

# tu07\_re\_BasicNumpy\_HW

February 8, 2023

## 1 Numpy review homework

1. Make a numpy matrix from a Python list of lists...

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

```
[2]: np_matrix = np.array([[2, 4], [3, -1], [-2, 1]])  
np_matrix
```

```
[2]: array([[ 2,  4],  
           [ 3, -1],  
           [-2,  1]])
```

2. Make a 3D numpy matrix from a Python list of lists of lists!

```
[3]: np_matrix = np.array([[[1,2,3], [4,5,6]],  
                           [[7,8,9], [17,18,19]],  
                           [[13,14,15], [16,20,21]])  
np_matrix
```

```
[3]: array([[[ 1,  2,  3],  
            [ 4,  5,  6]],  
           [[ 7,  8,  9],  
            [17, 18, 19]],  
           [[13, 14, 15],  
            [16, 20, 21]])
```

3. Create a 5x3 array of Gaussian random numbers.

```
[94]: gaus = np.random.randn(5,3)  
gaus
```

```
[94]: array([[ 1.12508221, -0.02706748, -0.18929466],  
            [-0.83057479,  2.21326633, -0.69993258],  
            [ 1.44888825, -0.20490622,  0.27818277],  
            [ 0.50185478, -0.03503331, -1.07259301],  
            [ 1.35906972,  0.56607582,  2.35404244]])
```

4. Write a script to go through the array created in 3. and announce (print) the value and its row and column indexes.

Hint: Use nested for loops - one to loop through the rows and one to loop through the columns.

```
[109]: print('Row\t Column\t Value')

for row in range(5):
    for col in range(3):
        print(row, '\t', col, '\t', gaus[row,col])
```

| Row | Column | Value                 |
|-----|--------|-----------------------|
| 0   | 0      | 1.1250822100690685    |
| 0   | 1      | -0.027067479974490067 |
| 0   | 2      | -0.18929466073856904  |
| 1   | 0      | -0.830574794234174    |
| 1   | 1      | 2.2132663250094096    |
| 1   | 2      | -0.6999325804279395   |
| 2   | 0      | 1.4488882462303163    |
| 2   | 1      | -0.2049062187950202   |
| 2   | 2      | 0.27818277079422343   |
| 3   | 0      | 0.5018547770508129    |
| 3   | 1      | -0.03503330924170285  |
| 3   | 2      | -1.0725930061892837   |
| 4   | 0      | 1.3590697166905963    |
| 4   | 1      | 0.5660758209714674    |
| 4   | 2      | 2.354042441757096     |

5. Make an new array out of your random numbers such that the mean is 10 and the standard deviation is 3.

```
[6]: new_array = (gaus * 3) + 10
```

6. Count the number of values in your new array that are below 7.

```
[7]: new_array < 7
```

```
[7]: array([[False, False, False],
          [False, False, False],
          [False,  True, False],
          [ True, False, False],
          [ True,  True, False]])
```

7. Make a numpy sequence that has the even numbers from 2 up to (and including) 20.

```
[13]: seq = np.arange(2,21,2)
print(seq)
```

```
[ 2  4  6  8 10 12 14 16 18 20]
```

8. Get the second and third rows of your array.

```
[23]: print(gaus[1,:])  
      print(gaus[2,:])
```

```
[-0.3868072 -0.34415548  0.50067053]  
[ 0.3005157 -1.11851144 -0.10537119]
```

9. Compute the mean of the columns of your array.

```
[22]: print(np.mean(gaus[:,0]))  
      print(np.mean(gaus[:,1]))  
      print(np.mean(gaus[:,2]))
```

```
-0.7550798442724782  
-0.5318755615341366  
0.23648465214776682
```

```
[ ]:
```