

ICT for Civic Data — Turin University 2025–26



# From Data to Map

Crash Course — Day 2 Afternoon

**Define Your Angle**

## > Write down your angle

---

Based on this morning's discussion, define your proposal angle:

1. What **territory**? (be specific: a country, a region, a coastline)
2. What **issue**? (flood risk, drought, wildfire, cyclone exposure)
3. What **population or infrastructure** is affected?
4. In one sentence, what would your proposal **argue**?

Write this down. You will use it for the rest of the week.

# Data Formats and Sources

## > Where data lives

---

Location	What you find	How to get it
Open data portal	Downloadable files (CSV, Excel, GeoJSON)	Browse and download
API	Structured data you can query	Send a request, get a response
Web page	Tables, lists, text	Scraping (manual or AI-assisted)
PDF or report	Tables, figures, text	Extraction (AI, Tabula, OCR)

Your AI tool can help with all of these. But you need to know **what you are looking at**.

## > Common data formats

---

Format	What it is	Good for
<b>CSV</b>	Text file with commas separating values	Tabular data. Opens in any spreadsheet.
<b>GeoJSON</b>	JSON with coordinates	Map data. Goes straight onto Leaflet.
<b>Shapefile</b>	Legacy GIS format (3–6 files together)	Geographic boundaries. Needs conversion for web maps.
<b>JSON</b>	Nested, structured data	API responses. Needs flattening for tables.
<b>Excel (.xlsx)</b>	Spreadsheet with sheets and formulas	Common in government data. May need cleaning.

If you are unsure about a format, ask your AI tool: "I downloaded a .shp file. What is it and how do I use it on a Leaflet map?"

## > What is an API?

---

An **API** (Application Programming Interface) is a way to ask a database for specific data.

Instead of downloading a huge file and filtering it yourself, you send a **request** and get back only what you asked for.

**Example:** the REST Countries API

```
https://restcountries.com/v3.1/name/kenya
```

This returns data about Kenya: population, area, region, capital, coordinates.

APIs are useful because your AI tool can **write the request for you** and **process the response**.

# Finding Data for Your Angle

## > Two habits before you search

---

1. **Be precise.** The more specific your question to Gemini, the better the answer.

### Vague

"Find me some data about disasters."

### Precise

"I need geospatial data on flood events in the Mekong Delta region of Vietnam, with coordinates, dates, and severity. What open sources exist?"

2. **Tell Gemini you are a beginner.** It will adapt its language and explain technical choices instead of assuming you know.

## > Look at the data before going further

---

When you find a source: **download a sample and look at it** before asking Gemini to do anything with it.

- > How many rows? How many columns?
- > What do the column names mean?
- > Are there coordinates? In what format?
- > Is it the right geographic area? The right time period?

If you skip this, you will ask Gemini imprecise questions and get surprising results.

## > Working efficiently with AI tools

---

- > Use **web chat** (Gemini web, ChatGPT) for discovery and simple questions. Free, no tokens.
- > Use **CLI** (Gemini CLI, Claude Code) for working with files, building maps, running scripts. Costs tokens.
- > **Split complex tasks** into smaller steps. One prompt per step.
- > **Resume sessions:** use `gemini --resume` to reopen your latest session. Use separate tabs for separate tasks.

The CLI is powerful but expensive. Do your thinking in web chat, do your building in the CLI.

## > Git branches and GitHub Pages

---

When you ask the agent to publish to GitHub Pages, it may try to create a **gh-pages branch** instead of using **main**.

**What is a branch?** A parallel version of your repository. The agent creates one because many tutorials use a separate branch for published content.

### **Your options:**

- > Tell the agent: "publish on the main branch, not gh-pages"
- > Or: go to Settings → Pages and change the source to the gh-pages branch

Either works. Just be consistent.

## > Common issues with agentic tools

---

**The agent thinks too long.** If nothing happens for 5–10 minutes, press **Escape** to stop it. You may need to press Escape multiple times. Then rephrase your request more simply.

**The agent over-engineers.** Ask for a "simple Leaflet map that can be published on GitHub Pages." Without "simple," the agent will build something complex and waste your tokens.

**Browser cache.** After pushing changes, your GitHub Pages URL may show the old version. The agent may ask you to **hard refresh** (Ctrl+Shift+R on Windows, Cmd+Shift+R on Mac). This is safe: it just clears the cached version.

**The agent fails at simple tasks.** Sometimes the agent cannot download a file that is one click away. If that happens, download it yourself and drag-and-drop it into your Codespace file explorer. Not everything needs to go through the agent.

## > The process: Find, then Get, then Verify

---

Do them **separately**:

1. **Find**: ask Gemini what sources exist. Save the list to `sources.md` in your repo.
2. **Get**: pick one source. Ask Gemini to write a script that downloads it. Read the script. Run it.
3. **Verify**: add the data to your map. Does it look right?

If step 3 reveals a problem, go back to step 1 or 2.

## > Exercise: find, get, and map data for your angle

---

Individual work — rest of the afternoon and tomorrow morning

1. **Define** your angle (from earlier)
2. **Find** data sources relevant to your argument. Document them in `sources.md`.
3. **Get** one dataset. Use a script.
4. **Verify** by adding it as a layer on your map.
5. **Push** and check the live page.

If you finish one dataset, find and add another. This exercise continues tomorrow morning.

**Wrap-Up**

## > Where are you?

---

Quick round — share where you are in the process:

- > Have you **defined** your angle?
- > Have you **found** relevant data sources?
- > Have you started **getting** data?
- > Have you managed to **add a layer** to your map?

No pressure to be finished. We continue this tomorrow morning.

## > What we covered today

---

- > Practiced Find → Get → Verify with a shared dataset
- > Understood why precise prompts matter (beginner vs. expert assistant)
- > Connected concrete examples to broader proposal arguments
- > Defined individual angles
- > Started finding and mapping data for our own proposals

**Tomorrow morning:** continue the individual work, then we move to verification and cleaning.

**Questions?**