Méthodes de régression spatiale : un grand bol d'R

Philippe Apparicio Jérémy Gelb Jean Dubé Joan Carles Martori

2025-03-18

Table des matières

Liste des figures

Listes des tableaux

Préface

Résumé : Ce livre vise à décrire une panoplie de méthodes de régression spatiale avec le logiciel ouvert R. La philosophie de ce livre est de donner toutes les clés de compréhension et de mise en œuvre des méthodes abordées dans R. La présentation des méthodes est basée sur une approche compréhensive et intuitive plutôt que mathématique, sans pour autant négliger la rigueur statistique.

Remerciements : Ce manuel a été réalisé avec le soutien de la fabriqueREL. Fondée en 2019, la fabriqueREL est portée par divers établissements d'enseignement supérieur du Québec et agit en collaboration avec les services de soutien pédagogique et les bibliothèques. Son but est de faire des ressources éducatives libres (REL) le matériel privilégié en enseignement supérieur au Québec.

Maquette de la page couverture et identité graphique du livre : Andrés Henao Florez.

Mise en page : Philippe Apparicio et Marie-Hélène Gadbois Del Carpio.

Révision linguistique : Denise Latreille.

Pour citer cet ouvrage : Apparicio Philippe, Jérémy Gelb, Jean Dubé et Joan Carles Martori (2025). *Méthodes de régression spatiale : un grand bol d'R.* Université Laval et Université de Sherbrooke. fabriqueREL. Licence CC BY-SA.

Un manuel sous la forme d'une ressource éducative libre

Pourquoi un manuel sous licence libre?

Les logiciels libres sont aujourd'hui très répandus. Comparativement aux logiciels propriétaires, l'accès au code source permet à quiconque de l'utiliser, de le modifier, de le dupliquer et de le partager. Le logiciel R, dans lequel sont mises en œuvre les méthodes de régression spatiale décrites dans ce livre, est d'ailleurs à la fois un langage de programmation et un logiciel libre (sous la licence publique générale GNU GPL2). Par analogie aux logiciels libres, il existe aussi des **ressources éducatives libres (REL)** « dont la licence accorde les permissions désignées par les 5R (**Retenir — Réutiliser — Réviser — Remixer — Redistribuer**) et donc permet nécessairement la modification » (***fabriqueREL***). La licence de ce livre, CC BY-SA (figure 1), permet donc de :

- **Retenir**, c'est-à-dire télécharger et imprimer gratuitement le livre. Notez qu'il aurait été plutôt surprenant d'écrire un livre payant sur un logiciel libre et donc gratuit. Aussi, nous aurions été très embarrassés que des personnes étudiantes avec des ressources financières limitées doivent payer pour avoir accès au livre, sans pour autant savoir préalablement si le contenu est réellement adapté à leurs besoins.
- **Réutiliser**, c'est-à-dire utiliser la totalité ou une section du livre sans limitation et sans compensation financière. Cela permet ainsi à d'autres personnes enseignantes de l'utiliser dans le cadre d'activités pédagogiques.
- **Réviser**, c'est-à-dire modifier, adapter et traduire le contenu en fonction d'un besoin pédagogique précis puisqu'aucun manuel n'est parfait, tant s'en faut! Le livre a d'ailleurs été écrit intégralement dans R avec Quarto. Quiconque peut ainsi télécharger gratuitement le code source du livre sur github et le modifier à sa guise (voir l'encadré intitulé *Suggestions d'adaptation du manuel*).
- **Remixer**, c'est-à-dire « combiner la ressource avec d'autres ressources dont la licence le permet aussi pour créer une nouvelle ressource intégrée » (***fabriqueREL***).
- **Redistribuer**, c'est-à-dire distribuer, en totalité ou en partie le manuel ou une version révisée sur d'autres canaux que le site Web du livre (par exemple, sur le site Moodle de votre université ou en faire une version imprimée).

La licence de ce livre, CC BY-SA (figure 1), oblige donc à :

- Attribuer la paternité de l'auteur dans vos versions dérivées, ainsi qu'une mention concernant les grandes modifications apportées, en utilisant la formulation suivante : Apparicio Philippe, Jérémy Gelb, Jean Dubé et Joan Carles Martori (2025). *Méthodes de régression spatiale : un grand bol d'R*. Université Laval et Université de Sherbrooke. fabriqueREL. Licence CC BY-SA.
- Utiliser la même licence ou une licence similaire à toutes versions dérivées.

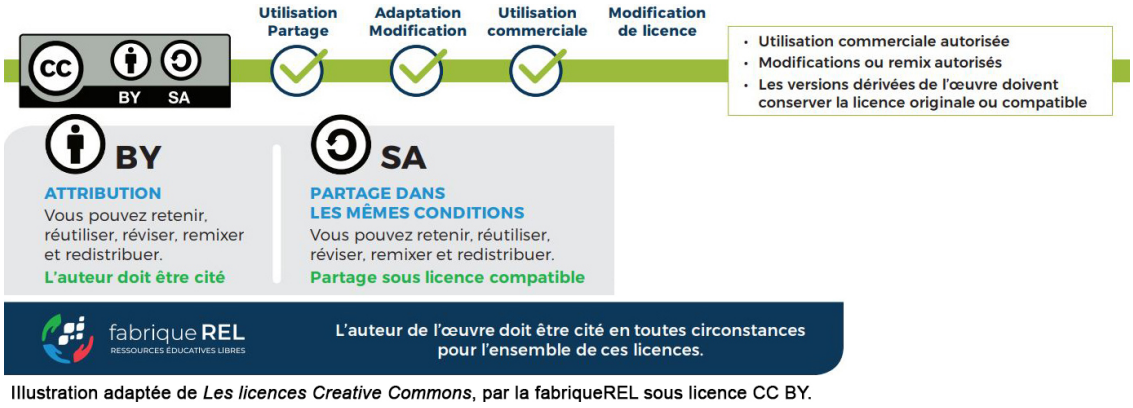


FIGURE 1 – Licence Creative Commons du livre

Astuce

Suggestions d'adaptation du manuel
Pour chaque méthode d'analyse spatiale abordée dans le livre, une description détaillée et une mise en œuvre dans R sont disponibles. Par conséquent, plusieurs adaptations du manuel sont possibles :

- Conserver uniquement les chapitres sur les méthodes ciblées dans votre cours.
- En faire une version imprimée et la distribuer aux personnes étudiantes.
- Modifier la description d'une ou de plusieurs méthodes en effectuant les mises à jour directement dans les chapitres.
- Insérer ses propres jeux de données dans les sections intitulées *Mise en œuvre dans R*.
- Modifier les tableaux et figures.
- Ajouter une série d'exercices.
- Modifier les quiz de révision.
- Rédiger un nouveau chapitre.
- Modifier des syntaxes R. Plusieurs *packages* R peuvent être utilisés pour mettre en œuvre telle ou telle méthode. Ces derniers évoluent aussi très vite et de nouveaux *packages* sont proposés fréquemment! Par conséquent, il peut être judicieux de modifier une syntaxe R du livre en fonction de ses habitudes de programmation dans R (utilisation d'autres *packages* que ceux utilisés dans le manuel par exemple) ou de bien mettre à jour une syntaxe à la suite de la parution d'un nouveau *package* plus performant ou intéressant.
- Toute autre adaptation qui permet de répondre au mieux à un besoin pédagogique.

Comment lire ce manuel?

Le livre comprend plusieurs types de blocs de texte qui en facilitent la lecture.

Package

Bloc *packages*
Habituellement localisé au début d'un chapitre, il comprend la liste des *packages* R utilisés pour un chapitre.

Objectif

Bloc objectifs
Il comprend une description des objectifs d'un chapitre ou d'une section.

Note

Bloc notes
Il comprend une information secondaire sur une notion, une idée abordée dans une section.

Aller plus loin

Bloc pour aller plus loin
Il comprend des références ou des extensions d'une méthode abordée dans une section.

Astuce

Bloc astuce
Il décrit un élément qui vous facilitera la vie : une propriété statistique, un *package*, une fonction, une syntaxe R.

Attention

Bloc attention
Il comprend une notion ou un élément important à bien maîtriser.

Exercice

Bloc exercice
Il comprend un court exercice de révision à la fin de chaque chapitre.

Comment utiliser les données du livre pour reproduire les exemples?

Ce livre comprend des exemples détaillés et appliqués dans R pour chacune des méthodes abordées. Ces exemples se basent sur des jeux de données structurés et mis à disposition avec le livre. Ils sont disponibles sur le *repo github* dans le sous-dossier `data`, à l'adresse https://github.com/SerieBoldR/RegressionsSpatiales/tree/main/data.

Une autre option est de télécharger le *repo* complet du livre directement sur *github* (https://github.com/SerieBoldR/RegressionsSpatiales) en cliquant sur le bouton `Code`, puis le bouton `Download ZIP` (figure 2). Les données se trouvent alors dans le sous-dossier nommé `data`.

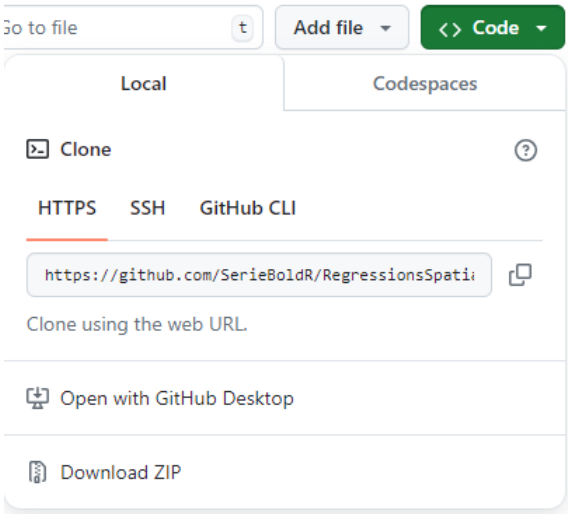


FIGURE 2 – Téléchargement de l'intégralité du livre

Liste des packages utilisés

Dans ce livre, nous utilisons la version 4.4.0 de R avec de nombreux *packages* que vous pouvez installer avec le code ci-dessous.

```
# Liste des packages
ListePackages <- c("adespatial", "car", "corrplot", "DHARMa", "dplyr", "future",
                   "future.apply",  "gamlss", "GGally", "ggplot2", "ggpubr", "geoR",
                   "GWmodel", "kableExtra", "mgcv", "mgcViz", "mgwrsar", "plm",
                   "ProbitSpatial", "randomcoloR", "RColorBrewer", "sf", "spatialreg",
                   "spdep", "spgwr", "splm", "spmoran", "tmap")
# Packages non installés dans la liste
PackagesNonInstalles <- ListePackages[!(ListePackages %in% installed.packages()[,"Package"])]
# Installation des packages manquants
if(length(new.packages)) install.packages(PackagesNonInstalles)
```

Bref historique sur les modèles de régressions spatiales en économie et en géographie

Depuis une cinquantaine d'années, les économistes et les géographes conçoivent et utilisent largement des méthodes de régression prenant en compte la dimension spatiale : modèles d'économétrie spatiale, régressions géographiquement pondérées, modèles généralisés additifs intégrant une dimension spatiale, modèles avec des vecteurs propres spatiaux, etc.

La notion d'autocorrélation spatiale, étroitement liée aux régressions spatiales, remonte à une période encore plus ancienne. Elle a été introduite notamment par les travaux de Patrick Moran, d'abord en 1948 puis en 1950, suivis par ceux de Robert Geary en 1954 et de Nathan Mantel en 1967 (Moran 1948; Moran 1950; Geary 1954; Mantel 1967). La formalisation des relations spatiales trouve son origine, plus précisément, dans la première loi de la géographie énoncée par Waldo Tobler qui stipule que « tout interagit avec tout, mais les objets proches ont plus de chance de le faire que les objets éloignés [*Everything is related to everything else, but near things are more related than distant things*] » (Tobler 1970).

En économie, le terme « économétrie spatiale » a été utilisé pour la première fois par Jean Paelinck (1978) à la fin des années 1970. Cependant, il est resté relativement peu exploité jusqu'aux travaux de Luc Anselin, qui a publié en 1988 un ouvrage de référence sur le sujet intitulé *Spatial econometrics: methods and models* (Anselin 1988). Il faut néanmoins attendre le développement des outils informatiques avant de voir les applications affluer. Ainsi, au début des années 1990, l'économétrie spatiale a véritablement pris son essor (voir Anselin (2010) pour un historique détaillé). Depuis, ce domaine a gagné en importance, s'établissant presque comme un incontournable pour la prise en compte des effets spatiaux, mais aussi, et surtout, des fameux effets de débordement. Ces dernières décennies, plusieurs ouvrages traitant des modèles économétriques spatiaux ont été publiés, principalement en anglais (Anselin et Florax 1995; Anselin, Florax et Rey 2013; LeSage et Pace 2009; Bivand et al. 2008; Anselin et Rey 2014). Ils méritent grandement d'être consultés, particulièrement celui de LeSage et Pace (2009). Peu de références sont disponibles en français. Le livre de référence de Dubé et Legros (2014) constitue vraisemblablement une référence intéressante, notamment par la présentation vulgarisée des modèles et des concepts.

En géographie, Daniel Griffith est certainement l'un des géographes les plus actifs dans le domaine des régressions spatiales, ayant fait plusieurs propositions pour contrôler la présence d'autocorrélation spatiale résiduelle. Ses premiers travaux remontent aux années 1980, et il a publié depuis un grand nombre d'ouvrages (notamment, Griffith 1988; Griffith 2003; Griffith, Chun et Li 2019). Puis, dans les années 1990, trois géographes – Stewart Fotheringham, Chris Brunsdon et Martin Charlton – ont introduit la régression géographiquement pondérée (*Geographically weighted regression* - GWR), une méthode permettant d'analyser localement les relations entre la variable dépendante et les variables indépendantes d'un modèle de régression (Fotheringham, Charlton et Brunsdon 1996; Brunsdon, Fotheringham et Charlton 1996; Brunsdon, Fotheringham et Charlton 1998). En 2003, la parution de leur ouvrage intitulé *Geographically weighted regression: the analysis of spatially varying relationships* (Fotheringham, Brunsdon et Charlton 2003) a largement contribué à la popularisation de cette méthode. Depuis, de nombreuses extensions de la GWR ont été proposées.

Structure du livre

Le manuel est structuré en six parties.

Partie 1. Notions de base

Dans cette première partie, nous présentons les jeux de données utilisés pour mettre en œuvre les différentes méthodes de régression spatiale présentées dans le livre (chapitre 1). Nous discutons aussi de plusieurs notions fondamentales et méthodes qu'il importe de bien maîtriser avant d'aborder les chapitres suivants consacrés aux méthodes de régression spatiale, notamment l'autocorrélation spatiale, la notion de variable spatialement décalée et la régression linéaire multiple. Nous conclurons ce chapitre en exposant les raisons qui justifient l'utilisation de différentes formes de régressions spatiales pour modéliser des données spatiales ou spatiotemporelles.

Partie 2. Spécification de la structure de covariance spatiale

Dans cette seconde partie, les types de régression spatiale retenus introduisent l'espace en spécifiant directement la structure de covariance induite par l'autocorrélation spatiale dans les matrices de covariance de distribution normale (chapitre 2). La méthode des moindres carrés généralisés (GLS) permet d'étendre le modèle des moindres carrés ordinaires (MCO) pour tenir compte de cette structure de covariance entre les observations. Cette approche est similaire à l'introduction d'effets aléatoires, ce qui a notamment conduit à la construction de GLMM (modèles linéaires généralisés à effets mixtes) introduisant des effets aléatoires distribués normalement avec une matrice de covariance structurée spatialement. Finalement, nous décrirons le modèle d'autorégression conditionnelle (*conditional autoregressive model*, CAR) qui s'applique à une variable dépendante continue dont la valeur pour une entité spatiale dépend de celles des entités spatiales proches ou voisines.

Partie 3. Modèles d'économétrie spatiale

Cette troisième partie comprend trois chapitres qui sont consacrés aux modèles d'économétrie spatiale qui vise à modéliser la **dépendance spatiale**. D'emblée, nous décrivons les principaux modèles spatiaux autorégressifs pour une variable dépendante continue qui permettent d'introduire l'autocorrélation spatiale sur les variables indépendantes (modèle SLX), la variable dépendante (SAR), le terme d'erreur (SEM), à la fois la variable dépendante et les variables indépendantes (SDM) et à la fois les variables indépendantes et le terme d'erreur (SDEM) (chapitre 3). Puis, nous abordons les modèles probit spatiaux pour modéliser une variable qualitative dichotomique (binaire) (chapitre 4). Finalement, nous décrivons d'autres extensions des modèles autorégressifs, soit les modèles spatiaux en panel qui permettent de modéliser des données spatiales longitudinales (chapitre 5).

Partie 4. Variable latente spatiale : lissage et filtrage spatial

Dans cette troisième partie, les modèles retenus ont la particularité d'ajouter un terme spatialement structuré dans leur équation de régression. Ce terme spatial est construit à partir d'un ensemble de fonctions de base multipliées par des coefficients. Ces fonctions de base suivent des patrons géographiques, ce qui leur permet de capturer une variable latente spatialement autocorrélée qui autrement aurait fini dans les résidus. Le premier chapitre de cette partie est consacré aux modèles généralisés additifs (GAM) qui permettent d'introduire l'espace de deux manières différentes : avec une *spline* bivariée construite à partir des coordonnées géographiques (x, y) pour capturer les variations continues dans l'espace; avec un lissage par champ aléatoire de Markov (*Markov Random Field* – MRF) pour modéliser la dépendance spatiale entre les unités spatiales voisines (chapitre 6). Dans le second, nous abordons les modèles linéaires généralisés avec des vecteurs spatiaux (*Spatial Eigenvector Generalized Linear Models*, SEVM) (chapitre 7). Ces modèles SEVM ajoutent des variables construites à partir de la décomposition de la matrice de pondération spatiale en vecteurs propres (*Moran Eigenvectors*). Ces deux types de modèles (GAM et SEVM) se ressemblent énormément dans leur conceptualisation de l'espace.

Partie 5. Régressions spatiales et hétérogénéité spatiale

Dans cette cinquième partie, nous abordons plusieurs types de régression spatiale qui permettent de faire varier les coefficients de régression dans l'espace. Premièrement, les régressions géographiquement pondérées (*Geographically weighted regression* - GWR) permettent d'explorer et de visualiser l'**hétérogénéité spatiale**, soit l'instabilité des relations entre la variable dépendante et les variables indépendantes. Premièrement, nous décrirons les formes dites classiques de la GWR qui s'appliquent à des variables dépendantes continues, dichotomiques (logistique) et de comptage (Poisson) (chapitre 8). Puis, nous abordons des extensions de la GWR, particulièrement la GWR mixte, la GWR multiéchelle et les GWR mixtes avec des variables spatialement décalées (MGWR-SAR) (chapitre 9). Finalement, nous verrons comment il est possible d'introduire des coefficients variant spatialement avec des modèles GAM (modèles généralisés additifs) et GLMM (modèles linéaires généralisés à effets mixtes) (chapitre 10).

Partie 6. Conclusions

Cette dernière partie regroupe les exercices corrigés (chapitre 11), une conclusion générale (chapitre 12) et la bibliographie.

Remerciements

De nombreuses personnes ont contribué à l'élaboration de ce manuel.

Ce projet a bénéficié du soutien pédagogique et financier de la ***fabriqueREL*** (ressources éducatives libres). Les différentes rencontres avec le comité de suivi nous ont permis de comprendre l'univers des ressources éducatives libres (REL) et notamment leurs fameux 5R (Retenir — Réutiliser — Réviser — Remixer — Redistribuer), de mieux définir le besoin pédagogique visé par ce manuel, d'identifier des ressources pédagogiques et des outils pertinents pour son élaboration. Ainsi, nous remercions chaleureusement les membres de la *fabriqueREL* pour leur soutien inconditionnel :

— Marianne Demers-Desmarais, bibliothécaire, Université Laval.
— Julie-Christine Gagné, conseillère en pédagogie universitaire, Université Laval.
— Claude Potvin, conseiller pédagogique à la fabriqueREL, Université Laval.
— Nadia Villeuneuve, bibliothécaire, Université Laval.

Nous remercions aussi les membres du comité de révision pour leurs commentaires et suggestions très constructifs. Ce comité est composé de plusieurs personnes étudiantes du Département de géomatique appliquée de l'Université de Sherbrooke et de l'École supérieure d'aménagement du territoire et de développement régional (ÉSAD) de l'Université de Laval.

— Diego Andres Cardenas Morales, étudiant au doctorat en aménagement du territoire et développement régional.
— Marie-Clara Delage, Victor Bibeau, Liam Messier et Jean-François Darveau, personnes étudiantes de troisième année au baccalauréat en géomatique appliquée à l'environnement.

Marie-Hélène Gadbois Del Carpio, étudiante à la maîtrise en géomatique appliquée et télédétection (type recherche). Elle a participé activement à la mise en page du manuel; elle est désormais une grande spécialiste des feuilles de style en cascade (CSS), de Quarto et LaTeX.

Finalement, nous remercions Denise Latreille, réviseuse linguistique et chargée de cours à l'Université Sherbrooke, pour la révision du manuel.

À propos des auteurs

Philippe Apparicio est professeur titulaire au Département de géomatique appliquée de l'Université de Sherbrooke. Il y enseigne aux programmes de 1er et 2e cycles de géomatique les cours *Transport et mobilité durable*, *Modélisation et analyse spatiale* et *Géomatique appliquée à la gestion urbaine*. Géographe de formation, ses intérêts de recherche incluent la justice et l'équité environnementale, la mobilité durable, les pollutions atmosphérique et sonore, et le vélo en ville. Il a publié une centaine d'articles scientifiques dans différents domaines des études urbaines et de la géographie mobilisant la géomatique et l'analyse spatiale.

Jean Dubé est professeur titulaire à l'École Supérieure d'aménagement du territoire et de développement régional (ÉSAD) de l'Université Laval. Économiste de formation, ses intérêts de recherche portent notamment sur l'évaluation d'impact des politiques publiques liées à l'aménagement et au développement. Ses recherches portent sur la mesure des externalités urbaines par le marché immobilier, les décisions de localisation des entreprises et des ménages et les déterminants de la croissance urbaine et régionale. Il est aussi co-éditeur de la Revue canadienne des sciences régionales (RCSR).

Jérémy Gelb a obtenu un doctorat en études urbaines à l'INRS en 2022 (*L'exposition des cyclistes aux pollutions atmosphérique et sonore en milieu urbain : comparaison empirique de plusieurs villes à travers le monde*), sous la supervision de Philippe Apparicio. Il est tombé dans la marmite de l'*open source* avec le triptyque QGIS, R et Python au début de sa maîtrise. Il a développé deux *packages* : `geocmeans` et `spNetwork`, permettant respectivement d'effectuer des analyses de classification floue non supervisée pondérée spatialement et des estimations de densité par noyau sur réseau. Il travaille actuellement comme conseiller en science des données à l'Autorité régionale de transport métropolitain qui gère la planification du transport collectif dans la région de Montréal. Ces travaux portent sur la qualité des milieux urbains, l'accessibilité spatiale, le transport, l'équité environnementale, les SIG et l'analyse spatiale.

Joan Carles Martori est professeur au département d'économie et de gestion de l'Université de Vic - Université centrale de la Catalogne (Barcelone, Espagne). Il enseigne les statistiques appliquées et l'économétrie dans des cours de premier et de deuxième cycle. Il est le chercheur principal du groupe Data Analysis and Modelling. Économiste de formation, ses recherches portent sur la ségrégation résidentielle, les inégalités et l'équité environnementale à l'aide des statistiques spatiales et de techniques économétriques. Il a contribué au développement du *package* R `AQuadtree` sur la confidentialité des données spatiales ponctuelles. En tant que statisticien appliqué, il collabore avec d'autres disciplines telles que l'éducation et la santé.

Partie 1. Notions de base

1 Autocorrélation spatiale, dépendance spatiale et hétérogénéité spatiale d'un modèle de régression

Avant d'explorer les différents types de régressions spatiales dans les chapitres suivants, il est primordial de comprendre plusieurs notions clés, comme l'autocorrélation spatiale, la dépendance spatiale et l'hétérogénéité spatiale. Aussi, de nombreuses méthodes de régression spatiale s'appuient sur l'utilisation de matrices de pondération spatiale. Nous proposons également un bref rappel sur la régression linéaire multiple.

Certaines sections de ce chapitre sont adaptées du manuel suivant : Apparicio P. et J. Gelb (2024). *Méthodes d'analyse spatiales : un grand bol d'R*. Université de Sherbrooke, Département de géomatique appliquée. fabriqueREL. Licence CC BY-SA.

Objectif

Objectifs d'apprentissage visés dans ce chapitre

À la fin de ce chapitre, vous devriez être en mesure de :

- connaître les principales matrices de pondération spatiale;
- maîtriser les notions d'autocorrélation spatiale, de dépendance spatiale et d'hétérogénéité spatiale;
- comprendre la notion de variable spatialement décalée;
- calculer une mesure d'autocorrélation spatiale globale (I de Moran) dans R;
- évaluer la dépendance spatiale d'un modèle de régression multiple en calculant le I de Moran sur ses résidus.

Package

Liste des *packages* utilisés dans ce chapitre

- Pour importer et manipuler des fichiers géographiques :
  - `sf` pour importer et manipuler des données vectorielles.
- Pour cartographier des données :
  - `ggplot2` et `ggpubr` pour construire des graphiques.
  - `tmap` pour construire des cartes thématiques.
  - `RColorBrewer` pour construire une palette de couleur.
- Pour les mesures d'autocorrélation spatiale :
  - `spdep` pour construire des matrices de pondération spatiales et calculer le I de Moran.

1.1 Description des jeux de données utilisés dans le manuel

Plusieurs jeux de données sont utilisés dans ce livre et sont déjà structurés et disponibles au format `.Rdata`.

⚠ **Attention**

Manipulation, structuration et cartographie de données spatiales
Ce livre n'a pas pour objectif de traiter la phase préparatoire consistant à structurer les données spatiales dans R avant l'application de modèles de régression spatiale. Nous partons du principe que les lectrices et lecteurs maîtrisent déjà l'utilisation des *packages* `sf` et `tmap` pour manipuler, structurer et cartographier des données spatiales. Si ce n'est pas le cas, nous vous encourageons à consulter le chapitre intitulé *Manipulation des données spatiales dans R* (Apparicio et Gelb 2024).

1.1.1 Jeu de données sur l'agglomération de Lyon

Premièrement, nous utiliserons le jeu de données spatiales `LyonIris` du *package* `geocmeans`. Ce jeu de données spatiales pour l'agglomération lyonnaise (France) comprend dix variables, dont quatre environnementales (EN) et six socioéconomiques (SE), pour les îlots regroupés pour l'information statistique (IRIS) de l'agglomération lyonnaise (tableau 1.1 et figure 1.1).

Tableau 1.1 – Statistiques descriptives du jeu de données LyonIris

Nom	Intitulé	Type	Moy.	E.-T.	Min.	Max.
Lden	Bruit routier (Lden dB(A))	EN	55,6	4,9	33,9	71,7
NO2	Dioxyde d'azote (ug/m^3)	EN	28,7	7,9	12,0	60,2
PM25	Particules fines ($PM_{2,5}$)	EN	16,8	2,1	11,3	21,9
VegHautPrt	Canopée (%)	EN	18,7	10,1	1,7	53,8
Pct0_14	Moins de 15 ans (%)	SE	18,5	5,7	0,0	54,0
Pct_65	65 ans et plus (%)	SE	16,2	5,9	0,0	45,1
Pct_Img	Immigrants (%)	SE	14,5	9,1	0,0	59,8
TxChom1564	Taux de chômage	SE	14,8	8,1	0,0	98,8
Pct_brevet	Personnes à faible scolarité (%)	SE	23,5	12,6	0,0	100,0
NivVieMed	Médiane du niveau de vie (milliers d'euros)	SE	21,8	4,9	11,3	38,7

1.1.2 Jeu de données sur la région métropolitaine de Montréal

Deuxièmement, nous utiliserons un jeu de données spatiales sur les parts modales dans la région de Montréal extraites du recensement de 2021 de Statistique Canada pour la région métropolitaine de Montréal (tableau 1.2). En guise d'exemple, les proportions des navetteurs utilisant respectivement le transport en commun et l'automobile par secteur de recensement sont présentées à la figure 1.2.

Tableau 1.2 – Statistiques descriptives du jeu de données sur Montréal

Nom	Intitulé	Moy.	E.-T.
prt_monoparental	Proportion de familles monoparentales	0,2	0,1
prt_minorite_vis	Proportion de minorités visibles	0,3	0,2
prt_chomage	Proportion de chômeurs	0,1	0,0
prt_personnes_faibles_revenu	Proportion de personnes à faible revenu	7,9	6,3

Tableau 1.2 – Statistiques descriptives du jeu de données sur Montréal

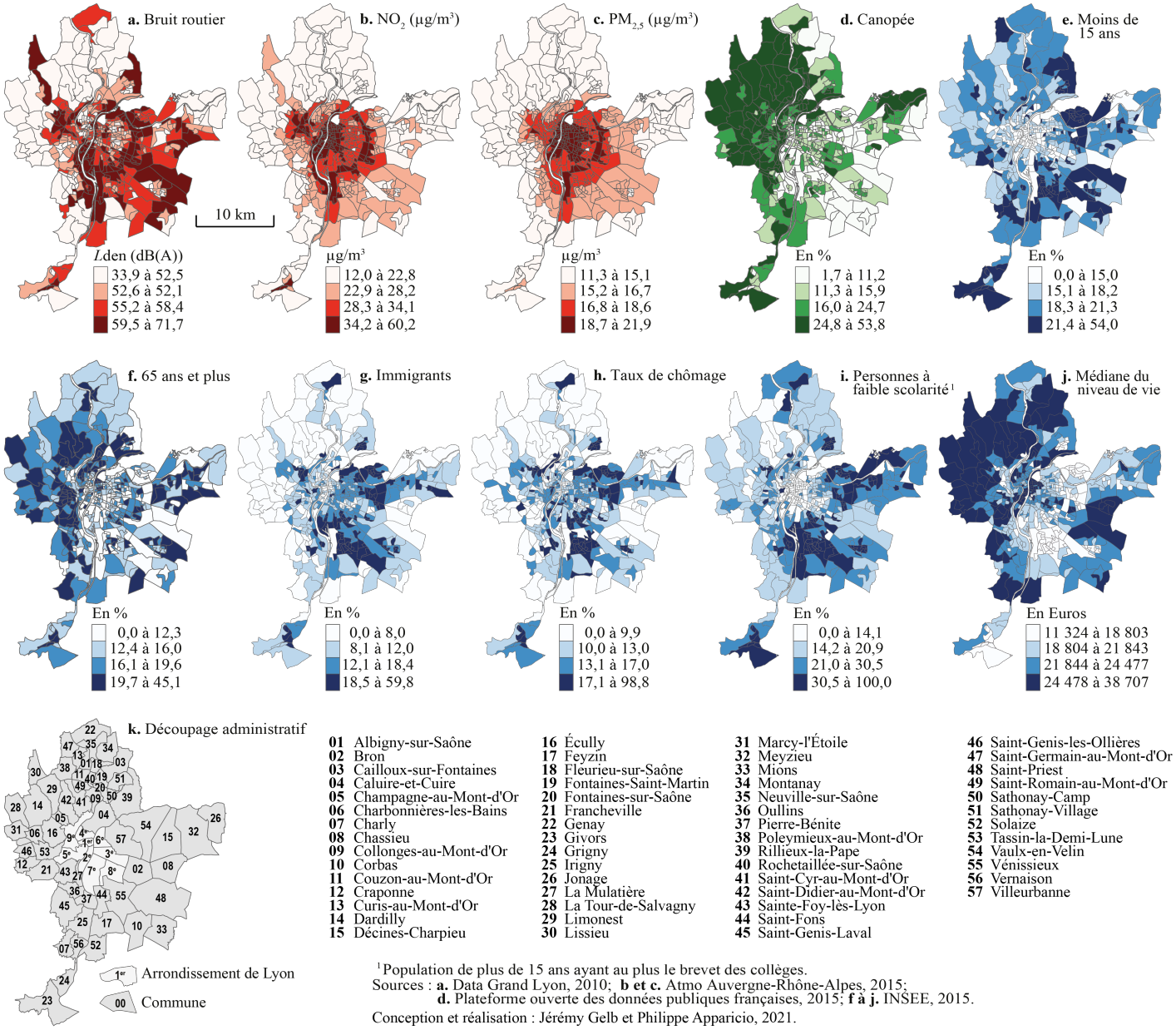
Nom	Intitulé	Moy.	E.-T.
revenu_median	Revenu médian des ménages (milliers de dollars)	40,2	9,5
densite_population_km2	Habitants au km2	5 758,2	5 429,8
mode_tc	Transports en commun (effectifs)	242,7	197,8
mode_auto	Automobile (effectifs)	1 200,3	811,9
mode_pieton	Marche (effectifs)	90,6	70,7
mode_velo	Vélo (effectifs)	28,0	34,2
total_commuters	Total navetteurs (effectifs)	1 588,4	803,5
acs_idx_emp_velo	Accessibilité aux emplois en heure de pointe à vélo	0,2	0,2
acs_idx_emp_tc_peak	Accessibilité aux emplois en heure de pointe en transport en commun	0,2	0,2
acs_idx_emp_pieton	Accessibilité aux emplois en heure de pointe à la marche	0,1	0,1
prt_tc	Proportion de navetteurs en transport en commun	0,2	0,1
prt_auto	Proportion de navetteurs en auto	0,7	0,2
prt_actif	Proportion de naveteurs en transport actif (vélo et marche)	0,1	0,1

1.1.3 Jeu de données sur Barcelone

Troisièmement, nous utiliserons un jeu de données spatiales (`Barcelone`) pour la ville de Barcelone (Catalogne, Espagne). Ce jeu de données spatiales comprend 14 variables, dont cinq environnementales (EN), six socioéconomiques (SE), une géographique (GE) et un variable binaire (BI), pour les 1068 entités polygonales de recensement formant la ville de Barcelone (tableau 1.3 et figure 1.1).

Tableau 1.3 – Statistiques descriptives du jeu de données sur Barcelone

Nom	Intitulé	Type	Moy.	E.-T.	Min.	Max.
NO2	Dioxyde d'azote (ug/m^3)	EN	37,0	6,2	16,3	61,4
PM25	Particules fines ($PM_{2,5}$)	EN	16,4	2,8	12,5	22,5
Arbres	Nombre d'arbres	EN	143,8	179,1	0,0	3 055,0
Surf	Surface bâtie (m^2)	EN	114 978,3	189 232,3	0,0	5 606 600,0
Parcs	Espaces verts (m^2)	EN	8 346,8	60 745,6	0,0	1 832 077,0
PopTot	Population	SE	1 506,3	341,0	583,0	3 446,0
Densite	Habitants au km^2	SE	42 160,8	22 311,7	78,2	153 770,3
Revenu	Revenu des ménages (Euros)	SE	37 462,1	12 683,9	19 056,0	89 015,0
Pct_Img	Immigrants non européens (%)	SE	10,9	6,6	1,1	51,0
Pct0_4	Moins de 4 ans (%)	SE	3,7	1,0	0,9	11,3
Pct_Plus80	80 ans et plus (%)	SE	7,7	2,4	1,2	23,8
Distance	Distance du centre (m)	GE	2 987,3	1 573,7	0,0	8 057,0
ClusterUE	Cluster of UE Immigrants	BI	0,6	0,5	0,0	1,0



[1] Population de plus de 15 ans ayant au plus le brevet des collèges.
Sources : **a.** Data Grand Lyon, 2010; **b et c.** Atmo Auvergne-Rhône-Alpes, 2015; **d.** Plateforme ouverte des données publiques françaises, 2015; **f à j.** INSEE, 2015.
Conception et réalisation : Jérémy Gelb et Philippe Apparicio, 2021.

FIGURE 1.1 – Cartographie des variables du jeu de données LyonIris

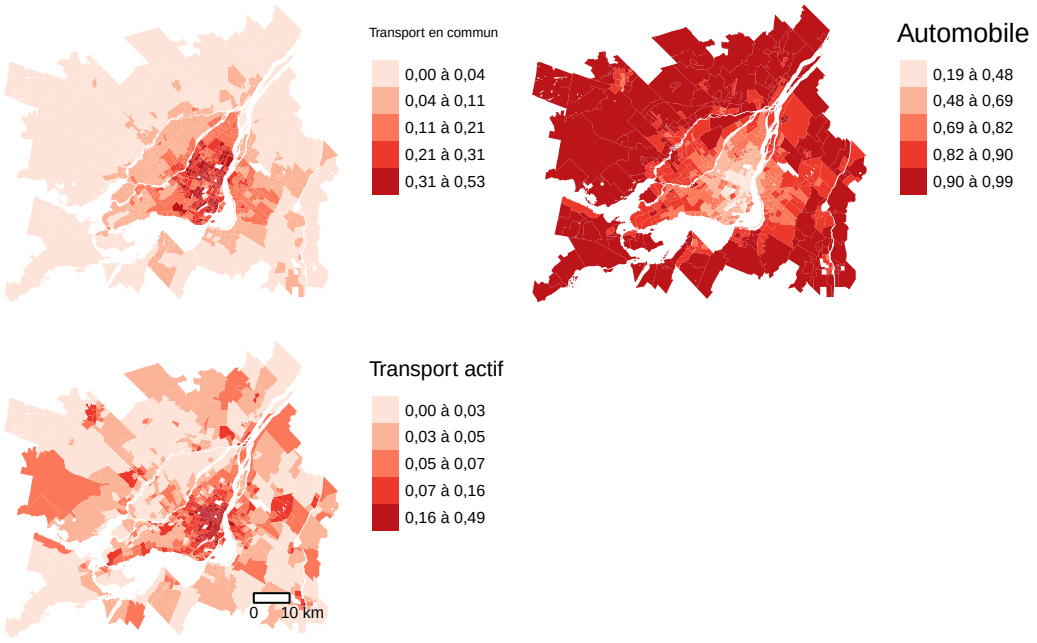


FIGURE 1.2 – Parts modales du transport en commun, de l'automobile et du transport actif (proportion)

1.2 Matrices de pondération spatiale

Objectif

Utilité des matrices de pondération spatiale
Ces matrices permettent de définir les relations spatiales entre les entités spatiales d'une couche géographique, et plus spécifiquement la manière de mesurer leur relation d'adjacence (voisinage) ou de proximité (distance). Nous verrons qu'elles sont largement utilisées dans les régressions spatiales, notamment dans les modèles d'économétrie spatiale abordés dans la seconde partie du livre. Elles sont aussi nécessaires au calcul des mesures d'autocorrélation globales ou locales (section 1.4.2) et à la construction des variables spatialement décalées (section 1.3).

Il existe huit principales matrices de pondération spatiale regroupées en deux grandes catégories : celles de contiguïté (basées sur l'adjacence) et celles de proximité (basées sur la distance) (tableau 1.4). Lorsque la couche géographique est composée de points, seules les matrices de proximité peuvent être utilisées.

TABLEAU 1.4 – Matrices de pondération spatiale selon la géométrie

Matrice	Points	Lignes	Polyg.	Raster
Matrices de contiguïté (basées sur l'adjacence)				
Partage d'un nœud (Queen)		X	X	X
Partage d'un segment (Rook)		X	X	X
Partage d'un nœud et ordre d'adjacence (Queen)		X	X	X
Partage d'un segment et ordre d'adjacence (Rook)		X	X	X
Matrices de proximité (basées sur la distance)				
Connectivité selon la distance	X	X	X	X

Inverse de la distance	X	X	X	X
Inverse de la distance au carré	X	X	X	X
Nombre de plus proches voisins	X	X	X	X

1.2.1 Matrices de contiguïté

La relation d'adjacence (de contiguïté) vise à déterminer si deux entités spatiales sont ou non voisines selon le partage soit d'un nœud, soit d'un segment (frontière commune). La contiguïté est liée à la notion de topologie qui prend en compte les relations de voisinage entre des entités spatiales, sans tenir compte de leurs tailles et de leurs formes géométriques. Elle peut être représentée à partir d'une **matrice de contiguïté** (avec une valeur de 1 quand deux entités sont voisines et de 0 pour une situation inverse) ou d'un **graphe** (formé de points représentant les entités spatiales et de lignes reliant les entités voisines) (figure 1.4).

Trois évaluations de la contiguïté sont représentées à la figure 1.5 :

- **Adjacence selon le partage d'un segment**, soit d'une frontière commune entre les polygones (A).
- **Adjacence selon le partage d'un nœud** (B).
- **Ordre d'adjacence selon le partage d'une frontière commune** (C). L'ordre d'adjacence indique le nombre de frontières à traverser pour se rendre à l'entité spatiale contiguë, soit :
  - **Ordre 1** : une frontière à traverser pour se rendre dans l'entité spatiale adjacente.
  - **Ordre 2** : deux frontières à traverser pour atteindre les entités de la deuxième couronne.
  - **Ordre 3** : trois frontières à traverser pour atteindre les entités de la troisième couronne.
  - Etc.

Bien entendu, les ordres d'adjacence peuvent être également définis selon le partage d'un nœud commun.

Habituellement appelée **W**, la matrice de contiguïté est binaire tant selon le partage d'un nœud (*Queen* en anglais) (équation 1.1) que d'un segment commun (*Rook* en anglais) (équation 1.2).

$$w_{ij} = \begin{cases} 1 & \text{si les entités spatiales } i \text{ et } j \text{ ont au moins un nœud commun; } i \neq j \\ 0 & \text{sinon} \end{cases} \tag{1.1}$$

$$w_{ij} = \begin{cases} 1 & \text{si les entités spatiales } i \text{ et } j \text{ partagent une frontière commune; } i \neq j \\ 0 & \text{sinon} \end{cases} \tag{1.2}$$

1.2.2 Matrices de proximité

Pour construire une matrice de pondération spatiale selon la proximité, nous pouvons utiliser plusieurs types de distances (Apparicio et al. 2017) : certaines sont cartésiennes, d'autres, dites réticulaires, sont calculées à partir d'un réseau de rues (figure 1.6).

Les distances cartésiennes – euclidienne et de Manhattan (équation 1.3 et équation 1.4) – sont facilement calculables à partir des coordonnées géographiques (x,y) lorsque la couche géographique est dans un système de projection plane (figure 1.6, a). Si la projection de la couche est sphérique (longitude/latitude), il convient d'utiliser la formule de haversine (basée sur la trigonométrie sphérique) pour obtenir la distance à vol d'oiseau (équation 1.5).

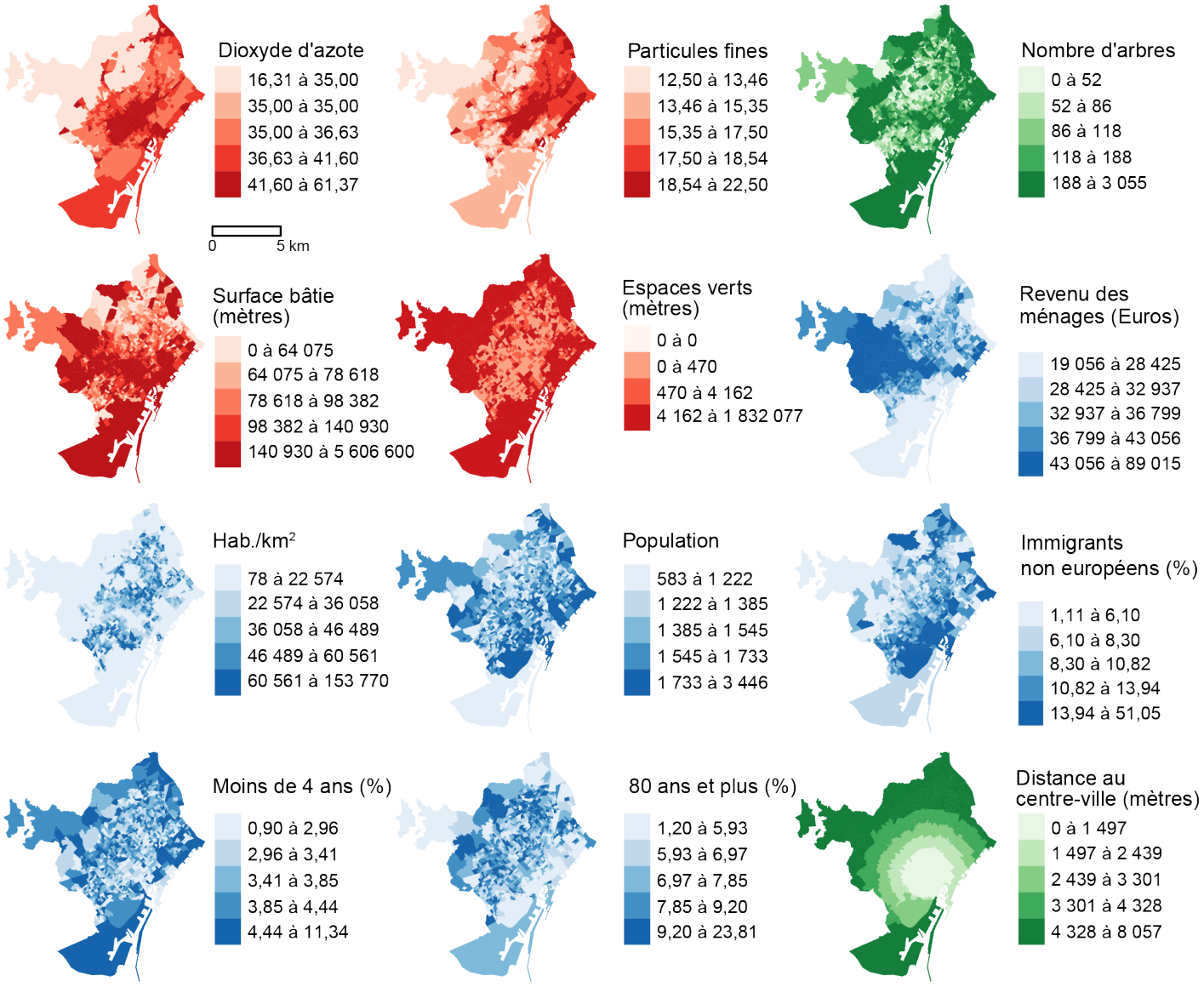
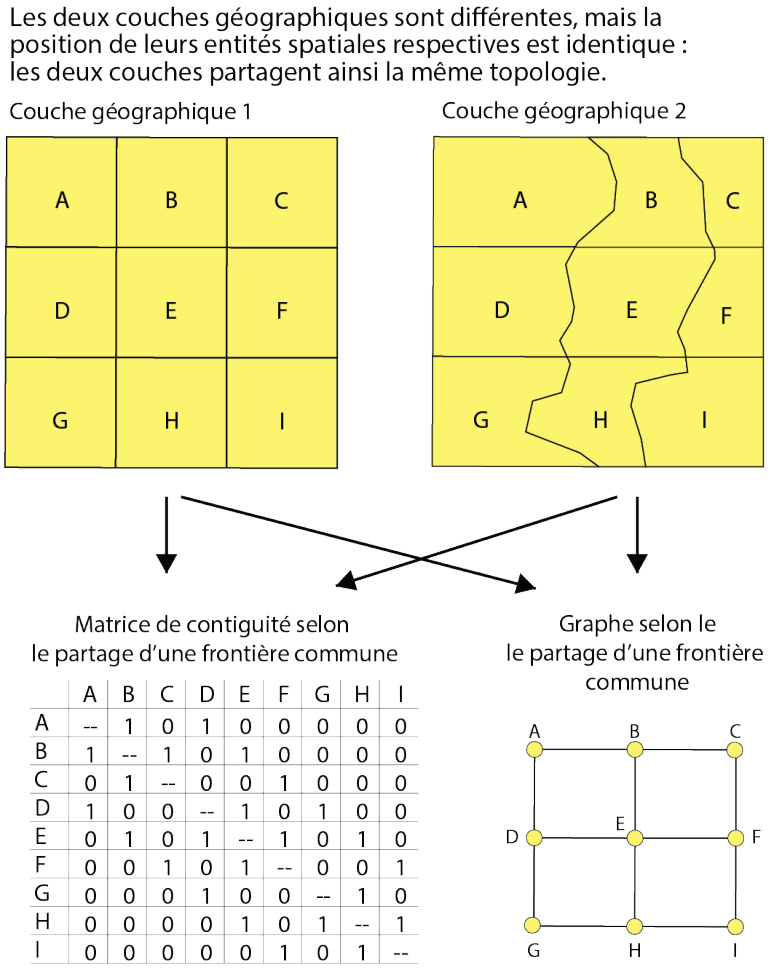


FIGURE 1.3 – Cartographie des variables du jeu de données de Barcelone



	A	B	C	D	E	F	G	H	I
A	--	1	0	1	0	0	0	0	0
B	1	--	1	0	1	0	0	0	0
C	0	1	--	0	0	1	0	0	0
D	1	0	0	--	1	0	1	0	0
E	0	1	0	1	--	1	0	1	0
F	0	0	1	0	1	--	0	0	1
G	0	0	0	1	0	0	--	1	0
H	0	0	0	0	1	0	1	--	1
I	0	0	0	0	0	1	0	1	--

FIGURE 1.4 – Relation topologique entre des entités spatiales polygonales

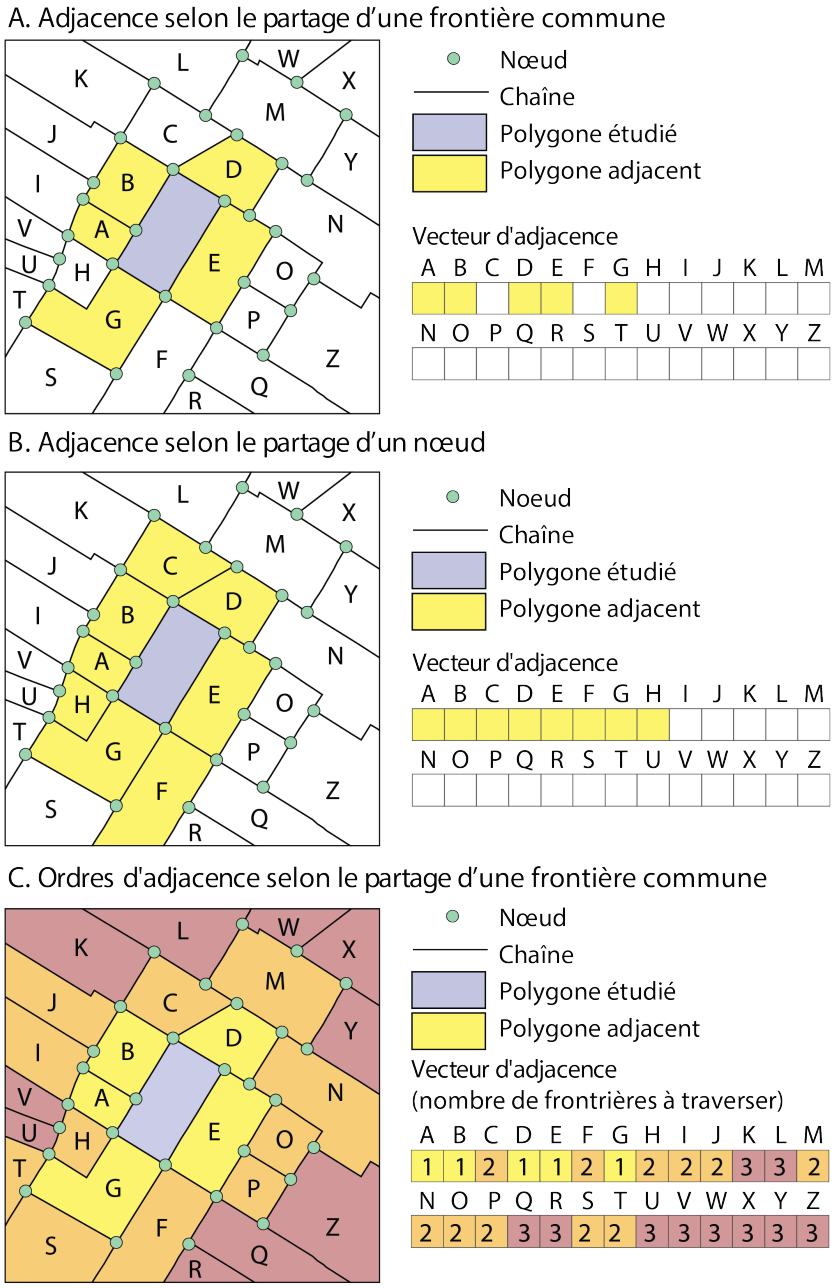


FIGURE 1.5 – Relations de voisinage et évaluation de la contiguïté

Par contre, comme leurs noms l'indiquent, le calcul des distances réticulaires (figure 1.6, b) nécessite d'avoir un réseau de rues dans un système d'information géographique (par exemple, l'extension *Network Analyst* d'ArcGIS Pro) ou dans R (notamment avec le *package* `R5R`) pour calculer le chemin le plus rapide (voir le chapitre intitulé *Mesures d'accessibilité spatiale selon différents modes de transport* (Apparicio et Gelb 2024)).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1.3}$$

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \tag{1.4}$$

$$d_{ij} = 2R \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\delta_i - \delta_j}{2}\right) + \cos\,\delta_i \cdot \cos\,\delta_j \cdot \sin^2\left(\frac{\phi_i - \phi_j}{2}\right)}\right) \tag{1.5}$$

avec R étant le rayon de la terre; δ_i et δ_j les coordonnées de longitude pour les points i et j; ϕ_i et ϕ_j les coordonnées de latitude pour les points i et j.

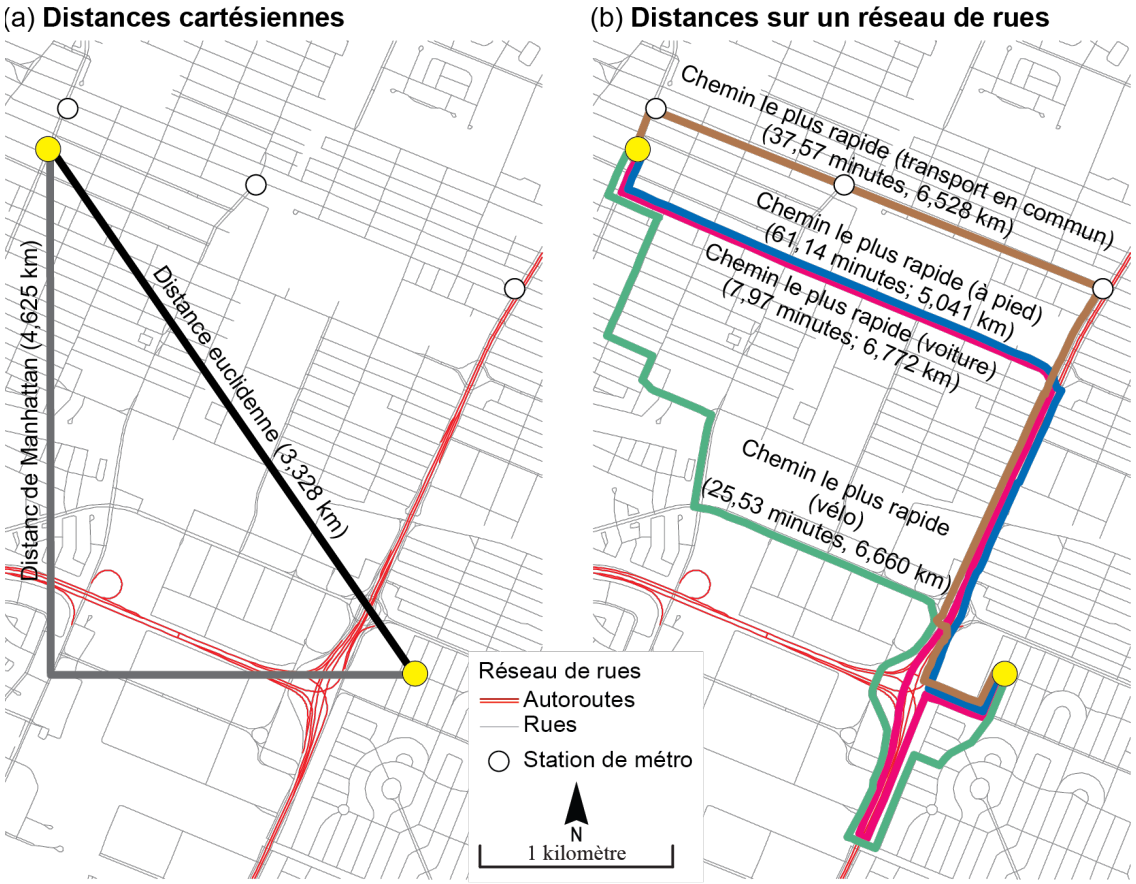


Figure adaptée de Apparicio et al. (2017).

FIGURE 1.6 – Les différents types de distance

1.2.2.1 Matrice de distance binaire (de connectivité)

À partir d'une matrice de distance entre les entités spatiales d'une couche géographique, il est possible de créer une matrice de pondération binaire (équation 1.6). Ce type de matrice est habituellement appelée **matrice de connectivité**. Il convient alors de fixer un seuil de distance maximal. Par exemple, avec un seuil de 500 mètres, $w_{ij} = 1$ si la distance entre les entités spatiales i et j est inférieure ou égale à 500 mètres; sinon $w_{ij} = 0$. Notez que pour des lignes et des polygones, la distance est habituellement calculée à partir de leurs centroïdes.

$$w_{ij} = \begin{cases} 1 & \text{si } d_{ij} \leq \bar{d}; i \neq j \\ 0 & \text{sinon} \end{cases} \tag{1.6}$$

avec d_{ij} étant la distance entre les entités spatiales i et j, et $\bar{d}$ étant un seuil de distance maximal fixé par la personne utilisatrice (par exemple, 500 mètres).

En guise d'exemple, à la figure 1.7, seuls les polygones jaunes seraient considérés comme voisins du polygone bleu avec un seuil de distance maximal fixé à 2,5 kilomètres (valeur de 1); les roses se verraient affecter la valeur de 0.

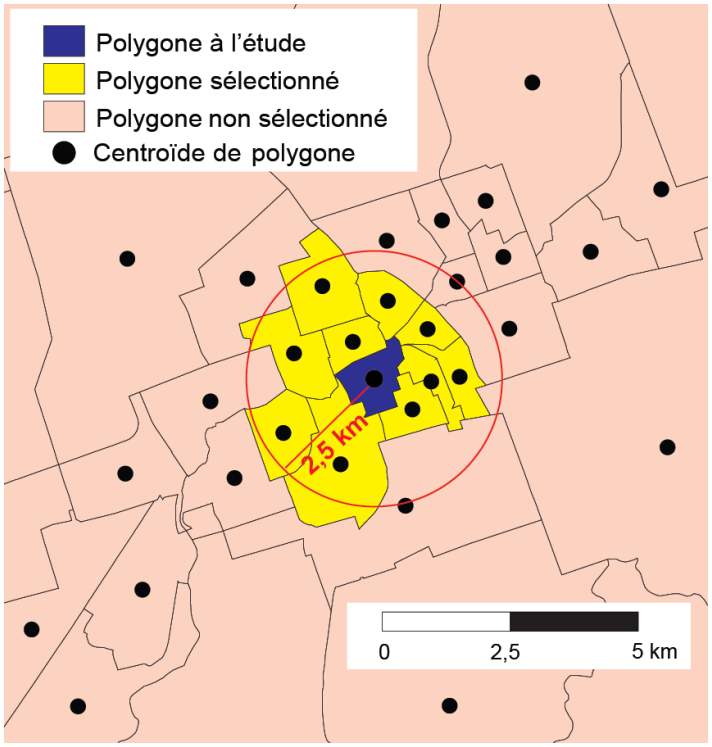


FIGURE 1.7 – Illustration de la connectivité basée sur la distance

1.2.2.2 Matrices basées sur la distance

Une fois les distances calculées entre les entités spatiales, les pondérations peuvent être calculées avec l'inverse de la distance ($1/d_{ij}$) ou l'inverse de la distance au carré ($1/d_{ij^2}$) (équation 1.7).

$$w_{ij} = \begin{cases} \frac{1}{d_{ij}^{\gamma}} & \\ 0 & \text{si } i = j \end{cases} \tag{1.7}$$

avec $\gamma = 1$ pour une matrice de l'inverse de la distance et $\gamma = 2$ pour l'inverse de la distance au carré.

Analysons le graphique à la figure 1.8. Premièrement, nous constatons que plus la distance est grande, plus la valeur de la pondération est faible et inversement. De la sorte, nous accordons un rôle plus important aux entités spatiales proches les unes des autres qu'à celles éloignées. Deuxièmement, les pondérations chutent beaucoup plus rapidement avec l'inverse de la distance au carré qu'avec l'inverse de la distance. Autrement dit, le recours à une matrice de pondération calculée avec l'inverse de la distance au carré a comme effet d'accorder un poids plus important aux entités géographiques très proches.

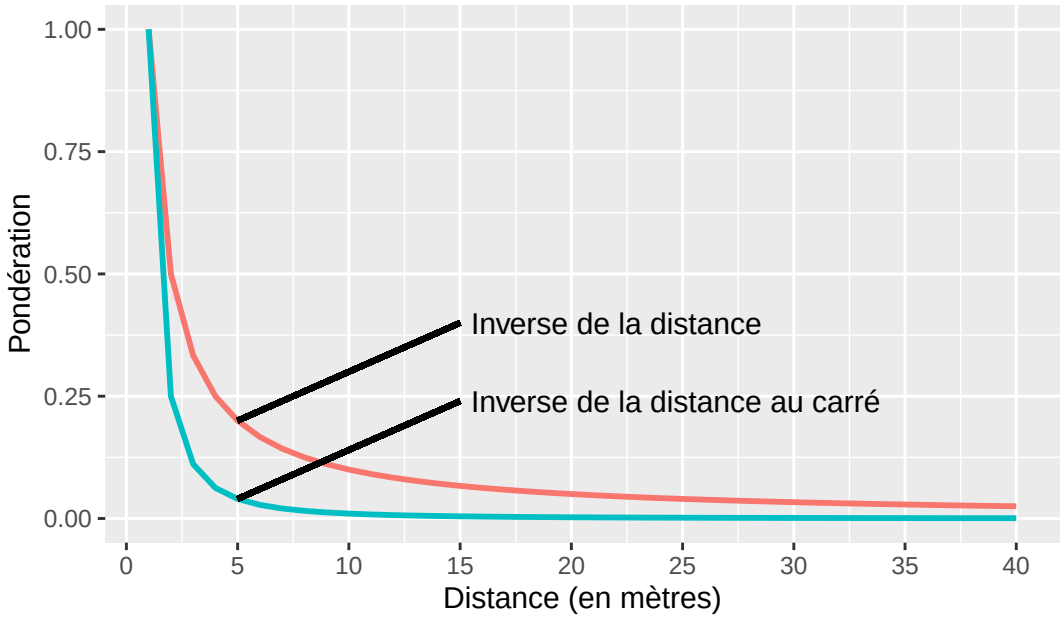


FIGURE 1.8 – Comparaison des matrices inverse de la distance et inverse de la distance au carré

Notez que l'équation 1.7 peut être légèrement modifiée en introduisant un seuil maximal de la distance au-delà duquel les pondérations sont mises à 0 (équation 1.8). Autrement dit, cela permet de ne pas tenir compte des entités spatiales distantes à plus d'un seuil fixé par l'analyste, ce qui est particulièrement intéressant lorsque vous analysez un phénomène dont la diffusion (ou propagation) cesse au-delà d'une certaine distance.

$$w_{ij} = \begin{cases} \frac{1}{d_{ij}^{\gamma}} & \text{si } d_{ij} \leq \bar{d} \\ 0 & \text{si } d_{ij} > \bar{d} \\ 0 & \text{si } i = j \end{cases} \tag{1.8}$$

1.2.2.3 Matrices selon le critère des plus proches voisins

Une autre façon très utilisée pour définir une matrice de proximité à partir d'une matrice de distance consiste à retenir uniquement les *n* plus proches voisins. La matrice est aussi binaire avec les valeurs de 1 si les observations sont parmi les

n plus proches de l'entité spatiale i et de 0 pour une situation inverse.

1.2.3 Standardisation des matrices de pondération spatiale en ligne

Il est recommandé de standardiser les matrices de pondération en ligne. La somme de la matrice de pondération sera alors égale au nombre d'entités spatiales de la couche géographique.

⚠ **Attention**

Quel est l'intérêt de la standardisation?
Nous verrons dans les sections suivantes que ces matrices sont utilisées pour évaluer le degré d'autocorrélation spatiale globale et locale. Or, il est fréquent de comparer les valeurs des mesures d'autocorrélation spatiale obtenues avec différentes matrices d'adjacence et de proximité (contiguïté selon le partage d'un nœud, d'une frontière commune; inverse de la distance, inverse de la distance au carré, etc.). Autrement dit, la standardisation des matrices de pondération spatiale permet de vérifier si le degré de (dis)ressemblance des entités spatiales en fonction d'une variable donnée est plus fort avec une matrice de contiguïté, d'inverse de la distance, d'inverse de la distance au carré, etc.

Pour illustrer comment réaliser une standardisation, nous utilisons une couche géographique comprenant peu d'entités spatiales, soit celle des quatre arrondissements de la ville de Sherbrooke (figure 1.9).

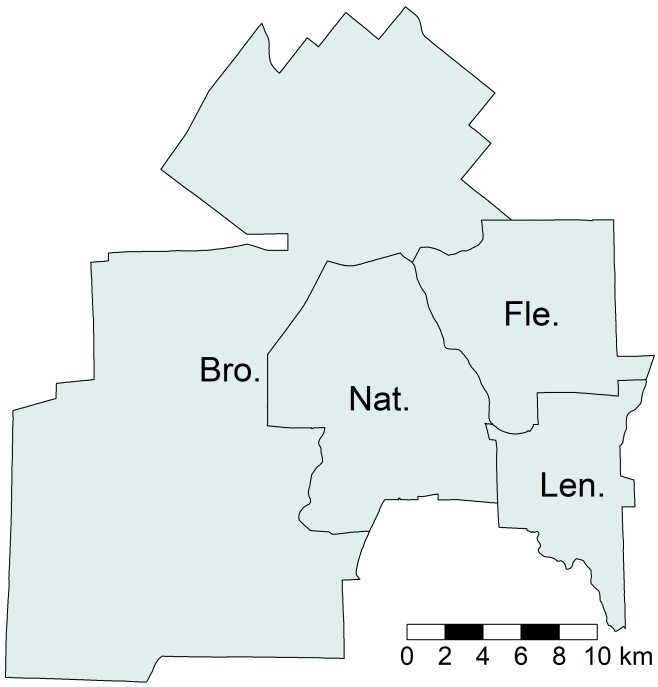


Figure 1.9 – Arrondissements de la ville de Sherbrooke

Au tableau 1.5, différentes matrices de contiguïté et de distance ont été calculées, puis standardisées. Voici comment interpréter les différentes sections du tableau :

- **Contiguïté selon le partage d'une frontière commune.** La valeur de 1 signale que deux arrondissements sont voisins, sinon la valeur est à 0. Tel qu'indiqué aux équations 1.1 et 1.2, un arrondissement ne peut être voisin de lui-même (ex.: valeur de 0 pour la cellule `Bro.` et `Bro.`). L'arrondissement de Brompton–Rock Forest–Saint-Élie–Deauville (`Bro.`) a deux voisins, soit ceux des Nations et de Fleurimont (`Nat.` et `Fle.`), comme indiqué par la valeur 2 dans la colonne `total`. Par contre, les arrondissements des Nations et de Fleurimont sont voisins de tous les autres (valeur de 3 dans la colonne `total`).
- **Standardisation de la matrice de contiguïté.** Il suffit de diviser chaque valeur de la matrice de contiguïté par la somme de la ligne correspondante. De la sorte, la somme de chaque ligne est égale à 1 et la somme de l'ensemble des valeurs de la matrice est égale au nombre d'entités spatiales (ici 4).
- **Distance (km).** Nous avons calculé la distance euclidienne en kilomètres entre les centroïdes des arrondissements.
- **Inverse de la distance.** Les valeurs sont obtenues avec la formule $1/_{dij}$. Par exemple, entre `Bro.` et `Nat.`, nous avons $1/7,9930 = 0,1251$.
- **Inverse de la distance au carré.** Les valeurs sont obtenues avec la formule $1/_{dij^2}$. Par exemple, entre `Bro.` et `Nat.`, nous avons $1/7,9930^2 = 0,0160$.
- **Standardisation de l'inverse de la distance.** Comme précédemment, il suffit de diviser chaque valeur de la matrice par la somme de la ligne correspondante. Par exemple, pour `Bro.` et `Nat.`, nous avons $0,1251/0,3241 = 0,3860$. Remarquez que la somme des lignes est bien égale à 1.
- **Standardisation de l'inverse de la distance au carré.** Comme précédemment, il suffit de diviser chaque valeur de la matrice par la somme de la ligne correspondante. Par exemple, pour `Bro.` et `Nat.`, nous avons $0,0160/0,0360 = 0,4440$. Remarquez que la somme des lignes est bien égale à 1.

Tableau 1.5 – Standardisation de matrices de pondération spatiale

Arrondissement	Bro.	Nat.	Len.	Fle.	Somme (lignes)
Matrice de contiguïté selon le partage d'une frontière commune					
Bro.	0,0000	1,0000	0,0000	1,0000	2,0000
Nat.	1,0000	0,0000	1,0000	1,0000	3,0000
Len.	0,0000	1,0000	0,0000	1,0000	2,0000
Fle.	1,0000	1,0000	1,0000	0,0000	3,0000
Standardisation de la matrice de contiguïté					
Bro.	0,0000	0,5000	0,0000	0,5000	1,0000
Nat.	0,3330	0,0000	0,3330	0,3330	1,0000
Len.	0,0000	0,5000	0,0000	0,5000	1,0000
Fle.	0,3330	0,3330	0,3330	0,0000	1,0000
Distance (km)					
Bro.	0,0000	7,9930	18,9940	16,1140	
Nat.	7,9930	0,0000	11,1190	9,1650	
Len.	18,9940	11,1190	0,0000	9,2590	
Fle.	16,1140	9,1650	9,2590	0,0000	
Matrice selon l'inverse de la distance					
Bro.	0,0000	0,1251	0,0526	0,0621	0,2398
Nat.	0,1251	0,0000	0,0899	0,1091	0,3241
Len.	0,0526	0,0899	0,0000	0,1080	0,2505
Fle.	0,0621	0,1091	0,1080	0,0000	0,2792
Matrice selon l'inverse de la distance au carré					

Bro.	0,0000	0,0160	0,0030	0,0040	0,0230
Nat.	0,0160	0,0000	0,0080	0,0120	0,0360
Len.	0,0030	0,0080	0,0000	0,0120	0,0230
Fle.	0,0040	0,0120	0,0120	0,0000	0,0280
Standardisation de l'inverse de la distance					
Bro.	0,0000	0,5220	0,2190	0,2590	1,0000
Nat.	0,3860	0,0000	0,2770	0,3370	1,0000
Len.	0,2100	0,3590	0,0000	0,4310	1,0000
Fle.	0,2220	0,3910	0,3870	0,0000	1,0000
Standardisation de l'inverse de la distance au carré					
Bro.	0,0000	0,6960	0,1300	0,1740	1,0000
Nat.	0,4440	0,0000	0,2220	0,3330	1,0000
Len.	0,1300	0,3480	0,0000	0,5220	1,0000
Fle.	0,1430	0,4290	0,4290	0,0000	1,0000

1.2.4 Mise en œuvre dans R

⚠ **Attention**

Construction des matrices dans R avec le *package* `spdep`.
Le *package* `spdep` dispose de différentes fonctions pour construire des matrices de contiguïté, de connectivité et de distance :

- `poly2nb` pour des matrices de contiguïté.
- `nblag` et `nblag_cumul` pour des matrices de contiguïté avec des ordres d'adjacence.
- `dnearneigh` pour des matrices de connectivité.
- `as.matrix(dist(coords))` et `mat2listw` pour des matrices de distance.
- `knn2nb` pour des matrices selon le critère des plus proches voisins.

1.2.4.1 Matrices de pondération spatiale selon la contiguïté

Pour créer des matrices de pondération spatiale selon la contiguïté, nous utilisons deux fonctions du *package* `spdep` :

- `poly2nb(objet sf, queen = TRUE)` crée une matrice de contiguïté sous la forme d'une classe `nb` (liste de voisins). Avec le paramètre `queen = TRUE`, la contiguïté est évaluée selon le partage d'un nœud; avec `queen = FALSE`, la contiguïté est évaluée selon le partage d'un segment (frontière). La matrice spatiale comprend une ligne par polygone avec les index de ceux qui sont adjacents. Par exemple, `Queen[[1]]` renvoie la liste des polygones voisins à la première entité spatiale, soit `24 36 44 73`, c'est-à-dire quatre voisins.
- `nb2listw(objet nb, zero.policy = TRUE, style = "W")` crée une matrice de pondération spatiale à partir de n'importe quelle matrice spatiale (de contiguïté ou de distance). Le paramètre `style = "W"`, qui est défini par défaut, permet de standardiser la matrice en ligne. Par exemple, `w_queen$weights[[1]]` renvoie les valeurs des pondérations pour la première entité spatiale, soit `0.25 0.25 0.25 25` (0,25 = 1 / 4 voisins). Pour obtenir une matrice non standardisée, vous devez écrire `style = "B"`, alors `w_queen$weights[[1]]` renverra les valeurs de `1 1 1 1`.

```
library(spdep)

## Utilisation du jeu de données sur l'agglomération de Lyon
load("data/Lyon.Rdata")

## Matrice selon le partage d'un nœud (Queen)
# Création de la matrice spatiale
nb_queen <- poly2nb(LyonIris, queen = TRUE)

# Affichage de la première ligne de la matrice
nb_queen[[1]]
```

```
[1] 27 36 44 73
```

```
# Création de la matrice de pondération avec une standardisation en ligne
w_queen <- nb2listw(nb_queen, zero.policy = TRUE, style = "W")

# Affichage de la première ligne des pondérations standardisées
w_queen$weights[[1]]
```

```
[1] 0.25 0.25 0.25 0.25
```

```
cat("La somme de la première ligne de la matrice de pondération est égale à",
    sum(w_queen$weights[[1]]))
```

```
La somme de la première ligne de la matrice de pondération est égale à 1
```

```
# Affichage de la première ligne des pondérations non standardisée
nb2listw(nb_queen, zero.policy = TRUE, style = "B")$weights[[1]]
```

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

```
## Matrice selon le partage d'un segment (Rook)
nb_rook <- poly2nb(LyonIris, queen = FALSE)
w_rook <- nb2listw(nb_rook, zero.policy = TRUE, style = "W")

## Comparaison des deux matrices de contiguïté
# Résultat de la matrice de pondération (Queen)
summary(w_queen)
```

```
Characteristics of weights list object:
Neighbour list object:
```

```
Number of regions: 506
Number of nonzero links: 2854
Percentage nonzero weights: 1.114687
Average number of links: 5.640316
Link number distribution:

  2   3   4   5   6   7   8   9  10  11  12  14  17
 13  50  84 108 103  70  41  22   7   5   1   1   1
13 least connected regions:
81 91 105 148 160 174 183 325 425 468 480 489 506 with 2 links
1 most connected region:
154 with 17 links

Weights style: W
Weights constants summary:
    n     nn  S0       S1       S2
W 506 256036 506 192.1431 2097.903
```

```
# Résultat de la matrice de pondération (Rook)
summary(w_rook)
```

```
Characteristics of weights list object:
Neighbour list object:
Number of regions: 506
Number of nonzero links: 2660
Percentage nonzero weights: 1.038916
Average number of links: 5.256917
Link number distribution:

  2   3   4   5   6   7   8   9  10  11  12  15
 14  60 104 126  97  58  24  13   6   2   1   1
14 least connected regions:
81 91 105 148 160 174 183 325 376 425 468 480 489 506 with 2 links
1 most connected region:
154 with 15 links

Weights style: W
Weights constants summary:
    n     nn  S0       S1       S2
W 506 256036 506 204.0416 2099.405
```

1.2.4.2 Matrices de pondération spatiale selon la contiguïté et un ordre d'adjacence

Le code ci-dessous génère les matrices de pondération spatiale standardisée en ligne selon les ordres d'adjacence de 1 à 3.

```
# Création des matrices d'ordre 1, 2 et 3
nb_queen1 <- poly2nb(LyonIris, queen = TRUE)
nb_queen2 <- nblag_cumul(nblag(nb_queen1, 2))
nb_queen3 <- nblag_cumul(nblag(nb_queen1, 3))

# Création des matrices de pondération spatiale standardisée en ligne
w_queen1 <- nb2listw(nb_queen1, zero.policy = TRUE, style = "W")
w_queen2 <- nb2listw(nb_queen2, zero.policy = TRUE, style = "W")
w_queen3 <- nb2listw(nb_queen3, zero.policy = TRUE, style = "W")
```

1.2.4.3 Matrice de connectivité (matrice distance binaire)

La fonction `dnearneigh(matrice des coordonnées ou points sf, d1 =, d2 =)` crée une matrice de connectivité à partir d'une couche de points. Les paramètres `d1` et `d2` permettent de spécifier le rayon de recherche (ex. : avec `d1 = 0` et `d2 = 2500`, le seuil maximal de distance est de 2500 mètres). Si votre couche `sf` comprend des `lignes` ou des `polygones`, utilisez la fonction `st_centroid` ou `st_point_on_surface()` pour les convertir en points (section 1.2.2).

```
## Conversion des polygones en points avec st_centroid
iris_centroides <- st_centroid(LyonIris)

## Matrice binaire avec un seuil de 2500 mètres
connect_2500m <- dnearneigh(iris_centroides, d1 = 0, d2 = 2500)

## Matrice de pondération spatiale standardisée en ligne
w_connect_2500m <- nb2listw(connect_2500m, zero.policy = TRUE, style = "W")
```

1.2.4.4 Matrices de pondération spatiale selon l'inverse de la distance et l'inverse de la distance au carré

Le code ci-dessous, qui permet de créer les matrices de l'inverse de la distance et de l'inverse de la distance au carré, comprend les étapes suivantes :

- Récupération des coordonnées géographiques des centroïdes des entités spatiales.
- Création de la matrice avec la distance euclidienne $n \times n$ (n étant le nombre d'entités spatiales de la couche).
- Calcul des matrices d'inverse de la distance et d'inverse de la distance au carré.
- Standardisation de ces deux matrices et transformation en objets `listw` avec la fonction `mat2listw`.

```
## Coordonnées des centroïdes des entités spatiales
coords <- st_coordinates(iris_centroides)

## Création de la matrice avec la distance euclidienne
distances <- as.matrix(dist(coords, method = "euclidean"))

# S'assurer que la diagonale de la matrice est à 0
diag(distances) <- 0
```

```
## Matrices d'inverse de la distance et d'inverse de la distance au carré
inv_distances <- ifelse(distances!=0, 1/distances, distances)
inv_distances2 <- ifelse(distances!=0, 1/distances^2, distances)

## Matrices de pondération spatiale standardisée en ligne
w_inv_distances  <- mat2listw(inv_distances, style = "W")
w_inv_distances2 <- mat2listw(inv_distances2, style = "W")
```

Astuce

Intégration d'autres types de distance
À la section 1.2.2, nous avons vu que plusieurs types de distance peuvent être utilisés : cartésiennes (euclidienne et de Manhattan) et réticulaires (chemin le plus rapide à pied, à vélo, en automobile et en transport en commun). Pour construire une matrice avec la distance de Manhattan, vous devez changer la valeur du paramètre `method` de la fonction `dist` comme suit : `as.matrix(dist(coords, method = "manhattan"))`.
Pour intégrer une distance réticulaire, vous devez la calculer, soit dans R (Apparicio et Gelb 2024), soit dans un logiciel de système d'information géographique (QGIS ou ArcGIS Pro avec l'extension *Network Analyst* par exemple) et l'importer dans R. Le reste du code sera alors identique.

Nous avons vu qu'il est possible d'utiliser une matrice en fixant une distance maximale au-delà de laquelle les pondérations sont mises à 0 (équation 1.8). Le code ci-dessous permet de créer des matrices de pondération standardisée avec l'inverse de la distance et l'inverse de la distance au carré avec des seuils de 2500 et de 5000 mètres.

```
## Coordonnées des centroïdes des entités spatiales
coords <- st_coordinates(iris_centroides)

## Création de la matrice de distance
distances <- as.matrix(dist(coords, method = "euclidean"))

## Création de différentes matrices avec différents seuils
inv_distances_1000m  <- ifelse(distances<=1000 & distances!=0, 1/distances, 0)
inv_distances_2500m  <- ifelse(distances<=2500 & distances!=0, 1/distances, 0)
inv_distances_5000m  <- ifelse(distances<=5000 & distances!=0, 1/distances, 0)
inv_distances2_2500m <- ifelse(distances<=2500 & distances!=0, 1/distances^2, 0)
inv_distances2_5000m <- ifelse(distances<=5000 & distances!=0, 1/distances^2, 0)

## Matrices de pondération spatiale standardisée en ligne
w_inv_distances_1000 <- mat2listw(inv_distances_1000m, style = "W", zero.policy = TRUE)
w_inv_distances_2500 <- mat2listw(inv_distances_2500m, style = "W", zero.policy = TRUE)
w_inv_distances_5000 <- mat2listw(inv_distances_5000m, style = "W", zero.policy = TRUE)
w_inv_distances2_2500 <- mat2listw(inv_distances2_2500m, style = "W", zero.policy = TRUE)
w_inv_distances2_5000 <- mat2listw(inv_distances2_5000m, style = "W", zero.policy = TRUE)
```

Spécifier un seuil de distance trop réduit peut toutefois être problématique. Par exemple, sur les 506 IRIS de l'agglomération de Lyon, 68 n'ont pas de voisins à 1000 mètres, indiqués par le résultat suivant : `68 regions with no links`.

```r
# Résultats de la matrice de pondération spatiale avec un seuil de 1000 mètres
summary(w_inv_distances_1000, zero.policy = TRUE)
```

```
Characteristics of weights list object:
Neighbour list object:
Number of regions: 506
Number of nonzero links: 3992
Percentage nonzero weights: 1.559156
Average number of links: 7.889328
68 regions with no links:
5 9 11 16 20 22 26 29 39 46 49 51 59 64 67 69 79 81 88 97 98 99 103 105
108 113 114 117 118 121 124 126 128 129 130 131 133 135 137 141 143 151
153 157 158 160 161 162 163 165 166 167 168 169 171 176 177 178 179 181
184 185 186 370 377 387 415 506
86 disjoint connected subgraphs
Link number distribution:

 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
68 39 25 28 27 30 34 40 38 14 18 12 13 16  8 10 11 11 10 13  7 11  7  3  5  5
26 28
 2  1
39 least connected regions:
7 8 13 14 15 25 28 35 41 42 43 48 52 56 61 65 71 72 85 86 91 93 94 96 106 116 138 140 148 155 172 174 183 187 22
1 most connected region:
424 with 28 links

Weights style: W
Weights constants summary:
    n     nn  S0       S1       S2
W 438 191844 438 203.0076 1789.585
```

⚠ Attention

Réduction de la taille des matrices de distance

Plusieurs logiciels (notamment ArcGIS Pro et GeoDa) réduisent par défaut la taille des matrices de distance de la façon suivante : 1) construction d'une matrice de distance uniquement pour l'entité la plus proche (la matrice résultante est donc de dimension $n \times 1$); 2) obtention de la distance maximale dans cette matrice, soit la distance la plus grande entre une entité spatiale et celle la plus proche; 3) construction de la matrice de distance finale avec comme seuil la distance maximale obtenue à l'étape précédente.

Cette réduction procure deux avantages importants :

- **Une diminution considérable des temps de calcul**, surtout pour les couches géographiques comprenant un nombre très élevé d'entités spatiales. Par exemple, avec une couche de 50 entités spatiales, la matrice des distances comprendra 2500 valeurs ($50 \times 50 = 2500$) tandis qu'avec 1000 entités spatiales, elle en comprendra un million ($1000 \times 1000 = 1\,000\,000$).
- Comme décrit plus haut, il est préférable d'**éviter d'avoir une matrice de distances avec des entités spatiales sans voisins**, puisque cela a un impact négatif sur les mesures d'autocorrélation spatiale.

La syntaxe ci-dessous permet ainsi de construire des matrices de pondération (inverse de la distance et inverse de la distance au carré) à partir de la distance maximale d'un SR et son voisin le plus proche.

```
## Coordonnées des centroïdes des entités spatiales
coords <- st_coordinates(iris_centroides)

## Trouver le plus proche voisin
k1 <- knn2nb(knearneigh(coords))

## Affichage de la distance la plus proche pour les 506 IRIS
round(unlist(nbdists(k1, coords)), 0)
```

```
  [1]  506  739  776  451 1000  284  547  629 1206  449 1145  603  727  965  370
 [16] 1835  389  334  725 1264  523 1543  449  710  990 1692  617  643 1145  509
 [31]  489  488  600  334  841  407  770  301 1072  884  426  985  899  417  249
 [46] 1316  588  426 1067  402 1366  718  452  657  696  600  759  482 1841  225
 [61]  999  452  229 1616  418  220 1144  547 1482  284  985  926  417  716  678
 [76]  355  345  603 1020  188 2040  325  841  770  767  727  767 1082  614  622
 [91]  746  225  974  841  348  926 1108 1354 1290  545  220  355 1723  338 1686
[106]  370  188 1130  249  451  348  340 1063 1256  471  746 1094 1507  303  471
[121] 2332  831  301 1129  229 1638  677 2237 1092 2157 1050  452 1058  389 2523
[136]  682 1385  418  692  629 1097  459 1543  520  340  325  617  718  303  495
[151] 2218  369 1175  619  993  392 3366 1072  453 1050 1464 1385 1780  657 1181
[166] 1170 1324 1170 1477  352 1477  752  394  852  611 1562 1279 1279 1539  352
[181] 1035  611  394 1834 1300 1057  852  249  460  427  370  272  410  343  319
[196]  346  235  271  491  159  398  386  272  403  512  325  209  286  270  261
[211]  286  157  832  452  450  464  280  492  236  561  276  186  804  219  576
[226]  290  275  560  321  221  583  252  456  460  325  191  314  756  346  482
[241]  472  523  863  259  271  247  238  341  383  699  423  247  221  509  358
[256]  385  316  539  219  602  368  883  599  346  393  245  599  492  940  398
[271]  458  290  414  475  321  286  438  276  347  660  394  155  269  249  328
[286]  393  503  270  392  630  269  281  725  394  408  259  493  517  523  459
[301]  284  438  390  280  311  407  489  534  329  413  301  368  311  462  289
[316]  251  276  387  293  815  467  340  492  261  165  475  602  235  404  384
[331]  976  300  407  486  309  211  319  275  341  756  283  395  299  469  360
[346]  360  870  261  159  211  530  236  292  209  329  346  519  531  597  394
[361]  290  290  165  270  531  343  175  335  281 1046  398  390  267  267  507
[376]  358 1071  341  249  361  292  540  373  157  273  286 1173  253  423  442
[391]  472  274  299  318  335  345  487  503  278  442  341  660  249  238  238
[406]  511  295  725  469  220  481  422  416  456 1038  416  509  318  462  245
[421]  658  341  570  175  291  374  269  355  408  312  514  287  786  520  380
[436]  286  360  520  800  444  475  343  438  641  383  444  508  325  530  448
[451]  533  562  490  290  293  284  155  252  318  289  348  429  210  318  447
[466]  318  297  546  495  311  376  376  443  546  220  277  327  688  362  146
[481]  830  460  431  620  603  725  396  495  362  277  146  298  422  389  389
[496]  422  327  678  688  425  607  289  522  396  431 2365
```

```
## Trouver la distance maximale
plusprochevoisin_max <- max(unlist(nbdists(k1, coords)))
cat("Distance maximale au plus proche voisin :", round(plusprochevoisin_max, 0), "mètres")
```

```
Distance maximale au plus proche voisin : 3366 mètres
```

```
## Matrices des distances avec la valeur maximale
# Voisins les plus proches avec le seuil de distance maximale
voisins_distmax <- dnearneigh(coords, 0, plusprochevoisin_max)

# Distances avec le seuil maximum
distances <- nbdists(voisins_distmax, coords)

# Inverse de la distance
inv_distances <- lapply(distances, function(x) (1/x))

# Inverse de la distance au carré
inv_distances2 <- lapply(distances, function(x) (1/x^2))

## Matrices de pondération spatiale standardisée en ligne
w_inv_distances  <- nb2listw(voisins_distmax, glist = inv_distances, style = "W", zero.policy = TRUE)
w_inv_distances2 <- nb2listw(voisins_distmax, glist = inv_distances2, style = "W", zero.policy = TRUE)
```

1.2.4.5 Matrices de pondération spatiale selon le critère des plus proches voisins

La fonction `knearneigh` du *package* `spdep` crée des matrices de distance selon le critère des plus proches voisins, dont le nombre est fixé avec le paramètre `k`.

```
## Coordonnées géographiques des centroïdes des polygones
coords <- st_coordinates(st_centroid(LyonIris))

## Matrices des plus proches voisins de 2 à 5
k2 <- knn2nb(knearneigh(coords, k = 2))
k3 <- knn2nb(knearneigh(coords, k = 3))
k4 <- knn2nb(knearneigh(coords, k = 4))
k5 <- knn2nb(knearneigh(coords, k = 5))

## Matrices de pondération spatiale standardisée en ligne
w_k2 <-  nb2listw(k2, zero.policy = FALSE, style = "W")
w_k3 <-  nb2listw(k3, zero.policy = FALSE, style = "W")
w_k4 <-  nb2listw(k4, zero.policy = FALSE, style = "W")
w_k5 <-  nb2listw(k5, zero.policy = FALSE, style = "W")
```

Astuce

L'ensemble des matrices en quelques lignes de code!

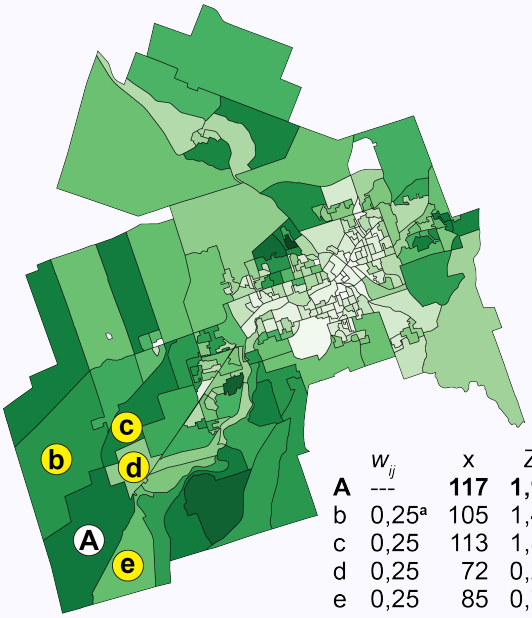
```
library(dplyr)

# Nom de la couche sf départ à changer au besoin
Couche.sf <- LyonIris

# Matrice de contiguïté
nb_queen <- poly2nb(Couche.sf, queen = TRUE)
nb_rook  <- poly2nb(Couche.sf, queen = FALSE)
w_queen <- nb2listw(nb_queen, zero.policy = TRUE, style = "W")
w_rook <- nb2listw(nb_rook, zero.policy = TRUE, style = "W")

# Matrice de contiguïté (ordre 2 à 5)
w_rook2 <- nblag_cumul(nblag(nb_rook, 2)) %>% nb2listw(zero.policy = TRUE, style = "W")
w_rook3 <- nblag_cumul(nblag(nb_rook, 3)) %>% nb2listw(zero.policy = TRUE, style = "W")
w_rook4 <- nblag_cumul(nblag(nb_rook, 4)) %>% nb2listw(zero.policy = TRUE, style = "W")
w_rook5 <- nblag_cumul(nblag(nb_rook, 5)) %>% nb2listw(zero.policy = TRUE, style = "W")

# Matrice de connectivité binaire
centroides <- st_centroid(Couche.sf)
w_connect_2500m <- dnearneigh(centroides, d1 = 0, d2 = 2500) %>%
                   nb2listw(zero.policy = TRUE, style = "W")

# Inverse de la distance et inverse de la distance au carré (matrices complètes)
distances <- as.matrix(dist(st_coordinates(centroides), method = "euclidean"))
diag(distances) <- 0
w_inv_distances <- ifelse(distances!=0, 1/distances, distances) %>% mat2listw(style = "W")
w_inv_distances2 <- ifelse(distances!=0, 1/distances^2, distances) %>% mat2listw(style = "W")

# Inverse de la distance et inverse de la distance au carré (matrices réduites)
coords <- st_coordinates(centroides)
plusprochevoisin_max <- max(unlist(nbdists(knn2nb(knearneigh(coords)),coords)))
dists <- nbdists(dnearneigh(coords, 0, plusprochevoisin_max), coords)
w_inv_distances_reduite <- lapply(dists, function(x) (1/x)) %>%
                           nb2listw(voisins_distmax, glist = ., style = "W", zero.policy = TRUE)
w_inv_distances2_reduite <- lapply(dists, function(x) (1/x^2)) %>%
                            nb2listw(voisins_distmax, glist = ., style = "W", zero.policy = TRUE)

# Inverse de la distance et inverse de la distance au carré avec un seuil de distance
w_inv_distances_1000m  <- ifelse(distances<=1000 & distances!=0, 1/distances, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)
w_inv_distances_2500m  <- ifelse(distances<=2500 & distances!=0, 1/distances, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)
w_inv_distances_5000m  <- ifelse(distances<=5000 & distances!=0, 1/distances, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)
w_inv_distances2_2500m <- ifelse(distances<=2500 & distances!=0, 1/distances^2, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)
w_inv_distances2_5000m <- ifelse(distances<=5000 & distances!=0, 1/distances^2, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)

## Matrices des plus proches voisins de 2 à 5
w_k2 <- knn2nb(knearneigh(coords, k = 2)) %>% nb2listw(zero.policy = FALSE, style = "W")
```

1.3 Variable spatialement décalée

À partir des matrices de pondérations standardisées en ligne, qu'elles soient de contiguïté ou de proximité, il est possible de calculer une version spatialement décalée d'une variable initiale. De la sorte, pour une entité spatiale *i*, il est possible de connaître la valeur moyenne d'une variable pour ses entités voisines ou proches (dépendamment si la matrice est définie selon la contiguïté ou la proximité). Au chapitre 2, nous verrons que ces variables spatiales décalées sont utilisées dans plusieurs modèles d'économétrie spatiale.

Note

Comment calculer une variable spatialement décalée avec une matrice de pondération spatiale?
À la figure 1.10, nous détaillons le calcul de la valeur d'une variable spatialement décalée pour l'entité spatiale **A** à partir d'une matrice de contiguïté (selon le partage d'un segment) standardisée en ligne. Notez que **A** est adjacente à quatre entités spatiales (b, c, d et e).



	w_{ij}	x	Zx	W.x	W.Zx
A	---	**117**	**1,96**[b]		
b	0,25[a]	105	1,49	26,25[c]	0,37[d]
c	0,25	113	1,81	28,25	0,45
d	0,25	72	0,20	18,00	0,05
e	0,25	85	0,71	21,25	0,18
				93,75	**1,05**

Moyenne = 66,96. Écart-type = 25,50.
[a] w_{ij} = poids standardisés en ligne de la matrice de contiguïté, soit 1/4 voisins.
[b] cote z = (117 − 66,96) / 25,50 = 1,96
[c] 105 x 0,25 = 26,25 [d] 1,49 x 0,25 = 0,37
$W.x_a$ = 93,75, soit la moyenne de X pour les entités spatiales adjacentes à A
$W.Zx_a$ = 1,05, soit la moyenne de la cote Z pour les entités spatiales adjacentes à A

FIGURE 1.10 – Illustration du calcul d'une variable spatialement décalée

La fonction `lag.listw` du *package* `spdep` permet de créer une variable spatialement décalée en spécifiant la matrice de pondération spatiale et un vecteur numérique : `wzx <- lag.listw(listW, zx)`. Dans la syntaxe ci-dessous, nous créons une version spatiale décalée de la variable *dioxyde d'azote* (`NO2`) de la couche `sf` *LyonIris* avec une matrice de pondération spatiale standardisée en ligne (définie selon la contiguïté avec le partage d'un segment, `w_rook`), puis nous cartographions les deux versions de la variable (initiale et spatialement décalée) à la figure 1.11.

```
# Chargement des données
library(tmap)
library(spdep)
load("data/Lyon.Rdata")
# Matrice de pondération spatiale standardisée en ligne selon la contiguïté
w_rook<- nb2listw(poly2nb(LyonIris, queen = FALSE), zero.policy = TRUE, style = "W")

# Variable spatialement décalée
LyonIris$W_NO2 <- lag.listw(w_rook, LyonIris$NO2)

# Cartographie avec tmap
tmap_mode("plot")
legende_parametres <- list(text.separator = "à",
                           decimal.mark = ",",
                           big.mark = " ")

carte1 <- tm_shape(LyonIris)+
            tm_borders(col = "gray25", lwd=.2)+
            tm_fill(col = "NO2", palette = "Reds", n = 5, style = "quantile",
                  legend.format = legende_parametres,
                      title = "(a) Variable initiale (NO2)")+
          tm_layout(frame = FALSE)+tm_scale_bar(breaks=c(0,5))

carte2 <- tm_shape(LyonIris)+
            tm_borders(col = "gray25", lwd=.2)+
            tm_fill(col = "W_NO2", palette = "Reds", n = 5, style = "quantile",
                      legend.format = legende_parametres,
                      title = "(b) Spatialement décalée (NO2)")+
          tm_layout(frame = FALSE)

tmap_arrange(carte1, carte2, ncol = 2, nrow = 1)
```

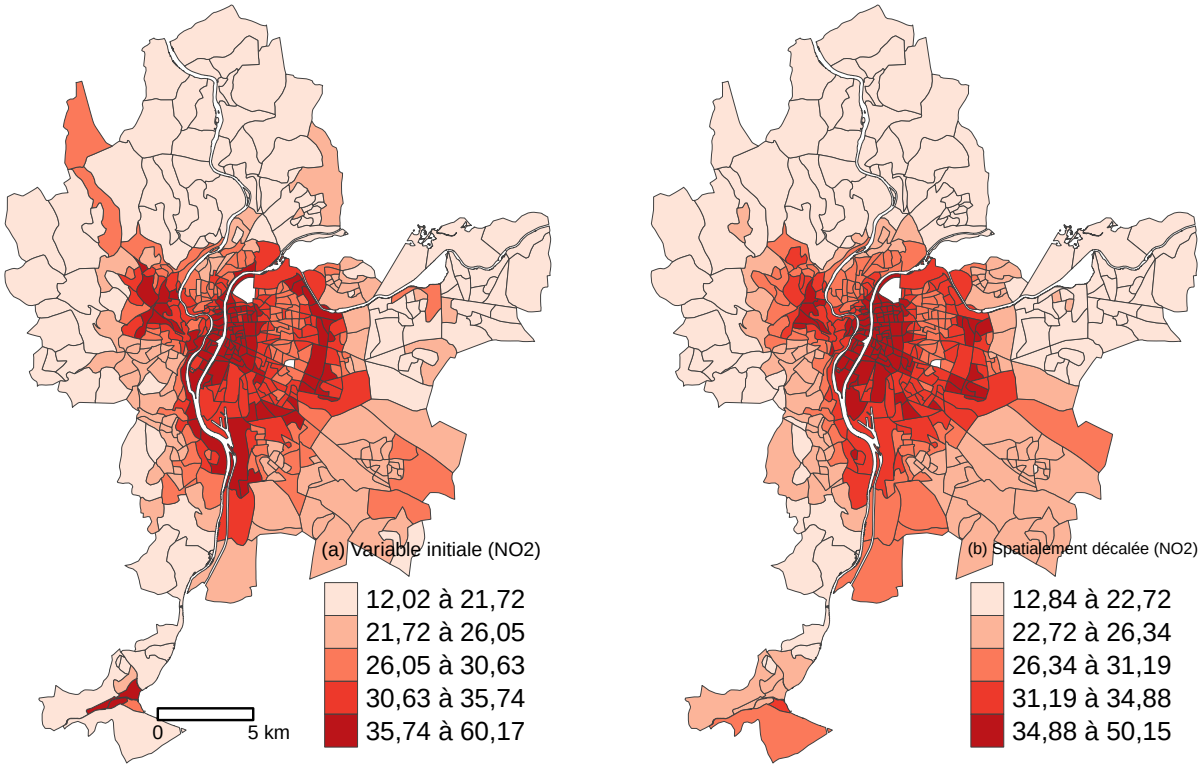


FIGURE 1.11 – Exemple de variable spatialement décalée (dioxyde d'azote)

1.4 Autocorrélation spatiale

L'autocorrélation spatiale permet d'estimer la corrélation d'une variable par rapport à sa localisation dans l'espace. Autrement dit, elle permet de vérifier si les entités proches ou voisines ont tendance à être (dis)semblables en fonction d'un phénomène donné (soit une variable). Nous distinguons trois formes d'autocorrélation spatiale :

- **(a) Autocorrélation spatiale positive** : lorsque les entités spatiales voisines ou proches se ressemblent davantage que celles non contiguës ou éloignées. Cela renvoie ainsi à la première loi de la géographie : « tout interagit avec tout, mais les objets proches ont plus de chance de le faire que les objets éloignés » (traduction libre) (Tobler 1970).
- **(b) Autocorrélation spatiale négative** : lorsque les entités spatiales voisines ou proches ont tendance à être dissemblables, comparativement à celles non contiguës ou éloignées.
- **(c) Absence d'autocorrélation spatiale** : lorsque les valeurs de la variable sont distribuées aléatoirement dans l'espace; autrement dit, lorsqu'il n'y a pas de relation entre le voisinage ou la proximité des entités spatiales et leur degré de ressemblance.

Objectif

Différentes mesures d'autocorrélation spatiale

Nous distinguons habituellement les statistiques d'autocorrélation spatiale globales et locales :

- Les statistiques d'autocorrélation spatiale globales renvoient une valeur pour l'ensemble de l'espace d'étude. Succinctement, elles permettent de vérifier si les entités proches ou voisines d'une couche géographique ont tendance à être (dis)semblables en fonction d'un phénomène donné (soit une variable).
- Les statistiques d'autocorrélation spatiale locales renvoient des valeurs pour chacune des entités spatiales. Succinctement, il s'agit de vérifier si chaque entité spatiale est significativement (dis)semblable de celles voisines ou proches.

Nous aborderons ici uniquement une **statistique d'autocorrélation spatiale globale**, soit le ***I* de Moran**, pour évaluer l'autocorrélation spatiale d'une variable continue. Pour aborder une panoplie de mesures globales ou locales, nous vous invitons à consulter le chapitre intitulé *Autocorrélation spatiale* (Apparicio et Gelb 2024).

1.4.1 Formulation du I de Moran

Pour évaluer le degré d'autocorrélation spatiale d'une variable continue, les deux principales statistiques utilisées sont le I de Moran (1950) et le c de Geary (1954). Nous présentons ici uniquement le I de Moran pour deux raisons principales. Premièrement, étant basée sur la covariance, son interprétation est bien plus facile que celle du c de Geary (basé sur la variance des écarts), c'est-à-dire qu'elle est très similaire au bien connu coefficient de corrélation de Pearson (Apparicio et Gelb 2022). Deuxièmement, elle constitue la mesure la plus utilisée. Le I de Moran s'écrit :

$$I = \frac{n}{\Sigma_{i=1}^{n}\Sigma_{j=1}^{n} w_{ij}} \frac{\Sigma_{i=1}^{n}\Sigma_{j=1}^{n} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2} \text{ avec :} \tag{1.9}$$

- n, le nombre d'entités spatiales dans la couche géographique;
- w_{ij}, la valeur de la pondération spatiale entre les entités spatiales i et j;
- x_i et x_j, les valeurs de la variable continue pour les entités spatiales i et j;
- $\bar{x}$, la valeur moyenne de la variable X à l'étude.

Note

Standardisation de la matrice de pondération et *I* de Moran

Nous avons vu que si la matrice de pondération spatiale est standardisée en ligne (section 1.2.3), alors chaque ligne de la matrice vaut 1 et la somme de l'ensemble des valeurs de la matrice est égale au nombre d'entités spatiales (n). Or, dans l'équation 1.9, $\Sigma_{i=1}^{n}\Sigma_{j=1}^{n} w_{ij}$ représente la somme des pondérations de la matrice, soit n si elles sont standardisées en ligne. Puisque $\frac{n}{n} = 1$, alors l'équation du I de Moran est simplifiée comme suit :

$$I = \frac{\Sigma_{i=1}^{n}\Sigma_{j=1}^{n} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2} \tag{1.10}$$

Comme évoqué dans la section 1.2.3, cela démontre l'intérêt de la standardisation : la comparaison des valeurs du I de Moran obtenues avec différentes matrices de contiguïté afin de sélectionner (éventuellement) celle avec laquelle l'autocorrélation spatiale est la plus forte.

1.4.2 Interprétation du I de Moran

Avec une matrice standardisée, la statistique du I de Moran varie de -1 à 1 et s'interprète de la façon suivante :

- quand $I > 0$, l'autocorrélation est positive, c'est-à-dire que les entités géographiques ont tendance à se ressembler d'autant plus qu'elles sont voisines ou proches;
- quand $I = 0$, l'autocorrélation est nulle, c'est-à-dire que la contiguïté ou la proximité spatiale des zones ne jouent aucun rôle;
- quand $I < 0$, l'autocorrélation est négative, c'est-à-dire que les entités géographiques ont tendance à être dissemblables d'autant plus qu'elles sont voisines ou proches.

Les limites de -1 et 1 sont les maximums théoriques du I de Moran. Dans la pratique, elles sont limitées par la matrice spatiale utilisée dans le calcul. En effet, selon la matrice spatiale, le maximum du I de Moran peut être inférieur à 1, et son minimum supérieur à -1. Le calcul de ces bornes propres à chaque matrice spatiale peut se faire en utilisant les maximums et minimums des valeurs propres de $\frac{W+W^T}{2}$, quand la matrice spatiale est standardisée. À titre d'exemple, nous calculons ci-dessous les maximums et minimums possibles pour une matrice de contiguïté selon le partage d'un nœud (*Queen*) pour les 506 IRIS de l'agglomération de Lyon. Il apparaît ainsi que pour la matrice de contiguïté selon le partage d'un nœud (*Queen*), quelle que soit la variable analysée, la valeur de I de Moran ne pourra pas dépasser les limites -0,64 et 1,04.

```
# Matrice de contiguïté selon le partage d'un nœud (Queen)
nb_queen <- poly2nb(LyonIris, queen = TRUE)
w_queen <- nb2listw(nb_queen, style = 'W')
queen_mat <- listw2mat(w_queen)
values <- range(eigen((queen_mat + t(queen_mat))/2)$values)
print(round(values,2))
```

```
[1] -0.69  1.04
```

1.4.3 Significativité du I de Moran

Comme pour le coefficient de corrélation calculé entre deux variables, il est possible de tester la significativité de la valeur du I de Moran obtenue. Sans que nous détaillions les calculs de significativité, notez qu'il existe trois manières de tester la significativité :

- selon l'hypothèse de la normalité;
- selon l'hypothèse de la randomisation;
- selon des permutations Monte-Carlo (habituellement avec 999 échantillons).

Aller plus loin

Comment calculer les trois tests de significativité du *I* de Moran?
Pour une description détaillée du calcul des trois tests, consultez l'ouvrage de Jean Dubé et Diego Legros (2014).

1.4.4 Mise en œuvre dans R

Objectif

Calcul du *I* de Moran dans R
Pour illustrer le calcul de *I* de Moran dans R, nous utilisons la couche des IRIS de l'agglomération de Lyon. Les étapes suivantes sont réalisées :

1. Construire une panoplie de matrices de pondération spatiale selon la contiguïté, la connectivité, la proximité et le critère des plus proches voisins.
2. Comparer les valeurs de significativité (p) pour une variable continue (`NO2`).
3. Pour cette même variable, trouver avec quelle matrice la valeur du *I* de Moran est la plus forte.

1.4.4.1 Étape 1. Construction des matrices de pondération spatiale

```
library(dplyr)
library(sf)
library(spdep)
load("data/Lyon.Rdata")
# Matrices de contiguité
nb_queen <- poly2nb(LyonIris, queen = TRUE)
nb_rook <- poly2nb(LyonIris, queen = FALSE)
w_queen <- nb2listw(nb_queen, zero.policy = TRUE, style = "W")
w_rook <- nb2listw(nb_rook, zero.policy = TRUE, style = "W")

# Matrices de contiguité (ordre 2 à 5)
w_rook2 <- nblag_cumul(nblag(nb_rook, 2)) %>% nb2listw(zero.policy = TRUE, style = "W")
w_rook3 <- nblag_cumul(nblag(nb_rook, 3)) %>% nb2listw(zero.policy = TRUE, style = "W")
w_rook4 <- nblag_cumul(nblag(nb_rook, 4)) %>% nb2listw(zero.policy = TRUE, style = "W")
w_rook5 <- nblag_cumul(nblag(nb_rook, 5)) %>% nb2listw(zero.policy = TRUE, style = "W")

# Matrices de connectivité binaire
centroides <- st_centroid(LyonIris)
w_connect_2500m <- dnearneigh(centroides, d1 = 0, d2 = 2500) %>%
                   nb2listw(zero.policy = TRUE, style = "W")

# Inverse de la distance et inverse de la distance au carré (matrices complètes)
distances <- as.matrix(dist(st_coordinates(centroides), method = "euclidean"))
diag(distances) <- 0
w_inv_distances <- ifelse(distances!=0, 1/distances, distances) %>% mat2listw(style = "W")
w_inv_distances2 <- ifelse(distances!=0, 1/distances^2, distances) %>% mat2listw(style = "W")

# Inverse de la distance et inverse de la distance au carré (matrices réduites)
coords <- st_coordinates(centroides)
plusprochevoisin_max <- max(unlist(nbdists(knn2nb(knearneigh(coords)),coords)))
```

```
dists <- nbdists(dnearneigh(coords, 0, plusprochevoisin_max), coords)
w_inv_distances_reduite  <- lapply(dists, function(x) (1/x)) %>%
                            nb2listw(voisins_distmax, glist = ., style = "W", zero.policy = TRUE)
w_inv_distances2_reduite <- lapply(dists, function(x) (1/x^2)) %>%
                            nb2listw(voisins_distmax, glist = ., style = "W", zero.policy = TRUE)

# Inverse de la distance et inverse de la distance au carré avec un seuil de distance
w_inv_distances_1000m  <- ifelse(distances<=1000 & distances!=0, 1/distances, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)
w_inv_distances_2500m  <- ifelse(distances<=2500 & distances!=0, 1/distances, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)
w_inv_distances_5000m  <- ifelse(distances<=5000 & distances!=0, 1/distances, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)
w_inv_distances2_2500m <- ifelse(distances<=2500 & distances!=0, 1/distances^2, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)
w_inv_distances2_5000m <- ifelse(distances<=5000 & distances!=0, 1/distances^2, 0) %>%
                          mat2listw(style = "W", zero.policy = TRUE)

## Matrices des plus proches voisins de 2 à 5
w_k2 <- knn2nb(knearneigh(coords, k = 2)) %>% nb2listw(zero.policy = FALSE, style = "W")
w_k3 <- knn2nb(knearneigh(coords, k = 3)) %>% nb2listw(zero.policy = FALSE, style = "W")
w_k4 <- knn2nb(knearneigh(coords, k = 4)) %>% nb2listw(zero.policy = FALSE, style = "W")
w_k5 <- knn2nb(knearneigh(coords, k = 5)) %>% nb2listw(zero.policy = FALSE, style = "W")
```

1.4.4.2 Étape 2. Calcul du I de Moran et des trois tests de significativité

Les fonctions `moran.test` et `moran.mc` du *package* `spdep` permettent de calculer le I de Moran selon les trois façons de tester la significativité :

- selon l'hypothèse de la normalité avec le paramètre `randomisation = FALSE`
  - `moran.test(ObjetSf$Variable, listw = MatriceW, zero.policy = TRUE, randomisation = FALSE)`
- selon l'hypothèse de la randomisation avec le paramètre `randomisation = TRUE`
  - `moran.test(ObjetSf$Variable, listw = MatriceW, zero.policy = TRUE, randomisation = TRUE)`
- selon des permutations Monte-Carlo avec le paramètre `nsim` (ci-dessous avec 999 permutations)
  - `moran.mc(ObjetSf$Variable, listw=MatriceW, zero.policy = TRUE, nsim = 999)`

Bien entendu, dans les sorties des trois méthodes, la valeur du I de Moran est la même, contrairement à la valeur de p qui peut varier.

```
moran.test(LyonIris$NO2,          # nom de l'objet sf et de la variable continue
           listw=w_queen,         # nom de la matrice de pondération spatiale
           zero.policy = TRUE,
           randomisation = FALSE) # significativité selon l'hypothèse de la normalité
```

```
    Moran I test under normality

data:  LyonIris$NO2
weights: w_queen

Moran I statistic standard deviate = 30.135, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.8189249687     -0.0019801980      0.0007420591
```

```
moran.test(LyonIris$NO2,          # nom de l'objet sf et de la variable continue
           listw=w_queen,         # nom de la matrice de pondération spatiale
           zero.policy = TRUE,
           randomisation = TRUE)  # significativité selon l'hypothèse de la randomisation
```

```
    Moran I test under randomisation

data:  LyonIris$NO2
weights: w_queen

Moran I statistic standard deviate = 30.135, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.8189249687     -0.0019801980      0.0007420917
```

```
moran.mc(LyonIris$NO2,            # nom de l'objet sf et de la variable continue
         listw=w_queen,           # nom de la matrice de pondération spatiale
         zero.policy = TRUE,
         nsim=999)                # 999 permutations
```

```
    Monte-Carlo simulation of Moran I

data:  LyonIris$NO2
weights: w_queen
number of simulations + 1: 1000

statistic = 0.81892, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

La statistique du I de Moran ($I = 0{,}82$, $p < 0{,}001$) indique que la variable *dioxyde d'azote* a une forte autocorrélation spatiale positive.

1.4.4.3 Étape 3. Identification de la plus forte autocorrélation spatiale selon les différentes matrices

La syntaxe ci-dessous permet de calculer la statistique du I de Moran avec plusieurs matrices de pondération spatiale.

```
## Création d'une liste pour toutes les matrices
liste_matrices <- list(w_queen = w_queen,
                       w_rook= w_rook,
                       w_rook2 = w_rook2,
                       w_rook3 = w_rook3,
                       w_rook4 = w_rook4,
                       w_rook5 = w_rook5,
                       w_connect_2500m = w_connect_2500m,
                       w_inv_distances = w_inv_distances,
                       w_inv_distances2 = w_inv_distances2,
                       w_inv_distances_reduite = w_inv_distances_reduite,
                       w_inv_distances2_reduite = w_inv_distances2_reduite,
                       w_inv_distances_1000m = w_inv_distances_1000m,
                       w_inv_distances_2500m = w_inv_distances_2500m,
                       w_inv_distances2_2500m = w_inv_distances2_2500m,
                       w_inv_distances2_5000m = w_inv_distances2_5000m,
                       w_k2 = w_k2,
                       w_k3 = w_k3,
                       w_k4 = w_k4,
                       w_k5 = w_k5)

## Vecteur pour le I de Moran et la valeur de p
moranI <- c()
pvalue <- c()
i<-0

## Boucle pour calculer le I de Moran avec la liste des matrices
for (e in liste_matrices){
   i<-i+1
   test <-moran.mc(LyonIris$NO2,
                   listw=e,
                   zero.policy = TRUE,
                   nsim=999)
   moranI[i] <- test$statistic
   pvalue[i] <- test$p.value
}

# Création d'un DataFrame avec les valeurs du I de Moran et de p
moran_resultats <- data.frame(Matrices = names(liste_matrices),
                              MoranIs = round(moranI, 4),
                              Pvalues = pvalue)
print(moran_resultats)
```

```
                  Matrices MoranIs Pvalues
1                   w_queen  0.8189   0.001
2                    w_rook  0.8244   0.001
3                   w_rook2  0.7174   0.001
4                   w_rook3  0.6215   0.001
5                   w_rook4  0.5192   0.001
6                   w_rook5  0.4194   0.001
7             w_connect_2500m  0.6327   0.001
8             w_inv_distances  0.1946   0.001
9            w_inv_distances2  0.4769   0.001
10  w_inv_distances_reduite  0.6301   0.001
11 w_inv_distances2_reduite  0.7011   0.001
12    w_inv_distances_1000m  0.6302   0.001
13    w_inv_distances_2500m  0.6766   0.001
14   w_inv_distances2_2500m  0.7235   0.001
15   w_inv_distances2_5000m  0.6554   0.001
16                     w_k2  0.8083   0.001
17                     w_k3  0.7903   0.001
18                     w_k4  0.7865   0.001
19                     w_k5  0.7844   0.001
```

La lecture détaillée du tableau 1.6 permet d'avancer plusieurs constats intéressants :

- D'emblée, signalons que toutes les valeurs du I de Moran sont positives et significatives, témoignant d'une autocorrélation spatiale positive.
- Concernant les **matrices de contiguïté**, l'autocorrélation spatiale est plus forte selon le partage d'un segment que d'un nœud (0,8244 contre 0,8189). Par conséquent, si nous devons choisir une matrice de contiguïté, il serait préférable d'utiliser celle définie selon le partage d'un segment (*Rook*).
- Sans surprise, plus nous ajoutons des **ordres d'adjacence**, plus la valeur de la statistique du I de Moran est faible, passant de 0,7174 à 0,4194 du deuxième au cinquième ordre.
- Concernant les **matrices de proximité**, la méthode de l'inverse de la distance au carré, qui accorde un poids plus important aux entités spatiales très proches (comparativement à l'inverse de la distance), renvoie des valeurs toujours plus élevées, et ce, que la matrice soit complète ou réduite. Aussi, les matrices de distance réduites présentent toujours des valeurs plus fortes que les matrices complètes.
- Concernant les matrices **selon le critère des plus proches voisins**, l'autocorrélation spatiale diminue légèrement de $k = 2$ à $k = 5$. D'ailleurs, la valeur la plus forte est pour deux voisins ($I = 0{,}8083$).

Tableau 1.6 – Résultats du I de Moran selon les différentes matrices

Nom	Description	I de Moran	p (999 permutations)
Matrices de contiguïté			
w_queen	Partage d'un nœud	0,8189	0,001
w_rook	Partage d'un segment	0,8244	0,001
Matrices de contiguïté selon le partage d'un segment et ordre d'adjacence			
w_rook2	Ordre 2	0,7174	0,001
w_rook3	Ordre 3	0,6215	0,001

w_rook4	Ordre 4	0,5192	0,001
w_rook5	Ordre 5	0,4194	0,001
Matrice de connectivité			
w_connect_2500m	2500 mètres	0,6327	0,001
Matrices de distance (complètes)			
w_inv_distances	Inverse de la distance	0,1946	0,001
w_inv_distances2	Inverse de la distance au carré	0,4769	0,001
Matrices de distance (réduites)			
w_inv_distances_reduite	Inverse de la distance	0,6301	0,001
w_inv_distances2_reduite	Inverse de la distance au carré	0,7011	0,001
Matrices de distance avec un seuil maximal			
w_inv_distances_1000m	Inverse de la distance (2500 mètres)	0,6302	0,001
w_inv_distances_2500m	Inverse de la distance (5000 mètres)	0,6766	0,001
w_inv_distances2_2500m	Inverse de la distance au carré (2500 mètres)	0,7235	0,001
w_inv_distances2_5000m	Inverse de la distance au carré (5000 mètres)	0,6554	0,001
Matrices selon le critère des plus proches voisins			
w_k2	2 voisins	0,8083	0,001
w_k3	3 voisins	0,7903	0,001
w_k4	4 voisins	0,7865	0,001
w_k5	5 voisins	0,7844	0,001

Quelle est la matrice de pondération spatiale avec laquelle la dépendance spatiale de la variable est la plus forte?

Pour la trouver, nous construisons un graphique avec les valeurs du I de Moran triées par ordre décroissant. La valeur la plus forte est obtenue avec la matrice de contiguïté selon le partage d'un segment.

```
library(ggplot2)
ggplot(data=moran_resultats, aes(x=reorder(Matrices,MoranIs), y=MoranIs)) +
  geom_segment( aes(x=reorder(Matrices,MoranIs),
                    xend=reorder(Matrices,MoranIs),
                    y=0, yend=MoranIs)) +
  geom_point( size=4,fill="red",shape=21)+
  xlab("Matrice de pondération spatiale") +
  ylab("I de Moran")+
  coord_flip()
```

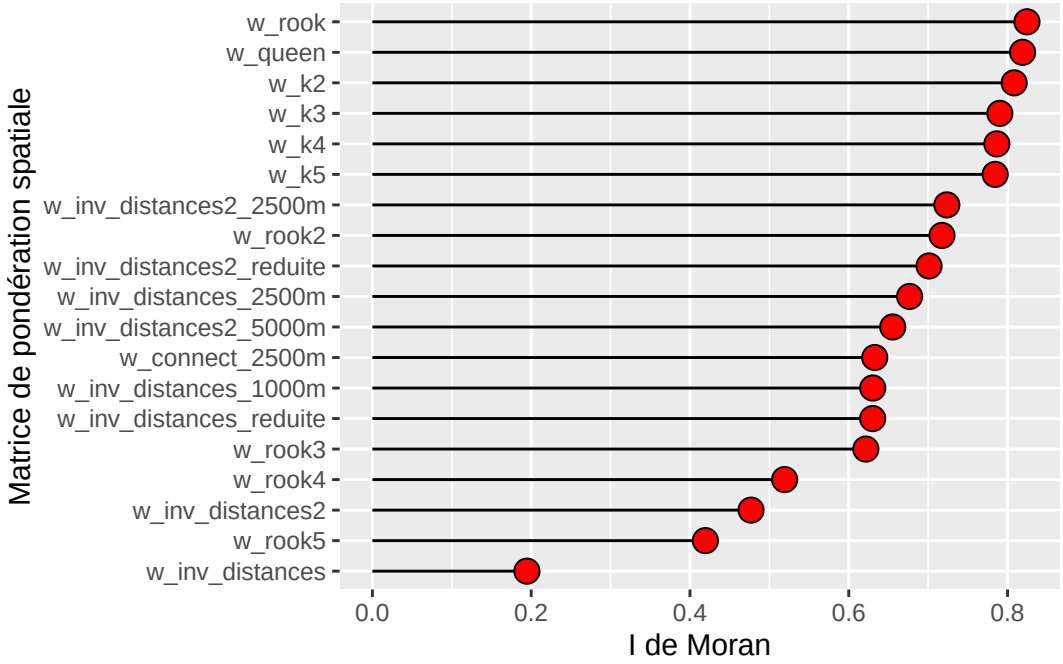


FIGURE 1.12 – Valeurs du *I* de Moran selon les différentes matrices de pondération spatiale

1.4.4.4 Étape 4. Comparaison des valeurs du I de Moran pour plusieurs variables avec la même matrice

La syntaxe ci-dessous permet de calculer la statistique du *I* de Moran pour plusieurs variables avec la même matrice de pondération spatiale (ici, matrice de contiguïté selon le partage d'un segment).

```
## Vecteur pour les variables à analyser
liste_vars <- c("Lden", "NO2", "PM25", "VegHautPrt",
                "Pct0_14", "Pct_65", "Pct_Img", "TxChom1564", "Pct_brevet", "NivVieMed")

## Boucle pour calculer le I de Moran pour les différentes variables
moran_resultats2 <- t(sapply(liste_vars, function(e){
test <- moran.mc(LyonIris[[e]],
                 listw=w_rook,
                 zero.policy = TRUE,
                 nsim=999)
result <- c(round(test$statistic,4), test$p.value)
}))

moran_resultats2 <- data.frame(moran_resultats2)
names(moran_resultats2) <- c('MoranIs', 'Pvalues')
moran_resultats2$Variable <- liste_vars
rownames(moran_resultats2) <- NULL
print(moran_resultats2)
```

```
   MoranIs Pvalues   Variable
```

```
1   0.5677   0.001       Lden
2   0.8244   0.001        NO2
3   0.9371   0.001       PM25
4   0.5869   0.001 VegHautPrt
5   0.4421   0.001    Pct0_14
6   0.3013   0.001     Pct_65
7   0.4708   0.001    Pct_Img
8   0.2944   0.001 TxChom1564
9   0.5323   0.001 Pct_brevet
10  0.6286   0.001  NivVieMed
```

De nouveau, en quelques lignes de code, il est aisé de réaliser un graphique pour comparer les valeurs du I de Moran pour les différentes variables (figure 1.13).

```
library(ggplot2)

# Construction d'un graphique avec les résultats I de Moran pour les différentes variables
ggplot(data=moran_resultats2, aes(x=reorder(Variable,MoranIs), y=MoranIs)) +
    geom_segment(aes(x = reorder(Variable,MoranIs),
                     xend = reorder(Variable,MoranIs),
                     y = 0 ,
                     yend=  MoranIs)) +
    geom_point( size = 4, fill = "red",shape = 21)+
    xlab("Variable continue") + ylab("I de Moran")+
    coord_flip()
```

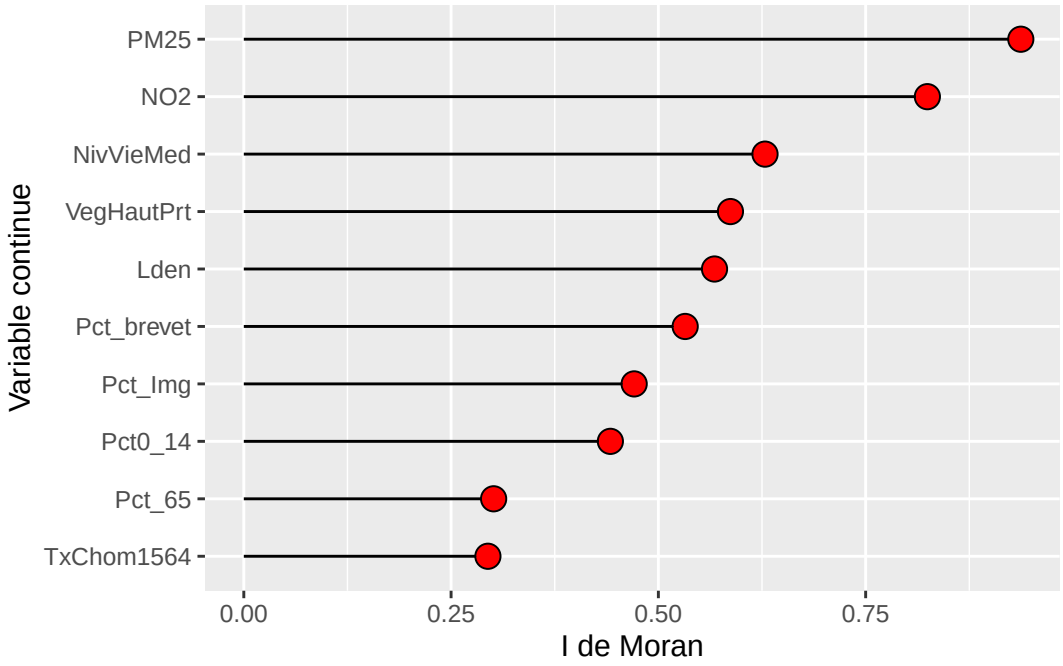


Figure 1.13 – Valeurs du I de Moran pour les dix variables

1.5 Bref retour sur la régression linéaire multiple

À titre de rappel, la régression linéaire multiple permet d'exprimer une variable d'intérêt, habituellement notée par le terme variable dépendante (y) en fonction de plusieurs variables que l'on pense liées, théoriquement, à la variable d'intérêt. Celles-ci sont habituellement désignées comme des variables explicatives ou indépendantes (notées x_k).

L'équation de régression s'écrit alors :

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_K x_{Ki} + \varepsilon_i \tag{1.11}$$

Où :

- y_i, est la valeur de la variable dépendante y pour l'observation spatiale i.
- x_{ki}, est la valeur de la variable indépendante x_k pour l'observation spatiale i.
- α, est le paramètre d'ordonnée à l'origine, ou encore la constante. Elle représente la valeur moyenne de la variable dépendante lorsque l'ensemble des variables indépendantes (ou explicatives) sont nulles.
- $k = 1, 2, \ldots, K$ est le nombre de variables indépendantes.
- β_1 à β_K, sont les coefficients de régression liés à chacune des variables indépendantes.
- ε_i, est le terme d'erreur, soit la partie de la valeur de la variable dépendante qui n'est pas expliquée par le modèle de régression. Le terme d'erreur est habituellement supposé de moyenne nulle, de variance homogène et indépendant.

À noter que le terme d'erreur est, par définition, inobservable. En revanche, il est possible d'approximer cette valeur en calculant le résidu, $\hat{\varepsilon}_i$, parfois aussi noté $\hat{e}_i$ (équations 1.12 et 1.13).

$$\hat{\varepsilon}_i = \hat{y}_i - y_i \tag{1.12}$$

$$\hat{\varepsilon}_i = y_i - \left[\hat{\alpha} + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \cdots + \hat{\beta}_K x_{Ki}\right] \tag{1.13}$$

Il existe plusieurs manières d'exprimer cette relation sous forme plus compacte. L'une de celles-ci consiste à introduire le terme de sommation dans l'équation :

$$y_i = \alpha + \sum_{k=1}^{K} \beta_k x_{ki} + \varepsilon_i \tag{1.14}$$

Une autre approche, plus pratique et compacte, consiste à utiliser la notation matricielle :

$$y = \iota\alpha + \mathbf{X}\beta + \varepsilon \tag{1.15}$$

Où :

- y, est un vecteur de dimension $(N \times 1)$ où N est le nombre total d'observations mobilisées dans l'analyse, avec $i = 1, 2, \ldots, N$.
- ι est un vecteur de dimension $(N \times 1)$ comportant uniquement des valeurs de 1.
- $\mathbf{X}$, une matrice de dimension $(N \times K)$ renfermant l'ensemble des K vecteurs de variables explicatives en plus d'une colonne de 1, dans la première colonne de la matrice, pour la constante, d'où $(K + 1)$.
- α est un scalaire identifiant l'ordonnée à l'origine.

- β, est un vecteur de dimension $(K+1)$ qui renferme l'ensemble des coefficients de régression, pour les K variables indépendantes et la constante.
- ε, un vecteur de dimension $(N \times 1)$ des termes d'erreurs.

De manière plus explicite, les différents vecteurs et matrices prennent les formes suivantes :

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_N \end{bmatrix} \tag{1.16}$$

$$\iota = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{1.17}$$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1k} & \cdots & x_{1K} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2k} & \cdots & x_{2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & x_{i3} & \cdots & x_{ik} & \cdots & x_{iK} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & x_{N3} & \cdots & x_{Nk} & \cdots & x_{NK} \end{bmatrix} \tag{1.18}$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \\ \vdots \\ \beta_K \end{bmatrix} \tag{1.19}$$

Les modèles de régression linéaire sont largement mobilisés lorsqu'on s'intéresse à l'effet d'une variable spécifique sur le comportement de la variable d'intérêt. Le terme « effet marginal » est souvent associé aux coefficients estimés, β_k. Les coefficients permettent de simuler l'effet d'une variation d'une variable dépendante, Δx_k, sur la variation du comportement de la variable expliquée, Δy.

En termes plus techniques, les effets marginaux permettent d'exprimer l'effet de la dérivée partielle, c'est-à-dire d'isoler comment la variable d'intérêt bouge lorsqu'une des variables indépendantes bouge :

$$\frac{\partial y}{\partial x_k} = \beta_k \tag{1.20}$$

Ces effets sont habituellement l'essence de l'intérêt des modèles de régression. Par exemple, une variation de deux unités de la variable x_k devrait entraîner une variation dans y de $2\beta_k$ unités, toutes choses étant égales par ailleurs. Cette analyse est souvent le centre d'intérêt des modèles de régression, bien que l'interprétation causale ne soit pas nécessairement directe (Dubé et al. 2024).

Aller plus loin

Régression linéaire multiple
Afin de bien maîtriser les principes de base et les hypothèses de la régression linéaire multiple, de mesurer la qualité d'ajustement du modèle, d'introduire des variables explicatives particulières (variable qualitative dichotomique ou polytomique, variable d'interaction, etc.), d'interpréter les résultats d'un modèle de régression et de le mettre en œuvre dans R, nous vous invitons à lire le chapitre intitulé Régression linéaire multiple (Apparicio et Gelb 2022).

1.6 Pourquoi recourir à des régressions spatiales?

Deux motivations principales peuvent conduire à réaliser des régressions spatiales : la **dépendance spatiale** ou l'**hétérogénéité** du modèle de régression classique appliqué à des données spatiales (Chi et Zhu 2019, 41-43).

1.6.1 Dépendance spatiale

Dans un modèle, les résidus (ϵ) sont la différence entre les valeurs observées (y_i) et les valeurs prédites par le modèle ($\widehat{y_i}$). Une des hypothèses de la régression linéaire multiple est que les observations doivent être indépendantes les unes des autres (*indépendance du terme d'erreur*). Le non-respect de cette hypothèse produit des résultats biaisés, notamment pour les coefficients de régression. Un simple test d'autocorrélation spatiale entre les résidus permet de vérifier si cette hypothèse est respectée (section 1.4).

Le rejet de cette hypothèse d'indépendance rend les coefficients estimés, selon la forme fonctionnelle postulée (ou processus générateur des données – PGD) par le modèle, biaisés et/ou biaise le calcul des écarts-types, invalidant ainsi les tests de significativité usuels. La présence d'autocorrélation spatiale entre les résidus est donc un premier test nécessaire pour s'assurer de la qualité des résultats issus du modèle de régression linéaire.

À noter qu'une autocorrélation spatiale entre certaines variables observables, dépendantes ou indépendantes est possible sans pour autant que les résidus soient autocorrélés. Dans ce cas, les relations entre les variables indépendantes permettent de capter l'effet spatial contenu dans la variable dépendante et les résultats d'estimations sont valides si les résidus sont homoscédastiques, c'est-à-dire de variance homogène (autrement, le problème peut facilement être réglé par un simple ajout d'option dans la commande du modèle de régression). Ce n'est donc pas sur les variables observables que le test de détection d'autocorrélation spatiale est important, mais bien sur les résidus du modèle.

Pour être valides, les résultats d'un modèle de régression linéaire construit avec des données spatiales ne devraient pas avoir des résidus spatialement autocorrélés. Si c'est le cas, alors certaines options sont envisageables pour tenter de remédier à la situation. La plus simple étant l'ajout de variables indépendantes spatialement structurées, afin de capter l'autocorrélation résiduelle. Les méthodes plus avancées proposent d'intégrer formellement cette relation spatiale dans le PGD.

Pour illustrer le problème de dépendance spatiale d'un modèle, nous calculons le modèle MCO suivant avec le jeu de données des IRIS de l'agglomération lyonnaise : `lm(PM25 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed, data = LyonIris)`.

```
load("data/Lyon.Rdata")
## Modèle MCO
modele_MCO <- lm(PM25 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed, data = LyonIris)
summary(modele_MCO)
```

```
Call:
lm(formula = PM25 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
    NivVieMed, data = LyonIris)

Residuals:
    Min      1Q  Median      3Q     Max
-6.8510 -1.1962 -0.0094  1.2337  7.5446

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 23.382778   0.775600  30.148  < 2e-16 ***
Pct0_14     -0.151750   0.016326  -9.295  < 2e-16 ***
Pct_65      -0.049786   0.014571  -3.417 0.000685 ***
Pct_Img      0.088013   0.013240   6.648 7.82e-11 ***
Pct_brevet  -0.076002   0.009635  -7.888 1.94e-14 ***
NivVieMed   -0.113065   0.026357  -4.290 2.15e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.731 on 500 degrees of freedom
Multiple R-squared:  0.3539,    Adjusted R-squared:  0.3474
F-statistic: 54.78 on 5 and 500 DF,  p-value: < 2.2e-16
```

Pour vérifier la dépendance spatiale du modèle, nous calculons le I de Moran sur les résidus avec la fonction `lm.morantest` du *package* `spdep` et nous cartographions les résidus avec le *package* `tmap`. La statistique du I de Moran sur les résidus ($I = 0{,}66$, $p < 0{,}001$) indique clairement que le modèle MCO a un problème de dépendance spatiale et que ses coefficients de régression reportés plus haut sont certainement biaisés. Aussi, la cartographie des coefficients de régression montre des résidus positifs dans les IRIS des quartiers centraux, tandis que les résidus négatifs sont concentrés dans les IRIS des quartiers ou des municipalités périphériques (figure 1.14). Par conséquent, le recours à des régressions spatiales est certainement nécessaire pour remédier à ce problème de dépendance spatiale du modèle MCO.

```
library(spdep)
library(tmap)
## Modèle MCO
LyonIris$MCO.Residus <- modele_MCO$residuals

## Matrice de contiguïté
nb_rook <- poly2nb(LyonIris, queen = TRUE)
w_rook<- nb2listw(nb_rook, zero.policy = TRUE, style = "W")
```

```r
# I de Moran sur les résidus du modèle global (MCO)
lm.morantest(modele_MCO, w_rook)
```

```
    Global Moran I for regression residuals

data:
model: lm(formula = PM25 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
NivVieMed, data = LyonIris)
weights: w_rook

Moran I statistic standard deviate = 24.453, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Observed Moran I      Expectation         Variance
    0.6552777081    -0.0053724124     0.0007299083
```

```r
# Cartographie des résidus
tmap_mode("plot")
tm_shape(LyonIris)+
         tm_borders(col = "gray25", lwd=.5)+
         tm_fill(col = "MCO.Residus",
                 n = 6,
                 style = "pretty",
                 legend.format = list(text.separator = "à",
                                      decimal.mark = ","),
                 midpoint = 0,
                 palette = "-RdBu",
                 title = "MCO") +
         tm_layout(frame = FALSE)+
         tm_scale_bar(breaks = c(0,5))
```

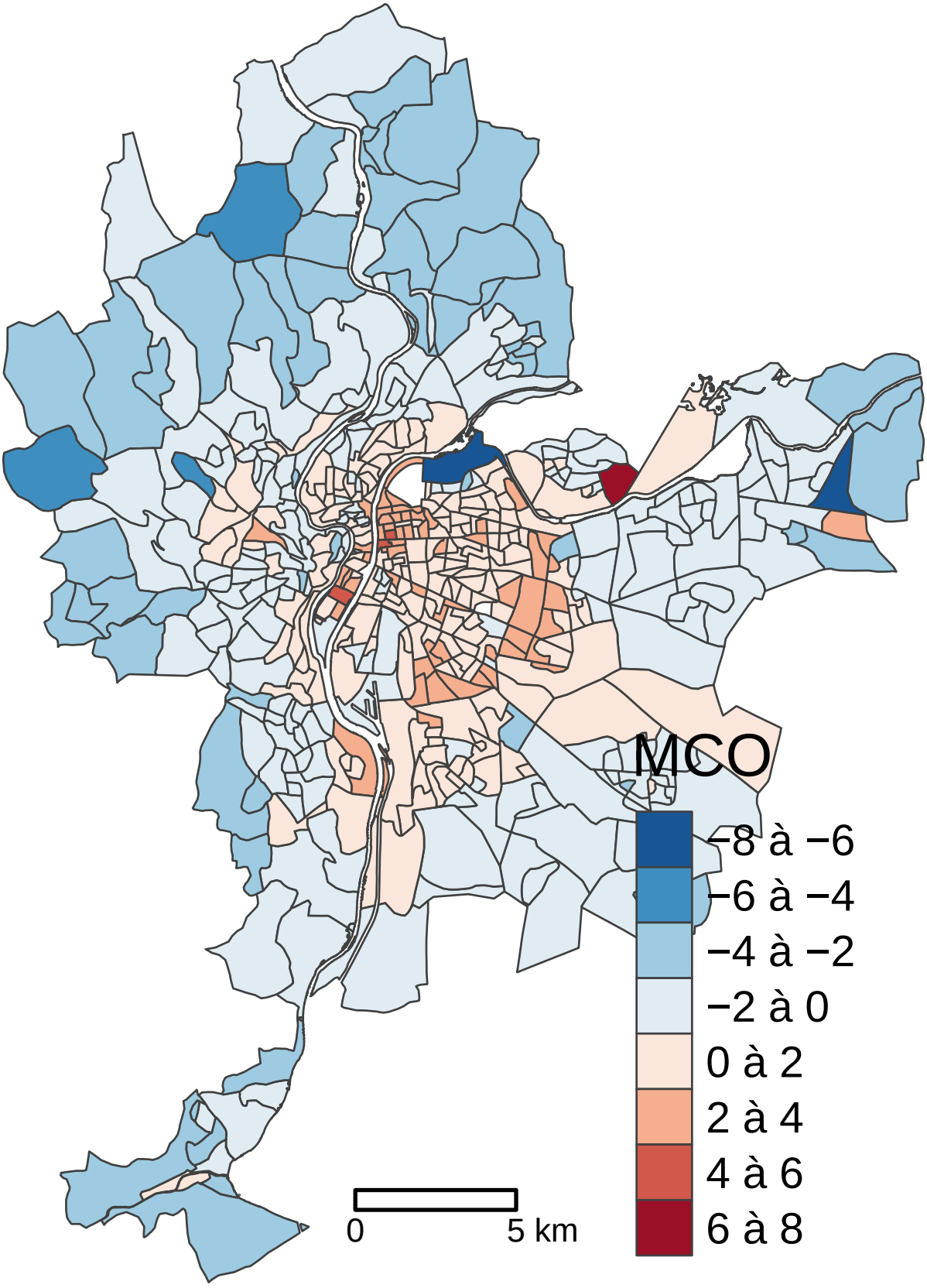


Figure 1.14 – Cartographie des résidus d'un modèle MCO

Aller plus loin

Dépendance spatiale et méthodes de régression spatiale : quelles solutions?
Les **modèles des moindres carrés généralisés** (GLS) et les **modèles linéaires généralisés à effets mixtes** (GLMM) permettent d'ajuster la corrélation spatiale sur les termes d'erreur.
Le **modèle d'autorégression conditionnelle** (CAR) permet de tenir compte de la dépendance locale de la variable dépendante (chapitre 2).
Les modèles d'économétrie spatiale (chapitre 3, chapitre 4 et chapitre 5) sont des solutions particulièrement intéressantes pour remédier au problème de dépendance spatiale d'un modèle et obtenir des coefficients de régression non biaisés, et ce, en introduisant l'autocorrélation spatiale de différentes façons : sur les variables indépendantes, sur la variable dépendante, sur le terme d'erreur, etc.).
Les modèles linéaires généralisés (GAM) (chapitre 6) permettent d'introduire des lissages spatiaux pour capturer les tendances spatiales soit de manière continue (avec une spline bivariée sur les coordonnées géographiques), soit de manière discrète (avec un lissage par champ aléatoire de Markov). Aussi, les **modèles linéaires généralisés avec des vecteurs spatiaux** (chapitre 7) permettent de capturer les tendances spatiales en résumant la matrice de pondération spatiale W.

1.6.2 Hétérogénéité spatiale

L'hétérogénéité ou la non-stationnarité (***spatial heterogeneity*** ou ***nonstationarity***) se manifeste quand les relations entre la variable dépendante et les variables indépendantes varient dans l'espace (en d'autres termes à travers les entités spatiales du jeu de données). Cela contrevient ainsi à l'hypothèse de l'homogénéité spatiale ou de stationnarité spatiale selon laquelle les coefficients du modèle doivent être constant dans l'espace.

Aller plus loin

Hétérogénéité ou non-stationnarité spatiale et méthodes de régression spatiale : quelles solutions?
Les régressions géographiquement pondérées (chapitre 8 et chapitre 9), sont des outils d'exploration particulièrement adaptés pour cartographier et analyser l'instabilité des relations entre la variable dépendante et les variables indépendantes d'un modèle de régression.
Les **modèles généralisés additifs** (GAM) et les **modèles généralisés à effet mixtes** (GLMM) permmettent aussi d'intégrer des coefficients variant spatialement (chapitre 9)

1.7 Quiz de révision

Questions

- **Parmi les matrices de pondération spatiale ci-dessous, lesquelles sont des matrices de contiguïté?**
  - Partage d'un nœud
  - Partage d'un segment
  - Partage d'un nœud et ordre d'adjacence
  - Partage d'un segment et ordre d'adjacence
  - Connectivité selon la distance

  Relisez au besoin la section 1.2.

— **En anglais, comment est appelée une matrice selon le partage d'un nœud?**

  — Rook
  — Queen

  Relisez au besoin le début de la section 1.2.

— **Comparativement à une matrice de l'inverse de la distance, une matrice de l'inverse de la distance au carré accorde un poids plus important aux entités proches.**

  — Vrai
  — Faux

  Relisez au besoin la section 1.2.2.2.

— **Quels sont les avantages de la standardisation en ligne des matrices de pondération spatiale?**

  — La somme de chaque ligne est égale à 1.
  — La somme de l'ensemble de la matrice est égale au nombre d'entités spatiales.
  — La standardisation permet de comparer la dépendance spatiale selon différentes matrices.
  — La standardisation augmente la vitesse des calculs.

  Relisez au besoin la section 1.2.3.

— **Avec une matrice standardisée, la statistique du I de Moran varie théoriquement de :**

  — de -1 à 1
  — de 0 à 100
  — de moins l'infini à plus l'infini

  Relisez au besoin la section 1.4.2.

— **Quelles sont les trois manières de tester la significativité du I de Moran?**

  — Avec l'hypothèse de la normalité
  — En relançant plusieurs fois les calculs
  — Avec l'hypothèse de la randomisation
  — Avec la méthode Monte-Carlo (habituellement avec 999 échantillons)

  Relisez au besoin la section 1.4.3.

— **Comment mesurer la dépendance spatiale d'un modèle MCO?**

  — En analysant le R2
  — En calculant le I de Moran sur les résidus du modèle
  — En simulant des variables spatiales

  Relisez au besoin la section 1.6.1.

— **Quel est le synonyme de l'hétérogénéité spatiale?**

  — Non-stationnarité
  — Variation spatiale
  — Corrélation spatiale

  Relisez au besoin la section 1.6.2.

Réponses

— Parmi les matrices de pondération spatiale ci-dessous, lesquelles sont des matrices de contiguïté?
  — Partage d'un nœud

  - Partage d'un segment
  - Partage d'un nœud et ordre d'adjacence
  - Partage d'un segment et ordre d'adjacence
- En anglais, comment est appelée une matrice selon le partage d'un nœud?
  - Queen
- Comparativement à une matrice de l'inverse de la distance, une matrice de l'inverse de la distance au carré accorde un poids plus important aux entités proches.
  - Vrai
- Quels sont les avantages de la standardisation en ligne des matrices de pondération spatiale?
  - La somme de chaque ligne est égale à 1.
  - La somme de l'ensemble de la matrice est égale au nombre d'entités spatiales.
  - La standardisation permet de comparer la dépendance spatiale selon différentes matrices.
- Avec une matrice standardisée, la statistique du I de Moran varie théoriquement de :
  - de -1 à 1
- Quelles sont les trois manières de tester la significativité du I de Moran?
  - Avec l'hypothèse de la normalité
  - Avec l'hypothèse de la randomisation
  - Avec la méthode Monte-Carlo (habituellement avec 999 échantillons)
- Comment mesurer la dépendance spatiale d'un modèle MCO?
  - En calculant le I de Moran sur les résidus du modèle
- Quel est le synonyme de l'hétérogénéité spatiale?
  - Non-stationnarité

1.8 Exercices de révision

Exercice

Exercice 2. Matrice de contiguïté selon le partage d'un segment et le I de Moran

```
load("data/Lyon.Rdata")
library(spdep)
## Matrice de contiguïté selon le partage d'un nœud (Queen)
à compléter
# I de Moran sur la variable Lden selon l'hypothèse de la normalité
à compléter)
# I de Moran sur la variable Lden selon la normalité selon l'hypothèse de la randomisation
à compléter
# I de Moran sur la variable Lden selon des permutations Monte-Carlo
à compléter
```

Correction à la section 11.1.1.

Exercice

Exercice 2. Modèle MCO et dépendance spatiale

```
load("data/Lyon.Rdata")
library(spdep)
# Modèle MCO
formule <- "Lden ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed"
à compléter
# Résultats du modèle
à compléter
## Matrice de contiguïté (Rook)
à compléter

# I de Moran sur les résidus du modèle global (MCO)
à compléter

# Cartographie des résidus
à compléter
```

Correction à la section 11.1.2.

Partie 2. Spécification de la structure de covariance spatiale

2 Modèles intégrant une structure de covariance spatiale (en cours de rédaction)

2.1 Modèles des moindres carrés généralisés (GLS)

2.2 Modèle d'autorégression conditionnelle (CAR)

2.3 Modèles linéaires généralisés à effets mixtes (GLMM)

2.4 Quiz de révision

2.5 Exercices de révision

Exercice

Exercice 1. À compléter
Complétez le code ci-dessous.
Correction à la section 11.2.1.

Exercice

Exercice 2. À compléter
Complétez le code ci-dessous.
Correction à la section 11.2.2.

Exercice

Exercice 3. À compléter
Complétez le code ci-dessous.
Correction à la section 11.2.3.

Partie 3. Économétrie spatiale

3 Modèles d'économétrie spatiale

Dans ce chapitre, nous décrivons uniquement les modèles économétriques spatiaux dont la variable dépendante est continue. Sommairement, ces modèles sont des extensions de la régression linéaire multiple, mais dans lequel une spécification explicite des relations spatiales entre les observations est introduite. Cette spécification vise ultimement à capter des effets spatiaux qui influencent le processus générateur de données (PGD), c'est-à-dire le modèle économétrique que nous spécifions.

Objectif

Objectifs d'apprentissage visés dans ce chapitre
À la fin de ce chapitre, vous devriez être en mesure de :

- saisir les raisons motivant le choix d'un modèle de régression (externalités, effets d'entraînement, effets mixtes, etc.);
- comprendre les différents modèles spatiaux autorégressifs (SLX, SAR, SEM, SDM, SDEM, Manski);
- utiliser une stratégie pour choisir un modèle (tests du multiplicateur de Lagrange, tests du ratio de vraisemblance, critère d'information d'Akaike);
- vérifier si le modèle autorégressif a corrigé ou non la dépendance spatiale du modèle MCO;
- mettre en pratique ces modèles spatiaux dans R.

Package

Liste des *packages* utilisés dans ce chapitre

- Pour importer et manipuler des fichiers géographiques :
  - `sf` pour importer et manipuler des données vectorielles.
- Pour construire des cartes et des graphiques :
  - `tmap` est certainement le meilleur *package* pour la cartographie.
  - `ggplot2` et `ggpubr` pour construire des graphiques.
- Pour construire des modèles spatiaux :
  - `spdep` pour construire des matrices de pondération spatiales et calculer le I de Moran.
  - `spatialreg` pour construire des modèles économétriques spatiaux.

Attention

Modèles d'économétrie spatiale pour une variable dépendante continue
Avant de lire ce chapitre, il convient de bien maîtriser la régression linéaire multiple. Une brève récapitulation est proposée à la section 1.5.

3.1 Les différents modèles spatiaux autorégressifs

Selon Jean Dubé et Diègo Legros, quelques raisons expliquent « le choix d'un modèle autorégressif : la présence d'externalités, les effets d'entraînement, l'omission de variables importantes, la présence d'hétérogénéité spatiale des comportements, les effets mixtes » (2014, 120). Les effets mixtes représentent une combinaison d'effets spatiaux. Dans tous les cas, la prise en compte des effets spatiaux passe par la spécification d'une matrice de pondération spatiale, notée W et de dimension $(N \times N)$. Une synthèse des différents modèles spatiaux autorégressifs, décrits dans les sections suivantes, est présentée au tableau 3.1, indiquant s'ils sont ou non mixtes, et le type d'effet qu'ils permettent de prendre en compte.

Tableau 3.1 – Synthèse des différents modèles d'économétrie spatiale

	SLX	SAR	SEM	SDM	SDEM
Externalités spatiales	X			X	X
Effets d'entraînement		X		X	
Omission de variables			X		X
Hétérogénéité spatiale			X		X
Effets mixtes				X	X
Application de la matrice de pondération spatiale	VI	VD	TE	VI et VD	VI et TE

VD = Variable dépendante. VI = Variables indépendantes. TE = terme d'erreur.

3.1.1 Modèle SLX : prise en compte des caractéristiques des voisins

3.1.1.1 Description du modèle SLX

Dans un modèle SLX (***s**patial **l**ag of **X** model*), la dimension spatiale est intégrée par une possible influence des variables indépendantes des observations voisines ou des caractéristiques des voisins. Cette spécification vise habituellement à intégrer ce que l'on identifie dans la littérature par les effets d'externalités spatiales, c'est-à-dire capter l'influence des caractéristiques d'un voisin ($\mathbf{W}\Delta x_k$) sur le comportement d'intérêt d'une observation donnée ($\Delta \mathbf{y}$) (figure 3.1).

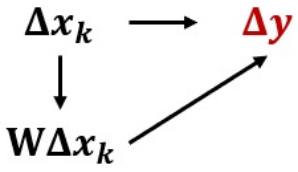


Figure 3.1 – Effets marginaux dans un modèle SLX

Deux principaux avantages sont liés à cette spécification. Premièrement, cette approche peut se faire simplement en ajoutant des variables à l'équation de départ, soit la spécification du PGD original. La création de variables indépendantes, supposées exogènes, spatialement décalées peut être effectuée de manière simple (section 1.3). Deuxièmement, ce modèle peut être estimé par la méthode des moindres carrés ordinaire (MCO), ce qui n'est pas possible avec les autres spécifications.

Note

Rappel sur les variables spatialement décalées
Dans le premier chapitre (section 1.3), nous avons vu comment calculer une variable spatialement décalée avec une matrice de pondération spatiale (figure 1.10). À titre de rappel, lorsque cette dernière est standardisée en ligne, elle correspond à la valeur moyenne dans les unités voisines ou proches (dépendamment si la matrice est définie selon la contiguïté ou la proximité).

Le modèle SLX prend la spécification suivante :

$$\mathbf{y} = \iota\alpha + \mathbf{X}\beta + \mathbf{WX}\theta + \varepsilon \tag{3.1}$$

Où :

- $\mathbf{y}$ est la variable dépendante, soit un vecteur de dimension $(N \times 1)$ où N est le nombre total d'observations mobilisées dans l'analyse, avec $i = 1, 2, \ldots, N$.
- ι est un vecteur de dimension $(N \times 1)$ comportant uniquement des valeurs de 1.
- $\mathbf{X}$ est une matrice de dimension $(N \times K)$ renfermant l'ensemble des K vecteurs de variables explicatives en plus d'une colonne de 1, dans la première colonne de la matrice, pour la constante, d'où $(K + 1)$.
- α est un scalaire identifiant l'ordonnée à l'origine.
- β est un vecteur de dimension $(K+1)$ qui renferme l'ensemble des coefficients de régression, pour les K variables indépendantes et la constante.
- $\mathbf{WX}$ est une matrice de variables indépendantes décalées spatialement, résultant d'une opération matricielle simple, soit $\mathbf{WX} = \mathbf{W} \times \mathbf{X}$, où W est de dimension (NN), $\mathbf{X}$ est de dimension $(N \times K)$, d'où $\mathbf{WX}$ est de dimension $(N \times K)$, où chaque observation représente la moyenne des valeurs des variables indépendantes autour d'une observation donnée.
- θ est un vecteur de coefficients associés aux valeurs spatialement décalées des variables indépendantes (excluant la constante) de dimension $(K \times 1)$.
- ε est un vecteur de dimension $(N \times 1)$ des termes d'erreurs.

Le PGD suggère que la valeur de chaque unité spatiale du modèle est ainsi expliquée à la fois par ses propres caractéristiques et celles de ses voisins ou des observations à proximité en fonction de la matrice de pondération spatiale ($\mathbf{W}$).

Une des particularités du modèle SLX est qu'il introduit une complexité dans l'interprétation des résultats et le calcul des effets marginaux puisque la variable x_k apparaît à deux endroits dans le modèle : d'abord dans la matrice des variables indépendantes originales ($\mathbf{X}$) et ensuite dans la matrice de variables indépendantes spatialement décalées ($\mathbf{WX}$). Lorsque la matrice de pondérations est standardisée en ligne, le calcul des effets marginaux est alors donné par la dérivée partielle suivante :

$$\frac{\partial y}{\partial x_k} = \mathbf{I}\beta_k + \mathbf{W}\theta_k \tag{3.2}$$

Pour une matrice de pondération spatiale standardisée en ligne, cette expression se réduit à la valeur du coefficient associé à la variable indépendante originale, β_k, en plus de la valeur du coefficient associé à la variable décalée spatialement, θ_k. L'expression synthétise l'effet marginal total, alors que le coefficient β_k est habituellement désigné par l'effet direct et le coefficient θ_k synthétise l'effet indirect, puisque venant des voisins.

C'est d'ailleurs la valeur du coefficient θ_k qui introduit ce que l'on qualifie d'**externalité spatiale**. Elle exprime comment la variable dépendante varie lorsque les caractéristiques des voisins changent. Les changements dans la variable d'intérêt peuvent ainsi venir d'une variation des caractéristiques des voisins.

Ce modèle est aussi désigné sous le terme d'effet spatial local, par opposition à l'effet global (voir modèle SAR, section 3.1.2). La variation des caractéristiques des voisins a une portée limitée sur le changement de valeur de la variable dépendante. Seulement certaines observations sont affectées : celles dont les caractéristiques des voisins ont changé.

⚠ **Attention**

Modèle SLX et correction de la dépendance spatiale du modèle MCO de départ
Si le modèle SLX offre une solution intéressante pour régler le potentiel problème d'autocorrélation spatiale des résidus, rien n'assure pour autant qu'il permet de régler tous les problèmes, même si les coefficients associés aux variables spatialement décalées sont statistiquement significatifs. L'inclusion de variables indépendantes additionnelles peut ne pas suffire à capter ce qui, au départ dans la régression par MCO, se cache dans les résidus du modèle. Un test d'autocorrélation spatiale sur les résidus du modèle SLX est un réflexe nécessaire à avoir.

3.1.1.2 Modèle SLX dans R

Pour illustrer la mise en œuvre dans R des différents modèles d'économétrie spatiale, nous utilisons le jeu de données sur Lyon (section 1.1.1). Les modèles seront construits avec le dioxyde d'azote (NO_2) comme variable dépendante et cinq variables sociodémographiques comme variables indépendantes (section 1.1.1). Le modèle MCO (non spatial) a déjà été présenté à la section 1.6.1.

Le modèle SLX est construit avec la fonction `lmSLX` du *package* `spatialreg` (Bivand, Millo et Piras 2021). Dans le code ci-dessous, remarquez le paramètre `listw = w_rook` qui est utilisé pour spécifier la matrice de pondération spatiale.

```
## Chargement des packages et des données
library(sf)
library(tmap)
library(spdep)
library(spatialreg)
load("data/Lyon.Rdata")

## Matrice de contiguïté selon le partage d'un segment (Rook)
nb_rook <- poly2nb(LyonIris, queen = FALSE)
w_rook <- nb2listw(nb_rook, zero.policy = TRUE, style = "W")

## Construction du modèle MCO
modele_MCO <- lm(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                 data = LyonIris)

## Construction du modèle SLX
modele_SLX <- spatialreg::lmSLX(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                   listw = w_rook,   # matrice de pondération spatiale
                   data = LyonIris)  # dataframe
```

```
## Résultats du modèle
summary(modele_SLX)
```

```
Call:
lm(formula = formula(paste("y ~ ", paste(colnames(x)[-1], collapse = "+"))),
    data = as.data.frame(x), weights = weights)

Coefficients:
                Estimate    Std. Error  t value     Pr(>|t|)
(Intercept)      5.068e+01   4.188e+00   1.210e+01   1.040e-29
Pct0_14         -2.040e-01   6.268e-02  -3.255e+00   1.211e-03
Pct_65          -3.771e-02   5.361e-02  -7.033e-01   4.822e-01
Pct_Img          1.041e-01   4.849e-02   2.146e+00   3.235e-02
Pct_brevet      -7.363e-02   3.550e-02  -2.074e+00   3.857e-02
NivVieMed       -1.844e-01   1.106e-01  -1.667e+00   9.617e-02
lag.Pct0_14     -7.759e-01   1.030e-01  -7.537e+00   2.315e-13
lag.Pct_65      -6.454e-02   9.115e-02  -7.081e-01   4.792e-01
lag.Pct_Img      6.465e-01   8.593e-02   7.524e+00   2.526e-13
lag.Pct_brevet  -3.013e-01   6.157e-02  -4.893e+00   1.344e-06
lag.NivVieMed   -1.805e-02   1.750e-01  -1.031e-01   9.179e-01
```

⚠ **Attention**

Effets directs, indirects et totaux

La formulation d'un modèle SLX implique deux types d'effets pour les variables indépendantes (X) :

- Les **effets directs**, soit ceux des caractéristiques des entités spatiales. Ils correspondent aux coefficients β des variables indépendantes ($\mathbf{X}$). Autrement dit, pour une observation i, à chaque augmentation d'une unité d'une caractéristique $\mathbf{X}$, la valeur de y_i va varier (augmenter ou diminuer) en fonction du coefficient β.
- Les **effets indirects**, soit ceux des caractéristiques des entités spatiales voisines ou proches définies selon la matrice de pondération spatiale. Ils correspondent aux coefficients θ des variables indépendantes spatialement décalées ($\mathbf{WX}$). Autrement dit, les valeurs de $\mathbf{WX}$ des entités spatiales proches ou voisines j de i vont aussi être amenées à varier, impactant alors les valeurs y_i selon les coefficients θ.
- Pour calculer l'impact total, il suffit de sommer son effet direct (β_k) et son effet indirect (θ_k) pour obtenir son effet total.

Le code suivant permet de calculer ces effets directs et indirects.

```
## Effets directs, indirects et totaux (uniquement les coefficients)
impacts(modele_SLX)
```

```
Impact measures (SlX, glht):
                Direct    Indirect      Total
Pct0_14    -0.20403803 -0.77590830 -0.9799463
```

```
Pct_65     -0.03770918 -0.06453809 -0.1022473
Pct_Img     0.10406359  0.64653923  0.7506028
Pct_brevet -0.07363272 -0.30128171 -0.3749144
NivVieMed  -0.18440960 -0.01804718 -0.2024568
```

```
## Effets directs, indirects et totaux (coefficients, valeurs de z et de p)
summary(impacts(modele_SLX))
```

```
Impact measures (SlX, glht, n-k):
                Direct    Indirect      Total
Pct0_14    -0.20403803 -0.77590830 -0.9799463
Pct_65     -0.03770918 -0.06453809 -0.1022473
Pct_Img     0.10406359  0.64653923  0.7506028
Pct_brevet -0.07363272 -0.30128171 -0.3749144
NivVieMed  -0.18440960 -0.01804718 -0.2024568
========================================================
Standard errors:
               Direct   Indirect      Total
Pct0_14    0.06268202 0.10295210 0.10045332
Pct_65     0.05361420 0.09114695 0.08556272
Pct_Img    0.04849085 0.08593145 0.08060028
Pct_brevet 0.03549819 0.06157121 0.05975821
NivVieMed  0.11063207 0.17499339 0.14911021
========================================================
Z-values:
               Direct   Indirect     Total
Pct0_14    -3.2551283 -7.5365951 -9.755241
Pct_65     -0.7033432 -0.7080664 -1.194998
Pct_Img     2.1460460  7.5238953  9.312658
Pct_brevet -2.0742665 -4.8932234 -6.273857
NivVieMed  -1.6668729 -0.1031306 -1.357766

p-values:
           Direct    Indirect   Total
Pct0_14    0.0011334 4.8184e-14 < 2.22e-16
Pct_65     0.4818419 0.47890    0.23209
Pct_Img    0.0318693 5.3069e-14 < 2.22e-16
Pct_brevet 0.0380546 9.9198e-07 3.5221e-10
NivVieMed  0.0955397 0.91786    0.17454
```

À la lecture des valeurs de *p*, nous constatons que seule la variable `Pct0_14` a des impacts direct et indirect significatifs au seuil 0,01. L'augmentation d'un point de pourcentage de la population de moins de 15 ans est associée localement à une réduction de 0,20 de la concentration annuelle du dioxyde d'azote. La même augmentation d'un point de pourcentage dans les entités voisines entraîne une réduction de 0,78. L'effet total est donc une réduction de 0,98.

Dépendance spatiale du modèle SLX?

Ce modèle a-t-il corrigé le problème de dépendance spatiale du modèle de régression linéaire classique? Avec une valeur du I de Moran de 0,605 ($p < 0{,}001$), les résidus sont toujours fortement autocorrélés spatialement (figure 3.2).

```
legende_parametres <- list(text.separator = "à",
                           decimal.mark = ",",
                           big.mark = " ")
lm.morantest(modele_SLX, w_rook, alternative="two.sided")
```

```
    Global Moran I for regression residuals

data:
model: lm(formula = formula(paste("y ~ ", paste(colnames(x)[-1],
collapse = "+"))), data = as.data.frame(x), weights = weights)
weights: w_rook

Moran I statistic standard deviate = 21.951, p-value < 2.2e-16
alternative hypothesis: two.sided
sample estimates:
Observed Moran I      Expectation         Variance
    0.6046602748    -0.0072844321     0.0007771643
```

```
LyonIris$SLX.Residus <- residuals(modele_SLX)
tm_shape(LyonIris)+
  tm_fill(col = "SLX.Residus", n = 5, style = "quantile",
          legend.format = legende_parametres,
          palette = "-RdBu", title = "Résidus") +
tm_layout(frame = FALSE) +
tm_scale_bar(breaks = c(0,5))
```

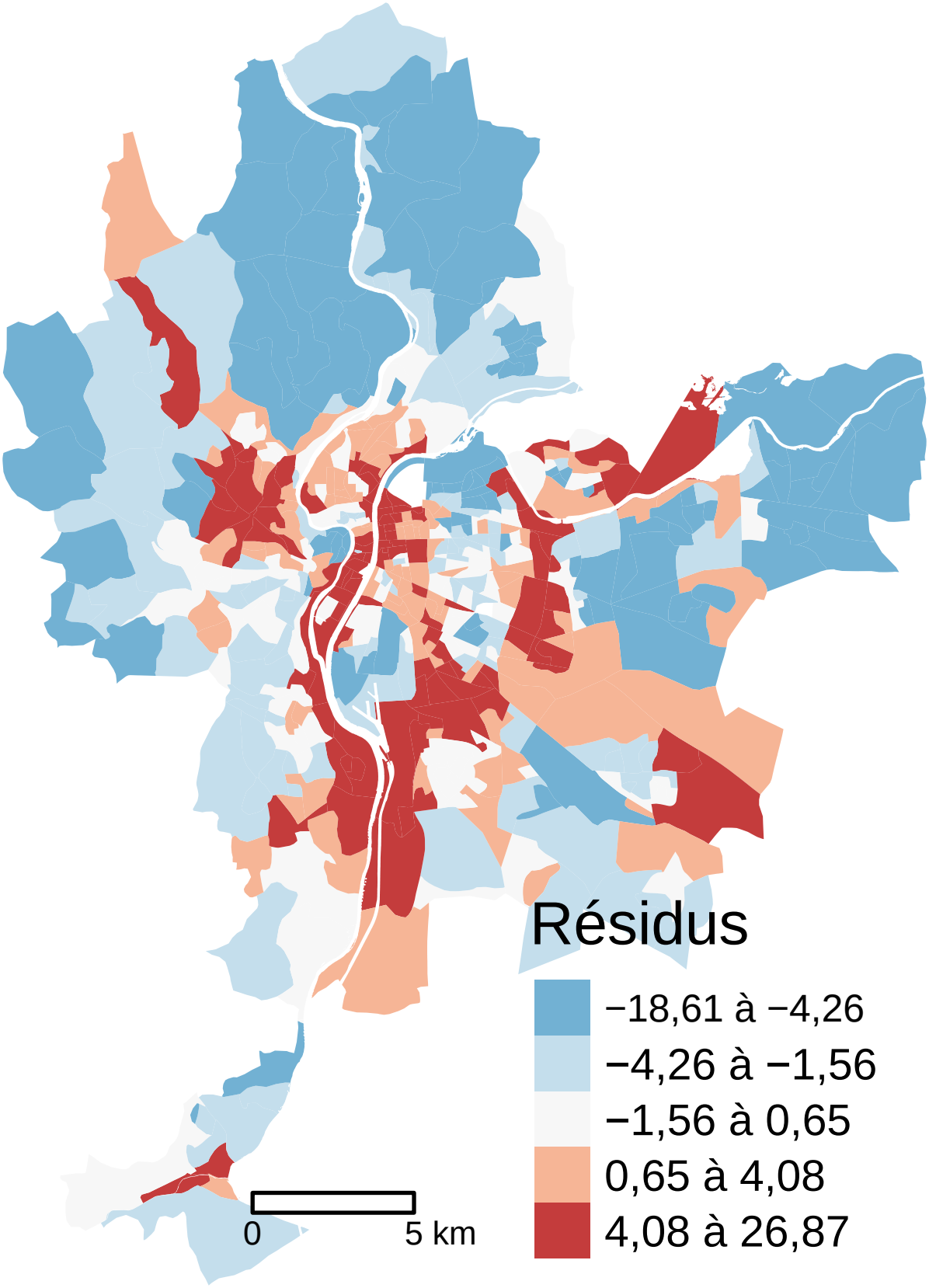


FIGURE 3.2 – Cartographie des résidus du modèle SLX

3.1.2 Modèle SAR : autocorrélation spatiale sur la variable dépendante

3.1.2.1 Description du Modèle SAR

Une autre spécification consiste à inclure, comme variable additionnelle au modèle de départ, la variable dépendante spatialement décalée, notée $\mathbf{Wy}$. Le modèle, généralement désigné par le terme SAR (***s**patial **a**uto**r**egressive*), prend la forme suivante :

$$\mathbf{y} = \rho\mathbf{W}\mathbf{y} + \iota\alpha + \mathbf{X}\beta + \epsilon \tag{3.3}$$

Où :

- $\mathbf{Wy}$ est un vecteur de la variable dépendante spatialement décalée de dimension $(N \times 1)$.
- ρ est un paramètre (scalaire, donc de dimension (1×1)) autorégressif, qui varie habituellement entre -1 et 1 (autrement un problème de non-stationnarité survient).

Par opposition au modèle SLX, cet ajout nécessite une autre méthode d'estimation que les moindres carrés ordinaire puisque la variable additionnelle du côté droit de l'équation est endogène et qu'elle dépend des valeurs de la variable dépendante, qui elles, dépendent des valeurs prises par les caractéristiques des observations. La simple construction de variables spatialement décalées ne peut être considérée. Le processus générateur des données (PGD) s'exprime donc sous une forme un peu plus complexe, soit :

$$y = (\mathbf{I} - \rho W)^{-1} [\iota\alpha + X\beta + \varepsilon] \tag{3.4}$$

Où :

- $\mathbf{I}$ est une matrice identité, c'est-à-dire une matrice où la diagonale comporte des 1 et où les autres éléments sont nuls, de dimension $(N \times N)$.

Le modèle nécessite une méthode d'estimation appropriée. La première approche, par maximum de vraisemblance, fut proposée dès le départ par Luc Anselin (1988). Cette approche a été largement mobilisée dans les premières applications. Bien qu'intéressante, cette approche repose sur certaines hypothèses, dont la distribution du terme d'erreur et l'absence d'hétéroscédasticité.

Plus récemment, Kelejian et Prucha (1998) ont proposé une approche basée sur les moindres carrés généralisés (MCG). Cette approche a l'avantage d'être plus rapide en plus de permettre de corriger l'absence d'hétéroscédasticité. Elle a cependant le désavantage de reposer sur l'hypothèse importante d'exogénéité de la matrice de pondération spatiale, qui sert d'instrument lors du processus d'estimation.

Ce modèle est habituellement qualifié d'**effet de débordement** ou d'**effet d'entraînement**. Comparativement au modèle SLX, les effets marginaux sont identifiés comme étant globaux puisqu'une variation de la valeur de la variable dépendante introduit une variation dans la valeur des variables dépendantes décalées spatialement qui, à leur tour, influencent la variation dans les valeurs dépendantes des observations voisines et ainsi de suite (figure 3.3). Cette boucle de rétroaction peut être interprétée comme la résultante d'un équilibre général spatial.

La dérivée partielle du modèle SAR procure une forme plus complexe :

$$\frac{\partial \mathbf{y}}{\partial x_k} = (\mathbf{I} - \rho\mathbf{W})^{-1} \beta_k \tag{3.5}$$

FIGURE 3.3 – Effets marginaux dans un modèle SAR

C'est donc dire que les effets marginaux sont un peu plus complexes à analyser et à calculer. Une approche classique, celle proposée par LeSage et Pace (2009), consiste à calculer le résultat du produit matriciel à partir d'une approximation basée sur une somme à perpétuité et à extraire la diagonale de la matrice résultante. La diagonale représente l'effet direct et la somme des éléments hors diagonale représentent les effets indirects, afin d'obtenir l'effet marginal total. C'est habituellement cette décomposition qui est proposée par la plupart des logiciels, incluant R. Cette approche peut paraître un peu plus complexe et plus difficilement interprétable puisque les effets marginaux directs ne sont pas définis par les coefficients β_k.

Deux autres approches sont également proposées. Une première, proposée par Abreu *et al.* (2004), reprend la sommation matricielle, mais en proposant une décomposition différente sur la base d'un vocabulaire différent également. Les auteurs proposent néanmoins que le coefficient de régression β_k s'interprète comme un effet direct.

Une seconde, proposée par Kim *et al.* (2003), Steimetz (2010), Dubé et Legros (2014) et Dubé *et al.* (2017), consiste essentiellement à tirer profit des propriétés de la matrice de pondération spatiale standardisée en ligne afin d'y proposer une décomposition simple. Dans ce cas, les auteurs proposent les règles suivantes pour calculer les effets marginaux :

- β_k représente l'effet marginal direct.
- $\beta_k \left(1-\rho\right)^{-1}$, ou encore $\frac{\beta_k}{(1-\rho)}$, identifie l'effet marginal total.
- $\rho\beta_k \left(1-\rho\right)^{-1}$, ou encore $\frac{\rho\beta_k}{(1-\rho)}$, représente l'effet marginal indirect qui peut être obtenu en retranchant l'effet direct de l'effet total.

Des tests non linéaires peuvent être mobilisés afin de tester la significativité des effets marginaux indirects et totaux, alors que le test de significativité du paramètre procure une façon simple de vérifier si l'effet direct est statistiquement significatif.

Tout comme pour la spécification SLX, rien n'indique pour autant que le modèle SAR permet de capter l'essentiel de l'autocorrélation spatiale résiduelle. Un test sur les résidus doit idéalement être mobilisé pour vérifier cette hypothèse critique. La détection d'une autocorrélation spatiale résiduelle significative peut invalider les résultats obtenus par cette approche.

3.1.2.2 Modèle SAR dans R

Le modèle SAR est construit avec la fonction `lagsarlm` du *package* `spatialreg`.

```
## Construction du modèle
modele_SAR <- lagsarlm(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                       listw = w_rook,  # matrice de pondération spatiale
                       data = LyonIris, # dataframe
                       type = 'lag')    # Modèle lag par défaut
```

```r
## Résultats du modèle
summary(modele_SAR, Nagelkerke = TRUE)
```

```
Call:lagsarlm(formula = NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
    NivVieMed, data = LyonIris, listw = w_rook, type = "lag")

Residuals:
      Min        1Q    Median        3Q       Max
-12.86859  -1.88111  -0.49760   0.94464  18.21351

Type: lag
Coefficients: (asymptotic standard errors)
             Estimate Std. Error z value  Pr(>|z|)
(Intercept)  7.838906   1.646232  4.7617 1.919e-06
Pct0_14     -0.098708   0.030554 -3.2306  0.001235
Pct_65      -0.034543   0.026957 -1.2814  0.200044
Pct_Img      0.030241   0.024491  1.2348  0.216917
Pct_brevet  -0.019234   0.017855 -1.0772  0.281384
NivVieMed   -0.098413   0.048985 -2.0090  0.044534

Rho: 0.87939, LR test value: 620.31, p-value: < 2.22e-16
Asymptotic standard error: 0.01942
    z-value: 45.283, p-value: < 2.22e-16
Wald statistic: 2050.5, p-value: < 2.22e-16

Log likelihood: -1366.157 for lag model
ML residual variance (sigma squared): 10.181, (sigma: 3.1908)
Nagelkerke pseudo-R-squared: 0.78962
Number of observations: 506
Number of parameters estimated: 8
AIC: 2748.3, (AIC for lm: 3366.6)
LM test for residual autocorrelation
test value: 0.6198, p-value: 0.43112
```

Dans les résultats ci-dessus, la valeur de rho est de 0,88 (LR = 620, $p < 0{,}001$), traduisant un très fort effet d'entraînement. Autrement dit, lorsqu'en moyenne la concentration de dioxyde d'azote augmente dans les IRIS voisines ($\mathbf{Wy}$), elle augmente aussi fortement dans chaque IRIS ($\mathbf{y}$).

Effets directs, indirects et totaux

À nouveau, il est possible d'utiliser la fonction `impacts` pour obtenir les effets marginaux directs et indirects.

```r
## Effets directs, indirects et totaux (uniquement les coefficients)
impacts(modele_SAR, listw = w_rook, R = 999)
```

```
Impact measures (lag, exact):
               Direct   Indirect      Total
Pct0_14    -0.13878038 -0.6796248 -0.8184052
Pct_65     -0.04856624 -0.2378349 -0.2864012
Pct_Img     0.04251743  0.2082131  0.2507306
Pct_brevet -0.02704205 -0.1324283 -0.1594703
NivVieMed  -0.13836534 -0.6775923 -0.8159576
```

```
## Effets directs, indirects et totaux (coefficients, valeurs de z et de p)
summary(impacts(modele_SAR, listw = w_rook, R = 999), zstats = TRUE, short = TRUE)
```

```
Impact measures (lag, exact):
               Direct   Indirect      Total
Pct0_14    -0.13878038 -0.6796248 -0.8184052
Pct_65     -0.04856624 -0.2378349 -0.2864012
Pct_Img     0.04251743  0.2082131  0.2507306
Pct_brevet -0.02704205 -0.1324283 -0.1594703
NivVieMed  -0.13836534 -0.6775923 -0.8159576
========================================================
Simulation results ( variance matrix):
========================================================
Simulated standard errors
              Direct  Indirect     Total
Pct0_14    0.04284721 0.2510316 0.2893022
Pct_65     0.03809471 0.2003769 0.2374676
Pct_Img    0.03467206 0.1822224 0.2159454
Pct_brevet 0.02459973 0.1285136 0.1525461
NivVieMed  0.06958725 0.3721877 0.4383374

Simulated z-values:
             Direct   Indirect     Total
Pct0_14    -3.241813 -2.7919347 -2.902731
Pct_65     -1.280737 -1.2203411 -1.235189
Pct_Img     1.213985  1.1613708  1.174922
Pct_brevet -1.037336 -0.9896786 -1.001044
NivVieMed  -1.951803 -1.8336896 -1.866821

Simulated p-values:
           Direct    Indirect  Total
Pct0_14    0.0011877 0.0052394 0.0036992
Pct_65     0.2002862 0.2223356 0.2167602
Pct_Img    0.2247535 0.2454911 0.2400258
Pct_brevet 0.2995793 0.3223312 0.3168055
NivVieMed  0.0509616 0.0667001 0.0619266
```

L'interprétation des effets directs se rapproche de celle des coefficients classiques. Ainsi, selon ce modèle, l'augmentation du niveau de vie médian de 1000 € dans un IRIS est associée à une diminution moyenne de la concentration de dioxyde

d'azote de 0,14 dans cet IRIS. L'effet total est de -0,82, indiquant qu'en moyenne, l'augmentation de 1000 € du niveau de vie médian dans un IRIS est associée à une diminution moyenne de 0,82 de la concentration de dioxyde d'azote dans l'ensemble des IRIS. Au final, l'effet indirect est simplement la différence entre l'effet total et l'effet direct. Nous pouvons constater ici que les effets indirects sont plus importants que les effets directs.

Dépendance spatiale du modèle SAR?

Ce modèle a-t-il corrigé le problème de dépendance spatiale du modèle de régression linéaire classique? Avec une valeur du I de Moran de -0,014 (p > 0,10), les résidus ne sont plus spatialement autocorrélés (figure 3.4).

```
## Résidus du modèles
LyonIris$SAR.Residus <- resid(modele_SAR)

## Autocorrélation spatiale des résidus
moran.mc(LyonIris$SAR.Residus, w_rook, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  LyonIris$SAR.Residus
weights: w_rook
number of simulations + 1: 1000

statistic = -0.014281, observed rank = 340, p-value = 0.66
alternative hypothesis: greater
```

```
## Cartographie des résidus
legende_parametres <- list(text.separator = "à",
                           decimal.mark = ",",
                           big.mark = " ")

tm_shape(LyonIris)+
  tm_fill(col = "SAR.Residus", n = 5, style = "quantile",
          legend.format = legende_parametres,
          palette = "-RdBu", title = "Résidus") +
tm_layout(frame=FALSE) +
tm_scale_bar(breaks = c(0,5))
```

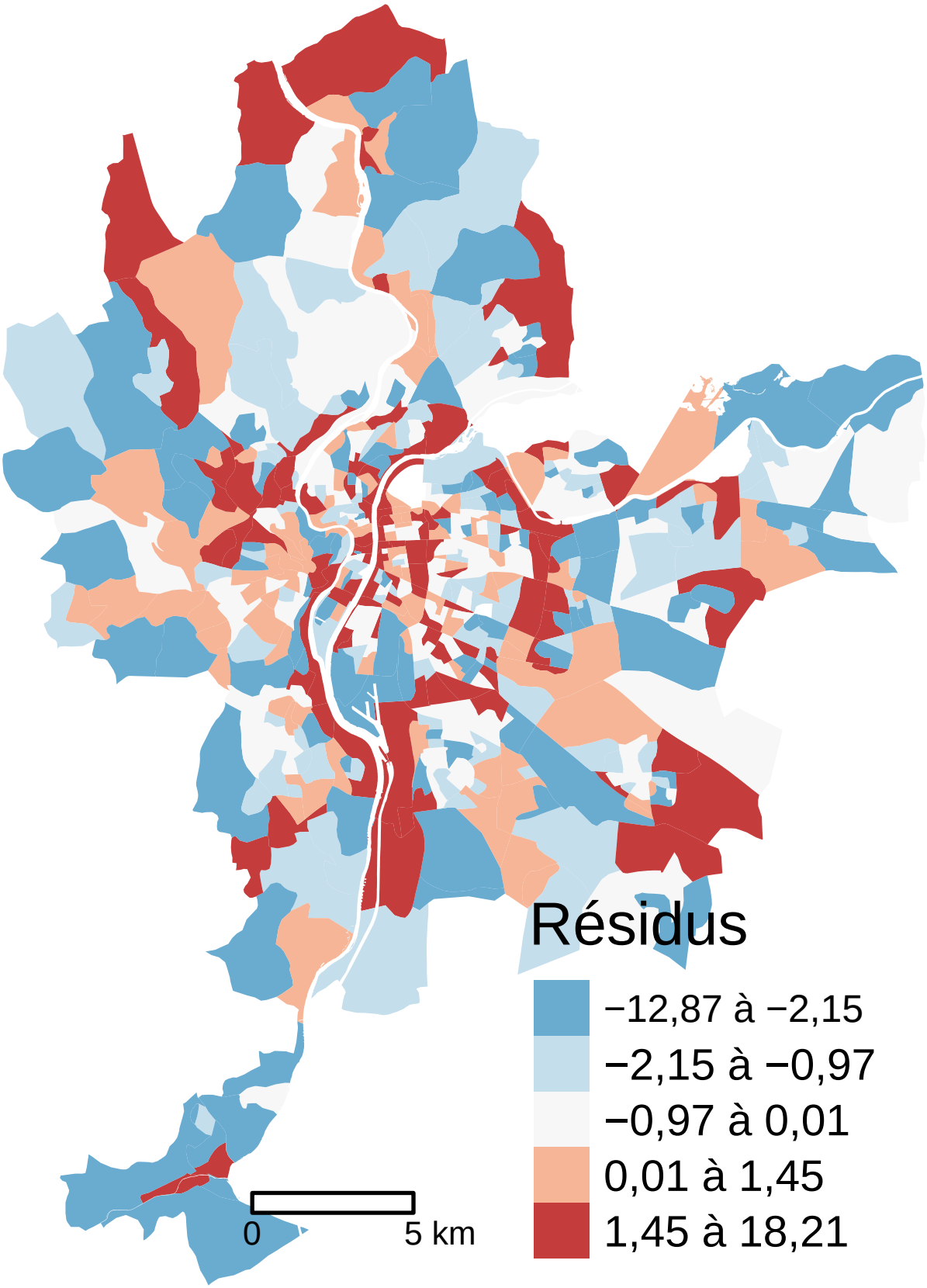


Figure 3.4 – Cartographie des résidus du modèle SAR

3.1.3 Modèle SEM : autocorrélation spatiale sur le terme d'erreur

3.1.3.1 Description du modèle SEM

L'autocorrélation spatiale résiduelle peut être liée à deux explications simples. La première est liée à **l'omission de variables spatiales importantes et significatives**, pour laquelle nous ne disposons d'aucune information spécifique. La seconde peut provenir d'une **forme d'hétérogénéité spatiale** qui n'est pas intégrée à l'analyse. Nous pouvons ainsi imaginer que les comportements sont liés dans l'espace pour des raisons qui ne sont pas observables, et encore moins mesurables.

Dans les deux cas, nous pouvons simplement corriger le problème sans pour autant tenter de tirer profit d'une forme spécifique de relations spatiales (externalités, effets d'entraînement). C'est essentiellement dans ces cas que les modèles avec une spécification autorégressive sur les termes d'erreurs (ou SEM – ***s**patial **e**rror **m**odel*) peuvent être intéressants : ils ne permettent pas d'apporter une richesse supplémentaire à l'analyse et à l'interprétation, mais ils permettent vraisemblablement de corriger le problème d'interdépendance des termes d'erreurs. Plus spécifiquement, le modèle SEM s'exprime par la spécification suivante :

$$\mathbf{y} = \iota\alpha + \mathbf{X}\beta + \nu \tag{3.6}$$

$$\nu = \lambda\mathbf{W}\nu + \varepsilon$$

Où :

- $\mathbf{W}$ est une matrice de pondération spatiale, standardisée en ligne, de dimension $(N \times N)$.
- λ est un paramètre (scalaire, donc de dimension (1×1)) autorégressif qui varie habituellement entre -1 et 1 (autrement un problème de non-stationnarité survient).
- ν est un vecteur de termes d'erreurs spatialement autocorrélés de dimension $(N \times 1)$.
- ε est un vecteur de termes supposés indépendants de dimension $(N \times 1)$.

La forme réduite du PGD est donné par l'expression suivante :

$$\mathbf{y} = \iota\alpha + \mathbf{X}\beta + (\mathbf{I} - \lambda\mathbf{W})^{-1}\varepsilon \tag{3.7}$$

Où :

- $\mathbf{I}$ est une matrice identité, c'est-à-dire une matrice où la diagonale comporte des 1 et où les autres éléments sont nuls, de dimension $(N \times N)$.

Tout comme pour le cas du modèle SAR, le modèle SEM ne peut être estimé par la méthode des moindres carrés ordinaire vu la forme complexe que prend le terme d'erreur dans la forme réduite du PGD. Le modèle est habituellement estimé par maximum de vraisemblance.

Un des avantages avec le modèle SEM est que l'interprétation des résultats est similaire à celle que l'on retrouve dans les régressions linéaires classiques. La dérivée partielle de la forme réduite du PGD est simplement égale au coefficient lié à la variable souhaitée :

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_k} = \mathbf{I}\beta_k \tag{3.8}$$

Or, l'avantage de la structure spatiale des données est justement d'apporter une richesse supplémentaire à l'analyse que l'on ne peut intégrer avec des données aspatiales. Pour cette raison, les modèles SEM ne sont pas particulièrement prisés par les chercheuses et chercheurs qui tentent d'intégrer l'espace aux modèles. Ils représentent une forme de solution de dernier recours lorsqu'il est impossible de capter la structure résiduelle spatiale autrement.

Tout comme pour les autres spécifications, il est judicieux de vérifier si le modèle permet adéquatement de contrôler l'autocorrélation résiduelle. À noter que le modèle SEM peut également mener à une réécriture qui, elle, permet d'intégrer des effets spatiaux qui permettent d'influencer l'interprétation des résultats. Cette forme est désignée par la spécification du modèle Durbin spatial.

3.1.3.2 Modèle SEM dans R

Le modèle SEM est construit avec la fonction `errorsarlm` du *package* `spatialreg`.

```
## Construction du modèle
modele_SEM <- errorsarlm(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                         listw = w_rook,  # matrice de pondération spatiale
                         data = LyonIris) # dataframe

## Résultats du modèle
summary(modele_SEM, Nagelkerke = TRUE)
```

```
Call:errorsarlm(formula = NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
    NivVieMed, data = LyonIris, listw = w_rook)

Residuals:
      Min        1Q    Median        3Q       Max
-12.86150  -1.83161  -0.44106   0.91029  17.94924

Type: error
Coefficients: (asymptotic standard errors)
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 30.544576   2.358173 12.9526  < 2e-16
Pct0_14     -0.035019   0.033393 -1.0487  0.29431
Pct_65      -0.026039   0.028970 -0.8988  0.36874
Pct_Img     -0.016770   0.026176 -0.6407  0.52175
Pct_brevet   0.023708   0.019074  1.2430  0.21388
NivVieMed   -0.146309   0.060273 -2.4274  0.01521

Lambda: 0.91138, LR test value: 613.15, p-value: < 2.22e-16
Asymptotic standard error: 0.01651
    z-value: 55.201, p-value: < 2.22e-16
Wald statistic: 3047.2, p-value: < 2.22e-16

Log likelihood: -1369.737 for error model
```

```
ML residual variance (sigma squared): 9.9971, (sigma: 3.1618)
Nagelkerke pseudo-R-squared: 0.78662
Number of observations: 506
Number of parameters estimated: 8
AIC: 2755.5, (AIC for lm: 3366.6)
```

Dans les résultats ci-dessus, la valeur de *lambda* est de 0,91 (LR = 613, $p < 0{,}001$), traduisant une très forte autocorrélation spatiale sur le terme d'erreur.

Dépendance spatiale du modèle SEM?

Ce modèle a-t-il corrigé le problème de dépendance spatiale du modèle de régression linéaire classique? Avec une valeur du I de Moran de -0,012 ($p = 0{,}616$), les résidus ne sont plus spatialement autocorrélés.

```
## Autocorrélation spatiale des résidus
moran.mc(resid(modele_SEM), w_rook, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  resid(modele_SEM)
weights: w_rook
number of simulations + 1: 1000

statistic = -0.011827, observed rank = 373, p-value = 0.627
alternative hypothesis: greater
```

Les modèles MCO et SEM sont-ils différents?

Pour le vérifier, Pace et Lesage (2008) proposent d'appliquer le test de Hausman qui s'interprète comme suit :

- Si $p < 0{,}05$, alors les coefficient du modèle MCO sont biaisés par l'erreur spatiale et il est préférable d'utiliser le modèle SEM. Cela est le cas pour notre modèle.
- Si $p > 0{,}05$, les résultats des modèles ne sont pas différents et nous pouvons conserver le modèle MCO.

```
Hausman.test(modele_SEM)
```

```
    Spatial Hausman test (asymptotic)

data:  NULL
Hausman test = -467.64, df = 6, p-value < 2.2e-16
```

3.1.4 Modèle SDM : autocorrélation spatiale sur la variable dépendante et les variables indépendantes

3.1.4.1 Description du modèle SDM

Le modèle spatial Durbin (SDM - ***S**patial **D**urbin **M**odel*) est un modèle mixte qui permet à la fois d'intégrer la relation spatiale sur la variable dépendante (**effets d'entraînement ou de débordement**) et sur les variables indépendantes (**externalités**). Elle permet, explicitement, de combiner les modèles SLX et SAR bien que son inspiration soit issue d'une réécriture de la forme réduite du modèle SEM, où chaque élément est multiplié par le terme $\mathbf{I} - \lambda\mathbf{W}$:

$$\mathbf{y} = \lambda\mathbf{W}\mathbf{y} + \iota\alpha + \mathbf{X}\beta + \mathbf{W}\mathbf{X}\beta\lambda + \varepsilon \tag{3.9}$$

Où :

— $\beta\lambda$ est un nouvel ensemble de paramètres liés aux variables indépendantes spatialement décalées, qui est habituellement substitué par θ.

Tout comme pour le SEM et le SAR, le modèle SDM ne peut être estimé par la méthode des moindres carrés ordinaire. Il faut recourir à la méthode du maximum de vraisemblance ou encore à la méthode des moments généralisés, cette dernière approche offrant une flexibilité plus intéressante pour la présence d'hétéroscédasticité.

De la même manière, les coefficients estimés par le modèle permettent une décomposition plus sophistiquée des effets marginaux. La forme réduite du PGD est donnée par :

$$\mathbf{y} = (\mathbf{I} - \lambda\mathbf{W})^{-1}\left[\iota\alpha + \mathbf{X}\beta + \mathbf{W}\mathbf{X}\theta + \varepsilon\right] \tag{3.10}$$

Ainsi, le calcul des effets marginaux est donné par :

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_k} = (\mathbf{I} - \lambda\mathbf{W})^{-1}\left[\mathbf{I}\beta_k + \mathbf{W}\theta_k\right] \tag{3.11}$$

En ce qui concerne la proposition de certains auteurs (Kim, Phipps et Anselin 2003; Steimetz 2010; Dubé et Legros 2014; Dubé et al. 2017), les effets marginaux (figure 3.5) peuvent être exprimés par la décomposition suivante lorsque la matrice de pondération est standardisée en ligne :

— Effet direct : β_k.
— Effet total : $\frac{(\beta_k+\theta_k)}{(1-\rho)}$.
— Effet indirect : effet total − effet direct = $\frac{(\rho\beta_k+\theta_k)}{(1-\rho)}$.

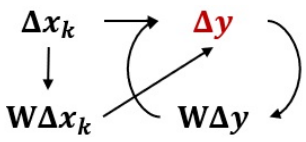


FIGURE 3.5 – Effets marginaux dans un modèle SDM

À noter que les logiciels statistiques retournent les effets marginaux sur la base de la décomposition suggérée par LeSage et Pace (2009), issue de l'approximation du produit matriciel. Si l'effet marginal total est identique dans les deux cas, la

différence vient essentiellement de la manière de calculer l'effet marginal direct, qui influence également le calcul de l'effet marginal indirect obtenu par soustraction.

3.1.4.2 Modèle SDM dans R

Le modèle SDM est construit avec la fonction `lagsarlm` du *package* `spatialreg`. Notez que le paramètre `type = "mixed"` spécifie l'utilisation d'un modèle mixte.

```
## Construction du modèle
modele_SDM <- lagsarlm(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                       listw = w_rook,    # matrice de pondération spatiale
                       data = LyonIris,   # dataframe
                       type = "mixed")

## Résultats du modèle
summary(modele_SDM, Nagelkerke = TRUE)
```

```
Call:lagsarlm(formula = NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
    NivVieMed, data = LyonIris, listw = w_rook, type = "mixed")

Residuals:
      Min        1Q    Median        3Q       Max
-12.60922  -1.77753  -0.43909   0.99252  18.15526

Type: mixed
Coefficients: (asymptotic standard errors)
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)     8.1130457  2.5671301  3.1604 0.001576
Pct0_14        -0.0574046  0.0344908 -1.6643 0.096043
Pct_65         -0.0238715  0.0293647 -0.8129 0.416256
Pct_Img         0.0048364  0.0266560  0.1814 0.856025
Pct_brevet      0.0112746  0.0195259  0.5774 0.563656
NivVieMed      -0.1463876  0.0605853 -2.4162 0.015682
lag.Pct0_14    -0.1242574  0.0581170 -2.1381 0.032512
lag.Pct_65      0.0255480  0.0499646  0.5113 0.609125
lag.Pct_Img     0.1559952  0.0482138  3.2355 0.001214
lag.Pct_brevet -0.0883930  0.0342496 -2.5809 0.009856
lag.NivVieMed   0.1032469  0.0960201  1.0753 0.282257

Rho: 0.84127, LR test value: 492.38, p-value: < 2.22e-16
Asymptotic standard error: 0.023363
    z-value: 36.009, p-value: < 2.22e-16
Wald statistic: 1296.7, p-value: < 2.22e-16
```

```
Log likelihood: -1353.106 for mixed model
ML residual variance (sigma squared): 9.9845, (sigma: 3.1598)
Nagelkerke pseudo-R-squared: 0.8002
Number of observations: 506
Number of parameters estimated: 13
AIC: 2732.2, (AIC for lm: 3222.6)
LM test for residual autocorrelation
test value: 0.0748, p-value: 0.78447
```

Effets directs, indirects et totaux

```r
# Effets directs, indirects et totaux (uniquement les coefficients)
impacts(modele_SDM, listw = w_rook, R = 999)
```

```
Impact measures (mixed, exact):
               Direct    Indirect       Total
Pct0_14    -0.12369039 -1.02079497 -1.14448536
Pct_65     -0.02177191  0.03233406  0.01056215
Pct_Img     0.06632543  0.94692639  1.01325182
Pct_brevet -0.01903815 -0.46681402 -0.48585217
NivVieMed  -0.15403413 -0.11775603 -0.27179016
```

```r
# Effets directs, indirects et totaux (coefficients, valeurs de z et de p)
summary(impacts(modele_SDM, listw = w_rook, R = 999), zstats = TRUE, short = TRUE)
```

```
Impact measures (mixed, exact):
               Direct    Indirect       Total
Pct0_14    -0.12369039 -1.02079497 -1.14448536
Pct_65     -0.02177191  0.03233406  0.01056215
Pct_Img     0.06632543  0.94692639  1.01325182
Pct_brevet -0.01903815 -0.46681402 -0.48585217
NivVieMed  -0.15403413 -0.11775603 -0.27179016
========================================================
Simulation results ( variance matrix):
========================================================
Simulated standard errors
               Direct  Indirect     Total
Pct0_14    0.04451434 0.3531121 0.3811288
Pct_65     0.03570909 0.2831950 0.3059362
Pct_Img    0.03243311 0.2751572 0.2958278
Pct_brevet 0.02466679 0.1991886 0.2147353
NivVieMed  0.06701802 0.4853489 0.5175905

Simulated z-values:
               Direct   Indirect       Total
```

```
Pct0_14    -2.7857641 -2.9004363 -3.01259189
Pct_65     -0.6153219  0.1036986  0.02416951
Pct_Img     2.0601048  3.4545688  3.43904478
Pct_brevet -0.7858859 -2.3708512 -2.28947845
NivVieMed  -2.3001532 -0.2526541 -0.53474150

Simulated p-values:
           Direct    Indirect   Total
Pct0_14    0.0053402 0.00372644 0.00259027
Pct_65     0.5383421 0.91740858 0.98071740
Pct_Img    0.0393885 0.00055117 0.00058377
Pct_brevet 0.4319344 0.01774717 0.02205157
NivVieMed  0.0214395 0.80053552 0.59282862
```

Dépendance spatiale du modèle SDM?

Avec une valeur du I de Moran de -0,005 (p > 0,10), les résidus du modèle SDEM ne sont pas spatialement autocorrélés.

```
moran.mc(resid(modele_SDM), w_rook, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  resid(modele_SDM)
weights: w_rook
number of simulations + 1: 1000

statistic = -0.0046127, observed rank = 466, p-value = 0.534
alternative hypothesis: greater
```

3.1.5 Modèle SDEM : autocorrélation spatiale sur les variables indépendantes et sur le terme d'erreur

3.1.5.1 Description du modèle SDEM

Une autre variante permettant de tirer profit de l'avantage des régressions spatiales, c'est-à-dire l'introduction d'effets liés aux externalités spatiales des observations, consiste à jumeler les modèles SLX et SEM pour obtenir le modèle Durbin avec termes d'erreurs spatialement liés (SDEM - ***S**patial **D**urbin **E**rror **M**odel*). Le modèle s'écrit :

$$\mathbf{y} = \iota\alpha + \mathbf{X}\beta + \mathbf{WX}\theta + \nu \tag{3.12}$$

$$\nu = \lambda\mathbf{W}\nu + \varepsilon$$

Tout comme pour les spécifications SAR, SEM et SDM, le modèle SDEM ne peut être estimé par la méthode des moindres carrés ordinaire et nécessite une méthode d'estimation appropriée.

Entre le modèle SEM et le modèle SDEM, il est clairement avantageux d'opter pour une spécification SDEM si le but est d'intégrer des effets spatiaux et de tenir compte des possibles relations spatiales dans l'interprétation des résultats de régression. Le calcul des effets marginaux (figure 3.6) permet de tenir compte de la spécificité des externalités spatiales.

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x_k}} = \mathbf{I}\beta_k + \mathbf{W}\theta_k \tag{3.13}$$

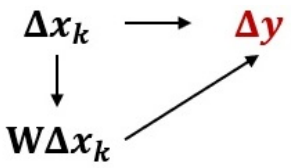


FIGURE 3.6 – Effets marginaux dans un modèle SDEM

Dans le cas où la matrice de pondération spatiale $\mathbf{W}$ est standardisée en ligne, l'interprétation des effets marginaux est plutôt simple et directe :

- β_k est l'effet marginal direct.
- θ_k est l'effet marginal indirect et l'effet d'externalité spatiale.
- $\beta_k + \theta_k$ est l'effet marginal total.

Finalement, comme pour les spécifications précédentes, il est sage de vérifier, avec l'aide d'une statistique générale, si les résidus du modèle permettent de ne pas rejeter l'hypothèse d'indépendance entre les termes d'erreurs. Or, puisque la structure spatiale est explicitement intégrée dans le terme d'erreur du modèle, cette spécification réussit généralement à contrôler le problème de manière efficace.

3.1.5.2 Modèle SDEM dans R

Construction du modèle SDEM dans R

Le modèle SDEM est construit avec la fonction `errorsarlm` du *package* `spatialreg`. Notez que le paramètre `etype = "mixed"` spécifie l'utilisation d'un modèle mixte.

```
## Construction du modèle
modele_SDEM <- errorsarlm(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                          listw = w_rook,  # matrice de pondération spatiale
                          data = LyonIris, # dataframe
                          etype = 'emixed')

## Résultats du modèle
summary(modele_SDEM, Nagelkerke = TRUE)
```

```
Call:errorsarlm(formula = NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
    NivVieMed, data = LyonIris, listw = w_rook, etype = "emixed")

Residuals:
      Min        1Q    Median        3Q       Max
-12.99324  -1.82407  -0.45644   1.06084  18.21108

Type: error
Coefficients: (asymptotic standard errors)
               Estimate Std. Error z value  Pr(>|z|)
(Intercept)  37.061010   6.501018  5.7008 1.192e-08
Pct0_14      -0.081998   0.041699 -1.9664   0.04925
Pct_65       -0.026329   0.034714 -0.7585   0.44817
Pct_Img       0.004656   0.031028  0.1501   0.88072
Pct_brevet    0.009785   0.023884  0.4097   0.68203
NivVieMed    -0.167855   0.068005 -2.4683   0.01358
lag.Pct0_14  -0.176747   0.102345 -1.7270   0.08417
lag.Pct_65    0.010533   0.089183  0.1181   0.90599
lag.Pct_Img   0.092785   0.079704  1.1641   0.24437
lag.Pct_brevet -0.038048   0.056688 -0.6712   0.50211
lag.NivVieMed  -0.102531   0.172405 -0.5947   0.55204

Lambda: 0.8976, LR test value: 464.09, p-value: < 2.22e-16
Asymptotic standard error: 0.018242
    z-value: 49.204, p-value: < 2.22e-16
Wald statistic: 2421, p-value: < 2.22e-16

Log likelihood: -1367.25 for error model
ML residual variance (sigma squared): 10.046, (sigma: 3.1696)
Nagelkerke pseudo-R-squared: 0.78871
Number of observations: 506
Number of parameters estimated: 13
AIC: 2760.5, (AIC for lm: 3222.6)
```

Effets directs, indirects et totaux

```
## Effets directs, indirects et totaux (uniquement les coefficients)
impacts(modele_SDEM, listw = w_rook, R = 999)
```

```
Impact measures (SDEM, glht):
                Direct    Indirect       Total
Pct0_14    -0.081997642 -0.17674683 -0.25874447
Pct_65     -0.026329370  0.01053248 -0.01579689
Pct_Img     0.004656039  0.09278511  0.09744115
Pct_brevet  0.009784961 -0.03804813 -0.02826317
```

```
NivVieMed  -0.167855498 -0.10253070 -0.27038620
```

```
## Effets directs, indirects et totaux (coefficients, valeurs de z et de p)
summary(impacts(modele_SDEM, listw = w_rook, R = 999), zstats = TRUE, short = TRUE)
```

```
Impact measures (SDEM, glht, n):
                Direct    Indirect       Total
Pct0_14    -0.081997642 -0.17674683 -0.25874447
Pct_65     -0.026329370  0.01053248 -0.01579689
Pct_Img     0.004656039  0.09278511  0.09744115
Pct_brevet  0.009784961 -0.03804813 -0.02826317
NivVieMed  -0.167855498 -0.10253070 -0.27038620
========================================================
Standard errors:
              Direct   Indirect      Total
Pct0_14    0.04169878 0.10234506 0.13146453
Pct_65     0.03471367 0.08918350 0.11192948
Pct_Img    0.03102807 0.07970364 0.09949722
Pct_brevet 0.02388387 0.05668833 0.07344175
NivVieMed  0.06800483 0.17240549 0.21172909
========================================================
Z-values:
              Direct   Indirect      Total
Pct0_14    -1.9664279 -1.7269698 -1.9681695
Pct_65     -0.7584727  0.1180989 -0.1411325
Pct_Img     0.1500589  1.1641264  0.9793354
Pct_brevet  0.4096890 -0.6711810 -0.3848379
NivVieMed  -2.4682880 -0.5947067 -1.2770385

p-values:
           Direct   Indirect Total
Pct0_14    0.049249 0.084173 0.049049
Pct_65     0.448168 0.905989 0.887765
Pct_Img    0.880718 0.244373 0.327414
Pct_brevet 0.682034 0.502105 0.700358
NivVieMed  0.013576 0.552040 0.201589
```

Dépendance spatiale du modèle SDEM?

Avec une valeur du I de Moran de -0,010 (p > 0,10), les résidus du modèle SDEM ne sont pas spatialement autocorrélés.

```
moran.mc(resid(modele_SDEM), w_rook, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I
```

```
data:  resid(modele_SDEM)
weights: w_rook
number of simulations + 1: 1000

statistic = -0.010362, observed rank = 406, p-value = 0.594
alternative hypothesis: greater
```

3.1.6 Modèle généralisé : autocorrélation spatiale de l'ensemble des variables et des composantes du modèle

3.1.6.1 Description du modèle généralisé

Le modèle spatial généralisé (*Manski model* en anglais), qui est essentiellement une extension du modèle SDM lorsque le processus autorégressif est ajouté aux termes d'erreurs, permet une représentation complète des possibilités. Il permet également de réfléchir à une stratégie du général vers le particulier pour la sélection de la spécification finale :

$$\mathbf{y} = \lambda \mathbf{W}\mathbf{y} + \iota\alpha + \mathbf{X}\beta + \mathbf{W}\mathbf{X}\theta + \nu \tag{3.14}$$

$$\nu = \lambda \mathbf{W}\nu + \varepsilon$$

La spécification tient compte, tout comme le modèle SDM, des effets de débordement (ou effets d'entraînement) spatiaux ainsi que des effets d'externalités spatiales.

Le modèle spatial généralisé ne peut être estimé par la méthode des moindres carrés ordinaire. Il faut recourir à la méthode du maximum de vraisemblance ou encore à la méthode des moments généralisés, cette dernière approche offrant une flexibilité intéressante pour la présence d'hétéroscédasticité.

D'une part, certains auteurs notent que des problèmes peuvent survenir lorsque les matrices de pondération spatiale mobilisées pour estimer le modèle sont identiques pour l'ensemble des processus autorégressifs (Le Gallo 2002; Anselin et Bera 1998). D'autre part, Elhorst (2014) suggère d'estimer d'abord cette spécification afin de faire un choix de modèle, puisqu'il procure une structure emboîtée (voir section suivante).

3.1.6.2 Modèle généralisé (Manski) dans R

```r
## Construction du modèle
modele_Manski <- sacsarlm(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                          listw = w_rook,  # matrice de pondération spatiale
                          data = LyonIris, # dataframe
                          type="sacmixed")

## Résultats du modèle
summary(modele_Manski, Nagelkerke = TRUE)
```

```
Call:sacsarlm(formula = NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
    NivVieMed, data = LyonIris, listw = w_rook, type = "sacmixed")

Residuals:
      Min        1Q    Median        3Q       Max
-12.54832  -1.80538  -0.43054   0.99266  18.04011

Type: sacmixed
Coefficients: (asymptotic standard errors)
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    7.7927132  2.8939671  2.6927 0.007087
Pct0_14       -0.0555696  0.0347795 -1.5978 0.110094
Pct_65        -0.0233490  0.0294070 -0.7940 0.427199
Pct_Img        0.0035044  0.0268385  0.1306 0.896114
Pct_brevet     0.0122230  0.0196832  0.6210 0.534608
NivVieMed     -0.1463864  0.0606995 -2.4117 0.015880
lag.Pct0_14   -0.1211022  0.0602980 -2.0084 0.044601
lag.Pct_65     0.0258497  0.0495836  0.5213 0.602133
lag.Pct_Img    0.1541563  0.0497015  3.1016 0.001924
lag.Pct_brevet -0.0874516  0.0348103 -2.5122 0.011997
lag.NivVieMed  0.1050658  0.0955590  1.0995 0.271556

Rho: 0.84762
Asymptotic standard error: 0.037124
    z-value: 22.832, p-value: < 2.22e-16
Lambda: -0.027606
Asymptotic standard error: 0.11539
    z-value: -0.23924, p-value: 0.81092

LR test value: 646.48, p-value: < 2.22e-16

Log likelihood: -1353.074 for sacmixed model
ML residual variance (sigma squared): 9.9326, (sigma: 3.1516)
Nagelkerke pseudo-R-squared: 0.80022
Number of observations: 506
Number of parameters estimated: 14
AIC: 2734.1, (AIC for lm: 3366.6)
```

```r
## Effets marginaux
impacts(modele_Manski, listw = w_rook)
```

```
Impact measures (sacmixed, exact):
                Direct    Indirect      Total
Pct0_14    -0.12204341 -1.03736491 -1.1594083
Pct_65     -0.02095459  0.03736568  0.0164111
```

```
Pct_Img     0.06560041  0.96904739  1.0346478
Pct_brevet -0.01824321 -0.47544430 -0.4936875
NivVieMed  -0.15390071 -0.11726571 -0.2711664
```

```
## Effets marginaux avec les valeurs de z
summary(impacts(modele_Manski,listw = w_rook, R = 500), zstats=TRUE)
```

```
Impact measures (sacmixed, exact):
                Direct    Indirect      Total
Pct0_14    -0.12204341 -1.03736491 -1.1594083
Pct_65     -0.02095459  0.03736568  0.0164111
Pct_Img     0.06560041  0.96904739  1.0346478
Pct_brevet -0.01824321 -0.47544430 -0.4936875
NivVieMed  -0.15390071 -0.11726571 -0.2711664
========================================================
Simulation results ( variance matrix):
Direct:

Iterations = 1:500
Thinning interval = 1
Number of chains = 1
Sample size per chain = 500

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

               Mean      SD Naive SE Time-series SE
Pct0_14    -0.12453 0.04359 0.001949       0.001949
Pct_65     -0.02089 0.03593 0.001607       0.001607
Pct_Img     0.06644 0.03343 0.001495       0.001495
Pct_brevet -0.01757 0.02560 0.001145       0.001131
NivVieMed  -0.15509 0.06867 0.003071       0.003071

2. Quantiles for each variable:

               2.5%      25%      50%       75%    97.5%
Pct0_14    -0.21368 -0.15221 -0.12284 -0.096142 -0.03894
Pct_65     -0.09103 -0.04422 -0.01988  0.002914  0.05010
Pct_Img     0.00613  0.04216  0.06715  0.088535  0.13026
Pct_brevet -0.06593 -0.03378 -0.01685 -0.002912  0.03923
NivVieMed  -0.28177 -0.20528 -0.15493 -0.108990 -0.01781

========================================================
Indirect:

Iterations = 1:500
```

```
Thinning interval = 1
Number of chains = 1
Sample size per chain = 500

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

               Mean     SD Naive SE Time-series SE
Pct0_14    -1.08869 0.3781 0.016909       0.016909
Pct_65      0.03405 0.3204 0.014327       0.014327
Pct_Img     0.99868 0.3326 0.014873       0.014873
Pct_brevet -0.47026 0.2134 0.009546       0.009546
NivVieMed  -0.10219 0.5242 0.023443       0.020085

2. Quantiles for each variable:

              2.5%     25%      50%     75%    97.5%
Pct0_14    -1.8894 -1.2953 -1.07364 -0.8474 -0.41249
Pct_65     -0.6296 -0.1658  0.03295  0.2261  0.69366
Pct_Img     0.4799  0.8014  0.96797  1.1785  1.74462
Pct_brevet -0.9924 -0.5825 -0.46489 -0.3314 -0.09253
NivVieMed  -1.1752 -0.3944 -0.09501  0.1910  0.88677

============================================================
Total:

Iterations = 1:500
Thinning interval = 1
Number of chains = 1
Sample size per chain = 500

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

               Mean     SD Naive SE Time-series SE
Pct0_14    -1.21322 0.4031  0.01803        0.01803
Pct_65      0.01316 0.3430  0.01534        0.01636
Pct_Img     1.06512 0.3509  0.01569        0.01569
Pct_brevet -0.48784 0.2293  0.01025        0.01025
NivVieMed  -0.25728 0.5610  0.02509        0.02180

2. Quantiles for each variable:

              2.5%     25%      50%      75%    97.5%
Pct0_14    -2.0543 -1.4408 -1.21240 -0.95481 -0.49126
Pct_65     -0.6858 -0.1924  0.01043  0.22193  0.72540
Pct_Img     0.4834  0.8490  1.03759  1.25714  1.84304
```

```
Pct_brevet -1.0351 -0.6196 -0.48653 -0.34056 -0.06371
NivVieMed  -1.4262 -0.5820 -0.25786  0.04803  0.82753

========================================================
Simulated standard errors
               Direct  Indirect     Total
Pct0_14    0.04358933 0.3780919 0.4031120
Pct_65     0.03592614 0.3203598 0.3430010
Pct_Img    0.03343381 0.3325685 0.3508939
Pct_brevet 0.02560254 0.2134449 0.2292614
NivVieMed  0.06866623 0.5242057 0.5609844

Simulated z-values:
               Direct   Indirect       Total
Pct0_14    -2.8568773 -2.8794220 -3.00962391
Pct_65     -0.5815345  0.1062997  0.03837265
Pct_Img     1.9872235  3.0029174  3.03543695
Pct_brevet -0.6863325 -2.2032136 -2.12786140
NivVieMed  -2.2586165 -0.1949353 -0.45861682

Simulated p-values:
           Direct    Indirect  Total
Pct0_14    0.0042783 0.0039840 0.0026157
Pct_65     0.5608802 0.9153446 0.9693906
Pct_Img    0.0468976 0.0026741 0.0024019
Pct_brevet 0.4925035 0.0275797 0.0333486
NivVieMed  0.0239073 0.8454436 0.6465094
```

```
## Autocorrélation spatiale des résidus
moran.mc(resid(modele_Manski), w_rook, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  resid(modele_Manski)
weights: w_rook
number of simulations + 1: 1000

statistic = -0.0009215, observed rank = 546, p-value = 0.454
alternative hypothesis: greater
```

3.2 Quel modèle choisir?

Évidemment, la question qui brûle les lèvres est de savoir quelle spécification nous devrions retenir pour l'analyse empirique et la présentation des résultats. Deux avenues sont possibles.

D'une part, il est possible que **certains impératifs théoriques guident la chercheuse ou le chercheur vers une spécification particulière**. Ainsi, si l'on souhaite vérifier comment les caractéristiques des milieux avoisinants influencent une variable d'intérêt, alors les spécifications SLX ou SDEM sont à privilégier puisqu'elles permettent explicitement d'aborder la question. Si l'intérêt est de voir comment une variation locale d'une variable d'intérêt peut avoir de l'influence à la grandeur du territoire, alors la spécification SAR est probablement à privilégier puisqu'elle permet d'évaluer l'impact spatial « global » d'un changement à microéchelle. Il est également possible qu'une question de recherche trouve un écho parmi les deux possibilités, dans lequel des cas le modèle SDM s'avère intéressant.

Si une chercheuse ou un chercheur **souhaite essentiellement corriger la présence d'autocorrélation spatiale entre les résidus** sans intégrer explicitement de liens spatiaux dans l'interprétation du modèle, alors la spécification SEM peut s'avérer intéressante. Il faut néanmoins avouer que cette correction de l'autocorrélation spatiale n'apporte que peu d'intérêt en ce qui concerne l'interprétation, sauf celui de corriger les écarts-types des coefficients estimés.

D'autre part, **le choix peut également être basé sur des considérations plus statistiques**. Dans ce cas, on souhaite identifier, sur la base de critères préalablement définis, le modèle qui offre la meilleure performance. Bien que le test de Moran permette de détecter la présence d'autocorrélation spatiale dans les résidus, le test n'est malheureusement pas mobilisable pour tenter d'identifier une spécification préférable.

Il existe trois différentes stratégies de tests : les tests de ratio de vraisemblance, qui nécessitent l'estimation des modèles contraints (les spécifications linéaires classiques) et non contraints (les modèles économétriques spatiaux); les tests de Wald, qui nécessitent aussi l'estimation des modèles contraints et non contraints; et les tests du multiplicateur de Lagrange (ou tests LM), qui nécessitent uniquement l'estimation des modèles contraints.

3.2.1 Tests du multiplicateur de Lagrange (LM) sur le modèle MCO

Une procédure relativement simple existe pour discriminer entre les spécifications SAR, SEM et MCO. Puisque les modèles contraints sont relativement simples et directs, les tests LM sont souvent privilégiés en pratique. Ces tests (en version simple et robuste) ont été largement popularisés par Anselin *et al.* (1996) pour vérifier si le recours à un modèle autorégressif est nécessaire, comparativement à un modèle de régression classique (MCO).

Les tests sont calculés sur la base des estimations obtenues par MCO pour la spécification linéaire classique et avec l'aide d'une matrice de pondération spatiale préalablement identifiée et calculée. Pour chaque spécification des modèles économétriques spatiaux, il existe une forme spécifique pour les tests-LM. Comment alors comparer les différentes statistiques pour identifier la spécification à retenir?

La démarche générale suivante, schématisée à la figure 3.7, peut être utilisée pour guider le choix d'une spécification :

1. Si toutes les valeurs des tests (simples et robustes) sont non significatives ($p > 0{,}05$), alors le recours à un modèle autorégressif n'est pas nécessaire. Nous pouvons conserver le modèle de régression classique (MCO).
2. Si les valeurs de `LMlag` ou de `RLMlag` sont non significatives ($p > 0{,}05$), alors le recours au modèle SAR n'est pas nécessaire.
3. Si les valeurs de `LMerr` ou de `RLMerr` sont non significatives ($p > 0{,}05$), alors le recours au modèle SEM n'est pas nécessaire.
4. Si les valeurs de `RLMlag` et de `RLMerr` sont significatives ($p < 0{,}001$), nous choisissons le modèle ayant la plus forte statistique.
   - Si `RLMlag` > `RLMerr`, il est suggéré d'utiliser la spécification SAR.
   - Si `RLMerr` > `RLMlag`, il est suggéré d'utiliser la spécification SEM.

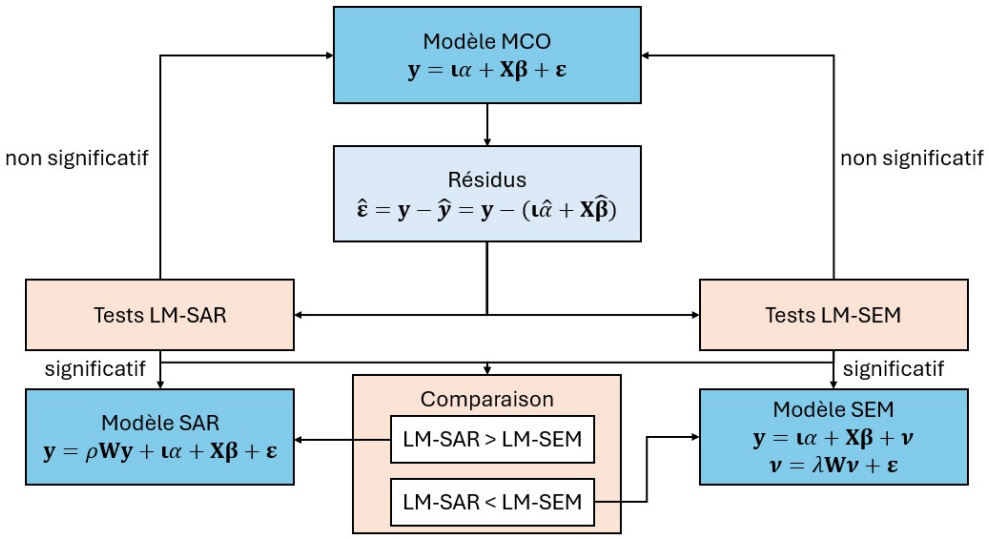


FIGURE 3.7 – Démarche pour choisir entre les modèles MCO, SAR et SEM

Dans R, ces tests LM sont calculés sur le modèle MCO avec la fonction `lm.LMtests` et une matrice de pondération spatiale. Dans les résultats ci-dessous, nous ne retenons pas le modèle SEM car la valeur de 0,740 pour le `RLMerr` n'est pas significative (p = 0,3898). Par contre, les valeurs de `LMlag` et de `RLMlag` (555 et 123) sont significatives, ce qui justifie la sélection du modèle SAR.

```
## Tests de Lagrange
summary(lm.LMtests(model = modele_MCO,
                   listw = w_rook,
                   test = c("LMlag","LMerr","RLMlag","RLMerr")))
```

```
    Rao's score (a.k.a Lagrange multiplier) diagnostics for spatial
    dependence
data:
model: lm(formula = NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
NivVieMed, data = LyonIris)
test weights: listw

         statistic parameter p.value
RSlag    554.65778         1  <2e-16 ***
RSerr    432.83282         1  <2e-16 ***
adjRSlag 122.56452         1  <2e-16 ***
adjRSerr   0.73955         1  0.3898
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.2 Tests du ratio de vraisemblance sur les modèles spatiaux

En partant de la structure emboîtée que représente le modèle spatial généralisé, il est possible de proposer une stratégie qualifiée de « général vers le particulier » à partir de la significativité des différents paramètres associés aux variables décalées spatialement (figure 3.8). Cette stratégie, basée sur les tests du ratio de vraisemblance qui sont issus de l'estimation des modèles non contraints, dans sa forme la plus générale, et des modèles contraints, dans sa forme la plus spécifique (Elhorst 2014).

En comparant la vraisemblance des modèles, il est possible de vérifier si les deux spécifications sont relativement similaires. Si les vraisemblances sont similaires, la spécification contrainte est préférée à celle non contrainte puisqu'elle est plus parcimonieuse, c'est-à-dire qu'elle permet d'exprimer le même type de relation avec moins de paramètres. Si les vraisemblances sont trop différentes, la spécification générale est alors préférée puisqu'elle permet de mieux expliquer la variable dépendante.

Cette procédure permet également de discriminer entre certaines spécifications intégrant plusieurs processus autorégressifs, tels que les modèles SDM et SDEM par rapport, respectivement, aux modèles SAR et SLX, et SEM et SLX. Elle permet également d'orienter le choix d'une forme fonctionnelle sur un processus de sélection complet, puisque celui considère toutes les formes fonctionnelles et les variantes possibles.

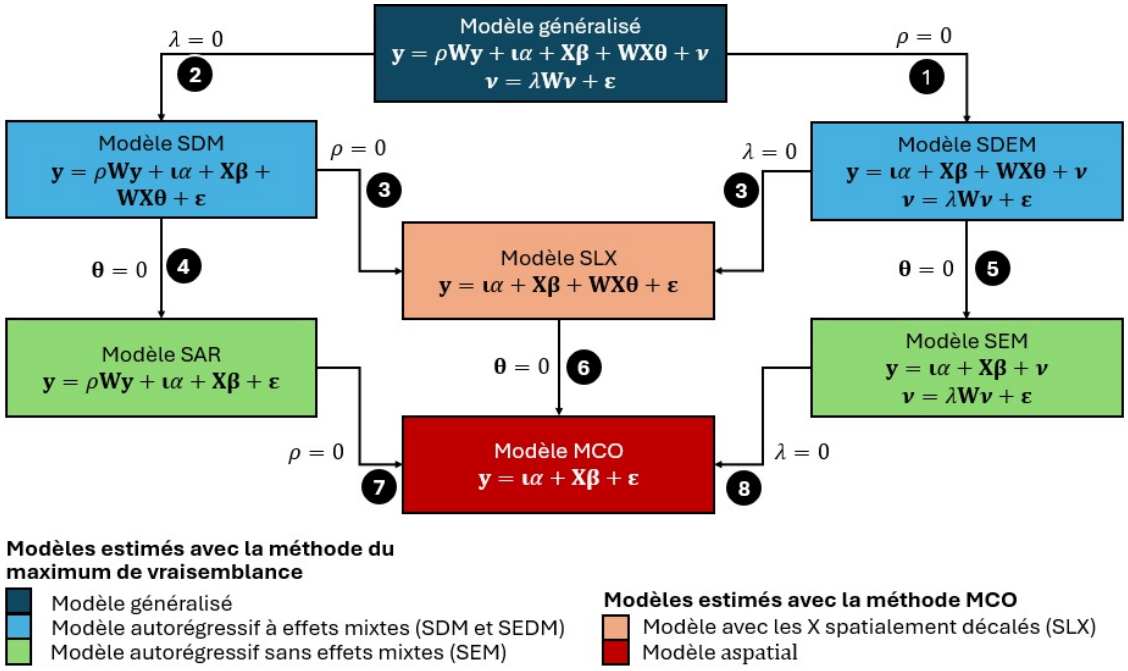


FIGURE 3.8 – Démarche à partir du modèle généralisé (modèle Manski)

Note

Description de la stratégie « général vers le particulier »
Globalement, la procédure est la suivante :
Étape A. Sélection d'un modèle mixte à partir des coefficients autorégressifs du modèle généralisé :

1. Si la valeur du coefficient ρ (rho) n'est pas significativement différente de 0 alors nous calculons le modèle mixte SDEM.
2. Si la valeur du coefficient λ (lambda) n'est pas significativement différente de 0, alors nous calculons le modèle mixte SDM.

Étape B. Sélection du modèle SLX

3. Si les valeurs des coefficients ρ du modèle SDM et λ du modèle SDEM ne sont pas significativement différentes de 0 alors nous retenons le modèle SLX.

Étape C. Sélection d'un modèle sans effets mixtes

4. Si la valeur du coefficient ρ du modèle SDM est significativement différente de 0, mais que tous les coefficients θ (theta) des variables indépendantes spatialement décalées $\mathbf{WX}$ ne sont pas différents de 0, alors nous retenons le modèle SAR.
5. Si la valeur du coefficient λ du modèle SDEM est significativement différente de 0, mais que tous les coefficients θ (theta) des variables indépendantes spatialement décalées $\mathbf{WX}$ ne sont pas différents de 0, alors nous retenons le modèle SEM.

Étape D. Choix du modèle MCO

6. Si tous les coefficients θ (theta) des variables indépendantes spatialement décalées $\mathbf{WX}$ du modèle SLX ne sont pas différents de 0, alors nous retenons le modèle aspatial (MCO).
7. Si le coefficient ρ (rho) du modèle SAR n'est pas significativement différent de 0, alors nous retenons le modèle aspatial (MCO).
8. Si le coefficient λ (lambda) du modèle SEM n'est pas significativement différent de 0, alors nous retenons le modèle aspatial (MCO).

Appliquons la stratégie « général vers le particulier » à notre jeu de données. Pour l'étape A, nous constatons que le coefficient lambda n'est pas significativement différent de 0 (λ = -0,028, p = 0,811), contrairement à celle de rho (ρ = 0,848, p < 0,001). Par conséquent, nous construisons uniquement le modèle SDM.

```
formule <- 'NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed'
# Étape A. Sélection d'un modèle mixte à partir
# des coefficients autorégressifs du modèle généralisé
manski <- sacsarlm(formule, listw = w_rook, data = LyonIris, type="sacmixed")

manski_lambda <- list(lambda = manski$lambda,
                      se     = manski$lambda.se,
                      z      = manski$lambda / manski$lambda.se,
                      p      = 2 * (1 - pnorm(abs(manski$lambda / manski$lambda.se))))

manski_rho <- list(rho = manski$rho,
                   se  = manski$rho.se,
```

```
                    z    = manski$rho / manski$rho.se,
                    p    = 2 * (1 - pnorm(abs(manski$rho / manski$rho.se))))

manski_resultats <- data.frame(Parametre = c("lambda", "rho"),
                               Coef      = c(manski_lambda$lambda, manski_rho$rho),
                               z         = c(manski_lambda$z, manski_rho$z),
                               p         = c(manski_lambda$p, manski_rho$p))
```

Tableau 3.2 – Résultats des coefficients autorégressifs du modèle généralisé

Paramètre	Coef	z	p
lambda	-0,028	-0,239	0,811
rho	0,848	22,832	0,000

Les résultats du modèle SDM montrent que :

1. Pour l'étape B, le coefficient rho ($\rho = 0{,}841$, p < 0,001) est différent de 0, alors nous écartons le modèle SLX.
2. Pour l'étape C, Plusieurs coefficients θ (theta) des variables indépendantes spatialement décalées $\mathbf{WX}$ ne sont pas différents de 0, notamment `lag.Pct_Img` et `lag.Pct_brevet`, alors nous écartons aussi le modèle SAR.
3. Au final, nous retenons donc le modèle SDM.

```
# Modèle Manski
manski <- sacsarlm(formule, listw = w_rook, data = LyonIris, type="sacmixed")

# Modèle SDM
SDM <- lagsarlm(formule, listw = w_rook, data = LyonIris, type = "mixed")
summary(SDM)
```

```
Call:lagsarlm(formula = formule, data = LyonIris, listw = w_rook,
    type = "mixed")

Residuals:
      Min        1Q    Median        3Q       Max
-12.60922  -1.77753  -0.43909   0.99252  18.15526

Type: mixed
Coefficients: (asymptotic standard errors)
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    8.1130457  2.5671301  3.1604 0.001576
Pct0_14       -0.0574046  0.0344908 -1.6643 0.096043
Pct_65        -0.0238715  0.0293647 -0.8129 0.416256
Pct_Img        0.0048364  0.0266560  0.1814 0.856025
Pct_brevet     0.0112746  0.0195259  0.5774 0.563656
```

```
NivVieMed      -0.1463876  0.0605853 -2.4162 0.015682
lag.Pct0_14    -0.1242574  0.0581170 -2.1381 0.032512
lag.Pct_65      0.0255480  0.0499646  0.5113 0.609125
lag.Pct_Img     0.1559952  0.0482138  3.2355 0.001214
lag.Pct_brevet -0.0883930  0.0342496 -2.5809 0.009856
lag.NivVieMed   0.1032469  0.0960201  1.0753 0.282257

Rho: 0.84127, LR test value: 492.38, p-value: < 2.22e-16
Asymptotic standard error: 0.023363
    z-value: 36.009, p-value: < 2.22e-16
Wald statistic: 1296.7, p-value: < 2.22e-16

Log likelihood: -1353.106 for mixed model
ML residual variance (sigma squared): 9.9845, (sigma: 3.1598)
Number of observations: 506
Number of parameters estimated: 13
AIC: 2732.2, (AIC for lm: 3222.6)
LM test for residual autocorrelation
test value: 0.0748, p-value: 0.78447
```

3.2.3 Comparaison des modèles mixtes et non mixtes

Avec les tests du ratio de vraisemblance sur les modèles spatiaux, nous pouvons vérifier dans R si le recours d'un modèle mixte est justifié comparativement à un modèle non mixte. Dans le code ci-dessous, nous vérifions si le modèle SDM est statistiquement différent du modèle SAR avec les fonctions `LR.Sarlm` et `anova`. Les résultats signalent un écart significatif des valeurs du log-vraisemblance (26,101, $p < 0{,}001$). Par conséquent, ce modèle mixte a un apport significatif.

```
## SDM et SEM sont-ils significativement différents?
LR.Sarlm(modele_SDM, modele_SAR)
```

```
    Likelihood ratio for spatial linear models

data:
Likelihood ratio = 26.101, df = 5, p-value = 8.528e-05
sample estimates:
Log likelihood of modele_SDM Log likelihood of modele_SAR
                   -1353.106                    -1366.157
```

```
anova(modele_SDM, modele_SAR)
```

```
           Model df    AIC  logLik Test L.Ratio    p-value
modele_SDM     1 13 2732.2 -1353.1    1
modele_SAR     2  8 2748.3 -1366.2    2  26.101 8.5283e-05
```

À l'inverse, la différence entre les valeurs du log-vraisemblance des modèles SDEM et SEM n'est pas significative (4,9728, p = 0,42), signalant que l'utilisation d'un modèle SDEM comparativement à un modèle SEM n'est pas nécessaire.

```
## SDEM et SEM sont-ils significativement différents?
LR.Sarlm(modele_SDEM, modele_SEM)
```

```
    Likelihood ratio for spatial linear models

data:
Likelihood ratio = 4.9728, df = 5, p-value = 0.4192
sample estimates:
Log likelihood of modele_SDEM  Log likelihood of modele_SEM
                    -1367.250                     -1369.737
```

```
anova(modele_SDEM, modele_SEM)
```

```
            Model df    AIC  logLik Test L.Ratio p-value
modele_SDEM     1 13 2760.5 -1367.2    1
modele_SEM      2  8 2755.5 -1369.7    2  4.9728  0.4192
```

3.2.4 Mesures AIC et BIC et dépendance spatiale

Bien que la procédure des tests précédente soit intéressante, elle peut néanmoins mener à une certaine difficulté dans l'identification de la forme fonctionnelle préférable. Un arbitrage peut alors être mené à partir de statistiques alternatives telles que les critères d'information.

Le critère d'information d'Akaike (AIC) et le critère d'information bayésien (BIC) sont largement utilisés pour évaluer la qualité d'ajustement du modèle. Plus leurs valeurs sont faibles, plus la spécification est intéressante pour expliquer la variabilité de la variable dépendante, et moins la variabilité du terme d'erreur est grande. Il est donc possible de comparer les valeurs des différents critères afin d'orienter le choix de la spécification finale (MCO, SLX, SAR, SEM, SDM et SDEM).

Nous pouvons aussi comparer l'autocorrélation spatiale des résidus des modèles avec le I de Moran.

```
## Valeurs d'AIC
AICs <- AIC(modele_MCO, modele_SLX, modele_SAR, modele_SEM,
            modele_SDM, modele_SDEM)

## Valeurs de BIC
BICs <- BIC(modele_MCO, modele_SLX, modele_SAR, modele_SEM,
            modele_SDM, modele_SDEM)

## Autocorrélation spatiale des résidus
imoran_mco <-  moran.mc(resid(modele_MCO), w_rook, nsim = 999)
```

```
imoran_slx <-  moran.mc(resid(modele_SLX), w_rook, nsim = 999)
imoran_slm <-  moran.mc(resid(modele_SAR), w_rook, nsim = 999)
imoran_sem <-  moran.mc(resid(modele_SEM), w_rook, nsim = 999)
imoran_sdm <-  moran.mc(resid(modele_SDM), w_rook, nsim = 999)
imoran_sdem <- moran.mc(resid(modele_SDEM), w_rook, nsim = 999)

imoran_s <- c(imoran_mco$statistic, imoran_slx$statistic,
              imoran_slm$statistic, imoran_sem$statistic,
              imoran_sdm$statistic, imoran_sdem$statistic)

imoran_p <- c(imoran_mco$p.value, imoran_slx$p.value,
              imoran_slm$p.value, imoran_sem$p.value,
              imoran_sdm$p.value, imoran_sdem$p.value)

## Tableau
Comparaison <- data.frame(Modele = c("MCO", "SLX", "SAR", "SEM", "SDM", "SDEM"),
                          AIC = AICs$AIC,
                          BIC = BICs$BIC,
                          dl = AICs$df,
                          MoranI = imoran_s,
                          MoranIp = imoran_p)
Comparaison
```

```
  Modele      AIC      BIC dl       MoranI MoranIp
1    MCO 3366.626 3396.212  7  0.587312061   0.001
2    SLX 3222.594 3273.313 12  0.604660275   0.001
3    SAR 2748.314 2782.126  8 -0.014281059   0.681
4    SEM 2755.474 2789.286  8 -0.011826605   0.634
5    SDM 2732.212 2787.157 13 -0.004612686   0.526
6   SDEM 2760.501 2815.446 13 -0.010361653   0.597
```

Quelques lignes de code suffisent pour créer deux graphiques permettant de comparer visuellement les résultats des différents modèles (figure 3.9).

```
library(ggplot2)
library(ggpubr)

## Graphique pour l'autocorrélation spatiale
g1 <- ggplot(data=Comparaison, aes(x=reorder(Modele,MoranI), y=MoranI)) +
      geom_segment(aes(x=reorder(Modele, MoranI),
                   xend=reorder(Modele, MoranI),
                   y=0, yend=MoranI)) +
  geom_point( size=4,fill="red",shape=21)+
  xlab("Modèle") + ylab("I de Moran")+
  labs(title="Autocorrélation spatiale des résidus",
```

```
       caption="Plus la valeur du I de Moran est faible, \nmoins il y a d'autocorrélation spatiale.")

## Graphique pour les valeurs d'AIC
g2 <- ggplot(data=Comparaison, aes(x=reorder(Modele,AIC), y=AIC)) +
  geom_segment(aes(x=reorder(Modele, AIC),
                   xend=reorder(Modele, AIC),
                   y=0, yend=AIC)) +
  geom_point( size=4,fill="red",shape=21)+
  xlab("Modèle") + ylab("AIC")+
  labs(title="Qualité d'ajustement du modèle",
       caption="Plus la valeur d'AIC est faible, \nplus le modèle est performant.")

## Figure avec les deux graphiques
ggarrange(g1, g2)
```

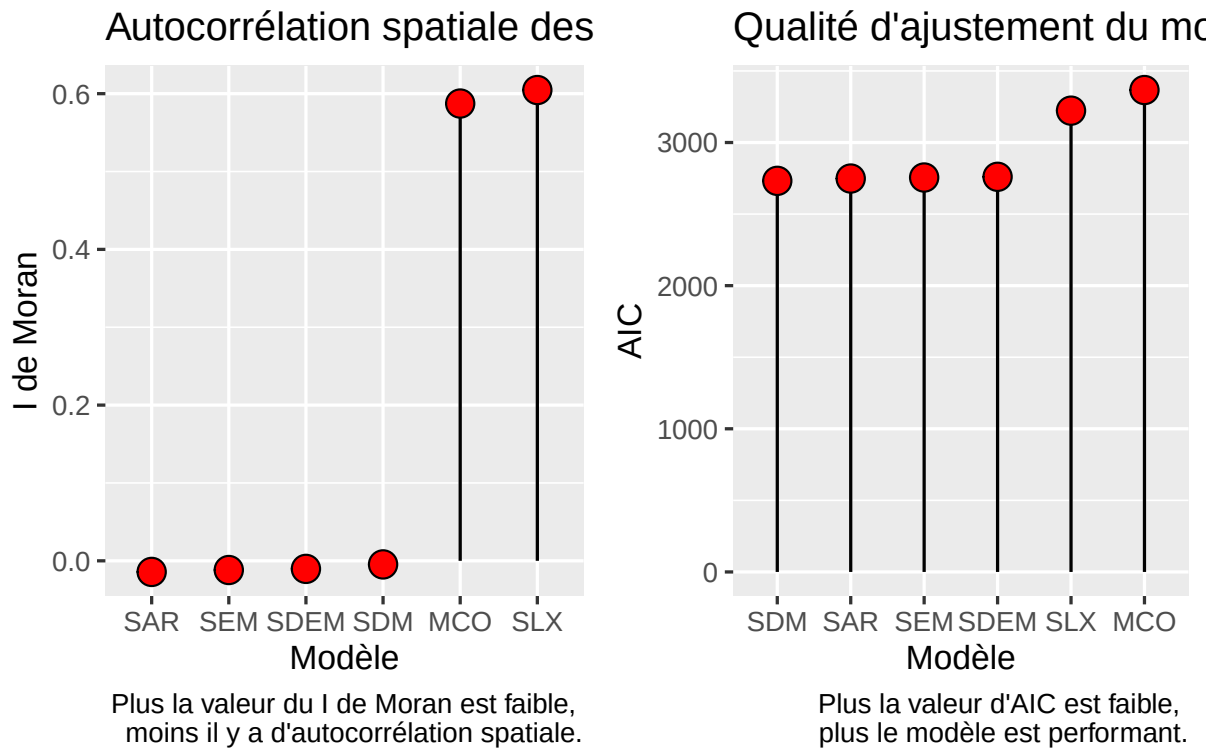


FIGURE 3.9 – Comparaison des différents modèles

Les résultats démontrent que :

- Les modèles MCO et SLX ont un problème de dépendance spatiale puisque leurs résidus sont significativement autocorrélés spatialement. Par conséquent, ils ne devraient pas être retenus.
- Les modèles SDM, SAR et SEM sont les plus performants avec les valeurs d'AIC les plus faibles.

Astuce

Quelques lignes de code pour tous les modèles!

```
formule <- 'NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed'
MCO     <- lm(formule, data = LyonIris)
SLX     <- lmSLX(formule, listw = w_rook, data = LyonIris)
SDM     <- lagsarlm(formule, listw = w_rook, data = LyonIris, type = "mixed")
SDEM    <- errorsarlm(formule, listw = w_rook, data = LyonIris, etype = 'emixed')
SAR     <- lagsarlm(formule,listw = w_rook, data = LyonIris, type = 'lag')
SEM     <- errorsarlm(formule, listw = w_rook, data = LyonIris)
Manski  <- sacsarlm(formule, listw = w_rook, data = LyonIris, type="sacmixed")
```

3.3 Quiz de révision

Questions

- **Dans un modèle SLX, la matrice de pondération spatiale est appliquée au niveau de :**
  - Variable dépendante
  - Variables indépendantes
  - Terme d'erreur
  - Variable dépendante et variables indépendantes
  - Variable dépendante et terme d'erreur

  Relisez au besoin la section 3.1.1.
- **Dans un modèle SAR, la matrice de pondération spatiale est appliquée au niveau de :**
  - Variable dépendante
  - Variables indépendantes
  - Terme d'erreur

  Relisez au besoin la section 3.1.2.
- **Dans un modèle SEM, la matrice de pondération spatiale est appliquée au niveau de :**
  - Variable dépendante
  - Variables indépendantes
  - Terme d'erreur

  Relisez au besoin la section 3.1.3.
- **Dans un modèle mixte SDM, la matrice de pondération spatiale est appliquée au niveau de :**
  - Variable dépendante
  - Variables indépendantes
  - Terme d'erreur

  Relisez au besoin la section 3.1.4.
- **Dans un modèle SDEM, la matrice de pondération spatiale est appliquée au niveau de :**
  - Variable dépendante

  - Variables indépendantes
  - Terme d'erreur

  Relisez au besoin la section 3.1.5.
- **Quelle est la procédure popularisée par Luc Anselin pour discriminer entre les spécifications SAR, SEM et MCO?**

  - Tests du multiplicateur de Lagrange (LM) sur le modèle MCO
  - Mesures AIC et BIC
  - I de Moran
  - Tests du ratio de vraisemblance sur les modèles spatiaux

  Relisez au besoin la section 3.2.1.

Réponses

- Dans un modèle SLX, la matrice de pondération spatiale est appliquée au niveau de :
  - Variables indépendantes
- Dans un modèle SAR, la matrice de pondération spatiale est appliquée au niveau de :
  - Variable dépendante
- Dans un modèle SEM, la matrice de pondération spatiale est appliquée au niveau de :
  - Terme d'erreur
- Dans un modèle mixte SDM, la matrice de pondération spatiale est appliquée au niveau de :
  - Variable dépendante
  - Variables indépendantes
- Dans un modèle SDEM, la matrice de pondération spatiale est appliquée au niveau de :
  - Variables indépendantes
  - Terme d'erreur
- Quelle est la procédure popularisée par Luc Anselin pour discriminer entre les spécifications SAR, SEM et MCO?
  - Tests du multiplicateur de Lagrange (LM) sur le modèle MCO

3.4 Exercices de révision

Exercice

Exercice 1. Réalisation de modèles de régression autorégressifs spatiaux

```
library(sf)
library(spatialreg)
# Matrice de contiguïté selon le partage d'un segment (Rook)
load("data/Lyon.Rdata")
Rook <- poly2nb(LyonIris, queen=FALSE)
w_rook <- nb2listw(Rook, zero.policy=TRUE, style = "W")
# Modèles
formule <- "PM25 ~  Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed"
# Modèles MOC, SLX, SDM, SDEM, SAR, SEM, Manski
à compléter
```

Correction à la section 11.3.1.

Exercice

Exercice 2. Tests du multiplicateur de Lagrange (LM) sur le modèle MCO

```
à compléter
```

Correction à la section 11.3.2.

Exercice

Exercice 3. Mesures AIC et BIC et dépendance spatiale

```
## Valeurs d'AIC et de BIC
à compléter
## Autocorrélation spatiale des résidus
à compléter
## Tableau
Comparaison <- à compléter
Comparaison
```

Correction à la section 11.3.3.

4 Modèles probit spatiaux pour une variable dépendante dichtonomique (en cours de rédaction)

Dans ce chapitre, nous décrirons les modèles probit spatiaux, très utiles lorsque la variable dépendante est une variable qualitative binaire.

Objectif

Objectifs d'apprentissage visés dans ce chapitre
À la fin de ce chapitre, vous devriez être en mesure de :

- comprendre les raisons motivant le choix d'un modèle d'économétrie spatiale avec une variable dépendante qualitative binaire;
- mettre en pratique et interprétrer les résultats de ces deux types modèles spatiaux dans R.

Package

Liste des *packages* utilisés dans ce chapitre

- Pour importer et manipuler des fichiers géographiques :
  - `sf` pour importer et manipuler des données vectorielles.
- Pour construire des cartes et des graphiques :
  - `tmap` est certainement le meilleur *package* pour la cartographie.
  - `ggplot2` et `ggpubr` pour construire des graphiques.
- Pour construire des modèles spatiaux :
  - `spdep` pour construire des matrices de pondération spatiales et calculer le I de Moran.
  - `Matrix` pour construire des matrices de pondération spatiales peu denses.
  - `ProbitSpatial` pour construire des modèles économétriques spatiaux avec variable dépendante binaire.

4.1 Bref retour sur le modèle probit

4.2 Les différents modèles probit spatiaux

4.3 Quiz de révision

4.4 Exercices de révision

Exercice

Exercice 1. À compléter
Complétez le code ci-dessous.
Correction à la section 11.4.1.

Exercice

Exercice 2. À compléter
Complétez le code ci-dessous.
Correction à la section 11.4.2.

Exercice

Exercice 3. À compléter
Complétez le code ci-dessous.
Correction à la section 11.4.3.

5 Modèles d'économétrie spatiale en panel (en cours de rédaction)

Dans ce chapitre, nous abordons une extension des modèles autorégressifs, soit les modèles spatiaux par panel qui permettent de modéliser des données spatiales longitudinales.

Objectif

Objectifs d'apprentissage visés dans ce chapitre

À la fin de ce chapitre, vous devriez être en mesure de :

- comprendre les différentes formulations des modèles spatiaux par panel (SLPDM, SEPDM et SEPDM);
- assimiler les principes fondamentaux de ces différentes modèles;
- identifier le modèle spatial par panel le plus approprié (SLPDM, SEPDM et SEPDM);
- analyser les résultats produits par ces différentes modèles;
- Mettre en pratique ces modèles spatiaux par panel dans R.

Package

Liste des *packages* utilisés dans ce chapitre

- Pour importer et manipuler des fichiers géographiques :
  - `sf` pour importer et manipuler des données vectorielles.
- Pour construire des cartes et des graphiques :
  - `tmap` pour construire des cartes thématiques.
  - `ggplot2` est un *package* pour construire des graphiques.
- Pour les régressions :
  - `spdep` pour construire des matrices spatiales et calculer des mesures d'autocorrélation spatiale.
  - `plm` pour construire des modèles de régression par panel.
  - `splm` pour construire des modèles de régression spatiale par panel.

5.1 Bref retour sur les modèles en panel

5.2 Formulation des différents modèles spatiaux par panel

5.2.1 Description des différents modèles

5.2.2 Modèle SLPDM : autocorrélation sur la variable dépendante

5.2.3 Modèle SEPDM : autocorrélation sur le terme d'erreur

5.2.4 Modèle SEPDM : autocorrélation sur la variable dépendante et les variables indépendantes

5.3 Sélection du modèle spatial par panel le plus approprié

5.4 Mise en œuvre dans R

5.5 Quiz de révision

5.6 Exercices de révision

Exercice

Exercice 1. À compléter
Complétez le code ci-dessous.

```
library(sf)
library(tmap)
```

Correction à la section 11.5.1.

Exercice

Exercice 2. À compléter
Complétez le code ci-dessous.

```
library(sf)
library(tmap)
```

Correction à la section 11.5.2.

Exercice

Exercice 3. À compléter
Complétez le code ci-dessous.

```
library(sf)
library(spatstat)
library(tmap)
```

Correction à la section 11.5.3.

Partie 4. Variable latente spatiale : lissage et filtrage spatial

6 Modèles généralisés additifs

Dans ce chapitre, nous abordons deux formes de modèles généralisés additifs (*Generalized additive model* – GAM) qui permettent d'introduire l'espace de deux manières différentes : les modèles GAM avec une spline bivariée avec les coordonnées géographiques (x, y) pour capturer les variations continues dans l'espace; les modèles GAM avec un lissage par champ aléatoire de Markov (*Markov random field* – MRF) pour modéliser la dépendance spatiale entre les unités spatiales voisines.

Objectif

Objectifs d'apprentissage visés dans ce chapitre
À la fin de ce chapitre, vous devriez être en mesure de :

- comprendre pourquoi utiliser un modèle GAM avec une spline bivariée sur les coordonnées géographiques ou avec un lissage par champ aléatoire de Markov;
- analyser les résultats produits par ces deux types de modèles GAM;
- mettre en pratique ces modèles dans R.

Package

Liste des *packages* utilisés dans ce chapitre

- Pour importer et manipuler des fichiers géographiques :
  - `sf` pour importer et manipuler des données vectorielles.
- Pour mesurer l'autocorrélation :
  - `spdep` pour construire des matrices de pondération spatiales et calculer le I de Moran.
- Pour construire des cartes et des graphiques :
  - `ggplot2` pour construire des graphiques.
  - `tmap` est certainement le meilleur *package* pour la cartographie.
- Pour construire des modèles GAM :
  - `mgcv`, le *package* de référence pour ajuster des modèles GAM. De nombreux autres *packages* proposent des extensions pour ce type de modèle et se basent principalement sur les outils mis à disposition par `mgcv` pour créer des bases et leurs matrices de pénalisation (`gamm4`, `brms`, `gamlss`, `qgam`, etc.).
  - `DHARMa`, un *package* permettant d'effectuer des diagnostics sur des résidus de modèles GLM.

6.1 Bref retour sur les GAM

Les modèles généralisés additifs (GAM) constituent une famille de modèles de régression très flexibles permettant notamment de modéliser des relations non paramétriques entre les variables indépendantes et la variable dépendante.

Dans un modèle de régression classique, nous partons du principe que les variables $\mathbf{X}$ (indépendantes) ont un impact linéaire sur la variable $\mathbf{Y}$ (dépendante). Cette hypothèse simplifie grandement les relations entre les variables observées

et permet d'interpréter les résultats plus facilement (analyse des coefficients de régression). Or, cette hypothèse peut être fausse et les relations non linéaires ont plus tendance à être la norme que l'exception.

Dans un GAM, plutôt que modéliser une relation linéaire, nous partons du principe que la relation modélisée peut être non linéaire et suivre une fonction f. Il s'agit alors de relations non paramétriques, car la forme de cette fonction s'ajuste aux données et ne suit pas une forme prédéterminée.

Ces fonctions f sont le plus souvent estimées avec des splines. Le fonctionnement de base est assez simple. Pour chaque variable $\mathbf{X}$ devant suivre une relation non linéaire, la variable est remplacée par k sous-variable (x_1, x_2, …, x_k). k représente le niveau de complexité attendu de la relation : plus il est grand, plus l'ajustement de la fonction est complexe. Le modèle estime un coefficient pour chacune de ces sous-variables, et la somme des effets de ces sous-variables permet de reconstruire la relation non linéaire.

La figure 6.1 permet d'illustrer ce principe. Les lignes grises représentent les sous-variables, aussi appelées bases, qui sont ensuite multipliées par leur coefficient respectif pour obtenir leurs effets (en couleur), puis la ligne bleue représente leur somme. La ligne bleue capture très efficacement la relation entre les variables $\mathbf{X}$ et $\mathbf{Y}$.

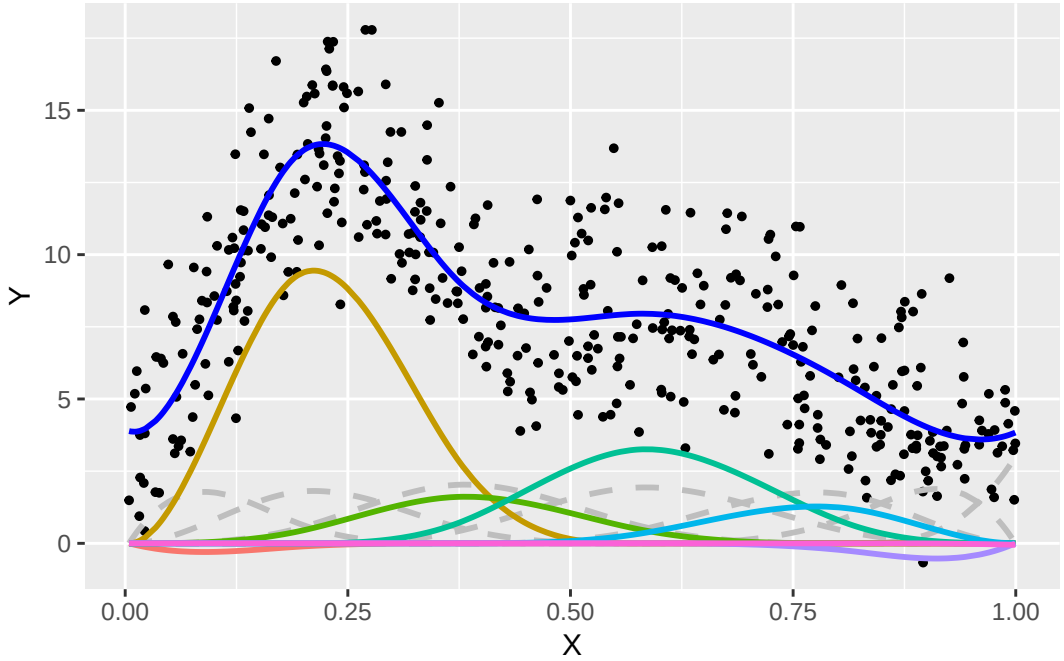


Figure 6.1 – Exemple de spline

⚠ **Attention**

Modèles généralisés additifs
Pour une explication détaillée du fonctionnement des splines, vous pouvez consulter le chapitre sur les GAM du livre *Méthodes quantitatives en sciences sociales : un grand bol d'R* (Apparicio et Gelb 2022).

6.2 Comment utiliser une spline pour ajouter l'espace dans un modèle GAM?

Nous avons vu qu'une spline permet de modéliser l'impact potentiellement non linéaire d'une variable indépendante sur une variable dépendante. Or, il est possible d'utiliser des splines pour intégrer un effet spatial dans un modèle, et ce, en utilisant un type de spline particulier : la spline bivariée qui capture l'effet simultané de deux variables indépendantes sur une variable dépendante.

Dans un contexte spatial, une spline bivariée capture la façon dont un changement dans les coordonnées X et Y affecte la variable dépendante. De plus, compte tenu de la forme des fonctions de base utilisées, les splines sont particulièrement adaptées pour capturer des relations spatiales avec une forte autocorrélation spatiale positive.

Prenons un exemple avec un jeu de données fictif représentant des mesures de pollution relevées à différentes localisations sur un territoire (figure 6.2).

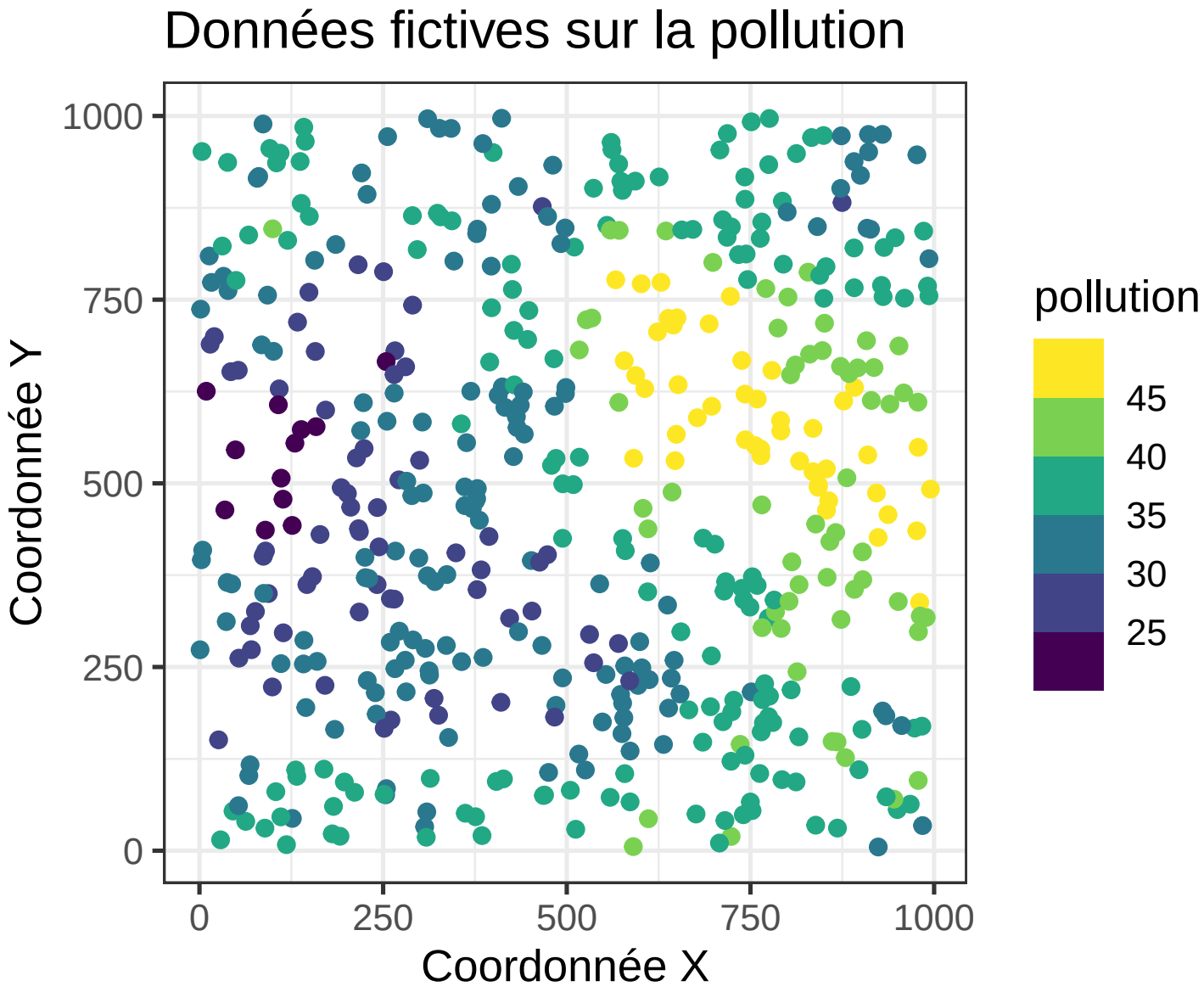


Figure 6.2 – Exemple d'un jeu de données fictif avec une forte dépendance spatiale

Nous pourrions créer des bases d'une spline qui varient à la fois dans les coordonnées spatiales X et Y. L'animation à la figure 6.3 illustre des bases obtenues (k = 25) pour les données ci-dessus. La hauteur dans la figure est utilisée pour représenter la variation originale des bases qui sera ensuite ajustée aux données grâce aux coefficients du modèle de régression.

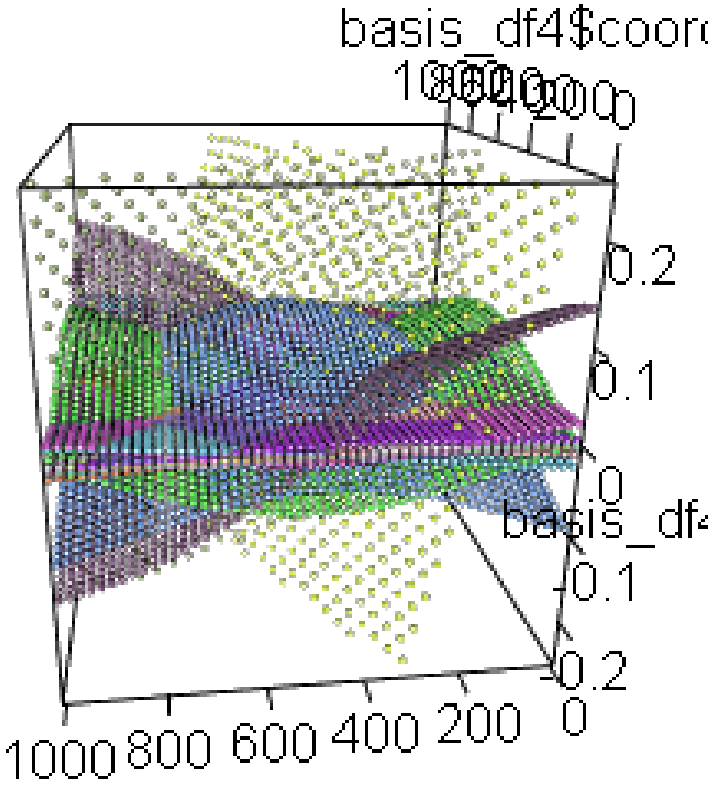


Figure 6.3 – Représentation des bases d'une spline bivariée

En ajustant un coefficient pour chaque base et en combinant leurs effets, nous obtenons un effet spatial permettant de prédire la variable de pollution, cartographié à la figure 6.4. Nous constatons ainsi que le modèle a été capable de reproduire les tendances principales des niveaux de pollution en retrouvant les secteurs avec des valeurs fortes et faibles. Plus l'autocorrélation spatiale est élevée, plus ce type de modèle est efficace pour capturer l'effet spatial.

D'un point de vue plus formel, un modèle GAM peut être décrit par l'équation suivante :

$$\begin{aligned} y &\sim D(\mu, \theta) \\ g(\mu) &= \beta_0 + \beta X + \sum_{j=1}^{n} (f_j(\zeta_j)) \end{aligned} \tag{6.1}$$

avec :

- y, la variable dépendante.
- D, une distribution avec une espérance μ et ses autres paramètres θ.
- X, les variables indépendantes dont l'effet est supposé linéaire par le modèle.

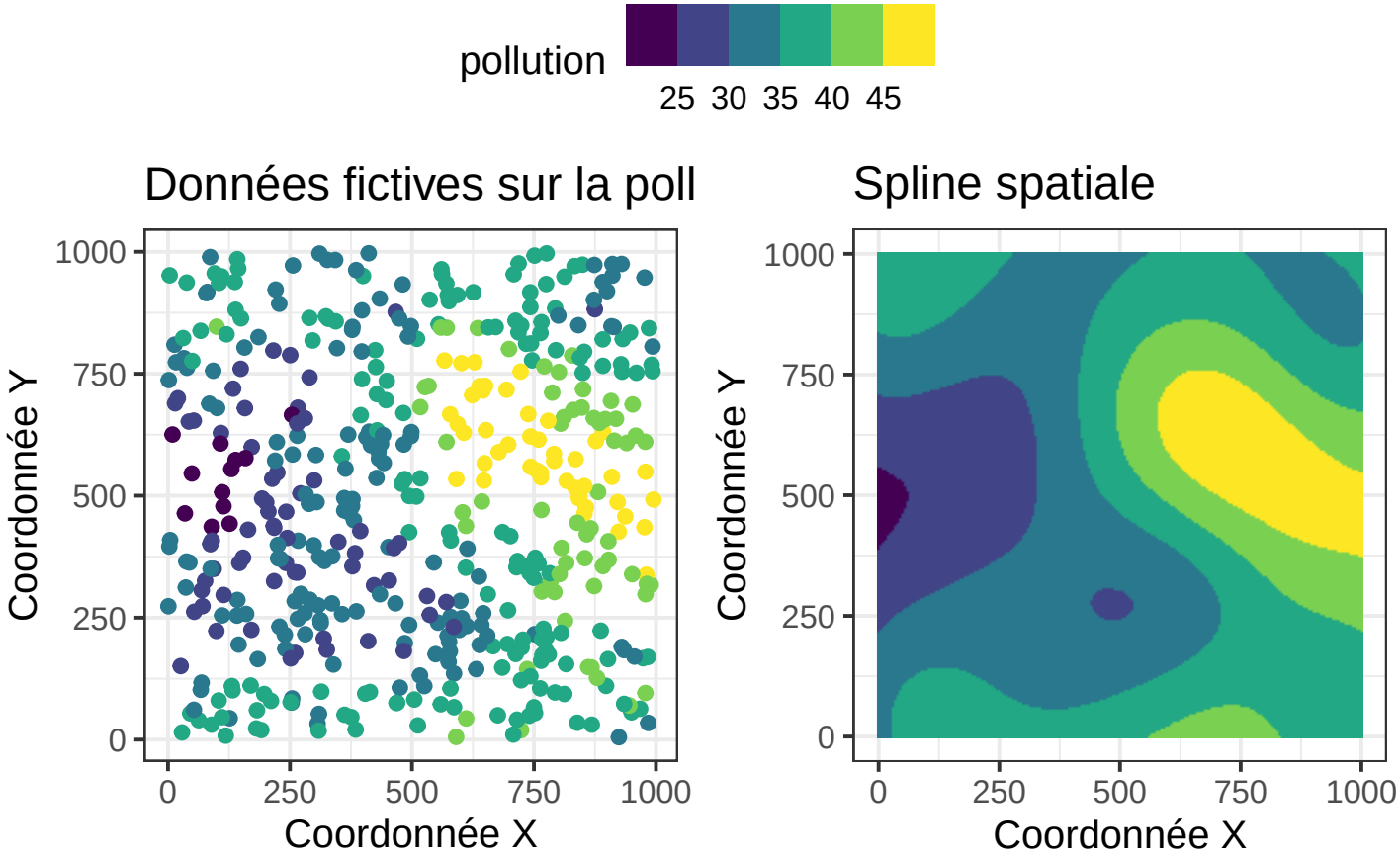


FIGURE 6.4 – Captation de l'effet spatial par une spline bivariée

- β, les coefficients des variables indépendantes.
- β_0, la constante.
- ζ, les variables dont l'effet est supposé non linéaire par le modèle.
- f_j, une fonction modélisant l'impact de la variable ζ_j.

Pour un modèle incluant seulement une spline bivariée sur les coordonnées géographiques des observations, le modèle peut se résumer à l'équation suivante :

$$\begin{aligned} y &\sim D(\mu, \theta) \\ g(\mu) &= \beta_0 + \beta X + f(sp) \end{aligned} \tag{6.2}$$

avec sp une matrice comprenant deux colonnes, soit les coordonnées cartésiennes des observations.

6.3 Pourquoi recourir à un modèle GAM?

Les modèles GAM sont très flexibles puisqu'ils sont une extension directe des modèles linéaires généralisés (GLM). Ils peuvent donc être appliqués à des variables dépendantes suivant de nombreuses distributions (normale, gamma, binomiale, Poisson, Tweedie, log-normal, etc.). Cependant, ils imposent une conceptualisation particulière de l'espace. En effet, un modèle GAM n'inclut pas directement la notion de dépendance spatiale dans sa formulation. L'hypothèse d'indépendance des observations est conservée (contrairement à un modèle des moindres carrés généralisés, GLS). Le modèle intègre simplement un terme flexible correspondant à l'ajout d'un ensemble de nouvelles variables $\mathbf{X}$ structurées afin de capturer un effet avec une autocorrélation spatiale positive.

Si nous reprenons l'exemple de la pollution atmosphérique, le modèle est un simple GLM dans lequel nous supposons que les niveaux de pollution suivent une distribution normale et que les observations sont indépendantes. Cependant, la moyenne de cette distribution peut être prédite avec une spline calculée à partir des coordonnées spatiales des observations. Le modèle ne part donc pas du principe que deux observations proches sont plus similaires, mais plutôt qu'il existe un effet spatial permettant de prédire les niveaux moyens de pollution attendus. Un modèle GAM ne vise donc pas explicitement à corriger l'effet de la dépendance spatiale, mais plutôt à capturer un effet spatialement structuré et non capturé par les variables indépendantes $\mathbf{X}$. Sans l'ajout de la spline, cet effet se serait retrouvé dans les résidus du modèle.

La cartographie de l'effet spatial dans un modèle GAM permet de visualiser les secteurs de l'espace d'étude dans lesquels, toutes choses étant égales par ailleurs, la variable dépendante est plus forte ou plus faible que la moyenne régionale. Cet effet spatial peut être vu comme un ajustement local de la prédiction. Une forme de malus ou de bonus que le modèle ajoute à sa prédiction pour tenir compte de la variation spatiale de la variable dépendante qui n'a pas été expliquée par les variables indépendantes.

Puisqu'un modèle GAM ne prend pas directement en compte l'autocorrélation spatiale, il est tout à fait possible qu'il échoue à réduire suffisamment la dépendance spatiale des résidus d'un modèle global (non spatial). Il faudra alors privilégier d'autres formes de modèles. De même, si notre cadre théorique nous amène à penser que nos observations s'influencent mutuellement dans l'espace, il sera beaucoup plus cohérent d'utiliser un modèle économétrique (chapitre 2).

Prenons un exemple concret pour lequel l'utilisation d'un modèle GAM serait particulièrement pertinente. Dans une étude récente, Gelb et Apparicio (2022) cherchent à modéliser les niveaux d'exposition de cyclistes aux pollutions atmosphériques et sonores. Les données utilisées ont été collectées par des cyclistes qui parcouraient différents environnements urbains

équipés de capteurs de bruit et de pollution. Plus spécifiquement, l'objectif de l'étude était de déterminer quelles caractéristiques de l'environnement urbain réduisent ou augmentent significativement les niveaux d'exposition des cyclistes. Dans leur cadre théorique (figure 6.5), les auteurs distinguent :

1. Les effets du micro-environnement, causés par les caractéristiques locales comme le type de rue utilisé, la densité bâtie, la présence de végétation, etc.
2. La pollution d'arrière-plan, causée par des phénomènes géographiques régionaux comme la localisation des industries et des autoroutes, la topographie, etc.

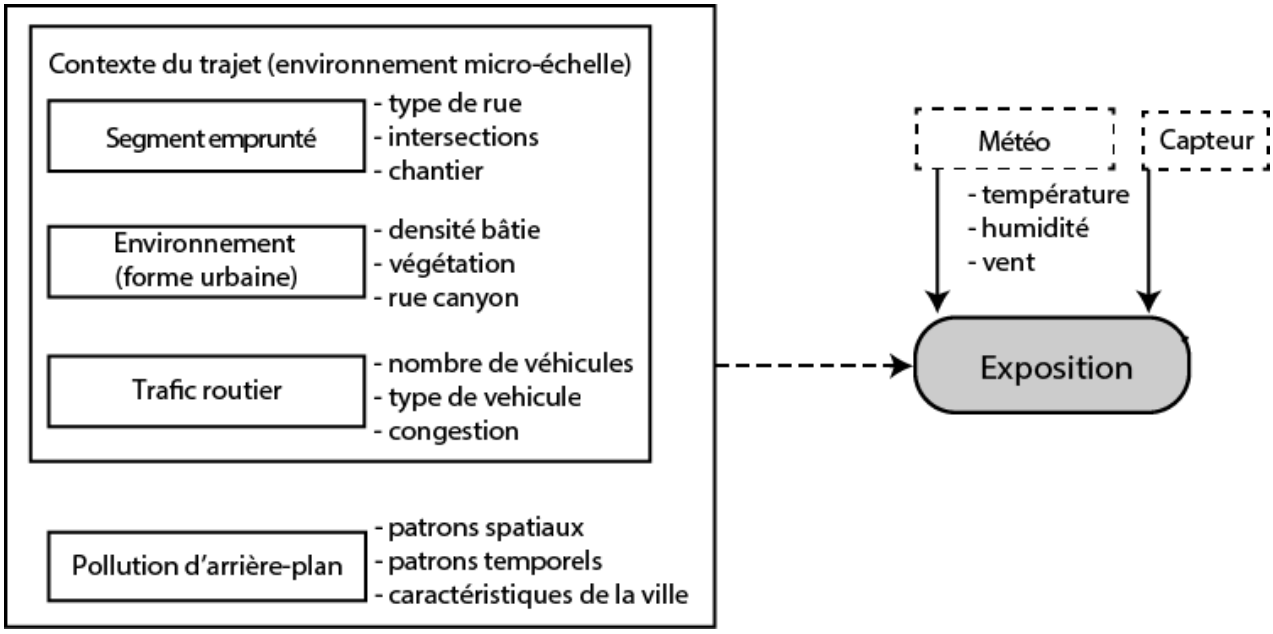


Figure 6.5 – Cadre théorique de l'exposition des cyclistes

L'exposition mesurée pour un cycliste à un endroit t dépend donc des caractéristiques propres de cet endroit, mais aussi de la pollution d'arrière-plan. Par exemple, si nous prenons une rue A, en tous points identique à une rue B, nous pouvons nous attendre à mesurer des niveaux de pollution plus élevés pour B si celle-ci est située dans un secteur plus industrialisé de la ville. Cette différence ne s'explique pas par des caractéristiques propres à A ou à B, puisqu'elles sont identiques, mais bien par un effet d'arrière-plan produit par différents phénomènes géographiques et donc spatialement autocorrélé. Par conséquent, dans cet exemple, l'intégration de la pollution d'arrière-plan avec une spline spatiale est particulièrement pertinente, car l'interprétation de l'effet produit par cette spline s'intègre directement dans le cadre théorique.

6.4 Markov random field (MRF) ou spline bivariée?

Nous avons vu jusqu'ici qu'il est possible d'incorporer un effet spatial dans un modèle GAM en intégrant une spline bivariée sur les coordonnées spatiales des observations. Cette approche est cependant moins pertinente quand les observations sont des polygones de tailles variées. En effet, intégrer les centroïdes des polygones dans le modèle revient à nettement déformer l'espace de nos données puisque des polygones voisins peuvent avoir des centroïdes très éloignés.

Lorsque les données ont une géométrie irrégulière et que le voisinage est plus important que la proximité (distance), il est recommandé d'utiliser un autre type de fonction appelée un champ de Markov aléatoire (*Markov random field, MRF*).

🚀 Aller plus loin

Modèle CAR versus GAM avec MRF
Mathématiquement parlant, un modèle de type CAR (*conditionnal autoregressive model*) est très proche d'un modèle GAM intégrant un MRF puisque la structure spatiale intégrée par un modèle CAR est un MRF : la valeur de la variable dépendante d'une entité est **conditionnée** par la valeur de ses voisins.

Dans l'équation 6.3, le terme η représente l'effet spatial. Il est le plus souvent modélisé comme un effet provenant d'une distribution normale multivariée, centrée sur zéro et dont la matrice de covariance est calculée grâce à un terme λ et à une matrice de pondération spatiale W. Ce type de modèle est communément appelé un modèle à effet mixte, car il combine des effets **fixes** (coefficients β) et des effets **aléatoires** (λ). Les effets aléatoires ne sont pas estimés de la même façon que les effets fixes : ils sont en fait pénalisés. Pour une introduction sur les modèles à effets mixtes, vous pouvez consulter ce chapitre sur les GLMM du livre *Méthodes quantitatives en sciences sociales : un grand bol d'R* (Apparicio et Gelb 2022).

$$
\begin{aligned}
y &\sim D(\mu, \theta) \\
g(\mu) &= \beta_0 + \beta X + \eta \\
\eta &\sim Normal(0, (I - \lambda W)\sigma^2)
\end{aligned}
\tag{6.3}
$$

Il existe un lien direct entre les modèles à effets mixtes et les GAM. En effet, il est possible de reformuler un GLMM comme un GAM et vice-versa, ce qui signifie qu'une spline pénalisée peut être utilisée pour représenter un effet aléatoire (Wood 2017).

Dans R, avec le *package* `mgcv`, il est possible d'utiliser une spline pénalisée par une matrice de pondération spatiale W pour approximer l'effet aléatoire que nous aurions modélisé avec un modèle de type CAR. L'intérêt d'utiliser un GAM plutôt qu'un CAR est que nous pouvons contrôler directement le degré de lissage (autocorrélation spatiale) du terme spatial dans le modèle grâce au nombre de degrés de liberté (k) de la spline. En utilisant un nombre de degrés de liberté maximal ($k = n - 1$), soit une base par observation, alors nous obtenons un *full rank MRF*, autorisant de facto la plus grande variance possible du terme spatial. En réduisant k, nous obtenons des modèles d'ordre inférieur (*lower rank MRF*) dont le terme spatial est plus lisse et moins complexe.

$$
\begin{aligned}
y_i &\sim D(\mu_i, \theta) \\
g(\mu_i) &= \beta_0 + \beta X_i + f(W_i)
\end{aligned}
\tag{6.4}
$$

6.5 Mise en œuvre et analyse dans R

Pour illustrer la réalisation des différents modèles GAM (non spatial, avec une spline bivariée sur les coordonnées géographiques et avec une spline spatiale de type MRF), nous utilisons le jeu de données sur les parts modales dans la région métropolitaine de Montréal (section 1.1.2). Ce jeu de données comprend trois variables mesurant respectivement la proportion des individus utilisant la voiture, le transport collectif ou un mode actif (marche ou vélo) comme mode de transport principal pour leurs déplacements domicile-travail en 2021 pour les secteurs de recensement (figure 1.2).

Lorsqu'un ensemble de variables sont des proportions d'un total (leur somme donne 1 ou 100), alors l'ensemble de ces variables est désigné comme des données compositionnelles. Ces données doivent être analysées avec des méthodes spécifiques, car elles ont un comportement particulier. La diminution de l'une d'entre elles provoque nécessairement l'augmentation d'une autre. Elles ne sont donc pas indépendantes et certaines catégories ont même tendance à exister simultanément ou au contraire à se repousser. L'analyse de données compositionnelles est un domaine en soit et sort du matériel que nous souhaitons couvrir dans ce livre. Il existe d'excellentes ressources et *packages* R sur le sujet (Boogaart et Tolosana-Delgado 2013; van den Boogaart, Tolosana-Delgado et Bren 2024; Tsagris et Athineou 2025).

Nous utilisons ici l'approche du log des ratios (*alr*). Il s'agit d'une transformation qui peut être appliquée aux données compositionnelles pour pouvoir les analyser avec des méthodes classiques de statistiques. Une autre méthode de transformation est habituellement privilégié, soit celle isométrique des log ratios (*ilr*), mais cette dernière produit des résultats plus difficiles à interpréter (Greenacre, Martinez-Alvaro et Blasco 2021). Nous utiliserons donc l'approche *alr* dans cet exemple. La transformation *alr* est décrite par la formule suivante :

$$\text{alr}(x) = \left[\log \frac{x_1}{x_D}, \cdots, \log \frac{x_{D-1}}{x_D}\right] \tag{6.5}$$

Pour une variable compositionnelle *x* comprenant *D* dimensions (*catégories*), la transformation *alr* consiste à calculer *D-1* log de ratio entre les différentes dimensions *D* et une référence parmi *D*. Ainsi, le nombre de dimensions de la version transformée de *x* est toujours de *D-1*.

Dans cet exemple, nous utilisons le ratio entre les personnes utilisatrices du transport collectif (TC) et de l'automobile. Puisque la transformation *alr* implique une division, puis un log, il est important que ni le numérateur ni le dénominateur ne soit égal à 0. Or, nous avons quelques secteurs de recensement (SR) avec des parts modales TC à 0. Puisqu'il semble peu probable qu'aucune personne dans un SR n'utilise le transport collectif, nous remplaçons les 0 par le plus faible pourcentage trouvé dans les SR avec la valeur supérieure à 0.

```
library(sf, quietly = TRUE)
library(ggplot2, quietly = TRUE)
library(tmap, quietly = TRUE)
library(dplyr, quietly = TRUE)
library(ggpubr, quietly = TRUE)

# Chargement des données
load("data/Mtl.Rdata")

# Nous préparons ici les données en calculant les ratios
# remplaçant les 0 et calculant la transformation alr
data_mtl <- data_mtl %>%
  mutate(
    prt_tc = ifelse(prt_tc == 0, min(prt_tc[prt_tc>0]) ,prt_tc)
  ) %>%
  mutate(
    ratio = (prt_tc / prt_auto),
    alr_val = log(prt_tc / prt_auto)
  )
```

```
plot1 <- ggplot(data_mtl) +
  geom_histogram(aes(x = ratio), fill = 'grey', color = 'black', bins = 30) +
  theme_bw() +
  labs(x = 'part modale TC / part modale auto',
       y = 'effectif'
       )

plot2 <- ggplot(data_mtl) +
  geom_histogram(aes(x = alr_val), fill = 'grey', color = 'black', bins = 30) +
  theme_bw() +
  labs(x = 'log(part modale TC / part modale auto)',
       y = 'effectif'
       )

ggarrange(plot1, plot2, nrow = 2)
```

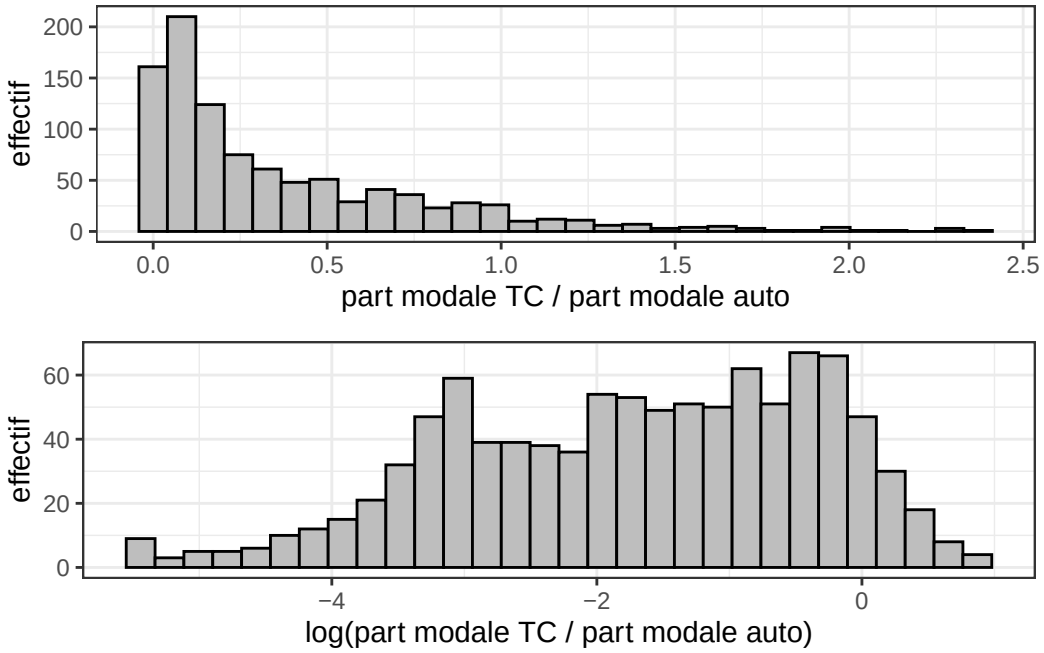


FIGURE 6.6 – Distributions du ratio et du log ratio entre les parts modales TC et automobile

La figure 6.6 illustre la distribution de la variable que nous souhaitons modéliser. Le simple ratio a des valeurs comprises entre 0 et 2,5. Une valeur de 0 signifie que personne dans le secteur de recensement n'utilise le transport collectif comme mode principal pour ses déplacements pendulaires. Une valeur de 1 indique que cette proportion est égale à celle des personnes utilisant la voiture. Une valeur plus grande que 1 indique une dominance du mode TC plutôt que de l'automobile. Sans surprise, pour la majorité des secteurs de recensement, la valeur est inférieure à 1. Le log du ratio suit une distribution moins asymétrique qui sera probablement plus facile à modéliser.

```
library(spdep, quietly = TRUE)
library(dplyr, quietly = TRUE)

# Matrice de de contiguïté selon le partage d'un nœud
queen_nb <- poly2nb(data_mtl, queen = TRUE)
queen_w <- nb2listw(queen_nb, style = 'W', zero.policy = TRUE)

# Calcul du I de moran de la variable ratio
moran_i <- moran.mc(data_mtl$alr_val,
                    listw = queen_w,
                    zero.policy = TRUE, nsim = 999)

# Texte pour la valeur du I de Moran
moran_text <- format(as.numeric(round(moran_i$statistic,3)) , decimal.mark =",")

# Cartographie de la variable
tm_shape(data_mtl) +
  tm_fill(col="alr_val", n = 7,
          style = "jenks",
          midpoint = log(1),
          legend.format = list(text.separator = "à",
                               decimal.mark = ",",
                               big.mark = " ",
                               digits = 2),
          palette = "-RdBu", title = 'part modale TC / part modale auto') +
tm_layout(frame=FALSE, legend.outside = TRUE) +
tm_scale_bar(breaks = c(0,10)) +
tm_xlab(paste0('I de Moran = ', moran_text,
               " (matrice de contiguïté selon le partage d'un nœud)."))
```

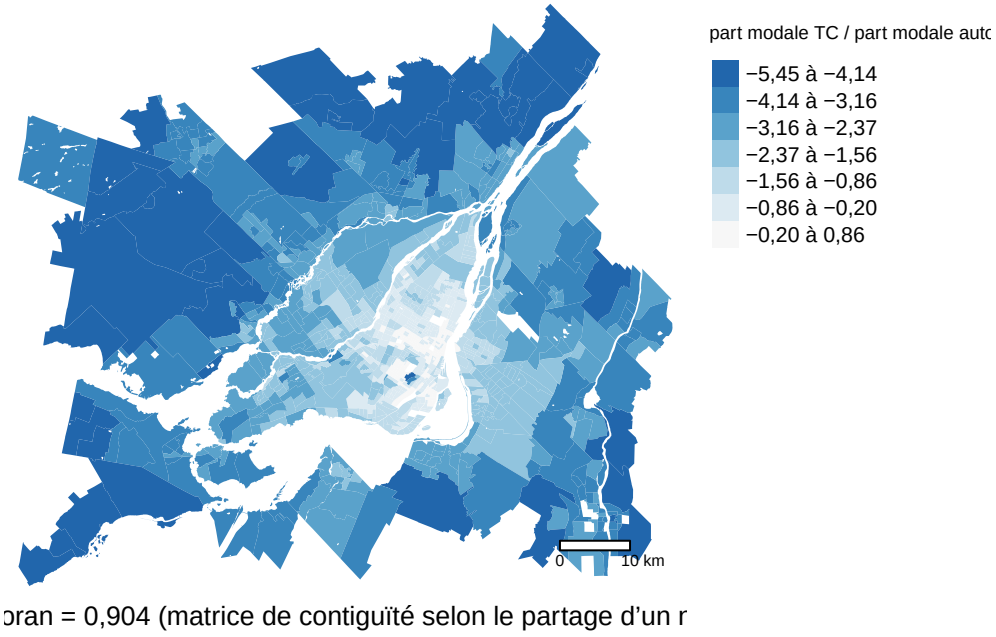


FIGURE 6.7 – Distribution géographique de la variable dépendante

Sans surprise, notre variable dépendante est fortement spatialement autocorrélée. Plus un secteur est éloigné du centre-ville, plus le transport collectif est délaissé au profit de la voiture. Les secteurs dans lesquels la part modale du TC est supérieure à celle de l'automobile sont situés au centre de l'île de Montréal, à proximité des lignes de métro.

6.5.1 Modèles GAM sans intégration de l'espace

Pour modéliser cette variable, nous utilisons deux ensembles de variables indépendantes extraites du recensement de 2021 de Statistique Canada pour les secteurs de recensement de la région métropolitaine de Montréal :

1. Des variables sociodémographiques :

— Pourcentage de personnes issues des minorités visibles (`prt_minorite_vis`).
— Pourcentage de ménages monoparentaux (`prt_monoparental`).
— Taux de chômage (`prt_chomage`).
— Revenu médian des ménages en milliers de dollars (`revenu_median`).
— Pourcentage de personnes à faible revenu (`prt_personnes_faibles_revenu`).

2. Des variables mesurant les niveaux d'accessibilité :

— Niveau d'accessibilité aux emplois en heure de pointe en transport collectif (`acs_idx_emp_tc_peak`).
— Niveau d'accessibilité aux emplois à pied : (`acs_idx_emp_pieton`).

Les indicateurs d'accessibilité sont exprimés comme un score variant de 0 (pire accessibilité au Canada) à 1 (meilleure accessibilité au Canada). Ils servent à capturer la mesure dans laquelle le transport collectif ou la marche permet d'atteindre des emplois (lieux d'activités et de consommation). Dans ce modèle relativement simple, nous nous attendons à ce que des secteurs avec des populations plus vulnérables (variables sociodémographiques) présentent une part modale du transport collectif plus élevée que celle de l'automobile (dépendance au TC). Nous souhaitons cependant vérifier si cette

relation tient toujours une fois que nous prenons en considération la performance du transport collectif et la possibilité d'atteindre des activités à pied.

```
library(mgcv)
library(car)

# Vérification de la multicolinéarité avant d'ajuster un premier modèle
vif(lm(alr_val ~ prt_minorite_vis + prt_monoparental +
         prt_chomage + revenu_median + prt_personnes_faibles_revenu +
         acs_idx_emp_tc_peak + acs_idx_emp_pieton,
       data = data_mtl
         ))
```

```
            prt_minorite_vis             prt_monoparental
                    1.892323                     1.750261
                prt_chomage                revenu_median
                    3.007752                     2.227562
prt_personnes_faibles_revenu          acs_idx_emp_tc_peak
                    5.343381                     2.977366
          acs_idx_emp_pieton
                    2.760775
```

Les valeurs du facteur d'inflation de la variance (VIF) sont relativement faibles, excepté celle du pourcentage de personne à faible revenu qui dépasse 5. Par conséquent, nous décidons de la retirer du modèle.

Nous commençons par ajuster un simple GLM (non spatial), assumant une distribution normale de la variable dépendante, une fois contrôlées les variables indépendantes (`family = gaussian`). Pour valider les résidus de ce modèle, nous utilisons la méthode des résidus simulés.

Aller plus loin

Les résidus simulés
Pour une description des résidus simulés, vous pourrez consulter ce chapitre sur les GLM du livre *Méthodes quantitatives en sciences sociales : un grand bol d'R* (Apparicio et Gelb 2022).

```
library(DHARMa)
library(mgcViz)

# Modèle GLM avec une distribution normale
glm_base <- gam(alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
                          acs_idx_emp_tc_peak + acs_idx_emp_pieton,
                data = data_mtl,
                family = gaussian)

# Validation des résidus
res <- simulateResiduals(glm_base, plot = FALSE)
plot(res)
```

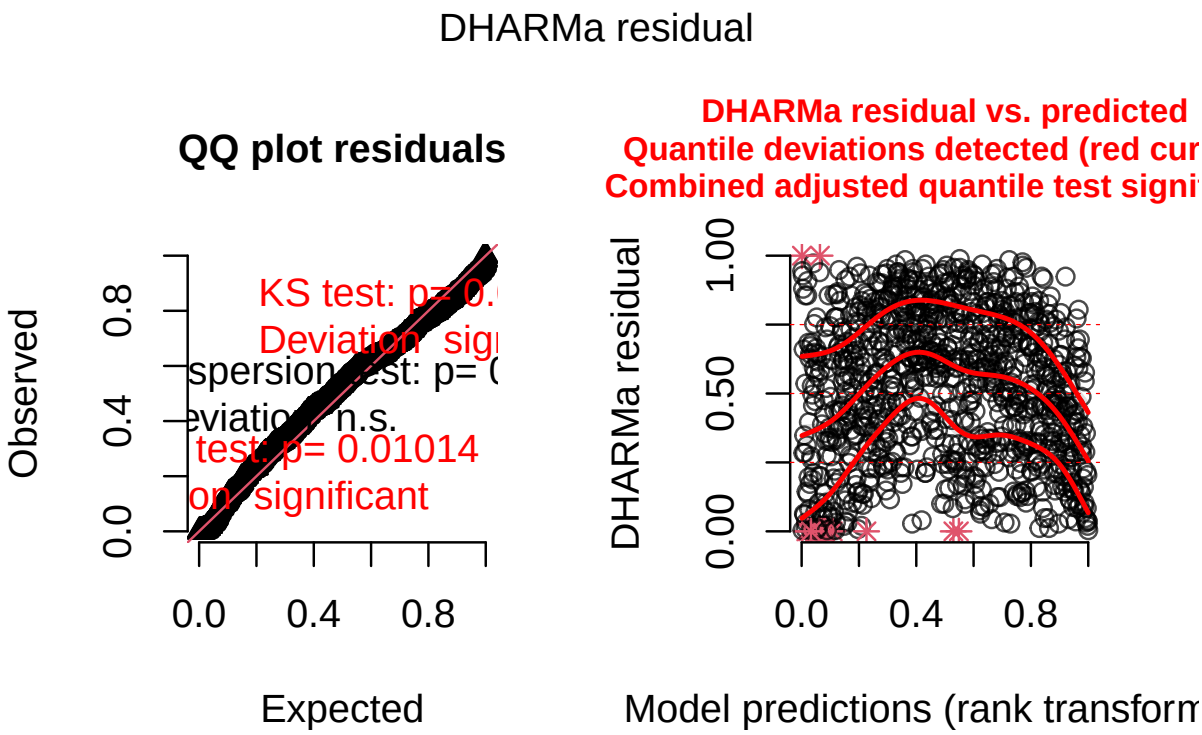


FIGURE 6.8 – Résidus du GLM avec une distribution gaussienne

La figure 6.8 indique que les résidus simulés ne suivent pas vraiment une distribution uniforme et sont marqués par des valeurs extrêmes. Pour améliorer ce premier modèle, nous utilisons une distribution de Student (`family = scat`). Cette dernière ressemble beaucoup à la distribution normale, mais admet davantage de valeurs extrêmes. Autrement dit, elle permet donc de réduire l'impact des valeurs extrêmes dans le modèle.

```
# Modèle GLM avec une distribution normale de Student
glm_base <- gam(alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
                          acs_idx_emp_tc_peak + acs_idx_emp_pieton,
                data = data_mtl,
                family = scat)

# Validation des résidus
res <- simulateResiduals(glm_base, plot = FALSE)
plot(res)
```

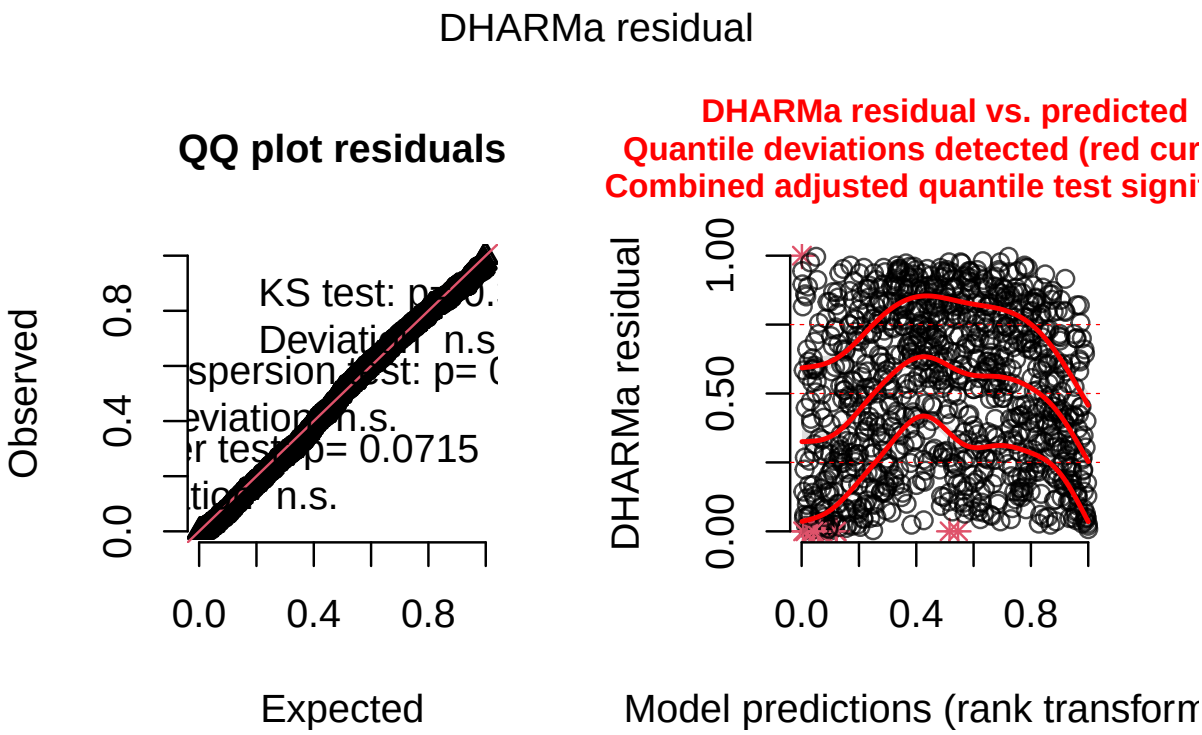


FIGURE 6.9 – Résidus du GLM avec une distribution de Student

Une nette amélioration est observable à la figure 6.9, bien que les quantiles des résidus tendent encore à dévier des valeurs attendues. Plusieurs raisons pourraient expliquer cette situation, mais il est fort probable qu'elle soit provoquée par l'autocorrélation spatiale dans les résidus.

```
# I de Moran des résidus des deux modèles (gaussien et Student)
moran.mc(residuals(glm_base, type = 'pearson'),
                   listw = queen_w, nsim = 999,
                   zero.policy = TRUE)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(glm_base, type = "pearson")
weights: queen_w
number of simulations + 1: 1000

statistic = 0.52098, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

```
moran_i <- moran.mc(residuals(res),
                    listw = queen_w, nsim = 999,
                    zero.policy = TRUE)
```

```
print(moran_i)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(res)
weights: queen_w
number of simulations + 1: 1000

statistic = 0.54085, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

```
moran_text <- format(as.numeric(round(moran_i$statistic,3)),
                     decimal.mark =",")

data_mtl$base_residual <- residuals(glm_base, type = 'response')

# Cartographie
tm_shape(data_mtl) +
  tm_fill(col="base_residual", n = 7,
          style = "jenks",
          midpoint = 0,
          legend.format = list(text.separator = "à",
                               decimal.mark = ",",
                               big.mark = " ",
                               digits = 2),
          palette = "-RdBu", title = 'Résidus simulés') +
tm_layout(frame=FALSE, legend.outside = TRUE) +
tm_scale_bar(breaks = c(0,10))+
tm_xlab(paste0('I de Moran = ', moran_text,
               " (matrice de contiguïté selon le partage d'un nœud)."))
```

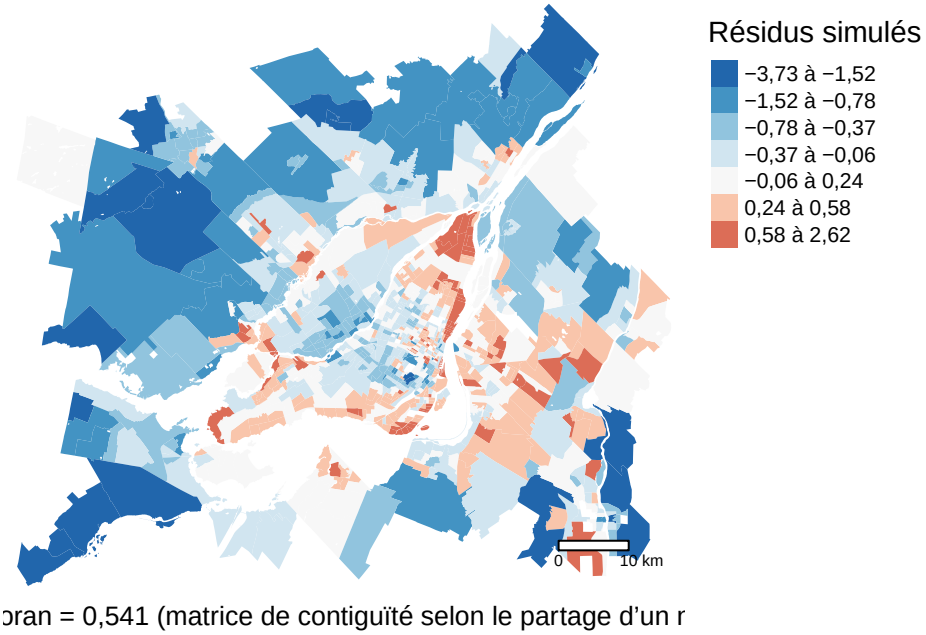


FIGURE 6.10 – Cartographie des résidus du modèle GLM

En effet, la figure 6.10 démontre clairement que la distribution des résidus du modèle GLM (avec une distribution de Student) n'est pas aléatoire spatialement. Les valeurs négatives indiquent des secteurs dans lesquels le modèle tend à prédire un log ratio trop favorable au transport collectif et inversement dans les secteurs avec des valeurs positives.

6.5.2 GAM avec spline spatiale bivariée sur les coordonnées géographiques

Nous ajustons ici un modèle GAM en y intégrant une spline bivariée sur les coordonnées des centroïdes des secteurs de recensement. Le rôle de cette spline est de capturer une tendance spatiale à l'utilisation plus ou moins prononcée du transport collectif. Cet effet correspond à l'agrégat d'un ensemble de variables que nous n'avons pas pu mesurer, par exemple la revendication écologique, la difficulté de stationner une voiture, la présence d'autres modes de déplacement durables connecté au transport en commun, la démographie, etc. Notons ici que nous utilisons une spline de type processus gaussien (`bs = 'gp', m = 3`) avec une structure de covariance Matérn qui est communément utilisée en géostatistique pour le krigeage.

```
# Coordonnées X et Y
XY <- st_coordinates(st_centroid(data_mtl))
data_mtl$X <- XY[,1]
data_mtl$Y <- XY[,2]

# Modèle GAM avec une spline bivariée sur les coordonnées géographiques
gam1 <- gam(alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
                      acs_idx_emp_tc_peak + acs_idx_emp_pieton +
                      s(X,Y, bs = 'gp', m = 3),
            data = data_mtl,
            family = scat)
```

```
# Résidus du modèle
res <- simulateResiduals(gam1, plot = FALSE)
plot(res)
```

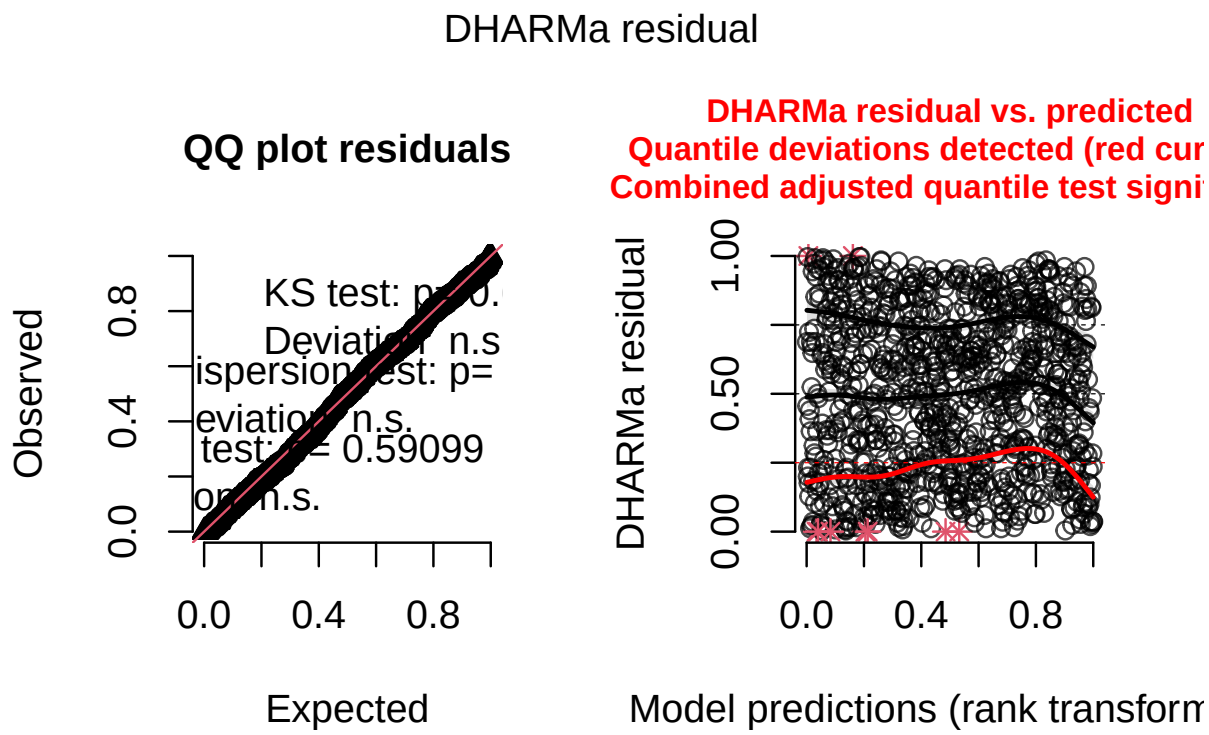


FIGURE 6.11 – Résidus du modèle GAM avec une spline bivariée sur les coordonnées géographiques

Les résidus simulés de ce nouveau modèle se comportent bien mieux que les résidus que nous avions obtenus pour les deux modèle GLM n'intégrant pas l'espace (figure 6.11). Nous pouvons maintenant nous pencher sur l'autocorrélation spatiale des résidus simulés qui devraient suivre une distribution uniforme et aléatoire spatialement.

```
data_mtl$sim_residual <- residuals(res)
test <- moran.mc(data_mtl$sim_residual, listw = queen_w, nsim = 999, zero.policy = TRUE)
print(test)
```

```
    Monte-Carlo simulation of Moran I

data:  data_mtl$sim_residual
weights: queen_w
number of simulations + 1: 1000

statistic = 0.30525, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

```
moran_text <- format(as.numeric(round(test$statistic, 3)),
                     decimal.mark =",")

tm_shape(data_mtl) +
  tm_fill(col="sim_residual", n = 7,
          style = "jenks",
          midpoint = 0.5,
          legend.format = list(text.separator = "à",
                               decimal.mark = ",",
                               big.mark = " ",
                               digits = 2),
          palette = "-RdBu",
          title = 'Résidus simulés') +
tm_layout(frame=FALSE, legend.outside = TRUE) +
tm_scale_bar(breaks = c(0,10))+
tm_xlab(paste0('I de Moran = ', moran_text,
               " (matrice de contiguïté selon le partage d'un nœud)."))
```

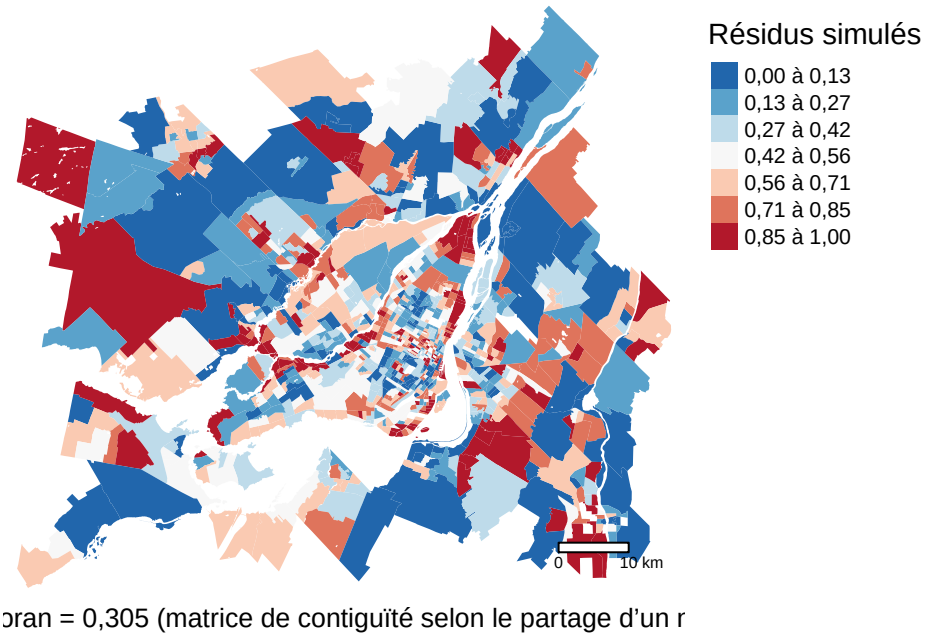


FIGURE 6.12 – Cartographie des résidus du modèle GAM avec une spline bivariée sur les coordonnées géographiques

La valeur du I de Moran (I = 0,305), calculée à partir de la matrice de contiguïté selon le partage d'un nœud (*Queen*), indique toujours une autocorrélation spatiale non négligeable des résidus (figure 6.12).

Toutefois, le modèle GAM n'est pas basé sur les relations de voisinage entre les observations (contiguïté), mais sur la localisation de leurs centroïdes (proximité) pour estimer le terme spatial. Dans ce contexte, il serait plus judicieux d'évaluer l'autocorrélation des résidus du modèle en utilisant une matrice de pondération spatiale basée sur l'inverse des distances.

```
dists <- 1/as.matrix(dist(XY))
diag(dists) <- 0
dist_w <- mat2listw(dists)
moran.mc(data_mtl$sim_residual, listw = dist_w, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  data_mtl$sim_residual
weights: dist_w
number of simulations + 1: 1000

statistic = 0.025581, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

Nous constatons alors que le modèle est parvenu à retirer l'autocorrélation spatiale des résidus (I de Moran de 0,026), si elle est évaluée avec la façon dont l'espace est pris en compte dans le modèle. Interprétons maintenant les résultats du modèle avec la fonction `summary`.

```
results <- summary(gam1)
print(results)
```

```
Family: Scaled t(4.458,0.307)
Link function: identity

Formula:
alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
    acs_idx_emp_tc_peak + acs_idx_emp_pieton + s(X, Y, bs = "gp",
    m = 3)

Parametric coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -2.646739   0.121130 -21.850  < 2e-16 ***
prt_minorite_vis    0.876472   0.100188   8.748  < 2e-16 ***
prt_monoparental    1.398126   0.277368   5.041 4.64e-07 ***
revenu_median      -0.010175   0.001691  -6.017 1.78e-09 ***
acs_idx_emp_tc_peak 3.936414   0.187545  20.989  < 2e-16 ***
acs_idx_emp_pieton -1.548512   0.177466  -8.726  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
         edf Ref.df Chi.sq p-value
s(X,Y) 26.81   28.9  941.5  <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.902   Deviance explained = 81.4%
-REML =  543.3  Scale est. = 1         n = 986
```

Le R^2 ajusté est très élevé (0,902), indiquant que le modèle est capable d'expliquer une part significative de la déviance de nos données. Chacune des variables explicatives a un impact significatif sur le ratio entre les parts modales de TC et de l'automobile. Pour interpréter leurs effets, nous devons tenir compte de l'impact de la transformation *alr*. Pour rappel, notre modèle prédit la moyenne de notre logarithme du ratio entre la part TC et la part automobile. Il est possible de passer de l'échelle logarithme à l'échelle naturelle en utilisant la fonction exponentielle (`exp`). Si nous transformons les coefficients obtenus avec la fonction `exp`, alors ils expriment l'impact **multiplicatif** des variables indépendantes sur le ratio entre la part TC et la part automobile. Si ce ratio augmente, c'est que l'automobile est moins utilisée et/ou que le TC est plus utilisé et inversement.

Tableau 6.1 – Résultats du modèle GLM avec une spline bivariée sur les coordonnées géographiques

Variable	Coefficient	exp(coefficent)	Valeur de p
Constante	-2,647	0,071	0
prt_minorite_vis	0,876	2,402	0
prt_monoparental	1,398	4,048	0
revenu_median	-0,010	0,990	0
acs_idx_emp_tc_peak	3,936	51,235	0
acs_idx_emp_pieton	-1,549	0,213	0

R2 ajusté = 0,9021.

Voici une interprétation des effets obtenus (tableau 6.1) :

- Comparativement à un SR avec 0 % de personnes issues des minorités visibles, un SR avec 25 % de ce groupe de la population verrait augmenter la moyenne du log du ratio de $0,876 \times 0,25 = 0,219$, ce qui correspond à une multiplication par 1,245 du ratio, ou encore à une augmentation de 24,5 % du ratio. Il semblerait donc que toutes choses étant égales par ailleurs, la concentration des minorités visibles soit associée à une plus grande proportion d'utilisation du TC et inversement, à une plus faible utilisation de la voiture.
- Comparativement à un SR avec 0 % de ménages monoparentaux, un SR avec 25 % de ce groupe de la population verrait augmenter la moyenne du log du ratio de $1,398 \times 0,25 = 0,3495$, ce qui correspond à une multiplication par 1,418 du ratio, ou encore à une augmentation de 41,8 % du ratio. Il semblerait donc que toutes choses étant égales par ailleurs, la concentration des ménages monoparentaux soit associée à une plus grande proportion d'utilisation du TC et inversement, à une plus faible utilisation de la voiture.
- Lorsque le revenu médian dans un SR augmente de 1000 dollars, nous observons une réduction de la moyenne du log du ratio de -0,01, soit une multiplication du ratio par 0,99. En d'autres termes, le ratio diminue de 1 % à chaque millier de dollars supplémentaire. Il semblerait donc que dans les SR les plus nantis, l'utilisation du TC soit plus délaissée au profit de la voiture.
- Lorsque l'accessibilité aux emplois en transport collectif est plus élevée, la moyenne du log du ratio augmente. Dans un SR avec le pire niveau d'accessibilité possible, augmenter l'accessibilité au niveau du premier quartile (au moins mieux que 25% des SR) serait associé à une augmentation de la moyenne du log du ratio de $3,936 \times 0,25 = 0,984$, ce qui correspond à une multiplication par 2,675 du ratio.

— Cependant, lorsque l'accessibilité à pied augmente, nous observons une réduction de la moyenne du log du ratio. Dans un SR avec le pire niveau d'accessibilité possible, augmenter l'accessibilité au niveau du premier quartile (au moins mieux que 25 % des SR) serait associé à une augmentation de la moyenne du log du ratio de $-1,549 \times 0,25 = -0,387$, ce qui correspond à une multiplication par 0,679 du ratio, ou encore à une réduction de 32,1 %. Ce résultat s'explique vraisemblablement par l'utilisation des modes de transport actifs plutôt que par une utilisation accrue de la voiture.

Dans le résumé du modèle, nous pouvons observer que la spline utilisée pour capturer l'effet spatial à un impact significativement différent de 0 et comporte environ 27 degrés de liberté. Afin d'observer l'effet de cette spline, nous utilisons l'approche des effets marginaux. Cette approche consiste à effectuer une prédiction avec le modèle de la valeur attendue de la variable dépendante en conservant identiques toutes les variables indépendantes, sauf la localisation dans l'espace. En d'autres termes, nous essayons de voir comment le modèle s'ajuste lorsqu'il doit prédire la valeur Y pour une observation spécifique, mais que celle-ci est déplacée dans l'espace. Ainsi, la variation de la prédiction est affectée uniquement par l'espace.

Pour obtenir l'effet uniquement spatial, nous pouvons mettre arbitrairement toutes les valeurs à 0 des différentes variables indépendantes du modèle. Ainsi, même multipliées par leurs coefficients, leur contribution dans la prédiction restera 0. Il suffit ensuite de retirer la constante pour ne conserver que l'effet de la spline.

```
#' Nous commençons par construire un dataframe de prédictions. Toutes les
#' variables indépendantes sont fixées à 0 et nous allons
#' prédire les valeurs tous les 500 m

bbox <- st_bbox(data_mtl)
df_pred <- expand.grid(
  prt_minorite_vis = 0,
  prt_monoparental = 0,
  revenu_median = 0,
  acs_idx_emp_tc_peak = 0,
  acs_idx_emp_pieton = 0,
  X = seq(bbox[[1]], bbox[[3]], 500),
  Y = seq(bbox[[2]], bbox[[4]], 500)
)

# Nous effectuons ensuite la prédiction sur le prédicteur linéaire
df_pred$pred_log <- predict.gam(gam1, newdata = df_pred, type = 'link')

# Nous retirons ensuite la constante pour ne garder que l'effet spatial
df_pred$pred_log_cent <- df_pred$pred_log  - gam1$coefficients[[1]]

# Nous pouvons ensuite utiliser la fonction exp pour faciliter l'interprétation
# puisque le ratio sera plus facile à lire que le log ratio
df_pred$pred_exp <- exp(df_pred$pred_log_cent)

# Il ne reste ensuite qu'à cartographier notre effet spatial
sf_pred <- df_pred %>%
  st_as_sf(coords = c('X', 'Y'), crs = st_crs(data_mtl)) %>%
  subset(lengths(st_intersects(.,data_mtl)) > 0)
```

```
map_spline <- tm_shape(sf_pred) +
                tm_dots(col="pred_exp", n = 7,
                        style = "fisher",
                        midpoint = 1,
                        legend.format = list(text.separator = "à",
                                             decimal.mark = ",",
                                             big.mark = " ",
                                             digits = 2),
                        palette = "-RdBu",
                        title = paste0('effet de la spline bivariée\n(edf = ',
                                       round(results$edf), ')')) +
              tm_shape(data_mtl) +
                tm_borders('darkgrey', lwd = 0.01) +
              tm_layout(frame=FALSE, legend.outside = TRUE) +
              tm_scale_bar(breaks = c(0,10))

map_spline
```

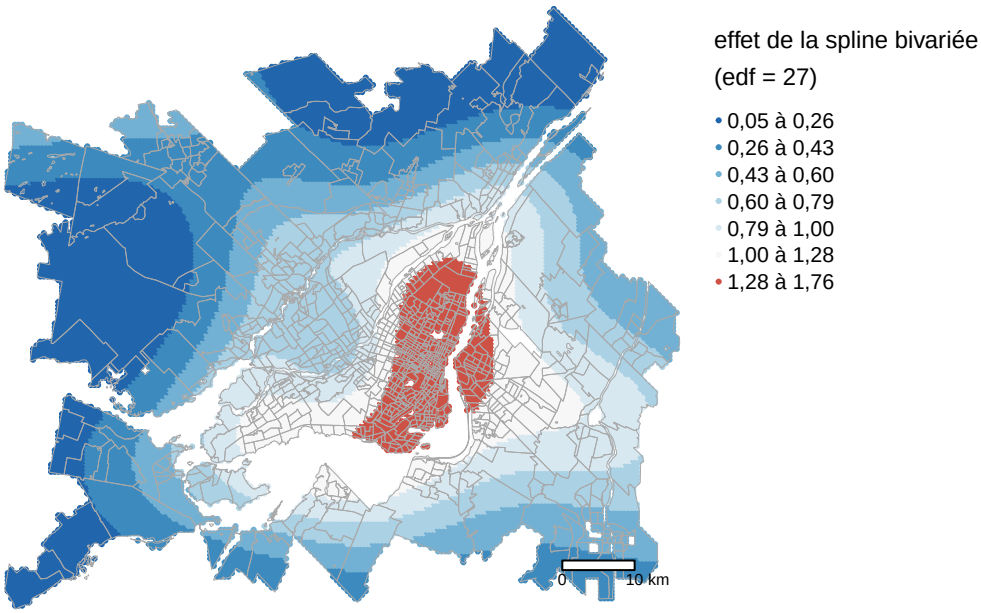


Figure 6.13 – Représentation de l'impact de la spline spatiale bivariée

La figure 6.13 permet de visualiser l'effet multiplicatif de la spline spatiale sur le ratio prédit par le modèle. Ainsi, toute chose étant égale par ailleurs, le ratio de part modale TC/auto est de 30 % à 70 % plus fort dans les quartiers centraux de Montréal, allant de Lachine à Anjou. L'extrémité ouest de Longueuil fait aussi partie de cette zone avec le bonus spatial le plus important. Aussi, on constate que l'effet spatial diminue avec la distance au centre-ville. Pour la majeure partie de Laval, le ratio est réduit de près de 50 %.

6.5.3 GAM avec spline spatiale de type MRF

Puisque nos observations sont des polygones, utiliser leurs centroïdes pour représenter l'espace est une approximation maladroite. Nous allons donc opter pour un *Markov random field* afin de modéliser les variations spatiales en fonction du voisinage.

L'enjeu avec ce type de modèle est de sélectionner son rang (k). Pour cela, nous testons plusieurs valeurs allant de 10 à 250 autorisant une complexité grandissante du terme spatial. Notons d'emblée que notre spline bivariée précédente avait nécessité seulement 25 degrés de liberté supplémentaires. La sélection de k (le nombre de degrés de liberté accordé au MRF) reposera sur deux critères : l'autocorrélation spatiale encore présente dans les résidus (I de Moran calculé sur les résidus de Pearson) et l'AIC (pénalisant la complexité des modèles).

Pour accélérer le calcul, nous réalisons cette opération en traitement parallèle (*multiprocessing*). Quatre cœurs indépendants s'occuperont d'ajuster les différents modèles.

```
# Package pour le traitement en parallèle
library(future)
library(future.apply)

# nous préparons les informations de voisinage pour le terme MRF
nb <- poly2nb(data_mtl)
names(nb) <- as.integer(attr(nb, "region.id"))
data_mtl$ID_LOC <- as.integer(attr(nb, "region.id"))

ks <- seq(10,250,20)

# nous allons itérer sur les valeurs de ks et ajuster nos modèles
# nous récupérons à chaque fois les informations pertinentes dans nos
# résidus

S0 <- sum(sapply(queen_w$weights, sum))

future::plan(future::multisession(workers = 4))

resultats <- future_sapply(ks, function(k){

  model <- gam(alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
                 acs_idx_emp_tc_peak + acs_idx_emp_pieton +
               s(ID_LOC, bs = 'mrf', xt = list(nb = nb), k = k),
         data = data_mtl, family = scat)

  resid <- residuals.gam(model, type = 'scaled.pearson')
  I <- moran(resid, queen_w, n = length(resid), S0 = S0, zero.policy = TRUE)
  aic_score <- AIC(model)
  return(c(I[[1]],aic_score,k))
})
```

```
df_plot <- data.frame(t(resultats))
names(df_plot) <- c('moranI', 'aic', 'k')
```

```
# Graphique pour le I de Moran
plot1 <- ggplot(df_plot) +
  geom_line(aes(x = k, y = moranI), color = 'black') +
  geom_point(aes(x = k, y = moranI), color = 'red') +
  labs(x = 'k', y = 'I de Moran') +
  theme_bw()

# Graphique pour les valeurs d'AIC
plot2 <- ggplot(df_plot) +
  geom_line(aes(x = k, y = aic), color = 'black') +
  geom_point(aes(x = k, y = aic), color = 'red') +
  labs(x = 'k', y = 'AIC')+
  theme_bw()

# Combinaison des deux graphiques
ggarrange(plot1, plot2)
```

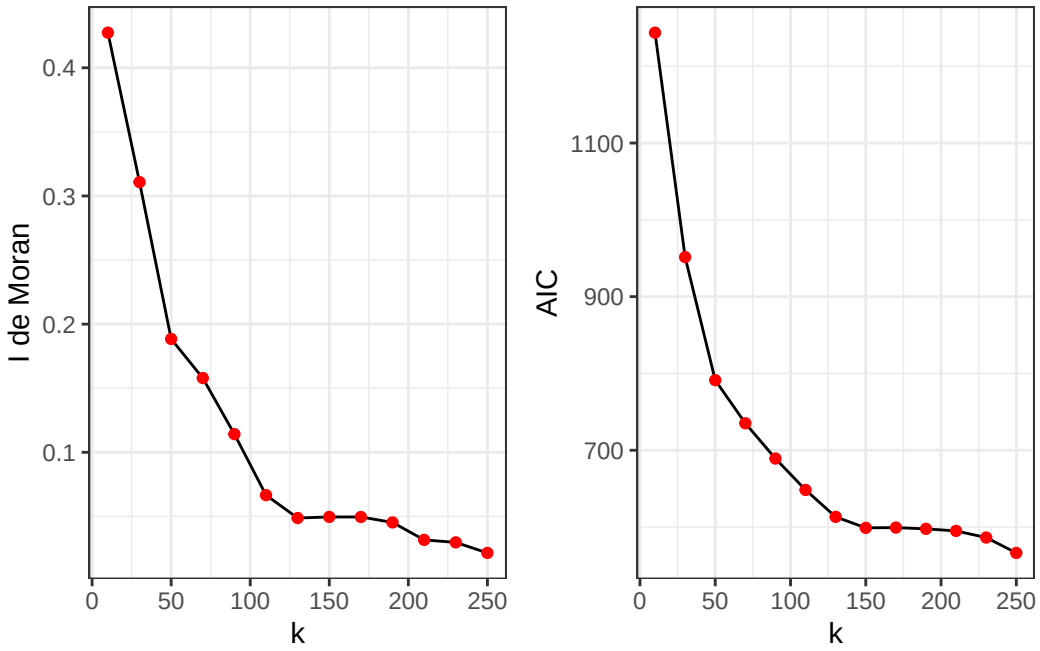


FIGURE 6.14 – Critères de sélection de k

La figure 6.14 permet de constater que l'AIC et le I de Moran diminuent à mesure que k augmente. Pour le I de Moran, nous constatons qu'au-delà de $k = 100$, nous passons à des valeurs inférieures à 0,05. Aussi l'AIC semble se stabiliser lorsque $k = 150$. Nous pourrions donc ajuster un modèle avec $k = 150$ pour avoir un bon compromis entre AIC et réduction du I de Moran des résidus.

```r
gam2 <- gam(alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
                  acs_idx_emp_tc_peak + acs_idx_emp_pieton +
              s(ID_LOC, bs = 'mrf', xt = list(nb = nb), k = 150),
         data = data_mtl, family = scat)

res <- simulateResiduals(gam2, plot = FALSE)
plot(res)
summary(gam2)
```

```
Family: Scaled t(3.245,0.22)
Link function: identity

Formula:
alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
    acs_idx_emp_tc_peak + acs_idx_emp_pieton + s(ID_LOC, bs = "mrf",
    xt = list(nb = nb), k = 150)

Parametric coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)         -2.555219   0.108923 -23.459  < 2e-16 ***
prt_minorite_vis     1.303376   0.106976  12.184  < 2e-16 ***
prt_monoparental     0.907048   0.239301   3.790 0.000150 ***
revenu_median       -0.008566   0.001473  -5.816 6.04e-09 ***
acs_idx_emp_tc_peak  3.118112   0.199688  15.615  < 2e-16 ***
acs_idx_emp_pieton  -1.102568   0.288893  -3.817 0.000135 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
            edf Ref.df Chi.sq p-value
s(ID_LOC) 126.7  144.4   2219  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.925   Deviance explained = 83.7%
-REML = 417.26  Scale est. = 1         n = 986
```

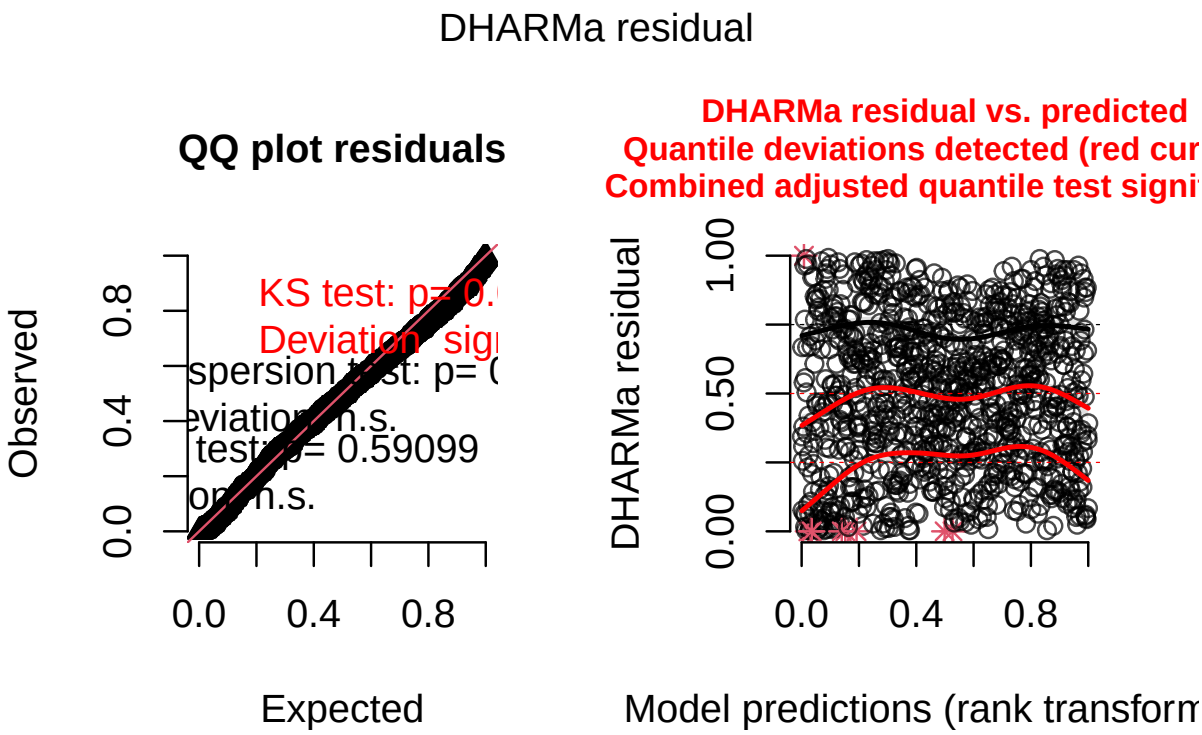


FIGURE 6.15 – Résidus du modèle GAM avec une spline de type MRF avec k = 150

TABLEAU 6.2 – Résultats du modèle GLM avec une spline spatiale de type MRF

Variable	Coefficient	exp(coefficent)	Valeur de p
Constante	-2,555	0,078	0
prt_minorite_vis	1,303	3,682	0
prt_monoparental	0,907	2,477	0
revenu_median	-0,009	0,991	0
acs_idx_emp_tc_peak	3,118	22,604	0
acs_idx_emp_pieton	-1,103	0,332	0

R2 ajusté = 0,9255.

Il apparaît que les résidus simulés de ce modèle suivent une distribution assez proche d'une distribution uniforme. Nous constatons aussi que toutes les variables indépendantes sont encore significatives et affectent notre variable dépendante dans la même direction (tableau 6.2).

```
moran.mc(residuals(res), listw = queen_w, nsim = 999, zero.policy = TRUE)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(res)
```

```
weights: queen_w
number of simulations + 1: 1000

statistic = -0.022059, observed rank = 145, p-value = 0.855
alternative hypothesis: greater
```

Nous avons bien résolu le problème de l'autocorrélation spatiale de résidus. Nous pouvons donc à présent visualiser l'effet spatial de notre *Markov random field*. Pour cela, nous allons une fois encore appliquer la méthode des effets marginaux.

```
# On extrait toutes les observations du jeu de données
pred_df <- data_mtl[c('prt_minorite_vis', 'prt_monoparental', 'revenu_median',
                      'acs_idx_emp_tc_peak', 'acs_idx_emp_pieton', 'ID_LOC')]

# On met toutes les valeurs à 0 sauf l'identifiant des SR
pred_df[c(1:5)] <- 0

# On effectue la prédiction
pred_df$log_ratio <- predict.gam(gam2, newdata = pred_df, type = 'link')

# On retire l'effet de la constante
pred_df$log_ratio <- pred_df$log_ratio - gam2$coefficients[[1]]

# On bascule en exponentiel pour avoir l'effet sur le ratio
pred_df$effet_sp <- exp(pred_df$log_ratio)
results_mrf <- summary(gam2)

# on peut maintenant cartographier cet effet
map_mrf <- tm_shape(pred_df) +
  tm_fill(col="effet_sp", n = 7,
          style = "jenks",
          midpoint = exp(0),
          legend.format = list(text.separator = "à",
                               decimal.mark = ",",
                               big.mark = " ",digits = 2),
          palette = "-RdBu", title = paste0('Effet du MRF\n(edf = ', round(results_mrf$edf), ')')) +
  tm_layout(frame=FALSE, legend.outside = TRUE) +
  tm_scale_bar(breaks = c(0,10))
tmap_arrange(map_spline, map_mrf, nrow = 2)
```

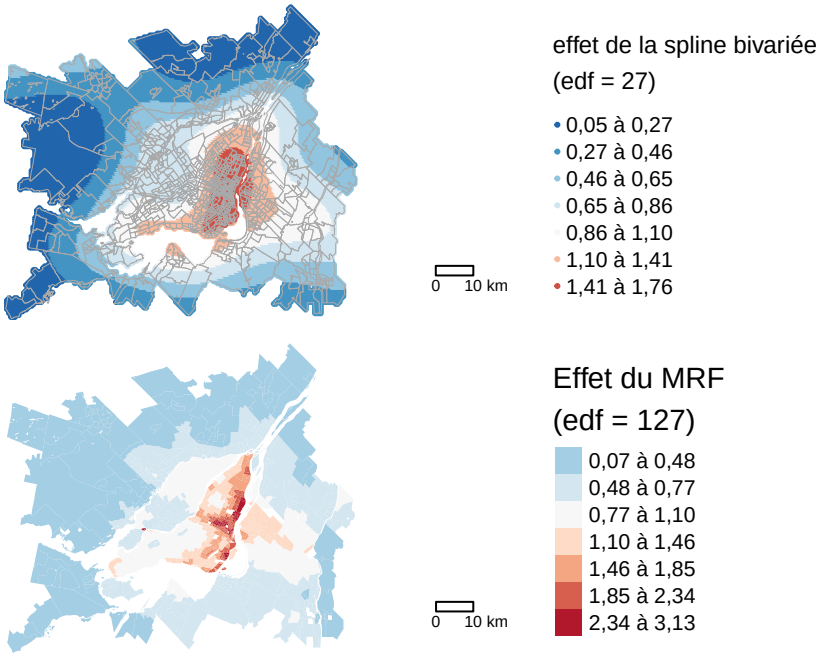


FIGURE 6.16 – Comparaison de la spline bivariée et du MRF

Ce terme spatial ressemble beaucoup à celui que nous avions obtenu avec la spline bivariée construite avec les coordonnées des centroïdes. Cependant, nous lui avons accordé un plus grand nombre de degrés de liberté et il s'ajuste mieux à la géographie de nos secteurs de recensement (polygones). Il aurait été possible de donner plus de flexibilité à la spline bivariée en utilisant également le paramètre `k`, elle serait cependant restée moins adaptée que le MRF. Par contre, l'avantage intéressant de l'approche par spline bivariée est qu'elle permet de prédire des valeurs à de nouvelles localisations, ce que ne permet pas le MRF.

6.6 Quiz de révision

Questions

— **Quelle est l'hypothèse clé des modèles de régression linéaire classique qui est assouplie dans les GAM?**

- Les résidus doivent suivre une distribution normale.
- Les relations entre les variables indépendantes et la variable dépendante doivent être linéaires.
- Les variables indépendantes ne doivent pas être corrélées entre-elles.

Relisez au besoin la section 6.1.

— **Dans un GAM, que représente la fonction f ?**

- Une fonction linéaire fixe.
- Une fonction non paramétrique ajustée aux données.
- Une fonction quadratique définie par l'utilisateur.
- La somme d'un ensemble de bases non linéaires multipliées par leurs coefficients.

Relisez au besoin le début de la section 6.1.

— **Comment un modèle GAM peut-il être utilisé pour intégrer l'espace?**

  — Une spline spécialisée est utilisée pour capturer une tendance spatiale d'arrière-plan dans les données
  — Le modèle intègre directement l'autocorrélation spatiale dans son hypothèse quant à la distribution de ses résidus
  — Le modèle intègre une version spatialement décalée de la variable Y
  — Le modèle intègre une version spatialement décalée des variables X

  Relisez au besoin la section 6.3.

— **Dans quel cas préfère-t-on utiliser un RMF plutôt qu'une spline bivariée sur les coordonnées géographiques?**

  — Lorsque les données sont représentées dans l'espace comme des points.
  — Lorsque l'espace est représenté plus adéquatement par une relation de voisinage entre les observations.
  — Lorsque le terme spatial que l'on espère ajuster doit être moins spatialement autocorrélé

  Relisez au besoin la section 6.4.

— **Pour analyser l'effet spatial modélisé avec une spline, on peut :**

  — Regarder le nombre de degrés de liberté (edf) de la spline.
  — Interpréter les coefficients de chacune des bases de la spline.
  — Cartographier les résidus du modèle.
  — Cartographier l'effet de la spline grâce à la méthode des effets marginaux

  Relisez au besoin la section 6.5.2.

Réponses

— Quelle est l'hypothèse clé des modèles de régression linéaire classique qui est assouplie dans les GAM?
  — Les relations entre les variables indépendantes et la variable dépendante doivent être linéaires.
— Dans un GAM, que représente la fonction f ?
  — Une fonction non paramétrique ajustée aux données.
  — La somme d'un ensemble de bases non linéaires multipliées par leurs coefficients.
— Comment un modèle GAM peut-il être utilisé pour intégrer l'espace?
  — Une spline spécialisée est utilisée pour capturer une tendance spatiale d'arrière-plan dans les données
— Dans quel cas préfère-t-on utiliser un RMF plutôt qu'une spline bivariée sur les coordonnées géographiques?
  — Lorsque l'espace est représenté plus adéquatement par une relation de voisinage entre les observations.
— Pour analyser l'effet spatial modélisé avec une spline, on peut :
  — Regarder le nombre de degrés de liberté (edf) de la spline.
  — Cartographier l'effet de la spline grâce à la méthode des effets marginaux

6.7 Exercices de révision

Exercice

Exercice 1. Réalisation d'un modèle GAM aspatiale

```
library(sf)
library(mgcv)
library(car)
library(DHARMa)
library(mgcViz)
library(spdep)
library(dplyr)
# Chargement du jeu de données sur l'agglomération lyonnaise
load("data/Lyon.Rdata")

# Vérification de la multicolinéarité (VIF)
vif(lm(NO2 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed, data = LyonIris))

# Construction du modèle GLM gaussien
Modele.GAM1 <- gam(à compléter)
# Résidus simulés du modèle gaussien
GAM1.res <- à compléter
plot(GAM1.res)

# Construction du modèle GLM avec une distribution de Student
Modele.GAM2 <- gam(à compléter)
# Résidus simulés avec une distribution de Student
GAM2.res <- à compléter
plot(GAM2.res)

# Comparaison des deux modèles
AIC(Modele.GAM1, Modele.GAM2)

# Résultats du modèle gaussien
summary(à compléter)

# Matrice de contiguïté selon le partage d'un nœud
queen_nb <- à compléter
queen_w <- à compléter

# I de Moran sur les résidus gaussiens
moran.mc(residuals(à compléter))

# I de Moran sur les résidus simulés
moran.mc(à compléter)
```

Correction à la section 11.6.1.

Exercice

Exercice 2. Réalisation d'un modèle GAM avec une spline bivariée

```
# Chargement du jeu de données sur l'agglomération lyonnaise
load("data/Lyon.Rdata")
# Ajout des coordonnées x et y dans la couche sf LyonIris
à compléter
# Modèle GAM avec la distribution de student et une spline
# sur les coordonnées x et y avec la distribution de Student
Modele.GAMSpline <- à compléter
summary(Modele.GAMSpline)
```

Correction à la section 11.6.2.

7 Modèles linéaires généralisés avec des vecteurs spatiaux

Daniel Griffith (2019) a proposé une autre technique pour ajouter l'espace dans un modèle, connue sous le nom du filtrage spatial (*spatial filtering*). Cette technique est très flexible, car elle s'applique à n'importe quel type de modèles GLM, GLMM et GAM. L'approche est relativement simple : elle consiste à ajouter dans le modèle un ensemble de variables supplémentaires dont le rôle est de capturer des tendances spatiales qui autrement seraient présentes dans les résidus. Ces variables supplémentaires sont appelées des vecteurs propres spatiaux (*spatial eigenvectors*, SEV) et sont obtenues à partir de la matrice de pondération spatiale décrivant l'espace de nos données. Plus exactement, les SEV sont obtenus en décomposant la matrice spatiale de la même manière que nous décomposons la variance d'un jeu de données lorsque nous réalisons une analyse en composantes principales (ACP). Pour un rappel sur l'ACP, vous pouvez vous référer au chapitre sur les analyses factorielles du manuel *Méthodes quantitatives en sciences sociales : un grand bol d'R* (Apparicio et Gelb 2022). Ces vecteurs propres sont aussi appelés *Moran eigenvectors* (MEM).

Objectif

Objectifs d'apprentissage visés dans ce chapitre

À la fin de ce chapitre, vous devriez être en mesure de :

- préparer des vecteurs propres de Moran à partir d'une matrice de pondération spatiale;
- utiliser ces vecteurs propres pour appliquer la méthode du filtrage spatial;
- analyser les résultats produits par un modèle de type SEVM (*Spatial Eigenvector Model*).

Package

Liste des *packages* utilisés dans ce chapitre

- Pour calculer des vecteurs propres de Moran :
  - `adespatial` dédié à l'analyse spatiale multidimensionnelle.
- Pour ajuster des modèles SEVM :
  - `spmoran` et `spatialreg` dédiés aux régressions spatiales.
  - `mgcv` et `gamlss` pour modéliser des GLM avec une grande flexibilité.
- Pour analyser les résidus des modèles :
  - `DHARMa` pour valider la distribution de modèle GLM complexes.
  - `spdep` pour construire des matrices de pondération spatiales et calculer le I de Moran.
- Pour construire des cartes et des graphiques :
  - `tmap` est certainement le meilleur *package* pour la cartographie.
  - `ggplot2` pour construire des graphiques.

7.1 Vecteurs propres de Moran

À partir d'une matrice spatiale W de taille $n \times n$, il est possible de calculer $n - 1$ vecteurs propres de Moran qui représentent chacun un agencement spatial possible. Ces vecteurs propres sont obtenus en décomposant la variance de la matrice spatiale de façon analogue à une analyse en composantes principales (ACP).

Aller plus loin

Analyse en composantes principales (ACP)
Pour un rappel sur l'ACP, nous vous invitons à lire la section suivante du livre *Méthodes quantitatives en sciences sociales : un grand bol d'R* (Apparicio et Gelb 2022).
Brièvement, une ACP est une méthode de réduction des données qui permet de résumer la variance d'un jeu de données ayant k colonnes avec un ensemble de l vecteurs propres avec $k > l$. Les vecteurs propres sont des nouvelles variables obtenues avec une combinaison linéaire des colonnes originales k. Le premier vecteur propre correspond à la tendance principale du jeu de données, formée par les plus importantes corrélations entre les variables originales. Ces vecteurs propres sont orthogonaux, ce qui signifie qu'ils ne sont pas corrélés entre eux.

Appliquer l'ACP à une matrice spatiale revient à extraire des variables capables de résumer l'organisation spatiale présente dans la matrice de pondération spatiale *W*. Le premier vecteur propre issu de l'ACP appliquée à *W* représente alors l'agencement spatial avec la plus forte autocorrélation spatiale positive possible. À l'inverse, le dernier vecteur propre représente l'agencement spatial avec la plus forte autocorrélation spatiale négative possible. Comme pour l'ACP, ces vecteurs propres spatiaux sont non corrélés entre eux. Ils sont aussi appelés les vecteurs propres de Moran dû au lien entre leur formulation et celle de la mesure d'autocorrélation spatiale du *I* de Moran.

Prenons un exemple concret avec les données de secteurs de recensement de Montréal. La figure 7.1 permet de visualiser la valeur du *I* de Moran associée à chacun des vecteurs propres obtenus sur la matrice de contiguïté (W) selon le partage d'un nœud (*Queen*). Nous constatons ainsi que le premier vecteur propre a une valeur de *I* de Moran proche de 1 et que l'autocorrélation spatiale reste positive jusqu'au vecteur 375 environ. Au-delà de ce seuil, l'autocorrélation spatiale des vecteurs devient négative.

Nous pouvons visualiser la géographie propre de chacun de ces vecteurs. À la figure 7.2, nous illustrons quatre de ces vecteurs présentant respectivement une autocorrélation spatiale positive, une distribution spatiale aléatoire et finalement une autocorrélation spatiale négative.

Rappelons ici que ces vecteurs sont obtenus pas décomposition de la matrice de pondération spatiale. Il ne s'agit pas de données qui ont été collectées sur le territoire d'étude. Ils représentent des géographies possibles qui peuvent émerger de notre matrice de pondération spatiale W.

7.2 Filtrage spatial

Dans un modèle utilisant l'approche du filtrage spatial, l'idée est d'ajouter dans l'équation une sélection de vecteurs spatiaux permettant de réduire l'autocorrélation spatiale dans les résidus. Pour chaque vecteur spatial ajouté au modèle, un coefficient est ajusté. La somme des vecteurs spatiaux multipliés par leurs coefficients respectifs est donc un terme dans le modèle tentant de capturer des tendances spatiales plus ou moins complexes qui autrement auraient fait partie des résidus.

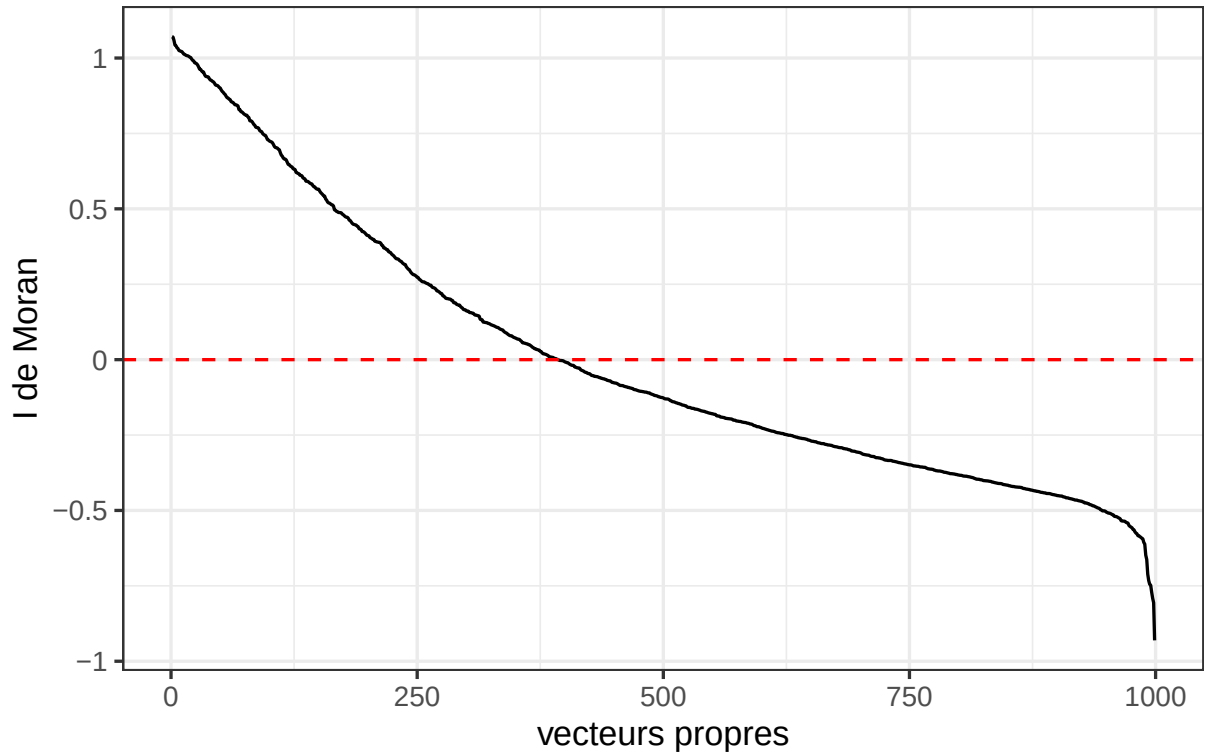


FIGURE 7.1 – Autocorrélation des vecteurs propres issus d'une matrice de contiguïté (Queen) sur les secteurs de recensement de la région métropolitaine de recensement de Montréal

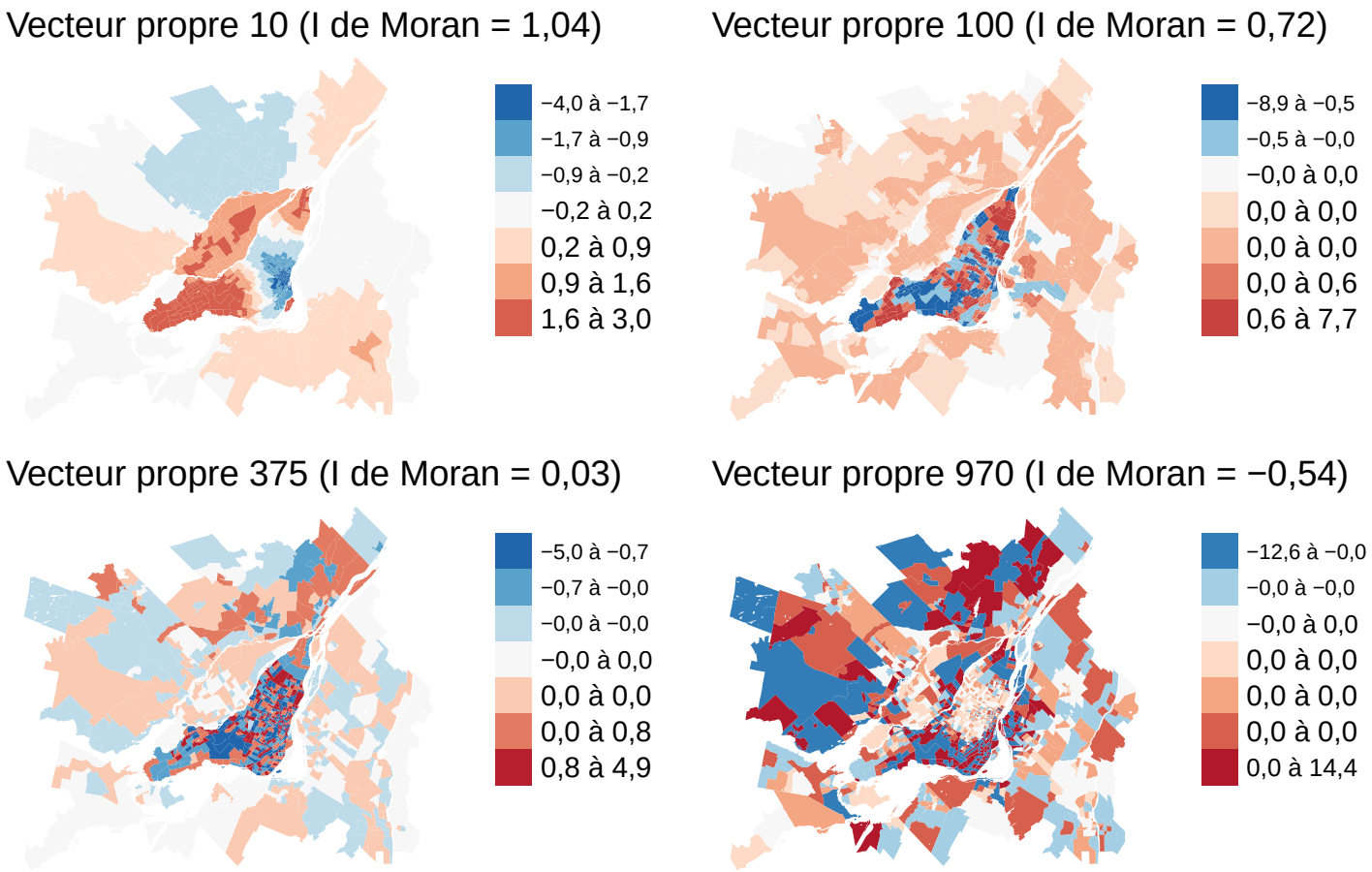


FIGURE 7.2 – Représentation de quatre vecteurs propres issue d'une matrice spatiale (Queen) sur les secteurs de recensement de la RMR de Montréal

$$
\begin{aligned}
y &\sim D(\mu, \theta) \\
g(\mu) &= \beta_0 + \beta X + \zeta \\
\zeta &= \alpha Z
\end{aligned}
\tag{7.1}
$$

avec :

- y, la variable dépendante.
- D, une distribution avec une espérance μ et ses autres paramètres θ.
- β_0, la constante.
- X, les variables indépendantes dont l'effet est supposé linéaire.
- β, les coefficients des variables indépendantes.
- ζ, le terme spatial utilisé pour le filtrage spatial.
- α, les coefficients des vecteurs propres spatiaux.
- Z, les vecteurs propres spatiaux ajoutés au modèle.

Note

Filtrage spatial et GAM
Il existe une forte similarité dans la formulation d'un modèle GAM et d'un modèle utilisant du filtrage spatial. Pour rappel, un modèle GAM introduit l'espace avec une spline. Or, une spline est une combinaison d'un ensemble de fonctions de base (obtenues sur les coordonnées spatiales ou la matrice de contiguïté) multipliées par des coefficients. Dans un modèle avec filtrage spatial, la logique est similaire, mais les fonctions de base sont cette fois-ci les vecteurs propres spatiaux.

L'enjeu principal avec la méthode du filtrage spatial est de sélectionner adéquatement les vecteurs propres à intégrer dans le modèle. En effet, il n'est pas possible de tous les sélectionner, car nous aurions alors plus de coefficients à estimer que d'observations dans les données. Le choix des vecteurs propres doit valider les critères suivants :

1. Réduire l'autocorrélation spatiale des résidus (**correction de la dépendance spatiale**).
2. Améliorer l'ajustement du modèle (**contribution à la prédiction**).
3. Ne pas complexifier indûment le modèle (**principe de parcimonie**).

Il existe actuellement trois approches permettant de sélectionner les vecteurs propres spatiaux à intégrer dans un modèle. Ces trois approches partent de postulats différents et ont chacune leurs avantages et inconvénients que nous décrivons dans les trois prochaines sous-sections.

7.2.1 Sélection itérative

La première approche proposée (Griffith 2003) consiste à appliquer une procédure itérative de sélection des vecteurs propres spatiaux, à la manière d'une régression pas à pas (*stepwise regression*). Elle ajoute un vecteur propre spatial au modèle s'il vérifie les trois critères suivants :

1. Il permet d'augmenter le R^2 du modèle.
2. Il permet de réduire l'autocorrélation spatiale des résidus.
3. Il existe une corrélation significative entre le vecteur propre et la variable dépendante.

Puisque les vecteurs propres sont non corrélés, leur ordre d'ajout dans le modèle n'a pas d'importance. Il est donc conseillé d'ordonner les vecteurs propres en fonction de leur corrélation avec la variable dépendante pour effectuer leur sélection.

Il est assez simple d'étendre cette approche à des GLM ou même à des GLMM. Pour ces modèles, l'objectif ne sera pas de maximiser le R^2, mais plutôt d'obtenir une réduction significative de la déviance (*likelihood ratio test*). Notons que pour ce type de modèle, il est important de vérifier l'autocorrélation spatiale des résidus standardisés ou simulés, car de nombreux GLM supposent une hétéroscédasticité des résidus. Pour un rappel sur le *likelihood ratio test*, vous pouvez vous référer à cette section du livre *Méthodes quantitatives en sciences sociales : un grand bol d'R* (Apparicio et Gelb 2022).

L'approche par sélection itérative est très intuitive et relativement facile à mettre en œuvre. Cependant, elle comporte plusieurs défauts. Lorsque le jeu de données est grand, elle devient extrêmement coûteuse en temps de calcul, car le modèle doit être ajusté un grand nombre de fois. Aussi, bien que l'ordre d'ajout des vecteurs propres ne soit pas important, leur ajout séquentiel peut poser un problème. En effet, des combinaisons de vecteurs propres peuvent être particulièrement efficaces pour effectuer le filtrage spatial, ce qui n'est pas permis par la procédure pas à pas (*stepwise*). Notons ici que des approches alternatives utilisant par exemple des algorithmes génétiques ont été proposées (Helbich et Griffith 2016), bien qu'elles ne soient pas communément utilisées.

7.2.2 Effet aléatoire

Murakami et Griffith (2015) ont proposé une approche par effets aléatoires pour effectuer le filtrage spatial. Dans cette approche, les coefficients des vecteurs propres ne sont pas estimés comme des effets fixes (simples coefficients), mais comme des effets aléatoires provenant d'une distribution normale. La formulation du modèle devient alors :

$$\begin{aligned} y &\sim D(\mu, \theta) \\ g(\mu) &= \beta_0 + \beta X + \alpha Z \\ \alpha &\sim N(0, \sigma) \end{aligned} \tag{7.2}$$

avec :

- σ, un terme de variance estimé par le modèle contrôlant la variation des coefficients des vecteurs propres spatiaux α qui tend à s'écarter de 0.

L'estimation des coefficients comme des effets aléatoires entraîne une pénalisation de leur estimation. Ainsi, le modèle devient capable de retirer de lui-même les vecteurs propres non pertinents. Habituellement, les vecteurs propres inclus dans ce modèle sont les vecteurs propres ayant une autocorrélation spatiale positive, car, dans la vaste majorité des cas, les résidus sont positivement spatialement autocorrélés.

Fait intéressant, cette approche pénalisée par effets aléatoires revient à imposer une pénalité de type *ridge* ($l2$) aux coefficients des vecteurs spatiaux. L'objectif des régressions pénalisées est de faire tendre vers zéro les coefficients les moins importants d'un modèle afin de le rendre plus parcimonieux. Dans une formulation utilisant la pénalisation *ridge*, un paramètre d'intensité de la pénalisation doit être sélectionné, λ. Ce paramètre joue le même rôle que σ qui contrôle la dispersion des coefficients dans la formulation par effets aléatoires. λ et σ sont conceptuellement équivalents.

Notez que cette approche exclut les vecteurs propres présentant une autocorrélation spatiale négative. De plus, contrairement à l'approche par sélection itérative, elle n'a pas pour objectif de retirer l'autocorrélation spatiale des résidus, bien qu'elle y parvienne souvent. Aussi, la pénalisation des coefficients par la méthode des effets aléatoires ne constitue pas une technique de sélection de variables. Elle permet de réduire le nombre de degrés de liberté effectifs du modèle (un

effet aléatoire ne représentant pas exactement un effet fixe), sans pour autant retirer réellement les vecteurs propres les moins pertinents.

7.2.3 Sélection par lasso

Les régressions de type lasso intègrent une pénalité de type $l1$ lors de l'estimation des coefficients du modèle afin de forcer ceux ayant le moins d'impact lors de l'ajustement du modèle à avoir une valeur de zéro. Contrairement à la pénalisation de type $l2$ (*ridge*) mentionnée plus haut, la méthode lasso peut plus directement contraindre un coefficient à avoir une valeur nulle.

La méthode par lasso a été proposée comme alternative à la sélection pas à pas (*step-wise*), notamment pour réduire le temps de calcul (Seya et al. 2015). Contrairement à l'approche pas à pas, l'approche lasso permet d'ajuster une seule fois le modèle, tout en sélectionnant directement les vecteurs propres pertinents. Notez que la pénalisation par lasso n'est appliquée qu'aux coefficients des vecteurs propres spatiaux.

7.3 Mise en œuvre et analyse dans R

Pour illustrer l'utilisation des modèles SEVM, nous utilisons le même exemple que pour les modèles GAM (section 6.5). Nous avons mentionné un peu plus haut que ces deux types de modèles se ressemblaient beaucoup dans leur façon d'intégrer l'espace dans leur formulation.

Nous modélisons donc une fois encore le log du ratio entre les parts modales du transport collectif et de la voiture pour les déplacements pendulaires pour les secteurs de recensement de la région métropolitaine de Montréal en 2021.

Afin d'éviter les répétitions, nous présentons directement les différents modèles SEVM sans reprendre le modèle GLM non spatial.

7.3.1 Modèle SEVM par sélection itérative

La première approche pour développer un SEVM consiste à sélectionner les vecteurs propres spatiaux à l'aide de la méthode décrite à la section 7.2.1. L'approche est notamment implémentée dans le *package* `spatialreg`.

```
library(sf, quietly = TRUE)
library(ggplot2, quietly = TRUE)
library(tmap, quietly = TRUE)
library(dplyr, quietly = TRUE)
library(ggpubr, quietly = TRUE)
library(spdep, quietly = TRUE)
library(spatialreg, quietly = TRUE)
library(DHARMa, quietly = TRUE)
library(adespatial, quietly = TRUE)
library(mgcv, quietly = TRUE)
library(gamlss, quietly = TRUE)

# Chargement des données
```

```
data_mtl <- st_read('data/chap06/data_sr_access.gpkg', quiet = TRUE)

# Les variables d'accessibilité sont multipliées par 100 pour s'exprimer
# sur l'échelle 0-100 plutôt que 0-1
data_mtl$acs_idx_emp_pieton <- data_mtl$acs_idx_emp_pieton * 100
data_mtl$acs_idx_emp_tc_peak <- data_mtl$acs_idx_emp_tc_peak * 100

# Nous préparons ici les données en calculant les ratios
# remplaçant les 0 et calculant la transformation alr
data_mtl <- data_mtl %>%
  mutate(
    prt_tc = mode_tc / total_commuters,
    prt_auto = mode_auto / total_commuters,
  ) %>%
  filter(!is.na(prt_tc)) %>%
  mutate(
    prt_tc = ifelse(prt_tc == 0, min(prt_tc[prt_tc>0]) ,prt_tc)
  ) %>%
  mutate(
    ratio = (prt_tc / prt_auto),
    alr_val = log(prt_tc / prt_auto)
  ) %>%
  st_transform(32188)
```

La première étape consiste à créer notre matrice de contiguïté standardisée en ligne. Nous pouvons ensuite utiliser la fonction `ME` qui implémente un algorithme de sélection similaire à celui originalement proposé (Griffith et Peres-Neto 2006). Notez que la fonction `ME` n'ajuste pas le modèle final, mais fait seulement la sélection des vecteurs propres.

```
# Construction de la matrice de contiguïté
nbs <- poly2nb(data_mtl, queen = TRUE)
listw <- nb2listw(nbs, style = 'W', zero.policy = TRUE)
```

```
# Sélection des vecteurs propres de Moran
moran_ev_sel <- ME(alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
                   acs_idx_emp_tc_peak + acs_idx_emp_pieton,
                   data = data_mtl,
                   family = gaussian,
                   listw = listw,
                   alpha = 0.05,
                   zero.policy=TRUE)

# Nombre de vecteurs propres
ncol(moran_ev_sel$vectors)
```

La fonction ME a sélectionné 46 vecteurs propres que nous pouvons utiliser pour ajuster le modèle complet.

```
# Construction du modèle
model_sel <- glm(alr_val ~ prt_minorite_vis + prt_monoparental + revenu_median +
                 acs_idx_emp_tc_peak + acs_idx_emp_pieton +
                 fitted(moran_ev_sel),
                 data = data_mtl,
                 family = gaussian)

# Validation des résidus
res <- simulateResiduals(model_sel, plot = FALSE)
plot(res)

# Autocorrélation spatiale des résidus
moran.mc(residuals(res), listw = listw, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(res)
weights: listw
number of simulations + 1: 1000

statistic = 0.084006, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

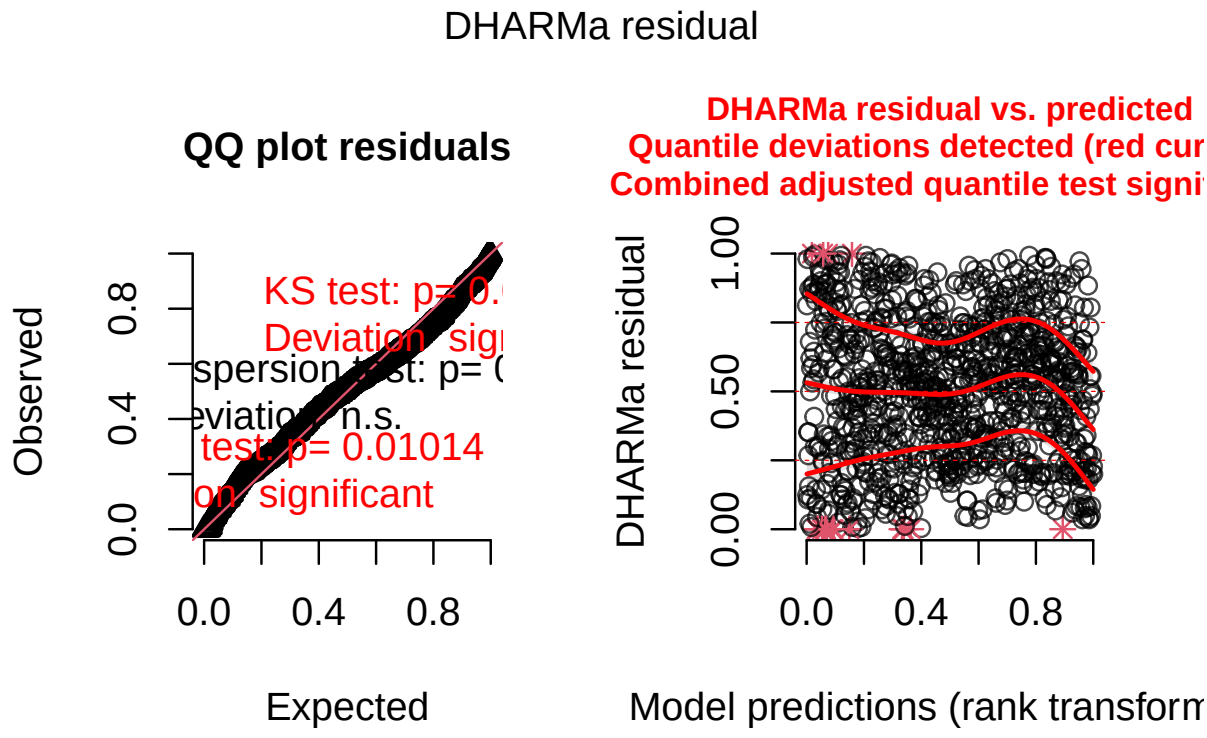


Figure 7.3 – Résidus simulés du modèle SEVM par sélection

La figure 7.3 montre que les résidus simulés ne suivent pas une distribution uniforme. Nous avions rencontré un enjeu similaire lorsque nous avions ajusté les modèles GAM dans la section 6.5. Nous avions notamment déterminé qu'une distribution de Student était plus adaptée qu'une simple distribution normale pour notre GLM. La fonction `ME` permet de sélectionner les vecteurs propres spatiaux pour un nombre limité de GLM. Pour pouvoir appliquer l'approche de sélection à d'autres types de modèles, il est nécessaire de l'implémenter manuellement. Nous répliquons ici l'approche décrite à la section 7.2.1.

```
# Précalcul des vecteurs propres
moran_ev <- mem(listw, MEM.autocor = 'all')

# Ajout des vecteurs propres dans les données originales
data_mtl2 <- cbind(data_mtl, moran_ev)

# Calcul de la corrélation entre Y et les vecteurs propres pour les intégrer
# selon l'ordre de leur corrélation avec Y
corr_vals <- sapply(names(moran_ev), function(x){
  cor(
    x = data_mtl2$alr_val,
    y = data_mtl2[[x]])
})
corr_vals <- abs(corr_vals)

vector_names <- names(moran_ev)
vector_names <- vector_names[order(corr_vals, decreasing = TRUE)]

# Préparation de l'équation de base de notre modèle

base_formula <- "alr_val ~ prt_minorite_vis +
                          prt_monoparental + revenu_median +
                          acs_idx_emp_tc_peak + acs_idx_emp_pieton"

# Nous ajustons aussi un modèle de base de référence
# qui sera comparé aux différentes propositions
base_model <- gam(alr_val ~ prt_minorite_vis +
                           prt_monoparental + revenu_median +
                           acs_idx_emp_tc_peak + acs_idx_emp_pieton,
                  family = scat,
                  data = data_mtl2)

# Calcul du I de Moran du modèle de référence
resid <- residuals(base_model, type = "pearson")
S0 <- sum(sapply(listw$weights, sum))
moran_ref <- moran(resid, listw,
                   n = length(resid), S0 = S0, zero.policy = TRUE)$I

# Préparation d'une liste vide dans laquelle nous stockerons les
# noms des vecteurs propres à ajouter
```

```
vecteurs_retenus <- list()

# Lancement des itérations sur chacun des vecteurs
for(vec in vector_names){
  # pour chaque itération, nous devons proposer une version
  # modifiée de la formule
  candidate_formula <- as.formula(paste0(base_formula, '+', vec))
  # avec cette formule, nous pouvons ajuster notre modèle GLM
  model <- gam(candidate_formula,
               family = scat,
               data = data_mtl2)
  # Nous regardons en premier si le fit du modèle est meilleur avec
  # un test de rapport de vraissemblance
  test1 <- anova(model, base_model, test = 'Chisq')
  p_val <- test1$`Pr(>Chi)`[[2]]

  if(is.na(p_val)){
    p_val <- 1
  }

  # si le test est significatif, nous pouvons ensuite tester
  # l'autocorrélation spatiale des résidus
  if(p_val < 0.01){

    resid <- residuals(model, type = "pearson")
    moran_value <- moran(resid,
                         listw = listw,
                         n = length(resid),
                         S0 = S0,
                         zero.policy = TRUE)$I
    diff <- moran_ref - moran_value

    # si le I de Moran est réduit d'au moins 0.0005, alors on garde le vecteur
    # propre dans le modèle.
    if(diff > 0.0005){
      # on doit alors mettre à jour notre formule de base
      base_formula <<- paste0(base_formula, '+',vec)

      # on met à jour notre modèle de base
      base_model <<- model

      # on met à jour la valeur de I de Moran de référence
      moran_ref <<- moran_value

      # on va aussi stocker quelques informations sur les vecteurs ajoutés
      vecteurs_retenus[[length(vecteurs_retenus) + 1]] <- list(
```

```
        'vecteur' = vec,
        'new_moran' = moran_value,
        'new_aic' = model$aic
      )
    }
  }
}
retenu <- data.frame(do.call(rbind, vecteurs_retenus))
```

Tableau 7.1 – Vecteurs propres retenus

Vecteur propre	I de Moran	AIC
MEM4	0,50	1 392,86
MEM3	0,47	1 350,14
MEM8	0,47	1 344,17
MEM5	0,44	1 298,24
MEM1	0,42	1 278,19
MEM9	0,42	1 265,54
MEM17	0,41	1 241,89
MEM15	0,39	1 196,29
MEM16	0,38	1 189,13
MEM43	0,35	1 160,88
MEM19	0,34	1 143,41
MEM121	0,33	1 136,94
MEM20	0,32	1 124,48
MEM6	0,32	1 119,16
MEM115	0,31	1 100,44
MEM22	0,29	1 053,37
MEM106	0,29	1 042,85
MEM39	0,28	1 034,37
MEM107	0,28	1 024,68
MEM95	0,27	1 012,36
MEM28	0,27	1 007,68
MEM50	0,25	981,69
MEM38	0,24	957,51
MEM14	0,23	949,84
MEM53	0,22	943,90
MEM172	0,22	935,87
MEM13	0,21	925,88
MEM24	0,21	916,87
MEM54	0,20	906,48
MEM73	0,19	897,24
MEM37	0,19	892,23
MEM27	0,18	886,52
MEM31	0,17	874,96

Tableau 7.1 – Vecteurs propres retenus

Vecteur propre	I de Moran	AIC
MEM40	0,14	829,67
MEM101	0,14	824,92
MEM44	0,13	810,20
MEM210	0,13	804,94
MEM48	0,12	783,58
MEM248	0,12	777,71
MEM111	0,10	752,07
MEM94	0,10	740,48
MEM25	0,09	720,77
MEM45	0,09	713,39
MEM186	0,09	704,67
MEM135	0,08	697,60

La procédure que nous avons appliquée a retenu 45 vecteurs propres (tableau 7.1) et est parvenue à réduire très significativement le I de Moran des résidus (I = 0,158).

```
# Validation brève des résidus
res <- simulateResiduals(base_model, plot = FALSE)
plot(res)

# I de Moran sur les résidus du modèle
moran.mc(residuals(res), listw = listw, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(res)
weights: listw
number of simulations + 1: 1000

statistic = 0.082123, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

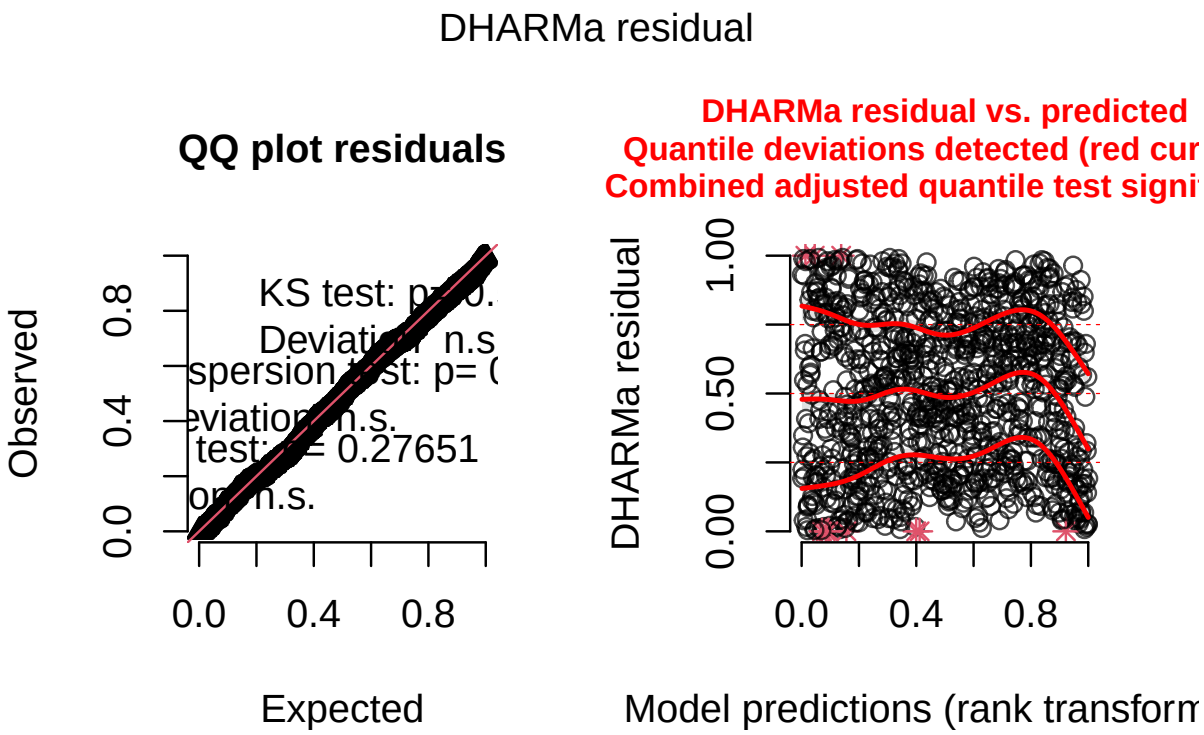


FIGURE 7.4 – Résidus simulés du modèle SEVM avec distribution de Student par sélection pas à pas

Nous notons une nette amélioration dans la distribution des résidus (figure 7.4). Nous effectuerons l'interprétation des résultats de ce modèle en comparaison avec ceux obtenus par les deux autres versions (section 7.3.4).

7.3.2 Modèle SEVM avec effet aléatoire

Pour rappel, le modèle SEVM par effet aléatoire consiste à ne retenir que les vecteurs propres avec de l'autocorrélation positive et à pénaliser les coefficients des vecteurs propres en les modélisant comme un effet aléatoire provenant d'une distribution normale centrée sur 0. Cette approche est plus difficile à appliquer dans R, notamment parce qu'elle relève intrinsèquement d'une conception bayésienne du modèle. Le *package* `spmoran` offre une implémentation, mais celle-ci se limite principalement sur le modèle linéaire classique (distribution gaussienne des résidus). Toutefois, `spmoran` peut être utilisé pour modéliser d'autres types de distribution, mais il utilise pour cela des fonctions de lien complexes transformant la variable dépendante pour garantir qu'elle suit une distribution normale. Cette approche rend l'interprétation des effets des variables indépendantes très difficile.

Cependant, comme nous l'avons mentionné plus haut (section 7.2.2), la pénalisation des coefficients d'un modèle induite par leur introduction comme un effet aléatoire est équivalente à leur pénalisation par la méthode *ridge* (*l1 regularization*). Il existe plusieurs *packages* dans R permettant d'implémenter ce type de pénalisation dans des GLM, dont `glmnet`, `ridge`, et `gamlss`. Le package `mgcv` que nous avons présenté dans la section sur les modèles GAM est également capable d'inclure une pénalité de type *ridge* sur les coefficients des termes linéaires. Il faut pour cela utiliser un paramètre méconnu appelé `paraPen`. Considérant que la distribution de Student a donné de bons résultats jusqu'ici avec la fonction `gam` et le paramètre `family = scat`, nous allons continuer à l'utiliser. De plus, le package `mgcv` est capable de déterminer automatiquement le paramètre λ calibrant la force de la pénalisation.

```
# Nous devons commencer par sélectionner tous les vecteurs avec un I de Moran positif
# Précalcul des vecteurs propres
moran_ev <- mem(listw, MEM.autocor = 'all')

# Pour rappel, cette méthode limite les vecteurs propres à ceux ayant une autocorrélation
# spatiale positive, nous allons donc retenir ici les 100 premiers vecteurs
ok_vectors <- as.matrix(moran_ev[,1:100])
```

Nous calculons ensuite le modèle avec la fonction `gam` du package `mgcv`.

```
model_ridge <- gam(alr_val ~ prt_minorite_vis +
                      prt_monoparental + revenu_median +
                      acs_idx_emp_tc_peak + acs_idx_emp_pieton +
                      ok_vectors,
                   family = scat,
                   paraPen = list('ok_vectors' = list(diag(ncol(ok_vectors)))),
                   method = "ML",
                   data = data_mtl)
```

```
# Validation brève des résidus
res <- simulateResiduals(model_ridge, plot = FALSE)
plot(res)

# I de Moran sur des résidus
moran.mc(residuals(res), listw = listw, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(res)
weights: listw
number of simulations + 1: 1000

statistic = 0.054179, observed rank = 998, p-value = 0.002
alternative hypothesis: greater
```

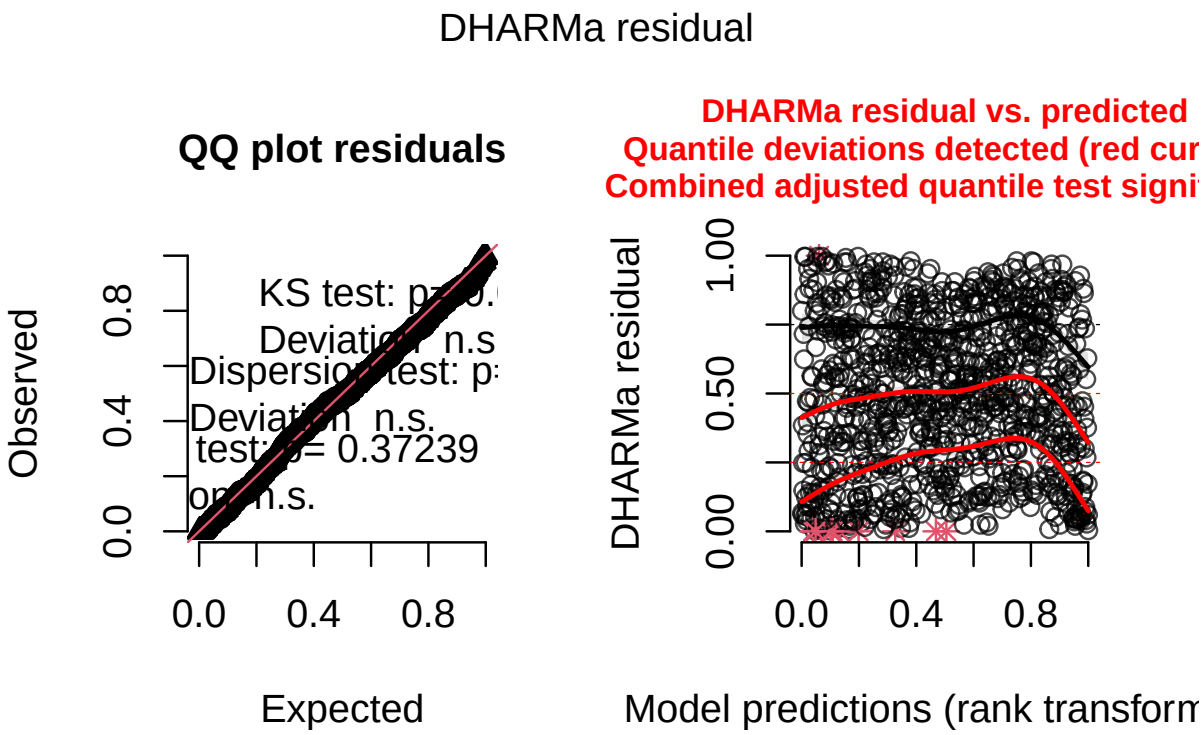


FIGURE 7.5 – Résidus du modèle RE-SEVM

Les résidus du modèle SEVM (figure 7.5) avec effets aléatoires sont assez proches d'une distribution uniforme et n'ont plus d'autocorrélation spatiale (I = 0,054).

7.3.3 Modèle SEVM avec pénalisation lasso

Le *package* de prédilection pour créer des modèles avec des pénalisations est `glmnet`, car il permet notamment d'intégrer des pénalisations de type lasso, *ridge*, et *elastic-net*. Cependant, les distributions supportées par `glmnet` sont assez limitées. Il est aussi possible d'utiliser le *package* `gamlss` qui est beaucoup plus flexible et qui peut utiliser les fonctions de `glmnet` pour introduire des pénalités. L'approche ici est similaire à celle présentée dans la section précédente. Nous commençons par précalculer les vecteurs propres et retenir uniquement les vecteurs avec une autocorrélation spatiale positive. Nous introduisons ensuite ces vecteurs avec une pénalisation sur leurs coefficients dans le modèle. Dans le package `gamlss`, la famille `TF` est comparable à la famille `scat` de `mgcv`. Aussi, `gamlss` est capable de choisir le meilleur degré de pénalisation selon un critère (`ML` pour maximum de vraisemblance et `GAIC` pour AIC généralisé) sélectionné par la personne utilisatrice. Notez que la fonction `ri` utilisée ici permet aussi d'ajuster une pénalisation de type *ridge* avec le paramètre `Lp = 2`.

```
# Nous devons commencer par sélectionner tous les vecteurs
# avec un I de Moran positif
# Précalcul des vecteurs propres
moran_ev <- mem(listw, MEM.autocor = 'all')

# Pour rappel, cette méthode limite les vecteurs propres à ceux ayant une
# autocorrélation spatiale positive, nous allons donc retenir ici les
# 100 premiers vecteurs
```

```
ok_vectors <- as.matrix(moran_ev[,1:100])
data_mtl2 <- cbind(data_mtl, ok_vectors)
mem_vars <- colnames(ok_vectors)
model_lasso <- gamlss(alr_val ~ prt_minorite_vis +
                      prt_monoparental + revenu_median +
                      acs_idx_emp_tc_peak + acs_idx_emp_pieton +
                      ri(x.vars = mem_vars,
                         method = 'GAIC',
                         Lp = 1),
                     family = TF,
                     control = gamlss.control(n.cyc = 50),
                     data = data_mtl2)
```

```
# Pour utiliser le package DHARMa et visualiser les résidus simulés d'un modèle
# GAMLSS, il faut extraire manuellement des simulations du modèle.
# Nous allons extraire ici 1000 simulations

# Extraction des paramètres prédits par le modèle :
# - l'espérance
pred_mu <- predict(model_lasso, what = 'mu', type = 'response')
# - la variance
pred_sigma <- predict(model_lasso, what = 'sigma', type = 'response')
# - le nombre de degrés de libertés
pred_nu <- predict(model_lasso, what = 'nu', type = 'response')
# Réalisation de 1000 simulations
nsim <- 1000
sims <- t(sapply(1:length(pred_mu), function(i){
  sim <- rTF(nsim, mu = pred_mu[[i]],
             sigma = pred_sigma[[i]],
             nu = pred_nu[[i]])
  return(sim)
}))

res <- createDHARMa(simulatedResponse = sims,
                    observedResponse = data_mtl2$alr_val,
                    fittedPredictedResponse = pred_mu
                    )
plot(res)
# validation brève des résidus
moran.mc(residuals(res), listw = listw, nsim = 999)
```

```
    Monte-Carlo simulation of Moran I

data:  residuals(res)
weights: listw
```

```
number of simulations + 1: 1000

statistic = 0.039251, observed rank = 980, p-value = 0.02
alternative hypothesis: greater
```

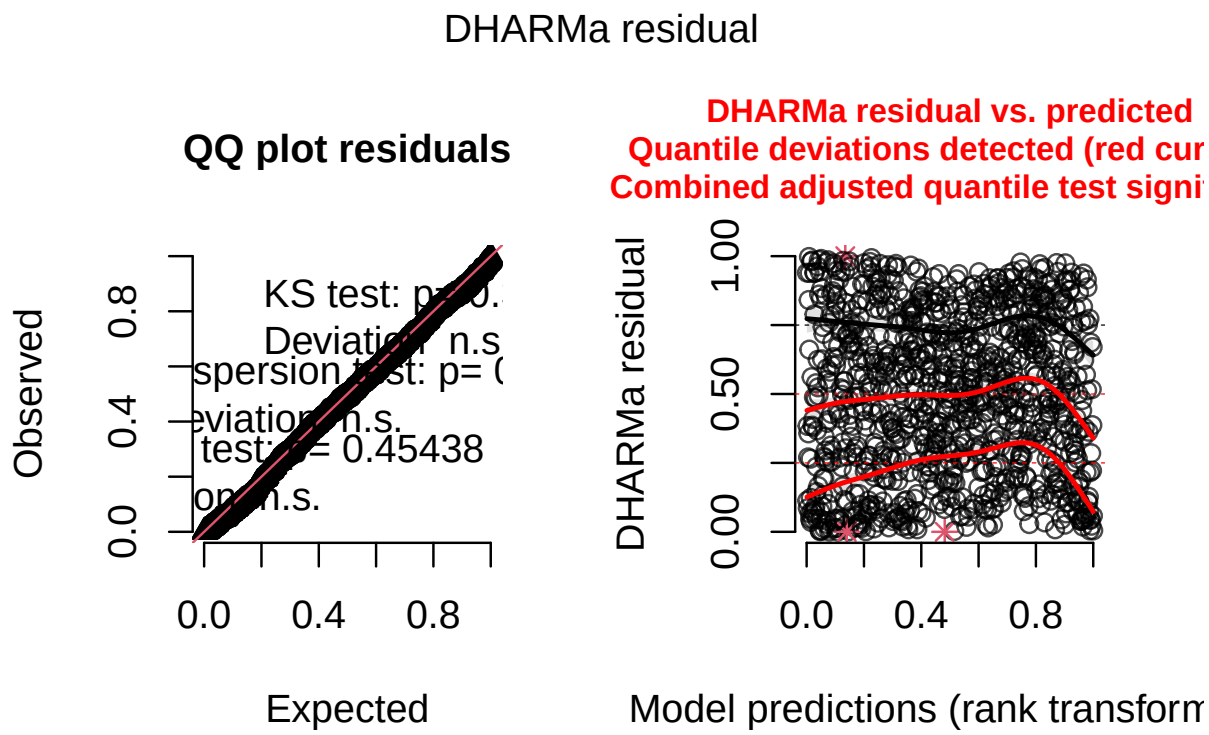


Figure 7.6 – Résidus du modèle lasso-SEVM

Le modèle semble également avoir des résidus correctement ajustés et ne présentant plus de dépendance spatiale (I = 0,040).

7.3.4 Comparaison des trois modèles SEVM

Nous avons ajusté trois modèles implémentant l'approche de *spatial filtering* et comparons ici leurs résultats. Nous regardons d'abord les coefficients des variables indépendantes estimés par les modèles, puis nous comparons les termes spatiaux.

Il est intéressant de noter que les trois modèles ne produisent pas exactement les mêmes résultats. Bien qu'ils renvoient des effets sensiblement de mêmes ordres de grandeur et de mêmes directions, nous observons en particulier que :

— Le modèle par sélection itérative indique un effet du pourcentage de minorités visibles beaucoup plus faible que les deux autres modèles. Une augmentation de 10 points de pourcentage des minorités visibles est associée à une multiplication par 1,096 du ratio entre les parts modales du transport en commun et de l'automobile. Pour le modèle lasso, la multiplication serait de 1,144.

Tableau 7.2 – Comparaison des trois modèles SEV

	Sélection itérative		Effet aléatoire		Lasso	
	coef.	p	coef.	p	coef.	p
Constante	-3,104	***	-2,970	***	-2,809	***
Minorités visibles (%)	0,914	***	1,476	***	1,347	***
Ménages monoparentaux (%)	1,393	***	0,999	***	1,012	***
Revenu médian (1000 $)	-0,007	***	-0,007	***	-0,008	***
Accessibilité aux emplois en TC (%)	0,051	***	0,041	***	0,037	***
Accessibilité aux emplois à pied (%)	-0,008	***	-0,004		-0,004	*

I de Moran sur les résidus : sélection pas à pas (0,082), effet aléatoire (0,054), Lasso (0,039).

— Le modèle utilisant la méthode des effets aléatoires indique que l'accessibilité aux emplois à pied n'a pas d'effet significatif. Le modèle utilisant la pénalisation lasso indique une faible significativité (< 0,05) et un coefficient deux fois plus petit que le modèle par sélection itérative.
— Les trois modèles s'accordent globalement sur l'impact du revenu médian. Une augmentation de 1000 $ de ce dernier dans un secteur de recensement (SR) est associé à une réduction de 0,73 % du ratio des parts modales du transport en commun et de l'automobile pour le premier modèle et de 0,72 % et 0,78 % pour les deux autres modèles.
— Sans surprise, l'accessibilité aux emplois en transport collectif a un impact important et significatif dans les trois modèles. Une augmentation de 10 points de cet indicateurs (allant de 0 à 100) est associée à une augmentation de la moyenne du ratio entre les parts modales TC et automobile de 65,9 %, 50,7 % et 44,8 % respectivement.

Une partie des différences dans les résultats obtenus s'explique par une prise en compte différente de l'espace par les trois modèles. Nous pouvons représenter les trois termes spatiaux en utilisant la méthode des effets marginaux. Afin de comparer facilement les trois effets, nous représentons les trois cartes avec la même légende.

```
#save(base_model, model_ridge, model_lasso,
#     model_sel,retenu,moran_ev_sel, data_mtl2,
#     file = 'data/chap07/models_sevm.rda')

#_____________________________________________________
# Extraction du terme spatial du modèle par sélection
# Préparation du jeu de données avec tous les prédicteurs à 0
# sauf les vecteurs propres spatiaux
pred_df <- cbind(data_mtl, moran_ev)
pred_df$prt_minorite_vis <- 0
pred_df$prt_monoparental <- 0
pred_df$revenu_median <- 0
pred_df$acs_idx_emp_tc_peak <- 0
pred_df$acs_idx_emp_pieton <- 0
# Réalisation de la prédiction et retrait de la constante
pred_sel <- predict(base_model, newdata = pred_df)
pred_sel <- pred_sel - base_model$coefficients[[1]]
# Transformation avec la fonction exp pour passer du log ratio au simple ratio
pred_sel <- exp(pred_sel)
```

```
#_____________________________________________________
# Extraction du terme spatial du modèle par effet aléatoire
# préparation du jeu de données avec tous les prédicteurs à 0
# sauf les vecteurs propres spatiaux
pred_df <- data_mtl
pred_df$prt_minorite_vis <- 0
pred_df$prt_monoparental <- 0
pred_df$revenu_median <- 0
pred_df$acs_idx_emp_tc_peak <- 0
pred_df$acs_idx_emp_pieton <- 0
pred_df$ok_vectors <- as.matrix(moran_ev[,1:100])
# Réalisation de la prédiction et retrait de constante
pred_ridge <- predict(model_ridge, newdata = pred_df)
pred_ridge <- pred_ridge - model_ridge$coefficients[[1]]
# Transformation avec la fonction exp pour passer du log ratio au simple ratio
pred_ridge <- exp(pred_ridge)

#_____________________________________________________
# Extraction du terme spatial du modèle par lasso
# Préparation du jeu de données avec tous les prédicteurs à 0
# sauf les vecteurs propres spatiaux
pred_df <- cbind(data_mtl, moran_ev)
pred_df$prt_minorite_vis <- 0
pred_df$prt_monoparental <- 0
pred_df$revenu_median <- 0
pred_df$acs_idx_emp_tc_peak <- 0
pred_df$acs_idx_emp_pieton <- 0
# pred_df$ok_vectors <- as.matrix(moran_ev[,1:100])
mem_vars <- colnames(ok_vectors)
# Réalisation de la prédiction et retrait de la constante
pred_lasso <- predict(model_lasso, newdata = pred_df, what = 'mu')
pred_lasso <- pred_lasso - model_lasso$mu.coefficients[[1]]
# Transformation avec la fonction exp pour passer du log ratio au simple ratio
pred_lasso <- exp(pred_lasso)

data_mtl$effet_sel <- pred_sel
data_mtl$effet_ridge <- pred_ridge
data_mtl$effet_lasso <- pred_lasso

# Détermination d'une palette de couleurs pour la légende
library(classInt)
breaks <- classIntervals(c(pred_sel,
                           pred_ridge,
                           pred_lasso),
                         style = "fisher",
                         n = 7)
```

```
legende_parametres <- list(text.separator = "à",
                           decimal.mark = ",",
                           digits = 2)

carte1 <- tm_shape(data_mtl) +
  tm_fill(col="effet_sel",
          breaks = breaks$brks,
          midpoint = 1,
          legend.format = legende_parametres,
          palette = "-RdBu") +
  tm_layout(frame=FALSE, legend.show = FALSE, title = 'SEL-SEVM')

carte2 <- tm_shape(data_mtl) +
  tm_fill(col="effet_ridge",
          breaks = breaks$brks,
          midpoint = 1,
          legend.format = legende_parametres,
          palette = "-RdBu") +
  tm_layout(frame=FALSE, legend.show = FALSE, title = 'RE-SEVM')

carte3 <- tm_shape(data_mtl) +
  tm_fill(col="effet_lasso",
          breaks = breaks$brks,
          midpoint = 1,
          legend.format = legende_parametres,
          palette = "-RdBu") +
  tm_layout(frame=FALSE, legend.show = FALSE, title = 'LASSO-SEVM')

# Cette carte est ajoutée pour ne pas apparaître mais afficher la légende
# commune dans le 4ème panneau
carte4 <- tm_shape(data_mtl) +
  tm_fill(col="effet_lasso",
          breaks = breaks$brks,
          midpoint = 1,
          legend.format = legende_parametres,
          palette = "-RdBu", title = 'effet spatial') +
  tm_layout(legend.only = TRUE,scale=1.5, asp=0)

tmap_arrange(carte1, carte2, carte3, carte4, nrow = 2, ncol = 2)
```

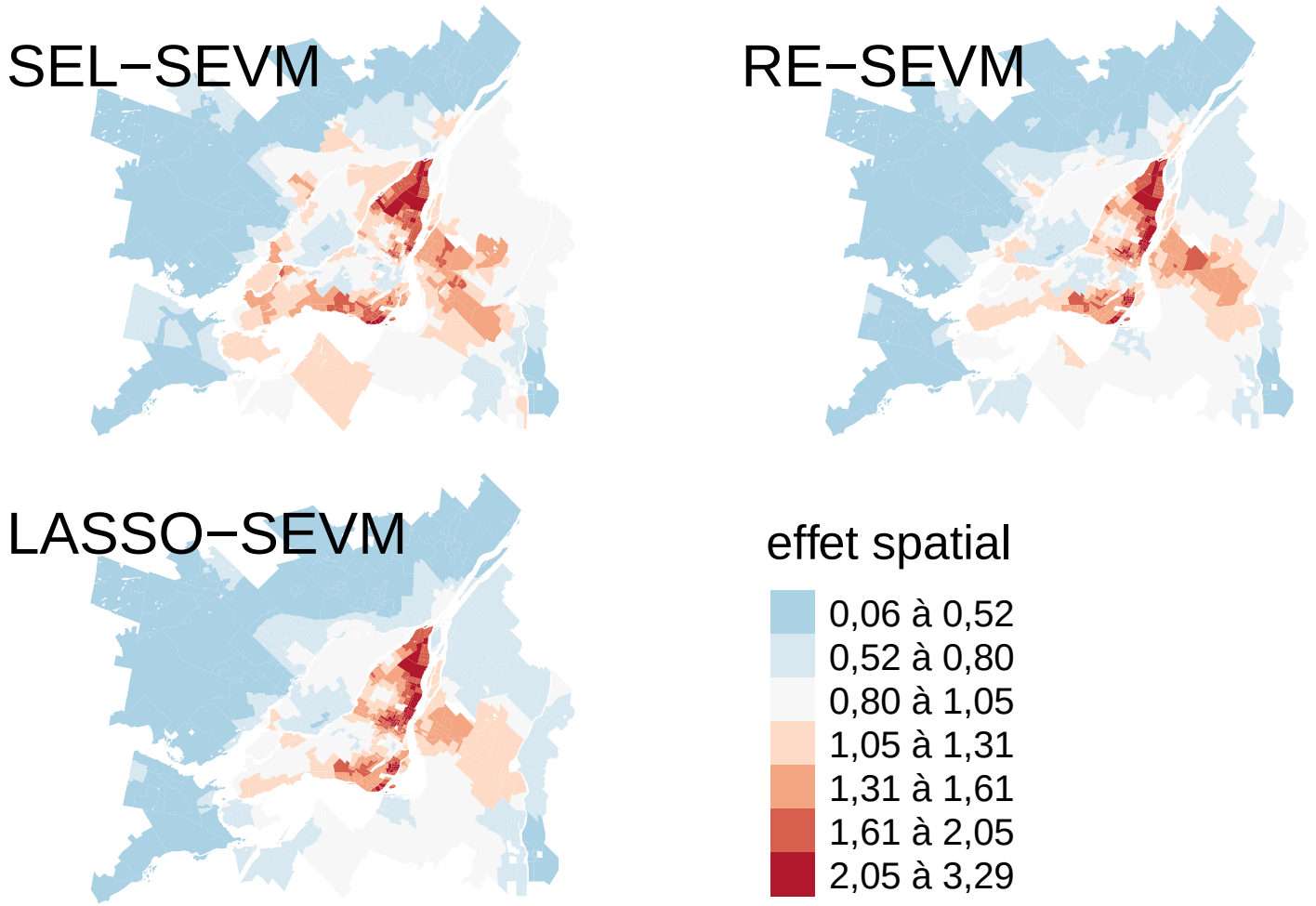


Figure 7.7 – Comparaison des termes spatiaux des trois modèles SEVM

Les effets spatiaux capturés par les trois modèles sont très similaires (figure 7.7). La corrélation de Pearson est cependant beaucoup plus élevée entre le terme spatial des modèles RE-SEVM et LASSO-SEVM (0,98) qu'avec le modèle SEL-SEVM (0,8). Une différence notable s'observe pour l'arrondissement du Plateau-Mont-Royal : les modèles RE-SEVM et LASSO-SEVM ont des termes spatiaux avec des valeurs bien plus élevées pour ce secteur dans lequel la part modale active (marche + vélo) est très élevée. Cela explique probablement la réduction de l'effet de la variance indépendante d'accessibilité à pied aux emplois dans ces modèles.

Les conclusions que nous pouvons tirer des trois modèles sont donc similaires, mais avec tout de même des différences non négligeables. Ces différences s'expliquent par la façon dont les vecteurs propres spatiaux ont été intégrés dans le modèle. Notons que les approches par effets aléatoires et pénalisation lasso ont tendance à être plus rapides à estimer et reposent sur un fondement théorique plus défendable. En effet, leurs pénalisations respectives peuvent être interprétées comme des a priori bayésiens utilisés pour réguler le modèle. La méthode par sélection correspond plus à une démarche ad hoc, mais qui a été utilisée à de nombreuses reprises et est relativement facile à implémenter avec n'importe quel type de modèle puisque les vecteurs propres spatiaux sont introduits comme de simples effets fixes.

7.4 Quiz de révision

Questions

- **Comment un modèle SEVM intègre-t-il les effets spatiaux?**
  - En ajustant la distribution conditionnelle supposée de la variable dépendante pour intégrer la dépendance spatiale.
  - En ajoutant au modèle des variables spatialement décalées.
  - En ajoutant au modèle des vecteurs propres spatiaux.

  Relisez au besoin la section 7.2.
- **Qu'est-ce qu'un vecteur propre spatial?**
  - Une variable quantitative continue
  - Une variable construite à partir d'une matrice de pondération spatiale
  - Une variable avec une autocorrélation spatiale positive

  Relisez au besoin la section 7.1.
- **En quoi consiste l'approche classique de construction d'un modèle SEVM?**
  - Elle applique un processus itératif de sélection des vecteurs propres à ajouter dans le modèle.
  - Elle sélectionne uniquement les vecteurs propres améliorant le R2 du modèle.
  - Elle sélectionne uniquement les vecteurs propres avec une autocorrélation spatiale positive.

  Relisez au besoin la section 7.2.1.
- **L'approche LASSO et l'approche par effet aléatoire ont un point commun. Lequel?**
  - Elles sont plus lentes à calculer que la méthode par sélection
  - Elles retiennent les vecteurs propres pertinents en incluant une pénalisation
  - Elles sont moins efficaces pour capturer la corrélation spatiale dans les résidus

  Relisez au besoin la section 7.2.2 et la section 7.2.3.
- **Comment peut-on interpréter les effets spatiaux dans un modèle SEVM?**
  - En analysant les coefficients des vecteurs spatiaux intégrés au modèle.
  - En analysant le degré de pénalisation sélectionné par la méthode GAIC.
  - En cartographiant les effets spatiaux par la méthode des effets marginaux.

  Relisez au besoin la section 7.3.4.

Réponses

- Comment un modèle SEVM intègre-t-il les effets spatiaux?
  - En ajoutant au modèle des vecteurs propres spatiaux.
- Qu'est-ce qu'un vecteur propre spatial?
  - Une variable quantitative continue
  - Une variable construite à partir d'une matrice de pondération spatiale
- En quoi consiste l'approche classique de construction d'un modèle SEVM?
  - Elle applique un processus itératif de sélection des vecteurs propres à ajouter dans le modèle.
- L'approche LASSO et l'approche par effet aléatoire ont un point commun. Lequel?
  - Elles retiennent les vecteurs propres pertinents en incluant une pénalisation
- Comment peut-on interpréter les effets spatiaux dans un modèle SEVM?

— En cartographiant les effets spatiaux par la méthode des effets marginaux.

Partie 5. Régressions spatiales et hétérogénéité spatiale

8 Régressions géographiquement pondérées classiques

La régression géographiquement pondérée (*geographically weighted regression* - **GWR**, en anglais) a été formalisée au milieu des années 1990 par Chris Brunsdon, Steward Fotheringham et Martin Charlton (1996), puis largement décrite dans un ouvrage de référence (Fotheringham, Brunsdon et Charlton 2003). Dans le cadre de ce chapitre, nous abordons uniquement la GWR classique qui s'applique à une variable dépendante continue (GWR gaussienne).

Objectif

Objectifs d'apprentissage visés dans ce chapitre
À la fin de ce chapitre, vous devriez être en mesure de :

- comprendre pourquoi utiliser une GWR;
- assimiler les principes fondamentaux d'une GWR classique (distribution gaussienne);
- analyser les résultats produits par une GWR;
- comprendre les limites et critiques de la GWR;
- mettre en pratique une GWR dans R.

Package

Liste des *packages* utilisés dans ce chapitre

- Pour importer et manipuler des fichiers géographiques :
  - `sf` pour importer et manipuler des données vectorielles.
  - `dplyr` pour manipuler les données.
- Pour construire des cartes et des graphiques :
  - `tmap` pour les cartes.
  - `ggplot2` et `ggpubr` pour construire des graphiques.
  - `corrplot` et `GGally` pour créer des graphiques de matrices de corrélation.
- Pour construire des modèles de régression :
  - `spgwr` et `GWmodel` pour construire des GWR gaussienne, logistique et Poisson.
  - `spdep` pour construire des matrices de pondération spatiale et calculer le I de Moran.

8.1 Principe de base

8.1.1 Pourquoi recourir à une régression géographiquement pondérée?

Dans le chapitre 3, nous avons vu que les modèles d'économétrie spatiale visent à contrôler la **dépendance spatiale** d'un modèle de régression classique (MCO) afin d'améliorer l'estimation des coefficients de régression. L'objectif des modèles de régression géographiquement pondérée est différent : ils visent à analyser les variations spatiales de la relation entre la variable dépendante et les variables indépendantes. Autrement dit, les modèles GWR visent à explorer

l'instabilité spatiale du modèle MCO afin d'analyser localement la relation entre la variable dépendante et les variables indépendantes. Pour une description détaillée en français de la GWR, consultez Apparicio *et al.* (2007).

8.1.2 Formulation de la GWR

Contrairement à la régression linéaire classique et aux modèles spatiaux autorégressifs qui produisent une équation pour l'ensemble du tableau de données, la GWR produit une équation pour chaque unité spatiale *i* et ainsi des valeurs locales de R^2, β_0, β_k, *t* de Student, etc. La résolution de cette équation de régression locale est aussi basée sur la méthode des moindres carrés et sur une matrice de pondération spatiale ($\mathbf{W}(\mathbf{i})$) dont les valeurs décroissent en fonction de la distance séparant les entités spatiales *i* et *j*. Autrement dit, plus *j* est proche de *i*, plus sa pondération est élevée et donc plus son rôle dans la détermination de l'équation de régression locale de *i* est important.

De la sorte, la GWR est une extension de la régression linéaire multiple classique où (u_i, v_i) représente les coordonnées géographiques du centroïde de l'unité spatiale et où les paramètres β_0 et β_k peuvent varier dans l'espace (équation 8.1).

$$y_i = \beta_0(u_i, v_i) + \sum_{j=1}^{k} \beta_j(u_i, v_i)x_{ij} + \epsilon_i \tag{8.1}$$

avec :

- (u_i, v_i), les coordonnées géographiques de l'unité spatiale *i*.
- y_i, la variable dépendante pour l'unité spatiale *i*.
- $\beta_0(u_i, v_i)$, la constante pour l'unité spatiale *i* aux coordonnées géographiques (u_i, v_i).
- $\beta_j(u_i, v_i)$, le coefficient de régression pour la variable x_j (avec *k* variables indépendantes) pour l'unité spatiale *i* aux coordonnées géographiques (u_i, v_i).
- x_{ij}, la valeur de la variable indépendante x_j pour l'unité spatiale *i*.
- ϵ_i, le terme d'erreur pour l'unité spatiale *i*.

Fotheringham *et al.* (2003) proposent deux principales fonctions noyaux (*kernel*) pour définir la matrice de pondération spatiale $\mathbf{W}(\mathbf{i})$ dans le modèle GWR : une fonction gaussienne (équation 8.2) et une fonction bicarrée (équation 8.3) où d_{ij} représente la distance euclidienne entre les points *i* et *j* et *b*, le rayon de zone d'influence autour du point *i* (*bandwidth*). Il existe une différence fondamentale entre les deux : la fonction gaussienne accorde un poids non nul à tous les points de l'espace d'étude aussi loin soient-ils, tandis que la fonction bicarrée ne tient pas compte des points distants à plus de *b* mètres de *i*, tel qu'illustré à la figure 8.1 avec une valeur fixée à 5000 mètres en guise d'exemple.

$$w_{ij} = \exp\left(-\frac{1}{2}\left(\frac{d_{ij}}{b}\right)^2\right) \tag{8.2}$$

$$w_{ij} = \left(1 - \left(\frac{d_{ij}}{b}\right)^2\right)^2 \text{ si } d_{ij} < b, \text{ sinon } w_{ij} = 0 \tag{8.3}$$

Aller plus loin

Autres fonctions noyaux (*kernel*) pour la GWR

Outre les fonctions noyaux (*kernel*) gaussienne et bicarrée, il est aussi possible d'utiliser des fonctions exponentielle (équation 8.4), tricube (équation 8.5) et Boxcar (équation 8.6) implémentées dans les fonctions `bw.gw` et `gwr.basic` du *package* `GWmodel`.

$$w_{ij} = \exp\left(-\frac{d_{ij}}{b}\right) \tag{8.4}$$

$$w_{ij} = \left(1 - \left(\frac{d_{ij}}{b}\right)^3\right)^3 \text{ si } d_{ij} < b, \text{ sinon } w_{ij} = 0 \tag{8.5}$$

$$w_{ij} = 1 \text{ si } d_{ij} < b, \text{ sinon } w_{ij} = 0 \tag{8.6}$$

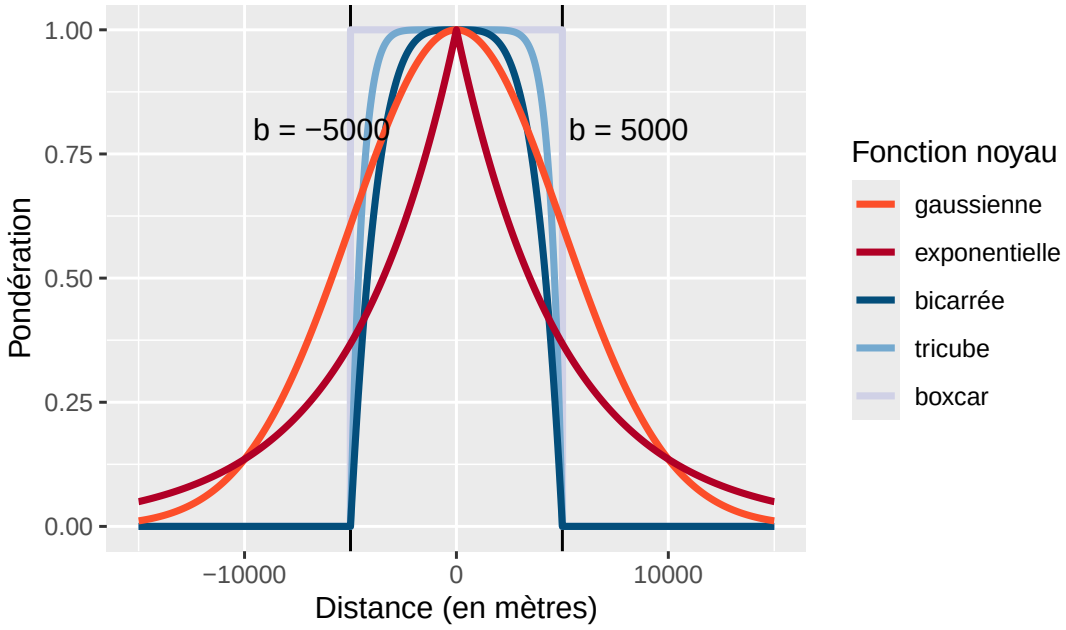


FIGURE 8.1 – Fonctions noyaux (*kernel*) pour définir la matrice de pondération W(i)

Dans le modèle GWR, la valeur de b est soit fixée par la personne utilisatrice, soit optimisée avec la valeur de CV (*cross-validation*) ou celle de l'AIC. Notez qu'il est possible d'optimiser la taille de la zone d'influence à partir de la distance (euclidienne le plus souvent, mais d'autres types de distances peuvent être utilisées) ou du nombre de plus proches voisins.

8.1.3 Exemple applicatif de la GWR

Pour démontrer l'utilité de la GWR comme outil d'exploration de l'instabilité spatiale d'un modèle classique, nous utilisons le jeu de données sur Lyon (section 1.1.1). Les modèles de régression (global et GWR) sont construits avec le dioxyde d'azote (NO_2) comme variable dépendante et cinq variables sociodémographiques comme variables indépendantes.

8.1.3.1 Résultats du modèle global

Les résultats de la régression linéaire multiple (modèle global) sont présentés au tableau 8.1. Les variables sociodémographiques (variables indépendantes) introduites dans le modèle expliquent 28,3 % de la variance du dioxyde d'azote (variable dépendante). La constante et tous les coefficients de régression du modèle sont significatifs (p < 0,05). Toutes choses étant égales par ailleurs, le pourcentage d'immigrants est associé positivement avec le dioxyde d'azote (β = 0,283), ce qui traduit une certaine iniquité environnementale pour ce groupe de population. À l'inverse, toutes les autres variables sont associées négativement.

Tableau 8.1 – Résultats du modèle global (régression linéaire multiple)

Variable	Coef.	Erreur type	t	p
Constante	49,433	2,995	16,502	0,000
Pct0_14	-0,534	0,063	-8,461	0,000
Pct_65	-0,150	0,056	-2,674	0,008
Pct_Img	0,283	0,051	5,532	0,000
Pct_brevet	-0,240	0,037	-6,451	0,000
NivVieMed	-0,316	0,102	-3,107	0,002

Nombre d'observations = 506. R carré = 0,2832. R carré ajusté = 0,276.

L'objectif de la construction d'un modèle GWR est d'examiner l'instabilité spatiale de ce modèle, en analysant comment les associations observées dans le modèle global entre les cinq variables sociodémographiques et le dioxyde d'azote varient dans l'espace.

8.1.3.2 Résultats du modèle GWR

Définition de la zone d'influence

Puisque la superficie des 506 IRIS varie considérablement (avec de petits IRIS au centre de l'agglomération et de très grands IRIS en périphérie, figure 1.1), nous préférons optimiser la zone d'influence (b) avec le nombre de plus proches voisins et non la distance euclidienne avec l'approche *cross-validation*. Le résultat final de l'optimisation du nombre de plus proches voisins est de 92.

Comparaison des modèles global et GWR

L'analyse de variance entre les résidus des deux modèles démontre que le modèle GWR améliore de façon significative la capacité prédictive du modèle de régression globale (tableau 8.2). Les valeurs de R^2 passent d'ailleurs de 0,283 pour le modèle global à 0,784 pour le modèle GWR.

Tableau 8.2 – Analyse de variance entre les modèles global et GWR

	dl	Somme des carrés	Moyenne des carrés	Valeur F
Résidus du modèle global	6,00	22 346,02		
Modèle GWR	101,13	15 581,60	154,08	
Résidus du modèle GWR	398,87	6 764,42	16,96	9,09

R carré du modèle global = 0,2832. R carré du modèle GWR = 0,7838.

Cartographie des valeurs de *t*

Afin de décrire les variations spatiales des associations entre la variable dépendante et les variables indépendantes, nous cartographions les valeurs de *t* pour chacune des cinq variables indépendantes (figure 8.2). Pour ce faire, nous utilisons les seuils de ± 1,96, 2,58 et 3,29, indiquant des seuils de signification à 5 %, 1 % et 0,1 %. La cartographie des valeurs de *t* illustre clairement les variations spatiales de l'association entre les variables indépendantes et le dioxyde d'azote.

Premièrement, le modèle global avait révélé que toutes les variables indépendantes étaient significativement associées au seuil de 0,1 %, toutes choses étant égales par ailleurs. Or, localement certaines variables ne sont plus significatives au seuil de 5 % (valeurs de *t* comprises entre -1,96 et 1,96), soit les IRIS en gris à la figure 8.2. Par exemple, le pourcentage d'enfants de moins de 15 ans est associé négativement au dioxyde d'azote uniquement dans la partie sud de l'agglomération.

Deuxièmement, le sens de la relation, c'est-à-dire du coefficient et de sa valeur de *t* associée, change pour certaines variables indépendantes. Cela est particulièrement le cas de la variable *médiane du niveau de vie (milliers d'euros)* qui affiche des valeurs positives dans le centre-ville, mais des valeurs négatives en périphérie. Autrement dit, une hausse du niveau de vie des résidents est associée à une réduction du dioxyde d'azote dans les quartiers et municipalités périphériques de l'agglomération, mais à une augmentation dans les quartiers centraux.

Cartographie du nombre de variables significatives et de la variable plus significative

Pour compléter l'exploration des résultats de la GWR, il est possible de répondre à deux dernières questions avec de nouvelles analyses cartographiques (figure 8.3) :

— Quelle est la variable indépendante la plus significative localement?
— Combien de variables indépendantes sont localement significatives (au seuil de 5 %)?

Les deux variables les plus significatives localement sont celles relatives au niveau de vie et à la population de moins de 15 ans (toutes deux 124 IRIS). Aussi, 159 modèles locaux ne présentent aucune variable significative au seuil retenu.

8.2 Mise en œuvre et analyse dans R

Il existe deux principaux *packages* pour mettre en œuvre des modèles GWR, soit `spgwr` (Bivand et Yu 2023) et `GWmodel` (Isabella Gollini et al. 2015; Binbin Lu et al. 2014). La construction d'un modèle GWR comprend les étapes suivantes :

1. Sélection de la taille de la zone d'influence (*bandwidth*) optimale.
2. Réalisation de la GWR avec la taille de la zone d'influence optimale.
3. Comparaison des modèles MCO et GWR.
4. Cartographie des résultats du modèle GWR (R^2, coefficients, valeurs de *t*, etc.).

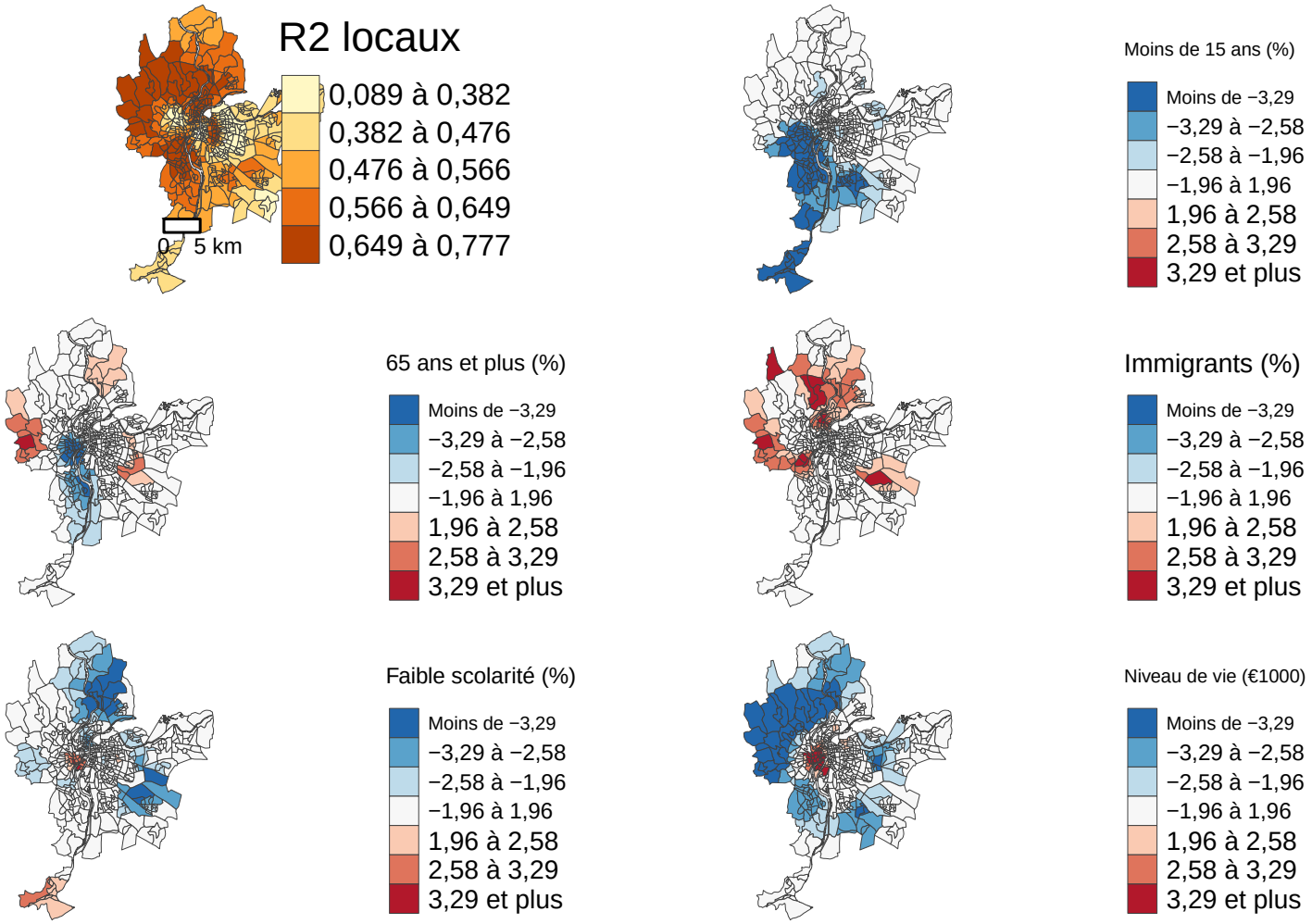


FIGURE 8.2 – Cartographie des valeurs locales de R^2 et de t

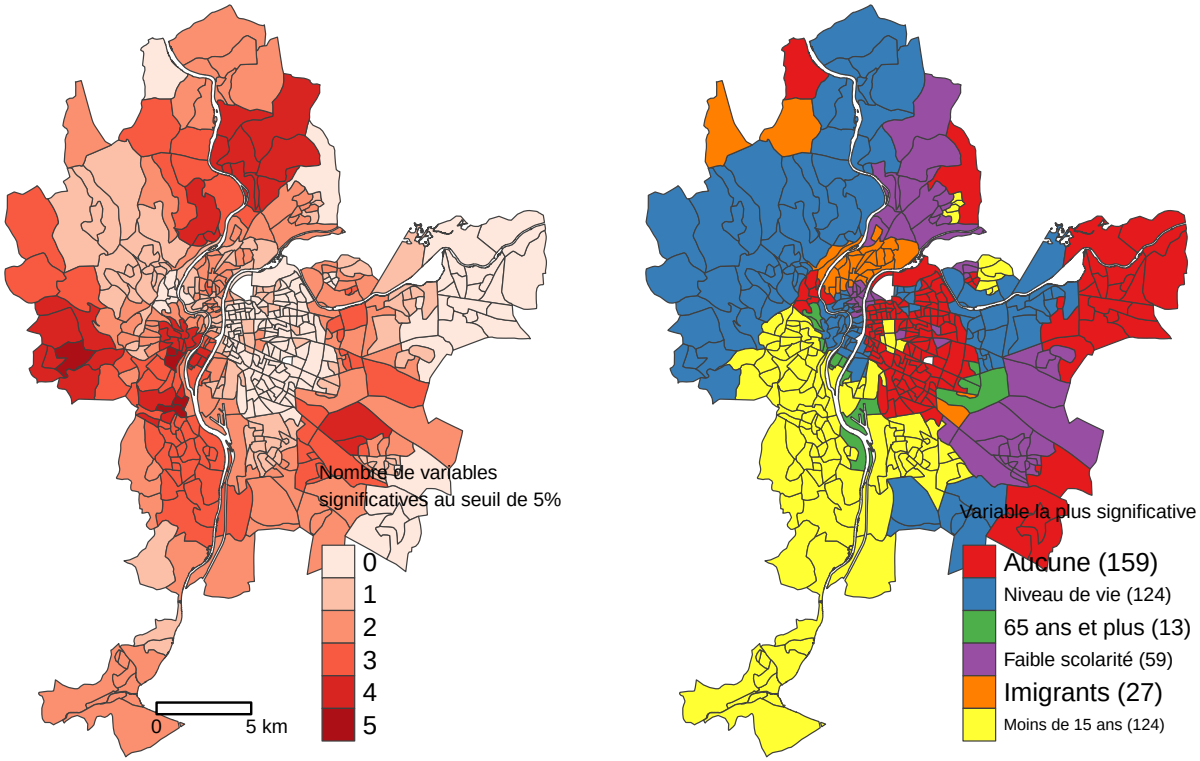


FIGURE 8.3 – Variables indépendantes significatives du modèle GWR

8.2.1 GWR avec le package `spgwr`

8.2.1.1 Définition de la taille de la zone d'influence

La sélection de la taille de la zone d'influence optimale est réalisée avec la fonction `gwr.sel` pour laquelle :

- le paramètre `gweight` permet de spécifier une fonction noyau (*kernel*) gaussienne (`gwr.gauss`) ou bicarrée (`gwr.gauss`).
- le paramètre `adapt` permet de spécifier si vous optimisez le nombre de plus proches voisins (`adapt=TRUE`) ou la distance (`adapt=FALSE`).

```
## Chargement des packages et des données
library(sf)
library(tmap)
library(spgwr)
load("data/Lyon.Rdata")

## Ajout des coordonnées X et Y dans LyonIris
xy <- st_coordinates(st_centroid(LyonIris))
LyonIris$X <- xy[,1]
LyonIris$Y <- xy[,2]

## Optimisation du nombre de voisins avec le CV
```

```
bwCVa_voisins <- gwr.sel(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                         data = LyonIris,
                         method = "cv",          # Méthode cv ou AIC
                         gweight = gwr.bisquare, # gwr.gauss ou gwr.bisquare
                         adapt = TRUE,           # adaptatif
                         verbose = FALSE,
                         RMSE = TRUE,
                         longlat = FALSE,
                         coords=cbind(LyonIris$X,LyonIris$Y))

## Optimisation du nombre de voisins avec l'AIC
bwAICa_voisins <- gwr.sel(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                          data = LyonIris,
                          method = "AIC",          # Méthode cv ou AIC
                          gweight = gwr.bisquare,  # gwr.gauss ou gwr.bisquare
                          adapt = TRUE,            # adaptatif
                          verbose = FALSE,
                          RMSE = TRUE,
                          longlat = FALSE,
                          coords=cbind(LyonIris$X,LyonIris$Y))

## Optimisation de la distance avec le CV
bwCV_dist <- gwr.sel(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                     data = LyonIris,
                     method = "cv",     # méthode cv ou AIC
                     gweight=gwr.Gauss, # gwr.gauss ou gwr.bisquare
                     adapt=FALSE,       # non adaptatif
                     verbose = FALSE,
                     RMSE = TRUE,
                     longlat = FALSE,
                     coords=cbind(LyonIris$X,LyonIris$Y))

## Optimisation de la distance avec l'AIC
bwAIC_dist <- gwr.sel(NO2 ~  Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                      data = LyonIris,
                      method = "AIC",      # méthode cv ou AIC
                      gweight=gwr.Gauss,   # gwr.gauss ou gwr.bisquare
                      adapt=FALSE,         # non adaptatif
                      RMSE = TRUE,
                      verbose = FALSE,
                      longlat = FALSE,
                      coords=cbind(LyonIris$X,LyonIris$Y))

## Affichage des résultats d'optimisation
cat("Sélection de la taille de la zone optimale (bandwidth)",
    "\n avec le nombre de plus proches voisins :",
```

```
    "\n  CV =", round(bwCVa_voisins,4), "nombre de voisins =",
    round(bwCVa_voisins*nrow(LyonIris)),
    "\n  AIC =", round(bwAICa_voisins,4), "nombre de voisins =",
    round(bwAICa_voisins*nrow(LyonIris)),
    "\nSélection de la taille de la zone optimale (bandwidth) avec la distance :",
    "\n  CV =", round(bwCV_dist, 0),  "mètres",
    "\n  AIC =", round(bwAIC_dist, 0), "mètres")
```

```
Sélection de la taille de la zone optimale (bandwidth)
 avec le nombre de plus proches voisins :
  CV = 0.1818 nombre de voisins = 92
  AIC = 0.1067 nombre de voisins = 54
Sélection de la taille de la zone optimale (bandwidth) avec la distance :
  CV = 1315 mètres
  AIC = 1662 mètres
```

Les résultats ci-dessus montrent que le nombre de plus proches voisins pourrait être de 92 selon l'approche *cross-validation* et de 54 selon la méthode basée sur l'AIC. Si la valeur de b est basée sur la distance, elle serait alors optimale à 1315 et à 1662 mètres selon les deux méthodes.

8.2.1.2 Réalisation de la GWR

Avec la fonction **gwr**, nous estimons un modèle GWR avec un noyau (*kernel*) bicarré et un nombre optimisé de plus proches voisins selon la méthode CV, soit 92.

```
# Modèle global : régression linéaire multiple
modele_global <- lm(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed, data = LyonIris)

# Modèle GWR
modele_gwr <- gwr(NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed,
                  data = LyonIris,
                  adapt= bwCVa_voisins,
                  gweight = gwr.bisquare,
                  hatmatrix = TRUE,
                  se.fit = TRUE,
                  coords = cbind(LyonIris$X,LyonIris$Y),
                  longlat = FALSE)
```

Le code ci-dessous permet de renvoyer les statistiques univariées des coefficients des 506 régressions locales, réalisées pour chacune des 506 entités spatiales (IRIS), et les statistiques d'ajustement du modèle GWR (AIC, R^2 quasi-global, etc.).

```
modele_gwr
```

```
Call:
gwr(formula = NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
    NivVieMed, data = LyonIris, coords = cbind(LyonIris$X, LyonIris$Y),
    gweight = gwr.bisquare, adapt = bwCVa_voisins, hatmatrix = TRUE,
    longlat = FALSE, se.fit = TRUE)
Kernel function: gwr.bisquare
Adaptive quantile: 0.1818192 (about 92 of 506 data points)
Summary of GWR coefficient estimates at data points:
                  Min.   1st Qu.    Median   3rd Qu.      Max.  Global
X.Intercept. 12.429518 30.249511 38.619342 48.038863 60.098584 49.4330
Pct0_14      -1.094802 -0.360556 -0.215643 -0.047687  0.382801 -0.5335
Pct_65       -0.715331 -0.158253 -0.031353  0.076086  0.464992 -0.1505
Pct_Img      -0.331892 -0.049146  0.077177  0.240755  0.670433  0.2829
Pct_brevet   -0.655221 -0.221655 -0.084954  0.047835  0.598456 -0.2400
NivVieMed    -1.140895 -0.560649 -0.214717  0.193768  1.311228 -0.3162
Number of data points: 506
Effective number of parameters (residual: 2traceS - traceS'S): 107.1278
Effective degrees of freedom (residual: 2traceS - traceS'S): 398.8722
Sigma (residual: 2traceS - traceS'S): 4.118116
Effective number of parameters (model: traceS): 81.53263
Effective degrees of freedom (model: traceS): 424.4674
Sigma (model: traceS): 3.992025
Sigma (ML): 3.656286
AICc (GWR p. 61, eq 2.33; p. 96, eq. 4.21): 2945.674
AIC (GWR p. 96, eq. 4.22): 2829.504
Residual sum of squares: 6764.424
Quasi-global R2: 0.783008
```

8.2.1.3 Comparaison des modèles MCO et GWR

Le R^2 du modèle GWR est bien supérieur à celui du modèle classique MCO (0,783 contre 0,283).

```
# Calcul des R2 pour les modèles MCO et GWR
r2_global <- summary(modele_global)$r.squared
rss <- sum((modele_gwr$lm$y - modele_gwr$SDF$pred)^2)
tss <- sum((modele_gwr$lm$y - mean(modele_gwr$SDF$pred))^2)
r2_gwrquasiglobal <- 1 - (rss / tss)
cat("R2 global (LM) = ", round(r2_global, 3),
    "\nR2 quasi-global (GWR) : ", round(r2_gwrquasiglobal, 3))
```

```
R2 global (LM) =  0.283
R2 quasi-global (GWR) :  0.784
```

Fotheringham *et al.* (2003) proposent plusieurs tests pour comparer les modèles GWR et classique qui sont implémentés dans le *package* `spgwr` : fonctions `BFC99.gwr.test(modele_gwr)`, `BFC02.gwr.test(modele_gwr)`,

`LMZ.F1GWR.test(modele_gwr)`, `LMZ.F2GWR.test(modele_gwr)`. Si les valeurs de p de ces tests sont inférieures à 0,05, alors le modèle GWR améliore de façon significative la capacité prédictive du modèle de régression globale, ce que confirment les résultats ci-dessous.

```
anova(modele_gwr)
```

```
Analysis of Variance Table
                    Df  Sum Sq Mean Sq F value
OLS Residuals     6.00 22346.0
GWR Improvement 101.13 15581.6 154.078
GWR Residuals   398.87  6764.4  16.959  9.0854
```

```
BFC99.gwr.test(modele_gwr)
```

```
    Brunsdon, Fotheringham & Charlton (1999) ANOVA

data:  modele_gwr
F = 9.0854, df1 = 350.93, df2 = 430.81, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
SS GWR improvement   SS GWR residuals
         15581.596           6764.424
```

```
BFC02.gwr.test(modele_gwr)
```

```
    Brunsdon, Fotheringham & Charlton (2002, pp. 91-2) ANOVA

data:  modele_gwr
F = 3.3035, df1 = 500.00, df2 = 398.87, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
SS OLS residuals SS GWR residuals
       22346.021         6764.424
```

```
LMZ.F1GWR.test(modele_gwr)
```

```
    Leung et al. (2000) F(1) test

data:  modele_gwr
F = 0.37946, df1 = 430.81, df2 = 500.00, p-value < 2.2e-16
alternative hypothesis: less
```

```
sample estimates:
SS OLS residuals SS GWR residuals
       22346.021         6764.424
```

```
LMZ.F2GWR.test(modele_gwr)
```

```
    Leung et al. (2000) F(2) test

data:  modele_gwr
F = 3.4476, df1 = 142.92, df2 = 500.00, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
  SS OLS residuals SS GWR improvement
          22346.02           15581.60
```

Un autre test (`LMZ.F3GWR.test`) permet de répondre à la question suivante : est-ce que les coefficients de régression du modèle GWR varient spatialement de façon significative? Les résultats ci-dessous démontrent que c'est le cas pour toutes les variables indépendantes et la constante ($p < 0{,}001$).

```
LMZ.F3GWR.test(modele_gwr)
```

```
Leung et al. (2000) F(3) test

            F statistic Numerator d.f. Denominator d.f.     Pr(>)
(Intercept)      2.2771       134.3880           430.81 1.629e-10 ***
Pct0_14          2.7767       141.7244           430.81 5.636e-16 ***
Pct_65           2.0918       169.0472           430.81 8.399e-10 ***
Pct_Img          1.9486       106.4400           430.81 1.550e-06 ***
Pct_brevet       2.4445       121.2830           430.81 1.629e-11 ***
NivVieMed        3.6926       138.8118           430.81 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pour comparer **la dépendance spatiale** des deux modèles (MCO versus GWR), nous calculons le I de Moran sur leurs résidus avec une matrice de contiguïté standardisée selon le partage d'un segment (*Rook*). Avec une valeur du I de Moran de 0,587 ($p < 0{,}001$), les résidus du modèle global (MCO) sont fortement autocorrélés spatialement, traduisant ainsi un problème de dépendance spatiale du modèle. Bien qu'elle soit toujours significative, l'autocorrélation spatiale des résidus du modèle GWR est bien plus faible (0,288, $p < 0{,}001$). La cartographie des résidus des deux modèles à la figure 8.4 corrobore ces résultats.

```
library(spdep)
## Résidus des modèles global (MCO) et GWR
LyonIris$MCO.Residus <- modele_global$residuals
LyonIris$GWR.Residus <- modele_gwr$SDF$pred - modele_gwr$lm$y

## Matrice de contiguïté
rook_nb <- poly2nb(LyonIris, queen = FALSE)
rook_w <- nb2listw(rook_nb, zero.policy = TRUE, style = "W")

# I de Moran sur les résidus du modèle global (MCO)
moran.test(LyonIris$MCO.Residus, rook_w, randomisation = FALSE)
```

```
    Moran I test under normality

data:  LyonIris$MCO.Residus
weights: rook_w

Moran I statistic standard deviate = 20.986, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5873120615     -0.0019801980      0.0007885196
```

```
# I de Moran sur les résidus du modèle GWR (MCO)
moran.test(LyonIris$GWR.Residus, rook_w, randomisation = FALSE)
```

```
    Moran I test under normality

data:  LyonIris$GWR.Residus
weights: rook_w

Moran I statistic standard deviate = 10.323, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.2878952422     -0.0019801980      0.0007885196
```

```
## Cartographie des résidus du modèle MCO
tmap_mode("plot")
carte1 <- tm_shape(LyonIris)+
            tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="MCO.Residus", n = 6, style = "pretty",
                  legend.format = list(text.separator = "à",
```

```
                                    decimal.mark = ","),
                midpoint = 0,
                palette = "-RdBu",
                title = "MCO") +
            tm_layout(frame=FALSE)

## Cartographie des résidus du modèle GWR
carte2 <- tm_shape(LyonIris)+
            tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.Residus", n = 6, style = "pretty",
                legend.format = list(text.separator = "à",
                                    decimal.mark = ","),
                midpoint = 0,
                palette = "-RdBu",
                title = "GWR") +
            tm_layout(frame=FALSE) +
            tm_scale_bar(breaks = c(0,5))

tmap_arrange(carte1, carte2, nrow = 1)
```

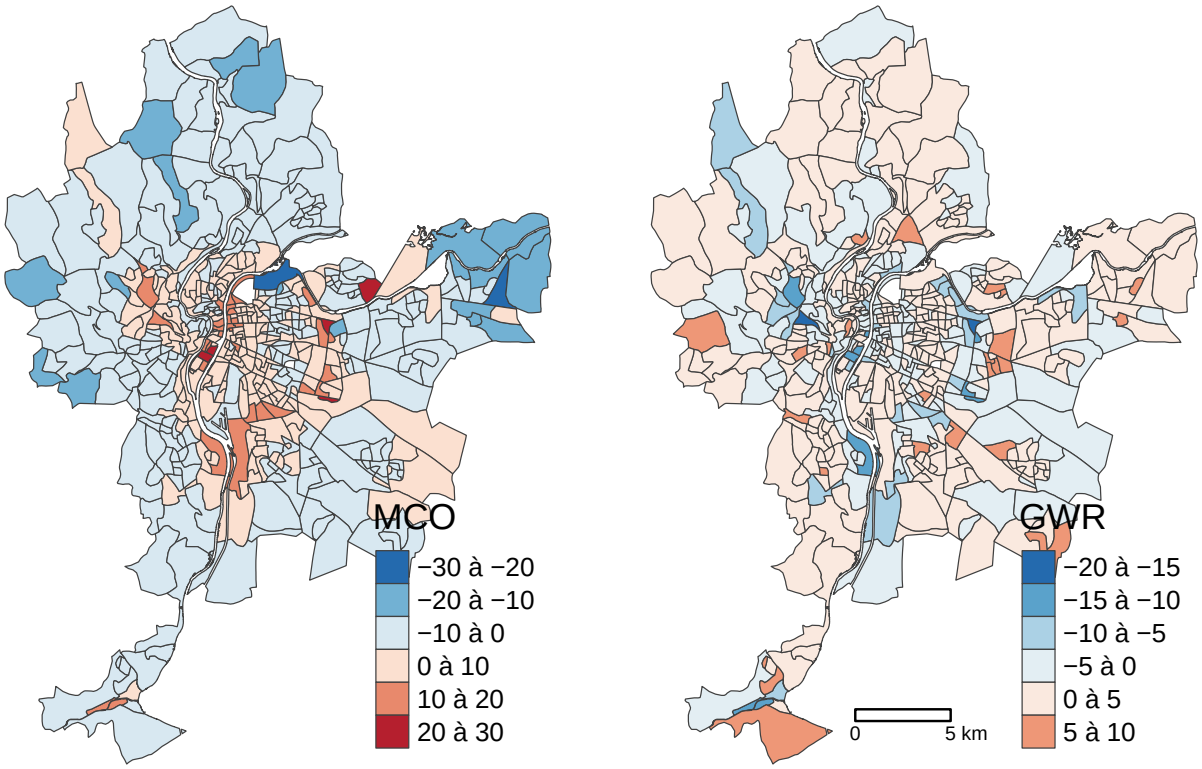


FIGURE 8.4 – Cartographie des résidus des modèles MCO et GWR

8.2.1.4 Cartographie des résultats du modèle GWR

Dans un premier temps, nous ajoutons les valeurs locales des R^2, des coefficients de régression et des valeurs de t dans la couche `sf`. Notez que les résultats locaux de la GWR sont stockés dans l'objet `modele_gwr$SDF`.

```
## Récupération du R carré local
LyonIris$GWR.R2 <- modele_gwr$SDF$localR2

## Récupération des coefficients de régression et calcul des valeurs de t locales
names(modele_gwr$SDF)
```

```
 [1] "sum.w"              "(Intercept)"        "Pct0_14"
 [4] "Pct_65"             "Pct_Img"            "Pct_brevet"
 [7] "NivVieMed"          "(Intercept)_se"     "Pct0_14_se"
[10] "Pct_65_se"          "Pct_Img_se"         "Pct_brevet_se"
[13] "NivVieMed_se"       "gwr.e"              "pred"
[16] "pred.se"            "localR2"            "(Intercept)_se_EDF"
[19] "Pct0_14_se_EDF"     "Pct_65_se_EDF"      "Pct_Img_se_EDF"
[22] "Pct_brevet_se_EDF"  "NivVieMed_se_EDF"   "pred.se"
```

```
vars_indep <- c("Pct0_14", "Pct_65", "Pct_Img", "Pct_brevet", "NivVieMed")

for(e in vars_indep){
  # Nom des nouvelles variables
  var_coef <- paste0("GWR.", "B_", e)
  var_t    <- paste0("GWR.", "T_", e)
  # Récupération des coefficients pour les variables indépendantes
  LyonIris[[var_coef]] <- modele_gwr$SDF[[e]]
  # Calcul des valeurs de t pour les variables indépendantes
  LyonIris[[var_t]]    <- modele_gwr$SDF[[e]] / modele_gwr$SDF[[paste0(e, "_se")]]
}
```

Cartographie et analyse des R^2 locaux

Les deux lignes de code ci-dessous permettent d'obtenir un résumé sommaire des statistiques descriptives (minimum, premier et troisième quartiles, moyenne et maximum) des valeurs locales de R^2, mais aussi de constater qu'elles varient de 0,19 à 0,74 pour 95 % des 506 observations (IRIS).

```
summary(LyonIris$GWR.R2)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0888  0.4205  0.5281  0.5081  0.6322  0.7774
```

```
round(quantile(LyonIris$GWR.R2, probs = c(0.025, 0.975)), 4)
```

```
  2.5%  97.5% 
0.1876 0.7390 
```

Le code ci-dessous permet ensuite de cartographier les R^2 locaux de la GWR (figure 8.5).

```
library(tmap)
# Paramètres pour la légende
legende_parametres <- list(text.separator = "à",
                           decimal.mark = ",")

tm_shape(LyonIris)+
  tm_borders(col="gray25", lwd=.5)+
  tm_fill(col="GWR.R2",
          palette="YlOrBr",
          n=5, style="quantile",
          legend.format = legende_parametres,
          title = "R2 locaux")+
tm_layout(frame=FALSE)+
tm_scale_bar(breaks=c(0,5))
```

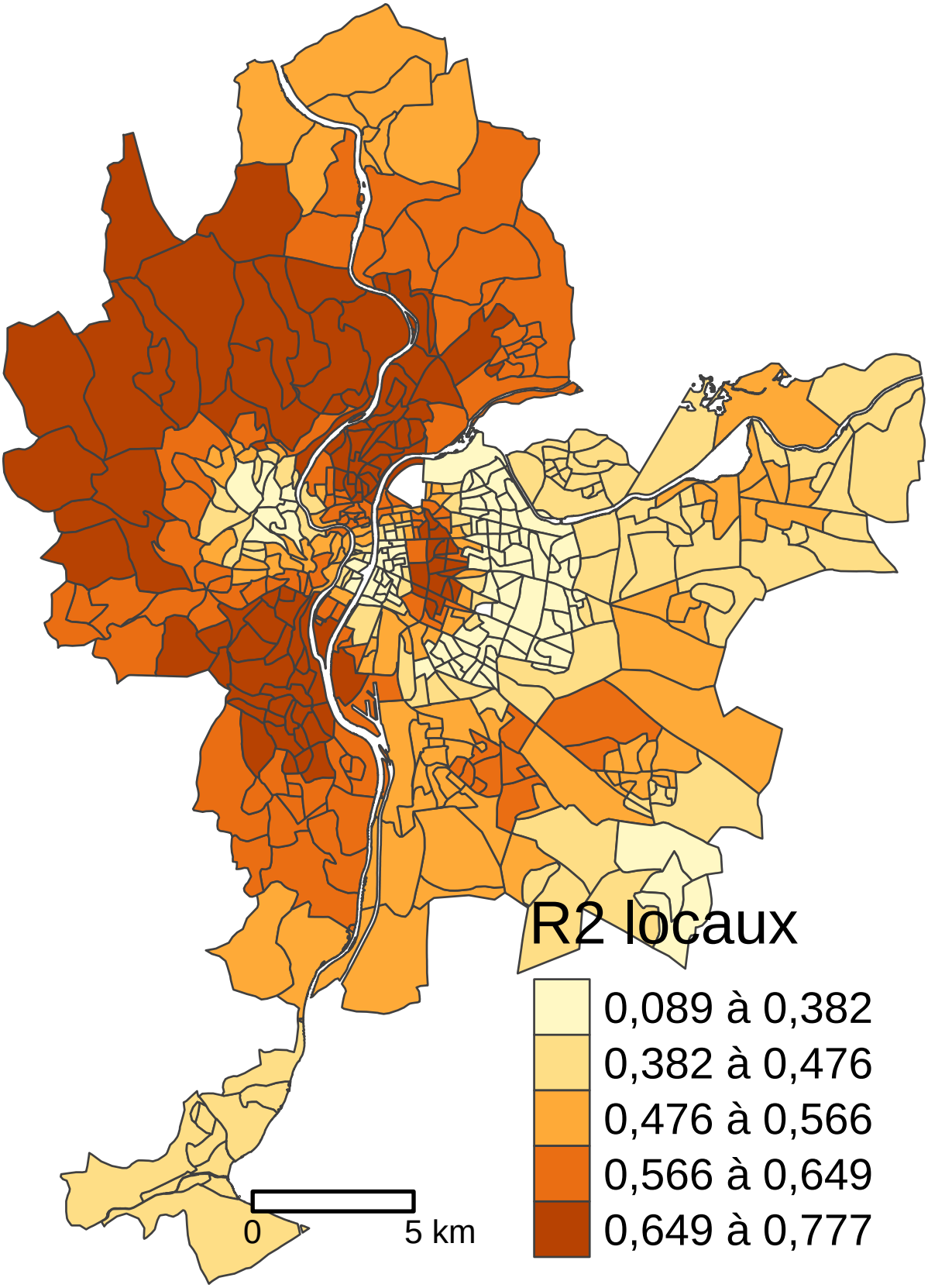


Figure 8.5 – Cartographie des R carrés locaux de la GWR

Cartographie des coefficients de régression locaux

Le code ci-dessous permet ensuite de cartographier les coefficients locaux de la GWR (figure 8.6).

```
# Paramètres pour la légende
legende_parametres <- list(text.separator = "à",
                           decimal.mark = ",",
                           big.mark = " ")

carte1 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.B_Pct0_14", palette="YlOrBr", n=4, style="pretty",
                      legend.format = legende_parametres,
                    title = "Moins de 15 ans (%)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)

carte2 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.B_Pct_65", palette="YlOrBr", n=4, style="pretty",
                      legend.format = legende_parametres,
                    title = "65 ans et plus (%)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)

carte3 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.B_Pct_Img", palette="YlOrBr", n=4, style="pretty",
                      legend.format = legende_parametres,
                    title = "Immigrants (%)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)

carte4 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.B_Pct_brevet", palette="YlOrBr", n=4, style="pretty",
                      legend.format = legende_parametres,
                    title = "Faible scolarité (%)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)

carte5 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5,)+
            tm_fill(col="GWR.B_NivVieMed", palette="YlOrBr", n=4, style="pretty",
                      legend.format = legende_parametres,
                    title = "Niveau de vie (€1000)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)+
          tm_scale_bar(breaks=c(0,5))

tmap_arrange(carte1, carte2, carte3, carte4, carte5, ncol = 2, nrow=3)
```

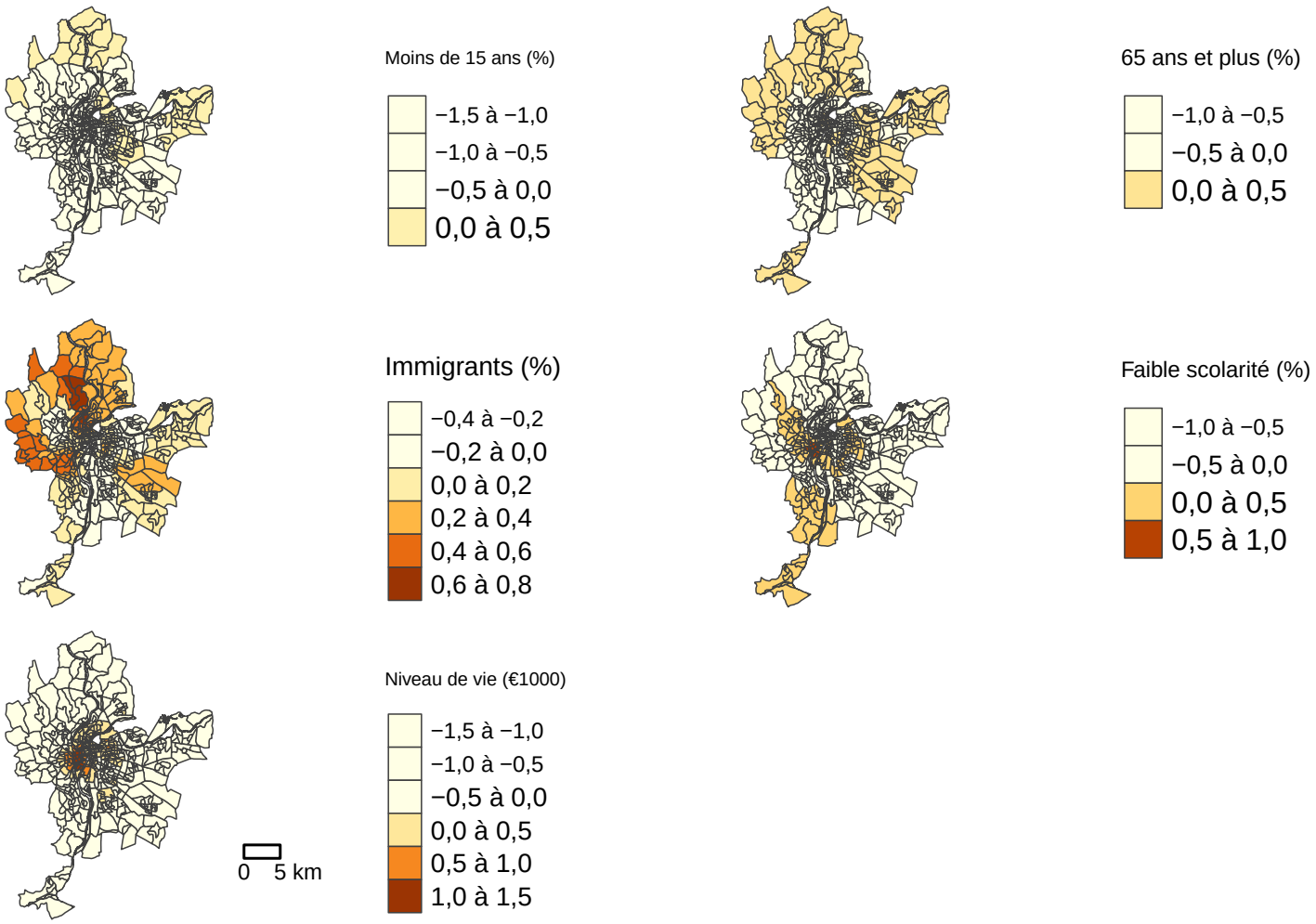


FIGURE 8.6 – Cartographie des coefficients de régression locaux de la GWR

Cartographie des valeurs de t locales

Pour cartographier les valeurs de t, nous utilisons les seuils de ± 1,96, 2,58 et 3,29, indiquant des seuils de signification à 5 %, 1 % et 0,1 % (figure 8.7).

```
# Bornes pour les valeurs de t
classes_intervalles = c(-Inf, -3.29, -2.58, -1.96, 1.96, 2.58, 3.29, Inf)

# Paramètres pour la légende
legende_parametres <- list(text.separator = "à",
                           text.less.than = "Moins de",
                           text.or.more =  "et plus",
                           decimal.mark = ",",
                           big.mark = " ")

# Construction des cartes
```

```
carte1 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.T_Pct0_14", palette="-RdBu",
                  midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende_parametres,
                  title = "Moins de 15 ans (%)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)

carte2 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.T_Pct_65", palette="-RdBu",
                  midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende_parametres,
                  title = "65 ans et plus (%)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)

carte3 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.T_Pct_Img", palette="-RdBu",
                  midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende_parametres,
                  title = "Immigrants (%)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)

carte4 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.T_Pct_brevet", palette="-RdBu",
                  midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende_parametres,
                  title = "Faible scolarité (%)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)

carte5 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="GWR.T_NivVieMed", palette="-RdBu",
                 midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende_parametres,
                  title = "Niveau de vie (€1000)")+
          tm_layout(frame=FALSE, legend.outside = TRUE)+
          tm_scale_bar(breaks=c(0,5))

# Combinaison des cartes dans la même figure
tmap_arrange(carte1, carte2, carte3, carte4, carte5, ncol = 2, nrow=3)
```

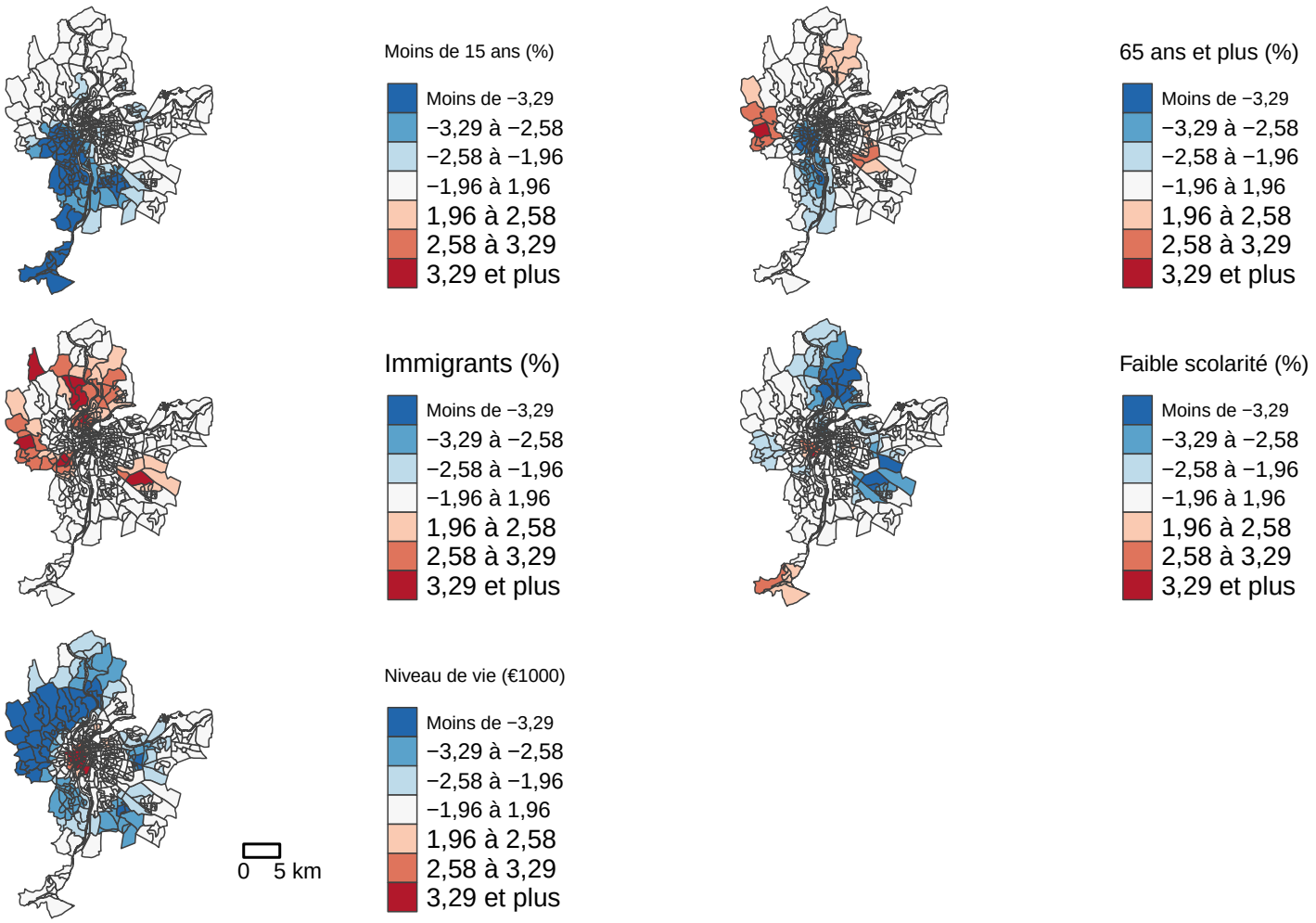


FIGURE 8.7 – Cartographie des valeurs de t locales de la GWR

Cartographie du nombre de variables significatives

Nous pouvons aussi cartographier le nombre de variables localement significatives aux seuils de 5 % et de 1 %.

```
## Identifier la variable la plus significative avec les valeurs de t
Vars_t <- paste0("GWR.T_", c("Pct0_14", "Pct_65", "Pct_Img", "Pct_brevet", "NivVieMed"))
lyon_df  <- st_drop_geometry(LyonIris)
lyon_df <- abs(lyon_df[,Vars_t])
plus_sign <- Vars_t[apply(lyon_df[Vars_t],1,which.max)]
plus_sign <- substr(plus_sign, 7, nchar(plus_sign))
max_abs_tvalue <- apply(lyon_df[Vars_t], 1, max)
plus_sign <- ifelse(max_abs_tvalue<1.96, "Aucune", plus_sign)

## Nombre de variables significatives au seuil de 5%, soit abs(t)= 1,96)
LyonIris$nb_signif_1.96 <- as.factor(rowSums(lyon_df > 1.96))
LyonIris$nb_signif_2.58 <- as.factor(rowSums(lyon_df > 2.58))
```

```
LyonIris$plus_sign      <- as.factor(plus_sign)

## Cartographie
carte1 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
          tm_fill(col="nb_signif_1.96", palette="Reds",
                  title = "Sign. au seuil de 5%")+
          tm_layout(frame=FALSE)+ tm_scale_bar(breaks=c(0,5))
carte2 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
          tm_fill(col="nb_signif_2.58", palette="Reds",
                  title = "Sign. au seuil de 1%")+
          tm_layout(frame=FALSE)
tmap_arrange(carte1, carte2, ncol=2, nrow=1)
```

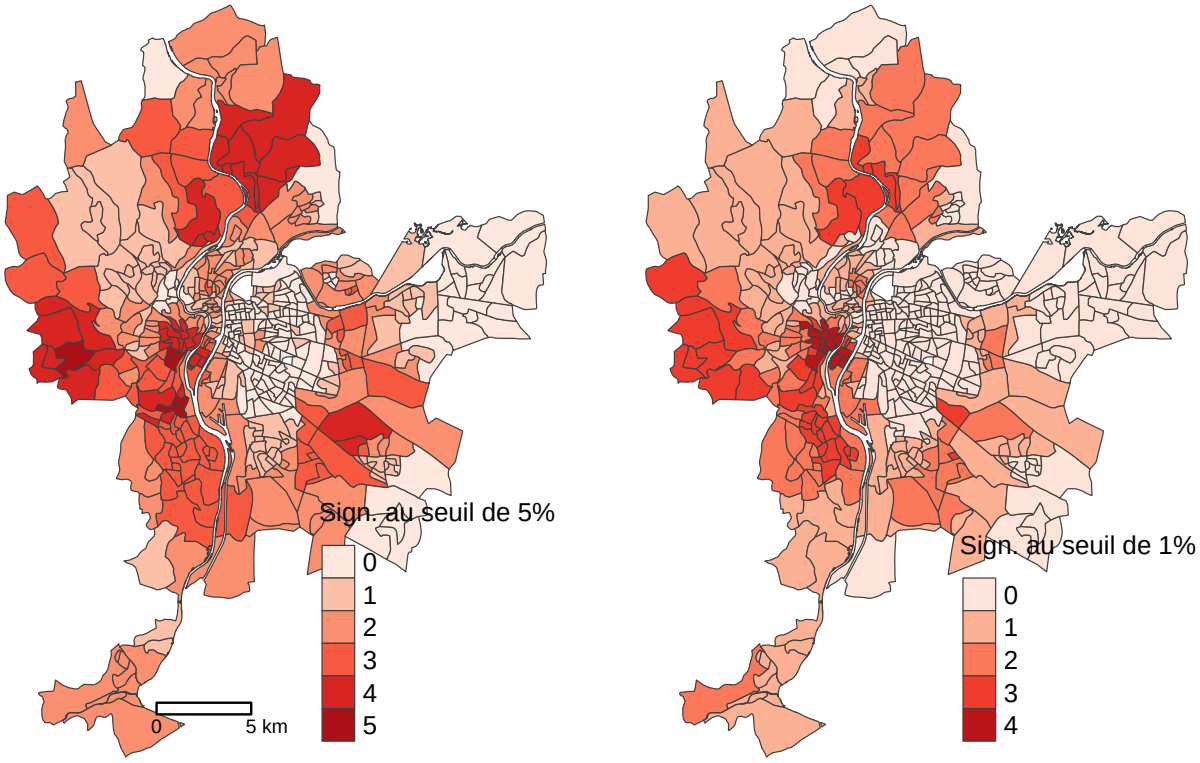


FIGURE 8.8 – Nombre de variables significatives aux seuils de 5% et 1%

Cartographie de la variable la plus significative avec la valeur de t

Finalement, le code ci-dessous permet de repérer la variable la plus significative au seuil de 5 %, c'est-à-dire avec la plus forte valeur absolue pour la valeur de *t*.

```
tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
  tm_fill(col="plus_sign", palette="Set1",
          title = "Variable la plus significative")+
tm_layout(frame=FALSE)+ tm_scale_bar(breaks=c(0,5))
```

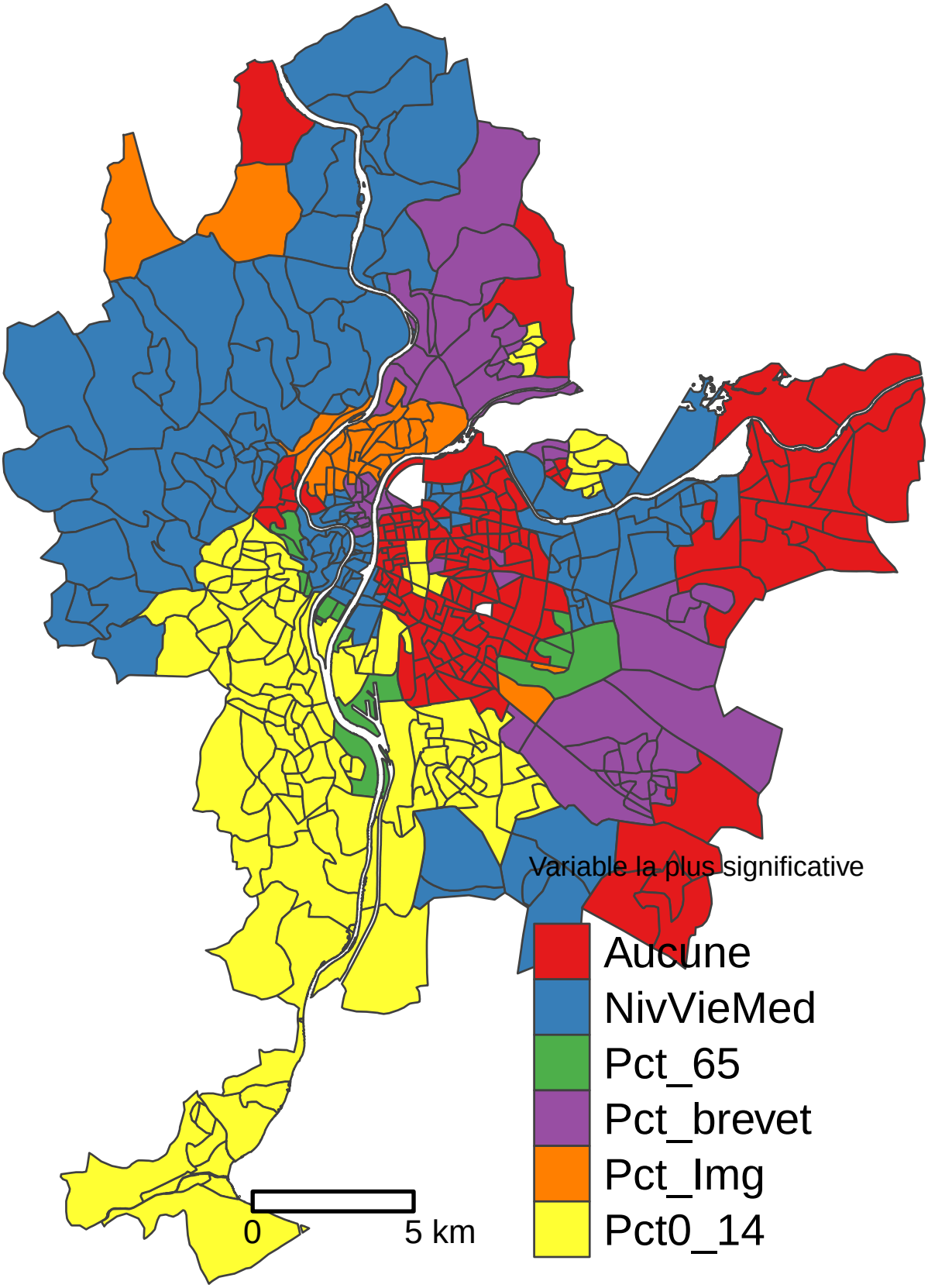


FIGURE 8.9 – Variable indépendante la plus significative au seuil de 5 %

8.2.2 GWR avec le package `GWmodel`

La construction d'une GWR est certainement plus simple (moins verbeuse) avec les fonctions `bw.gwr` et `gwr.basic` du *package* `GWmodel`.

8.2.2.1 Définition de la taille de la zone d'influence

La fonction `bw.gwr` permet de trouver la valeur de la zone d'influence (b).

```
library(GWmodel)
load("data/Lyon.Rdata")
# Construction de la matrice de distance
xy <- st_coordinates(st_centroid(LyonIris))
matrice_distances <- gw.dist(dp.locat=xy, p = 2, longlat = FALSE)

## Optimisation du nombre de voisins avec le CV
bwCV_voisins <- bw.gwr(NO2 ~  Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed,
                       data = LyonIris,
                       dMat = matrice_distances,
                       approach="CV",
                       kernel="bisquare",
                       adaptive=TRUE,
                       longlat = FALSE)
```

```
Adaptive bandwidth: 320 CV score: 17072.1
Adaptive bandwidth: 206 CV score: 14142.57
Adaptive bandwidth: 134 CV score: 12329.97
Adaptive bandwidth: 91 CV score: 11919.32
Adaptive bandwidth: 63 CV score: 12156.26
Adaptive bandwidth: 107 CV score: 11981.63
Adaptive bandwidth: 79 CV score: 12001.41
Adaptive bandwidth: 96 CV score: 11900.81
Adaptive bandwidth: 101 CV score: 11912.07
Adaptive bandwidth: 94 CV score: 11919.6
Adaptive bandwidth: 98 CV score: 11911.68
Adaptive bandwidth: 95 CV score: 11911.87
Adaptive bandwidth: 97 CV score: 11899.93
Adaptive bandwidth: 97 CV score: 11899.93
```

```
## Optimisation du nombre de voisins avec l'AIC
bwAIC_voisins <- bw.gwr(NO2 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed,
                        data = LyonIris,
                        dMat = matrice_distances,
                        approach="AIC",
                        kernel="bisquare",
```

```
                        adaptive=TRUE,
                        longlat = FALSE)
```

```
Adaptive bandwidth (number of nearest neighbours): 320 AICc value: 3181.763
Adaptive bandwidth (number of nearest neighbours): 206 AICc value: 3072.458
Adaptive bandwidth (number of nearest neighbours): 134 AICc value: 2989.605
Adaptive bandwidth (number of nearest neighbours): 91 AICc value: 2943.187
Adaptive bandwidth (number of nearest neighbours): 63 AICc value: 2925.002
Adaptive bandwidth (number of nearest neighbours): 47 AICc value: 2927.612
Adaptive bandwidth (number of nearest neighbours): 74 AICc value: 2929.903
Adaptive bandwidth (number of nearest neighbours): 57 AICc value: 2922.621
Adaptive bandwidth (number of nearest neighbours): 52 AICc value: 2922.879
Adaptive bandwidth (number of nearest neighbours): 58 AICc value: 2923.89
Adaptive bandwidth (number of nearest neighbours): 54 AICc value: 2922.612
Adaptive bandwidth (number of nearest neighbours): 54 AICc value: 2922.612
```

```
## Optimisation de la distance avec le CV
bwCV_dist <- bw.gwr(NO2 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed,
                    data = LyonIris,
                    dMat = matrice_distances,
                    approach="CV",
                    kernel="gaussian",
                    adaptive=FALSE,
                    longlat = FALSE)
```

```
Fixed bandwidth: 22632.98 CV score: 23240.62
Fixed bandwidth: 13990.75 CV score: 22942.89
Fixed bandwidth: 8649.556 CV score: 21987.02
Fixed bandwidth: 5348.517 CV score: 19330.61
Fixed bandwidth: 3308.362 CV score: 15656.55
Fixed bandwidth: 2047.478 CV score: 13534
Fixed bandwidth: 1268.208 CV score: 12883.62
Fixed bandwidth: 786.5929 CV score: 69868.32
Fixed bandwidth: 1565.863 CV score: 13058.8
Fixed bandwidth: 1084.247 CV score: 13320.38
Fixed bandwidth: 1381.902 CV score: 12891.23
Fixed bandwidth: 1197.941 CV score: 12968.27
Fixed bandwidth: 1311.635 CV score: 12869.97
Fixed bandwidth: 1338.475 CV score: 12872.81
Fixed bandwidth: 1295.047 CV score: 12872.27
Fixed bandwidth: 1321.887 CV score: 12870.16
Fixed bandwidth: 1305.299 CV score: 12870.46
Fixed bandwidth: 1315.551 CV score: 12869.91
Fixed bandwidth: 1317.971 CV score: 12869.95
Fixed bandwidth: 1314.055 CV score: 12869.91
```

```
Fixed bandwidth: 1316.475 CV score: 12869.92
Fixed bandwidth: 1314.98 CV score: 12869.9
Fixed bandwidth: 1314.627 CV score: 12869.91
Fixed bandwidth: 1315.198 CV score: 12869.9
```

```
## Optimisation de la distance avec l'AIC
bwAIC_dist <- bw.gwr(NO2 ~  Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed,
                     data = LyonIris,
                     dMat = matrice_distances,
                     approach="AIC",
                     kernel="gaussian",
                     adaptive=FALSE,
                     longlat = FALSE)
```

```
Fixed bandwidth: 22632.98 AICc value: 3362.597
Fixed bandwidth: 13990.75 AICc value: 3354.75
Fixed bandwidth: 8649.556 AICc value: 3328.284
Fixed bandwidth: 5348.517 AICc value: 3241.508
Fixed bandwidth: 3308.362 AICc value: 3114.667
Fixed bandwidth: 2047.478 AICc value: 3043.093
Fixed bandwidth: 1268.208 AICc value: 3050.841
Fixed bandwidth: 2529.093 AICc value: 3065.054
Fixed bandwidth: 1749.823 AICc value: 3036.6
Fixed bandwidth: 1565.863 AICc value: 3036.749
Fixed bandwidth: 1863.517 AICc value: 3038.25
Fixed bandwidth: 1679.556 AICc value: 3036.195
Fixed bandwidth: 1636.129 AICc value: 3036.214
Fixed bandwidth: 1706.396 AICc value: 3036.289
Fixed bandwidth: 1662.969 AICc value: 3036.176
Fixed bandwidth: 1652.717 AICc value: 3036.181
Fixed bandwidth: 1669.305 AICc value: 3036.18
Fixed bandwidth: 1659.053 AICc value: 3036.177
Fixed bandwidth: 1665.389 AICc value: 3036.177
Fixed bandwidth: 1661.473 AICc value: 3036.176
Fixed bandwidth: 1660.549 AICc value: 3036.176
```

```
## Affichage des résultats d'optimisation
cat("Sélection de la taille de la zone optimale (bandwidth)",
    "\n avec le nombre de plus proches voisins :",
    "\n  CV =", round(bwCV_voisins, 4), "nombre de voisins =",  round(bwCV_voisins*nrow(LyonIris)),
    "\n  AIC =", round(bwAIC_voisins, 4), "nombre de voisins =", round(bwAIC_voisins*nrow(LyonIris)),
    "\nSélection de la taille de la zone optimale (bandwidth) avec la distance :",
    "\n  CV =", round(bwCV_dist, 0),  "mètres",
    "\n  AIC =", round(bwAIC_dist, 0), "mètres")
```

```
Sélection de la taille de la zone optimale (bandwidth)
```

```
 avec le nombre de plus proches voisins :
  CV = 97 nombre de voisins = 49082
  AIC = 54 nombre de voisins = 27324
Sélection de la taille de la zone optimale (bandwidth) avec la distance :
  CV = 1315 mètres
  AIC = 1661 mètres
```

8.2.2.2 Réalisation de la GWR

Comparativement à la fonction `gwr` du *package* `spgwr`, `gwr.basic` du *package* `GWmodel` offre l'avantage de fournir l'ensemble des résultats :

- Ceux de la régression linéaire multiple (modèle global).
- Un sommaire statistique des coefficients de régression de la GWR.
- Les statistiques de la qualité d'ajustement du modèle (AIC, AICc, R^2, R^2 ajusté).
- Les tests pour comparer les modèles global et GWR (tests F1, F2).
- Le test F3 pour vérifier si les coefficients de régression varient ou non spatialement de façon significative.

```
# Calcul de la GWR
modele_gwr <- gwr.basic(NO2 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed,
                        data = LyonIris,
                        dMat = matrice_distances,  # Matrice de distances
                        bw = bwCV_voisins,       # Zone d'influence
                        kernel = "bisquare",     # Gaussien ou bicarrée
                        adaptive=TRUE,           # Nombre de voisins (TRUE) ou distance (FALSE)
                        F123.test = TRUE,        # Tests F123
                        longlat = FALSE)
modele_gwr
```

```
   ***********************************************************************
   *                       Package   GWmodel                             *
   ***********************************************************************
   Program starts at: 2025-03-18 19:51:35.373361
   Call:
   gwr.basic(formula = NO2 ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet +
    NivVieMed, data = LyonIris, bw = bwCV_voisins, kernel = "bisquare",
    adaptive = TRUE, longlat = FALSE, dMat = matrice_distances,
    F123.test = TRUE)

   Dependent (y) variable:  NO2
   Independent variables:  Pct0_14 Pct_65 Pct_Img Pct_brevet NivVieMed
   Number of data points: 506
   ***********************************************************************
   *                    Results of Global Regression                     *
   ***********************************************************************
```

```
   Call:
    lm(formula = formula, data = data)

   Residuals:
    Min      1Q  Median      3Q     Max
-27.733  -4.457  -0.499   3.507  29.160

   Coefficients:
               Estimate Std. Error t value Pr(>|t|)
   (Intercept) 49.43296    2.99550  16.502  < 2e-16 ***
   Pct0_14     -0.53352    0.06305  -8.461 2.94e-16 ***
   Pct_65      -0.15047    0.05627  -2.674  0.00774 **
   Pct_Img      0.28287    0.05113   5.532 5.12e-08 ***
   Pct_brevet  -0.24004    0.03721  -6.451 2.63e-10 ***
   NivVieMed   -0.31625    0.10180  -3.107  0.00200 **

   ---Significance stars
   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
   Residual standard error: 6.685 on 500 degrees of freedom
   Multiple R-squared: 0.2832
   Adjusted R-squared: 0.276
   F-statistic:  39.5 on 5 and 500 DF,  p-value: < 2.2e-16
   ***Extra Diagnostic information
   Residual sum of squares: 22346.02
   Sigma(hat): 6.658629
   AIC:  3366.626
   AICc:  3366.851
   BIC:  2933.798
   ***********************************************************************
   *          Results of Geographically Weighted Regression              *
   ***********************************************************************

   ********************Model calibration information*********************
   Kernel function: bisquare
   Adaptive bandwidth: 97 (number of nearest neighbours)
   Regression points: the same locations as observations are used.
   Distance metric: A distance matrix is specified for this model calibration.

   ****************Summary of GWR coefficient estimates:******************
                  Min.   1st Qu.    Median   3rd Qu.    Max.
   Intercept  13.249694 30.566393 38.903053 48.035033 59.6124
   Pct0_14    -1.094930 -0.371546 -0.224906 -0.053956  0.3386
   Pct_65     -0.711775 -0.157939 -0.033630  0.073991  0.4514
   Pct_Img    -0.319353 -0.042105  0.078841  0.243274  0.6657
   Pct_brevet -0.624419 -0.224105 -0.092008  0.041021  0.5844
   NivVieMed  -1.108017 -0.555822 -0.213561  0.202347  1.2844
   ************************Diagnostic information*************************
```

```
  Number of data points: 506
  Effective number of parameters (2trace(S) - trace(S'S)): 103.4065
  Effective degrees of freedom (n-2trace(S) + trace(S'S)): 402.5935
  AICc (GWR book, Fotheringham, et al. 2002, p. 61, eq 2.33): 2949.788
  AIC (GWR book, Fotheringham, et al. 2002,GWR p. 96, eq. 4.22): 2839.039
  BIC (GWR book, Fotheringham, et al. 2002,GWR p. 61, eq. 2.34): 2743.804
  Residual sum of squares: 6933.276
  R-square value:  0.7775915
  Adjusted R-square value:  0.7203234
  ******************F test results of GWR calibration*******************
  ---F1 test (Leung et al. 2000)
   F1 statistic Numerator DF Denominator DF     Pr(>)
        0.38534          Inf            500 < 2.2e-16 ***
  ---F2 test (Leung et al. 2000)
   F2 statistic Numerator DF Denominator DF Pr(>)
         3.5405     -31.0892            500   NaN
  ---F3 test (Leung et al. 2000)
              F3 statistic Numerator DF Denominator DF     Pr(>)
  Intercept        2.3480     133.2095            Inf < 2.2e-16 ***
  Pct0_14          2.8529     141.2525            Inf < 2.2e-16 ***
  Pct_65           2.1479     168.6132            Inf 3.399e-16 ***
  Pct_Img          1.9938     105.6136            Inf 5.756e-09 ***
  Pct_brevet       2.4930     120.8152            Inf < 2.2e-16 ***
  NivVieMed        3.8747     138.4279            Inf < 2.2e-16 ***
  ---F4 test (GWR book p92)
   F4 statistic Numerator DF Denominator DF     Pr(>)
        0.31027    402.59351            500 < 2.2e-16 ***

  ---Significance stars
  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
  ***********************************************************************
  Program stops at: 2025-03-18 19:51:49.137952
```

8.2.2.3 Cartographie des résultats du modèle GWR

Une fois le modèle obtenu, nous disposons de l'ensemble des résultats locaux (coefficients, valeur prédite, résidus, valeurs de t et de R^2) dans l'objet `modele_gwr$SDF` qui sera une couche `sf` comme le jeu de données initial. Il est alors aisé de reproduire les cartes présentées à la section 8.2.1.4. En guise d'exemple, dans le code ci-dessous, nous cartographions les valeurs locales du R^2 et uniquement celles de *t* pour la variable *médiane du niveau de vie* (figure 8.10).

```
gwr_resultats <- modele_gwr$SDF
names(gwr_resultats)
```

```
 [1] "Intercept"     "Pct0_14"       "Pct_65"        "Pct_Img"
 [5] "Pct_brevet"    "NivVieMed"     "y"             "yhat"
```

```
 [9] "residual"      "CV_Score"      "Stud_residual" "Intercept_SE"
[13] "Pct0_14_SE"    "Pct_65_SE"     "Pct_Img_SE"    "Pct_brevet_SE"
[17] "NivVieMed_SE"  "Intercept_TV"  "Pct0_14_TV"    "Pct_65_TV"
[21] "Pct_Img_TV"    "Pct_brevet_TV" "NivVieMed_TV"  "Local_R2"
[25] "geometry"
```

```
legende_parametres <- list(text.separator = "à",
                           text.less.than = "Moins de",
                           text.or.more =  "et plus",
                           decimal.mark = ",",
                           big.mark = " ")

classes_intervalles = c(-Inf, -3.29, -2.58, -1.96, 1.96, 2.58, 3.29, Inf)

# Cartographie du R2
carte1 <- tm_shape(gwr_resultats)+
            tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="Local_R2",
                    palette="YlOrBr",
                    n=5,
                    style="quantile",
                    legend.format = legende_parametres,
                    title = "R2 locaux")+
          tm_layout(frame=FALSE)+
          tm_scale_bar(breaks=c(0,5))

# Cartographie d'une valeur de t
carte2 <- tm_shape(gwr_resultats)+
            tm_borders(col="gray25", lwd=.5)+
            tm_fill(col="NivVieMed_TV",
                    palette="-RdBu",
                    midpoint = NA,
                    breaks = classes_intervalles,
                    legend.format = legende_parametres,
                    title = "Valeur de t\nNiveau de vie (€1000)")+
            tm_layout(frame=FALSE)

tmap_arrange(carte1, carte2)
```

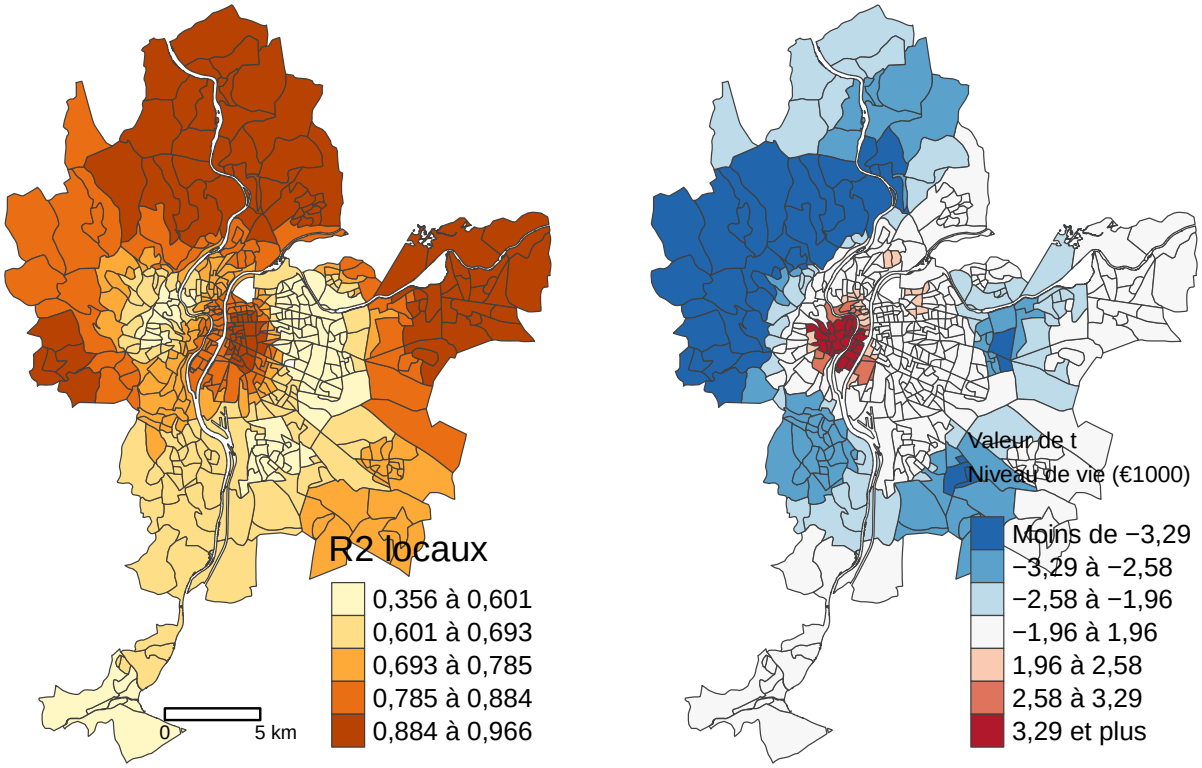


FIGURE 8.10 – Exemple de cartographie avec les résultats de la GWR obtenus avec le *package* `GWmodel`

8.3 GWR classiques pour d'autres distributions

Aller plus loin

Modèles GWR avec d'autres distributions
Dans ce chapitre, nous avons décrit la GWR classique avec une distribution gaussienne qui est une extension de la régression linéaire multiple. Pour construire ce type de GWR classique, nous avons utilisé les fonctions des *packages* `spgwr` (`gwr.sel` et `gwr`) et `GWmodel` (`bw.gwr` et `gwr.basic`). De la même manière, il est possible des construire des modèles GWR qui sont des extensions des modèles linéaires généralisés (GLM), notamment pour des variables dépendantes qualitatives (modèle logistique binomial), de comptage (Poisson, quasi-poisson), continues (gaussienne, inverse gaussienne, Gamma). Pour ce faire, il faut utiliser les fonctions `bw.ggwr` et `ggwr` du *package* `spgwr` et spécifier le type de distribution avec l'argument `family` dont la valeur par défaut est `gaussian()`. Cet argument permet de spécifier la famille de distribution et la fonction de lien en utilisant les familles de distribution disponibles dans la fonction de `glm` (*Generalized Linear Models* pour régressions linéaires généralisées) :

- `binomial(link = "logit")`
- `gaussian(link = "identity")`
- `Gamma(link = "inverse")`
- `inverse.gaussian(link = "1/mu^2")`
- `poisson(link = "log")`
- `quasi(link = "identity", variance = "constant")`
- `quasibinomial(link = "logit")`
- `quasipoisson(link = "log")`

Avec le *package* `GWmodel`, vous pourrez utiliser les fonctions `bw.ggwr` et `gwss` qui permettent de réaliser des régressions linéaires pondérées avec des distributions de Poisson et binomiale (avec l'argument `family = "poisson"` et `family = "binomial"`).

8.4 Limites et critiques des GWR

Bien que la GWR classique représente un outil d'exploration de données spatiales particulièrement intéressant pour évaluer l'importance de l'instabilité spatiale d'un modèle de régression (Apparicio, Séguin et Leloup 2007; Mennis et Jordan 2005; Calvo et Escolar 2003), elle comprend plusieurs limites :

- **Type de distance**. Dans le cadre de ce chapitre, nous avons calculé la GWR avec la distance euclidienne qui n'est pas adaptée à tous les phénomènes à l'étude. Par exemple, Lu, Charlton et Fotheringham (2011) utilisent des distances calculées sur un réseau de rues pour modéliser le prix de vente des maisons à Londres avec un modèle GWR. Pour ce faire, il est possible d'utiliser les fonctions `bw.gwr` `gwr.basic` du *package* `GWmodel` dont l'argument `dmat` permet de spécifier une matrice de distance prédéfinie.
- **Choix du noyau (kernel) et de la taille de la zone d'influence**. Ces deux paramètres de la GWR peuvent affecter grandement les résultats. En effet, la zone d'influence (définie avec une distance ou un nombre donné de plus proches voisins) peut entraîner localement un nombre insuffisant d'observations, compromettant ainsi la robustesse des équations de régression locales.
- **Prédicteurs locaux et globaux**. Nous avons vu que la fonction `LMZ.F3GWR.test` permet de déterminer si les coefficients de régression présentent des variations significatives dans l'espace. Or, certaines variables indépendantes peuvent être introduites globalement et non localement, comme nous le verrons dans le prochain chapitre (section 9.1).

La GWR fait face aussi à d'importantes critiques :

- **Validité des modèles locaux**. Pour une régression linéaire multiple, il est primordial de réaliser un diagnostic pour s'assurer du nombre suffisant d'observations, de l'absence de multicolinéarité excessive et de la normalité et l'homoscédasticité des résidus (lire par exemple cette section (Apparicio et Gelb 2022)). Or, ce diagnostic devrait être réalisé pour tous les modèles locaux obtenus pour les n observations du jeu de données. Par exemple, si le modèle global ne pose pas de problème de multicolinéarité excessive, rien ne garantit qu'elle ne soit pas présente dans plusieurs modèles locaux (Wheeler et Tiefelsdorf 2005).
- **Corrélation entre les coefficients locaux**. Dans un article intitulé *Multicollinearity and correlation among local regression coefficients in geographically weighted regression*, Wheeler et Tiefelsdorf (2005) signalent les coefficients de régression locaux peuvent être fortement corrélés entre eux, en raison d'un problème de multicolinéarité excessive des modèles locaux. En guise d'exemple, la figure 8.11 montre bien que certaines paires de coefficients locaux sont fortement corrélées.

```
library("GGally")

df <- data.frame(modele_gwr$SDF)
df <- df[, 2:7]
names(df) <- c("Constante" , "Pct0_14", "Pct_65", "Pct_Img", "Pct_brevet", "NivVieMed")

# Graphique avec le package GGally
ggpairs(df, progress = FALSE)

# Il est aussi possible de réaliser un graphique avec le package corrplot
# library("corrplot")
# p <- cor(df, method="pearson")
# couleurs <- colorRampPalette(c("#053061", "#2166AC" , "#4393C3", "#92C5DE",
#                                "#D1E5F0", "#FFFFFF", "#FDDBC7", "#F4A582",
#                                "#D6604D", "#B2182B", "#67001F"))
# corrplot.mixed(p, lower="number", lower.col = "black",
#                       upper = "ellipse", upper.col=couleurs(100))
```

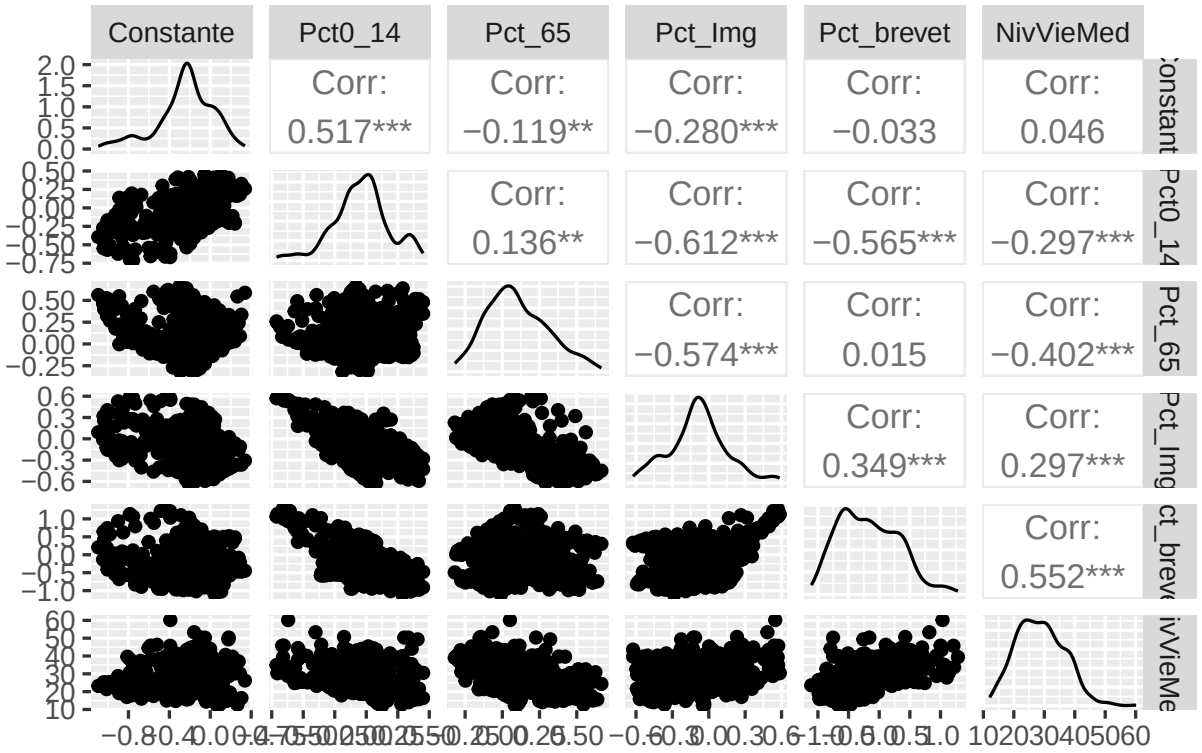


Figure 8.11 – Corrélation de Pearson entre les coefficients de régression locaux de la GWR

8.5 Quiz de révision

Questions

- **Un modèle de régression géographiquement pondérée produit autant de régressions que d'entités spatiales dans le jeu de données à l'étude.**
  - Vrai
  - Faux

  Relisez au besoin la section 8.2.1.
- **Quelles sont les deux principales fonctions noyaux (kernel) pour définir la pondération W(i) dans un modèle GWR?**
  - Fonction quadratique
  - Fonction gaussienne
  - Fonction bicarrée
  - Fonction Epanechnikov
  - Fonction triangulaire

  Relisez au besoin la section 8.1.2.
- **Quelles sont les deux méthodes pour optimiser la zone d'influence optimale avec la fonction gwr.sel?**
  - Méthode des moindres carrés
  - Méthode CV (cross-validation)

  - Méthode basée sur la matrice de corrélation
  - Méthode basée sur l'AIC

  Relisez au besoin la section 8.2.1.
- **Quelles fonctions du package spgwr permettent de comparer les modèles MCO (global)?**

  - anova(Modele.GWR)
  - BFC99.gwr.test(Modele.GWR)
  - BFC02.gwr.test(Modele.GWR)
  - LMZ.F1GWR.test
  - LMZ.F2GWR.test
  - LMZ.F3GWR.test

  Relisez au besoin la section 8.2.1.3.
- **Quelle fonction du package spgwr permet de vérifier si les coefficients de régression varient significativement dans l'espace?**

  - gwr.sel
  - LMZ.F1GWR.test
  - LMZ.F2GWR.test
  - LMZ.F3GWR.test

  Relisez au besoin la section 8.2.1.3.

Réponses

- Un modèle de régression géographiquement pondérée produit autant de régressions que d'entités spatiales dans le jeu de données à l'étude.
  - Vrai
- Quelles sont les deux principales fonctions noyaux (kernel) pour définir la pondération W(i) dans un modèle GWR?
  - Fonction gaussienne
  - Fonction bicarrée
- Quelles sont les deux méthodes pour optimiser la zone d'influence optimale avec la fonction gwr.sel?
  - Méthode CV (cross-validation)
  - Méthode basée sur l'AIC
- Quelles fonctions du package spgwr permettent de comparer les modèles MCO (global)?
  - anova(Modele.GWR)
  - BFC99.gwr.test(Modele.GWR)
  - BFC02.gwr.test(Modele.GWR)
  - LMZ.F1GWR.test
  - LMZ.F2GWR.test
- Quelle fonction du package spgwr permet de vérifier si les coefficients de régression varient significativement dans l'espace?
  - LMZ.F3GWR.test

8.6 Exercices de révision

Exercice

Exercice 1. Réalisation d'un GWR classique

```
library(sf)
library(spgwr)
load("data/Lyon.Rdata")
# Ajout des coordonnées x et y
xy <- à compléter
LyonIris$X <- à compléter
LyonIris$Y <- à compléter
# Optimisation du nombre de voisins avec le CV
formule <- "PM25 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed"
bwCV_voisins <- gwr.sel(à compléter)
# Réalisation de la GWR
modele_gwr <- gwr(à compléter)
# Affichage des résultats
modele_gwr
```

Correction à la section 11.8.1.

Exercice

Exercice 2. Comparaison des modèles MCO et GWR

```
library(sf)
library(spgwr)
# Modèle MCO
modele_global <- à compléter
# Comparaison des R2
r2_global <- à compléter
rss <- à compléter
tss <- à compléter
r2_gwrquasiglobal <- à compléter
cat("R2 global (LM) = ", round(r2_global, 3),
    "\nR2 quasi-global (GWR) : ", round(r2_gwrquasiglobal, 3))
# Comparaison des R2
à compléter
# Les coefficients du modèle GWR varient-ils significativement dans l'espace?
à compléter
```

Correction à la section 11.8.2.

Exercice

Exercice 3. Cartographie des R^2 et des valeurs de t

```
library(tmap)
# Récupération du R carré locaux et valeurs de t locales
à compléter
## Cartographie des R2 locaux
à compléter

# Cartographie des valeurs de t locales
## Paramètres pour la légende
legende_parametres <- list(text.separator = "à",
                           text.less.than = "Moins de",
                           text.or.more =  "et plus",
                           decimal.mark = ",",
                           big.mark = " ")
## Cartographie
à compléter
```

Correction à la section 11.8.3.

9 Extensions de la régression géographiquement pondérée (en cours de rédaction)

Dans le chapitre 8, nous avons présenté les trois formes classiques de la régression géographiquement pondérée (GWR) qui permettent de modéliser des variables dépendantes continues (GWR gaussienne), dichotomiques (GWR logistique) ou des variables de comptage (GWR Poisson). Depuis la publication de l'ouvrage de référence de Steward Fotheringham, Chris Brunsdon et Martin Charlton (2003), plusieurs extensions ont vu le jour. Parmi celles-ci, nous abordons dans ce chapitre :

- la GWR mixte qui permet de spécifier des variables indépendantes variant spatialement et d'autres étant fixes (Fotheringham, Brunsdon et Charlton 2003).
- La régression géographiquement pondérée multiéchelle (*Multiscale Geographically Weighted Regression* – MGWR) (Fotheringham, Yang et Kang 2017).
- Les GWR mixtes intégrant des variables spatialement décalées (MGWR-SAR) (Geniaux et Martinetti 2018).

Objectif

Objectifs d'apprentissage visés dans ce chapitre
À la fin de ce chapitre, vous devriez être en mesure de :
- comprendre pourquoi utiliser différentes extensions de la GWR (GWR-mixte, MGWR, GTWR et MGWR-SAR);
- assimiler les principes fondamentaux de ces différentes extensions de la GWR;
- appréhender les différentes extensions de la GWR (GWR-mixte, MGWR, GTWR et MGWR-SAR);
- analyser les résultats produits par ces différentes extensions;
- mettre en pratique ces extensions de la GWR dans R.

Package

Liste des *packages* utilisés dans ce chapitre
- Pour importer et manipuler des fichiers géographiques :
  - `sf` pour importer et manipuler des données vectorielles.
  - `dplyr` pour manipuler les données.
- Pour construire des cartes et des graphiques :
  - `tmap` pour les cartes.
  - `ggplot2` pour construire des graphiques.
- Pour construire différentes extensions de la GWR :
  - `GWmodel` pour construire des GWR mixtes, des MGWR et des GTWR.
  - `mgwrsar` pour construire des GWR avec des variables spatialement décalées.
  - `spdep` pour construire des matrices de pondération spatiales et calculer le I de Moran.

9.1 Régression géographiquement pondérée mixte

9.1.1 Principe de base de la GWR mixte

9.1.1.1 Pourquoi recourir à une GWR mixte

9.1.1.2 Formulation de la GWR mixte

9.1.2 Mise en oeuvre de la GWR mixte dans R

9.2 Régression géographiquement pondérée multiéchelle

9.2.1 Principe de base de la MGWR

9.2.1.1 Pourquoi recourir à une GWR mixte

9.2.1.2 Formulation de la GWR mixte

9.2.2 Mise en oeuvre de la MGWR dans R

9.3 Régression géographiquement pondérée mixte avec des variables spatialement décalée

9.3.1 Principe de base de la MGWR-SAR

9.3.1.1 Pourquoi recourir à une MGWR-SAR

9.3.1.2 Formulation de la MGWR-SAR

9.3.2 Mise en oeuvre de la MGWR-SAR dans R

9.4 Quiz de révision

9.5 Exercices de révision

Exercice

Exercice 1. À compléter

```
library(sf)
library(spgwr)
```

Correction à la section 11.9.1.

Exercice

Exercice 2. À compléter

```
library(sf)
library(spgwr)
```

Correction à la section 11.9.2.

Exercice

Exercice 2. À compléter
Correction à la section 11.9.3.

10 Modèles GAM et GLMM avec des coefficients variant spatialement (en cours de rédaction)

10.1 Modèles généralisés additifs (GAM)

10.2 Modèles linéaires généralisés à effets mixtes (GLMM)

10.3 Quiz de révision

10.4 Exercices de révision

Exercice

Exercice 1. À compléter
Complétez le code ci-dessous.
Correction à la section 11.10.1.

Exercice

Exercice 2. À compléter
Complétez le code ci-dessous.
Correction à la section 11.10.2.

Exercice

Exercice 3. À compléter
Complétez le code ci-dessous.
Correction à la section 11.10.3.

Partie 6. Conclusions

11 Correction des exercices

11.1 Exercices du chapitre 1

11.1.1 Exercice 1

```
load("data/Lyon.Rdata")
library(spdep)
## Matrice de contiguïté selon le partage d'un nœud (Queen)
nb_queen <- poly2nb(LyonIris, queen = TRUE)
w_queen <- nb2listw(nb_queen, zero.policy = TRUE, style = "W")

# I de Moran sur la variable lden elon l'hypothèse de la normalité
moran.test(LyonIris$Lden, listw=w_queen, zero.policy = TRUE, randomisation = FALSE)

# I de Moran sur la variable lden selon la normalité selon l'hypothèse de la randomisation
moran.test(LyonIris$Lden, listw=w_queen, zero.policy = TRUE, randomisation = TRUE)

# I de Moran sur la variable lden selon des permutations Monte-Carlo
moran.mc(LyonIris$Lden, listw=w_queen, zero.policy = TRUE, nsim = 999)
```

11.1.2 Exercice 2

```
library(spdep)
load("data/Lyon.Rdata")

# Modèle MCO
formule <- "Lden ~ Pct0_14 + Pct_65 + Pct_Img + Pct_brevet + NivVieMed"
modele_mco <- lm(formule, data = LyonIris)

# Résultats du modèle
summary(modele_mco)

## Matrice de contiguïté (Rook)

nb_rook <- poly2nb(LyonIris, queen = FALSE)
w_rook <- nb2listw(nb_rook, zero.policy = TRUE, style = "W")
```

```
# I de Moran sur les résidus du modèle global (MCO)
lm.morantest(modele_mco, w_rook)

# Cartographie des résidus
LyonIris$MCO.Residus <- modele_mco$residuals
tmap_mode("plot")
tm_shape(LyonIris)+
         tm_borders(col="gray25", lwd=.5)+
         tm_fill(col="MCO.Residus", n = 6, style = "pretty",
                 legend.format = list(text.separator = "à",
                                      decimal.mark = ","),
                 midpoint = 0,
                 palette = "-RdBu",
                 title = "MCO") +
         tm_layout(frame = FALSE)+
         tm_scale_bar(breaks = c(0,5))
```

11.2 Exercices du chapitre 2

11.2.1 Exercice 1

11.2.2 Exercice 2

11.2.3 Exercice 3

11.3 Exercices du chapitre 3

11.3.1 Exercice 1

```
library(sf)
library(spatialreg)
load("data/Lyon.Rdata")

# Matrice de contiguïté selon le partage d'un segment
nb_rook <- poly2nb(LyonIris, queen=FALSE)
w_rook <- nb2listw(nb_rook, zero.policy = TRUE, style = "W")

# Modèles
formule <- "PM25 ~  Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed"

# Modèles MOC, SLX, SDM, SDEM, SAR, SEM, Manski
MCO     <- lm(formule, data = LyonIris)
```

```
SLX     <- lmSLX(formule, listw = w_rook, data = LyonIris)
Manski  <- sacsarlm(formule, listw=w_rook, data = LyonIris, type="sacmixed")
SDM     <- lagsarlm(formule, listw = w_rook, data = LyonIris, type = "mixed")
SDEM    <- errorsarlm(formule, listw=w_rook, data = LyonIris, etype = 'emixed')
SAR     <- lagsarlm(formule,listw=w_rook, data = LyonIris, type = 'lag')
SEM     <- errorsarlm(formule, listw=w_rook, data = LyonIris)
```

11.3.2 Exercice 2

```
summary(lm.LMtests(model = MCO,
                   listw = w_rook,
                   test = c("LMlag","LMerr","RLMlag","RLMerr")))
```

11.3.3 Exercice 3

```
## Valeurs d'AIC et de BIC
AICs <- AIC(MCO, SLX, SDM, SDEM, SAR, SEM)
BICs <- BIC(MCO, SLX, SDM, SDEM, SAR, SEM)

## Autocorrélation spatiale des résidus
IMoran_MCO  <- moran.mc(resid(MCO), w_rook, nsim = 999)
IMoran_SLX  <- moran.mc(resid(SLX), w_rook, nsim = 999)
IMoran_SAR  <- moran.mc(resid(SAR), w_rook, nsim = 999)
IMoran_SEM  <- moran.mc(resid(SEM), w_rook, nsim = 999)
IMoran_SDM  <- moran.mc(resid(SDM), w_rook, nsim = 999)
IMoran_SDEM <- moran.mc(resid(SDEM), w_rook, nsim = 999)

MoranI_s <- c(IMoran_MCO$statistic, IMoran_SLX$statistic,
              IMoran_SAR$statistic, IMoran_SEM$statistic,
              IMoran_SDM$statistic, IMoran_SDEM$statistic)

MoranI_p <- c(IMoran_MCO$p.value, IMoran_SLX$p.value,
              IMoran_SAR$p.value, IMoran_SEM$p.value,
              IMoran_SDM$p.value, IMoran_SDEM$p.value)

## Tableau
Comparaison <- data.frame(Modele = c("MCO", "SLX", "SAR", "SEM", "SDM", "SDEM"),
                          AIC = round(AICs$AIC,2),
                          BIC = round(BICs$BIC,2),
                          dl = AICs$df,
                          MoranI = round(MoranI_s,3),
                          MoranIp = round(MoranI_p,3))
Comparaison
```

11.4 Exercices du chapitre 4

11.4.1 Exercice 1

11.4.2 Exercice 2

11.4.3 Exercice 3

11.5 Exercices du chapitre 5

11.5.1 Exercice 1

11.5.2 Exercice 2

11.5.3 Exercice 3

11.6 Exercices du chapitre 6

11.6.1 Exercice 1

```
library(sf)
library(mgcv)
library(car)
library(DHARMa)
library(mgcViz)
# Chargement du jeu de données sur l'agglomération Lyonnaise
load("data/Lyon.Rdata")

# Vérification de la multicolinéarité (VIF)
vif(lm(PM25 ~ Lden+Pct_65+Pct_Img+Pct_brevet+NivVieMed, data = LyonIris))

# Construction du modèle GLM gaussien
modele_gam1 <- gam(PM25 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed,
                   family = gaussian, data = LyonIris)
# Résidus simulés du modèle gaussien
gam1_res <- simulateResiduals(modele_gam1, plot = FALSE)
plot(gam1_res)

# Construction du modèle GLM avec une distribution de Student
modele_gam2 <- gam(PM25 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed,
                   family = scat, data = LyonIris)

# Résidus simulés avec une distribution de Student
```

```r
gam2_res <- simulateResiduals(modele_gam2, plot = FALSE)
plot(gam2_res)

# Comparaison des deux modèles
AIC(modele_gam1, modele_gam2)

# Résultats du modèle gaussien
summary(modele_gam2)

# Matrice de contiguïté selon le partage d'un nœud
queen_nb <- poly2nb(LyonIris, queen = TRUE)
queen_w <- nb2listw(queen_nb, style = 'W', zero.policy = TRUE)

# I de Moran sur les résidus gaussiens
moran.mc(residuals(modele_gam2,
                   type = 'pearson'),
                   listw = queen_w, nsim = 999,
                   zero.policy = TRUE)

# I de Moran sur les résidus simulés
moran.mc(residuals(gam2_res,
                   type = 'pearson'),
                   listw = queen_w, nsim = 999,
                   zero.policy = TRUE)
```

11.6.2 Exercice 2

```r
# Chargement du jeu de données sur l'agglomération Lyonnaise
load("data/Lyon.Rdata")

# Ajout des coordonnées x et y dans la couche sf LyonIris
XY <- st_coordinates(st_centroid(LyonIris))
LyonIris$X <- XY[,1]
LyonIris$Y <- XY[,2]

# Modèle GAM avec la distribution de student et une spline
# sur les coordonnées x et y avec la distribution de Student
modele_gam_spline <- gam(NO2 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed+
                         s(X,Y, bs = 'gp', m = 3),
                         data = LyonIris, family = scat)

summary(modele_gam_spline)
```

11.7 Exercices du chapitre 7

11.7.1 Exercice 1

11.7.2 Exercice 2

11.7.3 Exercice 3

11.8 Exercices du chapitre 8

11.8.1 Exercice 1

```
library(sf)
library(spgwr)
load("data/Lyon.Rdata")
# Ajout des coordonnées x et y
xy <- st_coordinates(st_centroid(LyonIris))
LyonIris$X <- xy[,1]
LyonIris$Y <- xy[,2]

# Optimisation du nombre de voisins avec le CV
formule <- "PM25 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed"
bwaCV_voisins  <- gwr.sel(formule,
                          data = LyonIris,
                          method = "cv",
                          gweight=gwr.bisquare,
                          adapt=TRUE,
                          verbose = FALSE,
                          RMSE = TRUE,
                          longlat = FALSE,
                          coords=cbind(LyonIris$X,LyonIris$Y))

# Optimisation du nombre de voisins avec l'AIC
formule <- "PM25 ~ Pct0_14+Pct_65+Pct_Img+Pct_brevet+NivVieMed"
bwaCV_voisins  <- gwr.sel(formule,
                          data = LyonIris,
                          method = "CV",
                          gweight=gwr.bisquare,
                          adapt=TRUE,
                          verbose = FALSE,
                          RMSE = TRUE,
                          longlat = FALSE,
                          coords=cbind(LyonIris$X,LyonIris$Y))
```

```
# Réalisation de la GWR
modele_gwr <- gwr(formule,
                  data = LyonIris,
                  adapt=bwaCV_voisins,
                  gweight=gwr.bisquare,
                  hatmatrix=TRUE,
                  se.fit=TRUE,
                  coords=cbind(LyonIris$X,LyonIris$Y),
                  longlat=FALSE)

# Affichage des résultats
modele_gwr
```

11.8.2 Exercice 2

```
# Modèle MCO
modele_global <- lm(formule, data = LyonIris)

# Comparaison des R2
r2_global <- summary(modele_global)$r.squared
rss <- sum((modele_gwr$lm$y - modele_gwr$SDF$pred)^2)
tss <- sum((modele_gwr$lm$y - mean(modele_gwr$SDF$pred))^2)
R2_GWRquasiglobal <- 1 - (rss / tss)
cat("R2 global (LM) = ", round(r2_global, 3),
    "\nR2 quasi-global (GWR) : ", round(R2_GWRquasiglobal, 3))

# Comparaison des R2
anova(modele_gwr)
LMZ.F1GWR.test(modele_gwr)
LMZ.F2GWR.test(modele_gwr)

# Les coefficients du modèle GWR varient-ils significativement dans l'espace?
LMZ.F3GWR.test(modele_gwr)
```

11.8.3 Exercice 3

```
library(tmap)

# Récupération du R carré locaux et valeurs de t locales
LyonIris$GWR.R2 <- modele_gwr$SDF$localR2
vars_indep <- c("Pct0_14", "Pct_65", "Pct_Img", "Pct_brevet", "NivVieMed")
for(e in vars_indep){
```

```
  # Nom des nouvelles variables
  var.coef <- paste0("GWR.", "B_", e)
  var.t    <- paste0("GWR.", "T_", e)
  # Récupération des coefficients pour les variables indépendantes
  LyonIris[[var.coef]] <- modele_gwr$SDF[[e]]
  # Calcul des valeurs de t pour les variables indépendantes
  LyonIris[[var.t]]    <- modele_gwr$SDF[[e]] / modele_gwr$SDF[[paste0(e, "_se")]]
}

# Cartographie des R2 locaux
legendes_parametres <- list(text.separator = "à",
                            decimal.mark = ",",
                            big.mark = " ")
tm_shape(LyonIris)+
  tm_borders(col="gray25", lwd=.5)+
  tm_fill(col="GWR.R2",
          palette="YlOrBr",
          n=5, style="quantile",
          legend.format = legendes_parametres,
          title = "R2 locaux")+
  tm_layout(frame = FALSE)+
  tm_scale_bar(breaks=c(0,5))

# Cartographie des valeurs de t locales
## Paramètres pour la légende
legende.parametres <- list(text.separator = "à",
                           text.less.than = "Moins de",
                           text.or.more =  "et plus",
                           decimal.mark = ",",
                           big.mark = " ")

## Cartographie
classes_intervalles = c(-Inf, -3.29, -2.58, -1.96, 1.96, 2.58, 3.29, Inf)
carte1 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
          tm_fill(col="GWR.T_Pct0_14", palette="-RdBu",
                  midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende.parametres,
                  title = "Moins de 15 ans (%)")+
          tm_layout(frame = FALSE, legend.outside = TRUE)

carte2 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
          tm_fill(col="GWR.T_Pct_65", palette="-RdBu",
                  midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende.parametres,
```

```
                title = "65 ans et plus (%)")+
          tm_layout(frame = FALSE, legend.outside = TRUE)

carte3 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
          tm_fill(col="GWR.T_Pct_Img", palette="-RdBu",
                  midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende.parametres,
                  title = "Immigrants (%)")+
          tm_layout(frame = FALSE, legend.outside = TRUE)

carte4 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
          tm_fill(col="GWR.B_Pct_brevet", palette="-RdBu",
                  midpoint = NA,
                  breaks = classes_intervalles,
                  legend.format = legende.parametres,
                  title = "Faible scolarité (%)")+
          tm_layout(frame = FALSE, legend.outside = TRUE)

carte5 <- tm_shape(LyonIris)+ tm_borders(col="gray25", lwd=.5)+
          tm_fill(col="GWR.T_NivVieMed", palette="-RdBu",
                 midpoint = NA,
                 breaks = classes_intervalles,
                 legend.format = legende.parametres,
                 title = "Niveau de vie (€1000)")+
          tm_layout(frame = FALSE, legend.outside = TRUE)+
          tm_scale_bar(breaks=c(0,5))

tmap_arrange(carte1, carte2, carte3, carte4, carte5, ncol = 2, nrow=3)
```

11.9 Exercices du chapitre 9

11.9.1 Exercice 1

11.9.2 Exercice 2

11.9.3 Exercice 3

11.10 Exercices du chapitre 10

11.10.1 Exercice 1

11.10.2 Exercice 2

11.10.3 Exercice 3

12 Conclusion générale (en cours de rédaction)

Bibliographie

Anselin, Luc. 1988. *Spatial econometrics: methods and models*. Kluwer Academic Publishers.

———. 2010. « Thirty years of spatial econometrics. » *Papers in regional science* 89 (1): 3-26. https://doi.org/10.1111/j.1435-5957.2010.00279.x.

Anselin, Luc et AK Bera. 1998. « Spatial dependence in linear regression models with an introduction to spatial econometrics, A. Ullah and DEA Giles (eds.), Handbook of Applied Economics Statistics. » In *Spatial Dependence in linear Regression Models with an Introduction to Spatial Econometrics*, sous la dir. de Aman Ullah, 237-289. CRC Press. https://doi.org/10.1201/9781482269901-36.

Anselin, Luc, Anil K Bera, Raymond Florax et Mann J Yoon. 1996. « Simple diagnostic tests for spatial dependence. » *Regional science and urban economics* 26 (1): 77-104. https://doi.org/10.1016/0166-0462(95)02111-6.

Anselin, Luc et Raymond JGM Florax. 1995. *New directions in spatial econometrics*. Springer.

Anselin, Luc, Raymond Florax et Sergio J Rey. 2013. *Advances in spatial econometrics: methodology, tools and applications*. Springer Science.

Anselin, Luc et Sergio J Rey. 2014. *Modern spatial econometrics in practice: A guide to GeoDa, GeoDaSpace and PySAL*. GeoDa Press LLC.

Apparicio, Philippe et Jérémy Gelb. 2022. *Méthodes quantitatives en sciences sociales : un grand bol d'R*. FabriqueREL, Licence CC BY-SA. https://serieboldr.github.io/MethodesQuantitatives/.

———. 2024. *Méthodes d'analyse spatiales : un grand bol d'R*. FabriqueREL, Licence CC BY-SA. https://serieboldr.github.io/MethodesAnalyseSpatiale/.

Apparicio, Philippe, Jérémy Gelb, Anne-Sophie Dubé, Simon Kingham, Lise Gauvin et Éric Robitaille. 2017. « The approaches to measuring the potential spatial access to urban health services revisited: distance types and aggregation-error issues. » *International journal of health geographics* 16 (1): 32. https://doi.org/10.1186/s12942-017-0105-9.

Apparicio, Philippe, Anne-Marie Séguin et Xavier Leloup. 2007. « Modélisation spatiale de la pauvreté à Montréal: apport méthodologique de la régression géographiquement pondérée. » *The Canadian Geographer/Le Géographe canadien* 51 (4): 412-427. https://doi.org/10.1111/j.1541-0064.2007.00189.x.

Binbin Lu, Paul Harris, Martin Charlton et Christopher Brunsdon. 2014. « The GWmodel R package: further topics for exploring spatial heterogeneity using geographically weighted models. » *Geo-spatial Information Science* 17 (2): 85-101. https://doi.org/10.1080/10095020.2014.917453.

Bivand, Roger, Giovanni Millo et Gianfranco Piras. 2021. « A review of software for spatial econometrics in R. » *Mathematics* 9 (11): 1276. https://doi.org/10.3390/math9111276.

Bivand, Roger, Edzer J Pebesma, Virgilio Gómez-Rubio et Edzer Jan Pebesma. 2008. *Applied spatial data analysis with R*. Vol. 747248717. Springer.

Bivand, Roger et Danlin Yu. 2023. *spgwr: Geographically Weighted Regression*. s.n. https://CRAN.R-project.org/package=spgwr.

Boogaart, K Gerald van den et Raimon Tolosana-Delgado. 2013. *Analyzing compositional data with R*. Vol. 122. Springer.

Brunsdon, Chris, A Stewart Fotheringham et Martin Charlton. 1998. « Spatial nonstationarity and autoregressive models. » *Environment and Planning A* 30 (6). SAGE Publications Sage UK: London, England: 957-973.

Brunsdon, Chris, A Stewart Fotheringham et Martin E Charlton. 1996. « Geographically weighted regression: a method for exploring spatial nonstationarity. » *Geographical analysis* 28 (4). Wiley Online Library: 281-298.

Calvo, Ernesto et Marcelo Escolar. 2003. « The local voter: A geographically weighted approach to ecological inference. » *American Journal of Political Science* 47 (1): 189-204. https://doi.org/10.1111/1540-5907.00013.

Chi, Guangqing et Jun Zhu. 2019. *Spatial regression models for the social sciences*. SAGE publications.

Dubé, Jean, Julie Le Gallo, François Des Rosiers, Diègo Legros et Marie-Pier Champagne. 2024. « An integrated causal framework to evaluate uplift value with an example on change in public transport supply. » *Transportation research part E: logistics and transportation review* 185. Elsevier: 103500.

Dubé, Jean et Diègo Legros. 2014. *Econométrie spatiale appliquée des microdonnées*. ISTE Group.

Dubé, Jean, Diègo Legros, Marius Thériault et François Des Rosiers. 2017. « Measuring and interpreting urban externalities in real-estate data: A spatio-temporal difference-in-differences (stdid) estimator. » *Buildings* 7 (2): 51. https://doi.org/10.3390/buildings7020051.

Elhorst, JP. 2014. *Spatial econometrics: from cross-sectional data to spatial panels*. springer. https://link.springer.com/book/10.1007/978-3-642-40340-8.

Fotheringham, A Stewart, Chris Brunsdon et Martin Charlton. 2003. *Geographically weighted regression: the analysis of spatially varying relationships*. John Wiley & Sons.

Fotheringham, A Stewart, Martin Charlton et Chris Brunsdon. 1996. « The geography of parameter space: an investigation of spatial non-stationarity. » *International journal of geographical information systems* 10 (5). Taylor & Francis: 605-627.

Fotheringham, A Stewart, Wenbai Yang et Wei Kang. 2017. « Multiscale geographically weighted regression (MGWR). » *Annals of the American Association of Geographers* 107 (6). Taylor & Francis: 1247-1265. https://doi.org/10.1080/24694452.2017.1352480.

Geary, Robert C. 1954. « The contiguity ratio and statistical mapping. » *The incorporated statistician* 5 (3): 115-146. https://doi.org/10.2307/2986645.

Gelb, Jérémy et Philippe Apparicio. 2022. « Cyclists' exposure to air and noise pollution, comparative approach in seven cities. » *Transportation Research Interdisciplinary Perspectives* 14: 100619. https://doi.org/10.1016/j.trip.2022.100619.

Geniaux, Ghislain et Davide Martinetti. 2018. « A new method for dealing simultaneously with spatial autocorrelation and spatial heterogeneity in regression models. » *Regional Science and Urban Economics* 72: 74-85. https://doi.org/10.1016/j.regsciurbeco.2017.04.001.

Greenacre, Michael, Marina Martinez-Alvaro et Agustin Blasco. 2021. « Compositional data analysis of microbiome and any-omics datasets: a validation of the additive logratio transformation. » *Frontiers in microbiology* 12: 727398. https://doi.org/10.3389/fmicb.2021.727398.

Griffith, Daniel A. 1988. *Advanced Spatial Statistics: Special Topics in the Exploration of Quantitative Spatial Data Series*. Kluwer Academic Publishers.

———. 2003. *Spatial filtering*. Springer.

Griffith, Daniel A., Yongwan Chun et Bin Li. 2019. *Spatial regression analysis using eigenvector spatial filtering*. Academic Press.

Griffith, Daniel A. et Pedro R Peres-Neto. 2006. « Spatial modeling in ecology: the flexibility of eigenfunction spatial analyses. » *Ecology* 87 (10): 2603-2613. https://doi.org/10.1890/0012-9658(2006)87%5B2603:SMIETF%5D2.0.CO;2.

Helbich, Marco et Daniel A. Griffith. 2016. « Spatially varying coefficient models in real estate: Eigenvector spatial filtering and alternative approaches. » *Computers, Environment and Urban Systems* 57: 1-11. https://doi.org/10.1016/j.compenvurbsys.2015.12.002.

Isabella Gollini, Binbin Lu, Martin Charlton, Christopher Brunsdon et Paul Harris. 2015. « GWmodel: An R Package for Exploring Spatial Heterogeneity Using Geographically Weighted Models. » *Journal of Statistical Software* 63 (17): 1-50. https://doi.org/10.18637/jss.v063.i17.

Kelejian, Harry H et Ingmar R Prucha. 1998. « A generalized spatial two-stage least squares procedure for estimating a spatial autoregressive model with autoregressive disturbances. » *The journal of real estate finance and economics* 17. Springer: 99-121. https://doi.org/10.1023/A:1007707430416.

Kim, Chong Won, Tim T Phipps et Luc Anselin. 2003. « Measuring the benefits of air quality improvement: a spatial hedonic approach. » *Journal of environmental economics and management* 45 (1): 24-39. https://doi.org/10.1016/S0095-

0696(02)00013-X.

Le Gallo, Julie. 2002. « Econométrie spatiale : l'autocorrélation spatiale dans les modèles de régression linéaire. » *Economie prevision* 155 (4): 139-157.

LeSage, James P et R. Kelly Pace. 2009. *An introduction to spatial econometrics*. 123. CRC Press.

Lu, Binbin, Martin Charlton et A Stewart Fotheringham. 2011. « Geographically weighted regression using a non-Euclidean distance metric with a study on London house price data. » *Procedia Environmental Sciences* 7: 92-97. https://doi.org/10.1016/j.proenv.2011.07.017.

Mantel, Nathan. 1967. « The detection of disease clustering and a generalized regression approach. » *Cancer research* 27 (2): 209-220.

Mennis, Jeremy L et Lisa Jordan. 2005. « The distribution of environmental equity: Exploring spatial nonstationarity in multivariate models of air toxic releases. » *Annals of the Association of American Geographers* 95 (2): 249-268. https://doi.org/10.1111/j.1467-8306.2005.00459.x.

Moran, Patrick. 1948. « The interpretation of statistical maps. » *Journal of the Royal Statistical Society. Series B (Methodological)* 10 (2): 243-251. https://www.jstor.org/stable/2983777.

———. 1950. « A test for the serial independence of residuals. » *Biometrika* 37 (1/2): 178-181. https://doi.org/10.2307/2332162.

Murakami, Daisuke et Daniel A. Griffith. 2015. « Random effects specifications in eigenvector spatial filtering: a simulation study. » *Journal of Geographical Systems* 17. Springer: 311-331.

Pace, R Kelley et James P LeSage. 2008. « A spatial Hausman test. » *Economics Letters* 101 (3): 282-284. https://doi.org/10.1016/j.econlet.2008.09.003.

Paelinck, Jean. 1978. « Spatial econometrics. » *Economics Letters* 1 (1): 59-63. https://doi.org/10.1016/0165-1765(78)90097-6.

Seya, Hajime, Daisuke Murakami, Morito Tsutsumi et Yoshiki Yamagata. 2015. « Application of LASSO to the eigenvector selection problem in eigenvector-based spatial filtering. » *Geographical analysis* 47 (3): 284-299. https://doi.org/10.1111/gean.12054.

Steimetz, Seiji SC. 2010. « Spatial multipliers in hedonic analysis: a comment on "spatial hedonic models of airport noise, proximity, and housing prices". » *Journal of Regional Science* 50 (5): 995-998. https://doi.org/10.1111/j.1467-9787.2010.00679.x.

Tobler, Waldo R. 1970. « A computer movie simulating urban growth in the Detroit region. » *Economic geography* 46 (sup1): 234-240. https://doi.org/10.2307/143141.

Tsagris, Michail et Giorgos Athineou. 2025. *Compositional: Compositional Data Analysis*. s.n. https://CRAN.R-project.org/package=Compositional.

van den Boogaart, K. Gerald, Raimon Tolosana-Delgado et Matevz Bren. 2024. *Compositions: Compositional Data Analysis*. s.n. https://CRAN.R-project.org/package=compositions.

Wheeler, David et Michael Tiefelsdorf. 2005. « Multicollinearity and correlation among local regression coefficients in geographically weighted regression. » *Journal of Geographical Systems* 7 (2): 161-187. https://doi.org/10.1007/s10109-005-0155-6.

Wood, Simon N. 2017. *Generalized additive models: an introduction with R*. Chapman; hall/CRC.