# Assignment 4: Final Project (35%)

**Due:** April 1

For the final project, you will complete an end-to-end analysis to using real data. The project consists of the following steps:

- Identify the analytic question(s) that you would like to answer.
- Identify and obtain the relevant data.
- Clean and transform the data into a usable format.
- Create appropriate visualizations of the data.
- Perform an analytic technique such as clustering, classification, or regression, with corresponding visualizations.

## Instructions

Submit a Jupyter notebook or Python script through D2L Dropbox. Your submission should include the following features:

1. A written introduction to your project that introduces the analytic question(s) and the data sources. **Data for this project must come from at least 2 different sources.**

2. Include the Python code for cleaning and transforming the data. This includes removing duplicates, changing column data types, replacing values, renaming columns/indices, and filtering missing values and/or removing outliers.

3. Summarize your data with visualizations for the appropriate data types. **Your project must include a minimum of 3 visualizations.** Examples of visualizations are histograms, bar charts, heatmaps, line graphs, etc.

4. Perform some analytic method on the data. Report the result and generate a corresponding visualization (plot the result).

5. Where necessary, use of external packages not included in the course materials will be allowed on a per-case basis. If you want to use an external package we haven't covered, contact me to discuss.

Please refer to the rubric at the end of this document for evaluation details.

## Requirements

a. Working Jupyter notebook or Python script.

b. Make sure to include clear comments and docstrings in your code.

## Submission

Submit your assignment through D2L Dropbox on Apr 1.

## Evaluation

This assignment is graded out of 16 points using the following rubric and is worth 35% of the final grade.

Learners may receive partial scores or a zero for unacceptable work.

| Criteria | Does Not Meet Expectations 1 | Partially Meets Expectations 2 | Meets Expectations 3 | Exceeds Expectations 4 | Max Points |
|---|---|---|---|---|---|
| **Code Functionality** | Inadequate comprehension of concepts and conventions presented in course materials. Syntax, runtime, and semantic errors throughout. | Partial comprehension of concepts and conventions presented in course materials. Some runtime and semantic errors and almost no syntax errors. | Adequate comprehension of concepts and conventions presented in course materials. No syntax or runtime errors and almost no semantic errors. | Broad and in-depth comprehension of concepts and conventions presented in course materials. No syntax, runtime, or semantic errors. | 4 |
| **Required Components** | Response is missing/does not address required components indicated in the instructions. | Response is missing/does not address some of the required components indicated in the instructions. | Response includes all components and meets the requirements indicated in the instructions. | Response includes and exceeds the requirements indicated in the instructions. | 4 |
| **Python-specific Code** | Offers brute-force solutions to problems. No functions, for/while loops, or iteration is used. | Brute-force solution with no built-in functions. Iteration used incorrectly or inconsistently. | Some iteration used to solve problems. Built-in functions are used incorrectly or inconsistently. | Well written code firmly using Python's particular style. Built-in functions are used when possible, and iteration is used correctly. Some list or dictionary comprehensions used where applicable. | 4 |
| **Writing Quality** | Unclear organization. Many grammatical, spelling, or punctuation errors. | Some signs of logical organization. A few grammar, spelling, or punctuation errors. | Organization supports purpose. Well-constructed body of code. Almost no grammatical, spelling, or punctuation errors. | Organization fully and imaginatively supports purpose. Well-constructed body of code. No grammatical, spelling or punctuation errors. | 4 |