Simulation of a Six-Sided Dice: Roll and Analysis

Gbenga Agunbiade[1]

[1]Department of Physics and Astronomy, University of Kansas, Lawrence, KS 66045

gsagunbiade@ku.edu

Abstract. *This project uses a Python algorithm to carry out a frequency analysis of throwing a die. Imaging a 6-sided single dice, the score of this dice ranges from 1 to 6 and it can be any of the 6 sided, with each value having an equal probability of 1/6. This dice experiment is simulated using a computer algorithm to simulate throwing a die in 383 times and recording the number of times each of the 6 sides appeared. The frequency of each side can also be obtained in percentage. The code produces two histograms using the Matplotlib library. The first histogram shows the distribution of the face numbers of several dice rolls, and the second histogram shows the distribution of the sum of the face numbers for each experiment. Both histograms are plotted as probability distributions.Theoretically, each side is expected to have similar frequency of 1/6 probability.*

1. Introduction

A dice roller is a tool used to randomly generate numbers by simulating the roll of a die. This allows a simple and convenient way to generate random numbers, Although, dice has different number of sides, but the most popular dice is a six-sided one. Dice rollers are useful in various applications, such as simulation and statistical analysis, where random numbers are required, they are also found useful in many tabletop games, such as board games and role-playing games, where the outcome of a game is determined by the roll of a die. A dice analysis provides a way to obtain the similarity of two probability distributions. It is also known as the Dice similarity coefficient (DSC). The DSC is calculated as the ratio of twice the sum of the product of the probabilities of two events, to the sum of the individual probabilities of each event:

$$DSC = 2 * (P(A) * P(B))/(P(A) + P(B)) \tag{1}$$

where P(A) and P(B) are the probabilities of events A and B, respectively. The DSC has a value between 0 and 1, having a value of 1 means a perfect overlap between the two events while value of 0 shows that the two events are completely independent and have no overlap.

2. Dice Roll

The Dice roll uses a Python script that simulates rolling a certain number of dice, each with a specific set of probabilities for each of the faces. The user can provide various input parameters to the script through the command line such as the seed for the random number generator, the number of dice and the number of experiments to run, as well as the probabilities of each face of the dice. The script also allows for the option to output the results to a file. The script uses the Random class defined in the Random.py file. The class implements a simple random number generator. In this script, the input parameters are processed using the sys.argv list, which contains the command line arguments passed to the script. The default values for the input parameters are specified in the beginning of the script. The user can provide custom values for these parameters by providing the appropriate flag followed by the value. For example, to change the seed for the random number generator, the user can provide the -seed flag followed by the desired seed value. The script checks for the presence of each flag and if it is found, it updates the corresponding value. If the user provides an invalid value for any of the parameters, the script ignores that value and uses the default value instead. The script also provides a help message that can be displayed by passing the -h or –help flag to the script. Finally, the script uses the random instance of the Random class to generate random numbers and simulate the dice rolls. The results of each experiment are stored in a NumPy array, and the mean, variance, and standard deviation are calculated and displayed.

3. Dice Analysis

The dice analysis code is written in Python and using the NumPy and Matplotlib libraries. This code picks a filename as input through the command line with the option -input and loads the data in the file using np.loadtext. The data is then flattened and the sum of the elements in each row is computed and stored in add. The code produces two histograms using the Matplotlib library. The first histogram as presented in figure 1 shows the distribution of the face numbers of several dice rolls, and the second histogram presented in figure 2 shows the distribution of the sum of the face numbers for each experiment. Both histograms are plotted as probability distributions. Finally, the code shows the figures created by the histograms and the program only ends when the figures are closed.

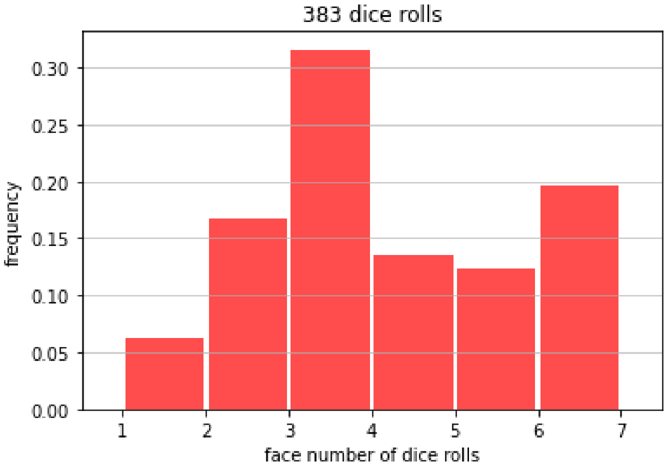

Figura 1. The distribution of the face numbers of 383 dice rolls

4. Conclusion

Dice analysis in probability is usually useful during information retrieval, text classification, and document clustering, where it is important to determine the similarity of documents or text items based on the distribution of their content words. It is also used in

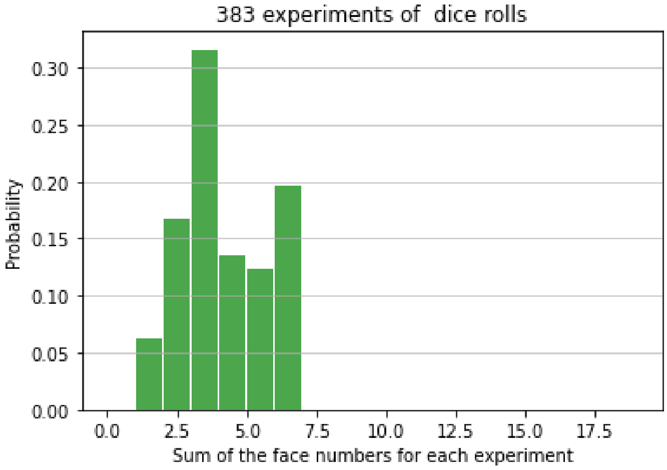

Figura 2. The distribution of the sum of the face numbers for each experiment

computer vision and image processing, where it can be used to evaluate the similarity of images or regions in images. In Summary, dice analysis presents a simple and effective way to get the similarity of two probability distributions and is widely used in many areas due to its ability to provide a quantitative measure of similarity.

5. Reference

Galton F. (1890) Dice for Statistical Experiments, Nature volume 42, pages13–14.