

Analyse de la ségrégation spatiale à Paris

Stage de fin de DUT STID

Axel CARTIER



17 juin 2019



IUT DE PARIS



Tutoré par : Julien RANDON-FURLING

- 1 Contexte de la mission
- 2 Recueil des données
- 3 Développement de la méthode
 - Traitement des données brutes
 - Programmation de la méthode
- 4 Résultats de l'analyse
- 5 Conclusion

IUT DE PARIS





La mission

Objectif :

- Retrouver des résultats publiés dans

**From urban
segregation to spatial
structure detection**

Journal Title
XX(X):1-18
©The Author(s) 2018
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Julien Randon-Furling¹, Madalina Olteanu^{1,2}, Antoine Lucquiaud¹

Contraintes techniques :

- Complexité en temps
- Réutilisabilité du programme
- Autre langage de programmation



Présentation de la méthode [1]



Données de L'INSEE [2]

Figure 1 – Données du recensement 2013 en Ile de France

COMMUNE	ARM	IRIS	...	HLML	...	IPONDL	...	TRIRIS	TYPC	TYPL	VOIT	WC
75056	75101	751010101	...	2	...	3.586660	...	750011	3	2	1	Z
75056	75101	751010101	...	2	...	3.586660	...	750011	3	2	1	Z
75056	75101	751010101	...	2	...	3.586660	...	750011	3	2	0	Z
75056	75101	751010101	...	2	...	3.586660	...	750011	3	2	1	Z
75056	75101	751010101	...	2	...	3.586660	...	750011	3	2	1	Z
75056	75101	751010101	...	2	...	1.302288	...	750011	3	2	X	Z
75056	75101	751010101	...	2	...	3.544388	...	750011	3	2	0	Z
75056	75101	751010101	...	2	...	3.740054	...	750011	3	2	0	Z
75056	75101	751010101	...	2	...	1.302286	...	750011	3	2	0	Z
75056	75101	751010101	...	1	...	3.740059	...	750011	3	2	X	Z
...

70 variables pour 2 694 072 observations



Données de L'IGN [3]

	INSEE_COM	NOM_COM	IRIS	CODE_IRIS	NOM_IRIS	TYP_IRIS	geometry
0	01001	L'Abergement-Clémenciat	0000	010010000	L'Abergement-Clémenciat	Z	POLYGON ((846847.69999999131 6563787.000003602,...
1	01002	L'Abergement-de-Varey	0000	010020000	L'Abergement-de-Varey	Z	POLYGON ((889088.1999998859 6549317.300003688,...
2	01004	Ambérieu-en-Bugey	0102	010040102	Longeray-Gare	H	POLYGON ((882210.5999998887 6542241.900003731,...
3	01004	Ambérieu-en-Bugey	0202	010040202	Tiret-Les Allymes	H	POLYGON ((882210.5999998887 6542241.900003731,...
4	01004	Ambérieu-en-Bugey	0101	010040101	Les Perouses-Triangle d'Activité	H	POLYGON ((880308.3099998899 6542546.350003731,...
...
50148	69255	Vaugneray	0000	692550000	Vaugneray	Z	POLYGON ((825415.0999999206 6513888.200003888,...
50149	71578	Clux-Villeneuve	0000	715780000	Clux-Villeneuve	Z	POLYGON ((865238.2999999148 6651468.600003094,...
50150	72137	Villeneuve-en-Perseigne	0000	721370000	Villeneuve-en-Perseigne	Z	POLYGON ((497663.2000000729 6823406.000002102,...
50151	73263	Saint-Offenge	0000	732630000	Saint-Offenge	Z	POLYGON ((933179.5999998527 6518449.700003878,...
50152	77316	Orvanne	0000	773160000	Orvanne	Z	POLYGON ((685093.8000000056 6805315.100002186,...

50153 rows x 7 columns

Figure 2 – Découpage géographique de l'IGN¹ au 1^{er} janvier 2015



Nombre de logements et de logements sociaux par IRIS

```
1 FD_LOGEMTZA_2013 <-  
2   read.csv("/Users/Maxwel/Desktop/WORK EN COURS/STAGE/DATA/rp2013_logemtza_txt/FD_LOGEMTZA_2013.txt", sep=";")  
3 DATA<-  
4   FD_LOGEMTZA_2013[which(FD_LOGEMTZA_2013$COMMUNE==75056),]  
5 DATA<-  
6   DATA[which(DATA$CATL==1),c(3,28,50)]  
7 DF2<-  
8   DATA %>% group_by(IRIS,HLML) %>% summarise(IPONDLS=sum(IPONDLS))  
9 CLOGS=DF2[DF2$HLML==1,-2]  
10 CLOGT= aggregate(DF2$IPONDLS, by=list(Category=DF2$IRIS), FUN=sum)  
11 colnames(CLOGT)[1] <- "IRIS"  
12 DFF<-merge(CLOGT,CLOGT,by="IRIS",all.y = TRUE)  
13 colnames(DFF)[1] <- "CODE_IRIS"  
14 colnames(DFF)[2] <- "logs"  
15 colnames(DFF)[3] <- "logt"  
16 DFF[is.na(DFF)] <- 0  
17 write.table(DFF, file = "/Users/Maxwel/Desktop/WORK EN COURS/STAGE/LOGPARISPF.CSV", sep = ";")  
18
```

Figure 3 – Programme R



Préparation des données

```
## IMPORT + PREPARATION DATA ##
2 data = pd.read_csv("CHEMIN_DONNEES_TRAITES", sep=";")
  IRIS_B=gp.read_file('CHEMIN_SHAPEFILE')
4 PARIS=IRIS_B[IRIS_B['NOM_COM'].str.match('Paris')]
  PARIS=PARIS.drop(PARIS[PARIS["NOM_COM"]=="Parisot"].index)
6 PARIS=PARIS.drop(PARIS[PARIS["NOM_COM"]=="Paris-l'Hôpital"].index)
  PARIS=PARIS.drop(PARIS[PARIS['NOM_IRIS'].str.match('Bois')].index)
8 PARIS=PARIS.drop(PARIS[PARIS['NOM_IRIS'].str.match('Jardin du Luxembourg')].index)
  PARIS['x']=PARIS.geometry.centroid.x
10 PARIS['y']=PARIS.geometry.centroid.y
  df=PARIS.ix[:,('x','y','CODE_IRIS','NOM_IRIS','geometry')]
12 df['CODE_IRIS']=pd.to_numeric(df.CODE_IRIS)
  data2= pd.merge(df,data,on="CODE_IRIS",how="outer")
14 data2=data2.fillna(value=0)
  data2=data2[data2["logt"]!=0]
16 df=data2
  df['taux_base']=df["logs"]/df["logt"]
18 df['sorted_index']= None
  df['state_logt']= None
20 df['state_logs']= None
  df['taux']= None
22 df['check']=None
```




Structure de l'algorithme

```
import numpy as np
2 import pandas as pd
import sklearn as skl
4 import matplotlib.pyplot as plt
import datetime
6 import geopandas as gp
## RANDOMISATION INDICES ##
8 df=shuffle(df)
df=df.reset_index(drop=True)
10 ## CALCUL DES SEUILS ##
taux_global=df.logs.sum()/df.logt.sum()
12 seuil=0.05
s= [taux_global*(1-seuil), taux_global*(1+seuil)]
14 ## CALCUL KDTREE ##
kdt = KDTree(df[['x', 'y']])
16 ## CALCUL INDEX PLUS PROCHES VOISINS ##
for x in range(len(df)):
18     origin= np.stack((df.x[x], df.y[x]), axis=-1)
    origin_kdt = np.expand_dims(origin, axis=0)
20     nearest_point_index = kdt.query(origin_kdt, k=len(df),
        return_distance=True)
    df.sorted_index[x]=nearest_point_index[1]
22     df.state_logs[x]= np.ones((1, len(df)))
    df.state_logt[x]= np.ones((1, len(df)))
24     df.taux[x]= np.ones((1, len(df)))
```



Structure de l'algorithme

```
## CALCUL DES NIVEAUX D'AGREGATION ##
2 for i in range(len(df)) :
    for j in range(len(df)) :
4         if j == 0 :
            df.state_logt[i][0,j]=df.logt[i]
            df.state_logs[i][0,j]=df.logs[i]
6             df.taux[i][0,j]=df.state_logs[i][0,j]/df.state_logt[i][0,j]
8             if s[0] <df.taux[i][0,j]<s[1] :
                df.check[i]= 0
10         else :
            df.state_logt[i][0,j]= df.state_logt[i][0,(j-1)] + df.logt[
df.sorted_index[i][0,j]]
12             df.state_logs[i][0,j]= df.state_logs[i][0,(j-1)] + df.logs[
df.sorted_index[i][0,j]]
            df.taux[i][0,j]= df.state_logs[i][0,j]/df.state_logt[i][0,j]
14             if (s[0] <df.taux[i][0,j]<s[1]) and (df.check[i] is None) :
                df.check[i]=j
```

Moins de logements sociaux dans les quartiers attractifs

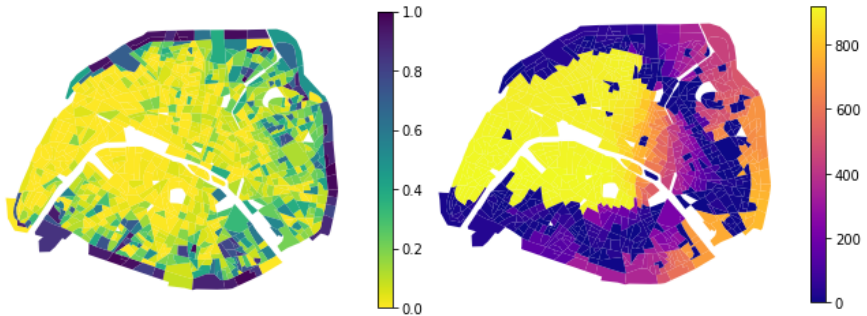


Figure 4 – Taux de logements sociaux pour chaque point de départ (à droite) et rapidité de convergence vers la moyenne parisienne (à gauche)

Un parc de logements polarisé

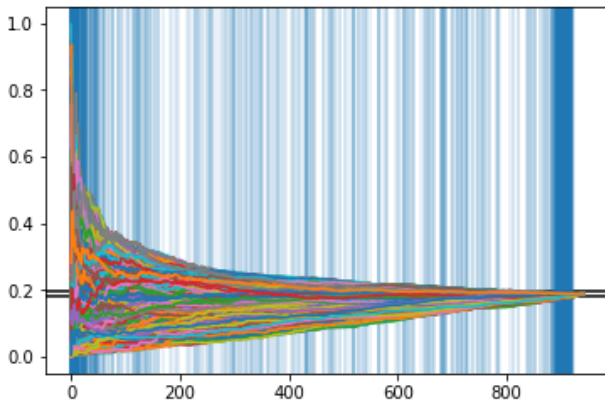


Figure 5 – Trajectoires du taux de logements sociaux en fonction du niveau d'agrégation



Autres applications et prolongements envisagés

Réaliser une infographie sur le thème de la ségrégation

Faire fonctionner l'algorithme avec d'autres données :

- IRIS des autres grandes villes françaises
- Données carroyés à 200m de l'INSEE en région parisienne

Méthode d'agrégation

Définition : distance euclidienne

Distance donnée par $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$ pour deux points x et y dans un espace métrique à n dimension(s)

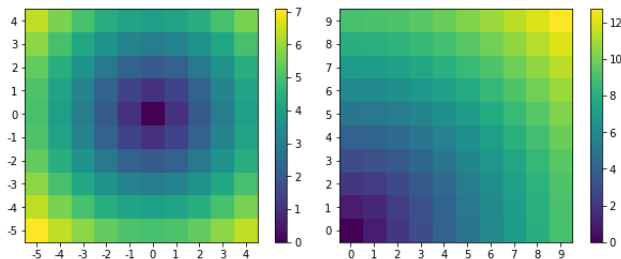


Figure 6 – Distance euclidienne à (0,0)



Un stage extrêmement enrichissant

Intérêts de la méthode

- Approche multi-échelle
- Applications diverses [4]

Quelques nuances

- Les données sont déclaratives
- Optimisation du programme



Références



Julien Randon-Furling, Madalina Olteanu et Antoine Lucquiaud. "From Urban Segregation to Spatial Structure Detection". In : *Environment and Planning B : Urban Analytics and City Science* (déc. 2017).



Insee. *Données logements ordinaires 2013*.
<https://www.insee.fr/fr/statistiques/2409491?sommaire=2409559>.
Consulté le 02-05-19. 2016.



IGN. *Contours IRIS*. <http://professionnels.ign.fr/contoursiris>.
Consulté le 02-05-19. 2017.



Madalina Olteanu, Julien Randon-Furling et William A. V. Clark. "Segregation through the multiscalar lens". In : *Proceedings of the National Academy of Sciences* (2019).

Merci pour votre attention