Scientific Data Analysis, Data Science, and Machine Learning in Python

An independent study with two parts:

Part 1: Consolidate and advance our understanding of mainstream and cutting edge scientific data analysis techniques using Chapters 1-8 of Pasha, Astronomical Python

Part 2: Survey data science techniques up to and including neural net and deep learning implementations (which are the relevant preparation for a subsequent study of natural-language processing and LLMs) using Chapters 1-11, 13-15, and 18-19 Grus, Data Science from Scratch, 2nd Ed.

Term 6 of Academic Year 2024-2025, Deep Springs College

Mentor: **Prof. Brian Hill**

Student: Hexi Jin (DS 23)

Materials

Required

- Imad Pasha, *Astronomical Python*
  - Pasha's examples use data hosted at **https://zenodo.org/records/10732223**
- Joel Grus, *Data Science from Scratch, 2nd Edition*
  - We may need to copy over resources from Grus's GitHub repo for the book **https://github.com/joelgrus/data-science-from-scratch**

Optional

- Both Pasha and Grus include adequate introductions to Python features as they use them, but you may want a more systematic introduction to use as a reference. An excellent one is David Beazley, *Python Distilled.* It is actually a distillation and update of his time-tested *Python: Essential Reference,* which was growing overly-long as the Python language feature set kept growing.

Actual Daily Schedules (Kept Retrospectively)

- **Daily Schedule - Part 1**
- **Daily Schedule - Part 2**

Daily Schedule Part 1 (Actual — Kept Retrospectively)

Regular meeting schedule is Wednesdays and Saturdays, 11:00-12:00

Back to **Course home page**

Part 1: Scientific Python (using Imad Pasha, *Astronomical Python*)

Part 1 Uses Pasha and lasts for the first three weeks of Term 6

Week 1 — Shell and Python Quick-Start/Review

- May 16 — Complete Chapters 1 to 3: Unix (shell) Basics, Installing Python, and the Astronomy/Scientific Data Analysis Stack — Problem Set 0: Get Anaconda downloaded and installed and use the IPython interface — Discussed Python language features, syntax, and style (PEP 8), differences between Windows and Unix shells, globbing, and Python's Operating System Insulation Layer (OSIL)
- May 17 — Complete Chapter 4: Introduction to Python — Problem Set 1: Use for loops to compute the first 20 Fibonacci numbers (screenshot your solution in IPython) — Discussed notebook tools and IDEs

Week 2 Matplotlib and Numpy

- May 21 — Complete Chapter 5: Visualization with Matplotlib — Install (if not already part of your Python distribution) and start working in Jupyter Lab — Problem Set 2: Make some histogram and scatter plots using the **iris dataset** (save your plots as a Jupyter Lab notebook)
- May 25 — Complete Chapter 6: Numerical Computing with NumPy — Create a github account, fork the repo: brianhill/scientific-data-analysis — Then figure out how to get a local copy onto your machine of your fork (`hexijin/scientific-data-analysis`) and this will involve installing git on your machine (which will be different for Mac or Windows) — Started learning shell access to git, and the add, commit, push cycle (which we will be adding more to once that is routine)

Week 3 — SciPy and AstroPy

- May 28 — Complete Chapter 7: Scientific Computing with SciPy — Problem Set 3 (in addition to working through all the code in the chapter): Do Exercise 7.1 — Introduced the linear algebra concepts and notation for column vectors, row vectors, and matrix and vector multiplication
- June 1 — Complete Chapter 8: Astropy and Astronomical Packages — As Problem Set 4 (in addition to working through all the code in the chapter): Do the Chapter 8 exercises — Finally, it's time to add to your git knowledge the ideas of origin and upstream, and a second cyle of operations: how to fetch from upstream (my GitHub repo), rebase (in your local repo), and push your rebased changes to your origin (your GitHub fork of my repo)

See also **Daily Schedule - Part 2**

Daily Schedule Part 2 (Actual — Kept Retrospectively)

Regular meeting schedule is Wednesdays and Saturdays, 11:00-12:00

Back to **Course home page**

See also **Daily Schedule - Part 1**

Part 2: Data Science Foundations (using Joel Grus, *Data Science from Scratch, 2nd Edition*)

Part 2 Uses Grus and lasts for the remaining three-and-a-half weeks of Term 6

Week 4 — Yet Another Review of Python — Some Vector and Matrix Algebra — Statistics and Probability

- June 4 — Chapters 1-3: Another excellent review of Python and Matplotlib which will help systematize your understanding of the language features you were using in Pasha's book — The assignment is to do the review of the three chapters, but to completely stop using Jupyter or Jupyter lab, and instead get everything working in PyCharm Professional Edition (free for students) or VS Code (but I have zero experience with that) — When Grus says (at the beginning of Chapter 2) that you should not be tampering with your base Python environment, he is completely correct (so learn how to make a venv that you could call grus or dsfs and then switch to it — if you didn't already do that for working through Pasha)
- June 7 — Chapters 4-6: Linear Algebra (wherein Grus introduces his Vector and Matrix implementations which could have been classes, or could have leveraged numpy, but which he craftily used type aliases, because that was the simplest way to implement from scratch), Statistics, and Probability (due to having taken last fall's Bayesian Statistics class, the math in Chapters 5 and 6 will be review)

Week 5 — Optimization (aka Minimization and Maximization) — Working with Data

- June 11 — Chapters 7 and 8: Hypotheses & Inference and Gradient Descent — Make a local repo from the magic hexijin.github.io GitHub repo, put an index.md file in it, and then push to origin main — The only remaining step to having **your own home page** is to enable GitHub pages in this repo — For more advanced reading, Grus recommends this **Overview of Gradient Descent** by Eric Ruder
- June 15 — Chapters 9 and 10: Getting and Working with Data (including subtracting the mean and dividing by the standard deviation to get rescaled data sets, and a load of utilities for doing principal component analysis, that Grus somewhat-too-rapidly introduced at the end of Chapter 10)

Week 6 — Machine Learning — Linear Regression

- June 19 — Chapters 11 and 13: Machine Learning and Naive Bayes: Machine Learning and Naive Bayes (and you may need to pick up some material from Chapter 12 on k-Nearest Neighbors which we are otherwise skipping)
- June 21 — Chapters 14 and 15: Simple Linear Regression and Multiple Regression

Week 7 — Neural Networks — Deep Learning

- June 21 — Chapters 18 and 19: Neural Networks and Deep Learning (in the interest of getting to Neural Networks and Deep Learning in our final week, we are skipping Chapters 16 and 17 on Logistic Regression and Decision Trees) — Also as part of this week's material, do this live coding session to see how a real pro codes, including type-hinting, systematic adherence to style choices, and code testing: **Joel Grus - Building a Deep Learning Library** (build the code in PyCharm as Grus builds it in VS Code, pausing the live coding demonstration whenever you need to catch up with him) — This live coding session is effectively a blindingly-fast version of Chapters 18 and 19