CSCI 1100 — Critical Computer Science 1

Homework 7

Dictionaries & Cleaning Data

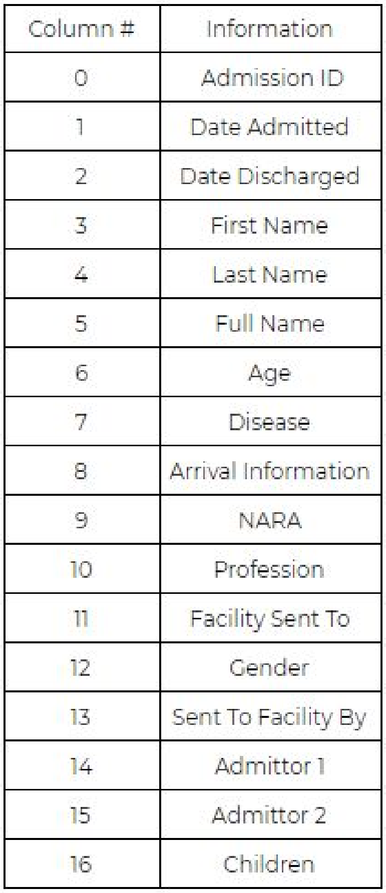
Reading: “Automated Inequality,” by Virginia Eubanks, Chapter 4, “The Allegheny Algorithm”

OVERVIEW

This homework is worth 100 points total toward your overall homework grade and is due Thursday, April 18th, 2019 at 11:59:59 pm. There will be only one part of this homework, and the file should be submitted as:

```
hw7Part1.py
README.txt
```

This homework will examine a data set on Irish immigration from Bellevue Hospital in 1920s New York City, scrutinizing the data, and creating statistical analysis. In this assignment, you will use dictionaries and simple sorting and indexing to explore the biases, assumptions, and systems of power extant in this data. You will examine how data is structured, what decisions data collectors and “social sorters” like immigration admittors make, how those social decisions become naturalized and cleaned in data, and how these decisions have historical impact.



Column #	Information
0	Admission ID
1	Date Admitted
2	Date Discharged
3	First Name
4	Last Name
5	Full Name
6	Age
7	Disease
8	Arrival Information
9	NARA
10	Profession
11	Facility Sent To
12	Gender
13	Sent To Facility By
14	Admittor 1
15	Admittor 2
16	Children

Input

The bulk of the homework will focus on reading in reading in a (large) data set, parsing it, “cleaning” the data, and storing the data into nested dictionaries. From these dictionaries, you should read through the data and write functions that identify trends and important statistics.

You should first focus on properly reading in the data. The file you read in is a .tsv, meaning each entry is separated by a tab character. To the left is a chart of how the data is laid out, as if it were indexed by line, which we strongly recommend you replicate for ease of parsing.

The data is not ‘cleaned’, meaning if you look through the set briefly, you’ll see extraneous characters that don’t make sense. Your code should handle these characters, either ignoring them or some other type of consideration.

As you are cleaning your data, be sure to note down the decisions you’re making for your README. What have you chosen to do with the data that doesn’t fit your program? Why doesn’t it fit? How much have you altered the “original” dataset? What do these “dirty” elements of the dataset represent?

After reading in the data properly, your code needs to store the data in proper entries in nested dictionaries. There should be two main dictionaries: one for Admittors, and one for Emigrants.
Here’s an example structure of what creating a nested dictionary would look like:

```
emigrants = dict()
emigrants[‘Susan’] = { ‘admittor_id’ : ’432511107’,  ‘gender’ : ‘female’, ‘disease’ :
‘pregnant’, ‘admittor_1’ : ‘G.W. Anderson’, ….. }
```

You can retrieve different entries, and information, in a few different ways with nested dictionaries. For example,

```
print(emigrants[‘Susan’])
```

Would result in the output,

```
{ ‘admittor_id’ : ’432511107’,  ‘gender’ : ‘female’, ‘disease’ : ‘pregnant’,
‘admittor_1’ : ‘G.W. Anderson’, ….. }
```

While this code,

```
print(emigrants[‘Susan’][‘gender’])
print(emigrants[‘Susan’][‘disease’])
print(emigrants[‘Susan’][‘admittor_1’])
```

Would print out,

```
Female
Pregnant
G.W. Anderson
```

From the provided files, you have four data sets with varying amounts of entries: 1000, 2500, 5000, and 10000. Do not test with the larger files until you are sure your code works. For grading, we will be testing with all of them.

Creating Dictionaries, Constructing Class

From this data you’ve parsed and stored into dictionaries, the trends and stats you’ve chosen to track will be created. You have some choice in this matter, however we do want you to store particular data points for each Emigrant and Admittor. For emigrants, you are required to track their name, diagnosis, and what location they were sent to. For admittors, you must track their name, number of patients, and the statistics of how many emigrants they sent to each facility (i.e., M.G. Leonard sent 1508 to Facility A, 265 to Facility B, and 12 to Facility C). As you noticed, there’s a lot more data provided than what we’re requiring you to track. Below are three examples of what you can track and calculate with your data and how you should format input.

```
With the Bellevue data, these are the three things chosen to track:
        1: G.W. Anderson's total patients and the ratio of the diagnoses given

        2: M.G. Leonard's total patients and the ration of the diagnoses given

        3: The comparison of illnesses diagnosed to each gender

        What would you like to view?
```

For your statistics, you should use two of the stats tracked in my example and add two of your own. At the top of your code, you should type out an explanation (approximately 250 words) about your chosen statistics, why you chose them, and some analysis about the results. We'll reward the extra credit based on what you've chosen.

In a README.txt file, use your designed sorting function to explore the dataset. **In prior weeks' lab, we've discussed the "social construction" of class, identity, and data**. Answer the questions in your read me and turn it in to get full points on this portion.

For this homework, you will be graded on:

- Parsing in and cleaning the data
- Tracking two of the given statistics
- Creating two of your own statistics
- Taking in continuous input to see those statistics
- Formatting output
- Using dictionaries
- Proper code structure
- Commenting, variable names
- Completion of the README