

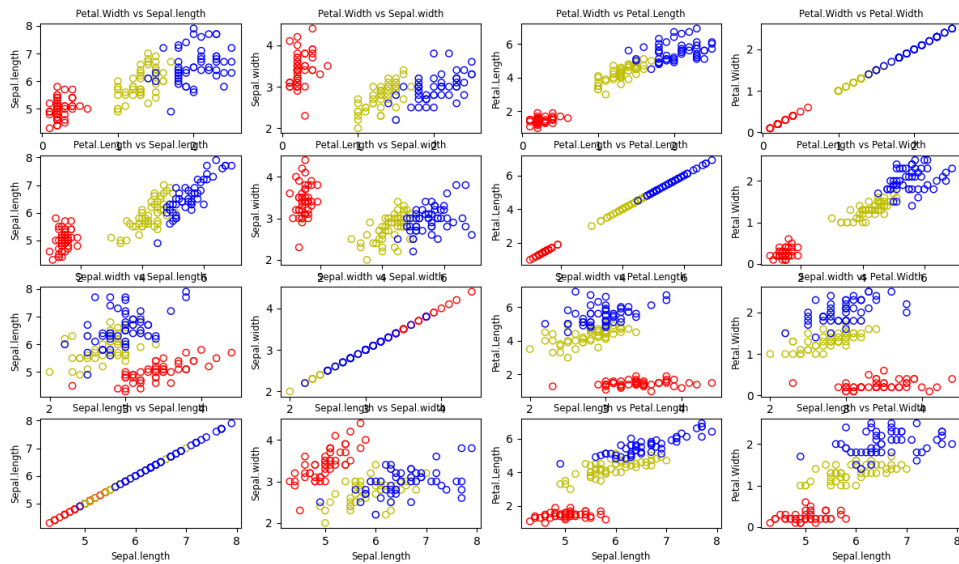
数据可视化作业

姓名：路修远 学号：2018211148 班级：2018211303

作业1

使用IRIS数据集，在一个figure中绘制出右侧的16个子图。分别使用花瓣长度、花瓣宽度、花萼长度和花萼宽度这四种数据，两两组合，形成散点。

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 # 读取数据
6 BasePath = r'D:\python_repo\homework\11\sample' # csv文件的保存路径
7 iris = pd.read_csv(BasePath+'\\iris.csv')
8 print(iris)
9 colors = ['r', 'y', 'b'] # 定义三种散点的颜色
10 Species = iris.Species.unique() # 对类别去重
11 print(Species)
12
13 names = ['Sepal.Length', 'Sepal.Width', 'Petal.Length', 'Petal.Width']
14
15 figure, ax = plt.subplots(nrows=4, ncols=4, figsize=(40,40))
16 for i in range(0, 16):
17     r = i // 4
18     c = i % 4
19     ax[r][c].set_title(names[3-r] + ' vs ' + names[c], fontsize='small')
20     ax[r][c].set_xlabel(names[3-r], fontsize='small')
21     ax[r][c].set_ylabel(names[c], fontsize='small')
22     for j in range(3):
23         ax[r][c].scatter(iris.loc[iris.Species == Species[j], names[3-r]],
24                           iris.loc[iris.Species == Species[j], names[c]],
25                           s=35, c='', edgecolors=colors[j], label=Species[j])
26
27 plt.show()
```



作业2

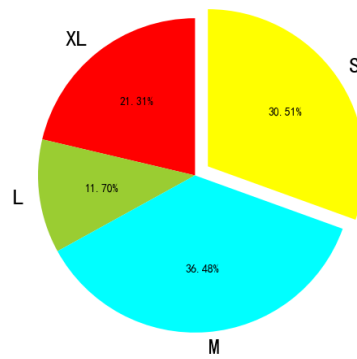
1. 找一组自己感兴趣的真实数据，绘制出饼图。并看看数据 的项数在什么范围比较合适在饼图中展示；
2. 调整数据的顺序或角度，使得第一个扇区在12点方向开始；
3. 调整字体的大小、标签的位置等参数。

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5
6 plt.rcParams['font.sans-serif']=['SimHei'] #解决中文乱码
7 plt.figure(figsize=(6,9)) #调节图形大小
8 labels = ['XL','L','M','S'] #定义标签
9 sizes = [461,253,789,660] #每块值
10 colors = ['red','yellowgreen','cyan','yellow'] #每块颜色定义
11 explode = (0,0,0,0.1) #将某一块分割出来，值越大分割出的间隙越大
12 patches,text1,text2 = plt.pie(sizes,
13                               explode=explode,
14                               labels=labels,
15                               colors=colors,
16                               autopct = '%3.2f%%', #数值保留固定小数位
17                               startangle =90, #逆时针起始角度设置
18                               pctdistance = 0.6) #数值距圆心半径倍数距离
19 for t in text1:
20     t.set_size(20)
21 plt.title('服装设计',fontsize=30)
22 plt.axis('equal')
23 plt.show()
24

```

服装设计



作业3

在中国地图上展示每个省 的高考人数或大学数量。

```
1 from pyecharts import options as opts
2 from pyecharts.charts import Map
3 import random
4 from pyecharts.render import make_snapshot
5 from snapshot_phantomjs import snapshot
6 import csv
7
8
9 def data():
10     with open(r"D:\python_repo\homework\11\sample\中国大学数量.csv",'r') as f:
11         reader = csv.reader(f)
12         head = next(reader)
13         next(reader)
14         lz = []
15         for line in reader:
16             value = float(line[1][:-1])
17             lz.append([line[0],value])
18             print([line[0],value])
19         return lz
20
21
22 def map_visualmap() -> Map:
23     c = (
24         Map()
25         .add("各省2017年考生数量", [z for z in data()], "china")
26         .set_global_opts(
27             title_opts=opts.TitleOpts(title="Map-VisualMap (连续型)"),
28             visualmap_opts=opts.VisualMapOpts(min_=0,max_=100))
29         .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
30     )
31     return c
32
33 make_snapshot(snapshot, map_visualmap().render(), "map1.png")
```

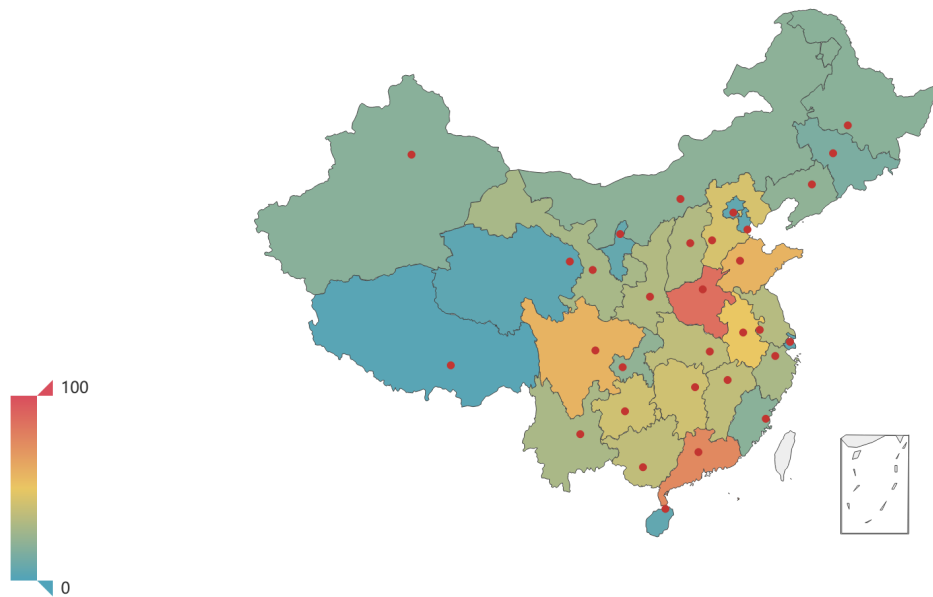
```

34
35 for i in range(5):
36     map_visualmap().render("map"+str(i)+".html")
37
38 print("done")

```

Map-VisualMap (连续型)

各省2017年考生数量



作业4

展示自己家乡所在城市的温度变化热力图，要求至少有10天的数据。

```

1 from pyecharts import options as opts
2 from pyecharts.charts import Geo
3 from pyecharts.globals import ChartType
4 import random
5 from pyecharts.render import make_snapshot
6 from snapshot_phantomjs import snapshot
7 import csv
8
9
10 cities = '合肥市、芜湖市、蚌埠市、淮南市、马鞍山市、淮北市、铜陵市、安庆市、黄山市、滁州市、阜阳市、宿州市、六安市、亳州市、池州市、宣城市'
11 Anhui_cities = cities.split(',')
12 tmp = []
13 with open(r'D:\python_repo\homework\11\hw\da.csv', 'r') as f:
14     reader = csv.reader(f)
15     for line in reader:
16         tmp.append(line)
17
18
19 def geo_guangdong(title, day) -> Geo:
20     c = (
21         Geo()
22         .add_schema(maptype="安徽")
23         .add(
24             title,
25             [list(z) for z in zip(Anhui_cities, tmp[day])],
26             type_=ChartType.HEATMAP,

```

```

27         )
28         .set_global_opts(
29             visualmap_opts=opts.VisualMapOpts(max_=18), #
is_pieewise=True),
30             title_opts=opts.TitleOpts(title="安徽省11月份各地市温度变化情况"),
31         )
32     )
33     return c
34
35
36 for i in range(10):
37     str_date = "11月" + str(i+1) + "日"
38     make_snapshot(snapshot, geo_guangdong(str_date,i).render(),
39                 str(i+1)+".png", pixel_ratio=1)
40     print(str_date)
41

```

数据如下:

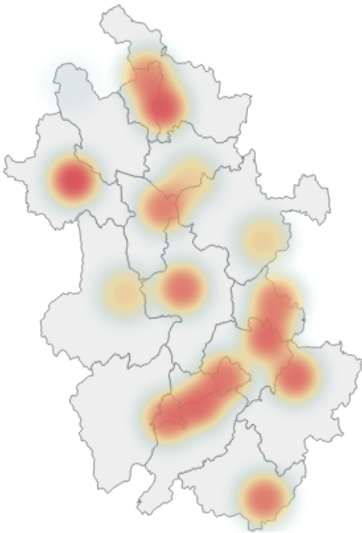
```

1  16.0 ,16.5 ,10.5 ,15.5 ,15.5 ,15.0 ,16.0 ,16.0 ,16.0 ,10.5 ,18.5 ,17.5 ,10.5
   ,2.5 ,16.5 ,16.5
2  15.0 ,16.0 ,10   ,14.5 ,16.5 ,12.0 ,16.0 ,16.0 ,15.5 ,10   ,15.0 ,17.5 ,10
   ,4.0 ,16.5 ,16.0
3  13.0 ,13.0 ,10   ,13.0 ,13.0 ,11.5 ,14.0 ,14.5 ,13.5 ,10   ,13.5 ,16.5 ,10
   ,4.5 ,13.5 ,12.5
4  14.0 ,14.5 ,8.5   ,14.0 ,14.5 ,13.5 ,15.0 ,15.0 ,15.0 ,8.5   ,15.0 ,16.5 ,8.5
   ,4.0 ,15.5 ,15.0
5  15.0 ,16.5 ,9.5   ,14.5 ,17.0 ,13.0 ,18.0 ,17.0 ,18.5 ,9.5   ,18.0 ,14.0 ,9.5
   ,1.5 ,17.5 ,17.0
6  15.5 ,17.0 ,11.0 ,17.0 ,17.5 ,15.5 ,17.0 ,17.5 ,19.0 ,11.0 ,20.5 ,14.0 ,11.0
   ,1.0 ,17.0 ,16.0
7  14.5 ,16.5 ,10.5 ,17.0 ,17.0 ,14.0 ,17.0 ,17.0 ,17.5 ,10.5 ,18.0 ,14.5 ,10.5
   ,2.5 ,16.0 ,16.0
8  14.0 ,14.5 ,8.0   ,15.0 ,15.0 ,12.0 ,14.5 ,15.0 ,15.5 ,8.0   ,15.5 ,14.5 ,8.0
   ,3.5 ,14.5 ,13.5
9  13.5 ,14.0 ,9.0   ,14.0 ,15.0 ,12.0 ,14.5 ,14.5 ,13.5 ,9.0   ,14.5 ,17.5 ,9.0
   ,2.5 ,14.0 ,13.5
10 12.5 ,14.0 ,10.0 ,14.5 ,14.5 ,13.0 ,14.0 ,14.0 ,13.5 ,10.0 ,14.0 ,17.5 ,10.0
    ,2.0 ,14.0 ,13.5
11

```

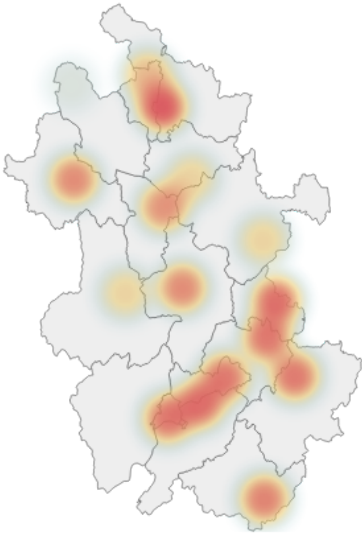
安徽省11月份各地市温度变化情况

11月1日



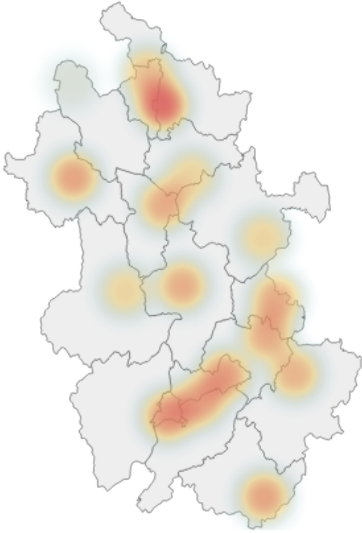
安徽省11月份各地市温度变化情况

11月2日



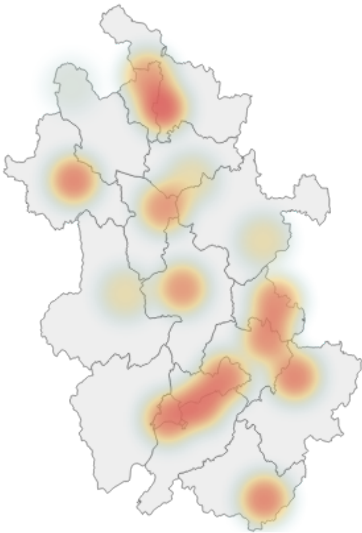
安徽省11月份各地市温度变化情况

11月3日



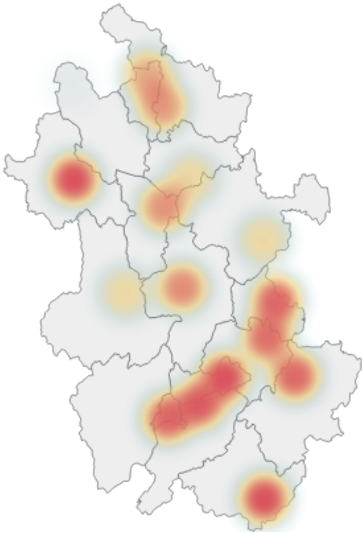
安徽省11月份各地市温度变化情况

11月4日



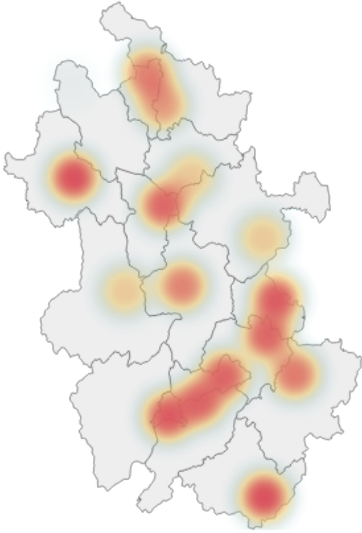
安徽省11月份各地市温度变化情况

11月5日



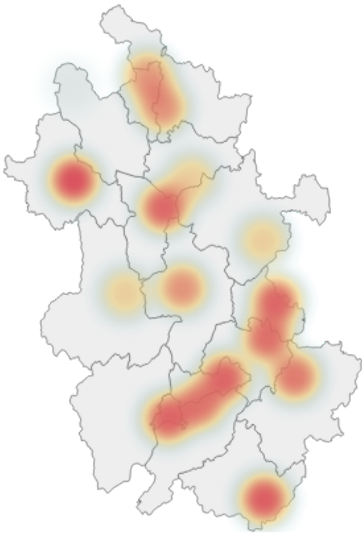
安徽省11月份各地市温度变化情况

11月6日



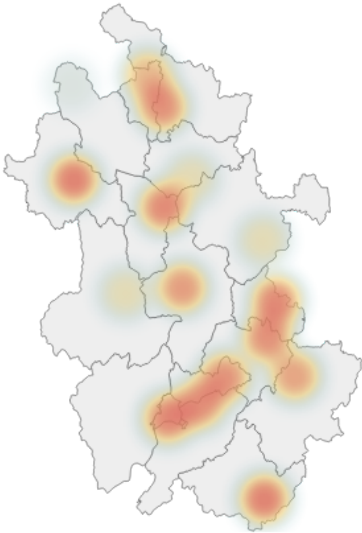
安徽省11月份各地市温度变化情况

11月7日



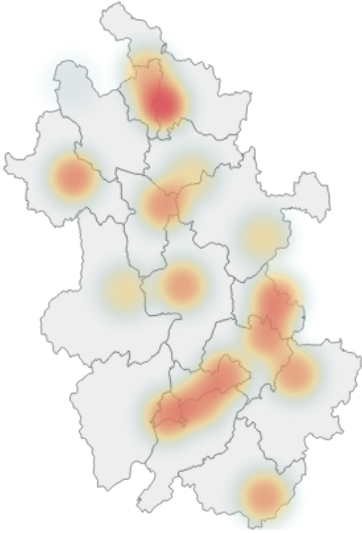
安徽省11月份各地市温度变化情况

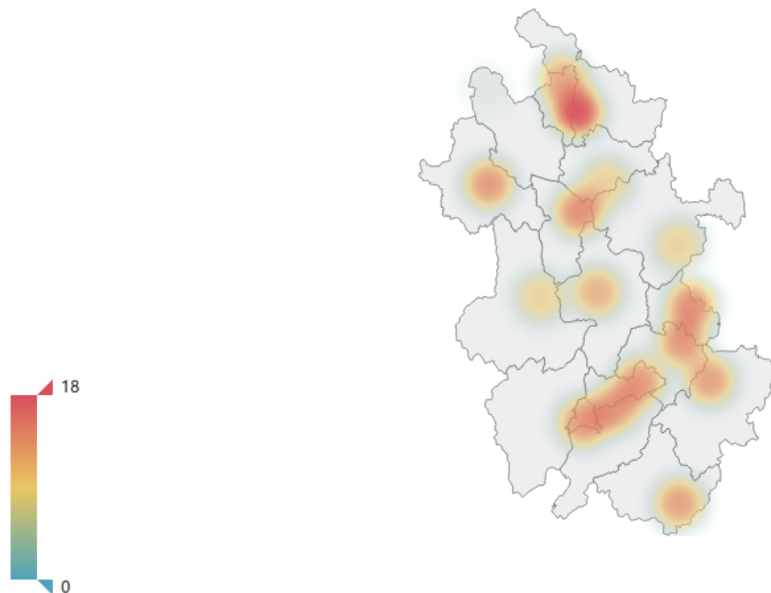
11月8日



安徽省11月份各地市温度变化情况

11月9日

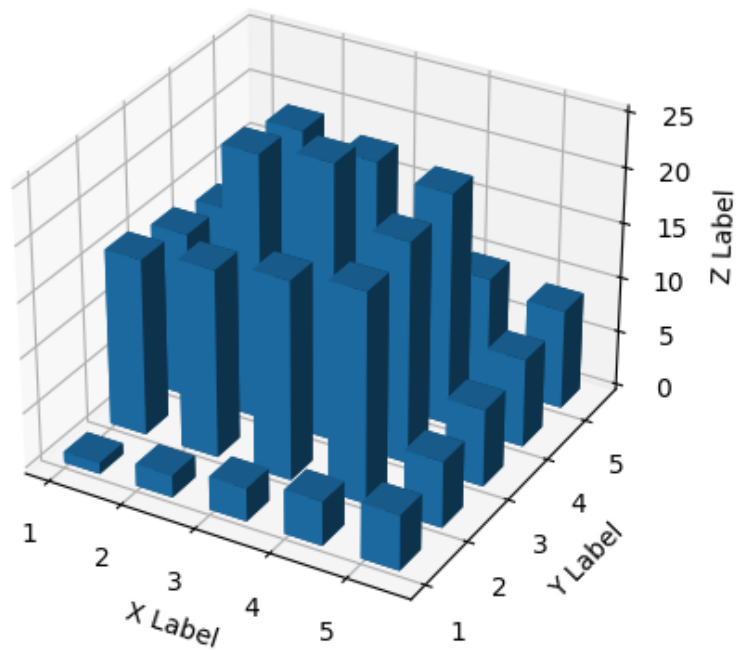




作业5

生成一个直方图，有25根直方柱。要求直方柱的最小值是1，最大值是25，要求沿着边缘，从外到内逐步增大

```
1 import matplotlib.pyplot as plt
2 from mpl_toolkits.mplot3d import Axes3D
3 import numpy as np
4
5 # 设置画布
6 fig = plt.figure()
7 ax = fig.add_subplot(projection='3d')
8 ax.set_xlabel('X Label')
9 ax.set_ylabel('Y Label')
10 ax.set_zlabel('Z Label')
11
12 # 设置x、y、z的关系
13 x = np.array([1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1,
14               2, 3, 4, 5])
15 y = np.array([1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5,
16               5, 5, 5, 5])
17 z = np.array([1, 2, 3, 4, 5, 16, 17, 18, 19, 6, 15, 24, 25, 20, 7, 14, 23,
18               22, 21, 8, 13, 12, 11, 10, 9])
19 bottom = np.zeros_like(z)
20
21 # 绘制3D直方图
22 width = depth = 0.5
23 ax.bar3d(x, y, bottom, width, depth, z, shade=True)
24 ax.set_xticks([1, 2, 3, 4, 5])
25 ax.set_yticks([1, 2, 3, 4, 5])
26
27 # 显示图形
28 plt.show()
```



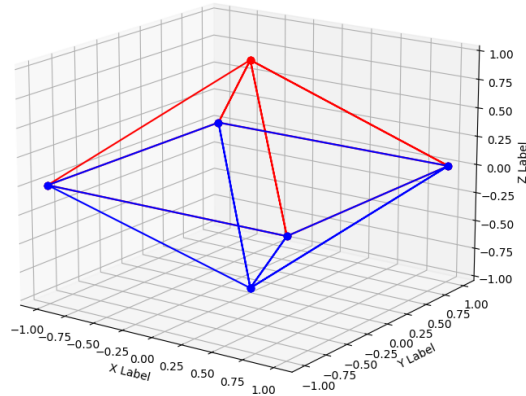
作业6

1. 生成一个金字塔的线图
2. 生成一上一下两个金字塔，叠放在一起

```

1  import numpy as np
2  from mpl_toolkits.mplot3d import Axes3D
3  import matplotlib.pyplot as plt
4
5  # 设置画布
6  fig = plt.figure()
7  ax = fig.add_subplot(projection='3d')
8  ax.set_xlabel('X Label')
9  ax.set_ylabel('Y Label')
10 ax.set_zlabel('Z Label')
11
12 # 设置x、y、z轴数据
13 x = np.array([0, -1, 1, 0, 1, 1, 0, 1, -1, 0, -1, -1])
14 y = np.array([0, -1, -1, 0, -1, 1, 0, 1, 1, 0, 1, -1])
15 z1 = np.array([1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0])
16 z2 = -z1
17
18
19 # 使用plot画3D线图
20 plt.plot(x, y, z1, 'r', marker='o')
21 plt.plot(x, y, z2, 'b', marker='o')
22 ax.set_zticks([-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1])
23 ax.set_xticks([-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1])
24 ax.set_yticks([-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1])
25
26 # 显示线图

```



作业7

生成一个散点图，如下图所示。 $z = x^2 + y^2$

```

1  import matplotlib.pyplot as plt
2  import numpy as np
3  from mpl_toolkits.mplot3d import Axes3D
4  import math
5
6  # 设置画布
7  fig = plt.figure(figsize=(40,40))
8  ax = fig.add_subplot(projection='3d')
9  ax.set_xlabel('X Label')
10 ax.set_ylabel('Y Label')
11 ax.set_zlabel('Z Label')
12
13 # 设置x、y、z的值
14 x = np.random.randint(-100, 100, 5000)
15 y = np.random.randint(-100, 100, 5000)
16 z1 = x**2 + y**2 - 20000
17 z2 = -z1
18
19 # 绘制散点图
20 ax.scatter(x, y, z1, zdir='z', s=20, c='r', marker='^', depthshade=True)
21 ax.scatter(x, y, z2, zdir='z', s=20, c='b', marker='v', depthshade=True)
22
23 # 显示图画
24 plt.show()

```

