**Création des cartes**

Comme énoncé lors de l’introduction, ce projet vise à créer une application de visualisation de données sur la position géographique des banques en fonction de données socio-économiques à l’échelle des zones d’emploi en France métropolitaine.

Pour cela, nous avons besoin de tracer des cartes, et pour tracer des cartes, l’un des moyens les plus utilisés est d’utiliser des *shapefiles.* Un shapefile (ou « fichier de forme ») est une combinaison de fichiers permettant de stocker les informations relatives aux tracés des cartes telle que l’emplacement, la forme ou les attributs des entités géographiques. Étant donné que nous voulons étudier l’emplacement géographique des banques au niveau des zones d’emplois, nous avons récupéré sur le site de l’**Insee** le shapefile relatif aux tracés des zones d’emploi pour l’année 2020.

Une fois cet ensemble de fichiers récupéré, nous avons dû le formater afin de récupérer seulement les données dont nous avions besoin. En effet, le découpage en zone d’emploi se fait en France métropolitaine et dans les DROM-COM. Comme nous considérons seulement la France métropolitaine, nous avons enlevé du jeu de données toutes les données faisant référence à l’outre-mer.

Après cela, nous avons pu nous atteler au tracé de nos premières cartes. Pour ce faire, nous avons utilisé le package ***sf***de ***R***. Ce package combine les fonctionnalités des packages ***sp***, ***rgdal***,***rgeos*** permettant d’importer, de manipuler et de transformer les données spatiales. Il propose des objets plus simples dont la manipulation est plus aisée. L’une de ses grandes forces est qu’il est compatible avec tous les opérateurs du ***tidyverse****.*

Pour mieux comprendre, comme vous pouvez vous référer à la Figure X, les objets de classe ***sf*** sont des data frames dont la dernière colonne est composée de *géométries*. Cette colonne est de classe *sfc* (*simple feature column*) et chaque la ligne de la colonne est de classe *sfg* (*simple feature geometry*). Plus généralement, cela correspond à des listes de coordonnées (*longitude / latitude)*. Ce format est très pratique car il permet de lier dans un même objet les données ainsi que les coordonnées géographiques associées (*géométries)*.

Une image contenant texte

Description générée automatiquement

Figure X : Représentation de l’objet de classe***sf*** faisant référence aux zones d’emplois 2020

Dans le shapefile que nous avons récupéré, nous voyons sur la Figure X que nous avons six colonnes.

Nous y retrouvons :

* Le codes des communes
* Le nom des communes,
* Le code de la zone d’emploi associée,
* Le nom de la zone d’emploi,
* La partie régionale de la zone d’emploi trans-régionale
* Les *géométries*,correspondant au tracé des frontières de chaque commune

Ainsi lorsque nous traçons la carte en fonction de ces données, vous pouvez constater sur la Figure X1 que nous avons le tracé des frontières de chaque commune. Toutefois, nous pouvons colorer la carte par zone d’emploi.

Notre objectif est donc le suivant : nous devons avoir une carte où les frontières seraient le contour de chaque bloc de couleur que nous pouvons voir sur la carte ci-dessous.

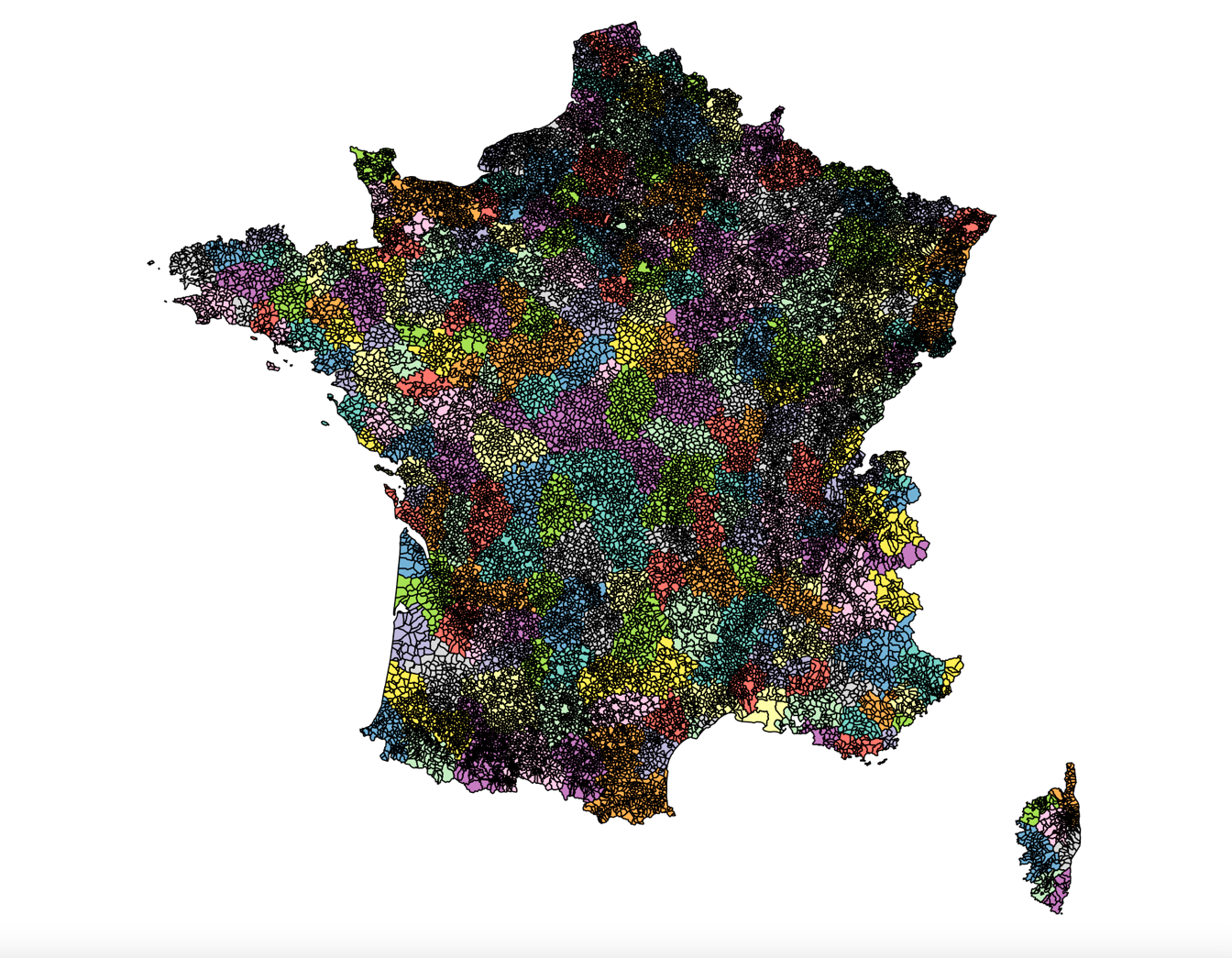


Figure X1 : Tracé des communes colorées par zone d’emplois

Nous avons donc dû retravailler sur ce fichier de données de sorte à obtenir seulement une data frame composée de deux colonnes. La première composée de la liste des codes de zones d’emplois et la deuxième composée de *géométries* répertoriant les coordonnées géographiques des contours de chaque zone d’emploi.

Ce qu’il faut comprendre ici, est que le logiciel lit les *géométries* comme des polygones. C’est-à-dire que chaque commune est considérée comme un polygone. Ainsi, au sein de chaque bloc de couleur de la carte, nous devons combiner tous les polygones de sorte à n’en obtenir qu’un seul qui serait celui de la zone d’emploi associée.

Comme expliqué un peu plus haut, les objets ***sf*** sont compatibles avec les opérateurs du ***tidyverse***. Nous avons donc utilisé des fonctions de ***dplyr***, un package de base du ***tidyverse***, qui permet d’effectuer de nombreuses manipulations sur les data frames. Nous avons tout d’abord utilisé la fonction *group\_by*(), qui nous a permis de regrouper toutes les communes ayant la même zone d’emploi. Et nous avons fini par appliquer sur ce regroupement de variables la fonction *summarize*() qui permet de créer une nouvelle data frame où chaque ligne sera composée du numéro de la zone d’emplois et les *géométries* associées.

Après ces opérations réalisées, nous avons obtenu le tracé présent dans la Figure X2.

Une image contenant texte, carte

Description générée automatiquement

Figure X2 : Tracé des zones d’emploi

Une fois les coordonnées des zones d’emplois bien définies, nous avons commencé à regarder comment nous pourrions ajouter les variables socio-économiques ainsi que la position géographique des banques. Ces données furent récupérées par une partie du groupe pendant que l’autre travaillait sur la carte.

Les premières données que nous avons recueilli étaient celles des banques. Nous avons donc commencé par tracer les positions géographiques des banques. Vous pouvez en voir un exemple sur la Figure X3.

Une image contenant carte

Description générée automatiquement

Figure X3 : Position géographique des banques Banque Populaire

Après avoir compris comment afficher la position des banques, nous avons voulu colorer les zones d’emploi en sélectionnant un critère socio-économique.

Pour tracer ces cartes, nous avons dans un premier temps utilisé le package ***ggplot2*** qui permet une certaine aisance pour superposer des objets provenant de classe différente et de jeux de données différents. Cela fût très utile dans notre cas car nous avions d’un côté un jeu de données de classe ***sf*** (*cf. shapefile zones d’emploi*) et de l’autre une simple data frame contenant les données socio-économiques par zone d’emploi.

Ainsi, nous avions enfin une carte permettant de visualiser la position géographique des banques en fonction de critères socio-économiques par zone d’emploi. Vous pouvez en voir un exemple dans la Figure X4.

Une image contenant carte

Description générée automatiquement

Figure X4 : Position géographique des banques Banque Populaire en fonction du taux de pauvreté par zone d’emploi

Le but étant de fournir une application de visualisation de données, nous avons voulu rendre ces cartes interactives : qu’on puisse zoomer, qu’on puisse sélectionner seulement une partie de la carte, qu’on puisse bouger sur la carte, etc… Et pour cela, il faut passer par le package ***plotly***. Grâce à la fonction *ggplotly*(), nous pouvons transformer un objet *ggplot* en objet *plotly*, ce qui nous a permis d’obtenir des cartes interactives et dynamiques. Toutefois, ce type de représentation ne nous fournit pas toutes les informations. En effet, nous avons bien une représentation globale de la France avec la position des banques associées aux critères socio-économiques, mais nous ne savons plus quelle zone d’emploi est laquelle. C’est pour cela que par la suite, nous avons fait le choix d’ajouter sur la carte, le code associé à chaque zone d’emploi. Par ailleurs, ce type de carte ne nous fournit pas le nom des villes.

Afin d’aller plus loin dans la représentation, nous avons voulu essayer de produire des cartes directement via ***plotly***. Pour cela nous sommes passés par l’API *Mapbox*.

*Mapbox* est une entreprise américaine spécialisée dans la cartographie en ligne. Elle a développé pour de nombreux langage une API permettant d’accéder aux services de *Mapbox* dont nombreux d’entre eux utlisent des données d’*OpenStreetMap*.

Pour utiliser l’API sous ***R***, il faut créer un compte *Mapbox* afin d’avoir accès à un *token*. C’est ce dernier qui nous permettra d’avoir accès à tous les services. Nous avons donc regardé comment afficher la position des banques en fonction de critères socio-économiques par zone d’emploi. Afin de comprendre comment fonctionne l’API avec ***plotly***, nous avons divisé le travail en trois. Nous avons tout d’abord regardé comment afficher les banques sur la carte, puis regardé comment colorer les zones d’emploi en fonction de critères socio-économiques, et enfin essayer de combiner les deux.

Vous pouvez voir le résultat sur la Figure X5.

Une image contenant carte

Description générée automatiquement Une image contenant carte

Description générée automatiquement

Figure X5 : Position géographiques de toutes les banques

Nous voyons qu’avec *Mapbox*, les cartes sont plus précisent. En effet, nous avons le nom des villes, les noms des rues, les reliefs et si nous zoomons beaucoup, nous pouvons même voir la forme des bâtiments. De plus, ce type de carte est plus ergonomique et l’utilisation est plus aisée et plus ludique que la version précédente.

Une image contenant carte

Description générée automatiquement

Figure X6 : Représentation du taux de pauvreté par zone d’emploi

Nous avons par la suite tracé les zones d’emploi en fonction de critères socio-économiques. Mais comme vous pouvez le voir sur la Figure X6, la carte n’est pas lisible. Le code couleur n’a aucun sens et la légende n’est pas lisible. ***Plotly*** lit les pourcentages comme des valeurs discrètes et non continues. C’est pour cela que les valeurs sont ainsi affichées dans la légende. Il est sans doute possible de pouvoir jouer sur les paramètres afin d’ajouter une palette de couleurs et de rendre les valeurs continues, mais nous voyons bien qui si nous combinons la carte dans la Figure X5 et celle de la Figure X6, le résultat ne serait pas très lisible. Nous n’avons d’ailleurs pas réussi à combiner les deux cartes.

La principale différence est qu’avec ***ggplot***, nous traçons la carte sur un fond vierge alors qu’avec ***plotly*** et *Mapbox*, nous affichons les points et traçons les points sur une carte de base. Comme nous avons pu le voir, *Mapbox* offre de nombreuses fonctionnalités qui permettent une lecture simple est aisée des cartes. Toutefois, pour le type de représentation que nous voulons, c’est-à-dire afficher la position géographique des banques ainsi que les variables socio-économique, la version ***ggplot*** semble plus adaptée. Toutefois, si nous voulons avoir plus d’informations sur la position des banques, alors la version avec *Mapbox* serait mieux.