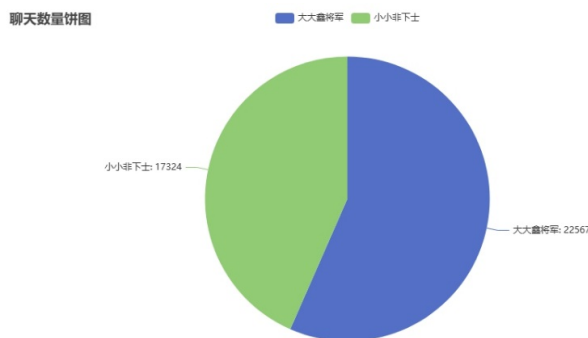


这是做出来的效果图



CSDN @Clannad47

推荐使用一个叫[留痕](#)的软件，下载完后扫描自己的本地微信文件，后在好友界面点击导出聊天记录为CSV即可。



到处后可以看到导出的文件路径我这里的是一般是你Memotrace安装位置的.\data\聊天记录\ (记上后面会用)。

## 二、可视处理

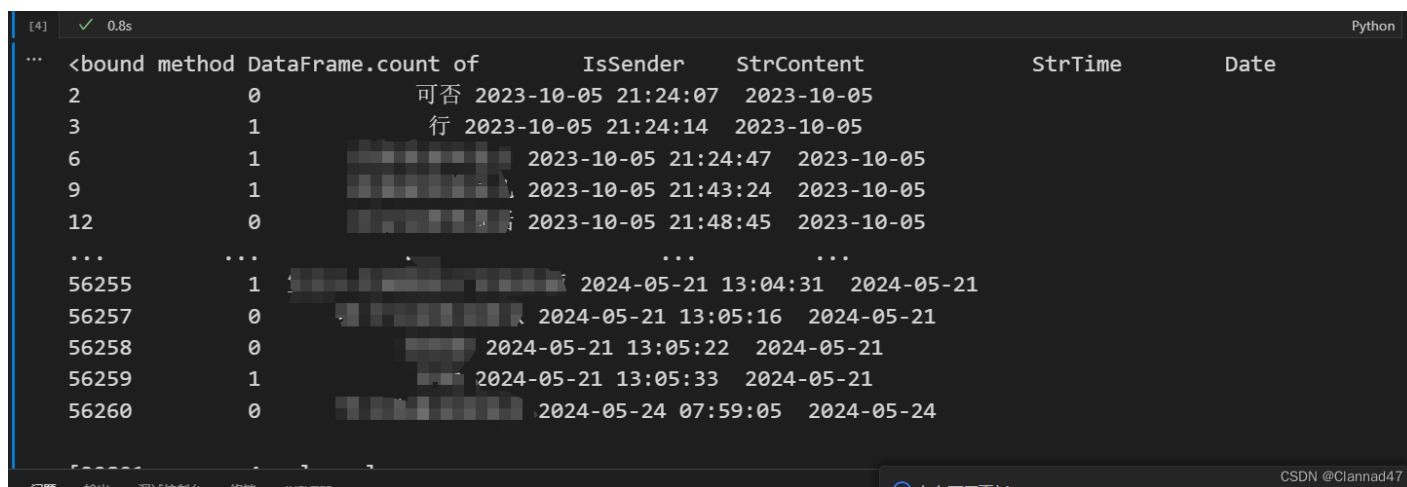
使用python进行可视化操作，这里我使用的是pyecharts库。以下是可视化代码。

数据预处理：

```
##读取群聊文件的地址，这里改成你导出的群聊文件地址
fileurl=r"你的群聊文件地址"

##数据预处理
def data_process(url):
    df=pd.read_csv(url)
    df = df[df['Type'] == 1] # 只保留文本聊天
    ##保留特定群员信息
    df=df[(df['NickName']=="你的名字")| (df['NickName']=='她的名字')] ##这里换成你想保留人的群聊名字
    selected_columns = ['IsSender', 'StrContent', 'StrTime']
    df = df[selected_columns] # 只取'IsSender','StrContent','StrTime'列
    df['StrTime'] = pd.to_datetime(df['StrTime'])
    df['Date'] = df['StrTime'].dt.date
    return df
```

这里只留下文本类聊天消息，可以筛选特定的群友，这里取IsSender（发件人）、StrContent（消息）和StrTime（时间）三列数据。下面是数据处理后的格式。



数据可视化过程，这里画了四个图聊天频率月份图（直方图）、饼图、聊天日分布图（直方图）、聊天词云图这里就直接把完整的代码贴出。

```
import pandas as pd
import pyecharts as pc
from pyecharts import options as opts
import jieba

##数据预处理
def data_process(url):
    df=pd.read_csv(url)
    df = df[df['Type'] == 1] # 只保留文本聊天
    ##保留特定群员信息
    df=df[(df['NickName']=="你的名字")| (df['NickName']=='她的名字')] ##这里换成你想保留人的群聊名字
    selected_columns = ['IsSender', 'StrContent', 'StrTime']
    df = df[selected_columns] # 只取'IsSender','StrContent','StrTime'列
    df['StrTime'] = pd.to_datetime(df['StrTime'])
```

```

    df['StrTime'] = pd.to_datetime(df['StrTime'])
    df['Date'] = df['StrTime'].dt.date
    return df

##读取群聊文件的地址，这里改成你导出的群聊文件地址
fileurl=r"你的群聊文件地址"

df=data_process(fileurl)

def zhuzhuangtu():
# 每天聊天频率柱状图
    chat_frequency = df['Date'].value_counts().sort_index()
    total_messages = len(df)
    date_labels = [date.strftime('%m-%d') for date in chat_frequency.index]
    bar = pc.charts.Bar()
    bar.add_xaxis(date_labels)
    bar.add_yaxis('频率', list(chat_frequency))
    bar.render("./data/zhuzhuangtu.html")
    return bar

# 双方信息数量对比
def pie():
    sent_by_me = df[df['IsSender'] == 1]['StrContent']
    sent_by_others = df[df['IsSender'] == 0]['StrContent']
    count_sent_by_me = len(sent_by_me)
    count_sent_by_others = len(sent_by_others)
    labels = ['你的名字', '她的名字']
    sizes = [count_sent_by_me, count_sent_by_others]
    pie=pc.charts.Pie()
    pie.add("",[list(z) for z in zip(labels,sizes)])
    # 添加标签
    pie.set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {c}"))
    return pie

##每小时聊天记录
# 根据一天中的每一个小时进行统计聊天频率，并生成柱状图
def xiaoshitu():

    df['DateTime'] = pd.to_datetime(df['StrTime'])
    df['Hour'] = df['DateTime'].dt.hour

    hourly_counts = df['Hour'].value_counts().sort_index().reset_index()
    hourly_counts.columns = ['Hour', 'Frequency']
    bar1=pc.charts.Bar()
    bar1.add_xaxis(hourly_counts['Hour'].to_list())
    bar1.add_yaxis('频率',hourly_counts['Frequency'].to_list())
    return bar1

##绘制词云
def ciyun():

    # 获取词云内容
    words = df['StrContent'].to_string()
    ##分词

    # 定义要去除的词列表

```

```

filter_list = ['的', '了', '是', '在', '和', '我', '之', '我们', '你们', '他们', '有', '也', '不', '人', '这',
               '上', '到', '们', '小', '中', '为', '上', '们', '都', '会', '他', '地', '那', '要', '就', '以',
               '出', '从', '到', '.....', '呀', '吧']

words = jieba.cut(words, cut_all=False)
filtered_words = [word for word in words if word not in filter_list and word.isalpha()]
# words=' '.join(filtered_words )
# print(type(filtered_words))
# 计算词频
word_dict = {}
for word in filtered_words:
    if len(word) > 1:
        if word in word_dict:
            word_dict[word] += 1
        else:
            word_dict[word] = 1
# 将词频统计结果转换为列表
word_list = [(word, value) for word, value in word_dict.items()]
# print(word_list)

# 绘制词云
wordcloud = pc.charts.WordCloud()
wordcloud.add("", word_list, word_size_range=[20,100])
return wordcloud

# 主函数
if __name__ == '__main__':
    zhuzhuang=zhuzhuangtu()
    pietu=pie()
    xiaoshi=xiaoshitu()
    ciyuntu=ciyun()
    #
    zhuzhuang.set_global_opts(title_opts={"text": "聊天月份频率图"})
    pietu.set_global_opts(title_opts={"text": "聊天数量饼图"})
    xiaoshi.set_global_opts(title_opts={"text": "聊天全天分布图"})
    ciyuntu.set_global_opts(title_opts={"text": "词云"})
    page = pc.charts.Page(layout=pc.charts.Page.SimplePageLayout)
    page.add(zhuzhuang)
    page.add(pietu)
    page.add(xiaoshi)
    page.add(ciyuntu)

    # 设置页面标题
    page.page_title="聊天信息总结"

    # 保存页面
    page.render("你要保存的地址+文件名.html")

```

感谢阅读。