Project 1 – Explanatory visualization

PERE-PAU VÁZQUEZ & OSCAR ARGUDO & IMANOL MUÑOZ

Information Visualization | 2025 - 2026

1. Objective

The goal of the project is to create an explanatory visualization using Altair and Streamlit to analyze the NSF terminated grants from the Trump administration. We attach three files, one containing a list of terminated grants with details on those, another containing a subset of grants where there is information on whether the grant was present in the list of grants to be killed according to senator Ted Cruz, and a third one that contains a list of words that are flagged as potential targets for Trump's administration grants. You need to join the two initial files and clean the resulting file properly.

The purpose of this first practical work is to create a series of visualizations that analyze the impact of the cancellation of grants has been. To do so, you need to create visualizations that answer the following questions:

- **Q1:** How are the cancellations distributed by states?
- **Q2:** What are the institutions that have been more affected in terms of number of cancelled grants. How does this compare to the others?
- **Q3:** What are the institutions that have been more affected in terms of budget, and how does this compare to the others?
- **Q4:** Is there any correlation between the cancelled grants and the list of flagged words?
- **Q5:** Is there any correlation between the cancelled grants and the list of grants in Cruz's list? And with respect to reinstated grants?

You need to clean the datasets and augment those datasets when necessary. The cleaning process must be explained in the delivery document, in such a way that we could reproduce the steps to obtain the same results.

For each question, a chart (or more than one) needs to be designed and implemented. You may tackle the problem in different ways: there is not a single solution. You also need to provide a text of a maximum of 200 words per question (and for the final visualization too) where you explain the design decisions and how these help the users to answer the questions properly. This should include aspects such as: what type of chart did you select and why, what were the different steps of design process you followed, what changes you applied to improve legibility, reduce clutter, distinguish elements, how would a person answer the question with your chart, what other alternatives did not work. Be concise. You do not need to erase the previous steps in design you followed, they are actually encouraged. But only a final description per task is necessary.

A final visualization is also required that includes all questions. In this case, you also have to explain, in a maximum of 200 words, what changes were made to make charts consistent, aesthetically pleasant, etc. The final visualization must be then incorporated in a Streamlit application (https://streamlit.io/). You have to teach yourselves on how to implement applications using Streamlit, which is very easy. You can start checking the initial videos of this playlist: https://www.youtube.com/playlist?list=PLtqF5YXg7GLmCvTswG32NqQypOuYkPRUE, or you can start with some of the videos by Fanilo Andrianasolo: https://www.youtube.com/@andfanilo. He has a subsection for starters with some interesting videos.

DATA

Data is included in Racó.

DELIVERY INSTRUCTIONS

The work can be implemented in pairs or individually. You have to provide the clean data. You have to describe the cleaning procedure, so that we can generate the clean data from the raw data following your steps. This description must go in the Colab document.

The delivery must consist on a single ZIP file with a name that includes the authors, that contains the datasets (raw and clean), the Colab file(s) (*ipnyb*), the Python Streamlit application, and optional extra documents if required. The ZIP, Streamlit, and Colab files must be named after the names of the authors. Treat the Colab document as a report, include titles, boldfaces, etc., to make it easier to read.

The deadline for the delivery of this lab project is the 9th of November.

Upon delivery, an interview with the teacher of your lab will be held to help us understand your development. Such an interview is compulsory; it is part of the delivery. During the interview, the teacher will ask you questions about the project, such as design decisions, or implementation details. How you answer those questions will determine your individual grade. These interviews will happen during a regular lab session, or in the teacher's office. They will likely be individual, independently of whether the project was done in pairs or not.

IMPORTANT REMARKS

Besides your performance during the interview, the project grade will consider the number of variables included in the visualizations (these may include new calculated variables, such as aggregated data, averages, maxima, minima, etc.). Additionally, we will value the number of non-trivial tasks (adequately described in the documentation) that can be properly solved with your visualization tool. In this sense, adding extra tasks/questions is valued positively.

Don't leave the project for the last day or do the minimum amount of work. In case of doubt, ask us whether the current work is enough or needs more effort.

THIS CHECKLIST SERVES AS GUIDANCE FOR YOUR DELIVERY

Global checklist:

- Name the file after the name(s) of the author(s).
- Include the name(s) of the authors also as the first line in your notebook.
- Include the clean data.
- Compress all files in a single zip (do not use RAR or other formats) file.
- Also include the names of the authors in the notebook (e.g., a text cell showing who authored the document) and in the Streamlit app (e.g., in an "About" option).
- Ensure the names of the data files inside the notebook correspond to the ones you deliver.
- Make a single delivery per group.

- Ensure you properly cleaned the data.
- Ensure the code executes without errors: last-minute changes may lead to typos.
- Ensure you include a step-by-step design process (does not need to include all minimal steps, if you prefer, but do not forget to include the discussion on why do you change something).
- Do not mix charts with other technologies, everything must be created using altair.
- Ensure they do not include non-properly cleaned data (e.g., N/A or undefined fields).

Google Colab document. For every chart, you must consider:

- Color blindness (e.g., coding anything just with a red-green palette may be problematic).
- Check the consistency of colors across the whole visualization (same color, same meaning in different charts).
- Do not forget to add meaningful titles, labels, messages if necessary...
- When using colors with opacity different from zero, check the interactions with the other elements (are they visible?).
- Think whether you need to normalize values.

For the Streamlit vis:

- All questions should be solved in a single page (with not much scrolling, e.g., maximum double the available screen on a 15" laptop screen).
- The application should run with a simple "streamlit run <application_name>" in any computer, everything should be available in the same folder or addressed properly.
- Do not forget to include a final visualization that answers all the questions.
- Align things, be consistent. You can make use of both vertical and horizontal alignments to facilitate comparisons.
- Extra questions you answer must also go into the final vis.
- Ensure charts can be visually compared properly (consider changing scales, palettes...)
- Use space cleverly (put related things together and unrelated things far away).
- The higher the number of variables (and questions answered) included, the better.
- If you do not use all your data in your vis, ensure you have filtered it previously, don't force Streamlit to execute with data it is not used.
- (Second lab) Ensure the default values make sense (e.g., if including selections, check that the initial configuration of the chart has a default state that is meaningful).
- (Second lab) Minimize the number of interactions required to solve the problems.