Module 1: Python essentials for health professionals

Data analysis with Python for health specialists

Yaé Ulrich Gaba

2026

AIRINA Labs

Why are we here?

You are health professionals. You already know:

- How to interpret lab values
- What a p-value means (roughly)
- That Excel has limits

Today you learn one new tool that makes your existing skills more powerful.

Clinical knowledge
+ Python

= reproducible research

Why Python, not Excel?

	Excel	Python
Reproducibility	✗ Click-based	✓ Script-based
Scale	Slow at 100K rows	Handles millions
Survival analysis	No	`lifelines`
Disease mapping	No	`geopandas`
ML prediction	No	`scikit-learn`
Cost	$150/yr	Free

Your workspace: Jupyter notebooks

Code cell: `print("Hello, health data!")`

Output: Hello, health data!

Markdown cell: Formatted text, headings, notes

☁ **Google Colab:** *colab.research.google.com* — no installation needed

💻 **Local:** Anaconda → Jupyter Notebook

Variables: storing patient data

<MINTED>

Naming convention
Use descriptive names: *patient_age*, not *x*. Your future self will thank you.

Collections: lists and dictionaries

List — ordered values:

<MINTED>

Dictionary — key-value pairs:

<MINTED>

Think of a dictionary as a patient chart in code.

Control flow: clinical classification

<MINTED>

Clinical mapping: AHA/ACC 2017 guidelines → Python code.

Loops: screening multiple patients

<MINTED>

Functions: reusable calculations

<MINTED>

Write once, use everywhere. Functions = reproducibility.

Importing libraries

<MINTED>

These five lines start **every** health data analysis notebook.

Install if missing
pip install pandas matplotlib seaborn scipy
In Colab: *!pip install lifelines*

What you can do after this module

1. Store patient data in Python variables, lists, and dictionaries
2. Classify clinical measurements with `if/elif/else`
3. Write reusable functions (BMI, eGFR, BP category)
4. Loop through patient records for batch screening
5. Import the core libraries for the rest of this course

Next: Module 2 — Health data with Pandas

Exercises (Hour 3)

1. **BP classifier:** Write *`classify_bp(systolic, diastolic)`* returning AHA category
2. **BMI batch:** Create 5 patient dicts with weight/height, compute and print BMI for each
3. **eGFR function:** Implement the CKD-EPI equation
4. **Mini-project:** Build a patient intake tool that prints a formatted clinical summary