Final Project: Data Analysis

- **Statistics 159/259, Fall 2025**
- **Due Wednedsay 12/17/2025, 11:59PM PT**
- Prof. F. Pérez and GSIs J. Butler and S. Andrade, Department of Statistics, UC Berkeley.
- This assignment is worth a maximum of **100 points**.
- Assignment type: **group**.

This project will be a free-form analysis; your task is to find an online dataset, think of some interesting questions that you can ask of it, and then perform a complete and thorough analysis of the data to answer your questions. Most importantly, you will share your complete set of results in a well-documented and organized manner, using the practices we've been developing during the course.

Project Deliverables

Data

You should aim to store your data in your GitHub repository. Depending on the size of your data, you may choose to include it in the repository or not (use your judgment - up to a few dozen megabytes it's not a problem, something in the Gigabyte range is too big to put into a repository). If the data cannot be included in the repository, your code should access it remotely and store more manageable subsets of it relevant for plotting (similar to what we recommended for Project 1 and Project 2). If you don't include the data in the repository, consider the implications of doing so for long-term reproducibility: is the source you are using a reliable one? Will it be there in 10 years? You can also explore other ways of storing datasets, such as Zenodo or Huggingface.

Functional structure of your code and testing

You should identify steps amenable to be put into standalone functions and write a proper function for those, with a complete docstring describing the purpose of the function, their inputs and types, and their outputs.

You may choose to put analysis functions into a pure Python file that you import separately or in the Jupyter notebook itself. Make the choice that you find cleanest/most fluid for your workflow.

Whether you do it in a notebook or python scripts, make sure to write at least a few tests for utility functions that you write.

Note: this is a *requirement*. You must structure your code to include at least two functions that you import from a separate python script, that you test and document properly (feel free to use more as your analysis dictates).

Analysis notebooks and supporting code

Break down your analysis into as many notebooks as is reasonable for convenient reading and execution. There is no hard and fast rule for this, just as there isn't for how many paragraphs or figures a good scientific paper should have. You should consider readability, execution time, total notebook length, etc, as criteria in your decisions. This essay by Stephen Wolfram, the creator of Mathematica (part of the inspiration for the Jupyter Notebook), has some good thoughts on what makes a well written computational narrative.

Consider how to "chain" your analyses with intermediate results being stored so they don't need to be recomputed from scratch each time. Also, be mindful of saving key figures you may want to reuse in your report to disk, so the main narrative notebook can reuse them for display and discussion.

You must also render all of your analysis notebooks each as a separate PDF file using MyST and store them in a directory called `pdf_builds`. You're going to be making your projects into MyST projects and deploying them to websites later; as discussed in the MyST lecture, MyST also let's you render notebooks and markdown files as PDFs.

Main narrative notebook

Write a `main.ipynb` notebook that summarizes and discusses your results. It can contain figures referenced from disk or simple summary information (for example if you want to display part of a dataframe) from variables read from disk, but *no significant computations at all.* Think of this as your "paper".

Discuss the assumptions you can make about the data to justify your analysis. You have no control over the original data acquisition and measurement, and for many of you this project will likely fall under the broad purview of *Exploratory Data Analysis.* If you propose any statistical hypothesis/model in your analysis, discuss your justifications, considering how the data was acquired. We are not expecting you to spend a lot of time dealing with how to best model or describe the data. Try to identify a few scientifically interesting but simple questions you can address and then focus on writing the code for such an analysis.

Author Contributions section: your `main.ipynb` notebook should contain, at the end, a brief section titled *Author Contributions.* This section should indicate, for each team member, what they did in the project. It doesn't have to be a long, detailed story, a few sentences per person should suffice. All team members must agree to the language in this section. (By the way, this is standard practice in many scientific journals). While in principle the project grade is the same for all team members, we reserve the right to lower the grade of anyone who doesn't contribute to the team effort.

Like with each of the analysis notebooks in the previous section, you should also build your main narrative as a PDF using MyST. Please also include a bibliography of works that you cite. This could be previous work that motivates the question you are investigating and why it's interesting, any methods that you might be employing from the literature, or the dataset you are using. We ask that you do this by including all of your references in a `BibTeX` file and cite them in this main narrative by referencing the relevant entry in your `BibTeX` file. For more details, see the Lab 8 material.

Reproducibility support

- Environment: like in previous homeworks, provide an `environment.yml` file that creates an environment with all necessary dependencies to run your analysis. We recommend you to create a new virtual environment with this project that you can update as you include new libraries.
- `Makefile`: you will also create a `Makefile` with `env` and `all` targets. The `env` target should make the environment with all necessary libraries, and update it with the relevant packages in `environment.yml` if it is already installed, and `all` should run all notebooks (using `nbconvert`).
- Don't forget to set random seeds for anything that has a stochastic component.

Good repository practices

- Work as a team. Work collaboratively with your team members by adding commit messages and working in different branches, merging them into the main branch as your group sees fit. Also, consider taking advantage of issues as you review each other's work asynchronously, and even consider opening issues and opening PRs that close them, as we practiced in Peer Review 2.
- `README.md`: Write a short self-contained description of the project, including the motivation behind your project and the analysis you conducted. Include all relevant information about how to run the analysis, including installation steps (for any packages, environments, etc.), testing, and automation. A good README should be short but also provide all the useful information to help the user to start running the analysis.
- `LICENSE`: you should explicitly state which licensing conditions apply to your work (see Github's help). We strongly suggest looking at Victoria Stodden's ENABLING REPRODUCIBLE RESEARCH: LICENSING SCIENTIFIC INNOVATION. She suggests a "Reproducible research standard" that licenses code, data and text/media materials

in comparable terms that maximize resharing potential while providing due credit guarantees for the original authors.

- `.gitignore`: provide a git ignore file that prevents the automatic inclusion of unwanted files. This also helps you avoid noisy `git status` output that lists things you know you won't actually want to put under git's control.
- A `binder` link: also deploy your repository on Binder, including the badge in your repository and ensuring that all notebooks can be executed for enhanced reproducibility.

Website

Finally, we ask that you deploy your analyses to a MyST-generated website, hosted on GitHub Pages. Please take the time to consider how to best organize your websites to communicate the overall narrative of your project, from the main narrative document, to the individual analysis notebooks. Please also include the Binder badge somewhere in the website.

Release

We also ask that you tag your repository on GitHub when you are finished, marking a version release of your project. We also ask that each of you make an ORCID, archive your release on Zenodo and reference the ORCIDs of your group members, and then include the resulting DOI with a DOI badge in your GitHub repository. See the Lecture 12 slides on how to do this.

Grading

We will evaluate the project as a whole. All the previous elements play a central role for the purposes of this class, and we expect to see all of them included in the final deliverable. The project is worth 100 points, and 30% of the final grade. To get some inspiration about how to organize and structure your projects, feel free to take a look at the following projects which scored very highly in the Spring 2023 version of this class.

- Airbnb Europe Dataset Analysis
- An Analysis on Animal Shelter Data from Sonoma County, CA
- Analysis of the relationship between cigarette sales per capita and median income by States in the US