**2.如何创建一维数组？（L1）**

**Q：创建一个从0到9的一维数组**

期望输出：

#> array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

**参考答案：**

arr = np.arange(10)

arr

#> array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

**3.如何创建布尔数组？（L1）**

**Q：创建一个3×3的所有真值的数组**

**参考答案：**

np.full((3, 3), True, dtype=bool)

#> array([[ True, True, True],

#> [ True, True, True],

#> [ True, True, True]], dtype=bool)

# Alternate method:

np.ones((3,3), dtype=bool)

**4.如何从一维数组中提取满足给定条件的项？（L1）**

**Q：从 arr 中提取所有奇数**

输入：

arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

期望输出：

#> array([1, 3, 5, 7, 9])

**参考答案：**

# Input

arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

# Solution

arr[arr % 2 == 1]

#> array([1, 3, 5, 7, 9])

**5.如何在 numpy 数组中替换满足条件与其他值的项？（L1）**

**Q：用-1替换 arr 中的所有奇数**

输入：

arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

期望输出：

#> array([ 0, -1, 2, -1, 4, -1, 6, -1, 8, -1])

**参考答案：**

arr[arr % 2 == 1] = -1

arr

#> array([ 0, -1, 2, -1, 4, -1, 6, -1, 8, -1])

**6.如何在不影响原始数组的情况下替换满足条件的项？（L2）**

**Q：在不改变 arr 的情况下，用1代替 arr 中的所有奇数**

输入：

arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

期望输出：

out

#> array([ 0, -1, 2, -1, 4, -1, 6, -1, 8, -1])

arr

#> array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

**参考答案：**

arr = np.arange(10)

out = np.where(arr % 2 == 1, -1, arr)

print(arr)

out

#> [0 1 2 3 4 5 6 7 8 9]

array([ 0, -1, 2, -1, 4, -1, 6, -1, 8, -1])

**7.如何重新排列数组？（L1）**

**Q：将一维数组转换成2行的2D数组**

输入：

np.arange(10)

#> array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

期望输出：

#> array([[0, 1, 2, 3, 4],

#> [5, 6, 7, 8, 9]])

**参考答案：**

arr = np.arange(10)

arr.reshape(2, -1) # Setting to -1 automatically decides the number of cols

#> array([[0, 1, 2, 3, 4],

#> [5, 6, 7, 8, 9]])

**8.如何垂直排列两个数组？（L2）**

**Q：垂直排列数组A和B**

输入

a = np.arange(10).reshape(2,-1)

b = np.repeat(1, 10).reshape(2,-1)

期望输出：

#> array([[0, 1, 2, 3, 4],

#> [5, 6, 7, 8, 9],

#> [1, 1, 1, 1, 1],

#> [1, 1, 1, 1, 1]])

**参考答案：**

a = np.arange(10).reshape(2,-1)

b = np.repeat(1, 10).reshape(2,-1)

# Answers

# Method 1:

np.concatenate([a, b], axis=0)

# Method 2:

np.vstack([a, b])

# Method 3:

np.r\_[a, b]

#> array([[0, 1, 2, 3, 4],

#> [5, 6, 7, 8, 9],

#> [1, 1, 1, 1, 1],

#> [1, 1, 1, 1, 1]])

**9.如何水平排列两个数组？（L2）**

**Q：水平排列两数组A和B**

输入

a = np.arange(10).reshape(2,-1)

b = np.repeat(1, 10).reshape(2,-1)

期望输出：

#> array([[0, 1, 2, 3, 4, 1, 1, 1, 1, 1],

#> [5, 6, 7, 8, 9, 1, 1, 1, 1, 1]])

**参考答案：**

a = np.arange(10).reshape(2,-1)

b = np.repeat(1, 10).reshape(2,-1)

# Answers

# Method 1:

np.concatenate([a, b], axis=1)

# Method 2:

np.hstack([a, b])

# Method 3:

np.c\_[a, b]

#> array([[0, 1, 2, 3, 4, 1, 1, 1, 1, 1],

#> [5, 6, 7, 8, 9, 1, 1, 1, 1, 1]])

**10.如何在没有硬编码的情况下在 numpy 中生成自定义序列？（L2）**

**Q：在没有硬编码的情况下创建下面的模式。只使用numpy函数和下面的输入数组A。**

输入：

a = np.array([1,2,3])`

期望输出：

#> array([1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3])

**参考答案：**

np.r\_[np.repeat(a, 3), np.tile(a, 3)]

#> array([1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3])

**11.如何获取两个Python Numpy数组之间的公共项？（L2）**

**Q：获取A和B之间的公共项**

输入：

a = np.array([1,2,3,2,3,4,3,4,5,6])

b = np.array([7,2,10,2,7,4,9,4,9,8])

期望输出：

array([2, 4])

**参考答案：**

a = np.array([1,2,3,2,3,4,3,4,5,6])

b = np.array([7,2,10,2,7,4,9,4,9,8])

np.intersect1d(a,b)

#> array([2, 4])

**12.如何从一个数组中移除那些存在于另一个数组中的项？（L2）**

**Q：从数组A移除数组B中存在的所有项**

输入：

a = np.array([1,2,3,4,5])

b = np.array([5,6,7,8,9])

期望输出：

array([1,2,3,4])

**参考答案：**

a = np.array([1,2,3,4,5])

b = np.array([5,6,7,8,9])

# From 'a' remove all of 'b'

np.setdiff1d(a,b)

#> array([1, 2, 3, 4])

**13.如何获得两个数组元素匹配的位置？（L2）**

**Q：获取A和B元素匹配的位置**

输入：

a = np.array([1,2,3,2,3,4,3,4,5,6])

b = np.array([7,2,10,2,7,4,9,4,9,8])

期望输出：

#> (array([1, 3, 5, 7]),)

**参考答案：**

a = np.array([1,2,3,2,3,4,3,4,5,6])

b = np.array([7,2,10,2,7,4,9,4,9,8])

np.where(a == b)

#> (array([1, 3, 5, 7]),)

**14.如何从一个Numpy数组中提取给定范围内的所有数字？（L2）**

**Q：从A中获取5到10之间的所有项。**

输入：

a = np.array([2, 6, 1, 9, 10, 3, 27])

期望输出：

(array([6, 9, 10]),)

**参考答案：**

a = np.arange(15)

# Method 1

index = np.where((a >= 5) & (a <= 10))

a[index]

# Method 2:

index = np.where(np.logical\_and(a>=5, a<=10))

a[index]

#> (array([6, 9, 10]),)

# Method 3: (thanks loganzk!)

a[(a >= 5) & (a <= 10)]

**15.如何创建一个python函数来处理scalars并在numpy数组上工作？（L2）**

**Q：转换适用于两个标量的函数maxx，以处理两个数组。**

输入：

def maxx(x, y):"""Get the maximum of two items"""if x >= y:

return x

else:

return y

maxx(1, 5)

#> 5

期望输出：

a = np.array([5, 7, 9, 8, 6, 4, 5])

b = np.array([6, 3, 4, 8, 9, 7, 1])

pair\_max(a, b)

#> array([ 6., 7., 9., 8., 9., 7., 5.])

**参考答案：**

def maxx(x, y):"""Get the maximum of two items"""if x >= y:

return x

else:

return y

pair\_max = np.vectorize(maxx, otypes=[float])

a = np.array([5, 7, 9, 8, 6, 4, 5])

b = np.array([6, 3, 4, 8, 9, 7, 1])

pair\_max(a, b)

#> array([ 6., 7., 9., 8., 9., 7., 5.])

**16. 如何交换二维numpy数组中的两列？（L2）**

**Q：在数组arr中交换列1和2。**

arr = np.arange(9).reshape(3,3)

arr

**参考答案：**

# Input

arr = np.arange(9).reshape(3,3)

arr

# Solution

arr[:, [1,0,2]]

#> array([[1, 0, 2],

#> [4, 3, 5],

#> [7, 6, 8]])

**17.如何交换二维numpy数组中的两行？（L2）**

**Q：在数组arr中交换两行1和2**

arr = np.arange(9).reshape(3,3)

arr

**参考答案：**

# Input

arr = np.arange(9).reshape(3,3)

# Solution

arr[[1,0,2], :]

#> array([[3, 4, 5],

#> [0, 1, 2],

#> [6, 7, 8]])

**18.如何反转二维数组的行？（L2）**

**Q：反转二维数组arr的行**

# Input

arr = np.arange(9).reshape(3,3)

**参考答案：**

# Input

arr = np.arange(9).reshape(3,3)

# Solution

arr[::-1]

array([[6, 7, 8],

[3, 4, 5],

[0, 1, 2]])

**19.如何反转二维数组的列？（L2）**

**Q：反转二维数组 arr 的列**

# Input

arr = np.arange(9).reshape(3,3)

**参考答案：**

# Input

arr = np.arange(9).reshape(3,3)

# Solution

arr[:, ::-1]

#> array([[2, 1, 0],

#> [5, 4, 3],

#> [8, 7, 6]])

**20.如何创建包含5到10之间的随机浮点数的二维数组？（L2）**

**Q：创建一个形状为5x3的二维数组，包含5到10之间的随机十进制数。**

**参考答案：**

# Input

arr = np.arange(9).reshape(3,3)

# Solution Method 1:

rand\_arr = np.random.randint(low=5, high=10, size=(5,3)) + np.random.random((5,3))

# print(rand\_arr)

# Solution Method 2:

rand\_arr = np.random.uniform(5,10, size=(5,3))

print(rand\_arr)

#> [[ 8.50061025 9.10531502 6.85867783]

#> [ 9.76262069 9.87717411 7.13466701]

#> [ 7.48966403 8.33409158 6.16808631]

#> [ 7.75010551 9.94535696 5.27373226]

#> [ 8.0850361 5.56165518 7.31244004]]

**21.如何在numpy数组中只打印小数点后三位？（L1）**

**Q：只打印或显示numpy数组rand\_arr的小数点后三位。**

输入：

rand\_arr = np.random.random((5,3))

**参考答案：**

# Input

rand\_arr = np.random.random((5,3))

# Create the random array

rand\_arr = np.random.random([5,3])

# Limit to 3 decimal places

np.set\_printoptions(precision=3)

rand\_arr[:4]

#> array([[ 0.443, 0.109, 0.97 ],

#> [ 0.388, 0.447, 0.191],

#> [ 0.891, 0.474, 0.212],

#> [ 0.609, 0.518, 0.403]])

**22. 如何通过e式科学记数法（如1e10）来打印一个numpy数组？（L1）**

**Q：通过e式科学记数法来打印rand\_arr（如1e10）**

输入：

# Create the random array

np.random.seed(100)

rand\_arr = np.random.random([3,3])/1e3

rand\_arr

#> array([[ 5.434049e-04, 2.783694e-04, 4.245176e-04],

#> [ 8.447761e-04, 4.718856e-06, 1.215691e-04],

#> [ 6.707491e-04, 8.258528e-04, 1.367066e-04]])

期望输出：

#> array([[ 0.000543, 0.000278, 0.000425],

#> [ 0.000845, 0.000005, 0.000122],

#> [ 0.000671, 0.000826, 0.000137]])

**参考答案：**

# Reset printoptions to default

np.set\_printoptions(suppress=False)

# Create the random array

np.random.seed(100)

rand\_arr = np.random.random([3,3])/1e3

rand\_arr

#> array([[ 5.434049e-04, 2.783694e-04, 4.245176e-04],

#> [ 8.447761e-04, 4.718856e-06, 1.215691e-04],

#> [ 6.707491e-04, 8.258528e-04, 1.367066e-04]])

np.set\_printoptions(suppress=True, precision=6) # precision is optional

rand\_arr

#> array([[ 0.000543, 0.000278, 0.000425],

#> [ 0.000845, 0.000005, 0.000122],

#> [ 0.000671, 0.000826, 0.000137]])

**23. 如何限制numpy数组输出中打印的项目数？（L1）**

**Q：将numpy数组a中打印的项数限制为最多6个元素。**

输入：

a = np.arange(15)

# > array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

期望输出：

# > array([ 0, 1, 2, ..., 12, 13, 14])

**参考答案：**

np.set\_printoptions(threshold=6)

a = np.arange(15)

a

# > array([ 0, 1, 2, ..., 12, 13, 14])

**24. 如何打印完整的numpy数组而不截断（L1）**

**Q：打印完整的numpy数组a而不截断。**

输入：

np.set\_printoptions(threshold=6)

a = np.arange(15)

a

# > array([ 0, 1, 2, ..., 12, 13, 14])

期望输出：

a

# > array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

**参考答案：**

# Input

np.set\_printoptions(threshold=6)

a = np.arange(15)

# Solution

np.set\_printoptions(threshold=np.nan)

a

# > array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

**25. 如何导入数字和文本的数据集保持文本在numpy数组中完好无损？（L2）**

**Q：导入鸢尾属植物数据集，保持文本不变。**

**参考答案：**

# Solution

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

# Print the first 3 rows

iris[:3]

# > array([[b'5.1', b'3.5', b'1.4', b'0.2', b'Iris-setosa'],

# > [b'4.9', b'3.0', b'1.4', b'0.2', b'Iris-setosa'],

# > [b'4.7', b'3.2', b'1.3', b'0.2', b'Iris-setosa']], dtype=object)

**26. 如何从1维元组数组中提取特定列？（L2）**

**Q：从前面问题中导入的一维鸢尾属植物数据集中提取文本列的物种。**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None)

**参考答案：**

# \*\*给定：\*\*

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None)

print(iris\_1d.shape)

# Solution:

species = np.array([row[4] for row in iris\_1d])

species[:5]

# > (150,)

# > array([b'Iris-setosa', b'Iris-setosa', b'Iris-setosa', b'Iris-setosa',

# > b'Iris-setosa'],

# > dtype='|S18')

**27. 如何将1维元组数组转换为2维numpy数组？（L2）**

**Q：通过省略鸢尾属植物数据集种类的文本字段，将一维鸢尾属植物数据集转换为二维数组iris\_2d。**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None)

**参考答案：**

# \*\*给定：\*\*

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None)

# Solution:

# Method 1: Convert each row to a list and get the first 4 items

iris\_2d = np.array([row.tolist()[:4] for row in iris\_1d])

iris\_2d[:4]

# Alt Method 2: Import only the first 4 columns from source url

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

iris\_2d[:4]

# > array([[ 5.1, 3.5, 1.4, 0.2],

# > [ 4.9, 3. , 1.4, 0.2],

# > [ 4.7, 3.2, 1.3, 0.2],

# > [ 4.6, 3.1, 1.5, 0.2]])

**28. 如何计算numpy数组的均值，中位数，标准差？（L1）**

**Q：求出鸢尾属植物萼片长度的平均值、中位数和标准差(第1列)**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

# Solution

mu, med, sd = np.mean(sepallength), np.median(sepallength), np.std(sepallength)

print(mu, med, sd)

# > 5.84333333333 5.8 0.825301291785

**29. 如何规范化数组，使数组的值正好介于0和1之间？（L2）**

**Q：创建一种标准化形式的鸢尾属植物间隔长度，其值正好介于0和1之间，这样最小值为0，最大值为1。**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

# Solution

Smax, Smin = sepallength.max(), sepallength.min()

S = (sepallength - Smin)/(Smax - Smin)

# or

S = (sepallength - Smin)/sepallength.ptp() # Thanks, David Ojeda!

print(S)

# > [ 0.222 0.167 0.111 0.083 0.194 0.306 0.083 0.194 0.028 0.167

# > 0.306 0.139 0.139 0. 0.417 0.389 0.306 0.222 0.389 0.222

# > 0.306 0.222 0.083 0.222 0.139 0.194 0.194 0.25 0.25 0.111

# > 0.139 0.306 0.25 0.333 0.167 0.194 0.333 0.167 0.028 0.222

# > 0.194 0.056 0.028 0.194 0.222 0.139 0.222 0.083 0.278 0.194

# > 0.75 0.583 0.722 0.333 0.611 0.389 0.556 0.167 0.639 0.25

# > 0.194 0.444 0.472 0.5 0.361 0.667 0.361 0.417 0.528 0.361

# > 0.444 0.5 0.556 0.5 0.583 0.639 0.694 0.667 0.472 0.389

# > 0.333 0.333 0.417 0.472 0.306 0.472 0.667 0.556 0.361 0.333

# > 0.333 0.5 0.417 0.194 0.361 0.389 0.389 0.528 0.222 0.389

# > 0.556 0.417 0.778 0.556 0.611 0.917 0.167 0.833 0.667 0.806

# > 0.611 0.583 0.694 0.389 0.417 0.583 0.611 0.944 0.944 0.472

# > 0.722 0.361 0.944 0.556 0.667 0.806 0.528 0.5 0.583 0.806

# > 0.861 1. 0.583 0.556 0.5 0.944 0.556 0.583 0.472 0.722

# > 0.667 0.722 0.417 0.694 0.667 0.667 0.556 0.611 0.528 0.444]

**30. 如何计算Softmax得分？（L3）**

**Q：计算sepallength的softmax分数。**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

sepallength = np.array([float(row[0]) for row in iris])

# Solution

def softmax(x):"""Compute softmax values for each sets of scores in x.

https://stackoverflow.com/questions/34968722/how-to-implement-the-softmax-function-in-python"""

e\_x = np.exp(x - np.max(x))

return e\_x / e\_x.sum(axis=0)

print(softmax(sepallength))

# > [ 0.002 0.002 0.001 0.001 0.002 0.003 0.001 0.002 0.001 0.002

# > 0.003 0.002 0.002 0.001 0.004 0.004 0.003 0.002 0.004 0.002

# > 0.003 0.002 0.001 0.002 0.002 0.002 0.002 0.002 0.002 0.001

# > 0.002 0.003 0.002 0.003 0.002 0.002 0.003 0.002 0.001 0.002

# > 0.002 0.001 0.001 0.002 0.002 0.002 0.002 0.001 0.003 0.002

# > 0.015 0.008 0.013 0.003 0.009 0.004 0.007 0.002 0.01 0.002

# > 0.002 0.005 0.005 0.006 0.004 0.011 0.004 0.004 0.007 0.004

# > 0.005 0.006 0.007 0.006 0.008 0.01 0.012 0.011 0.005 0.004

# > 0.003 0.003 0.004 0.005 0.003 0.005 0.011 0.007 0.004 0.003

# > 0.003 0.006 0.004 0.002 0.004 0.004 0.004 0.007 0.002 0.004

# > 0.007 0.004 0.016 0.007 0.009 0.027 0.002 0.02 0.011 0.018

# > 0.009 0.008 0.012 0.004 0.004 0.008 0.009 0.03 0.03 0.005

# > 0.013 0.004 0.03 0.007 0.011 0.018 0.007 0.006 0.008 0.018

# > 0.022 0.037 0.008 0.007 0.006 0.03 0.007 0.008 0.005 0.013

# > 0.011 0.013 0.004 0.012 0.011 0.011 0.007 0.009 0.007 0.005]

**31. 如何找到numpy数组的百分位数？（L1）**

**Q：找到鸢尾属植物数据集的第5和第95百分位数**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

# Solution

np.percentile(sepallength, q=[5, 95])

# > array([ 4.6 , 7.255])

**32. 如何在数组中的随机位置插入值？（L2）**

**Q：在iris\_2d数据集中的20个随机位置插入np.nan值**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='object')

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='object')

# Method 1

i, j = np.where(iris\_2d)

# i, j contain the row numbers and column numbers of 600 elements of iris\_x

np.random.seed(100)

iris\_2d[np.random.choice((i), 20), np.random.choice((j), 20)] = np.nan

# Method 2

np.random.seed(100)

iris\_2d[np.random.randint(150, size=20), np.random.randint(4, size=20)] = np.nan

# Print first 10 rows

print(iris\_2d[:10])

# > [[b'5.1' b'3.5' b'1.4' b'0.2' b'Iris-setosa']

# > [b'4.9' b'3.0' b'1.4' b'0.2' b'Iris-setosa']

# > [b'4.7' b'3.2' b'1.3' b'0.2' b'Iris-setosa']

# > [b'4.6' b'3.1' b'1.5' b'0.2' b'Iris-setosa']

# > [b'5.0' b'3.6' b'1.4' b'0.2' b'Iris-setosa']

# > [b'5.4' b'3.9' b'1.7' b'0.4' b'Iris-setosa']

# > [b'4.6' b'3.4' b'1.4' b'0.3' b'Iris-setosa']

# > [b'5.0' b'3.4' b'1.5' b'0.2' b'Iris-setosa']

# > [b'4.4' nan b'1.4' b'0.2' b'Iris-setosa']

# > [b'4.9' b'3.1' b'1.5' b'0.1' b'Iris-setosa']]

**33. 如何在numpy数组中找到缺失值的位置？（L2）**

**Q：在iris\_2d的sepallength中查找缺失值的数量和位置（第1列）**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float')

iris\_2d[np.random.randint(150, size=20), np.random.randint(4, size=20)] = np.nan

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

iris\_2d[np.random.randint(150, size=20), np.random.randint(4, size=20)] = np.nan

# Solution

print("Number of missing values: \n", np.isnan(iris\_2d[:, 0]).sum())

print("Position of missing values: \n", np.where(np.isnan(iris\_2d[:, 0])))

# > Number of missing values:

# > 5

# > Position of missing values:

# > (array([ 39, 88, 99, 130, 147]),)

**34. 如何根据两个或多个条件过滤numpy数组？（L3）**

**Q：过滤具有petallength（第3列）> 1.5 和 sepallength（第1列）< 5.0 的iris\_2d行**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

# Solution

condition = (iris\_2d[:, 2] > 1.5) & (iris\_2d[:, 0] < 5.0)

iris\_2d[condition]

# > array([[ 4.8, 3.4, 1.6, 0.2],

# > [ 4.8, 3.4, 1.9, 0.2],

# > [ 4.7, 3.2, 1.6, 0.2],

# > [ 4.8, 3.1, 1.6, 0.2],

# > [ 4.9, 2.4, 3.3, 1. ],

# > [ 4.9, 2.5, 4.5, 1.7]])

**35. 如何从numpy数组中删除包含缺失值的行？（L3）**

**Q：选择没有任何nan值的iris\_2d行。**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

iris\_2d[np.random.randint(150, size=20), np.random.randint(4, size=20)] = np.nan

# Solution

# No direct numpy function for this.

# Method 1:

any\_nan\_in\_row = np.array([~np.any(np.isnan(row)) for row in iris\_2d])

iris\_2d[any\_nan\_in\_row][:5]

# Method 2: (By Rong)

iris\_2d[np.sum(np.isnan(iris\_2d), axis = 1) == 0][:5]

# > array([[ 4.9, 3. , 1.4, 0.2],

# > [ 4.7, 3.2, 1.3, 0.2],

# > [ 4.6, 3.1, 1.5, 0.2],

# > [ 5. , 3.6, 1.4, 0.2],

# > [ 5.4, 3.9, 1.7, 0.4]])

**36. 如何找到numpy数组的两列之间的相关性？（L2）**

**问题：在iris\_2d中找出SepalLength（第1列）和PetalLength（第3列）之间的相关性**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

# Solution 1

np.corrcoef(iris[:, 0], iris[:, 2])[0, 1]

# Solution 2

from scipy.stats.stats import pearsonr

corr, p\_value = pearsonr(iris[:, 0], iris[:, 2])

print(corr)

# Correlation coef indicates the degree of linear relationship between two numeric variables.

# It can range between -1 to +1.

# The p-value roughly indicates the probability of an uncorrelated system producing

# datasets that have a correlation at least as extreme as the one computed.

# The lower the p-value (<0.01), stronger is the significance of the relationship.

# It is not an indicator of the strength.

# > 0.871754157305

**37. 如何查找给定数组是否具有任何空值？（L2）**

**Q：找出iris\_2d是否有任何缺失值。**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

np.isnan(iris\_2d).any()

# > False

**38. 如何在numpy数组中用0替换所有缺失值？（L2）**

**Q：在numpy数组中将所有出现的nan替换为0**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

iris\_2d[np.random.randint(150, size=20), np.random.randint(4, size=20)] = np.nan

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])

iris\_2d[np.random.randint(150, size=20), np.random.randint(4, size=20)] = np.nan

# Solution

iris\_2d[np.isnan(iris\_2d)] = 0

iris\_2d[:4]

# > array([[ 5.1, 3.5, 1.4, 0. ],

# > [ 4.9, 3. , 1.4, 0.2],

# > [ 4.7, 3.2, 1.3, 0.2],

# > [ 4.6, 3.1, 1.5, 0.2]])

**39. 如何在numpy数组中查找唯一值的计数？（L2）**

**Q：找出鸢尾属植物物种中的独特值和独特值的数量**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

**参考答案：**

# Import iris keeping the text column intact

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

# Solution

# Extract the species column as an array

species = np.array([row.tolist()[4] for row in iris])

# Get the unique values and the counts

np.unique(species, return\_counts=True)

# > (array([b'Iris-setosa', b'Iris-versicolor', b'Iris-virginica'],

# > dtype='|S15'), array([50, 50, 50]))

**40. 如何将数字转换为分类（文本）数组？（L2）**

**Q：将iris\_2d的花瓣长度（第3列）加入以形成文本数组，这样如果花瓣长度为：**

* Less than 3 --> 'small'
* 3-5 --> 'medium'
* '>=5 --> 'large'

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

# Bin petallength

petal\_length\_bin = np.digitize(iris[:, 2].astype('float'), [0, 3, 5, 10])

# Map it to respective category

label\_map = {1: 'small', 2: 'medium', 3: 'large', 4: np.nan}

petal\_length\_cat = [label\_map[x] for x in petal\_length\_bin]

# View

petal\_length\_cat[:4]

<# > ['small', 'small', 'small', 'small']

**41. 如何从numpy数组的现有列创建新列？（L2）**

**Q：在iris\_2d中为卷创建一个新列，其中volume是**（pi x petallength x sepal\_length ^ 2）/ 3

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris\_2d = np.genfromtxt(url, delimiter=',', dtype='object')

# Solution

# Compute volume

sepallength = iris\_2d[:, 0].astype('float')

petallength = iris\_2d[:, 2].astype('float')

volume = (np.pi \* petallength \* (sepallength\*\*2))/3

# Introduce new dimension to match iris\_2d's

volume = volume[:, np.newaxis]

# Add the new column

out = np.hstack([iris\_2d, volume])

# View

out[:4]

# > array([[b'5.1', b'3.5', b'1.4', b'0.2', b'Iris-setosa', 38.13265162927291],

# > [b'4.9', b'3.0', b'1.4', b'0.2', b'Iris-setosa', 35.200498485922445],

# > [b'4.7', b'3.2', b'1.3', b'0.2', b'Iris-setosa', 30.0723720777127],

# > [b'4.6', b'3.1', b'1.5', b'0.2', b'Iris-setosa', 33.238050274980004]], dtype=object)

**42. 如何在numpy中进行概率抽样？（L3）**

**Q：**随机抽鸢尾属植物的种类，使得刚毛的数量是云芝和维吉尼亚的两倍

输入：

# Import iris keeping the text column intact

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

**参考答案：**

# Import iris keeping the text column intact

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

# Solution

# Get the species column

species = iris[:, 4]

# Approach 1: Generate Probablistically

np.random.seed(100)

a = np.array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'])

species\_out = np.random.choice(a, 150, p=[0.5, 0.25, 0.25])

# Approach 2: Probablistic Sampling (preferred)

np.random.seed(100)

probs = np.r\_[np.linspace(0, 0.500, num=50), np.linspace(0.501, .750, num=50), np.linspace(.751, 1.0, num=50)]

index = np.searchsorted(probs, np.random.random(150))

species\_out = species[index]

print(np.unique(species\_out, return\_counts=True))

# > (array([b'Iris-setosa', b'Iris-versicolor', b'Iris-virginica'], dtype=object), array([77, 37, 36]))

方法2是首选方法，因为它创建了一个索引变量，该变量可用于取样2维表格数据。

**43. 如何在按另一个数组分组时获取数组的第二大值？（L2）**

**Q：第二长的物种setosa的价值是多少**

输入：

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

**参考答案：**

# Import iris keeping the text column intact

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

# Solution

# Get the species and petal length columns

petal\_len\_setosa = iris[iris[:, 4] == b'Iris-setosa', [2]].astype('float')

# Get the second last value

np.unique(np.sort(petal\_len\_setosa))[-2]

# > 1.7

**44. 如何按列对2D数组进行排序（L2）**

**Q：**根据sepallength列对虹膜数据集进行排序。

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

**参考答案：**

# Sort by column position 0: SepalLength

print(iris[iris[:,0].argsort()][:20])

# > [[b'4.3' b'3.0' b'1.1' b'0.1' b'Iris-setosa']

# > [b'4.4' b'3.2' b'1.3' b'0.2' b'Iris-setosa']

# > [b'4.4' b'3.0' b'1.3' b'0.2' b'Iris-setosa']

# > [b'4.4' b'2.9' b'1.4' b'0.2' b'Iris-setosa']

# > [b'4.5' b'2.3' b'1.3' b'0.3' b'Iris-setosa']

# > [b'4.6' b'3.6' b'1.0' b'0.2' b'Iris-setosa']

# > [b'4.6' b'3.1' b'1.5' b'0.2' b'Iris-setosa']

# > [b'4.6' b'3.4' b'1.4' b'0.3' b'Iris-setosa']

# > [b'4.6' b'3.2' b'1.4' b'0.2' b'Iris-setosa']

# > [b'4.7' b'3.2' b'1.3' b'0.2' b'Iris-setosa']

# > [b'4.7' b'3.2' b'1.6' b'0.2' b'Iris-setosa']

# > [b'4.8' b'3.0' b'1.4' b'0.1' b'Iris-setosa']

# > [b'4.8' b'3.0' b'1.4' b'0.3' b'Iris-setosa']

# > [b'4.8' b'3.4' b'1.9' b'0.2' b'Iris-setosa']

# > [b'4.8' b'3.4' b'1.6' b'0.2' b'Iris-setosa']

# > [b'4.8' b'3.1' b'1.6' b'0.2' b'Iris-setosa']

# > [b'4.9' b'2.4' b'3.3' b'1.0' b'Iris-versicolor']

# > [b'4.9' b'2.5' b'4.5' b'1.7' b'Iris-virginica']

# > [b'4.9' b'3.1' b'1.5' b'0.1' b'Iris-setosa']

# > [b'4.9' b'3.1' b'1.5' b'0.1' b'Iris-setosa']]

**45. 如何在numpy数组中找到最常见的值？（L1）**

**Q：在鸢尾属植物数据集中找到最常见的花瓣长度值（第3列）。**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

**参考答案：**

# Input:

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

# Solution:

vals, counts = np.unique(iris[:, 2], return\_counts=True)

print(vals[np.argmax(counts)])

# > b'1.5'

**46. 如何找到第一次出现的值大于给定值的位置？（L2）**

**Q：在虹膜数据集的petalwidth第4列中查找第一次出现的值大于1.0的位置。**

# Input:

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

**参考答案：**

# Input:

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

# Solution: (edit: changed argmax to argwhere. Thanks Rong!)

np.argwhere(iris[:, 3].astype(float) > 1.0)[0]

# > 50

**47. 如何将大于给定值的所有值替换为给定的截止值？（L2）**

**问题：**从数组a中，替换所有大于30到30和小于10到10的值。

**给定：**

np.random.seed(100)

a = np.random.uniform(1,50, 20)

**答案：**

# Input

np.set\_printoptions(precision=2)

np.random.seed(100)

a = np.random.uniform(1,50, 20)

# Solution 1: Using np.clip

np.clip(a, a\_min=10, a\_max=30)

# Solution 2: Using np.where

print(np.where(a < 10, 10, np.where(a > 30, 30, a)))

# > [ 27.63 14.64 21.8 30. 10. 10. 30. 30. 10. 29.18 30.

# > 11.25 10.08 10. 11.77 30. 30. 10. 30. 14.43]

**48. 如何从numpy数组中获取最大n值的位置？（L2）**

**Q：获取给定数组a中前5个最大值的位置。**

np.random.seed(100)

a = np.random.uniform(1,50, 20)

**参考答案：**

# Input

np.random.seed(100)

a = np.random.uniform(1,50, 20)

# Solution:

print(a.argsort())

# > [18 7 3 10 15]

# Solution 2:

np.argpartition(-a, 5)[:5]

# > [15 10 3 7 18]

# Below methods will get you the values.

# Method 1:

a[a.argsort()][-5:]

# Method 2:

np.sort(a)[-5:]

# Method 3:

np.partition(a, kth=-5)[-5:]

# Method 4:

a[np.argpartition(-a, 5)][:5]

**49. 如何计算数组中所有可能值的行数？L4（）**

**Q：按行计算唯一值的计数。**

输入：

np.random.seed(100)

arr = np.random.randint(1,11,size=(6, 10))

arr

> array([[ 9, 9, 4, 8, 8, 1, 5, 3, 6, 3],

> [ 3, 3, 2, 1, 9, 5, 1, 10, 7, 3],

> [ 5, 2, 6, 4, 5, 5, 4, 8, 2, 2],

> [ 8, 8, 1, 3, 10, 10, 4, 3, 6, 9],

> [ 2, 1, 8, 7, 3, 1, 9, 3, 6, 2],

> [ 9, 2, 6, 5, 3, 9, 4, 6, 1, 10]])

期望输出：

> [[1, 0, 2, 1, 1, 1, 0, 2, 2, 0],

> [2, 1, 3, 0, 1, 0, 1, 0, 1, 1],

> [0, 3, 0, 2, 3, 1, 0, 1, 0, 0],

> [1, 0, 2, 1, 0, 1, 0, 2, 1, 2],

> [2, 2, 2, 0, 0, 1, 1, 1, 1, 0],

> [1, 1, 1, 1, 1, 2, 0, 0, 2, 1]]

输出包含10列，表示从1到10的数字。这些值是各行中数字的计数。

例如，cell(0，2)的值为2，这意味着数字3在第一行中恰好出现了2次。

**参考答案：**

# Input:

np.random.seed(100)

arr = np.random.randint(1,11,size=(6, 10))

arr

# > array([[ 9, 9, 4, 8, 8, 1, 5, 3, 6, 3],

# > [ 3, 3, 2, 1, 9, 5, 1, 10, 7, 3],

# > [ 5, 2, 6, 4, 5, 5, 4, 8, 2, 2],

# > [ 8, 8, 1, 3, 10, 10, 4, 3, 6, 9],

# > [ 2, 1, 8, 7, 3, 1, 9, 3, 6, 2],

# > [ 9, 2, 6, 5, 3, 9, 4, 6, 1, 10]])

# Solution

def counts\_of\_all\_values\_rowwise(arr2d):

# Unique values and its counts row wise

num\_counts\_array = [np.unique(row, return\_counts=True) for row in arr2d]

# Counts of all values row wise

return([[int(b[a==i]) if i in a else 0 for i in np.unique(arr2d)] for a, b in num\_counts\_array])

# Print

print(np.arange(1,11))

counts\_of\_all\_values\_rowwise(arr)

# > [ 1 2 3 4 5 6 7 8 9 10]

# > [[1, 0, 2, 1, 1, 1, 0, 2, 2, 0],

# > [2, 1, 3, 0, 1, 0, 1, 0, 1, 1],

# > [0, 3, 0, 2, 3, 1, 0, 1, 0, 0],

# > [1, 0, 2, 1, 0, 1, 0, 2, 1, 2],

# > [2, 2, 2, 0, 0, 1, 1, 1, 1, 0],

# > [1, 1, 1, 1, 1, 2, 0, 0, 2, 1]]

# Example 2:

arr = np.array([np.array(list('bill clinton')), np.array(list('narendramodi')), np.array(list('jjayalalitha'))])

print(np.unique(arr))

counts\_of\_all\_values\_rowwise(arr)

# > [' ' 'a' 'b' 'c' 'd' 'e' 'h' 'i' 'j' 'l' 'm' 'n' 'o' 'r' 't' 'y']

# > [[1, 0, 1, 1, 0, 0, 0, 2, 0, 3, 0, 2, 1, 0, 1, 0],

# > [0, 2, 0, 0, 2, 1, 0, 1, 0, 0, 1, 2, 1, 2, 0, 0],

# > [0, 4, 0, 0, 0, 0, 1, 1, 2, 2, 0, 0, 0, 0, 1, 1]]

**50. 如何将数组转换为平面一维数组？(L2)**

**Q：将array\_of\_arrays转换为扁平线性1d数组。**

输入：

# Input:

arr1 = np.arange(3)

arr2 = np.arange(3,7)

arr3 = np.arange(7,10)

array\_of\_arrays = np.array([arr1, arr2, arr3])

array\_of\_arrays

# > array([array([0, 1, 2]), array([3, 4, 5, 6]), array([7, 8, 9])], dtype=object)

期望输出：

# > array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

**参考答案：**

# Input:

arr1 = np.arange(3)

arr2 = np.arange(3,7)

arr3 = np.arange(7,10)

array\_of\_arrays = np.array([arr1, arr2, arr3])

print('array\_of\_arrays: ', array\_of\_arrays)

# Solution 1

arr\_2d = np.array([a for arr in array\_of\_arrays for a in arr])

# Solution 2:

arr\_2d = np.concatenate(array\_of\_arrays)

print(arr\_2d)

# > array\_of\_arrays: [array([0, 1, 2]) array([3, 4, 5, 6]) array([7, 8, 9])]

# > [0 1 2 3 4 5 6 7 8 9]

**51. 如何在numpy中为数组生成单热编码？（L4）**

**Q：计算一次性编码(数组中每个唯一值的虚拟二进制变量)**

输入：

np.random.seed(101)

arr = np.random.randint(1,4, size=6)

arr

# > array([2, 3, 2, 2, 2, 1])

期望输出：

# > array([[ 0., 1., 0.],

# > [ 0., 0., 1.],

# > [ 0., 1., 0.],

# > [ 0., 1., 0.],

# > [ 0., 1., 0.],

# > [ 1., 0., 0.]])

**参考答案：**

# Input:

np.random.seed(101)

arr = np.random.randint(1,4, size=6)

arr

# > array([2, 3, 2, 2, 2, 1])

# Solution:

def one\_hot\_encodings(arr):

uniqs = np.unique(arr)

out = np.zeros((arr.shape[0], uniqs.shape[0]))

for i, k in enumerate(arr):

out[i, k-1] = 1

return out

one\_hot\_encodings(arr)

# > array([[ 0., 1., 0.],

# > [ 0., 0., 1.],

# > [ 0., 1., 0.],

# > [ 0., 1., 0.],

# > [ 0., 1., 0.],

# > [ 1., 0., 0.]])

# Method 2:

(arr[:, None] == np.unique(arr)).view(np.int8)

**52. 如何创建按分类变量分组的行号？（L3）**

**Q：创建按分类变量分组的行号。使用以下来自鸢尾属植物物种的样本作为输入。**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

species = np.genfromtxt(url, delimiter=',', dtype='str', usecols=4)

species\_small = np.sort(np.random.choice(species, size=20))

species\_small

# > array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',

# > 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',

# > 'Iris-virginica', 'Iris-virginica', 'Iris-virginica',

# > 'Iris-virginica', 'Iris-virginica', 'Iris-virginica'],

# > dtype='<U15')

期望输出：

# > [0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 6, 7]

**参考答案：**

# Input:

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

species = np.genfromtxt(url, delimiter=',', dtype='str', usecols=4)

np.random.seed(100)

species\_small = np.sort(np.random.choice(species, size=20))

species\_small

# > array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',

# > 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',

# > 'Iris-virginica', 'Iris-virginica', 'Iris-virginica',

# > 'Iris-virginica'],

# > dtype='<U15')

print([i for val in np.unique(species\_small) for i, grp in enumerate(species\_small[species\_small==val])])

[0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 5, 6, 7, 8, 0, 1, 2, 3, 4, 5]

**53. 如何根据给定的分类变量创建组ID？(L4)**

**Q：根据给定的分类变量创建组ID。使用以下来自鸢尾属植物物种的样本作为输入。**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

species = np.genfromtxt(url, delimiter=',', dtype='str', usecols=4)

species\_small = np.sort(np.random.choice(species, size=20))

species\_small

# > array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',

# > 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',

# > 'Iris-virginica', 'Iris-virginica', 'Iris-virginica',

# > 'Iris-virginica', 'Iris-virginica', 'Iris-virginica'],

# > dtype='<U15')

期望输出：

# > [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2]

**参考答案：**

# Input:

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

species = np.genfromtxt(url, delimiter=',', dtype='str', usecols=4)

np.random.seed(100)

species\_small = np.sort(np.random.choice(species, size=20))

species\_small

# > array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',

# > 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',

# > 'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',

# > 'Iris-virginica', 'Iris-virginica', 'Iris-virginica',

# > 'Iris-virginica'],

# > dtype='<U15')

# Solution:

output = [np.argwhere(np.unique(species\_small) == s).tolist()[0][0] for val in np.unique(species\_small) for s in species\_small[species\_small==val]]

# Solution: For Loop version

output = []

uniqs = np.unique(species\_small)

for val in uniqs: # uniq values in groupfor s in species\_small[species\_small==val]: # each element in group

groupid = np.argwhere(uniqs == s).tolist()[0][0] # groupid

output.append(groupid)

print(output)

# > [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2]

**54. 如何使用numpy对数组中的项进行排名？（L2）**

**Q：为给定的数字数组a创建排名。**

输入：

np.random.seed(10)

a = np.random.randint(20, size=10)

print(a)

# > [ 9 4 15 0 17 16 17 8 9 0]

期望输出：

[4 2 6 0 8 7 9 3 5 1]

**参考答案：**

np.random.seed(10)

a = np.random.randint(20, size=10)

print('Array: ', a)

# Solution

print(a.argsort().argsort())

print('Array: ', a)

# > Array: [ 9 4 15 0 17 16 17 8 9 0]

# > [4 2 6 0 8 7 9 3 5 1]

# > Array: [ 9 4 15 0 17 16 17 8 9 0]

**55. 如何使用numpy对多维数组中的项进行排名？（L3）**

**Q：创建与给定数字数组a相同形状的排名数组。**

输入：

np.random.seed(10)

a = np.random.randint(20, size=[2,5])

print(a)

# > [[ 9 4 15 0 17]

# > [16 17 8 9 0]]

期望输出：

# > [[4 2 6 0 8]

# > [7 9 3 5 1]]

**参考答案：**

# Input:

np.random.seed(10)

a = np.random.randint(20, size=[2,5])

print(a)

# Solution

print(a.ravel().argsort().argsort().reshape(a.shape))

# > [[ 9 4 15 0 17]

# > [16 17 8 9 0]]

# > [[4 2 6 0 8]

# > [7 9 3 5 1]]

**56. 如何在二维numpy数组的每一行中找到最大值？（L2）**

**Q：计算给定数组中每行的最大值。**

输入：

np.random.seed(100)

a = np.random.randint(1,10, [5,3])

a

# > array([[9, 9, 4],

# > [8, 8, 1],

# > [5, 3, 6],

# > [3, 3, 3],

# > [2, 1, 9]])

**参考答案：**

# Input

np.random.seed(100)

a = np.random.randint(1,10, [5,3])

a

# Solution 1

np.amax(a, axis=1)

# Solution 2

np.apply\_along\_axis(np.max, arr=a, axis=1)

# > array([9, 8, 6, 3, 9])

**57. 如何计算二维numpy数组每行的最小值？（L3）**

**Q：为给定的二维numpy数组计算每行的最小值。**

输入：

np.random.seed(100)

a = np.random.randint(1,10, [5,3])

a

# > array([[9, 9, 4],

# > [8, 8, 1],

# > [5, 3, 6],

# > [3, 3, 3],

# > [2, 1, 9]])

**参考答案：**

# Input

np.random.seed(100)

a = np.random.randint(1,10, [5,3])

a

# Solution

np.apply\_along\_axis(lambda x: np.min(x)/np.max(x), arr=a, axis=1)

# > array([ 0.44444444, 0.125 , 0.5 , 1. , 0.11111111])

**58. 如何在numpy数组中找到重复的记录？（L3）**

**Q：在给定的numpy数组中找到重复的条目(第二次出现以后)，并将它们标记为True。第一次出现应该是False的。**

输入：

# Input

np.random.seed(100)

a = np.random.randint(0, 5, 10)

print('Array: ', a)

# > Array: [0 0 3 0 2 4 2 2 2 2]

期望输出：

# > [False True False True False False True True True True]

**参考答案：**

# Input

np.random.seed(100)

a = np.random.randint(0, 5, 10)

## Solution

# There is no direct function to do this as of 1.13.3

# Create an all True array

out = np.full(a.shape[0], True)

# Find the index positions of unique elements

unique\_positions = np.unique(a, return\_index=True)[1]

# Mark those positions as False

out[unique\_positions] = False

print(out)

# > [False True False True False False True True True True]

**59. 如何找出数字的分组均值？（L3）**

**Q：在二维数字数组中查找按分类列分组的数值列的平均值**

输入：

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

期望输出：

# > [[b'Iris-setosa', 3.418],

# > [b'Iris-versicolor', 2.770],

# > [b'Iris-virginica', 2.974]]

**参考答案：**

# Input

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = np.genfromtxt(url, delimiter=',', dtype='object')

names = ('sepallength', 'sepalwidth', 'petallength', 'petalwidth', 'species')

# Solution

# No direct way to implement this. Just a version of a workaround.

numeric\_column = iris[:, 1].astype('float') # sepalwidth

grouping\_column = iris[:, 4] # species

# List comprehension version

[[group\_val, numeric\_column[grouping\_column==group\_val].mean()] for group\_val in np.unique(grouping\_column)]

# For Loop version

output = []

for group\_val in np.unique(grouping\_column):

output.append([group\_val, numeric\_column[grouping\_column==group\_val].mean()])

output

# > [[b'Iris-setosa', 3.418],

# > [b'Iris-versicolor', 2.770],

# > [b'Iris-virginica', 2.974]]

**60. 如何将PIL图像转换为numpy数组？（L3）**

**Q：从以下URL导入图像并将其转换为numpy数组。**

URL = 'https://upload.wikimedia.org/wikipedia/commons/8/8b/Denali\_Mt\_McKinley.jpg'

**参考答案：**

from io import BytesIO

from PIL import Image

import PIL, requests

# Import image from URL

URL = 'https://upload.wikimedia.org/wikipedia/commons/8/8b/Denali\_Mt\_McKinley.jpg'

response = requests.get(URL)

# Read it as Image

I = Image.open(BytesIO(response.content))

# Optionally resize

I = I.resize([150,150])

# Convert to numpy array

arr = np.asarray(I)

# Optionaly Convert it back to an image and show

im = PIL.Image.fromarray(np.uint8(arr))

Image.Image.show(im)

**61. 如何删除numpy数组中所有缺少的值？（L2）**

**Q：从一维numpy数组中删除所有NaN值**

输入：

np.array([1,2,3,np.nan,5,6,7,np.nan])

期望输出：

array([ 1., 2., 3., 5., 6., 7.])

**参考答案：**

a = np.array([1,2,3,np.nan,5,6,7,np.nan])

a[~np.isnan(a)]

# > array([ 1., 2., 3., 5., 6., 7.])

**62. 如何计算两个数组之间的欧氏距离？（L3）**

**Q：计算两个数组a和数组b之间的欧氏距离。**

输入：

a = np.array([1,2,3,4,5])

b = np.array([4,5,6,7,8])

**参考答案：**

# Input

a = np.array([1,2,3,4,5])

b = np.array([4,5,6,7,8])

# Solution

dist = np.linalg.norm(a-b)

dist

# > 6.7082039324993694

**63. 如何在一维数组中找到所有的局部极大值(或峰值)？（练习）**

**Q：找到一个一维数字数组a中的所有峰值。峰顶是两边被较小数值包围的点。**

输入：

a = np.array([1, 3, 7, 1, 2, 6, 0, 1])

期望输出：

# > array([2, 5])

其中，2和5是峰值7和6的位置。

**64. 如何从二维数组中减去一维数组，其中一维数组的每一项从各自的行中减去？（练习）**

**Q：从2d数组a\_2d中减去一维数组b\_1D，使得b\_1D的每一项从a\_2d的相应行中减去。**

a\_2d = np.array([[3,3,3],[4,4,4],[5,5,5]])

b\_1d = np.array([1,1,1]

期望输出：

# > [[2 2 2]

# > [2 2 2]

# > [2 2 2]]

**65. 如何查找数组中项的第n次重复索引？（练习）**

**Q：找出x中数字1的第5次重复的索引。**

x = np.array([1, 2, 1, 1, 3, 4, 3, 1, 1, 2, 1, 1, 2])

期望输出：8

**66. 如何将numpy的datetime 64对象转换为datetime的datetime对象？（L2）**

**Q：将numpy的datetime64对象转换为datetime的datetime对象**

# Input: a numpy datetime64 object

dt64 = np.datetime64('2018-02-25 22:10:10')

**参考答案：**

# Input: a numpy datetime64 object

dt64 = np.datetime64('2018-02-25 22:10:10')

# Solution

from datetime import datetime

dt64.tolist()

# or

dt64.astype(datetime)

# > datetime.datetime(2018, 2, 25, 22, 10, 10)

**67. 如何计算numpy数组的移动平均值？（L3）**

**Q：对于给定的一维数组，计算窗口大小为3的移动平均值**。

输入：

np.random.seed(100)

Z = np.random.randint(10, size=10)

**参考答案：**

def moving\_average(a, n=3) :

ret = np.cumsum(a, dtype=float)

ret[n:] = ret[n:] - ret[:-n]

return ret[n - 1:] / n

np.random.seed(100)

Z = np.random.randint(10, size=10)

print('array: ', Z)

# Method 1

moving\_average(Z, n=3).round(2)

# Method 2: # Thanks AlanLRH!

# np.ones(3)/3 gives equal weights. Use np.ones(4)/4 for window size 4.

np.convolve(Z, np.ones(3)/3, mode='valid') .

# > array: [8 8 3 7 7 0 4 2 5 2]

# > moving average: [ 6.33 6. 5.67 4.67 3.67 2. 3.67 3. ]

**68. 如何在给定起始点、长度和步骤的情况下创建一个numpy数组序列？（练习）**

**Q：创建长度为10的numpy数组，从5开始，在连续的数字之间的步长为3。**

期望输出：

array([ 5, 8, 11, 14, 17, 20, 23, 26, 29, 32])

**69. 如何从给定的一维数组创建步长？（练习）**

**Q：从给定的一维数组arr中，利用步进生成一个二维矩阵，窗口长度为4，步距为2，类似于 [[0,1,2,3], [2,3,4,5], [4,5,6,7]..]**

输入：

arr = np.arange(15)

arr

# > array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14])

期望输出：

# > [[ 0 1 2 3]

# > [ 2 3 4 5]

# > [ 4 5 6 7]

# > [ 6 7 8 9]

# > [ 8 9 10 11]

# > [10 11 12 13]]