30538 Final Project: Reproducible Research

Peter Ganong and Maggie Shi

2024-10-13

Project Description and Instructions

The goal of this project is to showcase your knowledge of Python by applying it to a research project about a policy topic you are interested in. This is not a research methods course, so the quality of the research is ancillary to the quality of your programming. You will be graded on the parts: coding, writeup, and an in-class presentation.

You may work on this project alone, or in *groups of up to 3*. All groups must be formed declared in the Canvas proposal before any work is done - it is not possible to join one after. Groups can consist of members from any section in PPHA 30538. However, you are responsible for ensuring that every group member can attend the presentation slot you sign up for and that the presentation slot occurs during one of the members' sections.

It is required that you use GitHub, and we may use your past commits to understand your thought process for partial credit. If you working in a group, note that as we are grading we will be looking for multiple commits per individual throughout the project. Expectations for the scope of the project will be higher for group than for individuals, and the division of labor should be approximately evenly across both individuals. While we will lean toward giving the same grade for all group members, it is possible that individuals may receive different grades based on the commit history.

Grading

Coding (70%)

The code for the project should have the following components:

1. Data wrangling (25%)

- You must use a minimum of *two* datasets, at least one of which should be retrieved automatically from the web using APIs or web scraping.
- All processing of the data should be handled by your code, including all merging and reshaping.
- Any automatic data retrieval must have an option to toggle accessing the web off if the data is already downloaded.

2. Plotting (25%)

   - From that data, you will create a minimum of *two* static plots using `altair` or `geopandas`
   - As well as one `shiny` app with one dynamic plot
     - You can also add additional dynamic plots into your app to substitute for a static plot. So, a `shiny` app with 3 dynamic plots will count for full credit.

3. Reproductibility (10%)

   - The project and files should be structured and documented so that someone could fork your repository and reproduce your results (see "Final Repository" below).
   - This means that your README should document the order in which codes should be run, and what needs to be edited (e.g., where the user should set their path) by the user.
   - If a dataset is retrieved automatically, then the final results do not have to reproduce exactly but the code should run smoothly even if the underlying data changes.

4. Git (10%)

   - You should submit your project as a Git repository.
   - Your final repository should have one remaining branch: `main`
   - We reserve the right to check the git commit history to ensure that all members have contributed to the project.

5. Extra credit: text processing (up to 10%)

   - Introduce some form of text analysis using natural language processing methods discussed in class.

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.
- 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 them and your Shiny app. Discuss the policy implications of your findings.

- Finish with a discussion of directions for future work.
- The top of your writeup should include the names of all group members and Github user IDs.

Presentation (15%)

- Presentations will occur in the last 2 lectures of class.
- On the day of the presentation, one of the group members will be *randomly selected* to give a *8-minute in-class presentation*. All group members must present.
- The presentation will largely mirror the structure of the writeup, but will be more focused on discussing the research question and results as opposed to explaining the details of the coding.
- Discuss each static figure and demo your Shiny app.

Final Repository

Your final repository must contain the following:

- Documentation and Meta-files
  - A `README` file summarizing project and code
  - A `requirements.txt` file
  - A `.gitignore` file that ignores irrelevant files or large data files
- Writeup: a user should be able to knit your `.qmd` file and re-generate the HTML version of your writeup
  - The `.qmd` file associated with your write-up
  - An HTML and PDF'd version of your writeup
  - A folder named `pictures` that contains the files for any pictures required to knit your writeup
- 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 are using a dataset that is automatically retrieved from online, you should archive a version of the data you pulled in the `data` folder, and indicate clearly in your code and the `README` file where a user could replace the data retrieval with the path of the archived data.
  - If the dataset is too large to be hosted on Github, it can hosted on Drive or Dropbox and the link should be provided in the README file as well as indicated in your code.

- Shiny app
  - A folder named `shiny-app` that contains the code and any additional files needed to deploy your app
  - A user should be able to deploy your app directly from the command line within this folder

Key Dates

- By November 1
  - Proposal submitted to Canvas quiz
  - (Optional) meeting with Peter, Maggie, or Ozzy
  - Sign up for presentation slot
- December 2- December 5: in-class presentations
- December 7, 5PM: final repository submitted via Gradescope

Suggestions and Tips

- We encourage you to create a project on a subject that is relevant to your interests, and other Harris classes. If your research idea is a much larger project, think of how you can develop a basic framework for it using this project, which can then later be expanded into a proper research project.
- If you feel stuck coming up with research ideas, feel free to contact the teaching team so we can discuss your interests and make suggestions.
- You may use libraries and methods we did not go over in class, but ones that we did go over should be preferred if they duplicate the functionality.
- Remember all citation rules from the academic dishonesty policy in the syllabus.
- Effort put into organizing your code and making it readable (e.g., using functions, using variable names, and adding comments) will be rewarded.
- Effort put into cleaning up your output (e.g., clearly labeled graph axes, good use of colors or marks) will also be rewarded.
- Your final GitHub repo should be organized - do not leave useless files there (e.g. `DS_store` files), and keep things in folders.
- The entire point of reproducible research is to make it possible for others (and for a future you who has had time to forget what you did and why) to understand, replicate, and modify your work. Keeping this in mind as you work will be good for your grade, and helpful to you in the future if you expand on the project.
- Free shinyapps.io pages will run slowly, particularly if your data is large. Keep this in mind when planning your Shiny app, especially if you have large shapefiles.