Project On Pandas and Matplotlib

Create a DataFrame

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Firstly get the how many numbers of column are required
number_of_columns = int(input("Enter the number of columns : "))

columns = []

for i in range(number_of_columns):
# Then add the column names in that column
    columns.append(input("Enter the column name :"))


dct = {}

# Then get the how many percentage data store in the column
number_of_percentage_in_each_col = int(input("enter the number of
percentage of  in each column  : "))

for i in range(number_of_columns):

    lst = []

    for j in range(number_of_percentage_in_each_col):
# Then store the each percentage of student in the list
        lst.append(input(f"enter the percentage of column
{columns[i]}  : "))


# After that store the percenatge values in the  list data into the
dictionary
        dct[columns[i]] = lst
    dct[columns[i]] = lst
Data = pd.DataFrame(dct)
print(Data)


Enter the number of columns : 3
Enter the column name :Class1
Enter the column name :Class2
Enter the column name :Class3
enter the number of percentage of  in each column  : 5
```

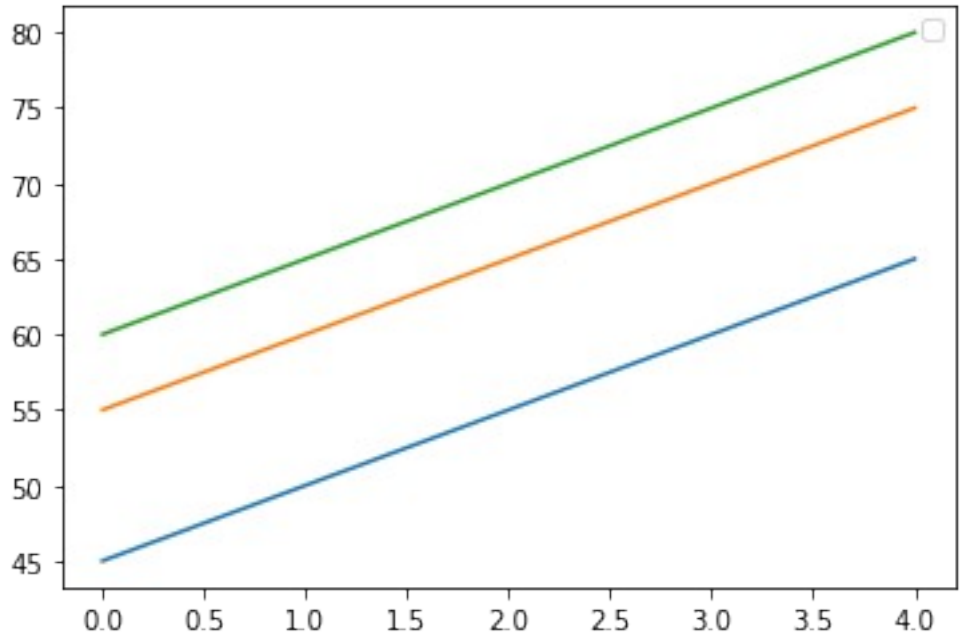
```
enter the percentage of column Class1  : 45
enter the percentage of column Class1  : 50
enter the percentage of column Class1  : 55
enter the percentage of column Class1  : 60
enter the percentage of column Class1  : 65
enter the percentage of column Class2  : 55
enter the percentage of column Class2  : 60
enter the percentage of column Class2  : 65
enter the percentage of column Class2  : 70
enter the percentage of column Class2  : 75
enter the percentage of column Class3  : 60
enter the percentage of column Class3  : 65
enter the percentage of column Class3  : 70
enter the percentage of column Class3  : 75
enter the percentage of column Class3  : 80
  Class1 Class2 Class3
0     45     55     60
1     50     60     65
2     55     65     70
3     60     70     75
4     65     75     80
```

columns of the DataFrame and plot a Comparison Plot(plot every data in single plot) With different Colors

```
import numpy as np
Class1 = np.array([45,50,55,60,65])
Class2 = np.array([55,60,65,70,75])
Class3 = np.array([60,65,70,75,80])
plt.plot(Class1)
plt.plot(Class2)
plt.plot(Class3)
plt.legend()
```

```
No artists with labels found to put in legend.  Note that artists
whose label start with an underscore are ignored when legend() is
called with no argument.
```

```
<matplotlib.legend.Legend at 0x19b2c70a970>
```



compare the data using Line Plot as well as Scatter Plot.

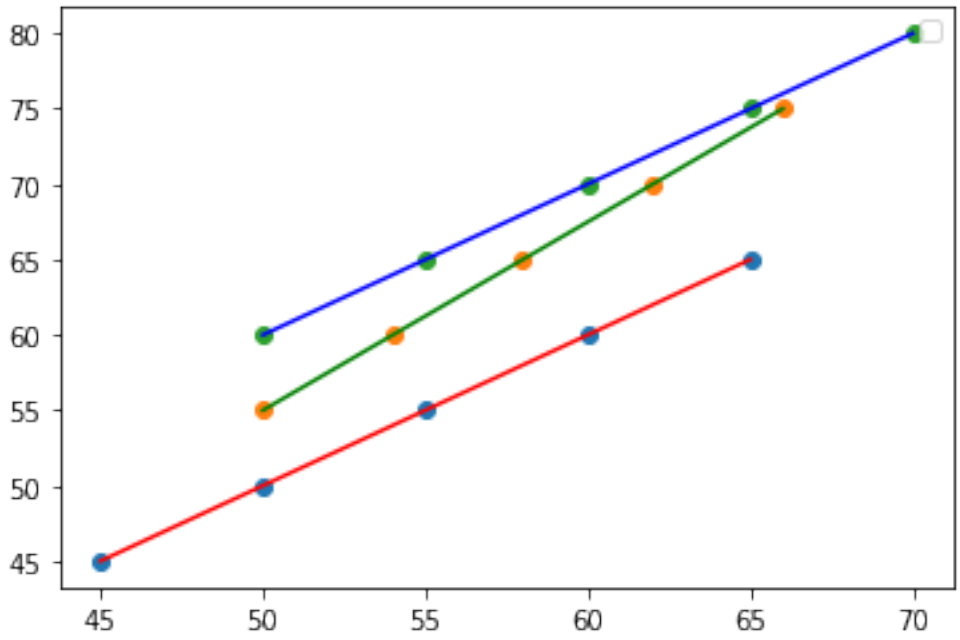
```
import numpy as np
Class_0 = np.array([45,50,55,60,65])
Class_1 = np.array([50,54,58,62,66])
Class_0 = np.array([45,50,55,60,65])
Class_2 = np.array([50,55,60,65,70])
Class_0 = np.array([45,50,55,60,65])
Class_3 = np.array([55,60,65,70,75])

plt.plot(Class_0,Class1,"r")
plt.plot(Class_1,Class2,"g")
plt.plot(Class_2,Class3,"b")
plt.legend()
```

```
Class_0 = np.array([45,50,55,60,65])
Class_1 = np.array([50,54,58,62,66])
Class_0 = np.array([45,50,55,60,65])
Class_2 = np.array([50,55,60,65,70])
Class_0 = np.array([45,50,55,60,65])
Class_3 = np.array([55,60,65,70,75])
plt.scatter(Class_0,Class1)
plt.scatter(Class_1,Class2)
plt.scatter(Class_2,Class3)
plt.legend()
```

```
No artists with labels found to put in legend.  Note that artists
whose label start with an underscore are ignored when legend() is
called with no argument.
No artists with labels found to put in legend.  Note that artists
whose label start with an underscore are ignored when legend() is
called with no argument.

<matplotlib.legend.Legend at 0x19b2c6dac40>
```



compare the data using there subplots

```
Class_0 = np.array([45,50,55,60,65])
Class_1 = np.array([50,54,58,62,66])

plt.subplot(1,2,1)
plt.plot(Class_0,Class1,marker = "o",color = "r")
plt.legend()

Class_0 = np.array([45,50,55,60,65])
Class_2 = np.array([50,55,60,65,70])
plt.subplot(1,2,2)
plt.plot(Class_0,Class2,marker = "*",color = "g")
plt.legend()

Class_0 = np.array([45,50,55,60,65])
Class_3 = np.array([55,60,65,70,75])

plt.subplot(1,2,2)
```

```
plt.plot(Class_0,Class3,marker = "^",color = "Pink")
plt.legend()
```

```
No artists with labels found to put in legend.  Note that artists
whose label start with an underscore are ignored when legend() is
called with no argument.
No artists with labels found to put in legend.  Note that artists
whose label start with an underscore are ignored when legend() is
called with no argument.
No artists with labels found to put in legend.  Note that artists
whose label start with an underscore are ignored when legend() is
called with no argument.
```

```
<matplotlib.legend.Legend at 0x19b2c757fa0>
```