STAT535: Statistical Computing **Fall 2020**

Problem Set 3: Simulation

P. Flaherty

Objective This problem set will introduce simulation and visualization. This builds on your previous problem set, so make sure that your word game passes all tests. You will build a computer player for the word game and then use that player to offer hints to the user. You will visualize the quality of the hints provided by the computer player.

Collaboration You may work with other students. However, each student must writeup and hand in their assignment individually. Be sure to indicate whom you worked with in the comments of your submission.

Logistics Problems 1 and 2 use functions that you wrote in `ps2.py`. While you are testing your code on your own computer, put `ps2.py` in the same folder as your `.py` files for this problem set and use `import ps2` to import those functions into your problem set 3 python code. When you submit your `.py` files to gradescope, you do not need to include `ps2.py` because the autograder already contains the correct solutions in `ps2.py` and your solution code will use those functions instead.

For Problem 3, you will be modifying a jupyter notebook. Instead of submitting `.py` files, you will be submitting your finished notebook. To do so, select ``` ``Download as...-> HTML'' ``` from the `File` menu to produce a HTML output of your finished notebook. Then, open the HTML file in your web browser and `print` the webpage as a pdf file. Submit the pdf file to gradescope.

Finally, remember to submit `ps3a.py`, `ps3b.py` and your pdf file for problem 3 to moodle.

1. **Computer Player** In this problem you will write a computer player for the word game that you implemented in problem set 2. The computer player will generate candidate words using Monte Carlo simulation. You will explore the relationship between the Monte Carlo parameters and the quality of the hints provided.

   (a) Implement `choose_word`. Using the framework provided, implement the function `choose_word` which takes a hand and N – the number of Monte Carlo candidates to generate and returns the best scoring word and the score of that word.

   The function should only return valid words. If no valid word can be found, it should return `None` for the word and 0 for the score.

   (b) Implement `play_mc_hand`. Using the framework provided, implement the function `play_hand` using the `choose_word` function. It should take the hand and N. It should return (1) a list containing the words played and (2) the total score of the sequence of words played.

   (c) Update `play_hand`. Update the branching logic in `play_hand` so that if the user enters '?' the function `choose_word` is run with $N = 100$. It should simply print the suggested word and the corresponding score for the user and then allow them to continue play. Make sure you handle the situation where a valid word is not found – this is not an error condition.

(d) Implement `play_n_mc_hand`. using the framework provided, implement `play_n_mc_hand`. This function takes (1) `hand`, (2) `N`, (3) `n`, where `n` is the number of times the same hand is played and each word is selected from `N` candidate Monte Carlo samples. The function should return (1) list of word lists (2) a list of scores.

2. **Reproducible Monte Carlo**

   You will write unit tests for your Monte Carlo computer player. The Monte Carlo player randomly samples plays, so you will need to set the random number generator seed for reproducible results.

   Implement `test_mc_player`. Write a function called `test_mc_player`. The function should test the following hands three times and produce the same score each time.

   1. `helloworld`
   2. `UMasswins`
   3. `statisticscomputing`

   The function should take three parameters: (1) the hand, (2) the number of Monte Carlo samples, and (3) the random number seed. It should return True if the test is passed and False if not.

3. **Visualizing Hands** Is the best hand a far-and-away best or is it only marginally better than many other hands? In order to answer this question we need a way to visualize the set of possible scores from our computer players. There are two variables in the MC player function - N: the number of MC samples to choose for each round and n the number of times to play the hand.

   (a) Write a function called `plot_mc_hand` that takes the hand `helloworld`, a number of Monte Carlo samples N, and a number of times to play the hand n and generates a plot that shows a histogram of the scores produced by the Monte Carlo samples of playing the hand. be sure to add a title to your plot so we know what you are showing.

   (b) Show how the distribution of scores for $N \in \{1, 10, 100\}$ and $n \in \{10, 100\}$.

   (c) What do you notice about the maximum score and the distribution of scores for the hand as these MC parameters change?