

Rapport de projet tutoré

L'ACCESSIBILITÉ
POTENTIELLE LOCALISÉE
(APL)

DELMAS Léo
MASSIAS Paul
VILLEDIEU Baptiste
FIE-5 - Promotion 2024

Commenditaires: INSA Strasbourg
Représenté par: GOEPP Virginie
& COULIBALY Amadou

Tuteur école: LAMINE Elyes

Table des matières

Liste des Figures	3
Remerciements	4
Glossaire	5
I. Introduction	6
1. Contexte	6
2. Enjeux et objectifs	7
3. Livrable du projet	8
II. Analyse	9
1. Analyse de l'application existante	9
2. La méthode de calcul	10
III. Recherches	12
1. Le calcul et les données	12
2. Définition de nouveaux objectifs	12
IV. Réalisations	13
1. Partie cartographie	13
1.1 - Analyse	13
1.2 Conception et développement	16
1.2.1 Maquettage	16
1.2.2 Choix technologiques	17
1.2.3 Développement de l'application	17
2. Partie calcul	19
2.1 Analyse	19
2.2 Développement	21
V. Gestion de Projet	22
VI. Conclusions et Perspectives	24
1. Les limites	24
2. Devenir du projet	24
Bibliographie	26
Annexes	27

Liste des Figures

Figure 1 : Diagramme de classes du système de soins en France.....	8
Figure 2 : Extrait de l'application Shiny du DREES.....	9
Figure 3 : Tableau de la conversion des activités des professionnels de santé en ETP.....	10
Figure 4 : Tableau de la conversion des distances qui sépare le patient d'un professionnel de santé.....	11
Figure 5 : Extrait de notre tableau récapitulatif des recherches.....	12
Figure 6 : Schéma fonctionnel de la solution.....	13
Figure 7 : Diagramme des cas d'utilisation de notre application.....	14
Figure 8 : Modèle Logique de Données de notre base de données.....	15
Figure 9 : Maquette de l'application.....	16
Figure 10 : Fond de carte Leaflet.....	17
Figure 11 : Schéma de panachage des informations.....	18
Figure 12 : Extrait de code de la méthode d'import de fichiers.....	19
Figure 13 : Extrait du GitLab sur l'architecture du calcul.....	19
Figure 14 : Extrait du code GitLab permettant d'importer les données.....	20
Figure 15 : Exemple de tableau idéal pour le calcul de la pondération.....	21
Figure 16 : Extrait du code Python du calcul de la valeur ETP.....	21
Figure 17 : Diagramme de Gantt initial de notre projet.....	22
Figure 18 : Diagramme de Gantt final de notre projet.....	23
Figure 19: Proposition de projection pour la suite du projet.....	25

Remerciements

Nous tenons d'abord à remercier Madame Virginie GOEPP et Monsieur Amadou COULIBALY pour nous avoir fait confiance tout au long de ce projet. Leur disponibilité a été une source de motivation non négligeable et à jouer un grand rôle dans ce projet. Les conseils avisés de Madame GOEPP et Monsieur COULIBALY nous ont permis d'aller dans la bonne direction durant toute la durée du projet. Nous les remercions pour leur contribution précieuse et sommes reconnaissants d'avoir pu collaborer avec eux.

Nous remercions aussi notre tuteur Monsieur Elyes Lamine qui a été force de proposition lors de ce projet.

Nous remercions ensuite Monsieur Bastide pour avoir créé un environnement propice au bon déroulement de ce projet tutoré pendant la durée de celui-ci.

Glossaire

APL : Accessibilité potentielle localisée

DREES : Direction de la recherche, des études, de l'évaluation et des statistiques

IRDES : Institut de recherche et de documentation en économie de la santé

CNRS : Centre national de la recherche scientifique

ENGEEES : École nationale du génie de l'eau et de l'environnement de Strasbourg

SNIIRAM : Système national d'information inter-régimes de l'Assurance maladie

ETP : Équivalents temps plein

I. Introduction

1. Contexte

L'accès aux soins en France est au cœur des préoccupations sociétales, soulignant la nécessité de garantir une santé publique équitable et de qualité pour l'ensemble de la population. Le système de santé français se repose sur les principes de solidarité et d'universalité. L'assurance maladie, financée par les cotisations et les impôts, assure la couverture maladie en France.

Cependant, la qualité et la rapidité de l'accès aux soins peuvent varier en fonction de divers facteurs, tels que la localisation géographique, la disponibilité des professionnels de santé et les inégalités socio-économiques. Les déserts médicaux sont l'exemple parfait de zones géographiques où il y a une qualité à l'accès aux soins relativement faible. Ces régions, situées principalement en milieu rural, se caractérisent par une pénurie significative de professionnels de santé qui peut entraîner des retards dans la prise en charge des patients et peut augmenter les inégalités en matière de santé. Les déserts médicaux soulignent la nécessité d'attirer et de maintenir des professionnels de santé dans ces régions.

L'accessibilité potentielle localisée (APL), développée par la DREES et l'IRDES en 2012, permet de mettre en évidence ces zones géographiques. L'APL est un indicateur local, disponible au niveau de chaque commune, qui tient compte de l'offre et de la demande de soins issues des communes environnantes. Aujourd'hui, en France, cette valeur est disponible pour l'ensemble des territoires français pour certains professionnels de santé libéraux : les médecins généralistes, les infirmiers, les masseurs-kinésithérapeutes et les sage-femmes. En outre, cette méthode de calcul ne s'applique pas encore aux autres professionnels de santé libéraux tels que les spécialistes.

C'est pourquoi ICube affilié à l'INSA Strasbourg a sollicité l'aide de l'école ISIS afin de trouver une solution dans le but de développer l'APL à l'ensemble de professionnels de santé libéraux. ICube est un laboratoire de recherche multidisciplinaire créé en 2013 et basé à Strasbourg. Celui-ci rassemble des chercheurs issus de différentes disciplines telles que l'informatique, l'imagerie, les sciences de l'ingénieur, les mathématiques et les sciences de la vie. Ce projet est sous l'égide du CNRS, de l'INSA Strasbourg, de l'université de Strasbourg et de l'ENGEEES compte près de 650 membres.

Ce projet s'inscrit donc dans une problématique actuelle dans le monde de la santé dont nous avons pu comprendre les enjeux et les objectifs.

2. Enjeux et objectifs

En France, il y a des disparités géographiques dans l'offre de soins, les zones rurales disposent d'un accès limité à la médecine. La méthode de calcul de l'APL permet d'avoir des informations sur ces régions et de détecter de potentiels déserts médicaux. Cependant, ces données ne s'adaptent pas à toutes les professions de santé comme les spécialistes. Ainsi, il existe un flou sur l'accès à ce type de soins. Pourtant, il serait nécessaire aujourd'hui d'obtenir les données de l'APL de tous les professionnels de santé libéraux.

En effet, cela permettrait de détecter les zones géographiques qu'il faut renforcer dans l'offre de soins pour améliorer la prise en charge et ne pas laisser certaines personnes sans soin. Ces faiblesses peuvent être causées par différentes choses et pas seulement par la pénurie de professionnels dans certaines localités. Par exemple, dans certaines régions, il existe un réel manque d'information sur les services de santé présents, les habitants ne sont pas informés sur l'offre de soins et leurs recherches peuvent être entravées.

De plus, certains individus rencontrent des obstacles financiers causés par des frais de déplacements par exemple et ne peuvent donc pas bien se soigner. L'universalité de l'APL à toutes les professions de santé libérales pourrait permettre d'appuyer les zones géographiques où l'accès est limité ou encore d'en découvrir de nouvelles.

L'accès à ces données aiderait les décideurs du monde de la santé à optimiser la localisation des services de santé. La solution voulue a pour objectif d'identifier et de cartographier les services de santé existants en prenant en compte leur localisation ainsi que leurs activités. Cela permettra de tirer des conclusions et d'optimiser l'offre de soins dans certaines régions. Il assurerait aussi un suivi temporel de l'indicateur d'APL et des ressources de soins présentes comme les établissements de santé ou les professionnels. En poursuivant ces objectifs, le projet contribue à l'amélioration de la santé globale de la population en favorisant un accès plus équitable et efficace aux services médicaux.

3. Livrable du projet

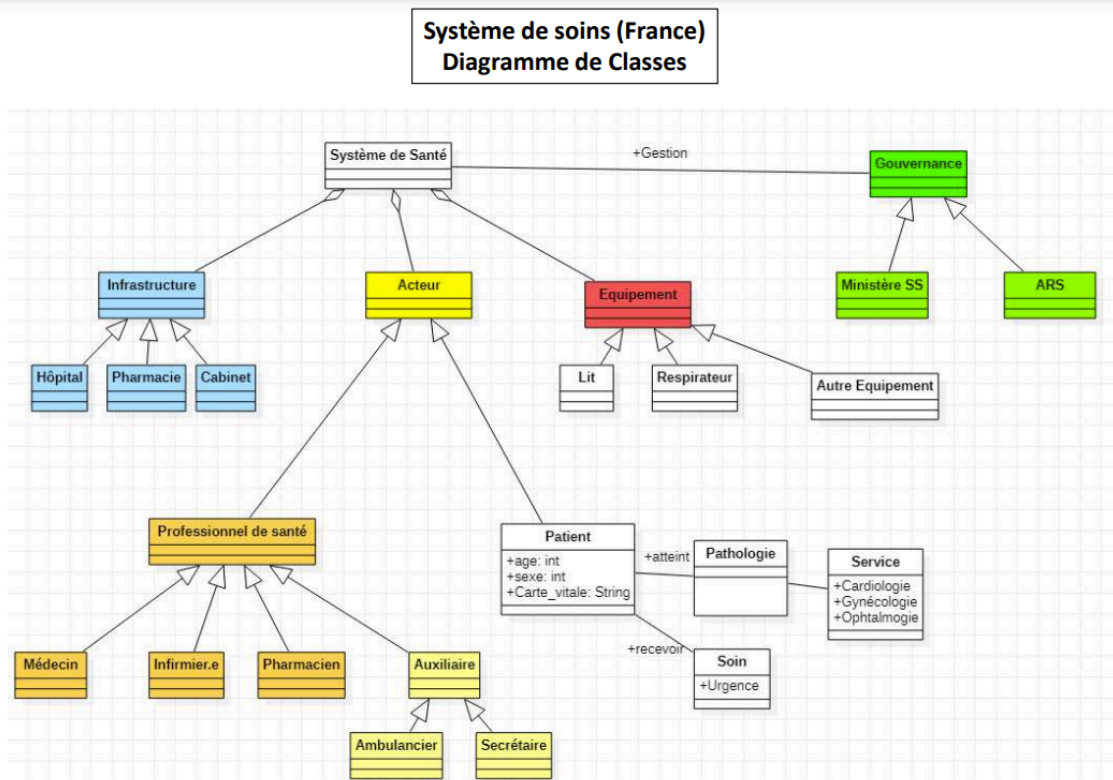


Figure 1 : Diagramme de classes du système de soins en France

En début de projet nous a été fourni ce diagramme de classes du système de soin en France. L'objectif final étant que tout le système de soins défini soit représenté sur la cartographie afin d'avoir une vue sur les différents éléments qui le composent au global. Néanmoins dans le cadre de ce projet tutoré nous nous sommes occupés de la partie de la représentation qui concerne les "Professionnels de santé".

Ce projet tutoré a pour mission de couvrir ce besoin et d'élargir le calcul de l'accessibilité potentielle localisée à l'ensemble des professionnels de santé libéraux. Pour cela, initialement, il nous était demandé de réaliser une application web de cartographie. Dans celle-ci, il fallait afficher les professionnels de santé et les infrastructures de soins pour une certaine zone à un instant donné. Ensuite, le calcul et la visualisation de l'APL pour les professionnels de santé libéraux ainsi qu'une documentation des recherches et de la solution représentent les principales fonctionnalités que doit contenir le livrable de ce projet.

II. Analyse

1. Analyse de l'application existante

Comme énoncé précédemment, il existe déjà aujourd'hui une application développée par la DREES assurant la cartographie de l'APL en France. Elle se nomme Shiny et donne accès aux informations sur l'APL au niveau des médecins généralistes, des sage-femmes, des infirmiers, des masseurs-kinésithérapeutes et des chirurgiens-dentistes. Sur cette application, il est possible d'observer l'indicateur de l'année 2015 à celle de 2022. De plus, elle offre la possibilité d'indiquer une borne d'âge sur les professionnels de santé avec trois paramètres différents : moins de 65 ans, moins de 62 ans et sans borne d'âge. Il existe aussi différents onglets, un se nommant « Tableau » qui donne accès à la donnée de l'APL selon la commune en détaillant la valeur de l'indicateur ainsi que la population standardisée. Le dernier onglet s'appelle « Téléchargement » et permet d'exporter les données de l'APL par commune en France mais aussi l'APL agrégée au niveau du territoires.

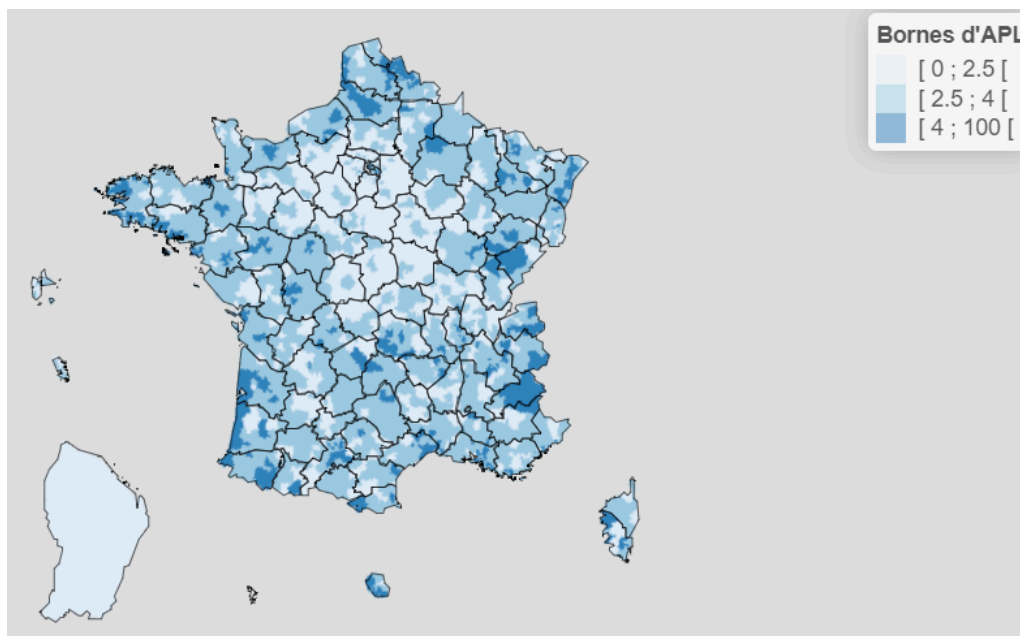


Figure 2 : Extrait de l'application Shiny du DREES

L'application Shiny affiche la cartographie ci-dessus de l'APL au niveau des médecins généralistes de moins de 65 ans pour la France entière. Il est possible de détecter certaines zones géographiques où la valeur de l'indicateur est très basse. En effet, d'après la légende, toutes les régions représentées en blanc ont une valeur de l'APL minimale et plus la couleur est claire moins l'APL est élevée. Ainsi, Shiny permet de découvrir des régions où l'accès aux soins est compliqué et de potentiels déserts médicaux.

Une représentation comme celle-ci est intéressante et il faudrait s'en inspirer pour notre application même s'il faut la généraliser à l'ensemble des professionnels de santé libéraux.

2. La méthode de calcul

Pour calculer ces données présentes dans l'application, la DREES et l'IRDES a créé une méthode particulière en 2012. Celle-ci a été modifiée en 2015 dans une logique d'utilisation opérationnelle de l'indicateur pour la construction de zonages des professionnels de santé. Dans l'indicateur de l'APL, nous ne prenons en compte seulement les professionnels de santé exerçant hors hôpital à titre libéral ou alors les salariés d'un centre de santé (valable pour les infirmiers, les médecins généralistes et les chirurgiens-dentistes). De plus, pour les infirmiers, les sage-femmes, les masseurs-kinésithérapeutes et les chirurgiens-dentistes, seulement les professionnels de moins de 65 ans sont pris en compte. Ensuite, il faut comptabiliser l'offre de soins par l'intermédiaire des ETP (équivalents temps plein). Pour obtenir les ETP, il faut réaliser une conversion en fixant la valeur d'un ETP au niveau de l'activité médiane libérale. Puis, en rapportant cette activité à la valeur médiane nous obtenons la valeur de l'ETP. Si un professionnel possède une activité inférieure à la borne minimale, alors sa valeur d'ETP est nulle. Au contraire, si elle est supérieure à la borne maximale, l'ETP est de 1.

Déciles de distributions	ETP
<5%	0
5 à 10%	0,2
10 à 25%	0,5
25 à 50%	0,7
>50%	1

Figure 3 : Tableau de la conversion des activités des professionnels de santé en ETP

La répartition des ETP se réalise comme sur le tableau ci-dessus et ces valeurs seront prises en compte dans le calcul de l'APL. En outre, la comptabilisation de la demande soins apparaît dans la méthode. Celle-ci est standardisée par la structure par âge de la population, afin de faire la distinction entre certaines classes d'âge. Pour cela, le poids d'une tranche d'âge est calculé comme la consommation moyenne de soins de cette tranche d'âge, rapportée à la consommation moyenne nationale. Ainsi, pour obtenir la valeur utilisée dans le calcul, il faudra multiplier les populations standardisées par âge avec les poids de pondérations.

L'offre de soins disponible par habitant décroît avec la distance. Pour les médecins généralistes, les infirmières, les masseuses-kinésithérapeutes et les chirurgiens-dentistes, l'offre est parfaitement accessible à moins de 10 minutes de la commune de résidence du patient, puis elle est réduite d'un tiers entre 10 et 15 minutes, de deux tiers entre 15 et 20 minutes. Elle est considérée comme nulle au-delà. Pour les sage-femmes, à moins de 10 minutes elle est considérée comme parfaitement accessible, puis réduite d'un tiers jusqu'à moins de 20 minutes. Jusqu'à 30 minutes, elle est encore réduite d'un tiers et au-delà elle est considérée comme nulle. Ces données sont rassemblées dans le tableau ci-dessous :

Distance au professionnel	Médecins généralistes	Infirmières	Masseuses-kinésithérapeutes	Sages-femmes	Chirurgiens-dentistes
Moins de 10 minutes	1	1	1	1	1
Entre 10 et 15 minutes	2/3	2/3	2/3	2/3	2/3
Entre 15 et 20 minutes	1/3	1/3	1/3	2/3	1/3
Entre 20 et 30 minutes	0	0	0	1/3	0
Plus de 30 minutes	0	0	0	0	0

Figure 4 : Tableau de la conversion des distances qui sépare le patient d'un professionnel de santé

Avec toutes ces valeurs récupérées et converties, nous obtenons tout d'abord le calcul du ratio.

$$R_j = \frac{m_j}{\sum_{d_{ij} < d_0} p_i * w(d_{ij})}$$

Pour une commune j, le ratio est calculé à partir de la valeur m qui représente l'offre des médecins en ETP de la commune. Il faut ensuite multiplier le nombre d'habitants standardisés par leur structure d'âge des communes i (p_i) situées à une distance de j inférieure à d_0 . Sachant que d_0 est déterminée à partir du moment où la conversion est considérée comme nulle. Par exemple, pour un médecin généraliste, $d_0=20$ minutes alors que pour les sage-femmes $d_0=30$ minutes. La somme de ses multiplications donne la valeur du dénominateur du ratio.

Enfin, avec le ratio calculé, le calcul de l'APL est possible.

$$APL_i = \sum_{d_{ij} < d_0} w(d_{ij}) R_j$$

L'APL de la commune i est calculé à partir des ratios des communes j situées à une distance inférieure à d_0 , cette distance est la même que pour le calcul du ratio. En additionnant les valeurs des multiplications du ratio avec la distance, la valeur de l'APL pour la commune i est obtenue. Ainsi, pour chaque commune, il faut calculer tout d'abord le ratio pour pouvoir obtenir l'accessibilité potentielle localisée de celle-ci et de ses communes environnantes.

III. Recherches

1. Le calcul et les données

Dans un premier temps, nous avons effectué de nombreuses recherches pour recueillir des informations sur la méthode de calcul ainsi que sur les données que nous allons utiliser dans la programmation de celle-ci. Il existe un GitLab qui contient le code en R de l'application Shiny et qui a été un socle de travail dans la future implémentation du calcul. Nous avons trouvé de nombreux articles et publications sur le thème de l'APL pour nous permettre de bien comprendre cet indicateur ainsi que son utilité. Cependant, nous avons vite remarqué qu'il était impossible d'avoir accès aux données utiles. En effet, les données sur les consultations des médecins ne sont pas publiques et se trouvent sur le site du SNIIRAM. Une demande pour avoir une vue sur ces bases de données est nécessaire et peut se réaliser sur le Health Data Hub. En moyenne, il faut 362 jours pour avoir une réponse à la demande et elle n'est pas obligatoirement positive. Après une réunion avec le client, il a été décidé de redéfinir de nouveaux objectifs pour la bonne réalisation du projet. Pour ne pas perdre les recherches faites et aider les potentiels repreneurs du projet, nous avons récapitulé les recherches dans un tableau en indiquant pour chaque, le nom du document, son lien, un petit résumé et quelques commentaires sur ce que le document nous a apporté.

Nom du document	Lien	Résumé
L'indicateur d'accessibilité potentielle localisée (APL)	https://drees.solidarites-sante.gouv.fr/sources-outils-et-enguetes/lindicateur-daccessibilite-potentielle-localisee-apl	Explication des ETP contenus dans la méthode de calcul de l'APL. Présence de la pondération de la distance selon l'accessibilité. Données présentes pour les médecins généralistes, les sage-femmes et les kinés. Beaucoup d'articles en annexes plus détaillés
L'accessibilité potentielle localisée (APL)	https://data.drees.solidarites-sante.gouv.fr/explore/dataset/530_l-accessibilite-potentielle-localisee-apl/information/	Rapide résumé de l'APL. Lien pour se rendre sur l'application shiny qui cartographie l'APL en France. Nombreux dossiers archivés et fichier excels avec différentes versions de l'APL.
Shiny	https://drees.shinyapps.io/carto-apl/	Application de visualisation de l'APL
Déserts médicaux : comment les définir ? Comment les mesurer ?	https://drees.solidarites-sante.gouv.fr/publications/les-dossiers-de-la-drees/deserts-medicaux-comment-les-definir-comment-les-mesurer	Rapide résumé sur les déserts médicaux et certains chiffres pour expliquer ce phénomène

Figure 5 : Extrait de notre tableau récapitulatif des recherches

2. Définition de nouveaux objectifs

En absence de donnée, notre tuteur nous a émis l'idée de la génération de donnée. En effet c'est une pratique très courante dans des domaines comme la santé où nous manquons de données dû au caractère très privé de celles-ci. Néanmoins nous étions ici face à une absence complète de données, il était donc difficile de générer des données synthétiques de qualités en ne se basant sur rien. Nous n'avons donc pas creusé cette piste beaucoup plus loin car elle n'était pas nécessaire pour montrer la faisabilité du projet.

Les données sur les professionnels de santé libéraux n'étant pas disponibles, nous avons décidé, en collaboration avec nos clients, de redéfinir les objectifs du projet. L'application portera essentiellement sur les médecins généralistes pour lesquels les données sont disponibles. Elle affichera donc l'indicateur de l'APL pour les villes sur une carte pour garder la partie cartographie. De l'autre côté, nous allons définir une partie sur le calcul qui pourra s'appliquer pour les autres professionnels de santé. Le livrable du projet comprendra

toujours l'application web avec une partie cartographie mais aussi une implémentation qui permettra le calcul de l'APL. Ainsi, les repreneurs du projet pourront insérer les données et pourront afficher celles-ci sur l'application web.

IV. Réalisations

Ainsi concernant l'architecture de la solution finale nous avons décidé de scinder la solution en 2 parties comme peut le montrer le schéma ci-dessous :

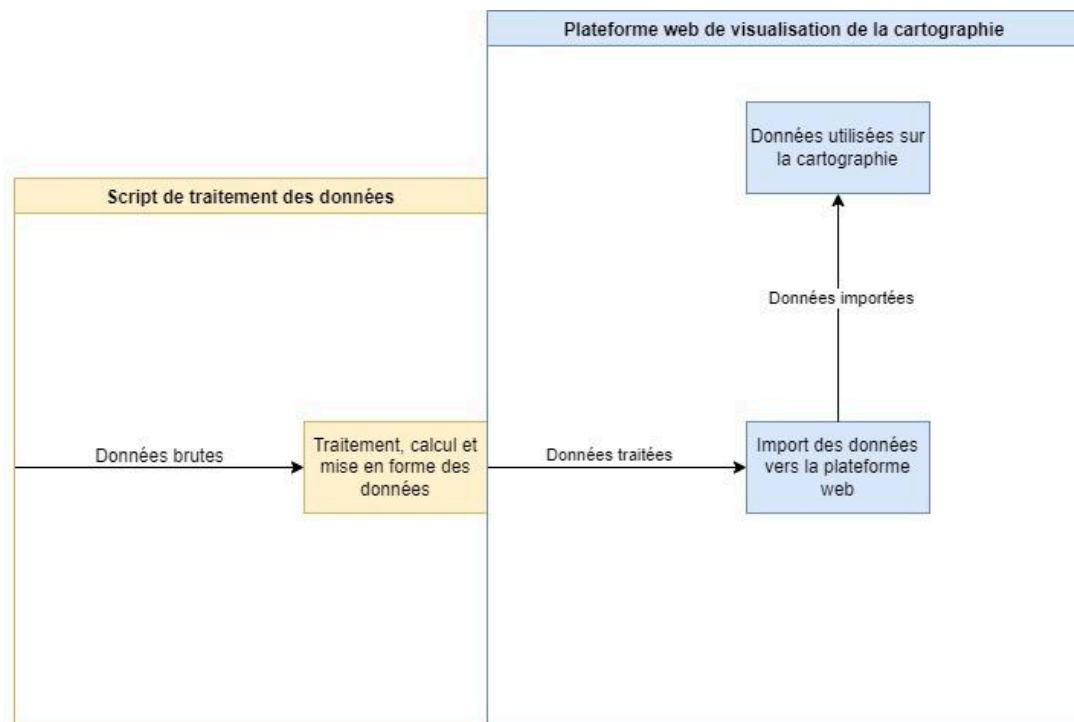


Figure 6 : Schéma fonctionnel de la solution

Ainsi d'un côté en jaune, nous avons un script qui nous permet de traiter des données afin de calculer l'APL et les rendre dans une forme qui permet l'importation vers la plateforme web. Une fois les données récupérées, il suffit ensuite de les importer sur la plateforme web afin de les visualiser sur le fond de carte. Dans la suite de ce rapport nous nous appliquerons à vous décrire les deux parties de cette solution.

1. Partie cartographie

1.1 - Analyse

La partie cartographie se focalise sur la description de l'analyse de la solution web permettant l'affichage des données.

Afin d'être sûr des besoins des clients nous avons réalisé plusieurs versions d'un Diagramme de Cas d'Utilisations de la méthode UML. Après l'avoir retravaillé conjointement, nous avons pu en obtenir une version finale que vous voyez ci-dessous.

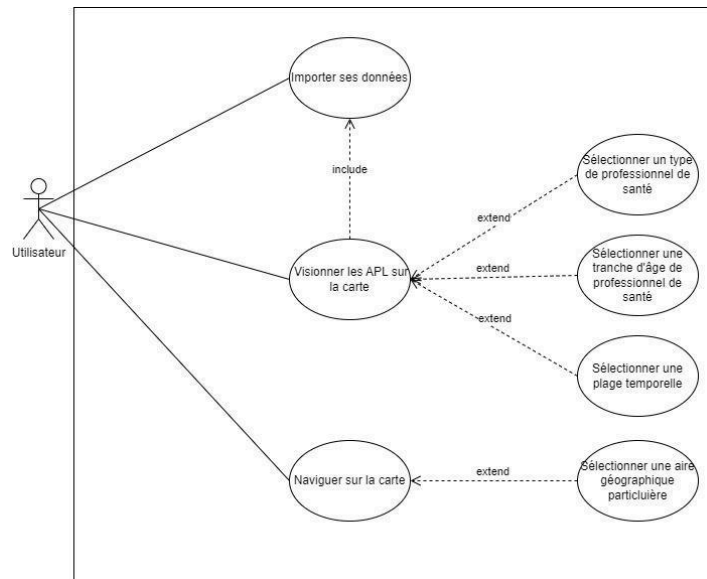


Figure 7 : Diagramme des cas d'utilisation de notre application

Nous avons pensé mettre seulement un utilisateur dans la mesure où nous aimerions développer cette application en SPA (Single Page Application) afin de faciliter son utilisation. Néanmoins, il est tout à fait possible d'imaginer un back-end un peu plus développé avec création de compte, identification et administration, permettant ainsi le déploiement de rôles et de permissions. Cependant, dans l'optique actuelle d'un premier prototype, ce genre de fonctionnalités n'étaient pas nécessaires.

Concernant les cas d'utilisations maintenant nous en comptons 3 principaux et 4 extensions de ceux-ci :

- Importer ses données :

L'utilisateur devra être en capacité d'importer ses données une fois celles-ci traitées et mises en forme par le script de calcul. Ce cas d'utilisation est un pré-requis au cas d'utilisation "Visualiser les APL sur la carte" .

- Visualiser les APL sur la carte :

L'utilisateur devra pouvoir voir s'afficher sur la carte des valeurs d'APL qu'il a importé précédemment (relation d'inclusion).

Ce cas d'utilisation peut être étendu par trois cas d'utilisations supplémentaires :

- Sélectionner un type de professionnel de santé :

Ce cas d'utilisation permet de spécifier un type de professionnel de santé à visualiser. Par défaut, les médecins généralistes sont affichés.

- Sélectionner une tranche d'âge des professionnels de santé :

Ce cas d'utilisation permet de spécifier une tranche d'âge des professionnels de santé afin de pouvoir potentiellement prévoir des tensions au niveau des soins.

- Sélectionner une plage temporelle :

Ce cas d'utilisation permettra à l'utilisateur de spécifier une ou plusieurs années pour la visualisation des données.

- Naviguer sur la carte :

L'utilisateur devra pouvoir se déplacer sur la carte de manière à pouvoir zoomer/dézoomer, et se déplacer latéralement ou verticalement. Ce cas d'utilisation ne nécessite pas d'avoir importé des données au préalable.

Ce cas d'utilisation peut être étendu par un cas d'utilisation supplémentaire :

- Sélectionner une aire géographique particulière :

Ce cas d'utilisation permet à l'utilisateur de spécifier une aire géographique à cibler sur la carte, départements, régions, ou villes.

Voici donc concernant les cas d'utilisations de l'application, de plus nous avons créé un schéma de base données qui pourrait supporter les données de notre application.

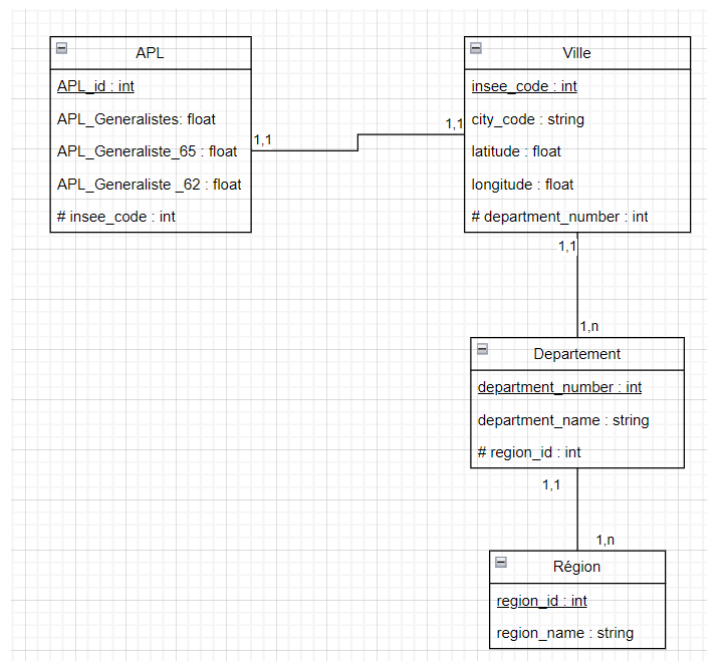


Figure 8 : Modèle Logique de Données de notre base de données

Ce schéma est un Modèle Logique de Données dans lequel nous avons conservé les cardinalités afin de le rendre moins confus. Ainsi vous pouvez voir les clés étrangères présentes dans les tables, elles sont représentées par des attributs possédant un # en début de ligne. Les attributs soulignés sont les clés primaires des tables. Ce schéma représente la forme réelle que prendra la base de données implémentée sous un système de gestion de base de données.

1.2 Conception et développement

1.2.1 Maquettage

Pour être sûrs d'avoir cadré correctement le besoin avec le client, nous avons maqueté la solution afin de présenter une première ébauche de l'apparence de la plateforme de visualisation de l'APL. Pour ce faire nous avons utilisé Balsamiq qui est un outil de création de maquettes de sites web.

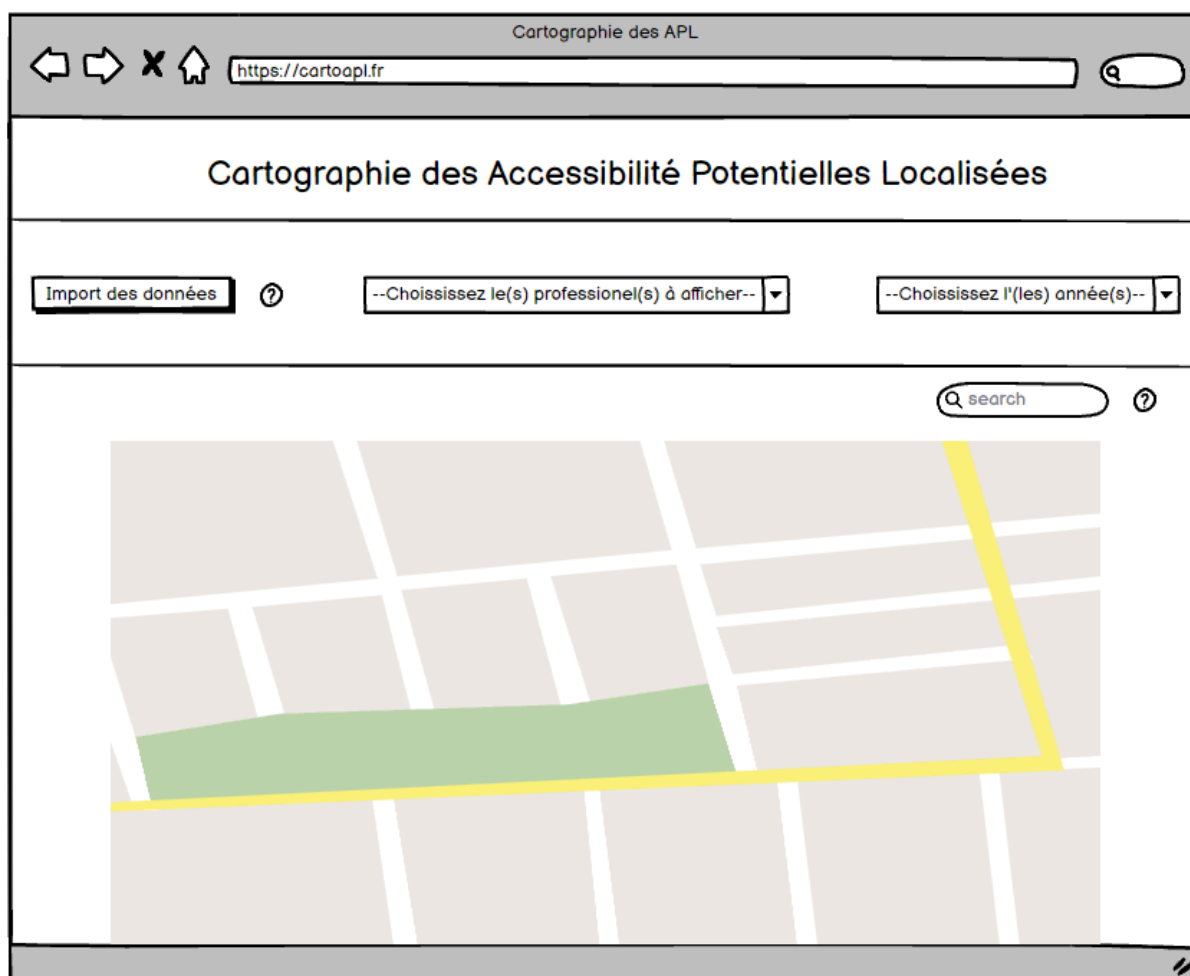


Figure 9 : Maquette de l'application

Sur cette interface on peut voir plusieurs éléments. En haut de la page, son descriptif et en dessous, un bandeau pliable qui abrite 3 éléments. Le premier permet d'importer des données d'APL précédemment traitées avec le script de calcul que vous verrez dans la prochaine partie. Les deux boutons suivants permettent d'afficher des professionnels de santé particuliers ainsi que les années souhaitées. La barre située en dessous permet de rechercher une aire géographique particulière selon la définition des cas d'utilisations.

1.2.2 Choix technologiques

Afin de développer cette plateforme, il nous a fallu faire des choix concernant les technologies à utiliser. Pour commencer le langage, pour des raisons de temps il nous a fallu trouver un framework pour lequel nous aurions peu ou pas besoin d'apprendre, le langage, la syntaxe, la logique. Pour cela nous avons choisi PHP Codeigniter 4, que l'un de nous maîtrisait suffisamment pour développer cette application.

Ensuite le second choix technologique a été porté sur le moyen d'afficher le fond de carte puis les données. En effet énormément de librairies différentes proposent ce genre de services, nous les avons donc comparé selon 3 critères : la rapidité d'installation, la documentation disponible, et finalement l'esthétique de la carte et de ses interfaces.

La librairie qui a répondu à ces attentes a été Leaflet qui est une bibliothèque JavaScript. Elle possède beaucoup de contributeurs ainsi que d'utilisateurs, comme le projet OpenStreetMap par exemple. Voici donc à quoi ressemble la carte Leaflet :

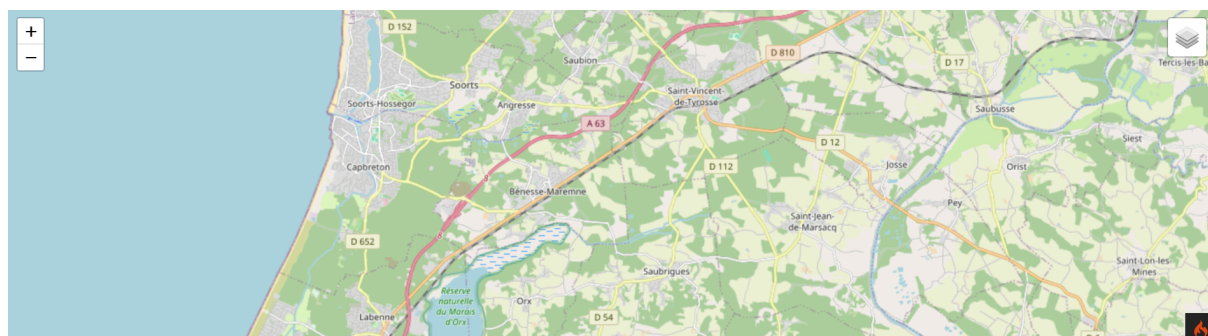


Figure 10 : Fond de carte Leaflet

Les boutons en haut à gauche permettent le zoom et le dézoom, le boutons en haut à droite sert à afficher les différentes visualisations qu'il est possible d'afficher.

1.2.3 Développement de l'application

Dans un premier temps l'objectif était d'avoir une application la plus légère et simple possible, nous avons donc voulu développer la plateforme en partant du principe que la visualisation se ferait en utilisant des fichiers de configurations, un stable avec les villes et d'autres à importer sur la plateforme avec les APL.

Nous avons donc cherché un moyen d'afficher les villes sur la carte. Pour cela nous avons utilisé le portail open.data.gouv pour récupérer au format json, une description normalisée de toutes les villes de France. Nous disposions aussi d'un fichier au format CSV qui recensait, lui, les valeurs des APL dans les différentes villes de France. Pour représenter le traitement des fichiers vers un élément utilisable dans le code nous avons schématisé les contenus de chaque fichiers ainsi que le contenu de ce qui va être utilisé dans le code.

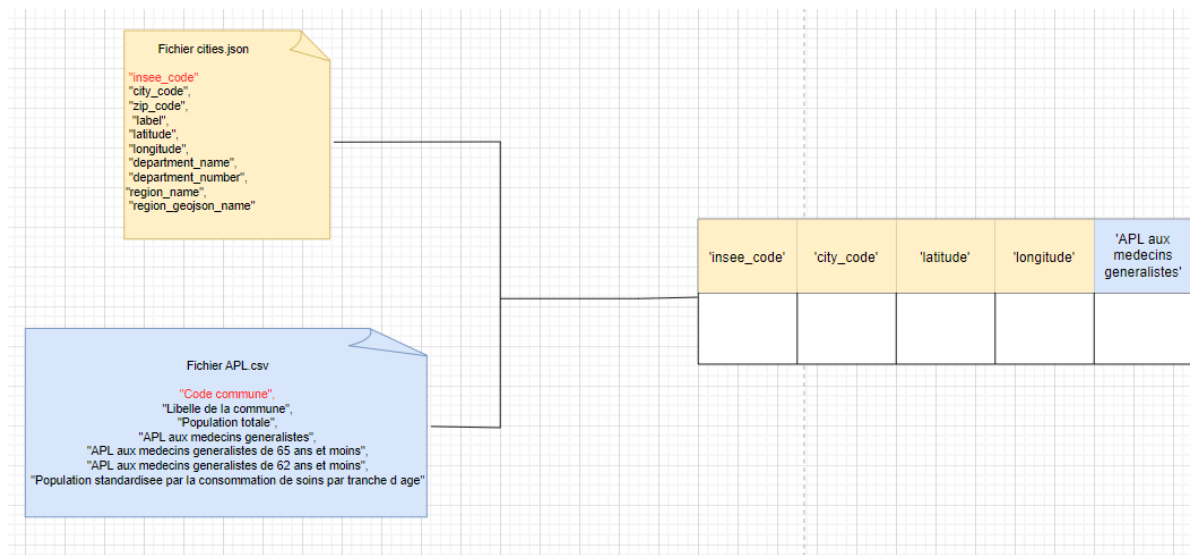


Figure 11 : Schéma de panachage des informations

Sur ce schéma on voit en jaune le fichier des villes et bleu le fichier des APL, le code recherche à partir d'un fichier les informations de l'autre. Ce qui les lie est le 'insee_code' et 'code_commune', ces codes sont identiques et permettent d'identifier l'un à partir de l'autre.

Ensuite dans les informations récupérées vous pouvez voir 'latitude' et 'longitude', qui nous servent littéralement de coordonnées pour placer les villes sur la carte avec un marqueur. Sur ce marqueurs nous retrouvons aussi le 'city_code' qui est le nom de la commune ainsi que l' 'APL aux médecins généralistes'.

Le problème de cette conception est la performance. Le code, pour réussir à créer sa liste de marqueurs à afficher doit lire 2 fichiers extrêmement volumineux, trouver des correspondances, les traiter, les stocker et recommencer. Le plus petit fichier des deux, le CSV, fait 34 936 lignes, le json lui fait environ 469 000 lignes. L'exécution de la fonction s'arrêtait systématiquement à cause du temps de traitement trop long. Une fois cette limite abolie, la première exécution complète à pris 15 minutes.

Après avoir optimisé la lecture des fichiers, la méthode de parcours des fichiers ainsi que l'affichage des marqueurs, le temps d'exécution est tombé à 2min20s, seulement, pour un client web, cela reste élevé.

```

// Clé de correspondance
$cleJSON = 'insee_code';

// Vérifier si la clé existe dans le fichier JSON (
$libelleCommuneCSV = $element['Code commune'];

// Parcourir les villes
foreach ($jsonData['cities'] as $data) {
    $libelleCommuneJSON = $data[$cleJSON];

    if ($libelleCommuneCSV === $libelleCommuneJSON) {
        // Correspondance trouvée, ajouter les propriétés spécifiques au résultat final
        $resultat[] = [
            'insee_code' => $data[$cleJSON],
            'city_code' => $data['city_code'],
            'latitude' => $data['latitude'],
            'longitude' => $data['longitude'],
            'department_name' => $data['department_name'],
            'APL_aux_medecins_generalistes' => $element['APL aux medecins generalistes'],
        ];
        break; // Sortir de la boucle une fois qu'une correspondance a été trouvée
    }
}

```

Figure 12 : Extrait de code de la méthode d'import de fichiers.

Vous pouvez voir sur la capture d'écran que le cœur de la fusion des fichiers n'a rien de très compliqué ni de très lourd. Nous définissons la clé de correspondance ensuite nous récupérons le code_commune de la commune du fichier CSV ensuite nous parcourons le fichier json et lorsque les codes communes sont égaux, nous pouvons récupérer nos informations, les stocker et passer à la ville suivante.

Finalement nous avons modélisé une base de données que vous avez pu voir précédemment afin de finir d'améliorer les performances de l'application. Celle-ci n'a pas encore été intégrée mais serait intéressante à creuser en approfondissement.

2. Partie calcul

2.1 Analyse

Pour ce qui est de la partie calcul, afin de le préparer malgré l'absence de données, nous nous sommes appuyés sur le GitLab existant.

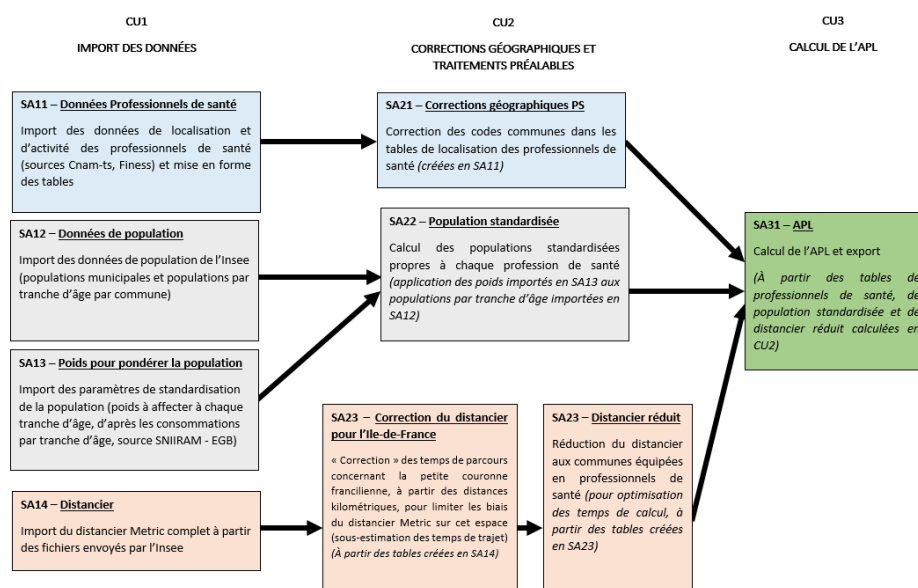


Figure 13 : Extrait du GitLab sur l'architecture du calcul

Ainsi, nous sommes tombés d'accord pour réaliser 4 fonctions pour le calcul de l'APL :

- Le calcul des ETP à partir des données des médecins par commune et des données des consultations (données pour le moment inaccessibles du SNIIRAM). Le résultat de cette fonction est les ETP pour chaque ville et sera présenté dans une table d'une base de données dans le futur du projet et dans un document Excel pour le moment.
- Le calcul de la population standardisée par âge à partir population par âge (données de l'INSEE) par commune et la pondération de l'âge (données disponibles sur les poids de pondération). Les valeurs, de la population standardisée par âge de chaque commune, obtenues seront aussi stockées dans la base de données à l'avenir.
- Le calcul du ratio à partir des tables obtenues avec les deux premières fonctions et en utilisant la pondération relative à la distance (tableau présenté en introduction). Nous obtenons les ratios pour chaque commune présentés une nouvelle fois dans une table.

- Le calcul final de l'APL par commune à partir du résultat de la fonction du Ratio et de la pondération relative à la distance. L'indicateur, pour chaque ville, final obtenu sera contenu dans une table et dans le futur, ces données pourront être récupérées par l'application web et affichées sur la carte.

Pour résumer, le but de ce programme est de récupérer les résultats de l'APL pour chaque commune en France dans une table. Celle-ci sera appelée dans la partie cartographie du code pour pouvoir afficher toutes les données calculées.

D'autre part pour la partie import des données nous nous sommes dans un premier temps inspirés du gitlab de l'APL actuel.

```
import_lib <- function() {
  if (nb_fichiers == 1) {
    if (prof == "sf" & mil_prof == "2016") {
      lib <- readxl::read_excel(paste0(data_act_cnam, "sagesfemmes_2016.xlsx")) %>%
        data.table()
    } else if (prof == "inf") {
      lib <- paste0(data_act_cnam, "apl_etp_idel_", mil_prof, "_12", substr(mil_prof, 3, 4), ".sas7bdat") %>%
        sas7bdat::read.sas7bdat() %>%
        data.table()
    } else {
      lib <- paste0(data_act_cnam, "apl_etp_", prof, "_l_", mil_prof, "_12", substr(mil_prof, 3, 4), ".sas7bda
        sas7bdat::read.sas7bdat() %>%
        data.table()
    }
  }
  colnames(lib) <- tolower(colnames(lib))
  if (mil_prof == "2016") {
    if (prof != "mg") {
      setnames(lib, old = "commune", new = "libcom")
    }
    if (prof %in% c("mk", "inf")) {
      setnames(lib, old = "num_pfs_ano", new = "num_cab_ano")
      setnames(lib, old = "ipp_ano", new = "num_ipp_ano")
      setnames(lib, old = "cab_princ", new = "cab")
    }
  }
}
```

Figure 14 : Extrait du code GitLab permettant d'importer les données

Nous pouvons observer qu'un import de tableau excel est effectué afin de traiter les données. Néanmoins, n'ayant pas accès aux fichiers de données présents en entrée nous ne pouvions pas comprendre sur quoi agissait ce code.

Néanmoins nous connaissons les données dont disposaient les créateurs de l'APL. En voici la liste :

- Nombre de consultations et commune d'activité de chaque médecin en France (demande auprès SNIIRAM)
- Population par commune et par tranche d'âge (demande auprès de l'INSEE)
- Toutes les communes à moins de 20 minutes de chaque commune et la durée du trajet (demande auprès de Metric)

Ainsi nous avons imaginé des tableaux excel idéaux à l'entrée de nos fonctions afin de réaliser un calcul de l'APL.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Commune J	Tranche d'âge	Pondération	Population	Population pondérée					Commune J	Population pondérée		
2	Commune 1	Moins de 5 ans	1,24	12	14,88					Commune 1	1 265,10		
3	Commune 1	de 5 à 9 ans	0,74	24	17,76					Commune 2	1 257,76		
4	Commune 1	de 10 à 14 ans	0,62	36	22,32								
5	Commune 1	de 15 à 19 ans	0,65	59	38,35								
6	Commune 1	de 20 à 24 ans	0,68	70	47,60								
7	Commune 1	de 25 à 29 ans	0,71	80	56,80								
8	Commune 1	de 30 à 34 ans	0,76	52	39,52								
9	Commune 1	de 35 à 39 ans	0,81	48	38,88								
10	Commune 1	de 40 à 44 ans	0,86	64	55,04								
11	Commune 1	de 45 à 49 ans	0,93	74	68,82								
12	Commune 1	de 50 à 54 ans	1,03	90	92,70								
13	Commune 1	de 55 à 59 ans	1,12	85	95,20								
14	Commune 1	de 60 à 64 ans	1,17	56	65,52								
15	Commune 1	de 65 à 69 ans	1,25	63	78,75								
16	Commune 1	de 70 à 74 ans	1,40	75	105,00								
17	Commune 1	de 75 à 79 ans	1,57	76	119,32								
18	Commune 1	de 80 à 84 ans	1,76	69	121,44								
19	Commune 1	de 85 à 89 ans	1,88	57	107,16								
20	Commune 1	Plus de 90 ans	1,74	46	80,04								
21	Commune 2	Moins de 5 ans	1,24	12	14,88								
22	Commune 2	de 5 à 9 ans	0,74	24	17,76								
23	Commune 2	de 10 à 14 ans	0,62	36	22,32								
24	Commune 2	de 15 à 19 ans	0,65	59	38,35								
25	Commune 2	de 20 à 24 ans	0,68	55	37,40								
26	Commune 2	de 25 à 29 ans	0,71	57	40,47								
27	Commune 2	de 30 à 34 ans	0,76	44	33,44								
28	Commune 2	de 35 à 39 ans	0,81	79	63,99								
29	Commune 2	de 40 à 44 ans	0,86	72	61,92								
30	Commune 2	de 45 à 49 ans	0,93	88	81,84								
31	Commune 2	de 50 à 54 ans	1,03	72	74,16								
32	Commune 2	de 55 à 59 ans	1,12	58	64,96								

Figure 15 : Exemple de tableau idéal pour le calcul de la pondération

Ces tableaux sont donc une représentation théorique des données.

2.2 Développement

L'APL actuelle est codée en R et il aurait été pertinent de réutiliser ce même code si nous avions accès aux mêmes données. Mais ce code ne pouvant réellement être réutilisé, nous avons de ce fait, opté pour un langage que nous maîtrisons mieux, Python.

On a en entrée les données d activité et de localisation des professionnels de santé
+ Code
+ Markdown

```

import pandas as pd

df = pd.read_csv("tableau_SNIIRAM.csv")

```

Python

On repère la médiane, tous les professionnels de santé au dessus on leur impute ETP = 1; de 25% à la médiane = 0,7; 10 à 25 % = 0,5 . 5 à 10 % = 0,2; en dessous de 5 = 0

```

# Calcul de la médiane des nombres de consultations
median_consultations = df['Nombre de consultations'].median()

# Appliquer les règles pour attribuer les ETP en fonction des pourcentages
df['ETP'] = 0.0 # Initialise la colonne ETP à 0

df.loc[df['Nombre de consultations'] > median_consultations, 'ETP'] = 1.0
df.loc[(df['Nombre de consultations'] <= median_consultations) & (df['Nombre de consultations'] > df['Nombre de consultations'].quantile(0.75)) & (df['Nombre de consultations'] > df['Nombre de consultations'].quantile(0.5)), 'ETP'] = 0.7
df.loc[(df['Nombre de consultations'] <= df['Nombre de consultations'].quantile(0.5)) & (df['Nombre de consultations'] > df['Nombre de consultations'].quantile(0.25)), 'ETP'] = 0.5
df.loc[(df['Nombre de consultations'] <= df['Nombre de consultations'].quantile(0.25)), 'ETP'] = 0.2
df.loc[(df['Nombre de consultations'] <= df['Nombre de consultations'].quantile(0.05)), 'ETP'] = 0.0

```

Python

Figure 16 : Extrait du code Python du calcul de la valeur ETP

L'objectif étant de poursuivre le projet par la suite nous nous sommes donnés pour mission de faire le code le plus expliqué possible pour que la reprise se passe le plus facilement possible.

Nous avons donc pour chaque fonction établi un code précis permettant d'amener au calcul de l'APL dans l'attente de l'obtention des données.

V. Gestion de Projet

Nous avons décidé de nous appuyer sur la méthode agile pour la réalisation du projet, seulement, ce n'était pas complètement du SCRUM puisque nous n'avons pas trouvé pertinent d'utiliser celle-ci dans un groupe d'uniquement trois personnes. Ainsi, sans forcément réaliser de sprint, nous nous sommes fixés des objectifs entre chaque réunion pour assurer la bonne réalisation du projet. En effet, pour assurer la collaboration entre notre équipe et le client, nous avons réalisé des réunions bimensuelles. Avant chaque point, nous écrivions une fiche de synthèse sur ce que nous avons fait depuis la dernière. Pendant celles-ci, nous faisons un compte rendu pour nous permettre de savoir ce qu'il fallait faire pour la prochaine fois. Les réunions nous ont permis de se rendre compte des problèmes liés à l'inaccessibilité aux données de santé et de redéfinir le besoin ainsi que le livrable final. Pour stocker et partager les fichiers, que ce soient les recherches, les comptes rendus ou les fiches de synthèse, nous avons utilisé l'outil Google Drive. Pour les réunions et le calendrier, nous nous sommes aidés de Microsoft Teams. Le partage de code a été assuré par l'outil GitLab pour nous permettre de développer l'application en parallèle sans avoir de problème.

Initialement, nous avons prévu une certaine organisation pendant l'ensemble du projet disponible dans le diagramme de Gantt suivant :

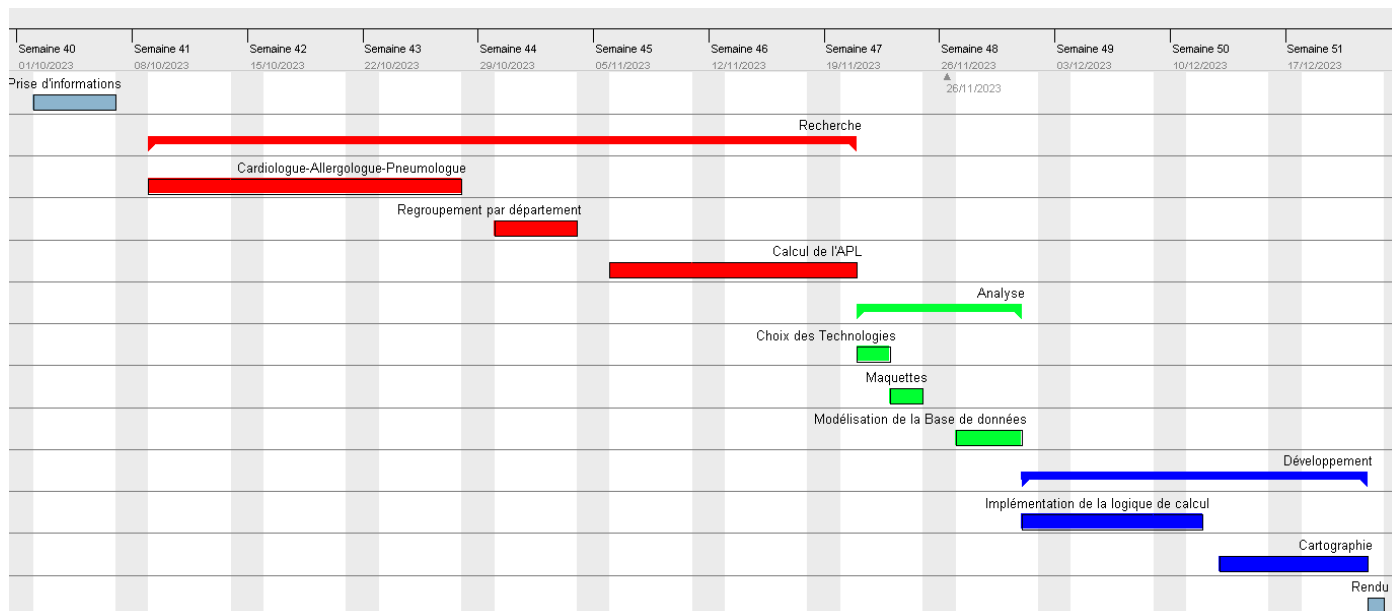


Figure 17 : Diagramme de Gantt initial de notre projet

Dans ce diagramme, il existe trois grandes parties, une portant sur les recherches, une autre sur l'analyse et la dernière contient toute la partie implémentation. Nous avons décidé de commencer tous les trois sur la partie recherche puisque cette partie était la plus importante pour bien comprendre le sujet dont la méthode de calcul et l'accès aux données. Nous nous concentrons sur les cardiologues, les allergologues et pneumologues pour ce qui est des

autres professionnels de santé à étudier. Cependant, après plusieurs recherches, les données n'étant pas disponibles, nous avons pris de la hauteur et nous nous sommes concentrés sur l'ensemble des médecins pour avoir le plus de données possibles. Finalement, la partie recherche s'est principalement concentrée sur la partie calcul de l'APL et la recherche de solutions pour l'accès aux données. La partie analyse s'est déroulée comme prévu et nous avons pu structurer notre solution avec les choix des technologies, les maquettes. La modélisation de la base de données s'est étendue légèrement sur la partie développement pour qu'elle soit bien complète. Nous avons réalisé quelques changements pour la partie développement puisqu'il était plus pertinent pour nous d'implémenter les deux parties en parallèle plutôt qu'une après l'autre.

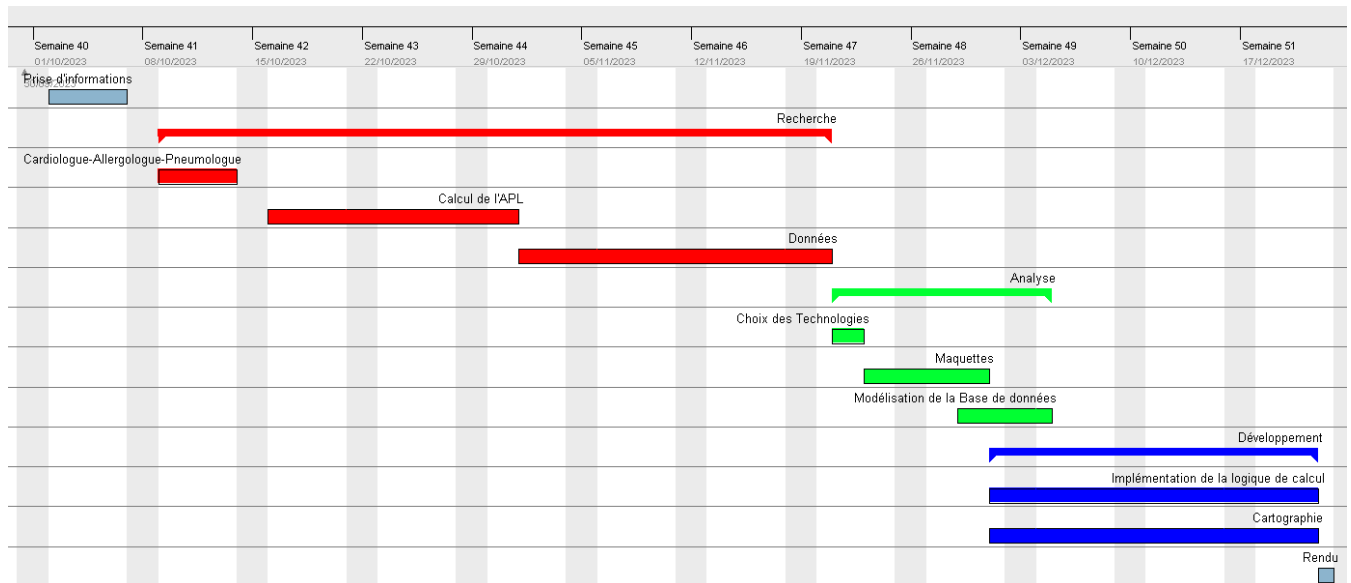


Figure 18 : Diagramme de Gantt final de notre projet

Pour la répartition dans le projet, les recherches et l'analyse ont été réalisées par l'ensemble de l'équipe du projet. Pour la partie implémentation, Paul s'est occupé de la partie cartographie, Léo de la majorité du calcul, Baptiste aidait sur la partie calculatoire et clôturait la partie recherche. Le rapport ainsi que la soutenance ont été faits par nous trois. Pour conclure notre gestion de projet, nous avons été vraiment solidaires et chacun a fait de tout pour aider le groupe à obtenir un livrable final correspondant aux demandes du client.

VI. Conclusions et Perspectives

1. Les limites

Lors de ce projet, nous avons été confrontés à certains problèmes qui nous ont empêchés de répondre à l'objectif initial. Notamment, le fait de ne pas avoir accès aux données du SNIIRAM nous a ralenti. Nous avons passé beaucoup de temps sur l'obtention de celles-ci puis sur les documents demandant leurs accès. Ensuite, nous avons décidé de ne pas continuer puisque le délai était d'environ un an.

D'un autre côté, initialement, la partie cartographie venait chercher des données dans des fichiers stockés de manière locale. Nous nous sommes rapidement heurtés à certains problèmes causés directement par l'énorme quantité de données à traiter. En effet, le traitement de la cartographie pouvait s'interrompre et quand cela fonctionnait, cela prenait un temps anormalement élevé. Ainsi, il nous est apparu logique de mettre en place la base de données modélisée durant l'analyse.

2. Devenir du projet

Pour conclure, la solution actuelle ne répond pas aux objectifs initiaux, présentant seulement les médecins généralistes libéraux et pas l'ensemble des professionnels de santé. En outre, l'application web répond aux besoins de la redéfinition du livrable. En effet, les deux parties sont disponibles, la partie cartographie permet d'observer les valeurs de l'APL pour les communes françaises. La partie calcul contient les différentes étapes de la méthode et idéalement, il faudrait juste utiliser la base de données pour avoir le bon indicateur pour chaque commune. La base de données étant déjà modélisée, il faudra juste la compléter avec les informations du SNIIRAM, de l'INSEE et les distances entre les communes. Concernant les données liées à la distance, les créateurs de l'APL actuel ont fait appel à Metric. Cependant il s'agit de distance directe d'un point A à un point B et ne prend pas en compte les trajets réels. Il serait peut être possible d'ajouter une nouvelle couche de précision à l'APL actuel en prenant en main l'API de Google Maps. Ensuite, il suffira de lier les parties en indiquant cela dans la cartographie pour afficher les APL pour l'ensemble des professionnels de santé.

La suite du projet va commencer par une prise en main de notre solution pour comprendre comment elle s'articule avec les deux parties. En parallèle, il serait intéressant de faire des demandes d'accès pour les données tout en se renseignant sur la méthode de calcul pour bien la comprendre. Lorsque les données seront disponibles, il faudra mettre à jour la base de données mais aussi potentiellement l'implémentation selon la structure de ces données. Il serait intéressant de faire une partie vérification de la qualité de l'APL pour s'assurer de la fonctionnalité de l'application web. Après avoir fait cela, il restera seulement à réaliser de manière récurrente des mises à jour sur les données.

L'autre étape importante va être d'étendre l'APL à l'ensemble du système de soins défini par l'ICube.

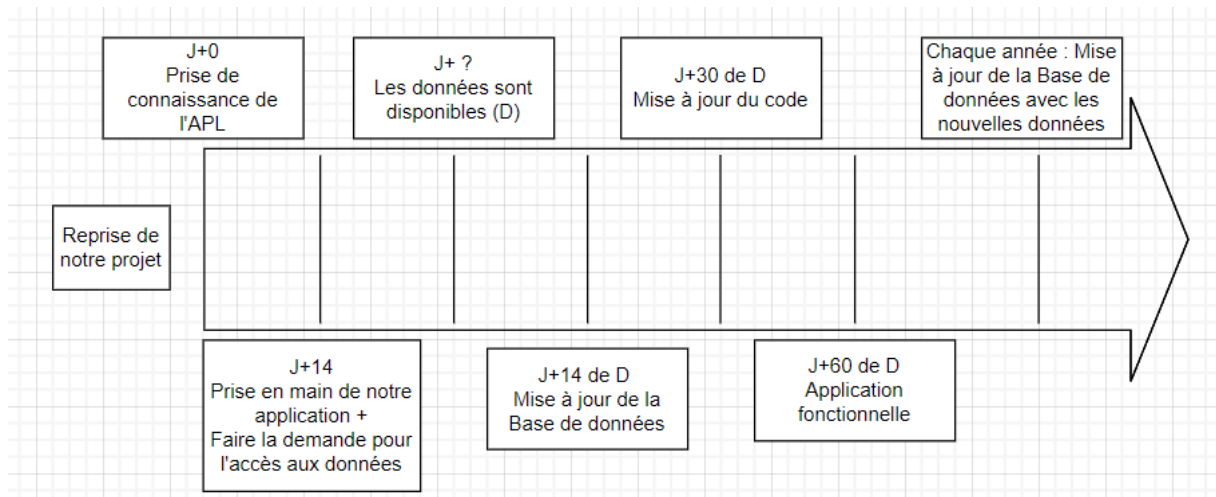


Figure 19: Proposition de projection pour la suite du projet

Bibliographie

Application Shiny : <https://drees.shinyapps.io/carto-apl/>

Documents sur l'APL :

<https://drees.solidarites-sante.gouv.fr/sources-outils-et-enquetes/lindicateur-daccessibilite-potentielle-localisee-apl>

https://data.drees.solidarites-sante.gouv.fr/explore/dataset/530_l-accessibilite-potentielle-localisee-apl/information/

<https://drees.solidarites-sante.gouv.fr/publications/documents-de-travail-1998-2019/laccessibilite-potentielle-localisee-apl-une-nouvelle>

<https://drees.solidarites-sante.gouv.fr/sites/default/files/2020-10/dter124.pdf>

GitLab de l'APL : https://gitlab.com/DREES_code/public/outils/production-apl

Annexes

Annexe 1 : Suite fonction calcul ETP

Rajoute colonne Valeur ETP pour chaque médecin

```
# Enregistrez le DataFrame avec la nouvelle colonne ETP
df.to_csv("tableau_SNIIRAM_modifie.csv", index=False)
```

Python

Ensuite on fait la somme ETP des médecins pour chaque commune

```
# Charger le tableau
df = pd.read_csv("tableau_SNIIRAM_modifie.csv")

# Grouper par commune et sommer les ETP
result = df.groupby('Commune')['ETP'].sum().reset_index()
```

Python

On extrait la valeur en ETP de chaque commune

```
# Enregistrer le résultat dans un nouveau fichier CSV
result.to_csv("etp_par_commune.csv", index=False)
```

Python

Annexe 2 : Fonction calcul population pondérée

On a en entrée le nombre d'habitant par commune, par tranche d'âge et sa pondération,

```
import pandas as pd

# Chargez le fichier CSV dans un DataFrame
df = pd.read_csv('PopulationStrdr.csv')
```

Python

On multiplie la population et la pondération pour chaque ligne

```
# Effectuez la multiplication et ajoutez une nouvelle colonne "Population pondérée"
df['Population pondérée'] = df['Population'] * df['Pondération']
```

Python

On fait la somme de la population pondéré pour chaque commune

```
# Groupez par commune et effectuez la somme des populations pondérées
resultats = df.groupby('Commune')['Population pondérée'].sum().reset_index()

# Affichez les résultats
print(resultats)
```

Python

On exporte le tableau sous format csv

```
# Enregistrez les résultats dans un nouveau fichier CSV
resultats.to_csv('PopulationSortie.csv', index=False)
```

Python

Annexe 3 : Calcul du ratio

```
# Ajouter la colonne "ETP" de df2 à df3 avec la condition "Commune" = "Commune J"
df3 = pd.merge(df3, df2[['Commune', 'ETP']], left_on='Commune J', right_on='Commune', how='left')

# Renommer la colonne ajoutée
df3 = df3.rename(columns={'ETP': 'ETP_J'})
```

Python

On définit à présent le coefficient de décroissement en fonction du temps de trajet.

[+ Code](#) [+ Markdown](#)

```
#Calcul du décroissement de distance
def calcul_decroissement(row):
    trajet = row['Trajet']
    if trajet < 10:
        return row['Population_J'] * 1
    elif 10 <= trajet < 15:
        return row['Population_J'] * (2 / 3)
    elif 15 <= trajet <= 20:
        return row['Population_J'] * (2 / 3)
    else:
        return None

# Appliquer la fonction pour créer la colonne "Décroissement"
df3['Décroissement'] = df3.apply(calcul_decroissement, axis=1)
```

Python

On importe d'abord le tableau avec toutes les communes J de France et les communes I à moins de 20 minutes ainsi que la durée de trajet. On importe le tableau exporter depuis CalculETP et même chose avec le tableau de CalculPop.

```
import pandas as pd

# Charger les trois tableaux Excel dans des DataFrames pandas
df1 = pd.read_excel('CommunesMetrics.xlsx')
df2 = pd.read_excel('chemin_vers_tableau2.xlsx')
df3 = pd.read_excel('chemin_vers_tableau3.xlsx')
```

Python

On fusionne ces tableaux

```
import pandas as pd

# Charger les trois tableaux Excel dans des DataFrames pandas
df1 = pd.read_excel('CommunesMetrics.xlsx')
df2 = pd.read_excel('etp_par_commune.csv')
df3 = pd.read_excel('PopulationSortie.csv')

# Ajouter la colonne "Population" de df1 à df3 avec la condition "Commune" = "Commune I"
df3 = pd.merge(df3, df1[['Commune', 'Population']], left_on='Commune I', right_on='Commune', how='left')

# Renommer la colonne ajoutée
df3 = df3.rename(columns={'Population': 'Population I'})
```

On calcule à présent le ratio de chaque commune J

```
#Calcul du ratio
# Groupez les données par Commune J
grouped_df = df.groupby('Commune J')

# Initialisez une liste pour stocker les résultats
resultats = []

# Parcourez chaque groupe
for commune, group in grouped_df:
    # Calculez l'ETP de la commune actuelle
    etp_commune = group['ETP_J'].iloc[0]

    # Calculez la somme des Décroissements pour cette commune
    décroissement_sum = group['Décroissement'].sum()

    # Effectuez le calcul final
    ratio = etp_commune / décroissement_sum

    # Ajoutez le résultat à la liste
    resultats.append({'Commune J': commune, 'Ratio J': ratio})

# Créez un DataFrame avec les résultats
resultats_df = pd.DataFrame(resultats)

# Fusionnez resultats_df avec le DataFrame d'origine df
df = pd.merge(df, resultats_df, on='Commune J', how='left')
```

Annexe 4 : Calcul APL

On calcul à présent l'APL pour chaque commune J

```
#Calcul APL pour chaque commune I

# Groupez les données par Commune I et calculez la somme des produits des ratios par Décroissement
result_sum = df.groupby('Commune I').apply(lambda x: (x['Ratio J'] * x['Décroissement']).sum()).reset_index()

# Renommez les colonnes
result_sum.columns = ['Commune I', 'APL']

# Affichez le résultat
print(result_sum)
```

Python

On exporte ensuite en CSV pour être utiliser par l'appli web

```
# Exporter le DataFrame result_sum vers un fichier CSV
result_sum.to_csv('result_sum.csv', index=False)
```

Python