Le jeu de données **Iris** est un ensemble de données classique utilisé pour des tâches de classification, d'apprentissage automatique et d'analyse statistique. Il contient des informations sur 150 fleurs d'iris, réparties en trois espèces : *Iris setosa*, *Iris versicolor*, et *Iris virginica*. Les colonnes incluent la longueur et la largeur des sépales et des pétales.

comment nettoyer et manipuler ce jeu de données en utilisant Python avec des bibliothèques comme **pandas** et **seaborn** pour l'analyse. Voici un plan général de ce que nous allons faire :

1. **Chargement des données Iris.**
2. **Vérification et traitement des valeurs manquantes.**
3. **Normalisation des données.**
4. **Filtrage des colonnes spécifiques.**
5. **Visualisation des données pour exploration.**

Voici comment procéder.

1. Chargement des données Iris

Le jeu de données Iris est souvent intégré à des bibliothèques comme **Seaborn** ou **Scikit-learn**. Nous allons utiliser seaborn pour le charger.

import seaborn as sns

import pandas as pd

Chargement des données Iris

iris = sns.load_dataset('iris')

Affichage des 5 premières lignes

print(iris.head())

```
[1]: import seaborn as sns
     import pandas as pd
     # Chargement des données Iris
     iris = sns.load_dataset('iris')

     # Affichage des 5 premières lignes
     print(iris.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

Vérification et traitement des valeurs manquantes

Il est important de vérifier s'il y a des valeurs manquantes avant de faire l'analyse.

Vérification des valeurs manquantes

print(iris.isnull().sum())

Si des valeurs manquantes sont présentes, elles peuvent être remplacées par des valeurs statistiques comme la moyenne ou supprimées.

[3]:
```
# Vérification des valeurs manquantes
print(iris.isnull().sum())
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Normalisation des données

Il est parfois nécessaire de normaliser les données pour certaines méthodes statistiques ou d'apprentissage automatique.

from sklearn.preprocessing import StandardScaler

Sélection des colonnes numériques (sans la colonne 'species')

iris_features = iris.drop(columns='species')

Normalisation des données

scaler = StandardScaler()

iris_normalized = pd.DataFrame(scaler.fit_transform(iris_features), columns=iris_features.columns)

Ajout de la colonne 'species' de nouveau

iris_normalized['species'] = iris['species']

print(iris_normalized.head())

[5]:
```
from sklearn.preprocessing import StandardScaler
# Sélection des colonnes numériques (sans la colonne 'species')
iris_features = iris.drop(columns='species')
# Normalisation des données
scaler = StandardScaler()
iris_normalized = pd.DataFrame(scaler.fit_transform(iris_features), columns=iris_features.columns)

# Ajout de la colonne 'species' de nouveau
iris_normalized['species'] = iris['species']
print(iris_normalized.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0     -0.900681     1.019004     -1.340227    -1.315444  setosa
1     -1.143017    -0.131979     -1.340227    -1.315444  setosa
2     -1.385353     0.328414     -1.397064    -1.315444  setosa
3     -1.506521     0.098217     -1.283389    -1.315444  setosa
4     -1.021849     1.249201     -1.340227    -1.315444  setosa
```

Filtrage des colonnes spécifiques

Exemple de filtrage sur les colonnes longueur des sépales et des pétales

filtered_data = iris[['sepal_length', 'petal_length', 'species']]

print(filtered_data.head())

```
[9]: # Exemple de filtrage sur les colonnes longueur des sépales et des pétales
filtered_data = iris[['sepal_length', 'petal_length', 'species']]
print(filtered_data.head())
```

```
   sepal_length  petal_length species
0           5.1           1.4  setosa
1           4.9           1.4  setosa
2           4.7           1.3  setosa
3           4.6           1.5  setosa
4           5.0           1.4  setosa
```

Visualisation des données

La visualisation est un excellent moyen d'explorer les données.

import matplotlib.pyplot as plt

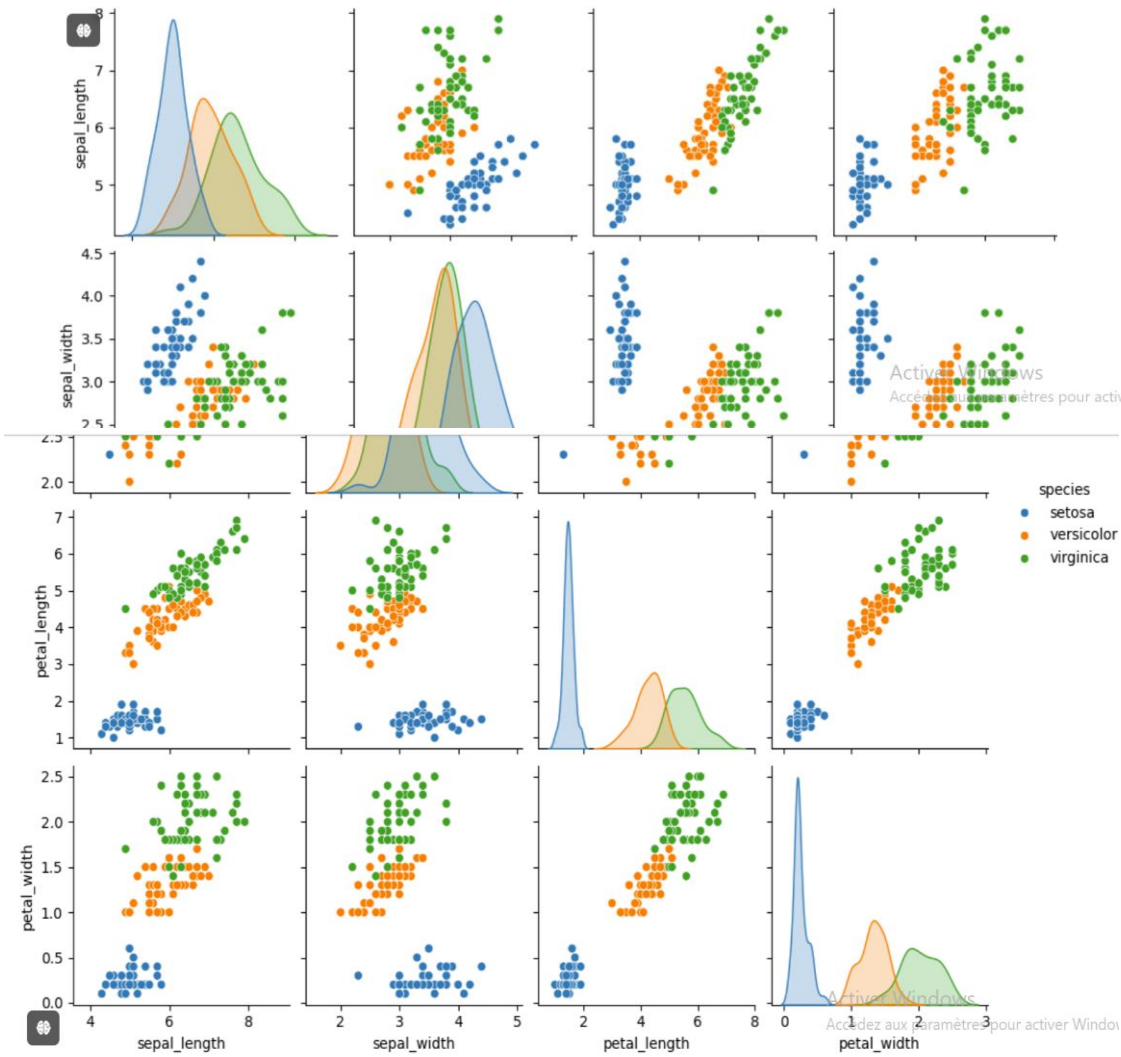
Visualisation de la distribution des espèces

sns.pairplot(iris, hue="species")

plt.show()

```
[11]: import matplotlib.pyplot as plt

# Visualisation de la distribution des espèces
sns.pairplot(iris, hue="species")
plt.show()
```



Activer Windows
Accédez aux paramètres pour activer Windows.