CSCE100 Introduction to Informatics
Fall 2021

Programming Assignment 3: Data File Statistics Generator

Points: 100 points. Assignment Date: September 21, 2021 Due Date: September 28, 2021

Objectives

1. To familiarize with writing and running Python programs and the Python environment
2. To familiarize with the use of loops (e.g., the for and while loops)
3. To familiarize with data structures, particularly arrays/lists
4. To familiarize with file input/output in Python
5. To be exposed to the use of built-in functions
6. To be exposed to the use of built-in modules or packages (e.g., import csv)
7. To familiarize with the use of online documentations on Python

Problem

Write a program that will read in a csv (comma-separated values) file and read in all the values from the csv file and store them in arrays/lists. After that, the program will perform analyses on the data, and store the results in an output file in the csv-format. Here are some additional requirements:

- You are required to come up with the csv file itself. It must have at least 100 rows of data, and each row has at least five attributes or columns. Some of these attributes should be numeric, and some of these attributes should be categorical (such as colors = ['blue', 'green', 'red'] or quality = ['good', 'average', 'poor']). (15 points)
- You are required to submit a (a) description of the csv file itself, (b) the rationale behind your selection of the csv file for your analysis, and (c) what each column/attribute represents. (15 points)
- Your program is required to perform ***at least five types*** of analyses of your choice (20 points). Here are some ideas:
    - *Histogram of categorical values*: e.g., 10 entries with 'blue', 5 entries with 'green', 10 entries with 'red';
    - *Pie chart values*: e.g., 40% with 'blue', 20% with 'green', 40% with 'red';
    - *Correlations*: Series 1 and Series 2 have a correlation of 0.712;
    - *Associations*: For all entries with 'blue', their quality is 'good' (80%), 'average' (15%), and 'poor' (5%))
    - *Statistics* such as average, minimum, maximum, standard deviation, etc.
- Your program is required to read in the above csv file. (*Hint*: Use import csv; use the csv module to read row by row using a loop.) (5 points)
- Your program is required to generate the output of your analyses to an ***output csv file***. (5 points)

- You are required to write a description of the analyses that your program performs on the data: (1) how they are computed, and (2) why and how they are informative analyses. (10 points)
- You must document your program (see https://devguide.python.org/documenting/).
    - Name, Date, Affiliation, a description of the program, what inputs does it need, what outputs does it generate (5 points)
    - Inline comments in the program (5 points)

Example Input/Output: None

Handin

1. The submission deadline for all handins is 11:00 AM September 28, 2021. **Late handins will *not* be accepted or graded.**
2. You are required to handin a screen capture of your "testing session" using your program. (5 points)
3. You are required to handin all program files. (5 points)
4. You are required to handin all input and output files. (5 points)
5. You are required to handin your description file that consists of three parts: **input file description, output file description, and analyses description**. (5 points)
6. You are required to handin online the above files to Canvas under Programming Assignment #3.

Think About

Now, think about what if we want to modify how the columns are arranged (e.g., adding new attributes, removing some attributes) or how the rows are to be filtered so that only certain subsets are being considered for analyses. Do we need then to create many versions of the csv input file? What are some potential pitfalls with maintaining many versions of the csv input file? Furthermore, think about the output file format. What if our users or customers want something different, e.g., in a different format, or to derive or visualize different types of results from your analyses? How would you then create an output file that is more universally useful? (Hint: Think about databases.)