**Lab 6: Risk Dice**

**CS 122L • 15 (+6 Extra credit) Points Total**

**Objectives**

* Calculate the probabilities of each possible outcome for a "Risk"-style dice battle and graph the outcomes with a pie chart.
* Practice loops, conditionals, and graphs.

**Overview**

When you play the board game "Risk", you move armies around a map of the world and conquer new territories in battles against the other players. The battles are resolved by both sides rolling dice. Depending on the situation, the outcome may be weighted towards one side or the other, but due to the luck of the dice nothing is certain. For this project we want to determine the probabilities of success, failure, or somewhere in-between for a particular Risk battle, then graph those probabilities on a pie chart.

**Identify and Explore the Problem**

Let us consider a typical case in Risk: one player is attacking with two units, while the other player is defending with two units. Here are the rules for that battle and the possible outcomes:

* There are two possible points (victories) for one battle.
* The attacker rolls 2D6 (2 six-sided dice) and the defender rolls 2D6.
* The highest die the attacker rolls is paired with the highest die the defender rolls, and the next-highest die the attacker rolls is paired with the second-highest die (the only one remaining) the defender rolls.
* For each pair of dice, if the attacker's die roll is higher than the defender's die roll, the attacker scores a point. Otherwise, the defender's die is higher than or equal to the attacker's and the defender scores a point.

Write a MATLAB program that calculates the probabilities of the following outcomes in the scenario described above:

* Outcome 1: Attacker wins both victories.
* Outcome 2: Defender wins both victories.
* Outcome 3: Attacker and defender each win one victory.

The results should be displayed in a pie chart that shows the probability percentage of each outcome.

**Create a Mathematical Model**

Consider: the number of possible results of rolling a single D6 is 6. The probability of rolling a 1 through 6 on a D6 is 6/6, or 1.0 (100%). The probability of rolling a 1 or a 5 is 2/6, or 0.33 (33%).

The probability of any outcome for any similar problem is the number of times the outcome occurs divided by the total number of possible results.

Instead of using statistics formulas or random numbers to calculate the answer for the Risk dice, we can use the power of the computer to iterate through each possible result that can be rolled on 4D6, see which of the three outcomes that particular result falls into, and then divide each result by the total number of possible results.

**Identify and Implement a Computational Method for Solving the Model**

A single loop is sufficient to obtain each possible result of rolling a single die. For example, here is a MATLAB program that will calculate the probability of a 2 or a 5 being rolled on a D6:

count = 0;

for d = 1:6

if d==2 | d==5

count = count + 1;

end

end

probability = count / 6;

Two loops would be required to try every possible combination of rolling two dice. For each value 1..6 that the first die could roll, the second die could roll 1..6 as well, giving 36 possible combinations (62 = 36). For example, here's a program that calculates the probability of a roll on 2D6 adding up to 7:

count = 0;

for d1 = 1:6

for d2 = 1:6

if d1+d2 == 7

count = count + 1;

end

end

end

probability = count / 36;

**Implementation**

Download the Lab6 folder from Bb Learn. As usual, move it to your cs122 folder, cd into the new Lab6 folder and add the Lab6 folder to the path.

Open the Lab6.m file. For this program, we want to look at every possible combination of 4 dice being rolled. Here's an algorithm that will work:

* Look at every possible combination of 2D6 attack dice with 2D6 defense dice. Hint: make 4 nested loops, each looping through values 1..6. There should be loops for "a1" and "a2" as well as "d1" and "d2". Inside the innermost loop, those four values will represent one of the 64 combinations of 4D6.
  + For the current combination, sort the die values for the 2D6 attack dice into order and the die values for the 2D6 defense dice into order so that we can pick out the highest two of each. Hint: make a vector out of the values you want to sort and use the "sort" function; e.g. "attacker = sort([a1 a2]);" would set "attacker" to be a vector containing the values of a1 and a2 arranged in order from lowest to highest.
  + Pick out the first and second highest for both the attacker and the defender. These will be the values at the ends of the sorted vectors.
  + If the first highest attack die is greater than the first highest defense die *and* the second highest attack die is greater than the second highest defense die, increase the count of how many times Outcome 1 has happened.
  + If both of the attacker's highest dice are less than or equal to the defender's corresponding dice, add one to the number of times Outcome 2 has happened.
  + If both the previous conditions are false, add one to the count of Outcome 3
* Calculate the probabilities for each outcome. Store them in the variables attack\_wins, defense\_wins, and tie.
* Display the number of each outcome 1, 2, and 3 in a pie chart. Give it an appropriate title and a legend that identifies each pie slice as either "Attacker Wins 2", "Defender Wins 2", or "Both Win 1". (Hint: make sure you are using the probabilities calculated in the previous step).

. Hint: see the Wikipedia Risk entry for the probability values you should end up with!

**Extra Credit (up to +6 points)**

For extra credit, several additional .m files have been included. Each extra credit problem has its own associated .m file. Be sure to read the comments at the top of the file to ensure that you are using the appropriate variable names. The following are the extra credit problems, and the associated .m files:

* Lab6EC1: Three attack dice versus one defense die (two possible outcomes: attacker wins 1 or defender wins 1).
* Lab6EC2: Two attack dice versus one defense die (two possible outcomes: attacker wins 1 or defender wins 1).
* Lab6EC3: One attack die versus two defense dice (two possible outcomes: attacker wins 1 or defender wins 1).

**Deliverables**

* Your project report (see below)
* Your zipped Lab6 folder containing Lab6.m, and all extra credit files
  + Submit the extra files even if you did not complete them
  + If you are not sure how to zip folders, please ask your TA for assistance

**Project Report**

Below is the point distribution for required sections in the lab report. Be sure that each section is labeled clearly. Refer to the lab submission guidelines for details on what goes in each section.

|  |  |  |
| --- | --- | --- |
| **Section** | **Points** | **Notes** |
| 1.Task Description | 1 |  |
| 2.Learning Objectives | 1 |  |
| 3.Approach | 1 |  |
| 4.Mathematical Concepts | 1 |  |
| 5.Program Description | 2 |  |
| 6.Source Code | 7 | All code, including any for the extra credit, should go in this section. |
| 7.Code Execution Results | 1 | All pie charts, including any for the extra credit, should go in this section. |
| 8.Conclusions | 1 |  |
|  | **15 total** |  |

Submit your results in the correct place in Blackboard Learn (<http://bblearn.nau.edu>) by the due date.