Stat 220 Lab 2: Probability Calculation for Red Dragon Inn

The Red Dragon Inn is an online Dungeons and Dragons community, connecting players worldwide. In this lab, you will develop a tool that helps users calculate the probabilities of specific events occurring during gameplay, enhancing their strategic planning. For instance, users may wish to calculate the probability of the sum of two six-sided dice being 10 or more.

In Dungeons and Dragons, the notation mdn denotes the process of rolling m dice, each with n sides. Here, rolling two six-sided dice is expressed as $2d6$, and rolling three eight-sided dice is denoted by $3d8$. You are tasked with creating a utility to assist players in the following two scenarios:

A Calculating the probability of rolling m dice, each with n sides, and having the sum of the dice be greater than x. This calculation should only be implemented for $m = 1$ through 4 and n is at most 20.

B Determining the probability of rolling m dice, each with n sides, and having at least one die show a value greater than or equal to x.

Your task is to develop two functions, one for each scenario, where users can input values for m , n, and x to compute the required probability. Ensure to document your code well, with comments that effectively guide users through each step of the computation process.

Here's a hint to help you get started with computing probabilities in Python. The following code snippet shows how to calculate the probability of rolling a sum greater than or equal to a specific target using three six-sided dice (3d6):

```
import itertools
import pandas as pd
import numpy as np

# List out all possible rolls, in this case for 3d6
rolls = list(itertools.product(range(1, 7), repeat=3))

# Find the sum of all possible rolls
sums = np.array([sum(roll) for roll in rolls])

# Find the proportion of those sums that are greater than or equal to the target, in this case 14
np.mean(sums >= 14)
```

This code uses the `itertools.product` function to generate all possible combinations of outcomes for the dice rolls and then calculates the proportion of those combinations that meet or exceed the target sum. You can use this code to get started in the function you need for Scenario A.

Evaluate the performance of your tool using the following scenario:

A player is in a challenging situation where they have a choice. They can choose between the following challenges:

- Roll 1d20 and get a result of 17 or more
- Roll 3d4 and get a result of 10 or more
- Roll 4d6 and have at least one dice show a 6

Use your tool to compute the probabilities and report the best choice for the player.

Deliverables:

Submit a comprehensive report to Red Dragon Inn containing:

1. A well-documented code capable of computing the probabilities specified in scenarios A and B.
2. A walkthrough example showcasing the application of your code in resolving the situation delineated above. Be sure to highlight or emphasize the probabilities you find.

You can again complete this in a Python notebook with everything clearly labeled and formatted.