SCTO Validation Platform

R add-on Package Risk Assessment

Release date: 2025-01-01

This document is an integral component of the SCTO Validation Platform

---

SCTO PLATFORMS

ℹ Document development, review and version history

Development and Review

Authored/revised by:

Name	Date
Michael Coslovsky,[1] Nicole Graf,[2] Julien Sauser,[3] Christina Huf,[4] Christine Otieno,[5] Elio Carreras[6]	2025-01-01

Version History

Version	Date	Author(s)	Summary of Changes
1.0	2025-01-01	Michael Coslovsky, Nicole Graf, Julien Sauser, Christina Huf, Christine Otieno, Elio Carreras	Initial version

1 Purpose

This document describes the metrics used by members of the Swiss Clinical Trial Organization for the evaluation of add-on packages for the statistical software R and provides guidelines for their application. It provides a table of the metrics and a short explanation of the considerations behind each of the metrics.

Assessing the risk associated with using a software package is one of the steps required for working under a validated environment, as determined in the **computerized systems validation policy for R**. With respect to R add-on packages, the risk of (using) an R add-on package is the opposite of the confidence we have in the package's delivered output: R add-on packages which we are confident in using are of low risk, and vice versa. The metrics described below are the building stones in determining this risk. A final risk score is determined based on a weighting scheme including these metrics.

Following this assessment, the assessor determines whether any functions of the R add-on package need to be tested (function/unit testing) for the package's intended use in a specific R product. The documentation of any actions following the initial risk assessment is listed separately.

This document is part of the **SCTO Computerized Systems Validation Policy for R** and refers to the SCTO's standard operation procedure R add-on package Function testing procedure.

Based on the principals below, the table in the addendum provides the metrics' values to be documented upon the evaluation of an R add-on package for the SCTO platform statistics. Note that the evaluation is per R add-on package with a specific version number. Package version changes and updates require re-assessment of the metrics for the new version.

[1] Head data-analysis, Department Clinical Research, University of Basel
[2] Statistician, Clinical Trials Unit, Kantonspital St. Gallen
[3] Statistician, CHUV
[4] Head Quality Management, Department Clinical Research, University of Bern
[5] IT Quality Manager, Department Clinical Research, University of Basel
[6] Senior Statistical Programmer, SAKK

2 Abbreviations & Definitions

SOP

Standard operating procedure

R

The R statistical programming language and environment

R Package

- A collection of code and functions providing an extension to base R installation
- provided by members of the R community
- available via CRAN and other distribution repositories

R add-on Package

An R package that is not part of the base distribution of R or part of the “R recommended packages”

R recommended packages

The collection of “recommended packages”, developed and validated by members of the R Development Core Team, as listed in the document R: Regulatory Compliance and Validation Issues, The R Foundation for Statistical Computing, 2021.

R product

Any deliverable produced using R via the processes below. Typical statistical products include sample size estimation, statistical analysis plan, an analysis report etc.

Risk

As per ISO 31073 risk is defined as:

The effect of uncertainty on objectives, where:

1. An effect is a deviation from the expected. It can be positive, negative or both, and can address, create or result in opportunities and threats.
2. Objectives can have different aspects and categories, and can be applied at different levels.
3. Risk is usually expressed in terms of risk sources, potential events, their consequences and their likelihood.

In the context of risk assessments and risk metrics the focus is usually on negative effects.

3 Scope

This document describes the metrics to be used for assessing the risk of an R add-on package according to the **SCTO Computerized Systems Validation Policy for R** guidelines on R package validation. It is applicable to all R add-on packages.

Out of scope are R packages included in the official R-distribution including Base R packages. Also out of scope is the collection of “recommended packages”, developed and validated by members of the R Development Core Team, as listed in the document R: Regulatory Compliance and Validation Issues, The R Foundation for Statistical Computing, 2021. Base R packages and the “recommended packages” are considered validated with the R base installation, as stated in the policy document.

4 Guideline

4.1 The R package risk metrics explained

The following metrics are considered when assessing an R add-on package. These are listed in Appendix-1 below. Here we provide a precise definition and, in parentheses, the name of the metric in the collection form:

1. **Package name, version, release date**: version and release date of the specific version being assessed. These are required for identifying the package and for documentation. Packages will be re-assessed upon update and version changes; documentation of older versions remains in the document. These values are not included in the risk calculation.
2. **Purpose** (`statistical_package`) We define three risk levels for a package, depending on the package's purpose and methodology:
   a. "non-statistical" packages: packages that deal only with data-wrangling and manipulation (e.g., dplyr) or with reporting processes (e.g., Sweave, xtable). Such processes are of 'low risk' as no statistical calculations are performed, and data-errors are, comparatively, easy to detect. Similarly, packages associated with application interfaces such as Shiny application are considered "non-statistical".
   b. "Statistical with published methods": packages that perform statistical calculations, the majority of which based on known methods, or on methods that have been published in peer reviewed journals. These packages obtain a "medium risk" status.
   c. "Statistical non-published methods": packages that perform statistical calculations, but the majority of underlying methods have not been published in a peer reviewed journal. These packages obtain a status "high risk".
3. **Author** (`author`): The author(s) of a package will be viewed as indicator for its trustworthiness. If package authors (noted as 'aut' in the package description, e.g., as listed on CRAN) are well-known within the statistical, data-science and R communities and have credentials based on their qualifications, education, present and past affiliations, the risk of the respective package will be scored as low. If package authors have credentials based on their qualifications, education, present and past affiliations, but are not well-known within the statistical, data-science and R communities, the risk of the respective package will be scored as medium. If package authors are not well-known within the statistical, data-science and R communities and have no clear credentials based on their qualifications, education, present and past affiliations, the risk of the respective package will be scored as high.
   Note that whether an author is 'well-known' in the community is a subjective assessment and accepted as such; in addition, groups of authors are evaluated as a collective.
4. **Maintainer** (`maintainer`): The package has a named maintainer who's contact details (email) are available and published. A positive answer provides a "low risk" score. A package with no named maintainer is scored as high. The rationale behind this metric lies in the fact that the indication of name and email is evidence of a package's active maintenance and availability of contact in case of bugs and/or suggestions.
5. **Number of dependencies** (`n_dependencies`): This metric assesses the level of risk associated with the number of dependencies a package relies on. Dependencies are other packages or processes that the evaluated package depends on, as listed in CRAN under "depends" and/or "imports". The risk of unexpected behaviour increases with the number of dependencies, since there is a greater likelihood of issues on a specific routine if updates are performed on a dependent package. Great care should be considered while using packages involving many dependencies.
   The listed number is converted into a [0, 1] score, with 0 representing low number of dependencies (= low risk) and 1 representing many dependencies (= high risk). Taking a similar approach for the transforming the number to a score as the 'riskmetric' package (R validation hub, 2023), we use a simplification of the classic logistic curve $1/\left(1+exp(-k(x-x[0])\right)$ as a scoring function. A sigmoid midpoint is 4 dependencies, ie., x[0] = 4 and logistic growth rate of k = 0.5.

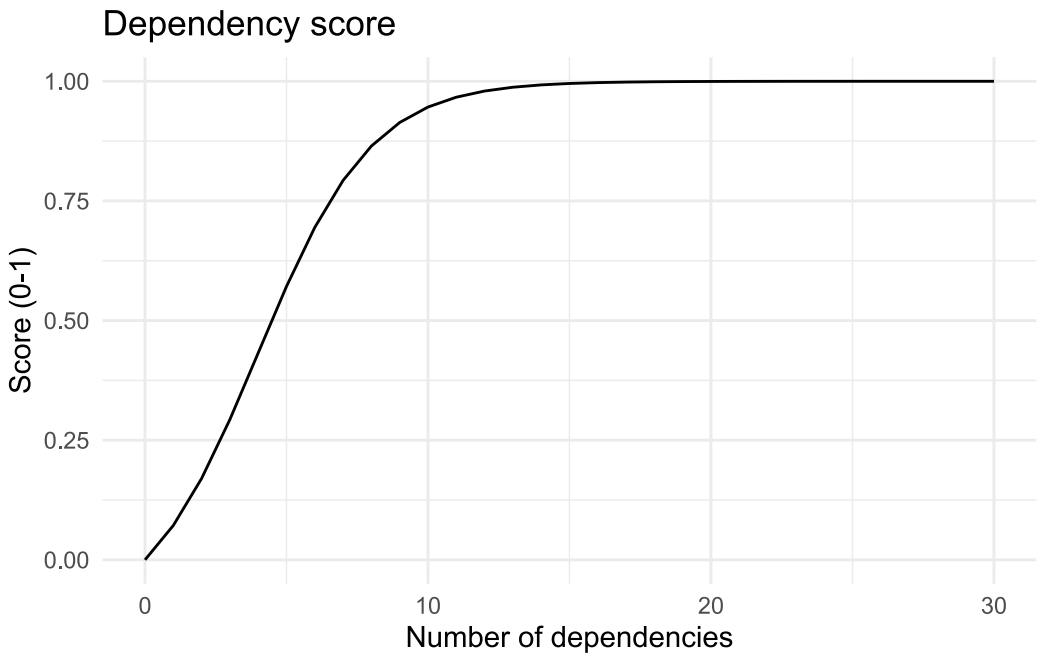

Figure 1

6. **CRAN or Bioconductor** (`on_cran`): CRAN and Bioconductor impose a set of requirements that a package has to meet for it to be released on their official platform. Thus, they provide a procedure of evaluating a minimal quality standard for a technical, though not statistical, appropriateness of the package. This metric assesses whether the package is on CRAN or Bioconductor. Being on CRAN/Bioconductor provides low risk (yes = 0) while not being on them represents high risk (no = 1).
7. **Documentation of source code** (`source_code_documented`): Ideally, the source code is available (for example, on github) for examination and commented (yes = 0). Source code that is not commented or difficult to follow, or is not available, is considered not documented (no = 1).
8. **Number of downloads in the last year** (`nr_downloads_12_months`): More downloads suggest more extensive community and user testing and greater chances of bugs or errors being identified and reported. To fill in check the logs of CRAN's or Bioconductor's reporting systems and report the number of downloads for the package in the last 12 months. Using the `cranlogs::cran_downloads()` function to this end is acceptable.
   The number of downloaded packages is converted to a score [0,1], with 0 representing low risk (many downloads) and 1 high risk (few downloads). For the conversion of the number into a score we follow a similar approach as approach taken by the 'riskmetric' package (R validation hub, 2023) and use a simplification of the classic logistic curve $1/\left(1+exp(-k(x-x[0])\right)$ with the logistic growth rate k = −0.5 and a log-scale for the number of downloads (log(x)). The midpoint lying at log(100,000 downloads).

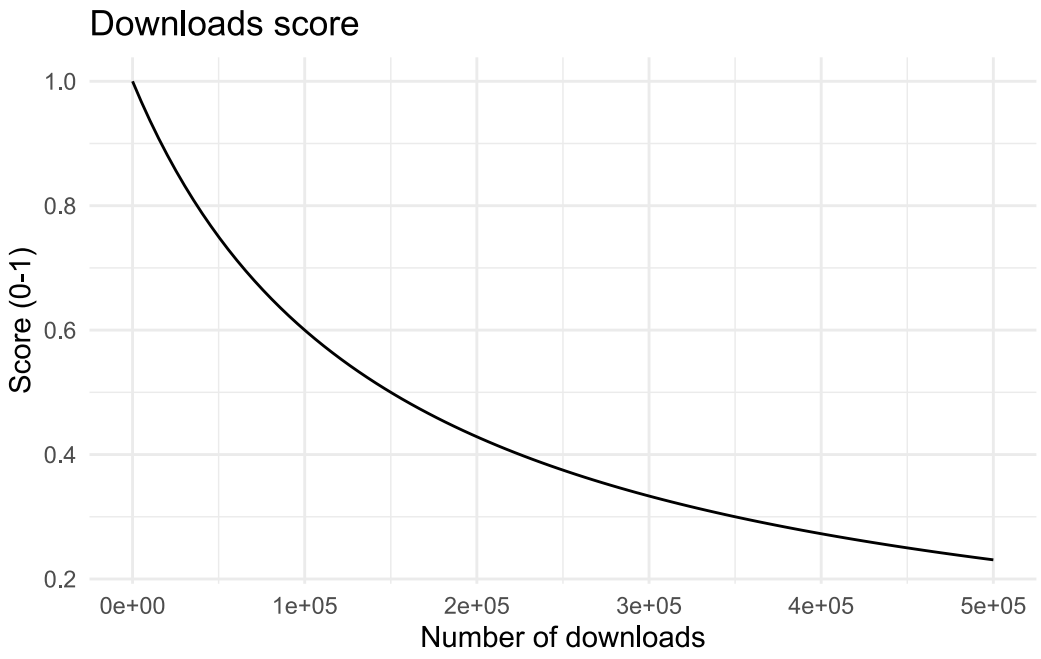

Figure 2

9. **Bug reporting** (`bug_reporting_active`): Available option for reporting bugs suggests higher chance that errors have been corrected. Check whether there is an option to report bugs – ideally via url or email. Yes = 0 (low risk); no = 1 (high risk).
10. **Vignettes** (`has_vignettes`): Does the package have one/more available vignettes? Vignettes provide an explanation of the use of the package, increasing its trustworthiness and correct use. Having at least one vignette suggests a "low" risk for this category (yes = 0); having no vignettes suggests a "high" risk score (no = 1).
11. **Tested functions** (`has_tests`): Perform a search and give a general grade of low/medium/high based on the answers to the following questions: does the package have unit and/or function tests performed by the authors? are they comprehensive? are they well documented? Assess the above to determine whether tests were conducted sufficiently and documented. Accordingly, you can classify into low risk (=0); insufficient testing could be medium risk (=0.5); no documented testing at all are categorized as high risk (=1).
    Note that test functions are often listed in 'test' folder in the package's source files, e.g., on github.

4.2 Calculation of the final risk score

The final score is a weighted summary of all the measures above in the range [0, 1], with lower scores representing lower risk and higher scores higher risk. In this version of the guideline all measures are considered equally important, and the score is a simple arithmetic mean of the measures.

The [0, 1] score is then categorized to low, medium and high-risk R add-on packages:

- Score ≤0.25: low risk
- 0.25 < score ≤ 0.75: medium risk
- Score >0.75: high risk

As described in the SCTO computerized systems validation policy for R, the risk associated with the R add-on package is evaluated alongside the risk associated with a specific project to determine which actions may be required to use the package. High risk R add-on packages, or medium risk packages in high-risk projects, for example, may need (project-) specific function testing.

4.3 Documentation

The assessment of risk associated with an R add-on package should be documented and may be updated with time.

The SCTO has developed a platform on GitHub, on which the risk assessment can be performed (https://github.com/SwissClinicalTrialOrganisation/validation_platform). R add-on package assessments are done as “issues” on the GitHub platform (https://github.com/SwissClinicalTrialOrganisation/validation_platform/issues).

The platform allows performing the assessment and calculating the final risk based on the above metrics, while recording and documenting the assessment. Apart from the metrics themselves (Section 6), the additional points listed in (Section 7) are collected and documented while performing the assessment on the platform.

5 References

R Validation Hub, Kelkhoff D, Gotti M, Miller E, K K, Zhang Y, Milliman E, Manitz J (2023). _riskmetric: Risk Metrics to Evaluating R Packages_. R package version 0.2.3, https://CRAN.R-project.org/package=riskmetric.

ISO 31073:2022(en) Risk management - Vocabulary

R: Regulatory Compliance and Validation Issues, The R Foundation for Statistical Computing, 2021

6 Appendix 1: The SCTO R add-on package risk metrics

Parameter	Explanation	Possible values	Risk level*
`package`	name of the package as called for installation		
`version`	Version number of the package evaluated		
`release_date`	Date of release of the current evaluated version of the package		
`author`	name of the main author or developing group	well-known or known credentials / has credentials / no clear credentials or group association	low / med / high
`maintainer`	Is there a maintainer listed for the package and are their contact details available?	Yes / No	low/high
`package_purpose`	Statistical packages implement statistical or machine learning algorithms. Non- statistical packages are used for reporting for example	non-statistical / statistical-published / statistical	low / med / high
`n_dependencies`	Number of dependencies. The more packages a package depends on, the more chance for errors/bugs to be found	Nr of dependencies (transformed to [0,1]	0-1
`on_cran`	is the package on available from CRAN or bioconductor?	Yes / No	low / high

Parameter	Explanation	Possible values	Risk level*
source_code_documented	is source code available, accessible and documented (i.e., well-structured and including comments) or is the source code unavailable or not clearly commented?	Yes / No	low / high
nr_downloads_12_months	Checked using the cranlogs packages implementation - potentially the CRAN (/biconductor) reporting system	Nr of downloads (transformed to [0,1])	0-1
bug_reporting_active	address for bug reporting exists	Yes / No	low / high
has_vignettes	does the package have one/more vignettes?	Yes / No	low / high
has_tests	does the package have unit and/or function tests performed by the authors? are they comprehensive? are they well documented	yes-comprehensive / yes-not-comprehensive / no	low / med / high
final_risk	a global risk, based on and weighing in all the grading metrics. At version one, the weight of all metrics is the same, and the global risk is the average of all metrics.	[0,1] with 0 = low risk and 1 = high risk. Cutpoints categorize into low / medium / high	low / med / high

* Risk level is always a value [0, 1]. For binary metrics low = 0 and high = 1. 'medium risk' takes a value of 0.5. Continuous metrics such as 'nr of downloads' are transformed to a value [0, 1].

7 Appendix 2: The SCTO R add-on package documentation

Parameter	Explanation
package_name	name of the package as called for installation
version	Version number of the package evaluated
release_date	Date of release of the current package
date_of_risk_assessment	The date on which the package risk was performed. Testing probably happens after (create_date).
final_risk	As calculated from the SCTO risk measures (table above)
assessor_name	Who did the risk assessment
scto_relationship	SCTO association
comments	for example, specific functions that are not recommended or specific issues