**Python - *numpy* basics exercises**

<https://www.machinelearningplus.com/python/101-numpy-exercises-python>

# 1. Import numpy as np and see the version

# Import numpy as `np` and print the version number.

import numpy as np

np.\_\_version\_\_

# 2. How to create a 1D array?

# Create a 1D array of numbers from 0 to 9

# np.array(range(0,10))

np.arange(10)

# 3. How to create a boolean array?

# Create a 3×3 numpy array of all True’s

np.full(shape=(3,3), fill\_value=True, dtype=bool)

# 4. How to extract items that satisfy a given condition from 1D array?

# Extract all odd numbers from arr

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

arr[arr % 2 == 1]

# 5. How to replace items that satisfy a condition with another value in numpy array?

# Replace all odd numbers in arr with -1

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

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

arr

# 6. How to replace items that satisfy a condition without affecting the original array?

# Replace all odd numbers in arr with -1 without changing arr

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

arr2 = np.where(arr % 2 == 1, -1, arr)

arr

arr2

# 7. How to reshape an array?

# Convert a 1D array to a 2D array with 2 rows

np.arange(10)

np.reshape(2, 5)

# 8. How to stack two arrays vertically?

# Stack arrays a and b vertically

a = np.arange(10)

b = np.repeat(1, 10)

# version 1:

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

# version 2:

np.vstack([a,b])

# 9. How to stack two arrays horizontally?

# Stack the arrays a and b horizontally.

# version 1:

np.concatenate([a,b])

# version 2:

np.hstack([a,b])

# 10. How to generate custom sequences in numpy without hardcoding?

# Create the following pattern without hardcoding. Use only numpy functions and the below input array a.

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

#desired output: array([1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3])

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

# np.tile(A, reps) -> construct an array by repeating "reps" times "A"

np.tile(a,3)

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

# 11. How to get the common items between two python numpy arrays?

# Get the common items between a and b

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)

# 12. How to remove from one array those items that exist in another?

# From array a remove all items present in array b

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

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

np.setdiff1d(a,b)

# 13. How to get the positions where elements of two arrays match?

# Get the positions where elements of a and b match

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)

# to get the values: a[np.where(a==b)]

# 14. How to extract all numbers between a given range from a numpy array?

# Get all items between 5 and 10 from a.

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

# version 1

a[np.where((a >= 5) & (a <= 10))] ### ! parentheses around (a..5) and (a..10) obligatory !

# version 2

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

# 16. How to swap two columns in a 2d numpy array?

# Swap columns 1 and 2 in the array arr.

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

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

# 17. How to swap two rows in a 2d numpy array?

# Swap rows 1 and 2 in the array arr:

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

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

# 18. How to reverse the rows of a 2D array?

# Reverse the rows of a 2D array arr.

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

arr[::-1,:] # ! no brackets here !

# same as

arr[::-1]

arr[::-1,]

# 19. How to reverse the columns of a 2D array?

# Reverse the columns of a 2D array arr.

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

arr[:,::-1]

# 21. How to print only 3 decimal places in python numpy array?

# Print or show only 3 decimal places of the numpy array rand\_arr.

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

np.set\_printoptions(precision=3)

rand\_arr[:4]

# 22. How to pretty print a numpy array by suppressing the scientific notation (like 1e10)?

# Pretty print rand\_arr by suppressing the scientific notation (like 1e10)

np.random.seed(100)

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

# Reset printoptions to default

np.set\_printoptions(suppress=False)

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

# 23. How to limit the number of items printed in output of numpy array?

# Limit the number of items printed in python numpy array a to a maximum of 6 elements.

a = np.arange(15)

np.set\_printoptions(threshold=6)

# 24. How to print the full numpy array without truncating

# Print the full numpy array a without truncating.

np.set\_printoptions(threshold=6)

a = np.arange(15)

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

# 25. How to import a dataset with numbers and texts keeping the text intact in python numpy?

# Import the iris dataset keeping the text intact.

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

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

# => to retain the species, a text field , set dtype to object

# if set dtype=None, 1d array of tuples would have been returned.

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

# Print the first 3 rows

iris[:3]

# 26. How to extract a particular column from 1D array of tuples?

# Extract the text column species from the 1D iris imported in previous question.

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)

# (150,)

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

species[:5]

# 27. How to convert a 1d array of tuples to a 2d numpy array?

# Convert the 1D iris to 2D array iris\_2d by omitting the species text field.

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

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

# version 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]

# version 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]

# 28. How to compute the mean, median, standard deviation of a numpy array?

# Find the mean, median, standard deviation of iris's sepallength (1st column)

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])

np.mean(sepallength[1:])

np.median(sepallength[1:])

np.std(sepallength[1:])

# 29. How to normalize an array so the values range exactly between 0 and 1?

# Create a normalized form of iris's sepallength whose values range exactly between 0 and 1 so that the minimum has value 0 and maximum has value 1.

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

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

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

(sepallength - Smin)/(Smax - Smin)

# or

(sepallength - Smin)/sepallength.ptp() # ptp() => range of values (maximum - minimum) along an axis (peak-to-peak)

# 30. How to compute the softmax score?

# Compute the softmax score of sepallength.

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

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

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)

softmax(sepallength)

# 31. How to find the percentile scores of a numpy array?

# Find the 5th and 95th percentile of iris's sepallength

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

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

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

# 32. How to insert values at random positions in an array?

# Insert np.nan values at 20 random positions in iris\_2d dataset

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

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

# version 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

# version 2

np.random.seed(100)

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

# 33. How to find the position of missing values in numpy array?

# Find the number and position of missing values in iris\_2d's sepallength (1st column)

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

# Number of missing values

np.isnan(iris\_2d[:, 0]).sum()

# Position of missing values

np.where(np.isnan(iris\_2d[:, 0]))

# 34. How to filter a numpy array based on two or more conditions?

# Filter the rows of iris\_2d that has petallength (3rd column) > 1.5 and sepallength (1st column) < 5.0

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])

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

iris\_2d[condition]

# 35. How to drop rows that contain a missing value from a numpy array?

# Select the rows of iris\_2d that does not have any nan value.

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])

# version 1:

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

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

# version 2:

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

# 36. How to find the correlation between two columns of a numpy array?

# Find the correlation between SepalLength(1st column) and PetalLength(3rd column) in iris\_2d

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[:,[0]]

# version 1

np.corrcoef(iris\_2d[:, 0], iris\_2d[:, 2]) # returns correlation coefficient matrix

corr = np.corrcoef(iris\_2d[:, 0], iris\_2d[:, 2])[0, 1]

# version 2

from scipy.stats.stats import pearsonr

pearsonr(iris\_2d[:, 0], iris\_2d[:, 2])

corr, p\_value = pearsonr(iris\_2d[:, 0], iris\_2d[:, 2])

# 37. How to find if a given array has any null values?

# Find out if iris\_2d has any missing values.

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()

# 38. How to replace all missing values with 0 in a numpy array?

# Replace all ccurrences of nan with 0 in numpy array

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

iris\_2d[np.isnan(iris\_d2)] = 0

# 39. How to find the count of unique values in a numpy array?

# Find the unique values and the count of unique values in iris's species

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')

# Extract the species column as an array

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

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

# 40. How to convert a numeric to a categorical (text) array?

# Bin the petal length (3rd) column of iris\_2d to form a text array, such that if petal length is:

# Less than 3 --> 'small'

# 3-5 --> 'medium'

# '>=5 --> 'large'

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\_2d[:,2].astype("float"), [0,3,5,10])

# array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 4, 1,

# 1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

# 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

# 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2,

# 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3,

# 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 3, 2, 2, 3, 3, 4, 3,

# 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3])

# np.digitize => Return the indices of the bins to which each value in input array belongs.

# Each index i returned is such that bins[i-1] <= x < bins[i] if bins is monotonically increasing,

# or bins[i-1] > x >= bins[i] if bins is monotonically decreasing.

# 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]

# 41. How to create a new column from existing columns of a numpy array?

# Create a new column for volume in iris\_2d, where volume is (pi x petallength x sepal\_length^2)/3

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')

# 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

# => newaxis used to increase the dimension of the existing array by one more dimension

# 1D array will become 2D array, 2D array will become 3D array, 3D array will become 4D array, etc.

volume = volume[:, np.newaxis]

# Add the new column

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

# 42. How to do probabilistic sampling in numpy?

# Randomly sample iris's species such that setose is twice the number of versicolor and virginica

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

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

# Get the species column

species = iris[:, 4]

# version 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])

# version 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))

# => prefer version 2: creates an index variable that can be used to sample 2d tabular data.

# 43. How to get the second largest value of an array when grouped by another array?

# What is the value of second longest petallength of species setosa

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')

# Get the species and petal length columns

# iris[:,4] == b"Iris-setosa" => select the setosa species

# iris[,[2]] => select the 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]

# 44. How to sort a 2D array by a column

# Sort the iris dataset based on sepallength column.

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')

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

# 45. How to find the most frequent value in a numpy array?

# Find the most frequent value of petal length (3rd column) in iris dataset.

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')

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

vals[np.argmax(counts)]

# 46. How to find the position of the first occurrence of a value greater than a given value?

# Find the position of the first occurrence of a value greater than 1.0 in petalwidth 4th column of iris dataset.

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

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

np.where(iris[:,3].astype(float) > 1)[0][0]

# np.where(condition[, x, y])

# If both x and y are specified, the output array contains elements of x where condition is True,

# and elements from y elsewhere.

# If only condition is given, return the tuple condition.nonzero(), the indices where condition is True.

# np.argwhere(iris[:,3].astype(float) > 1)[0]

# np.argwhere(a)

# Find the indices of array elements that are non-zero, grouped by element.

# 47. How to replace all values greater than a given value to a given cutoff?

# From the array a, replace all values greater than 30 to 30 and less than 10 to 10.

np.random.seed(100)

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

# version 1: Using np.clip

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

# version 2: Using np.where

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

# version 3 (less elegant)

a[a > 30] = 30

a[a < 10] = 10

# 48. How to get the positions of top n values from a numpy array?

# Get the positions of top 5 maximum values in a given array a.

np.random.seed(100)

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

# version 1:

a[a.argsort()][-5:] # a.argsort() -> return sorted indices

# version 2:

np.sort(a)[-5:]

# version 3:

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

# version 4:

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

# np.partition(a, kth, axis=-1, kind='introselect', order=None)

# => return a partitioned copy of an array

# => creates a copy of the array with its elements rearranged in such a way that

# the value of the element in k-th position is in the position it would be in a sorted array.

# All elements smaller than the k-th element are moved before this element and all equal or greater are moved behind it.

# The ordering of the elements in the two partitions is undefined.

# 49. How to compute the row wise counts of all possible values in an array?

# Compute the counts of unique values row-wise.

# Output contains 10 columns representing numbers from 1 to 10. The values are the counts of the numbers in the respective rows.

# For example, Cell(0,2) has the value 2, which means, the number 3 occurs exactly 2 times in the 1st row.

np.random.seed(100)

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

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

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

# 50. How to convert an array of arrays into a flat 1d array?

# Convert array\_of\_arrays into a flat linear 1d array.

arr1 = np.arange(3)

arr2 = np.arange(3,7)

arr3 = np.arange(7,10)

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

# version 1

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

# version 2

np.concatenate(array\_of\_arrays)

# 51. How to generate one-hot encodings for an array in numpy?

# Compute the one-hot encodings (dummy binary variables for each unique value in the array)

np.random.seed(101)

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

# version 1

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)

# version 2

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

# 52. How to create row numbers grouped by a categorical variable?

# Create row numbers grouped by a categorical variable. Use the following sample from iris species as input.

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))

[i for val in np.unique(species\_small)

for i, grp in enumerate(species\_small[species\_small==val])]

# 53. How to create group ids based on a given categorical variable?

# Create group ids based on a given categorical variable. Use the following sample from iris species as input.

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))

# version 1 - list comprehension:

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 group

for s in species\_small[species\_small==val]: # each element in group

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

output.append(groupid)

# 54. How to rank items in an array using numpy?

# Create the ranks for the given numeric array a.

np.random.seed(10)

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

a.argsort().argsort()

# 55. How to rank items in a multidimensional array using numpy?

# Create a rank array of the same shape as a given numeric array a.

np.random.seed(10)

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

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

# .ravel() -> Return a contiguous flattened array.

# 56. How to find the maximum value in each row of a numpy array 2d?

# Compute the maximum for each row in the given array.

np.random.seed(100)

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

# version 1

np.amax(a, axis=1)

# version 2

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

# 57. How to compute the min-by-max for each row for a numpy array 2d?

# Compute the min-by-max for each row for given 2d numpy array.

np.random.seed(100)

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

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

# 58. How to find the duplicate records in a numpy array?

# Find the duplicate entries (2nd occurrence onwards) in the given numpy array and mark them as True. First time occurrences should be False.

np.random.seed(100)

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

# 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

# 59. How to find the grouped mean in numpy?

# Find the mean of a numeric column grouped by a categorical column in a 2D numpy array

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')

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

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

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

# version 1: List comprehension

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

# version 2: For Loop version

output = []

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

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

# 60. How to convert a PIL image to numpy array?

# Import the image from the following URL and convert it to a numpy array.

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. How to drop all missing values from a numpy array?

# Drop all nan values from a 1D numpy array

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

a[~np.isnan(a)]

# 62. How to compute the euclidean distance between two arrays?

# Compute the euclidean distance between two arrays a and b.

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

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

np.linalg.norm(a-b)

# 63. How to find all the local maxima (or peaks) in a 1d array?

# Find all the peaks in a 1D numpy array a. Peaks are points surrounded by smaller values on both sides.

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

doublediff = np.diff(np.sign(np.diff(a)))

peak\_locations = np.where(doublediff == -2)[0] + 1

# 64. How to subtract a 1d array from a 2d array, where each item of 1d array subtracts from respective row?

# Subtract the 1d array b\_1d from the 2d array a\_2d, such that each item of b\_1d subtracts from respective row of a\_2d.

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

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

a\_2d - b\_1d[:,None]

# 65. How to find the index of n'th repetition of an item in an array

# Find the index of 5th repetition of number 1 in x.

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

n = 5

# version 1: List comprehension

[i for i, v in enumerate(x) if v == 1][n-1]

# version 2: Numpy version

np.where(x == 1)[0][n-1]

# 66. How to convert numpy's datetime64 object to datetime's datetime object?

# Convert numpy's datetime64 object to datetime's datetime object

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

from datetime import datetime

dt64.tolist()

# or

dt64.astype(datetime)

# 67. How to compute the moving average of a numpy array?

# Compute the moving average of window size 3, for the given 1D array.

np.random.seed(100)

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

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

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

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

return ret[n - 1:] / n

# version 1

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

# version 2

# 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')

# np.convolve() => returns the discrete, linear convolution of two one-dimensional sequences.

# The convolution operator is often seen in signal processing, where it models the effect of a linear time-invariant system on a signal [1].

# In probability theory, the sum of two independent random variables is distributed according to the convolution of their individual distributions.

# np.ones() # => returns a new array of given shape and type, filled with ones.

# 68. How to create a numpy array sequence given only the starting point, length and the step?

# Create a numpy array of length 10, starting from 5 and has a step of 3 between consecutive numbers

length = 10

start = 5

step = 3

def seq(start, length, step):

end = start + (step\*length)

return np.arange(start, end, step)

seq(start, length, step)

# 69. How to fill in missing dates in an irregular series of numpy dates?

# Given an array of a non-continuous sequence of dates. Make it a continuous sequence of dates, by filling in the missing dates.

dates = np.arange(np.datetime64('2018-02-01'), np.datetime64('2018-02-25'), 2)

# version 1

filled\_in = np.array([np.arange(date, (date+d)) for date, d in zip(dates, np.diff(dates))]).reshape(-1)

# add the last day

output = np.hstack([filled\_in, dates[-1]])

output

# version 2: For loop version

out = []

for date, d in zip(dates, np.diff(dates)):

out.append(np.arange(date, (date+d)))

filled\_in = np.array(out).reshape(-1)

# add the last day

output = np.hstack([filled\_in, dates[-1]])

output

# 70. How to create strides from a given 1D array?

# From the given 1d array arr, generate a 2d matrix using strides, with a window length of 4 and strides of 2, like [[0,1,2,3], [2,3,4,5], [4,5,6,7]..]

arr = np.arange(15)

arr

def gen\_strides(a, stride\_len=5, window\_len=5):

n\_strides = ((a.size-window\_len)//stride\_len) + 1

# return np.array([a[s:(s+window\_len)] for s in np.arange(0, a.size, stride\_len)[:n\_strides]])

return np.array([a[s:(s+window\_len)] for s in np.arange(0, n\_strides\*stride\_len, stride\_len)])

gen\_strides(np.arange(15), stride\_len=2, window\_len=4)

[ [python](https://mzuer.github.io/tag/python) [numpy](https://mzuer.github.io/tag/numpy) [vector](https://mzuer.github.io/tag/vector) [array](https://mzuer.github.io/tag/array) ]

* [**email**](mailto:zuffereymarie0@gmail.com)

* [**github**](https://github.com/mzuer)