Computational Reproduciblity

Amrit Singh

2021-02-26

# 1 Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc’s Markdown supports, e.g., a math equation .

The **bookdown** package can be installed from CRAN or Github:

install.packages("bookdown")  
# or the development version  
# devtools::install\_github("rstudio/bookdown")

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading #.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.org/tinytex/>.

# 2 Introduction

## 2.1 Introduction

## 2.2 Advanced

* reproducibility crisis

# 3 Literature

Here is a review of existing methods.

# 4 Containerization

## 4.1 Introduction

A common problem observed during data analysis includes packages/libraries that are not compatible with specific R/Python software. Other times different project use specific versions of packages and the underlying R/Python versions. This leads to “dependency hell” which can be easily solved using virtual environments. A virtual environment is common to isolate the dependency requirements for specific projects which may rely on a specific R/Python version as well as specific versions of the software libraries.

### 4.1.1 R

renv, packrat

### 4.1.2 Python

Conda is useful package manager and can create virtual environments for pretty much anay programming language.[[1]](#footnote-30)

Similar to how shipping containers carry material goods in an enclosed space, software containers consist of application related goods (*e.g.* files and dependencies) in an enclosed environment. Containerization ensure reproducibility by bundling the necessary resources required to run a given set of analyses, web application, pipelines *etc.* irrespective of the operation system (OS). Thus, containers can be run on any OS (*e.g.* Mac, Windows, Linux)[[2]](#footnote-32).

Glossary:

## 4.2 Advanced

# 5 Version Control

## 5.1 Introduction

## 5.2 Advanced

# 6 Pipelines

## 6.1 Introduction

## 6.2 Advanced

# 7 Continuous Integration and Continuous Delivery

## 7.1 Introduction

## 7.2 Advanced

# 8 Final Words

We have finished a nice book.

1. <https://docs.conda.io/en/latest/> [↑](#footnote-ref-30)
2. <https://stackoverflow.com/questions/50974960/whats-the-difference-between-docker-and-python-virtualenv> [↑](#footnote-ref-32)