Unit 3: Core tools

Choosing the Right Tools

Lesson 31

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

- Know what the challenges are when working with CLI and notebooks

Outline

- Strengths/weaknesses analysis

Strengths & weaknesses

Experimental Code	Jupyter	Scripts & CLI
Integrated work	- Code & analysis persistently embedded in same visual context - Linear workflows are easy to capture	- Visuals/results are separate from code - Arbitrary workflows can be captured in scripts
State maintenance	- Easy for data and code to fall out of sync - State is lost when kernel dies	- State persists in files - State must be explicitly maintained
Debugging	- Hard to write good unit tests	- Easy to write unit tests
Long-running compute	- Long running jobs block use of notebook	- Long running scripts can be easily kicked off in parallel and monitored
Collaboration	- Simultaneous editing of notebooks is tricky	- Using version control, is very smooth.
Sharing	- Embedded documentation makes orientation easy	- README & source documentation makes orientation straightforward

Lesson wrap-up

Takeaways

- Both notebook and CLI tools are valuable parts of data science projects
- Use them when their strengths match the activity

Up next

- Dashboards and bokeh