

# Rapport de projet Python

Un projet réalisé par : **BREGER Aymeric**, **BURY Jean-Baptiste**, **BENYEMNA Hamza**.

## I. Intro :

Le but de ce projet est de réaliser une étude statistique sur les données de la déclaration des valeurs foncières de biens sur les années précédentes, mais plus spécifiquement sur l'année 2020. Pour cela, nous utilisons le notebook **Jupyter** afin de traiter les données, réaliser des études, puis les afficher pour les interpréter. Ces données peuvent être exportées sous format html pour les intégrer dans une page web que l'on peut créer à partir de **Django**.

## II. Chargement et nettoyage des données :

Pour se faire, nous avons converti le fichier txt en csv bien plus pratique pour l'exploitation. Par ailleurs, le fichier était désordonné avec des séparateurs étranges et des colonnes entièrement vides. Nous avons rencontré beaucoup de problèmes quant à l'exploitation des données car nous devons palier à des données manquantes qui faussaient nos calculs. Pour un maximum d'ergonomie, nous avons d'abord éliminer tous les séparateurs '|' du fichier txt pour le convertir en csv. Ensuite nous avons remplacé tous les 'Nan' dans les colonnes nécessaires aux calculs par des '0'. Après cela, nous avons pu ajouter une colonne primordiale qu'est celle du **prix du m<sup>2</sup>**.

## III. Statistiques et interprétations :

### Prix moyen du m<sup>2</sup> :

Cette information est la plus intéressante selon nous. Nous l'avons décliné sous différentes formes afin d'obtenir des interprétations contrastées. Ainsi, nous avons des graphiques pourtant sur les moyennes du prix du m<sup>2</sup> par ville, département et type de biens. Au-delà de la moyenne, nous avons relevé les départements avec le prix du m<sup>2</sup> le plus cher, ce qui est le plus intéressant pour quelqu'un cherchant un bien à acheter selon un budget à respecter. Cela permet aussi de se situer selon sa ville, comparer cela avec son département et son type de biens si l'on cherche à vendre ou acheter un bien.

### Surface moyenne des terrains :

Nous avons fait la distinction de surface moyenne des terrains par ville en utilisant le code postal qui nous était données dans les fichiers, par département avec leurs codes de

département. Ainsi nous avons pu utiliser ces informations pour tracer différents graphiques et donc constater quels étaient les villes ou départements avec le plus de surface moyennes. Ces informations peuvent servir pour les personnes qui cherchent de grands espaces notamment pour les locaux industriels.

### **Valeur foncière moyenne :**

Comme pour la moyenne des terrains nous avons trier les valeurs foncières des biens selon leurs villes et selon les départements. Donc nous avons pu regarder quels étaient les départements avec la valeur foncière avec la meilleure opportunité selon la ville ou département. Ainsi, grâce à ces statistiques, on peut analyser quel bien entre dans notre budget ou non.

### **Nombres de biens fonciers :**

Une fois encore nous avons fait la différence entre les départements et villes Cependant cette fois ci nous avons compté les différents Type de bien grâce à la colonne '**Type local**'.

Nous avons ensuite ressorti en fonction des types de biens (Maison, Dépendance, appartement, etc..) les informations qui nous intéressaient comme la liste de ses derniers ou leurs ratios. Cette statistique permet de connaître la densité de maison, ... dans la ville ou le département qui nous intéresse. Pour les personnes qui cherchent un endroit plutôt paisible et moins dense ou plutôt un endroit 'vivant'.

### **Le nombre de transactions :**

Ici aussi nous avons fait la distinction entre les villes et les départements pour compter toutes les transactions dans chacun d'eux grâce à la 'Nature mutation'. Donc nous avons pu voir quel étaient les villes et départements avec le plus de transactions. Une ville ou un département avec le plus de transaction peut signifier un endroit avec de bonnes affaires à se faire ou un endroit où il est plus facile de vendre son bien.

### **Nombre de pièces moyennes :**

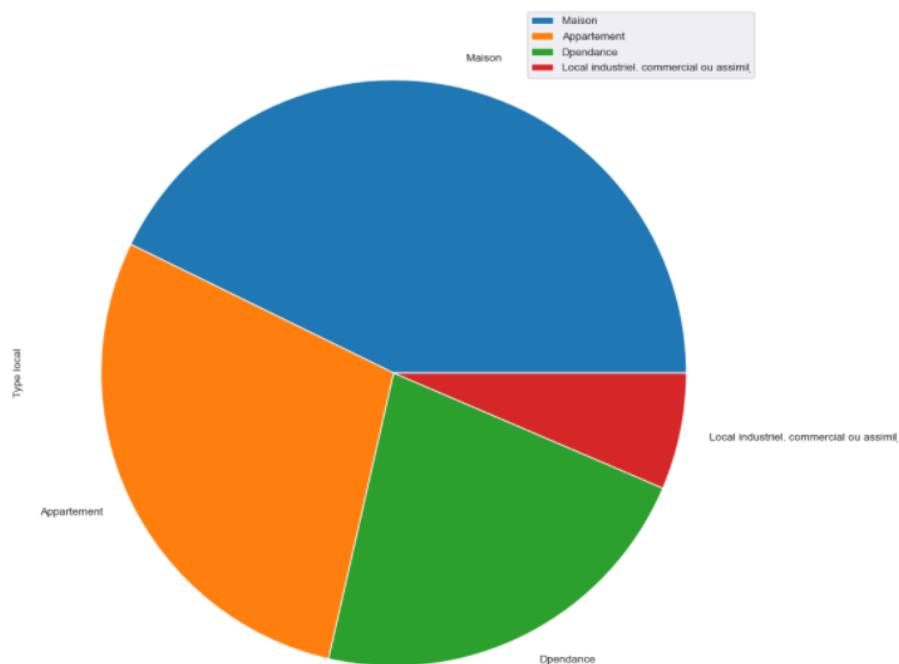
Par département nous avons pu voir que les moyennes étaient relativement proches les uns des autres contrairement aux villes qui ont une moyenne bien plus disperser les unes des autres. Ce module permet aux acheteurs ou vendeurs de trier les biens selon les besoins d'une famille ou groupe de personne qui compte y vivre. Plus il y a de personnes, plus on a besoin de pièces.

## IV. Affichages :

Grâce au statistique que nous avons établis aux préalables nous avons pu effectuer différents graphiques pour mieux imaginer notre interprétation.

Pour mieux illustrer nos propos en voici quelqu'un :

### Pie chart :



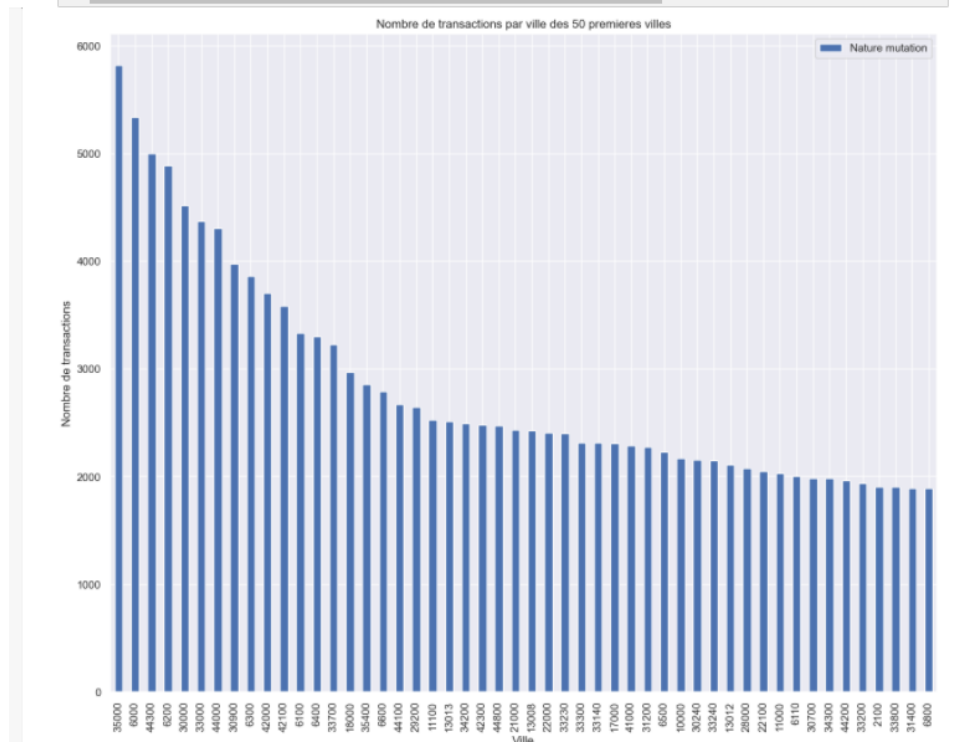
Nombre de biens fonciers par département

Le pie chart est le plus adapté pour représenter des proportions. C'est pourquoi nous l'avons choisi pour la représentation des types de biens en proportions.

### Histogramme :

### Transactions par villes

```
In [ ]: transaction_par_ville = nombre_transaction_par_ville.to_frame()
transaction_par_ville.head()
transaction_par_ville.sort_values(by = ["Nature mutation"], ascending=False)[0:50]
fig10 = transaction_par_ville.plot(kind='bar', title='Nombre de transactions par ville des 50 premières villes', ylabel='Nor
```



L'histogramme est l'outil le plus adapté à la plupart de nos modules statistiques car elles s'appuient sur des moyennes et des valeurs, qui représentent l'axe des ordonnées, selon une date, un type de biens, ... qui forme l'axe des abscisses.

### Map :

```

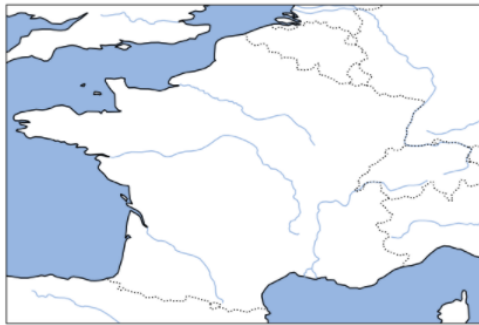
Entrée [285]: #code pour faire la map
fig = plt.figure(figsize=(7,7))
ax = fig.add_subplot(1, 1, 1, projection=ccrs.PlateCarree())
ax.set_extent([-5, 10, 42, 52])

ax.add_feature(cfeature.OCEAN.with_scale('50m'))
ax.add_feature(cfeature.COASTLINE.with_scale('50m'))
ax.add_feature(cfeature.RIVERS.with_scale('50m'))
ax.add_feature(cfeature.BORDERS.with_scale('50m'), linestyle=':')
ax.plot(df.long, df.lat, '.')
ax.set_title('France');

-----
AttributeError                                Traceback (most recent call last)
<ipython-input-285-af4eea9819b0> in <module>
      8 ax.add_feature(cfeature.RIVERS.with_scale('50m'))
      9 ax.add_feature(cfeature.BORDERS.with_scale('50m'), linestyle=':')
----> 10 ax.plot(df.long, df.lat, '.')
      11 ax.set_title('France');

~\anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5463         if self._info_axis._can_hold_identifiers_and_holds_name(name):
   5464             return self[name]
   5465         return object.__getattr__(self, name)
-> 5466
   5466         def __setattr__(self, name: str, value) -> None:
   5467
AttributeError: 'Series' object has no attribute 'long'

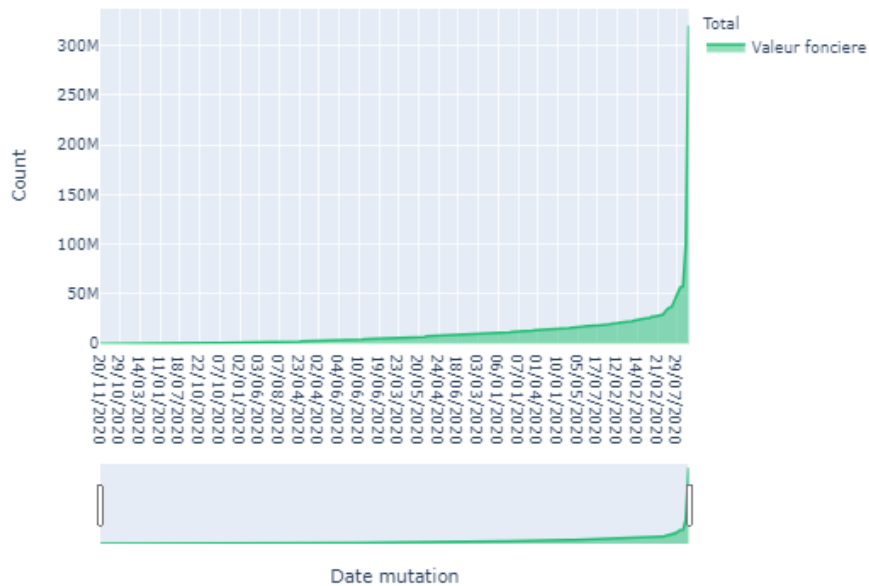
```



Nous avons essayé d’afficher la carte de France avec les informations sur les différents départements que nous avions mais après de (très) nombreuses tentatives nous avons décidé de consacrer du temps pour d’autres graphiques. En effet, cela est dû au fait que la librairie **geopandas** n’a pas pu s’installer, et ce, même depuis le Shell car il y avait des problèmes d’incompatibilité. De plus, nous ne pouvons pas procéder comme lors du TD8 car la France ne possède pas autant de fonctionnalité et de code country comme les Etats-Unis.

## Diagramme Interactif :

value over time in 2020



On peut grâce à ce graphique regarder sur une période de temps quels sont les dates ou les biens sont vendu. Ce graphique est pratique car on peut regarder en détail les données en zoomant sur la valeur qui nous intéresse pour la date souhaitée.

## V. Exportation HTLM :

Nous avons exporté notre fichier HTML deux façons différentes.

La première manière est d'utiliser la librairie **nbconvert** que l'on a installé directement à partir de l'**Anaconda Shell**.

La deuxième consiste en téléchargeant notre notebook depuis l'onglet 'File', 'Download as' puis on choisit l'extension que l'on souhaite (cad html).

## VI. Django :

Malgré un temps considérable passé sur des tutoriels, nous ne sommes pas parvenus à faire un Framework Django lié au notebook de Jupyter. En effet, nous savon que Django est un outil simple et ludique de développement de plateforme (web ou application). Nous avons réussi à créer un projet Django (qui est des plus rudimentaire), cependant nous n'arrivions pas à lier notre site à notre notebook exporté de Jupyter. Ainsi, nous avons préféré ne pas y

consacrer plus de temps afin d'améliorer la qualité de nos modules statistiques et de nos visuels.

## **VII. Conclusion :**

Nous avons réussi à tirer et interpréter toutes les informations qui nous paraissaient pertinentes dans les fichiers qui nous étaient mis à dispositions. Ainsi nous avons pu proposer des graphiques sur celle-ci pour en avoir une meilleure interprétation. Hormis le graphique sous forme de map et Django, où nous avons rencontré de grandes difficultés, nous avons remplis la totalité du "cahier des charges" qui nous étaient demander de fournir.

**De plus nous avons repartie les taches pour que chaque membre de l'équipe se retrouve avec un travail équitable.**

**Pourcentage de travail :**

**Hamza : 33% (calcul des modules et recherches Django principalement, recherche d'idées de modules statistiques)**

**Aymeric : 33% (tracer les histogrammes, histogrammes des valeurs corrélées entre 2019/2020, export HTML)**

**Jean-Baptiste : 33% (recherche Django, calcul de modules statistiques, tracer Graphique de 2020)**