

Numpy

We have seen python basic data structures in our last section. They are great but lack specialized features for data analysis. Like, adding rows, columns, operating on 2d matrices aren't readily available. So, we will use *numpy* for such functions.

```
In [2]: import numpy as np
```

Numpy operates on *nd* arrays. These are similar to lists but contains homogenous elements but easier to store 2-d data.

```
In [2]: l1 = [1,2,3,4]
nd1 = np.array(l1)
print(nd1)

l2 = [5,6,7,8]
nd2 = np.array([l1,l2])
print(nd2)
```

```
[1 2 3 4]
[[1 2 3 4]
 [5 6 7 8]]
```

Sum functions on np.array()

```
In [3]: print(nd2.shape)

print(nd2.size)

print(nd2.dtype)
```

```
(2, 4)
8
int64
```

Question 1

Create an identity 2d-array or matrix (with ones across the diagonal).

[Hint: ** You can also use **np.identity()** function]

```
In [4]: np.identity(2)
```

```
Out[4]: array([[1., 0.],
               [0., 1.]])
```