Day 01: Exploring Data with `pandas`

MSU REU Machine Learning Short Course

Learning Goals for Today

By the end of this activity, you'll be able to:

1. **Load real data** and spot problems before analysis
2. **Use documentation** as your primary learning tool (not just tutorials)
3. **Visualize relationships** between features and understand what separates classes
4. **Run the EDA loop** , which is the backbone of every real data analysis project

This is not "toy data"; you'll encounter these kinds of problems in research.

The Dataset — SDSS

Sloan Digital Sky Survey: 100,000 observations, 17 features, three classes.

Example of a distant galaxy (NGC 5681) corresponding to the GALAXY class

Class	What it is
STAR	A nearby star in our galaxy
GALAXY	A distant galaxy
QSO	A quasar — the bright core of a very distant galaxy

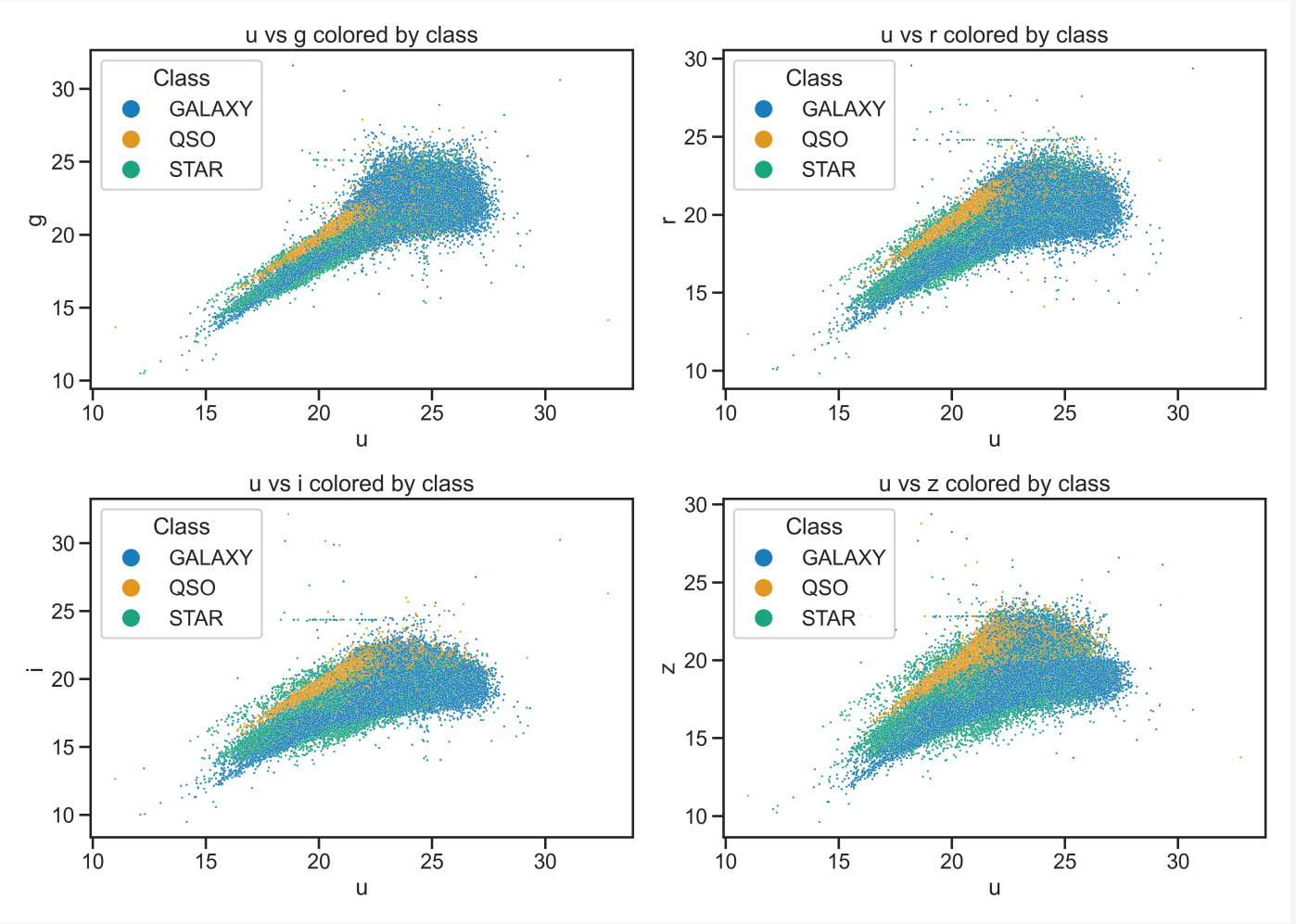
Why this dataset?

- **Real**: Not cleaned, not toy. Real telescopes collected this, with real problems.
- **Diverse features**: Positions, brightness measurements, spectroscopic data.
- **One week, one dataset**: We'll build classifiers, try regressions, all on the same problem.
- **You can trust it**: Well-documented, published, used in research.

> The dataset is at `activities/data/star_classification.csv`

What does the data look like?

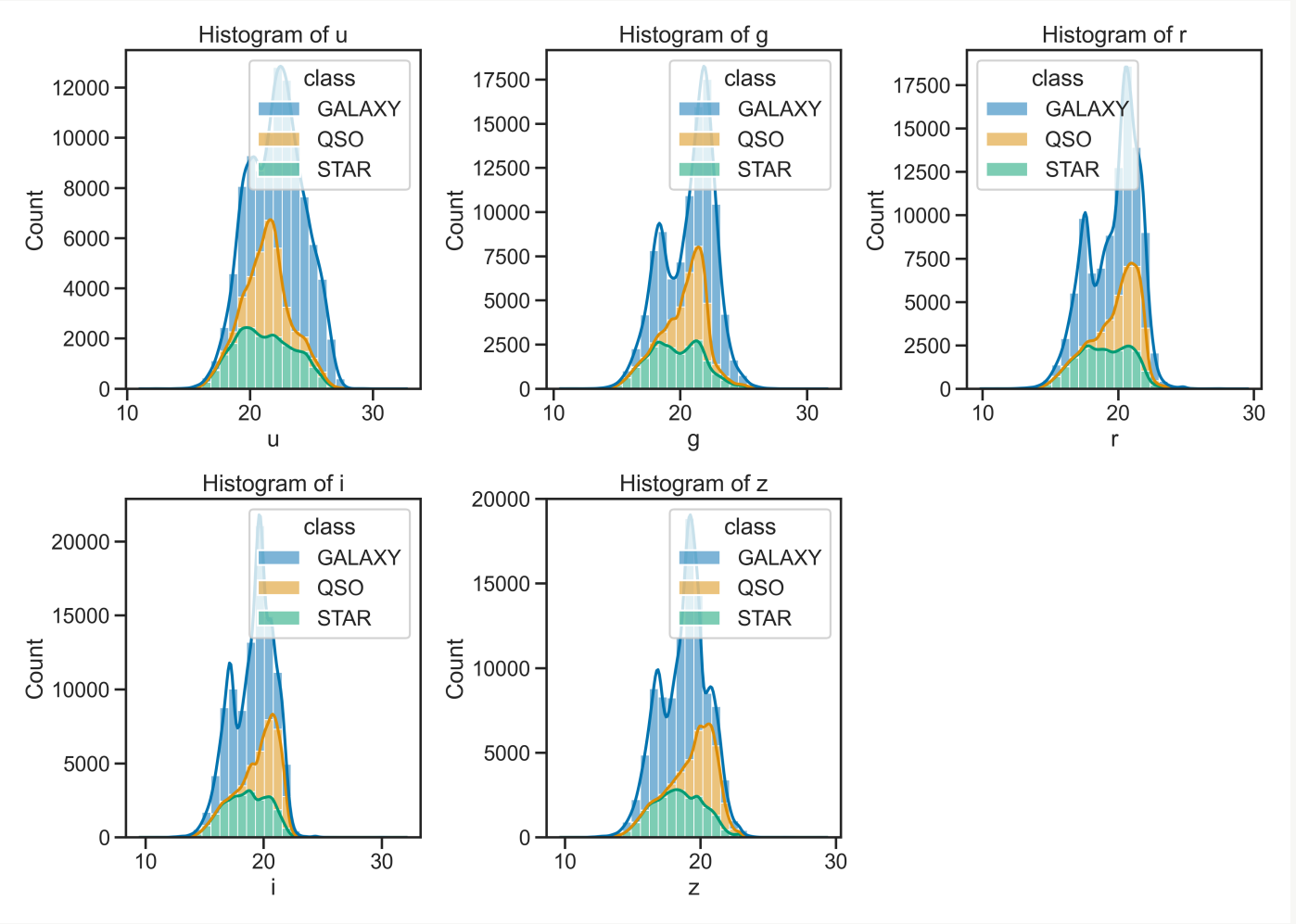
- Five photometric color bands: `u`, `g`, `r`, `i`, `z` — brightness at different wavelengths.
- **Think of it like filters:** green light vs red light vs infrared. Stars look different from galaxies in these measurements.



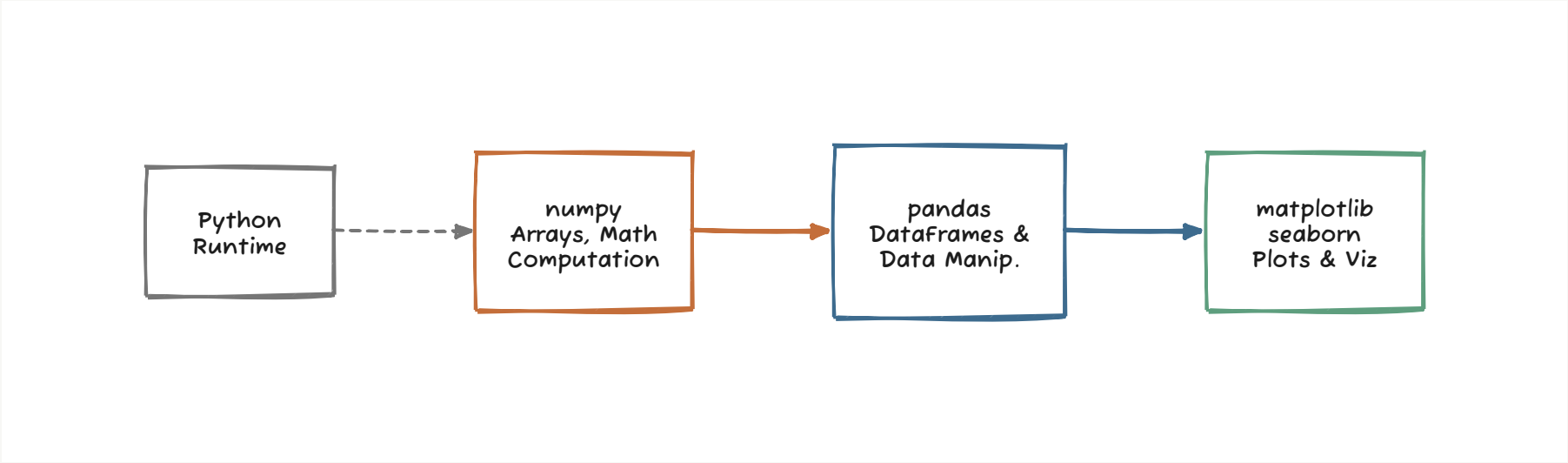
Distributions by class

Redshift `z` separates classes dramatically. Color bands? Less so.

What you'll discover: Some features are strong signals, others are noise. You'll explore this in the activity and ask: which features actually matter for classification?



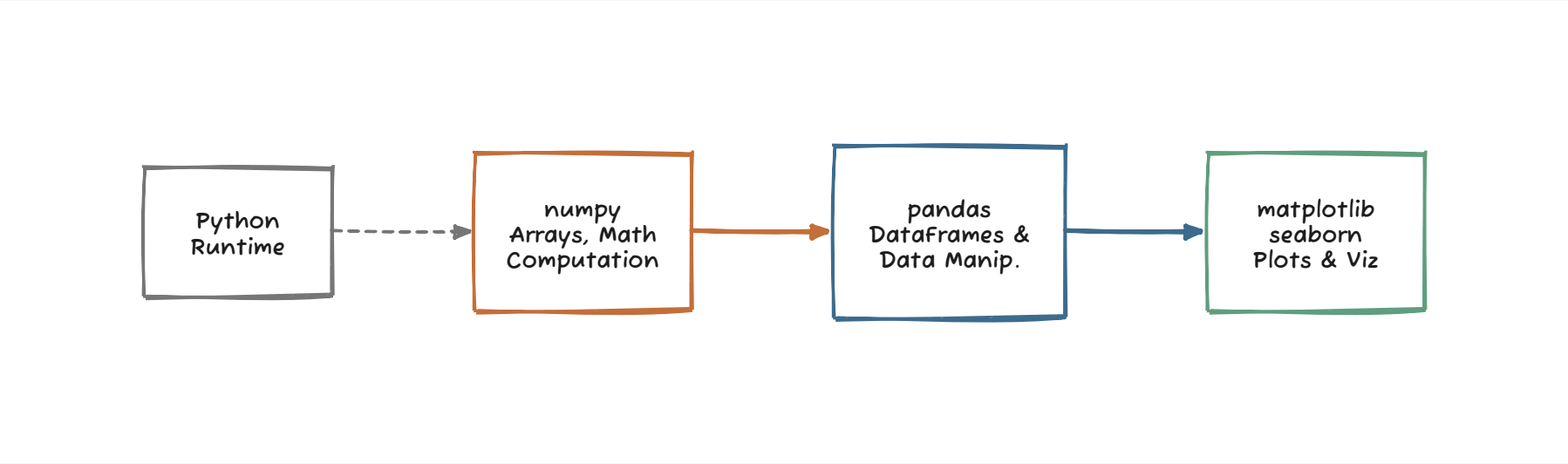
The Scientific Python Stack



We'll use three libraries throughout the course.

- **numpy** — fast array math, the foundation everything else builds off `numpy`
- **pandas** — labeled tables (like a spreadsheet, but scriptable). `pandas` is your data workbench.
- **matplotlib / seaborn** — plots. `seaborn` makes the beautiful plots; `matplotlib` is the engine.

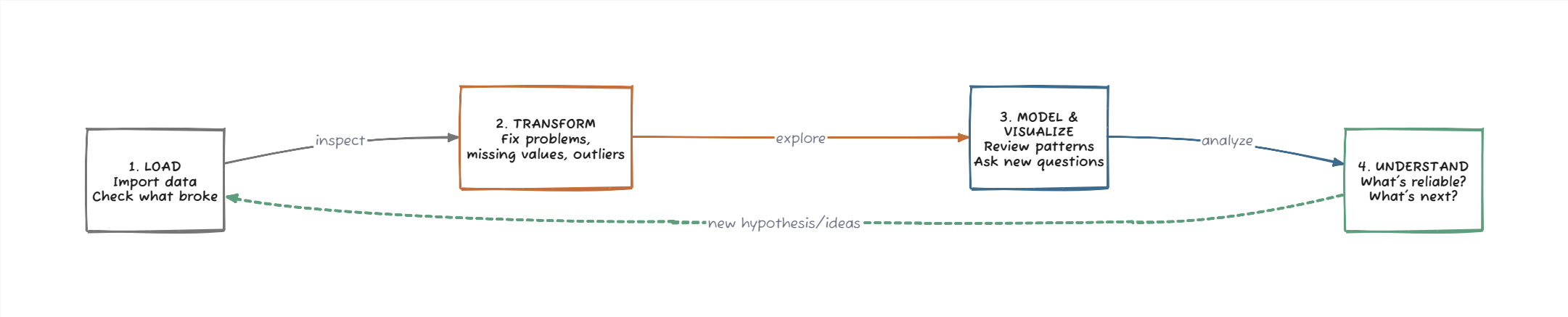
The Scientific Python Stack



Today's focus:

- `pandas` for loading, cleaning, and exploring.
- `seaborn` for understanding patterns.

The EDA Loop



1. **Load**: Import data and check what broke.
2. **Transform**: Fix problems like missing values, invalid numbers, outliers.
3. **Model & Visualize**: Plot patterns, run analyses. Ask new questions.
4. **Understand**: What's reliable? What's the next hypothesis?

You'll run this loop many times and each pass teaches you something new.

Activity 01: Exploring Data with `pandas`

What you'll do:

1. **Load and inspect** the SDSS dataset (100,000 stars, galaxies, quasars)
2. **Clean it** — find and remove invalid photometric values (real data is messy)
3. **Visualize** with scatter plots and histograms, colored by class
4. **Discover patterns** by answering: which features separate classes best?

> Work in pairs/groups on these research questions:
>
> - Correlation between color bands
> - How redshift differs by object type
> - Which measurements are actually useful for classification
>
> **This prepares you for the classifiers we'll build later.**

Key `pandas` commands

These five operations will solve 80% of your data work. You'll use these repeatedly in the activity.

```
import pandas as pd

df = pd.read_csv("data/star_classification.csv")  # load the data
df.head()          # peek at the first 5 rows
df.info()          # what columns? any missing data? what types?
df.describe()      # mean, median, min, max — spot outliers here

# filter: remove bad data (e.g., negative brightness is impossible)
df = df[df["u"] > 0]

# select: work with only the features you care about
features = df[["u", "g", "r", "i", "z"]]
```