**实验5 Python科学计算实践**

**姓名：宁浩 学号：2312190617 班级：软件2301 成绩：**

**1. 实验内容与目标**

本次实验旨在了解和掌握python科学计算工具，包括numpy和scipy的数值计算、pandas数据分析、matplotlib画图等，具备快速高效的数据处理和一定的可视化分析能力。需掌握的相关知识点主要包括：

1) numpy数组array的创建、使用，数组元素的切片，各种常见函数、通用函数（universal function）的运用，np.random模块组织和获取随机数据，以及numpy对文件数据的处理方法np.loadtxt()和np.savetxt()。了解微积分、线性代数、最优化等领域相关的scipy科学计算方法；

2) pandas的数据结构：Series和DataFrame，以及相关数据探索分析方法;

3) matplotlib画图：折线图、散点图、柱状图、直方图、饼图，以及多子图布局方式。（注：请自学matplotlib绘图库，还可以包括*seaborn*）

请完成下列实验练习题，报告书写要求同第1次实验作业。注意，请在本报告中将自己的姓名、学号、班级书写正确, 并在规定时间内完成本次实验，将报告的word文档(非pdf版本)以附件形式提交至学习通平台。

**2. 实验练习题**

**答题要求**：将正确运行的**源代码书写在题目下方**，并紧跟着贴上一份**源代码的截图**和运行结果的**截图**。答题格式同第1次实验作业。

(1)使用Scipy模块找到以下函数的最小值：。

源代码：

from scipy.optimize import minimize

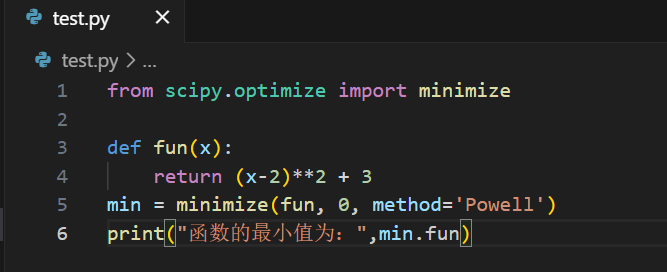
def fun(x):

    return (x-2)\*\*2 + 3

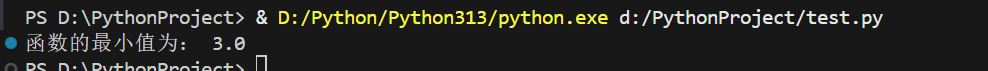
min = minimize(fun, 0, method='Powell')

print("函数的最小值为：",min.fun)

源代码截图：



运行结果截图：



(2)在同一张图中绘制函数。

源代码：

from matplotlib import pyplot as plt

import numpy as np

x = np.linspace(0, 5, 100)

plt.plot(x, np.sin(x), label='sin(x)')

plt.plot(x, np.log(x), label='log(x)')

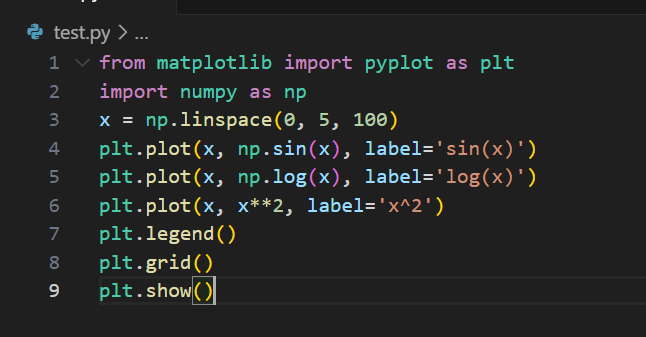
plt.plot(x, x\*\*2, label='x^2')

plt.legend()

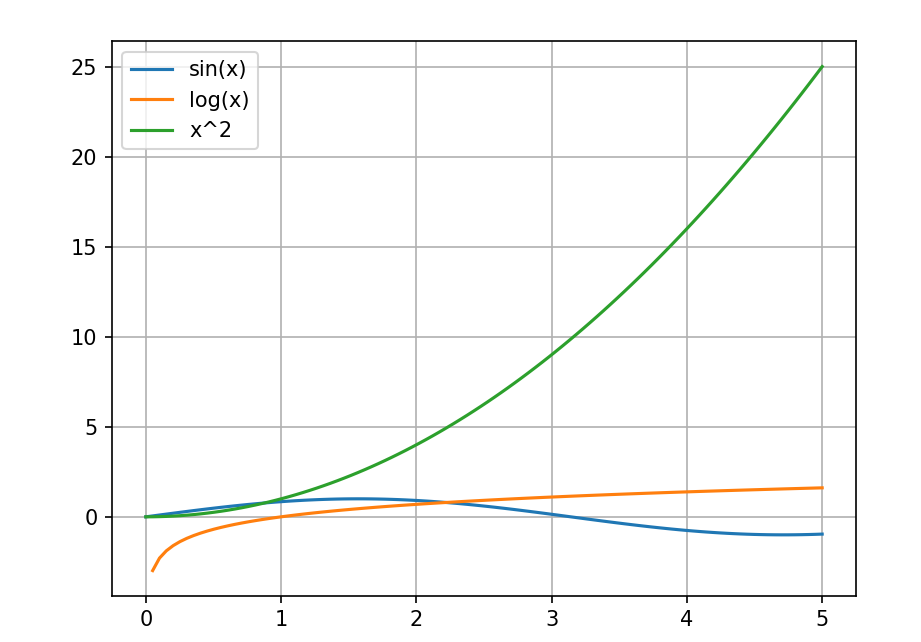
plt.grid()

plt.show()

源代码截图：



运行结果截图：



(3)数组元素索引。利用numpy.random模块随机产生一个9行10列的二维数组，其元素要求服从标准正态分布。完成以下该数组的索引任务，打印输出相关结果：

(a) 使用索引方式获取第2行第5列、第6行第3列的元素。

(b) 使用切片方式获取第3行至第5行和第4列至第6列的数据。

(c) 使用切片与整数序列索引混合的方式，获取第3行至第5行且为第1列、第2列和第4列的数据。

(d) 使用布尔索引方法将数组中取值大于1的元素重新赋值为10.00，小于-1的元素重新赋值为-10.00，并打印输出这个新数组。

源代码：

import numpy as np

a=np.random.rand(9,10)

print("第2行第5列元素为:",a[1,4],"第6行第3列的元素为:",a[5,2])

print("第3行至第5行元素为:\n",a[2:5],"\n第4列至第6列的元素为:\n",a[:,3:6])

print("第3行至第5行且为第1列、第2列和第4列的数据为:\n",a[2:5,[0,2,3]])

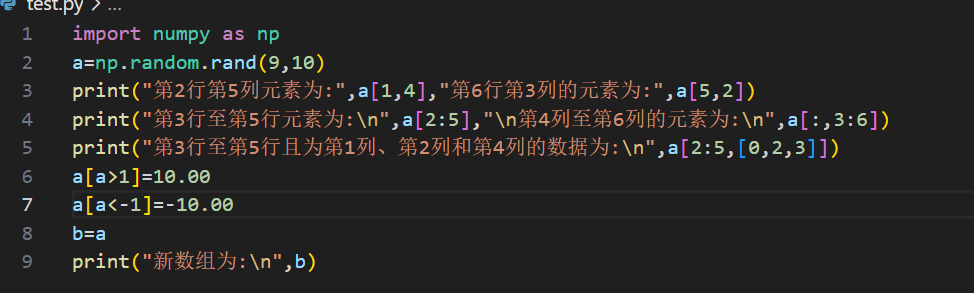
a[a>1]=10.00

a[a<-1]=-10.00

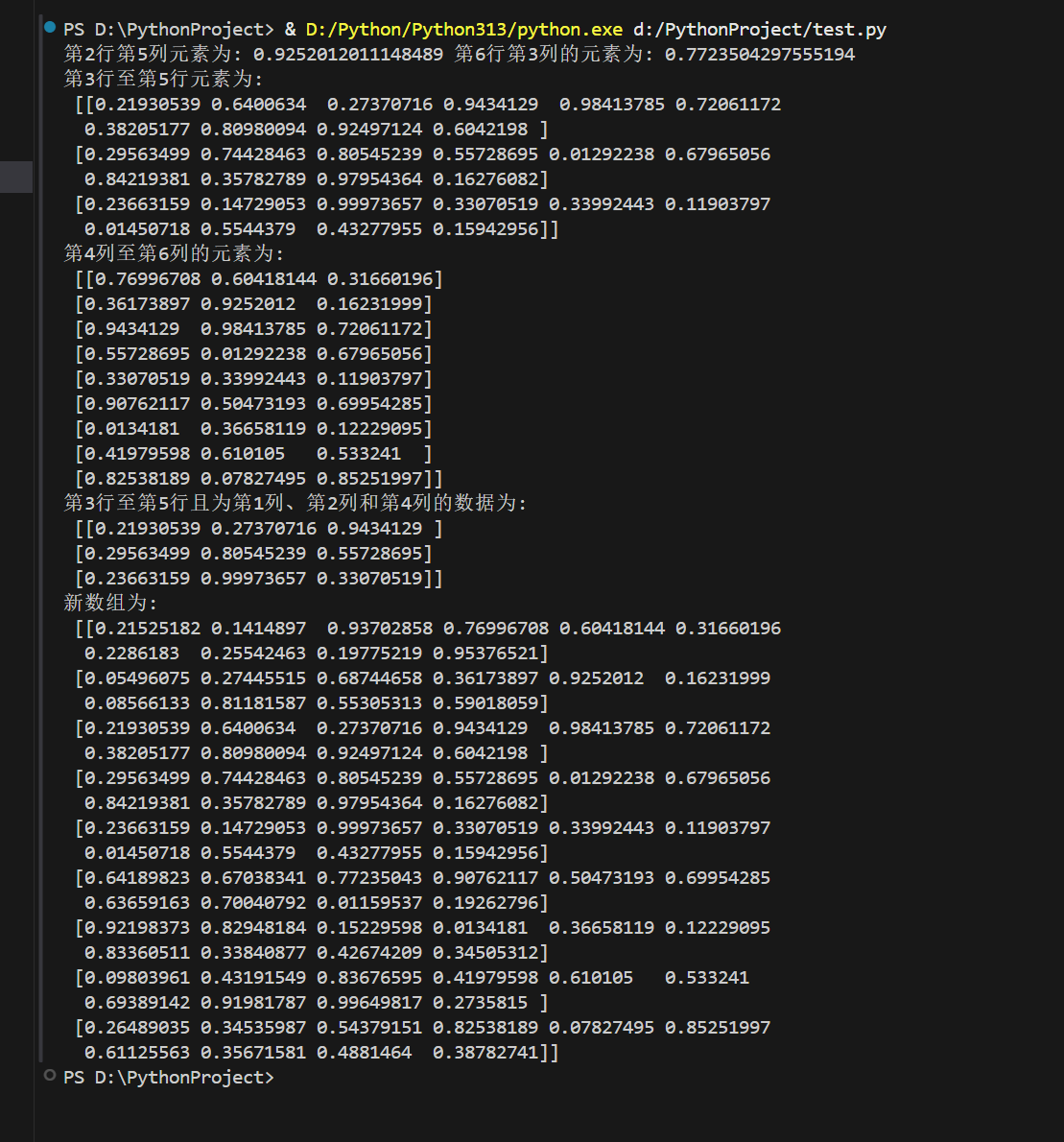
b=a

print("新数组为:\n",b)

源代码截图：



运行结果截图：



(4)利用numpy中相关模块，随机产生20000个服从[0, 1]区间上均匀分布的实数，同时随机产生20000 个服从均值为5，标准差为2的正态分布的实数，由此获得两个一维数组，分别记为A 和B。请计算下列式子的值：

(a) A+B

(b) A和B的乘积

(c) A/B

(d) ln(C + 1)+ ln(D + 1)

(e) A和B的内积

(f)数组B全体元素的中位数、标准差和四分位数（第一四分位数和第三四分位数）

源代码：

import numpy as np

A=np.random.uniform(0,1,20000)

B=np.random.normal(5,2,20000)

print("A+B:",A+B)

print("A\*B:",A\*B)

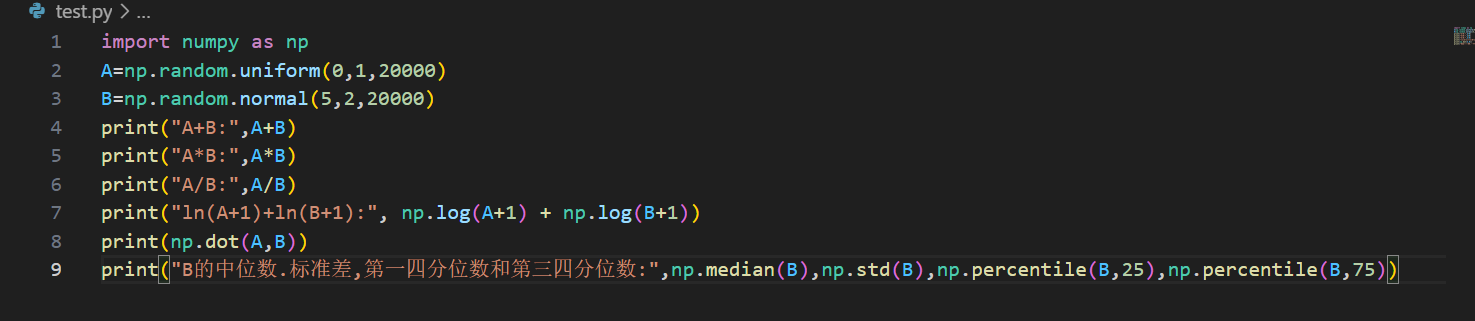
print("A/B:",A/B)

print("ln(A+1)+ln(B+1):", np.log(A+1) + np.log(B+1))

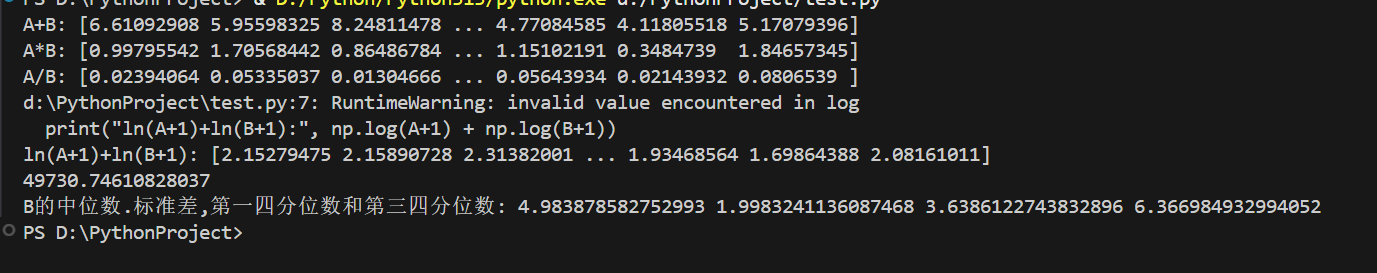
print(np.dot(A,B))

print("B的中位数.标准差,第一四分位数和第三四分位数:",np.median(B),np.std(B),np.percentile(B,25),np.percentile(B,75))

源代码截图：



运行结果截图：



(5)首先用numpy内的函数随机创建一个8\*8的矩阵，将该矩阵正则化，输出正则化后的矩阵，并且输出正则化以后的矩阵内最大的元素。

【正则化定义：假设a是矩阵中的一个元素，max/min分别是矩阵元素的最大最小值，则正则化后a = (a - min)/(max - min)】

源代码：

from functools import reduce

import numpy as np

matrix = np.random.rand(8, 8)

max=np.max(matrix)

min=np.min(matrix)

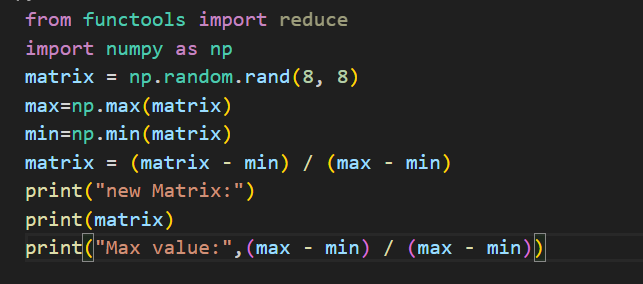
matrix = (matrix - min) / (max - min)

print("new Matrix:")

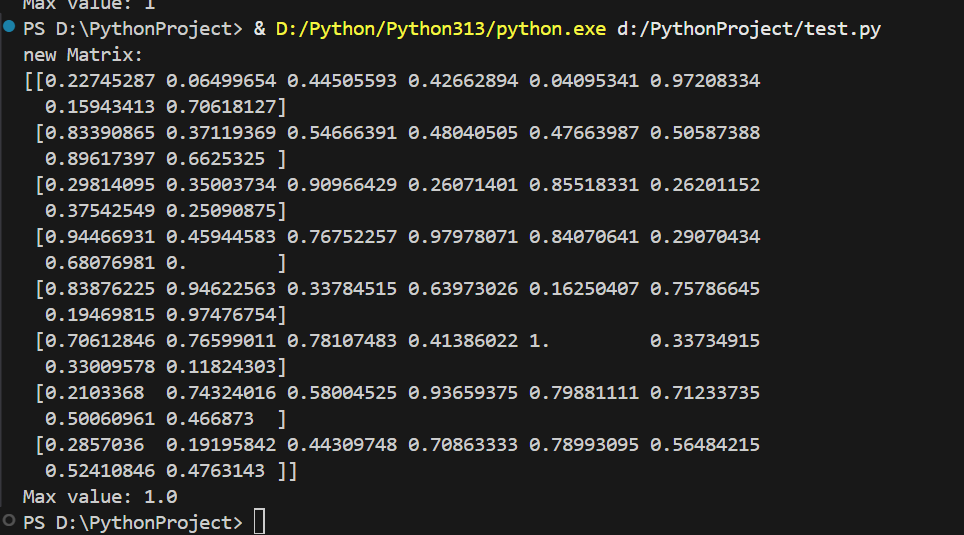
print(matrix)

print("Max value:",(max - min) / (max - min))

源代码截图：



运行结果截图：



(6)利用numpy.random模块，服从区间[10., 20.]上的均匀分布，随机生成一个10行5列的矩阵(二维数组)。通过Python编程依序完成下列任务， 打印输出原数组和产生的新数组：

(a)矩阵的每一行的元素都减去该行的平均值。

(b)利用numpy.savetxt()，将经过(a)操作变换后的新数组数据保存至文本文件”dat.csv”, 保持10行5列不变，并要求小数点后保留3位，两个数据之间用逗号隔开。

(c)利用numpy.loadtxt()从“dat.csv”读取数组数据，原地交换当前数组的第一行和第二行数据。

(d)要求按第2列数据对当前数组进行排序。例如，原数组为

[[1 7 9]

[7 8 1]

[8 4 2]]

若按第2列进行排序，结果如下：

[[8 4 2]

[1 7 9]

[7 8 1]]

源代码：

import numpy as np

a = np.random.uniform(10.0, 20.0, size=(10, 5))

print(a)

for line in a:

line -= line.mean(axis=0)

np.savetxt(r'D:\PythonProject\filename\dat.txt', fmt='%0.3f', delimiter=',', X=a)

b = np.loadtxt(r'D:\PythonProject\filename\dat.txt', delimiter=',')

b[0], b[1] = b[1], b[0]

# 按第2列(索引为1)对数组B进行排序

c = np.argsort(b[:, 1],axis=0, kind='quicksort', order=None)

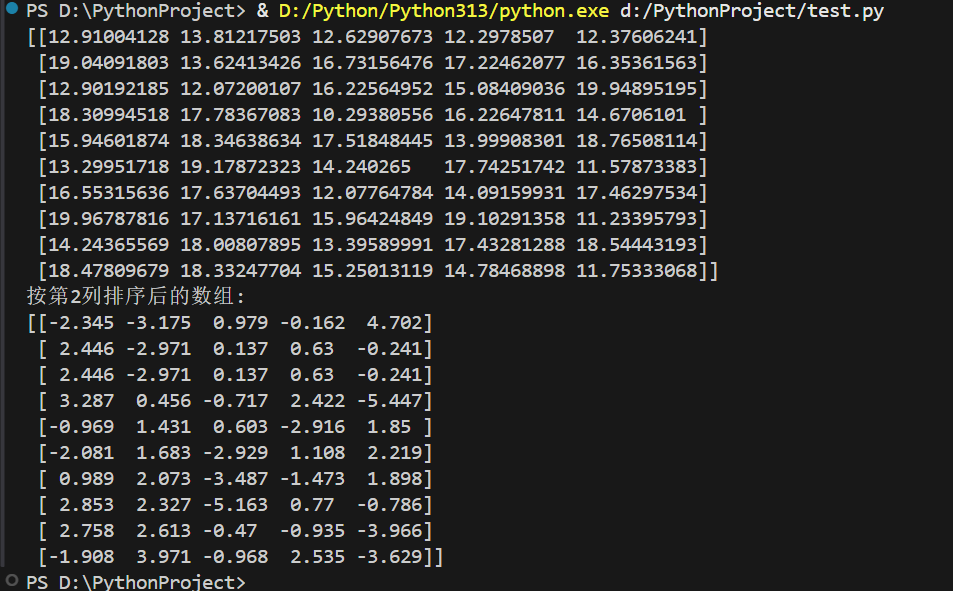
print("按第2列排序后的数组:")

print(b[c])

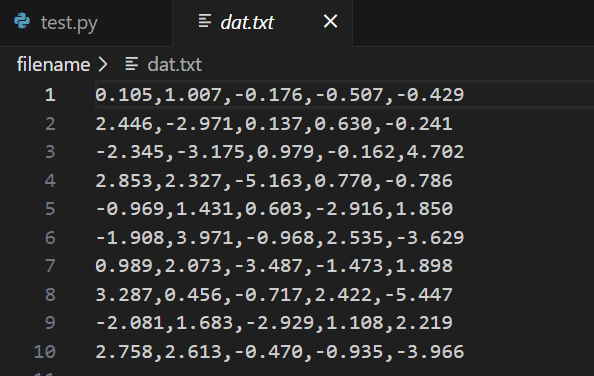
源代码截图：



运行结果截图：



文件(dat.csv)内容截图：



(7)根据data.xls中的内容创建一个DataFrame，计算人均二氧化碳排放量（总排放量/人口）并添加新列CO2 per Capita，输出碳排放总量最高的国家。

源代码：

import pandas as pd

import numpy as np

src = pd.read\_excel(r'D:\PythonProject\filename\data.xls')

df = pd.DataFrame(src)

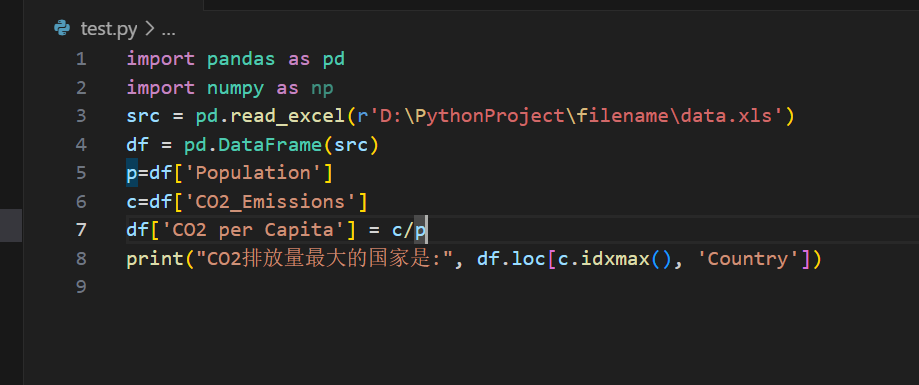
p=df['Population']

c=df['CO2\_Emissions']

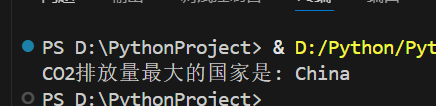
df['CO2 per Capita'] = c/p

print("CO2排放量最大的国家是:", df.loc[c.idxmax(), 'Country'])

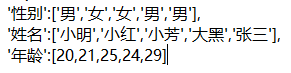
源代码截图：



运行结果截图：



(8)用pandas.DataFrame创建一个数据结构，数据结构内容如下：



请编程完成以下练习：

(a)打印出该数据结构.

(b)用iloc()函数返回该数据结构第零排第零列([0, 0])的元素.

(c)用head()函数输出数据结构前3行，并且打印输出。

(d)用describe() 函数查看数据按列的统计信息，可显示数据的数量、缺失值、最小最大数、平均值、分位数，将其打印出来。

(e)将该数据结构用values()函数转换成ndarray类型（numpy的 N 维数组对象），打印出转换好的数据。

源代码：

import pandas as pd

import numpy as np

df = pd.DataFrame({

'性别': ['男', '女', '女', '男', '男'],

'姓名': ['小明', '小红', '小芳', '大黑', '张三'],

'年龄': [20, 21, 25, 24, 29],

})

print(df)

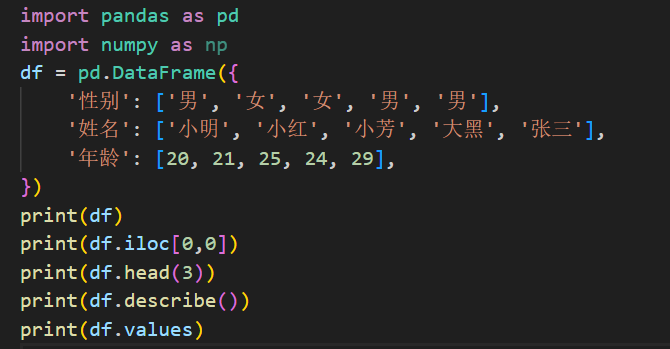
print(df.iloc[0,0])

print(df.head(3))

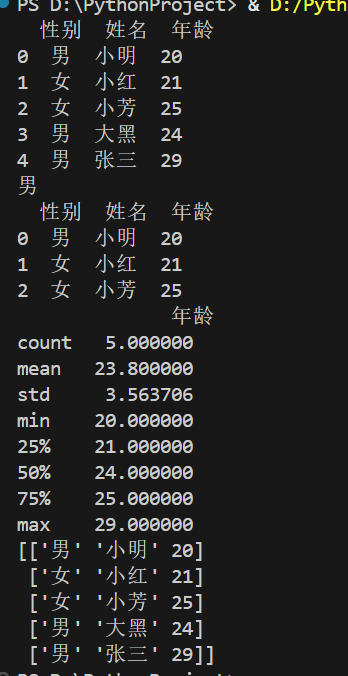
print(df.describe())

print(df.values)

源代码截图：



运行结果截图：



(9)文件夹中的iris.csv数据集为鸢尾花数据集，内包含 3 类鸢尾花，共 150 条记录，每类各 50 个数据，每条记录都有 4 项特征：花萼长度、花萼宽度、花瓣长度、花瓣宽度。程序代码iris\_process.py旨在用seaborn库观察鸢尾花数据集内各个属性之间的关系，通过运行该代码，可以学习到seaborn创建散点图、折线图、柱状图、以及直方图的操作，请认真阅读代码，理解代码的含义（每运行一种类型图，需将其他类型图给注释掉），并要求在原代码的基础上画出花萼长度与花萼宽度之间关系的折线图。

源代码：

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# 设置默认风格

sns.set(style="darkgrid")

# 读入数据

iris = pd.read\_csv(r'D:\PythonProject\filename\iris.csv')

# 设置中文显示字体为黑体

sns.set(font='SimHei')

# 主题

sns.set\_style("whitegrid") # 白色网格背景

sns.set\_style("darkgrid") # 灰色网格背景

sns.set\_style("dark") # 灰色背景

sns.set\_style("white") # 白色背景

sns.set\_style("ticks") # 四周加边框和刻度

# 查看数据大小

print(iris.shape)

# 查看前十行数据

print(iris.head(10))

# 查看数据信息

print(iris.info())

# 散点图

#ax = sns.scatterplot(x='花萼长度', y='花萼宽度', data=iris)

#折线图

#ax = sns.lineplot(x=iris['叶片宽度'], y=iris['叶片长度'])

# 柱状图

#sns.barplot(x=iris['品种'].value\_counts().index, y=iris['品种'].value\_counts().values)

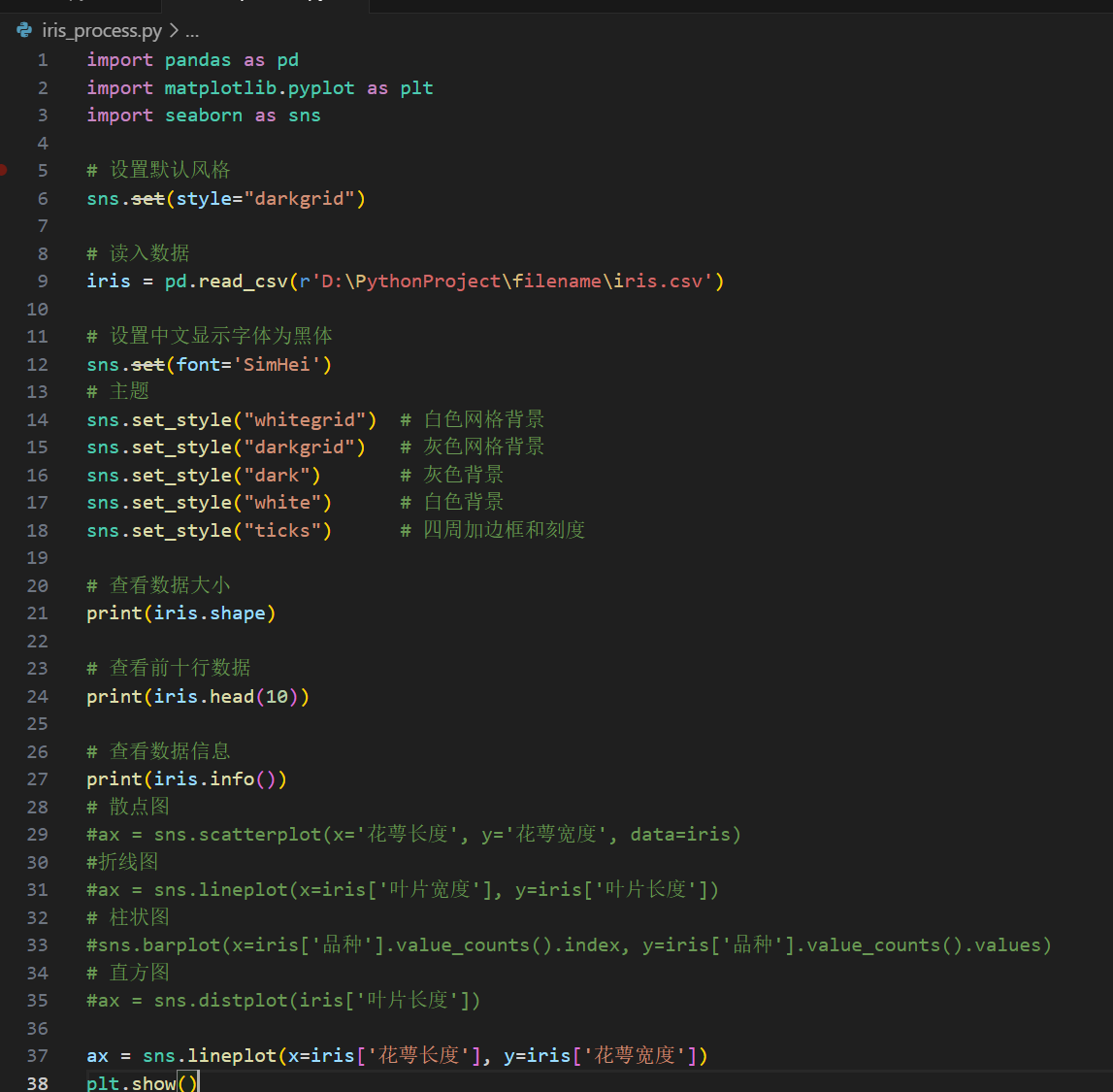
# 直方图

#ax = sns.distplot(iris['叶片长度'])

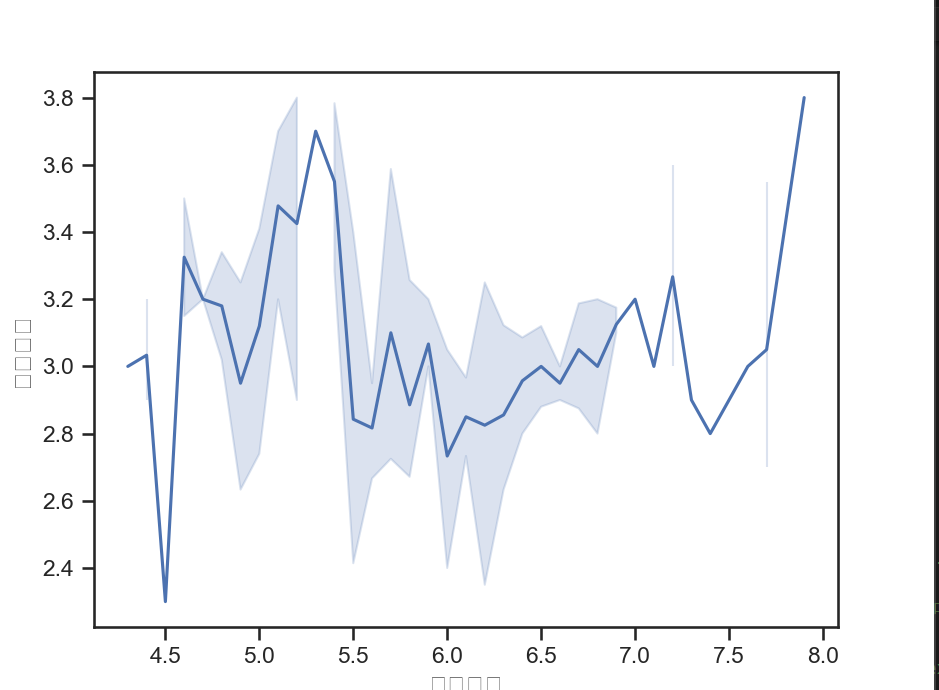
ax = sns.lineplot(x=iris['花萼长度'], y=iris['花萼宽度'])

plt.show()

源代码截图：



运行结果截图：



(10)产生10000个随机数，服从均值为-1，标准差为1的正态分布N(-1,1)，利用matplotlib画出其直方图(100个bin)，即这10000个数的实际分布图。可参考源文件 “histgram\_plot.py”中的代码。  
源代码：

import numpy as np

import matplotlib.pyplot as plt

import matplotlib as mpl

plt.rcParams['font.sans-serif'] = ['SimHei', 'Microsoft YaHei', 'SimSun'] # 优先使用这几种中文字体

plt.rcParams['axes.unicode\_minus'] = False # 解决负号'-'显示为方框的问题

a = np.random.normal(-1, 1, 10000)

plt.figure(figsize=(10, 6))

counts, bins, patches = plt.hist(a, bins=100, alpha=0.7, color='blue', edgecolor='black')

plt.title('N(-1,1)正态分布的10000个随机数', fontsize=14)

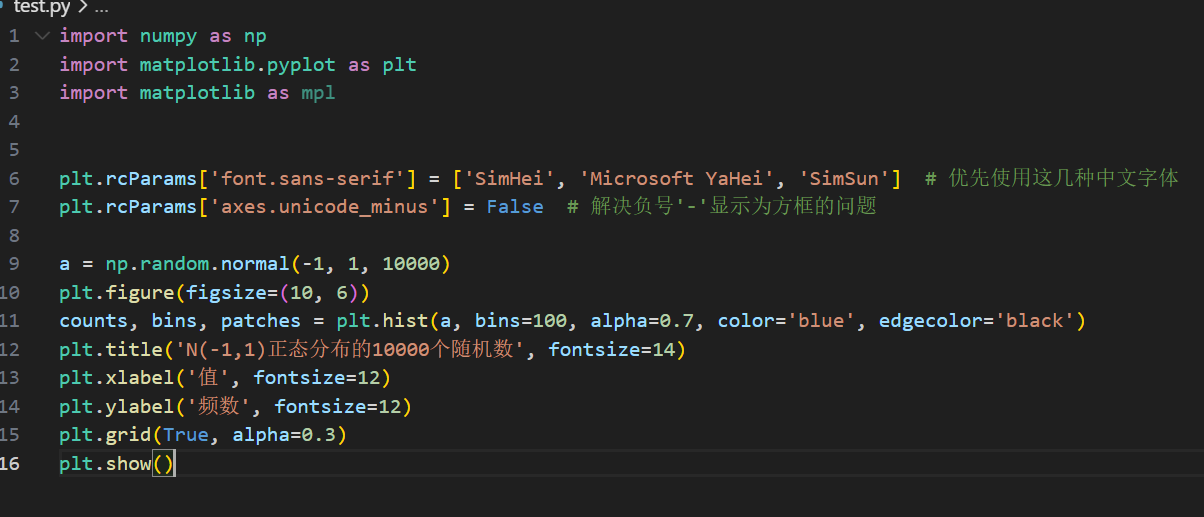
plt.xlabel('值', fontsize=12)

plt.ylabel('频数', fontsize=12)

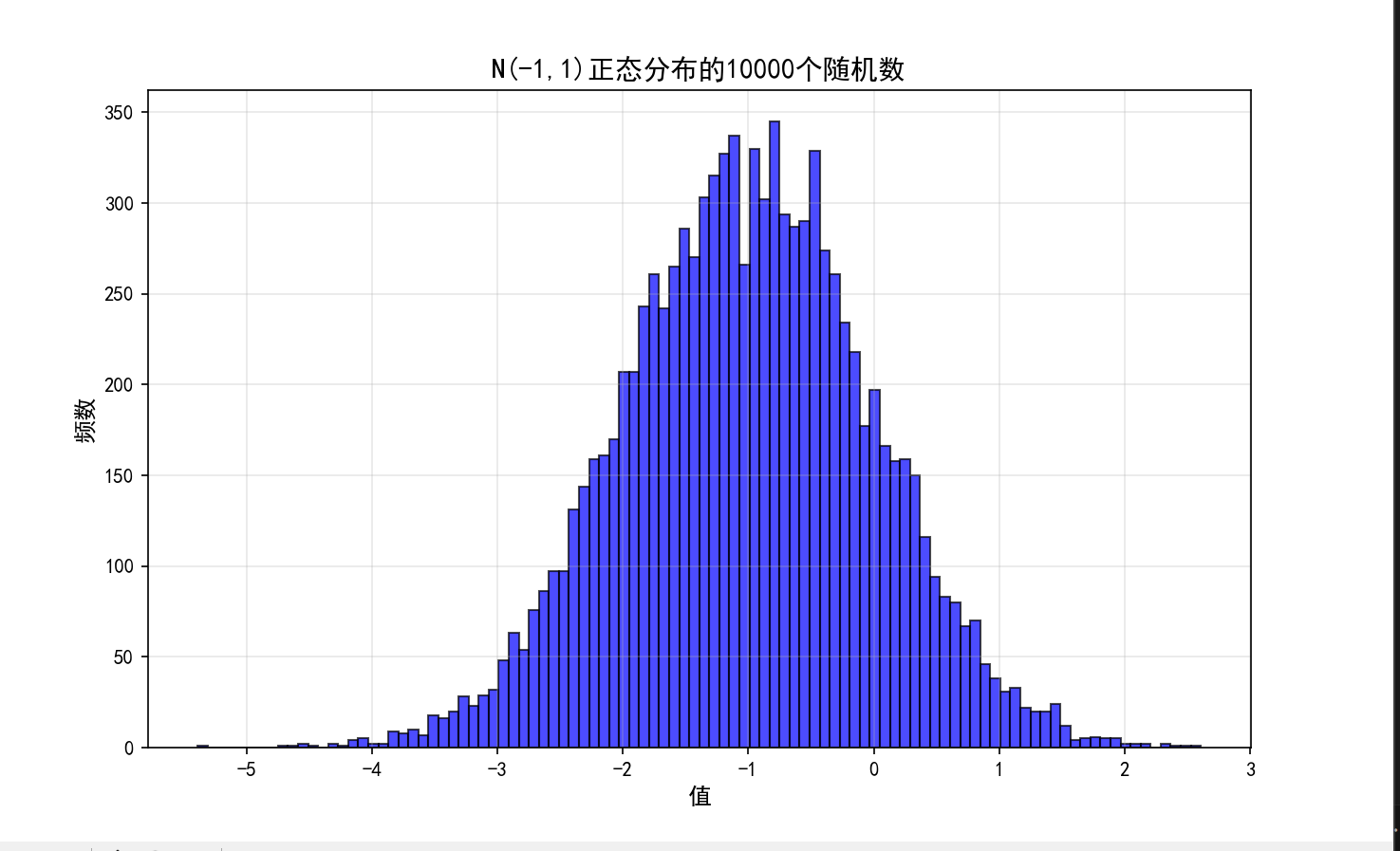
plt.grid(True, alpha=0.3)

plt.show()

源代码截图：



运行结果截图：



(11)根据“user\_data.xls”里的数据创建一个DataFrame，并完成以下题目：

1. 使用Pandas将访问日期解析为“年-月-日”格式，并提取每个订单的访问月份。
2. 使用Pandas计算每个月用户购买金额的总和，并以DataFrame的形式呈现。
3. 使用Matplotlib绘制一个柱状图，横轴为月份，纵轴为购买金额。

源代码：

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# 设置中文字体支持

plt.rcParams['font.sans-serif'] = ['SimHei', 'Microsoft YaHei'] # 设置中文字体

plt.rcParams['axes.unicode\_minus'] = False # 解决负号显示问题

src = pd.read\_excel(r'D:\PythonProject\filename\user\_data.xls')

df=pd.DataFrame(src)

df['访问日期'] = pd.to\_datetime(df['访问日期'])

df['年-月-日'] = df['访问日期'].dt.strftime('%Y-%m-%d')

df['月份'] = df['访问日期'].dt.month

print('用户编号，月份')

print(df['用户编号'],df['月份'])

# 计算每月购买金额总和，返回Series

monthly\_sum = df.groupby(['月份'])['购买金额'].sum()

print(monthly\_sum)

plt.figure(figsize=(10, 6))

# 使用Series的index和values

plt.bar(monthly\_sum.index, monthly\_sum.values, color='skyblue')

plt.title('每月购买金额统计', fontsize=14)

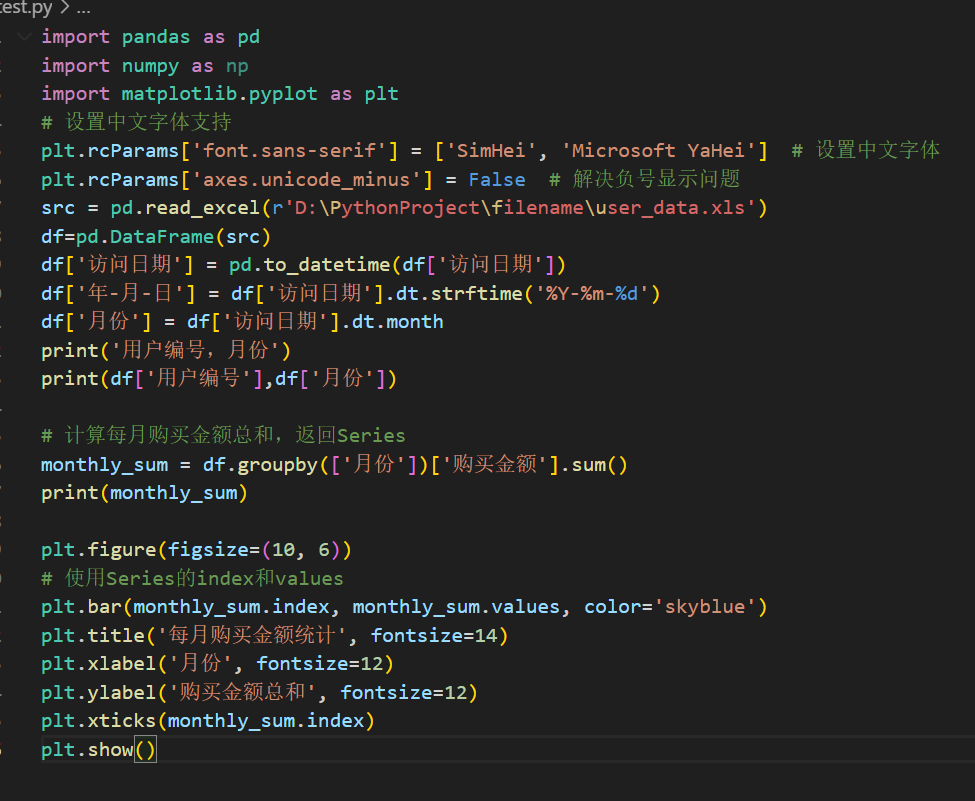
plt.xlabel('月份', fontsize=12)

plt.ylabel('购买金额总和', fontsize=12)

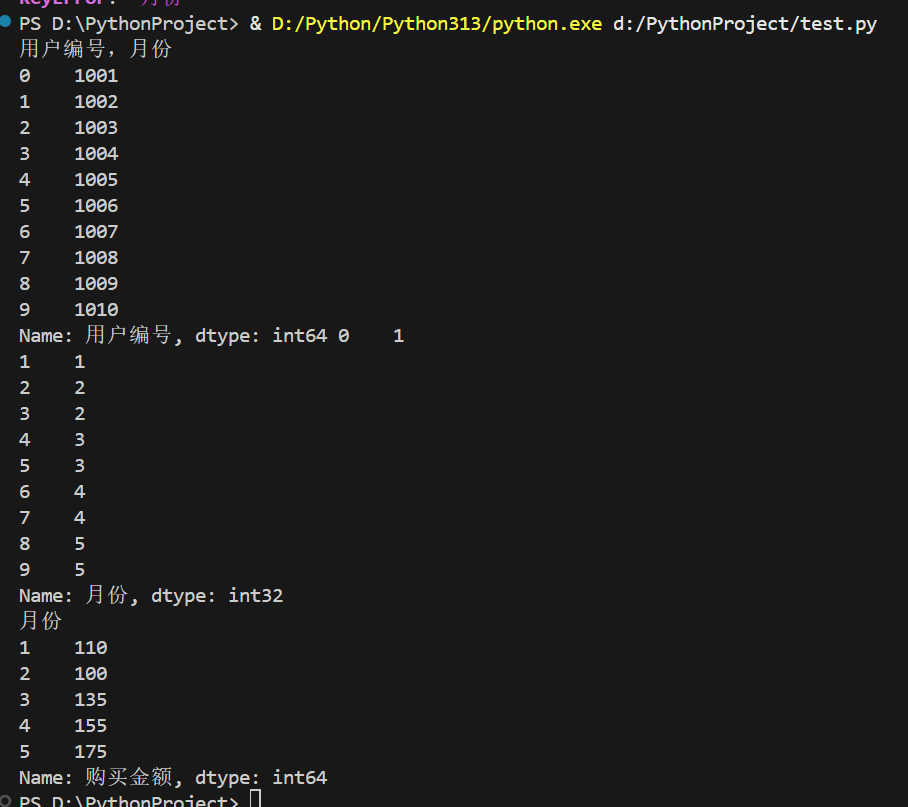
plt.xticks(monthly\_sum.index)

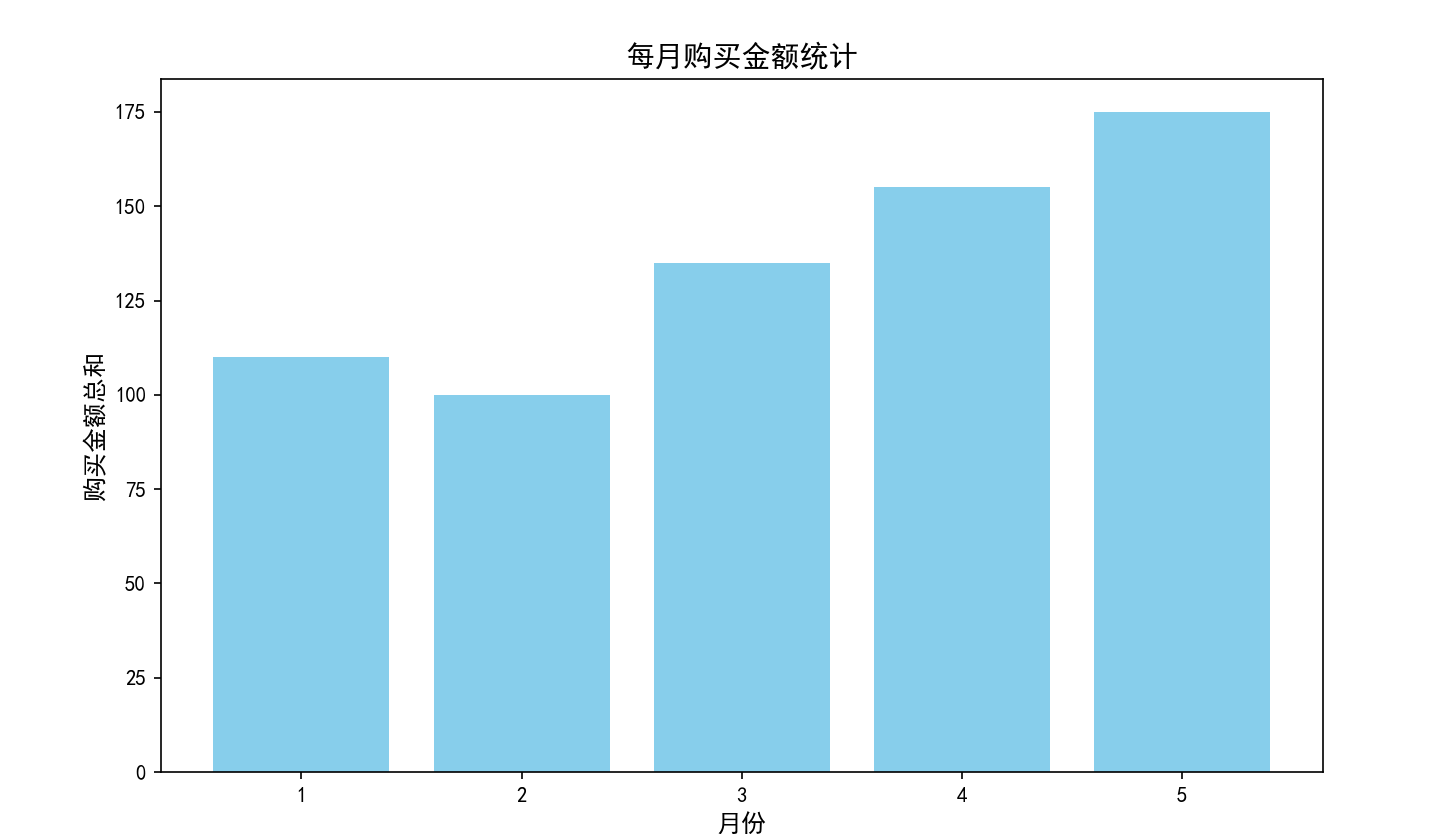
plt.show()

源代码截图：



运行结果截图：





(12)某公司进行了一次为期7天的用户行为跟踪，记录了每天的用户访问数、注册数、下单数与成交额（单位：元），现将数据存储在如下 Python 字典中：

data = {

"日期": ["2025-04-01", "2025-04-02", "2025-04-03", "2025-04-04", "2025-04-05", "2025-04-06", "2025-04-07"],

"访问数": [320, 450, 510, 390, 620, 480, 560],

"注册数": [120, 180, 200, 140, 260, 200, 240],

"下单数": [45, 70, 85, 50, 110, 90, 100],

"成交额": [9000, 14000, 16000, 10000, 21000, 18000, 19500]

}

请完成以下任务：

#### (a) 使用Pandas将以上数据创建为DataFrame并打印输出。

#### (b) 新增一列“转化率”，计算方式为：转化率 = 下单数 / 访问数，保留两位小数。

#### (c) 利用 Matplotlib 或 Seaborn，绘制“日期”与“成交额”的折线图。

#### (d) 使用 NumPy计算“成交额”列的均值、标准差。

源代码：

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

df=pd.DataFrame(data = {

"日期": ["2025-04-01", "2025-04-02", "2025-04-03", "2025-04-04", "2025-04-05", "2025-04-06", "2025-04-07"],

"访问数": [320, 450, 510, 390, 620, 480, 560],

"注册数": [120, 180, 200, 140, 260, 200, 240],

"下单数": [45, 70, 85, 50, 110, 90, 100],

"成交额": [9000, 14000, 16000, 10000, 21000, 18000, 19500]

})

print(df)

df['转化率']=(df['下单数']/df['访问数']).round(2)\*100

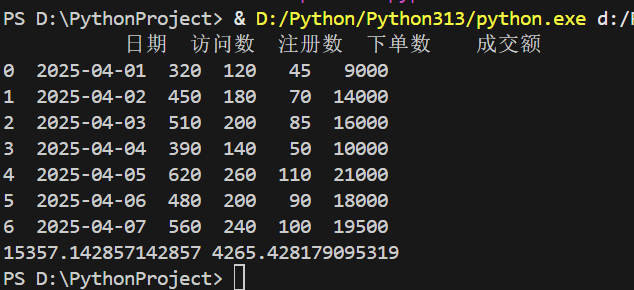
plt.figure(figsize=(10, 6))

plt.plot(df['日期'], df['成交额'], color='blue', marker='o')

plt.show()

print(df['成交额'].to\_numpy().mean(),df['成交额'].to\_numpy().std())

源代码截图：





运行结果截图：

