

TP 004 Scipy énoncés

Préalable fonctionnel :

<https://www.w3schools.com/python/scipy/index.php>

Ces questions couvriront différents modules de SciPy tels que l'intégration, l'optimisation, l'algèbre linéaire, la manipulation de données statistiques, et plus encore. Ce TP est conçu pour mieux comprendre l'utilisation de SciPy.

Les objectifs pédagogiques de ce TP visent à fournir aux apprenants les compétences et connaissances nécessaires pour manipuler et analyser des données numériques en Python. Dans votre compte rendu vous devrez en quelques phrases préciser ce que chaque objectif évoque pour vous. Votre évaluation porte sur la réponse mais aussi sur la présentation de cette dernière : **les illustrations, explications, documentations détaillées et enrichissement** seront particulièrement valorisés dans la notation.

SciPy est une bibliothèque centrale dans l'écosystème Python pour la science et l'ingénierie. Elle offre des outils robustes pour diverses applications numériques. Voici quelques-uns des cas d'utilisation les plus courants de SciPy :

Optimisation : Utilisation de routines pour la minimisation de fonctions, la recherche de racines, et l'optimisation de courbes.

Traitement de signal : Filtrage, fenêtrage, et d'autres manipulations de signaux numériques.

Algèbre linéaire : Opérations sur les matrices et les vecteurs, y compris la résolution de systèmes d'équations linéaires.

Statistiques : Fonctions pour la manipulation et l'analyse de données statistiques, y compris des distributions de probabilité, des tests statistiques, et plus.

Intégration numérique : Méthodes pour l'intégration numérique de fonctions.

Interpolation : Outils pour l'interpolation de données et la création de splines.

Traitement d'images : SciPy, en conjonction avec NumPy et souvent avec des bibliothèques spécialisées comme PIL ou imageio, peut être utilisé pour effectuer des opérations telles que le filtrage, la morphologie, les transformations, et plus.

Transformation de Fourier : Pour analyser les composantes fréquentielles des signaux.

En ce qui concerne les questions spécifiques sur la manipulation d'images, de sons ou de vidéos :

- Manipulation d'images : SciPy est très utilisé pour manipuler des images, particulièrement avec le sous-module `scipy.ndimage`, qui fournit des fonctions de filtrage, de morphologie, de mesure, et d'autres transformations d'images.

En résumé, bien que SciPy soit extrêmement puissant pour les applications numériques, ses capacités dans le traitement audiovisuel sont limitées, et d'autres bibliothèques sont souvent préférables pour des tâches spécifiques dans ces domaines.

Questions pratiques sur SciPy

1. Lecture de données

Question 1: Utilisez `scipy.io.loadmat` pour lire un fichier MATLAB `.mat` fictif contenant les variables `x` et `y`. (Supposez que le fichier s'appelle `data.mat`).

2. Filtrage de base

Question 2: Appliquez un filtre médian simple avec `scipy.ndimage.median_filter` sur un tableau numpy `np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])`.

3. Redimensionnement d'image

Question 3: Utilisez `scipy.ndimage.zoom` pour redimensionner une image (représentée par un tableau 2D) de taille (10,10) pour doubler sa taille.

4. Génération de nombres aléatoires

Question 4: Générez une matrice 3x3 de nombres aléatoires tirés d'une distribution normale en utilisant `scipy.stats.norm`.

5. Calcul de distance

Question 5: Calculez la distance euclidienne entre les points (1, 2) et (4, 6) en utilisant `scipy.spatial.distance.euclidean`.

6. Normalisation des données

Question 6: Normalisez un tableau de valeurs `np.array([1, 2, 3, 4, 5])` à l'aide de la méthode z-score (`scipy.stats.zscore`).

7. Calcul des percentiles

Question 7: Calculez le 50ème percentile (médiane) des données dans l'array `np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])` en utilisant `scipy.stats.scoreatpercentile`.

8. Détermination du minimum et maximum

Question 8: Trouvez les valeurs minimales et maximales dans un tableau numpy `np.array([10, 20, 30, 40, 50])` et affichez les indices de ces valeurs.

9. Opérations matricielles simples

Question 9: Multipliez deux matrices `A = np.array([[1, 2], [3, 4]])` et `B = np.array([[2, 0], [1, 2]])` en utilisant `scipy.linalg.matmul`.

10. Trouver les racines d'un polynôme

Question 10: Trouvez les racines du polynôme

$$x^2 - 4x + 4$$

en utilisant `scipy.roots`.

TP SciPy : Manipulation de Données

11. Importation de données CSV *

Question 11: Utilisez `numpy.genfromtxt` pour charger un fichier CSV fictif nommé `data.csv` qui contient deux colonnes de nombres.

12. Calcul de moyenne mobile

Question 12: Calculez la moyenne mobile sur 3 points pour le tableau `np.array([10, 20, 30, 40, 50, 60, 70])` en utilisant `scipy.ndimage.uniform_filter1d`.

13. Filtrage de données

Question 13: Utilisez `scipy.signal.medfilt` pour appliquer un filtre médian sur le tableau `np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])` avec une taille de kernel de 3.

14. Normalisation Min-Max *

Question 14: Normalisez les données du tableau `np.array([10, 20, 30, 40, 50])` en utilisant la normalisation Min-Max (les valeurs doivent être ajustées entre 0 et 1).

15. Détecter et remplacer les valeurs NaN *

Question 15: Remplacez toutes les valeurs NaN par la moyenne des autres valeurs dans le tableau `np.array([1, 2, np.nan, 4, 5])`.

16. Détecter des outliers *

Question 16: Détectez les outliers dans le tableau `np.array([1, 2, 3, 100, 5, 6, 7, 8])` en utilisant la méthode des écarts interquartiles (IQR).

17. Extraction de caractéristiques *

Question 17: Calculez la médiane, la moyenne, l'écart-type et la variance du tableau

`np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])`.

18. Corrélation entre données

Question 18: Calculez le coefficient de corrélation de Pearson entre deux tableaux `np.array([1, 2, 3, 4, 5])` et `np.array([5, 4, 3, 2, 1])`.

19. Tri et rangs *

Question 19: Triez le tableau `np.array([40, 10, 30, 20])` et déterminez les rangs de chaque élément.

20. Élimination des doublons *

Question 20: Éliminez les doublons du tableau `np.array([1, 2, 2, 3, 4, 4, 5])` pour ne garder que les éléments uniques.

TP SciPy : Manipulation de Données avec SciPy

21. Filtrage Gaussien

Question 21: Utilisez `scipy.ndimage.gaussian_filter` pour appliquer un filtrage gaussien sur le tableau `np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])` avec un sigma de 1.

22. Interpolation de données

Question 22: Interpolez les valeurs manquantes dans la série `np.array([1, 2, np.nan, 4, 5])` en utilisant `scipy.interpolate.interp1d`.

23. Détection de pointes (spike detection)

Question 23: Détectez les pointes dans un signal `np.array([1, 2, 1, 3, 1, 5, 1, 6, 1])` en utilisant `scipy.signal.find_peaks`.

24. Densité de probabilité

Question 24: Calculez la fonction de densité de probabilité pour un ensemble de données normalement distribuées en utilisant `scipy.stats.norm.pdf` sur l'array `np.array([1, 2, 3, 4, 5])`.

25. Test statistique

Question 25: Réalisez un test t de Student pour vérifier si deux ensembles de données `np.array([1, 2, 3, 4, 5])` et `np.array([2, 3, 4, 5, 6])` ont des moyennes statistiquement différentes, en utilisant `scipy.stats.ttest_ind`.

TP SciPy et NumPy : Traitement d'Image

Question 1: Lecture et affichage d'image

Question : Chargez une image fournie (nommée `image.png`) en utilisant `matplotlib.pyplot.imread` et affichez l'image.

Question 2: Conversion en niveaux de gris

Question : Convertissez l'image couleur en niveaux de gris en utilisant une pondération pour les composantes RGB et affichez le résultat.

Question 3: Filtrage spatial

Question : Appliquez un filtre médian à l'image en niveaux de gris pour réduire le bruit et affichez l'image filtrée.

Question 4: Détection de contours

Question : Détectez les contours de l'image filtrée en utilisant le filtre de Sobel et affichez l'image des contours.

Question 5: Réduction de dimension

Question : Réduisez la résolution de l'image en niveaux de gris d'un facteur 2 dans chaque dimension en utilisant la sous-échantillonnage et affichez l'image résultante.

Question 6: Histogramme des intensités

Question : Calculez et affichez l'histogramme des intensités de l'image en niveaux de gris.

TP avancé / Data sciences SciPy

1. Intégration numérique

Question 1: Utilisez `scipy.integrate.quad` pour calculer l'intégrale de la fonction

$$f(x) = e^{-x^2}$$

de 0 à 1.

2. Résolution d'équations différentielles

Question 2: Résolvez l'équation différentielle

$$y' = y, y(0) = 1$$

de 0 à 2 en utilisant `scipy.integrate.solve_ivp`.

3. Optimisation

Question 3: Trouvez le minimum de la fonction

$$f(x) = (x - 3)^2$$

en utilisant `scipy.optimize.minimize`.

4. Ajustement de courbe

Question 4: Ajustez une ligne droite $y=ax+b$ aux points donnés

(1,2),(2,3),(3,5)

(1,2),(2,3),(3,5) en utilisant `scipy.optimize.curve_fit`.

5. Algèbre linéaire

Question 5: Résolvez le système d'équations linéaires $3x+4y=5$ et $2x-y=1$ en utilisant `scipy.linalg.solve`.

6. Traitement de signal

Question 6: Filtrez un signal simple en utilisant un filtre passe-bas Butterworth avec `scipy.signal.butter` et `scipy.signal.filtfilt`.

7. Statistiques

Question 7: Calculez la médiane, la moyenne et l'écart-type d'un échantillon aléatoire normal avec `scipy.stats`.

8. Interpolation

Question 8: Interpolez un ensemble de points pour estimer une fonction non-linéaire à l'aide de `scipy.interpolate.interpld`.

9. Transformation de Fourier

Question 9: Effectuez la transformation de Fourier rapide d'un signal simple avec `scipy.fft.fft` et visualisez les amplitudes.

10. Équations non linéaires

Question 10: Résolvez l'équation non linéaire

$$x^2 - 2x - 8 = 0$$

en utilisant `scipy.optimize.root`.

QUESTIONS DE FOND ET COMPLÉMENTS ETUDE DE CAS

Exercice 1 : Poser puis résoudre 3 problèmes avec scipy. Exposer 2 résolutions a votre voisin / binome.

Exercice 2 : Commentez 3 codes sources utilisant Scipy