

Étude théorique : stations de base du réseau téléphonique français

Progression du stage

Paul MÉHAUD, Brendan SÉVELLEC

České Vysoké Učení Technické v Praze

mai - août 2024

Déroulé de la présentation

Introduction

Données

Semaine du 07/05/24 au 14/05/24

Semaine du 14/05/24 au 21/05/24

Semaine du 21/05/24 au 27/05/24

Semaine du 27/05/24 au 31/05/24

Semaine du 03/06/24 au 06/06/24

From 10/06/24 to 14/06/24

From 17/06/24 to 21/06/24

Introduction

Contexte général

Objectifs

- Déterminer si les stations de base sont en zone urbaine ou rurale ;
- Chercher les stations de bases voisines les unes des autres pour aider à déterminer si les utilisateurs sont en mouvement.

Méthodes

- Approche par la théorie des graphes ;
- Approche par le machine learning.

Données

Données

Arcep

Autorité de régulation des communications électroniques, des postes et de la distribution de la presse.

Jeu de données

Le jeu de données 2023_T4_sites_Metropole.csv^a représente les stations de bases au trimestre 4 de 2023 avec leur position géographique (taille : 16,7 Mo).

a. <https://data.arcep.fr/mobile/sites/>

A retenir :

- 108 838 sites ;
- 29 attributs.

A quoi ressemble notre base ?

code_op	nom_op	num_site	id_site.partage	id_station_anfr	x	y	latitude	longitude	nom_reg
20801	Orange	00000001A1	nan	0802290015	687035	6985761	49,97028	2,81944	Hauts-de-France
20801	Orange	00000001B1	nan	0642290151	422853	6249263	43,28861	-0,41389	Nouvelle-Aquitaine
20801	Orange	00000001B2	nan	0332290026	416932	6422196	44,84112	-0,58333	Nouvelle-Aquitaine
20801	Orange	00000001B3	nan	0472290005	511106	6349234	44,21666	0,63556	Nouvelle-Aquitaine
20801	Orange	00000001C1	nan	0512290147	836824	6889450	49,09028	4,87333	Grand Est
nom_dep	insee_dep	nom_com	insee_com	site_2g	site_3g	site_4g	site_5g	mes_4g.trim	site_ZB
Somme	80	Curlu	80231	1	1	1	0	0	0
Pyrénées-Atlantiques	64	Jurançon	64284	1	1	1	1	0	0
Gironde	33	Bordeaux	33063	1	1	1	1	0	0
Lot-et-Garonne	47	Agen	47001	1	1	1	0	0	0
Marne	51	Sainte-Menehould	51507	1	1	1	0	0	0
site.DCC	site.strategique	site.capa_240mbps	date.ouverturecommerciale_5g	site.5g_700.m.hz	site.5g_.800.m.hz				
0	0	0	nan	0	0			0	0
0	0	1	2020-12-14	0	0			0	0
0	0	1	2021-02-22	0	0			0	0
0	0	1	nan	0	0			0	0
0	0	1	nan	0	0			0	0
		site.5g_1800.m.hz	site.5g_2100.m.hz	site.5g_3500.m.hz					
		0	0	0					
		0	1	0					
		0	0	1					
		0	0	0					
		0	0	0					

Table 1 – Premières valeurs de la base

Description (1/2)

Ce qui nous intéresse

1. *longitude, latitude* : coordonnées de chaque site ;
2. *nom_op* : nom commercial de l'opérateur ;
3. *nom_reg, nom_dep* et *nom_com* : nom de la région, du département et de la commune d'implantation du site ;
4. *site_xg* : équipement du site en technologie xG ($x \in \{2, \dots, 5\}$) ;
5. *num_site* : identifiant du site issu du SI de l'opérateur.

Description (2/2)

Ce qu'il faut retenir

1. Répartition équitable du nombre de sites en fonction de l'opérateur ($\simeq 27\,000$) ;
2. 99,6% des sites équipés en 4G ;
3. 6 stations en moyenne par commune.

La construction de cette base ne nous permet pas de faire de statistiques descriptives intéressantes.

Semaine du 07/05/24 au 14/05/24

Semaine du 07/05/24 au 14/05/24

Brendan

Affichage des données sur une carte

Détail

- Découverte d'une bibliothèque d'affichage de données géographiques interactives : Folium ;
- Affichage des données et colorisation selon plusieurs critères : technologies (2G, 3G, ...) ou opérateurs (Free, SFR, Bouygues Telecom ou Orange).

Problème

Le nombre de données à afficher est très important et rend la visualisation très saccadée.

Solution

Afficher seulement une partie des données à la fois selon différents critères de sélection : par opérateurs, technologie ou région.

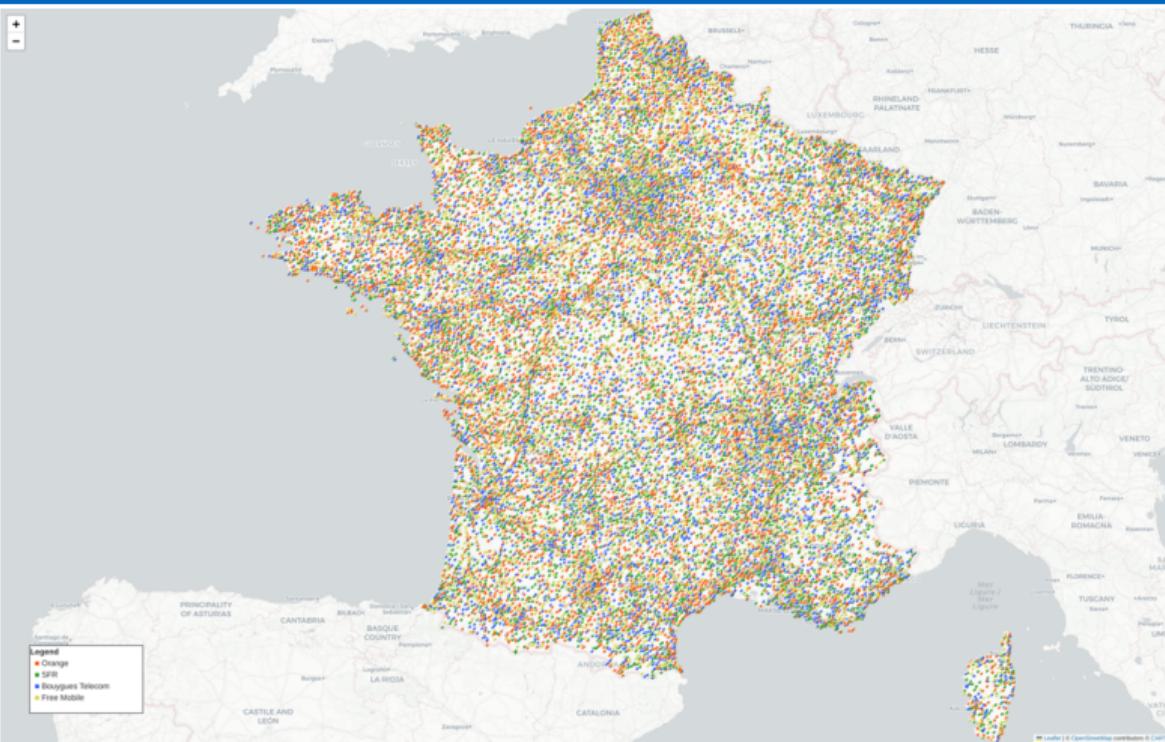


Figure 1 – Les opérateurs dans toute la France

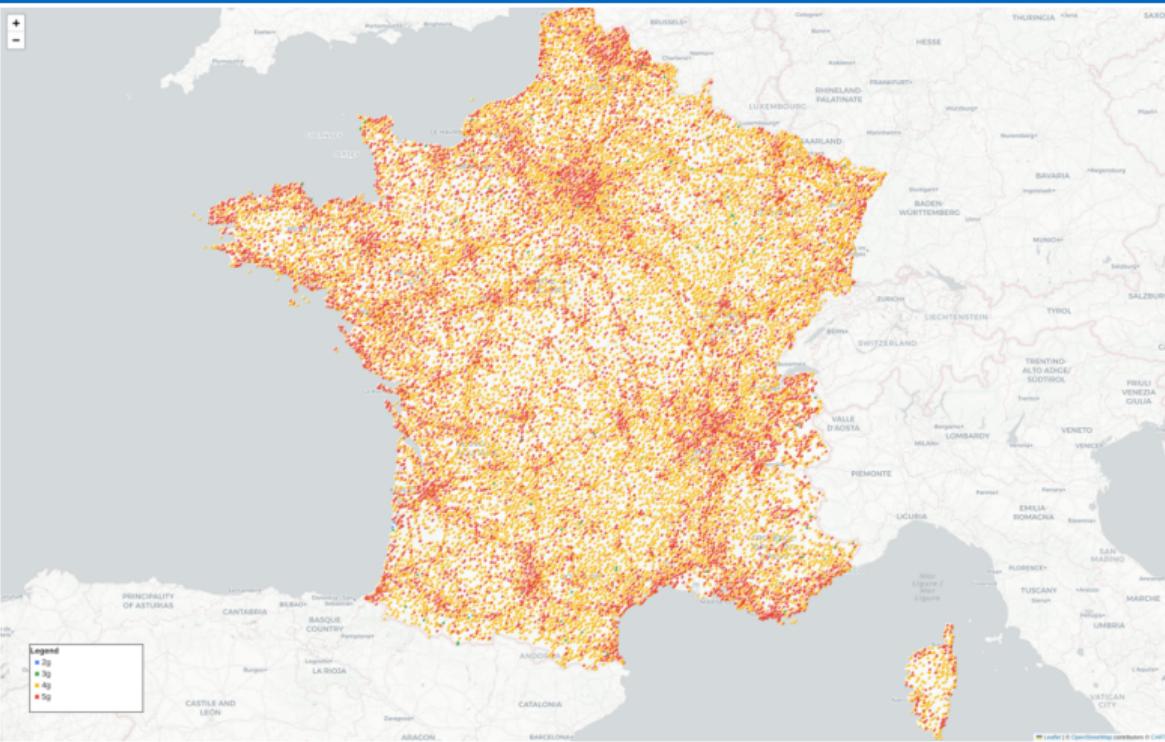


Figure 2 – Les technologies dans toute la France

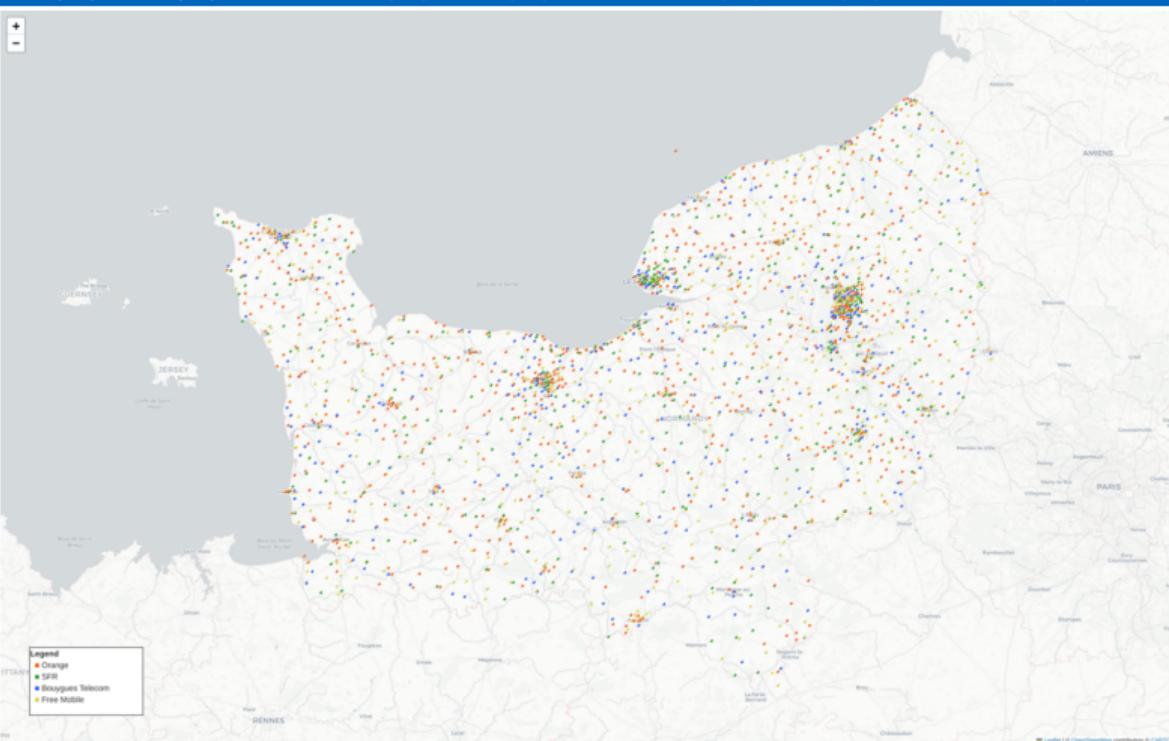


Figure 3 – Les opérateurs en Normandie

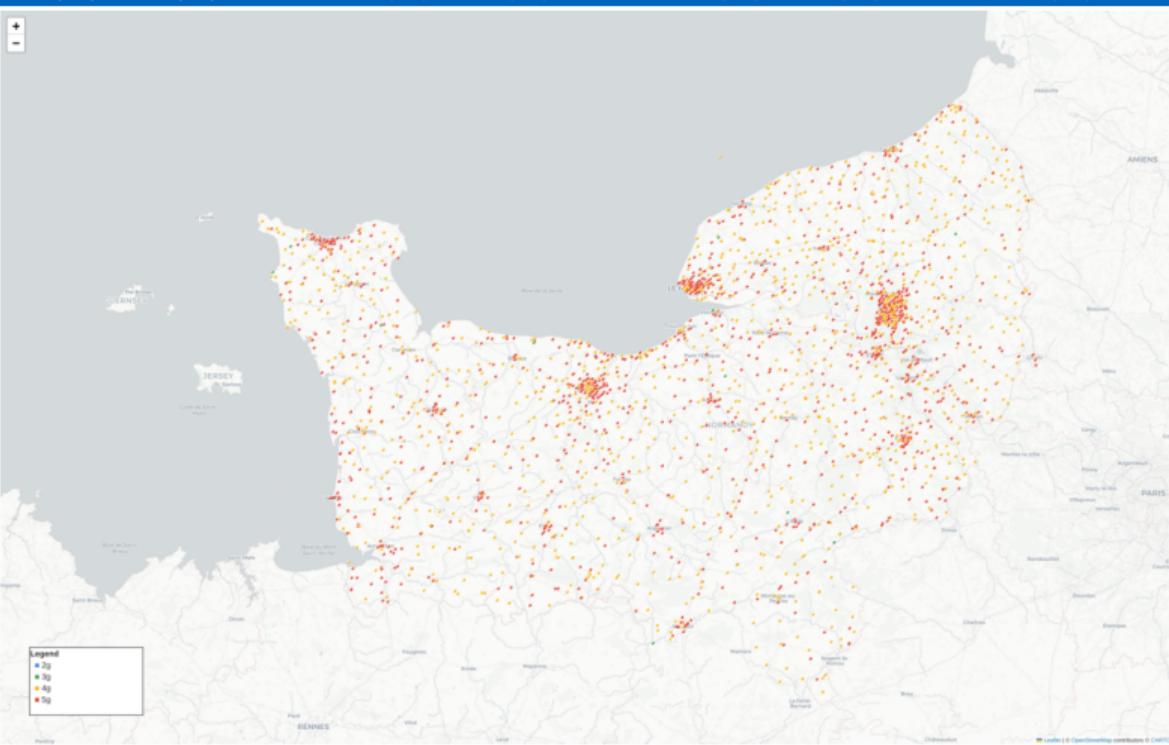


Figure 4 – Les technologies en Normandie

Détection des villes

Il est très clair que les stations de bases sont très regroupé au sein des villes. Il semble donc qu'il soit possible de détecter si les stations de bases sont en zone rurale ou urbaine à l'aide de la densité de stations de base

DBscan (1996)

L'algorithme DBscan (Density-Based spatial clustering of applications with noise) est un algorithme qui s'appuie sur la densité estimée des clusters pour effectuer le partitionnement.

Paramètres

- ε : dissimilarité maximum entre deux individus ;
- n_{\min} : cardinal minimum de chaque classe.

Théorie : l'algorithme

1. Pour chaque point p_j :

$$\begin{aligned} N(p_i) &\leftarrow \{p_j, j \in N = \{1, \dots, n\} \mid d(i, j) \leq \varepsilon\} \\ C &\leftarrow \{p_i \mid |N(p_i)| \geq n_{\min}\} \end{aligned}$$

2. Construire le graphe de voisinage $G = (X, U)$, avec

$$X = \{p_i \mid i \in N\} \text{ et } U = \{ij \mid i, j \in N, d(i, j) \leq \varepsilon\}$$

3. Trouver les composantes connexes des sommets de G (notées G_1, \dots, G_p);

4. Pour chaque $p_i \notin C$:

$$\begin{aligned} j^* &= \arg \min_{1, \dots, p} (d(p_i, C_k)) \\ \text{si } d(p_i, C_{j^*}) &\leq \varepsilon \text{ alors} \\ C_{j^*} &\leftarrow C_{j^*} \cup \{p_i\} \end{aligned}$$

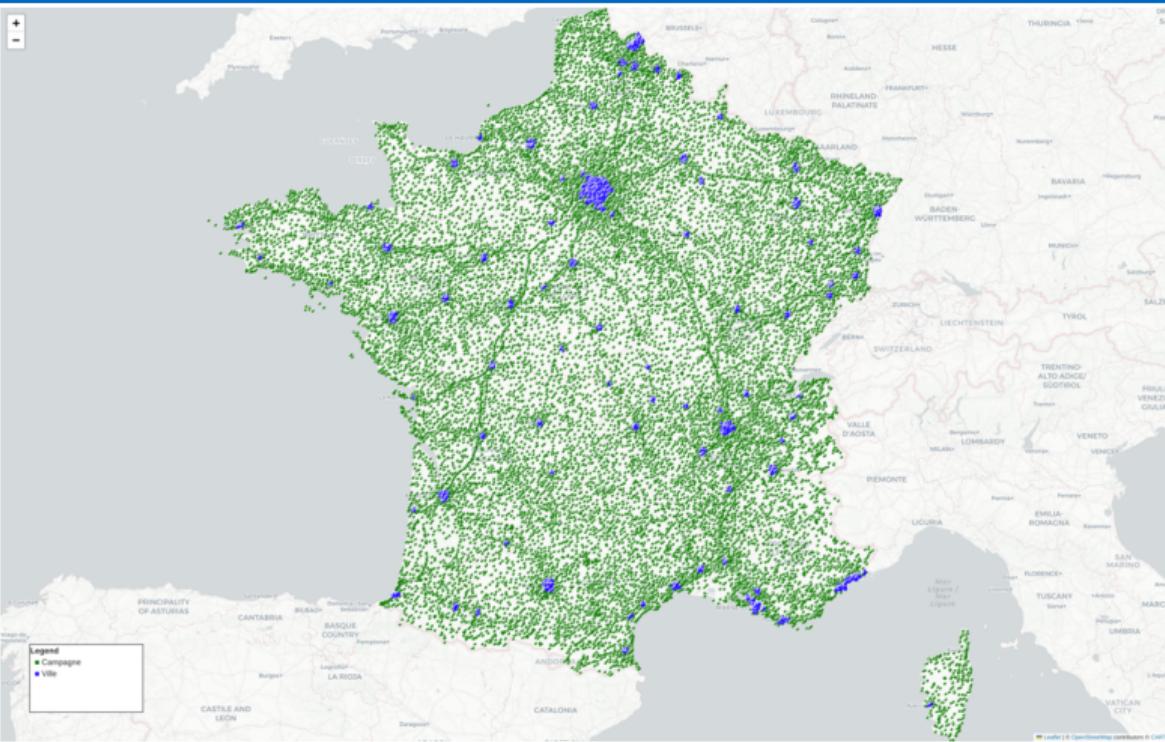


Figure 5 – Les villes détectées en France pour l'opérateur Orange avec $\epsilon = 0.03$ et $n_{min} = 15$

Diagramme de Voronoï

Définition

Le diagramme de Voronoï associé à un ensemble de points $(d_i)_{1 \leq i \leq n}$ est un pavage de l'espace tel que chaque pavé P_i représente l'ensemble des points plus proches de d_i que de n'importe quel d_j avec $j \neq i$ i.e.

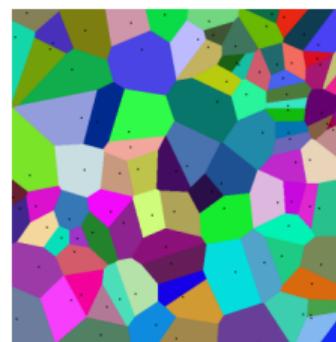


Figure 6 – Exemple de diagramme de Voronoï

Triangulation de Delaunay

Définition

La triangulation de Delaunay d'un ensemble P de points du plan est une triangulation $DT(P)$ telle qu'aucun point de P n'est à l'intérieur du cercle circonscrit d'un des triangles de $DT(P)$. Les triangulations de Delaunay maximisent le plus petit angle de l'ensemble des angles des triangles, évitant ainsi les triangles « allongés ».



Figure 7 – Exemple de triangulation de Delaunay

Lien entre les 2

Propriétés

- La triangulation de Delaunay d'un ensemble discret P de points est le graphe dual du diagramme de Voronoï associé à P ;
- Il est donc très facile de passer de l'un à l'autre (en temps polynomial) ;
- il existe des algorithmes pour trouver une triangulation de Delaunay en $O(n \log(n))$.

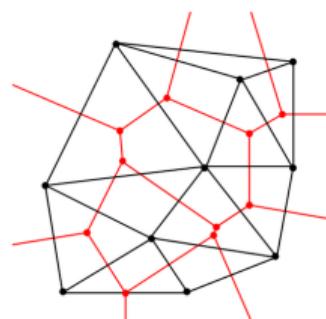


Figure 8 – Lien entre Delaunay et Voronoï

Retour au problème

Application à notre cas d'étude

Nous allons pouvoir utiliser ces notions en définissant :

- les voisins potentiels comme étant les triangles adjacents dans la triangulation de Delaunay
- les zones de couverture des antennes comme les pavés associé dans le diagramme de Voronoï

Semaine du 07/05/24 au 14/05/24

Paul

Résumé des épisodes précédents

Mon travail, pour la semaine qui vient de s'écouler, s'articule autour de trois axes majeurs :

- Découverte des données ;
- Documentation ;
- Reprise du travail de l'année précédente.

Documentation

La majeure partie du travail de la semaine écoulée a consisté à se former aux différents outils de Python afin de pouvoir effectuer sereinement le travail.

Outils utilisés

- `pandas.DataFrame` : gestion des données ;
- `scipy.spatial.Delaunay` : création de la triangulation de Delaunay ;
- `networkx.Graph` : création de graphes ;
- `matplotlib.pyplot` : affichage des résultats.

Reprise du travail précédent (1/3)

La première tâche consistait à essayer de faire fonctionner le code fourni. Résultat : il ne fonctionne pas...
Décision : refaire par moi-même. Cependant, j'ai gardé les idées.

Approche pour déterminer les voisins de stations de base

- Faire une triangulation de Delaunay (liste de voisins potentiels) ;
- Eliminer les voisins distants de plus de 15 km ;
- Garder le voisin le plus proche dans chaque cadrant autour de chaque station (6 cadrants) ;
- Garder le voisin le plus proche quand deux stations voisines sont séparées par un angle faible.

Reprise du travail précédent (2/3)

Ayant refait l'implantation moi-même, j'ai décidé d'utiliser les graphes au lieu de simplement les dataFrames : permettra de faciliter l'application de la théorie des graphes.

Apports de cette nouvelle représentation

- On travail directement sur le graphe de Delaunay ;
- Le traitement des voisins est beaucoup plus facile ;
- La représentation graphique est plus claire.

Reprise du travail précédent (3/3) : Résultats

Pour l'instant seul les deux premiers critères de filtrage sont fonctionnels. Voici ce que l'on obtient sur le département du Gard :

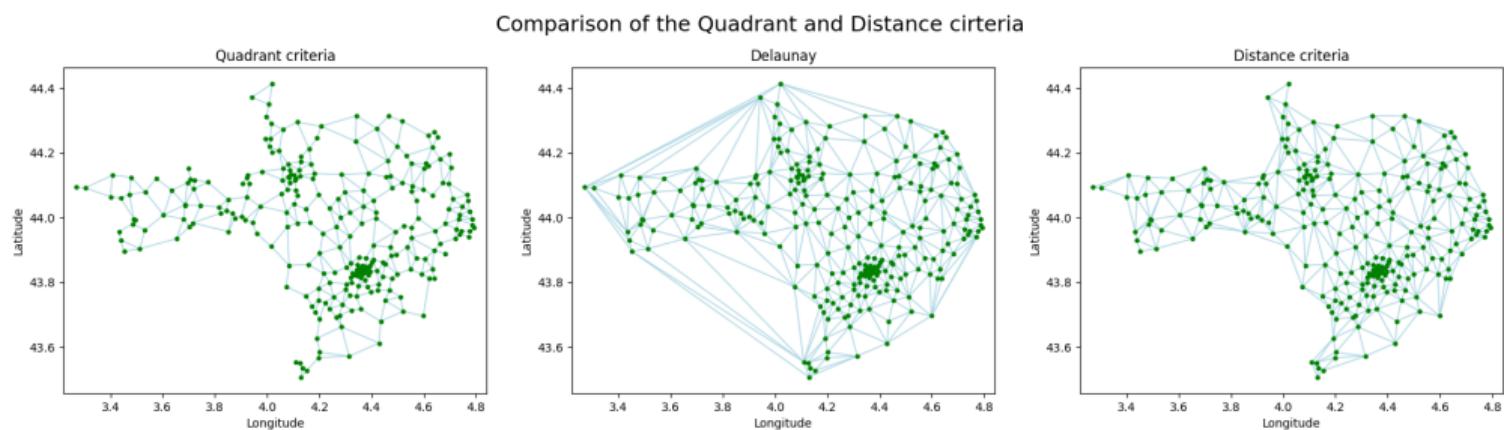


Figure 9 – Evolution de la triangulation de Delaunay en fonction de critères de filtrage

Perspectives d'amélioration et travail à venir

Améliorations

- Vérifier que l'algorithme du critère du quadrant donne les bons résultats ;
- Optimisation dudit algorithme.

Travail à venir

- Implanter une version fonctionnelle du critère de l'angle ;
- Se renseigner sur l'état de l'art de la théorie des graphes.

Semaine du 07/05/24 au 14/05/24

Brendan

Réalisation d'une interface graphique en Python

Contexte

- Beaucoup de cartes à tracer car beaucoup de paramètres ;
- Cartes gourmandes en ressources et pas adaptées à des notebooks Python ;
- Réalisation d'une application permettant de facilement tracer ces cartes au sein d'un navigateur web.

Semaine du 14/05/24 au 21/05/24

Semaine du 14/05/24 au 21/05/24

Le jeu de donnée

Le jeu de donnée

Arcep

L'autorité de régulation des communications électroniques, des postes et de la distribution de la presse (Arcep) est une autorité administrative indépendante française chargée de réguler les communications électroniques et postales et la distribution de la presse.

Mon Réseau Mobile

Mon Réseau Mobile est la plate-forme cartographique regroupant l'ensemble des données géographiques en lien avec les réseaux dits « mobiles » (2G, 3G, 4G, 5G) régulés par l'Arcep.

Nouvelle mise à jour

Une nouvelle mise à jour des données est prévue de **jeudi 20 juin 2024 à 17h40** : données du 1^{er} trimestre de 2024.

A cette adresse : <https://www.data.gouv.fr/fr/datasets/mon-reseau-mobile/#/discussions>, on peut poser nos questions sur le jeu de données.

Arborescence du jeu de données

Les fichiers de données sont rangés par trimestre de publication, zone (France métropolitaine/Outre-mer) et département le cas échéant :



Figure 10 – Architecture de la base de donnée

Fréquences de mises à jour

Fréquence de publication des données :

Fichiers	Fréquence de publication
Cartes de couverture théorique 2G, 2G3G, 3G	Semestrielle (T2, T4)
Cartes de couverture théorique 4G	Trimestrielle (T1, T2, T3, T4)
Sites fournissant un service mobile	Trimestrielle (T1, T2, T3, T4)
Mesures de qualité de service Arcep	Annuelle
Dispositif de couverture ciblée	Trimestrielle (T1, T2, T3, T4)
Mesure de crowdsourcing	Variable

Figure 11 – Fréquence de publication de mises à jour

Semaine du 14/05/24 au 21/05/24

Faisons parler les données

Quelques chiffres en vrac

Tout d'abord, remarquons que chaque station peut être identifiée à son `num_site` (seul deux stations n'en n'ont pas). Ensuite, voici ce que l'on a découvert :

Des chiffres sympathiques

- `site_zb` : 10 596 (Site issu du programme « zones blanches – centres-bourgs ») ;
- `site_dcc` : 10 627 (Site issu du « Dispositif de Couverture Ciblée ») ;
- `site_strategique` : 144 (Site issu du programme « France Mobile ») ;
- `mes_4g_trim` : 1 618 (Equipement du site en technologie 4g au cours du dernier trimestre (du 30/06/2022 au 30/09/2022)) ;
- `id_site_partage` : 5 453 (Sites mutualisés entre plusieurs opérateurs) ;
- `site_capa_204mbps` : 92 664 (Site dont la capacité maximum théorique est supérieure ou égale à 240 Mbs).

Quelques chiffres en vrac : précision sur les indicateurs

Ensuite, voici ce que l'on a découvert :

site_zb

Le premier programme, initié en 2003 et nommé « zones blanches – centres-bourgs » consistait à apporter des services de téléphonie mobile, SMS et internet mobile à très haut débit, dans plus de 3500 centres-bourgs de communes de France qui ne bénéficiaient d'aucune couverture mobile.¹

site_dcc

Le dispositif de couverture ciblée vise à assurer une couverture mobile de qualité dans des zones non ou mal couvertes, en construisant jusqu'à 5 000 nouveaux sites par opérateur, dont une partie sera mutualisée.²

1. <https://www.tactis.fr/zone-blanche-zone-grise/>

2. <https://agence-cohesion-territoires.gouv.fr/france-mobile-54>

Comparaison des différents équipements en terme de technologies (1/7)

Voici tout d'abord un graphique sur la présence d'une technologie en fonction de l'opérateur (une technologie présente sur un site n'exclue pas la présence d'une autre technologie) :

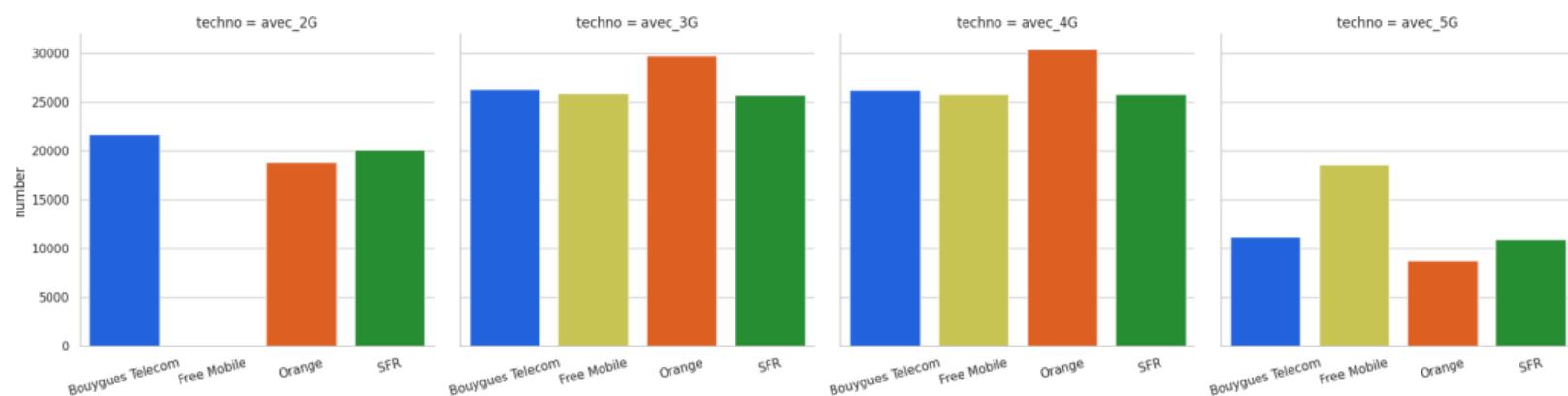


Figure 12 – Nombres de sites équipés d'au moins une technologie

Comparaison des différents équipements en terme de technologies (2/7)

Technologie	Bouygues Telecom	Free Mobile	Orange	SFR	Total
2G	4	0	10	20	34
3G	30	102	65	37	234
4G	15	15	602	106	738
5G	0	0	3	0	3
2-3G	66	0	21	80	167
2-4G	12	0	63	67	142
2-5G	0	0	0	0	0
3-4G	3889	7225	9288	4909	25311
3-5G	0	0	0	0	0
4-5G	6	0	109	38	153
2-3-4G	11044	0	11697	9831	32572
2-3-5G	0	0	1	0	1
2-4-5G	0	0	5	10	15
3-4-5G	681	18607	1638	835	21761
2-3-4-5G	10584	0	7038	10085	27707
Total	26331	25949	30540	26018	108838

Table 2 – Résumé des données de présence de technologie

Comparaison des différents équipements en terme de technologies (3/7)

Maintenant nous nous intéressons à la fréquence de présence de certaines technologies et pas d'autres :

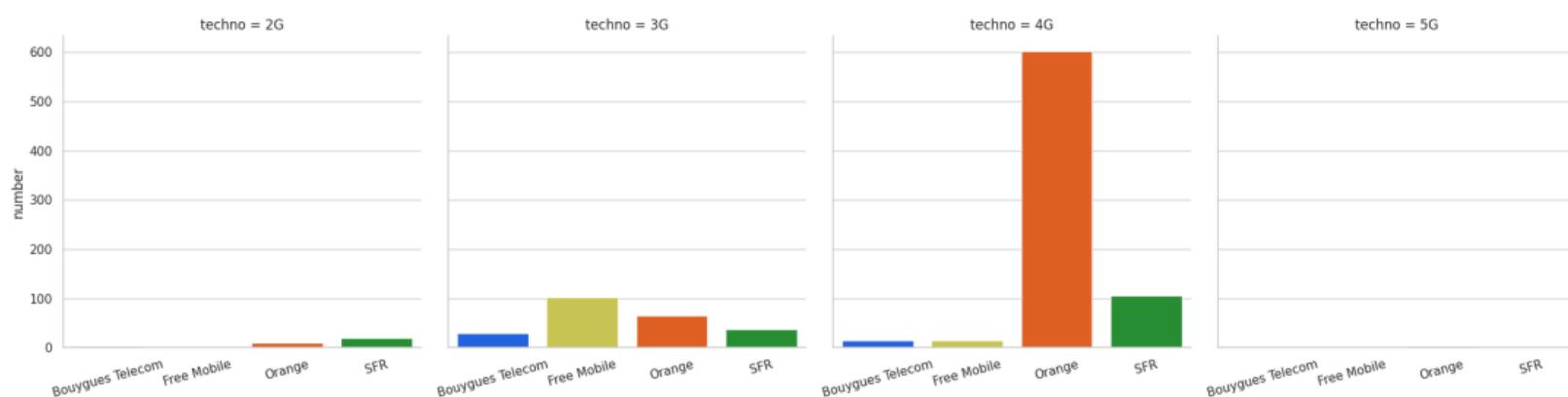


Figure 13 – Nombres de sites équipés d'une unique technologie

Comparaison des différents équipements en terme de technologies (4/7)

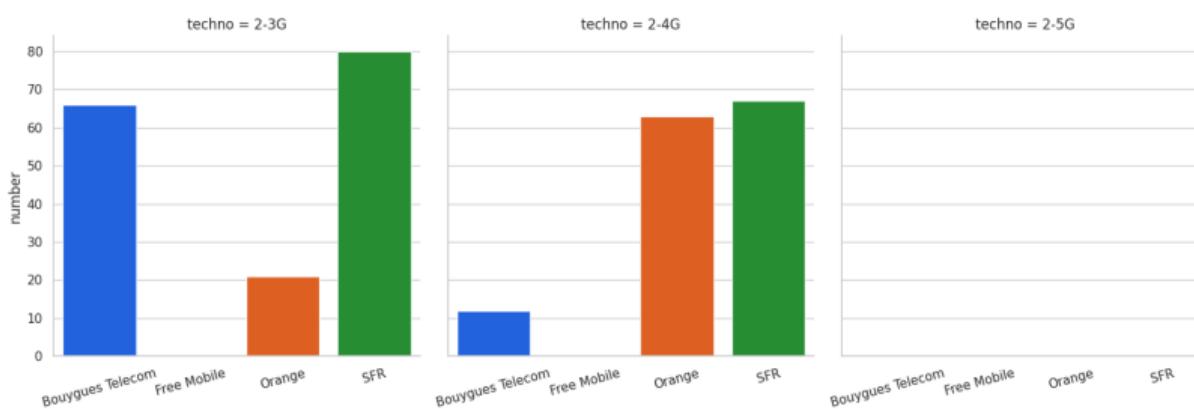


Figure 14 – Nombres de sites équipés de deux technologies

Comparaison des différents équipements en terme de technologies (5/7)

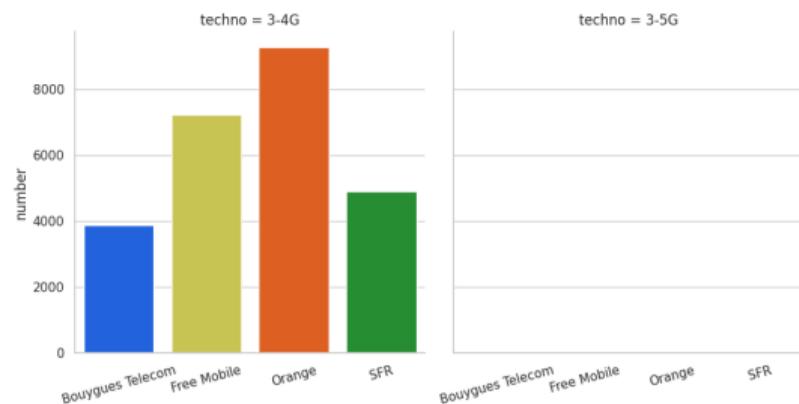


Figure 15 – Nombres de sites équipés de deux technologies (suite)

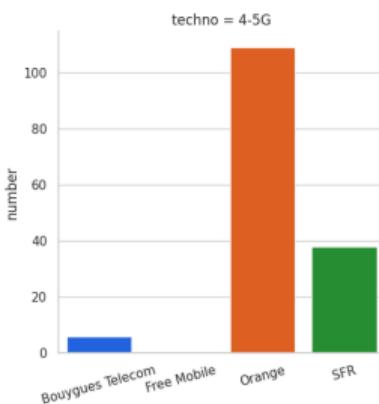


Figure 16 – Nombres de sites équipés de deux technologies (suite-bis)

Comparaison des différents équipements en terme de technologies (6/7)

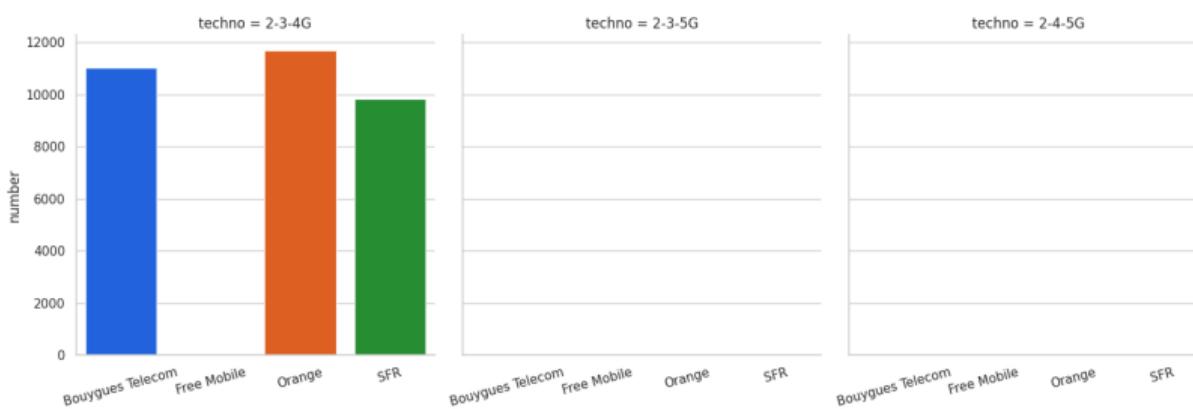


Figure 17 – Nombres de sites équipés de trois technologies

Comparaison des différents équipements en terme de technologies (7/7)

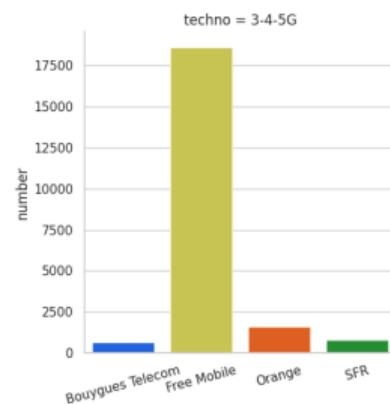


Figure 18 – Nombres de sites équipés de trois technologies (suite)

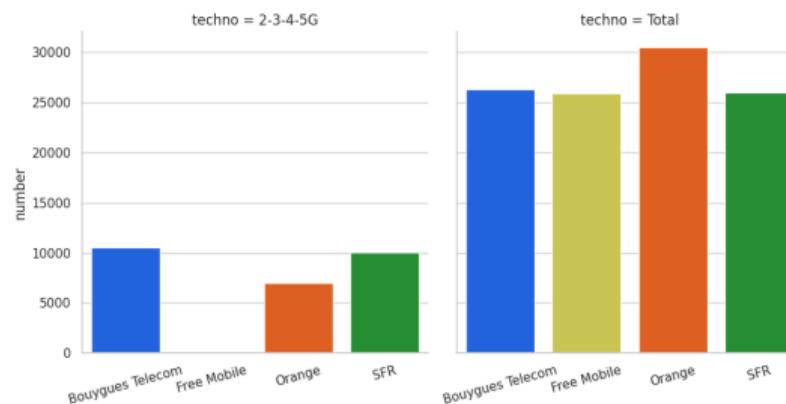


Figure 19 – Nombre de sites équipés de toutes les technologies et total

Semaine du 14/05/24 au 21/05/24

Affichage plus détaillé des cartes

Les stations de base par opérateur

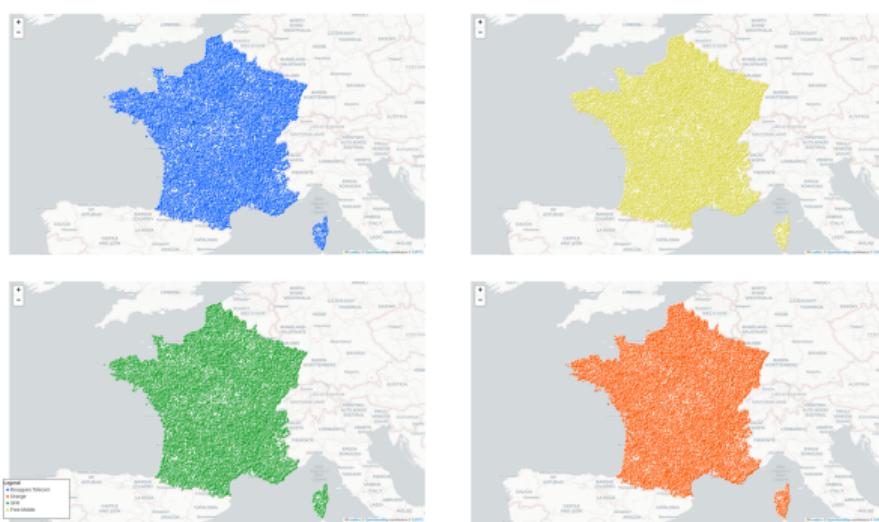


Figure 20 – Les stations de base par opérateurs

Les stations 2G

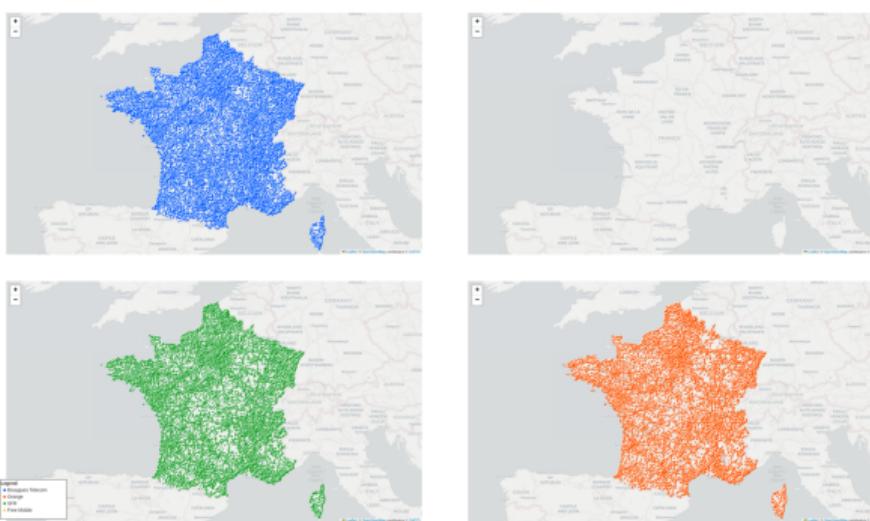


Figure 21 – Les stations 2G

Les stations 3G

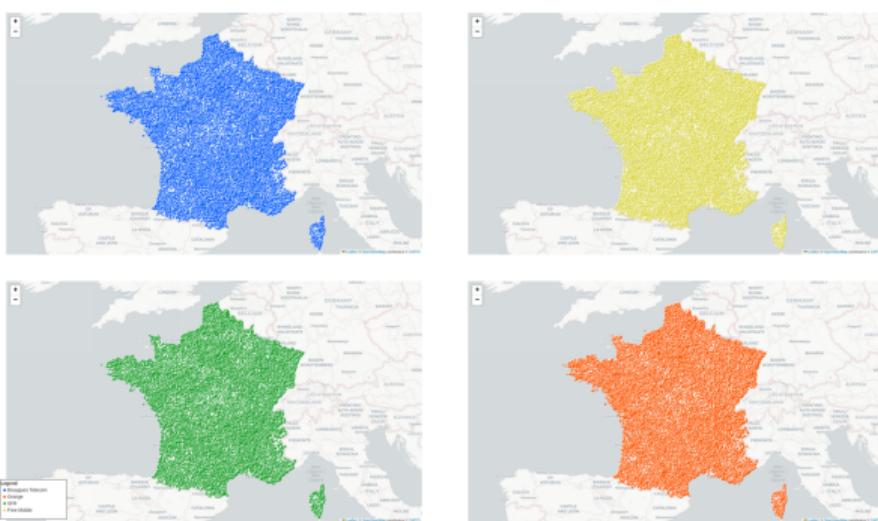


Figure 22 – Les stations 3G

Les stations 4G

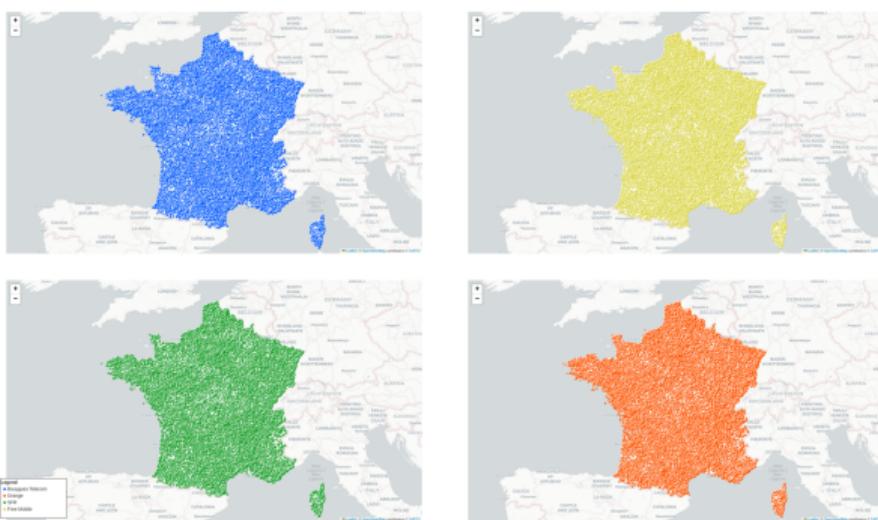


Figure 23 – Les stations 4G

Les stations 5G

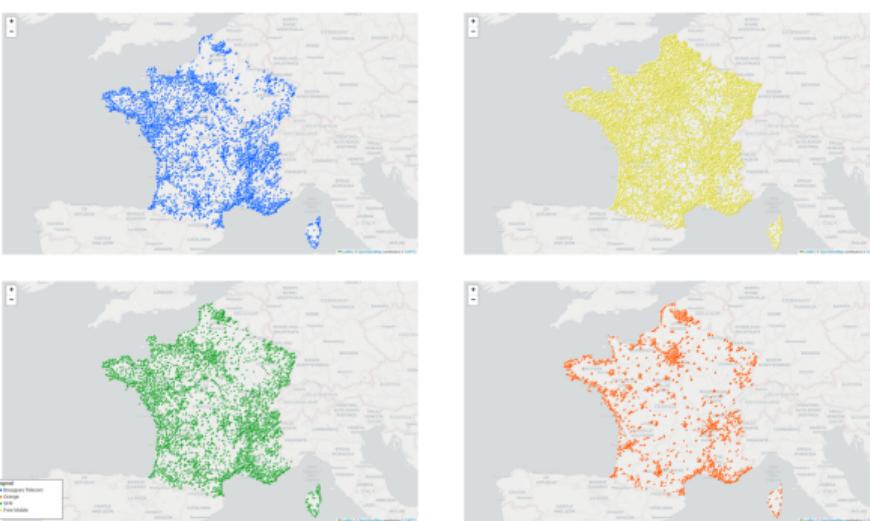


Figure 24 – Les stations 5G

Semaine du 14/05/24 au 21/05/24

Evolution du nombre de stations de base au cours du temps

2023_T4_sites_5G_historique_comptage

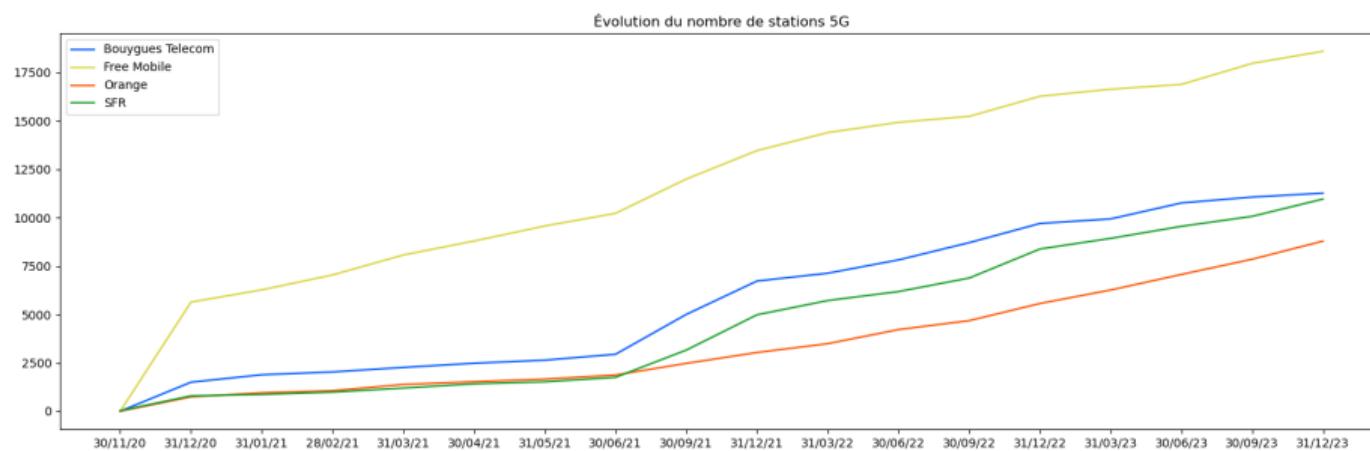


Figure 25 – Evolution du nombre de stations 5G en France

Compilation des jeux de données sites

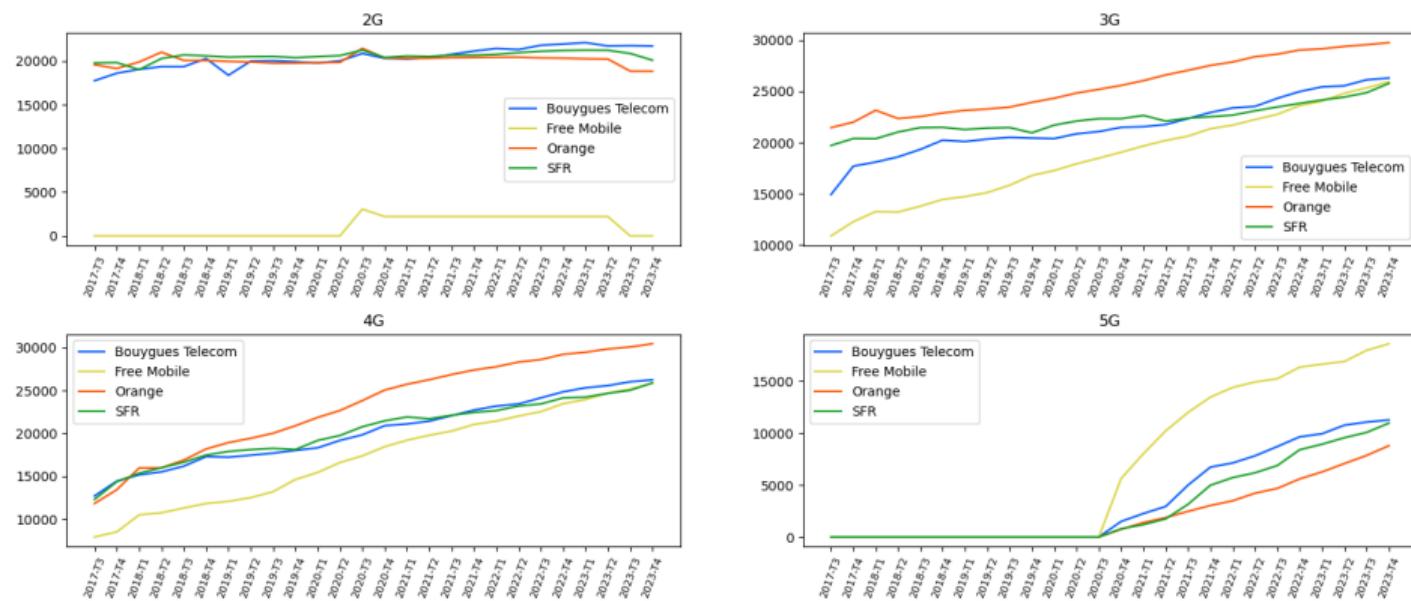


Figure 26 – Evolution du nombre de stations toutes technologies en France

Dispositif de couverture ciblée, éléments d'analyse

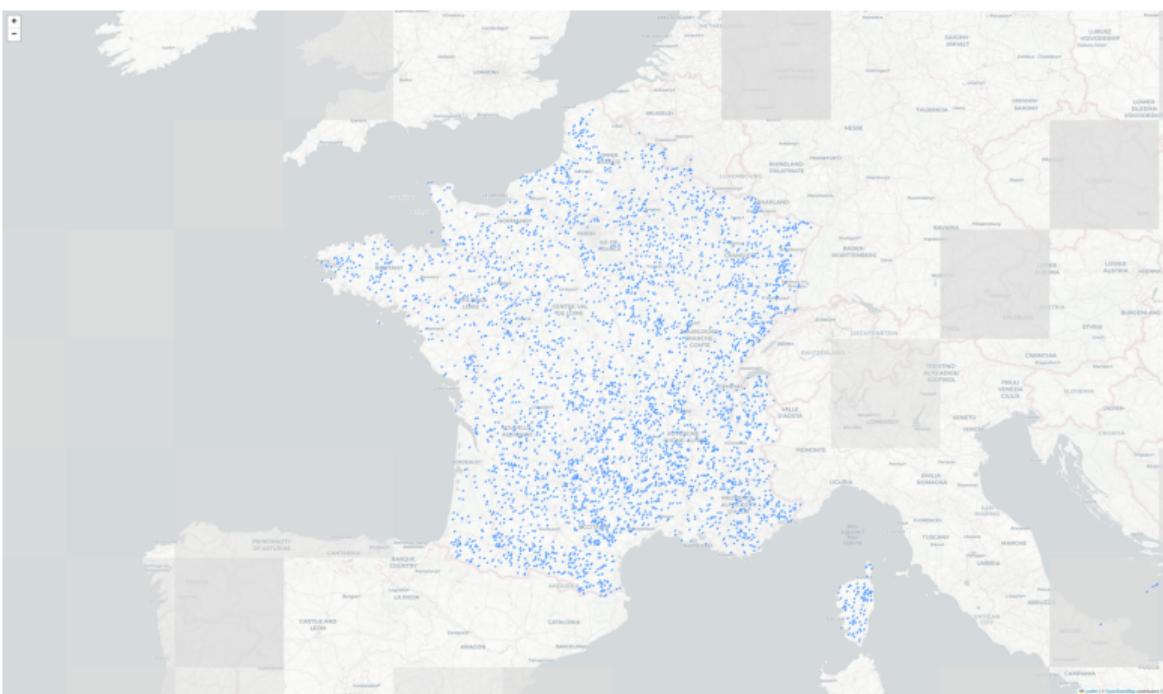
infos générales

- 4621 stations de bases potentielles ;
- 26 attributs.

anomalies

- 412 lignes de ce jeu de données ont les colonnes `x_lambert_93` ou `y_lambert_93` non renseigné ;
- Plusieurs points sont placés en dehors de la France.

Dispositif de couverture ciblée



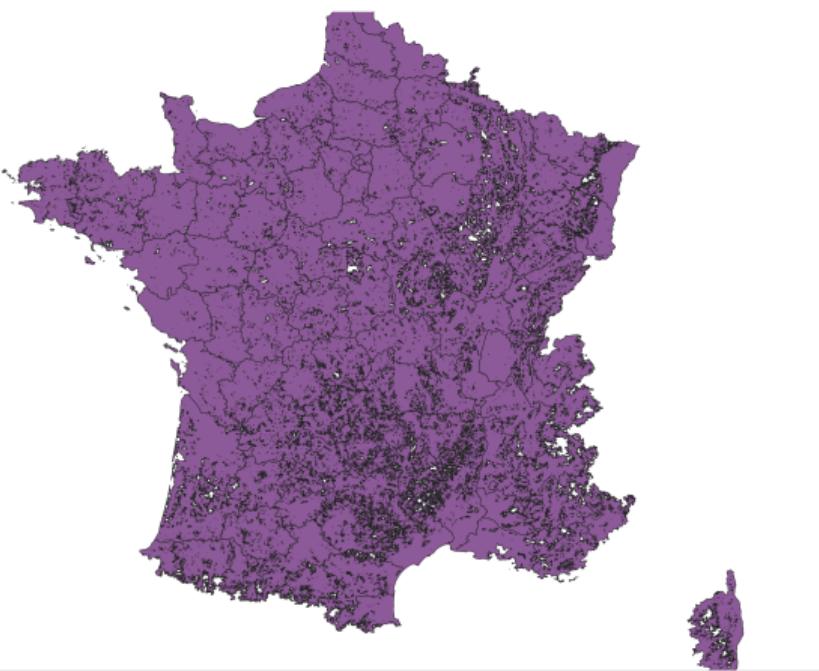
Couverture théorique

Informations

- format .gpkg (geopackage) ;
- ouverture à l'aide du logiciel libre QGIS.

Utilité

- Aurait pu permettre de déterminer si 2 stations de base sont voisines (si leur couverture se chevauche) ;
- Non réalisable à l'aide de ce jeu de donnée : seule la couverture globale de chaque région est donnée.



Semaine du 21/05/24 au 27/05/24

Semaine du 21/05/24 au 27/05/24

Statistiques stations de bases - départements

Méthodologie

On va utiliser une base de donnée complémentaire, reprise de celle trouvée l'année dernière.

Une nouvelle base de donnée¹

Cette base de donnée renseigne, par département, la **superficie** (en km²), la **population** et la **densité de population** au km².

A partir de cette base, on va donc pouvoir extraire le nombre d'habitants par stations (normalisé par la taille du département) et la densité de station par département.

Calcul du nombre d'habitants par stations normalisé

Soit λ le nombre d'habitants par stations, par km². Tout d'abord, on se donne γ , le rapport entre le nombre d'habitant et la surface du département, qui est donné par la densité de population. Ensuite, on calcule notre résultat comme suit :

$$\lambda = \gamma \times \frac{1}{\text{nombre de stations du département}}$$

1. <https://france.ousuisje.com/departements/classement/superficie.php>

Nombre d'habitants par stations non normalisé

C'est le même calcul que précédemment, sauf qu'à la place de γ , on utilise simplement la population du département.

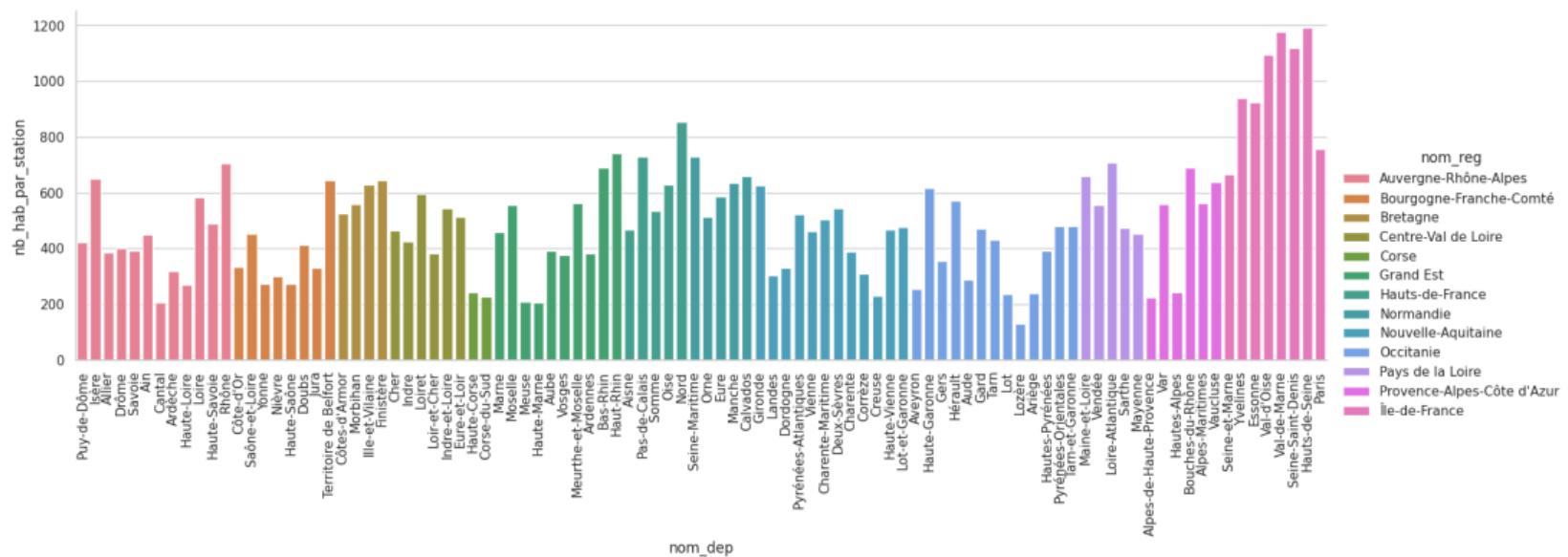


Figure 27 – Répartition du nombre d'habitants par station, en fonction du département

Nombre d'habitants par stations normalisé (1/2)

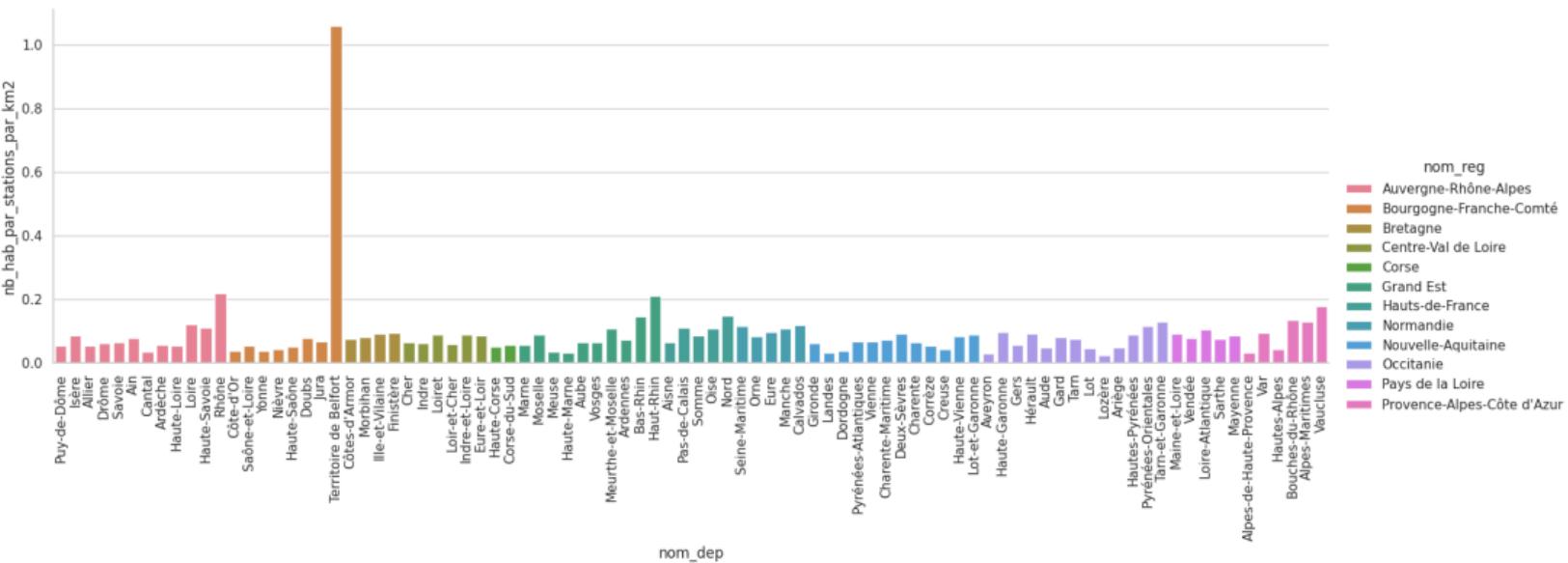


Figure 28 – Répartition du nombre d'habitants par station, en fonction du département (normalisé), sans l'Île-de-France

Nombre d'habitants par stations normalisé (2/2)

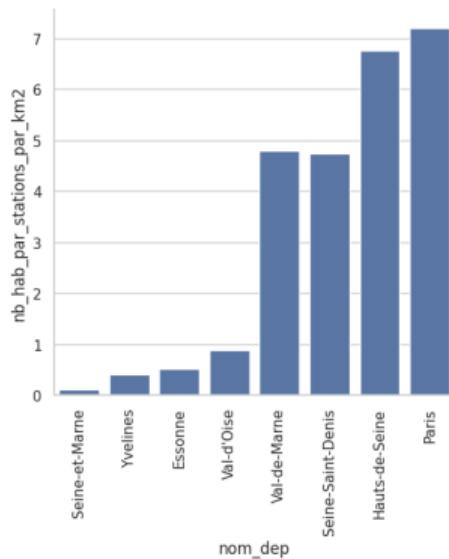


Figure 29 – Répartition du nombre d'habitants par station, en fonction du département (normalisé), sur l'Île-de-France

Densité de stations (1/2)

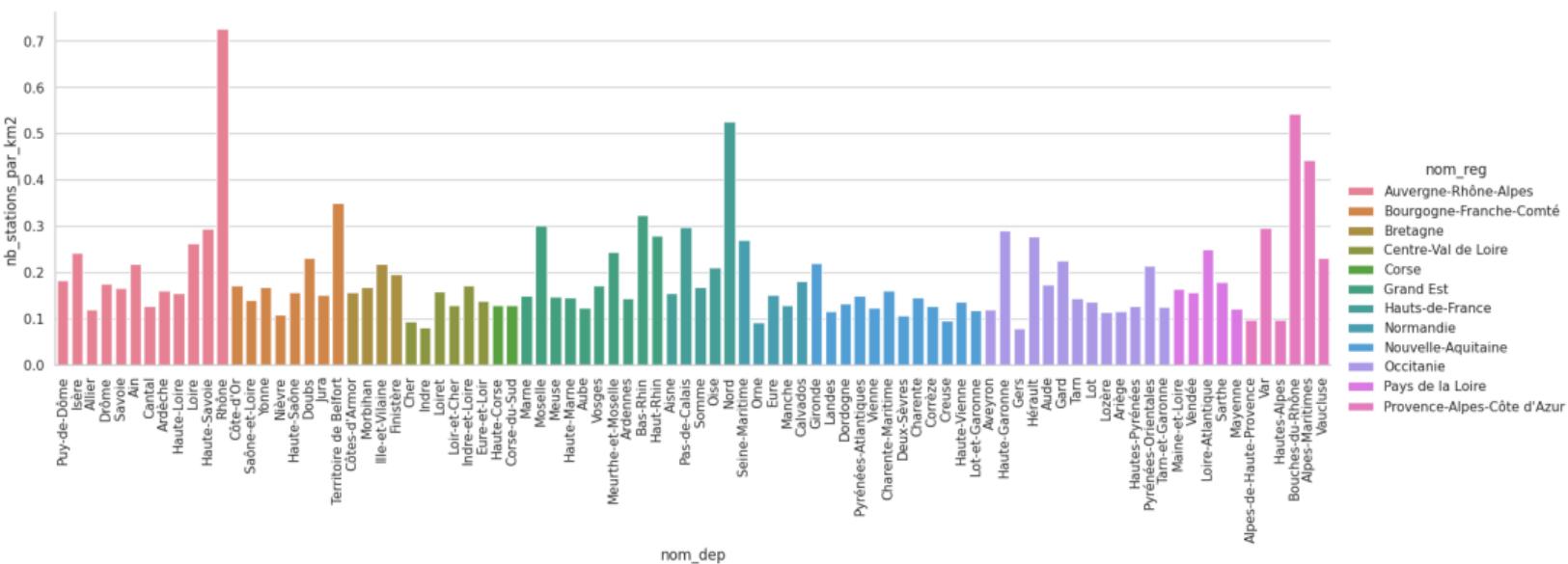


Figure 30 – Nombre de stations de base au km², sans l'Île-de-France

Densité de stations (2/2)

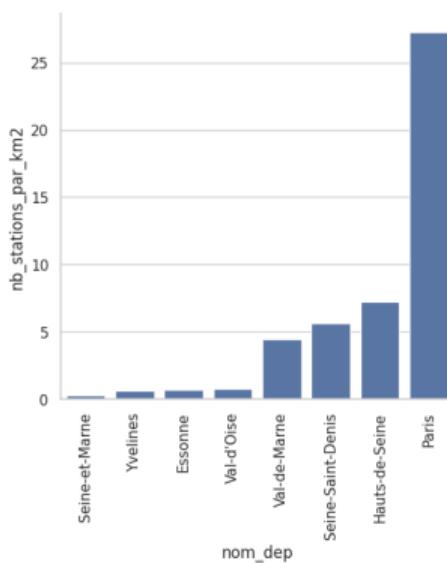


Figure 31 – Nombre de stations de base au km^2 , sur l'Île-de-France

Fréquences d'émission des stations 5G

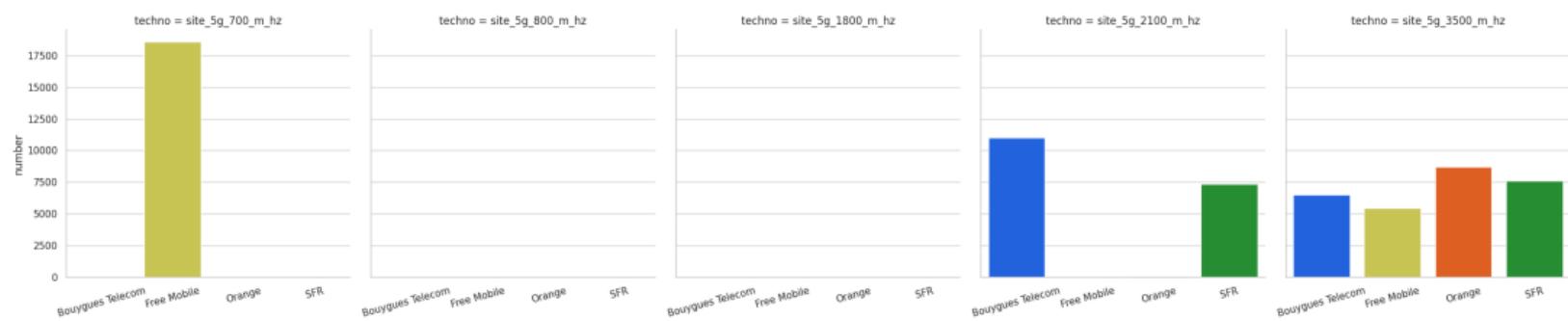


Figure 32 – Répartition des fréquences d'émission des stations 5G, en fonction des opérateurs

Semaine du 21/05/24 au 27/05/24

Détection de voisins - méthode PAQUIRY

Principe général

La recherche de voisins va s'articuler autour de deux axes :

Triangulation de Delaunay

On crée un graphe de voisinage grâce à la triangulation de Delaunay. Ceci nous donne donc, pour chaque station de base, une liste de voisins potentiels. Il faut ensuite vérifier que ce sont bien des voisins réels.

Critères de sélection

Pour être sûr qu'un voisin est un voisin réel on applique trois critères, dans l'ordre suivant :

1. la distance maximale ;
2. le plus proche voisin dans un quadrant ;
3. l'angle minimum entre deux voisins.

Cette méthode a été élaborée par Delphine PAQUIRY l'été dernier.

Critère de distance

Principe

Ici, on élimine tous les voisins qui sont distants de plus de 15 km.

Intérêt du critère

Il est logique de penser que deux stations trop éloignées géographiquement ne sont pas voisines.

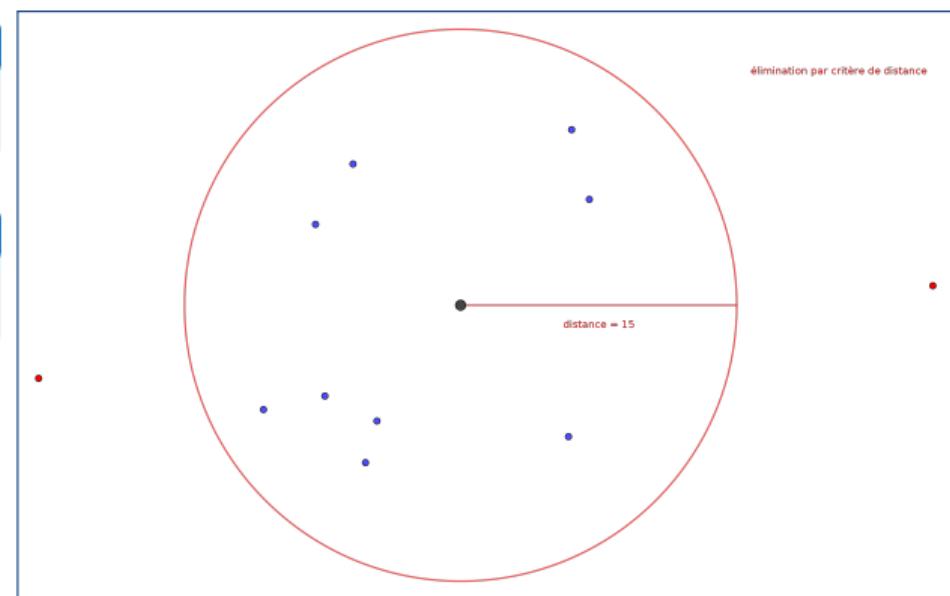


Figure 33 – Illustration du critère de distance

Critère des cadrants

Principe

Ici, on ne garde que le plus proche voisin dans un quadrant. On a choisi 6 cadrants car c'est le chiffre qui nous donne les meilleurs résultats.

Intérêt du critère

On évite les effets de cluster.

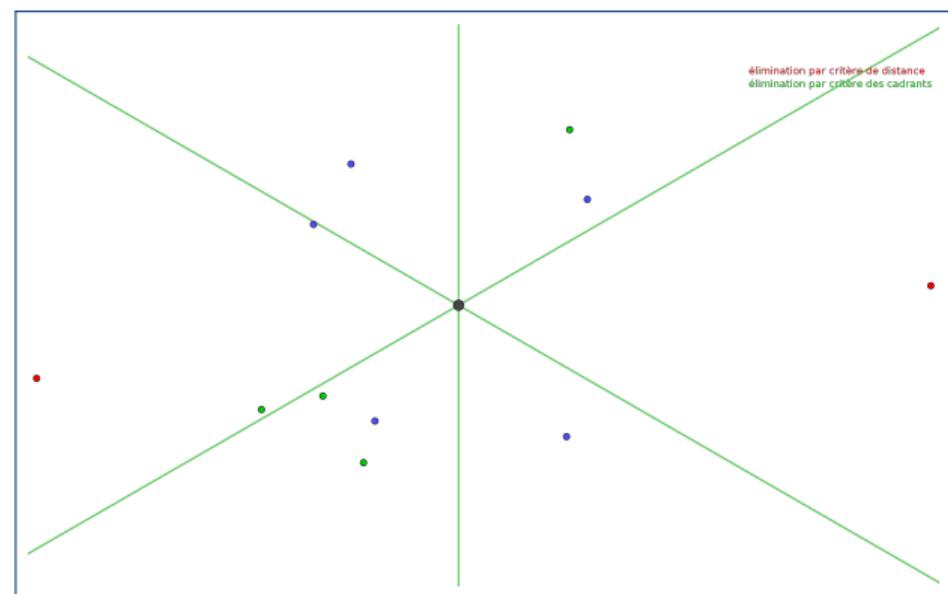


Figure 34 – Illustration du critère des cadrants

Critère de l'angle

Principe

Ici, si deux voisins sont séparés d'un angle inférieur à 30° , on ne garde que le plus proche. Cette valeur d'angle est arbitraire et pourrait être variable en fonction de l'urbanité de la station.

Intérêt du critère

Si deux stations sont trop proches angulairement parlant, la plus proche fait écran par rapport à la plus éloignée.

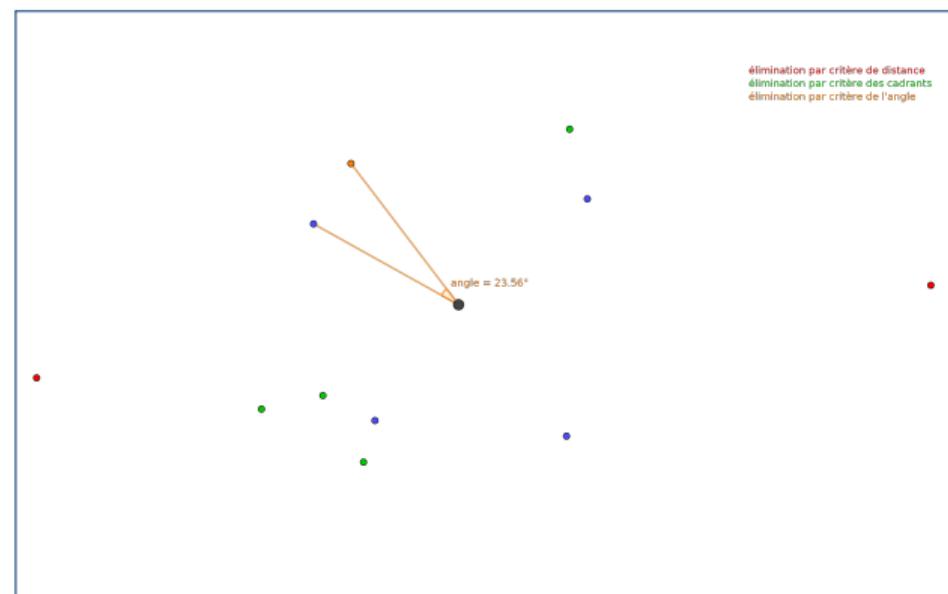


Figure 35 – Illustration du critère de l'angle

Voisins réels

Voici donc ce que l'on obtient à la fin, un graphe de voisinage :

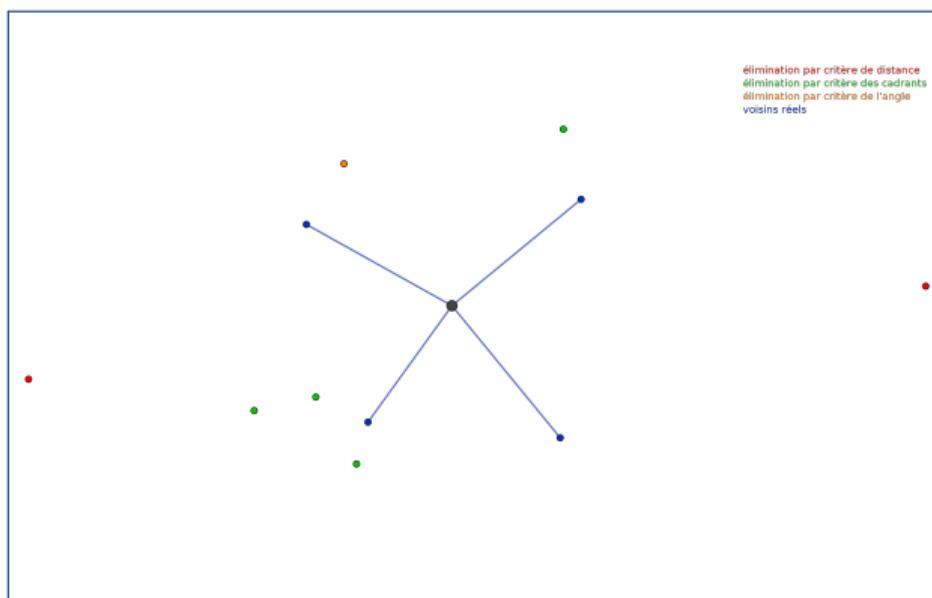


Figure 36 – Résultat final - voisins réels

Semaine du 21/05/24 au 27/05/24

Détection automatique des villes

DBScan

Principe

- Méthode de référence de la classification à densité (déttection de clusters en se basant sur la concentration de points) ;
- Se base sur deux paramètres : ε et n_{min} qui caractérisent respectivement la distance maximale pour que deux points soient considérés « proches » et la quantité minimale de points proches pour qu'un cluster soit créé.

Inconvénients

- Le choix du paramètre ε est hasardeux ;
- Ne se comporte pas bien avec des jeux de données avec des clusters de densité variable (certains clusters avec des points plus rapprochés que d'autres) ;
- Réagit mal au bruit ;
- Est binaire : un point appartient ou n'appartient pas à un cluster, pas d'entre deux.

Fonctionnement DBScan (1/2)

Etape 0 : Initialisation

Soient n points de \mathbb{R}^p , $\epsilon \in \mathbb{R}$ et $n_{min} \in \mathbb{N}$.
Ici, on prend $n_{min} = 3$ et $\epsilon = 2$

Etape 1 - Les voisins

Pour chaque point, on cherche tous les points à une distance inférieure à ϵ (dont lui même).
On les appellera les voisins.

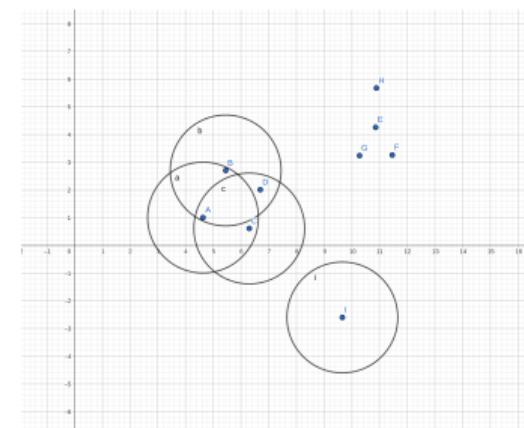


Figure 37 – DBScan : Détection des voisins

Point	A	B	C	...	I
N(Points)	{A, B, C}	{A, B, D}	{A, C, D}		{I}

Fonctionnement DBScan (2/2)

Etape 2 - Construction du graphe de voisinage des noyaux

- Les points ayant au moins n_{min} voisins sont considérés comme des noyaux. On construit alors un graphe dont les sommets sont les noyaux et il existe une arête entre deux noyaux si et seulement si ils sont voisins.
- Les composantes connexes de ce graphe sont les clusters. On rattache alors aux clusters les points voisins d'au moins un noyau de ce cluster, on les appelle points de bordure.
- Enfin, les sommets qui n'apparaissent pas dans le graphe sont considérés comme du bruit (pas de classe).

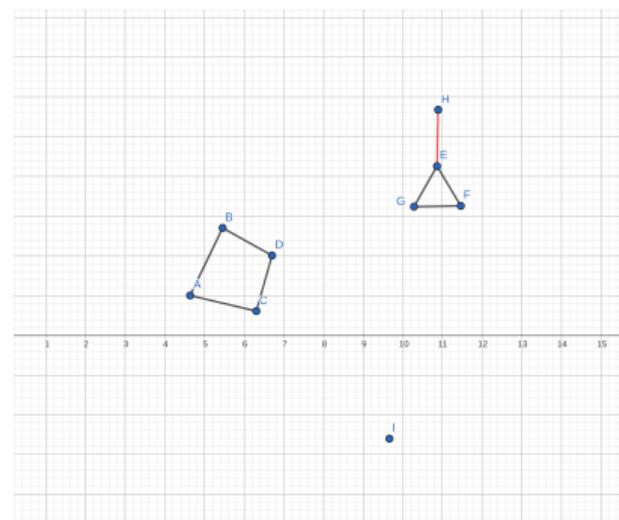


Figure 38 – DBScan : création des clusters

Changement de métrique

Une façon d'améliorer le problème de mauvaise prise en compte du bruit est de changer de métrique :

Core distance

La *core distance* d'un point du jeu de donnée est la distance entre ce point et son k -ième plus proche point (k à fixer).

Nouvelle métrique

La distance entre deux points x et y d'un jeu de donnée peut être définie comme la plus grande valeur entre les *core distances* de x et y et la distance euclidienne entre x et y .

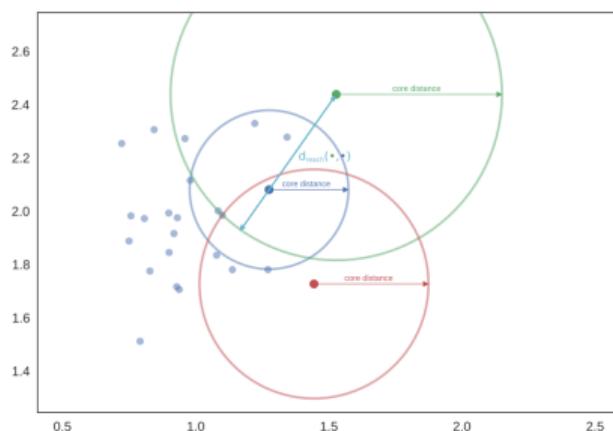


Figure 39 – Nouvelle métrique

Fonctionnement HDBScan²

Principe général

HDBScan consiste à fusionner les méthodes de classification hiérarchique ascendante et DBScan pour permettre de s'affranchir du paramètre ε .

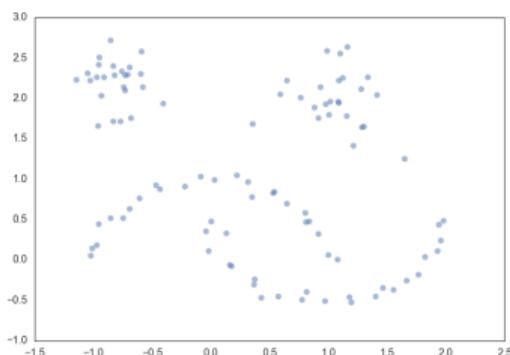


Figure 40 – Le jeu de donnée exemple³

2. http://link.springer.com/chapter/10.1007%2F978-3-642-37456-2_14
3. https://hdbSCAN.readthedocs.io/en/latest/how_hdbSCAN_works.html

Fonctionnement HDBScan

Etape 1 - Construction de l'arbre couvrant de poids minimum

- Représenter les données par un graphe complet où chaque point est un sommet
- Valuer les arêtes par la métrique expliquée précédemment ;
- Trouver un arbre couvrant de poids minimum dans ce graphe ;

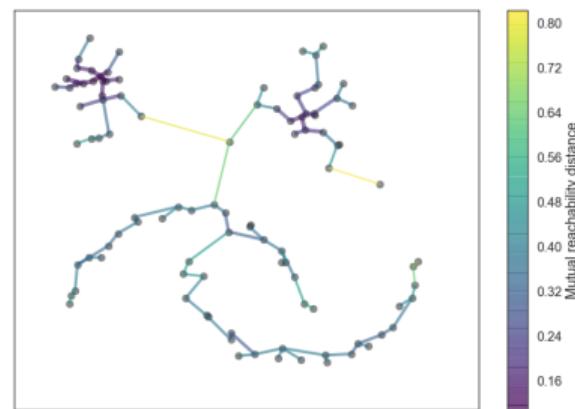


Figure 41 – Arbre couvrant de poids minimum

Fonctionnement HDBScan

Etape 2 - Création du dendrogramme

- Ne garder que les arêtes correspondant à un certain seuil que l'on fait varier.
- Observer l'évolution des classes (composantes connexes) en fonction de cette distance.
- Construire un dendrogramme à partir de ces informations.

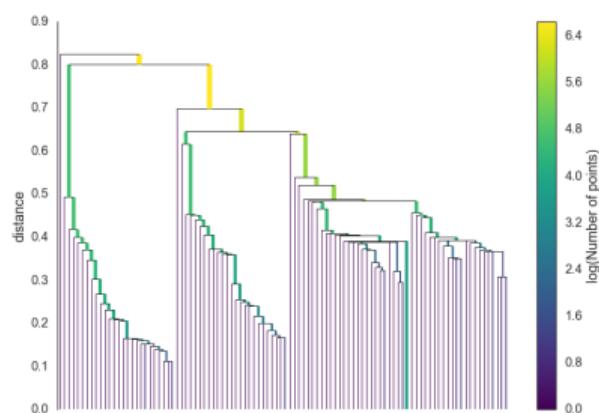


Figure 42 – Dendrogramme

Fonctionnement HDBScan

Etape 3 - Condensation du dendrogramme

On va chercher à condenser le dendrogramme précédent. Pour cela, on considère qu'au départ il n'y a qu'un cluster. On considère qu'un cluster n'est plus considéré comme tel quand il possède moins d'individu qu'un certain seuil (appelons le n_{min}). A chaque séparation dans le dendrogramme, il y'a trois cas de figure :

- Si les deux parties comportent au moins n_{min} individus, alors on considère que le cluster parent meurt et donne naissance à deux enfants.
- Si une seule des parties comporte au moins n_{min} individus, alors on considère que ce cluster est la continuation du précédent.
- Si les deux parties comportent moins de n_{min} individus, alors on considère que le cluster meurt.

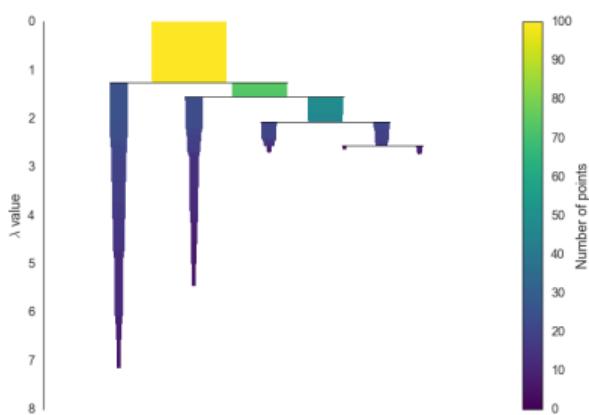


Figure 43 – Dendrogramme condensé

Fonctionnement HDBScan

Etape 4 - Selection des clusters

On va choisir les clusters que l'on veut grader au sein du dendrogramme condensé en choisissant les clusters les plus "durables". Graphiquement, cela revient à choisir les clusters dont l'aire est la plus importante dans le dendrogramme condensé. A la fin on souhaite une partition, donc qu'on ne peut sélectionner à la fois un cluster et son descendant.

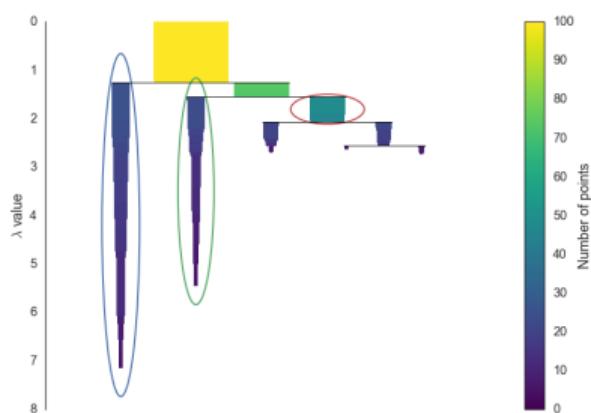


Figure 44 – Clusters choisis

Fonctionnement HDBScan

Etape 5 - Résultat

A la fin, on obtient les clusters ainsi qu'une probabilité pour chaque point qui caractérise la probabilité que le point appartienne à son cluster. Cette probabilité est déterminée en observant à quel moment le point se déconnecte de son cluster dans le dendrogramme.

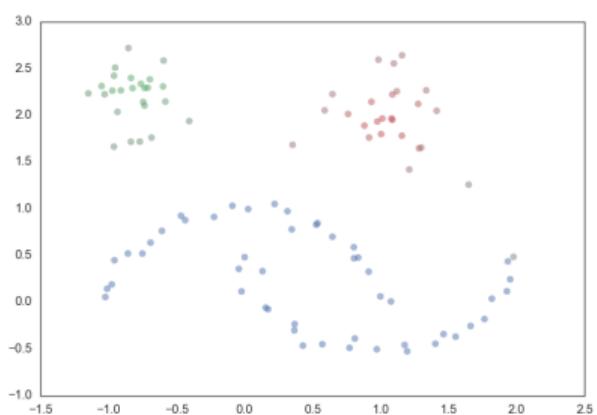


Figure 45 – Résultats sur le jeu de donnée d'exemple

HDBScan

Avantages

- Permet de prendre en compte les problèmes de classe à densité variable ;
- Introduit un modèle probabiliste.

Application à notre cas d'utilisation

- Appliqué de la même manière que DBScan : si un point est dans un cluster, alors on considère qu'il est en ville, sinon en campagne ;
- La probabilité discutée précédemment permet de rajouter une incertitude sur l'appartenance (ou non) d'une station de base à une ville.

Paramètres de `sklearn.cluster.HDBSCAN`⁴

- `min_cluster_size=5` : groupings smaller than this size will be left as noise ;
- `min_samples=None` : The number of samples in a neighborhood for a point to be considered as a core point ;
- `cluster_selection_epsilon=0.0` : Clusters below this value will be merged ;
- `max_cluster_size=None` ;
- `metric='euclidean'` ;
- `metric_params=None` ;
- `alpha=1.0` : A distance scaling parameter ;
- `algorithm='auto'` : Exactly which algorithm to use for computing core distances ;
- `leaf_size=40` : Leaf size for trees responsible for fast nearest neighbour queries when a KDTree or a BallTree are used as core-distance algorithms ;
- `n_jobs=None` : Number of jobs to run in parallel to calculate distances ;
- `cluster_selection_method='eom'` : The method used to select clusters from the condensed tree ;
- `allow_single_cluster=False` : allow the result to be a single cluster ;
- `store_centers=None` ;
- `copy=False`.

⁴ <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.HDBSCAN.html>

Résultats

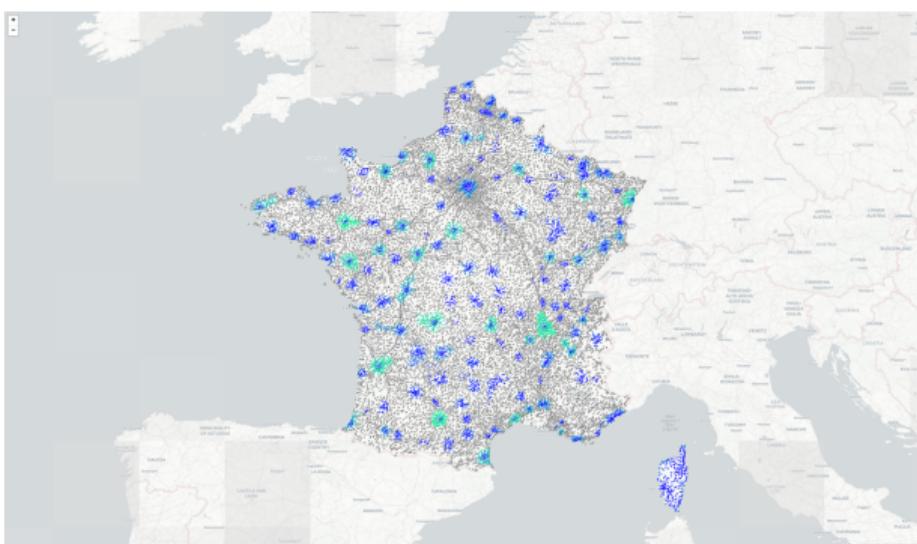


Figure 46 – Application d'HDBScan `min_cluster_size=5, min_samples=40`

Semaine du 27/05/24 au 31/05/24

Semaine du 27/05/24 au 31/05/24

Comparaison des méthodes de détection des villes

HDBScan vs DBScan

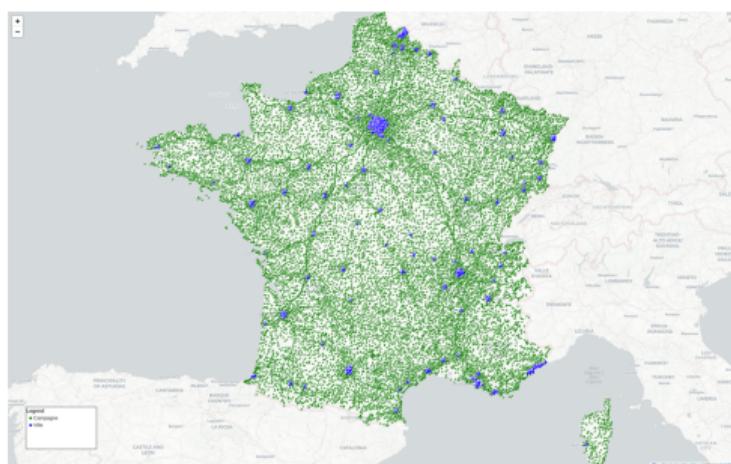


Figure 47 – Les villes détectées en France avec DBScan pour l'opérateur Orange avec $\epsilon = 0.03$ et $n_{min} = 15$



Figure 48 – Les villes détectées en France avec HDBScan pour l'opérateur Orange avec $\text{min_cluster_size}=5$, $\text{min_samples}=40$

Semaine du 27/05/24 au 31/05/24

Amélioration des critères de sélection

Prise en compte de la probabilité d'être dans une ville

Concept

Chaque station possède une probabilité d'être dans une ville. On va donc utiliser ce paramètre afin de moduler l'angle et la distance minimale entre 2 stations.

Choix d'implantation

Pour l'instant, on a séparé les stations en 4 catégories, en fonction de la probabilité d'être une ville :

- $p = 1$: `distance_max = 1` et `min_angle = 45`;
- $p = 0$: `distance_max = 15` et `min_angle = 15`;
- $p \in]1; 0,6[$: `distance_max = 5` et `min_angle = 30`;
- $p \in [0,6; 0[$: `distance_max = 10` et `min_angle = 20`.

Pistes de travail

Il va maintenant falloir effectuer des expérimentations pour trouver les bons paramètres et peut-être essayer de savoir quel paramètre est le plus pertinent.

Améliorations des cadrants

KNN

Dans un premier temps, nous avons décidé de pouvoir choisir le nombre de voisins par cadran que nous souhaitons conserver. Cette amélioration ne semble pas très concluante car la méthode devient trop permissive pour $k \geq 2$ (4190 / 4337 voisins conservés).

Optimisation du positionnement des cadrants

La position de cadran est choisie en testant un angle α qui est décalage angulaire pour positionner le cadran. L'angle α choisi est la valeur entre 0 et 60 degrés qui maximise la distance entre les points et la limite de cadran la plus proche.

Illustration de l'optimisation du positionnement du cadran

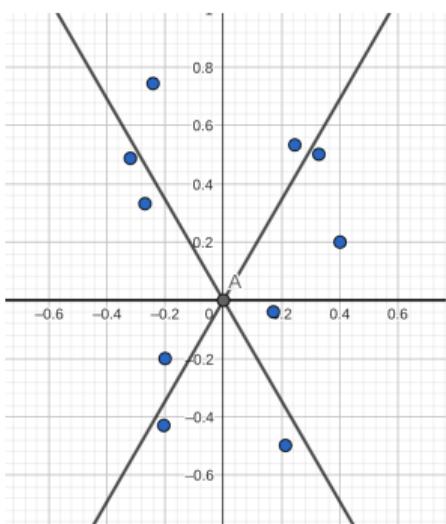


Figure 49 – Quadrants non optimisés

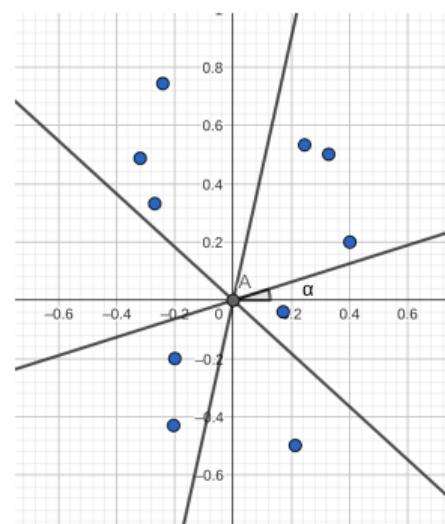


Figure 50 – Quadrants optimisés

Semaine du 03/06/24 au 06/06/24

Semaine du 03/06/24 au 06/06/24

Comparaison des résultats de 2 détection de villes

Affichage

Affichage des différences

Afin de visualiser les différences, nous pouvons tracer les 2 classifications en même temps sur une carte en adoptant ce code couleur :

- Rouge : station en ville dans les 2 classification ;
- Bleu : station en ville dans exactement 1 classification ;
- Vert : station en campagne dans les 2 classifications.

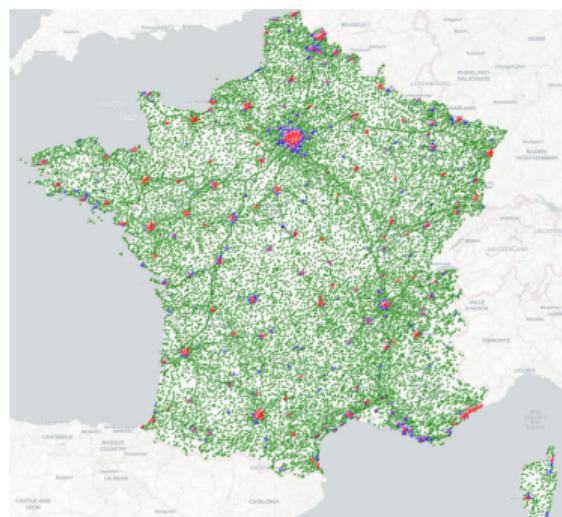


Figure 51 – Comparaison des résultats de détection de villes : DBScan vs HDBScan

Indicateurs

Indicateurs de base

Nous avons développé quatre indicateurs de base afin de caractériser la similarité entre deux classifications :

- a : Le pourcentage de station en ville dans les deux classifications ;
- b : Le pourcentage de station en ville dans la première classification et non la deuxième ;
- c : Le pourcentage de station en ville dans la deuxième classification et non la première ;
- d : Le pourcentage de station en campagne dans les deux classifications.

Résultats

Sur les classifications présentées dans la slide précédentes, voici les résultats obtenus :

- $a = 0,688$;
- $b = 0,028$;
- $c = 0,051$;
- $d = 0,232$.

Semaine du 03/06/24 au 06/06/24

Nouvelle manière de détecter les villes

Changement de cap

Problèmes rencontrés avec DBScan et HDBScan

- Méthodes complexes donc peu prévisibles ;
- Résultats non-satisfaisants quand le méthode est appliquée à seulement une partie de la France.
- DBScan est trop binaire, HDBScan est à densité variable donc ne détecte pas toutes les villes de la même façon (problème avec Paris notamment)
- Les probabilités de HDBScan ne caractérisent pas exactement la probabilité d'être en ville mais plutôt la certitude avec laquelle on peut rattacher un point à son cluster.

Réflexion sur une méthode plus simple

Nous souhaitions savoir quelle station se trouvait en ville, car on sait qu'en ville, les stations sont plus proches les unes des autres, donc les rayons de couverture plus courts. Cependant, il n'est pas nécessaire de détecter les villes pour cela, nous pouvons simplement regarder la distance moyenne aux k plus proches voisins.

Méthodologie

Nouvelle méthode

Au lieu d'utiliser une méthode de clustering, nous allons utiliser quelque chose de plus simple. On classifie chaque station selon la distance moyenne aux 3 plus proches voisins. Soit d cette distance, on regroupe les stations de la manière suivante :

- $d \in]0, 1]$: centre ville dense ;
- $d \in]1, 2]$: couronne périurbaine ;
- $d \in]2, 5]$: campagne ;
- $d \in]5, 10]$: campagne profonde ;
- $d \in]10, \infty[$: trou paumé ou valeur aberrante.

Méthodologie : choix techniques

Calcul des plus proches voisins

Nous utilisons la bibliothèque `sklearn.neighbors.NearestNeighbors`¹.

Choix du nombre de voisins

Après plusieurs expérimentations, nous avons choisi de conserver 3 voisins dans notre calcul. En effet, un chiffre inférieur à 2 serait aberrant car ce ne serait pas une vraie moyenne (ici on cherche plus une mesure de la densité de stations). De plus, un chiffre supérieur à 4 prendrait en compte des stations trop éloignées, ce qui serait aberrant.

Métrique

Nous avons décidé de partir de la projection de Lambert 93 (qui est déjà présente dans nos données) pour calculer cette distance. L'avantage est le gain de temps (environ 30 fois plus rapide), sans perte de performance, par rapport à la conversion en km depuis les coordonnées Longitude, Latitude.

1. <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html>

Mise à jour des critères

Angle

Voici les différents paliers que nous appliquons :

- $d \in]0, 1]$: `angle_min = 40°` ;
- $d \in]1, 2]$: `angle_min = 30°` ;
- $d \in]2, 5]$: `angle_min = 25°` ;
- $d \in]5, 10]$: `angle_min = 15°` ;
- $d \in]10, \infty[$: `angle_min = 10°`.

Distance

Voici les différents paliers que nous appliquons :

- $d \in]0, 1]$: `distance_max = 2 km` ;
- $d \in]1, 2]$: `distance_max = 5 km` ;
- $d \in]2, 5]$: `distance_max = 10 km` ;
- $d \in]5, 10]$: `distance_max = 15 km` ;
- $d \in]10, \infty[$: `distance_max = 15 km`.

From 10/06/24 to 14/06/24

From 10/06/24 to 14/06/24

Modification of criteria

New criteria and city classification

After some reflexions, we have agreed to change the way we classify the city-ness of each base station :

New city-ness classification

- $d \in]0, 1]$: city center ;
- $d \in]1, 2]$: urban area ;
- $d \in]2, 4]$: extra-urban area ;
- $d \in]4, \infty[$: countryside.

Angle

- $d \in]0, 1]$: $\text{angle_min} = 40^\circ$;
- $d \in]1, 2]$: $\text{angle_min} = 30^\circ$;
- $d \in]2, 4]$: $\text{angle_min} = 25^\circ$;
- $d \in]4, \infty[$: $\text{angle_min} = 15^\circ$.

Distance

- $d \in]0, 1]$: $\text{distance_max} = 2 \text{ km}$;
- $d \in]1, 2]$: $\text{distance_max} = 5 \text{ km}$;
- $d \in]2, 4]$: $\text{distance_max} = 10 \text{ km}$;
- $d \in]4, \infty[$: $\text{distance_max} = 15 \text{ km}$.

From 10/06/24 to 14/06/24

City detection results comparison

Reminder : the brute results

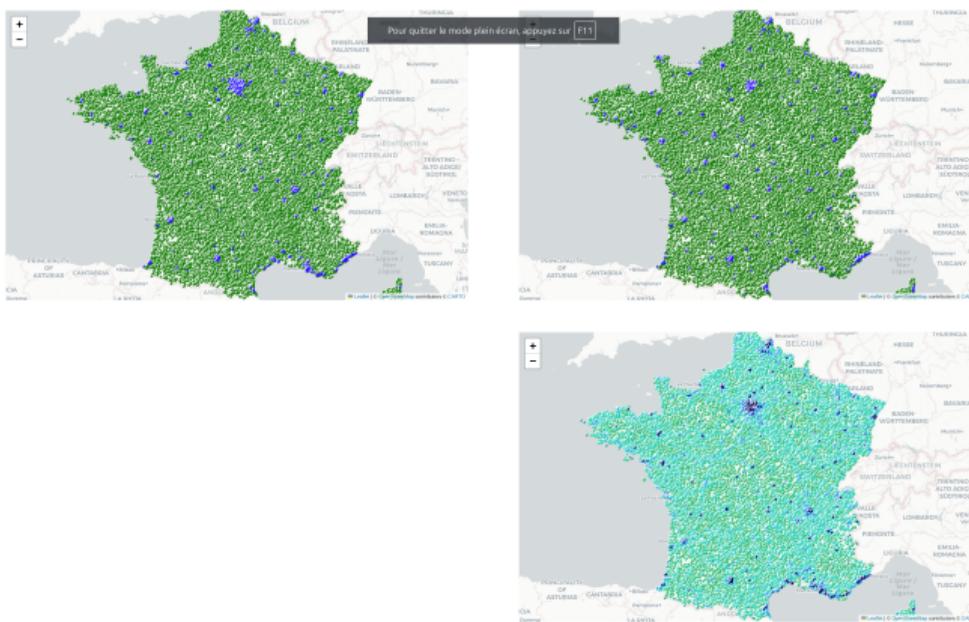


Figure 52 – City detection by respectively DBScan, HDBScan and 3-NN methods

Graphical methods comparison

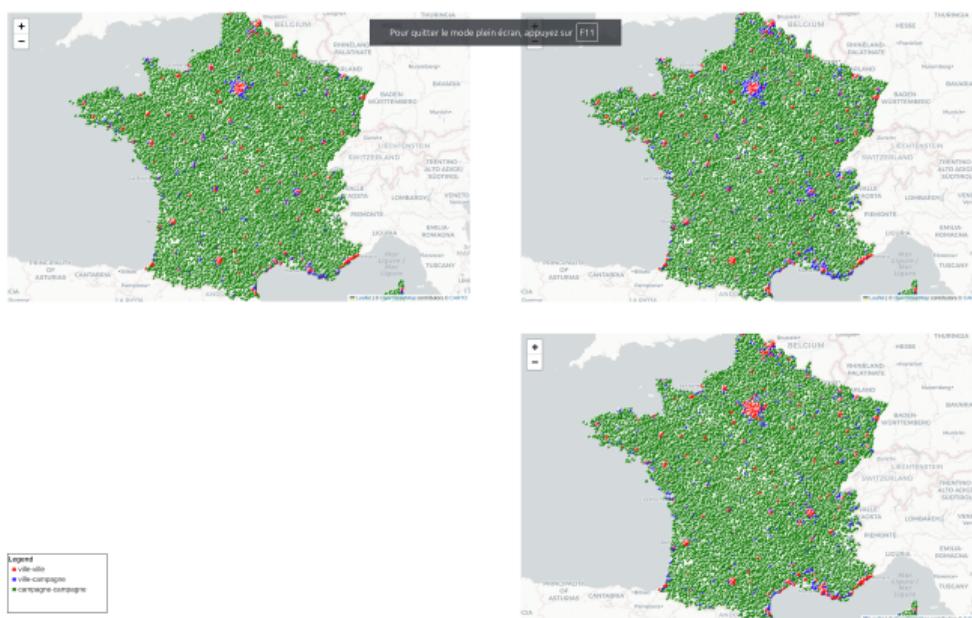


Figure 53 – DBScan vs HDBScan / HDBScan vs 3-NN / DBScan vs 3-NN

Numerical methods comparison

DBScan vs HDBScan

- $a = 0.688$
- $b = 0.028$
- $c = 0.052$
- $d = 0.232$

HDBScan vs 3-NN

- $a = 0.626$
- $b = 0.114$
- $c = 0.010$
- $d = 0.250$

DBScan vs 3-NN

- $a = 0.630$
- $b = 0.086$
- $c = 0.007$
- $d = 0.277$

From 10/06/24 to 14/06/24

Improvement of City detection

Reminder : fonctionnement de base 3-NN

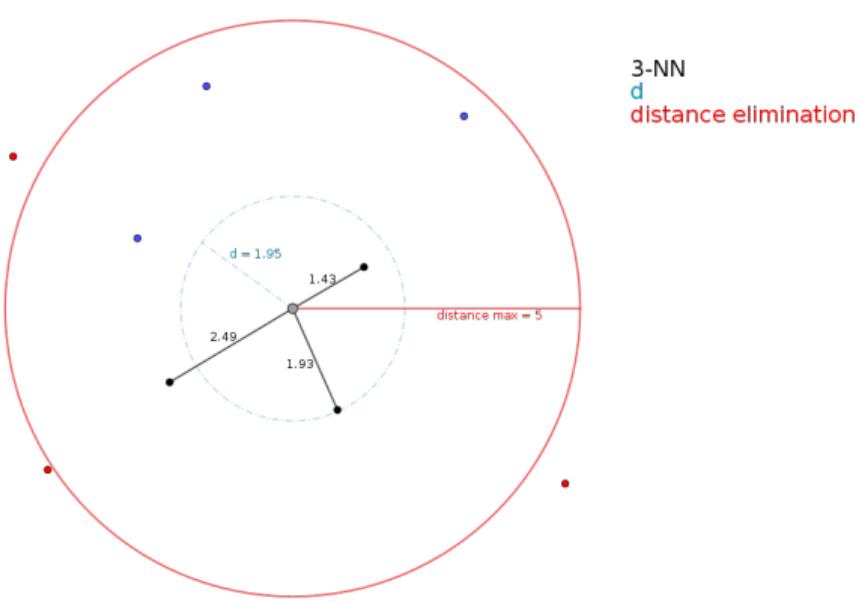


Figure 54 – How 3-NN method works

From 17/06/24 to 21/06/24

From 17/06/24 to 21/06/24

Road detection method - in progress

Methodology

We have seen that one classification with either DBScan or HDBScan isn't enough. So, why not combining them.

The base idea

We will combine the clusters of DBScan, HDBScan and OPTICS (another density base clustering method) to try to find roads.

One problem

We are now able to detect road/railways but the problem is that there is still little clusters that are useless.

New method illustration (1/2)

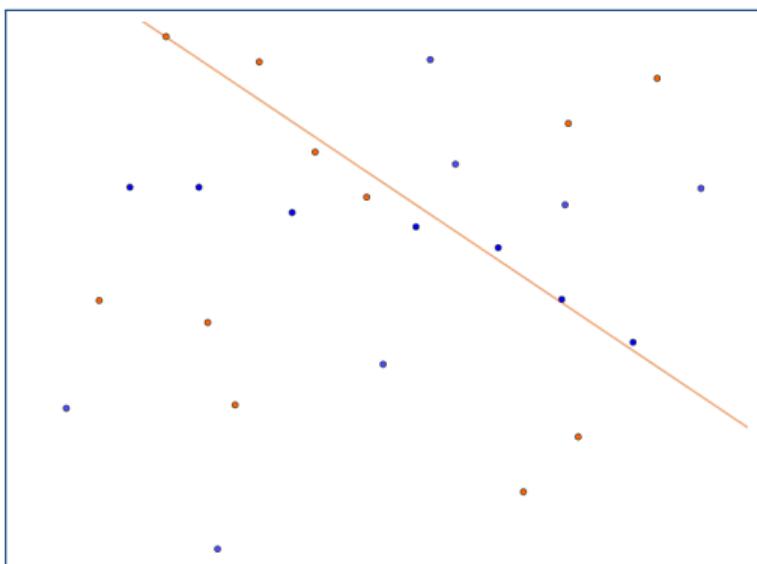


Figure 55 – One clustering method

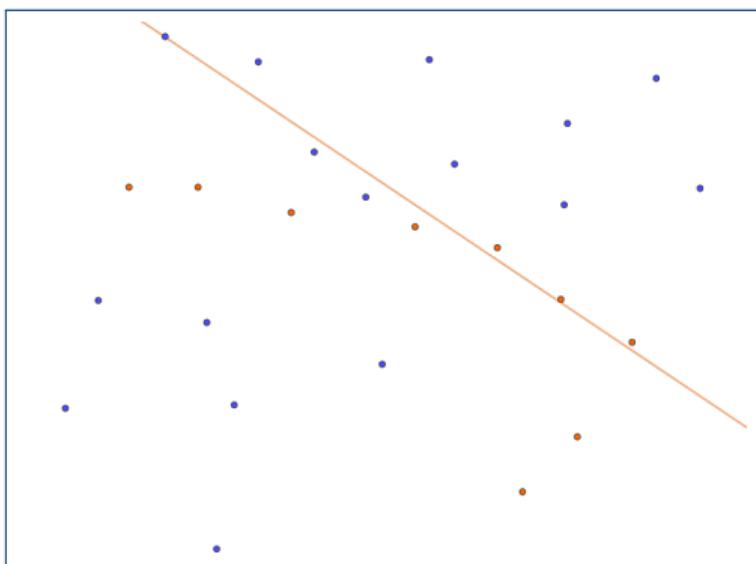


Figure 56 – Another clustering method

New method illustration (2/2)

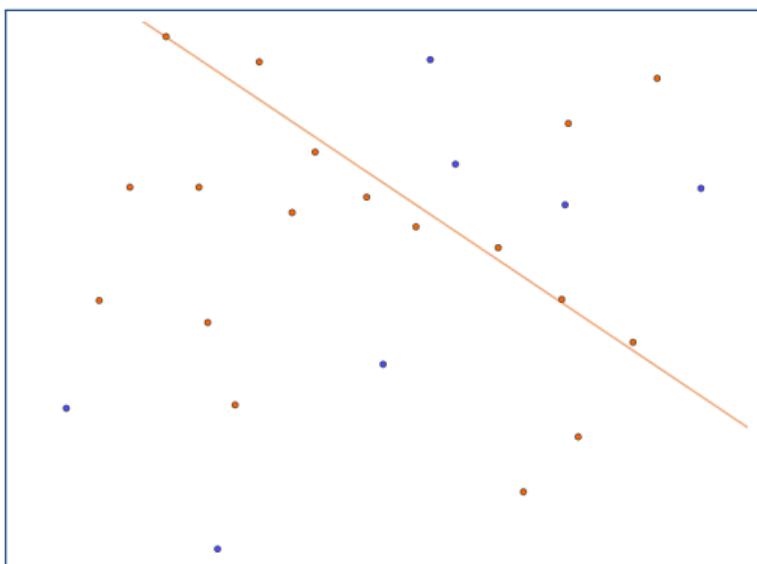


Figure 57 – Merge of the clusters

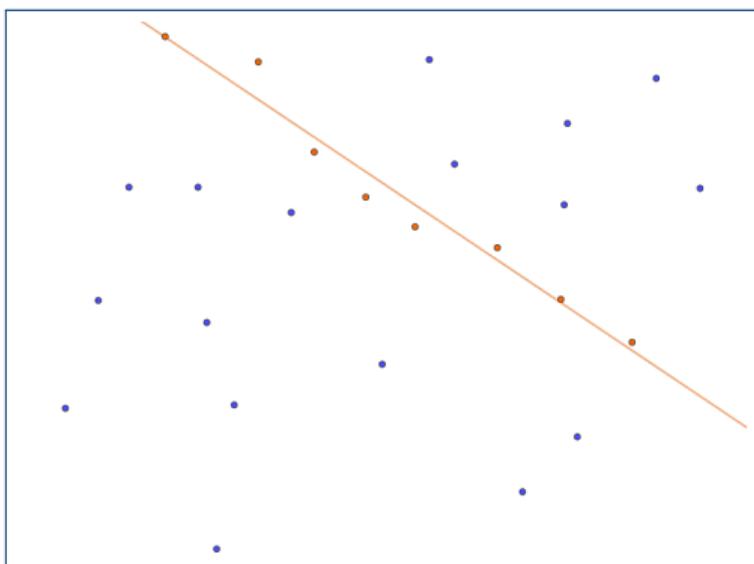


Figure 58 – Elimination of noise

DBScan clustering

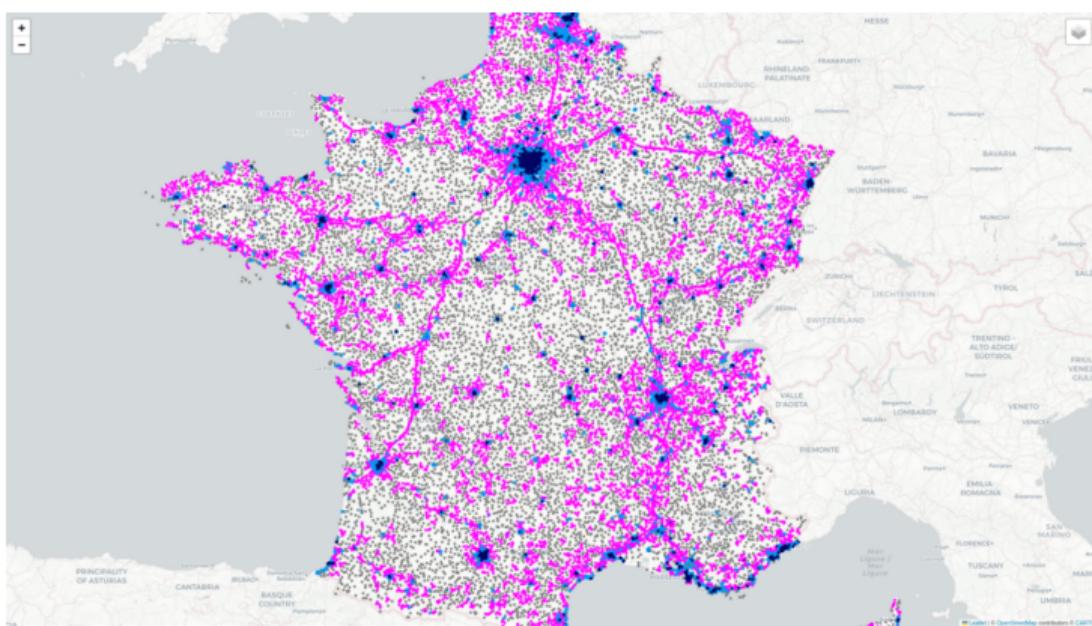


Figure 59 – DBScan clustering

HDBScan clustering

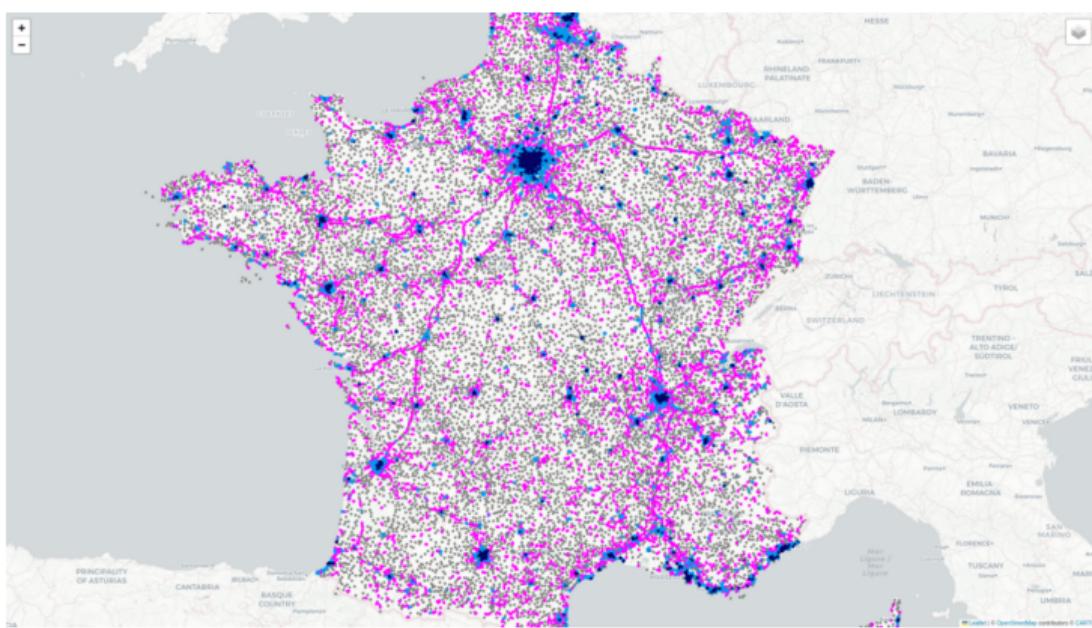


Figure 60 – HDBScan clustering

OPTICS clustering

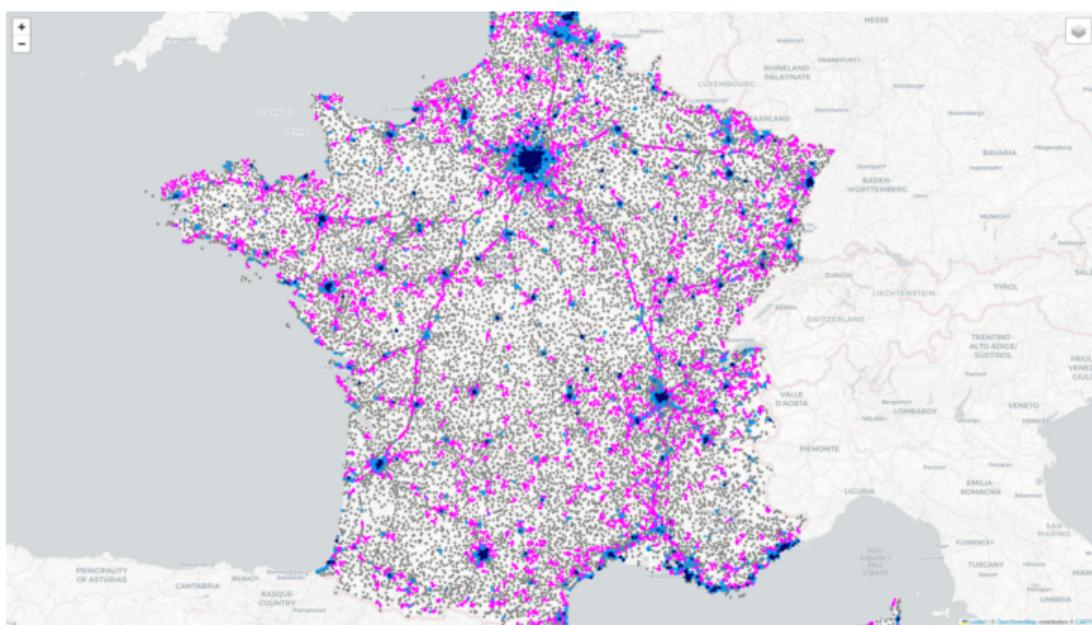


Figure 61 – OPTICS clustering

Result

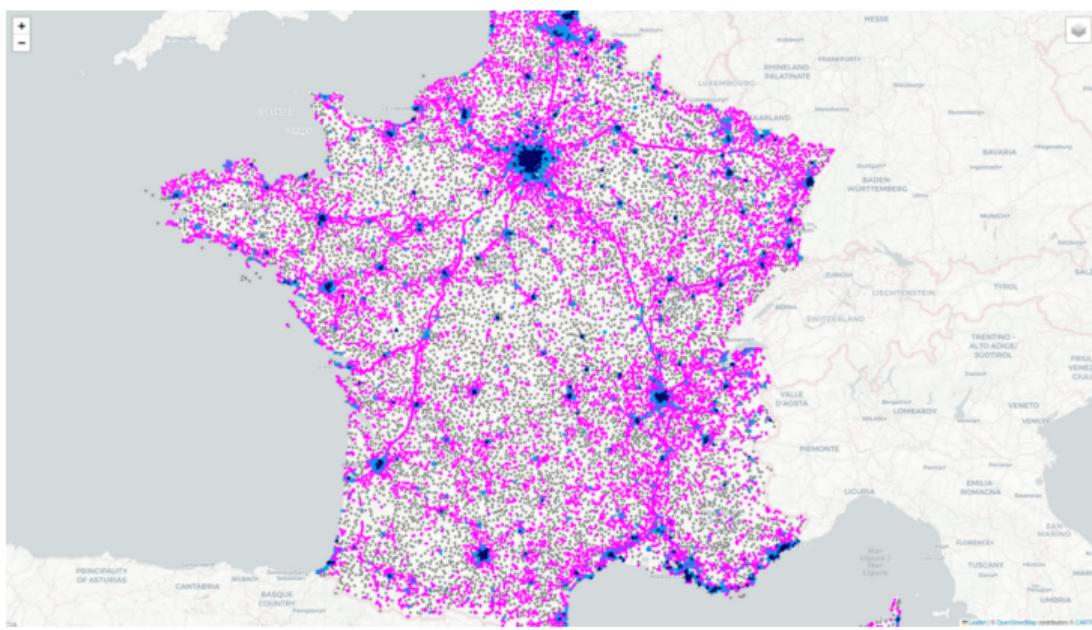


Figure 62 – Road detection

From 17/06/24 to 21/06/24

[**More details and problems**](#)

Detailed clusters detected

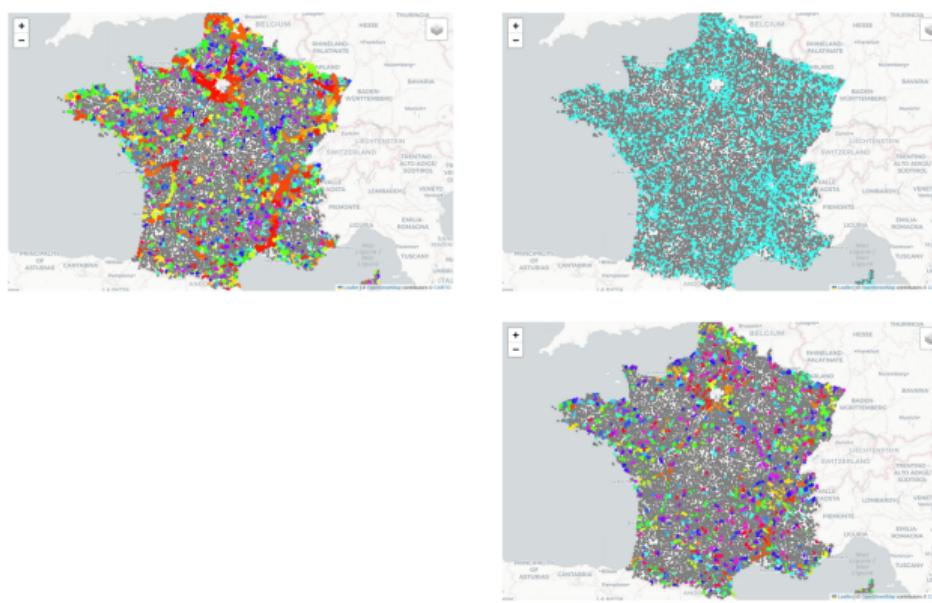


Figure 63 – Detailed clusters detected in countryside by each method

Problems and ideas

Problems

- A lot of little citys detected : not only roads ;
- The areas around big citys are a mess ;
- HDBScan has only one cluster.

Ideas

- Refine the parameters of each method ;
- Use linear regressions to detect parts of roads and maybe help propagate them.

From 17/06/24 to 21/06/24

Altair AI Studio Software Review

Altair AI Studio Software Review

Objective is to evaluate Altair AI Studio for their potential to enhance our research on classifying the terrain of mobile base stations. Available at : <https://altair.com/altair-ai-studio>

Altair AI Studio

This is a platform designed for data analysis and machine learning model building. Possible benefits for us :

- Supports clustering and classification algorithms.
- Support for various machine learning algorithms.
- Enables effective result visualization (graphs) for enhanced analysis.

Key Features :

- Integration with various data sources.
- Interactive model creation and testing.
- Support for various machine learning algorithms.

Users :

- Data researchers
- Analysts
- Machine learning developers

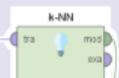
Example of Usage in Altair AI Studio

GEOGRAPHIC DISTANCES: Calculate the nearest antenna to a given client position by using a "1-nearest-neighbor" model of the antennas and applying it to the client position.

Step 1. Retrieve the dataset with antenna positions. Some simple ETL is performed by selecting only the appropriate attributes and indicating the objective of the model: finding a CellId, which will be the model's label.



Step 2. A k-NN model (with k=1) is trained with the dataset. It will help us find the nearest antenna.



Step 3. Retrieve the data from the clients. The objective of the process is to find the nearest antenna to each of them.



Step 4. Apply the 1-NN model to find the nearest antenna (CellId) to each of the clients positions.



Outputs:

- The 1-NN model
- The client locations together with the Id of the nearest antenna.

Result:

ID	Prediction(CellId)	CoordinateX	CoordinateY
1	1251	611814	5636166
2	1305	595066	5627691
3	4187	599336	5588247
4	1305	587844	5601378
5	7131	608191	5634863
6	8757	592244	5659340
7	1305	589278	5589412
8	312	616828	5662313
9	1305	580590	5626004
10	1305	590716	5624514

Figure 64 – How Altair AI Software works

From 17/06/24 to 21/06/24

New Possible Approach for Terrain Classification

More accurate verification of the classification of base station locations

We want to validate and enhance the previous classification of base station locations by leveraging a new geographic dataset. We found and will try to use a comprehensive dataset from data.enseignementsup-recherche.gouv.fr, which includes precise geographic data points across France with pre-defined zone classifications.

Methodology for Base Station Classification

Previously, we worked with a dataset containing locations of all base stations. Now, we utilize an additional dataset with geographic points across France, each classified into zones (urban, suburban, rural). By comparing base station coordinates with these geographic points, we determine the zone of each base station. The closest geographic point's classification is assigned to the base station, providing a more accurate terrain classification.

Benefits :

- Cross-reference base station coordinates with geographic data points to accurately assign zone classifications.
- Provides an additional layer of verification and accuracy for our classification methods.

Classification of points in the new dataset

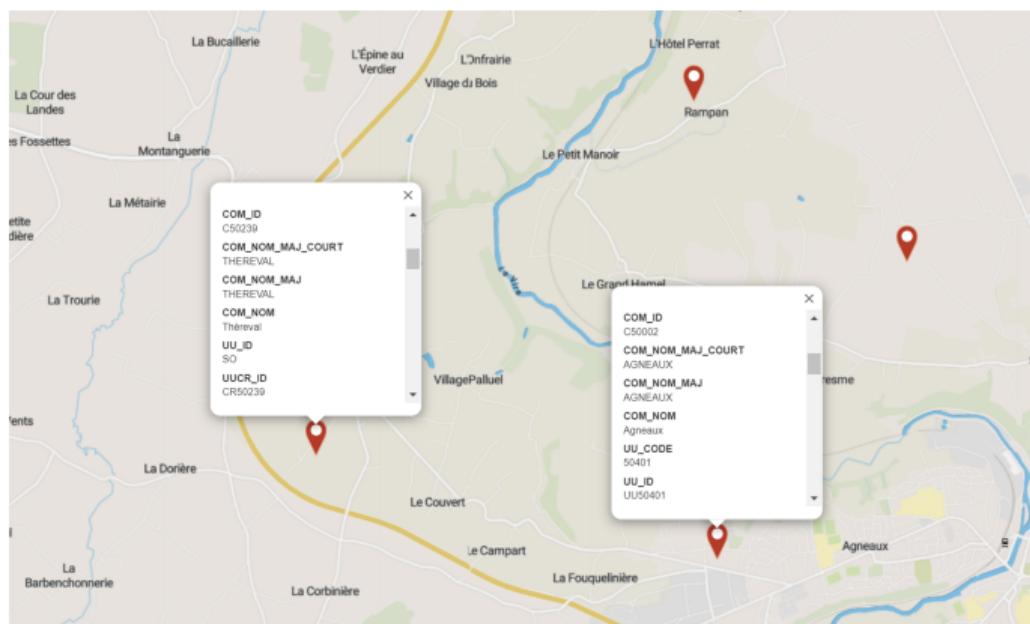


Figure 65 – Classification of points from the new dataset

Classification of points in the new dataset

Geographic identifiers :

- **UU (Urban Units)** : Areas classified as urban units.
- **CR (Rural Communes)** : Areas classified as rural communes.
- **AU (Urban Areas)** : Larger urban areas or agglomerations.

Methodology and Challenges

This method helps classify base station locations by comparing them with geographic data points. However, we need to preprocess the dataset carefully and address the issue of having a limited number of points. To ensure high accuracy, we might need to augment the data or combine it with other datasets for better coverage and precision

From 17/06/24 to 21/06/24

Another new database

New possibilities on city/countryside classification

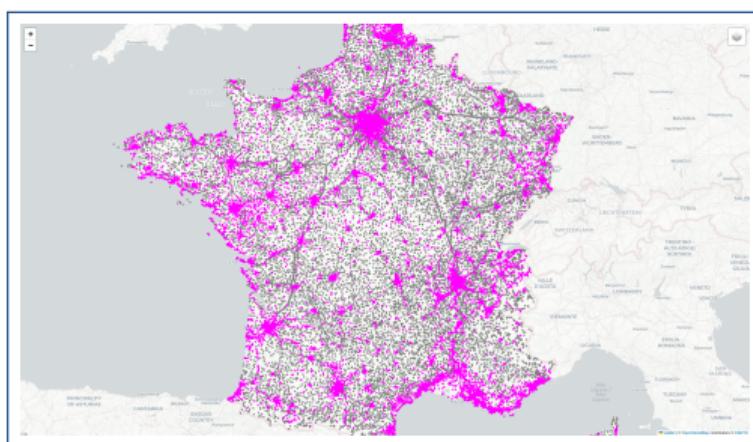


Figure 66 – Database we presented last week

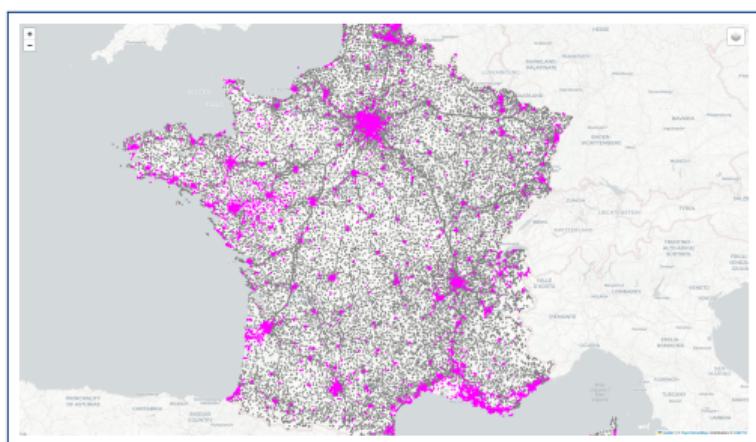


Figure 67 – A new database

What's this ?

We found a new database on the population of each french commune in 2021².

INSEE ?

It is the National Institute of Statistics and Economics Studies. It collects, analyses and disseminates information on the French economy and society.

Why this one ?

Firstly, this database is really recent (updated on 26/01/2024). Then, the other database we introduced was too permissive in the definition of what is a city (a commune with more than 2000 inhabitants). So we wanted something more flexible.

2. <https://www.insee.fr/fr/statistiques/7739582>

From 17/06/24 to 21/06/24

Back to road detection

HDBScan parameters adaptation

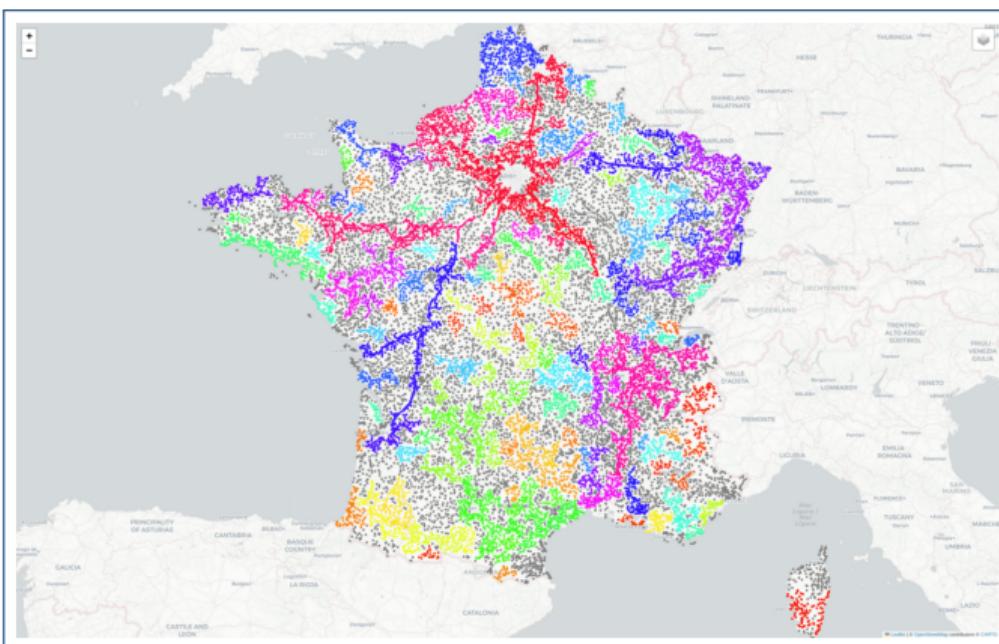


Figure 68 – Result of HDBScan with new parameters on countryside to detect roads

A bit of math theory (1/2)

Multiple linear regression

Let a dataset be composed of m vectors $(x^i)_{i \in \{1 \dots m\}}$ of \mathbb{R}^n , ie $\forall i \in \{1, \dots, m\}$, $x^i = (x_1^i, \dots, x_n^i)$.

Let $(y^i)_{i \in \{1 \dots m\}}$ be m associated values in \mathbb{R} .

Then, we can find a linear function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (ie $f(x_1, \dots, x_n) = c_1 x_1 + \dots + c_n x_n$) that minimises the error :
 $\|\sum_{i=1}^m (f(x^i) - y^i)\|$

f is called a multiple linear regression associated to $(x^i)_{i \in \{1 \dots m\}}$ and $(y^i)_{i \in \{1 \dots m\}}$.

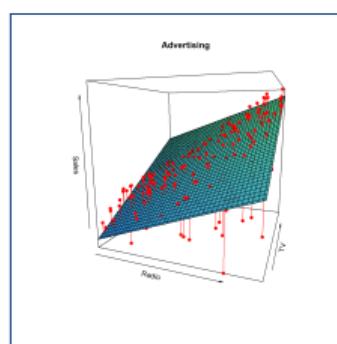


Figure 69 – Example of multiple linear regression with $n = 2$

A bit of math theory (2/2)

Polynomial regression

Let a dataset be composed of m values $(x^i)_{i \in \{1 \dots m\}}$ of \mathbb{R} .

Let $(y^i)_{i \in \{1 \dots m\}}$ be m associated values in \mathbb{R} .

We can transform the data in m vectors $(x^i)_{i \in \{1 \dots m\}}$ of \mathbb{R}^{n+1} defined by

$(x_0^i, x_1^i, x_2^i, \dots, x_n^i) = (1, x^i, (x^i)^2, \dots, (x^i)^n)$, and then apply the multiple linear regression seen earlier on this transformed data.

We therefore obtain a function

$$f : \begin{cases} \mathbb{R} \rightarrow \mathbb{R} \\ x \mapsto c_1 + c_2 x + \dots + c_n x^n \end{cases}$$

That is a degree n polynome that approximates the relation between x and y .

Quantify the quality of the approximation

As for the simple linear regression, we obtain a R coefficient between 0 and 1 that indicates if the approximation is close to the original data (ie if the polynome we find at the end is close to the real appearance of the data).

Application of the method on each cluster of hdbSCAN

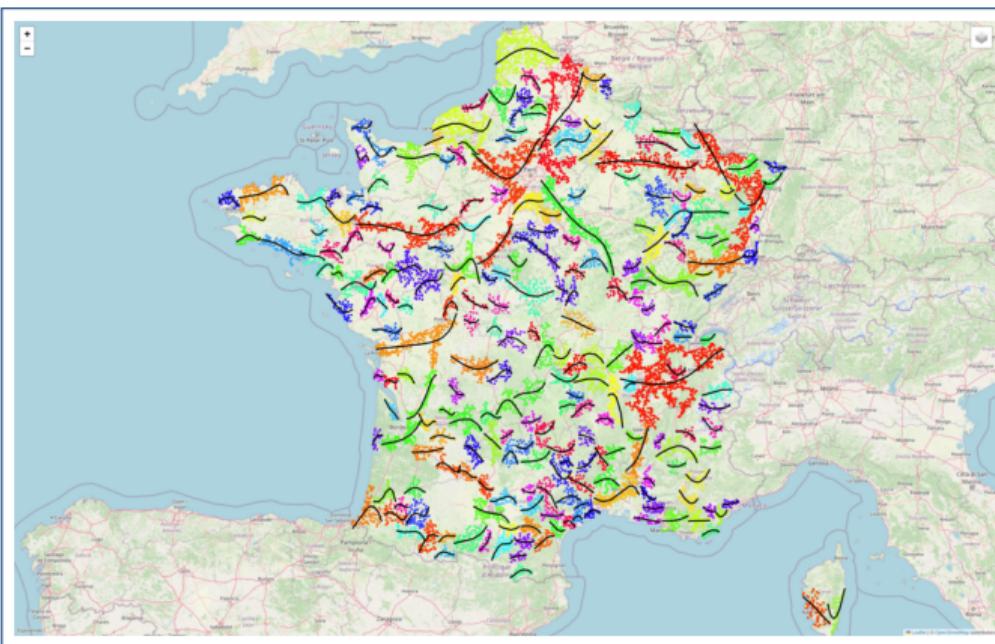


Figure 70 – Approximation of each cluster by a degree 20 polynome

Filtration of the cluster by keeping the ones that have a R coefficient above a value

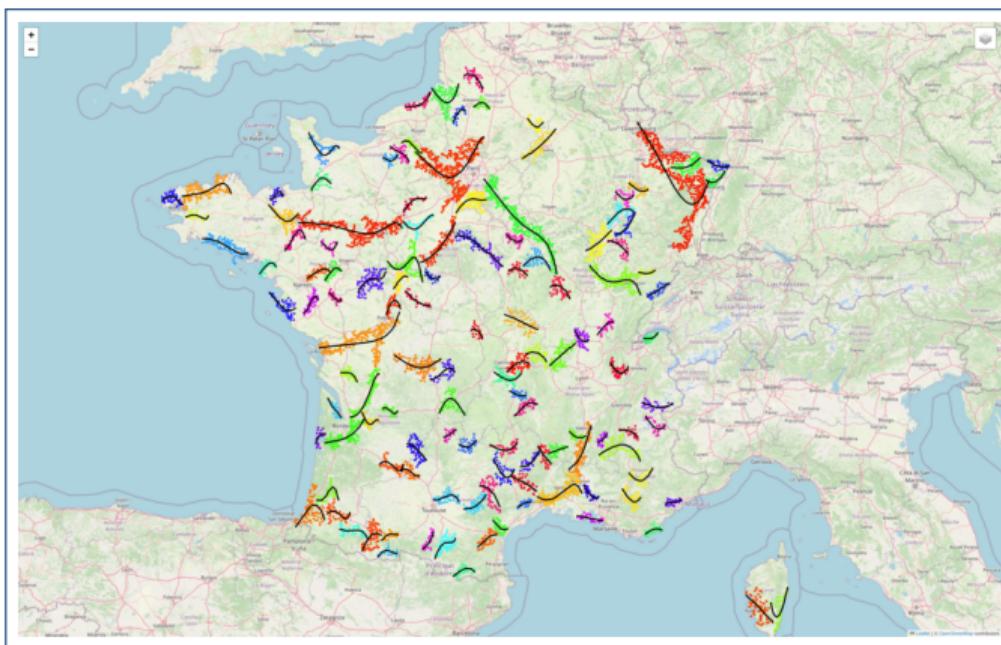


Figure 71 – Approximation of each cluster by a degree 20 polynome, keeping only the ones with a R coefficient above 0.2