Assignment: Working With Data

Learning Objectives:

- Learn how to retrieve data from files.
- Utilize and manipulate data to get various outputs.
- Visualize data using matplotlib.
- Reflect on real-world implications of dealing with sensitive data and bias contained within datasets.

Part 0: Getting Started

Before getting started you will need to download the following:

- **`healthcare.py`**
- **`healthcare_data.csv`**

Make sure that these files are in your CS111 folder, and open them on VSCode. You will be doing all of your work within the **`healthcare.py`** file. Do not touch anything in the **`healthcare_data.csv`** file. It is there for you to look at and see what data you are searching through, but if you change anything in the dataset, that will impact the results that you are trying to get out. Throughout this assignment there will be reflection questions (indicated in italics) that you should answer in a file named **`reflection.txt`**. Include this in your submission.

Also, this assignment will require you to read some documentation about matplotlib. There will be some guidance later in the assignment to help you out.

Dealing With Sensitive Data:

As you are aware, information and data related to your health and healthcare are deeply private and should not be shared with just anyone. Data in this sector is personal and not something that the average person should have ready access to. If someone is to see this information, there are waivers that have to be signed so that there can be consequences applied if there is a breach of a patient’s privacy. With this data being so private, that also means that there are bad actors out there trying to access this data because it has inherent value due to how sensitive it is. That means if you are handling this data, it is crucial to be careful in how you handle it.

Making Generalizations:

Another aspect that we need to consider when working with medical data is the bias that is inherent in a potential dataset. When you are given a dataset, any dataset, you have to be aware of blind spots that might be baked into the numbers that you are given. An example of this in the medical field is in the datasets used to train AI models to diagnose various diseases and conditions. In these datasets, there is an underrepresentation of minorities as well as societal biases that appears in how the AI is able to identify and diagnose members of the affected groups. This means that when you work with data in this field, you have to be aware of the groups that you are either excluding or misrepresenting.

Recommended Reading:
Addressing bias in big data and AI for health care: A call for open science - PMC

Your Task:

In this assignment, you will create a small program to help you navigate the dataset that you have been given. You will pull out various numbers from a csv file and print it out so that it can be useful. Then you will both do some calculations with said data and use the Python library matplotlib to visualize the data as well.

Part 1: Know Your Data

Before you can do anything with the dataset, you have to know what information is stored and how it is organized within the dataset. The data that you are looking at is in the format of a .csv file. A .csv file is a ‘comma-separated-values’ file which, as the name indicates, each line of the data has fields that are separated by commas. In this particular dataset the fields are:

- Name
- Age
- Gender
- Blood Type
- Medical Condition
- Date of Admission
- Doctor
- Hospital
- Admission Type
- Discharge Date
- Medication
- Test Results

This may be a little difficult to read though, so, in order to make better sense of the dataset, we can write a program that can pull out specific fields from the file.

To start, go to the **healthcare.py** file and create a variable called “filename” and then, using the list of fields above, create lists with the variable names reflecting each of the different columns.

Python

```
filename = "healthcare_data.csv"

# Lists to store each column
names = []
ages = []
# TODO: Continue the lists below
```

Once you have a place to store all the information from the dataset, it is time to actually pull out the information from the file. Below is some code to help you get started on pulling this information out of the .csv file.

Python

```
# Open the file and read the data
with open(filename) as file:
    next(file)  # Skip the header line
    for line in file:
        fields = line.strip().split(",")
        names.append(fields[0])
        ages.append(int(fields[1]))
# TODO: Continue appending to the lists below
```

Now that you have gone through and pulled out the data from the file, to make sure things are stored properly, write a print statement that lists the first element of each of the categories. If you have done things correctly, it should print out:

Python

```
Bobby JacksOn, 30, Male, B-, Cancer, 2024-01-31, Matthew Smith,
Northview Medical, Urgent, 2024-02-02, Paracetamol, Normal
```

Then print only the first 5 medical conditions. This will print:

Python

```
Cancer
Obesity
Obesity
Diabetes
Cancer
```

Then print how many patients there are in the patient list. Your program should print ‘55500’.

Reflection: List some of the dangers that you can think of with giving outsiders access to identifiable data. How do people deal with this in the real world?

Part 2: What Can Your Data Tell You

Seeing the data from the file is a great start and a step in the right direction when it comes to actually working with the information that is being reported! The next step is to do some calculations with the information that you have pulled out.

The first bit of analysis you should do is finding what percent of people are in each of the following age groups:

- Under 18
- Between 18 and 30
- Between 30 and 50
- Between 50 and 70
- Over 70

Then you should write a print statement that lists out these calculated percentages in the terminal window.

Now, find the percentage distribution for the following categories:

- Gender
- Hospital

Reflection: Look at the distribution of the demographic data, is there equal representation across groups? Now look at the distribution of hospitals, is this equal? What implications could arise later if we try to draw conclusions from this data?

Part 3: Visualizing the Data

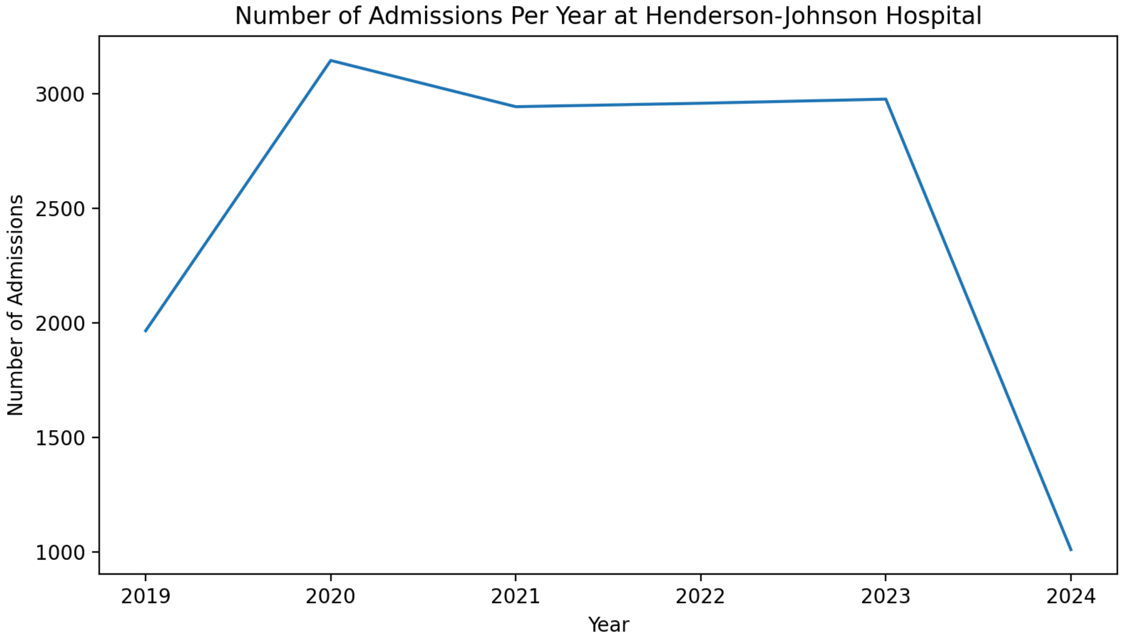
Printing out the results of these calculations to the terminal is all nice and good, but not the most useful way to see the outcomes of these calculations. If you were asked to show the findings you had to an employer or a primary investigator for your research, how you present and visualize you data can be very important. One of the ways that you could better visualize your data is through the library of matplotlib. This library has many different ways that you can create charts for your data. Before we take the next step, looking over the matplotlib documentation would be useful so that you can see how to use the functions that can create the final polished product.

For the sake of practicing using this tool, let's envision a potential scenario:

You have been hired by the Henderson-Johnson Hospital to do some data analysis for them. The hospital administration wants to know more about the patients that they are serving. You are being tasked to create a report for the next meeting for a few demographics. The first thing that you have been asked to create is a chart that shows the number of admissions per year.

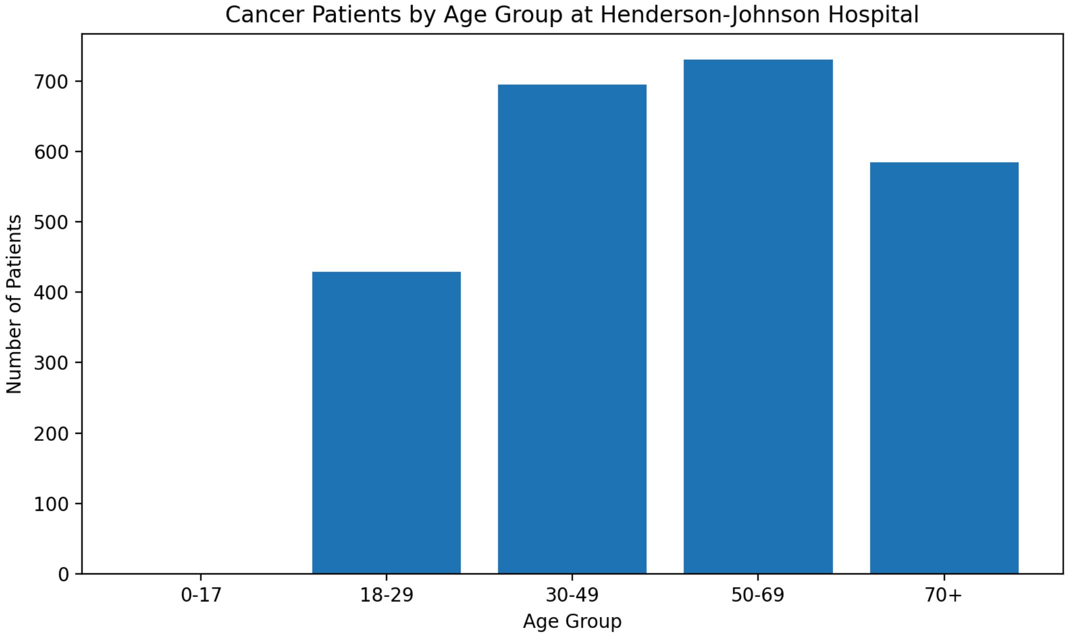
You should click on the link above to the matplotlib documentation site. This should take you to the homepage for the library. You can click on the Examples link, which will take you a list of various plot types that matplotlib can do. For this job, you can pick the generic plot(x,y) type. This will show you an example of how to create a plot in this format.

When you're done, you should get something that looks like this:



When you turn that in, the hospital admins were very impressed and they have asked you to create another chart! Now they want you to create a chart that can show the number of cancer patients by age group so that the doctors can see which age group is the most at risk. They want this to be a bar chart this time. You can go back to the matplotlib documentation and look up how to create this chart.

When you're done, it should look something like this:



Reflection: When you are getting output, how are you feeling about the output? Do you feel like this is accurate and that this dataset is representative? Do a bit of research and find another example of how data bias can impact technology. Who were the people affected? What was the harm that was caused?

Part 4: Reflection

In `reflection.txt` answer the indicated reflection questions in at least a few sentences. Make sure that you cite any sources that you use. This is not being graded on the quality of writing, rather it is being graded on the ideas that are being presented within the answers.

Grading Rubric

Criteria	Proficient	Exemplary
Logic	● One or more tests fail due to a small logic or formatting problem. ● Outputs are partially incorrect due to ○ Incorrect rounding ○ Mislabeled print output ○ Incomplete visualization ● Code executes without crashing but does not follow one or more instructions in the assignment outline.	● All automated tests (on pytest) passes. ● Code is able to run without errors, reproducing outputs that are more or less identical to the expected results. ● Data handling (file reading, list creation, calculations, plotting) is implemented following the directions outlined in assignment instructions.
Style	● Pylint score < 8.0 ● Minor stylistic issues, such as long lines, trailing whitespace, etc. ● Limited inline comments, but the code is still readable / understandable. ● Some warnings show up, but they do not interrupt with code readability or program execution.	● Pylint score ≥ 8.0 ● Code has consistent naming conventions (that is not inappropriate), spacing, indentation, docstrings, and encodings for file operations ● No major warnings, such as unused variables or inconsistent quotes. ● Clear structure that is easy to follow, with comments and descriptive variable names.
Reflection	● Answers all reflection prompts but with limited depth. ● Discusses ethical ideas but has insufficient supporting examples. ● Contains reflections but are rather brief, generic, or shallow.	● Gives a thoughtful reflection to all reflection prompts asked throughout the assignment. ● Demonstrates clear awareness of ethical considerations related to healthcare data, privacy, and bias. ● Discusses the ethics of handling sensitive data from at least three different perspectives. ● Shows personal insight or connection.