PhUSE US Connect 2020

Paper AS03

Reproducible Open Source Environments for Advanced Analytics

Sean Lopp, RStudio, Denver, United States

ABSTRACT

The pharmaceutical industry is increasingly using open source tools like R and Python for all phases of drug development including early stage research and late stage clinical trials. This adoption of open source tools leads to a pressing concern and need for reproducible analysis. Unlike legacy tools, reproducibility in the open source ecosystem is more complex. Analysis and platforms must account for many mixed components: operating systems, languages, packages, source code, and data products such as reports and web applications. This paper presents strategies organizations can adopt to build reproducible, open source analysis along with tools necessary to implement these strategies.

INTRODUCTION

Great data science work should be reproducible. Being able to repeat experiments is the foundation of all science. While everyone should have a plan for reproducible environments, here are a few signs to suggest environment management has gone wrong:

- Code that used to run no longer runs, even though the code has not changed.
- You are afraid to upgrade or install a new package, because it might break your code or someone else's.
- Typing install.packages in your environment doesn't do anything, or doesn't do the *right* thing.

There are a variety of strategies for creating reproducible environments. It is important to recognize that not everyone needs the same approach to reproducibility. There are three main strategies covered in this paper:

The Shared Baseline strategy helps administrators coordinate the work of many data scientists by providing common sets of packages to use across projects. The key to this strategy is determining a consistent set of packages that work together.

The Snapshot and Restore strategy is used when individual data scientists are responsible for managing a project and have full access to install packages. The strategy uses tools like renv to record a project's dependencies and restore them.

The Validated strategy is used when packages must be controlled and meet specific organization standards.

SHARED BASELINE

The shared baseline strategy fits when administrators or R champions are responsible for creating an environment where less experienced users can easily share and re-run work. The main goal is to ensure all users accessing a common compute platform, such as RStudio Server, are also accessing the same versions of packages. This is accomplished using fixed versions of R and frozen package repositories. A frozen repository is a way for organizations to always get a consistent set of packages, without having to pre-install all the packages. As an example, you could rsync CRAN to an internal server on January 1st and host it at https://r-pkgs.example.com/cran/012019. Then, no matter when an administrator or user installs new packages, they will always get a consistent set of packages. The next time a version of R is released, the new version of R can be associated with a new frozen repository, e.g. https://r-pkgs.example.com/cran/062019, allowing users to access updated and new packages while still remaining consistent.

SNAPSHOT

The snapshot and restore strategy fits when package access is open and users are responsible for reproducibility. This strategy is the most relevant for individual data scientists. The strategy has two key characteristics:

1. Users are able to freely access and install packages for a project
2. Users have the full responsibility to record the dependencies needed for a project

The strategy is implemented with the following steps:

1. Start a project by creating a project-specific library
2. Install and use packages from the project-specific library
3. Record the state of the library alongside of the code
4. Restore the library when the environment needs to be recreated

The strategy can be implemented using the renv package:

```
install.packages("renv")
renv::init()
renv::snapshot()
renv::status()
# to roll back renv::restore()
```

VALIDATED

The validated strategy is similar to the shared baseline strategy. The main difference is the validated strategy targets teams wishing to restrict access to a particular set of packages and teams wishing to approve or audit changes to the package environment. This strategy is appropriate if you require:

- licensing checks
- tests to ensure accurate package methods
- security audits

The strategy is implemented by creating a custom repository. The easiest approach is to use a purpose built tool such as RStudio Package Manager or an open source package like miniCRAN. In addition to a purpose-built repository, some organizations encapsulate the validated set of packages in a Docker image. Docker images allow you to isolate analysis and capture multiple levels of the data science stack including the operating system, R version, and R packages. Docker images are defined using Dockerfiles, which explicitly describe how the environment is to be created. In the case of the validated strategy, the Dockerfiles should include the steps to install the specific package versions that have been approved. ***It is not sufficient to use a Dockerfiles with just a install.packages*** command, as this approach will result in different package versions being built any time the image is rebuilt:

```
FROM ubuntu:xenial

# install a specific R version
ARG R_VERSION=3.5.2
RUN wget https://cdn.rstudio.com/r/ubuntu-1604/pkgs/r-${R_VERSION}_1_amd64.deb
RUN apt-get-y install gdebi-core
RUN gdebi r-${R_VERSION}_1_amd64.deb

#  install packages from a versioned repo
RUN R -e 'install.packages(..., repo = "https://rpkgs.company.com/frozen/repo/
123")'

# or install packages from an renv lock file
COPY renv.lock ./
RUN R -e 'install.packages("renv", repos = "https://cran.rstudio.com");
renv::restore()'
```

CONCLUSION

There are many tools and strategies available for organizations adoption open source tools while requiring strict reproducibility. The strategies themselves are supported by a variety of open source and commercial components. Organizations should determine which strategy and tools to use based on the goals and skills of their users and IT administrators. These strategies are especially important for the pharmaceutical industry as it works towards health authority submissions based in R.

RECOMMENDED READING

https://environments.rstudio.com

CONTACT INFORMATION

Contact the author at:

Sean Lopp

RStudio
Email: sean@rstudio.com