

# Exercices - Économétrie spatiale

Vincent Robitaille

```
library(spdep)
```

```
## Le chargement a nécessité le package : spData
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
## Le chargement a nécessité le package : sf
## Linking to GEOS 3.10.2, GDAL 3.4.1, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.4      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lmtest)
```

```
## Le chargement a nécessité le package : zoo
##
## Attachement du package : 'zoo'
##
## Les objets suivants sont masqués depuis 'package:base':
##
##      as.Date, as.Date.numeric
```

## Chapitre 2

**Question 2.2** *What is the meaning of spatially lagged variable ?* Le lag spatial est similaire au lag d'une série chronologique. Au lieu que la valeur  $y_t$  soit en partie déterminée par les valeurs passées, on parle plutôt de la variable  $y_i$  qui est influencée par les autres valeurs de la variable  $y$ . La valeur  $y$  observée pour un individu est donc influencée par la valeur  $y$  des autres individus avec lesquels il a une *connection* ou dont il est proche.

**Question 2.3** *What is the meaning of row-standardization of weight matrix ? In which case is this operation beneficial ?* La matrice de poids dont les lignes sont standardisées est construite en divisant chaque élément de la ligne par la somme des éléments de cette ligne. Les éléments de la ligne de la nouvelle matrice sommes

alors à zéro. Cette matrice est utile pour calculer les lag spatial en agissant comme une sorte de moyenne pondérée.

```
W21 <- matrix(0, nrow = 8, ncol = 8)
colnames(W21) <- c("R011", "R012", "R021", "R022",
                  "R031", "R032", "R041", "R042")
row.names(W21) <- c("R011", "R012", "R021", "R022",
                  "R031", "R032", "R041", "R042")
W21[1,] <- c(0, 1, 1, 0, 0, 0, 0, 1)
W21[2,] <- c(1, 0, 1, 1, 1, 0, 1, 1)
W21[3,] <- c(1, 1, 0, 1, 0, 0, 0, 0)
W21[4,] <- c(0, 1, 1, 0, 1, 0, 0, 0)
W21[5,] <- c(0, 1, 0, 1, 0, 1, 1, 0)
W21[6,] <- c(0, 0, 0, 0, 1, 0, 0, 0)
W21[7,] <- c(0, 1, 0, 0, 1, 0, 0, 1)
W21[8,] <- c(1, 1, 0, 0, 0, 0, 1, 0)
x <- mat2listw(W21, style = "W")

rom_regions <- c("R011", "R012", "R021", "R022", "R031", "R032", "R041", "R042")
rom_regions <- 1:8
nbrom <- read.gal("romania.GAL", region.id = rom_regions)

wrom <- nbrom |> nb2listw(style = "B")

wrom2 <- nbrom |> nb2listw(style = "W")

m <- wrom |> listw2mat()

m |> as.numeric() |> mean()
```

### Exercice 2.1

```
## [1] 0.40625
```

```
mr <- read.csv("romania_inf_mor_rate.csv", header = FALSE)
```

```
lagged_var <- lag.listw(wrom2, mr$V2)
```

### Exercice 2.4

1. Wales
2. Scotland
3. Northern Ireland
4. North East of England
5. North West of England
6. Yorkshire & Humberside
7. East Midlands
8. West Midlands
9. East Anglia (East of England)
10. Greater London
11. South East England
12. South West England

```
ukgal <- read.gal("UK.GAL", region.id = 1:12)
```

```
ukl <- ukgal |> nb2listw(style = "B")
```

```
ukl |> listw2mat()
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## 1      0      0      1      0      1      0      0      1      0      0      0      1
## 2      0      0      1      1      1      0      0      0      0      0      0      0
## 3      1      1      0      0      0      0      0      0      0      0      0      0
## 4      0      1      0      0      1      1      0      0      0      0      0      0
## 5      1      1      0      1      0      1      1      1      0      0      0      0
## 6      0      0      0      1      1      0      1      0      0      0      0      0
## 7      0      0      0      0      1      1      0      1      1      0      1      0
## 8      1      0      0      0      1      0      1      1      0      0      1      1
## 9      0      0      0      0      0      0      1      0      0      1      1      0
## 10     0      0      0      0      0      0      0      0      1      0      1      0
## 11     0      0      0      0      0      0      1      1      1      1      0      1
## 12     1      0      0      0      0      0      1      1      0      0      1      0
```

*# Exo 2.5*

```
ukdat <- read.csv("data1.csv", dec = ",")
```

```
laguk <- ukgal |> nb2listw(style = "W") |> lag.listw(ukdat$GVA)
```

```
names(laguk) <- c("Wales", "Scotland", "North Ireland", "North East of England",
                  "North West of England", "Yorkshire Humberside",
                  "East Midlands", "West Midlands", "East Anglia",
                  "Greater London", "South East England", "South West England")
```

```
laguk
```

## Exercice 2.5

```
##           Wales           Scotland           North Ireland
##           6.700000           5.000000           5.950000
## North East of England North West of England Yorkshire Humberside
##           8.233333           5.916667           6.300000
##           East Midlands           West Midlands           East Anglia
##           9.420000           8.166667           14.166667
##           Greater London           South East England           South West England
##           11.700000           10.300000           7.950000
```

**Exercice 2.9** Le lien original ne fonctionnait pas, j'ai trouvé ces données comme alternative pour les besoins de l'exercice. <https://public.opendatasoft.com/explore/dataset/us-state-boundaries/table/?location=3,20.11268,7.82227&basemap=jawg.light&dataChart=eyJxdWVyaWVzIjpbeyJjb25maWciOnsiZGF0YXNldCI6ImVzLXN0YXRILWJvdW5kYXJpZXMiLCJvcHRpb25zIjpb7fX0sImNoYXJ0cyI6W3siYWxpZ25Nb250aCI6dHJ1ZSwidHlwZSI6ImNvbHVtbiIsImZ1bmMiOiJBVkcilCJ5QXhpcyI6ImdpZCI6ImNjaWVudGlmaWNEaXNwbGF5Ijpb0cnVlLCJjb2xvciI6ImNlbnRjUxNUEifV0sInhBeGZlIjoibmFtZSI6Im1heHBvaW50cyI6NTAsInNvcnQiOiIifV0sInRpbWVzY2Y2ZSI6ImRpb25zYXlMZWdlbmQiOnRydWU5ImFsaWduTW9udGgiOnRydWV9>

J'ai aussi enlevé les territoires et états américains hors continent car sinon la fonction n'arrive pas à les associer à d'autres états (aucun voisin).

```

us <- read_sf("us-state-boundaries/us-state-boundaries.shp") |>
  filter(!(name %in% c("Guam",
                      "Palau",
                      "Marshall Islands",
                      "Northern Mariana Islands",
                      "Fed States of Micronesia",
                      "Puerto Rico",
                      "Commonwealth of the Northern Mariana Islands",
                      "Hawaii",
                      "Alaska",
                      "United States Virgin Islands",
                      "American Samoa"))))

# names(us)
# plot(us)
gus <- us |>
  ggplot()+
  geom_sf()+
  ggtitle("Carte des états américains (incluant DC) du continent")

contus <- us |> poly2nb(queen = TRUE)

lus <- contus |> nb2listw()
usWmat <- contus |> nb2mat()

```