Introduction to Computer Programming Lecture 7.2:

Numpy Functionality

Hemma Philamore

Department of Engineering Mathematics

NumPy ("Numerical Python"): scientific computing applications

- Fourier Transform
- Shape Manipulation
- **Mathematical and Logical Operations**
- **Linear Algebra**
- **Random Number Generation**

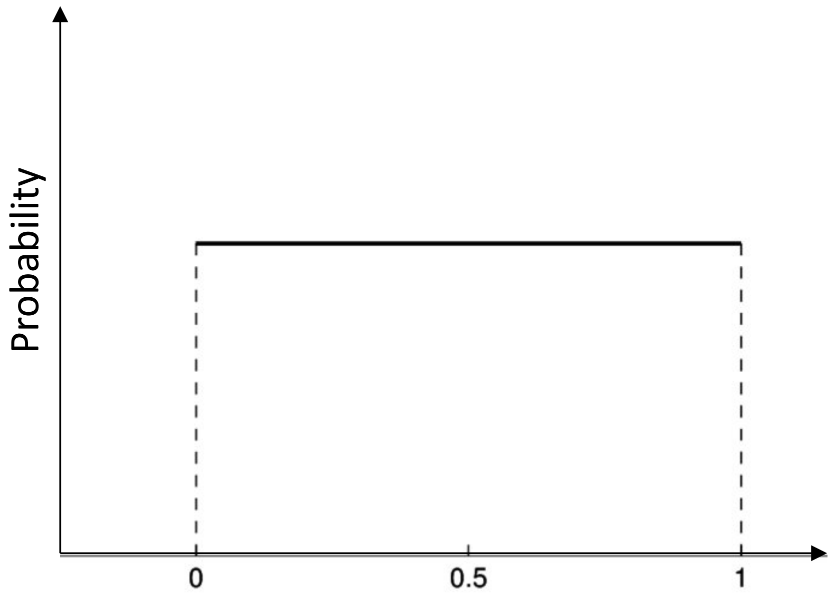
```
import numpy as np
```

Random

Uniform

```
R = np.random.rand(2, 2)
print(R)
```

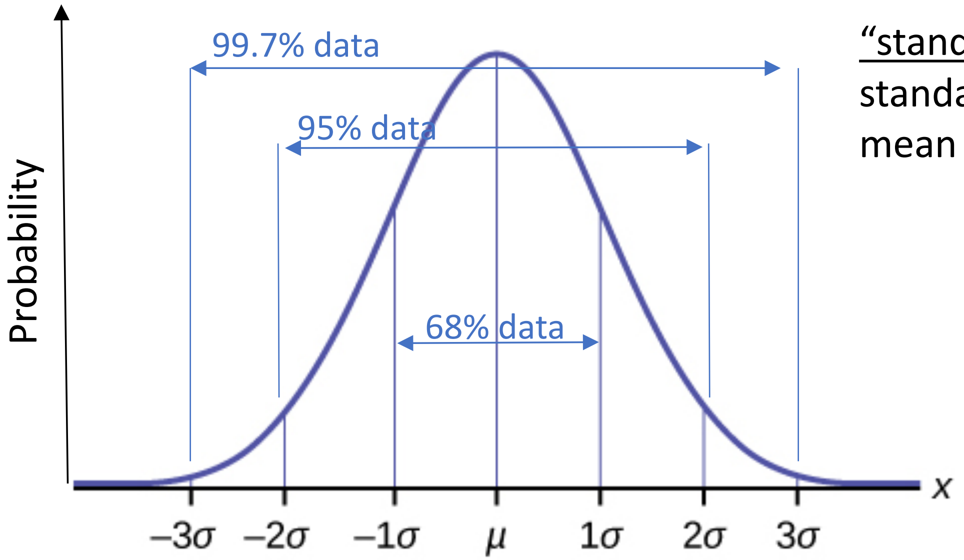
```
[[0.16204554 0.16171102]
 [0.13975907 0.13328215]]
```



Gaussian/ normal

```
R = np.random.randn(3, 3)
print(R)
```

```
[[ 1.79519988  0.37809335  0.9191431 ]
 [-1.01874789 -2.21842261  0.65666726]
 [-0.94934458  0.06886373  0.30048544]]
```



<u>"standard normal" distribution</u>

standard deviation $\sigma = 1$

mean $\mu = 0$

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

x_i = value i

μ =population mean

N =size of population

Min/Max

max

min

n vals

```
R = np.random.randint(50, 100, 5)
print(R)
```

```
[90 56 64 94 90]
```

```
xmin = R.min()
print(xmin)
```

```
56
```

```
xmax = R.max()
print(xmax)
```

```
94
```

Min/Max

max

min

n vals

```
R = np.random.randint(50, 100, 5)
print(R)
```

```
[90 56 64 94 90]
```

```
xmin = R.min()
print(xmin)
```

```
56
```

```
xmax = R.max()
print(xmax)
```

```
94
```

Namespaces

Rename sub-package

```python
import numpy as np
import numpy.random as r

R = r.randint(50, 100, 5)
print(R)
```

```
[56 53 50 64 58]
```

Import individual function

```python
import numpy as np
from numpy.random import randint

R = randint(50, 100, 5)
print(R)
```

```
[50 81 73 63 87]
```

Arithmetic

Numpy operations act **elementwise**, when applied to a numpy array

```
x = np.random.randint(50, 100, 5)
y = np.arange(5)
print(x, y)
```

```
[53 76 62 63 78] [0 1 2 3 4]
```

Elementwise addition

```
z = x + y
print(z)
```

```
[53 77 64 66 82]
```

Elementwise multiplication

```
u = x * y
print(u)
```

```
[  0  76 124 189 312]
```

Elementwise exponential function

```
e = np.exp(y)
print(e)
```

```
[ 1.          2.71828183  7.3890561  20.08553692 54.59815003]
```

Linear Algebra

Matrix multiplication with dot

```
x = np.arange(3)
y = np.arange(3)
print(x)
print(y)
```

```
[0 1 2]
[0 1 2]
```

```
print(x.dot(y))
```

```
5
```

$$[0 \quad 1 \quad 2] \cdot [0 \quad 1 \quad 2] = \text{error!}$$

Numpy manipulates 1D array

$$[0 \quad 1 \quad 2] \cdot \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = (0\times0 + 1\times1 + 2\times2)$$
$$= 5$$

```
x = np.arange(3).reshape(1,3)
y = np.arange(3).reshape(3,1)
print(x)
print(y)
```

```
[[0 1 2]]
[[0]
 [1]
 [2]]
```

```
print(x.dot(y))
```

```
[[5]]
```

$$[0 \quad 1 \quad 2] \cdot \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = (0\times0 + 1\times1 + 2\times2)$$
$$= 5$$

Linear Algebra

Matrix multiplication with dot

```
x = np.arange(3).reshape(1,3)
y = np.arange(3).reshape(3,1)
print(x)
print(y)
```

```
[[0 1 2]]
[[0]
 [1]
 [2]]
```

```
print(x.dot(y))
```

```
[[5]]
```

$$\begin{bmatrix} 0 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = (0\times0 + 1\times1 + 2\times2)$$
$$= 5$$

```
x = np.arange(5).reshape(1,5)
y = np.arange(5).reshape(1,5)
print(x)
print(y)
```

```
[[0 1 2 3 4]]
[[0 1 2 3 4]]
```

```
print(x.dot(y))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-110-9437f9fe4c3a> in <module>
----> 1 print(x.dot(y))

ValueError: shapes (1,5) and (1,5) not aligned: 5 (dim 1) != 1 (dim 0)
```

$$\begin{bmatrix} 0 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 2 \end{bmatrix} = \text{error!}$$

Inner dimensions of 2D arrays must be equal

Inner dimensions are equal (n columns matrix A = m rows matrix B)

```
x = np.ones((2, 3))
y = np.ones((3, 2))
```

```
print(x.dot(y))
```

```
[[3. 3.]
 [3. 3.]]
```

$$\underset{A}{\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\end{bmatrix}} \cdot \underset{B}{\begin{bmatrix}1 & 1\\1 & 1\\1 & 1\end{bmatrix}} = \begin{bmatrix}(1\times1\times1 + 1\times1\times1 + 1\times1\times1) & (1\times1\times1 + 1\times1\times1 + 1\times1\times1)\\(1\times1\times1 + 1\times1\times1 + 1\times1\times1) & (1\times1\times1 + 1\times1\times1 + 1\times1\times1)\end{bmatrix}$$

$$= \begin{bmatrix}3 & 3\\3 & 3\end{bmatrix}$$

Matrices have same dimensions

```
x = np.ones((2, 3))
y = np.ones((2, 3))
```

```
print(x.dot(y))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-103-9437f9fe4c3a> in <module>
----> 1 print(x.dot(y))

ValueError: shapes (2,3) and (2,3) not aligned: 3 (dim 1) != 2 (dim 0)
```

Matrix multiplication

```
print(x * y)
```

```
[[1. 1. 1.]
 [1. 1. 1.]]
```

$$\begin{bmatrix}1\times1 & 1\times1 & 1\times1\\1\times1 & 1\times1 & 1\times1\end{bmatrix}$$

Elementwise multiplication