

```
In [42]: from numpy import random, concatenate, full
from collections import Counter

def simulate_hand():

    # Roll 1: 5 dice
    hand = random.randint(1, 7, size=5)

    for _ in range(2): # Two opportunities to re-roll
        if len(set(hand)) == 1:
            #print("YAHTZEE!!!")
            break # Already a Yahtzee!

    # Strategy: Find the most common die value
    counts = Counter(hand)
    keep_val, keep_count = counts.most_common(1)[0]

    # Roll the dice we didn't keep
    num_to_roll = 5 - keep_count
    new_dice = random.randint(1, 7, size=num_to_roll)

    # Update hand: Kept dice + New dice
    hand = concatenate([full(keep_count, keep_val), new_dice])

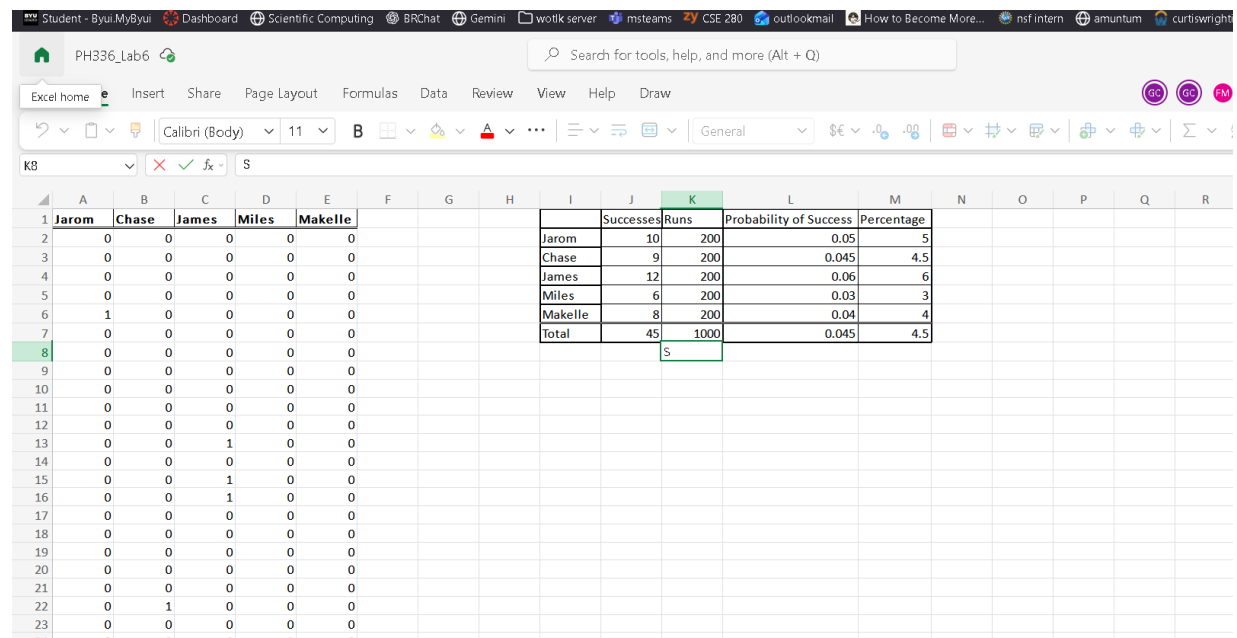
    return len(set(hand)) == 1 # Returns True if all 5 dice are the same

# Run simulation
N = 1_000_000
yahtzees = sum(simulate_hand() for _ in range(N))

print(f"Yahtzees with re-rolls: {yahtzees}")
print(f"Win Probability: {yahtzees/N:.4%}")
```

Yahtzees with re-rolls: 45826

Win Probability: 4.5826%



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Jarom	Chase	James	Miles	Makelle					Successes	Runs	Probability of Success	Percentage					
2	0	0	0	0	0				Jarom	10	200	0.05	5					
3	0	0	0	0	0				Chase	9	200	0.045	4.5					
4	0	0	0	0	0				James	12	200	0.06	6					
5	0	0	0	0	0				Miles	6	200	0.03	3					
6	1	0	0	0	0				Makelle	8	200	0.04	4					
7	0	0	0	0	0				Total	45	1000	0.045	4.5					
8	0	0	0	0	0													
9	0	0	0	0	0													
10	0	0	0	0	0													
11	0	0	0	0	0													
12	0	0	0	0	0													
13	0	0	1	0	0													
14	0	0	0	0	0													
15	0	0	1	0	0													
16	0	0	1	0	0													
17	0	0	0	0	0													
18	0	0	0	0	0													
19	0	0	0	0	0													
20	0	0	0	0	0													
21	0	0	0	0	0													
22	0	1	0	0	0													
23	0	0	0	0	0													

In []: