

cheatsheetNumPy

November 16, 2024

Impoter NumPy

```
import numpy as np
```

Créer un tableau

```
tableau = np.array([1, 2, 3, 4]) # Tableau à 1 dimension
matrice = np.array([[1, 2], [3, 4]]) # Tableau à 2 dimensions
```

Créer des tableaux spéciaux

```
np.zeros((2, 2)) # Tableau de zéros
np.ones((2, 2)) # Tableau de uns
np.arange(0, 10, 2) # Tableau de 0 à 10 avec un pas de 2
np.linspace(0, 1, 5) # 5 nombres entre 0 et 1
```

Dimensions et forme

```
tableau.shape # Dimensions du tableau
tableau.size # Nombre total d'éléments
tableau.ndim # Nombre de dimensions
```

Changer la forme d'un tableau

```
tableau.reshape((2, 2)) # Remodeler le tableau
```

Opérations arithmétiques sur les tableaux

```
tableau + 1 # Ajouter 1 à chaque élément
tableau * 2 # Multiplier chaque élément par 2
np.sqrt(tableau) # Racine carrée de chaque élément
```

Fonctions statistiques

```
np.mean(tableau) # Moyenne
np.sum(tableau) # Somme
np.std(tableau) # Écart type
np.max(tableau) # Valeur maximale
np.min(tableau) # Valeur minimale
```

Accès aux éléments d'un tableau

```
tableau[0]          # Premier élément
tableau[-1]         # Dernier élément
```

Découpage (slicing) d'un tableau

```
tableau[1:3]        # Du deuxième au troisième élément
tableau[:, :2]       # Un élément sur deux
matrice[:, 1]        # Deuxième colonne
matrice[0, :]        # Première ligne
```

Génération de nombres aléatoires

```
np.random.rand(3, 3)      # Tableau 3x3 de nombres aléatoires
np.random.randint(0, 10, (3, 3)) # Tableau 3x3 d'entiers entre 0 et 10
```

Droite de régression linéaire et coefficients Équation et coefficients de la droite

$y = ax + b$ # a = pente et b = ordonnée à l'origine

Exemple de données

```
x = np.array([1, 2, 3, 4, 5])
y = np.array([2, 3, 5, 7, 11])
```

Calcul des coefficients de la droite

```
coefficients = np.polyfit(x, y, degree) # degree = 1 Pour une droite
a = coefficients[0] # Pente
b = coefficients[1] # Ordonnée à l'origine
```

Tracer (Matplotlib) les données et la droite de régression

```
plt.plot(x, y, 'marqueur', label='Données')
plt.plot(x, a*x + b, 'couleur', label='Régression linéaire')
```

Calcul du coefficient de corrélation

```
coef_correlation = np.corrcoef(x, y)[0, 1]
```

Interprétations

1. Observation de la droite et des données (graphique)
 - La proximité des points de données par rapport à la droite de régression montre la force de la relation linéaire. Si les points sont proches de la droite, cela indique une forte corrélation linéaire. Si les points sont dispersés, la corrélation est plus faible.
2. Les coefficients de la droite de régression linéaire
 - a : La pente de la droite de régression indique le taux de changement de la variable dépendante (Y) par rapport à la variable indépendante (X). Si la pente est positive, cela signifie qu'il y a une tendance à la hausse de Y en fonction de X. Si la pente est négative, cela indique une tendance à la baisse.
 - b : Ordonnée à l'origine. Elle représente la valeur de Y lorsque X est égal à zéro.

3. Le coefficient de corrélation

- La valeur de coefficient varie entre -1 et 1:
 - Si $= 1$ Indique une corrélation linéaire positive parfaite.
 - Si $= -1$ Indique une corrélation linéaire négative parfaite.
 - Si $= 0$ Indique qu'il n'y a pas de corrélation linéaire.
- Interprétation du coefficient:
 - Si > 0 Les variables augmentent ensemble (corrélation positive).
 - si < 0 une variable augmente tandis que l'autre diminue (corrélation négative).
 - Si proche de 0 : La relation entre les variables est faible ou inexistante.