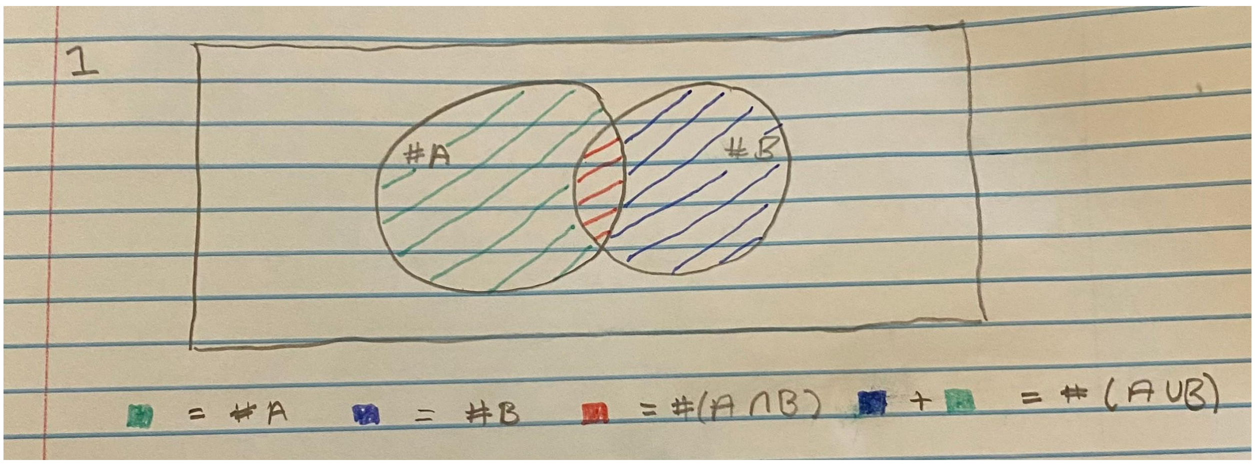
I Joshua Patton, declare that I have completed this assignment in accordance with the UAB Academic Integrity Code and the UAB CS Honor Code. I have read the UAB Academic Integrity Code and understand that any breach of the Code may result in severe penalties.

Student signature/initials: JDP

Date: 1/17/23

---

1. Prove #(A ∪B) = #(A) + #(B) −#(A ∩B)



Suppose that #A = green, #B = Blue and since what is in contained in both A and B would represent #(A ∩ B). By adding #A + #B would give you whats contained in both, however by subtracting #(A ∩ B) from this, we will get what is contained in #A or the #B, thus giving us #(A U B)

2. How many 5 card poker hands have a full house? (two of a kind and three of a kind). Calculate the result theoretically first.

2.

$\binom{4}{3}$ = 3 of a kind

= 4

$\binom{4}{2}$ = 2 of a kind

= 6

$\binom{13}{1}$ = # of chokes for Type of 3 of a kind

= 13

$\binom{12}{1}$ = # of chokes for 2 of a kind

= 12

$$\binom{13}{1}\binom{4}{3}\binom{12}{1}\binom{4}{2} = 13 \times 4 \times 12 \times 6 = 3744$$

Write a pseudo program to simulate the probability that a random drawn 5-card hand is a full house..

```
Import Random
Heart = [2 - 10, J, Q, K, A]
Spade = [2 - 10, J, Q, K, A]
Diamond = [2 - 10, J, Q, K, A]
Clubs = [2 - 10, J, Q, K, A]
Hand = []

n = 1000
for i in Range (n):
        for j in Range (5):
                r = random.randrange(1, 4)
                if (r == 1):
                        Hand.append (random.choice (Spade)
                elif (r == 2):
                        Hand.append (random.choice (Diamond)
                elif (r == 3):
                        Hand.append (random.choice (clubs)
                else:
                        Hand.append (random.choice (Hearts)
```

```
s = set (hand)

if (len (s) == 2)
    FHC = FHC + 1

Hand = []

Print (FHC / n)
```

Note: Another 4 bonus points for both undergrads and grads: implement the code in python, run it, and attach a screenshot of the code and result.

hw3.py > ...

```python
r = 1000
for j in range(r):
    for i in range(5):
        n = random.randrange(1,4)
        if(n == 1):
            hand.append(random.choice(spade))
        elif(n == 2):
            hand.append(random.choice(club))
        elif(n == 3):
            hand.append(random.choice(hearts))
        else:
            hand.append(random.choice(diamond))

    print("\n", hand)
    s = set(hand)

    if(len(s) == 2):
        print("Full House!!")
        fhc = fhc + 1

    hand = []

if(fhc > 1):
    print("You got %d full houses " %fhc, "out of %d attempts"  %r)

print(fhc / r)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
 ['6', '7', '2', '3', '7']

 ['2', '5', '2', 'A', 'K']

 ['4', 'K', '9', '9', '2']

 ['4', 'K', 'A', '5', '7']
You got 12 full houses  out of 1000 attempts
0.012
PS C:\Users\patto\Documents\CS355-Statistics&ProbabilityinCS\HW\HW3>
```

3. Can 950 balls be put into 100 bins such that no bin has 10 or more balls? If yes, give an example. If no, prove.

Since we cannot have more than 9 balls in each of 100 bins we can only have a maximum of 9*10 balls in order for us to meet the requirements for the question.

With 950 balls in 100 bins each bin would need to contain 950/100 = 9.5 balls in each container, but we cant split the balls in half so we would have 50 bins with 9 balls and 50 bins with 10 balls thus 9 * 50 = 450, 10 * 50 = 500, 450 + 500 = 950

4. Pg 60 #30

In this situation there are a few scenarios that can happen.

First being that the first dog is correct and the second is wrong which would be: $p(1 - p)$

Second being that the first dog in wrong and the second dog is correct which is: $(1 - p)p$

The probability of both dogs disagreeing would then be this: $p(1 - p) \times (1 - p)p$ or $2p(1 - p)$

If either of these happen then the hunter chooses the path, which choosing the right path would be 1/2

If both dogs choose the correct path it would be represented by p^2

Thus the probability of choosing the right path would be:

$p^2 + p(1 - p) \times 1/2$ which can simplify down to just p

Therefore, it is the same as just letting one dog choose which path to take.