1. **Razonamiento de la estrategia de Juan:** La estrategia de Juan maximiza sus probabilidades de obtener un puntaje alto mientras minimiza el riesgo de obtener un puntaje bajo. Si obtiene 0 puntos, tiene una segunda oportunidad para mejorar su puntaje. Si obtiene 1, 2 o 3 puntos, puede optar por mejorar su puntaje con un segundo lanzamiento. Solo si obtiene más de 3 puntos se planta, lo que indica que está satisfecho con su puntaje actual.
2. **Probabilidades de obtener cada puntaje:** Para calcular las probabilidades de que Juan obtenga cierto puntaje 𝑘*k*, podemos enumerar todos los posibles resultados de lanzar dos dados y sumar las probabilidades correspondientes para cada puntaje 𝑘*k*.

Las probabilidades son las siguientes:

* + Para 𝑘=1: Hay dos combinaciones posibles: (4, 1) y (1, 4), cada una con una probabilidad de 1/36​, entonces la probabilidad total es 2/36​.
  + Para 𝑘=2: Hay cuatro combinaciones posibles: (4, 2), (2, 4), (3, 3) y (1, 5), cada una con una probabilidad de 1/36, entonces la probabilidad total es 4/36​.
  + Para 𝑘=3: Hay cuatro combinaciones posibles: (4, 3), (3, 4), (5, 1) y (1, 5), cada una con una probabilidad de 1/36​, entonces la probabilidad total es 4/36​.
  + Para 𝑘=4: Hay tres combinaciones posibles: (4, 4), (5, 3) y (3, 5), cada una con una probabilidad de 1/36​, entonces la probabilidad total es 3/36​.
  + Para 𝑘=5: Hay dos combinaciones posibles: (4, 5) y (5, 4), cada una con una probabilidad de 1/36​, entonces la probabilidad total es 2/36​.
  + Para 𝑘=6: Hay dos combinaciones posibles: (4, 6) y (6, 4), cada una con una probabilidad de 1/36​, entonces la probabilidad total es 2/36​.

1. **Estrategia óptima de María:** La estrategia de María dependerá del puntaje que Juan haya obtenido. Si Juan obtiene un puntaje bajo, María puede arriesgarse más en su primer lanzamiento para tratar de superarlo. Si Juan obtiene un puntaje alto, María puede optar por una estrategia más conservadora y tratar de superarlo con un puntaje seguro.

5. Simulación del juego:

Ejecutaremos la simulación del juego un número específico de veces (por ejemplo, 1000, 10000 o 100000 veces) y calcularemos la frecuencia relativa de cada evento (Juan gana, María gana, empate) para determinar las probabilidades aproximadas de cada resultado.

1. Determinar si María tiene una ventaja significativa:

Utilizando las simulaciones realizadas, compararemos las frecuencias relativas de cada evento para determinar si María tiene una ventaja significativa al jugar este juego. Si la frecuencia de que María gane es considerablemente mayor que la de Juan o la de empate, entonces podría indicar que María tiene una ventaja significativa en este juego.

Código:

El juego se desarrolla de la siguiente manera:

Ambos jugadores lanzan dos dados y suman los resultados para obtener un puntaje.

Si Juan obtiene 0 puntos en su primera tirada, puede tirar los dos dados de nuevo para mejorar su puntaje. En este caso, su puntaje final será el mayor entre el obtenido en la primera tirada y el obtenido en la segunda.

Si Juan no obtiene 0 puntos en su primera tirada, puede elegir tirar de nuevo el dado que no haya obtenido un 4 (o ambos dados si ambos obtuvieron 4) con el objetivo de mejorar su puntaje. Sin embargo, esto puede empeorar su puntaje. En este caso, su puntaje final será el obtenido en esta segunda tirada.

Si Juan obtiene 1, 2 o 3 puntos en su primera tirada, utilizará su segunda tirada de uno de los dados para intentar mejorar su puntaje.

Si Juan obtiene más de 3 puntos en su primera tirada, se quedará con su puntaje y no utilizará su tirada adicional.

Después de que Juan juegue, le toca a María. María seguirá las mismas reglas que Juan, pero tendrá la ventaja de saber cuál es el puntaje final de Juan.

El código proporcionado incluye funciones para simular el juego:

lanzar\_dados(): Simula el lanzamiento de dos dados y devuelve los resultados como una tupla.

juan\_juega(): Simula la estrategia de juego de Juan y devuelve su puntaje final.

maria\_juega(puntaje\_juan): Simula la estrategia de juego de María, dada la puntuación final de Juan, y devuelve su puntaje final.

simular\_juego(): Realiza una simulación completa de un juego entre Juan y María y devuelve el resultado (quién gana o si hay un empate).

simular\_n\_veces(n): Realiza una simulación del juego un número especificado de veces (n) y devuelve las frecuencias de los resultados (cuántas veces gana Juan, María o hay un empate) como un diccionario.

Finalmente, el código simula el juego un número de veces especificado (1000, 10000 y 100000 veces) e imprime las frecuencias de los resultados de cada simulación.