```
import numpy as py
#creating array
arr=np.array([1,2,3,4,5])
print(arr)
zeros_arr=np.zeros((3,3),dtype=int)
print(zeros_arr)
ones_arr=np.ones((2,2),dtype=int)
print(ones_arr)
arange_arr=np.arange(10)
print(arange_arr)
#manuplating array
reshaped_arr=arr.reshape(5,1)
print(reshaped_arr)
sliced_arr=arr[2:4]
print(sliced_arr)
a1=np.array([1,2,3,4])
a2=np.array([1,2,3,4])
a3=np.vstack(a1+a2)
print(a3)
a3=np.stack(a1+a2)
print(a3)
b=np.split(a1,2)
print(b)
n=np.array([[1,2],[3,4]])
c=n.T
print(c)
#linear algebra with numpy
v1=np.array([[1,2],[3,4]])
v2=np.array([[5,6],[7,8]])
v3=np.dot(v1,v2)
print(v3)
v4=np.linalg.eig(v3)
print(v4)
```

```
[1 2 3 4 5]
[[0 0 0]
 [0 0 0]
 [0 0 0]]
[[1 1]
 [1 1]]
[0 1 2 3 4 5 6 7 8 9]
[[1]
 [2]
 [3]
 [4]
 [5]]
[3 4]
[[2]
 [4]
 [6]
 [8]]
[2 4 6 8]
[array([1, 2]), array([3, 4])]
[[1 3]
 [2 4]]
[[19 22]
 [43 50]]
(array([5.80198014e-02, 6.89419802e+01]), array([[-0.75781077, -0.40313049],
       [ 0.65247439, -0.91514251]]))
```

```
a1=np.array([1,2,3,4])
a2=np.array([1,2,3,4])
a3=np.add(a1,a2)
print(a3)
```

```
a1=np.array([1,2,3,4])
print(a1+3)
```

```
[4 5 6 7]
```

```
a1=np.array([[1,2,3],[4,5,6]])
b=np.sum(a1)
a2=np.array([[1,2,3],[4,5,6]])
print(b)
b=np.sum(a1+a2)
print(b)
print(np.sum(a1,axis=0))
print(np.sum(a1,axis=0))
```

```
#statistical operations
a1=np.array([1,2,3,4,5])
m=np.mean(a1)
m2=np.median(a1)
v1=np.var(a1)
sd=np.std(a1)
print(m,m2,v1,sd)

d=np.loadtxt("/content/1 2 3.txt")
d=np.savetxt("/content/date.txt",d)
print(d)
```

```
3.0 3.0 2.0 1.4142135623730951
None
```

```
import matplotlib.pyplot as plt
a=np.array([1,2,3,4,5,6,7,8,9,10])

plt.plot(a)
```

```
#importing pandas
import pandas as pd
a=["vasanthi","dhana","lalitha","vani","usha","devi"]
r=pd.Series(a,index=[44,45,33,42,41,23])
print(r)
df=pd.read_csv("/content/Rohit-Sharma-All-International-Cricket-Centuries.csv",sep=" , ")
print(df)
```

```
import pandas as pd
f=pd.read_csv("/content/txtfile (1).txt")

print(f)
```

```
import pandas as pd
f=pd.read_csv("/content/txtfile (1).txt")
print(f.loc[1])#which row ,col
```

```
df=pd.read_excel("/content/Historicalinvesttemp.xlsx")
print(df)
```

```
   Unnamed: 0                       Unnamed: 1 Unnamed: 2 Unnamed: 3
0         NaN                              NaN        NaN        NaN
1         NaN                              NaN        NaN        NaN
2         NaN                              NaN        NaN        NaN
3         NaN                              NaN        NaN        NaN
4         NaN  Annual Returns on Investments in        NaN        NaN
..        ...                              ...        ...        ...
85       2007                           0.0549     0.0988     0.0466
86       2008                            -0.37     0.2587      0.016
87       2009                           0.2646     -0.149      0.001
88        NaN                           stocks     tbills      bonds
89   averages                              NaN        NaN        NaN

[90 rows x 4 columns]
```

```
df=pd.read_excel("/content/Historicalinvesttemp.xlsx",sheet_name=1)
print(df)
```

```
  Unnamed: 0 Annual Returns on Investments in Unnamed: 2 Unnamed: 3
0       Year                           Stocks    T.Bills    T.Bonds
1       1928                           0.4381     0.0308     0.0084
2       1929                           -0.083     0.0316      0.042
3       1930                          -0.2512     0.0455     0.0454
4       1931                          -0.4384     0.0231    -0.0256
5       1932                          -0.0864     0.0107     0.0879
6       1933                           0.4998     0.0096     0.0186
7       1934                          -0.0119      0.003     0.0796
8       1935                           0.4674     0.0023     0.0447
```

```
df=pd.read_csv("/content/csvfile.csv")
mv=df['Data_value'].mean()
df=df.fillna(mv)
print(df)
```

```
df=pd.read_csv("/content/csvfile (1).csv")
print(df)
df.shape
```

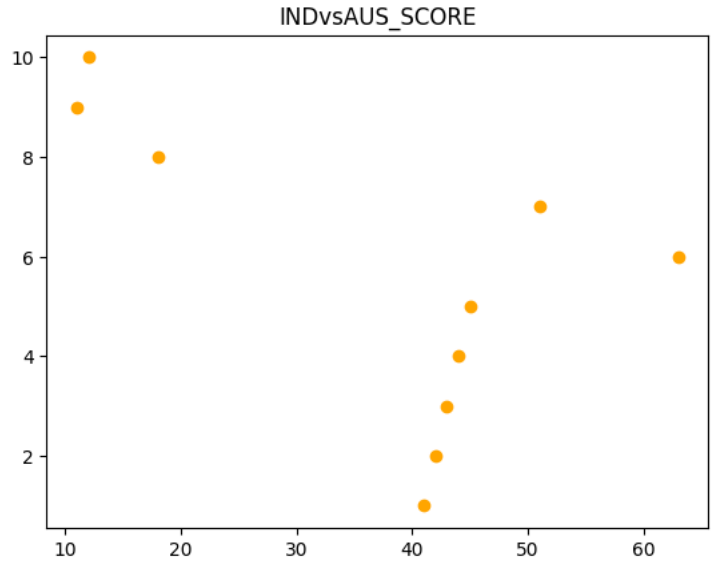
```
df=pd.read_csv("/content/csvfile (1).csv")
true=df.tail(10)
for i in range(22188,22178,-1):
  df.drop([i],axis=0, inplace=True) #removing those 10 values from original dataset
true.to_csv("manual_testing.csv")
ma=pd.read_csv("/content/manual_testing.csv")
print(ma.groupby(['Subject'])['UNITS'].count())
```

```
df=pd.read_csv("/content/csvfile (1).csv")
df=df.drop_duplicates()
print(df)
df.head(2)
```

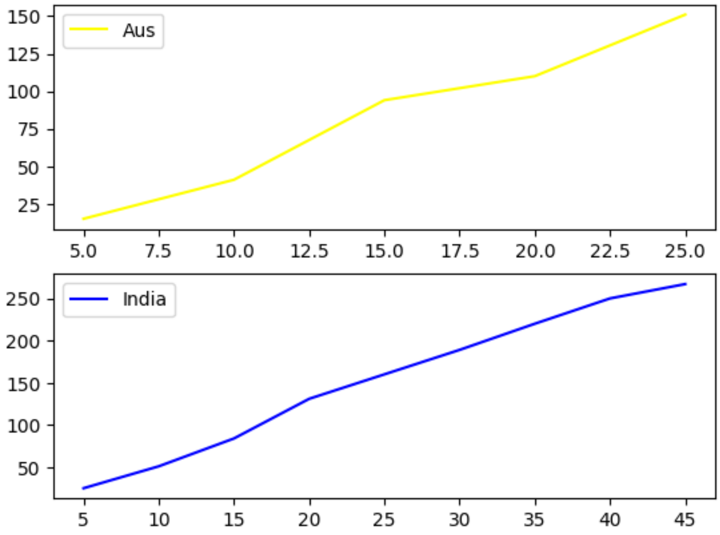
```
_pip install numpy
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import numpy as np
import matplotlib.pyplot as plt
runs=np.array([41,42,43,44,45,63,51,18,11,12])
w=np.array([1,2,3,4,5,6,7,8,9,10])
plt.scatter(runs,w,color='orange')
plt.title('INDvsAUS_SCORE')
plt.show()
```

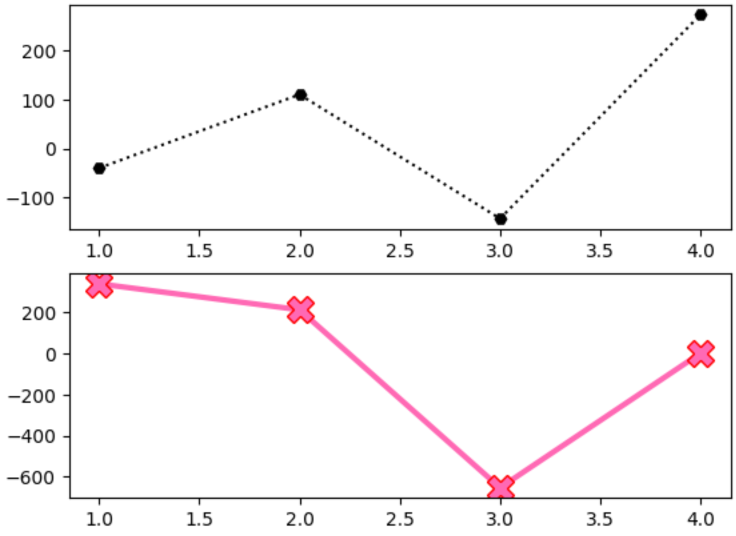


```
import numpy as np
import matplotlib.pyplot as plt
overs=np.arange(5,50,5)
overs_a=np.arange(5,30,5)
runs_i=np.array([25,51,84,131,160,189,220,250,267])
runs_a=np.array([15,41,94,110,151])
wickets=np.array([12,32,96])
plt.subplot(2,1,2)#row,col,index
plt.plot(overs,runs_i,color='blue',label='India')
plt.legend(loc='best')
plt.subplot(2,1,1)

plt.plot(overs_a,runs_a,color='yellow',label='Aus')

plt.legend(loc='best')
plt.show()
```



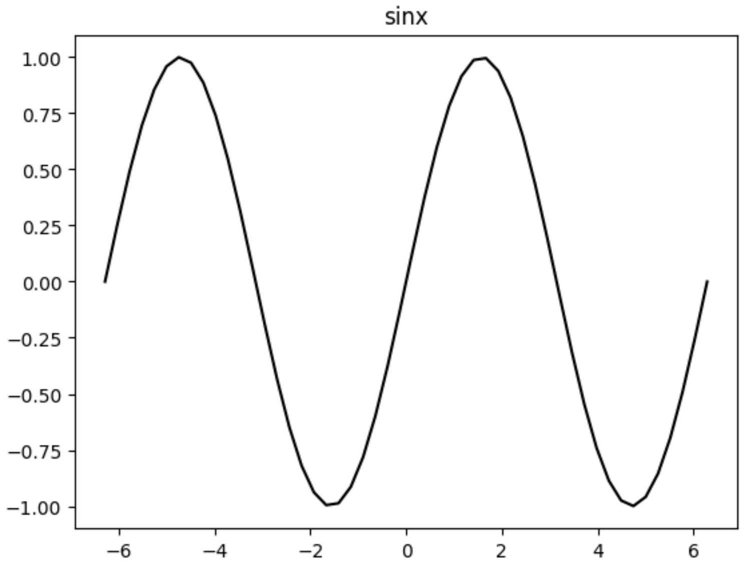
```
import matplotlib.pyplot as plt
a=[230,568,780,127,128]
b=[200,160,270,127,400]
years=[1,2,3,4]
profit_a=[(a[i]-a[i-1]) for i in range(1,len(a))]
profit_b=[(b[i]-b[i-1]) for i in range(1,len(b))]
plt.subplot(2,1,2)
plt.plot(years,profit_a,color='hotpink',linewidth='3',label='CompanyA',marker='X',ms='15',mec='r')
plt.subplot(2,1,1)
plt.plot(years,profit_b,color='black',linestyle='dotted',label='CompanyB',marker='H')
```

```
    [<matplotlib.lines.Line2D at 0x7f18b53538b0>]
```

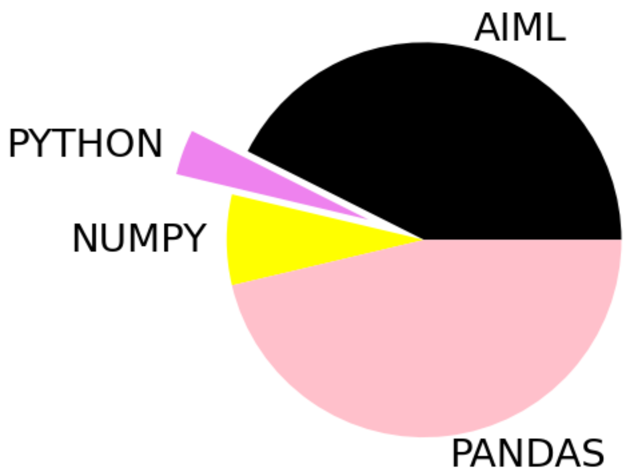


```
import numpy as np
import matplotlib.pyplot as plt
tiger=np.linspace(-2*np.pi, 2*np.pi, 50)
print(tiger)
plt.plot(tiger,np.sin(tiger),color='black')#x,y,color,lable
plt.title("sinx")
```

```
[-6.28318531 -6.02672876 -5.77027222 -5.51381568 -5.25735913 -5.00090259
 -4.74444605 -4.48798951 -4.23153296 -3.97507642 -3.71861988 -3.46216333
 -3.20570679 -2.94925025 -2.6927937  -2.43633716 -2.17988062 -1.92342407
 -1.66696753 -1.41051099 -1.15405444 -0.8975979  -0.64114136 -0.38468481
 -0.12822827  0.12822827  0.38468481  0.64114136  0.8975979   1.15405444
  1.41051099  1.66696753  1.92342407  2.17988062  2.43633716  2.6927937
  2.94925025  3.20570679  3.46216333  3.71861988  3.97507642  4.23153296
  4.48798951  4.74444605  5.00090259  5.25735913  5.51381568  5.77027222
  6.02672876  6.28318531]
Text(0.5, 1.0, 'sinx')
```

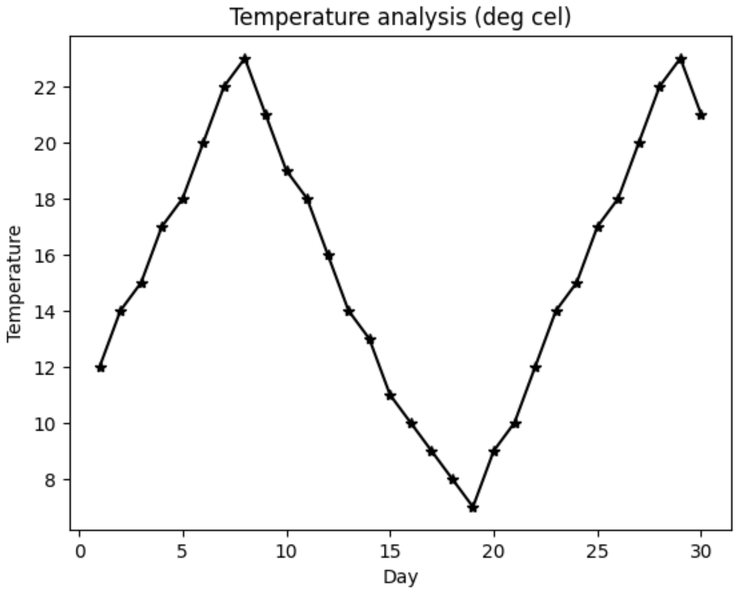


```
a=np.array([23,2,4,25])
labe=["AIML","PYTHON","NUMPY","PANDAS"]
col=["black","violet","yellow","pink"]

explo=[0,0.3,0,0]
plt.pie(a,labels=labe,explode=explo,startangle=360,colors=col,textprops={'fontsize' : 21})
plt.show()
```



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
d=pd.read_excel('/content/ex.xlsx')
print(d)
mv=d['Temp'].mean()
print("the mean of the temperstures is"mv)
c=np.max(d['Temp'])
a=d.loc[d['Temp'].idxmax(),'Day']
print("the max temp is ",c,"on the day",a)
e=np.min(d['Temp'])
l=d.loc[d['Temp'].idxmin(),'Day']
print("the max temp is ",e,"on the day",l)
threshold=17
v=(d['Temp']>threshold).sum()
print(v)
plt.plot(d['Day'],d['Temp'],color='black',marker='*')
plt.title('Temperature analysis (deg cel)')
plt.xlabel('Day')
plt.ylabel('Temperature')
plt.show()
```

```
    Unnamed: 0  Unnamed: 1  Unnamed: 2  Day  Temp
0          NaN         NaN         NaN    1    12
1          NaN         NaN         NaN    2    14
2          NaN         NaN         NaN    3    15
3          NaN         NaN         NaN    4    17
4          NaN         NaN         NaN    5    18
5          NaN         NaN         NaN    6    20
6          NaN         NaN         NaN    7    22
7          NaN         NaN         NaN    8    23
8          NaN         NaN         NaN    9    21
9          NaN         NaN         NaN   10    19
10         NaN         NaN         NaN   11    18
11         NaN         NaN         NaN   12    16
12         NaN         NaN         NaN   13    14
13         NaN         NaN         NaN   14    13
14         NaN         NaN         NaN   15    11
15         NaN         NaN         NaN   16    10
16         NaN         NaN         NaN   17     9
17         NaN         NaN         NaN   18     8
18         NaN         NaN         NaN   19     7
19         NaN         NaN         NaN   20     9
20         NaN         NaN         NaN   21    10
21         NaN         NaN         NaN   22    12
22         NaN         NaN         NaN   23    14
23         NaN         NaN         NaN   24    15
24         NaN         NaN         NaN   25    17
25         NaN         NaN         NaN   26    18
26         NaN         NaN         NaN   27    20
27         NaN         NaN         NaN   28    22
28         NaN         NaN         NaN   29    23
29         NaN         NaN         NaN   30    21
15.6
the max temp is  23 on the day 8
the max temp is  7 on the day 19
12
```

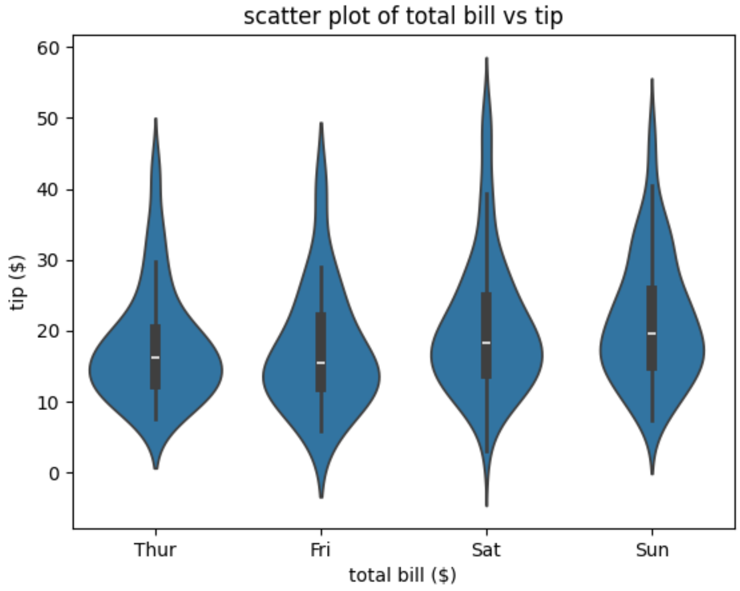


```
import seaborn as sns
import matplotlib.pyplot as plt
TimestampSeries=sns.load_dataset("tips")
print(a.head(10))
sns.scatterplot(x="",y="Day",data=tips)
plt.title("scatter plot of total bill vs tip")
plt.xlabel("total bill ($)")
plt.ylabel("tip ($)")
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-20-1696f60c398b> in <cell line: 3>()
      1 import seaborn as sns
      2 import matplotlib.pyplot as plt
----> 3 ex=sns.load_dataset("/content/ex.xlsx")
      4 sns.scatterplot(x="Temp",y="Day",data=ex)
      5 plt.title("scatter plot of total bill vs tip")

/usr/local/lib/python3.10/dist-packages/seaborn/utils.py in load_dataset(name,
cache, data_home, **kws)
    571         if not os.path.exists(cache_path):
    572             if name not in get_dataset_names():
--> 573                 raise ValueError(f"'{name}' is not one of the example
datasets.")
    574             urlretrieve(url, cache_path)
    575         full_path = cache_path

ValueError: '/content/ex.xlsx' is not one of the example datasets.
```

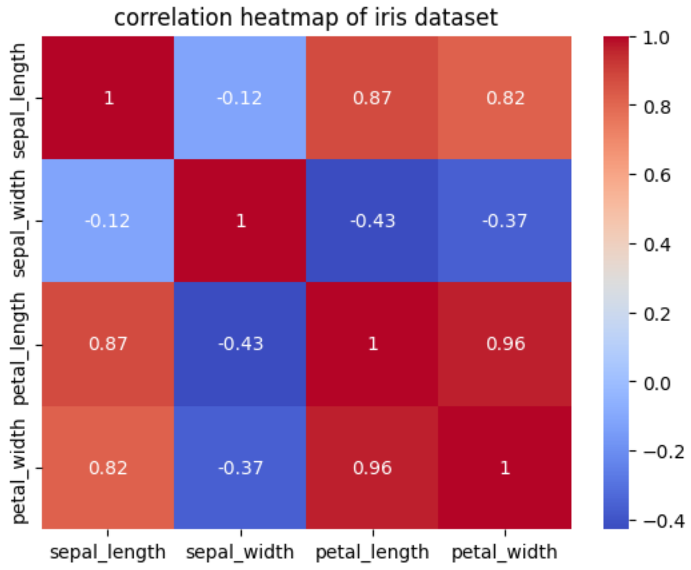
```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("tips")
print(tips.head(10))
sns.violinplot(x="day",y="total_bill",data=tips)
plt.title("scatter plot of total bill vs tip")
plt.xlabel("total bill ($)")
plt.ylabel("tip ($)")
plt.show()
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
5       25.29  4.71    Male     No  Sun  Dinner     4
6        8.77  2.00    Male     No  Sun  Dinner     2
7       26.88  3.12    Male     No  Sun  Dinner     4
8       15.04  1.96    Male     No  Sun  Dinner     2
9       14.78  3.23    Male     No  Sun  Dinner     2
```



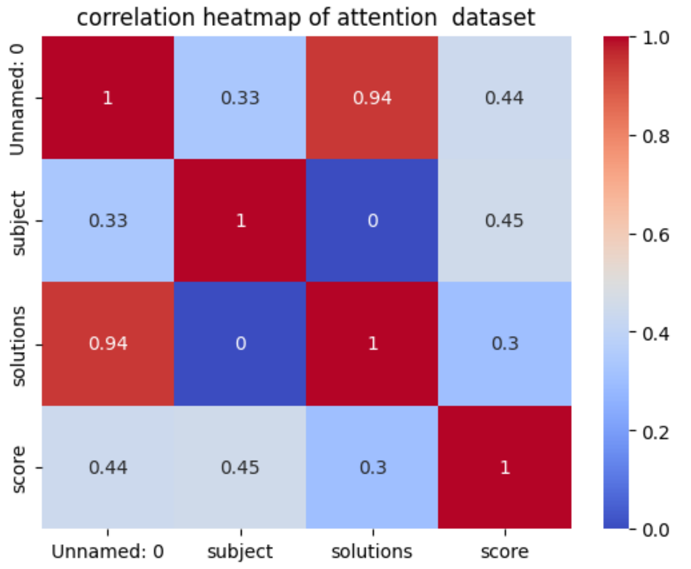
```
iris=sns.load_dataset("iris")
print(iris.head(10))
correlation_matrix=iris.corr()
sns.heatmap(correlation_matrix,annot=True,cmap="coolwarm")
plt.title("correlation heatmap of iris dataset")
plt.show()
```

```
   sepal_length  sepal_width  petal_length  petal_width species
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
5           5.4          3.9           1.7          0.4  setosa
6           4.6          3.4           1.4          0.3  setosa
7           5.0          3.4           1.5          0.2  setosa
8           4.4          2.9           1.4          0.2  setosa
9           4.9          3.1           1.5          0.1  setosa
<ipython-input-34-45a840667c33>:3: FutureWarning: The default value of numeric_only i
  correlation_matrix=iris.corr()
```



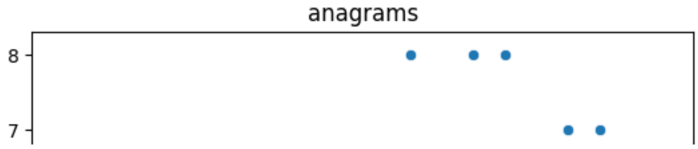
```
attention=sns.load_dataset("attention")
print(attention.head(10))
correlation_matrix=attention.corr()
sns.heatmap(correlation_matrix,annot=True,cmap="coolwarm")
plt.title("correlation heatmap of attention  dataset")
plt.show()
```

```
   Unnamed: 0  subject attention  solutions  score
0           0        1   divided          1    2.0
1           1        2   divided          1    3.0
2           2        3   divided          1    3.0
3           3        4   divided          1    5.0
4           4        5   divided          1    4.0
5           5        6   divided          1    5.0
6           6        7   divided          1    5.0
7           7        8   divided          1    5.0
8           8        9   divided          1    2.0
9           9       10   divided          1    6.0
<ipython-input-37-795b41a4264d>:3: FutureWarning: The default value of numeric_only i
  correlation_matrix=attention.corr()
```



```
import seaborn as sns
import matplotlib.pyplot as plt
a=sns.load_dataset("anagrams")
print(a.head(10))
sns.scatterplot(x="subidr",y="num1",data=a)
plt.title("anagrams")
plt.xlabel("subidr")
plt.ylabel("num1")
plt.show()
```

```
   subidr    attnr  num1  num2  num3
0       1  divided     2   4.0     7
1       2  divided     3   4.0     5
2       3  divided     3   5.0     6
3       4  divided     5   7.0     5
4       5  divided     4   5.0     8
5       6  divided     5   5.0     6
6       7  divided     5   4.5     6
7       8  divided     5   7.0     8
8       9  divided     2   3.0     7
9      10  divided     6   5.0     6
```


anagrams