UNIVERSITY OF SASKATCHEWAN

Department of Computer Science
176 Thorvaldson Building
110 Science Place, Saskatoon, SK, S7N 5C9, Canada
Telephine: (306) 966-4886, Facimile: (306) 966-4884

Assignment 9

Grow Weeds

Date Due: Thursday, December 5, 6pm **Total Marks: 27**

General Instructions

- **This assignment is individual work.** You may discuss questions and problems with anyone, but the work you hand in for this assignment must be your own work.
- **Assignments are being checked for plagiarism.** We are using state-of-the-art software to compare every pair of student submissions.
- **Read the assignment carefully.** All of it. The problem cannot be described in a few sentences, and except for a little bit of fun context, all parts of the question have important information crucial to your success.
- The assignment indicates what to hand in. Make sure your name and student number appear at the top of every document you hand in. These conventions assist the markers in their work. Failure to follow these conventions will result in needless effort by the markers, and a deduction of grades for you. Do not submit folders, or zip files, even if you think it will help.
- **Programs must be written in Python 3.** Marks will be deducted with severity if you submit code for Python 2.
- **Assignments must be submitted to Moodle.** There is a link on the course webpage that shows you how to do this.
- **Moodle will not let you submit work after the assignment deadline.** It is advisable to hand in each answer that you are happy with as you go. You can always revise and resubmit as many times as you like before the deadline; only your most recent submission will be graded.

Question 1 (27 points):

It's time to put all of your Python skills to work. This time we are giving you a non-trivial problem and a method to solve it, but giving you very little guidance on how to use python solve it. You are allowed to use any of the Python language features and problem solving methods that we have discussed in class or used on assignments or in labs in order to solve the problem. You will be graded not only for correctness, but also for making good choices about how you store data, how your organize your program, how well you document (comment) your program, and how well you test your program.

Problem Description

In order to determine the feasibility of growing weeds as a cash crop in Saskatchewan, samples have been taken from wild weeds in various locations. In each sample the amount of vitiamin C, and the amount of gamma-linolenic acid (GLA) has been measured. The samples with low levels of nutrients or samples containing toxins are discarded, so in the data only samples from edible weeds remain. The next step is to understand how many species of edible weeds are present at a each location.

Your task for this assignment is to write a program to display the different species of weed present at a given location, based only on the given data. It is unknown how many species may be involved at any one location , but the number is believed to be between two and four. At a particular location, the samples from one particular species will have similar levels of nutrients and determining the species present is a matter of grouping the data so samples in a given group belong to the same species. This grouping process is called clustering. Given a file of pairs of integers, giving the levels of Vitamin C and GLA, your task to to plot the data with each species given a different color, and to produce a representative for each species at that location. The representative of a species will have the average amounts of nutrients of the members of the species.

Problem Details

Input Data

A given input file is associated with a particular area of the province. Each line of the file contains the infomation for a particular sample and the Vitamin C and GLA measures for a sample is separated by a comma. For example

```
40,40
10,10
200,200
230,231
40,43
15,45
220,190
```

is the contents of the file `backyard.txt` giving information on seven samples, and the last sample has vitamin C level 220 and GLA level 190.

Species Identification

Since the number of species at a location is not known, we will need to consider the possibility that there are s species present, where s can vary from 2 to 4.

Given s and a data file we need to cluster the data into s groups and determine what the nutrient levels of an representative of a group would be. Note that the representative for a group will generally not be an

actual member of the group, but a "fake" member having the "average" nutrients of the members in the group.

To do this as a first cut, we first grab the first s samples, and use the nutrient levels of these samples as those of the representatives (reps) of the s species. Keep track of the reps for each species separately from the data on the samples, as although they start off as having the same nutrient levels as individual samples, later they will be "fake" members and have different nutrient levels then real actual samples do.

A sample will assumed to belong to the same species as one of the reps when it is closer to that rep than to any other rep. We measure closeness by the usual Euclidean distance, ie. the distance from sample (u,v) to rep (x,y) is $\sqrt{(u-x)^2+(v-y)^2}$. We now label all the samples as to which of the s reps they are closest to. This process is called labelling.

We next find new reps for each of the s groups. The new rep for a group will have a Vitamin C level which is the average of the Vitamin C levels of the samples in the group, and likewise it will have a GLA level which is the average of the GLA levels of the group members. This process is called selecting representatives.

Now since the nutrient levels of the reps have changed, we relabel all the samples so they again are labeled like the closest rep and we again update the nutrient levels of the reps since the group members have changed. This process of labelling and updating is repeated over and over until there are no more changes in the group memberships.

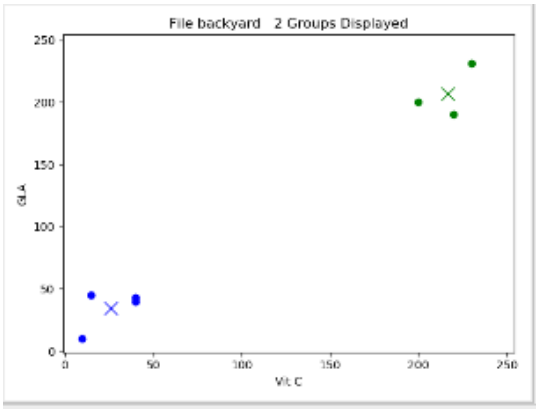
To recap, we iterate over the following two steps until the there are no more changes.

- Select representatives
- Use the representtives to label the samples

Display Results

For a given file and number of species s, plot the samples so that the samples from the same species have the same color, but samples from different species have different colors. Also plot the reps for each species.

For the `backyard.txt` input file with s equal to two, the plot should look like this:



We are providing you with **four input files** for four different locations. For each location submit the plot for the s which best seems to match the data at that location. Although there are some algorithmic methods to decide which s to use, here we will assume there is a human in the loop ("you") to decide which s to use for the plot you submit.

For example, for location file `locationA.txt` select best plot and save it in the file `a9-locationA.png`.

Testing

Because you do not have the answers in advance for most of the input files, you will have to employ your testing skills to ensure the correctness of your program. Therefore, you should create a set of tests and a matching test driver for each function that you write as part of solving the main problem. In a **separate file** from your main program, hand in any test drivers that you write.

Hints

- Divide up the work you need to do to solve this problem into subtasks and determine the best order in which to solve them. Use your lab sections during the next week to plan your approach!
- Make sure the code for each sub-task is working before you proceed to the next one (using testing).

What to Hand In

- Submit your solution to the main problem in a file called a9q1.py
- Submit your test drivers in a file called a9q1_tests.py
- Submit four output files `.png` files, one for each input location file. You save the plot for the s which best seems to match the data at a given location. For example, for location file `locationA.txt` select the best plot and submit it in the file `a9-locationA.png`. To save a plot in pycharm right click (control click on a Mac) on the thumbnail of the plot and a menu will pop up allowing you to save the plot as a `.png` file.
- You do not have to submit the provided input files themselves.

Evaluation Criteria

A detailed grading rubric is found at the end of this document. Evaluation criteria include:

- Appropriate organization of the program using functions and/or modules.
- Use of appropriate data types.
- Good use of commenting and docstrings.
- Evidence of thorough testing (test cases and test drivers).
- Correctness of program.

Detailed Grading Rubric

3 Marks: Organization of the program using functions and/or modules	
Grade	Criteria
3	Excellent: functions and/or modules break the program into cohesive units.
2	Good: a few functions are poorly designed.
1	Weak: Many functions are poorly designed.
0	None: Functions are not used or everything is in one function.

3 Marks: Use of Appropriate Data Types	
Grade	Criteria
3	Excellent: The most appropriate data types were chosen for storing program data.
2	Good: Mostly good choices for data types were made.
1	Weak: choices for data types lead to inefficiencies, difficult to read, or excessively long code.
0	None: No credible thought put into choosing data structures.

3 Marks: Good use of commenting and docstrings.	
Grade	Criteria
3	Excellent: docstring descriptions of function parameters, and return value are present, well-explained, and unambiguous; inline comments are plentiful and helpful.
2	Good: docstrings contain most or all expected information and most or all of it is adequately explained; inline comments are frequent, and mostly helpful.
1	Weak: much docstring information is missing and/or most explanations of parameters and return values are unclear; inline commenting is sparse and terse.
0	None: No attempt at documentation.

3 Marks: Evidence of Thorough Testing.	
Grade	Criteria
3	Excellent: each function's correctness is demonstrated by multiple sensible test cases.
2	Good: a few functions are under-tested; maybe one function is not tested.
1	Weak: minimal evidence of effort in testing.
0	None: No evidence of testing was submitted.

3 Marks: Correctness of file input.	
Grade	Criteria
3	The data in the four location input files is read correctly.
2	There are some problems in reading the data files.
1	There are many problems in reading the data files.
0	Data was not read from the input files.

8 Marks: Correctness of the species identification.	
Grade	Criteria
8	Excellent: The samples are correctly grouped into s species, using assign and update repeatedly
5	Good: The samples are grouped into s species mostly correctly.
2	Marginal: The samples are grouped into species, but not correctly
0	Weak: The samples were not grouped

4 Marks: Correctness of the plotting	
Grade	Criteria (Choose the best match)
4	Excellent: The program correctly displays the s species using different colors, the reps are also displayed
3	Good: the display is not perfectly correct, but approximately right
1	Weak: There is a plot produced, but it is far from correct
0	None: the program does not produce a final plot