Vamshi Jandhyala

Running Total of a Die

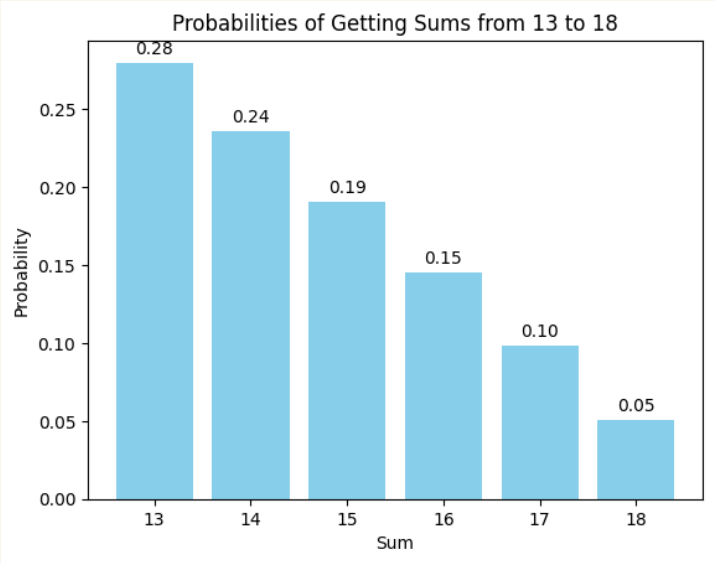
February 2024

Problem

An ordinary die is rolled until the running total of the rolls first exceeds 12. What is the most likely final total that will be obtained?

Solution 1: Monte Carlo

A straightforward simulation estimates the probabilities of final totals 13 through 18. The most likely final total is **13**, with probability approximately 28%.




```
import numpy as np
import matplotlib.pyplot as plt

def monte_carlo_simulation(n=100_000):
    results = []
    for _ in range(n):
        total = 0
        while total <= 12:
```

```
            total += np.random.randint(1, 7)
        results.append(total)
    return results

def calculate_probabilities(results):
    return {i: results.count(i) / len(results) for i in range(13, 19)}

def plot_probabilities(probabilities):
    fig, ax = plt.subplots()
    bars = ax.bar(probabilities.keys(), probabilities.values(), color='skyblue')
    ax.set_xlabel('Sum')
    ax.set_ylabel('Probability')
    ax.set_title('Probabilities of Final Totals from 13 to 18')
    for bar in bars:
        h = bar.get_height()
        ax.annotate(f'{h:.2f}',
                    xy=(bar.get_x() + bar.get_width() / 2, h),
                    xytext=(0, 3), textcoords="offset points",
                    ha='center', va='bottom')
    plt.show()

results = monte_carlo_simulation(100_000)
probabilities = calculate_probabilities(results)
plot_probabilities(probabilities)
```

Solution 2: integer partitions

The second approach is more interesting. We count the number of ways of arriving at a total between 13 and 18 such that every number in the sum is at most 6 and the running total up to the second-to-last number is at most 12. For example, $8+4+2$ is a valid way to reach 14 but $8+5+1$ is not, since $8+5$ already exceeds 12.

Write $p(n,k)$ for the number of partitions of n in which each part is at most k. The number of ways to reach each target is

Total	Count
13	$p(13,6)$
14	$p(8,6)+p(9,6)+p(10,6)+p(11,6)+p(12,6)$
15	$p(9,6)+p(10,6)+p(11,6)+p(12,6)$
16	$p(10,6)+p(11,6)+p(12,6)$
17	$p(11,6)+p(12,6)$
18	$p(12,6)$

The probability of reaching 14 exceeds that of reaching $15,16,17,18$. To decide between 13 and 14, compare $p(13,6)$ with $\sum_{i=8}^{12} p(i,6)$. The code below gives $\Pr[\text{total}=13]\approx 28\%$ and $\Pr[\text{total}=14]\approx 24\%$, confirming 13 is the most likely final total.

```
from collections import Counter
from math import factorial

def generate_partitions(number, max_part=6):
    def recurse(target, mx, current):
        if target == 0:
            yield list(current)
        else:
```

```
            start = min(mx, target)
            for p in range(start, 0, -1):
                yield from recurse(target - p, p, current + [p])
    return list(recurse(number, max_part, []))

def calculate_probability(number):
    def partition_probability(partition):
        counter = Counter(partition)
        l = len(partition)
        num = factorial(l)
        for _, cnt in counter.items():
            num *= 1 / factorial(cnt)
        return num / 6 ** l
    partitions = generate_partitions(number)
    return round(sum(partition_probability(p) for p in partitions), 2)

print("Total probability for 13:", calculate_probability(13))
prob = sum(calculate_probability(n) / 6 for n in range(8, 13))
print("Total probability for 14:", prob)
```