Sarcouncil Journal of Internal Medicine and Public Health

ISSN(Online): 2945-3674

Volume- 05| Issue- 05| 2026

Research Article **Received:** 05-08-2026 | **Accepted:** 30-08-2026 | **Published:** 14-09-2026

Beyond the Dataset: A Practice Framework for the Statistical Programmer's Role in Safeguarding FDA Submission Integrity

Radhika Aitha

Independent Researcher, USA.

Abstract: FDA regulatory submissions depend on statistical programming deliverables that reach well beyond dataset production tables, listings, and figures; machine-readable metadata; dual-programmer validation architecture; and an emerging generation of submission formats. This synthesis proposes a four-part framework describing the programmer's integrity-safeguarding role across these layers: TLF production and traceability, the metadata stack (define.xml and the newer Analysis Results Standard), quality-control and validation architecture (dual programming, Pinnacle 21 conformance checking), and emerging-format readiness (Dataset-JSON, R-based submissions). Drawing on FDA technical rejection criteria, CDISC standards documentation, and current practice literature, the analysis finds that submission integrity rests on interlocking layers, where a defect in one an inconsistent define.xml tag, an unresolved Pinnacle 21 warning propagates directly into technical rejection risk no matter how sound the underlying clinical analysis is. FDA's own technical rejection data show that 32 percent of submissions with study data had critical conformance issues, a failure point that arises before a reviewer ever reaches the clinical content. The framework gives programming teams a structured way to allocate QC attention across these four layers, rather than treating submission readiness as one undifferentiated checklist item.

Keywords: FDA submission, statistical programming, CDISC standards, data validation, regulatory compliance, Pinnacle 21.

INTRODUCTION

FDA has required CDISC standards for NDA, BLA, and ANDA submissions involving studies initiated after December 17, 2014, with binding compliance in force since December 17, 2016 (U.S. Food and Drug Administration, 2024; Applied Clinical Trials Online, 2024). The Study Data Technical Conformance Guide turns that mandate into specific, checkable expectations for how SDTM, ADaM, and Define-XML deliverables must be structured (U.S. Food and Drug Administration, 2025). Submission readiness tends to get discussed as a regulatory-affairs or medical-writing concern the eCTD backbone, the clinical study report narrative while the technical programming layers underneath receive comparatively little structured attention in the literature, despite carrying direct and immediate rejection risk. FDA's own Technical Rejection Criteria data show that 32 percent of submissions with study data had critical conformance issues, a failure point addressable at the programming level, well before a reviewer ever reaches the clinical content (Quanticate, 2024).

This article proposes a four-part framework describing where and how the statistical programmer safeguards submission integrity: TLF production and traceability, the metadata stack, quality-control and validation architecture, and emerging-format readiness. Its central claim is that these four layers interlock rather than operate independently; a defect introduced in one does not stay contained to it, and understanding these interlocks has direct value for how programming teams set their own QC priorities.

The stakes of getting this framing right extend beyond any single submission cycle. A technical rejection does not simply delay a review clock; it returns a submission to the sponsor before the FDA electronic document rooms where substantive review takes place ever receive it, which means the clinical, medical-writing, and regulatory-affairs effort invested in that submission accomplishes nothing until the underlying technical defect is identified and corrected. Understanding which of the four layers is most likely to produce this outcome, and why, is therefore not a narrow technical concern confined to the programming function; it has direct consequences for program timelines that regulatory strategy and portfolio planning depend on.

MATERIALS AND METHODS

This article synthesizes FDA technical guidance, CDISC standards documentation, and peer-reviewed and industry practice literature describing statistical programming's role in FDA submission integrity. Sources were located through targeted search of FDA guidance repositories, CDISC's standards library, PHUSE deliverables, and R Consortium publications, with priority given to current guidance and recently reported pilot outcomes over superseded material. Eighteen sources met relevance and currency criteria and

***Corresponding Author:** Radhika Aitha

DOI- https://doi.org/10.5281/zenodo.22756626

were organized against the four-part framework above: TLF production and traceability; the metadata stack, including define.xml and the Analysis Results Standard; quality-control and validation architecture, including dual programming, Pinnacle 21, and 21 CFR Part 11; and emerging-format readiness, including Dataset-JSON and R-based submissions.

Each layer was assessed for its characteristic failure points as documented in FDA technical rejection data and CDISC standards literature, the detection mechanism currently used to catch defects at that layer before or during submission, and the regulatory consequence that follows when a defect goes undetected. Interlock relationships between layers where a defect originating in one layer produces its consequence in another were identified by tracing, through the guidance and practice literature, how a specific failure type documented at one layer (for example, a define.xml tagging error) is described as producing a specific downstream outcome (a technical rejection triggered during automated conformance checking, before any human reviewer engages with the submission).

This is a literature-synthesis and practice-framework article. It does not involve original patient data, human subjects research, or animal research, so an ethics committee approval statement does not apply here. No primary or client-derived dataset was analyzed; every figure cited traces back to the reference literature at its point of use, and the ordinal comparisons presented in the accompanying table and figures reflect a qualitative synthesis of that literature rather than an independently validated measurement instrument.

RESULTS

TLF Production and Traceability

Tables, listings, and figures are the outputs FDA statisticians and medical officers review directly, which makes their traceability back to locked ADaM datasets and final programs a non-negotiable expectation rather than a best practice (U.S. Food and Drug Administration, 2025). A discrepancy between a primary efficacy table and the integrated summary text is treated as a review flag in its own right, independent of whether the underlying number is correct; the mismatch itself signals a process gap reviewers have learned to distrust. This standard applies with particular force to Module 2.7 clinical summaries and the integrated summaries of safety and efficacy they contain, where pooled datasets drawn from multiple studies raise the reconciliation burden considerably beyond what any single-study TLF requires.

Reproducibility is the second demand this layer places on the programmer, and it is a stricter standard than it might first appear. Every TLF submitted must be regenerable from the final, locked ADaM datasets using the final, locked programs not from an interim dataset version, and not from a program that was subsequently modified for a different output. This requirement means that version control discipline is not a peripheral good practice for TLF programming; it is the mechanism by which reproducibility claims can actually be verified if a reviewer asks a sponsor to demonstrate that a given table traces cleanly back to its stated inputs. Programmers who treat their working files casually during the iterative shell-development phase of a study, only tightening version discipline once outputs are believed final, risk discovering late in the process that the version actually used to generate a submitted table cannot be cleanly reconstructed.

Denominator consistency across related tables is a further, easily underestimated source of TLF risk. An adverse-event summary table and a corresponding by-treatment-arm breakdown must use identical, clearly footnoted denominator definitions whether the denominator is all randomized subjects, all treated subjects, or some other analysis population or the two tables will appear to disagree even when each is internally correct. This is precisely the kind of defect that a reviewer encounters as a credibility problem rather than a technical one: two tables that do not reconcile suggest, rightly or wrongly, that the underlying analysis was not conducted with the rigor the submission claims.

The Metadata Stack

Define.xml errors sit among the most common technical rejection reasons FDA reports, with specific error codes missing ts.xpt files, incorrect standardized-dataset file tags accounting for a disproportionate share of Commercial IND rejections (Clinical Standards Hub, 2024). FDA's technical rejection criteria formalize this: automated checks confirm that anchor files such as dm.xpt or adsl.xpt are present, since adsl.xpt in particular serves as the parent dataset for all downstream ADaM analysis, and its absence or mistagging halts the submission before any

Publisher: SARC Publisher

reviewer engages with the underlying science (Clinical Standards Hub, 2024; Onenpg, 2024).

The newer Analysis Results Standard extends this metadata layer further, binding ADaM records to the actual reported statistics so reviewers can query results programmatically rather than relying solely on TLF output (CDISC, 2024; Prometrika, 2024). This is a meaningful shift: it means metadata quality now has downstream consequences for how efficiently a reviewer can verify a submission's numbers, not just whether the submission validates cleanly on receipt. A programmer who treats ARM documentation as an afterthought to TLF production is, in effect, under-investing in exactly the artifact that determines how smoothly the statistical review phase proceeds once technical conformance has already been cleared.

Documenting analysis method, program reference, and result values for each reported statistic under the Analysis Results Standard is a materially different task from writing a define.xml variable label, and it asks something new of the programmer's workflow rather than simply adding a documentation step at the end. Where define.xml has traditionally described what a variable means, ARM-style metadata must describe how a specific reported number was produced and where in the codebase that derivation lives a linkage that is straightforward to maintain if it is built into the programming workflow from the outset, and considerably harder to reconstruct retroactively once a study has been running for months and multiple programmers have touched the same output.

Traceability failures at this layer rarely announce themselves as a single dramatic error. More often they surface as a slow accumulation of small inconsistencies: a variable renamed in one dataset iteration but not another, a derivation documented in the statistical analysis plan that drifted slightly from what the final program actually implements, a codelist value that was valid under an earlier controlled-terminology version but has since been superseded. Each of these, taken alone, might not trigger an automated rejection. Collectively, they are exactly the kind of pattern that erodes a reviewer's confidence in a submission's overall data quality, even when no single finding rises to the level of a formal technical rejection.

Quality-Control and Validation Architecture

Dual programming remains close to universal practice for primary efficacy and safety outputs despite not being a formal regulatory requirement, and it earns that status: documented error-detection rates of 92 to 98 percent come at a cost of 1.6 to 2.0 times the primary programming effort (PHUSE, 2025). This cost has periodically prompted debate over whether the practice is worth its overhead relative to leaner, single-programmer QC models supplemented by automated checks, though the prevailing industry position remains that dual programming's detection-rate advantage justifies the effort for primary efficacy and safety outputs specifically, even where lighter validation models are accepted for supporting or exploratory analyses (Clinical Tech Leader, 2024). This effort is not spent uniformly; teams that plan quality-control strategy before programming begins, rather than treating validation as a downstream cleanup task, are better positioned to absorb this cost without it becoming a bottleneck late in a submission timeline (Lexjansen, 2012).

Pinnacle 21 functions as the industry's shared validation language the same tool FDA itself uses to check submitted data, which means a sponsor that skips pre-validation is effectively submitting blind to a check the agency will run regardless (Certara, 2024; Pinnacle 21, 2024). The tool's findings are stratified by severity for a reason that maps directly onto submission risk: ERROR-level findings, which include missing required variables and invalid codelist values, must be resolved before submission, while WARNING-level findings can be documented with justification in the Study Data Reviewer's Guide rather than corrected outright, and INFO-level findings carry no resolution obligation at all. A programming team that treats every Pinnacle 21 finding as equally urgent wastes effort on low-risk INFO items while a genuinely disqualifying ERROR sits unresolved; a team that under-triages severity risks submitting with unresolved ERROR findings that all but guarantee a technical rejection.

Underneath both practices sits 21 CFR Part 11, which requires secure, time-stamped audit trails for any system used to create, modify, or delete the electronic records these programs and datasets constitute (SimplerQMS, 2024) a requirement that shapes which platforms and workflows are even permissible for this work, not just how carefully the work is checked afterward. Controlled programming environments such as SAS Life

Publisher: SARC Publisher

Sciences Analytics Framework satisfy this requirement through built-in role-based access control and version management, but only if programmers actually work within the environment's production area rather than treating its controls as an afterthought applied just before submission. A program developed and validated outside the controlled environment, then copied in at the last moment to satisfy an audit-trail requirement, does not carry a genuine record of its own development history which defeats the purpose Part 11 compliance is meant to serve, even though the final file may appear, superficially, to meet the letter of the requirement.

Emerging-Format Readiness

A 2023-2024 CDISC-PHUSE-FDA pilot demonstrated that Dataset-JSON can serve as a direct substitute for the decades-old XPT transport format, eliminating XPT's eight-character variable-name limit and its practical size constraints; FDA followed with an April 2025 Federal Register request for comment on adopting Dataset-JSON as XPT's long-term replacement (Clinical Leader, 2024; Federal Register, 2025). The practical case for this transition is not merely theoretical: pilot testing found that a single SDTM laboratory dataset shrank from roughly 2,700 kilobytes under XPT to roughly 640 kilobytes under Dataset-JSON, a compression ratio that matters directly for large, pooled ADaM datasets approaching the 5-gigabyte size ceiling that XPT enforces and that has historically forced sponsors to split otherwise-coherent datasets purely to satisfy a transport-format limitation.

In parallel, R-based submissions have moved from experimental to demonstrably reviewable: the R Consortium's Pilot 3 package was successfully reviewed by FDA in 2024, and a subsequent test submission incorporating a WebAssembly component was successfully received through FDA's eCTD gateway later that year, extending the precedent beyond a single proof-of-concept (R Consortium, 2024a; R Consortium, 2024b). The open-source admiral package now lets programmers build CDISC-compliant ADaM datasets in R rather than SAS, developed collaboratively across multiple sponsor organizations rather than as a single company's proprietary tool, which matters for long-term maintainability and cross-sponsor consistency in a way a closed, single-vendor alternative would not (Clinical Standards Hub, 2024b).

Neither transition is currently mandatory, and that distinction matters for how a programming team should prioritize adopting either one. Dataset-JSON remains at the request-for-comment stage rather than a binding requirement, and R-based submissions, while demonstrably reviewable, still represent a minority practice relative to SAS-based submission packages industry-wide. The readiness this domain calls for is therefore anticipatory rather than reactive: teams that begin building fluency with JSON schema validation, the CDISC Dataset-JSON specification, and R-based ADaM tooling now are positioned to adopt either format smoothly whenever a firmer regulatory expectation does emerge, rather than needing to build that capability under deadline pressure once a mandate is announced.

Table 1: Comparative failure points, detection mechanisms, and regulatory consequences across the four programming layers of FDA submission integrity (author's synthesis of cited sources).

Layer	Common Failure Point	Detection Mechanism	Regulatory Consequence
1. TLF Production	Discrepancy with SAP/shell specs or ISS/ISE text	Manual review, cross-check against locked ADaM	Reviewer distrust flag, even if underlying number is correct
2. Metadata Stack	Define.xml tagging errors, missing ts.xpt/anchor files	FDA Technical Rejection Criteria (automated)	Technical rejection before clinical review begins
3. QC/Validation	Unresolved Pinnacle 21 ERROR/WARNING findings	Dual programming, Pinnacle 21 conformance check	Submission-gateway rejection or reviewer-flagged non-conformance
4. Emerging Formats	Unfamiliarity with Dataset-JSON/R-based eCTD conventions	Pilot-program precedent, R Consortium review outcomes	Currently low - readiness gap, not yet a compliance failure

Publisher: SARC Publisher

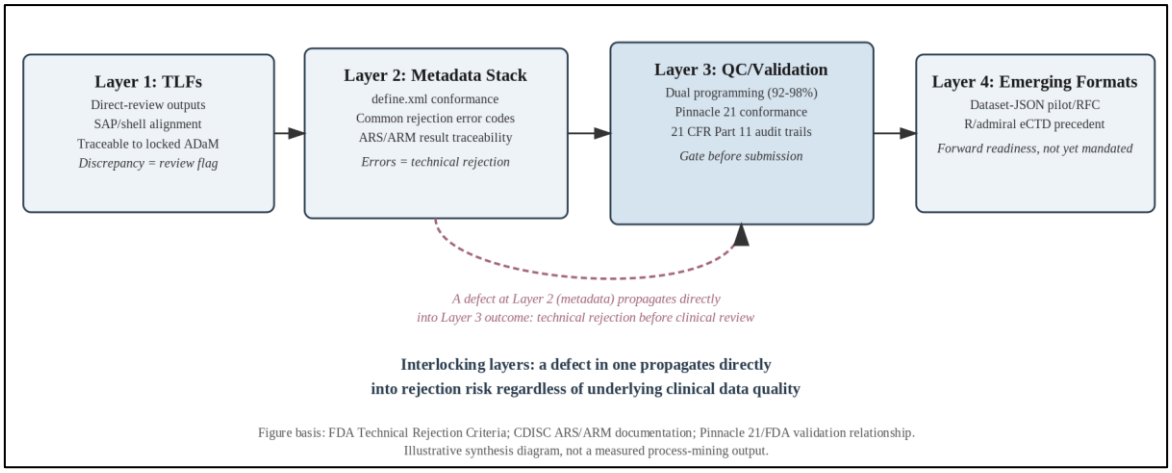

Figure 1: Interlocking layers of FDA submission programming integrity a defect at one layer (e.g., metadata) propagates directly into downstream consequence (e.g., technical rejection) regardless of underlying clinical data quality.

DISCUSSION

These four layers do not sit side by side as independent checklist items; they interlock. A metadata error is not merely a documentation problem, it is what turns an otherwise valid dataset into a technical rejection, because FDA's own validation tooling checks metadata conformance before a reviewer ever assesses clinical content. Treating TLF accuracy, metadata quality, dual-programming validation, and format readiness as four separate boxes to check, rather than four points where the same underlying submission can fail, understates how concentrated rejection risk actually is at these technical junctures.

The practical value of naming these interlocks explicitly is that it changes how a programming team allocates its own internal review time. A team that inspects each layer in isolation confirming TLFs match the SAP, confirming define.xml validates, confirming dual-programming discrepancies are resolved can complete every individual check successfully and still miss a defect that only becomes visible when two layers are examined together. A define.xml that validates cleanly in isolation but assigns a codelist value inconsistent with what the corresponding TLF footnote states is exactly this kind of cross-layer defect: neither the metadata check nor the TLF review, conducted separately, is designed to catch it, because each was built to verify its own layer's internal consistency rather than consistency across layers.

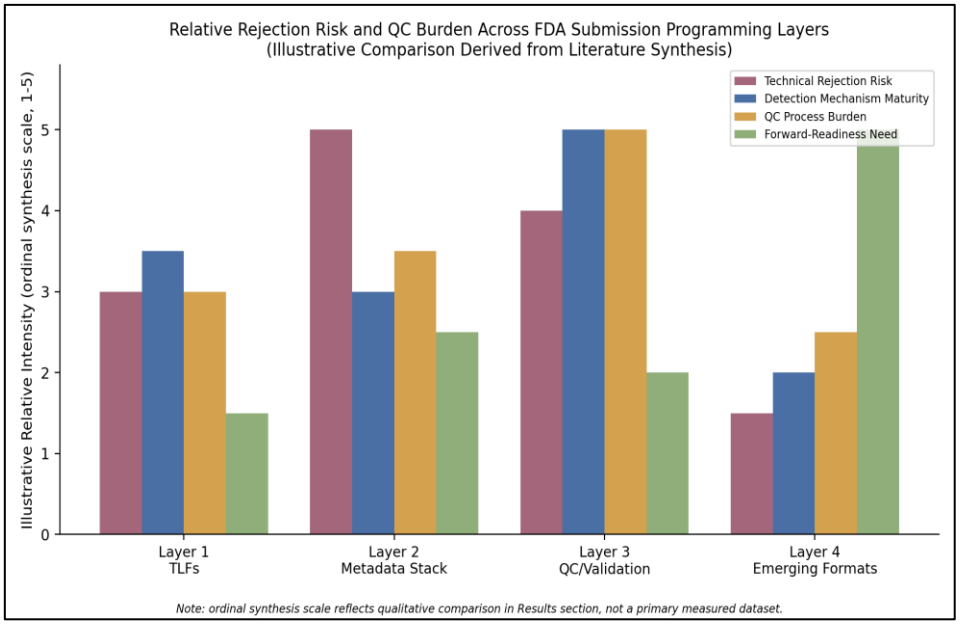

Figure 2: Relative rejection risk and QC burden across FDA submission programming layers illustrative ordinal synthesis derived from the literature comparison in Results, not a primary measured dataset.

Publisher: SARC Publisher

The SAS-to-R transition compounds this picture rather than simplifying it. Teams adopting R for FDA submissions now need QC and validation fluency across two language ecosystems at once, not a clean replacement of one with the other successful FDA review of R-based pilot packages proves the pathway works, but it does not remove the burden of maintaining dual-language validation competence during the transition (R Consortium, 2024a). Dataset-JSON and ARS/ARM readiness represent a similar forward-looking investment: programming teams that build this fluency now, ahead of a plausible eventual mandate shift, are better positioned than teams that wait for a compliance deadline to force the issue.

Table 2: Layer-specific QC investment priorities for statistical programming teams supporting FDA submissions.

Layer	Current QC Maturity	Recommended Investment Priority
1. TLF Production	High - well-established manual/automated cross-check practice	Maintain; automate SAP/shell-to-output cross-checks where possible
2. Metadata Stack	Moderate - TRC catches errors but only after submission	High - pre-submission define.xml/ARS validation before gateway submission
3. QC/Validation	High - dual programming and Pinnacle 21 are mature practices	Maintain; ensure Pinnacle 21 findings are documented, not just resolved
4. Emerging Formats	Low - Dataset-JSON and R/admiral are pilot-stage for most teams	Build now - forward-readiness investment ahead of plausible mandate shift

Limitations are worth stating directly. This article synthesizes existing guidance and practice literature rather than presenting original empirical measurement of rejection rates or validation outcomes at a specific organization. The interlock relationships described in Table 1, Table 2, Figure 1, and Figure 2 are a structured synthesis of the literature reviewed, not a measured causal model, and should be read accordingly. The 32 percent conformance-failure figure cited in the Introduction reflects a specific FDA analysis reported through an industry secondary source rather than a primary FDA publication directly reviewed here, and should be treated as an approximate order of magnitude rather than a precise, currently reported statistic.

A further limitation concerns how this framework generalizes across submission types and sponsor organizations. The four layers described here were derived from guidance and practice literature that applies broadly across NDA, BLA, and ANDA submissions, but the relative weight each layer carries within a specific submission varies by program. A submission built substantially on pooled data from multiple legacy studies will place unusually heavy demand on the metadata-stack layer, where reconciling variable definitions and codelist versions across studies conducted years apart is considerably harder than for a single, recently designed trial. A first-in-class submission with limited precedent for its endpoints may place correspondingly heavier demand on the TLF-traceability layer, where reviewers have fewer established expectations to check the submission against and therefore scrutinize internal consistency more closely. Programming teams applying this framework to their own submission planning should calibrate which layer merits the heaviest QC investment based on their program's specific characteristics, rather than assuming the four layers carry equal weight in every case.

This framework was also constructed from published guidance and practice literature rather than from direct observation of programming teams preparing actual submissions, which means the interlock claims presented here reflect an informed reading of the technical and regulatory literature rather than a systematically observed account of how defects at one layer are discovered to have originated at another in practice. A study designed to trace actual technical rejections back through a submission's programming history, layer by layer, would offer a valuable empirical test of whether the interlock pattern described here matches what teams experience when a rejection actually occurs.

CONCLUSION

Statistical programming's integrity-safeguarding role in FDA submissions spans four interlocking layers TLF production, the metadata stack, validation architecture, and emerging-format readiness and treating them as a single undifferentiated QC checklist item understates exactly where rejection risk concentrates and why it spreads once it appears. Programming teams that map their own QC investment against these four layers, rather than against a single generic

Publisher: SARC Publisher

submission-readiness checklist, are better positioned to catch the cross-layer defects that isolated, layer-by-layer review is structurally unable to see.

CRediT Author Statement

Radhika Aitha: Conceptualization, Methodology, Investigation, Writing – Original Draft, Writing – Review & Editing, Visualization.

REFERENCES

1. Nelson, B. "FDA binding guidance: a pivotal milestone for CDISC standards." *Applied Clinical Trials* 25.12 (2016): 42.
2. Certara. "Clinical Data Validation." *Pinnacle 21 by Certara* (2024). https://www.certara.com/pinnacle-21-enterprise-software/validation/
3. Clinical Leader. "No More XPT? Piloting New Dataset-JSON for FDA Submissions." *Clinical Leader* (2024). https://www.clinicalleader.com/doc/no-more-xpt-piloting-new-dataset-json-for-fda-submissions-0001
4. Clinical Standards Hub. "FDA Technical Rejection Criteria for Study Data: A Comprehensive Deep Dive." *Clinical Standards Hub* (2024a). https://www.clinstandards.org/blog/fda-technical-rejection-criteria-for-study-data-a-comprehensive-deep-dive-1771251343829
5. Clinical Standards Hub. "{admiral}: A Deep Dive into Open-Source ADaM Dataset Generation in R." *Clinical Standards Hub* (2024b). https://www.clinstandards.org/blog/admiral-a-deep-dive-into-open-source-adam-dataset-generation-in-r-1771562213140
6. Clinical Tech Leader. "Is Double Programming Really Required for Validation?" *Clinical Tech Leader.* (2024). https://www.clinicaltechleader.com/doc/is-double-programming-really-required-for-validation-0001
7. CDISC. "Getting Started with the New CDISC Analysis Results Standard." *CDISC EU Interchange* (2024).
8. Federal Register. "Electronic Study Data Submission; Data Standards; Clinical Data Interchange Standards Consortium Dataset-JavaScript Object Notation; Request for Comments." (2025). https://www.federalregister.gov/documents/2025/04/09/2025-06051/
9. Lexjansen. "Quality Assurance: Best Practices in Clinical SAS Programming." *NESUG* (2012).
10. Onenpg. "Avoiding Technical Rejections in Regulatory Submissions." *OneNPG Industry Insights* (2024). https://www.onenpg.com/industry-insights/technical-rejection-of-a-regulatory-submission/
11. PHUSE. "Best Practices for Quality Control and Validation." *PHUSE Deliverables* (2025).
12. Pinnacle 21. "FDA Publishes Official Validation Rules for Submission Data." *Pinnacle 21 News* (2024). https://www.pinnacle21.com/news/fda-publishes-official-validation-rules-submission-data
13. Prometrika. "Meeting Regulator's Analysis Needs: CDISC's Analysis Results Standard." *Prometrika Thought Leadership* (2024).
14. Quanticate. "What is the FDA's Technical Rejection Criteria for Study Data?" *Quanticate Blog* (2024).
15. R Consortium. "News from R Submissions Working Group – Pilot 3 Successfully Reviewed by FDA." *R Consortium* (2024a).
16. R Consortium. "Using R to Submit Research to the FDA: Pilot 4 Successfully Submitted to FDA Center for Drug Evaluation and Research." *R Consortium* (2024b).
17. SimplerQMS. "FDA 21 CFR Part 11 Audit Trail: Definition, Requirements, and Compliance." *SimplerQMS* (2024). https://simplerqms.com/21-cfr-part-11-audit-trail/
18. U.S. Food and Drug Administration. "Study Data Technical Conformance Guide Technical Specifications Document." *FDA* (2025). https://www.fda.gov/media/122913/download.

Source of support: Nil; **Conflict of interest:** Nil.

Cite this article as:

Aitha, R. "Beyond the Dataset: A Practice Framework for the Statistical Programmer's Role in Safeguarding FDA Submission Integrity." *Sarcouncil Journal of Internal Medicine and Public Health* 5.5 (2026): pp 1-7.

Publisher: SARC Publisher