

```
import numpy as np
matrix1=np.array([[1, 2, 3],[4, 5, 6],[7, 8, 9]])
matrix2=np.array([[9, 8, 7],[6, 5, 4],[3, 2, 1]])
result_matrix=matrix1+matrix2
```

```
!pip install numpy
```

```
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (1.23.5)
```

```
import numpy as np
arr=np.array([1,2,3,4,5])
print(arr)
```

```
a=np.zeros((3,3))
print(a)
```

```
b=np.ones((2,2))
print(b)
a=np.zeros((3,3),int)
print(a)
```

```
arange_arr=np.arange(10)
print(arange_arr)
```

```
#array manipulation
arr=np.array([1,2,3,4,5])
reshape_arr=arr.reshape(5,1)
print(reshape_arr)
```

```
a=arr[2:4]
print(a)
```

```
#mathematical application
```

```
import numpy as np
arr1=np.array([1,2,3,4])
arr2=np.array([3,4,5,6])
arr =arr1+arr2
print(arr)
print(arr2+3)
```

```
[1 2 3 4 5]
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

```
[[1. 1.]
 [1. 1.]]
[[0 0 0]
 [0 0 0]
 [0 0 0]]
[0 1 2 3 4 5 6 7 8 9]
[[1]
 [2]
 [3]
 [4]
 [5]]
[3 4]
[ 4 6 8 10]
[6 7 8 9]
```

```
import numpy as np
arr1=np.array([1,2,3,4])
arr2=np.array([6,7,8,9])
#vstack
c=np.vstack(arr1+arr2)
print(c)
#stack
c=np.stack(arr1+arr2)
print(c)
#split
a=np.split(arr1,2)
print(a)
```

```
[[ 7]
 [ 9]
 [11]
 [13]]
[ 7  9 11 13]
[array([1, 2]), array([3, 4])]
```

```
import numpy as np
#transpose
arr1=np.array([[1,2,3,4],[5,6,7,8]])
b=arr1.T
print(b)
```

```
[[1 5]
 [2 6]
 [3 7]
 [4 8]]
```

```
import numpy as np
#linear algebra with numpy
a=np.array([[1,2],[6,7]])
```

```

b=np.array([[4,3],[5,8]])
c=np.dot(a,b)
print(c)
#calculating eigen value and eigen vectors
d=np.linalg.eig(c)
print(d)

#create 2+3 matrix
a=np.array([[2,3,4],[4,5,6]])
b=np.sum(a)
print(b)
d=np.array([[1,2,3],[4,8,7]])
c=np.array([[8,7,6],[4,5,6]])
e=np.sum(d+c)
print(e)

[[14 19]
 [59 74]]
(array([-0.95553359, 88.95553359]), array([[-0.78577633, -0.24571251],
      [ 0.61851075, -0.96934275]]))
24
61

import numpy as np
a=np.array([[2,3,4],[5,6,7]])
c=np.sum(a, axis=0)
d=np.sum(a, axis=1)
print(c)
print(d)

[ 7  9 11]
[ 9 18]

import numpy as np
a=np.array([1,2,3,4,5])
b=np.mean(a)
c=np.median(a)
d=np.var(a)
e=np.std(a)
print(b)
print(c)
print(d)
print(e)

3.0
3.0
2.0
1.4142135623730951

data=np.loadtxt("/content/eee.txt",dtype=int)
print(data)

```

```
[[1 2 3 8]
 [4 5 6 7]
 [3 4 5 6]]
```

*#loading the data in.txt file and saving the data*

```
data=np.loadtxt("/content/data.txt",dtype=int)
data=np.savetxt("/content/dataa.txt",data)
print(data)
```

None

```
data=np.loadtxt("/content/data.txt",dtype=int)
print(data)
```

```
[[1 2 3 8]
 [4 5 6 7]
 [3 4 5 6]]
```

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

*#importing pandas*

```
import pandas as pd
a=["Iswarya","satya","Sravani","Bala","Keerthi","Harshitha"]
r=pd.Series(a,index=[62,11,45,33,54,53])
print(r)
```

```
62    Iswarya
11      satya
45    Sravani
33      Bala
54    Keerthi
53  Harshitha
dtype: object
```

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

	Player_Id	Player_Name	DOB	Batting_Hand	Bowling_Skill
0	1	SC Ganguly	8-Jul-72	Left_Hand	Right-arm medium
1	2	BB McCullum	27-Sep-81	Right_Hand	Right-arm medium
2	3	RT Ponting	19-Dec-74	Right_Hand	Right-arm medium
3	4	DJ Hussey	15-Jul-77	Right_Hand	Right-arm offbreak

```

4      5  Mohammad Hafeez  17-Oct-80  Right_Hand  Right-arm
offbreak
..      ...      ...      ...      ...
...
518      519      Subroto Das      NaN      NaN
NaN
519      520      K Srinivasan      NaN      NaN
NaN
520      521      VK Sharma      NaN      NaN
NaN
521      523      AV Wankhade  14-Mar-92  Right_Hand
NaN
522      524      B Aparajith  8-Jul-94  Right_Hand  Right-arm
offbreak

```

```

      Country  Is_Umpire  Unnamed: 7
0      India      0      NaN
1  New Zealand      0      NaN
2  Australia      0      NaN
3  Australia      0      NaN
4  Pakistan      0      NaN
..      ...      ...      ...
518  India      1      NaN
519  India      1      NaN
520  India      1      NaN
521  India      0      NaN
522  India      0      NaN

```

```
[523 rows x 8 columns]
```

```

import pandas as pd
df=pd.read_csv("/content/562.txt",sep=" ")
print(df)

```

```

      1
0      2
1      3
2      4
3      5
4      6
5      7
6      8
7      9
8     10

```

```

import pandas as pd
df=pd.read_excel("/content/Survey May-June 2020.xlsx")
print(df)

```

	ISO3	COUNTRYNAME	SDG_REGION	Q1_PP_NW \	
0	SDN	Sudan	Africa (Sub-Saharan)	Nation-wide	
1	DZA	Algerie	Africa (Northern)	Nation-wide	
2	NER	Niger	Africa (Sub-Saharan)	Nation-wide	
3	ZAF	South Africa	Africa (Sub-Saharan)	NaN	
4	TCD	Chad	Africa (Sub-Saharan)	NaN	
...	...	...	...	...	
117	TUN	Tunisia	Africa (Northern)	Nation-wide	
118	MLI	Mali	Africa (Sub-Saharan)	NaN	
119	PAK	Pakistan	Asia (Central and Southern)	NaN	
120	COM	Comoros	Africa (Sub-Saharan)	Nation-wide	
121	COK	Cook Islands	Oceania	Nation-wide	
		Q1_PP_PG	Q1_PP_PS	Q1_PP_DNK	
Q1_PP_NC \					
0		NaN	NaN	NaN	
NaN					
1		NaN	NaN	NaN	
NaN					
2		NaN	NaN	NaN	
NaN					
3		NaN	Phasing students	NaN	
NaN					
4		NaN	NaN	NaN	
NaN					
...		...	...	...	
...					
117		NaN	Phasing students	NaN	
NaN					
118		NaN	NaN	Do not know	
NaN					
119	Partial/Gradual		NaN	NaN	
NaN					
120	Partial/Gradual		NaN	NaN	
NaN					
121		NaN	NaN	NaN	
				Schools are not closed	
	Q1_PP_E0	Q1_P_NW	... Q1_P_NR	Q1_LS_NR	Q1_US_NR
Q12_PP_NR \					
0	06/09/2020	Nation-wide	...	NaN	NaN
NaN					
1	Not recorded	Nation-wide	...	NaN	NaN
NaN					
2	01/01/2021	NaN	...	NaN	NaN
NaN					
3	Not recorded	NaN	...	NaN	NaN
NaN					
4		NaN	NaN	NaN	NaN
					Not recorded

```

...
...
117 15/09/2020 Nation-wide ... NaN NaN NaN Not
recorded
118 11/05/2020 NaN ... NaN NaN NaN
NaN
119 01/06/2020 NaN ... NaN NaN NaN
NaN
120 16/05/2020 Nation-wide ... NaN NaN NaN
NaN
121 Not recorded Nation-wide ... NaN NaN NaN
NaN

```

```

      Q12_P_NR  Q12_LS_NR  Q12_US_NR  Q17_NR  Q18_NR  \
0          NaN          NaN          NaN          NaN          NaN
1  Not recorded  Not recorded  Not recorded          NaN          NaN
2          NaN          NaN          NaN  Not recorded          NaN
3          NaN          NaN          NaN          NaN          NaN
4          NaN          NaN          NaN          NaN          NaN
...
117          NaN          NaN          NaN          NaN          NaN
118          NaN          NaN          NaN          NaN          NaN
119          NaN          NaN          NaN          NaN          NaN
120          NaN          NaN          NaN          NaN          NaN
121          NaN          NaN          NaN          NaN          NaN

```

```

      Q19_NR
0  Not recorded
1  Not recorded
2          NaN
3          NaN
4          NaN
...
117          NaN
118          NaN
119          NaN
120          NaN
121          NaN

```

[122 rows x 250 columns]

```

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

```

```

      Player_Id  Player_Name  DOB  Batting_Hand
Bowling_Skill  \
0              1    SC Ganguly  8-Jul-72    Left_Hand    Right-arm
medium
1              2    BB McCullum  27-Sep-81    Right_Hand    Right-arm

```

medium					
2	3	RT Ponting	19-Dec-74	Right_Hand	Right-arm
medium					
3	4	DJ Hussey	15-Jul-77	Right_Hand	Right-arm
offbreak					
4	5	Mohammad Hafeez	17-Oct-80	Right_Hand	Right-arm
offbreak					
..	...	...	...	...	
...					
518	519	Subroto Das	NaN	NaN	
NaN					
519	520	K Srinivasan	NaN	NaN	
NaN					
520	521	VK Sharma	NaN	NaN	
NaN					
521	523	AV Wankhade	14-Mar-92	Right_Hand	
NaN					
522	524	B Aparajith	8-Jul-94	Right_Hand	Right-arm
offbreak					

	Country	Is_Umpire	Unnamed: 7
0	India	0	NaN
1	New Zealand	0	NaN
2	Australia	0	NaN
3	Australia	0	NaN
4	Pakistan	0	NaN
..	...	...	...
518	India	1	NaN
519	India	1	NaN
520	India	1	NaN
521	India	0	NaN
522	India	0	NaN

[523 rows x 8 columns]

```
import pandas as pd
df=pd.read_excel("/content/Survey May-June 2020.xlsx",sheet_name=0)
print(df)
```

	ISO3	COUNTRYNAME	SDG_REGION	Q1_PP_NW \
0	SDN	Sudan	Africa (Sub-Saharan)	Nation-wide
1	DZA	Algerie	Africa (Northern)	Nation-wide
2	NER	Niger	Africa (Sub-Saharan)	Nation-wide
3	ZAF	South Africa	Africa (Sub-Saharan)	NaN
4	TCD	Chad	Africa (Sub-Saharan)	NaN
..	...	...	...	...
117	TUN	Tunisia	Africa (Northern)	Nation-wide
118	MLI	Mali	Africa (Sub-Saharan)	NaN
119	PAK	Pakistan	Asia (Central and Southern)	NaN
120	COM	Comoros	Africa (Sub-Saharan)	Nation-wide



121	COK	Cook Islands			Oceania	Nation-wide
		Q1_PP_PG	Q1_PP_PS	Q1_PP_DNK		
Q1_PP_NC	\					
0		NaN	NaN	NaN		
NaN						
1		NaN	NaN	NaN		
NaN						
2		NaN	NaN	NaN		
NaN						
3		NaN	Phasing students	NaN		
NaN						
4		NaN	NaN	NaN		
NaN						
..		...	...	...		
...						
117		NaN	Phasing students	NaN		
NaN						
118		NaN	NaN	Do not know		
NaN						
119	Partial/Gradual		NaN	NaN		
NaN						
120	Partial/Gradual		NaN	NaN		
NaN						
121		NaN	NaN	NaN	Schools are not closed	
		Q1_PP_EO	Q1_P_NW	... Q1_P_NR	Q1_LS_NR	Q1_US_NR
Q12_PP_NR	\					
0	06/09/2020	Nation-wide	...	NaN	NaN	NaN
NaN						
1	Not recorded	Nation-wide	...	NaN	NaN	NaN
NaN						
2	01/01/2021	NaN	...	NaN	NaN	NaN
NaN						
3	Not recorded	NaN	...	NaN	NaN	NaN
NaN						
4	NaN	NaN	...	NaN	NaN	NaN
NaN						
..		...	...	...	...	...
...						
117	15/09/2020	Nation-wide	...	NaN	NaN	NaN
NaN						
118	11/05/2020	NaN	...	NaN	NaN	NaN
NaN						
119	01/06/2020	NaN	...	NaN	NaN	NaN
NaN						
120	16/05/2020	Nation-wide	...	NaN	NaN	NaN
NaN						

121	Not recorded	Nation-wide	...	NaN	NaN	NaN
NaN						

	Q12_P_NR	Q12_LS_NR	Q12_US_NR	Q17_NR	Q18_NR	\
0	NaN	NaN	NaN	NaN	NaN	
1	Not recorded	Not recorded	Not recorded	NaN	NaN	
2	NaN	NaN	NaN	Not recorded	NaN	
3	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	NaN	
..	...	...	...	...	...	
117	NaN	NaN	NaN	NaN	NaN	
118	NaN	NaN	NaN	NaN	NaN	
119	NaN	NaN	NaN	NaN	NaN	
120	NaN	NaN	NaN	NaN	NaN	
121	NaN	NaN	NaN	NaN	NaN	

	Q19_NR
0	Not recorded
1	Not recorded
2	NaN
3	NaN
4	NaN
..	...
117	NaN
118	NaN
119	NaN
120	NaN
121	NaN

[122 rows x 250 columns]

```
import pandas as pd
df=pd.read_excel("/content/mean2.xlsx",sheet_name=0)
mv=df['marks'].mean()
df=df.fillna(mv)
print(mv)

df=df.drop_duplicates()
print(df)
```

24.75

	sno	names	marks
0	1	satya	24
1	2	sravani	25
2	3	bala	25
3	4	keerthi	25

```
import pandas as pd
df=pd.read_excel("/content/mean2.xlsx",sheet_name=0)
```

```
df.head(2)
df.tail(2)
```

	sno	names	marks
2	3	bala	25
3	4	keerthi	25

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

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

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

```
(523, 8)
```

```
df=pd.read_csv("/content/Player.csv")
data_manual=df.tail(10)
data_manual.to_csv("manual_data")
```

```
df=pd.read_csv("/content/manual_data")
print(df)
```

	Unnamed: 0	Player_Id	Player_Name	DOB	Batting_Hand	\
0	513	514	I Shivram	NaN	NaN	
1	514	515	SD Ranade	NaN	NaN	
2	515	516	TH Wijewardene	NaN	NaN	
3	516	517	AL Hill	NaN	NaN	
4	517	518	RJ Tucker	NaN	NaN	
5	518	519	Subroto Das	NaN	NaN	
6	519	520	K Srinivasan	NaN	NaN	
7	520	521	VK Sharma	NaN	NaN	
8	521	523	AV Wankhade	14-Mar-92	Right_Hand	
9	522	524	B Aparajith	8-Jul-94	Right_Hand	

	Bowling_Skill	Country	Is_Umpire	Unnamed: 7
0	NaN	India	1	NaN
1	NaN	India	1	NaN
2	NaN	Sri Lanka	1	NaN
3	NaN	New Zealand	1	NaN
4	NaN	Australia	1	NaN
5	NaN	India	1	NaN
6	NaN	India	1	NaN
7	NaN	India	1	NaN
8	NaN	India	0	NaN
9	Right-arm offbreak	India	0	NaN

```
import pandas as pd
df=pd.read_csv("/content/Player.csv")
data_first=df.tail(10)
data_second=df.head(10)
result=pd.concat([data_first,data_second],axis=0)
result.to_csv("result")
```

```
df=pd.read_csv("/content/result")
print(df)
```

	Unnamed: 0	Player_Id	Player_Name	DOB	Batting_Hand	\
0	513	514	I Shivram	NaN	NaN	
1	514	515	SD Ranade	NaN	NaN	
2	515	516	TH Wijewardene	NaN	NaN	
3	516	517	AL Hill	NaN	NaN	
4	517	518	RJ Tucker	NaN	NaN	
5	518	519	Subroto Das	NaN	NaN	
6	519	520	K Srinivasan	NaN	NaN	
7	520	521	VK Sharma	NaN	NaN	
8	521	523	AV Wankhade	14-Mar-92	Right_Hand	
9	522	524	B Aparajith	8-Jul-94	Right_Hand	
10	0	1	SC Ganguly	8-Jul-72	Left_Hand	
11	1	2	BB McCullum	27-Sep-81	Right_Hand	
12	2	3	RT Ponting	19-Dec-74	Right_Hand	
13	3	4	DJ Hussey	15-Jul-77	Right_Hand	
14	4	5	Mohammad Hafeez	17-Oct-80	Right_Hand	
15	5	6	R Dravid	11-Jan-73	Right_Hand	
16	6	7	W Jaffer	16-Feb-78	Right_Hand	
17	7	8	V Kohli	5-Nov-88	Right_Hand	
18	8	9	JH Kallis	16-Oct-75	Right_Hand	
19	9	10	CL White	18-Aug-83	Right_Hand	

	Bowling_Skill	Country	Is_Umpire	Unnamed: 7
0	NaN	India	1	NaN
1	NaN	India	1	NaN
2	NaN	Sri Lanka	1	NaN
3	NaN	New Zealand	1	NaN
4	NaN	Australia	1	NaN
5	NaN	India	1	NaN
6	NaN	India	1	NaN
7	NaN	India	1	NaN
8	NaN	India	0	NaN
9	Right-arm offbreak	India	0	NaN
10	Right-arm medium	India	0	NaN
11	Right-arm medium	New Zealand	0	NaN
12	Right-arm medium	Australia	0	NaN
13	Right-arm offbreak	Australia	0	NaN
14	Right-arm offbreak	Pakistan	0	NaN
15	Right-arm offbreak	India	0	NaN
16	Right-arm offbreak	India	0	NaN

17	Right-arm medium	India	0	NaN
18	Right-arm fast-medium	South Africa	0	NaN
19	Legbreak googly	Australia	0	NaN

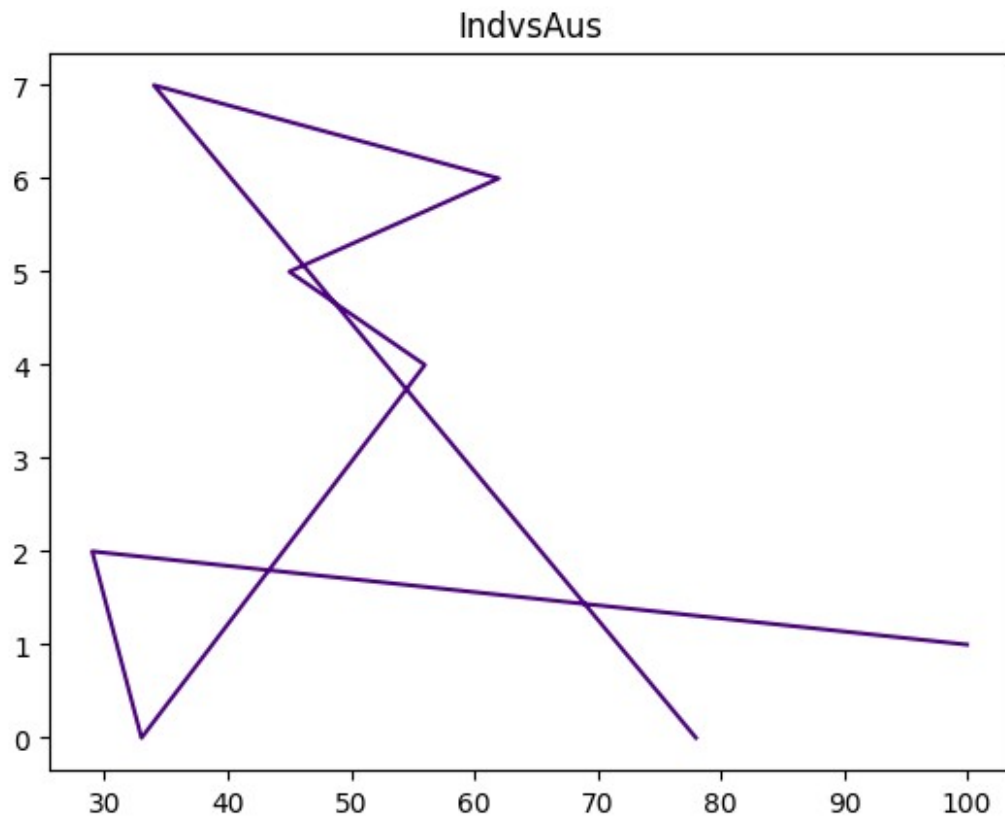
```
df=pd.read_csv("/content/result")
print(df.groupby(['Player_Id'])['Player_Name'].count())
```

Player\_Id

1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
514	1
515	1
516	1
517	1
518	1
519	1
520	1
521	1
523	1
524	1

Name: Player\_Name, dtype: int64

```
import numpy as np
import matplotlib.pyplot as plt
runs=np.array([100,29,33,56,45,62,34,78])
w=np.array([1,2,0,4,5,6,7,0])
plt.plot(runs,w,color='indigo')
plt.title('IndvsAus')
plt.show()
```

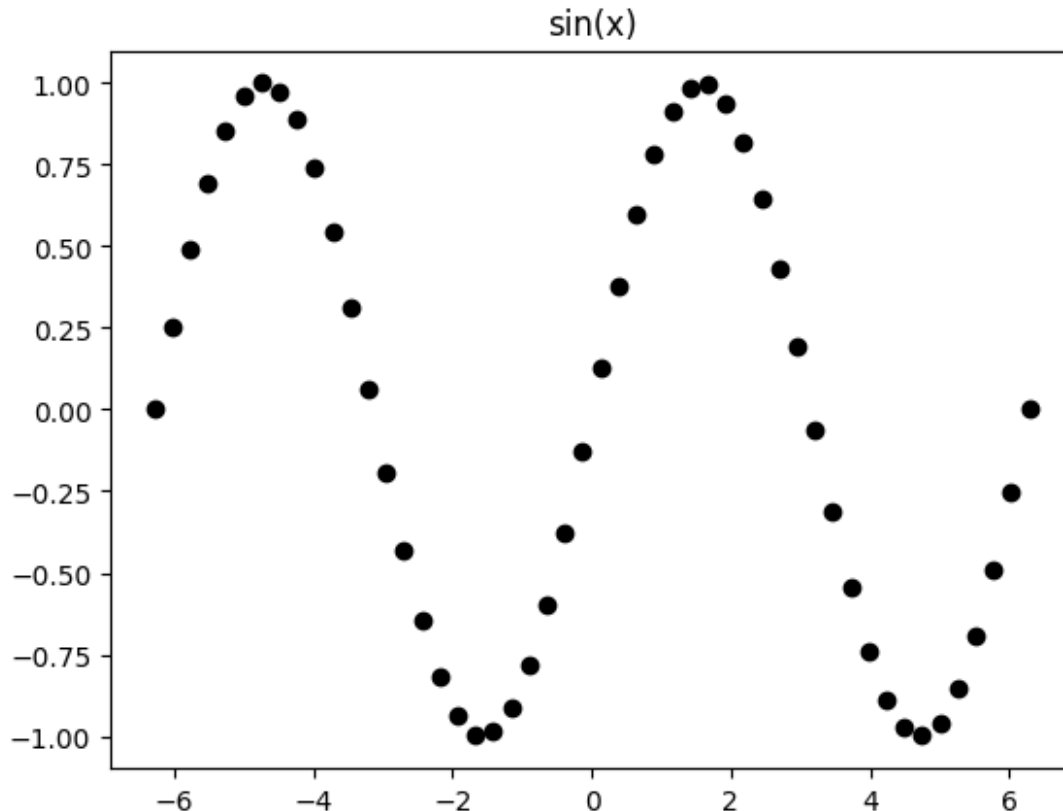


```
import numpy as np
import matplotlib.pyplot as plt
#Generate array of 200 values between -pi $ pi
tigar=np.linspace(-2*np.pi,2*np.pi,50)
print(tigar)

plt.scatter(tigar,np.sin(tigar),color='Black')
plt.title("sin(x)")
```

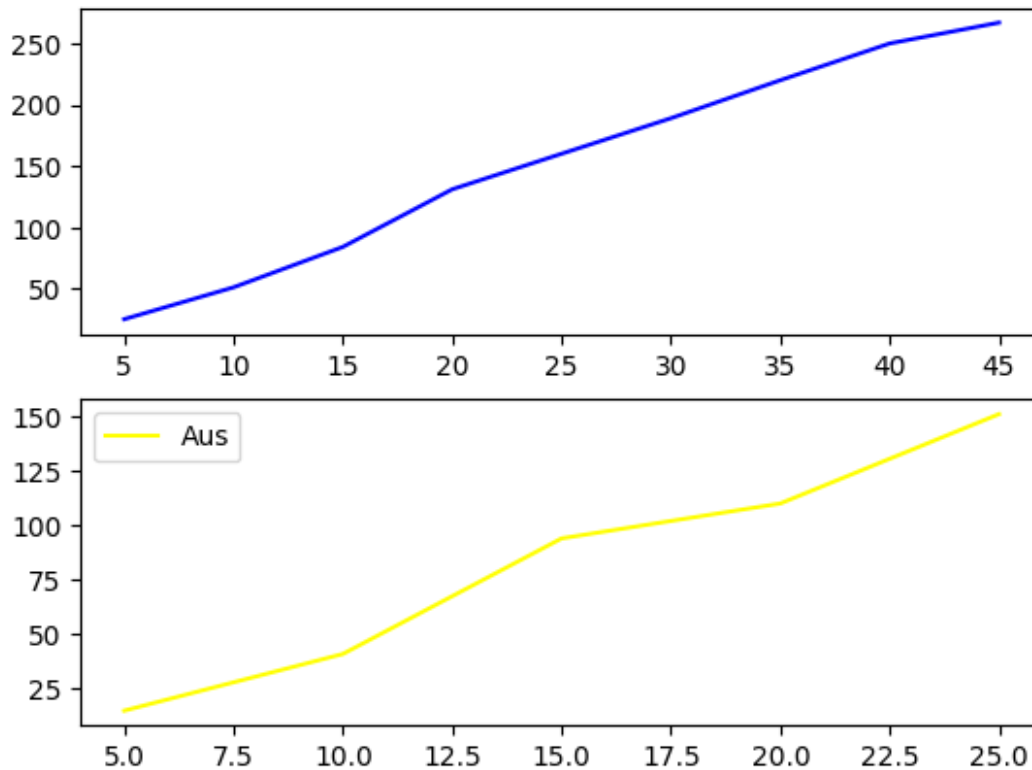
[-6.28318531	-6.02672876	-5.77027222	-5.51381568	-5.25735913	-
5.00090259	-4.74444605	-4.48798951	-4.23153296	-3.97507642	-3.71861988
-3.46216333	-3.20570679	-2.94925025	-2.6927937	-2.43633716	-2.17988062
-1.92342407	-1.66696753	-1.41051099	-1.15405444	-0.8975979	-0.64114136
-0.38468481	-0.12822827	0.12822827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



```
import numpy as np
import matplotlib.pyplot as plt
#creating X
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(2,1,1)
plt.plot(overs,runs_i,color='blue',label='India')
plt.subplot(2,1,2)
plt.legend(loc='best')
plt.plot(overs_a,runs_a,color='yellow',label='Aus')
plt.legend(loc='best')
plt.show()
```

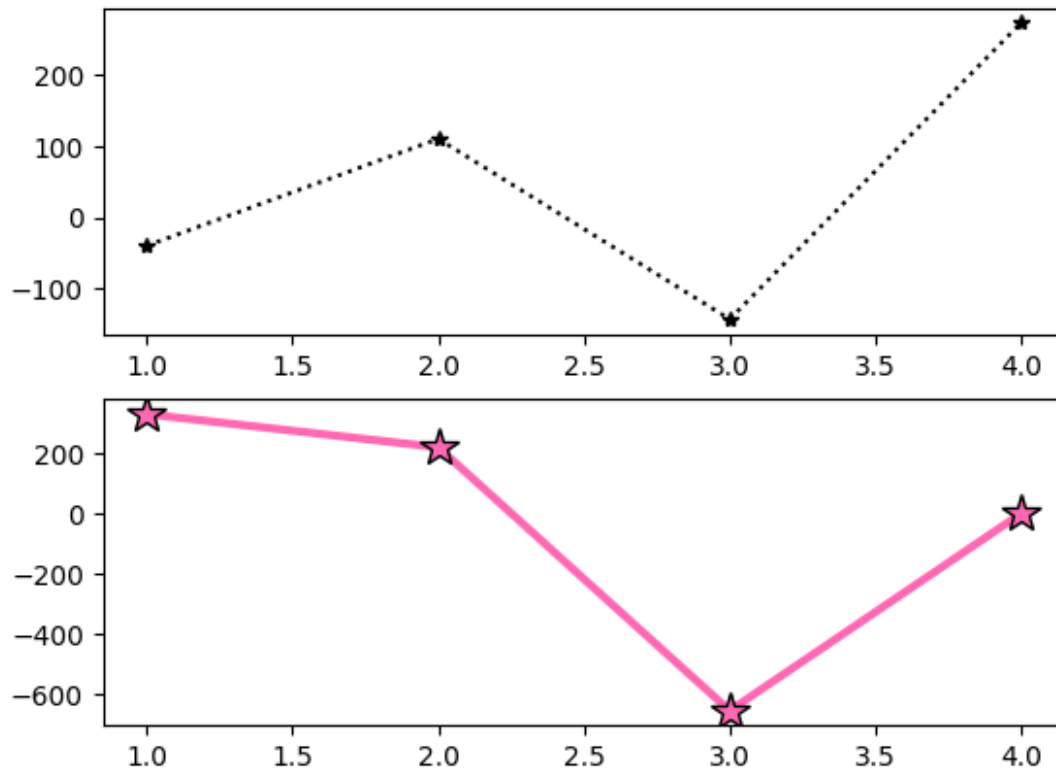
WARNING:matplotlib.legend:No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.



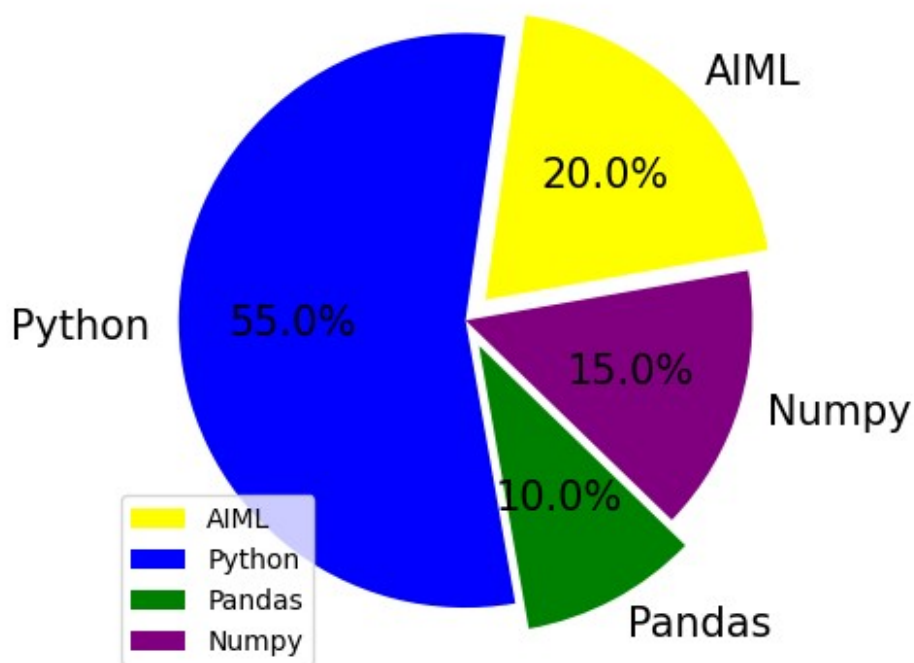
```
import matplotlib.pyplot as plt
a=[230,560,780,127,128]
b=[200,160,270,127,400]
years=[1,2,3,4]
profit_a=[(a[i]-a[i-1]) for i in range(1,len(a))]
profit_b=[(b[i]-b[i-1]) for i in range(1,len(b))]
plt.subplot(2,1,2)
plt.plot(years,profit_a,color='hotpink',linewidth='3',label='companyA',
,marker='*',ms='15',mec='k')
plt.subplot(2,1,1)
plt.plot(years,profit_b,color='black',linestyle='dotted',label='companyB',marker='*')
```

[<matplotlib.lines.Line2D at 0x7a9823be34c0>]





```
import matplotlib.pyplot as plt
a=np.array([20,55,10,15])
labe=["AIML", 'Python', 'Pandas', 'Numpy']
colour=['yellow', 'blue', 'green', 'purple']
explode=[0.1,0,0.1,0]
plt.pie(a,labels=labe,colors=colour, autopct="%1.1f%%",
textprops={'fontsize': 15}, explode=explode,startangle=10)
plt.legend()
plt.show()
```



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
data=pd.read_excel("/content/temp.xlsx")
data.to_csv("task.csv")
print(data)
```

	date	temperature
0	1	34
1	2	35
2	3	36
3	4	37
4	5	38
5	6	34
6	7	42
7	8	38
8	9	39
9	10	40
10	11	40
11	12	40
12	13	38
13	14	39
14	15	38
15	16	42
16	17	39
17	18	40
18	19	38

19	20	38
20	21	38
21	22	42
22	23	40
23	24	39
24	25	38
25	26	37
26	27	36
27	28	39
28	29	38
29	30	38

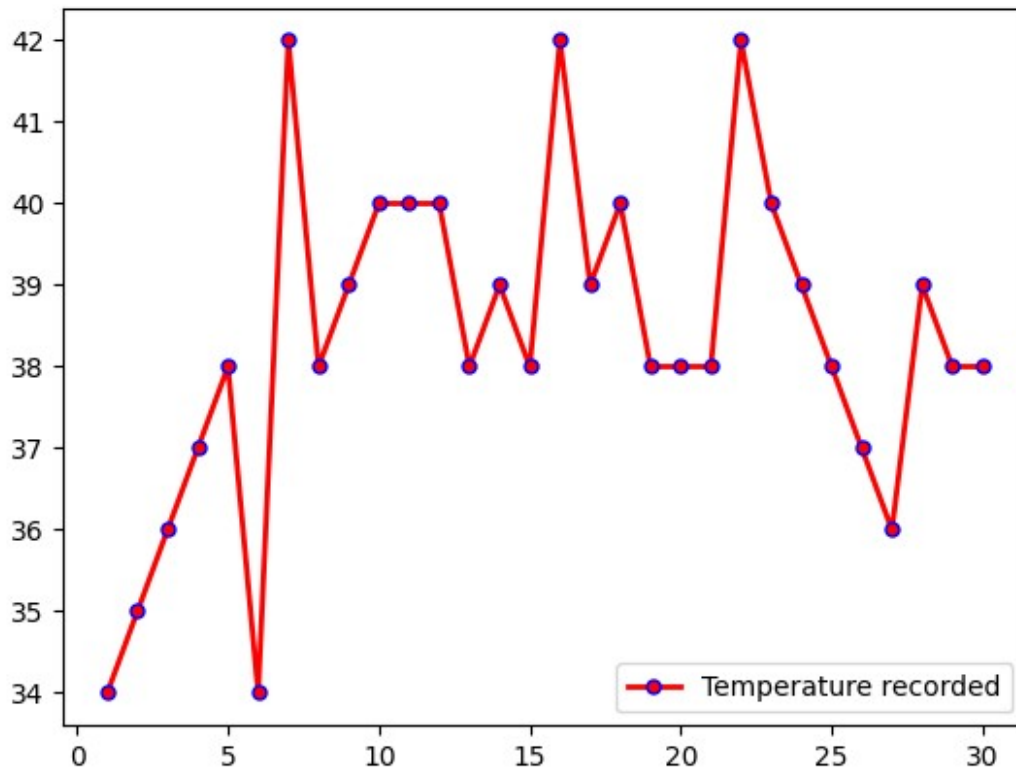
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
var1=pd.read_csv("/content/task.csv")
mean_val=var1['temperature'].mean()
var1=var1.fillna(mean_val)
a=np.array(var1['temperature'])
max_temp=np.max(a)
min_temp=np.min(a)
days=[a[i] for i in range(1, len(a)) if a[i]>mean_val]
date=np.array(var1['date'])
plt.plot(date,a,color='red',linewidth='2',label='Temperature
recorded',marker='.',ms='10',mec='b')
plt.legend(loc='best')
print('Mean of temperatures is',mean_val)
print('Maximum temperature is',max_temp)
print('Minimum temperature is',min_temp)
print('Number of days where the temperature exceeded a certain
threshold of',mean_val,'is',len(days),'days')
plt.show()
```

Mean of temperatures is 38.333333333333336

Maximum temperature is 42

Minimum temperature is 34

Number of days where the temperature exceeded a certain threshold of 38.333333333333336 is 13 days

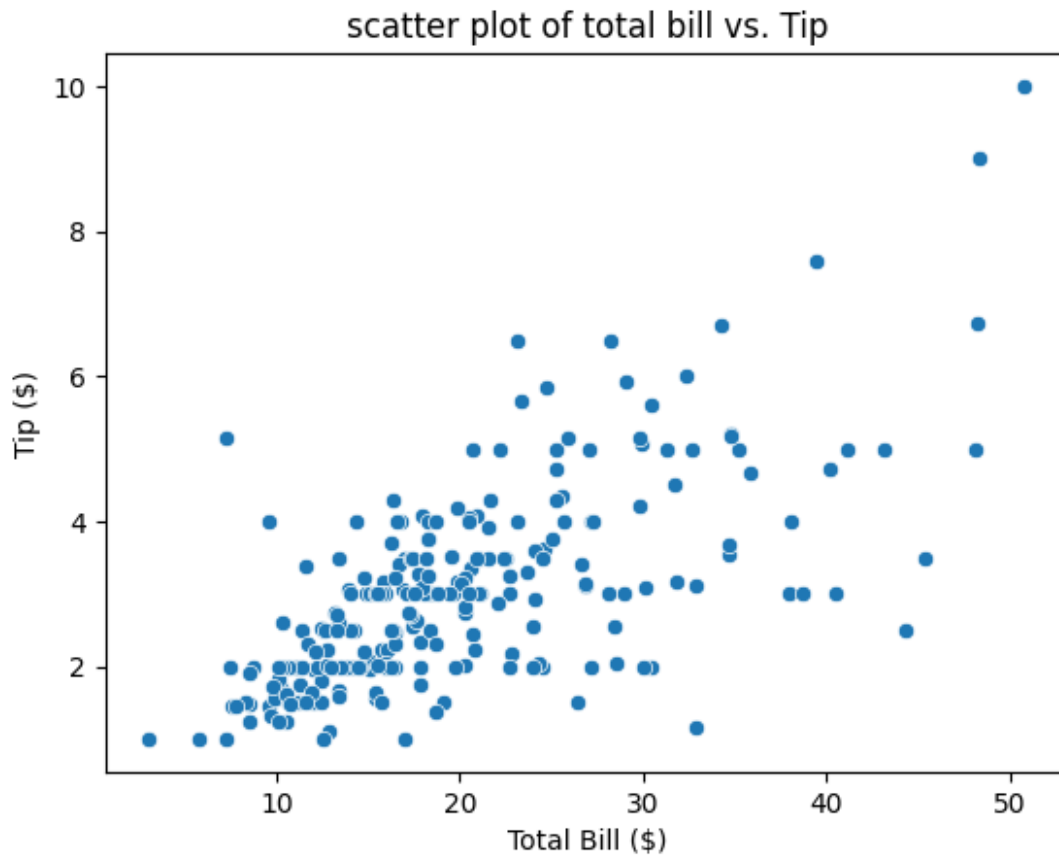


```
pip install seaborn
```

```
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) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (4.48.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (23.2)
```

Requirement already satisfied: pillow>=6.2.0 in  
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (9.4.0)  
Requirement already satisfied: pyparsing>=2.3.1 in  
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.1)  
Requirement already satisfied: python-dateutil>=2.7 in  
/usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in  
/usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn) (2023.4)  
Requirement already satisfied: six>=1.5 in  
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)

```
import seaborn as sns
import matplotlib.pyplot as plt
#Load example dataset
tips=sns.load_dataset("tips")
#Ceate a scatter plot
sns.scatterplot(x="total_bill",y="tip",data=tips)
plt.title("scatter plot of total bill vs. Tip")
plt.xlabel("Total Bill ($)")
plt.ylabel("Tip ($)")
plt.show()
```



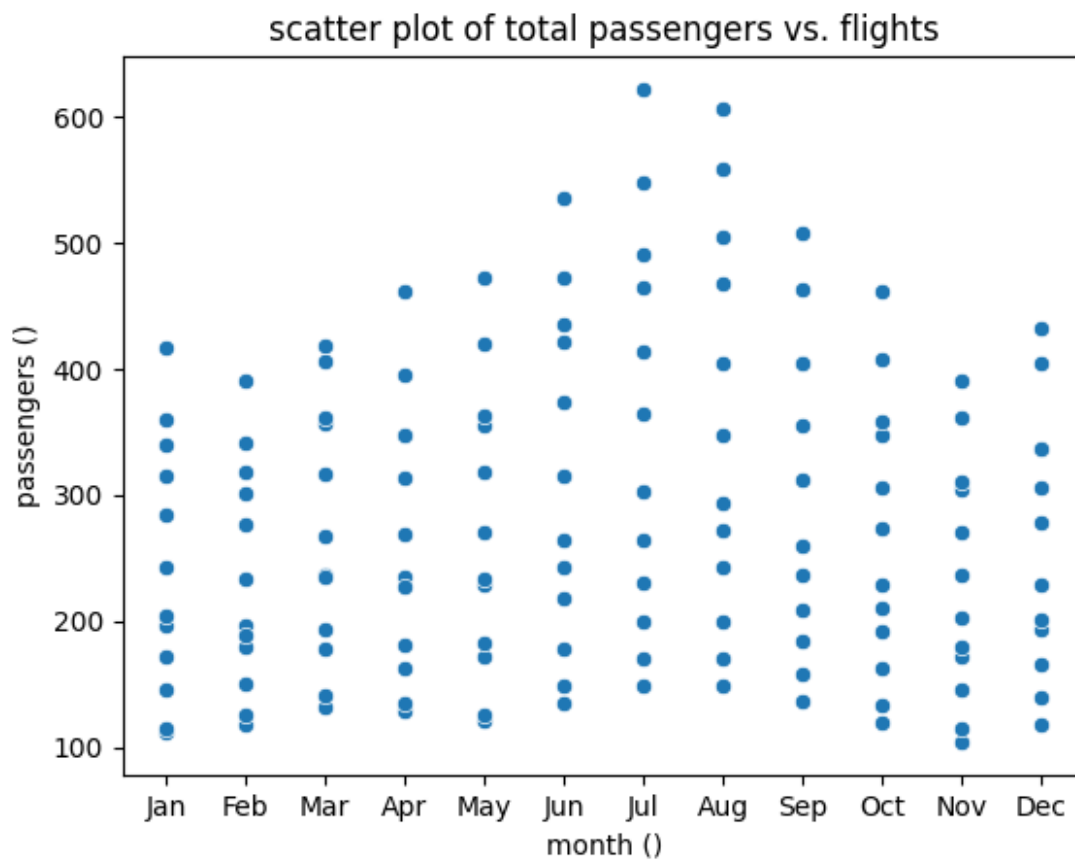
```
import seaborn as sns
import matplotlib.pyplot as plt
#Load example dataset
tips=sns.load_dataset("flights")
print(tips.head())
```

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121

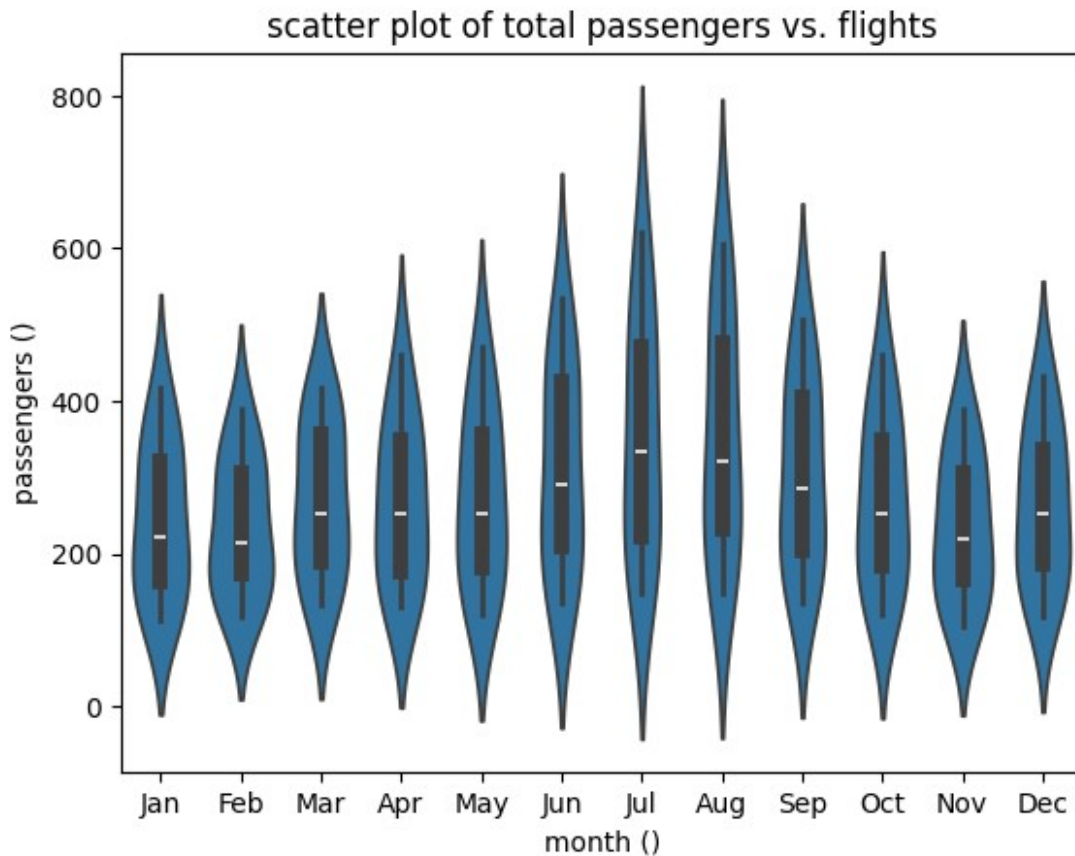
```
import seaborn as sns
import matplotlib.pyplot as plt
#Load example dataset
tips=sns.load_dataset("flights")

#Ceate a scatter plot
sns.scatterplot(x="month",y="passengers",data=tips)
plt.title("scatter plot of total passengers vs. flights")
plt.xlabel("month ")
```

```
plt.ylabel("passengers ()")  
plt.show()
```



```
import seaborn as sns  
import matplotlib.pyplot as plt  
#Load example dataset  
tips=sns.load_dataset("flights")  
sns.violinplot(x="month",y="passengers",data=tips)  
plt.title("scatter plot of total passengers vs. flights")  
plt.xlabel("month ()")  
plt.ylabel("passengers ()")  
plt.show()
```

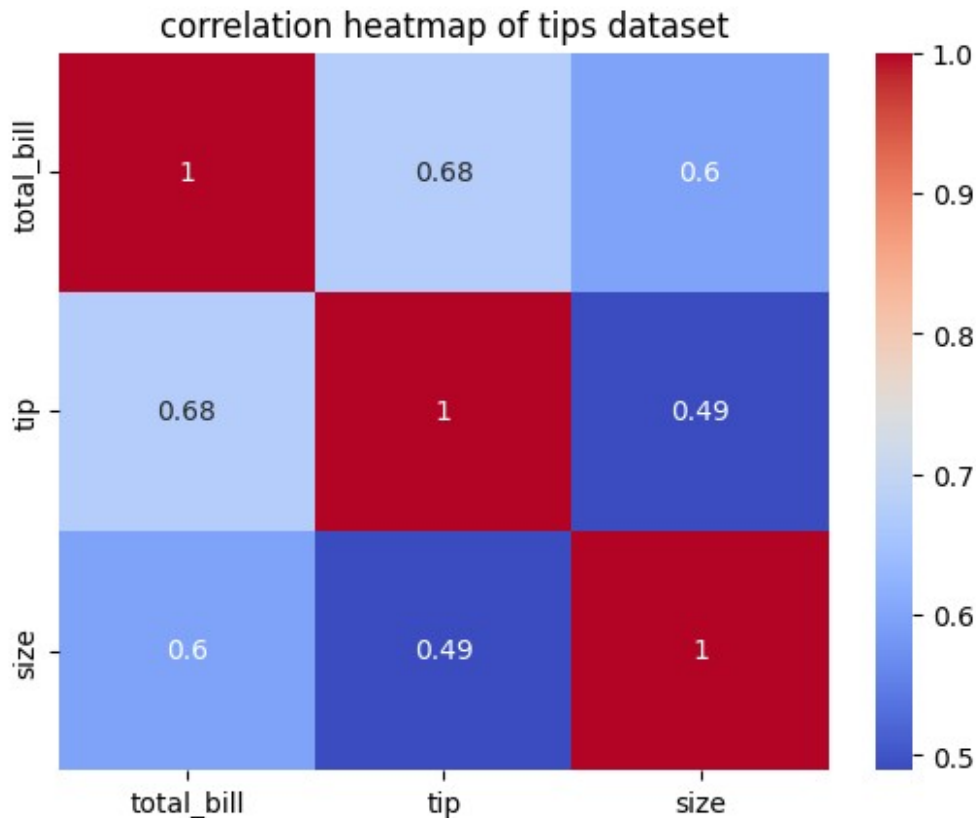


```
import seaborn as sns
import matplotlib.pyplot as plt
#Load example dataset
var1=sns.load_dataset("tips")
#compute correlation matrix
correlation_matrix=var1.corr()
#create a heatmap of the correlation matrix
sns.heatmap(correlation_matrix,annot=True,cmap="coolwarm")
plt.title("correlation heatmap of tips dataset")
plt.show()
```

<ipython-input-20-bbca969771a4>:6: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric\_only to silence this warning.

```
correlation_matrix=var1.corr()
```

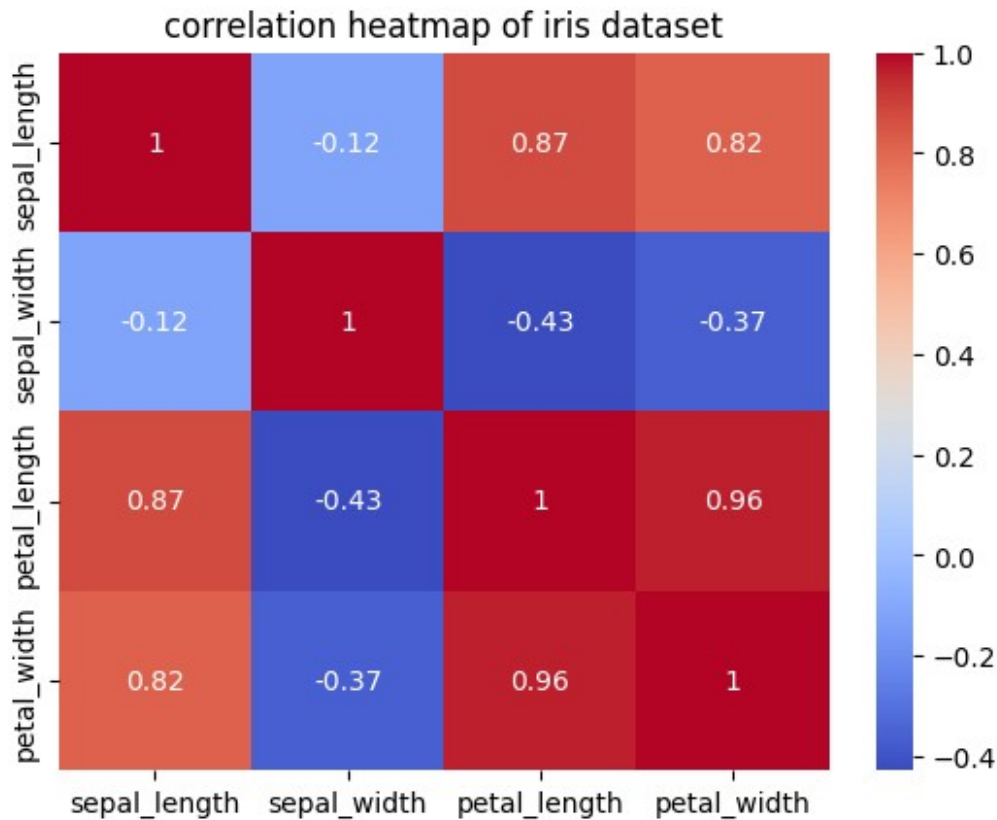




```
import seaborn as sns
import matplotlib.pyplot as plt
#Load example dataset
var1=sns.load_dataset("iris")
#compute correlation matrix
correlation_matrix=var1.corr()
#create a heatmap of the correlation matrix
sns.heatmap(correlation_matrix,annot=True,cmap="coolwarm")
plt.title("correlation heatmap of iris dataset")
plt.show()
```

<ipython-input-22-b1b09527bada>:6: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric\_only to silence this warning.

```
correlation_matrix=var1.corr()
```



```
import seaborn as sns
import matplotlib.pyplot as plt
#Load example dataset
var1=sns.load_dataset("flights")
#compute correlation matrix
correlation_matrix=var1.corr()
#create a heatmap of the correlation matrix
sns.heatmap(correlation_matrix,annot=True,cmap="coolwarm")
plt.title("correlation heatmap of flights dataset")
plt.show()
```

<ipython-input-24-4ccf1263bccf>:6: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric\_only to silence this warning.

```
correlation_matrix=var1.corr()
```

