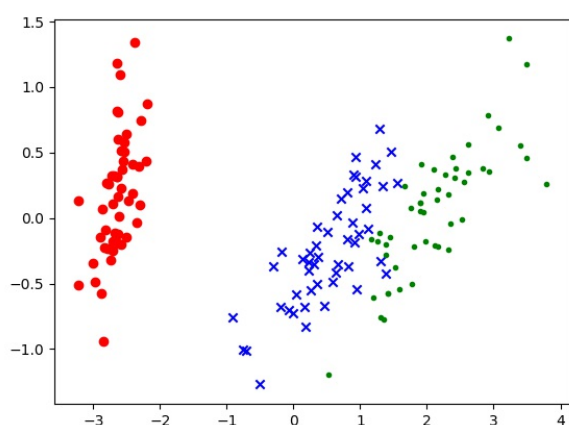


1.将Iris数据集通过PCA降维（两维）

```
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.datasets import load_iris
data=load_iris()
y=data.target
x=data.data
pca=PCA(n_components=2)
reduced_x=pca.fit_transform(x)

red_x,red_y=[],[]
blue_x,blue_y=[],[]
green_x,green_y=[],[]
for i in range(len(reduced_x)):
    if y[i]==0:
        red_x.append(reduced_x[i][0])
        red_y.append(reduced_x[i][1])
    elif y[i]==1:
        blue_x.append(reduced_x[i][0])
        blue_y.append(reduced_x[i][1])
    else:
        green_x.append(reduced_x[i][0])
        green_y.append(reduced_x[i][1])
#Visual
plt.scatter(red_x,red_y,c='r',marker='o')
plt.scatter(blue_x,blue_y,c='b',marker='x')
plt.scatter(green_x,green_y,c='g',marker='.')
plt.show()
```

Screenshot :



2.将Iris数据集通过PCA降维（三维）

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.decomposition import PCA
from sklearn.datasets import load_iris
```

```

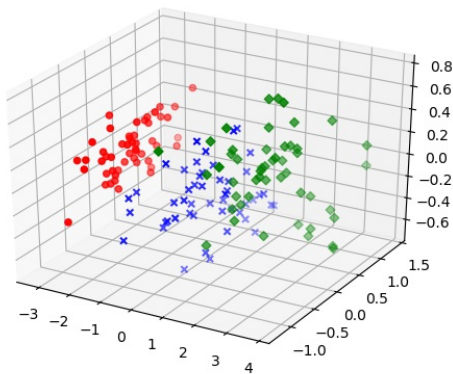
data=load_iris()
y=data.target
x=data.data
pca=PCA(n_components=3)
reduced_x=pca.fit_transform(x)

red_x,red_y,red_z=[],[],[]
blue_x,blue_y,blue_z=[],[],[]
green_x,green_y,green_z=[],[],[]
for i in range(len(reduced_x)):
    if y[i]==0:
        red_x.append(reduced_x[i][0])
        red_y.append(reduced_x[i][1])
        red_z.append(reduced_x[i][2])
    elif y[i]==1:
        blue_x.append(reduced_x[i][0])
        blue_y.append(reduced_x[i][1])
        blue_z.append(reduced_x[i][2])
    else:
        green_x.append(reduced_x[i][0])
        green_y.append(reduced_x[i][1])
        green_z.append(reduced_x[i][2])

#Visual
fig=plt.figure()
ax=fig.gca(projection='3d')
ax.scatter(red_x,red_y,red_z,c='r',marker='o')
ax.scatter(blue_x,blue_y,blue_z,c='b',marker='x')
ax.scatter(green_x,green_y,green_z,c='g',marker='D')
plt.show()

```

Screenshot :



3.折线

```

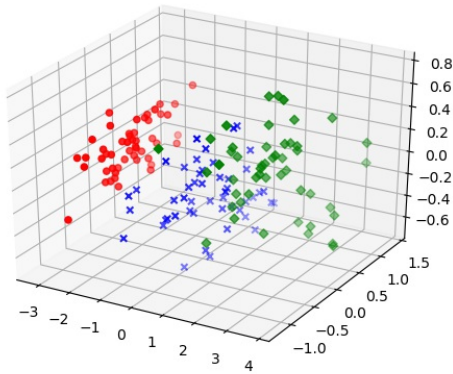
import matplotlib.pyplot as plt
def main():
    x_values = [x for x in range(1,20)]
    y_values = [x ** 2 for x in range(1,20)]
    plt.title('square number')
    plt.xlabel('value', fontsize=23)
    plt.ylabel('square', fontsize=23)
    plt.tick_params(axis='both',labelsize=23)
    plt.plot(x_values,y_values)

```

```
plt.show()

if __name__ == '__main__':
    main()
```

Screenshot :

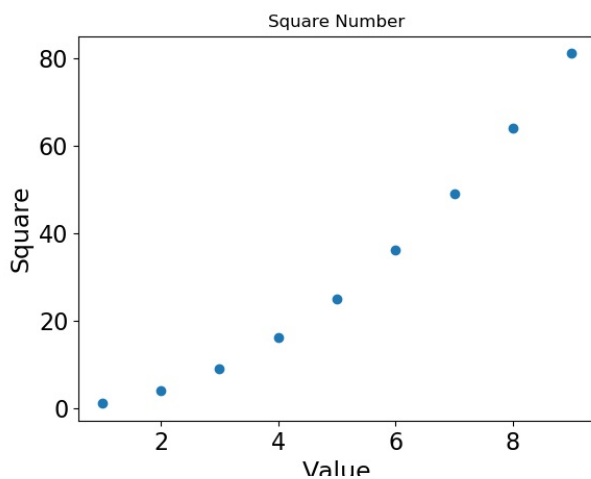


4.散点Scatter

```
import matplotlib.pyplot as plt
def main():
    x_values = [x for x in range(1,10)]
    y_values = [x ** 2 for x in range(1,10)]
    plt.title('Square Number')
    plt.xlabel('Value', fontsize=17)
    plt.ylabel('Square', fontsize=17)
    plt.tick_params(axis='both',labelsize=16)
    #plt.plot(x_values,y_values)
    plt.scatter(x_values,y_values)
    plt.show()

if __name__ == '__main__':
    main()
```

Screenshot :

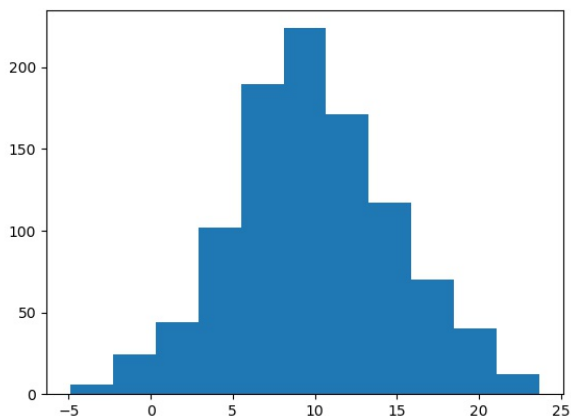


5.直方

```
import matplotlib.pyplot as plt
import numpy as np
def main():
    data=np.random.normal(10.0,5.0,1000)
    plt.hist(data,11)
    plt.show()

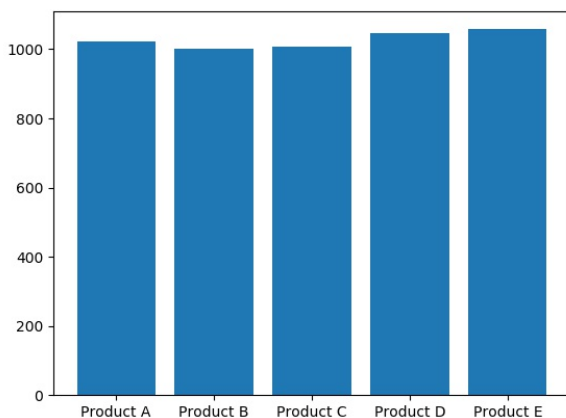
if __name__ == '__main__':
    main()
```

Screenshot :



```
import matplotlib.pyplot as plt
import pandas as pd
###print(np.random.randint(0,40,5))
from numpy.random import randint
###3from random import randint
x=['Product A', 'Product B', 'Product C', 'Product D', 'Product E']
y=[randint(1000,1100),randint(1000,1100),randint(1000,1100),randint(1000,1100),np.random.randint(1000,1100)]
plt.bar(x,y)
plt.show()
```

Screenshot :



6.legend

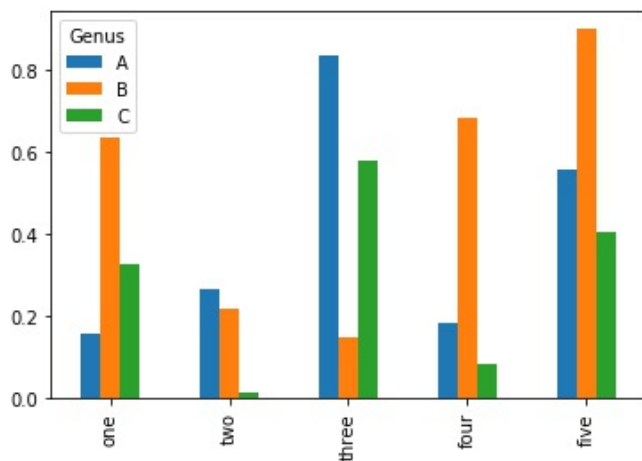
```
import matplotlib.pyplot as plt
import numpy as np

def main():
    data = np.random.normal(10.0,5.0,1000)
    plt.hist(data,10)
    plt.show()

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from numpy.random import randint

df=pd.DataFrame(np.random.rand(5,3),index=
['one', 'two', 'three', 'four', 'five'],columns=pd.Index(['A', 'B', 'C'],name='Genus'))
df.plot.bar()
```

Screenshot :



7.柱图

```
import numpy as np
import matplotlib.pyplot as plt
import mpl_toolkits.mplot3d

x= np.random.randint(0,400,10)
y= np.random.randint(0,400,10)
z= 800*abs(np.sin(x+y))

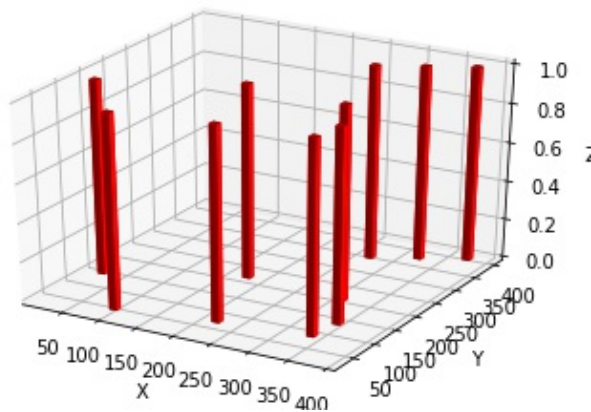
ax=plt.subplot(projection='3d')

ax.bar3d(x,y,np.zeros_like(z),dx=10,dy=10,dz=1,color='red')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

plt.show()
```

Screenshot :

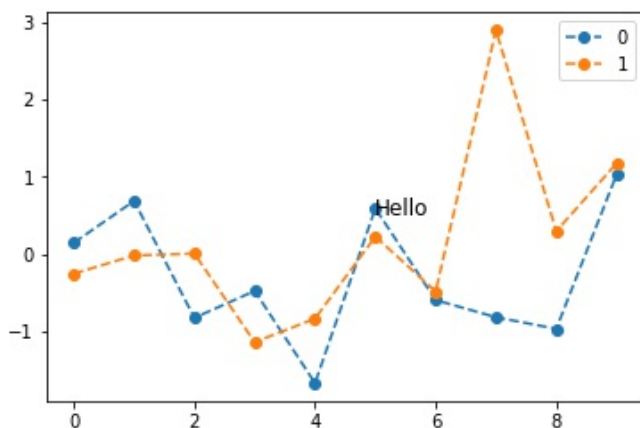


8.标点

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from numpy.random import randint

df=pd.DataFrame(np.random.randn(10,2))
df.plot(style='--o')
plt.text(5,0.5, 'Hello', fontsize=12)
```

Screenshot :



9.饼图

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
s=pd.Series(3*np.random.rand(4),index=['a', 'b', 'c', 'd'],name='series')
plt.axis('equal')
plt.pie(s,
        explode=[0.1,0,0,0],
        labels=s.index,
        colors=['r', 'g', 'b', 'c'],
        autopct='%.2f%%',
        pctdistance=0.6,
        labeldistance=1.2,
        shadow=True,
        startangle=0,
        radius=1.5,
```

```
frame=False)
```

Screenshot :

