matplotlib

July 3, 2025

```
[16]: pip install matplotlib
```

```
Requirement already satisfied: matplotlib in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (3.9.0)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (4.53.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.23 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (2.0.0)
Requirement already satisfied: packaging>=20.0 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in
c:\users\saini\appdata\local\programs\python\python311\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 23.2.1 -> 25.1.1
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
```

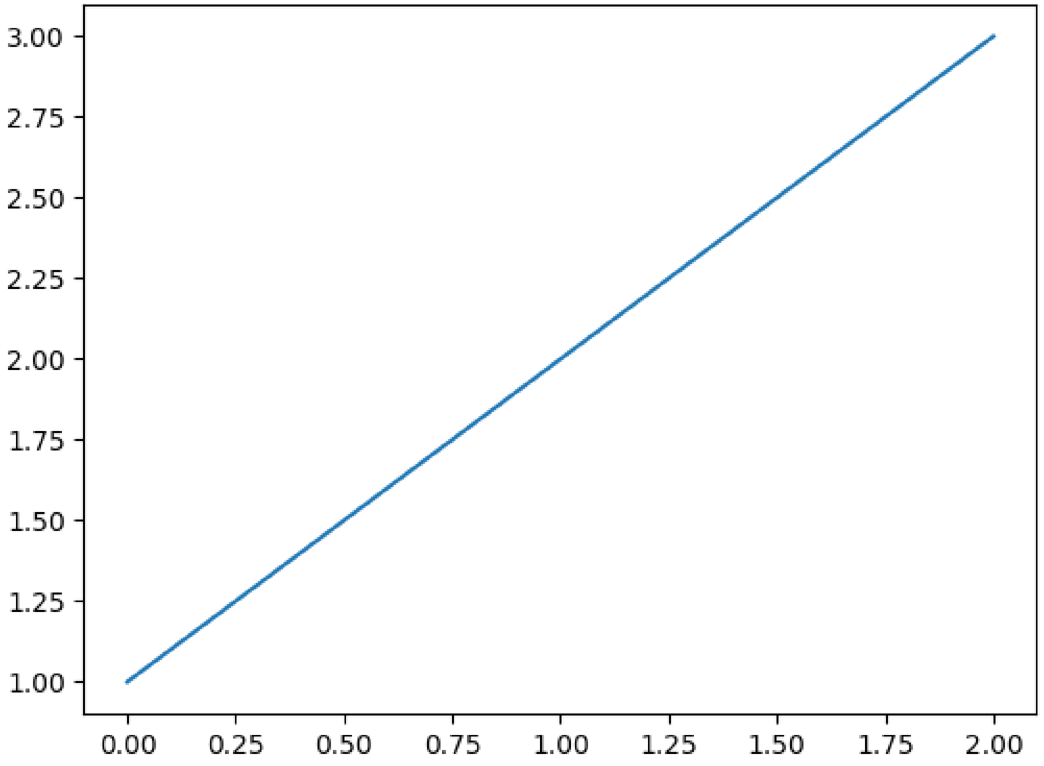
[17]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

[18]:
```
x = np.array([1,2,3])
x
```

[18]:
```
array([1, 2, 3])
```

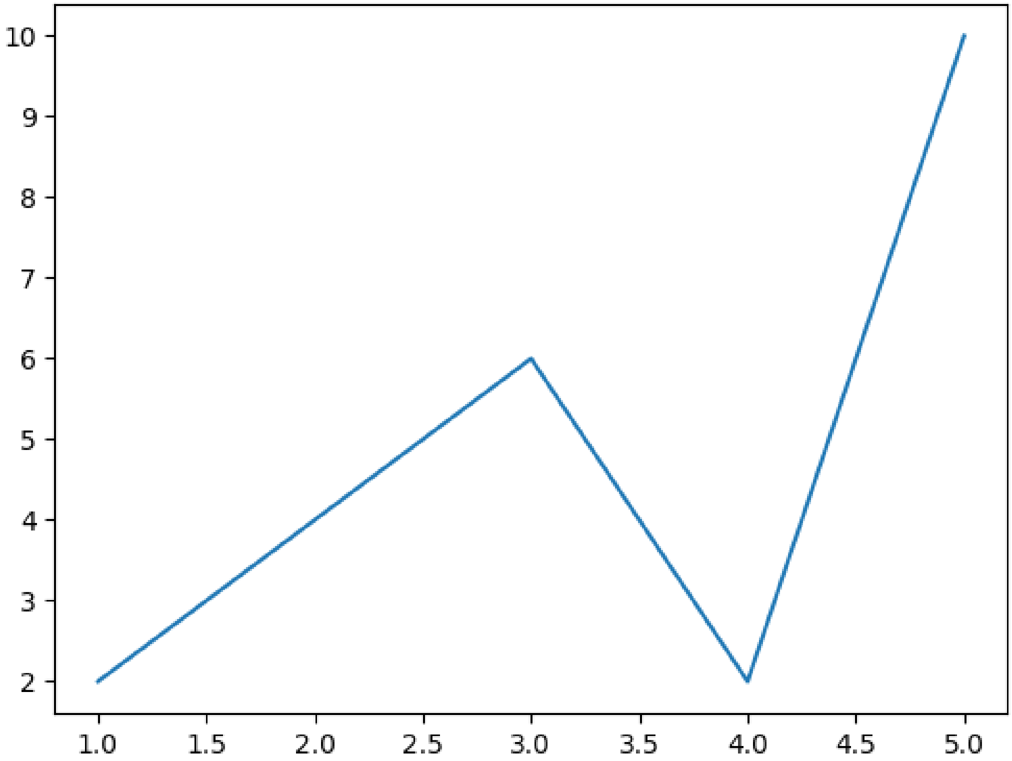
[19]:
```
# plot
# scatter
# histo
# bar
# piechart
```

[20]:
```
plt.plot(x)
plt.show()
```

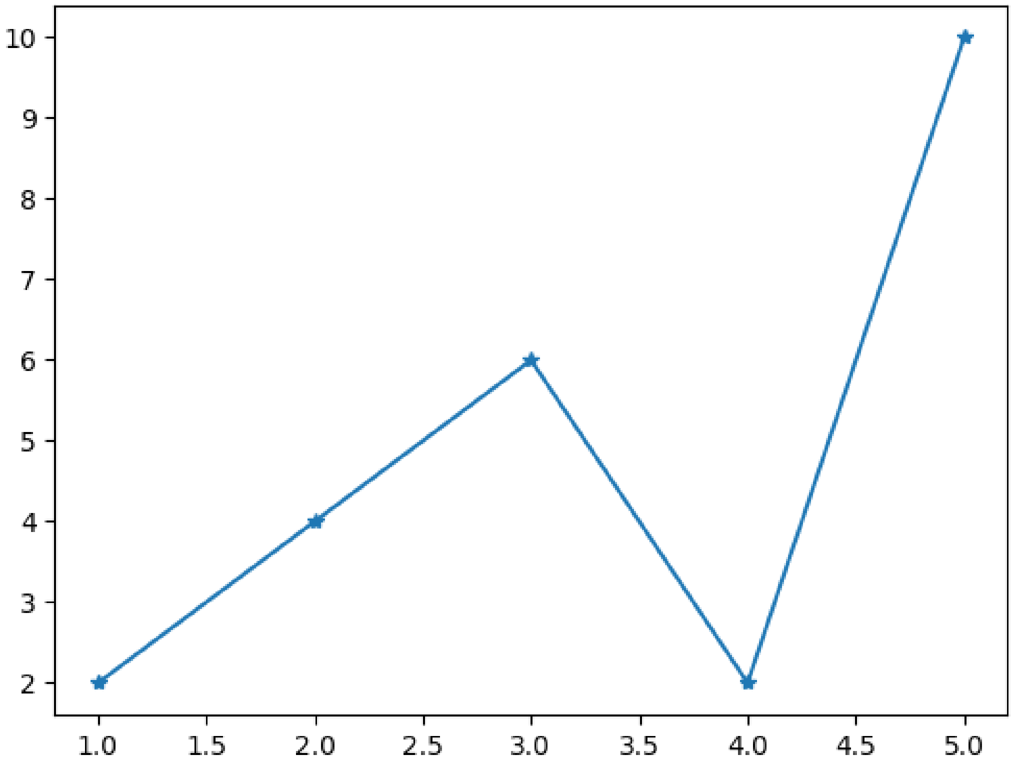


[21]:
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
```

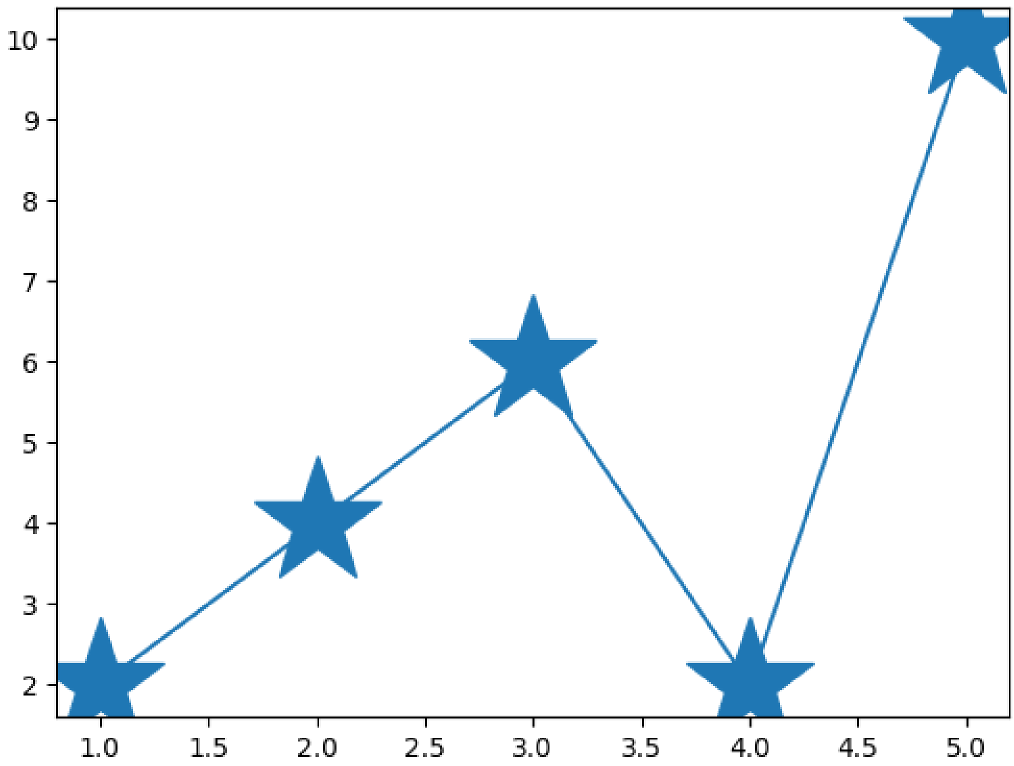
```
plt.plot(x,y)
plt.show()
```



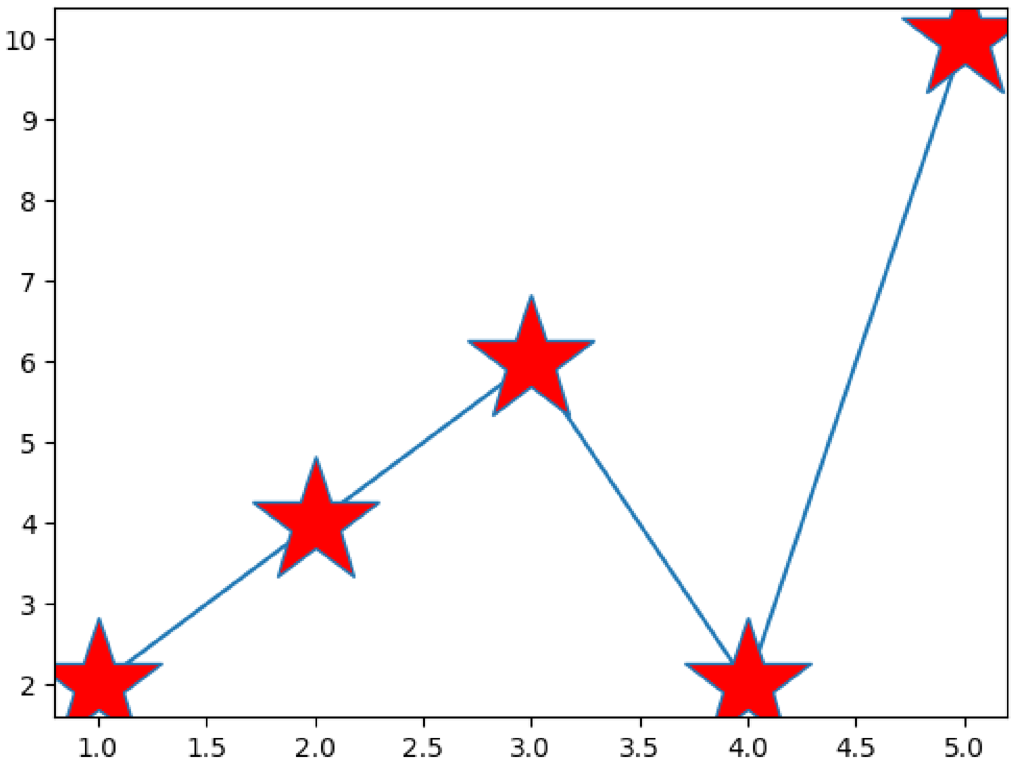
[22]:
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
plt.plot(x,y , marker='*')
plt.show()
```



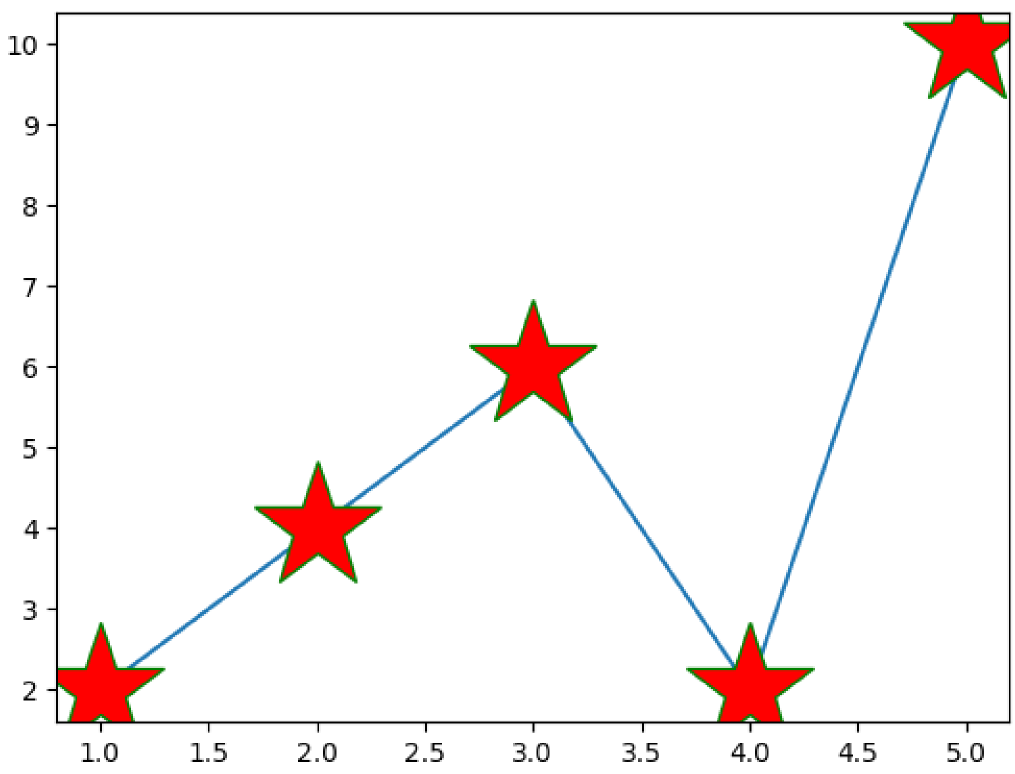
[23]:
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
plt.plot(x,y , marker='*',markersize=50)
plt.show()
```



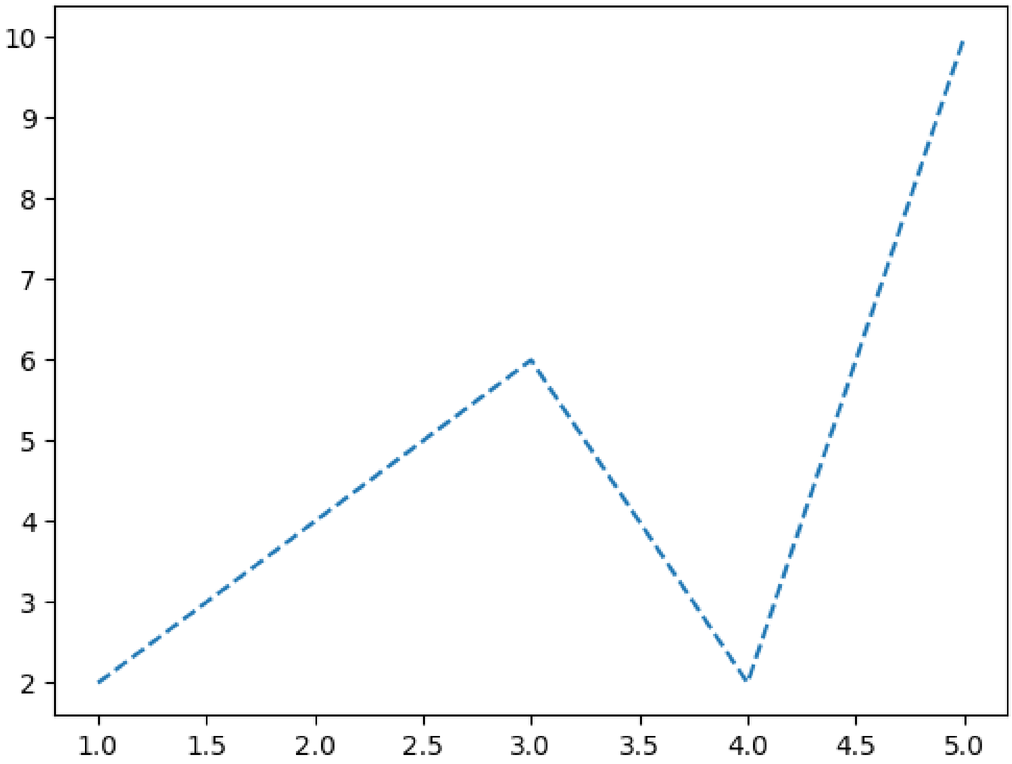
```
[24]: x = np.array([1,2,3,4,5])
      y = np.array([2,4,6,2,10])
      plt.plot(x,y , marker='*',markersize=50,markerfacecolor='red')
      plt.show()
```



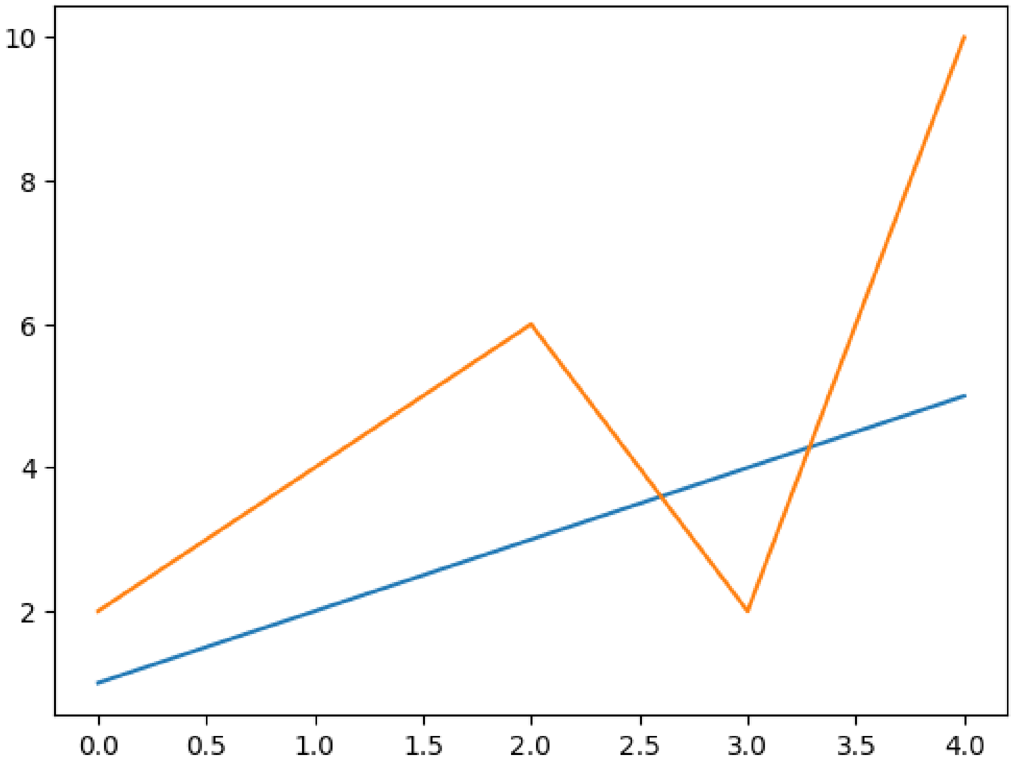
[25]:
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
plt.plot(x,y ,␣
 ↪marker='*',markersize=50,markerfacecolor='red',markeredgecolor='green')  #␣
 ↪mrs,mfc,mec
plt.show()
```



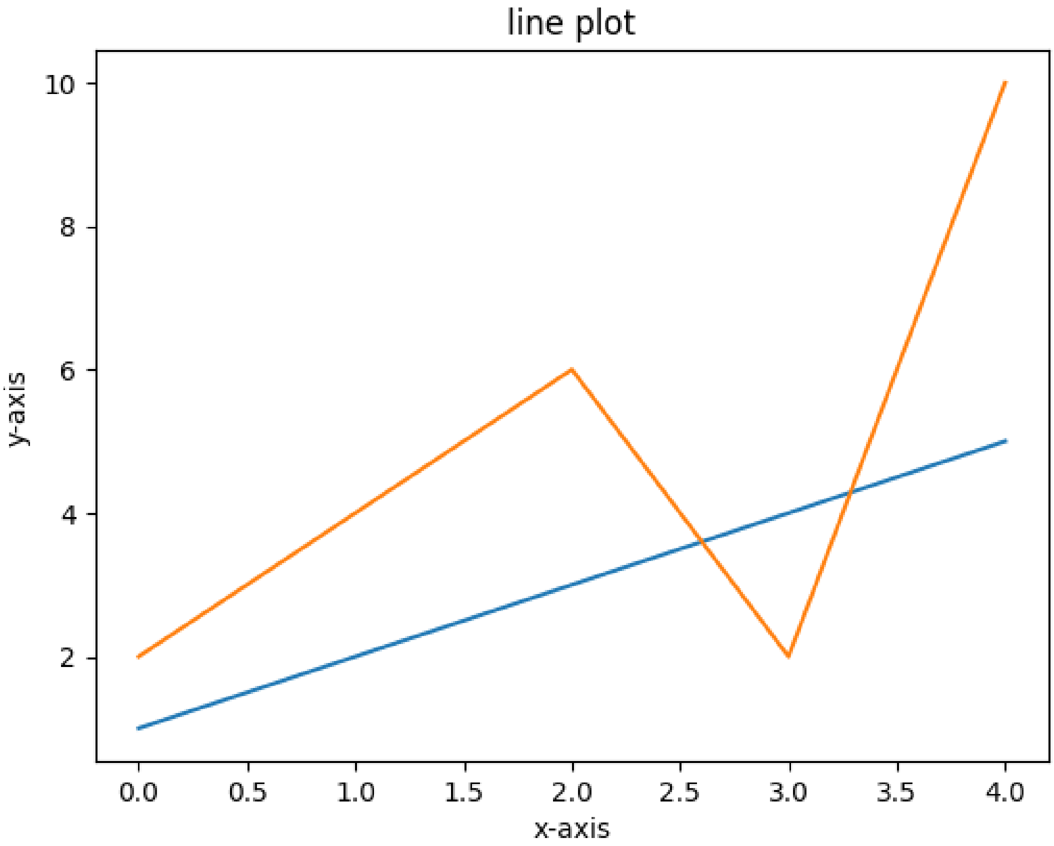
[26]:
```
x_point= np.array([1,2,3,4,5])
y_point= np.array([2,4,6,2,10])
plt.plot(x_point,y_point, linestyle='dashed')
plt.show()
```



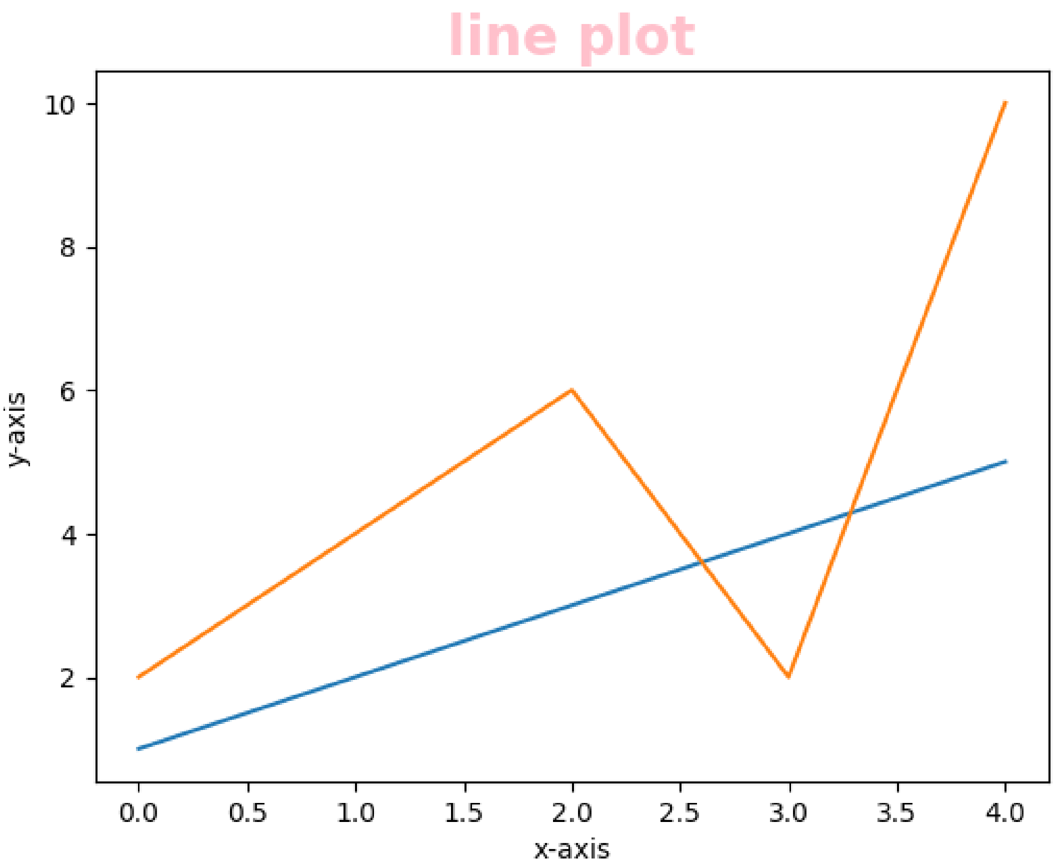
[27]:
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
plt.plot(x)
plt.plot(y)
plt.show()
```



[28]:
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
plt.title("line plot")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.plot(x)
plt.plot(y)
plt.show()
```

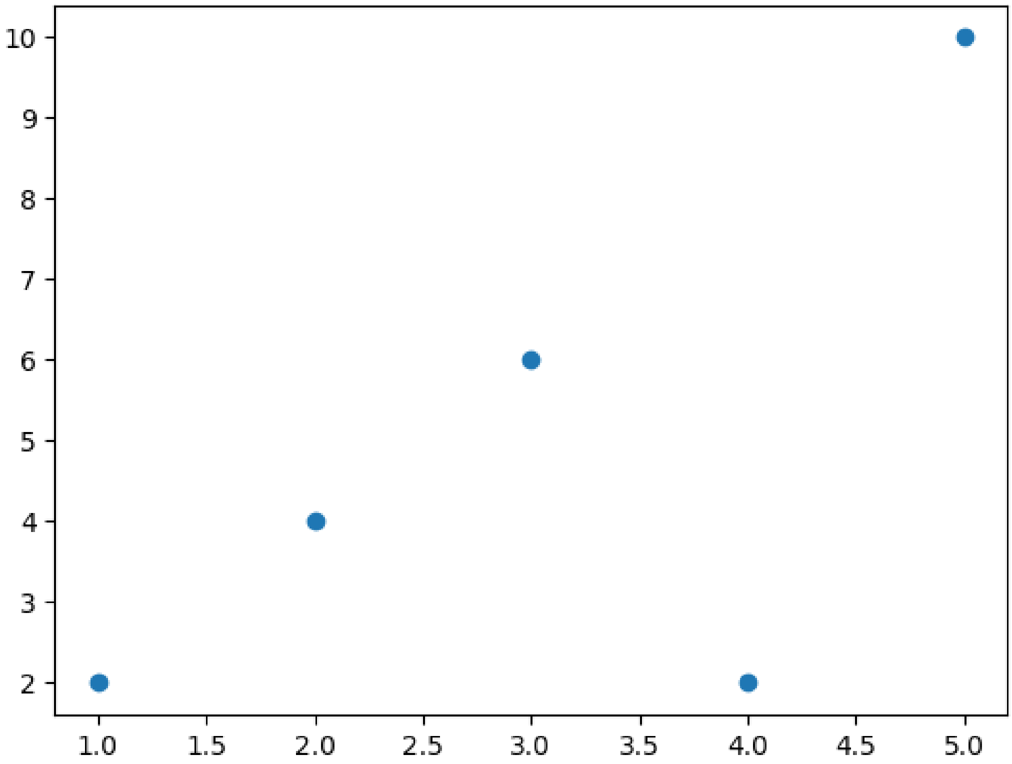


[29]:
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
plt.title("line plot",fontsize=20,fontweight='bold',color='pink')
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.plot(x)
plt.plot(y)
plt.show()
```



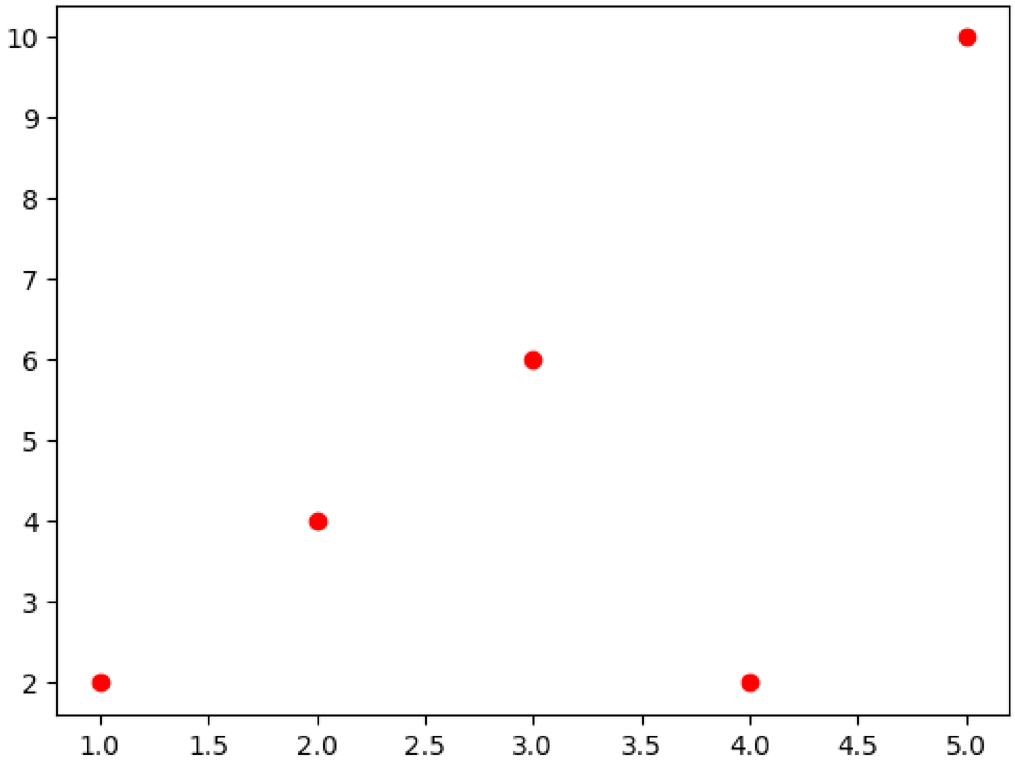
[30]:

```
# scatterplot
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
plt.scatter(x,y)
plt.show()
```



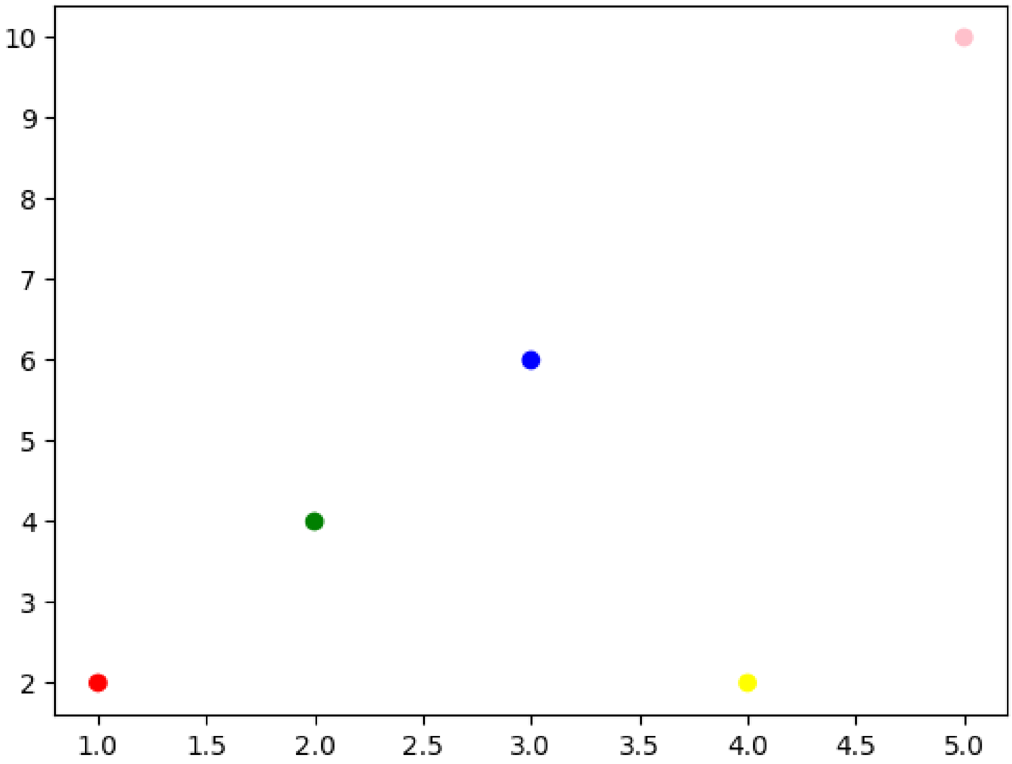
[31]:

```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
plt.scatter(x,y,color='red')
plt.show()
```

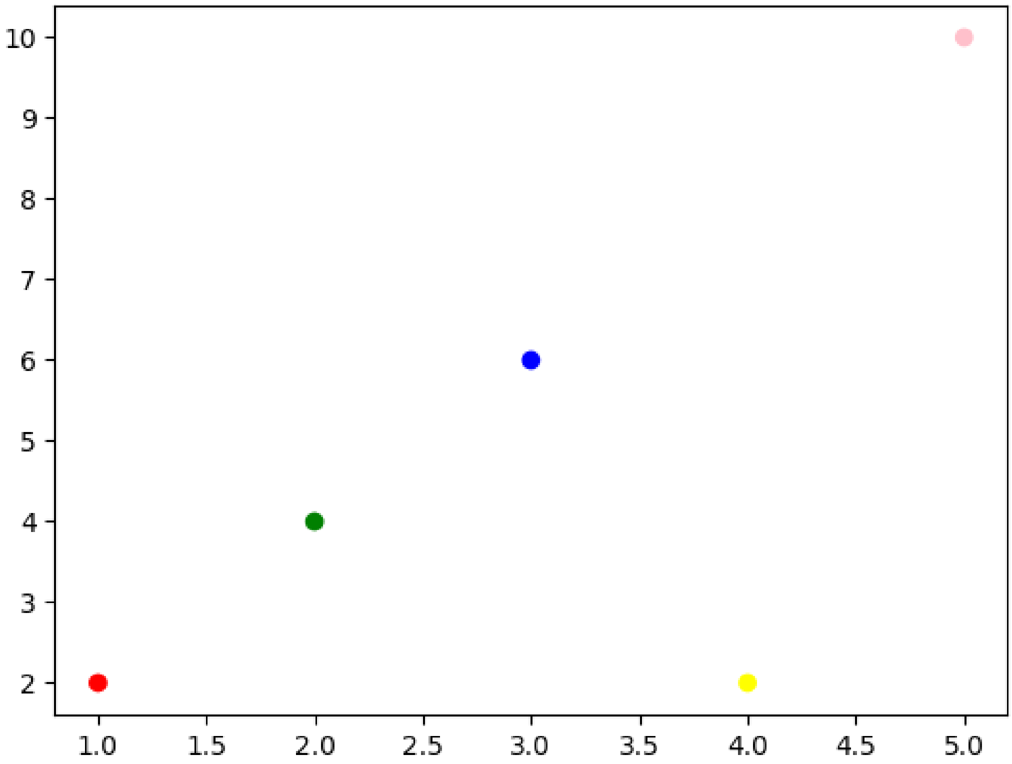


[32]:

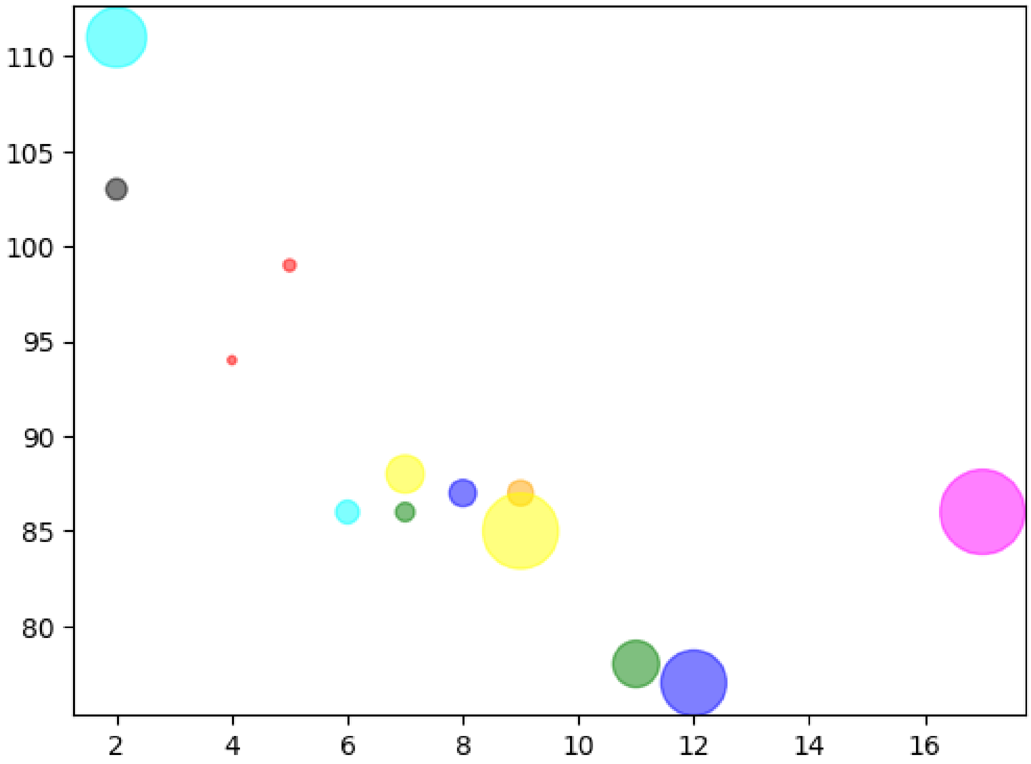
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
c=np.array(['red','green','blue','yellow','pink'])
plt.scatter(x,y,c=c)
plt.show()
```



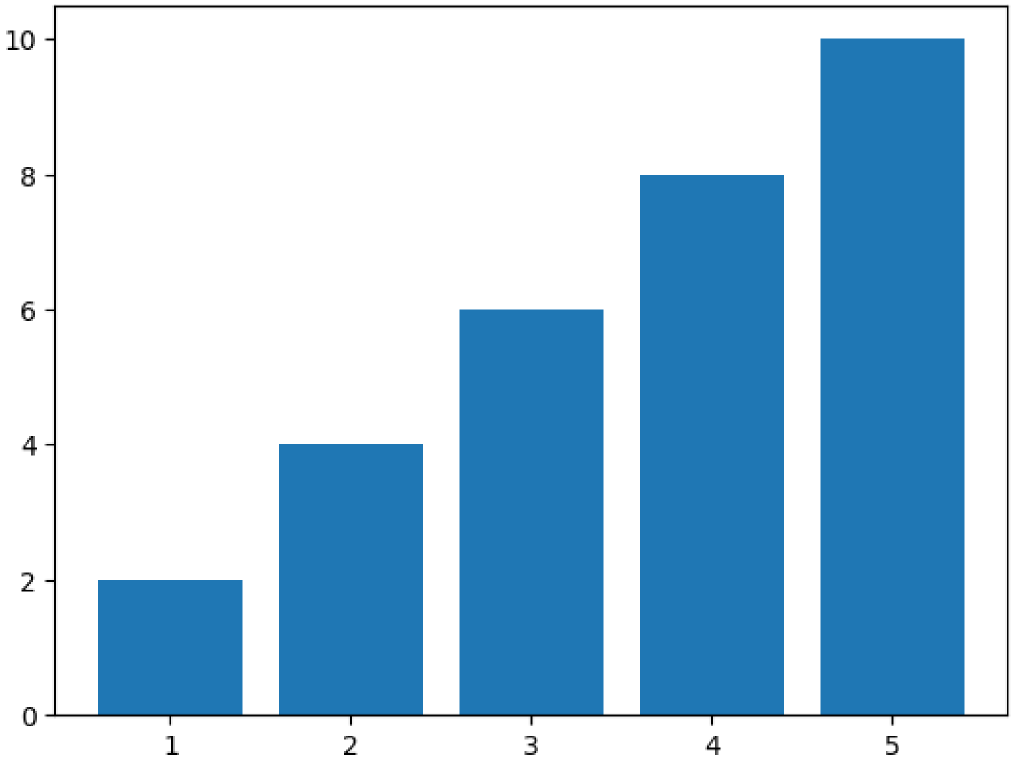
[33]:
```
x = np.array([1,2,3,4,5])
y = np.array([2,4,6,2,10])
c=np.array(['red','green','blue','yellow','pink'])
size=np.array([150,200,250,300,350,])
plt.scatter(x,y,c=c)
plt.show()
```



[34]:
```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
sizes = np.array([20,50,100,200,500,1000,60,90,10,300,600,800,75])
c = np.array(['red', 'green', 'blue', 'yellow', 'cyan', 'magenta', 'black',␣
 ↪'orange', 'red', 'green', 'blue', 'yellow', 'cyan'])
plt.scatter(x, y,alpha=0.5, s = sizes, c = c ) # camap = color map, viridis,␣
 ↪nipy_spectral, coolwarm
# plt.colorbar()
plt.show()
```

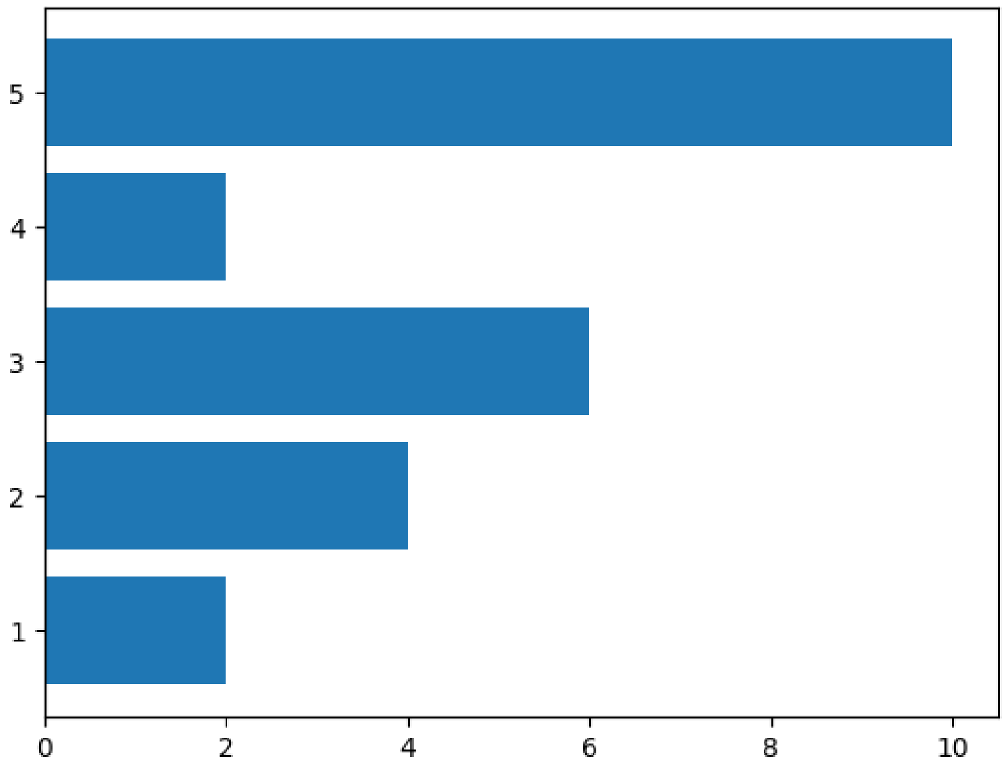


[35]:
```
x= np.array([1,2,3,4,5])
y= np.array([2,4,6,8,10])
plt.bar(x,y)
plt.show()
```

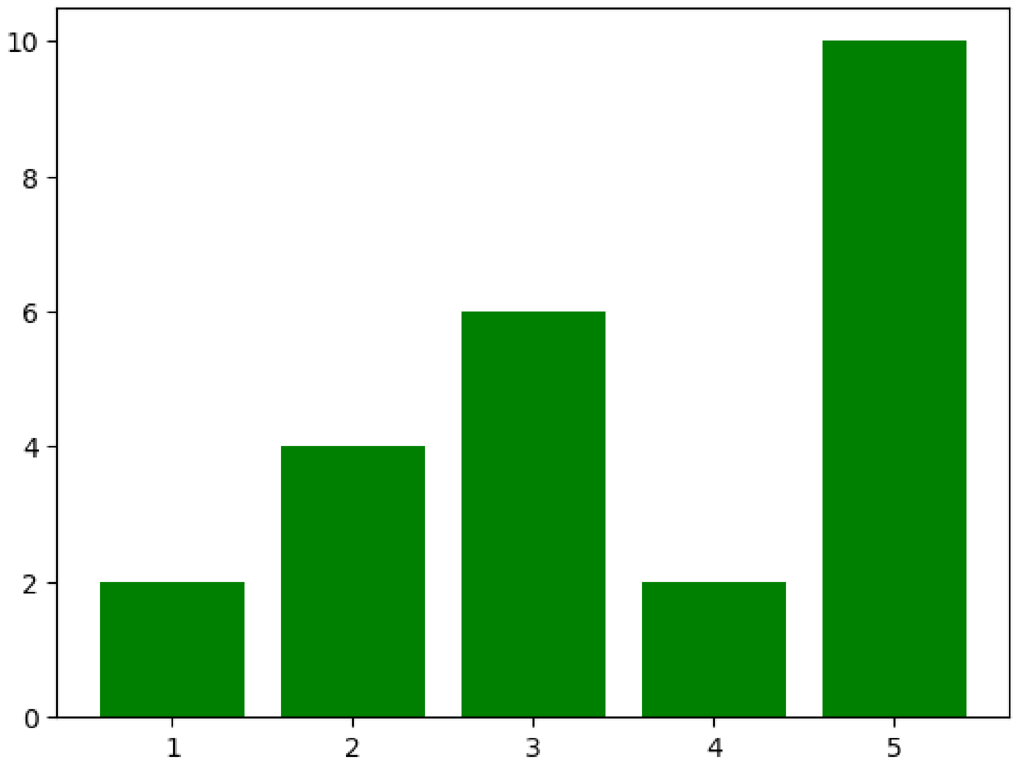


```
[45]: x=np.array([1,2,3,4,5])
      y=np.array([2,4,6,2,10])
      plt.barh(x,y)
      plt.show
```

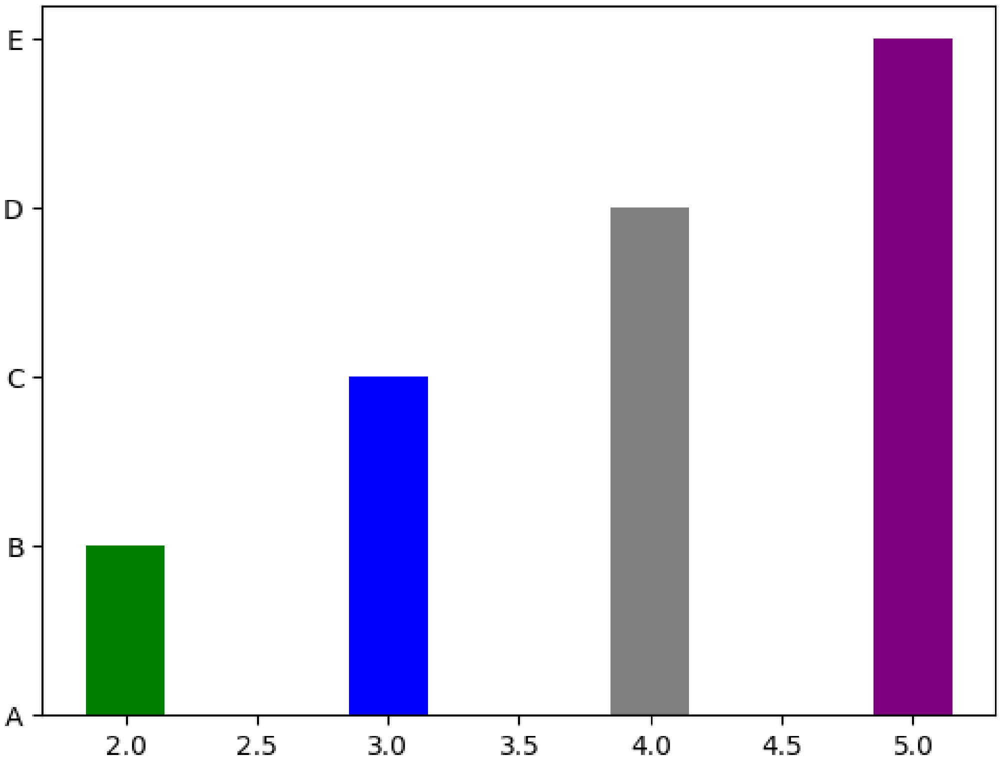
```
[45]: <function matplotlib.pyplot.show(close=None, block=None)>
```



[46]:
```
x=np.array([1,2,3,4,5])
y=np.array([2,4,6,2,10])
plt.bar(x,y,color="green")
plt.show()
```



[47]:
```
x = np.array([2, 2, 3, 4, 5])
y = np.array(['A', 'B', 'C', 'D', 'E'])
color = np.array(['red', 'green', 'blue', 'grey', 'purple'])
plt.bar(x, y, color=color, width=0.3)
plt.show()
```



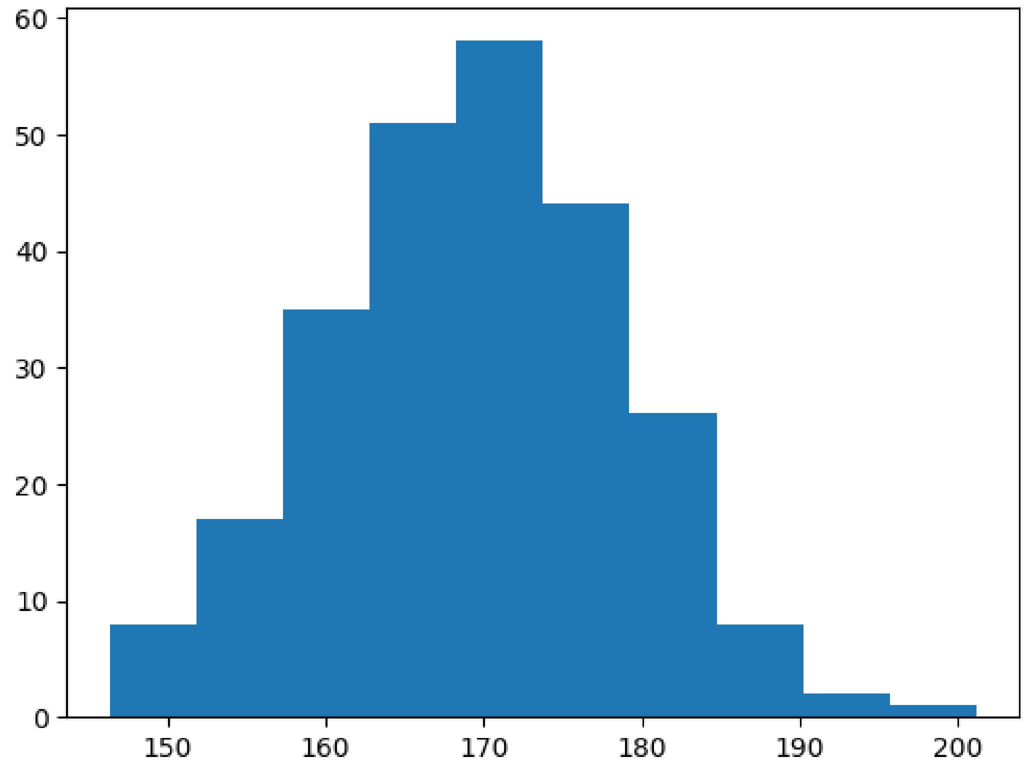
```
[ ]:
```

HISTPLOT

```
[36]: x=np.random.normal(170,10,250)
```

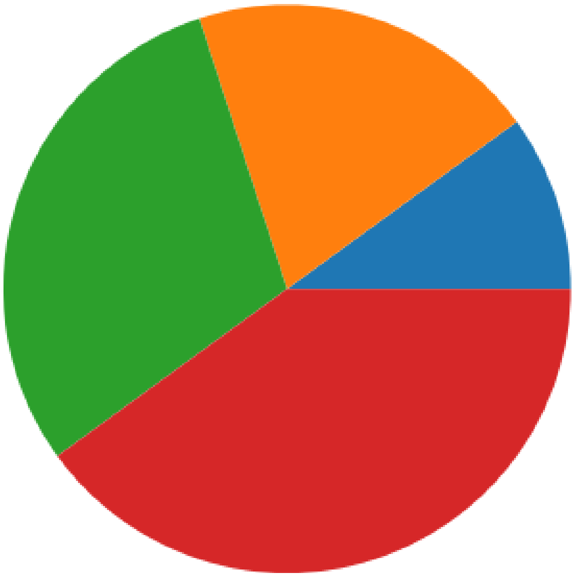
```
[37]: plt.hist(x)
```

```
[37]: (array([ 8., 17., 35., 51., 58., 44., 26.,  8.,  2.,  1.]),
       array([146.39190384, 151.86634774, 157.34079164, 162.81523554,
              168.28967944, 173.76412335, 179.23856725, 184.71301115,
              190.18745505, 195.66189895, 201.13634286]),
       <BarContainer object of 10 artists>)
```



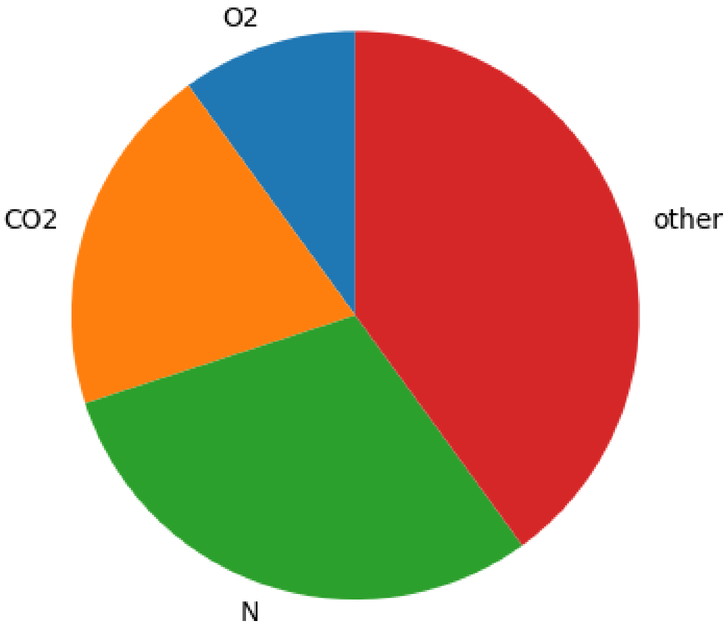
PIECHART

[38]:
```
x=np.array([1,2,3,4])
plt.pie(x)
plt.show()
```

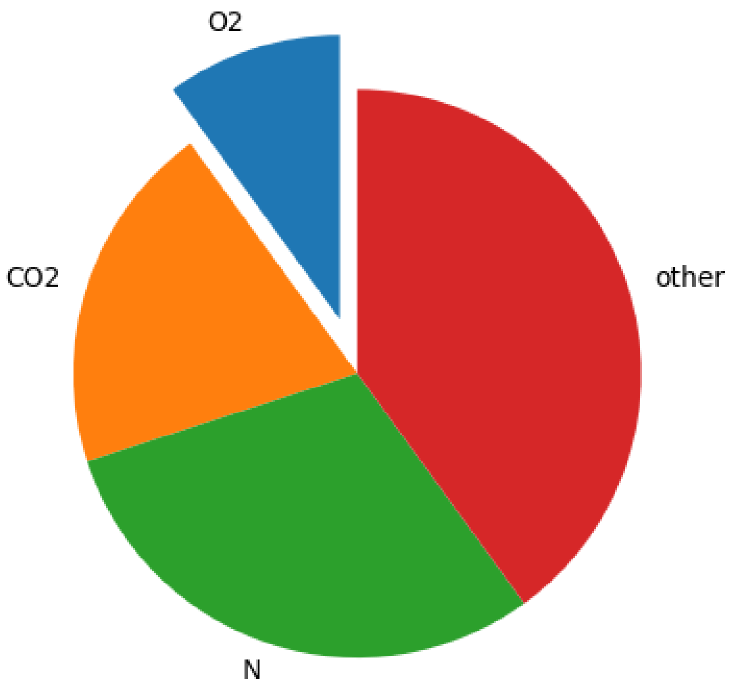


[41]:

```
x = np.array([1, 2, 3, 4])
label= np.array(["O2", "CO2", "N", "other"])
plt.pie(x, labels=label, startangle=90)
plt.show()
```

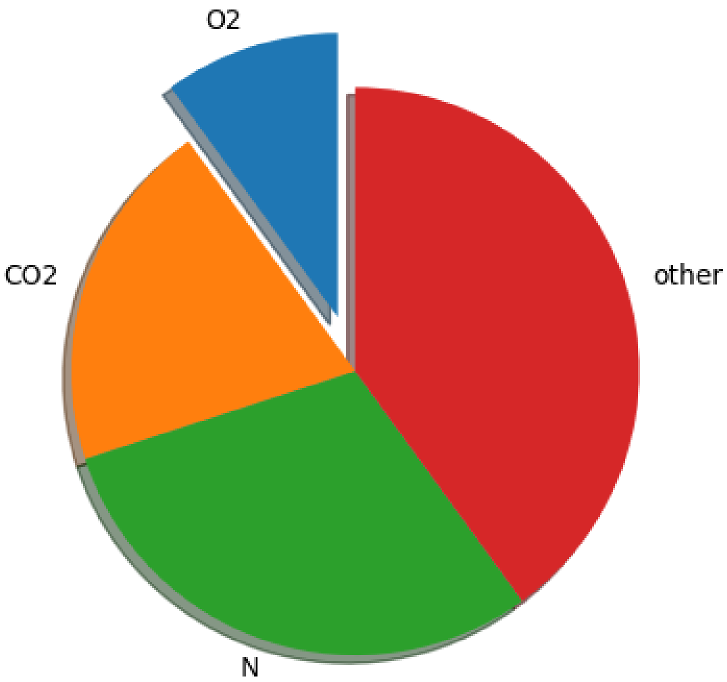


```
[42]: x = np.array([1, 2, 3, 4])
      label= np.array(["O2", "CO2", "N", "other"])
      myexplode = [0.2,0,0,0]
      plt.pie(x, labels=label, startangle=90, explode=myexplode)
      plt.show()
```

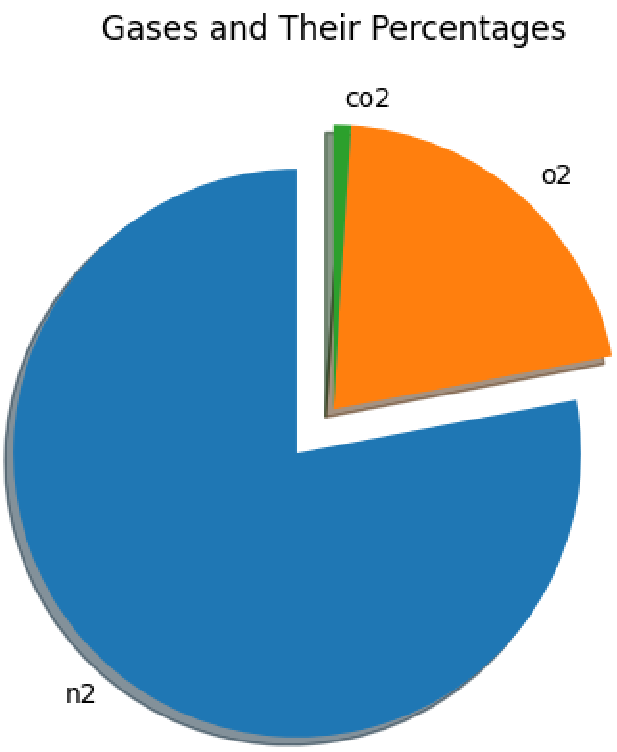


[43]:

```
x = np.array([1, 2, 3, 4])
label= np.array(["O2", "CO2", "N", "other"])
myexplode = [0.2,0,0,0]
plt.pie(x, labels=label, startangle=90, explode=myexplode, shadow=True)
plt.show()
```

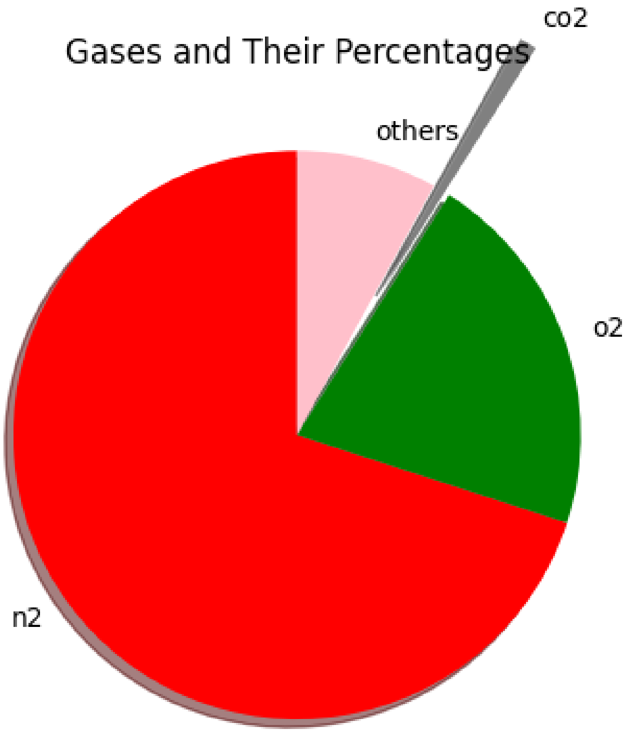


[48]:
```
gases = ['n2', 'o2', 'co2']
percentages = [78, 21, 1]
myexplode=[0.2,0,0]
plt.pie(percentages, labels=gases, startangle=90, explode=myexplode,␣
 ↪shadow=True)
plt.title("Gases and Their Percentages")
plt.show()
```

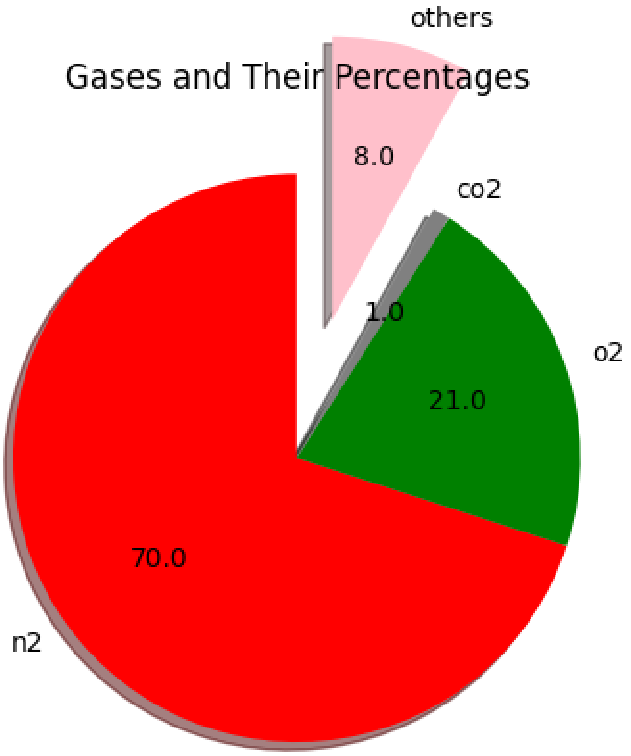


[49]:

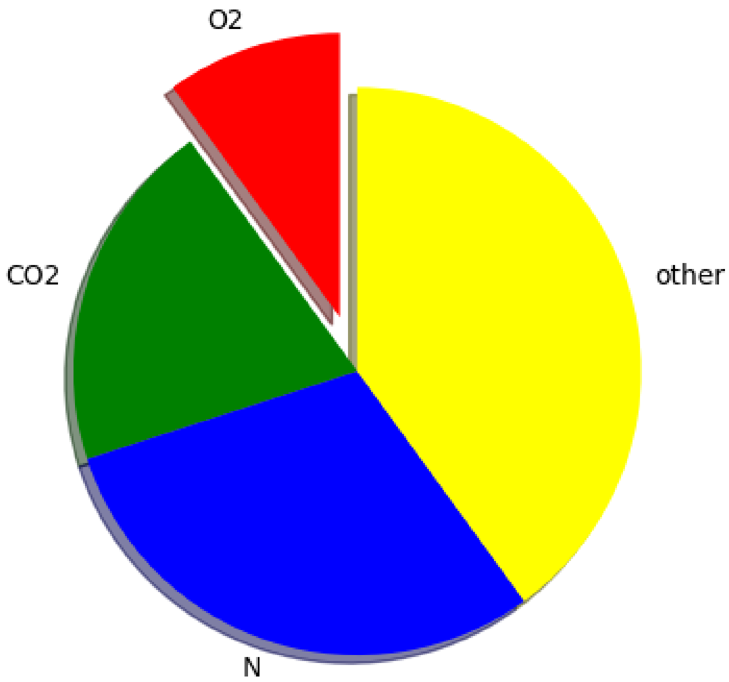
```
gases = ['n2', 'o2', 'co2','others']
percentages = [70, 21, 1,8]
myexplode=[0,0,0.6,0]
color=['red', 'green', 'grey','pink']
plt.pie(percentages, labels=gases, startangle=90, explode=myexplode,␣
  ↪shadow=True, colors=color)
plt.title("Gases and Their Percentages")
plt.show()
```



```
[50]: gases = ['n2', 'o2', 'co2','others']
      percentages = [70, 21, 1,8]
      myexplode=[0,0,0,0.5]
      color=['red', 'green', 'grey','pink']
      # autopct displays the percentage value on each slice of the pie chart
      plt.pie(percentages, labels=gases, startangle=90, explode=myexplode,␣
        ↪shadow=True, colors=color, autopct='%1.1f')
      plt.title("Gases and Their Percentages")
      plt.show()
```

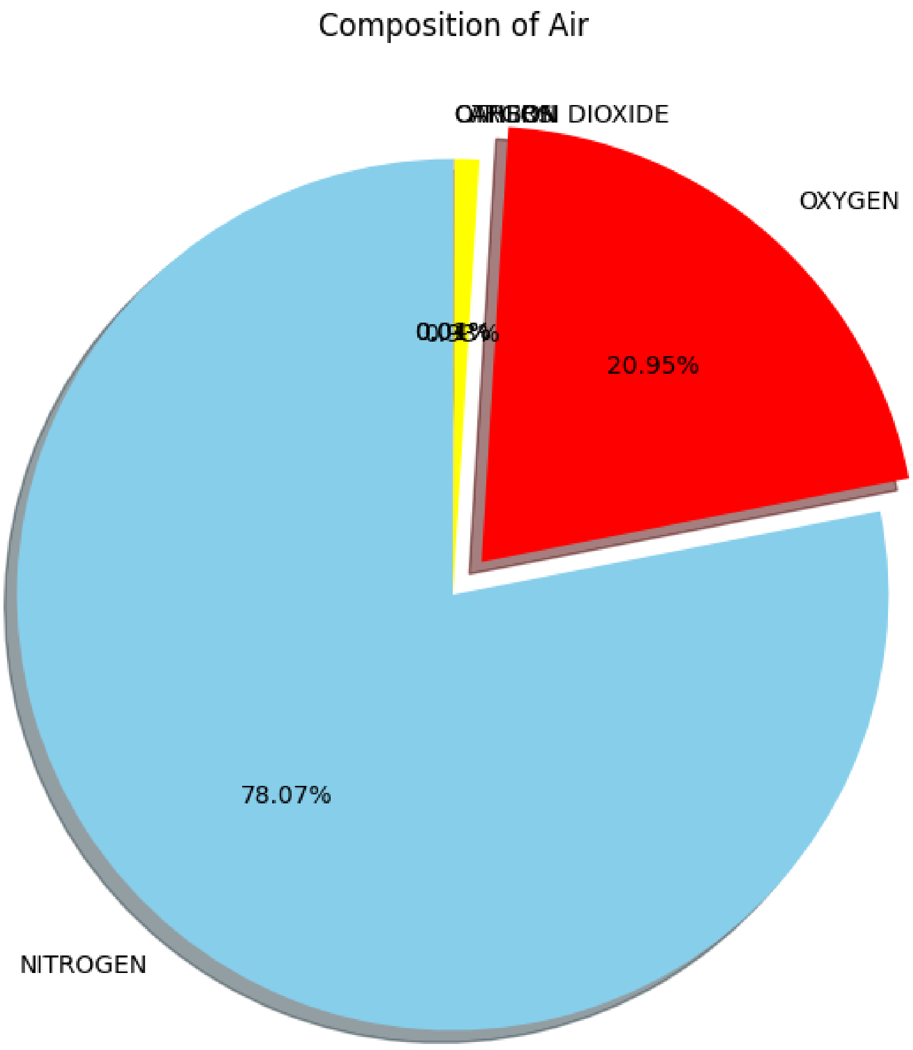


[44]:
```
x = np.array([1, 2, 3, 4])
label= np.array(["O2", "CO2", "N", "other"])
myexplode = [0.2,0,0,0]
color = ["red","green","blue","yellow"]
plt.pie(x, labels=label, startangle=90, explode=myexplode, shadow=True,␣
  ↪colors=color)
plt.show()
```



```
[51]: gases=["NITROGEN","OXYGEN","ARGON","CARBON DIOXIDE","OTHERS"]
      percentages=[78.09,20.95,0.93,0.04,0.01]
      colors=['skyblue','red','yellow','silver','gray']
      explode = [0, 0.1, 0, 0, 0]

      plt.figure(figsize=(8, 8))
      plt.pie(
          percentages,
          labels=gases,
          colors=colors,
          explode=explode,
          autopct='%1.2f%%',
          startangle=90,
          shadow=True
      )
      plt.title("Composition of Air")
      plt.show()
```


Composition of Air
ARGON
CARBON DIOXIDE
OXYGEN
20.95%
78.07%
NITROGEN

```
[ ]:
```