Ensayo avanzado Caso 2

Viacheslav Shalisko

2022-10-29

Prepar el entorno y cargar bibliotecas necesarias

```
set.seed(1234)
options(scipen = 6, digits = 3)

library(sp)
library(rgdal)
```

```
## 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/vshal/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/vshal/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(raster)
library(spdep)
```

```
## Loading required 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')`
```

```
## Loading required package: sf
```

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

```
library(ncf)
```

Lectura de datos fuente

```
mun <- readOGR("Insumos_carlos/LimiteMunicipal_MGJ2012_modificadoDecreto26837.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "G:\Mi unidad\UdeG_Docencia\CUCSH_Doctorado_TT3\2022B\Caso_2\Insumos_carlos\LimiteMunicipal
_MGJ2012_modificadoDecreto26837.shp", layer: "LimiteMunicipal_MGJ2012_modificadoDecreto26837"
## with 125 features
## It has 7 fields
## Integer64 fields read as strings:  OBJECTID CLAVE
```

```
mun
```

```
## class       : SpatialPolygonsDataFrame
## features    : 125
## extent      : 427476, 865451, 2096902, 2515726  (xmin, xmax, ymin, ymax)
## crs         : +proj=utm +zone=13 +datum=WGS84 +units=m +no_defs
## variables   : 7
## names       : OBJECTID,      NOMBRE,      REGIÓN, CLAVE,           km2,
DESCRIPCIO,             MINUS
## min values  :        1,      ACATIC, ALTOS NORTE,     1, 79.6982019022,
Mapa General de Jalisco 2012, publicado en el Periódico Oficial El Estado de Jalisco, el 27 de marzo d
e 2012, Acatlán de Juárez
## max values  :       99, ZAPOTLANEJO,      VALLES,    99, 3344.25486244, Mapa General de Jalisco 201
2, publicado en el Periódico Oficial El Estado de Jalisco, el 27 de marzo de 2012 y modificado por Dec
reto 26837/LXI/18 Mezquitic publicado en el Periódico Oficial El Estado de Jalisco, el 3 de junio de 2
018,       Zapotlanejo
```

```
DM <- readOGR("Insumos_carlos/Jal_Den_Mad_Todas.shp")
```

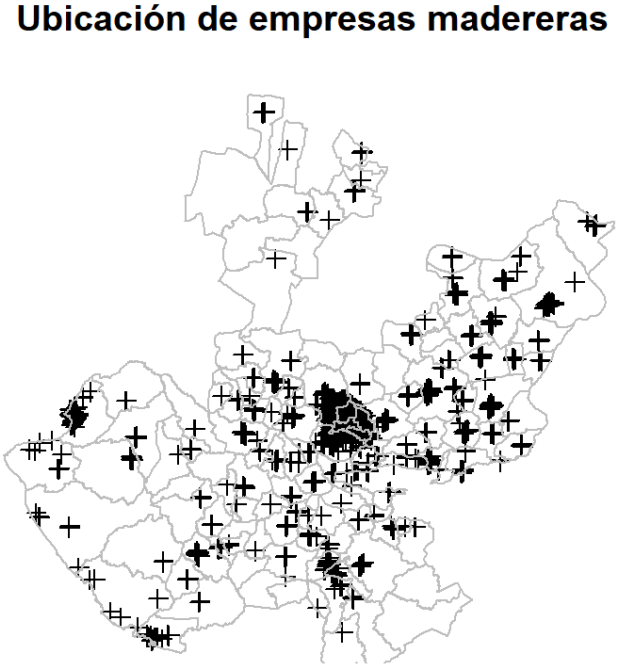
```
## OGR data source with driver: ESRI Shapefile
## Source: "G:\Mi unidad\UdeG_Docencia\CUCSH_Doctorado_TT3\2022B\Caso_2\Insumos_carlos\Jal_Den_Mad_Tod
as.shp", layer: "Jal_Den_Mad_Todas"
## with 2669 features
## It has 47 fields
## Integer64 fields read as strings:  tipEmpr Field
```

```
DM
```

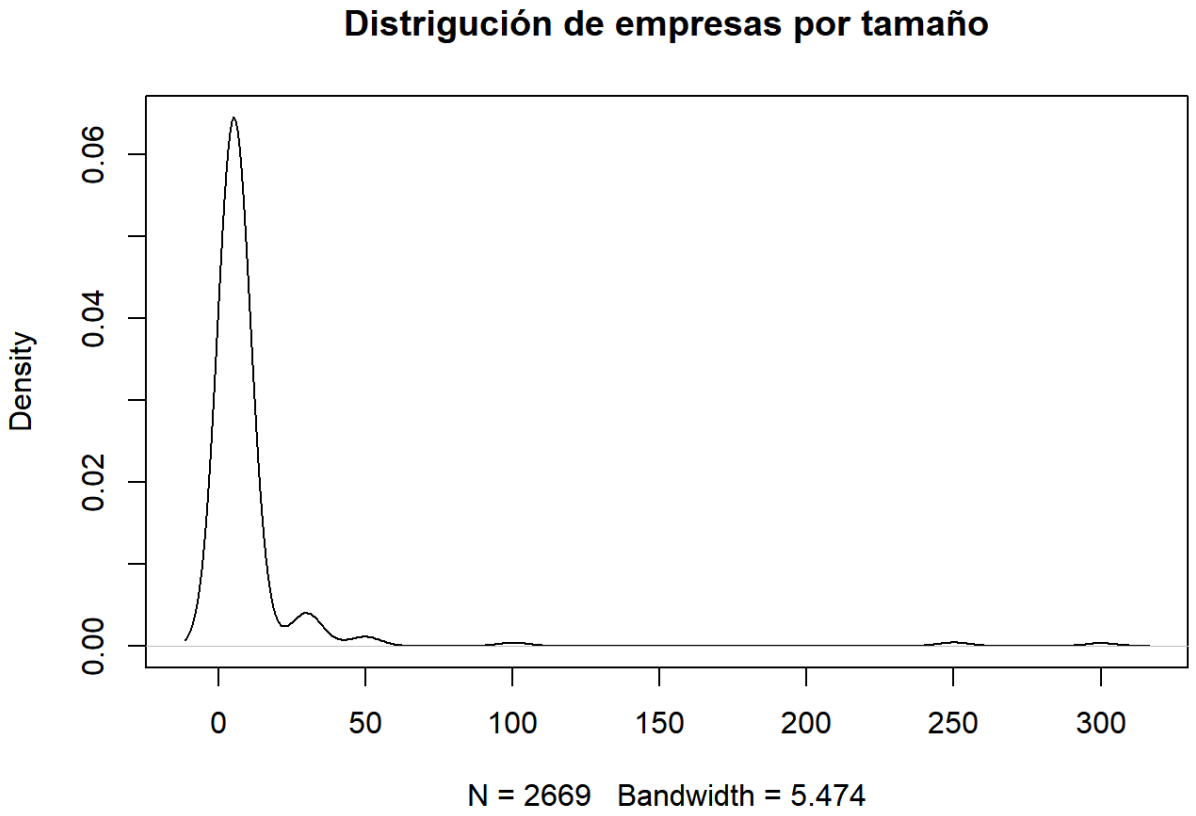
```
## class       : SpatialPointsDataFrame
## features    : 2669
## extent      : 445542, 852618, 2123027, 2503121  (xmin, xmax, ymin, ymax)
## crs         : +proj=utm +zone=13 +datum=WGS84 +units=m +no_defs
## variables   : 47
## names       :      id,                          clee,                             nom_estab,
raz_social, codigo_act,                                                  nombre_act,         per_oc
u, tipo_vial,    nom_vial,   tipo_v_e_1, nom_v_e_1,   tipo_v_e_2, nom_v_e_2, tipo_v_e_3, nom_v_e_3,
...
## min values  : 1625930, 14001321910000011000000000U1, ABASTECEDORA DE EMPAQUES DE OCIDENTE, ABASTECE
DORA DE EMPAQUES DE OCCIDENTE SA DE CV,     321111,                                                Aserraderos
integrados,  0 a 5 personas,   ANDADOR, 1° DE ABRIL,      ANDADOR,         1,      ANDADOR,         1,
ANDADOR,         1, ...
## max values  : 9365810, 15057322210000135001000000U3,                          XHIBE CARTON,
XHIBE CARTON SA DE CV,     333241, Tratamiento de la madera y fabricación de postes y durmientes, 6 a
10 personas,   RETORNO,    ZOQUIPAN, PROLONGACION,    ZUÑIGA, PROLONGACION,  ZARAGOZA,   VIADUCTO,  ZA
RAGOZA, ...
```

Visualizar datos fuente

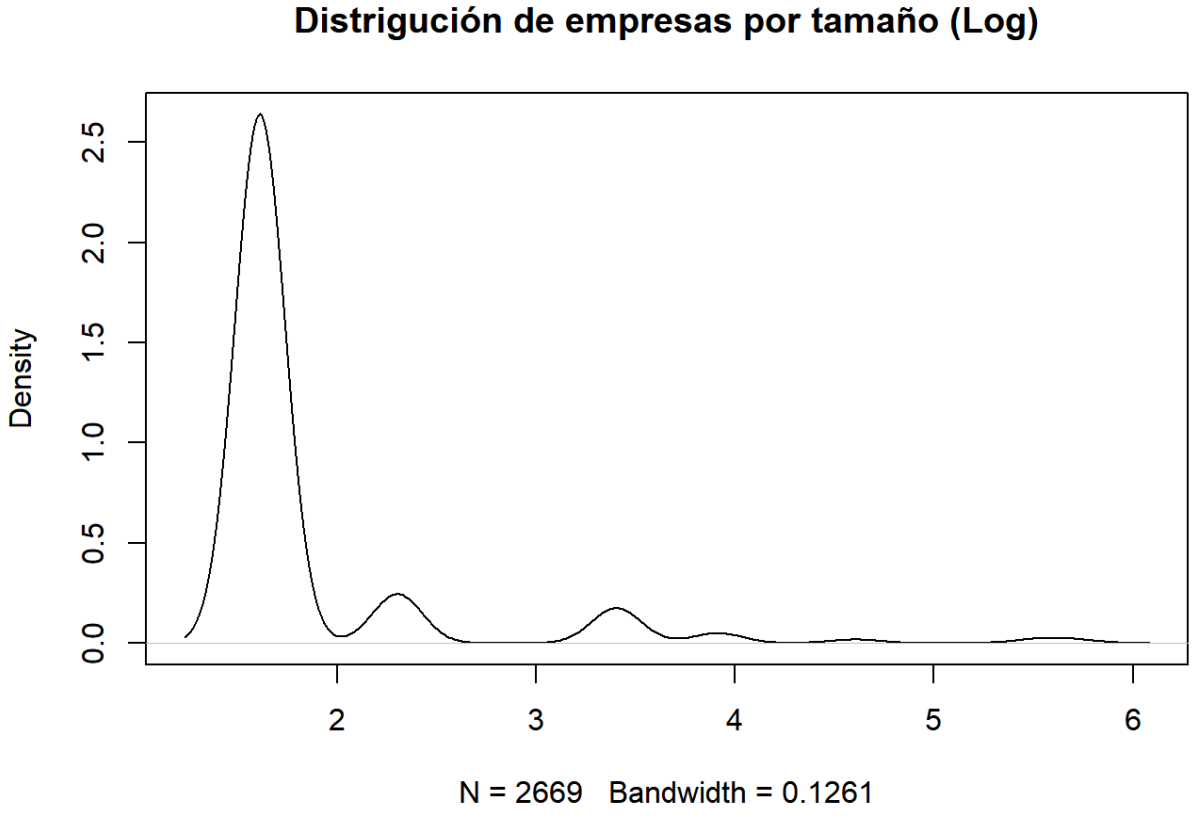
```
plot(DM, main = "Ubicación de empresas madereras")
plot(mun, border = "gray", add = TRUE)
```




```
plot(density(DM$Personas), main = "Distrigución de empresas por tamaño")
```




```
DM$Personas_log <- log(DM$Personas)
plot(density(DM$Personas_log), main = "Distrigución de empresas por tamaño (Log)")
```

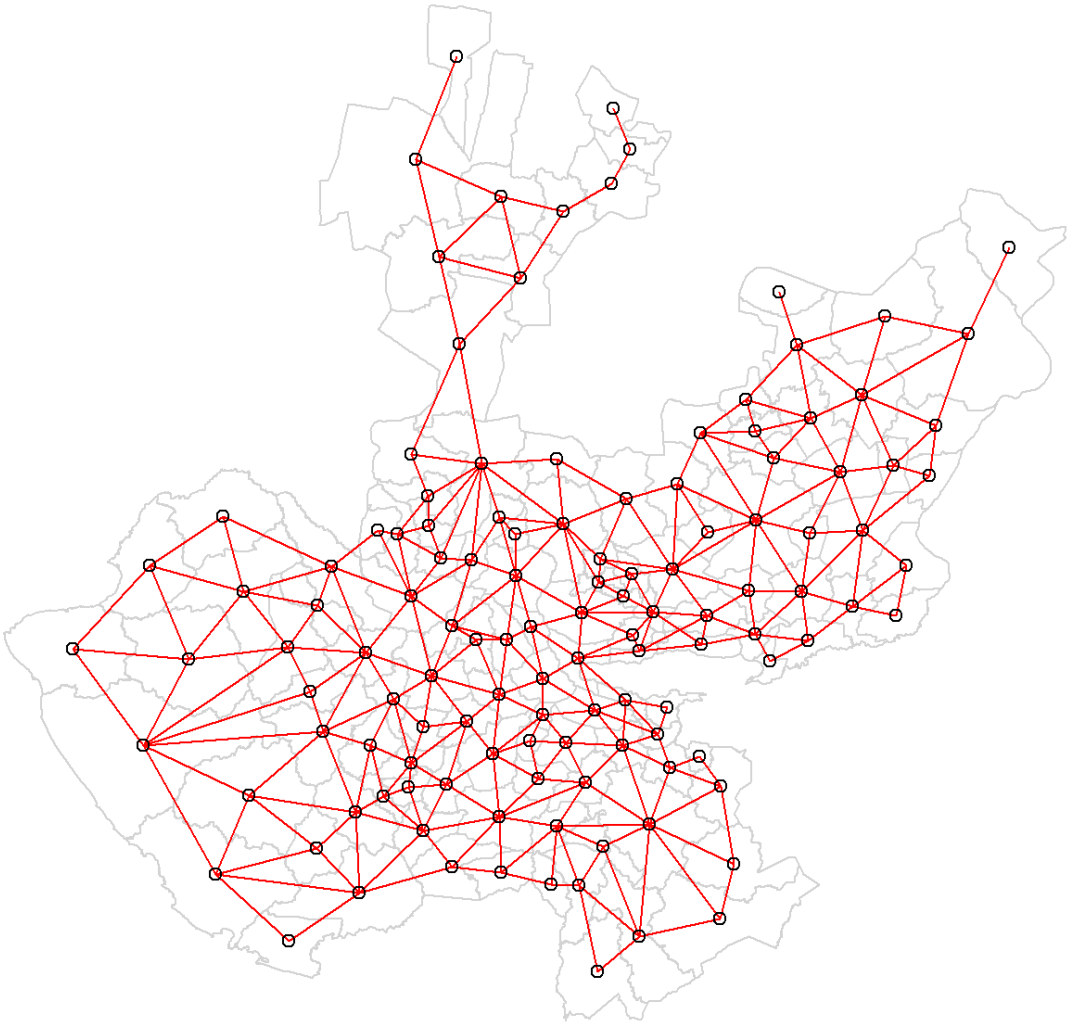


Identificar vecinos

```
mun_neighbours <- poly2nb(mun)
mun_neighbours
```

```
## Neighbour list object:
## Number of regions: 125
## Number of nonzero links: 616
## Percentage nonzero weights: 3.94
## Average number of links: 4.93
```

```
plot(mun, border = 'lightgrey')
plot(mun_neighbours, coordinates(mun), add = TRUE, col = 'red')
```



Calcular distancias con vecinos

```
mun_lw <- nb2listw(mun_neighbours)
mun_lw
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 125
## Number of nonzero links: 616
## Percentage nonzero weights: 3.94
## Average number of links: 4.93
##
## Weights style: W
## Weights constants summary:
##     n    nn  S0   S1  S2
## W 125 15625 125 58.1 519
```

Calcular número de personas involucrados en industria maderera por municipio

```
mun@data[,"CLAVE"] <- as.numeric(mun@data[,"CLAVE"])
DM_mun <- over(DM, mun[,"CLAVE"])
DM <- cbind(DM, DM_mun)

DM[,c("Personas","CLAVE")]
```

```
## class       : SpatialPointsDataFrame
## features    : 2669
## extent      : 445542, 852618, 2123027, 2503121  (xmin, xmax, ymin, ymax)
## crs         : +proj=utm +zone=13 +datum=WGS84 +units=m +no_defs
## variables   : 2
## names       : Personas, CLAVE
## min values  :        5,     1
## max values  :      300,   125
```

```
mun_sumas <- aggregate(Personas ~ CLAVE,
                       data = DM@data[,c("Personas","CLAVE")],
                       FUN = "sum")
mun <- merge(x = mun, y = mun_sumas, by = "CLAVE")

mun@data$Personas
```

```
##   [1]   55   35  120   15   55  130    5  320   20  195   NA    5  115   10  220
##  [16]   45   70  160   NA   60   15  120  575   45   70   50   10   NA   NA   65
##  [31]   NA    5   80   NA  115   75   65   10 4600   15   40   45   60   30   25
##  [46]  190   25   35   NA  125   15   20  385   15   20   NA   20   55   85   10
##  [61]   10  565   75   10  385  775   10   15 1715   NA   15  690   95   10    5
##  [76]  140  245  170   10   15   85  315   70   70  175   45   50   NA    5  185
##  [91]    5 1125  365    5   25 1575 2375    5   70 1250    5   15    5   60   NA
## [106]   10  385   25   40   15   30   25   25   15   15   NA   90   90 6485   85
## [121]   NA   25  140   50   10
```

```
mun@data[is.na(mun@data$Personas),"Personas"] <- 0
mun@data$Personas
```

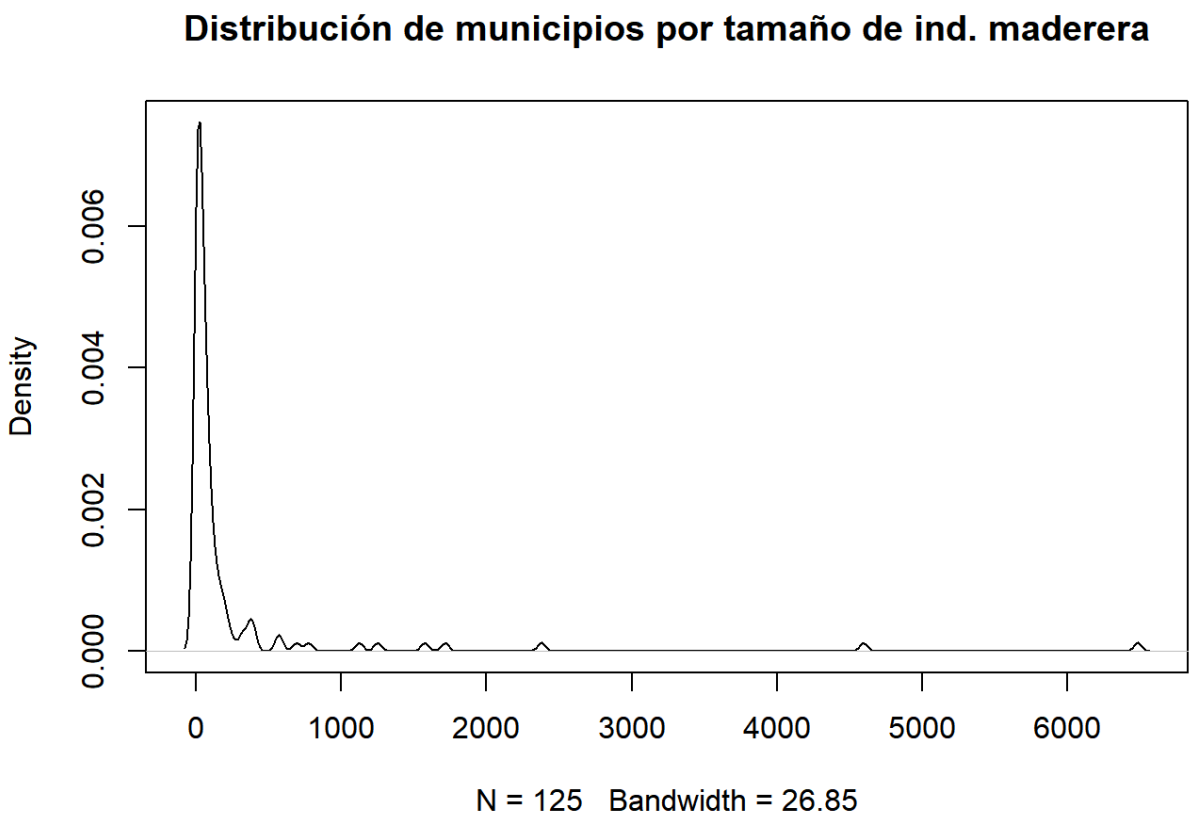
```
##   [1]   55   35  120   15   55  130    5  320   20  195    0    5  115   10  220
##  [16]   45   70  160    0   60   15  120  575   45   70   50   10    0    0   65
##  [31]    0    5   80    0  115   75   65   10 4600   15   40   45   60   30   25
##  [46]  190   25   35    0  125   15   20  385   15   20    0   20   55   85   10
##  [61]   10  565   75   10  385  775   10   15 1715    0   15  690   95   10    5
##  [76]  140  245  170   10   15   85  315   70   70  175   45   50    0    5  185
##  [91]    5 1125  365    5   25 1575 2375    5   70 1250    5   15    5   60    0
## [106]   10  385   25   40   15   30   25   25   15   15    0   90   90 6485   85
## [121]    0   25  140   50   10
```

```
mun@data$Personas_log <- log(mun@data$Personas)
mun@data[mun@data$Personas_log < 0,"Personas_log"] <- 0
mun@data$Personas_log
```

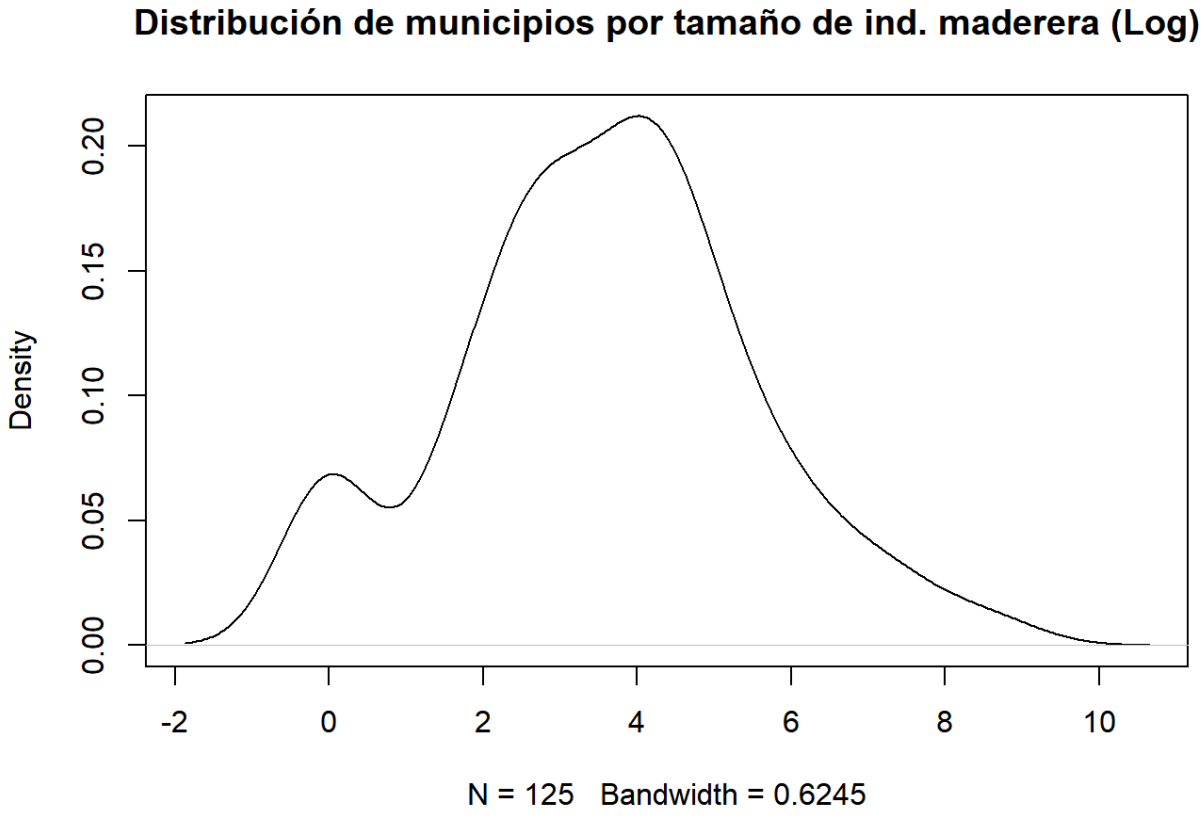
```
##   [1] 4.01 3.56 4.79 2.71 4.01 4.87 1.61 5.77 3.00 5.27 0.00 1.61 4.74 2.30 5.39
##  [16] 3.81 4.25 5.08 0.00 4.09 2.71 4.79 6.35 3.81 4.25 3.91 2.30 0.00 0.00 4.17
##  [31] 0.00 1.61 4.38 0.00 4.74 4.32 4.17 2.30 8.43 2.71 3.69 3.81 4.09 3.40 3.22
##  [46] 5.25 3.22 3.56 0.00 4.83 2.71 3.00 5.95 2.71 3.00 0.00 3.00 4.01 4.44 2.30
##  [61] 2.30 6.34 4.32 2.30 5.95 6.65 2.30 2.71 7.45 0.00 2.71 6.54 4.55 2.30 1.61
##  [76] 4.94 5.50 5.14 2.30 2.71 4.44 5.75 4.25 4.25 5.16 3.81 3.91 0.00 1.61 5.22
##  [91] 1.61 7.03 5.90 1.61 3.22 7.36 7.77 1.61 4.25 7.13 1.61 2.71 1.61 4.09 0.00
## [106] 2.30 5.95 3.22 3.69 2.71 3.40 3.22 3.22 2.71 2.71 0.00 4.50 4.50 8.78 4.44
## [121] 0.00 3.22 4.94 3.91 2.30
```

Visualizar distribución de municipios por tamaño de industria maderera

```
plot(density(mun$Personas), main = "Distribución de municipios por tamaño de ind. maderera")
```




```
plot(density(mun$Personas_log), main = "Distribución de municipios por tamaño de ind. maderera (Log)")
```

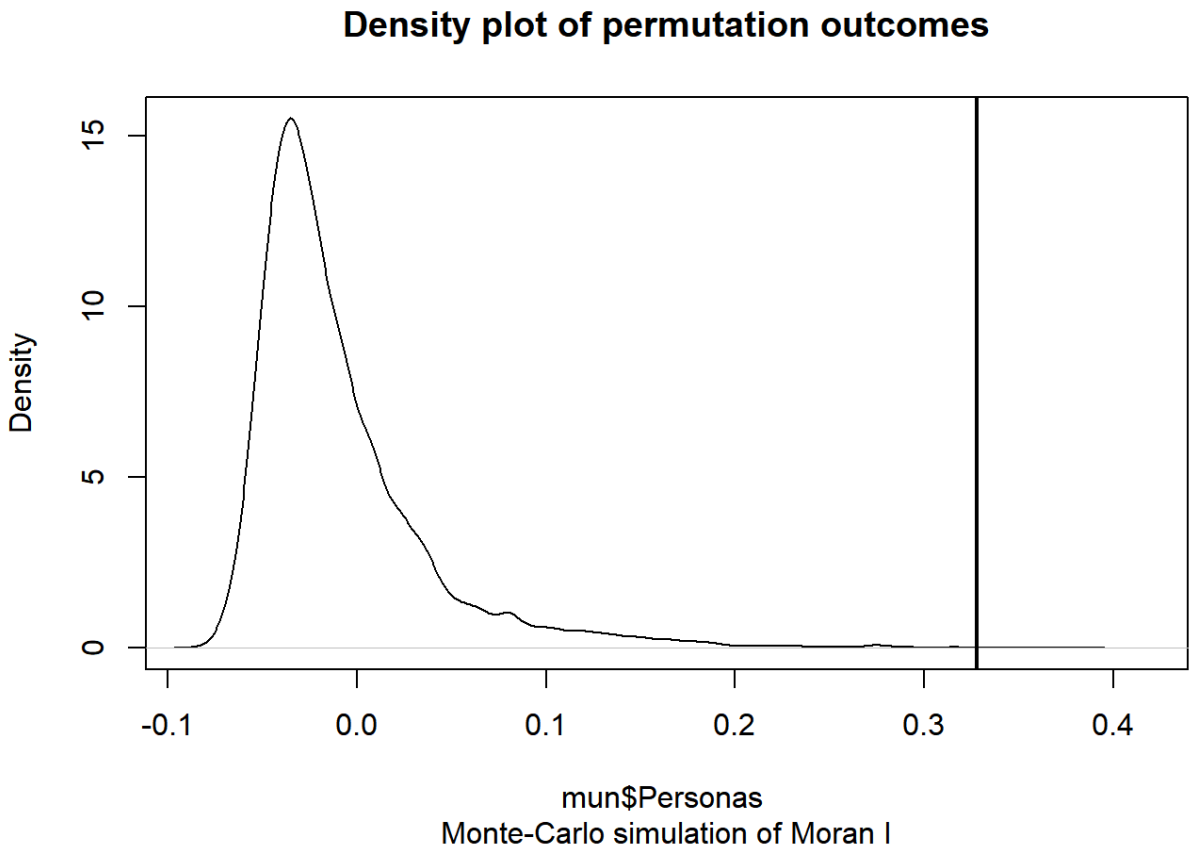


Calcular el indice I de Moran global

```
# Prueba de Moran con simulación de MC
IMglobal <- moran.mc(mun$Personas, mun_lw, nsim=9999, alternative="greater")
IMglobal
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  mun$Personas
## weights: mun_lw
## number of simulations + 1: 10000
##
## statistic = 0.3, observed rank = 9997, p-value = 0.0003
## alternative hypothesis: greater
```

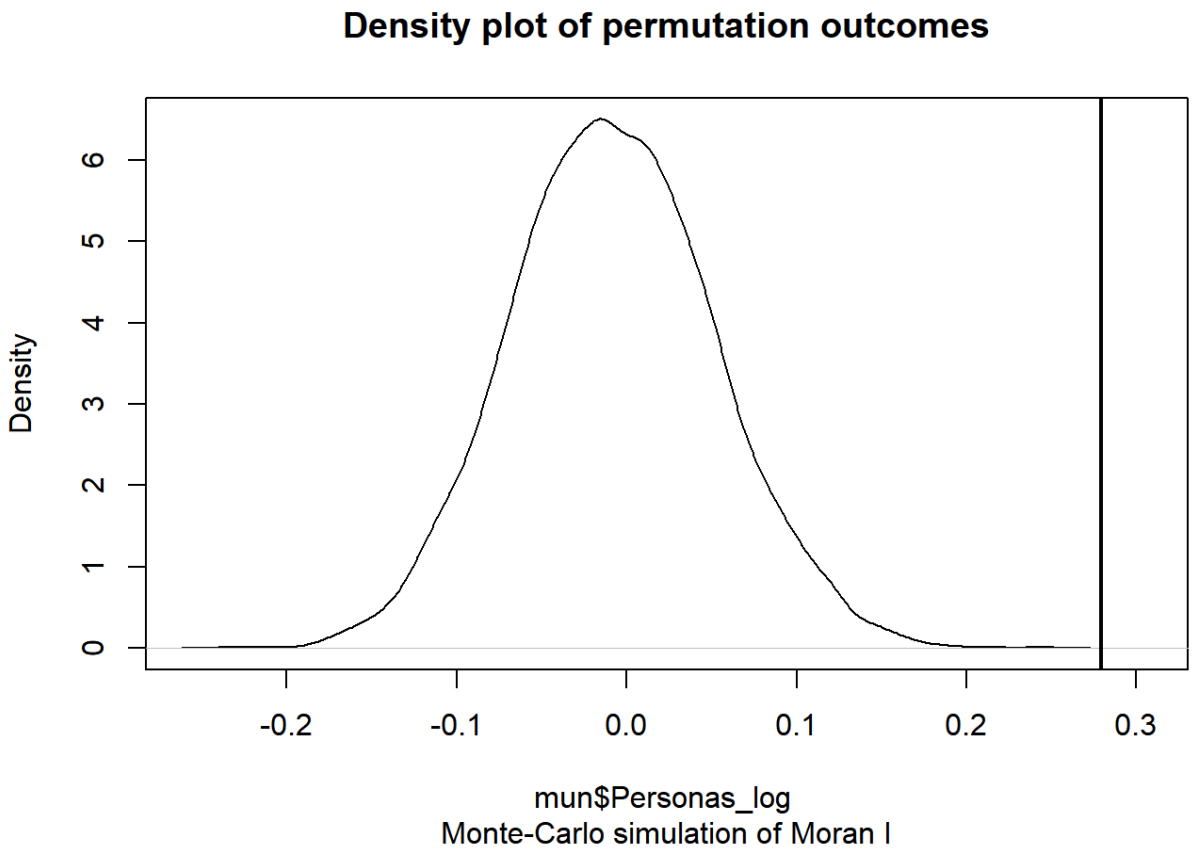
```
plot(IMglobal)
```




```
# Prueba de Moran con simulación de MC con datos logaritmicos
IMglobal_log <- moran.mc(mun$Personas_log, mun_lw, nsim=9999, alternative="greater")
IMglobal_log
```

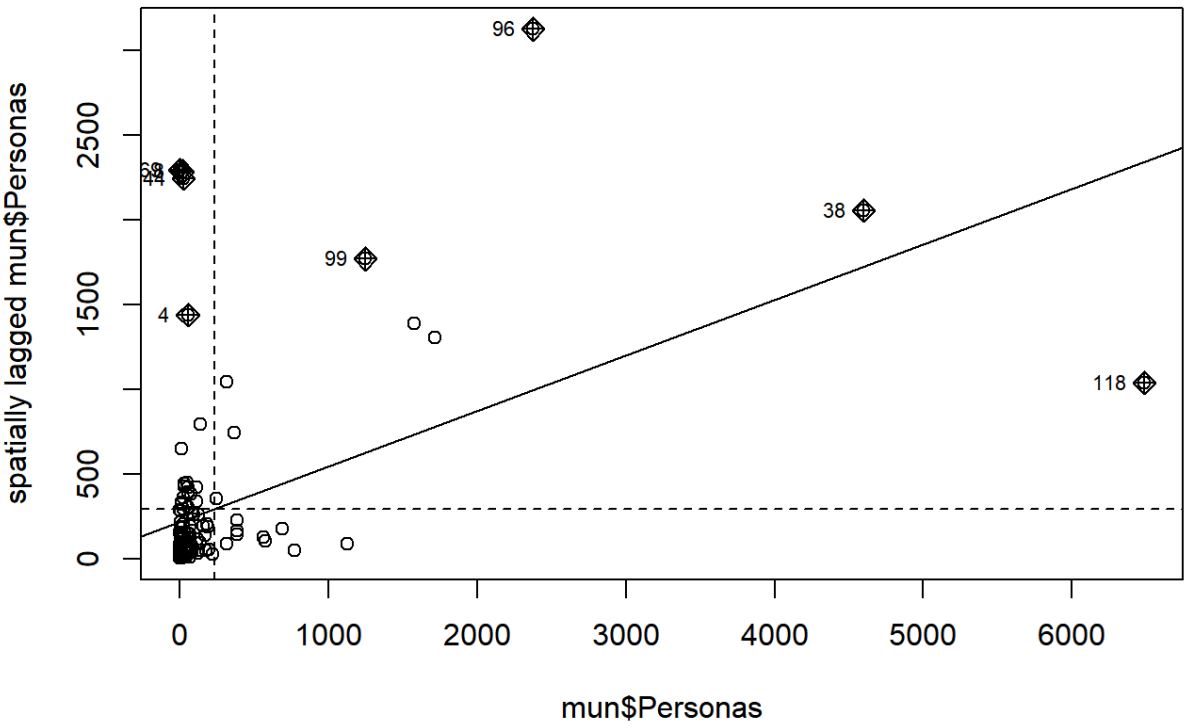
```
##
##  Monte-Carlo simulation of Moran I
##
## data:  mun$Personas_log
## weights: mun_lw
## number of simulations + 1: 10000
##
## statistic = 0.3, observed rank = 10000, p-value = 0.0001
## alternative hypothesis: greater
```

```
plot(IMglobal_log)
```

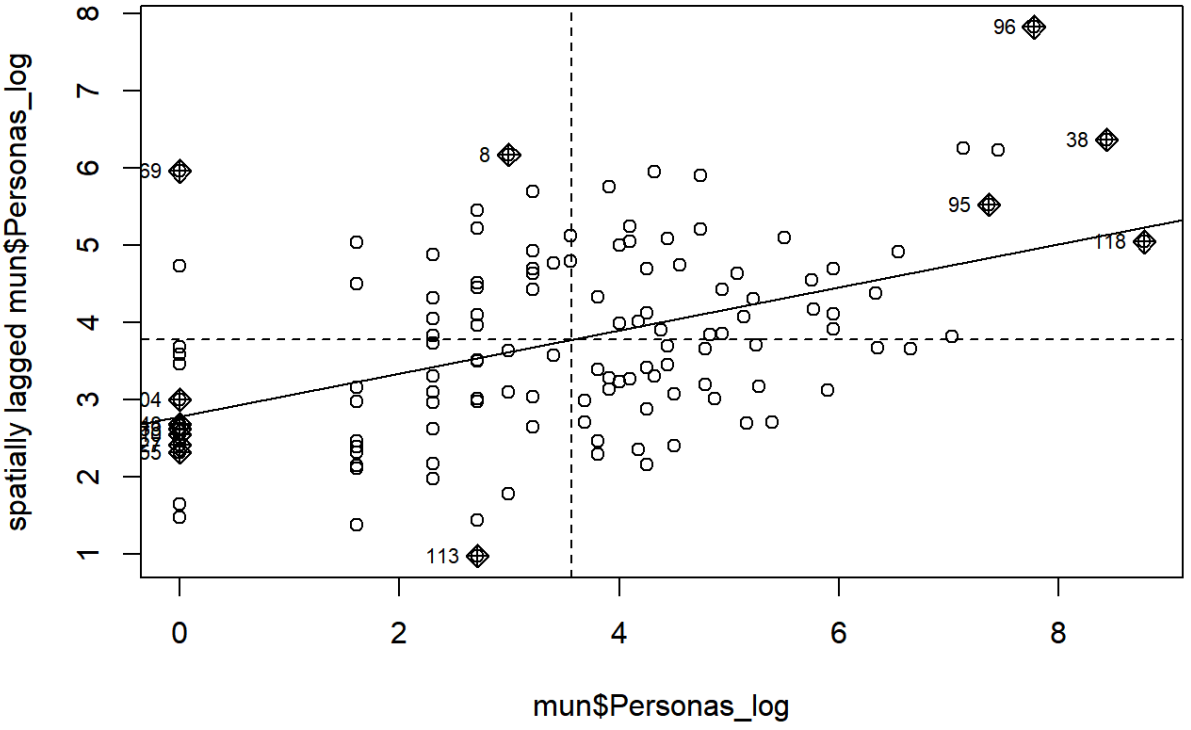


Visualizar grafica de Moran

```
moran.plot(mun$Personas, mun_lw, zero.policy=NULL)
```

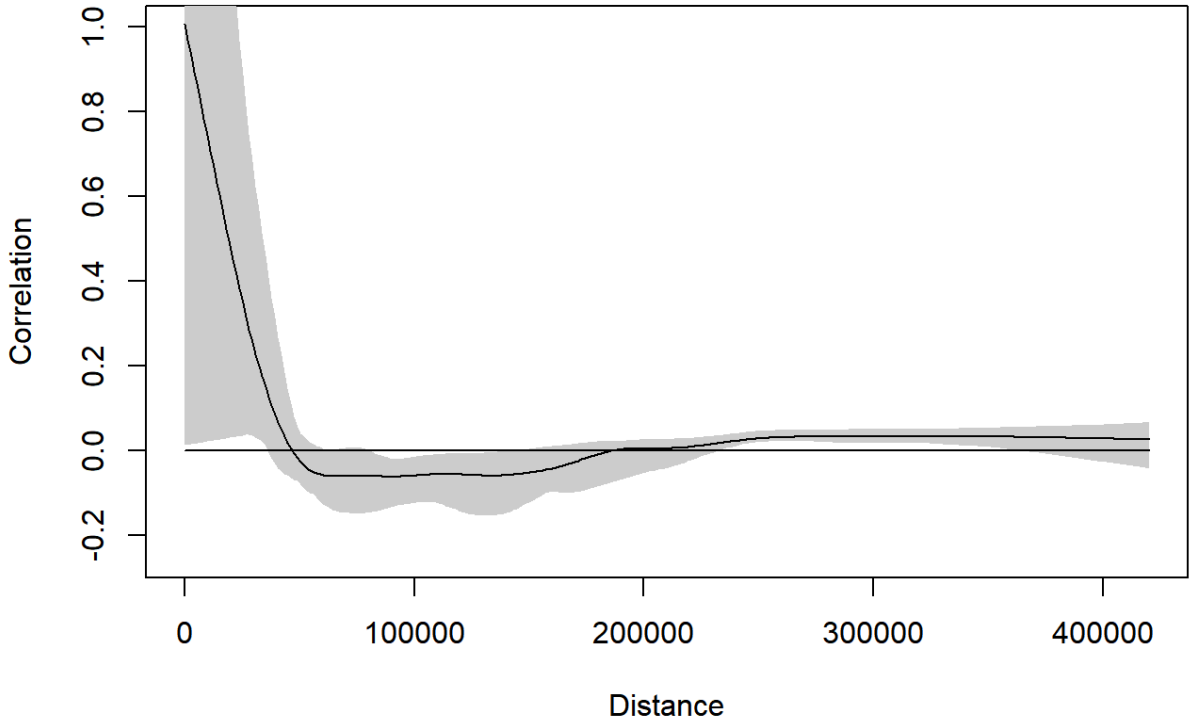



```
moran.plot(mun$Personas_log, mun_lw, zero.policy=NULL)
```



Generar grafica de correlación espacial en función de distancia

```
mun_corr <- spline.correlog(x = coordinates(mun)[,1], y = coordinates(mun)[,2],
                            z = mun$Personas, resamp = 20, quiet = TRUE)
plot(mun_corr, ylim = c(-0.25, 1))
```



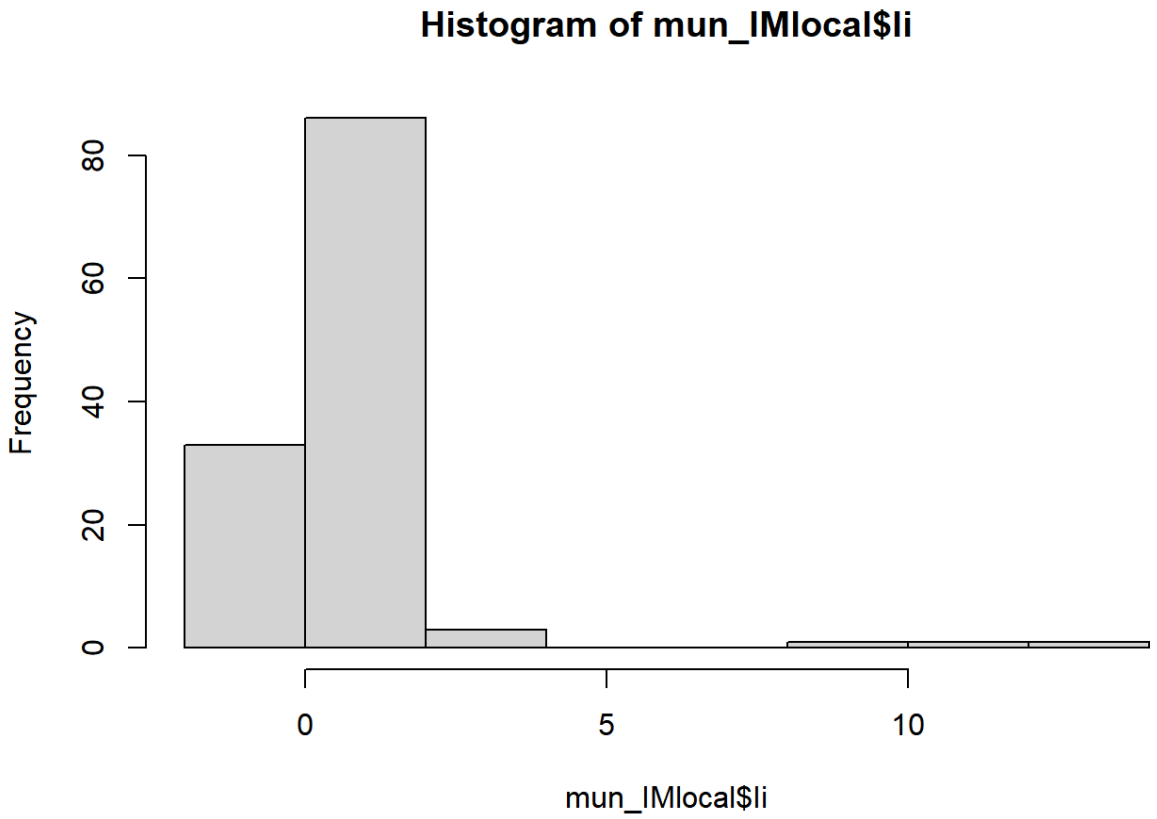
Analisis de Moral Local (analisis LISA)

Con datos sobre número de personas en industria maderera

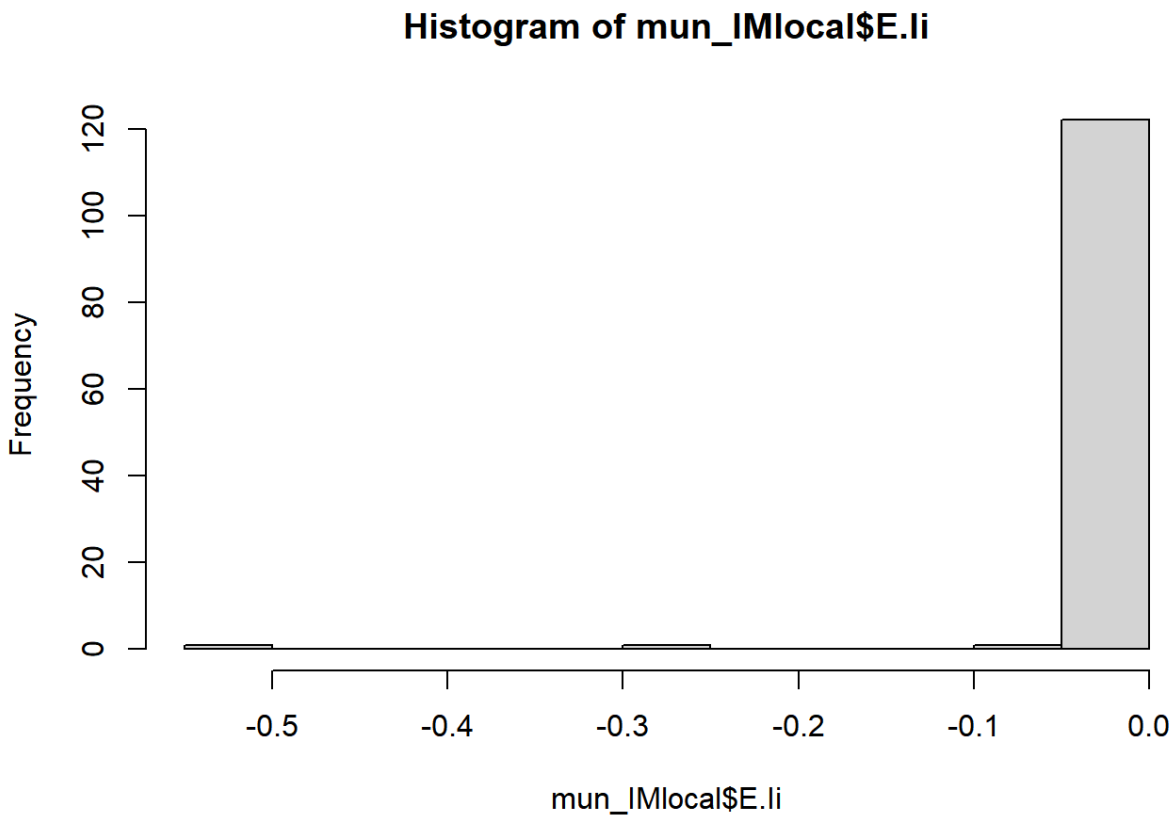
```
IMlocal <- localmoran(mun$Personas, mun_lw, zero.policy=TRUE)
head(IMlocal)
```

```
##        Ii      E.Ii  Var.Ii   Z.Ii Pr(z != E(Ii))
## 0 -0.0572 -0.000436 0.01787 -0.425       0.671109
## 1 -0.0651 -0.000540 0.01304 -0.565       0.571980
## 2  0.0226 -0.000176 0.00426  0.349       0.727055
## 3  0.0616 -0.000654 0.01992  0.441       0.659295
## 4 -0.3657 -0.000436 0.01054 -3.557       0.000376
## 5  0.0320 -0.000146 0.00216  0.692       0.488927
```

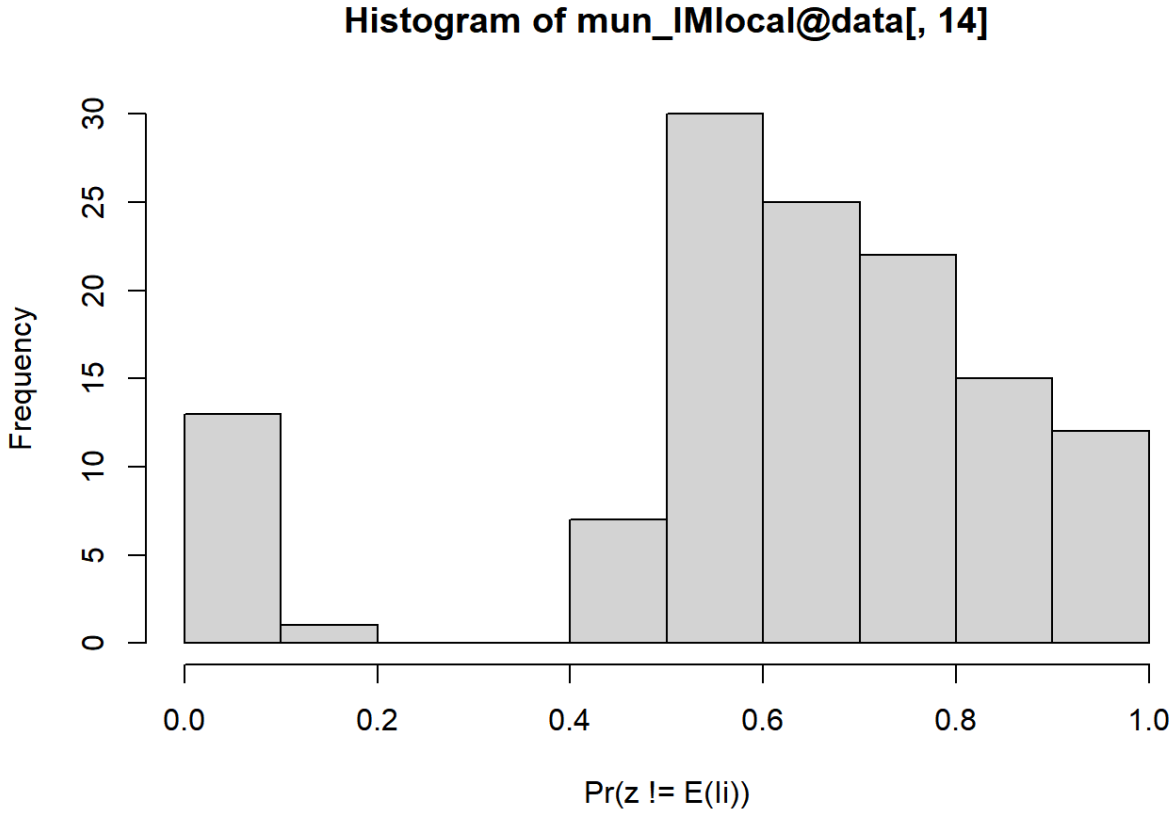
```
mun_IMlocal <- cbind(mun, IMlocal)
#str(mun_IMLocal@data)

hist(mun_IMlocal$Ii)
```




```
hist(mun_IMlocal$E.Ii)
```




```
hist(mun_IMlocal@data[,14], xlab = "Pr(z != E(Ii))")
```

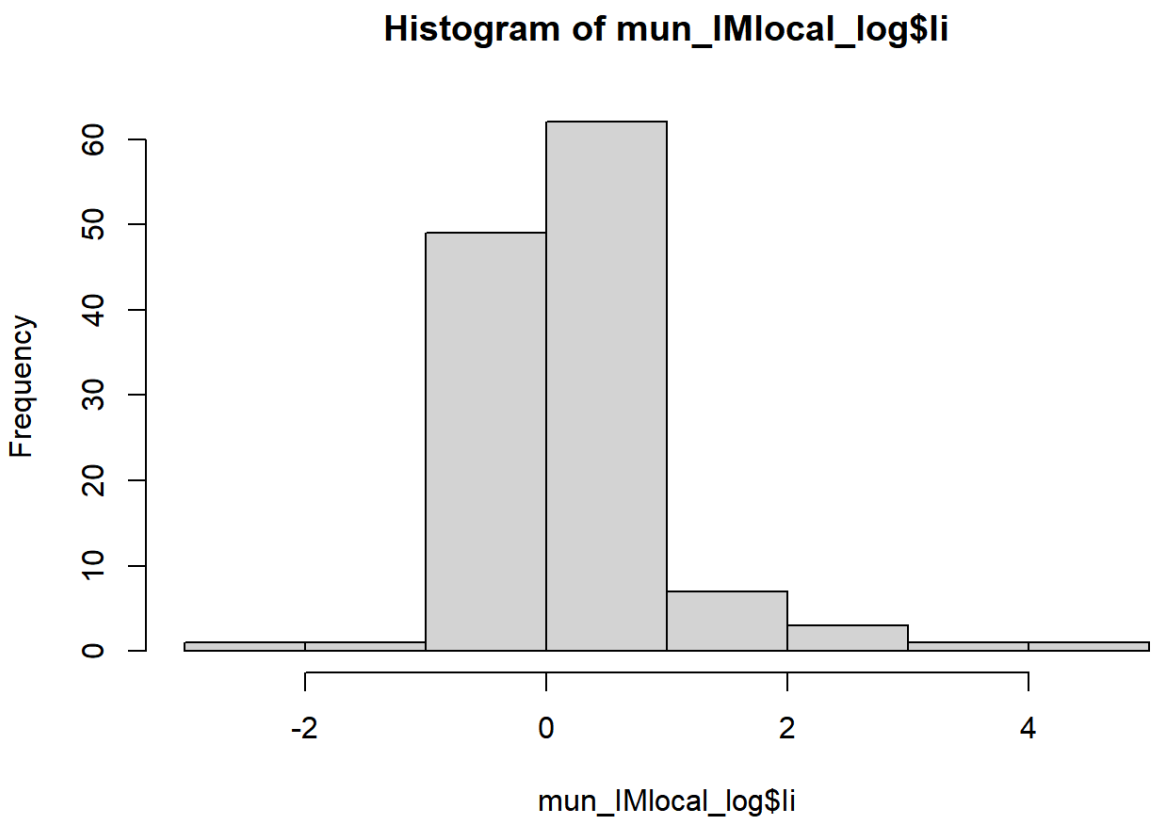


Con datos sobre número de logaritmo de personas en industria maderera

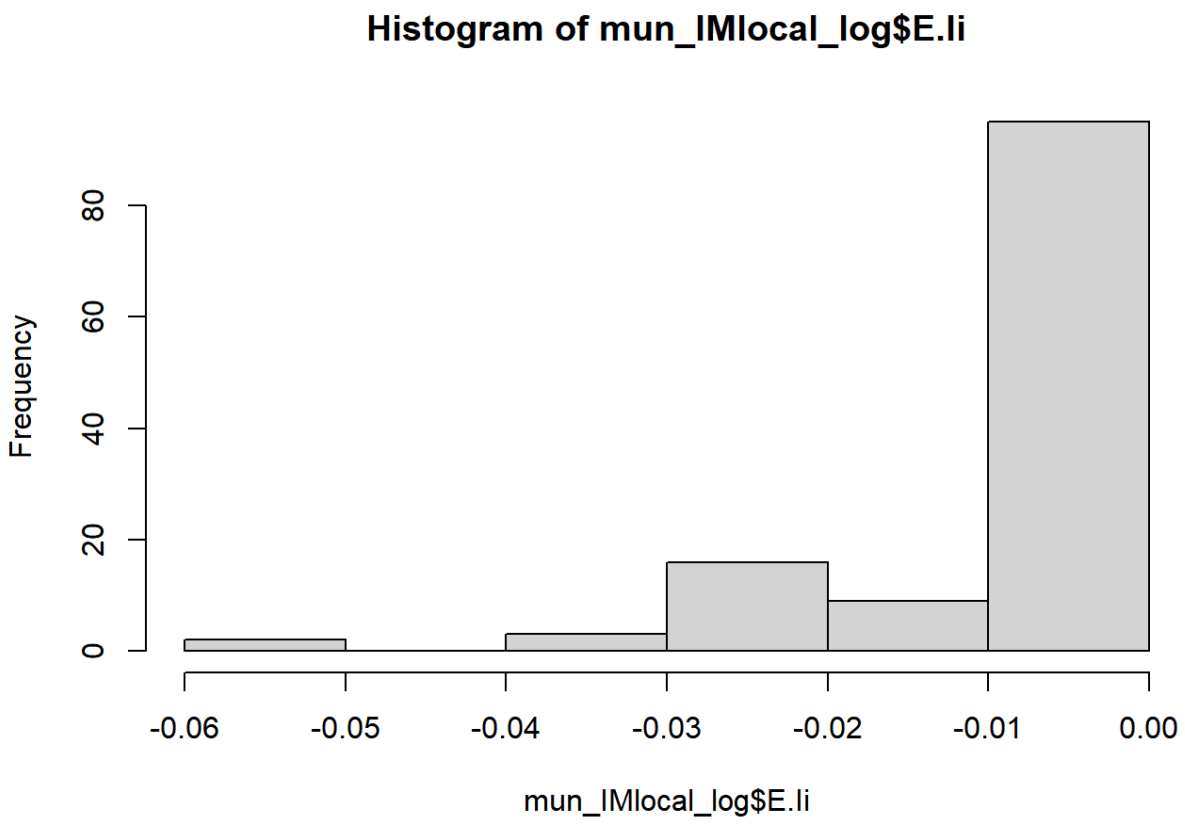
```
IMlocal_log <- localmoran(mun$Personas_log, mun_lw, zero.policy=TRUE)
head(IMlocal_log)
```

```
##        Ii         E.Ii    Var.Ii   Z.Ii Pr(z != E(Ii))
## 0  0.0489 -0.000411562 0.0168626  0.380         0.7041
## 1 -0.0061 -0.000000461 0.0000112 -1.827         0.0677
## 2  0.0295 -0.003189531 0.0768991  0.118         0.9062
## 3  0.1363 -0.001598593 0.0486597  0.625         0.5320
## 4  0.1677 -0.000411562 0.0099503  1.685         0.0920
## 5 -0.1898 -0.003622697 0.0531898 -0.807         0.4194
```

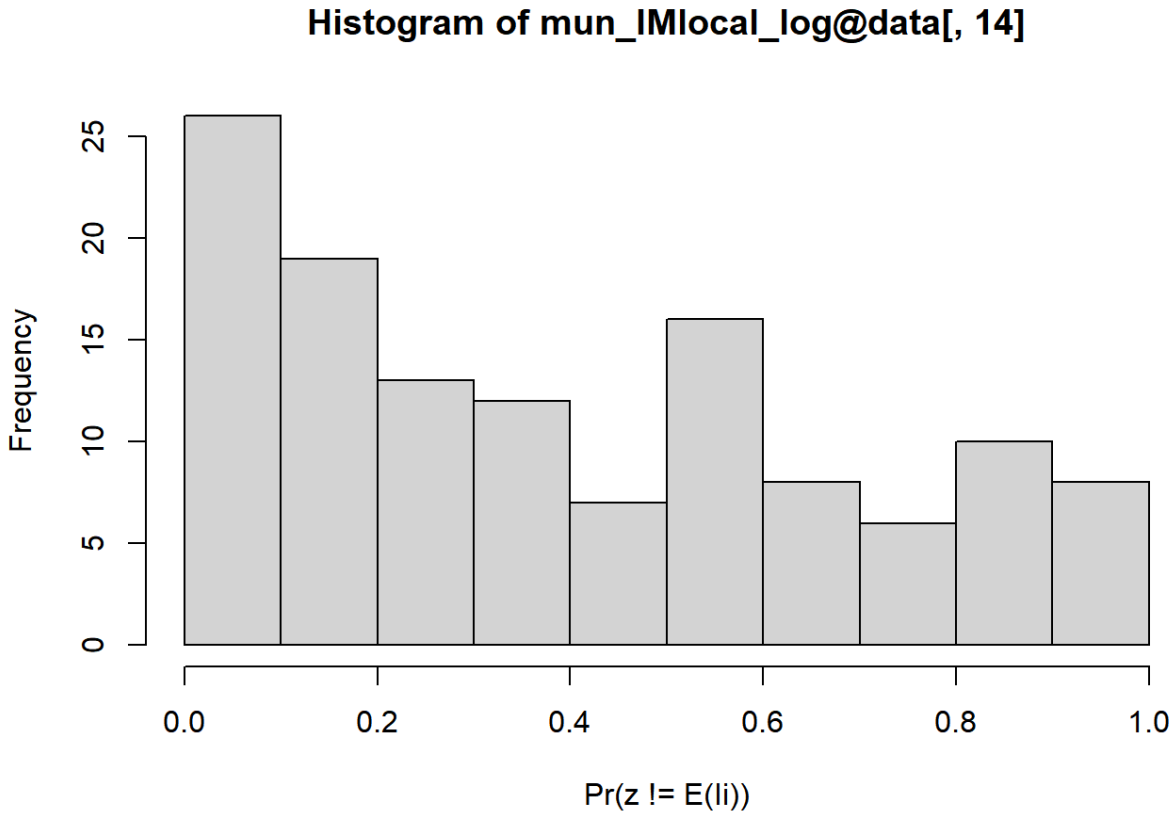
```
mun_IMlocal_log <- cbind(mun, IMlocal_log)
#str(mun_IMLocal_log@data)

hist(mun_IMlocal_log$Ii)
```




```
hist(mun_IMlocal_log$E.Ii)
```

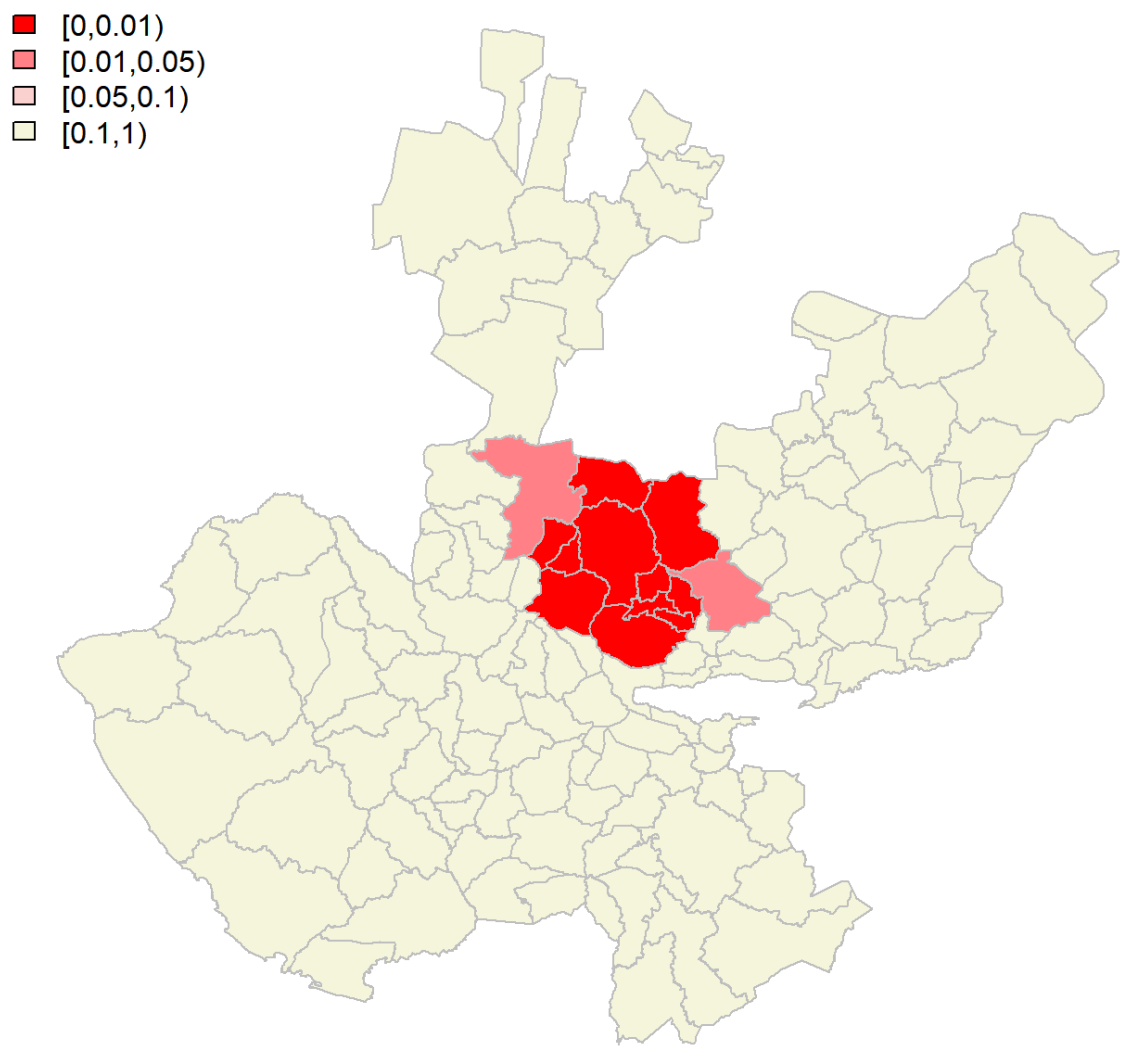



```
hist(mun_IMlocal_log@data[,14], xlab = "Pr(z != E(Ii))")
```

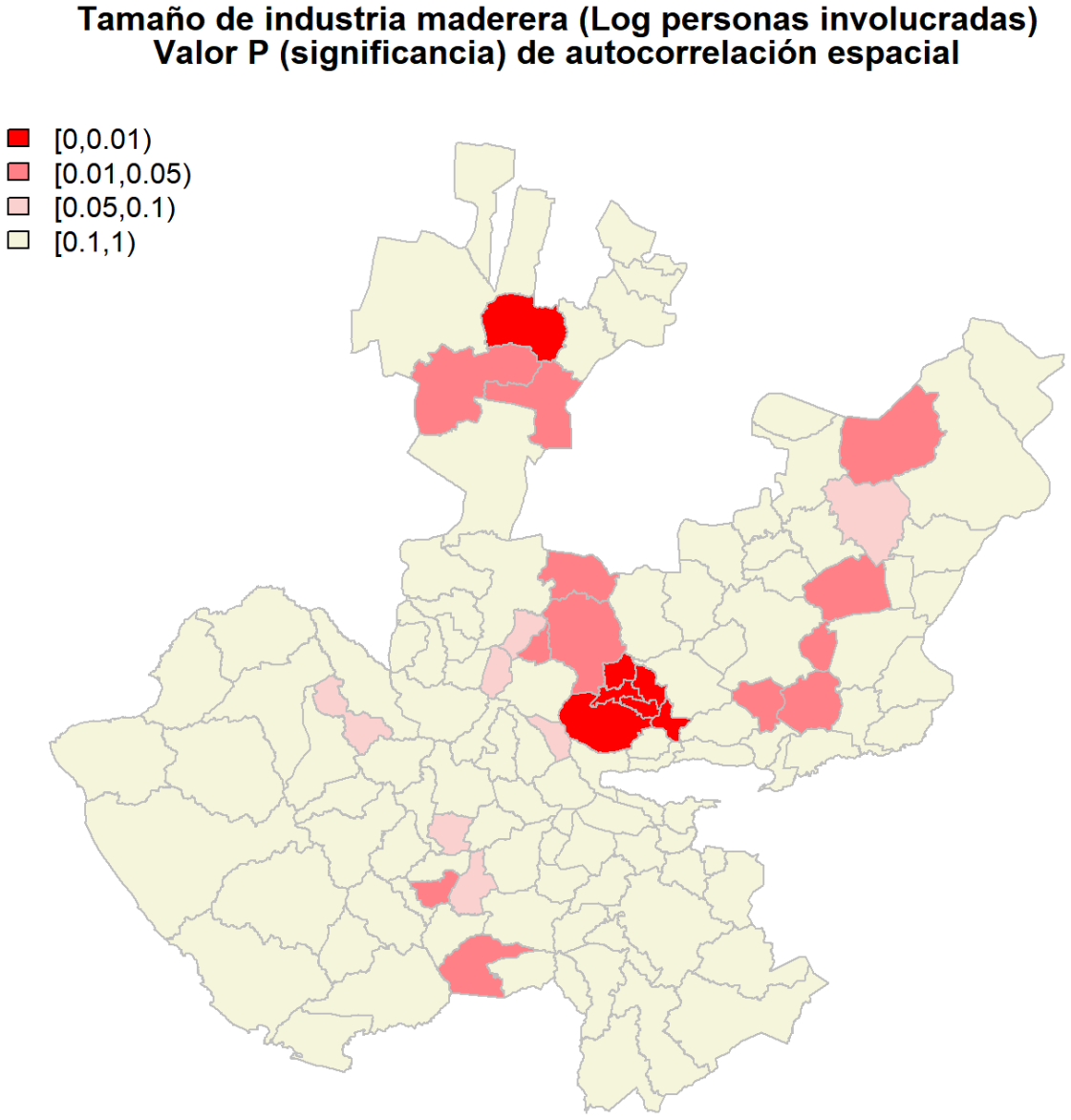


Visualizar valores de significancia en analisis de Moran Local

```
pal3 <- colorRampPalette(c("red","pink","beige"))

## Analisis con variable sin transformar (Personas)
# Variable categorica para colores P (significancia)
mun_IMlocal$P_factor <- cut(mun_IMlocal@data[,14], breaks = c(0,0.01,0.05,0.1,1),
                  right = FALSE)
mun_IMlocal$P_color <- pal3(4)[mun_IMlocal$P_factor]

plot(mun_IMlocal, col = mun_IMlocal$P_color,
    main = c("Tamaño de industria maderera (personas involucradas)","Valor P (significancia) de autoco
rrelación espacial"))
plot(mun, border = "gray", add = TRUE)
legend("topleft", legend = levels(mun_IMlocal$P_factor),
        fill = pal3(4), bty = "n")
```




```
## Analisis con variable logaritmica (Personas_log)
# Variable categorica para colores P (significancia)
mun_IMlocal_log$P_factor <- cut(mun_IMlocal_log@data[,14], breaks = c(0,0.01,0.05,0.1,1),
                  right = FALSE)
mun_IMlocal_log$P_color <- pal3(4)[mun_IMlocal_log$P_factor]

plot(mun_IMlocal_log, col = mun_IMlocal_log$P_color,
    main = c("Tamaño de industria maderera (Log personas involucradas)","Valor P (significancia) de au
tocorrelación espacial"))
plot(mun, border = "gray", add = TRUE)
legend("topleft", legend = levels(mun_IMlocal_log$P_factor),
        fill = pal3(4), bty = "n")
```

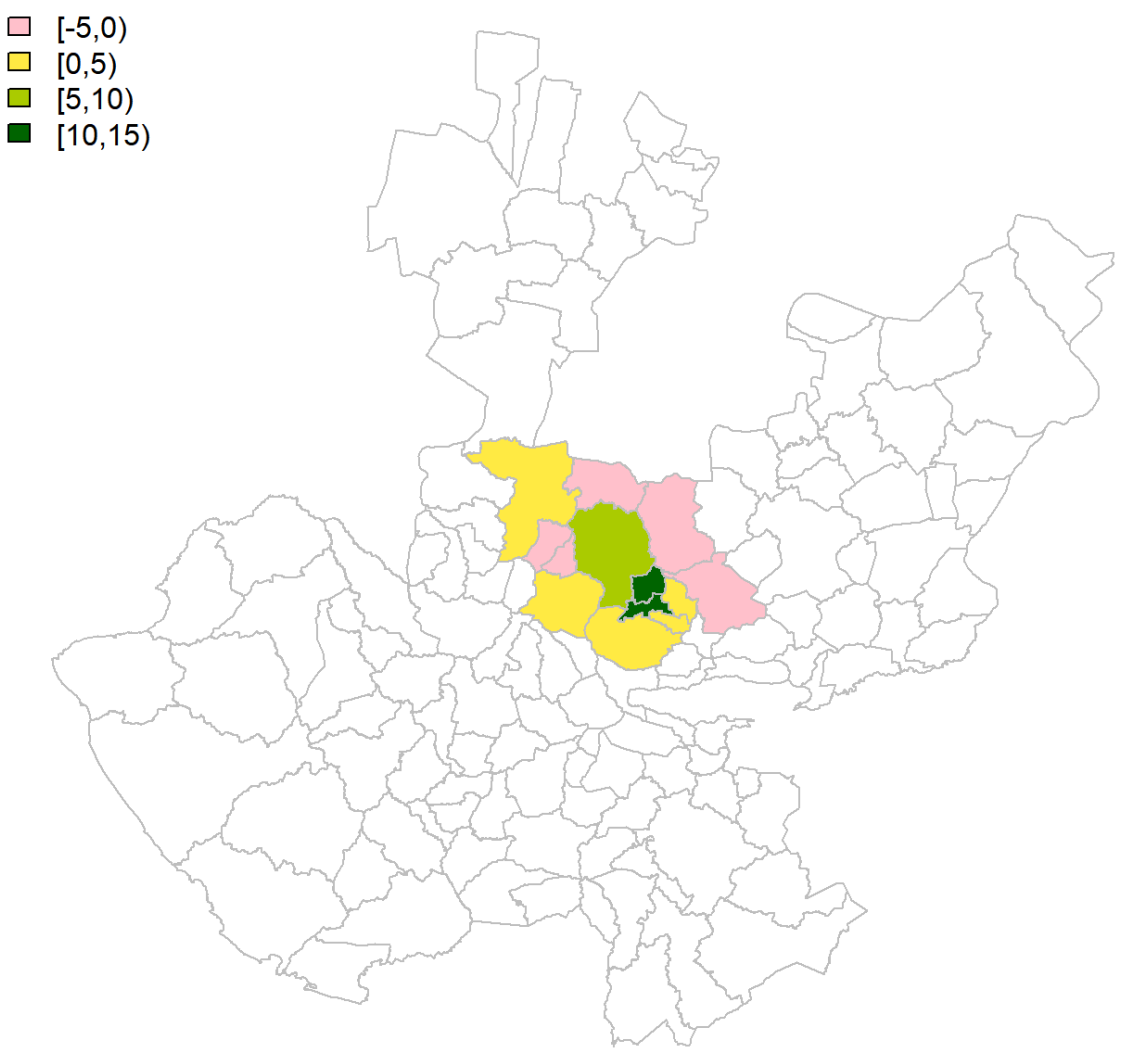


Seleccionar los muncipios con autocorrelación espacial significativa

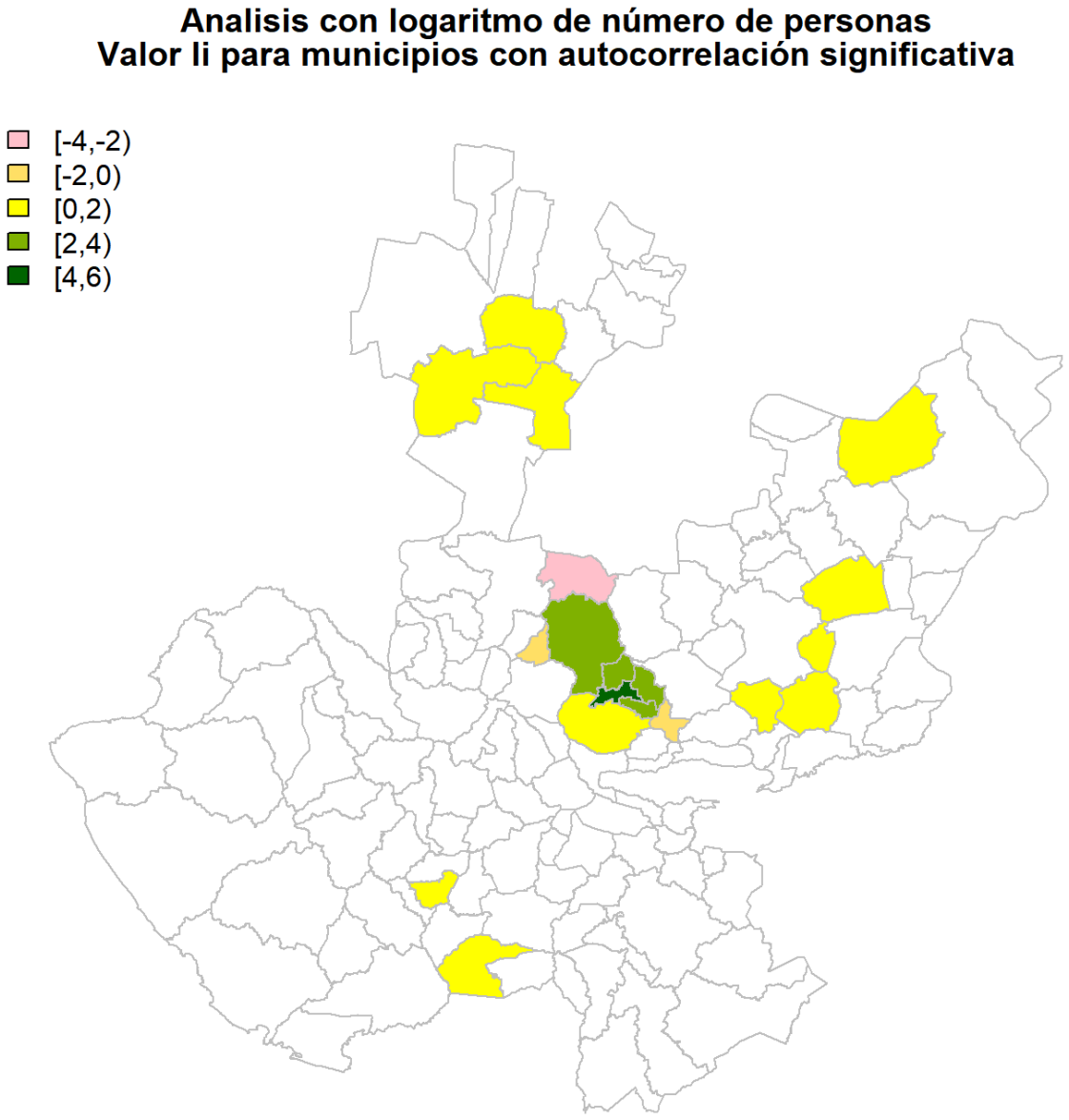
```
mun_IMlocal_sig <- mun_IMlocal[mun_IMlocal@data[,14] <= 0.05,]
mun_IMlocal_log_sig <- mun_IMlocal_log[mun_IMlocal_log@data[,14] <= 0.05,]
```

Visualizar valor de I de Moran local para los municipios con autocorrelación significativa

```
pal2 <- colorRampPalette(c("pink","yellow","darkgreen"))

## Analisis con variable sin transformar (Personas)
# Variable categorica para colores Ii
mun_IMlocal_sig$Ii_factor <- cut(mun_IMlocal_sig$Ii,
                                 breaks = pretty(mun_IMlocal_sig$Ii, n = 4),
                                 right = FALSE)
mun_IMlocal_sig$Ii_color <- pal2(4)[mun_IMlocal_sig$Ii_factor]

plot(mun, border = "gray", main = c("Analisis con número de personas",
       "Valor Ii para municipios con autocorrelación significativa"))
plot(mun_IMlocal_sig, col = mun_IMlocal_sig$Ii_color, add = TRUE)
plot(mun, border = "gray", add = TRUE)
legend("topleft", legend = levels(mun_IMlocal_sig$Ii_factor),
       fill = pal2(4), bty = "n")
```

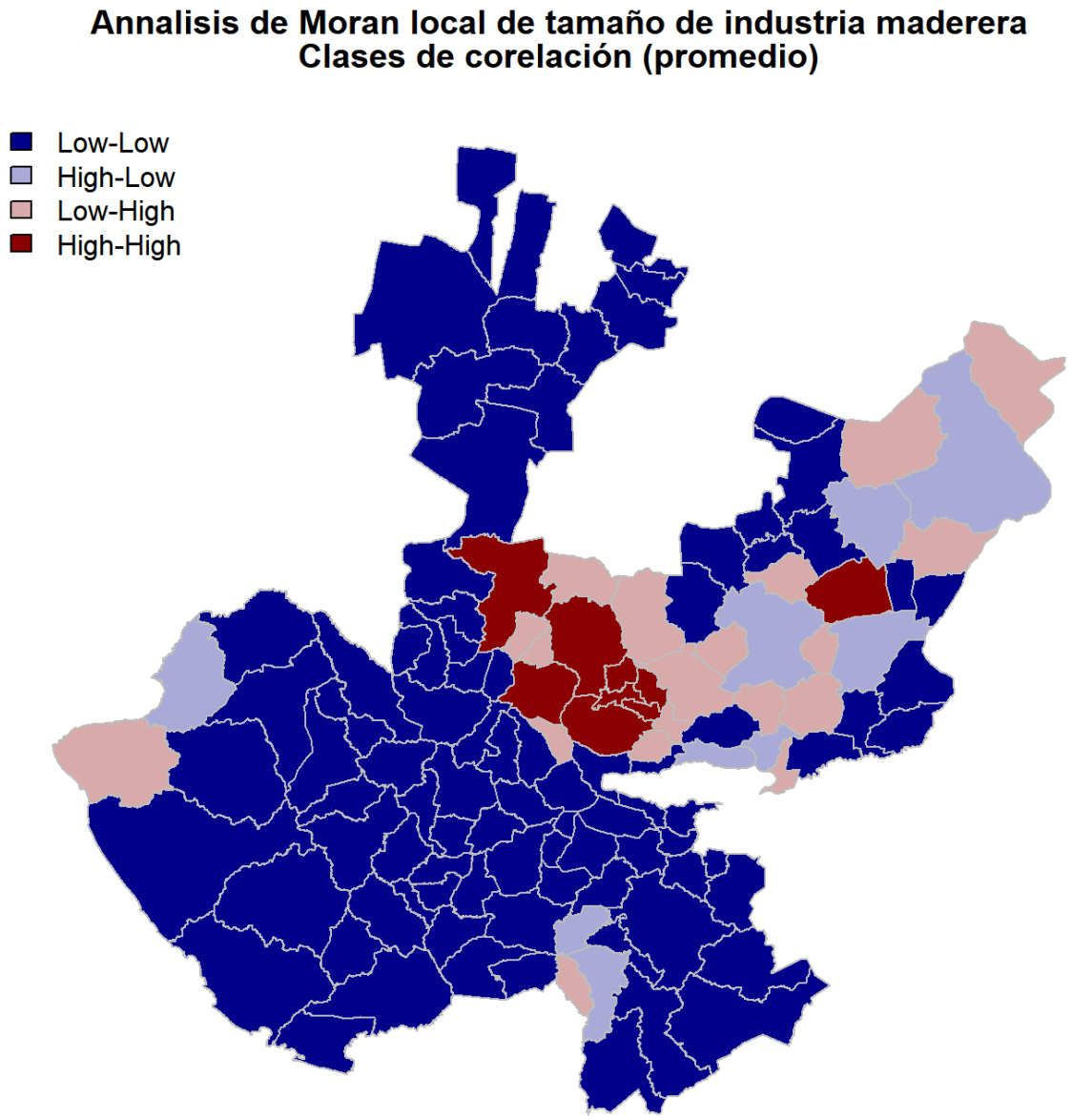



```
## Analisis con variable logaritmica (Personas_log)
# Variable categorica para colores Ii
mun_IMlocal_log_sig$Ii_factor <- cut(mun_IMlocal_log_sig$Ii,
                                     breaks = pretty(mun_IMlocal_log_sig$Ii, n = 4),
                                     right = FALSE)
mun_IMlocal_log_sig$Ii_color <- pal2(5)[mun_IMlocal_log_sig$Ii_factor]

plot(mun, border = "gray", main = c("Analisis con logaritmo de número de personas",
       "Valor Ii para municipios con autocorrelación significativa"))
plot(mun_IMlocal_log_sig, col = mun_IMlocal_log_sig$Ii_color, add = TRUE)
plot(mun, border = "gray", add = TRUE)
legend("topleft", legend = levels(mun_IMlocal_log_sig$Ii_factor),
       fill = pal2(5), bty = "n")
```



Visualizar los clusters de autocirrelación espacial por tipo

```
pal <- colorRampPalette(c("darkblue","white","darkred"))

IMlocal_at <- attr(x = IMlocal, which = "quadr")
mun_IMlocal <- cbind(mun_IMlocal, IMlocal_at)

plot(mun_IMlocal, col = pal(4)[mun_IMlocal$mean],
     main = c("Annalisis de Moran local de tamaño de industria maderera",
              "Clases de corelación (promedio)"))
plot(mun, border = "gray", add = TRUE)
legend("topleft", legend = levels(mun_IMlocal$mean),
       fill = pal(4), bty = "n")
```



```
IMlocal_log_at <- attr(x = IMlocal_log, which = "quadr")
mun_IMlocal_log <- cbind(mun_IMlocal_log, IMlocal_log_at)

plot(mun_IMlocal_log, col = pal(4)[mun_IMlocal_log$mean],
     main = c("Annalisis de Moran local de Log tamaño de industria maderera",
              "Clases de corelación (promedio)"))
plot(mun, border = "gray", add = TRUE)
legend("topleft", legend = levels(mun_IMlocal_log$mean),
       fill = pal(4), bty = "n")
```