

Exercices sur la bibliothèque Numpy

Exercice 1:

En statistique, le Median Absolute Deviation (MAD) est un estimateur robuste de la dispersion d'une matrice M est : $MAD = \text{median}(|M - \text{median}(M)|)$.

À l'aide des fonctions `numpy.median()` et `numpy.abs()`, écrire une fonction `mad(M, axis=None)` calculant le MAD d'un array 2D (nommé M), éventuellement le long d'un de ses axes.

N'oubliez pas que l'opération de différence est permise soit entre une matrice et un nombre entier ou réel (exp: $x - n$), soit entre deux matrices ayant la **même dimension**.

Exemple d'utilisation de la fonction `numpy.median()`:

```
In [17]: import numpy as np

In [18]: a = np.array([[10, 7, 4], [3, 2, 1]])
...: np.median(a)
Out[18]: 3.5

In [19]: np.median(a, axis=0)
...:
Out[19]: array([6.5, 4.5, 2.5])

In [20]: np.median(a, axis=1)
Out[20]: array([7., 2.]) # donne un tableau 1D

In [21]: np.median(a, axis=1, keepdims=True)
Out[21]:
array([[7.],
       [2.]]) # donne un tableau ayant la même dimension
               que le tableau a
```

Exercice 2:

- 1) Définissez la fonction `random_mat(N)` qui permet de générer et retourner une matrice $M(N \times 2)$ contenant N lignes et deux colonnes X_i et Y_i qui vérifient :
 - X_i suit une loi normale de moyenne 5 et de variance 25.
 - $Y_i = 2X_i$
- 2) Définissez la fonction `build_m(M, N, n=20)` qui choisit aléatoirement n valeurs dans la matrice M (par défaut $n=20$) et les remplace par `numpy.nan`. La matrice retournée par la fonction est une matrice contenant n valeurs nulles.
Pour choisir la position (i, j) de la valeur à remplacer, pensez à utiliser la fonction `numpy.random.randint(val)` qui retourne un entier aléatoire entre 0 et `val`.
- 3) Définissez la fonction `nettoyage(M)` qui permet de modifier chaque valeur manquante dans M par la médiane des valeurs de la colonne dont elle appartient. La matrice retournée par la fonction est une matrice sans valeurs nulles.
- 4) Testez les fonctions dans le programme principal et afficher leurs résultats