Unit 3: Core tools

Write CLI tools

Lesson 26

Derek Ruths

Overview of unit

Objectives:

- Understanding how data science activities necessitate certain kinds of tools
- Fundamentals with core data science tools

1. Why core tools?
2. Project organization
3. Python
4. Best practice: write CLI tools
5. Best practice: write unit tests
6. Best practice: resource referencing
7. Github
8. Jupyter notebooks
9. Jupyter & statefulness
10. Bokeh
11. Advanced bokeh
12. HW 3

Lesson overview

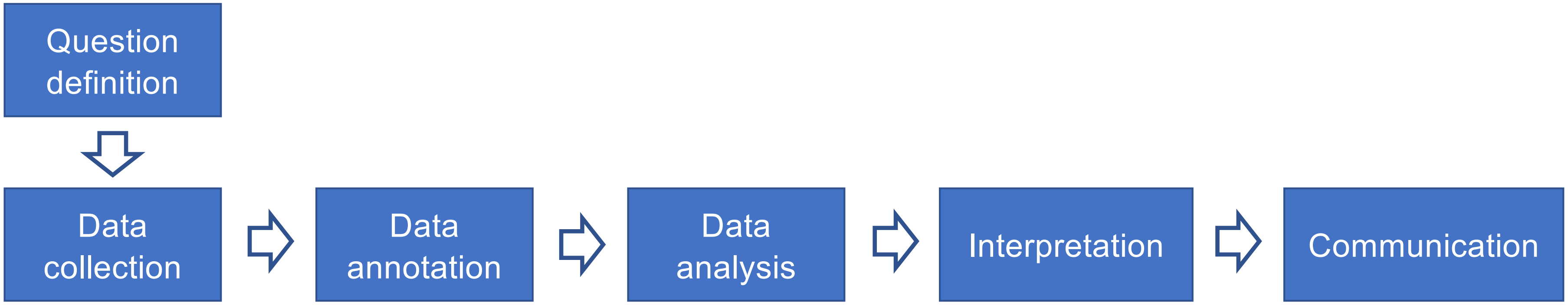
Objectives

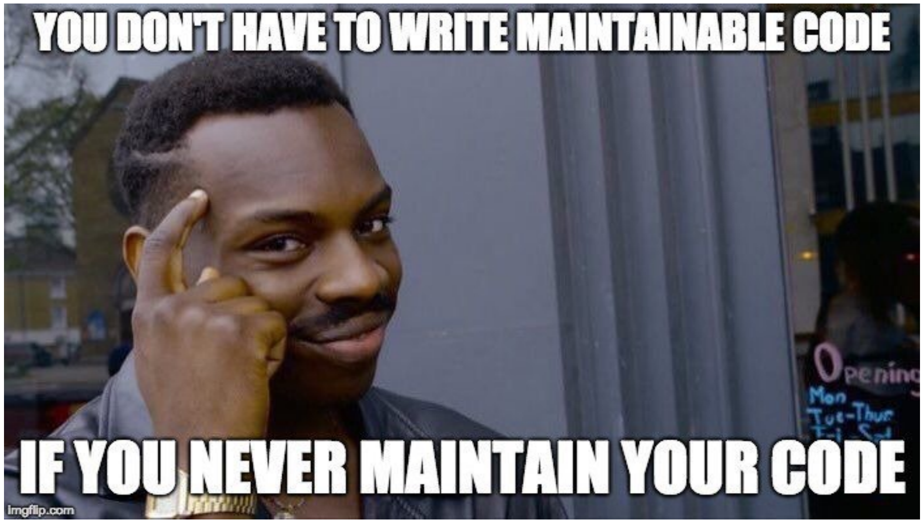
- Be convinced to never, ever write a script that has hardcoded parameters in it.

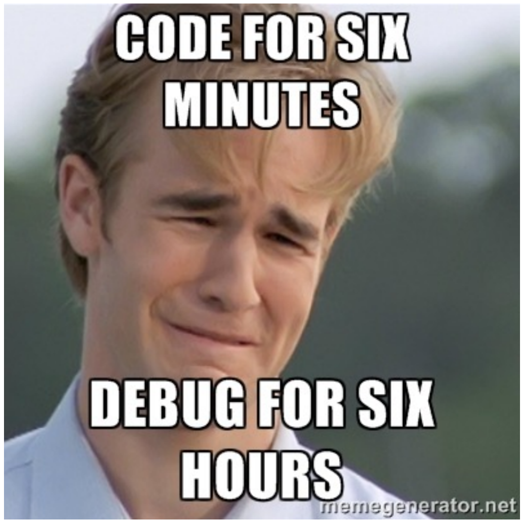
Outline

- Scripts as programs
- Argparse

Data science project phases




YOU DON'T HAVE TO WRITE MAINTAINABLE CODE
IF YOU NEVER MAINTAIN YOUR CODE
imgflip.com


CODE FOR SIX MINUTES
DEBUG FOR SIX HOURS
memegenerator.net

I DON'T ALWAYS WRITE MAINTAINABLE CODE
BUT WHEN I DO I PUT IT ALL IN ONE FILE
memegenerator.net

Data science scripts are code too!

- You should be able to easily figure out what parameters your code expects.
- You should be able to easily change what parameter values your code is using.

(that work together

Think of scripts as mini-programs

most important

1. Write programs to do one thing and do it well
2. Write programs to work together
3. Write programs to handle text streams, because that is a universal interface.

we want smaller programs!

↳ so much easier to maintain

How not to write a script...

hard coding arguments

bad!

command line tools --help

← in here

```
import argparse

parser = argparse.argumentParser()
parser.add_argument("file", help="...")
args=parser.parse_args()
```

← this woud tell you what the arg is used for

Introducing argparse

now use: args.file

Lesson wrap-up

Takeaways

- Scripts are great – but should use argparse

Up next

- Writing unit tests