OpenVal® CHEAT SHEET

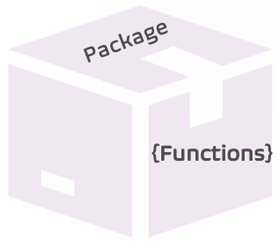
learn more at https://www.atorusresearch.com/openval/

Validated by Us,
Ready for You

Open-source languages, such as R, **accelerate your data analytics**, unlocking greater overall value for your clinical research

With open-source packages, users have access and opportunity to learn code that can be used in any company and on any project. This **dramatically reduces ramp-up time** and need for company or project specific macros

Through R, companies can leverage interactive and engaging web applications. Think interactive figures where the reviewer can change the parameter presented on a graph, or an interactive table where the reviewer can drill down on the actual data points. This allows a reviewer to **explore the data on their own**, eliminating time consuming ad hoc requests



A **function** is a piece of code that takes some inputs and does something specific with them. In R the majority of programming is done using functions

A **package** is a collection of R functions, and sometimes other programming languages, in a well-defined format. An R installation comes with a set of base packages, plus there are over 20,000 user contributed packages on CRAN that can be installed

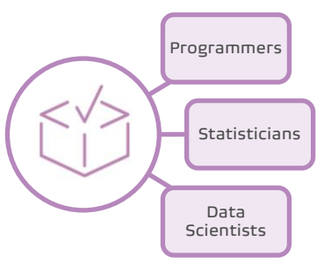
What is OpenVal®?

OpenVal® is a subscription-based validated package delivery system in support of the R programming language.

A subscription includes:
- Validated installation of the current version of OpenVal® packages within company environment
- Two Major Release installations of OpenVal®
- Patch Release installations as required
- Product Support of current and prior releases of OpenVal®

Each Major Release of OpenVal® contains:
- Additional packages
- An updated R version and updated package versions as applicable
- Release notes containing the changes from the previous release

Who Uses OpenVal®?



Anyone working on a task requiring a validated or GxP compliant environment is an end user of OpenVal®

Both novice and expert R users can leverage and benefit from OpenVal®

Why OpenVal®?

OpenVal® harnesses the best parts of open-source and applies the necessary control needed for a GxP environment

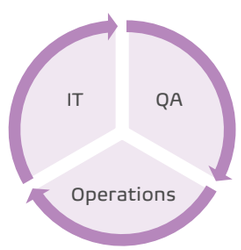
WE DID THE WORK, SO YOU DON'T HAVE TO

Leverage the flexibility, transparency, and deep community support of open-source with no extended build time, no cumbersome code, and no change management

Unlock the confidence and trust in our thousands of test scripts, cited references, and specialized programming hours

Use across multiple business units and departments giving users the capabilities to do all tasks they currently do in SAS® in R

OpenVal® expedites R adoption by reducing the effort of cross-collaboration between IT, QA and Operations. The framework is all code-based, resulting in programmatic execution, creating a Validation Report to store as an artifact of the validation process



How is OpenVal® Used?

OpenVal® can be installed in any platform with an R environment, and Atorus' experts can help with every step of the process

To use OpenVal®:

Open an R session
- All OpenVal® packages and their dependencies will already be installed

Use packages
- Choose from a variety of packages applicable to day-to-day clinical programming tasks

Generate output
- Packages are validated on system - giving the ability to create accurate results

General users cannot install additional packages or update existing package versions, which ensures a consistent and controlled environment

All releases of OpenVal® are maintained, and concurrent installations can be managed, allowing different projects to use different versions as applicable to maintain a stable environment for the duration of a project

Did You Know? Atorus is committed to the continued advancement of open-source technologies in clinical research. We have made investments into contributing to a multitude of consortia including PHUSE, PharmaSUG, CDISC, R/Pharma, R Validation Hub and R Consortium. Additionally, we have developed or collaborated on over 15 open-source clinical programming packages

atorus
Your Complete Data Solution

OpenVal® CHEAT SHEET

learn more at https://www.atorusresearch.com/openval/

Validated by Us,
Ready for You

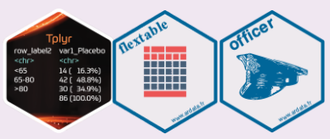
Discovery and Planning

Package selection is driven by **Atorus' industry subject matter experts** and collaboration with our customers

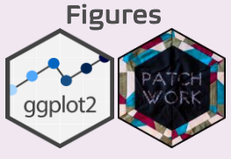
Current packages focus on common statistical programming activities enabling the **GxP use of R** for clinical programming

SDTM/ADaM

Tables



Figures



A snapshot date is determined for each OpenVal® release to create a **frozen point in time** for the environment

Analysis is performed to determine the **effects of changes** in the R version and the R package versions from one OpenVal® release to another

4-5
major/minor R version changes are released annually

75%
of the packages in OpenVal® are updated annually

Risk Assessment

Packages are assessed for their **level of risk** to determine the scope of the testing

Initial considerations for risk are looking at the type of package as well as **metrics** such as # of downloads, # and type of references, and vignettes

Low risk criteria:	**Medium risk criteria:**
• Utility/interactive/ visualization package • High # downloads • Has references • Has vignettes	• Statistics/PK/ machine learning package • Low # downloads • No references • No vignettes
High risk criteria: • Internal packages • Packages not on CRAN/Bioconductor	

Risk is determined based on category in which greatest number of requirements are met. Tie goes to the riskier category

Example logrx			
Utility package	= Low criteria	No references	= Medium criteria
Low # downloads	= Medium criteria	Has vignettes	= Low criteria
Risk = Medium			

A subject matter expert **looks at the package code** to determine the final assigned risk

Some packages may be accepted for validation without additional testing based on their high degree of reliability and **prior testing** within their software development lifecycle

If the risk of a desired package is too high Atorus will make **recommendations for alternative packages**

Testing

The **scope** of testing is determined using information from the previous phases

Requirements, test cases, and test code are written by subject matter experts to verify the package produces an **accurate and/or acceptable result**

- **Requirements** establish what needs to be tested
- **Test cases** describe how the requirements will be tested. They are written in English (i.e "we are testing that the package does x")
- **Test code** is written to verify the test cases

Testing uses Atorus' **automated framework**. This saves the human time of manual execution of tests

Our tests are run across **multiple operating systems**

Subject matter experts select the most applicable **testing scenario** to ensure the highest quality testing

Testing Scenarios	
Verified Output Comparison (preferred)	Comparison is done to known literature values
Expectation Testing	Verification is done that an expected message/warning/ error is returned
Double Program	An independent programmer creates output for comparison
Snapshot Testing	Output is produced then undergoes human verification. Tests are written to regenerate that output and compare

Installation

OpenVal® is set up to ensure **stability**, and to ensure the environment maintains **a validated state**

Installation into a company environment is **simple** consisting of the following steps:

- Install package system dependencies
- Install necessary R version
- Using a provided installation script, install all R packages
- Using a provided test execution script, run the testing to produce the Validation Report

Excerpt from report:

Risk Assessment			
Package	Version	Type	Risk
logrx	0.2.2	Utility	Medium

Validation Assessment			
Package	N Tests	Tests Passed	Tests Failed
logrx	19	19	0

{logrx} 0.2.2 - Medium			
Spec ID	Test ID	Test Description	Test Result
pkg_logrx.4 Logs the start and stop times for a short running program when run with execute (2)	pkg_logrx.4.1	Logs the start and stop times for a short running program	✓
	pkg_logrx.4.2	Logs the start and stop times for a long running program	✓
pkg_logrx.5 Logs the log's output file and path when run with execute (1)	pkg_logrx.5.1	Logs the log's output file and path when run with execute	✓

atorus
Your Complete Data Solution