

Introduction à la Science des données

Travail pratique 02 – Outils pour le calcul scientifique I

Professeurs: Andres Perez-Uribe & Carlos Peña
Assistants: Axel Fahy, Shabnam Ataee, Xavier Brochet
Emails : prenom.nom@heig-vd.ch

Objectifs :

- Se familiariser avec les bibliothèques Numpy et Matplotlib permettant la manipulation de structures de données de type array et la visualisation de données.

I. Exercices Numpy

1. Créez un array avec les entiers 0 à 10 et générez en sortie un array où les valeurs impaires de l'array ont été remplacées par des -1.

Sortie désirée : `array([0, -1, 2, -1, 4, -1, 6, -1, 8, -1])`

2. Créez un array avec les entiers de 0 à 100 et générez en sortie un array contenant seulement les nombres premiers.

Aide: commencez par créer une fonction qui calcule si un nombre entier est un nombre premier (c.a.d., qu'il n'est pas divisible par 2,3,4,5.... ni par \sqrt{n} , où n est le nombre testé).

3. Soit la matrice donnée ci-dessous, calculez la somme des éléments de chaque ligne et de chaque colonne en utilisant la méthode `numpy.sum()`

`MS = [[20, 12, 4, 45, 37, 29, 28], [11, 3, 44, 36, 35, 27, 19], [2, 43, 42, 34, 26, 18, 10], [49, 41, 33, 25, 17, 9, 1], [40, 32, 24, 16, 8, 7, 48], [31, 23, 15, 14, 6, 47, 39], [22, 21, 13, 5, 46, 38, 30]]`

4. Générez une matrice de dimensions 10x10 contenant des valeurs aléatoires entre -1.0 et 1.0. Calculez la valeur maximale, minimale et moyenne. Calculez la position où se trouve la valeur maximale et minimale (`argmax` et `argmin`) et la position du minimum par ligne (`axis=1`).

5. Créez une matrice x de dimensions 10x10 contenant des valeurs aléatoires entre 0.0 et 1.0 et calculez le carré de chaque élément. Comparez le temps de calcul (p.ex. utilisez la magic command %%timeit) d'une implémentation qui utilise une double boucle for et une implémentation qui fait le calcul demandé tout simplement ainsi : $x = x ** 2$
6. Créez un array de dimension 100 avec des valeurs aléatoires entre 0 et 1.0 et générez un nouvel array où les éléments plus petits que 0.5 sont élevés au carré. Comparez le temps de calcul a) d'une implémentation à l'aide d'une boucle for, et b) d'une implémentation qui utilise la méthode np.where().

II. Exercices Matplotlib

7. Générez un graphique de la fonction tangente hyperbolique définie par :

$$\tanh(x) = (1 - \exp(-2 * x)) / (1 + \exp(-2 * x)), \text{ pour } -5 < x < 5$$

Ajouter un titre, des labels et une grille.

8. Générez un graphique de type « vertically stacked sub-plot » des fonctions tangente hyperbolique et sigmoïde (ou fonction logistique) pour des valeurs de x entre $-5 < x < 5$. Ces fonctions sont définies par les expressions :

$$\tanh(x) = (1 - \exp(-2 * x)) / (1 + \exp(-2 * x))$$

$$\text{sigm}(x) = 1 / (1 + \exp(-x))$$

Ajouter un titre, des labels et une grille.

9. Générez un graphique de la courbe paramétrique ci-dessous pour les valeurs de t entre $-3.5 < t < 35$ et les valeurs de constantes : $R1 = 1.1$, $w1 = 1.1$, $R2 = 0.8$, $w2 = 1.7$, $fi = 1.2$, $kfi = 0.008$, et $k2 = -0.012$

$$x(t) = R1 * \cos(2 * \pi * w1 * t) + R2 * \sin(2 * \pi * w2 * t + fi * e(kfi * t)) * e(k2 * t)$$

$$y(t) = R1 * \sin(2 * \pi * w1 * t) + R2 * \cos(2 * \pi * w2 * t + fi * e(kfi * t)) * e(k2 * t)$$

Pour que la figure sorte plus jolie, vous pouvez indiquer la taille de celle-ci avec la méthode : plt.figure(figsize=(12,12))

On vous invite à explorer différentes valeurs de constantes ☺

10. Bar chart vs pie chart.

- a) Générez un pie chart à l'aide du code ci-dessous et comparez les valeurs de prévalence d'obésité de la population en Allemagne, Angleterre et Canada. D'après la figure, dans quel pays il y a plus d'obésité ?
- b) Générez un bar chart avec les mêmes données, ajouter un titre et des labels et faites la même analyse proposée en A. Qu'est-ce que vous pouvez conclure ?

```
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline

pays = ['USA', 'DE', 'UK', 'CA']
obesity = [30, 16, 20, 18]
x = np.arange(len(pays))
plt.pie(obesity, labels=pays)
```

III. Numpy niveau expert

Cette partie est optionnelle. Numpy offre une palette gigantesque de fonctions pour le traitement des arrays. Si vous voulez devenir un pro en Numpy, essayez cette série d'exercices :

<https://www.machinelearningplus.com/python/101-numpy-exercises-python/>

IV. Calcul d'un histogramme en Python

Programmez une fonction en Python qui reçoit en entrée un numpy array et donne en sortie un array indiquant des fréquences de valeurs. Supposez que l'histogramme génère par défaut la fréquence des valeurs dans 10 bins espacées de manière uniforme entre le min et le max des valeurs de l'array numpy en entrée. Générez un bar chart pour visualiser l'histogramme. Comparez votre résultat avec la fonction hist() de numpy.

Rapport

Pas de rapport.