

**《Python程序设计》课程综合实验**

|  |  |
| --- | --- |
| **题目:** | **弹幕爬取及词云图设计** |
| **专业：** | **数字媒体技术** |
| **班级：** | **20级1班** |
| **学号：** | **207020111** |
| **姓名：** | **王睿琪** |
| **指导老师：** | **杜萌** |

**日期：2021年6月16日**

目录

* 1. 项目背景
  2. 实验配置
  3. 实验步骤
  4. 遇到的问题和解决
  5. 实验结论
  6. 后期展望
  7. 程序源代码

一、项目背景

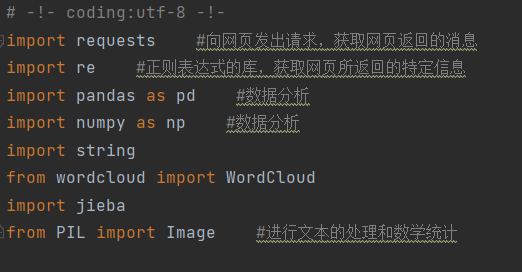
今年是建党100周年，《少年》这首歌曲也是我非常喜欢的歌曲，而这个视频将这两个非常完美的结合在了一起，同时也想让自己的第一次爬虫成果更加有意义。所以选择了这个视频。

二、实验配置

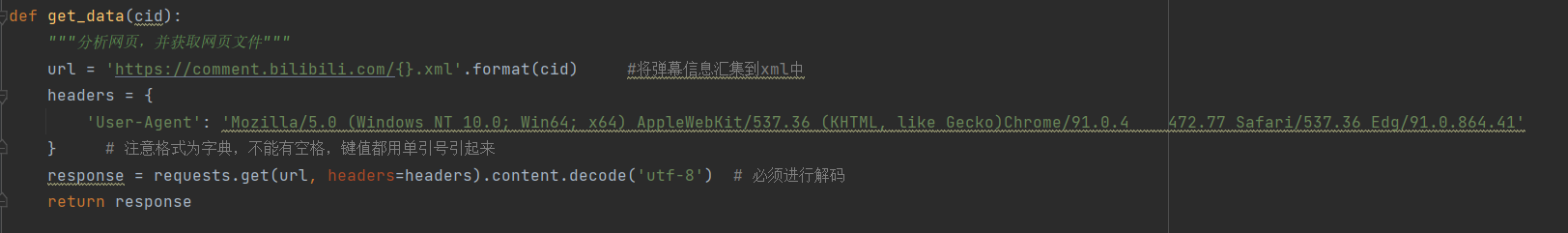
Pycharm2021.1.1软件

三、实验步骤

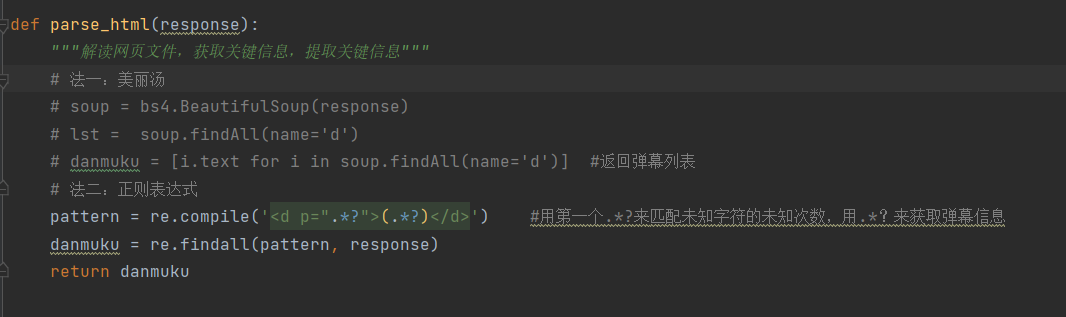
1、用到的库



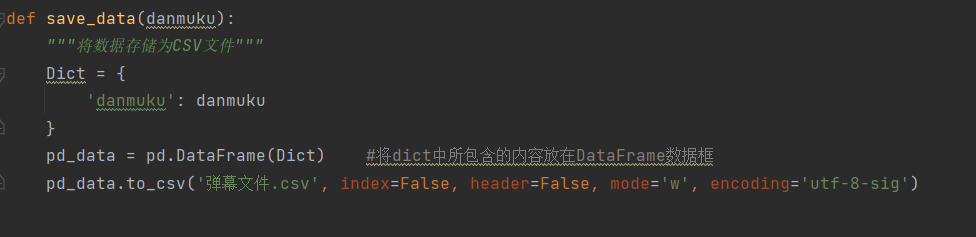
2、分析网页，获取网页文件，汇总弹幕信息



3、用正则表达式来解读网页文件，获取并提取关键信息



4、将文件储存为CSV文件



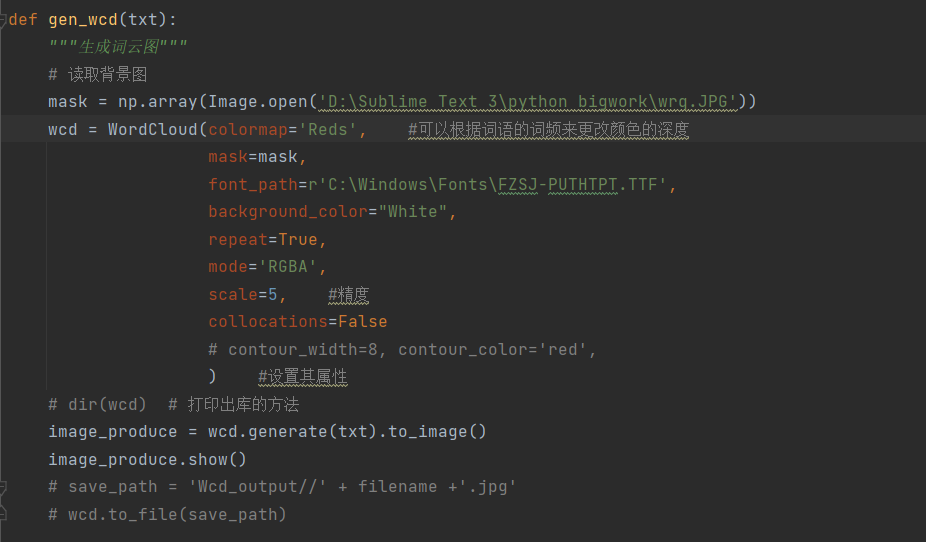
5、进行数据去重、分类聚类和词云图可视化



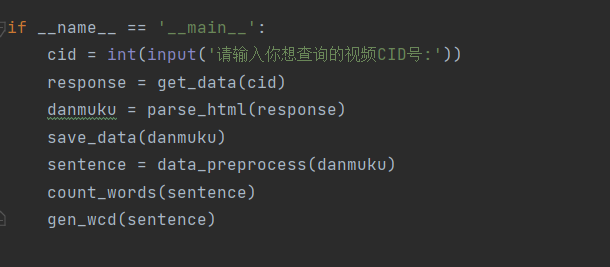
6、词频统计



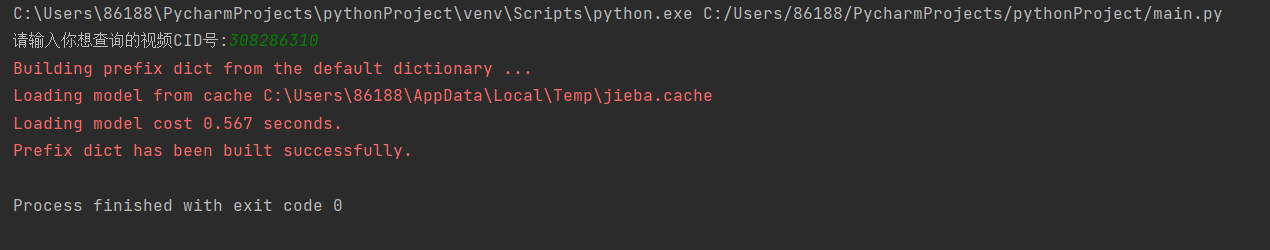
7、生成词云图

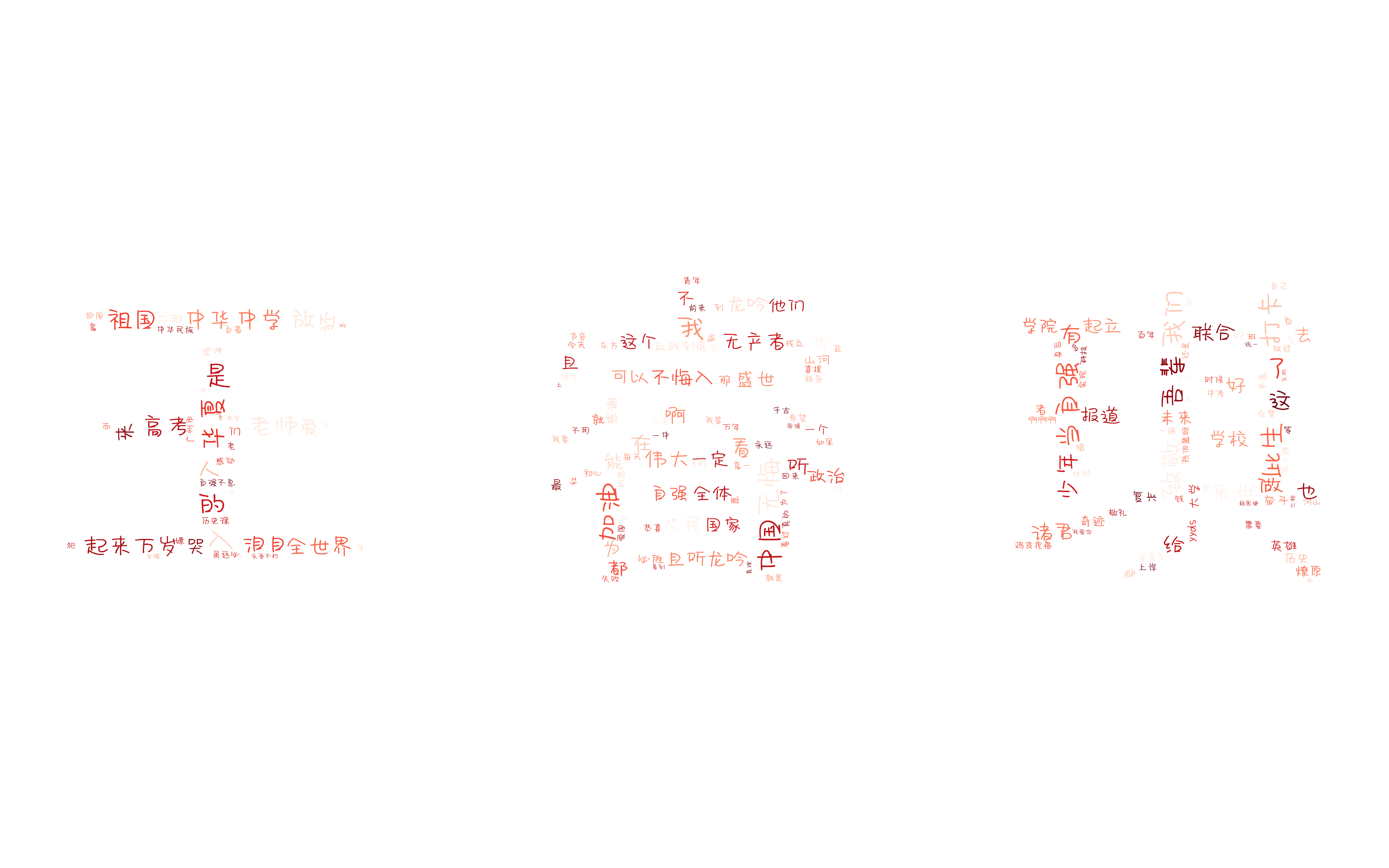


8、函数调用



9、运行程序，得到词云图



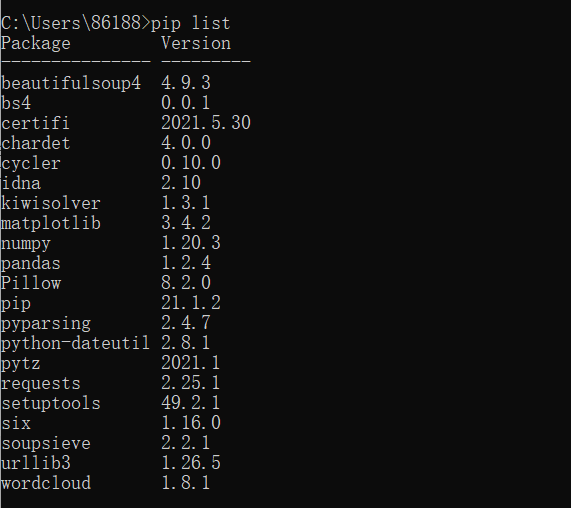


四、遇到的问题和解决

问题1：wordcloud模块的安装问题

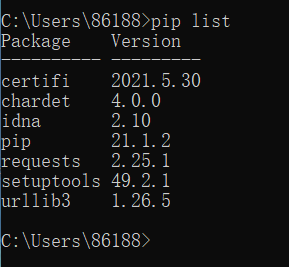
解决方式：用百度搜索

先下载一个visual studio，再在cmd中输入pip install wordcloud,再输入pip list查看是否成功安装，再在pycharm中安装一次就可以了。



问题2：import requests模块安装出现问题;

Pip list显示我的requests已经安装

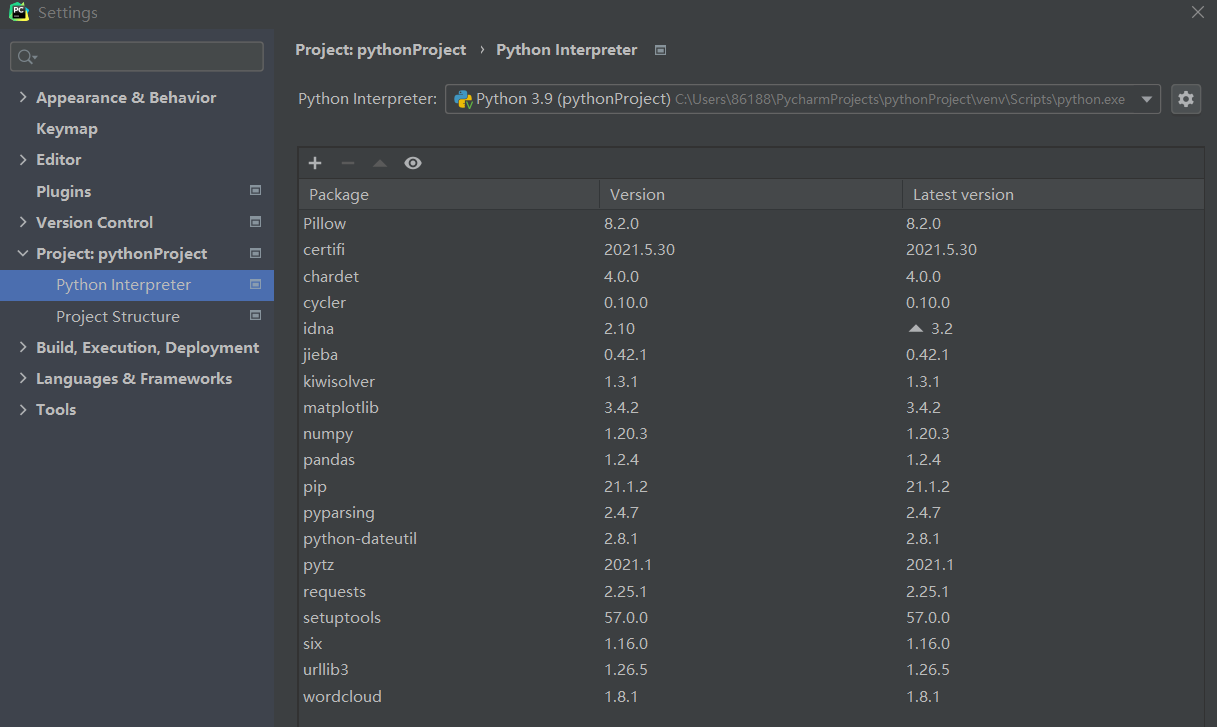


但运行后报错： ModuleNotFoundError: No module named 'requests'

解决方式：用百度搜索

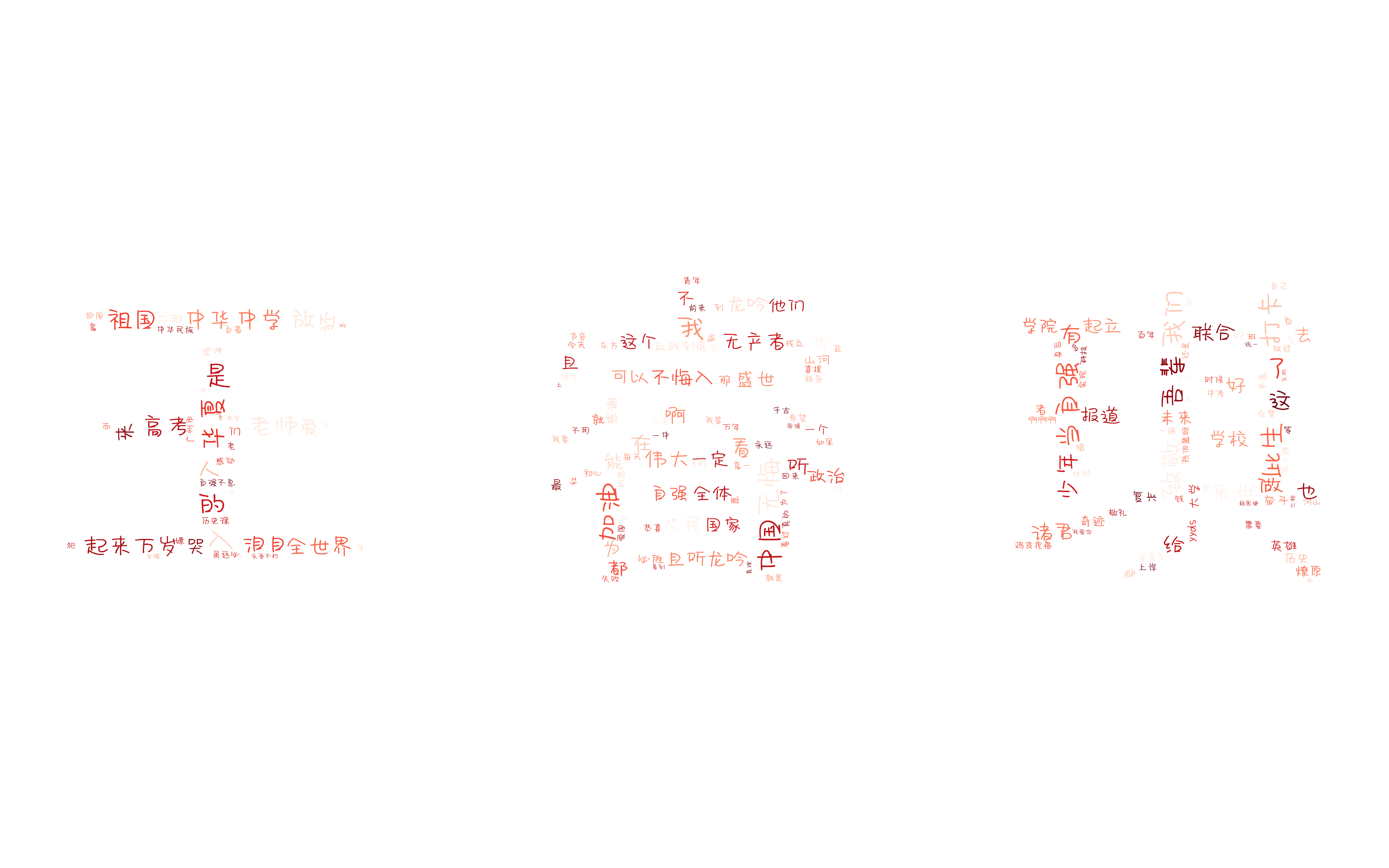
**依次点击File->Settings，打开Settings对话框，再打开Python Interpreter点‘+’，搜索requests，进行加载。**

**下载成功后将会在package中显示：**

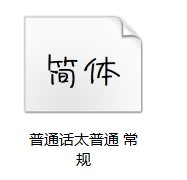


五、实验结论

1、词云图：



2、个性字体：C:\Windows\Fonts\FZSJ-PUTHTPT.TTF



六、后期展望

2020年，地球寂静的这一年里，空气质量发生巨大的改变，我可以用python爬取世界各地的近三年空气质量指数数据进行分析比较，用折线图的形式呈现。

七、程序源代码

# -!- coding:utf-8 -!-  
import requests #向网页发出请求，获取网页返回的消息  
import re #正则表达式的库，获取网页所返回的特定信息  
import pandas as pd #数据分析  
import numpy as np #数据分析  
import string  
from wordcloud import WordCloud  
import jieba  
from PIL import Image #进行文本的处理和数学统计  
  
  
def get\_data(cid):  
 *"""分析网页，并获取网页文件"""* url = 'https://comment.bilibili.com/{}.xml'.format(cid) #将弹幕信息汇集到xml中  
 headers = {  
 'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)Chrome/91.0.4 472.77 Safari/537.36 Edg/91.0.864.41'  
 } # 注意格式为字典，不能有空格，键值都用单引号引起来  
 response = requests.get(url, headers=headers).content.decode('utf-8') # 必须进行解码  
 return response  
  
  
def parse\_html(response):  
 *"""解读网页文件，获取关键信息，提取关键信息"""* # 法一：美丽汤  
 # soup = bs4.BeautifulSoup(response)  
 # lst = soup.findAll(name='d')  
 # danmuku = [i.text for i in soup.findAll(name='d')] #返回弹幕列表  
 # 法二：正则表达式  
 pattern = re.compile('<d p=".\*?">(.\*?)</d>') #用第一个.\*?来匹配未知字符的未知次数，用.\*？来获取弹幕信息  
 danmuku = re.findall(pattern, response)  
 return danmuku  
  
  
def save\_data(danmuku):  
 *"""将数据存储为CSV文件"""* Dict = {  
 'danmuku': danmuku  
 }  
 pd\_data = pd.DataFrame(Dict) #将dict中所包含的内容放在DataFrame数据框  
 pd\_data.to\_csv('弹幕文件.csv', index=False, header=False, mode='w', encoding='utf-8-sig')  
  
  
def data\_preprocess(danmuku):  
 # 数据去重  
 # 大小写替换 → 去标点符号、空白 → 去停用词 → 词干化，  
 # 分词 → 词频统计 → 构建 bag of words → 分类聚类  
 # 可视化 （词云图，统计分析图表）  
 with open('弹幕文件.csv', mode='r', encoding='utf-8') as f:  
 #加载用户自定义词典  
 jieba.load\_userdict(r'C:\Users\86188\PycharmProjects\pythonProject\urserdict.txt') # 导入自定义字符串  
 reader = f.read().replace('\n', '')  
 # 加载停用词表  
 stopwords = [line.strip() for line in open('stop\_wordslst.txt', encoding='utf-8').readlines()]  
 # 去掉所有的标点、空白和数字  
 pun\_num = string.punctuation + string.digits  
 table = str.maketrans('', '', pun\_num)  
 reader = reader.translate(table)  
 seg\_list = jieba.cut(reader, cut\_all=False) # 生成一个迭代器，不精确的分词模式  
 sentence = ''  
 for word in seg\_list:  
 if word not in stopwords and word.isspace() == False:  
 sentence += word  
 sentence += ','  
 sentece = sentence[:-1]  
 # sentence = ','.join(seg\_list) #生成字符串序列，用于词频统计和词云图制作  
 return sentence  
  
  
def count\_words(txt):  
 #词频统计  
 aDict = {} # 创建一个字典用来收纳词语-频次的键值对  
 words = txt.split(',')  
 for word in words:  
 aDict[word] = aDict.get(word, 0) + 1  
 # 显示所有列  
 pd.set\_option('display.max\_columns', None)  
 # 显示所有行  
 pd.set\_option('display.max\_rows', None)  
 pd\_count = pd.DataFrame(aDict, index=['times']).T.sort\_values('times', ascending=False)  
 pd\_count.to\_csv('count\_word.csv', encoding="utf-8")  
 # 直接将标称属性为value的字典转成dataframe时，需要设定index（不要让标量成为值，将其序列化）  
 # items = sorted(list(aDict.items()), key=lambda x: x[1], reverse=True)  
 # return items  
  
  
def gen\_wcd(txt):  
 *"""生成词云图"""* # 读取背景图  
 mask = np.array(Image.open('D:\Sublime Text 3\python bigwork\wrq.JPG'))  
 wcd = WordCloud(colormap='Reds', #可以根据词语的词频来更改颜色的深度  
 mask=mask,  
 font\_path=r'C:\Windows\Fonts\FZSJ-PUTHTPT.TTF',  
 background\_color="White",  
 repeat=True,  
 mode='RGBA',  
 scale=5, #精度  
 collocations=False  
 # contour\_width=8, contour\_color='red',  
 ) #设置其属性  
 # dir(wcd) # 打印出库的方法  
 image\_produce = wcd.generate(txt).to\_image()  
 image\_produce.show()  
 # save\_path = 'Wcd\_output//' + filename +'.jpg'  
 # wcd.to\_file(save\_path)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 cid = int(input('请输入你想查询的视频CID号:'))  
 response = get\_data(cid)  
 danmuku = parse\_html(response)  
 save\_data(danmuku)  
 sentence = data\_preprocess(danmuku)  
 count\_words(sentence)  
 gen\_wcd(sentence)