BM20BTECH11001-Lab1

September 26, 2021

0.0.1 Reading a file :

[1]:

```
def read_data(filename):
    with open(filename,'r') as f:
        text = f.readlines()
        print(text)
read_data('welcome.txt')
```

```
['Hello everyone!\n', 'Welcome to the course BM2033-Probability and Random
Processes. This course requires a background in libraries like
Pandas,matplotlib,numpy etc.\n', 'First class is on Thursday, 23rd Sept 2021 at
2.30 pm.\n']
```

[2]:

```
#Printing line by line
file = open('welcome.txt')

store_line = []

vowel_counts = {
    'a':0,
    'e':0,
    'i':0,
    'o':0,
    'u':0
}

for line in file:
    print(line)
file.close()

#Counting number of vowels
with open('welcome.txt') as f:
  while True:
    c = f.read(1)
    c = c.lower()
    if c == 'a':
        vowel_counts['a'] = vowel_counts['a'] + 1
    if c == 'e':
        vowel_counts['e'] = vowel_counts['e'] + 1
    if c == 'i':
        vowel_counts['i'] = vowel_counts['i'] + 1
    if c == 'o':
        vowel_counts['o'] = vowel_counts['o'] + 1
    if c == 'u':
        vowel_counts['u'] = vowel_counts['u'] + 1
    if not c:
      break
display_vowels = "The number of vowels is {}.\n"
print(display_vowels.
 ↪format(vowel_counts['a']+vowel_counts['e']+vowel_counts['i']+vowel_counts['o']+vowel_counts

#Find frequency of each word
words = []
word_frequency = {}
words_distinct = []
with open('welcome.txt') as f:
    for line in f:
        for word in line.split():
            words.append(word)
    for word in words:
        if word not in words_distinct:
            words_distinct.append(word)
    for word in words_distinct:
        word_frequency[word] = words.count(word)
    print(word_frequency,"\n")

#Writing file from last to first
with open('welcome.txt') as file:
    for line in file:
        store_line.append(line)
    for line in range(len(store_line)-1,-1,-1):
        print(store_line[line])
```

```
Hello everyone!

Welcome to the course BM2033-Probability and Random Processes. This course
requires a background in libraries like Pandas,matplotlib,numpy etc.

First class is on Thursday, 23rd Sept 2021 at 2.30 pm.

The number of vowels is 58.

{'Hello': 1, 'everyone!': 1, 'Welcome': 1, 'to': 1, 'the': 1, 'course': 2,
'BM2033-Probability': 1, 'and': 1, 'Random': 1, 'Processes.': 1, 'This': 1,
'requires': 1, 'a': 1, 'background': 1, 'in': 1, 'libraries': 1, 'like': 1,
'Pandas,matplotlib,numpy': 1, 'etc.': 1, 'First': 1, 'class': 1, 'is': 1, 'on':
1, 'Thursday,': 1, '23rd': 1, 'Sept': 1, '2021': 1, 'at': 1, '2.30': 1, 'pm.':
1}

First class is on Thursday, 23rd Sept 2021 at 2.30 pm.

Welcome to the course BM2033-Probability and Random Processes. This course
requires a background in libraries like Pandas,matplotlib,numpy etc.

Hello everyone!
```

0.0.2 Some exercises that you can do :

- print each line separately
- counting the number of vowels
- find frequency of each word
- write the file from last to first
- etc!!

0.0.3 Reading and parsing Tabular data:

[3]:
```
import pandas as pd
data=pd.read_csv('iris.csv')
print(data)
```

```
     sepal.length  sepal.width  petal.length  petal.width    variety
0             5.1          3.5           1.4          0.2     Setosa
1             4.9          3.0           1.4          0.2     Setosa
2             4.7          3.2           1.3          0.2     Setosa
3             4.6          3.1           1.5          0.2     Setosa
4             5.0          3.6           1.4          0.2     Setosa
..            …          …           …          …        …
145           6.7          3.0           5.2          2.3  Virginica
146           6.3          2.5           5.0          1.9  Virginica
147           6.5          3.0           5.2          2.0  Virginica
148           6.2          3.4           5.4          2.3  Virginica
149           5.9          3.0           5.1          1.8  Virginica

[150 rows x 5 columns]
```

Read the sepal.length and sepal.width columns

[4]:
```
#read the sepal length column
sepal_info = data[['sepal.length','sepal.width']]
sepal_info
```

```
[4]:      sepal.length  sepal.width
     0             5.1          3.5
     1             4.9          3.0
     2             4.7          3.2
     3             4.6          3.1
     4             5.0          3.6
     ..            …          …
     145           6.7          3.0
     146           6.3          2.5
     147           6.5          3.0
     148           6.2          3.4
     149           5.9          3.0

     [150 rows x 2 columns]
```

Selecting columns based on conditions

```
[5]: data[data['sepal.length']>=5.0]
```

```
[5]:      sepal.length  sepal.width  petal.length  petal.width    variety
     0             5.1          3.5           1.4          0.2     Setosa
     4             5.0          3.6           1.4          0.2     Setosa
     5             5.4          3.9           1.7          0.4     Setosa
     7             5.0          3.4           1.5          0.2     Setosa
     10            5.4          3.7           1.5          0.2     Setosa
     ..            …          …          …          …          …
     145           6.7          3.0           5.2          2.3  Virginica
     146           6.3          2.5           5.0          1.9  Virginica
     147           6.5          3.0           5.2          2.0  Virginica
     148           6.2          3.4           5.4          2.3  Virginica
     149           5.9          3.0           5.1          1.8  Virginica

     [128 rows x 5 columns]
```

0.0.4 Some exercises that you can do :

- slice rows and columns based indices
- find statistical information from data (Ex. Mean, Median, Std. Deviation etc)
- group data based on conditions
- sort data
- try loading other datasets and perform calculations based on column values etc
- write your results to a file
- etc!

```
[6]: #Slice rows and columns based indices
     data_df = pd.DataFrame(data)
     data_df.iloc[1:51, :4]
```

```
[6]:      sepal.length  sepal.width  petal.length  petal.width
     1             4.9          3.0           1.4          0.2
     2             4.7          3.2           1.3          0.2
     3             4.6          3.1           1.5          0.2
     4             5.0          3.6           1.4          0.2
     5             5.4          3.9           1.7          0.4
     6             4.6          3.4           1.4          0.3
     7             5.0          3.4           1.5          0.2
     8             4.4          2.9           1.4          0.2
     9             4.9          3.1           1.5          0.1
     10            5.4          3.7           1.5          0.2
     11            4.8          3.4           1.6          0.2
     12            4.8          3.0           1.4          0.1
     13            4.3          3.0           1.1          0.1
     14            5.8          4.0           1.2          0.2
     15            5.7          4.4           1.5          0.4
     16            5.4          3.9           1.3          0.4
     17            5.1          3.5           1.4          0.3
     18            5.7          3.8           1.7          0.3
     19            5.1          3.8           1.5          0.3
     20            5.4          3.4           1.7          0.2
     21            5.1          3.7           1.5          0.4
     22            4.6          3.6           1.0          0.2
     23            5.1          3.3           1.7          0.5
     24            4.8          3.4           1.9          0.2
     25            5.0          3.0           1.6          0.2
     26            5.0          3.4           1.6          0.4
     27            5.2          3.5           1.5          0.2
     28            5.2          3.4           1.4          0.2
     29            4.7          3.2           1.6          0.2
     30            4.8          3.1           1.6          0.2
     31            5.4          3.4           1.5          0.4
     32            5.2          4.1           1.5          0.1
     33            5.5          4.2           1.4          0.2
     34            4.9          3.1           1.5          0.2
     35            5.0          3.2           1.2          0.2
     36            5.5          3.5           1.3          0.2
     37            4.9          3.6           1.4          0.1
     38            4.4          3.0           1.3          0.2
     39            5.1          3.4           1.5          0.2
     40            5.0          3.5           1.3          0.3
     41            4.5          2.3           1.3          0.3
     42            4.4          3.2           1.3          0.2
     43            5.0          3.5           1.6          0.6
     44            5.1          3.8           1.9          0.4
     45            4.8          3.0           1.4          0.3
     46            5.1          3.8           1.6          0.2
```

```
47          4.6          3.2          1.4          0.2
48          5.3          3.7          1.5          0.2
49          5.0          3.3          1.4          0.2
50          7.0          3.2          4.7          1.4
```

```
[7]: #Statistics of data
     data_df.describe()
```

```
[7]:        sepal.length  sepal.width  petal.length  petal.width
     count    150.000000   150.000000    150.000000   150.000000
     mean       5.843333     3.057333      3.758000     1.199333
     std        0.828066     0.435866      1.765298     0.762238
     min        4.300000     2.000000      1.000000     0.100000
     25%        5.100000     2.800000      1.600000     0.300000
     50%        5.800000     3.000000      4.350000     1.300000
     75%        6.400000     3.300000      5.100000     1.800000
     max        7.900000     4.400000      6.900000     2.500000
```

```
[8]: #Grouping data based on conditions
     flower_large = pd.DataFrame(data_df[(data_df['sepal.length']>6) &␣
      ↪(data_df['petal.length']>4)])
     flower_large.describe()
```

```
[8]:        sepal.length  sepal.width  petal.length  petal.width
     count     60.000000    60.000000     60.000000    60.000000
     mean       6.680000     2.991667      5.345000     1.856667
     std        0.475787     0.304370      0.666746     0.377488
     min        6.100000     2.200000      4.300000     1.200000
     25%        6.300000     2.800000      4.775000     1.500000
     50%        6.550000     3.000000      5.350000     1.800000
     75%        6.900000     3.200000      5.800000     2.200000
     max        7.900000     3.800000      6.900000     2.500000
```

```
[9]: #Grouping data based on variety of flowers(mean)
     data_df.groupby(data_df['variety']).median()
```

```
[9]:             sepal.length  sepal.width  petal.length  petal.width
     variety
     Setosa               5.0          3.4          1.50          0.2
     Versicolor           5.9          2.8          4.35          1.3
     Virginica            6.5          3.0          5.55          2.0
```

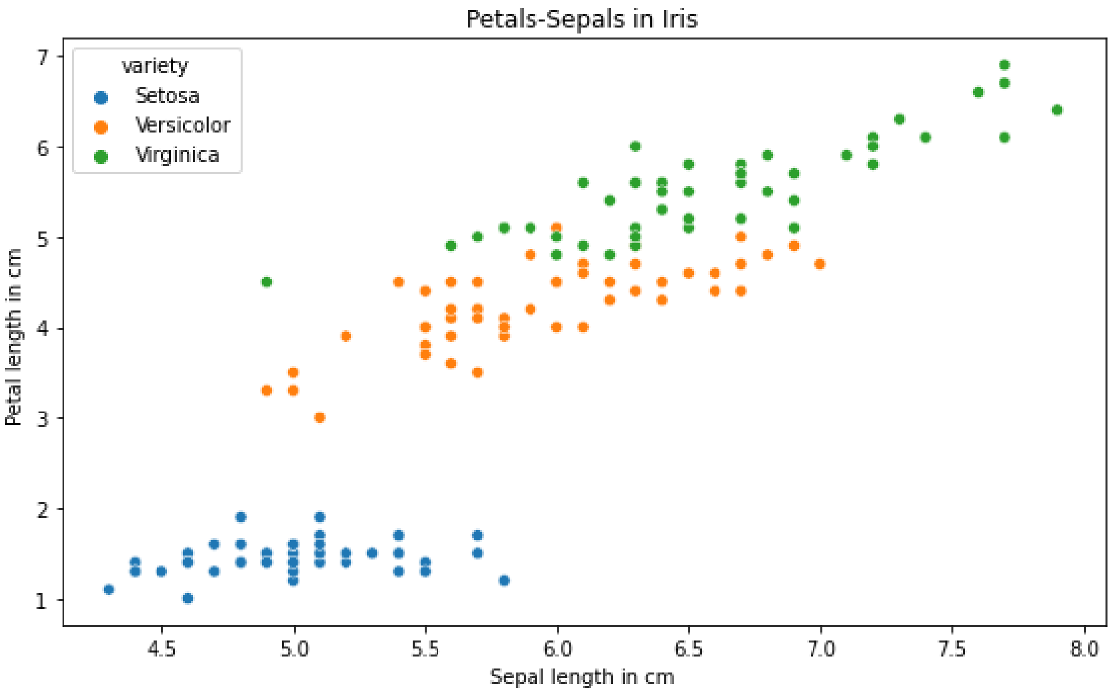
```
[10]: #Sort the data
      data_df.sort_values(by=['sepal.length','sepal.width','petal.length','petal.
       ↪width'])
```

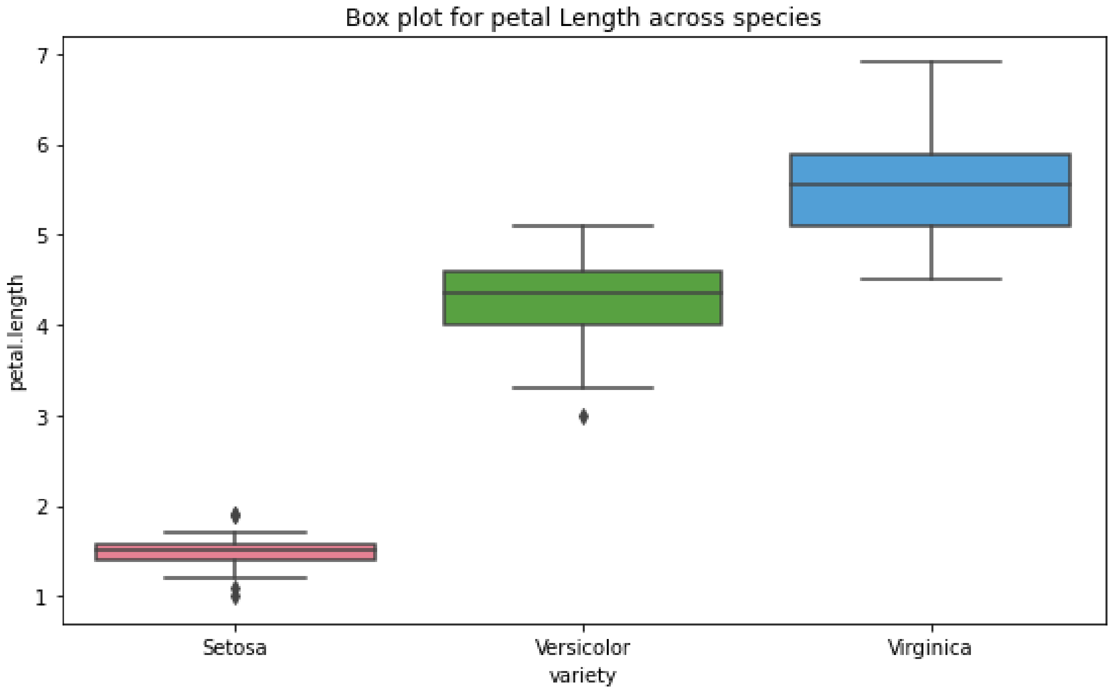
[10]:
```
     sepal.length  sepal.width  petal.length  petal.width    variety
13            4.3          3.0           1.1          0.1     Setosa
8             4.4          2.9           1.4          0.2     Setosa
38            4.4          3.0           1.3          0.2     Setosa
42            4.4          3.2           1.3          0.2     Setosa
41            4.5          2.3           1.3          0.3     Setosa
..            …            …             …            …          …
118           7.7          2.6           6.9          2.3  Virginica
122           7.7          2.8           6.7          2.0  Virginica
135           7.7          3.0           6.1          2.3  Virginica
117           7.7          3.8           6.7          2.2  Virginica
131           7.9          3.8           6.4          2.0  Virginica

[150 rows x 5 columns]
```

[11]:
```python
#Using external datasets(covid-19 India stats)
covid = pd.DataFrame(pd.read_csv('india_covid19.csv'))
covid = covid.sort_values(by=['Active','Total Cases'])
stats = covid['Total Cases'].describe()
f = open('covid_stats_india.txt','w')
f.writelines(str(stats))
f.close()
```

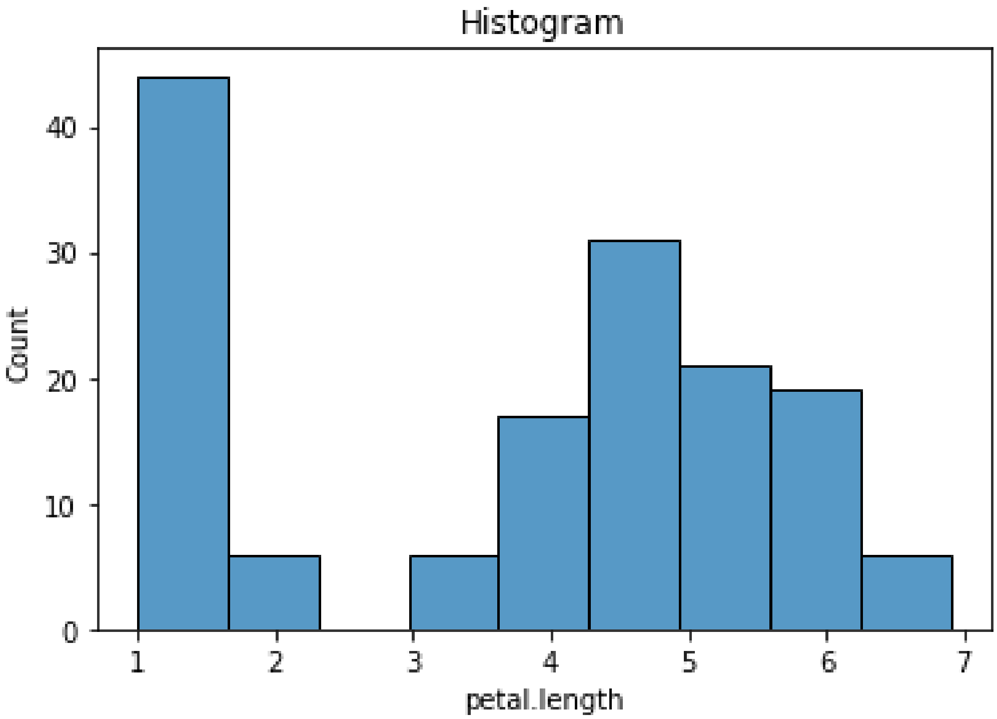
0.0.5 Plots and visualizations

[12]:
```python
import seaborn as sb
import matplotlib
import matplotlib.pyplot as plt

fig, axs = plt.subplots(2,figsize=(8,10))
axs[0].set_title('Petals-Sepals in Iris')
axs[0].set_xlabel('Sepal length in cm');
axs[0].set_ylabel('Petal length in cm ');
sb.scatterplot(x=data['sepal.length'],y=data['petal.length'],hue=data.
 ↪variety,ax=axs[0]);

axs[1].set_title('Box plot for petal Length across species')
sb.boxplot(x="variety", y="petal.length", palette="husl", data=data,ax=axs[1])
fig.tight_layout()
```





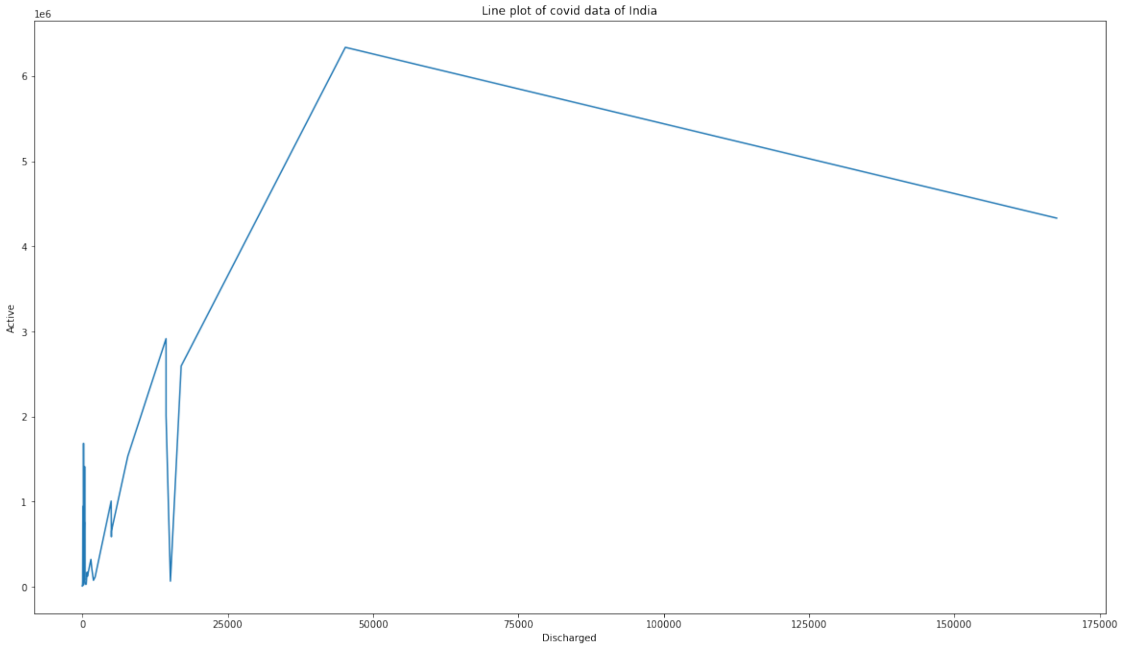
```
[13]: t = sb.histplot(data=data, x="petal.length")
      t.set_title('Histogram');
```

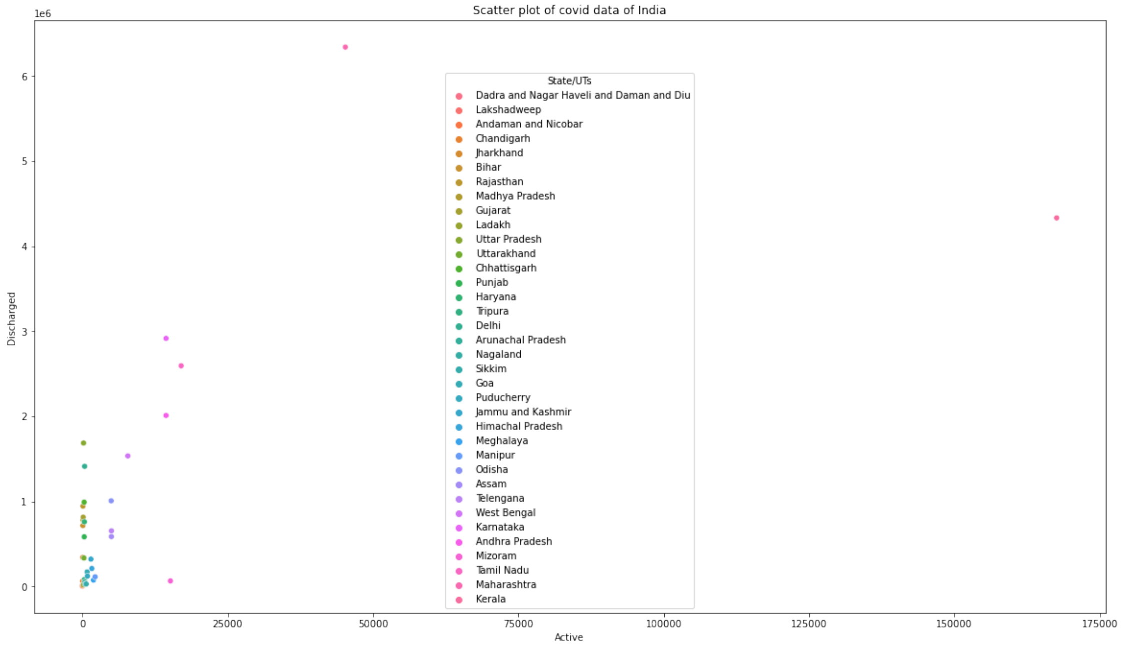


0.0.6 Some exercises that you can do :

- explore matplotlib, seaborn and other plotting libraries
- Try different kinds of plots like : line plots, scatterplots, bargraphs, pie charts, histograms,heatmaps etc
- plot 2D, 3D plots
- etc!

```
[14]: #Line plot and scatter plot on covid data of India
fig, axs = plt.subplots(2, figsize=(20, 25))
axs[0].set_title("Line plot of covid data of India")
axs[0].set_xlabel("Discharged")
axs[0].set_ylabel("Active")
sb.lineplot(x=covid['Active'], y=covid['Discharged'], ax=axs[0]);
axs[1].set_title("Scatter plot of covid data of India")
sb.scatterplot(x=covid['Active'], y=covid['Discharged'], hue=covid['State/
 ↪UTs'], ax=axs[1]);
```

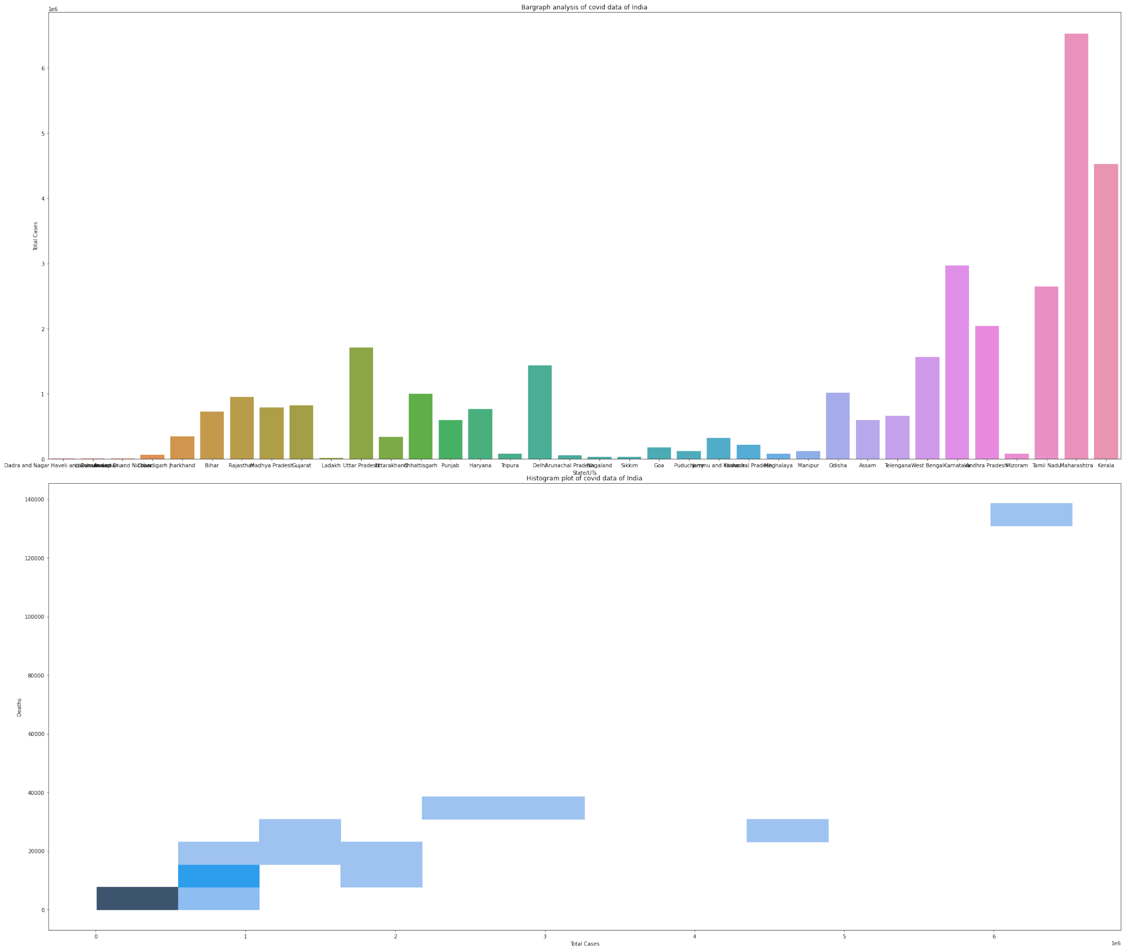




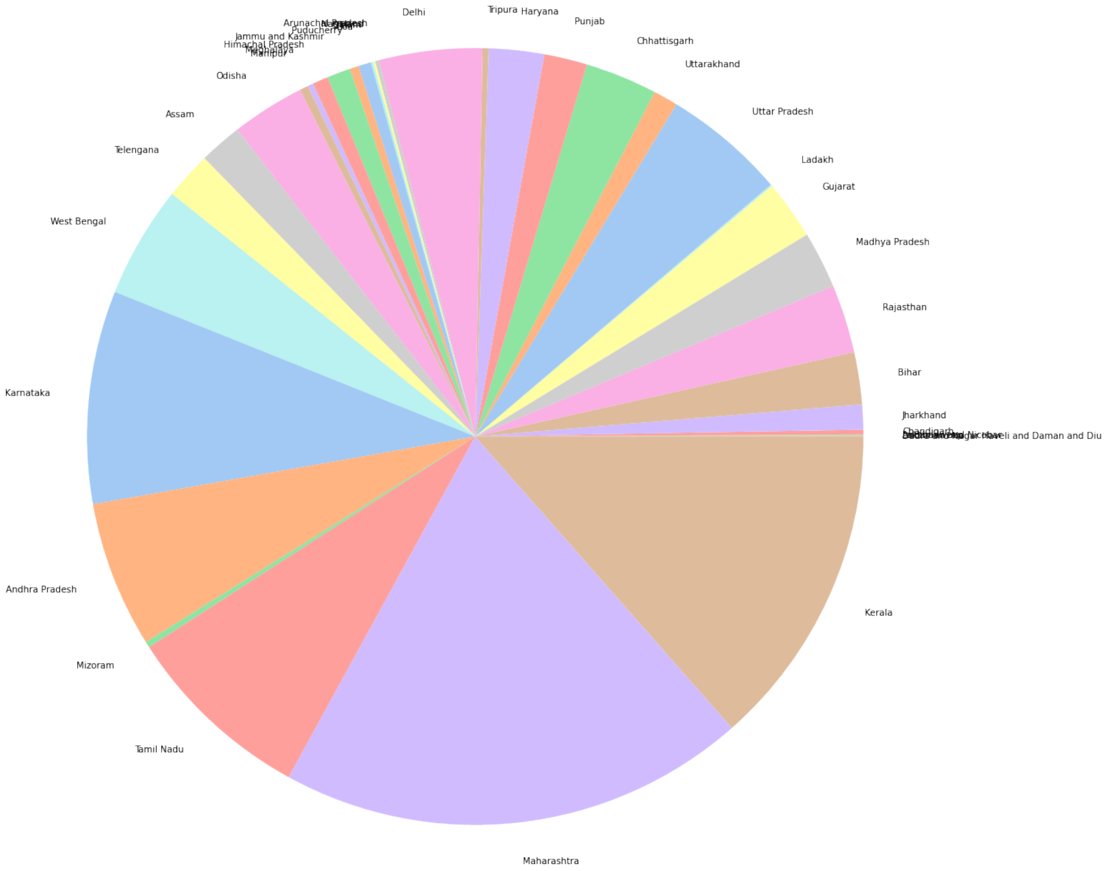
[15]:

```
#Plotting bargraph and histogram on covid19 data of India
fig, axs = plt.subplots(2, figsize=(30,25))
axs[0].set_title('Bargraph analysis of covid data of India')
axs[0].set_xlabel('State')
axs[0].set_ylabel('Total Cases')
sb.barplot(x=covid['State/UTs'], y=covid['Total Cases'], ax=axs[0]);
```

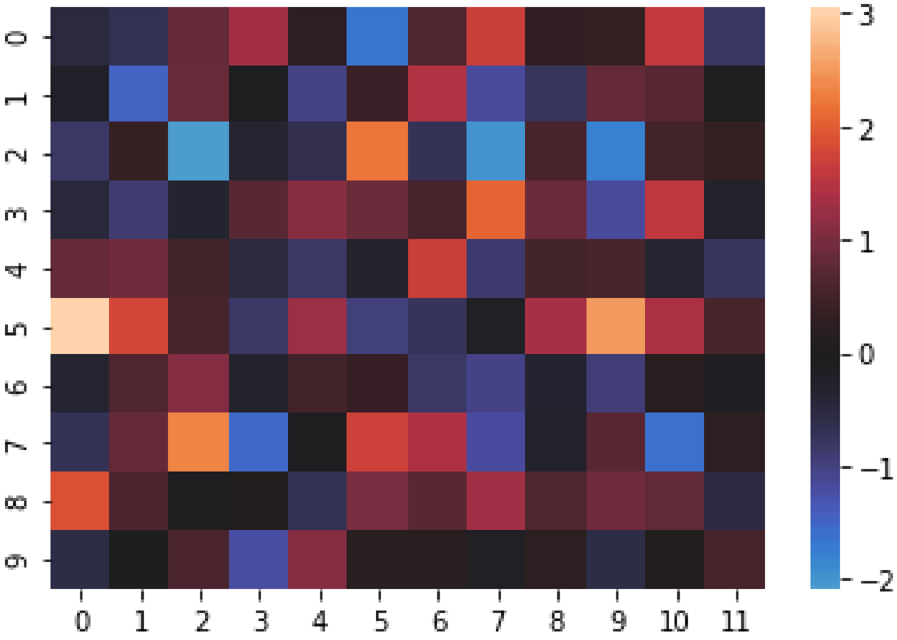
```
fig.tight_layout()
axs[1].set_title('Histogram plot of covid data of India')
sb.histplot(x=covid['Total Cases'], y=covid['Deaths'], ax=axs[1]);
```



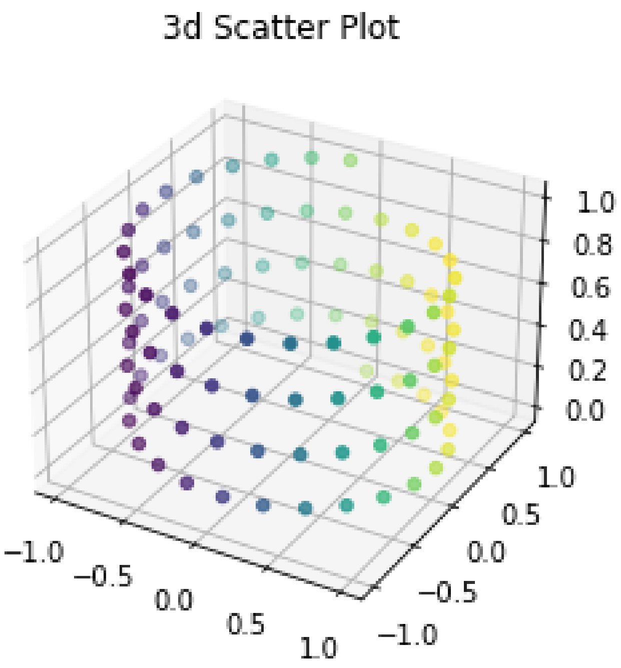
[16]:
```
#Plotting piecharts on covid19 data of India
plt.figure(figsize=(20,20))
plt.pie(covid['Total Cases'], labels=covid['State/UTs'], colors = sb.
 ↪color_palette('pastel')[0:35])
plt.show()
```



[17]:
```
#Plotting a heatmap
import numpy as np
normal_data = np.random.randn(10, 12)
normal_data
ax = sb.heatmap(normal_data, center=0)
```



[18]:
```
#Plotting 3D graphs
from mpl_toolkits import mplot3d
fig = plt.figure()
ax = plt.axes(projection ='3d')
z = np.linspace(0,1,100)
x = np.sin(25*z)
y = np.cos(25*z)
c = x+y
ax.scatter(x,y,z,c=c)
ax.set_title("3d Scatter Plot")
plt.show()
```



0.0.7 Read Image

[19]:

```
from PIL import Image
import numpy as np


img = Image.open('lena.png')

img.show()

image_data =np.array(img)
image_data
```

[19]:

```
array([[162, 162, 162, …, 170, 155, 128],
       [162, 162, 162, …, 170, 155, 128],
       [162, 162, 162, …, 170, 155, 128],
       …,
       [ 43,  43,  50, …, 104, 100,  98],
       [ 44,  44,  55, …, 104, 105, 108],
       [ 44,  44,  55, …, 104, 105, 108]], dtype=uint8)
```

0.1 References :

- Python Installation
- Miniconda Installation

- Basic Python
- Pandas
- Matplotlib
- Seaborn
- Numpy