



UFR DES SCIENCES

Master Mathématiques Appliquées, Statistiques

Parcours Statistiques Appliquées et Analyse Décisionnelle

Analyse des données géospatiales et cartographie

Réalisé par

ASMAA AGAYER

asmaa.agayer@etu.unicaen.fr

HAJAR BADRAOUI

hajar.badraoui@etu.unicaen.fr

NICODÈME KONÉ

21512368@etu.unicaen.fr

Encadré par

MR. CHRISTOPHE CHESNEAU

07 mars 2025

REMERCIEMENTS

Nous tenons à exprimer notre reconnaissance à toutes les personnes qui ont rendu possible la réalisation de ce projet, ainsi qu'à notre encadrant, Monsieur **Christophe Chesneau**, pour ses précieuses suggestions et son assistance technique. Son aide et sa collaboration nous ont été d'une grande aide pour la compréhension et la mise en œuvre des applications du projet.

Nous exprimons également notre gratitude envers Monsieur **Bertrand Maillot**, notre responsable de formation, pour son suivi attentif et son engagement sans relâche en faveur de notre apprentissage à l'Université de Caen Normandie.

Nos sincères remerciements vont aussi à l'équipe pédagogique de l'Université de Caen Normandie et à l'ensemble du corps professoral, dont l'enseignement de qualité et les ressources mises à notre disposition ont été déterminants pour la réussite de ce projet. Leur soutien et leur disponibilité ont été indispensables et inestimables.

Enfin, nous adressons nos sincères remerciements au jury d'avoir bien voulu évaluer notre travail. Leur expertise et leur regard critique nous permettront d'enrichir encore davantage cette réflexion et d'améliorer nos compétences.

RÉSUMÉ

L'accès aux soins ne devrait pas dépendre d'un code postal, pourtant, en Normandie comme ailleurs, des disparités territoriales persistent. Derrière ces inégalités se cachent des kilomètres de routes, des trajets plus longs et des infrastructures médicales inégalement réparties. **Peut-on réellement quantifier et visualiser ces écarts ?**

En combinant les données géospatiales, les analyses statistiques et la cartographie interactive, cette étude propose une approche innovante pour comprendre l'accessibilité aux établissements de santé. Avec R et Python, nous avons structuré les informations, exploité les sources disponibles et sélectionné les outils les plus adaptés pour analyser l'impact du réseau routier et la répartition des infrastructures médicales.

L'application de cette méthodologie à la Normandie met en évidence les zones les plus vulnérables, où l'accès aux soins peut dépasser 30 minutes de trajet réel. Contrairement aux simples cartes statiques, la modélisation des temps de déplacement permet d'identifier les points d'amélioration possibles : renforcement du maillage hospitalier, optimisation du réseau routier ou encore recours à la télémédecine.

Au-delà du constat, cette étude apporte des clés de lecture essentielles pour l'aménagement du territoire et l'élaboration de politiques de santé plus inclusives, afin que chaque habitant, quel que soit son lieu de résidence, puisse accéder aux soins dans des conditions équitables.

Table des matières

1	Données géospatiales	3
1.1	Définition des données géospatiales	3
1.2	Sources des données géospatiales	4
1.3	Enjeux des données géospatiales	6
2	Analyse géospatiale et cartographie	8
2.1	Types d'analyse géospatiale	8
2.2	R pour pour l'analyse et la cartographie géospatiales	9
2.2.1	Formats de données géospatiales dans R	9
2.2.2	Structures de données géospatiales dans R	9
2.2.3	Packages R pour l'analyse et la cartographie géospatiales	10
2.2.4	Manipulation et gestion des données géospatiales avec R	11
2.3	Analyse géospatiale avec Python	16
2.3.1	Packages Python pour l'analyse et la cartographie géospatiales	17
2.3.2	Manipulation et gestion des données géospatiales avec Python	18
3	Application : Analyse de l'accessibilité des établissements de santé en zone rurale en Normandie	23
3.1	Sources de données	23
3.2	Méthodologie	24
3.2.1	Collecte de données	24
3.2.2	Nettoyage des données	28
3.2.3	Analyse géospatiale de données avec Python	32
3.2.4	Analyse géospatiale de données avec R	45
3.3	Interprétation des sorties	49
3.3.1	Disparités spatiales dans l'accessibilité aux soins	49
3.3.2	Influence du réseau routier sur l'accessibilité aux soins	49
3.3.3	Temps de trajet réel et accessibilité médicale	49
3.3.4	Cartographie et identification des zones vulnérables	50
3.4	Comparaison entre Python et R	51

Table des figures

1.1	Exemple de données vectorielles	3
1.2	Exemple de données matricielles (raster)	4
1.3	Interface de logiciel SIG	5
1.4	Représentation des données par SIG	6
2.1	Limites administratives de France	12
2.2	Centroïde de la région Normandie	14
2.3	Zone tampon autour de la région Normandie	15
2.4	Carte choroplèthe de Normandie	16
2.5	Carte des arrondissements de Paris	19
2.6	Carte des IRIS de Paris	20
2.7	Carte des Arrondissements de Paris	21
2.8	Carte de répartition des périmètres des arrondissements de Paris	22
3.1	Limites administratives de Normandie	33
3.2	Établissements de santé et position par rapport à la Normandie	36
3.3	Longueur des routes principales en Normandie	39
3.4	Carte interactive des routes principales et des établissements de santé en Normandie (Caen).	41
3.5	Répartition des vitesses maximales sur les routes principales en Normandie	45
3.6	Diagramme en boîte illustrant la variabilité des distances aux routes principales en fonction du type d'établissement de santé	47
3.7	Indice de Moran	48

INTRODUCTION

L'analyse des données géospatiales et la conception de cartes sont devenues des outils essentiels pour comprendre et gérer les phénomènes liés à l'espace terrestre. Ces données, qui permettent de représenter des informations localisées sur la surface de la Terre, sont utilisées dans de nombreux domaines : urbanisme, environnement, gestion des ressources naturelles, transports, et bien d'autres.

La conception de cartes joue ici un rôle central, en traduisant des données complexes en visualisations claires, compréhensibles et exploitables. Ces cartes ne se limitent pas à une simple représentation : elles constituent également des outils puissants d'aide à la décision, que ce soit pour anticiper des risques, optimiser des réseaux ou organiser des interventions en situation d'urgence. Grâce aux avancées technologiques et au développement d'outils numériques, notamment les systèmes d'information géographique (SIG), il est désormais possible de croiser des bases de données géospatiales volumineuses avec des algorithmes performants. Ces outils permettent de produire des analyses détaillées et des cartes interactives adaptées à des problématiques globales comme l'aménagement du territoire, le suivi environnemental ou encore la gestion des crises.

Dans le cadre de ce projet, nous utiliserons les outils R et Python pour traiter et analyser des données géospatiales, ainsi que pour la création des cartes associées. Ces deux langages, largement reconnus pour leur puissance et efficacité dans le traitement et dans la visualisation des données, offrent une flexibilité et des bibliothèques spécialisées qui permettent de répondre efficacement à des problèmes complexes. Ainsi, ce projet vise à démontrer comment ces technologies peuvent être utilisées pour résoudre des enjeux pratiques en exploitant pleinement le potentiel des données géospatiales et des techniques de cartographie.

Dans cette étude, nous avons choisi d'analyser l'accessibilité des établissements de santé en zone rurale en Normandie. L'accès aux soins est un défi majeur, en particulier dans les régions où la densité de population est faible et les infrastructures médicales plus dispersées. Cette problématique soulève plusieurs questions : **Quelles sont les zones les plus éloignées des établissements de santé ? Existe-t-il des disparités dans l'accès aux soins en fonction du réseau routier ?** À travers l'analyse géospatiale et la cartographie, nous chercherons à identifier les communes les plus vulnérables en matière d'accessibilité aux services médicaux, afin de proposer des pistes d'amélioration et d'optimisation des infrastructures existantes.

Chapitre 1

Données géospatiales

1.1 Définition des données géospatiales

Les données géospatiales sont des données numériques qui représentent de l'information explicite de positionnement géographique. Elles comprennent trois types d'informations :

- **Des données géométriques** : position géographique et forme des objets.
- **Des attributs** : caractéristiques des objets.
- **Des métadonnées** : pour définir ou décrire le jeu de données.

Il existe deux types de données : les **données matricielles** et les **données vectorielles**.

1. **Données vectorielles** : sont basées sur des formes géométriques pour représenter les caractéristiques de la surface de la Terre. Elles se décomposent en trois éléments principaux :
 - **Points** : Ils représentent des emplacements spécifiques et singuliers sur la Terre. Les exemples incluent l'emplacement d'un point de repère, tel qu'un monument ou un puits d'eau.
 - **Lignes** : Les caractéristiques linéaires telles que les rivières, les routes ou les pipelines sont représentées à l'aide de lignes. Chaque ligne est essentiellement une série de points interconnectés.
 - **Polygones** : Les zones ou régions, comme les lacs, les forêts ou les limites urbaines, sont représentées par des polygones. Ces formes fermées encapsulent un espace particulier et détaillent souvent ses attributs, comme la taille de la zone ou le type de sol.

2. **Données matricielles (rasters)** : sont un format de données basé sur une grille qui

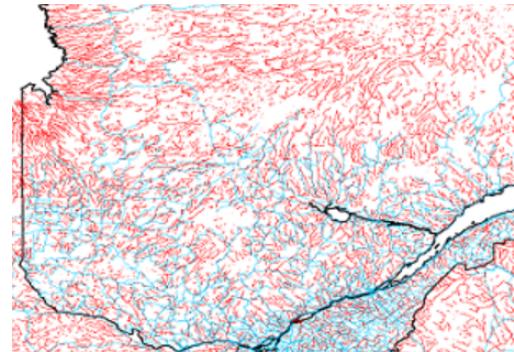


FIGURE 1.1 – Exemple de données vectorielles

capture des phénomènes continus, où chaque pixel ou cellule a une valeur représentant des informations, telles que l'élévation ou la température, pour un endroit spécifique de la surface de la Terre, qui peut représenter divers éléments continus :

- **Images** : L'imagerie satellite ou les photographies aériennes, où chaque pixel peut représenter une couleur ou une nuance.
- **Cartes** : Les cartes topographiques ou thématiques utilisent souvent des données matricielles pour indiquer des variations, telles que des changements d'altitude ou d'utilisation du sol.
- **Couches de données continues** : Ces couches peuvent présenter la température, les précipitations ou les niveaux de pollution, où chaque pixel indique l'intensité ou l'ampleur de cet attribut particulier.



FIGURE 1.2 – Exemple de données matricielles (raster)

1.2 Sources des données géospatiales

Les données géospatiales proviennent de diverses sources, chacune adaptée à des objectifs spécifiques :

- **Satellites** : Les satellites comme Landsat et Sentinel, dans le cadre du programme Copernicus, offrent une couverture à grande échelle de la Terre. Ils fournissent des images gratuites utiles pour la surveillance environnementale, la cartographie et l'observation des zones urbaines et rurales. Ces images permettent un suivi fréquent et à grande échelle des changements géographiques.
- **GPS (Systèmes de Positionnement Global)** : Essentiels pour des relevés de localisation précise à l'échelle locale, les GPS sont utilisés dans des secteurs comme la gestion des infrastructures, la navigation et l'agriculture de précision, offrant une précision au centimètre près.

-
- **Drones** : Ces appareils permettent de réaliser des levés à faible altitude, fournissant des données détaillées. Ils sont particulièrement utilisés pour la surveillance des cultures, la cartographie urbaine et la modélisation 3D des bâtiments, grâce à leur capacité à atteindre des zones difficilement accessibles.
 - **Systèmes d'Information Géographique (SIG)** : Un SIG, ou *Geographic Information System (GIS)* en anglais, est un logiciel permettant d'organiser et de présenter des données alphanumériques spatialement référencées (données géospatiales). Il permet d'acquérir, d'organiser, de gérer, de traiter et de représenter des données géographiques sous forme de plans et de cartes. Il existe de nombreux logiciels SIG. Les plus connus et les plus souvent utilisés en géographie sont ArcGIS et QGIS (logiciel Open Source).
 - Dans un SIG, chaque objet est relié à une fiche (table de données) contenant des informations pour le décrire, telles qu'un nom, une adresse, un descriptif, une distance, etc. Le contenu de ces fiches s'adapte en fonction des besoins.

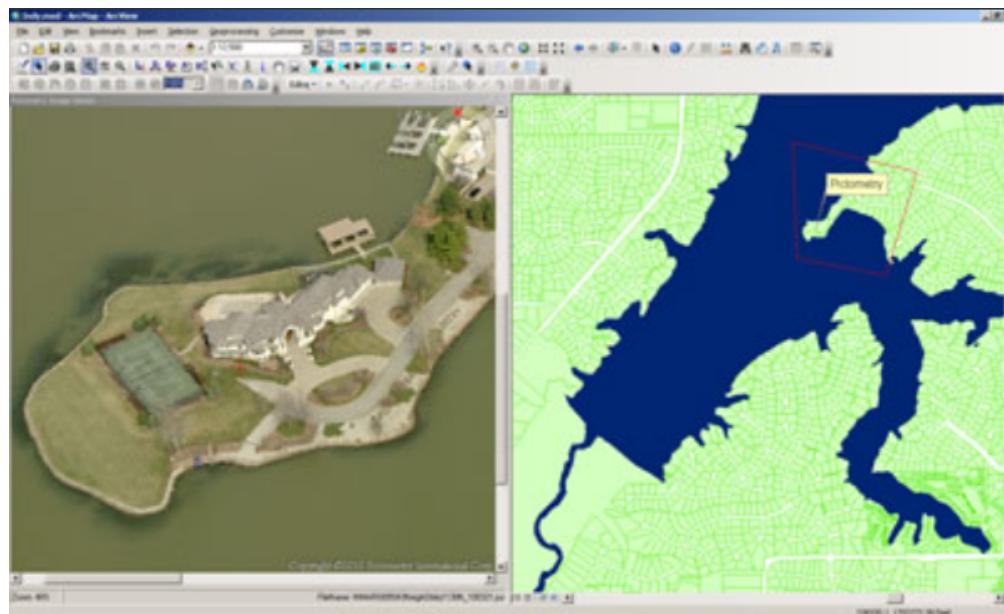


FIGURE 1.3 – Interface de logiciel SIG

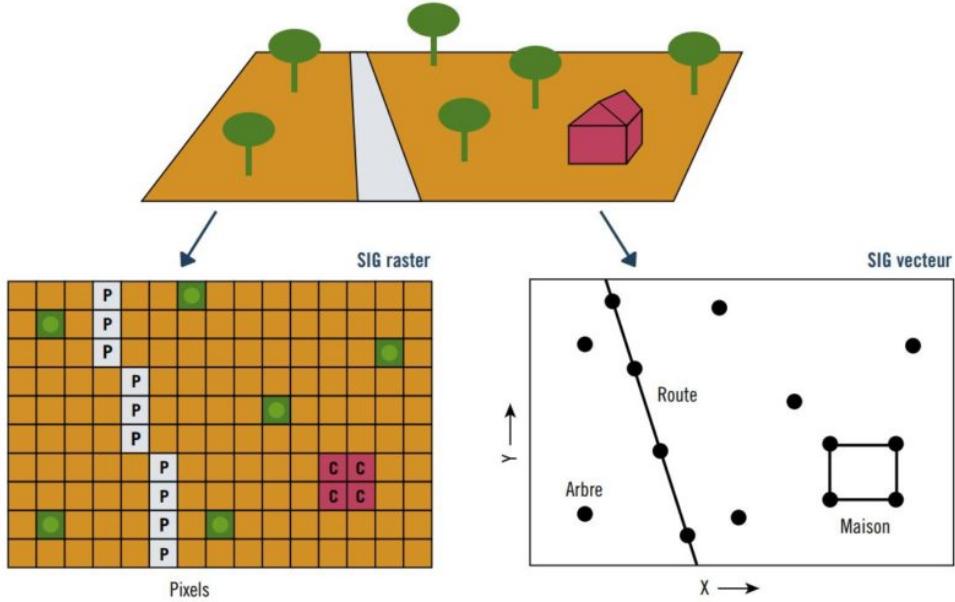


FIGURE 1.4 – Représentation des données par SIG

La figure est une représentation qui combine des données géospatiales : à droite, une visualisation des données vectorielles montrant des formes géométriques précises (points, lignes, polygones), et à gauche, une représentation des données raster avec une grille de pixels.

- **Cadastrages et relevés locaux** : Ces données fournissent des informations précises sur des zones spécifiques, comme les parcelles foncières, les routes et les infrastructures. Elles sont essentielles pour la gestion de l’urbanisme et la planification des transports.

1.3 Enjeux des données géospatiales

L’utilisation des données géospatiales présente plusieurs enjeux majeurs qui touchent à la fois la gestion des données, leur précision, leur accessibilité et leur traitement. Tout d’abord, la précision des données géospatiales est cruciale, notamment dans des applications sensibles telles que la navigation, les interventions d’urgence ou la gestion des ressources naturelles. Les erreurs de localisation peuvent avoir des conséquences graves, ce qui rend impératif un traitement de haute qualité pour garantir leur fiabilité. Cependant, l’accès aux données et leur coût constituent également un obstacle important. Bien que des données comme celles fournies par le programme Copernicus soient disponibles gratuitement, d’autres types de données, comme celles nécessitant des capteurs spécialisés ou des relevés de terrain, peuvent être coûteuses à acquérir. Cela crée des disparités d’accès qui peuvent limiter l’utilisation de certaines ressources.

Un autre enjeu majeur est la gestion de la vie privée. Certaines données géospatiales, comme les déplacements individuels ou les zones sensibles, peuvent potentiellement révéler des informations personnelles. Il est donc essentiel d’assurer une gestion éthique et sécurisée

de ces données, en particulier avec l'augmentation des technologies de surveillance et des systèmes de géolocalisation. Parallèlement, les données géospatiales sont souvent stockées dans des formats variés et selon des normes différentes, ce qui complique leur intégration et leur analyse. Les organisations doivent ainsi disposer des outils et des compétences nécessaires pour gérer cette diversité de formats et garantir une utilisation cohérente des données.

En outre, la gestion du big data géospatial soulève des questions techniques et organisationnelles complexes. Le volume de données générées par des sources comme les satellites, les capteurs ou les stations météorologiques est colossal.

Par exemple, environ 100 To de données liées aux conditions météorologiques sont générées chaque jour. Ce volume pose de sérieux défis en termes de stockage et d'accès aux données, nécessitant des infrastructures puissantes pour les gérer à grande échelle.

De plus, ces données existent dans une multitude de formats, ce qui implique un travail considérable de nettoyage et de reformatage avant qu'elles ne puissent être utilisées efficacement. Cette complexité est encore accrue lorsqu'il s'agit de combiner différentes sources de données pour réaliser une analyse cohérente et fiable.

Enfin, le traitement des données géospatiales nécessite des compétences spécialisées. Il implique souvent des tâches complexes, telles que l'alignement géospatial des couches de données ou la gestion des projections cartographiques. Ces tâches requièrent des analystes qualifiés et expérimentés, capables de manipuler des données complexes et de résoudre des problèmes géométriques avancés.

Chapitre 2

Analyse géospatiale et cartographie

L'analyse géospatiale est devenue un outil indispensable dans le monde d'aujourd'hui, nous permettant d'obtenir des informations précieuses sur notre environnement et de prendre des décisions éclairées. De l'urbanisme et de la surveillance environnementale à la gestion des catastrophes et à l'analyse de marché, l'analyse géospatiale nous offre la capacité d'explorer et de comprendre le monde qui nous entoure comme jamais auparavant.

Par exemple, en analysant les images satellite d'une ville, les analystes géospatiaux peuvent identifier les zones présentant des niveaux élevés de pollution atmosphérique ou les zones à risque d'inondation. Ces informations peuvent ensuite être utilisées pour élaborer des politiques et des interventions visant à résoudre ces problèmes.

Il existe de nombreux outils différents disponibles pour l'analyse géospatiale, allant des logiciels SIG propriétaires aux outils open source comme R et Python. Certains progiciels SIG populaires incluent ArcGIS, QGIS et GRASS GIS. Ces outils permettent aux utilisateurs de visualiser et d'analyser des données spatiales de diverses manières, notamment en créant des cartes, en effectuant des requêtes spatiales et en réalisant des statistiques spatiales.

2.1 Types d'analyse géospatiale

Il existe de nombreux types d'analyse géospatiale, selon le type de données analysées et les questions posées. Certains types courants d'analyse géospatiale comprennent :

- Analyse spatiale : Cela implique d'analyser les relations spatiales entre différents ensembles de données, telles que la distance entre deux points ou la distribution spatiale d'un élément particulier.
- Analyse de réseau : Il s'agit d'analyser les relations entre différents nœuds ou points d'un réseau, comme un réseau de transport ou un réseau social.
- Analyse du terrain : Cela implique l'analyse de la topographie d'une zone particulière, comme l'élévation et la pente d'une chaîne de montagnes.
- Géostatistique : cela implique l'analyse de modèles spatiaux de données à l'aide de méthodes statistiques, telles que le krigeage ou l'autocorrélation spatiale.

2.2 R pour l'analyse et la cartographie géospatiales

R est devenu un choix populaire pour l'analyse géospatiale. Dans cette section, nous explorerons certains des avantages de l'utilisation de R pour l'analyse géospatiale. L'utilisation de R a plusieurs avantages :

- **Intégration avec d'autres outils** : R s'intègre facilement avec Excel, SAS et SPSS, facilitant l'importation et l'exportation des données tout en permettant une analyse combinée.
- **Open source et gratuit** : Accessible sans frais, R bénéficie d'une large communauté de développeurs qui enrichissent continuellement ses fonctionnalités.
- **Visualisation avancée** : Grâce à des packages comme **ggplot2**, **leaflet** et **tmap**, R permet de créer des cartes interactives et statiques personnalisées et esthétiques.
- **Capacités analytiques étendues** : R offre des outils spécialisés pour le traitement des données spatiales, incluant le géocodage, l'interpolation spatiale, le regroupement spatial, ainsi que l'analyse de données raster (images satellites, altitudes).
- **Reproductibilité** : L'utilisation de scripts R permet de documenter le processus d'analyse, garantissant ainsi la traçabilité et la facilité de reproduction des sorties, un atout majeur pour les études géospatiales complexes.

2.2.1 Formats de données géospatiales dans R

Il existe plusieurs formats différents couramment utilisés pour représenter les données géospatiales dans R, notamment **les fichiers de formes**, **GeoJSON** et **KML**.

Les Shapefiles sont un format de données vectorielles largement utilisé dans les applications SIG. Les fichiers de formes se composent de plusieurs fichiers qui stockent différents types d'informations, notamment la géométrie, les attributs et les informations de projection.

GeoJSON est un format couramment utilisé pour les applications de cartographie Web. GeoJSON est un format léger basé sur JavaScript Object Notation (JSON).

KML est un format utilisé par Google Earth et d'autres applications géospatiales. Les fichiers KML sont constitués de données XML qui décrivent l'emplacement et les attributs des objets à la surface de la Terre.

2.2.2 Structures de données géospatiales dans R

Les données géospatiales sont généralement représentées dans R à l'aide de deux structures de données principales : les objets spatiaux et les trames de données.

Les objets spatiaux sont des objets qui représentent des données géospatiales et incluent des informations sur la géométrie, les attributs et la projection des données. Les objets spatiaux peuvent être créés à l'aide du package **sp** dans R.

Les trames de données sont une structure de données courante dans R qui peut être utilisée pour stocker des données tabulaires. Afin de travailler avec des données géospatiales dans R, il est souvent nécessaire de convertir des blocs de données en objets spatiaux à l'aide du package **sp**.

2.2.3 Packages R pour l'analyse et la cartographie géospatiales

Démarrer avec les packages R pour l'analyse et la cartographie géospatiales est une étape essentielle pour maîtriser l'art de l'analyse des données spatiales. R est un langage de programmation puissant doté d'une vaste bibliothèque de packages pour l'analyse et la cartographie géospatiales.

Avec ces packages, les utilisateurs de R peuvent créer des visualisations, effectuer des analyses spatiales et prendre des décisions éclairées basées sur des données spatiales.

Dans cette partie, nous allons présenter les principaux packages R utilisés pour l'analyse des données géospatiales et la création de cartographies. L'objectif est d'introduire les outils les plus répandus, puis de justifier notre choix de nous concentrer sur deux packages particuliers : `sf`, `rastercartography`

— Le package `sf`

Le package `sf` (Simple Features) est l'un des packages les plus populaires pour l'analyse et la cartographie géospatiales dans R remplaçant l'ancien package `sp`. Le package `sf` prend en charge une large gamme d'objets spatiaux, notamment des points, des lignes, des polygones et des grilles. Il comprend également des fonctions de lecture et d'écriture de données spatiales dans divers formats, notamment des fichiers de formes, GeoJSON et KML.

— Le package `raster`

Le package `raster` est conçu pour traiter des données matricielles géospatiales. Il permet d'effectuer des calculs sur les surfaces, tels que l'analyse de l'altitude, des températures ou des indices environnementaux. Bien qu'efficace, il tend à être remplacé par `terra`, plus performant pour les grands jeux de données.

— Le package `leaflet`

Le package `leaflet` est un autre package populaire pour l'analyse et la cartographie géospatiales dans R. Il fournit un moyen simple et flexible de créer des cartes interactives dans R. Le package est construit sur le package `htmlwidgets` et fournit un moyen de créer des cartes pouvant être intégrées dans des pages Web ou d'autres graphiques R. Le package `leaflet` prend en charge un large éventail de couches cartographiques, notamment les couches de tuiles, les couches WMS et les couches GeoJSON. Il fournit également une variété d'options pour personnaliser l'apparence de la carte, telles que l'ajout de marqueurs, de fenêtres contextuelles et de légendes.

— Le package `tmap`

Le package `tmap` est un package puissant pour créer des cartes thématiques dans R. Il fournit un moyen de créer des cartes qui montrent la distribution d'une variable particulière sur une zone géographique. Le package prend en charge un large éventail de types de cartes, notamment les cartes choroplèthes, les cartes de densité de points et les cartes de symboles proportionnels. Il fournit également une variété d'options pour personnaliser l'apparence de la carte, telles que l'ajout de légendes, d'étiquettes et de titres. Le package `tmap` prend également en charge les cartes interactives, qui permettent aux utilisateurs d'explorer les données plus en détail.

— Le package `ggplot2`

Le package `ggplot2` est un package populaire pour créer des graphiques statiques dans R. Il fournit un moyen de créer des cartes montrant la distribution d'une variable

particulière sur une zone géographique. Le package prend en charge un large éventail de types de cartes, notamment les cartes choroplèthes, les cartes de densité de points et les cartes de symboles proportionnels. Il fournit également une variété d'options pour personnaliser l'apparence de la carte, telles que l'ajout de légendes, d'étiquettes et de titres. Le package `ggplot2` prend également en charge les cartes interactives, qui permettent aux utilisateurs d'explorer les données plus en détail.

— **Le package cartography**

Le package `cartography` complète `sf` en facilitant la création de cartes thématiques professionnelles. Il prend en charge divers styles cartographiques, tels que les cartes choroplèthes et les flux, permettant de produire des visualisations informatives et esthétiques.

Le choix de la meilleure option pour l'analyse et la cartographie géospatiales dans R dépend des besoins et des préférences de l'utilisateur. Pour les utilisateurs qui souhaitent créer des cartes interactives, le package `leaflet` est un excellent choix. Pour les utilisateurs qui souhaitent créer des cartes thématiques, le package `tmap` est une excellente option. Pour les utilisateurs qui souhaitent créer des graphiques statiques, le package `ggplot2` est un choix fiable. Cependant, le package `sf` est une excellente option pour tous les types d'analyse spatiale et de cartographie. Il s'agit d'un package puissant et flexible capable de gérer un large éventail d'opérations spatiales et de formats de données. Enfin, pour ceux qui souhaitent produire des cartes thématiques statiques avec une grande variété de styles et d'options de mise en page, le package `cartography` est un excellent choix.

2.2.4 Manipulation et gestion des données géospatiales avec R

Dans cette section, nous allons explorer et manipuler des données géographiques de la France en utilisant les packages `raster` et `sf` dans R. Le package `raster` nous permet de charger rapidement des données géospatiales, et grâce à la fonction `getData()`, nous allons obtenir les données administratives de la France à partir de la base GADM. Ces données seront au format `sp` (`SpatialPointsDataFrame`), mais nous les convertirons ensuite au format `sf` (`Simple Features`), afin de pouvoir les manipuler plus facilement avec des outils modernes de géomatique, notamment le package `cartography`.

Nous commencerons par charger les limites administratives de la France au niveau des régions (niveau 1 de la hiérarchie GADM).

```
library(raster)
fr <- getData('GADM', country='FRA', level=1)
plot(fr)
is(fr)
frsf <- st_as_sf(fr, "sf") # Conversion au format sf
st_bbox(frsf) # Identifier les limites
```

Sortie :

Les coordonnées géographiques des limites de la France au niveau des régions sont extraites comme suit :

```
xmin      ymin      xmax      ymax  
-5.143751 41.333752 9.560416 51.089397
```

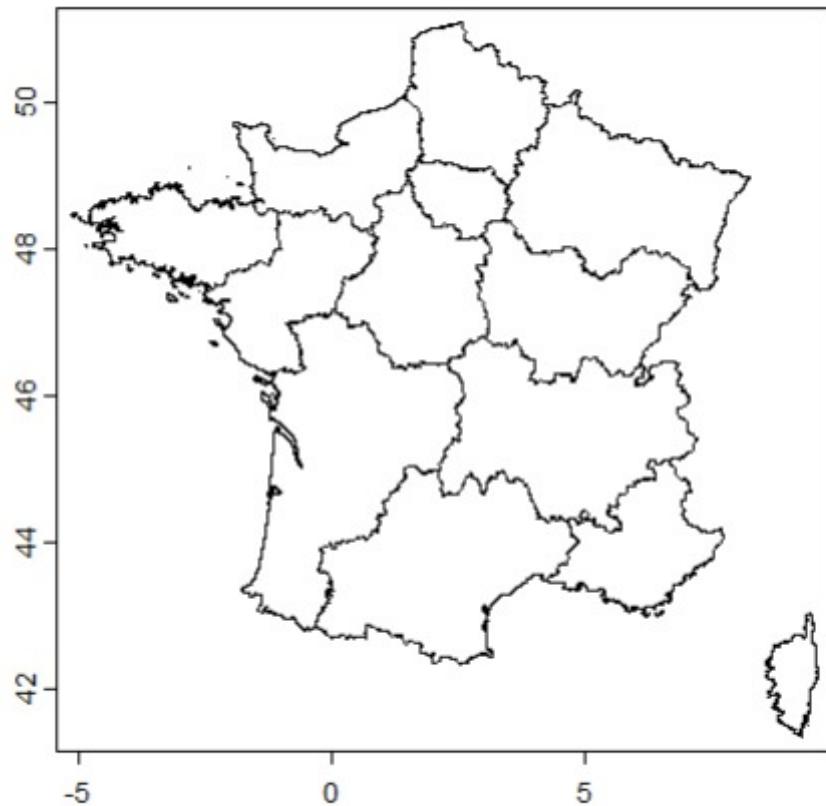
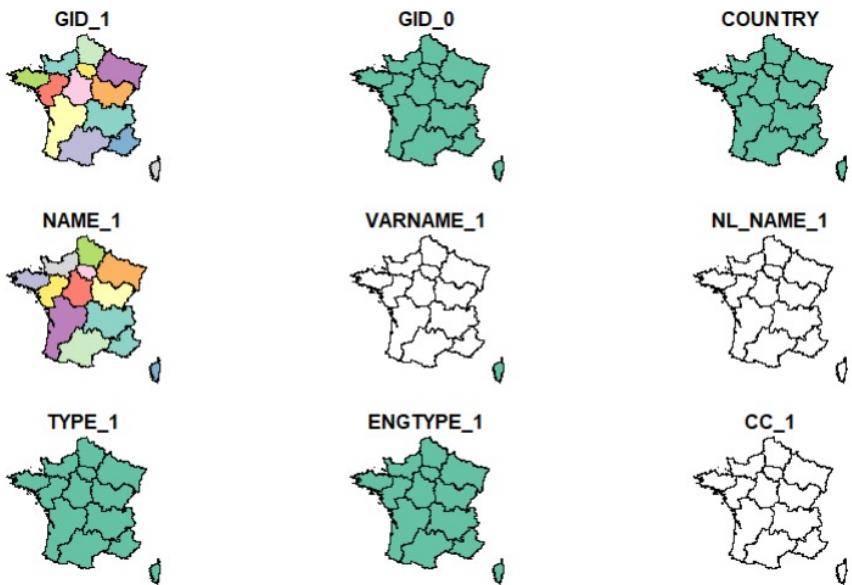


FIGURE 2.1 – Limites administratives de France

Une fois les données converties au format `sf`, nous pouvons maintenant procéder à l'affichage de la carte en utilisant différentes fonctions pour visualiser les régions en utilisant le code suivant :

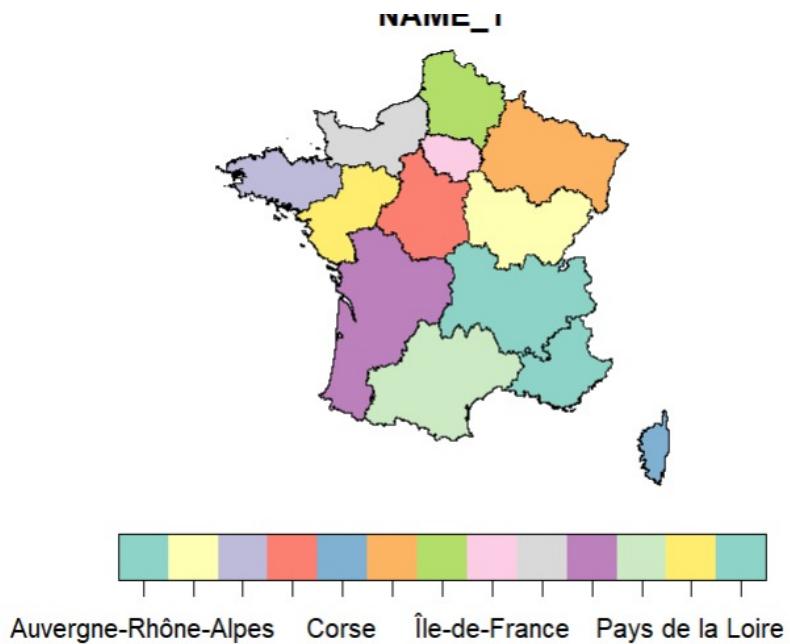
```
names(frsf)  
plot(frsf) # Toutes les cartes possibles
```

Sortie :



```
plot(frsf[4])
```

Sortie :



Nous allons maintenant calculer la superficie de la région Normandie, tant en mètres carrés (m^2) qu'en kilomètres carrés (km^2). Pour ce faire, nous utilisons le code suivant :

```
normandie$area_m2 <- st_area(normandie)
normandie$area_km2 <- as.numeric(normandie$area_m2) / 1e6 # Conversion en km2
print(normandie$area_km2)
```

Sortie :

La superficie de la région Normandie est calculée et le Sortie obtenu est :

30075.23

Ensuite, nous calculons le centroïde de la région Normandie, qui représente le centre géographique de cette région. Le code suivant permet de le calculer et de l'afficher sur une carte :

```
centroid_normandie <- st_centroid(normandie)
# le centroid sur la carte
plot(st_geometry(normandie), col = "lightblue", border = "darkblue",
main = "Normandie avec centroid")
plot(st_geometry(centroid_normandie), add = TRUE, col = "red", pch=16)
```

Sortie :

Normandie avec centroid



FIGURE 2.2 – Centroïde de la région Normandie

Nous allons créer une zone tampon de 50 km autour de la région Normandie. Une zone tampon est une région autour d'un objet géospatial, ici autour de la Normandie, et elle est utilisée pour effectuer des analyses spatiales comme la proximité ou la relation avec d'autres objets géographiques. Le code suivant permet de créer cette zone tampon en utilisant la fonction `st_buffer()`, avec une distance de 50 000 mètres (50 km) :

```
buffer_normandie <- st_buffer(normandie, dist = 50000) # 50 km en mètres
```

Ensuite, nous allons afficher cette zone tampon en la superposant sur la carte de la Normandie :

```
# la zone tampon
plot(st_geometry(buffer_normandie), col = "lightgreen", border = "darkgreen",
main = "Zone tampon autour de Normandie")
plot(st_geometry(normandie), add = TRUE, col = "lightblue",
border = "darkblue")
```

Sortie :

Zone tampon autour de Normandie

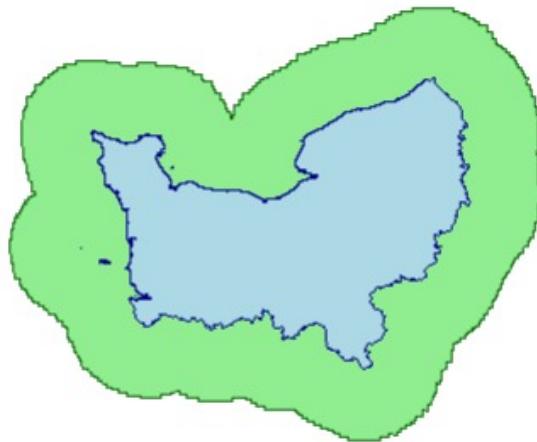


FIGURE 2.3 – Zone tampon autour de la région Normandie

Nous avons calculé la distance entre le centroïde de la Normandie et le centroïde de sa première région voisine en utilisant la fonction `st_distance()`. Le Sortie nous donne une distance en mètres.

```
distance_normandie <- st_distance(centroid_normandie, centroids[2, ])
print(distance_normandie)
```

Sortie : La distance entre la Normandie et sa première région voisine est de : 406934.1

mètres (soit environ 406.93 km)

Nous allons créer une carte choroplète représentant les régions de la Normandie, colorées en fonction de leur superficie. Voici le code pour créer cette carte avec le package `ggplot2`.

```
ggplot(data = normandie) +
  geom_sf(aes(fill = area_km2)) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  theme_minimal() +
```

```
labs(title = "Carte Choroplthe des Rgions de Normandie",
     fill = "Superficie (km)") +
theme(legend.position = "topright")
```

Sortie :

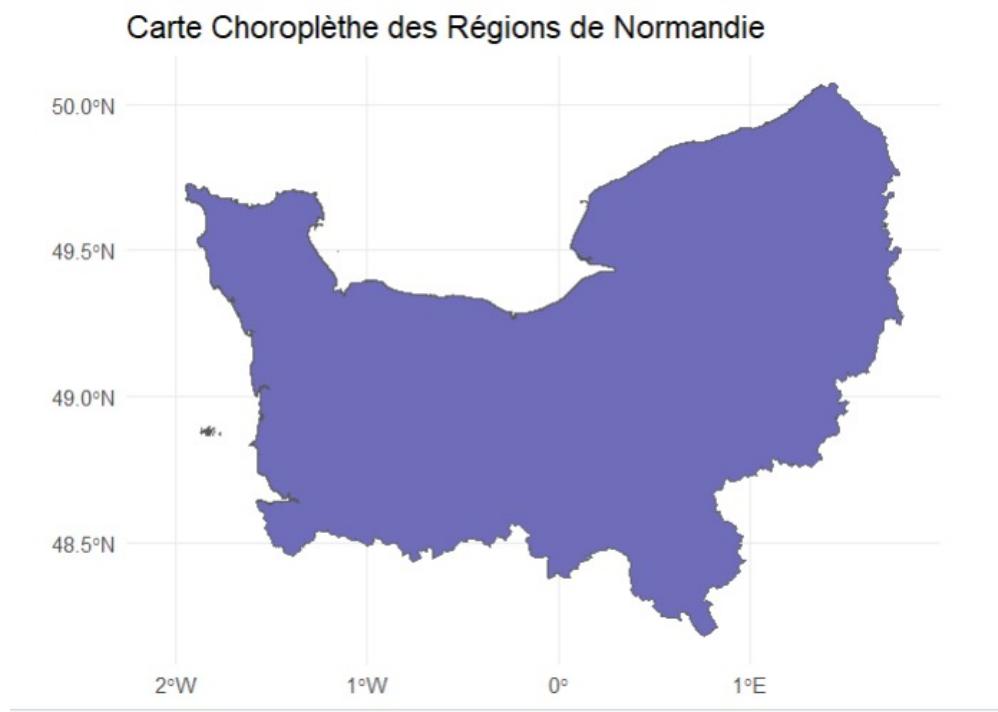


FIGURE 2.4 – Carte choroplète de Normandie

La carte choroplète générée montre les régions de la Normandie colorées en fonction de leur superficie, avec un dégradé de couleurs allant du bleu clair (pour les plus petites surfaces) au bleu foncé (pour les plus grandes surfaces).

Cette visualisation permet de voir clairement la répartition des superficies des différentes régions, offrant ainsi une meilleure compréhension de la dimension relative de chaque région. Les régions plus grandes sont facilement identifiables grâce à la couleur bleue foncée, tandis que les petites régions apparaissent en bleu clair.

2.3 Analyse géospatiale avec Python

Python est devenu un outil incontournable pour l'analyse géospatiale en raison de son **écosystème riche en bibliothèques**, de sa **capacité à gérer de grands volumes de données**, et de son interopérabilité avec les bases de données et les SIG.

Les analyses géospatiales en Python se structurent autour de plusieurs axes :

-
- **Manipulation et gestion des données géospatiales** : Lecture et traitement des données vectorielles et raster, transformation des projections, opérations géométriques et jointures spatiales.
 - **Analyse et modélisation spatiale** : Étude des réseaux et itinéraires, interpolation spatiale, analyse de la distribution des phénomènes géographiques, modélisation des zones à risque et accessibilité.
 - **Visualisation et cartographie** : Création de cartes statiques et interactives, classification thématique, représentation des flux et hotspots géographiques.
 - **Interopérabilité avec les SIG et bases de données spatiales** : Connexion à PostGIS, SpatiaLite et bases SIG, interaction avec QGIS et ArcGIS, exportation et intégration des données dans des flux de travail géomatiques.

2.3.1 Packages Python pour l'analyse et la cartographie géospatiales

Dans cette partie, nous allons présenter les principaux packages Python utilisés pour l'analyse des données géospatiales et la création de cartographies. L'objectif est d'introduire les outils les plus répandus, puis de justifier notre choix de nous concentrer sur trois packages particuliers : [GeoPandas](#), [Shapely](#) et [Folium](#).

- **Le package GeoPandas**

Le package [GeoPandas](#) est l'un des packages les plus populaires pour l'analyse et la manipulation des données géospatiales vectorielles en Python. Il est conçu pour simplifier le travail avec des données géométriques sous forme de points, de lignes et de polygones. Il étend les fonctionnalités de [Pandas](#) pour prendre en charge des objets géométriques. De plus, [GeoPandas](#) permet de lire et d'écrire des fichiers au format [Shapefile](#), [GeoJSON](#), et d'autres formats géospatiaux.

- **Le package Rasterio**

Le package [Rasterio](#) est conçu pour lire et écrire des données raster, comme les images satellites ou les cartes d'élévation. Il est particulièrement adapté pour la manipulation des fichiers [GeoTIFF](#) et autres formats raster. Il offre une interface simple pour lire des données géospatiales en format matriciel et permet de réaliser des opérations géospatiales complexes comme la transformation de la projection et le calcul de statistiques sur les images raster.

- **Le package Folium**

Le package [Folium](#) permet de créer des cartes interactives en utilisant [Leaflet.js](#). C'est une bibliothèque particulièrement utile pour afficher des données géospatiales dans un format interactif, où les utilisateurs peuvent zoomer et se déplacer sur la carte. [Folium](#) permet d'ajouter facilement des couches supplémentaires sur une carte, comme des points, des lignes, des polygones, ou encore des fonds de cartes provenant de sources comme OpenStreetMap ou Stamen Maps.

- **Le package Shapely**

Le package [Shapely](#) est utilisé pour effectuer des opérations géométriques sur des objets spatiaux. Il permet de manipuler des objets géométriques, tels que des points, des lignes et des polygones, et de réaliser des calculs géométriques tels que l'intersection, l'union, ou la différence. Il est souvent utilisé en complément de [GeoPandas](#).

pour des analyses géométriques avancées, comme le calcul des distances ou des zones d'intersection.

2.3.2 Manipulation et gestion des données géospatiales avec Python

Dans cette section, nous allons travailler avec un jeu de données géospatiales spécifique afin de réaliser des manipulations et des analyses spatiales. Nous utilisons ici des fichiers au format Shapefile et CSV, qui contiennent des informations géographiques pour les arrondissements de Paris, les zones IRIS (Ilots Regroupés pour l'Information Statistique) et des informations sur les arbres de la ville.

L'importation de ces fichiers peut être effectuée avec **GeoPandas** pour les données vectorielles et **Rasterio** pour les données raster :

```
import geopandas as gpd
gdf = gpd.read_file("data/quartiers.shp") # Chargement dun shapefile

import rasterio
with rasterio.open("data/image_satellite.tif") as src:
    image = src.read(1) # Extraction de la premire bande spectrale
```

Chargement des données vectorielles

Les données vectorielles représentent des objets géographiques sous forme de points, de lignes ou de polygones. Elles sont souvent stockées dans des formats comme **Shapefile** ou **GeoJSON**.

Pour charger des données géospatiales avec **GeoPandas** :

```
# On charge les fichiers Shapefile
arrondissements = gpd.read_file("arrondissements/arrondissements.shp")
metro = gpd.read_file(
    "transport/traces-des-lignes-de-transport-en-commun-idfm.shp"
)
IRIS = gpd.read_file("iris_paris/CONTOURS-IRIS.shp")
```

Dans le cas de fichiers CSV contenant des coordonnées géographiques, nous devons convertir les données en **GeoDataFrame** :

```
import pandas as pd

# On charge le fichier CSV
arbres = pd.read_csv("Arbre.csv", sep=";", header=0)

# On extrait des coordonnes depuis la colonne "geo_point_2d"
arbres[["Latitude", "Longitude"]] = arbres["geo_point_2d"].str.split(",",
expand=True).astype(float)
```

```
# On fait une transformation en GeoDataFrame
arbres = gpd.GeoDataFrame(arbres,
geometry=gpd.points_from_xy(arbres["Longitude"], arbres["Latitude"]),
crs=4326)
```

Reprojection des données spatiales

Les systèmes de coordonnées jouent un rôle clé dans les analyses géospatiales. Par défaut, de nombreuses données sont en **WGS 84** (EPSG:4326), mais les calculs de distance et de surface nécessitent souvent un système projeté comme **Lambert 93** (EPSG:2154 pour la France métropole).

```
import matplotlib.pyplot as plt

# Affichage du shapefile des arrondissements
ardt.plot(edgecolor="black", figsize=(8, 8))
plt.title("Carte des Arrondissements de Paris")
plt.show()
```

Sortie :

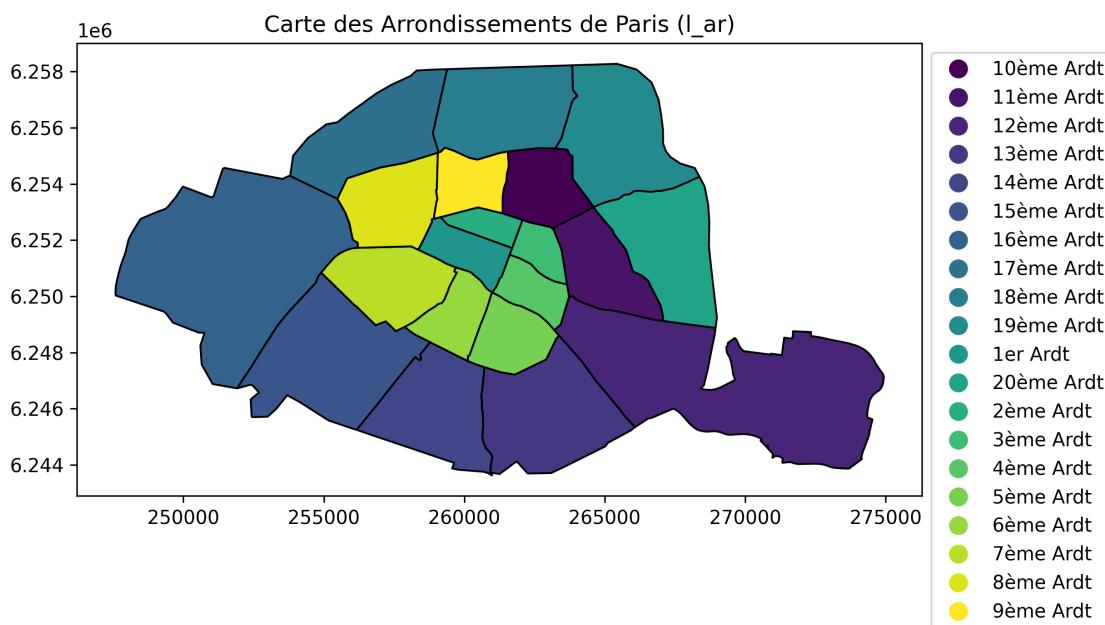


FIGURE 2.5 – Carte des arrondissements de Paris

Projection des IRIS au sein des arrondissements

Les IRIS (Ilots Regroupés pour l'Information Statistique) sont des unités géographiques utilisées pour l'analyse statistique en France. Afin de visualiser la répartition des IRIS au sein des arrondissements de Paris, il est essentiel que les données géographiques soient dans un système de projection cohérent. Dans ce cas, les données sont reprojetées dans le système de coordonnées Lambert 93 (EPSG:2154), qui est adapté aux données géographiques en France.

Une fois la reprojection effectuée, les IRIS peuvent être superposés sur les limites des arrondissements pour analyser leur répartition géographique.

```
# Creation de la figure pour afficher les IRIS par arrondissement
fig, ax = plt.subplots(figsize=(10, 10))

# Tracer la carte des IRIS colore par arrondissement
IRIS.plot(column="ARRONDISSEMENT", cmap="rainbow", edgecolor="black",
legend=False, ax=ax)

# Ajouter un titre
plt.title("Carte des IRIS par arrondissement Paris")

# Sauvegarde de la carte
plt.savefig("carte_IRIS_arrondissement.png", dpi=300, bbox_inches="tight")

# Afficher la carte
plt.show()
```

Sortie :

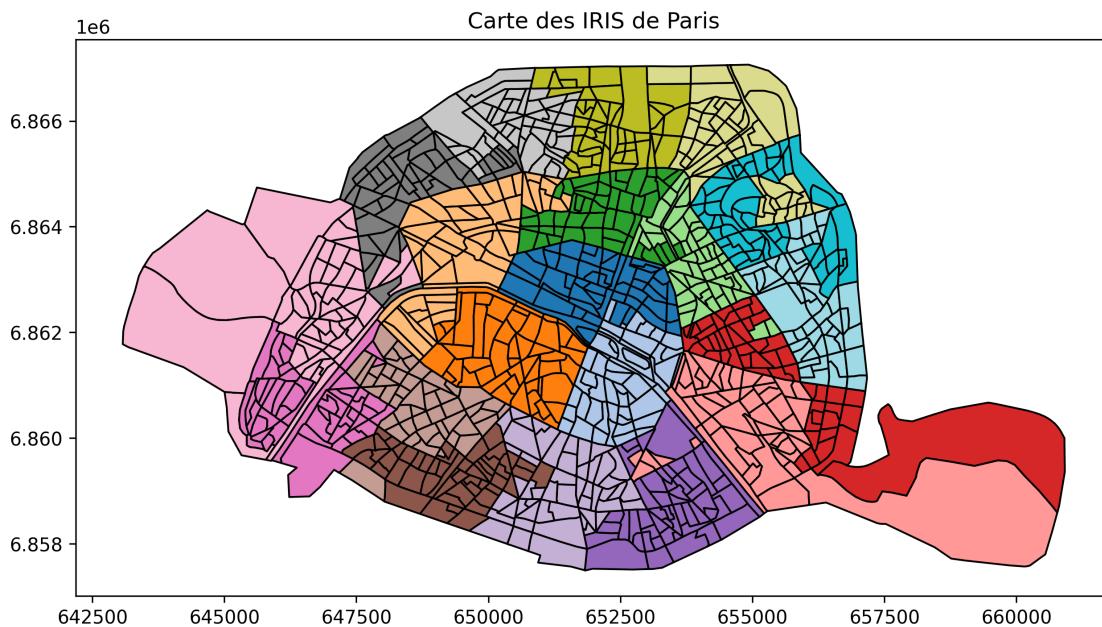


FIGURE 2.6 – Carte des IRIS de Paris

L'affichage des cartes peut être enrichi en superposant un fond cartographique issu de serveurs de tuiles.

La bibliothèque **Contextily** permet d'intégrer automatiquement des fonds de carte provenant de sources comme OpenStreetMap ou Stamen Maps, offrant ainsi une meilleure contextualisation géographique.

Cette approche est particulièrement utile pour visualiser des données spatiales dans leur environnement réel, sans nécessiter de couches SIG supplémentaires. En reprojectant les données dans un système compatible avec les fonds de carte, il devient possible d'afficher des cartes plus détaillées et lisibles, améliorant ainsi l'interprétation des Sorties.

```
import contextily as ctx

# Convertir en projection Web Mercator pour afficher le fond de carte
ardt = ardt.to_crs(epsg=3857)

# On trace la carte avec un fond OpenStreetMap
ax = ardt.plot(edgecolor="black", figsize=(8, 8), alpha=0.7)
ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik)
plt.title("Carte des Arrondissements avec Fond OpenStreetMap")
plt.savefig("carte_IRIS_streatmap.png", dpi=300, bbox_inches="tight")

plt.show()
```

Sortie :

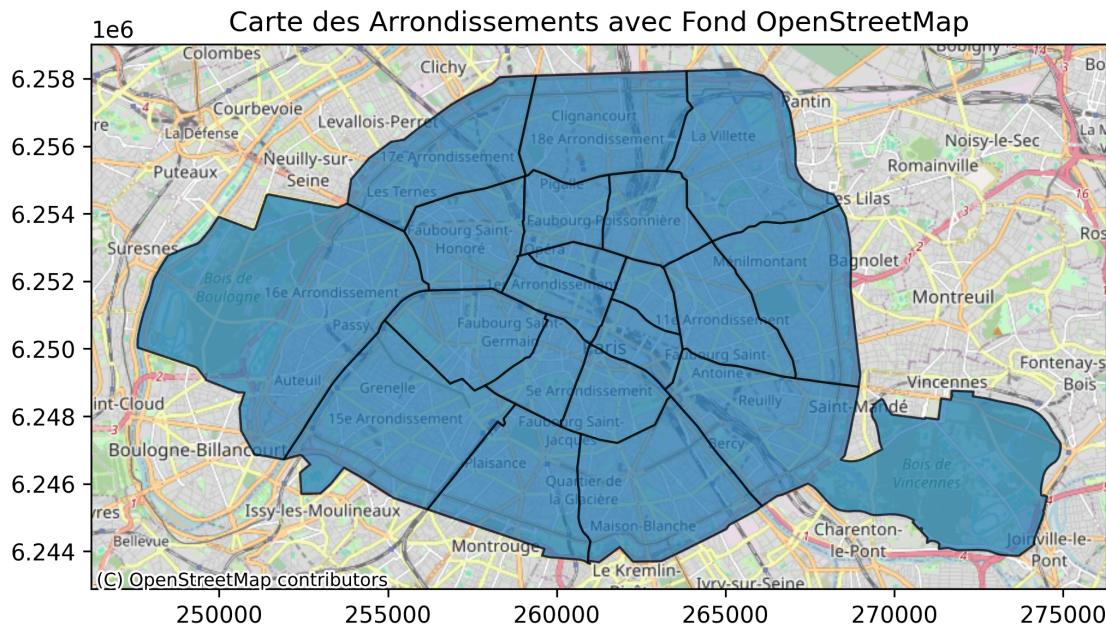


FIGURE 2.7 – Carte des Arrondissements de Paris

Calcul du périmètre

Le périmètre d'une zone est défini comme la longueur totale de son contour géométrique. Cette mesure est essentielle pour analyser la forme et la compacité des entités spatiales. Le périmètre est d'abord calculé en **mètres** (m), puis converti en **kilomètres** (km) afin de faciliter l'interprétation.

```
# Creation de la figure pour l'affichage du perimetre
fig, ax = plt.subplots(figsize=(10, 10))

# Tracer la carte en classifiant le perimetre en quantiles
ArdtINSEE.plot(
    column="perimetre_km", # Utilisation du primtre en kilomtres
    cmap="viridis",
    scheme="quantiles",
    legend=True,
    ax=ax,
    classification_kwds={"k": 5}
)

# Personnalisation de la carte
plt.title("Rpartition du primtre (km) par quantile")
plt.grid(False) # Dsactiver la grille
plt.axis("off") # Supprimer les axes

# Sauvegarde et affichage de la carte
plt.savefig("perimetre_IRIS.png", dpi=300, bbox_inches="tight")

plt.show()
```

Sortie :

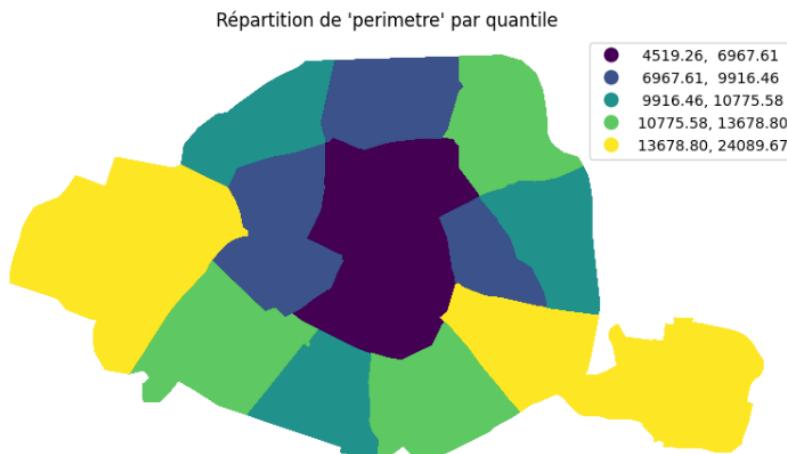


FIGURE 2.8 – Carte de répartition des périmètres des arrondissements de Paris

Chapitre 3

Application : Analyse de l'accessibilité des établissements de santé en zone rurale en Normandie

Cette section présente l'analyse de l'accessibilité aux établissements de santé en Normandie, avec une attention particulière portée aux zones rurales.

L'objectif principal de cette étude est d'identifier les communes sous-desservies par rapport à l'offre de soins, en vue de proposer des solutions concrètes pour améliorer les services de santé dans ces zones. Cette analyse s'appuie sur l'intégration de données géospatiales et démographiques, en combinant les limites administratives des communes, les infrastructures routières, et la localisation des établissements de santé. L'objectif est de fournir une évaluation précise de l'accessibilité, en tenant compte non seulement de la distance géographique mais aussi des réalités des infrastructures existantes.

3.1 Sources de données

L'analyse repose sur un ensemble diversifié de jeux de données géospatiaux, offrant une vision complète de la situation de l'accessibilité aux services de santé en Normandie. Les principales sources de données utilisées dans cette étude sont :

- **Données démographiques de l'INSEE** : Ces données ont permis d'identifier les zones rurales et d'évaluer la densité de la population dans chaque commune. Elles ont également permis d'analyser les populations potentiellement vulnérables en cas d'accès insuffisant aux soins.
- **Limites administratives des communes** : Les limites des communes de Normandie ont été extraites de la plateforme data.gouv.fr. Ces données géospatiales ont servi à définir les frontières des zones à analyser, en permettant une vue précise des délimitations administratives des communes.
- **Réseaux routiers d'OpenStreetMap (OSM)** : Les informations provenant d'OpenStreetMap ont été utilisées pour créer un modèle du réseau routier régional. Ce réseau a été essentiel pour évaluer l'accessibilité réelle aux établissements de santé en prenant

en compte les routes existantes, y compris les autoroutes, les routes principales et secondaires, mais aussi les routes rurales moins bien développées.

- **Établissements de santé (Fichier GeoJSON)** : La localisation des établissements de santé a été fournie sous le format GeoJSON, permettant de cartographier avec précision les hôpitaux, cliniques et centres de santé en Normandie.

3.2 Méthodologie

Nous allons commencer par l'analyse de données géospatiales en utilisant Python. Cette première étape nous permettra d'explorer, visualiser et traiter les données spatiales à l'aide de bibliothèques spécialisées telles que GeoPandas, Shapely, Folium, Contextily et osmnx.

La méthodologie utilisée pour ce projet suit plusieurs étapes clés :

3.2.1 Collecte de données

On présente une analyse basée sur des scripts Python, incluant les codes, leurs explications et les sorties obtenues.

Installation des bibliothèques

```
!pip install geopandas shapely folium matplotlib contextily  
osmnx networkx rtree openpyxl
```

Ce code installe les bibliothèques nécessaires pour la manipulation géospatiale, la cartographie et l'analyse des réseaux.

Importation des bibliothèques

```
import json  
import geopandas as gpd  
from shapely.geometry import Point, LineString, Polygon  
from shapely.ops import nearest_points  
import pandas as pd  
import numpy as np  
import folium  
import matplotlib.pyplot as plt  
import contextily as ctx  
import osmnx as ox  
import networkx as nx  
import os
```

Ce bloc de code importe :

- **json** : pour manipuler les données JSON.
- **geopandas** : pour le traitement des données géospatiales.
- **shapely.geometry** et **shapely.ops** : pour gérer et manipuler les objets géométriques.

- **pandas** et **numpy** : pour l'analyse et le traitement des données.
- **folium**, **matplotlib.pyplot**, **contextily** : pour la visualisation cartographique et l'ajout de fonds de cartes.
- **osmnx** : pour télécharger et analyser des données cartographiques OpenStreetMap, notamment les réseaux routiers.
- **networkx** : pour la modélisation et l'analyse des graphes, notamment pour la recherche de plus courts chemins sur des réseaux routiers.
- **os** : pour interagir avec le système de fichiers, gérer les chemins et les fichiers.

Chemin des fichiers

```
data_dir = "C:\\\\Users\\\\hp\\\\Desktop\\\\Projet_Analyse donnees gospatiales\n\\\\projet_accessibilite_sante\\\\normandie\\\\"

gis_osm_roads_fp = data_dir + "gis_osm_roads_free_1.shp"
etab_fp = data_dir + "Etab.geojson"
decoupage_fp = data_dir + "decoupage_Normandie.geojson"
demographie_fp = data_dir + "nr_ina_92.xlsx"
```

Ce bloc de code définit le chemin principal `data_dir` où sont stockées les données nécessaires à l'analyse. Ensuite, il construit les chemins spécifiques pour chaque fichier de données :

- `gis_osm_roads_fp` : fichier contenant les routes en format shapefile.
- `etab_fp` : fichier GeoJSON contenant les établissements de santé.
- `decoupage_fp` : fichier GeoJSON avec le découpage territorial de la Normandie.
- `demographie_fp` : fichier Excel contenant des données démographiques.

Chargement des données géospatiales et démographiques

```
roads_gdf = gpd.read_file(roads_fp)
etab_gdf = gpd.read_file(etab_fp)
decoupage_gdf = gpd.read_file(decoupage_fp)
```

```
demographie_df = pd.read_excel(demographie_fp)
```

Le GeoDataFrame `roads_gdf` contient des informations sur les routes, principalement sous forme de lignes (LINESTRING). Il inclut des colonnes telles que :

- `osm_id` : un identifiant unique pour chaque route provenant d'OpenStreetMap,
- `fclass` : le type de la route (par exemple, secondaire, autoroute),
- `name` : le nom de la route,
- `maxspeed` : la vitesse maximale autorisée,
- `geometry` : la géométrie associée, représentant le trajet de chaque route sous forme de ligne.

Le GeoDataFrame `etab_gdf` contient des informations sur divers établissements de santé (hôpitaux, cliniques). Il inclut des colonnes telles que :

- `title` : le nom de l'établissement,

-
- `field_type_etablissement` : le type d'établissement (par exemple, public ou privé),
 - `field_adresse_brut` : l'adresse de l'établissement,
 - `geometry` : la géométrie associée, sous forme de points (`POINT`), représentant la localisation de chaque établissement.

Le GeoDataFrame `découpage_gdf` contient des informations géographiques administratives, telles que les communes et départements. Ce GeoDataFrame inclut des colonnes comme :

- `COM_NV` : le nom de la commune,
- `INSEE_NV` : le code INSEE de la commune,
- `DPT` : le département auquel appartient la commune,
- `geometry` : la géométrie associée, représentée sous forme de polygones (`Polygon`), délimitant les zones géographiques de chaque commune.

Aperçu des données

Afin d'obtenir un premier aperçu des données, nous avons affiché les 5 premières lignes de chaque GeoDataFrame par le code suivant :

```
roads_gdf.head(), etab_gdf.head(), découpage_gdf.head()
```

Voici un résumé des premières lignes des trois GeoDataFrames :

Aperçu du GeoDataFrame des routes (`roads_gdf`)

<code>osm_id</code>	<code>fclass</code>	<code>name</code>	<code>maxspeed</code>	<code>geometry</code>
4045344	secondary	None	80	LINESTRING (-0.42526 49.16388, 49.163...)
4045346	trunk_link	None	90	LINESTRING (-0.30404 49.17008, 49.170...)
4261498	motorway	Autoroute de Normandie A 13	130	LINESTRING (-0.02186 49.24304, 49.244...)
4261501	motorway	Autoroute de Normandie A 13	130	LINESTRING (-0.02195 49.24313, 49.241...)
4298291	primary	Avenue de la Côte de Nacre	50	LINESTRING (-0.36101 49.203, -0.36093 49.20254)

Le GeoDataFrame `roads_gdf` contient des informations sur les routes, principalement sous forme de lignes (`LINESTRING`). Il inclut des colonnes telles que :

- `osm_id` : un identifiant unique pour chaque route provenant d'OpenStreetMap,
- `fclass` : le type de la route (par exemple, secondaire, autoroute),
- `name` : le nom de la route,
- `maxspeed` : la vitesse maximale autorisée,

-
- **geometry** : la géométrie associée, représentant le trajet de chaque route sous forme de ligne.

Aperçu du GeoDataFrame des établissements de santé (etab_gdf)

title	field_type_etablissement	field_adresse_brut	geometry
Clinique du Petit Colmoulin	PRIVE	Rue Robert Ancel	POINT (0.20569 49.52141)
Centre de rééducation des Jonquilles	PRIVE	74 rue de la libération	POINT (0.25239 49.51733)
Hôpital privé de l'Estuaire	PRIVE	505 rue Irene Joliot Curie	POINT (0.10079 49.52901)
Centre de la Hève	PRIVE	234 rue Stendhal	POINT (0.08719 49.52038)
Clinique Océane	PRIVE	514 rue Irène Joliot Curie	POINT (0.10412 49.52961)

Le GeoDataFrame `etab_gdf` contient des informations sur divers établissements de santé (hôpitaux, cliniques). Il inclut des colonnes telles que :

- **title** : le nom de l'établissement,
- **field_type_etablissement** : le type d'établissement (par exemple, public ou privé),
- **field_adresse_brut** : l'adresse de l'établissement,
- **geometry** : la géométrie associée, sous forme de points (POINT), représentant la localisation de chaque établissement.

Aperçu du GeoDataFrame des découpages administratifs (decoupage_gdf)

COM_NV	INSEE_NV	DPT	geometry
Saint-Sauveur-Villages	50550	50	POLYGON ((-0.42526 49.16388, -0.42529 49.163...))
Saint-Sauveur-le-Vicomte	50551	50	POLYGON ((-0.30404 49.17008, -0.30398 49.170...))
Saint-Senier-sous-Avranches	50554	50	POLYGON ((-0.02186 49.24304, -0.01722 49.244...))

Saint-Martin-de-Fontenay	14623	14	POLYGON ((-0.02195 49.24313, -0.02698 49.241...)
Saint-Vaast-la-Hougue	50562	50	POLYGON ((-0.36101 49.203, -0.36093 49.20254))

Le GeoDataFrame `découpage_gdf` contient des informations géographiques administratives, telles que les communes et départements. Ce GeoDataFrame inclut des colonnes comme :

- `COM_NV` : le nom de la commune,
- `INSEE_NV` : le code INSEE de la commune,
- `DPT` : le département auquel appartient la commune,
- `geometry` : la géométrie associée, représentée sous forme de polygones (`Polygon`), délimitant les zones géographiques de chaque commune.

3.2.2 Nettoyage des données

Les étapes suivantes décrivent les principales actions de nettoyage réalisées pour chaque type de donnée.

- **Données démographiques** : Les données INSEE ont été filtrées pour ne retenir que les informations pertinentes (population par commune, densité, etc.). Des doublons ont été supprimés et des colonnes inutiles ont été retirées pour rendre les données plus accessibles et faciles à analyser.
- **Réseau routier** : Le réseau routier a été filtré pour ne garder que les routes pertinentes pour l'analyse de l'accessibilité, en excluant les petites voies non asphaltées qui n'ont pas d'impact significatif sur l'accessibilité.
- **Établissements de santé** : Les établissements de santé ont été nettoyés en supprimant les points géographiques erronés (sans coordonnées géographiques) et les doublons. Seuls les établissements pertinents pour l'analyse (hôpitaux, cliniques, centres médicaux) ont été conservés.
- **Vérification des géométries** : Une attention particulière a été portée à la vérification des projections géographiques (système de référence des coordonnées ou CRS), afin de garantir la compatibilité des données géospatiales.

```
# Vérification de la structure du DataFrame
demographie_df.info()

# Affichage des premières lignes pour avoir une vue d'ensemble
demographie_df.head()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34955 entries, 0 to 34954
Data columns (total 3 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   1. Les communes normandes selon le type d'espace auquel elles appartiennent 34953 non-null object  
 1   Unnamed: 1                  34952 non-null object  
 2   Unnamed: 2                  34952 non-null object  
dtypes: object(3)
memory usage: 819.4+ KB
1. Les communes normandes selon le type d'espace auquel elles appartiennent  Unnamed: 1      Unnamed: 2
0              Sources : Insee, grille communale de densité, ...      NaN          NaN
1                               NaN          NaN          NaN
2                               NaN          NaN          NaN
3              depcom,C,6    cat_rur,C,7      cat_rur_C,37
4              01001        r2  Rural autonome peu dense

```

Cela montre que le DataFrame contient 34 955 lignes et 3 colonnes, avec des valeurs manquantes dans les colonnes `Unnamed: 1` et `Unnamed: 2`. Les colonnes n'ont pas de noms significatifs, ce qui nécessite une intervention pour les nettoyer.

Le premier aperçu des données avec `demographie_df.head()` révèle que certaines lignes contiennent des informations inutiles, comme des titres de source ou des lignes entièrement vides.

```

# Recharger le fichier en sautant les lignes inutiles
file_path = "C:\\\\Users\\\\hp\\\\Desktop\\\\Projet_Analyse donnees gospatiales
\\\\projet_accessibilite_sante
\\\\normandie\\\\nr_ina_92.xlsx"
demographie_df_clean = pd.read_excel(file_path, skiprows=3)

# Affichage des premières lignes pour vérifier
demographie_df_clean.head()

```

	Unnamed: 0	Unnamed: 1	Unnamed: 2
0	depcom,C,6	cat_rur,C,7	cat_rur_C,37
1	01001	r2	Rural autonome peu dense
2	01002	r1	Rural autonome très peu dense
3	01004	u1	Urbain densité intermédiaire
4	01005	r3	Rural sous faible influence d'un pôle

Les lignes 1 à 3 sont des lignes de métadonnées ou des titres et doivent être ignorées lors

du chargement des données. Nous avons donc décidé de les sauter en utilisant l'option `skiprows=3` lors du rechargement du fichier à partir de la ligne 4.

Après avoir rechargé le fichier, nous avons vérifié les premières lignes pour nous assurer que les données étaient correctement chargées.

```
# Renommer les colonnes pour plus de clarté
demographie_df_clean.columns = ['depcom', 'cat_rur', 'cat_rur_desc']
```

Ensuite, nous avons supprimé les lignes contenant des valeurs manquantes dans la colonne `depcom`, car cette colonne est essentielle pour l'identification des communes. Pour ce faire, nous avons utilisé la méthode `dropna()` sur la colonne `depcom` :

```
# Supprimer les lignes avec des valeurs manquantes dans 'depcom'
demographie_df_clean = demographie_df_clean.dropna(subset=['depcom'])
```

Après avoir supprimé les lignes avec des valeurs manquantes, nous avons réinitialisé l'index du DataFrame avec la méthode `reset_index(drop=True)` afin de garantir une numérotation continue des lignes.

```
# Reinitialiser l'index après suppression
demographie_df_clean.reset_index(drop=True, inplace=True)
```

Le nettoyage des données est désormais terminé, et le DataFrame est prêt à être utilisé pour des analyses plus approfondies.

```
# Affichage du DataFrame nettoyé
demographie_df_clean.head()
```

	depcom	cat_rur	cat_rur_desc
0	depcom,C,6	cat_rur,C,7	cat_rur_,C,37
1	01001	r2	Rural autonome peu dense
2	01002	r1	Rural autonome très peu dense
3	01004	u1	Urbain densité intermédiaire
4	01005	r3	Rural sous faible influence d'un pôle

Avec

- `depsom` : Code INSEE des communes
- `cat_rur` : Code du type d'espace (r1, r2, r3) pour rural et u1 pour urbain)
- `cat_rur_desc` : Description détaillée du type d'espace

Vérification du système de coordonnées (CRS)

Une fois les données nettoyées, il est important de vérifier que toutes les couches géospatiales utilisent le même système de coordonnées. Cela garantit que les données peuvent être superposées correctement. Voici comment nous avons vérifié le CRS de chaque couche géospatiale :

```
# Verification du CRS de chaque couche gospatiale
print("CRS des routes :", roads_gdf.crs)
print("CRS des etablissements de sante :", etab_gdf.crs)
print("CRS du decoupage administratif :", decoupage_gdf.crs)
```

La Sortie du code montre que toutes les couches utilisent le CRS **EPSG :4326**, qui est le système de coordonnées géographiques WGS 84 (basé sur la latitude et la longitude) :

```
CRS des routes : EPSG:4326
CRS des établissements de santé : EPSG:4326
CRS du découpage administratif : EPSG:4326
```

Cela signifie que toutes les couches géospatiales sont déjà dans le même système de coordonnées, ce qui permet de les superposer et d'effectuer les analyses géospatiales sans avoir besoin de les re-projeter.

Nettoyage du réseau routier

```
import geopandas as gpd

# Chargement des donnees
roads_gdf = gpd.read_file("roads.shp")

# Filtrer les routes principales
routes_pertinentes = roads_gdf[roads_gdf['fclass'].isin(['motorway',
    'primary', 'secondary', 'tertiary'])]

# Supprimer les colonnes inutiles
colonnes_utiles_routes = ['osm_id', 'fclass', 'name', 'ref', 'oneway',
    'maxspeed', 'geometry']
routes_pertinentes = routes_pertinentes[colonnes_utiles_routes]

# Vérifier les geometries valides
routes_pertinentes = routes_pertinentes[routes_pertinentes.is_valid]

# Afficher un aperu des donnees nettoyees
print(f"Nombre de routes retenues : {routes_pertinentes.shape[0]}")
print(routes_pertinentes.head())
```

Nombre de routes retenues : 44008							
	osm_id	fclass	name	ref	oneway	maxspeed	geometry
0	4045344	secondary		None	B	80	LINESTRING (-0.42526 49.16388, -0.42529 49.163...
2	4261498	motorway	Autoroute de Normandie	A 13	F	130	LINESTRING (-0.02186 49.24304, -0.01722 49.244...
3	4261501	motorway	Autoroute de Normandie	A 13	F	130	LINESTRING (-0.02195 49.24313, -0.02698 49.241...
4	4298291	primary	Avenue de la Côte de Nacre	None	B	50	LINESTRING (-0.36101 49.203, -0.36093 49.20254)
5	4298292	primary	Avenue de la Côte de Nacre	None	B	50	LINESTRING (-0.36067 49.20255, -0.36076 49.20301)

- Ce script filtre uniquement les routes principales (motorway, primary, secondary, tertiary).
- Il conserve uniquement les colonnes essentielles (osm_id, fclass, name, etc.).
- 44 008 routes ont été retenues, ce qui signifie que les données initiales étaient bien fournies.
- Toutes les géométries sont valides, donc il n'y a pas d'erreur géospatiale.

De même manière pour le nettoyage du découpage administratif et pour les établissements de santé.

3.2.3 Analyse géospatiale de données avec Python

Limites administratives

La première étape de notre analyse consiste à télécharger les limites administratives de la Normandie à partir des données publiques disponibles sur OpenStreetMap. Pour ce faire, nous avons utilisé la fonction `geocode_to_gdf` de la bibliothèque `osmnx`, qui permet de géocoder un nom de région et de récupérer les polygones correspondants sous forme de `GeoDataFrame`.

```
import osmnx as ox

# Definir le nom de la region
region_name = "Normandie, France"

# Telecharger les limites administratives de la Normandie depuis OpenStreetMap
normandie_gdf = ox.geocode_to_gdf(region_name)
```

Avant de procéder à des analyses plus avancées, nous avons vérifié le nombre de polygones récupérés et exploré brièvement les premières lignes des données pour nous assurer de leur pertinence et de leur qualité. Voici un exemple du code utilisé pour effectuer cette étape :

```
# Verifier les donnees recuperees
print(f"Nombre de polygones recuperes : {normandie_gdf.shape[0]}")
normandie_gdf.head()
```

Cela nous a permis de confirmer que le téléchargement avait bien fonctionné et de voir que le nombre de polygones récupérés est 1.

Après, Nous avons créé une visualisation simple des limites administratives de la Normandie. Cette carte nous permet d'avoir une vue d'ensemble des frontières de la région et de sa structure géographique. Voici le code utilisé pour générer cette carte :

```
# Visualiser la carte  
normandie_gdf.plot(edgecolor='black', facecolor='lightblue', figsize=(10, 10))
```

Sortie :

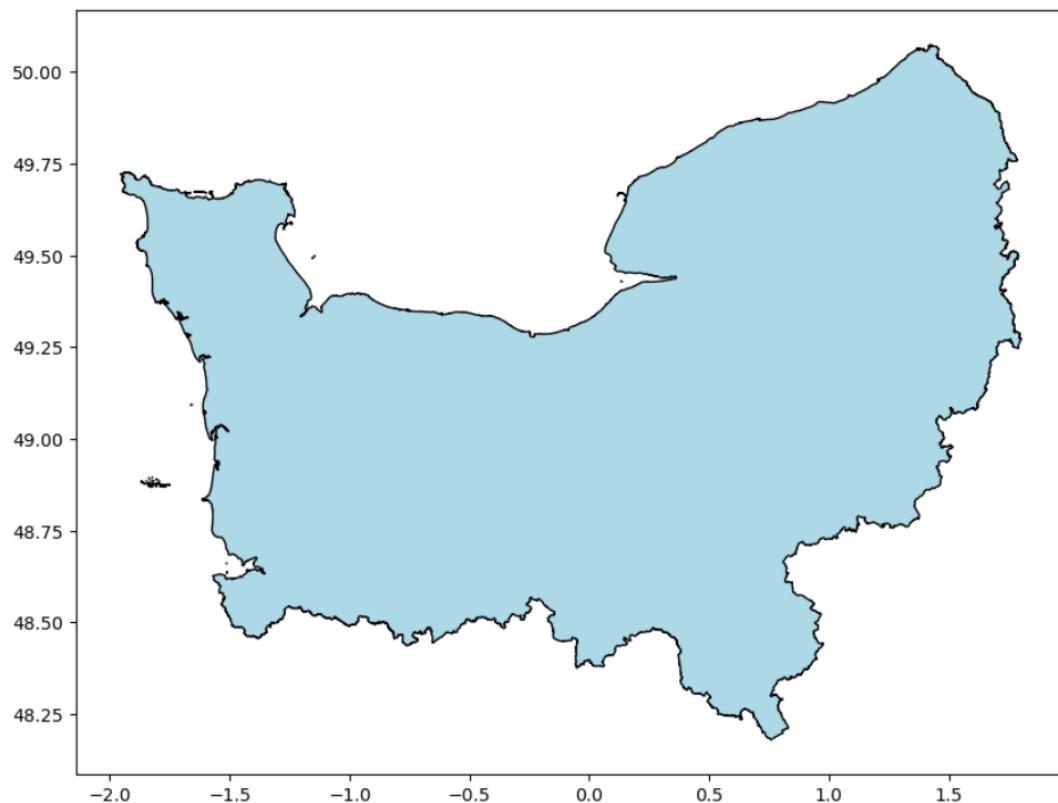


FIGURE 3.1 – Limites administratives de Normandie

La carte générée montre le polygone représentant les limites administratives de la Normandie, avec les contours en noir et un remplissage en bleu clair. Cette visualisation est une première étape cruciale pour avoir une idée claire de la structure géographique de la région et pour préparer des analyses géospatiales plus complexes à venir.

Calcul de la distance des établissements par rapport à la Normandie

Dans cette section, nous allons calculer la distance de chaque établissement par rapport aux limites administratives de la Normandie. Les établissements à l'intérieur de la région auront une distance de 0, tandis que ceux à l'extérieur auront une distance calculée jusqu'aux frontières de la Normandie.

Voici le code complet pour effectuer cette analyse géospatiale :

```
from shapely.geometry import Point  
from shapely.ops import unary_union  
import osmnx as ox
```

```

import geopandas as gpd

# Definir le nom de la rgion
region_name = "Normandie, France"

# Telecharger les limites administratives de la Normandie depuis OpenStreetMap
normandie_gdf = ox.geocode_to_gdf(region_name)

# Nettoyer les donnees et garder les colonnes utiles
normandie_gdf_clean = normandie_gdf[['display_name', 'geometry']].copy()
normandie_gdf_clean = normandie_gdf_clean[normandie_gdf_clean.is_valid]

# Fusionner les polygones de la Normandie en une seule gomtrie
normandie_union = unary_union(normandie_gdf_clean.geometry)

# Supposons que les etablissements sont dans un GeoDataFrame 'etab_gdf_clean'
# Nous devons nous assurer qu'ils sont dans un systeme de coordonnees projetees
etab_gdf_clean = etab_gdf_clean.to_crs(epsg=3857) # Coordonnees projetes
en metres
normandie_gdf_clean = normandie_gdf_clean.to_crs(epsg=3857)

# Verifier si les etablissements sont l'interieur de la Normandie
etab_gdf_clean['inside_normandie'] = etab_gdf_clean.geometry.apply(
    lambda x: x.within(normandie_union)
)

# Calculer la distance pour les tablissements l'extrieur de la Normandie
etab_gdf_clean['distance_to_boundary'] = etab_gdf_clean.geometry.apply(
    lambda x: 0 if x.within(normandie_union) else x.distance(normandie_union)
)

# Afficher les Sorties
print(etab_gdf_clean[['title', 'inside_normandie', 'distance_to_boundary']])
.head()

```

Les sorties montrent les établissements, s'ils sont à l'intérieur ou à l'extérieur de la Normandie, ainsi que la distance par rapport à la frontière de la région. Par exemple :

		title	inside_normandie	distance_to_boundary
0		Clinique du Petit Colmoulin	True	0.0
1	Centre de rééducation des Jonquilles		True	0.0
2	Hôpital privé de l'Estuaire		True	0.0
3	Centre de la Hève		True	0.0
4	Clinique Océane		True	0.0

Les établissements indiqués par True se trouvent à l'intérieur de la Normandie, tandis que

ceux marqués `False` se trouvent à l'extérieur, et la distance est calculée pour ces derniers.

Visualisation des établissements par rapport à la Normandie

Dans cette section, nous visualisons les établissements de santé et leur position par rapport aux limites administratives de la Normandie. Nous utilisons `matplotlib` pour générer une carte sur laquelle les établissements sont distingués selon leur emplacement : à l'intérieur ou à l'extérieur de la Normandie.

Le code suivant génère cette visualisation :

```
import matplotlib.pyplot as plt

# Crer une carte
fig, ax = plt.subplots(figsize=(10, 10))

# Tracer la Normandie
normandie_gdf_clean.plot(ax=ax, color='lightblue', edgecolor='black')

# Tracer les tablissements l'intrieur en vert
etab_gdf_clean[etab_gdf_clean['inside_normandie']].plot(ax=ax, color='green',
markersize=20, label=' l\'intrieur')

# Tracer les tablissements en dehors en rouge
etab_gdf_clean[~etab_gdf_clean['inside_normandie']].plot(ax=ax, color='red',
markersize=20, label=' l\'extrieur')

# Ajouter un titre et une legende
plt.title("Etablissements de sante et position par rapport la Normandie
(verification stricte)")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.legend()
plt.show()
```

Sortie :

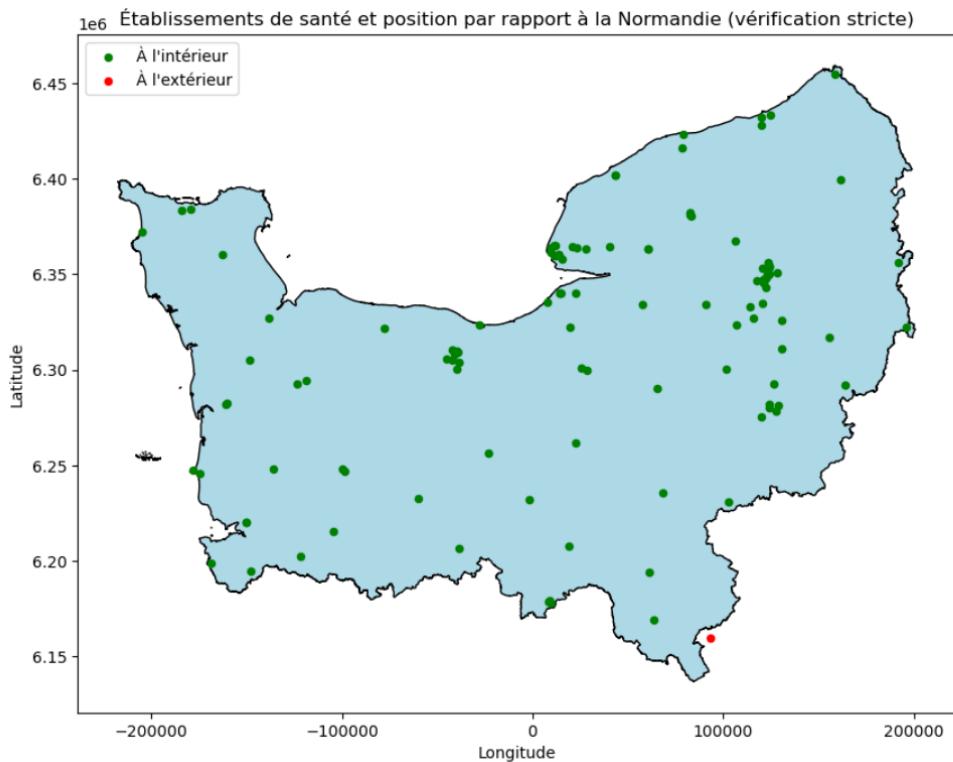


FIGURE 3.2 – Établissements de santé et position par rapport à la Normandie

La carte ci-dessus présente les informations géospatiales des établissements par rapport aux limites de la Normandie. Voici les éléments clés :

- La Normandie est représentée en bleu clair avec des contours noirs, délimitant la région.
- Les établissements situés à l'intérieur de la Normandie sont marqués par des points verts.
- Les établissements situés en dehors de la région sont indiqués par des points rouges.

Longueur totale des routes par catégorie

La première étape consiste à calculer la longueur totale des routes par catégorie (type de route). Nous avons utilisé les données sur les routes pertinentes et effectué les calculs suivants :

```
# Assurer que les gomtries sont projectées en mètres pour un calcul précis
# RGF93 / Lambert 93 (France)
routes_pertinentes = routes_pertinentes.to_crs(epsg=2154)
# Conversion en km
routes_pertinentes['length_km'] = routes_pertinentes.length / 1000

# Somme des longueurs par type de route
longueur_par_type = routes_pertinentes.groupby('fclass')['length_km'].sum()
.reset_index()
```

```
print(longueur_par_type)
```

Cela nous permet de visualiser la longueur totale des routes par catégorie comme suit :

- motorway : 623.01 km
- primary : 2373.41 km
- secondary : 4739.93 km
- tertiary : 13228.60 km

Calcul de la densité routière

La densité routière est calculée en divisant la longueur totale des routes par la superficie de la région. Dans notre cas, la surface de la Normandie est estimée à 29 906 km². La densité routière est donc obtenue avec la formule suivante :

```
# Calcul de la densité routière
densité_routière = routes_pertinentes['length_km'].sum() / superficie_normandie_km2
print(f"Densité routière en Normandie : {densité_routière:.2f} km de routes par km")
```

La densité routière en Normandie est de 0.70 km de routes par km², ce qui nous permet de conclure sur l'état du réseau routier dans la région. Il est possible de comparer cette densité avec celle d'autres régions pour voir si la Normandie dispose d'un réseau dense ou limité. Il est également utile de mettre en évidence des écarts entre les zones urbaines et rurales.

Distance moyenne aux routes principales

Cette analyse consiste à calculer la distance moyenne des établissements de santé aux routes principales. Nous appliquons la fonction suivante pour calculer cette distance.

```
from shapely.ops import nearest_points

# S'assurer que les deux jeux de données sont dans le même CRS
etab_gdf_clean = etab_gdf_clean.to_crs(epsg=2154)
routes_pertinentes = routes_pertinentes.to_crs(epsg=2154)

# Pour trouver la route la plus proche
def distance_route_proche(point, routes):
    nearest_geom = nearest_points(point, routes.unary_union)[1]
    return point.distance(nearest_geom) / 1000 # Conversion en km

# Appliquer la fonction aux établissements
etab_gdf_clean['distance_to_road_km'] = etab_gdf_clean.geometry.
apply(lambda x: distance_route_proche(x, routes_pertinentes))

# Vérification des sorties
print(etab_gdf_clean[['title', 'distance_to_road_km']].head())
```

Les sorties nous montrent la distance en kilomètres entre chaque établissement de santé et la route la plus proche. Par exemple, pour certains établissements de santé :

- Clinique du Petit Colmoulins : 10.76 km
- Centre de rééducation des Jonquilles : 10.40 km
- Hôpital privé de l'Estuaire : 14.09 km
- Centre de la Hève : 13.66 km
- Clinique Océane : 14.03 km

L'analyse des distances montre que certains établissements sont situés à plus de 10 km d'une route principale. C'est notamment le cas de la Clinique du Petit Colmoulins (10,76 km) et de l'Hôpital privé de l'Estuaire (14,09 km).

Ces distances indiquent que les infrastructures médicales ne sont pas toujours facilement accessibles par le réseau routier principal. Pour les patients résidant dans ces zones, cela peut entraîner des délais d'intervention plus longs, en particulier pour les urgences.

Histogramme de la longueur des routes par catégorie

Nous avons tracé un histogramme pour visualiser la longueur des routes par catégorie.

```
import matplotlib.pyplot as plt

# Tracer un histogramme
plt.figure(figsize=(8,5))
plt.bar(longueur_par_type['fclass'], longueur_par_type['length_km'])
plt.xlabel("Type de route")
plt.ylabel("Longueur totale (km)")
plt.title("Longueur des routes principales en Normandie")
plt.xticks(rotation=45)
plt.show()
```

Sortie :

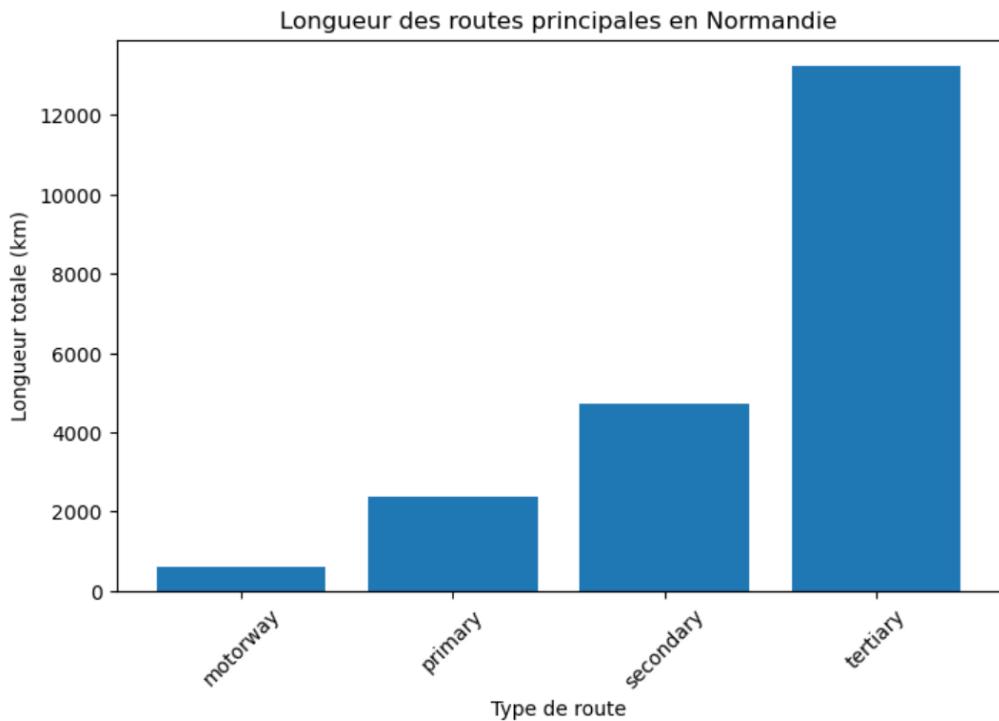


FIGURE 3.3 – Longueur des routes principales en Normandie

L’histogramme montre que la majorité des routes en Normandie sont des routes secondaires (4739,93 km) et tertiaires (13228,60 km), tandis que les autoroutes sont nettement moins nombreuses (623,01 km). Cette répartition est cohérente avec la typologie rurale de la région.

L’impact sur l’accessibilité aux établissements de santé est significatif : les patients vivant dans des communes rurales, souvent éloignées des autoroutes et routes primaires, peuvent être confrontés à des temps de trajet plus longs pour accéder aux soins. Cette disparité justifie l’analyse de la distance des établissements par rapport aux routes principales.

Carte des routes principales et des établissements de santé

Nous avons créé une carte interactive représentant les routes principales et les établissements de santé dans la région. La carte est générée à l’aide de la bibliothèque Folium, qui permet de créer des cartes interactives.

```
import folium

# Centrer la carte sur la Normandie
m = folium.Map(location=[49.5, 0.0], zoom_start=8)

# Ajouter les routes principales
for _, row in routes_pertinentes.iterrows():
    folium.PolyLine(
        list(row.geometry.coords),
```

```

        color="blue",
        weight=2,
        popup=f "Type: {row.fclass}"
    ).add_to(m)

# Ajouter les établissements de santé
for _, row in etab_gdf_clean.iterrows():
    folium.Marker(
        location=[row.geometry.y, row.geometry.x],
        popup=row.title,
        icon=folium.Icon(color="red", icon="plus-sign")
    ).add_to(m)

# Afficher la carte
m

```

Sortie :

La carte suivante présente les routes principales et les établissements de santé en Normandie (exemple de Caen). Les routes sont représentées par des lignes bleues, tandis que les établissements de santé sont signalés par des icônes rouges. En cliquant sur chaque marqueur, le nom de l'établissement s'affiche.

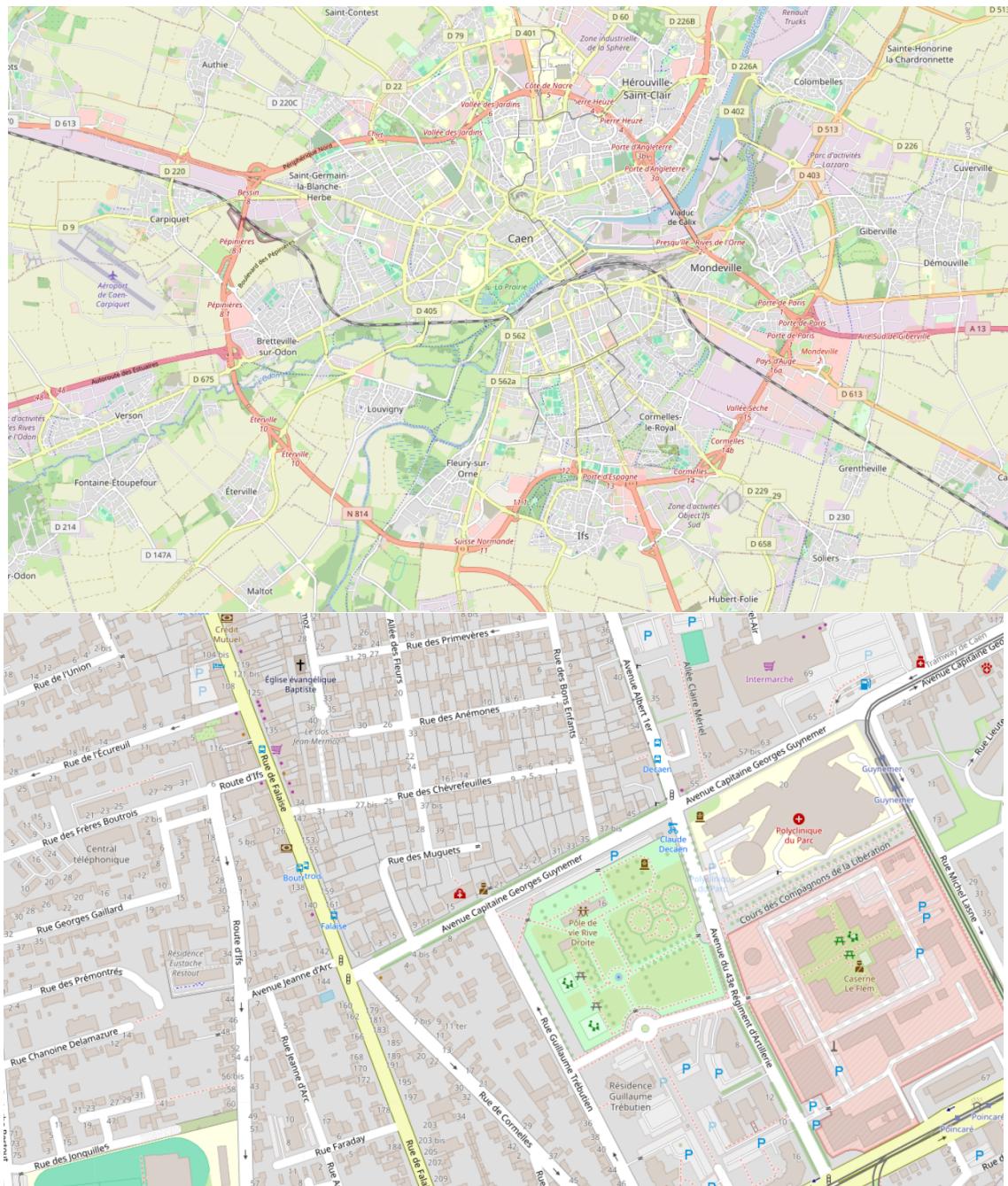


FIGURE 3.4 – Carte interactive des routes principales et des établissements de santé en Normandie (Caen).

La carte interactive met en évidence une concentration des établissements de santé le long des axes principaux, notamment autour de Caen et des autres grandes villes. À l'inverse, certaines zones rurales présentent un déficit de structures médicales accessibles via les routes principales.

Cette répartition confirme un problème d'inégalité territoriale dans l'accès aux soins. Les habitants des zones éloignées des grands axes peuvent avoir des temps de déplacement plus

longs, ce qui peut poser un problème en cas d'urgence.

Une simple visualisation spatiale avec cette carte, elle ne suffit pas pour évaluer précisément l'accessibilité des populations rurales aux soins.

Pour mesurer cette accessibilité, nous avons calculé la distance entre chaque commune rurale et l'établissement de santé le plus proche. Deux approches ont été utilisées :

1. Distance euclidienne (à vol d'oiseau)

La **distance Euclidienne** permet d'obtenir une estimation rapide de l'éloignement théorique d'une zone rurale par rapport au centre de soins le plus proche. Elle met en évidence les communes situées dans des zones éloignées des pôles hospitaliers, mais ne prend pas en compte les contraintes du réseau routier.

```
import geopandas as gpd
from shapely.ops import nearest_points

# Vrifier que les deux jeux de donnees utilisent le mme systme de coordonnes
etab_gdf_clean = etab_gdf_clean.to_crs(epsg=2154) # Lambert 93 (mtres)
routes_pertinentes = routes_pertinentes.to_crs(epsg=2154)

# Fonction pour trouver la route la plus proche
def distance_euclidienne(etablissement, routes):
    nearest_geom = nearest_points(etablissement, routes.geometry.
        unary_union)[1]
    return etablissement.distance(nearest_geom) / 1000 # Conversion en km

# Appliquer la fonction sur chaque tablissement
etab_gdf_clean['distance_route_euclidienne_km'] = etab_gdf_clean.geometry.
apply(lambda x: distance_euclidienne(x, routes_pertinentes))

# Vrifier les Sorties
print(etab_gdf_clean[['title', 'distance_route_euclidienne_km']].head())
```

Sortie :

	title	distance_euclidienne
0	Clinique du Petit Colmoulin	10.763661
1	Centre de rééducation des Jonquilles	10.402275
2	Hôpital privé de l'Estuaire	14.086472
3	Centre de la Hève	13.658367
4	Clinique Océane	14.027331

Ces Sorties montrent que certaines communes sont relativement proches des établissements de santé, mais ces distances ne tiennent pas compte des contraintes réelles du réseau routier.

2. Distance réelle par le réseau routier

L'utilisation des données routières d'OpenStreetMap (OSM) permet d'évaluer la distance effective en empruntant les axes routiers réels. En appliquant des algorithmes de plus court chemin avec **NetworkX**, nous avons déterminé la distance routière entre chaque commune et son hôpital de référence. Cette analyse permet d'identifier des communes situées à proximité relative d'un hôpital mais mal desservies par le réseau routier, ce qui peut limiter leur accessibilité aux soins.

```
import osmnx as ox
import networkx as nx

# Définir la zone d'analyse (Normandie)
place_name = "Normandie, France"

# Télécharger le graphe routier de la région
G = ox.graph_from_place(place_name, network_type="drive")

# Convertir les établissements en coordonnées latitude/longitude
etab_gdf_clean = etab_gdf_clean.to_crs(epsg=4326)

# Trouver le nœud du graphe routier le plus proche pour chaque établissement
etab_gdf_clean['nearest_node'] = etab_gdf_clean.geometry.
apply(lambda x: ox.distance.nearest_nodes(G, X=x.x, Y=x.y))

# Fonction pour calculer la distance routière vers la route la plus proche
def distance_routière(node, roads_nodes, G):
    shortest_distances = [nx.shortest_path_length(G, node, road_node,
weight='length')]
    for road_node in roads_nodes]
    return min(shortest_distances) / 1000 # Conversion en km

# Extraire les nœuds des routes principales
roads_nodes = [ox.distance.nearest_nodes(G, X=row.geometry.x, Y=row.geometry.y)
for _, row in routes_pertinentes.iterrows()]

# Appliquer la fonction sur chaque établissement
etab_gdf_clean['distance_route_routière_km'] = etab_gdf_clean['nearest_node']
.apply(lambda x: distance_routière(x, roads_nodes, G))

# Vérifier les sorties
print(etab_gdf_clean[['title', 'distance_route_routière_km']].head())
```

La distance euclidienne, ou distance à vol d'oiseau, correspond à la mesure théorique la plus courte entre deux points, en ligne droite, sans prendre en compte les obstacles physiques, le relief ou le réseau routier. Elle permet d'obtenir une estimation rapide de l'éloignement

d'un établissement de santé et constitue un indicateur utile pour une première analyse.

Toutefois, cette mesure ne reflète pas la réalité des déplacements, car elle ne considère pas les détours imposés par les routes et les infrastructures, ce qui explique pourquoi elle est souvent plus courte que la distance réelle.

À l'inverse, la distance réelle par le réseau routier intègre la structure effective des routes, en tenant compte des détours, de la densité du réseau et des contraintes infrastructurelles telles que les ponts, tunnels ou routes sinuuses. Calculée à l'aide d'algorithmes de plus court chemin appliqués à des graphes routiers, elle permet d'estimer plus précisément le temps et la difficulté d'accès aux établissements de santé.

Cette approche met en évidence des zones qui, bien que proches géographiquement d'un hôpital, restent difficiles d'accès en raison d'un réseau routier insuffisant.

L'interprétation des Sorties montre que lorsque la distance euclidienne est faible mais que la distance routière est élevée, cela traduit une accessibilité réduite due à des infrastructures routières inadéquates. Ces disparités sont particulièrement marquées en milieu rural, où les voies de circulation sont souvent limitées et sinuuses, rallongeant ainsi le temps d'accès aux soins.

Analyse des vitesses maximales sur les routes

Nous avons analysé les vitesses maximales autorisées sur les routes principales de la Normandie à partir des données d'OpenStreetMap (OSM).

Cette analyse permet de mieux comprendre les contraintes de vitesse auxquelles sont soumises les routes de la région, ce qui peut influencer les temps de trajet et, par conséquent, l'accessibilité aux établissements de santé.

```
import seaborn as sns
# Nettoyer les données en supprimant les valeurs manquantes et non numériques
routes_pertinentes = routes_pertinentes[pd.to_numeric(routes_pertinentes
['maxspeed'], errors='coerce').notna()]
routes_pertinentes['maxspeed'] = routes_pertinentes['maxspeed'].astype(float)

# Tracer une distribution des vitesses
plt.figure(figsize=(8,5))
sns.histplot(routes_pertinentes['maxspeed'], bins=10, kde=True)
plt.xlabel("Vitesse maximale (km/h)")
plt.ylabel("Nombre de routes")
plt.title("Répartition des vitesses maximales sur les routes principales")
plt.show()
```

Sortie :

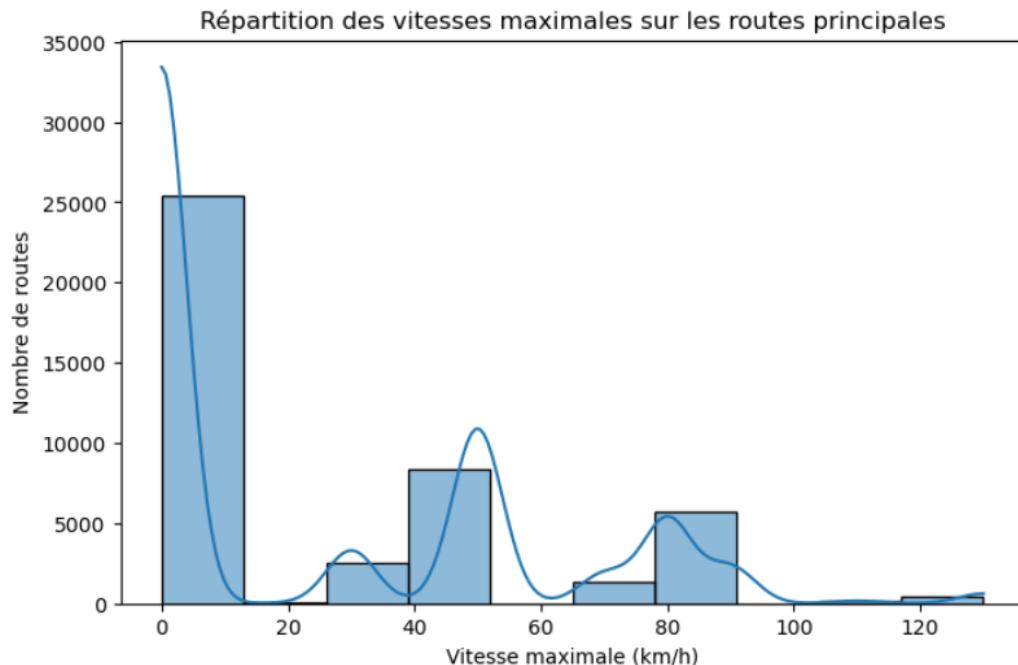


FIGURE 3.5 – Répartition des vitesses maximales sur les routes principales en Normandie.

Le graphique montre la répartition des vitesses maximales autorisées sur les routes principales. La majorité des routes ont une vitesse maximale qui varie entre 0 km/h et 90 km/h, avec une concentration notable autour de 0 km/h. Cela correspond généralement aux routes à très faible vitesse, comme les zones résidentielles, ainsi qu'aux routes secondaires et aux voies express.

La vitesse maximale autorisée sur les routes influence directement le temps de trajet vers les établissements de santé. Une majorité de routes à faible vitesse peut allonger ces trajets, rendant l'accès aux soins plus difficile, notamment en milieu rural.

Les routes limitées à 80-90 km/h permettent généralement un accès plus rapide aux hôpitaux et centres de santé, tandis que celles à très basse vitesse peuvent ralentir considérablement les déplacements, ce qui peut être problématique en cas d'urgence.

Cette analyse met en évidence l'importance d'un bon réseau routier pour garantir un accès équitable aux soins, en particulier dans les zones éloignées des infrastructures médicales.

3.2.4 Analyse géospatiale de données avec R

Avant de réaliser l'analyse des données géospatiales avec R, nous avons d'abord collecté les données et effectué un nettoyage afin d'assurer leur qualité et leur cohérence.

Maintenant, nous utiliserons R pour explorer, visualiser ces données à l'aide des packages suivants :

```
library(sf)
library(mapsf)
library(cartography)
library(tmap)
```

```

library(ggplot2)
library(spdep)
library(dplyr)
library(raster)
library(sp)
library(geojsonsf)

```

Identification des routes les plus proches des établissements de santé

L'analyse de l'accessibilité routière des établissements de santé en Normandie montre que chaque établissement est associé à la route principale la plus proche. En utilisant la fonction `st_nearest_feature()`, nous avons identifié la route la plus proche pour chaque établissement et ajouté son nom dans la colonne `nearest_road`.

```

#Distance routiere routiere reelle
# Trouver la route la plus proche
nearest_road <- st_nearest_feature(etab_sf, routes_pertinentes)
nearest_road
# Ajouter le nom de la route associe
etab_sf$nearest_road <- routes_pertinentes$name[nearest_road]
etab_sf$nearest_road
# Vrifier les correspondances
head(etab_sf[, c("title", "nearest_road")])

```

Sortie :

title	nearest_road
Clinique du Petit Colmoulin	Route de Trouville
Centre de rééducation des Jonquilles	Route de Trouville
Hôpital privé de l'Estuaire	Route de Trouville
Centre de la Hève	Route Cd 513
Clinique Océane	Route de Trouville
Groupe Hospitalier du Havre Hôpital Pierre Janet (Le Havre)	Route de Trouville

TABLEAU 3.4: Association des établissements de santé avec les routes les plus proches

La Route de Trouville apparaît plusieurs fois comme étant la route la plus proche pour plusieurs établissements. Cela pourrait indiquer une concentration des infrastructures de santé le long des principaux axes routiers.

Certains établissements sont proches d'axes secondaires (Route Cd 513). Cela pourrait indiquer que certains hôpitaux ou cliniques sont moins accessibles via le réseau routier principal.

Visualisation de la proximité des établissements de santé aux routes principales par type

Dans cette section, nous avons exploré la relation entre le type d'établissement de santé et sa proximité aux routes principales en utilisant un diagramme en boîte (boxplot).

Ce graphique permet de visualiser la répartition des distances aux routes principales en fonction des catégories d'établissements : publics, privés, privés lucratifs, privés solidaires et privés commerciaux.

Sortie :

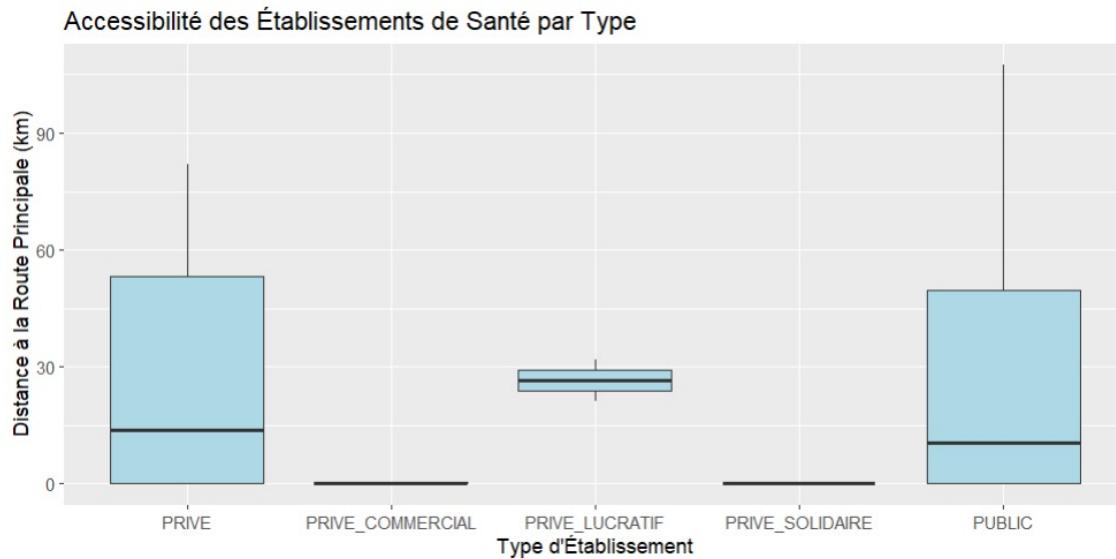


FIGURE 3.6 – Diagramme en boîte illustrant la variabilité des distances aux routes principales en fonction du type d'établissement de santé

L'analyse révèle une différence notable dans l'implantation des établissements de santé selon leur statut juridique.

- **Les établissements publics** couvrent une plus grande diversité de territoires, y compris des zones éloignées, ce qui peut poser des défis d'accessibilité.
- **Les établissements privés**, en particulier lucratifs, sont plus concentrés près des routes principales, suggérant une implantation stratégique dans les zones urbaines ou périurbaines.

Ces sorties soulignent l'importance de l'aménagement du territoire et du développement des infrastructures de transport pour garantir un accès équitable aux soins, notamment pour les populations rurales.

Évaluation de l'autocorrélation spatiale des distances routières

L'étude de l'autocorrélation spatiale permet d'évaluer si les établissements de santé partageant des caractéristiques similaires en termes d'accessibilité sont répartis de manière aléatoire ou s'ils forment des clusters géographiques spécifiques. Pour cela, nous utilisons l'**indice de Moran**, un indicateur clé en analyse spatiale.

```

#Autocorrelation spatiales (indice de Moran )

# les coordonnes des etablissements
coords_etab <- st_coordinates(etab_sf)
# voisinage spatial bas sur les k plus proches voisins (ici, k = 5)
voisinage_etab <- knn2nb(knearneigh(coords_etab, k = 5))

# matrice de poids spatial basee sur les voisinages
matrice_poids_etab <- nb2listw(voisinage_etab, style = "W")

# l'indice de Moran sur la distance aux routes principales
moran_test_etab <- moran.test(etab_sf$distance_route_km, matrice_poids_etab)
moran_test_etab

```

Sortie :

```

Moran I test under randomisation

data: etab_sf$distance_route_km
weights: matrice_poids_etab

Moran I statistic standard deviate = 17.608, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic      Expectation      Variance
0.90300965     -0.00862069     0.00268052

```

FIGURE 3.7 – Indice de Moran

L’indice de Moran calculé sur les distances des établissements de santé aux routes principales met en évidence une forte autocorrélation spatiale positive (Moran’s I = 0.903, p-value < 2.2e-16).

Ce Sortie signifie que les établissements ayant des distances similaires aux infrastructures routières ont tendance à être géographiquement regroupés, plutôt que répartis de manière aléatoire.

En d’autres termes, les établissements bien desservis par le réseau routier se concentrent dans certaines zones urbaines ou périurbaines, tandis que ceux plus éloignés des axes de transport se retrouvent également regroupés, souvent dans des zones rurales sous-desservies.

Cette configuration spatiale suggère des inégalités d’accessibilité aux soins de santé, certaines régions bénéficiant d’une infrastructure routière dense, tandis que d’autres, plus isolées, peuvent rencontrer des difficultés pour accéder rapidement aux services médicaux.

3.3 Interprétation des sorties

L'étude menée sur l'accessibilité aux établissements de santé en Normandie a permis d'identifier plusieurs tendances clés en croisant données géospatiales, analyses statistiques et visualisations cartographiques. L'utilisation combinée de R et Python nous a fourni une vision détaillée des infrastructures médicales et de leur accessibilité en fonction du réseau routier et de la distribution spatiale des communes rurales.

3.3.1 Disparités spatiales dans l'accessibilité aux soins

L'un des Sorties majeurs de notre analyse est la forte disparité entre les zones urbaines et rurales en termes d'accès aux établissements de santé. D'une part, les grandes villes comme Caen et Rouen concentrent la majorité des hôpitaux et des cliniques, offrant ainsi un accès rapide aux soins pour les habitants de ces zones.

D'autre part, les zones rurales, notamment dans l'ouest de la région et dans les départements de l'Orne et de la Manche, affichent un déficit important en infrastructures médicales. L'un des facteurs clés expliquant cette disparité est la densité de population : les hôpitaux sont principalement situés dans les zones à forte concentration démographique, laissant certaines communes rurales isolées et éloignées des structures de soins.

Cette répartition inégale impose aux patients vivant en milieu rural des déplacements plus longs pour accéder aux soins spécialisés, ce qui peut poser un problème en cas d'urgence médicale.

3.3.2 Influence du réseau routier sur l'accessibilité aux soins

L'analyse des infrastructures routières a révélé que les routes principales jouent un rôle central dans l'accessibilité aux établissements de santé. Nos Sorties montrent que les hôpitaux situés à proximité des autoroutes et des routes nationales sont plus accessibles, permettant un temps de trajet réduit et une meilleure couverture territoriale.

À l'inverse, les établissements situés en zones rurales sont souvent reliés uniquement par des routes secondaires ou tertiaires, ce qui rallonge les trajets et peut compliquer l'intervention des services d'urgence. Les calculs de distances entre établissements de santé et routes principales ont mis en évidence que plusieurs communes rurales se situent à plus de 15 km d'un axe routier majeur, rendant l'accès aux soins plus difficile pour ces populations isolées.

Ces sorties confirment donc que le réseau routier est un facteur déterminant dans l'accessibilité aux hôpitaux. Une amélioration des infrastructures de transport pourrait réduire les inégalités territoriales et faciliter l'accès aux soins pour les habitants des zones les plus éloignées.

3.3.3 Temps de trajet réel et accessibilité médicale

Grâce aux outils Python (`osmnx` et `networkx`), nous avons pu aller au-delà des distances à vol d'oiseau en calculant les temps de trajet réels en fonction du réseau routier existant. Les sorties obtenues confirment que certaines zones rurales nécessitent plus de 30 minutes de trajet pour atteindre un hôpital, un délai critique dans le cadre d'une urgence médicale.

De plus, les distances réelles sont souvent plus élevées que les distances théoriques en raison de la qualité et du type des routes disponibles (routes départementales, chemins secondaires). Une observation notable est que les hôpitaux publics sont souvent plus éloignés que les établissements privés en milieu rural, ce qui pose un problème d'accessibilité pour les populations à faible mobilité ou aux revenus limités.

Cette analyse démontre l'importance d'inclure les temps de trajet routiers dans les études d'accessibilité aux soins, plutôt que de se baser uniquement sur des distances en ligne droite.

3.3.4 Cartographie et identification des zones vulnérables

Les différentes cartes produites avec R (`cartography`, `ggplot2`) et Python (`folium`) ont permis de visualiser les zones les plus défavorisées en termes d'accessibilité aux soins.

- Les cartes choroplèthes ont mis en évidence les zones de désert médical où la distance aux établissements de santé est particulièrement élevée.
- Les cartes interactives ont permis d'explorer dynamiquement les distances entre établissements et routes principales, facilitant l'identification des communes où de nouveaux centres de soins pourraient être implantés.
- L'analyse spatiale a également révélé des incohérences, comme certaines communes à densité relativement élevée mais dépourvues de structures hospitalières suffisantes.

Ces cartes sont des outils précieux pour les décideurs locaux, car elles permettent une prise de décision éclairée sur l'aménagement du territoire et la répartition des services de santé.

5. Implications et pistes d'amélioration

L'étude met en lumière plusieurs leviers d'amélioration pour réduire les inégalités d'accès aux soins en milieu rural :

Renforcement des infrastructures médicales en zones rurales L'ouverture de centres de soins intermédiaires ou de maisons de santé dans certaines communes isolées permettrait de réduire les distances d'accès et d'assurer une meilleure couverture médicale.

Amélioration du réseau de transport Investir dans les routes départementales reliant les communes rurales aux hôpitaux les plus proches permettrait de réduire les temps de trajet et d'optimiser l'intervention des services d'urgence.

Mise en place de solutions alternatives Le développement de la télémédecine et des services de transport médicalisé pourrait compenser le manque d'hôpitaux physiques en zones rurales.

Optimisation des structures existantes L'analyse montre que certains hôpitaux pourraient améliorer leur capacité d'accueil et leur organisation pour mieux couvrir les territoires éloignés.

3.4 Comparaison entre Python et R

Travailler avec R et Python sur cette analyse a permis d'explorer différentes approches et de tirer parti des forces de chaque outil. L'un s'est révélé particulièrement performant pour la visualisation et l'analyse statistique des inégalités territoriales, tandis que l'autre a offert une meilleure flexibilité pour modéliser les trajets et estimer les temps d'accès réels aux établissements de santé.

Avec R, la manipulation des données géospatiales et la création de cartes thématiques précises ont été fluides. Il a facilité l'identification des zones mal desservies et permis de quantifier les écarts d'accessibilité entre les différentes communes. Les représentations obtenues ont clairement mis en évidence les disparités entre les zones urbaines bien équipées et les secteurs ruraux où l'accès aux soins reste problématique.

De son côté, Python a apporté une approche plus dynamique en permettant de calculer les temps de trajet réels en fonction du réseau routier. Contrairement aux distances en ligne droite souvent utilisées dans R, cette méthode a montré que certaines zones, pourtant proches d'un hôpital en apparence, nécessitent des trajets bien plus longs en raison de la structure des routes et des axes de transport. Ce constat a renforcé l'importance de prendre en compte l'accessibilité réelle et non uniquement la proximité géographique.

Quelques difficultés sont apparues au moment d'intégrer les données entre les deux outils. R était idéal pour produire des analyses statistiques et des cartes statiques détaillées, mais Python s'est montré plus efficace pour explorer les données de manière interactive et automatiser certaines tâches. La création de cartes interactives a également été plus intuitive avec Python, permettant d'affiner l'analyse en visualisant directement les connexions entre les infrastructures hospitalières et le réseau routier.

Au final, R et Python ont été complémentaires. L'un a permis d'étudier les disparités spatiales et de quantifier l'accessibilité aux soins, tandis que l'autre a offert une approche plus réaliste des trajets et des temps d'accès. Cette combinaison d'outils a apporté une vision plus précise et plus nuancée de l'accessibilité des établissements de santé en Normandie.

Bibliographie

- [Cas16] Nicholas CASAJUS. *Analyses Spatiales Sous R.* Rapp. tech. Document académique. Université du Québec à Rimouski, Département de Géographie, 2016. URL : <https://www.scribd.com/document/335963072/Analyses-Spatiales-Sous-R>.
- [Gal24a] Lino GALIANA. *De belles cartes avec Python : mise en pratique.* 2024. URL : <https://pythonds.linogaliana.fr/content/visualisation/maps.html>.
- [Gal24b] Lino GALIANA. *Manipuler des données spatiales avec GeoPandas.* 2024. URL : https://pythonds.linogaliana.fr/content/manipulation/03_geopandas_intro.html.
- [Gir16] Timothée GIRAUD. *Cartographie avec R.* 2016. URL : https://osgeo-fr.github.io/presentations_foss4gfr/2016/J1/R_Cartography_T_Giraud_FOSS4G-fr-2016/FOSS4G-fr-2016.html.
- [Ins23] INSTITUT NATIONAL DE LA STATISTIQUE ET DES ÉTUDES ÉCONOMIQUES (INSEE). *Populations légales 2021 - Normandie.* Données de population pour l'analyse. 2023. URL : https://www.insee.fr/fr/statistiques/fichier/7734153/Popleg2021_NORMANDIE.pdf.
- [Ins24a] INSTITUT NATIONAL DE L'INFORMATION GÉOGRAPHIQUE ET FORESTIÈRE (IGN). *BD TOPO®.* Réseau routier utilisées pour l'analyse. 2024. URL : <https://geoservices.ign.fr/bdtopo>.
- [Ins24b] INSTITUT NATIONAL DE LA STATISTIQUE ET DES ÉTUDES ÉCONOMIQUES (INSEE). *Dénombrement et géolocalisation des équipements en 2023 (commerce, services, sport, santé...)* Base permanente des équipements (BPE) 2023. 2024. URL : <https://www.insee.fr/fr/statistiques/8217525?sommaire=8217537>.
- [Jol17] François-Xavier JOLLOIS. *TP9 - Cartographie sous python.* 2017. URL : <https://fxjollois.github.io/cours-2016-2017/analyse-donnees-massives-tp9.html>.
- [LNM19] Robin LOVELACE, Jakub NOWOSAD et Jannes MUENCHOW. *Geocomputation with R.* CRC Press, 2019. URL : <https://r.geocompx.org/>.

-
- [LNM24] Robin LOVELACE, Jakub NOWOSAD et Jannes MUENCHOW. *Geocomputation avec R*. Chapitre : La donnée géographique dans R. CRC Press, 2024. URL : <https://r.geocompx.org/fr/spatial-class.html>.
 - [Mar20] Philippe MARCHAND. *Introduction au traitement des données géospatiales avec R*. 2020. URL : https://pmarchand1.github.io/atelier_rgeo/atelier_rgeo.html.
 - [Ope24a] OPEN DATA NORMANDIE. *Établissements hospitaliers en Normandie*. Données des établissements utilisées pour l'analyse. 2024. URL : https://opendata.normandie.fr/datasets/7a86241202f046d2a12bb70b30662fde_0.
 - [Ope24b] OPENSTREETMAP FRANCE. *Découpage administratif communal français issu d'OpenStreetMap*. Découpage utilisées pour l'analyse. 2024. URL : <https://www.data.gouv.fr/fr/datasets/dcoupage-administratif-communal-francais-issu-d-openstreetmap/>.
 - [RAW23] Sergio J. REY, Dani ARRIBAS-BEL et Levi J. WOLF. *Geographic Data Science with Python*. Taylor Francis Group, 2023. URL : <https://geographicdata.science/book/intro.html>.
 - [Roe19] Nicolas ROELANDT. *Données géospatiales et cartographie avec R*. 2019. URL : https://roelandtn.frama.io/slides/2090628_meetup_Raddict_datageo.html.
 - [Sal20a] Elena SALETTE. *Cartographie interactive avec leaflet*. 2020. URL : <https://thinkr.fr/cartographie-interactive-comment-visualiser-mes-donnees-spatiales-de-maniere-dynamique-avec-leaflet/>.
 - [Sal20b] Elena SALETTE. *Cartographie interactive avec R*. 2020. URL : <https://thinkr.fr/cartographie-interactive-avec-r-la-suite/>.
 - [Wik24] WIKIPÉDIA CONTRIBUTORS. *OpenStreetMap*. Données de transport. 2024. URL : <https://fr.wikipedia.org/wiki/OpenStreetMap>.