From Jupyter to VSCode

Hamish Gibbs

Jupyter notebooks (.ipynb)

- Google Colab runs a Jupyter Notebook
  - **Pros:**
    - Minimal set-up
    - Interactive coding
  - **Cons:**
    - Difficult to re-use code
    - Insufficient for larger projects
    - Hard to integrate with other programming tools

Python files (.py)

- In the **scripting** vs. **programming** dichotomy:
  - Jupyter Notebooks are better for **scripting.**
- In practice, Python programs are usually composed of **Functions** and **Classes** in `.py` files.
- *Example:* check out the pandas codebase.
- Writing Python like this requires a different development environment.

Advanced Python development

- Enter: Visual Studio Code (VSCode).
  - VSCode is an Integrated Development Environment (IDE).
  - Another (probably familiar) IDE: RStudio!
- IDEs offer an integrated terminal (for running code), auto-complete, debugging, extensions.
- Why VSCode?
  - The best open source IDE (my opinion) with lots of Python extensions.
  - Other Python IDE options: Spyder, PyCharm, Sublime.

Python files (.py)

- Programming in a text editor (not a Notebook):
  - **Pros:**
    - Easy to write re-usable code
    - Can scale to larger projects
    - Easier to collaborate
  - **Cons:**
    - More difficult setup
    - No interactivity by default

Combined workflow

- Can’t we have our cake and eat it too?
- We want:
  - Minimal set-up
  - Interactive coding
  - Easy to write re-usable code
  - Can scale to larger projects
  - Easier to collaborate

Combined workflow

- Combining VsCode with Python plugins, we get:
  - Minimal set-up ❌
  - Interactive coding ✅
  - Easy to write re-usable code ✅
  - Can scale to larger projects ✅
  - Easier to collaborate ✅

Aims: this afternoon

- Unfortunately, VSCode requires a bit more set up (and can cause installation headaches).
  - *That's why I'm here!*
- Aims:
  1. Download VSCode
  2. Download VSCode plugins: Python and Jupyter.
  3. Get Python code to run interactively in code cells: `# %%`.
     - See: Python Interactive window.

Extra

- If you breeze through the VSCode setup process:
  - Finish this morning's tutorials.
  - Then, try to import functions between `.py` files. [hint]
  - Then, try running your `.py` file from the command line. [hint]
  - Then, try to use `if __name__ == "__main__":` in your `.py` file. [hint]