Lab_2

Write a code for all answers.

Import NumPy as np

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

Create an array of 10 zeros

```
In [2]: zeros_array = np.zeros(10)
        zeros_array
```

```
Out[2]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

Create an array of 10 ones

```
In [3]: ones_array = np.ones(10)
        ones_array
```

```
Out[3]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

Create an array of 10 fives

```
In [4]: fives_array = ones_array * 5
        fives_array
```

```
Out[4]: array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

Create an array of the integers from 10 to 50

```
In [5]: array_range_all = np.arange(10, 50+1)
        array_range_all
```

```
Out[5]: 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])
```

Create an array of all the even integers from 10 to 50

```
In [6]: array_range_even = np.arange(10, 50+1, 2)
        array_range_even
```

```
Out[6]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
               44, 46, 48, 50])
```

Create a 3x3 matrix with values ranging from 0 to 8

```
In [7]: A = np.arange(9)
        A = A.reshape(3, 3)
        A
```

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

Create a 3x3 identity matrix

```
In [8]:  B = np.eye(3)
         B
```

```
Out[8]:  array([[1., 0., 0.],
                [0., 1., 0.],
                [0., 0., 1.]])
```

Use NumPy to generate a random number between 0 and 1

```
In [9]:  from numpy import random
         x = random.rand()
         x
```

```
Out[9]:  0.47160407569195595
```

Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution

```
In [10]:  random_array = np.random.randn(25)
          random_array
```

```
Out[10]:  array([-0.24268591, -1.06406602, -0.20867844,  0.55001954,  1.77955678,
                  0.88570481,  0.16623592, -1.46455262, -0.07162527,  0.91529099,
                  0.52667408, -1.66007446,  1.65187737,  0.52605809,  0.03743938,
                 -0.47025411,  1.36004015,  0.04264455, -0.63113803, -0.59203478,
                  0.06246459,  2.11486793,  0.10436554, -0.43451637,  0.06952413])
```

Create the following matrix:

```
In [13]:  matrix = np.arange(0.01, 1.01, 0.01).reshape(10, 10)
          matrix
```

```
Out[13]:  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.  ]])
```

Create an array of 20 linearly spaced points between 0 and 1:

```
In [11]:  linear_array = np.linspace(0, 1, 20)
          linear_array
```

```
Out[11]:  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.        ])
```

Numpy Indexing and Selection

Now you will be given a few matrices, and be asked to replicate the resulting matrix outputs:

```
In [14]:  mat = np.arange(1,26).reshape(5,5)
          mat
```

```
Out[14]:  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]])
```

```
In [39]:  # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
```

```
In [15]:  mat[2:, 1:]
```

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

```
In [29]:  # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
```

```
In [16]:  mat[3, 4]
```

```
Out[16]:  20
```

```
In [30]:  # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
```

```
In [17]:  mat[:3, 1:2]
```

```
Out[17]:  array([[ 2],
                 [ 7],
                 [12]])
```

```
In [31]:  # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
```

```
In [18]:  mat[4, :]
```

```
Out[18]:  array([21, 22, 23, 24, 25])
```

```
In [32]:  # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
```

```
In [21]:  mat[3:, :]
```

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

Now do the following

Get the sum of all the values in mat

```
In [22]:  np.sum(mat)
```

```
Out[22]:  325
```

Get the standard deviation of the values in mat

```
In [23]: np.std(mat)
```

```
Out[23]: 7.211102550927978
```

Get the sum of all the columns in mat

```
In [24]: mat.sum(axis=0)
```

```
Out[24]: array([55, 60, 65, 70, 75])
```

```
In [ ]:
```