder alignment.	Engage programmers early in protocol, data strategy, and submission planning.
Proactive Programming Mindset	Programmers anticipated issues instead of responding late.	Faster timelines and fewer corrective cycles.	Position programmers as proactive contributors.
Shared Data Quality Ownership	Cross-functional understanding improved accountability.	Improved consistency, traceability, and interpretability.	Promote shared responsibility for data quality.
Traceability & Compliance Focus	Regulatory and operational literacy strengthened traceability.	Smoother regulatory reviews and stronger documentation.	Embed traceability principles into standards and training.
Experiential Learning Models	Hands-on learning proved more effective than passive training.	Sustained behavioral change.	Use rotations, shadowing, and collaborative reviews.
Standardized Practices	Recurring issues addressed earlier through shared insights.	Predictable and consistent delivery outcomes.	Align training with real study deliverables.
Career Development Impact	Programmers evolved beyond coding roles.	Greater strategic contribution to trials.	Formalize cross-functional learning in career progression.

CONCLUSION

While emerging capabilities such as Generative AI, large language models (LLMs), and automation are increasingly influential in shaping the future of statistical programming, their effective and responsible use depends on a strong foundational understanding of clinical and regulatory context. Automation can accelerate execution and AI can enhance efficiency, but without cross-functional insight into trial conduct, data standards, and regulatory expectations, these technologies risk amplifying errors rather than mitigating them. Cross-functional learning therefore provides the essential groundwork upon which advanced technologies can be successfully leveraged.

In summary, the case studies presented in this paper demonstrate that cross-functional learning is not an optional enhancement but a foundational requirement for the evolution of statistical programming. As clinical trials grow in complexity and regulatory expectations continue to intensify, organizations must move beyond viewing programmers solely as technical executors. Embedding structured exposure to clinical operations, data management, and regulatory affairs into programmer development pathways enables a deeper understanding of the end-to-end clinical data

lifecycle. This foundation empowers statistical programmers to deliver compliant, traceable, and submission-ready outputs while effectively leveraging emerging technologies such as automation and advanced analytics. Ultimately, elevating statistical programmers beyond code and transforms them into clinical data scientists who contribute strategically to trial execution, regulatory success, and data-driven decision-making. By investing in cross-functional learning, the industry strengthens both the quality of clinical evidence and its ability to meet the evolving demands of modern drug development.

REFERENCES

Clinical Data Interchange Standards Consortium - https://www.cdisc.org/

The Specification for File Format Types Using eCTD Specifications.
https://www.fda.gov/drugs/electronic-regulatory-submission-and-review/electronic-common-technical-document-ectd

* Figures 2,3 and 4 are generated using AI platform based on prompts using content from respective case study.
Microsoft. (2026). Microsoft Copilot (GPT-5 chat model) [Large language model].
https://copilot.microsoft.com/

ACKNOWLEDGMENTS

I would like to thank all my colleagues at Ephicacy who shared their experiences, read my drafts, and gave me valuable and constructive feedback. Thank you for your support.

CONTACT INFORMATION

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

Author Name: Saumilkumar Tripathi
Company: Ephicacy Lifescience Analytics
Address: No. 6, 2nd Floor,
2nd Main, Arekere,
Off Bannerghatta Road,
Bengaluru - 560076,
Karnataka, India.
Work Phone: +91-8390914975
Email: saumil.tripathi@ephicacy.com
Website: https://www.ephicacy.com/