

Chapitre : Traitement de données en tables

Une des utilisations principales de l'informatique de nos jours est le traitement de quantités importantes de données dans des domaines très variés :

"un site de commerce en ligne peut avoir à gérer des bases données pour des dizaines de milliers d'articles en vente, de clients, de commandes ; un hôpital doit pouvoir accéder efficacement à tous les détails de traitements de ses patients..."

Mais si les logiciels de gestion de base de données (SGDB) sont des programmes hautement spécialisés pour effectuer ce genre de tâches le plus efficacement et sûrement possible, il est facile de mettre en œuvre les opérations de base dans un langage de programmation comme Python.

I) Les données

Définition : Donnée

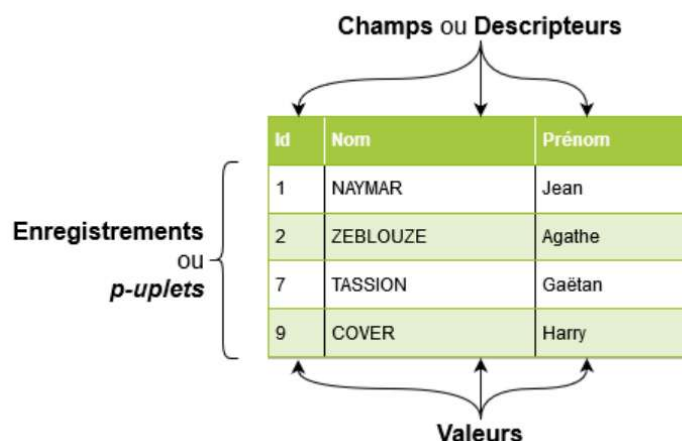
Une donnée (data en anglais) est une valeur attribuée à une entité pour la décrire. Cette entité peut être un objet, une personne, un événement, etc.

Une donnée peut être élémentaire ou complexe.

- Une donnée élémentaire représente une caractéristique de base (un nom, un numéro, etc.). Cette donnée est caractérisée par un descripteur qui permet de donner le format dans lequel cette donnée est représentée.
- Une donnée complexe est constituée de plusieurs données élémentaires.

II) Les données en tables

Organisées en table, les données se présentent sous la forme suivante :



III) Les fichiers .csv

Le format CSV (pour comma separated values, soit en français valeurs séparées par des virgules) est un format très pratique pour représenter des données structurées. Dans ce format, chaque ligne représente un enregistrement et, sur une même ligne, les différents champs de l'enregistrement sont séparés par une virgule (d'où le nom). Cependant en France la virgule est déjà utilisée pour les nombres décimaux. Donc un .csv peut fonctionner avec d'autre séparateur le point virgule, la tabulation, les deux points.

Voilà à quoi ressemble un fichier csv. C'est un fichier texte séparé par des points virgules. Voici les impôts des foyers du 92 en 2020.

```
Index;Dép.;Commune;Libellé de la commune;Revenu fiscal de référence par tranche (en euros);Nombre de foyers fiscaux;Revenu fiscal de référence des foyers fiscaux;Impôt net (total)*;Nombre de foyers fiscaux imposés;Revenu fiscal de référence des foyers fiscaux imposés;Traitements et salaires : Nombre de foyers concernés;Traitements et salaires : Montant;Retraites et pensions : Nombre de foyers concernés;Retraites et pensions :Montant 0;922;2;Antony;0 à 10
000;5007;18765.611;-118.186;376;1692.962;2287;13714.648;985;7899.593
1;922;2;Antony;10 001 à 12
000;988;10892.822;-85.364;77;854.984;638;7330.074;344;4694.528 2;922;2;Antony;12 001
à 15 000;1586;21618.272;-59.781;139;1909.639;1233;18160.2;369;5771.733
3;922;2;Antony;15 001 à 20
000;3678;64373.852;850.72;2353;41451.418;2692;50072.424;1084;20689.649
4;922;2;Antony;20 001 à 30
000;6202;153271.296;7845.854;4914;121909.982;4636;122843.008;1781;44467.995
5;922;2;Antony;30 001 à 50
000;7097;274498.654;21480.86;6297;245006.593;5048;200932.88;2492;89388.879
6;922;2;Antony;50 001 à 100
000;7325;513966.608;55007.968;7122;501007.008;5723;402856.998;2282;122901.114
7;922;2;Antony;+ de 100
000;2877;497653.925;90418.277;2854;494411.232;2573;352568.007;621;41232.243
8;922;2;Antony;Total;34760;1555041.04;175340.348;24132;1408243.818;24830;1168478.239;9
9;921;4;Asnières-sur-Seine;0 à 10
000;10264;36634.296;-178.069;483;2201.151;4634;27994.714;2075;17237.037
10;921;4;Asnières-sur-Seine;10 001 à 12
000;1859;20450.231;-53.122;125;1371.863;1337;15705.509;541;7509 11;921;4;Asnières-
sur-Seine;12 001 à 15
000;3059;41510.01;-92.017;215;2929.504;2437;35703.544;663;10218.825
12;921;4;Asnières-sur-Seine;15 001 à 20
000;6102;106589.38;1488.753;3684;64968.015;4676;88115.886;1614;30373.081
13;921;4;Asnières-sur-Seine;20 001 à 30
000;10129;249777.924;13129.733;7815;193663.955;8408;223131.43;2089;49211.867
14;921;4;Asnières-sur-Seine;30 001 à 50
```

000;10277;392311.451;34508.997;9187;352519.118;8625;345912.05;2154;71161.375
15;921;4;Asnières-sur-Seine;50 001 à 100

000;7654;533764.939;61356.891;7443;520128.771;6668;480225.693;1367;68869.299
16;921;4;Asnières-sur-Seine;+ de 100

000;3360;614389.75;121621.208;3341;611640.171;3094;474529.313;429;27630.548
17;921;4;Asnières-sur-
Seine;Total;52704;1995427.981;231782.374;32293;1749422.548;39879;1691318.139;10932;282
18;922;7;Bagneux;0 à 10

000;5342;20181.849;-93.58;255;1157.484;2576;16222.356;1038;8525.206
19;922;7;Bagneux;10 001 à 12

000;1056;11638.025;-39.05;n.c.;n.c.;757;8993.876;336;4626.868 20;922;7;Bagneux;12 001 à
15 000;1836;24949.995;-63.392;134;1837.612;1458;21340.5;431;6546.676
21;922;7;Bagneux;15 001 à 20

000;3827;66871.948;807.327;2158;37932.214;2950;55727.068;1028;19389.874
22;922;7;Bagneux;20 001 à 30

000;4926;120075.765;5198.956;3463;84598.601;3987;104198.583;1222;28341.728
23;922;7;Bagneux;30 001 à 50

000;3992;151346.206;10000.905;3253;124822.254;3309;129794.362;1058;33160.49
24;922;7;Bagneux;50 001 à 100

000;1908;127899.015;13054.781;1863;125138.93;1638;111731.855;429;19986.722
25;922;7;Bagneux;+ de 100

000;325;49928.431;8938.425;n.c.;n.c.;296;36366.491;60;2945.944
26;922;7;Bagneux;Total;23212;572891.234;37804.372;11516;425132.036;16971;484375.091;560
27;921;9;Bois-Colombes;0 à 10

000;2283;8877.889;-64.526;167;826.456;1074;6531.457;454;3550.742 28;921;9;Bois-
Colombes;10 001 à 12 000;474;5213.576;-26.966;40;442.5;316;3733.738;153;2083.685
29;921;9;Bois-Colombes;12 001 à 15

000;801;10896.792;-19.457;92;1250.507;606;8833.614;202;3098.641 30;921;9;Bois-
Colombes;15 001 à 20

000;1813;31730.116;438.545;1167;20614.343;1316;24814.76;532;10110.654 31;921;9;Bois-
Colombes;20 001 à 30

000;3134;77769.698;4173.288;2523;62864.586;2450;64926.422;789;19313.402
32;921;9;Bois-Colombes;30 001 à 50

000;3432;131764.313;11810.045;3142;121158.615;2704;108332.067;884;30608.799
33;921;9;Bois-Colombes;50 001 à 100

000;2959;206103.941;23204.089;2876;200782.535;2450;174162.774;703;35578.987
34;921;9;Bois-Colombes;+ de 100

000;1555;281840.341;54663.03;1543;279926.149;1450;220427.369;207;13003.585
35;921;9;Bois-
Colombes;Total;16451;754196.666;94178.048;11550;687865.691;12366;611762.201;3924;1173
36;922;12;Boulogne-Billancourt;0 à 10

000;10768;40002.583;-33.458;974;3941.402;4450;29940.156;2381;18655.672
37;922;12;Boulogne-Billancourt;10 001 à 12

000;1969;21659.057;-123.266;232;2551.872;1243;14079.259;665;8953.646
38;922;12;Boulogne-Billancourt;12 001 à 15

000;3236;43926.988;-152.929;332;4474.217;2418;34788.974;776;11996.521
39;922;12;Boulogne-Billancourt;15 001 à 20
000;6707;117461.317;1737.757;4501;79519.652;4753;89080.963;1998;37953.432
40;922;12;Boulogne-Billancourt;20 001 à 30
000;12137;302026.181;17302.813;10084;252134.685;9411;250058.266;2962;73197.927
41;922;12;Boulogne-Billancourt;30 001 à 50
000;15228;585365.85;55866.009;14017;540360.651;11742;465808.237;4054;140872.274
42;922;12;Boulogne-Billancourt;50 001 à 100
000;13637;960180.472;119188.991;13188;930708.97;10398;724664.033;3995;219637.731
43;922;12;Boulogne-Billancourt;+ de 100
000;8623;2053472.863;423642.607;8556;2043338.936;7338;1227384.39;1838;149660.254
44;922;12;Boulogne-
Billancourt;Total;72305;4124095.311;617428.524;51884;3857030.385;51753;2835804.278;1866
45;922;14;Bourg-la-Reine;0 à 10
000;1622;5882.583;-52.087;131;505.669;724;4096.433;298;2256.96 46;922;14;Bourg-la-
Reine;10 001 à 12 000;334;3658.707;-19.556;n.c.;n.c.;216;2422.886;114;1575.248
47;922;14;Bourg-la-Reine;12 001 à 15
000;452;6151.123;-21.911;43;590.319;338;4858.43;112;1679.158 48;922;14;Bourg-la-
Reine;15 001 à 20
000;1167;20428.084;268.551;779;13721.251;865;16136.377;327;6167.656
49;922;14;Bourg-la-Reine;20 001 à 30
000;1859;46338.317;2495.533;1514;37858.621;1371;36541.195;543;13740.587
50;922;14;Bourg-la-Reine;30 001 à 50
000;2408;92777.74;7851.31;2147;83162.381;1712;68154.844;769;28547.889
51;922;14;Bourg-la-Reine;50 001 à 100
000;2383;167835.486;18598.306;2325;163997.245;1754;123153.468;854;47998.737
52;922;14;Bourg-la-Reine;+ de 100
000;1072;193783.265;37255.872;n.c.;n.c.;919;131014.059;277;21181.717 53;922;14;Bourg-
la-
Reine;Total;11297;536855.305;66376.018;8033;492822.982;7899;386377.692;3294;123147.952
54;922;19;Châtenay-Malabry;0 à 10
000;3235;12494.587;-78.919;182;801.033;1574;9677.7;633;5295.393 55;922;19;Châtenay-
Malabry;10 001 à 12 000;670;7375.452;-17.158;n.c.;n.c.;491;5755.32;205;2621.431
56;922;19;Châtenay-Malabry;12 001 à 15
000;1072;14553.113;-38.776;66;897.516;841;12241.138;254;3855.549 57;922;19;Châtenay-
Malabry;15 001 à 20
000;2514;44117.27;580.716;1529;26938.906;1955;37409.367;647;12108.235
58;922;19;Châtenay-Malabry;20 001 à 30
000;3520;86622.425;4086.074;2599;64265.727;2868;76154.459;806;19439.413
59;922;19;Châtenay-Malabry;30 001 à 50
000;3653;140903.862;10768.613;3188;123809.741;2858;115319.123;1039;35461.37
60;922;19;Châtenay-Malabry;50 001 à 100
000;3007;207510.546;22992.464;2924;202225.413;2371;164778.688;917;49674.518
61;922;19;Châtenay-Malabry;+ de 100
000;974;153126.232;28596.158;n.c.;n.c.;842;110816.19;250;18024.285 62;922;19;Châtenay-

Malabry;Total;18645;666703.487;66889.172;11506;571787.733;13800;532151.985;4751;146480
 63;922;20;Châtillon;0 à 10
 000;3079;11194.856;-34.237;227;931.586;1345;8674.736;627;4987.251
 64;922;20;Châtillon;10 001 à 12
 000;634;7000.053;-26.124;n.c.;n.c.;432;4936.305;205;2823.761 65;922;20;Châtillon;12 001 à
 15 000;1066;14446.723;-29.873;100;1355.168;836;12255.492;233;3497.982
 66;922;20;Châtillon;15 001 à 20
 000;2400;42051.048;604.65;1559;27494.232;1786;33773.184;661;12751.259
 67;922;20;Châtillon;20 001 à 30
 000;4141;103124.712;5786.3;3402;85343.513;3194;85434.242;1097;27259.728
 68;922;20;Châtillon;30 001 à 50
 000;4856;186663.791;16275.181;4426;170726.631;3889;155340.157;1276;43728.558
 69;922;20;Châtillon;50 001 à 100
 000;3993;276840.758;31342.96;3923;272480.257;3389;240571.328;952;46118.059
 70;922;20;Châtillon;+ de 100
 000;1129;183632.623;34017.662;n.c.;n.c.;1024;139655.073;187;11106.325
 71;922;20;Châtillon;Total;21298;824954.564;87936.519;14817;741768.489;15895;680640.517;5
 72;922;22;Chaville;0 à 10
 000;1596;6221.737;12.115;119;555.868;774;5370.473;314;2584.22 73;922;22;Chaville;10
 001 à 12 000;300;3325.01;-10.037;n.c.;n.c.;219;2566.97;84;1148.508 74;922;22;Chaville;12
 001 à 15 000;576;7842.623;-25.577;63;860.268;455;6822.493;126;1912.164
 75;922;22;Chaville;15 001 à 20
 000;1269;22255.468;306.034;826;14622.595;981;18587.857;313;5974.617
 76;922;22;Chaville;20 001 à 30
 000;2301;57269.691;3111.054;1870;46853.678;1817;49245.925;528;13439.007
 77;922;22;Chaville;30 001 à 50
 000;2444;94095.42;7619.778;2164;83732.029;1844;73453.834;728;26106.056
 78;922;22;Chaville;50 001 à 100
 000;2311;161973.53;17643.953;2238;157296.254;1814;130065.188;664;35849.55
 79;922;22;Chaville;+ de 100
 000;1063;181687.46;34081.264;n.c.;n.c.;974;149015.774;185;13634.199
 80;922;22;Chaville;Total;11860;534670.939;62738.584;8365;485225.964;8878;435128.514;2942
 81;922;23;Clamart;0 à 10
 000;5067;19505.516;-167.936;332;1492.462;2222;13801.094;1213;9950.866
 82;922;23;Clamart;10 001 à 12
 000;1038;11432.678;-47.676;107;1180.371;676;7798.939;384;5246.829
 83;922;23;Clamart;12 001 à 15
 000;1667;22630.095;-75.563;132;1788.002;1269;18516.137;425;6524.416
 84;922;23;Clamart;15 001 à 20
 000;3728;65381.219;857.232;2286;40368.381;2765;52107.531;1088;20496.02
 85;922;23;Clamart;20 001 à 30
 000;5646;139232.791;6594.026;4277;106108.826;4220;110707.709;1646;40361.091
 86;922;23;Clamart;30 001 à 50
 000;5893;227210.139;16748.707;5171;200714.669;4319;169532.433;1991;69126.172
 87;922;23;Clamart;50 001 à 100

```

000;4977;346473.937;37143.625;4843;337949.298;3925;272002.669;1533;77314.528
88;922;23;Clamart;+ de 100
000;2175;411734.398;81470.769;2163;410029.718;1968;297492.461;408;27737.504
89;922;23;Clamart;Total;30191;1243600.773;142523.184;19311;1099631.727;21364;941958.97:
90;921;24;Clichy;0 à 10
000;8896;32166.164;-127.296;412;1792.366;3903;24234.647;1915;15820.341
91;921;24;Clichy;10 001 à 12
000;1587;17485.353;-40.228;n.c.;n.c.;1143;13652.437;478;6511.154 92;921;24;Clichy;12 001
à 15 000;2657;36044.685;-55.73;187;2537.448;2136;31339.557;585;9000.218
93;921;24;Clichy;15 001 à 20
000;5177;90461.601;1300.627;3036;53562.596;3997;75880.686;1341;24929.859
94;921;24;Clichy;20 001 à 30
000;7616;187442.83;9858.253;5843;144531.778;6499;172718.226;1390;32595.843
95;921;24;Clichy;30 001 à 50
000;6659;252492.635;21903.964;5873;224020.101;5784;230784.347;1197;38085.755
96;921;24;Clichy;50 001 à 100
000;3479;235341.47;27525.125;3379;228966.652;3120;215975.799;526;24133.435
97;921;24;Clichy;+ de 100
000;795;122801.587;23164.435;n.c.;n.c.;733;97920.784;100;5336.835
98;921;24;Clichy;Total;36866;974236.325;83529.15;19619;778553.154;27315;862506.483;7532;
99;921;25;Colombes;0 à 10
000;10590;38127.728;-194.874;511;2215.473;4763;29160.836;2199;17921.067
100;921;25;Colombes;10 001 à 12
000;1987;21867.272;-76.263;147;1618.41;1370;16134.571;651;8977.055

```

Appétissant !!!

On va le traiter pour utiliser ces données de façon plus lisible. Pour se faire on va utiliser la librairie spécialisée dans le traitement des données : `pandas`

Avant de se lancer dans le code, il faut comprendre qu'un csv n'est qu'un fichier texte. Vous allez donc recopier (Copier/Coller) le long texte au dessus de appétissant jusqu'à Index (inclus) dans un fichier qu'on appelle `donnees.csv`

Maintenant que c'est fait, on va rentrer les données dans un `DataFrame`. Le dataframe sera un tableau avec toutes les données

```
In [2]: #On importe La Librairie
import pandas as pd
```

```
In [3]: #On charge Les données.
dataframe = pd.read_csv("donnees.csv", sep = ";")
```

Que signifie `sep = ";"` ? Et pourquoi le met-on ? Reponse :

```
In [3]: #Regardons Le dataframe maintenant.
dataframe
```

Out[3]:

	Index	Dép.	Commune	Libellé de la commune	Revenu fiscal de référence par tranche (en euros)	Nombre de foyers fiscaux	Revenu fiscal de référence des foyers fiscaux	Impôt net (total)*	N _f in
0	0	922	2	Antony	0 à 10 000	5007	18765.611	-118.186	
1	1	922	2	Antony	10 001 à 12 000	988	10892.822	-85.364	
2	2	922	2	Antony	12 001 à 15 000	1586	21618.272	-59.781	
3	3	922	2	Antony	15 001 à 20 000	3678	64373.852	850.720	
4	4	922	2	Antony	20 001 à 30 000	6202	153271.296	7845.854	
...
96	96	921	24	Clichy	50 001 à 100 000	3479	235341.470	27525.125	
97	97	921	24	Clichy	+ de 100 000	795	122801.587	23164.435	
98	98	921	24	Clichy	Total	36866	974236.325	83529.150	
99	99	921	25	Colombes	0 à 10 000	10590	38127.728	-194.874	
100	100	921	25	Colombes	10 001 à 12 000	1987	21867.272	-76.263	

101 rows × 14 columns



Pour faire une boucle sur toutes les lignes on utilise la méthode de `DataFrame` :

`iterrows`

```
In [7]: for ligne in dataframe.iterrows():
        ...
```

```
In [8]: ligne
```

```
Out[8]: (100,
Index                                     100
Dép.                                       921
Commune                                    25
Libellé de la commune                      Colombes
Revenu fiscal de référence par tranche (en euros) 10 001 à 12 000
Nombre de foyers fiscaux                  1987
Revenu fiscal de référence des foyers fiscaux 21867.272
Impôt net (total)*                        -76.263
Nombre de foyers fiscaux imposés         147
Revenu fiscal de référence des foyers fiscaux imposés 1618.41
Traitements et salaires : Nombre de foyers concernés 1370
Traitements et salaires : Montant        16134.571
Retraites et pensions : Nombre de foyers concernés 651
Retraites et pensions :Montant           8977.055
Name: 100, dtype: object)
```

Quel est le format d'une ligne ? Comment accéder à la donnée `Commune` dans cette ligne ? Marquer le nombre d'essai/erreurs que vous avez fait avant de trouver la donnée. Pour moi au début il m'en a fallu 7.

```
In [9]: #Votre Code
```

Exercice

- Faites une boucle qui affiche toutes les "Commune"
- Faites une boucle qui affiche tous les "Revenu fiscal de référence des foyers fiscaux"

```
In [34]: print("Entrer votre code ici")
```

Entrer votre code ici

```
In [35]: print("Entrer votre code ici")
```

Entrer votre code ici

IV) Recherche dans une table

Définition : Recherche dans un table

La recherche dans une table consiste à obtenir les valeurs de certains descripteurs (champs) avec éventuellement des critères les concernant.

Exercice

- Faites une boucle qui affiche toutes les "Commune" dont le "Dép." est 922
- Faites une boucle qui affiche tous les "Revenu fiscal de référence des foyers fiscaux" de Clichy

Astuce : Ajouter un "if" dans votre boucle

```
In [36]: print("Entrer votre code ici")
```

Entrer votre code ici

```
In [37]: print("Entrer votre code ici")
```

Entrer votre code ici

Pandas est optimisé pour ne pas faire de boucle. La façon de faire est la suivante.

Qu'affiche la cellule suivante : (lancer la cellule suivante puis répondre)

```
In [ ]: dataframe[dataframe["Dép." ] == 922]
```

Si on ne veut avoir que les **Commune** on fait :

```
In [11]: dataframe[dataframe["Dép." ] == 922]["Commune"]
```

```
Out[11]: 0      2
         1      2
         2      2
         3      2
         4      2
         ..
        85     23
        86     23
        87     23
        88     23
        89     23
         Name: Commune, Length: 72, dtype: int64
```

Entrer la commande pour avoir "Impôt net (total)*" de Clichy

```
In [ ]:
```

Grace à la commande **list** et pas mal d'autre. Calculer la moyenne des "Impôt net (total)" de Clichy. Penser à multiplier par le nombre d'habitant ^^

```
In [ ]:
```

V) Recherche de Doublons

Exercice

- Faites une liste avec toutes les villes ?
- Faites une seconde liste qui répertorie tous les doublons.

```
In [ ]:
```

II) Utilisation de Pandas

Voici la Cheat Sheet de Pandas : https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf.

Essayer de tester pas mal de commande de pandas pour résoudre l'exercice suivant :

Exercice

- Trier la table par nombre croissant de foyers fiscaux
- Faire deux tables (dataframe) : une table pour Asnières et une table pour Clichy
- Faire une table regroupant ces deux tables

In []: