Análisis de datos espaciales: precio de la vivienda en Ohio (condado de Lucas)

J. Ramajo

2020

library(spData)  
data(house, package="spData")  
class(house)

## [1] "SpatialPointsDataFrame"  
## attr(,"package")  
## [1] "sp"

summary(house@data)

## price yrbuilt stories TLA   
## Min. : 2000 Min. :1835 one :12954 Min. : 120   
## 1st Qu.: 41900 1st Qu.:1924 bilevel : 509 1st Qu.:1070   
## Median : 65500 Median :1950 multilvl: 723 Median :1318   
## Mean : 79018 Mean :1945 one+half: 3125 Mean :1462   
## 3rd Qu.: 97000 3rd Qu.:1964 two : 8042 3rd Qu.:1682   
## Max. :875000 Max. :1998 two+half: 2 Max. :7616   
## three : 2   
## wall beds baths halfbaths   
## stucdrvt: 204 Min. :0.000 Min. :0.000 Min. :0.0000   
## ccbtile : 129 1st Qu.:3.000 1st Qu.:1.000 1st Qu.:0.0000   
## metlvnyl: 4235 Median :3.000 Median :1.000 Median :0.0000   
## brick : 3633 Mean :2.987 Mean :1.242 Mean :0.3412   
## stone : 86 3rd Qu.:3.000 3rd Qu.:1.000 3rd Qu.:1.0000   
## wood :11174 Max. :9.000 Max. :7.000 Max. :3.0000   
## partbrk : 5896   
## frontage depth garage garagesqft   
## Min. : 0.00 Min. : 0.0 no garage: 3488 Min. : 0.0   
## 1st Qu.: 40.00 1st Qu.: 105.0 basement : 78 1st Qu.: 280.0   
## Median : 50.00 Median : 120.0 attached : 9018 Median : 396.0   
## Mean : 64.27 Mean : 119.1 detached :12555 Mean : 369.5   
## 3rd Qu.: 75.00 3rd Qu.: 140.0 carport : 218 3rd Qu.: 484.0   
## Max. :810.00 Max. :1587.0 Max. :5755.0   
##   
## rooms lotsize sdate avalue   
## Min. : 1.000 Min. : 702 Min. :930104 Min. : 1714   
## 1st Qu.: 5.000 1st Qu.: 4700 1st Qu.:941025 1st Qu.: 38514   
## Median : 6.000 Median : 6800 Median :960520 Median : 59800   
## Mean : 6.115 Mean : 13332 Mean :957704 Mean : 73641   
## 3rd Qu.: 7.000 3rd Qu.: 11100 3rd Qu.:970818 3rd Qu.: 91114   
## Max. :20.000 Max. :429100 Max. :981005 Max. :788114   
##   
## s1993 s1994 s1995 s1996   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :0.0000 Median :0.0000 Median :0.0000   
## Mean :0.1286 Mean :0.1467 Mean :0.1629 Mean :0.1908   
## 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
##   
## s1997 s1998 syear age   
## Min. :0.0000 Min. :0.0000 1993:3260 Min. :0.0100   
## 1st Qu.:0.0000 1st Qu.:0.0000 1994:3719 1st Qu.:0.3500   
## Median :0.0000 Median :0.0000 1995:4130 Median :0.4900   
## Mean :0.1984 Mean :0.1727 1996:4838 Mean :0.5366   
## 3rd Qu.:0.0000 3rd Qu.:0.0000 1997:5032 3rd Qu.:0.7500   
## Max. :1.0000 Max. :1.0000 1998:4378 Max. :1.6400   
##

# writeOGR(house, layer = "HOUSE\_PRICES", driver="ESRI Shapefile", dsn = "~/Documentos/R/EcMetrics\_Ramajo/HOUSPR\_OHIO")  
#  
library(sf)

## Linking to GEOS 3.8.1, GDAL 3.1.1, PROJ 6.3.1

library(leaflet)   
library(ggplot2)  
library(maptools)

## Loading required package: sp

## Checking rgeos availability: TRUE

library(RColorBrewer)   
library(classInt)  
library(rgdal)

## rgdal: version: 1.5-12, (SVN revision 1018)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 3.1.1, released 2020/06/22  
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/4.0/Resources/library/rgdal/gdal  
## GDAL binary built with GEOS: TRUE   
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ\_VERSION: 631]  
## Path to PROJ shared files: /Library/Frameworks/R.framework/Versions/4.0/Resources/library/rgdal/proj  
## Linking to sp version:1.4-2  
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,  
## use options("rgdal\_show\_exportToProj4\_warnings"="none") before loading rgdal.

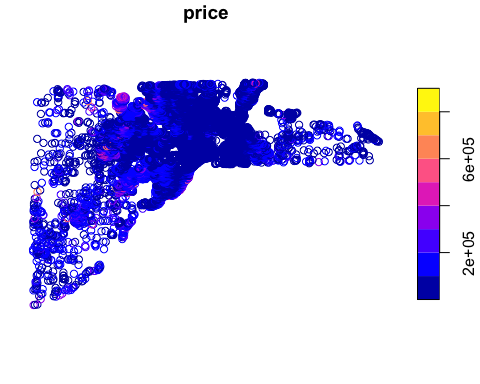
library(viridis)

## Loading required package: viridisLite

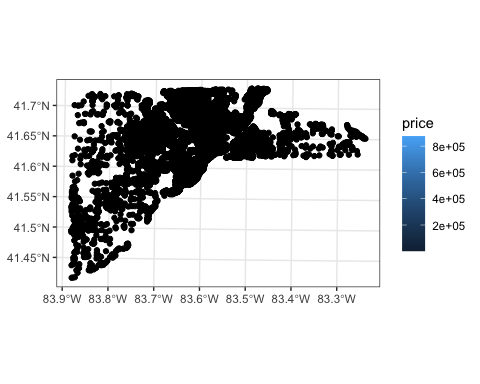
#  
# Representación de datos espaciales  
#  
# Convirtiendo objeto sp a sf  
house\_sf <- st\_as\_sf(house)  
class(house\_sf)

## [1] "sf" "data.frame"

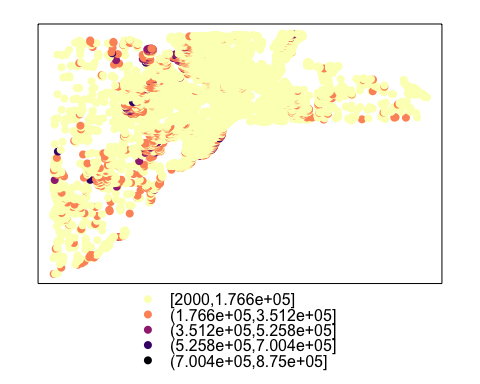
plot(house\_sf[1])



ggplot(house\_sf) + geom\_sf(aes(fill=price))+ theme\_bw()



spplot(house, "price", col.regions = rev(magma(10)))



# leaflet(house\_sf)

#  
# Indicadores de asociación espacial (global y local)  
#  
library(spdep)   
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Información sobre las coordenadas (longitud y latitud) de los datos  
coords <- coordinates(house)  
# Matrices de pesos espaciales  
# Vecinos más próximos (k=6 mediana de vecinos en la base de datos)  
house.6nn <- knearneigh(coords, k=6)   
house.6nn.nb <- knn2nb(house.6nn)  
# Distancia Euclidiana máxima entre los centroides  
# rn <- row.names(house\_sf)  
# house.1nn.nb <- knn2nb(knearneigh(coords, k=1))  
# max.dist <- max(unlist(nbdists(house.1nn.nb, coords)))  
# house.dist.nb <- dnearneigh(coords, d1=0, d2=max.dist, row.names=rn, longlat =T)   
# Datos para la matriz de distancias inversas al cuadrado posterior  
# dlist <- nbdists(house.dist.nb,coords)   
# id2list <- lapply(dlist,function(x) 1/x^2)  
#  
# Gráficos de vecinos  
#  
plot(st\_geometry(house\_sf))   
plot(house.6nn.nb, coords, add=TRUE, col="red")



#  
#plot(st\_geometry(house\_sf))   
# plot(house.dist.nb, coords, add=TRUE, col="red")  
# Cálculo de las matrices de pesos W (estandarizadas por filas)  
house.6nn.w <- nb2listw(neighbours=house.6nn.nb, style="W")  
# house.id2.w <- nb2listw(neighbours=house.dist.nb,glist=id2list,style="W")  
# house.d.w <- nb2listw(neighbours=house.dist.nb, style="W")  
#  
# Estadísticos de autocorrelación espacial  
#  
# Global  
moran.test(house\_sf$price, listw=house.6nn.w)

##   
## Moran I test under randomisation  
##   
## data: house\_sf$price   
## weights: house.6nn.w   
##   
## Moran I statistic standard deviate = 226.39, p-value < 2.2e-16  
## alternative hypothesis: greater  
## sample estimates:  
## Moran I statistic Expectation Variance   
## 7.765886e-01 -3.943840e-05 1.176791e-05

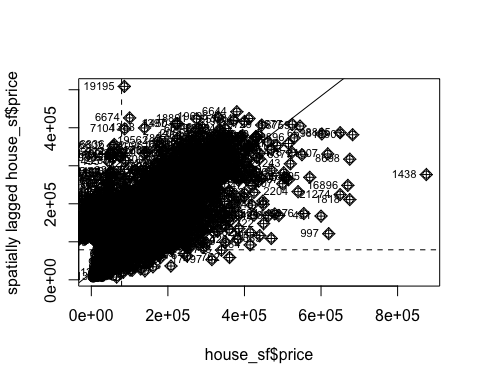
geary.test(house\_sf$price, listw=house.6nn.w)

##   
## Geary C test under randomisation  
##   
## data: house\_sf$price   
## weights: house.6nn.w   
##   
## Geary C statistic standard deviate = 149.55, p-value < 2.2e-16  
## alternative hypothesis: Expectation greater than statistic  
## sample estimates:  
## Geary C statistic Expectation Variance   
## 2.164137e-01 1.000000e+00 2.745408e-05

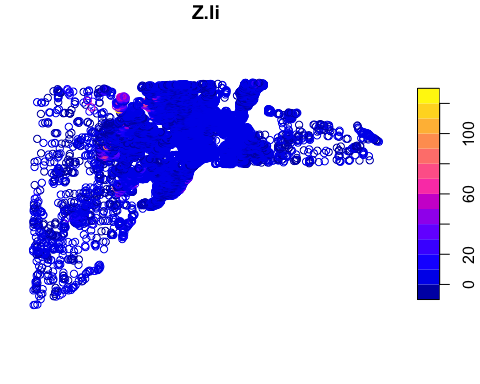
# Local  
LocalI <- as.data.frame(localmoran(house\_sf$price, listw=house.6nn.w))  
str(LocalI)

## 'data.frame': 25357 obs. of 5 variables:  
## $ Ii : num 3.324 0.321 0.273 3.408 0.921 ...  
## $ E.Ii : num -3.94e-05 -3.94e-05 -3.94e-05 -3.94e-05 -3.94e-05 ...  
## $ Var.Ii : num 0.167 0.167 0.167 0.167 0.167 ...  
## $ Z.Ii : num 8.146 0.787 0.668 8.35 2.256 ...  
## $ Pr(z > 0): num 1.89e-16 2.16e-01 2.52e-01 3.42e-17 1.20e-02 ...

moran.plot(house\_sf$price, listw=house.6nn.w)



# Clusters locales  
house\_LocalI\_sf <- bind\_cols(house\_sf,LocalI) #  
plot(house\_LocalI\_sf["Z.Ii"])



#  
# Modelos econométricos espaciales  
#  
library(spatialreg)

form <- formula(log(price) ~ age + log(lotsize) + rooms)  
# Modelo lineal (LM) estimado por MCO  
model.LS <- lm(formula=form, data=house\_sf)  
summary(model.LS)

##   
## Call:  
## lm(formula = form, data = house\_sf)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.2148 -0.2090 0.0662 0.2914 2.8415   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.793591 0.042969 204.65 <2e-16 \*\*\*  
## age -1.475724 0.012249 -120.47 <2e-16 \*\*\*  
## log(lotsize) 0.244074 0.004395 55.53 <2e-16 \*\*\*  
## rooms 0.135360 0.002383 56.79 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4791 on 25353 degrees of freedom  
## Multiple R-squared: 0.6057, Adjusted R-squared: 0.6057   
## F-statistic: 1.298e+04 on 3 and 25353 DF, p-value: < 2.2e-16

#  
# Modelo con retardo espacial (SLM)  
#  
# Estimación S2SLS  
model.SLM.STSLS <- stsls(formula=form, data=house\_sf, listw=house.6nn.w)  
summary(model.SLM.STSLS)

##   
## Call:stsls(formula = form, data = house\_sf, listw = house.6nn.w)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.477141 -0.140196 0.041652 0.203692 2.416681   
##   
## Coefficients:   
## Estimate Std. Error t value Pr(>|t|)  
## Rho 0.5224030 0.0085411 61.163 < 2.2e-16  
## (Intercept) 4.0596763 0.0832439 48.768 < 2.2e-16  
## age -0.7498412 0.0147366 -50.883 < 2.2e-16  
## log(lotsize) 0.1155989 0.0037733 30.636 < 2.2e-16  
## rooms 0.0929439 0.0018359 50.626 < 2.2e-16  
##   
## Residual variance (sigma squared): 0.11674, (sigma: 0.34167)

# Estimación ML  
# model.SLM.ML <- lagsarlm (formula=form, data=house\_sf, listw=house.6nn.w, method="eigen")  
# summary(model.SLM.ML)  
# Especificación MESS (matrix exponential spatial speciﬁcation)  
# model.SLM.MESS <- lagmess(formula=form, data=house\_sf, listw=house.6nn.w)   
# summary(model.SLM.MESS)  
#  
# Modelo con errores espaciales (SEM)  
#  
# Estimación GMM  
model.SEM.GMM <- GMerrorsar(formula=form, data=house\_sf, listw=house.6nn.w)  
summary(model.SEM.GMM)

##   
## Call:GMerrorsar(formula = form, data = house\_sf, listw = house.6nn.w)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.15054 -0.20445 0.10883 0.33476 2.54146   
##   
## Type: GM SAR estimator  
## Coefficients: (GM standard errors)   
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 8.8091372 0.0483159 182.324 < 2.2e-16  
## age -0.8666105 0.0137436 -63.056 < 2.2e-16  
## log(lotsize) 0.2269016 0.0051107 44.398 < 2.2e-16  
## rooms 0.1031042 0.0020170 51.118 < 2.2e-16  
##   
## Lambda: 0.69144 (standard error): 0.011157 (z-value): 61.975  
## Residual variance (sigma squared): 0.11859, (sigma: 0.34437)  
## GM argmin sigma squared: 0.11893  
## Number of observations: 25357   
## Number of parameters estimated: 6

# Estimación ML  
# model.SEM.ML <- errorsarlm (formula=form, data=house\_sf, listw=house.6nn.w, method="eigen")  
# summary(model.SEM.ML)  
#  
# Modelo combinado (SAC -> SLM + SEM)  
#  
# Estimación GS2SLS  
model.SAC.GSTSLS <- gstsls(formula=form, data=house\_sf, listw=house.6nn.w)  
summary(model.SAC.GSTSLS)

##   
## Call:gstsls(formula = form, data = house\_sf, listw = house.6nn.w)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.38234 -0.13633 0.03348 0.18794 2.53190   
##   
## Type: GM SARAR estimator  
## Coefficients: (GM standard errors)   
## Estimate Std. Error z value Pr(>|z|)  
## Rho\_Wy 0.5164497 0.0089487 57.712 < 2.2e-16  
## (Intercept) 3.9106907 0.0895989 43.647 < 2.2e-16  
## age -0.7049521 0.0141052 -49.978 < 2.2e-16  
## log(lotsize) 0.1338697 0.0043946 30.462 < 2.2e-16  
## rooms 0.0971244 0.0018576 52.284 < 2.2e-16  
##   
## Lambda: 0.36055  
## Residual variance (sigma squared): 0.10676, (sigma: 0.32675)  
## GM argmin sigma squared: 0.10716  
## Number of observations: 25357   
## Number of parameters estimated: 7