**PSI Special Interest Group: Application and Implementation of Methodologies in Statistics (AIMS)**

**Want to use R? Here are some top tips to consider before you start.**

In the pharmaceutical industry (especially phases I-IV), most of us are likely to use SAS more often than R. There is also probably a general opinion amongst the non-R users, that SAS has been validated extensively, that the FDA endorse the use of SAS and that by using SAS we have to do no validation of the SAS procedures ourselves. Conversely, R feels like it has never been validated and that any packages in R can be created, edited and updated in an uncontrolled way. Therefore, before using an R package for regulatory analysis, each user would need to perform and document validation on each of the packages they want to use. By doing such validation, you’d gather evidence that the package is performing as expected.

This article will hopefully provide a counter argument to those ideas and help you feel more confident to use R. As statisticians, I challenge you to assess each package on a case by case basis and make up your own mind of whether you feel confident to use the package for your analysis.

**Why use R at all?**

R is a very powerful object-orientated language which can efficiently produce analyses and high quality graphics. It can be used for high performance computing and allows easy integration with other software. It can also be used for complex analyses when they are not available in SAS. It is available to download for free from http://cran.r-project.org/.

**What do we mean by validation?**

The FDA (2014) defines validation as “Establishing documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specifications and quality attributes”. However, validation can be broken down into 2 parts; validation of the installation (to confirm it is performing as the author intended for given examples) and validation of the packages within the installation (to confirm the analysis that the package performs is statistically accurate). This article only discusses the latter and users still need to ensure installation validation has been performed according to the company software installation SOPs as applicable.

**How do SAS and R differ with regards to validation?**

No statistical software can be purchased as “pre-validated” software in the regulatory context. Validation is entirely on the end user to implement. Given the dominance of SAS in the industry, many pharmaceutical companies have substantial evidence collected over a long period of time, detailing the testing & validating of SAS procedures. Therefore, rather than reinventing the wheel (duplicating testing which has already been performed in the past by others), we make the decision to use the SAS procedures based on its history of extensive previous use. Therefore, why can’t we apply the same thought process for the use of R packages? R packages have received “validation” through a variety of sources. Validation by the original authors of the package, by users of the package in academia, by users in a wide variety of industries (Google, Amazon and Microsoft), by pharmaceutical companies (particularly pre-clinical or in proteomic/genomic research where R can be the main software used), by companies such as Mango Solutions (providers of data science software and services) and even by the FDA themselves, who have been using R internally now for some time. Whilst we cannot deny that continual validation of each new package and its future versions is necessary, in the same way as a SAS user would trust a long standing SAS package (such as PROC LIFETEST, PROC GLM, PROC LOGISTIC), can we start to think about R packages in the same way? If we believe this to be the case, then the question this article would like to answer is: Which R packages can we add to our safe list and what should we do if we need a use a package not on our safe list?

**Did you know about “Official R distribution” packages?**

Official R distribution packages are packages formally released by the R Foundation for statistical computing and are sometimes called the Base R plus Recommended Packages. These packages were developed, tested and updated by a core group of experts. A full list of these packages along with further information can be found in the following guidance document (R Foundation for statistical computing, 2014). Further information about the packages; including how to use them and how to test the installation, can be found in the manuals <http://cran.r-project.org/manuals.html>. Many of these R packages have been available for over 20 years and have therefore been open to extensive worldwide peer critique. This large worldwide team of “testers” have been checking the packages using a wide range of scenarios to identify where the package fails or doesn’t perform as expected. Over many years of use, it could be argued that any bugs are now fixed and hence we have a set of packages for which we can rely on to produce trustworthy results.

**What if the package you need is an add-on package downloadable via Comprehensive R Archive Network (CRAN) or other software distribution?**

It’s highly likely, that you will need to use an R package that is not part of the official R distribution packages. Don’t panic though; this still does not necessarily mean you’ll have to spend weeks and weeks on R software package testing! You need to determine the level of risk you will be taking and then decide what validation, if any, you think is appropriate. Consider the following points before using a package.

1. What would be the impact of an error in the analysis?
   1. Primary analysis for a phase III regulatory submission – High risk
   2. Exploratory, sensitivity or secondary analysis – Potentially lower risk
2. How likely is it that the package will not perform as intended?
   1. The age of the package used in combination with its revision history and frequency of downloads or citations can help to assess risk. For example:
      1. Brand New package with 0 revisions – High risk
      2. <2 year old package still receiving regular new revisions some large changes - High risk
      3. 2-5 year old package still receiving regular new revisions but very small changes – Medium risk
      4. >5 year old package which had revisions in its first few years but is now stable and is still widely cited – Low risk (likely to have been tested to a level that you would be confident to use it without further testing)
      5. A package which is not widely cited (very few downloads) or is no longer maintained – High risk (it may be defunct and soon to be retired from CRAN altogether)
   2. Does the author/maintainer of the package have a firm reputation?
      1. New PhD graduate who has written 1 R package – High risk (the package is likely not to have been used much to date, not tested much in different settings and could therefore be quite unstable)
      2. An author who presents at the useR! or other respected conference or who is a member of the core group of the R Foundation for statistical computing – Lower risk
      3. An author or maintainer of several long established R packages – Lower risk
      4. An author who has published this method in peer reviewed journals which have been widely cited – Lower risk
      5. An author with clear, accurate documentation using standard software development, bug fixes and version control procedures – Lower risk

Once the level of risk is established, you may feel comfortable preceding with your analysis, in the same way you would with a SAS procedure. However, if you are still in doubt of the validity of the package, you may decide to perform your own validation. One method to increase your confidence in a package is to perform replication of the R analysis in SAS or another software. Even if you can’t repeat the exact same analysis, perhaps a simpler analysis can be replicated which would help you feel more confident about the risk associated with the package. Remember to retain all evidence of such replication and validation.

**Example: I want to use the package “rvest”, how do I convince myself that it’s “safe” to use?**

You can investigate the package using the CRAN website or from within R software once rvest is installed.

On the CRAN site, you can find a list of all packages listed by name: <https://cran.r-project.org/web/packages/available_packages_by_name.html>. Clicking on the “rvest” link takes you to the index which contains the latest version and when it was published. In addition, it has the reference manual and archive (<https://cran.r-project.org/src/contrib/Archive/rvest/>). The archive reveals the first version was 22 Nov 2014 and there were 4 versions prior to the one you had installed.

If you prefer to explore the detail in R, you can use the installed.packages() function. The priority below tells you if the package is one of the Base R or Recommended packages. It turns out that rvest is an add-on package and hence further research into its validation will be required.

>installed.packages()

Version Priority

rvest "0.3.2" NA

stringr "1.2.0" NA

xml2 "1.1.1" NA

base "3.4.1" "base"

boot "1.3-19" "recommended"

class "7.3-14" "recommended"

compiler "3.4.1" "base"

Using the packageDescription () function, provides the full detail of the version you have installed and name of the author.

>packageDescription("rvest")

Package: rvest

Version: 0.3.2

Title: Easily Harvest (Scrape) Web Pages

Description: Wrappers around the 'xml2' and 'httr' packages to make it easy to download, then manipulate, HTML and XML.

Authors@R: c( person("Hadley", "Wickham", , "hadley@rstudio.com", c("aut", "cre")), person("RStudio", role = "cph") )

Depends: R (>= 3.0.1), xml2

Imports: httr (>= 0.5), selectr, magrittr

Suggests: testthat, knitr, png, stringi (>= 0.3.1), rmarkdown, covr

Encoding: UTF-8

License: GPL-3

LazyData: true

VignetteBuilder: knitr

RoxygenNote: 5.0.1

URL: https://github.com/hadley/rvest

BugReports: https://github.com/hadley/rvest/issues

NeedsCompilation: no

Packaged: 2016-06-16 16:20:43 UTC; hadley

Author: Hadley Wickham [aut, cre], RStudio [cph]

Maintainer: Hadley Wickham <hadley@rstudio.com>

Repository: CRAN

Date/Publication: 2016-06-17 08:57:12

Built: R 3.4.1; ; 2017-09-10 10:15:33 UTC; windows

-- File: C:/Users/Lyn/Documents/R/win-library/3.4/rvest/Meta/package.rds

A quick google search of the author, Hadley Wickham reveals he is the Chief Scientist for RStudio and an Adjunct Professor of Statistics at the University of Auckland, Stanford University, and Rice University. In addition to rvest, he’s authored ggplot2, dplyr, tidyr, strongr, readr, xml2 and many more packages. Therefore, we start to gain confidence in the quality of the package. However, to convince myself that it would be OK to use, I can read up on the manual and also run a test using one of the functions html\_text() to ensure that that part of the package performs as expected.

The rvest package function html\_text() can be used to find extract attributes and text from html. I use it below to extract information about the number of historic versions that exist for each R package. I’m informed there have been 5 rvest versions which is consistent with what I found visually by checking the archive. Clearly any such replication of results, should have the documentation retained in case of later scrutiny, even if unofficial.

> num.versions=function(package){

+ page=read\_html(paste0("https://cran.r-project.org/src/contrib/Archive/", package,"/"))

+ doc=html\_text(page)

+ paste("Number of versions:",length(unlist(str\_extract\_all(doc, "tar\\.gz")))+1)}

> num.versions("rvest")

[1] "Number of versions: 5"

To see the source code, simply type the function name and the source code is listed:

> library

function (package, help, pos = 2, lib.loc = NULL, character.only = FALSE,

logical.return = FALSE, warn.conflicts = TRUE, quietly = FALSE,

verbose = getOption("verbose"))

{

testRversion <- function(pkgInfo, pkgname, pkgpath) {

You may also want to explore the news() function which gives detail of any bugs that have been identified and fixed.

**What about the regulatory requirements?**

The FDA’s statistical software clarifying statement states that “statistical software is not explicitly discussed in Title 21 of the Code of Federal Regulations [e.g. in 21CFR part 11]” and that the FDA do not require pharmaceutical statisticians to use any specific software but that the software used should be documented in the submission including the version and build identification. See the following for more detail: [https://www.fda.gov/ForIndustry/DataStandards/StudyDataStandards/ucm445917.htm](https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.fda.gov%2FForIndustry%2FDataStandards%2FStudyDataStandards%2Fucm445917.htm&data=02%7C01%7CTaylorLyn%40prahs.com%7C08a723d3628c4eafff9008d4fab9ea49%7C1cef9a5962ec418a96662b3afc2d2cb0%7C0%7C0%7C636409123787489692&sdata=JI%2Bqegxrk2H%2FopsHhQwpV7ZFx%2FBYxqmLW7a1pIIIW%2FE%3D&reserved=0)

The R Foundation (<https://www.r-project.org/doc/R-FDA.pdf>), state that “R is not intended to create, maintain, modify or delete Part 11 relevant records but to perform calculations and draw graphics”… “In conjunction with local policies regarding record access control, retention and archival, R meets the FDA requirements for the inspection, review and copying of records”. In other words, if data is collected and stored in a data management database system, then whilst the analysis is created in R, if it is subsequently output to another storage system (likely an electronic Trial Master File), then it’s that system which must comply with 21 CRF Part 11 data storage rules. It is recommended though to use date(), Sys.time(), Sys.Date() and Sys.timezone() functions to apply date/time stamps to analyses.

**Conclusions**

In summary, we cannot simply ignore the validation of R, however, we should also not be held back by the fear of the lack of validation. By taking a few simple steps, you assess the risk you would be exposed to by using the package and then make your own mind up whether you feel confident or not to use it.

Lyn Taylor (PRA HealthSciences on behalf of PSI AIMS SIG) with special thanks to Marc Schwartz (R Foundation Ordinary Member) for his guidance

All of the AIMS SIG’s work is located on the PSI website: <http://www.psiweb.org/about-us/sigs-special-interest-groups/aims>. We welcome any comments or suggestions for future articles. Please also get in touch if you would like to share ways in which you have used R in the pharmaceutical industry.

**References**

FDA, Glossary of Computer System Software Development Terminology, (<https://www.fda.gov/ICECI/Inspections/InspectionGuides/ucm074875.htm>), last updated 25Nov2014

R Foundation for statistical computing, R: Regulatory Compliance and Validation Issues: A Guidance Document for the Use of R in Regulated Clinical Trial Environments, (<https://www.r-project.org/doc/R-FDA.pdf>), 14Dec2014