CPSC 436V: Project Proposal

Over the rest of the term you will work in teams of three towards creating an interactive visualization project using JavaScript & D3. *Read this document carefully!*

You'll form a team, submit three milestones, and give an in-person demo of your final visualization:

- **Team formation** *(Due: Feb 14, Canvas → People → Groups)*
- **Proposal: A detailed project plan** *(Due: Mar 7, Canvas)*
- **WIP: A first implementation of your visualization based upon your proposal** *(Due: Mar 25, git repo)*
- **Final project and write-up** *(Due: Apr 8, git repo)*
- **In-person demo** (approx. 5-8 minutes per group; date TBA, after final submission)

Important: You have only grace days for individual assignments; not for the team project. Milestones submitted after the due dates will be graded with 0 points.

General Project Requirements

Implement a visual analysis app or interactive data-driven story (narrative visualization) in D3 based on the following requirements:

- Publicly available dataset(s).
- At least 3 different views / visualization components *(e.g. 3 bar charts only count as 1 component)*.
- At least 1 of these views must be an "innovative view" that is either (a) an extension of an existing visualization type, or (b) a novel visualization type.
- Multiple views coordinated with linked highlighting. A click/hover/selection interaction within one view must trigger a change in a different view. At least 2 views need to be linked. Ideally, these views are linked bidirectionally.
- At least 2 UI widgets that allow users to filter the data or update certain views interactively (e.g. dropdown, radio button, range slider, calendar).
- Include interactive tooltips, in at least one view, that are shown when users hover over marks.
- You don't necessarily need to show everything within a single-screen dashboard. You may want to use a tab navigation or a scrollytelling approach.
- We don't allow custom backends (Node.js, Python, etc) and database systems, such as Postgres or MySQL, although they could facilitate more powerful applications. In this course project, you should put your efforts into the front-end development (JavaScript, D3, HTML, CSS). Your web application should be stand-alone and have an `index.html` file as entry point, similar to the programming exercises. You

can either store static (*optionally preprocessed*) datasets in your git repo or access live data through an API.

- You can use CSS frameworks, such as Bootstrap or Materialize, and include external libraries (jQuery, leaflet.js, moment.js, ...) but you need to implement visualizations in D3. Layers such as NVD3, Vega-lite, Highcharts, etc. are not allowed.

Talk to a TA in the lab to discuss your idea, particularly regarding the *innovative view*. Note, if you have good arguments why you can't fulfill all of the requirements above because you want to include other features, we may be able to accomodate that desire, but make sure to clear it with a TA first.

The goal of the project is to synthesize the material from both the programming exercises and the foundational exercises. We will evaluate projects using the following high-level criteria:

- Visual design: Effectiveness of visualizations and interactions, which should link back to usage scenarios, tasks, and intended audience.
- Level of technical difficulty and quality of implementation.
- Addresses the goals and requirements.
- Quality and clarity of your writing.
- Teamwork

You will need to submit peer evaluations as part of the WIP milestone and final-project submission. Also, we will aks you to fill out a self-assessment at the end of the project. We reserve the right to assign different grades to each group member based on these assessments.

Team Formation

Form a team on **Canvas** (`People` → `Groups` → `+ Group`) by Feb 14, 11:59pm.

- Use the pinned Piazza post to find teammates.
- Work closely in teams of **three**! We do not allow teams with more or less than 3 people. If you have a very unusual circumstance with a strong need for a team of two, we will consider your request. Talk to a TA about it in lab, and if your request is granted state who approved it in your proposal writeup.
- We will create github team repositories after you formed a team.

Submission

Submit your project proposal as a *PDF* on Canvas by Sat Mar 7, 11:59pm.

You will need to submit work for the other milestones in your git team repos (github.students.cs.ubc.ca). All development should happen here. We'll be considering your commit messages when grading the project: be sure to commit early and often, and to use informative commit messages.

Project Proposal Requirements

When submitting your project proposal as a single PDF, please include the following sections in this order:

1. Basic info
2. Overview
3. Description of the data
4. Usage scenario & tasks
5. Description of visualization & initial sketch
6. Work breakdown and schedule

Your proposal should be no more than 1,500 words of text (for context, this equates to a little over 1.5 pages of single-spaced Times New Roman size 11 text). You will also submit 1-5 sketches of your application, included in the same PDF (which should be no longer than 3 pages in total).

Sections 1-4

The first four sections of your proposal will be marked as a whole, and you will be assessed on the quality and clarity of your writing, the feasibility of what you propose, and your initial project description and sketch.

Section 1: Basic Info

- Project title
- A list of all team members and student IDs (IDs from github.students.cs.ubc.ca).

Section 2: Overview

A few sentences that describe what problem your visualization is tackling and how. You don't necessarily need to create a project that is purely focused on exploratory data analysis. You may choose to tell an interactive data-driven story intended to be consumed by the general public. The theme of your visualization can draw from any topic, including current affairs, history, natural disasters, and research findings from the sciences and humanities. State the intended audience. Be brief and clear.

Example:

> Missed medical appointments cost the healthcare system a lot of money and affects the quality of care. If we could understand what factors lead to missed appointments it may be possible to reduce their frequency. To address this challenge, I propose building a data visualization that allows health care

> administrators to visually explore a dataset of missed appointments. My app will use show the distribution of factors contributing to appointment show/no show and allow users to explore different aspects of this data by filtering and re-ordering on different variables in order to compare factors that contribute to absence.

Section 3: Description of the data and data preprocessing

You are free to choose any publicly-available dataset(s). You may use one of these resources:

- Kaggle Datasets
- Google Dataset Search
- Data is Plural (or as Google Sheet)

In your report, briefly describe the dataset and the variables that you will visualize. All data has to be publicly available since we will publish the final projects on the course website. Provide a link to your data sources.

Please note, if your dataset has *a lot* of variables and you plan to visualize them all, then provide a high level descriptor of the variable types, for example say the dataset contains *demographic variables* instead of describing every single variable.

Example:

> I will be visualizing a dataset of approximately 300,000 missed patient appointments. Each appointment has 15 associated variables that describe the patient who made the appointment (PatientID, Gender, Age), the health status of the patient (Hypertension, Diabetes, Alcohol intake, physical disabilities), information about the appointment itself (appointment ID, appointment date), whether the patient showed up (status), and if a text message was sent to the patient about the appointment (SMSsent). Using this data I will also derive a new variable, which is the predicted probability that a patient will show up for their appointment (ProbShow).

In the above example, specific variables names are indicated in the parenthesis; remember if your dataset has *a lot* of varibles stick to summaries and don't provide specific variable names. The example also differentiates variables that come with the dataset (i.e. Age) from new variables that you might derive for your visualizations (i.e ProbShow) - you should make a similar distinction in your write-up.

Preprocessing

Do you need to preprocess or clean up the original data? Do need to join multiple datasets? How do you plan to implement the processing pipeline? (i.e. in a separate tool, such as Trifacta Data Wrangler; R scripts that are executed once; on the fly in JavaScript).

Section 4: Usage scenarios & tasks

The purpose of the usage scenario is to get you to think about how someone else might use the web application you're going to design, and to think about those needs before you start hacking. Usage scenarios are typically written in a narrative style and include the specific context of usage, tasks associated with that usage context, and a hypothetical walkthrough of how the user would accomplish those tasks with your visualization. If you are using a Kaggle dataset, you may use their "Overview (inspiration)" to create your usage scenario, or you may come up with your own inspiration.

Example usage scenario with tasks (tasks are indicated in brackets, i.e. [task])

> Mary is a policy maker with the Canadian Ministry of Health and she wants to understand what factors lead to missed appointments in order to devise an intervention that improves attendance numbers. She wants to be able to [explore] a dataset in order to [compare] the effect of different variables on absenteeism and [identify] the most relevant variables around which to frame her intervention policy. When Mary logs on to the "Missed Appointments app", she will see an overview of all the available variables in her dataset, according to the number of people that did or did not show up to their medical appointment. She can filter out variables for head-to-head comparisons, and/or rank order patients according to their predicted probability of missing an appointment. When she does so, Mary may notice that "physical disability" appears to be a strong predictor missing appointments, and in fact patients with a physical disability also have the largest number of missed appointments. She hypothesizes that patients with a physical disability could be having a hard time finding transportation to their appointments, and decides she needs to conduct a follow-on study since transportation information is not captured in her current dataset.

Note that in the above example, "physical disability" being an important variable is fictional - you don't need to conduct an analysis of your data to figure out what is important or not, you just need to imagine what someone could find, and how they may use this information.

Section 5: Description of your visualization & sketch

Building from your usage scenario, give a high-level description of the visualization interface you will build. What are must-have features without which you would consider your project a failure? Remember to be realistic since you are actually required to implement this interactive visualization, and you will be assessed on how much, and why, your finalproject deviates from this initial proposal.

Briefly describe the characteristics of your "innovative view component" that is either an extension of an existing visualization type or a novel visualization type. What makes it special?

Your sketch can be hand-drawn or mocked up using Powerpoint, a graphics editor, or wireframe tools, such as Balsamiq. Most likely, you will need more than one image to show your proposed visual design and interaction sequences. Don't use more than five sketches. Please note, this is a very basic illustrative guide that should help you during the implementation of the first prototype. It is by no means the limit of what you should submit as the final project.

Section 6: Work breakdown and schedule

Include a list of project milestones, breaking down the work into a series of smaller chunks that are meaningful and useful for your specific project. Your milestone list shouldn't just be completely generic (eg "requirements analysis, design, initial coding, iterative refinement, final writeup"). While something analagous to these stages may well make sense for your project, you should also be thinking about how to break down the work into components that are appropriate for your project in specific (eg "create initial static version of view X", "link views X and Y to each other"). As above, think about these questions: What are must-have features? What are optional features that can be implemented after the second milestone, if there is time?

Milestones must be associated with two numbers: the estimate of the number of hours each chunk of work will take, and the target date for completion of that chunk. Also consider how you'll split up the work between group members, to distribute the work equally, and write that plan down in this section.

The scope of the work should be roughly 50 hours per person (150 hours total across the team of three), for the entire project -- across all three milestones.