





Next Steps

Computational Text Analysis Week

A Reproducible Research Workshop

Simon Stone

Research Data Services

Dartmouth College





You made it!



Three days of intense Python taming and Text Analysis are behind you!



Where to go from here?

Code local!

🎓 Dartmouth's JupyterHub is a great place to teach, but:

- Computational resources and storage space are limited
- Working with datasets can get tedious (uploading and managing files)
- The Hub gets regularly reset to “factory settings” at the end of a term

💻 For your own projects, we recommend working on your own computer

👍 The following slides will walk you through our *recommended setup*TM



Steps

1. The Python interpreter and standard libraries
2. A code editor: Visual Studio Code
3. Support for Jupyter notebooks
4. A well-organized project folder
5. ???
6. Success!

Python interpreter and standard library

- 🐍 Python is available in many distributions
- 📦 Anaconda, for example, bundles many third-party data science libraries and some additional tools with the official basic Python
- 👎 To get started, we recommend against using such bundles:
 - They can contain a lot of advanced features or libraries you may not need (Anaconda: 5 GB versus “vanilla” Python: 0.09 GB)
 - They can make it more difficult to understand your programming environment
- 👍 **Note:** If you already have Anaconda installed and are comfortable using it, that’s great, too!



Python interpreter and standard library

- Download the official Python distribution for your system:
 - <https://www.python.org/downloads/>
 - Consider using one version older than the most recent one
 - For example, use 3.11 instead of 3.12 (as of December 2023)
 - Not all third-party libraries may have already been made compatible with a brand-new version of Python
- Install as normal for your system

A code editor: Visual Studio Code

- 🧑 A code editor is a text editor with coding-related superpowers
- 🤔 Many such editors are available for Python (Spyder, PyCharm, ...)
- 👉 Our recommendation: Visual Studio Code
 - ❤️ Free, open-source
 - 🌐 Huge user base
 - 🧩 Modular design using extensions (“There is an extension for that!”)
 - 🧰 Simple to use, yet many powerful (but entirely optional) features

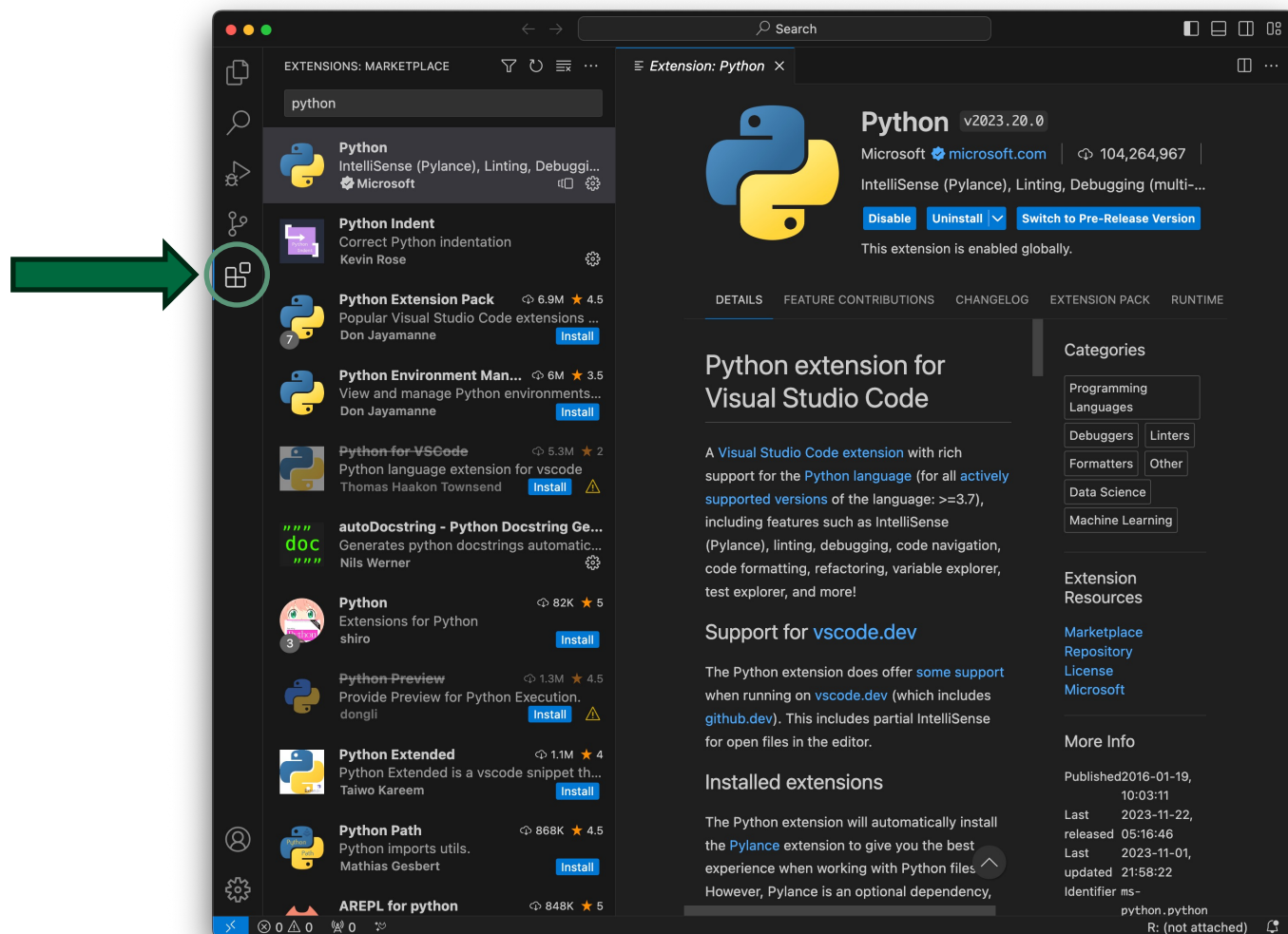


A code editor: Visual Studio Code

- Download the version for your system from the official website:
 - <https://code.visualstudio.com/>
- Install as normal for your system
- Open VS Code from your applications menu (or desktop shortcut)

Support for Python and Jupyter notebooks

- Go to the Extensions tab
- Search for “python”
- Install the Python extension (by Microsoft)
- Search for “jupyter”
- Install the Jupyter extension



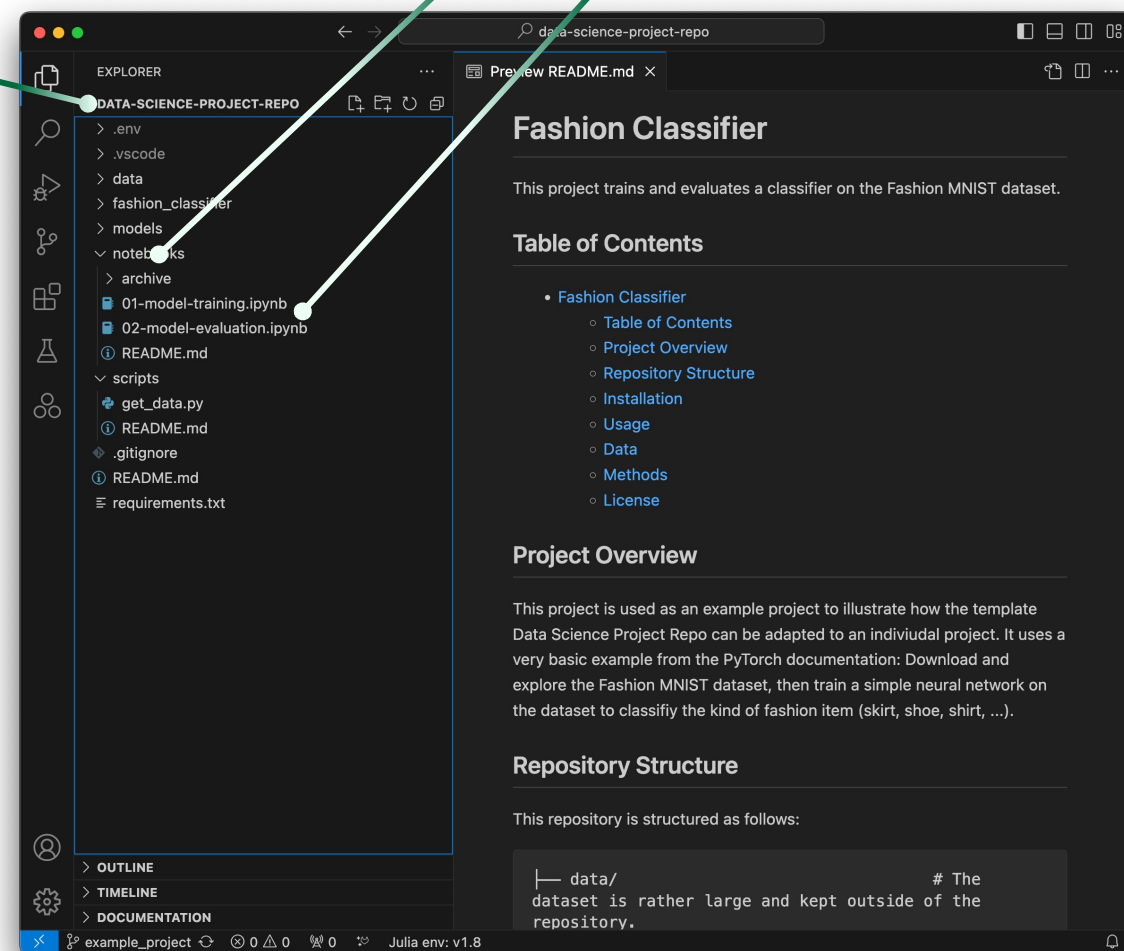
A well-organized project folder

- VS Code works best if you use a dedicated project folder:
 - Create a folder for your project
 - Open the folder in VS Code
 - File -> Open Folder
 - Create subfolders and code files as required

Project folder

Subfolder

Notebooks



Example: <https://dartgo.org/rds-example-project>



More next steps



Setting up a virtual environment for your project



Debugging Python in VS Code

(watch this space for an upcoming workshop on this topic!)



Automatically format your code (and flag potential problems)



More on our Research Guide



About the Reproducible Research Group

- Joint venture of **Research Computing @ ITC** and **Research Data Services @ Library**
- Consult with **experts** on
 - research data management,
 - data visualization,
 - biomedical research support,
 - spatial data and GIS,
 - high performance and research computing,
 - statistical analysis,
 - economics and social sciences data
- **Meet** the people on campus that support your reproducible research lifecycle
- **Engage** in community discussions to learn from other researchers on campus
- Attend a workshop to **learn** practical tools and tips



About Research Data Services

Research Data Management

Data Management Plans (DMPs) for sponsored projects

Finding and using 3rd party data

Collection and cleaning of data

Organization and documentation

Publishing and Repositories

Data Analysis/Visualization

Textual, numeric, spatial data

Reproducible research workflows

Scripting in R: tidyverse core package (i.e. ggplot, dplyr, tydr, tibble, etc.)

Scripting in Python: NumPy, SciPy, Pandas, Scikit-learn, Matplotlib, Seaborn, (OpenCV, PyTorch, TensorFlow, Tesseract, NLTK, etc.)

Computational Scholarship

Computational project planning

Collections as Data

Storytelling with data and visualizations

Text and data mining

Digital Humanities support

Computational Pedagogy



Work with us

ResearchDataHelp@groups.dartmouth.edu

Jeremy Mikecz
Research Data Science Specialist
jeremy.m.mikecz@dartmouth.edu
dartgo.org/jeremyappts

Simon Stone
Research Data Science Specialist
simon.stone@dartmouth.edu
dartgo.org/meetwithsimon

Lora Leligdon
Head of Research Data Services
lora.c.leligdon@dartmouth.edu
dartgo.org/lora



Thank you!



<https://www.library.dartmouth.edu/research-data-services>