Lab 1, What is Data Science?

Due date: Monday, April 4, 10:00am

Lab Assignment

Assignment Preparation

This is a pair programming lab. You are responsible for finding your teammates for this assignment, and you need to do it fast. I will assign partners for everyone who has not been able to find one.

The Task

Data Science is often described as the process of finding insight in data. Most of our time this quarter will be spent looking at the different steps of the process and discussing the different types of data.

The purpose of this lab is to get you to do two things:

1. Remember Python. Most of software development in this course will use Python as the language of choice. We want to make sure that everyone has a good grasp on Python and its functionality, as we will not be spending much time in this class discussing various quirks of Python's syntax and semantics.
2. Think about data. The key difference between data science and any other type of work with data (data management, data mining, etc...) is that data science essentially requires an overarching process that includes understanding of the data, ability to ask questions about it, data manipulation and data cleaning, data analysis, and visualization and explanation of results. You need to start looking at this as a holistic process from the very first day of the class.

The general idea behind this lab is straightforward:

1. Pick a partner.
2. You are given a dataset. Study it.
3. Figure out what interesting nuggets of information this dataset may contain.
4. Formulate specific questions you propose to answer by analyzing the data in this dataset.
5. Manipulate the data given to you to put it in the form that makes answering the questions possible.
6. Analyze the data to answer your questions.
7. Find ways to visualize obtained answers.
8. Record all your work in a lab report.

Specific details for each step are given below.

The Dataset: Baby Names

About three months ago, a data science web portal Kaggle (`http://www.kaggle.com`) received a submitted dataset documenting the frequency of appearance of various baby names over the past 135 years.

CSV file format. CSV (comma-separated values) files are text files storing one or more records. Each record is stored on a single line of the file. Each record consists of values of multiple fields. Typically, each record in a CSV file has exactly the same fields, and the values of these fields are stored in the same order for each record. The values of fields are separated by a comma.

The dataset. You are provided with a number of "slices" of the original dataset. The Kaggle Baby Names dataset contains 1.8 million individual records. Each file provided to you contains a small portion of these records.

File format. Each file has records in exactly the same format. A single record specifies, given a name and a year, how many times the name was given to a baby that year. The full format of a record is:

```
RecordID, Name, Year, Gender, Count
```

Here:

`RecordId`:	Unique identifier of the record (line number in the original CSV file of the Kaggle dataset)
`Name`:	The exact spelling of the baby name
`Year`:	The year for which the data is provided
`Gender`:	"M" (Male) of "F" (Female)
`Count`:	Number of times the name was given to the babies of given gender in a given year

The first line of each CSV file provided to you contains the list of fields (it reads exactly as the line above).

A few sample lines from the CSV files look as follows:

```
1,Mary,1880,F,7065
75351,Harry,1906,M,1631
1796017,Wesley,2014,F,43
1811586,Wesley,2014,M,3112
```

The first line states that in 1880, 7065 newborn girls received name Mary. The second line specifies that in 1906, 1631 newborn boys received name Harry. The third line states in 2014 there were 43 newborn girls who received name Wesley. The fourth line shows that the number of boys who were named Wesley in the same year was 3112.

You are provided the following files.

names1000.csv file. `names1000.csv` contains a subset of records documenting all baby names that appeared with the frequency of 1000 or more during a specific year. This file contains about 50.5 thousand records.

names2014.csv file. This file contains information about all baby names used with frequency of 5 or higher in 2014. The file contains about 33 thousand records.

names-top1880.csv. This file contains information about the popularity of the top five most frequent 1880 male and female names (John, William, James, Charles, George, Mary, Anna, Elizabeth, Minnie,) throughout the history. Please note that I have not filtered by gender, so in years when there were men named Anna or women named John, those records will also be present. This file contains around 2600 records.

names-unisex.csv. This file contains information about the popularity, throughout the entire history recorded in the original dataset, of names that the Wikipedia article "Unisex name"[1] mentions as being English unisex names[2] The file contains 6971 records.

[1] https://en.wikipedia.org/wiki/Unisex_name

[2] I added the name "Dana" to the list.

names-BrZa.csv. This file contains my attempt to bring into a single file the information about the frequency of occurrences of all versions of the female name Brittany and the male name Zachary. Not every record in this file represents a variant of one of those two names (e.g., the file contains male names `Britt` and `Zahran` and female names `Zakhya` and `Brita`). The total number of records in this file is 5930.

For your study, you can you any one or more files provided to you.

Design

Your assignment has a design, implementation and analysis stages. On the design stage, perform the following tasks:

1. Study the dataset provided to you.
2. Come up with some questions that you believe are interesting and are worth asking of data.
3. Select the dataset file or files that would allow to answer each of the questions.
4. Determine if any data transformation/data cleaning needs to be performed in order for you to be able to answer the questions you posed.

Questions. You must ask **at least three different questions** of the data. You may ask more questions. Each additional successfully performed analytical task (answer to a question) will yield extra credit. When asking questions, please make sure you follow the following rules:

- Your question must be **interesting**. Simply asking "how many boys were named David in 1992?" is not an interesting question (neither is it a complex question). The answer to your question, if successfully obtained shall provide some insight into something that may be of interest to people.
- Your question must be **reasonably complex**. Answering the question shall require a certain amount of computational effort.

Data transformations. A **data transformation** for the purposes of this lab is *any manipulation of data that produces - either in main memory, or on disk (as a file) a representation of data with format that is different than the input data format.* Certain questions are best answered via fairly straightforward computations that occur over data that is represented differently than your input CSV files. It is *your responsibility* to figure out what sort of transformations are a good idea for your data analysis.

Data cleaning. A **data cleaning procedure** for the purposes of this lab is *any manipulation of input data that produces as a result a subset of data records, and, possibly, a subset of their attributes.* Certain questions require you to work only with specific records in your input data files. Your file may contain extra records. A simple data cleaning operation/procedure detects those records and removes them from consideration. More complex data cleaning procedures may also merge records that are deemed to contain information about the same object/entity.

Implementation

On the implementation stage of this assignment you will develop Python code for all operations and computations you need in order to answer the questions that you have posed.

You can develop the code in any way you deem necessary - either as a single collection of functions, or as multiple scripts, each conducting one computation on the data.

However you choose to do it, please make sure to thoroughly document what your code is doing. Each Python script must contain a header comment with your names and an explanation of the purpose of the script.

Additionally, you must create and maintain a `README` file, where for each question you ask in this lab, you document the exact process that you used to perform the computations answering the question. This will help during the grading process.

Analysis

Your goal on this stage of the assignment is to study the output of your computations, and determine what is the best way to describe your findings for someone else (an outside reader). You may need to determine how to show the results of your analysis so that they are easy to understand and interpret.

On this stage you can use whatever tools necessary (from Python to Microsoft Excel, to any other software you want) to present your data in a form you want.

Your findings shall be documented in a written report, whose structure and contents we discuss below.

Report

Each team must submit a lab report documenting the work done. The report shall be written in a form of an academic article (complete sentences, meaningful paragraphs, etc.) and shall contain the following parts:

- **Title, authorship.** The title of your report (including the fact that

it is a report for Lab 1 of this course) and the names of all students on the team shall be on the first page of the report.

- **Introduction.** A **short overview** of the work done in this lab.
- **Design.** Describe each of the questions you asked of the data, and the way in which proceeded to answer the question.
- **Implementation.** Briefly describe the implementation of your analysis for each of the questions.
- **Results and analysis.** Include the results you have obtained for each of the questions, and provide your own analysis of the observed results. Explain what you are seeing.
- **Conclusion.** Provide a brief summary of your completed work.

Submission

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

You shall submit two files:

- Lab archive. Put all Python code, `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 `lab01.zip` or `lab01.tar.gz`.
- Lab report. Submit your lab report as a separate file. Please submit a PDF version of your report. I encourage the use of LaTeXbut since most of you may be unfamiliar with it, you can use any text processor to write your report.

Use `handin` to submit as follows:

```
$ handin dekhtyar lab01 <FILES>
```

Good Luck!