

Teaching and tutorials that I used to build this tutorial



Julian Pistorius
Sanjana Sudarshan

https://github.com/sanjanasudarshan/container_camp_workshop_2019/blob/master/docker/dockerintro.rst

Derek Powell <http://www.derekmpowell.com/posts/2018/02/docker-tutorial-2/>

Feedback



Pink sticky note

Place on the top of your computer when you need help or have a question.

~ At the end of the lesson, write one thing that could be improved ~

Green sticky note

Place on the top of your computer to indicate that you've completed a task and you don't have any questions.

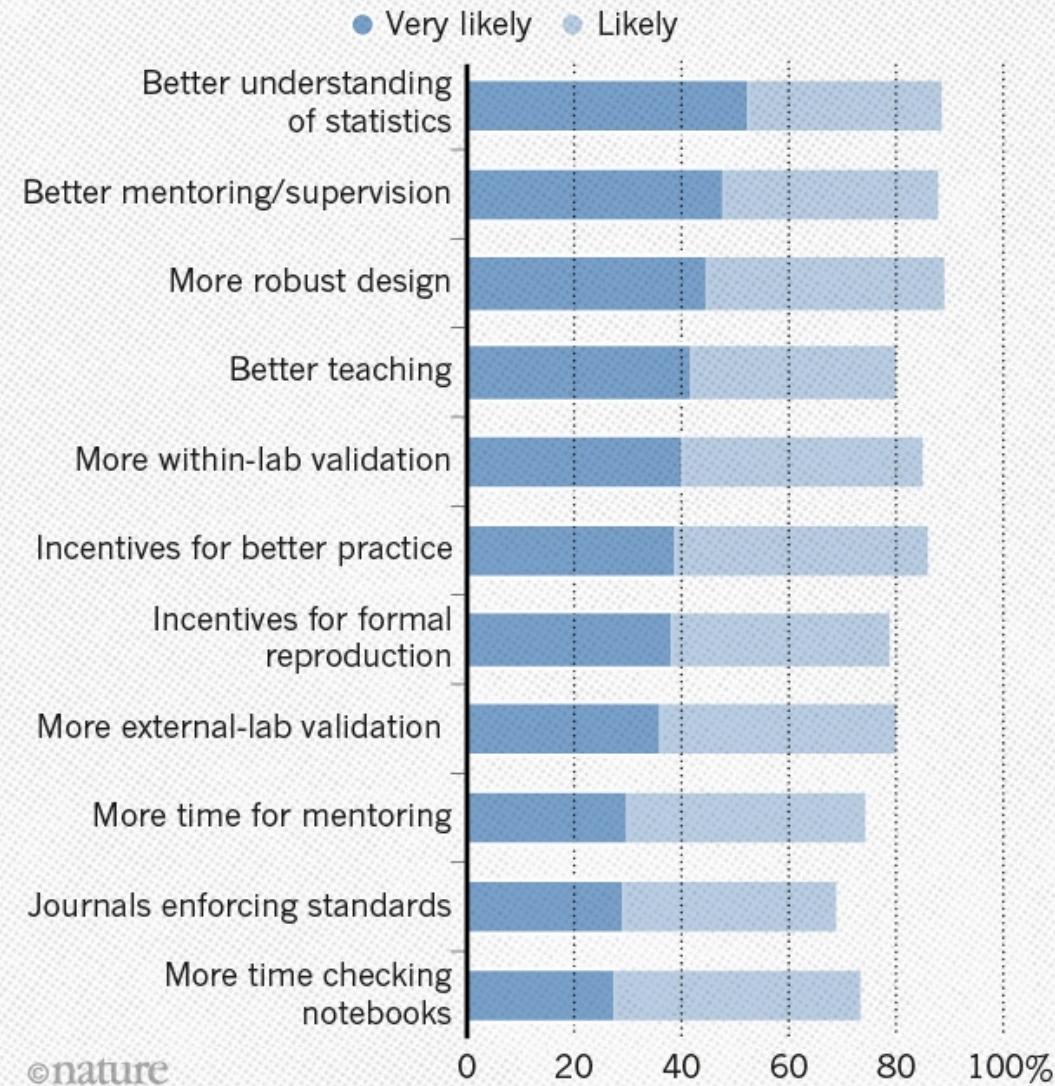
~ At the end of the lesson, write one thing that worked for you ~

IS THERE A REPRODUCIBILITY CRISIS?



WHAT FACTORS COULD BOOST REPRODUCIBILITY?

Respondents were positive about most proposed improvements but emphasized training in particular.



Write Code

Navigate tabs Open in new window Save Find and replace Compile as notebook Run selected code

```
# Good start....  
# Cursors of shared users  
# Re-run previous code  
# Source with or without Echo  
# Show file outline  
# Multiple cursors/column selection with Alt + mouse drag.  
# Code diagnostics that appear in the margin. Hover over diagnostic symbols for details.  
# Syntax highlighting based on your file's extension  
# Tab completion to finish function names, file paths, arguments, and more.  
# Multi-language code snippets to quickly use common blocks of code.  
# Jump to function in file  
# Change file type  
# Working Directory  
# Press ↑ to see command history  
# Maximize, minimize panes  
# Drag pane boundaries
```

R Support

Import data file with wizard History of past commands to run/add to source Display .RPres slideshows
File > New File > R Presentation

Environment History Build Git Presentation

Import Dataset Global Environment Load workspace Save workspace Delete all saved objects Search inside environment Choose environment to display from list of parent environments Display objects as list or grid

Data Values Functions

iris 150 obs. of 5 variables
a 1
foo function (x)

Displays saved objects by type with short description View in data viewer View function source code

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Home IDEcheatsheet

Name

Create folder Upload file Delete file Rename file

Copy... Move... Export... Set As Working Directory Go To Working Directory

Change directory

Path to displayed directory

450 B Dec 24, 2015, 8:55 AM

A File browser keyed to your working directory. Click on file or directory name to open.

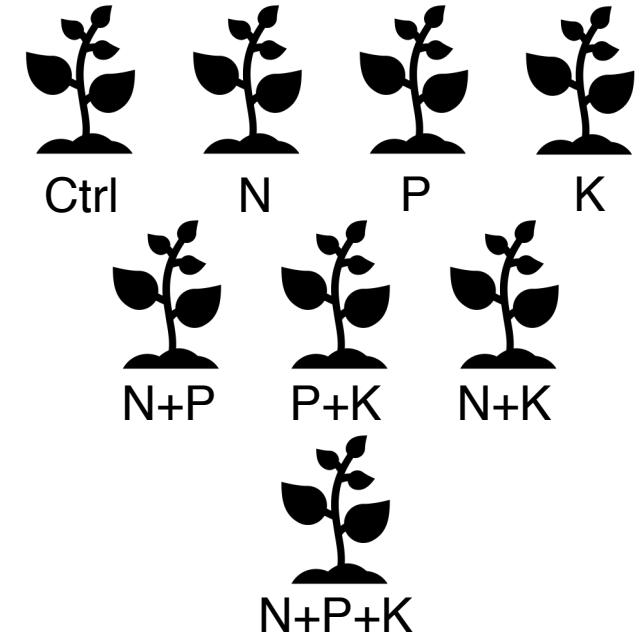
RStudio IDE Cheat Sheet

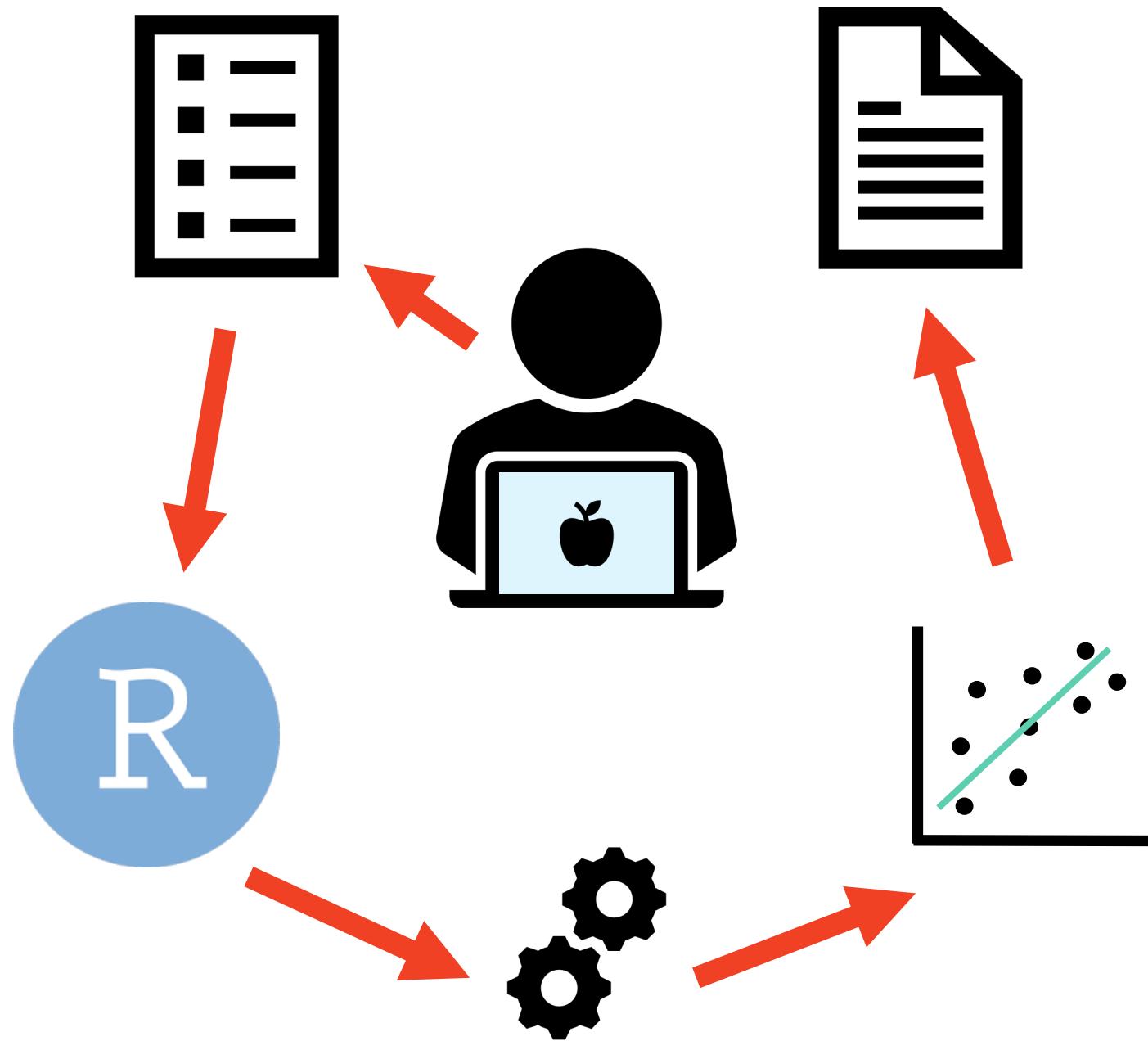
learn more at www.rstudio.com

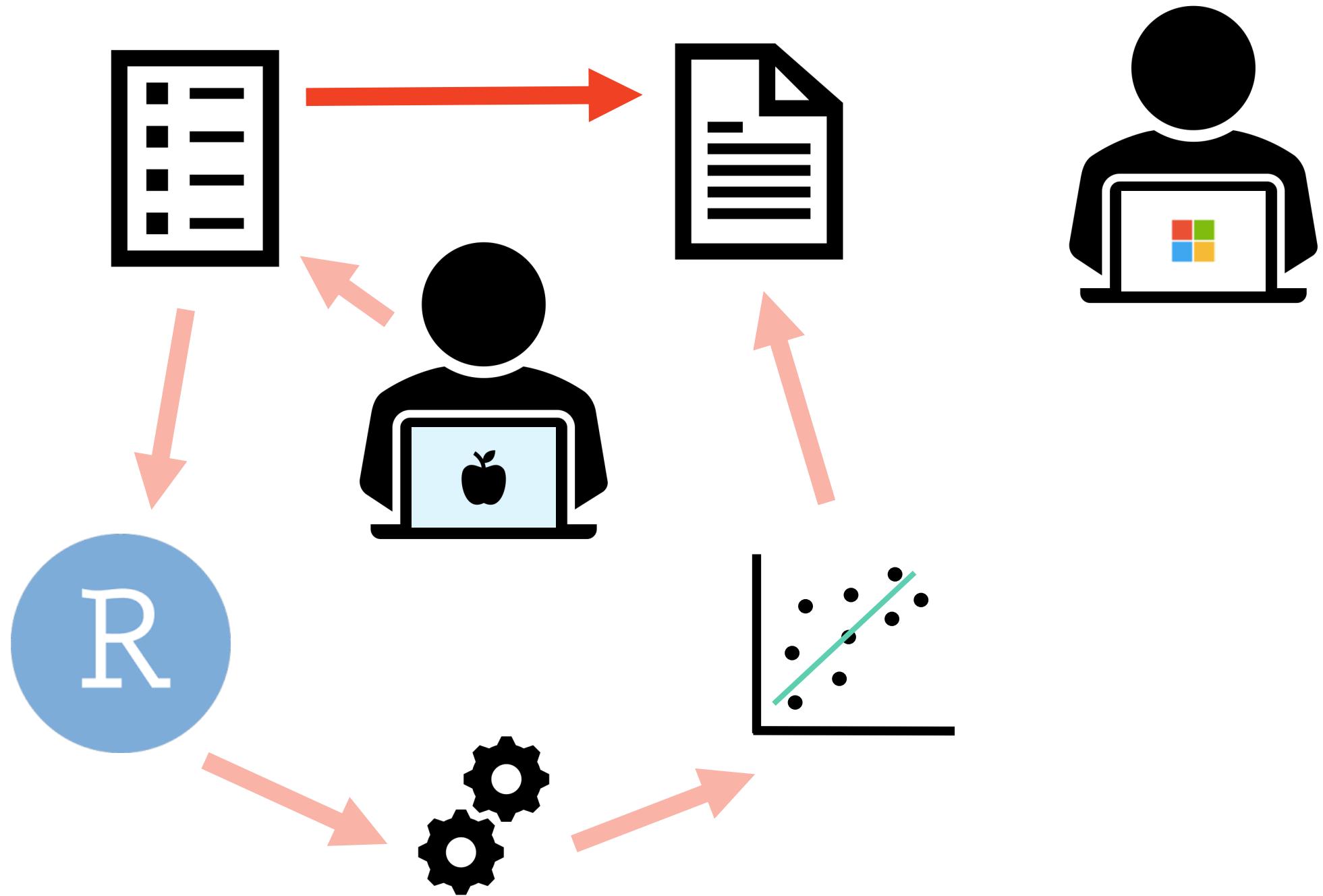


Let's analyze our data!

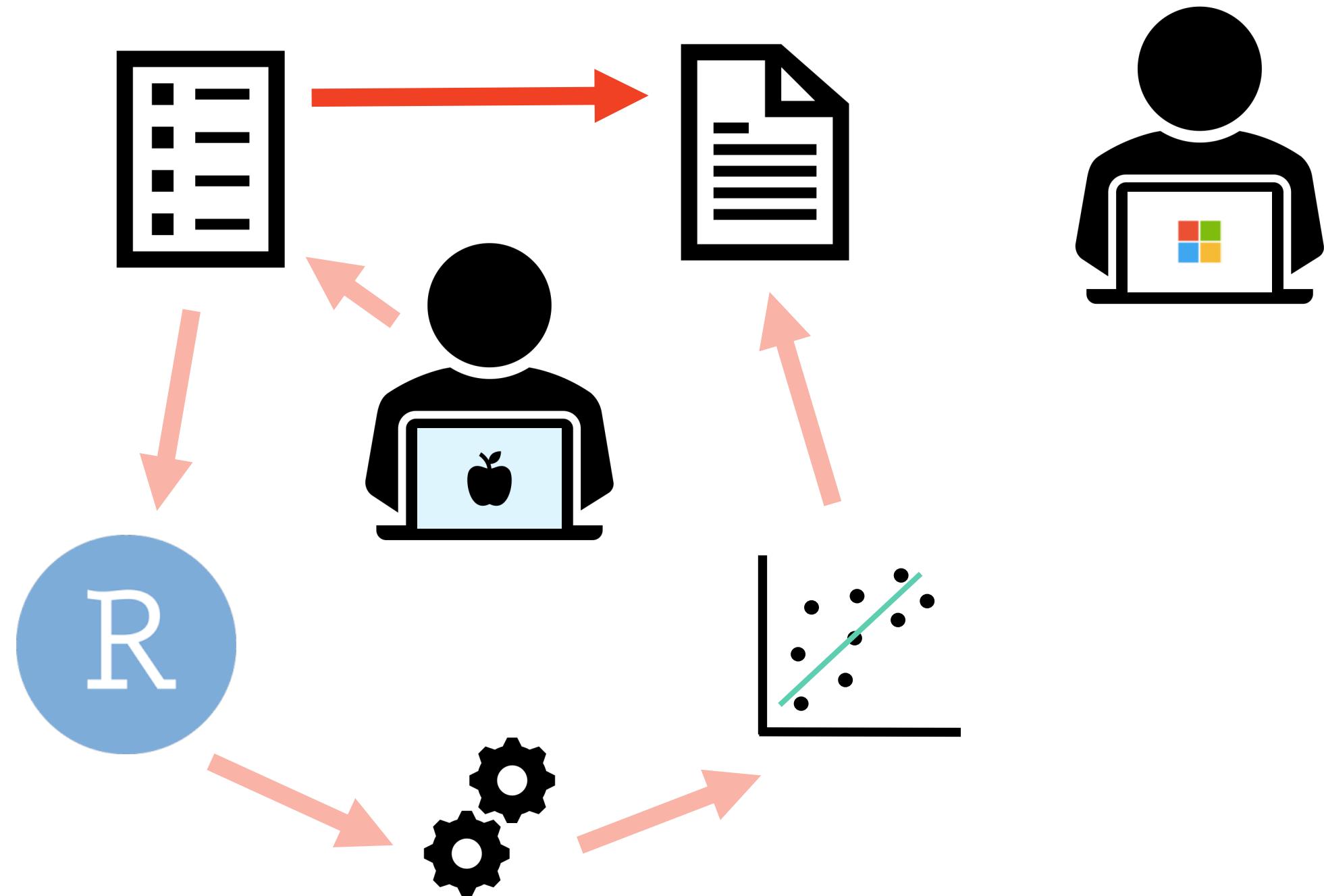
1. Create a new folder “docker-rstudio-tutorial”
2. Create a new R script in RStudio
3. Navigate to tinyurl.com/npk-script
4. Copy the entire page
5. Paste into the new R script
6. Save the script in “docker-rstudio-tutorial” as myScript.R
7. Run the script and examine the output

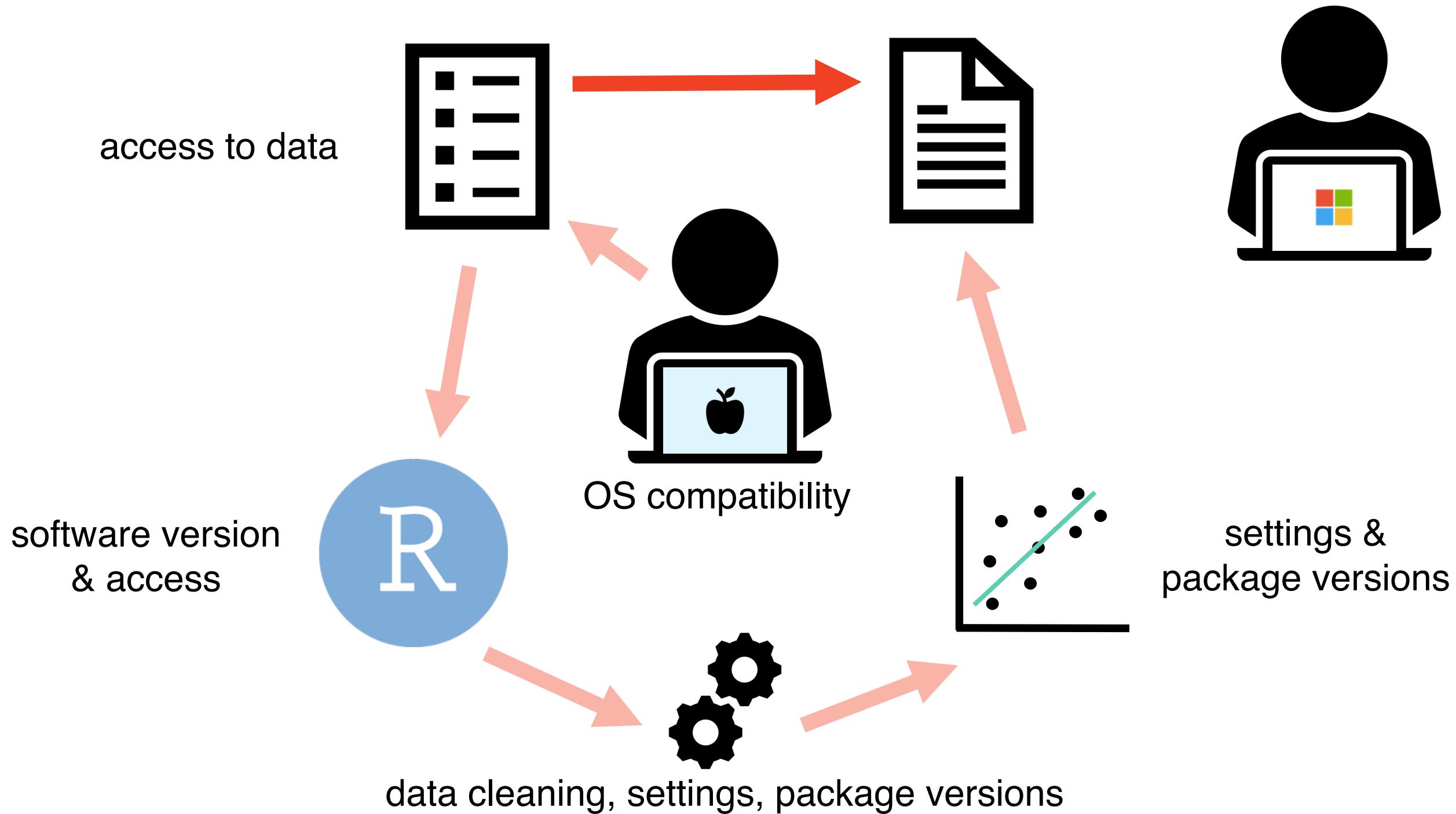




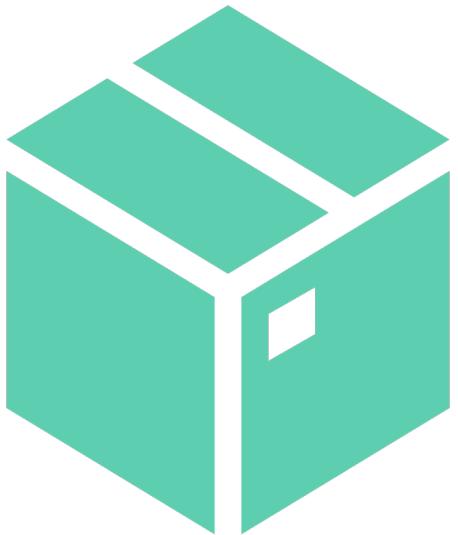


What are
some
barriers to
reproducing
results?



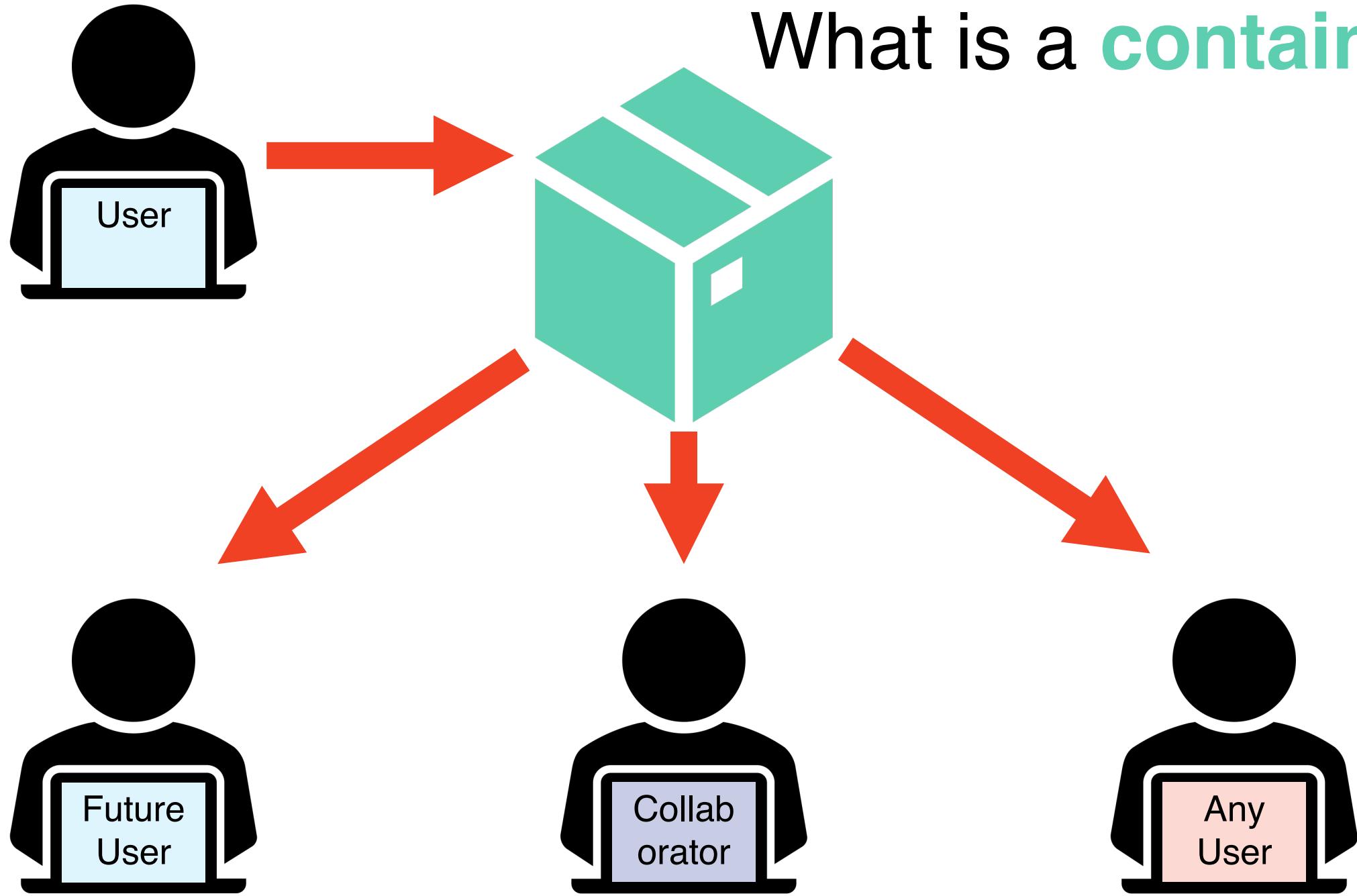


One piece
of the
solution:



a **container**

What is a container?

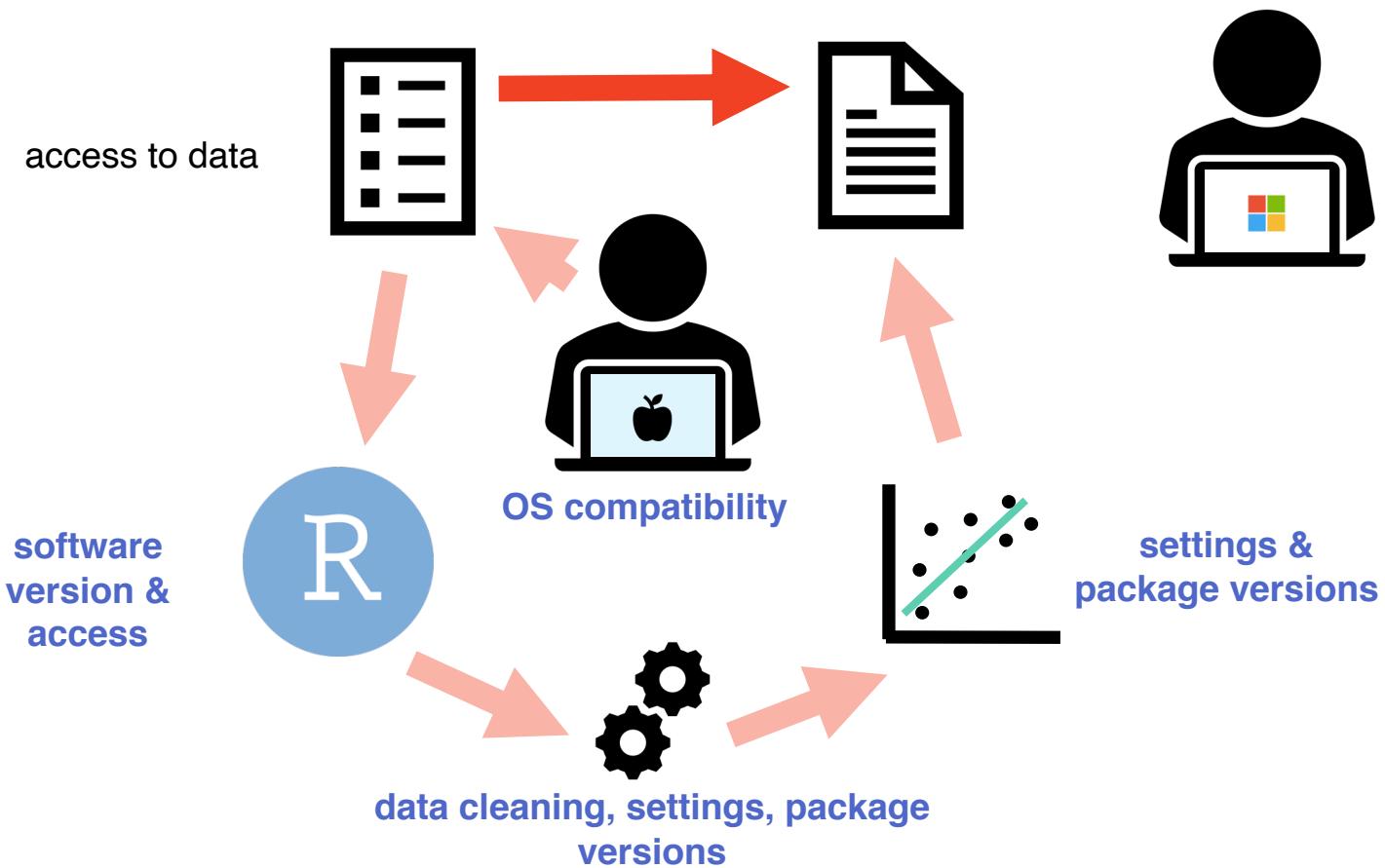


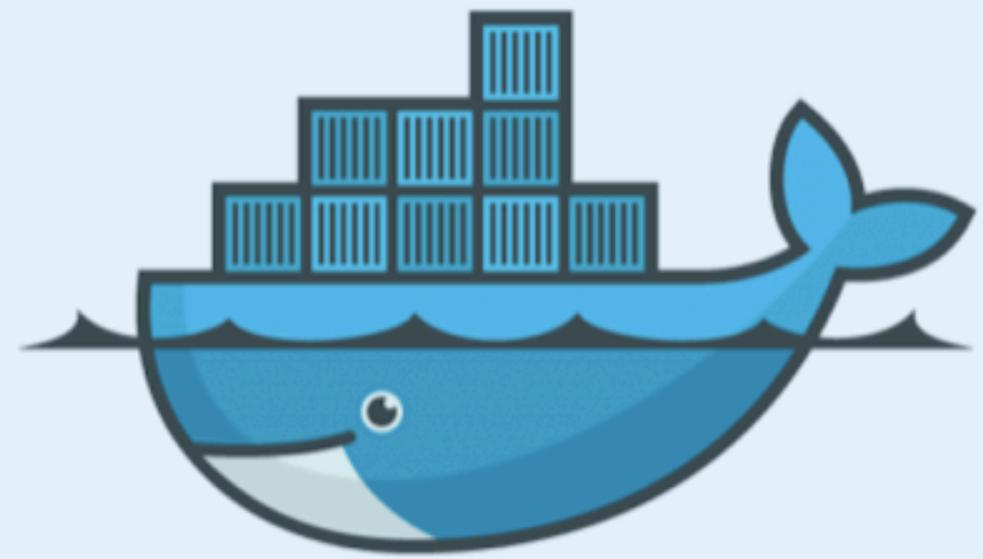
Using containers for reproducible research

1. Write instructions for reproducing your computing environment
2. Compile the pieces
3. Make it public
4. Others can reproduce your computing environment and pair this with your data and code

Using containers for reproducible research

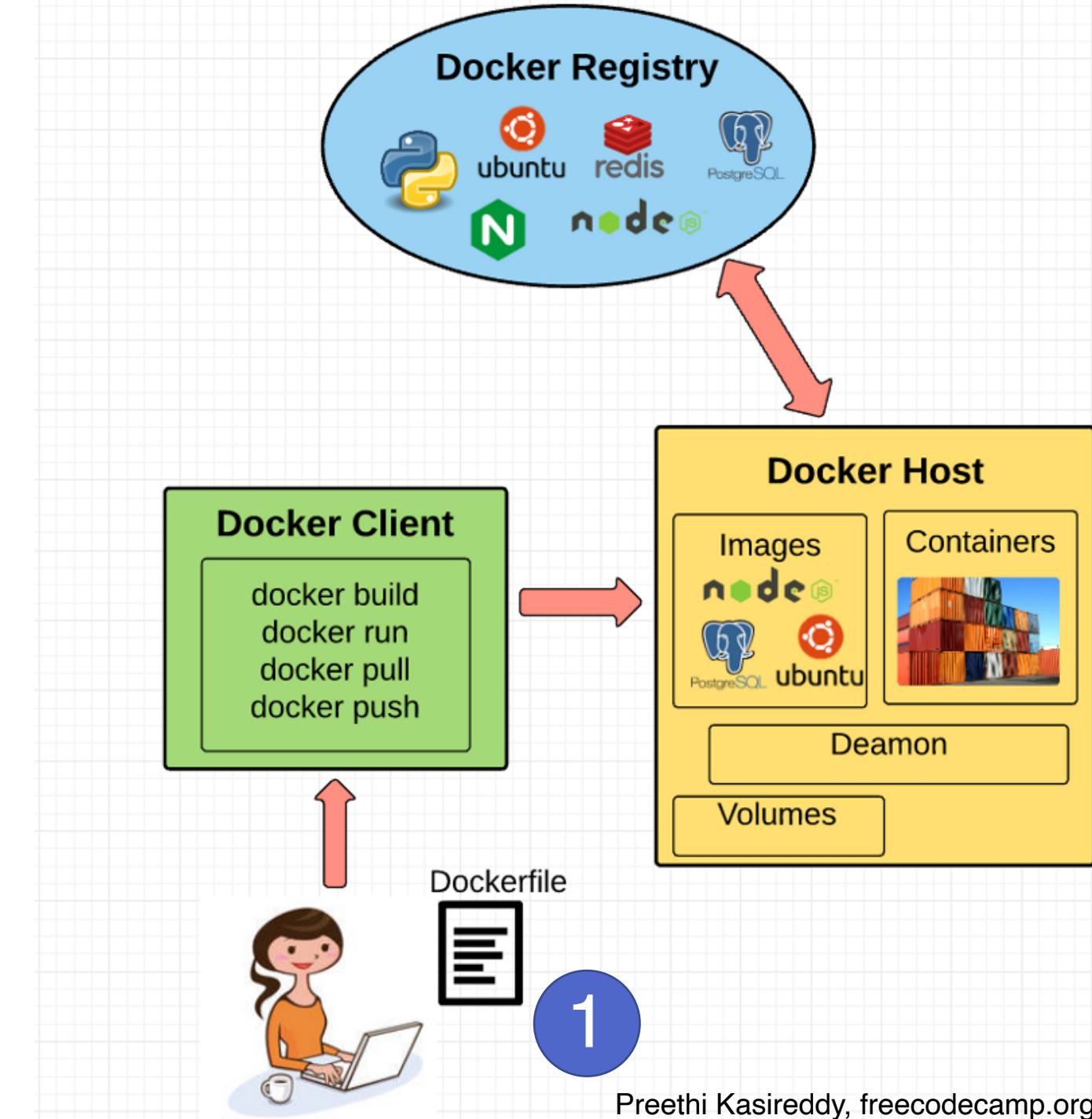
1. Write instructions for reproducing your computing environment
2. Compile the pieces
3. Make it public
4. Others can reproduce your computing environment and pair this with your data and code



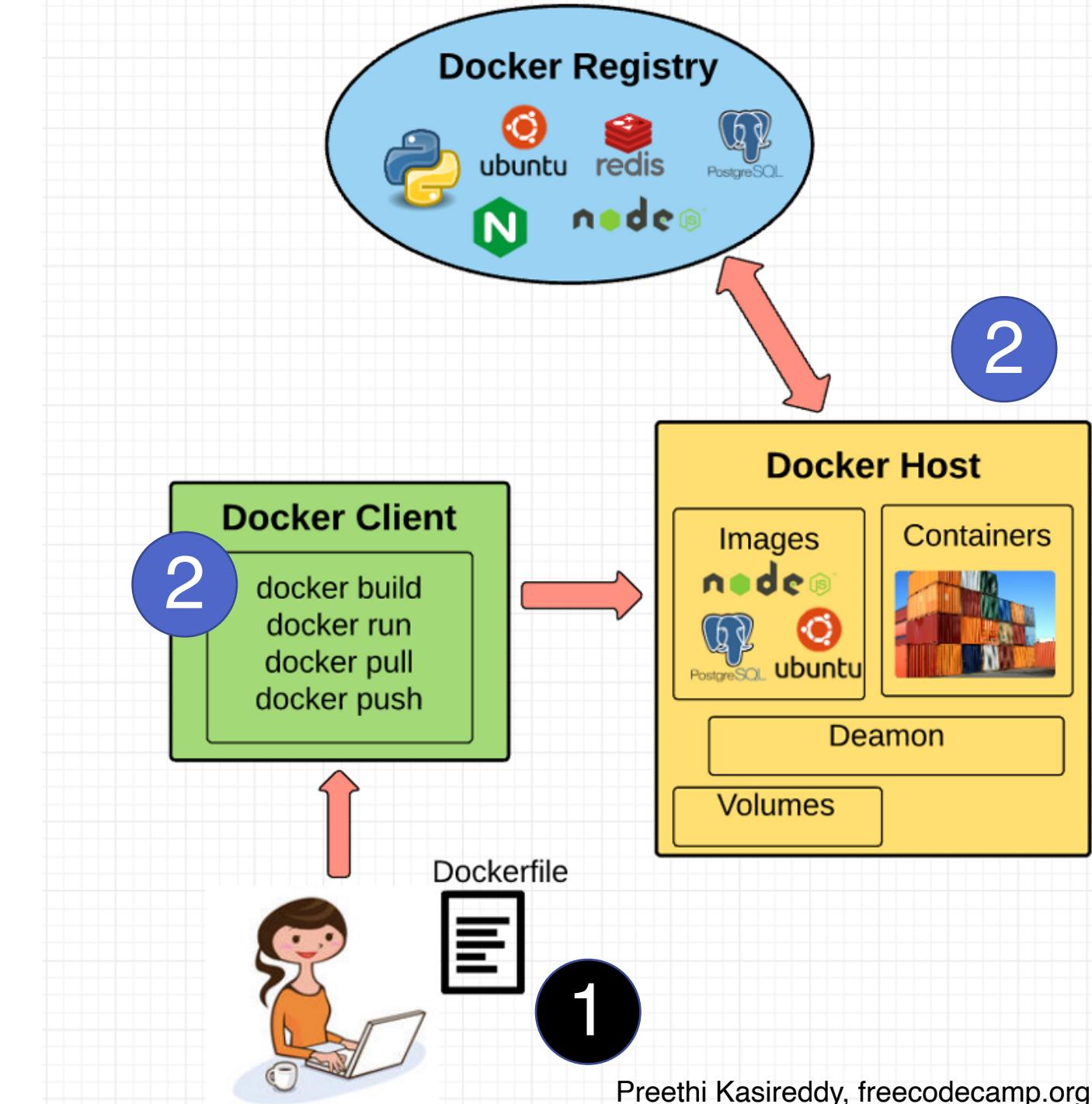


docker

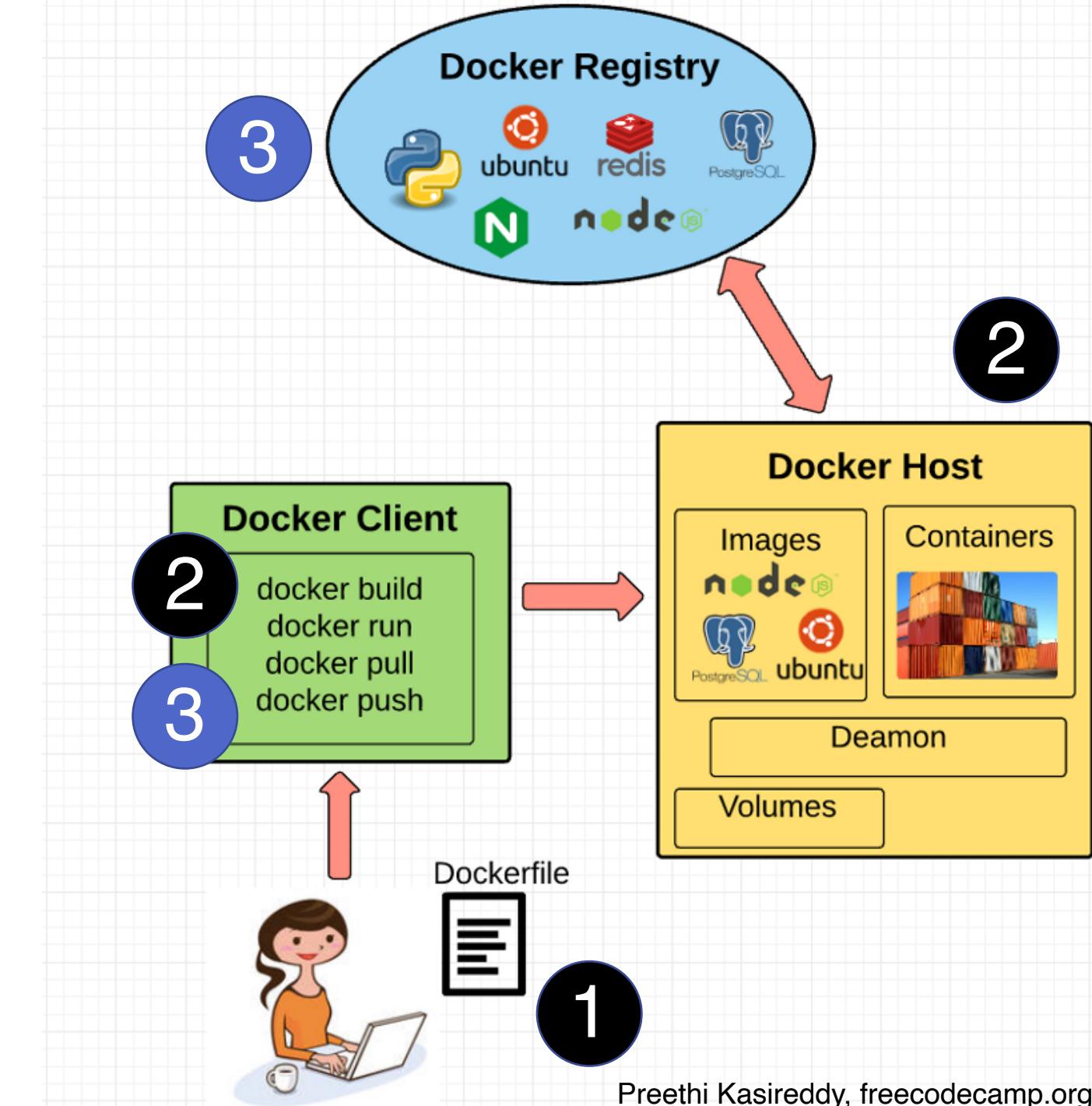
1. Write instructions for reproducing your computing environment
2. Compile the pieces
3. Make it public
4. Others can reproduce your computing environment and pair this with your data and code



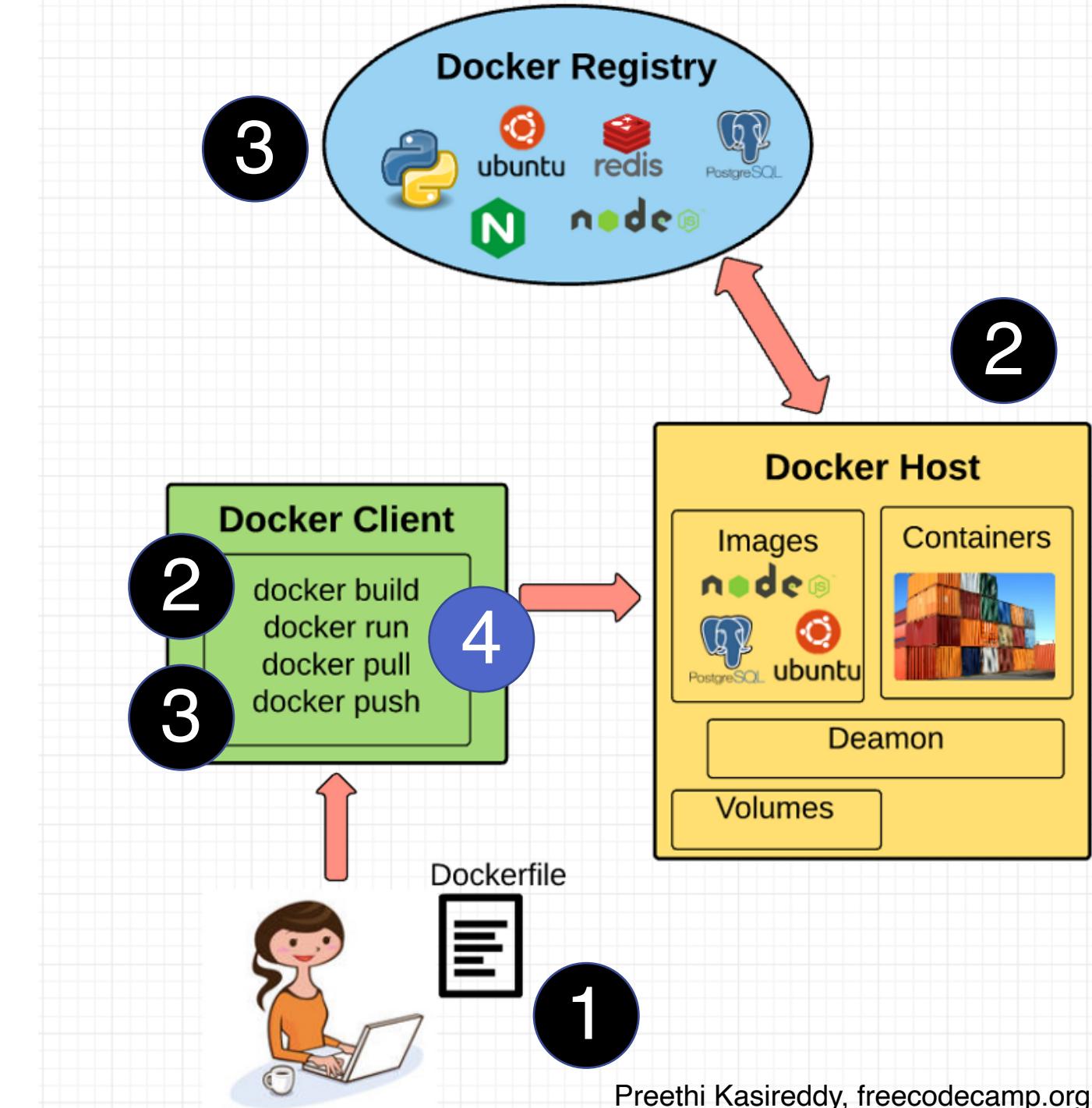
1. Write instructions for reproducing your computing environment
- 2. Compile the pieces**
3. Make it public
4. Others can reproduce your computing environment and pair this with your data and code



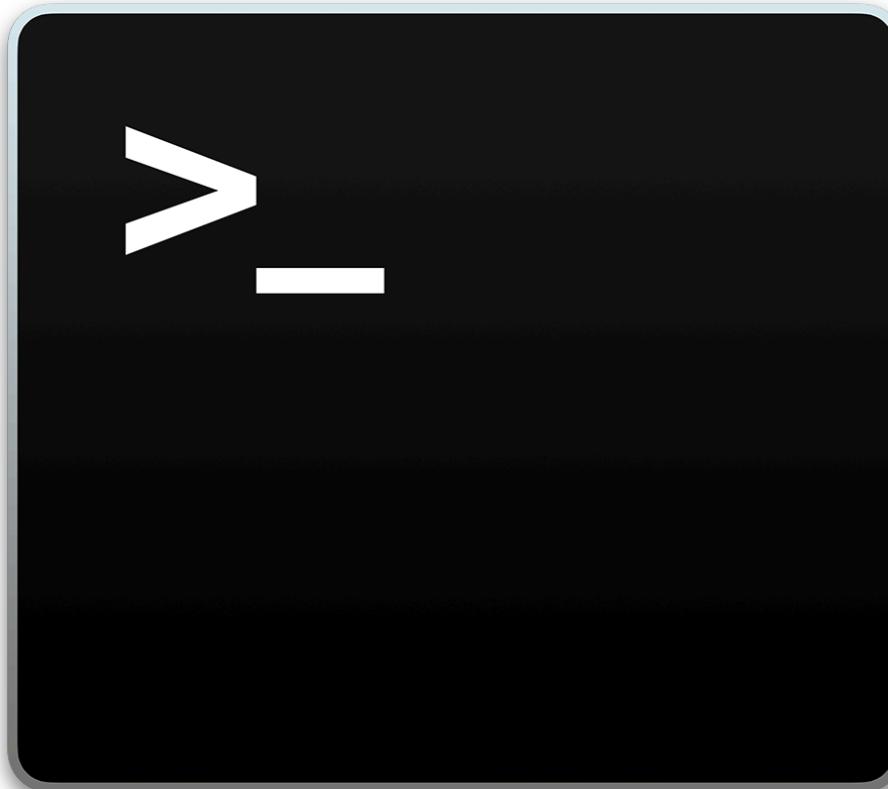
1. Write instructions for reproducing your computing environment
2. Compile the pieces
- 3. Make it public**
4. Others can reproduce your computing environment and pair this with your data and code



1. Write instructions for reproducing your computing environment
2. Compile the pieces
3. Make it public
4. **Others can reproduce your computing environment and pair this with your data and code**



We're going to use command line 😬😬😬



We're going to use command line 😬😬😬

Mac:

- Finder
- Applications
- Utilities
- Terminal

Windows:

- Windows key
- “cmd”
- Right-click and choose “run as administrator”

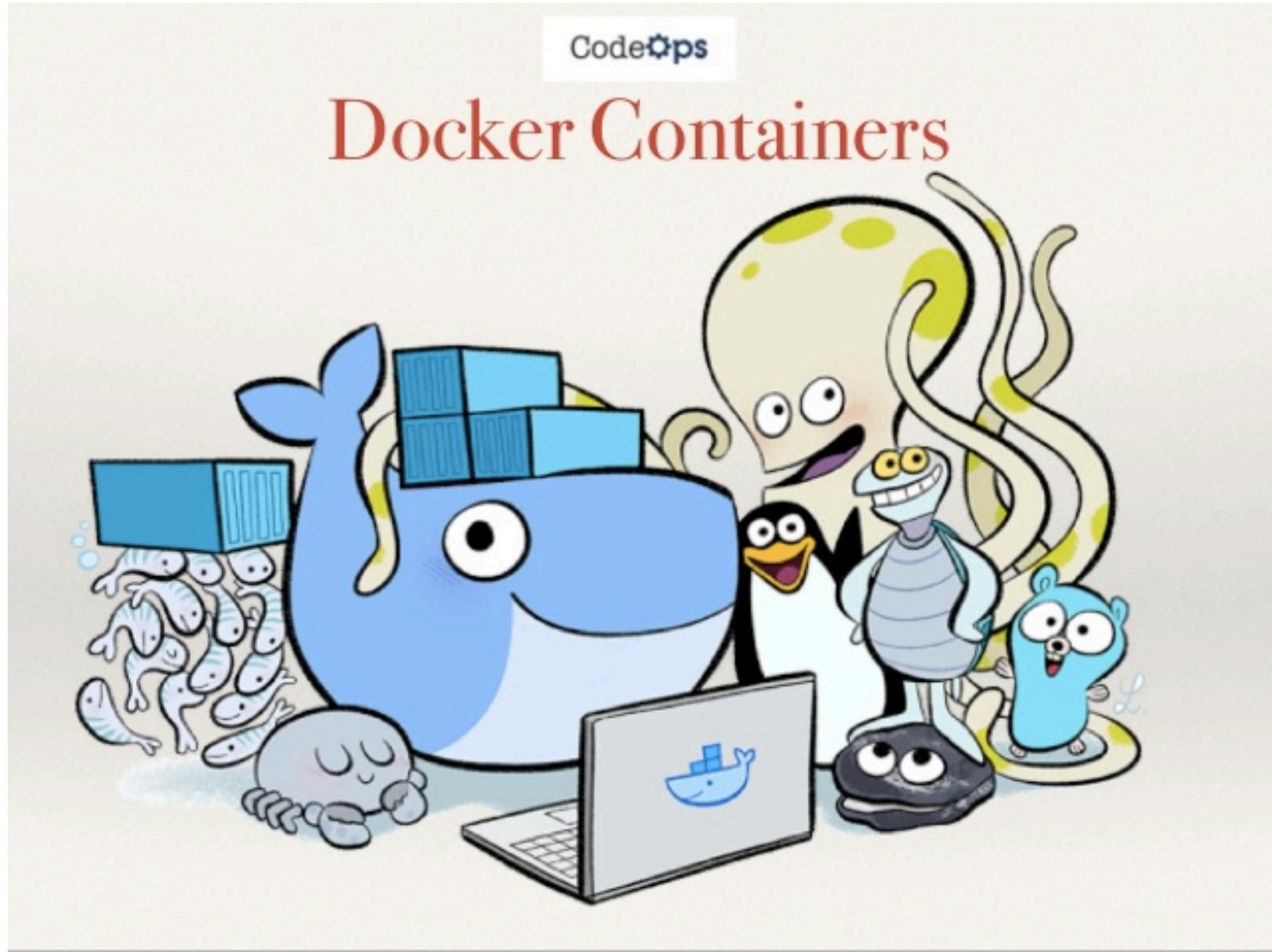


We're going to use command line 😬😬😬



Copy and paste commands:
tinyurl.com/docker-rstudio-tutorial

Part I: Practice with other Docker containers



Testing the Docker installation

```
> docker run hello-world
```

Running a Docker container

```
> docker run -d -e USER=<username>  
-e PASSWORD=<password> -p 8787:8787  
rocker/tidyverse
```



all
one
line

Replace with a username and password

Running a Docker container

```
> docker run -d -e USER=<username>  
-e PASSWORD=<password> -p 8787:8787  
rocker/tidyverse
```



all
one
line

All Docker commands start with “docker”
Here, we’re using the “run” command

Running a Docker container

```
> docker run -d -e USER=<username>  
-e PASSWORD=<password> -p 8787:8787  
rocker/tidyverse
```



all
one
line

“-[letter(s)]” is a way to specify options

-d = detached

The container will run in the background

Running a Docker container

```
> docker run -d -e USER=<username>  
-e PASSWORD=<password> -p 8787:8787  
rocker/tidyverse
```



all
one
line

-e = environment variable

Set the username and password

Running a Docker container

```
> docker run -d -e USER=<username>  
-e PASSWORD=<password> -p 8787:8787  
rocker/tidyverse
```



all
one
line

Maps a port from inside of the Docker container to your computer, which you'll access through a web browser

Running a Docker container

```
> docker run -d -e USER=<username>  
-e PASSWORD=<password> -p 8787:8787  
rocker/tidyverse
```



all
one
line

The image that we're using to run the container

The Rocker Project maintains R images for Docker

Others include: rocker/rstudio, rocker/geospatial, etc.

Running a Docker container

```
> docker run -d -e USER=<username>  
-e PASSWORD=<password> -p 8787:8787  
rocker/rstudio
```



Now that it's done, type “localhost:8787” into your browser

Stopping a Docker container

```
> docker ps
```

Get a list of running docker containers

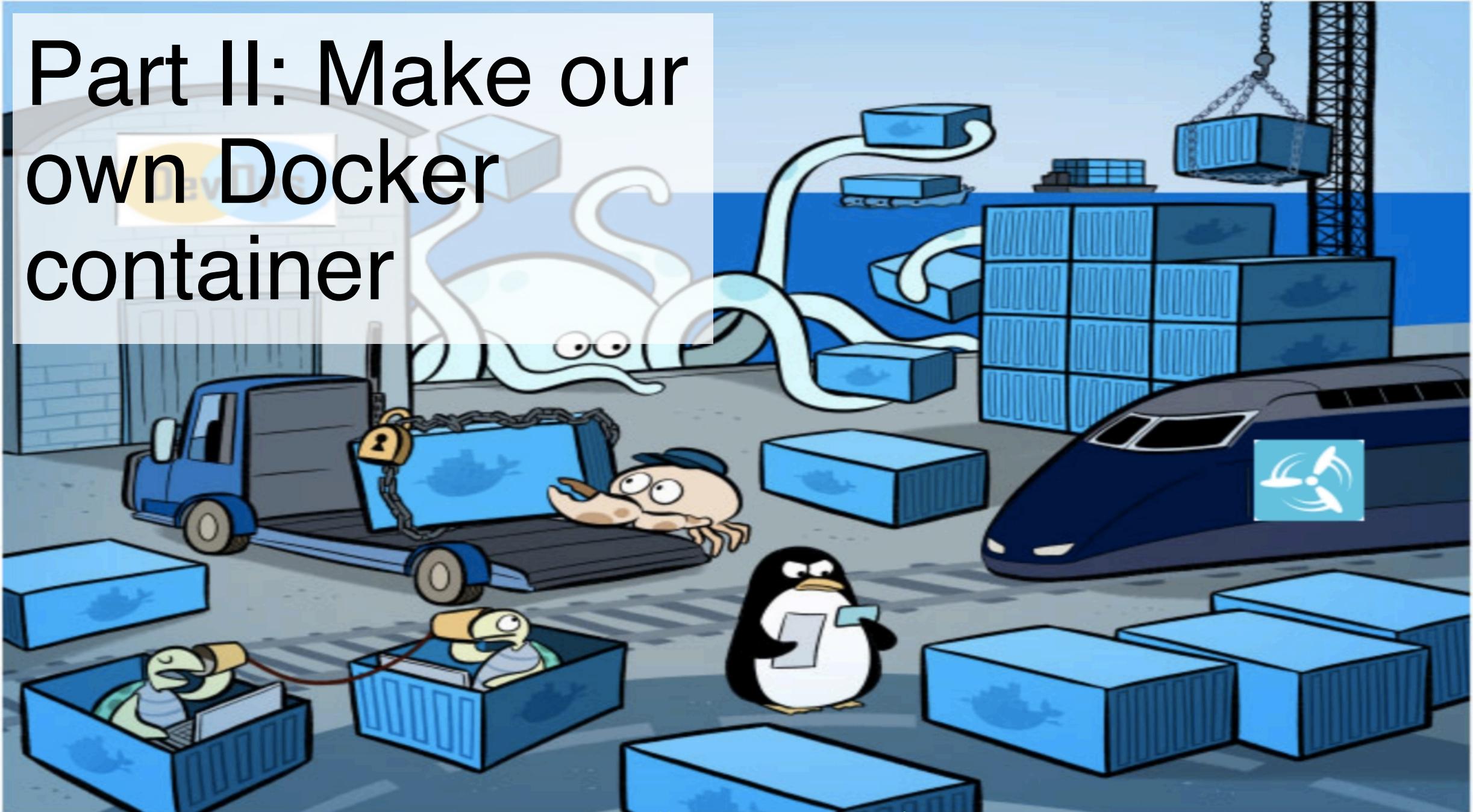
Copy the container ID (letters & numbers)

Stopping a Docker container

```
> docker stop <container ID>
```

Stop the container

Part II: Make our own Docker container



Some command line basics

cd = change directory

navigate to folder

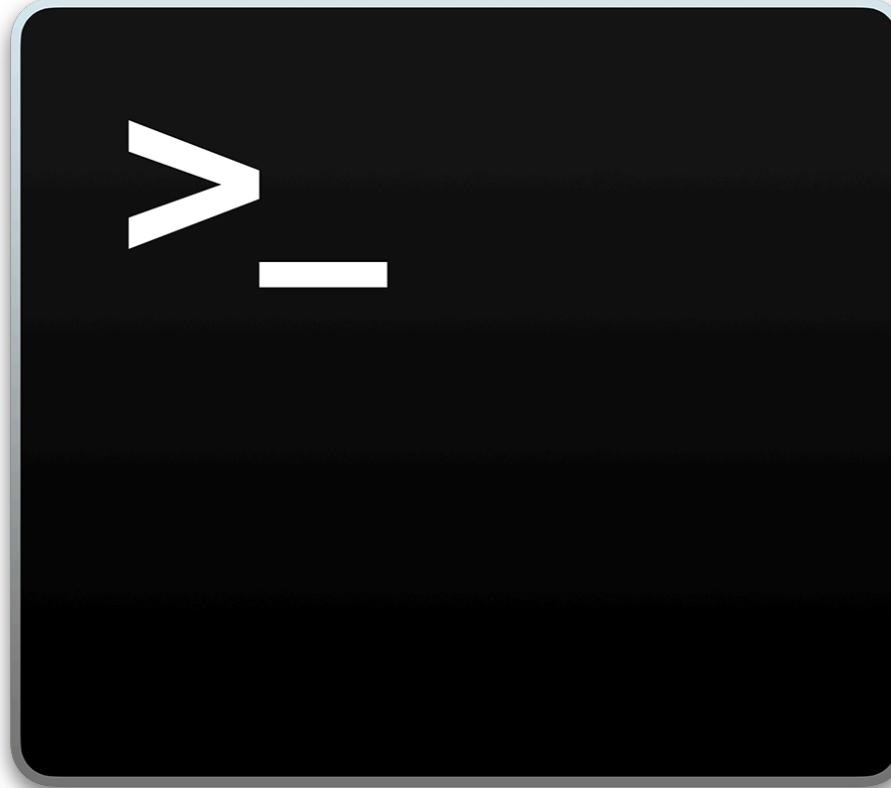
ls = list files

peek in folder

vi = open text editor

create a document

arrow keys = your best friends



Write the Dockerfile

```
> cd <file path to docker-rstudio-tutorial>  
> vi Dockerfile (Windows: copy con Dockerfile)
```

vi: i (to insert text, paste below)

```
FROM rocker/tidyverse:3.6.1
```

```
RUN R --no-restore --no-save
```

```
-e 'devtools::install_version("nlme", version="3.1-137")'  
-e 'devtools::install_version("ggplot2", version="3.1.1")'
```



Write the Dockerfile

vi

[esc] (to stop entering text)

:wq

[enter] (to exit the document)

copy con

[ctrl] + z

[enter] (to exit the document)

Build the container

```
> docker build -t <Docker username>/docker- } all  
rstudio-tutorial:1.0 . } one  
line
```

Don't forget the period!!!

“:1.0” adds the tag 1.0 in case you want to update it with newer versions (e.g., 2.0)

Run your container and add your code

```
> docker run -d -e USER=<username>  
-e PASSWORD=<password> -p 8787:8787  
-v <file path to docker-rstudio-  
tutorial>:/home/rstudio  
<Docker username>/docker-rstudio-tutorial:1.0
```



all
one
line

Replace with a username, password, and file path

Push image

```
> docker login  
> docker push <Docker username>/docker-  
rstudio-tutorial:1.0
```

} all
one
line

Navigate to Docker Hub (docker.com)
to see your image!