

Import de données numériques

Code sous licence creative commun CC BY-NC-SA BY Gaëlle Rebolini

Le programme présenté ci-dessous est adapté à des fichiers .csv (type tableur) obtenus lors de pointages vidéo. Il devra évidemment être adapté pour des fichiers obtenus lors d'autres expériences.

1. Enregistrer ou exporter le fichier contenant votre tableau de données sous format .csv dans le dossier contenant votre notebook (fichier .ipynb) ou votre programme python (fichier.py). Attention, pour l'utilisation avec l'ENT Nero version 2018, petite subtilité à la fin.
 - Dans Regressi, enregistrer le fichier sous le format (type) OpenOffice, CSV (choisir *réel Vrai CSV* dans la fenêtre qui s'affiche alors).
 - Dans Loggerpro, exporter le fichier comme CSV...
 - Dans Aviméca, exporter les données dans Regressi puis vous reporter à la ligne ci-dessus.

Attention : les logiciels de pointage retournent des tableaux de colonnes avec des entêtes (une à deux lignes) qu'il faudra par la suite retranscrire sous forme de listes (une liste par colonne) sans tenir compte des entêtes.

Voici une capture d'écran du fichier parabole.csv obtenu à l'aide de Regavi/Regressi ouvert sous Excel

| | A | B | C |
|----|------|-------------|------------|
| 1 | t | x | y |
| 2 | s | m | m |
| 3 | 0 | -0,00280894 | 0 |
| 4 | 0,04 | 0,06460572 | 0,14325617 |
| 5 | 0,08 | 0,14044722 | 0,26684972 |
| 6 | 0,12 | 0,21347978 | 0,37639855 |
| 7 | 0,16 | 0,28651233 | 0,47190267 |
| 8 | 0,2 | 0,36235383 | 0,55336205 |
| 9 | 0,24 | 0,43538639 | 0,61796778 |
| 10 | 0,28 | 0,51403683 | 0,66571983 |
| 11 | 0,32 | 0,58426044 | 0,69380928 |
| 12 | 0,36 | 0,66291089 | 0,71347189 |
| 13 | 0,4 | 0,73875239 | 0,71347189 |
| 14 | 0,44 | 0,81459389 | 0,69661822 |
| 15 | 0,48 | 0,89043539 | 0,66010194 |
| 16 | 0,52 | 0,96627689 | 0,61796778 |
| 17 | 0,56 | 1,03930944 | 0,55336205 |
| 18 | 0,6 | 1,11515094 | 0,46909372 |
| 19 | 0,64 | 1,19099244 | 0,37358961 |
| 20 | 0,68 | 1,26964289 | 0,26123183 |
| 21 | 0,72 | 1,3398665 | 0,13482933 |
| 22 | | | |

Capture sous excel

Le même fichier ouvert sous Jupiter Notebook

```

1 t;x;y
2 s;m;m
3 0;-0,0028089444369039;0
4 0,04;0,0646057220487898;0,143256166282099
5 0,08;0,140447221845195;0,266849721505871
6 0,12;0,213479777204697;0,376398554545123
7 0,16;0,286512332564198;0,471902665399856
8 0,2;0,362353832360603;0,553362054070069
9 0,24;0,435386387720105;0,617967776118859
10 0,28;0,514036831953414;0,665719831546225
11 0,32;0,584260442876012;0,693809275915264
12 0,36;0,662910887109321;0,713471886973591
13 0,4;0,738752386905726;0,713471886973591
14 0,44;0,814593886702132;0,696618220352168
15 0,48;0,890435386498537;0,660101942672417
16 0,52;0,966276886294942;0,617967776118859
17 0,56;1,03930944165444;0,553362054070069
18 0,6;1,11515094145085;0,469093720962952
19 0,64;1,19099244124725;0,373589610108219
20 0,68;1,26964288548056;0,261231832632063
21 0,72;1,33986649640316;0,134829332971387
22 |

```

Capture sous jupyter

2. Les cellules suivantes contiennent les lignes de code qui vous permettront d'afficher votre tableau de données sous forme de listes (une liste par colonne de votre tableau)

In [1]:

```

# Chargement de la bibliothèque csv afin de pouvoir lire par la suite
# le fichier csv

import csv

```

In [2]:

```

# création de la fonction appelée charge_fichier_csv() qui
# permettra de récupérer les données des colonnes d'un fichier.csv

def charge_fichier_csv(fichier):

    # ouverture du fichier .csv

    with open(fichier, 'r', encoding='utf-8') as f :

        # lecture du fichier à l'aide de la fonction csv.reader.
        # Il faut préciser le délimiteur de colonnes utilisé dans le
        # fichier .csv (ici c'est le ;)

        rfichier = csv.reader(f, delimiter=";")

        # création et initiation du tableau sous forme de liste qui recevra
        # les listes de nombres réels correspondant aux colonnes

```

```

tableau=[]

# le contenu d'une cellule est initialement lu comme une chaîne de
# caractères
# nous voulons obtenir des listes de nombres réels correspondant
# aux colonnes de notre tableau csv,
# donc :
#     - il ne faut pas prendre en compte les lignes
#       correspondant aux entêtes
#     - il faut convertir les chaînes de caractères en nombres
#       réels décimaux

# attention : les virgules des nombres décimaux doivent être
# remplacées par des points

# test permettant de déterminer si le contenu de la première cellule
# de la ligne du tableau csv n'est pas un nombre en testant le premier
# caractère de ce contenu.
# Si le test est validé, grâce à la commande "continue", on revient au
# début de la boucle for sans continuer d'exécuter son contenu car on
# suppose que la ligne du tableau csv est une ligne d'entête et on
# change de ligne!

    for row in rfichier:      # pour chaque ligne du fichier csv

        if row[0][0] not in ("0","1","2","3","4","5","6","7","8",
                             "9",",",".","):
            continue

# on parcourt chaque cellule d'une ligne du tableau csv correspondant à
# une colonne du tableau csv.

    for i in range (len(row)):

# création d'une liste vide qui contiendra les valeurs des cellules
# d'une colonne du fichier csv.
        X = []

# ajout de cette liste au tableau
        tableau.append(X)

# ajout dans cette liste du contenu de la cellule i de la ligne en la
# convertissant en nombre réel décimal
# une fois qu'une seule ligne a été entièrement parcourue, on interrompt
# la boucle "for row in rfichier" grâce à la commande "break" afin de ne
# pas créer de liste vide supplémentaire à chaque nouvelle ligne

```

```

        tableau[i].append(float(row[i].replace(",",'.')))
    break

# on reprend le parcours des lignes du tableau (sauf celles parcourues
# précédemment)

    for row in rfichier:
        for i in range (len(row)):
            tableau[i].append(float(row[i].replace(",",'.')))

    return (tableau)

```

Voici la fonction sans commentaire afin d'y voir un peu plus clair !

In [3]:

```

def charge_fichier_csv(fichier):
    """
    Charge un fichier csv et le renvoie sous forme de tableau

    :param: nom de fichier
    :returns: tableau des données
    """
    with open(fichier, 'r', encoding='utf-8') as f :
        rfichier = csv.reader(f, delimiter=";")
        tableau=[]
        for row in rfichier:
            if row[0][0] not in ("0","1","2","3","4","5","6","7","8",
                                "9"," ","","."):
                continue
            for i in range (len(row)):
                X = []
                tableau.append(X)
                tableau[i].append(float(row[i].replace(",",'.')))
            break
        for row in rfichier:
            for i in range (len(row)):
                tableau[i].append(float(row[i].replace(",",'.')))
    return (tableau)

```

In [4]:

```

# Le début du chemin n'a pas besoin d'être spécifié si le fichier
# .csv se trouve dans le même dossier que ce fichier notebook

tableau = charge_fichier_csv('09-fichiers-csv-parabole.csv')
t=tableau[0]
print(t)
x=tableau[1]
print(x)
y=tableau[2]
print(y)

```

[0.0, 0.04, 0.08, 0.12, 0.16, 0.2, 0.24, 0.28, 0.32, 0.36, 0.4, 0.44, 0.48, 0.52, 0.56, 0.6, 0.64, 0.68, 0.72]

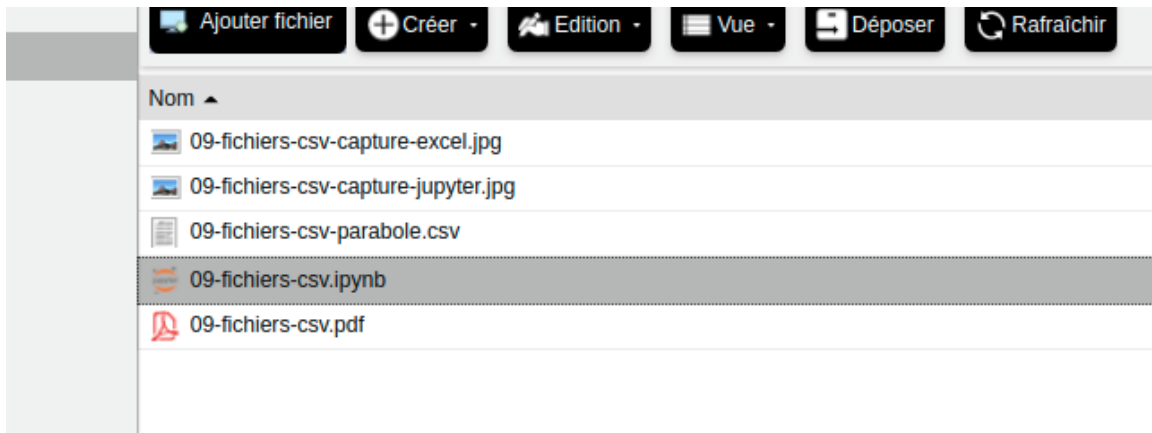
[-0.002808944, 0.064605722, 0.140447222, 0.213479777, 0.286512333, 0.362353832, 0.435386388, 0.514036832, 0.584260443, 0.662910887, 0.738752387, 0.814593887, 0.890435386, 0.966276886, 1.039309442, 1.115150941, 1.190992441, 1.269642885, 1.339866496]

[0.0, 0.143256166, 0.266849722, 0.376398555, 0.471902665, 0.553362054, 0.617967776, 0.665719832, 0.693809276, 0.713471887, 0.713471887, 0.69661822, 0.660101943, 0.617967776, 0.553362054, 0.469093721, 0.37358961, 0.261231833, 0.134829333]

WARNING : subtilité pour le jupyter ENT Nero 2018

Même si vous avez mis vos fichiers CSV dans le même répertoire que votre fichier ipynb dans votre espace “Mes documents” de l’ENT, l’importation des données csv ne fonctionnera pas (en passant, l’affichage des images de ce notebook non plus, d’ailleurs..)

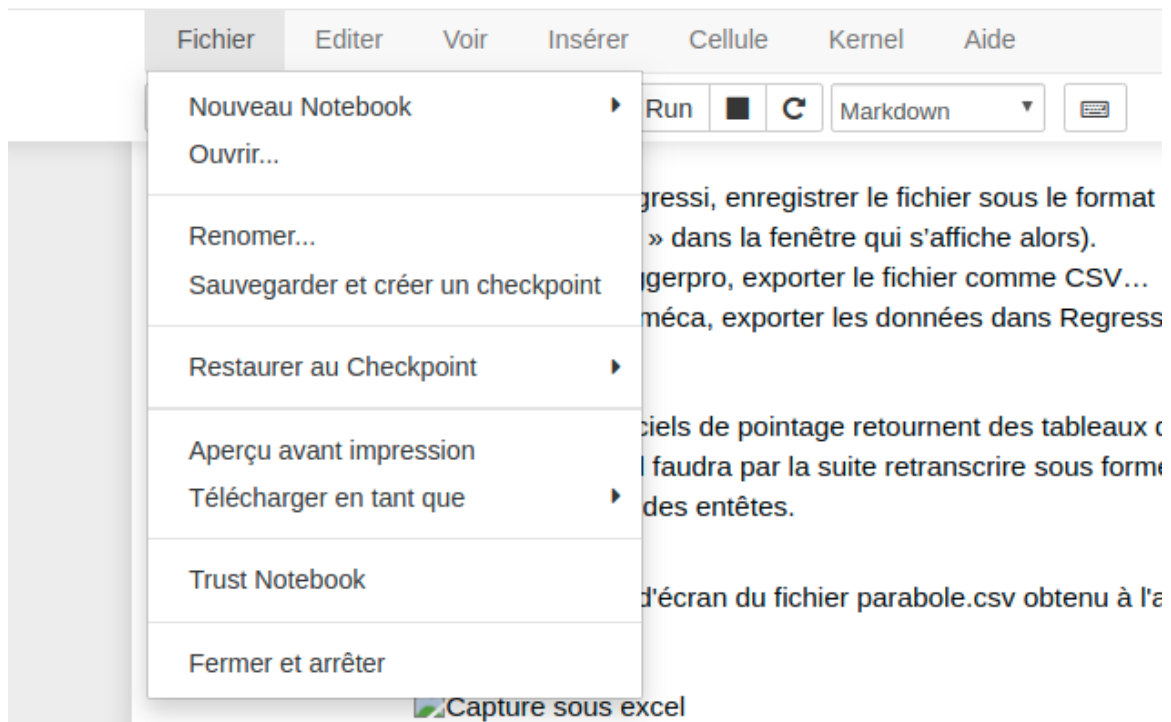
En fait sur cette version 2018, lors de l’exécution d’un notebook ipython, l’ENT copie ce fichier dans un répertoire temporaire vide... donc raté pour les fichiers CSV.



Fichiers dans le même répertoire

Il existe une solution qui ressemble un peu à un bricolage, mais qui fonctionne en attendant 2019 :

- dans le notebook jupyter, cliquer sur “Fichier > Ouvrir”
- le répertoire de travail de votre notebook apparaît alors dans un onglet séparé
- vous pouvez alors y placer votre fichier csv avec le bouton “Upload”



Le même fichier ouvert sous Jupiter Notebook

Capture sous jupyter

1. Les cellules suivantes contiennent les lignes de code qu

Mettre le fichier dans le répertoire de travail - étape 1

Files
Running
Clusters

Select items to perform actions on them.

Upload
Nouveau

0

/ write / 72555375 / 218380958

Name
Last Modified

| | |
|--|--------------------------|
| .. | il y a quelques secondes |
| <input type="checkbox"/> 09-fichiers-csv.ipynb | Running il y a 3 minutes |
| <input type="checkbox"/> 09-fichiers-csv-capture-excel.jpg | il y a 11 minutes |
| <input type="checkbox"/> 09-fichiers-csv-capture-jupyter.jpg | il y a 11 minutes |
| <input type="checkbox"/> 09-fichiers-csv-parabole.csv | il y a 2 minutes |

Mettre le fichier dans le répertoire de travail - étape 2

Cette difficulté devrait ne plus exister à la rentrée 2019.