1. **问题描述**
2. 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. 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；

调整数据的顺序或角度，使得第一个扇区在12点方向开始；

调整字体的大小、标签的位置等参数。

1. 在中国地图上展示每个省的高考人数或大学数量。
2. 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。
3. 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
4. 生成一个金字塔的线图；

生成一上一下两个金字塔，叠放在一起。

1. 生成一个散点图，z=x^2+y^2
2. **实验环境**

Microsoft Windows 10 版本18363

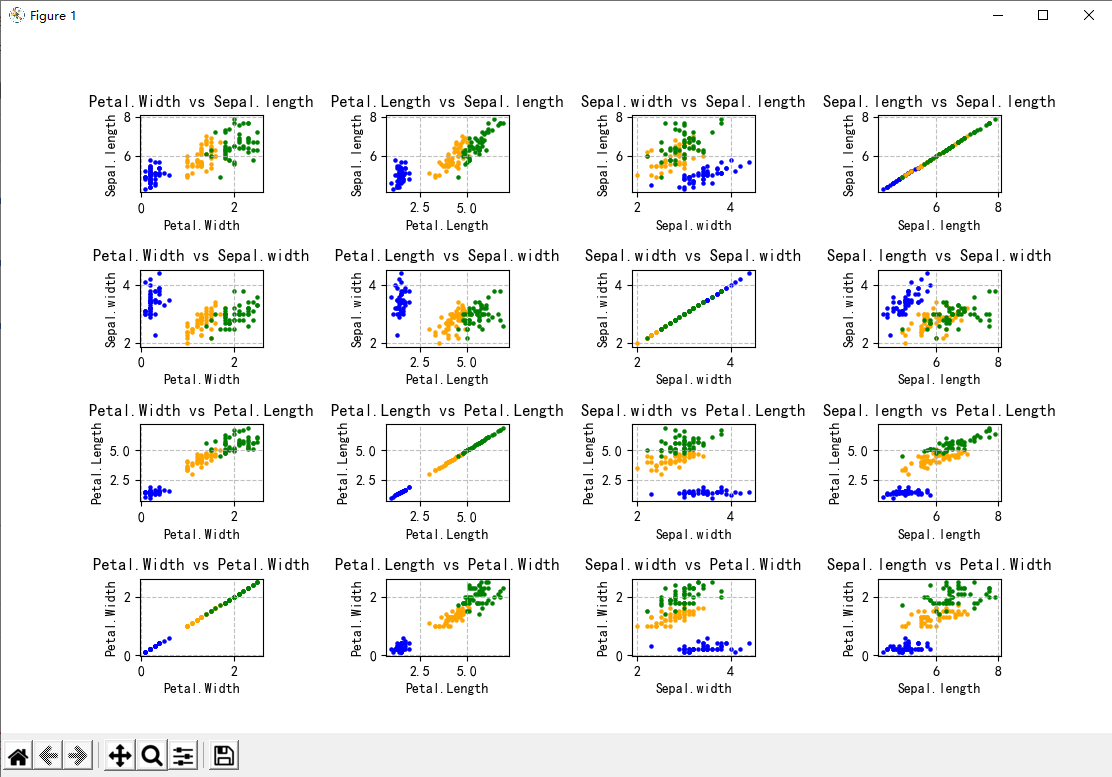
PyCharm 2020.2.1 (Community Edition)

Python 3.8(Scrapy 2.4.0 + numpy 1.19.4 + pandas 1.1.4 + matplotlib 3.3.3)

1. **实验步骤及结果**
2. 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。

from matplotlib import pyplot as plt  
import numpy as np  
import pandas as pd  
  
from pylab import \*  
mpl.rcParams['font.sans-serif'] = ['SimHei']  
  
df = pd.read\_csv("./iris.csv", encoding='utf-8', dtype=str)  
df = pd.DataFrame(df, columns=['Sepal.length', 'Sepal.width', 'Petal.Length', 'Petal.Width', 'Species'], dtype=str)  
  
colors = ['blue', 'orange', 'green']  
  
df['Sepal.length'] = df['Sepal.length'].astype(np.float)  
df['Sepal.width'] = df['Sepal.width'].astype(np.float)  
df['Petal.Length'] = df['Petal.Length'].astype(np.float)  
df['Petal.Width'] = df['Petal.Width'].astype(np.float)  
  
Species = df.Species.unique()  
print(Species)  
  
fig, ax = plt.subplots()  
  
labelx = ['Petal.Width', 'Petal.Length', 'Sepal.width', 'Sepal.length']  
labely = ['Sepal.length', 'Sepal.width', 'Petal.Length', 'Petal.Width']  
  
for index in range(16):  
 plt.subplot(4, 4, index + 1)  
 plt.xlabel(labelx[index % 4])  
 plt.ylabel(labely[int(index / 4)])  
 plt.title(labelx[index % 4] + ' vs ' + labely[int(index / 4)])  
 plt.grid(True, linestyle='--', alpha=0.8)  
 for i in range(len(Species)):  
 plt.scatter(df.loc[df.Species == Species[i], labelx[index % 4]], df.loc[df.Species == Species[i], labely[int(index / 4)]], s=5, color=colors[i], label=Species[i])  
  
  
plt.subplots\_adjust(left=None, bottom=None, right=None, top=None,  
 wspace=1, hspace=1)  
plt.show()



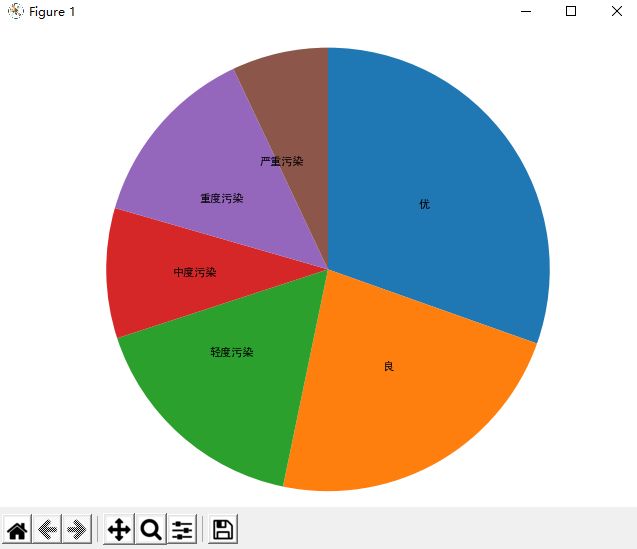
1. 找一组自己感兴趣的真实数据，绘制出饼图。并看看数据的项数在什么范围比较合适在饼图中展示；

调整数据的顺序或角度，使得第一个扇区在12点方向开始；

调整字体的大小、标签的位置等参数。

选用了之前北京2010-2015年PM值状况所对应的的空气质量等级进行了饼图展示，调整了字体大小、位置、饼图大小、每一块的顺序(顺逆时针)、初始块的角度等。

from matplotlib import pyplot as plt  
import numpy as np  
import pandas as pd  
  
from pylab import \*  
mpl.rcParams['font.sans-serif'] = ['SimHei']  
  
section = [0, 35, 75, 115, 150, 250, 99999]  
section\_name = ["优", "良", "轻度污染", "中度污染", "重度污染", "严重污染"]  
  
orig\_df = pd.read\_csv("./BeijingPM20100101\_20151231.csv", encoding='utf-8', dtype=str)  
orig\_df = pd.DataFrame(orig\_df, columns=['year', 'month', 'day', 'PM\_US Post'])  
df = orig\_df.dropna(0, how='any')  
  
df['PM\_US Post'] = df['PM\_US Post'].astype(np.int)  
  
result = pd.DataFrame(pd.cut(df['PM\_US Post'], section, labels=section\_name, ordered=False))  
result\_count = result['PM\_US Post'].value\_counts(sort=False)  
  
plt.pie(result\_count, labels=result\_count.index, startangle=90, counterclock=False, radius=1.5, textprops={'fontsize':8}, labeldistance=0.5)  
plt.show()

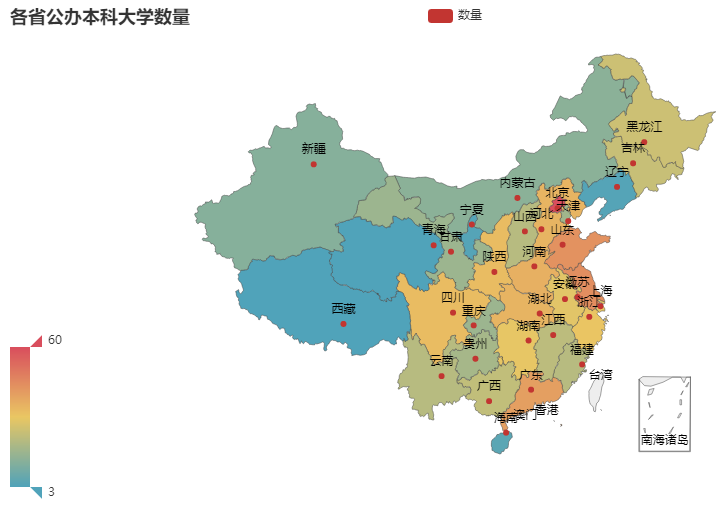


可以发现该饼图的严重污染部分已经发生了错位，因而饼图适合于5~6块，即最小块不要小于15%左右会比较好看。

1. 在中国地图上展示每个省的高考人数或大学数量

选择展示各省/市的公办本科大学数量。

from pyecharts import options as opts  
from pyecharts.charts import Map  
import numpy as np  
import pandas as pd  
import random  
  
df = pd.read\_csv("./中国大学数量.csv", encoding='utf-8', dtype=np.str)  
df = pd.DataFrame(df, columns=['省/市', '公办本科大学数量'])  
df['省/市'] = df['省/市'].astype(np.str)  
df['公办本科大学数量'] = df['公办本科大学数量'].astype(np.int)  
  
class Data:  
 @staticmethod  
 def province() -> list:  
 res = []  
 for i in range(len(df)):  
 res.append(str(df['省/市'][i]))  
 return res  
  
 @staticmethod  
 def values() -> list:  
 res = []  
 for i in range(len(df)):  
 res.append(int(df['公办本科大学数量'][i]))  
 return res  
  
  
def map2() -> Map:  
 c = (  
 Map()  
 .add("数量", [list(z) for z in zip(Data.province(), Data.values())], "china").set\_global\_opts(  
 title\_opts=opts.TitleOpts(title="各省公办本科大学数量"),  
 visualmap\_opts=opts.VisualMapOpts(min\_=int(df['公办本科大学数量'].min()), max\_=int(df['公办本科大学数量'].max())))  
 .set\_series\_opts(label\_opts=opts.LabelOpts(is\_show=True))  
 )  
 return c  
  
  
map2().render("map2.html")

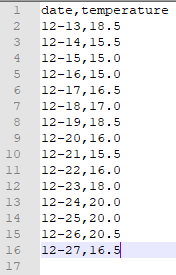


1. 展示自己家乡所在城市的温度变化热力图，要求至少有10天的数据。

选取深圳市作图，爬取如下网址未来十天深圳市的天气数据

https://www.tianqi.com/shenzhen/15/

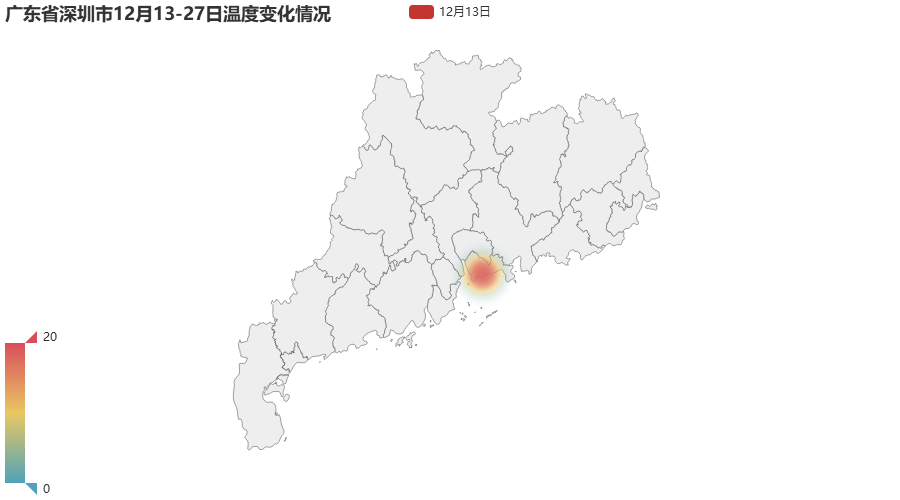
import scrapy  
from weather.items import WeatherItem  
  
class TianqiSpider(scrapy.Spider):  
 name = 'tianqi'  
 allowed\_domains = ['tianqi.com']  
 start\_urls = ['https://www.tianqi.com/shenzhen/15']  
  
 def parse(self, response):  
 item = WeatherItem()  
 data = response.xpath('//ul[@class="weaul"]/li')  
 for each in data:  
 item['date'] = each.xpath('a/div[1]/span[1]/text()').extract()  
 low = each.xpath('a/div[4]/span[1]/text()').extract()[0]  
 high = each.xpath('a/div[4]/span[2]/text()').extract()[0]  
 item['temperature'] = str((int(low) + int(high)) / 2)  
 yield item  
 pass



爬虫代码及结果如上

from pyecharts import options as opts  
from pyecharts.charts import Geo  
from pyecharts.globals import ChartType  
from pyecharts.render import make\_snapshot  
from snapshot\_phantomjs import snapshot  
import pandas as pd  
import numpy as np  
  
df = pd.read\_csv("./weather.csv", encoding='utf-8', dtype=np.str)  
df = pd.DataFrame(df, columns=['temperature'])  
df['temperature'] = df['temperature'].astype(np.float)  
day = 0  
  
class Data:  
 guangdong\_city = ["深圳市"]  
  
 @staticmethod  
 def values() -> list:  
 res = []  
 for i in range(len(df)):  
 res.append(float(df['temperature'][i]))  
 return res  
  
def geo\_guangdong(title) -> Geo:  
 c = (  
 Geo()  
 .add\_schema(maptype="广东")  
 .add(  
 title,  
 [list(z) for z in zip(Data.guangdong\_city, [Data.values()[day]])],  
 type\_=ChartType.HEATMAP,  
 )  
 .set\_global\_opts(  
 visualmap\_opts=opts.VisualMapOpts(min\_=0, max\_=int(df['temperature'].max())),#is\_piecewise=True),  
 title\_opts=opts.TitleOpts(title="广东省深圳市12月13-27日温度变化情况"),  
 )  
 )  
 return c  
  
for i in range(10):  
 day = i  
 str\_date="12月" + str(i+13) + "日"  
 make\_snapshot(snapshot, geo\_guangdong(str\_date).render(),  
 str(i+1)+".png", pixel\_ratio=1)

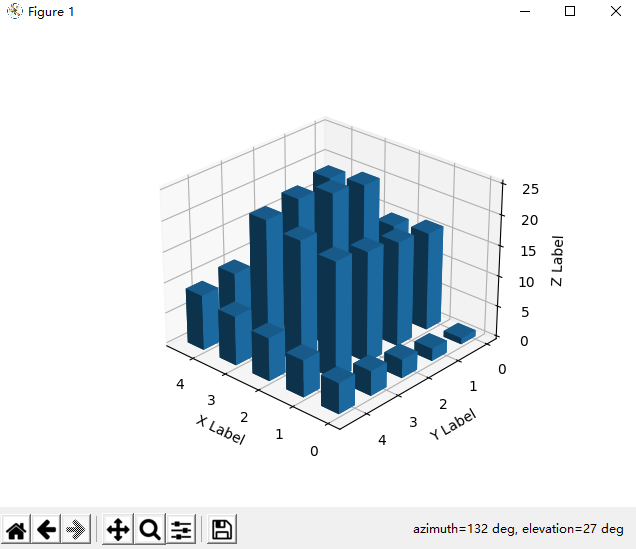
生成10天热力图代码如上。



其中一张如上，其余另附。

1. 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

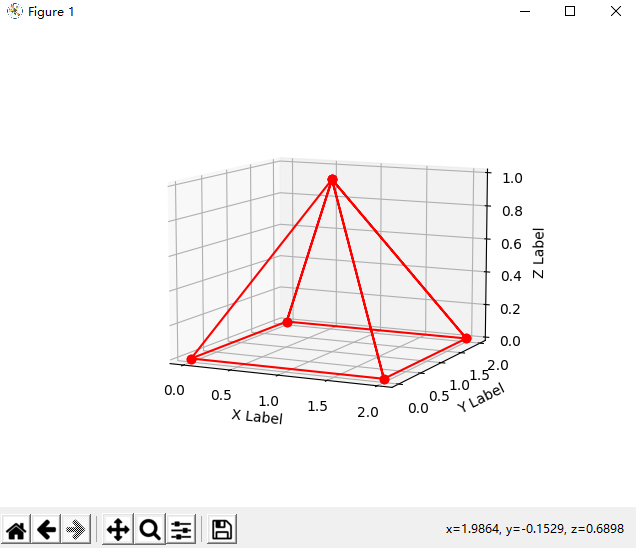
import numpy as np  
import matplotlib.pyplot as plt  
import math  
from mpl\_toolkits.mplot3d import Axes3D  
  
#1.生成fig对象和ax对象  
fig = plt.figure()  
ax = fig.add\_subplot(projection='3d')  
ax.set\_xlabel('X Label')  
ax.set\_ylabel('Y Label')  
ax.set\_zlabel('Z Label')  
  
#2.生成数据  
x = []  
y = []  
z = [i + 1 for i in range(25)]  
  
left = up = -1  
down = right = 5  
px = py = 0  
x.append(px)  
y.append(py)  
  
def add(px, py):  
 x.append(px)  
 y.append(py)  
  
while True:  
 if px == 2 and py == 2:  
 break  
 while py + 1 < down:  
 py = py + 1  
 add(px, py)  
 left = left + 1  
 while px + 1 < right:  
 px = px + 1  
 add(px, py)  
 down = down - 1  
 while py - 1 > up:  
 py = py - 1  
 add(px, py)  
 right = right - 1  
 while px - 1 > left:  
 px = px - 1  
 add(px, py)  
 up = up + 1  
  
bottom = [0 for \_ in range(25)]  
width = depth = 0.5  
  
#3.调用bar3d，画3D直方图  
ax.bar3d(x, y, bottom, width, depth, z, shade=True)  
  
#4.显示图形  
plt.show()

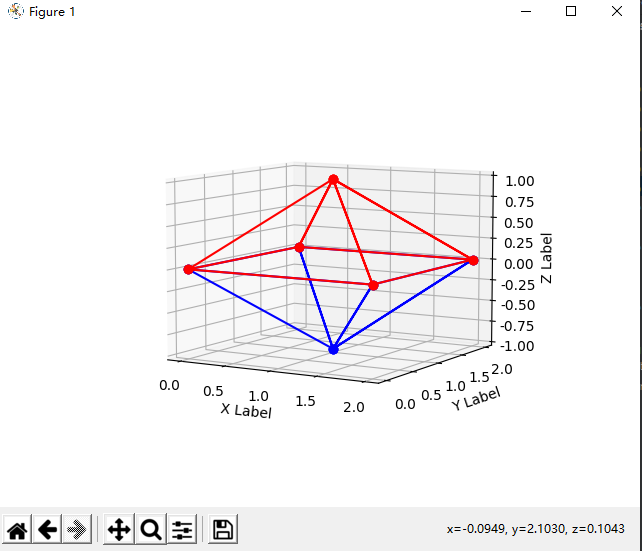


1. 生成一个金字塔的线图；

生成一上一下两个金字塔，叠放在一起。

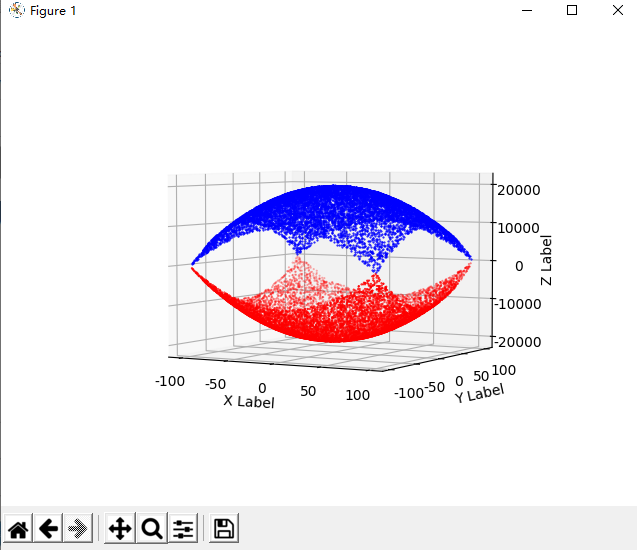
import numpy as np  
import matplotlib.pyplot as plt  
import math  
from mpl\_toolkits.mplot3d import Axes3D  
  
#1.生成fig对象和ax对象  
fig = plt.figure()  
ax = fig.add\_subplot(projection='3d')  
ax.set\_xlabel('X Label')  
ax.set\_ylabel('Y Label')  
ax.set\_zlabel('Z Label')  
  
#2.生成数据  
x = np.array([1,0,2,1,2,2,1,2,0,1,0,0])  
y = np.array([1,0,0,1,0,2,1,2,2,1,2,0])  
z = np.array([1,0,0,1,0,0,1,0,0,1,0,0])  
z\_minus = -1 \* z;  
  
#3.调用plot，画3D的线图  
ax.plot(x,y,z\_minus,"b",marker='o')  
ax.plot(x,y,z,"r",marker='o')  
  
#4.显示图形  
plt.show()





1. 生成一个散点图，z=x^2+y^2

import numpy as np  
import matplotlib.pyplot as plt  
import random  
import math  
from mpl\_toolkits.mplot3d import Axes3D  
  
#1.生成fig对象和ax对象  
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.生成数据  
x = np.linspace(-100,100,10000)  
np.random.shuffle(x)  
y = np.linspace(-100,100,10000)  
np.random.shuffle(y)  
z = x\*\*2 + y\*\*2 - 20000  
z\_minus = -1 \* z  
ax.scatter(x, y, z, zdir='z', s=1, c='r', marker='.', depthshade=True)  
ax.scatter(x, y, z\_minus, zdir='z', s=1, c='b', marker='.', depthshade=True)  
  
#4.显示图形  
plt.show()



1. **心得与体会**

在本次实验中，通过几道题目的引导，逐步掌握了python中使用matplotlib绘制二维散点图、饼图，绘制三维的直方图、线图、散点图等，以及利用pyecharts在地图上绘制各种图标。非常感谢老师的实验设计！