NumPy Exercises

These exercises range from dead simple to rather complex. Each has code examples to get you going (often they include a complete solution, but the idea is for you to experiement)
Begin a new Jupyter notebook (or a new Python module if you prefer) and write your code in that

Compute the mean standard deviation and variance of a given array

```
x = np.arange(6)

np.mean(x) # compare with np.average(x)

np.std(x)

r1= np.var(x)
```

For example:

```
Original array: [0 1 2 3 4 5]

Mean: 2.5

std: 1.707825127659933

variance: 2.9166666666666665
```

Compute cross-correlation of two given arrays (use np.cov)

```
x = np.array([0, 1, 3])

y = np.array([2, 4, 5])

np.cov(x, y))
```

Count the number of occurrences of each value in a given array of non-negative integers (use np.bincount)

```
array1 = [0, 1, 6, 1, 4, 1, 2, 2, 7]

np.bincount(array1)
```

Compute and display the histogram of numbers against a set of 'bins' (explore with more complex data than shown here)

```
nums = np.array([0.5, 0.7, 1.0, 1.2, 1.3, 2.1])

bins = np.array([0, 1, 2, 3])

plt.hist(nums, bins=bins)

plt.show()
```

Create a 3×6 numpy array of boolean 'True' values, then transpose it

Create an array of the integers 1 to 1000, then populate another array from the even numbers that exist in the first array. Populate another array with just the primes from the first array. Time this operation and aim for very large sets of data

Make an array that contains 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])

Use zip to combine all the following Irish places with their country or origin (ie)

(You don't need numpy but you could generate a numpy array of the results)

["Dublin","Cork","Limerick","Galway","Waterford","Drogheda","Kilkenny","Wexford","Sligo","Clonmel","Dundalk","Bray","Ennis","Tralee","Carlow","Naas","Athlone","Letterkenny","Tullamore","Killarney","Arklow","Cobh","Castlebar","Midleton","Mallow","Ballina","Enniscorthy","Wicklow","Cavan","Athenry","Longford","Dungarvan","Nenagh","Trim","Thurles","Youghal","Monaghan","Buncrana","Ballinasloe","Fermoy","Westport","Carrick-on-Suir","Birr","Tipperary","Carrickmacross","Kinsale","Listowel","Clonakilty","Cashel","Macroom","Castleblayney","Kilrush","Skibbereen","Bundoran","Templemore","Clones","Newbridge","Mullingar","Balbriggan","Greystones","Leixlip","Tramore","Shannon","Gorey","Tuam","Edenderry","Bandon","Loughrea","Ardee","Mountmellick","Bantry","Boyle","Ballyshannon","Cootehill","Ballybay","Belturbet","Lismore","Kilkee","Granard"]

Find the missing values in a numpy array loaded from the internet
Also show the position of those missing values

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

np.isnan(iris_2d[:, 0]).sum())

np.where(np.isnan(iris_2d[:, 0])))

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]

Find any rows that contain a missing value from iris_2d and drop them from the 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

# There is no direct numpy function...

any_nan_in_row = np.array([~np.any(np.isnan(row)) for row in iris_2d])

iris_2d[any_nan_in_row][:5]

# alternatively

iris_2d[np.sum(np.isnan(iris_2d), axis = 1) == 0][:5]
```

Find the correlation between SepalLength(1st column) and PetalLength(3rd column) in iris_2d

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

# It can range between -1 to +1.

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])

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