

main

September 29, 2024

1 Midterm 1, Spring 2024: Ranking Poker Hands

Version 1.0.1

History: - 1.0.1: Fix typo. - 1.0.0: Initial release.

All of the header information is important. Please read it..

Topics, number of exercises: This problem builds on your knowledge of conditional logic, implementing math as code, and Python's representation of numbers. It has 8 exercises, numbered 0 to 7. There are 15 available points. However, to earn 100% the threshold is 10 points. (Therefore, once you hit 10 points, you can stop. There is no extra credit for exceeding this threshold.)

Exercise ordering: Each exercise builds logically on previous exercises, but you may solve them in any order. That is, if you can't solve an exercise, you can still move on and try the next one. Use this to your advantage, as the exercises are **not** necessarily ordered in terms of difficulty. Higher point values indicate more complicated exercises which may be more time consuming to solve.

Demo cells: Code cells starting with the comment `### define demo inputs` load results from prior exercises applied to the entire data set and use those to build demo inputs. These must be run for subsequent demos to work properly, but they do not affect the test cells. The data loaded in these cells may be rather large (at least in terms of human readability). You are free to print or otherwise use Python to explore them, but we did not print them in the starter code.

Debugging you code: Right before each exercise test cell, there is a block of text explaining the variables available to you for debugging. You may use these to test your code and can print/display them as needed (careful when printing large objects, you may want to print the head or chunks of rows at a time).

Exercise point breakdown:

- Exercise 0: 1 point(s)
- Exercise 1: 1 point(s)
- Exercise 2: 2 point(s)
- Exercise 3: 1 point(s)
- Exercise 4: 3 point(s)
- Exercise 5: 3 point(s)
- Exercise 6: 2 point(s)
- Exercise 7: 2 point(s)

Final reminders:

- Submit after **every exercise**