



**《物联网技术与大数据导论》实验报告**

**学 院**：

**班 级**：

**学 号**：

**姓 名**：

**指导教师**：

教 务 处

2022 年 4 月

题目

**实验2根据给定的AXP\_NAN文件进行相应的操作，可参考视频**

1. 将AXP\_NAN数据读入，并赋值给变量pfnan
2. 利用isnull()函数判定pfnan是否包含缺失值。
3. 利用dropna()函数删除包含缺失值的行。
4. 再次将AXP\_NAN数据读入，并赋值给变量pfnan，然后利用SimpleImputer()函数填充数据，填充数据为中位数值(median()函数)。
5. 利用decribe()函数观测数据。
6. 利用loc()函数将pfnan的最后一条数据改为[3,200,1,2,3,4],并利用iloc选择前4列数据显示。
7. 利用drop(‘Volume’，axis=1)和boxplot显示箱图。
8. 利用3σ方法判定缺失值。公式如下:数据与平均值(mean())差的绝对值大于数据标准差(std())的3倍。

**实验3 根据数据集boston()完成以下操作,可参考**

1. 调用sklearn模块下的datasets数据集。
2. 将boston()数据库赋值给bo。(使用datasets.load\_boston)
3. 利用bo的4,5,6这3列构造dataframe类型数据df，并利用feature\_names()函数将bo的列名称给df。
4. 利用公式(df-df.min)/(df.max-df.min)来对df数据进行规范化,并赋值给df2。
5. 调用sklearn的preprocessing模块，并利用其中的minmax\_scale函数对df进行规范化，并赋值给df3。
6. 利用preprocessing模块中的scale()函数对df数据进行规范化，并赋值给df4。

代码：

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import preprocessing

from sklearn import datasets

from sklearn.impute import SimpleImputer

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

pfnan = pd.read\_csv('AXP\_NAN.csv', encoding='gbk').set\_index(["Date"])

# 201911130237

# 李华龙

pfnan.isnull().any()

pfnan[pfnan.isnull().values==True]

pfnan1 = pfnan.dropna(axis=0, how='any')

pfnan1

imp = SimpleImputer(missing\_values=np.nan, strategy='mean')

pfnan2 = pd.DataFrame(imp.fit\_transform(pfnan),index = pfnan.index, columns = pfnan.columns)

pfnan2

pfnan2.describe(include='all')

pfnan3 = pfnan.copy()

pfnan3.loc["2020/4/22",:] = np.array([3,200,1,2,3,4])

pfnan3

pfnan3.head(4)

pfnan4 = pfnan2.drop(['Volume'],axis=1)

pfnan4

fig = plt.subplot(1,1,1)

bplot = fig.boxplot(pfnan4.T,

notch=True,

vert=True,

patch\_artist=True)

#颜色填充

colors = ['lightgreen', 'lightblue']

for patch, color in zip(bplot['boxes'], colors):

patch.set\_facecolor(color)

fig.yaxis.grid(True) #在y轴上添加网格线

#fig.set\_xticks([y+1 for y in range(len(pfnan4.T))] )

fig.set\_xlabel('xlabel') #设置x轴名称

fig.set\_ylabel('ylabel') #设置y轴名称

# 加刻度名称

plt.setp(fig, xticks=[1,2,3,4,5],

xticklabels=list(pfnan4.columns))

plt.show()

for index, row in pfnan.iterrows():

#设定法则的左右边界

left=row.mean()-3\*row.std()

right=row.mean()+3\*row.std()

#获取在范围内的数据

new\_num=row[(left<row)&(row<right)]

if new\_num.shape != (6,):

print(index)

bo = datasets.load\_boston()

df = pd.DataFrame(bo.data,columns = bo.feature\_names).iloc[:,4:7]

df.head()

df2 = (df-df.min())/(df.max()-df.min())

df2

min\_max\_scaler = preprocessing.MinMaxScaler()

df3 = min\_max\_scaler.fit\_transform(df)

df3

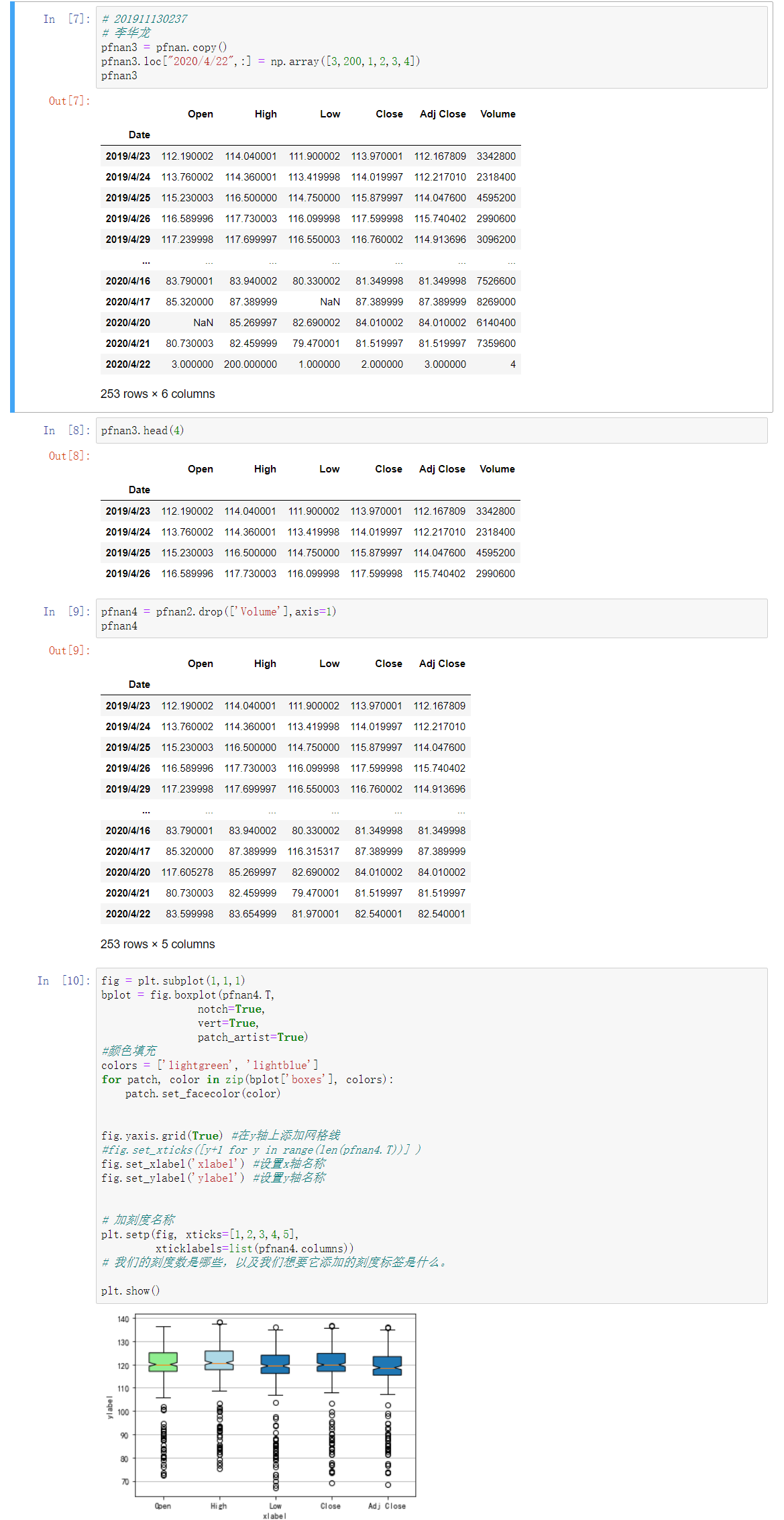
df4 = preprocessing.scale(df, axis=0)

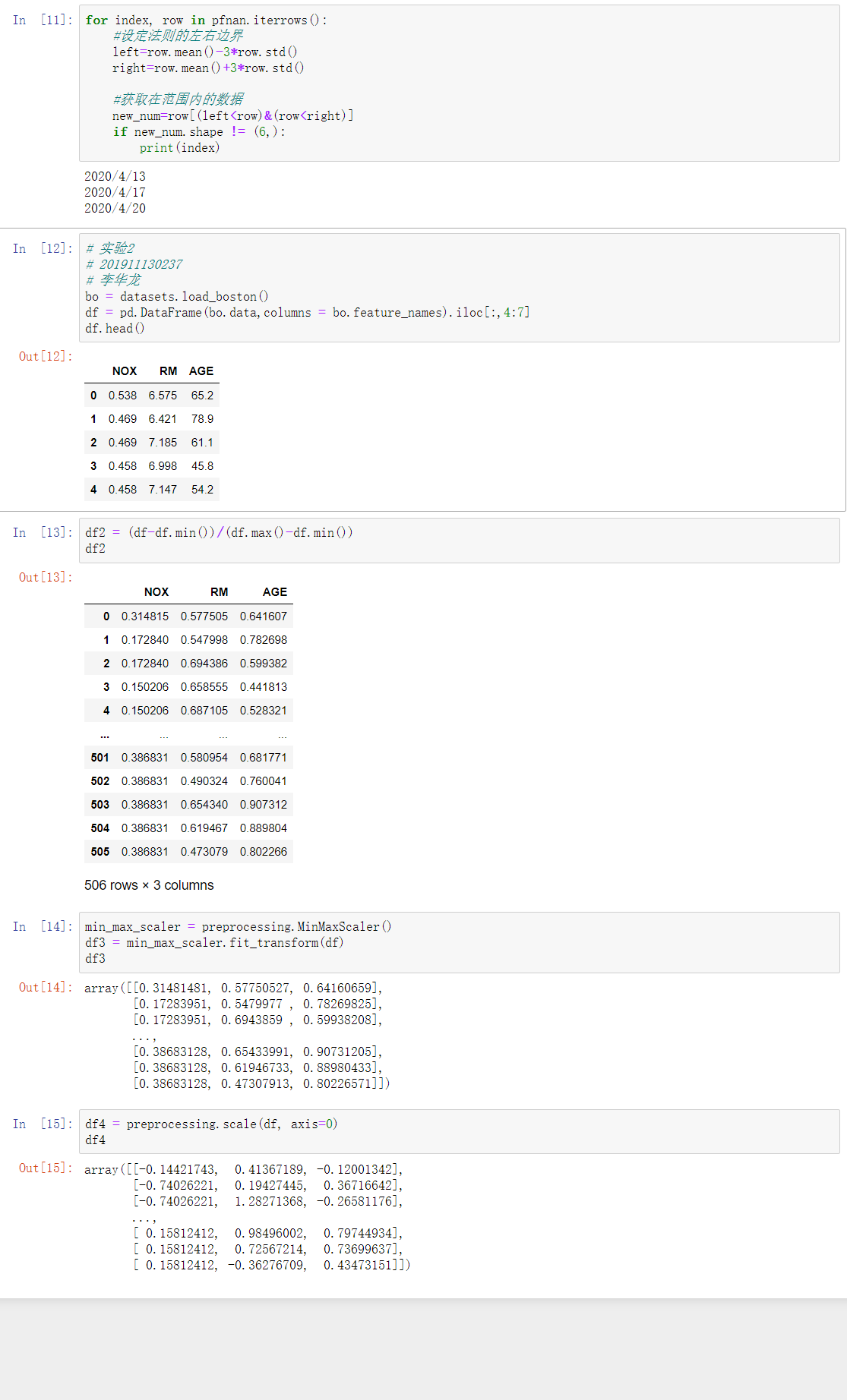
df4

结果与分析：









实验过程记录：

实验过程较为顺畅，不懂的也通过搜索引擎查询方法案例解决了。