

数据可视化作业 2

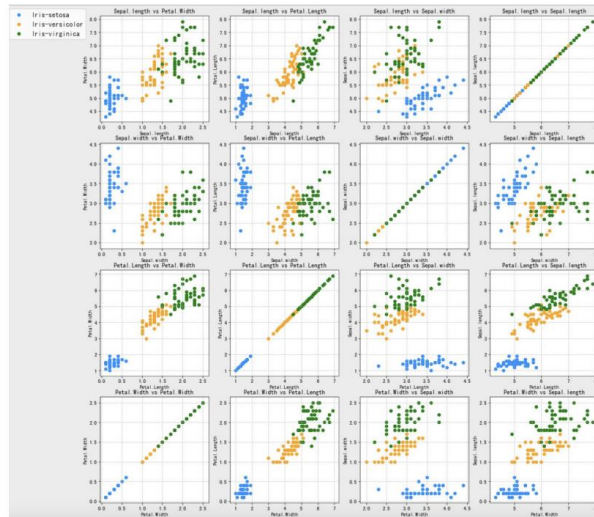
作业一

1. 作业要求

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。



2. 解决方案

1. 设置图大小

```
plt.rcParams['figure.figsize'] = (12.0, 12.0)
```

1200×1200 像素

2. 定义散点颜色，对花类别去重

```
iris = pd.read_csv('iris.csv')
print(iris)
colors = ['r', 'y', 'b'] # 定义三种散点的颜色
Species = iris.Species.unique() # 对类别去重
```

3. 创建数组 order1 和 order2 用于按顺序绘制 16 个子图

```
order1=['Sepal.Length', 'Sepal.Width', 'Petal.Length', 'Petal.Width']
order2=['Petal.Width', 'Petal.Length', 'Sepal.Width', 'Sepal.Length']
```

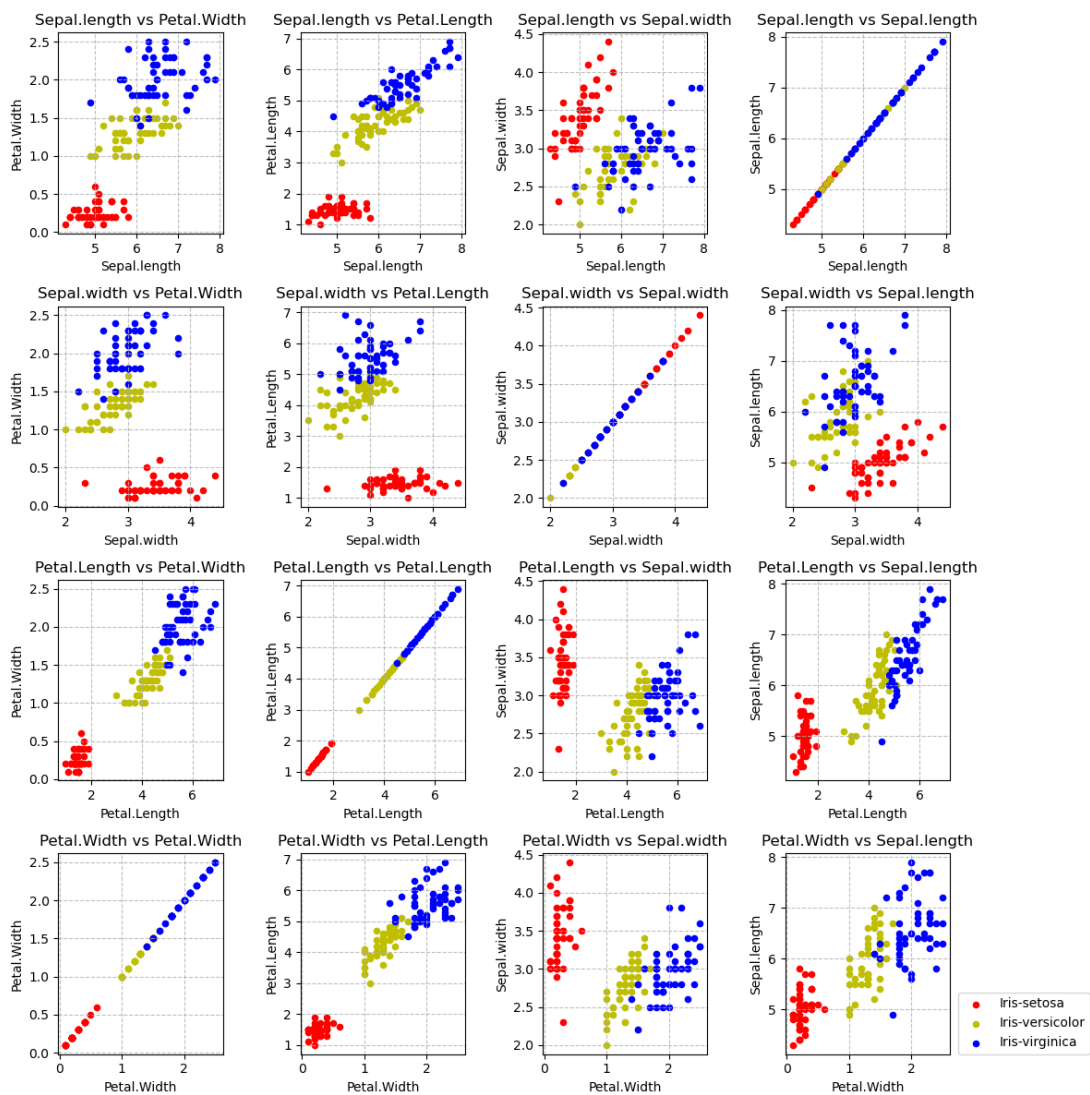
4. 绘制子图并添加标题和 x、y 轴标签，设置网格线

```

for r in range(4):
    for c in range(4):
        plt.subplot(4,4,4*r+c+1)
        for i in range(len(Species)):
            plt.scatter(iris.loc[iris.Species == Species[i], order1[r]],
                        iris.loc[iris.Species == Species[i], order2[c]],s=20,c=colors[i],label=Species[i])
            # 添加轴标签和标题
        plt.title(order1[r]+' vs '+order2[c])
        plt.xlabel(order1[r])
        plt.ylabel(order2[c])
        plt.grid(True, linestyle='--', alpha=0.8) # 设置网格线
        plt.style.use('seaborn-bright')
plt.legend(bbox_to_anchor=(1.05, 0), loc=3, borderaxespad=0) # 设置图例在图外右下角
plt.show()

```

3. 最终图形



作业二

1. 作业要求

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。

2. 解决方案

1. 选择数据来自 Kaggle 题目，泰坦尼克号上乘客的个人信息，文件名 train.csv

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
11,1,3,"Sandstrom, Miss. Marguerite Rut",female,4,1,1,PP 9549,16.7,G6,S
12,1,1,"Bonnell, Miss. Elizabeth",female,58,0,0,113783,26.55,C103,S
13,0,3,"Saunderscock, Mr. William Henry",male,20,0,0,A/5. 2151,8.05,,S
14,0,3,"Andersson, Mr. Anders Johan",male,39,1,5,347082,31.275,,S
15,0,3,"Vestrom, Miss. Hulda Amanda Adolfina",female,14,0,0,350406,7.8542,,S
16,1,2,"Hewlett, Mrs. (Mary D Kingcome) ",female,55,0,0,248706,16,,S
17,0,3,"Rice, Master. Eugene",male,2,4,1,382652,29.125,,Q
18,1,2,"Williams, Mr. Charles Eugene",male,,0,0,244373,13,,S
19,0,3,"Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)",female,31,1,0,345763,18,,S
20,1,3,"MasseMani, Mrs. Fatima",female,,0,0,2649,7.225,,C
21,0,2,"Fynney, Mr. Joseph J",male,35,0,0,239865,26,,S
22,1,2,"Beesley, Mr. Lawrence",male,34,0,0,248698,13,D56,S
23,1,3,"McGowan, Miss. Anna "Annie" ",female,15,0,0,330923,8.0292,,Q
```

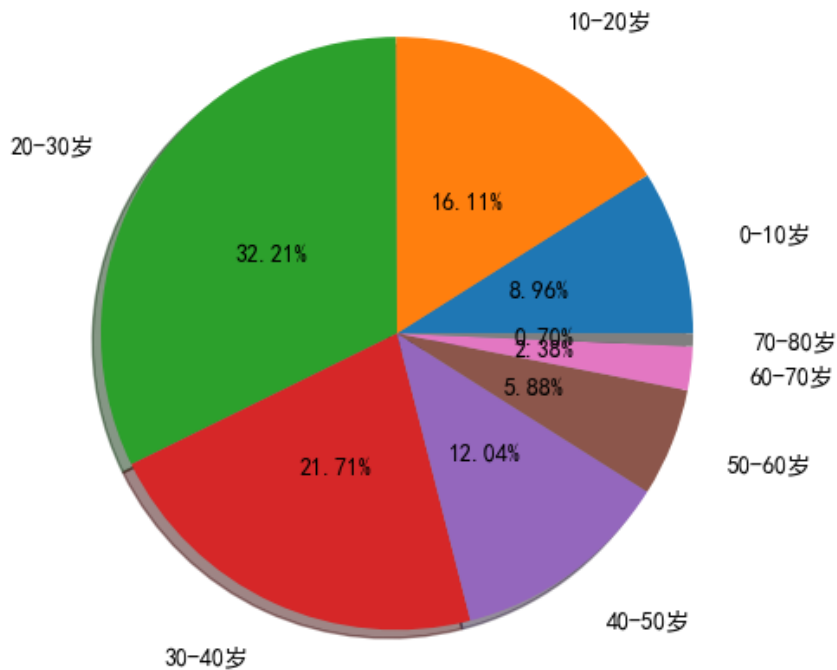
2. 只对乘客年龄一列进行数据分析

```
df=pd.read_csv('train.csv',usecols=['Age'])
df.dropna()#去除空数据
matplotlib.rcParams['font.sans-serif'] = ['SimHei']#设置中文字体
```

3. 首先试着以 10 岁为跨度进行分析并绘制饼图

```
sections = [0, 10, 20, 30, 40, 50, 60, 70, 80] # 设置年龄分段
sections_name = ['0-10岁', '10-20岁', '20-30岁', '30-40岁', '40-50岁', '50-60岁', '60-70岁', '70-80岁'] # 年龄分段标签
```

泰坦尼克号乘客各年龄段人数



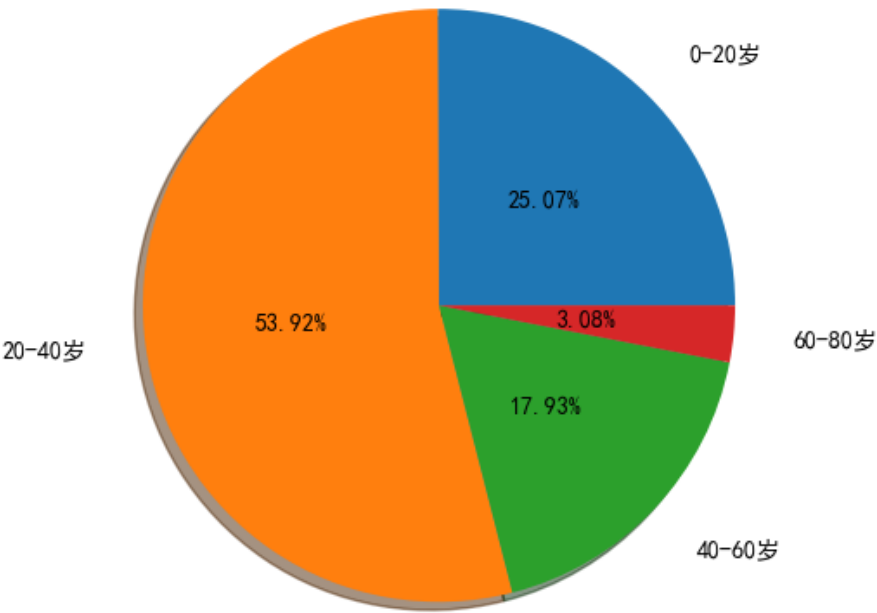
可以观察到年龄分段过于密集，元素过多，数据项数不合理

4. 以 20 岁为跨度进行年龄划分，并绘制饼图，调整起始角度为 0(12 点)，开启阴影，园内文本距圆心距离 0.5 倍圆心距，标签文本距圆心 1.2 倍圆心距，设置标题

```
sections=[0,20,40,60,80]#设置年龄分段
sections_name = ['0-20岁', '20-40岁', '40-60岁', '60-80岁']#年龄分段标签
result = pd.cut(df.Age, sections, labels=sections_name)
count = pd.value_counts(result, sort=False)#计算每个年龄人数list
plt.title("泰坦尼克号乘客各年龄段人数")
plt.pie(count, labels=sections_name, labeldistance=1.2, autopct="%1.2f%%", shadow=True,
        startangle=0, pctdistance=0.5)
plt.show()
```

3. 最终图形

泰坦尼克号乘客各年龄段人数



作业三

1. 作业要求

作业3:
在中国地图上展示每个省
的高考人数或大学数量

省/市	同考人数		大学数量	
	2017年	2016年	211&985大学数量	公办本科大学数量
河南	86.3万	82万	1	37
广东	75.7万	73.3万	4	41
山东	58.3万	60.2万	3	44
四川	58.3万	57.13万	5	34
安徽	49.9万	50.99万	3	30
河北	43.6万	42.31万	0	36
贵州	41.2万	37.38万	1	19
湖南	41.1万	40.16万	3	31
广西	36.5万	33万	1	24
江西	36.5万	36.06万	1	23
湖北	36.2万	36.14万	1	36
江苏	33万	36.04万	11	45
陕西	31.9万	32.8万	1	34
山西	31.7万	33.9万	1	22
云南	29.3万	28万	1	22
浙江	29.1万	30.74万	1	32
甘肃	28.5万	29.6万	1	17
重庆	21.1万	24.88万	2	17
辽宁	20.8万	21.82万	4	4
内蒙古	19.8万	20.11万	1	14
福建	18.8万	17.5万	2	22
黑龙江	18.8万	19.7万	4	26
新疆	18.4万	16.61万	2	13
吉林	14.3万	14.85万	3	25
宁夏	6.9万	6.9万	1	4
北京	6万	6.12万	26	60
海南	5.7万	6.04万	1	5
天津	5.7万	6万	4	18
上海	5万	5.1万	9	31
青海	4.6万	4.5万	1	3
西藏	2.8万	2.4万	1	3

2. 解决方案

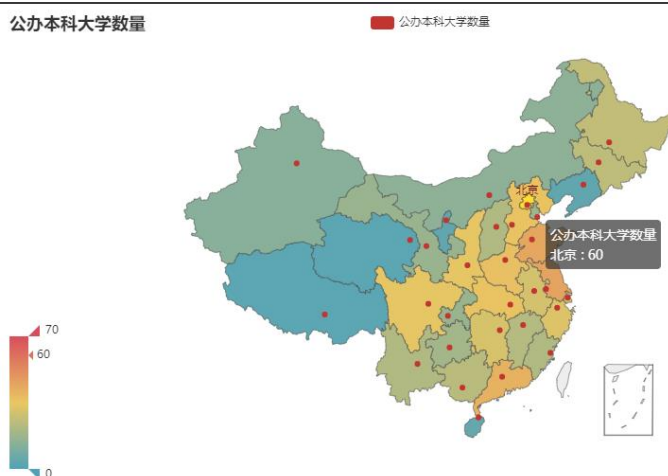
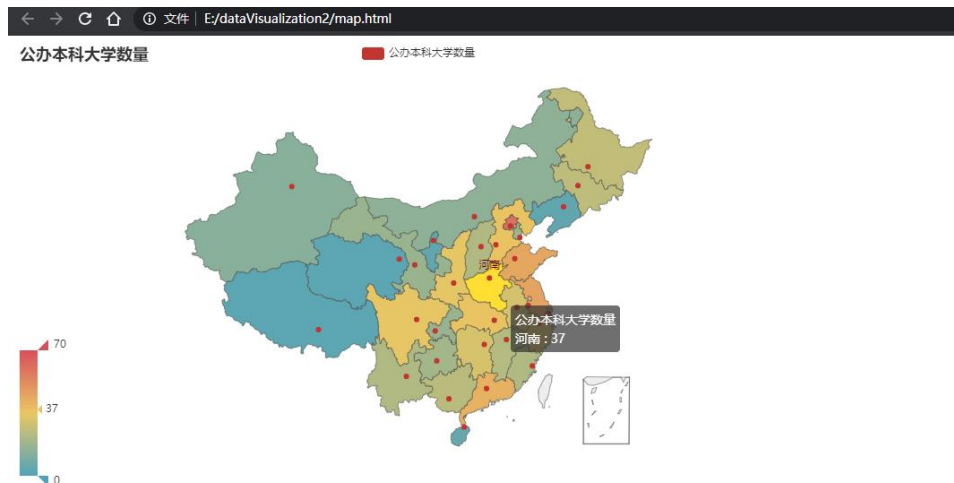
1. 以最后一列公办本科大学数量代表本省大学数量，取得该数据

```
class Data:
    provinces = df.iloc[1:, 0].values.tolist()#省份
    @staticmethod
    def values() -> list:
        return df.iloc[1:,4].values.tolist()#公办本科大学数量列
```

2. 调用 pyecharts 库生成 map.html

```
def map_visualmap() -> Map:
    c = (
        Map()
        .add("公办本科大学数量", [list(z) for z in zip(Data.provinces, Data.values())], "china")
        .set_global_opts(
            title_opts=opts.TitleOpts(title="公办本科大学数量"),
            visualmap_opts=opts.VisualMapOpts(min_=0, max_=70))
        .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
    )
    return c
map_visualmap().render("map.html")
```

3. 最终图形



作业四

1. 作业要求

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。

2. 解决方案

1. 在中国天气网查询淮南市 12 月 5 日到 12 月 14 日天气，并记录在 values 中



class Data:

```
city = ["淮南市"]
```

```
values=[[3],[2],[2],[5],[5],[5],[1],[3],[-1],[-4]]
```

2. 调用 pyecharts 库对每一天的数据生成 html 并截图成 png

```
def geo_hometown(title,i) -> Geo:
    c = (Geo()
        .add_schema(maptype="淮南")
        .add(
            title, [list(z) for z in zip(Data.city, Data.values[i])],
            type=ChartType.HEATMAP)
        .set_global_opts(
            visualmap_opts=opts.VisualMapOpts(min_=-4,max_=5, is_piecewise=True),
            title_opts=opts.TitleOpts(title="淮南市近十天温度变化情况"),
        )
    )
    return c

for i in range(10):
    str_date = "12月" + str(i + 5) + "日"
    geo_hometown(str_date,i).render(str_date + ".html")
    make_snapshot(snapshot, geo_hometown(str_date,i).render(),
                  str(i + 1) + ".png", pixel_ratio=1)
```

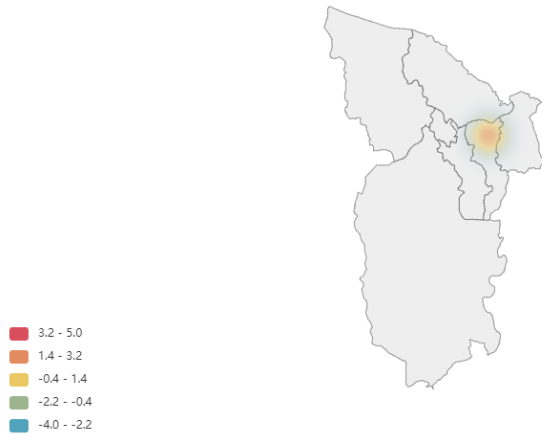
3. 将 10 张 png 合成成一张 gif

```
frames=[]
for i in range(10):
    frames.append(imageio.imread(str(i+1)+".png"))
imageio.mimsave("huainan.gif", frames, 'GIF', duration=0.35)
```

3. 最终图形

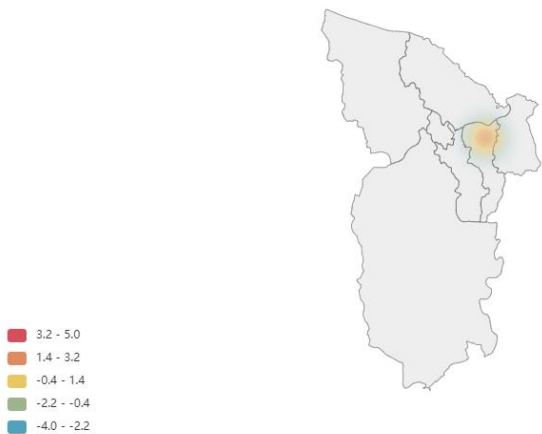
淮南市近十天温度变化情况

12月5日



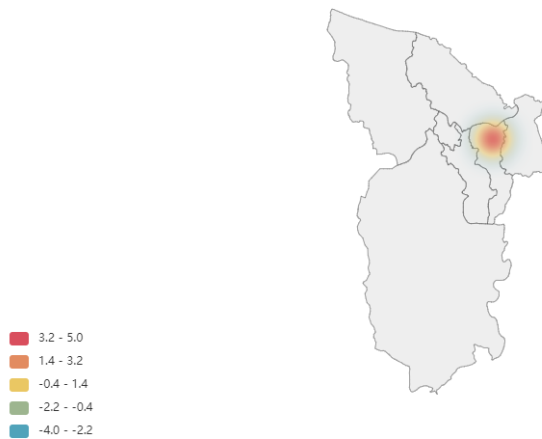
淮南市近十天温度变化情况

12月6日

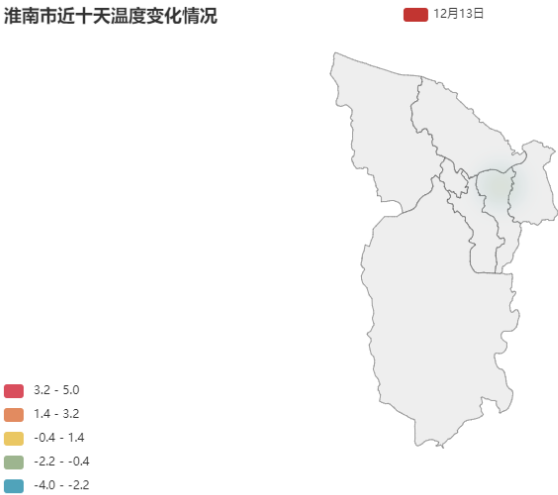


淮南市近十天温度变化情况

12月8日



淮南市近十天温度变化情况



1	2020/12/14 0:11	PNG 文件	1,762 KB
2	2020/12/14 0:11	PNG 文件	1,762 KB
3	2020/12/14 0:11	PNG 文件	1,762 KB
4	2020/12/14 0:11	PNG 文件	1,762 KB
5	2020/12/14 0:11	PNG 文件	1,762 KB
6	2020/12/14 0:11	PNG 文件	1,762 KB
7	2020/12/14 0:11	PNG 文件	1,762 KB
8	2020/12/14 0:11	PNG 文件	1,762 KB
9	2020/12/14 0:12	PNG 文件	1,762 KB
10	2020/12/14 0:12	PNG 文件	1,762 KB
12月5日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月6日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月7日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月8日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月9日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月10日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月11日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月12日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月13日	2020/12/14 0:11	Chrome HTML D...	4 KB
12月14日	2020/12/14 0:12	Chrome HTML D...	4 KB

生成的 gif 链接

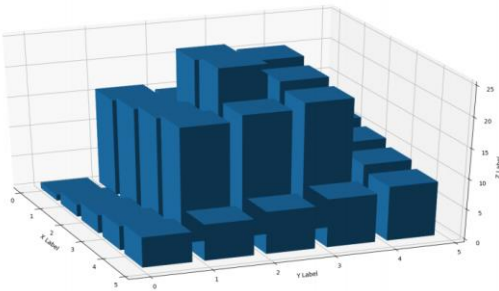
<https://bl.sbing.org/file/chevereto-jia/2020/12/14/QcJWK.gif>

作业五

1. 作业要求

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



2. 解决方案

1. 设置生成 3D 图，x 轴、y 轴、z 轴标签

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
```

2. 根据题目要求生成 x 轴、y 轴、z 轴数据

```
x = np.array([0, 1, 2, 3, 4, 4, 4, 4, 4, 3, 2, 1, 0, 0, 0, 0, 1, 2, 3, 3, 3, 2, 1, 1, 2]) # 生成x轴的数据
y = np.array([0, 0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4, 4, 3, 2, 1, 1, 1, 1, 2, 3, 3, 3, 2, 2]) # 生成y轴的数据
z = np.array([i for i in range(1, 26)]) # 生成z轴的数据
```

3. 绘制 3D 图

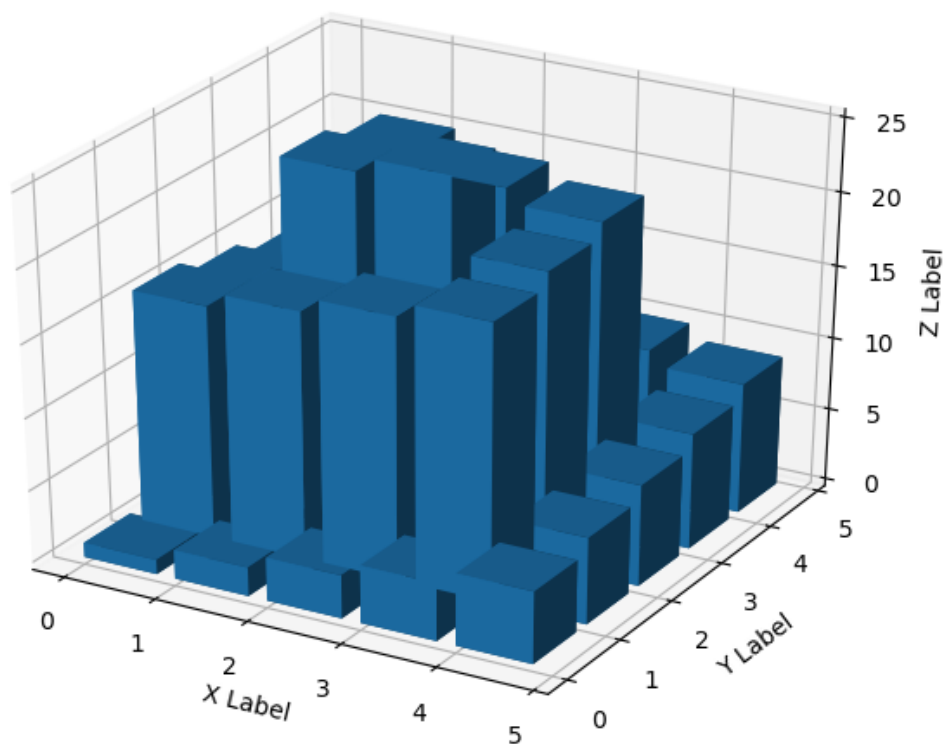
```
print('=====bottom')
bottom = np.zeros_like(z) # 产生一个全零的矩阵，形状和z一样。bottom表示直方图从哪个数值开始是底部

width = depth = 0.8 # width和depth表示直方柱的宽度和深度在单元格中比例
# 3. 调用bar3d，画3D直方图
ax.bar3d(x, y, bottom, width, depth, z, shade=True)

ax.set_xticks([0, 1, 2, 3, 4, 5])
ax.set_yticks([0, 1, 2, 3, 4, 5])

# 4. 显示图形
plt.show()
```

3. 最终图形



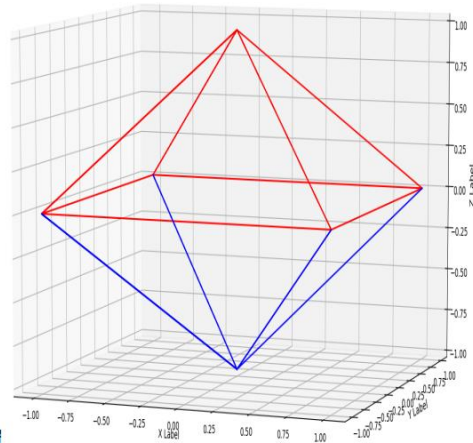
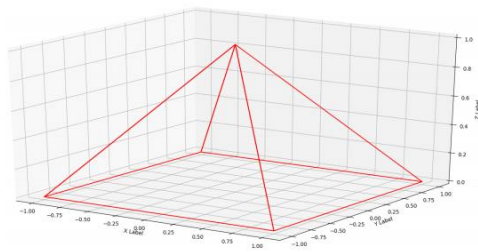
作业六

1. 作业要求

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



2. 解决方案

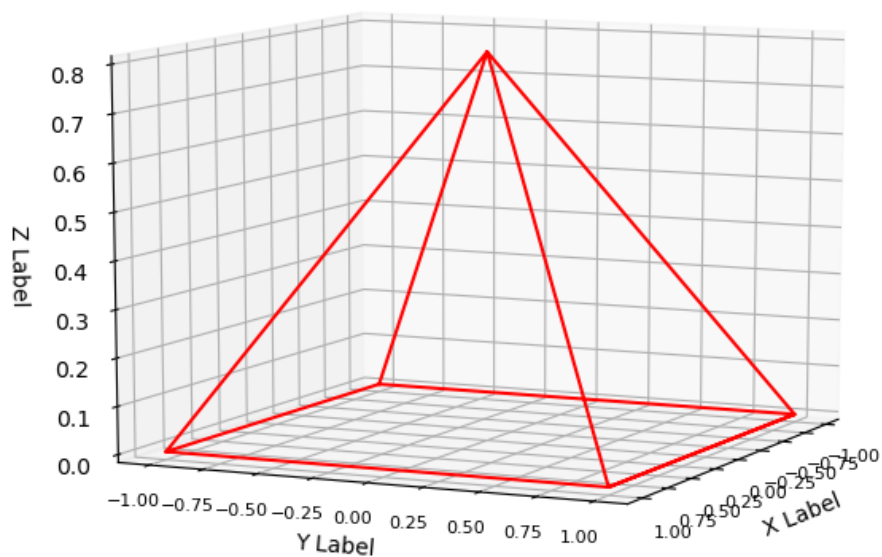
1. 设置 3D 图视角，x 轴、y 轴、z 轴标签

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.view_init(elev=10,azim=25)
plt.yticks(fontsize=8)
plt.xticks(fontsize=8)
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
```

2. 首先绘制单个金字塔，因为无法一笔画完，最后需返回再连线，设置 x、y、z 轴数据

```
# 2. 生成数据
x1 = np.array([-1,0,1,-1,-1,0,1,-1,1,1]) # 生成x轴的数据
y1 = np.array([-1,0,-1,-1,1,0,1,1,1,-1]) # 生成y轴的数据
z1 = np.array([0,0.8,0,0,0,0.8,0,0,0,0])
z2 = np.array([0,-0.8,0,0,0,-0.8,0,0,0,0])
# 3. 调用plot, 画3D的线图
ax.plot(x1, y1, z1, "r")
```

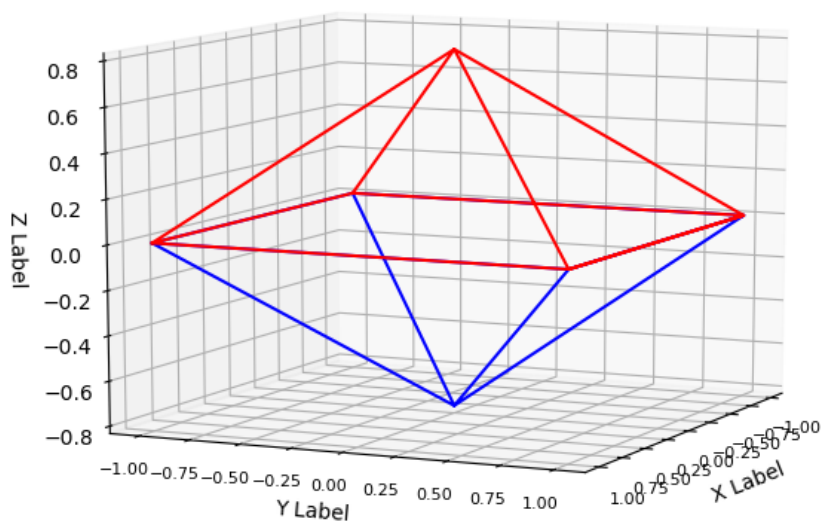
3. 绘制 3D 图



4. 绘制上下两个金字塔并叠在一起

```
x1 = np.array([-1,0,1,-1,-1,0,1,-1,1,1]) # 生成x轴的数据
y1 = np.array([-1,0,-1,-1,1,0,1,1,1,-1]) # 生成y轴的数据
z1 = np.array([0,0.8,0,0,0,0.8,0,0,0,0])
z2 = np.array([0,-0.8,0,0,0,-0.8,0,0,0,0])
# 3. 调用plot, 画3D的线图
ax.plot(x1, y1, z2, "b")
ax.plot(x1, y1, z1, "r")
```

3. 最终图形

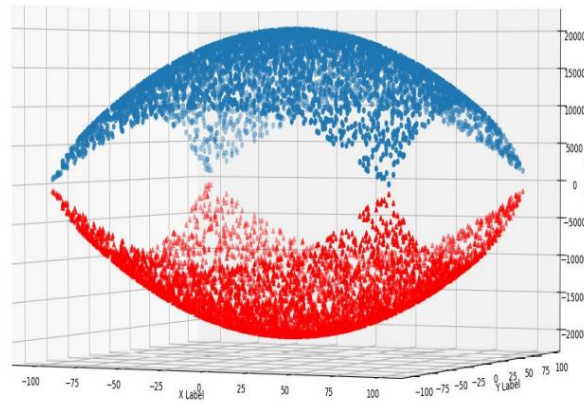


作业七

1. 作业要求

作业7

生成一个散点图，如下图所示。 $z=x^2+y^2$



2. 解决方案

1. 设置 x 轴、y 轴、z 轴标签、图标和角度

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.view_init(elev=10, azim=30)
ax.set_title('scatter diagram')
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
```

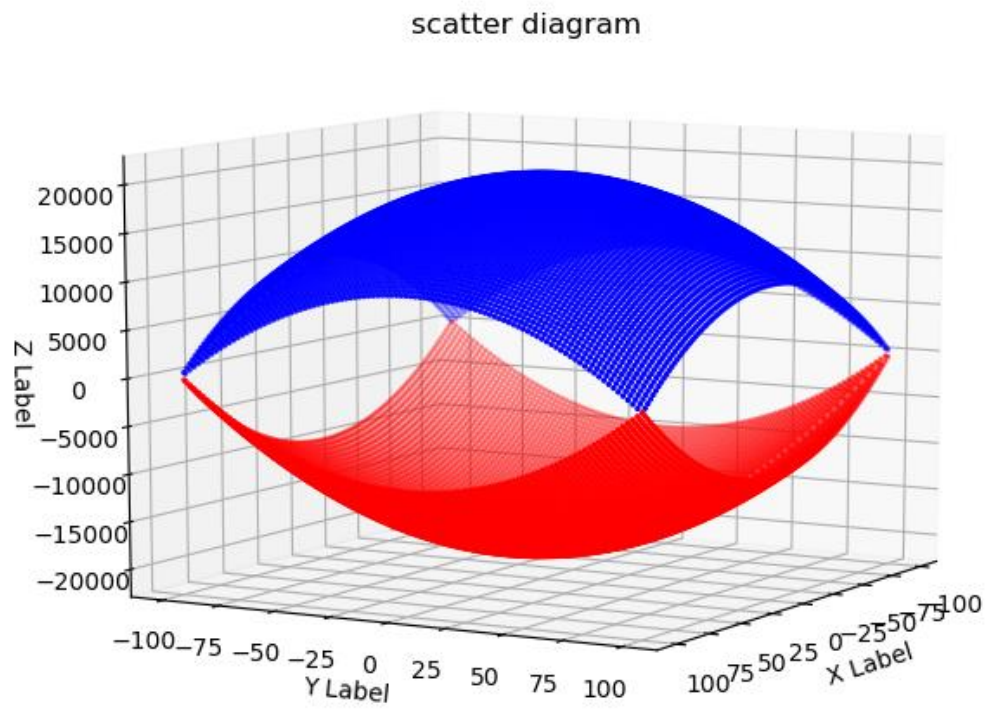
2. 根据图形设置 x 轴、y 轴、z 轴数据

```
# 2.生成数据
_x1=list(range(-100,101,2))*100
x1 = np.array(_x1) # 生成x轴的数据
_y1 = []
for i in range(-100, 101,2):
    for j in range(100):
        _y1.append(i)
y1 = np.array(_y1) # 生成y轴的数据
z1 = 20000-x1*x1-y1*y1
z2 = x1*x1+y1*y1-20000
```

3. 绘制散点图

```
# 3.调用plot, 画3D的线图
ax.scatter(x1, y1, z1,s=2, c="b")
ax.scatter(x1,y1,z2,s=2,c="r")
#ax.plot(x1, y1, z1, "b")
plt.show()
```

3. 最终图形



附录

```
1. import pandas as pd
2. import numpy as np
3. import imageio
4. import math
5. import random
6. import matplotlib
7. import matplotlib.pyplot as plt
8. from pyecharts import options as opts
9. from pyecharts.charts import Map
10. from pyecharts.render import make_snapshot
11. from snapshot_phantomjs import snapshot
12. from pyecharts.charts import Geo
13. from pyecharts.globals import ChartType
14. from mpl_toolkits.mplot3d import Axes3D
15.
16.
17. def job1():
18.     plt.rcParams['figure.figsize'] = (12.0, 12.0)
19.     # 读取数据
20.     iris = pd.read_csv('iris.csv')
21.     print(iris)
22.     colors = ['r', 'y', 'b'] # 定义三种散点的颜色
23.     Species = iris.Species.unique() # 对类别去重
24.     print(Species)
25.     order1=['Sepal.length','Sepal.width','Petal.Length','Petal.Width']
26.     order2=['Petal.Width','Petal.Length','Sepal.width','Sepal.length']
27.     for r in range(4):
28.         for c in range(4):
29.             plt.subplot(4,4,4*r+c+1)
30.             for i in range(len(Species)):
31.                 plt.scatter(iris.loc[iris.Species == Species[i], order1[r]],
32.                             iris.loc[iris.Species == Species[i], order2[c]],
33.                             s=20 ,c=colors[i], label=Species[i])
34.             # 添加轴标签和标题
35.             plt.title(order1[r]+' vs '+order2[c])
36.             plt.xlabel(order1[r])
37.             plt.ylabel(order2[c])
38.             plt.grid(True, linestyle='--', alpha=0.8) # 设置网格线
39.             plt.style.use('seaborn-bright')
```

```

39.     plt.legend(bbox_to_anchor=(1.05, 0), loc=3, borderaxespad=0) #设置图例在
    图外右下角
40.     plt.show()
41.
42. def job2():
43.     df=pd.read_csv('train.csv',usecols=['Age'])
44.     df.dropna()#去除空数据
45.     matplotlib.rcParams['font.sans-serif'] = ['SimHei']#设置中文字体
46.
47.     #sections = [0, 10, 20, 30, 40, 50, 60, 70, 80] # 设置年龄分段
48.     #sections_name = ['0-10岁', '10-20岁', '20-30岁', '30-40岁', '40-50岁',
    '50-60岁', '60-70岁', '70-80岁'] # 年龄分段标签
49.
50.     sections=[0,20,40,60,80]#设置年龄分段
51.     sections_name = ['0-20岁', '20-40岁', '40-60岁', '60-80岁']#年龄分段标
    签
52.     result = pd.cut(df.Age, sections, labels=sections_name)
53.     count = pd.value_counts(result, sort=False)#计算每个年龄人数 list
54.     plt.title("泰坦尼克号乘客各年龄段人数")
55.     plt.pie(count, labels=sections_name, labeldistance=1.2, autopct="%1.2f%%",
    shadow=True,
56.             startangle=0, pctdistance=0.5)
57.     plt.show()
58.
59. def job3():
60.     df=pd.read_csv('中国大学数量.csv')
61.
62.     class Data:
63.         provinces = df.iloc[1:, 0].values.tolist()#省份
64.         @staticmethod
65.         def values() -> list:
66.             return df.iloc[1:,4].values.tolist()#公办本科大学数量列
67.
68.     def map_visualmap() -> Map:
69.         c = (
70.             Map()
71.             .add("公办本科大学数量", [list(z) for z in zip(Data.provinces, Data.values())], "china")
72.             .set_global_opts(
73.                 title_opts=opts.TitleOpts(title="公办本科大学数量"),
74.                 visualmap_opts=opts.VisualMapOpts(min_=0,max_=70))
75.             .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
76.         )
77.     return c

```



```

78.     map_visualmap().render("map.html")
79.
80. def job4():
81.     class Data:
82.         city = ["淮南市"]
83.         values=[[3],[2],[2],[5],[5],[5],[1],[3],[-1],[-4]]
84.
85.     def geo_hometown(title,i) -> Geo:
86.         c = (Geo())
87.         .add_schema(maptype="淮南")
88.         .add(
89.             title, [list(z) for z in zip(Data.city, Data.values[i])],
90.             type_=ChartType.HEATMAP)
91.         .set_global_opts(
92.             visualmap_opts=opts.VisualMapOpts(min_=-
93.         4 ,max_=5, is_pieewise=True),
94.             title_opts=opts.TitleOpts(title="淮南市近十天温度变化情况"),
95.         )
96.     return c
97.
98.     for i in range(10):
99.         str_date = "12月" + str(i + 5) + "日"
100.         geo_hometown(str_date,i).render(str_date + ".html")
101.         make_snapshot(snapshot, geo_hometown(str_date,i).render(),
102.             str(i + 1) + ".png", pixel_ratio=1)
103.
104.     frames=[]
105.     for i in range(10):
106.         frames.append(imageio.imread(str(i+1)+".png"))
107.     imageio.mimsave("huainan.gif", frames, 'GIF', duration=0.35)
108.
109. def job5():
110.     fig = plt.figure()
111.     ax = fig.add_subplot(projection='3d')
112.     ax.set_xlabel('X Label')
113.     ax.set_ylabel('Y Label')
114.     ax.set_zlabel('Z Label')
115.
116.     x = np.array([0, 1, 2, 3, 4, 4, 4, 4, 4, 3, 2, 1, 0, 0, 0, 0, 1, 2, 3,
117.         3, 3, 2, 1, 1, 2]) # 生成 x 轴的数据
118.     y = np.array([0, 0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4, 4, 3, 2, 1, 1, 1, 1,
119.         2, 3, 3, 3, 2, 2]) # 生成 y 轴的数据
120.     z = np.array([i for i in range(1, 26)]) # 生成 z 轴的数据

```

```

119.
120.     print('=====bottom')
121.     bottom = np.zeros_like(z) # 产生一个全零的矩阵，形状和 z 一样。bottom 表示
    直方图从哪个数值开始是底部
122.
123.     width = depth = 0.8 # width 和 depth 表示直方柱的宽度和深度在单元格中比例
124.     # 3.调用 bar3d, 画 3D 直方图
125.     ax.bar3d(x, y, bottom, width, depth, z, shade=True)
126.
127.     ax.set_xticks([0, 1, 2, 3, 4, 5])
128.     ax.set_yticks([0, 1, 2, 3, 4, 5])
129.
130.     # 4.显示图形
131.     plt.show()
132.
133. def job6():
134.     fig = plt.figure()
135.     ax = fig.add_subplot(projection='3d')
136.     ax.view_init(elev=10,azim=25)
137.     plt.yticks(fontsize=8)
138.     plt.xticks(fontsize=8)
139.     ax.set_xlabel('X Label')
140.     ax.set_ylabel('Y Label')
141.     ax.set_zlabel('Z Label')
142.     # 2.生成数据
143.     x1 = np.array([-1,0,1,-1,-1,0,1,-1,1,1]) # 生成 x 轴的数据
144.     y1 = np.array([-1,0,-1,-1,1,0,1,1,1,-1]) # 生成 y 轴的数据
145.     z1 = np.array([0,0.8,0,0,0,0.8,0,0,0,0])
146.     z2 = np.array([0,-0.8,0,0,0,-0.8,0,0,0,0])
147.     # 3.调用 plot, 画 3D 的线图
148.     ax.plot(x1, y1, z2, "b")
149.     ax.plot(x1, y1, z1, "r")
150.
151.     plt.show()
152.
153. def job7():
154.     fig = plt.figure()
155.     ax = fig.add_subplot(projection='3d')
156.     ax.view_init(elev=10, azim=30)
157.     ax.set_title('scatter diagram')
158.     ax.set_xlabel('X Label')
159.     ax.set_ylabel('Y Label')
160.     ax.set_zlabel('Z Label')
161.     # 2.生成数据

```

```
162.     _x1=list(range(-100,101,2))*100
163.     x1 = np.array(_x1)  # 生成 x 轴的数据
164.     _y1 = []
165.     for i in range(-100, 101,2):
166.         for j in range(100):
167.             _y1.append(i)
168.     y1 = np.array(_y1)  # 生成 y 轴的数据
169.     z1 = 20000-x1*x1-y1*y1
170.     z2 = x1*x1+y1*y1-20000
171.     # 3.调用 plot, 画 3D 的线图
172.     ax.scatter(x1, y1, z1,s=2, c="b")
173.     ax.scatter(x1,y1,z2,s=2,c="r")
174.     plt.show()
175.
176.
177. if __name__ == '__main__':
178.     job1()
179.     job2()
180.     job3()
181.     job4()
182.     job5()
183.     job6()
184.     job7()
```