Project 3: Estimating Dice Probabilities Using Randomly Generated Experimental Data
PHSX815

David Coria

April 12, 2021

1 Introduction

For Project 3, I have chosen to study the probabilities associated with rolling a six-sided dice. If you were to roll a dice enough times, you should be able to calculate the probability associated with each face of the dice. In a fair, unbiased dice, the probabilities should be equal–or as close to equal as possible. The more times you roll the dice, the more accurate your estimated probability becomes. In a dice where each face has a different probability, you should still be able to estimate the true probabilities for each face if you roll the dice enough times. That being said, it is important to be able to determine the probability of a certain event happening from nothing but data from previous events. So, for this project, I have created a program to roll a biased dice a certain number of times as well as a program which takes these results and computes the probability of rolling each side.

2 Code and Experimental Simulation

The goal for this dice rolling experiment is to roll a biased dice a certain number of times, save the results, and later determine from this data the probability of the dice landing on a certain face. So, the first part of this projects focuses on developing code to simulate rolling a biased dice a certain number of times. The dice is biased, which means that the probabilities associated with each face are not equal.

2.1 DiceRoll.py

This program is used to simulate a dice rolling experiment. The script accepts user input for -Nrolls, the number of times the biased dice is rolled in each experiment, and -Nexp, the total number of experiments. Thus, the biased dice I developed is rolled a total number of $Nrolls * Nexp$ times. The biased die is made such that the probability of the first face $p1$ is determined from a random uniform distribution i.e. [0, 1]. The probability of the second face $p2$ is determined similarly, but this time using a random uniform distribution from [0, 1 - p1]. The probability of the third face $p3$ is again determined similarly, but this time using a random uniform distribution from [0, 1 - p1 - p2]. The last 3 sides are a continuation of this process so as to ensure the probabilities of all 6 sides sum to 1. Finally, I randomly shuffle the list of probabilities so that it is not always face 6 that is assigned the smallest probability. The script then performs the experiments (Nexp times with Nrolls in each experiment) and stores the data in an array which is then saved as a .txt file as specified by the user. If no output filename is specified, the script will simply print the array of dice rolls. Using the data stored in this .txt file, we may proceed to the analysis portion of the project where we determine the probabilities associated with each side of the biased die.

```
In [488]: rolls[0,:]
Out[488]:
array([2, 3, 3, 2, 2, 4, 3, 3, 3, 2, 3, 2, 2, 3, 3, 2, 3, 3, 3, 3, 3, 2,
       2, 3, 3, 3, 3, 3, 3, 4, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 4])

In [489]: rolls[1,:]
Out[489]:
array([3, 2, 3, 4, 3, 3, 3, 2, 3, 3, 3, 4, 3, 3, 3, 5, 1, 4, 2, 2, 2, 3,
       2, 2, 5, 2, 3, 3, 2, 2, 3, 2, 2, 3, 4, 4, 6, 3, 6, 3, 4, 3, 2, 3,
       3, 3, 2, 3, 3, 2])

In [490]: rolls[2,:]
Out[490]:
array([2, 3, 4, 5, 3, 3, 2, 2, 3, 3, 2, 3, 4, 3, 2, 2, 2, 3, 3, 6, 4, 2,
       3, 6, 3, 4, 3, 3, 3, 2, 3, 3, 2, 3, 3, 3, 3, 2, 6, 3, 5, 3, 2, 3,
       4, 4, 3, 2, 4, 2])

In [491]: rolls[3,:]
Out[491]:
array([3, 5, 3, 3, 3, 4, 3, 2, 2, 5, 3, 2, 3, 3, 3, 3, 3, 3, 4, 1, 2, 2,
       2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 4, 3, 3, 3, 1, 3, 3, 3,
       2, 3, 3, 3, 4, 5])
```

Figure 1: Four of one hundred dice rolling experiments. Each experiment in this trial consisted of 50 dice rolls.

3 Analysis

This portion of the project receives the .txt file storing the results of the experimental dice rolls as input and uses histograms to count the number of rolls associated with each face to estimate the probability of the biased die landing on each face.

3.1 DiceRollAnalysis.py

First, the script reads in the file storing the dice rolling experiments from the previous program. Using numpy's built in histogram function, we can count the number of times we rolled each face– for each experiment ($Nrolls$ number of rolls) and for the entire set of experiments ($Nexp * Nrolls$ number of rolls.) The probabilities can then be calculated within each experiment AND for the whole set of experiments. The probability for each face for each experiment is thus $P_{exp} = \frac{face\ counts}{number\ of\ rolls}$ and the probability for the entire set is $P = \frac{face\ counts}{total\ rolls}$ where $total\ rolls = Nexp * Nrolls$.

Next, I focus on determining the uncertainties of the probabilities calculated for each face. To do this, I fit a Gaussian to the histogram of probabilities for each face. The parameters of the Gaussian carry the information we're after. The mean of the each Gaussian distribution represents the probability of each face we determined earlier, and the standard deviation is then the uncertainty associated with each probability measurement. These Gaussian fits are shown in Figure 2 at the end of this write-up.

4 Conclusion

Originally, I thought my parameter of interest would be the probability of each side of the die. However, another interesting parameter would be the number of rolls and how it affects the precision of the probability measurements. I wanted to automate this same analysis for a certain number of experiments with each consisting of a different number of rolls, however, I wasn't quite sure how to do that without manually repeating the code I present here. This way, I could've better answered the question of how well probabilities can be estimated from that data and how the uncertainty would change with the

number of rolls in an experiment. Ultimately, the analysis I DID manage to come up with can calculate the probabilities associated with each face of the die– even though I was unable to test what number of rolls is sufficient to reconstruct the original, randomly generated probabilities of each side. This code could, in theory, be expanded for dice with many more sides.
Here are the probabilities determined for each face of the die:

$$P1 = 0.021 \pm 0.02$$
$$P2 = 0.23 \pm 0.05$$
$$P3 = 0.603 \pm 0.07$$
$$P4 = 0.084 \pm 0.04$$
$$P5 = 0.036 \pm 0.02$$
$$P6 = 0.026 \pm 0.03$$

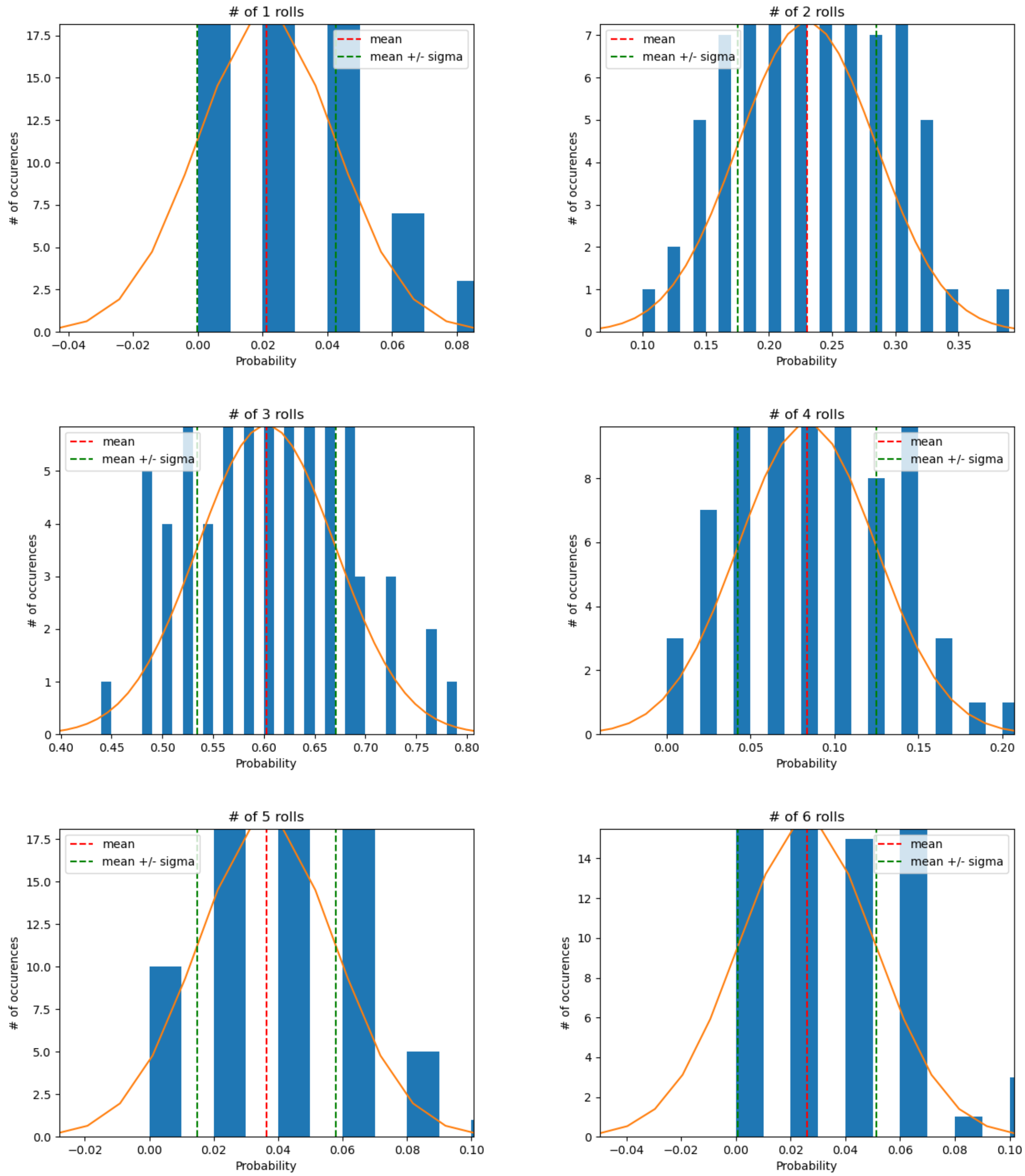

Figure 2: Each plot here is the histogram of probabilities for each face of the die. I fit a Gaussian to each to reproduce the probabilities we determined earlier. These probabilities are shown in the Conclusion section. The uncertainties associated with each probability are simply the standard deviations as determined by the Gaussian fit of the histogram as seen here.