`PHSX815_Project2`: Dice Rolling with Rayleigh Sampling

Kurt Hamblin

March 22, 2021

1 Introduction

Sampling from other, non-categorical, distributions is a great way to create biased dice in a random manner. In the previous project, I incorporated dice rolling into my monopoly simulation, but the focus was not on the actual dice rolling. Here, I focus specifically on the dice rolls themselves, and compare rolling dice by sampling from a Rayleigh distribution to rolling fair dice.

This paper is organized as follow: Sec. 2 provides an overview of the utilized code and experimental design, and Sec. 3 presents the results, analysis, and conclusions.

2 Overview of Code and Experimental Procedure

For this experiment, we must implement both dice rolling and a Rayleigh distribution to sample from. The dice rolling code is the same as used in project (1), but we will go over it in more detail below. Additionally, we use the implementation of the Rayleigh distribution from Week 2.

2.1 Dice Rolling

To simulate dice rolling, we add a method to the Random class, `Random.roll_die(Nsides, weights)`, that returns an integer from 1 to Nsides. We include the ability to roll a biased die with the inclusion of "weights," which allow the user to specify the exact probability of each side of the die being rolled. The algorithm divides the space from 0 to 1 into slices corresponding to each side of the die (either by splitting 0 to 1 evenly, or using the weights as the boundaries for the 'slices'), and then generates a random float from 0 to 1, and returns the side that this random float corresponds to.

This method can be used for any N-sided die, and does not make use of any `Numpy.random` methods.

2.2 Rayleigh Sampling

We use the `Random.Rayleigh()` method to sample from a Rayleigh distribution in order to decide the side probabilities. To do so, we sample the distribution N times (for a N-sided die), and divide each sampled number by the sum of the sampled numbers. This way. all the numbers total to 1, and we can consider each number as now being the probability of each individual side. These probabilities, or "weights," are then passed into the `Random.roll_die()` method to roll a biased die. We will call a die, with weights sampled from the Rayleigh distribution, a Rayleigh-biased die.

2.3 Experimental Design

For this project, we choose to simulate 10^4 experiments for both the biased and unbiased dice 6-sided dice, but we will vary the the number of rolls per experiment, simulating for 30 rolls and 500 rolls per experiment. We settled on these numbers after testing, and determined them to be sufficiently different and useful for the analysis in this project. We aim to determine the power of a log-likelihood test using each of these sample sizes, where we test the following hypotheses:

Hypothesis $\mathbf{H_0}$: The die is fair

Hypothesis $\mathbf{H_1}$: The die is Rayleigh-biased

2.4 log-likelihood ratios

For the analysis in this project, we choose to make use of log-likelihood ratios (LLRs) to quantitatively compare the hypotheses. In our analysis, we calculate the LLRs using the following for the Null hypothesis (fair n_s-sided die):

$$P(x|H_0) = 1/n_s \tag{1}$$

For the probability of data X, given $H1$, we iterate through the simulated data and total the frequency counts of each side of the die, and weight each total by the total number of rolls, so that we have (or rather approach) the underlying categorical distribution for a die with weights sampled from a Rayleigh distribution. Then:

$$P(x|H_1) = \frac{\text{Num of x in Rayleigh-biased data}}{\text{total number of rolls}} \tag{2}$$

Then, for any given experiment with N rolls, the LLR becomes:

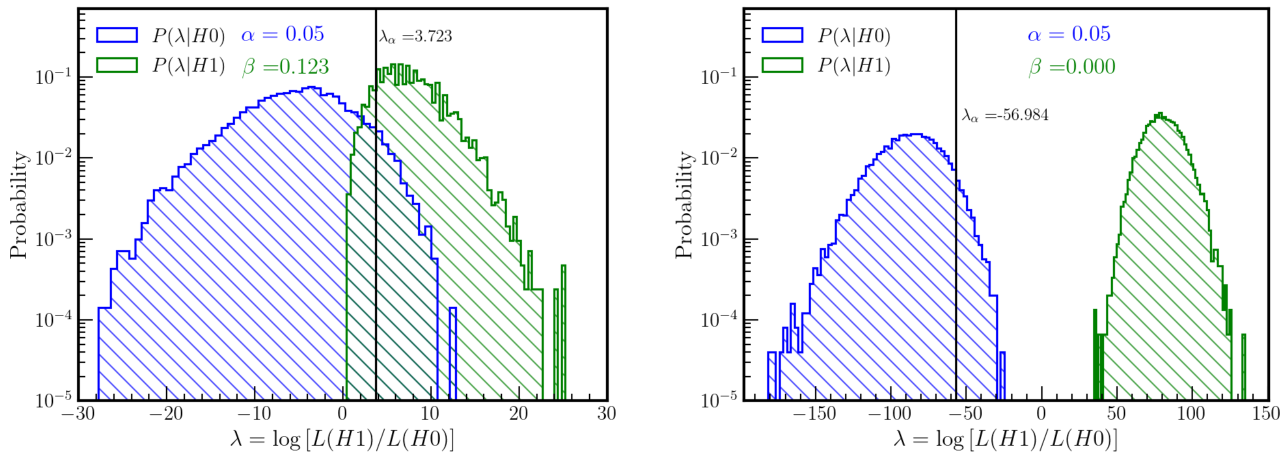
$$LLR = \sum_{i=0}^{N} \ln\left(\frac{P(x_i|H_1)}{P(x_i|H_0)}\right) = \sum_{i=0}^{N} \ln\left(n_s P(x_i|H_1)\right) \tag{3}$$

Where $P(x_i|H_1)$ is found from Equation (2).

3 Results and Conclusions

We adopt a significance level of $\alpha = 0.05$, and will find the corresponding β value for each set of experiments. We present the results in Figure (1). We first note that or large numbers of rolls per experiment, β becomes 0, and thus the test becomes perfectly powerful.This result makes sense, since for many rolls, the distribution of dice rolls samples from the Rayleigh distribution is expected to look much different than that of a fair die. However, for the experiment set with 30 rolls per experiment, we find $\beta = 0.123$, so the test power is now only 87.7%. With this test then, we would fail to correctly accept that the die is fair 12.3% of the time

For both experiment sets, we also find the critical likelihood λ_α, which allows the results of future experiments to be assessed. If the log-likelihood ratio of a future experiment is greater than λ_α, we can reject H0 in favor of H1 with 95% confidence.



(a) 30 Rolls per Experiment

(b) 500 Rolls per Experiment

Figure 1: Log-Likelihood ratio plots for sets of 10^4 dice rolls with 30 rolls per experiment (left) and 500 rolls per experiment (right). It is immediately evident that, for large numbers of rolls per experiment, β becomes 0, and the LLR test becomes 100% powerful.