Fiche TD avec le logiciel R : tdr86

Analyser une carte, une variable, un tableau

A.B. Dufour & S. Dray

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, entités surfacique ou ponctuelle. 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 données étudiées tout au long du document concernent l'Irlande du Sud.

2 Les données

Les données à analyser ont été présentées dans Geary [1954]. Elles concernent les comtés de l'Irlande du Sud à l'exception du comté et de la ville de Dublin trop urbanisés et sont stockées dans `ade4`.

```
library(ade4)
library(adegraphics)
library(sp)
data(irishdata)
names(irishdata)
```

```
 [1] "area"            "county.names"    "xy"              "tab"
 [5] "contour"         "link"            "area.utm"        "xy.utm"
 [9] "link.utm"        "tab.utm"         "contour.utm"     "Spatial"
[13] "Spatial.contour"
```

```
s.Spatial(irishdata$Spatial)
```

L'objet `irishdata$tab` contient un tableau de 12 variables mesurées sur les 25 comtés.

```
dim(irishdata$tab)
```

```
[1] 25 12
```

```
names(irishdata$tab)
```

```
 [1] "T0.10"      "T10.50"     "Tup50"      "cow"        "other"      "pig"
 [7] "sheep"      "town.pop"   "car"        "radio"      "sales"      "single.man"
```

```
head(irishdata$tab)
```

```
    T0.10 T10.50 Tup50 cow other pig sheep town.pop car radio sales single.man
S01   318    469   213  67   252  56   531      402  43   169    66         603
S02   401    560    39  99   231  97    56      173  26    56    49         734
S03   388    544    68 110   285  32   116      244  22    67    28         683
S04   332    504   164 146   256 137   148      526  38   130    66         601
S05   698    259    42 102   248  22   463      189  21    80    45         624
S06   457    509    34  69   239  44   801      281  22    87    40         691
```

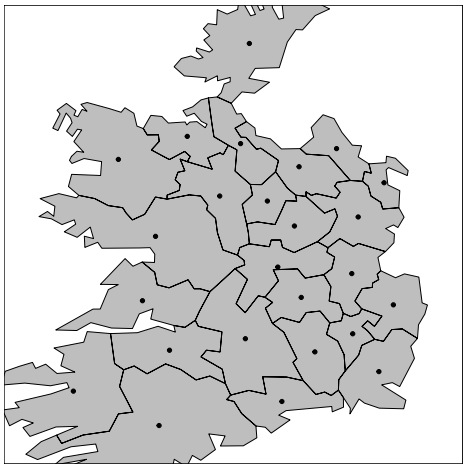
1. `T0.10` : Pourcentage de petites exploitations agricoles
2. `T10.50` : Pourcentage d'exploitations agricoles moyennes
3. `Tup50` : Pourcentage de grandes exploitations agricoles

4. `cow` : Nombre de vaches laitières pour 1000 acres de terres cultivables et de pâturage
5. `other` : Autre bétail pour 1000 acres de terres cultivables et de pâturage
6. `pig` : Nombre de cochons pour 1000 acres de terres cultivables et de pâturage
7. `sheep` : Nombre de moutons pour 1000 acres de terres cultivables et de pâturage
8. `town.pop` : Pourcentage d'habitants dans les villes et les villages relativement à la population totale du comté
9. `car` : Taux de voitures (privées) pour 1000 habitants
10. `radio` : Taux de radio (privées) pour 1000 habitants
11. `sales` : Vente au détail par livre et par personne
12. `single.man` : Pourcentage d'hommes célibataires relativement à l'ensemble des hommes, pour la tranche d'âge 30-34 ans.

Noter que les pourcentages et les taux ont été multiplés par 10.

L'enregistrement des données est de type ponctuel. Chaque comté est caractérisé par des coordonnées géographiques.

```
s.label(irishdata$xy.utm, plabel.cex=0, Sp=irishdata$Spatial, pgrid.draw=FALSE)
```



La liste des comtés avec leur position dans le fichier se trouve en annexe 1.

Partie I. Une carte

1 Le 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 R, la véritable librairie pour gérer les graphes de voisinage est `spdep` (classe `nb`).

```
library(spdep)
```

1.1 Unités surfaciques

Deux unités surfaciques sont voisines si elles ont une frontière commune (`poly2nb`) :

```
ir.nb <- poly2nb(irishdata$Spatial)
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
```

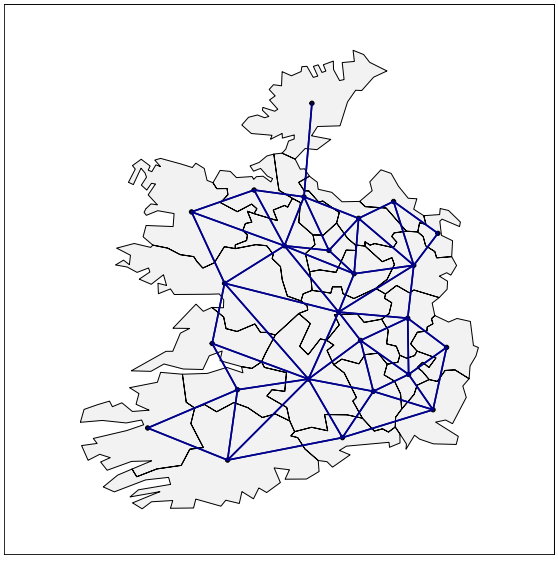
Il y a 25 comtés irlandais. Si tous les comtés étaient reliés entre eux, le nombre total de liens serait $25 \times 25 = 625$. En réalité, on dénombre 108 liens non nuls.

Avec une pondération uniforme $\left(\frac{1}{625}\right)$, le pourcentage de poids non nuls vaut $100 \times \frac{108}{625}$ soit 17.28%.

La moyenne des liens par comté vaut $\frac{108}{25} = 4.32$

Enfin, on peut visualiser ces relations entre comtés :

```
s.Spatial(irishdata$Spatial, nb = ir.nb, pSp.col = grey(0.95),
pnb.edge.lwd = 2, pnb.edge.col="blue4", plabels.cex=0)
```

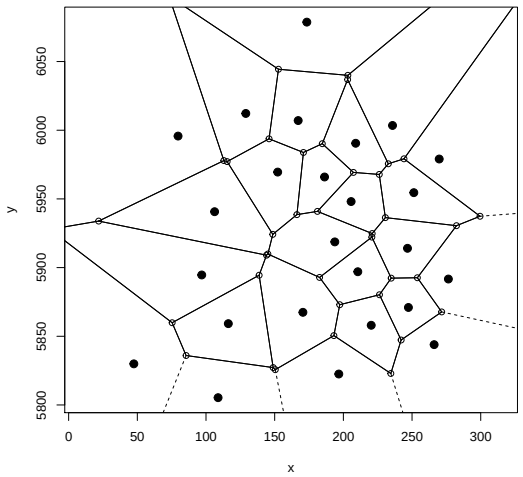


Les points sont les sommets du graphe, les paires de points sont les arêtes du graphe. On peut utiliser un graphe de voisinages pour exprimer la forme d'espaces particuliers comme les réseaux hydrographiques, les frontières infranchissables...

1.2 Unités ponctuelles

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

```
library(tripack)
coords <- irishdata$xy.utm
colnames(coords) <- c("x","y")
plot(coords,asp = 1, pch = 20, cex = 2)
plot(voronoi.mosaic(coords), add = T)
```

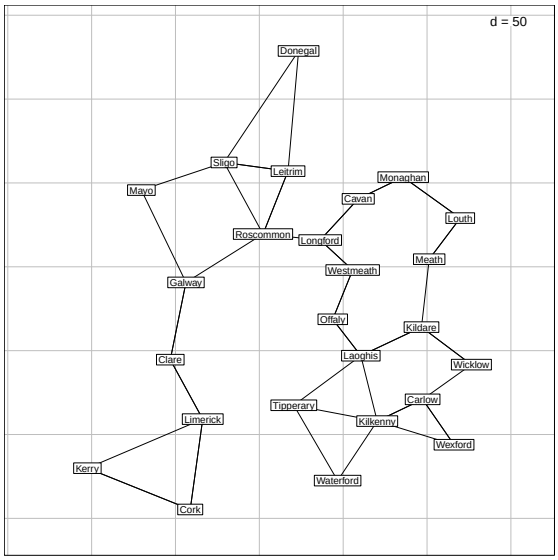


On considère les types de voisinage ci-dessous associés à des unités **ponctuelles** :

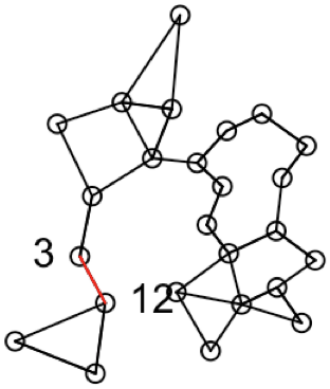
Graphe de voisinage	Fonction associée	Lien
Gabriel	`gabrielneigh()`	`graph2nb()`
voisins relatifs	`relativeneigh()`	`graph2nb()`
plus proches voisins	`knearneigh()`	`knn2nb()`

Si on prend, à titre d'exemple, le graphe associé aux deux plus proches voisins, on note que le comté de Clare et le comté de Limerick sont liés. Mais y-avait-il un pont au-dessus de la mer en 1950 ?

```
ir.nb2 <- knearneigh(as.matrix(coords),2)
s.label(coords, nb = knn2nb(ir.nb2), plabel.cex=0.75)
```



Il est possible d'enlever des voisins lorsque le lien n'a pas de sens. Pour ce faire, on peut utiliser la fonction `edit.nb`.




```
edit.nb(knn2nb(ir.nb2),coords)
Identifying contiguity for deletion ...
Delete this line (y/n) n
Options: quit[q] refresh[r] continue[c] r
Options: quit[q] continue[c]q
Neighbour list object:
Number of regions: 25
Number of nonzero links: 50
Percentage nonzero weights: 8
Average number of links: 2
Non-symmetric neighbours list
```

Exercice.

- Construire les graphes de voisinage de (1) Gabriel, (2) des voisins relatifs, (3) des plus proches voisins avec (a) 1 seul voisin et (b) 3 voisins.
- Représenter les graphes de voisinage (1), (2), (3a) et (3b). Discuter leurs ressemblances et leurs dissemblances.
- Choisir un des graphes de voisinage pour le reste de l'étude.

2 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 $w_{ij} > 0$ 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].

La fonction reprend le graphe défini au paragraphe 3.1 et donne des poids aux arêtes. Il y a 5 options (cf annexe 2 pour les détails) dont on retient les deux suivantes : **W** et **U**.

```
pond.w <- nb2listw(ir.nb, style="W")
pond.u <- nb2listw(ir.nb, style="U")
```

Si on reprend l'exemple de deux voisins `ir.nb2` et que l'on souhaite calculer les pondérations de voisinage associées à ces deux options, procéder comme suit :

```
pond.nb2.w <- nb2listw(knn2nb(ir.nb2), style="W")
pond.nb2.u <- nb2listw(knn2nb(ir.nb2), style="U")
```

Noter que si le message `Empty neighbour sets found` apparait, il faut rajouter l'instruction `zero.policy = TRUE`.

▶ Option **W**.
Elle donne un poids égal à l'inverse du nombre de voisins. La matrice **W** est alors de somme unité par ligne.

Le nombre de voisins du comté du Kerry (comté numéro 7, cf annexe 1) est :

```
pond.nb2.w$weight[7]
```

```
[[1]]
[1] 0.5 0.5
```

Comme on a construit ici deux voisins par comté, il est normal de retrouver deux pondérations. La somme en ligne valant 1, on a donc bien un poids de 0.5 pour chaque voisin.

Exercice.

1. Afficher la pondération associée au comté de Limerick avec la pondération conservée.
2. Retrouver que la somme par ligne vaut 1.

▶ Option **U**
Chaque comté a deux voisins. Le poids associé à un voisin est 1/50 où 50 est le nombre total de voisins dans l'étude (25×2).

```
pond.nb2.u$weight[7]
```

```
[[1]]
[1] 0.02 0.02
```

```
1/50
```

```
[1] 0.02
```

Exercice.

1. Afficher la pondération associée au comté de Limerick avec la pondération conservée.
2. Retrouver que la somme totale vaut 1.

Cas particulier
On peut aussi importer directement une liste de poids, comme celle des longueurs de frontières et transformer le résultat.

```
ir.w <- irishdata$link.utm
ir.list.w <- apply(ir.w, 1, function(x) x[x!=0])
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,]
```

```
  Carlow     Cavan     Clare      Cork   Donegal    Galway     Kerry   Kildare
 0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000  16.02926
Kilkenny   Laoghis   Leitrim  Limerick  Longford     Louth     Mayo      Meath
40.25003  13.90437   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
Monaghan    Offaly Roscommon     Sligo Tipperary Waterford Westmeath   Wexford
 0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000  32.36183
 Wicklow
49.41801
```

```
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
```

Le comté de Carlow (1) a des frontières communes avec les comtés de Laoghis, Kildare, Kilkenny, Wicklow et Wexford. Le poids attribué est alors proportionnel à la frontière commune entre Carlow et chacun des comtés.

```
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
```

```
sum(unlist(pond.ext.w$weights))
```

```
[1] 25
```

On retrouve, dans cette option, que la somme des poids associés à un comté vaut 1.

▶ 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
```

Dans ce cas, la pondération est calculée sur l'ensemble des frontières.

Partie II. Une carte, une variable

1 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].

Une fois l'indice choisi, on réalise un test statistique où l'hypothèse H_0 est l'absence de structure spatiale.

1.1 Indice de Geary

1. Si la variable à étudier Z est gaussienne, on réalise le test à l'aide de la fonction `geary.test` avec l'argument `randomisation = FALSE`.

2. Si la variable à étudier Z est non gaussienne, les observations sont distribuées dans l'espace par tirage au hasard dans l'espace des $n!$ permutations. On réalise le test à l'aide de la fonction `geary.test` avec l'argument `randomisation = TRUE`.

On considère la variable `car` taux de voitures (privées) pour 1000 habitants et les deux pondérations de voisinage : celle liée aux deux plus proches voisins qu'on a conservé à titre d'exemple (`pond.nb2.w`) et celle liée aux longueurs des frontières `pond.w.ext`

- La normalité de la variable `car` est vérifiée.

```
shapiro.test(irishdata$tab$car)
        Shapiro-Wilk normality test
data:  irishdata$tab$car
W = 0.93653, p-value = 0.123
```

- Le résultat donné par les liens entre les points-comtés est :

```
geary.test(irishdata$tab$car,pond.nb2.w, randomisation = FALSE)
        Geary C test under normality
data:  irishdata$tab$car
weights: pond.nb2.w

Geary C statistic standard deviate = 3.8423, p-value = 6.094e-05
alternative hypothesis: Expectation greater than statistic
sample estimates:
Geary C statistic       Expectation          Variance
       0.29311874        1.00000000        0.03384615
```

- Le résultat donné par les liens entre les frontières des comtés est :

```
geary.test(irishdata$tab$car,pond.ext.w, randomisation = FALSE)
        Geary C test under normality
data:  irishdata$tab$car
weights: pond.ext.w

Geary C statistic standard deviate = 3.5459, p-value = 0.0001956
alternative hypothesis: Expectation greater than statistic
sample estimates:
Geary C statistic       Expectation          Variance
        0.4596604         1.0000000         0.0232210
```

Dans les deux cas, on conclut à une autocorrélation spatiale entre les comtés pour la variable "taux de voitures (privées) pour 1000 habitants.

Exercice. Est-ce que le taux de vaches laitières pour 1000 acres de terres cultivables et de pâturage est une variable spatialisée ?

1.2 Indice de Moran

1. Si la variable à étudier Z est gaussienne, on réalise le test à l'aide de la fonction `moran.test` avec l'argument `randomisation = FALSE`.
2. Si la variable à étudier Z est non gaussienne, les observations sont distribuées dans l'espace par tirage au hasard dans l'espace des $n!$ permutations. On réalise le test à l'aide de la fonction `moran.test` avec l'argument `randomisation = TRUE`. Il existe également une version Monte-Carlo avec la fonction `moran.mc` où l'on précise le nombre de répétitions en dernier argument.

On considère toujours la variable `car` taux de voitures pour 1000 habitants et les deux pondérations de voisinage : celle liée aux points `pond.nb2.w` et celle liée aux longueurs des frontières `pond.w.ext`.

- La normalité de la variable `car` est vérifiée.

```
shapiro.test(irishdata$tab$car)
        Shapiro-Wilk normality test
data:  irishdata$tab$car
W = 0.93653, p-value = 0.123
```

- Le résultat donné par les liens entre les points-comtés est :

```
moran.test(irishdata$tab$car, pond.nb2.w, randomisation = FALSE)
        Moran I test under normality
data:  irishdata$tab$car
weights: pond.nb2.w

Moran I statistic standard deviate = 3.7639, p-value = 8.364e-05
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.60810839       -0.04166667        0.02980235
```

- Le résultat donné par les liens entre les frontières des comtés est :

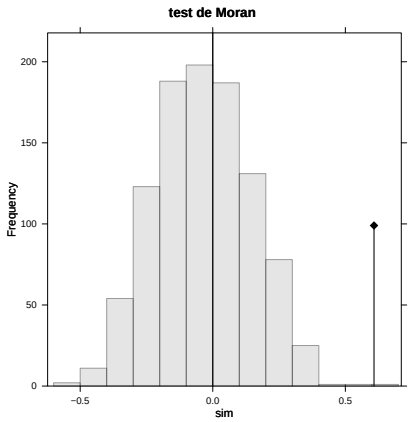
```
moran.test(irishdata$tab$car, pond.ext.w, randomisation = FALSE)
        Moran I test under normality
data:  irishdata$tab$car
weights: pond.ext.w

Moran I statistic standard deviate = 4.3012, p-value = 8.495e-06
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.57664986       -0.04166667        0.02066554
```

Dans les deux cas, on conclut à une autocorrélation spatiale entre les comtés pour la variable "taux de voitures (privées) pour 1000 habitants.

Noter que pour le test de Moran selon la méthode de Monte-Carlo, la distribution de la variable peut être représentée ainsi que la valeur observée.

```
resmo1 <- moran.mc(irishdata$tab$car, pond.nb2.w, 999)
plot(as.randtest(resmo1$res, resmo1$statistic), main="test de Moran")
```



Exercice. Est-ce que le taux de vaches laitières pour 1000 acres de terres cultivables et de pâturage est une variable spatialisée ?

1.3 La représentation de Moran et le lag vecteur

L'indice de Moran s'écrit en général :

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

mais le plus souvent sous la forme $\mathbf{L} = \left[\frac{w_{ij}}{w_{i\bullet}}\right]$ (la somme en lignes vaut 1 et la somme totale n) ou sous sa forme matricielle :

$$I = \frac{\mathbf{z}^\top \mathbf{L} \mathbf{z}}{\mathbf{z}^\top \mathbf{z}}$$

On pose $\tilde{\mathbf{z}} = \mathbf{L}\mathbf{z}$.
$\tilde{\mathbf{z}}$ contient, pour chaque observation, la moyenne des valeurs de la variable calculée sur les points voisins avec les poids relatifs du voisinage spatial.

Bien que ce ne soit ni une corrélation (il faudrait que $\mathbf{z}$ et $\tilde{\mathbf{z}}$ soient normés), ni une covariance ($\mathbf{z}$ est centré mais pas $\tilde{\mathbf{z}}$) I est appelé coefficient d'autocorrélation.

$\tilde{\mathbf{z}}$ est appelé le lag vecteur soit le vecteur retard. On le calcule à l'aide de la fonction :

```
car.w <- lag.listw(pond.nb2.w, scalewt(irishdata$tab$car))
as.vector(car.w)
```

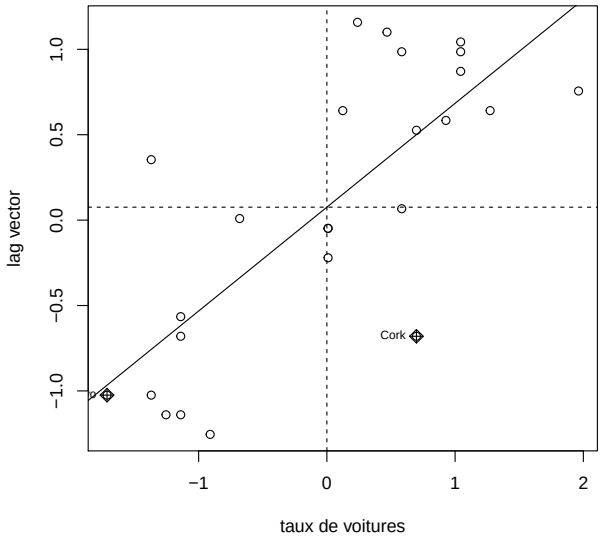
```
 [1]  0.641110014  0.009191541 -0.565279798 -0.680174065 -1.139751137 -1.139751137
 [7]  0.353874345  0.583662880  0.985792818  0.526215747 -1.024856869 -0.220596994
[13] -0.048255592  0.985792818 -1.024856869  0.756004282 -0.048255592  0.641110014
[19] -0.680174065 -1.254645404  0.870898550  1.043239952  0.066638675  1.158134219
[25]  1.100687085
```

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 abscisse, 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 :

```
moran.plot(as.vector(scalewt(irishdata$tab$car)), pond.nb2.w,
           labels=row.names(ir.nb2$x), xlab="taux de voitures", ylab="lag vector")
```

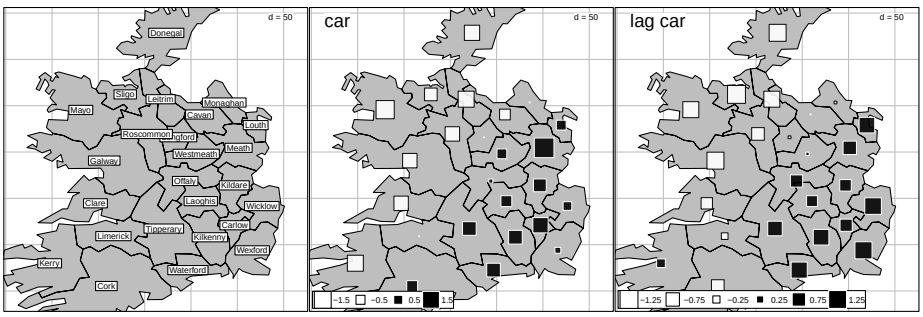


La représentation souligne que le comté de Cork a des voisins très différents tandis que le comté de Mayo a des voisins semblables.

```
which.min(as.vector(scalewt(irishdata$tab$car)))
[1] 15
row.names(ir.nb2$x)[15]
[1] "Mayo"
```

On retrouve ce constat dans les représentations ci-dessous.

```
geo <- s.label(coordinates(irishdata$Spatial), Sp = irishdata$Spatial, plot = FALSE)
gcar <- s.value(irishdata$xy.utm, scalewt(irishdata$tab$car), Sp = irishdata$Spatial,
       psub.text = "car", psub.cex = 2, plot = FALSE)
glag <- s.value(irishdata$xy.utm,car.w, Sp = irishdata$Spatial, psub.text = "lag car",
       psub.cex = 2, plot = FALSE)
ADEgS(list(geo, gcar, glag), layout = c(1,3))
```



Exercice. Etudier la représentation de Moran et le lag vecteur pour la pondération `pond.ext.w` et le taux de vaches laitières pour 1000 acres de terres cultivables et de pâturage.

1.4 Les indicateurs locaux

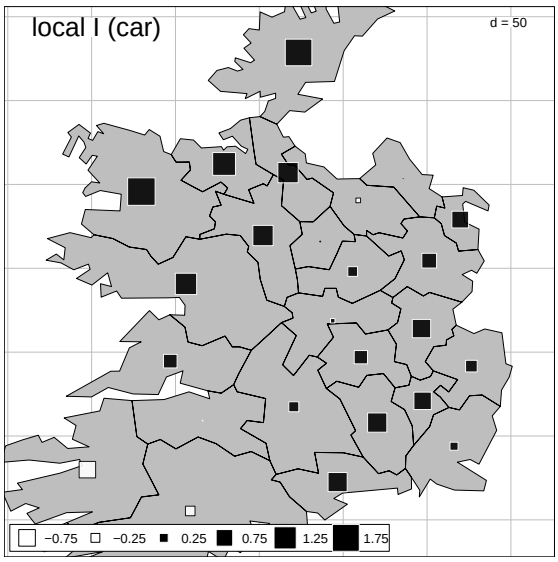
Ce type d'indicateur peut être cartographié et servir à mettre en évidence des zones locales de forte autocorrélation positive ou négative. Les indices locaux sont testés par approximation normale, on peut ajuster les p-values par une procédure permettant de tenir compte des tests multiples.
On définit l'Indice local de Moran par :

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

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

```
                Ii        E.Ii    Var.Ii        Z.Ii   Pr(z > 0)
Carlow   0.81663932 -0.04166667 0.2010875  1.91403378 0.027807920
Cavan   -0.08196801 -0.04166667 0.2138010 -0.08715949 0.534727632
Clare    0.53045811 -0.04166667 0.4134942  0.88972553 0.186806640
Cork    -0.28610842 -0.04166667 0.3160647 -0.43479785 0.668145416
Donegal  1.71828666 -0.04166667 0.9622238  1.79416802 0.036393178
Galway   1.16726360 -0.04166667 0.2120101  2.62556647 0.004325246
```

```
s.value(irishdata$xy.utm, localM.car[,1], Sp = irishdata$Spatial,
        psub.text = "local I (car)", psub.cex = 2)
```



Exercice. Est-ce que le taux de vaches laitières pour 1000 acres de terres cultivables et de pâturage est une variable localement spatialisée ?

Noter que $E(I_i)$ et $V(I_i)$ sont définies en annexe 3.

Partie III. Une carte, un tableau

1 Analyse d'un tableau

La synthèse d'un ensemble de variables contenu dans un tableau est assurée par une analyse multivariée. On note **X** le tableau, **Q** la pondération associée aux variables et **D** la pondération associée aux individus.

Soit, à titre d'exemple, le tableau **X** comprenant trois pourcentages associés à l'utilisation de la langue irlandaise dans les 25 comtés étudiés. Les données (https ://data.gov.ie/data) datent de 2011 et représentent, pour chaque comté, les valeurs moyennes des pourcentages :

1. d'irlandais aptes à parler leur langue `aptitude`,
2. d'irlandais parlé dans le cadre du système éducatif `dans_sys_educ`,
3. d'irlandais parlé hors système éducatif `hors_sys_educ`

```
langue <- read.table("https://pbil.univ-lyon1.fr/R/donnees/irishspeaking2011.txt",
                     header=TRUE, row.names=1)
head(langue)
```

```
        dans_sys_educ hors_sys_educ aptitude
Carlow          29.38          1.71    37.82
Cavan           33.65          1.53    36.24
Clare           27.59          2.04    47.59
Cork            28.61          2.12    46.10
Donegal         32.02          7.96    39.78
Galway          26.38          8.16    50.51
```

On réalise une analyse en composantes principales normée et on obtient un résumé des données par une ACP :

```
acplang <- dudi.pca(langue, scannf=FALSE, nf=3)
summary(acplang)
```

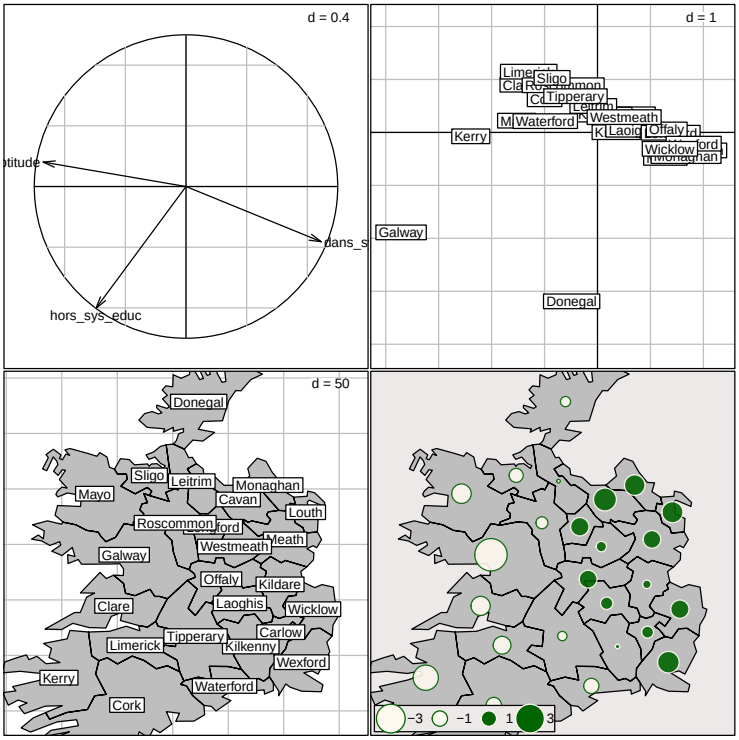
```
Class: pca dudi
Call: dudi.pca(df = langue, scannf = FALSE, nf = 3)
Total inertia: 3

Eigenvalues:
    Ax1     Ax2     Ax3
 2.0344  0.8023  0.1633

Projected inertia (%):
    Ax1     Ax2     Ax3
 67.815  26.743   5.442

Cumulative projected inertia (%):
    Ax1   Ax1:2   Ax1:3
  67.81   94.56  100.00
```

```
gl1 <- s.corcircle(acplang$co, plabel.boxes.draw=FALSE, plot=FALSE)
gl2 <- s.label(acplang$li, plot=FALSE)
gl3 <- s.label(irishdata$xy.utm, Sp = irishdata$Spatial, plot=FALSE)
gl4 <- s.value(irishdata$xy.utm, acplang$li[,1], Sp = irishdata$Spatial, method = "size",
        symbol = "circle", col = c("floralwhite","darkgreen"), pbackground.col = "snow2",
        pgrid.draw=FALSE,         plot=FALSE)
ADEgS(c(gl1,gl2,gl3,gl4), layout=c(2,2))
```



D'une façon plus générale, $\mathbf{X}$ peut être un `data frame` quelconque contenant des variables quantitatives (`numeric`) et des variables qualitatives (`factor`) voire même des qualitatives à modalités ordonnées (`ordered`) voire encore un mélange des deux. Les analyses à un tableau associées à ces différents cas sont l'analyse en composantes principales (`dudi.pca`), l'analyse des correspondances multiple (`dudi.acm`) et l'analyse de Hill-Smith (`dudi.hillsmith`).

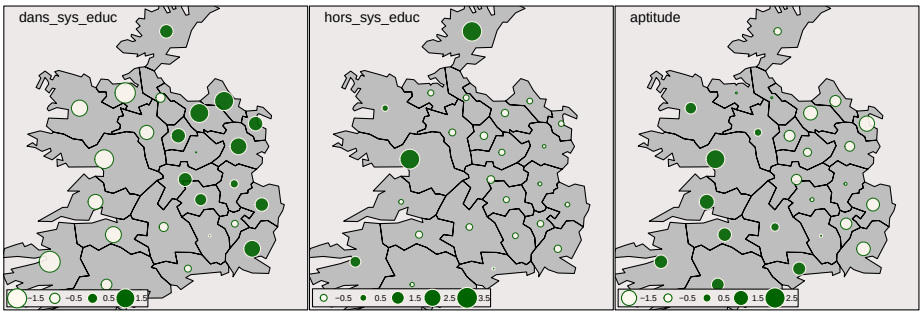
Exercice.
On considère le tableau `irishdata$tab` décrit au paragraphe 2 de l'introduction générale.

- Réaliser l'ACP normée du tableau. Donner le nombre d'axes retenus et le résumé statistique associé.
- Représenter le cercle des corrélations des axes 1 et 2.
- Représenter le premier plan factoriel.
- Représenter le premier et le second axes sur la carte spatialisée.

2 Structure spatiale

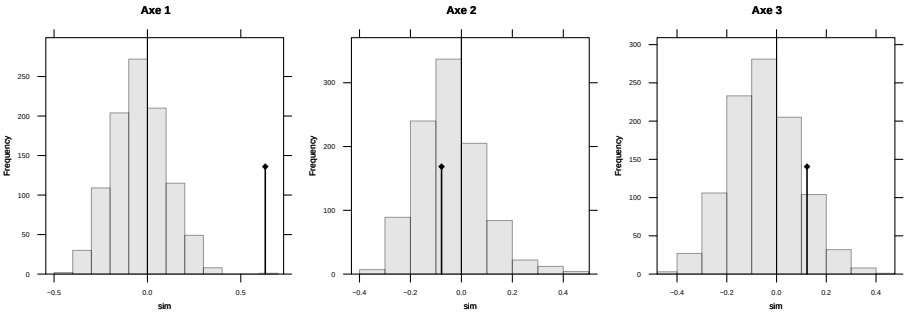
La distribution spatiale des trois pourcentages peut être visualisée :

```
s.value(irishdata$xy.utm, acplang$tab, Sp = irishdata$Spatial, method = "size", psub.cex=1.5,
        symbol = "circle", col = c("floralwhite", "darkgreen"), pbackground.col = "snow2",
                pgrid.draw=FALSE)
```



La question est de savoir si on peut faire une synthèse de ces distributions spatiales. En prenant comme relation de voisinage `pond.ext.w`, on calcule et teste la mesure d'autocorrélation sur les axes de l'ACP normée.

```
testmoran <- function(x) {
w <- moran.mc(acplang$l1[,x], pond.ext.w, 999)
plot(as.randtest(w$res, w$statistic), main=paste("Axe",x), plot=FALSE)
}
grm1 <- testmoran(1)
grm2 <- testmoran(2)
grm3 <- testmoran(3)
ADEgS(c(grm1, grm2, grm3), layout=c(1,3))
```



Le second axe de l'ACP représente 26.74% de l'inertie totale. Représenter le test de Moran sous forme bivariée donne une information complémentaire.

```
w <- as.data.frame(apply(acplang$li, 2, function(var) lag.listw(x=pond.ext.w,var)))
row.names(w) <- row.names(acplang$li)
s.match(acplang$li, w, plabel.cex=0.75)
```

Chaque comté est représenté par une flèche. Le début de la flèche (•) positionne le comté sur le premier plan de l'ACP ; l'extrêmité de la flèche positionne le comté lorsque l'information spatiale a été introduite c'est-à-dire à la moyenne des positions de ces voisins.

Les résultats pour le comté de Galway (6) sont :

```
acplang$li[6,1:2]
```

```
          Axis1     Axis2
Galway -3.709344 -1.887922
```

```
#
pond.ext.w$weights[[6]]
```

```
    Clare       Mayo     Offaly  Roscommon  Tipperary
0.24405426 0.29440136 0.06363695 0.30431541 0.09359202
```

```
voisGalway <- c(3,15,18,19,21)
posvois1 <- acplang$li[voisGalway,1]
posvois2 <- acplang$li[voisGalway,2]
sum(pond.ext.w$weights[[6]]*posvois1)
```

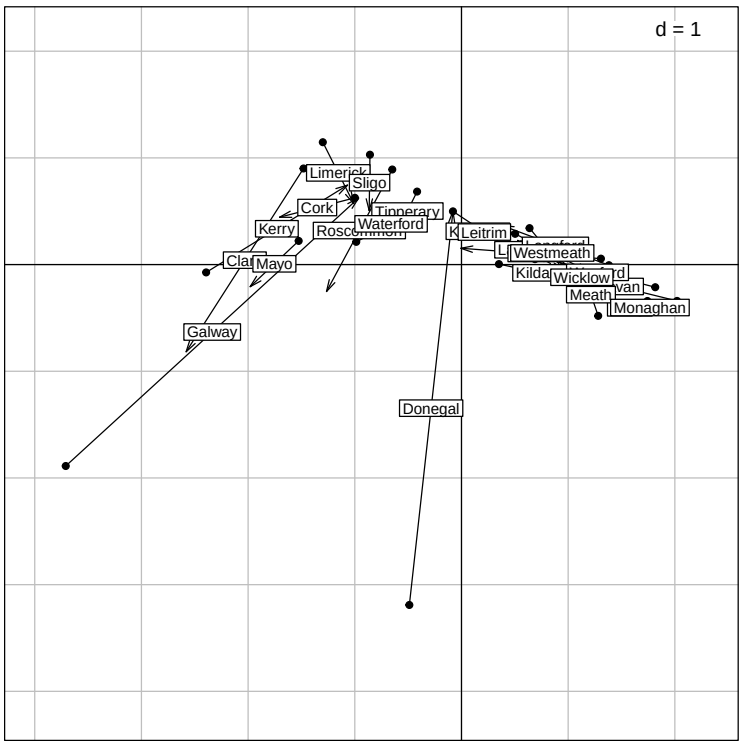
```
[1] -0.9637989
```

```
sum(pond.ext.w$weights[[6]]*posvois2)
```

```
[1] 0.6238494
```

```
#
w[6,1:2]
```

```
            Axis1     Axis2
Galway -0.9637989 0.6238494
```



Exercice.
On considère les résultats de l'ACP du tableau `irishdata$tab`.

- Représenter la structure spatiale des 12 variables.
- Réaliser les tests d'autocorrélation sur les 4 premiers axes de l'analyse avec la pondération `pond.ext.w`.
- Réaliser le *Moran scatterplot* associé au premier plan factoriel. Discuter.

Dans ce paragraphe, on a réalisé une synthèse du tableau puis on a cherché une structure spatiale de cette synthèse. On a donc une approche indirecte qui n'est pas forcément optimale (une structure forte n'est pas forcément spatialisée). L'ordination sous contrainte spatiale a pour objectif de faire une synthèse (analyse multivariée) de structures spatiales (autocorrélation). Ces deux objectifs devant être satisfaits simultanément.

3 La méthode multispati

3.1 L'analyse

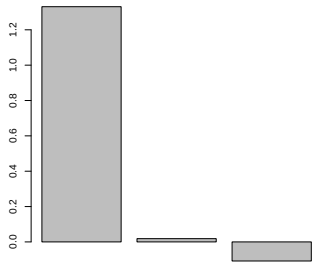
La fonction utilise un quadruplet $\left(\underset{n\times p}{\mathbf{X}}, \underset{p\times p}{\mathbf{Q}}, \underset{n\times n}{\mathbf{D}}, \underset{n\times n}{\mathbf{L}}\right)$ dont les dimensions sont indiquées en associant un schéma de dualité $(\mathbf{X}, \mathbf{Q}, \mathbf{D})$ - objet de la classe `dudi` - et une pondération de voisinage $\mathbf{L}$ - objet de la classe `listw`).

Au triplet classique d'une analyse multivariée est donc associé un opérateur de retard $\mathbf{L}$ qui permet de calculer $\mathbf{Y} = \mathbf{LX}$ où chaque valeur initiale au point i de la variable j est remplacée par la moyenne des valeurs des voisins de i pour la même variable j. Pour une variable on a $\mathbf{y} = \mathbf{Lx}$ et le graphe du couple $(\mathbf{x}, \mathbf{y})$ est le *Moran scatterplot* d'Anselin.

Ainsi étendu, l'opération génère un deuxième tableau totalement apparié au premier et donc un deuxième nuage de n points de $\mathbb{R}^p$ qu'on peut projeter sur les axes principaux.

Un analyse ordinaire maximise la variance projetée. L'analyse sous contrainte spatiale garde une part de cette propriété mais intègre le lien de voisinage.

```
library(adespatial)
acpsp <- multispati(acplang, pond.ext.w, scannf = FALSE, nfposi = 2, nfnega = 0)
barplot(acpsp$eig)
```



On compare l'analyse simple et l'analyse spatialisée grâce au résumé statistique.

```
summary(acpsp)
```

```
Multivariate Spatial Analysis
Call: multispati(dudi = acplang, listw = pond.ext.w, scannf = FALSE,
    nfposi = 2, nfnega = 0)
Scores from the initial duality diagram:
       var      cum     ratio      moran
RS1 2.0344429 2.034443 0.6781476  0.63099245
RS2 0.8022873 2.836730 0.9455767 -0.07790907
RS3 0.1632698 3.000000 1.0000000  0.12255809

Multispati eigenvalues decomposition:
         eig       var     moran
CS1 1.33077143 1.9930904 0.6676924
CS2 0.01842294 0.1672601 0.1101455
```

La première partie du résumé est associée à l'analyse en composantes principales normée et comprend : les valeurs propres (les variances des `acplang$li`), les valeurs propres cumulées, les inerties et les indices de Moran par axe.
La seconde partie du résumé est associée à l'analyse spatiale et comprend : les valeurs propres positives, les variances des `acpsp$li`, les indices de Moran par axe.

```
 acpsp$eig
[1]  1.33077143  0.01842294 -0.10797168
 vardes <- function(x) sum((x-mean(x))^2)/length(x)
 sapply(acpsp$li, vardes)
      CS1       CS2
1.9930904 0.1672601
 moran.mc(acpsp$li[,1], pond.ext.w, 999)
        Monte-Carlo simulation of Moran I
data:  acpsp$li[, 1]
weights: pond.ext.w
number of simulations + 1: 1000

statistic = 0.66769, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
 moran.mc(acpsp$li[,2], pond.ext.w, 999)
        Monte-Carlo simulation of Moran I
data:  acpsp$li[, 2]
weights: pond.ext.w
number of simulations + 1: 1000

statistic = 0.11015, observed rank = 859, p-value = 0.141
alternative hypothesis: greater
```

Les variances de l'ACP normée sont plus grandes que les variances de l'ACP spatialisée :

Axe 1	2.0344	1.9931
Axes 1 et 2	2.8367	$1.9931 + 0.1673 = 2.1604$

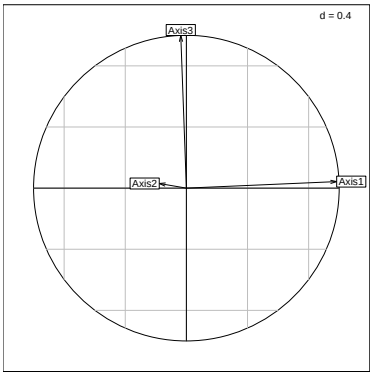
mais on constate un petit effet spatial :

$$1.3308 > 2.0344 \times 0.6310 = 1.2837$$

3.2 Les représentations graphiques

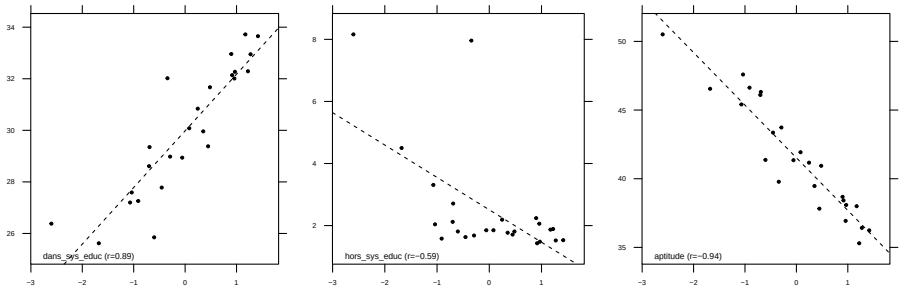
On peut représenter la projection des trois axes de l'analyse simple (ACP normée) sur le plan des deux premiers axes de l'analyse spatialisée.

```
 s.corcircle(acpsp$as)
```

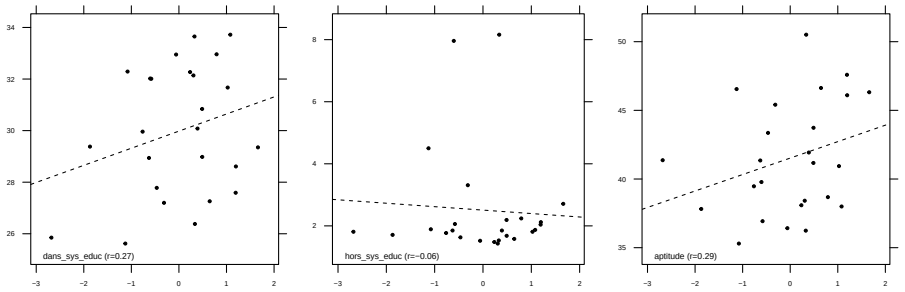


L'axe 1 de l'ACP normée est fortement corrélé à l'axe 1 de l'analyse spatialisée.
L'axe 3 de l'ACP normée est fortement corrélé à l'axe 2 de l'analyse spatialisée.

```
score(acplang, xax=1)
```

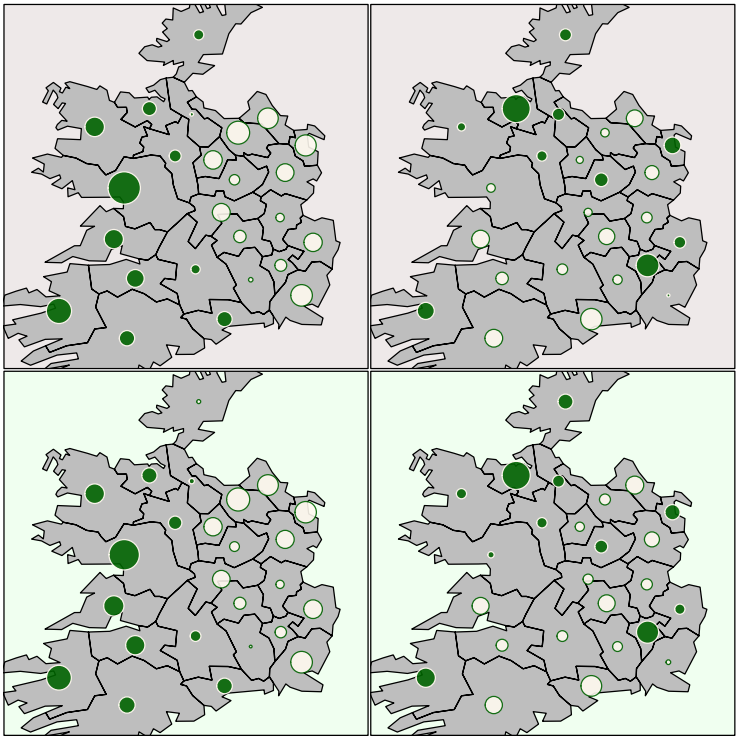



```
score(acplang, xax=3)
```



On représente en haut la cartographie des axes 1 et 3 de l'ACP simple (fond crème) et en bas les axes 1 et 2 de l'ACP spatialisée (fond vert).

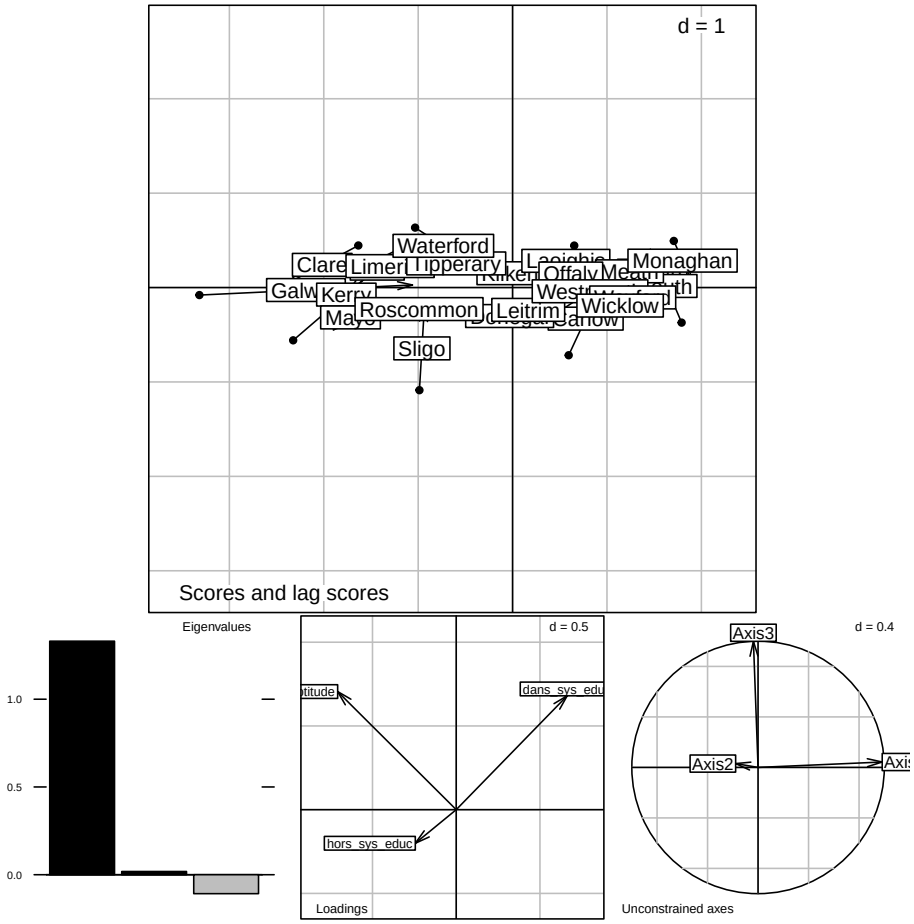
```
gnsp1 <- s.value(irishdata$xy.utm, acplang$li[,1], Sp = irishdata$Spatial,
                 method = "size", symbol = "circle",
                 col = c("darkgreen", "floralwhite"), pbackground.col = "snow2",
                 pgrid.draw=FALSE, plegend.drawKey=FALSE, plot=FALSE)
gnsp2 <- s.value(irishdata$xy.utm, acplang$li[,3], Sp = irishdata$Spatial,
                 method = "size", symbol = "circle",
                 col = c("darkgreen", "floralwhite"), pbackground.col = "snow2",
                 pgrid.draw=FALSE, plegend.drawKey=FALSE, plot=FALSE)
gsp1 <- s.value(irishdata$xy.utm, acpsp$li[,1], Sp = irishdata$Spatial,
                method = "size", symbol = "circle",
                col = c("darkgreen", "floralwhite"), pbackground.col = "honeydew",
                pgrid.draw=FALSE, plegend.drawKey=FALSE, plot=FALSE)
gsp2 <- s.value(irishdata$xy.utm, acpsp$li[,2], Sp = irishdata$Spatial,
                method = "size", symbol = "circle",
                col = c("darkgreen", "floralwhite"), pbackground.col = "honeydew",
                pgrid.draw=FALSE, plegend.drawKey=FALSE, plot=FALSE)
ADEgS(c(gnsp1,gnsp2,gsp1,gsp2), layout=c(2,2))
```



`multispati` possède une représentation graphique globale comprenant 4 graphiques :

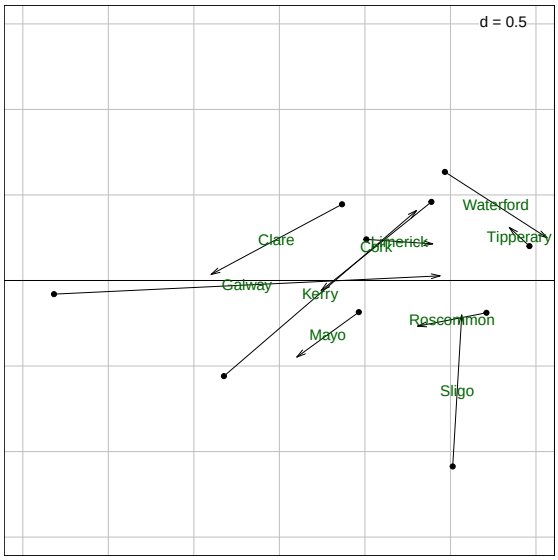
1. la représentation des comtés : le début de la flèche donne la position du comté sur l'axe de l'ACP normée, l'extrêmité de la flèche donne la position du comté avec l'introduction spatiale,
2. le graphe des valeurs propres,
3. les coefficients des variables initiales dans la construction des axes spatialisés,
4. la projection des axes de l'ACP normée sur les axes de l'analyse spatialisée.

```
plot(acpsp)
```

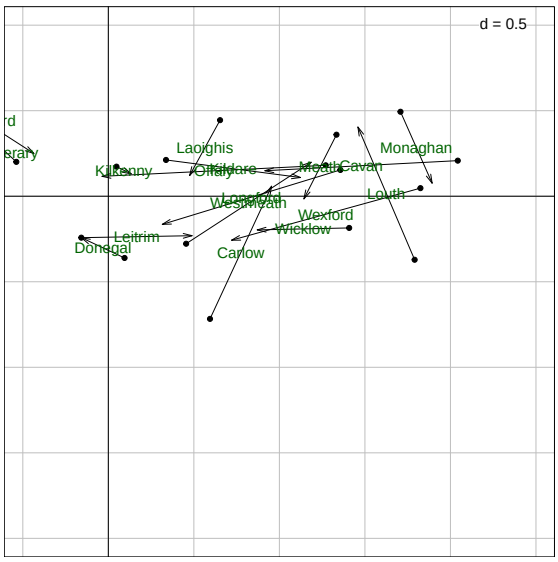

d = 1
Waterford
Clare
Limer
Tipperary
Monaghan
Kerry
Roscommon
Leitrim
Wicklow
Sligo
Scores and lag scores
Eigenvalues
1.0
0.5
0.0
d = 0.5
dans_sys_edu
hors_sys_educ
Loadings
d = 0.4
Axis3
Axis2
Unconstrained axes

avec un agrandissement de la première figure (fonction `zoom`) :

```
gspli <- s.match(acpsp$li, acpsp$ls, plabel.boxes.draw=FALSE, plabel.col="darkgreen",
                 plot=FALSE)
zoom(gspli, 2, center=c(-2,0))
```




```
zoom(gspli, 2, center=c(1,-0.5))
```



Exercice.
On considère le tableau `irishdata$tab`.

- Réaliser l'ACP spatialisée du tableau avec la pondération `pond.ext.w`.
- Représenter le premier axe de l'ACP spatialisée sur la carte.
- Comparer les deux cartes.

4 Annexes

Annexe 1

Position	Comté
1	Carlow
2	Cavan
3	Clare
4	Cork
5	Donegal
6	Galway
7	Kerry
8	Kildare
9	Kilkenny
10	Laoghis
11	Leitrim
12	Limerick
13	Longford
14	Louth
15	Mayo
16	Meath
17	Monaghan
18	Offaly
19	Roscommon
20	Sligo
21	Tipperary
22	Waterford
23	Westmeath
24	Wexford
25	Wicklow

Annexe 2 : Options de pondérations

W, pondération par standardisation ligne

```
pond.w <- nb2listw(ir.nb, style="W")
head(pond.w$weight)
```

```
[[1]]
[1] 0.2 0.2 0.2 0.2 0.2
[[2]]
[1] 0.2 0.2 0.2 0.2 0.2

[[3]]
[1] 0.3333333 0.3333333 0.3333333

[[4]]
[1] 0.25 0.25 0.25 0.25

[[5]]
[1] 1

[[6]]
[1] 0.2 0.2 0.2 0.2 0.2
```

```
unlist(lapply(pond.w$weight, 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
```

Le comté de Carlow [1] a 5 voisins de même poids $\frac{1}{5} = 0.2$.
Le comté de Clare [3] a 3 voisins de même poids $\frac{1}{3} = 0.3333$.
Les sommes en lignes (i.e. par comté) valent 1.

B, pondération par codage binaire simple

```
pond.b <- nb2listw(ir.nb, style="B")
head(pond.b$weight)
```

```
[[1]]
[1] 1 1 1 1 1
[[2]]
[1] 1 1 1 1 1

[[3]]
[1] 1 1 1

[[4]]
[1] 1 1 1 1

[[5]]
[1] 1

[[6]]
[1] 1 1 1 1 1
```

```
sum(unlist(pond.b$weight))
```

```
[1] 108
```

Le comté de Carlow [1] a 5 voisins.
Le comté de Clare [3] a 3 voisins.
La somme totale de ces pondérations est le nombre de liens non nuls soit 108.

C, pondération par standardisation globale

```
pond.c <- nb2listw(ir.nb, style="C")
head(pond.c$weight)
```

```
[[1]]
[1] 0.2314815 0.2314815 0.2314815 0.2314815 0.2314815
[[2]]
[1] 0.2314815 0.2314815 0.2314815 0.2314815 0.2314815

[[3]]
[1] 0.2314815 0.2314815 0.2314815

[[4]]
[1] 0.2314815 0.2314815 0.2314815 0.2314815

[[5]]
[1] 0.2314815

[[6]]
[1] 0.2314815 0.2314815 0.2314815 0.2314815 0.2314815
```

```
sum(unlist(pond.c$weight))
```

```
[1] 25
```

Il y a 108 liens pour 25 comtés donc une pondération vaut $25/108 = 0.2315$.
Le comté de Carlow [1] a 5 voisins de même pondération 0.2315.
Le comté de Clare [3] a 3 voisins de même pondération 0.2315.
La somme totale de ces pondérations est le nombre de comtés soit 25.

U, pondération par standardisation totale

```
pond.u <- nb2listw(ir.nb, style="U")
head(pond.u$weight)
```

```
[[1]]
[1] 0.009259259 0.009259259 0.009259259 0.009259259 0.009259259
[[2]]
[1] 0.009259259 0.009259259 0.009259259 0.009259259 0.009259259

[[3]]
[1] 0.009259259 0.009259259 0.009259259

[[4]]
[1] 0.009259259 0.009259259 0.009259259 0.009259259
```

```
[[5]]
[1] 0.009259259

[[6]]
[1] 0.009259259 0.009259259 0.009259259 0.009259259 0.009259259
 sum(unlist(pond.u$weight))
[1] 1
```

Il y a 108 liens donc une pondération vaut $1/108 = 0.009259$.
Le comté de Carlow [1] a 5 voisins de même pondération 0.009259.
Le comté de Clare [3] a 3 voisins de même pondération 0.009259.
La somme totale de ces pondérations vaut 1.

minmax, pondération proposée par Kelejian et Puncha (2010)

```
 pond.minmax <- nb2listw(ir.nb, style="minmax")
 head(pond.minmax$weight)
[[1]]
[1] 0.125 0.125 0.125 0.125 0.125
[[2]]
[1] 0.125 0.125 0.125 0.125 0.125

[[3]]
[1] 0.125 0.125 0.125

[[4]]
[1] 0.125 0.125 0.125 0.125

[[5]]
[1] 0.125

[[6]]
[1] 0.125 0.125 0.125 0.125 0.125
 (totw <- unlist(lapply(pond.b$weight,sum)))
 [1] 5 5 3 4 1 5 2 5 5 5 5 4 4 2 3 6 3 7 7 3 8 4 5 4 3
 max(totw)
[1] 8
 1/max(totw)
[1] 0.125
```

Le comté de Carlow [1] a 5 voisins de même pondération 0.125.
Le comté de Clare [3] a 3 voisins de même pondération 0.125.

S, pondération proposée par Tiefelsdorf (1999)

```
 pond.s <- nb2listw(ir.nb, style="S")
 head(pond.s$weight)
[[1]]
[1] 0.2194519 0.2194519 0.2194519 0.2194519 0.2194519
[[2]]
[1] 0.2194519 0.2194519 0.2194519 0.2194519 0.2194519

[[3]]
[1] 0.2833112 0.2833112 0.2833112

[[4]]
[1] 0.2453547 0.2453547 0.2453547 0.2453547

[[5]]
[1] 0.4907095

[[6]]
[1] 0.2194519 0.2194519 0.2194519 0.2194519 0.2194519
 sum(unlist(lapply(pond.s$weight,sum)))
[1] 25
```

On calcule $\sum_{i=1}^{25} \sqrt{d_i}$ où d_i est le nombre de voisins du comté i, information stockée dans l'objet `totw`. La pondération associée au comté i est définie par :

$$25 \times \frac{1}{\sum_{i=1}^{25} \sqrt{d_i}} \times \frac{1}{\sqrt{d_i}}$$

On obtient :

```
totw
 [1] 5 5 3 4 1 5 2 5 5 5 5 4 4 2 3 6 3 7 7 3 8 4 5 4 3
sum(sqrt(totw))
[1] 50.94664
25/(sum(sqrt(totw))*sqrt(5)) # Comté de Carlow
[1] 0.2194519
25/(sum(sqrt(totw))*sqrt(3)) # Comté de Clare
[1] 0.2833112
```

Le comté de Carlow [1] a 5 voisins de même pondération $25 \times \frac{1}{50.95} \times \frac{1}{\sqrt{5}} = 0.2195$
Le comté de Clare [3] a 3 voisins de même pondération $25 \times \frac{1}{50.95} \times \frac{1}{\sqrt{3}} = 0.2833$
La somme totale de ces pondérations vaut 25.

Annexe 3

L'indice local de Moran est :

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

Son espérance mathématique vaut :

$$E(I_i) = \frac{-w_i}{(n-1)}$$

Sa variance mathématique vaut :

$$V(I_i) = \frac{\sum_{i=1}^{n} \sum_{j=1;j\neq i}^{n} w_{ij}^2 \,(n-b_2)}{(n-1)} + \frac{2\sum_{i=1;k\neq i}^{n} \sum_{h=1;h\neq i}^{n} w_{ik} w_{ih}\,(2b_2-n)}{(n-1)(n-2)} - \frac{w_i^2}{(n-1)^2}$$

avec $b_2 = \frac{m_4}{m_2^2}$ et $m_4 = \frac{1}{n}\sum_{i=1}^{n} z_i^4$

m_2 et m_4 sont les moments centrés d'ordre 2 et 4.

Références

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.

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

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

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

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.