Spatial CODA

Supplemental material

Thi Huong An Nguyen, Anne Ruiz-Gazen, Christine Thomas-Agnan, Thibault Laurent

Contents

We provide the data and the **R** code used in the article "Spatial CODA" so that readers may reproduce all the figures, tables and statistics presented in the article with the **R** software.

If you use this code, please cite:

Nguyen T.H.A, Ruiz-Gazen, A., Thomas-Agnan C. and T. Laurent (2019). Spatial CODA. *WP*.

1 Prerequisites

Required packages:

```
install.packages(c("compositions", "mvnfast", "quantmod", "plot3D", "sp"))
```

Loading packages:

```
require("classInt") # discretize numeric variable
require("compositions") # compositional data
require("ggplot2") # ggplot functions
require("mvnfast") # multivariate Student distribution
require("quantmod")  # import financial data
require("plot3D") # plot distribution in 3D
require("RColorBrewer") # palette colors with R
require("rgdal") # import spatial data
require("sp") # spatial data
require("spdep") # spatial econometric modelling
```

Information about the current R session :

```
sessionInfo()
```

```
## R version 3.5.3 (2019-03-11)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.6 LTS
##
## Matrix products: default
```

```
## BLAS: /usr/lib/openblas-base/libblas.so.3
## LAPACK: /usr/lib/libopenblasp-r0.2.18.so
##
## locale:
##  [1] LC_CTYPE=fr_FR.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=fr_FR.UTF-8        LC_COLLATE=fr_FR.UTF-8
##  [5] LC_MONETARY=fr_FR.UTF-8    LC_MESSAGES=fr_FR.UTF-8
##  [7] LC_PAPER=fr_FR.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=fr_FR.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] spdep_0.8-1         spData_0.3.0        Matrix_1.2-15
##  [4] rgdal_1.3-6         sp_1.3-1            RColorBrewer_1.1-2
##  [7] plot3D_1.1.1        quantmod_0.4-13     TTR_0.23-4
## [10] xts_0.11-2          zoo_1.8-4           mvnfast_0.2.5
## [13] ggplot2_3.1.0       compositions_1.40-2 bayesm_3.1-1
## [16] energy_1.7-5        robustbase_0.93-3   tensorA_0.36.1
## [19] classInt_0.3-1
##
## loaded via a namespace (and not attached):
##  [1] gtools_3.8.1      tidyselect_0.2.5  xfun_0.5
##  [4] purrr_0.2.5       splines_3.5.3     lattice_0.20-38
##  [7] expm_0.999-3      colorspace_1.4-0  htmltools_0.3.6
## [10] yaml_2.2.0        rlang_0.3.1       e1071_1.7-0
## [13] pillar_1.3.1      glue_1.3.0        withr_2.1.2
## [16] plyr_1.8.4        stringr_1.3.1     munsell_0.5.0
## [19] gtable_0.2.0      coda_0.19-2       evaluate_0.12
## [22] misc3d_0.8-4      knitr_1.21        curl_3.3
## [25] class_7.3-15      DEoptimR_1.0-8    Rcpp_1.0.0
## [28] scales_1.0.0      gdata_2.18.0      deldir_0.1-16
## [31] digest_0.6.18     gmodels_2.18.1    stringi_1.2.4
## [34] dplyr_0.8.0.1     grid_3.5.3        LearnBayes_2.15.1
## [37] tools_3.5.3       magrittr_1.5      lazyeval_0.2.1
## [40] tibble_2.0.1      crayon_1.3.4      pkgconfig_2.0.2
## [43] MASS_7.3-51.1     assertthat_0.2.0  rmarkdown_1.11
## [46] R6_2.3.0          boot_1.3-20       nlme_3.1-137
## [49] compiler_3.5.3
```

2 Simulation study

This section demonstrates how to obtain the results presented in the section 3 of the article. We first present our functions which can be adapted to another framework different from our simulation process.

2.1 Simulation of spatial multivariate Y

The function *simu_spatial_multi_y()* simulates a multivariate Y of the form $Y = Y\Gamma + WYR + X\beta + \epsilon$ where ϵ follows either a multivariate Gaussian (**method_simulate = "N"**), or the Independent multivariate Student (**method_simulate = "IT"**) distributions.

Input arguments are :

- **X**, the matrix of explanatory variables of size $n \times K$,
- **beta_true**, the β matrix of size $K \times L$:

$$\begin{pmatrix} \beta_{11} & \dots & \beta_{1L} \\ \beta_{21} & \dots & \beta_{2L} \\ \vdots & & \vdots \\ \beta_{K1} & \dots & \beta_{KL} \end{pmatrix}$$

- **method_simulate**, the method of simulation (a character among "N", "IT"),
- **Sigma**, the matrix of size $L \times L$,
- **GAMMA**, the matrix of size $L \times L$,
- **RHO**, the matrix of size $L \times L$,
- **W**, the matrix of size $n \times n$,
- **nu**, for Student distribution only.

The function returns a matrix of size $n \times L$. To load the function:

```
r   source("./R/simu_spatial_multi_y.R")
```

2.2 Examples

2.2.1 Preparation of the data

Import the Midi-Pyrénées communes boundaries into **R** which was used in Goulard et al. (2017):

```
mapMAP <- readOGR(dsn = "contours", layer = "ADTCAN_region")
```

We convert the type of the identification units into numeric values:

```
r   mapMAP@data$CODE <- as.numeric(as.character(mapMAP@data$CODE))
```

The number of observations equals to n:

```
r   n <- nrow(mapMAP)
```

We consider one spatial weight matrix W, based on the 10-nearest neighbours and row-normalized. W is relatively sparse (96.5% of null values).

```
coords <- coordinates(mapMAP)
W1.listw <- nb2listw(knn2nb(knearneigh(coords, 10)),
                     style = "W")
W_simu <- listw2mat(W1.listw)
```

2.2.2 Simulation of a multivariate SAR process

2.2.2.1 Example when $L = 2$

We plan to simulate a multivariate Y of size $L = 2$:

```
L_simu <- 2
```

We simulate the explanatory variables:

```r
set.seed(1234)
x1 <- rnorm(n, 15, 3)
x2 <- rbinom(n, 100, 0.45)/100
x3 <- log(round(runif(n, 1, n),0))
x_simu <- cbind(rep(1, n), x1, x2, x3)
p_simu <- ncol(x_simu)
```

The β matrix is

$$\begin{pmatrix} 5 & 2 \\ 1/4 & -1 \\ 6 & -3 \\ 1 & 3 \end{pmatrix},$$

```r
beta_true <- matrix(c(5, 2, 0.25, -1, 6, -3, 1, 3), byrow = T,
                    nrow = p_simu, ncol = L_simu)
```

the Σ matrix is

$$\begin{pmatrix} 10 & 8 \\ 8 & 10 \end{pmatrix},$$

```r
Sigma <- matrix(c(10, 9, 9, 10),
                nrow = L_simu, ncol = L_simu)
```

the Γ matrix is

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

```r
GAMMA <- matrix(c(0, 0, 0, 0),
                nrow = L_simu, ncol = L_simu)
```

and the R matrix is

$$\begin{pmatrix} 0.75 & 0.1 \\ 0.3 & 0.15 \end{pmatrix}.$$

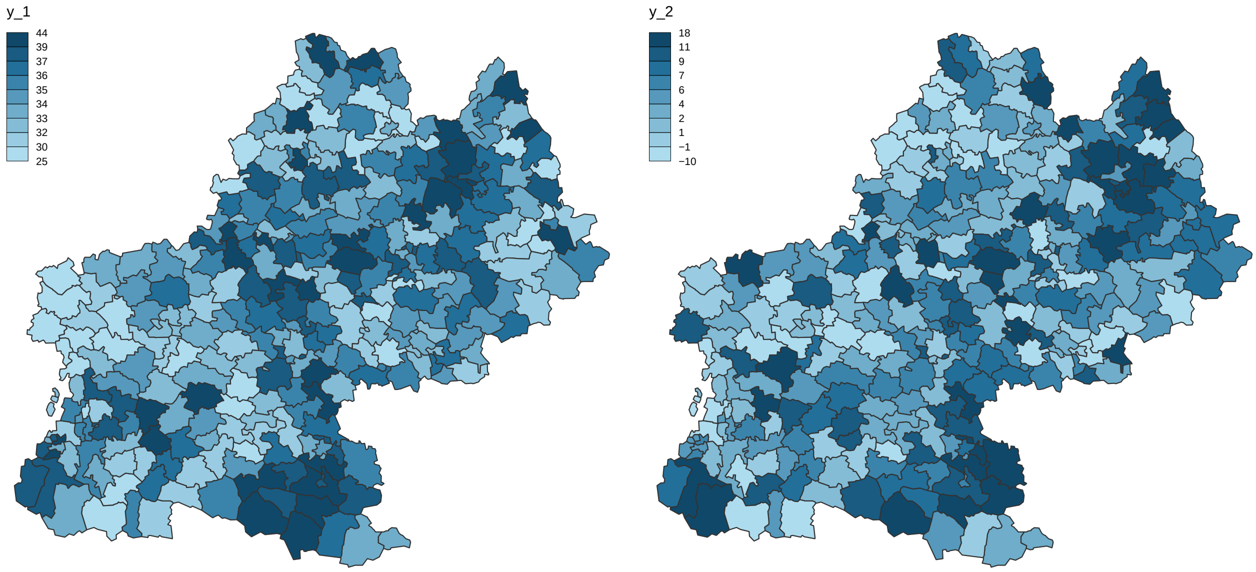
```r
RHO <- matrix(c(0.5, 0.1, 0.2, 0.4),
              nrow = L_simu, ncol = L_simu)
```

We simulate the process:

```r
set.seed(1)
y_N <- simu_spatial_multi_y(X = x_simu, beta_true = beta_true, method_simulate = "N",
                            Sigma = Sigma, GAMMA = GAMMA, RHO = RHO, W = W_simu)
mapMAP@data[, c("y_1", "y_2")] <- y_N
```

We plot the two component of Y on the map:

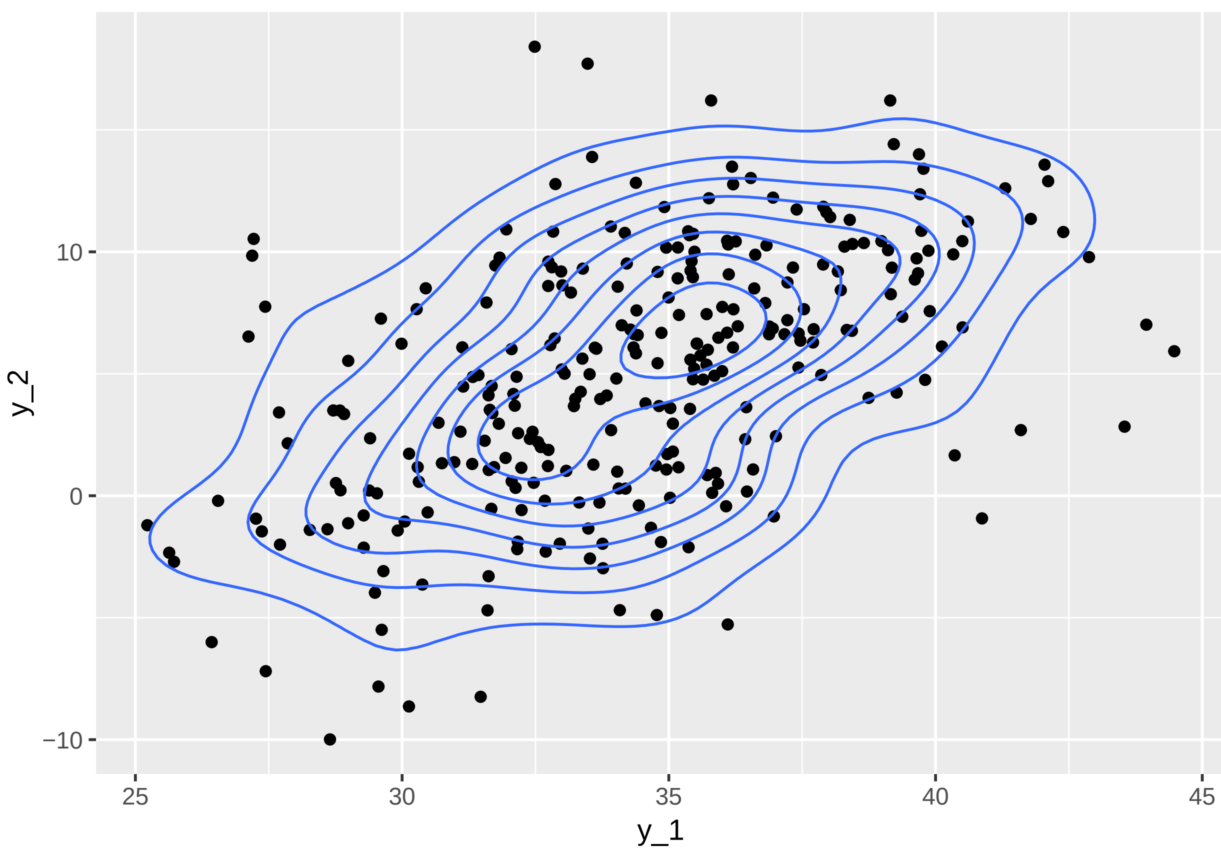
```r
library("cartography")
op <- par(mfrow = c(1, 2), oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))
choroLayer(spdf = mapMAP, var = "y_1", legend.pos = "topleft",
           method = "quantile")
choroLayer(spdf = mapMAP, var = "y_2", legend.pos = "topleft",
           method = "quantile")
```




```r
par(op)
```

We also plot the joint distribution:

```r
y_N_df <- data.frame(y_1 = y_N[, 1], y_2 = y_N[, 2])
sp <- ggplot(y_N_df, aes(x = y_1, y = y_2)) +
  geom_point()
sp + geom_density_2d()
```



2.2.2.2 Example when $L = 3$

We simulate another multivariate sample when $L = 3$.

```
L_3 <- 3
```

We keep the same explanatatory variable X. However, the β matrix is now equal to :

$$\begin{pmatrix} 5 & 2 & 10 \\ 1/4 & -1 & 4 \\ 6 & -3 & -5 \\ 1 & 3 & 5 \end{pmatrix}$$

and the Σ matrix is

$$\begin{pmatrix} 12 & 10 & 10 \\ 10 & 15 & 10 \\ 10 & 10 & 20 \end{pmatrix}$$

```
beta_true_3 <- matrix(c(5, 2, 10, 0, -1, 0, 6, -3, -5, 1, 0, 5), byrow = T,
                      nrow = p_simu, ncol = L_3)
Sigma_3 <- matrix(10, nrow = L_3, ncol = L_3)
diag(Sigma_3) <- c(12, 15, 20)
```

The Γ matrix is

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
GAMMA_3 <- matrix(c(0, 0, 0, 0, 0, 0, 0, 0, 0),
                  nrow = L_3, ncol = L_3)
```

The ρ matrix is

$$\begin{pmatrix} 0.75 & 0.1 & 0.1 \\ 0.1 & 0.25 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}$$

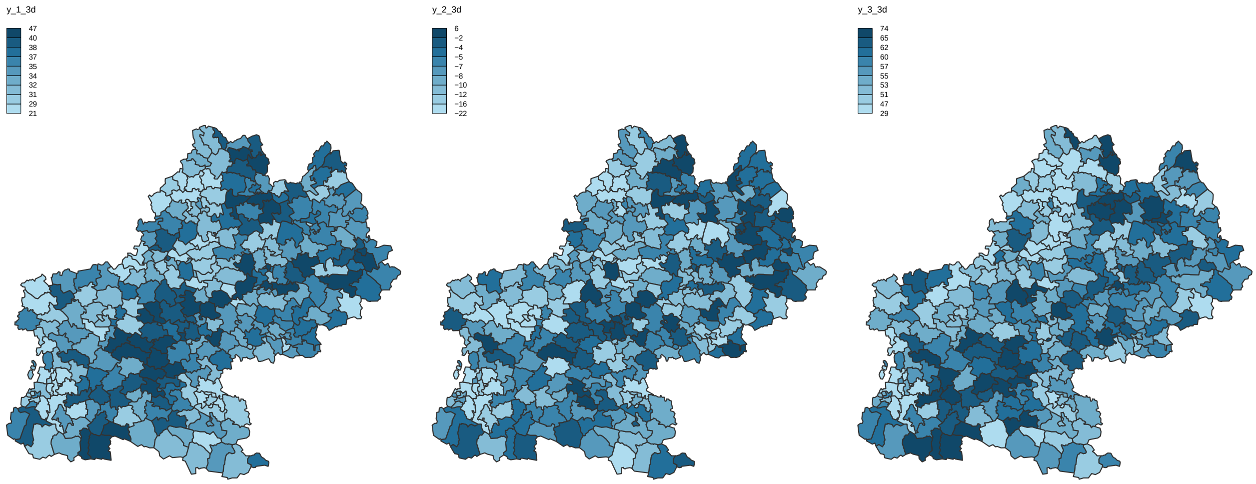
```
RHO_3 <- matrix(c(0.5, 0.1, 0.1, 0.1, 0.35, 0.1, 0.1, 0.1, 0.4),
                nrow = L_3, ncol = L_3)
```

Simulation of a "N" distribution

```
y_N_3d <- simu_spatial_multi_y(X = x_simu, beta_true = beta_true_3,
                               method_simulate = "N", Sigma = Sigma_3,
                               GAMMA = GAMMA_3, RHO = RHO_3, W = W_simu)
mapMAP@data[, c("y_1_3d", "y_2_3d", "y_3_3d")] <- y_N_3d
```

We plot the three component of Y on the map:

```
op <- par(mfrow = c(1, 3), oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))
choroLayer(spdf = mapMAP, var = "y_1_3d", legend.pos = "topleft",
           method = "quantile")
choroLayer(spdf = mapMAP, var = "y_2_3d", legend.pos = "topleft",
           method = "quantile")
choroLayer(spdf = mapMAP, var = "y_3_3d", legend.pos = "topleft",
           method = "quantile")
```




```
par(op)
```

3 Estimation

3.1 Estimation of the parameters by a 2SLS method

The function *estimate_spatial_multi_N()* estimates the coefficients associated to the multivariate Gaussian SAR model. The algorithm is based on Kelejian and Prucha (1998).

Input arguments are :

- **Y**, a matrix of size $n \times L$
- **X**, a matrix of explanatory variables of size $n \times K$,
- **W**, a spatial weight matrix of size $n \times n$,
- **GAMMA_esti**, a boolean which indicates if we estimate or nor the parameter associated to Γ.

The function returns a list with :

- the estimate of the β parameters
- the estimate of the Γ matrix
- the estimate of the R matrix
- the estimate of the Σ matrix

To load the function:

```
r   source("./R/estimate_spatial_multi_N.R")   source("./R/estimate_spatial_multi_gen_N.R")
```

3.1.1 Examples:

Simulated data when $L = 2$

```
r   (res_multi_N <- estimate_spatial_multi_N(Y = y_N, X = x_simu, W = W_simu))
```

```
## $res_beta   ##            [,1]       [,2]   ## [1,] -7.3252929 -9.0677619   ## [2,]
0.3043095 -0.9922779   ## [3,]  6.9474474 -0.8165804   ## [4,]  1.1161649  3.0405949
##    ## $GAMMA   ##      [,1] [,2]   ## [1,]    0    0   ## [2,]    0    0   ##    ##
$RHO   ##           [,1]       [,2]   ## [1,] 0.8358371 0.01494557   ## [2,] 0.4021348
```

0.29272685 ## ## $SIGMA ## [,1] [,2] ## [1,] 9.014998 8.105109 ## [2,] 8.105109 9.279858

```
(res_multi_N <- estimate_spatial_multi_N(Y = y_N, X = x_simu, W = W_simu,
                                        GAMMA_esti = T))
```

```
## $res_beta
##           [,1]      [,2]
## [1,]  3.070692 -3.033701
## [2,]  1.441934 -1.242947
## [3,]  7.883639 -6.539398
## [4,] -2.369809  2.121177
##
## $GAMMA
##           [,1]     [,2]
## [1,] 0.0000000 1.146477
## [2,] 0.8237296 0.000000
##
## $RHO
##            [,1]       [,2]
## [1,]  0.3747987 -0.3206592
## [2,] -0.2863690  0.2804157
##
## $SIGMA
##           [,1]      [,2]
## [1,]  2.627891 -2.305596
## [2,] -2.305596  2.043972
```

Simulated data when $L = 2$

```
(res_multi_N_3D <- estimate_spatial_multi_N(Y = y_N_3d, X = x_simu,
                                           W = W_simu))
```

```
## $res_beta
##            [,1]         [,2]       [,3]
## [1,] 10.5530510 -2.636599430 24.8769438
## [2,]  0.0969592 -0.877363929  0.1132813
## [3,]  0.6918015 -3.812773183 -9.9230991
## [4,]  1.0590189 -0.002301595  5.3311221
##
## $GAMMA
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    0    0    0
## [3,]    0    0    0
##
## $RHO
##            [,1]      [,2]       [,3]
## [1,]  0.3950117 0.2003413 0.09472490
## [2,]  0.2139942 0.3923740 0.09734543
## [3,] -0.7940431 0.4328434 0.71622391
##
## $SIGMA
##          [,1]     [,2]     [,3]
## [1,] 13.57275 12.09942 12.66735
## [2,] 12.09942 17.64009 13.09129
```

```
## [3,] 12.66735 13.09129 24.79805
```

```
(res_multi_N_3D <- estimate_spatial_multi_N(Y = y_N_3d, X = x_simu, W = W_simu,
                                             GAMMA_esti = T))
```

```
## $res_beta
##            [,1]        [,2]         [,3]
## [1,]  1.3856919 -10.0494379   2.96466195
## [2,]  0.1286905  -0.9416618  -0.06689815
## [3,]  4.7615822  -3.9168973 -11.26933143
## [4,] -0.9533235  -0.8398716   3.12568153
##
## $GAMMA
##           [,1]       [,2]       [,3]
## [1,] 0.0000000 0.08490875 0.37750736
## [2,] 0.6245331 0.00000000 0.03304701
## [3,] 2.0825857 0.02478601 0.00000000
##
## $RHO
##              [,1]        [,2]        [,3]
## [1,]  0.676598882 0.003623774 -0.18392038
## [2,] -0.006462927 0.252950026  0.01451754
## [3,] -1.621992979 0.005890032  0.51653839
##
## $SIGMA
##             [,1]       [,2]       [,3]
## [1,]   6.4544527  0.7385622 -14.113839
## [2,]   0.7385622  7.5057496  -2.551571
## [3,] -14.1138391 -2.5515712  31.514558
```

```
(res_multi_N_gen_3D <- estimate_spatial_multi_gen_N(Y = y_N_3d, X = x_simu,
  W = W_simu, ind_beta = matrix(c(T, F, T, T, T, T, T, F, T, F, T, T), 4, 3),
  ind_RHO = matrix(c(T, T, T, T, T, F, F, T, T), 3, 3),
  ind_GAMMA = matrix(c(F, T, T, T, F, F, F, T, F), 3, 3)))
```

```
## $res_beta
##            [,1]       [,2]      [,3]
## [1,]  8.8776703 -2.1090810 -6.373815
## [2,]  0.0000000 -0.8843098  0.000000
## [3,]  2.8643163 -3.8420793 -9.120064
## [4,] -0.3398033  0.0000000  5.230563
##
## $GAMMA
##            [,1]        [,2]      [,3]
## [1,] 0.00000000 -0.07034131 0.2667993
## [2,] 0.04550243  0.00000000 0.0000000
## [3,] 0.00000000 -0.10555678 0.0000000
##
## $RHO
##           [,1]      [,2]      [,3]
## [1,] 0.3423796 0.1725941 0.0000000
## [2,] 0.1446937 0.3939227 0.1045764
## [3,] 0.5744307 0.0000000 0.3887645
##
## $SIGMA
```

```
##             [,1]      [,2]      [,3]
## [1,] 10.180133  9.485319  7.604104
## [2,]  9.485319 16.615843 13.920295
## [3,]  7.604104 13.920295 25.553121
```

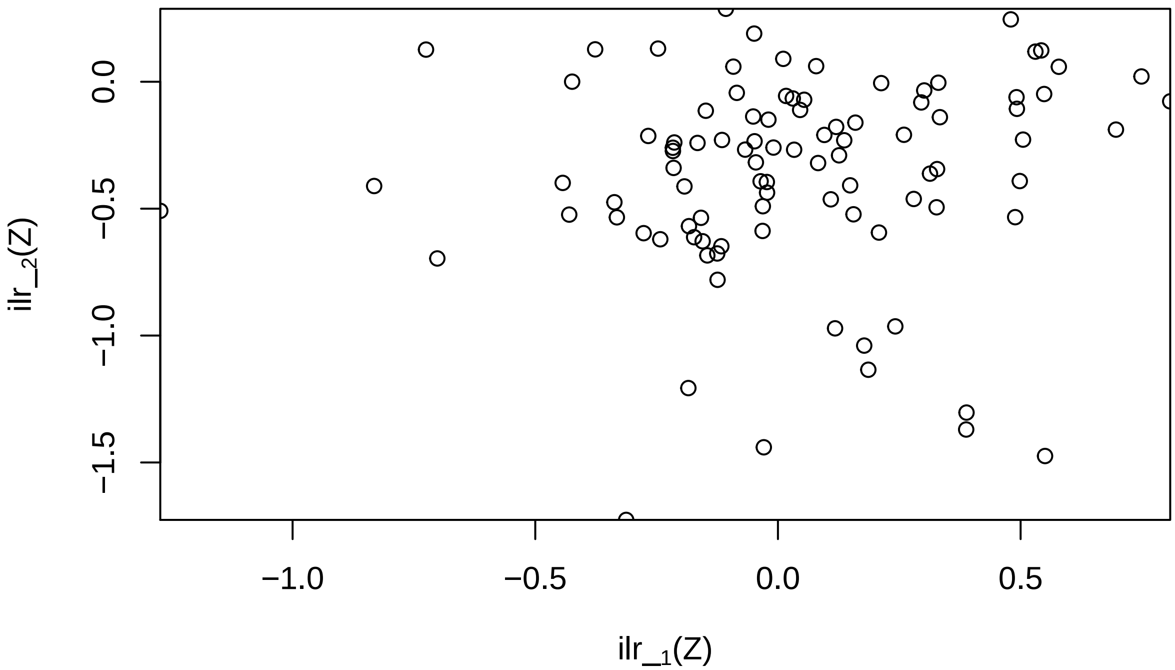
3.2 Application to the real data

We first load the data:

```r
source("R/preparation_base_ilr.R")
```

Then, we plot the data.

```r
Ye <- as(y_ilr, "matrix")
plot(Ye[,1], Ye[,2], xlab = expression(paste("ilr","_"[1],'(Z)')),
     ylab = expression(paste("ilr","_"[2],'(Z)')),
     xaxs = "i", yaxs = "i")
```



We prepare the explanatory variables:

```r
Xe <- as(cbind(1, x2_df[, c("age3_ilr1", "age3_ilr2",
                            "unemp_rate", "income_rate")]),
         "matrix")
```

3.2.1 Multivariate Gaussian model

We estimate first a multivariate gaussian model by using the *lm()* function.

```r
res_N <- lm(Ye ~ Xe - 1)
```

Then, we look the spatial distribution of the residuals. For this, we first compute a spatial weight matrix based on the 4-nearest neighbours.

```r
coords_fr <- coordinates(dep.2015.spdf)
W_listw <- nb2listw(knn2nb(knearneigh(coords_fr, 4)),
                    style = "W")
W_dep <- listw2mat(W_listw)
```

We test the spatial autocorrelation in the residuals component by component:

```r
moran.mc(residuals(res_N)[, 1], listw = W_listw, nsim = 1000)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  residuals(res_N)[, 1]
## weights: W_listw
## number of simulations + 1: 1001
##
## statistic = 0.25723, observed rank = 1001, p-value = 0.000999
## alternative hypothesis: greater
```

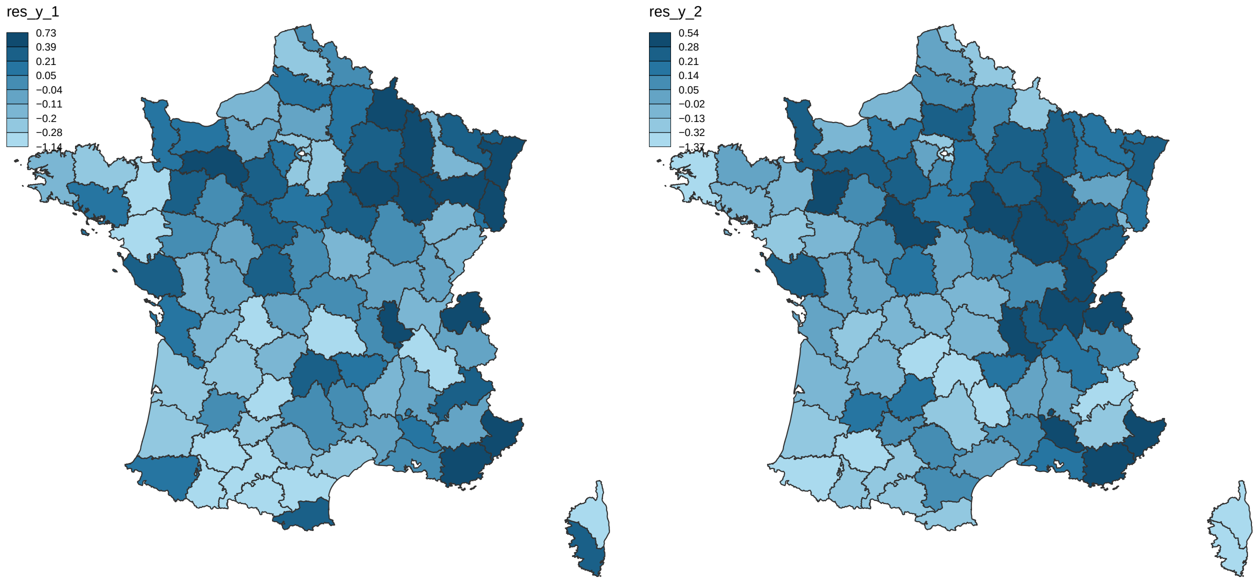
```r
moran.mc(residuals(res_N)[, 2], listw = W_listw, nsim = 1000)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  residuals(res_N)[, 2]
## weights: W_listw
## number of simulations + 1: 1001
##
## statistic = 0.24837, observed rank = 1001, p-value = 0.000999
## alternative hypothesis: greater
```

We plot the residuals:

```r
dep.2015.spdf@data[, c("res_y_1", "res_y_2")] <- residuals(res_N)
```

```r
library("cartography")
op <- par(mfrow = c(1, 2), oma = c(0, 0, 0, 0), mar = c(0, 0, 1, 0))
choroLayer(spdf = dep.2015.spdf, var = "res_y_1", legend.pos = "topleft",
           method = "quantile", legend.values.rnd = 2)
choroLayer(spdf = dep.2015.spdf, var = "res_y_2", legend.pos = "topleft",
           method = "quantile", legend.values.rnd = 2)
```




```r
par(op)
```

We estimate a multivariate gaussian SAR model by using the *lm()* function.

```r
(res_sar_N <- estimate_spatial_multi_N(Ye, Xe, W_dep, GAMMA_esti = F))
```

```
## $res_beta
##            [,1]       [,2]
## [1,] -0.1969978 -3.1077680
## [2,]  0.2719789  0.8525314
## [3,]  0.2965863 -0.2870640
## [4,] -4.5082459 13.2323330
## [5,]  1.7306086  2.0687909
## 
## $GAMMA
##      [,1] [,2]
## [1,]    0    0
## [2,]    0    0
## 
## $RHO
##           [,1]      [,2]
## [1,] 0.8895999 0.5202213
## [2,] 0.5246872 0.6698683
## 
## $SIGMA
##            [,1]       [,2]
## [1,] 0.08637361 0.01996587
## [2,] 0.01996587 0.08599642
```

4 References

- Goulard M., Laurent T. and Thomas-Agnan C. (2017). About predictions in spatial autoregressive models: optimal and almost optimal strategies, Spatial Economic Analysis, 12:2-3, 304-325, DOI: 10.1080/17421772.2017.1300679

- Nguyen T.H.A, Ruiz-Gazen, A., Thomas-Agnan C. and T. Laurent (2019). Multivariate Student versus Multivariate Gaussian Regression Models with Application to Finance. Journal of Risk and Financial Management, 12(1), 28.