Studio 11: Data Science

CSCI 128

Introduction

For week 11 we will be using a module that is not runnable on Bartik: `pyplot`. Thus, for this week only, we are hosting studio problems on Gradescope instead. This document describes the prompts you will need to know to complete the studio problems.

As with previous weeks, you do not need to attempt all problems. Problems this week are a mix of practice with `csv` and `pyplot`. Please complete at **least 3 problems** of your choice to earn full credit for Studio 11. We highly recommend completing at least one of the two plotting problems (P4 and P5) to better prepare you for Assessment 11.

Unlike with Gradescope assessments, the studio problem autograders will allow unlimited submissions; feel free to try things out until they work. We will not offer local autograders for these studio problems.

Problem 1: ReadWriteCSV

Your program must open the file `numbers.csv`. The first line in the file is a single digit (we'll call this num) and the remaining lines are sets of integers separated by commas. For each line of numbers, if num is repeated num times in that line, write true to a new csv file called `results.csv`. Otherwise, write false to the results file.

Sample Input 1 (contents of `numbers.csv`)

```
3
3,2,3,3
1,2,3
4,8
1,3,3,3
```

Sample Output 1 (contents of `results.csv`)

```
true
false
false
true
```

Sample Input 2 (contents of `numbers.csv`)

```
2
8,9,3
4,2,2,5
2,2
6,2,2,2,8,9
2,7,4,2
```

Sample Output 2 (contents of `results.csv`)

```
false
true
true
false
true
```

Problem 2: Transpose

Given a comma-separated file `matrix.csv` containing a 2D matrix, transpose the matrix (that is, swap the rows and columns) and output it to `matrix_transpose.csv` in a comma separated format.

Sample Input 1 (contents of `matrix.csv`)

```
1,2,3
4,5,6
```

Sample Output 1 (contents of `matrix_transpose.csv`)

```
1,4
2,5
3,6
```

Sample Input 2 (contents of `matrix.csv`)

```
can
you
handle
one
column?
```

Sample Output 2 (contents of `matrix_transpose.csv`)

```
can,you,handle,one,column?
```

Problem 3: ExamStats

Given a file containing comma-separated student answers to a True False (T/F) exam with an answer key (`stats.csv`), print as output the high, low, average, and median student scores to two decimal places as a percent.

Assume the correct answers are the first line of the file, followed by student answers. Each question is worth one point. If there is an even number of students, then the median is the average of the middle two scores.

Note: Since this problem will be graded on Gradescope, please be sure to prepend your printed output lines with the string ”OUTPUT”.

Hint: sorted(), max(), min(), and sum() will be useful functions here!

Sample Input 1 (contents of `stats.csv`)

```
T,T,T,T
F,F,F,F
T,F,T,F
T,T,T,T
F,F,F,T
```

Sample Output 1 (printed to console)

```
OUTPUT High: 100.00%, Low: 0.00%, Median: 37.50%, Mean: 43.75%
```

Sample Input 2 (contents of `stats.csv`)

```
T
F
```

Sample Output 2 (printed to console)

```
OUTPUT High: 0.00%, Low: 0.00%, Median: 0.00%, Mean: 0.00%
```

Sample Input 3 (contents of `stats.csv`)

```
T,T,T,T,F
F,F,F,F,F
T,F,T,F,F
T,T,T,T,F
F,F,F,T,F
F,F,F,F,F
```

Sample Output 3 (printed to console)

```
OUTPUT High: 100.00%, Low: 20.00%, Median: 40.00%, Mean: 48.00%
```

Problem 4: DieRolls

Write a program that takes a random seed value and a number of iterations: `n` as user input. Your program should import the random library, set the random seed, and then simulate `n` six-sided die rolls. The code should generate random integer values between 1 and 6.

Your program will create, and save as an image named rolls_bar_chart.png, a bar chart showing the number of times each die side appeared during the rolls. This chart should not include any additional formatting, otherwise it will not pass the autograder. See the examples below for specific charts generated.

Recall how to set the random seed and generate a random integer:

```
import random
seed = input()
random.seed(seed)

# generates a random integer between 5 (inclusive) and 10 (inclusive)
random.randint(5, 10)
```

Sample Input 1

```
abc
10
```

Sample Output 1 (contents of `rolls_bar_chart.png`)

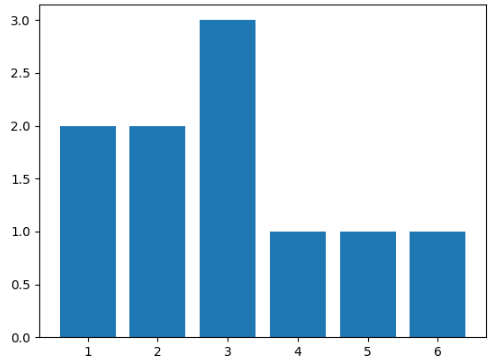

Figure 1: Output Image for Example 1

Sample Input 2

```
abc
1000
```

Sample Output 2 (contents of `rolls_bar_chart.png`)

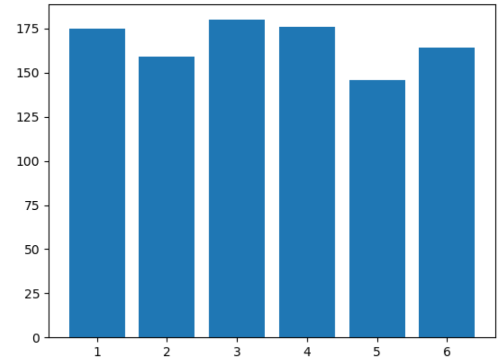

Figure 2: Output Image for Example 2

Sample Input 3

```
uniform
100000
```

Sample Output 3 (contents of `rolls_bar_chart.png`)

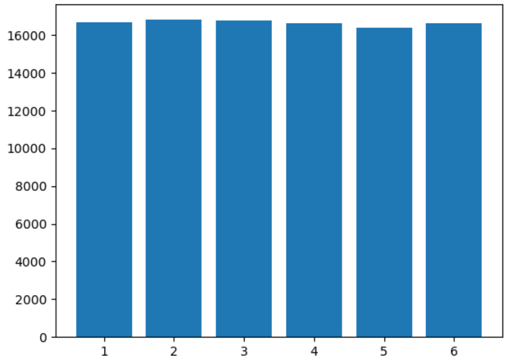

Figure 3: Output Image for Example 3

Problem 5: HeightWeight

Write a program that takes a csv data file named `height_weight.csv` as input. Your program should parse this file into lists of weights and heights, and then plot these values with a scatterplot.

`height_weight.csv` contains two columns of numeric data. The first is heights in inches, the second is weights in pounds. Values are separated by commas.

To make the plot more readable, also add a plot title and labels. The title should be ”Height vs. Weight”. The x-axis should be labeled: ”Height (in)” and the y-axis labeled: ”Weight (lb)”.

The plot generated by your program should be saved as height_weight_scatter.png

Sample Input 1 (contents of `height_weight.csv`)

```
73.847017017515,241.893563180437
68.7819040458903,162.310472521300
74.1101053917849,212.7408555565
71.7309784033377,220.042470303077
```

Sample Output 1 (contents of height_weight_scatter.png)

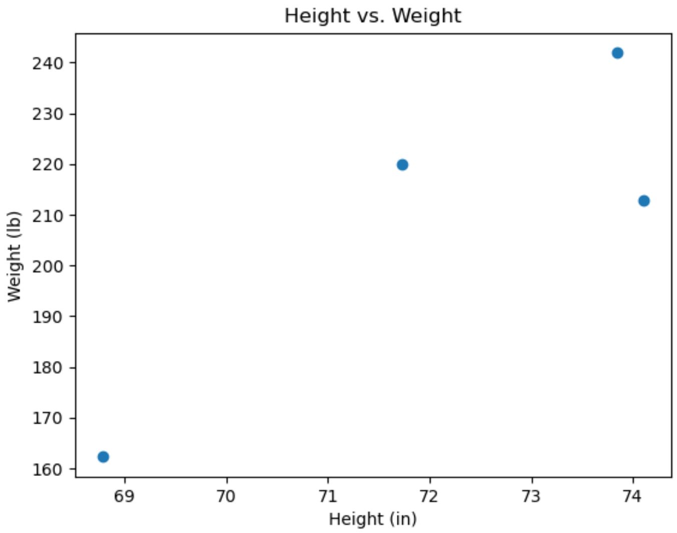

Figure 4: Output Image for Example 1

Sample Input 2 (contents of `height_weight.csv`)

```
66.1491319608781,149.173566007975
75.2059736142212,228.761780615196
67.8931963386043,162.006651848287
68.1440327982008,192.343976579187
69.0896314289256,184.435174408406
72.8008435165003,206.828189420354
67.4212422817167,175.213922399227
68.4964153568827,154.342638925955
68.6181105502058,187.506843155807
74.0338076216678,212.910225325521
71.5282160355709,195.032243233835
```

Sample Output 2 (contents of height_weight_scatter.png)

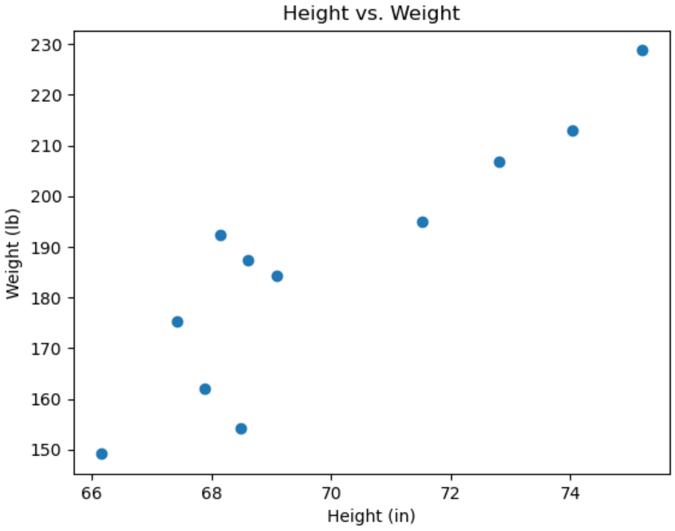

Figure 5: Output Image for Example 2