

▼ Discussion 1

11 Sept 2023

This notebook contains the questions and the solutions to the topics covered in Discussion 1.

```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

▼ Review of numpy

```
#create a numpy array with elements from 1 to 5
arr1 = np.array([1,2,3,4,5])
arr1
```

```
array([1, 2, 3, 4, 5])
```

```
#create an array with elements from 0 to 10
arr2 = np.array(range(11))
arr2
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
#create an array with even numbers from 0 to 18
arr = np.arange(0,20,2)
arr
```

```
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

```
#create an array with 10 equally spaced elements from 0 to 6
arr = np.linspace(0,6,10)
arr
```

```
array([0.          , 0.66666667, 1.33333333, 2.          , 2.66666667,
       3.33333333, 4.          , 4.66666667, 5.33333333, 6.          ])
```

Creating a 2D matrix as follows:

```
[[1,2,3]
```

```
[4,5,6]
```

```
[7,8,9]]
```

```
matrix = np.array([[1,2,3],[4,5,6],[7,8,9]])
matrix
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
matrix.T
```

```
array([[1, 4, 7],
       [2, 5, 8],
       [3, 6, 9]])
```

```
data = np.random.randint(1,101,(5,5))
```

```
data
```

```
array([[68, 62,  5, 37, 93],
       [31, 38, 60, 47, 72],
       [78, 93, 87, 90, 78],
       [92, 11, 27, 20, 26],
       [93, 63, 41, 44, 58]])
```

Problem 1: Calculate the sum, mean, median and standard deviation of 'data'.

```
#problem 1
#sum, mean, median, std dev
mean_data = np.mean(data)
sum_data = np.sum(data)
median_data = np.median(data)
std_data = np.std(data)

print(mean_data)
print(sum_data)
print(median_data)
print(std_data)

56.56
1414
60.0
27.13533489750956

#finding the dot product of 2 random 5x5 arrays
arr1 = np.random.rand(5,5)
arr2 = np.random.rand(5,5)

#method 1
print(arr1.dot(arr2))

[[0.84130372 0.51712601 0.70269174 0.98516978 0.6591656 ]
 [1.48246524 1.40536053 1.83921681 1.81719543 0.77959355]
 [2.05402451 1.39425896 1.78866475 1.92846364 1.62198328]
 [1.98210725 1.63396388 1.71610661 2.17720699 1.12929985]
 [0.89137678 0.72233889 1.02014441 0.98467356 0.69416711]]

#method 2
print(arr1@arr2)

[[0.84130372 0.51712601 0.70269174 0.98516978 0.6591656 ]
 [1.48246524 1.40536053 1.83921681 1.81719543 0.77959355]
 [2.05402451 1.39425896 1.78866475 1.92846364 1.62198328]
 [1.98210725 1.63396388 1.71610661 2.17720699 1.12929985]
 [0.89137678 0.72233889 1.02014441 0.98467356 0.69416711]]

#method 3
print(np.dot(arr1, arr2))

[[0.84130372 0.51712601 0.70269174 0.98516978 0.6591656 ]
 [1.48246524 1.40536053 1.83921681 1.81719543 0.77959355]
 [2.05402451 1.39425896 1.78866475 1.92846364 1.62198328]
 [1.98210725 1.63396388 1.71610661 2.17720699 1.12929985]
 [0.89137678 0.72233889 1.02014441 0.98467356 0.69416711]]
```

Problem 2: Calculate the mean of 10 elements from the matrix called 'population' defined below without replacement.

```
population = np.arange(1,101)

#problem 2
#calculate the mean of 10 elements from population (without replacement)

#it is a good idea to break the problem in parts

#part 1 is to first take a random sample
random_sample = np.random.choice(population, 10, replace=False)

#part 2 is to find the mean of that sample
mean_sample = np.mean(random_sample)
mean_sample

57.9
```

▼ Review of pandas

```
#reading the csv file (available on Piazza)
df = pd.read_csv('INSERT THE PATH TO THE FILE HERE/SuperheroDataset.csv')
```

```
df.head()
```

	Unnamed: 0.1	Unnamed: 0	Name	Url	Intelligence	Strength	Speed	Durability	Power	Combat	...	Height	Weight	
0	0	0	3-D Man	/3-d-man/10-226/	80.0	35.0	45.0	35.0	25.0	55.0	...	6'2 // 188 cm	200 lb // 90 kg	t
1	1	1	A-Bomb	/a-bomb/10-10060/	75.0	100.0	20.0	80.0	25.0	65.0	...	6'8 // 203 cm	980 lb // 441 kg	\
2	2	2	Abe Sapien	/abe-sapien/10-956/	95.0	30.0	35.0	65.0	100.0	85.0	...	6'3 // 191 cm	145 lb // 65 kg	
3	3	3	Abin Sur	/abin-sur/10-1460/	80.0	90.0	55.0	65.0	100.0	65.0	...	6'1 // 185 cm	200 lb // 90 kg	
4	4	4	Abomination	/abomination/10-1/	85.0	80.0	55.0	90.0	65.0	95.0	...	6'8 // 203 cm	980 lb // 441 kg	t

5 rows x 29 columns

Problem 3: This is your data, and you're the data scientist. To perform EDA, you first need to clean the data. Some suggestions are as follows:

- 1. Deleting irrelevant columns
- 2. Replacing the NaN values with 0

```
#printing column names for easier accessibility
column_names = df.columns
print(column_names)

Index(['Unnamed: 0.1', 'Unnamed: 0', 'Name', 'Url', 'Intelligence', 'Strength',
      'Speed', 'Durability', 'Power', 'Combat', 'Full name', 'Alter Egos',
      'Aliases', 'Place of birth', 'First appearance', 'Creator', 'Alignment',
      'Gender', 'Race', 'Height', 'Weight', 'Eye color', 'Hair color',
      'Occupation', 'Base', 'Team Affiliation', 'Relatives', 'Skin color',
      'Total Power'],
      dtype='object')

cols_to_drop = ['Unnamed: 0.1', 'Unnamed: 0','Url', 'Full name', 'Alter Egos',
      'Aliases', 'Place of birth', 'First appearance','Height', 'Weight','Occupation', 'Base', 'Team Affiliation', 'Relatives']

df = df.drop(columns = cols_to_drop)

df.head()
```

```

    Name Intelligence Strength Speed Durability Power Combat Creator Alignment Gender Race Eye color Hair color
0 3-D Man 80.0 35.0 45.0 35.0 25.0 55.0 Marvel good Male - Brown G
#replacing NaN values with 0s
df = df.fillna(0)

df.head()

```

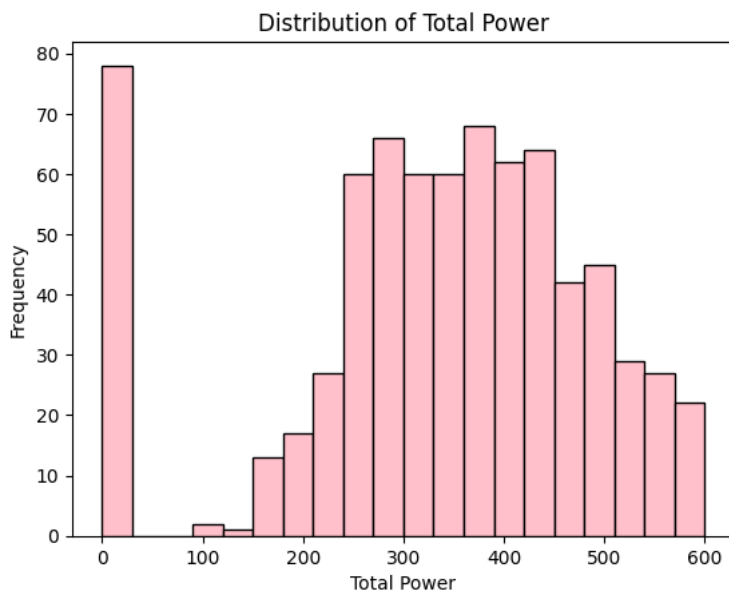
	Name	Intelligence	Strength	Speed	Durability	Power	Combat	Creator	Alignment	Gender	Race	Eye color	Hair color
0	3-D Man	80.0	35.0	45.0	35.0	25.0	55.0	Marvel Comics	good	Male	-	Brown	Green
1	A-Bomb	75.0	100.0	20.0	80.0	25.0	65.0	Marvel Comics	good	Male	Human	Yellow	No Hair
2	Abe Sapien	95.0	30.0	35.0	65.0	100.0	85.0	Dark Horse Comics	good	Male	Ichthyo Sapien	Blue	No Hair
3	Abin Sur	80.0	90.0	55.0	65.0	100.0	65.0	DC Comics	good	Male	Ungaran	Blue	No Hair
4	Abomination	85.0	90.0	55.0	80.0	65.0	85.0	Marvel	bad	Male	Human /	Green	No Hair

Problem 4 - Perform EDA by making any 2 plots to explore the data better.

```

#this can be different for everyone, here's just an example
#histogram for 'Total Power' distribution
plt.hist(df['Total Power'], bins=20, edgecolor='k', color = 'pink')
plt.title('Distribution of Total Power')
plt.xlabel('Total Power')
plt.ylabel('Frequency')
plt.show()

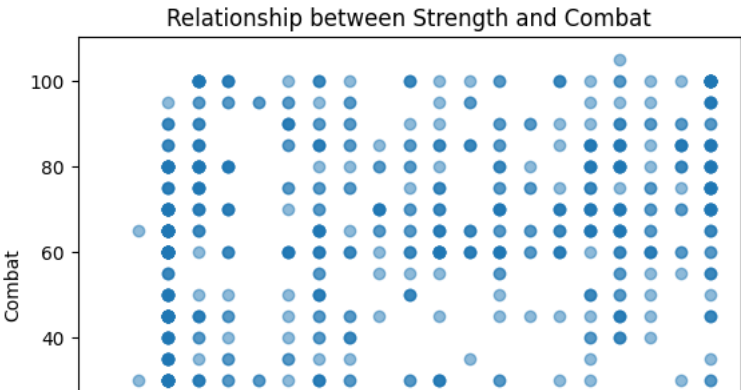
```



```

#scatter plot of 'Strength' vs. 'Combat'
plt.scatter(df['Strength'], df['Combat'], alpha=0.5)
plt.title('Relationship between Strength and Combat')
plt.xlabel('Strength')
plt.ylabel('Combat')
plt.show()

```



Problem 5: Make a separate column called 'avg' in the same dataframe consisting of average of intelligence, strength, speed, durability, power and combat.

```
|          |
df['avg'] = df[['Intelligence', 'Strength', 'Speed', 'Durability', 'Power', 'Combat']].mean(axis=1)
df.head()
```

	Name	Intelligence	Strength	Speed	Durability	Power	Combat	Creator	Alignment	Gender	Race	Eye color	Hair color	Score
0	3-D Man	80.0	35.0	45.0	35.0	25.0	55.0	Marvel Comics	good	Male	-	Brown	Grey	
1	A-Bomb	75.0	100.0	20.0	80.0	25.0	65.0	Marvel Comics	good	Male	Human	Yellow	No Hair	
2	Abe Sapien	95.0	30.0	35.0	65.0	100.0	85.0	Dark Horse Comics	good	Male	Ichthyo Sapien	Blue	No Hair	1
3	Abin Sur	80.0	90.0	55.0	65.0	100.0	65.0	DC Comics	good	Male	Ungaran	Blue	No Hair	
4	Abomination	85.0	90.0	55.0	80.0	65.0	85.0	Marvel	bad	Male	Human /	Green	No	