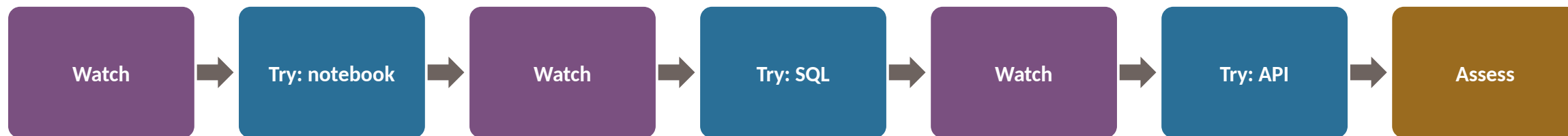


MODULE 1

Acquisition and the reproducible workflow

Short recorded segments with a hands-on exercise between each, so nothing runs long and you stay active.

Watch a little, then do a little



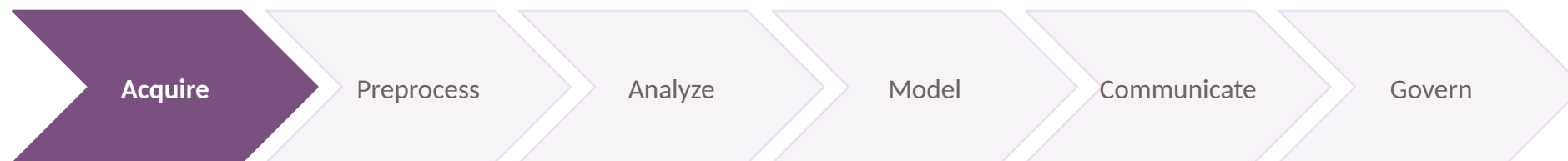
Each recorded segment is short. You pause, practice the concept hands-on, then return for the next piece.

PART 1 OF 3

The data-science lifecycle, and provenance

About 7 minutes on camera

Where this module sits in the lifecycle



This module builds the acquisition layer: getting data in from flat files, relational databases, REST APIs, and the live web. Everything downstream inherits its quality and its traceability.

Provenance is a first-class artifact

- Record lineage: where each field came from, by what method, and when.
- Capture retrieval timestamps and the exact parameters of every pull.
- Trace every field, do not summarize it as “pulled from the API.”

THE DATASHEET IDEA

field → source

field → endpoint / query

field → parameters

field → retrieval time

field → transformations

NOW TRY IT

Reproducible notebook on Binder

- Open a small public repo on Binder and run it top to bottom.
- Confirm you get identical output, this is the environment model your own work must match.

About 10 minutes, then come back for Part 2.

PART 2 OF 3

The reproducibility stack

About 7 minutes on camera

What reproducibility actually requires

Version-controlled code. every change recoverable and attributable

Pinned dependencies. the same environment runs the same way later

Seeded randomness. a run is repeatable, not lucky

Code committed, data and secrets never. keep keys and data out of the repo

NOW TRY IT

Select Star SQL

- Work the query chapters through joins and aggregation.
- Keep the exact text of two queries you write, you will reuse that pattern to pull from the database source later.

About 15 minutes, then come back for Part 3.

PART 3 OF 3

Acquisition modes and the ethics of scraping

About 8 minutes on camera

Four ways data comes in, four hidden failures

Flat files

Hidden failure: silently truncated or wrong encoding

Databases

Hidden failure: a wrong join inflates or drops rows

REST APIs

Hidden failure: un-paged or rate-limited, so truncated

Web scraping

Hidden failure: the page changes under you

The moving target

- A paginated, rate-limited API whose schema drifts will break a naive pull.
- Handle pagination, back off on a 429, and retry idempotently so you neither lose nor duplicate records.
- Fingerprint the response schema on every page, so a mid-collection change is caught the moment it appears.

NOW TRY IT

Hoppscotch: paging a real API

- Send a GET to JSONPlaceholder /posts, then /posts?userId=1, and read the status and headers.
- Page through /comments with _page and _limit, and record how many records you retrieve versus how many exist.

About 10 minutes, then finish Part 3.

The ethics of scraping

- Read and honor robots.txt; respect rate limits and crawl-delay.
- Terms of Service carry weight: technically reachable is not the same as permitted.
- Prefer a public API over scraping the rendered page when one exists.

NOW TRY IT

Responsible scraping sandbox

- At quotes.toscrape.com, read robots.txt first.
- Extract quotes and authors from the first two pages only, with a polite delay and a descriptive user-agent.

About 10 minutes.

Formative deliverable (completion points)

- Build a short reproducible notebook that pulls the same small dataset three ways: a flat file, a SQL query, and an API call.
- For each pull, record source, retrieval time, and parameters in a provenance cell.
- Checked for reproducibility and whether the three pulls reconcile. Feedback before the graded work opens.

How it counts. Submit the notebook with a short reflection or evidence you ran it. Credited on completion, not scored for correctness. This is the practice that makes the graded simulation go well.

GRADED ASSESSMENT

The Provenance Incident

- Reconcile four sources into one weekly table for a board deck, and prove where every number came from.
- A schema change fires mid-collection with no warning, so you must detect it and keep the numbers trustworthy.
- It is a separate item in this module, seeded per student, so the answer has to be your own reasoning.