

A thick dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the word 'Optimisation'. In the bottom left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

Optimisation

# Cahier de laboratoire : analyse comparative des réseaux routiers

*Génération d'un plan de navigation à partir d'une  
scène #PFE23-R-198*

Auteurs : Alexis MARIE, David MARCHES, Luca BANKOFSKI, Martial BROSTIN, Théo HELLER, Thomas MABILLE

## Sommaire

Sommaire .....	1
Récupération et traitement des fichiers OSM d'OpenStreetMap .....	2
Analyse de la corrélation entre la densité du réseau routier et la taille de la ville .....	4
Comparaison de la construction intrinsèque des villes en fonction de leurs tailles .....	7
Vérification de la proportion de routes à sens unique .....	9
Conclusion globale .....	10
Références .....	11

# Récupération et traitement des fichiers OSM d'OpenStreetMap

## Objectif de l'expérience

Le but de cette expérience est dans un premier temps de récupérer les fichiers OSM d'OpenStreetMap afin de les nettoyer et de générer les graphes représentant le réseau routier des villes. Dans un second temps, nous voulons générer deux fichiers CSV qui nous serviront à régénérer ces graphes pour nos simulations et un troisième fichier CSV qui nous servira pour l'analyse comparative des réseaux routiers.

## Protocole expérimental

Dans un premier temps, il est nécessaire d'aller sur OpenStreetMap et de télécharger les fichiers OSM des villes choisies. Pour cela, il suffit d'aller sur OpenStreetMap [1], de sélectionner la zone géographique voulue, puis de cliquer sur "Exporter".

Une fois les fichiers OSM téléchargés, il faut les traiter avec un programme Python pour ne récupérer que le nécessaire. Le but est de récupérer, pour chaque ville, les sommets du réseau routier ainsi que les routes, donc les arcs qui formeront notre graphe orienté. Pour ce faire, nous allons effectuer plusieurs étapes pour nettoyer nos données :

- Supprimer tous les sommets qui ne font pas partie du réseau routier ou qui sont isolés ;
- Supprimer toutes les voies piétonnes ou isolées pour ne garder que le réseau routier praticable en voiture ;
- Calculer la distance entre deux sommets pour chaque arc du graphe grâce à la formule haversienne [2] pour prendre en compte la courbure de la Terre ;
- Simplifier le graphe en supprimant tous les sommets ayant exactement deux voisins et dont les deux routes reliées à ces voisins ont les mêmes caractéristiques (même vitesse maximale et même type de routes sens unique ou double sens). La nouvelle route résultant de la suppression de ce sommet aura comme distance la somme des distances des deux routes précédentes.

Une fois ce nettoyage effectué, nous récupérerons pour les graphes de chaque ville les données suivantes avec le même programme Python que nous enregistrons dans un fichier CSV nommé "graph\_info.csv" :

- Le nom de la ville ;
- Le nombre de sommets ;
- Le nombre d'arcs ;
- Le nombre d'arcs à sens unique ;
- Le nombre d'arcs à double sens ;
- Le nombre d'arcs qui n'ont pas de spécifications sur la vitesse maximale ;
- La vitesse moyenne sur l'ensemble du réseau.

Nous allons aussi enregistrer les deux fichiers CSV suivants, permettant de reconstruire nos graphes orientés traités :

Fichier CSV pour les sommets :

- Identifiant du sommet ;

- Latitude du sommet ;
- Longitude du sommet.

Fichier CSV pour les arcs :

- Identifiant du sommet 1 ;
- Identifiant du sommet 2 ;
- Vitesse maximale sur cette portion ;
- Type de voie (sens unique ou double sens) ;
- Longueur de la voie.

## Données

Les différents fichiers CSV générés sont disponibles sur le git du projet :

1. Les CSV des arcs et des sommets de nos 51 villes [3]
2. Le CSV contenant les informations spécifiques de chaque ville "graph\_info.csv" [4]

## Analyses préliminaires

A cette étape, nous n'avons pas encore commencé l'analyse des données et nous avons seulement récupéré tout ce qui nous sera nécessaire à la fois pour analyser nos réseaux routiers mais aussi pour pouvoir recréer nos graphes à partir des CSV.

## Conclusion

Cette première partie de récupération des données s'est bien passée et nous a permis de récupérer des informations précieuses sur 51 villes choisies arbitrairement. Pour la suite de cette étude préliminaire, il va falloir analyser plus en détail les données obtenues dans le CSV "graph\_info.csv".

# Analyse de la corrélation entre la densité du réseau routier et la taille de la ville

## Objectif de l'expérience

L'objectif de cette expérience est de trouver s'il y a corrélation entre la densité du réseau routier d'une ville et sa taille.

## Protocole expérimental

A partir du fichier CSV "graph\_info.csv" [4] créé précédemment, nous pouvons commencer notre étude visant à déterminer quels paramètres du réseau routier varient d'une ville à l'autre.

Pour chaque ville, nous pouvons maintenant calculer la densité du graphe orienté avec la formule suivante :

$$Densité = \frac{\text{Nombre de routes}}{\text{Nombre total de routes possibles}} = \frac{\text{Nombre de routes}}{\text{Nombre de sommets}} \times \frac{1}{\text{Nombre de sommets} - 1}$$

Cette formule va nous aider à déterminer ce qu'il faut chercher pour confirmer notre hypothèse concernant la corrélation entre la densité et la taille des villes. Pour cela, il va falloir vérifier deux choses. D'une part, il va falloir vérifier si le nombre de routes par nombre de sommets dans nos villes peut être considéré comme une constante (En effet, qu'elle soit grande ou petite, les intersections dans une ville ne sont pas extensibles et sont souvent l'intersection de 3 ou 4 routes). Cela nous permettra d'écrire notre formule précédente sous cette forme avec C une constante :

$$Densité = \frac{C}{\text{Nombre de sommets} - 1}$$

D'autre part, si cette information est vraie, il faudra vérifier qu'il y a bien une régression linéaire de la densité en fonction du nombre de sommets d'une ville de la forme  $y = \frac{C}{x}$  avec C une constante.

On va donc utiliser un jupyter notebook pour récupérer le CSV "graph\_info.csv" que nous allons convertir en dataset. Nous allons ensuite créer une colonne "edges/nodes" qui représentera le nombre de route moyenne que chaque intersection possède dans une ville. Pour cela on n'oublie pas de compter les routes à double sens en double et d'y ajouter le nombre de routes à sens unique car nous travaillons avec des graphes orientés.

On va ensuite analyser la moyenne et l'écart type de cette nouvelle colonne, en vérifiant que celle-ci soit cohérente avec un faible écart-type. On peut aussi tracer une boîte à moustache pour visualiser ces résultats.

Dans un second temps nous allons créer une colonne "density" qui représentera la densité de chaque graphe généré et ce, en utilisant la formule présentée précédemment. On essaiera ensuite de trouver une régression entre la colonne "density" et la colonne "nodes\_number" grâce à des méthodes de machine learning.

## Observation

On remarque lorsque l'on effectue une régression linéaire entre la colonne "density" et "nodes\_number" qu'il est difficile de trouver une corrélation. Cependant en prenant le logarithme de ces deux colonnes on trouve une régression linéaire presque parfaite ce qui implique qu'il existe une corrélation puissance entre ces deux variables.

## Données et analyses préliminaires

Les données complètes de cette expérience sont disponibles sur le git de notre projet [5].

En effet, on remarque qu'il y a bien une constante entre le nombre de routes divisé par le nombre d'intersections dans une ville comme le montre ce schéma:

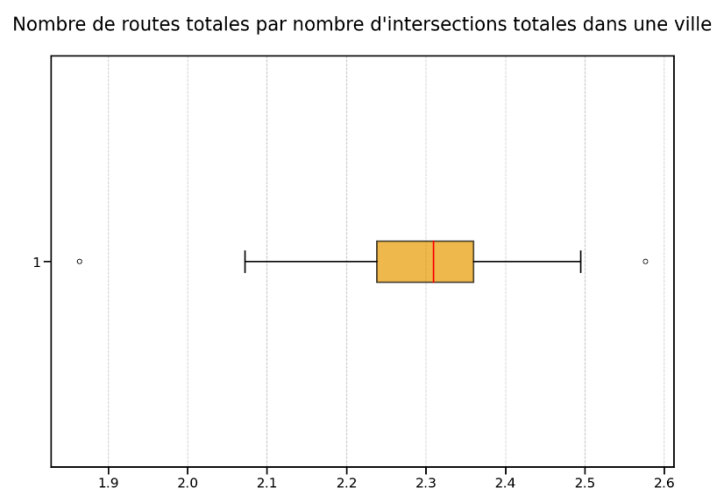


Figure 1 - Proportion du nombre de routes par nombre d'intersections dans une ville.

Sans enlever les valeurs aberrantes, on trouve une moyenne de 2.29 et un écart-type de 0.1

Cela signifie qu'on peut bel et bien réécrire la formule de la densité sous cette forme :

$$Densité = \frac{C}{Nombre\ de\ sommets - 1}$$

On devrait donc trouver qu'il y a bien une régression linéaire de la densité en fonction du nombre de sommets d'une ville de la forme  $y = \frac{const}{x}$  et c'est en effet ce que l'on vérifie dans nos résultats :

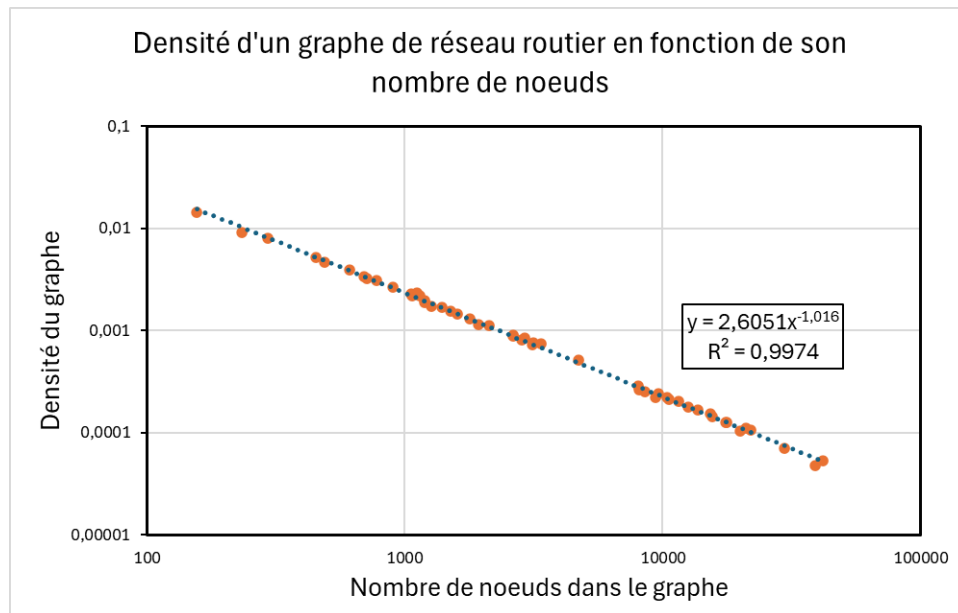


Figure 2 - Densité d'un graphe de réseau routier en fonction de son nombre de nœuds.

Dans ce schéma, on remarque que notre  $R^2$  vaut presque 1 et est donc presque parfait. Nos résultats sont donc très pertinents. Pour les valeurs de la régression on trouve bien une constante divisée par  $x$  qui représente ici notre nombre de nœuds ce qui finit de vérifier nos attendus.

## Conclusion

On remarque que l'on a bien une relation très forte entre la densité d'un graphe de réseau routier de ville et son nombre de nœuds ce qui va dans le sens de notre hypothèse formulé dans le protocole expérimental à savoir que la taille d'une ville n'a pas d'influence significative sur la structure intrinsèque de son réseau routier. Cependant, pour le vérifier complètement, il faut encore faire une expérience sur la construction de nos réseaux routiers.

# Comparaison de la construction intrinsèque des villes en fonction de leurs tailles

## Objectif de l'expérience

L'objectif de cette expérience est de vérifier que nos graphes de réseau routier sont bien tous construits de la même façon, peu importe leur taille. Pour cela, nous allons cette fois-ci analyser la probabilité qu'une intersection soit une intersection à 2, 3, 4 ou plus de voies.

## Protocole expérimental

Pour cette expérience nous reprendrons les 2 CSV créés précédemment, permettant de recréer nos graphes orientés. Nous allons programmer une fonction qui aura simplement pour but de compter combien de voisins aura chaque nœud de notre ville.

Une moyenne sera ensuite faite pour chaque ville et nous donnera une distribution de probabilité qu'a une intersection d'être le centre de 1, 2, 3 ou plus de voies.

Nous comparerons ensuite cette distribution entre chaque ville pour s'assurer qu'elle est similaire peu importe sa taille.

## Observation

Nous remarquons quelque chose de peu intuitif dans nos résultats mais l'intersection la plus courante dans nos villes est l'intersection à 3 voies et non pas 4 comme on pourrait le penser.

## Données et analyses préliminaires

Les données de cette expérience sont disponibles sur le git de notre projet [5].

On peut tout de même constater sur le graphique Figure 3 qui représente les 51 distributions du nombre de voisins, qu'elles ont toutes la même allure ce qui est confirmé par nos calculs.



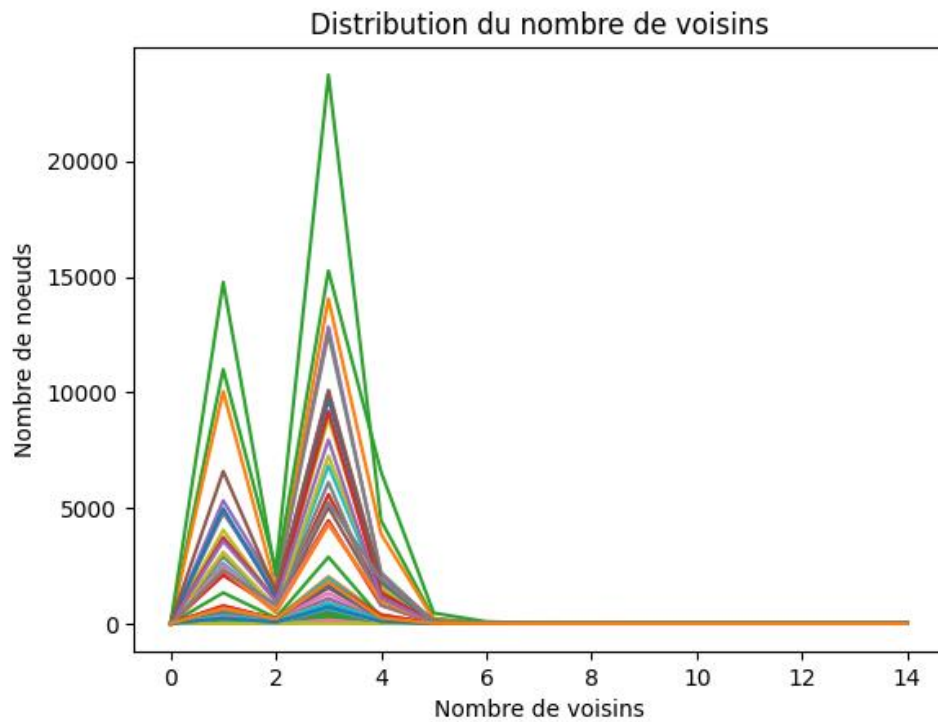


Figure 3 - Allure de la distribution du nombre de voisins de chaque nœud de nos 51 villes.

Cela veut dire que peu importe la taille, la construction intrinsèque de chaque ville est la même.

## Conclusion

Nous avons donc pu voir à travers ces deux dernières expériences que l'on peut étudier nos villes indépendamment de leur taille puisque celle-ci n'influence en aucun cas la construction intrinsèque de leur réseau routier. C'est un résultat majeur qui nous aidera à gagner beaucoup de temps dans la suite de nos simulations et nous permettra de nous pencher sur des variables plus pertinentes comme par exemple la proportion de route à sens unique dans le réseau routier.

# Vérification de la proportion de routes à sens unique

## Objectif de l'expérience

Dans cette expérience nous voulons savoir si nous devrions considérer la proportion de routes à sens unique comme une variable fixe ou non dans la suite de notre étude.

## Protocole expérimental

Pour vérifier si la proportion de voies à sens unique varie d'une ville à l'autre, il suffit de calculer la proportion de voies à sens unique grâce à notre fichier CSV "graph\_info.csv" généré précédemment [4] et de réaliser une étude statistique en examinant la moyenne et l'écart type de cette proportion.

## Données et analyses préliminaires

Les données de cette expérience sont disponibles sur le git de notre projet [5].

En faisant l'étude statistique sur nos 51 villes, on remarque que la moyenne de la proportion de ces routes est de 20%. Cependant l'écart-type est assez grand et nous indique que 95% des valeurs sont situées entre 4% et 36%. On ne peut donc pas vraiment considérer que cette variable soit fixe. Il faudra donc bel et bien la prendre en compte dans la suite de notre étude.

count	51.000000
mean	0.198368
std	0.084173
min	0.026207
25%	0.147751
50%	0.196327
75%	0.245917
max	0.476053

Figure 4 - Résultat de notre étude statistique sur la proportion de routes à sens uniques dans nos villes.

## Conclusion

Nous avons donc mis en lumière la nécessité de prendre en compte la proportion de routes à sens unique dans la suite de notre étude puisque contrairement à la taille des villes, cette variable pourrait fortement influencer nos résultats.

## Conclusion globale

Nous avons donc validé, grâce à ces expériences, les 3 hypothèses formulées dans le protocole expérimentale *PFE23-R-198\_PE\_Optimisation\_Génération-graphes*.

D'une part, nous avons démontré que nous pouvions analyser nos villes indépendamment de leur taille. D'autre part, nous avons remarqué qu'une variable importante à prendre en compte serait la proportion de routes à sens unique puisque cette variable n'est pas fixe et pourrait avoir une grande influence sur les résultats de nos simulations.

Dans le document qui a servi à calculer tous les résultats de cette expérience [4] vous pourrez aussi retrouver d'autres résultats statistiques sur nos graphes générés qui n'ont pas ou peu été utilisé pour la suite du projet.

## Références

[1] <https://www.openstreetmap.org/export>

[2] <https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-a-sphere-the-haversine-formula/ba-p/902128>

[3] [https://github.com/alexismarieece/PFE\\_RECHERCHE\\_VOITURE/tree/theo/Sources/map](https://github.com/alexismarieece/PFE_RECHERCHE_VOITURE/tree/theo/Sources/map)

[4] [https://github.com/alexismarieece/PFE\\_RECHERCHE\\_VOITURE/tree/theo/Sources](https://github.com/alexismarieece/PFE_RECHERCHE_VOITURE/tree/theo/Sources)

[5] [https://github.com/alexismarieece/PFE\\_RECHERCHE\\_VOITURE/blob/theo/Parsing\\_graph/graph\\_density\\_analysys.ipynb](https://github.com/alexismarieece/PFE_RECHERCHE_VOITURE/blob/theo/Parsing_graph/graph_density_analysys.ipynb)