

# Assignment 7

## Details

1. Author : Shivendra Bhonsle
2. Roll Number : 33109
3. Batch : K9
4. Class : TE9

## Problem Statement

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

## Implementation details

1. Dataset URLs
  - A. Facebook metrics : <https://archive.ics.uci.edu/ml/datasets/Facebook+metrics>  
(<https://archive.ics.uci.edu/ml/datasets/Facebook+metrics>).
  - B. Heart Disease : <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>  
(<https://archive.ics.uci.edu/ml/datasets/Heart+Disease>).
2. Python version : 3.7.4
3. Imports :
  - A. pandas
  - B. numpy
  - C. matplotlib
  - D. seaborn

## Dataset details

1. Facebook Metrics :
  - A. Given dataset is a representative of some of the Facebook metrics which are associated with the posts on social media.
  - B. These metrics are indicative of the engagement of the users with the corresponding post.
  - C. It includes various types of posts and their details
2. Heart Disease Dataset :
  - A. This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date.
  - B. The "goal" field refers to the presence of heart disease in the patient.
  - C. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).
  - D. The names and social security numbers of the patients were recently removed from the database, replaced with dummy values

## Importing required libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
%matplotlib inline
```

## A) Visualization for Facebook metrics dataset

### 1) Loading the dataset

In [2]:

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

Out[2]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Cc
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	

### 2) Distribution of data based on type of Post

In [3]:

```
# Acquiring unique post values
post_types = facebook_dataset.Type.unique()
post_types
```

Out[3]:

```
array(['Photo', 'Status', 'Link', 'Video'], dtype=object)
```

In [4]:

```
# Generating frequency data for each type of post
```

```
frequency_data = {}  
for post in post_types:  
    subset = facebook_dataset[facebook_dataset.Type == post]  
    frequency_data[post] = subset.shape[0]
```

```
frequency_data
```

Out[4]:

```
{'Photo': 426, 'Status': 45, 'Link': 22, 'Video': 7}
```

In [5]:

```
fig = plt.figure(figsize=(8, 8))

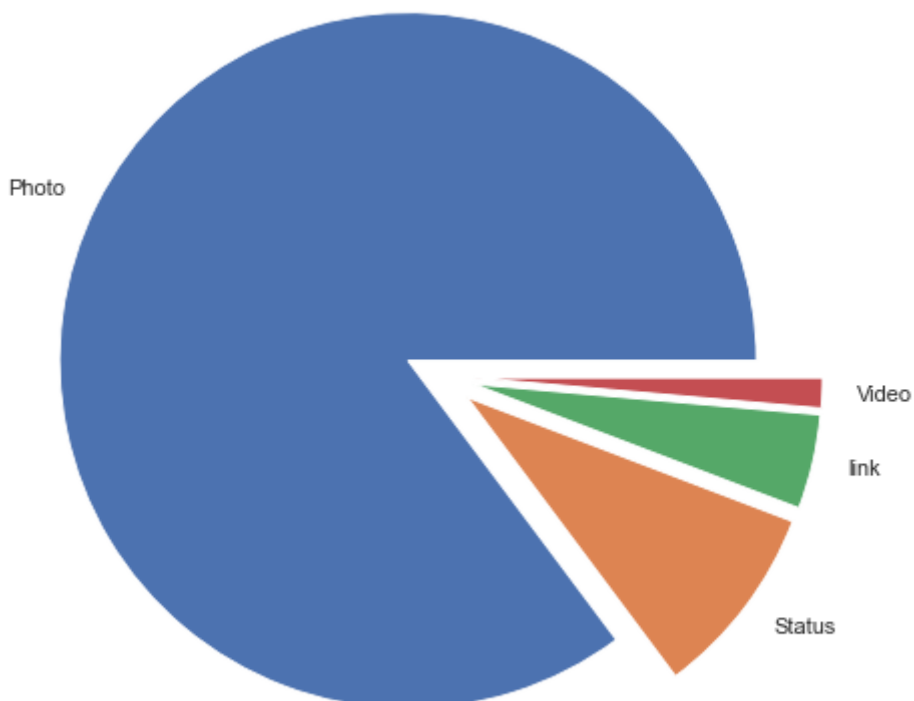
# Adds subplot on position 1
ax = fig.add_subplot(111)

# Generating Legend for pie chart
legend = [
    "Photo",
    "Status",
    "link",
    "Video"
]

# Defining explode values
explode = [0.1, 0.1, 0.1, 0.1]

# Generating and displaying piechart
plt.pie(
    x=frequency_data.values(),
    labels=legend,
    explode=explode,
)
plt.title("Composition of post types in data (Pie Chart)", fontsize=20)
plt.show()
```

Composition of post types in data (Pie Chart)



### 3) Likes per type of data

In [6]:

```
# Generating data for count of Likes
likes_per_type = {}

for post in post_types:
    subset = facebook_dataset[facebook_dataset.Type == post]
    likes_per_type[post] = subset.like.sum()

likes_per_type
```

Out[6]:

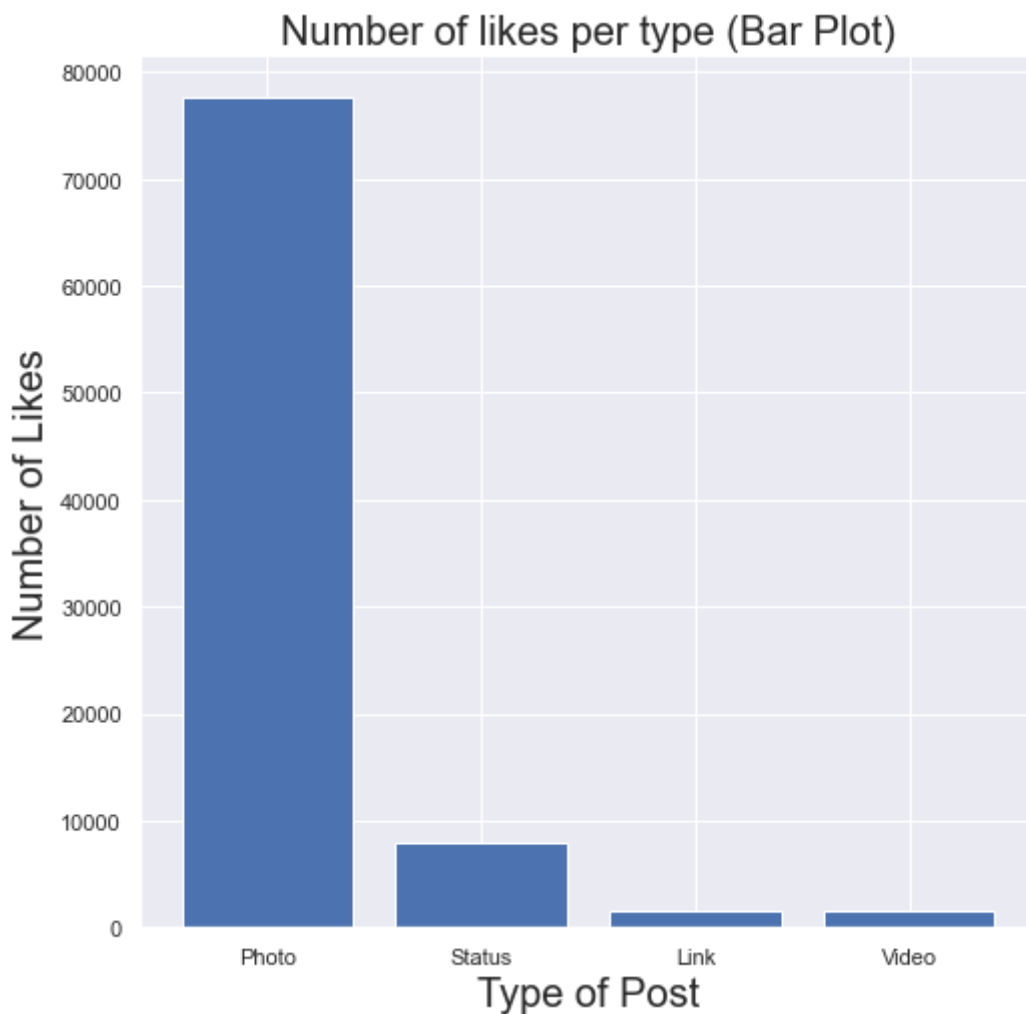
```
{'Photo': 77610.0, 'Status': 7952.0, 'Link': 1613.0, 'Video': 1620.0}
```

In [7]:

```
# Generating bar graph
fig = plt.figure(figsize=(8, 8))

# Adds subplot on position 1
ax = fig.add_subplot(111)

# Generating and displaying bar chart
plt.bar(
    x=likes_per_type.keys(),
    height=likes_per_type.values()
)
plt.xlabel("Type of Post", fontsize=20)
plt.ylabel("Number of Likes", fontsize=20)
plt.title("Number of likes per type (Bar Plot)", fontsize=20)
plt.show()
```



## 4) Counting number of paid and unpaid posts

In [8]:

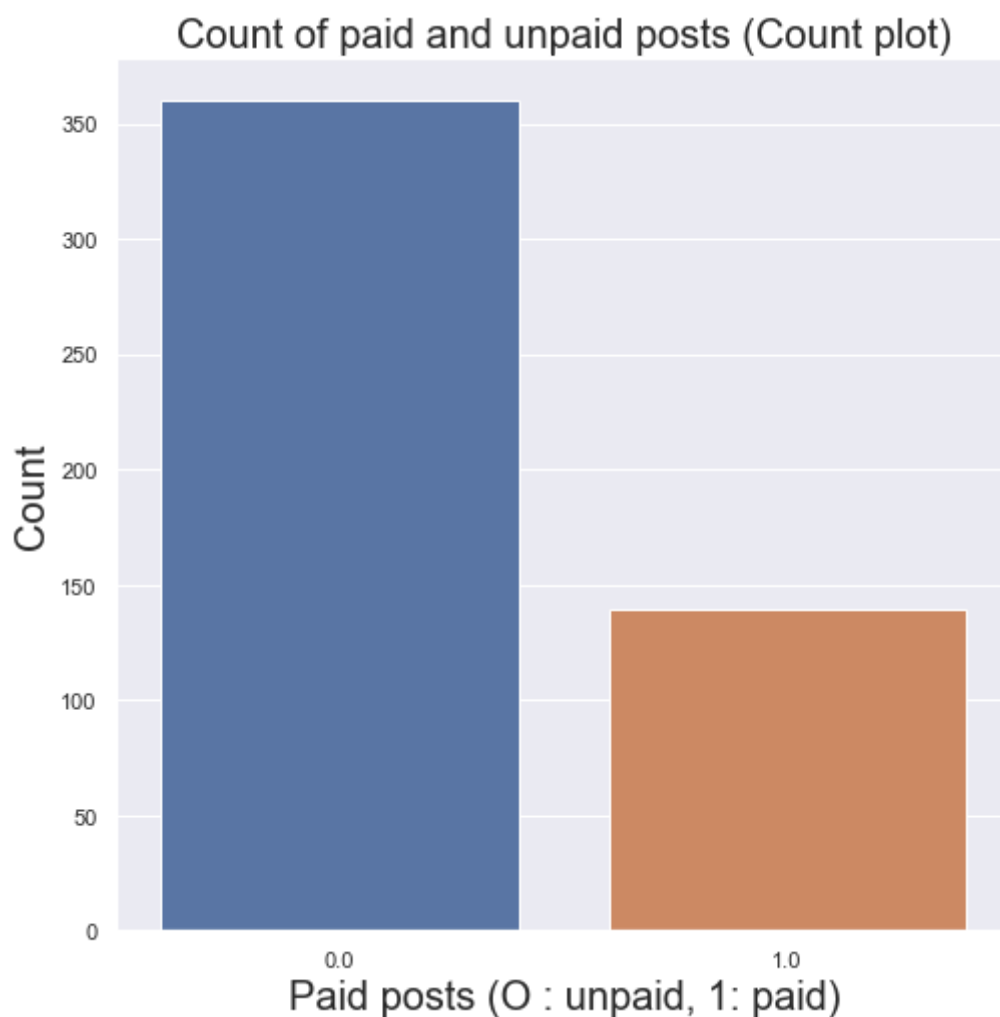
```
# Generating bar graph
fig = plt.figure(figsize=(8, 8))

# Adds subplot on position 1
ax = fig.add_subplot(111)

sns.countplot(x=facebook_dataset.Paid)

plt.xlabel("Paid posts (0 : unpaid, 1: paid)", fontsize=20)
plt.ylabel("Count", fontsize=20)
plt.title("Count of paid and unpaid posts (Count plot)", fontsize=20)

plt.show()
```



## B) Heart Disease dataset

# 1) Loading the dataset

In [9]:

```
heart_dataset = pd.read_csv("./processed.cleveland.csv", header=None)
heart_dataset.head()
```

Out[9]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.0	6.0	0
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	2
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.0	7.0	1
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	0
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.0	3.0	0

## 2) Renaming columns

In [10]:

```
heart_dataset.columns = [
    "age",
    "sex",
    "chest_pain",
    "trestbps",
    "cholesterol",
    "fbs",
    "restecg",
    "thalach",
    "exang",
    "oldpeak",
    "slope",
    "ca",
    "thal",
    "num"
]
```



In [11]:

```
heart_dataset.head()
```

Out[11]:

	age	sex	chest_pain	trestbps	cholesterol	fbs	restecg	thalach	exang	oldpeak	slope	c
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.

### 3) Quartile spread of thalach feature

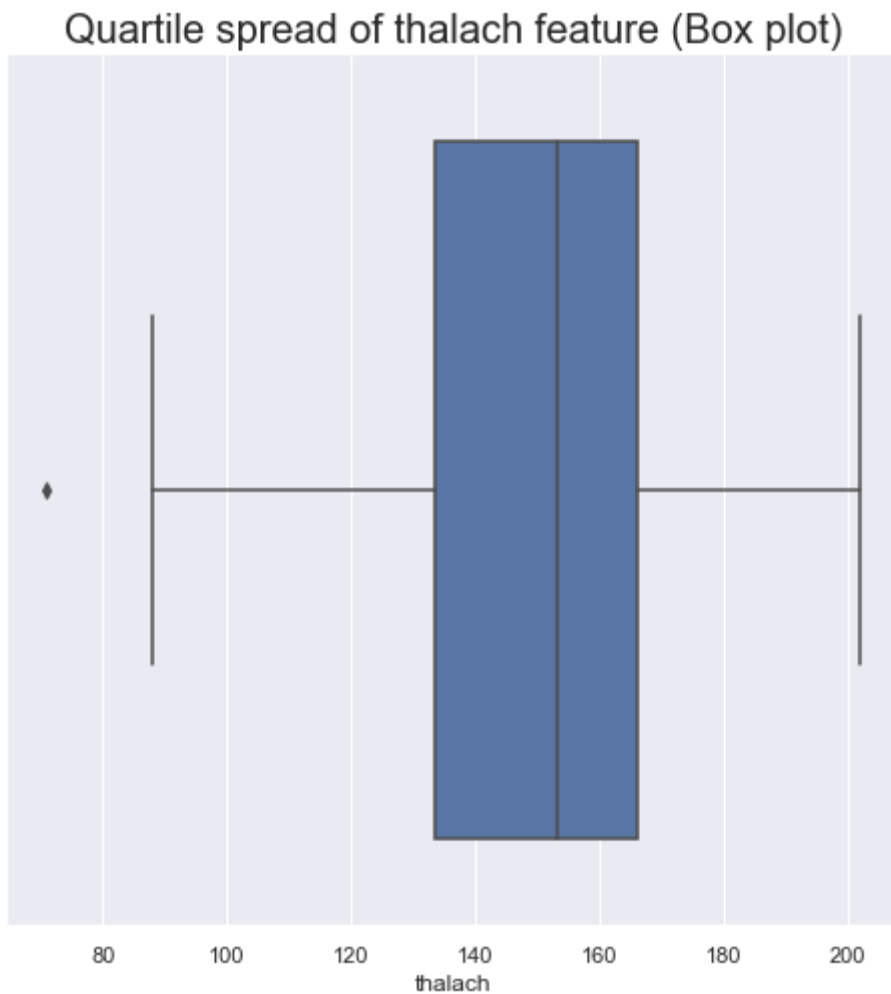
In [12]:

```
# Generating bar graph
fig = plt.figure(figsize=(8, 8))

# Adds subplot on position 1
ax = fig.add_subplot(111)

sns.boxplot(x=heart_dataset.thalach)
plt.title("Quartile spread of thalach feature (Box plot)", fontsize=20)

plt.show()
```



## 4) Distribution of age in entire dataset

In [13]:

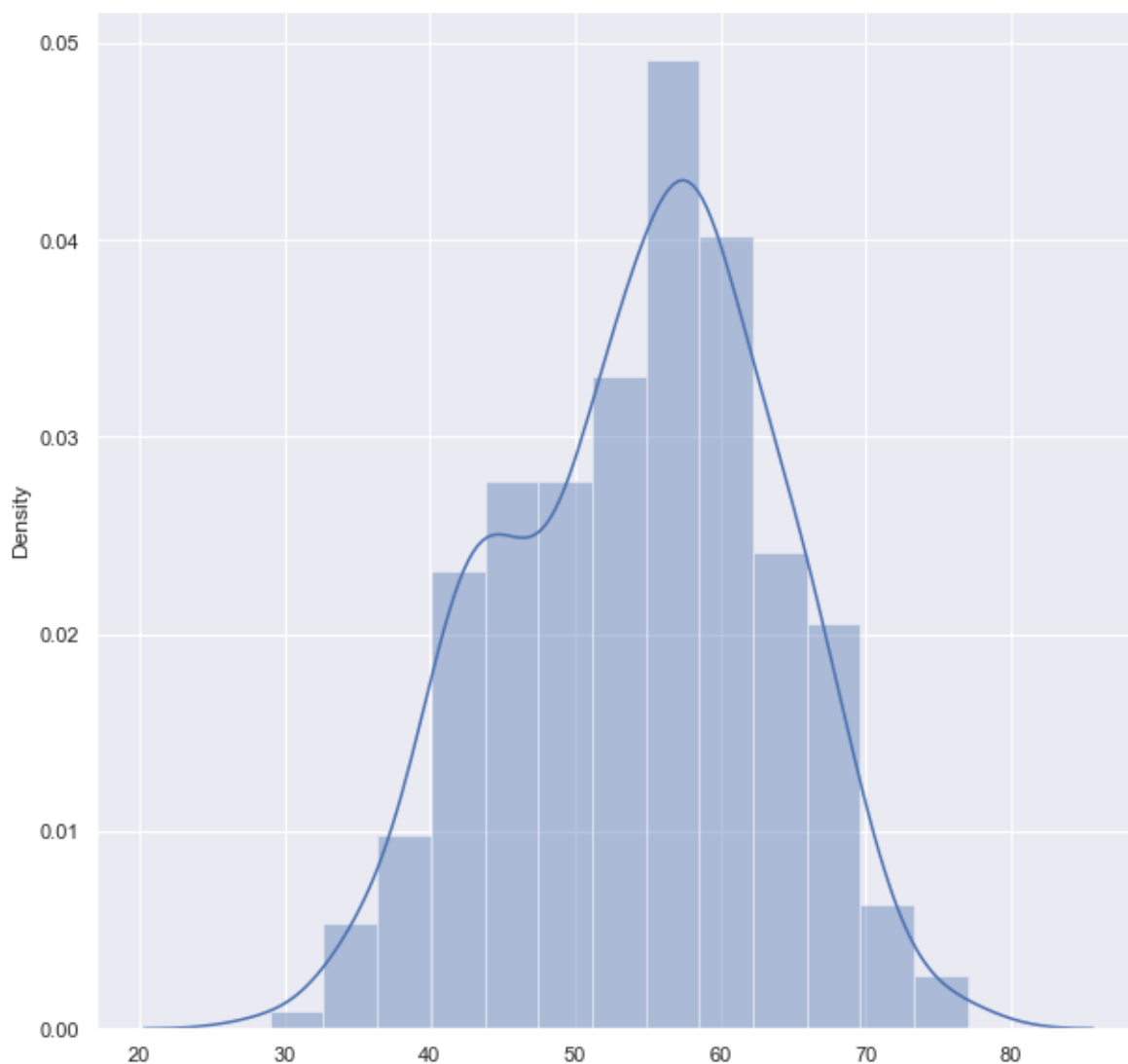
```
# Generating bar graph
fig = plt.figure(figsize=(10, 10))

# Adds subplot on position 1
ax = fig.add_subplot(111)

sns.distplot(x=heart_dataset.age)
plt.show()
```

C:\Users\ShivendraBhonsle\anaconda3\lib\site-packages\seaborn\distributions.py:2557: 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).

warnings.warn(msg, FutureWarning)



## 5) Checking correlation using heatmap

In [14]:

```
# Generating bar graph
fig = plt.figure(figsize=(15, 15))

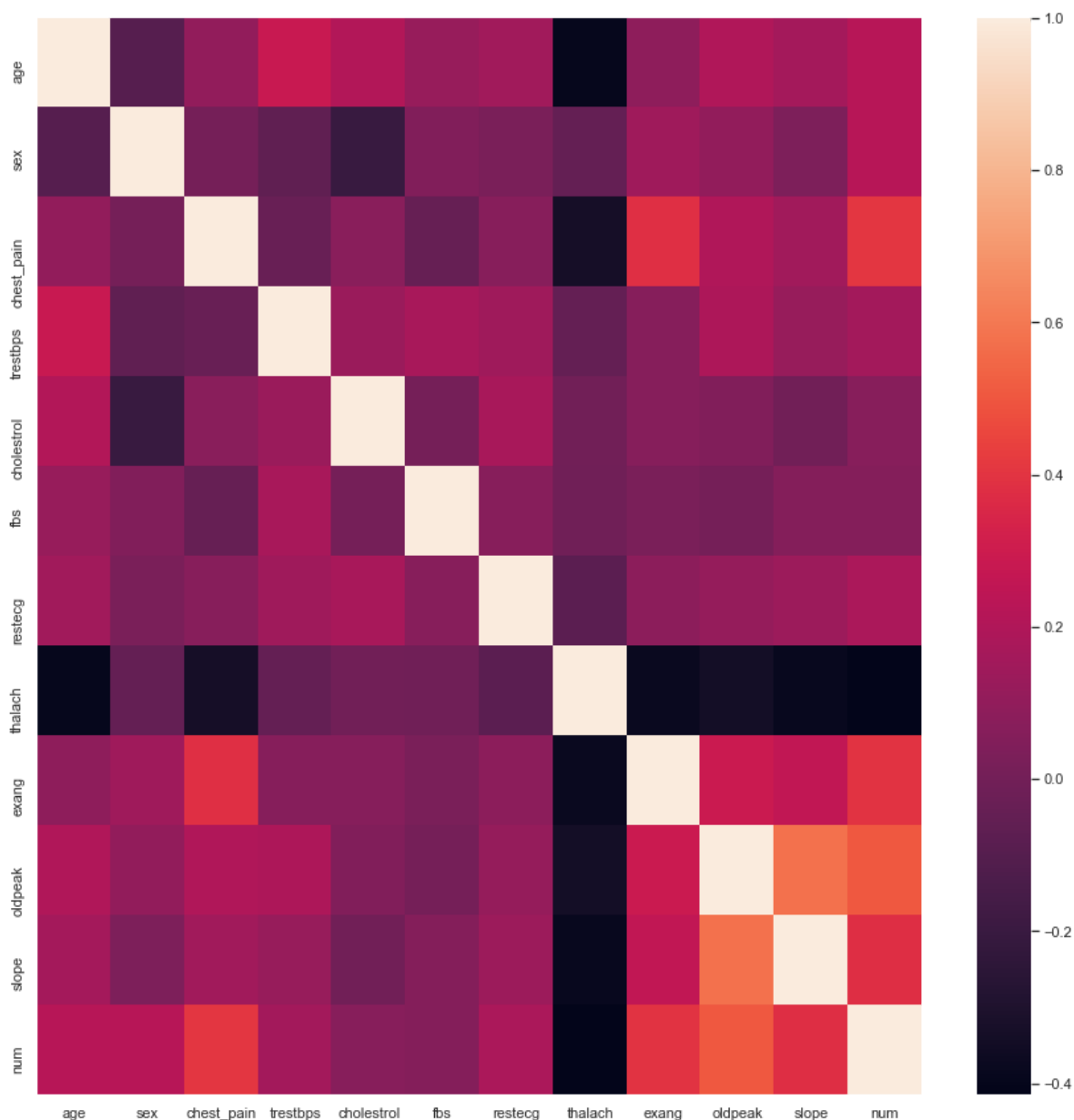
# Adds subplot on position 1
ax = fig.add_subplot(111)

sns.heatmap(heart_dataset.corr())

plt.plot()
```

Out[14]:

[]



# Conclusion

1. Implemented following visualization methods :

- A. Pie chart
- B. Bar chart
- C. Count plot
- D. Box plot
- E. Distribution plot (Histogram)
- F. Heatmap

## End of Notebook