How to do well on this project

- **Balance execution and ambition.** Grading for this project will be a mixture of the quality of the work and the degree of difficulty. If you are downloading a Kaggle dataset, do not expect high scores unless you are extremely thorough and polished. If you are combining three different APIs in complicated formats with slightly mismatching IDs, we will tolerate more messiness.

- **Start now.** Most of your initial ideas will not work. The phases are designed to force you to work in stages, but the project will take time and you will get stuck. Leave space to think about problems and find solutions. TAs will have more time to give feedback before the deadline crunch.

- **If you are in a group, work as a group.** I (Prof. Mimno) have supervised hundreds of group projects over the last seven years. Most of the problems I hear about with group work are either â€œPerson X isnâ€™t doing anything!â€ or â€œPerson Y wonâ€™t let me do anything!â€ Reflect on this. Communication and clear expectations sound obvious, but it can be difficult to put these into practice when everyone is busy and especially now when everyone is remote.

- **Work together.** The biggest correlation I have seen with group success is the ability to schedule and attend meetings. Do not divide the project into discrete tasks and staple them together at the very end; this (a) never works and (b) is really obvious. Everyone in a group should be contributing to every part of the project to some degree.

Rubric

The formal requirements for the project are listed in a previous section. This section lists characteristics that we associate with more or less advanced projects.

Your job as a writer and as a data scientist is effectively communicating what your dataset is about. The technical details of the data set will be described in your datasheet (follow a template as in the examples in sections 3.1-3.5 of this article on datasheets: https://arxiv.org/pdf/1803.09010.pdf). We expect the total length should be 1500-3000 words. Inside this range, length will not be a factor in grading.

The Introduction should be the exposition of the article where you can use less rigorous language. Your language should be generally accessible. Aim for this to be readable by someone who hasn't taken this class (maybe your roommate, your family, or you at the start of the semester). It should still be formal, but someone should come to the end and want to read more. 538 articles (fivethirtyeight.com) might be a good baseline tone for this.

Advanced introductions will immediately tell us what the setting is, what you found, and why it matters. They will add details as they are needed. Language will be polished and free from errors (Note: if your group does not include a native English speaker, make a note of that). Beginning writeups will be less focused and organized. They may jump to technical details without explaining why results are important. They may have spelling and grammatical errors, or awkward or incomplete sentences, indicating that they were written in haste and never reviewed.

Datasheet: As described above, in the style of Gebru et al. Think of this as the â€œorigin storyâ€ of your data set. Answer all of the questions listed in the previous section. You can write this in any style as long as it's easy to read as a Q&A. Datasheet will be graded on content, not style. Follow sections 3.1- 3.5 (Motivation to Uses) from https://arxiv.org/pdf/1803.09010.pdf.

Data analysis and evaluation of significance: Here you will clearly detail your methods used in each part. Qualitative claims made in the exposition should have numerical backing here (instead of â€œX is larger than Yâ€ write â€œX is 3.65 times larger than Yâ€). This should read like a scientific paper, but

does not need to be "stuffy" or overly indirect: "we did ..." is more natural than "... was done". A reader should be able to replicate your experiments and findings via their own code after reading this.

Itâ€™s important to organize your analysis. Common organizational patterns:Â

- Big to small. Start with a high-level description of the complete dataset, then add more detail and increase specificity until you are looking at individual data points.

- Small to big. The opposite: start with individual data points, then â€œzoom outâ€ progressively until you get to a broad, top-level overview.

- Bites at the apple. Visit different facets of the dataset. This could be subsets of the observations along different criteria, or a series of aggregate views where you are grouping by different variables (eg alumni by state, then by industry, then by major).

In most cases you will try many possible analyses. You don't have to report everything that you did. Find a good selection that makes sense. In most datasets there are potentially thousands of different functions that you could analyze. Why are the ones you chose the most interesting?

Advanced analyses will be clear, logical, and methodical. Mathematical modeling will have clear purpose that answers relevant questions and contributes to an overall perspective. Results will be contextualized with significance tests or comparisons to alternative simpler explanations. Reasonable "next questions" should be followed or acknowledged, though you don't have to follow every lead. Beginning analyses will be disorganized and haphazard. They will apply models without context or purpose. They report results without considering whether those results are meaningful or random

noise.

Code: As notebooks with evaluated cells. We won't run them or attempt to debug errors. The most crucial part is to comment your code so that we can quickly understand what it does. This doesn't need to be exhaustive, but you should be keeping your reader updated on what's going on every few lines. Some code may be oriented towards pre-processing and data curation, other code may be oriented towards analysis and presentation of results.

Advanced code will be succinct and well-organized, with comments that indicate expected uses and assumptions for inputs and outputs. Repeated tasks will be broken into functions. Variable names will be informative. Points of failure are anticipated and checked for.

Beginning code will be unclear and disorganized, possibly with large sections of unused code. Variable names will be ambiguous or misleading. Comments will be missing or will simply repeat information that is obvious from context. Variables will be short and uninformative.

Conclusions should reflect on what you accomplished and where you might go from here. These can be hard to write without feeling repetitive. The conclusion is a good place to mention things that you tried that did not work, or data that you could not find but that you would add in a hypothetical further version.

A common question is "do I need to ... to get a good grade?"Â

It's an open-ended project with additive grading. We only give points for what you do, we never take points off for what you don't do. There are many things that we consider difficult (combining multiple datasets, reformatting data, collecting from web pages), so if you find that any of them make sense, we will recognize that in our consideration of how ambitious you are. None of them are required.

What we want you to do is make an argument based on a data set. If the perfect

data set already exists, great! You have more time to work on the details of the modeling and the presentation. In many cases the data set you want doesn't exist in the form you are looking for, and you need to do some work to create it. We want you to have tools to do that if needed. But even if you think you have exactly the data you want, you may find that in investigating it you realize that there are additional questions that require more data collection.

Students often find this kind of open-ended project difficult because it requires more independence and feels more risky. We've built in multiple low-stakes checkpoints to help make sure we give you feedback and reduce the feeling of "flying blind". But it's also the most realistic and valuable experience to prepare you for what you will do after graduation, and the thing that alums most often remember years later.

I also recognize that doing open-ended projects is more difficult than it has been other years. It's harder to meet with teammates and with course staff. We will take this into account when giving final scores.