Docker for Reproducible Research

Hareem Naveed, MunichRe

About Me

Data Science For Social Good

THE UNIVERSITY OF CHICAGO

Center for Data Science and Public Policy

THE UNIVERSITY OF CHICAGO

Munich RE

Tools for Data Scientists

- GitHub
- Jupyter Notebooks
- RMarkdown
- Docker

Tools for ~~Data Scientists~~ Economists

- GitHub
- Jupyter Notebooks
- RMarkdown
- Docker

Agenda

- Reproducible Research
- What is Docker
- Anatomy of a Docker file
- Containers and Registries
- How to set up your development environment on any machine using Docker
- How to deploy a simple model using flask, python, and Docker

If this has happened to you...

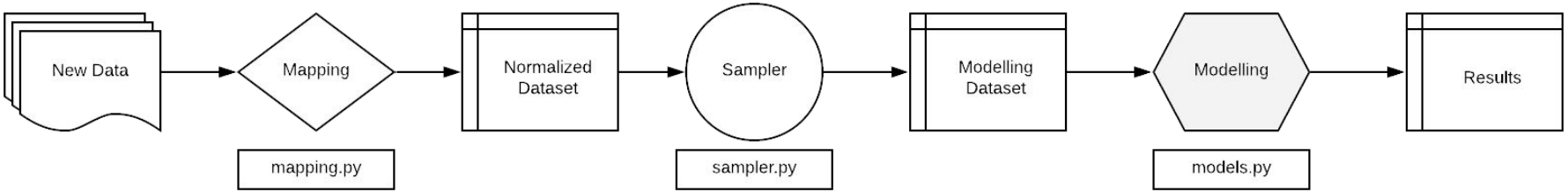
“I installed everything like you told me to, it still doesn’t work!”

“I can’t be bothered to install this on Mac/Windows/Linux.”

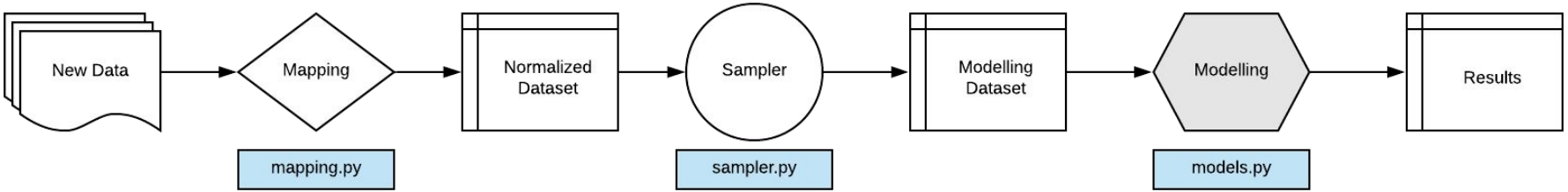
“Our model deployment is simple: just send me the records as an Excel file and I’ll score each one for you.”

...you need Docker

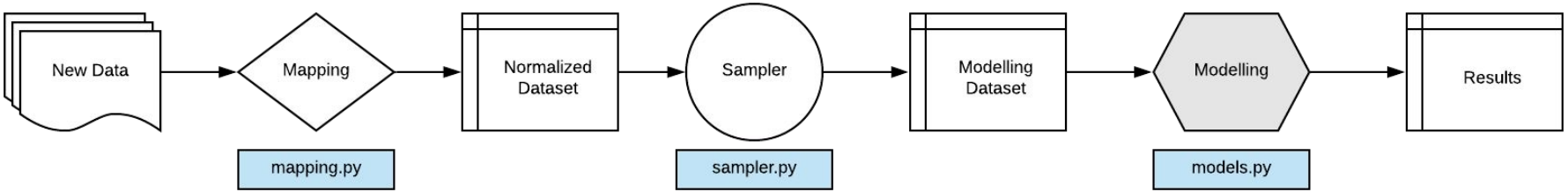
Research is a set of Transformations



Research is a set of Transformations



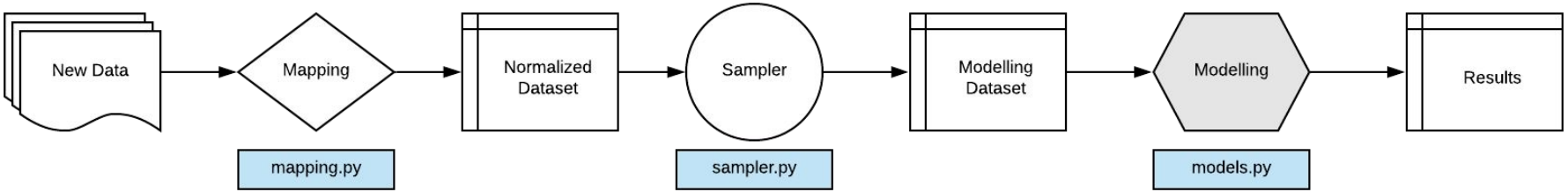
Research is a set of Transformations



What do I need to rerun this code?

- Python
- System dependencies
- Package

Research is a set of Transformations



What do I need to rerun this code?

- Python (which version?)
- System dependencies (which version?)
- Packages (which version?)

Reproducible Research

1. Re-running original analyses (same code, same data)

Reproducible Research

1. Re-running original analyses (same code, same data)

2. Applying the analysis (same code, different data)

Reproducible Research

1. Re-running original analyses (same code, same data)

   Re-create a development environment

2. Applying the analysis (same code, different data)

   Deploy your model as an API

What is Docker

- A lightweight virtualization tool
- Take your code and your environment, and freeze it
- Anybody else running that docker container will reproduce your exact environment
- Typical economist, spends a lot of time modelling, visualizing
- May not have the organizational capacity to have a data engineer help prototype dashboards and model deployment

Examples of how I have used Docker

- Built dashboards for a transit agency to dynamically calculate and visualize an equity metric
- Deployed machine learning models for clients to access via API
- Handed off machine learning models for clients to run locally
- Built pipelines where each task was represented as a Docker container
- Deployed documentation as a web-service (similar to readthedocs!)

Terms

IMAGES

- Read-only template that docker containers are built from
- Images usually built from other images, with some customizations

CONTAINERS

- Small virtualization that helps you install, build and run your code/workflow in isolation
- “ephemeral”

Terms (Good to Know)

DOCKER DAEMON

- Use the docker cli to make calls to the daemon
- Daemon manages the orchestration for docker

Anatomy of a Dockerfile

```
FROM python:3

RUN apt-get update && apt-get install -y gnumeric

WORKDIR /code

COPY xlsx_to_csv.sh /code

RUN chmod +x /code/xlsx_to_csv.sh

ENTRYPOINT [ "/bin/bash", "/code/xlsx_to_csv.sh" ]
```

Anatomy of a Dockerfile

FROM

The FROM statement specifies the base image. In our example, we are taking the `postgres` base image from Dockerhub.

LABEL

The LABEL statement adds metadata to the image. It is optional, but is helpful if you are pushing your containers to a shared registry so people know who to contact in case of an issue.

RUN

The RUN statement is the workhorse of the Dockerfile. In our case, we are using it to run shell commands. These commands have nothing to do with Docker but are basic Linux commands.

Anatomy of a Dockerfile

WORKDIR

The WORKDIR statement is often used to specify a working directory. Any subsequent commands will assume that is the working directory.

ADD

The ADD statement lets you copy files from the host machine to the docker container.

CMD

The CMD statement is used to provide defaults when executing a container. Only one CMD statement is valid per container, and if you provide several, only the last one will be used by the container.

Anatomy of a Dockerfile

```
FROM python:3

RUN apt-get update && apt-get install -y gnumeric

WORKDIR /code

COPY xlsx_to_csv.sh /code

RUN chmod +x /code/xlsx_to_csv.sh

ENTRYPOINT [ "/bin/bash", "/code/xlsx_to_csv.sh" ]
```

Containers & Registries

- Build a container
- Pull from dockerhub
- Push to dockerhub

Recreate Dev Environment

```
FROM python:3

RUN apt-get update && apt-get install -y python3-pip

COPY requirements.txt .

RUN pip install -r requirements.txt

# Install jupyter
RUN pip3 install jupyter

# Create a new system user
RUN useradd -ms /bin/bash demo

# Change to this new user
USER demo

# Set the container working directory to the user home folder
WORKDIR /home/demo

# Start the jupyter notebook
ENTRYPOINT ["jupyter", "notebook", "--ip=0.0.0.0"]
```

Recreate Dev Environment

```
docker build -t devenv .
```

Recreate Dev Environment

```
Hareems-MacBook-Air:dev-env hareemnaveed$ docker build -t devenv .
Sending build context to Docker daemon 6.656 kB
Step 1/9 : FROM python:3
3: Pulling from library/python
16ea0e8c8879: Pull complete
50024b0106d5: Pull complete
ff95660c6937: Pull complete
9c7d0e5c0bc2: Downloading 33.12 MB/51.79 MB
29c4fb388fdf: Downloading 18.81 MB/192 MB
8659dae93050: Download complete
d4b57a0e98ff: Downloading  14.2 MB/29.52 MB
9d187245c5fc: Waiting
cd75e768c3f7: Pulling fs layer
```

Recreate Dev Environment

```
docker run -p 8888:8888 devenv
```

Deploy a Model

- Write an app.py script
- Pickled object
- Deploy!

Deploy a Model

```
docker build -t model .

docker run -p 1000:80 model
```

...Another way I have used Docker

- Presentation to RLadies Toronto; my computer died 5 minutes before my talk

Data Science for Social Good

- Improving Workplace Safety in Chile through Proactive Inspections
- Tackling Tenant Harassment in New York City: A Data-Driven Approach

www.dssgfellowship.org

https://datascienceforsocialgood.pt/