

MATLAB语言快速入门与数据可视化

关注公众号：【数模加油站】，免费领取更多数模相关资料

1. MATLAB基本入门、矩阵及运算规则

1.1 软件介绍

1.2 界面使用

1.3 运算基础

1.4 矩阵基础

1.1 软件介绍

1.1.1 Matlab的介绍

1.1.2 Matlab的安装

1.1.3本章小节

1.1.4课后小练

1.1 软件介绍

1.1.1 Matlab的介绍

MATLAB 是美国 MathWorks 公司出品的商业数学软件，用于数据分析、无线通信、深度学习、图像处理与计算机视觉、信号处理、量化金融与风险管理、机器人，控制系统等领域。

MATLAB 是 matrix 和 laboratory 两个词的组合，意为矩阵实验室，软件主要面对科学计算、可视化以及交互式程序设计的高科技计算环境。它将数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等诸多强大功能集成在一个易于使用的视窗环境中，为科学研究、工程设计以及必须进行有效数值计算的众多科学领域提供了一种全面的解决方案，并在很大程度上摆脱了传统非交互式程序设计语言（如C、Fortran）的编辑模式。

1.1 软件介绍

1.1.2 Matlab的安装

版本选择：这里根据电脑价格给大家一个建议：价格小于等于 4000 元安装 MATLAB2017 版本、价格在 4000 到 8000 之间安装 MATLAB2021 版本，价格高于 8000 安装你能获得的最新版本。

MATLAB 的获取：

- ① 免费试用 1 个月：访问 MATLAB 官网：<https://ww2.mathworks.cn>
- ② 大学购买的正版 MATLAB 资源
- ③ 参加 MATLAB 赞助或支持的竞赛：
<https://ww2.mathworks.cn/academia/student-competitions.html>

1.1 软件介绍

1.1.4 课后小练

- 安装 MATLAB 软件

1.2.界面使用

1.2.1 Matlab界面介绍

1.2.2 创建Matlab脚本

1.2.3 Matlab文件管理

1.2.4 Matlab帮助系统

1.2.5本章小节

1.2.6课后小练

1.2.界面使用

1.2.1 Matlab界面介绍

MATLAB 的界面默认分成了四个大的区域：

- 最上方的**菜单栏**：包含主页、绘图和 APP(应用程序或工具箱)；
- 左侧的**当前文件夹**：用来快速查看并访问文件夹中的文件；
- 中间的**命令行窗口**：可以在命令行中输入命令（由提示符 (>>) 开始）；
- 右侧的**工作区**：可以用来查看目前 MATLAB 内存中保存的所有变量或者对象。

1.2.界面使用

1.2.2 创建Matlab脚本

利用.m 后缀的脚本文件（又称为 m 文件）编程：

- 点击MATLAB 菜单栏：主页——新建——脚本，或者使用快捷键Ctrl+N
- 点击MATLAB 菜单栏：编辑器——保存，或者使用快捷键Ctrl+S
- 点击MATLAB 菜单栏：编辑器——运行，或者使用快捷键F5

1.2.界面使用

1.2.2 创建Matlab脚本

强大的实时脚本：

MATLAB 从 2016 的版本开始，引入了实时脚本的功能，实时脚本的文件后缀是.mlx。

你可以把实时脚本理解成一个非常好用的交互式文档，在这个文档中也可以写上代码，还可以加上说明性的文本，甚至可以插入图片和公式。在执行代码时，返回的结果可以和代码一起显示 和保存。（熟悉 Python 的同学应该知道，这个功能类似于 Jupyter Notebook）

1.2.界面使用

1.2.3 Matlab文件管理

修改当前文件夹常用方式：

- ① 点击浏览文件夹，然后选择需要修改到的文件夹即可



1.2.界面使用

1.2.3 Matlab文件管理

修改当前文件夹常用方式：

- ① 点击浏览文件夹，然后选择需要修改到的文件夹即可
- ② 右键单击编辑器打开的m 文件名称，选择第一个选项



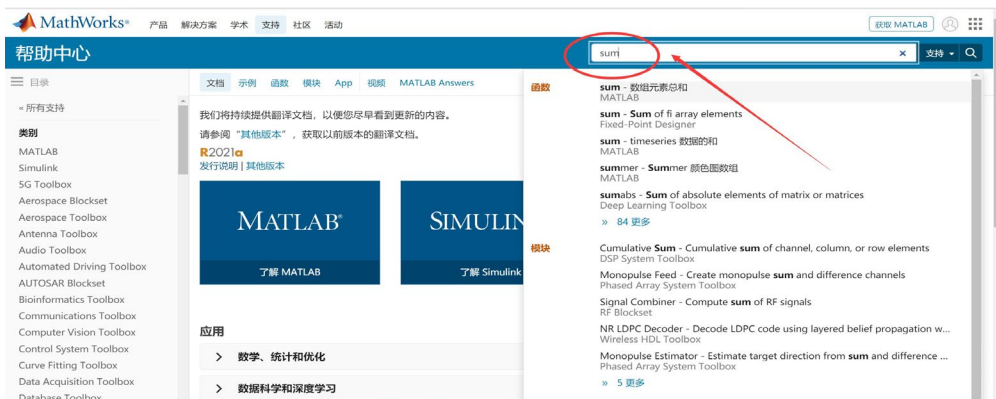
1.2.界面使用

1.2.4 Matlab帮助系统

查找 MATLAB 求和函数 “sum” 的帮助信息为例：

① 在 MATLAB 官网搜索

MATLAB 的帮助中心：<https://ww2.mathworks.cn/help/index.html>



1.2.界面使用

1.2.4 Matlab帮助系统

查找 MATLAB 求和函数“sum” 的帮助信息为例：

- ① 在 MATLAB 官网搜索
- ② 使用 help 命令（有网络和无网络连接的情况下得到的信息可能会有差异）

命令行窗口

```
>> help sum
sum Sum of elements.
  S = sum(X) is the sum of the elements of the vector X. If X is a matrix,
  S is a row vector with the sum over each column. For N-D arrays,
  sum(X) operates along the first non-singleton dimension.

  S = sum(X,DIM) sums along the dimension DIM.

  S = sum(...,TYPE) specifies the type in which the
  sum is performed, and the type of S. Available options are:

  'double' - S has class double for any input X
  'native' - S has the same class as X
  'default' - If X is floating point, that is double or single,
              S has the same class as X. If X is not floating point,
              S has class double.
```

1.2.界面使用

1.2.5本章小节

- 了解了 MATLAB 的界面。MATLAB 的界面由各个区域构成。
- 学习了如何在 MATLAB 中创建后缀为.m 的普通脚本，并给大家介绍了在 MATLAB 新版本中引入的更强大的后缀为.mlx 的实时脚本。
- 介绍了打开 MATLAB 代码的方式，推荐大家通过修改 MATLAB 当前文件夹的方式来打开代码。
- 介绍了 MATLAB 的帮助系统，与同类软件相比，MATLAB 的帮助系统非常完善，大家在未来的学习中要经常查阅 MATLAB 的帮助文档。

1.2.界面使用

1.2.6课后小练

请完成下面的一系列任务：

第一步：新建一个名为“`ass_1_2`”的文件夹。

第二步：打开 **MATLAB**，新建一个脚本文件，在里面任意输入一些你学过的代码。

第三步：将该脚本保存到“`ass_1_2`”的文件夹中，命名为“`ass_1_2.m`”。

第四步：更改 **MATLAB** 的当前文件夹为“`ass_1_2`”的文件夹。

第五步：执行你写的脚本，将计算结果作为注释放到每一行的后面。

第六步：清空工作区和命令行窗口。

1.3.运算基础

1.3.1 Matlab的变量

1.3.2 基本运算符的使用

1.3.3矩阵运算

1.3.4 逻辑运算

1.3.5本章小节

1.3.6课后小练

1.3.运算基础

1.3.1 Matlab的变量

在MATLAB 中变量的命名应遵循如下规则:

- 变量名必须以字母开头，之后可以是任意的字母、数字或下划线_。
- 变量名区分字母的大小写，例如 a 和 A 代表不同的变量。
- 变量名不超过 63 个字符，第 63 个字符以后的字符将被忽略。
- 不能定义与 MATLAB 关键字同名的变量（例如 if 或 end）。要获取关键字的完整列表，请在命令行输入 iskeyword 并运行。
- 特殊变量：ans、pi、inf/-inf等

MATLAB 使用等号 “=”给变量赋值

1.3.运算基础

1.3.1 Matlab的变量

MATLAB 数据类型:

- 数字：整数和浮点数
- 字符与字符串：' ' 与" "
- 矩阵：[]

1.3.运算基础

1.3.2 基本运算符的使用

针对数值、字符或者逻辑值

➤ 加法： +

➤ 减法： -

➤ 乘法： *

➤ 除法： /

➤ 乘方： ^

1.3.运算基础

1.3.3矩阵运算

1.3.4 逻辑运算

1.3.运算基础

1.3.5本章小节

- MATLAB 变量命名的规范， MATLAB 中预定义的特殊变量。
- MATLAB常用变量类型：数值、字符和字符串及矩阵。
- 基本运算符的使用，加减乘除。

1.3.运算基础

1.3.6课后小练

对两个字符变量做基本运算。

1.4.矩阵基础

1.4.1 矩阵的创建

1.4.2 矩阵元素的修改删除

1.4.3 矩阵的拼接重构重排

1.4.4 矩阵的运算

1.4.5本章小节

1.4.6课后小练

1.4.矩阵基础

1.4.1 矩阵的创建

在MATLAB中，矩阵的创建方法主要有三种，分别是:直接输入法、函数创建法和导入本地文件中的数据。

- ① 直接输入法
- ② 函数创建法
- ③ 导入本地文件中的数据

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

1.4.矩阵基础

1.4.1 矩阵的创建

在MATLAB中，矩阵的创建方法主要有三种，分别是:直接输入法、函数创建法和导入本地文件中的数据。

① 直接输入法

适用于矩阵中元素数量较少的情况。

输入矩阵时要以中括号 [] 作为标识符号，矩阵的所有元素必须都在中括号内。矩阵的同行元素之间用空格或逗号分隔，行与行之间用分号或回车键分隔。

1.4.矩阵基础

1.4.1 矩阵的创建

- ① 直接输入法
- ② 函数创建法

MATLAB提供的用来生成某些特定的矩阵的函数，常用函数：

第一组函数: zeros、ones 和 eye。这三个函数分别用来创建全为0的矩阵、全为1的矩阵和单位矩阵。

第二组函数: rand、randi和randn。这三个函数分别用来创建均匀分布的随机数、均匀分布的随机整数和标准正态分布的随机数。

1.4.矩阵基础

1.4.1 矩阵的创建

- ① 直接输入法
- ② 函数创建法
- ③ 导入本地文件中的数据

MATLAB可读取本地的文件，支持的常见格式如下：

- txt、.dat或.csv (适用于带分隔符的文本文件)
- xls、.xlsb、.xlsm、.xlsx、.xltm、.xltx或.ods (适用于电子表格文件)

1.4.矩阵基础

1.4.2 矩阵元素的修改删除

矩阵元素的修改：

- 直接利用等号赋值的方法对矩阵中引用位置的元素进行修改
- 使用线性索引（单下标的索引）的方式对矩阵的元素进行修改
- 注意：如果你在赋值时将一个或多个元素置于矩阵现有的行和列索引的边界之外，则会将矩阵的大小进行拓展，**MATLAB** 会将没有赋值的位置的元素自动用 0 填充，使其保持为完整的矩形。

1.4.矩阵基础

1.4.2 矩阵元素的修改删除

矩阵元素的删除：

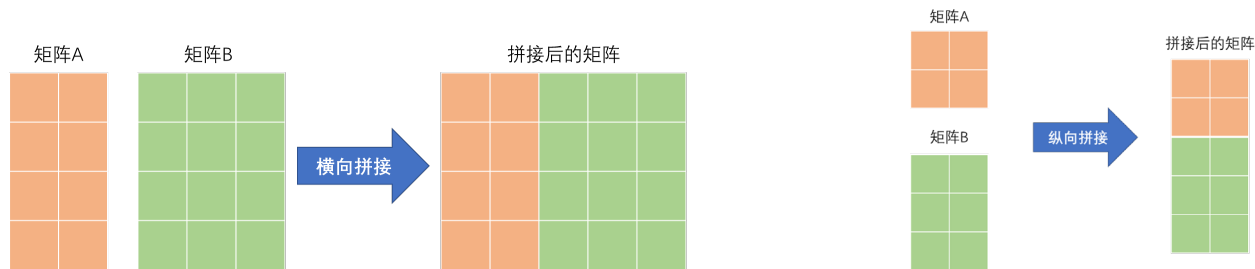
- 如果我们将等号右侧变成空向量`[]`，则可以删除对应位置的元素。需要注意的是，通常只能**删除矩阵的整行或者整列**，否则会报错。
- 也可以通过线性索引来删除矩阵的元素。使用线性索引删除后，MATLAB 会将矩阵中剩下的元素按照线性索引的顺序放入到一个向量中。另外，使用线性索引可以删除任意位置的元素，不需要删除矩阵的一整行或者一整列。

1.4.矩阵基础

1.4.3 矩阵的拼接重构重排

矩阵的拼接：

- 横向拼接：A 和 B 的行数相同，那么使用 $[A, B]$ 、 $[A \ B]$ 以及 $\text{cat}(2,A,B)$ 都能将 A 和 B 横向拼接成一个大的矩阵。
- 纵向拼接：A 和 B 的列数相同，那么使用 $[A; B]$ 以及 $\text{cat}(1,A,B)$ 都能将 A 和 B 纵向拼接成一个大的矩阵。



1.4.矩阵基础

1.4.3 矩阵的拼接重构

矩阵的重构重排：

- **reshape函数**：更改矩阵的形状，`reshape(A, m, n)`或者 `reshape(A,[m,n])`
- **sort 函数**：对向量或者矩阵进行排序，`sort(A, dim)`，在最后面加一个输入参数'descend'，变成从大到小的降序排列
 - `dim = 1` 时，沿着行方向(从上至下)对矩阵的每一列升序排列
 - `dim = 2` 时，沿着列方向(从左至右)对矩阵的每一行升序排列
- **sortrows函数**：基于矩阵的某一列对矩阵进行排序，同一行的元素不会改变。`sortrows(A,列)`，在最后面加一个输入参数'descend'，变成从大到小的降序排列

1.4.矩阵基础

1.4.4 矩阵的运算

调用函数运算

善用帮助文档!!!

函数名	函数的作用
sum	求和函数
prod	求乘积函数 (product)
cumsum	计算累积和 (cumulative sum)
diff	计算差分 (difference)
mean	计算平均值
median	计算中位数
mode	计算众数
var	计算方差 (variance)
std	计算标准差 (standard deviation)
min	求最小值 (minimum value)
max	求最大值 (maximum value)

1.4.矩阵基础

1.4.4 矩阵的运算

算数运算

➤ 矩阵加减

五种算术运算兼容模式

按对应位置的元素运算

情形	示例：计算 $A+B$		计算结果	解释
两个大小完全相同的输入	A = 6 5 6 2 9 2	B = 7 8 9 8 6 2	ans = 13 13 15 10 15 4	将 A 和 B 对应位置的元素相加
有一个输入是标量（常数）	A = 2 1 3 7 2 4	B = 4	ans = 6 5 7 11 6 8	矩阵的每个元素都加上这个标量
一个输入是矩阵，另一个输入是具有相同行数的列向量	A = 3 6 5 2 6 8	B = 6 5	ans = 9 12 11 7 11 13	把 B 堆叠成完全相同的三列，然后再和 A 相加 相当于 <code>repmat(B,1,3)</code>
一个输入是矩阵，另一个输入是具有相同列数的行向量	A = 3 5 6 6 9 4	B = 3 9 6	ans = 6 14 12 9 18 10	把 B 堆叠成完全相同的两行，然后再和 A 相加 相当于 <code>repmat(B,2,1)</code>
一个输入是列向量，另一个输入是行向量。	A = 2 5	B = 1 8 3	ans = 3 10 5 6 13 8	把 A 堆叠成完全相同的三列，把 B 堆叠成完全相同的两行，然后相加 相当于 <code>repmat(A,1,3)</code> <code>repmat(B,2,1)</code>

1.4.矩阵基础

1.4.4 矩阵的运算

算数运算

- 矩阵加减
- 矩阵乘除乘方
 - $*$ 和 $.*$ 分别表示现代中的矩阵相乘和两个矩阵中对应元素相乘
 - $/$ (右除)、 \backslash (左除) 和 $./$
 - $^$ 和 $.^$
- 矩阵转置 ‘

1.4.矩阵基础

1.4.4 矩阵的运算

关系运算

==	~=	>	>=	<	<=
等于	不等于	大于	大于等于	小于	小于等于

关系运算符可以用来比较两个数组中元素的关系，如果比较的结果为真，则MATLAB 会返回**逻辑值 1**；如果结果为假，则会返回**逻辑值 0**。

1.4.矩阵基础

1.4.5本章小节

- 矩阵的三种创建方式，直接输入、函数创建和导入数据。
- 矩阵元素的修改删除，引用位置和线性索引两种方式。对矩阵元素的删除中若使用引用位置一般是删除整行或者整列。
- 矩阵的拼接重构，横向和纵向拼接，直接拼接和cat函数。
- 矩阵的运算，调用函数、算术运算和关系运算。

1.4.矩阵基础

1.4.6课后小练

假设有6名学生，矩阵的每一行代表一名学生。这六名同学的四门科目的成绩对应着四列，例下面这个矩阵保存着这六名同学在四门科目上的成绩。如第一名同学的第一科成绩为95，第二科成绩为80，依此类推。

$$\text{score} = \begin{bmatrix} 95 & 80 & 85 & 79 \\ 95 & 67 & 78 & 90 \\ 95 & 67 & 78 & 75 \\ 95 & 67 & 64 & 73 \\ 86 & 85 & 82 & 84 \\ 86 & 87 & 84 & 88 \end{bmatrix}$$

- (1) 请基于第二科的成绩按升序对这六名同学进行排序。当第二科成绩相同时，请保持其在矩阵中出现的先后顺序。
- (2) 请基于第一科的成绩按升序对这六名同学进行排序。当第一科成绩相同时，基于第三科成绩升序排列。如果第一科和第三科都相同，就保持在矩阵中出现的先后顺序。
- (3) 请基于第一科的成绩对这六名同学进行降序排列。当第一科成绩相同时，基于第三科成绩降序排列。如果第一科和第三科都相同，就保持在矩阵中出现的先后顺序。