1. **问题描述**
2. 处理和分析北京空气质量数据

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. 处理和分析北京房价数据
2. 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）。
3. 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（参考PPT第21页图形上半部分的表现形式）。
4. 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。
5. **实验环境**

Microsoft Windows 10 版本18363

PyCharm 2020.2.1 (Community Edition)

Python 3.8(Scrapy 2.4.0 + numpy 1.19.4 + pandas 1.1.4)

1. **实验步骤及结果**
2. 处理和分析北京空气质量数据

from matplotlib import pyplot as plt  
import numpy as np  
import pandas as pd  
  
from pylab import \*  
mpl.rcParams['font.sans-serif'] = ['SimHei']  
  
section = [0, 35, 75, 115, 150, 250, 99999]  
section\_name = ["优", "良", "轻度污染", "中度污染", "重度污染", "严重污染"]  
labels = ['北京', '成都', '广州', '上海', '沈阳']  
  
def classify(s):  
 fileNameStr = s  
 orig\_df = pd.read\_csv(fileNameStr, encoding='utf-8', dtype=str)  
 orig\_df = pd.DataFrame(orig\_df, columns=['year', 'month', 'day', 'PM\_US Post'])  
 df = orig\_df.dropna(0, how='any')  
  
 df['PM\_US Post'] = df['PM\_US Post'].astype(int)  
  
 result = pd.cut(df['PM\_US Post'], section, labels=section\_name, ordered=False)  
 # print(result)  
 return pd.DataFrame(pd.value\_counts(result, sort=False))  
  
  
data\_bj = classify('./BeijingPM20100101\_20151231.csv')  
print(data\_bj.describe())  
data\_cd = classify('./ChengduPM20100101\_20151231.csv')  
data\_gz = classify('./GuangzhouPM20100101\_20151231.csv')  
data\_sh = classify('./ShanghaiPM20100101\_20151231.csv')  
data\_sy = classify('./ShenyangPM20100101\_20151231.csv')  
  
x = np.arange(len(section\_name))  
width = 0.1  
fig, ax = plt.subplots(figsize=(8, 4))  
rects1 = ax.bar(x-2\*width, np.array(data\_bj['PM\_US Post']), width, label='bj')  
rects2 = ax.bar(x-width, np.array(data\_cd['PM\_US Post']), width, label='cd')  
rects3 = ax.bar(x, np.array(data\_gz['PM\_US Post']), width, label='gz')  
rects4 = ax.bar(x+width, np.array(data\_sh['PM\_US Post']), width, label='sh')  
rects5 = ax.bar(x+2\*width, np.array(data\_sy['PM\_US Post']), width, label='sy')  
plt.xticks(range(0, 6), section\_name)  
  
plt.show()

北京：



成都：



广州：

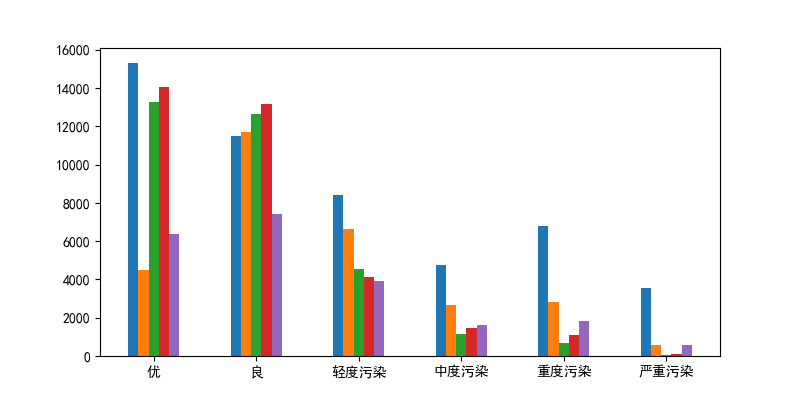


上海：



沈阳：





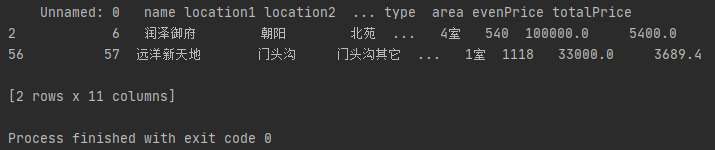
蓝色、橘色、绿色、红色、紫色分别代表北京、成都、广州、上海、沈阳。

可以看出成都、沈阳的污染情况主要为良，相比其他城市的平均情况都较劣。

北京的污染较轻天数较多，但占比较小，因为中度污染及严重污染的天数也比其他城市要多得多，两极分化。

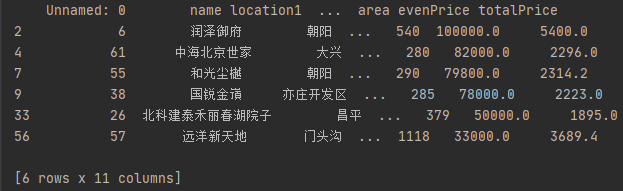
1. 处理和分析北京房价数据
2. 对上周得到的北京市新房的房价数据，先进行异常值的处理，要求将总价在均值三倍标准差以外的房屋列出来，展示其基本信息（如果太多可以只展示一部分），并分析其原因（找4条即可）。

import pandas as pd  
  
fileNameStr = './result-handled.csv'  
df = pd.read\_csv(fileNameStr, encoding='utf-8')  
  
total\_min = df['totalPrice'].mean()  
total\_std = df['totalPrice'].std()  
total\_range\_min = total\_min - 3 \* total\_std  
total\_range\_max = total\_min + 3 \* total\_std  
  
newdf = df[df['totalPrice'] <= total\_range\_max]  
newdf = newdf[newdf['totalPrice'] >= total\_range\_min]  
  
print(newdf.append(df).drop\_duplicates(keep=False))



找到该两处房源总价在三倍标准差外。其面积都在500平、甚至1000平以上，不属于常规房源，应该为别墅。

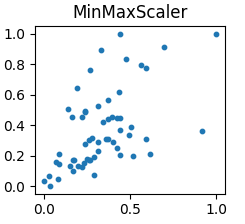
由于题目需要找到四条，但是没有爬那么多的房源数据，因而将标准降到一个标准差，以下为结果



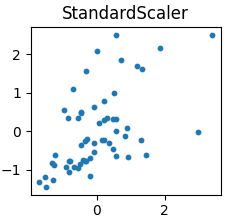
可以看到，与之前的结果类似，这几处房源也属于超大户型的，不属于普通房源，因而总价相比均值差距较大。

1. 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（参考PPT第21页图形上半部分的表现形式）。

0-1归一化：

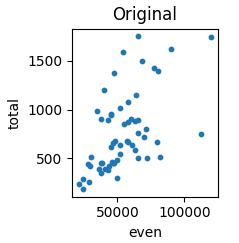


Z-Score归一化：



df = newdf  
  
#生成fig对象和ax对象  
fig = plt.figure()  
#ax1 = fig.add\_subplot(111)  
ax1 = fig.add\_subplot(231)  
  
#子图1：原始图像  
x1 = df["evenPrice"] # PRES\_new2  
y1 = df["totalPrice"]  
ax1.scatter(x1, y1, s=10)  
#ax1.set\_xlim(0,1250)  
#ax1.set\_ylim(0,1250)  
ax1.set\_xlabel("even")  
ax1.set\_ylabel("total")  
ax1.set\_title("Original")  
print(type(x1))  
  
  
#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)  
  
from sklearn.preprocessing import MinMaxScaler  
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")  
  
#查看单个特征，在归一化之后的分布有何变化。  
ax4 = fig.add\_subplot(234)  
ax5 = fig.add\_subplot(235)  
ax6 = fig.add\_subplot(236)  
ax4.set\_title("Orig")  
ax5.set\_title("MinMax")  
ax6.set\_title("Standard")  
def count\_elements(scores): #定义转换函数，统计每个数值对应多少个  
 scorescount = {} #定义一个字典对象  
 for i in scores:  
 scorescount[int(i)] = scorescount.get(int(i), 0) + 1 #累加每个分数值的人数  
 return scorescount  
  
def count\_elements1(scores): #定义转换函数，统计每个数值对应多少个  
 scorescount = {} #定义一个字典对象  
 for i in scores:  
 scorescount[int(i\*100)] = scorescount.get(int(i\*100), 0) + 1 #累加每个分数值的人数  
 return scorescount  
  
counted1 = count\_elements(y1) #可以改为y1,2,3，看看湿度的分布情况  
counted2 = count\_elements1(y2) #0,1  
counted3 = count\_elements1(y3) #standard  
  
ax4.bar(counted1.keys(),counted1.values(),0.8,alpha=0.5,color='r')  
ax5.bar(counted2.keys(),counted2.values(),0.8,alpha=0.5,color='r')  
ax6.bar(counted3.keys(),counted3.values(),0.8,alpha=0.5,color='r')  
  
#显示图形  
plt.show()

1. 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。



观察原始图像知，斜率即为面积，因而可以根据户型大小进行离散化处理。根据《财政部国家税务总局住房和城乡建设部关于调整房地产交易环节契税个人所得税优惠政策的通知》(财税[2010]94号)，按照户型大小规定了不同户型的契税比率，分类如下：

小户型：<90m2

中户型：90~144m2

大户型：>144m2

section = [0, 90, 144, 99999]  
labels = ['小户型', '中户型', '大户型']  
df['totalPrice'] = df['totalPrice'].astype(int)  
df['evenPrice'] = df['evenPrice'].astype(int)  
df['cal'] = df['totalPrice'] \* 10000 / df['evenPrice']  
  
df['sizetype'] = pd.cut(df['cal'], section, labels=labels, ordered=False)  
df = df.sort\_values(by=['sizetype', 'cal'])  
  
print(df)  
  
df.to\_csv("result-classified.csv", encoding="utf-8")

分组结果为：

 百分比分别为16.129%、41.935%、41.935%



1. **心得与体会**

在本次实验中，通过两道题目的引导，我了解到了python中利用pandas进行数据分析及处理的基本方法，以及使用sklearn包进行的一些简单的数据操作，特别是根据数据值进行分类、异常值的查找、两种归一化、数据离散化的分析方法。非常感谢老师的实验设计！