Aide-mémoire PYTHON

Seules les fonctions présentes dans ce document ou documentées dans l'énoncé sont utilisables par les candidats.

Listes

`[]`	Créer une liste vide
`[a]*n`	Créer une liste avec n fois l'élément `a`
`L.append(a)`	Ajoute l'élément `a` à la fin de la liste `L`
`L1 + L2`	Concatène les deux listes `L1` et `L2`
`len(L)`	Renvoie le nombre d'éléments de la liste `L`
`L.pop(k)`	Renvoie le $k^{\text{ème}}$ élément de la liste `L` et l'enlève de `L`
`L.count(a)`	Renvoie le nombre d'occurences de `a` dans la liste `L`
`L.remove(a)`	Enlève une fois la valeur `a` de la liste `L`
`max(L)`	Renvoie le plus grand élément de la liste `L`
`min(L)`	Renvoie le plus petit élément de la liste `L`
`sum(L)`	Renvoie la somme de tous les éléments de la liste `L`
`del(L)`	Supprime la liste `L` de la mémoire
`a in L`	Vaut `True` si `a` se trouve au moins une fois dans `L` et `False` sinon

Modules mathématiques `numpy` et `math`

```
import numpy as np, math as m
```

`np.array(L)`	Transforme la liste `L` en vecteur ou matrice `numpy`
`np.linspace(a,b,n)`	Crée un vecteur de n valeurs uniformément réparties entre a et b (inclus)
`np.zeros([n,m])`	Crée la matrice nulle de taille $n \times m$
`np.ones([n,m])`	Crée la matrice de taille $n \times m$ dont tous les coefficients valent 1
`np.eye(n)`	Crée la matrice identité de taille n
`np.diag(L)`	Crée la matrice diagonale dont les termes diagonaux sont les éléments de la liste `L`
`np.transpose(M)`	Renvoie la transposée de M
`np.dot(M,P)`	Renvoie le produit matriciel MP
`np.sum(M)`	Renvoie la somme de tous les éléments de M
`np.prod(M)`	Renvoie le produit de tous les éléments de M
`np.max(M)`	Renvoie le plus grand élément de M
`np.min(M)`	Renvoie le plus petit élément de M
`np.shape(M)`	Renvoie dans un couple le format de la matrice M
`np.mean(M)`	Renvoie la moyenne des éléments de M
`np.var(M)`	Renvoie la variance des éléments de M
`np.std(M)`	Renvoie l'écart-type des éléments de M
`np.median(M)`	Renvoie la médiane des éléments de M
`np.cumsum(M)`	Renvoie la matrice des sommes cumulées (gauche $\to$ droite, haut $\to$ bas) des éléments de M
`np.arange(a,b,eps)`	Renvoie la liste des flottants de a à b de pas constant eps
`np.abs(x)`	Renvoie $\lvert x \rvert$
`np.floor(x)`	Renvoie $\lfloor x \rfloor$
`np.sqrt(x)`	Renvoie $\sqrt{x}$ si $x \geq 0$
`np.log(x)`	Renvoie $\ln(x)$ si $x > 0$
`np.exp(x)`	Renvoie e^x
`np.e`	Renvoie e
`np.pi`	Renvoie π
`m.factorial(k)`	Renvoie $k!$

Sous module d'algèbre linéaire `linalg` de `numpy`

```
import numpy.linalg as al
```

`al.inv(M)`	Renvoie l'inverse de la matrice carrée M si elle est inversible
`al.eig(M)`	Si la matrice M est diagonalisable, renvoie un couple `D,P` où `D` est le vecteur des coefficients diagonaux d'une matrice diagonale semblable à M et `P` une matrice de passage associée
`al.matrix_power(M,n)`	Renvoie la puissance n-ième de la matrice carrée M
`al.matrix_rank(M)`	Renvoie le rang de la matrice M

Sous module `random` de `numpy` pour la simulation probabiliste

```
import numpy.random as rd
```

`rd.random([r,s])`	Simule une réalisation d'une matrice (r, s) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi uniforme $\mathcal{U}([0, 1])$
`rd.randint(a,b,[r,s])`	Simule une réalisation d'une matrice (r, s) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi uniforme discrète $\mathcal{U}(\llbracket a, b-1 \rrbracket)$
`rd.binomial(n,p,[r,s])`	Simule une réalisation d'une matrice (r, s) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi binomiale $\mathcal{B}(n, p)$
`rd.geometric(p,[r,s])`	Simule une réalisation d'une matrice (r, s) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi géométrique $\mathcal{G}(p)$
`rd.poisson(a,[r,s])`	Simule une réalisation d'une matrice (r, s) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi de Poisson $\mathcal{P}(a)$
`rd.exponential(a,[r,s])`	Simule une réalisation d'une matrice (r, s) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi exponentielle $\mathcal{E}(\frac{1}{a})$
`rd.normal(m,d,[r,s])`	Simule une réalisation d'une matrice (r, s) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi normale $\mathcal{N}(m, d^2)$

Si le paramètre `[r,s]` est remplacé par `r`, ces fonctions renvoient la réalisation d'un vecteur de longueur `r` correspondant à la loi en question, et si ce paramètre est omis, elles renvoient un seul coefficient suivant les mêmes contraintes.

Sous module graphique `pyplot` de `matplolib`

```
import matplotlib.pyplot as plt
```

`plt.plot(X,Y,options)`	Crée la courbe des points définis par les listes `X`, abscisses, et `Y`, ordonnées suivant les options graphiques définies par la chaîne de caractères `options`
`plt.bar(X,Y)`	Crée le diagramme en bâtons défini par les listes ou vecteurs `X`, abscisses et `Y`, ordonnées
`plt.hist(X,Y)`	Crée l'histogramme des valeurs définies par la liste `X`, `Y` étant soit le nombre de classes, soit les bornes des classes
`plt.boxplot(X)`	Génère le diagramme en boite basé sur le vecteur de données `X`
`plt.axis('equal')`	Rend le repère orthonormé
`plt.xlim(xmin,xmax)`	Fixe les bornes de l'axe des abscisses
`plt.ylim(ymin,ymax)`	Fixe les bornes de l'axe des ordonnées
`plt.show()`	Affiche le graphique

Module `pandas` de traitement de données

```
import pandas as pd
```

`pd.read_csv('data.csv')`	Crée un tableau Pandas à partir du fichier `data.csv`, les colonnes étant nommées par les valeurs se trouvant dans la première ligne du fichier csv
`df.head(n)`	Crée le sous-tableau Pandas constitué des n premières lignes du tableau Pandas `df` déjà créé
`df.shape`	Désigne le couple nombre le lignes, nombre de colonnes du tableau Pandas `df`
`df.mean`	Désigne le vecteur des moyennes de chaque colonne du tableau Pandas `df`
`df.std`	Désigne le vecteur des écart-types de chaque colonne du tableau Pandas `df`
`df.median`	Désigne le vecteur des médianes de chaque colonne du tableau Pandas `df`
`df.count()`	Renvoie un tableau indiquant le nom des colonnes du tableau Pandas `df` et les effectifs de chacune d'elles
`len(df)`	Renvoie le nombre de lignes du tableau Pandas `df`
`df.describe()`	Renvoie un tableau dont les lignes sont, dans l'ordre, les effectifs, les moyennes, les écart-types, le minimum, les quartiles et le maximum, de chaque colonnes du tableau Pandas `df`.
`df.sort_values(by='col')`	Renvoie le résultat du tri du tableau Pandas `df` suivant la colonne `col` par ordre croissant `df.sort_values(by='col',ascending=False)` pour le tri du tableau par ordre décroissant