Python 数据预处理第二次作业

班级：2018211313 班　　学号：2018211366　　姓名：蒋潇逸

版本：1.0

更新：*November 28, 2020*

本文档是 Python 数据预处理第二次作业报告。

目录

1 作业要求及内容 **2**

1.1 题目 1 2

1.2 题目 2 2

2 实验设备环境 **2**

3 题目 1 **2**

3.1 划分标准 2

3.2 代码实现 3

3.3 运行结果 4

3.4 结论分析 5

4 题目 2 **6**

4.1 要求 1 6

4.2 要求 2 7

4.3 要求 3 9

1 作业要求及内容

1.1 题目 1

对比分析北京、上海、广州、沈阳、成都，2015 年的空气质量，按照空气质量指数分级标准，计算出不同城市每个级别对应的天数各有多少，并给出综合的分析结论。

1.2 题目 2

处理和分析北京房价数据

- 对上周得到的北京市新房的房价数据，先进行异常值的处理，要求将总价在均值三倍标准差以外的房屋列出来，展示其基本信息（如果太多可以只展示一部分），并分析其原因（找 4 条即可）。
- 将总价在三倍标准差以内的数据，针对单价和总价两列进行 0-1 归一化及 Z-Score 归一化处理。结果使用散点图的形式表示。
- 将房屋的单价进行离散化处理，自行设定每个区间的长度并给出设置的理由，并给出每个区间的房屋数量和所占比例。

2 实验设备环境

Windows 10 专业版　　PyCharm 2019.3.3 x64　　编程语言 Python

3 题目 1

3.1 划分标准

查阅网上的资料可知，空气质量衡量标准如下表所示

表 1: 空气质量衡量标准

优秀	良好	轻度污染	中度污染	重度污染
≤ 50	≤ 100	≤ 200	≤ 300	> 300

3.2 代码实现

分析北京的空气质量代码如下所示，先打开对于的 csv 文件，求得 2015 年每天的 pm2.5 指数，再利用 cut 函数，对每天进行分级，并画出其饼状图，其他四个城市的处理类似，故不再列出。

```
import pandas as pd
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
# 打开CSV文件
fileNameStr = 'BeijingPM20100101_20151231.csv'
df = pd.read_csv(fileNameStr, encoding='utf-8')

# 查看数据集的基本情况
print("2:head============================")
print(df.head())
print("2:describe========================")
print(df.describe())
print("2:info============================")
print(df.info())

# 查看是否有缺失值
print("3=======================")
print(df.isnull().sum().sort_values(ascending=False))

df.dropna(axis=0, how='all', subset=['PM_Taiyuanjie', 'PM_US Post', '
    PM_Xiaoheyan'], inplace=True)

# 计算平均值
df['sum'] = df[['PM_Taiyuanjie', 'PM_US Post', 'PM_Xiaoheyan']].sum(
    axis=1)
df['count'] = df[['PM_Taiyuanjie', 'PM_US Post', 'PM_Xiaoheyan']].
    count(axis=1)
df['ave'] = round(df['sum'] / df['count'], 2)

# 按照年做汇总，查看年的平均值
df[df.year == 2015].groupby(["year", "month", "day"])["ave"].mean().
    to_csv("bj_ave_pm_day.csv")

```

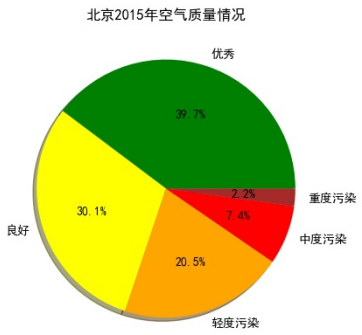
```
fileNameStr = 'bj_ave_pm_day.csv'
df = pd.read_csv(fileNameStr, encoding='utf-8')  # 不加dtype=str

sections = [0, 50, 100, 200, 300, 2000]  # 划分为不同长度的区间
section_names = ["优秀", "良好", "轻度污染", "中度污染", "重度污染"]
    # 设置每个区间的标签
result = pd.cut(df.ave, sections, labels=section_names)
print("-------------- result--------------")
print(result)
print("-------------- result type--------------")
print(type(result))
print("-------------- result count--------------")
count = pd.value_counts(result)  # 各部分的数量
print(count)  # 按照区间计数

plt.pie(count, labels=section_names, labeldistance=1.1, autopct="%1.1
   f%%", shadow=True, startangle=0, pctdistance=0.6,
        colors=["green", "yellow", "orange", "red", "Brown"])
plt.title("北京2015年空气质量情况")
plt.savefig("C:\\Users\\Lenovo\\Desktop\\bj.jpg")
plt.show()
```

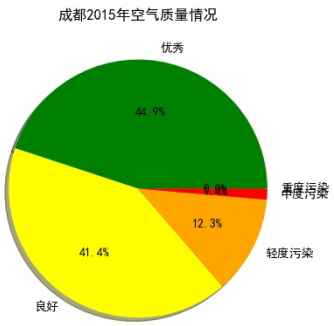
3.3 运行结果

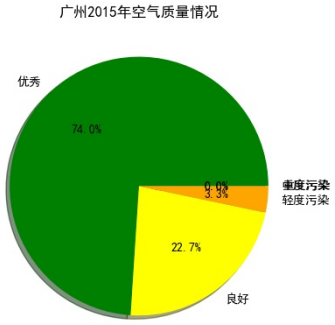
得到如下结果

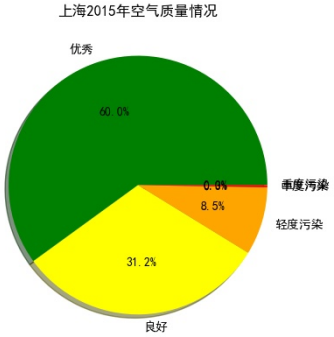
表 2: 各个城市的空气质量

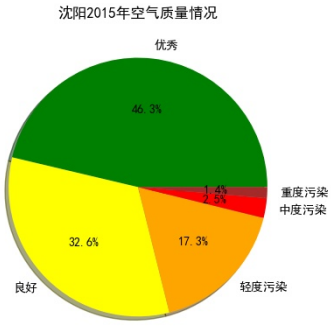
城市	优秀	良好	轻度污染	中度污染	重度污染
北京	145	110	75	27	8
成都	164	151	45	5	0
广州	270	83	12	0	0
上海	219	114	31	1	0
沈阳	169	119	63	9	5











3.4 结论分析

五个城市中，广州空气质量优秀的天数最多，相应的其污染天数也是最少的，故广州在这五个城市中空气质量最好，其次是上海，对于成都和沈阳两个城市来说，其空气质量优秀的天数相差不多，成都空气质量良好的天数比沈阳多，空气质量污染的天数比沈阳少，故综合来看，成都的空气质量比沈阳要好，北京的空气质量最差。故综上所述，各个城市的空气质量排名为广州，上海，成都，沈阳，北京。

4 题目 2

4.1 要求 1

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

# 打开CSV文件
fileNameStr = 'lianjia.csv'
df = pd.read_csv(fileNameStr, encoding='utf-8')  # 不加dtype=str
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签

# 查看数据集的基本情况
print("2:head=======================================")
print(df.head())
print("2:describe===================================")
print(df.describe())
print("2:info=======================================")
print(df.info())

# 计算出totalPrice列均值和标准差
# 的异常值，并将异常值的行号，放入一个列表
totalPrice_mean = df.totalPrice.mean()
totalPrice_std = df.totalPrice.std()
print(totalPrice_mean)
print(totalPrice_std)
print(totalPrice_mean - 3 * totalPrice_std, totalPrice_mean + 3 *
    totalPrice_std)

# 求出totalPrice列中数据在3倍三倍标准差之外的数据
index_list = df[(df.totalPrice > totalPrice_mean + 3 * totalPrice_std
    ) | (df.totalPrice < totalPrice_mean - 3 * totalPrice_std)].index.
    tolist()
value_list = df[(df.totalPrice > totalPrice_mean + 3 * totalPrice_std
    ) | (df.totalPrice < totalPrice_mean - 3 * totalPrice_std)]

```

```
print("there are {} items:".format(len(index_list)))
print(index_list)
print(value_list)
```

```
882.6033983957218
851.500274091864
-1671.89742387987 3437.1042206713137
there are 4 items:
[68, 76, 148, 173]
       name position_1 position_2  ...    area unitPrice  totalPrice
68    懋源·璟岳         丰台         玉泉营  ...   465.0  113000.0      5254.5
76   北京壹号总部         大兴          亦庄  ...     NaN   28000.0      5777.0
148   远洋新天地         门头沟       门头沟其它  ...  1118.0   33000.0      3689.4
173    润泽御府         朝阳          北苑  ...   540.0  100000.0      5400.0
```

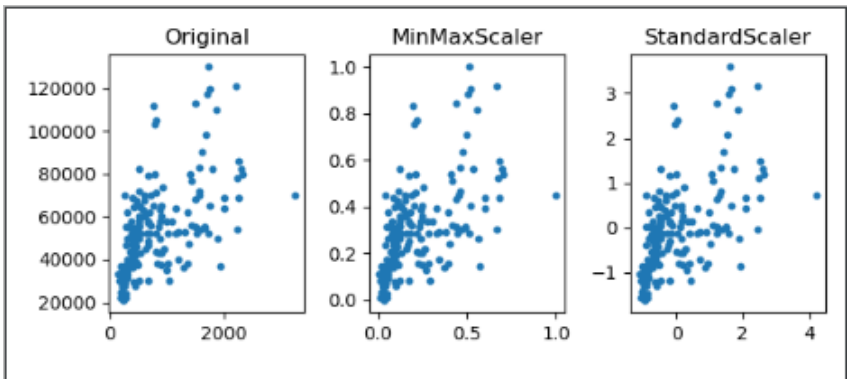
	A	B	C	D	E	F	G	H	I
1	name	position_1	position_2	position_3	room	area	unitPrice	totalPrice	
39	润泽御府	朝阳	北苑	北京市朝阳区北五环顾家庄桥向北约2.6公里	4室	540	100000	5400.0000	
42	懋源·璟岳	丰台	玉泉营	南三环西路99号院	4室	465	113000	5254.5000	
95	远洋新天地	门头沟	门头沟其它	长安街西延线与清河路南延交汇处（东南侧）	1室	1118	33000	3689.4000	
177	北京壹号总部	大兴	亦庄	台湖镇光机电一体化产业基地科创东二街5号	null	null	28000	5777	
189									

原因：这四个楼盘的面积都比较大导致总价比较贵，特别是远洋新天地，虽然该楼盘的单价只有 33000 元，但是它的占地面积到达了 1118 平米。

4.2 要求 2

```
df["totalPrice_new"] = df["totalPrice"]
for i in index_list:
    df["totalPrice_new"][i] = np.nan

fig = plt.figure()

#子图1：原始图像
x1 = df["totalPrice_new"]
y1 = df["unitPrice"]
ax1 = fig.add_subplot(231)
ax1.scatter(x1, y1, s=10)
ax1.set_title("Original")

# 子图2:(0,1)归一化，采用MinMaxScaler函数
ax2 = fig.add_subplot(232)
min = x1.min()
max = x1.max()
ave = x1.mean()
std = x1.std()
```

```
x2 = (x1 - min) / (max - min)

scaler = MinMaxScaler()
y_reshape = y1.values.reshape(-1, 1)   # 变成n行1列的二维矩阵形式
y2 = scaler.fit_transform(y_reshape)   # 调用MinMaxScaler的
    fit_transform转换方法，进行归一化处理

ax2.scatter(x2, y2, s=10)
ax2.set_title("MinMaxScaler")

# 子图3:Z-score归一化，采用StandardScaler函数
ax3 = fig.add_subplot(233)

scaler_std = StandardScaler()
x_reshape = x1.values.reshape(-1, 1)
x3 = scaler_std.fit_transform(x_reshape)

y_reshape = y1.values.reshape(-1, 1)
y3 = scaler_std.fit_transform(y_reshape)

ax3.scatter(x3, y3, s=10)
ax3.set_title("StandardScaler")

plt.show()
```

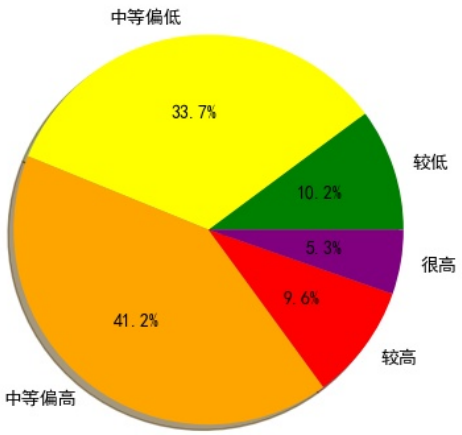


4.3 要求 3

表 3: 房价衡量标准以及数量

	较低	中等偏低	中等偏高	较高	很高
	≤ 30000	≤ 50000	≤ 70000	≤ 100000	> 100000
数量	19	63	77	18	10
比例	10.2%	33.7%	41.2%	9.6%	5.3%

```
sections = [0, 30000, 50000, 70000, 100000, 1000000]  # 划分为不同长度的区间
section_names = ["较低", "中等偏低", "中等偏高", "较高", "很高"]  # 设置每个区间的标签
result = pd.cut(df['unitPrice'], sections, labels=section_names)
print("-------------- result--------------")
print(result)
print("-------------- result type--------------")
print(type(result))
print("-------------- result count--------------")
count = pd.value_counts(result)  # 各部分的数量
print(count)
plt.pie(count, labels=section_names, labeldistance=1.1, autopct="%1.1f%%", shadow=True, startangle=0, pctdistance=0.6,
        colors=["green", "yellow", "orange", "red", "purple"])
plt.show()
```



房价衡量标准设置理由：房价应该尽量服从正态分布。设置的衡量标准，应该满足在此标准下，房价呈现中间多两头少的趋势，根据链家爬取到达数据，大部分房价位于3万到7万之间，故这两数取平均，分为3万到5万中等偏低，5万到7万中等偏高，低于3万即为较低。由于高于7万的房价相差较多，最高能达到13万每平，故在此基础上再以10万为界线分为7万到10万较高和大于10万很高。