IEMS5731 Data Science in Practice (Spring 2025 and Summer 2025)
Assignment 1
Expected time: 6 hours

Learning outcomes:
1. To practise basic Python programming science. 2. To obtain hands-on experience with `NumPy`, `Pandas` and `SciPy` packages. 3. To understand the basic concepts of a data science project.

Instructions:

1. Do your own work. You are welcome to discuss the problems with your fellow classmates. Sharing ideas is great, and do write your own explanations.
2. If you use help from the AI tools, e.g. ChatGPT, write clearly how much you obtain help from the AI tools. No marks will be taken away for using any AI tools with a clear declaration.
3. All work should be submitted onto the blackboard before the due date.
4. You are advised to submit the following two items.
   a. A .pdf file containing the answers of the written part.
   b. A Assignment1_1155xxxxxx.py file storing all your programs for the five problems. (1155xxxxxx refers to your student ID)
5. Do type/write your work neatly. If we cannot read your work, we cannot grade your work.
6. We will grade your work based on Anaconda Python version 1.12.
7. The sample output in the specification is run based on Windows. Your output may be different if you run the programs on Mac or Linux machines.
8. If you do not put down your name, student ID in your submission (both .pdf and .py), you will receive a 10% mark penalty out of the assignment 1.
9. Due date:
   a. Session A: 21st January, 2025 (Tuesday) 23:59
   b. Session B: 5th April, 2025 (Saturday) 23:59

Short questions (25%)
Answer all questions.

1. You are working for data engineering in a small local IT company. Your company is planning to develop some AI models for some European companies. You boss ask you something about EU AI Act:

    a. What are the **four levels of risks** in the EU AI Act? (4%)

    b. Suggest **three advantages** of setting the EU AI Act. (3%)

    c. Give **three different recognition systems** falling into different levels of risks. (6%)

    d. Give **three different AI systems in games** falling into different levels of risks. (6%)

    e. If we are training a model for detecting spam emails using deep learning. This model keeps updating the knowledge based on the new spam emails. What is the level of risk of this model? Explain your choice. (2%)

    f. If we are training an AI teacher using deep learning. This model assists the students in solving high school mathematics problems. What is the level of risk of this model? Explain your choice. (2%)

    g. If we are training a deep learning model for generating the appearance of cartoon characters. What is the level of risk of this model? Explain your choice. (2%)

Practical problems (75%)
Work on the following five problems in a single .py file.
If your Python program cannot be run, you will receive no scores for the problem.
Put down your student ID as a comment in your first line of the program.

2. Write a function that generates the hollow right angled isosceles triangle (see execution below) based on the input size (positive integer). Pay attention that no trailing spaces appear at the end of each line. (15%)
   This problem covers string manipulation and nested loops.

   Code:

```
# <Your student ID>
import numpy as np
import pandas as pd
# Problem 2
def problem_2(n):
    output = ""
    # write your logic here

    return output
```

   Execution:

```
> print(problem_2(3))
@
@@
@@@

> s = problem_2(6)
> print(s)
@
@@
@ @
@  @
@   @
@@@@@@
```

3. Write a function that calculates and returns all non-negative eigenvalues of the input NumPy matrix. Also, returns the corresponding eigenvectors. (15%)
This problem covers basic functions and NumPy operations.

Code:

```
# Problem 3
from numpy.linalg import eig
def problem_3(mat):
    # write your logic here
    eigenvalues, eigenvectors = np.array([0]), np.array([[0]])

    return eigenvalues, eigenvectors
```

Execution:

```
> m = np.array([[2, 2, 4], [1, 3, 5], [2, 3, 4]])
> eigenvalues, eigenvectors = problem_3(m)
> print("non-negative eigenvalues :", eigenvalues)
non-negative eigenvalues : [8.80916362 0.92620912]
> print("corresponding eigenvectors :", eigenvectors)
corresponding eigenvectors : [[-0.52799324, -0.77557092],
 [-0.604391  ,  0.62277013],
 [-0.59660259, -0.10318482]]
```

4. Write a function that identifies the students (Student ID) having a total at least 85, in which the “Midterm examination” counts 20%, “Final examination” counts 80%. (15%)
This problem covers basic functions and Pandas operations.

Code:

```
# Problem 4
def problem_4(record):
    # write your logic here
    output = []

    return output
```

Execution:

```
> record = pd.read_csv("score.csv")
> list_of_student_id = problem_4(record)
> print("list of student having overall >= 85: ", list_of_student_id)
list of student having overall >= 85:  [1155100002, 1155100010]
```

5. Write a function to select the rows from a dataframe such that the score of ‘Midterm examination’ is not within one sample standard deviation of the mean of the ‘Midterm examination’. (15%)
This problem covers Pandas operations and basic statistics.

Code:

```
# Problem 5
def problem_5(record):
    # write your logic here, df is a dataframe, instead of number
    df = 0

    return df
```

Execution:

```
> record = pd.read_csv("score.csv")
> result = problem_5(record)
> print("dataframe for midterm not within one sd: \n", result)
dataframe for midterm not within one sd:
    Session  Student ID Surname First name  Midterm examination
Final examination
0         A  1155100000    Chan   Tai Ming                   73              65
5         A  1155100005      Li    Xiaomin                   91              56
10        B  1155100010    Wang      Bowen                   70              96
12        C  1155100012    Wong     Po Man                   73              66
14        C  1155100014    Wang      Bowen                   98              70
```

6. Write a function to calculate the p-value and Chi-square statistics from the Chi squared tests based on a list of the absolute difference of two dice. The purpose of the test is to see if the two dice are fair dice based on the absolute differences. You can assume that all absolute differences are between 0 and 5 inclusively. (15%)
This problem covers basic statistics on Chi squared tests.

Code:

```
# Problem 6
from scipy.stats import chisquare
def problem_6(list_of_difference):
    p = 0
    chi2 = 0
    # write your logic here

    return p, chi2
```

Execution:

```
> abs_diff = [0,1,0,2,4,5,3,2,1,1,1,2,4,2,3,0,1,2,2,2,3,5]
> p, chi2 = problem_6(abs_diff)
> print("p-value :", p)
p-value : 0.8584965940462701
> print("chi-square :", chi2)
chi-square : 1.9318181818181817
```

< End of Assignment >