python 中的数据处理 2

杨晨
学号 2021212171
北京邮电大学计算机学院

日期：2024 年 2 月 26 日

1 概述

1.1 实验内容

处理北京空气质量数据。

1. 对 PM 指数进行异常值的处理：假设 PM 指数最高为 500，将 PM_Dongsi、PM_Dongsihuan、PM_Nongzhanguan 三列中超过 500 的数据，修改为 500。
2. 对 PRES 和 TEMP 数据进行最大最小归一化和标准化归一化，并使用散点图进行展示。
3. 针对北京每天的 PM 平均值（对多个测试站点和多个时间的值求平均），统计不同颜色代表的指数等级（指数等级见课件第 23 页）各有多少天。

1.2 开发环境

- Windows10
- PyCharm 2023.2.4 (Professional Edition)

2 实验过程

2.1 对 PM 异常值的处理

数据预处理的第一步是将数据集加载到 Pandas 的 DataFrame 中。检查三个地点（PM_Dongsi、PM_Dongsihuan、PM_Nongzhanguan）的 PM2.5 浓度值是否超过 500，如果超过则用 500 代替。

```
df = pd.read_csv('BeijingPM20100101_20151231.csv', encoding='utf-8')

# 将PM_Dongsi、PM_Dongsihuan、PM_Nongzhanguan三列中超过500的数据，修改为500。
df.loc[df['PM_Dongsi'] > 500, 'PM_Dongsi'] = 500
df.loc[df['PM_Dongsihuan'] > 500, 'PM_Dongsihuan'] = 500
df.loc[df['PM_Nongzhanguan'] > 500, 'PM_Nongzhanguan'] = 500
```

然后将处理过的数据保存到 CSV 文件中。

```
# 保存处理后的数据
df.to_csv('data.csv', index=False)
```

2.2 归一化

接下来，使用最小最大归一化和标准化归一化方法对气压（PRES）和温度（TEMP）变量进行归一化处理。最小-最大归一化将数值映射到 0 和 1 之间的范围，而标准化归一化则将数据转换为平均值为 0，标准偏差为 1。

```
# 对PRES和TEMP数据进行最大最小归一化和标准化归一化
x_reshape = df['PRES'].values.reshape(-1, 1)
y_reshape = df['TEMP'].values.reshape(-1, 1)

# 最大最小归一化
min_max_scaler = MinMaxScaler()
normalized_pres = min_max_scaler.fit_transform(x_reshape)
normalized_temp = min_max_scaler.fit_transform(y_reshape)

# 标准化归一化
standard_scaler = StandardScaler()
standardized_pres = standard_scaler.fit_transform(x_reshape)
standardized_temp = standard_scaler.fit_transform(y_reshape)
```

为了直观显示变量之间的关系，绘制了散点图。生成了三个散点图，分别代表不同的归一化方法：原始、最大最小归一化和标准化归一化。每个散点图的 x 轴代表气压（PRES），y 轴代表温度（TEMP）。通过散点图，可以分析不同归一化技术下气压和温度之间的相关性。

```
# 绘制散点图
fig, axes = plt.subplots(1, 3, figsize=(15, 8))
axes[0].scatter(df['PRES'], df['TEMP'], marker='o')
axes[1].scatter(normalized_pres, normalized_temp, marker='o')
axes[2].scatter(standardized_pres, standardized_temp, marker='o')
axes[0].set_xlabel('PRES')
axes[0].set_ylabel('TEMP')
axes[0].set_title('Original')
axes[1].set_xlabel('normalized PRES')
axes[1].set_ylabel('normalized TEMP')
axes[1].set_title('Normalized')
axes[2].set_xlabel('standardized PRES')
axes[2].set_ylabel('standardized TEMP')
axes[2].set_title('Standardized')
fig.tight_layout()
plt.savefig('scatter.png')
plt.show()
```

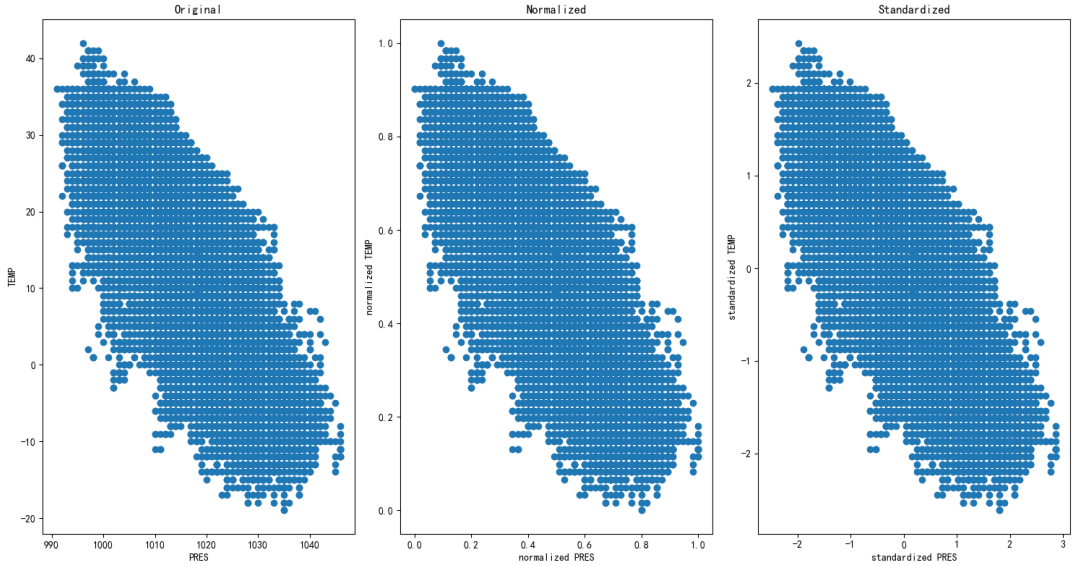
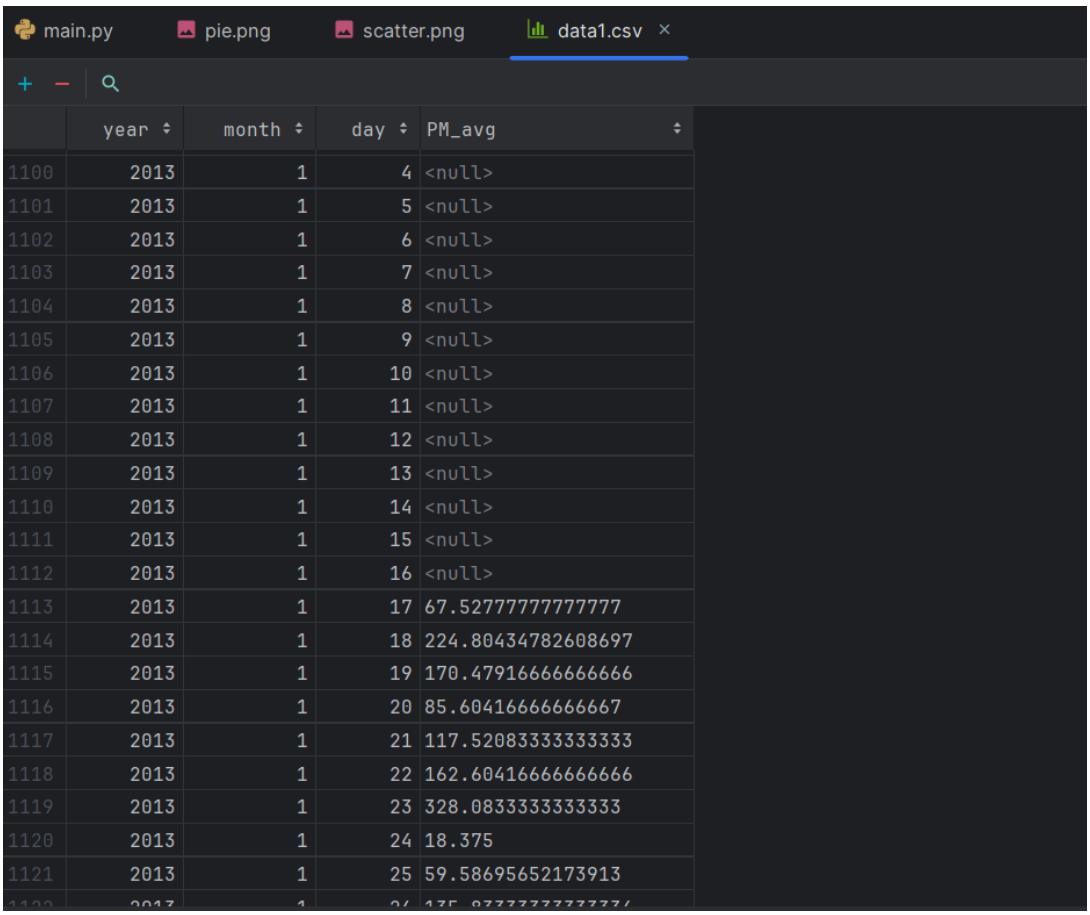


图 1: 散点图

2.3 统计污染情况

通过取三个地点 PM2.5 浓度的平均值，创建了一个新列 PM_avg。然后按年、月、日对数据集进行分组，并计算出每组的 PM_avg 的平均值。处理后的数据被保存，只保留年、月、日和 PM_avg 列。

```
# 新建一列，将PM_Dongsi、PM_Dongsihuan、PM_Nongzhanguan三列的数据进行平均，赋值给新建的列PM_avg
df['PM_avg'] = df[['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan']].mean(axis=1)
# 按照year、month、day进行分组，求PM_avg的平均值
df = df.groupby(['year', 'month', 'day'])['PM_avg'].mean()
# 保存处理后的数据，只保留year、month、day和PM_avg四列
df.to_csv('data1.csv', index=True, header=True)
```



main.py pie.png scatter.png data1.csv

	year	month	day	PM_avg
1100	2013	1	4	<null>
1101	2013	1	5	<null>
1102	2013	1	6	<null>
1103	2013	1	7	<null>
1104	2013	1	8	<null>
1105	2013	1	9	<null>
1106	2013	1	10	<null>
1107	2013	1	11	<null>
1108	2013	1	12	<null>
1109	2013	1	13	<null>
1110	2013	1	14	<null>
1111	2013	1	15	<null>
1112	2013	1	16	<null>
1113	2013	1	17	67.52777777777777
1114	2013	1	18	224.80434782608697
1115	2013	1	19	170.47916666666666
1116	2013	1	20	85.60416666666667
1117	2013	1	21	117.52083333333333
1118	2013	1	22	162.60416666666666
1119	2013	1	23	328.0833333333333
1120	2013	1	24	18.375
1121	2013	1	25	59.58695652173913

图 2: data1.csv

为了对污染程度进行分类，PM 平均值被分为不同的部分：0-50、51-100、101-150、151-200、201-300 和 301-500。这些区段分别被标记为” 优”、” 良”、” 轻度污染”、” 中度污染”、” 重度污染” 和” 严重污染”。计算了属于每个污染等级类别的天数，并用饼状图将其直观显示出来。

```
sections = [0, 50, 100, 150, 200, 300, 501]
sections_color = ['green', 'yellow', 'orange', 'red', 'purple', 'maroon']
sections_name = ['优', '良', '轻度污染', '中度污染', '重度污染', '严重污染']
result = pd.cut(df, sections, labels=sections_name)
# 统计各个污染程度的天数
print(pd.value_counts(result))
# 绘制饼图
plt.figure(figsize=(6, 6))
plt.pie(pd.value_counts(result), labels=sections_name, colors=sections_color,
    autopct='%1.1f%%')
plt.title('Beijing PM2.5')
plt.savefig('pie.png')
plt.show()
```

统计结果如下：

```
优        375
良        337
```

```
轻度污染    191
中度污染     77
重度污染     74
严重污染     25
Name: PM_avg, dtype: int64
```

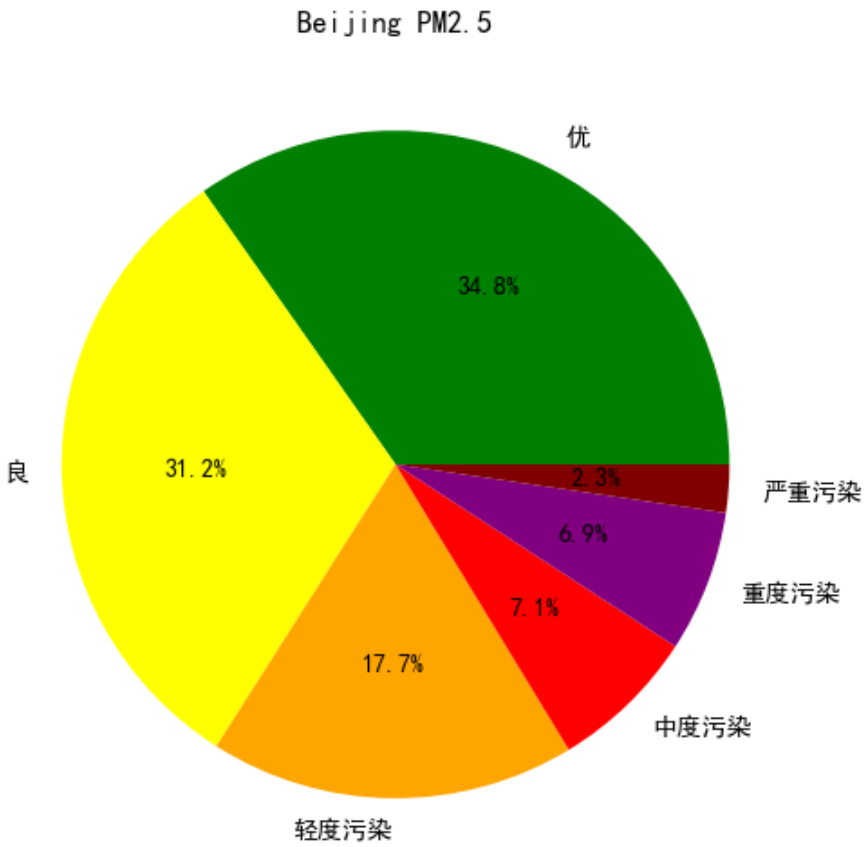


图 3: 饼图

3 附录：完整代码

```python
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler
import matplotlib.pyplot as plt
import matplotlib

# 设置中文显示
matplotlib.rc("font", family="SimHei", weight="bold")
plt.rcParams["axes.unicode_minus"] = False
```

```
df = pd.read_csv('BeijingPM20100101_20151231.csv', encoding='utf-8')

# 将PM_Dongsi、PM_Dongsihuan、PM_Nongzhanguan三列中超过500的数据，修改为500。
df.loc[df['PM_Dongsi'] > 500, 'PM_Dongsi'] = 500
df.loc[df['PM_Dongsihuan'] > 500, 'PM_Dongsihuan'] = 500
df.loc[df['PM_Nongzhanguan'] > 500, 'PM_Nongzhanguan'] = 500

# 保存处理后的数据
df.to_csv('data.csv', index=False)

# 对PRES和TEMP数据进行最大最小归一化和标准化归一化
x_reshape = df['PRES'].values.reshape(-1, 1)
y_reshape = df['TEMP'].values.reshape(-1, 1)

# 最大最小归一化
min_max_scaler = MinMaxScaler()
normalized_pres = min_max_scaler.fit_transform(x_reshape)
normalized_temp = min_max_scaler.fit_transform(y_reshape)

# 标准化归一化
standard_scaler = StandardScaler()
standardized_pres = standard_scaler.fit_transform(x_reshape)
standardized_temp = standard_scaler.fit_transform(y_reshape)

# 绘制散点图
fig, axes = plt.subplots(1, 3, figsize=(15, 8))
axes[0].scatter(df['PRES'], df['TEMP'], marker='o')
axes[1].scatter(normalized_pres, normalized_temp, marker='o')
axes[2].scatter(standardized_pres, standardized_temp, marker='o')
axes[0].set_xlabel('PRES')
axes[0].set_ylabel('TEMP')
axes[0].set_title('Original')
axes[1].set_xlabel('normalized PRES')
axes[1].set_ylabel('normalized TEMP')
axes[1].set_title('Normalized')
axes[2].set_xlabel('standardized PRES')
axes[2].set_ylabel('standardized TEMP')
axes[2].set_title('Standardized')
fig.tight_layout()
plt.savefig('scatter.png')
plt.show()

# 新建一列，将PM_Dongsi、PM_Dongsihuan、PM_Nongzhanguan三列的数据进行平均，赋值给新
    建的列PM_avg
df['PM_avg'] = df[['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan']].mean(axis=1)
# 按照year、month、day进行分组，求PM_avg的平均值
```

```
df = df.groupby(['year', 'month', 'day'])['PM_avg'].mean()
# 保存处理后的数据,只保留year、month、day和PM_avg四列
df.to_csv('data1.csv', index=True, header=True)

# 污染程度划分
sections = [0, 50, 100, 150, 200, 300, 501]
sections_color = ['green', 'yellow', 'orange', 'red', 'purple', 'maroon']
sections_name = ['优', '良', '轻度污染', '中度污染', '重度污染', '严重污染']
result = pd.cut(df, sections, labels=sections_name)
# 统计各个污染程度的天数
print(pd.value_counts(result))
# 绘制饼图
plt.figure(figsize=(6, 6))
plt.pie(pd.value_counts(result), labels=sections_name, colors=sections_color,
    autopct='%1.1f%%')
plt.title('Beijing PM2.5')
plt.savefig('pie.png')
plt.show()
```