Econometria espacial com R - Aula 05

Unicamp, julho de 2017

Raphael Saldanha, Eduardo Almeida

Probit e Tobit espacial

Pacotes

```
library(McSpatial)
```

```
## Loading required package: lattice
## Loading required package: locfit
## locfit 1.5-9.1    2013-03-22
## Loading required package: maptools
## Loading required package: sp
## Checking rgeos availability: TRUE
## Loading required package: quantreg
## Loading required package: SparseM
##
## Attaching package: 'SparseM'
## The following object is masked from 'package:base':
##
##     backsolve
## Loading required package: RANN
```

Shapefile

```
# Pacotes
library(rgdal)
```

```
## rgdal: version: 1.2-8, (SVN revision 663)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 1.11.3, released 2015/09/16
##  Path to GDAL shared files: /usr/share/gdal/1.11
##  Loaded PROJ.4 runtime: Rel. 4.9.2, 08 September 2015, [PJ_VERSION: 492]
##  Path to PROJ.4 shared files: (autodetected)
##  Linking to sp version: 1.2-5
```

```
lula.shp <- readOGR("data", "lula2002", encoding = "ISO-8859-1")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "data", layer: "lula2002"
## with 5507 features
## It has 23 fields
```

Variável de resposta

Para estes tipos de modelo, precisamos ter ou criar uma variável dicotômica para ser a variável de resposta do modelo.

```
lula.shp@data$VITORIA <- lula.shp@data$PT > lula.shp@data$PPS & lula.shp@data$PSB & lula.shp@data$ PSDB
table(lula.shp@data$VITORIA)
```

```
##
## FALSE  TRUE
##   358  5149
```

Matriz de vizinhança

Para usar o pacote `McSpatial`, a matriz de vizinhança precisa ser criada através de um comando do próprio pacote, não podendo ser usada diretamente as matrizes que criamos com o pacote `spdep`. Abaixo, alguns exemplos de matrizes com este pacote. Vamos usar a última nos exemplos.

```
# Matriz de pesos espaciais tipo "Queen"
wmat <- makew(lula.shp)$wmat
```

```
## Loading required package: Matrix
```

```
# Matriz de pesos espaciais tipo "Rook"
wmat <- makew(lula.shp, method="rook")$wmat

# Matriz de pesos espaciais tipo k-vizinhos
wmat <- makew(lula.shp, method="knear", knum=1)$wmat
```

Especificação

```
esp <- VITORIA ~ F1N + F2N + F3N + F4N
```

Logit espacial

```
fit1 <- splogit(form = esp, data = lula.shp@data, wmat = wmat)
```

```
##
## Call:
## glm(formula = form, family = binomial(link = "logit"), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.93267   0.08999   0.16209   0.37299   1.57369
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.3403     0.3012  -7.769 7.90e-15 ***
## F1N          10.4619     0.5675  18.436  < 2e-16 ***
## F2N           3.1929     6.3813   0.500  0.61683
## F3N           2.2868     0.8369   2.733  0.00629 **
## F4N           2.9712     0.4366   6.805 1.01e-11 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2649.2  on 5506  degrees of freedom
## Residual deviance: 2017.9  on 5502  degrees of freedom
## AIC: 2027.9
##
## Number of Fisher Scoring iterations: 8
##
## STANDARD LOGIT ESTIMATES
## LINEARIZED GMM LOGIT ESTIMATES
##             Estimate Std. Error  z-value Pr(>|z|)
## (Intercept) -2.48384    0.34628 -7.17286  0.00000
## F1N          9.22088    0.55403 16.64317  0.00000
## F2N          3.44637   10.64316  0.32381  0.74608
## F3N          1.74773    0.61619  2.83636  0.00456
## F4N          3.02821    0.39190  7.72692  0.00000
## WXB          0.21576    0.04304  5.01282  0.00000
## Number of observations =  5507
```

A especificação dos instrumentos envolve, por default, tanto X quanto WX . Para mudar esta especificação, você pode listar quais variáveis devem ser ponderadas pela matriz de pesos espaciais com o argumento `winst`, quais variáveis não devem ser ponderadas com o argumento `inst`, ou usando ambos os argumentos. Saiba mais sobre os efeitos destas combinações com `?splogit`, em Details.

Probit espacial

Estimado por GMM

```r
fit2 <- spprobit(form = esp, data = lula.shp@data, wmat = wmat)
```

```
##
## Call:
## glm(formula = form, family = binomial(link = "probit"), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.05165   0.05328   0.13326   0.38482   1.45241
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1556     0.1511  -7.645 2.08e-14 ***
## F1N           5.5295     0.2923  18.915  < 2e-16 ***
## F2N           1.4137     2.2864   0.618  0.53638
## F3N           1.3159     0.4163   3.161  0.00157 **
## F4N           1.5268     0.2272   6.721 1.80e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 2649.2  on 5506  degrees of freedom
## Residual deviance: 1988.1  on 5502  degrees of freedom
## AIC: 1998.1
##
## Number of Fisher Scoring iterations: 8
##
## STANDARD PROBIT ESTIMATES
## LINEARIZED GMM PROBIT ESTIMATES
##             Estimate Std. Error   z-value Pr(>|z|)
## (Intercept) -1.25933    0.10908 -11.54552   0.0000
## F1N          4.85698    0.23126  21.00256   0.0000
## F2N          1.64951    0.72345   2.28005   0.0226
## F3N          1.08684    0.22738   4.77982   0.0000
## F4N          1.58326    0.15459  10.24194   0.0000
## WXB          0.21644    0.03179   6.80787   0.0000
## Number of observations =  5507
```

Estimado por ML

Rodar em casa com tempo ;-)

```
fit3 <- spprobitml(form = esp, data = lula.shp@data, wmat = wmat)
```

```
## Standard Probit Estimates
##
## Call:
## glm(formula = form, family = binomial(link = "probit"), data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.05165   0.05328   0.13326   0.38482   1.45241
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1556     0.1511  -7.645 2.08e-14 ***
## F1N           5.5295     0.2923  18.915  < 2e-16 ***
## F2N           1.4137     2.2864   0.618  0.53638
## F3N           1.3159     0.4163   3.161  0.00157 **
## F4N           1.5268     0.2272   6.721 1.80e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2649.2  on 5506  degrees of freedom
## Residual deviance: 1988.1  on 5502  degrees of freedom
## AIC: 1998.1
##
## Number of Fisher Scoring iterations: 8
##
## Conditional on rho
## rho =  0.2483219
##          Estimate Std. Error    z-value     Pr(>|z|)
## VITORIA -1.188460  0.1423065 -8.3514121 0.000000e+00
```

```
## F1N      4.996325  0.2690747 18.5685397 0.000000e+00
## F2N      1.237156  2.3078522  0.5360637 5.919145e-01
## F3N      1.100396  0.3753069  2.9319894 3.367983e-03
## F4N      1.361295  0.2173777  6.2623485 3.792222e-10
## Unconditional Standard Errors
##           Estimate Std. Error    z-value     Pr(>|z|)
## VITORIA -1.1884602 0.25282116 -4.7007942 2.591516e-06
## F1N      4.9963251 0.44425006 11.2466504 0.000000e+00
## F2N      1.2371558 7.93365211  0.1559377 8.760821e-01
## F3N      1.1003958 0.60557561  1.8171073 6.920069e-02
## F4N      1.3612952 0.27192094  5.0062169 5.551022e-07
## rho      0.2483219 0.05051368  4.9159339 8.836021e-07
## Number of observations =  5507
```

Painel espacial

Para iniciarmos nossos painéis espaciais, pode ser interessante abrir espaço na memória do computador. Para apagar todos os objetos, use o comando abaixo.

Limpando a memória

```r
rm(list = ls())
```

Pacotes

O pacote `plm` é responsável pelos painéis convencionais (não espaciais) que usaremos para comparação. O pacote `splm` é responsável pelos painéis espaciais. Os autores do pacote lançaram um artigo sobre ele neste link.

```r
library(plm)
```

```
## Loading required package: Formula
```

```r
library(splm)
```

Shapefile

Vamos usar um shapefile de estados brasileiros com dados sobre criminalidade.

```r
# Pacotes
library(rgdal)

crime.shp <- readOGR("data", "Crime_UF3", encoding = "ISO-8859-1")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "data", layer: "Crime_UF3"
## with 27 features
## It has 27 fields
```

```r
# Plotar o mapa
plot(crime.shp)
```

Dados

Uma olhada nos dados.

```
str(crime.shp@data)
```

```
## 'data.frame':    27 obs. of  27 variables:
##  $ NAME2_  : Factor w/ 27 levels "ACRE","ALAGOAS",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ UF_CODE : num  12 27 16 13 29 23 53 32 52 21 ...
##  $ UF      : Factor w/ 27 levels "Acre","Alagoas",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ IGINI91 : num  0.623 0.625 0.582 0.623 0.664 0.654 0.614 0.598 0.585 0.599 ...
##  $ IGINI00 : num  0.648 0.691 0.637 0.683 0.669 0.675 0.64 0.608 0.611 0.659 ...
##  $ IDHMR91 : num  0.603 0.556 0.649 0.64 0.572 0.563 0.801 0.653 0.667 0.505 ...
##  $ IDHMR00 : num  0.64 0.598 0.666 0.634 0.62 0.616 0.842 0.719 0.717 0.558 ...
##  $ IDHNM91 : num  0.624 0.548 0.691 0.664 0.59 0.593 0.799 0.69 0.7 0.543 ...
##  $ IDHNM00 : num  0.697 0.649 0.753 0.713 0.688 0.7 0.844 0.765 0.776 0.636 ...
##  $ TURB91  : num  0.619 0.589 0.809 0.714 0.591 0.654 0.947 0.74 0.808 0.4 ...
##  $ TURB00  : num  0.664 0.68 0.89 0.749 0.671 0.715 0.956 0.795 0.879 0.595 ...
##  $ RPC91   : num  145 109 191 180 120 ...
##  $ RPC00   : num  181 140 211 174 160 ...
##  $ FVT99   : num  68.6 171.1 0 342.2 454.6 ...
##  $ FVT00   : num  196.2 80.1 0 245 433.9 ...
##  $ W_FVT99 : num  506 338 215 432 280 ...
##  $ W_FVT00 : num  398 330 316 334 248 ...
##  $ W_GINI99: num  0.618 0.648 0.619 0.62 0.621 ...
##  $ W_GINI00: num  0.648 0.667 0.655 0.634 0.647 ...
##  $ W_IDHR99: num  0.631 0.584 0.599 0.636 0.601 ...
##  $ W_IDHR00: num  0.658 0.629 0.629 0.67 0.654 ...
##  $ W_IDH99 : num  0.662 0.602 0.65 0.662 0.629 ...
##  $ W_IDH00 : num  0.724 0.692 0.723 0.735 0.715 ...
##  $ W_TURB99: num  0.648 0.657 0.525 0.621 0.672 ...
##  $ W_TURB00: num  0.695 0.717 0.665 0.705 0.753 ...
##  $ W_RPC99 : num  171 130 142 181 149 ...
##  $ W_RPC00 : num  204 169 169 221 205 ...
```

```
head(crime.shp@data)
```

```
##     NAME2_ UF_CODE       UF IGINI91 IGINI00 IDHMR91 IDHMR00 IDHNM91
```

```
## 0     ACRE      12     Acre   0.623   0.648   0.603   0.640   0.624
## 1  ALAGOAS      27  Alagoas   0.625   0.691   0.556   0.598   0.548
## 2    AMAPA      16    Amap    0.582   0.637   0.649   0.666   0.691
## 3 AMAZONAS      13 Amazonas   0.623   0.683   0.640   0.634   0.664
## 4    BAHIA      29    Bahia   0.664   0.669   0.572   0.620   0.590
## 5    CEARA      23    Cear    0.654   0.675   0.563   0.616   0.593
##   IDHNM00 TURB91 TURB00  RPC91  RPC00  FVT99  FVT00  W_FVT99  W_FVT00
## 0   0.697  0.619  0.664 144.73 180.70  68.56 196.21 506.4300 398.3250
## 1   0.649  0.589  0.680 109.13 139.91 171.13  80.13 337.8333 330.0200
## 2   0.753  0.809  0.890 190.59 211.39   0.00   0.00 214.8700 315.5000
## 3   0.713  0.714  0.749 180.09 173.92 342.25 244.96 431.7820 334.1840
## 4   0.688  0.591  0.671 119.71 160.19 454.65 433.89 280.0213 248.0637
## 5   0.700  0.654  0.715 113.86 156.24 117.66 176.21 284.6900 258.1550
##   W_GINI99 W_GINI00 W_IDHR99 W_IDHR00  W_IDH99  W_IDH00 W_TURB99 W_TURB00
## 0 0.617500 0.648500 0.631000 0.658500 0.662000 0.724000 0.648000 0.695000
## 1 0.648333 0.666667 0.584333 0.629000 0.602333 0.691667 0.657333 0.716667
## 2 0.619000 0.655000 0.599000 0.629000 0.650000 0.723000 0.525000 0.665000
## 3 0.619800 0.633800 0.636200 0.670400 0.662200 0.734800 0.621200 0.705000
## 4 0.620750 0.647375 0.600875 0.653625 0.628625 0.714500 0.671625 0.753125
## 5 0.638000 0.659250 0.559750 0.618000 0.587750 0.681750 0.642500 0.709500
##    W_RPC99  W_RPC00
## 0 170.9150 203.8800
## 1 129.5167 169.1500
## 2 141.5200 168.5900
## 3 181.0200 220.7360
## 4 148.9112 205.1125
## 5 113.6650 159.8025
```

Veja que ele já tem algumas variáveis defasadas espacialmente.

Matriz de vizinhança

Para rodar os paineis espaciais, vamos precisar de uma matriz de vizinhança.

```r
w1 <- nb2listw(poly2nb(crime.shp, queen = TRUE))
summary(w1)
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 27
## Number of nonzero links: 102
## Percentage nonzero weights: 13.99177
## Average number of links: 3.777778
## Link number distribution:
##
## 1 2 3 4 5 6 7 8
## 2 6 7 2 4 4 1 1
## 2 least connected regions:
## 2 20 with 1 link
## 1 most connected region:
## 4 with 8 links
##
## Weights style: W
## Weights constants summary:
```

```
##    n  nn S0       S1       S2
## W 27 729 27 16.95143 115.6406
```

Variáveis defasadas espacialmente

Caso queira criar outras variáveis defasadas espacialmente, use o seguinte comando.

```
crime.shp@data$lag_IDHMR91 <- lag.listw(w1, crime.shp@data$IDHMR91)
```

Para criar variáveis defasadas por uma matriz W, use o seguinte comando.

Empilhar dados

Para empilhar os dados de modo automático, vamos precisar que a base de dados tenha uma organização básica.

- O primeiro campo deve ser o de identificação;
- Os nomes das variáveis devem conter a especificação da variável e o ano, algo como "PIB2000" e "PIB2010".
- Não devem existir outras variáveis além da identificação e dados do painel.

Vejamos a base de dados do exemplo.

```
dados <- crime.shp@data
names(dados)
```

```
##  [1] "NAME2_"      "UF_CODE"     "UF"          "IGINI91"     "IGINI00"
##  [6] "IDHMR91"     "IDHMR00"     "IDHNM91"     "IDHNM00"     "TURB91"
## [11] "TURB00"      "RPC91"       "RPC00"       "FVT99"       "FVT00"
## [16] "W_FVT99"     "W_FVT00"     "W_GINI99"    "W_GINI00"    "W_IDHR99"
## [21] "W_IDHR00"    "W_IDH99"     "W_IDH00"     "W_TURB99"    "W_TURB00"
## [26] "W_RPC99"     "W_RPC00"     "lag_IDHMR91"
```

Primeiro, vamos retirar as variáveis que não precisamos, como as de nome da UF.

```
dados$NAME2_ <- NULL
dados$UF <- NULL
```

Vamos organizar os nomes das variáveis. Neste exemplo, entendi que 99 irá percenter no painel como se fosse 1991.

```
names(dados) <- c("coduf", "IGINI91", "IGINI00", "IDHMR91", "IDHMR00", "IDHNM91", "IDHNM00", "TURB91",
```

Para colocar os dados em painel, criamos uma função. Veja abaixo.

```
painel <- function(id, dados){
  require(reshape2)

  dadosp <- reshape2::melt(dados, id=id)
  dadosp$varname <- as.character(gsub("[[:digit:]]", "", dadosp$variable))
  dadosp$year <-  as.character(gsub("[[:alpha:]]", "", dadosp$variable))

  sp <- split(dadosp, f = dadosp$varname)

  dadosp <- data.frame(sp[[1]][,1], sp[[1]]$year)

  for(i in 1:length(sp)){
```

```r
    dadosp <- cbind(dadosp, sp[[i]]$value)
  }

  names(dadosp) <- c("id","ano",names(sp))

  return(dadosp)
}
```

Depois de declarada, vamos colocar os dados em painel.

```r
dadosp <- painel("coduf", dados)
```

```
## Loading required package: reshape2
```

```r
View(dadosp)
```

Especificação do modelo

```r
esp <- FVT ~ IDHMR + IGINI + RPC + TURB
```

Modelo não espacial de efeitos fixos

```r
fe <- plm(esp, data=dadosp)
```

Modelo não espacial de efeitos aleatórios

```r
re <- plm(esp, data=dadosp, model="random")
```

Teste de Hausman

```r
ph <- phtest(fe, re) # H0: efeitos aleatórios
print(ph)
```

```
##
##  Hausman Test
##
## data:  esp
## chisq = 22.845, df = 4, p-value = 0.000136
## alternative hypothesis: one model is inconsistent
```

Teste Pesaran CD (cross-section dependence)

```r
cd <- pcdtest(esp, data=dadosp) # H0: ausência de dependência CS
```

```
## Warning: Insufficient number of observations in time to estimate
## heterogeneous model: using within residuals
```

```r
print(cd)
```

```
##
##  Pesaran CD test for cross-sectional dependence in panels
##
## data:  FVT ~ IDHMR + IGINI + RPC + TURB
## z = -0.67937, p-value = 0.4969
## alternative hypothesis: cross-sectional dependence
```

Modelo OLS

```r
modOLS <- plm(esp, data=dadosp)
summary(modOLS)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = esp, data = dadosp)
##
## Balanced Panel: n=27, T=2, N=54
##
## Residuals :
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -1.5698e+02 -3.9605e+01 -1.5588e-13  3.9605e+01  1.5698e+02
##
## Coefficients :
##         Estimate Std. Error t-value Pr(>|t|)
## IDHMR  3117.0580  1844.8603  1.6896  0.10462
## IGINI  1523.1320  1183.7552  1.2867  0.21100
## RPC      -1.2625     1.0501 -1.2023  0.24149
## TURB  -2129.3507   606.2171 -3.5125  0.00187 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    323990
## Residual Sum of Squares: 200690
## R-Squared:      0.38057
## Adj. R-Squared: -0.42737
## F-statistic: 3.53278 on 4 and 23 DF, p-value: 0.021797
```

SAR

```r
modSAR <- spml(esp, data=dadosp, listw=w1, lag=TRUE, model="within", effect="individual", spatial.error=
summary(modSAR)
```

```
## Spatial panel fixed effects lag model
##
##
## Call:
## spml(formula = esp, data = dadosp, listw = w1, model = "within",
##     effect = "individual", lag = TRUE, spatial.error = "none")
```

```
##
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -1.5267e+02 -3.7177e+01 -1.6342e-13  3.7177e+01  1.5267e+02
##
## Spatial autoregressive coefficient:
##        Estimate Std. Error t-value Pr(>|t|)
## lambda  0.13991    0.15052  0.9295   0.3526
##
## Coefficients:
##          Estimate  Std. Error t-value Pr(>|t|)
## IDHMR  3113.50470  1192.08351  2.6118 0.009006 **
## IGINI  1496.29994   764.02283  1.9584 0.050177 .
## RPC      -1.14189     0.67826 -1.6836 0.092266 .
## TURB  -2108.78770   391.08042 -5.3922 6.96e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
impSAR <- impacts(modSAR, listw=w1, time=2)
```

```
## Note: method with signature 'diagonalMatrix#Matrix' chosen for function 'kronecker',
##  target signature 'ddiMatrix#dgCMatrix'.
##  "ANY#sparseMatrix" would also be valid

## Note: method with signature 'dsparseMatrix#dsparseMatrix' chosen for function 'kronecker',
##  target signature 'dtTMatrix#dgCMatrix'.
##  "TsparseMatrix#sparseMatrix" would also be valid
```

```
summary(impSAR, zstats=TRUE, short=TRUE)
```

```
## Impact measures (lag, trace):
##             Direct     Indirect        Total
## IDHMR  3130.910653  489.0623431  3619.972996
## IGINI  1504.664958  235.0354423  1739.700401
## RPC      -1.148271   -0.1793652    -1.327637
## TURB  -2120.576810 -331.2436471 -2451.820457
## ========================================================
## Simulation results ( variance matrix):
## ========================================================
## Simulated z-values:
##          Direct   Indirect     Total
## IDHMR  2.602727  0.7953104  2.332441
## IGINI  1.975586  0.7328942  1.843334
## RPC   -1.589076 -0.6772583 -1.496084
## TURB  -4.986712 -0.8550443 -3.570650
##
## Simulated p-values:
##       Direct    Indirect Total
## IDHMR 0.0092486 0.42643  0.0196775
## IGINI 0.0482016 0.46362  0.0652802
## RPC   0.1120433 0.49824  0.1346318
## TURB  6.1416e-07 0.39253  0.0003561
```

SEM

```
modSEM <- spml(esp, data=dadosp, listw=w1, lag=FALSE, model="within", effect="individual", spatial.erro
summary(modSEM)
```

```
## Spatial panel fixed effects error model
##
##
## Call:
## spml(formula = esp, data = dadosp, listw = w1, model = "within",
##     effect = "individual", lag = FALSE, spatial.error = "b")
##
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -1.5586e+02 -4.0632e+01 -8.2423e-13  4.0632e+01  1.5586e+02
##
## Spatial error parameter:
##     Estimate Std. Error t-value Pr(>|t|)
## rho  0.15213    0.16351  0.9304   0.3522
##
## Coefficients:
##         Estimate  Std. Error t-value  Pr(>|t|)
## IDHMR  2905.85256  1166.39022  2.4913   0.01273 *
## IGINI  1668.60798   797.85772  2.0914   0.03650 *
## RPC      -1.10762     0.66404 -1.6680   0.09531 .
## TURB  -2116.27268   388.85109 -5.4424 5.258e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

SAC

```
modSAC <- spml(esp, data=dadosp, listw=w1, lag=TRUE, model="within", effect="individual", spatial.error
```

```
## Note: method with signature 'sparseMatrix#ANY' chosen for function 'kronecker',
##  target signature 'dgCMatrix#dgeMatrix'.
##  "ANY#Matrix" would also be valid

## Note: method with signature 'dsparseMatrix#dsparseMatrix' chosen for function 'kronecker',
##  target signature 'dgCMatrix#dgTMatrix'.
##  "sparseMatrix#TsparseMatrix" would also be valid

## Note: method with signature 'sparseMatrix#matrix' chosen for function '%*%',
##  target signature 'dgTMatrix#matrix'.
##  "TsparseMatrix#ANY" would also be valid
```

```
summary(modSAC)
```

```
## Spatial panel fixed effects sarar model
##
##
## Call:
## spml(formula = esp, data = dadosp, listw = w1, model = "within",
##     effect = "individual", lag = TRUE, spatial.error = "b")
##
```

```
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -1.5312e+02 -3.7400e+01 -2.7001e-13  3.7400e+01  1.5312e+02
##
## Spatial error parameter:
##     Estimate Std. Error t-value Pr(>|t|)
## rho -0.17032    0.39728 -0.4287   0.6681
##
## Spatial autoregressive coefficient:
##        Estimate Std. Error t-value Pr(>|t|)
## lambda  0.26099    0.32116  0.8126   0.4164
##
## Coefficients:
##          Estimate  Std. Error t-value Pr(>|t|)
## IDHMR  3258.59998  1195.45912  2.7258 0.006414 **
## IGINI  1300.22830   724.02610  1.7958 0.072521 .
## RPC      -1.16548     0.70058 -1.6636 0.096196 .
## TURB  -2059.45097   401.25895 -5.1325 2.86e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
impSAC <- impacts(modSAC, listw=w1, time=2)
summary(impSAC, zstats=TRUE, short=TRUE)
```

```
## Impact measures (lag, trace):
##             Direct     Indirect        Total
## IDHMR  3325.669360 1083.7353232  4409.404683
## IGINI  1326.989953  432.4259960  1759.415949
## RPC      -1.189467   -0.3876114    -1.577078
## TURB  -2101.839147 -684.9259743 -2786.765121
## ========================================================
## Simulation results ( variance matrix):
## ========================================================
## Simulated z-values:
##           Direct   Indirect      Total
## IDHMR  1.2052794  0.1453059  0.1979137
## IGINI  0.7769948  0.1295189  0.1612875
## RPC   -1.3150148 -0.1624924 -0.2707244
## TURB  -0.9316626 -0.1361414 -0.1729949
##
## Simulated p-values:
##       Direct  Indirect Total
## IDHMR 0.22810 0.88447  0.84311
## IGINI 0.43716 0.89695  0.87187
## RPC   0.18850 0.87092  0.78660
## TURB  0.35151 0.89171  0.86266
```

Especificação com lag

```r
esp_lag <- FVT ~ W_IDHR + W_GINI + W_RPC + W_TURB
```

SDM

```r
modSDM <- spml(esp_lag, data=dadosp, listw=w1, lag=TRUE, model="within", effect="individual", spatial.e
summary(modSDM)
```

```
## Spatial panel fixed effects lag model
##
##
## Call:
## spml(formula = esp_lag, data = dadosp, listw = w1, model = "within",
##     effect = "individual", lag = TRUE, spatial.error = "none")
##
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -2.0971e+02 -3.5187e+01  2.0428e-13  3.5187e+01  2.0971e+02
##
## Spatial autoregressive coefficient:
##        Estimate Std. Error t-value Pr(>|t|)
## lambda  0.07991    0.16453  0.4857   0.6272
##
## Coefficients:
##           Estimate  Std. Error t-value Pr(>|t|)
## W_IDHR  2464.59941  1917.18677  1.2855  0.19861
## W_GINI -3521.14356  1885.05952 -1.8679  0.06177 .
## W_RPC     -0.87483     1.10673 -0.7905  0.42926
## W_TURB  -458.85899   820.91035 -0.5590  0.57619
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
impSDM <- impacts(modSDM, listw=w1, time=12)
summary(impSDM, zstats=TRUE, short=TRUE)
```

```
## Impact measures (lag, trace):
##               Direct      Indirect         Total
## W_IDHR  2491.0003021  187.64953717  2678.6498393
## W_GINI -3558.8622067 -268.09263949 -3826.9548462
## W_RPC     -0.8842016   -0.06660779    -0.9508094
## W_TURB  -463.7743046  -34.93658091  -498.7108855
## ========================================================
## Simulation results ( variance matrix):
## ========================================================
## Simulated z-values:
##             Direct   Indirect      Total
## W_IDHR  1.1618655  0.1886638  1.1042377
## W_GINI -1.8784369 -0.2786932 -1.7704805
## W_RPC  -0.6445114 -0.1004753 -0.6146074
## W_TURB -0.5304574 -0.1167380 -0.5175445
##
## Simulated p-values:
##        Direct   Indirect Total
```

```
## W_IDHR 0.245290 0.85036  0.269490
## W_GINI 0.060321 0.78048  0.076647
## W_RPC  0.519244 0.91997  0.538814
## W_TURB 0.595795 0.90707  0.604776
```

SDEM

```r
modSDEM <- spml(esp_lag, data=dadosp, listw=w1, lag=FALSE, model="within", effect="individual", spatial
summary(modSDEM)
```

```
## Spatial panel fixed effects error model
##
##
## Call:
## spml(formula = esp_lag, data = dadosp, listw = w1, model = "within",
##     effect = "individual", lag = FALSE, spatial.error = "b")
##
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -2.1031e+02 -3.2468e+01  7.8160e-14  3.2468e+01  2.1031e+02
##
## Spatial error parameter:
##     Estimate Std. Error t-value Pr(>|t|)
## rho 0.075105   0.168273  0.4463   0.6554
##
## Coefficients:
##           Estimate  Std. Error t-value Pr(>|t|)
## W_IDHR  2531.06305  1930.95382  1.3108  0.18993
## W_GINI -3636.76004  1880.85510 -1.9336  0.05317 .
## W_RPC     -0.99387     1.10583 -0.8987  0.36879
## W_TURB  -437.52348   812.49014 -0.5385  0.59023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

SLX

```r
modSLX <- plm(esp_lag, data=dadosp)
summary(modSLX)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = esp_lag, data = dadosp)
##
## Balanced Panel: n=27, T=2, N=54
##
## Residuals :
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -210.768  -32.762    0.000   32.762  210.768
##
## Coefficients :
##           Estimate  Std. Error t-value Pr(>|t|)
```

```
## W_IDHR  2504.72611  2948.26918  0.8496   0.4043
## W_GINI -3589.21472  2885.26516 -1.2440   0.2260
## W_RPC     -0.97475     1.68440 -0.5787   0.5684
## W_TURB  -451.79254  1261.83358 -0.3580   0.7236
##
## Total Sum of Squares:    323990
## Residual Sum of Squares: 243920
## R-Squared:      0.24712
## Adj. R-Squared: -0.7349
## F-statistic: 1.88735 on 4 and 23 DF, p-value: 0.14685
```