

Outline

- Reproducibility: frameworks and concepts
- Why do we need to make our analyses reproducible?
 - Trust in science - use stats from outreach pres
 - NSF requirement
- What are the barriers you hit when trying to reproduce someone's analysis? X
 - Understanding
 - Software
 - Bugs/errors
- What is a container?
- How do I build a container on my computer?
- How do I share it?
- How do I access other people's containers?
- What are alternatives to containers?

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.

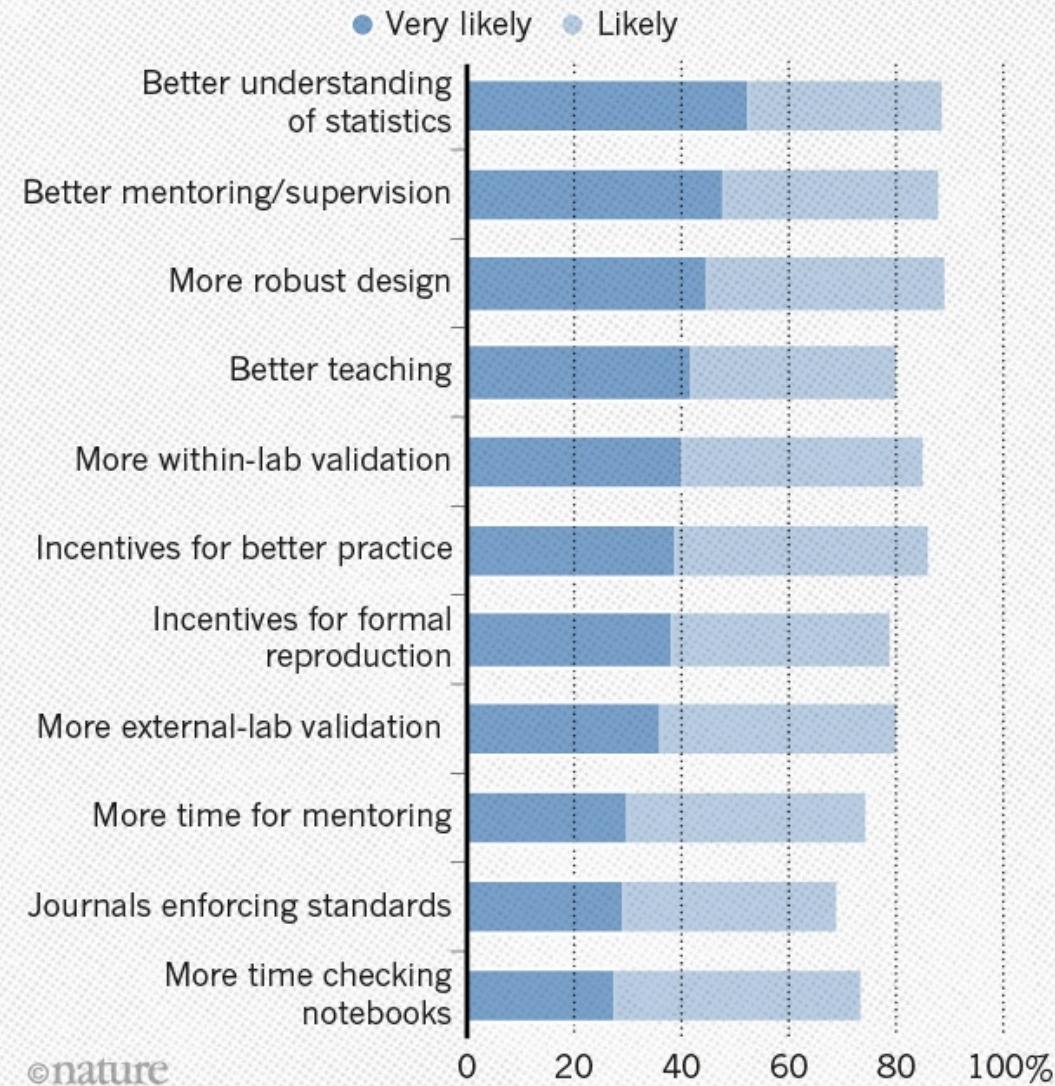
~ At the end of the lesson, write one thing that worked for you ~

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

The screenshot shows the RStudio IDE interface with several features highlighted:

- File menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Help.
- Toolbar:** Source on Save, Run, Source.
- Code Editor:** A script editor with the following code:

```
1 # Good start...
2 Cursors of shared users Re-run previous code Source with or without Echo Show file outline
3
4
5 Multiple cursors/column selection with Alt + mouse drag.
6 "P0030001"
7 "P0030002"
8 "P0030003"
9 "P0030004"
10
11
12+ get_digit <-function() {
13  ("num" %% (10 ^ n))
14  %% (10 ^ (n - 1))
15 }
16
17 fo
18  for {snippet}
19  foo {GlobalEnv}
20  force {base}
21 Jump to function in file
22
23 (Top Level) ▾
```
- Margin:** Shows code diagnostics like "P0030001" through "P0030004".
- Completion:** Shows tab completion for "get_digit".
- Snippets:** Shows a snippet for "fo".
- Console:** Shows command history and output:

```
> foo(1)
[1] 2
> foo <- function(x) x + 1
> foo(2)
foo(2)
> foo(1)
```
- Help:** Shows "Working Directory" and "Press ↑ to see command history".
- Panes:** Shows "Maximize, minimize panes" and "Drag pane boundaries".

R Support

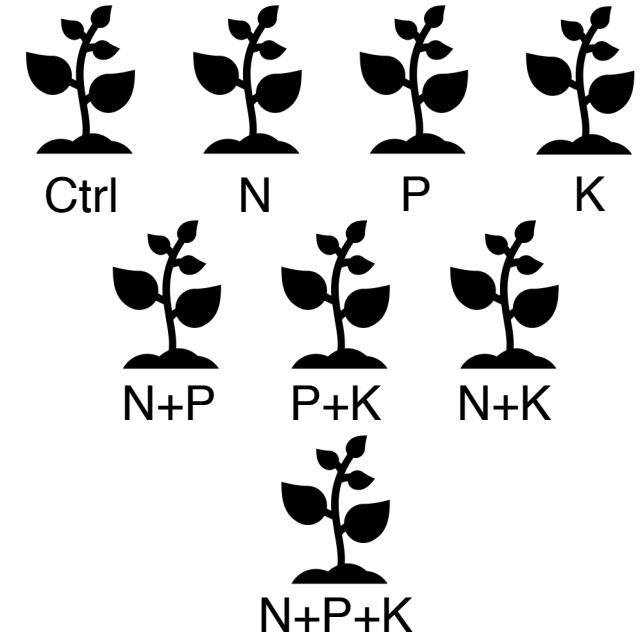
Import data file with wizard History of past commands to run/add to source Display .RPres slideshows
File > New File > R Presentation

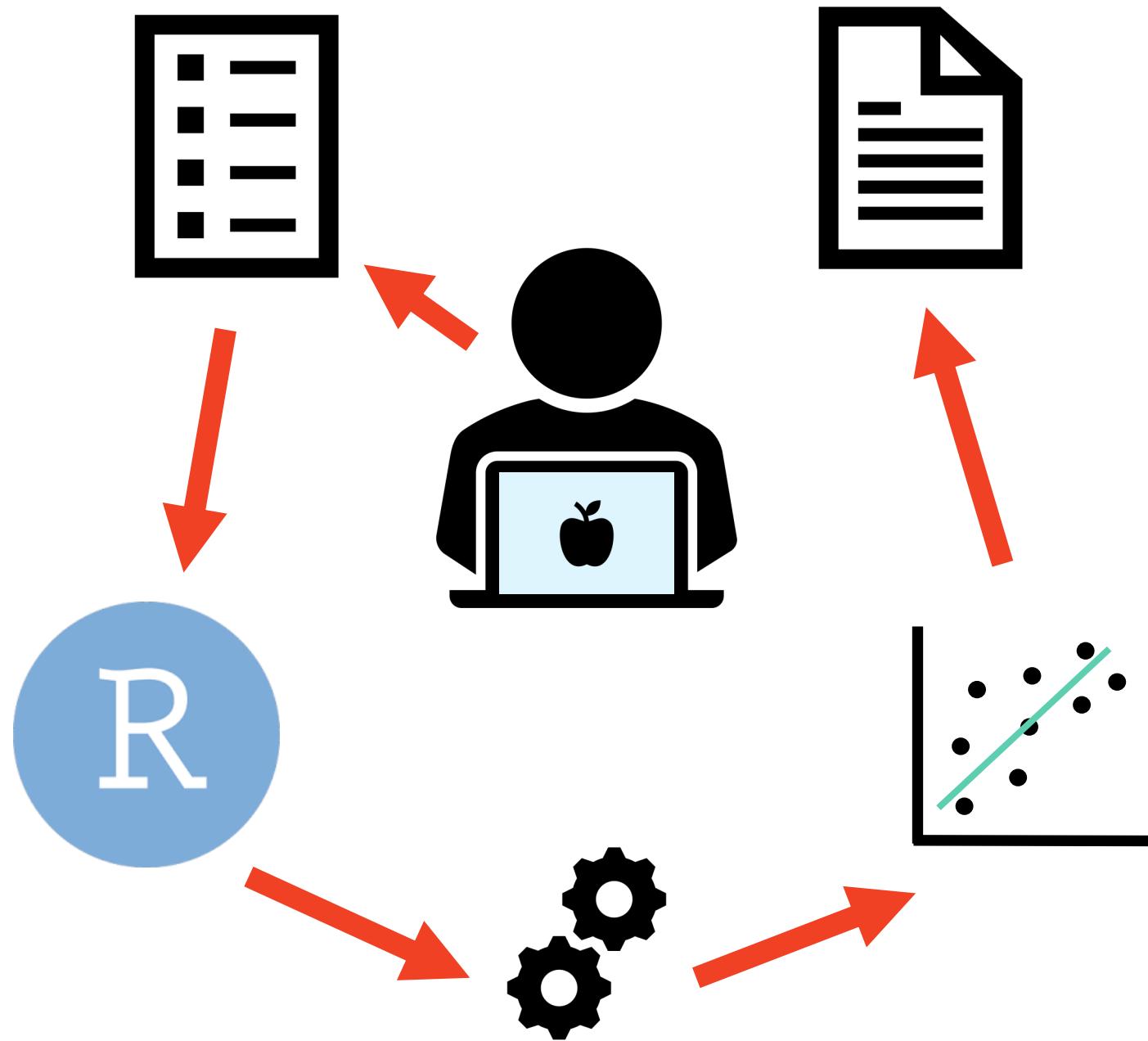
The screenshot shows the RStudio IDE interface with several features highlighted:

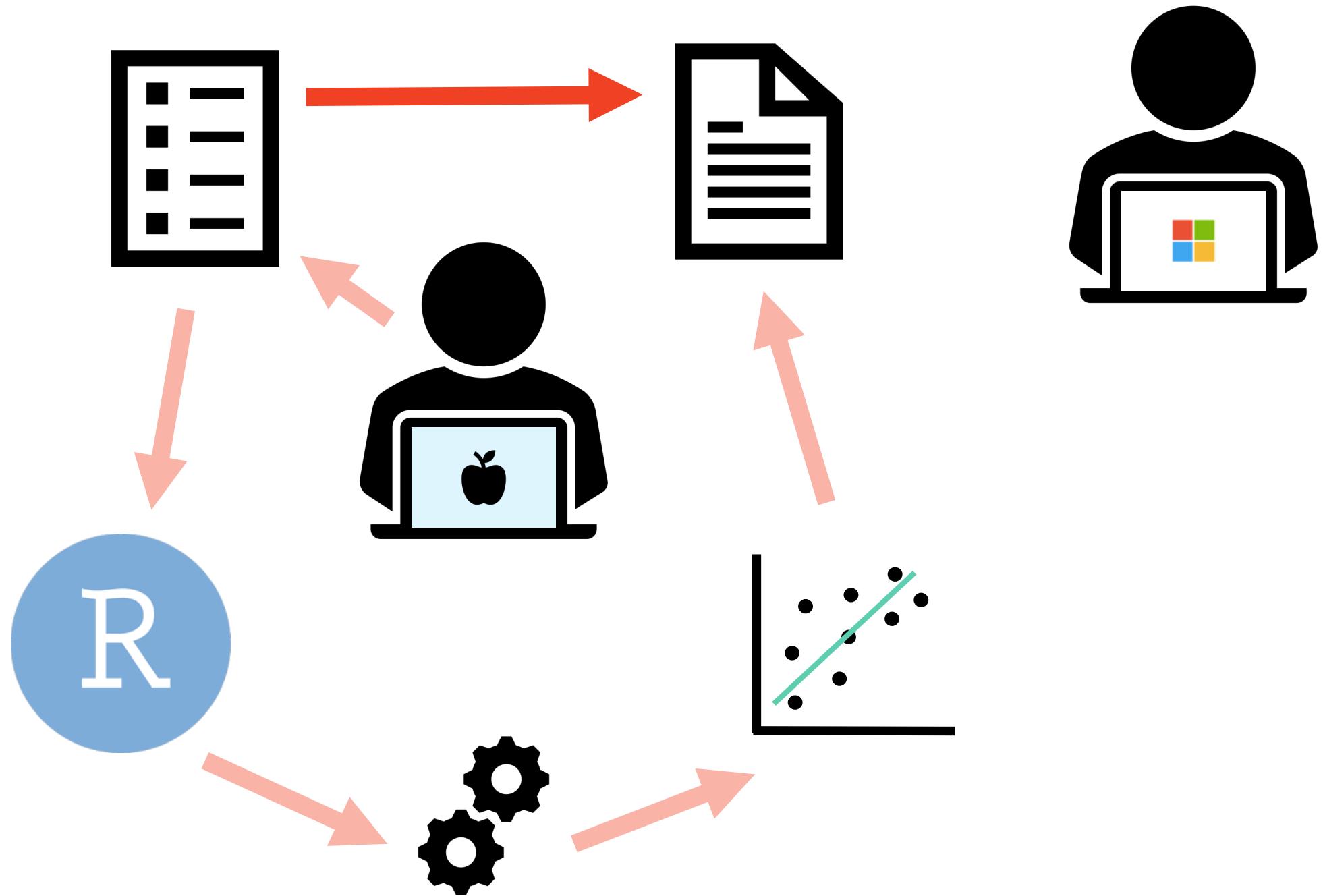
- Environment Tab:** Environment, History, Build, Git, Presentation.
- Global Environment:** Shows "Load workspace", "Save workspace", "Delete all saved objects", and "Search inside environment".
- Data View:** Shows "Data", "Values", and "Functions".
 - Data: iris, 150 obs. of 5 variables.
 - Values: a, 1.
 - Functions: foo, function (x).
- Viewer:** Shows "Displays saved objects by type with short description", "View in data viewer", and "View function source code".
- File Browser:** Shows "Files", "Plots", "Packages", "Help", "Viewer".
 - Files: New Folder, Upload, Delete, Rename, More.
 - Home: IDEcheatsheet.
 - Operations: Create folder, Upload file, Delete file, Rename file.
 - Context menu: Copy..., Move..., Export..., Set As Working Directory, Go To Working Directory.
- Path:** Shows "Path to displayed directory" and "Change directory".
- File Browser Description:** A file browser keyed to your working directory. Click on file or directory name to open.

Let's analyze our data!

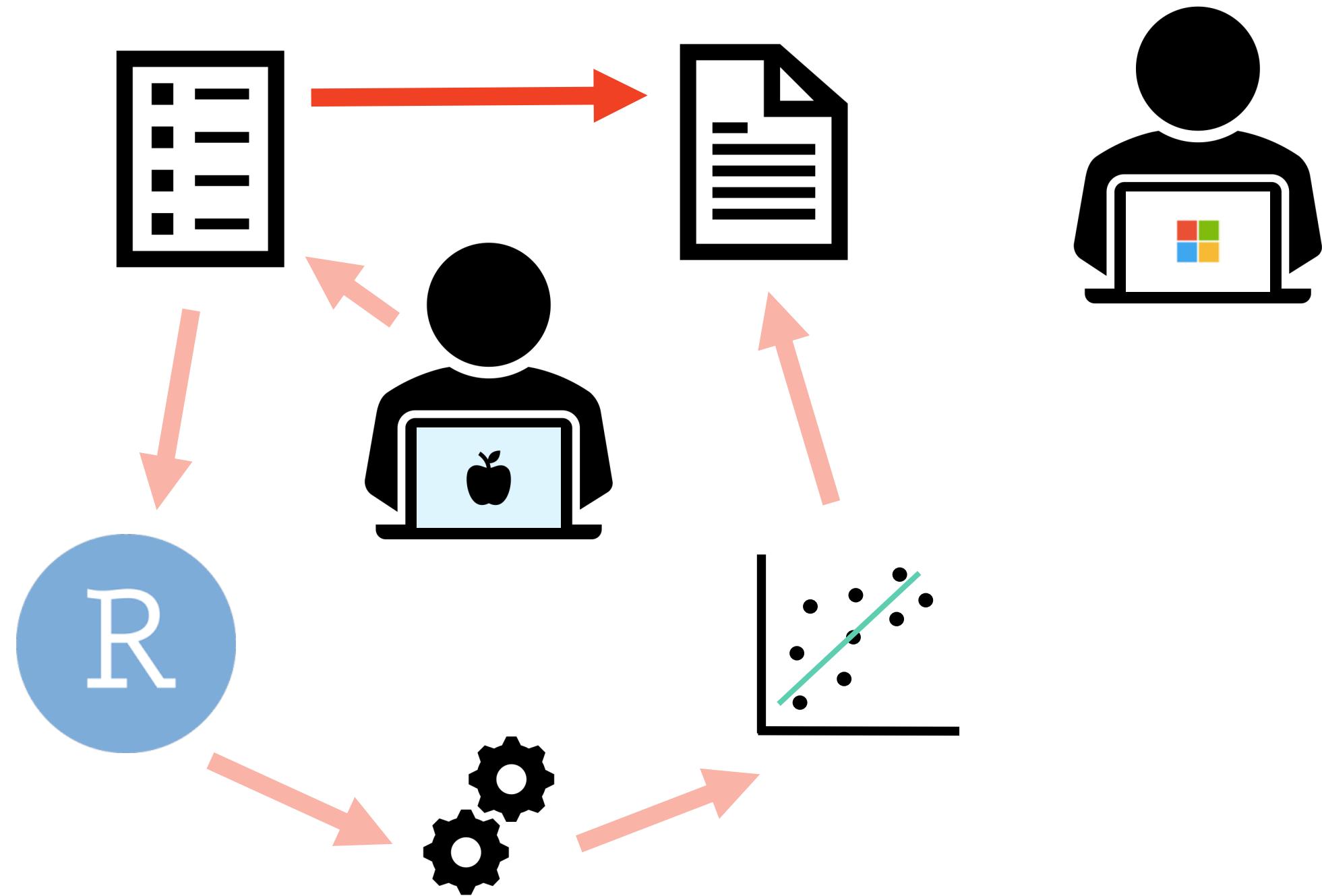
1. Create a new folder “docker-rstudio-tutorial”
2. Create a new R script in RStudio
3. Navigate to tinyurl.com/npk-script
4. Copy the entire page
5. Paste into the new R script
6. Save the script in “docker-rstudio-tutorial” as myScript.R
7. Run the script and examine the output

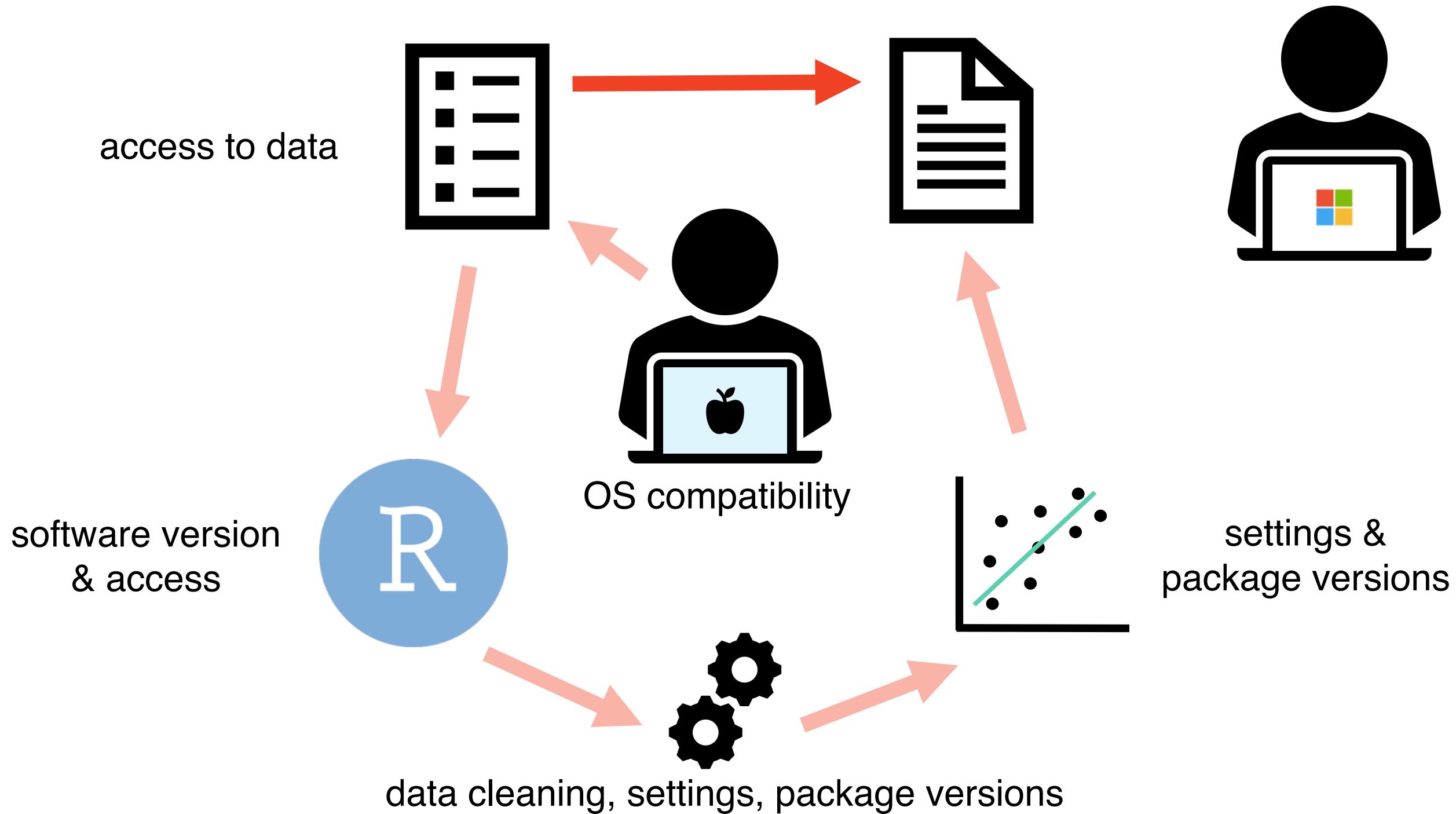




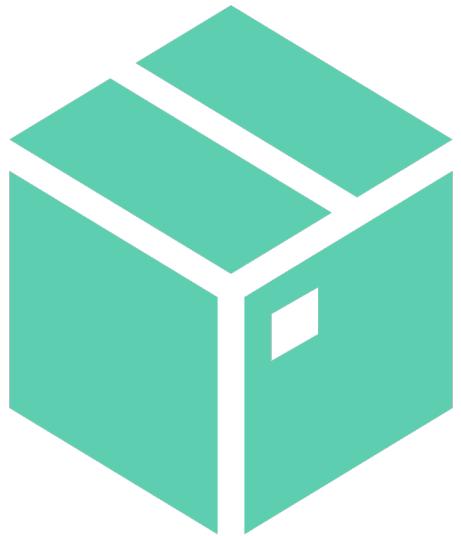


What are
some
barriers to
reproducing
results?



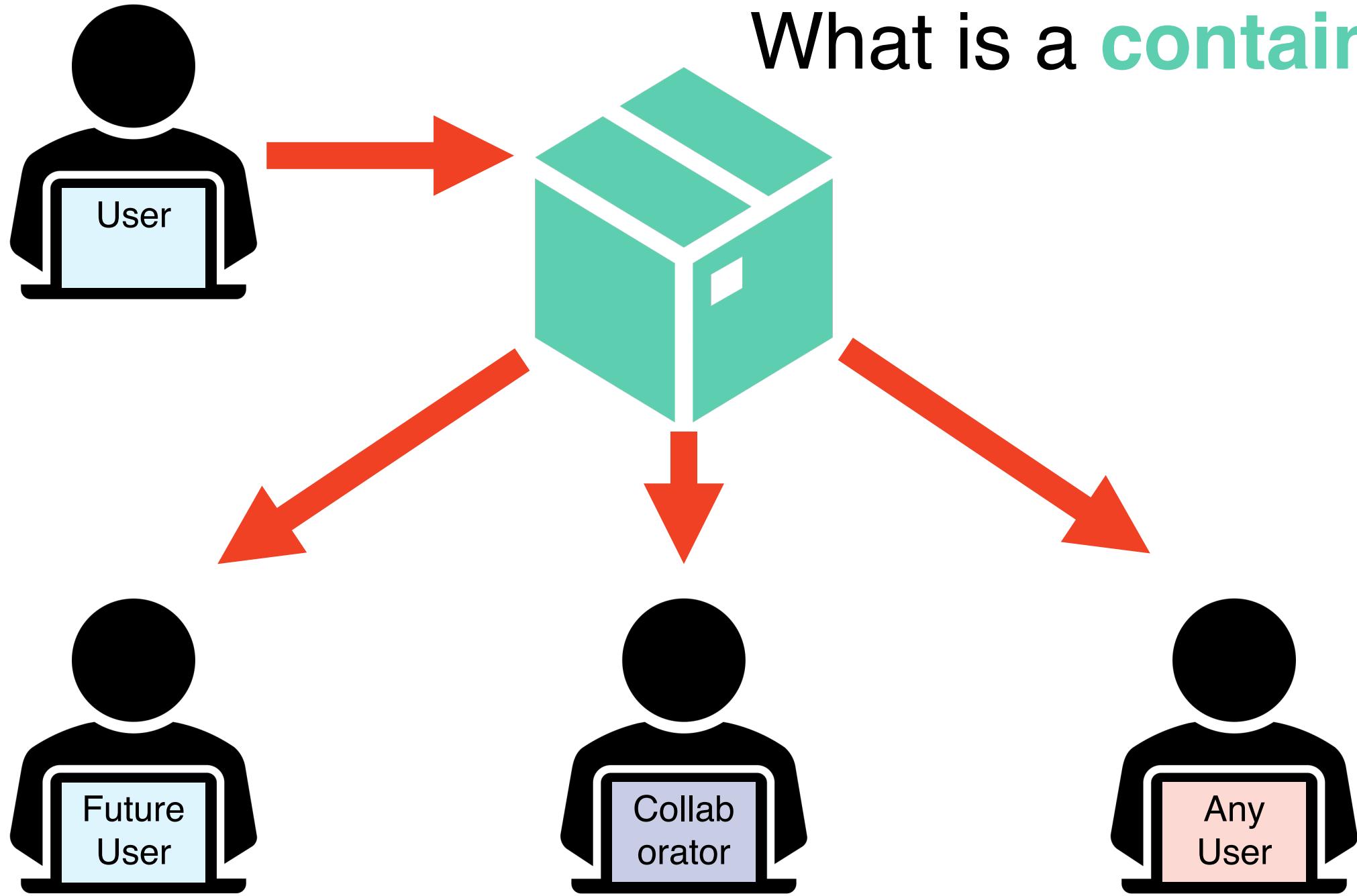


One piece
of the
solution:



a **container**

What is a container?

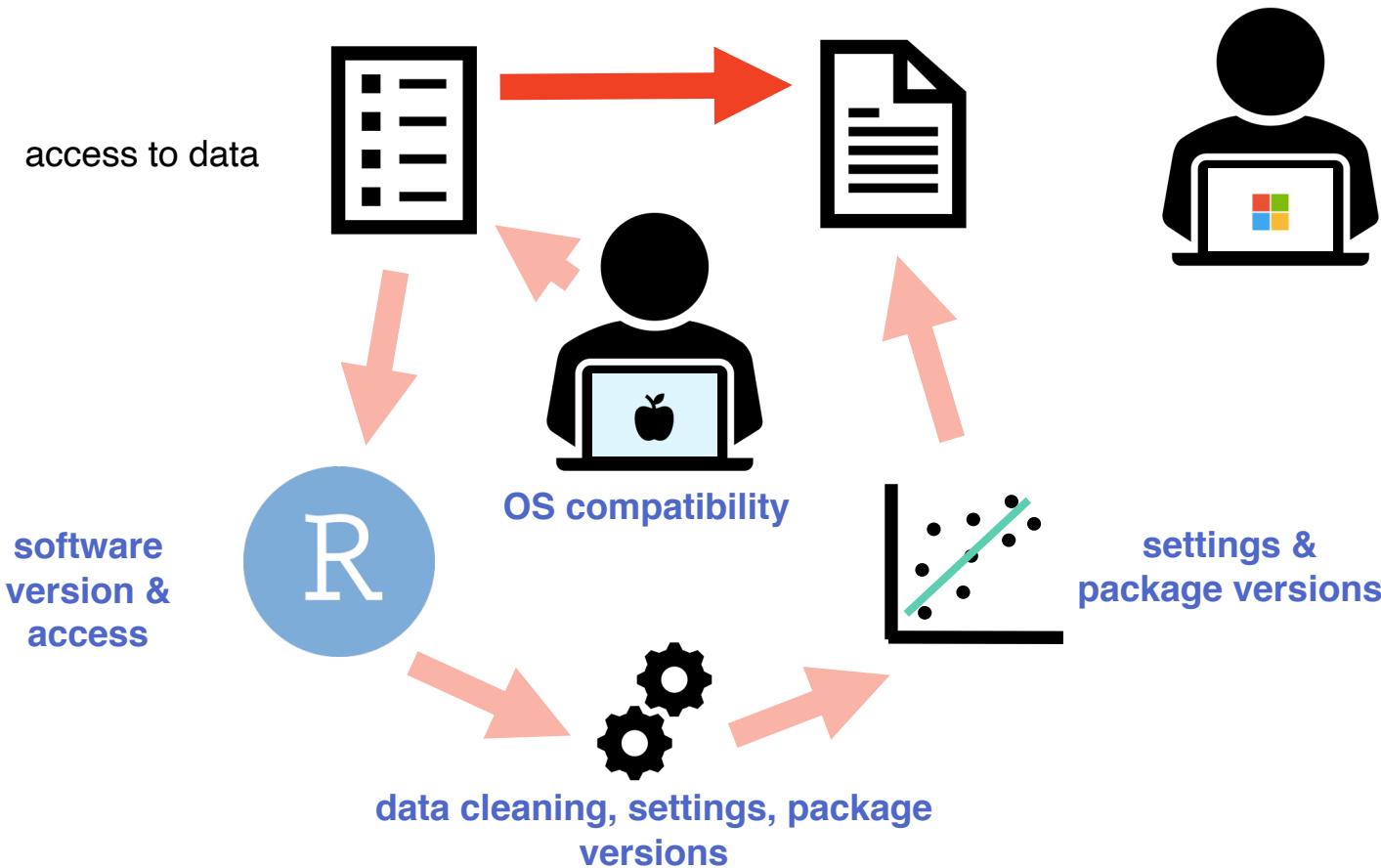


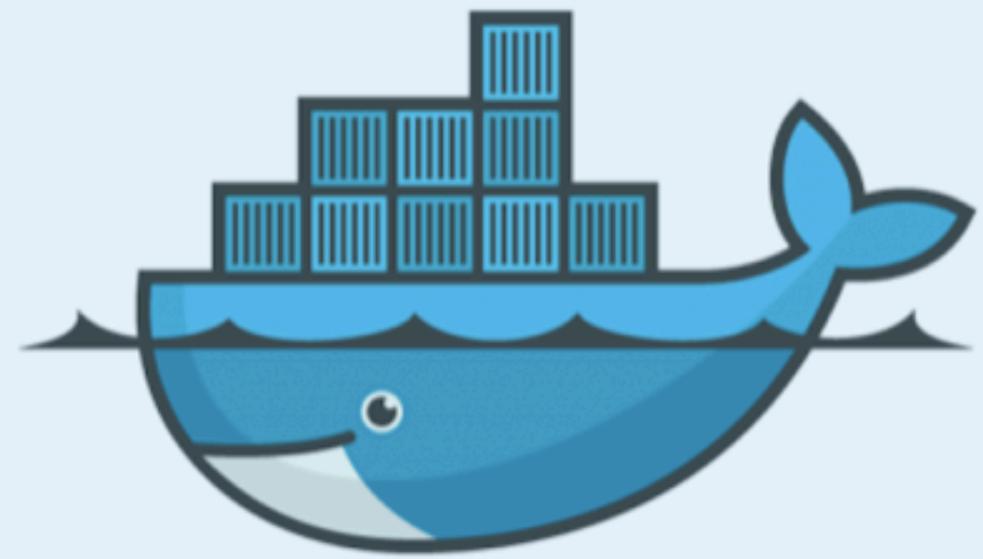
Using containers for reproducible research

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

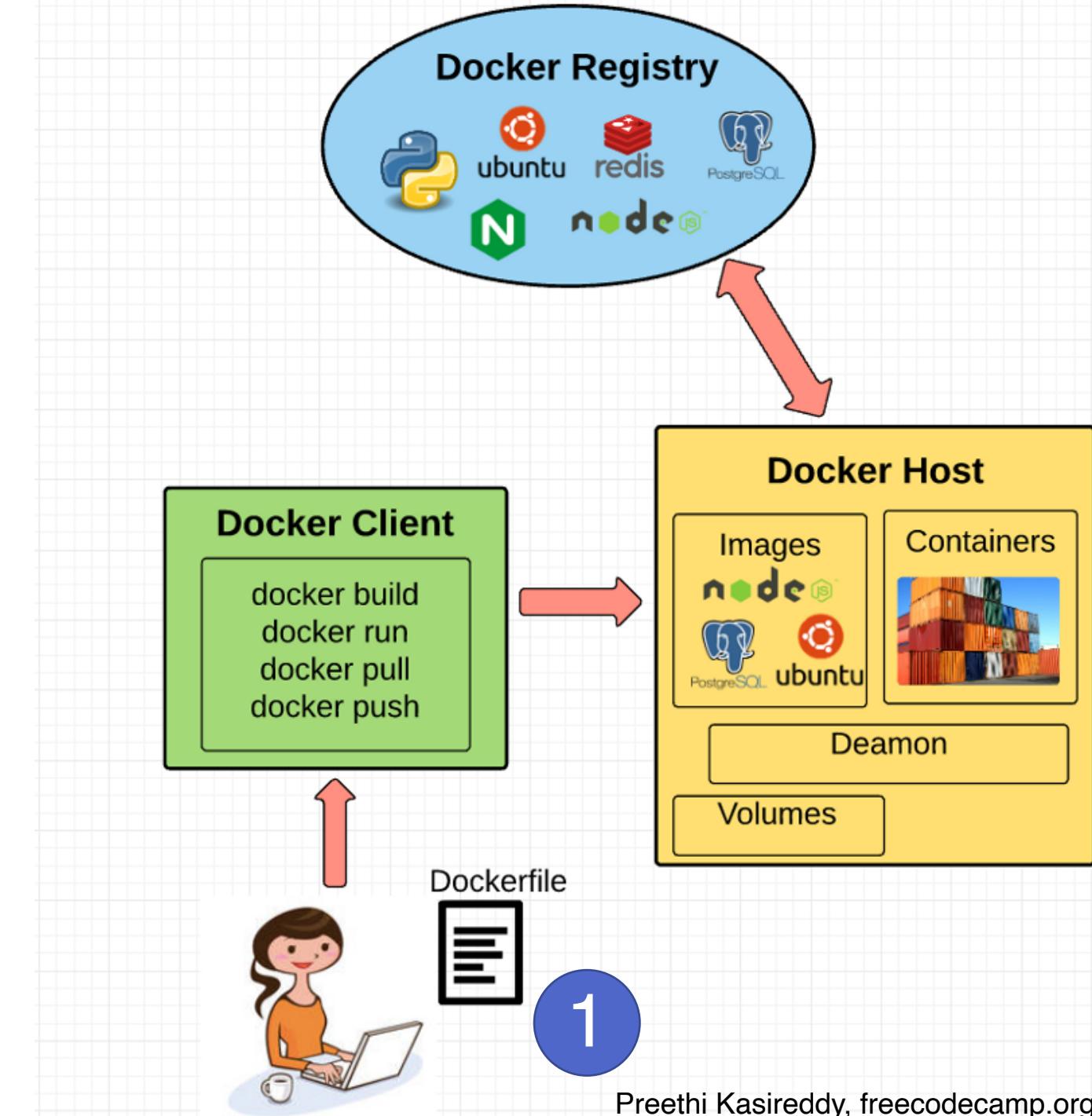
1. Write instructions for reproducing your computing environment
2. Compile the pieces
3. Make it public
4. Others can reproduce your computing environment and pair this with your data and code



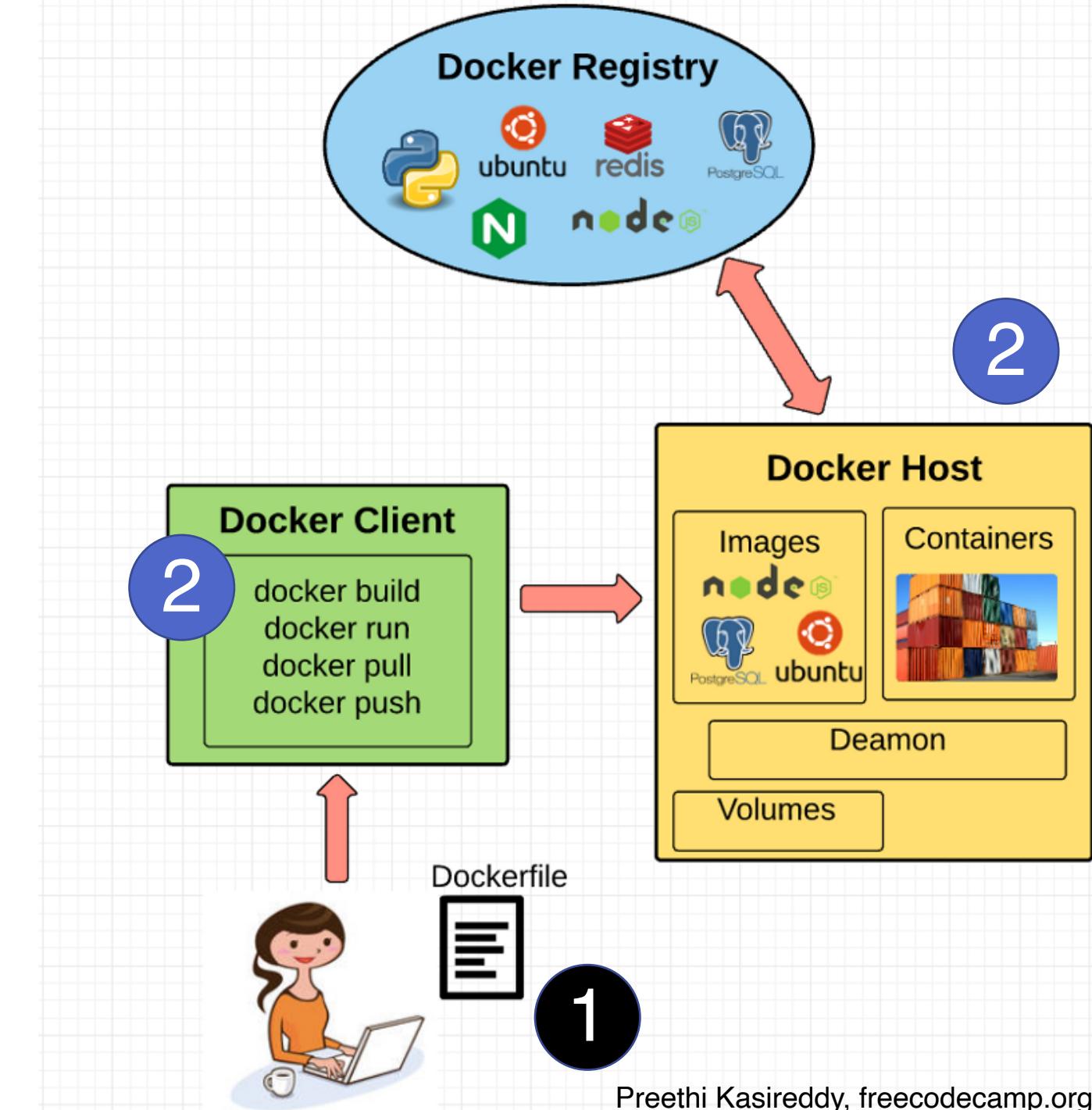


docker

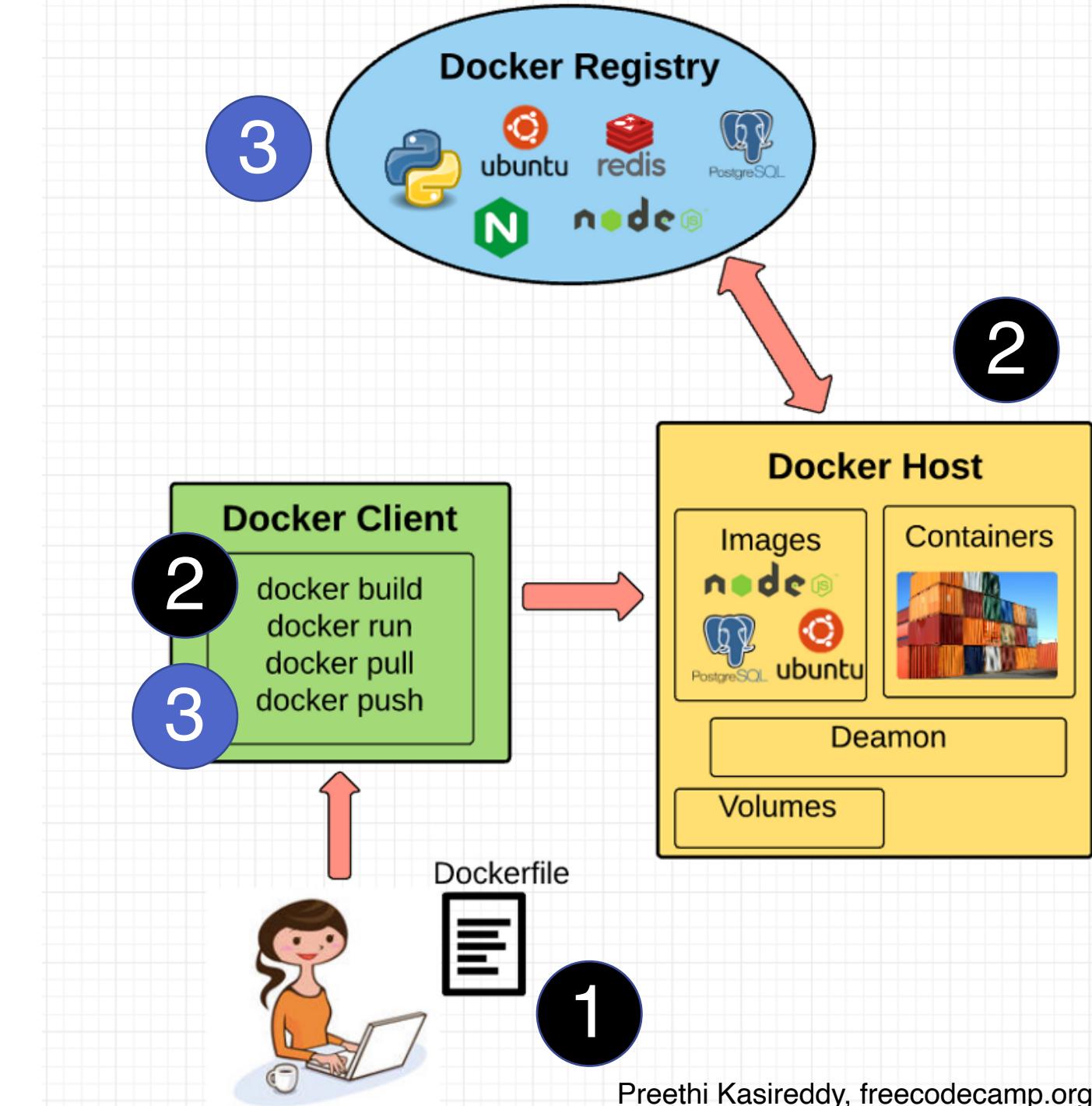
- 1. Write instructions for reproducing your computing environment**
2. Compile the pieces
3. Make it public
4. Others can reproduce your computing environment and pair this with your data and code



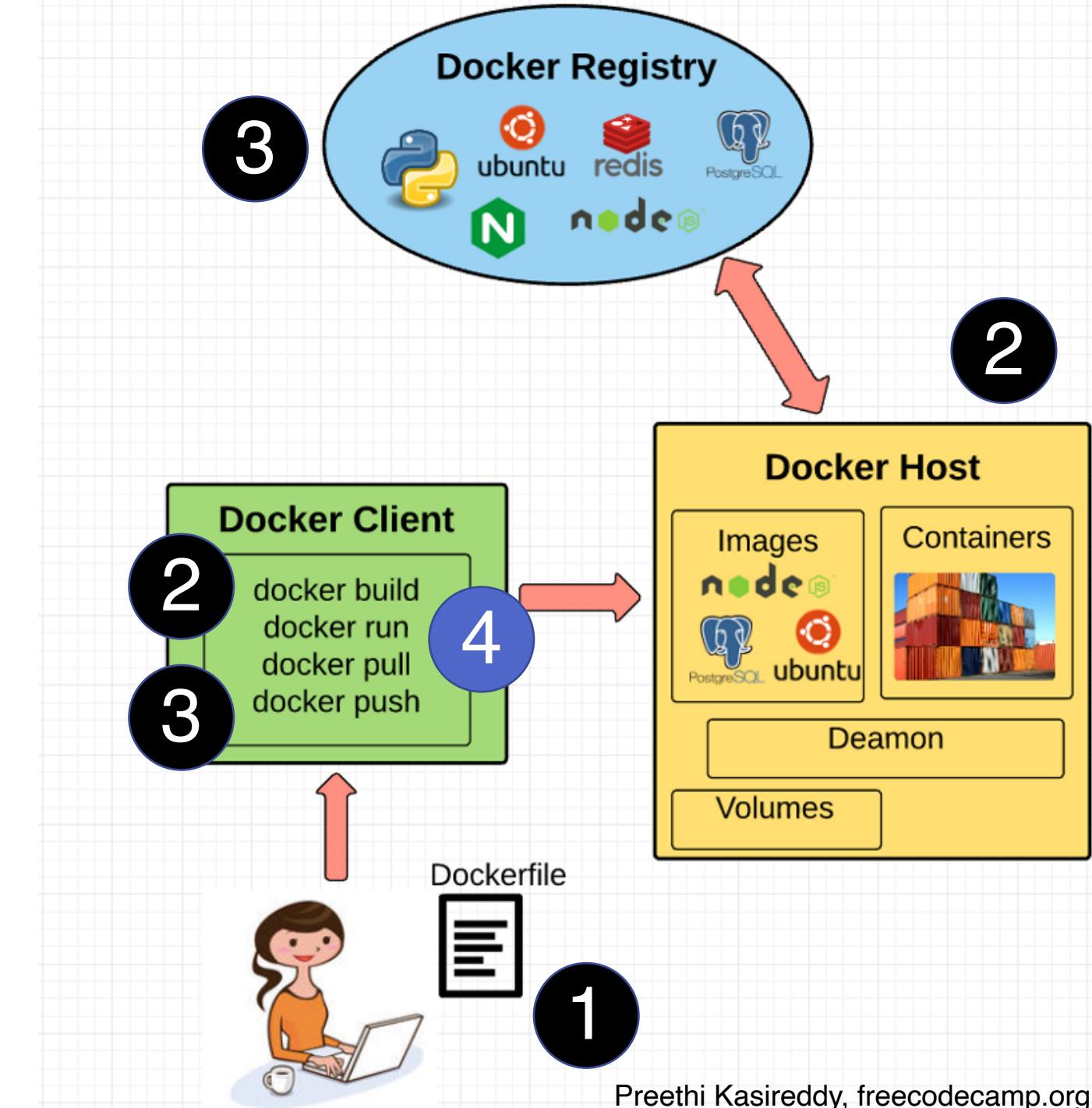
1. Write instructions for reproducing your computing environment
- 2. Compile the pieces**
3. Make it public
4. Others can reproduce your computing environment and pair this with your data and code



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:

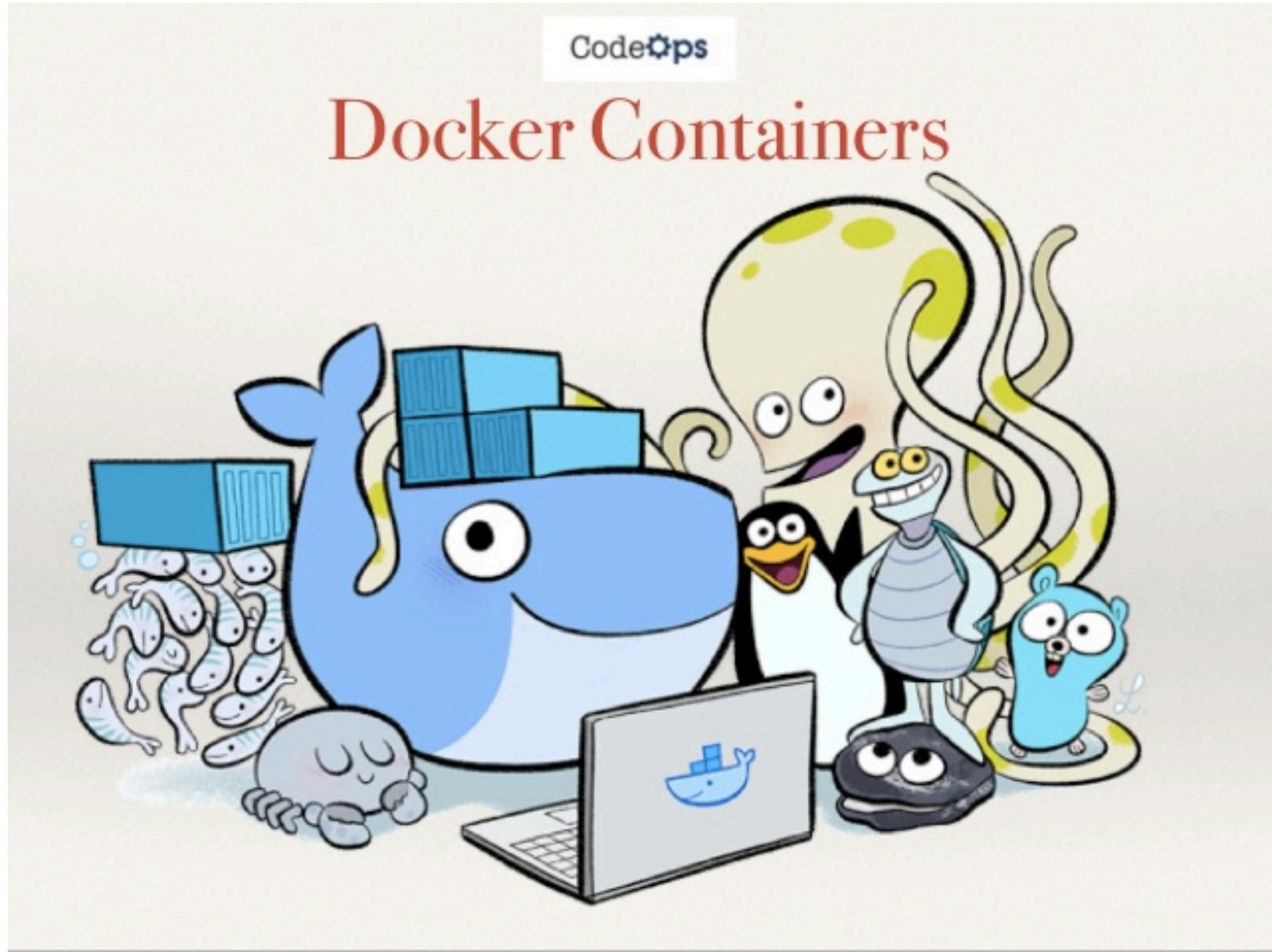
- Finder
- Applications
- Utilities
- Terminal



Windows:

- Search
- Command prompt
- Right-click and choose “run as administrator”

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
```

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 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
```

“-[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
```

-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
```

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
```

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

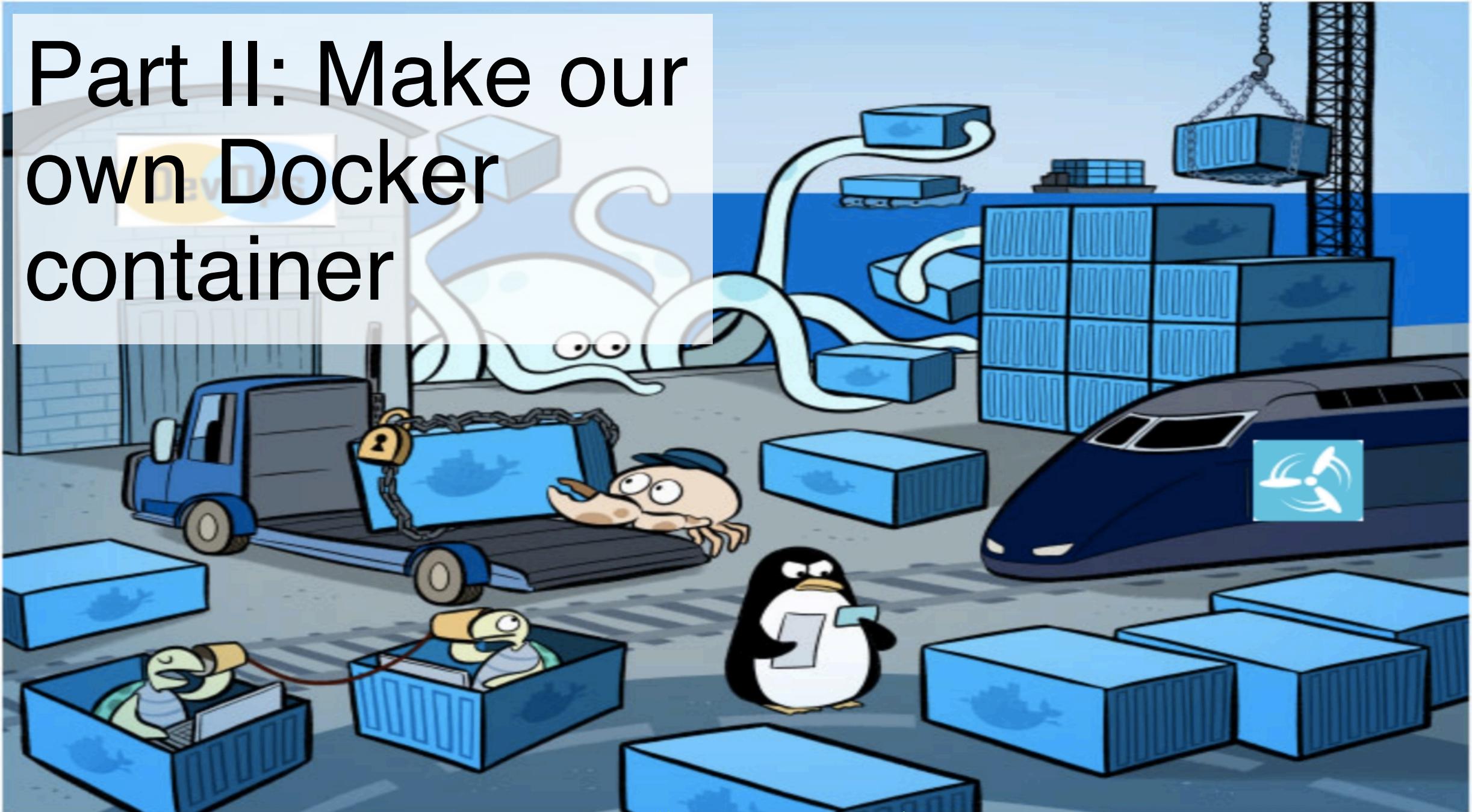
Copy the container ID (letters & numbers)

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

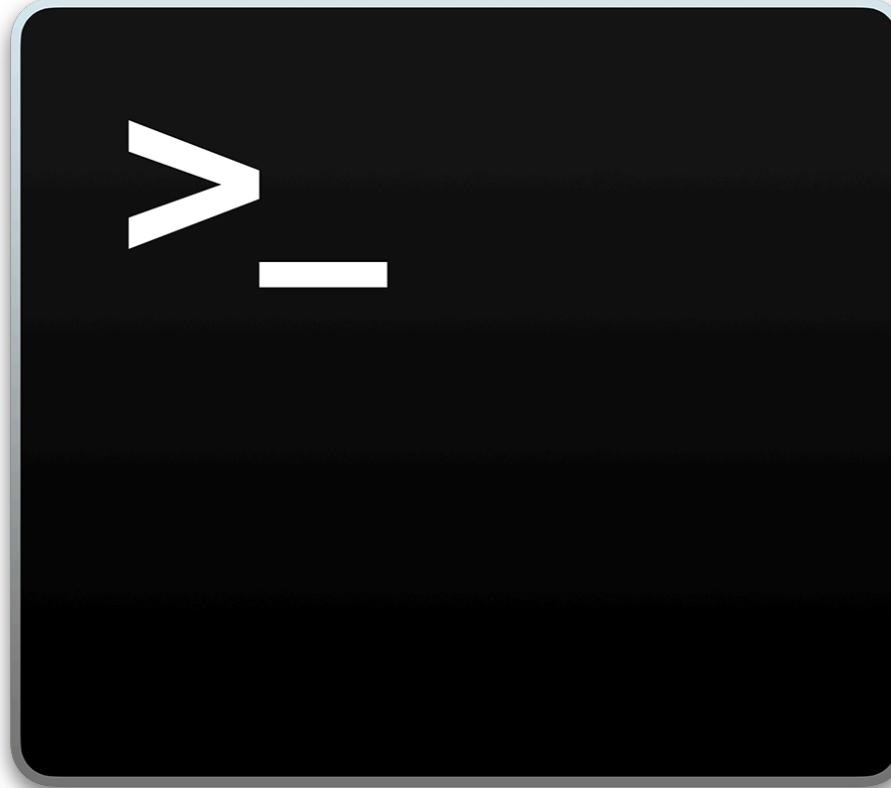
ls = list files

peek in folder

vi = open text editor

create a document

arrow keys = your best friends



Write the Dockerfile

```
> cd <file path/docker-rstudio-tutorial>  
> vi Dockerfile  
          (Windows: use \)
```

i (to insert text)

```
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")'
```

} all one
line

Write the Dockerfile

```
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")'
```

} all one line

[esc] (to stop entering text)

:wq

[enter] (to exit the document)

Build 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 ~/docker-rstudio-tutorial:/home/rstudio
<Docker username>/docker-rstudio-tutorial:1.0
```



Replace with a username, password, and file path
~ is the home directory for Mac
Windows: use Users\username

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!