

Histogramming N-dimensional arrays in NumPy with "histogramdd"

`numpy.ndarray`

A multidimensional, homogeneous (`dtype=float` by default) array of fixed-size items;

shape (tuple of integers): with every "[" we add a **dimension** (or **axis**):

```
a = np.array([[...[x1, x2, ... , xN]...]])
```

```
print(type(a))
```

```
<class 'numpy.ndarray'>
```

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

 the "simple" array

```
print(b.shape)
```

(5,) ⇒ to indicate a *tuple* of integers, not a simple integer (the second dimension is not defined, `b.ndim=1`)

```
b2 = b[np.newaxis, :]
```

```
print(b2.shape, b2)
```

```
(1, 5) [[1 2 3 4 5]]
```

`numpy.histogramdd`

Compute the multidimensional histogram of some data.

Notice the unusual interpretation ...

When an array, each row is a coordinate in a D-dimensional space !

Try with:

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

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

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

```
a = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])
```

```
h, e = np.histogramdd(a, bins = 2)
```

histogramdd.ipynb

https://drive.google.com/file/d/1ox3hhjlj-unKUEwoBwwPYO9vEZImV0tn/view?usp=drive_link