Homework 0x00 - Reading CSV Files and Using matplotlib

Start Assignment

- Due Sunday by 11:59pm
- Points 10
- Submitting a file upload
- File Types pdf

Overview

This short homework assignment is to be completed individually and will provide a chance to practice writing a simple Python script to plot data from a file. The assignment will also provide some support on upcoming lab assignments in which you will collect present data using lab hardware.

The Assignment

Write a Python program for a PC (not a Microcontroller) which plots data from a CSV (comma separated variable) file on a graph. The file, available here, is named **data.csv (https://canvas.calpoly.edu/courses/161863/files/18262032?wrap=1)** **(https://canvas.calpoly.edu/courses/161863/files/18262032/download?download_frd=1)** and follows these conventions:

- There are two or more columns of data, the columns being separated by a comma. There is an unspecified number of rows of data, each separated by a newline character, "\n".

- The data may be integers or floating point numbers; your program must run without error for either type but it is permitted to convert all numbers to be of floating point type if you so choose.

- Only the first two columns contain valid data; data in other columns should be ignored by your script.

- There may be whitespace (spaces or tabs) between columns, before the data, or after the data; it must have no effect on your script.
- There may be rows which do not contain two readable numbers; these rows must be ignored by your script and not cause it to crash or stop reading data.
- The first line of the file contains two strings, separated by commas, indicating header labels for the two columns of data.
- Some rows of data may have comments at the end starting with a "#" sign (to be referred to as a "pound" sign). These comments should be ignored while plotting and otherwise not affect the operation of your script.

Example CSV File

The following snippet shows how the first 15 lines of the CSV file will look.

```
Time (s), Height (m)
0.0,0.0
0.12048192771084336,0.11732910990121602
0.24096385542168672,0.22741080854081042
0.36144578313253006,0.32896568339007426,3.1415
# 0.0 0.0 0.0, 0.0
0.48192771084337344,0.420900033048894423423425067902706987059670329875987039867
0.6024096385542168,0.5023132288357176
0.7228915662650601,0.5725018879845621
waffles!!!
0.8433734939759036,0.6309609470332831
0.9638554216867469,0.6773817654486048
1.0843373493975903,0.711647427129721
```

Requirements and Deliverables

Your script must meet the following requirements:

- You will need to import the `pyplot` plotting interface from the `matplotlib` module to generate plots. You can do this import using `from matplotlib import pyplot`. No other imports are allowed for the assignment. Everything else must be done using standard Python functions.

- There must be labels on the graph's horizontal and vertical axes corresponding to the column labels in the CSV file. These labels must be extracted from the CSV by your script, not "hardcoded" as static labels.

- You may use *some* code found online, but you are encouraged to attempt the assignment honestly with the goal of learning.

  For example, you will benefit from searching online for topics like "python split string" or "python convert string to float"

- For each "rejected" row from the CSV, print a line of text to the Python console indicating the line number from the CSV and the reason why the row was rejected. Note that the file starts on "line 1" with the column labels.

- Your code should not be optimized for the specific CSV file provided for testing. That is, any CSV file meeting the specifications made above should work with your script. Note that, while grading, a different CSV file may be used for testing.

- Make sure that your code uses "relative" pathing to access the CSV file. That is, when opening the file, your code should reference the location of the CSV file relative to the location of the Python script, not as an absolute path on your personal computer.

- Your script should run on any file named "data.csv" in the same directory as the script as soon as it is run. That is, your script is *not* intended to be run from the command line with command line arguments.

When you have completed the script, create an extremely brief PDF memo that shows your produce plot and a transcript of the log of rejected rows from the CSV; also include your code as an attachment at the end of the memo. Submit the memo here before the deadline of Sunday 9/28 at 11:59PM.