R: Interoperability

Roger Bivand

Thursday, 5 September 2019, 16:18-16:30

Use R for comparison

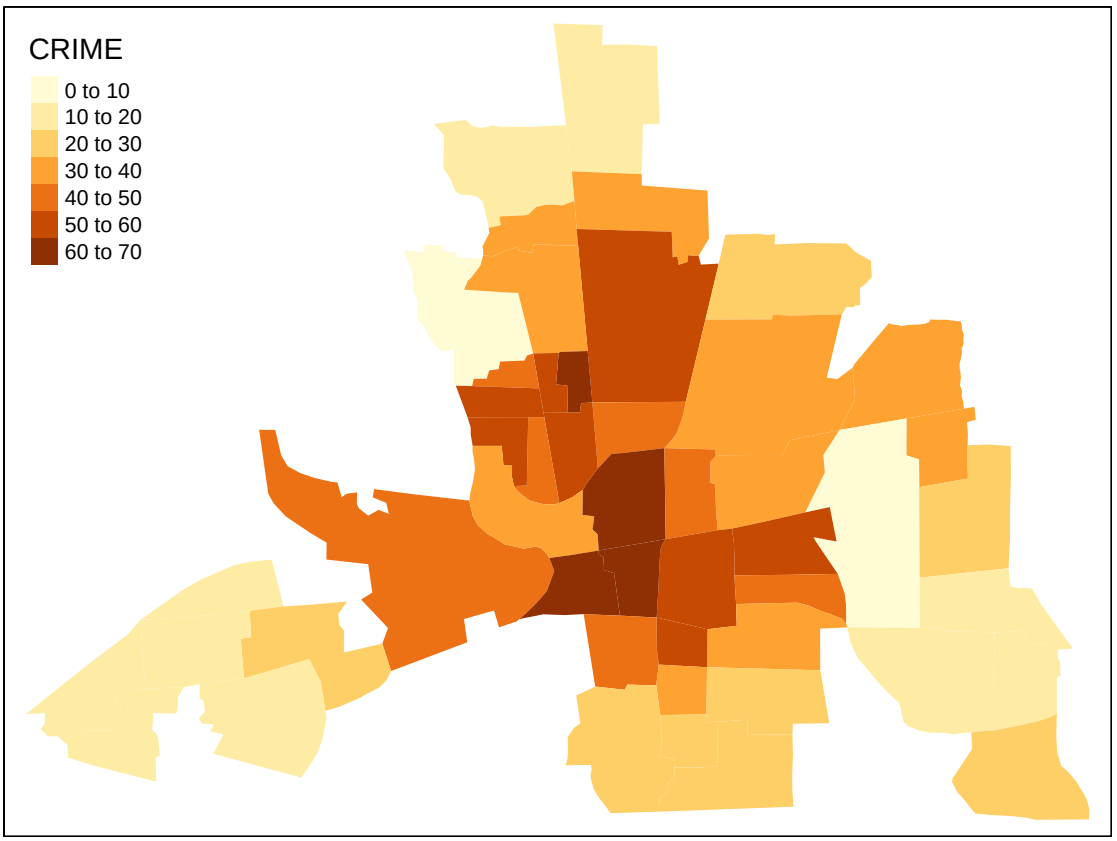
```
library(sf)
```

```
## Linking to GEOS 3.7.2, GDAL 3.0.1, PROJ 6.2.0
```

```
col_file <- system.file("shapes/columbus.shp", package="spData")[1]
col_sf <- st_read(col_file, quiet=TRUE)
names(col_sf)
```

```
##  [1] "AREA"       "PERIMETER"  "COLUMBUS_"  "COLUMBUS_I" "POLYID"
##  [6] "NEIG"       "HOVAL"      "INC"        "CRIME"      "OPEN"
## [11] "PLUMB"      "DISCBD"     "X"          "Y"          "NSA"
## [16] "NSB"        "EW"         "CP"         "THOUS"      "NEIGNO"
## [21] "geometry"
```

```
library(tmap)
tm_shape(col_sf) + tm_fill("CRIME", style="pretty")
```



Create spatial weights

```
library(spdep)
```

```
## Loading required package: sp
```

```
## Loading required package: spData
```

```
nb_sf <- poly2nb(col_sf)
nb_sf
```

```
## Neighbour list object:
## Number of regions: 49
## Number of nonzero links: 236
## Percentage nonzero weights: 9.829238
## Average number of links: 4.816327
```

```
lw <- nb2listw(nb_sf, style="W")
```

ESDA

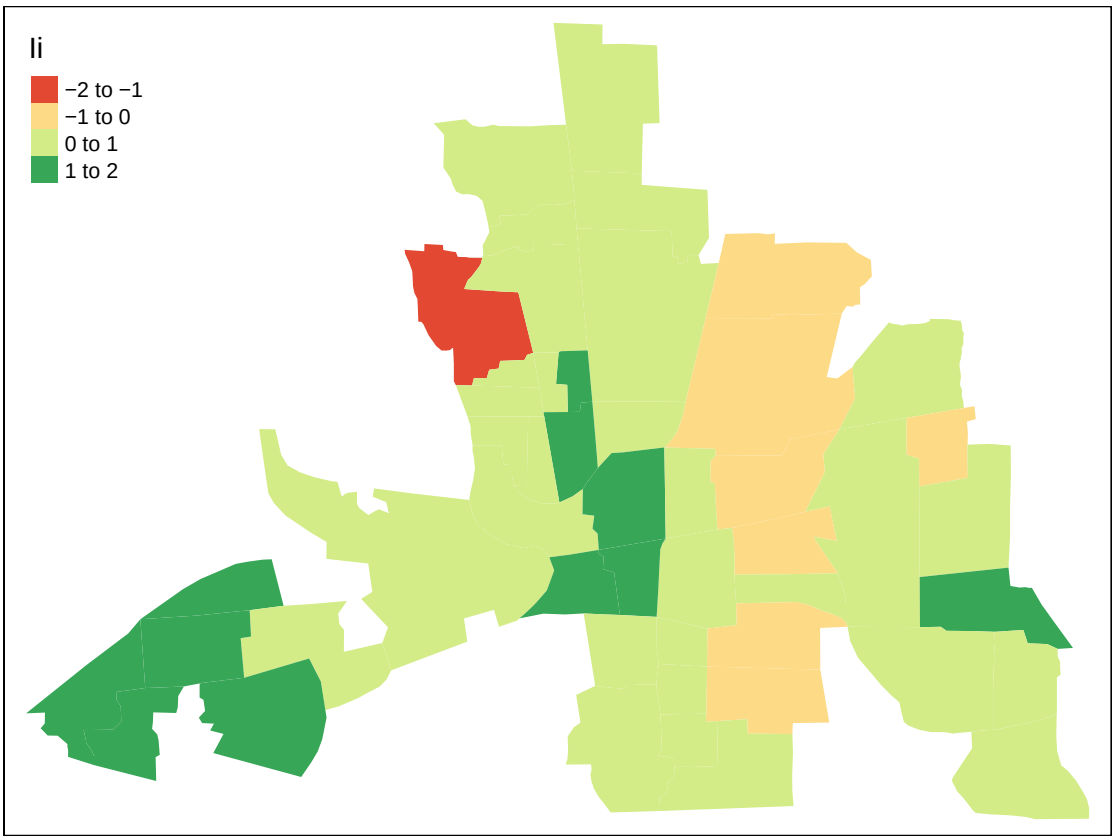
```
moran.test(col_sf$CRIME, lw)
```

```
##
##  Moran I test under randomisation
##
## data:  col_sf$CRIME
## weights: lw
##
## Moran I statistic standard deviate = 5.5894, p-value = 1.139e-08
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.500188557      -0.020833333       0.008689289
```

```
loc_I_sf <- localmoran(col_sf$CRIME, lw)
sum(loc_I_sf[,"Ii"])/Szero(lw)
```

```
## [1] 0.5001886
```

```
col_sf$Ii <- loc_I_sf[,"Ii"]
tm_shape(col_sf) + tm_fill("Ii", midpoint=0, style="pretty")
```



Spatial regression

```
library(spatialreg)
ev <- eigenw(lw)
```

```
err <- errorsarlm(CRIME ~ INC + HOVAL, data=col_sf, listw=lw, method="eigen", control=list(pre_eig=ev))
summary(err)
```

```
##
## Call:errorsarlm(formula = CRIME ~ INC + HOVAL, data = col_sf, listw = lw,
##     method = "eigen", control = list(pre_eig = ev))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -34.65998  -6.16943  -0.70623   7.75392  23.43878
##
## Type: error
## Coefficients: (asymptotic standard errors)
##              Estimate Std. Error z value  Pr(>|z|)
## (Intercept) 60.279469   5.365594 11.2344 < 2.2e-16
## INC         -0.957305   0.334231 -2.8642 0.0041806
## HOVAL       -0.304559   0.092047 -3.3087 0.0009372
##
## Lambda: 0.54675, LR test value: 7.2556, p-value: 0.0070679
## Asymptotic standard error: 0.13805
##     z-value: 3.9605, p-value: 7.4786e-05
```

```
## Wald statistic: 15.686, p-value: 7.4786e-05
##
## Log likelihood: -183.7494 for error model
## ML residual variance (sigma squared): 97.674, (sigma: 9.883)
## Number of observations: 49
## Number of parameters estimated: 5
## AIC: 377.5, (AIC for lm: 382.75)
```

Impacts

```
lag <- lagsarlm(CRIME ~ INC + HOVAL, data=col_sf, listw=lw, method="eigen", control=list(pre_eig=ev))
summary(impacts(lag, R=2000, evalues=ev), short=TRUE, zstats=TRUE)
```

```
## Impact measures (lag, evalues):
##           Direct   Indirect      Total
## INC   -1.1008955 -0.7176834 -1.8185788
## HOVAL -0.2795832 -0.1822627 -0.4618459
## ========================================================
## Simulation results (asymptotic variance matrix):
## ========================================================
## Simulated standard errors
##           Direct  Indirect     Total
## INC   0.30732197 0.3856292 0.5728124
## HOVAL 0.09328472 0.1255561 0.1945475
##
## Simulated z-values:
##          Direct  Indirect     Total
## INC   -3.577907 -1.979275 -3.252087
## HOVAL -3.048507 -1.628950 -2.513031
##
## Simulated p-values:
##       Direct     Indirect Total
## INC   0.00034636 0.047785 0.0011456
## HOVAL 0.00229982 0.103324 0.0119699
```

Write GAL file

```
td <- tempdir()
tf <- file.path(td, "col_queen.gal")
write.nb.gal(nb_sf, tf)
```

Using reticulate to run Python from R in an R markdown notebook

```
library(reticulate)
use_python(python='/usr/bin/python3')
py_config()
```

```
## python:         /usr/bin/python3
## libpython:      /usr/lib64/libpython3.7m.so
## pythonhome:     /usr:/usr
```

```
## version:        3.7.4 (default, Jul  9 2019, 16:48:28)  [GCC 8.3.1 20190223 (Red Hat 8.3.1-2)]
## numpy:          /usr/local/lib64/python3.7/site-packages/numpy
## numpy_version:  1.16.0
##
## python versions found:
##  /usr/bin/python3
##  /usr/bin/python
```

```
pkgr <- import("pkg_resources")
np <- import("numpy")
pkgr$get_distribution("numpy")$version
```

```
## [1] "1.16.0"
```

```
libpysal <- import("libpysal")
pkgr$get_distribution("libpysal")$version
```

```
## [1] "4.0.1"
```

```
gpd <- import("geopandas")
pkgr$get_distribution("geopandas")$version
```

```
## [1] "0.4.0"
```

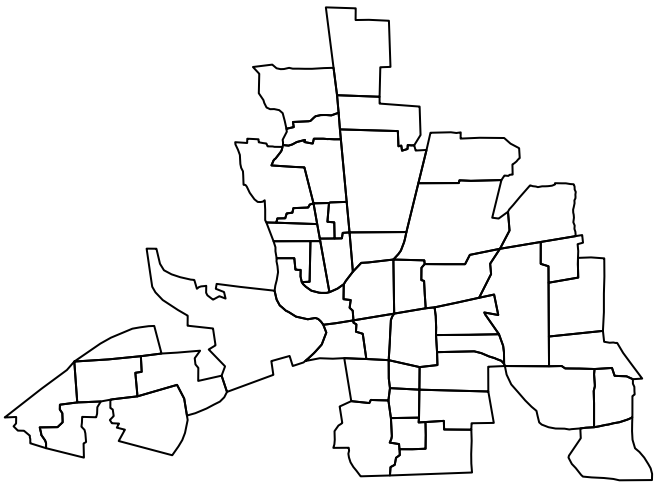
```
col_ps <- gpd$read_file(col_file)
```

Passing shapely geometries to sf

```
shapely <- import("shapely")
pkgr$get_distribution("shapely")$version
```

```
## [1] "1.6.4.post2"
```

```
oo <- st_as_sf(data.frame(unlist(lapply(col_ps$geometry, shapely$wkt$dumps))), wkt=1)
plot(st_geometry(oo))
```



Read GAL file

```
nb_ps <- libpysal$weights$Queen$from_dataframe(col_ps)
nb_gal_ps <- libpysal$io$open(tf)$read()
```

ESDA

```
esda <- import("esda")
pkgr$get_distribution("esda")$version
```

```
## [1] "2.0.0"
```

```
y <- np$array(col_ps[,"CRIME"])
mi <- esda$Moran(y, nb_ps, two_tailed=FALSE)
mi$I
```

```
## [1] 0.5001886
```

```
mi <- esda$Moran(y, nb_gal_ps, two_tailed=FALSE)
mi$I
```

```
## [1] 0.5001886
```

```
loc_I_ps <- esda$Moran_Local(y, nb_ps)
col_ps["Is"] <- loc_I_ps$Is
```

Spatial regression

```
spreg <- import("spreg")
pkgr$get_distribution("spreg")$version
```

```
## [1] "1.0.4"
```

```
x <- np$array(col_ps[, c("INC", "HOVAL")])
y <- matrix(y, ncol=1)
mlerr_ps <- spreg$ML_Error(y, x, nb_ps)
```

Comparison

```
rbind(R=coefficients(err)[c(2:4,1)], PySAL=c(mlerr_ps$betas))
```

```
##       (Intercept)        INC      HOVAL    lambda
## R        60.27947 -0.9573053 -0.3045593 0.5467531
## PySAL    60.27947 -0.9573053 -0.3045593 0.5467530
```

Python directly in R markdown

Read shapefile

```
import numpy as np
import libpysal as libpysal
```

```
import geopandas as gpd
col_ps = gpd.read_file('/home/rsb/lib/r_libs/spData/shapes/columbus.shp')
```

Create weights

```
nb_ps = libpysal.weights.Queen.from_dataframe(col_ps)
nb_ps.cardinalities
```

```
## {0: 2, 1: 3, 2: 4, 3: 4, 4: 8, 5: 2, 6: 4, 7: 6, 8: 8, 9: 4, 10: 5, 11: 6, 12: 4, 13: 6, 14: 6, 15: 8
```

ESDA

```
import esda as esda
mi = esda.Moran(col_ps[['CRIME']].values, nb_ps, two_tailed='false')
mi.I
```

```
## 0.5001885571828611
```

Spatial regression

```
import spreg as spreg
mlerr_ps = spreg.ML_Error(col_ps[['CRIME']].values, col_ps[['INC', 'HOVAL']].values, nb_ps)
```

```
## /usr/local/lib64/python3.7/site-packages/scipy/optimize/_minimize.py:761: RuntimeWarning: Method 'bou
##   "defaulting to absolute tolerance.", RuntimeWarning)
```

```
mlerr_ps.betas
```

```
## array([[60.2794697 ],
##        [-0.95730534],
##        [-0.30455926],
##        [ 0.54675303]])
```