

Ce script python est destiné à convertir un fichier de données dyadiques d'une structure individuelle à une structure dyadique. Il nécessite python3+ et la librairie pandas.

Pour installer pandas : `python -m pip install --upgrade pandas` dans le terminal.

Structure individuelle :

## Individual

<i>Dyad</i>	<i>Person</i>	<i>X</i>	<i>Y</i>	<i>Z</i>
1	1	5	9	3
1	2	2	8	3
2	1	6	3	7
2	2	4	6	7
3	1	3	6	5
3	2	9	7	5

Structure dyadique :

<u>Dyad</u>					
<i>Dyad</i>	<i>X<sub>1</sub></i>	<i>Y<sub>1</sub></i>	<i>Z<sub>1</sub></i>	<i>X<sub>2</sub></i>	<i>Y<sub>2</sub></i>
1	5	9	3	2	8
2	6	3	7	4	6
3	3	6	5	9	7

## Code

In [1]:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import os
import pandas as pd
```

Indiquer le chemin du dossier contenant le fichier .csv à la place de path (entre guillemets)

In [2]:

```
os.chdir('D:\\Google Drive\\1-These\\13-EATMINT2\\Analyses\\Questionnaires')
```

Indiquer le nom du fichier .csv à la place de file (entre guillemets)

In [3]:

```
df1 = pd.read_csv('questionnaire_raw_data_individual_structure.csv', sep=';')
```

Voir les premières lignes de votre fichier :

In [4]:

```
df1.head()
```

Out[4]:

	Dyade	Sujet	Sexe	Condition	Score Raven	Activation_soi	Anxiété_soi	Colère_soi	Contrôle_soi	Déception_soi	...	Mon partenaire mettait nos différences en évidence	N partenaire n'é I récept ce qui lui dis
0	1	1	H	CBVB	20.0	4	1.0	3	2.0	6.0	...	1.0	
1	1	2	H	CBVB	22.0	4	1.0	1	2.0	5.0	...	1.0	
2	2	1	H	CBVH	19.0	5	2.0	3	4.0	2.0	...	2.0	
3	2	2	H	CBVH	17.0	4	3.0	1	5.0	1.0	...	1.0	
4	3	1	H	CHVB	23.0	4	6.0	2	3.0	6.0	...	5.0	

5 rows × 125 columns



Supprimer les colonnes pour lesquelles le changement de structure ne s'applique pas

In [5]:

```
data = df1.drop(['Dyade', 'Sujet', 'Sexe', 'Condition'], axis = 1)  
data.head()
```

Out[5]:

	Score Raven	Activation_soi	Anxiété_soi	Colère_soi	Contrôle_soi	Déception_soi	Désespoir_soi	Ennui_soi	Espoir_soi	Fierté_soi	...	I n d
0	20.0	4	1.0	3	2.0	6.0	2.0	5	NaN	3	...	
1	22.0	4	1.0	1	2.0	5.0	2.0	1	5.0	2	...	
2	19.0	5	2.0	3	4.0	2.0	1.0	1	3.0	5	...	
3	17.0	4	3.0	1	5.0	1.0	2.0	1	7.0	5	...	
4	23.0	4	6.0	2	3.0	6.0	2.0	1	2.0	5	...	

5 rows × 121 columns



Mettre les noms des colonnes dans une variable

In [6]:

```
variables = data.columns
```

Créer un nouveau tableau de données

In [7]:

```
new_data = pd.DataFrame()
```

Pour chaque colonne, mettre dans p1 les valeurs paires et dans p2 les valeurs p2. Donner un index identique pour p1 et p2. Ajouter les deux colonnes à la suite dans le tableau de données

In [8]:

```
for i in variables:
    p1 = data[i].iloc[::2]
    p1.index = range(1, len(p1)+1)

    p2 = data[i].iloc[1::2]
    p2.index = range(1, len(p2)+1)

    v = i
    v1 = i + '_p1'
    v2 = i + '_p2'

    new_data[v1] = p1
    new_data[v2] = p2
```

In [9]:

```
new_data.head()
```

Out[9]:

	Score Raven_p1	Score Raven_p2	Activation_soi_p1	Activation_soi_p2	Anxiété_soi_p1	Anxiété_soi_p2	Colère_soi_p1	Colère_soi_p2	Contrôle
--	----------------	----------------	-------------------	-------------------	----------------	----------------	---------------	---------------	----------

1	20.0	22.0	4	4	1.0	1.0	3	1
2	19.0	17.0	5	4	2.0	3.0	3	1
3	23.0	21.0	4	4	6.0	4.0	2	5
4	21.0	21.0	4	4	3.0	3.0	2	3
5	23.0	22.0	4	4	1.0	2.0	1	1

5 rows × 242 columns



Exporter le fichier .csv restructuré

In [10]:

```
new_data.to_csv('out.csv', sep=';')
```