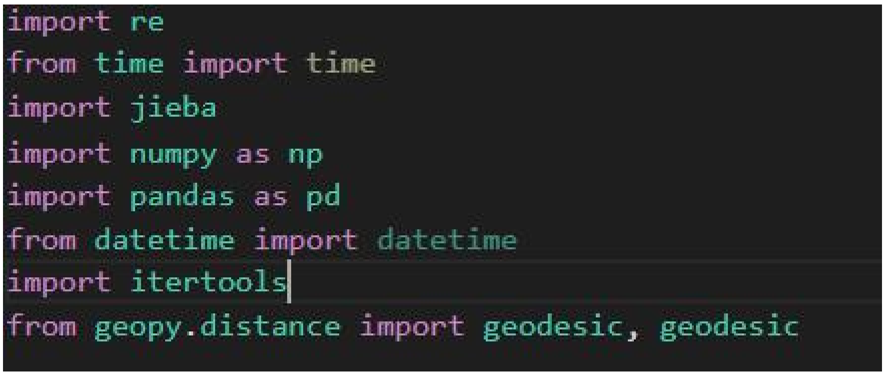
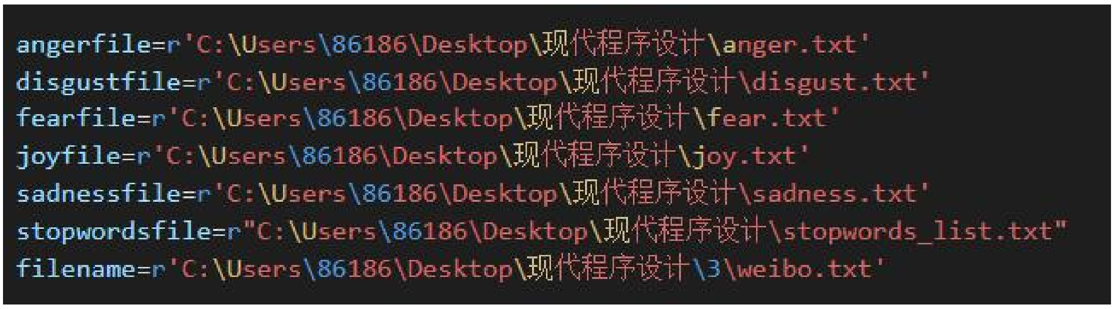
初始化：

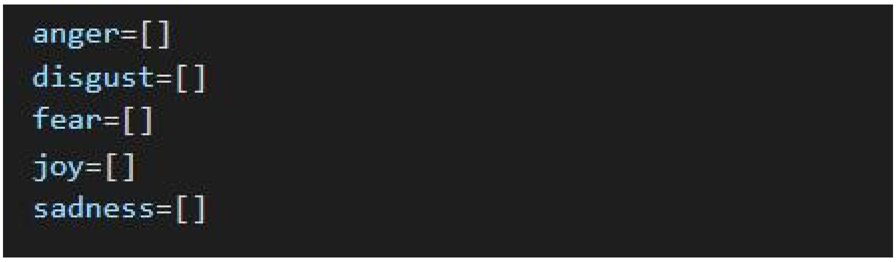


```python
import re
from time import time
import jieba
import numpy as np
import pandas as pd
from datetime import datetime
import itertools
from geopy.distance import geodesic, geodesic
```

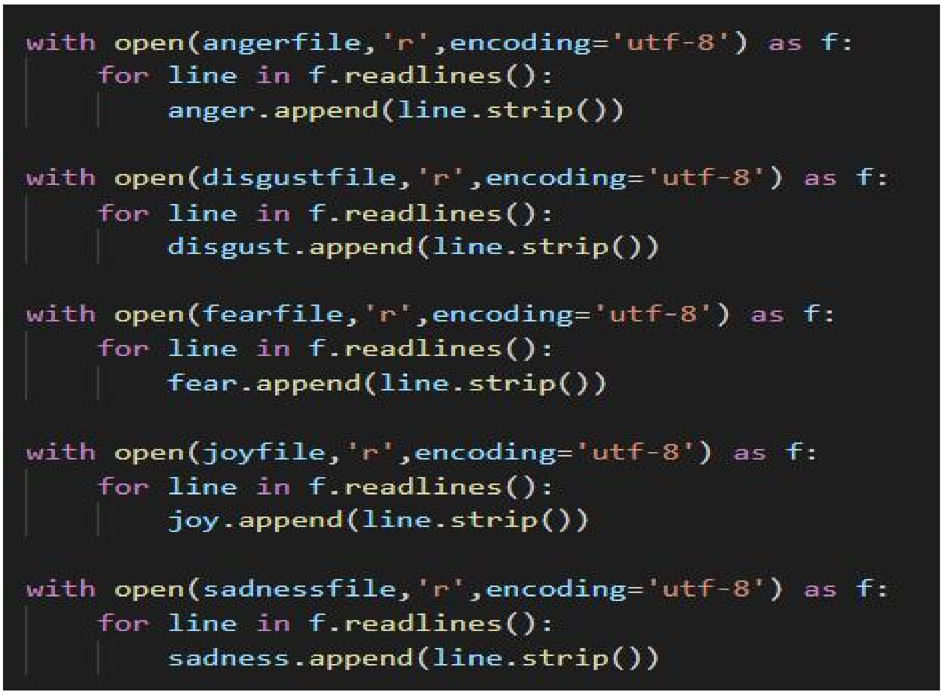


```python
angerfile=r'C:\Users\86186\Desktop\现代程序设计\anger.txt'
disgustfile=r'C:\Users\86186\Desktop\现代程序设计\disgust.txt'
fearfile=r'C:\Users\86186\Desktop\现代程序设计\fear.txt'
joyfile=r'C:\Users\86186\Desktop\现代程序设计\joy.txt'
sadnessfile=r'C:\Users\86186\Desktop\现代程序设计\sadness.txt'
stopwordsfile=r"C:\Users\86186\Desktop\现代程序设计\stopwords_list.txt"
filename=r'C:\Users\86186\Desktop\现代程序设计\3\weibo.txt'
```

创建情绪词典：



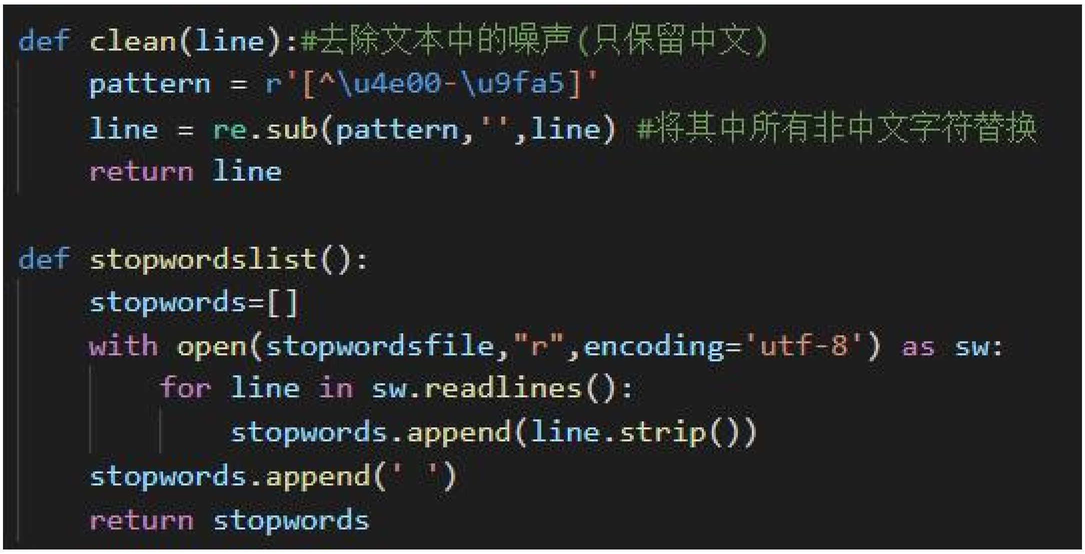
```python
anger=[]
disgust=[]
fear=[]
joy=[]
sadness=[]
```



```python
with open(angerfile,'r',encoding='utf-8') as f:
    for line in f.readlines():
        anger.append(line.strip())

with open(disgustfile,'r',encoding='utf-8') as f:
    for line in f.readlines():
        disgust.append(line.strip())

with open(fearfile,'r',encoding='utf-8') as f:
    for line in f.readlines():
        fear.append(line.strip())

with open(joyfile,'r',encoding='utf-8') as f:
    for line in f.readlines():
        joy.append(line.strip())

with open(sadnessfile,'r',encoding='utf-8') as f:
    for line in f.readlines():
        sadness.append(line.strip())
```

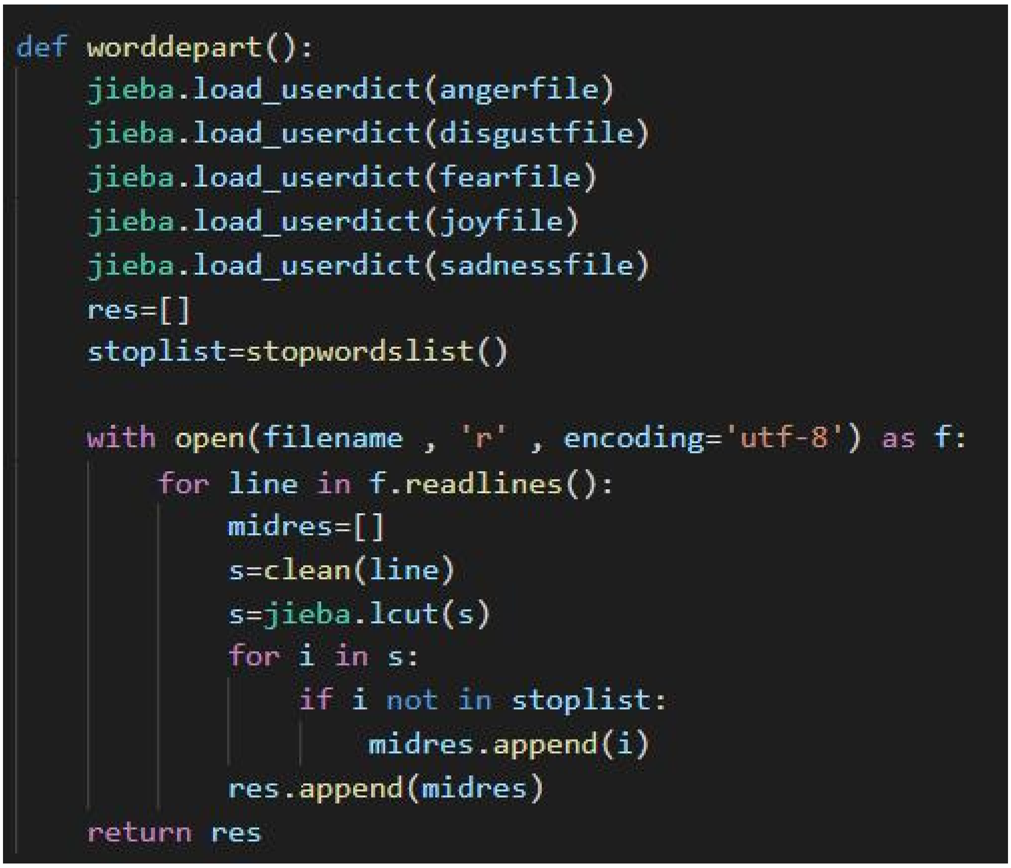
1. 实现一个函数，对微博数据进行清洗，去除噪声（如 url 等），过滤停用词。注意分词的时候应该将情绪词典加入 Jieba 的自定义词典，以提高这些情绪词的识别能力。

创建停用词列表，并且事先实现去除噪声的功能，这步只考虑分词，过去除噪声只保留中文



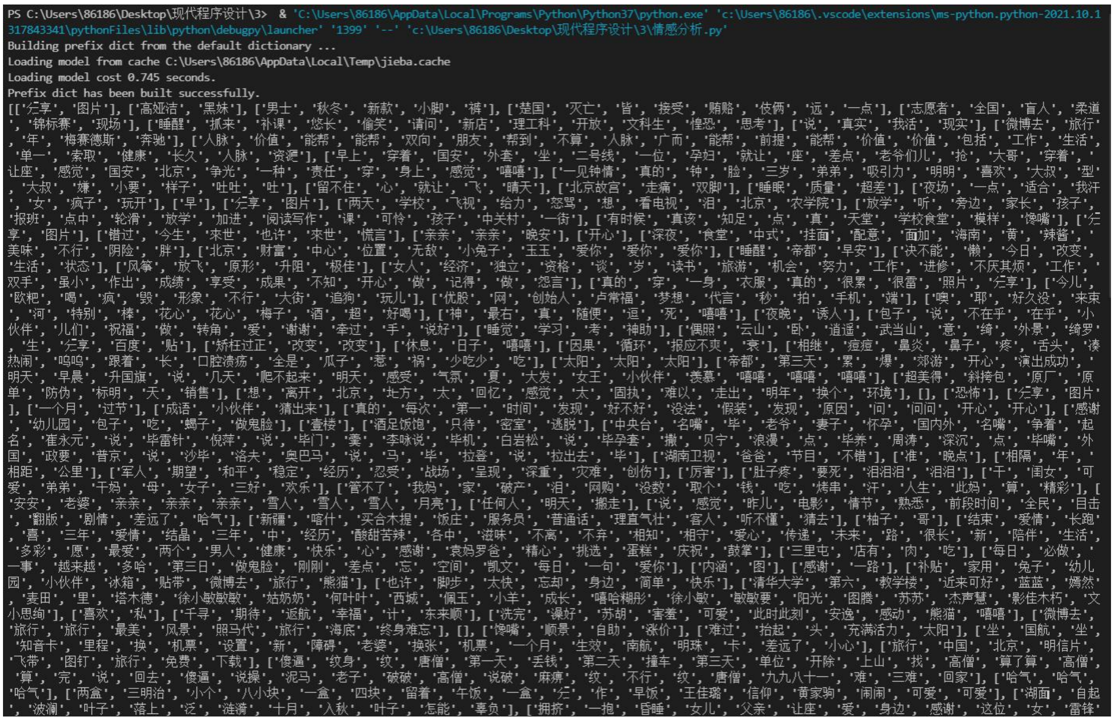
```python
def clean(line):#去除文本中的噪声(只保留中文)
    pattern = r'[^\u4e00-\u9fa5]'
    line = re.sub(pattern,'',line) #将其中所有非中文字符替换
    return line

def stopwordslist():
    stopwords=[]
    with open(stopwordsfile,"r",encoding='utf-8') as sw:
        for line in sw.readlines():
            stopwords.append(line.strip())
    stopwords.append(' ')
    return stopwords
```

实现分词的函数：



```python
def worddepart():
    jieba.load_userdict(angerfile)
    jieba.load_userdict(disgustfile)
    jieba.load_userdict(fearfile)
    jieba.load_userdict(joyfile)
    jieba.load_userdict(sadnessfile)
    res=[]
    stoplist=stopwordslist()

    with open(filename , 'r' , encoding='utf-8') as f:
        for line in f.readlines():
            midres=[]
            s=clean(line)
            s=jieba.lcut(s)
            for i in s:
                if i not in stoplist:
                    midres.append(i)
            res.append(midres)
    return res
```

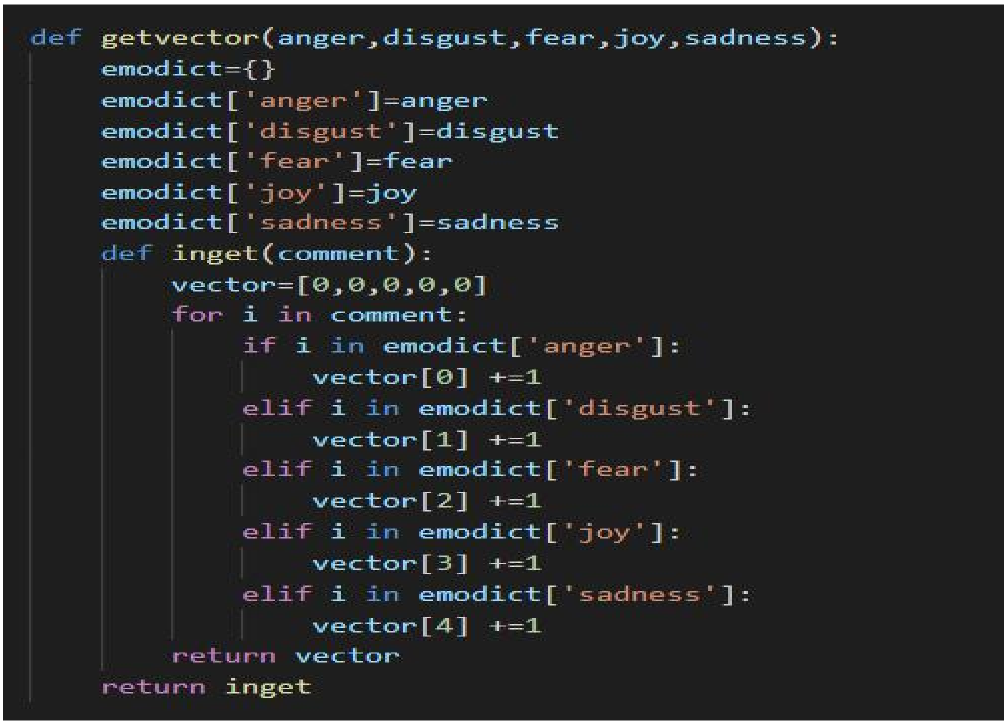
运行结果：



```
PS C:\Users\86186\Desktop\现代程序设计\3>  & 'C:\Users\86186\AppData\Local\Programs\Python\Python37\python.exe' 'c:\Users\86186\.vscode\extensions\ms-python.python-2021.10.1
317843341\pythonFiles\lib\python\debugpy\launcher' '1399' '--' 'c:\Users\86186\Desktop\现代程序设计\3\情感分析.py'
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\86186\AppData\Local\Temp\jieba.cache
Loading model cost 0.745 seconds.
Prefix dict has been built successfully.
[['分享', '图片'], ['高娅洁', '黑妹'], ['男士', '秋冬', '新款', '小脚', '裤'], ['楚国', '灭亡', '皆', '接受', '贿赂', '伎俩', '远', '一点'], ['志愿者', '全国', '盲人', '柔道
', '锦标赛', '现场'], ['睡醒', '抓来', '补课', '悠长', '偷笑', '请问', '新店', '理工科', '开放', '文科生', '惶恐', '思考'], ['说', '真实', '我活', '现实'], ['微博去', '旅行'
, '年', '梅赛德斯', '奔驰'], ['人脉', '价值', '能帮', '能帮', '双向', '朋友', '帮到', '不算', '人脉', '广而', '能帮', '前提', '能帮', '价值', '价值', '包括', '工作', '生活',
 '单一', '索取', '健康', '长久', '人脉', '资源'], ['早上', '穿着', '国安', '外套', '坐', '二号线', '一位', '孕妇', '就让', '座', '差点', '老爷们儿', '抢', '大哥', '穿着', '
让座', '感觉', '国安', '北京', '争光', '一种', '责任', '穿', '身上', '感觉', '嘻嘻'], ['一见钟情', '真的', '钟', '脸', '三岁', '弟弟', '吸引力', '明明', '喜欢', '大叔', '型'
, '大叔', '嫌', '小要', '样子', '吐吐', '吐'], ['留不住', '心', '就让', '飞', '晴天'], ['北京故宫', '走痛', '双脚'], ['睡眠', '质量', '超差'], ['夜场', '一点', '适合', '我汗
', '女', '疯子', '玩开'], ['早'], ['分享', '图片'], ['两天', '学校', '飞视', '给力', '怒骂', '想', '看电视', '泪', '北京', '农学院'], ['放学', '听', '旁边', '家长', '孩子',
'报班', '点中', '轮滑', '放学', '加进', '阅读写作', '课', '可怜', '孩子', '中关村', '一街'], ['有时候', '真该', '知足', '点', '真', '天堂', '学校食堂', '模样', '馋嘴'], ['分
享', '图片'], ['错过', '今生', '来世', '也许', '来世', '慌言'], ['亲亲', '亲亲', '晚安'], ['开心'], ['深夜', '食堂', '中式', '挂面', '配意', '面加', '海南', '黄', '辣酱', '
美味', '不行', '阴险', '胖'], ['北京', '财富', '中心', '位置', '无敌', '小兔子', '玉玉', '爱你', '爱你', '爱你'], ['睡醒', '帝都', '早安'], ['决不能', '懒', '今日', '改变',
'生活', '状态'], ['风筝', '放飞', '原形', '升阻', '极佳'], ['女人', '经济', '独立', '资格', '谈', '岁', '读书', '旅游', '机会', '努力', '工作', '进修', '不厌其烦', '工作', '
双手', '虽小', '作出', '成绩', '享受', '成果', '不知', '开心', '做', '记得', '做', '怨言'], ['真的', '穿', '一身', '衣服', '真的', '很累', '很雷', '照片', '分享'], ['今儿',
'欧粑', '喝', '疯', '毁', '形象', '不行', '大街', '追狗', '玩儿'], ['优股', '网', '创始人', '卢常福', '梦想', '代言', '秒', '拍', '手机', '端'], ['噢', '耶', '好久没', '来束
', '河', '特别', '棒', '花心', '花心', '梅子', '酒', '超', '好喝'], ['神', '最右', '真', '随便', '逗', '死', '嘻嘻'], ['夜晚', '诱人'], ['包子', '说', '不在乎', '在乎', '小
伙伴', '儿们', '祝福', '做', '转角', '爱', '谢谢', '牵过', '手', '说好'], ['睡觉', '学习', '考', '神助'], ['偶照', '云山', '卧', '逍遥', '武当山', '意', '绮', '外景', '绮罗
', '生', '分享', '百度', '贴'], ['矫枉过正', '改变', '改变'], ['休息', '日子', '嘻嘻'], ['因果', '循环', '报应不爽', '衰'], ['相继', '痘痘', '鼻炎', '鼻子', '疼', '舌头', '凑
热闹', '呜呜', '跟着', '长', '口腔溃疡', '全是', '瓜子', '惹', '祸', '少吃少', '吃'], ['太阳', '太阳', '太阳'], ['帝都', '第三天', '累', '爆', '郊游', '开心', '演出成功', '
明天', '早晨', '升国旗', '说', '几天', '爬不起来', '明天', '感受', '气氛', '夏', '大发', '女王', '小伙伴', '羡慕', '嘻嘻', '嘻嘻', '嘻嘻'], ['超美得', '斜挎包', '原厂', '原
单', '防伪', '标明', '天', '销售'], ['想', '离开', '北京', '北方', '太', '回忆', '感觉', '太', '固执', '难以', '走出', '明年', '换个', '环境'], [], ['恐怖'], ['分享', '图片
'], ['一个月', '过节'], ['成语', '小伙伴', '猜出来'], ['真的', '每次', '第一', '时间', '发现', '好不好', '没法', '假装', '发现', '原因', '问', '问问', '开心', '开心'], ['感谢
', '幼儿园', '包子', '吃', '蝎子', '做鬼脸'], ['壹楼'], ['酒足饭饱', '只待', '密室', '逃脱'], ['中央台', '名嘴', '毕', '老爷', '妻子', '怀孕', '国内外', '名嘴', '争着', '起
名', '崔永元', '说', '毕雷针', '倪萍', '说', '毕门', '羹', '李咏说', '毕机', '白岩松', '说', '毕孕套', '撒', '贝宁', '浪漫', '点', '毕养', '周涛', '深沉', '点', '毕嘴', '外
国', '政要', '普京', '说', '沙毕', '洛夫', '奥巴马', '说', '马', '毕', '拉登', '说', '拉出去', '毕'], ['湖南卫视', '爸爸', '节目', '不错'], ['准', '晚点'], ['相隔', '年', '
相距', '公里'], ['军人', '期望', '和平', '稳定', '经历', '忍受', '战场', '呈现', '深重', '灾难', '创伤'], ['厉害'], ['肚子疼', '要死', '泪泪泪', '泪泪'], ['干', '闺女', '可
爱', '弟弟', '干妈', '母', '女子', '三好', '欢乐'], ['管不了', '我妈', '家', '破产', '泪', '网购', '没数', '取个', '钱', '吃', '烤串', '汗', '人生', '此妈', '算', '精彩'], [
'安安', '老婆', '亲亲', '亲亲', '亲亲', '雪人', '雪人', '雪人', '月亮'], ['任何人', '明天', '搬走'], ['说', '感觉', '昨儿', '电影', '情节', '熟悉', '前段时间', '全民', '目击
', '翻版', '剧情', '差远了', '哈气'], ['新疆', '喀什', '买合木提', '饭庄', '服务员', '普通话', '理直气壮', '客人', '听不懂', '猜去'], ['柚子', '哥'], ['结束', '爱情', '长跑
', '喜', '三年', '爱情', '结晶', '三年', '中', '经历', '酸甜苦辣', '各中', '滋味', '不离', '不弃', '相知', '相守', '爱心', '传递', '未来', '路', '很长', '新', '陪伴', '生活',
 '多彩', '愿', '最爱', '两个', '男人', '健康', '快乐', '心', '感谢', '袁妈罗爸', '精心', '挑选', '蛋糕', '庆祝', '鼓掌'], ['三里屯', '店有', '肉', '吃'], ['每日', '必做', '
一事', '越来越', '多哈', '第三日', '做鬼脸', '刚刚', '差点', '忘', '空间', '凯文', '每日', '一句', '爱你'], ['内涵', '图'], ['感谢', '一路'], ['补贴', '家用', '兔子', '幼儿
园', '小伙伴', '冰箱', '贴带', '微博去', '旅行', '熊猫'], ['也许', '脚步', '太快', '忘却', '身边', '简单', '快乐'], ['清华大学', '第六', '教学楼', '近来可好', '蓝蓝', '嫣然
', '麦田', '里', '塔木德', '徐小敏敏敏', '姑奶奶', '何叶叶', '西城', '佩玉', '小羊', '成长', '嘻哈糊彤', '徐小敏', '敏敏要', '阳光', '图腾', '苏苏', '杰声慧', '影佳木朽', '文
小思绚'], ['喜欢', '私'], ['千寻', '期待', '返航', '幸福', '计', '东来顺'], ['洗完', '澡好', '苏胡', '害羞', '可爱', '此时此刻', '安逸', '感动', '熊猫', '嘻嘻'], ['微博去',
'旅行', '旅行', '最美', '风景', '照马代', '旅行', '海底', '终身难忘'], [], ['馋嘴', '顺景', '自助', '涨价'], ['难过', '抬起', '头', '充满活力', '太阳'], ['坐', '国航', '坐',
 '知音卡', '里程', '换', '机票', '设置', '新', '障碍', '老婆', '换张', '机票', '一个月', '生效', '南航', '明珠', '卡', '差远了', '小心'], ['旅行', '中国', '北京', '明信片',
'飞带', '图钉', '旅行', '免费', '下载'], ['傻逼', '纹身', '纹', '唐僧', '第一天', '丢钱', '第二天', '撞车', '第三天', '单位', '开除', '上山', '找', '高僧', '算了算', '高僧',
 '算', '完', '说', '回去', '傻逼', '说操', '泥马', '老子', '破破', '高僧', '说破', '麻痹', '纹', '不行', '纹', '唐僧', '九九八十一', '难', '三难', '回家'], ['哈气', '哈气',
'哈气'], ['两盒', '三明治', '小个', '八小块', '一盒', '四块', '留着', '午饭', '一盒', '分', '作', '早饭', '王佳璐', '信仰', '黄家驹', '闹闹', '可爱', '可爱'], ['湖面', '自起
', '波澜', '叶子', '落上', '泛', '涟漪', '十月', '入秋', '叶子', '怎能', '辜负'], ['拥挤', '一抱', '昏睡', '女儿', '父亲', '让座', '爱', '身边', '感谢', '这位', '女', '雷锋'
```

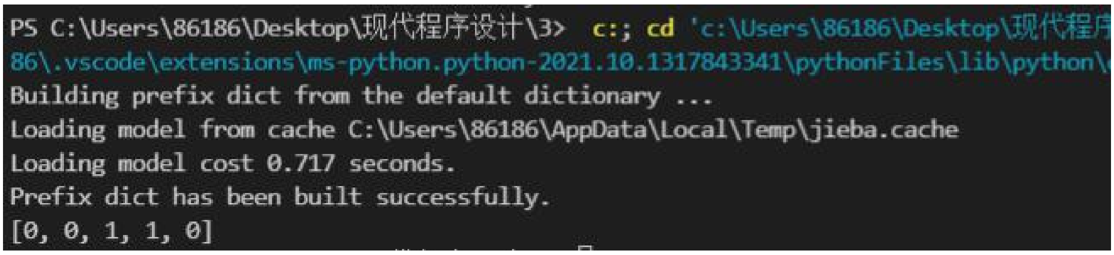
2. 实现两个函数，实现一条微博的情绪分析，返其情绪向量或情绪值。目前有两种方法，一是认为一条微博的情绪是混合的，即一共有 n 个情绪词，如果 joy 有 n1 个，则 joy 的比例是 n1/n；二是认为一条微博的情绪是唯一的，即 n 个情绪词里，anger 的情绪词最多，则该微博的情绪应该为 angry。注意，这里要求用闭包实现，尤其是要利用闭包实现一次加载情绪词典且局部变量持久化的特点。同时，也要注意考虑一些特别的情况，如无情绪词出现，不同情绪的情绪词出现数目一样等，并予以处理（如定义为无情绪，便于在后面的分析中去除）。

方法一：情绪向量



```
def getvector(anger,disgust,fear,joy,sadness):
    emodict={}
    emodict['anger']=anger
    emodict['disgust']=disgust
    emodict['fear']=fear
    emodict['joy']=joy
    emodict['sadness']=sadness
    def inget(comment):
        vector=[0,0,0,0,0]
        for i in comment:
            if i in emodict['anger']:
                vector[0] +=1
            elif i in emodict['disgust']:
                vector[1] +=1
            elif i in emodict['fear']:
                vector[2] +=1
            elif i in emodict['joy']:
                vector[3] +=1
            elif i in emodict['sadness']:
                vector[4] +=1
        return vector
    return inget
```

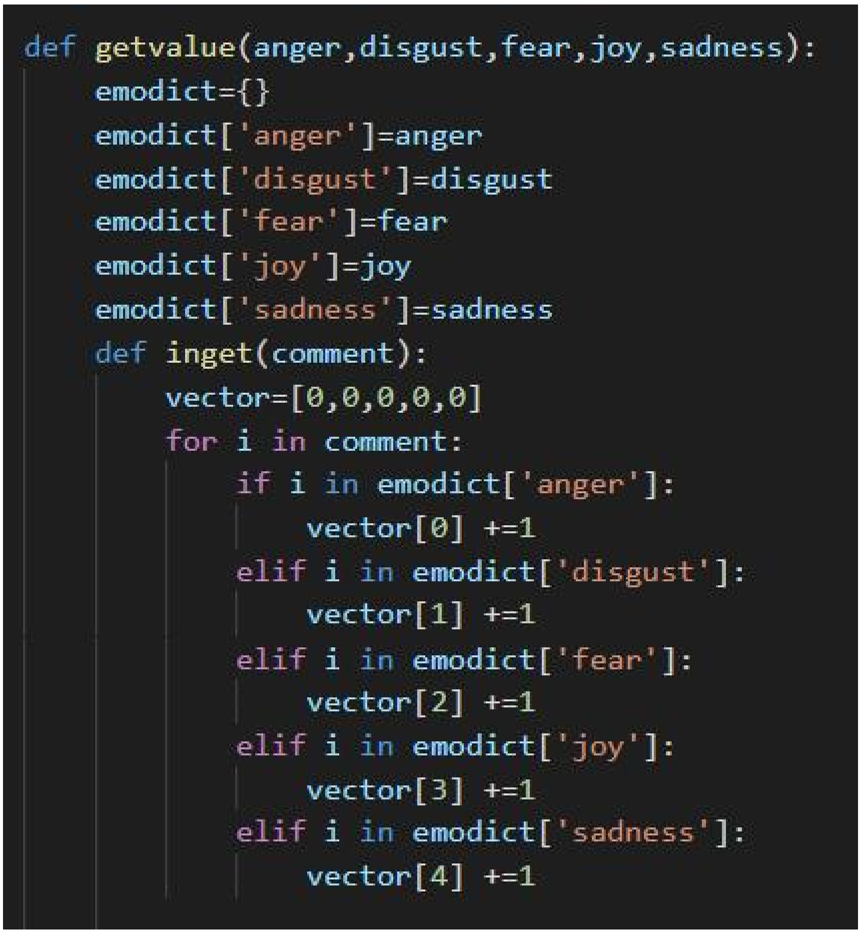
运行结果：



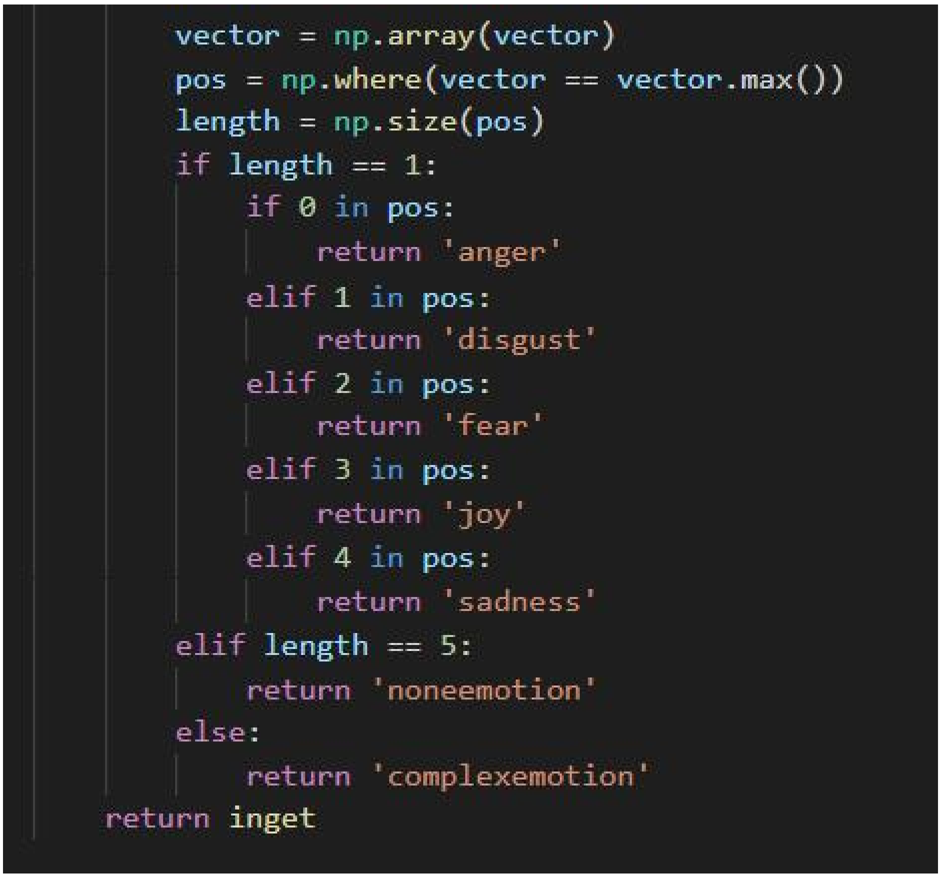
```
PS C:\Users\86186\Desktop\现代程序设计\3>  c:; cd 'c:\Users\86186\Desktop\现代程序
86\.vscode\extensions\ms-python.python-2021.10.1317843341\pythonFiles\lib\python\
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\86186\AppData\Local\Temp\jieba.cache
Loading model cost 0.717 seconds.
Prefix dict has been built successfully.
[0, 0, 1, 1, 0]
```

方法二：唯一值

是认为一条微博的情绪是唯一的，即 n 个情绪词里，anger 的情绪词最多，则该微博的情绪应该为 angry，如果没有情绪词则为 nonemotion，如果最大情绪词数目有多个则为 complexemotion

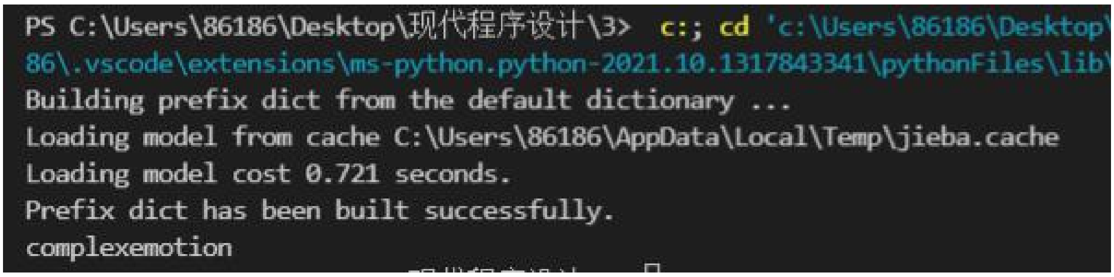


```python
def getvalue(anger,disgust,fear,joy,sadness):
    emodict={}
    emodict['anger']=anger
    emodict['disgust']=disgust
    emodict['fear']=fear
    emodict['joy']=joy
    emodict['sadness']=sadness
    def inget(comment):
        vector=[0,0,0,0,0]
        for i in comment:
            if i in emodict['anger']:
                vector[0] +=1
            elif i in emodict['disgust']:
                vector[1] +=1
            elif i in emodict['fear']:
                vector[2] +=1
            elif i in emodict['joy']:
                vector[3] +=1
            elif i in emodict['sadness']:
                vector[4] +=1
```



```
        vector = np.array(vector)
        pos = np.where(vector == vector.max())
        length = np.size(pos)
        if length == 1:
            if 0 in pos:
                return 'anger'
            elif 1 in pos:
                return 'disgust'
            elif 2 in pos:
                return 'fear'
            elif 3 in pos:
                return 'joy'
            elif 4 in pos:
                return 'sadness'
        elif length == 5:
            return 'noneemotion'
        else:
            return 'complexemotion'
    return inget
```

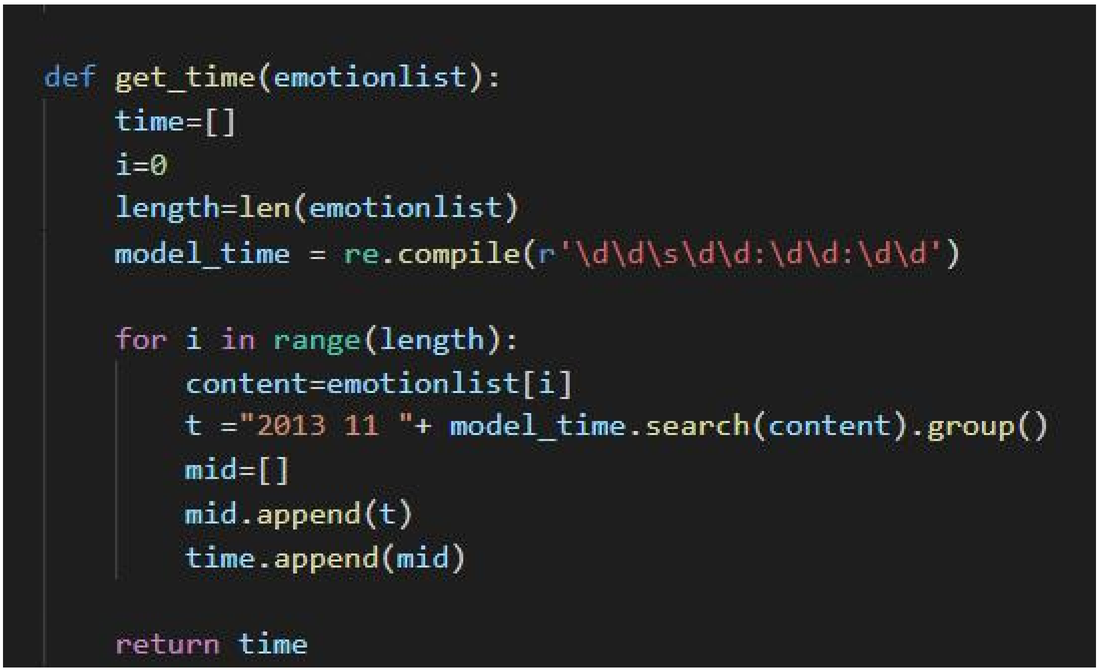
运行结果：



```
PS C:\Users\86186\Desktop\现代程序设计\3>  c:; cd 'c:\Users\86186\Desktop\
86\.vscode\extensions\ms-python.python-2021.10.1317843341\pythonFiles\lib\
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\86186\AppData\Local\Temp\jieba.cache
Loading model cost 0.721 seconds.
Prefix dict has been built successfully.
complexemotion
```

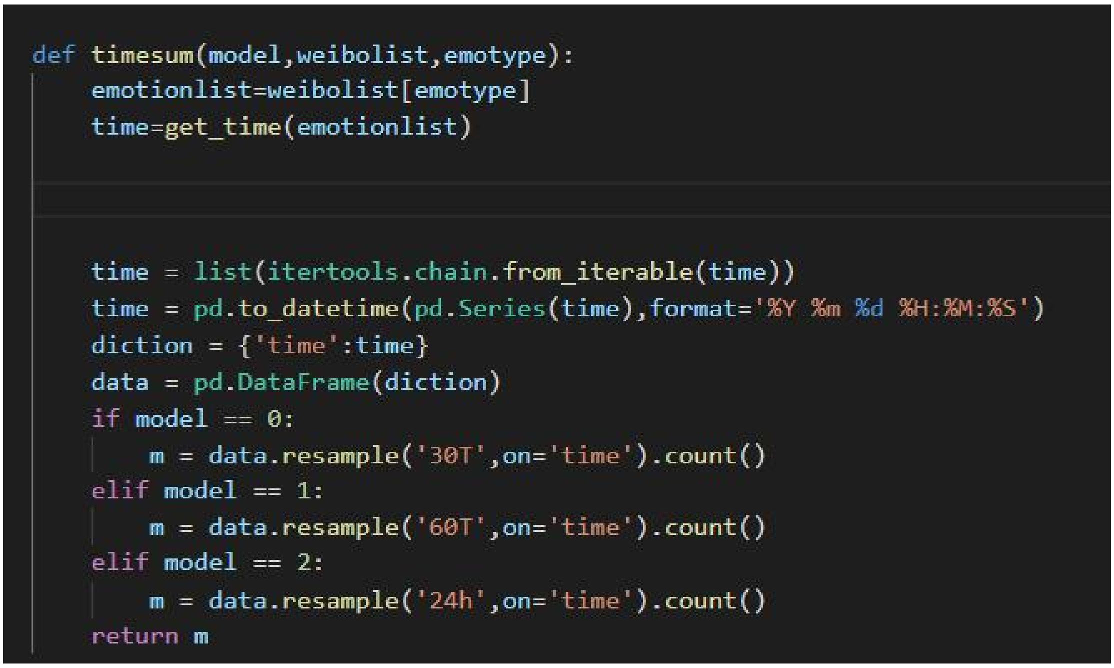
3. 微博中包含时间，可以讨论不同时间情绪比例的变化趋势，实现一个函数，可以通过参数来控制并返回对应情绪的时间模式，如 joy 的小时模式，sadness 的周模式等。

为了实现这个功能，首先要将每条微分类，一共有七种分别是 anger，disgust，fear，joy，sadness，noneemotion，complexemotion。用列表储存微博的原文本。然后利用正则表达式将某个情绪的时间提取出来，并利用列表储存



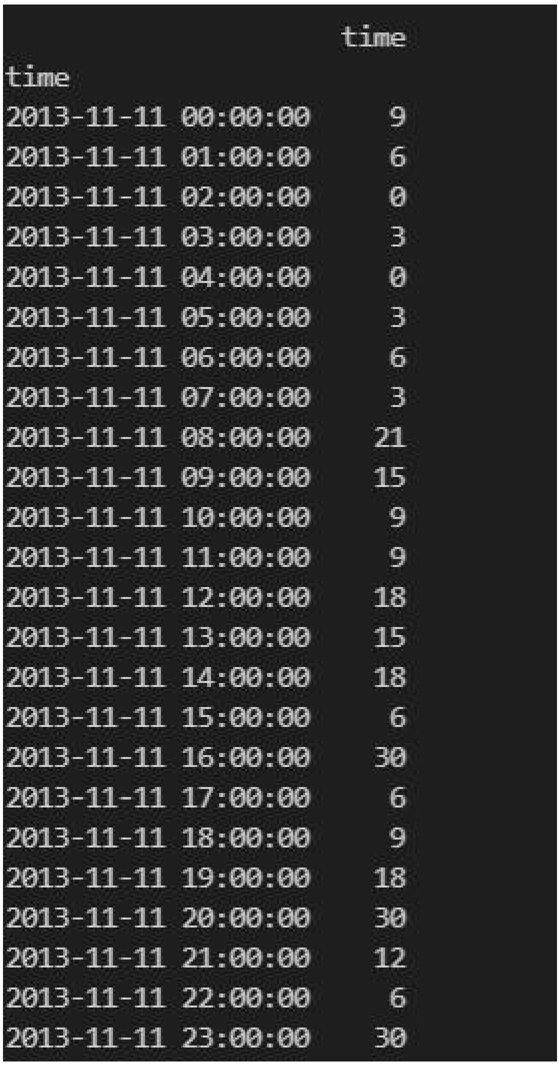
```
def get_time(emotionlist):
    time=[]
    i=0
    length=len(emotionlist)
    model_time = re.compile(r'\d\d\s\d\d:\d\d:\d\d')

    for i in range(length):
        content=emotionlist[i]
        t ="2013 11 "+ model_time.search(content).group()
        mid=[]
        mid.append(t)
        time.append(mid)

    return time
```

然后利用 pandas 库来统计某个情绪在不同时间段的频数，model 为不同的模式，0 为每半小时，1 为每小时，2 为每天，emotype 为不同情绪类型，0 为 anger，1 为 disgust，2 为 fear，3 为 joy，4 为 sadness

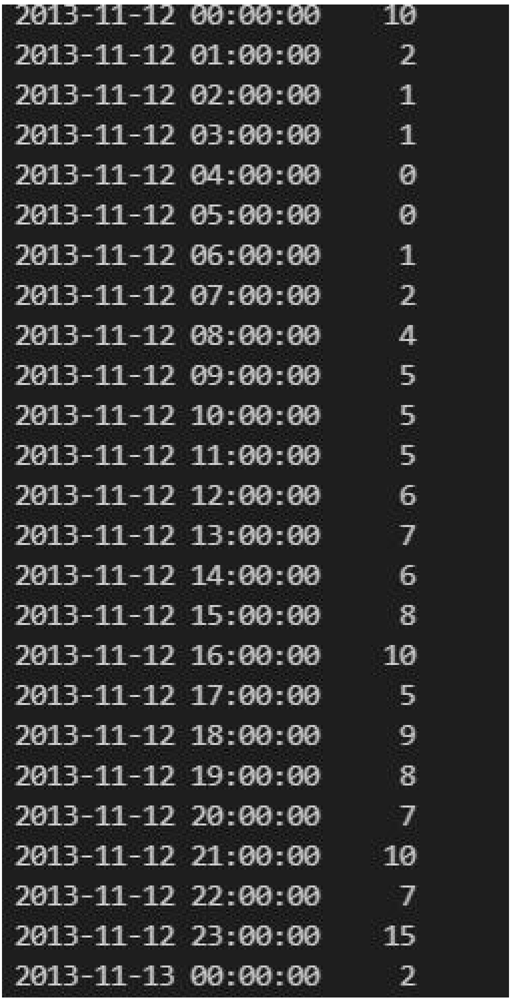


```
def timesum(model,weibolist,emotype):
    emotionlist=weibolist[emotype]
    time=get_time(emotionlist)



    time = list(itertools.chain.from_iterable(time))
    time = pd.to_datetime(pd.Series(time),format='%Y %m %d %H:%M:%S')
    diction = {'time':time}
    data = pd.DataFrame(diction)
    if model == 0:
        m = data.resample('30T',on='time').count()
    elif model == 1:
        m = data.resample('60T',on='time').count()
    elif model == 2:
        m = data.resample('24h',on='time').count()
    return m
```

运行结果：



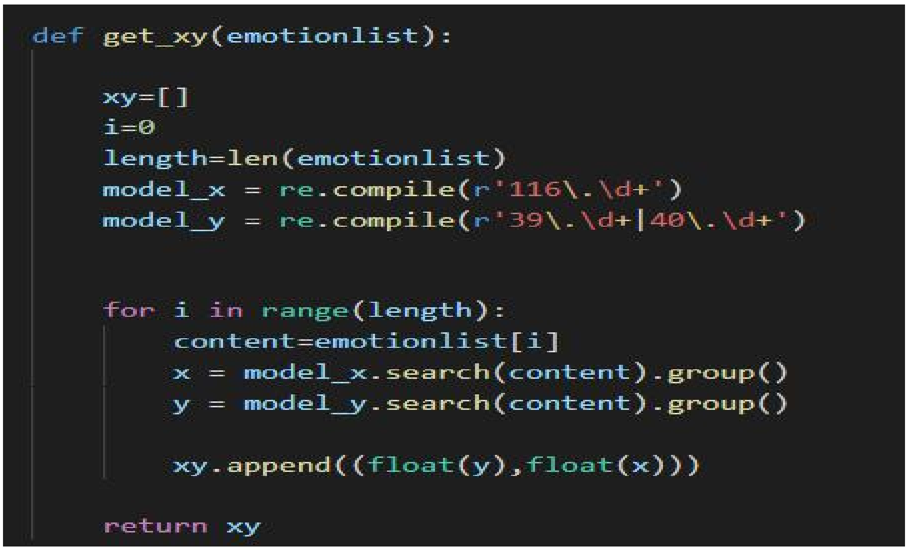
```
                         time
time
2013-11-11 00:00:00      9
2013-11-11 01:00:00      6
2013-11-11 02:00:00      0
2013-11-11 03:00:00      3
2013-11-11 04:00:00      0
2013-11-11 05:00:00      3
2013-11-11 06:00:00      6
2013-11-11 07:00:00      3
2013-11-11 08:00:00     21
2013-11-11 09:00:00     15
2013-11-11 10:00:00      9
2013-11-11 11:00:00      9
2013-11-11 12:00:00     18
2013-11-11 13:00:00     15
2013-11-11 14:00:00     18
2013-11-11 15:00:00      6
2013-11-11 16:00:00     30
2013-11-11 17:00:00      6
2013-11-11 18:00:00      9
2013-11-11 19:00:00     18
2013-11-11 20:00:00     30
2013-11-11 21:00:00     12
2013-11-11 22:00:00      6
2013-11-11 23:00:00     30
```



```
2013-11-12 00:00:00     10
2013-11-12 01:00:00      2
2013-11-12 02:00:00      1
2013-11-12 03:00:00      1
2013-11-12 04:00:00      0
2013-11-12 05:00:00      0
2013-11-12 06:00:00      1
2013-11-12 07:00:00      2
2013-11-12 08:00:00      4
2013-11-12 09:00:00      5
2013-11-12 10:00:00      5
2013-11-12 11:00:00      5
2013-11-12 12:00:00      6
2013-11-12 13:00:00      7
2013-11-12 14:00:00      6
2013-11-12 15:00:00      8
2013-11-12 16:00:00     10
2013-11-12 17:00:00      5
2013-11-12 18:00:00      9
2013-11-12 19:00:00      8
2013-11-12 20:00:00      7
2013-11-12 21:00:00     10
2013-11-12 22:00:00      7
2013-11-12 23:00:00     15
2013-11-13 00:00:00      2
```

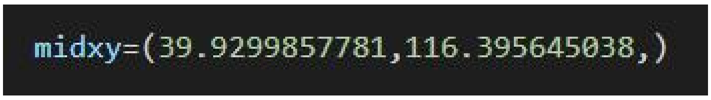
4. 微博中包含空间，可以讨论情绪的空间分布，实现一个函数，可以通过参数来控制并返回对应情绪的空间分布，即围绕某个中心点，随着半径增加该情绪所占比例的变化，中心点可默认值可以是城市的中心位置。

首先获取每条评论的坐标，储存在列表中



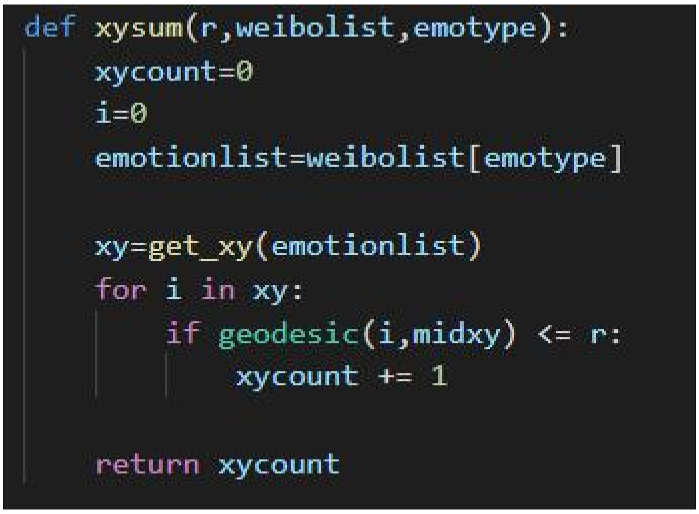
```python
def get_xy(emotionlist):

    xy=[]
    i=0
    length=len(emotionlist)
    model_x = re.compile(r'116\.\d+')
    model_y = re.compile(r'39\.\d+|40\.\d+')


    for i in range(length):
        content=emotionlist[i]
        x = model_x.search(content).group()
        y = model_y.search(content).group()

        xy.append((float(y),float(x)))

    return xy
```

定义北京市中心的坐标：



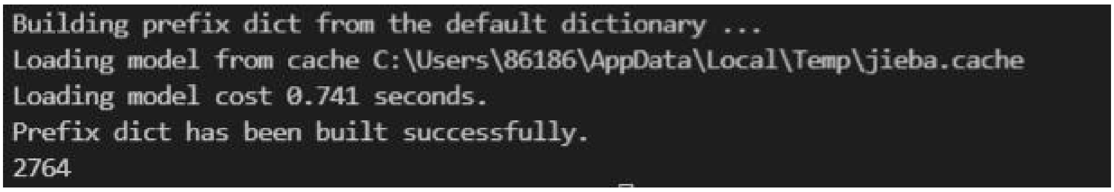
```
midxy=(39.9299857781,116.395645038,)
```

然后计算每条评论的坐标距离中心的位置，小于半径 r 的进行统计，emotype 为不同情绪类型，0 为 anger，1 为 disgust，2 为 fear，3 为 joy，4 为 sadness



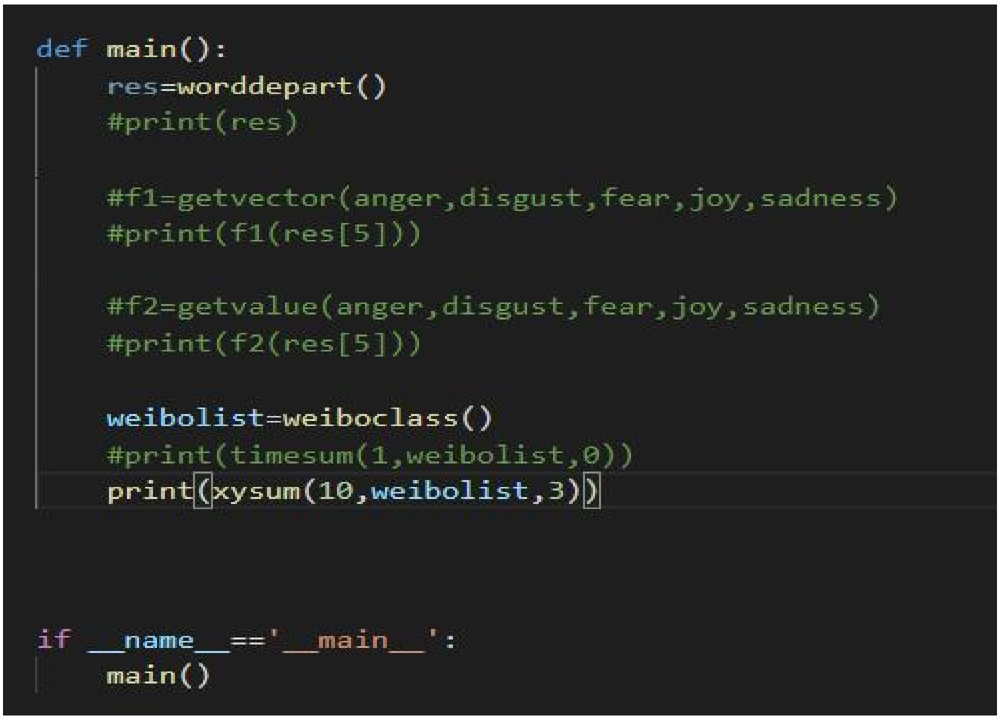
```
def xysum(r,weibolist,emotype):
    xycount=0
    i=0
    emotionlist=weibolist[emotype]

    xy=get_xy(emotionlist)
    for i in xy:
        if geodesic(i,midxy) <= r:
            xycount += 1

    return xycount
```

运行结果：



```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\86186\AppData\Local\Temp\jieba.cache
Loading model cost 0.741 seconds.
Prefix dict has been built successfully.
2764
```

主函数：



```
def main():
    res=worddepart()
    #print(res)

    #f1=getvector(anger,disgust,fear,joy,sadness)
    #print(f1(res[5]))

    #f2=getvalue(anger,disgust,fear,joy,sadness)
    #print(f2(res[5]))

    weibolist=weiboclass()
    #print(timesum(1,weibolist,0))
    print(xysum(10,weibolist,3))



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