Exercices - Économétrie spatiale

Vincent Robitaille

```r
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
```

```r
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 error
```

```r
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
```

```r
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(spatialreg)
```

```
## Le chargement a nécessité le package : Matrix
##
## Attachement du package : 'Matrix'
##
## Les objets suivants sont masqués depuis 'package:tidyr':
```

```
##
##     expand, pack, unpack
##
##
## Attachement du package : 'spatialreg'
##
## Les objets suivants sont masqués depuis 'package:spdep':
##
##     get.ClusterOption, get.coresOption, get.mcOption,
##     get.VerboseOption, get.ZeroPolicyOption, set.ClusterOption,
##     set.coresOption, set.mcOption, set.VerboseOption,
##     set.ZeroPolicyOption
```

Chapitre 1

```
df1 <- read.csv("data/data1.csv",dec = ",")
```

Q1

```
fit1.1 <- lm(GVA ~ Labor_prod + Business_br,
             data = df1)
fit1.1 |> summary()
```

```
##
## Call:
## lm(formula = GVA ~ Labor_prod + Business_br, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1398 -0.9172 -0.4388  1.0958  2.5365
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -22.31118    3.38594  -6.589 0.000100 ***
## Labor_prod    0.27750    0.05346   5.191 0.000571 ***
## Business_br   0.42239    0.47243   0.894 0.394567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.791 on 9 degrees of freedom
## Multiple R-squared:  0.9072, Adjusted R-squared:  0.8866
## F-statistic: 43.99 on 2 and 9 DF,  p-value: 2.259e-05
```

Ce modèle est le modèle retenu. Le taux de naissance des entreprises n'est pas significatif dans le premier modèle et le $R\check{s}_a$ du second modèle est sensiblement plus élevé. Ce modèle permet d'expliquer environ 89% des variations observées dans le GVA.

```
fit1.2 <- lm(GVA ~ Labor_prod, data = df1)
fit1.2 |> summary()
```

```
##
## Call:
## lm(formula = GVA ~ Labor_prod, data = df1)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5300 -0.8375 -0.4193  1.0386  3.0149
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -21.38866    3.19237  -6.700 5.36e-05 ***
## Labor_prod    0.31460    0.03335   9.432 2.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.773 on 10 degrees of freedom
## Multiple R-squared:  0.899,  Adjusted R-squared:  0.8889
## F-statistic: 88.97 on 1 and 10 DF,  p-value: 2.708e-06
```

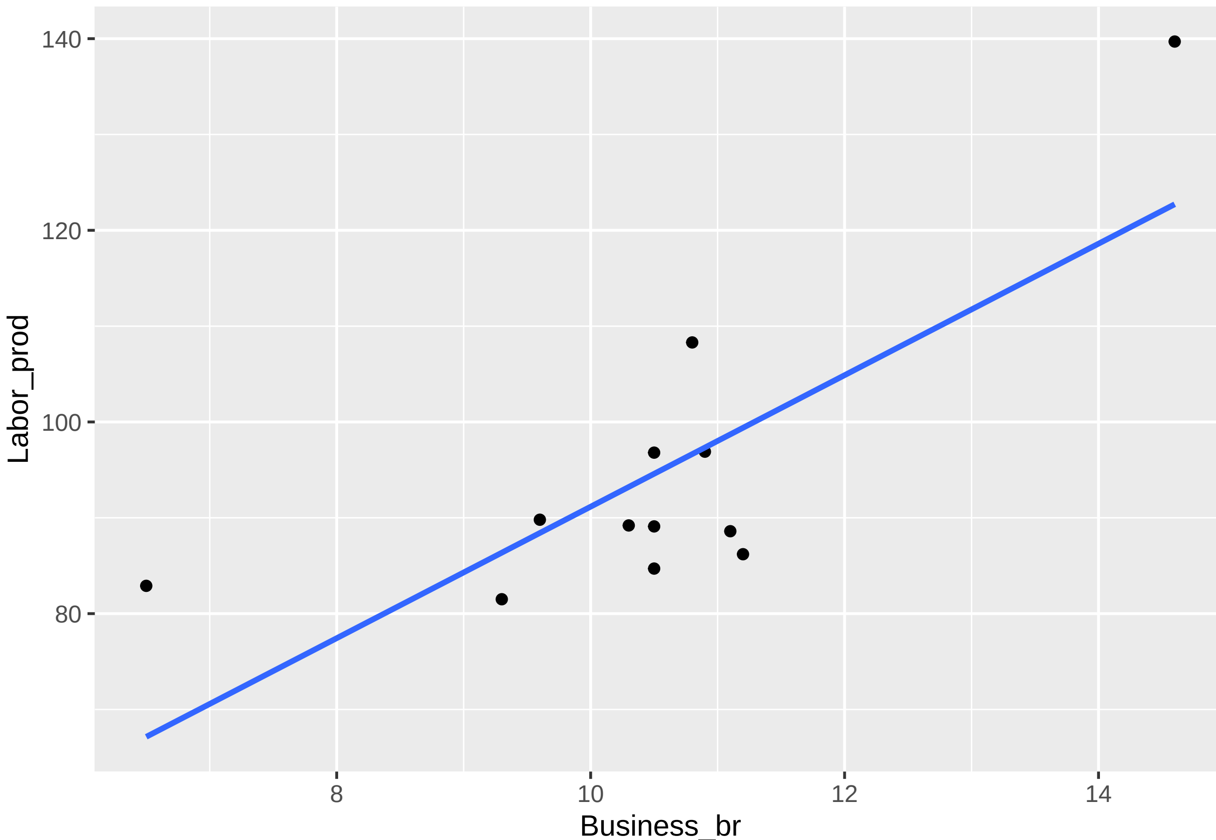
```r
fit1.3 <- lm(GVA ~ Business_br, data = df1)
fit1.3 |> summary()
```

```
##
## Call:
## lm(formula = GVA ~ Business_br, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8006 -1.5308 -0.6353  1.8746  5.6300
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.0547     5.9992  -2.676  0.02325 *
## Business_br   2.3264     0.5646   4.121  0.00208 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.397 on 10 degrees of freedom
## Multiple R-squared:  0.6293, Adjusted R-squared:  0.5923
## F-statistic: 16.98 on 1 and 10 DF,  p-value: 0.002075
```

```r
fit1.4 <- lm(Labor_prod ~ Business_br, data = df1)

df1 |>
  #select(Business_br, Labor_prod) |>
  ggplot(aes(x = Business_br, y = Labor_prod)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

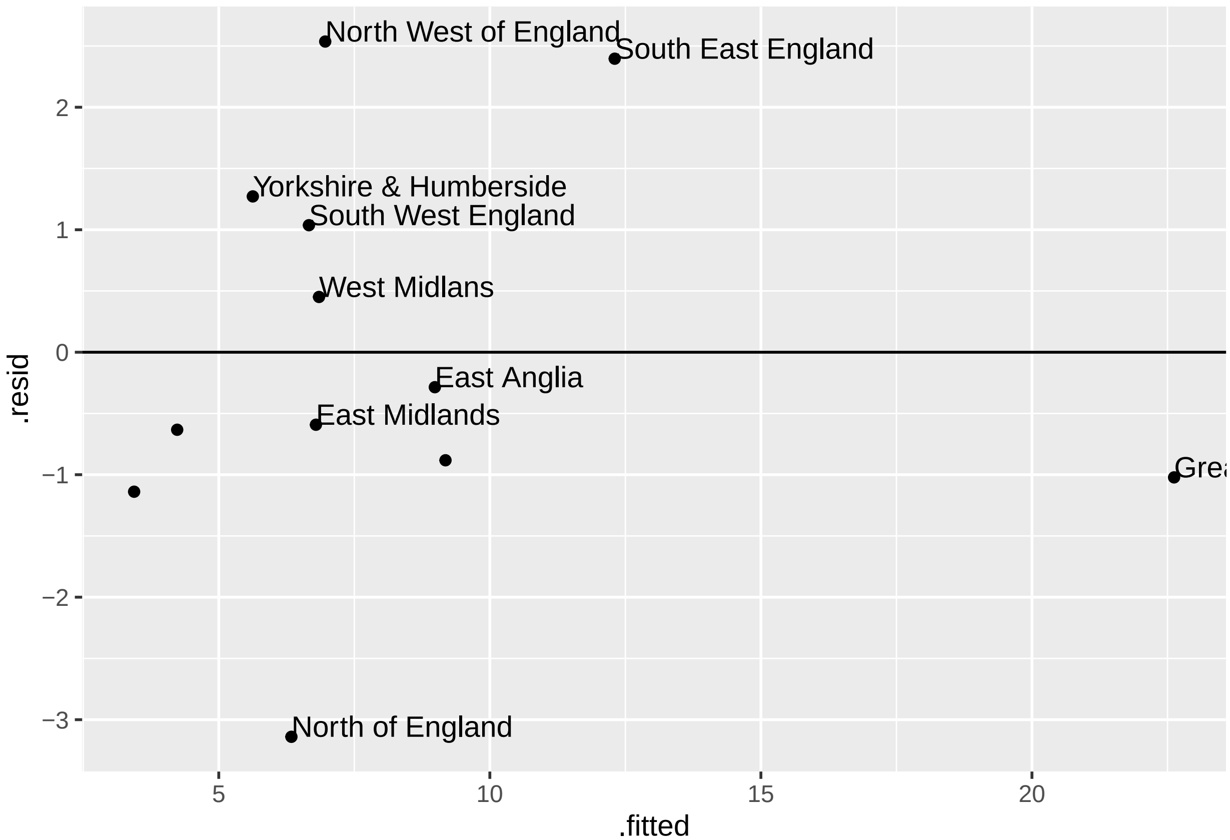



```
fit1.4 |> summary()
```

```
##
## Call:
## lm(formula = Labor_prod ~ Business_br, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.192  -6.589  -2.225   4.571  16.980
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.546     18.718   1.205  0.25613
## Business_br    6.861      1.761   3.895  0.00298 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.6 on 10 degrees of freedom
## Multiple R-squared:  0.6027, Adjusted R-squared:  0.563
## F-statistic: 15.17 on 1 and 10 DF,  p-value: 0.002984
```

? Patern géographique ?

```
fit1.1 |>
  ggplot(aes(x = .fitted, y = .resid, label = df1$Region)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_text(hjust=0, vjust=0)
```



Le test de Breusch-Pagan ne permet pas de rejeter l'hypothèse nulle d'homoscédasticité.

Le test de Jarque-Bera ne permet pas de rejeter l'hypothèse nulle de normalisé des résidus.

```
fit1.1 |>
  bptest()
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit1.1
## BP = 1.5183, df = 2, p-value = 0.4681
```

```
fit1.1$residuals |>
  jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  fit1.1$residuals
## X-squared = 0.091982, df = 2, p-value = 0.9551
```

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("RO11", "RO12", "RO21", "RO22",
                   "RO31", "RO32", "RO41", "RO42")
row.names(W21) <- c("RO11", "RO12", "RO21", "RO22",
                    "RO31", "RO32", "RO41", "RO42")
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("RO11", "RO12", "RO21", "RO22",
                 "RO31", "RO32", "RO41", "RO42")
rom_regions <- 1:8
nbrom <- read.gal("data/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("data/romania_inf_mor_rate.csv",
               header = FALSE)

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

Exercice 2.4

1. Wales
2. Scotland
3. Northen 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

```r
ukgal <- read.gal("data/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
```

```r

# Exo 2.5

ukdat <- read.csv("data/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=eyJxdWVyaWVzIjpbeyJjb25maWciOnsiZGF0YXNldCI6InVzLXN0YXRlLWJvdW5kYXJpZXMiLCJvcHRpb25zIjp7fX0sImNoYXJ0cyI6W3siYWxpZ25Nb250aCI6dHJ1ZSwidHlwZSI6ImNvbHVtbiIsImZ1bmMiOiJBVkciLCJ5QXhpcyI6ImdpZCIsInNjaWVudGlmaWNEaXNwbGF5Ijp0cnVlLCJjb2xvciI6IiNGRjUxNUEifV0sInhBeGlzIjoibmFtZSIsIm1heHBvaW50cyI6NTAsInNvcnQiOiIifV0sInRpbWVzY2FsZSI6IiIsImRpc3BsYXlMZWdlbmQiOnRydWUsImFsaWduTW9udGgiOnR

ydWV9

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).

```r
us <- read_sf("data/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()
```

Chapitre 3

Question 3.2 L'estimation par maximum de vraisemblance d'un modèle SARAR(1, 1) est intensif d'un point de vue computationnel et il n'existe présentement pas de preuve formelle que les MLE possèdent les propriétés asymptotiques habituelles d'un MLE (incluant estimateur consistant ?).

L'estimateur GS2SLS est consistent, mais pas pleinement efficient. L'alternative est le *Best Feasible GS2SLS* (BFG2SLS). Celui-ci atteint la borne inférieur pour la variance de l'estimateur dans les grand échantillons (Cramér-Rao ?). Numériquement intensif à calculer pour des grands échantillons.

Exercice 3.2

$$\begin{aligned}
y &= X\beta + u, \qquad u = \rho W u + \varepsilon \\
\varepsilon &= (I - \rho W)u \\
(I - \rho W)y &= (I - \rho W)X\beta + \underbrace{(I - \rho W)u}_{\varepsilon} \\
y &= \rho W y + X\beta - \rho\beta W X + \varepsilon, \quad \gamma = -\rho\beta \\
y &= \rho W y + X\beta + \gamma W X + \varepsilon
\end{aligned}$$

On peut voir que le SEM peut être réécrit sous la forme d'une SLM avec lag spatial des variables indépendantes. Puisque le lag spatial de y fait parti des variables explicatives, l'estimation par OLS est problématique car les termes d'erreur sont corrélés avec celle-ci. De plus, la présence de $\gamma = -\rho\beta$ dans le modèle fait en sorte que le modèle n'est plus linéaire dans ses paramètres.

```r
data(boston)
```

Exercice 3.6

```r
fit1 <- lm(MEDV ~ CRIM + RM + INDUS + NOX + AGE + DIS + RAD + PTRATIO +
             B + LSTAT + TAX, data = boston.c)
fit1 |>
  summary()
```

exemple 3.3

```
##
## Call:
## lm(formula = MEDV ~ CRIM + RM + INDUS + NOX + AGE + DIS + RAD +
##     PTRATIO + B + LSTAT + TAX, data = boston.c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7429  -2.8887  -0.7514   1.8144  26.8277
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.308337   5.199690   7.175 2.66e-12 ***
## CRIM         -0.103402   0.033339  -3.102 0.002035 **
## RM            4.074379   0.420639   9.686  < 2e-16 ***
## INDUS         0.018212   0.062015   0.294 0.769138
## NOX         -17.829176   3.889690  -4.584 5.79e-06 ***
## AGE          -0.002647   0.013353  -0.198 0.842957
## DIS          -1.210182   0.186123  -6.502 1.94e-10 ***
## RAD           0.304603   0.066878   4.555 6.62e-06 ***
## PTRATIO      -1.131146   0.126079  -8.972  < 2e-16 ***
## B             0.009853   0.002735   3.603 0.000346 ***
## LSTAT        -0.525072   0.051543 -10.187  < 2e-16 ***
## TAX          -0.010901   0.003710  -2.939 0.003452 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.838 on 494 degrees of freedom
## Multiple R-squared:  0.7293, Adjusted R-squared:  0.7233
## F-statistic:   121 on 11 and 494 DF,  p-value: < 2.2e-16
```

```r
# Test hétéroscédasticité
fit1 |>
  bptest()
```

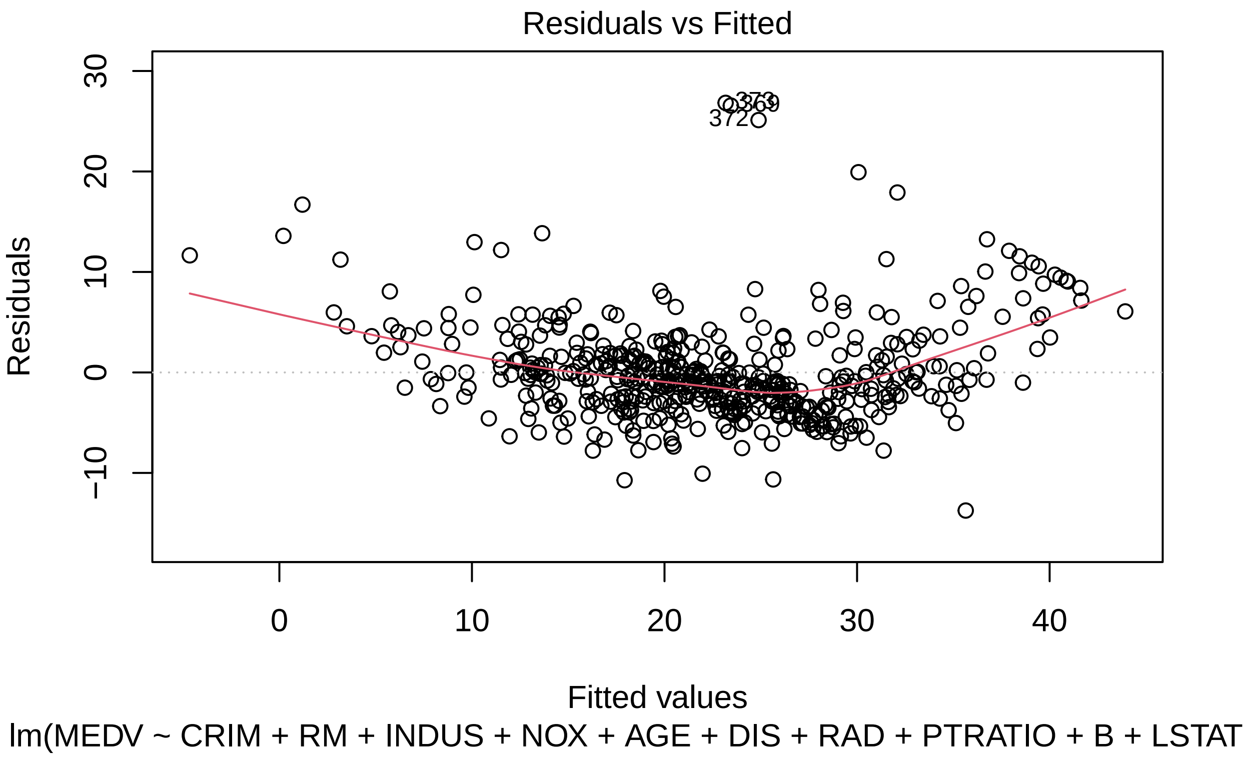
```
##
##  studentized Breusch-Pagan test
##
## data:  fit1
## BP = 59.214, df = 11, p-value = 1.297e-08
```

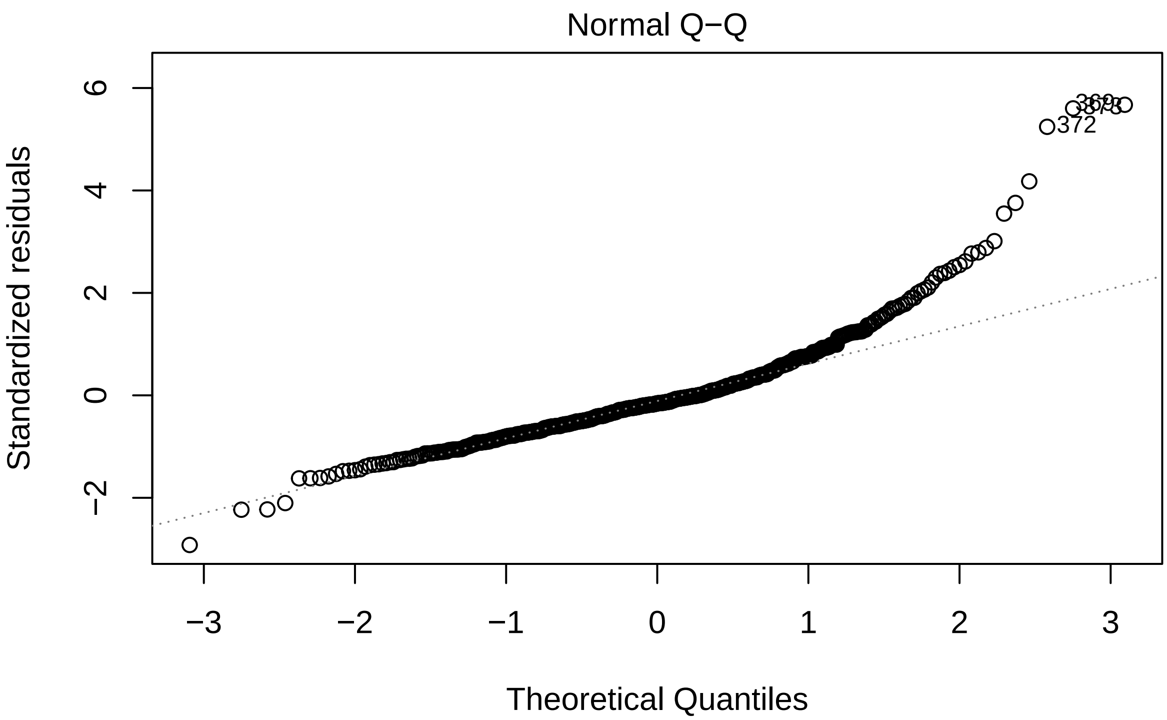
```r
# Test normalité résidu
fit1$residuals |>
  jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  fit1$residuals
## X-squared = 936.74, df = 2, p-value < 2.2e-16
```

```r
dist.centroid <- 3.99
```

```r
fit1 |> plot(which = c(1,2))
```



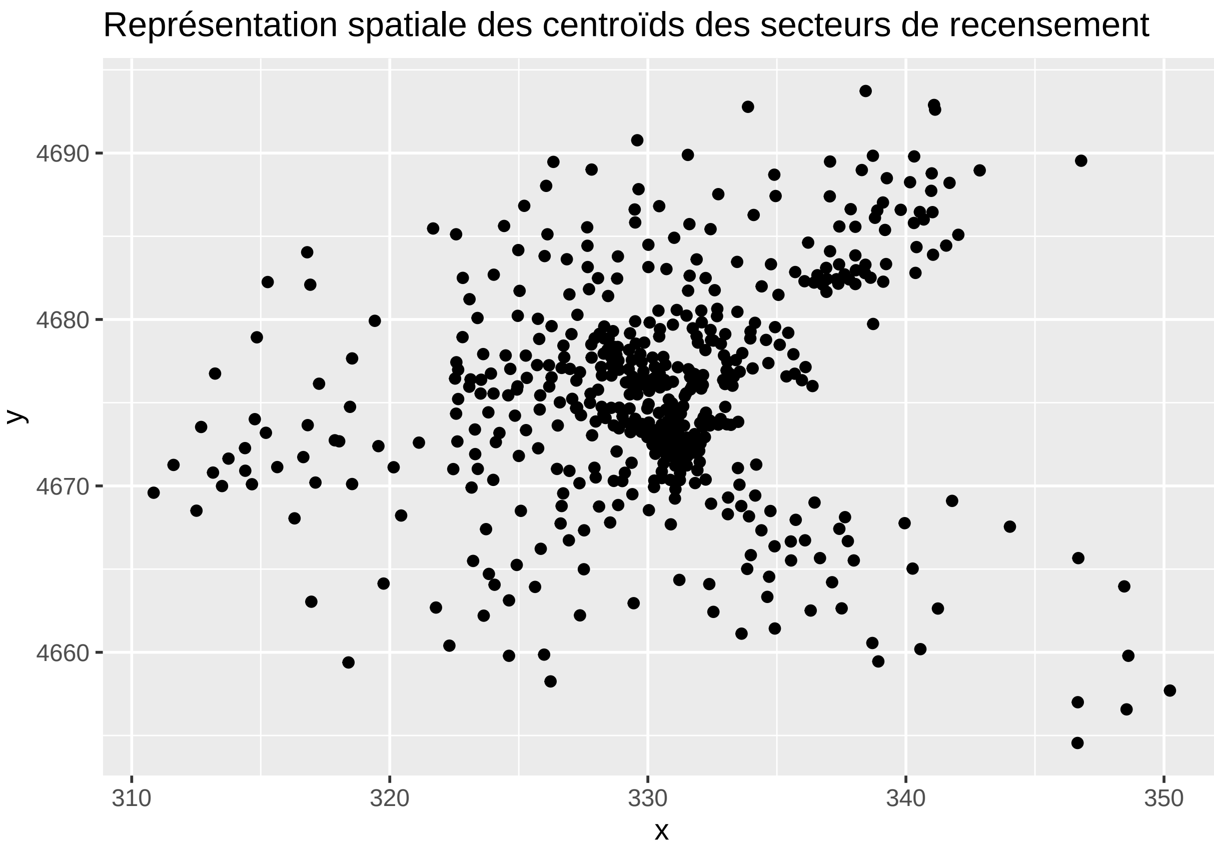


lm(MEDV ~ CRIM + RM + INDUS + NOX + AGE + DIS + RAD + PTRATIO + B + LSTAT

On peut voir qu'avec les graphiques et tests présentés que les résidus du modèle ne sont pas normalement distribués et qu'ils ne sont pas homoscédastiques.

On test ensuite la corrélation spatiale dans les résidus.

```
boston.utm |>
  ggplot(aes(x=x, y=y)) +
  geom_point() +
  ggtitle("Représentation spatiale des centroïds des secteurs de recensement")
```



Comme dans l'exemple du livre, le test avec le seuil de distance à 3.99 montre la présence d'une corrélation spatiale significative dans les résidus.

```
dmat1 <- dnearneigh(boston.utm, 0, d2 = dist.centroid, longlat = FALSE)
```

```
## Warning in dnearneigh(boston.utm, 0, d2 = dist.centroid, longlat = FALSE):
## neighbour object has 2 sub-graphs
```

```
dmat1 <- dmat1 |> nb2listw()

lm.morantest(fit1, dmat1, alternative = "two.sided")
```

```
##
##  Global Moran I for regression residuals
##
## data:
## model: lm(formula = MEDV ~ CRIM + RM + INDUS + NOX + AGE + DIS + RAD +
## PTRATIO + B + LSTAT + TAX, data = boston.c)
## weights: dmat1
##
## Moran I statistic standard deviate = 6.7338, p-value = 1.652e-11
## alternative hypothesis: two.sided
## sample estimates:
## Observed Moran I      Expectation         Variance
##     0.0780022170    -0.0071438650     0.0001598831
```

```
# Estimation forme SEM
fit3.1 <- errorsarlm(
  formula = MEDV ~ CRIM + RM + INDUS + NOX + AGE + DIS + RAD + PTRATIO +
```

```r
             B + LSTAT + TAX,
  data = boston.c, listw = dmat1
  )
fit3.1 |> summary()
```

```
##
## Call:errorsarlm(formula = MEDV ~ CRIM + RM + INDUS + NOX + AGE + DIS +
##     RAD + PTRATIO + B + LSTAT + TAX, data = boston.c, listw = dmat1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -13.63314  -2.66307  -0.71901   1.79259  26.34075
##
## Type: error
## Coefficients: (asymptotic standard errors)
##                Estimate  Std. Error  z value  Pr(>|z|)
## (Intercept)  40.6458124   5.2937581   7.6781 1.621e-14
## CRIM         -0.1188867   0.0324540  -3.6632 0.0002491
## RM            3.8507202   0.4062432   9.4789 < 2.2e-16
## INDUS        -0.0059026   0.0618481  -0.0954 0.9239677
## NOX         -20.4193253   4.0011873  -5.1033 3.338e-07
## AGE          -0.0195181   0.0140191  -1.3923 0.1638451
## DIS          -1.4560841   0.2717635  -5.3579 8.419e-08
## RAD           0.3219532   0.0732975   4.3924 1.121e-05
## PTRATIO      -1.0407873   0.1366262  -7.6178 2.576e-14
## B             0.0098856   0.0026439   3.7391 0.0001847
## LSTAT        -0.5149470   0.0496189 -10.3780 < 2.2e-16
## TAX          -0.0112409   0.0038685  -2.9058 0.0036637
##
## Lambda: 0.57191, LR test value: 25.792, p-value: 3.8025e-07
## Asymptotic standard error: 0.089708
##     z-value: 6.3752, p-value: 1.8267e-10
## Wald statistic: 40.644, p-value: 1.8267e-10
##
## Log likelihood: -1496.717 for error model
## ML residual variance (sigma squared): 21.332, (sigma: 4.6186)
## Number of observations: 506
## Number of parameters estimated: 14
## AIC: 3021.4, (AIC for lm: 3045.2)
```

```r
#fit3.1 |> bptest()
fit3.1$residuals |> jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  fit3.1$residuals
## X-squared = 1054, df = 2, p-value < 2.2e-16
```

```r
fit3.2 <- GMerrorsar(
  formula = MEDV ~ CRIM + RM + INDUS + NOX + AGE + DIS + RAD + PTRATIO +
             B + LSTAT + TAX,
  data = boston.c, listw = dmat1
  )
fit3.2 |> summary()
```

```
##
## Call:GMerrorsar(formula = MEDV ~ CRIM + RM + INDUS + NOX + AGE + DIS +
##     RAD + PTRATIO + B + LSTAT + TAX, data = boston.c, listw = dmat1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -13.02504  -2.89354  -0.71152   1.94452  26.75988
##
## Type: GM SAR estimator
## Coefficients: (GM standard errors)
##                Estimate  Std. Error  z value  Pr(>|z|)
## (Intercept)  40.5245582   5.2994923   7.6469 2.065e-14
## CRIM         -0.1180917   0.0325913  -3.6234 0.0002907
## RM            3.8591297   0.4082023   9.4540 < 2.2e-16
## INDUS        -0.0044561   0.0620706  -0.0718 0.9427686
## NOX         -20.2981369   4.0071806  -5.0654 4.075e-07
## AGE          -0.0186184   0.0140273  -1.3273 0.1844113
## DIS          -1.4431412   0.2636080  -5.4746 4.386e-08
## RAD           0.3217374   0.0731746   4.3968 1.098e-05
## PTRATIO      -1.0462088   0.1365008  -7.6645 1.799e-14
## B             0.0098673   0.0026561   3.7149 0.0002033
## LSTAT        -0.5156174   0.0498751 -10.3382 < 2.2e-16
## TAX          -0.0112381   0.0038747  -2.9004 0.0037267
##
## Lambda: 0.53872 (standard error): 0.60881 (z-value): 0.88488
## Residual variance (sigma squared): 21.557, (sigma: 4.6429)
## GM argmin sigma squared: 21.555
## Number of observations: 506
## Number of parameters estimated: 14
```

```r
fit3.3 <- errorsarlm(
  formula = MEDV ~ CRIM + RM + INDUS + NOX,
  data = boston.c, listw = dmat1
  )
fit3.3 |> summary()
```

```
##
## Call:errorsarlm(formula = MEDV ~ CRIM + RM + INDUS + NOX, data = boston.c,
##     listw = dmat1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -14.57612  -3.01967  -0.68837   1.98561  38.10993
##
## Type: error
## Coefficients: (asymptotic standard errors)
##              Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  -8.02740    3.70116 -2.1689  0.030091
## CRIM         -0.18980    0.03455 -5.4936 3.938e-08
## RM            6.52080    0.42291 15.4187 < 2.2e-16
## INDUS        -0.28169    0.06402 -4.4000 1.083e-05
## NOX         -12.59444    4.22584 -2.9803  0.002879
##
## Lambda: 0.70401, LR test value: 48.049, p-value: 4.1578e-12
## Asymptotic standard error: 0.069917
```

```
##     z-value: 10.069, p-value: < 2.22e-16
## Wald statistic: 101.39, p-value: < 2.22e-16
##
## Log likelihood: -1603.493 for error model
## ML residual variance (sigma squared): 32.138, (sigma: 5.669)
## Number of observations: 506
## Number of parameters estimated: 7
## AIC: 3221, (AIC for lm: 3267)
```

```r
fit3.4 <- GMerrorsar(
  formula = MEDV ~ CRIM + RM + INDUS + NOX,
  data = boston.c, listw = dmat1
  )
fit3.4 |> summary()
```

```
##
## Call:GMerrorsar(formula = MEDV ~ CRIM + RM + INDUS + NOX, data = boston.c,
##     listw = dmat1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -13.45597  -3.40510  -0.70401   2.74143  39.39475
##
## Type: GM SAR estimator
## Coefficients: (GM standard errors)
##                Estimate Std. Error z value  Pr(>|z|)
## (Intercept)   -9.352231   3.645678 -2.5653  0.010309
## CRIM          -0.187254   0.034845 -5.3738 7.708e-08
## RM             6.630713   0.427108 15.5247 < 2.2e-16
## INDUS         -0.266051   0.064548 -4.1217 3.761e-05
## NOX          -11.512572   4.165848 -2.7636  0.005717
##
## Lambda: 0.61776 (standard error): 0.53152 (z-value): 1.1623
## Residual variance (sigma squared): 33.169, (sigma: 5.7592)
## GM argmin sigma squared: 33.16
## Number of observations: 506
## Number of parameters estimated: 7
```

Exercice 3.7 Je n'arrive pas à trouver les données pour la courbe de Philips, je fais l'exercice pour la loi d'Okun

```r
ita_regions <- c(2, 3, 9, 1, 15, 19, 18, 11, 17, 4, 5, 12, 6, 10, 13, 7, 14, 8, 20, 16)
nbitaly <- read.gal("data/Italy.GAL",
                    region.id = ita_regions
                    )
witaly <- nb2listw(nbitaly)
italy_econ <- openxlsx::read.xlsx("data/ita_econ.xlsx")
colnames(italy_econ) <- c(
  # "id",
  "Region", "Var_unempl", "Var_rGDP")

fit3.7.1 <- lm(Var_unempl ~ Var_rGDP, data = italy_econ)
fit3.7.1 |> summary()
```

```
##
```

```
## Call:
## lm(formula = Var_unempl ~ Var_rGDP, data = italy_econ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4449 -1.7419 -0.3307  1.4994  6.2162
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.971      1.283   8.551 9.38e-08 ***
## Var_rGDP      -3.326      0.835  -3.984 0.000871 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.562 on 18 degrees of freedom
## Multiple R-squared:  0.4686, Adjusted R-squared:  0.4391
## F-statistic: 15.87 on 1 and 18 DF,  p-value: 0.0008705
```

```
fit3.7.1 |> bptest()
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit3.7.1
## BP = 0.022502, df = 1, p-value = 0.8808
```

```
fit3.7.1$residuals |> jarque.bera.test()
```

```
##
##  Jarque Bera Test
##
## data:  fit3.7.1$residuals
## X-squared = 1.2331, df = 2, p-value = 0.5398
```

Résultats différents de l'exemple 2.3 ?

```
lm.morantest(fit3.7.1, listw = witaly, alternative = "two.sided")
```

```
##
##  Global Moran I for regression residuals
##
## data:
## model: lm(formula = Var_unempl ~ Var_rGDP, data = italy_econ)
## weights: witaly
##
## Moran I statistic standard deviate = -0.25586, p-value = 0.7981
## alternative hypothesis: two.sided
## sample estimates:
## Observed Moran I      Expectation         Variance
##      -0.09274142      -0.04667655       0.03241371
```