Assignment #3 - Sourcing Data, Summary Statistics, and Basic Data Visualization

👀 Click on this link: https://classroom.github.com/a/z4lVbdw8 (https://classroom.github.com/a/z4lVbdw8) to accept this assignment in GitHub classroom. This will create your homework respository. Clone your new repository.

Goals

1. source and document a dataset
2. use plain Python and built in modules to extract and transform data
3. work with numpy to calculate summary statistics
4. use matplotlib for simple data visualizations

Requirements

- 1 x notebooks:
  - `src/homework03.ipynb`
- 1 x original data set:
  - `data/raw/*`
  - ⚠️ include in repository if possible
  - otherwise, link to data set if file size > 100MB

Overview

In this assignment, you'll create a single notebook, (`ipynb`) that contains documentation and code. There are 6 parts to this assignment:

1. write some questions that *may* be answered by using summary statistics and / or creating visualizations
2. select and document a data set
   - ⚠️ the data should be comma / pipe / tab delimited...
   - ⚠️ the data set should have at least **one numeric column** and one column that contains **categorical data (https://en.wikipedia.org/wiki/Categorical_variable)**
   - include documentation regarding the source of the data
   - include the data set in your repository so that your notebooks can be run by the graders without having to perform any setup (if data set is > 100MB, link to file instead)
3. use "regular" python with built-in modules to create a data pipeline to extract or transform data
4. work with numpy or regular Python to calculate summary statistics
5. create at least two visualizations with matplotlib
6. write a conclusion answering your initial questions based on the summary statistics you calculated or visualizations you created
   - describe the results of your calculations and visualizations
   - describe whether or not they were able to answer your initial questions

Part 1 - Questions

Write some questions (minimally 2) that you think can be answered by finding the *right* dataset. 👀 See Vicky Rampin's (Librarian for Research Data Management and Reproducibility at NYU) slide titled "Steps to finding the *right* data" (https://tinyurl.com/find-data) to help construct your questions.

Focus on questions that *may* be answered by using summary statistics on numeric columns in a data set. You can also consider questions that can be answered by using rudimentary analysis on text-based columns that have a predefined set of possible values (categorical data), such as counts / frequency.

⚠️ **If you have trouble thinking of research topics / questions, you can do parts 1 and 2 in reverse order** ... by finding an interesting data set first, and then formulating questions afterwards.

Write your questions ... along with *where* you think you'll find answers within a markdown cell in your notebook `homework03.ipynb`. Use the following template to do this (fill in the parts marked with TODO).

Note that the markdown has some guidance for forming your question. For example, you might want to specify the population (who) that your question pertains to (for example, all students at NYU). However, if Who, What, Where, When or How aren't relevant for your question, you can put in not applicable.

```
# Homework 03
```

```
## Part 1 - Questions

### Question 1:

TODO (write the question)

* Who (population): TODO
* What (subject, discipline): TODO
* Where (location): TODO
* When (snapshot, longitudinal): TODO
* How much data do you need to do the analysis/work: TODO

### Question N:

TODO (write the question)

* Who (population): TODO
* What (subject, discipline): TODO
* Where (location): TODO
* When (snapshot, longitudinal): TODO
* How much data do you need to do the analysis/work: TODO

### Who Might Collect Relevant Data / What Articles or Publications Cite a Relevant Data Set?

TODO (answer question here)
```

Part 2 - Selecting a Data Set, Adding Documentation

Find a data set that you'd like to work with

Find a data set that you're interested in and that may answer your questions. It should be publicly available online as a text file (csv, for example). Download your data from a source that has some information regarding the provenance of the data (*where did it come from!?*).

Again, 👀 See Vicky Rampin's (Librarian for Research Data Management and Reproducibility at NYU) slides on finding data. Specifically, start with the Some good general data sources to start (https://tinyurl.com/find-data) slide to help in your search for a dataset.

Some guidelines:

1. do not use data sets that we've worked on or discussed in class, such as the Starbucks Drink Menu or Candy dataset because, uh... we probably already wrote (or will write) some code for it 🙃
2. try to write code that's different from - or builds significantly on - the programs that we've done in class (it's not adequate to simply use class sample code with a different data set 🤷)

Download the Data

If the dataset is less than 100MB:

1. download the data into the project repository (typically, you would not save your data as part of your repository, but this will streamline the grading process)
2. place your data in `data/raw/name-of-your-file`
3. you can ignore the `example-data.csv` that's already present in the folder

Write some documentation

1. In the notebook `homework03.ipynb` ...
2. Use the markdown below as a template to cite the data source and describe the data set
3. Fill in the parts marked as "TODO"

```
## About the Data

1. Name / Title: (TODO name of data set)
2. Link to Data: (TODO link to any documentation about the data that you've found )
3. Source / Origin:
        * Author or Creator: TODO
        * Publication Date: TODO
        * Publisher: TODO
        * Version or Data Accessed: TODO
4. License: (TODO name of license)
5. Can You Use this Data Set for Your Intended Use Case? (TODO answer this question)
```

Show Some Data and Document its Format

Document the format of the dataset in your `homework03.ipynb` notebook using the template below:

```
## Part 2 - Selecting a Data Set, Adding Documentation

### Overview

Format: (TODO add what file format the data is in)
Size: (TODO how large is the file in KB, MB, GB, etc. ... use finder, windows explorer for this)
Number of Records: (TODO how many rows)

### Sample of Data

TODO show a few lines of data from the actual file.  ⚠ Use "regular" Python to do this in this code
block.  Assuming that jupyter-lab was started in your root directory: with open('../data/raw/example
-data.csv', 'r')

### Fields or Column Headers

* Field/Column 1: (TODO add field name and potential type using Python types)
* Field/Column 2: (TODO same as above)
* Field/Column N: (TODO same as above)
```

Part 3 - Extract / Transform

Create a markdown cell containing a description of:

- what columns you'll be using from your dataset
- how you're planning to convert the data into analogous python types for those columns
- any kind of cleaning that you'll have to perform:
    - *normalizing* data, such as multiple spellings for the same categorical value (Yes, Y, YES)
    - filling in missing values or using a special indicator for special values

- etc.

Use the following as a template

```
## Part 3 - Extract / Transform

(TODO describe your process for extracting, transforming, cleaning your incoming data)
```

Underneath your markdown cell, create a new code cell. Use:

- "regular" Python 🐍
- you'll also probably want to use the `csv` module (https://docs.python.org/3/library/csv.html)
  - see the docs (https://docs.python.org/3/library/csv.html)

    ```
    with open('example.csv') as f:
      reader = csv.reader(f)
      for row in reader:
      # row is a list of strings!
    ```

- but do not use `pandas` 👷‍♂️ 🐼 for now

To do the following:

1. once again, read the file (make sure you're reading the file relative to the root of your project:
   - assuming that you start jupyter from the root, and your notebook is in `src`, use `../data/raw/example-data.csv`
2. "extract" at least 3 columns that you want to work with:
   - one with numeric data
   - another with categorical data
   - and any *kind* of data for the third column
3. transform at least 1 of those columns to a Python numeric type
4. store this data in a data type of your choosing; **you'll use this data later**
   - for example, you can read everything into a 2-dimensional `list`, multiple `numpy` arrays, a dictionary, etc.
   - you must avoid using `pandas` `Dataframe`s
5. clean your data if necessary:
   - identify missing values and replace with an appropriate marker (for example, you might decide to use `None` as the value that represents a missing value)
   - alternatively, find some strategy to fill in missing values (for example, default to 0, or No)
   - normalize casing for categorical data
   - etc.

You can create as many variables, helper functions, or even classes as needed.

Part 4 - Descriptive Statistics

In this part, you *can* use an external library, `numpy`.

Using Python ("plain" python, `numpy` or the `statistics` module), calculate the following descriptive statistics on at least one of the numeric columns that you saved from the previous part.

1. examine number of records, range / outliers with min and max
2. measure central tendency by calculating at least **one** of the following:
   - mean
   - median
   - mode
3. measure dispersion with **at least one** of the following
   - variance
   - standard deviation

For your categorical data, calculate:

1. frequency
2. unique values

You can use `numpy` 's functions / methods. You can also try out Python's `statistics` module, which is built in.

You can intersperse markdown blocks and Python code to display your calculations:

```
## Part 4 – Descriptive Statistics

### Analysis on Numeric Data

#### Central Tendency

TODO: add a code block to display results

#### Dispersion

TODO: add a code block to display results

#### Outliers

TODO: add a code block to display results

#### Other

TODO: add a code block to display results

###  Analysis on Categorical Data

#### Frequency

TODO: add a code block to display results

#### Unique Values

TODO: add a code block to display results
```

Part 5 - Visualizations

Using your dataset, create at least two visualizations with matplotlib. Do this in a code cell using regular Python with `matplotlib` so that your charts and graphs appear in the notebook

Describe what you're trying to show with the visualizations in a markdown cell above your code:

```
## Part 5 – Visualizations

TODO Describe the visualizations in the previous cell
```

Part 6 - Conclusion

Write a conclusion answering your questions based on the results in any of the previous parts.

- describe the results of your calculations / analysis
- describe whether or not they were able to answer your questions
- if you were unable to reach any conclusions, discuss why this was the case

Do this in markdown cell:

```
## Part 6 - Conclusion

TODO: write your conclusion here (interpret results of calculations; does it help answer your origin
al questions?)
```

Part 7 - Heights and Weights

Examine the relationship between height and weight using the National Health Interview Survey (https://www.cdc.gov/nchs/nhis/2019nhis.htm?utm_source=pocket_mylist (https://www.cdc.gov/nchs/nhis/2019nhis.htm?utm_source=pocket_mylist)) data. To do this, start with a markdown cell:

```
## Part 7 - Heights and Weights

TODO: add your code cells below this!
```

Then, follow these steps:

1. Download the zip file, uncompress it and add it to your repository: https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Datasets/NHIS/2019/adult19csv.zip (https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Datasets/NHIS/2019/adult19csv.zip)
2. Read the codebook (https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NHIS/2019/adult-codebook.pdf) and/or summary (https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Dataset_Documentation/NHIS/2019/adult-summary.pdf) to:
    - find the field names for height (in inches) and weight (in pounds)
    - find out if there are any markers for "missing" values
3. In a code block:
    1. read in the file using the `csv` module
    2. extract the data for height and weight, making sure to discard any rows that have a missing value for either
    3. use numpy.corrcoef (https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html) to calculate the correlation coefficient for height and weight and display the results
    4. with matplotib, create scatter plot (https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.scatter.html) to show the relationship between height and weight
    5. the plot will be crowded due to a high number of points; use `alpha=0.01` to make the points transparent
    6. additionally, you can try adding jitter so that the points don't align: see our book, think stats (https://greenteapress.com/thinkstats/html/thinkstats010.html#scatterplot2) for an example (search for jitter in text below the images)