Data Science & Research Reproducibility

Sang-Yun Oh

Department of Statistics and Applied Probability

UC SANTA BARBARA

What is Computational Reproducibility?

- Reviewable Research
  The descriptions of the research methods can be independently assessed

- Replicable Research
  Data, code, and/or software tools are made available to duplicate the research result (may not be public)

- Confirmable Research
  The main conclusions can be obtained independently using the description of algorithms and methodology provided in the publication.

(Stodden et al., 2013)

What is Computational Reproducibility?

- Auditable Research
  Sufficient records, including data and software, have been archived (potentially privately) so that the research can be defended later if necessary or differences between independent confirmations resolved

- Open or Reproducible Research
  Auditable research made openly available, so that one may (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

(Stodden et al., 2013)

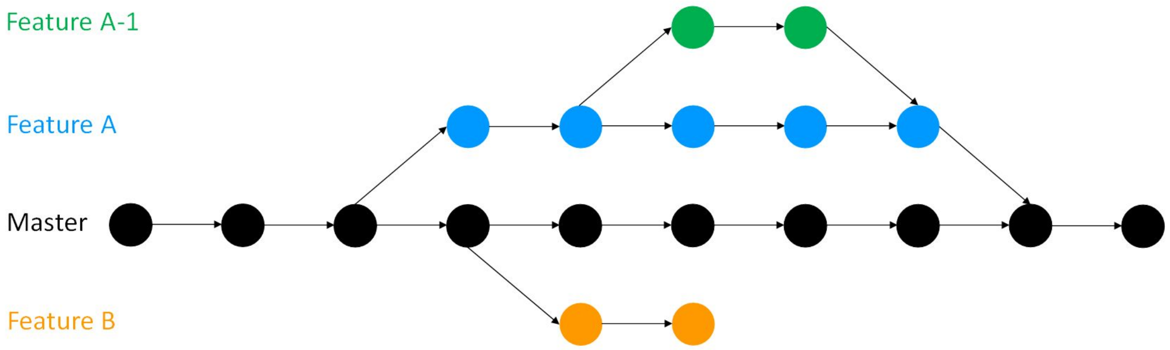
Ingredients for Computational Reproducibility

- Data (real or simulated)
- Algorithm implementation
- Analysis pipeline
  Preprocessing, analysis, post-processing, external validation, etc.
- Generated report (figures and tables)

Ingredients for Computational Reproducibility

- Data (real or simulated)
- Algorithm implementation
- Analysis pipeline
  Preprocessing, analysis, post-processing, external validation, etc.
- Generated report (figures and tables)
- Computational environment
  Operating system, R/Python versions, package versions

Version Control Everything and Test End-to-End

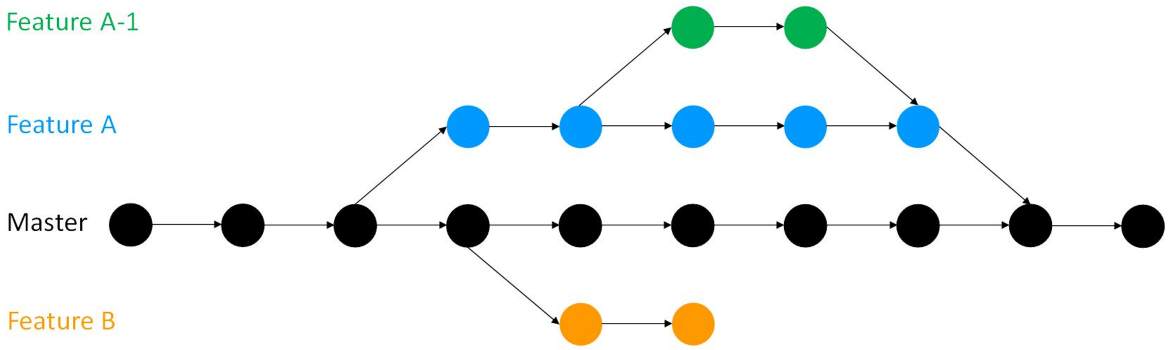


[The Turing Way]

- Computational environment
- Algorithm implementation
- Analysis pipeline

UC SANTA BARBARA

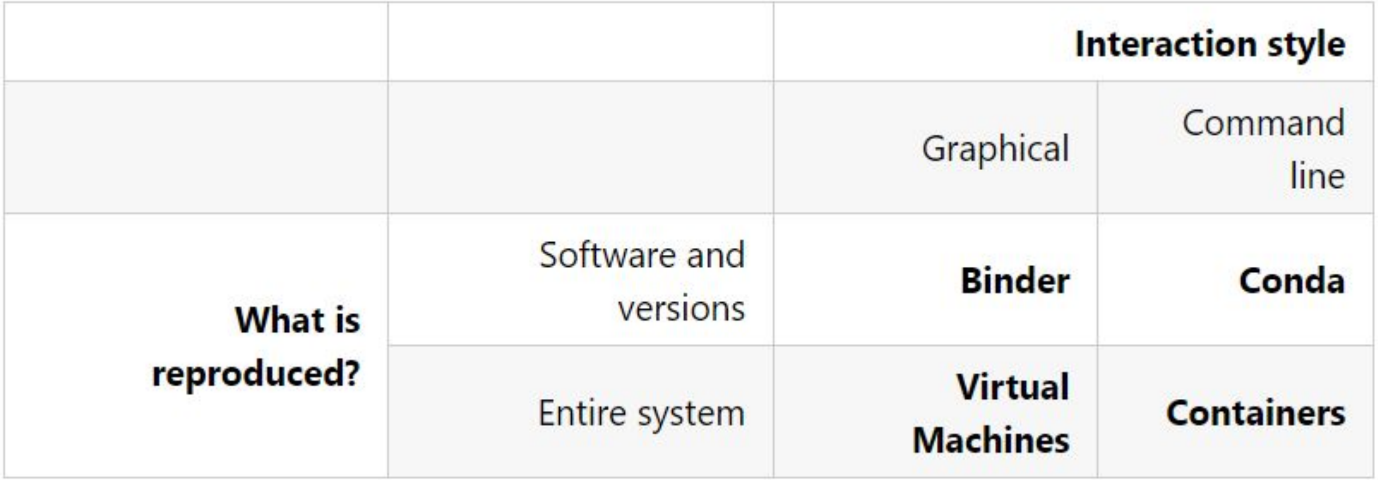
Version Control Everything and Test End-to-End



The Turing Way

- Computational environment (scriptable software installations)
- Algorithm implementation (R/Python/other source code)
- Analysis pipeline (glue scripts: Make, Python, etc.)

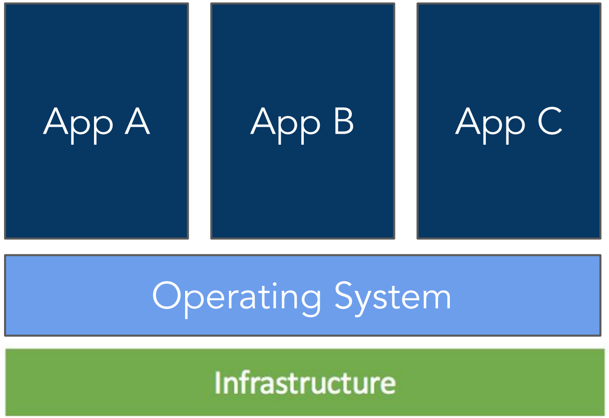
Different Level of Reproducibility



		Interaction style	
		Graphical	Command line
What is reproduced?	Software and versions	**Binder**	**Conda**
	Entire system	**Virtual Machines**	**Containers**

The Turing Way

Personal Computers



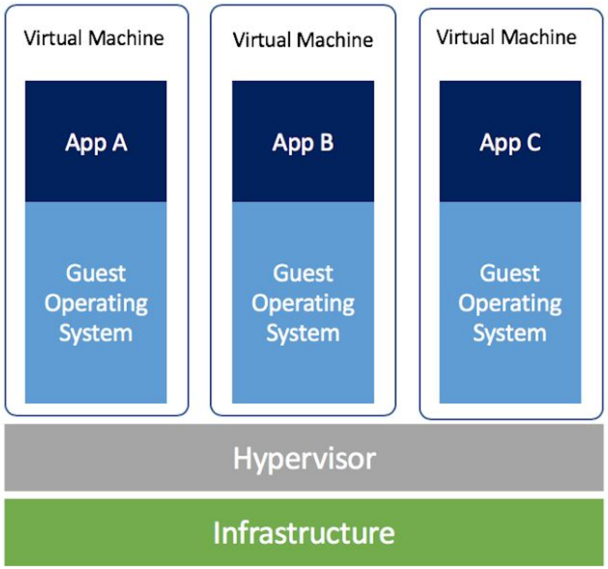
1. Physical hardware infrastructure
2. Download and Install OS
3. Install applications (scriptable in Linux)
4. Run application
5. Problems: interaction between Apps, compatibility with OS and infrastructure

Package Management System (Conda)

- Conda is open source cross-platform package management system
- Keep track of packages and dependencies
- Multiple environments can coexist
- Package availability depends on platform:

  https://anaconda.org/search?q=gcc

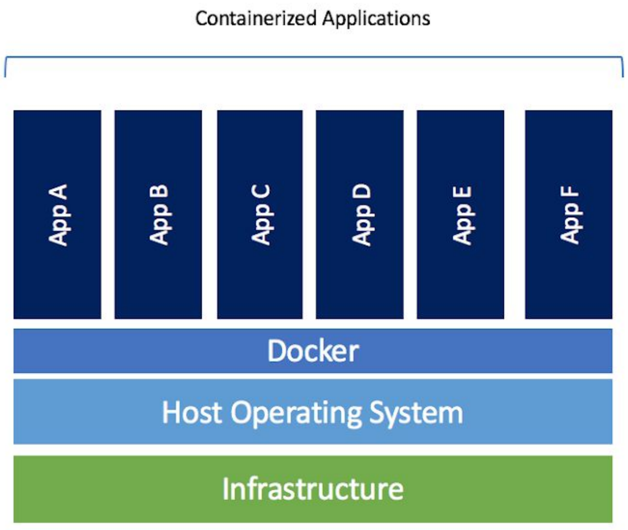
Virtual Machines (VM)



Software emulation of a physical computer

1. One OS + One App = No problems?
2. How to communicate between VMs?
3. How to share storage between VMs?
4. Replicated OS seems wasteful

Docker



Docker isolates applications under one OS

1. Install Docker
2. Download or build application image
3. Run application in a container
4. One container instance is similar to VM
5. Storage can be shared through Host OS

Image Source

Docker vs. VM

- Docker is lighter than VMs

  VM emulate a full computer and installs a full OS

- Docker is OS dependent, VM is OS independent

  Not an issue for centrally managed environments

- VM and Docker applications can co-exist

Binder

- Binder runs on Jupyter framework
- Launches a Docker image around a repository
- Docker image is created from repo2docker
- https://mybinder.org runs Docker image for free
- Many types of environments are possible

Jupyter Hub for Data Science Courses at UCSB

- Currently running for 10 courses per quarter at UCSB
  - ~5 classes using R / Rstudio
  - ~5 classes using Python Notebooks
- Serve approximately 1000+ students per quarter
- Students can access to their from any web browser
- Can easily give computer exams

Reproducibility Platforms

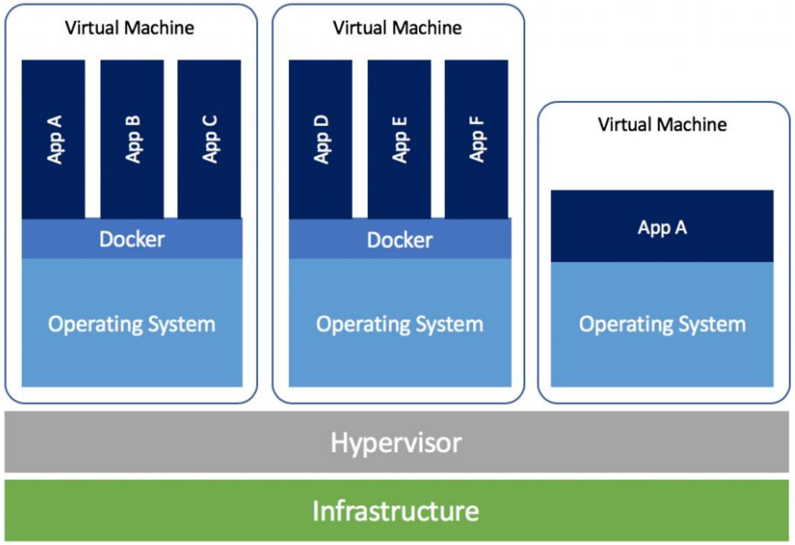
- CodeOcean [not free]
  Jupyterlab with additional features, hardware, and collaboration
  Nature Publishing, EBSCO

- WholeTale [free for now]
  NSF funded platform development
  Similar to CodeOcean

- CodeOcean and WholeTale exports environment specifications

- Built on Docker and Jupyter framework

Everyday Reproducible Research Computing



Setup 1

- One docker app for one project/student: e.g., project, teaching, grad student

Setup 2

- Each grad student gets one VM instance
- Self-manage multiple docker apps e.g. 1 for research 1 for teaching

Image Source

UC SANTA BARBARA