MSc Data Science and Artificial Intelligence admission level check
M1 intake

The MSc Data Science and Artificial Intelligence (DSAI) is a master's degree of excellence, requiring a solid grounding in programming, algorithms, statistics and probability.

To self-assess your level in these subjects, please complete the exercises on Python and Statistics. We recommend that you take the time to do these exercises, and above all that you do them yourself, without any help or use of Chat GPT.

If you struggle to complete these exercises, it may indicate that following the MSc DSAI program will be challenging and may lead to difficulties in succeeding. Therefore, please take this assessment seriously and submit your solutions within this document in a single PDF format. You fill find the exercices in the following pages.

EXERCISES IN LINEAR ALGEBRA

How did you find the exercices ?

Please tick a box.

Very easy	Easy	Medium	Hard	Very hard
☐	☐	☐	☐	☐

Comment if any :

EXERCISES IN PROBABILITY AND STATISTICS

How did you find the exercices

Please tick a box.

Very easy	Easy	Medium	Hard	Very hard
☐	☐	☐	☐	☐

Comment if any :

EXERCISES IN PYTHON PROGRAMMING

How did you find the exercices ?

Please tick a box.

Very easy	Easy	Medium	Hard	Very hard
☐	☐	☐	☐	☐

Comment if any :

Admission MSc DSAI

Michel RIVEILL

December 2025

1 Linear algebra

Exercise I. We consider the following system of linear equations:

$$\begin{cases} 2x_1 + 3x_2 + x_3 & = 8 \\ 4x_1 + 7x_2 + 5x_3 & = 20 \\ -2x_2 + 2x_3 & = 0\,. \end{cases}$$

1. Write this system as $Ax = b$, where A is a matrix and x and b are vectors.
2. Solve for x. How many solutions do you find?

Exercise II. Let us consider the matrix

$$A = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

1. Show that the eigenvalues of A are $0, 1$, and 3.
2. For each eigenvalue, find an eigenvector.
3. Is A diagonalizable? If yes, write it down.

2 Probability and Statistics

You can find the table needed to solve the exercises here: `https://engineering.purdue.edu/\textasciitildereibman/ece302/normal_cdf.pdf`

Exercise III. Let X be a discrete random variable with the following probability distribution function:

$$P(X = 1) = 0.2, \quad P(X = 2) = 0.3, \quad P(X = 3) = 0.4, \quad P(X = 4) = 0.1$$

Calculate the mean and variance of X.

Exercise IV. Let three independant real value random variables X_1, X_2 et X_3 with $E[X_1] = 2$, $E[X_2] = 3$, $E[X_3] = 0$, $V[X_1] = 1$, $V[X_2] = 2$, $V[X_3] = 3$:

1. Compute $E[2X_1 + 3]$
2. Compute $V[2X_1 + 3]$
3. Give the correlation matrix of the random vector (X_1, X_2, X_3)
4. Compute $E[X_1 + X_2 + X_3]$
5. Compute $V[X_1 + X_2 + X_3]$
6. Compute $E\left[\frac{X_1+X_2+X_3}{3}\right]$
7. Compute $V\left[\frac{X_1+X_2+X_3}{3}\right]$

Exercise V. Let X a random variable following a standard normal distribution, and let consider $Y = X^2$

1. Let $y \geq 0$, what is the condition on X to have $Y \leq y$?
2. Show that $F_Y(y) := P(Y \leq y) = P(-\sqrt{y} \leq X \leq \sqrt{y})$, where F_Y is the cumulative distribution function (cdf) of the random variable Y.
3. Let F_X the cdf of X, prove that $F_Y(y) = F_X(\sqrt{y}) - F_X(-\sqrt{y})$ for $y \geq 0$
4. Using $F_X(-x) = 1 - F_X(x)$ (property of the cdf of normal distribution) deduce that $F_Y(y) = 2F_X(\sqrt{y}) - 1$
5. Using the cdf of the $\mathcal{N}(0; 1)$ deduce $P(Y < 3)$ and $P(Y < 5)$
6. What is the link between $f_Y(y)$, the probability distribution function (pdf) of Y and $F_Y(y)$? (give the formula)
7. Deduce that $f_Y(y) = \frac{1}{\sqrt{y}} F'_X(\sqrt{y}) = \frac{1}{\sqrt{y}} f_X(\sqrt{y})$ for all $y \geq 0$ with f_X the pdf of an $\mathcal{N}(0; 1)$.
8. We remind that $f_X(x) = \dfrac{1}{\sqrt{2\pi}} \exp\left(-\dfrac{x^2}{2}\right)$ $\forall x \in \mathbb{R}$. Thus deduce f_Y

Exercise VI. Let assumed that $X_1, \cdots, X_n$ are n independent and identically distributed random variables from a normal distribution $\mathcal{N}(\mu, \sigma^2)$.

Let denote by $\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$

1. Compute $E[\bar{X}_n]$
2. Compute $V[\bar{X}_n]$
3. Give the distribution of $\bar{X}_n$

3 Python programming

Exercise VII. It is commonly said that one human year is equivalent to 7 dog years. However this simple conversion fails to recognize that dogs reach adulthood in approximately two years. As a result, some people believe that it is better to count each of the first two human years as 10.5 dog years, and then count each additional human year as 4 dog years.

Write a program that implements the conversion from human years to dog years described in the previous paragraph. Ensure that your program works correctly for conversions of less than two human years and for conversions of two or more human years. Your program should display an appropriate error message if the user enters a negative number.

Exercise VIII. A univariate quadratic function has the form $f(x) = ax^2 + bx + c$, where a, b and c are constants, and a is non-zero. The roots of a quadratic function can be found by finding the values of x that satisfy the quadratic equation $ax^2 + bx + c = 0$. A quadratic function may have 0, 1 or 2 real roots. These roots can be computed using the quadratic formula:

$$root = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The portion of the expression under the square root sign is called the discriminant. If the discriminant is negative then the quadratic equation does not have any real roots. If the discriminant is 0, then the equation has one real root. Otherwise the equation has two real roots, and the expression must be evaluated twice, once using a plus sign, and once using a minus sign, when computing the numerator.

Write a program that computes the real roots of a quadratic function. Your program should begin by prompting the user for the values of a, b and c. Then it should display a message indicating the number of real roots, along with the values of the real roots (if any).

Exercise IX. The edit distance between two strings is a measure of their similarity—the smaller the edit distance, the more similar the strings are with regard to the minimum number of insert, delete and substitute operations needed to transform one string into the other.

Consider the strings *kitten* and *sitting*. The first string can be transformed into the second string with the following operations: Substitute the k with an s, substitute the e with an i, and insert a g at the end of the string. This is the smallest number of operations that can be performed to transform *kitten* into *sitting*. As a result, the edit distance is 3.

Write a recursive function that computes the edit distance between two strings. Use the following algorithm:

```
Let s and t be the strings
```

```
If the length of s is 0 then
    Return the length of t
Else if the length of t is 0 then
    Return the length of s
Else
    Set cost to 0
    If the last character in s does not equal the last character in t then
        Set cost to 1
    Set d1 equal to the edit distance between all characters \
       except the last one in s, and all characters in t, plus 1
    Set d2 equal to the edit distance between all characters in s, \
       and all characters except the last one in t, plus 1
    Set d3 equal to the edit distance between all characters \
       except the last one in s, and all characters \
       except the last one in t, plus cost
    Return the minimum of d1, d2 and d3
```

Use your recursive function to write a program that reads two strings from the user and displays the edit distance between them.

Exercise X. Follow the steps:

- Create a class, `Triangle`. Its `__init__()` method should take `self`, `angle1`, `angle2`, and `angle3` as arguments. Make sure to set these appropriately in the body of the `__init__()` method.
- Create a variable named `number_of_sides` and set it equal to 3.
- Create a method named `check_angles`. The sum of a triangle's three angles is It should return `True` if the sum of `self.angle1`, `self.angle2`, and `self.angle3` is equal 180, and `False` otherwise.
- Create a variable named `my_triangle` and set it equal to a new instance of your `Triangle` class. Pass it three angles that sum to 180 (e.g. 90, 30, 60).
- Print out `my_triangle.number_of_sides` and print out `my_triangle.check_angles()`.