[I Python 2](#_Toc120721270)

[1 Package Pandas 2](#_Toc120721271)

[A Fonction/Objet DataFrame 2](#_Toc120721272)

[B Fonction/Objet Series 3](#_Toc120721273)

[C Fonction concat 3](#_Toc120721274)

[2 Package Numpy 5](#_Toc120721275)

[A Fonction/Objet array 5](#_Toc120721276)

[3 Package Matplotlib.pyplot 5](#_Toc120721277)

[4 Package Sklearn 5](#_Toc120721278)

[II Git 6](#_Toc120721279)

[III Machine Learning 6](#_Toc120721280)

[1 Gaussian Mixture Models 6](#_Toc120721281)

[2 Kmeans 6](#_Toc120721282)

[IV Ordonnanceur 6](#_Toc120721283)

# I Python

## 1 Package Pandas

Package qui permet de créer, manipuler des dataframe, voir fonction DataFrame donc.

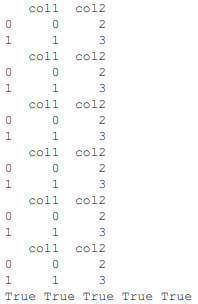
### A Fonction/Objet DataFrame

Les objets de type dataframe sont une forme de tableau à 2 dimensions avec des colonnes qui définissent des variables et des lignes qui définissent des individus indexés.

Chaque colonne peut avoir son type de valeur.

Plusieurs exemples pour faire le même dataframe :





La fonction prend en argument obligatoire data où on met un objet rempli de données à 2 dimensions max. Cet objet peut être un dictionnaire, un numpy.array ou une liste. Il faut bien regarder la documentation à propos de cet argument pour être sûr des bonnes dimensions qu’il prend.

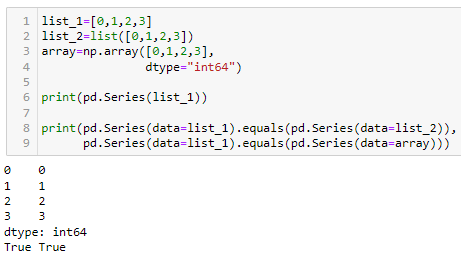
Elle peut aussi prendre l’argument index qui peut être une liste ou un numpy.array (d’une seule dimension) qui va fixer l’index qu’on veut pour data (par défaut [0,1,2,…].

Elle peut aussi prendre l’argument dtype qui va convertir si possible les données de data dans le type qu’on veut.

Elle peut aussi prendre l’argument columns qui peut être une liste ou un numpy.array (d’une seule dimension).

### B Fonction/Objet Series

Un objet pd.Series est une sorte de liste. Chaque colonne d’un dataframe est une Series.

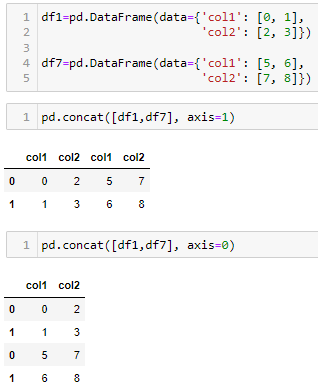


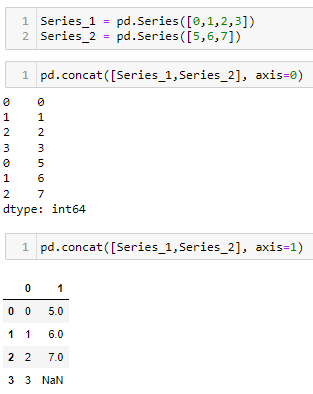
Elle peut aussi prendre l’argument index qui peut être une liste ou un numpy.array (d’une seule dimension) qui va fixer l’index qu’on veut pour data (par défaut [0,1,2,…].

Elle peut aussi prendre l’argument dtype qui va convertir si possible les données de data dans le type qu’on veut.

### C Fonction concat

Elle permet de concaténer des dataframe entre eux ou des pd.Series entre elles.





## 2 Package Numpy

Il sert principalement à créer, manipuler les objets de type numpy.array.

### A Fonction/Objet array

Ça ressemble à une liste mais ça n’en est pas une. Ça a un avantage en termes d’espace dans la mémoire.

List est un type originel de Python. Il est facile à utiliser mais beaucoup de fonctions rapides sur des numpy array ne fonctionnent pas sur des list, en particulier toutes les fonctions issues des packages numpy, pandas, sklearn.

## 3 Package Matplotlib.pyplot

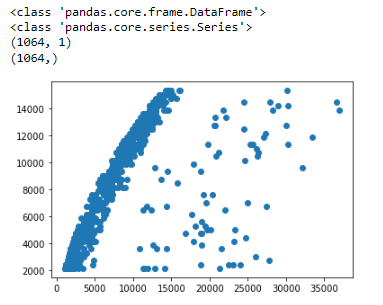
Pour faire des graphiques simples, avec un code assez lisse. Ca manque de disponibilité de personnalisation, seaborn est mieux pour ça.

## 4 Package Sklearn

Pour créer, entraîner et tester des modèles de machine learning.

Ici un exemple pour un modèle supervisé (avec y donné donc) :





### Conseils

Vérifier le type d’un objet et ses dimensions.

Mettre le nom de l’argument + = avant un argument dans une fonction.

Regarder dans la documentation d’une fonction quels type et dimension d’objet elle prend, pour chaque argument.

Comparer les types et les dimensions de son objet et des différentes valeurs à l’intérieur de ces objets avec ceux d’un exemple disponible dans la documentation.

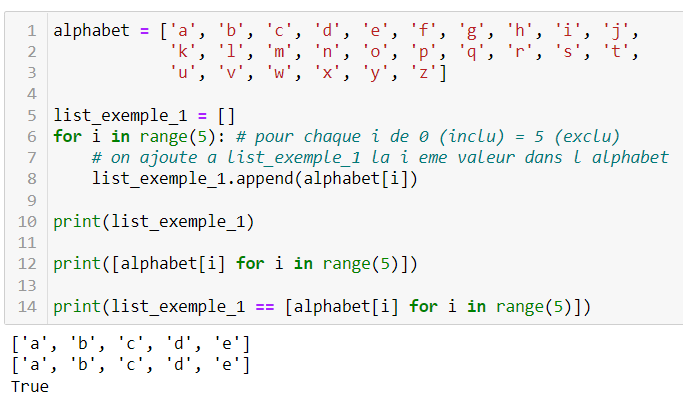
Eviter de convertir plusieurs fois un objet d’un type à l’autre juste pour pouvoir appliquer des fonctions qui seraient disponibles seulement pour un certain type.

Vérifier quand on a une liste s’il s’agit d’un numpy.array ou d’une liste classique.

Eviter dans un même code d’avoir à manipuler à la fois des numpy.array, des dataframe, des list et des dictionnaires en même temps.

Diminuer les boucles dans les boucles dans les boucles…

Essayer pour une boucle assez courte de la faire en une ligne : cela évite notamment lorsque l’on veut ajouter des éléments dans une liste, à d’abord l’initialiser vide et la remplir avec la fonction .append(), on peut d’ailleurs carrément ne pas avoir à mettre la liste dans une variable.



# II Git

# III Machine Learning

## 1 Gaussian Mixture Models

## 2 Kmeans

# IV Ordonnanceur