

Enoncés des exercices de la série Numpy

Rappels

Code

- Ecrire un code propre
- Attention aux noms des variables, fonctions
- Un commentaire pour decrire vos blocs de code ou lignes importantes
- N'hesitez pas à faire des fonctions, essayez de simplifier le rôle de votre fonction : généralement une fonction == un objectif
- Essayez d'abord de faire un code fonctionnel, ensuite le simplifier et le clarifier

Figure

- Une figure doit toujours contenir un titre pour chaque axe
- Si plusieurs lignes sur votre figure, ajoutez une legende
- N'hesitez pas à specifier des bornes
- Attention aux choix des couleurs
- Lorsque vous sauvegardez votre figure au format **.jpg** ou **.png**, ajoutez l'option dpi (densite de pixels)

```
plt.savefig('ma_figure.png', format='png', dpi=200)
```

Exercice 1 : entrainement avec numpy

Réalisez les tâches suivantes:

- Créer une matrice 5 x 5 de zéros
- Créer une matrice de nombres entiers aléatoires de taille 10 x 10
- Créer un vecteur contenant 100 éléments compris entre 0 et 10; appliquer la fonction power avec comme puissance 3/
- Tracer avec matplotlib le résultat obtenu en fonction du vecteur source.
- Ouvrir le fichier sensor.dat avec la librairie numpy
- Tracer la colonne 1 en fonction de la colonne 2

Exercice 2 : estimation du nombre pi

1. Numpy

Objectif : estimer le nombre pi.

Etapes :

- On definit les dimensions d'un domaine rectangulaire 2 m x 2 m
- On definit un cercle centré dans ce rectangle avec un rayon de 1 m , ce cercle doit être contenu dans le domaine
- Maintenant on tire la position (x,y) de n points dans le domaine rectangulaire
- On compte le nombre de points, *ndisque*, contenu dans le cercle

Le nombre de points dans le cercle est proportionel à la surface du disque. On a ainsi la relation suivante : $\frac{S_{disque}}{S_{domaine}} = \frac{n_{disque}}{n_{domaine}}$

En deduire une valeur de pi

1. Matplotlib

Tracer les deux figures suivantes :

Figure 1 :

- Tracer les points tirés aléatoirement pour un nombre de tirage relativement faible (inutile de surcharger la figure)
- Tracer les points détectés à l'intérieur du cercle
- Tracer le cercle (trouver la fonction matplotlib qui permet de dessiner un cercle)

Figure 2 :

- Lancez votre programme pour differentes valeurs du nombre de tirage n, enregistrez dans un fichier **.txt** la valeur du nombre pi associe à la valeur n
- Tracez la courbe $\pi_{estimate} = f(n)$

Exercice 3 : analyse des données d'un coureur

- Importer le module "datetime" et le module "pandas"
- Charger le fichier "coureur1.xls" à l'aide du module pandas
- Créer une nouvelle table ne contenant que : le temps, l'altitude, la pulsation et la vitesse
- Trouver une méthode permettant d'effacer les lignes contenant un "Nan"
- Calculer la vitesse maximale et minimale atteinte par le coureur
- Créer 3 nouvelles colonnes à partir de la colonne temps: une contenant les heures, une autre les minutes et une dernière les secondes (n'hésitez pas à utiliser la fonction apply)
- Créer une nouvelle colonne de type datetime.time
- Tracer la vitesse en fonction du temps
- Tracer l'altitude en fonction du temps
- Tracer la pulsation en fonction du temps
- A l'aide de la méthode des trapèze, calculer la vitesse moyenne
- Créer une fonction d'interpolation

Exercice 4 (falculatif) : analyse de données SNCF

Importer le fichier "regularite-mensuelle-tgv.csv"

Objectif : tracer le temps de trajet moyen de la ligne Rennes - Paris Montparnasse au cours des dernières années