

Enedis Lab Bretagne

Data Challenge 2023

Résumé

En partenariat avec les universités Rennes 1 et Rennes 2, l'Enedis Lab Bretagne a décidé de partager des cas d'usages nécessitant une analyse poussée des données de l'entreprise afin de répondre à des problématiques concrètes. Cette démarche s'inscrit dans une volonté de rapprochement avec l'écosystème universitaire local et d'engagement auprès des étudiants.

Cette année, deux sujets sont proposés: le premier axé sur l'optimisation du positionnement des bases opérationnelles d'Enedis en Bretagne et le second sur la recherche de zones d'implantation de bornes de recharge de véhicules électriques.

L'objectif de ce document est d'amener le plus de précisions possible sur les deux sujets proposés et les données associées.

Avertissement

Par mesure de sécurité et dans un but de respect de la législation en vigueur, toute donnée qui sera partagée aux étudiants participant au challenge aura fait l'objet d'une anonymisation en amont.

Il sera également demandé aux étudiants de ne pas conserver de copie des jeux de données fournis par Enedis après la fin du challenge.

Accessibilité			
▼ Libre	Interne	Restreinte	☐ Confidentielle

SOMMAIRE

1 — Prérequis techniques	3
1.1. Bibliothèques utiles	3
1.2. Format de restitution	3
2 — Sujet n°1	
2.1. Les données	
2.1.1. Temps de trajet	
2.1.2. Niveau d'interventions	4
2.1.3. Liste des communes	
2.2. Aides au code	5
2.2.1. Obtenir un temps de trajet depuis les données fournies	5
2.2.2. Obtenir un temps de trajet absent des données fournies	6
3 — Sujet n°2	7
3.1. Les données	7
3.1.1. Capacités du réseau basse tension projetées sur des portions de route	
3.1.2. Trafic journalier sur les routes nationales du département	8
3.1.3. Densité de population par commune et évolution depuis 1968	8
3.1.4. Positions des bornes de recharge déjà implantées sur le domaine public	
3.2. Aides au code	
3.2.1. Importer des données au format geojson ou shape	9
3.2.2. Calcul de distance entre deux polygones	10



1 — Prérequis techniques

En accord avec les professeurs des deux universités, les deux langages de programmation autorisés pour résoudre les problématiques proposées sont Python et R. Toute autre technologie utilisée ne fera donc pas l'objet d'une évaluation.

1.1. Bibliothèques utiles



Python

Packages	Utilisation	Documentation
Pandas	Charger et manipuler les données	https://pandas.pydata.org/docs/
Requests	Utiliser une API	https://requests.readthedocs.io/en/latest/
Scikit-learn	Machine learning	https://scikit- learn.org/stable/user_guide.html
Matplotlib	Visualisation	https://matplotlib.org/stable/users/index.html
Os	Opérations système	https://docs.python.org/fr/3/library/os.html
Geopandas	Manipuler des données géographiques	https://geopandas.org/en/stable/
	1	I



Packages	Utilisation	Documentation
Httr	Utiliser une API	https://cran.r- project.org/web/packages/httr/vignettes/quickstart.html
Machine-learning in R	Machine learning	https://www.datacamp.com/tutorial/machine-learning-in-r
Ggplot2	Visualisation	https://ggplot2.tidyverse.org/
Rdal	'Geospatial' Data Abstraction Library	https://www.rdocumentation.org/packages/rgdal/versions/1.5-32

Ces listes ne sont en aucun cas exhaustives. D'autres librairies peuvent être utilisées si nécessaire. De plus, il ne s'agit pas de modules obligatoires. Certains peuvent ne pas être utiles selon les choix de résolution des cas d'usage.

1.2. Format de restitution

Le choix concernant les supports de restitution des travaux est laissé libre aux participants. Il serait tout de même préférable de réaliser deux livrables distincts : un document de synthèse des analyses et des réflexions menées ainsi que le script de code réalisé. Pour faciliter la compréhension de ce dernier, on peut utiliser des fichiers au format Notebook (type Jupyter Notebook en python) qui permettent de créer et de partager des documents interactifs contenant du code dynamique et exécutable, des visualisations de contenus, des textes de documentation et des équations.

Pour plus de précisions, se référer aux consignes établies dans le règlement du challenge.



2 - Sujet nº1

Comme présenté lors de la phase d'introduction du challenge, le premier cas d'usage présenté porte sur l'optimisation du positionnement des sites Enedis en Bretagne. L'objectif est de définir l'emplacement idéal des bases opérationnelles d'Enedis (villes sur lesquelles elles seraient implantées) ainsi que leur champ d'action (communes qui leur seraient rattachées).

2.1. Les données

2.1.1. Temps de trajet

Les calculs des temps de trajet entre deux communes ont été réalisés grâce à l'Open Source Routing Machine (ou OSRM). Il s'agit d'un moteur de recherche d'itinéraire haute performance qui permet d'obtenir les plus courts chemins dans un réseau routier.

Le fichier temps_trajet_30.csv contient, pour chaque commune de la région Bretagne, le temps de trajet en voiture (en secondes) vers les communes environnantes. Afin de réduire le volume de données fournies, il est à noter qu'il s'agit des villes de destination se situant à moins de 30 minutes de route de la commune de départ.

Format:

Colonne	Description courte
départ	Commune de départ
destination	Commune d'arrivée
durée	Temps de trajet sur route (en secondes)

Le fichier est encodé en latin-1 et utilise le point-virgule comme délimiteur.

2.1.2. Niveau d'interventions

L'emplacement d'un site ne doit pas dépendre exclusivement du fait qu'il soit bien desservi par le réseau routier. En effet, certaines communes sont plus sources d'interventions que d'autres et doivent donc être « priorisées » dans votre analyse afin d'être les plus proches possible des lieux de départ des agents.

L'activité d'Enedis sur une commune est souvent répartie en deux grandes catégories d'interventions : les opérations clientèle (souvent réalisées au niveau du compteur d'un particulier ou professionnel) ainsi que les opérations réseau (réalisées sur les ligne moyenne et basse tension).

Le fichier **niveau_interventions.csv** associe à chaque commune un niveau d'activité clientèle et réseau parmi les suivants :

- Très bas
- Bas
- Moyen
- Haut
- Très Haut

Ces niveaux d'activité sont basés sur les interventions réalisées en 2022.



Format:

Colonne	Description courte
code_insee_commune	Code INSEE de la commune
Niveau d'activité clientèle	Quantité d'interventions clientèle sur cette commune en 2022
Niveau d'activité réseau	Quantité d'interventions réseau sur cette commune en 2022

Le fichier est encodé en ansi et utilise le point-virgule comme délimiteur.

2.1.3. Liste des communes

Les deux jeux de données précédents ont été construits grâce à la liste des 1207 communes de la région Bretagne.

Vous pouvez retrouver cette liste dans le fichier communes_bre.csv. Les champs permettant de faire le lien entre les tables sont les suivants :

- Nom Officiel Commune Majuscule (noms des communes)
- Code Officiel Commune (code INSEE)

Si besoin, vous pouvez retrouver une documentation complète de ce fichier à sa source :

https://data.bretagne.bzh/explore/dataset/communes-

france/table/?disjunctive.reg_name&disjunctive.dep_name&disjunctive.arrdep_name&disjunctive.ze2020_na me&disjunctive.bv2012_name&disjunctive.epci_name&disjunctive.ept_name&disjunctive.com_name&disjunctive.ve.ze2010_name&disjunctive.com_is_mountain_area&sort=year

2.2. Aides au code

2.2.1. Obtenir un temps de trajet depuis les données fournies







2.2.2. Obtenir un temps de trajet absent des données fournies



```
# modules nécessaires
import requests as r
import json
# choix des lieux de départ et arrivée
latitude depart = 32.30069
longitude_depart = -49.96452
latitude arrivee = 5.74205
longitude_arrivee = 36.00096
# url de la requête
url = 'http://router.project-osrm.org/route/v1/car/'\
      f'{longitude_depart}, {latitude_depart};'\
f'{longitude_arrivee}, {latitude_arrivee}?overview=false'
r = requests.get(url) # exécution
if r.status code != 200: # cas d'erreur
   temps_trajet = None
else: # succès
    res = r.json()
    temps_trajet = res['routes'][0]['duration']
```



```
# modules nécessaires
library(httr)
# choix des lieux de départ et d'arrivée
latitude depart <- 48.292447150614976
longitude depart <- -1.723622895613565
latitude_arrivee <- 48.50228959077094
longitude_arrivee <- -1.5150621072780754
# url de la requête
url <- sprintf(paste0("http://router.project-osrm.org/route/v1/car/",</pre>
                       "%s,%s;%s,%s?overview=false"),
               longitude depart, latitude depart,
               longitude_arrivee, latitude_arrivee)
# exécution
r <- httr::GET(url)
# extraire la durée si la requête renvoie le code 200 (succès)
if (r\$status\_code == 200) {
    duree <- httr::content(r)$routes[[1]]$duration</pre>
 else {
    duree <- NULL
```



3 — Sujet n°2

Comme présenté en introduction du challenge, le deuxième sujet porte sur la recherche d'une implantation idéale de bornes de recharge pour véhicules électriques. On s'intéresse ici au territoire du département de l'Ille-et-Vilaine. L'objectif est de localiser des zones d'implantation potentielles qui ne nécessiteraient pas d'investissements majeurs sur le réseau. Autrement dit, en utilisant les capacités des différents tronçons du réseau déjà disponibles.

3.1. Les données

Ce sujet pouvant être considéré comme un peu plus difficile que le premier, il n'est pas obligatoire d'utiliser l'ensemble des données fournies mais bien celles qui vous paraissent les plus utiles.

3.1.1. Capacités du réseau basse tension projetées sur des portions de route

La capacité de soutirage d'un tronçon du réseau correspond à la puissance électrique disponible que l'on pourrait utiliser (ou soutirer) sans modifier la ligne.

Le fichier capa_route_35.geojson contient les positions géographiques de tous les tronçons du réseau routier sur le département ainsi que la puissance de soutirage associée à la ligne basse tension la plus proche.

Ce jeu de données a d'abord été construit en récupérant les tracés des routes grâce aux données d'Open Street Map. Les routes sont ensuite découpées selon leur proximité avec les tronçons du réseau basse tension. Dans les données fournies, le tronçon associé est le plus proche dans un rayon de 50 mètres.

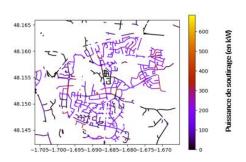


Fig : Exemple de visualisation des données fournies

Dans un but de simplification, l'information fournie n'est pas la capacité de soutirage brute (en kW) pour chaque tronçon mais le nombre de bornes de recharge dites « accélérée » que l'on pourrait y installer.

Format:

Colonne	Description courte
id_troncon	ldentifiant du tronçon
id_route	Identifiant de la route associée
id_commune	Code INSEE de la commune
nb_bornes_pot	Nombre de bornes pouvant être installées sur le tronçon
geometry	Coordonnées géographiques du tronçon

Attention, ce jeu de données est assez volumineux (233 507 lignes). Il faut compter plusieurs dizaines de secondes afin de l'importer.



3.1.2. Trafic journalier sur les routes nationales du département

Un des enjeux majeurs du déploiement d'infrastructures de recharge de véhicules électriques sur le territoire est leur proximité avec l'utilisateur final. Certains particuliers qui possèdent une maison peuvent installer une borne directement chez eux mais il ne s'agira pas de la grande majorité des utilisateurs de voitures électriques. Pour les autres, il leur faudra se déplacer vers une borne de recharge publique située près de leur lieu de travail ou de leur domicile. On peut donc imaginer qu'il serait intéressant d'implanter dans un premier temps des infrastructures de recharge proches de grands axes routiers.

Le fichier trafic_routier_35.shp contient, pour chaque route nationale du département de l'Ille-et-Vilaine, des informations concernant le comptage de véhicule au cours d'une année. Les données sont stockées dans un format « shapefile », très utilisé dans les systèmes d'informations géographiques.

Format:

Une description de l'ensemble des champs peut être trouvée dans le fichier catalogue-données-traffic.xlsx. On ne s'intéressera ici qu'aux colonnes essentielles (vous pouvez utiliser d'autres données qui vous semblent toutes aussi pertinentes pour votre analyse).

Colonne	Description courte
ID_SIR_TMJ	Identifiant du comptage
CODEINSEE	Code insee localisation commune compteur
AXE	Nom de la route
MJATVTCJN	Moyenne journalière annuelle tout véhicule tout jour confondu
geometry	Coordonnées géographiques de la route

3.1.3. Densité de population par commune et évolution depuis 1968

L'opération de comptage du nombre moyen de véhicule circulant sur une route est assez fastidieuse. Elle n'a donc pas été réalisée sur l'ensemble des routes du département mais seulement sur les nationales. Afin de pallier au manque d'information sur les plus petites routes qui peuvent également être très fréquentées, il peut être intéressant de s'intéresser à la densité de population d'une commune. En effet, on peut émettre l'hypothèse qu'une borne implantée dans une commune très peuplée sera sûrement plus utilisée que si elle avait été installée dans un petit village.

Le fichier insee_dens_pop_35.xlsx présente la densité de population (nombre d'habitants/km2) pour chaque commune de l'Ille-et-Vilaine ainsi que son évolution depuis 1968.

Format:

Colonne	Description courte
codgeo	Code INSEE de la commune
libgeo	Nom de la commune
an	Année de la mesure
dens_pop	Densité de population



3.1.4. Positions des bornes de recharge déjà implantées sur le domaine public

En plus des différentes informations contenues dans les jeux de données présentés précédemment, on peut également s'intéresser à l'état actuel du déploiement d'infrastructures de recharge sur le territoire. En effet, cela peut servir à identifier des zones propices au déploiement de bornes de recharges et qui ne sont pas (ou pas assez) équipées actuellement.

Pour cela, le fichier **etalab-schema_irve_35.geojson** contient la liste des 936 bornes de recharges recensées sur le département.

Format:

Colonne	Description courte
code_insee_commune	Code INSEE de la commune
puissance_nominale	Puissance de la borne
geometry	Coordonnées géographiques de la borne

Plus d'informations concernant les autres champs de ce jeu de données ici : https://www.data.gouv.fr/fr/datasets/5448d3e0c751df01f85d0572/

3.2. Aides au code

3.2.1. Importer des données au format geojson ou shape



```
# modules nécessaires
import geopandas as gpd

# import de données au format geojson
data_geo = gpd.read_file('data\\capa_route_35.geojson')

# import de données au format shapefile
data_shp = gpd.read_file('data\\trafic_routier_35.shp')

# les données sont ensuite exploitables sous la forme d'un
dataframe classique
```



3.2.2. Calcul de distance entre deux géométries

Il est possible de calculer la distance Euclidienne entre deux figures géométriques. Cependant, il est important de faire attention au système de coordonnées dans lequel on se place. En effet, la mesure de distance peut en être grandement impactée. Dans notre cas, on ne peut pas calculer une distance Euclidienne en mètres dans un système de coordonnées en degrés au risque de récupérer des mesures incohérentes. Pour cela, il est nécessaire de projeter les coordonnées dans un nouveau système (type EPSG:32643).



```
df_capa = gpd.read_file('data\\capa_route_35.geojson')
#Projection dans un système de coordonnées adapté
df_capa = df_capa.to_crs(32643)

df_traff = gpd.read_file('data\\trafic_routier_35.shp')
#Projection dans un système de coordonnées adapté
df_traff = df_traff.to_crs(32643)

#Distance Euclidienne entre deux points
df_capa.geometry[0].distance(df_traff.geometry[0])
```

