Project Assignment #1 Analysis

Name: Ashley Indar

Background: My first project was based on intellectual curiosity of The Random Function covered in Week 1 of my Python class. As someone who struggles with coding, I wanted my assignment to be simple so I can spend a lot of time understanding the ins and outs of utilizing the various functions in Python to create a fully functional and bug free program. In doing so, I was able to achieve the goal of my project, which was to gain a more sound understanding of Python, to continue growing my skill-sets to develop more intermediate projects down the line. To accomplish this, I tested the random function to see if the method to selecting numbers in a guessing game, by playing it a specific number of times. This interactive guessing game asked the player to guess a number between 1-101, and was played in 3 rounds of 25 guesses each to prove my hypothesis incorrect.

Hypothesis: I had to tailor my hypothesis to the function of my project. Once I began writing and running my code, I realized I desired to test the method of probability using the random function. Initially, my intention was to write a program to test whether any given integer will be produced more than once, however I wanted my project to be based on user input rather than strictly computer generated. My hypothesis was based on the Probability and Random Variables concept I learned in my Discrete Structures in Math class last semester. This concept outlines that the probability (or probability distribution) of a certain event taking place can be calculated based on dividing the number of desired outcomes by the number of possible outcomes. The equation is as follows:

$$Probability = \frac{desired\ outcomes}{possible\ outcomes}$$

Ultimately, after playing the game 3 sets of 25 times, my hypothesis was proven to be true, which is that there is a method to guessing the correct number in under 25 tries, using the concept of probability. In my simulation, the player chose the midpoint value between the maximum and minimum possible integer as their first guess. Once indicated whether to guess higher or lower, the player guessed the midpoint between their initial guess and the maximum or minimum, respectively. For example, for their first guess the player guessed 50 (as the maximum value was 101). If the computer indicated for them to guess lower, they would guess 25, as that is the midpoint between 1 - 50. They then repeated this process until they guessed correctly. In conclusion, my hypothesis was correct in that there is a method to guessing the correct number even using the random function in Python.

Expanding Hypothesis: Initially, I thought my sample set was too vast and immediately decreased it. However, I quickly saw that was not an efficient means to testing my hypothesis. I decided to use my initial sample set of 101 which yielded the expected result. I can further test the limits of my hypothesis by increasing my sample set size, or customize my code in such a

way to require the user to select a value from an unknown array of numbers. This in itself however, will pose it's own issues to be worked through.

Data: The purpose of this game was to have the player enter a number between 1-101. They had 25 tries to do so. I imported from the random library to generate the correct number. The user was prompted to enter their number, and after every guess was told whether they needed to guess higher, lower, or if they were correct. A print statement was displayed if their guess was correct, and how many tries it took for them to accomplish that. Ultimately, the point of this simple code was to test probability, and prove the science behind the random function, using a sample set of 101 (integers between 1-101). The user played this game 3 sets of 25 times to prove my hypothesis, that there is a method to guessing the correct number in 25 tries or under.

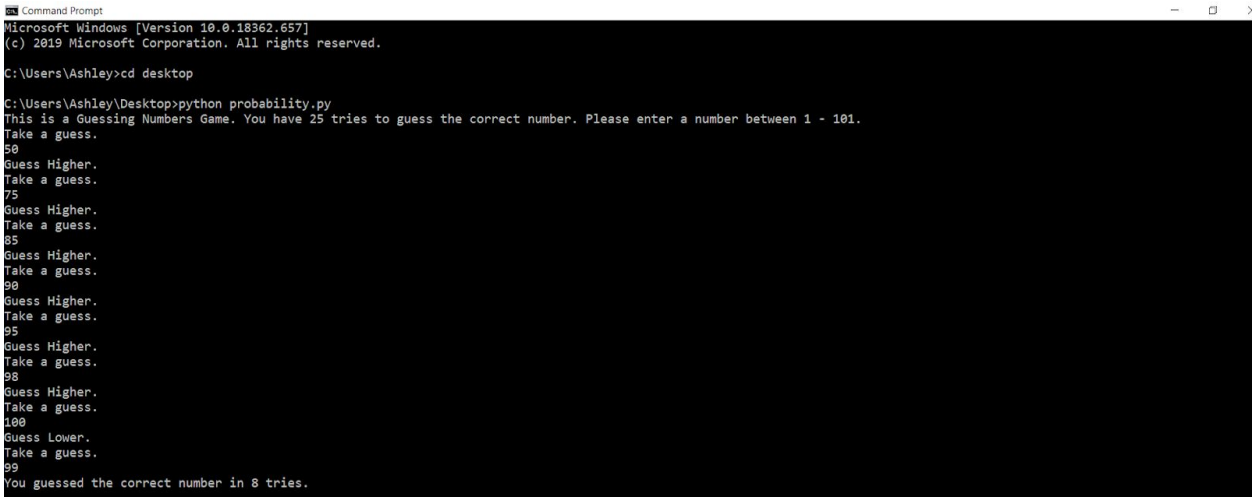
Analysis Steps:

1. Import the random library to begin my process and assign "num" to randomly select between 1-101.
2. Initialize variable 'guesses' to 0.
3. Set the range of numbers for the player to guess from 1 - 101 calling from the randint function and storing the correct guess value in num
4. Print statement to instruct the user how to play the game and prompting them to enter their first guess.
5. Finally, if they do not guess the correct number in 25 tries, notify them that their guesses were wrong, and what the correct number was
6. Utilize a while loop and if statements to indicate whether the user needs to guess higher, lower, or if their guess is correct and how many tries it took for them to accomplish that.

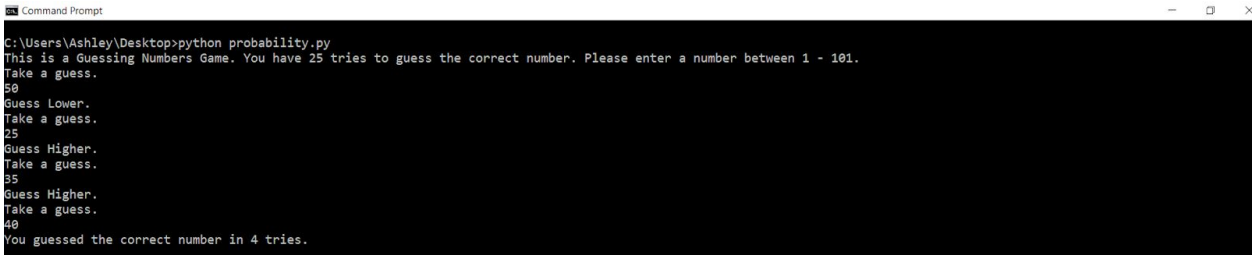
Roadblocks & Reflections: My main roadblock and challenge in this project was that I discovered a deeper and more intriguing claim to test during my research process (as projected). As I tested the vast computational concept of probability which contains many gray areas, I had to reconfigure my overall project to tailor it to my new vision. Tailoring my hypothesis to better suit the fact that I was testing probability, gave me better clarity on how to test this concept further if I desire to do so in the future. My only other challenge was steering clear of simple syntax errors and correcting any that I encountered. If I were to do it again, I would import from the graphics library to create and use graphics to make gameplay more engaging for the user while testing this concept.

Gameplay: Please see below for the player's input and method to guessing the correct number in each simulation.

Attempt 1:



Attempt 2:



Attempt 3: