Airbnb

February 26, 2018

1 1. Variables intervenants dans ce fichier

```
In [1]: import pandas as pa
```

```
In [2]: T=pa.read_csv('/Users/fredericbro/Documents/Lycee(henrimoissan)/Prof (henri-moissan)/Ann
```

```
In [3]: T.head()
```

Out[3]:

	id_location	id_hote	quartier	latitude	longitude	type	\
0	3508970	17667828	Reuilly	48.848748	2.376042	Maison/appt entier	
1	13222966	12188988	Reuilly	48.847312	2.396370	Maison/appt entier	
2	7337128	6403392	Reuilly	48.846610	2.407639	Maison/appt entier	
3	5764597	14716027	Reuilly	48.846477	2.394853	Maison/appt entier	
4	7861852	41437694	Reuilly	48.839819	2.406376	Piece privee	

	prix	nuits_min	nb_avis	dernier_avis	avis_par_mois	disponibilite
0	80	2	10	19/03/2017	0.46	282
1	55	1	3	29/07/2016	0.30	0
2	104	1	12	30/04/2016	0.58	365
3	45	3	13	31/03/2017	1.21	2
4	35	2	1	13/10/2015	0.06	0

Pour déterminer les types de variables :

```
In [4]: T.dtypes
```

```
Out[4]: id_location        int64
        id_hote            int64
        quartier          object
        latitude         float64
        longitude        float64
        type              object
        prix               int64
        nuits_min          int64
        nb_avis            int64
        dernier_avis      object
        avis_par_mois    float64
        disponibilite      int64
        dtype: object
```

2 2. Résumés statistiques

```
In [5]: T.describe()
```

```
Out[5]:         id_location       id_hote      latitude     longitude          prix  \
        count  5.645000e+04  5.645000e+04  56450.000000  56450.000000  56450.000000
        mean   9.151755e+06  2.949098e+07     48.863797      2.344829     96.123649
        std    5.360795e+06  2.939809e+07      0.018484      0.033683     99.359560
        min    2.525000e+03  2.626000e+03     48.811993      2.222334      0.000000
        25%    4.569749e+06  7.126164e+06     48.850397      2.323577     55.000000
        50%    8.967390e+06  1.878864e+07     48.864619      2.348001     75.000000
        75%    1.382447e+07  4.123796e+07     48.879138      2.369989    105.750000
        max    1.805586e+07  1.242594e+08     48.905374      2.467834   7790.000000

                  nuits_min      nb_avis  avis_par_mois  disponibilite
        count  56450.000000  56450.000000   42301.000000   56450.000000
        mean       3.712383     14.936687       1.169358     146.042799
        std       60.507872     29.754285       1.378569     141.108874
        min        1.000000      0.000000       0.010000       0.000000
        25%        1.000000      0.000000       0.270000       0.000000
        50%        2.000000      4.000000       0.670000      94.000000
        75%        3.000000     15.000000       1.530000     292.000000
        max    10000.000000    488.000000      18.000000     365.000000
```

- Certains biens ont une disponibilité nulle.
- Des avis par mois ne sont pas renseignés

3 3. Revenu annuel engendré par un logment

```
In [6]: T1=T[(T['disponibilite']!=0) & (T['nb_avis']>0)]
```

```
In [7]: T1.count()
```

```
Out[7]: id_location      33358
        id_hote          33358
        quartier         33352
        latitude         33358
        longitude        33358
        type             33358
        prix             33358
        nuits_min        33358
        nb_avis          33358
        dernier_avis     33357
        avis_par_mois    33357
        disponibilite    33358
        dtype: int64
```

```
In [8]: T1.describe()
```

```
Out[8]:          id_location       id_hote      latitude     longitude          prix  \
        count   3.335800e+04  3.335800e+04  33358.000000  33358.000000  33358.000000
        mean    8.404524e+06  2.717184e+07     48.863465      2.344537     99.196445
        std     5.475130e+06  2.861701e+07      0.017790      0.032390     86.684861
        min     2.525000e+03  2.626000e+03     48.811993      2.227926     10.000000
        25%     3.370552e+06  6.028940e+06     48.851060      2.324800     55.000000
        50%     7.900280e+06  1.598678e+07     48.864056      2.347904     79.000000
        75%     1.353581e+07  3.856009e+07     48.877432      2.367272    110.000000
        max     1.796882e+07  1.228353e+08     48.901698      2.467046   5000.000000

                   nuits_min       nb_avis  avis_par_mois  disponibilite
        count   33358.000000  33358.000000   33357.000000   33358.000000
        mean        3.923407     23.316206       1.343011     198.019126
        std        78.447253     35.745289       1.457522     126.022373
        min         1.000000      1.000000       0.010000       1.000000
        25%         2.000000      4.000000       0.350000      76.000000
        50%         2.000000     10.000000       0.840000     216.000000
        75%         3.000000     27.000000       1.820000     317.000000
        max     10000.000000    488.000000      18.000000     365.000000
```

```
In [9]: T2=T1.describe()
```

```
In [10]: T2['nb_avis']*T2['prix']
```

```
Out[10]: count    1.112756e+09
         mean     2.312885e+03
         std      3.098575e+03
         min      1.000000e+01
         25%      2.200000e+02
         50%      7.900000e+02
         75%      2.970000e+03
         max      2.440000e+06
         dtype: float64
```

4 4. Graphiques

```
In [11]: import pylab as pl
```

On se limite qu'aux biens de disponibilité non nulle

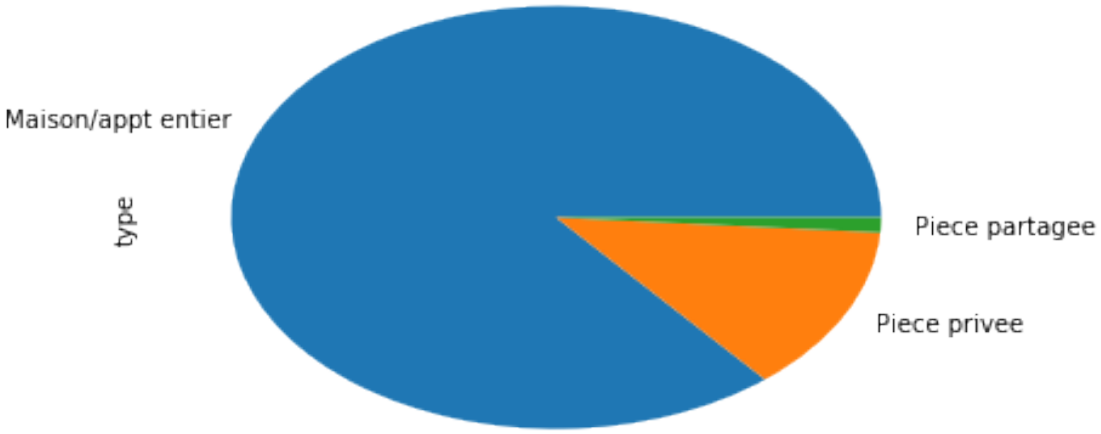
```
In [12]: T=T[T['disponibilite']!=0]
```

4.1 a) Type de location

```
In [13]: T['type'].value_counts()
```

```
Out[13]: Maison/appt entier    35496
         Piece privee           5258
         Piece partagee          466
         Name: type, dtype: int64
```

```
In [14]: T['type'].value_counts()/T['type'].count()

Out[14]: Maison/appt entier    0.861135
         Piece privee          0.127559
         Piece partagee        0.011305
         Name: type, dtype: float64

In [15]: T['type'].value_counts().plot.pie()

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x108d3b358>

In [16]: pl.show()
```



5 b) Occupation par quartier

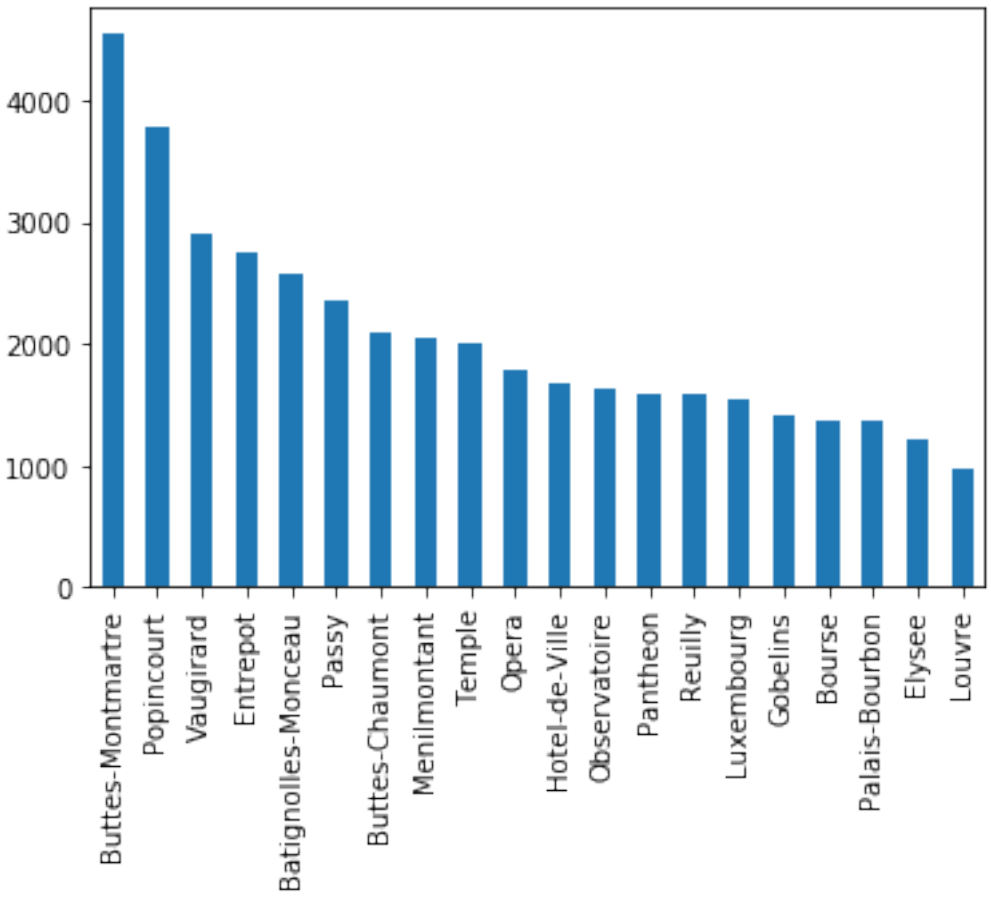
```
In [17]: T['quartier'].value_counts()

Out[17]: Buttes-Montmartre      4538
         Popincourt             3783
         Vaugirard              2905
         Entrepot               2758
         Batignolles-Monceau    2573
         Passy                  2365
         Buttes-Chaumont        2094
         Menilmontant           2047
         Temple                 2014
         Opera                  1775
         Hotel-de-Ville         1678
```

```
Observatoire            1627
Pantheon                1591
Reuilly                 1584
Luxembourg              1550
Gobelins                1409
Bourse                  1369
Palais-Bourbon          1364
Elysee                  1209
Louvre                   981
Name: quartier, dtype: int64
```

```
In [18]: T['quartier'].value_counts().plot(kind='bar')
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1115ad6d8>
```

```
In [19]: pl.show()
```



5.1 c) Hôte multi-loueurs

```
In [20]: T['id_hote'].value_counts()
```

```
Out[20]: 12984381     101
         2288803       70
         7612270       66
         3972699       66
         3943828       64
         11593703      62
         2667370       60
         3971743       58
         152242        54
         7642792       53
         288574        51
         97916688      50
         5044753       49
         13013633      47
         21630783      46
         24495283      46
         49869502      46
         1322370       45
         39922748      44
         23025598      44
         2107478       44
         789620        42
         67879895      41
         5056483       39
         64224220      38
         7608379       38
         49237581      38
         5027164       35
         17037121      35
         656386        35
                      ...
         12820295       1
         4939583        1
         8550138        1
         3702170        1
         22980352       1
         14137090       1
         10862339       1
         21995271       1
         22802185       1
         20439862       1
         119811857      1
         18628371       1
         97063967       1
         3749904        1
         27688731       1
         2580483        1
         81332569       1
```

```
30189344       1
43431714       1
25003812       1
15438059       1
732977         1
132916         1
15142709       1
8787766        1
34107193       1
5001673        1
11880252       1
31647195       1
26148864       1
Name: id_hote, Length: 33997, dtype: int64
```

```
In [21]: T['id_hote'].value_counts().value_counts().sort_index()
```

```
Out[21]: 1      31303
         2       1878
         3        384
         4        124
         5         68
         6         40
         7         28
         8         12
         9         16
         10         7
         11        14
         12         9
         13         7
         14        10
         15         8
         16         7
         17         7
         18         4
         19         2
         20         4
         21         8
         22         5
         23         1
         24         3
         25         3
         26         4
         27         1
         29         2
         30         2
         31         5
         34         1
```

```
35          3
38          3
39          1
41          1
42          1
44          3
45          1
46          3
47          1
49          1
50          1
51          1
53          1
54          1
58          1
60          1
62          1
64          1
66          2
70          1
101         1
Name: id_hote, dtype: int64
```

6 5. Représentation des locations

In [22]:
```
import folium
from folium import plugins
```

In [23]:
```
Loc=[]
for lat,long in zip(T['latitude'],T['longitude']):
    Loc.append((lat,long))
```

In [24]:
```
carte = folium.Map(location=[48.863797,2.344829],tiles='cartodbpositron', zoom_start=13
```

In [25]:
```
plugins.HeatMap(Loc,radius=10,blur=0).add_to(carte)
```

Out[25]:
```
<folium.plugins.heat_map.HeatMap at 0x11251e278>
```

In [26]:
```
carte
```

Out[26]:
```
<folium.folium.Map at 0x112026278>
```

In [27]:
```
#carte.save('C:\Lycee(henrimoissan)\Prof (henri-moissan)\Annee 2017\philippe\Activite\A
```