

PHSX815_Project1:

Generating and analyzing a dice-roll for a D&D campaigns

Johnpaul Mbagwu

1. Introduction

Millions of people have played the well-known tabletop role-playing game Dungeons Dragons (D&D) for many years Klasturp (2006). In this game, players construct characters and go on quests in a fantastical setting full of magic, monsters, and intrigue. The use of dice to decide the results of the characters' actions is one of the game's fundamental components.

Different types of dice are used in DD for various purposes. The polyhedral dice set, which has a four-sided die (d4), a six-sided die (d6), an eight-sided die (d8), a ten-sided die (d10), a twelve-sided die (d12), and a twenty-sided die C. Basilan et al. (2023), is the most popular variety of dice used (d20). The player rolls a number of these dice to determine the outcome when a character performs an action with an unknown outcome, such as making an attack or attempting to cast a spell.

The character's skills and the situation surrounding the action are what determine how many dice are rolled. The "face" of the dice being rolled is this number. A character might roll a d20 for the attack roll when making a sword attack and then apply their strength bonus to the result. The assault is successful if the sum is more than the target's armour class or equal to it. The outcome of a D&D game can be significantly influenced by the roll of the dice, which gives each encounter a sense of chance and excitement.

2. Hypotheses to Explain why a face number of dice is rolled

In D&D campaigns, there are various theories that can explain why a "face" number of dice is rolled: Rolling a "face" number of dice in D&D is primarily done to mimic chance and randomness in the game. This makes each encounter more unpredictable and keeps players on their toes. The number of dice rolled can be used to indicate a character's traits and skills, such as strength, intellect, or dexterity. When making an attack roll, for instance, a character with a high strength score would roll more dice, whereas a character with a high intelligence score might roll more dice when making a spellcasting check. The game designers can balance the game mechanics and make sure that no one character or player has an undue advantage by rolling a "face" number of dice. This keeps the game interesting and difficult for all players.

To Promote Role-Playing: The use of dice in D&D promotes imaginative play and role-playing among players. The game allows players to make decisions and conduct actions using the skills of their characters and the results of a dice roll, which aids in bringing the narrative to life. D&D campaigns use the "face" number of dice to simulate chance, depict character skills, balance game mechanics, and promote role-playing, among other things.

3. Code and Experimental Simulation

This Python program simulates dice rolls according to a predetermined probability distribution for each dice face. The user can select the simulation's parameters, including the seed

for the random number generator, the number of dice rolls in each experiment, the number of experiments, and the chance of each face, using a number of command line arguments. To create random integers, the code imports a special "Random" class from a different file. If the '-output' flag is provided, the user has the option of saving the simulation's results to a file. When the '-h' or '--help' flags are used, the script outputs a help message.

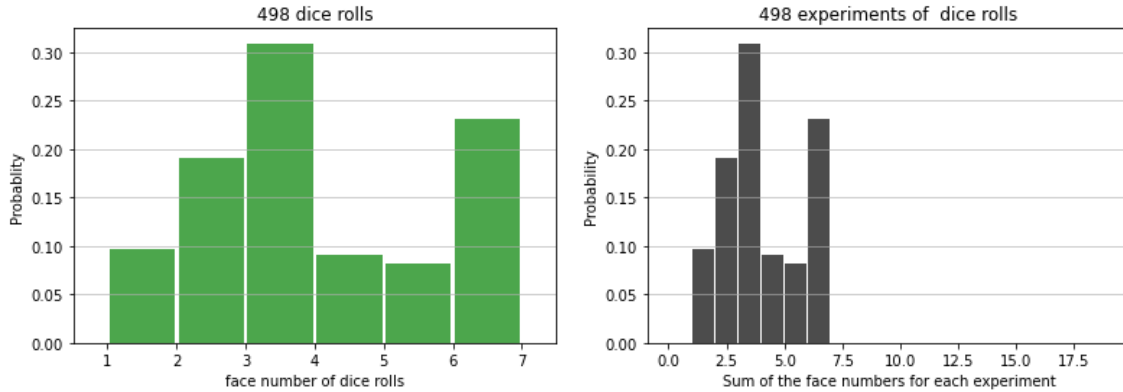


Figure 1: The faces of the dice are represented on the x-axis in this illustration, and the probability of each result is represented on the y-axis. The probability of each face appearing is shown by the histogram on the left, and the probability of the total number of faces appearing in each experiment is shown by the histogram on the right. This can be used to determine the likelihood that a particular result will occur when throwing many dice for the D&D campaigns

4. Analysis

The code is a straightforward Python script that analyzes dice rolls statistically using the numpy and matplotlib libraries. To determine whether the user has supplied a -input flag or --help flag, the script first parses the command-line arguments. The script reads the file designated following the -input flag if it is present. The script outputs usage details if the --help or -h flag is present or if the -input flag is missing. The script then uses the load txt function in numpy to load the data.txt file's contents before flattening it into a 1-D array with the help of the flatten function. Additionally, it uses the sum function along the axis=0 to calculate the total of the face numbers for each experiment (the sum of elements in each column).

The script then uses the hist function in Matplotlib to produce two histograms. The categorical probability distribution of the face numbers on dice rolls is shown in the first histogram, and the categorical probability distribution of the sum of face numbers is shown in the second histogram, for each experiment. With x and y labels, titles, gridlines, and limits on the x-axis, the script formats the plots. The script then shows the plots and waits for the user to close them.

5. Conclusion

This code creates two histograms using a Python script. The categorical probability distribution of a set of dice rolls' face numbers is shown in the first histogram. The categorical probability distribution of the sum of the face numbers from a set of experiments, each trial consisting of a set of dice rolls, is shown in the second histogram. These histograms' data are taken from a file called "data.txt." The data in the file should be in the form of a numpy array. The libraries sys, math, numpy, and matplotlib.pyplot are all used in the code.

The code provides the user a few ways to input data, such as the "-input" flag, which specifies the name of the input file, and the "-h" or "--help" switches, which print a usage statement. Only if the script's name is "main" will the code be configured to execute. When a script is imported as a module in another script, this typical technique is used to stop the code from running. The matplotlib.pyplot library's plt.hist() function, which accepts the data and some parameters like the number of bins and the color of the bars as inputs, is used in the code to create the histograms. The x and y labels, the title, and the grid are just a few formatting options that are determined by the code for the plots. Using the plt.show() function, the user can see the produced graphs.

References

- Ma Leticia Jose C. Basilan, <https://orcid.org/0000-0003-3105-2252>, Maycee Padilla, and <https://orcid.org/0000-0001-5025-12872>, maleticiajose.basilan@deped.gov.ph, maycee.padilla@deped.gov.ph, Department of Education- SDO Batangas Province, Batangas, Philippines. Assessment of teaching english language skills: Input to digitized activities for campus journalism advisers. *International Multidisciplinary Research Journal*, 4(4), January 2023.
- Lisbeth Klasttrup. Death matters: Understanding gameworld experiences. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '06, page 29–es, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933808. doi: 10.1145/1178823.1178859. URL <https://doi.org/10.1145/1178823.1178859>.