

Project: Course enrollment dashboard

(For Claude Opus 4.7 Adaptive)

Tools for assessing and visualizing historical class enrollment data

Note that I have successfully run `download_schedules.py` and `parse_schedules.py` for all academic years from 1990-1991 through 2025-2026.

I am interested in making an interactive display or dashboard that uses all the CSVs we've created. This dashboard would be accessible online and would allow users to generate and see graphs of course enrollment over time for courses of interest. Before I specify features I'd like this dashboard to have, I'd like to know if this is a straightforward or feasible goal. Please comment.

Response

Yes — this is well within reach, and there are several well-trodden paths for a project of this scope. A few observations to frame your feature thinking:

Data size is not a concern. Roughly 144 terms × hundreds of courses × multiple sections gives a combined CSV that's likely in the tens of MB and a few hundred thousand to a couple million rows. That's small enough to hold entirely in memory on a free-tier host, and small enough that a browser can even handle it client-side if you want a fully static deployment.

Hosting and stack — multiple easy options:

- **Streamlit Community Cloud** (free) is probably the lowest-friction path: write a Python script with pandas + Plotly/Altair, push to GitHub, point Streamlit at it. Deployment is essentially one click. Good fit since you already have everything in Python and CSV.
- **Hugging Face Spaces** (free) supports Streamlit or Gradio with similar ease.
- **Plotly Dash** on Render or Fly.io if you want more layout control.
- **A fully static site** using DuckDB-WASM or a pre-built Parquet file, hosted on GitHub Pages or Netlify. No server, infinite uptime, zero ongoing cost — slightly more setup, but very robust.
- **Observable Framework** is excellent if you eventually want polished, interactive notebook-style dashboards.

For a dashboard centered on "show me enrollment over time for course X," any of the first three would be straightforward — probably a single afternoon's work to get a usable v1 online.

The real wrinkles aren't technical, they're about course identity. Before we build, you'll want to decide how to handle:

- *Course number changes.* "PHYS 101" today might have been a different number in 1995. A simple match on (Subject, Course_Number) will give a jagged time series whenever numbers shift.
- *Multiple sections per term.* Do you sum Enrollment across all CRNs for a (term, course) pair, or let the user see sections broken out? (Probably sum by default, with a toggle.)
- *Format types.* Lectures vs. labs vs. discussions — usually you want to aggregate or filter to just lectures so you're not double-counting students.
- *Subject code changes.* Departments occasionally rename or split (e.g., a department gets a new code).

Worth checking whether any of your 163 subjects appeared/disappeared mid-record.

- *Cross-listed courses*. Same students, two listings.

None of these are blockers — they're design decisions. You can build a perfectly useful v1 that just matches (Subject, Course_Number) literally and notes the caveats in a sidebar, then refine the matching logic later.

Go ahead and lay out the features you want. Once I see those I can recommend a specific stack and sketch the structure.

About the course catalog

[deleted; scraping the course catalogs]

Resume...

Here are my answers to the issues you raised: Course number changes: It's fine if courses change and therefore the time series are jagged. The user should be aware of this, and resolving it is beyond the scope of this task. Multiple sections per term: Let the user see sections broken out; sum by default, with a toggle. Format types: This should be selectable, with the default of filtering to just lectures. Other options should be to filter to the first item of a given course number (which will likely be the lecture), or to sum over all. Subject code changes. Like Course number changes, we will ignore this. Cross-listed courses: We will ignore this.

Here are features the dashboard should have:

- The user selects a subject from a dropdown menu or by typing a subject code (for example, "PHYS").
- The user selects a course number from a dropdown menu or by typing a number (for example, "251").
- The user can click a button to add the selected course to the plot. (Perhaps an "add" button, or perhaps arrows that move a course from the selection window to the selected list.) There will be a list of selected courses; these can also be removed. There will also be a way to remove (or clear) all courses from the display.
- The user selects a starting academic year and an ending year to plot, with the defaults being 1990 (the first year for which we have data) and 2025 (the present academic year, 2025-26). This range applies to all selected courses.
- There are options for format, as noted above.
- There is an option to export the plot as a .png file, and to export the selected data as a CSV file.
- The x-axis for the plot will be year. Consider the academic year to begin on October 1, or roughly $0.75 +$ calendar year. Let's plot the year for Fall term as 0.87 plus the academic year number, so Fall 2025 is plotted as 2025.87 . Consider winter as starting January 1, so Winter as offset by 0.125 , so Winter 2026 is 2026.125 . For Spring, 0.375 , and summer 0.625 . - Draw a dashed vertical line at the boundary of a new academic year, i.e. at calendar year + 0.75 .
- There should be an option to exclude summer, default True. (In general, we don't want to consider the summer term.)
- I have all the course catalog information in a CSV; see the "Catalog scraper development" chat. If there are 5 or fewer courses plotted, show the catalog description for these courses in a box below the plot.
- There should be a "normalize" Option (default false) to normalize the enrollment to the first (nonzero) term of the range. If this is true, we'll be able to easily see and compare relative changes in enrollment.
- Make a README file that can also be displayed via the dashboard. It should note our decisions on issues like Cross-listed courses.

Later I may want options for coloring courses by subject. If you have suggestions, please provide them.

Let us make this for Streamlit Community Cloud. I have a Github account already. Please provide instructions and

/ or code.

[enter the above]

Claude responds with instructions etc. I save this separately: Response regarding implementation.docx

Structure:

uo-enrollment-dashboard/ `app.py` Streamlit app (~530 lines, all in one file) `prepare_data.py` One-off step: `combined.csv` → `data/enrollment.parquet` `requirements.txt` `README.md` Also displayed in the app's "About" tab `.gitignore` `data/` `README.md` placeholder; the real files arrive here from `prepare_data.py`

Move code to C:\Users\raghu\Documents\Misc Academia\UO Statistics\UO Course Schedules and Analysis\Enrollment Dashboard\Code

Local test... Works!

Revision:

I have tested locally! This is great! I have a few changes I'd like to make: 1 To make the graphs more readable, it would be good to use different symbols. Cycle through various symbols, such as open and closed circles, squares, etc., for courses. 2 I'd like an option to sum the enrollment over an academic year, i.e. summing the enrollment for a course for Fall, Winter, and Spring, and Summer if Summer is not excluded. Default false. If this option is selected, for each course the graph should plot a horizontal line from the start (Calendar year + 0.75) to the end (AY + 1.5 if summer is excluded, AY + 1.75 if summer is included) for each academic year. The summing should also be indicated in the graph title. In other words, each course gets one "step like" graph, which each step spanning the year.

Another revision ("Z")

Great! A few more changes: 1 For the "step like" graph for annual sums, there should be a line connecting the right edge of each step to the left edge of the subsequent step, to make it easier for the reader to follow the curve of a course. In other words, I don't want disconnected horizontal lines. 2 Some courses have had a "Z" added to the course number, for example "CH 221" is now "CH 221Z". Would it be feasible to combine the enrollment data for courses that differ only by a "Z" at the end?

These are implemented. Impressively, Claude correctly notes the reason for the "Z" -- "UO has been appending the Z suffix to many introductory courses (CH 221 → CH 221Z, BI 221 → BI 221Z, etc.) to mark statewide-transferable equivalents"

Publish

I make a Streamlit account; allow Streamlit to access my Github account

URL: <https://uohistoricalcourseenrollmentdashboard052026rp.streamlit.app/>