

Course Title: Introduction to Scientific Python

Course Number: G440

Credits: 1

Course Directors: Teon Brooks, PhD, brookst3@mskcc.org

Course Prerequisites: None; Open to first year Engineering PhD students and select senior PhD students as approved.

Grading Policy: Pass/Fail

Course Description and Learning Objectives

This curriculum has been structured to emphasize the foundations in Python and its four core scientific computing libraries: `numpy`, `scipy`, `pandas`, and `matplotlib`. This course will also highlight packages used for statistical analyses and machine learning. This course wraps up with a capstone project to bring together all of these concepts in a practical and applied way.

Course Structure

The course meets for 10 sessions from 1:30 pm – 3:00 pm. The final project session will also run from 1:30 pm – 3:00 pm. Changes to that schedule are communicated to students via email. Active learning and in-class programming exercises will be equally used and integrated to complement traditional lectures. Additionally, there will be preparatory session on September 9, 2026.

Teaching Fellows

Teaching Fellows, drawn from senior GSK students and the postdoctoral community at MSK, are present in the course sessions. Their role is to act as an additional source of information/assistance, to help keep the discussion sessions moving.

Assignments and Methods for Assessing Student Achievement

This course is graded on a pass/fail basis; class participation, homework assignments, and a final project will form the basis of the grading. A weekly assignment will be given throughout the course to reinforce material.

Course Evaluation

Students are expected to complete surveys regarding the lectures and overall course via their student portal. This feedback will be used to evaluate the effectiveness and relevance of the topics and provide direction for the subsequent iterations of the course.

Academic Dishonesty, Plagiarism and Artificial Intelligence

The Policy can be found in the [Student and Faculty Handbook](#) linked on the GSK Website.

Generative AI and Agentic Programming Tools

The first four weeks of this course are focused on building your foundational understanding of Python and git. It is crucial to your learning that you do this without the use of an agentic programming tool (e.g., Claude Code, GitHub Copilot, Cursor). We will have a class dedicated to using an agentic programming tool, but working effectively with one requires that you first understand the fundamentals of programming.

Getting Started

miniforge

For this class, we will be installing Python on our computers. This will give us maximum flexibility and this setup can be used for future courses and labwork. Before class, please follow the instruction for installing the conda-forge distribution of Python ([conda-forge | community-driven packaging for conda](#)). The simplest conda instance we will use is miniforge. This is a collection of the core Python library along with the package manager, conda.

```
bash Miniforge3-$(uname)-$(uname -m).sh
```

We will spend the first class getting you all set with environments and additional tooling. If you have any problems with this installation, feel free to email at the address above.

git

In the class, we will master the basic commands of git. Git is a program that lets you save your work and manage different versions of your work. git was made for collaboration and it lets you contribute your changes to a broader project. Imagine you want to add your analysis to a project you are part of. We will go into this in more depth across the first four modules, starting with cloning the lecture notes repository in Module 1.

Before you download anything, see if you have git installed on your machine. To do so, go to your terminal and type:

```
which git
```

If it replies `git not found`, then you will need to install git. Here's a [link](#).

GitHub Desktop App

The GitHub Desktop App is optional but it can be useful to visualize your files. You can download it below: [Download GitHub Desktop | GitHub Desktop](#)

Claude Code

Students will be asked to use Claude Code during the last session of class. Students may register for a Claude Pro Account (\$20 a month) and then submit their receipts to Stacey Lara following the [MSK reimbursement policy](#). GSK will reimburse for up to 3 months of Claude usage (\$60 total). Note you must not enter PHI or sensitive data/code into Claude.

Course Materials

The course materials can be found at [gsk-intro-scientific-python-course](#). We will be using the gapminder dataset and RNA-seq data in class, and will spend some time downloading the course materials using git. I would like you to create a new folder in your home directory called workspace. We will retrieve all the lecture notes, homework assignments, and updates to the course through this git repo.

Homework

There will be a weekly assignment to further emphasize the course materials. It will be assigned on Wednesdays and to be completed by the following Monday.

Workgroups

Students will be arranged into working groups for projects and assignments as follows; please be advised that although you can work in groups, each student should submit his/her/their own individual assignments.

Group 1		Group 2		Group 3
Bui, Hien		Schlau, Steven		Yuan, Eric John
Parikh, Julie		Pryor, Nora		Tavakoli, Nassim
Hodson, Isabella		Patwari, Korey		Baniya, Subha
Group 4		Group 5		
Hodo, Yuki		Kroepfl, Gabrielle		
Uwayesu, Rosine		Artzi, Dorin		
Vegas, Isabella		Yao, Melissa		
		Zhang, Nora		

Course Schedule

Week 1

Module 1: Intro to Python

September 14, 2026

This class is focused on getting comfortable with using the Terminal. We will be learning some git basics and using interactive Python through the command line.

By the end of this class, you will be able to:

- Launch and navigate a command-line interface
- Clone a git repository and navigate the resulting directory
- Use the interactive Python shell
- Identify and use core Python data types (str, int, float, list, dict)
- Call built-in functions and import libraries

1. Class overview
 1. Introductions
 2. Review syllabus
2. Getting the course materials with git
 1. Cloning the lecture notes repository
 2. Navigating the cloned directory
 3. `git status` and understanding your remote (origin)
3. Setting up Python environment
 1. Checking Installation
 2. What is the command line
4. What is Python
 1. Python data types
 2. Built-in functions and libraries

Module 2: Jupyter Notebooks, Control Flow, and Functions

September 16, 2026

This class is focused on using Python in a notebook context and exploring its interface. This will be the primary way we interact with Python for the remainder of the class. We will also learn some new git commands as well.

By the end of this class, you will be able to:

- Create and run code in a Jupyter Notebook
- Save and track your own work with `git add`, `git commit`, and `git log`
- Use `.gitignore`

- Write conditional logic and loops
 - Define and call functions
 - Describe what an object/class is
1. Jupyter Notebooks
 2. Saving your work with git
 1. `git add` and `git commit`
 2. `git log`
 3. `.gitignore`
 3. Control Flow (If/Else, For loops)
 4. Functions
 5. Objects/Classes
 6. Bringing it all together
-

Week 2

Module 3: Intro to NumPy and SciPy

September 21, 2026

This class covers using arrays in NumPy and basic statistics with SciPy. We will also learn how to retrieve updates to a git repo and how to read file diffs.

By the end of this class, you will be able to:

- Pull updates from a remote repository with `git pull` and read a `git diff`
 - Create and manipulate NumPy arrays (`ndarray`)
 - Perform numeric operations and simple simulations with NumPy
 - Run and interpret a correlation and a t-test with SciPy
1. Staying in sync with git
 1. `git pull`
 2. Reading a `git diff`
 2. Intro to NumPy
 1. `NDarray`
 2. Numeric operations
 3. Random numbers and Simulations
 3. Intro to SciPy - Basic stats
 1. Correlation
 2. T-tests

Module 4: Intro to Matplotlib and Seaborn

September 23, 2026

This class covers building plots using Matplotlib and Seaborn. In Matplotlib, we will learn the anatomy of plots and how to change them directly. We will then use Seaborn for a high-level interface. We will cover saving and pushing your work with git.

By the end of this class, you will be able to:

- Push local commits to a remote with `git push`, and describe a minimal GitHub collaboration workflow
 - Build and customize a plot in Matplotlib
 - Build subplots in Matplotlib
 - Produce a basic statistical plot in Seaborn
1. Building a plot in Matplotlib
 2. Building subplots in Matplotlib
 3. Intro to Seaborn
 4. Collaborating with git
 1. Remotes and `git push`
 2. A minimal GitHub workflow
-

Week 3

Module 5: Intro to Pandas

September 28, 2026

This class introduces Pandas, a dataframe library used to load and manipulate data.

By the end of this class, you will be able to:

- Create and inspect a pandas DataFrame
 - Load external data into a DataFrame
 - Compute summary statistics on a DataFrame
 - Use group-apply-combine to summarize data by category
1. Dataframes
 2. Loading Data
 3. Summary Statistics
 4. Group-apply-combine

Module 6: Pandas, continued

September 30, 2026

This class covers more hands-on uses of Pandas.

Week 4

Module 7: Data Science in Practice

October 5, 2026

This class focuses on introducing statsmodels and scikit-learn for data analysis and data modeling.

By the end of this class, you will be able to:

- Fit and interpret a statistical model with statsmodels
- Train and evaluate a basic model with scikit-learn
 1. statsmodels
 2. scikit-learn

Module 8: Agentic Programming

October 7, 2026

This class introduces agentic programming as a companion in data science. We will cover design, planning, and iteration.

Week 5

Module 9: Agentic Programming cont.; Advanced Topic

October 19, 2026

Module 10: Final Project Presentation

October 21, 2026

