Devanshi Patel

assignment1

September 4, 2023

#NumPy Assignment 1

[1]:
```
import numpy as np
```

[3]:
```
#### Create an array of 10 zeros
z = np.zeros(10)
z
```

[3]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

[5]:
```
#### Create an array of 10 ones
z = np.ones(10)
z
```

[5]:
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

[7]:
```
#### Create an array of 10 fives
z = np.ones(10)*5
z
```

[7]:
```
array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

[10]:
```
#### Create an array of the integers from 10 to 50
array = [i for i in range(10, 51)]
print(array)
```

```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
50]
```

[56]:
```
arr = np.arange(10,51)
arr
```

[56]:
```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
       27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
       44, 45, 46, 47, 48, 49, 50])
```

[13]:
```
#### Create an array of all the even integers from 10 to 50
even_array = [i for i in range(10, 51) if i % 2 == 0]
print(even_array)
```

```
[10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48,
50]
```

[57]:
```
even_array = np.arange(10,51,2)
even_array
```

[57]:
```
array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
       44, 46, 48, 50])
```

[14]:
```
#### Create a 3x3 matrix with values ranging from 0 to 8
matrix = [[i for i in range(j, j + 3)] for j in range(0, 9, 3)]
for row in matrix:
    print(row)
```

```
[0, 1, 2]
[3, 4, 5]
[6, 7, 8]
```

[15]:
```
matrix = np.array([[0,1,2],[3,4,5],[6,7,8]])
matrix
```

[15]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

[20]:
```
#### Create a 3x3 identity matrix
x = np.eye(3)
x
```

[20]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

[23]:
```
#### Use NumPy to generate a random number between 0 and 1
np.random.rand(1,1)
```

[23]:
```
array([[0.07746918]])
```

[26]:
```
#### Use NumPy to generate an array of 25 random numbers sampled from a␣
 ↪standard normal distribution
np.random.normal(0,1,25)
```

[26]:
```
array([ 0.39942825, -1.53586092, -0.05898107,  0.74632238,  0.11598959,
        0.50283582,  0.57228476,  1.01448965, -0.77749903, -0.15244119,
       -1.23611453,  2.38057288,  1.11478193, -0.01332509, -1.68493857,
        0.11122062, -0.03121761, -0.27372372,  1.34920543, -1.17325096,
        0.77146819,  0.20142229,  0.01665144,  0.05203722, -1.33232653])
```

[29]:
```
#### Create the following matrix:
values = np.arange(0.01, 1.01, 0.01)
array = values.reshape(10, 10)
array
```

[29]:
```
array([[0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 ],
       [0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2 ],
       [0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 ],
       [0.31, 0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 ],
       [0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49, 0.5 ],
       [0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6 ],
       [0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67, 0.68, 0.69, 0.7 ],
       [0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8 ],
       [0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9 ],
       [0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1.  ]])
```

[30]:
```
#### Create an array of 20 linearly spaced points between 0 and 1:
a = np.linspace(0,1,20)
a
```

[30]:
```
array([0.        , 0.05263158, 0.10526316, 0.15789474, 0.21052632,
       0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,
       0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,
       0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.        ])
```

[31]:
```
## Numpy Indexing and Selection
mat = np.arange(1,26).reshape(5,5)
mat
```

[31]:
```
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

[38]:
```
subarray = mat[2:5, 1:5]
subarray
```

[38]:
```
array([[12, 13, 14, 15],
       [17, 18, 19, 20],
       [22, 23, 24, 25]])
```

[43]:
```
subarray1 = mat[3,4]
subarray1
```

[43]:
```
20
```

[45]:
```
subarray2 = mat[0:3,1:2]
subarray2
```

[45]:
```
array([[ 2],
       [ 7],
       [12]])
```

[46]:
```
subarray3 = mat[4:,]
subarray3
```

[46]:
```
array([[21, 22, 23, 24, 25]])
```

[51]:
```
subarray4 = mat[3:,:5]
subarray4
```

[51]:
```
array([[16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

[53]:
```
#### Get the sum of all the values in mat
sum = np.sum(mat)
sum
```

[53]:
```
325
```

[54]:
```
#### Get the standard deviation of the values in mat
standard_derivation = np.std(mat)
standard_derivation
```

[54]:
```
7.211102550927978
```

[55]:
```
#### Get the sum of all the columns in mat
col_sum = np.sum(mat,axis=0)
col_sum
```

[55]:
```
array([55, 60, 65, 70, 75])
```

[ ]:
```

```