**Numpy (Numerical Python)**

**What is NumPy?**
Foundation package for scientific computing in Python

Why NumPy?
• NumPy ‘ndarray’ is a much more efficient way of storing and manipulating “numerical data” than the built-in Python data structures.
• Libraries written in lower-level languages, such as C, can operate on data stored in NumPy ‘ndarray’ without copying any data.

**N-Dimensional Array (NDArray)**

What is NdArray?
Fast and space-efficient multidimensional array (container for homogeneous data) providing vectorized arithmetic operations

---

### Numpy Cheat Sheet

**What is NumPy?**
Foundation package for scientific computing in Python

**Why NumPy?**
- NumPy ‘ndarray’ is a much more efficient way of storing and manipulating “numerical data” than the built-in Python data structures.
- Libraries written in lower-level languages, such as C, can operate on data stored in NumPy ‘ndarray’ without copying any data.

**N-Dimensional Array (NDArray)**

**What is NdArray?**
Fast and space-efficient multidimensional array (container for homogeneous data) providing vectorized arithmetic operations

---

#### Create NdArray

```python
np.array(seq1)
# seq1 - any sequence like object, i.e. [1, 2, 3]
```

#### Create Special NdArray

```python
1, np.zeros(10)
# one dimensional ndarray with 10 elements of value 0
2, np.ones(2, 3)
# two dimensional ndarray with 6 elements of value 1
3, np.empty(3, 4, 5)
# three dimensional ndarray of uninitialized values
4, np.eye(N) or np.identity(N)
# creates N by N identity matrix
```

#### NdArray version of Python's `range`

```python
np.arange(1, 10)
```

#### Get # of Dimension

```python
ndarray1.ndim
```

#### Get Dimension Size

```python
dimsize, dim2size, .. = ndarray1.shape
```

#### Get Data Type

```python
ndarray1.dtype
```

#### Explicit Casting

```python
ndarray2 = ndarray1.astype(np.int32)
```

---

#### Boolean indexing

```python
ndarray1[(names == 'Bob')] | (names == 'Will', 2):
# '2' means select from 3rd column on
```

#### Fancy indexing (aka ‘indexing using integer arrays’)

Select a subset of rows in a particular order:

```python
ndarray1[ [3, 8, 4] ]
ndarray1[ [-1, 6] ]
# negative indices select rows from the end
```

#### Fancy indexing ALWAYS creates a copy of the data.

**COMMON OPERATIONS**

1. **Transposing**
   • A special form of reshaping which returns a ‘view’ on the underlying data without copying anything.

   ```python
   ndarray1.transpose() or
   ndarray1.T
   ndarray1.swapaxes(0, 1)
   ```

2. **Vectorized wrappers (for functions that take scalar values)**
   • `math.sqrt()` works on only a scalar

   ```python
   np.sqrt(seq1) # any sequence (list, ndarray, etc) to return a ndarray
   ```

3. **Vectorized expressions**
   • `np.where((cond, x, y) is a vectorized version of the expression 'x if condition else y`)

   ```python
   np.where((matrixArray > 0, 1, -1) => a new array (same shape) of 1 or -1 values
   np.where((cond, 1, 0), argmax) *
   => Find the first True element
   ```

4. **Aggregations/Reductions Methods (i.e. mean, sum, std)**

   ```python
   ndarray1.mean() or
   ndarray1.sum
   ```

5. **Boolean arrays methods**

   ```python
   ndarray1 > 0).sum()
   ```

   **Note:** These methods also work with non-boolean arrays, where non-zero elements evaluate to True.

6. **Sorting**

   ```python
   ndarray1.sort()
   ```

7. **Set methods**

   ```python
   np.unique(ndarray1)
   ```

8. **Random number generation (np.random)**

   • Supplements the built-in Python random with functions for efficiently generating whole arrays of sample values from many kinds of probability distributions.

   ```python
   samples = np.random.normal(size=(3, 3))
   ```

---

* Created by Arianne Colton and Sean Chen
* www.datasciencefree.com
* Based on content from “Python for Data Analysis” by Wes McKinney
* Updated: August 18, 2016