IS597PR Final Projects – Fall 2020

Due Dates: ~Oct 30th=proposal in Moodle forum; Draft/review stage materials in GitHub by Dec 2; Presentations on Dec 2nd and 9th in class or pre-recorded; Peer voting & feedback on projects open Dec 2nd through Dec 10th ; Dec. 13th=All finalized files in GitHub for grading.

Overview and High-Level Requirements:

Many previous students have done nice work, of which they and I were proud. In a few cases, their projects and this course directly helped them get a job. You have an even wider range of choices available than most earlier semesters, so just avoid the pitfalls, apply yourselves the best you can and try to have some fun with it during this stressful time.

1. You may choose to *work as teams of 1, 2, or 3 students*. Multi-member teams must clearly show collaboration from every member and are expected to perform at a proportionally-increased level of complexity, sophistication, depth, and/or scope to earn a high grade. Projects will be evaluated and graded as a teamwork product. You must work together, and hold each other accountable for contributions, quality, and ethical behavior. If there's a problem early on (such as someone insists on plagiarizing, notify the instructor immediately).
2. Select from one of these TYPES of analytics projects to implement in Python:
    I. Your original Monte Carlo simulation
    II. Your original analysis linking 2+ published data sets from distinct sources, investigating some topic like: societal or environmental changes possibly affected by changes in laws, industry, new inventions, or corporate practices; or analysis of complex biases within data; or testing scientific hypotheses; or uncovering evidence of corruption in any industry, company, or government; historical changes of health, economic, or other aggregate life quality factors between countries, cities, or similar.
    III. A formal critique of weaknesses, mistakes, or scope limitations in someone else's published data analysis and your code rewrite/enhancement of the program to improve the rigor of the analysis AND make the code more reliable & maintainable.
3. Do not plan a statistical Machine Learning-focused project. Enroll in Machine Learning Team Projects if that interests you. Thus, packages such as sklearn, PyTorch and TensorFlow and similar should not be used here.
4. You must submit your original unique work, created specifically for 597PR. If the project is related to work you did earlier or are now doing for any other course or a job, then you must get prior written approval from all the relevant instructors and the supervisor. Not doing so is subject to sanctions per the Student Code.
5. PROPOSAL Stage. Post your summary into Moodle's **"Final Project Topic Proposals"** forum. See expected information there.
6. PROPERLY CITE ALL YOUR SOURCES! Any citation style is fine, but make sure you do it and do so continuously, not just at the very end. Students who have used code or other material without indicating its true source AND clearly delineating which parts are not original will be reported to iSchool & UI Grad College through FAIR. If the review confirms plagiarism, the sanction is reduced grade (dependent on scope), with a course grade of F for significant cases, regardless of your previous scores.

7. Unlike all other assignments in 597PR, you are allowed and expected to openly publish your unique project work. You may consider it part of your student portfolio, link it from resume, etc. Typically, this is done on Github.com since you'll be committing there as work proceeds. Make sure everything you put there is work you'll be proud of. At every commit, you should be verifying citations for all code, data, etc. Remember every commit is a "version" of the project and is public. It is expected to be a FORK from here: https://github.com/iSchool-597PR/2020Fall_projects
8. Draft Stage & Peer Reviews: The program and documentation should be sufficiently operational for meaningful and beneficial peer code review, but does not have to be 100% finished. Make sure your GitHub repository is up to date with all the work you've done so far (code, documentation, example outputs). It's okay if there are some final scenario explorations or even minor flaws left to resolve in your project at this stage, but you want constructive feedback from others. All students will also be submitting evaluations about other teams' projects, details will be given in class.
9. Presentations: You will create and deliver a presentation to the class (either live in Zoom or as pre-recorded video in Illinois MediaSpace) that summarizes your project's purpose, hypotheses, design reasoning, and results so far, possibly with a quick demo.

10. Final submission expectations:
    - □ Edit your README.md to create a good introduction and overview of the project, written with new visitors to your repository in mind. Summarize the conclusions you came up with, including how results are either supportive of or refute your hypotheses. You can embed images into the README file, if that is relevant.
    - □ Use the "Factors in Code Quality and Code Reviews" like a checklist. Apply as many of the skills we've discussed this semester as applicable, to create the best quality program you can. Example expectations:
        i. Doctests and/or pyunit tests: 1-person team minimum 30% actual test coverage; 2-person minimum 50% actual test coverage; (3-person minimum 80% actual coverage AND use Travis-CI or GitHub Actions to automate the test suite during your development, not just after it's complete).
        ii. All functions (methods included) need complete Docstrings. 1-person=minimum 4 functions; 2-person=7+ functions; 3-person= 10+ functions
        iii. 3-person teams also should incorporate one of these efficiency techniques that will be discussed in class: Selective compilation (e.g. Numba or Cython); and/or parallel processing.
    - □ Consider each hypothesis or alternative situation you proposed to investigate (you may have added more after feedback). Your program code should be able to run the simulation for all of the hypotheses just by changing top-level parameter values and/or through using different input data files -- do not hard-code such configuration aspects into the functions themselves.

(Type I projects) Specifics for Monte Carlo Simulations:

- □ Design your own scenario -- make certain your simulation is original in some way(s).
- □ You can simulate an engineering or manufacturing problem, business/management situation, (certain types of) human behaviors, physical phenomena, or a game. To encourage original thinking, **AVOID scenarios that have been done many times and/or**

discussed in class, such as : a "random walk" of stock prices, stock options, or similarly naïve financial “predictions”; a traffic simulation with just a few intersections or only one road; parking lots or parking meters; customer seating/dining at a restaurant or serving them at a counter; the games *Tic-tac-toe*, *four*-in-a-row*, *Go-moku*, *chutes and ladders*, *Monopoly*, *Rock-Paper-Scissors, Blackjack, Poker*. If you want to model a sports game or tournament, ask first, too many of these have been done already. It is okay to build one on communicable disease pandemics if you wish, just make sure you extend it beyond the class discussion or explore it in a different way.

□ You must have several well-chosen random variables in the model, to explore a variety of possible outcomes and derive the non-obvious probabilities of the overall model. Make sure you think carefully to choose appropriate ranges of values and a sensible distribution type for every randomized aspect. For example, if you simulate “number of swimmers in the pool” at each point in time as a uniform distribution, it’s wrong. If you simulate the individual finish times of all runners in a marathon as uniform, triangular, or even normal, it’s wrong. We’ll discuss this in class.

i. 1-person teams must have at least 2 different kinds of randomized variables plus the deterministic aspects and control variables in the model. Most interesting simulations require more.
ii. 2-person teams must have at least 3 different kinds …
iii. 3-person teams must have at least 4 different kinds …

□ Controls & Experiments: 1-person teams must have at least 1 hypothesis to test by running an experiment using the simulation in different modes to compare the results. Multi-person teams are expected to go further, typically testing multiple hypotheses with corresponding experiments, but variations can have other manifestations.

□ If there is any relevant public data available, try to incorporate real data as part of your simulation model. For example, any useful sports simulation must use some real performance statistics about players and/or teams, so its randomized variables can be sampled from realistic ranges & distributions. If data you seek is not in downloadable form, you can still research the scenario to make realistic estimates.

(Type II Projects) Specifics for an Original Data Analysis [Non-simulation]:

An **original** analysis linking 2+ published data sets from distinct sources, investigating some topic like: societal or environmental changes over time (possibly) affected by changes in laws, industry, new inventions, or corporate practices; or analysis of complex biases within data; or testing scientific hypotheses; or showing evidence of corruption in any industry, company, or government; historical changes of health, economic, or other aggregate life quality factors between countries, cities, or similar.

There are *thousands* of public data sets that could be of interest to you, from many US and foreign government agencies, scientific organizations, universities, companies, and more. Look at data.gov, worldbank.org, or similar big repositories where you can search for open data that interests you.

Some earlier assignments were similar to this. This should be larger in scope, sophistication, or code quality than regular assignments you did in 597PR, especially for multi-student teams.

Warning: many previous plagiarism cases in this course have come from students copying chunks of analysis and program code from one or more Kaggle.com “kernels” or similar data analysis found on GitHub and presenting it as their own work. If you do the same, from ANY uncredited source, you will likely fail this course. If something on Kaggle strongly interests you, consider a Type III project on it instead. Most of the kernels on Kaggle are low-quality code and many have weak or even pointless analysis, which leaves a lot of room for you to improve them.

(Type III Projects) Specifics for a Formal Critique and Improvement of Weaknesses or Limitations in a Published Data Analysis:

This type of project is very flexible, but might require more effort during the Proposal stage, to make certain that the previously-published analysis you choose has sufficient weaknesses in the code, data, or conclusions to justify reworking it. The original publication you critique and rework does not have to be a scholarly peer-reviewed analysis (you’d have a hard time identifying a good candidate in the short time we have). So, it could be a less formal analysis that was published on an open website (e.g. a blog, GitHub.com, Kaggle.com, etc.).

I don’t want to see a project where you essentially just convert an already-good analysis from another language into Python. The original might even be in Python already, but it needs to be flawed, fragile, or incomplete in its code, statistical analysis, and/or conclusions. That could have happened because they used biased or insufficient data (that you will improve, augment, or work around), made logical mistakes during analysis, or based conclusions on incorrect programming. Another more subtle possibility is that they improperly ignored uncertainties within the raw or processed data.

Most of the kernels on Kaggle are low-quality code and many have weak or even pointless analysis, both of which leave a lot of room for you to improve them. But the popular topics and data sets on Kaggle can have hundreds of “kernels” posted by different people using the same data. That density doesn’t leave much room for improvement without danger of plagiarism. So, whether it’s on Kaggle or elsewhere, find a data analysis topic to improve that does NOT already have dozens or hundreds of people’s versions and commentary on them.