Programmation III

Traitement et analyses des données de Demande Valeurs Foncières (DVF) de l'Ille-et-Vilaine

Valentin Foucher
Josselin Honoré

CONTEXTE

Le fichier DVF (Demandes de Valeurs Foncières) produit par le 'Ministère de l'Économie, des Finances et de la Souveraineté industrielle et numérique' est maintenant accessible en open data, en conformité avec un décret de décembre 2018 relatif à la publication électronique d'informations sur les valeurs foncières déclarées lors de mutations immobilières. La publication de ces informations vise à rendre les marchés fonciers et immobiliers plus transparents. Le fichier comportant des informations à caractère personnel, son utilisation nécessite certaines précautions :

1. ne doit pas permettre de ré-identifier les personnes concernées, de manière indirecte
2. ne doit pas permettre l'indexation des informations sur les moteurs de recherche

Ce jeu de données "Demandes de valeurs foncières", publié par la DGFiP, recense les transactions immobilières des 5 dernières années en métropole et DOM-TOM (sauf Alsace, Moselle et Mayotte). Les informations sont tirées des actes notariés et des données cadastrales.
Une documentation détaillée comportant 'Foire aux questions', Conditions générales d'utilisation', 'Information des personnes concernées par le traitement informatique' et 'Notice descriptive des fichiers de valeurs foncières' est accessible via le site https://www.data.gouv.fr/fr/datasets/demandes-de-valeurs-foncieres/#resources. Elle permet notamment de comprendre, à travers leur description détaillée, les champs renseignés et les données présentes dans la base de données mais aussi toute information relative à l'utilisation de la base.

Le premier jeu de données que nous avons utilisé dans le cadre de notre projet est dérivé du fichier DVF. Les fichiers sont produits par Etalab (département de la Direction Interministérielle du Numérique qui coordonne la conception et la mise en œuvre de la stratégie d'open data de l'État) dans un format normalisé et enrichi. Nous avons récupéré ces données dans des fichiers aux formats csv.
Des informations détaillées concernant les enrichissements faits par rapport à la base de données initiales ainsi que d'autres informations complémentaires peuvent être trouvées à cette adresse https://www.data.gouv.fr/fr/datasets/demandes-de-valeurs-foncieres-geolocalisees/.

Le second jeu de données que nous avons utilisé a été produit par Rennes Métropole. Les données disponibles dans plusieurs formats à cette adresse https://geo.data.gouv.fr/fr/datasets/63a0ef5654cb7cf3e28684e6b7dc7ca84eaf9799 ont été mises à jour jusqu'en 2021. Une documentation succincte présente sur le site nous indique que les données sont des données géospatiales décrivant par des polygones les bâtiments de

Rennes Métropole. Les bâtiments possèdent un identifiant unique composé d'un digramme (code commune + suite de 6 caractères numériques) et sont décrits par des points géolocalisés. Les données ont été collectées à partir des permis de construire (Ville de Rennes), de bases topographiques et orthophotographiques ou de restitution photogrammétrique. L'objectif de la constitution d'une telle base était de maintenir une base de référence locale pour les besoins de la ville. Rennes Métropole, précurseure de l'open data, la base créée en 2005 avait aussi pour objectif d'être mise à jour plus fréquemment que les sources de données externes comme la DGFiP. Nous avons utilisé le fichier dans son format geoJSON.

Nous donnerons plus de précisions sur les bases dans la suite du dossier.

INTRODUCTION

Dans le cadre de notre projet nous souhaitions utiliser ces données de DVF. Notre objectif était de représenter ces données de manière visuelle voire interactive, grâce aux compétences acquises pendant notre Master dans la manipulation des données. Mais comment développer un outil de visualisation de données servant à l'étude des marchés immobiliers et fonciers ?

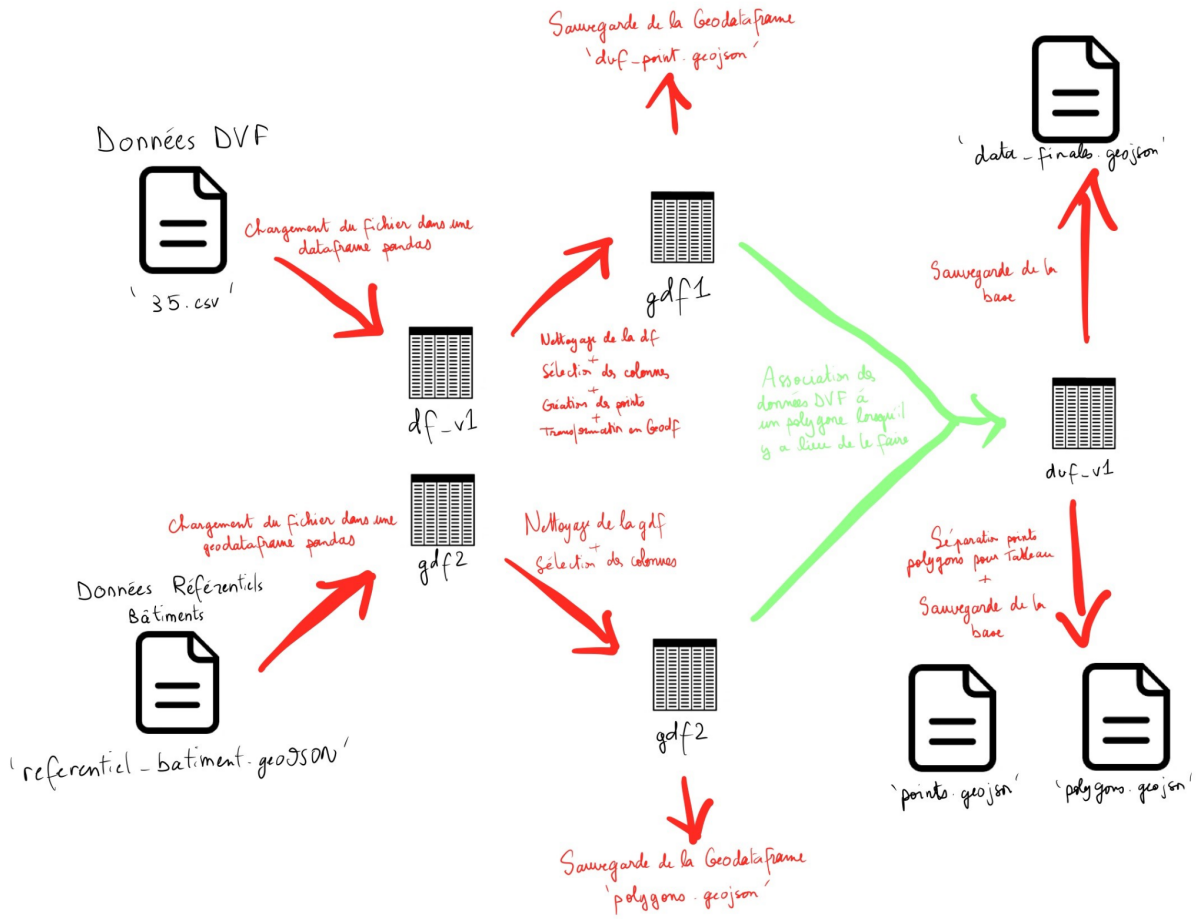
Après avoir mis en place notre projet via les canaux que nous souhaitions utiliser (planification via Trello, mise en place d'un repository GitHub) nous avons commencé par nous approprier les données avec lesquelles nous souhaitions travailler. Une première exploration sommaire de notre base nous a permis de dire qu'au vu de la quantité de données à notre disposition il était préférable de nous concentrer dans un premier temps sur la ville de Rennes. Nous avons ainsi planifié notre projet dans une optique de Minimum Viable Product (MVP), c'est-à-dire avec pour objectif de réaliser dans un premier temps une version fonctionnelle présentant seulement les éléments basiques de notre projet pour ensuite venir ajouter des options. Dans le même temps, la réalisation d'un état de l'art des solutions qui nous étaient accessibles pour visualiser les données nous a mené à nous familiariser avec la librairie python folium pour représenter sur une carte nos données. Cette première exploration sommaire ainsi que nos premiers tests avec folium nous ont poussé à chercher d'autres données géospatiales afin de les croiser avec nos données pour les représenter plus finement. Nous avons alors trouvé la base de données produite par Rennes Métropole représentant les bâtiments par des polygones de la librairie python shapely.

Notre première tentative de production visait donc à lier la base de données des valeurs foncières aux bâtiments afin de représenter ces données via folium. Cependant, les cartes

créées avec folium étaient particulièrement lourdes et donc très peu agréables à explorer. C'est pourquoi, nous avons finalement trouvé notre MVP lorsque nous avons opté pour l'utilisation de Tableau pour la partie visualisation de notre projet.

PRÉ-TRAITEMENT DES DONNÉES

La première partie de notre projet consiste en le traitement de nos deux bases de données afin de les rendre utilisables lors de la visualisation. En nous intéressant à la documentations et en explorant les données, nous avons pu établir un schéma décrivant les étapes clés de notre projet (cf. fig. 1) et déterminer les pré-traitements à effectuer sur les données.



Pour les données DVF, nous nous sommes concentrés sur le fichier propre au département de l'Ille-et-Vilaine disponible via à l'adresse https://files.data.gouv.fr/geo-dvf/latest/csv/2022/departements/. Le fichier comporte plus de 35000 lignes. L'exploration du fichier nous a permis de déterminer les dimensions de la base DVF qui nous intéressaient à savoir :

- 'id_mutation' : id de l'élément

- 'date_mutation' : dates de l'élément dans le format yyyy-mm-dd
- 'nature_mutation' : prend les valeurs 'Vente' ou 'Vente en l'état futur d'achèvement'
- 'valeur_fonciere' : valeur de l'élément
- 'adresse_numero', 'adresse_suffixe', 'adresse_nom_voie', 'adresse_code_voie', 'code_postal', 'code_commune', 'nom_commune', 'code_departement' : les informations relatives à l'adresse de l'élément (qui ne sera en réalité pas utilisée et éliminée)
- 'id_parcelle' : identifiant de parcelle compatible avec les fichiers cadastraux proposés par Etalab
- 'type_local' : prend les valeurs 'Maison', 'Appartement', 'Dépendance' ou 'Local industriel. commercial ou assimilé' et décrit le type de local
- 'longitude', 'latitude' : Géocodage latitude/longitude à la parcelle en coordonnées WGS-84

Dans un premier traitement simplifié qui servira d'exemple pour illustrer cette partie, nous avons sélectionné uniquement les dimensions 'valeur_fonciere', 'longitude' et 'latitude'. Dans un second traitement de nos données, nous avons ensuite enrichie la base créée pour la visualisation avec les dimmensions 'id_mutation', 'date_mutation', 'nature_mutation', 'adresse_numero', 'adresse_suffixe', 'adresse_nom_voie', 'adresse_code_voie', 'code_postal', 'code_commune', 'nom_commune', 'code_departement', 'id_parcelle', 'type_local', 'nature_mutation' et 'type_local'. Le but étant de pouvoir visualiser et analyser les données de valeur foncières sous différents paramètres.

Pour les données de référentiels des bâtiments à Rennes, nous avons utilisé le fichier geoJSON accessible via l'adresse mentionnée plus haut. L'exploration sommaire du fichier nous a permis de comprendre sa structure afin de déterminer quels étaient les champs qui nous intéressaient. Le fichier comporte plus de 145000 éléments. Nous avons choisi de ne garder pour cette base que le champ 'geometry' des éléments qui sont des types d'objets de la librairie shapely.

Notre objectif était alors d'associer aux valeurs foncières de l'Ille-et-Vilaine un forme géométrique lorsque c'est possible.

Avant de pouvoir réaliser une fonction qui ferait l'association, il était donc nécessaire de préparer nos données afin de pouvoir réaliser nos opérations dessus. Pour la première dataframe issue du chargement du fichier 35.csv, il a fallu via un script :
1. éliminer les colonnes qui ne comportaient aucune valeur (inser. ligne code ici)
2. éliminer les doublons
3. éliminer toutes les colonnes sauf 'valeur_fonciere', 'longitude' et 'latitude'
4. éliminer les lignes ayant des valeurs manquante dans au moins une des colonnes gardées
5. transformer la data frame en une geodata frame en se servant de 'longitude' et 'latitude' pour créer un Point shapely à mettre dans le champ 'geometry' de chaque élément
6. définir la projection pour la geodataframe via l'attribut crs

7. sauvegarder la geodataframe dans un fichier geoJSON pour pouvoir utiliser le résultat du pré-traitement

De la même manière, il a fallu préparer les données de la seconde data frame issue du chargement des données de référentiels bâtiments de la ville de Rennes. Nous avons donc réaliser un script permettant de :
1. sauvegarder dans un fichier les 3 premiers éléments de la geodataframe
2. ne garder que la colonne 'geometry'
3. supprimer les doublons
4. supprimer les éléments n'ayant pas de champ 'geometry' rempli
5. sauvegarder la geodataframe dans un fichier geoJSON pour pouvoir utiliser le résultat du pré-traitement

PRODUCTION DE NOTRE BASE

Une fois que nous avons préparé nos deux bases de données, il s'agissait d'associer aux données de valeurs foncières un polygone lorsque le point de la première base est compris dans le polygone de la deuxième. Il a fallu évidemment vérifier la compatibilité en termes de projection pour s'assurer que les comparaisons faites aient du sens. Avant de pouvoir réaliser cette étape, il a été nécessaire de transformer un petit peu les données contenant les polygones des bâtiments de Rennes. En effet, le format des objets contenus dans ce fichier est 'Polygon Z', c'est-à-dire que la 3ème dimension est présente. Il a donc fallu enlever cette dimension afin de pouvoir représenter plus facilement nos données ainsi que pour simplifier l'association de la base DVF aux polygones. Nous avons pour ce faire écrit la fonction 'convert_to_2d(multipolygon)' qui prend une liste de Polygones Z pour retourner une liste de polygon en 2D.

Notre script s'occupe ensuite de faire l'association et de remplacer le Point de la colonne 'geometry' de la data frame DVF si celui-ci est compris dans un polygone.

La base de données créée est ensuite sauvegardée dans un fichier geoJSON qui va nous servir pour la visualisation dans Tableau.

Lors de nos premiers tests sur Tableau, nous n'avons pas pu visualiser les données produites car la solution ne prenait pas en compte les géometry mélangées. Il a donc fallu adapter notre code pour qu'il génère après l'association 2 fichiers différents, un pour les données possédant un point pour les géolocaliser, un autre pour les données géolocalisées par un polygone.

PROBLÈME RENCONTRÉ DURANT LE PROJET

Durant le prétraitement ou dans la production de notre base, nous avons rencontré un problème d'optimisation. Le nombre d'éléments dans les bases est conséquent ce qui a rendu leurs analyses et traitements complexes. Les ordinateurs que nous avons à dispositions se rapprochent plus des notebook que d'ordinateurs destinés à la programmation. Cela à résulter en une certaine difficulté à traiter les bases d'un seul coup. En effet, pour ce qui est de l'association des données de valeurs foncières à un polygone lorsque le point de la première base est compris dans le polygone de la deuxième, cela nous a pris plus d'une nuit pour obtenir un résultat. Nous avons pour ce qui est des autres modifications des bases, d'abord travaillé avec des échantillons pour une question de rapidité d'exécution, et une fois un résultat satisfaisant obtenu, effectué le traitement sur la base de données.

Nous avons aussi rencontré des problèmes pour utiliser le fichier généré dans Tableau. En effet, la solution de visualisation ne supporte pas encore les fichiers avec des objets géométriques de différents types. Nous avons alors dû séparer dans deux fichiers les valeurs associées à des points qui ne correspondaient pas à des bâtiments décrits dans la base de données des référentiels bâtiments de Rennes ainsi que les valeurs qui ont bien été associées à des polygones.

VISUALISATION AVEC TABLEAU

Une fois notre base créée, nous souhaitions représenter nos données via la solution Tableau. Cela nous a semblé être un choix judicieux tant cet outil se démocratise dans les métiers utilisant la data. Seulement nous avons rencontré quelques difficultés lors de la prise en main notamment pour pouvoir faire reconnaître nos données au format geoJSON (de manière à pouvoir afficher nos formes).

Enfin, la solution Tableau met à disposition des développeurs des solutions leur permettant d'inclure les visualisations dans des web applications ou sur des sites web. Dans une logique de MVP, cette étape pourrait intervenir avant les pistes d'évolution citées par la suite.

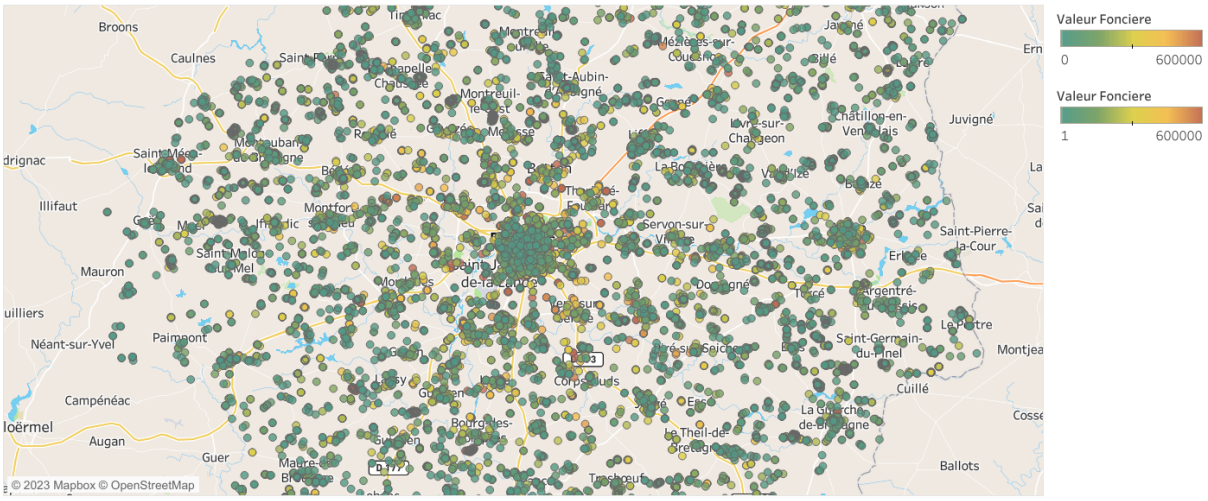
EVOLUTION

De manière à faire évoluer encore notre projet, la prochaine étape est le passage à l'échelle de plusieurs départements. Le fonctionnement serait le même pour le traitement que nous avons fait au niveau d'un département. La difficulté résiderait alors dans la recherche de bases de données de référentiels de bâtiments. Pour contourner cette erreur, nous avons sélectionné dans le traitement plus avancé la colonne 'id_parcelle' qui est une colonne qui a été ajoutée par Etalab aux données DVF originales dans le but de faire correspondre cet identifiant avec les plans cadastraux des parcelles aussi présents en open data sur les sites du gouvernement. La valeur foncière affichée sur l'outil de visualisation Tableau serait donc limitée à la parcelle et non plus au bâtiment lorsqu'il y aurait correspondance. On pourrait imaginer un traitement qui permettrait d'associer les valeurs aux bâtiments en priorité lorsqu'il y a des données disponibles et sinon aux parcelles des plans cadastraux.

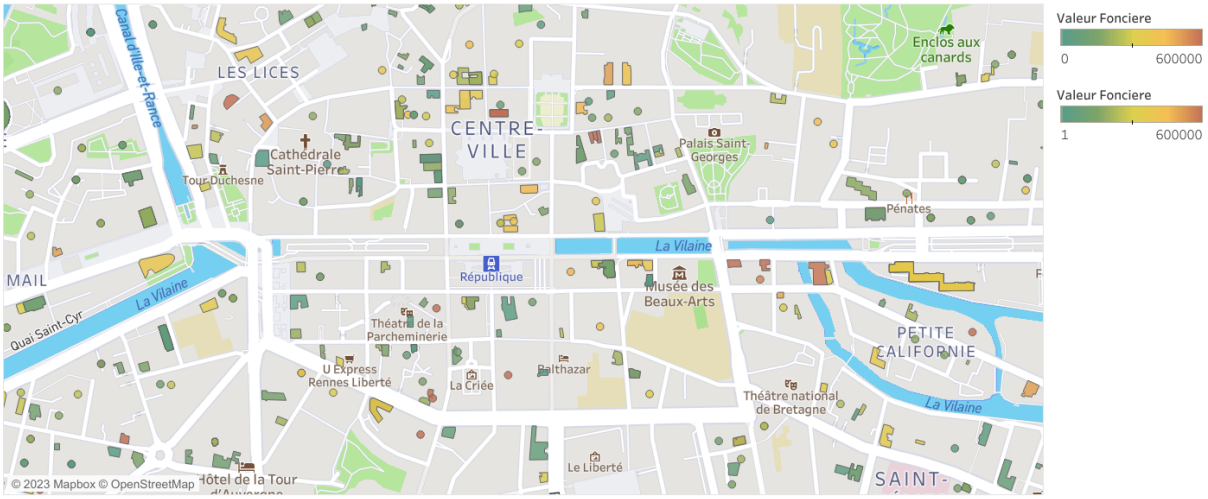
Un autre point clé pour passer à l'échelle le projet serait de rajouter des filtres selon le 'niveau de zoom'. C'est une fonctionnalité qui est accessible dans Tableau. Si l'on s'intéresse toujours à la représentation des valeurs foncières, il faudrait imaginer un traitement supplémentaire qui associerait la moyenne des valeurs foncières d'un département à une forme qui représente le département. De manière à pouvoir visualiser les différences de prix moyen entre départements à l'échelle de la France par exemple. La même logique pourrait être appliquée pour les régions voire même les pays. Il faudrait alors gérer en plus l'unité dans laquelle seraient exprimées les valeurs foncières. Dans l'outil de visualisation tableau, il serait alors possible de cliquer sur la France par exemple pour qu'un zoom sur les formes représentant les régions apparaisse. Puis en cliquant sur les régions un filtre sur les formes des départements apparaît. Et enfin un clique sur un département affichera le niveau de visualisation que nous avons mis en place.

Un point d'amélioration sur lequel nous n'avons pas eu réellement le temps de travailler mais qui aurait pu être intéressant est la définition d'un niveau de granularité intermédiaire entre le département et le niveau que nous avons mis en place. Le but serait d'avoir des quartiers par exemple pour pouvoir visualiser les différences de valeurs entre les quartiers d'une ville.

plage des prix, polygones



plage des prix, polygones



plage des prix, polygones