

# Making your R-based research project portable

## Introduction to the {renv} package

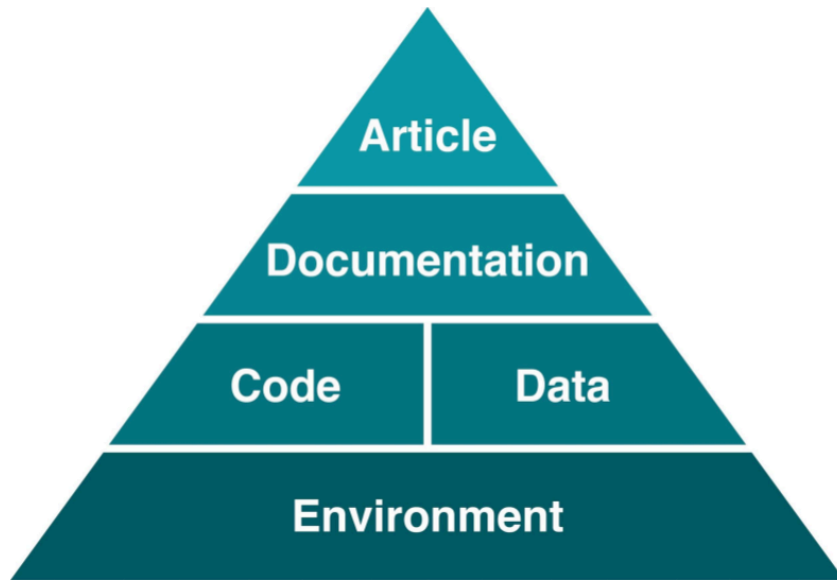
Ernest Guevarra

2024-03-06

# Outline

- All about environments
- System dependencies management
- Project-local R dependencies management

# All about environments



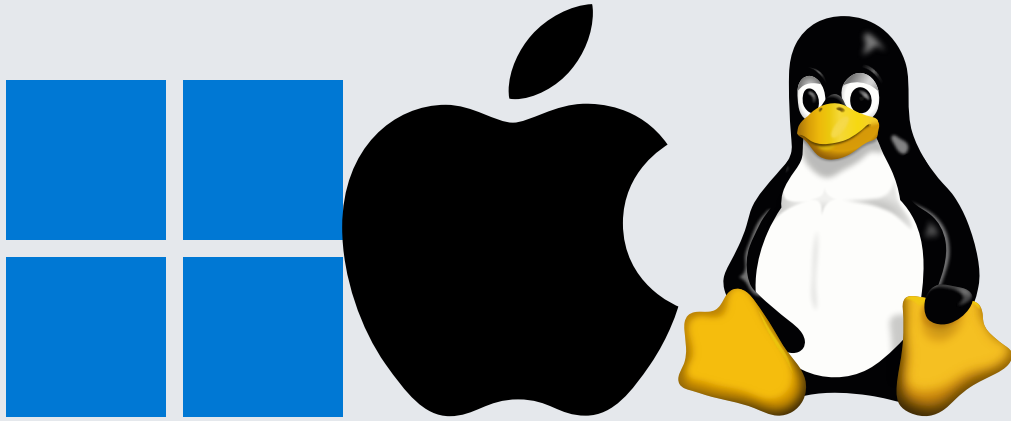
**Portability of an R workflow  
will depend on  
reproducibility of its  
related environments**

**No one cares what operating system you run as long as it stays up.**

**Bruce Perens**

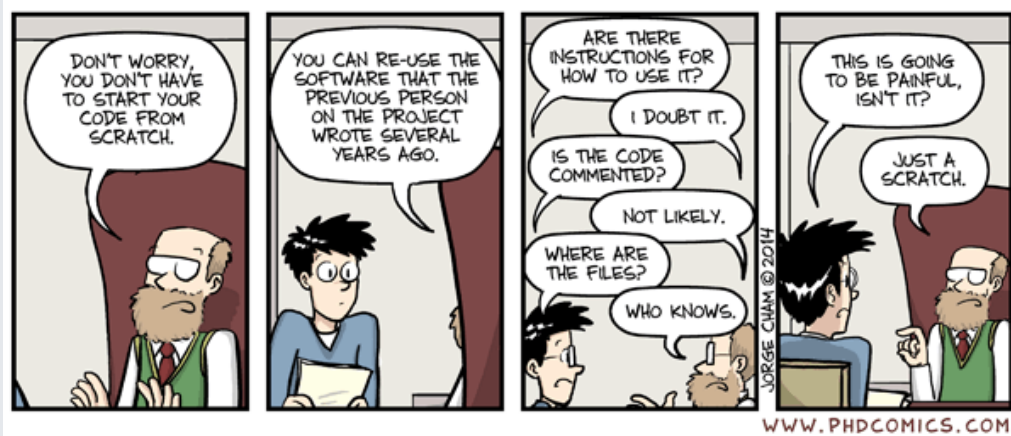
**created the definition of Open Source and wrote the first manifesto of Open Source**

# Different systems, different requirements



- each operating system (and each of its versions) may/will require specific dependencies in order to install R
- each operating system (and each of its versions) may/will require specific dependencies in order to install some R packages

# System dependencies management



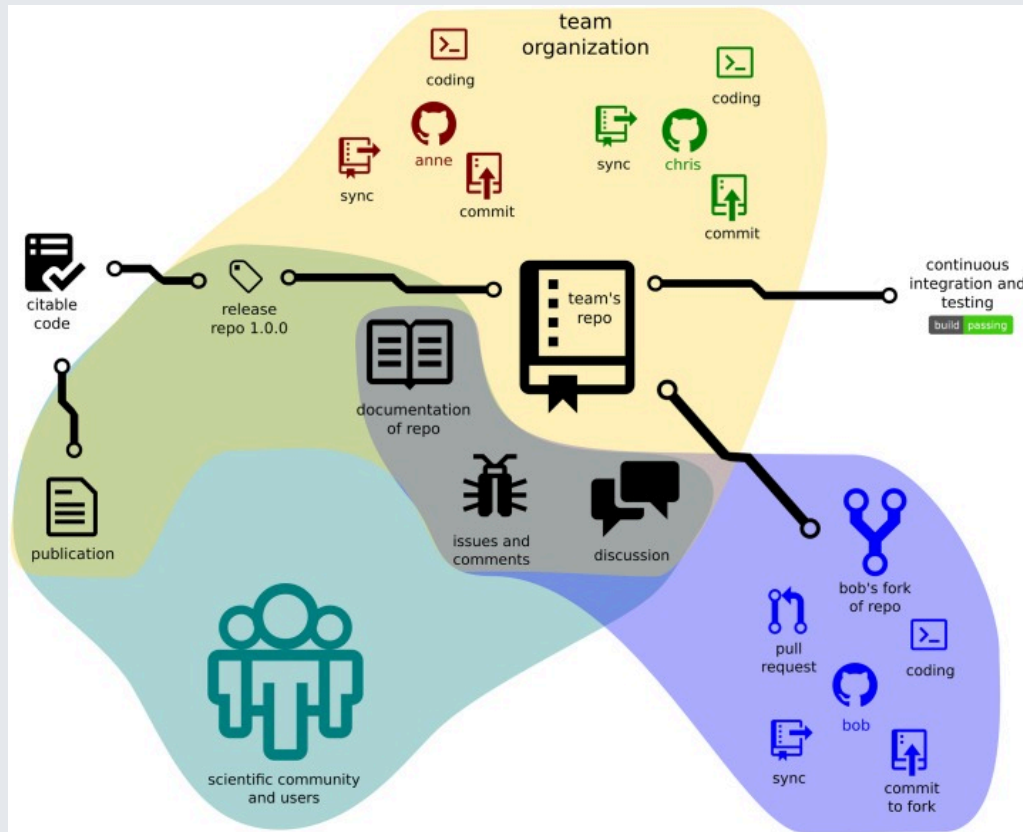
- solutions/approaches will depend on your use case but the most universal step is **documentation**
- **Supervisor-supervisee** or **small research team** settings - establish compatibility guidelines between members and document system requirements for each project;
- **Medium to largish research team** settings
  - use turn-key solutions that standardise R setup using cloud-based or software-as-a-service (SaaS) solutions (e.g., RStudio Cloud); or,
  - setup machines used to be consistent with team's R workflow system requirements through containerisation (e.g., **Docker**)

# Using containers via Docker



- **Containers** are standardised units of packaged software that have everything the software needs to run including libraries, system tools, code, and runtime.
- **Containers** can be deployed either onto remote machines or onto a local machine using **Docker**
- **Rocker** is a community-organised and community-maintained hub of Docker containers that are pre-built and specified with different variants and configurations of R for various operating systems and for various types of R workflows e.g., base, RStudio, spatial analysis, machine learning, etc.
- Containers support portability because we are able to "carry" with us almost any operating system + R configuration that we might need in our R workflows

# Project-local R dependencies management



- The more people collaborate on code and R workflows, the higher the chances that R package dependencies will increase
- The more complex the type of R analysis workflow that is being implemented, the higher the chances that R package dependencies will increase
- Management of R dependencies will be critical in ensuring portability
- The `{renv}` package facilitates this R dependencies management



# The {renv} package



- Initialise a new project-local environment with a private R library;
- Work in the project as normal, installing and removing new R packages as they are needed in the project;
- Save the state of the project library to the lockfile;
- Continue working on your project, installing and updating R packages as needed; and,
- Again save the state of your project library if your attempts to update R packages were successful, or restore to a previous state as encoded in the lockfile if you encounter problems updating a package.

Questions?

# Thank you!

Slides can be viewed at <https://oxford-ihtm.io/open-reproducible-science/session10.html>

PDF version of slides can be downloaded at <https://oxford-ihtm.io/open-reproducible-science/pdf/session10-portable-r-projects.pdf>

R scripts for slides available [here](#)