

```
In [ ]: ### python Numpy&Matplotlib&Seaborn Practice ###
```

```
In [4]: pip install numpy
```

Requirement already satisfied: numpy in c:\pavan9\lib\site-packages (1.24.3)
Note: you may need to restart the kernel to use updated packages.

```
In [3]: pip install pytest
```

Requirement already satisfied: pytest in c:\pavan9\lib\site-packages (7.3.1)
Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: configparser in c:\pavan9\lib\site-packages (from pytest) (2.0.0)

Requirement already satisfied: packaging in c:\pavan9\lib\site-packages (from pytest) (23.1)

Requirement already satisfied: pluggy<2.0,>=0.12 in c:\pavan9\lib\site-packages (from pytest) (1.0.0)

Requirement already satisfied: exceptiongroup>=1.0.0rc8 in c:\pavan9\lib\site-packages (from pytest) (1.1.1)

Requirement already satisfied: tomli>=1.0.0 in c:\pavan9\lib\site-packages (from pytest) (2.0.1)

Requirement already satisfied: colorama in c:\pavan9\lib\site-packages (from pytest) (0.4.6)

```
In [5]: pip install pandas
```

Requirement already satisfied: pandas in c:\pavan9\lib\site-packages (2.0.2)

Requirement already satisfied: python-dateutil>=2.8.2 in c:\pavan9\lib\site-packages (from pandas) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in c:\pavan9\lib\site-packages (from pandas) (2023.3)

Requirement already satisfied: tzdata>=2022.1 in c:\pavan9\lib\site-packages (from pandas) (2023.3)

Requirement already satisfied: numpy>=1.21.0 in c:\pavan9\lib\site-packages (from pandas) (1.24.3)

Requirement already satisfied: six>=1.5 in c:\pavan9\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

Note: you may need to restart the kernel to use updated packages.

```
In [6]: import pandas as pd
```

```
In [8]: pip install matplotlib
```

Note: you may need to restart the kernel to use updated packages.

```
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.HTTPSConnection object at 0x0000022CAEE56500>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/matplotlib/
WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.HTTPSConnection object at 0x0000022CAEE56830>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/matplotlib/
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.HTTPSConnection object at 0x0000022CAEE56B60>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/matplotlib/
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.HTTPSConnection object at 0x0000022CAEE56D10>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/matplotlib/
WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.HTTPSConnection object at 0x0000022CAEE56EC0>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/matplotlib/
ERROR: Could not find a version that satisfies the requirement matplotlib (from versions: none)
ERROR: No matching distribution found for matplotlib
```

In [9]: pip install matplotlib

Collecting matplotlib

```
  Downloading matplotlib-3.7.1-cp310-cp310-win_amd64.whl (7.6 MB)
    0.0/7.6 MB ? eta :-:-:-
    0.1/7.6 MB 1.7 MB/s eta 0:00:05
    0.1/7.6 MB 1.4 MB/s eta 0:00:06
    0.1/7.6 MB 774.0 kB/s eta 0:00:10
    -
    -
    --
    --
    ---
```

0.3/7.6 MB 1.2 MB/s eta 0:00:07
0.4/7.6 MB 1.4 MB/s eta 0:00:06
0.5/7.6 MB 1.5 MB/s eta 0:00:05
0.5/7.6 MB 1.4 MB/s eta 0:00:06
0.6/7.6 MB 1.5 MB/s eta 0:00:05
0.7/7.6 MB 1.7 MB/s eta 0:00:05
0.8/7.6 MB 1.7 MB/s eta 0:00:04
1.0/7.6 MB 1.7 MB/s eta 0:00:04
1.1/7.6 MB 1.7 MB/s eta 0:00:04
1.2/7.6 MB 1.9 MB/s eta 0:00:04
1.3/7.6 MB 1.8 MB/s eta 0:00:04
1.4/7.6 MB 1.9 MB/s eta 0:00:04
1.4/7.6 MB 1.9 MB/s eta 0:00:04
1.5/7.6 MB 1.9 MB/s eta 0:00:04
1.6/7.6 MB 1.8 MB/s eta 0:00:04
1.8/7.6 MB 1.9 MB/s eta 0:00:04
1.8/7.6 MB 1.9 MB/s eta 0:00:04
2.0/7.6 MB 1.9 MB/s eta 0:00:03
2.1/7.6 MB 2.0 MB/s eta 0:00:03
2.3/7.6 MB 2.1 MB/s eta 0:00:03
2.4/7.6 MB 2.1 MB/s eta 0:00:03
2.6/7.6 MB 2.1 MB/s eta 0:00:03
2.7/7.6 MB 2.2 MB/s eta 0:00:03
3.0/7.6 MB 2.3 MB/s eta 0:00:03
3.1/7.6 MB 2.3 MB/s eta 0:00:02
3.3/7.6 MB 2.3 MB/s eta 0:00:02
3.5/7.6 MB 2.4 MB/s eta 0:00:02
3.6/7.6 MB 2.4 MB/s eta 0:00:02
3.8/7.6 MB 2.4 MB/s eta 0:00:02
3.9/7.6 MB 2.4 MB/s eta 0:00:02
4.1/7.6 MB 2.4 MB/s eta 0:00:02
4.3/7.6 MB 2.5 MB/s eta 0:00:02
4.5/7.6 MB 2.6 MB/s eta 0:00:02
4.7/7.6 MB 2.6 MB/s eta 0:00:02
4.9/7.6 MB 2.7 MB/s eta 0:00:02
5.1/7.6 MB 2.7 MB/s eta 0:00:01
5.4/7.6 MB 2.8 MB/s eta 0:00:01
5.9/7.6 MB 2.8 MB/s eta 0:00:01
5.9/7.6 MB 2.8 MB/s eta 0:00:01
6.2/7.6 MB 2.9 MB/s eta 0:00:01
6.2/7.6 MB 2.9 MB/s eta 0:00:01
6.2/7.6 MB 2.7 MB/s eta 0:00:01
6.5/7.6 MB 2.7 MB/s eta 0:00:01
6.6/7.6 MB 2.8 MB/s eta 0:00:01
6.7/7.6 MB 2.7 MB/s eta 0:00:01
6.8/7.6 MB 2.7 MB/s eta 0:00:01
7.1/7.6 MB 2.8 MB/s eta 0:00:01
7.4/7.6 MB 2.9 MB/s eta 0:00:01
7.5/7.6 MB 2.9 MB/s eta 0:00:01
7.6/7.6 MB 2.9 MB/s eta 0:00:01
7.6/7.6 MB 2.8 MB/s eta 0:00:00

Collecting contourpy>=1.0.1 (from matplotlib)

```
  Downloading contourpy-1.1.0-cp310-cp310-win_amd64.whl (470 kB)
```

```
          0.0/470.4 kB ? eta -:-:--  
----- 327.7/470.4 kB 6.8 MB/s eta 0:00:01  
----- 470.4/470.4 kB 5.9 MB/s eta 0:00:00  
Collecting cycler>=0.10 (from matplotlib)  
  Downloading cycler-0.11.0-py3-none-any.whl (6.4 kB)  
Collecting fonttools>=4.22.0 (from matplotlib)  
  Downloading fonttools-4.40.0-cp310-cp310-win_amd64.whl (1.9 MB)  
          0.0/1.9 MB ? eta -:-:--  
----- 0.2/1.9 MB 3.6 MB/s eta 0:00:01  
----- 0.5/1.9 MB 5.0 MB/s eta 0:00:01  
----- 0.6/1.9 MB 4.5 MB/s eta 0:00:01  
----- 0.8/1.9 MB 4.5 MB/s eta 0:00:01  
----- 1.1/1.9 MB 4.8 MB/s eta 0:00:01  
----- 1.4/1.9 MB 5.0 MB/s eta 0:00:01  
----- 1.5/1.9 MB 4.7 MB/s eta 0:00:01  
----- 1.7/1.9 MB 4.6 MB/s eta 0:00:01  
----- 1.9/1.9 MB 4.6 MB/s eta 0:00:01  
----- 1.9/1.9 MB 4.3 MB/s eta 0:00:00  
Collecting kiwisolver>=1.0.1 (from matplotlib)  
  Downloading kiwisolver-1.4.4-cp310-cp310-win_amd64.whl (55 kB)  
          0.0/55.3 kB ? eta -:-:--  
----- 30.7/55.3 kB ? eta -:-:--  
----- 55.3/55.3 kB 205.6 kB/s eta 0:00:00  
Requirement already satisfied: numpy>=1.20 in c:\pavan9\lib\site-packages (from matplotlib) (1.24.3)  
Requirement already satisfied: packaging>=20.0 in c:\pavan9\lib\site-packages (from matplotlib) (23.1)  
Collecting pillow>=6.2.0 (from matplotlib)  
  Downloading Pillow-9.5.0-cp310-cp310-win_amd64.whl (2.5 MB)  
          0.0/2.5 MB ? eta -:-:--  
--- 0.2/2.5 MB 5.9 MB/s eta 0:00:01  
----- 0.4/2.5 MB 5.1 MB/s eta 0:00:01  
----- 0.7/2.5 MB 5.8 MB/s eta 0:00:01  
----- 0.8/2.5 MB 4.9 MB/s eta 0:00:01  
----- 1.0/2.5 MB 4.8 MB/s eta 0:00:01  
----- 1.1/2.5 MB 4.2 MB/s eta 0:00:01  
----- 1.3/2.5 MB 4.2 MB/s eta 0:00:01  
----- 1.6/2.5 MB 4.3 MB/s eta 0:00:01  
----- 1.9/2.5 MB 4.7 MB/s eta 0:00:01  
----- 2.1/2.5 MB 4.5 MB/s eta 0:00:01  
----- 2.3/2.5 MB 4.6 MB/s eta 0:00:01  
----- 2.3/2.5 MB 4.6 MB/s eta 0:00:01  
----- 2.4/2.5 MB 4.0 MB/s eta 0:00:01  
----- 2.5/2.5 MB 4.0 MB/s eta 0:00:01  
----- 2.5/2.5 MB 3.8 MB/s eta 0:00:00  
Collecting pyparsing>=2.3.1 (from matplotlib)  
  Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)  
          0.0/98.3 kB ? eta -:-:--  
----- 98.3/98.3 kB 5.5 MB/s eta 0:00:00  
Requirement already satisfied: python-dateutil>=2.7 in c:\pavan9\lib\site-packages (from matplotlib) (2.8.2)  
Requirement already satisfied: six>=1.5 in c:\pavan9\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)  
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler, contourpy, matplotlib  
Successfully installed contourpy-1.1.0 cycler-0.11.0 fonttools-4.40.0 kiwisolver-
```

1.4.4 matplotlib-3.7.1 pillow-9.5.0 pyparsing-3.0.9

Note: you may need to restart the kernel to use updated packages.

In [10]: pip install seaborn

```

Collecting seaborn
  Downloading seaborn-0.12.2-py3-none-any.whl (293 kB)
    0.0/293.3 kB ? eta -:-:--
    ----
    30.7/293.3 kB ? eta -:-:--
    ----
    30.7/293.3 kB ? eta -:-:--
    -----
    92.2/293.3 kB 655.4 kB/s eta 0:00:01
    -----
    112.6/293.3 kB 544.7 kB/s eta 0:00:01
    -----
    194.6/293.3 kB 908.0 kB/s eta 0:00:01
    -----
    286.7/293.3 kB 983.9 kB/s eta 0:00:01
    -----
    286.7/293.3 kB 983.9 kB/s eta 0:00:01
    -----
    293.3/293.3 kB 788.4 kB/s eta 0:00:00

Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\pavan9\lib\site-packages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in c:\pavan9\lib\site-packages (from seaborn) (2.0.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\pavan9\lib\site-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.1.0)
Requirement already satisfied: cycler>=0.10 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.40.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\pavan9\lib\site-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: tzdata>=2022.1 in c:\pavan9\lib\site-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\pavan9\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
Installing collected packages: seaborn
Successfully installed seaborn-0.12.2
Note: you may need to restart the kernel to use updated packages.

```

In [1]: pip install numpy

Requirement already satisfied: numpy in c:\pavan9\lib\site-packages (1.24.3)

Note: you may need to restart the kernel to use updated packages.

In [2]: import numpy as np

In []: Numpy is used for working with arrays and matrix data structures

In [4]: print(np.__version__)

1.24.3

```
In [ ]: Creating Arrays
```

```
In [5]: a=np.array(10)          #0-Dimension array#
print(a)
```

```
10
```

```
In [7]: a=np.array([10,20,30]) #1-Dimension array#The difference between array and List
array separated with space and list elements separated
print(a)
```

```
[10 20 30]
```

```
In [9]: a=np.array([[10,20],[30,40]])##2-Dimensional Array##
print(a)
```

```
[[10 20]
 [30 40]]
```

```
In [10]: a[1][0]
```

```
Out[10]: 30
```

```
In [11]: a=np.array([[[10,20],[30,40]],[[50,60],[70,80]]])###3 dimension###
print(a)
```

```
[[[10 20]
 [30 40]

 [[50 60]
 [70 80]]]
```

```
In [12]: a[1][1][0]
```

```
Out[12]: 70
```

```
In [14]: a=[10,20,30]
b=np.asarray(a,dtype=float)
print(b)
```

```
[10. 20. 30.]
```

```
In [16]: a=[[10,20],[30,40]]
b=np.asarray(a,dtype=int,order="C")#C-MEANS ----Row Major Order##
print(b)
```

```
[[10 20]
 [30 40]]
```

```
In [ ]: For observing order difference we are using nditer() function
```

```
In [17]: for i in np.nditer(b):
    print(i)
```

```
10
20
30
40
```

```
In [19]: a=[[10,20],[30,40]]
b=np.asarray(a,dtype=int,order="F")####F---means column Major Order#
```

```
print(b)
```

```
[[10 20]
 [30 40]]
```

```
In [20]: for i in np.nditer(b):
    print(i)
```

```
10
30
20
40
```

```
In [24]: import numpy as np
```

```
a = b"welcome to numpy"
b = np.frombuffer(a, dtype="S1", count=10)
print(b)
```

```
[b'w' b'e' b'l' b'c' b'o' b'm' b'e' b' ' b't' b'o']
```

```
In [25]: import numpy as np
```

```
a = b"welcome to numpy"
b = np.frombuffer(a, dtype="S1", offset=9)
print(b)
```

```
[b'o' b' ' b'n' b'u' b'm' b'p' b'y']
```

```
In [ ]: Initializing Arrays
```

```
In [27]: a=np.zeros(3)
print(a)
```

```
[0. 0. 0.]
```

```
In [29]: a=np.zeros([3,3])
print(a)
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

```
In [30]: a=np.zeros([3,3,3])
print(a)
```

```
[[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]]
```

```
[[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]]
```

```
[[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]]]
```

```
In [33]: a=np.full([2,3],1)
print(a)
```

```
[[1 1 1]
 [1 1 1]]
```

```
In [35]: a=np.full([2,3,5],10)
print(a)
```

```
[[[10 10 10 10 10]
 [10 10 10 10 10]
 [10 10 10 10 10]]]
```

```
[[[10 10 10 10 10]
 [10 10 10 10 10]
 [10 10 10 10 10]]]
```

```
In [37]: a=np.random.rand(2,3,6)
print(a)
```

```
[[[0.78166242 0.79206377 0.84691469 0.8414453 0.73292303 0.2990105 ]
 [0.95041393 0.09947303 0.61925336 0.51708761 0.95606922 0.90219233]
 [0.79202305 0.59961704 0.45763684 0.51221164 0.02789825 0.54634397]]]
```

```
[[[0.21231884 0.94939272 0.07606676 0.37379355 0.68598442 0.37292219]
 [0.13668423 0.93429479 0.30910141 0.16633127 0.11822894 0.74959484]
 [0.08582567 0.10033764 0.09769458 0.25594642 0.43589449 0.44787349]]]
```

```
In [38]: a=np.eye(2)
print(a)
```

```
[[1. 0.]
 [0. 1.]]
```

```
In [41]: %%%%arange%%%%
```

```
[10. 20. 30. 40. 50. 60. 70. 80. 90.]
```

```
In [46]: a=np.arange(10,110,10,dtype=float)
print(a)
```

```
[ 10.  20.  30.  40.  50.  60.  70.  80.  90. 100.]
```

```
In [47]: a.reshape(2,5)
```

```
Out[47]: array([[ 10.,  20.,  30.,  40.,  50.],
 [ 60.,  70.,  80.,  90., 100.]])
```

```
In [49]: a=np.linspace(10,100,10,endpoint=False)----#Here 10 is equal intervals required#
print(a)
```

```
[10. 19. 28. 37. 46. 55. 64. 73. 82. 91.]
```

```
In [50]: a=np.logspace(0,10,10,base=2,dtype=int)
print(a)
```

```
[ 1  2  4 10 21 47 101 219 474 1024]
```

```
In [ ]: Array Properties
```

```
In [52]: a=np.arange(10,110,10)
print(a)
b=a.reshape(2,5)
print(b)
```

```
[ 10  20  30  40  50  60  70  80  90 100]
[[ 10  20  30  40  50]
 [ 60  70  80  90 100]]
```

```
In [53]: np.size(b)
```

```
Out[53]: 10
```

```
In [54]: np.shape(b)
```

```
Out[54]: (2, 5)
```

```
In [57]: b.dtype
```

```
Out[57]: dtype('int32')
```

```
In [ ]: #####Array operations#####
```

```
In [60]: a=np.array([[10,20,30],[40,50,60]])  
print(a)
```

```
[[10 20 30]  
 [40 50 60]]
```

```
In [61]: a[0]
```

```
Out[61]: array([10, 20, 30])
```

```
In [62]: a[0][1]
```

```
Out[62]: 20
```

```
In [63]: s=np.arange(10,100,10)  
print(s)
```

```
[10 20 30 40 50 60 70 80 90]
```

```
In [64]: s[2:5]
```

```
Out[64]: array([30, 40, 50])
```

```
In [65]: a=np.array([10,20,30])  
print(a)
```

```
[10 20 30]
```

```
In [66]: b=np.copy(a)  
print(b)
```

```
[10 20 30]
```

```
In [67]: c=b.view()#view is reference of b,that means if something is changed in b,it refl  
print(c)
```

```
[10 20 30]
```

```
In [68]: print(b)
```

```
[10 20 30]
```

```
In [71]: b[0]=300  
print(b)  
print(c)
```

```
[300  20  30]  
[300  20  30]
```

```
In [74]: a=np.array([10,20,30,60,40,60,40,90])  
print(a)  
b=a.reshape(2,4)  
print(b)
```

```
[10 20 30 60 40 60 40 90]  
[[10 20 30 60]  
 [40 60 40 90]]
```

```
In [75]: print(b)
```

```
[[10 20 30 60]  
 [40 60 40 90]]
```

```
In [76]: np.sort(b)
```

```
Out[76]: array([[10, 20, 30, 60],  
                 [40, 40, 60, 90]])
```

```
In [78]: np.sort(b, axis=0)
```

```
Out[78]: array([[10, 20, 30, 60],  
                 [40, 60, 40, 90]])
```

```
In [79]: np.sort(b, axis=1)
```

```
Out[79]: array([[10, 20, 30, 60],  
                 [40, 40, 60, 90]])
```

```
In [81]: a=np.array([10,20,30])  
b=np.array([40,50,60])  
c=np.append(a,b)  
print(c)
```

```
[10 20 30 40 50 60]
```

```
In [82]: np.insert(c,1,35)
```

```
Out[82]: array([10, 35, 20, 30, 40, 50, 60])
```

```
In [86]: np.delete(c,3)  
print(c)
```

```
[10 20 30 40 50 60]
```

```
In [88]: c=np.concatenate((a,b))  
print(c)
```

```
[10 20 30 40 50 60]
```

```
In [90]: print(a)  
print(b)
```

```
[10 20 30]  
[40 50 60]
```

```
In [92]: res=np.stack((a,b))  
print(res)
```

```
[[10 20 30]
 [40 50 60]]
```

```
In [94]: a=np.array([10,20,30,40,50,60])
a.reshape(3,2)
```

```
Out[94]: array([[10, 20],
 [30, 40],
 [50, 60]])
```

```
In [95]: b=np.array([100,200,300,400,500,600])
b.reshape(3,2)
```

```
Out[95]: array([[100, 200],
 [300, 400],
 [500, 600]])
```

```
In [98]: a=np.array([[100, 200],
 [300, 400],
 [500, 600]])
```

```
In [99]: b=np.array([[10, 20],
 [30, 40],
 [50, 60]])
```

```
In [100... a,b
```

```
Out[100]: (array([[100, 200],
 [300, 400],
 [500, 600]]),
 array([[10, 20],
 [30, 40],
 [50, 60]]))
```

```
In [101... np.vstack((a,b))
```

```
Out[101]: array([[100, 200],
 [300, 400],
 [500, 600],
 [ 10,  20],
 [ 30,  40],
 [ 50,  60]])
```

```
In [102... np.hstack((a,b))
```

```
Out[102]: array([[100, 200, 10, 20],
 [300, 400, 30, 40],
 [500, 600, 50, 60]])
```

```
In [103... np.dstack((a,b))
```

```
Out[103]: array([[[100, 10],
 [200, 20]],
 [[300, 30],
 [400, 40]],
 [[500, 50],
 [600, 60]]])
```

```
In [105...]: x=np.arange(10,110,10)
           print(x)
```

```
[ 10  20  30  40  50  60  70  80  90 100]
```

```
In [108...]: x
```

```
Out[108]: array([ 10,  20,  30,  40,  50,  60,  70,  80,  90, 100])
```

```
In [107...]: m=np.split(x,2)
           print(m)
```

```
[array([10, 20, 30, 40, 50]), array([ 60,  70,  80,  90, 100])]
```

```
In [111...]: a=np.array([10,20,30])
           b=np.array([40,50,60])
```

```
In [112...]: print(a,b)
```

```
[10 20 30] [40 50 60]
```

```
In [113...]: c=np.add(a,b)
           print(c)
```

```
[50 70 90]
```

```
In [115...]: import numpy as np

a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
d = np.subtract(a, b)
print(d)
```

```
[-3 -3 -3]
```

```
In [116...]: import numpy as np
```

```
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
d = np.multiply(a, b)
print(d)
```

```
[ 4 10 18]
```

```
In [117...]: a=np.array([10,20,30,40,50,60,70,80,90])
```

```
In [118...]: print(a)
```

```
[10 20 30 40 50 60 70 80 90]
```

```
In [119...]: b=np.min(a)
           print(b)
```

```
10
```

```
In [120...]: c=np.max(a)
           print(c)
```

```
90
```

```
In [121...]: d=np.sum(a)
           print(d)
```

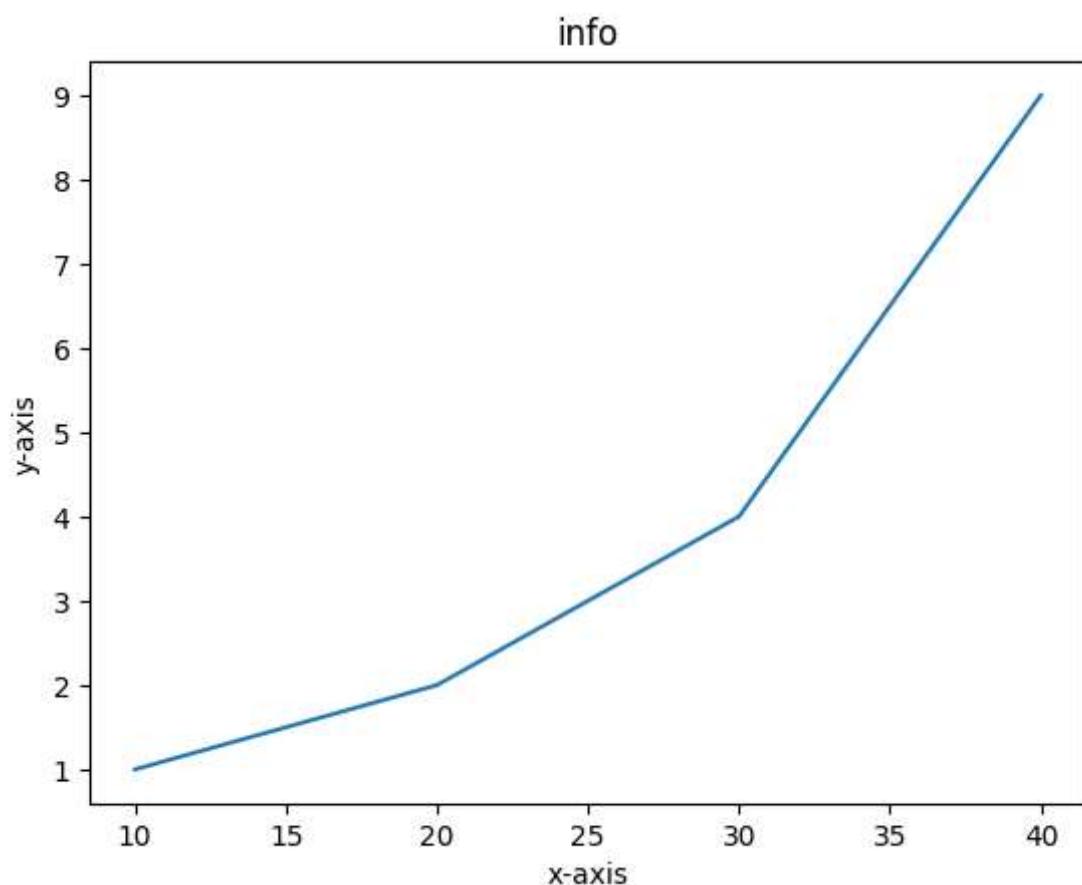
450

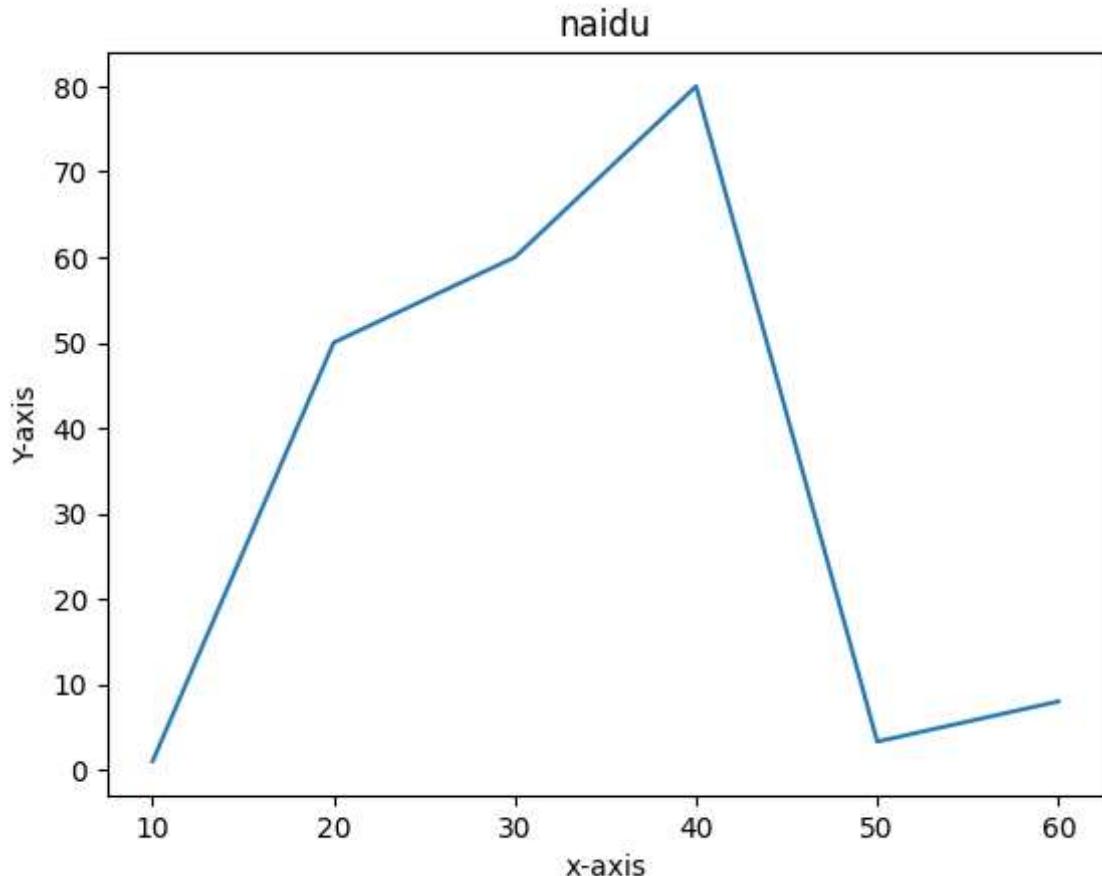
In []:

Matplotlib

In []: Matplotlib is one of the most popular python packages used for data visualization.

In []: it is a cross - platform library for making 2d plots from data in arrays.

In [11]: `from matplotlib import pyplot as plt`In [13]:
`x=[10,20,30,40]
y=[1,2,4,9]
plt.plot(x,y)
plt.title("info")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()`In [14]:
`x=[10,20,30,40,50,60]
y=[1,50,60,80,3.3,8]
plt.plot(x,y)
plt.title("naidu")
plt.xlabel("x-axis")
plt.ylabel("Y-axis")
plt.show()`



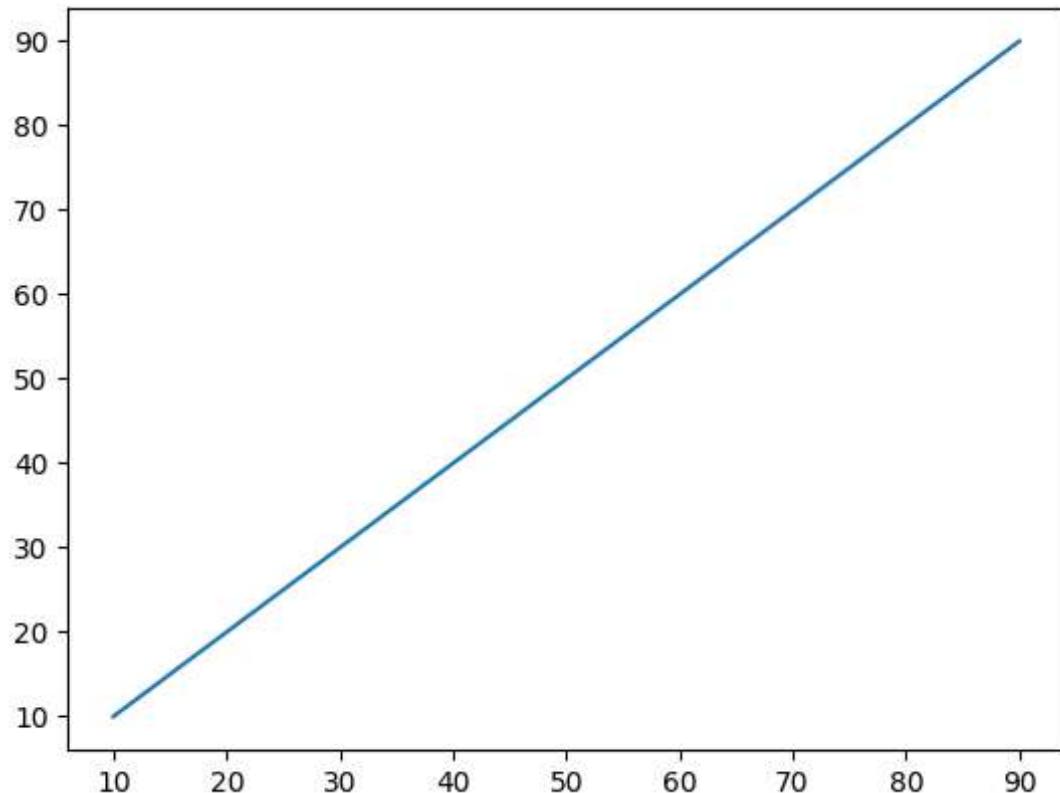
```
In [16]: import numpy as np
```

```
In [17]: a=np.arange(10,100,10)
```

```
In [18]: b=np.arange(10,100,10)
```

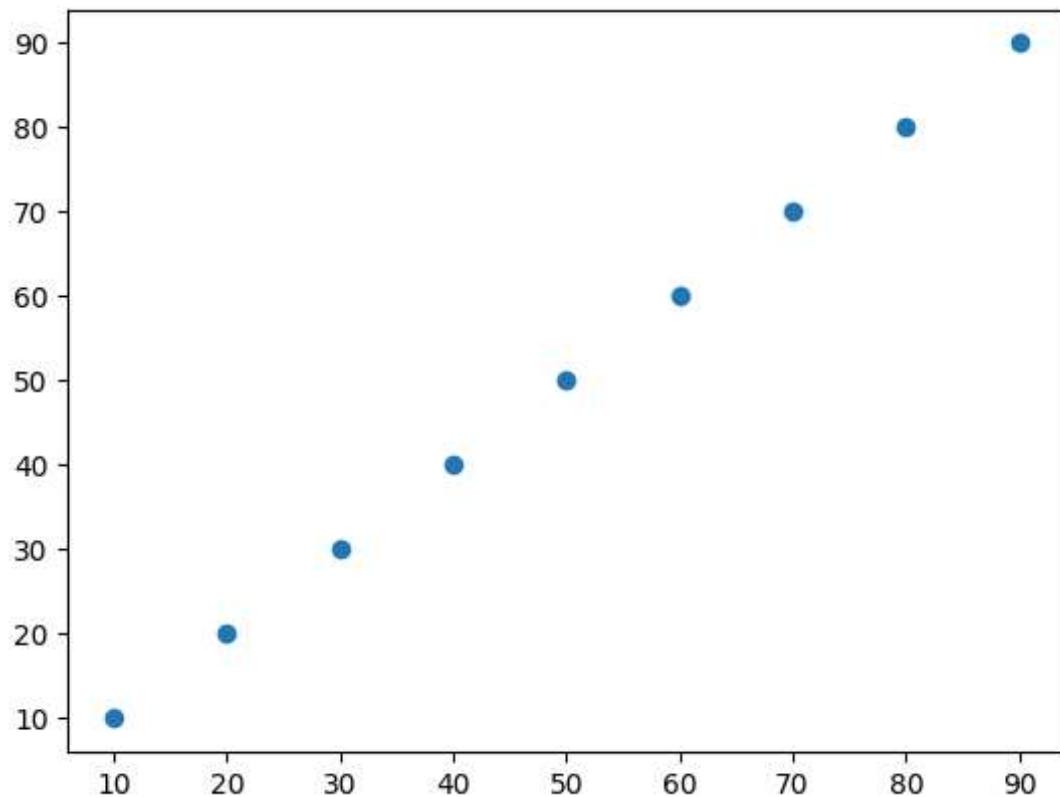
```
In [20]: plt.plot(a,b)
```

```
Out[20]: [<matplotlib.lines.Line2D at 0x1fc...>]
```



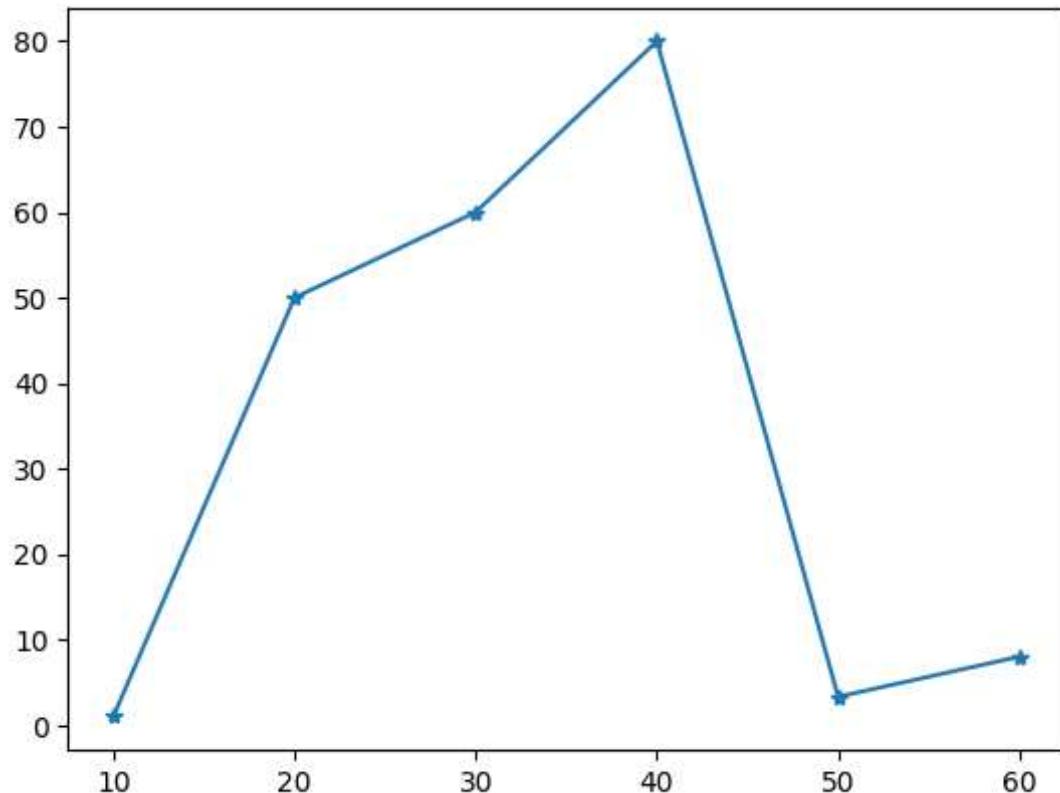
```
In [21]: plt.plot(a,b, "o")
```

```
Out[21]: [<matplotlib.lines.Line2D at 0x1fcfa78eba90>]
```



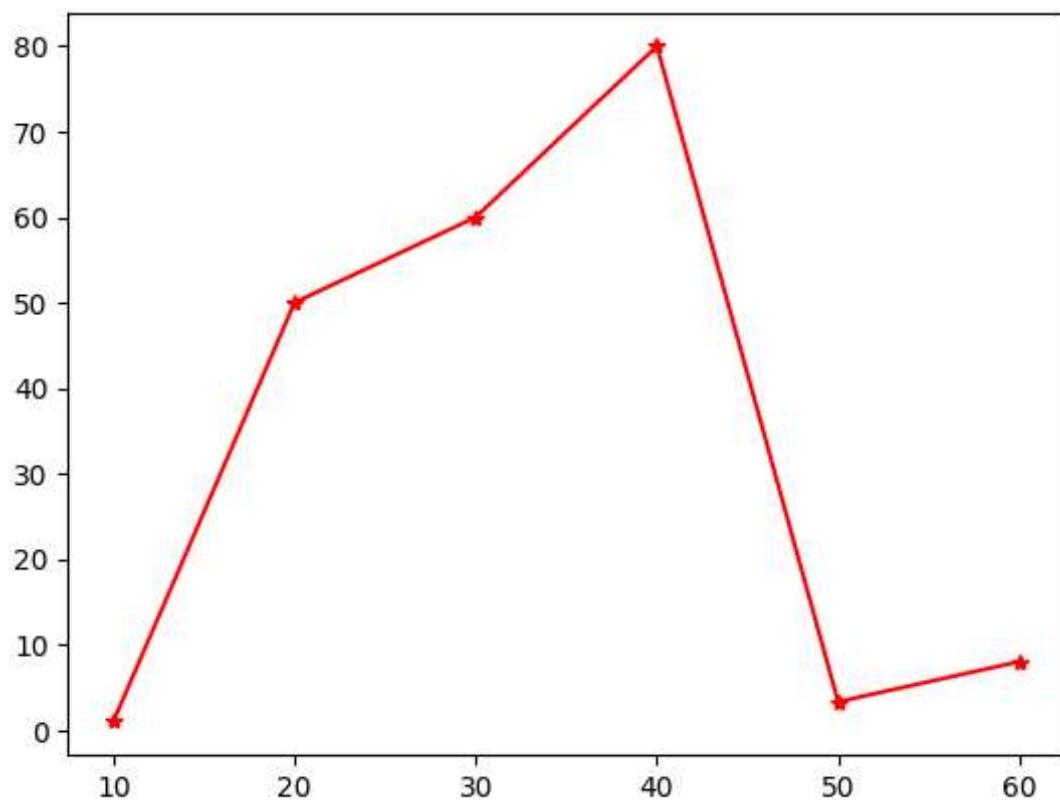
```
In [22]: plt.plot(x,y,marker="*")
```

```
Out[22]: [<matplotlib.lines.Line2D at 0x1fcfa7994370>]
```



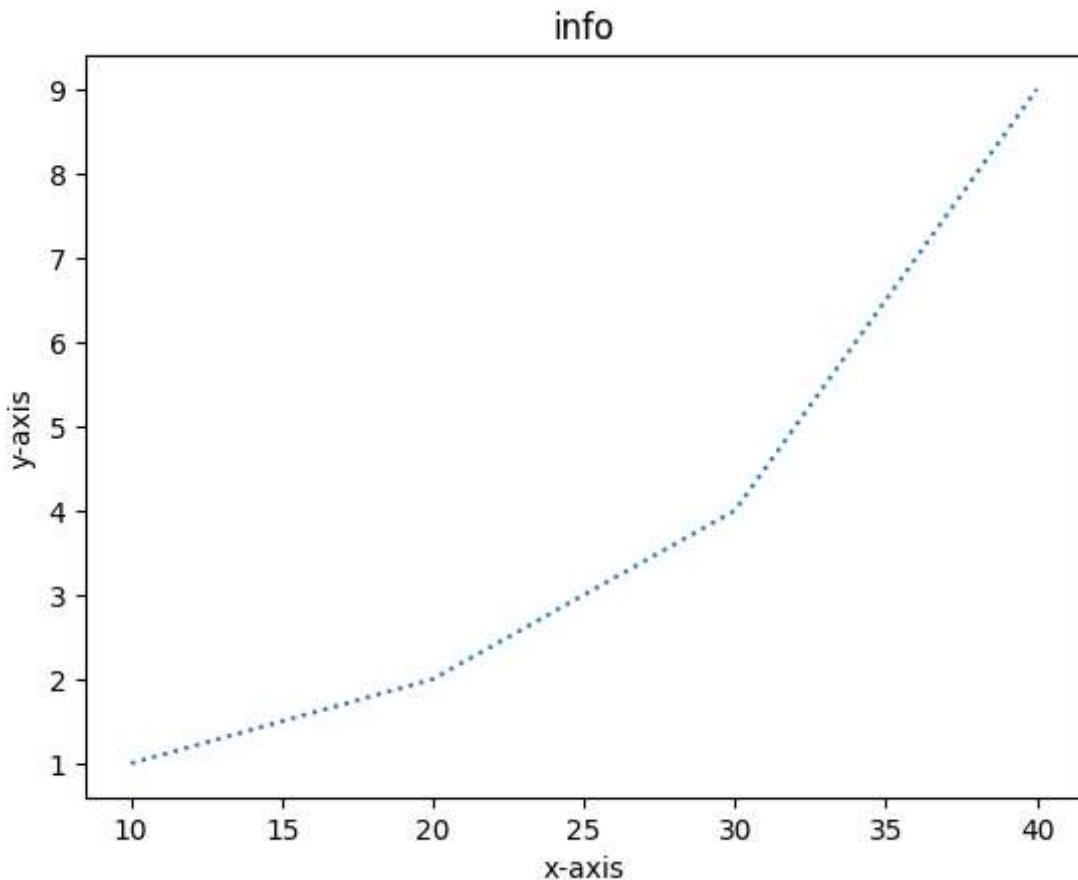
```
In [23]: plt.plot(x,y,marker="*",color="r")
```

```
Out[23]: <matplotlib.lines.Line2D at 0x1fcfa6320af0>
```

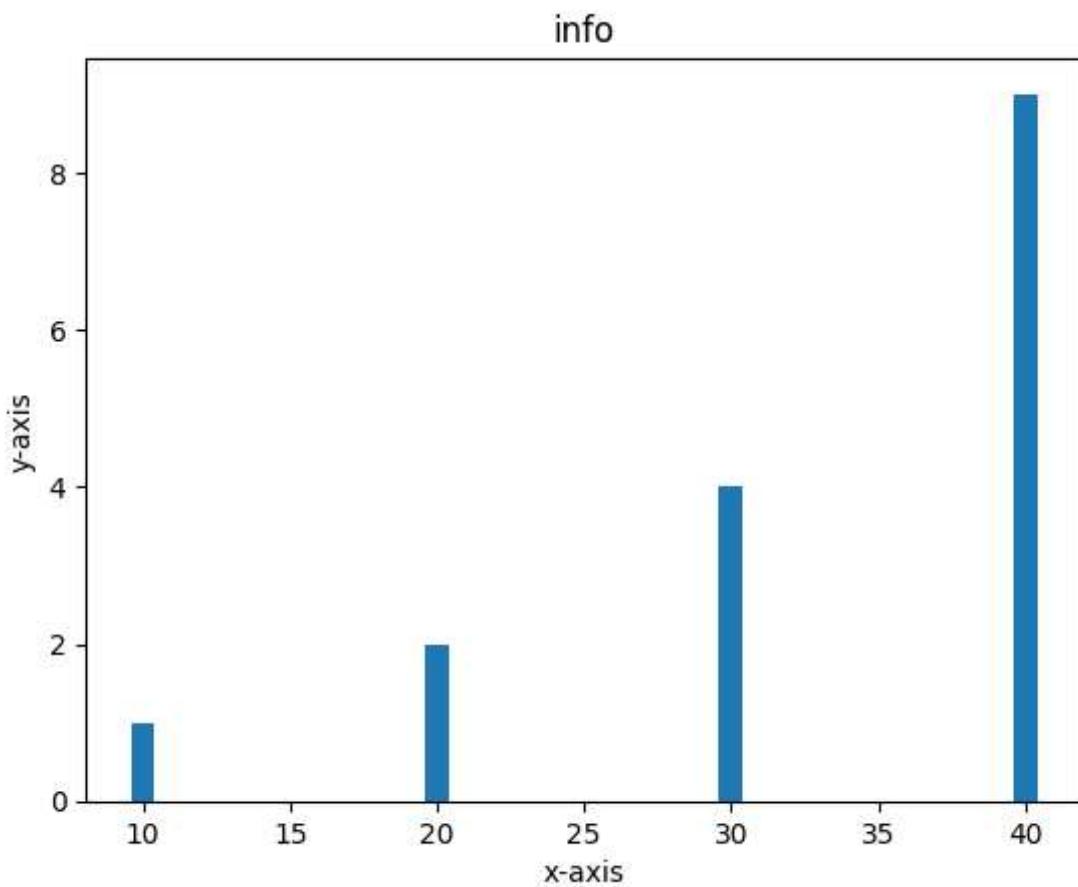


```
In [25]: x=[10,20,30,40]
y=[1,2,4,9]
plt.plot(x,y,linestyle="dotted")
plt.title("info")
plt.xlabel("x-axis")
```

```
plt.ylabel("y-axis")
plt.show()
```

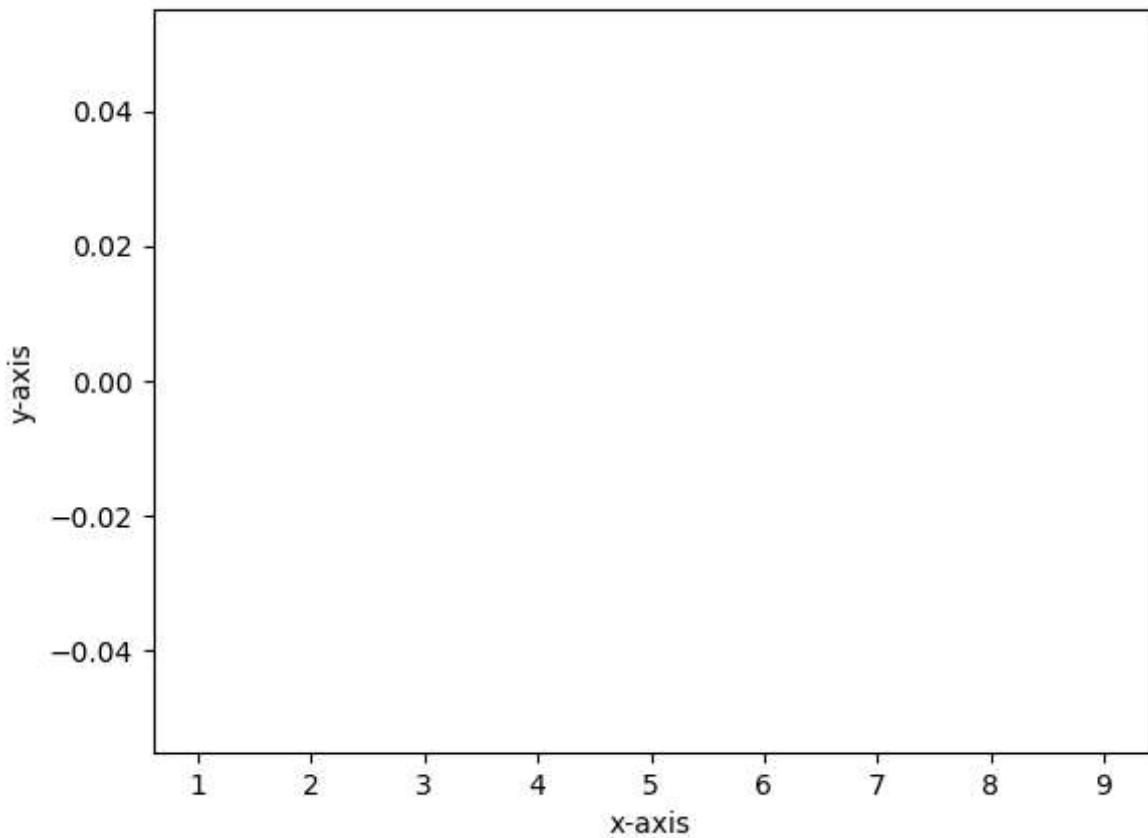


```
In [27]: x=[10,20,30,40]
y=[1,2,4,9]
plt.bar(x,y)
plt.title("info")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```



```
In [28]: x=[10,20,30,40]
y=[1,2,4,9]
plt.hist(x,y)
plt.title("info")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```

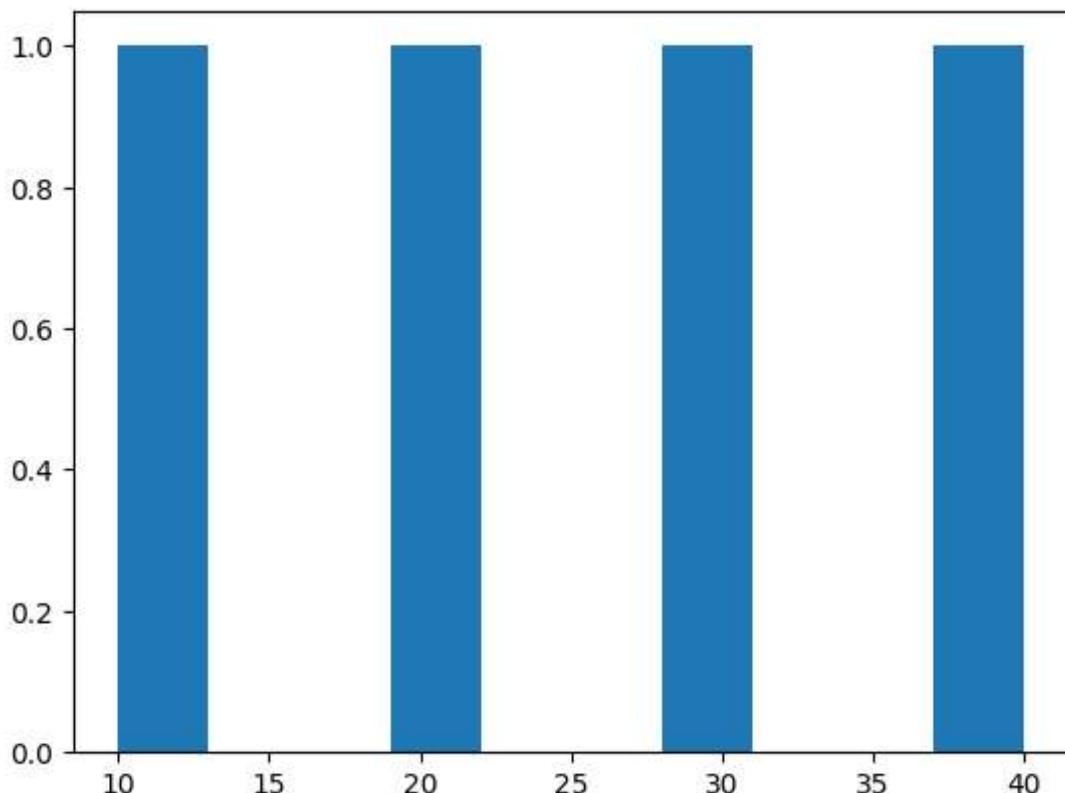
info



```
In [29]: x=[10,20,30,40]
```

```
plt.hist(x)
```

```
Out[29]: (array([1., 0., 0., 1., 0., 0., 1., 0., 0., 1.]),  
 array([10., 13., 16., 19., 22., 25., 28., 31., 34., 37., 40.]),  
 <BarContainer object of 10 artists>)
```



```
In [30]: x=[10,20,30,40]
```

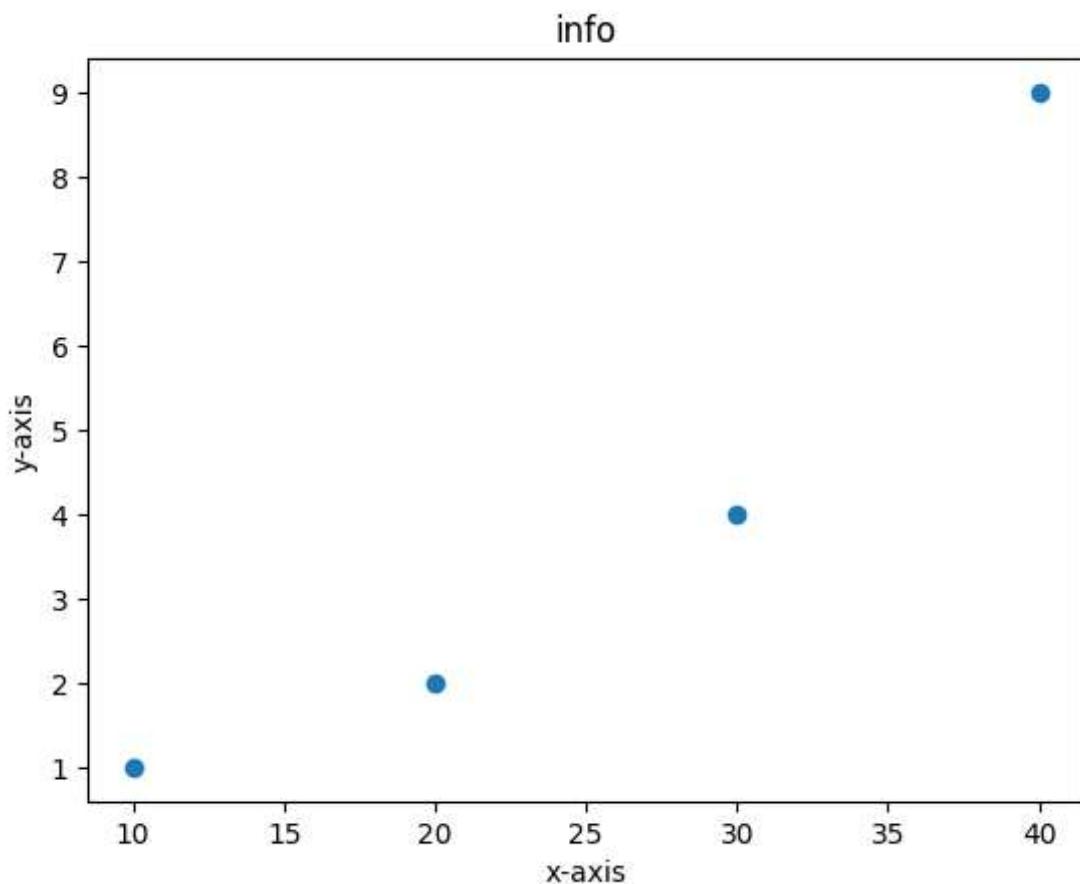
```
plt.pie(x)
```

```
Out[30]: ([<matplotlib.patches.Wedge at 0x1fca7f57520>,
<matplotlib.patches.Wedge at 0x1fca7f57430>,
<matplotlib.patches.Wedge at 0x1fca7f57df0>,
<matplotlib.patches.Wedge at 0x1fca7f902b0>],
[Text(1.0461621663333946, 0.3399186987098808, ''),
Text(0.33991867422268784, 1.0461621742897658, ''),
Text(-1.0461621902025062, 0.3399186252483017, ''),
Text(0.3399188211458418, -1.0461621265515308, '')])
```

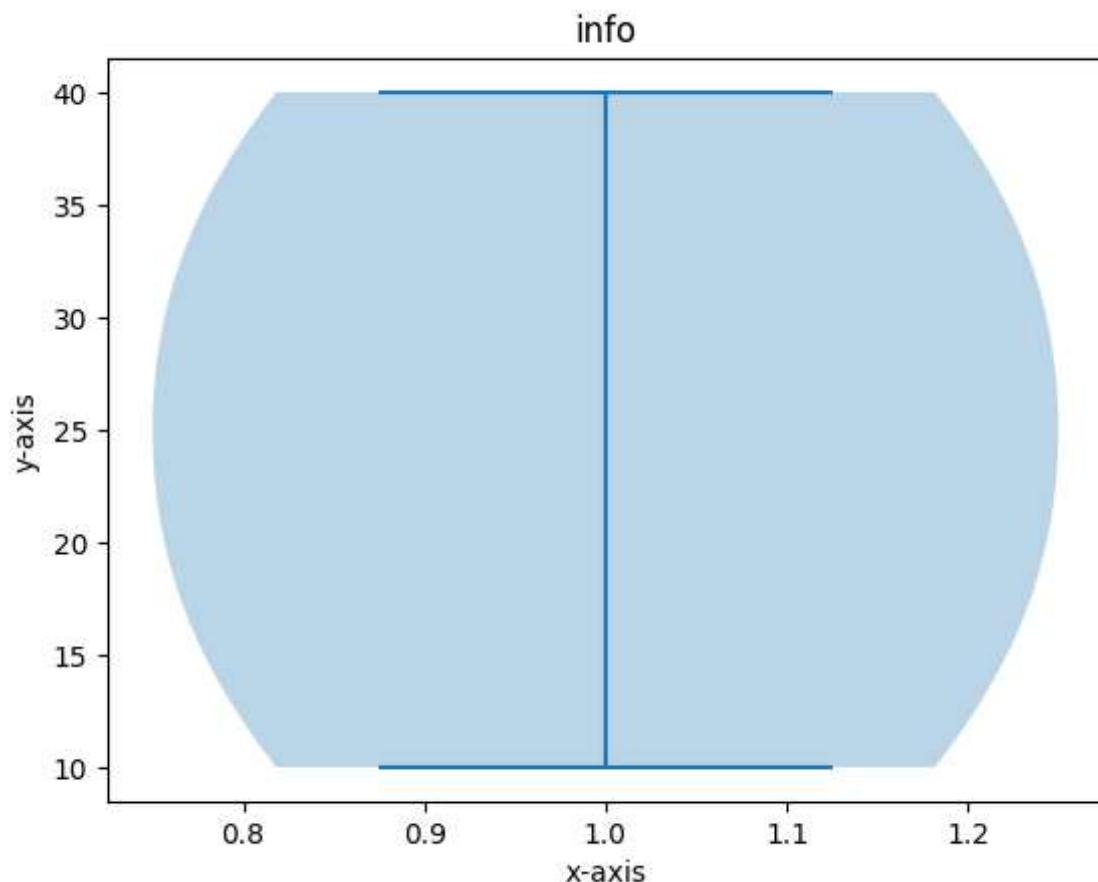


```
In [31]: x=[10,20,30,40]
```

```
y=[1,2,4,9]
plt.scatter(x,y)
plt.title("info")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```



```
In [32]: x=[10,20,30,40]
plt.violinplot(x)
plt.title("info")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```



In []: *###Seaborn###*

In [33]: `pip install seaborn`

```
Requirement already satisfied: seaborn in c:\pavan9\lib\site-packages (0.12.2)
Not e: you may need to restart the kernel to use updated packages.

Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\pavan9\lib\site-packages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in c:\pavan9\lib\site-packages (from seaborn) (2.0.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\pavan9\lib\site-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.1.0)
Requirement already satisfied: cycler>=0.10 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.40.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\pavan9\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\pavan9\lib\site-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: tzdata>=2022.1 in c:\pavan9\lib\site-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\pavan9\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
```

```
In [ ]: Seaborn is advanced Version of Matplotlib
```

```
In [34]: import seaborn as sns
```

```
In [35]: import numpy as np
```

```
In [36]: import pandas as pd
```

```
In [37]: c = pd.read_excel("C:\\\\Users\\\\synai\\\\Downloads\\\\Orders-With Nulls.xlsx")
df=pd.DataFrame(c)
```

```
In [38]: df
```

Out[38]:

	Order ID	Order Date	Order Quantity	Sales	Ship Mode	Profit	Unit Price	Customer Name	Cus Se
0	3	2010-10-13	6	261.5400	Regular Air	-213.250	38.94	Muhammed MacIntyre	Bu
1	6	2012-02-20	2	6.9300	Regular Air	-4.640	2.08	Ruben Dartt	Cor
2	32	2011-07-15	26	2808.0800	Regular Air	1054.820	107.53	Liz Pelletier	Cor
3	32	2011-07-15	24	1761.4000	Delivery Truck	-1748.560	70.89	Liz Pelletier	Cor
4	32	2011-07-15	23	160.2335	Regular Air	-85.129	7.99	Liz Pelletier	Cor
...
1002	7171	2011-02-13	17	303.1865	Regular Air	92.592	20.99	Andy Gerbode	Cor
1003	7174	2012-03-10	10	141.9200	Regular Air	12.200	13.73	Thomas Thornton	Cor
1004	7175	2010-02-07	10	748.2500	Delivery Truck	-86.990	70.98	Helen Andreada	Cor
1005	7203	2009-01-08	25	21752.0100	Regular Air	9296.348	896.99	Ruben Dartt	Cor
1006	7239	2011-06-29	50	6206.1600	Regular Air	1416.270	120.33	Craig Carroll	Bu

1007 rows × 10 columns

In [39]:

```
sns.distplot(df["Sales"])
plt.show()
```

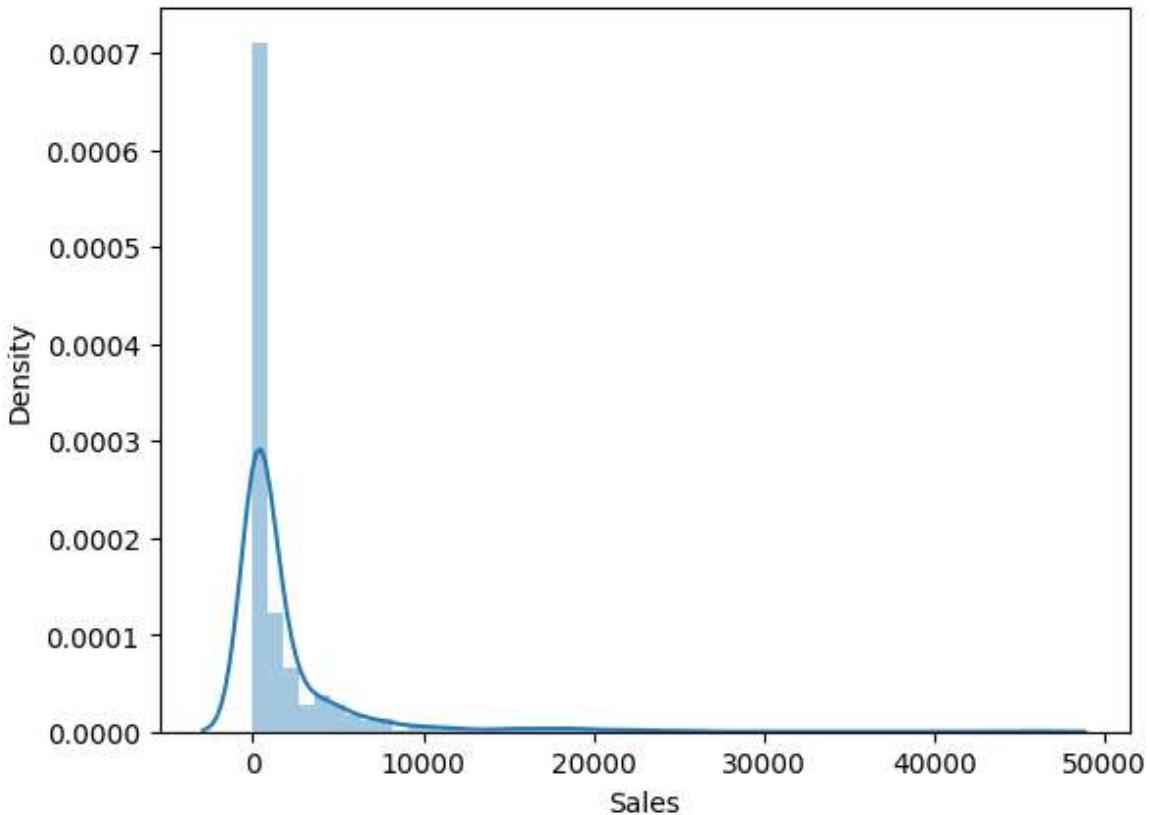
C:\Users\synai\AppData\Local\Temp\ipykernel_4176\4001346301.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df["Sales"])
```

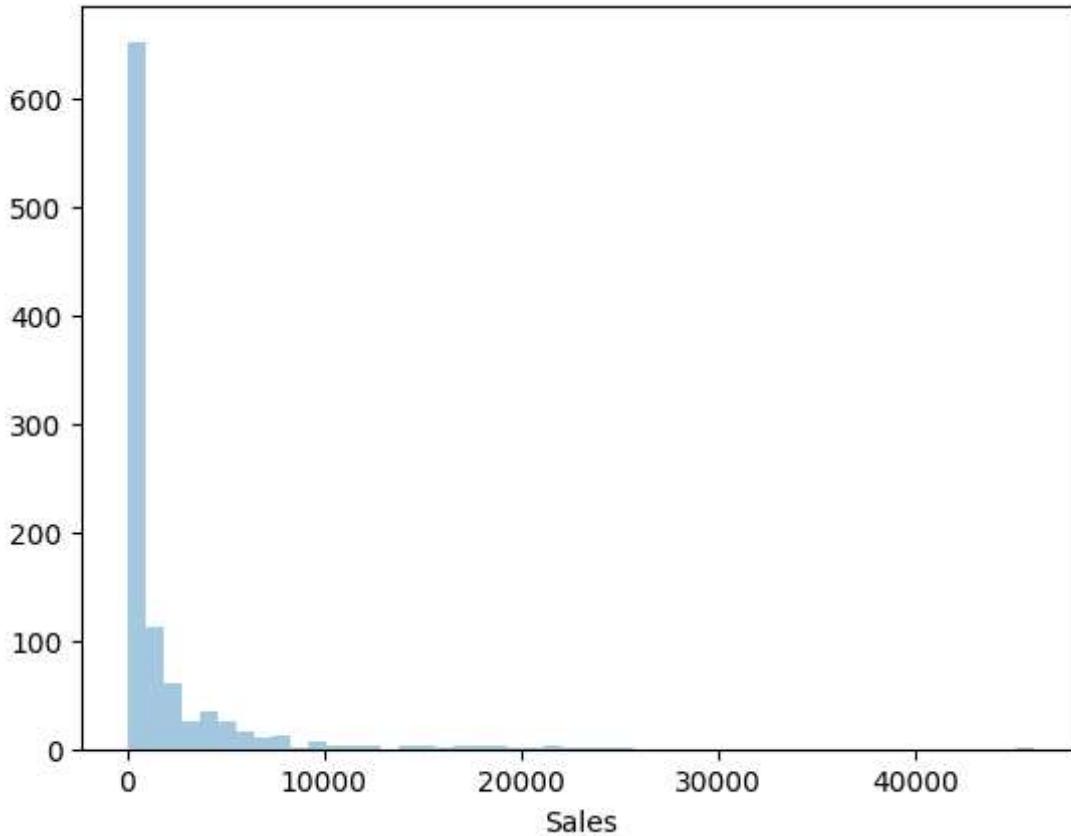


```
In [40]: sns.distplot(df["Sales"], kde=False)#####Histogram Plot#####
plt.show()
```

```
C:\Users\synai\AppData\Local\Temp\ipykernel_4176\3696167773.py:1: UserWarning:
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
sns.distplot(df["Sales"], kde=False)
```



```
In [42]: import seaborn as sns
import matplotlib.pyplot as plt

# Assuming you have a DataFrame named 'df' with columns 'Sales' and 'Order Quant

# Scatter plot
sns.scatterplot(data=df, x="Sales", y="Order Quantity")

# Display the plot
plt.show()
```

