tidyverse, tidymodels, r-lib, and gt R packages: Regulatory Compliance and Validation Issues

A Guidance Document for the use of affiliated R packages in Regulated Clinical Trial Environments

September 2020

RStudio PBC
250 Northern Ave
Boston, MA USA 02210

Tel: (+1) 844 448 1212
Email: info@rstudio.com

1. Purpose and Introduction

The purpose of this document is to demonstrate that certain R packages affiliated with the tidyverse[15], tidymodels[14], and r-lib[8] GitHub organizations, along with the gt package[2], when used in a qualified fashion, can support the appropriate regulatory requirements for validated systems, thus ensuring that resulting electronic records are "trustworthy, reliable and generally equivalent to paper records." For R, please see the document below:

R: Regulatory Compliance and Validation Issues
A Guidance Document for the Use of R in Regulated Clinical Trial Environments[10]

What qualifies as a record?

Validation guidance is a result of regulation, most notably the US Food and Drug Administration's 21 CFR Part 11[5]. This regulation was originally written to apply to health records. The definition of a record has subsequently been clarified and extended.

A key consideration for companies subject to this regulation is determining the extent to which analytic software systems such as R, and the outputs they produce (plots, tables, models, predicted values, etc.) constitute records or derived records subject to Part 11 compliance[5].

RStudio, following the work of the R core foundation and the R Validation Hub, does not consider the outputs created, in whole or in part, by R and R packages as records directly subject to compliance regulations[10]. However, these outputs, and as a result, the software used to create them, should follow the spirit of the regulation and strive to be trustworthy and reliable. Risk-based monitoring was first documented by the FDA in the reflection paper and guidance in 2011. Following the interpretation of FDA guidance, RStudio recommends that organizations consider a risk-based approach to the use of R and R packages[1]. This document outlines important considerations to the risk of using a set of R packages affiliated with RStudio, PBC. Separate from these packages is RStudio's products, which are developed using a controlled process that consists of distinct development phases as outlined here:

RStudio: Regulatory Compliance and Validation Issues
A Guidance Document for the Use of RStudio Professional Products in Regulated Clinical Trial Environments[12]

Risk-Based Approach and Open Source Software

Before covering the specific details of the Software Development Cycle and 21 CFR Part 11 compliance functions related to these affiliated packages, it is worth quickly noting the role of a

risk-based approach in open source software. Noting that the use of analytic software is often complex, the FDA has clarified[3] that organizations should take a risk-based approach aimed at ensuring analysis is trustworthy and reliable. Each organization will adopt its own standards and definitions for risk.

This document is intended to provide a reasonable consensus position on the part of RStudio relative to the use of packages (for the tidyverse, tidymodels, and r-lib package ecosystems as well as gt) within regulated environments and to provide a common foundation for people to meet their own internal standard operating procedures, documentation requirements, and regulatory obligations. Risk assessment and management are possible. **More so, it is our view that at a much deeper and fundamental level, open source software fulfills the role of a "trustworthy and reliable" system** far beyond any closed-source, proprietary software. Specifically, open source software is always available, to those interested, to be inspected and reviewed. By its very nature, the availability of open source software is not subject to the rise or fall of specific corporations. Nor is its use, review, and improvement subject to the economic means of the user. As a result, the outputs and methods of open source software are more amenable to being shared, more open to challenges and improvements, and significantly more repeatable and reproducible.

What is an R package?

R packages are extensions of the base R language for loading code, data, and documentation[]. R packages exist as components in an ecosystem of software used together for analysis. In addition to this document, we advise that users refer to:

Regulatory Guidance for the R Language[10]

Regulatory Guidance for Shiny and R Markdown

Regulatory Guidance for the RStudio Professional Software[12]

R Validation Hub[1]

The R Validation Hub resource strives to provide organizations with a risk-based approach to validating R add-on packages that are not expressly addressed in this document or other guidance documents[1]. The R Validation Hub is supported by The R Consortium, Inc., a group organized under an open source governance and foundation model to support the worldwide community of users, maintainers, and developers of R software[6]. As of August 2020, Its members include organizations such as Roche/Genentech, Microsoft, Google, Merck, Janssen Pharmaceuticals, and more leading institutions and companies dedicated to the use, development, and growth of R. Such efforts include supporting the R in Pharma conference as well as the R Validation Hub.

Scope of this Guidance

This document applies to those R packages included in the tidyverse, tidymodels, and r-lib GitHub organizations, as well as the gt package, which together share a common set of development standards and guidelines. Examples of these packages include ggplot2, dplyr, tidyr, parsnip, recipes, httr, and many more. The specific packages included in these GitHub organizations will vary over time. This document discusses the principles that guide the development of these packages, but it is incumbent on each user of the packages at a moment in time to qualify their installation, ensuring a complete and compatible cohort of packages. These packages may depend on packages that do not follow the principles outlined in this guidance, but for the most part we encourage organizations to focus validation efforts on the top-level functions used in their analysis. For more details, see section 6.1 of *A Risk-based Approach* [1]. As this document describes, the packages in these organizations indicate their maturity and readiness for use through a lifecycle designation[4]. The principles and practices described in this guidance are used throughout the packages in these organizations, but we strongly advise that regulatory use be limited to packages with a stable or maturing lifecycle designation.

This document is NOT in any fashion, applicable to any other R-related software or add-on packages. It is important to note that there is a significant obligation on the part of the end-user's organization to define, create, implement and enforce R installation, validation, and utilization related Standard Operating Procedures (SOPs). The details and content of any such SOPs are beyond the scope of this document.

This document is not intended to be prescriptive, does not render a legal opinion, and does not confer or impart any binding or other legal obligation. It should be utilized by the reader and his or her organization as one component in the process of making informed decisions as to how best to meet relevant obligations within their own professional working environment.

RStudio, PBC makes no warranties, expressed or implied, in this document.

2 Software Development Life Cycle (SDLC)

2.1 Operational Overview

The R packages in these organizations follow a common development lifecycle[9] and share core design principles codified in a style guide[16] that is followed by the authors of these packages and recommended to all potential R package authors. These specific packages are used quite frequently, often having millions of downloads a month. The size of the R user community provides for an extensive review of source code and testing in enterprise settings with all having full access to the source code. Additionally, this enables a superior ability to

anticipate and verify the performance and the results produced by the packages discussed within this document.

2.2 Source Code Management

The source code is managed using Git, a widely-used open source version control software. The source code is stored in public Git repositories and made available through Github, a Microsoft Affiliate that hosts Git repositories. Specifically:

- Tidyverse Source Code: https://github.com/tidyverse/tidyverse
- Tidymodels Source Code: https://github.com/tidymodels
- R-lib Source Code: https://github.com/r-lib
- gt Source Code: https://github.com/rstudio/gt

2.3 Testing and Validation

The packages are tested through a combination of unit tests, CRAN checks[18], and integration tests. Unit tests are written to cover specific functions and features provided by each package. The test suites are run in an automated fashion, and the test results, as well as the test coverage (the amount of code tested by the package unit tests), are publicly displayed. CRAN checks, also run automatically, provide a thorough test of whether the package source code can be built and installed across a variety of operating systems. Any package accepted on CRAN must pass a series of automated tests that enforce the CRAN submission policies[18]. These checks also account for checks for consistency in function definition and documentation. The results of these tests are available publicly for each package. Tests are automatically executed when any changes are proposed to the code, documentation, tests, or metadata in the package. These tests are run across multiple versions of R and multiple operating systems. The tidymodels packages also contain automated system and integration tests that run nightly[14]. Finally, packages released on CRAN undergo a test for compatibility with other dependent and reverse dependent packages[18].

The approach to testing, information on code coverage, and the details of CRAN checks are thoroughly documented[9].

2.4 Release Cycle

Packages are developed incrementally, following the best practices of the Git version control system. At specific points in time, a package will be released by incrementing the package version number, tagging the commit as a release in Git, archiving the new sources, and submitting the package to CRAN.

For each release, comprehensive release notes are maintained within the repository, and a

summary of the major changes and features in a release are documented in a blog post[11]. The tidyverse style guide provides the full details of the release process and guidelines[16].

2.5 Availability of Current and Historical Archive Versions

The source code for all packages, including every revision to this source code, is maintained in GitHub, a distributed service that provides free and public access to the projects' Git repositories (see section 2.2).

The released and archived versions of each package are maintained on CRAN, an extensive network of repository mirrors. Archive package versions can be found via the CRAN Package Archive. RStudio maintains a CRAN mirror with current and archived packages at https://cran.rstudio.com. The RStudio mirror uses Amazon Cloudfront to maintain copies of CRAN on servers all over the globe. These copies are updated off of the main CRAN mirror in Austria once per day. RStudio created this mirror to provide a consistently fast option around the world, a reliable option for users, and to provide a rich source of data about R and package usage. In addition, RStudio maintains a history of the CRAN mirror, accessible at https://packagemanager.rstudio.com, the benefits of which are extensively outlined in different strategies for reproducing package libraries[7], particularly useful in creating qualified environments.

The packages are leased using public licenses that are not subject to commercial control, ensuring they are fundamentally available in a greater sense than any commercial software.

2.6 Maintenance, Support, and Retirement

Each package provides an indication of its lifecycle status following these considerations[4]. As a summary, packages can have the following status: experimental, maturing, stable, superseded, archived, dormant, and questioning. Of these, maturing and stable are the most appropriate for clinical use, as in both cases work is done to maintain backwards compatibility.

RStudio understands support and maintenance to encompass a wide range of activities, corresponding to the open nature of the package development and use. Documentation standards[16] are applied to assist users in learning and appropriately using the software. A community forum affiliated with RStudio and the packages covered in this guidance document provides an opportunity for direct Q&A. Specific issues related to the software, such as requests for new features or identification of bugs, can be submitted and tracked on the relevant package GitHub site. For example, the ggplot2 issues board.

In addition, this cohort of packages takes special care to ensure compatibility with older versions of R, as documented in this R version policy[13].

2.7 Qualified Personnel

All development of the relevant packages occurs through open contributions to the package source code. Each contribution can be specifically enumerated using the Git version control system. For convenience, the history of contributors is available for each package. For example, the contributors to the dplyr package. Contributors come from a variety of backgrounds, but all of them agree to a code of conduct. Contributions are peer reviewed in the open, and contributions are considered in light of the testing and style guides previously documented.

Furthermore, due to the size of the contributor base, ggplot2 extends these principles and includes detailed contribution guidelines and governance structures.

While a wide range of contributors has enabled the success of these packages, a core set of primary contributors, many of whom are employed by RStudio, have relevant professional qualifications such as advanced degrees in related subject matter, peer reviewed publications, conference talks, and/or industry experience. Many members of RStudio hold a Ph.D. and/or Master's degrees from accredited academic institutions and have published extensively in peer reviewed journals. Several have written books on statistical computing technologies and applications using the packages detailed in this document.

2.8 Physical and Logical Security

The physical and logical security of the package source code is handled through GitHub's disaster recovery plan and security standards.

2.9 Disaster Recovery

The disaster recovery of the package source code is handled through GitHub's disaster recovery plan.

3 21 CFR Part 11 Compliance Functionality

3.1 Overview

The United States regulation, known as Title 21 CFR Part 11[5], or the "Electronic Records; Electronic Signatures" rule, provides information about what constitutes trustworthy and reliable electronic records and electronic signatures. FDA industry guidance for the use of electronic health record data in clinical investigations is here[17]. RStudio continues to monitor FDA regulations and guidelines that pertain to packages covered in this document. People can use

RStudio products and packages to build data collection, analysis, and other systems that can be used in compliance with Part 11. Compliance with this regulation ultimately depends on how a package is installed and used, how users are trained, and other factors. Users need to use packages according to the system requirements, install it according to the installation instructions, and use it via the user documentation. Users should refer to the predicate rule or consult the FDA or its guidance documents to determine whether packages comply with regulatory expectations.

The R add-on packages addressed in this document always operate through the base R programming language. As a result, these add-on packages inherit the CFR Part 11 compliance guidelines and interpretation documented for the base R language[10]. For this reason, we recommend all compliance considerations for R packages begin with the interpretation and compliance guidelines for R itself.

The following sections act only to address specific instances where the R add-on packages addressed in this document **differ from, amend, or supersede** the compliance guidance provided for the base R language.

As mentioned in the introduction, a key consideration is whether the outputs, methods, and results derived from the use of these R packages constitute records. Following the base R guidance, RStudio understands that these outputs are not records, and therefore CFR 11 does not directly apply. Quoting the base R guidance:

R's design and development are focused on reporting, by enabling leading edge statistical analysis and presentation, rather than on data management tasks as illustrated by transaction/data processing and related functionality.

In the following sections, the term record means an electronic record that is interpreted to fall within the remit of Part 11 as defined in FDA Guidance for Industry Part 11, Electronic Records; Electronic Signatures– Scope and Application (2003).

R is not intended to create, maintain, modify or delete Part 11 relevant records but to perform calculations and draw graphics[10].

However, where the use of these R packages may be interpreted as creating derived records, we provide guidance for CFR Part 11 compliance.

3.2 11.10(b) The ability to generate accurate and complete copies of *records* in both human readable and electronic form suitable for inspection, review, and copying

The R packages covered under this guidance create a wide variety of outputs including graphics (ggplot2), tables (gt), summaries and analysis of records (tidyverse), and models (tidymodels). Where applicable, these outputs are designed to be human readable and are suitable for inspection, review, and copying.

Further, a core design principle uniting these R packages is a functional programming interface that results in *code* that is also human readable and intelligible, assisting in the interpretation of the steps applied to the raw records to create the resulting outputs.

11.10(c) Protection of records to enable their accurate and ready retrieval throughout the records retention period

This section inherits the same guidance found in the base R guidance[10].

11.10(d) Limiting system access to authorized individuals

This section inherits the same guidance found in the base R guidance[10].

11.10(e) Use of secure, computer-generated, time-stamped audit trails to independently record the data and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying

This section inherits the same guidance found in the base R guidance[10].

3.3 11.10(f) Use of operational system checks to enforce permitted sequencing of steps and events, as appropriate

RStudio, following the interpretation provided in the base R guidance, understands this item to mean the effective use of an interface that reduces errors made by an operator. The packages covered in this guidance provide an opinionated interface designed to limit operator error while maintaining the flexibility essential to analysis software. The packages share a common design philosophy for capturing and surfacing useful error messages: https://style.tidyverse.org/error-messages.html[16].

Where relevant, additional packages provide checks and interfaces that guide appropriate use. For example, the tidymodels packages provide interfaces to data pre-processing and model fitting that structure these procedures to discourage inappropriate applications of methods.

In conjunction with code reviews and validation conducted by RStudio and community peer review (as described elsewhere in this document), these features provide for the use of the packages in a production environment.

Finally, these packages follow R core's system checks for package installation, ensuring the cohort of packages are correctly installed. Further, meta packages like tidyverse and tidymodels aid the user in acquiring an appropriate set of packages, and clearly display messages to delineate what has been installed and made available to the user. This interface reduces error and helps encourage appropriate use.

11.10(g) Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand

This section inherits the same guidance found in the base R guidance[10].

11.10(h) Use of device (e.g., terminal) checks to determine, as appropriate, the validity of the source of data input or operational instruction

This section inherits the same guidance found in the base R guidance[10].

11.10(j) The establishment of, and adherence to, written policies that hold individuals accountable and responsible for actions initiated under their electronic signatures, in order to deter record and signature falsification

This section inherits the same guidance found in the base R guidance[10].

3.4 11.10(k) Use of appropriate controls over systems documentation

RStudio understands this item to mean that there must be control over who can change and access system documentation. As documented in section 2, the packages provide for specific contribution guidelines *including contributions to documentation*, and documentation is maintained and governed in the same version control systems as the source code.

Section 11.30 Controls for Open Systems - the system shall employ procedures and controls designed to ensure the authenticity, integrity and as appropriate the confidentiality of electronic records from the point of their creation to the point of their receipt. Additional measures such as document encryption and use of appropriate digital signature standards to ensure, as necessary under the circumstances record authenticity, integrity and confidentiality

This section inherits the same guidance found in the base R guidance[10].

Key References

1. A Risk-Based Approach for Assessing R Package Accuracy within a Validated Infrastructure. *Nicholls, A. et al.* 2020-01-23. https://www.pharmar.org/white-paper/

2. gt Package Site. Accessed 2020-09-25. https://gt.rstudio.com/

3. Guidance for Industry: Part 11, Electronic Records; Electronic Signatures -- Scope and Application. 2003-08. https://www.fda.gov/media/75414/download

4. Lifecycle Badges; Tidyverse. Accessed 2020-09-25. https://www.tidyverse.org/lifecycle/

5. Part 11, Electronic Records; Electronic Signatures - Scope and Application Guidance for Industry. 2003-08-01. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application

6. R Consortium About Page. Accessed 2020-09-25. https://www.r-consortium.org/about

7. Reproducible Environments. *Lopp, S.* Accessed 2020-09-25. https://environments.rstudio.com/

8. R-Lib Github Organization. Accessed 2020-09-25. https://github.com/r-lib

9. R Packages. *Wickham, H. Bryan, J.* Accessed 2020-09-25. https://r-pkgs.org/

10. R: Regulatory Compliance and Validation Issues A Guidance Document for the Use of R in Regulated Clinical Trial Environments. *R Foundation for Statistical Computing.* 2020-03-2018. https://www.r-project.org/doc/R-FDA.pdf

11. RStudio Blog Package News Category. Accessed 2020-09-25. https://blog.rstudio.com/categories/packages

12. RStudio: Regulatory Compliance and Validation Issues. *RStudio, Inc.* 2019-06-26. https://rstudio.com/wp-content/uploads/2019/06/rstudio_compliance_validation.pdf

13. R Version Support. *Averick, M.* 2019-04-01. https://www.tidyverse.org/blog/2019/04/r-version-support/

14. Tidymodels. Accessed 2020-09-25. https://www.tidymodels.org/

15. Tidyverse. Accessed 2020-09-25. https://www.tidyverse.org/

16. Tidyverse Style Guide. *Wickham, H.* Accessed 2020-09-25. https://style.tidyverse.org/

17. Use of Electronic Health Record Data in Clinical Observations: Guidance for Industry. 2018-07. https://www.fda.gov/media/97567/download

18. Writing R Extensions, Version 4.0.2. *R Core Team.* 2020-06-02. *https://cran.r-project.org/doc/manuals/r-release/R-exts.pdf*