

# Assignment 7

Name : Vishwajit Shelke

Roll Number : 33172 K-9

Problem Statement -

Visualize the data using Python libraries matplotlib, seaborn by plotting the graphs for assignment no. 2 and 3

In [ ]:

```
# importing libraries
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
import plotly.express as px
import plotly.figure_factory as ff
```

## Heart Disease Dataset - Data Loading

In [ ]:

```
# reading the csv file
df = pd.read_csv("heart.csv")
df.head()
```

Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

In [ ]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   age         1025 non-null   int64
1   sex         1025 non-null   int64
2   cp          1025 non-null   int64
3   trestbps    1025 non-null   int64
4   chol        1025 non-null   int64
5   fbs         1025 non-null   int64
6   restecg     1025 non-null   int64
7   thalach     1025 non-null   int64
8   exang       1025 non-null   int64
9   oldpeak     1025 non-null   float64
10  slope       1025 non-null   int64
11  ca          1025 non-null   int64
12  thal        1025 non-null   int64
13  target      1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

In [ ]:

```
df.describe()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.000000	0.149268	0.526910
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527010
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000

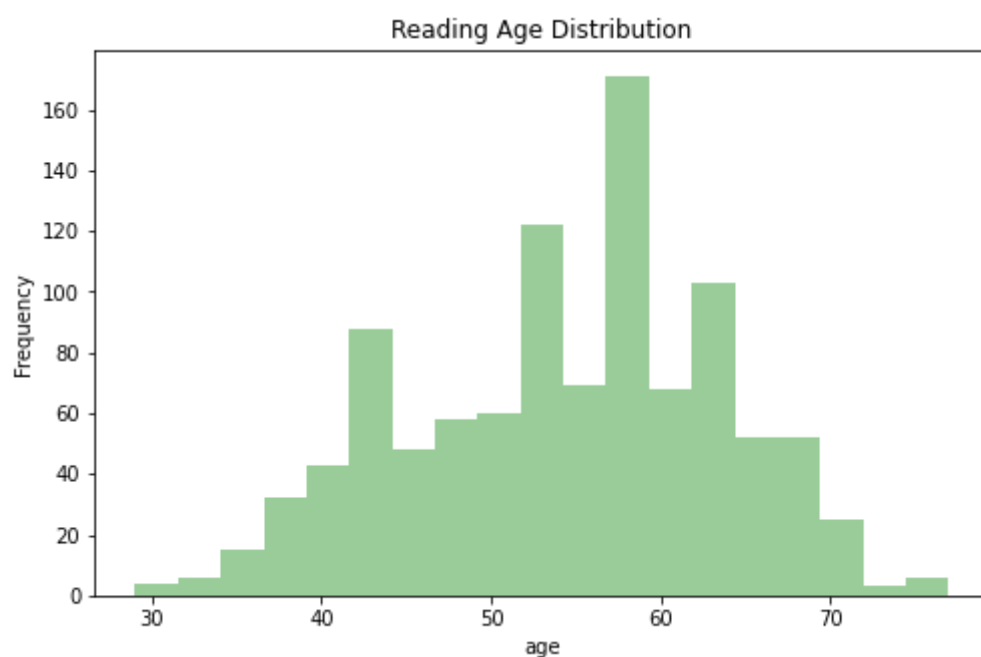
## Distribution of Age in the dataset

In [ ]:

```
plt.figure(figsize=(8,5))
sns.distplot(df['age'], color='g', kde=False)
plt.ylabel('Frequency')
plt.title('Reading Age Distribution')
plt.show()
```

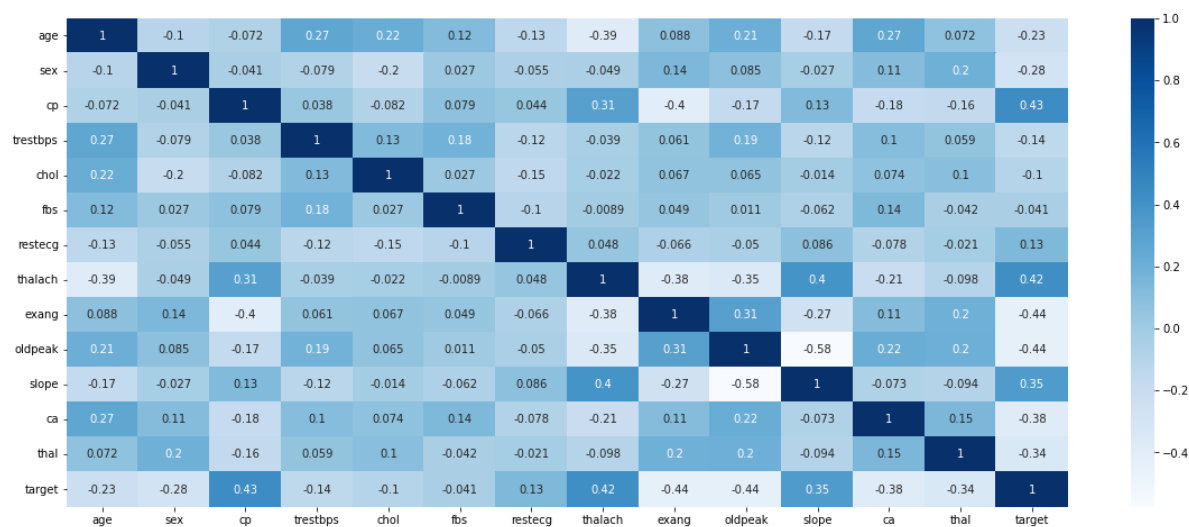
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).



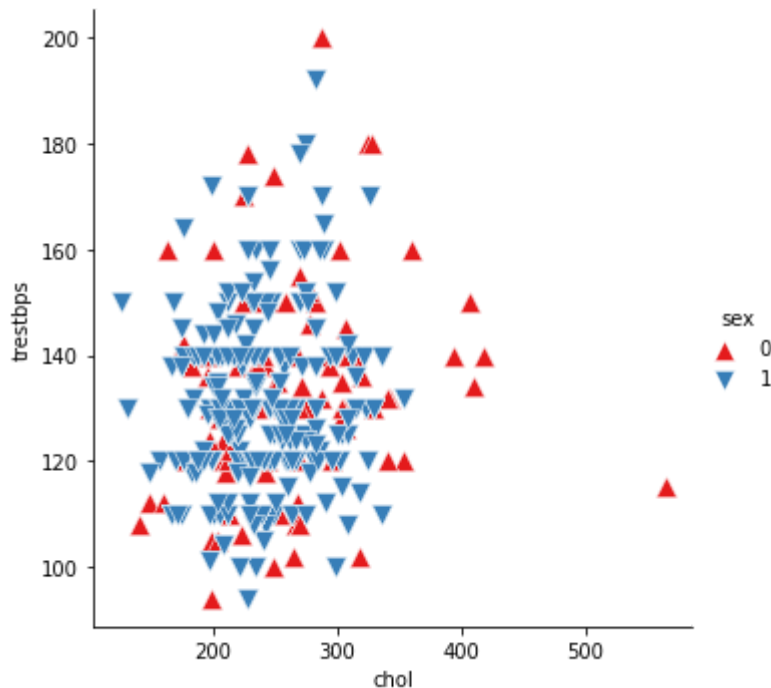
In [ ]:

```
plt.figure(figsize=(20,8))
sns.heatmap(df.corr(), annot = True, cmap='Blues')
plt.show()
```



In [ ]:

```
g = sns.FacetGrid(df, hue="sex", palette="Set1", height=5, hue_kws={"marker": ["^", "v"]})
g.map(plt.scatter, "chol", "trestbps", s=100, linewidth=.5, edgecolor="white")
g.add_legend();
```

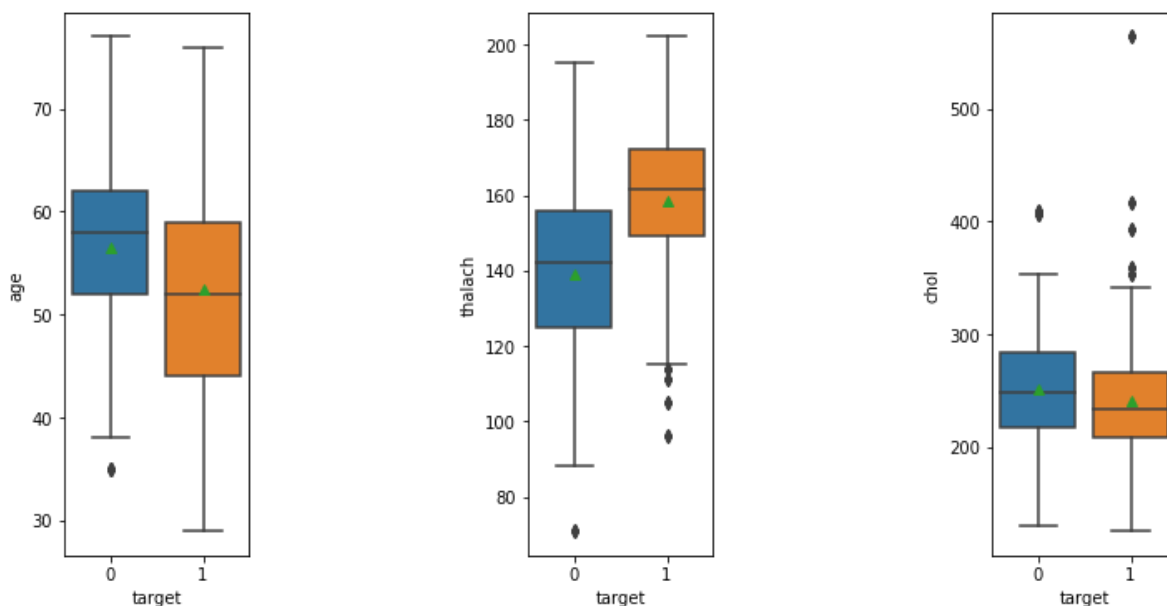


In [ ]:

```
fig, axs = plt.subplots(ncols=3, figsize=(12,6))
plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=1.5, hspace=None)
sns.boxplot(x="target", y="age", data=df, ax=axs[0], showmeans=True)
sns.boxplot(x="target", y="thalach", data=df, ax=axs[1], showmeans=True)
sns.boxplot(x="target", y="chol", data=df, ax=axs[2], showmeans=True)
```

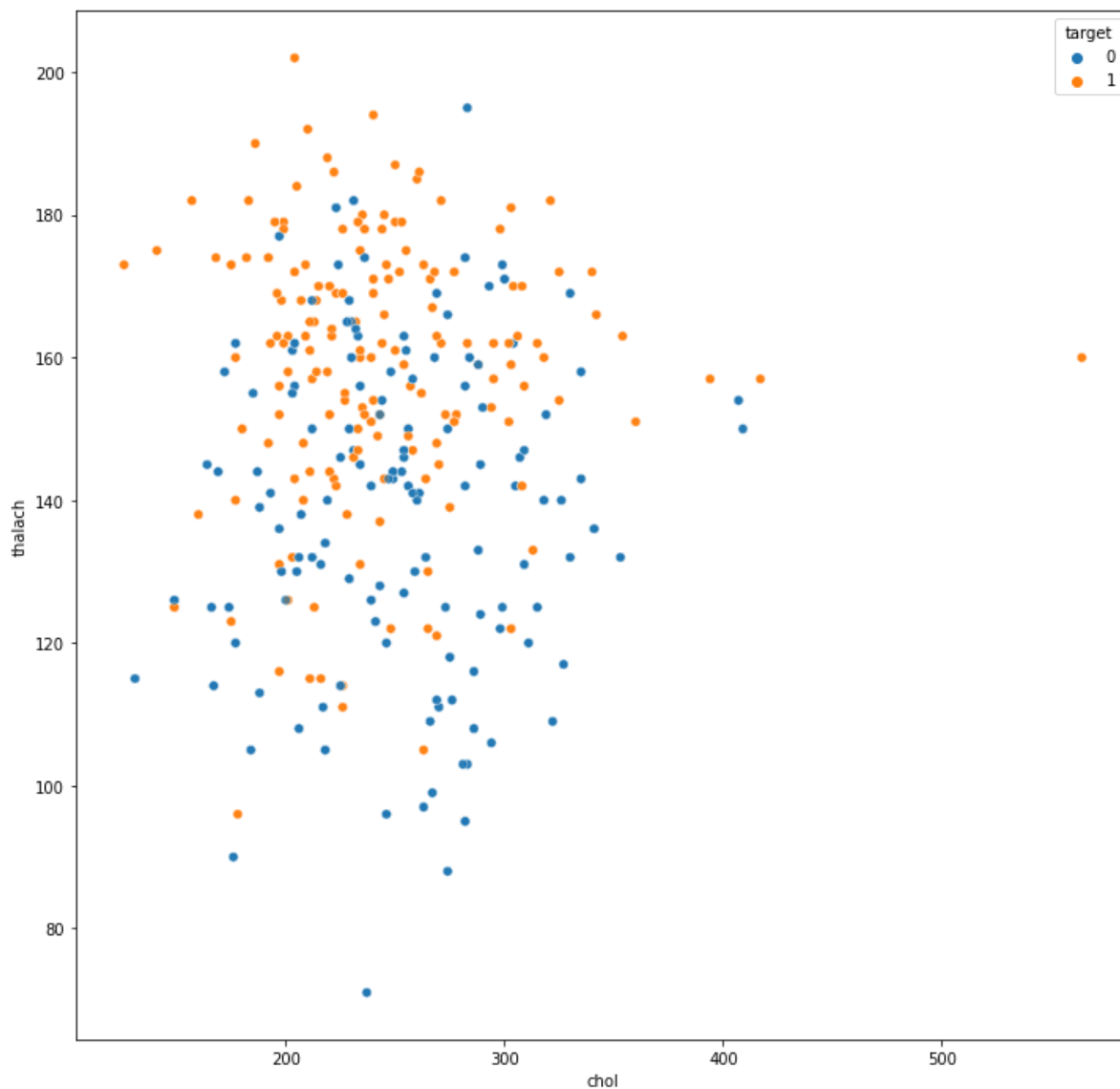
Out[8]:

&lt;matplotlib.axes.\_subplots.AxesSubplot at 0x7fa117c58890&gt;



In [ ]:

```
fig, ax = plt.subplots(1, 1, figsize=(12, 12))  
sns.scatterplot(data=df, x='chol', y='thalach', hue='target', alpha=0.7)  
plt.show()
```



## Facebook Metrics - Data Loading

In [ ]:

```
df_fb = pd.read_csv("dataset_Facebook.csv", sep=";")
df_fb.head()
```

Out[11]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	I Con
0	139441	Photo	2	12	4	3	0.0	2752	5091	178	
1	139441	Status	2	12	3	10	0.0	10460	19057	1457	
2	139441	Photo	3	12	3	3	0.0	2413	4373	177	
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211	
4	139441	Photo	2	12	2	3	0.0	7244	13594	671	



In [ ]:

```
df_fb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 19 columns):
#   Column
Non-Null Count  Dtype
---  -
0   Page total likes
500 non-null    int64
1   Type
500 non-null    object
2   Category
500 non-null    int64
3   Post Month
500 non-null    int64
4   Post Weekday
500 non-null    int64
5   Post Hour
500 non-null    int64
6   Paid
499 non-null    float64
7   Lifetime Post Total Reach
500 non-null    int64
8   Lifetime Post Total Impressions
500 non-null    int64
9   Lifetime Engaged Users
500 non-null    int64
10  Lifetime Post Consumers
500 non-null    int64
11  Lifetime Post Consumptions
500 non-null    int64
12  Lifetime Post Impressions by people who have liked your Page
500 non-null    int64
13  Lifetime Post reach by people who like your Page
500 non-null    int64
14  Lifetime People who have liked your Page and engaged with your post
500 non-null    int64
15  comment
500 non-null    int64
16  like
499 non-null    float64
17  share
496 non-null    float64
18  Total Interactions
500 non-null    int64
dtypes: float64(3), int64(15), object(1)
memory usage: 74.3+ KB
```



In [ ]:

```
df_fb.describe()
```

Out[13]:

	Page total likes	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime P Total Rea
<b>count</b>	500.000000	500.000000	500.000000	500.000000	500.000000	499.000000	500.000000
<b>mean</b>	123194.176000	1.880000	7.038000	4.150000	7.840000	0.278557	13903.360000
<b>std</b>	16272.813214	0.852675	3.307936	2.030701	4.368589	0.448739	22740.780000
<b>min</b>	81370.000000	1.000000	1.000000	1.000000	1.000000	0.000000	238.000000
<b>25%</b>	112676.000000	1.000000	4.000000	2.000000	3.000000	0.000000	3315.000000
<b>50%</b>	129600.000000	2.000000	7.000000	4.000000	9.000000	0.000000	5281.000000
<b>75%</b>	136393.000000	3.000000	10.000000	6.000000	11.000000	1.000000	13168.000000
<b>max</b>	139441.000000	3.000000	12.000000	7.000000	23.000000	1.000000	180480.000000

## Subset Generation in Facebook Metrics Dataset

In [ ]:

```
photos = df_fb[df_fb["Type"] == "Photo"]
status = df_fb[df_fb["Type"] == "Status"]
link = df_fb[df_fb["Type"] == "Link"]
video = df_fb[df_fb["Type"] == "Video"]
```

## Data Visualization in Facebook Metrics

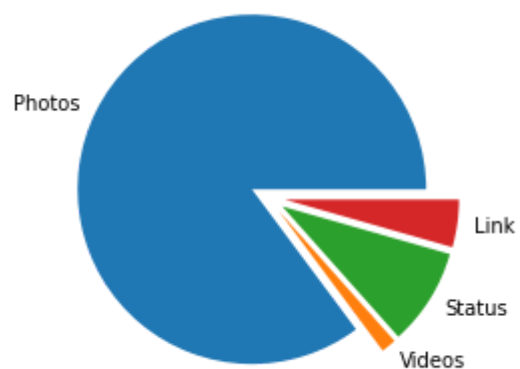
In [ ]:

```
# Distribution of Total Interactions
```

```
a = np.array([photos.shape[0], video.shape[0], status.shape[0], link.shape[0]])  
labels = ["Photos", "Videos", "Status", "Link"]  
  
plt.pie(a, labels = labels, explode = [0.1, 0.1, 0.1, 0.1])  
plt.plot()
```

Out[15]:

[]



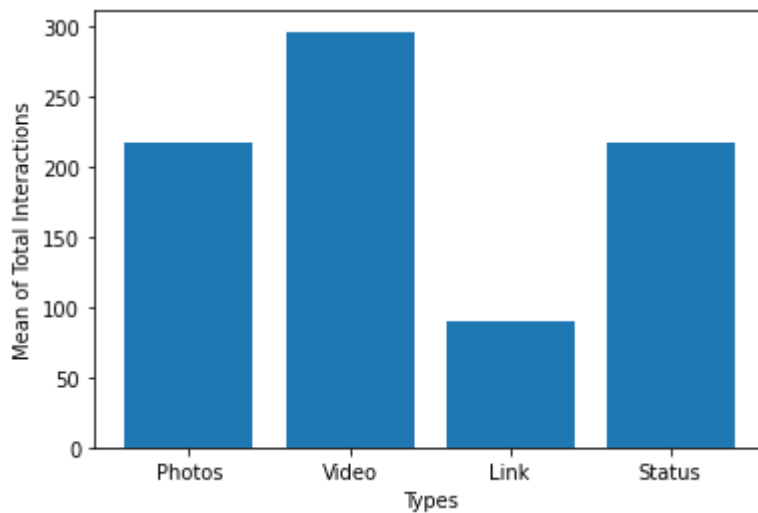
In [ ]:

```
# Mean of Each type of Interaction
```

```
total = np.array([photos["Total Interactions"].mean(), video["Total Interactions"].mean(),  
x = ["Photos", "Video", "Link", "Status"]  
plt.xlabel("Types")  
plt.ylabel("Mean of Total Interactions")  
plt.bar(x, total)  
plt.plot()
```

Out[16]:

[]



In [ ]:

```
fig = px.scatter(df_fb, x='like', y='comment',  
                 color='Type',  
                 )  
fig.show()
```

In [ ]: