III) Réalisation

1) Récupération des données

De nos jours, une façon possible d’automatiser la récupération de données est le web scraping. D’après Wikipédia, le web scraping est une technique d'extraction du contenu de sites web, via un script ou un programme, dans le but de le transformer pour permettre son utilisation dans un autre contexte. Grâce à cette technique, lorsque les pages ont une même structure, il suffit de disposer de tous les liens auxquels nous voulons soutirer des données. Afin de produire notre projet final, basé sur la visualisation de données spatiales associées aux banques coopératives en France, nous avons dû créer deux jeux de données à l’aide de cette méthode.

Tout d’abord nous avons récupéré les données sur toutes les banques présentées auparavant. Cette base de données a été nommée **bdd\_coordonnees\_banques2022***.* Elle recense le nom de la banque, son type, son adresse ainsi que sa longitude et latitude. Ensuite, une deuxième récupération de données été nécessaire. En effet, les données socio-économiques pour chaque zone d'emploi de la France métropolitaine de 2020 ont été récupérées et stockées dans un fichier nommé **bdd\_social\_ze2020**. Toutes les variables obtenus sont présentés dans la partie *Présentation des données.*

Ce qui suit va vous présenter étapes par étapes notre démarche lors de la création de ses deux fichiers. Nous commencerons par expliquer le processus mise en place pour la récupération des coordonnées des banques pour ensuite vous expliquer la récupération des données socio-économiques.

Banques

La première tâche à effectuer lors de notre projet était l’une des plus importante : nous devions récupérer les données, sur lesquelles nous devions travailler. Cinq banques en France métropolitaine sont à récupérer ici depuis leur site officiel. Cependant pour la BNP Paribas, nous n'avons pas pu récupérer les données sur leur site donc nous avons dû utiliser un autre site où les données étaient disponibles. Vous trouverez ce lien en *Référence X.* Pour être plus précis, l’objectif premier était d’uniquement récupérer l’adresse de toutes les agences, banque par banque.

Comme précisé plus haut, nous avons décidé de faire l’intégralité de notre projet en langage ***R***. Nous avons dû installer différentes librairies (nommé *package* dans ce langage, nous utiliserons cette appellation plus tard dans le rapport). La première librairie utile pour le web scraping est ***rvest***. Elle aide à l’extraction d’information sur des pages web et facilite l’expression des tâches courantes de web scraping. Une deuxième bibliothèque utile pour notre récupération de données est le package ***plyr***. Il regroupe un ensemble d’outils qui permettent de résoudre des problèmes tels que diviser une grosse structure de données en morceaux homogènes, appliquer une fonction à chaque morceau et enfin combiner tous les résultats. Troisièmement, nous avons utilisé la librairie ***dplyr***, elle fournit une grammaire de manipulation de données, fournissant un ensemble de verbes aidant à la résolution de manipulation de structure de données. Une autre librairie utile a été ***BanR***. Ce package utilise l'api BAN (Base Adresse Nationale). BAN est un jeu de données public des adresses françaises produit par OpenStreetMap, la Poste, l'IGN et Etalab. Ce package comporte plusieurs fonctions pour trouver une longitude et latitude à partir d'une adresse et inversement. Enfin, vous comprendrez un peu plus loin mais nous avons eu besoin d’un package de manipulation de chaine de caractères, nous nous sommes servis de ***stringr*** ainsi que ***stringi***.

A présent nous avons présentés presque tous les outils nécessaires pour pouvoir commencer le web scraping, un dernier et pas des moindres est l’extension de Google ***selectorGadge*t**. Il facilite la génération et la découverte de sélecteur CSS sur des sites complexes. En plus simple, il permet d’attribuer une espèce d’adresse à un élément remarquable d’un site internet (un titre ou un tableau). Lorsque l’on lance l’extension, une boîte s’ouvre en bas à droite de l’écran, il suffit de cliquer sur l’élément de la page que vous voulez récupérer. Si un sélecteur correspond, un cadre vert apparaitra et nous pouvons récupérer l’adresse de l’élément. Ci-dessous, vous trouverez un exemple pour sélectionner les adresses du Crédit Agricole :

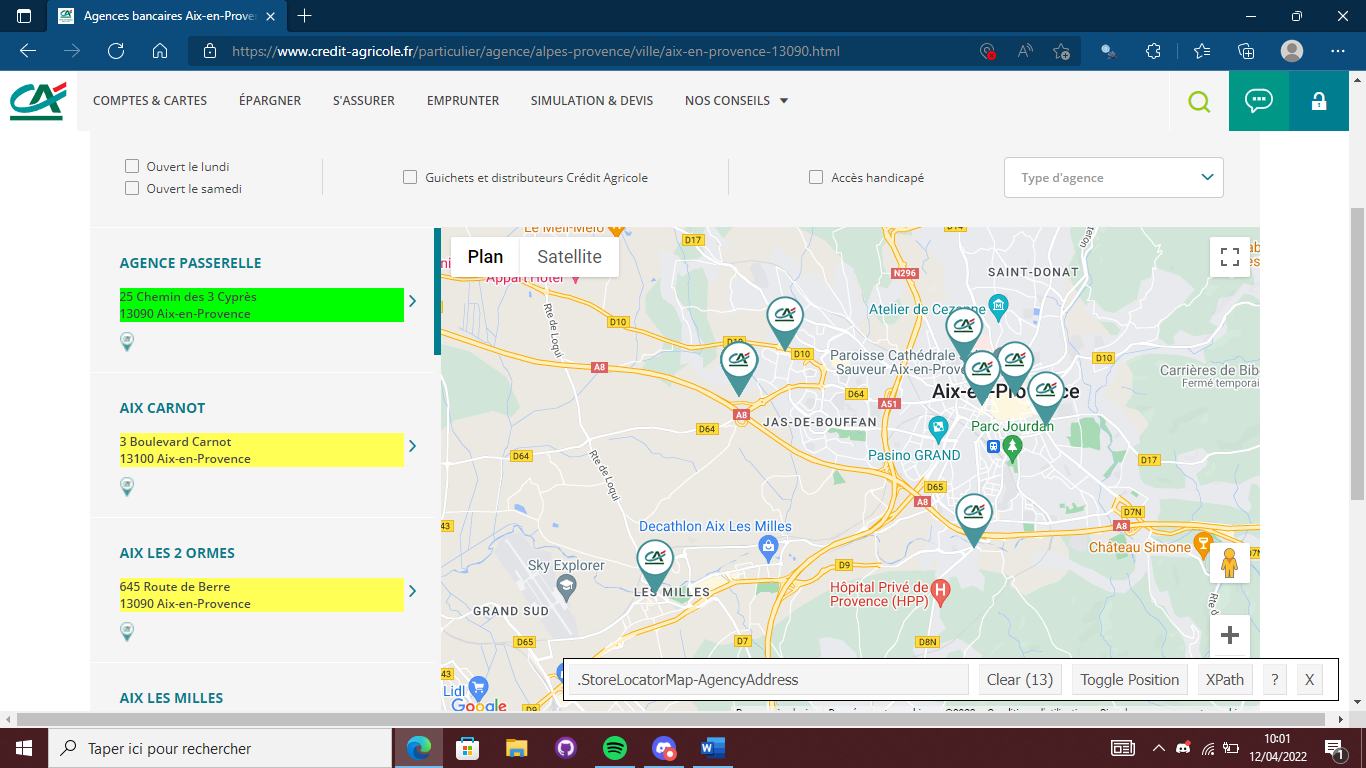


Figure X : Exemple de récupération d’adresse pour le Crédit Agricole à l’aide de l’outil selectorGagnet

Sur l’exemple ci-dessus, vous pouvez voir que la première adresse (en vert) a été sectionnée, les adresses qui suivent (en jaune) sont mises en évidence car l’extension reconnaît que ce sélecteur est du même type que celui sélectionné en vert. Dans la boîte en bas à droite, on retrouve le nom de l’adresse sélectionné : « .StoreLocatorMap-AgencyAddress » et le nombre de sélecteur similaire disponible sur cette page (ici 13). Ainsi en utilisant ce nom nous allons pouvoir récupérer toutes les informations sélectionnées en vert et jaune, c’est-à-dire les 13 adresses.

Afin d’automatiser la récupération de données, nous avions besoin de comprendre la logique qui reliait tous les liens. En effet, les pages ayant toutes à peu près la même structure, une fois que l’on possède une boucle générant chaque lien, il est assez aisé de récupérer les données en même temps. Seule subtilité, pour chaque banques nous avons dû adopter une technique différente. Ce qui suit vous explique donc banque par banque la récupération de lien.

Le site du Crédit Agricole présente les régions puis dans chaque région les villes par ordre alphabétique où contiennent au moins une agence. Lorsque les villes contiennent des arrondissements, ils sont détaillées (Paris 01 par exemple). Ainsi pour mieux comprendre, voici l’url qui est sous la forme suivante :

**https://www.credit-agricole.fr/particulier/agence/**region**/ville/**ville-code postal**.html**

Dans ce cas, nous avons dû utiliser un fichier annexe contenant les codes postaux associés aux villes de France. Sur le site certaines villes ont plusieurs codes postaux ce qui entrainent des doublons d’adresses. Ses doublons seront bien sur effacés à la fin de la récupération.

Les sites de la Banque Populaire et de la Société Générale sont construits de la même manière. Les urls sont de la forme :

**https://agences.**nom de la banque.fr**/banque-assurance/agences-**departement-numero du département

Nous avons fait un web scraping pour récupérer la liste de chaque département ainsi que leur numéro. A partir de ce lien, nous pouvons récupérer toutes les adresses d’un département et ainsi récupérer toutes les adresses des agences présents pour ses banques.

Le site du Crédit Mutuel est composé d’une liste de département puis des villes contenant une agence dans un département sélectionné. Dans une troisième page imbriquée nous avons une liste d’adresses d’agences de la ville choisie ainsi que d’autres aux alentours. Ça ne pose pas de problèmes en revanche cela entraine une quantité très importante de doublons d’adresses.

Le Crédit Mutuel de Bretagne est composé de plusieurs pages également imbriquée, les noms des départements puis le nom des villes avec leur codes postaux et enfin sur une troisième page l’adresse exacte de l’agence. Nous avons dû formater l’url pour accéder à la page des villes et à celle de l’adresse.

Pour la BNP Paribas, le site utilisé lui aussi divise les agences en département. Si le nombre d’agences est important, la page est divisée en plusieurs pages. L’url est sous la forme suivante :

**https://www.moneyvox.fr/pratique/agences/bnp-paribas/**numero du département/ s’il y a plusieurs pages il faut rajouter un « numéro de la page/ »

Après avoir réalisé ce web scraping nous avons fait un premier tri des adresses car nous avions un nombre conséquent de doublons. Cela a été causé par le format du site ou encore du fait que l’agence était partagée en différente parties (par exemple : l’agence à Montélimar 23 rue Raymond Daujat qui a une partie « classique » et une partie réservée aux entreprises et apparait donc deux fois dans les adresses récupérées).

Tous les liens générés et toutes les adresses récupérés par banques, nous nous retrouvions avec une base de données comportant le nom de la banque, son type ainsi que son adresse. Or pour ce projet nous voulions et nous avions besoin des points exactes des banques, c’est-à-dire nous avions besoin de la longitude et la latitude de chaque adresses. Une dernière étape était donc primordiale.

Nous avons utilisé le package ***BanR*** qui, grâce la fonction geocode, nous permet de convertir une adresse française en longitude et latitude. Cependant, il y a des fois où cette fonction ne nous renvoyée pas de résultats ou encore nous renvoyée une fausse conversion car elle ne reconnaissait pas l’adresse entrée en paramètre. Ainsi pour chaque banque il a fallu modifier certaines des adresses manuellement.

Une fois tout cela réalisé nous avons dû faire un deuxième tri des doublons d’adresses. En effet, encore certaines adresses de la même banque étaient situées au même endroit du fait que certaines adresses changeaient seulement de codes postaux mais désignées la même agence. Ainsi pour palier à cette erreur nous avons utilisé le couple longitude-latitude pour que ce couple soit unique. Nous obtenons après cela, toutes banques confondus, 14704 agences différentes en France métropolitaine.

Socio-économique

Une autre tâche à réaliser était la récolte de toutes nos variables socio-économique. Nous disposions déjà de données concernant certaines variables cependant nous avons dû les remettre au bout du jour pour que cela correspondent le plus possible aux données actuelles.

Dans le même principe que la récupération des coordonnées des banques nous avons récolté les différentes variables socio-économiques. Cette récupération est divisée en quatre parties.

Premièrement, 23 variables ont été récolté sur le site de l’Insee caractérisant les revenus et pauvreté des ménages en 2019. Dans le même principe que la récupération des banques nous avons récolté tous les liens pour nous permettre d’automatiser la récupération de ses 23 variables socio-économique pour chaque zone d’emploi. Voici la forme de l’url :

**https://www.insee.fr/fr/statistiques/6037462?geo=ZE2020-**zoned’emploi

Comme vous pouvez le voir ce lien n’a pas de difficultés particulières nous devions simplement disposer de toutes les zones d’emplois et réalisé une boucle récupérant toutes les données. Ci- dessous un exemple réalisée pour la première zone d’emploi.

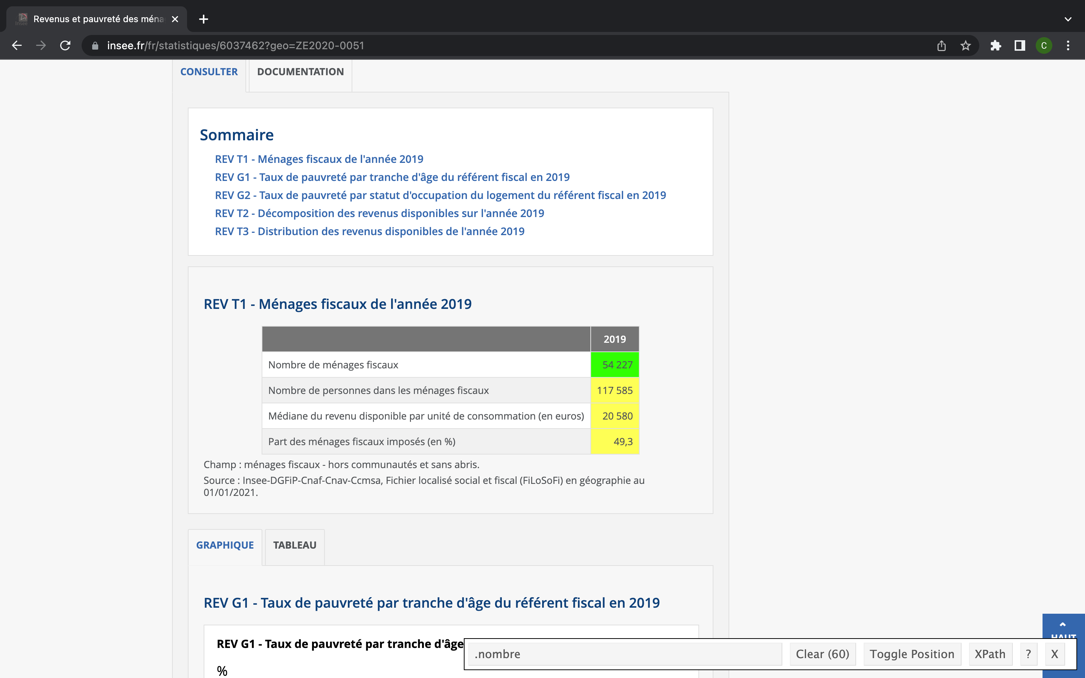


Figure X : Exemple de la récupération des revenus et pauvreté des ménages en 2019 pour la zone d’emploi 0051 à l’aide de l’outil selectorGagnet

Une fois les données récupérées nous avons dû les modifier pour obtenir la forme et le type que nous désirions. Tout cela a été possible à l’aide du package ***stringr.***

Deuxièmement, nous avons dû récupérer 7 variables caractérisant les emplois datant de 2018 présentes sur un fichier Excel déjà existant trouvé sur l’Insee. Vous pouvez retrouver ce fichier au lien en *Référence X*. Nous ne voulions que certaines informations présentes dans ce fichier c’est pourquoi nous avons réalisé un petit code qui récupérer uniquement les variables qui nous paraissaient le plus important.

Troisièmement, la récolte des données concernant le taux de chômage en 2020 a été réalisé de la même manière que dit précédemment pour les variables caractérisant les emplois. Il existe déjà un fichier Excel sur l’Insee avec tous les taux de chômage depuis 2003 jusqu’à 2020. Vous pouvez retrouver ce fichier au lien en *Référence X.*

Enfin, concernant la récupération des informations sur la population active en 2017, nous l’avons réalisé exactement de la même manière que pour les variables sur le revenus et pauvreté des ménages. Voici l’url :

**https://www.insee.fr/fr/statistiques/4515512?sommaire=4515574&geo=ZE2020-**zoned’emploi

Une fois avoir réalisé notre web scraping et nettoyage des données pour chaque types de variables nous avons réalisé une concaténation de toutes ses données et nous obtenons le jeu de données ci-dessous déjà présenté dans *Présentation* :

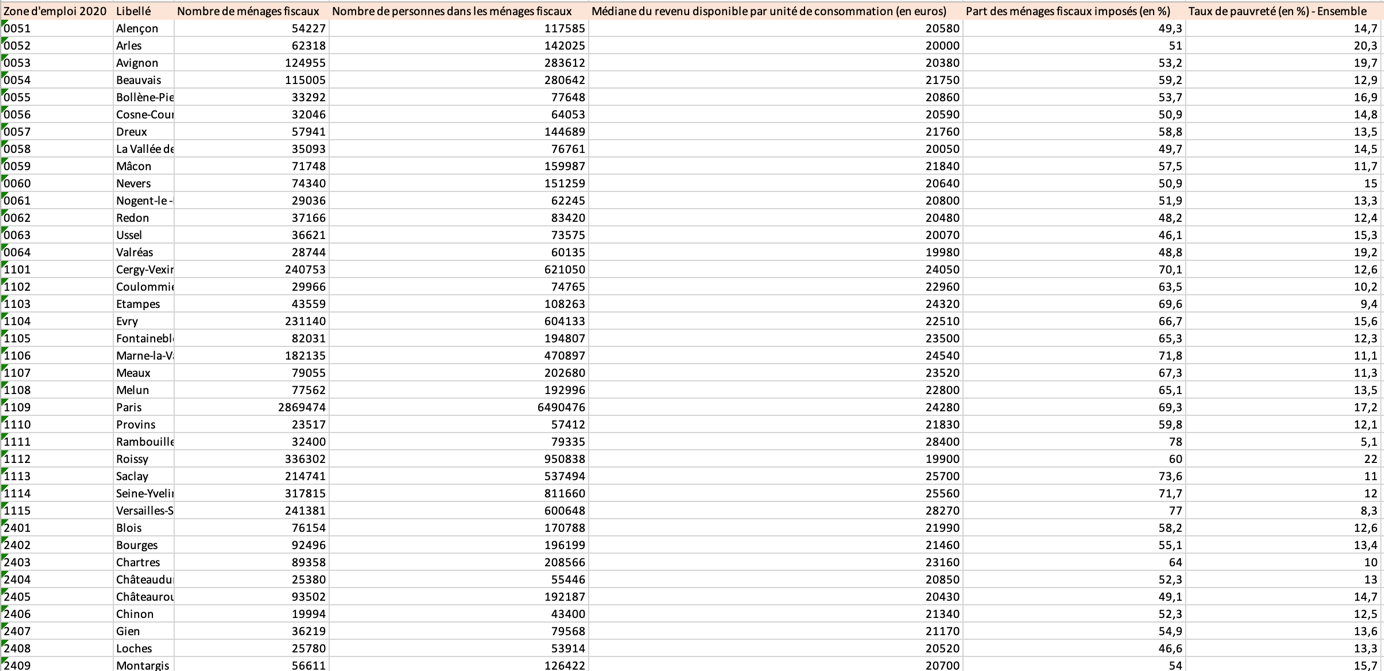


Figure X : Data frame caractérisant les données socio-économiques

Toutes les données maintenant récoltées et archivées dans les bases de données, nous avons pu commencer à jouer avec.