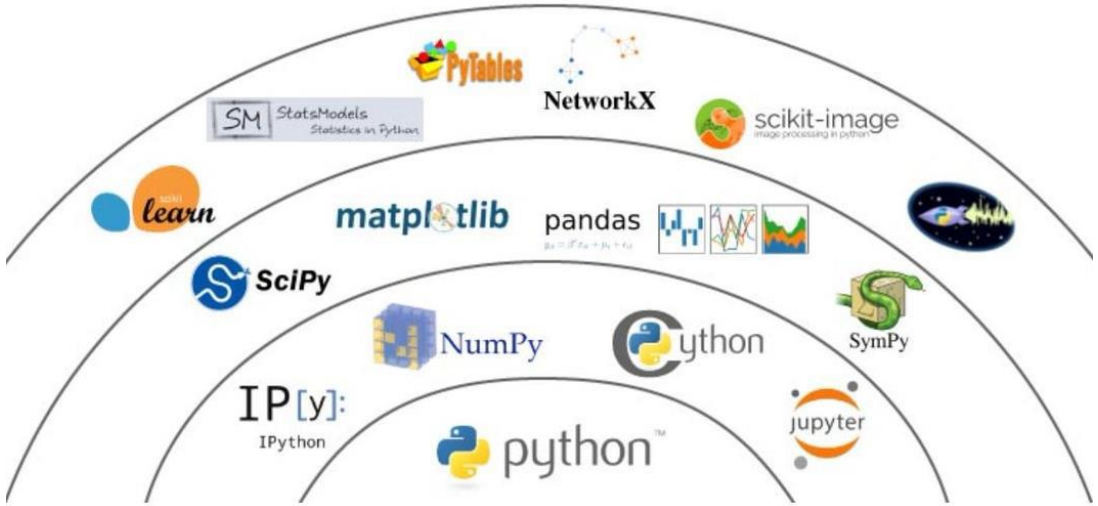
04 NumPy数据处理

黄雅婷 13810090248

北京工商大学 人工智能学院

课程内容

- 基本数据结构：ndarray
- 数组生成
- 数组属性与函数
- 数组索引
- 矢量运算
- NumPy数组文件存取



Python数据科学生态系统

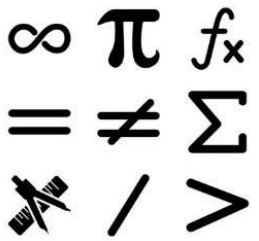
数据处理基本步骤

TimeSeriesData.csv

1.读取数据

2.数据预处理

基本数学



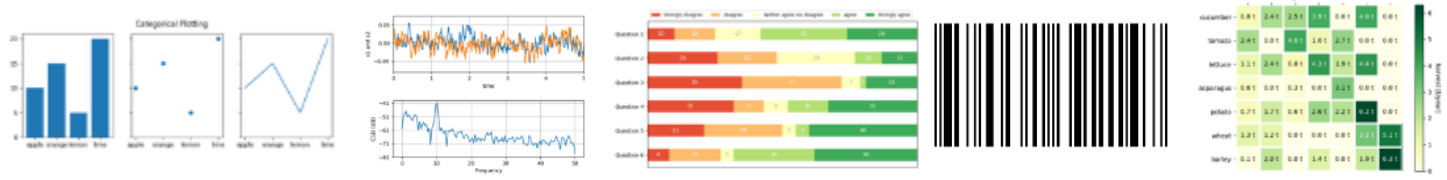
科学计算

3.数据分析

机器学习

Machine Learning with Scikit-Learn

4.数据可视化



5. 分析与结论

环境配置

- 必备库 NumPy 、 pandas 、 matplotlib

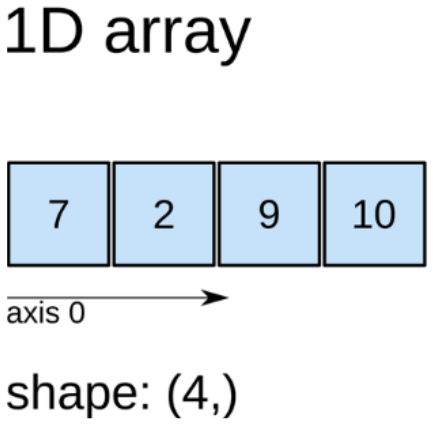
```
[3]  M↓
     import numpy as np

---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
 in
----> 1 import numpy as np

ModuleNotFoundError: No module named 'numpy'
```

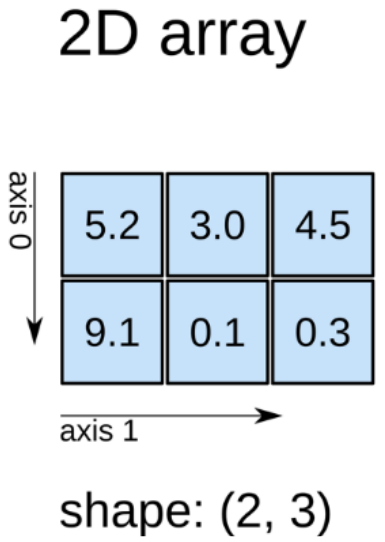
- 自己配置：终端输入
  - pip install numpy
  - pip install pandas
  - pip install matplotlib

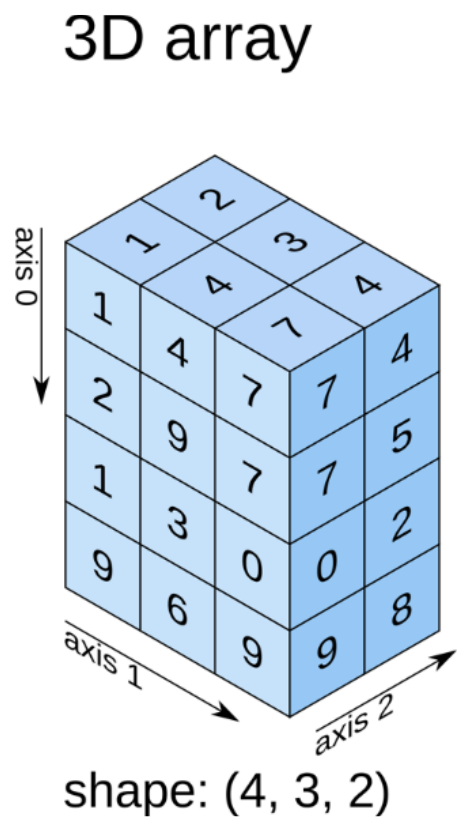
```
[1]: ## 画图必备
     import numpy as np
     import matplotlib.pyplot as plt
     import pandas as pd
```

NumPy基础

- ndarray数组对象
- 创建数组
- 数组属性与函数
- 数组索引
- 矢量运算







NumPy-矢量运算

- 相同尺寸：逐个元素运算

$$\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix} + - \times \div \begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix} = ?$$

- 不同尺寸：
  - 标量*向量：向量的数乘 $10 \times (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6) = (0 \quad 10 \quad 20 \quad 30 \quad 40 \quad 50 \quad 60)$
  - （4x3）矩阵+（1x3）向量
    -（数组广播）

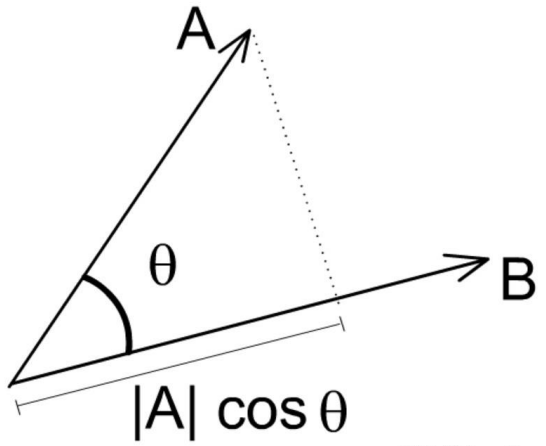
$$\begin{pmatrix}0 & 0 & 0\\ 1 & 1 & 1\\ 2 & 2 & 2\\ 3 & 3 & 3\end{pmatrix} + (1 \quad 2 \quad 3) = ?$$

线性代数乘法

- 向量点乘
- (得到标量)

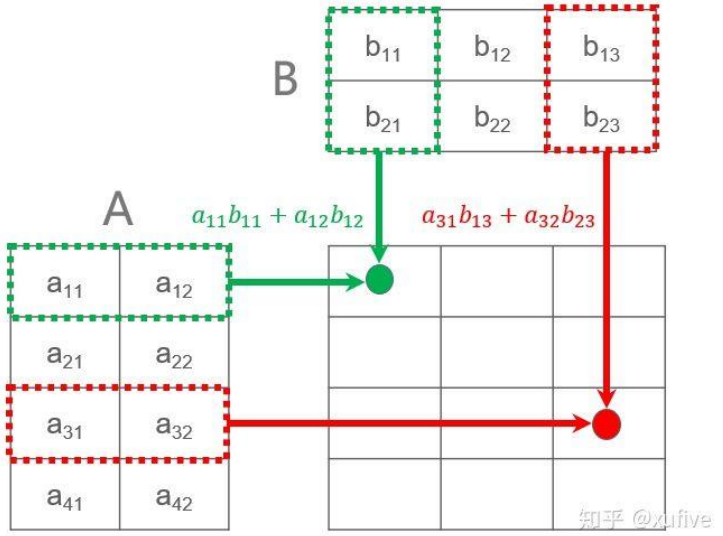
```
np.dot(vec1, vec2)
```

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}|cos\theta = x_1 x_2 + y_1 y_2$$



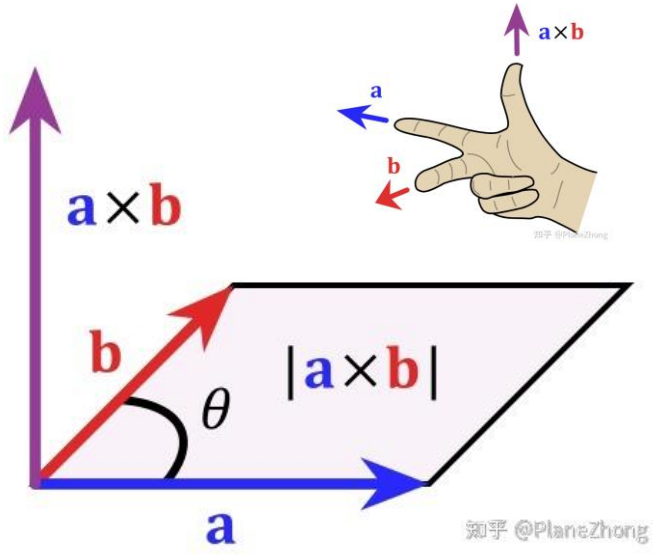
- 矩阵点乘
- （注意维度）

```
arr1.dot(arr2)
```



- 向量叉乘

```
np.cross(vec1, vec2)
np.cross(vec2, vec1)
```



NumPy数组文件

- 这是二进制文件，不是文本文件
- NumPy用于储存数组数据的标准二进制文件类型叫做“.npy”文件。
- NumPy用以在单个文件中储存多个数组的二进制档案格式叫做“.npz”格式。
- save，savez和load

```
[82]: x = np.array([1, 2, 3])
      path_to_np = root / 'L03_documents' / "my_array.npy"
      # 将NumPy数组存入硬盘
      np.save(path_to_np, x)

[84]: # 从硬盘读取NumPy数组
      y = np.load(path_to_np)
      y

[84]: array([1, 2, 3])

[85]: # 将三个数组储存到NumPy档案文件中
      a0 = np.array([1, 2, 3])
      a1 = np.array([4, 5, 6])
      a2 = np.array([7, 8, 9])

      path_to_npz = root / 'L03_documents' / "my_arrays.npz"
      # 我们使用关键词参数 `soil`，`crust`，和 `bedrock` 来
      # 作为档案中对应数组的名字。
      np.savez(path_to_npz, soil=a0, crust=a1, bedrock=a2)

[88]: # 打开档案并通过名字访问每个数组
      with np.load(path_to_npz) as my_archive_file:
          out0 = my_archive_file["soil"]
          out1 = my_archive_file["crust"]
          out2 = my_archive_file["bedrock"]
      out0, out1, out2

[88]: (array([1, 2, 3]), array([4, 5, 6]), array([7, 8, 9]))
```

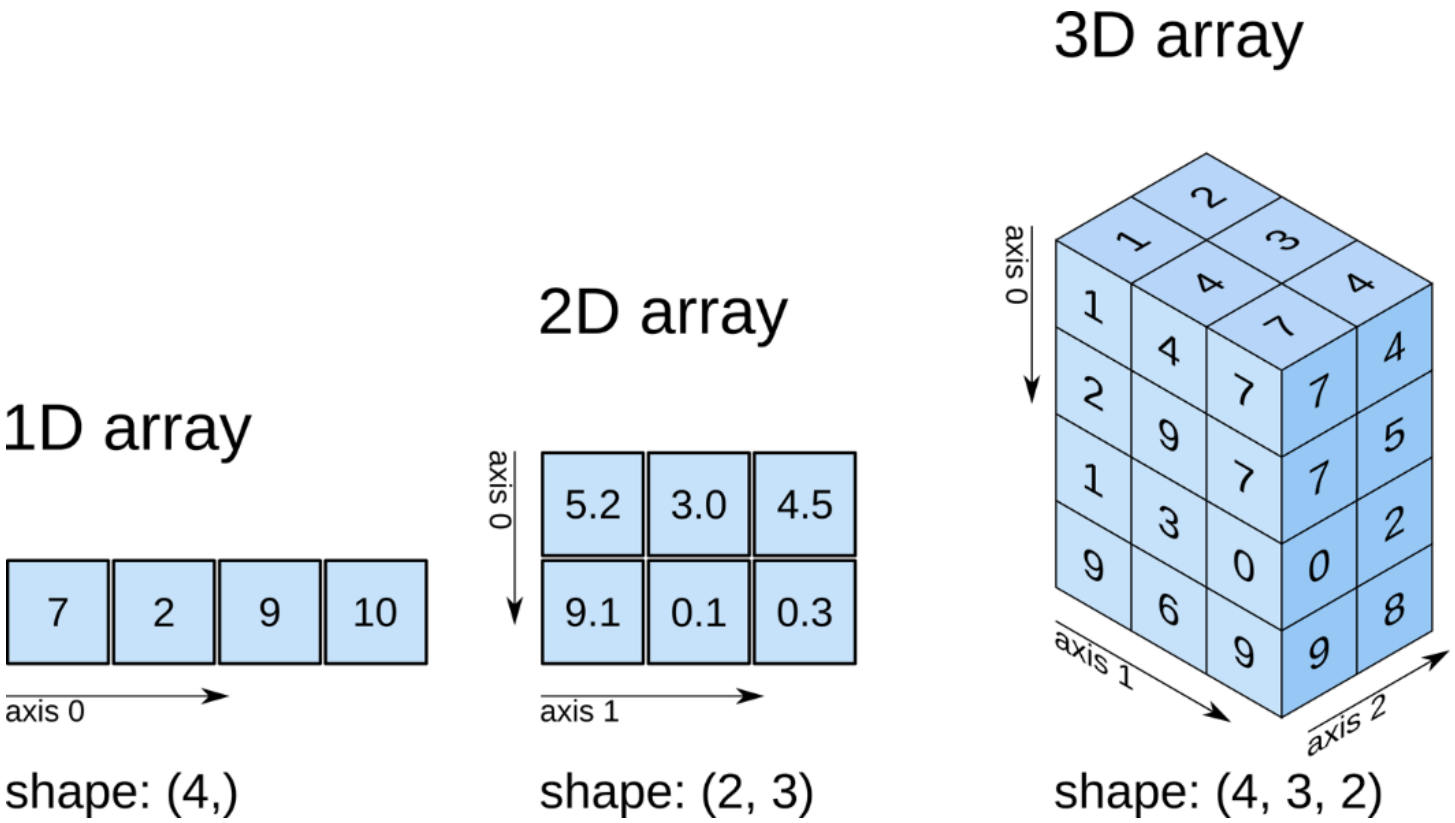
作业：

- 完成三个练习

主观题 10分

设置

练习：创建图中所示的数组（看不见的数自己编）



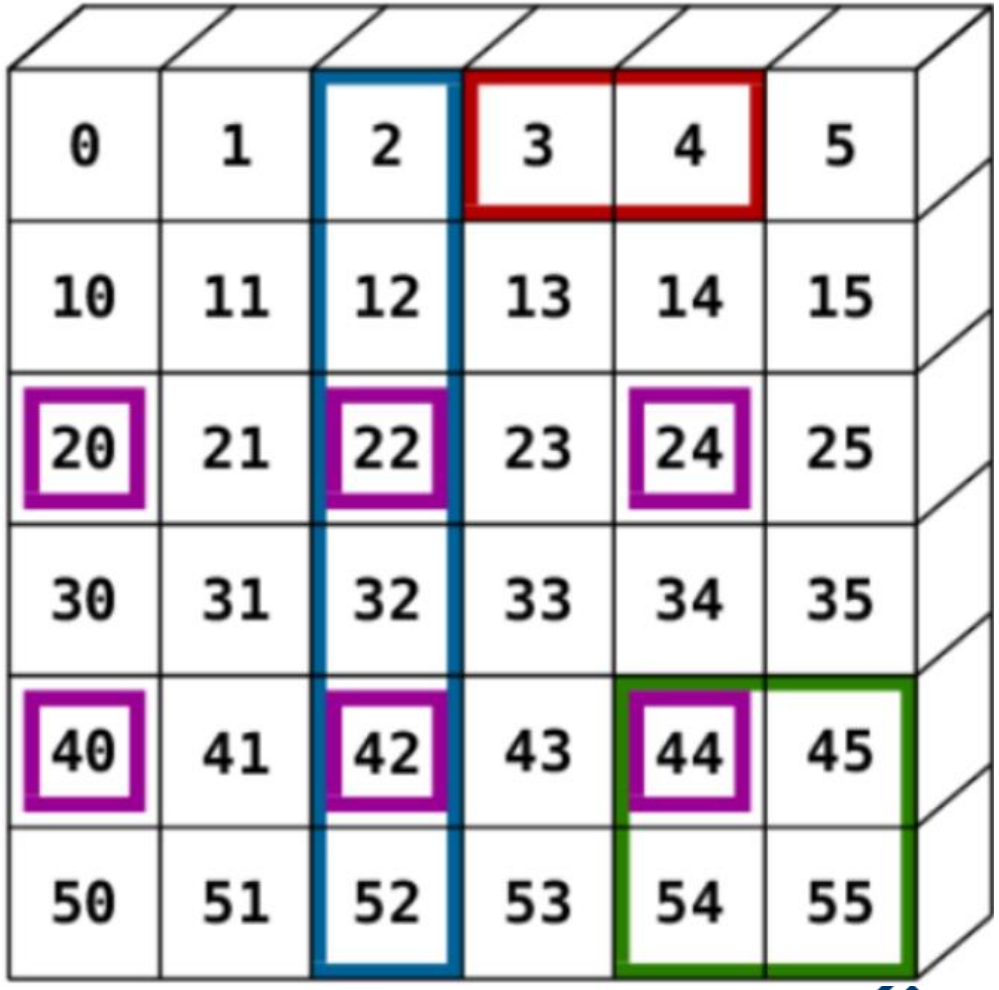
作答

主观题 10分

设置

练习：取出图中矩阵不同颜色对应的内容

作答



计算每个同学每一门成绩和平均成绩的差值

应用场景：计算学生成绩

假设你有着6个学生的成绩簿，每个成绩簿有着3个考试的成绩，你将这些成绩储存在一个形状为 (6,3) 的数组中

```
grades = np.array([[ 0.79,  0.84,  0.84],
                   [ 0.87,  0.93,  0.78],
                   [ 0.77,  1.00,  0.87],
                   [ 0.66,  0.75,  0.82],
                   [ 0.84,  0.89,  0.76],
                   [ 0.83,  0.71,  0.85]])
```

作答