ICT for Civic Data — Turin University 2025-26

Enrichment and Cleaning

Crash Course — Day 3 Afternoon

civicliteraci.es

Cleaning for Visualisation

> The goal

By the end of this afternoon:

- > Clean the FloodArchive data in Google Sheets
- > Build a **visualisation page** with charts that slice the data in different ways
- > Publish it on GitHub Pages as a **separate page** from your map

Tomorrow we will connect everything with a landing page.

> Why cleaning matters

You want to build charts like:

- Flood events **by country** (top 20)
- Flood events **per year** (trend over time)
- Breakdown **by cause**
- Total displaced **by severity**

But with the current data:

- "Philippines" is spelled 5 different ways → 5 bars instead of 1
- "Heavy rain" has 9 variants → 9 slices instead of 1
- Dead = 0 could mean zero or unknown → totals are unreliable

Clean first. Visualise second.

> The FloodArchive: what's actually in it?

We have been using this data all week. Let's look at it properly.

Open the FloodArchive CSV in **Google Sheets**:

1. Go to Google Sheets → File → Import → Upload
2. Upload `FloodArchive_clustered.csv`
3. Choose "Comma" as separator

Now we can see every row and every column. This is what the AI has been working with.

> What does messy data look like?

Let's explore. In Google Sheets, try these:

- **Sort by Country** (Data → Sort sheet → Column C). Scroll through. Notice anything?
- **Sort by MainCause**. How many ways is "Heavy rain" spelled?
- **Filter Dead = 0**. Does 0 mean zero people died, or unknown?

Take 5 minutes to explore. Write down what you find.

> Country name problems

In the data	Should be
Phillipines, Philipines, Philippine	Philippines
Philippines (with trailing space)	Philippines
DR Congo, Democratic Republic Congo, Democratic Republic of the Congo	Pick one, use it everywhere
USA, USA., USA (trailing space)	USA
UK, UK (trailing space)	UK

A computer treats "Philippines" and "Philippines " as two different countries. So does a filter. So does a chart.

> MainCause problems

In the data	Count	Problem
Heavy rain	2,131	
Heavy Rain	893	Different capitalisation
heavy rain	4	Different capitalisation
Heavy Rain	2	Double space
HeavyRain	1	Missing space
Heay Rain	1	Typo
Heavy RAin	2	Typo
0	11	Missing value coded as zero
(blank)	3	Actually missing

Nine representations of the same cause. A bar chart will show nine categories instead of one.

> Validation field problems

The Validation column records the source that confirmed the flood event.

In the data	Count
FloodList	502
FloodLIst	38
Flood List	8
Flood LIst	1

Four versions of the same source. And 188 entries have "0" as validation, meaning: unvalidated.

> The ambiguity of zero

In this dataset, **0 does not always mean zero**:

- `Dead = 0` for 1,554 events (30%). Did nobody die, or is the number unknown?
- `Displaced = 0` for 1,525 events (30%). Same question.
- `GlideNumber = 0` for 4,652 events (91%). This is a missing ID, not a zero.

You have to decide what 0 means in each column and document your decision. The data will not tell you.

Cleaning in Google Sheets

> Why Google Sheets for cleaning?

- You can **see** every value and every change
- Your colleague can see it too (shared spreadsheet)
- Formulas are visible in the cell (traceability)
- No code to review or debug

For small-to-medium datasets (under ~50,000 rows), a spreadsheet is often the most transparent cleaning tool.

> Getting data into Google Sheets

Two ways to import data:

Upload: File → Import → Upload your CSV. Choose the right separator (comma, semicolon, or tab).

IMPORTDATA formula: pull data directly from a URL:

```
=IMPORTDATA("https://raw.githubusercontent.com/user/repo/main/data.csv")
```

After importing with a formula, **detach the data** so you can edit it: select all → Copy → Edit → Paste special → **Values only**. Now the cells contain values, not formulas.

> Useful Google Sheets functions

Function	What it does	Example
`LEFT(A2, 4)`	Extract characters from the left	Extract year from "1985-01-01"
`TRIM(A2)`	Remove leading/trailing spaces	Fix "Philippines "
`UPPER(A2) / LOWER(A2)`	Change case	Standardise text
`SUBSTITUTE(A2, "old", "new")`	Replace text within a cell	Fix specific values
`COUNTIF(A:A, "value")`	Count matching cells	How many "Philippines"?

You do not need to memorise these. Ask your AI chatbot: "what Google Sheets formula would do X?"

> Pivot tables for exploration

A pivot table **counts and summarises** data without changing it.

1. Select all your data
2. Insert → Pivot table → New sheet
3. Set **Rows** = what you want to group by (e.g., Country)
4. Set **Values** = what you want to count (e.g., COUNTA of ID)

Try: flood events **by year**. Use LEFT to extract the year from the Began column first, then pivot on that.

Add **conditional formatting** (Format → Conditional formatting → Colour scale) to spot patterns visually: where are the red spots?

> Technique 1: Filter and overwrite

Use column filters to isolate incorrect values, then fix them in bulk.

1. Select the column → Data → Create a filter
2. Click the filter dropdown → uncheck all → select only the wrong value (e.g., "Phillipines")
3. Now only the wrong rows are visible. Type the correct value in the first visible cell, then **copy it down** to all filtered rows.
4. Remove the filter. The corrected values are now in place alongside the rest.

This is more controlled than Find and Replace: you see exactly which rows you are changing.

> Technique 2: handling invisible spaces

Trailing spaces are invisible but break everything.

Google Sheets may **suggest automated cleaning** (a small prompt appears when it detects trailing spaces). This catches spaces **after** the text but can miss spaces **before** it.

To catch both, use the `TRIM` formula in a helper column:

```
=TRIM(C2)
```

TRIM removes leading, trailing, and double spaces. Apply it to the column, then copy-paste **values only** to replace the originals.

Always verify after cleaning: filter by the original and trimmed values to confirm they now match.

> For larger datasets: OpenRefine

Google Sheets works for thousands of rows. For tens of thousands or more, **OpenRefine** is the tool.

- **Text facets**: see all unique values in a column instantly, click to filter
- **Clustering**: automatically group similar values and merge them
- **Timeline facets**: filter dates with a visual histogram
- **Trim whitespace**: one click to remove leading/trailing spaces

You don't need to memorise OpenRefine. Ask your AI chatbot: "remind me how to do clustering in OpenRefine" and it will walk you through it.

> OpenRefine clustering algorithms

Different algorithms catch different problems. Try several.

- **Key collision**: catches exact duplicates (Brazil/Brasil)
- **Nearest neighbour**: catches near-matches (Phillipines/Philippines)
- **Phonetic** (metaphone): catches sound-alikes (Cote d'Ivoire variations)

OpenRefine can also **fetch URLs** to call APIs (e.g., REST Countries to get capital cities), parse JSON responses, and extract substrings using GREL formulas.

> Exercise: clean the FloodArchive data

Individual work — 45 minutes

In your Google Sheets copy of FloodArchive:

1. **Standardise Country names**: fix the variants we identified (Philippines, DR Congo, USA, UK, etc.)
2. **Standardise MainCause**: consolidate the "Heavy rain" variants. Decide what to do with "0" and blank values.
3. **Handle ambiguous zeros**: in Dead and Displaced columns, decide what 0 means. Add a note or a new column documenting your decision.
4. **Fix Validation**: consolidate FloodList variants

You do not need to clean every row. Focus on the problems we identified and document what you did.

Proposals:
What to Pitch

> Don't pitch doing what they already do, but better

The worst pitch: "we will help you do what you already do, but with data."

The funder already has processes. They don't need someone to tell them to keep doing the same thing with a spreadsheet on top.

A strong pitch shows **new possibilities**: things they thought were impossible, or things they didn't know were possible.

> Find the most compelling point

Your proposal will have many ideas. Not all of them are equally compelling.

- **Keep data presentation simple.** The funder is not a data expert.
- **Embed evaluation** in the methodology, but don't make evaluation the pitch. Nobody gets excited about better evaluation.
- **The case study is your proof of capability.** "I'm showing you that I can do the work."
- It's not about these specific numbers. It's an example of how you would use data to support decisions.

> Show new capabilities

Many organisations already collect data but don't know what to do with it.

Your proposal should show: **with the data you already gather, here is what becomes possible.**

- A dashboard that turns routine data into situational awareness
- A map that connects data points no one had connected before
- An early warning system built from data they already have

How data gives the funder **opportunities to do things differently**.

Tips and Q&A

> CSV is not always comma-separated

Although CSV stands for "comma-separated values," you may encounter:

- **Semicolon-separated** files (common in Europe, where commas are decimal separators)
- **Tab-separated** files (.tsv)

This is why Google Sheets asks you to choose the separator when importing. If your data looks wrong after import, try a different separator.

> Your Codespace is a rented computer

When your Codespace shuts off (it does this automatically to save resources):

- Your **files are preserved** (data, scripts, maps)
- Your **apps restart fresh** (Gemini CLI needs to be relaunched)

Think of it as: the data stays, the tools restart.

> Controlling the AI's pace

If a task is complex, tell the AI to go **step by step** and explain each step before executing.

- Prevents the AI from interpreting your request in a way you don't want
- Makes it easier to go back if something is wrong
- Gives you context that helps you learn

Give the AI **context about what you're doing and why**. This forces it to think more carefully about its approach.

> Community data and empowerment

Several of you are working on proposals involving community-based data practices.

A key framing: the goal is not to teach the community new things. It is to help them **demonstrate that what they already know and do is powerful**.

- > Surface existing knowledge and practices
- > Use data to highlight impact that is already happening
- > Empower the community to advocate with evidence from their own experience

Wrap-Up

> What we produced today

Morning:

- Walked through case study → angle → proposal for health facilities in flood zones
- Built an enriched map with a filter for at-risk facilities
- Learned the five-element prompt structure

Afternoon:

- Explored the FloodArchive data and found real problems
- Cleaned the data in Google Sheets and OpenRefine
- Learned pivot tables and conditional formatting for data exploration

Tomorrow morning: we turn the cleaned data into a visualisation page with charts, published on GitHub Pages alongside your map.

Questions?