Coût de péages dans le sud de la France

El Khmissi Mohamed
Niasse Gueladio
Fontana Quentin

13 Décembre 2021

Table of contents

Outline

A savoir

1. **Langage de Programmation** : Python
2. **Cible** : Reseau autoroutier du Sud de la France (A9, A61, A62, A66, A75 et A709 entre Montpellier, Perpignan, Pamier et Toulouse).

Fonctionnalites

1. **Produire une Map interactive** :
   - selections des gares de depart et d'arrivee parmis une liste de gares de peage aux choix,
   - visualisation geographique du chemin a parcourir,
   - information sur la distance et sur le cout du trajet.
2. Proposer le **trajet le moins cher** en fonction du nombre d'arrets souhaites en cours de route.
3. Produire une description de la **distribution des prix** entre deux gares de peage.

Outline

Traitement puis importation des donnees dans VScode

1. Creation du dataframe "prices.csv" :
   –> conversion et sauvegarde en format .csv des donnees fournies a la page 3 du pdf :
   https://public-content.vinci-autoroutes.com/PDF/Tarifs-peage-asf-vf/ASF-C1-TARIFS-WEB-2021-maille-vf.pdf
2. Creation du dataframe "coordonnees.csv" :
   –> Importation du fichier gares-peage-2019.csv disponible a l'url suivant :
   https://www.data.gouv.fr/en/datasets/gares-de-peage-du-reseau-routier-national-concede/
3. Transformation du format des coordonnees "X" et "Y" du fichier coordonnees.csv :
   lambert93 (epsg :2154) –> GPS WGS84 (epsg :4326)

Completion et data-cleaning

1. Suppression des lignes et des colonnes inutiles dans les fichiers `coordonnees.csv` et `prices.csv`
2. Ajouts des gares de peages et de leurs donnees relatives dans le fichier `coordonnees.csv`

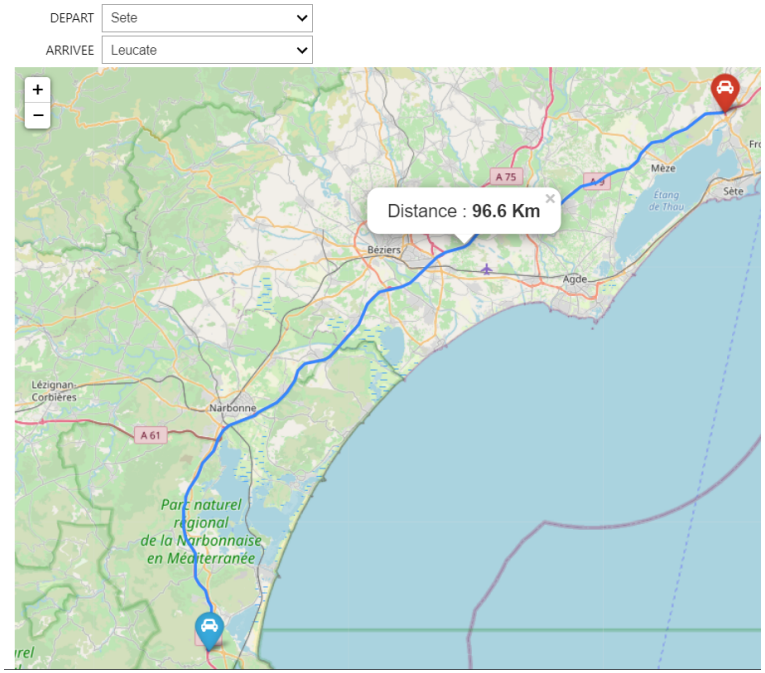
Voir code : `data_cleaning.py`

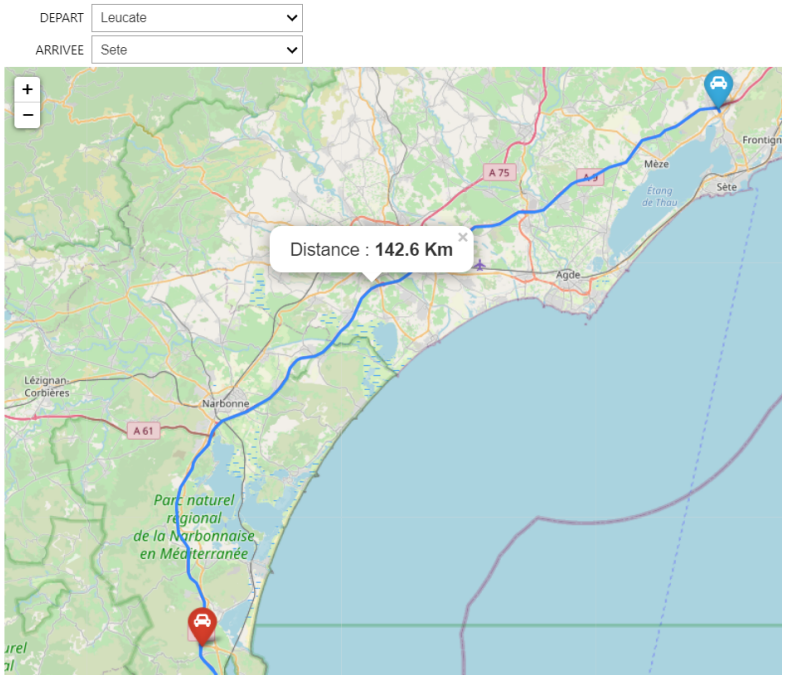
Création de la Map

1. La partie dataframe : dans cette partie on a utiliser le package `pandas` pour lire et préparer les bases de données `coordonneesclean.csv` et `pricesclean.csv` et une liste contient les noms des 36 villes qu'on a.

2. La partie visualisation : dans cette partie on a utiliser les packages `folium`, `openrouteservice` et `ipywidgets` pour visualiser la map interactive.

Probléme des distances d'aller-retour

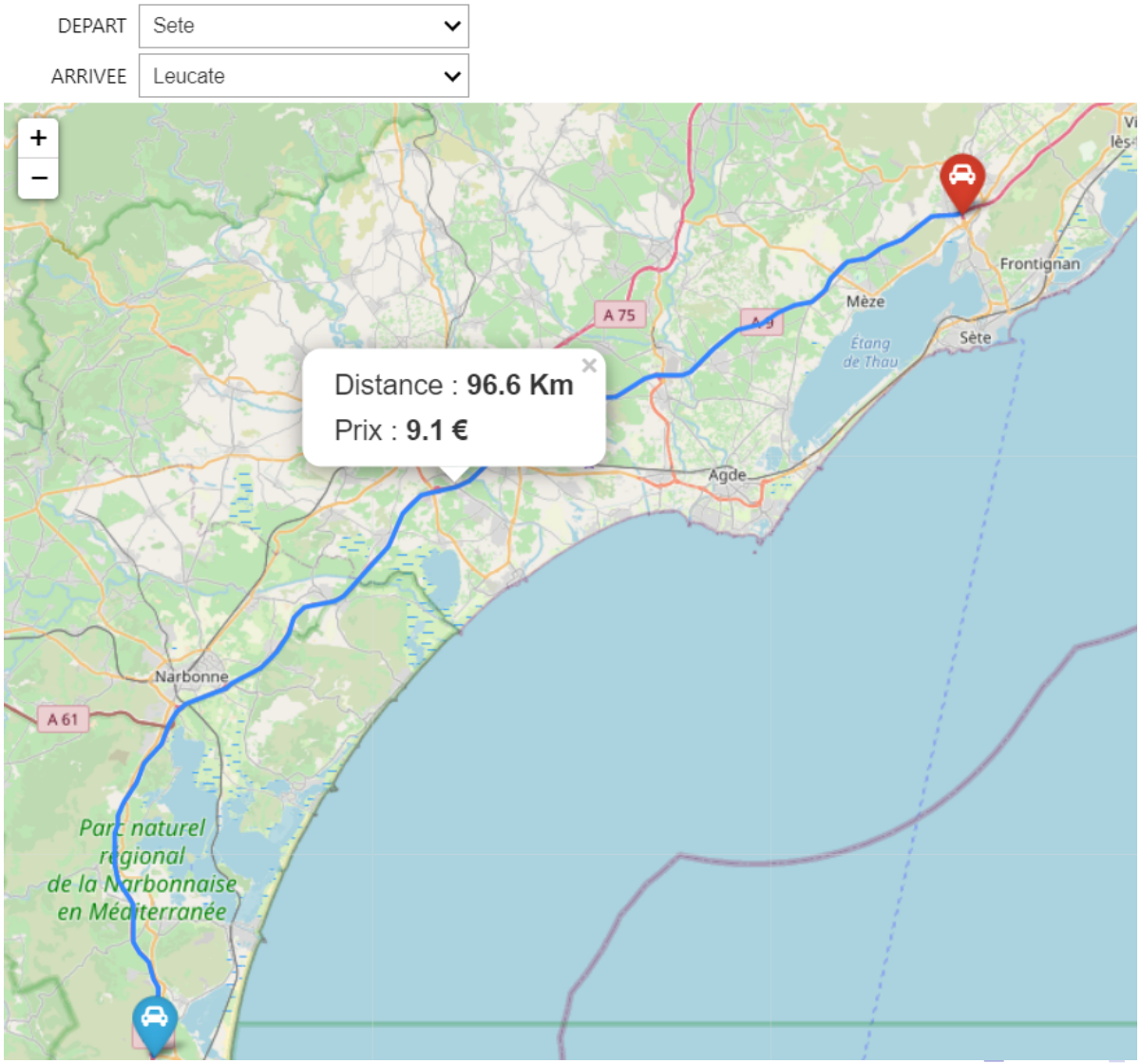




Résolution du probléme des distances

```
#problem solving of different outward and return distances

if i < j:

    cor = [x, y]


    for i in range(0, len(cor)-1):···
    return m

elif i > j:

    cor = [y, x]

    for i in range(0, len(cor)-1):···
    return m

else:
    print("Choisissez deux villes différentes")
```

Exemple du visualisation



Problematique

Comment determiner le trajet a parcourir lorsqu'un voyageur decide de s'arreter k fois en cours de route afin que ce trajet lui revienne le moins cher ?

Generation du dataframe d'un trajet correctement indexe

1. Partitionnement et regroupement des gares de peages
2. La fonction : `nb_de_gare_sur_trajet(i,j)` :
   –> renvoie le nombre de gares qui separe la gare de depart et d'arrivee.
3. La fonction composee : `Le_Trajet(tab_algo(i,j))` :
   –> renvoie un dataframe ordonne constitue des gares presentes sur le trajet de la gare i a j et dont l'index va de 0 a `nb_de_gare_sur_trajet(i,j)+1`.

Exemple

Alfonse est a Perpignan Sud (13) et veut se rendre a Beziers Ouest (7).

```
nb_de_gare_sur_trajet(13,7) ···
```

```
5
```

```
Le_Trajet(tab_algo(13,7)) ···
```

	level_0	index	ROUTE	NOMGARE	X	Y
0	3	13	A0009	Perpignan sud	2.861366	42.664169
1	2	12	A0009	Perpignan nord	2.895039	42.782804
2	1	11	A0009	Leucate	2.969864	42.939080
3	0	10	A0009	Sigean	2.952680	43.033247
4	0	9	A0009	Narbonne sud	2.993876	43.163068
5	1	8	A0009	Narbonne est	3.034480	43.176588
6	2	7	A0009	Beziers ouest	3.222949	43.303664

Generation de k+1 sous-trajets

1. Fonction `liste_des_tuples(p,n)` :
   –> renvoie la liste de tout les arrangements ordonnes possibles de p parmis n allant de 0 à l'index de la gare d'arrivee.

2. Fonction `suite_de_trajet(p,n)` :
   –> renvoie pour chaque arrangement possible generes par la fonction precedente une suite de couple representant la suite de k+1 sous-trajets.

Exemple

```
liste_des_tuples(4,7) ···
```

```
[(0, 1, 2, 6),
 (0, 1, 3, 6),
 (0, 1, 4, 6),
 (0, 1, 5, 6),
 (0, 2, 3, 6),
 (0, 2, 4, 6),
 (0, 2, 5, 6),
 (0, 3, 4, 6),
 (0, 3, 5, 6),
 (0, 4, 5, 6)]
```

```
suite_de_trajet(4,7) ···
```

```
[[(0, 1), (1, 2), (2, 6)],
 [(0, 1), (1, 3), (3, 6)],
 [(0, 1), (1, 4), (4, 6)],
 [(0, 1), (1, 5), (5, 6)],
 [(0, 2), (2, 3), (3, 6)],
 [(0, 2), (2, 4), (4, 6)],
 [(0, 2), (2, 5), (5, 6)],
 [(0, 3), (3, 4), (4, 6)],
 [(0, 3), (3, 5), (5, 6)],
 [(0, 4), (4, 5), (5, 6)]]
```

Chemin le moins cher

- Fonction `cout_minimum(data,depart,arrivee)` :
  - arguments :
    - data = `Le_Trajet(tab_algo(i,j))`
    - depart = 0
    - arrivee = `nb_de_gare_sur_trajet(i,j)+1`

  –> renvoie le cout d'un trajet lorsqu'il se fait sans arret.

Chemin le moins cher

- Fonction `cout_minimum_tuple(depart,arrivee,k,data)` :
  - arguments :
    - data = `Le_Trajet(tab_algo(i,j))`
    - depart = 0
    - arrivee = `nb_de_gare_sur_trajet(i,j)+1`
    - k = nombre d'arrets

  –> renvoie le cout minimum d'un trajet en fonction de k.

Exemple

Alfonse decide de s'arreter deux fois en cours de route. Il voudrait savoir a quelle gare de peage sortir afin que le cout de son trajet soit minimise. Il utilise donc notre algorithme.

```
chemin_moins_cher(7,13,2) ···
```

```
Si vous sortez aux gares :
(Leucate,Perpignan nord), votre trajet vous coutera 7.0 €.
Vous aurez ainsi fait une economie de 1.4 € sur le coût des
péages.
```

Objectif

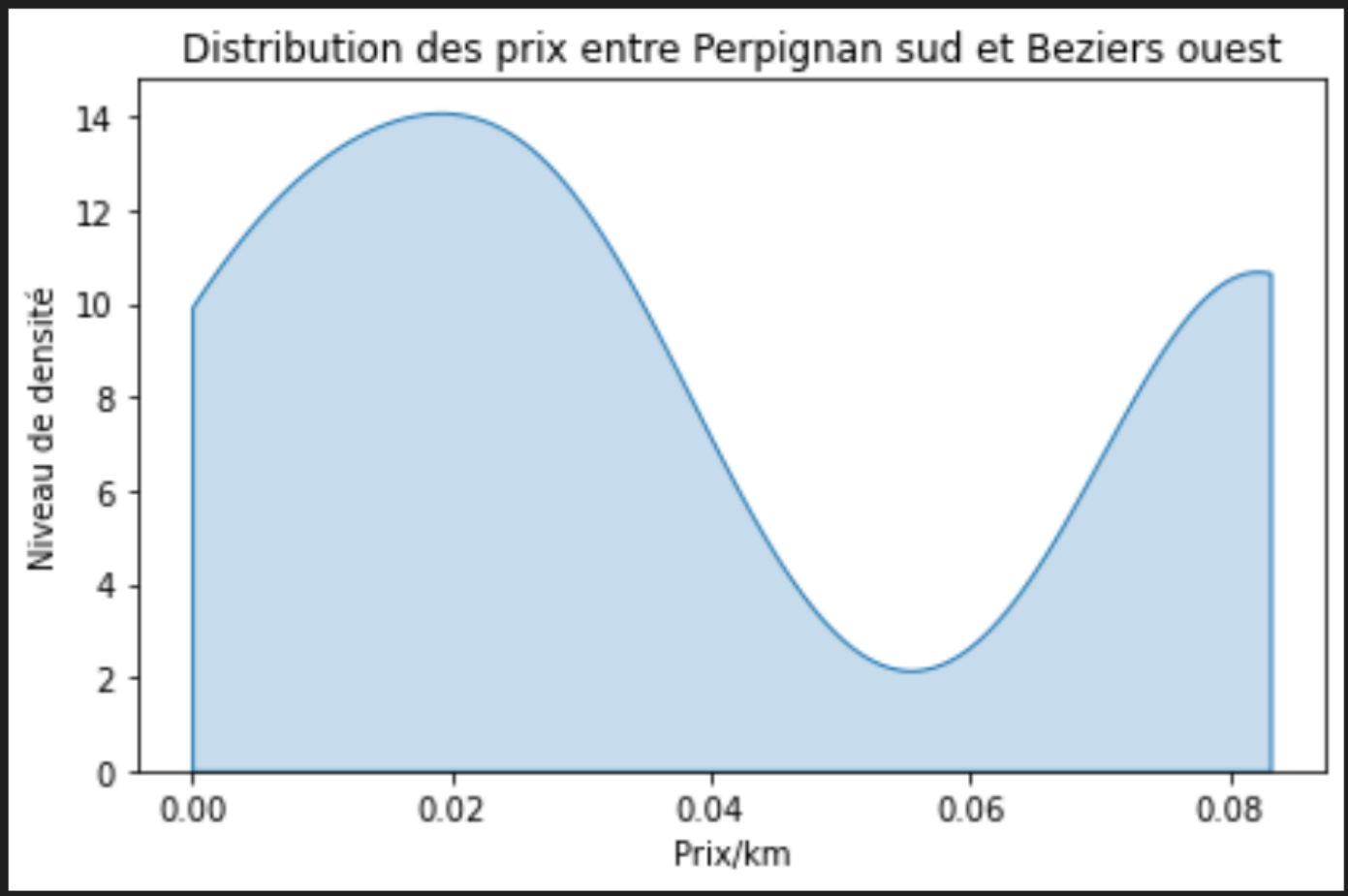
Determiner la distribution des prix des peages par kilometre suivant un trajet donne.

Principe

1. Choisir un trajet a etudier
2. Rechercher dans le dataframe des prix la donnee relative au cout de peage entre chaque gares successives presentes sur le trajet.
3. Diviser respectivement chacun des prix retenus par la distance en kilometres qui separe les gares.
4. Rapprocher tout les resultats sous la forme d'une courbe de densitee dans un plot a l'aide de la fonction `kdeplot` fournie par le package `seaborn`.

Exemple

```
prices_distribution(13,7,0.5) ···
```



Outline

– Objectif du project

- Pandas
- Folium
- osmix
- networkx

Outline