**数据结构实验报告13**

**学号：** 117060400225 **姓名**： 池艳 **班级：** 应用统计学二班  **指导老师：** 林卫中

**实验名称**： 程序设计方法论

**实验要求：（1）安装第三方库的复习**

1. **对微信朋友圈数据的获取**
2. **词频分析**

**实验题目：程序设计方法论**

**算法实现：**

1.登陆微信朋友圈

import itchat

import numpy as np

import matplotlib.pyplot as plt

import matplotlib as mpl

import re

import jieba

import PIL.Image as Image

from wordcloud import WordCloud, ImageColorGenerator

#登录朋友圈

def login():

    itchat.login()

    friends=itchat.get\_friends(update=True)[0:]

    return friends

friends = login()

2.朋友圈男女性别比例

#获取朋友圈数据

def get\_var(var, friends):

    variable = []

    for i in friends:

        value = i[var]

        variable.append(value)

    return variable

#朋友圈性别比例

def analyseGender(friends):

    male=female=other=0

    sexes = get\_var('Sex', friends)

    for sex in sexes:

        if sex == 1:

            male += 1

        elif sex == 2:

            female += 1

        else:

            other += 1

    total = len(friends[1:])

    malecol = round( float(male)/total \* 100, 2)

    femalecol = round( float(female)/total \* 100, 2)

    othercol = round( float(other)/total \* 100, 2)

    print('男性好友：{:.2f}%%'.format( malecol))

    print('女性好友：{:.2f}%%'.format( femalecol))

    print('不明性别好友：{:.2f}%%'.format( othercol))

    #plot code

    mpl.rcParams['font.sans-serif']=['SimHei']

    mpl.rcParams['axes.unicode\_minus'] = False

    map = {

        'Male':(malecol, '#7199cf'),

        'Female': (femalecol, '#4fc4aa'),

        'other': (othercol, '#e1a7a2')

    }

    fig = plt.figure( figsize=(5,5))

    ax = fig.add\_subplot(111)

    ax.set\_title( '朋友圈性别')

    xticks = np.arange(3) + 0.15

    bar\_width = 0.5

    names = map.keys()

    values = [ x[0] for x in map.values()]

    colors = [ x[1] for x in map.values()]

    #柱状图

    bars = ax.bar( xticks, values, width=bar\_width, edgecolor='none')

    ax.set\_ylabel('比例')

    ax.set\_xlabel('性别')

    ax.grid()

    ax.set\_xticks( xticks)

    ax.set\_xticklabels( names)

    ax.set\_xlim( [bar\_width/2 - 0.5, 3 - bar\_width/2])

    ax.set\_ylim( [0, 100])

    for bar, color in zip( bars, colors):

        bar.set\_color( color)

        height = bar.get\_height()

        plt.text( bar.get\_x(), bar.get\_height()/4.+ height, '{:.2f}%'.format( float(height)))

    plt.show()

    #饼状图

    fig1 = plt.figure( figsize=(5,5))

    ax = fig1.add\_subplot(111)

    ax.set\_title('饼图')

    labels = ['{}\n{}%'.format(name, value) for name, value in zip( names, values)]

    ax.pie(values, labels=labels, colors=colors)

    plt.show()

analyseGender(friends)

3.省份归属

def analyseProvince(friends):

    provlist = get\_var('Province', friends)

    provdict = {}

    for p in provlist:

        provdict[p] = provdict.get(p,0) + 1

    provdict = sorted(provdict.items(), key= lambda x : x[1], reverse=True)

    #画图

    figpro = plt.figure(figsize=(10,5))

    axpro = figpro.add\_subplot(111)

    axpro.set\_title('省份')

    xticks = np.linspace(0.5,20,10)

    bar\_width = 0.8

    pros= []

    values = []

    count = 0

    for d in provdict:

        pros.append(d[0])

        values.append(d[1])

        count += 1

        if count >= 10:

            break

    colors = ['#FFEC88', '#FFE4C4','#FFC125','#FFB6C1','#CDCDB4','#CDC8B1','#CDB79E','#CDAD00','#CD96CD',\

              '#CD853F']

    bars = axpro.bar( xticks, values, width=bar\_width, edgecolor='none')

    axpro.set\_ylabel('人数')

    axpro.set\_xlabel('省份')

    axpro.grid()

    axpro.set\_xticks( xticks)

    axpro.set\_xticklabels(pros)

    axpro.set\_xlim(0,20)

    axpro.set\_ylim([0,100])

    for bar, color in zip( bars, colors):

        bar.set\_color(color)

        height = bar.get\_height()

        plt.text( bar.get\_x()+bar.get\_width()/4., height, '{}'.format(height))

    plt.show()

analyseProvince(friends)

4.词频统计

def drawWordcloudPlot(counts):

    coloring = np.array(Image.open("E:/baidupic/alice\_color.png"))

    wc = WordCloud(background\_color="white",

                   max\_words=2000,

                   mask=coloring,

                   max\_font\_size=60,

                   random\_state=42,

                   scale=2,

                   font\_path="c:/Windows/Fonts/SimHei.ttf")

    wc.generate\_from\_frequencies(counts)

    image\_colors = ImageColorGenerator(coloring)

    plt.imshow(wc)

    plt.axis("off")

    plt.savefig('friendSign.jpg')

    plt.show()

def analyseSignature(friends):

    signatures = get\_var('Signature', friends)

    siglist = []

    for sign in signatures:

        sign = sign.strip().replace("span", "").replace("class", "").replace("emoji", "")

        rep = re.compile("lf\d+\w\*|[<>/=]")

        sign = rep.sub("", sign)

        siglist.append(sign)

    text = "".join(siglist)

    wlist = jieba.cut(text, cut\_all=True)

    counts = {}

    for word in wlist:

        if len(word) == 1:

            continue

        else:

            counts[word] = counts.get(word, 0) + 1

    wdict = {}

    for d in counts.items():

        if d[1] > 2:

            wdict[d[0]] = d[1]

    drawWordcloudPlot(wdict)

analyseSignature(friends)

5.个性签名中出现相同字的词频统计

def drawWordcloudPlot(counts):

    coloring = np.array(Image.open("E:/alice\_color.png"))

    wc = WordCloud(background\_color="white",

                   max\_words=2000,

                   mask=coloring,

                   max\_font\_size=60,

                   random\_state=42,

                   scale=2,

                   font\_path="c:/Windows/Fonts/SimHei.ttf")

    wc.generate\_from\_frequencies(counts)

    image\_colors = ImageColorGenerator(coloring)

    plt.imshow(wc)

    plt.axis("off")

    plt.savefig('e:/friendSign.jpg')

    plt.show()

def analyseSignature(friends):

    signatures = get\_var('Signature', friends)

    siglist = []

    for sign in signatures:

        sign = sign.strip().replace("span", "").replace("class", "").replace("emoji", "")

        rep = re.compile("lf\d+\w\*|[<>/=]")

        sign = rep.sub("", sign)

        siglist.append(sign)

    text = "".join(siglist)

    wlist = jieba.cut(text, cut\_all=True)

    counts = {}

    for word in wlist:

        if len(word) == 1:

            continue

        else:

            counts[word] = counts.get(word, 0) + 1

    wdict = {}

    for d in counts.items():

        if d[1] > 2:

            wdict[d[0]] = d[1]

    drawWordcloudPlot(counts)

analyseSignature(friends)

**实验结果：**