Fiche TD avec le logiciel  : ter4

Une variable et une carte

S. Dray & D. Chessel

La fiche regroupe quelques éléments de base concernant l'analyse de données spatialisées. Elle s'appuie essentiellement sur l'utilisation des paquets `ade4` et `spdep`.

Table des matières

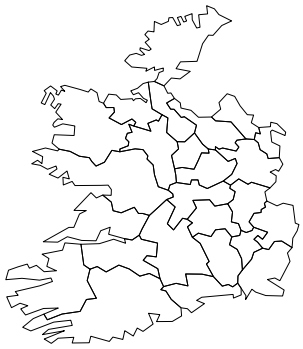
1 Introduction

Une grande partie des données acquises, en génétique, en écologie ou en biologie des populations est géoréférencée. Pour chaque individu échantillonné, on dispose d'une information spatiale sous la forme de coordonnées spatiales, entité surfacique. La gestion et la représentation cartographique de ces données est évoquée dans `http://pbil.univ-lyon1.fr/R/fichestd/ter5.pdf`. L'objectif de cette fiche est de présenter les principaux outils permettant la mise en évidence et la quantification de structures spatiales. Les principales mesures d'autocorrélation et les test associés sont évoqués.

2 Voisinage spatiale

En statistique spatiale, l'espace est défini par une relation de voisinage, donc une matrice qui a autant de lignes et de colonnes qu'il y a de points de mesures. Cette matrice contient à la ligne i et à la colonne j la valeur 1 si les points i et j sont voisins, 0 sinon. Dans `ade4`, les graphes sont de la classe `neig` mais la véritable librairie de R pour gérer les graphes de voisinage est `spdep`. On passe de l'un à l'autre par `neig2nb` et `nb2neig`. Par exemple, deux unités surfaciques sont voisines si elles ont une frontière commune (`poly2nb`). Un des articles fondateurs de ce domaine traite des comtés d'Irlande :

```
library(ade4)
data(irishdata)
names(irishdata)
 [1] "area"         "county.names" "xy"           "tab"          "contour"
 [6] "link"         "area.utm"     "xy.utm"       "link.utm"     "tab.utm"
[11] "contour.utm"
area.plot(irishdata$area.utm)
```



```
library(spdep)
ir.neig <- neig(area = irishdata$area.utm)
ir.neig
Carlow .
Cavan ..
Clare ...
Cork ....
Donegal .....
Galway ..1...
Kerry ...1...
Kildare 1.......
Kilkenny 1........
Laoghis 1......11.
```

```
Leitrim .1..1......
Limerick ..11..1.....
Longford .1........1..
Louth ..............
Mayo .....1.........
Meath .1.....1.....1..
Monaghan .1..........1.1.
Offaly .....1.1.1.....1..
Roscommon .....1....1.1.1..1.
Sligo ..........1...1...1.
Tipperary ..11.1..11.1.....1...
Waterford ...1....1..........1.
Westmeath .1..........1..1.11....
Wexford 1........1............1..
Wicklow 1......1..............1.
```

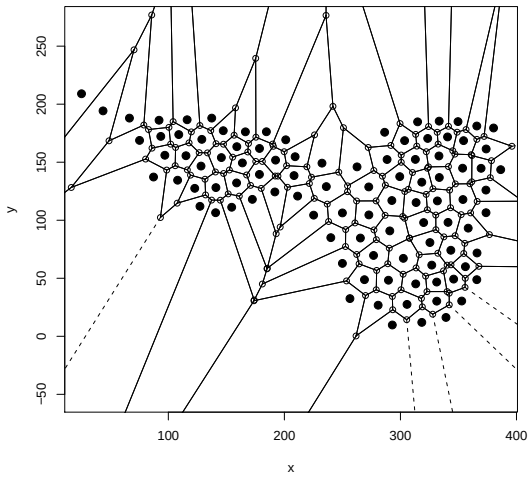
```
area.plot(irishdata$area.utm, graph = ir.neig)
ir.nb <- neig2nb(ir.neig)
ir.nb
```

```
Neighbour list object:
Number of regions: 25
Number of nonzero links: 108
Percentage nonzero weights: 17.28
Average number of links: 4.32
```

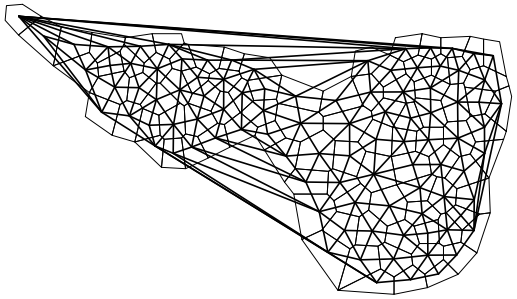
Les points sont les sommets du graphe, les paires de points sont les arêtes du graphe. On peut utiliser un graphe de voisinages pour exprimer la forme d'espaces particuliers comme les réseaux hydrographiques, les frontières infranchissables...

Dans le cas de données ponctuelles, il existe de nombreuses façons de définir le voisinage spatiale. Le pavage de Voronoi est à l'origine de plusieurs types de voisinage [Jaromczyk and Toussaint, 1992].

```
data(mafragh)
library(tripack)
plot(mafragh$xy, asp = 1, pch = 20, cex = 2)
plot(voronoi.mosaic(mafragh$xy), add = T)
```

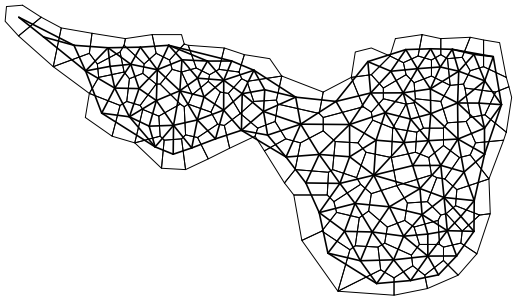



```
library(spdep)
maf1 <- tri2nb(mafragh$xy)
area.plot(mafragh$area, graph = nb2neig(maf1))
```

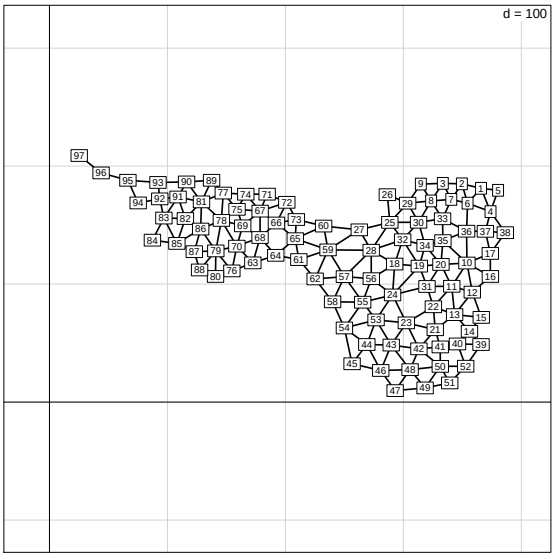


Après manipulation :

```
area.plot(mafragh$area, graph = mafragh$neig)
```



```
w <- gabrielneigh(as.matrix(mafragh$xy))
s.label(mafragh$xy, neig = nb2neig(graph2nb(w)), clab = 0.75)
```



Le graphe de Gabriel est un sous-graphe du graphe de Voronoi. Il est défini par : x et y sont voisins s'ils le sont au sens de la triangulation de Delaunay et si :

$$d(x,y) \leq min_z(\sqrt{d^2(x,z) + d^2(y,z)})$$

Deux points sont connectés si aucun autre point ne se trouve à l'intérieur du cercle de diamètre défini par ces 2 points [Gabriel and Sokal, 1969].

Définition, références dans la documentation de `gabrielneigh` dans `tripack`. `gabrielneigh` donne des objets de la classe `graph` (liste de couples de voisins

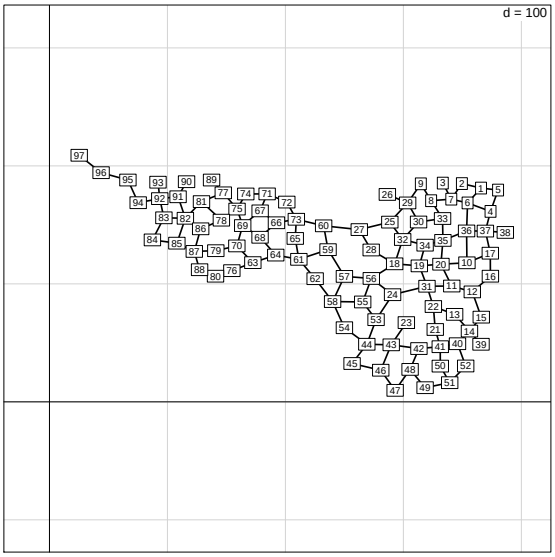
et coordonnées), `graph2nb` transforme les objets de la classe `graph` en objet de la classe `nb`.

Le graphe des voisins relatifs est aussi disponible. Dans ce graphe, x et y sont voisins s'ils le sont au sens de la triangulation de Delaunay et si :

$$d(x, y) \leq min_z(max(d(x, z), d(y, z)))$$

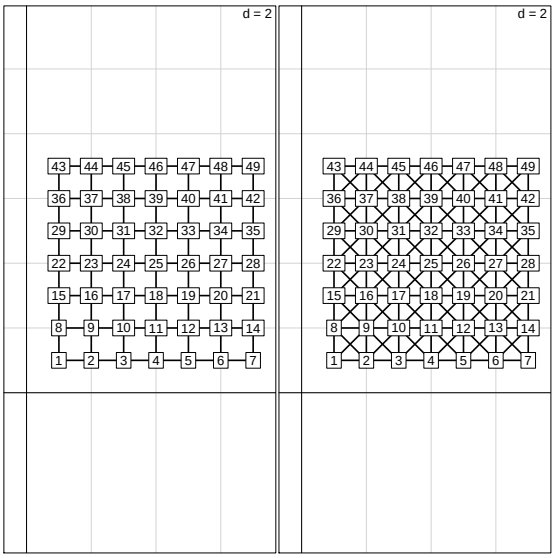
C'est un sous-graphe du précédent. Il est ici trop faible pour l'objectif.

```
w = relativeneigh(as.matrix(mafragh$xy))
s.label(mafragh$xy, neig = nb2neig(graph2nb(w)), clab = 0.75)
```



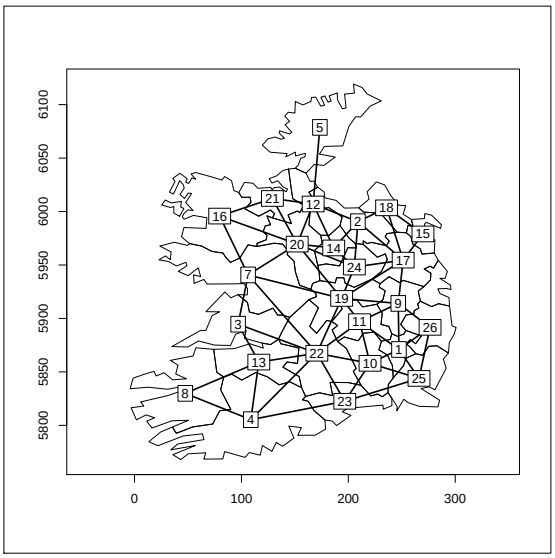
Mais pour travailler sur des grilles :

```
par(mfrow = c(1, 2))
s.label(expand.grid(1:7, 1:7), neig = nb2neig(cell2nb(7, 7, type = "rook")))
s.label(expand.grid(1:7, 1:7), neig = nb2neig(cell2nb(7, 7, type = "queen")))
```



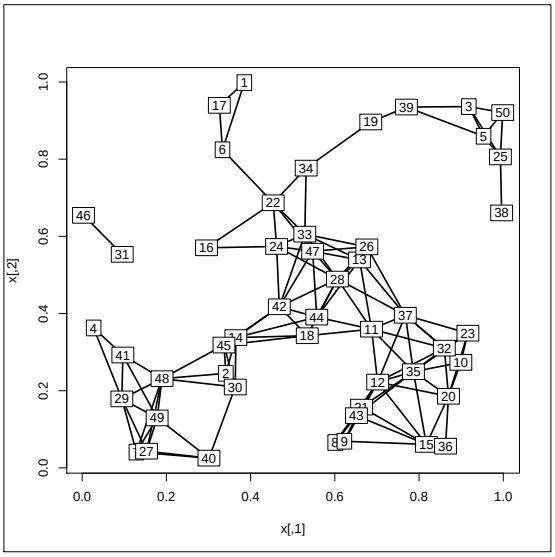
Pour les enregistrements surfaciques dans `spdep` :

```
data(eire)
maptools:::plot.polylist(eire.polys.utm)
s.label(eire.coords.utm[-6, ], add.plot = T, neig = ir.neig)
```



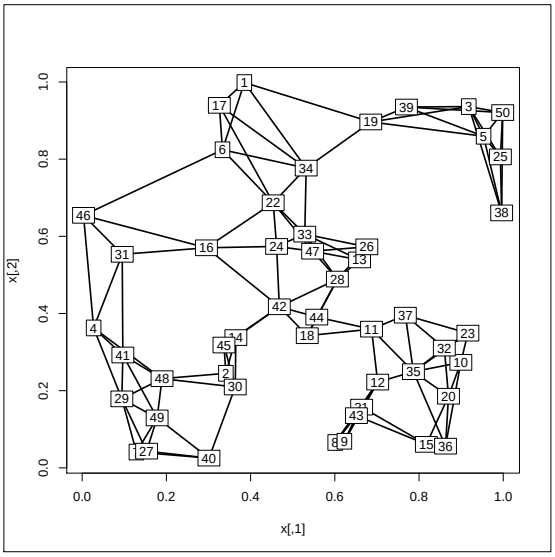
Tous les calculs sur graphes de voisinage utiliseront les structures de données de `spdep`. Passer les graphes de voisinage (`neig` de `ade4`) aux graphes de voisinage (`nb` de `spdep`) par `neig2nb`. Passer les fichiers `area` (`ade4`) aux listes de polygones (`spdep`) par `area2poly`.
Les représentations graphiques sont équivalentes dans les deux librairies, mais les concepts de poids de voisinage sont en oeuvre dans `spdep`.
On pourra aussi utiliser le voisinage par distance. Deux points sont voisins si et seulement si leur distance est supérieure à $d1$ et inférieure à $d2$:

```
x <- cbind(runif(50), runif(50))
plot(x)
w <- dnearneigh(x, 0, 0.2)
s.label(as.data.frame(x), add.p = T, neig = nb2neig(w))
```

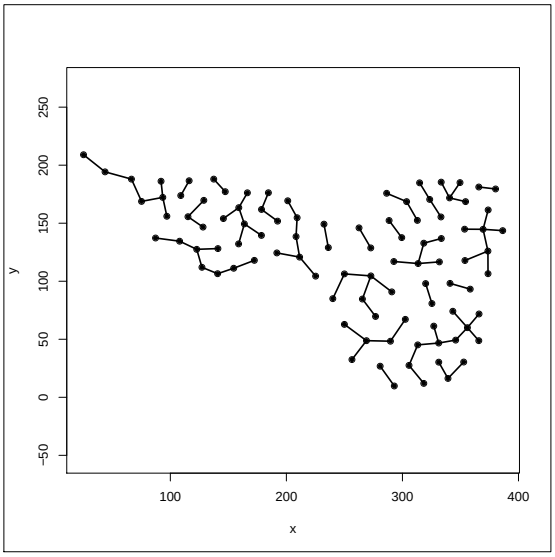


ou encore le voisinage par plus proches voisins :

```
w <- knn2nb(knearneigh(x, 4))
plot(x)
s.label(as.data.frame(x), add.p = T, neig = nb2neig(w))
```




```
plot(mafragh$xy, asp = 1)
s.label(mafragh$xy, add.p = T, neig = nb2neig(knn2nb(knearneigh(as.matrix(mafragh$xy),
    1))), clab = 0)
```



Le voisinage par les plus proches voisins conduit à un nombre constant de voisins, ce qui est un avantage (pondération uniforme par points) qu'on paye immédiatement par la non symétrie.

3 Pondération de voisinage

Une pondération de voisinage est toujours associée à un graphe de voisinage. Ce qui est pondéré, c'est le lien entre voisins. `spdep` propose les principales options dans ses procédures. Pondérer un voisinage est essentiellement une question pratique qui fournit une matrice $\mathbf{W}$ à n lignes et n colonnes telles que $w_{ij} \geq 0$ si i et j voisins, $w_{ij} = 0$ sinon. Dans un objet de la classe `listw`, on a d'abord une liste à n composantes qui sont des vecteurs donnant les numéros des voisins (on peut ou non tolérer des points sans voisins) puis une liste à n composantes qui sont des vecteurs donnant les poids des voisins.

Une remarque est très importante : la librairie de R. Bivand ne contient jamais de matrices et aucune des fonctions présentes ne manipule des matrices de voisinages (qui contiennent énormément de valeurs nulles). Ces fonctions n'ont donc pratiquement pas de limites en nombre de points, car elles n'utilisent que des listes de voisins et des listes de poids de voisinage. Les notations matricielles sont donc ici purement conceptuelles. Il y a au moins deux manières principales de pondérer pratiquement les voisinages. Le plus simple est de laisser agir la fonction `nb2listw`. Ces pratiques sont présentes dans l'ouvrage fondateur de Cliff and Ord [1973].

```
pond.w <- nb2listw(ir.nb, style = "W")
pond.b <- nb2listw(ir.nb, style = "B")
pond.c <- nb2listw(ir.nb, style = "C")
pond.u <- nb2listw(ir.nb, style = "U")
pond.s <- nb2listw(ir.nb, style = "S")
names(pond.w)
```

```
[1] "style"      "neighbours" "weights"
```

La fonction reprend le graphe et donne des poids aux arêtes. Il y a 5 options dont deux sont fortement utilisés :

W row standardised : l'option **W**, par défaut, donne un poids égal à l'inverse du nombre de voisins. La matrice W est alors de somme unité par ligne et nous l'appelerons **L** (pour profils lignes) :

```
pond.w$weights[1]
[[1]]
[1] 0.2 0.2 0.2 0.2 0.2
unlist(lapply(pond.w$weights, sum))
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
sum(unlist(pond.w$weights))
[1] 25
```

U globally standardised : Chaque couple de voisins à un poids égal à 1 divisé par le nombre total de voisins. La matrice **F** est alors de somme unité :

```
pond.u$weights[1]
[[1]]
[1] 0.00925926 0.00925926 0.00925926 0.00925926 0.00925926
unique(unlist(pond.u$weights))
[1] 0.00925926
sum(unlist(pond.u$weights))
[1] 1
```

On peut de plus importer directement une liste de poids, comme celle des longueur de frontières et transformer le résultat.

```
ir.list.w <- apply(irishdata$link.utm, 1, function(x) x[x != 0])
ir.w <- irishdata$link.utm
pond.ext.w <- nb2listw(ir.nb, glist = ir.list.w, style = "W")
pond.ext.u <- nb2listw(ir.nb, glist = ir.list.w, style = "U")
```

Pour l'option W :

```
sumlig <- apply(ir.w, 1, sum)
ir.w[1, ]/sumlig[1]
   Carlow     Cavan     Clare      Cork   Donegal    Galway     Kerry   Kildare
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.1054810
 Kilkenny   Laoghis   Leitrim  Limerick  Longford     Louth      Mayo     Meath
0.2648664 0.0914981 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 Monaghan    Offaly Roscommon     Sligo Tipperary Waterford Westmeath   Wexford
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.2129579
  Wicklow
0.3251965
pond.ext.w$weights[1]
[[1]]
  Kildare  Kilkenny   Laoghis   Wexford   Wicklow
0.1054810 0.2648664 0.0914981 0.2129579 0.3251965
unlist(lapply(pond.ext.w$weights, sum))
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
sum(unlist(pond.ext.w$weights))
[1] 25
```

Pour l'option U :

```
ir.w[1, ]/sum(ir.w)
     Carlow       Cavan       Clare        Cork     Donegal      Galway       Kerry
0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
    Kildare    Kilkenny     Laoghis     Leitrim    Limerick    Longford       Louth
0.003831792 0.009621765 0.003323838 0.000000000 0.000000000 0.000000000 0.000000000
       Mayo       Meath    Monaghan      Offaly   Roscommon       Sligo   Tipperary
0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
  Waterford   Westmeath     Wexford     Wicklow
0.000000000 0.000000000 0.007736091 0.011813367
pond.ext.u$weights[[1]]
    Kildare    Kilkenny     Laoghis     Wexford     Wicklow
0.003831792 0.009621765 0.003323838 0.007736091 0.011813367
sum(unlist(lapply(pond.ext.u$weights, sum)))
[1] 1
```

4 Autocorrélation

L'autocorrélation est la corrélation d'une variable avec elle-même, lorsque les observations sont considérées avec un décalage dans le temps (autocorrélation temporelle) ou dans l'espace (autocorrélation spatiale). Il y a autocorrélation positive quand des régions voisines tendent à avoir des valeurs semblables (ex : régions homogènes, gradients réguliers) ; l'autocorrélation est négative quand, dans des régions voisines, il y a alternance de valeurs fortes et faibles. Les mesures d'autocorrélation les plus utilisées sont celle de Moran [1948, 1950] et de Geary [1954].

4.1 Geary et Moran

n est le nombre de mesures (unités statistiques) et $\mathbf{W}$ est la matrice des poids de voisinages. x_i est la valeur de l'unité statistique i et $z_i = x_i - \bar{x}$. La notation classique est $\sum_{(2)} y_{ij} = \sum_{i,j=1;i\neq j}^{n} y_{ij}$.

Le I de Moran est en général :

$$I = \frac{n \sum_{(2)} w_{ij} z_i z_j}{\sum_{(2)} w_{ij} \sum_{i=1}^{n} {z_i}^2}$$

ce qui désigne quelquefois (définition **F**) :

$$I = \frac{\mathbf{z}^T \mathbf{F} \mathbf{z}}{\sum_{i=1}^{n} {z_i}^2 / n}$$

mais le plus souvent (somme par ligne de $\mathbf{L}$ vaut 1, la somme totale vaut n) :

$$I = \frac{\mathbf{z}^T \mathbf{L} \mathbf{z} / n}{\sum_{i=1}^{n} {z_i}^2 / n}$$

Le c de Geary vaut :

$$c = \frac{\sum_{(2)} w_{ij} (x_i - x_j)^2}{2 \sum_{(2)} w_{ij} \sum_{i=1}^{n} {z_i}^2 / (n-1)}$$

qui est utilisé souvent comme :

$$c = \frac{\sum_{(2)} f_{ij} (x_i - x_j)^2}{2 \sum_{i=1}^{n} {z_i}^2 / (n-1)}$$

4.2 *Moran scatterplot* et *lag vector*

L'indice de Moran utilisé avec la matrice $\mathbf{L}$ peut se réécrire :

$$I = \frac{\mathbf{z}^T \mathbf{L} \mathbf{z}}{\mathbf{z}^T \mathbf{z}} = \frac{\mathbf{z}^T \tilde{\mathbf{z}}}{\mathbf{z}^T \mathbf{z}}$$

$\tilde{\mathbf{z}} = \mathbf{L}\mathbf{z}$ contient, pour chaque observation, la moyenne des valeurs de la variable calculée sur les points voisins avec les poids relatifs de voisinage spatiale. On appelle I un coefficient d'autocorrélation bien que ce ne soit pas une corrélation (il faudrait que $\mathbf{z}$ et $\tilde{\mathbf{z}}$ soient normés), ni même une covariance (il faudrait que

$\mathbf{z}$ et $\tilde{\mathbf{z}}$ soient centrés, ce qui est vrai pour le premier, mais pas pour le second). $\tilde{\mathbf{z}} = \mathbf{Lz}$ est appelé le *lag vector* (vecteur retard). On le calcule à l'aide de la fonction `lag.listw` :

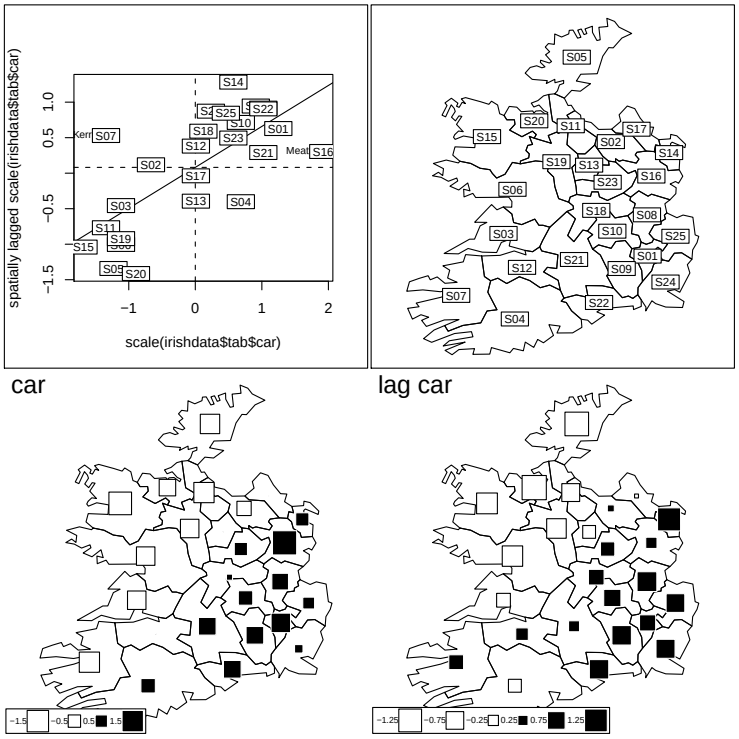
```
car.w <- lag.listw(pond.ext.w, scale(irishdata$tab$car))
as.vector(car.w)
 [1]  0.62853256  0.11807554 -0.45601242 -0.40129552 -1.34186935 -1.00344720
 [7]  0.53010515  0.94513792  0.91541293  0.71471974 -0.76799220  0.38119921
[13] -0.39395808  1.28224034 -1.04052777  0.30532894 -0.03709378  0.59106504
[19] -0.92147893 -1.41289362  0.29105566  0.90512630  0.49463874  0.87308924
[25]  0.84404585
```

Anselin [1996] propose d'étudier la relation entre $\mathbf{z}$ et $\tilde{\mathbf{z}}$ par une régression linéaire. Les résultats sont représentés sur un graphique bivarié, le *Moran scatterplot*. En abcisse, on place les valeurs d'une variable, en ordonnée la moyenne des valeurs des voisins (*lag vector*). La droite est l'estimation du modèle $\tilde{z} = az + b$. Les deux droites pointillées passent par les moyennes. Il y a 4 quadrants, on interprète :

- ⋆ fort-fort, faible-faible : groupement spatial
- ⋆ fort-faible, faible-fort : aberration spatiale

La pente reflète l'autocorrélation. La fonction `moran.plot` se charge du travail :

```
par(mfrow = c(2, 2))
moran.plot(scale(irishdata$tab$car), pond.ext.w)
w = cbind.data.frame(scale(irishdata$tab$car), car.w)
s.label(w, label = row.names(irishdata$tab), add.plot = T)
area.plot(irishdata$area.utm)
s.label(irishdata$xy.utm, label = row.names(irishdata$tab), add.plot = T)
area.plot(irishdata$area.utm)
s.value(irishdata$xy.utm, scale(irishdata$tab$car), add.plot = T,
    sub = "car", csub = 2)
area.plot(irishdata$area.utm)
s.value(irishdata$xy.utm, car.w, add.plot = T, sub = "lag car",
    csub = 2)
```



4.3 Tests

L'absence de structure spatiale est décrite par l'hypothèse nulle *z_i est la réalisation d'une variable aléatoire gaussienne de loi $N(\mu, \sigma^2)$* (modèle gaussien) ou par l'hypothèse nulle *les observations sont distribuées dans l'espace par tirage au hasard dans l'espace des $n!$ permutations des n premiers entiers* (modèle non paramétrique). Dans ce cas, on peut soit utiliser une approximation de la loi de la statistique basée sur les moments, soit générer des tirages aléatoires (test de randomisation).

Dans Cliff and Ord [1973], on trouve les principaux résultats. Pour l'indice de Moran (I), on a $E(I) = -1/(n-1)$, pour le c de Geary, $E(c) = 1$. L'autocorrélation positive se traduit par $I > -1/(n-1)$ ou $c < 1$. L'autocorrélation négative se traduit par $I < -1/(n-1)$ ou $c > 1$.

Pour faire le test de Geary, dans le modèle non paramétrique de l'équiprobabilité des $n!$ permutations des données :

```
geary.test(irishdata$tab$car, pond.ext.u)
```

```
        Geary's C test under randomisation
data:  irishdata$tab$car
weights: pond.ext.u

Geary C statistic standard deviate = 3.226, p-value = 0.0006277
alternative hypothesis: Expectation greater than statistic
sample estimates:
Geary C statistic       Expectation          Variance
       0.47582356        1.00000000        0.02640158
```

Pour le modèle gaussien :

```
geary.test(irishdata$tab$car, pond.ext.u, randomisation = FALSE)
```

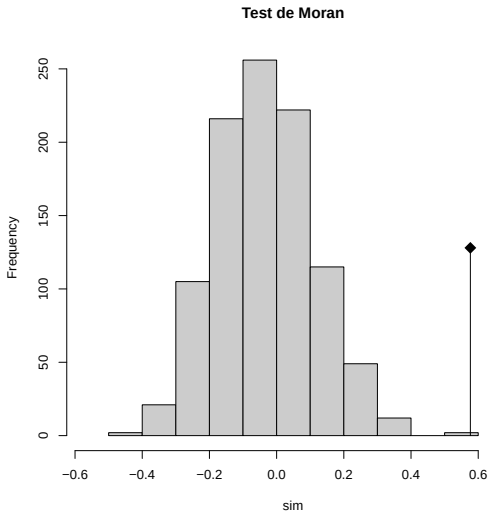
```
        Geary's C test under normality
data:  irishdata$tab$car
weights: pond.ext.u

Geary C statistic standard deviate = 2.8893, p-value = 0.001930
alternative hypothesis: Expectation greater than statistic
sample estimates:
Geary C statistic       Expectation          Variance
       0.47582356        1.00000000        0.03291267
```

Pour le test de Moran, dans le modèle non paramétrique de l'équiprobabilité des $n!$ permutations des données :

```
moran.test(irishdata$tab$car, pond.ext.w)
        Moran's I test under randomisation
data:  irishdata$tab$car
weights: pond.ext.w

Moran I statistic standard deviate = 4.2099, p-value = 1.277e-05
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
       0.57664986       -0.04166667        0.02157112
```

Pour la version Monte-Carlo :

```
t1 <- moran.mc(irishdata$tab$car, pond.ext.w, 999)
t1
        Monte-Carlo simulation of Moran's I
data:  irishdata$tab$car
weights: pond.ext.w
number of simulations + 1: 1000

statistic = 0.5766, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater


plot(as.randtest(t1$res, t1$statistic), main = "Test de Moran")
```



4.4 Indicateurs locaux

L'indice de Moran s'écrit

$$I = \frac{n \sum_{(2)} w_{ij} z_i z_j}{\sum_{(2)} w_{ij} \sum_{i=1}^{n} {z_i}^2}$$

.

Dans le *Moran scatterplot*, on considère les deux parties z_i et $\sum_{j=1;j\neq i}^{n} w_{ij}z_j$ pour l'ensemble des points i. On peut également décomposer cet indice, en considérant la quantité $z_i \sum_{j=1;i\neq j}^{n} w_{ij}z_j$ pour chaque point i. C'est ce que font les indicateurs locaux (il en existe également pour l'indice de Geary).

Anselin [1995] propose un *Local Indicator of Spatial Association* (LISA) qui est égal à :

$$I_i = \frac{z_i \sum_{j=1;j\neq i}^{n} w_{ij}z_j}{\sum_{k=1}^{n} {z_k}^2/n}$$

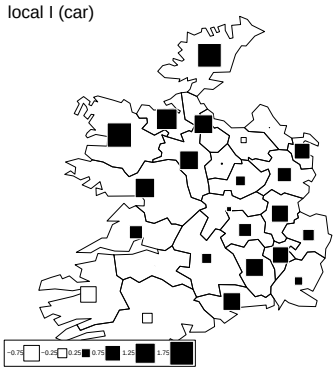
On a $I\frac{\sum_{(2)} w_{ij}}{n} = \sum_i I_i$.

Ce type d'indicateur peut être cartographié et servir à mettre en évidence des zones locales de forte autocorrélation positive ou négative. Les indices locaux sont testés par approximation normale, on peut ajuster les `pvalues` par une procédure permettant de tenir compte des tests multiples.

```
localM.car <- localmoran(scale(irishdata$tab$car), pond.ext.w)
head(localM.car)
```

```
           Ii        E.Ii     Var.Ii       Z.Ii   Pr(z > 0)
[1,]  0.81663932 -0.04166667 0.2010875  1.9140338 0.027807920
[2,] -0.08196801 -0.04166667 0.2138010 -0.0871595 0.534727632
[3,]  0.53045811 -0.04166667 0.4134942  0.8897255 0.186806640
[4,] -0.28610842 -0.04166667 0.3160647 -0.4347978 0.668145416
[5,]  1.71828666 -0.04166667 0.9622238  1.7941680 0.036393178
[6,]  1.16726360 -0.04166667 0.2120101  2.6255665 0.004325246
```

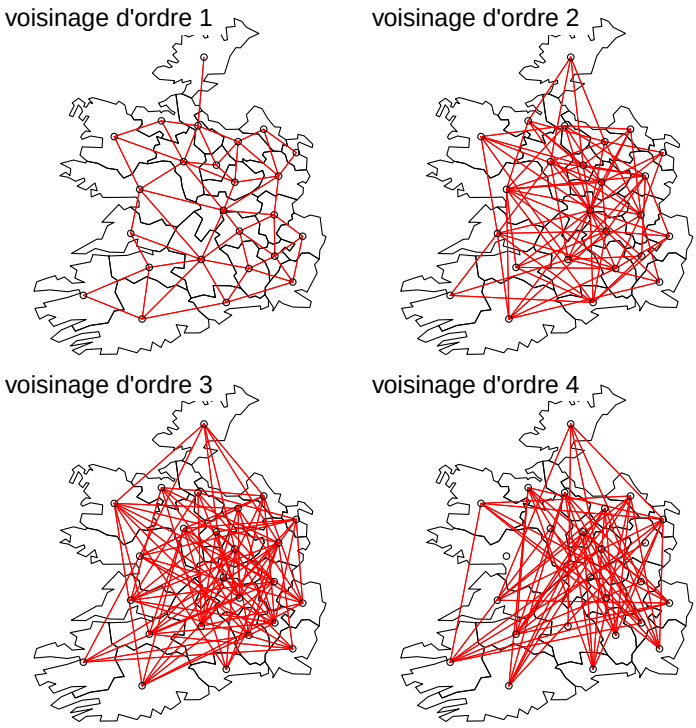
```
area.plot(irishdata$area.utm)
s.value(irishdata$xy.utm, localM.car[, 1], add.plot = T, sub = "local I (car)",
    csub = 2)
```



4.5 Corrélogramme

Lorsqu'on mesure l'autocorrélation, on mesure le lien entre la valeur observée en un point et les valeurs observées aux points voisins. Dans le graphe, les voisins sont définis par deux points reliés par une arête. On parle de voisinage d'ordre 1. On peut alors définir un voisinage d'ordre 2 qui consiste à prendre comme voisin d'un point, les voisins d'ordre 1 de de ses voisins d'ordre 1. Ainsi de suite. Dans R, la fonction `nblag` permet de calculer des voisinage d'ordre n :

```
par(mfrow = c(2, 2))
nb.ir.lag <- nblag(ir.nb, 4)
for (i in 1:4) {
    area.plot(irishdata$area.utm, sub = paste("voisinage d'ordre",
        i), csub = 2)
    plot(nb.ir.lag[[i]], irishdata$xy.utm, add = TRUE, col = "red")
}
```

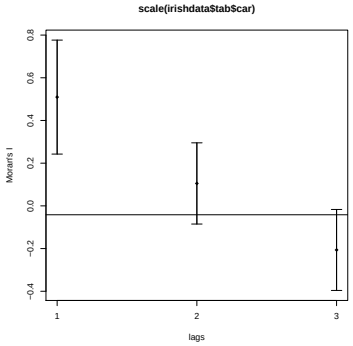


On peut alors regarder comment évolue un indice d'autocorrélation avec l'ordre du voisinage. Ce principe consiste à effectuer un corrélogramme. Il permet d'appréhender des structures plus complexes, opérant à différentes échelles. L'utilisation du corrélogramme sur les données d'Irlande a pour simple intérêt d'illustrer l'utilisation de la fonction car les structures sont simples et le nombre de régions est restreint :

```
sp.ir.cor <- sp.correlogram(ir.nb, scale(irishdata$tab$car), order = 3,
    method = "I")
sp.ir.cor
```

```
Spatial correlogram for scale(irishdata$tab$car)
method: Moran's I
    estimate expectation   variance standard deviate Pr(I) two sided
1  0.5095066  -0.0416667  0.0178121           4.1298       3.631e-05 ***
2  0.1050830  -0.0416667  0.0090418           1.5433         0.12276
3 -0.2064772  -0.0416667  0.0089932          -1.7379         0.08223 .
---
Signif. codes:  0
```

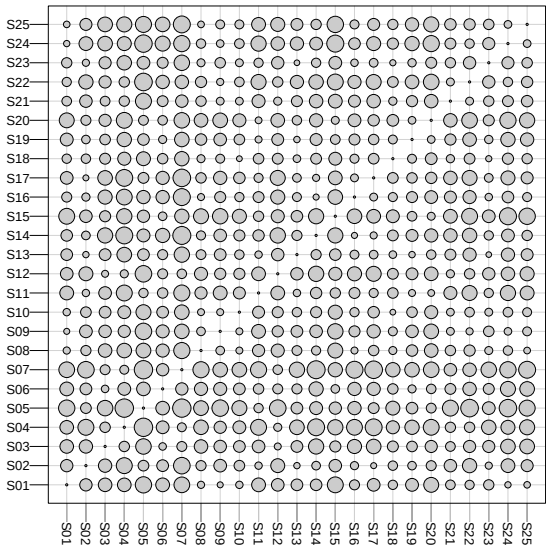
```
plot(sp.ir.cor)
```



5 Espace comme distances

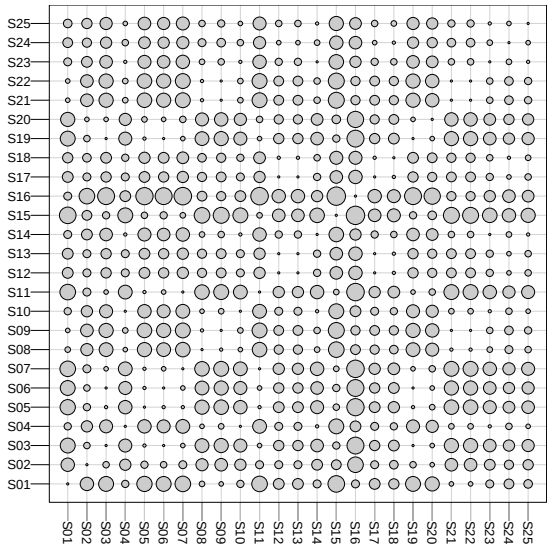
On peut voir l'espace comme des coordonnées spatiales (x, y), un graphe de voisinage. On peut également utilisée la notion de distance. Le cas le plus simple est celui de la distance euclidienne canonique : $d(A, B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$. On calcule les distances entre toutes les observations pour obtenir une matrice :

```
d.geo <- dist(irishdata$xy.utm)
table.dist(d.geo, csize = 0.5, label = row.names(irishdata$tab))
```



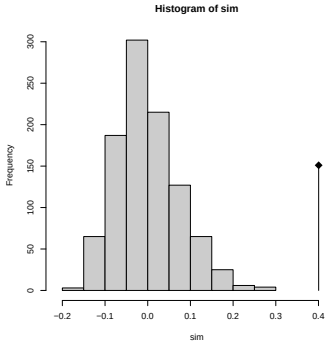
On peut également calculer une distance pour une variable quantitative, pour plus de détails voir Gower and Legendre [1986] :

```
d.car <- dist(scale(irishdata$tab$car))
table.dist(d.car, csize = 0.5, label = row.names(irishdata$tab))
```



Le test de Mantel [Mantel, 1967] permet de quantifier et de tester le lien entre deux matrices de distances. L'espace est connu par une matrice **S** de distances spatiales. Pour les données, on déduit une distance entre les individus consignée dans une matrice de distances **D**. La corrélation entre les deux est mesurée directement par $\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}d_{ij}$. Les couples ii ne jouent aucun rôle puisque les distances sont nulles. Peu importe également que l'on compte une fois ou deux fois les couple ij et ji . Seul importe le type de permutations utilisées. Une des matrices est laissée en place et dans l'autre, lignes et colonnes sont permutées à l'identique. Pour chacune des permutations de ce type, on calcule la statistique $\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}d_{ij}$ et on compare la valeur observée à l'ensemble des permutations. L'habitude veut que l'on corrige par les moyennes et les écarts-types pour faire apparaître exactement la corrélation entre les deux statistiques :

```
r1 <- mantel.randtest(d.geo, d.car)
r1
Monte-Carlo test
Call: mantel.randtest(m1 = d.geo, m2 = d.car)
Observation: 0.4004051

Based on 999 replicates
Simulated p-value: 0.001
Alternative hypothesis: greater

      Std.Obs   Expectation      Variance
5.4732801921 -0.0006517009  0.0053692839


plot(r1)
```



6 Espace comme tableau

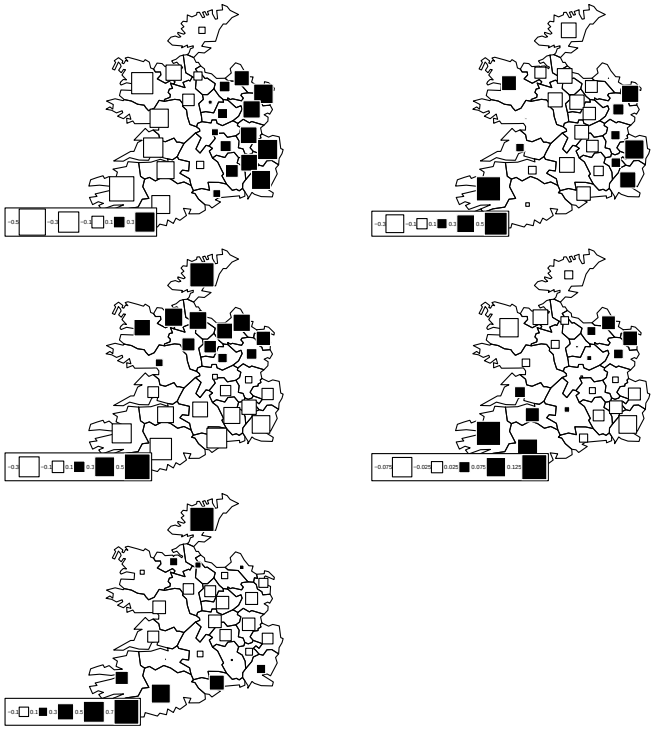
Le test de Mantel utilise une distance spatiale. Les tests de Geary et Moran utilisent un graphe de voisinage. On peut aussi tester le rôle de l'espace par une simple régression multiple sur les polynômes des coordonnées spatiales (*trend surface*) [Gittins, 1968].

```
 x <- irishdata$xy.utm[, 1]
 y <- irishdata$xy.utm[, 2]
 z <- scale(irishdata$tab$car)
 anova(lm(z ~ x + y + I(x * y) + I(x^2) + I(y^2)))
Analysis of Variance Table
Response: z
          Df  Sum Sq Mean Sq F value    Pr(>F)
x          1 10.8839 10.8839 37.6973 6.682e-06 ***
y          1  6.1491  6.1491 21.2977 0.0001890 ***
I(x * y)   1  0.3482  0.3482  1.2061 0.2858294
I(x^2)     1  1.0663  1.0663  3.6934 0.0697566 .
I(y^2)     1  0.0668  0.0668  0.2312 0.6361092
Residuals 19  5.4857  0.2887
---
Signif. codes:  0
```

On peut construire des polynômes orthogonaux, le résultat pour l'ensemble des variables ne change pas :

```
 polyxy2 <- poly(as.matrix(irishdata$xy.utm), degree = 2)
 anova(lm(z ~ polyxy2))
Analysis of Variance Table
Response: z
          Df  Sum Sq Mean Sq F value    Pr(>F)
polyxy2    5 18.5143  3.7029  12.825 1.524e-05 ***
Residuals 19  5.4857  0.2887
---
Signif. codes:  0
```

```
 par(mfrow = c(3, 2))
 for (i in 1:5) {
     area.plot(irishdata$area.utm)
     s.value(irishdata$xy.utm, polyxy2[, i], add.plot = T)
 }
```



Il existe d'autres alternatives basées sur le calcul des vecteurs propres d'un graphe de voisinage. On obtient des vecteurs (cartes) orthogonales maximisant l'indice de Moran [MÃ©ot et al., 1993, Borcard and Legendre, 2002, Griffith, 1996, Dray et al., 2006].

Références

L. Anselin. Local indicators of spatial association. *Geographical Analysis*, 27 : 93–115, 1995. Absent.

L. Anselin. The Moran scatterplot as an ESDA tool to assess local instability in spatial association. In M.M. Fischer, H.J. Scholten, and D. Unwin, editors, *Spatial analytical perspectives on GIS*, pages 111–125. Taylor and Francis, London, 1996. Absent.

D. Borcard and P. Legendre. All-scale spatial analysis of ecological data by means of principal coordinates of neighbour matrices. *Ecological Modelling*, 153 :51–68, 2002.

A.D. Cliff and J.K. Ord. *Spatial autocorrelation*. Pion, London, 1973.

S. Dray, P. Legendre, and P.R. Peres-Neto. Spatial modeling : a comprehensive framework for principal coordinate analysis of neighbor matrices (PCNM). *Ecological Modelling*, 196 :483–493, 2006.

K.R. Gabriel and R.R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18 :259–278., 1969.

R.C. Geary. The contiguity ratio and statistical mapping. *The incorporated Statistician*, 5 :115–145, 1954.

R. Gittins. Trend-surface analysis of ecological data. *Journal of Ecology*, 56 : 845–869, 1968.

J.C. Gower and P. Legendre. Metric and euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3 :5–48, 1986.

D. A. Griffith. Spatial autocorrelation and eigenfunctions of the geographic weights matrix accompanying geo-referenced data. *Canadian Geographer*, 40 (4) :351–367, 1996.

J.W. Jaromczyk and G.T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9) :1502–1517, 1992.

N. Mantel. The detection of disease clustering and a generalized regression approach. *Cancer Research*, 27(2) :209–220, 1967.

P.A.P. Moran. The interpretation of statistical maps. *Journal of the Royal Statistical Society Series B-Methodological*, 10 :243–251, 1948.

P.A.P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37 : 17–23, 1950.

A. MÃ©ot, D. Chessel, and R. Sabatier. OpÃ©rateurs de voisinage et analyse des donnÃ©es spatio-temporelles. In J.D. Lebreton and B. Asselain, editors, *BiomÃ©trie et environnement*, pages 45–72. Masson, Paris, 1993.