Introduction to **Applied Data Science in Python**

QBS 101.5

Simon Stone[✤], Jeremy Mikecz [✤], Carly Bobak[✦]

[✤]Research Data Services, [✦]Research Computing

Dartmouth College

Applied Data Science in Python
Instructors

Carly Bobak, PhD

- Biomedical Informatics Specialist
- PhD in Quantitative Biomedical Sciences from Dartmouth College (2021)
- Teaches QBS 181: Data Wrangling
- Published papers on diverse public health issues, including TB, Covid, HIV, cancer, sepsis, smoking cessation, and obesity.
- Has two dogs who tend to crash QBS events

Jeremy Mikecz, PhD

- Research Data Science Specialist
- PhD in Latin American History, University of California
- Offers consultations and workshops on computational text analysis, data visualization, GIS and mapping, Python & R programming
- Published various articles and a book (in-press) showcasing how visualization and mapping can aid historical research

Dr. Simon Stone

- Research Data Science Specialist
- Doctoral degree in Electrical and Computer Engineering from Technische Universität Dresden
- Offers consultations and workshops on Data Science, Machine Learning, and Software Development
- Published papers on speech technology, sensor design, signal processing, and machine learning
- Freelance software developer by night

Applied Data Science in Python
Course materials

- Brand new class!
- Materials are created as we go along
- You can help shape them!
- Materials will consist of lecture slides and code-along notebooks

Class repository:

https://github.com/Simon-Stone/qbs-applied-data-science

Get the materials:

```
git clone https://github.com/Simon-Stone/qbs-applied-data-science
```

Update the materials:

```
git pull
```

Introduction
Data Science

Challenge:

Define *Data Science*!

Data science is the study of data to extract meaningful insights for business.

https://aws.amazon.com/what-is/data-science/

aws

Data science [...] uncover[s] actionable insights hidden in an organization's data.

https://www.ibm.com/topics/data-science

IBM

[Data science] models and analyzes key data to continually improve how businesses utilize data.

https://cloud.google.com/learn/training/machinelearning-ai

Google

Introduction

Data Science

Data Science is OSEMN!*

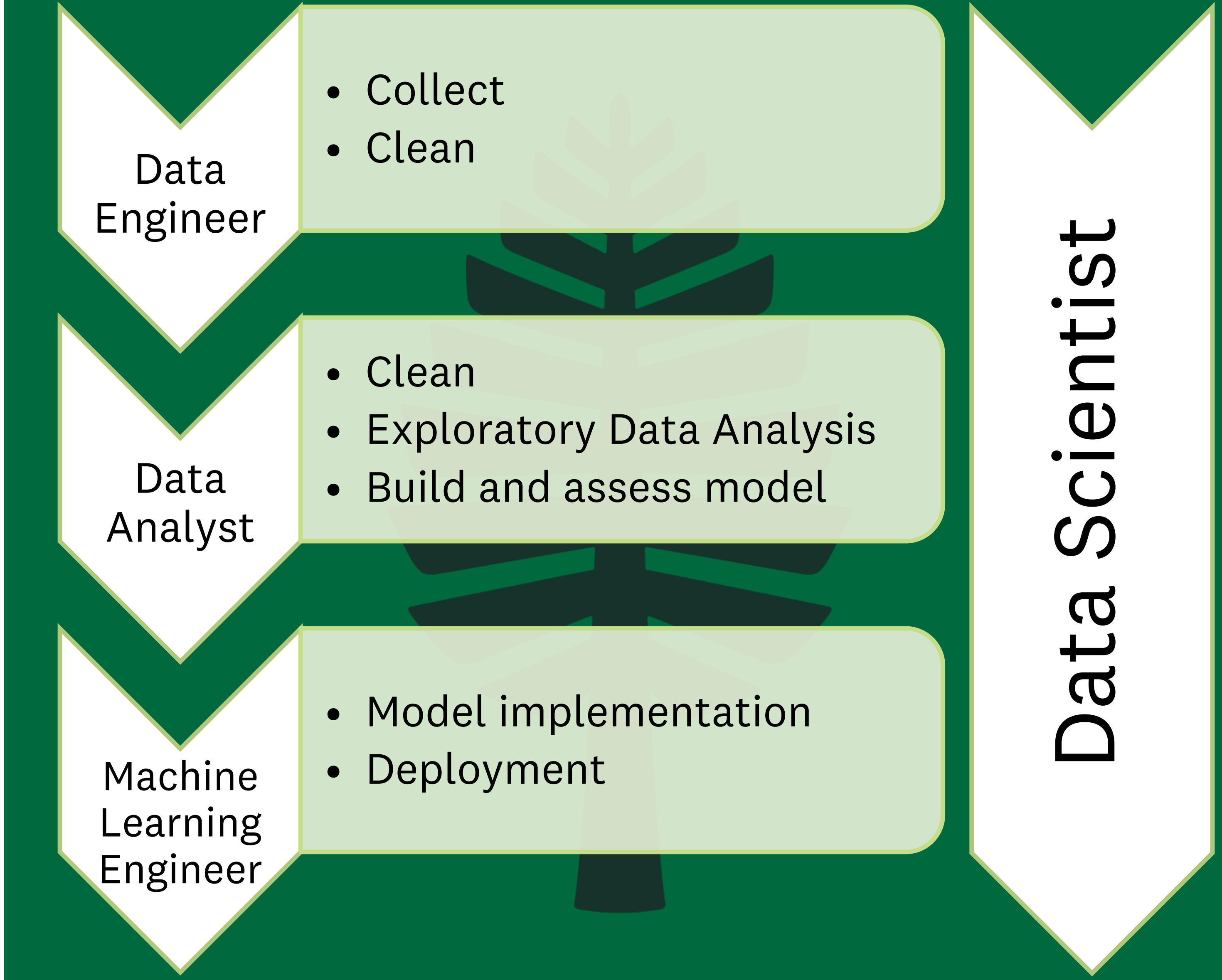
📥 Obtain

🧼 Scrub

🔎 Explore

🤖 Model

⁉️ iNterpret



Data Engineer
- Collect
- Clean

Data Analyst
- Clean
- Exploratory Data Analysis
- Build and assess model

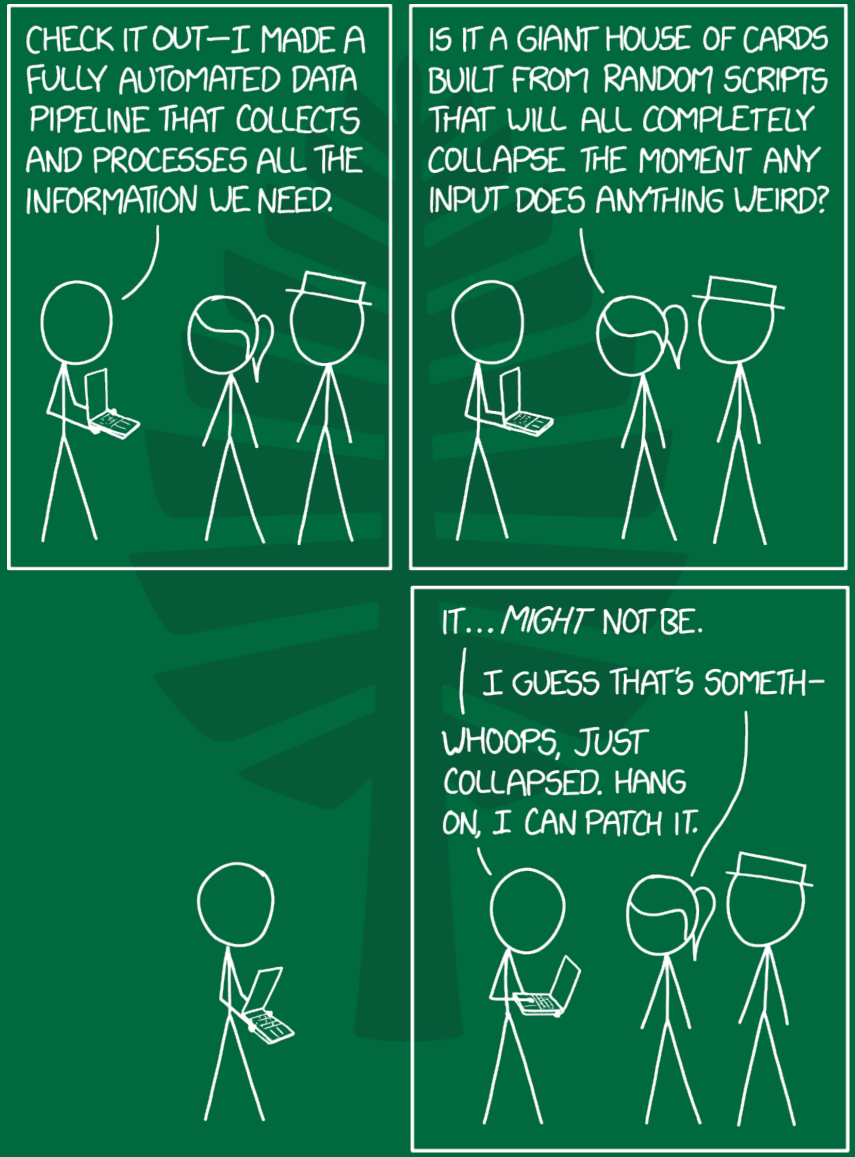
Machine Learning Engineer
- Model implementation
- Deployment

Data Scientist

*pronounced “awesome” - /ˈɔ.səm/
https://www.datascience-pm.com/osemn

Outline

- 🎬 Introduction
- 📅 Schedule
- 🎓 Learning Objectives
- ❄️ What makes a Data Science project special?
- 🐍 What makes a Python project special?
- 🛠️ Tools of the Trade
- 🥜 Summary
- 💻 Assignment


CHECK IT OUT—I MADE A FULLY AUTOMATED DATA PIPELINE THAT COLLECTS AND PROCESSES ALL THE INFORMATION WE NEED.
IS IT A GIANT HOUSE OF CARDS BUILT FROM RANDOM SCRIPTS THAT WILL ALL COMPLETELY COLLAPSE THE MOMENT ANY INPUT DOES ANYTHING WEIRD?
IT… MIGHT NOT BE.
I GUESS THAT'S SOMETH–
WHOOPS, JUST COLLAPSED. HANG ON, I CAN PATCH IT.

Munroe, R. (n.d.). *Data Pipeline*. xkcd. https://xkcd.com/2054/ (CC BY-NC 2.5)

Learning Objectives

1. **Complete a data science project** through the process of finding, cleaning, compiling, exploring, analyzing, visualizing, and modeling data.
2. **Become proficient at writing and applying Python** code to complete these data science steps.
3. **Practice effectively communicating** (and providing critical feedback on) data science research.
4. Apply the principles of **reproducible research** throughout the data science lifecycle.

Format

Class times and location:

- Monday, 4 pm – 5:30 pm
- Wednesday, 4 pm – 5:30 pm
- Remsen 312

- Classes will consist of lectures, code-alongs, hands-on project work, or a mix of all the above
- There will be assignments to continue the engagement with each week’s content
- At the end of each Wednesday class, there is time for project check-ins, show & tell, or troubleshooting
- Each student is expected to hand in a final project

Format
Final Project

Project requirements:

- Devise a business or research question that can be answered through data
- Identify a suitable dataset
- Implement all major steps of a data science project to find an answer to the chosen question

Deliverables:

- Project pitch (details to come)
- Fully reproducible project repository
- Final presentation

Schedule

Week 1 (June 26 & June 28):
Getting Started with Applied Data Science & Python

- Introduction to Applied Data Science with Python
- Data Wrangling

Week 2 (July 3 & July 5):
Texts, Maps, and Graphs

- Data Visualization
- NLP I

Week 3 (July 10 & July 12):
Texts as Data

- NLP II
- NLP III

Week 4 (July 17 & July 19):
Machine Learning

- Scikit-learn
- PyTorch

Week 5 (July 24 & July 26):
Project Work

- Project proposals due (Monday July 24)
- *No class meetings*

Week 6 (July 31 & August 2):

- *No class meetings*

Week 7 (August 7 & August 9):
Collaboration and Documentation

- Documenting, Sharing and Evaluating Code
- Peer Review

Week 8 (August 14 & August 16):
Code Revision Work

- *No Class Meetings*

Week 9 (August 21 & August 23):

- Final Presentations and revised code due (Wed. Aug 23)

Policy on Generative AI

Use of generative AI is generally accepted in this course if it meets the following requirements:

- You are expected to take full ownership of your work, including the use of generative AI
- You need to be able to understand and reflect any output of generative AI you want to include in your work
    - “It’s a tool, you’re its master!”
- Just like with code taken from other sources (e.g., Stack Overflow), any code other than your own must be cited (see Week 7: Documentation)
- You are expected to be able to explain and discuss every line of code in your final project

Applied Data Science with Python
What makes a Data Science project special

Typical project has **two phases**:

1. Exploration phase
   - 🧭 Explore data, imputations, feature engineering, model selection
2. Reporting and “deployment” phase
   - 📊 Generate result tables and graphs, refactor code, export trained models

Subject matter domains differ, tasks are similar

- 🤹 Multiple stakeholders to consider, “interrogate”, and convince

Applied Data Science with Python

What makes a Python project special

💪 Powerful language with incredible community and industry support

- Often large number of (inter-)dependencies

🚀 Rapid prototyping

🖥️ (Mostly) cross-platform compatible on Windows, Mac OS, Linux

😵‍💫 Sprawling, sometimes confusing ecosystem

😫 Deployment / reproducibility can be challenging

Applied Data Science with Python
Tools of the Trade

- Integrated Development Environment (IDE) for writing code
- Command line scripts for project setup and configuration
- Jupyter notebooks for prototyping and reporting
- Version control for code (and sometimes data)

Not part of this course:

- Cloud infrastructure (Amazon Web Services, Google Cloud, Microsoft Azure)

Tools of the Trade
Anti-patterns

“A commonly-used process, structure or pattern of action that, despite initially appearing to be an appropriate and effective response to a problem, has more bad consequences than good ones.”

*Gang of Four**

* Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.. ISBN: 0-201-63361-2

- Not using an IDE
- Not organizing your project in one place
- Using a too simple or too elaborate project setup
- Overusing Jupyter notebooks (sometimes scripts are better)
- Not managing your Python environment

Tools of the Trade

Recommended Setup for this class

Recommended **Python distribution**:

🐍 Vanilla Python 3.8 or newer: https://www.python.org/

Recommended **IDE**:

📝 Visual Studio Code: https://code.visualstudio.com/

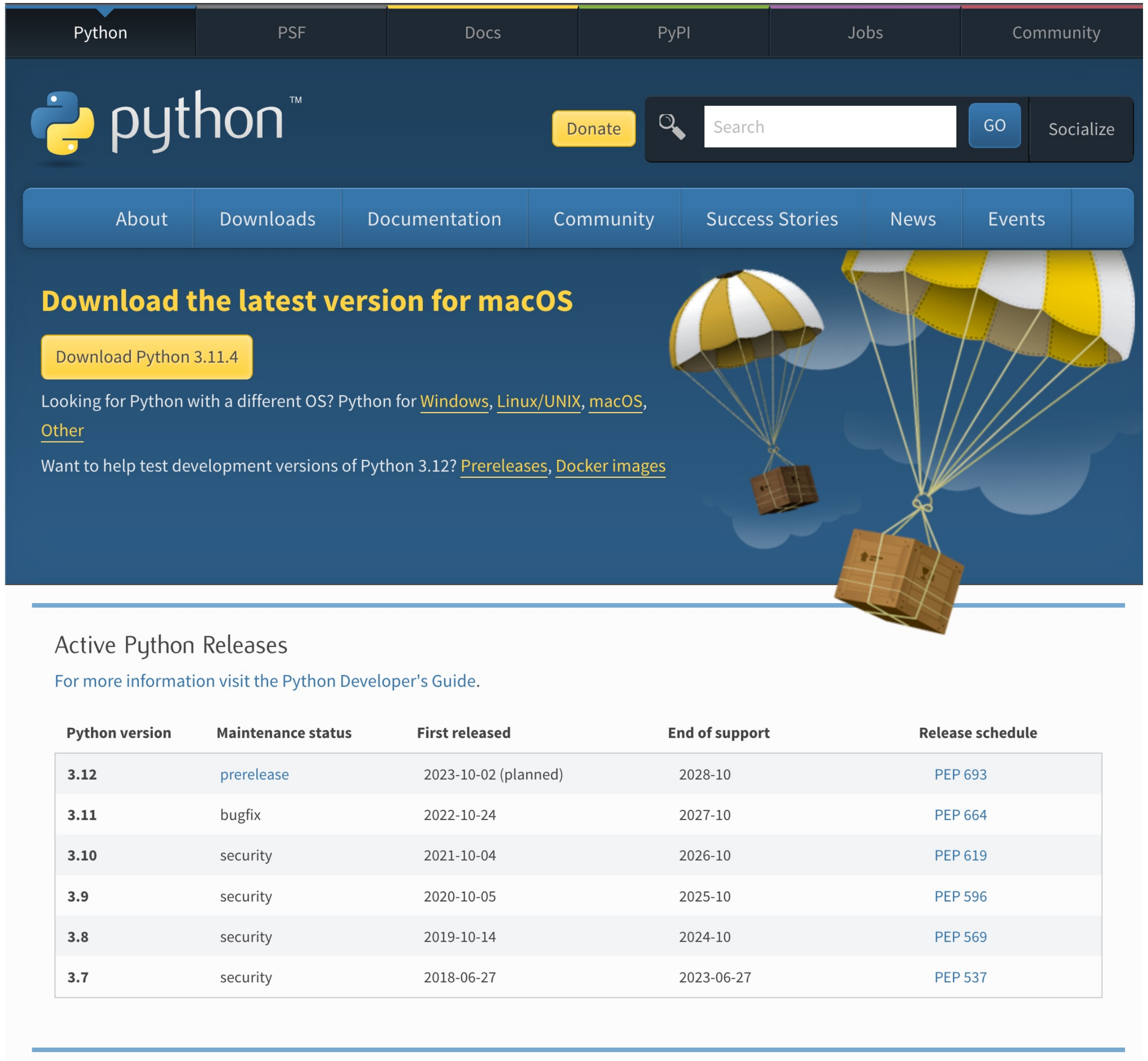
Recommended **version control** hosting platform

GitHub: https://www.github.com

Tools of the Trade
Python

- Recommended minimum **version 3.8+**
- Installer from python.org recommended
- Recommended **environment management:**
  - Using Python’s built-in virtualenv: https://docs.python.org/3/library/venv.html
  - Install packages into virtual environment
  - Document project requirements using:

```
pip freeze > requirements.txt
```



Download the latest version for macOS

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python 3.12? Prereleases, Docker images

Active Python Releases

For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support	Release schedule
3.12	prerelease	2023-10-02 (planned)	2028-10	PEP 693
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	security	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537

https://www.python.org/downloads/

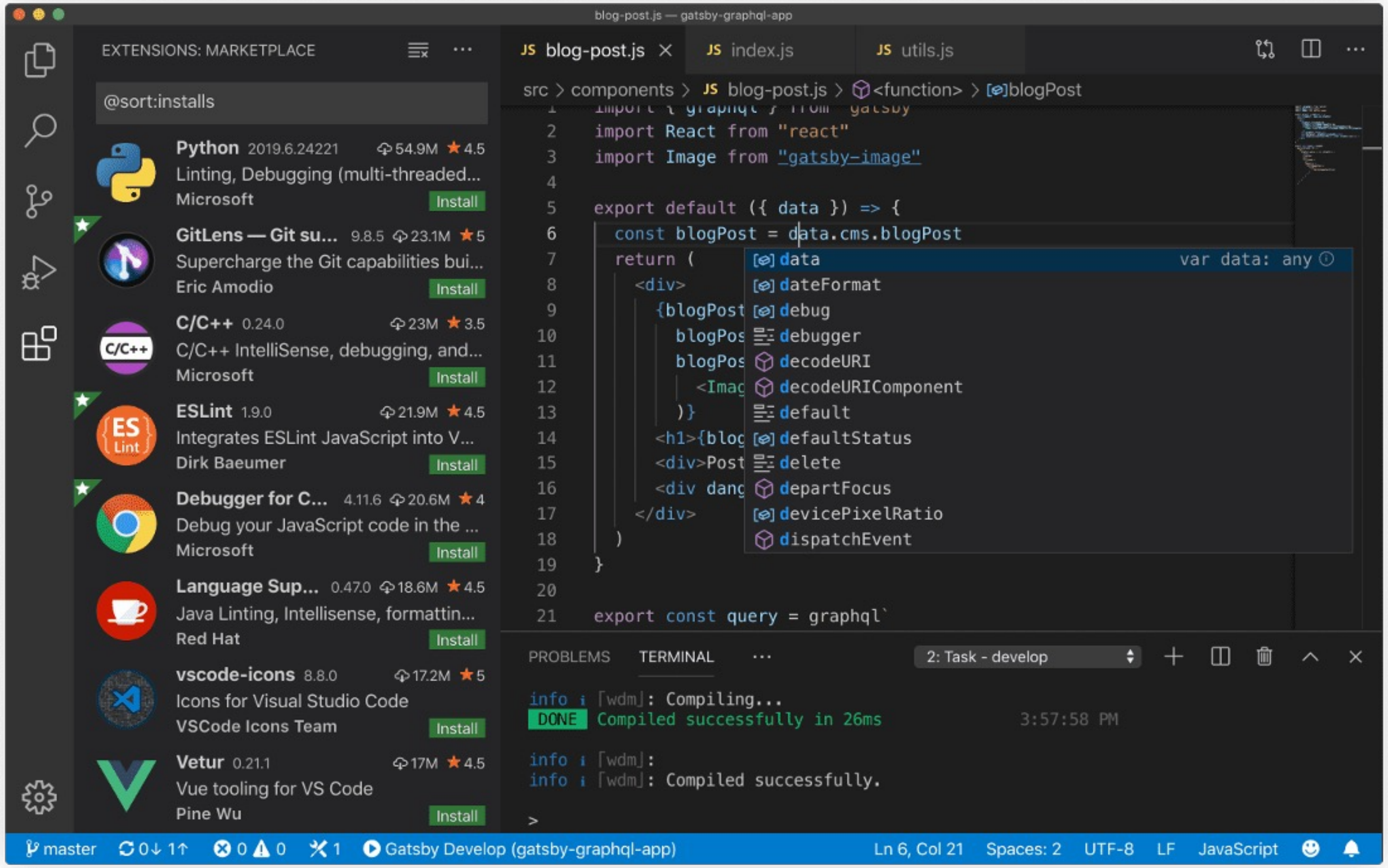
Tools of the Trade

Python

- Python is supported by Jupyter notebooks
- Notebooks are a convenient way to combine code, text, and images in a single file
- Notebooks are great for developing a narrative in your analysis
- Notebooks are not a good choice for project setup, config, or model deployment
- Use regular Python scripts (*.py) for that

Tools of the Trade
Visual Studio Code

- Cross-platform (Windows, Mac, Linux)
- Open source
- Lightweight (no project files required)
- Support for all sorts of languages
- Syntax highlighting and autocomplete
- Extensible in millions of ways
- Natively supports Jupyter notebooks
- Git built-in
- Easy to get started
- Many powerful advanced features



blog-post.js — gatsby-graphql-app

EXTENSIONS: MARKETPLACE

@sort:installs

Python 2019.6.24221 54.9M ★4.5
Linting, Debugging (multi-threaded...
Microsoft Install

GitLens — Git su... 9.8.5 23.1M ★5
Supercharge the Git capabilities bui...
Eric Amodio Install

C/C++ 0.24.0 23M ★3.5
C/C++ IntelliSense, debugging, and...
Microsoft Install

ESLint 1.9.0 21.9M ★4.5
Integrates ESLint JavaScript into V...
Dirk Baeumer Install

Debugger for C... 4.11.6 20.6M ★4
Debug your JavaScript code in the ...
Microsoft Install

Language Sup... 0.47.0 18.6M ★4.5
Java Linting, Intellisense, formattin...
Red Hat Install

vscode-icons 8.8.0 17.2M ★5
Icons for Visual Studio Code
VSCode Icons Team Install

Vetur 0.21.1 17M ★4.5
Vue tooling for VS Code
Pine Wu Install

blog-post.js × | index.js | utils.js

src > components > blog-post.js > <function> > blogPost

```
import { graphql } from "gatsby"
import React from "react"
import Image from "gatsby-image"

export default ({ data }) => {
  const blogPost = data.cms.blogPost
  return (
    <div>
      {blogPost
        blogPos
        blogPos
          <Imag
        )}
      <h1>{blog
      <div>Post
      <div dang
    </div>
  )
}

export const query = graphql`
```

```
data                var data: any
dateFormat
debug
debugger
decodeURI
decodeURIComponent
default
defaultStatus
delete
departFocus
devicePixelRatio
dispatchEvent
```

PROBLEMS | TERMINAL | 2: Task - develop

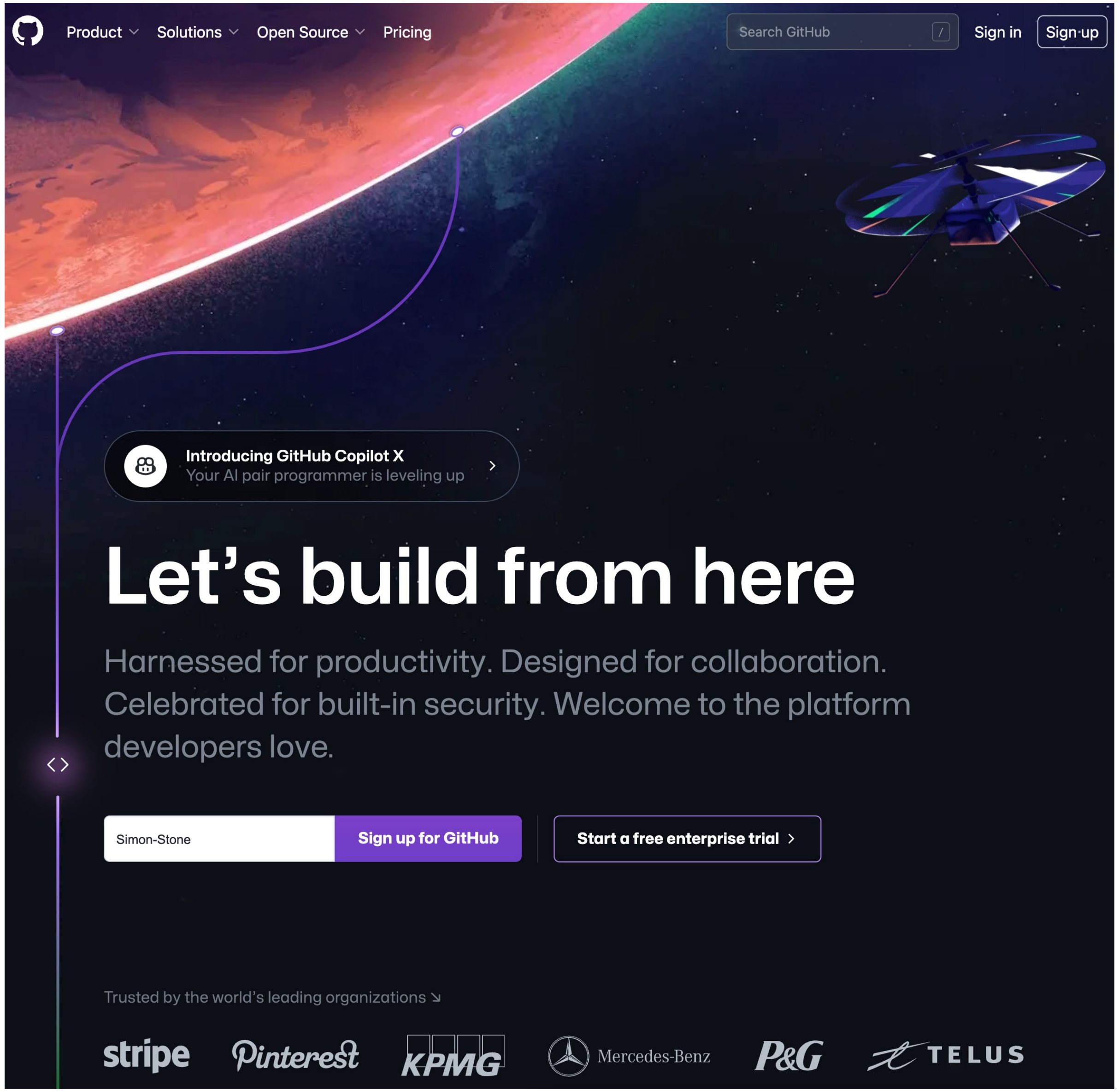
```
info ℹ ｢wdm｣: Compiling...
 DONE  Compiled successfully in 26ms                    3:57:58 PM

info ℹ ｢wdm｣:
info ℹ ｢wdm｣: Compiled successfully.

>
```

master 0↓ 1↑ ⊗ 0 ⚠ 0 ✕ 1 Gatsby Develop (gatsby-graphql-app) Ln 6, Col 21 Spaces: 2 UTF-8 LF JavaScript

https://code.visualstudio.com/

Tools of the Trade
GitHub

- Web-based platform for individuals and teams to collaboratively develop software projects
- Uses the Git version control system
- Also offers tools and services for
  - project management,
  - continuous integration and deployment,
  - documentation
- Building up portfolio of projects on GitHub can benefit your job applications



Let's build from here

Harnessed for productivity. Designed for collaboration. Celebrated for built-in security. Welcome to the platform developers love.

Trusted by the world's leading organizations ↘

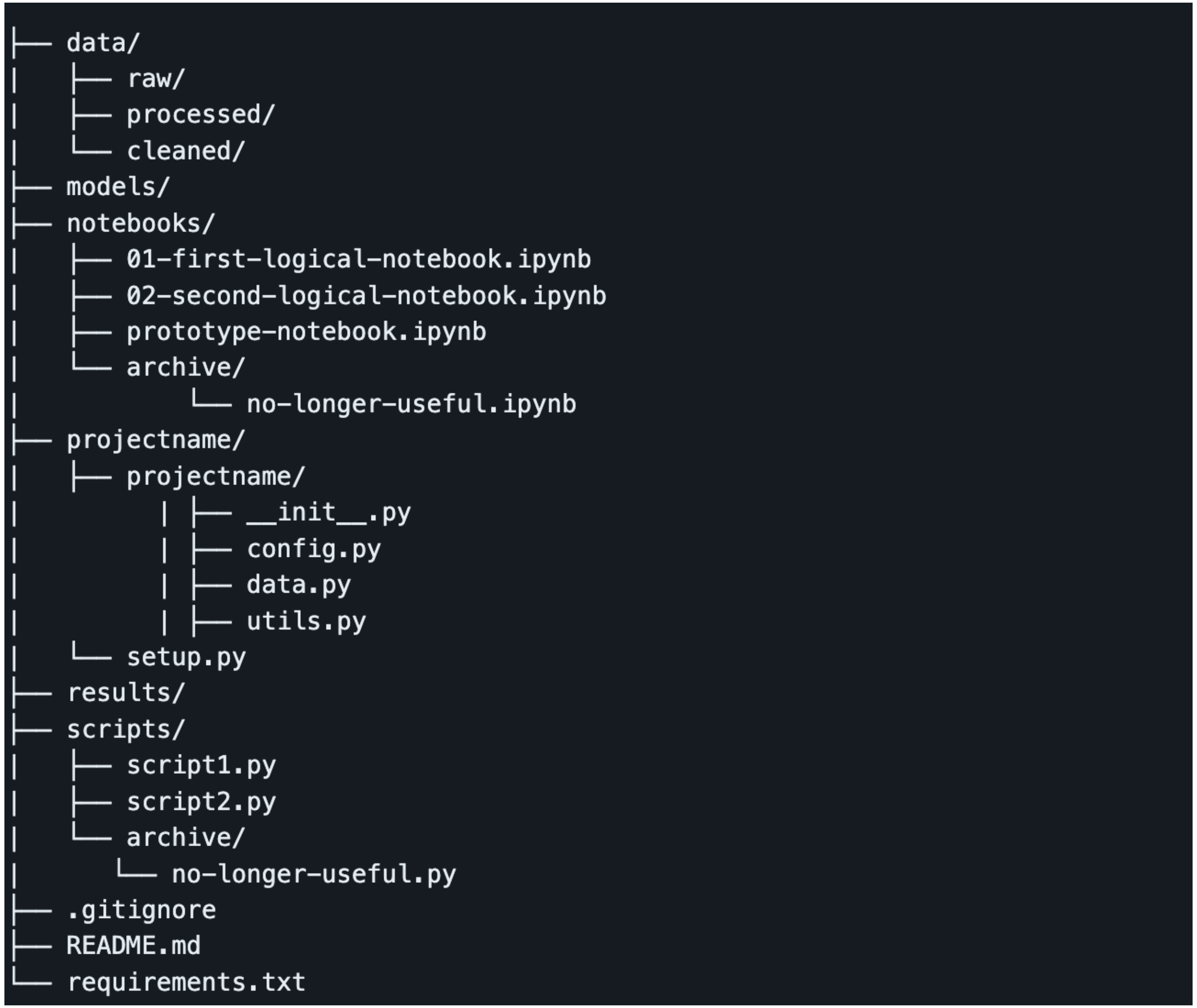
https://www.github.com/

Tools of the Trade
GitHub

- Repository structure:
  - Many templates available on the internet:
    - http://drivendata.github.io/cookiecutter-data-science/
    - https://gist.github.com/ericmjl/27e50331f24db3e8f957d1fe7bbbe510
- Try them out, learn what works, adjust and adapt to fit your flow
- Consider creating a GitHub template repo

Tools of the Trade
GitHub

- An example structure you can use in this class:
- Tour of the example repo
- Not all of this will be helpful in every project



```
├── data/
|   ├── raw/
|   ├── processed/
|   └── cleaned/
├── models/
├── notebooks/
|   ├── 01-first-logical-notebook.ipynb
|   ├── 02-second-logical-notebook.ipynb
|   ├── prototype-notebook.ipynb
|   └── archive/
|           └── no-longer-useful.ipynb
├── projectname/
|   ├── projectname/
|         | ├── __init__.py
|         | ├── config.py
|         | ├── data.py
|         | ├── utils.py
|   └── setup.py
├── results/
├── scripts/
|   ├── script1.py
|   ├── script2.py
|   └── archive/
|      └── no-longer-useful.py
├── .gitignore
├── README.md
└── requirements.txt
```

https://github.com/Simon-Stone/data-science-project-repo

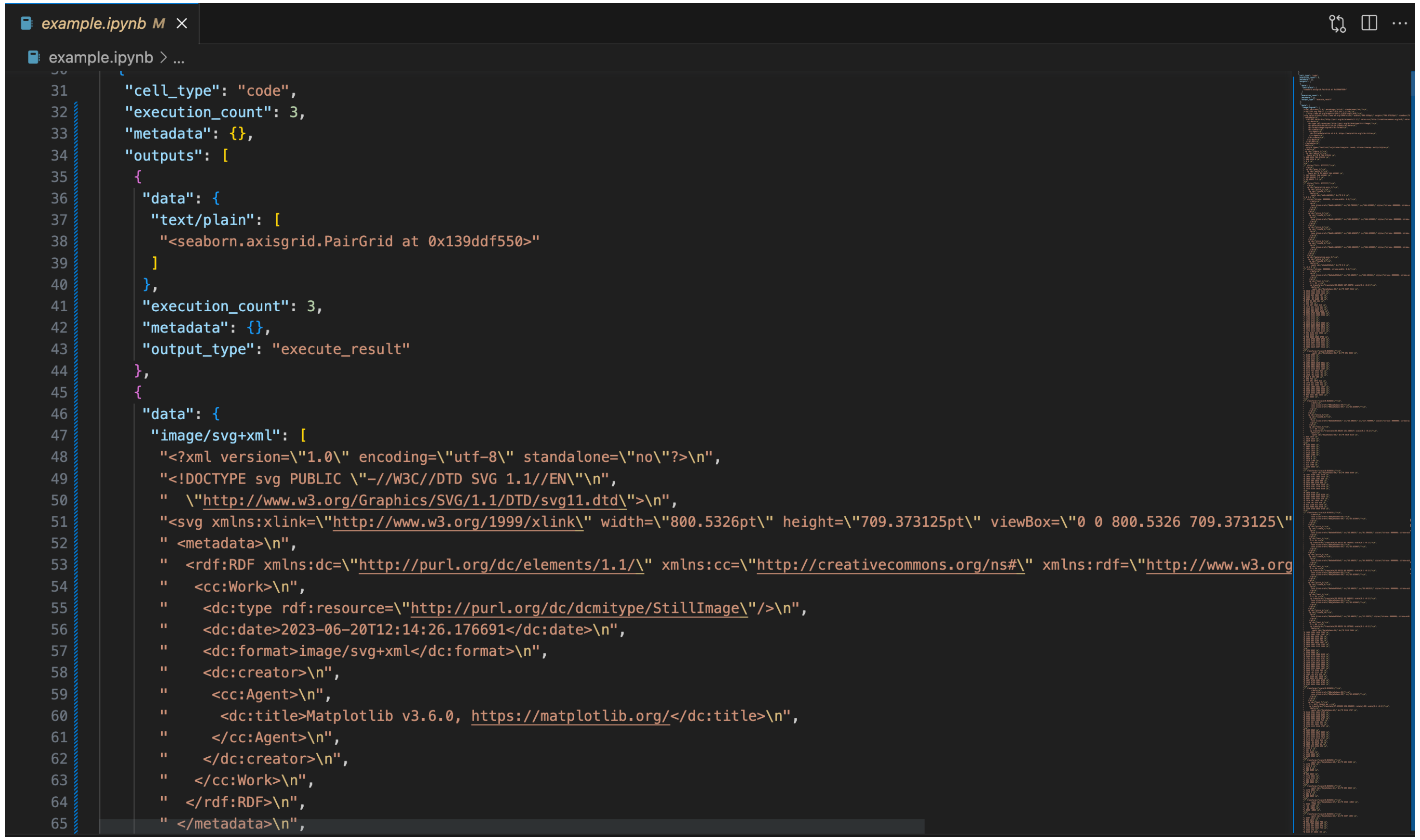
Tools of the Trade
Bonus tip: Jupyter notebooks in Version Control

The problem:

- Jupyter notebooks contain metadata, binary blobs, or other artifacts from execution
- Git tracks changed lines in text files
- Diffs become almost meaningless

Solution:

- Always clear the output before committing
- Use pre-commit hooks
- Use `nbstripout` (https://github.com/kynan/nbstripout)



```
"cell_type": "code",
"execution_count": 3,
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "<seaborn.axisgrid.PairGrid at 0x139ddf550>"
   ]
  },
  "execution_count": 3,
  "metadata": {},
  "output_type": "execute_result"
 },
 {
  "data": {
   "image/svg+xml": [
    "<?xml version=\"1.0\" encoding=\"utf-8\" standalone=\"no\"?>\n",
    "<!DOCTYPE svg PUBLIC \"-//W3C//DTD SVG 1.1//EN\"\n",
    "  \"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd\">\n",
    "<svg xmlns:xlink=\"http://www.w3.org/1999/xlink\" width=\"800.5326pt\" height=\"709.373125pt\" viewBox=\"0 0 800.5326 709.373125\"
    " <metadata>\n",
    "  <rdf:RDF xmlns:dc=\"http://purl.org/dc/elements/1.1/\" xmlns:cc=\"http://creativecommons.org/ns#\" xmlns:rdf=\"http://www.w3.org
    "   <cc:Work>\n",
    "    <dc:type rdf:resource=\"http://purl.org/dc/dcmitype/StillImage\"/>\n",
    "    <dc:date>2023-06-20T12:14:26.176691</dc:date>\n",
    "    <dc:format>image/svg+xml</dc:format>\n",
    "    <dc:creator>\n",
    "     <cc:Agent>\n",
    "      <dc:title>Matplotlib v3.6.0, https://matplotlib.org/</dc:title>\n",
    "     </cc:Agent>\n",
    "    </dc:creator>\n",
    "   </cc:Work>\n",
    "  </rdf:RDF>\n",
    " </metadata>\n",
```

Assignment

- Set up a skeleton project repository
- Start thinking about a ”business/research question” you want to explore in your project
- Start thinking about the kind of data you want to work with
- Start looking for corresponding data
  - Is it already available as a dataset? API? Possible and allowed to scrape?

Thank you