

Docker

These slides were copied from Charlotte Mann who was the GSI last year, and were originally Adapted from slides by Dr. Zoe Rehnberg from Winter 2020

Where to start...

- Johann's reproducibility template
 - <https://github.com/johanngb/rep-template>
- Hunt supplement
- Youtube ([one of many options](#))

Basic Ingredients

- Dockerfile
- makefile
- .dockerignore

Dockerfile

- Instructions to build docker image
- Lines run sequentially and cached
- A line is only run if it (or what it refers to) has changed

makefile

- Defines a set of commands
- Command is executed when
`make [commandName]`
is called (with docker image running)

.dockerignore

- Includes files/directories that can be ignored to improve processing time

Dockerfile

```
FROM johanngb/rep-int:latest
```

FROM: Loads base image

```
WORKDIR /home/rep/  
COPY . /home/rep
```

WORKDIR: Sets image working directory

COPY: Copies files from local machine to docker image

```
RUN su - rep -c 'cd /home/rep/docker/build && make'
```

```
CMD ["R"]
```

CMD: Provides default for execution

RUN: runs a command and commits result (e.g. loading packages)

Dockerfile cont.

- Full Dockerfile [reference](#)
- `ENV [key]=[value]` ([link](#))
 - Sets environment variables
 - Example:

```
ENV PATH="home"
WORKDIR ${PATH}
```
- [Notes](#) on `COPY`:
 - `COPY test.txt relativeDir/`
 - **copies to** `<WORKDIR>/relativeDir/`
 - `COPY test.txt /absoluteDir/`
 - **copies to** `/absoluteDir/`

makefile

```
1 .PHONY: build rerun download analysis clean
2
3 build: download analysis
4
5 rerun: clean analysis
6
7 download:
8     cd data && R -e 'rmarkdown::render("download.Rmd")'
9
10 analysis:
11     cd analysis && R -e 'rmarkdown::render("01_preprocess.Rmd")'
12     cd analysis && R -e 'rmarkdown::render("02_main_analysis.Rmd")'
13
14 clean:
15     echo "Deleting all processed data and output..."
16     rm -f data/processed/*
17     rm -f analysis/*.html
18     rm -f output/*
19     rm -f archive.tar.bz2
20
21 archive:
22     rm -f archive.tar.bz2
23     tar --exclude='.git' -cjf /tmp/archive.tar.bz2 . && mv /tmp/archive.tar.bz2 .
```

. PHONY identifies commands that should be run regardless of the state of the file system

name of command

lines to execute in command

.dockerignore

Include things like:

`.git`

`.gitignore`

`README.md`

`extraDirectory/*`

[documentation](#)

Base image

FROM [baseImage]

- Docker has a list of “official images”
 - <https://github.com/docker-library/official-images/tree/master/library>
- Some base images:
 - R: <https://github.com/rocker-org/rocker>
 - R: <https://hub.docker.com/r/johanngb/rep-int>
 - Python: <https://github.com/docker-library/python>

Building an image

Build your image:

```
$docker build -t [imageName]
```

.



Establishes the local working directory

Check existing images:

```
$docker images
```

Clear old image versions:

```
$docker system prune
```

Dockerhub

- [Repositories](#) like github
- Push your image to a repository
 - First, name the image in reference to repository:

```
$docker build -t <hub-user>/<repo-name>[:<tag>] or
```

```
$docker tag <imageName> <hub-user>/<repo-name>[:<tag>]
```

- Then push:

```
$docker push <hub-user>/<repo-name>:<tag>
```

Connecting to Project 4 requirements

- The command

```
docker run -it --rm yourdockerhubname/yourimagename
```

should provide a bash terminal where commands can be run to reproduce the analysis

- The following commands should be available:

- `make clean` -- deletes everything except for the code (i.e., markdown files) and raw data (as originally downloaded)
- `make` -- runs all analyses (except downloading raw data and making current predictions)
- `make predictions` -- makes current predictions and outputs them to the screen
- `make rawdata` -- deletes and re-downloads the raw data

- Other make commands, if/as appropriate

- The output should look like:

```
"YYYY-MM-DD", XX.X, XX.X, XX.X, ..., XX.X
```

where

- `"YYYY-MM-DD"` is the current date
- There are 300 numbers of the form `XX.X` where each X is a digit
- The numbers are predictions in degrees fahrenheit
- The order is given by:
Anchorage +1 day, minimum; Anchorage +1 day, average; Anchorage +1 day, maximum;
Anchorage +2 days, minimum; ...; Washington DC +5 days, maximum

Connecting to Project 4 requirements

- What will need to be in the Dockerfile?
 - Command to run bash
 - Base image
 - Where to work in the image
 - Copy files-scripts! from local directly
 - Install packages
 - Run make - ensure intermediate files are there
- What will need to be in the makefile?
 - Commands to run analyses
 - Way to download historical data (R/Python)

Connecting to Project 4 requirements

- What is the best way to organize your scripts?
 - Draw out an ideal workflow
 - Pseudo-code makefile
- What is the best way to format the data?