Modélisation des données Spatiales

Pâquarse Delvich Van Mahouvi, Ronan Gonnet, Romain Ponsort

2022-12-12

clear data & close graphs

Définition du répertoire

```
setwd("~/Cours M2 MAS/Modele de donnees spartiales/Projet Delvich-Ronan-Romain")
```

Les packages

```
library(magrittr)
```

```
Warning: le package 'magrittr' a été compilé avec la version R 4.2.2
```

```
library(RColorBrewer)
library(hrbrthemes)
```

```
NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

      Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and

      if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```
library(rgdal)
```

```
Le chargement a nécessité le package : sp

Please note that rgdal will be retired by the end of 2023,
plan transition to sf/stars/terra functions using GDAL and PROJ
at your earliest convenience.

rgdal: version: 1.5-32, (SVN revision 1176)
Geospatial Data Abstraction Library extensions to R successfully loaded
Loaded GDAL runtime: GDAL 3.4.3, released 2022/04/22
Path to GDAL shared files: C:/Users/Paqua/AppData/Local/R/win-library/4.2/rgdal/gdal
GDAL binary built with GEOS: TRUE
Loaded PROJ runtime: Rel. 7.2.1, January 1st, 2021, [PJ_VERSION: 721]
Path to PROJ shared files: C:/Users/Paqua/AppData/Local/R/win-library/4.2/rgdal/proj
PROJ CDN enabled: FALSE
Linking to sp version:1.5-0
To mute warnings of possible GDAL/OSR exportToProj4() degradation,
use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.
```

```
library(dplyr)
```

```

Attachement du package : 'dplyr'

Les objets suivants sont masqués depuis 'package:stats':

    filter, lag

Les objets suivants sont masqués depuis 'package:base':

    intersect, setdiff, setequal, union
```

Contents

Introduction

Le cambriolage est un phénomène inquiétant en France. En 2018, 1,7% des ménages ont déclaré avoir été victimes d'un cambriolage de leur résidence principale (INSEE). Pour 55% d'entre eux le cambriolage n'a pas abouti et s'est arrêté au stade de la tentative.

La moitié des victimes de cambriolage déposent plainte à la police ou à la gendarmerie. En 2019, les services de police et de gendarmerie ont enregistré 217 500 cambriolages et tentatives de cambriolage de résidence principale.

On a décidé d'appuyé nos analyses sur des données départementales de l'année 2019. L'objectif sera donc d'apprécier l'influence du taux de cambriolage des départements voisins sur un département mais aussi l'influence d'autres facteurs comme le taux de chômage sur ce fléau.

1 Ouverture de la base

L'intégralité des données utilisées proviennent de l'INSEE. Pour plusieurs variables, le nom de la variable a été modifié pour permettre une utilisation plus simple lors de leurs traitements.

```
base <- openxlsx::read.xlsx(xlsxFile = "bdd_projet.xlsx", rowNames = TRUE)
```

2 Visualisation et pré-traitement des données

2.1 Visualisation des données

```
stargazer::stargazer(base, type = "text", title = "vue des données")
```

```
vue des données
===============================================================
Statistic                    N   Mean   St. Dev.  Min     Max
---------------------------------------------------------------
Cambriolage                  96  5.835   2.302   1.900  12.100
Taux_chomage_2019            96  8.125   1.588   4.925  13.625
Domicile_au_moins_une_voiture 96 84.842   7.183   33.500 91.800
Pop_active                   96 44.936   2.297   39.893 55.033
Ratio_homme_femme            96  0.942   0.019   0.880  0.990
Part_Maison                  96 63.217   17.420  0.800  86.500
Part_Appartement             96 35.861   17.255  12.900 96.900
Policier_Gendarme            96  3.223   1.262   1.600  11.700
Personne_Seule               96 369.000  31.708   294    518
---------------------------------------------------------------
```

Pour une meilleure compréhension de nos variables, on observe également la nature des variables potentielles.

```
str(base)
```

```
'data.frame':   96 obs. of  10 variables:
 $ Code                         : chr  "01" "02" "03" "04" ...
 $ Cambriolage                  : num  8.2 7 5 5 2.4 8 4.6 4.6 4.4 6.8 ...
 $ Taux_chomage_2019            : num  6.05 11.65 8.9 9.75 7.97 ...
 $ Domicile_au_moins_une_voiture: num  91 83.1 85 87.9 88.4 77.7 90 83 87.7 82.7 ...
 $ Pop_active                   : num  47.9 44.5 42.6 42.7 45.2 ...
 $ Ratio_homme_femme            : num  0.97 0.95 0.92 0.96 0.96 0.9 0.95 0.95 0.97 0.94 ...
 $ Part_Maison                  : num  63.2 75.5 71.7 60.8 43.1 23.7 74.7 71.1 79.6 65.6 ...
 $ Part_Appartement             : num  36 23.7 27.6 38 56.2 75.6 24.5 28.2 19.1 33.4 ...
```

```
 $ Policier_Gendarme             : num  2 3 4.1 3.9 4.5 3.3 2.6 3.5 3.7 3.2 ...
 $ Personne_Seule                : num  322 338 413 389 389 402 356 352 388 386 ...
```

- Verification des valeur manquantes

```
sum(is.na(base))
```

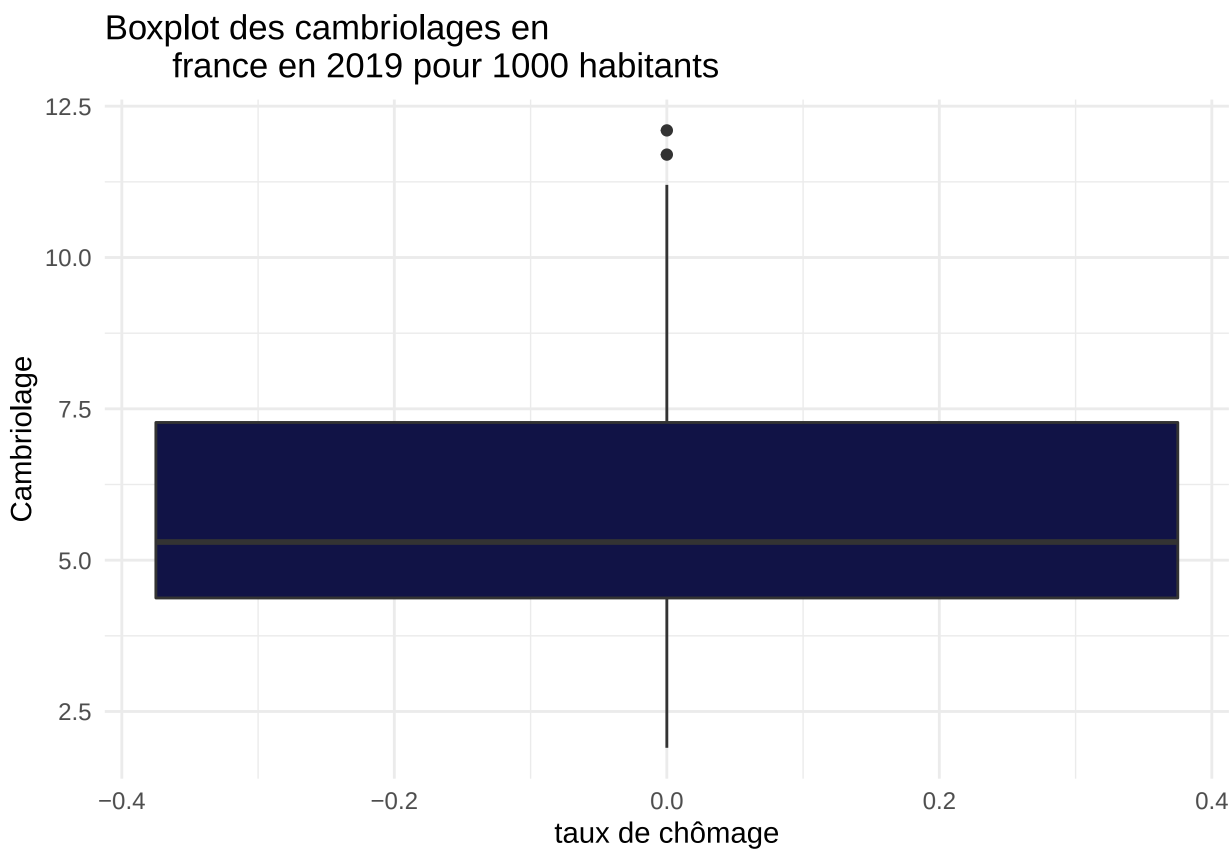
```
[1] 0
```

On peut donc remarquer l'absence de valeurs manquantes dans notre base de données.

2.2 Statistique descriptive Univariée

2.2.1 BoxPlot du taux de cambriolage pour 1000 habitants

```
options(width = 200)
library(ggplot2)
ggplot(base) +
  aes(x = Cambriolage) +
  geom_boxplot(fill = "#111346", position = "dodge2", orientation = "y") +
  labs(y = "taux de chômage",title = "Boxplot des cambriolages en
       france en 2019 pour 1000 habitants") +
  theme_minimal() +
  theme(plot.subtitle = element_text(face = "italic")) +
  coord_flip()
```

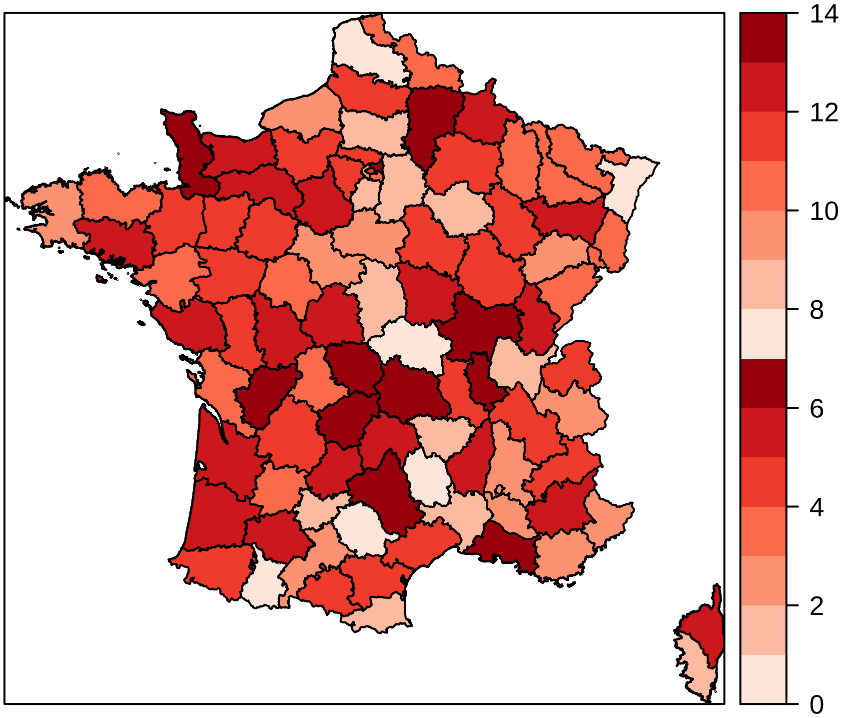


2.2.2 Cartographie du taux de cambriolage pour 1000 habitants

```r
# lecture du shapefile
carte <- raster::getData('GADM', country='FRA', level=2)
```

```
Warning in raster::getData("GADM", country = "FRA", level = 2): getData will be removed in a future ver
. Please use the geodata package instead
```

```r
# fusion des tables et suppression des variables inutiles
carte@data <- merge(carte@data, base, by.x = "CC_2", by.y = "Code")

# fusion des tables et suppression des variables inutiles
carte@data <- carte@data%>%
  dplyr::select(c("CC_2", "Cambriolage","NAME_1", "NAME_2", "Domicile_au_moins_une_voiture",
                  "Pop_active", "Ratio_homme_femme", "Taux_chomage_2019",
                  "Ratio_homme_femme", "Part_Maison", "Part_Appartement",
                  "Policier_Gendarme", "Personne_Seule"
                  ))

sp::spplot(carte, "Cambriolage", sub = "Taux de cambriolage en france pour 1000 habitants
           en 2019",
           at = seq(0, 14, 1), col.regions = brewer.pal(7, "Reds"),
           border = "white")
```



Taux de cambriolage en france pour 1000 habitants en 2019

2.3 Statistiques bivariée

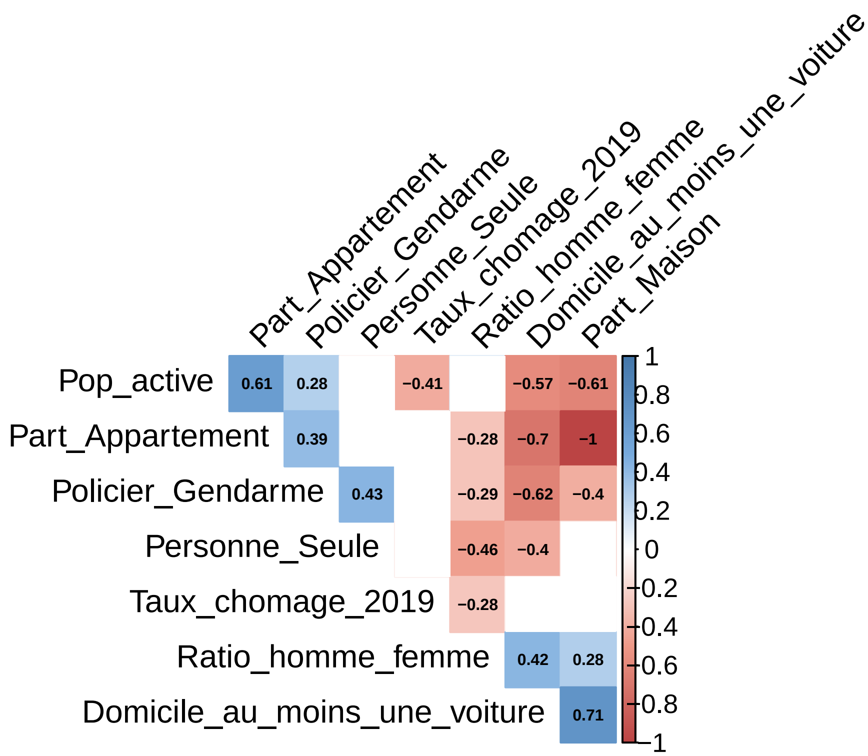
2.3.1 Matrice de corrélation des variables explicatives

```r
cor.mtest <- function(mat, ...) {
  mat <- as.matrix(mat)
  n <- ncol(mat)
  p.mat<- matrix(NA, n, n)
  diag(p.mat) <- 0
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      tmp <- cor.test(mat[, i], mat[, j], ...)
      p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
    }
  }
  colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
  p.mat
}

# Matrice de p-value de la corrélation
p.mat <- cor.mtest(base[,c(-1, -2)])

# Affichage pour sortir
stargazer::stargazer(p.mat, type = "text", title = "Matrice de corrélation")
```

```
Matrice de corrélation
====================================================================================================
                              Taux_chomage_2019 Domicile_au_moins_une_voiture Pop_active Ratio_homme_fem
----------------------------------------------------------------------------------------------------
Taux_chomage_2019                     0                     0.416                 0.00004        0.007
Domicile_au_moins_une_voiture       0.416                     0                      0          0.00002
Pop_active                         0.00004                    0                      0           0.310
Ratio_homme_femme                   0.007                  0.00002                 0.310           0
Part_Maison                         0.934                     0                      0           0.005
Part_Appartement                    0.943                     0                      0           0.005
Policier_Gendarme                   0.897                     0                    0.007         0.004
Personne_Seule                      0.470                  0.00004                 0.663        0.00000
----------------------------------------------------------------------------------------------------
```

```r
M <- cor(base[,c(-1, -2)])
my_colors <- brewer.pal(5, "Spectral")
my_colors <- colorRampPalette(my_colors)(100)
ord <- order(M[1, ])
M_ord <- M[ord, ord]
#ellipse::plotcorr(M_ord , col=my_colors[M_ord*50+50] , mar=c(1,1,1,1))

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot::corrplot(M, method="color", col=col(200),
         type="upper", order="hclust",
         addCoef.col = "black", # Ajout du coefficient de corrélation
         tl.col="black", tl.srt=45, #Rotation des etiquettes de textes
         # Combiner avec le niveau de significativité
         p.mat = p.mat, sig.level = 0.05, insig = "blank",
         # Cacher les coefficients de corrélation sur la diagonale
```

```
        diag=FALSE, number.cex = 0.5
)
```




```
rm(my_colors, ord, p.mat, col, cor.mtest, M_ord)
```

2.3.2 Choix des variables explicatives pour la suite

Le choix des variables explicatives pour la suite de notre variable s'est faite en nous basant sur la littérature (notamment les autres études faites sur la criminalité en France), combiné à l'importance des variables pour un modèle de randomForest. L'objectif n'étant de tester ou d'apprendre des techniques de sélection de variable, les résultats de variable importance du randomForest n'est pas présenté dans le présente document.

```
carte@data <- carte@data%>%
  dplyr::select(c("CC_2", "Cambriolage","NAME_1", "NAME_2", "Taux_chomage_2019", "Part_Maison",
                "Policier_Gendarme", "Personne_Seule"))
```

3 Définition des matrices de distance

3.1 Définition de la matrice des poids : Relations basées sur la connectivité :

On récupère les coordonnées spatiales. Pour les polygones spatiaux, il renvoie les centroïdes des zones d'emploi.

```
coord <-coordinates(carte)
```

3.1.1 Définition de la matrice des poids - Connectivité : Alternative Contiguity matrix

```
library(spdep)
```

```
Le chargement a nécessité le package : spData
```

```
To access larger datasets in this package, install the spDataLarge package with: `install.packages('spDa
```

```
Le chargement a nécessité le package : sf
```

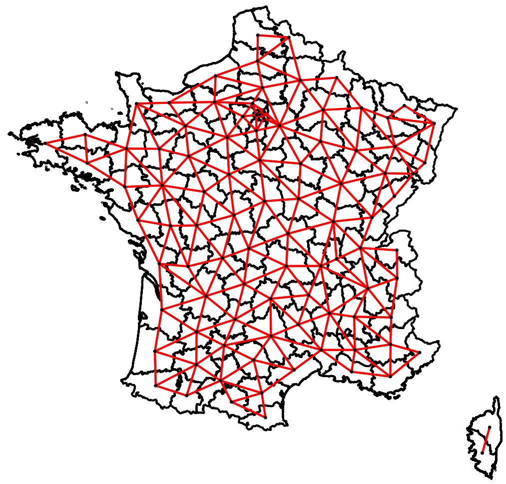
```
Linking to GEOS 3.9.1, GDAL 3.4.3, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
carte.nb <- poly2nb(carte)
summary(carte.nb)
```

```
Neighbour list object:
Number of regions: 96
Number of nonzero links: 478
Percentage nonzero weights: 5.186632
Average number of links: 4.979167
Link number distribution:

 1  2  3  4  5  6  7  8 10
 2  6 10 16 19 30 11  1  1
2 least connected regions:
67 68 with 1 link
1 most connected region:
87 with 10 links
```

On remarque que les centroïdes ont en moyenne 4.979167 liens. Affichage des liens :

```
plot(carte)
plot(carte.nb, col="red",cex=0.1,coord,add=TRUE)
```



```
cont.w <- nb2listw(carte.nb,style="W")
cont.w
```

```
Characteristics of weights list object:
Neighbour list object:
Number of regions: 96
Number of nonzero links: 478
Percentage nonzero weights: 5.186632
Average number of links: 4.979167

Weights style: W
Weights constants summary:
   n   nn S0       S1      S2
W 96 9216 96 44.07222 391.414
```

La fonction nb2listw complète la liste de voisins avec des poids spatiaux pour le schéma de codage choisi. La fonction d'aide can.be.simmed vérifie si un objet de poids spatial est similaire à symétrique et peut être transformé de cette façon pour produire des valeurs propres réelles ou pour la décomposition de Cholesky. On regarde que les villes ont en moyenne 5 voisins.

3.1.2 Définition de la matrice des poids - Connectivité : Alternative Closest neighbors

```
### Definition of the 5 closest neighbors
cartePPV5.knn<-knearneigh(coord,k=5)
cartePPV5.nb<- knn2nb(cartePPV5.knn)
# Matrice de poids
PPV5.w<-nb2listw(cartePPV5.nb,style="W")
```

3.2 Définition de la matrice des poids - Euclidean distance

Relation basée sur l'inverse de la distance. Prendre l'inverse de la distance est la manière la plus naturelle de modéliser une influence décroissante avec la distance.

Ici, nous travaillons avec des département (96 au total), donc les distances entre départements sont plus faibles que les distances entre communes (vu en cours). Malgré ces petites valeurs, il faudra également divisée par 1000 pour avoir les distances en km et ainsi opéré un choix pour lambda en nous basant sur le graphique de la page 47 du chapitre 1.

```
distance<- fields::rdist(coord,coord)/1000 #for a distance in km
# La règle des distances notifie que la diagonale doit être égale à 0
diag(distance)<-0
```

–> Choix du seuil et Choix de lambda

Pour le choix du seuil de dichotomisation, on a choisit d'évaluer sur la matrice des distances, la distance moyenne qui sépare les département entre eux. Pour avoir une répartition pas trop déséquilibrée, on a donc pris la médiane des moyennes. Par exemple, pour le département d'Ille-et-Vilaine, on a pris la distance moyenne entre ledit département et tous les autres. Ceci répété pour tous les départements, ensuite on a pris la médiane des moyennes.

Par ailleurs, pour le choix de lambda, on a opté pour lambda = 1. Ainsi, on accorde plus de poids aux variables qui sont spatialement plus proches (plus petite distance) afin de laisser moins de poids aux variables plus éloignées.

```
# Choix du seuil
seuil <- median(apply(distance, MARGIN = 2, FUN = mean))

# Dichotomisation
distance[distance>= seuil ]<-0

#gamma =1
dist1<- ifelse(distance>0, 1.0/distance,0)

# Matrice de poids
dist.w1<-mat2listw(dist1, row.names = NULL, style="W")
rm(seuil, distance, dist1)
```

Ainsi, nous avons les 3 matrices de poids :

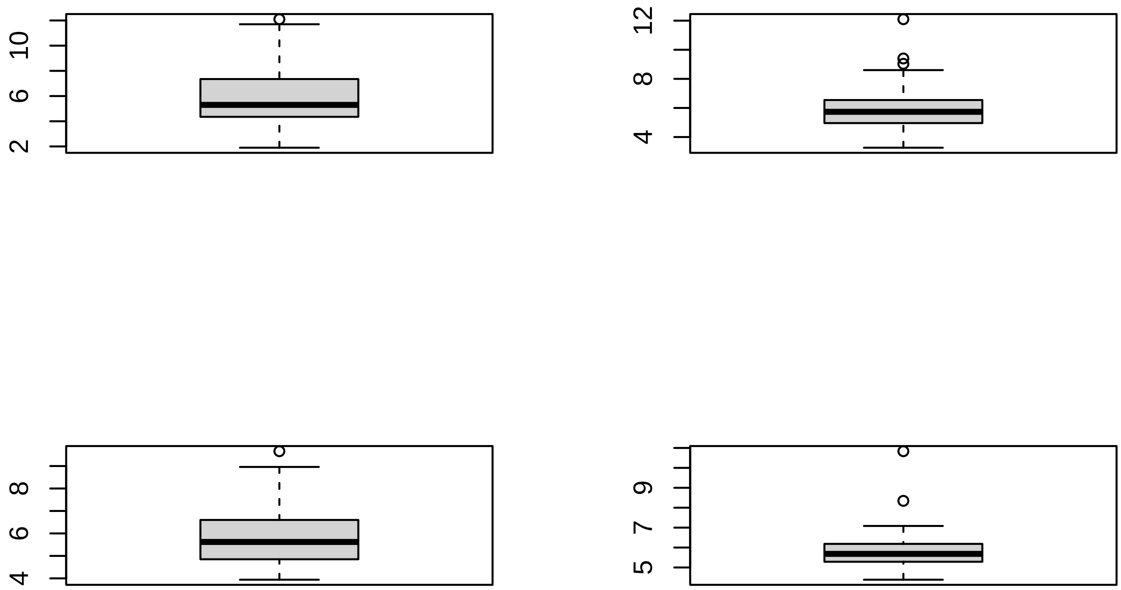
- Matrice de contiguïte (connectivity) : cont.w
- Critère des plus proche voisins (connectivity) : PPV5.w
- Inverse de la distance (lambda = 1) : dist.w1

3.3 Comparaison des statistiques pour observer l'influence du choix de la matrice de poids sur les résultats

On représente la boite à moustache du cambriolage en plus de celui de cambriolage par différente matrice afin d'avoir un aperçu visuel de l'effet de la matrice de poids sur le cambriolage

```
cambriolage <- cbind(base$Cambriolage)
cont.w_cambriolage <- lag.listw(cont.w, cambriolage)
ppv5.w_cambriolage <- lag.listw(PPV5.w, cambriolage)
dist.w1_cambriolage <- lag.listw(dist.w1, cambriolage)
par(mfrow = c(2,2))
graphics::boxplot.matrix(cambriolage)
graphics::boxplot.matrix(cont.w_cambriolage)
```

```r
graphics::boxplot.matrix(ppv5.w_cambriolage)
graphics::boxplot.matrix(dist.w1_cambriolage)
```




```r
rm(cambriolage, cont.w_cambriolage, ppv5.w_cambriolage, dist.w1_cambriolage)
```

4 Analyse de l’autocorrélation Spatiale

4.1 Analyse de l’autocorrélation Spatiale Globale : Indice de Moran

Les indices d’autocorrélation spatiale sont utilisés pour caractériser la corrélation entre les mesures qui sont géographiquement similaires à un phénomène mesuré. Si WY est le vecteur des moyennes de la variable Y (où W est la matrice des poids spatiaux) au voisinage de chaque unité spatiale, les indices d’autocorrélation spatiale se présentent comme suit : (confère cours p.9 chapter 2).

- H0 : no spatial autocorrelation

Pour eviter les problèmes liés à la dimension, nous allons centrés la variable cible.

```r
carte$Cambriolage_centered <- carte$Cambriolage - mean(carte$Cambriolage)
```

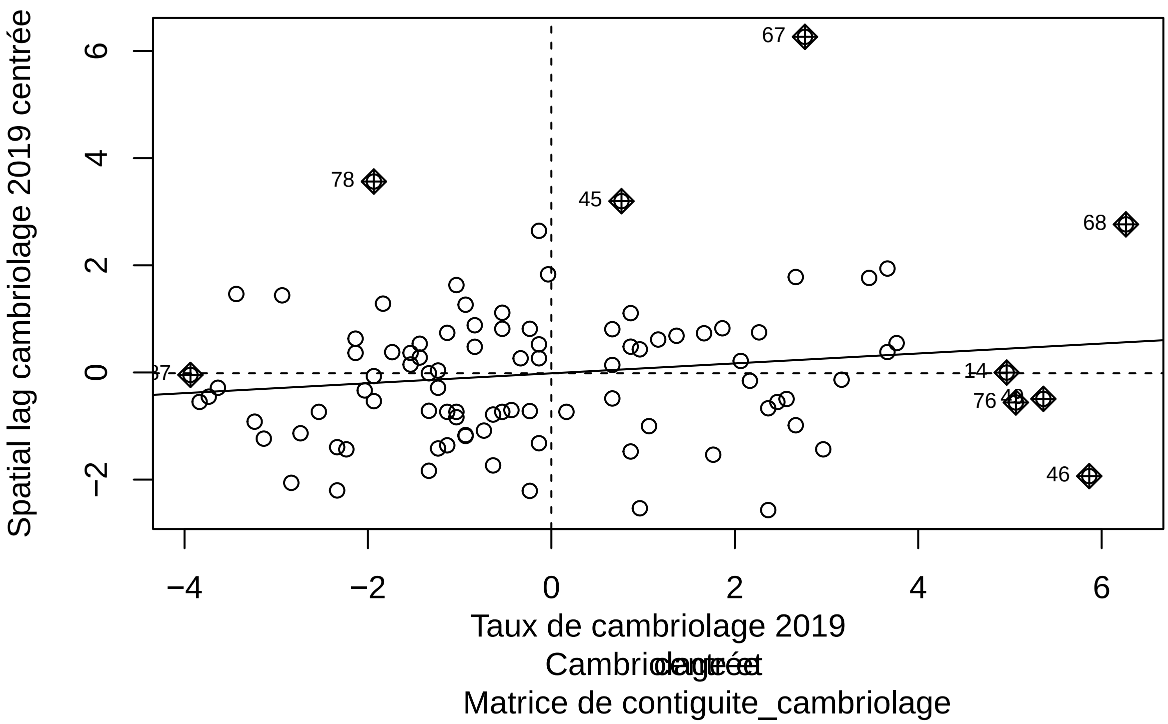
4.1.1 Matrice de contiguïte

```r
MoranTest_Cont_W <- moran.test(carte$Cambriolage_centered, cont.w, randomisation = FALSE,
                               alternative = "greater")

MoranTest_Cont_W
```

4.1.1.1 Autocorrelation Spatiale Globale

```
    Moran I test under normality

data:  carte$Cambriolage_centered
weights: cont.w

Moran I statistic standard deviate = 1.5276, p-value = 0.06331
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.092570053      -0.010526316       0.004554958
```

```r
moran.plot(x = carte$Cambriolage_centered, listw = cont.w, xlab = "Taux de cambriolage 2019
           centrée", ylab = "Spatial lag cambriolage 2019 centrée", sub = "Cambriolage et
           Matrice de contiguite_cambriolage",
           labels = as.character(row.names(carte)))
```



- Interprétation :

Moran I statistic standard deviate = 1.5276 > 0 ==> Possible Positive spatial autocorrelation p-value = 0.06331 > 0.05 ==> On ne rejette pas HO (No Spatial Correlation)

```r
locm_w <-localmoran(carte$Cambriolage_centered, cont.w)
head(locm_w)
```

4.1.1.2 Autocorrelaytion Spatiale locale (LISA)

```
           Ii          E.Ii     Var.Ii       Z.Ii Pr(z != E(Ii))
1 -0.30150224 -0.011221677 0.16808882 -0.7080249      0.4789298
5  0.13646569 -0.002722010 0.04112331  0.6863677      0.4924813
6 -0.07627672 -0.001400731 0.01795866 -0.5587346      0.5763429
7 -0.14037003 -0.001400731 0.02118986 -0.9546733      0.3397429
8 -0.95932469 -0.023686852 0.42512082 -1.4349990      0.1512874
9 -0.06414222 -0.009403665 0.17124178 -0.1322783      0.8947642
```

```
summary(locm_w)
```

```
       Ii                E.Ii                Var.Ii              Z.Ii          Pr(z != E(Ii))
 Min.   :-2.16413   Min.   :-7.876e-02   Min.   :0.000038   Min.   :-2.7974   Min.   :0.005152
 1st Qu.:-0.12911   1st Qu.:-1.346e-02   1st Qu.:0.025591   1st Qu.:-0.5891   1st Qu.:0.199421
 Median : 0.03409   Median :-4.433e-03   Median :0.094415   Median : 0.1971   Median :0.415789
 Mean   : 0.09257   Mean   :-1.053e-02   Mean   :0.281479   Mean   : 0.1448   Mean   :0.430990
 3rd Qu.: 0.21659   3rd Qu.:-1.500e-03   3rd Qu.:0.295164   3rd Qu.: 0.8613   3rd Qu.:0.609910
 Max.   : 3.30213   Max.   :-2.520e-06   Max.   :6.965858   Max.   : 2.7550   Max.   :0.978165
```

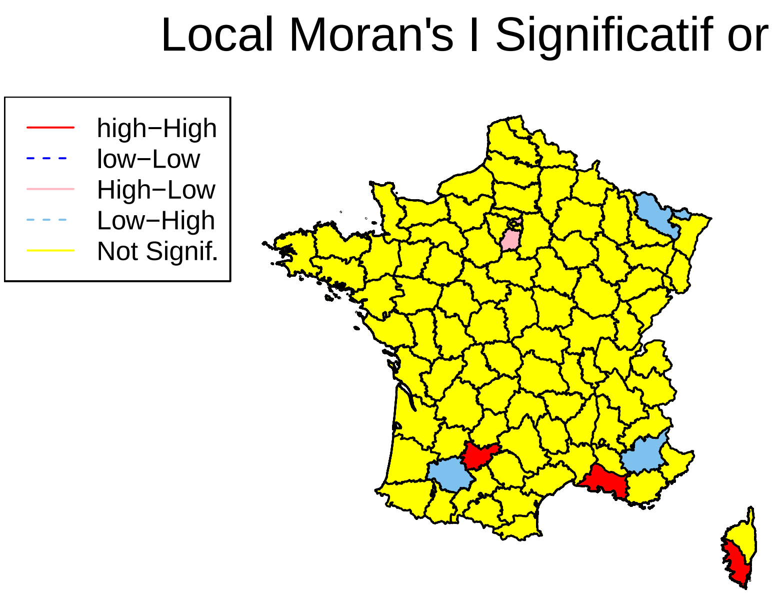
```
carte$ContW_cambriolage <- lag.listw(cont.w, carte$Cambriolage_centered)
#create a new variable for quadrants
carte$quad_sig <- NA

#HH
carte@data[(carte$Cambriolage_centered >= 0 & carte$ContW_cambriolage >= 0) &
             (locm_w[,5] <= 0.05), "quad_sig"] <- 1.0

#LL
carte@data[(carte$Cambriolage_centered <= 0 & carte$ContW_cambriolage <= 0) &
             (locm_w[,5] <= 0.05), "quad_sig"] <- 2.0

#HL
carte@data[(carte$Cambriolage_centered >= 0 & carte$ContW_cambriolage <= 0) &
             (locm_w[,5] <= 0.05), "quad_sig"] <- 3.0

#LH
carte@data[(carte$Cambriolage_centered <= 0 & carte$ContW_cambriolage >= 0) &
             (locm_w[,5] <= 0.05), "quad_sig"] <- 4.0

#pas significatifs
carte@data[(locm_w[,5] > 0.05), "quad_sig"] <- 5.0

table(carte$quad_sig)
```

4.1.1.3 Cartes des LISAs prenant en compte la significativité

```
 1  3  4  5
 3  2  3 88
```

```
labels <- c("high-High", "low-Low", "High-Low", "Low-High", "Not Signif.")

# need for the map
breaks <- seq(1, 5, 1)
```

```r
carte$np <- findInterval(carte$quad_sig, breaks)

col.map <- c("red",  "blue","lightpink", "skyblue2", "yellow")
plot(carte, col = col.map[carte$np])
mtext("Local Moran's I Significatif or Not", cex = 1.5, side = 3, line = 1)
legend("topleft", legend = labels, col=c("red",  "blue","lightpink", "skyblue2","yellow"),
       lty=1:2, cex=0.8)
```



4.1.2 Matrice des plus proches voisins

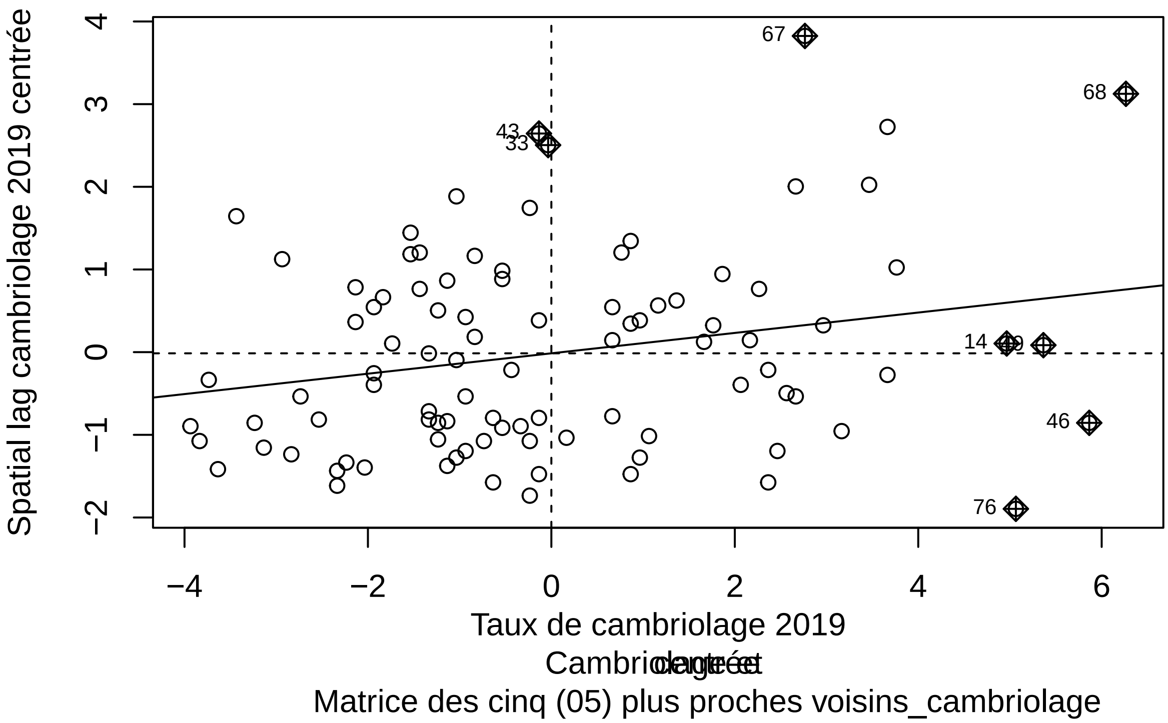
```r
MoranTest_ppv5.w <- moran.test(carte$Cambriolage_centered, PPV5.w, randomisation = FALSE,
                               alternative = "greater")

MoranTest_ppv5.w
```

4.1.2.1 Indice de Moran

```
    Moran I test under normality

data:  carte$Cambriolage_centered
weights: PPV5.w

Moran I statistic standard deviate = 2.2755, p-value = 0.01144
alternative hypothesis: greater
```

```
sample estimates:
Moran I statistic       Expectation          Variance
       0.12326334       -0.01052632        0.00345693
```

```
moran.plot(x = carte$Cambriolage_centered, listw = PPV5.w, xlab = "Taux de cambriolage 2019
           centrée", ylab = "Spatial lag cambriolage 2019 centrée", sub = "Cambriolage et
           Matrice des cinq (05) plus proches voisins_cambriolage",
           labels = as.character(row.names(carte)))
```



- Interprétation :

Moran I statistic standard deviate = 2.2755 > 0 ==> Positive spatial autocorrelation p-value = 0.01144 < 0.05 ==> On rejette HO (Spatial Autocorrelation)

```
locm_ppv <-localmoran(carte$Cambriolage_centered, PPV5.w)
head(locm_ppv)
```

4.1.2.2 Autocorrelaytion Spatiale locale (LISA)

```
           Ii          E.Ii      Var.Ii        Z.Ii Pr(z != E(Ii))
1 -0.09711941 -0.011221677 0.20397295 -0.1901934      0.8491576
2  0.12536339 -0.002722010 0.04990244  0.5733750      0.5663908
3 -0.02940138 -0.001400731 0.02571354 -0.1746173      0.8613804
4 -0.18550082 -0.001400731 0.02571354 -1.1480823      0.2509346
5 -1.07722747 -0.023686852 0.42512082 -1.6158279      0.1061315
6  0.05967118 -0.009403665 0.17124178  0.1669226      0.8674310
```

```r
summary(locm_ppv)
```

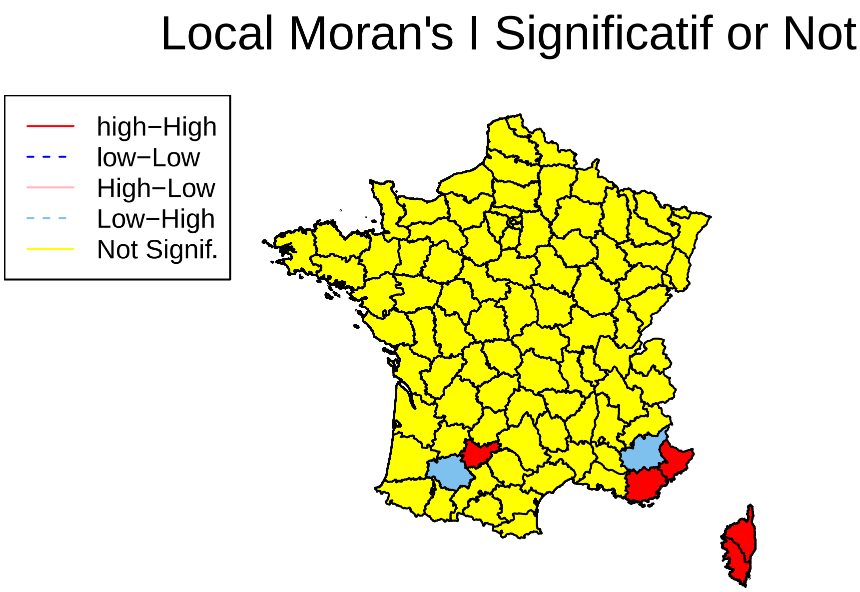
```
       Ii                E.Ii                Var.Ii               Z.Ii          Pr(z != E(Ii))
 Min.   :-1.83029   Min.   :-7.876e-02   Min.   :0.0000463   Min.   :-2.6237   Min.   :0.0001157
 1st Qu.:-0.15024   1st Qu.:-1.346e-02   1st Qu.:0.0275376   1st Qu.:-0.7491   1st Qu.:0.1955577
 Median : 0.05847   Median :-4.433e-03   Median :0.0811361   Median : 0.3465   Median :0.3721853
 Mean   : 0.12326   Mean   :-1.053e-02   Mean   :0.1875115   Mean   : 0.2163   Mean   :0.4111885
 3rd Qu.: 0.22596   3rd Qu.:-1.500e-03   3rd Qu.:0.2441476   3rd Qu.: 1.0641   3rd Qu.:0.6287198
 Max.   : 3.73213   Max.   :-2.520e-06   Max.   :1.3338876   Max.   : 3.8550   Max.   :0.9766180
```

```r
carte$ppvW_cambriolage <- lag.listw(PPV5.w, carte$Cambriolage_centered)
#create a new variable for quadrants
base$quad_sig_ppv <- NA

#HH
carte@data[(carte$Cambriolage_centered >= 0 & carte$ppvW_cambriolage >= 0) &
             (locm_ppv[,5] <= 0.05), "quad_sig_ppv"] <- 1.0

#LL
carte@data[(carte$Cambriolage_centered <= 0 & carte$ppvW_cambriolage <= 0) &
             (locm_ppv[,5] <= 0.05), "quad_sig_ppv"] <- 2.0

#HL
carte@data[(carte$Cambriolage_centered >= 0 & carte$ppvW_cambriolage <= 0) &
             (locm_ppv[,5] <= 0.05), "quad_sig_ppv"] <- 3.0

#LH
carte@data[(carte$Cambriolage_centered <= 0 & carte$ppvW_cambriolage >= 0) &
             (locm_ppv[,5] <= 0.05), "quad_sig_ppv"] <- 4.0

#pas significatifs
carte@data[(locm_ppv[,5] > 0.05), "quad_sig_ppv"] <- 5.0

table(carte$quad_sig_ppv)
```

4.1.2.3 Maps de Lisa

```
 1  4  5
 5  2 89
```

```r
labels <- c("high-High", "low-Low", "High-Low", "Low-High", "Not Signif.")

# need for the map
breaks <- seq(1, 5, 1)

carte$np_1 <- findInterval(carte$quad_sig_ppv, breaks)

col.map <- c("red",  "blue","lightpink", "skyblue2", "yellow")
plot(carte, col = col.map[carte$np_1])
mtext("Local Moran's I Significatif or Not", cex = 1.5, side = 3, line = 1)
legend("topleft", legend = labels, col=c("red",  "blue","lightpink", "skyblue2","yellow"),
       lty=1:2, cex=0.8)
```



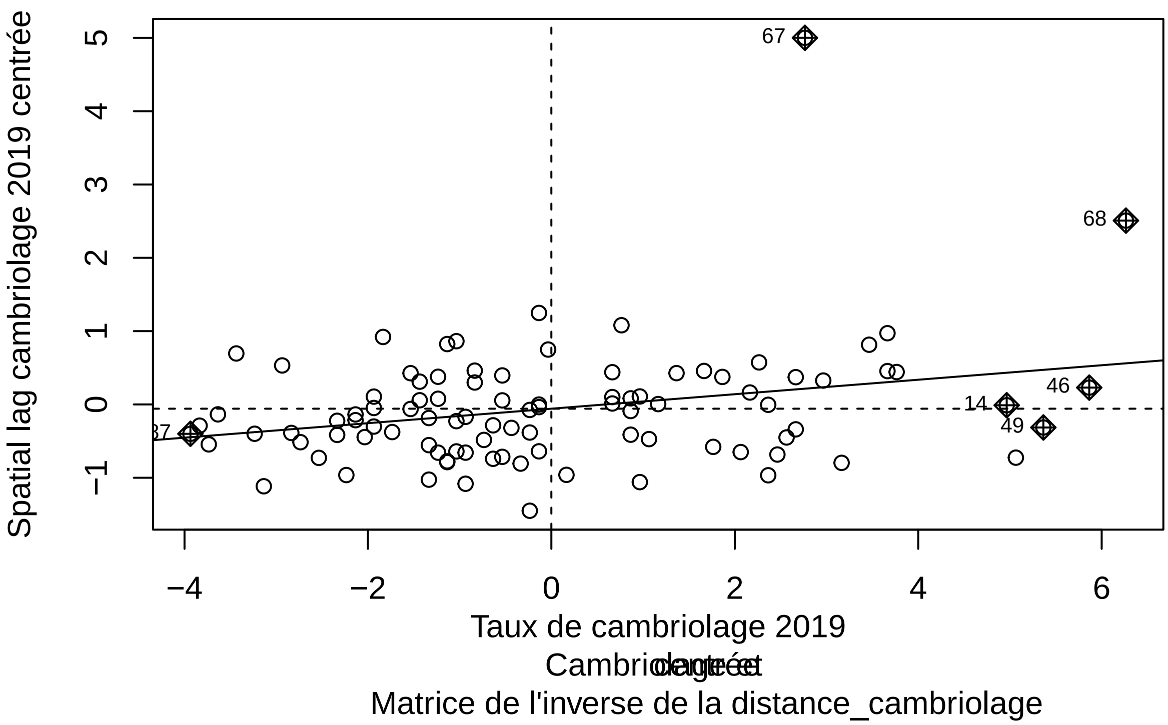
4.1.3 L'inverse de la distance

```
MoranTest_dist.w1 <- moran.test(carte$Cambriolage_centered, dist.w1, randomisation = FALSE,
                                alternative = "greater")

MoranTest_dist.w1
```

4.1.3.1 Indice de Moran

```
    Moran I test under normality

data:  carte$Cambriolage_centered
weights: dist.w1

Moran I statistic standard deviate = 4.3791, p-value = 5.96e-06
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.0987729599     -0.0105263158      0.0006229771
```

```
moran.plot(x = carte$Cambriolage_centered, listw = dist.w1, xlab = "Taux de cambriolage 2019
           centrée", ylab = "Spatial lag cambriolage 2019 centrée", sub = "Cambriolage et
           Matrice de l'inverse de la distance_cambriolage",labels = as.character(row.names(carte)))
```



- Interprétation :

Moran I statistic standard deviate = 4> 0 ==> Possible Positive spatial autocorrelation p-value = 6e-06 < 0.05 ==> On rejette HO (Spatial Autocorrelation)

```
locm_dist <-localmoran(carte$Cambriolage_centered, dist.w1)
head(locm_dist)
```

4.1.3.2 Autocorrélation Spatial Local (LISA)

```
            Ii         E.Ii      Var.Ii        Z.Ii Pr(z != E(Ii))
1 -0.002251716 -0.011221677 0.015063191  0.07308564     0.94173797
2  0.001073872 -0.002722010 0.001712024  0.09173979     0.92690478
3 -0.073446969 -0.001400731 0.002213230 -1.53143325     0.12566235
4 -0.047639575 -0.001400731 0.001324925 -1.27031484     0.20397250
5 -0.455431872 -0.023686852 0.045847198 -2.01637329     0.04376096
6  0.067152881 -0.009403665 0.010304398  0.75417307     0.45074530
```

```
summary(locm_dist)
```

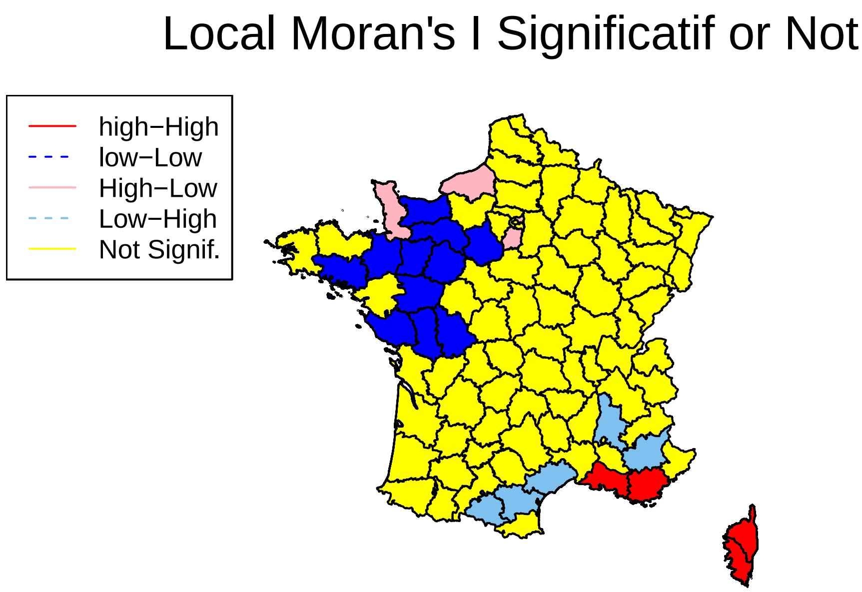
```
       Ii                E.Ii                 Var.Ii               Z.Ii          Pr(z != E(Ii))
 Min.   :-0.70106   Min.   :-7.876e-02   Min.   :0.0000071   Min.   :-3.0021   Min.   :0.0003798
 1st Qu.:-0.03402   1st Qu.:-1.346e-02   1st Qu.:0.0017767   1st Qu.:-1.1507   1st Qu.:0.0550597
 Median : 0.03973   Median :-4.433e-03   Median :0.0076849   Median : 0.7165   Median :0.1690172
 Mean   : 0.09877   Mean   :-1.053e-02   Mean   :0.0541316   Mean   : 0.4017   Mean   :0.2828768
 3rd Qu.: 0.16853   3rd Qu.:-1.500e-03   3rd Qu.:0.0330086   3rd Qu.: 1.5842   3rd Qu.:0.4168653
 Max.   : 2.99508   Max.   :-2.520e-06   Max.   :2.2603645   Max.   : 3.5537   Max.   :0.9915076
```

```r
carte$distw1_cambriolage <- lag.listw(dist.w1, carte$Cambriolage_centered)
#create a new variable for quadrants
carte$quad_sig_dist <- NA

#HH
carte@data[(carte$Cambriolage_centered >= 0 & carte$distw1_cambriolage >= 0) &
             (locm_dist[,5] <= 0.05), "quad_sig_dist"] <- 1.0

#LL
carte@data[(carte$Cambriolage_centered <= 0 & carte$distw1_cambriolage <= 0) &
             (locm_dist[,5] <= 0.05), "quad_sig_dist"] <- 2.0

#HL
carte@data[(carte$Cambriolage_centered >= 0 & carte$distw1_cambriolage <= 0) &
             (locm_dist[,5] <= 0.05), "quad_sig_dist"] <- 3.0

#LH
carte@data[(carte$Cambriolage_centered <= 0 & carte$distw1_cambriolage >= 0) &
             (locm_dist[,5] <= 0.05), "quad_sig_dist"] <- 4.0

#pas locm_dist
carte@data[(locm_dist[,5] > 0.05), "quad_sig_dist"] <- 5.0

table(carte$quad_sig_dist)
```

4.1.3.3 Maps de Lisa

```
 1  2  3  4  5
 4 11  4  5 72
```

```r
labels <- c("high-High", "low-Low", "High-Low", "Low-High", "Not Signif.")

# need for the map

breaks <- seq(1, 5, 1)

carte$np_2 <- findInterval(carte$quad_sig_dist, breaks)

col.map <- c("red",  "blue","lightpink", "skyblue2", "yellow")
plot(carte, col = col.map[carte$np_2])
mtext("Local Moran's I Significatif or Not", cex = 1.5, side = 3, line = 1)
legend("topleft", legend = labels, col=c("red",  "blue","lightpink", "skyblue2","yellow"),
       lty=1:2, cex=0.8)
```



- Résumé des matrices

1. Matrice de contiguïté : No Spatial Autocorrelation ;
2. Matrice des plus proche voisin : Spatial Autocorrelation ;
3. Matrice de l'inverse de la distance : Spatial Autocorrelation

5 Estimation des modèles via la stratégie de test Top-Down

5.1 Estimation des modèles

Pour la suite, on utilisera la matrice de l'inverse de la distance. Elle contient le plus d'indice significatif parmi les trois. Une comparaison sera faite avec les résultats avec la matrice de contiguïté pour observer les éventuelles différences.

```
library(spatialreg)
```

```
Warning: le package 'spatialreg' a été compilé avec la version R 4.2.2
```

```
Le chargement a nécessité le package : Matrix
```

```
Attachement du package : 'spatialreg'
```

```
Les objets suivants sont masqués depuis 'package:spdep':

    get.ClusterOption, get.coresOption, get.mcOption, get.VerboseOption, get.ZeroPolicyOption, set.Clus
    set.ZeroPolicyOption
```

```
formule <- Cambriolage ~ Taux_chomage_2019 + Policier_Gendarme + Part_Maison + Personne_Seule
```

5.1.1 Modèle OLS

```
Criminality_OLS <- lm(formula = formule, data = carte)
```

5.1.2 Modèle SLX

```
carte$Taux_chomage_2019_M <- carte$Taux_chomage_2019 -  mean(carte$Taux_chomage_2019)
carte$W_Taux_chomage_2019_M <- lag.listw(dist.w1, carte$Taux_chomage_2019_M)

carte$Policier_Gendarme_M <- carte$Policier_Gendarme - mean(carte$Policier_Gendarme)
carte$W_Policier_Gendarme_M <- lag.listw(dist.w1, carte$Policier_Gendarme_M)

carte$Part_Maison_M <- carte$Part_Maison - mean(carte$Part_Maison)
carte$W_Part_Maison_M <- lag.listw(dist.w1, carte$Part_Maison_M)

carte$Personne_Seule_M <- carte$Personne_Seule - mean(carte$Personne_Seule)
carte$w_Personne_Seule_M <- lag.listw(dist.w1, carte$Personne_Seule_M)

Criminality_SLX <- lm(Cambriolage_centered ~ Taux_chomage_2019_M + Policier_Gendarme_M +
                       Part_Maison_M + Personne_Seule_M + W_Taux_chomage_2019_M +
                        W_Policier_Gendarme_M + W_Part_Maison_M + w_Personne_Seule_M,data = carte)
```

5.1.3 Modèle SAR

```
Criminality_SAR <- lagsarlm(formula = formule, listw = dist.w1, data = carte)
```

5.1.4 Modèle SEM

```
Criminality_SEM = errorsarlm(formula = formule, listw=dist.w1,  data=carte)
```

5.1.5 Modèle SDM

```
Criminality_SDM <- lagsarlm(formula = formule, listw = dist.w1, type = "mixed", data = carte)
```

5.2 Strategy of test Top Down approach LeSage & Pace (2009)

- SDM VS SAR : On test

H0 : $\delta = 0$, If H0 rejected = SDM

```
sdm_vs_sar <- LR.Sarlm(Criminality_SDM,Criminality_SAR)
print(sdm_vs_sar)
```

```

    Likelihood ratio for spatial linear models

data:
Likelihood ratio = 2.6416, df = 4, p-value = 0.6195
sample estimates:
Log likelihood of Criminality_SDM Log likelihood of Criminality_SAR
                        -190.3348                         -191.6556
```

On a $p - value = 0.6195 > 0.05$ => On ne rejette pas HO donc $\delta = 0$

- sar_vs_ols

```
sar_vs_ols <- LR.Sarlm(Criminality_SAR,Criminality_OLS)
print(sar_vs_ols)
```

```

    Likelihood ratio for spatial linear models

data:
Likelihood ratio = 0.70297, df = 1, p-value = 0.4018
sample estimates:
Log likelihood of Criminality_SAR Log likelihood of Criminality_OLS
                        -191.6556                         -192.0071
```

$p - value = 0.4018$, on ne rejette pas nulle HO on retient donc le Modele OLS

- SDM vs SLX

H0 : rho=0, teta not = 0 et teta + rho*beta != 0 unconstrained model : SDM constrainted model : SLX If H0 reject : SDM

```
sdm_vs_slx <- LR.Sarlm(Criminality_SDM, Criminality_SLX)
print(sdm_vs_slx)
```

```

    Likelihood ratio for spatial linear models

data:
Likelihood ratio = 0.024426, df = 1, p-value = 0.8758
sample estimates:
Log likelihood of Criminality_SDM Log likelihood of Criminality_SLX
                        -190.3348                         -190.3470
```

$p - value = O.8759 > 5\%$, on accepte HO

- slx_vs_ols

```
slx_vs_ols<-LR.Sarlm(Criminality_SLX,Criminality_OLS)
print(slx_vs_ols)
```

```

    Likelihood ratio for spatial linear models

data:
Likelihood ratio = 3.3201, df = 4, p-value = 0.5057
sample estimates:
Log likelihood of Criminality_SLX Log likelihood of Criminality_OLS
                        -190.3470                         -192.0071
```

$p - value = 0.5058 > 5\%$ ==> p = 0 ==> Modele OLS

- sdm_vs_sem

```
sdm_vs_sem <-LR.Sarlm(Criminality_SDM,Criminality_SEM)
print(sdm_vs_sem)
```

```

    Likelihood ratio for spatial linear models
```

```
data:
Likelihood ratio = 3.3142, df = 4, p-value = 0.5067
sample estimates:
Log likelihood of Criminality_SDM Log likelihood of Criminality_SEM
                        -190.3348                         -191.9919
```

$p - value = 0.5067 > 0.05$, on accepte HO

```
sem_vs_ols<-LR.Sarlm(Criminality_SEM,Criminality_OLS)
print(sem_vs_ols)
```

```

    Likelihood ratio for spatial linear models

data:
Likelihood ratio = 0.030321, df = 1, p-value = 0.8618
sample estimates:
Log likelihood of Criminality_SEM Log likelihood of Criminality_OLS
                        -191.9919                         -192.0071
```

$p - value = 0.8618$, on chosit donc le modèle OLS

5.3 Visualisation du modèle

```
summary(Criminality_OLS)
```

```

Call:
lm(formula = formule, data = carte)

Residuals:
    Min      1Q  Median      3Q     Max
-5.1055 -1.0781 -0.2373  1.1486  5.6935

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       12.025975   2.719695   4.422  2.7e-05 ***
Taux_chomage_2019  0.470191   0.119143   3.946 0.000156 ***
Policier_Gendarme -0.184602   0.177755  -1.039 0.301780
Part_Maison       -0.071505   0.011779  -6.070  2.9e-08 ***
Personne_Seule    -0.013267   0.006621  -2.004 0.048071 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.837 on 91 degrees of freedom
Multiple R-squared:  0.3904,    Adjusted R-squared:  0.3636
F-statistic: 14.57 on 4 and 91 DF,  p-value: 3.109e-09
```

5.4 Interprétation du modèle

5.5 Test sur le modele

- Si e taux de chômage augmente de 1 point de pourcentage, le taux de cambriolage pour 1000 habitants augmente de 0.47 point de pourcentage ;
- Nombre de policier augmente d'une unité, le taux de cambriolage pour 1000 habitants diminue de 0.18 point de pourcentage ;

- Une augmentation de la part maison entrainant une très petite diminution du taux de cambriolage pour 1000 habitants ;
- Une relation négative entre le nombre de personne seule et le taux de cambriolage pour 1000 habitants, cependant avec un coefficient très petit.

5.5.1 Test de Moran

```
Moran_Res <- lm.morantest(Criminality_OLS, listw = dist.w1, alternative = "two.sided")
Moran_Res
```

```

    Global Moran I for regression residuals

data:
model: lm(formula = formule, data = carte)
weights: dist.w1

Moran I statistic standard deviate = 0.70976, p-value = 0.4779
alternative hypothesis: two.sided
sample estimates:
Observed Moran I      Expectation         Variance
    0.0054479339    -0.0117653576     0.0005881748
```

5.6 Test avec une matrice des cinq plus proche voisins

```
Criminality_SAR_ppv <- lagsarlm(formula = formule, listw = PPV5.w, data = carte)
summary(Criminality_SAR_ppv)
```

```

Call:lagsarlm(formula = formule, data = carte, listw = PPV5.w)

Residuals:
     Min       1Q   Median       3Q      Max
-4.99161 -1.10979 -0.14664  1.03475  5.43651

Type: lag
Coefficients: (asymptotic standard errors)
                    Estimate Std. Error z value  Pr(>|z|)
(Intercept)       11.5971273  2.7851220  4.1640 3.128e-05
Taux_chomage_2019  0.4542496  0.1175188  3.8653 0.0001109
Policier_Gendarme -0.1864471  0.1728009 -1.0790 0.2806009
Part_Maison       -0.0702286  0.0116653 -6.0203 1.741e-09
Personne_Seule    -0.0133422  0.0064552 -2.0669 0.0387431

Rho: 0.087811, LR test value: 0.41772, p-value: 0.51807
Asymptotic standard error: 0.12457
    z-value: 0.70491, p-value: 0.48087
Wald statistic: 0.4969, p-value: 0.48087

Log likelihood: -191.7982 for lag model
ML residual variance (sigma squared): 3.1794, (sigma: 1.7831)
Number of observations: 96
Number of parameters estimated: 7
AIC: 397.6, (AIC for lm: 396.01)
```

```
LM test for residual autocorrelation
test value: 0.05437, p-value: 0.81563
```

On remarque que la significativité des mêmes variables explicatives que pour le modèle OLS avec la matrice de l'inverse de la distance. Celà témoigne donc de la robustesse de notre modèle.

Conclusion

Dans ce projet, nous avons donc expliqué le taux de cambriolage en France pour 1000 habitants.Tout d'abord, nous avons fais une statistique descriptive de nos données. Nous avons construit 3 matrices de poids différentes. On a choisit au final de conserver la matrice de l'inverse de la distance car c'est la plus significative sur les indices (Indice de Moran et LISA). Ensuite, nous avons effectué les modèles économétriques spatiaux, on a donc retenu le modèle de régression linéaire simple. La variable qui explique le mieux le taux de cambriolage en France pour 1000 habitants est le taux de chômage. On conclut donc qu'un département avec un taux de chômage important a d'avantage de cambriolage qu'un département avec un taux de chômage plus faible.