INFO 2950 Fall 2023 Handbook

Credit to Irena Papst, David Mimno, and Carlo Tomasi for contributions.

Course Goals

Our goal is to give you the ability to make arguments using data. This will require a combination of statistical methods, data processing, and real-world knowledge. The core of our practice will be a semester-long project with multiple phases. The schedule for the class is built around the phases of a typical data science project: data collection, exploration and summarization, model fitting, and hypothesis testing. Although we will learn about programming and statistics, it is just as important to focus on recognizing the dangers and limitations of data and modeling.

It's important to remember where the actual learning happens. Lectures will *introduce* mental models, technical terms, and programming practices, but this is only the beginning. In Friday discussion sessions we will collaboratively work through additional examples and practice specific skills. Homework is where you really engage at your own pace and put everything together. Finally, open-ended projects are where you have a chance to demonstrate what you have learned.

Course Logistics

- We will use **Canvas** for:
    - Posting course announcements (don't forget to check/set your notification preferences so you don't miss important announcements)
    - Posting slides and code from lectures within 24 hours of class (under the "Modules" section)

- We will use **Gradescope** for:
    - Posting homework (released Thursdays) and corresponding solutions
    - Submitting homework Homework files (due Thursdays)
    - Regrade requests
    - Submitting final project work
    - Posting exam scans & grades

- For asking/answering course-related questions, we will use:
    - **Ed Discussion** (by either public or private post)
    - **Student** (**Office**) **Hours**, available on Google Calendar (linked on Ed Discussion)

- For special circumstances:

    - Fill out the **excused absences form** if you will be unable to attend class due to conflicting events (e.g. sports travel, medical issues, etc.): https://cornell.ca1.qualtrics.com/jfe/form/SV_eKgedR7nJYqdVga

    - For other special circumstances, especially events that cannot be reasonably foreseen (e.g. concussions, family emergencies), please email the instructors at info2950-instructors@cornell.edu regarding assignment extensions or makeup exams

- For other course materials:
    - While our course outline does not follow a specific textbook, we recommend two free texts as reading material to supplement this class. Our goal is to provide multiple perspectives on the methods we use:
        - Joel Grus, Data Science from Scratch, Second Edition. You can access this from the library website: https://newcatalog.library.cornell.edu/catalog/15116623, look for the "Availability" box on the right. The O'Reilly site will display a time, this is an estimate of how long it will take to read, not the amount of time you have left.
        - Jake Vanderplas, Python Data Science Handbook. Full text online at https://jakevdp.github.io/PythonDataScienceHandbook/

In-class logistics

Lectures are held on Mondays and Wednesdays; attendance is mandatory. Discussion sessions are held on Fridays; attendance is mandatory.

Lectures will include an active learning component involving personal whiteboards, markers, and tissue paper (for erasing), all of which we will distribute at the beginning of class. Whiteboards will be used to allow students to do scratch work, interact with in-class activities, and foster engagement with the course material.

Please make sure to return both whiteboards and *fully capped* markers to the bins by the lecture doors at the end of class, and please throw out used tissue paper in the trash cans. If you do not cap your markers, they will dry out and we won't be able to use them! Please do not let markers or whiteboards walk away from the classroom: they are for INFO 2950 use only.

Discussion sessions will involve hands-on practice with coding, led by TAs. Please make sure that you come to class with a fully-charged laptop in order to make instruction most effective.

Homework logistics

Homework schedule

In general, homework will be due on Thursdays and released on Thursdays. Homework will be released one week before it is due. **The single most frequent suggestion from students last semester was "start homework early.**" Friday discussion sessions provide an opportunity to make sure you understand what is expected and clarify the intention of homework questions.

How to convert assignments for Gradescope submission

Please follow these instructions *exactly* to generate your assignment submissions for Gradescope. You may be tempted to deviate, since there are many ways to generate pdfs from Jupyter notebooks; however, this can often lead to errors (like losing images or LaTeX formatting) which would result in losing points on your assignment.

The example below is assuming a homework file named hw0.

1. Finish your assignment in your local copy of *hw0.ipynb*. Ensure that all cells run correctly with no errors (when you hit the "Run All" button, everything should run in one pass)
2. Select File -> Download as -> HTML (.html). This will generate *hw0.html* (usually in your Downloads folder)
3. Open *hw0.html* in your web browser (you can do this by double clicking)
4. In your browser, go to File -> Print -> print to PDF in order to generate *hw0.pdf*. **DO NOT** use "Export to PDF" to generate a pdf as it will not export everything correctly.
5. In gradescope, submit two files: *hw0.ipynb* and *hw0.pdf*. **DO NOT** submit your hw0.*html* file.

Slip days

All students have a total of 10 slip days. A slip day is a one-day no-questions-asked extension. Each 24-hour period after a due date incurs one additional slip day. Please use these slip days if you need them; there is no bonus for *not* using your slip days at the end of the term. If you're facing issues that cannot be solved by taking your allotted slip days, please write to info2950-instructors@cornell.edu as soon as possible. We recognize that this will be an unusual semester and we want to help you succeed in this course as much as we can.

Students with disabilities

We're committed to making this course work for all students. We receive notifications from Student Disability Services and take those into account. For timed in-person exams you will receive information about accommodations.

Many students have situations that do not necessarily fall under SDS, but nevertheless can make a difference in your ability to thrive in the class. Write to the course instructors and we can start a conversation about how best to support you.

Academic Integrity

Every professor and TA will tell you that dealing with academic integrity is the worst part of this job. It feels horrible for everyone and takes enormous amounts of time and emotional energy that robs other students. You are encouraged to discuss homework with other students, but you **must do your own work**. Never copy from another student or allow another student to copy from you. Allowing someone to not learn is cruel and the opposite of friendship. All answers to a given homework problem will be graded by a single TA, who will be unlikely to miss overly similar answers. If we determine that assignments are too similar, everyone involved will receive a **negative** score. Copying is usually an act of desperation, not an intentional plan. Start work early so that you are not tempted!

We are committed to correct grading. If you think we have made a mistake (we do!), please send in a regrade request through Gradescope. You should hear back within a few days, and if you do not, please check in with info2950-instructors@cornell.edu. Please ensure that regrade requests are submitted in a timely fashion. Note that although it is helpful to describe what you think was incorrect, we will reevaluate your answer completely, so your grade may go up *or go down*. TAs are often willing to overlook small errors or to grade charitably, and may not be sympathetic to small-point requests if they feel the original grade was lenient.

Self-care

We've all been through a lot the past several years. Students have shown incredible resilience, but everyone has a limit. Don't forget to check in with others regularly and be patient with everyone, including yourself. Sometimes, it can help to talk to someone–a professional or a peer. It's ok to not be ok.

If you are not in a good position to do your best work, first think about using a slip day. 24 hours can make a huge difference when things are piling up.

If you need more than 24 hours, or if things feel like they're spinning out of control, *talk to the instructors*. A lot of students worry about sending "the email". ***Don't worry, just send it***. Here's how it looks from our perspective. First, you are not the first student who has run into trouble. We don't think less of you. Second, you are not bothering us. We *want* to hear what you're going through. What we hate is when students wait too long to ask for help (usually finals week), when we could have worked something out to get them back on track if we had only known a couple months earlier. **There is no more satisfying feeling as a professor than "catching" someone before they fall too far and helping them succeed**.

Student Hours (Office Hours)

Most of the substantive learning you do in class will not occur in lecture or discussions, it will actually happen when you're getting hands-on experience with the material, through homeworks and your final projects. This kind of learning generally happens in informal study groups and office hours, especially when students are able to help each other (perhaps with some support from a nearby TA or instructor).

Both instructors have office hours. A better term for this time is "student hours." This is time that professors make available for students in this class. These are often underused! Many students report feeling nervous or embarrassed to talk to professors, but remember that we chose this job because we like helping students. In a large class it's often hard to get to know individual students. We really like the opportunity to talk to you in small groups, and it's a great way for us to get feedback about what might be confusing about homework and how to explain it better.

A Google calendar with student hours times and locations will be linked on Ed Discussion and Canvas.

Midterm and Final exams

While you will rarely be asked to solve data science problems with pen and paper without access to the internet, exams are a great way to evaluate how well you are doing. You can think of these exams as practicing for job interviews.

The midterm will take place during usual class time on Oct 2, but will occur in multiple locations. You will be assigned a specific room. You must attend in the room assigned or we will not have an exam for you.

The time and location of the final will be determined by the registrar.

Final project

Goal

This project is designed to give you experience with the full cycle of data science, from collecting observations to modeling to making arguments. **Alumni often tell us that the final project was the most useful and memorable part of this class.**

It should be something you are proud of and can display as part of a portfolio for job applications. The idea is that you take what you've learned through the course (via lectures,

discussion sections, homeworks) and apply it to a specific domain of interest to you. To do well on the final project, simply **show us what you've learned**!

Important dates

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.

Note: slip days are not allowed for project submissions.

- **Sept 7:** Phase 0 *(coordination and planning)* of the final project due
- **Sept 21:** Phase I *(brainstorming)* of the final project due
- **Oct 2:** In-class midterm exam
- **Oct 19:** Phase II *(data collection and exploratory data analysis)* of the final project due; **groups cannot change after phase II is submitted**
- **Nov 2:** Phase III *(preregistration of analyses)* of the final project due
- **Nov 16:** Phase IV *(draft of final results)* of the final project due
- **Dec 4:** Phase V *(final results)* of the final project due
- **Dec ?**: Final exam (date TBA)

A high-level overview

Pick a topic you find interesting, where there may be some quantitative data to analyze. Think hobbies, past classes you've found interesting, something you really care about, etc. It will be a lot easier to dedicate time to the project if you find the topic fundamentally interesting.

To get started, review at the curriculum we've covered over the semester; it has been designed to give you a sense of a common **data science workflow**. Apply that workflow to your project:

1. **Collect data.** Find (good) data (which will depend on the research questions you end up formulating). This may take some time, and you may not find exactly the data you want in one file or in one sitting. Your interest may also shift, the more you search and realize what kind of data is available within the topic. Be willing to keep looking for (additional) data and iterating on your topic!
2. **Explore your data.** Start with the most basic types of analyses (summary statistics, histograms, scatterplots) to get a sense of the data. What *could* the data tell you? What kind of questions would it fail to answer?
3. **Write down a few concrete research questions and hypotheses**. What have you noticed in exploring your data? What do you know about the processes that generated the data? Discuss these ideas with your collaborators (if working with others), with friends in the course, and with course staff. Do you need to gather any additional data? Slightly different data?

4. **Select your tools.** Which analyses would be most appropriate for the type of data you've gathered and the questions you'd like to ask? What types of investigations has this class prepared you to conduct?
5. **Build models, analyze them, and test your hypotheses.** Don't throw out analyses that fail to show significance; these can still teach you something about the context from which your data came, or the data itself, if interpreted properly.
6. **Interpret your results.** This is *so* important. Your results don't matter unless you can make people understand why they matter. What do these results mean for the "real world" beyond your dataset? Were you limited in your conclusions because of issues with the data? What were the issues and how did they specifically impact your analyses?

Rinse and repeat. Doesn't that look like a nice sequence of tasks? Unfortunately, it never works exactly that way! You will constantly go back and forth between steps. Instead of imagining the steps as 1 -> 2 -> … -> 5, think of them as 1 <-> 2 <-> … <-> 5. Keep trying things until you feel like you've put together an interesting story for your final report.

Prepare a focused final report. Pick the clearest and most interesting analyses and contextualize them in your final report. Don't just present numbers and plots: explain to the reader what they mean. Answer the question "so what?". Put additional analyses tangential to the final direction of your project in (optional) appendices.

Working in groups

You may work in teams of two to four students. Expectations for larger group teams will be higher than for smaller teams.

Groups will self-assign on Gradescope when submitting phase 0 of the project, and it can be updated when submitting phases I and II. Groups will be carried over in Gradescope for every subsequent phase. **You cannot join or leave a group after the phase II deadline.**

To make it easier to find prospective teammates, we will provide a short survey identifying study and communication habits. This system will provide suggestions, but is not binding. You will still need to communicate with team members and enter groups in Gradescope.

Whether or not you are working in a group or individually, we strongly recommend you use version control software to organize your project work, such as Github.

Barring exceptional cases, all members of a group will receive the same grade. Group work occasionally leads to disagreements about the level of effort contributed by individual group members. In some circumstances course staff may use logs of Github commits to provide some perspective, and in extremely rare cases we may differentiate grades. If you choose not to use version control software, or use one that does not make user history available, we will not be able to make this type of consideration for your group.

Grading

Unlike homework or exams, there are no predefined answers for the project. When we have specific answers in mind we use *subtractive* grading, where we start with full marks and take off points for errors or missing elements. For open-ended projects we use *additive* grading, where we start with a baseline score and add points for elements that you do well. Your grade will depend on two factors: how ambitious you are (degree of difficulty) and how well you accomplish your goals (execution). Do not ask us "what did I lose points for?", rather ask "what could I have gained more points for?"

Two graders will independently evaluate each project. If there is a substantial difference, a third grader will be added. In practice, when the TA staff looks at a large number of projects there is a strong consensus about which projects are more impressive. When TAs separately write down numeric grades and then compare, they are usually all within a few points of each other.

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

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.

Rubrics

These rubrics will be used by your project mentor and peer reviewers to give you helpful feedback on the current state of your project. Use them to self-assess ahead of submission; figure out which aspects of your work are sufficiently advanced, and which aspects could use more work, then spend time on the latter.

Remember: our priority in grading every preliminary phase of the project (before the final submission) is to give you good feedback that helps you continue to develop your project. We want to help you produce a final project that you're proud of!

Phase II rubric:

Category	Less developed projects	Typical projects	More developed projects
Research question(s)	Question is not clearly stated or significantly limits potential analysis.	Clearly states the research question(s), which have moderate potential for interesting analyses.	Clearly states complex research question(s) that leads to significant potential for interesting analyses.
Data cleaning	Data is minimally cleaned, with little documentation and description of the steps undertaken.	Completes all necessary data cleaning for subsequent analyses. Describes cleaning steps with some detail.	Completes all necessary data cleaning for subsequent analyses. Describes all cleaning steps in full detail, so that the reader has an excellent grasp of how the raw data was transformed into the analysis-ready dataset.
Data description	Simple description of some aspects of the dataset, little consideration for sources. The description is missing answers to applicable questions detailed in the "Datasheets for Datasets" paper.	Answers all relevant questions in the "Datasheets for Datasets" paper.	All expectations of typical projects + credits and values data sources.
Data limitations	The limitations are not explained in depth. There is no mention of how these limitations may affect the meaning of results.	Identifies potential harms and data gaps, and describes how these could affect the meaning of results.	Creatively identifies potential harms and data gaps, and describes how these could affect the meaning of results, and the impact of results on people. It is evident that significant thought has been put into the limitations of the collected data.
Exploratory data	Motivation for choice of analysis methods is	Sufficient plots and summary statistics to	All expectations of typical projects +

analysis	unclear. Does not justify decisions to either confirm / update research questions and data description.	identify typical values in single variables and connections between pairs of variables. Uses exploratory analysis to confirm/update research questions and data description.	analysis methods are carefully chosen to identify important characteristics of data.

Phase III rubric:

Category	**Less developed projects**	**Typical projects**	**More developed projects**
Preregistration statement	The preregistered analyses could be performed using the data collected, but it is not clear how they fit in the context of the real-world application from which the data originated.	The preregistered analyses are contextualized by the real-world application to a certain degree. The analyses are not described in a way that persuades the reader their results would be interesting, whether or not they turn out to be statistically significant.	The preregistered analyses reflect deep and critical thinking about the real-world application from which the data originates. The analyses are described in a way that persuades the reader that their results will be interesting, whether or not they turn out to be statistically significant.

Phase IV rubric (also applies to the final submission as Phase IV is essentially a dry run for the final project report):

Category	**Less developed projects**	**Typical projects**	**More developed projects**
Introduction	Less focused and organized. They may jump to technical details without explaining why results are important. Research questions are	Provides background information and context. Introduces key terms and data sources. Outlines research	All expectations of typical projects + clearly describes why the setting is important and what is at stake in the results of the analysis. Even if the reader doesn't know

	not clearly stated and/or results are not clearly summarized at the end of the introduction.	question(s). Ends with a brief summary of findings.	much about the subject, they know why they care about the results of your analysis.
Data description	Simple description of some aspects of the dataset, little consideration for sources. The description is missing answers to applicable questions detailed in the "Datasheets for Datasets" paper.	Answers all relevant questions in the "Datasheets for Datasets" paper.	All expectations of typical projects + credits and values data sources.
Preregistration statement	The preregistered analyses could be performed using the data collected, but it is not clear how they fit in the context of the real-world application from which the data originated.	The preregistered analyses are contextualized by the real-world application to a certain degree. The analyses are not described in a way that persuades the reader their results would be interesting, whether or not they turn out to be statistically significant.	The preregistered analyses reflect deep and critical thinking about the real-world application from which the data originates. The analyses are described in a way that persuades the reader that their results will be interesting, whether or not they turn out to be statistically significant.
Data analysis	Code closely matches examples from class, and does not go much further. Analyses selected are not clearly purposeful. Preregistered analyses are not presented.	Code goes further than the examples presented in class. Analyses selected are purposeful and further the data narrative, but questions raised are not adequately addressed. Preregistered analyses are presented.	All expectations of typical projects + analyses are carefully selected to answer all reasonable questions. Questions raised by one analysis are addressed in subsequent analyses.
Evaluation of	Metrics of statistical	Metrics of statistical	Metrics of statistical

significance	significance are present, but not interpreted for the reader and/or relevant to the analysis performed.	significance appropriate to the analysis performed are presented and are interpreted to some degree for the reader.	significance appropriate to the analysis performed are presented and clearly interpreted for the reader. Limitations of significance metrics are acknowledged.
Interpretation and conclusions	Results are presented as numeric values and plots, with little to no written discussion. Values are printed out of context, with no/few labels.	Values are interpreted in a way that is clear and addresses what the values mean and explain to some extent why they are important. Values are printed with clear labels.	Interprets numeric values in a way that supports a clear story and conclusion creatively ties analysis together to present the results of the analysis through a well-written discussion. Values are presented in context and with clear labels.
Limitations	The limitations are not explained in depth. There is no mention of how these limitations may affect the meaning of results.	Identifies potential harms and data gaps, and describes how these could affect the meaning of results.	Creatively identifies potential harms and data gaps, and describes how these could affect the meaning of results, as well as the impact of results on people.
Writing	May have spelling and grammatical errors, or awkward or incomplete sentences, indicating that they were written in haste without editing.	Language will be polished and free from errors *(Note: if your group does not include a native English speaker, make a note of that in your submission)*.	Writing is clear and complicated ideas are presented such that they are immediately understandable.
Organization and focus	Work appears to have been done independently by team members and then merged at the last moment. Analyses may be exhaustive but carry	Most elements of the project are clear and provide a connected conclusion. Some parts could have been removed to make the report more focused. There is a clear story	All elements of the project support a clear and connected conclusion. Every part is essential and cohesive. There is a clear story to the entire report that

	little meaning or interpretation. There is not a very clear story throughout the entire report.	that flows throughout most of the report.	flows throughout.

Peer review

One of the common concerns students have about open-ended projects is that it is not clear what you should do, or how it will be graded. In order to give you more feedback and to get a better sense of how projects are evaluated we will be using *peer review*. During Friday sections each student will provide feedback for a small number of projects submitted by other students. We will provide the exact rubrics that graders will use to evaluate your projects.

Dataset ideas

- Books read and their ratings on Goodreads (API here)
- Sports statistics available on sportsreference.com
- Temperature readings for an area in a year from the National Weather Service
- Text from books across several genres from Project Gutenberg
- Reddit posts and/or comments (API here) (historical data here)
- Cornell's Convokit (conversational text data) here
- FiveThirtyEight's datasets (here)

Deliverables

Please prepare the following materials for your **final** project submission:

- A **dataset** of moderate size and complexity.
    - It should be large enough to have interesting complexity, but not so big as to be unwieldy. As a rough guideline, your dataset should be longer than you could print on a single page in standard spreadsheet format, but smaller than 20 MB.
    - You may use existing datasets, combine data from APIs, or create entirely new data through instruments or surveys.
    - The dataset you turn in does *not* have to be the dataset that you initially collected. For example, you might download 50 MB of raw logs, but use filtering and aggregation to reduce the dataset to 100 kB for your actual analysis. We want you to submit your *analysis-ready data*, but you should describe your full data-collection protocol and any preprocessing done in the data description section of your final report (see below). All source code use for data collection and preprocessing should also be linked to in the source code section of your final report.

    - If your final, curated dataset is larger than 10MB, share a copy in Cornell Box and include a link to it in your final report.
- A **final report**, as a Jupyter notebook with executed cells, containing the following sections:
    - **Introduction**. What is the context of the work? What research question are you trying to answer? What are your main findings? Include a brief summary of your results.
    - **Data description.** This should be inspired by the format presented in Gebru et al, 2018. Answer any relevant questions from sections 3.1-3.5 of the Gebru et al article, especially the following questions:
        - What are the observations (rows) and the attributes (columns)?
        - Why was this dataset created?
        - Who funded the creation of the dataset?
        - What processes might have influenced what data was observed and recorded and what was not?
        - What preprocessing was done, and how did the data come to be in the form that you are using?
        - If people are involved, were they aware of the data collection and if so, what purpose did they expect the data to be used for?
        - Where can your raw source data be found, if applicable? Provide a link to the raw data (hosted on Github, in a Cornell Google Drive or Cornell Box).
    - **Preregistration statement.** List the two analyses you promised to perform in this final report from your Phase III submission.
    - **Data analysis.**
        - Use summary functions like mean and standard deviation along with visual displays like scatterplots and histograms to describe data.
        - Provide at least one model showing patterns or relationships between variables that addresses your research question. This could be a regression or clustering, or something else that measures some property of the dataset.
    - **Evaluation of significance.** Use hypothesis tests, simulation, randomization, or any other techniques we have learned to compare the patterns you observe in the dataset to simple randomness.
    - **Interpretation and conclusions.** What did you find over the course of your data analysis, and how confident are you in these conclusions? Detail your results more so than in the introduction, now that the reader is familiar with your methods and analysis. Interpret these results in the wider context of the real-life application from where your data hails.
    - **Limitations.** What are the limitations of your study? What are the biases in your data or assumptions of your analyses that specifically affect the conclusions you're able to draw?
    - **Source code**. Provide a link to your Github repository (or other file hosting site) that has all of your project code (if applicable). For example, you might include web scraping code or data filtering and aggregation code.

    - **Acknowledgments**. Recognize any people or online resources that you found helpful. These can be tutorials, software packages, Stack Overflow questions, peers, and data sources. Showing gratitude is a great way to feel happier! But it also has the nice side-effect of reassuring us that you're not passing off someone else's work as your own. Crossover with other courses is permitted and encouraged, but it must be clearly stated, and it must be obvious what parts were and were not done for 2950. Copying without attribution robs you of the chance to learn, and wastes our time investigating.
- **Appendix: Data cleaning description.** Submit an updated version of your data cleaning description from phase II that describes all data cleaning steps performed on your raw data to turn it into the analysis-read dataset submitted with your final project. The data cleaning description should be a separate Jupyter notebook with executed cells, and it should output the dataset you submit as part of your project (e.g. written as a .csv file).
- (Optional) **Other appendices.** You will almost certainly feel that you have done a lot of work that didn't end up in the final report. We want you to edit and focus, but we also want to make sure that there's a place for work that didn't work out or that didn't fit in the final presentation. You may include any analyses you tried but were tangential to the final direction of your main report. Graders may briefly look at these appendices, but they also may not. You want to make your final report interesting enough that the graders don't feel the need to look at other things you tried. "Interesting" doesn't necessarily mean that the results in your final report were all statistically significant; it could be that your results were not significant but you were able to interpret them in an interesting and informed way.

Deadlines

- **Phase 0:** *Coordination and planning.* Due **Sep 7**.
    - Fill out the work contract template included in the assignment. Managing your time and making regular, consistent progress is the biggest factor that makes projects successful.
    - **Submit a .pdf file** on Gradescope stating your method of communication (if applicable) and with your completed work contract.
- **Phase I:** *Brainstorming.* Due **Sep 21**.
    - Set up a Github repository for your project files (or using the file-sharing service of your choice).
    - Meet with your group if you are collaborating. Write down at least three ideas for datasets. Include as much information as possible about the availability of data.
    - **Submit a .pdf file** on Gradescope that contains a link to your Github repository and your dataset ideas. Include a "**Questions for reviewers**" section at the end of your submission, listing specific questions for your project mentor to answer in giving you feedback on this phase.
- **Phase II:** *Data collection and exploratory data analysis.* Due **Oct 19**.
    - Settle on a single idea and state your research question(s) clearly.
    - Carry out most of your data collection and cleaning.

    - Compute some relevant summary statistics, and show some plots of your data, as applicable to your research question(s). Use this exploratory data analysis to:
        - update your research question(s), if applicable;
        - update your data description, if applicable (e.g. if you collect additional data).
    - **Submit an executed Jupyter notebook (.ipynb) file & corresponding converted .pdf file** on Gradescope with the following sections:
        - **Research question(s).** State your research question (s) clearly.
        - **Data collection and cleaning.** Have an initial draft of your data cleaning appendix. Document every step that takes your raw data file(s) and turns it into the analysis-ready data set that you would submit with your final project. Include text cells describing your data collection (downloading, scraping, surveys, etc), and text cells describing any additional data curation/cleaning (merging data frames, filtering, transformations of variables, etc). Include code cells for data curation/cleaning, but not collection. Note: You should be saving data in intermediate files at several points through this process so that you are not starting from scratch every time you change something. This also makes sure you have documentation for everything that you have done for your reference, but we do not necessarily need to see all of it in code form.
        - **Data description.** Have an initial draft of your data description section. Your data description should be about your analysis-ready data.
        - **Data limitations.** Identify any potential problems with your dataset.
        - **Exploratory data analysis.** Perform an (initial) exploratory data analysis.
        - **Questions for reviewers.** List specific questions for your peer reviewers and project mentor to answer in giving you feedback on this phase.
- **Phase III:** *Preregistration of analyses.* Due **Nov 2**.
    - Preregister at least two analyses that you vow to present in your final report, no matter the results (significant or not).
        - *Do not perform these analyses, check they are significant, and then "preregister" them.* That defeats the purpose. You can learn a lot from associations that are not statistically significant!
        - Keep in mind that for your final report, you must provide at least one model showing patterns or relationships between variables that addresses your research question.
    - **Submit a .pdf file** on Gradescope with your preregistration. Include a "**Questions for reviewers**" section at the end of your submission, listing specific questions for your project mentor to answer in giving you feedback on this phase.
- **Phase IV**: *Draft of final results.* Due **Nov 16**.
    - **Submit an executed Jupyter notebook (.ipynb) file & corresponding converted .pdf file** on Gradescope, with all of the required elements, as detailed in the deliverables section above. Include a "**Questions for reviewers**" section at the end of your submission, listing specific questions for your peer reviewers and project mentor to answer in giving you feedback on this phase.

    - We often find that the moment we "finish" a project is also the time when we have the most ideas about how to continue it. The goal of this phase is to create a version of your project that *could* be complete, but with enough time remaining that you can revisit your analysis, fill in gaps, and continue logical extensions.
    - You will provide peer review for other groups' submissions.
- **Phase V:** *Final results.* Due **Dec 4**.
    - **Submit an executed Jupyter notebook (.ipynb) file & corresponding converted .pdf file** on Gradescope, with all of the required elements, as detailed in the deliverables section above.

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.** The instructors have supervised hundreds of group projects over the years. Most of the problems we hear about with group work are either “Person X isn’t doing anything!” or “Person Y won’t let me do anything!” 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.
- **Don't forget to analyze and reflect**. The strongest projects often differentiate themselves through their interpretation. Less advanced projects often stop with presentation of quantitative results and don't tell us what these results mean, why they matter, or what the limitations are.

Writing help

If you could use more support in your writing, the Cornell Writing Centers are a free resource available to students looking to improve their writing. They can help you with specific assignments (like the final project phases!) and they offer their services remotely.

Tips for producing an advanced final project

The formal requirements for the project are listed in the deliverables section, with more detailed guidance in the rubrics. This section gives tips for producing an advanced final project.

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). We expect the total length should be 1500-3000 words. Inside this range, length will not be a factor in grading.

Introduction: 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 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 “origin story” 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) in this article.

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 “X is larger than Y” write “X is 3.65 times larger than Y”). 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.

It’s important to organize your analysis. Common organizational patterns:

- 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 “zoom out” 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.

Interpretation and conclusions: This section 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.

Limitations: "Data" is a selective view of the world that may have been produced in any number of limited, skewed, or biased ways. "Models" are exactly that: miniature representations of real processes that capture some essential relationships but eliminate everything else. Identifying the limitations of your work is a critical part of the data science process.

Code: As notebooks with evaluated cells. We won't 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.

Grading scheme

Grades will be calculated as follows.

- 30% Homework. All homework assignments will count equally, regardless of their individual grading. For convenience in grading, we may grade one homework out of a total of 35 points and another out of 108, but both will count equally.
- 4% Project Phases 0, I, III, IV (low-stakes, graded for completion)
- 11% Project Phase II (data collection and exploration)
- 25% Project Phase V (final project)
- 10% Midterm exam
- 15% Final exam
- 5% Professionalism (attendance is mandatory in lectures and discussion sessions; we will take attendance).

The SONA extra credit system will remain in place (1 credit = 0.5 course points up to a maximum of 2 points), but it is not clear whether there will be any studies for you to participate in. If it appears that SONA is functionally unavailable, we will offer additional extra credit opportunities.

Github Tips

Github

Whether or not you work with others on the final project, we encourage you to set your project up in Github, a tool many in both academia and industry use to keep projects synchronized between different people and/or devices. Think of this project as an opportunity to learn the valuable practical skill of how to use Github!

Git is a version control system that allows multiple users to work together on documents. A git *repository* (repo) is a shared canonical version of a directory structure. Users *clone* the repo to create their own working copies. From a cloned repo, you can then *pull* changes made by other users. If you want to share your changes, you *commit* a new version of a file and *push* committed changes to the repo. Git detects which changes have been made to documents and when those changes were made. It works best with text-based documents, like Python scripts (but can also work for Jupyter notebooks, which are actually secretly .JSON files). Git can then aggregate and merge various changes to the same document, usually in an automatic (painless!) way. Git isn't inherently collaborative (at its most basic, it's for keeping one person's files organized on their own computer).

Github is an online platform for hosting git repos. Github also has useful interfaces for exploring repos and visualizing changes to them, which extends its functionality to enable collaboration across different users and/or devices.

Getting started

To get started with Github for the final project:

1. **Create a Github account**, if you don't already have an existing one.
2. **Download Github Desktop** here (unless you are comfortable working with git in the command line, in which case, go head, but we will not necessarily be able to support you).
    - Github Desktop is a Graphical User Interface for using git on your local machine and connecting it to a central Github repository in the cloud.
3. Set up Github Desktop with your Github account like this.
4. **One member of your group** first needs to create a central Github repository (often referred to in the documentation as "the remote (repository)" or "the origin"), which is hosted in the cloud by Github. Please follow these instructions and make your central/remote repository. Please make your repository public.
5. The person that created the repository should add group members as collaborators so that everyone can contribute to the repository.
6. Once everyone is a collaborator on the project repository, everyone should clone it to their computer. This creates a local copy of the repository as a folder on your machine that you can add to/edit the files within.

Working on the project

- After completing the above setup, you will be able to work in your project repository folder as you would normally work in a folder on your computer.
- All *saved* changes to files in that folder (repository) get **tracked in the background** by git, but changes are not visible to your collaborators until you send them to the remote/central repo in the Github cloud.
- When you save changes to your local files, they will show up in the **Changes** pane of the Github Desktop window (on the left). You will need to **commit** those changes locally and then **push** them to the central repository in the cloud for your collaborators to see them. Your other group members then need to **sync** their repo with the (now updated) remote one to retrieve your changes. Instructions for committing/pushing are here and for syncing here.
    - The documentation talks about different branches for your project, but for simplicity, just ignore this and remain in the default ("master") branch.
- When committing changes, try to write a **descriptive commit message**, so that your group members (and/or you in the future), know what has changed with the files in that specific commitment of changes.
    - Use the commit messages to keep track of what you've done and to keep yourself organized.
    - Try only to commit changes with **working code**.
    - It is good practice to commit **somewhat frequently**, after small changes (think every time you write a small chunk of working code, not every time you save your file).
    - You may also want to commit changes when you are getting ready to wrap up your work session on the project and move on to something else.

        - It can be helpful to write a note to yourself in the commit message that describes the **next thing** you want to do when you get back to the project.
- You can see a **history** of all commits (including those of your collaborators) in this History pane, so you know what everyone has been working on (commits are listed chronologically here, with the most recent at the top).
- You cannot push a file larger than **100 MB** to Github, so be wary of trying to push large data files. Instead, you may want to tell Github to ignore these files.
- When working on a Jupyter notebook with other people, if you have different versions of Python on your systems, Github will detect this change in your notebook every time you alternate work on it. This is not an issue, just something to be aware of. You can always discard this change (if it's the only change to the file, lest you lose real changes!).

Merge conflicts

Git is pretty good about merging changes to the same files automatically, but sometimes there are **merge conflicts** it needs you to resolve manually. These can be quite the headache! To avoid merge conflicts, it helps if you and your collaborator are not trying to work on the same file at the same time, or are at least sticking to different sections of the same file. If you do run into a merge conflict, try getting help here.

There is also the nuclear option, which discards all of your local changes which have not been sent to the Github cloud, and resets the files to the latest version in the remote repo. This can occasionally be helpful if you're in too deep with merge conflicts. As of this writing, this cannot be accomplished through Github Desktop and instead must be done via the command line. Here are some instructions.