

DSP505: Programming Lab for Data Science and Artificial Intelligence

TPL616: Advanced Programming for DSAI

Lab-3: Practice Problems (Numpy)

Date: 03-Sept-2025

Instructions:

1. Try to complete lab problems during the lab hour and submit it through canvas. If you can't complete it within the lab time, you can submit it by the end of tomorrow.
2. Prepare all your solution files in a zip file and name it as <Name.zip> and submit on canvas.
3. You can use a jupyter notebook to solve the problems.

Problems Set-0:

Practice the codes given in the slides, execute them and make sure you get the correct output. No need to submit this.

Problems Set-1:

1. A list baseball has already been defined in the below, representing the height of some baseball players in centimetres. Prepare this as a numpy array.

```
baseball= [180,215,210,210,188,176,209,200]
```

2. Create a numpy array from baseball list defined in Q#1. Name this new array np_height_in.
 - a. Print np_height_in.
 - b. Multiply np_height_in with 0.0254 to convert all height measurements from inches to metres. Store the new values in a new array, np_height_m.
 - c. Print out np_height_m and check if the output makes sense.

3. A list weight_lb has already been defined in the Python script below, representing the weight of some baseball players in pounds

```
Weight_lb=[172,190,165,175,166,159,182,168]
```

Create a numpy array from the weight_lb list with the correct units. Multiply by 0.453592 to go from pounds to kilograms. Store the resulting numpy array as np_weight_kg.

- a. Use np_height_m (given in Q#2) and np_weight_kg to calculate the BMI of each player. Use the following equation: $BMI = \frac{\text{weight (kg)}}{\text{height (m)}^2}$
 - b. Save the resulting numpy array as bmi.
 - c. Print out bmi.
4. The code that calculates the BMI of all baseball players is already concluded in Q#3. Create a boolean numpy array: the element of the array should be True if the corresponding baseball player's BMI is below 21. You can use the < operator for this. Name the array light.
- a. Print the array light.
 - b. Print out a numpy array with the BMIs of all baseball players whose BMI is below 21. Use light inside square brackets to do a selection on the bmi array. e.g. `light < 20`
5. In this question, baseball is a list of lists. The main list contains 4 elements. Each of these elements is a list containing the height and the weight of 4 baseball players, in this order. baseball is already given here for you

```
baseball=[[180,78.4],[215,102.7],[210,98.5],[188,75.2]]
```

- a. Use np.array() to create a 2D numpy array from baseball. Name it np_baseball.
 - b. Print out the type of np_baseball.
 - c. Print out the shape attribute of np_baseball. Use np_baseball.shape
6. You've contacted FIFA for some data and they handed you three lists. The lists are the following:

```
height = np.round(np.random.normal(1.75, 0.20, 1000), 2)
weight = np.round(np.random.normal(60.32, 15, 1000), 2)
age = np.round(np.random.normal(30, 10, 1000), 0)
```

let's stack it altogether

```
np_players = np.column_stack((height, weight, age))
```

Each element in the lists corresponds to a player. The first list, height, representing each player's height in metres. The second list, weight, represents each player's weight in kilograms. The third list, age, representing each player's age in years.

- a. Create a numpy array `np_height` that is equal to the first column of `np_players`.
- b. Print out the mean of `np_height`.
- c. Print out the median of `np_height`

Problems Set-2:

Generate 2D NumPy random array representing sales data (units sold) of 6 products over 8 weeks. Compute the following.

- a. Weekly Total Sales: Compute the total sales for each week (column-wise sum).
- b. Best-Selling Product: Find the product (row index) that sold the most units in total.
- c. Top 2 Weeks for Each Product: For each product (row), extract the indices of the two highest-selling weeks.
- d. Products Above Average
Compute the overall average sales (scalar). Return a Boolean mask of shape (6, 8) showing where sales exceed this average.

You can use the following numpy functions for solving.

`sum`, `mean`, `argmax`, `argsort`.