```
import numpy as np
print("Basic functions")
arr=np.array([1,2,3,4,5])
print(arr)
a_arr=np.zeros((3,3),dtype=int)
print(a_arr)
b_arr=np.ones((2,2),dtype=int)
print(b_arr)
print("range")
x=np.arange(10)
print(x)
print("reshape")
reshape=arr.reshape(5,1)
print(reshape)
print("Slice")
y=arr[2:4]
print(y)
```

```
[1 2 3 4 5]
[[0 0 0]
 [0 0 0]
 [0 0 0]]
[[1 1]
 [1 1]]
range
[0 1 2 3 4 5 6 7 8 9]
reshape
[[1]
 [2]
 [3]
 [4]
 [5]]
Slice
[3 4]
```

```
import pandas as pd
a=['jwalitha','vasanthi','ramya','lahari','varshitha','rupa','teja']
r=pd.Series(a, index=[22,34,54,64,33,65,12])
print(r)
```

```
22     jwalitha
34     vasanthi
54        ramya
64       lahari
33    varshitha
65         rupa
12         teja
dtype: object
```

```
import pandas as pd
df=pd.read_csv("/content/txtfile.txt", sep=" ")
print(df)
```

```
              The  photoelectric        effect,       discovered  \
0             The  photoelectric         effect            marks
1           Light       consists             of         photons,
2       Electrons            can         absorb          photons
3             For           this        process               to
4             The           work       function               is
5       Electrons           that           gain       sufficient
6             The        kinetic         energy               of
7        Einstein       proposed            the        quantized
8             The         energy            (E)               of
9            This       equation       suggests             that
10            The  photoelectric         effect     demonstrates
11      Classical           wave       theories        predicted
12       Instead,            the  photoelectric           effect
13            The  photoelectric         effect           played
14            The     phenomenon         occurs  instantaneously
```

```
15         There            is            a        threshold
16    Increasing           the    intensity               of
17          This      behavior       aligns             with
18       Metals,           due           to            their
19     Non-metals          can         also          display
20           The  photoelectric       effect            paved
21 Understanding           the  photoelectric          effect
22  Experimental  verification           of              the
23           The    phenomenon  highlighted              the
24           The  photoelectric       effect            forms
25           Its     discovery          and      explanation
26           The  photoelectric       effect               is
27  Applications          also       extend               to
28           The  photoelectric       effect              can
29       X-rays,   ultraviolet       light,              and
30           The       kinetic       energy               of
31           The  photoelectric       effect         provides
32       Surface      analysis   techniques             like
33 Understanding           the  photoelectric          effect
34 Photoemission    microscopy      employs              the
35           The  photoelectric       effect              has
36         Solar        panels      utilize              the
37           The    efficiency           of            solar
38       Quantum    efficiency     measures                a
39           The  photoelectric       effect       elucidates
40           Its  implications       extend           beyond
41            In      biology,          the    photoelectric
42       Medical       imaging   techniques             like
43 Understanding           the  photoelectric          effect
44 Environmental    monitoring      devices          utilize
45           The  photoelectric     effect's        quantized
46       Quantum     mechanics     provides                a
47 Wave-particle      duality,            a      fundamental
48          Wave     functions     describe              the

             by       Heinrich       Hertz           in          1887  \
0           the     phenomenon       where    electrons           are
1       packets             of     energy,        which      interact
2           and           gain     energy,  potentially      becoming
3        occur,            the      energy           of           the
4           the        minimum      amount           of        energy
5        energy           from    absorbed      photons           can
```

```
import pandas as pd
df=pd.read_csv("/content/csvfile.csv")
print(df.loc[1])
```

```
Series_reference                        BDCQ.SEA1AA
Period                                      2011.09
Data_value                                  78324.0
Suppressed                                      NaN
STATUS                                            F
UNITS                                        Number
Magnitude                                         0
Subject              Business Data Collection - BDC
Group               Industry by employment variable
Series_title_1                          Filled jobs
Series_title_2     Agriculture, Forestry and Fishing
Series_title_3                               Actual
Series_title_4                                  NaN
Series_title_5                                  NaN
Name: 1, dtype: object
```

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

```
72226.88404123182
```

```
import pandas as pd
df=pd.read_excel("/content/excel.xlsx")
mv=df['Age'].mean()
df=df.fillna(mv)
print(mv)
```

```
24.089
```

```
import pandas as pd
df=pd.read_excel("/content/excel.xlsx",sheet_name=1)
print(df)
```

```
          T. Almada  https://cdn.sofifa.net/players/245/371/24_60.png  22  \
0          L. Palma             https://cdn.sofifa.net/player_0.svg  23
1          R. Lavia  https://cdn.sofifa.net/players/263/620/24_60.png  19
2  W. Zaïre-Emery  https://cdn.sofifa.net/players/270/673/24_60.png  17
3       Gabri Veiga  https://cdn.sofifa.net/players/258/729/24_60.png  21
4    J. Bellingham  https://cdn.sofifa.net/players/270/964/24_60.png  17
5        K. Havertz  https://cdn.sofifa.net/players/235/790/24_60.png  24
6      A. Vermeeren  https://cdn.sofifa.net/players/269/859/24_60.png  18
7        R. Højlund  https://cdn.sofifa.net/players/259/399/24_60.png  20

  Argentina  ['CAM', 'CM', 'CF']  79  87       Atlanta United  2022 ~ 2025  \
0  Honduras               ['LW']  69  75               Celtic  2023 ~ 2028
1   Belgium              ['CDM']  73  86              Chelsea  2023 ~ 2030
2    France        ['CM', 'CDM']  77  89  Paris Saint Germain  2022 ~ 2025
3     Spain        ['CM', 'CAM']  78  89       Al Ahli Jeddah  2023 ~ 2026
4   England        ['CAM', 'CM']  64  82           Sunderland  2023 ~ 2028
5   Germany  ['CAM', 'RW', 'ST']  82  87              Arsenal  2023 ~ 2028
6   Belgium        ['CDM', 'CM']  74  87              Antwerp  2022 ~ 2026
7   Denmark               ['ST']  77  89    Manchester United  2023 ~ 2028

   €39.5M   €10K  2050
0   €2.2M   €22K  1794
1     €7M   €32K  1829
2    €24M    €9K  2080
3  €31.5M   €28K  1944
4   €1.5M    €1K  1714
5    €46M  €110K  2044
6   €9.5M    €7K  1883
7  €25.5M   €77K  1841
```

```
import numpy as np
print("Manipulation functions")
arr1=np.array([1,2,3,4])
arr2=np.array([1,2,3,4])
sum=np.add(arr1,arr2)
print(sum)
print("Broadcasting")
result=arr1+3
print(result)
print("Manipulation using vstack")
z=np.vstack(arr1+arr2)
print(z)
print("Manipulation using stack")
r=np.stack(arr1+arr2)
print(r)
print("Split function")
s=np.split(arr1,2)
print(s)
print("Transpose")
arr5=np.array([[1,2,3],[4,5,6]])
print(arr5)
t1=arr5.T
print(t1)
```

```
Manipulation functions
[2 4 6 8]
Broadcasting
[4 5 6 7]
Manipulation using vstack
[[2]
 [4]
 [6]
 [8]]
Manipulation using stack
[2 4 6 8]
Split function
[array([1, 2]), array([3, 4])]
Transpose
[[1 2 3]
 [4 5 6]]
[[1 4]
 [2 5]
 [3 6]]
```

```
import numpy as np
print("Sum of elements of a matrix")
a=np.array([[2,3,4],[4,8,5]])
x=np.sum(a)
print(x)
print("Sum of elements 2 matrices")
b=np.array([[2,9,8],[3,8,5]])
y=np.sum(a+b)
print(y)
d=np.sum(a, axis=0)
print(d)
e=np.sum(a, axis=1)
print(e)
```

```
Sum of elements of a matrix
26
Sum of elements 2 matrices
61
[ 6 11  9]
[ 9 17]
```

```
print("Matrix multiplication")
a=np.array([[1,2],[4,5]])
b=np.array([[7,8],[6,3]])
c=np.dot(a,b)
print(c)
x=np.linalg.eig(c)
print(x)
```

```
Matrix multiplication
[[19 14]
 [58 47]]
(array([ 1.25098427, 64.74901573]), array([[-0.61930684, -0.29262253],
       [ 0.78514906, -0.95622804]]))
```

```
import numpy as np
print("statistical operations")
print("Standard deviation")
a=np.array([1,2,3,4,5])
m=np.mean(a)
print(m)
med=np.median(a)
print(med)
vari=np.var(a)
print(vari)
std=np.std(a)
print(std)
```

```
statistical operations
Standard deviation
3.0
3.0
2.0
1.4142135623730951
```

```
data = np.loadtxt("/content/app.txt")
data=np.savetxt("/content/app2.txt",data)
print(data)
```

```
None
```

```
import pandas as pd
df=pd.read_excel("/content/excel.xlsx")
df=df.drop_duplicates()
print(df)
df.head(2)
df.shape
```

```
        Player_name                                            Images  Age  \
0         T. Almada  https://cdn.sofifa.net/players/245/371/24_60.png   22
1          L. Palma              https://cdn.sofifa.net/player_0.svg   23
2          R. Lavia  https://cdn.sofifa.net/players/263/620/24_60.png   19
3    W. Zaïre-Emery  https://cdn.sofifa.net/players/270/673/24_60.png   17
4       Gabri Veiga  https://cdn.sofifa.net/players/258/729/24_60.png   21
..              ...                                               ...  ...
93  Stefan Bajcetic  https://cdn.sofifa.net/players/271/975/24_60.png   18
94     A. Griezmann  https://cdn.sofifa.net/players/194/765/24_60.png   32
95      I. Bennacer  https://cdn.sofifa.net/players/226/754/24_60.png   25
96     Aleix García  https://cdn.sofifa.net/players/228/813/24_60.png   26
97         A. Hložek  https://cdn.sofifa.net/players/246/618/24_60.png   20

     National_team            Positions  Overall  Potential_overall  \
0        Argentina   ['CAM', 'CM', 'CF']       79                 87
1         Honduras               ['LW']       69                 75
2          Belgium              ['CDM']       73                 86
3           France        ['CM', 'CDM']       77                 89
4            Spain        ['CM', 'CAM']       78                 89
..             ...                  ...      ...                ...
93           Spain        ['CDM', 'CM']       72                 86
94          France         ['ST', 'CF']       88                 88
95         Algeria  ['CDM', 'CM', 'CAM']       84                 89
96           Spain        ['CM', 'CDM']       81                 84
97  Czech Republic        ['ST', 'CAM']       77                 85

           Current_club Current_contract   Value  Wage  Total_stats
0        Atlanta United      2022 ~ 2025  €39.5M  €10K         2050
1                Celtic      2023 ~ 2028   €2.2M  €22K         1794
2               Chelsea      2023 ~ 2030     €7M  €32K         1829
3   Paris Saint Germain      2022 ~ 2025    €24M   €9K         2080
4        Al Ahli Jeddah      2023 ~ 2026  €31.5M  €28K         1944
..                  ...              ...     ...   ...          ...
93            Liverpool      2021 ~ 2027     €5M  €15K         1821
94      Atlético Madrid      2022 ~ 2026    €74M €135K         2322
95                Milan      2019 ~ 2027  €56.5M  €84K         2266
96               Girona      2021 ~ 2026  €33.5M  €36K         2122
97  Bayer 04 Leverkusen      2022 ~ 2027  €23.5M  €39K         1950

[96 rows x 12 columns]
(96, 12)
```

```
import pandas as pd
df=pd.read_csv("/content/csvfile.csv")
dfn=df.tail(10)
for i in range(22188,22178,-1):
  df.drop([i],axis=0,inplace=True)
dfn.to_csv("new_file")
dfs=pd.read_csv("/content/new_file")
print(dfs.groupby(['Subject'])['UNITS'].count())
```

```
Subject
Business Data Collection - BDC    10
Name: UNITS, dtype: int64
```

```
dfs=pd.read_csv("/content/new_file")
print(dfs)
```

```
   Unnamed: 0 Series_reference   Period  Data_value Suppressed STATUS   UNITS  \
0       22179    BDCQ.SEE3999A  2016.03         NaN          Y      C  Number
1       22180    BDCQ.SEE3999A  2016.06         NaN          Y      C  Number
2       22181    BDCQ.SEE3999A  2016.09         NaN          Y      C  Number
3       22182    BDCQ.SEE3999A  2016.12         NaN          Y      C  Number
4       22183    BDCQ.SEE3999A  2017.03         NaN          Y      C  Number
5       22184    BDCQ.SEE3999A  2017.06         NaN          Y      C  Number
6       22185    BDCQ.SEE3999A  2017.09         NaN          Y      C  Number
7       22186    BDCQ.SEE3999A  2017.12         NaN          Y      C  Number
8       22187    BDCQ.SEE3999A  2018.03         NaN          Y      C  Number
9       22188    BDCQ.SEE3999A  2018.06         NaN          Y      C  Number

   Magnitude                         Subject  \
0          0  Business Data Collection - BDC
1          0  Business Data Collection - BDC
2          0  Business Data Collection - BDC
3          0  Business Data Collection - BDC
4          0  Business Data Collection - BDC
5          0  Business Data Collection - BDC
6          0  Business Data Collection - BDC
7          0  Business Data Collection - BDC
8          0  Business Data Collection - BDC
9          0  Business Data Collection - BDC

                                      Group  \
0  Territorial authority by employment variable
1  Territorial authority by employment variable
2  Territorial authority by employment variable
3  Territorial authority by employment variable
4  Territorial authority by employment variable
5  Territorial authority by employment variable
6  Territorial authority by employment variable
7  Territorial authority by employment variable
8  Territorial authority by employment variable
9  Territorial authority by employment variable

                         Series_title_1                      Series_title_2  \
0  Filled jobs (workplace location based)  Area Outside Territorial Authority
1  Filled jobs (workplace location based)  Area Outside Territorial Authority
2  Filled jobs (workplace location based)  Area Outside Territorial Authority
3  Filled jobs (workplace location based)  Area Outside Territorial Authority
4  Filled jobs (workplace location based)  Area Outside Territorial Authority
5  Filled jobs (workplace location based)  Area Outside Territorial Authority
6  Filled jobs (workplace location based)  Area Outside Territorial Authority
7  Filled jobs (workplace location based)  Area Outside Territorial Authority
8  Filled jobs (workplace location based)  Area Outside Territorial Authority
9  Filled jobs (workplace location based)  Area Outside Territorial Authority

  Series_title_3  Series_title_4  Series_title_5
0         Actual             NaN             NaN
1         Actual             NaN             NaN
2         Actual             NaN             NaN
3         Actual             NaN             NaN
4         Actual             NaN             NaN
```

```
5        Actual            NaN            NaN
6        Actual            NaN            NaN
7        Actual            NaN            NaN
8        Actual            NaN            NaN
```

```
import pandas as pd
df=pd.read_csv("/content/csvfile.csv")
print(df)
```

```
22185          0  Business Data Collection - BDC
22186          0  Business Data Collection - BDC
22187          0  Business Data Collection - BDC
22188          0  Business Data Collection - BDC

                                         Group  \
0                 Industry by employment variable
1                 Industry by employment variable
2                 Industry by employment variable
3                 Industry by employment variable
4                 Industry by employment variable
...                                          ...
22184  Territorial authority by employment variable
22185  Territorial authority by employment variable
22186  Territorial authority by employment variable
22187  Territorial authority by employment variable
22188  Territorial authority by employment variable

                              Series_title_1  \
0                                Filled jobs
1                                Filled jobs
2                                Filled jobs
3                                Filled jobs
4                                Filled jobs
...                                      ...
22184  Filled jobs (workplace location based)
22185  Filled jobs (workplace location based)
22186  Filled jobs (workplace location based)
22187  Filled jobs (workplace location based)
22188  Filled jobs (workplace location based)

                          Series_title_2 Series_title_3  Series_title_4  \
0       Agriculture, Forestry and Fishing         Actual             NaN
1       Agriculture, Forestry and Fishing         Actual             NaN
2       Agriculture, Forestry and Fishing         Actual             NaN
3       Agriculture, Forestry and Fishing         Actual             NaN
4       Agriculture, Forestry and Fishing         Actual             NaN
...                                   ...            ...             ...
22184  Area Outside Territorial Authority         Actual             NaN
22185  Area Outside Territorial Authority         Actual             NaN
22186  Area Outside Territorial Authority         Actual             NaN
22187  Area Outside Territorial Authority         Actual             NaN
22188  Area Outside Territorial Authority         Actual             NaN

       Series_title_5
0                 NaN
1                 NaN
2                 NaN
3                 NaN
4                 NaN
...               ...
22184             NaN
22185             NaN
22186             NaN
22187             NaN
22188             NaN

[22189 rows x 14 columns]
```

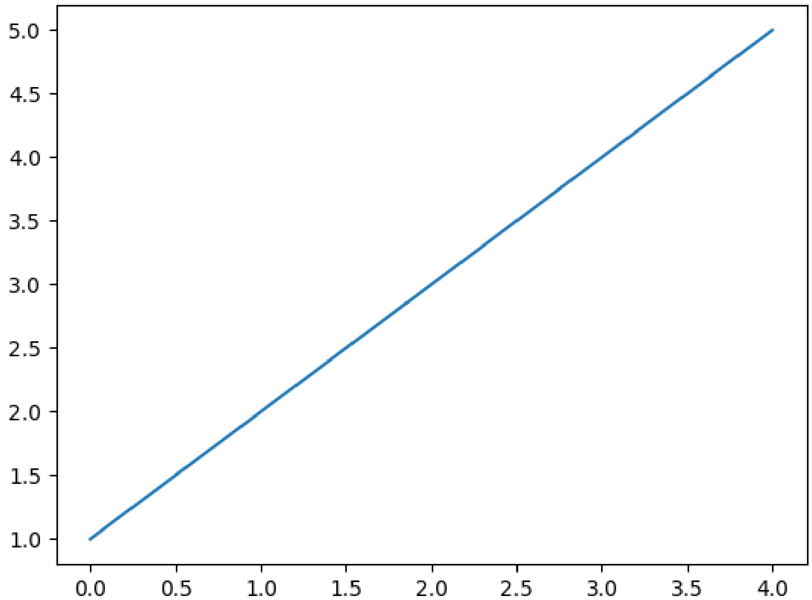
```
import pandas as pd
df=pd.read_csv("/content/csvfile.csv")
df.shape
```

```
(22189, 14)
```

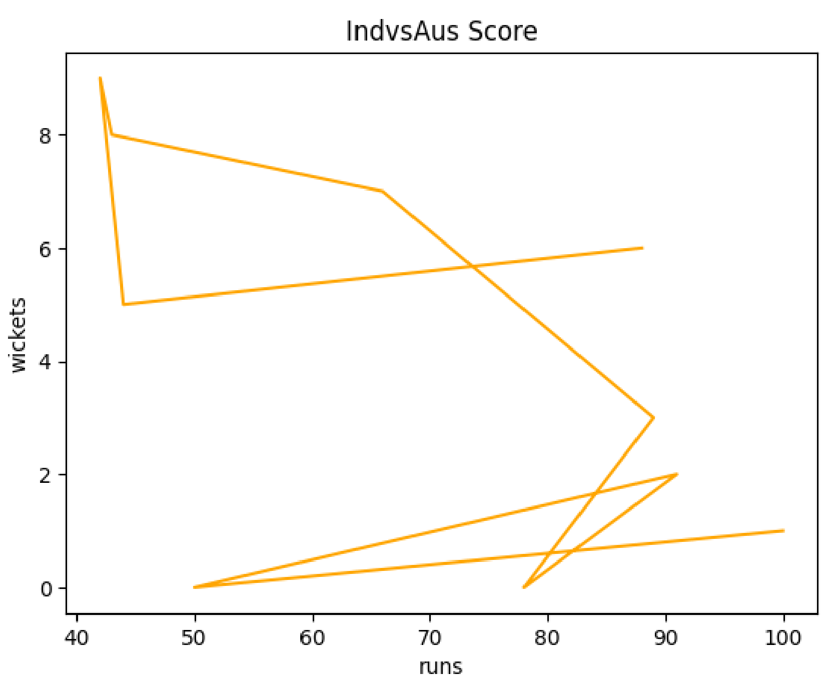
```
from google.colab import drive
drive.mount('/content/drive')
```

```
import matplotlib.pyplot as plt
a=np.array([1,2,3,4,5])
plt.plot(a)
```

```
[<matplotlib.lines.Line2D at 0x79fe9ab61030>]
```

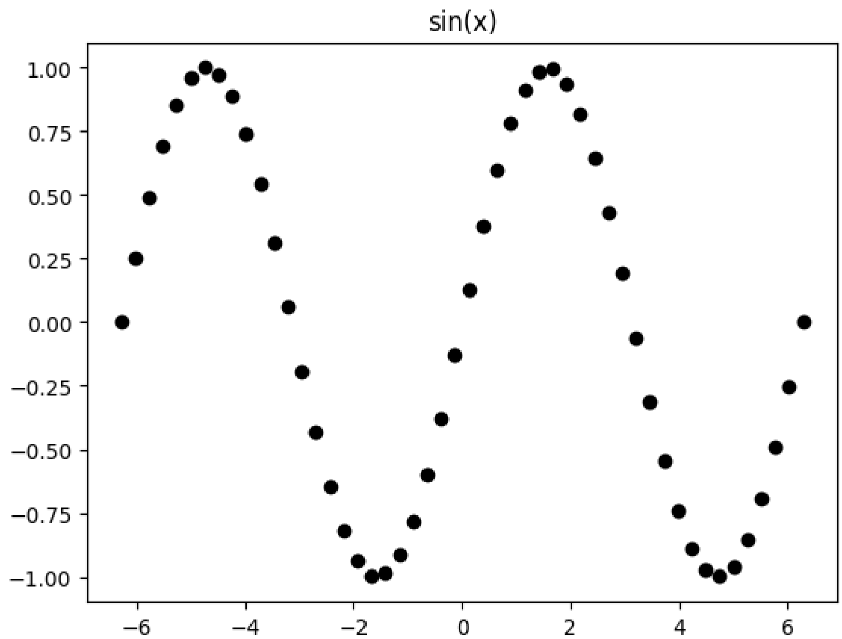



```
import numpy as np
import matplotlib.pyplot as plt
runs=np.array([100,50,91,78,89,66,43,42,44,88])
w=np.array([1,0,2,0,3,7,8,9,5,6])
plt.xlabel("runs")
plt.ylabel("wickets")
plt.plot(runs,w,color='orange')
plt.title('IndvsAus Score')
plt.show()
```



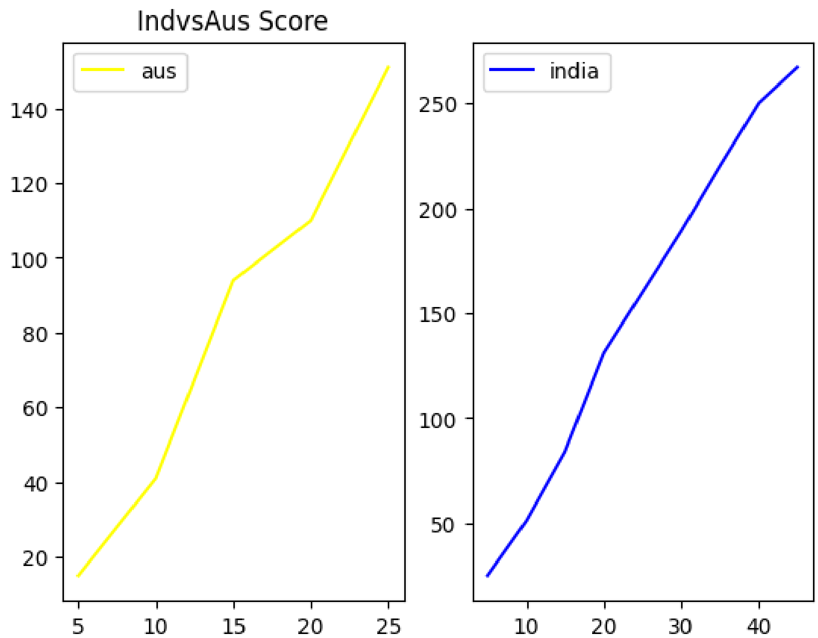
```
import numpy as np
import matplotlib.pyplot as plt
tigar= np.linspace(-2*np.pi,2*np.pi, 50)
print(tigar)
plt.scatter(tigar,np.sin(tigar),color='black')
plt.title("sin(x)")
plt.show()
```

```
[-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]
```



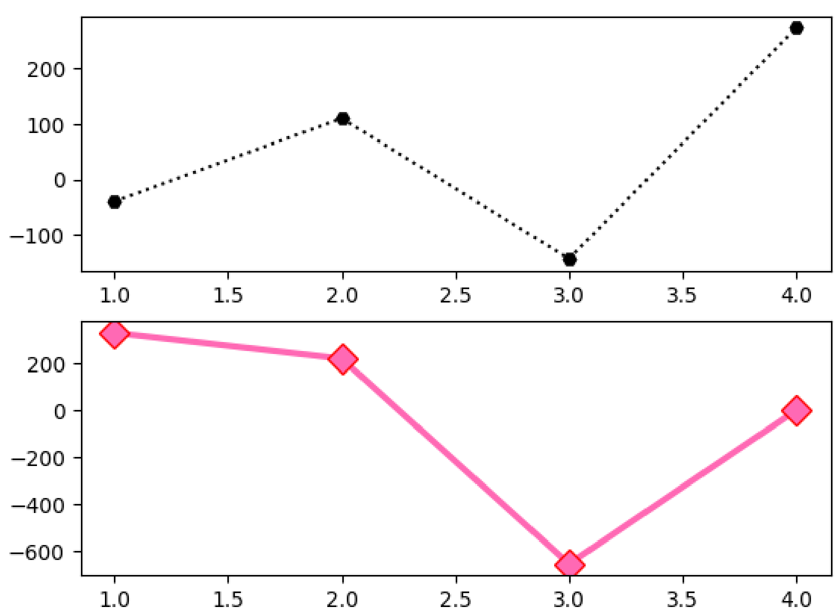

```
import numpy as np
import matplotlib.pyplot as plt
overs1=np.arange(5,50,5)
overs2=np.arange(5,30,5)
runs1=np.array([25,51,84,131,160,189,220,250,267])
runs2=np.array([15,41,94,110,151])
w=np.array([12,32,96])
plt.xlabel("runs")
plt.ylabel("wickets")
plt.subplot(1,2,2)
plt.plot(overs1,runs1,color='blue',label='india')
plt.legend(loc='best')
plt.subplot(1,2,1)
plt.plot(overs2,runs2,color='yellow',label='aus')
plt.legend(loc='best')
plt.title('IndvsAus Score')
plt.show()
```

```
<ipython-input-25-8534de5dbe9d>:10: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated since
  plt.subplot(1,2,2)
```

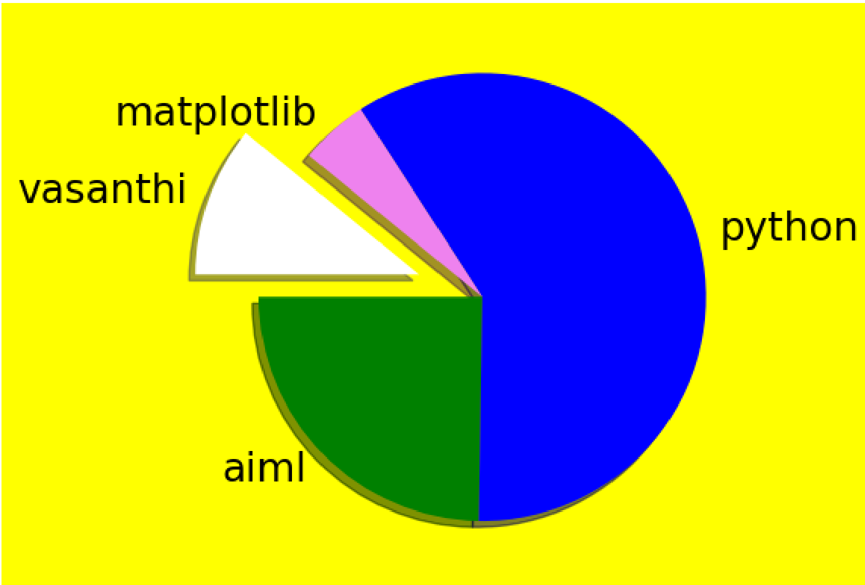



```
import matplotlib.pyplot as plt
a=[230,560,780,127,128]
b=[200,160,270,127,400]
years=[1,2,3,4]
profit_a=[(a[i]-a[i-1]) for i in range(1,len(a))]
profit_b=[(b[i]-b[i-1]) for i in range(1,len(b))]
plt.subplot(2,1,2)
plt.plot(years,profit_a,color='hotpink',linewidth='3',label='CompanyA',marker='D',ms='10',mec='r')
plt.subplot(2,1,1)
plt.plot(years,profit_b,color='black',linestyle='dotted',label='CompanyB',marker='H')
```

```
[<matplotlib.lines.Line2D at 0x7dbe4b007910>]
```

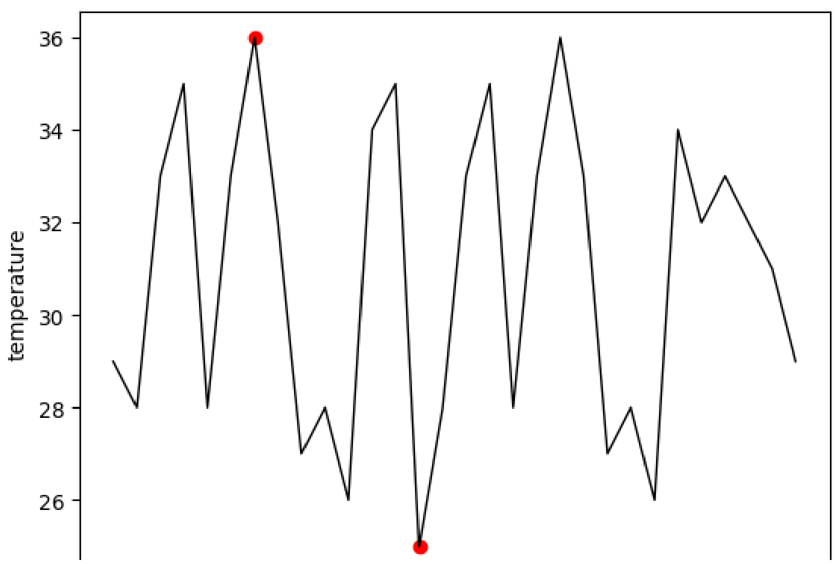


```
a=np.array([25,60,5,11])
labe=["aiml","python","matplotlib","vasanthi"]
col=['green','blue','violet','white']
explo=[0,0,0,0.3]
plt.figure(facecolor='yellow')
plt.pie(a,labels=labe,colors=col,explode=explo,textprops={'fontsize': 19},startangle=180,shadow=True)
plt.show()
```




```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_excel("/content/Book 5.xlsx")
print(df)
print("Average of temperature")
mv=df['temperature'].mean()
print(mv)
print("Highest temperature")
x=df['temperature'].max()
y= df.loc[df['temperature'].idxmax(), 'date']
print(x ,"on", y)
print("Lowest temperature")
a=df['temperature'].min()
b=df.loc[df['temperature'].idxmin(), 'date']
print(a ,"on" ,b)
threshold=32
print("Temperature above a given threshold of",threshold)
ab=(df['temperature'] > threshold).sum()
print(ab)
plt.plot(df['date'],df['temperature'],color='black',linewidth='1')
plt.xlabel("date")
plt.ylabel("temperature")
t=np.array([x,a])
n=np.array([y,b])
plt.scatter(n,t,color='red')
plt.show()
```

```
    date  temperature
0      1           29
1      2           28
2      3           33
3      4           35
4      5           28
5      6           33
6      7           36
7      8           32
8      9           27
9     10           28
10    11           26
11    12           34
12    13           35
13    14           25
14    15           28
15    16           33
16    17           35
17    18           28
18    19           33
19    20           36
20    21           33
21    22           27
22    23           28
23    24           26
24    25           34
25    26           32
26    27           33
27    28           32
28    29           31
29    30           29
Average of temperature
30.9
Highest temperature
36 on 7
Lowest temperature
25 on 14
Temperature above a given threshold of 32
13
```



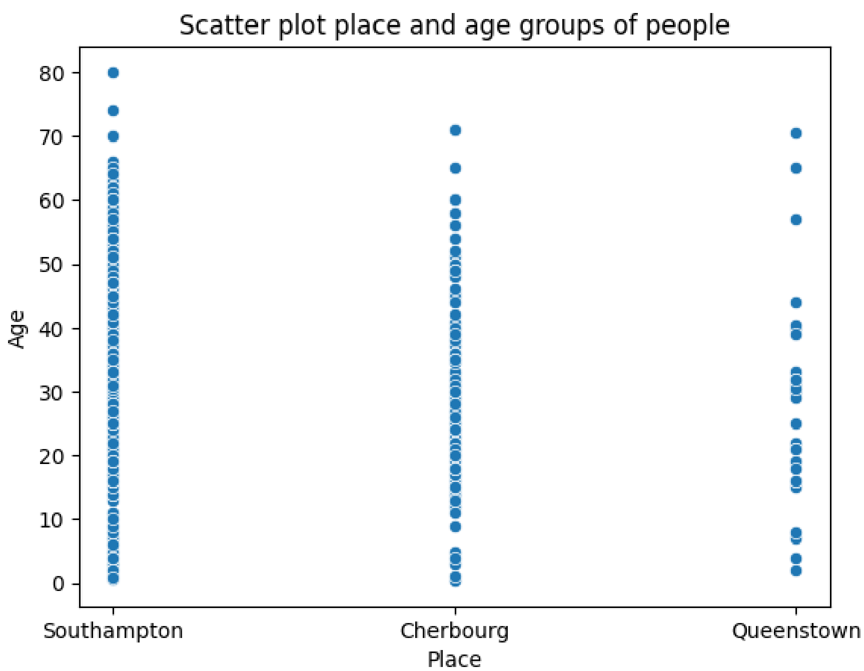

```
pip install seaborn
```

```
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.13.1)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.25.2)
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.5.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /usr/local/lib/python3.10/dist-packages (from seaborn) (3.7.
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->
```

```
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn) (20
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplot
```

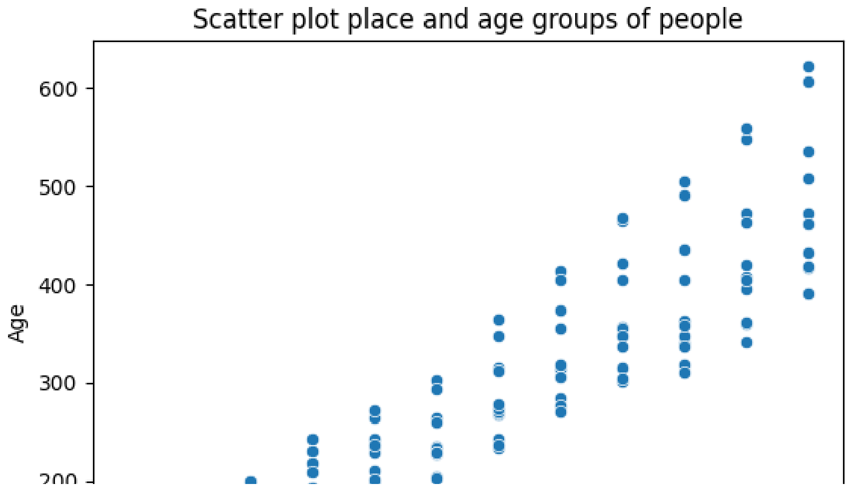
```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("titanic")
print(tips)
sns.scatterplot(x="embark_town",y='age',data=tips)
plt.title("Scatter plot place and age groups of people")
plt.xlabel("Place")
plt.ylabel("Age")
plt.show()
```

```
     survived  pclass     sex   age  sibsp  parch     fare embarked   class  \
0           0       3    male  22.0      1      0   7.2500        S   Third
1           1       1  female  38.0      1      0  71.2833        C   First
2           1       3  female  26.0      0      0   7.9250        S   Third
3           1       1  female  35.0      1      0  53.1000        S   First
4           0       3    male  35.0      0      0   8.0500        S   Third
..        ...     ...     ...   ...    ...    ...      ...      ...     ...
886         0       2    male  27.0      0      0  13.0000        S  Second
887         1       1  female  19.0      0      0  30.0000        S   First
888         0       3  female   NaN      1      2  23.4500        S   Third
889         1       1    male  26.0      0      0  30.0000        C   First
890         0       3    male  32.0      0      0   7.7500        Q   Third

       who  adult_male deck  embark_town alive  alone
0      man        True  NaN  Southampton    no  False
1    woman       False    C    Cherbourg   yes  False
2    woman       False  NaN  Southampton   yes   True
3    woman       False    C  Southampton   yes  False
4      man        True  NaN  Southampton    no   True
..     ...         ...  ...          ...   ...    ...
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

[891 rows x 15 columns]
```




```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("flights")
print(tips)
sns.scatterplot(x="year",y='passengers',data=tips)
plt.title("Scatter plot place and age groups of people")
plt.xlabel("Place")
plt.ylabel("Age")
plt.show()
```

```
     year month  passengers
0    1949   Jan         112
1    1949   Feb         118
2    1949   Mar         132
3    1949   Apr         129
4    1949   May         121
..    ...   ...         ...
139  1960   Aug         606
140  1960   Sep         508
141  1960   Oct         461
142  1960   Nov         390
143  1960   Dec         432

[144 rows x 3 columns]
```




```
import seaborn as sns
import matplotlib.pyplot as plt
tips=sns.load_dataset("tips")
print(tips)
sns.violinplot(x="day",y='total_bill',data=tips)
plt.title("Scatter plot place and age groups of people")
plt.xlabel("Place")
plt.ylabel("Age")
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
..          ...   ...     ...    ...  ...     ...   ...
239       29.03  5.92    Male     No  Sat  Dinner     3
240       27.18  2.00  Female    Yes  Sat  Dinner     2
241       22.67  2.00    Male    Yes  Sat  Dinner     2
242       17.82  1.75    Male     No  Sat  Dinner     2
243       18.78  3.00  Female     No Thur  Dinner     2

[244 rows x 7 columns]
```