Module 2: Health data with Pandas

Data analysis with Python for health specialists

Yaé Ulrich Gaba

2026

AIRINA Labs

What is a DataFrame?

A DataFrame is a table:

- Rows = observations (patients, countries, time points)
- Columns = variables (age, diagnosis, lab values)

Think of it as **Excel with superpowers**: scriptable, reproducible, and capable of handling millions of rows.

patient_id	age	dx
P001	45	HTN
P002	67	T2DM
P003	34	None

Creating a clinical DataFrame

```
import pandas as pd

data = {
    "patient_id": ["P001", "P002", "P003", "P004", "P005"],
    "age": [45, 67, 34, 52, 71],
    "sex": ["M", "F", "F", "M", "F"],
    "systolic_bp": [128, 158, 112, 145, 162],
    "hba1c": [5.4, 7.8, 5.1, 6.3, 8.2],
    "diagnosis": ["None", "T2DM", "None", "Pre-DM", "T2DM"]
}
df = pd.DataFrame(data)
df
```

Loading real health data

```python
# CSV from a URL (Gapminder life expectancy data)
url = ("https://raw.githubusercontent.com/datasets/"
       "gapminder/main/data/gapminder.csv")
gapminder = pd.read_csv(url)
print(f"Shape: {gapminder.shape}")  # (rows, columns)

# Excel file
df = pd.read_excel("patient_records.xlsx",
                   sheet_name="Lab Results")

# WHO Global Health Observatory API
url = ("https://apps.who.int/gho/athena/api/"
       "GHO/MDG_0000000026?format=csv")
mmr = pd.read_csv(url)
```

WHO GHO

Over 1 000 health indicators for 194 member states. Browse at

Exploring your data

The first thing you do with any health dataset — before any analysis — is **look at it**.

```
print(gapminder.shape)              # dimensions
print(gapminder.columns.tolist())   # column names
print(gapminder.dtypes)             # data types
gapminder.describe()                # summary statistics
gapminder.info()                    # non-null counts
```

These five commands give you a complete snapshot in 30 seconds.

Selecting and filtering

```
# Single column (Series)
ages = gapminder["lifeExp"]

# Multiple columns (DataFrame)
subset = gapminder[["country", "year", "lifeExp"]]

# Filter: diabetic patients (HbA1c >= 6.5)
diabetic = df[df["hba1c"] >= 6.5]

# Filter: African countries
africa = gapminder[gapminder["continent"] == "Africa"]

# Multiple conditions (use & and parentheses!)
high_risk = df[(df["sex"] == "F") &
               (df["age"] > 60) &
               (df["systolic_bp"] >= 140)]
```

Sorting and ranking

```
# Countries with highest life expectancy in 2007
recent = gapminder[gapminder["year"] == 2007]
top10 = recent.sort_values("lifeExp",
                           ascending=False).head(10)
print(top10[["country", "lifeExp"]])
```

`.sort_values()` + `.head(n)` is the quick way to find the top or bottom *n* observations.

Grouping and aggregation

Question: What is the average life expectancy per continent?

```
# Single aggregation
recent.groupby("continent")["lifeExp"].mean()

# Multiple aggregations
recent.groupby("continent")["lifeExp"].agg(
    ["mean", "median", "std", "count"]
)

# Patient counts by diagnosis
df.groupby("diagnosis")["patient_id"].count()
```

`.groupby()` + `.agg()` answers any "per group" question.

Creating new columns

```
# BMI from weight and height
df["bmi"] = df["weight_kg"] / (df["height_m"] ** 2)

# Age group (categorical from continuous)
df["age_group"] = pd.cut(
    df["age"],
    bins=[0, 18, 40, 60, 100],
    labels=["<18", "18-39", "40-59", "60+"]
)

# Binary flag
df["hypertensive"] = (df["systolic_bp"] >= 140).astype(int)
```

`pd.cut()` turns a continuous variable into clinical categories — essential for epidemiology.

Saving your results

```
# Save to CSV (most common)
df.to_csv("cleaned_patients.csv", index=False)

# Save to Excel
df.to_excel("cleaned_patients.xlsx",
            index=False, sheet_name="Patients")
```

Always save cleaned data for reproducibility. Keep the raw data untouched in a separate folder.

What you can do after this module

1. Load health data from CSV, Excel, or the WHO API into a DataFrame
2. Explore any dataset in 30 seconds (`.shape`, `.describe()`, `.info()`)
3. Filter patients by clinical criteria (BP thresholds, diagnosis, age)
4. Compute summary statistics per group (`.groupby()`)
5. Create clinical categories from continuous variables (`pd.cut()`)

Next: Module 3 — Data cleaning in health contexts

Exercises (Hour 3)

1. **Load and explore:** Load the Gapminder dataset, display shape, dtypes, and summary statistics
2. **Filter Africa:** Select African countries only; how many unique countries?
3. **Decade trends:** Compute mean life expectancy by decade for Africa vs. Europe
4. **Mini-project:** Build a complete WHO life expectancy analysis notebook with filtering, grouping, new columns, and a saved CSV