

# Projet1

Camille BRECHOTEAU, Benjamin PERSON, Fanny MARTIN

12/12/2020

## I- INTRODUCTION

Dans le cadre de notre troisième année du cycle de Licence en biologie-informatique ou en biologie option bio-informatique (BI) à l'Université Evry Val-d'Essonne, il nous est proposé un projet de groupe nous permettant de mettre en pratique nos connaissances en programmation R au travers d'un projet ayant pour finalité la reproduction du modèle de ségrégation spatiale proposée dans les années 1970 par Thomas C. Schelling qui a marqué les esprits en raison de l'effet pervers qu'elle suggérait : il existerait un lien entre une ségrégation spatiale et le résultat collectif de décisions individuelles qui ne visent pas à une telle ségrégation. Ainsi que l'étude de la densité de population et que du seuil d'intolérance dans ce dernier.

## II-FONCTION SIMU

### A-One.simu, le modèle de Schelling

On crée une fonction nommée "one.simu" qui permettra de réaliser les dynamiques des populations bleues et rouges selon le modèle de Schelling. Elle prend en paramètre un seuil (initialiser à 3), une taille de côté (longueur = largeur, initialisée à 50) ainsi que  $u$  (initialiser à 1 qui permet de faire varier la densité d'espaces vacants). On va créer une matrice carrée taille  $N \times N$  avec trois différentes valeurs possibles, 0 pour les espaces vacants (dont la proportion changera selon la valeur de  $u$ ), 1 pour les agents Bleus et 2 pour les agents Rouges.

On a utilisé une boucle while et 2 conditions d'arrêt, une qui donne un nombre d'itération maximal (on a choisi 1000) et l'autre qui stoppera la boucle en cas d'impossibilité de continuer.

A chaque itération : On commence par déterminer le nombre de voisins Bleus et le nombre de voisins Rouges pour toutes les cases de la matrice.

On détermine ensuite les listes des agents insatisfaits (bleu, export1 puis rouge, export2) et les listes des endroits vacants qui satisferont les bleus et rouges (import1 et import2).

On vérifie la condition d'arrêt, "stop", si jamais l'une des 4 listes déterminée précédemment est vide il faut stopper la boucle while.

Si la boucle peut continuer, on va sélectionner une position au hasard parmi les agents Bleus/Rouges insatisfaits ainsi qu'une case vacante parmi "import1" et "import2"

Enfin; on placera le Bleu insatisfait dans l'espace vacant qui lui permet d'être satisfait et on fera de même pour le Rouge. On implémentera i et la boucle recommencera.

Une fois la boucle terminée on renvoie la matrice modifiée.

```
#APPLIQUE LE MODE DE Schelling
one.simu<-function(seuil=3,N=50,u=1){
M <- matrix(sample(c(0,1,2), replace = TRUE, prob = c(u,1,1), size = N^2),
nrow = N)

V<- 1:N
G <- c(V[-1],V[1])
D <- c(V[N],V[-N])

i<-1
stop<-FALSE

while(i<=1000 & stop==FALSE)
{

nb1 <- (M[D,] == 1) + (M[G,] == 1) + (M[,D] == 1) + (M[,G] == 1) + (M[D,D] ==
1) + (M[G,D] == 1) + (M[D,G] == 1) + (M[G,G] == 1)
nb2 <- (M[D,] == 2) + (M[G,] == 2) + (M[,D] == 2) + (M[,G] == 2) + (M[D,D] ==
2) + (M[G,D] == 2) + (M[D,G] == 2) + (M[G,G] == 2)
#nb1, nb2 = nombre de 1 et de 2 autour de chaque case

#A DEPLACER
export1 <- (nb1 < seuil) & (M == 1)
export2 <- (nb2 < seuil) & (M == 2)
export1
export2

#OU DEPLACER
import1 <- (nb1 >= seuil) & (M == 0)
import2 <- (nb2 >= seuil) & (M == 0)
import1
import2

#CHOISIR UN AGENT INSATISFAIT
which(export1)
which(export2)
if( length(which(export1))==0 | length(which(export2))==0 |
length(which(import1))==0 | length(which(import2))==0){
  stop<-TRUE
}

if(stop==FALSE){
```

```

agent1<-sample(x=which(export1),size=1)
agent2<-sample(x=which(export2),size=1)
agent1
agent2

#CHOSIR ESPACE VACANT SATISFAISANT
#v1<-which(import1)
#v2<-which(import2)

vacant1<-sample(x=which(import1),size=1)
vacant2<-sample(x=which(import2),size=1)
vacant1
vacant2

#DEMENAGEMENT
M[agent1]<-0
M[vacant1]<-1
M[agent2]<-0
M[vacant2]<-2
}
i<-i+1
}
M
return(M)
}

```

## B-Two.simu, meusre d'agrégation

On va par la suite mesurer l'agrégation. On va créer une autre fonction qui prendra en paramètre une matrice M. Il existe plusieurs méthodes, on a choisi de calculer le N0 qui correspond au nombre d'agents entouré de 8 agents de même type que lui (ex: un Bleu entouré de 8 Bleus). Pour cela, on recalcule "nb1" et "nb2".

On utilise un accumulateur "N0" que l'on implémente quand on rencontre un agent entouré par 8 agents du même type en parcourant la matrice (grâce à la boucle for). La fonction retourne la valeur N0.

```

#MESURE AGREAGTION N0
two.simu<-function(M,N=50){

V<- 1:N
G <- c(V[-1],V[1])
D <- c(V[N],V[-N])

nb1 <- (M[D,] == 1) + (M[G,] == 1) + (M[,D] == 1) + (M[,G] == 1) + (M[D,D] ==
1) + (M[G,D] == 1) + (M [D,G] == 1) + (M[G,G] == 1)
nb2 <- (M[D,] == 2) + (M[G,] == 2) + (M[,D] == 2) + (M[,G] == 2) + (M[D,D] ==
2) + (M[G,D] == 2) + (M [D,G] == 2) + (M[G,G] == 2)

```

```

#nb1, nb2 = nombre de 1 et de 2 autour de chaque case
nb1
nb2

N0<-0
#parcour de la matrice,
for(j in 1:(N*N)){

  if (M[j]==1 & nb1[j]==8){
    N0<-N0+1

  }
  if (M[j]==2 & nb2[j]==8 ){
    N0<-N0+1
  }

}

return (N0)
}

```

### III- SIMULATION

On va maintenant utiliser les 2 fonctions afin d'observer et analyser les déplacements de populations en faisant varier 2 paramètres : le seuil d'intolérance et la densité d'espaces vacants.

On va utiliser 3 seuils différents : 3,4 et 5. Au-delà de 5 les agents sont trop souvent insatisfait et en dessous de 3 quasiment toujours satisfait (selon l'article) Pour ce qui est de la densita, on va varier entre 2, 15 et 33% d'espaces vacants.

#### A-Résultats

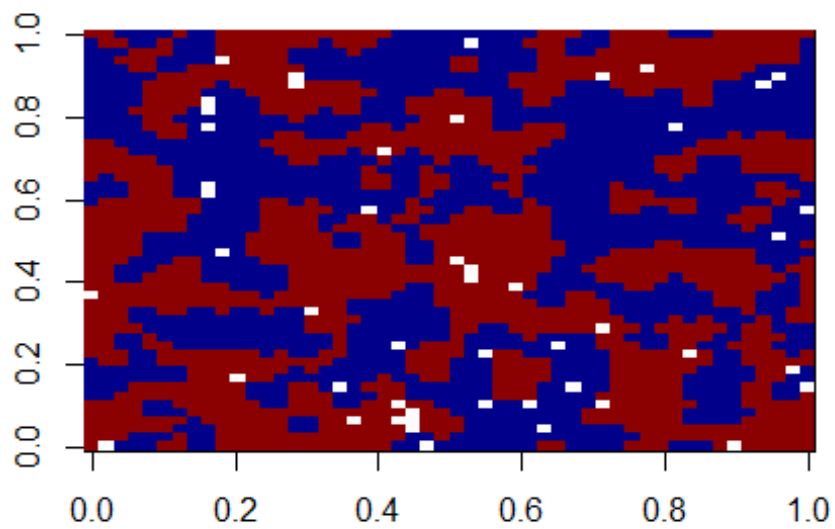
Voici des matrices avec les différents seuils et densités :

Seuil: 3, Densité: 2%

```

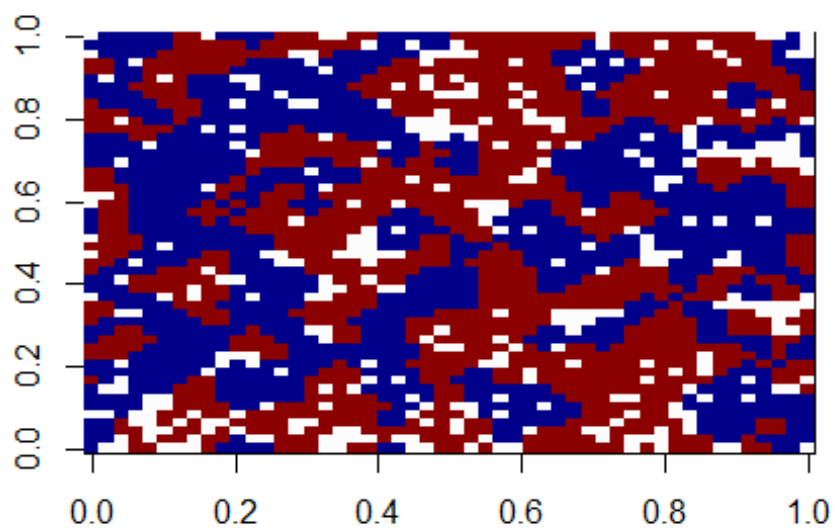
s3d02EX<-one.simu(3,50,0.04)#densité 2%
image(s3d02EX,col=c("gray99","darkblue","darkred"))

```



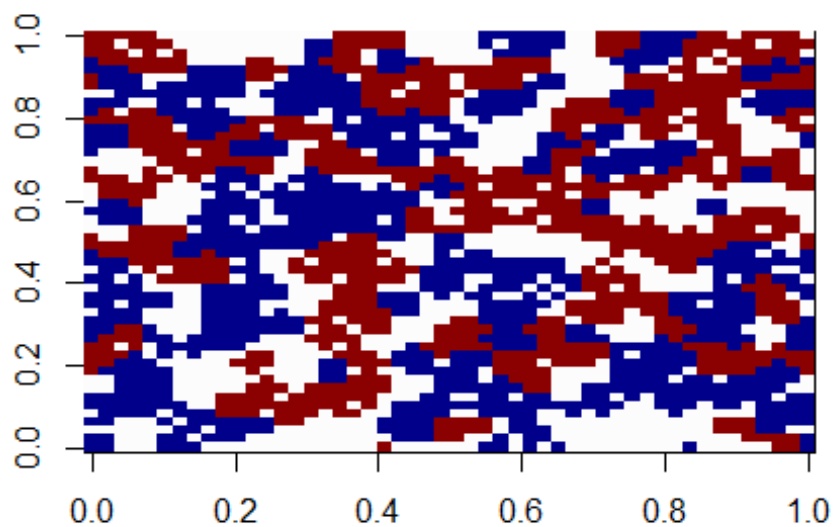
Seuil: 3, Densité: 15%

```
s3d15EX<-one.simu(3,50,0.35)#densité 15%
image(s3d15EX,col=c("gray99","darkblue","darkred"))
```



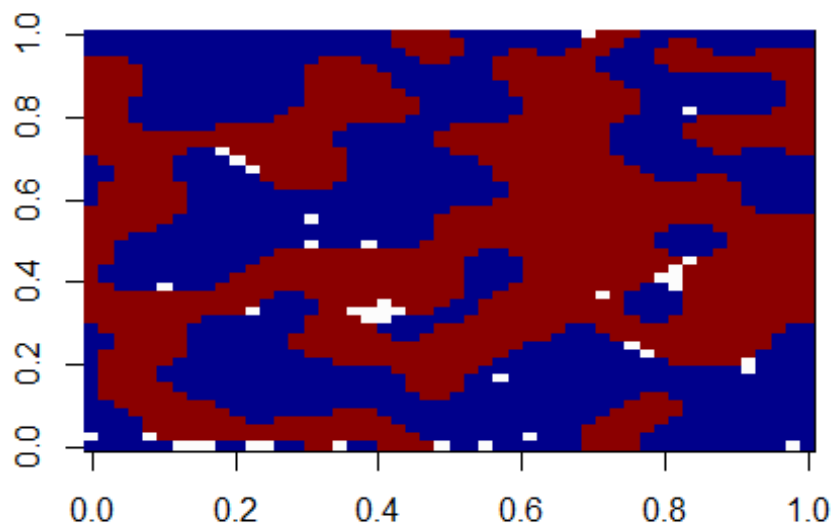
Seuil: 3, Densité: 33%

```
s3d33EX<-one.simu(3,50,1)#densité 33%  
image(s3d33EX,col=c("gray99","darkblue","darkred"))
```



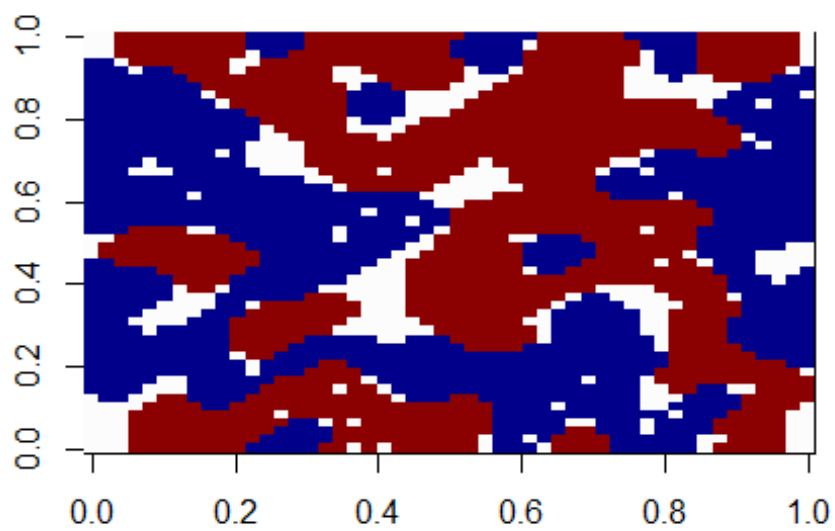
Seuil: 4, Densité 2%

```
s4d02EX<-one.simu(4,50,0.04)#densité 2%  
image(s4d02EX,col=c("gray99","darkblue","darkred"))
```



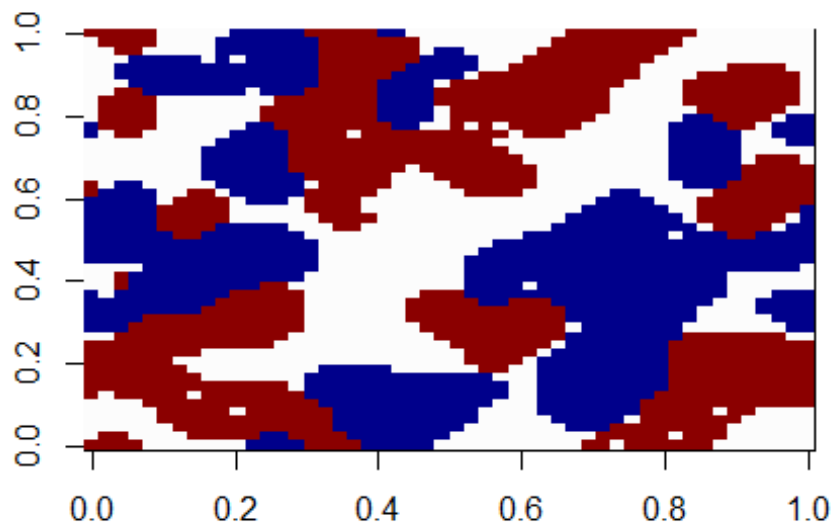
Seuil: 4, Densité 15%

```
s4d15EX<-one.simu(4,50,0.35)#densité 15%  
image(s4d15EX,col=c("gray99","darkblue","darkred"))
```



Seuil: 4, Densité 33%

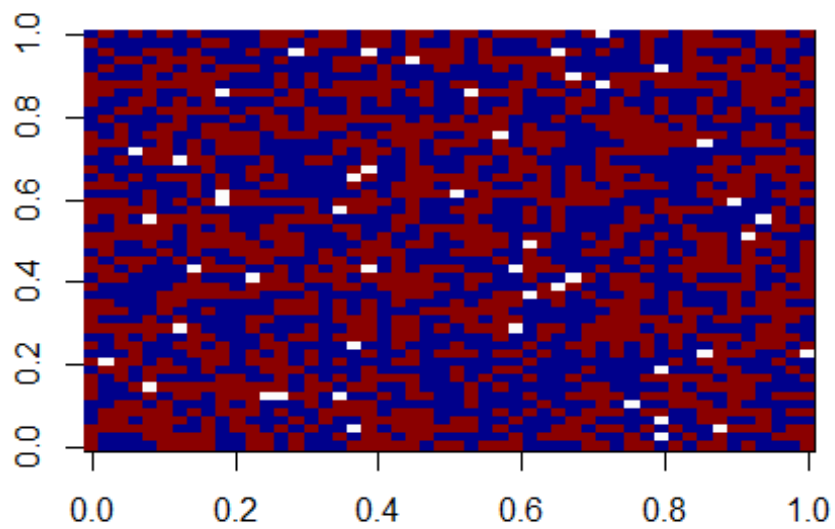
```
s4d33EX<-one.simu(4,50,1)#densité 33%  
image(s4d33EX,col=c("gray99","darkblue","darkred"))
```



Seuil: 5, Densité 2%

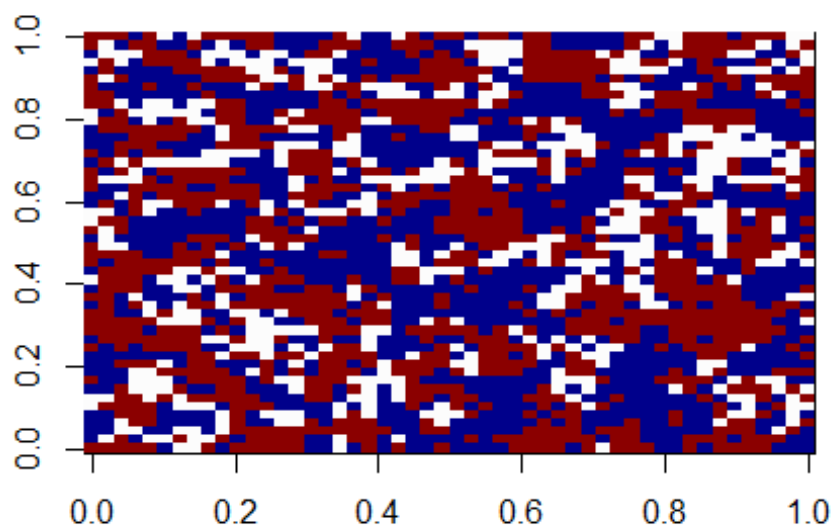
```
s5d02EX<-one.simu(5,50,0.04)#densité 2%  
image(s5d02EX,col=c("gray99","darkblue","darkred"))
```





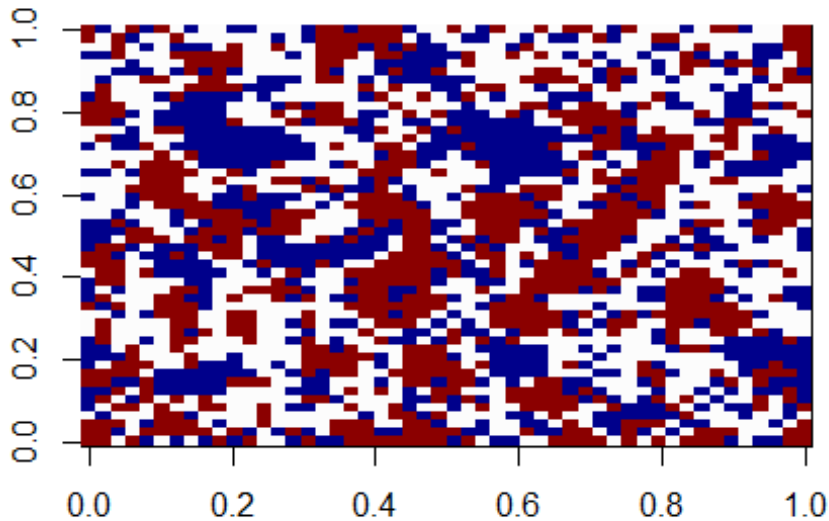
Seuil: 5, Densité 15%

```
s5d15EX<-one.simu(5,50,0.35)#densité 15%  
image(s5d15EX,col=c("gray99","darkblue","darkred"))
```



Seuil: 5, Densité 2%

```
s5d33EX<-one.simu(5,50,1)#densité 33%  
image(s5d33EX,col=c("gray99","darkblue","darkred"))
```



Afin pouvoir analyser l'effet du seuil et de la densité, on va recourir à "replicate". On va exécuter 100x chaque combinaison de seuil et densité. On obtiendra des listes de N0 de tous les seuils et densités.

```
#simu avec seuil 3 et densité de 20 a 60%  
s3d02<-replicate(100,two.simu(one.simu(3,50,0.04)))  
s3d15<-replicate(100,two.simu(one.simu(3,50,0.35)))  
s3d33<-replicate(100,two.simu(one.simu(3,50,1)))  
  
#simu avec seuil 4 et densité de 20 a 60%  
s4d02<-replicate(100,two.simu(one.simu(4,50,0.04)))  
s4d15<-replicate(100,two.simu(one.simu(4,50,0.35)))  
s4d33<-replicate(100,two.simu(one.simu(4,50,1)))  
  
#simu avec seuil 5 et densité de 20 a 60%  
s5d02<-replicate(100,two.simu(one.simu(5,50,0.04)))  
s5d15<-replicate(100,two.simu(one.simu(5,50,0.35)))  
s5d33<-replicate(100,two.simu(one.simu(5,50,1)))
```

On va créer un data.frame. On va utiliser les moyennes des mesures d'agrégation obtenues (N0).

```

d<-c(2,15,33)
echelle<-c(0,750,1500)
x3<-c(mean(s3d02),mean(s3d15),mean(s3d33))
x4<-c(mean(s4d02),mean(s4d15),mean(s4d33))
x5<-c(mean(s5d02),mean(s5d15),mean(s5d33))

d
## [1]  2 15 33

x3
## [1] 604.54 235.07 133.72

x4
## [1] 1301.01 1031.93  817.81

x5
## [1] 80.71 95.18 57.90

liste<-list(d,x3,x4,x5,echelle)
donnees<-data.frame(liste)
colnames(donnees)<-c("densité","mesure agrégation seuil 3", "mesure
agrégation seuil 4","mesure agrégation seuil 5","echelle")
print(summary(donnees))

##      densité      mesure agrégation seuil 3 mesure agrégation seuil 4
## Min.   : 2.00   Min.   :133.7           Min.   : 817.8
## 1st Qu.: 8.50   1st Qu.:184.4           1st Qu.: 924.9
## Median :15.00   Median :235.1           Median :1031.9
## Mean   :16.67   Mean   :324.4           Mean   :1050.2
## 3rd Qu.:24.00   3rd Qu.:419.8           3rd Qu.:1166.5
## Max.   :33.00   Max.   :604.5           Max.   :1301.0
## mesure agrégation seuil 5  echelle
## Min.   :57.90           Min.   :  0
## 1st Qu.:69.31           1st Qu.: 375
## Median :80.71           Median : 750
## Mean   :77.93           Mean   : 750
## 3rd Qu.:87.94           3rd Qu.:1125
## Max.   :95.18           Max.   :1500

donnees

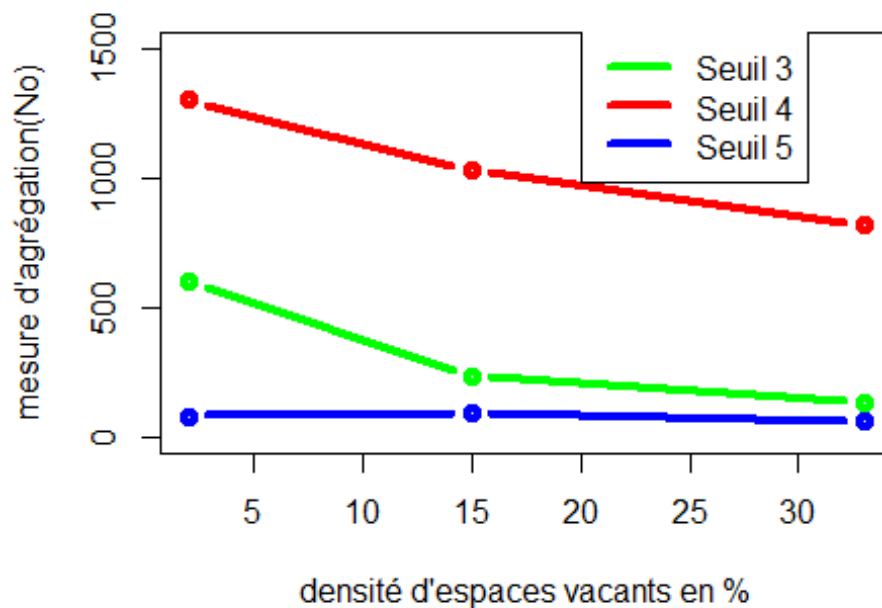
##      densité mesure agrégation seuil 3 mesure agrégation seuil 4
## 1          2                604.54                1301.01
## 2          15                235.07                1031.93
## 3          33                133.72                 817.81
##      mesure agrégation seuil 5 echelle
## 1                      80.71           0

```

## 2	95.18	750
## 3	57.90	1500

On réalise un graphe afin de pouvoir comparer toutes ces données :

```
plot(donnees$densité,donnees$echelle,type="b",lwd=7,xlab="densité d'espaces
vacants en %",ylab="mesure d'agrégation(No)",col="white")
lines(donnees$densité,donnees$`mesure agrégation seuil
3`,col="green",lwd=4,type="b")
lines(donnees$densité,donnees$`mesure agrégation seuil
4`,col="red",lwd=4,type="b")
lines(donnees$densité,donnees$`mesure agrégation seuil
5`,col="blue",lwd=4,type="b")
legend(20,1580, legend=c("Seuil 3", "Seuil 4","Seuil 5"), col=c("green",
"red","blue"), lty=1,lwd=4)
```



SEUIL 3 : Le N0 diminue lorsque la DENSITE augmente: 620.96 pour 2%, 236.92 pour 15% et 129.31 pour 33%.

SEUIL 4 : Le N0 diminue lorsque la DENSITE augmente : 1292.24 pour 2%, 1046.18 pour 15% et 95.25 pour 33%.

SEUIL 5 : Le N0 varie peu lorsque que la DENSITE augmente : 73.96 pour 2%, 95.25 pour 15% et 62.27 pour 33%.

Dans toutes les simulations le N0 est plus faible pour le SEUIL 5, un peu plus élevé pour le SEUIL 3 (même s'ils semblent se rapprocher du SEUIL 5 plus la DENSITE augmente). Le N0 du SEUIL 4 est quant à lui bien plus élevé que les autres.

## B-Interprétations:

On voit bien que la densité et le seuil de tolérance ont tous deux un impact sur la ségrégation des populations différentes.

En augmentant les espaces vides globalement le N0 diminue, il y a moins d'agents uniquement entourés d'agents du même type. Les agents ont moins de chances d'être entourés de "trop" d'agents de types différents. Mais il y a peu d'endroits où l'on retrouve assez d'agents du même type pour en être totalement entouré.

Le seuil 5 est exigeant, il est compliqué de trouver des espaces vacants qui satisferont le seuil pour les agents insatisfaits. Cela explique le faible N0. Peu d'agents auront 8 agents identiques autour d'eux. Il y a du mélange entre les populations, car de toute façon il est très compliqué pour un agent de trouver un endroit qui lui conviendra. On observe de nombreux petits "quartiers".

Le seuil 3 est peu exigeant, un agent ne demande "moins" de d'agents du même type. Il y a par conséquent plus d'espaces vacants qui conviendront aux agents insatisfaits. Le mélange entre populations sera tout de même un peu présent. On observe de nombreux quartiers légèrement plus importants que ceux du seuil 5.

Le seuil 4 est celui qui donnera le plus de "ségrégation" des populations. Le seuil est assez élevé pour que les agents de types différents cherchent à "s'éviter" mais il n'est pas trop élevé et n'empêche pas de trouver des espaces vacants satisfaisants. On observe moins de quartier, mais des quartiers de populations bien plus importants. Il y a très peu de "mélange".

## IV CONCLUSION

Pour conclure, on voit qu'en suivant le modèle de Schelling les données changent selon le seuil et la densité, mais que dans tous les cas les agents semblables ont tendance à se rassembler dans des zones entre eux ("quartiers"). Les décisions individuelles de chacun des agents entraînent une ségrégation spatiale collective.

On aurait aussi pu faire varier d'autres paramètres, telle que la taille de la matrice, la disposition de départ des agents (ici, on a choisi de les disposer aléatoirement.) mais aussi les tailles des différentes populations (on aurait pu mettre une population en infériorité numérique).