《人工智能与模式识别》

实 验 报 告 书

班级： 计科2002

学号： 202801108

姓名： 李蔚

指导教师： 翟婷婷

2022-2023 学年 第 二 学期

**实验名称： 熟悉MATLAB**

**实验时间： 2023 年 03 月 10 日 第 4 周 星期 五**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 题目 | 二 | 三 | | | | 总分 |
| 1 | 2 | 3 | 4 |
| 得分 |  |  |  |  |  |  |

一、实验目的

1．熟悉Matlab编程环境：命令窗口、M文件编辑调试器、工作空间、搜索路径等。

2. 掌握Matlab中的矩阵及其运算：向量/矩阵的创建、访问、拼接、拆分以及基本运算。

3. 掌握Matlab程序设计基础：程序控制结构、M文件的编写、M文件的调试等。

二、实验预习（预备知识的问题及回答）

1、 简述Matlab脚本文件与函数文件的主要区别。(1分)

**解答：**

输入和输出：脚本文件没有输入和输出参数，而函数文件则可以接收输入参数，并返回输出值。

代码复用：函数文件被设计用于代码复用，可以在其他脚本文件或函数文件中被调用，而脚本文件则是独立的代码块。

变量作用域：脚本文件中定义的变量在整个程序中都可以访问，而函数文件中定义的变量只在函数内部有效，函数的输出也只能在函数外部访问。

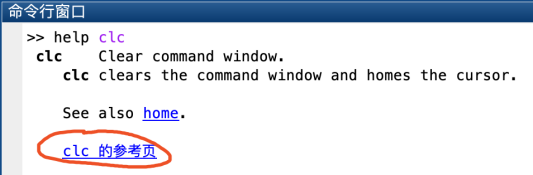
程序执行：脚本文件在Matlab命令窗口中执行时，会直接运行其中的代码，而函数文件必须通过调用函数名来执行。

文件命名：函数文件通常以.m文件扩展名保存，并使用与函数名相同的文件名，而脚本文件的文件名可以随意设置，但也通常以.m文件扩展名保存。

总的来说，函数文件是一种更加灵活和有用的程序形式，它们可以接收输入参数和返回输出值，能够在不同的程序中被重复使用。而脚本文件则是一种更加简单和直接的程序形式，它们适用于编写一些简单的程序和快速测试程序的功能。

三、实验内容（包含实验所用命令或相关程序源代码）

1. Matlab有非常棒的帮助文档，对于任何函数或命令，想要了解其功能是什么，其输入和输出是什么，只需要在命令行窗口中输入：help 命令名或函数名。例如，想知道clc的作用，输入：help clc，就会在命令行窗口中显示clc的简介，如下图所示：



当点击最下方的**clc参考页，**还能打开Matlab详细的帮助文档，里面提供了大量函数使用的例子，以供学习。现在要求用help 查询以下几个函数的功能和使用：size, mean, std, eig, repmat, randn。(3个1分，共2分)

**解答：**

size的功能是：size - 数组大小

此 MATLAB 函数 返回一个行向量，其元素是 A 的相应维度的长度。例如，如果 A 是一个3×4矩阵，则 size(A) 返回向量 [3 4]。

mean的功能是: mean - 数组的均值

此 MATLAB 函数 返回 A 沿大小不等于 1 的第一个数组维度的元素的均值。

std的功能是：std - 标准差

此 MATLAB 函数 返回 A 沿大小不等于 1 的第一个数组维度的元素的标准差。默认情况下，标准差按 N-1 实现归一化，其中 N 是观测值数量。

eig的功能是：eig - 特征值和特征向量

此 MATLAB 函数 返回一个列向量，其中包含方阵 A 的特征值。

repmat的功能是：repmat - 重复数组副本

此 MATLAB 函数 返回一个数组，该数组在其行维度和列维度包含 A 的 n 个副本。A 为矩阵时，B 大小为 size(A)\*n。

randn的功能是：randn - 正态分布的随机数

此 MATLAB 函数 返回一个从标准正态分布中得到的随机标量。

2. 在Matlab的命令行窗口中，按照顺序完成如下的练习：(每个0.5分，共4分)

(1) 创建一个2\*3矩阵a，其**第一列**元素为1, 2，第二列为3, 4，第三列为5, 6

1. 将矩阵a与a的**转置**矩阵相乘，得到的结果赋值给b
2. 计算矩阵b 的特征值和特征向量，将特征向量构成的矩阵表示为v (提示：用eig函数)

(4) 计算矩阵b与矩阵v的对应位置的元素之间的除法 (提示：用 ./ 运算)

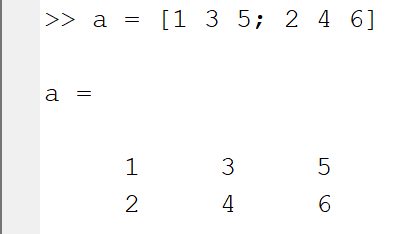
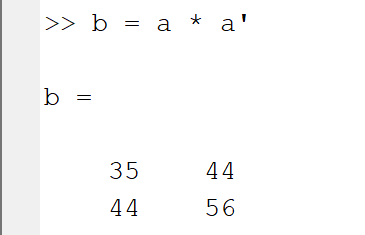
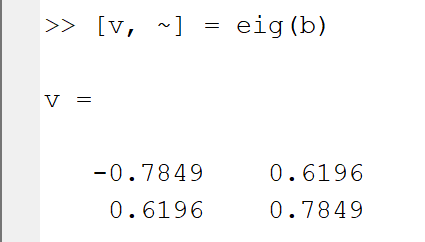
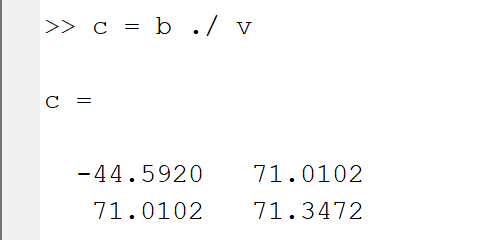
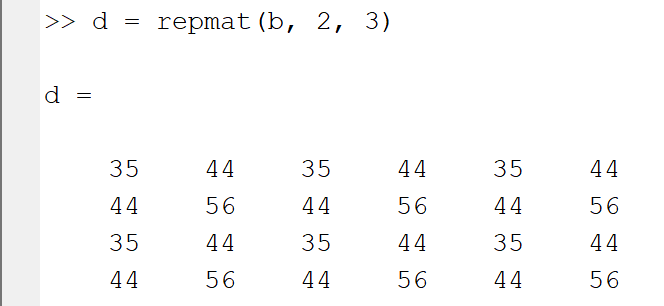
(5) 创建一个2\*3的分块矩阵d，每一个块的内容均为矩阵b (提示：用repmat函数)

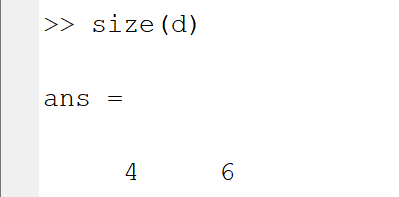
(6) 求矩阵d的维度，包括行数和列数 (提示：size函数)

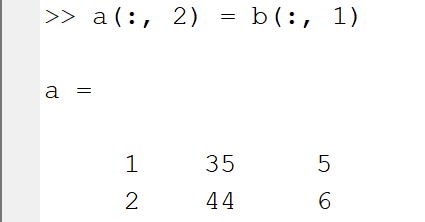
(7) 将矩阵a的第二列替换为矩阵b的第一列

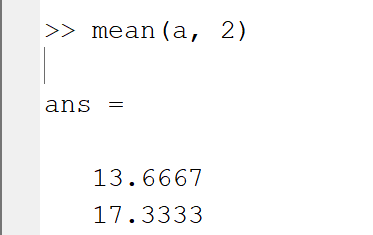
(8) 计算矩阵a每一**行**的元素的均值 (提示：用mean函数)

**代码及运行结果的截图：**

1. a = [1 3 5; 2 4 6]
2. b = a \* a'
3. [v, ~] = eig(b)
4. c = b ./ v
5. d = repmat(b, 2, 3)

(6) size(d)

(7) a(:, 2) = b(:, 1)

(8) mean(a, 2)

3. 对一个矩阵实现zscore标准化，即是将矩阵中的每一列元素的值变换为呈标准正太分布的值。假设在标准化前一个元素的值为x，则经过zscore标准化后，该元素的值为 ，其中为元素所在列的均值，是元素所在列的标准差。编写一个脚本文件，实现对下述矩阵A的zscore标准化，其中A = randn(20, 20) \* 2 +1。

(1)要求：使用for循环，依次对矩阵每列的元素进行处理。(3分)

**代码：**

A = randn(20, 20) \* 2 + 1;

mu = mean(A); % 每列均值

sigma = std(A); % 每列标准差

[m,n] = size(A);

A\_zscore = zeros(m,n);

for i = 1:n

A\_zscore(:,i) = (A(:,i) - mu(i)) / sigma(i);

end

(2)要求：只用一个表达式来实现上述功能 (提示需要用repmat函数和矩阵的”./”运算) (3分)

**代码：**

A = randn(20, 20) \* 2 + 1;  
A = (A - repmat(mean(A), size(A, 1), 1)) ./ repmat(std(A), size(A, 1), 1);

4. 编写一个函数文件，命名为myfunc.m，输入为任意矩阵，输出为经过zscore标准化后的矩阵。在命令行窗口中调用该函数，其中输入设置为A = randn(20, 20) \* 2 +2。(2分)

**代码：**

function output = myfunc(input)

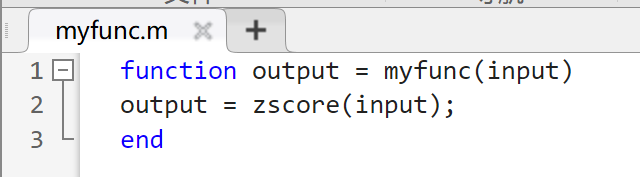
output = zscore(input);

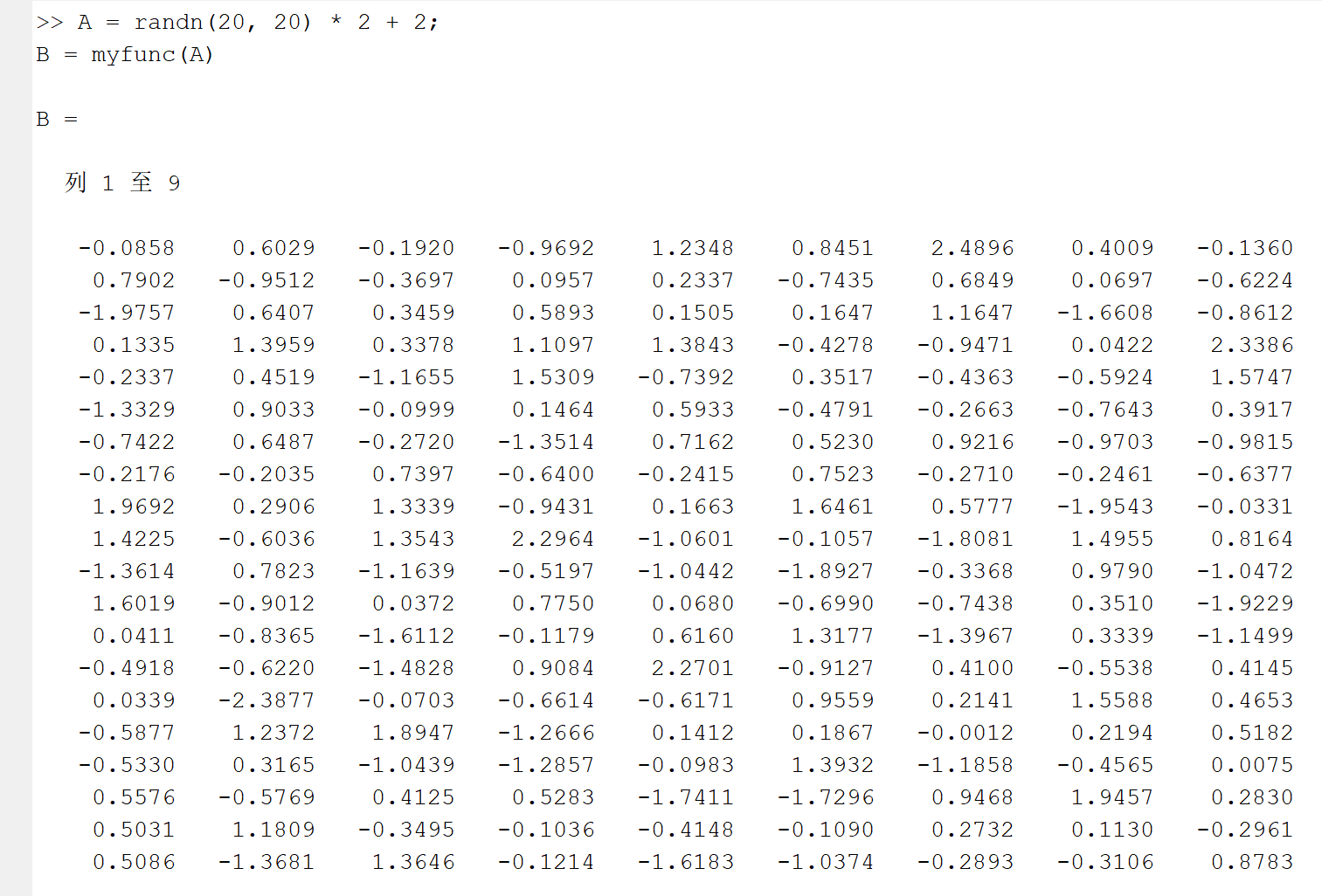
end

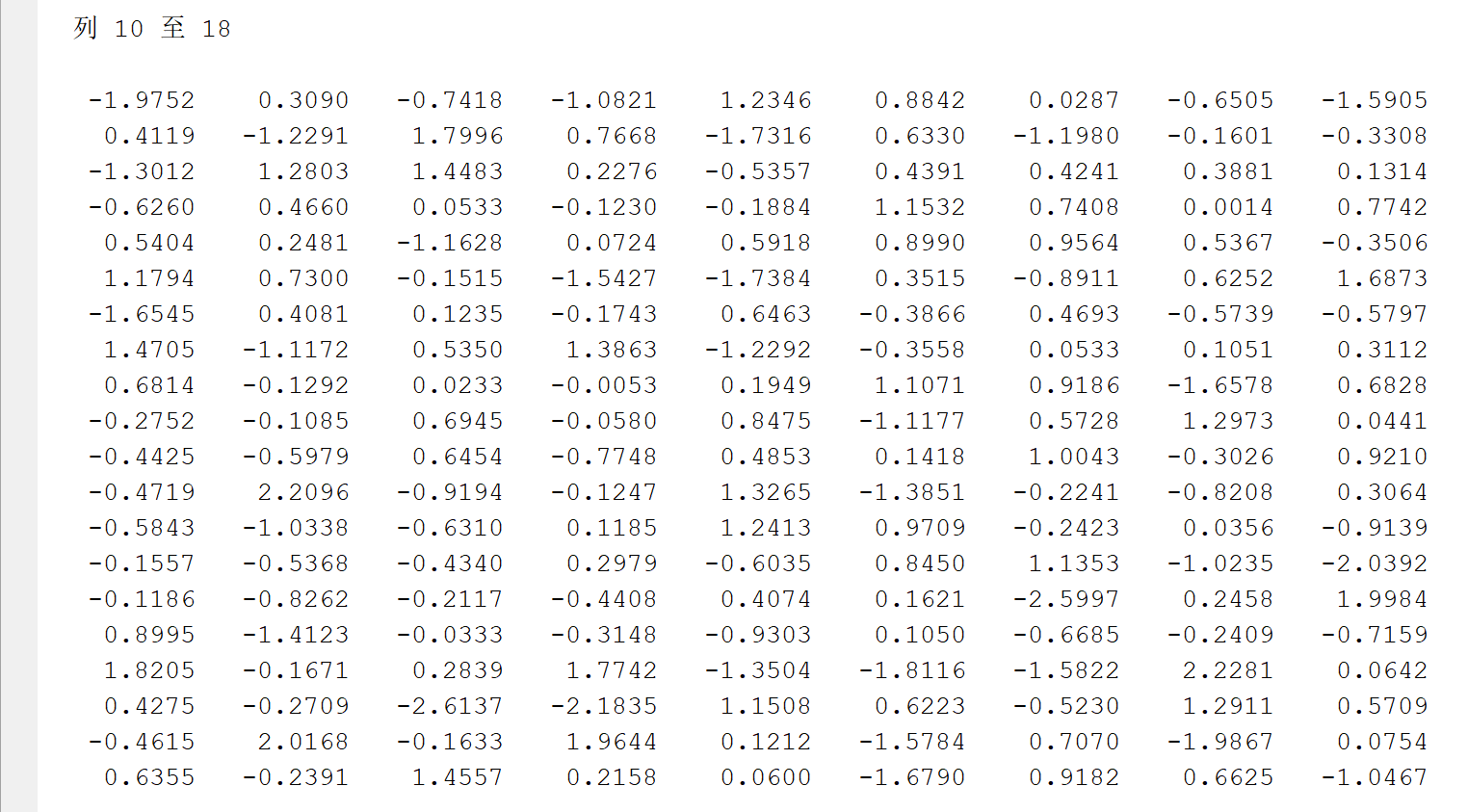
A = randn(20, 20) \* 2 + 2;

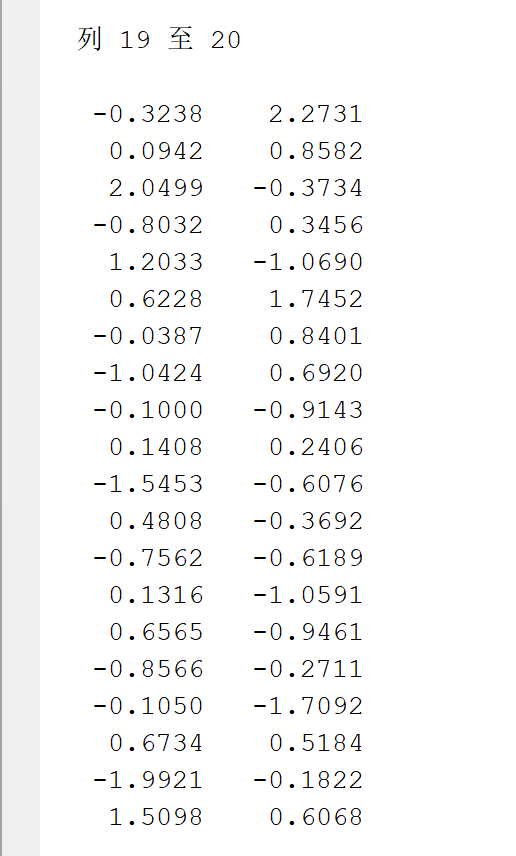
B = myfunc(A);

**运行结果的截图：**







****