

La librarie numpy

Table des matières

I. Installation de Numpy et statistiques de bases	3
II. Exercice : Quiz	7
III. Méthodes et fonctions	8
IV. Exercice : Quiz	11
V. Essentiel	12
VI. Auto-évaluation	13
A. Exercice	13
B. Test	13
Solutions des exercices	14

I. Installation de Numpy et statistiques de bases

Durée : 1 h

Environnement de travail : PC connecté à Internet

Contexte

Python est un langage de programmation qui devient de plus en plus populaire. Selon des sondages faits par le site Stackoverflow : « *Depuis quelques années, on constate une montée en puissance du langage de programmation Python* ». Il se situe facilement dans la liste des langages les plus recherchés par les entreprises.

On constate l'ampleur que cela prend en voyant que les plus grandes sociétés l'utilisent comme YouTube, Facebook, Netflix et bien d'autres encore (surtout les sociétés qui souhaitent faire de l'Intelligence Artificielle ou « IA »). Cela n'arrive pas par hasard parce qu'il permet de faire des tâches diverses et variées. Mais nous allons juste en citer quelques-uns comme : les statistiques, le *machine learning*, la création de sites web, etc.

Le langage Python possède notamment une large sélection de librairies de qualité. Une librairie, aussi appelée « *module* », est un ensemble de fonctions, de classe, etc. Prédéfini pour faire chacune des tâches particulières.

Numpy est une librairie open source et surtout la librairie la plus populaire dans le domaine de la *data science*. Mais elle permet également la manipulation de tableaux multidimensionnels, de faire divers calculs et de faire des analyses numériques.

Numpy est un ensemble de langage C et de Python et fait partie des bibliothèques Python les plus utilisées de manière générale. Des procédés de calcul complexes sont simplifiés par Numpy avec l'utilisation de fonctions adéquates pour des tâches particulières. Elle apporte aussi des fonctions statistiques et mathématiques.

La principale fonctionnalité apportée par Numpy est : « *array* ». Ou plus précisément la structure de donnée ndarray qui est une représentation d'une liste avec Numpy. Ce tableau est assez particulier parce que toutes les données stockées dans le tableau doivent toutes avoir le même type. Il permet de faire des calculs plus rapides par rapport à Python.

Définition Installation de Numpy

Numpy est le diminutif de « *Numerical Python* ». C'est une bibliothèque du langage Python utilisée par les *data scientists* par exemple. Numpy permet de faire des calculs sur les matrices. Les calculs effectués par celle-ci sont plus rapides que celles des simples listes sur Python.

La bibliothèque Numpy n'est généralement pas fournie avec Python. Mais il se peut que vous la connaissiez et que vous ayez déjà installé celle-ci.

Méthode

Pour voir tous les modules Python qui sont installés sur votre ordinateur, il faut d'abord ouvrir l'invite de commande avec « **Windows + R** », ensuite écrire « **cmd** » et enfin, taper la touche « **Entrer** ».

Quand on arrive sur l'invite de commande. La commande à utiliser est : « *pip list* ».

Si la librairie Numpy est déjà installée sur votre ordinateur, vérifiez bien la version. Assurez-vous d'avoir la dernière version qui est la 1.24, même si la version précédente 1.23 est plus stable.

Si tel n'est pas le cas, il est nécessaire de la mettre à jour. Pour ce faire, utilisez la commande « *pip3 install numpy --upgrade* ».

Si Numpy n'est pas encore installée, il suffit de le faire avec la commande : « *pip install numpy* ».

On peut désormais commencer à utiliser Numpy avec l'IDE que l'on préfère. Il faut juste importer le module Numpy au tout début de notre code. Comme ceci :

```
1 import numpy as np
```

Remarque

Le rajout de « **as np** » après « **import numpy** » n'est pas forcément nécessaire. C'est seulement une bonne pratique qui permet de raccourcir le terme Numpy et de ne pas devoir le réécrire entièrement à chaque fois.

Les statistiques de base

La statistique est une branche des mathématiques qui consiste à étudier des phénomènes par la collecte de données, leur traitement, leur analyse, l'interprétation des résultats et leur présentation afin de rendre les données compréhensibles par tous.

Comme beaucoup de sciences, elle possède une composante théorique ainsi qu'une composante appliquée. La composante théorique est basée sur les probabilités. La statistique appliquée est utilisée dans presque tous les domaines aujourd'hui : ingénierie, management, économie, politique, biologie, informatique, etc.

« Parmi les thèmes à propos desquels les statisticiens ne sont pas d'accord se trouve la définition de leur science. », Maurice Kendall.

Avant d'entrer dans les détails, regardons d'abord quelque vocabulaire à retenir. En effet, nous allons parler de beaucoup de choses durant ce cours, il est donc nécessaire de bien formaliser notre vocabulaire au début afin de ne pas se retrouver bloqué sur des « termes techniques ».

Vocabulaire

- **La population et les individus**

L'ensemble des individus ou unités statistiques auxquels on décide de s'intéresser est appelé population. Sa taille, habituellement notée par « N », est parfois grande, ou même infinie. Le problème à l'origine de la démarche statistique est la dépendance, qu'on ne peut pas ignorer pour le choix de la population, et aussi la façon dont on décide de la traiter.

- **La statistique (ou paramètre statistique)**

La description d'un aspect par tout nombre calculé à propos d'une population (médianes, moyennes, fréquences, quartiles, déciles, espérance) s'appelle la statistique.

- **La fréquence**

C'est le rapport d'un effectif particulier d'individus à la taille de la population. Parfois, Excel confond fréquence et effectif.

- **Les variables**

Une variable est une information dont la valeur a été recueillie, observée ou mesurée sur chaque individu. Pour que l'information varie pour chaque individu, on est obligé de mettre en œuvre la variable qui est composée des données que les statisticiens peuvent étudier et de construire ses classements d'individus. Pourtant, il existe deux types de variables, celles à valeurs ordinales, et les autres à valeurs numériques.

Une variable à valeurs ordinales est composée de valeurs qui sont définies par une relation d'ordre entre les différentes catégories possibles en y trouvant un ordre naturel. Par exemple la catégorie « *super* » sera meilleure que la catégorie « *bien* », elle-même meilleure que la catégorie « *médiocre* ».

Les variables numériques, aussi appelées variables quantitatives, sont des caractéristiques quantifiables. Ses valeurs sont des nombres. Ce type de variable peut être continu ou discret. Une variable est dite continue si elle prend un nombre infini de valeurs réelles possibles à l'intérieur d'un intervalle donné. Alors qu'une variable discrète ne peut prendre qu'un nombre maximum de valeurs réelles à l'intérieur d'un intervalle qui est donné, par exemple entre 1 et 20.

- **Médiane, quartiles, déciles, centiles (seulement pour une variable ordinale ou quantitative)**

- Médiane : c'est la valeur qui sépare la moitié inférieure du array de la moitié supérieure du array.
- Quartile : c'est une suite de valeurs coupées en trois quartiles, le deuxième quartile (Q2) correspond à la même valeur de la médiane, le premier quartile (Q1) correspond lui aux valeurs inférieures à Q2, et le troisième quartile (Q3) correspond aux valeurs supérieures à Q2.
- Décile : un décile est chacune des 9 valeurs qui divisent un jeu de données, nommé de D1 à D9, le premier décile (D1) correspond aux 10 premiers pourcents.
- Centile : un centile est comme un décile mais coupé en 100, C1 correspond donc au premier pourcentage.

La variance $V(X)$ et sa racine carrée, l'écart-type

En statistique, la variabilité des valeurs d'une série statistique¹ est mesurée par l'indicateur de dispersion dont sa valeur est toujours positive et plus grande que les valeurs de la série qui sont étalées.

La variance du vecteur $X = (x_1, \dots, x_i, \dots, x_n)$ s'écrit de la façon suivante :

$$\sigma(X) = \frac{1}{n} \sum_{i=1}^n (x_i - m)^2$$

Elle peut être calculée de plusieurs manières différentes :

Exemple

Avec la fonction `pstdev()` du module `statistics` :

Il s'agit de la fonction qui calcule l'écart-type.

```
1 import statistics
2 list=[10,5,9,4]
3 a = statistics.pstdev(list)
4 print(a)
```

Résultat : 2,5495097567963922

Avec la fonction `std()` de la bibliothèque `Numpy` :

Cette fonction calcule aussi l'écart-type avec la bibliothèque `Numpy`.

```
1 import numpy as np
2 list=[18,9,0,1,4]
3 a = np.std(list)
4 print(a)
```

¹ https://fr.wikipedia.org/wiki/S%C3%A9rie_statistique

Résultat : 6,590902821313632

Espérance mathématique

En théorie des probabilités, l'espérance mathématique d'une variable aléatoire réelle est, intuitivement, la valeur que l'on s'attend à trouver, en moyenne, si l'on répète un grand nombre de fois la même expérience aléatoire. Elle se note pour une variable X : $E(X)$.

Exemple

Elle se calcule avec la fonction suivante :

```
1 import statistics
2 list = [10,5,8,6,7,3]
3 x = statistics.mean(list)
4 print(x)
```

Résultat : 6,5

La covariance

La covariance entre deux variables X et Y est un nombre (une statistique) permettant de quantifier leurs écarts conjoints par rapport à leurs espérances respectives. Elle s'écrit $cov(X,Y)$:

$$cov(X, Y) \equiv E[(X - E[X])(Y - E[Y])]$$

Exemple

Elle se calcule avec la fonction suivante :

```
1 import numpy as np
2 a = np.array([5,4,9])
3 b = np.array([7,5,2])
4 r = np.cov(a,b)[0][1]
5 print(r)
```

Résultat : -5,5

La corrélation

Étudier la corrélation entre deux variables X et Y ou plusieurs variables ou encore statistiques numériques, c'est étudier l'intensité des liaisons qui peuvent exister entre ces variables.

Dans le cas de deux variables X et Y, elle s'écrit ainsi :

$$\text{cor}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma(X)\sigma(X)}$$

Exemple

Elle se calcule avec la fonction suivante :

```
1 import numpy as np
2 a = np.array([5,4,9])
3 b = np.array([7,5,2])
4 r = np.corrcoef(a,b)[0][1]
5 print(r)
```

Résultat : -0,826033187630902

Exercice : Quiz

[solution n°1 p.15]

Question 1

Numpy est préinstallée avec Python.

- ☐ Vrai
- ☐ Faux

Question 2

Un module est un ensemble de fonctions et de classes.

- ☐ Vrai
- ☐ Faux

Question 3

Quelle est la dernière version de Numpy ?

- ☐ La version 1.23
- ☐ La version 1.21
- ☐ La version 1.24
- ☐ La version 1.23.1

Question 4

Numpy a une signification particulière.

- ☐ Vrai
- ☐ Faux

Question 5

Une fonction permettant de calculer la corrélation existe-t-elle dans statistics ?

- ☐ Vrai
- ☐ Faux

III. Méthodes et fonctions

Les tableaux

Numpy propose un nouveau type appelé « *array* ». Il est à peu près similaire aux listes que vous connaissez sûrement déjà. Sa particularité réside dans le fait que tous ses éléments doivent être du même type. Il peut être 1D, 2D, 3D, ..., nD.

Les méthodes

Méthode `.copy()` :

La méthode `copy()` sert à copier des éléments.

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau type int
4 G = np.array([9,0,1,8,6,4], int)
5 #copier le tableau G
6 G_g = G.copy(G)
7 #afficher le tableau G et G_g
8 print(G)
9 print(G_g)
```

Résultat :

[9 0 1 8 6 4]

[9 0 1 8 6 4]

Méthode `.reshape()` :

Modifie la forme d'un tableau tout en gardant le même nombre d'éléments.

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau avec 20 éléments avec une écart de 1
4 u = np.arange(20)
5 #change le tableau un tableau a 4 lignes et 5 colonnes
6 o = u.reshape(4, 5)
7 #afficher le reshape du tableau u
8 print(o)
```

Résultat :

[[0 1 2 3 4]

5 6 7 8 9]

[10 11 12 13 14]

[15 16 17 18 19]]

Méthodes .ravel() ou .flatten() :

Transformation d'un tableau à n dimensions en un tableau à 1 dimension :

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau avec 20 éléments avec un écart de 1
4 m = np.arange(20)
5 #change le tableau n en tableau a 3 lignes et 4 colonnes
6 n = np.resize(m, (3,4))
7 a = n.ravel()
8 b = n.flatten()
9 #afficher le resize du tableau
10 print(a)
11 print(b)
```

Résultat :

```
[ 0 1 2 3 4 5 6 7 8 9 10 11]
```

```
[ 0 1 2 3 4 5 6 7 8 9 10 11]
```

Méthode .transpose() :

Cette méthode permet d'échanger les axes.

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau avec 10 éléments avec un écart de 1(avec reshape (5,2))
4 u = np.arange(10).reshape(5,2)
5 #transposition du tableau
6 w = u.transpose()
7 #afficher tableau w transposer
8 print(w)
```

Résultat : [[0 2 4 6 8]

```
[1 3 5 7 9]]
```

Les fonctions**La fonction np.array() :**

C'est une fonction qui est utilisée pour la création d'un tableau.

Pour un tableau unidimensionnel :

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau type int
4 t = np.array([0,8,7,5], int)
5 #afficher le tableau t(1D)
6 print(t)
```

Résultat : [0 8 7 5]

Pour un tableau multidimensionnel :

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau type float
4 t1 = np.array([[[0,8],[4,2]], [[8,4],[1,2]]], float)
5 #afficher le tableau t(3D)
6 print(t1)
```

Résultat :

```
[[[0. 8.] [4. 2.]] [[8. 4.] [1. 2.]]]
```

Le slicing [x:y:z] :

Le slicing est le même pour les arrays et les listes. Il sert à sélectionner une partie des données dans un tableau.

Pour un tableau unidimensionnel :

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau type float
4 t3 = np.array([8,6,9,4,5,10,12,2,4], int)
5 #[1:7:2] signifie une valeur sur 2 de l'indice 1 à 7
6 print(t3[1:7:2])
```

Résultat :

```
[ 6 4 10]
```

Pour un tableau multidimensionnel :

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau type int
4 T = np.array([[8,6],[4,5],[10,12],[2,4]], int)
5 #chaque dimension est séparée par une virgule[1D,2D,3D]
6 #[:,3] signifie un tableau sur 3
7 print(T[:,3])
```

Résultat :

```
[[8 6]
```

```
[2 4]]
```

Fonction np.copy(x)

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau type int
4 G = np.array([9,0,1,8,6,4], int)
5 #copier le tableau G
6 G_g = np.copy(G)
7 #afficher le tableau G et G_g
8 print(G)
9 print(G_g)
```

Résultat :

```
[9 0 1 8 6 4]
```

```
[9 0 1 8 6 4]
```

Fonction np.arange() :

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau de valeur compris entre 0 et 5 avec 0.1 d'écart à chaque fois
4 f = np.arange(0, 5, 0.1)
5 #afficher le tableau f
6 print(f)
```

Fonction np.linspace() :

np.arange() exclue le dernier élément

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau de valeur compris entre 0 à 10 avec 11 éléments
4 y = np.linspace(0, 10, 11)
5 #afficher le tableau y
6 print(y)
```

Résultat :

[0. 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.]

Remarque

np.linspace() n'exclut aucune valeur.

Fonction np.argmax()

Elle permet de savoir où se situe l'élément le plus grand dans une liste.

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau
4 a=np.array([7,8,2,6])
5 e = np.argmax(a)
6 print(e)
```

Résultat : 1

Fonction np.logspace() :

Ressemble à peu près à np.linspace(), en revanche, elle met des écarts logarithmiques entre chaque point.

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau de valeur compris entre 2 à 20 avec 50 éléments
4 w = np.logspace(2, 20, 50)
5 #afficher le tableau w
6 print(w)
```

Fonction np.resize() :

Modifie la forme d'un tableau et aussi le nombre d'éléments.

```
1 #importer le module numpy
2 import numpy as np
3 #créer un tableau avec 20 éléments avec un écart de 1
4 p = np.arange(20)
5 #change le tableau en un tableau a 1 ligne et 5 colonnes
6 l = np.resize(p, (1,5))
7 #afficher le resize du tableau l
8 print(l)
```

Résultat :

[[0 1 2 3 4]]

Exercice : Quiz

[solution n°2 p.16]

Question 1

Une fonction permettant de copier un tableau existe dans Numpy.

- ☐ Vrai
- ☐ Faux

Question 2

Le slicing est différent pour les arrays et les simples listes de Python.

- ☐ Vrai
- ☐ Faux

Question 3

La fonction `np.linspace` permet de créer un tableau d'une valeur initiale vers une valeur finale avec un nombre précis d'éléments et exclut la dernière valeur.

- ☐ Vrai
- ☐ Faux

Question 4

La fonction `np.zeros((2, 3))` retourne :

```
1 array([[0., 0.], [0., 0.], [0., 0.]])
```

- ☐ Vrai
- ☐ Faux

Question 5

Si `x = np.arange(10)`, alors `x[-2] = 5`.

- ☐ Vrai
- ☐ Faux

V. Essentiel

De nombreuses bibliothèques sont mises à disposition des développeurs en Python. Numpy en fait partie. Numpy signifie Numerical Python. La librairie est surtout populaire auprès des datas scientists. Elle est presque présente dans la plupart des écosystèmes en data science.

Pour pouvoir l'utiliser, il est nécessaire de l'installer. Vous pouvez vérifier avec la commande « `pip list` » si la bibliothèque Numpy est déjà installée. Sinon, il faut utiliser la commande « `pip3 install numpy` » ou « `pip install numpy` ». Ces commandes sont à exécuter dans l'invite de commandes ou dans le terminal Python. Après l'installation terminée, vous pouvez commencer à l'utiliser, mais il ne faut pas oublier de l'importer au début de chaque code.

Un de ses avantages est que c'est une librairie open source.

Elle permet de faire de nombreux calculs. Comme les calculs mathématiques de base (ex. : sinus, log, etc.). De plus, Numpy permet de faire des opérations sur les tableaux, que ce soient des tableaux multidimensionnels, unidimensionnels ou des matrices. Numpy introduit également la facilité de gestion des tableaux. Cette librairie a aussi beaucoup d'autres utilités autres que le calcul, comme stocker, manipuler et charger des données pouvant venir de sources différentes. Par exemple, une image peut être considérée comme un tableau à 2 dimensions (une matrice).

Numpy est utilisée dans de nombreux domaines comme le machine learning... Même si d'une certaine manière les arrays ressemblent un peu aux listes Python, Numpy permet de faire des calculs plus rapides, surtout avec les tableaux de grandes tailles. Mais ce type a aussi ses propres points faibles. Par exemple, il ne peut stocker qu'un seul type de données. Et comme le langage Python est là depuis très longtemps, il est facile de trouver des solutions lors des bugs rencontrés lors du développement.

La dernière version disponible actuellement est la 1.24 qui est sortie très récemment.

VI. Auto-évaluation

A. Exercice

Voici une matrice m :

```
1 #importer le module numpy
2 import numpy as np
3 m=np.array([[1,2,3,4,5,6,7,8,9,10],[11,12,13,14,15,16,17,18,19,20],
             [21,22,23,24,25,26,27,28,29,30],[31,32,33,34,35,36,37,38,39,40],
             [41,42,43,44,45,46,47,48,49,50],[51,52,53,54,55,56,57,58,59,60],
             [61,62,63,64,65,66,67,68,69,70],[71,72,73,74,75,76,77,78,79,80]])
```

Question 1

[solution n°3 p.17]

Récupérer les éléments de la 1^{ère} colonne de la 1^{ère} ligne à la dernière ligne incluse.

Question 2

[solution n°4 p.17]

Récupérer tous les éléments de la 5^e ligne avec la dernière ligne exclue :

Question 3

[solution n°5 p.17]

Récupérer la ligne 6 de la 6^e à la 7^e colonne.

Question 4

[solution n°6 p.17]

Récupérer la première ligne à la 4^{ème} ligne incluse, de la deuxième colonne à la 7^{ème} colonne.

B. Test

Exercice 1 : Quiz

[solution n°7 p.17]

Question 1

La ndarray est une liste de tuple.

- ☐ Vrai
- ☐ Faux

Question 2

Une fonction qui permet de connaître la position du plus grand élément d'une liste existe dans Numpy.

- ☐ Vrai
- ☐ Faux

Question 3

Les tableaux multidimensionnels de Numpy ont la même vitesse de calcul que les listes Python.

- ☐ Vrai
- ☐ Faux

Question 4

Numpy propose de nombreuses fonctions mathématiques de base.

- ☐ Vrai
- ☐ Faux

Question 5

La fonction `np.logspace` est similaire à `np.linspace`.

- ☐ Vrai
- ☐ Faux


Solutions des exercices

Exercice p. 7 Solution n°1**Question 1**

Numpy est préinstallée avec Python.

☐ Vrai

☒ Faux


 Si vous n'avez jamais utilisé la bibliothèque Numpy auparavant, il va falloir l'installer avec la commande : « **pip install numpy** ».

Question 2

Un module est un ensemble de fonctions et de classes.

☒ Vrai

☐ Faux

 Un module est un ensemble de fonctions et de classes utilisé pour effectuer des tâches particulières.

Question 3


Quelle est la dernière version de Numpy ?

☐ La version 1.23

☐ La version 1.21

☒ La version 1.24

☐ La version 1.23.1


 La dernière version de Numpy est la 1.24 mais elle est sortie très récemment. Pour l'instant il est conseillé d'utiliser la 1.23.1 ou la 1.23.

Question 4

Numpy a une signification particulière.

☒ Vrai

☐ Faux

 Numpy est le diminutif de « *Numerical Python* ».

Question 5

Une fonction permettant de calculer la corrélation existe-t-elle dans statistics ?

☒ Vrai

☐ Faux

 Pour cela on peut utiliser la fonction :

```
1 import statistics
2 statistics.corrccoef()
```


Exercice p. 11 Solution n°2

Question 1

Une fonction permettant de copier un tableau existe dans Numpy.

☒ Vrai

☐ Faux

 Pour cela on peut utiliser la fonction :


```
1 import numpy as np
2 np.arange()
```

Question 2

Le slicing est différent pour les arrays et les simples listes de Python.

☐ Vrai

☒ Faux


 Le slicing est similaire pour les arrays et les simples listes. Il permet de sélectionner une partie des données dans un tableau.

Question 3

La fonction np.linspace permet de créer un tableau d'une valeur initiale vers une valeur finale avec un nombre précis d'éléments et exclut la dernière valeur.

☐ Vrai

☒ Faux

 Lors de la création du tableau, aucune valeur n'est exclue. Même la dernière.


Question 4

La fonction np.zeros((2, 3)) retourne :

```
1 array([[0., 0.], [0., 0.], [0., 0.]])
```

☐ Vrai

☒ Faux

 La bonne réponse est :


```
1 array([[ 0., 0., 0.], [ 0., 0., 0.]])
```

Question 5

Si $x = \text{np.arange}(10)$, alors $x[-2] = 5$.

☐ Vrai

☒ Faux

 Si $\text{np.arange}(10)$, alors $x[-2] = 8$

p. 13 Solution n°3

```
1 b = m[0:0]
2 print(b)
```

p. 13 Solution n°4

```
1 a = m[4:-1, :]
2 print(a)
```

p. 13 Solution n°5

```
1 c = m[5,5:7]
2 print(c)
```

p. 13 Solution n°6


```
1 d = m[:,4,1:6]
2 print(d)
```

Exercice p. 13 Solution n°7**Question 1**

La ndarray est une liste de tuple.

☐ Vrai

☒ Faux


 C'est un moyen pour représenter les listes multidimensionnelles avec Numpy.

Question 2

Une fonction qui permet de connaître la position du plus grand élément d'une liste existe dans Numpy.

☒ Vrai

☐ Faux

 La fonction qui permet de faire cela est :


```
1 import numpy as np
2 np.argmax()
```

Question 3

Les tableaux multidimensionnels de Numpy ont la même vitesse de calcul que les listes Python.

☐ Vrai

☒ Faux


 Les arrays de Numpy sont plus rapides que les listes mais elles ne peuvent stocker que le même type de données.

Question 4

Numpy propose de nombreuses fonctions mathématiques de base.

☒ Vrai

☐ Faux


 Par exemple : `np.cos(array)`, `np.sin(array)`, `np.exp(array)`, `np.log(array)`.

Question 5

La fonction `np.logspace` est similaire à `np.linspace`.

☒ Vrai

☐ Faux

 La seule différence est qu'elle a un espacement logarithmique entre chaque point.