**The Sorting Hat**: A Program To Create Groups Based On Student Preferences

**Executive summary**: The SEG group brainstormed an idea to help create groups of students according to their working habits/preferences. This concept was formed to refine the random group allocation process currently in place so that students obtain the benefits of random allocation (i.e. meeting new people, simulating the workforce) and have a better chance of being in a compatible group with similar working styles.

**Aim**: To develop a standalone program that creates groups of students based on student preferences for group work and their working habits.

**Approach:**

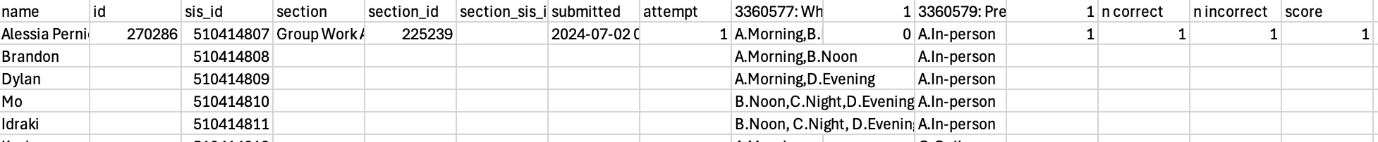
Coded in python, the sorting hat program uses pop-up message boxes to step the user through the program. The user chooses a csv file, and this csv file is the Canvas quiz download containing the results of the student preferences. The program then reads this input and creates a student object for each student’s preference.

To form groups, the function alphabetically sorts the students according to the meeting time preference first, and then by their preferred mode of meeting. Using this sorted list of students, the program forms groups of the desired number (or one less than the desired number).

The sorting hat writes these groups to a csv file named by the user and then the program ends.

**Technical Step-Through:**

*Input:* The csv file that contains the student identifier (i.e. SID or full name), their preference for the time of the day to meet and their preference for mode of meeting. The students in such csv file are from the same tutorial.



The input file is read and each student is created as a Student object. A list of Student objects is returned:

A computer screen with text

Description automatically generatedA screenshot of a computer program

Description automatically generated

The form\_groups function initially figures out how many groups can be made with the user’s desired group size. If the total number of students is not exactly divided by the desired group size, then the program works out how many groups need to have a group with desired size minus one:

A screenshot of a computer screen

Description automatically generated

The function then sorts the list of students first by preferred time and then by preferred mode of meeting. There is also a counter for the amount of groups so that the program knows when to switch the group size (if applicable):

A computer screen shot of text

Description automatically generated

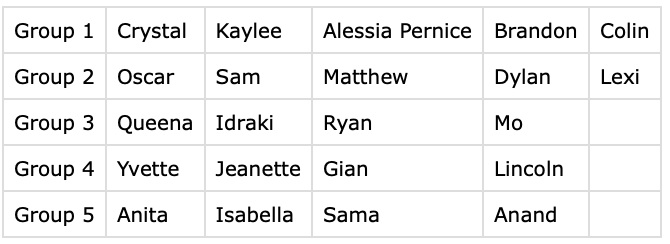
The program then iterates through each student in the list of sorted students and appends them to the current group. I added a couple lines there to track the preferences in each group, just as an FYI for testing. If the number of students in the current group is equal to desired group size then that group is added to the list of groups. I have added some lines in there to record the dominant preference, for testing reasons. Then the current group is reset to be able to add new students to it:

A screenshot of a computer program

Description automatically generated

*Output:* A csv file with the list of groups and the students in each one. At present, the student names are used, but this can easily be modified to return their SIDs:

A screen shot of a computer program

Description automatically generated

The main function runs and calls the other functions. There are message boxes that pop up to the user and allows them to choose their input csv file, desired group size and the name for the output csv file:

A screenshot of a computer program

Description automatically generated

**Issues:**

There are 2 main issues preventing the release of the Sorting Hat.

The first issue is creating a standalone executable file of the sorting hat. The aim of the sorting hat is to be an easy-to-use application that anyone can download and open to get the program running. Currently the sorting\_hat.py file works well but requires users to have downloaded python, installed relevant libraries and use terminal to run the python file. I have used Pyinstaller to create a standalone executable of the sorting\_hat.py file however this only works on computers with arm64 architecture (e.g. MacBook M2 chip) and still sometimes requires installation of some python libraries.

The second issue is the approach to forming groups. The aim of the sorting hat is to create groups of students in a way that maximises their group-work habits/preferences from the Canvas quiz. Currently, the program sorts the list of students according to their preferred time of meeting and then sorts students according to their preferred mode of meeting. The issue is that the preferred time of meeting can be a combination of answers, for example: “A. Morning, B. Noon” or “A. Morning, B. Noon, C. Evening”. The second sorting of students (i.e. preferences for mode of meeting) only sorts students with the exact same combination of meeting time preferences and will therefore make groups of students with potentially different mode of meeting preferences

. This is a very simple, heuristic approach that seems to work well enough for our intention, I did try a linear regression model but that was quite unsuccessful.

Other issues to consider:

* Unsatisfied students due to shortage of students with similar preferences.
* Might need to distinguish between tutorials if there is not a setting to do that through the Canvas quiz.

**Recommendation:**

Fix the 2 main issues mentioned above and dedicate some time to testing the program. Test cases can be made or to keep within budget (and timeline), black-box user testing can be conducted to ensure robustness for the time being. There are 20 hours left out of the allocated 30 to achieve this. Recommended to add more resourcing if after 20 hours, the 2 main issues have not been resolved.