



Data Science: NumPy, Matplotlib

Woong-Kee Loh
2022



NumPy Exercises

(Make sure you understand how the results are obtained)



NumPy: slicing rows of an array

```
import numpy as np
X = np.array ( [ [ 1, 2, 3] , [ 4, 5, 6] , [ 7, 8, 9] ,
                 [ 11, 12, 13] , [ 14, 15, 16] , [ 17, 18, 19] ,
                 [ 21, 22, 23] , [ 24, 25, 26] , [ 27, 28, 29] ] )
X[1]
```



NumPy: slicing columns of an array

```
import numpy as np
X = np.array ( [ [ 1, 2, 3] , [ 4, 5, 6] , [ 7, 8, 9] ,
                 [ 11, 12, 13] , [ 14, 15, 16] , [ 17, 18, 19] ,
                 [ 21, 22, 23] , [ 24, 25, 26] , [ 27, 28, 29] ] )
X[:, 1]
```



NumPy: dicing (slicing rows and columns of an array)

```
import numpy as np
X = np.array ( [ [ 1, 2, 3] , [ 4, 5, 6] , [ 7, 8, 9] ,
                 [ 11, 12, 13] , [ 14, 15, 16] , [ 17, 18, 19] ,
                 [ 21, 22, 23] , [ 24, 25, 26] , [ 27, 28, 29] ] )

print X[ 1, 1]
print X[:, 1, 1]
print X[1, :, 1]
print
print X[1:2, 1:2]
```



NumPy: Array max, min

```
import numpy as np
dataset = np.array ( [ [ 2, 4, 6, 8, 3, 2, 5] ,
                       [ 7, 5, 3, 1, 6, 8, 0] ,
                       [ 1, 3, 2, 1, 0, 0, 8] ] )
print np.max (dataset, axis=1) - np.min(dataset, axis=1)
```



NumPy: Array value normalization

```
import numpy as np
a = np.array ( [ 15.0, 20.0, 22.0, 75.0, 40.0, 35.0 ] )
a = a * .01
print a
```



NumPy: matrix and vector multiplication

```
import numpy as np
a = np.array ( [ [ 2, 4, 6, 8] )
b = np.array ( [ [ 1, 2, 3, 4],
                  [ 2, 3, 4, 5],
                  [ 3, 4, 5, 6],
                  [ 4, 5, 6, 7] ] )
c = np.dot (a, b)
print c
```




Arrays and Constructors

- `>>> a = ones((3,3),float)`
- `>>> print a`
- `[[1., 1., 1.],`
- `[1., 1., 1.],`
- `[1., 1., 1.]]`
- `>>> b = zeros((3,3),float)`
- `>>> b = b + 2.*identity(3) # "+" is overloaded`
- `>>> c = a + b`
- `>>> print c`



Overloaded Operators

- `>>> b = 2.*ones((2,2),float) #overloaded`
- `>>> print b`
- `[[2.,2.],`
- `[2.,2.]]`
- `>>> b = b+1 # Addition of a scalar is`
- `>>> print b # element-by-element`
- `[[3.,3.],`
- `[3.,3.]]`
- `>>> c = 2.*b # Multiplication by a scalar is`
- `>>> print c # element-by-element`



Array Functions

- `>>> from LinearAlgebra import *`
- `>>> a = zeros((3,3),float) + 2.*identity(3)`
- `>>> print inverse(a)`
- `>>> print determinant(inverse(a))`
- `>>> print diagonal(a)`
- `>>> print diagonal(a,1)`



Operations on Arrays

- Calculate $a+b$, $a-b$, $a*b$, a/b
- Save the result in c
- print c

```
import numpy as np
```

```
a = np.array([1,2,3])
```

```
b = np.array([4,5,6])
```



NumPy Coding Exercise

- We want to compute the BMI (body mass index) of 100 students.
 - $\text{BMI} = \text{weight} / (\text{height} * \text{height})$
(* weight in kilograms, height in meters *)
- Create a wt array and an ht array, each of size 100.
 - Fill the wt array with 100 random float numbers between 40.0 and 90.0.
 - Fill the ht array with 100 random integers between 140 and 200 (centimeters).
- Compute the BMI for the 100 students, store them in a bmi array, and print the array.
- Post the screen of the Python/NumPy code, and the first 10 elements of the bmi array to CyberCampus.



Matplotlib Coding Exercise

- Draw the bar chart, histogram, pie chart, and scatter plot of the (height, weight) data in the NumPy exercise. (Use 4 categories for the BMI index)

BMI	Weight status
Below 18.5	Underweight
18.5–24.9	Healthy
25.0–29.9	Overweight
30.0 and above	Obese

- Post the screen of the Python/Matplotlib code, and the plots to CyberCampus.



Matplotlib Coding Exercise (cont'd)

- Bar chart
 - Plot the student distribution for each bmi level (#bars = 4)
- Histogram
 - Plot the student distribution for each bmi level (#bins = 4)
- Pie chart
 - Plot the ratio of students for each bmi level
- Scatter plot
 - Plot (height, weight) points



Submission Guideline

- Submit a single WORD file for all exercises.