《数据处理与可视化》实验报告

学　号：1005183121　　姓　名：周子杰

日　期：2021.7.10　　得　分：

一、 实验内容

- 数据读取
- 数据清洗
- 数据可视化
- 相关性分析

二、 数据读取

首先，需要进行导包操作：

```
1. import pandas as pd
2. import numpy as np
```

接着，用 pandas 来进行对 csv 中数据的读取：

```
1. df = pd.read_csv("201506-citibike-tripdata.csv")
2. print(df.head())
```

输出前五行，得到如下结果：

```
PS D:\Practise\gongsunhui> python -u "d:\Practise\gongsunhui\dataclean.py"
   tripduration      starttime       stoptime  start station id  ... bikeid    usertype  birth year  gender
0          1338  6/1/2015 0:00  6/1/2015 0:22               128  ...  20721  Subscriber      1984.0       1
1           290  6/1/2015 0:00  6/1/2015 0:05               438  ...  21606  Subscriber      1997.0       1
2           634  6/1/2015 0:01  6/1/2015 0:11               383  ...  16595  Subscriber      1993.0       1
3           159  6/1/2015 0:01  6/1/2015 0:04               361  ...  16949  Subscriber      1981.0       1
4          1233  6/1/2015 0:02  6/1/2015 0:22               382  ...  17028    Customer         NaN       0
```

可以看到，读取成功。

三、 数据清洗

首先，先看是否有空值：

```
1.  print(df.isnull().sum())
```

```
PS D:\Practise\gongsunhui> python -u "d:\Practise\gongsunhui\dataclean.py"
tripduration                  0
starttime                     0
stoptime                      0
start station id              0
start station name            0
start station latitude        0
start station longitude       0
end station id                0
end station name              0
end station latitude          0
end station longitude         0
bikeid                        0
usertype                      0
birth year               130392
gender                        0
dtype: int64
```

发现有空值，删除空值所在行并保存数据到 cleanedDate.csv：

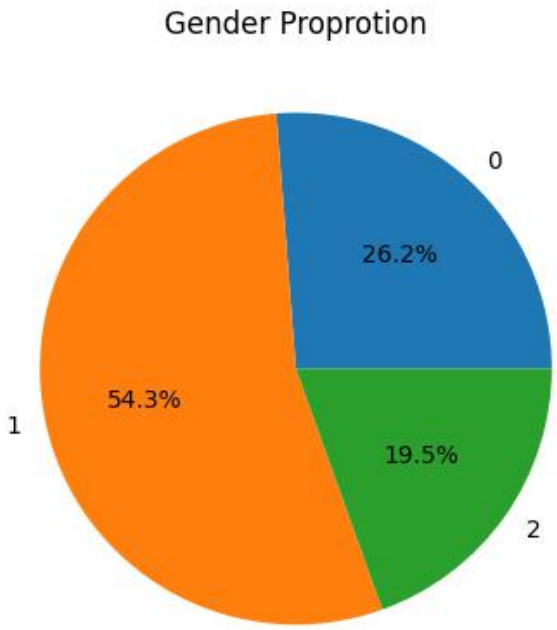
```
1.  df = df.dropna()
2.  df.to_csv("cleanedData.csv")
```

四、 数据可视化

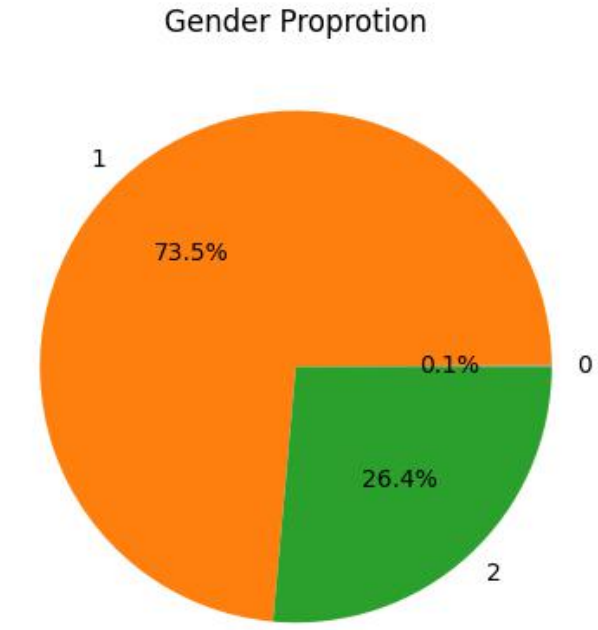
数据可视化用到的工具有 matplotlib.pyplot、pyecharts、seaborn 和百度 api（在 pyecharts 中调用绘制地图数据）。下面是我绘制的一些图像：

性别占比：

由于数据的局限性，我没有追加该地区男女比例的占比，这里仅仅展示骑行共享单车中的男女占比：



这是清洗前的数据，可以看到为完成信息（性别为未知）的用户占比为26.2%。

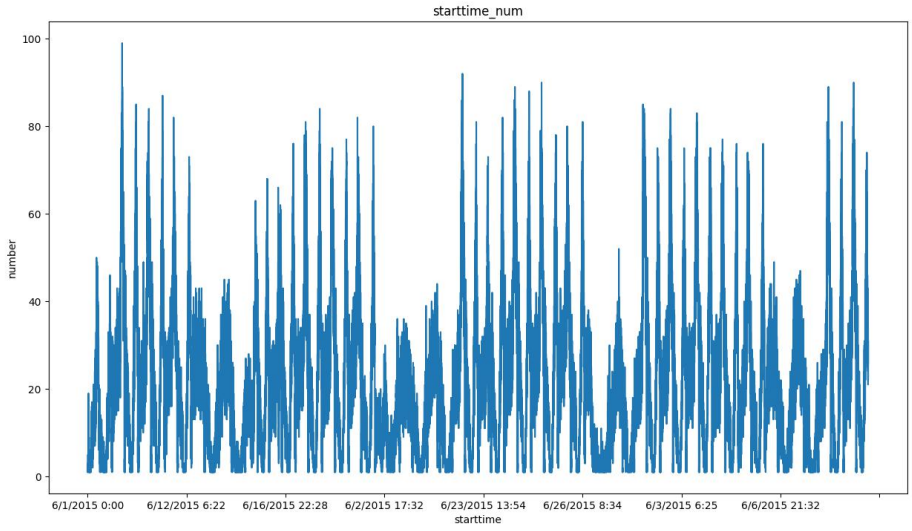


这是清洗完的数据，可以看到男性用户相比女性用户是偏多的。

```
# 饼图 不同性别占比
var=df.groupby(['gender']).sum().stack()
temp=var.unstack()
type(temp)
x_list = temp['tripduration']
label_list = temp.index
plt.pie(x_list,labels=label_list,autopct="%1.1f%%")
plt.title("Gender Proprotion")
plt.show()
```

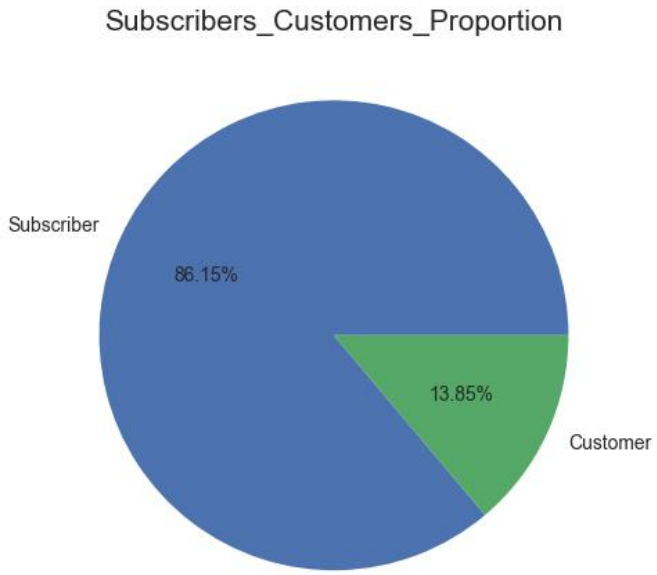
这是该饼图的代码，由于各个图像的代码都是类似的，后续图像就不再一一展示代码了。

不同时间出行量：

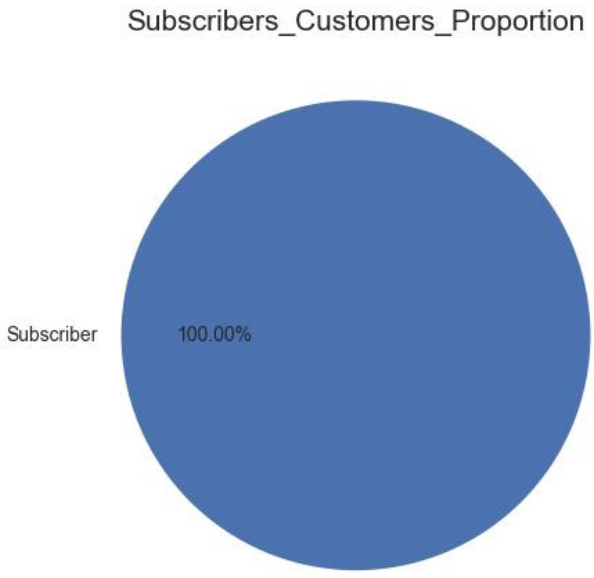


我们可以看到，对于同一天的不同时间，以及不同的日子人们的骑单车出行的情况是不同的。

订阅用户占比：



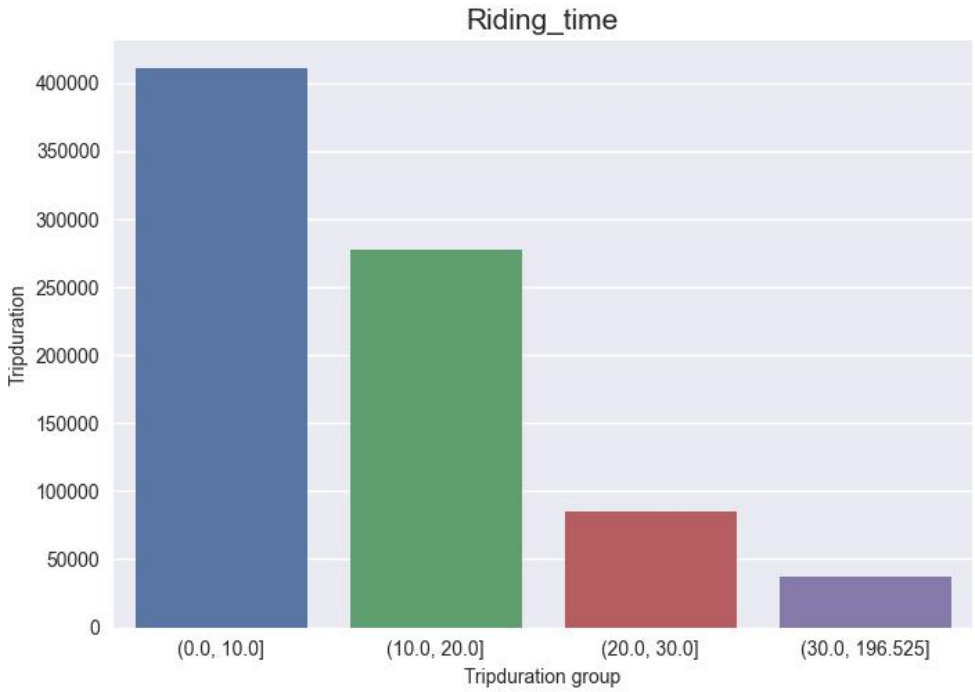
这是清洗前的数据，可以看到订阅用户占比为 86.15%



这是清洗后的数据，可以看到几乎全是订阅用户，我们可以得到结论：订阅用户更倾向于完善自己的信息（清洗掉的都是有空值的数据），但这也可能与软

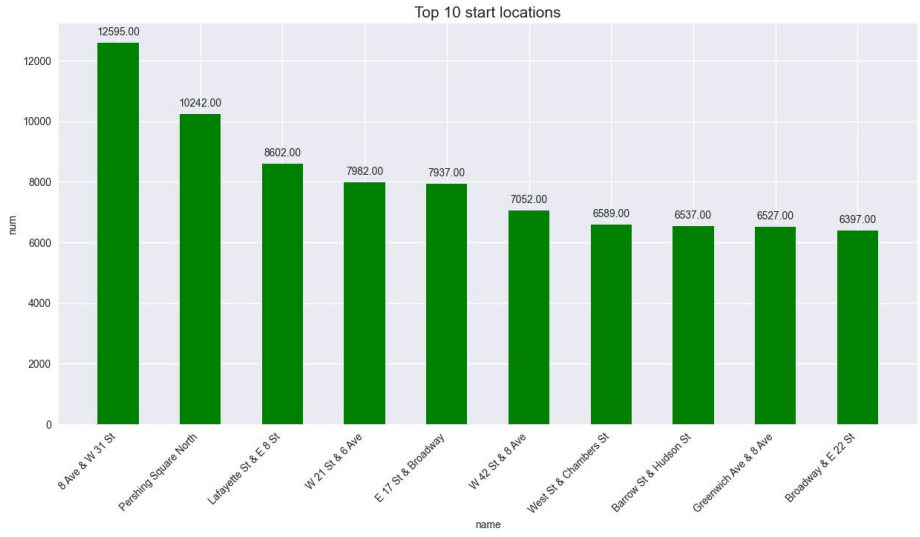
件中的相关约定有关。接下来的数据我讲采用订阅用户的数据来进行，也就是清洗完的数据。

骑行时长分布：

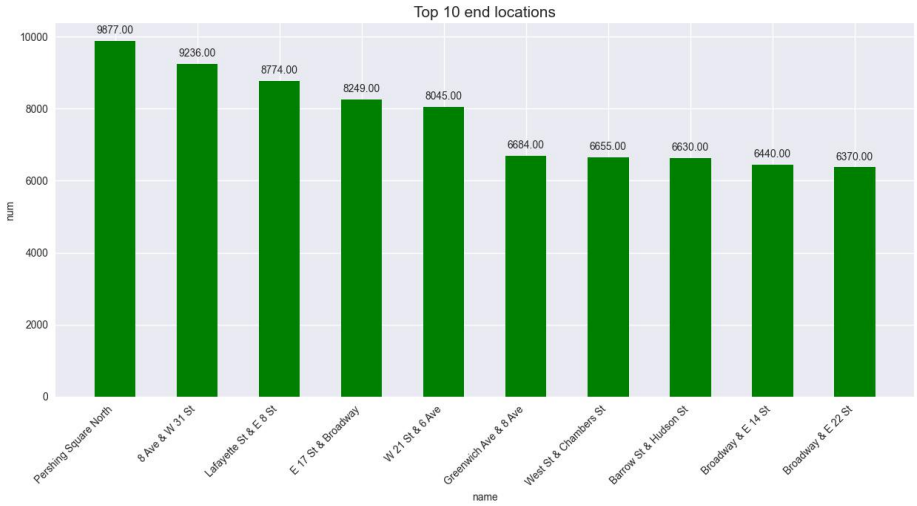


可以看到，大部分人的骑行时长在 0~10 之间，超过 20 的相对稀少。

最流行开始&结束地点



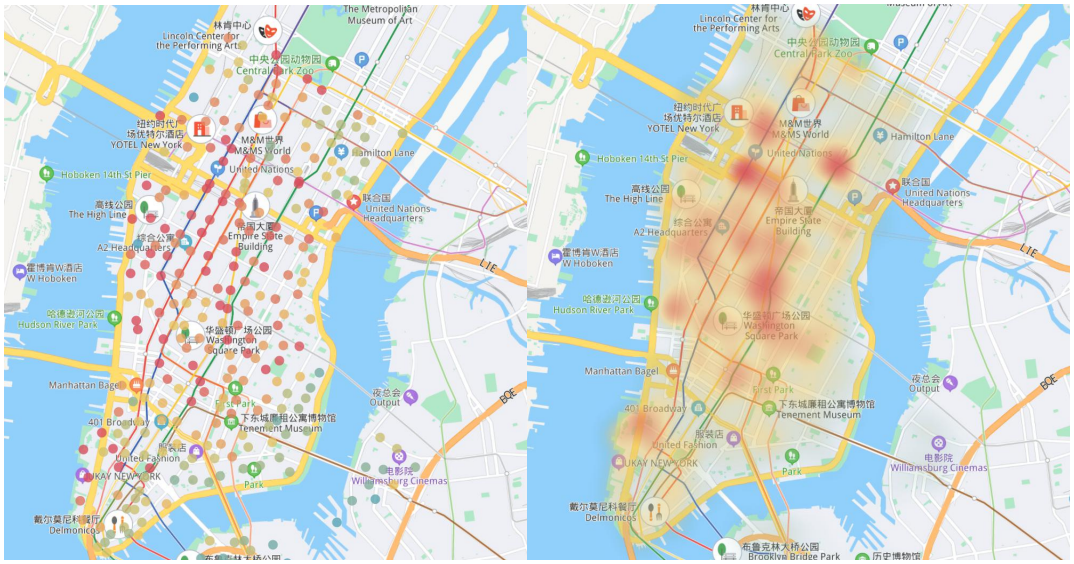
最流行开始地点



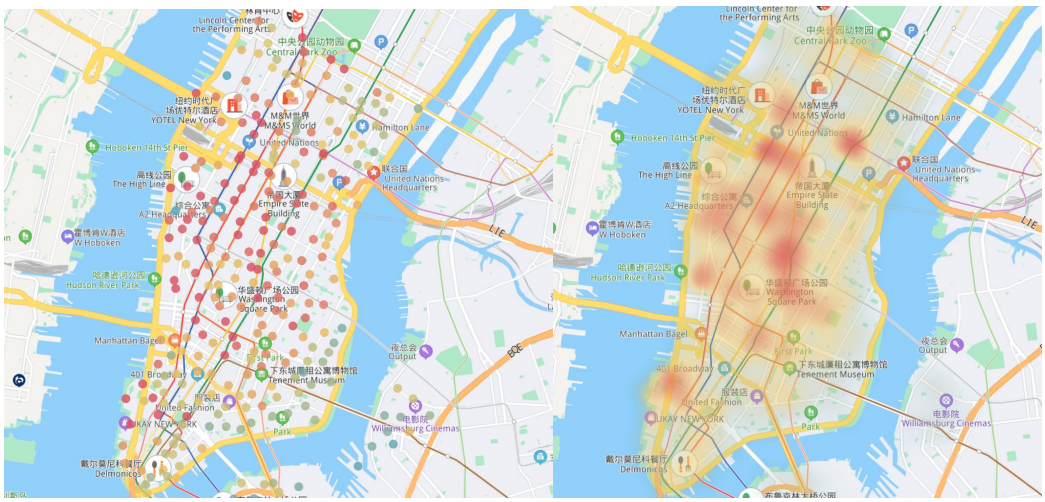
最流行结束地点

从上图中可以看出最流行的开始和结束地点，这些数据将有助于公司合理地安排共享单车投放地点和运输情况。

起点终点分布：



起点分布图+热力图



终点分布图+热力图

从中可以看出，骑单车出行的人集中在西部，起点终点的分布并没有明显区别。

他们的代码如下：（由于起点和终点的代码是类似的，这里就只展示起点代码了。同时也省略了读文件、写经纬度等一些操作，只展示了最核心的函数）

```
# 起点分布图
def draw_start_station_pos():
    list_station_val = read_start_pos()
    c = (
        BMap(init_opts=opts.InitOpts(width="2000px",height="1000px"))
        .add_schema(baidu_ak="GF7GOq6CbjMcGfDvFGuIFjrCKYi3zeTy", center=[-73.99069656,40.72502876],zoom=15)
        .add_coordinate_json("infor_start.json")
        .add(
            series_name="起点分布图",
            type_="scatter",
            data_pair=list_station_val
        )
        .set_global_opts(
            title_opts=opts.TitleOpts(title="起点分布图"),
            visualmap_opts=opts.VisualMapOpts(max_=5000),

        )
        .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
        .render("draw_start_station_pos.html")
    )

# 起点分布热力图
def draw_start_station_pos_heat():
```

```
    list_station_val = read_start_pos()
    c = (
        BMap(init_opts=opts.InitOpts(width="2000px",height="1000px"))
        .add_schema(baidu_ak="GF7GOq6CbjMcGfDvFGuIFjrCKYi3zeTy", center=[-73.99069656,40.72502876],zoom=15)
        .add_coordinate_json("infor_start.json")
        .add(
            series_name="起点分布图",
            type_="heatmap",
            data_pair=list_station_val
        )
        .set_global_opts(
            title_opts=opts.TitleOpts(title="起点分布图"),
            visualmap_opts=opts.VisualMapOpts(max_=12000,min_=0),

        )
        .render("draw_start_station_pos_heat.html")
    )
```

五、 相关性分析

皮尔森相关系数：

我们首先测试 gender 和 tripduration 的皮尔森相关系数，代码如下：

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math as math

df = pd.read_csv("cleanedData.csv")

def pearson(vector1, vector2):
    n = len(vector1)
    #simple sums
    sum1 = sum(float(vector1[i]) for i in range(n))
    sum2 = sum(float(vector2[i]) for i in range(n))
    #sum up the squares
    sum1_pow = sum([pow(v, 2.0) for v in vector1])
    sum2_pow = sum([pow(v, 2.0) for v in vector2])
    #sum up the products
```

```
    p_sum = sum([vector1[i]*vector2[i] for i in range(n)])
    #分子 num，分母 den
    num = p_sum - (sum1*sum2/n)
    den = math.sqrt((sum1_pow-pow(sum1, 2)/n)*(sum2_pow-pow(sum2, 2)/n))
    if den == 0:
        return 0.0
    return num/den

print(pearson(df["gender"],df["tripduration"]))
# print(pearson(2015-df["birth year"],df["tripduration"]))
```

运行可以得到如下结果：

```
PS D:\Practise\dataProcess> python -u "d:\Practise\dataProcess\pearson.py"
0.02147535960218521
```

由此可以知道性别和骑行长度的关系很小。

同理，可以得到年龄和骑行长度的关系，代码上仅仅是函数调用中变量的区别，就不再展示了（PS：之所以用 2015 而不是 2021 减去出生年是因为这个表为 2015 年的），这里仅仅展示结果：

```
PS D:\Practise\dataProcess> python -u "d:\Practise\dataProcess\pearson.py"
0.008453527444893966
```

可知骑行长度和年龄也没有必然联系。

最高温度与骑行人数关系：

由于是表中没有的数据，首先需要对气温进行数据挖掘，得到如下结果：

date	MaxTemp	MinTemp	Weather	Wind
2015-06-01 周一	30°	16°	雷阵雨~中雨	东南偏南风 1 级
2015-06-02 周二	13°	11°	中雨	东北偏东风 3 级
2015-06-03 周三	13°	11°	中雨~小雨	东北风 2 级
2015-06-04 周四	21°	12°	多云	东风 1 级
2015-06-05 周五	18°	12°	多云~小雨	东北偏东风 2 级
2015-06-06 周六	21°	16°	小雨	东北偏东风 1 级
2015-06-07 周日	24°	12°	多云~晴	东北偏北风 2 级
2015-06-08 周	23°	15°	晴~多云	东南风 2 级

一				
2015-06-09 周二	26°	18°	小雨~大雨	西南偏南风 2 级
2015-06-10 周三	28°	18°	多云~小雨	西南偏西风 2 级
2015-06-11 周四	27°	20°	晴	西南风 1 级
2015-06-12 周五	31°	22°	多云~雷阵雨	西风 1 级
2015-06-13 周六	31°	22°	多云~大雨	东南风 1 级
2015-06-14 周日	30°	19°	多云~晴	北风 2 级
2015-06-15 周一	31°	19°	多云~大雨	东南风 1 级
2015-06-16 周二	28°	18°	大雨~雷阵雨	西北风 1 级
2015-06-17 周三	25°	19°	雷阵雨~晴	西南风 1 级
2015-06-18 周四	27°	16°	小雨	东风 2 级
2015-06-19 周五	21°	18°	阴~大雨	东南偏东风 1 级
2015-06-20 周六	30°	18°	多云	西北偏北风 1 级
2015-06-21 周日	21°	20°	小雨~中雨	东风 1 级
2015-06-22 周一	31°	21°	大雨~雷阵雨	西风 2 级
2015-06-23 周二	30°	22°	晴~多云	西北偏西风 2 级
2015-06-24 周三	32°	20°	雷阵雨	西南偏西风 2 级
2015-06-26 周五	28°	19°	多云~小雨	西南偏西风 1 级
2015-06-27 周六	27°	17°	多云	东北偏东风 2 级
2015-06-28 周日	21°	15°	小雨~中雨	东风 2 级
2015-06-29 周一	22°	16°	小雨~多云	西北偏西风 1 级
2015-06-30 周二	24°	19°	多云	西风 2 级

接着我们对每日出行人数进行汇总，并对最大气温和出行人数进行皮尔僧相关系数的计算，得到如下结果：

```
PS D:\Practise\dataProcess> python -u "d:\Practise\dataProcess\pearson.py"
0.24350337152503987
```

可以看到每日的温度和出行人数还是有一些关系的。

六、 感想总结

很感谢老师能够提供这样子的一次历练机会，从一开始的不知所措到慢慢地学习实践中我学到了很多。

由于在以前数学建模的时候使用过 python 进行数据处理，所以在这里我也同样使用了 python 语言来进行处理。不得不说，python 真的是一个非常方便的脚本语言，他能够很方便地对数据进行清洗和绘图。同时，它拥有的包也很多，通过调用有些包中的接口能画出非常精美的图像，而且调用起来也很简单方便。

当然，对于这些图像的分析，也是非常重要的，从我所画的这些图中，能够得到不少信息，这些信息将帮助到人们更合理地去分配共享单车也具有一定的社会研究意义。

同时，我还研究了数据挖掘和皮尔森相关系数的计算，对一些数据进行了相关性的计算，比如温度和出行人数的比较、骑行长度和性别年龄的区别等待，也得到了一些有意思的结论，让我领略到了数据处理的魅力。