30538 Final Project: Accessible, Visual, Reproducible Policy-Oriented Data Analysis

Peter Ganong and Maggie Shi

The goal of this project is to showcase your knowledge of Python by applying it to a data question about a policy topic you are interested in.

You may work on this project alone, or in groups of up to three students. All groups must be formed declared in the proposal before any work is done - it is not possible to join one after.

Key Dates

- January 24: Proposal submitted via ungraded Canvas quiz
- March 2 - March 5: upload slides at start of day and give in-class presentation
- March 9 5PM: final writeup and repository submitted via public Github repo
  - 10% penalty for submitting up to 24 hours late
  - No credit for submitting more than 24 hours late

Proposal

The proposal should include:

- List of group members (1 to 3)
  - We recommend that you do so with other students in your section
    - You are allowed to have a group member from another section, but everyone in the group must be able to make it to **every** lecture and lab section of every group member.
- Motivating question(s)
  - This does not need to be a causal question, but it should be related to an area of public policy

- 3 possible datasets
  - These datasets should be either publicly available or there should be a clear path to acquiring the data within the quarter
  - In order to showcase the skills you have learned in this course, your data should be as disaggregated as possible. Ask yourself "could I easily do this data project in Excel?" and if the answer is "yes" then you should find a different dataset. For example, we strongly discourage the use of state- or country-level data.
- 4 ideas (2-3 sentences each) on static visualizations

Grading

Presentation (30%)

- Presentations are 10 minutes.
- One group members will be *randomly selected* to present in real-time.
- All present group members receive the same presenation grade. Absent group members get a grade of 0.
- The presentation will be of slides that largely mirror the structure of the writeup, but will be more focused on discussing the research question and results.
  - At the time you submit your presentation slides your project needs to have all the components required for the final version of the project (2 datasets, 2 static plots, 1 Streamlit app)
  - However, each of these components can still be "work-in-progress" and can (and should!) still be refined after the presentation, based on instructor and peer feedback

Coding (45%)

The code for the project should have the following components:

1. Dataset choice and wrangling (20% of total)
   - You must use a minimum of *two* datasets.
   - All processing of the data should be handled by your `.qmd` code, including all merging and reshaping.
2. Visualization (15% of total)
   - From that data, you will create a minimum of *two* static plots using `altair` or `geopandas`
   - As well as one `streamlit` app with one dynamic plot/component

    - You can also add additional dynamic plots into your app to substitute for a static plot. So, a `streamlit` app with 3 dynamic plots will count for full credit.
  - At least one visualization should be a spatial visualization.

3. Reproductibility (5% of total)

   - A user should such as a TA or an AI agent should be able to knit your `.qmd` file and re-generate one version of your writeup (HTML or PDF).
     - If the underlying dataset is large, you can ask the TA or AI agent to first download the data via a link included in the `README.md`. If you do this, specify in the README where the data should be saved in the repo. The user should not need to rename any files in order to get the code to run. In addition, your `.gitignore` should already be set up to ignore these files.
     - If the underlying dataset is non-public and cannot be uploaded, talk to Professor Ganong or Shi to come up with a reproducibility plan.

4. Git (5% of total)

   - Create multiple branches as you work for different pieces of the analysis. Branches may correspond to work done by different partners or to different features if you are working alone.
   - Your final repository should have one branch: `main`

While we will lean toward giving the same grade for all group members, it is possible that individuals may receive different grades if we see strong evidence of differential effort in the commit history.

Writeup (15%)

- You will then spend *no more than 3 pages* writing up your project.
- The primary purpose of this writeup is to inform us of what we are reading before we look at your code.
- The top of your writeup should include the names of all group members, what day/time is their lecture section, and Github user names.
- You should describe your research question, then discuss the approach you took and the coding involved, including discussing any weaknesses or difficulties encountered.
- Display your static plots and briefly describe what they show and how this relates to your research question.
- Discuss your Streamlit app and briefly describe what it shows and how it relates to your research question.

Responsiveness to presentation feedback (10%)

A professor or a head TA will give you written feedback on your presentation. Part of your grade for final submission will be on how good of a job you did responding to that feedback.

Public Repository

Your repository should be based on the template repo.

Material in the root directory

- Documentation and Meta-data
  - A `requirements.txt` or `requirements.yml` file
  - A `.gitignore` file that ignores unneeded files (e.g. `venv`)
  - `README.md` file that links to your Streamlit Community Cloud dashboard, documents data sources, and describes how they are processed
- Writeup
  - A file named `final_project.qmd`
  - Two knitted versions of your writeup: one HTML and one PDF'd

Material in subdirectories:

- Data
  - A folder named `data` that contains the initial, unmodified dataframes you download and the final versions of the dataframe(s) you built.
  - If you pre-process any data, you should do is in a file called `preprocessing.py`, which takes input from `data/raw-data` and outputs into `data/derived-data`
  - If a dataset is greater than 100MB, it can hosted on Drive or Dropbox and the link should be provided in your `README.md` file
  - If the dataset is not publicly available, your `README.md` file should include directions on how to gain access.
- Writeup dependencies
  - Any additional files required to knit `final_project.qmd` (e.g., externally-generated graphics)
- Streamlit app
  - A folder named `streamlit-app` that contains the code and any additional files needed to deploy your app
  - Deploy the app in Streamlit Community Cloud and include a URL

Tips for Success

Multiple Datasets

- Multiple variables from the same dataset does not, in general, constitute multiple data sources. If you believe you have a special situation where this guidance should not apply, send Professor Ganong a message in Ed to ask.

Sensitive Datasets

Sometimes datasets are not publicly available or are too large to share via GitHub (100 MB single file size limit). In this case, you should upload data onto Box and add the shared link to your `README.md` file on how to access the data.

Visualization Guidelines

We expect each of your visualizations to follow the visualization guidelines from `viz_1_intro`:

- All axes and units are properly labeled and legible
- No words or data points are cut off in your final output
- Encodings should be sensible and appropriate. Consider explicitly justifying which data types and encodings you chose in your writeup.

Static vs. Dynamic

- Only make a dynamic plot (i.e. a dashboard) when a static plot is insufficient to achieve your goals.
- We will discuss use cases for dashboards in more detail during the dashboard lecture. Slides are available on the student repo (look for slide ~53)

Policy Implications

- Tell us about policy implications of your findings.
- We will be looking for you to only draw conclusions which are *supported by your data*; implications which are unrelated to or not supported by your analysis will result in point deductions.

Replicability

Since we are going to re-run your code, get ahead and act like a grader yourself!

- Ask a friend outside your group to test the code for your final project
- Ask an AI agent to re-run the code for your final project (we will cover how to do this in week 8)

Presentation

- Write out exactly what you plan to say. However, do not just read a script. Instead, memorize the script and speak directly to the audience. If possible, present conversationally.
- We expect that you will interpret every plot you show, just as we interpreted the findings from plots in the visualization lectures.
- Time is scarce. In each plot you show, we expect that there is a clear takeaway or headline message.
- We expect to see fairly polished static visualizations (see guidelines above), but it is ok if the dashboard is a work-in-progress by the time of the presentation. For the dashboard, we expect there to be at least one portion that is finished that you will demo.