**Part 1**

文本

低可信度描述已自动生成这部分作业包括创建一段代码，可以读取两个价格列表(在CSV文件中给出)。第一个列表包含一些产品的当前价格，第二个列表是相同产品的相应参考价格。你的代码应该计算当前产品相对于参考价格的相对折扣。对于当前价格P和参考价格R的产品，相对折扣D应计算为:

注意，如果产品的价格增加(C > R)， D的值将是负的。折扣按折扣程度分为四类:

表格

描述已自动生成

你的任务是添加代码到提供的模板文件，以完成以下三个功能: read\_files(current\_price\_filename, reference\_price\_filename) -这个函数读取输入的CSV文件，并返回两个Numpy数组，其中包含文件中的值。这些应该被命名为current，用于当前价格和参考价格。

discount = calculate\_discount(current, reference) -此函数使用前一个函数读取的值来计算相同大小的Numpy数组，该数组包含根据上述公式计算的每个产品的折扣。

price\_classification = classify\_prices(discount)—该函数使用带有折扣的数组作为输入，并输出一个列表，其中包含每个价格的价格分类作为文本字符串。请注意，文本字符串应该与表格中描述的完全一样(即，没有空格，大写字母等)。

**提交的文件应该是基于提供的模板的单个Python文件。这部分作业只需要Numpy。代码将使用不同的价格列表进行测试，因此它应该足够强大，能够处理各种有效的情况。例如，这些文件总是只包含数字，并且大小相同，但是它们可能包含不同数量的产品价格。**

**Part 2**

**Task 1 -数据处理:**

给你一个数据集（“Vegetables.csv”），其中包含一些商品的价格信息2013 年美国蔬菜批发价格。每月和每年提供数据，并且分为几个类别：是有机的还是传统的，价格是在哪个终端记录（亚特兰大或旧金山）。

您的任务包括使用 Pandas 读取数据并创建以下四个函数：

1. read\_vegetable\_data(filename) – 该函数读取正确提供的数据文件并返回数据框“df”

2. generate\_plot\_1(df) – 此函数创建一个图 ( Plot1 )，其中 X 轴代表2013 年的月份，Y 轴代表价格。每个情节应该有一行蔬菜类型（即一行表示卷心菜，一行表示胡萝卜等），它将代表每个月蔬菜价格的平均价格。

3. generate\_plot\_2（df） -该函数创建一个图（Plot2）只有一个蔬菜类（如只有生菜），它绘制了有机蔬菜和蔬菜的平均价格随着时间的推移，传统的对应物。该图应有两行（一为有机，一为传统的）。

4. find\_cheapest\_month(df,vegetable,organic) ——这个函数应该返回月份（如从文件中提取的文本，例如“Jan-13”），其中蔬菜“vegetable”是所有车站最便宜。如果“organic”取值为True，则它应该指代有机版本，否则为常规版本。例如，find\_cheapest\_month (df,'Carrots', False)应该返回 'Feb-13’

**你必须完成这些，以确保:**

**数据集不会以任何方式被改变(即它不会在Excel或类似的方式读入Python之前被修改)。**

**无效数据N\A会被忽略，你可能会删除一些缺少数据的行。**

**您可以按照提供的Python模板来编写函数，而不需要更改它们的名称或输入参数。一旦填充了函数，运行代码将提供所需的所有输出。**

**您可能需要查阅官方的Pandas文档来完成这项任务。您必须提供一篇简短的评论(200字)，描述您为此使用的Pandas的高级功能，以及您如何处理出现的挑战(即无效或丢失数据等)。**

**Task 2骰子滚动:**

在这项任务中，您应该创建一个名为“dice\_probabilities（n\_ dice，replications）”的函数，该函数能够通过模拟大量随机掷骰子（replications）并观察输出来估计掷给定数量骰子（n\_dice）的所有可能结果的概率。如果有多个骰子，则输出被认为是所有骰子的总和。骰子被认为是标准的六面骰子。

函数应该返回两个Numpy数组（或Python列表）：dice\_values，dice\_probabilities。在dice\_values，骰子滚动的结果应该被存储，而dice\_probabilities应该包含相应滚动发生的概率。概率必须是介于0和1之间的数字（其中1代表100%）。请注意，这些应该是估计，不会是精确的，特别是对于“replications”的低值。

我们为您提供了一个模板文件来编写函数，这里我们提供了两个可能的代码输出示例

*For n\_dice = 1 and replications = 1000:*

dice\_values: [1 2 3 4 5 6]

dice\_probabilities: [0.173 0.172 0.176 0.148 0.164 0.167]

*For n\_dice = 2 and replications = 1000000*

dice\_values:[2 3 4 5 6 7 8 9101112]

dice\_probabilities: [0.027858 0.056126 0.083222 0.110896 0.138488 0.166662 0.138903

0.111272 0.083266 0.055206 0.028101]

**注意，这些概率可以精确地计算出来，但是这个任务并不需要这样做。答案必须是描述的模拟。**

**TASK 3接种疫苗预约分配:**

在这项任务中，您需要构建一个软件，该软件可以将患者分配到疫苗接种中心，考虑到该中心当天的可用性、中心与患者之间的距离以及患者是否有资格获得疫苗。

你会得到两个CSV格式的数据集，有以下标题:

Patient dataset:

Patient dataset: patient\_id, x\_coord, y\_coord, eligible

Centre dataset: centre\_id, x\_coord, y\_coord, capacity

id是用于识别患者和中心的自然数字，您的解决方案在报告分配时应该引用这些id。坐标是0到100之间的数字。对于患者来说，有资格接种疫苗的为1，没有资格接种的为0。只有符合条件的患者才能分配。该中心的容量是指每天的容量，即每天可以为病人接种疫苗的人数。除此以外，您的代码将有一个参数“days”(已在模板中给出)，它给出了可以接种疫苗的最大天数。例如，如果day= 3，患者必须分配在第0天、第1天或第2天接种疫苗。

**分配必须按照以下方式进行:** 按照给定的顺序浏览患者名单，符合条件的患者必须被分配到最近有能力为他们接种疫苗的中心。在该中心内，应尽可能早地安排他们。疫苗接种必须在特定的时间段内完成，该时间段作为参数“days”提供给函数。这意味着一个中心在某个点上可能是完全满的，而不可能是中心必须计算为欧几里德距离(Euclidean distance.)

如果（px,py）为患者的坐标，（cx,cy）为疫苗中心的坐标， d为他们之间的距离，则公式为：

文本

低可信度描述已自动生成

代码应该不改变其结构，任务包括完成以下三个功能:

read\_data(file\_patients, file\_centres)，它读取两个CSV数据文件，并以numpy数组的形式返回其内容:patients和centres。

Euclidean\_distance (x1, y1, x2, y2)，实现了上述距离公式。

allocation(patients, centres, days) ，这取决于read\_data函数读取的数据和参数“天”，这是一个自然数。这个函数必须实现分配 并返回以下三个变量:patient\_allocation, centre\_usage, total\_distance

·Patient\_allocation: 应该是一个numpy数组，包含与符合条件的患者一样多的行和四列。对于每个患者，第一列必须包含其ID，第二列必须包含分配给他们的中心，第三列必须包含分配给他们的日期，第四列必须包含到达中心的距离，第四列必须包含分配给他们的日期。例如，第[6，1，3，31.06]行意味着ID为6的患者已在第3天分配到疫苗接种中心1，并且需要旅行31.06个单位才能到达该中心。

·centre\_usage:应该是一个numpy数组，与CSV文件中中心的行数一样多，与参数“days”给出的列数一样多。数组的一个元素centre\_usage[i,j]应该包含第j天分配给中心i的病人数。

·Total\_distance: 应为单个数字，即所有符合条件的患者获得疫苗所需的所有距离（仅单程）之和。

**注意，可能有更好的方法来创建一个更短/更公平的分配，但是对于这个任务，您需要执行上面描述的算法。您的代码可能会使用不同的输入文件和“days”值进行测试。**

对于输入文件和“天数”的某些组合，可能没有足够的能力为所有患者接种疫苗。在这种情况下，您的代码仍然应该运行并按照提出的算法分配尽可能多的病人。不能分配的病人(将是列表中的最后一个)仍然应该在patient\_allocation中获得一行，其中第一列有他们的ID，其余的值为“-1”。

**模板的结构不应该被修改，代码应该只添加到提供的函数中，其中的注释:**

# Please, introduce your answer here

请注意，这可能在文件中的多个位置。不应修改函数的名称、输入参数或返回的名称/数量。这样做可能会导致您的作业被错误地标记。所有模板代码都应该按照提供的方式运行，但会提供无效答案（通常将所有内容设置为零，也不是“尚未完成”）。

只有Numpy、Pandas和Matplotlib才允许/需要完成作业。这些已经在模板文件中适当地导入。**您可以更改的行被标记为:**

# Change/remove this line

**或类似的。您也可以选择保留这些行，并在稍后的函数中修改变量值。 只有Numpy、Pandas和Matplotlib被允许/要求完成任务。这些已经被适当地导入到模板文件中。如果您对模板中可能修改或不修改的内容有疑问，请联系模块讲师之一。**