

Python : Manipulation de fichiers

Bibliothèque **pandas**

K. Boudjelaba
k.boudjelaba@carnus.fr

29 décembre 2021

Table des matières

1	Installation de la librairie pandas	2
1.1	Autres méthodes d'installation	2
2	Manipulation de fichiers avec pandas	2
2.1	Fichiers CSV	2
2.2	Manipulation des données	3
2.3	Enregistrement des données dans un fichier CSV	4
2.4	Lecture d'un fichier texte	7
3	Exercices	9

1 Installation de la librairie pandas

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Etape 1

Home

Environments

Learning

Community

Documentation

Developer Blog

Search Environments

base (root)

Etape 2

All

Channels

Update index...

Etape 3

pandas

Etape 4

pandas

Etape 5

Apply

Clear

Name	Description	Version
autovizwidget	An auto-visualization library for pandas dataframes	0.13.1
blaze	Numpy and pandas interface to big data	0.11.3
geopandas	Geographic pandas extensions.	0.6.1
pandas	High-performance, easy-to-use d...	0.25.1
pandas-datareader	Up to date remote data access for pandas, works for multiple versions of pandas	0.8.1
pandas-profiling	Generate profile report for pandas dataframe	1.4.1
pandasql	SqlDf for pandas	0.7.3
qgrid	Pandas dataframe viewer for jupyter notebook	1.1.1
streamz	Manage streaming data, optionally with dask and pandas	0.5.2

9 packages available matching "pandas" 1 package selected

1.1 Autres méthodes d'installation

Vous pouvez l'installer via le Terminal :

```
pip install pandas
Ou
conda install pandas
```

2 Manipulation de fichiers avec pandas

2.1 Fichiers CSV

Création et écriture d'un fichier CSV

```
import csv
with open('csv/Tempo.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["SN", "Nom", "Logiciel", "Date", "Entier", "Réel"])
    writer.writerow([1, "Linux", "Matlab", "11/01/2020", 20, 3])
    writer.writerow([2, "Mac Os", "LTspice", "12/02/2020", 50, 4.5])
    writer.writerow([3, "Windows-10", "Python 3", "14/01/2021", 30, 0.76])
```

Ouverture du fichier créé

```
import pandas as pd
df = pd.read_csv('csv/Tempo.csv')
df.head()
```

	SN	Nom	Logiciel	Date	Entier	Réel
0	1	Linux	Matlab	11/01/2020	20	3.00
1	2	Mac Os	LTspice	12/02/2020	50	4.50
2	3	Windows-10	Python 3	14/01/2021	30	0.76

```
df.shape
(3, 6)
```

2.2 Manipulation des données

Vérification du type de données

```
print(type(df['SN'][0]))
<class 'numpy.int64'>

print(type(df['Nom'][0]))
<class 'str'>

print(type(df['Logiciel'][0]))
<class 'str'>

print(type(df['Date'][0]))
<class 'str'>

print(type(df['Entier'][0]))
<class 'numpy.int64'>

print(type(df['Réel'][0]))
<class 'numpy.float64'>
```

Changement de format pour la date

```
import pandas
df = pandas.read_csv('csv/Tempo.csv', parse_dates=['Date'])
df.head()
```

	SN	Nom	Logiciel	Date	Entier	Réel
0	1	Linux	Matlab	2020-11-01	20	3.00
1	2	Mac Os	LTspice	2020-12-02	50	4.50
2	3	Windows-10	Python 3	2021-01-14	30	0.76

Remarque :

La conversion est réalisée suivant le standard américain pour coder les dates (mois/-jour/année). Alors qu'en France la date est codée différemment (jour/mois/année). Pour la troisième ligne, Python a trouvé que le numéro du mois est > 12 et il a permuté le mois et le jour.

```
print(type(df['Date'][0]))
<class 'pandas._libs.tslibs.timestamps.Timestamp'>
```

Affichage des colonnes

```
import pandas as pd
df.iloc[:,2]
```

	SN	Nom
0	1	Linux
1	2	Mac Os
2	3	Windows-10

```
import pandas as pd
df.iloc[:,(4)]
```

```
0    20
1    50
2    30
Name: Entier, dtype: int64
```

```
import pandas as pd
d1 = pd.read_csv("csv/Tempo.csv", usecols=[1,3])
d1.head()
```

	Nom	Date
0	Linux	11/01/2020
1	Mac Os	12/02/2020
2	Windows-10	14/01/2021

2.3 Enregistrement des données dans un fichier CSV

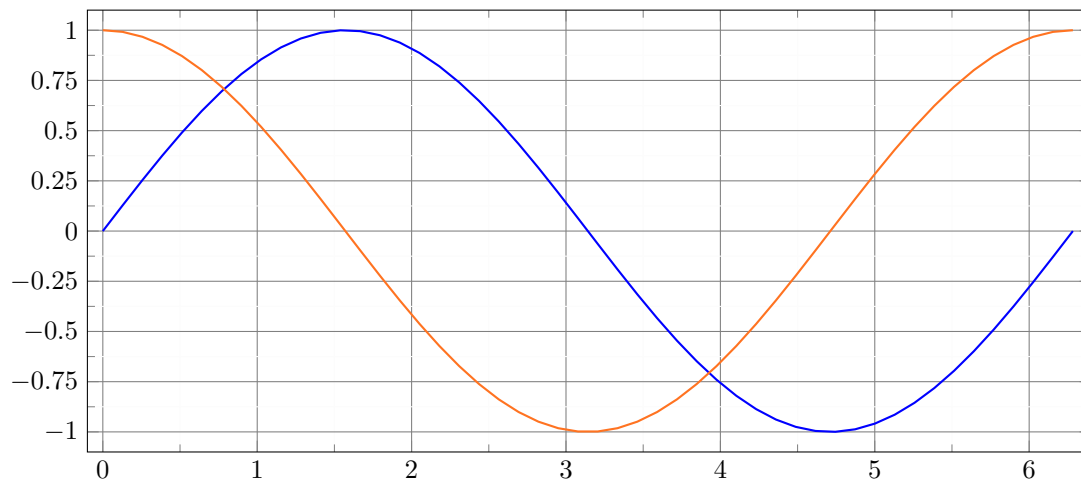
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

t = np.linspace(0,2*np.pi,50,endpoint=True)
s1 = np.sin(t)
s2 = np.cos(t)

data = np.zeros((len(t),3))
data[:,0] = t
data[:,1] = s1
data[:,2] = s2

np.savetxt("csv/sin_cos.csv",data,delimiter=",",header="t,Sinus,Cosinus",comments="")

plt.figure()
plt.plot(t,s1,t,s2)
plt.grid()
plt.show()
```



```
import pandas as pd
d = pd.read_csv("csv/sin_cos.csv")
d.head()
```

	t	Sinus	Cosinus
0	0.000000	0.000000	1.000000
1	0.128228	0.127877	0.991790
2	0.256457	0.253655	0.967295
3	0.384685	0.375267	0.926917
4	0.512913	0.490718	0.871319

```
d.shape
```

(50, 3)

Manipulation de données

```
test_list = ["BTS", "1", "SN", "2", "ec"]
print("liste originale : " + str(test_list))
# Conversion chaîne caractères en liste d'entiers et autres
res = [int(ele) if ele.isdigit() else ele for ele in test_list]
print("Liste après conversion : " + str(res))
```

```
liste originale : ['BTS', '1', 'SN', '2', 'ec']
Liste après conversion : ['BTS', 1, 'SN', 2, 'ec']
```

```

import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

Ch = ["a'103,10\\r\\n", "b'101,14\\r\\n", "c'102,16\\r\\n", "d'103,18\\r\\n"]

def nettoyage(Liste):
    newList = []
    for i in range(len(Liste)):
        tempList = Liste[i][2:]
        newList.append(tempList[:-6])
    return newList
NCh = nettoyage(Ch)
print(NCh)

def ecrire(Liste):
    file = open("txt/FichierTXT.txt", mode="w")
    for i in range(len(Liste)):
        file.write(Liste[i] + '\\n')
    file.close()

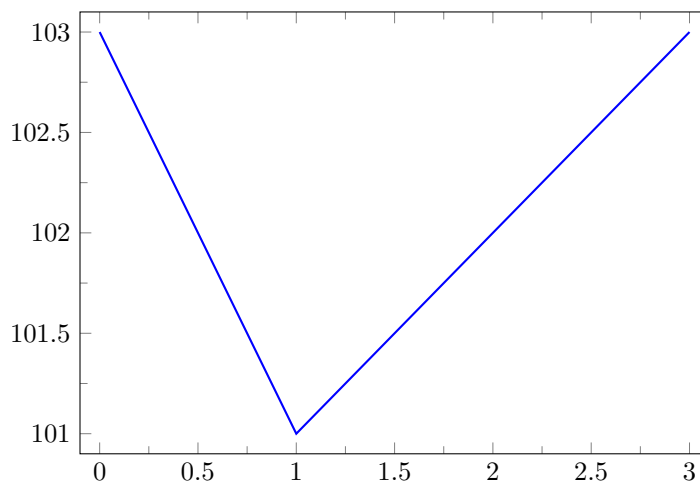
ecrire(NCh)

vall = np.loadtxt("txt/FichierTXT.txt", unpack=True)

plt.figure()
plt.plot(vall)

```

['103', '101', '102', '103']



Remarque :

Le délimiteur peut être un espace, un double espace | , ; \t ...

2.4 Lecture d'un fichier texte

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

f = open("txt/Donnees.txt" , 'r')
f.readline()
data = np.loadtxt(f)
f.close()

plt.figure()
plt.plot(data[:,0], data[:,1], "r-")
plt.plot(data[:,0], data[:,2], "b-")
plt.grid()
plt.show()
```

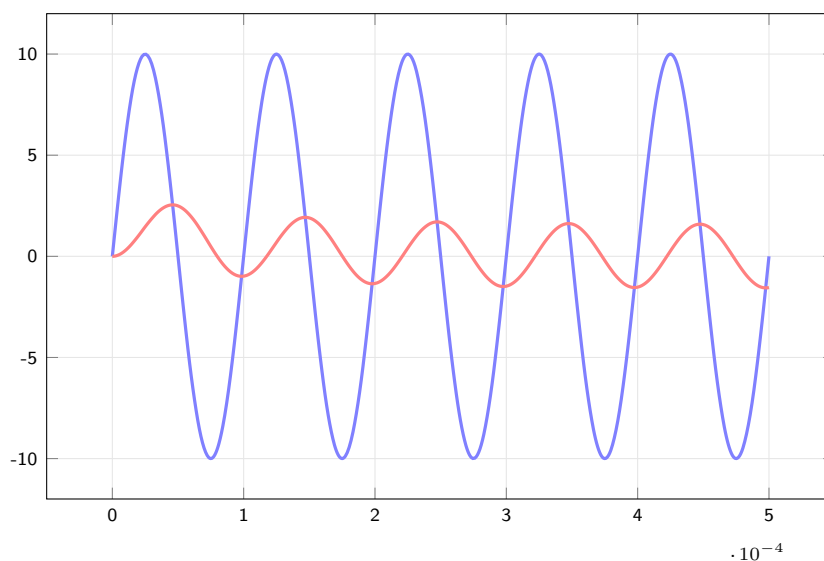


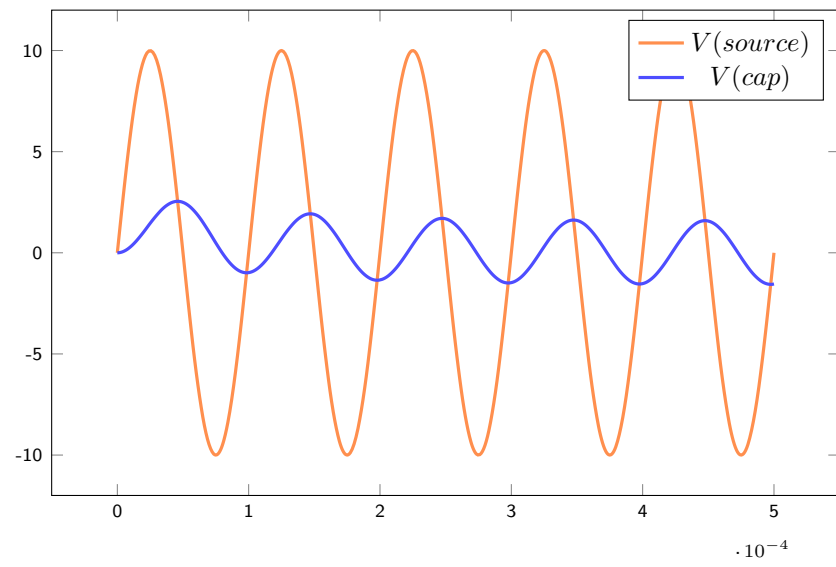
FIGURE 1 – Courbes des données

```
import pandas as pd

df = pd.read_csv('txt/Donnees.txt', sep='\s+', header=None, skiprows=1)
df.head()
```

	time	V(cap)	V(source)
0	0.000000e+00	0.000000	0.000000
1	9.142857e-08	0.000038	0.057450
2	1.828571e-07	0.000135	0.114890
3	2.742857e-07	0.000293	0.172320
4	3.657143e-07	0.000511	0.229741

```
df.plot('time', ['V(cap)', 'V(source)'])
plt.show()
```



3 Exercices

Exercice 1

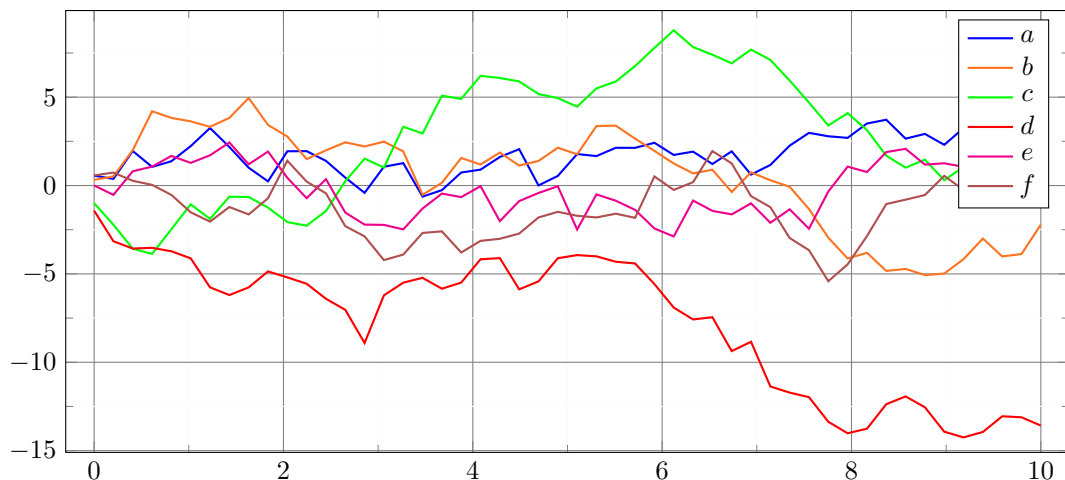
Le code suivant génère les données "x" et "y" ("y" contient plusieurs colonnes)

- Enregistrer ces données dans un seul fichier csv nommé Donnees.csv.
- Ecrire un code pour lire ce fichier.
- Lire les données contenues dans ce fichier.
- Calculer la moyenne de la deuxième colonne de "y".
- Tracer la deuxième colonne de "y" en fonction de "x".
- Supprimer la 3ème colonne de "y" et enregistrer ce nouveau tableau dans un fichier csv nommé Donnees3.csv.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x = np.linspace(0,10,50,endpoint=True)
y = np.cumsum(np.random.randn(50,6),0)

plt.figure(figsize=(10,5))
plt.plot(x,y)
plt.legend('abcdef', ncol=2, loc='upper left')
plt.grid()
plt.show()
```



Exercice 2

Soit la chaîne de caractère suivante :

"a'99.7;10\\r\\n", "b'101;14\\r\\n", "c'102;16\\r\\n", "d'103;18\\r\\n"

- Ecrire un code pour extraire les chiffres contenus dans cette chaîne (x1 doit contenir les valeurs 99.7, 101, 102 et 103; et x2 doit contenir les autres valeurs : 10, 14, 16 et 18)
- Enregistrer ces chiffres dans un fichier texte nommé Texte.txt
- Lire ce fichier texte et tracer x2 en fonction de x1

Exercice 3

En utilisant la librairie Pandas, construire le tableau suivant :

	Nom	Spécialité	CP	Chiffre1	Chiffre2	Date	Adresse
0	Nom1	SN	12000	1.5	10	12/03/2020 09 :55	20 Rue A, Rodez, Av 12000
1	Nom2	IMRT	31000	5.0	08	11/02/2020 11 :01	1 Avenue B, Toulouse, Hg 31000
2	Nom3	IMRT	75000	12.0	15	10/01/2020 13 :30	5 Place C, Paris, Pa 75001
3	Nom4	SN	12000	3.1	20	09/12/2019 17 :25	3 Rue D, Rodez, Av 12000

- Afficher ce tableau
- A partir de maintenant, vous allez travailler avec le fichier fourni : Fichier.csv (Voir mail ou GitHub)
- Ecrire un code pour lire le fichier "Fichier.csv"
- Ecrire un code pour extraire le jour à partir de la date et ajouter à la fin, la colonne Jour et afficher le tableau
- Ecrire un code pour extraire le mois à partir de la date et ajouter à la fin, la colonne Mois et afficher le tableau
- Ecrire un code pour extraire le nom de la ville à partir de l'adresse et ajouter à la fin, la colonne Ville et afficher le tableau
- Supprimer la colonne Jour
- Changer le format de la date
- Ecrire un code pour extraire les heures à partir de la date et ajouter à la fin, la colonne Heure et afficher le tableau
- Ecrire un code pour extraire les minutes à partir de la date et ajouter à la fin, la colonne Minute et afficher le tableau
- Ecrire un code pour mettre au format integer la colonne "Mois" et vérifier les résultats
- Ecrire un code pour mettre au bon format integer la colonne "Chiffre2" et vérifier les résultats
- Ajouter une colonne pour calculer la moyenne entre le "Chiffre1" et le "Chiffre2"
- Enregistrer votre fichier dans un fichier csv nommé FichierResultat.csv