Sharing computational environments with Binder

Leon Reteig (@LeonReteig), UPOD DataScience meeting, 24 July 2020

Reproducible computational environments

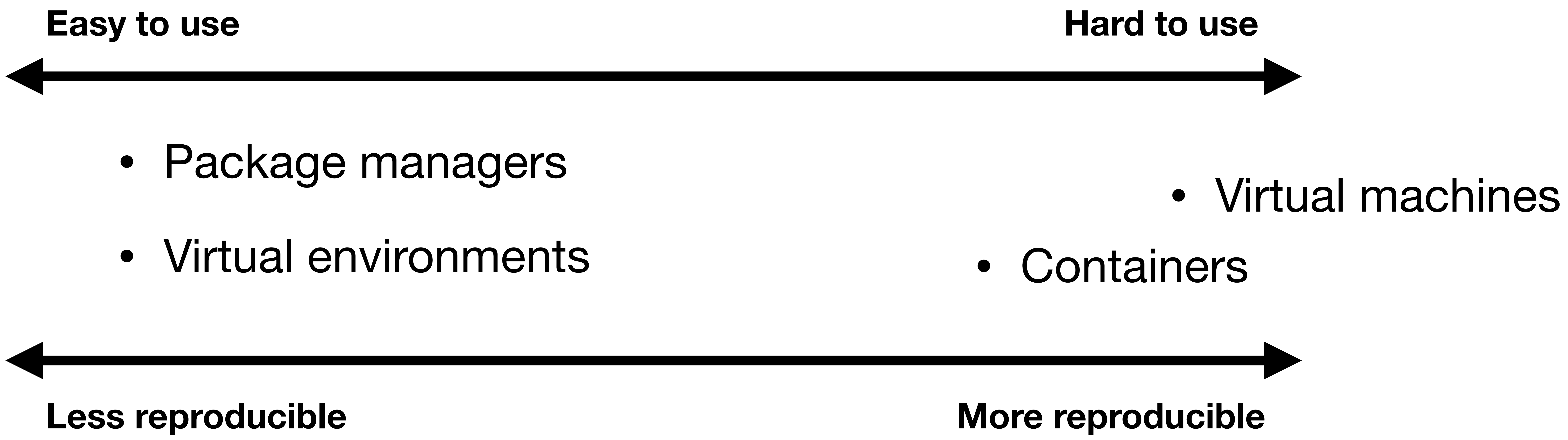
Code is not enough

You’ve written some code and wish to share it with others. What *else* do they need to actually run it?

- The programming language your code was written in
- Its dependencies (functions from other packages your code calls)
- Possibly an IDE (Jupyter, RStudio) to interact with your code in a certain way

Multiple solutions

The reproducibility spectrum



Package managers

- When you've *written a package*, specify its dependencies in a configuration file

  `DESCRIPTION`

  `setup.py`

- When users install the package, the dependencies are installed as well

Virtual environments

- When you're *developing* code, do so in an isolated environment

  `renv`

  `virtualenv, conda`

- Install new / remove old dependencies as necessary

- Write the environment description to a configuration file; restore the environment from said file

  lockfile `(renv.lock)`

  `requirements.txt, environment.yml`

Virtual machines

- Simulate *an entire computer system*
    - *Operating system*
    - *File system*
    - *Standard software*
    - *Etc.*
- Share an *image* of this system

VirtualBox

Containers

- Isolated environments that share some resources amongst each other / with the host system
- Less isolated than a full virtual machine, more lightweight

Singularity

What’s wrong with these?!

Package managers
Virtual environments

- Don’t *completely* capture the environment (e.g. operating system)
- **Leave the work up to the user:**
  - Install conda
  - Install Jupyter
  - Figure out how to recreate the environment

Virtual machines
Containers

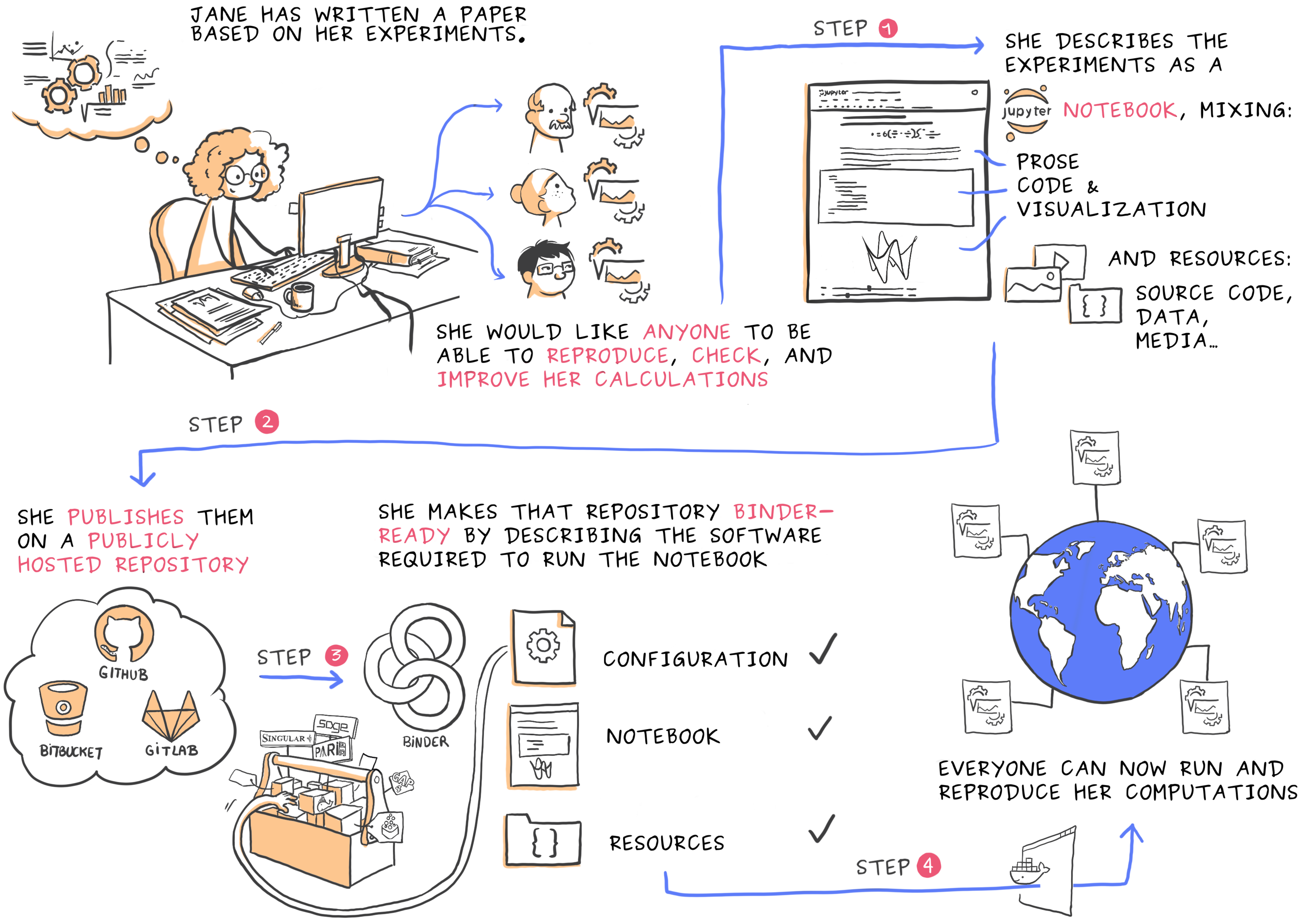
- Overkill for simple projects
- Not so easy to use
- **Leave the work up to the user:**
  - Figure out how to open the container and interact with it

Enter… Binder

A happy medium

- Makes a Docker image out of a publicly hosted repository
    - Based on a dockerfile
    - Based on only the environment config files (e.g. `requirements.txt`) that you probably already have in some form!
- Spins up a cloud instance of the container that anyone you give the URL to can interact with through their browser, using:
    - Jupyter Notebook (default)
    - JupyterLab
    - RStudio server

binder



JANE HAS WRITTEN A PAPER BASED ON HER EXPERIMENTS.

SHE WOULD LIKE ANYONE TO BE ABLE TO REPRODUCE, CHECK, AND IMPROVE HER CALCULATIONS

STEP 1

SHE DESCRIBES THE EXPERIMENTS AS A jupyter NOTEBOOK, MIXING:

PROSE
CODE &
VISUALIZATION

AND RESOURCES:

SOURCE CODE,
DATA,
MEDIA...

STEP 2

SHE PUBLISHES THEM ON A PUBLICLY HOSTED REPOSITORY

STEP 3

SHE MAKES THAT REPOSITORY BINDER-READY BY DESCRIBING THE SOFTWARE REQUIRED TO RUN THE NOTEBOOK

CONFIGURATION ✓

NOTEBOOK ✓

RESOURCES ✓

STEP 4

EVERYONE CAN NOW RUN AND REPRODUCE HER COMPUTATIONS

https://opendreamkit.org/2017/11/02/use-case-publishing-reproducible-notebooks/

Binder demo 1

A toy example with Python

http://bit.ly/zero-to-binder-python

Binder demo 2

A real-world example with R

https://github.com/lcreteig/sacc-tDCS

Use cases

Sharing all analyses for a paper (or the paper itself!)

- e.g. https://mybinder.org/v2/gh/annakrystalli/rrcompendiumDTB/master?urlpath=rstudio

Interactive package documentation

- e.g. scikit-learn docs: https://mybinder.org/v2/gh/scikit-learn/scikit-learn/0.23.X?urlpath=lab/tree/notebooks/auto_examples/feature_selection/plot_rfe_digits.ipynb

Teaching / live demos:

- Making sure everyone has the same environment and can follow along

Limitations

- Limited time / resources
  - Max 2 GB ram (min 1 GB)
  - <12 hours computation time (shuts down after 10 min of inactivity)
- Everything has to be public:
  - Repo with code
  - Data…

Resources

- Binder (with link to documentation): mybinder.org
- Minimal binder tutorials:
    - Julia: http://bit.ly/zero-to-binder-julia
    - Python: http://bit.ly/zero-to-binder-python
    - R: http://bit.ly/zero-to-binder-r
- Chapter from “The Turing Way” Guide for Reproducible Research: https://the-turing-way.netlify.app/reproducible-research/renv.html