

情景：

假设我们试图从一个“大理石”块（marble blocks）雕刻出一个雕塑，它的3D形状由一个3D数组给出。只包含1s 和 0s 来表示它是实心还是空心。(此部分被移除).

我们还有一个3D 数组的大型三维的浮点数。这些数字代表一块大理石内的局部密度，可能是使用了一些外部测量工具如X光或者超声波从外面对大理石的三个坐标轴扫描的。

在我们雕刻这些大理石块（marble blocks）的形状之前，我们想要确定每个大理石块的最佳雕刻方向（best orientation (rotation)），最佳方向由计算平均密度和稳定性以确定。

输出结果如下图形式，但是这些数字的排列规律是，每个block的结果，根据稳定性然后按照密度降序排列：

Shape File: shape_1.npy

Block File: marble_block_1.npy

Rotation: None	Mean density: 79.44	Stable
Rotation: 180 axis 0 to 1	Mean density: 78.24	Stable
Rotation: 180 axis 0 to 1, 90 axis 1 to 2	Mean density: 48.24	Stable
Rotation: 90 axis 1 to 2	Mean density: 64.11	Unstable
...		

Block File: marble_block_2.npy

...

Shape File: shape_2.npy

Block File: marble_block_1.npy

Rotation: 180 axis 0 to 1	Mean density: 82.04	Stable
Rotation: None	Mean density: 71.44	Stable
Rotation: 90 axis 1 to 2	Mean density: 64.11	Unstable
Rotation: 180 axis 0 to 1, 90 axis 1 to 2	Mean density: 48.24	Unstable
...		

要求1

1. 在被雕刻之后，哪个大理石 (marble_block)和哪个方向得到了最高的平均密度。给予了你几个大理石 (marble_block)的信息数据文件，但是编写程序使得程序可以读取并且计算大量类似于这种的数据文件，请不要复制粘贴。你的输出应该显示答案以及大理石块的名字和使用的哪个旋转 (rotations)，你需要使用Numpy的nanmean()函数来获取正确的密度。你这时候会发现float16溢出了，所以在使用nanmean()之前，转换成使用astype('float32')。

要求2

对于每个可能的方向，确定雕塑是否稳定和平衡。因此，您需要计算质心所在的位置，并确保它位于其底部的水平边界内。

有用的资源:

使用 SciPy 包函数计算质心. 请见: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.center_of_mass.html

这几个网页有助于理解要求2:

- <https://www.khanacademy.org/science/physics/linear-momentum/center-of-mass/v/center-of-mass-equation>

- <https://www.khanacademy.org/science/physics/linear-momentum/center-of-mass/a/what-is-center-of-mass> (本页中的“Topple Limit”部分有助于解释上面的要求 2。但是您不需要计算倾角)

计算质心位置后，您会发现另一个 SciPy 类对于确定问题 2 中的雕塑稳定性非常有帮助。应使用 ConvexHull 来确定质心是否位于底座“足迹”内的某个点上方:

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.ConvexHull.html> .另外给的一段Convex的python文件也展示了这个。

如果可以的话：

尝试尝试持续集成。如果您成功完成此操作，但在其他地方出现错误，则最多可以替换 10% 的分数。