Optiver Prove It 3

Tadhg Xu-Glassop

September 2024

1 Solutions to the Problems

Theorem 1.1: Expected Number of Die Rolls

Consider a fair n sided dice, and let X be the number of rolls until the cumulative sum is at least n. Then,

$$\mathbb{E}[X] = \left(1 + \frac{1}{n}\right)^{n-1}.$$

Proof. Let X be the number of rolls until the cumulative sum of rolling an n sided die is at least n. Note that X has a minimum possible value of 1, wherein we roll n on the first roll, and a maximum possible value of n, where we roll a 1 for every roll. So, by definition,

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{k=1}^{n} k \cdot \mathbb{P}(X = k) \\
&= \mathbb{P}(X = 1) \\
&\quad + \mathbb{P}(X = 2) + \mathbb{P}(X = 2) \\
&\quad \vdots \\
&\quad + \mathbb{P}(X = n) + \mathbb{P}(X = n) + \cdots + \mathbb{P}(X = n) \\
&= \mathbb{P}(X > 0) + \mathbb{P}(X > 1) + \cdots + \mathbb{P}(X > n - 1) \\
&= \sum_{k=0}^{n-1} \mathbb{P}(X > k).
\end{aligned}$$

Thus, we aim to find an expression for $\mathbb{P}(X > k)$. To do this, we will count the possible number of ways such a scenario can occur.

Suppose we roll a die k times, and so let $d_1, d_2, \ldots, d_k$ be the results of each die roll. The event $X > k$ is equivalent to $\sum_{i=1}^{k} d_i = \alpha$ for all integers α such that $k \leq \alpha \leq n - 1$, as the latter means our die sum is not yet equal to n, and thus more die rolls are required.

Before we can apply the stars and bars method, we must recognise that $d_i \geq 1$ (there is no 0 on this die), and thus we must set $d_i := d_i + 1$ for the method to be valid. This gives the equation,

$$\sum_{i=1}^{k} d_i = \alpha - k,$$

njhk and so finding a solution to this equation is equivalent to distributing $\alpha - k$ stars amongst $k - 1$ bars. Hence, the number of solutions is,

$$\binom{\alpha - k + k - 1}{k - 1} = \binom{\alpha - 1}{k - 1}.$$

Using this, and recognising that total number of sequences from k die rolls is n^k we can find an expression for the probability that $X > k$ as follows,

$$\begin{aligned}\mathbb{P}(X > k) &= \frac{1}{n^k}\sum_{\alpha=k}^{n-1}\binom{\alpha-1}{k-1} \\ &= \frac{1}{n^k}\binom{n-2+1}{k-1+1} \\ &= \frac{1}{n^k}\binom{n-1}{k},\end{aligned}$$

where the second equality is a result of the Hockey-stick identity. Thus, we find that

$$\begin{aligned}\mathbb{E}[X] &= \sum_{k=0}^{n-1}\frac{1}{n^k}\binom{n-1}{k} \\ &= \binom{n-1}{0} + \frac{1}{n}\binom{n-1}{1} + \cdots + \frac{1}{n^{n-1}}\binom{n-1}{n-1} \\ &= \left(1+\frac{1}{n}\right)^{n-1}.\end{aligned}$$

□

Corollary 1.1: Limiting Value

The expected number of required rolls approaches e for large n, that is,

$$\lim_{n\to\infty}\mathbb{E}[X] = e.$$

Proof. Taking $n \to \infty$ on the result from Theorem 1.1, we have,

$$\begin{aligned}\lim_{n\to\infty}\mathbb{E}[X] &= \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^{n-1} \\ &= \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^{n}\left(1+\frac{1}{n}\right)^{-1} \\ &= e(1) \\ &= e.\end{aligned}$$

□

Bonus Exercise

You and your friend take turns rolling a six-sided dice and keep a joint running total.

The player who first gets the sum to 6 or greater gets paid out the sum.

Do you want to go first or second?

Hypothesis: First, let's make a guess for what the correct answer is. Using Theorem 1.1 and letting $n = 6$, we find $\mathbb{E}[X] \approx 2.16$. Considering payouts only when $X = 2$ and $X = 3$, as we expect most games to last between 2 to 3 rolls, we can approximate this expected value, $\mathbb{E}[X] \approx 2.15 = 2 \cdot 0.85 + 3 \cdot 0.15$, that is, we expect roughly 85% of the games to last 2 rolls, and the rest to last 3 rolls. Now, for the expected payout, we can estimate the expected payout for $X = 2$ and $X = 3$ by multiplying the number of die rolls by the expected value of a die roll, 3.5. This gives Player 1's expected payout to be $0.15 \cdot 3 \cdot 3.5 = 1.575$ and Player 2's expected payout to be $0.85 \cdot 3.5 \cdot 2 = 5.95$, and so we prefer to go second.

Exact: To verify our hypothesis, we'll calculate the *exact* expected payout for both players. Let's consider a simplified game, where there is a single player and they get paid out the cumulative sum of the die rolls when

it is at least 6. Denote the payout to be P, and we will keep with our notation that X represents the number of die rolls until we have reached the sum, and $d_1, \ldots, d_k$ to be the results of our die rolls. Then, by law of total expectation,

$$\mathbb{E}[P] = \mathbb{E}[\mathbb{E}[P \mid X]] = \sum_k \mathbb{E}[P \mid X = k] \cdot \mathbb{P}[X = k]. \tag{1}$$

Thus, we aim to find expressions for $\mathbb{E}[P \mid X = k]$ and $\mathbb{P}[X = k]$.

To find $\mathbb{E}[P \mid X = k]$, we will introduce a new random variable S, which is the sum of the dice *before* the last die roll, so $S = \sum_{i=1}^{k-1} d_i$. Now, given the sum will be greater than 6 after the next roll, S must be no more than 5 and no less than $k-1$ (the worst case scenario of $d_1 = \cdots = d_{k-1} = 1$.) So, applying law of total expectation gives

$$\mathbb{E}\left[P \mid X = k\right] = \mathbb{E}\left[\mathbb{E}\left[P \mid S = s \mid X = k\right]\right] = \sum_{s=k-1}^{5} \mathbb{E}\left[P \mid S = s \cap X = k\right] \cdot \mathbb{P}\left(S = s \mid X = k\right). \tag{2}$$

To find $\mathbb{P}\left(S = s \mid X = k\right)$, we can apply an identical method used in the previous section, wherein we first count the possible ways the $k-1$ previous die can sum to an integer α where $k-1 \leq \alpha \leq 5$. That is, we seek the number of solutions to the equation,

$$\sum_{i=1}^{k-1} d_i = \alpha - k + 1,$$

where we subtracted $k-1$ to accommodate for the fact that $d_i \geq 1$ for all i. Applying stars and bars, finding a solution is equivalent to distributing $\alpha - k + 1$ stars amongst $k-2$ bars, so the number of solutions for some α is

$$\binom{\alpha - k + 1 + k - 2}{k-2} = \binom{\alpha - 1}{k-2}.$$

Now, summing out the α's to find the total number of solutions, we have by the Hockey-stick identity,

$$\sum_{\alpha=k-1}^{5} \binom{\alpha - 1}{k-2} = \binom{5}{k-1}.$$

To find the specific case when $\alpha = s$, we consider the number of solutions to the equation

$$\sum_{i=1}^{k-1} d_i = s - k + 1,$$

where again we subtract $k-1$ for the same reason as before. Applying the stars and bars method again, finding a solution is equivalent to distributing $s - k + 1$ stars amongst $k-2$ bars. Thus, by dividing the number of particular solutions for when $\alpha = s$ by the total number of solutions, we have

$$\mathbb{P}\left(S = s \mid X = k\right) = \frac{\binom{s-k+1+k-2}{k-2}}{\binom{5}{k-2}} = \frac{\binom{s-1}{k-2}}{\binom{5}{k-2}}.$$

Now, turning out attention to $\mathbb{E}\left[P \mid S = s \cap X = k\right]$, we recognise that because after rolling the $(k-1)$th die our next roll will result in the sum being at least 6, the payout P is equally likely to be any accessible number greater than 5. Now, given that the sum of the $(k-1)$ rolls is s, our next roll must be between $6 - s$ and 6, that is, $d_k \in A = \{6 - s, 6 - s + 1, \ldots, 6\}$. Thus, we have that

$$\begin{aligned}\mathbb{E}\left[P \mid S = s \cap X = k\right] &= \frac{6 + 7 + \ldots (6 + s)}{|A|} \\ &= \frac{\sum_{i=0}^{s} 6 + i}{s + 1}.\end{aligned}$$

So, applying this to equation 2, we find

$$\begin{aligned}\mathbb{E}\left[P \mid X=k\right] &= \sum_{s=k-1}^{5} \mathbb{E}\left[P \mid S=s \cap X=k\right] \cdot \mathbb{P}\left(S=s \mid X=k\right) \\ &= \sum_{s=k-1}^{5} \frac{\sum_{i=0}^{s} 6+i}{s+1} \frac{\binom{s-1}{k-2}}{\binom{5}{k-1}} \\ &= \frac{1}{\binom{5}{k-1}} \sum_{s=k-1}^{5} \frac{\sum_{i=0}^{s} 6+i}{s+1} \binom{s-1}{k-2}. \end{aligned} \tag{3}$$

Note that this expression for $\mathbb{E}\left[P \mid X=k\right]$ is not defined for $k=1$, as our method relied on the fact that we have a prior sum. However, $\mathbb{E}\left[P \mid X=1\right]$ is trivially 6, as this is the only possible dice sum that equals 6 off 1 die roll. We move forward with this clarification.

Finding $\mathbb{P}\left(X=k\right)$ is a lot easier! Recall from the proof of Theorem 1.1 that for a n sided die,

$$\mathbb{P}\left(X>k\right) = \frac{1}{n^k}\binom{n-1}{k} \implies \mathbb{P}\left(X>k\right) = \frac{1}{6^k}\binom{5}{k},$$

as we are dealing with a 6 sided die. Using this,

$$\begin{aligned}\mathbb{P}\left(X=k\right) &= \sum_{i=0}^{k} \mathbb{P}\left(X=i\right) - \sum_{i=0}^{k-1} \mathbb{P}\left(X=i\right) \\ &= \mathbb{P}\left(X<k+1\right) - \mathbb{P}\left(X<k\right) \\ &= 1 - \mathbb{P}\left(X>k\right) - \left(1 - \mathbb{P}\left(X>k-1\right)\right) \\ &= \frac{1}{6^{k-1}}\binom{5}{k-1} - \frac{1}{6^k}\binom{5}{k}. \end{aligned} \tag{4}$$

Now, we can finally get an expression for $\mathbb{E}\left[P\right]$. By applying equations 3 and 4 to equation 1, we have[1]

$$\mathbb{E}\left[P\right] = 6 * \frac{1}{6} + \sum_{k=2}^{6} \frac{1}{\binom{5}{k-1}} \sum_{s=k-1}^{5} \frac{\sum_{i=0}^{s} 6+i}{s+1} \binom{s-1}{k-2} \cdot \left(\frac{1}{6^{k-1}}\binom{5}{k-1} - \frac{1}{6^k}\binom{5}{k}\right). \tag{5}$$

And we have an exact expression for the expected payout!

Let the player who has action first be Player 1, and the other player Player 2, and denote their payouts to be P_1 and P_2 respectively. We realise that Player 1 will only receive the payout when X is odd, as they roll the dice on odd turns, and similarly Player 2 will only receive the payout when X is even. Thus, to find their expected payouts, we only need to sum the summand in 5 over the odd and even integers between 1 and 6 inclusive respectively[2]. Hence,

$$\mathbb{E}\left[P_1\right] = 6 * \frac{1}{6} + \sum_{k\in\{3,5\}} \frac{1}{\binom{5}{k-1}} \sum_{s=k-1}^{5} \frac{\sum_{i=0}^{s} 6+i}{s+1} \binom{s-1}{k-2} \cdot \left(\frac{1}{6^{k-1}}\binom{5}{k-1} - \frac{1}{6^k}\binom{5}{k}\right) \approx 2.8831790123$$

and

$$\mathbb{E}\left[P_2\right] = \sum_{k\in\{2,4,6\}} \frac{1}{\binom{5}{k-1}} \sum_{s=k-1}^{5} \frac{\sum_{i=0}^{s} 6+i}{s+1} \binom{s-1}{k-2} \cdot \left(\frac{1}{6^{k-1}}\binom{5}{k-1} - \frac{1}{6^k}\binom{5}{k}\right) \approx 4.5178755144.$$

Finally, since $\mathbb{E}\left[P_1\right] < \mathbb{E}\left[P_2\right]$, we prefer to go second, which aligns with our hypothesis.

Simulation: To check our answer, we can estimate $\mathbb{E}[P_1]$ and $\mathbb{E}[P_2]$ with a Monte-Carlo simulation in Python. The script used is available in A.2. Performing 1,000,000,000 simulations of the game, and taking the mean of Player 1 and 2's winnings across all simulations, we get

$$\mathbb{E}[\hat{P_1}] = 2.9065 \quad \text{and} \quad \mathbb{E}[\hat{P_2}] = 4.6583.$$

Not bad!

[1] Here, we remind ourselves the sum is not defined for $k=1$, but recall that $\mathbb{E}[P \mid X=1]$ is trivially 6.

[2] This was done in Python. See A.1 for more.

A Appendix

A.1 Calculating Summations

The below script `calculate.py` was used to calculate the summations to find $\mathbb{E}[P_1]$ and $\mathbb{E}[P_2]$.

```python
import numpy as np
from scipy.special import comb

# function to calculate E[P | X = k]
def E_P_X(k):

   if k ==1:
      return 6

   outer_sum =0

   for s in range(k-1, 6):
      inner_sum =np.sum([6+i for i in range(0,s+1)])

      binom_s_k =comb(s-1, k-2)

      term =(inner_sum /(s +1)) *binom_s_k

      outer_sum +=term

   binom_5_k =comb(5, k-1)
   return outer_sum /binom_5_k

# function to calculate P[X = k]
def P(k):
   return 1/ 6**(k-1) *comb(5,k-1) -1/6**k *comb(5,k)

odd_k =np.array([1,3,5])
even_k =np.array([2,4,6])

E_player2 =np.sum([E_P_X(k) *P(k) for k in even_k])
E_player1 =np.sum([E_P_X(k) *P(k) for k in odd_k])

print("Player 1 expected payout:", E_player1)
print("Player 2 expected payout:", E_player2)
```

```
$ python3 calculate.py
Player 1 expected payout: 2.8831790123456797
Player 2 expected payout: 4.517875514403292
```

A.2 Simulating the Game

The below script `simulate.py` was used to estimate $\mathbb{E}[P_1]$ and $\mathbb{E}[P_2]$ by means of Monte-Carlo Simulation. 1,000,000,000 simulations were used as the simulation itself is quite cheap, and performing such a number of simulations didn't require significant computation time.

```python
import random

def simulate_game():
   total_sum =0
   current_player =1
   while True:
      roll =random.randint(1, 6)
      total_sum +=roll
      if total_sum >=6:
         return current_player, total_sum

      # switch turns
      current_player =2 if current_player ==1 else 1
```

```python
def simulate_games(num_simulations):
    total_payoff_p1 =0
    total_payoff_p2 =0

    for _ in range(num_simulations):
        winner, payoff =simulate_game()
        if winner ==1:
            total_payoff_p1 +=payoff
        else:
            total_payoff_p2 +=payoff

    # return expected values
    expected_value_p1 =total_payoff_p1 /num_simulations
    expected_value_p2 =total_payoff_p2 /num_simulations

    return expected_value_p1, expected_value_p2

num_simulations =1000000000
ev_p1, ev_p2 =simulate_games(num_simulations)
print(f"Simulated EV for Player 1: {ev_p1:.4f}")
print(f"Simulated EV for Player 2: {ev_p2:.4f}")
```

```
$ python3 simulate.py
Simulated EV for Player 1: 2.9065
Simulated EV for Player 2: 4.6583
```