Lab 2, Part 2, Data Preparation: Tabular data

Due date: Friday, April 9, 11:59pm

Please note: a new lab will be coming out Friday morning (with a short deadline as well), so it might be a good idea to have this one finished by 10am on Friday.

Lab Assignment

Assignment Preparation

This is an individual lab and a continuation for Lab 2.

The lab is very short. There are two tasks, one of which is to fix the issues with the Lab 2 program. The second task is new and asks you to create a different tabular representation of data.

The Data Collection: CSU Statistics

Both tasks will be performed using the same CSU dataset as in Lab 2.

Note: If you code works better with CSV files that do not contain an empty line as line 2 of the file, feel free to eliminate that line directly from the files you are using, and submit the files with your assignment. This is **explicitly allowed** for both Lab 2-2 and (retroactively) for Lab 2.

Task 1: Fixing Lab 2 code.

Our initial goal is reproduced below for the sake of completeness.

> *We are interested in comparing the trends in enrollment and graduation on each campus. We are also interested in comparing these trends to the trends in the total number of faculty teaching on each campus. Essentially, we want to find out whether*

having a lowe student to faculty ratio (i.e. more faculty per the same number of students) improves graduation. We specifically are interested in looking at four-year graduation numbers - that is, for each enrollment class, we need to look at the number of degrees that was grants four years later. We realize that this is not necessarily the best way to estimate four-year graduation rates, as the number of degrees conferred will have students who received their degrees in their fifth or sixth year as well, but lacking other data, we will use this data.

The instructions provided to you, however, were incorrect as far as building the proper dataset that meets these expectations. For this task we will fix the errors in the specifications, and you need to go ahead and change your Lab 2 implementation to build a correct (w.r.t. the user needs presented above) dataset.

The initial specifications ignored the following important observation:

- Academic year is not aligned with the calendar year.
- What this means is that the enrollment for a specific academic year X is represented in a record for year X, but the number of degrees granted in that academic year is represented in a record for year $X+1$.

To account for this, we need to change our specification as follows (new information in ***boldface italics***).

1. Identify all data files in the CSU data collection that are relevant to this information request. ***This part stays as it was. You would need to use exactly the same files as before.***.
2. Construct a single data table, stored as a CSV file, that contains all data necessary for the subsequent data analysis. In particular, the table you build shall contain the following information:
   - Each row of the table shall represent the known "fate" of a single incoming class at a single campus. Essentially, you want to report the total number of students enrolled in a specific year, followed by the total number of degrees that was granted (presumably) to the students who enrolled in this year four years later. Additionally, if this information is available, the number of faculty members employed on campus in the year of enrollment shall be reported as well[1] ***This is our true intent, and it stays intact, save for a comment about the faculty numbers I make below.***
   - Each row shall contain the campus Id (a number from 1 to 23) as well as *the full name of the campus*. ***This stays intact.***

[1] The faculty numbers are available only for three years, if this information is not available for a specfic year, simply do not include it in your data table.

- Each row shall contain the enrollment year. ***This stays intact.***
- Each row shall contain the information about the students enrolled in a given year, and an estimate of the number of freshmen coming in during that year. ***This is where we need to change our formula. We note that for year X, the number of degrees obtained the year before needs to be found in a record for year X, NOT in the record for year $X-1$. The new formula therefore is:***

  $$freshmen = totalEnrollment(year) - (totalEnrollment(year-1) - \mathbf{degrees(year)})$$

- Each row shall contain the information about the estimated number of degrees granted to the freshmen who enrolled that year. ***We need to change this spec as well. Students coming to university in year X will be "graduating seniors" at the end of academic year $X+3$, which means that the actual year is in fact $X+4$.***
  This number is computed as

  $$fourYearGrads = \min(freshmen(year), \mathbf{degrees(year+4)})$$

- The table shall contain only the rows where all values listed above can be computed[2] ***This stays. The actual years for which the data is available might change due to the change in formulas though, so prepare for your output to be different.***
- For years for which the faculty numbers are available, the table shall also include the information about the total number for faculty employed on campus that year. ***This is still technically true. It has been pointed out, however, that due to other requirements and a fairly short span during which we know the faculty numbers, no years may have both the degrees/enrollments information and the faculty information. This means that while I still want you to have a "faculty" column in the output, this column will be empty in the dataset you produce. Still, your code must work for any data provided, so if I were to add faculty data to the CSV file, your code should produce proper output.***

The rest of your specs remain the same. Name your CSV file `trends.csv` and submit a `README` file explaining which Python programs from your submission complete this task, and which — Task 2.

[2]The degree information is available for fewer years than the enrollment information.

Task 2: A different kind of matrix

The goal of this task is to create a (possibly sparse) matrix of student enrollments in different disciplines in 2004.

1. Read the full set of specifications for this task. Based on what you read, select the CSV files you need for this task.
2. You need to create a CSV file that stores, in a single row, the *undergraduate enrollment numbers* for each discipline in a given campus. Each row shall also store some other information as specified below.
3. Your output file shall contain exactly 24 lines. The first line shall be formatted similarly to the first lines of your input CSV files and contains the names of all the columns in the output file. Lines 2 through 24 shall contain information about the 23 campuses in year 2004, one line per each campus.
4. The first column of the CSV file shall be `campus Id`. The second column shall be `full name of campus`.
5. The next 22 columns in the output CSV shall contain the information about the *undergraduate* enrollment in the 22 different disciplines (one discipline per column, use the order specified in the `disciplines.csv` file).
6. Column 24 in the output CSV file shall contain the `total number of graduate students` on each campus in 2004.
7. Column 25 shall contain `total number of graduate and undergraduate students` as computed by adding up the values in columns 3 through 24.
8. Column 26 shall contain the `total enrollment numbers` for that campus from the total enrollments data for 2004.
9. Column 27 shall contain the difference between the numbers in columns 25 and 26.
10. **Handling of missing values.** You have two choices on how to handle missing values. Pick one and be consistent about it.

Option 1. **Zeros**. You can represent missing values with zeroes (0). The reason for this is straightforward. If no undergraduates are enrolled in this discipline on a given campus, then there are zero students enrolled.

Option 2. **Nothing**. You can represent missing values by putting nothing in the specific column. Your output shall contain two commas in a row in such places (with no whitespace between them).

Name your output file `breakdown2004.csv`.

Please note, your code must **not** have hardcoded information about the list of disciplines. This information needs to be prepared on the fly by reading the data from the list of discipline names. If the list were to contain more than 22 disciplines or fewer than 22 disciplines, the format of your output CSV file shall adjust automatically to accommodate extra or missing columns (i.e., rather than the 22 columns listed above, you'd include the correct number of columns dynamically).

Note on File I/O

Your Python programs shall assume that the CSV file names are static (i.e., you can hardcode them into the program when you need them). They shall also assume that the files are present in the current directory (i.e., in the directory in which the programs themselves reside). If you need to make changes in Task 1 code to accommodate for it - **do it**.

Submission

For this lab we will use CSL `handin` tool.

You shall submit your work as follows:

- Lab archive. Put all Python code, your `README` file and all other files you need to submit (except for the report) into a single archive. You can submit either `.zip` files or `.tar.gz` files. Name your file `lab02.zip` or `lab03.tar.gz`. Your `README` file shall specify which Python programs are used for which task, and who those programs shall be run. If you changed the format of the input CSV files, make sure to put them in the archive you are submitting.

Use `handin` to submit as follows:

```
$ handin dekhtyar lab03 <FILES>
```

Good Luck!