**实验5 Python科学计算实践**

**姓名：朱俊哲 学号：2212190220 班级：计科2202 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) 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

x **=** np**.**round(np**.**random**.**normal(0**,** 2**,** **size** **=** (9**,** 10))**,** 1)

print(x)

print(f"第2行第5列:\n{x[1**,** 4]}")

print(f"第6行第3列:\n{x[5**,** 2]}")

print(f"第3行至第5行:\n{x[2**:**5**,** **:**]}")

print(f"第4列至第6列:\n{x[**:,** 3**:**6]}")

print(f"第3行至第5行且为第1列、第2列和第4列:\n{x[2**:**5**,** [0**,** 1**,** 3]]}")

print("重新排列后的数组:")

condition **=** x **>** 1

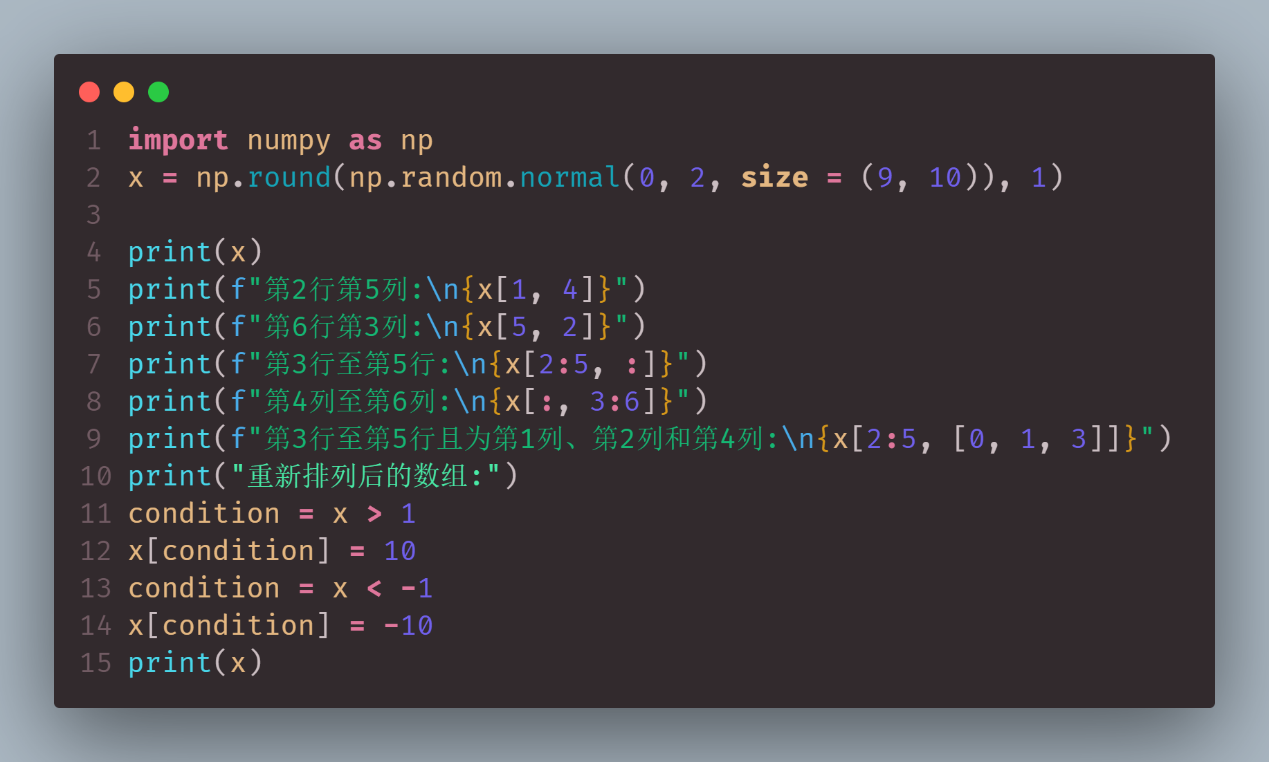
x[condition] **=** 10

condition **=** x **<** **-**1

x[condition] **=** **-**10

print(x)

源代码截图：



运行结果截图：



(2)利用numpy中相关模块，随机产生10000个服从 [-1,1]区间上均匀分布的实数，同时随机产生10000个服从标准正态分布的实数，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) exp(A)+exp(B)

(e) A和B的内积

(f)数组B全体元素的平均值，最大值，最小值

源代码：

**import** numpy **as** np

A **=** np**.**round(np**.**random**.**uniform(**-**1**,** 1**,** **size** **=** 10000)**,** 2)

B **=** np**.**round(np**.**random**.**uniform(**-**1**,** 1**,** **size** **=** 10000)**,** 2)

print(f"A = {A}")

print(f"B = {B}")

print(f"A + B = {A **+** B}")

print(f"A \* B = {A **\*** B}")

**if** 0 **in** B**:**

    print("B中有0, 无法计算A / B")

**else:**

    print(f"A / B = {A **/** B}")

print(f"exp(A) + exp(B) = {np**.**exp(A) **+** np**.**exp(B)}")

print(f"A · B = {np**.**dot(A**,** B)}")

print(f"B的平均值 = {np**.**mean(B)}")

print(f"B的最大值 = {np**.**max(B)}")

print(f"B的最小值 = {np**.**min(B)}")

源代码截图：



运行结果截图：



(3) 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

x **=** np**.**round(np**.**random**.**uniform(10**,** 20**,** **size** **=** [10**,** 5])**,** 2)

print(f"原始矩阵为:\n{x}")

y **=** x**.**copy()

**for** i **in** range(0**,** 10)**:**

    avg **=** np**.**sum(y[i]) **/** 5

**for** j **in** range(0**,** 5)**:**

        y[i][j] **=** y[i][j] **-** avg

np**.**savetxt("dat.csv"**,** y**,** **fmt** **=** "%.3f"**,** **delimiter** **=** ",")

index **=** np**.**argsort(x[**:,** 1])

sort\_x **=** x[index]

print(f"排序后的矩阵为:\n{sort\_x}")

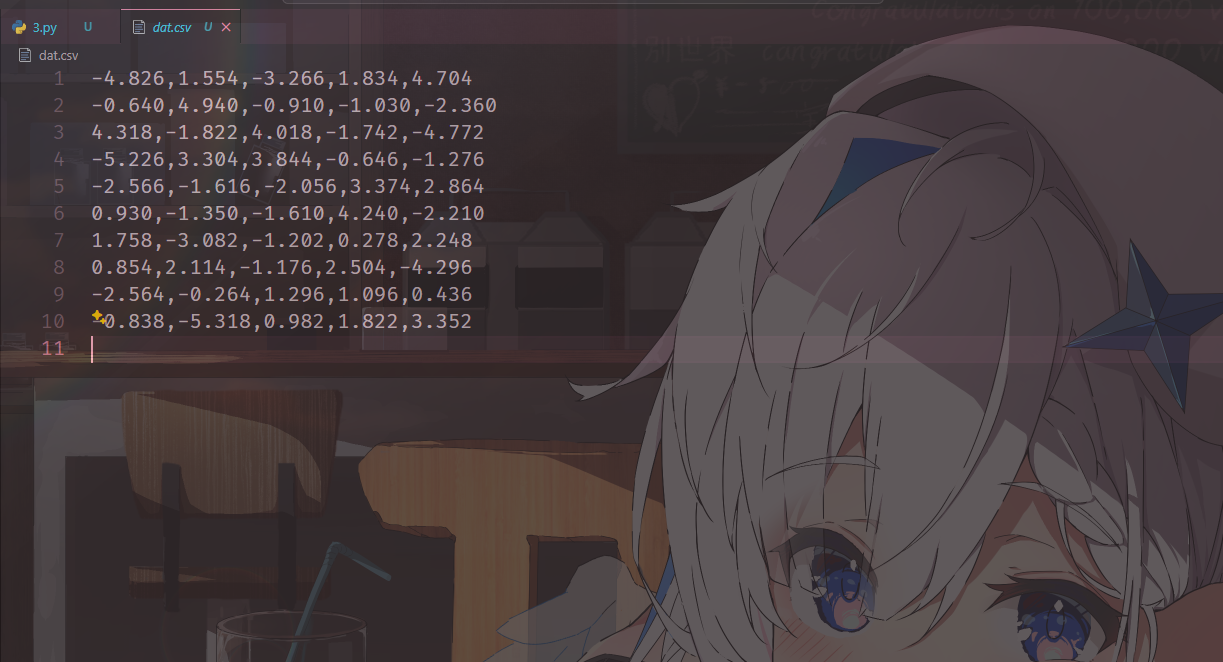
源代码截图：



运行结果截图：

****

文件(dat.csv)内容截图：

****

(4)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：

**import** numpy **as** np

a **=** np**.**random**.**rand(8**,** 8)

maxx **=** a**.**max()

minn **=** a**.**min()

**for** i **in** range(8)**:**

**for** j **in** range(8)**:**

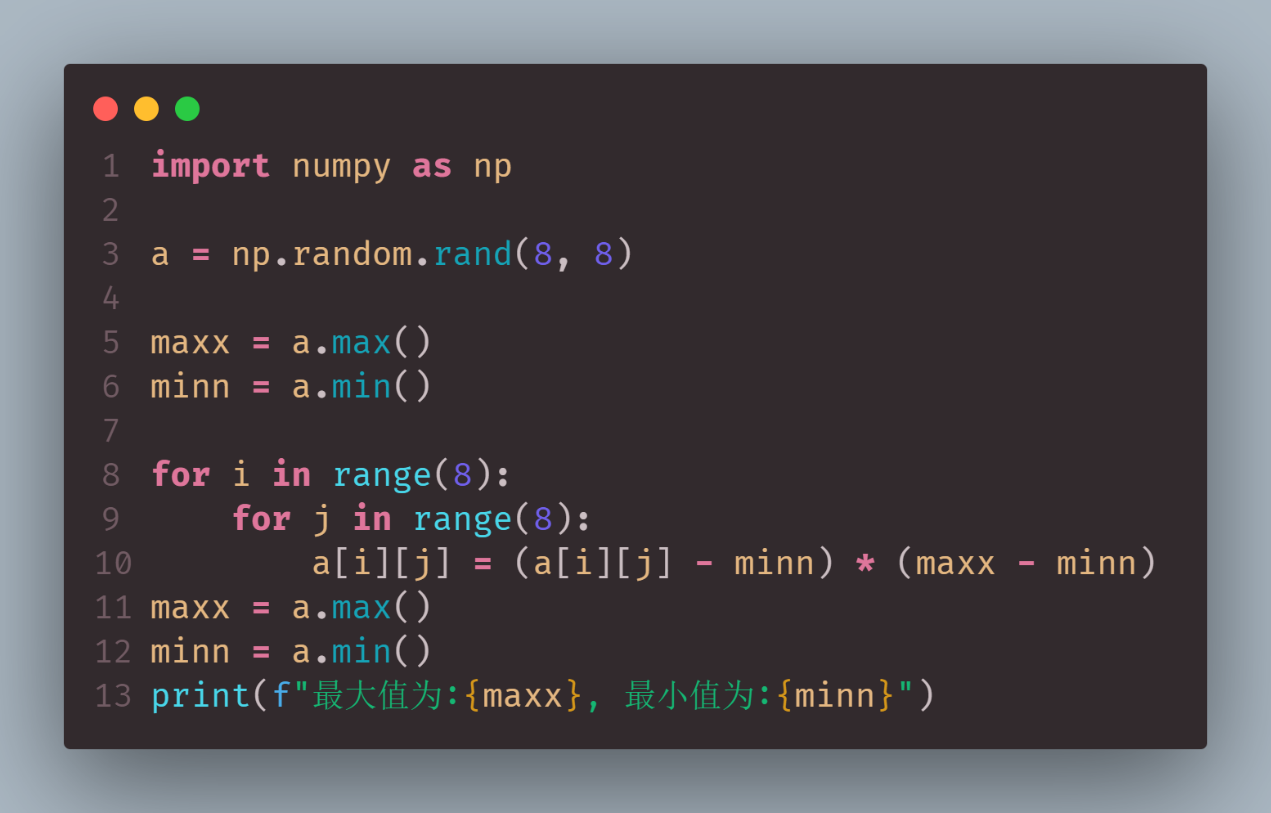
        a[i][j] **=** (a[i][j] **-** minn) **\*** (maxx **-** minn)

maxx **=** a**.**max()

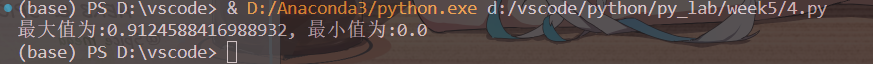
minn **=** a**.**min()

print(f"最大值为:{maxx}, 最小值为:{minn}")

源代码截图：



运行结果截图：



1. 使用Scipy模块找到以下函数的最小值：。

源代码：

**import** numpy **as** np

**from** scipy**.**optimize **import** fmin

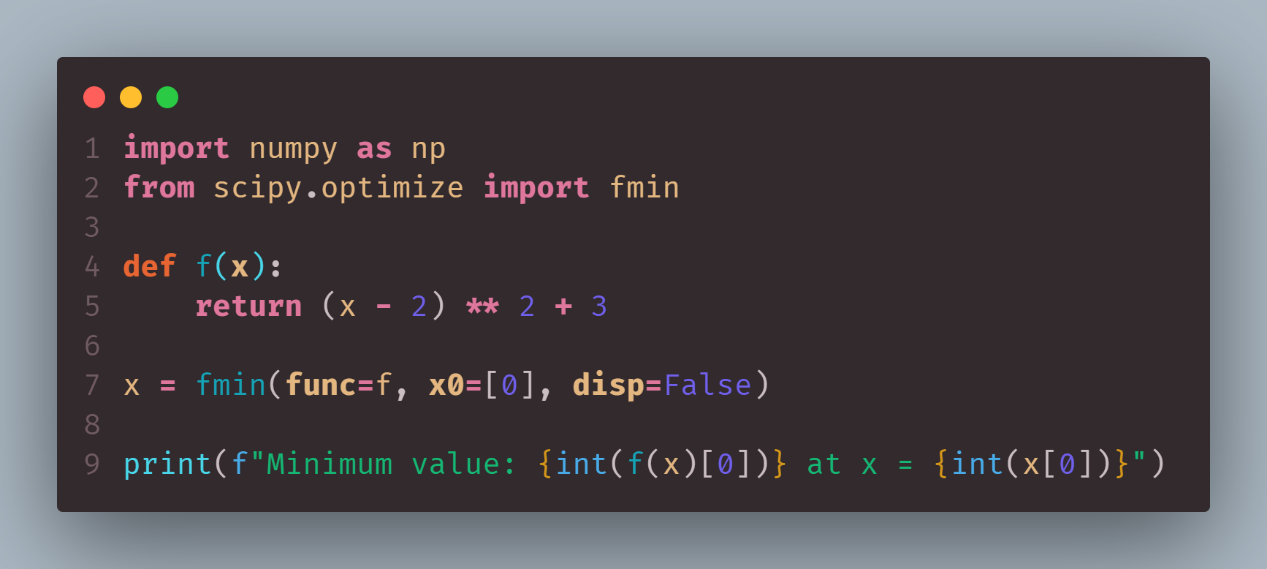
**def** f(**x**)**:**

**return** (x **-** 2) **\*\*** 2 **+** 3

x **=** fmin(**func=**f**,** **x0=**[0]**,** **disp=**False)

print(f"Minimum value: {int(f(x)[0])} at x = {int(x[0])}")

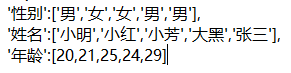
源代码截图：



运行结果截图：



1. 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

data **=** {'Sex'**:**['男'**,** '女'**,** '女'**,** '男'**,** '男']**,**

        'Name'**:**['小明'**,** '小红'**,** '小芳'**,** '大黑'**,** '张三']**,**

        'Age'**:**[20**,** 21**,** 25**,** 24**,** 29]}

df **=** pd**.**DataFrame(data)

print(df**,**"\n")

x **=** df**.**iloc[0**,** 0]

print("数据结构第零排第零列([0, 0])的元素: "**,** x**,** "\n")

line **=** df**.**head(3)

print("数据结构前3行\n"**,** line**,** "\n")

print("数据按列的统计信息\n"**,** df**.**describe()**,** "\n")

print("转换成ndarray类型的数据\n"**,** df**.**values)

源代码截图：



运行结果截图：



1. 根据data.xls里的内容创建一个DataFrame，计算人口密度（人口/面积）并添加新列Popu lation Density，输出人口最多的国家。

源代码：

**import** pandas **as** pd

file\_path **=** 'D:/vscode/python/py\_lab/week5/docs/data.xls'

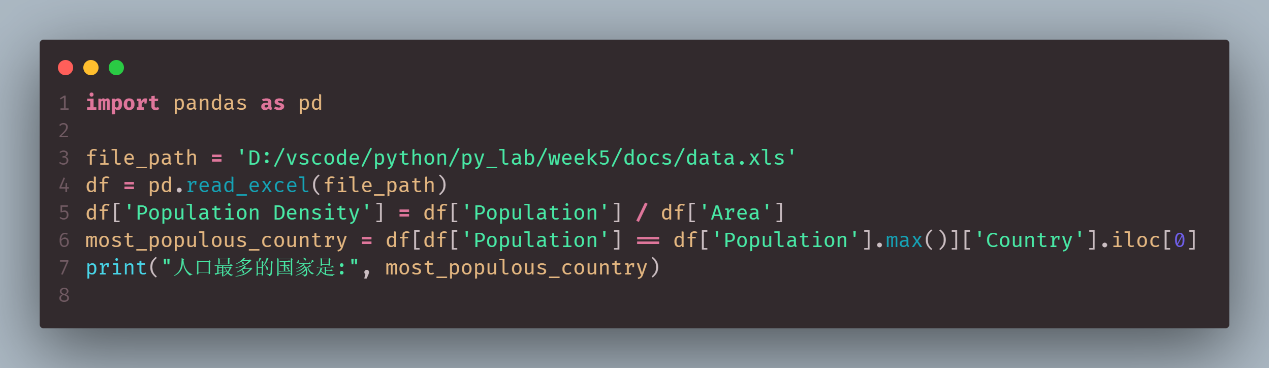
df **=** pd**.**read\_excel(file\_path)

df['Population Density'] **=** df['Population'] **/** df['Area']

most\_populous\_country **=** df[df['Population'] **==** df['Population']**.**max()]['Country']**.**iloc[0]

print("人口最多的国家是:"**,** most\_populous\_country)

源代码截图：

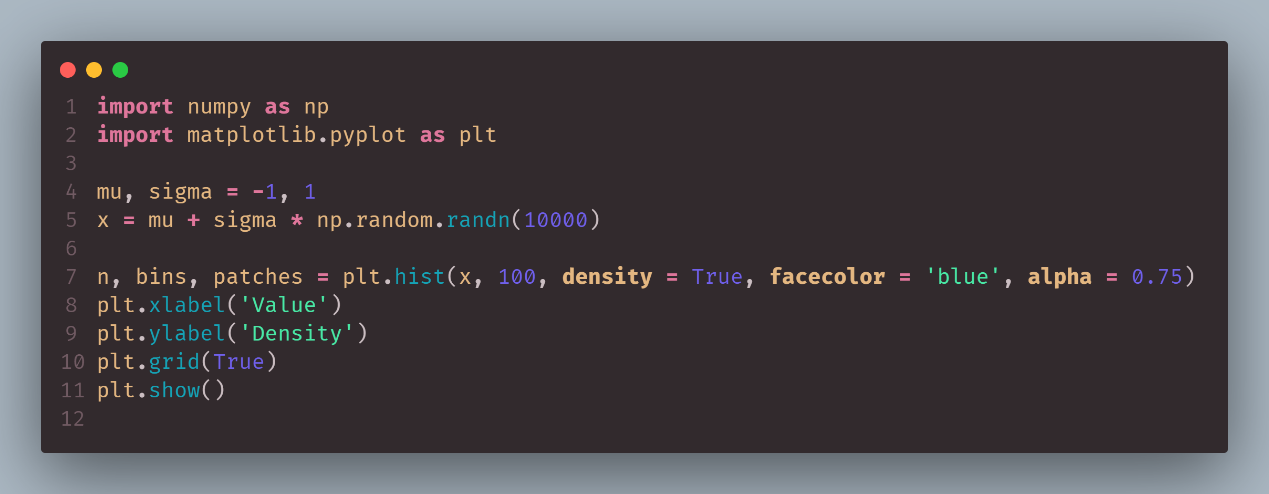


运行结果截图：

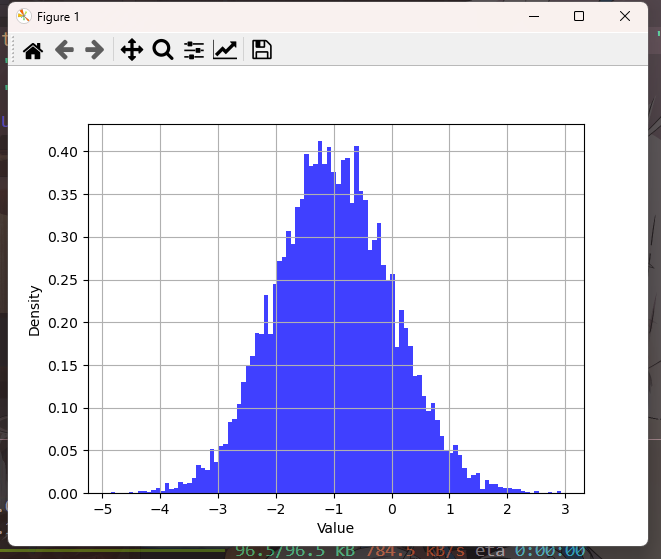


1. 产生10000个随机数，服从均值为-1，标准差为1的正态分布N(-1,1)，利用matplotlib画出其直方图(100个bin)，即这10000个数的实际分布图。可参考源文件 “histgram\_plot.py”中的代码。  
   源代码：
2. **import** numpy **as** np
3. **import** matplotlib**.**pyplot **as** plt
4. mu**,** sigma **=** **-**1**,** 1
5. x **=** mu **+** sigma **\*** np**.**random**.**randn(10000)
6. n**,** bins**,** patches **=** plt**.**hist(x**,** 100**,** **density** **=** True**,** **facecolor** **=** 'blue'**,** **alpha** **=** 0.75)
7. plt**.**xlabel('Value')
8. plt**.**ylabel('Density')
9. plt**.**grid(True)
10. plt**.**show()

源代码截图：



运行结果截图：



1. 在同一张图中绘制函数。

源代码：

**import** numpy **as** np

**import** matplotlib**.**pyplot **as** plt

x1 **=** np**.**linspace(**-**2 **\*** np**.**pi**,** 2 **\*** np**.**pi**,** 400)

x2 **=** np**.**linspace(0.1**,** 2 **\*** np**.**pi**,** 400)

y\_x2 **=** x1 **\*\*** 2

y\_log **=** np**.**log(x2)

y\_sin **=** np**.**sin(x1)

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

plt**.**plot(x1**,** y\_x2**,** **label** **=** '$y=x^2$'**,** **color** **=** 'red')

plt**.**plot(x2**,** y\_log**,** **label** **=** '$y=\ln(x)$'**,** **color** **=** 'blue')

plt**.**plot(x1**,** y\_sin**,** **label** **=** '$y=\sin(x)$'**,** **color** **=** 'green')

*# Adding labels and title*

plt**.**xlabel('x')

plt**.**ylabel('y')

plt**.**title('Plot of $y=x^2$, $y=\ln(x)$, and $y=\sin(x)$')

plt**.**axhline(0**,** **color** **=** 'black'**,** **linewidth** **=** 0.5)

plt**.**axvline(0**,** **color** **=** 'black'**,** **linewidth** **=** 0.5)

plt**.**grid(True)

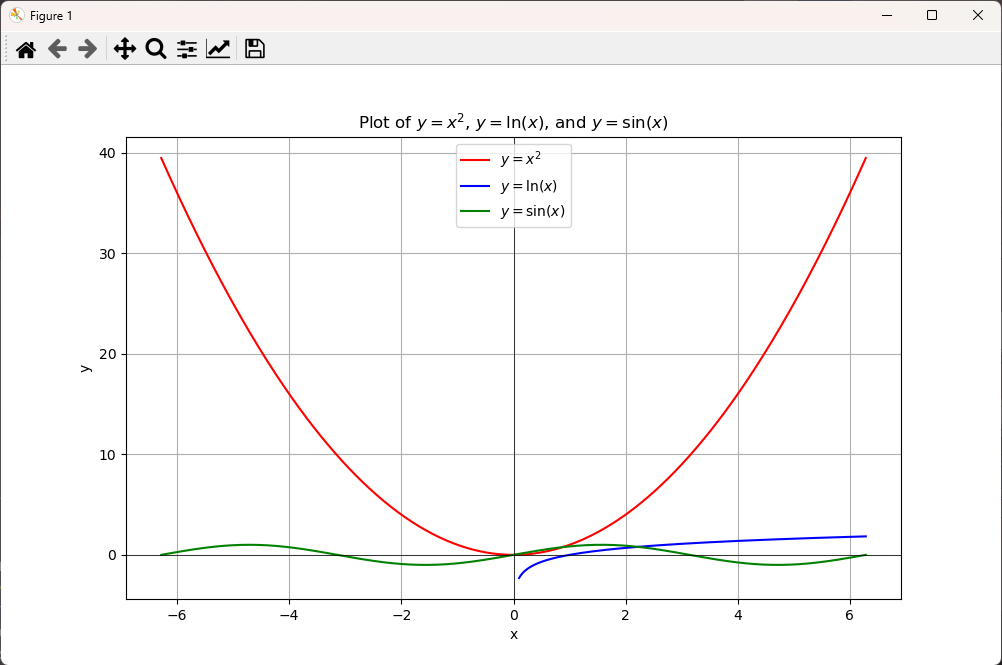
plt**.**legend()

plt**.**show()

源代码截图：



运行结果截图：



1. 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** seaborn **as** sns

**import** matplotlib**.**pyplot **as** plt

iris\_data\_path **=** 'D:/vscode/python/py\_lab/week5/docs/iris.csv'

iris **=** pd**.**read\_csv(iris\_data\_path)

sns**.**set(**style** **=** "darkgrid")

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

ax **=** sns**.**lineplot(**x=**'花萼长度'**,** **y=**'花萼宽度'**,** **data=**iris**,** **marker=**'o')

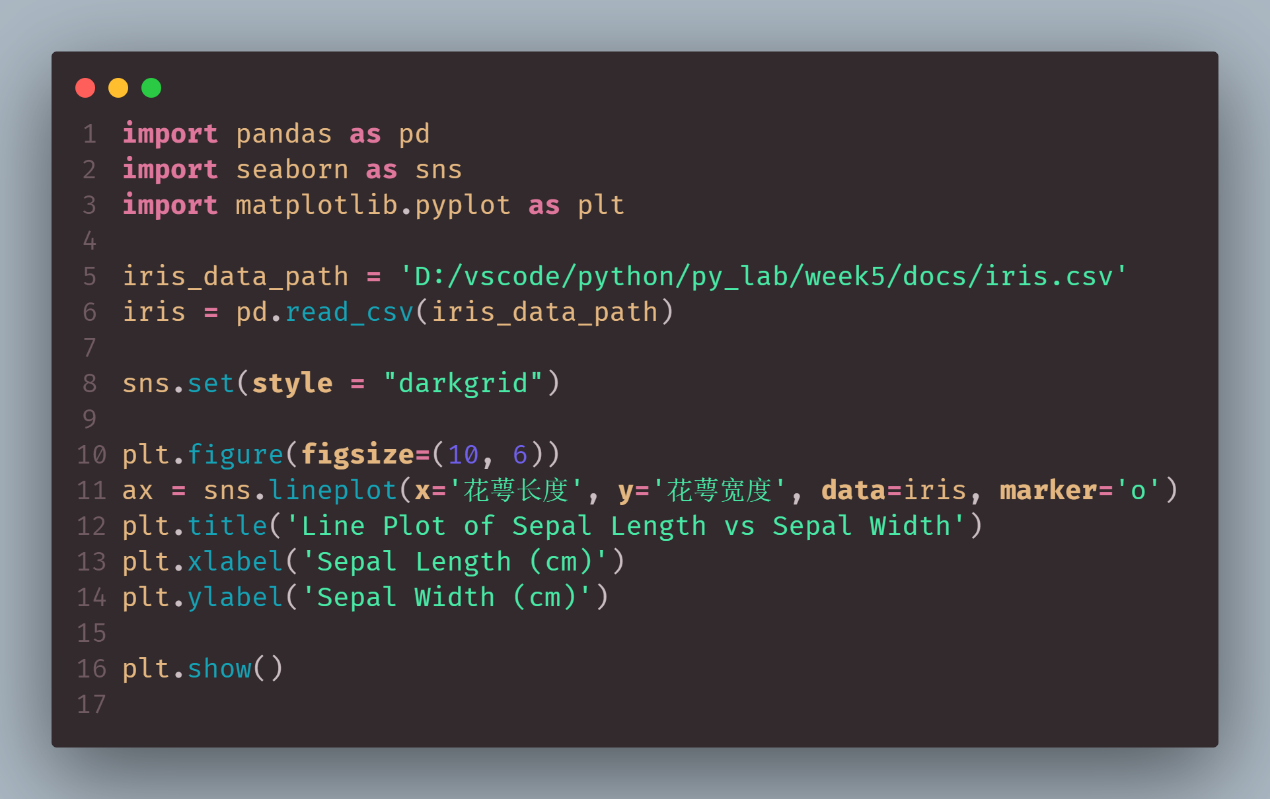
plt**.**title('Line Plot of Sepal Length vs Sepal Width')

plt**.**xlabel('Sepal Length (cm)')

plt**.**ylabel('Sepal Width (cm)')

plt**.**show()

源代码截图：



运行结果截图：



1. 根据“user\_data.xls”里的数据创建一个DataFrame，并完成以下题目：
2. 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。
3. 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。
4. 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：

源代码截图：

运行结果截图：