Note méthodologique :

4 étapes :

- Récupérer les données (format GTFS sur https://transport.data.gouv.fr/datasets/region/2?order_by=most_recent)
- Lire les données en python
- Traiter les données avec la méthodologie de l'indicateur
- (Cartographier les données)

Récupérer les données :

- Le format GTFS est un standard non officiel développé par Google spécialisé pour les transports en commun qui contient les données géographiques et horaires des transports.
- Le set *OURA* semble contenir toutes les informations des transports (routier et rail, urbain et interurbains) de la région : https://transport.data.gouv.fr/datasets/agregat-oura/.

Lire les données en python :

- Module *gtfs-kit* : https://openbase.com/python/gtfs-kit - ou *mzgtfs* : https://github.com/transitland/mapzen-gtfs

Traiter les données :

- Organiser les données pour faciliter la détection des arrêts proches : pour chaque ligne, associer un rectangle qui contient tous les arrêts la ligne et stocker ces rectangles dans un *csv*. Pour chaque point (X,Y) il suffit alors d'évaluer la distance du point au rectangle pour éliminer des lignes trop éloignées : cela permet de ne pas avoir à parcourir l'ensemble des arrêts à chaque étape mais seulement l'ensemble des lignes puis de regarder les arrêts des lignes susceptibles d'être assez proches.
- Récupérations des arrêts les plus proches pour chaque ligne en parcourant les sommets (plus proche arrêt retenu ssi il est dans la zone).
- Récupération du temps d'attente moyen à l'arrêt : 0,5 x (60 / nb d'arrêts durant l'heure de pointe)
- Somme temps de marche + temps d'attente = temps d'accès total puis tranformation en fréquence avec EDF = 30 / tps d'accès moyen
- Indicateur = EDF_max + 0.5 x Sum(EDF). *Remarque* : on pourrait prendre en compte l'angle entre les lignes (par exemple les terminus) pour adapter le coefficient 0.5 (si on a deux lignes parallèles la ligne secondaire est peu utile tandis que pour deux lignes perpendiculaires elles ne sont pas du tout en concurrence).
- Stockage (*csv*) de l'indicateur ou de la catégorie associée

Cartographier les données

- En découpant le territoire en carrés de centres (x,y) on peut alors cartographier l'accessibilité du territoire avec un code couleur (en python par exemple avec *Geopandas*).