Assignment 1

Due Oct. 12 2017

1.

mean = 154

median = 140

mode = 140

2.

range = 68 - 45 = 23

3.

Anything dealing with the collection, analysis, interpretation, presentation, and organization of data.

4.

Population

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Sample

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

5.

i.

Population mean is 9.5.

ii.

Population std is 4.9.

iii.

Population variance is 23.8

iv.

Sample std is 5.2

6.

Sample space is the set of all possible outcomes for the experiment.

7.

An event is a subset of a sample space.

8.

i. 6 outcomes of a 6-sided dice {1, 2, 3, 4, 5, 6}.

ii. Results of tossing a coin {H, T}.

9.

i.

You roll a 1; You roll a 6.

ii.

You toss a H; You toss a T.

10.

{(N, N), (N, E), (N, R), (N, O),

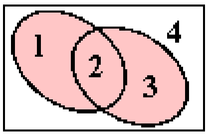
(E, N), (E, E), (E, R), (E, O),

(R, N), (R, E), (R, R), (R, O),

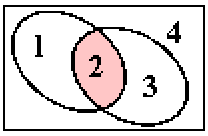
(O, N), (O, E), (O, R), (O, O)}

11.

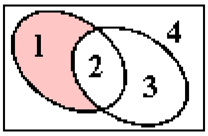
$A \cup B$



$A \cap B$



$A \cap \bar{B}$



12.

1. Let S be a sample space, then P(S) = 1;
2. For any event $0 \leq P(A) \leq 1$;
3. If A, B,... are mutually exclusive, then P (A U B U ...) = P(A) + P(B) + ...

13.

P (A U B) = P(A) + P(B) - P(A ∩ B)

14.

When events are mutually exclusive or not.

15.

0.5

16.

a)

2/6

b)

4/6

17.

4/36=1/9

18.

{(T, H), (T, T), (H, T), (H, H)}

P = 2/4 = 0.5

19.

a)

0.5^5

b)

0.5^5

20.

a)

$$P = \frac{\binom{13}{3}}{\binom{52}{3}} = 1.29411765e - 02$$

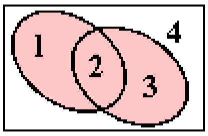
b)

$$P = \frac{\binom{39}{3}}{\binom{52}{3}} = 4.13529412e - 01$$

c)

$$P = \frac{\binom{4}{3}}{\binom{52}{3}} = 1.80995475e - 04$$

21.



Define ,

1+2: 'Women' = 55,

3+2: 'Over 30' = 60,

2: 'Women ∩ Over 30' = 25,

Our aim is to find out number of people over 30 or women, which is 'Women' ∪ 'Over 30',

equals to, 'Women' + 'Over 30' - 'Women ∩ Over 30' = 55 + 60 - 25 = 90

P = 90 / 150 = 0.6

22.

a)

For each question the probability of getting the correct answer $p = 0.5$.

```
import numpy as np

simulation_size = 1000 # run simulation 1000 times.
pass_ = 0
fail_ = 0
for i in range(simulation_size):
    results = np.random.binomial(1, 0.5, 3) # generate 3 results of B(n=1, p=0.5).
    # count pass and failure cases
    if np.sum(results)>1:
        pass_ += 1
    else:
        fail_ += 1
```

b)

$$P(pass) = P(2) + P(3)$$

For binomial distribution,

$$P = \binom{N}{k} p^k (1-p)^{N-k}$$

With $N = 3,\ p = 0.5,\ k = 2, 3.$

$$\begin{aligned} P(pass) &= \binom{3}{2} p^2 (1-p)^1 + \binom{3}{3} p^3 \\ &= 0.5 \end{aligned}$$

23.

```
import numpy as np

#Case1: Not Switching
simulation_size = 1000
wins = 0
lose = 0
for i in range(simulation_size):
    doors = np.zeros(3)
    doors[np.random.randint(3)] = 1 #Set one car behind doors
    choice = np.random.randint(3)   #Random choice made by player
    if doors[choice]:
        wins += 1
    else:
        lose += 1
#Calculate probability of winning without switching
probability_notswitch = wins / (wins + lose)

#Case2: Switching
wins = 0
lose = 0
for i in range(simulation_size):
    doors = np.zeros(3)
    car = np.random.randint(3)    #Set one car behind doors
    doors[car] = 1
    choice = np.random.randint(3) #Random choice made by player
    if choice == car:             #Host randomly choose one without car when player is right
        host = np.random.choice(np.delete(np.arange(3), choice))
    else:                         #host choose the only door without car when player is wrong
        host = np.delete(np.arange(3), [choice, car])
    choice = np.delete(np.arange(3), [choice, host]) # Player switch choice
    if doors[choice]:
        wins += 1
    else:
        lose += 1
#Calculate probability of winning when switching
probability_switch = wins/(wins + lose)
```