Spatial Demography **2013** 1(1): 131-139
http://spatialdemography.org

SPATIAL DEMOGRAPHY
ISSN 2164-7070 (online)

COLUMN

Spatial Analysis in R: Part 1

Getting data from the ACS into R and Exploratory Spatial Data Analysis

Corey Sparks
Editor, Software & Code

***Editorial note**: My column in Spatial Demography examines computer software and program code. One of the key issues facing any scholar is "how" to implement a particular analysis. Much of this activity has been fugitive from the formal literature but the recent trend toward sharing software features and programmable code is helping to reduce this confusion. This column will feature some tools that will assist spatial demographers in their craft. Suggestions or questions are welcomed in the Spatial Demography Forum on the journal website.*

Software for spatial analysis has come a long way in the past decade. Along with the improvements in software has come a flurry of modeling strategies, some new and some quite old but recently ported to programs that facilitate their use. Since one purpose of *Spatial Demography* is to highlight current techniques used by spatially minded population scientists, it occurred to me to write a series of columns related to actually doing some of these models in a software environment that all readers of the journal could use, R.

R is free-ware, and available for a wide variety of operating systems. (And, did I mention it's free?) If you are reading this and at a major research university or large federal or state agency, you may not care about free (even though your boss would probably appreciate it), but if you are an independent demographer, or working at a small university or college or private firm, then free can mean the difference between getting the job done and waiting for the next contract. R is maintained by a large group of scientists, like you and I who have come together to form the Comprehensive R Archive Network, or CRAN (http://cran.r-project.org/). R is written by users, many of whom have received funding from the feds in their own nation, but much R code is simply written by people who figured out a new method and wanted to share it with other scientists. R itself is a statistical programming language, and has an enormous capacity for different types of numerical and statistical analysis. I leave the reader to download and install R on their computers, there are user-friendly installers for most operating systems on the website above.

The purpose of this column and the next few to follow are simply to provide all readers of the journal with blow-by-blow examples of how to do methods that have become popular in spatial demographic analysis over the last few years. Topics will include getting data into R from the American Community Survey, exploratory spatial data analysis, simultaneous autoregressive (SAR) models, geographically weighted regression, and Bayesian spatial models.

I teach a course in spatial statistics in our applied demography doctoral program, and many of the examples that I will cover were created as part of developing that curriculum. Some of the examples have been published, some are not, but all are examples of how someone who is interested in learning how to do these methods can implement them on real, messy data with little to no financial investment.

Part 1) Getting data from ACS

The American Community Survey has replaced the Census long-form data, and is the source for most socioeconomic information in the US, especially after the year 2000. It is produced yearly, and estimates are produced at all geographic summary levels on a rolling basis annually after 2009. The examples I will be working with here use the 2005-2009 5-year ACS estimates for Bexar (pronounced 'bear') County, Texas. Bexar County is where the city of San Antonio is located (as well as the author), and has a population of 1,714,773 as of the 2010 Census. It is a dynamic community, with a long history and lots of social problems, including high teenage birth rates, high child poverty rates and a long history of social stratification.

First, we need to get some data from ACS. The following steps will retrieve a large number of variables that form the "Selected Social Characteristics" table from the 5 year ACS.

Go to Factfinder 2's website (http://factfinder2.census.gov/). Then go to Search, choose Topics and choose Dataset. For this example we want the 2009 ACS 5 yr estimates. Once you select this, then select Geographies. We want the Bexar County census tracts, so choose Census tract, then choose Within county. Select Bexar county from the alphabetical list. In the Geography results section select "All Census tracts within Bexar County Texas". Then choose Add from the Geography results window. This will retrieve whatever tables we choose for these tracts. Hit Close. Choose the *Selected Social Characteristics in the United States: 2005-2009* table from the available tables. This table contains a wide variety of social and economic variables, and is much faster for our puposes. Otherwise, you could search by table number or topic for specific data items. Finally, hit download. Factfinder will generate the data table for the selected geographies, and you will be prompted to download the data when the query finishes, usually in about a minute for this many geographies.
Once the file finishes downloading, unzip it, it will have a name like "aff_download.zip". It will unzip into a folder with the same name. Within the folder there are three files if you only selected the one table. For me, the files were ACS_09_5YR_DP5YR2_with_ann.csv, ACS_09_5YR_DP5YR2.txt and aff_download_readme_ann.txt. The first of these is a data file that contains annotations, or descriptions of each of the columns. If you examine the .csv file, you see (or at least the one I downloaded had this) several header rows in the data, these are not very useful for our purposes, and will need to be skipped over when reading in the data.

We also need geographies if we want to map or do spatial anlaysis of the ACS data, so we will use the Census's 2009 Tiger Line files at: (http://www.census.gov/geo/www/tiger/tgrshp2009/tgrshp2009.html). From here select "Download the 2009 Tiger/line shapefiles now", in State- and County –based shapefiles, select Texas and click submit. Select Census tract (Census 2000) on the left and choose Bexar county on the right and hit submit. On the next page choose Census tracts (Census 2000) and hit Download Selected Files. Again, you will need to unzip the file that downloads, and there is another zip file within it that also must be unzipped. Eventually, I ended up with a folder structure like:

48_TEXAS>48029_Bexar_County>tl_2009_48029_tract00

and in the lowest level of the folders, there are five files, which together provide the makings of a shapefile. If you choose to copy these elsewhere, copy all of these files to the same folder location. If you do not copy all of them, certain aspects of the data will be broken.

Part 2) Getting ACS data into R

A package for R was recently written which will read in tables from the ACS with a minimum of effort from the user (Glenn 2012). From here on out, R code will be presented in *italics*, and output from R in **bold text**.

Make sure you have a working internet

connection and in your R session window, type

install.packages("acs", repos="http://cran.us.r-project.org"))

Which will download and install the acs package.

then type

library(acs)

which loads the functions in the package.

Then we want to read the downloaded ACS table of social characteristics into R. You will have to tell R where the table is, so I would recommend you locating it.
R will start its session in a "working directory", to see where this is type:

getwd( )

mine said:

"/Users/ozd504"

now read the ACS data into R
myacs<-
read.acs("~/Downloads/aff_download/ACS_09_5YR_DP5YR2_with_ann.csv", geocols=3:1, skip=9)

this code reads the table in, skipping the first 9 lines (yours may be different, that's why we looked at the csv file earlier) and putting the first three columns of the data as the geographical identifiers. We could call this anything we want, *myacs* seemed like a simple choice. Now you should have an object in R that contains the ACS table.

Part 3) Read the tract data into R

Once again, since we need a library to read in the shapefile, I recommend installing several packages automatically, the one that we will need for our statistics is called spdep (Bivand 2011). Again, install this package:
install.packages("spdep", dependencies=T, repos="http://cran.us.r-project.org"))

this will install spdep and several other packages that we will need later to read in shapefiles, make maps and handle spatial data. Once it is finished, type

library(spdep)

Now we need to read the shapefile for the tracts into R. Type:

geodat<-
readShapePoly("~/Downloads/48_TEXAS/48029_Bexar_County/tl_2009_48029_tract00/tl_2009_48029_tract00.shp",
proj4string=CRS('+proj=longlat +datum=NAD83'))

This reads the polygon shapefile located in the folder described above into R, and tells R that the data are in a North American Datum 1983 geographic coordinate system.

Your data is now in R! Now we must join the ACS data to the geographies. The ACS data are in *myacs*, which is an object with specific properties. One of these properties, is that the program was smart enough to read the table and separate the estimates from the margins of error (MOE). We want to get a few of the estimates, from the object and put them into a data frame (how R typically stores data) and merge it to the shapefile. First, we extract a few variables from the acs object.

fertrate<-myacs@estimate[,75]
pnohs<-
myacs@estimate[,118]+myacs@estimate[,120]
pfornborn<-myacs@estimate[,184]
pspanspkhh<-myacs@estimate[,228]
pmarriedwomen<-myacs@estimate[,64]
tract<-substr(myacs@geography$Id,15,20)

There are variables for the general fertility rate, the percent of the population without a high school diploma, the percent foreign born, the percent of the population who speak Spanish, the percent of women who are married and an identifier for the census tract. The [, ##] syntax tells R to extract the xxth column of the estimates.

Again, I figured this out by looking at the csv file.

Next we assemble these new variables into a data frame

```
dat<-data.frame(fertrate=fertrate,
pnohs=pnohs, pfornborn=pfornborn,
pspanspkhh=pspanspkhh,
pmarriedwomen=pmarriedwomen, tract=tract)
```

if we look at the first few observations by using *head ( ) head (dat)*

fertrate	pnohs	Pfornborn	Pspanspkhh	Pmarried women	tract
6	35.4	27.4	62.4	21.9	110100
143	44.6	30.8	68.9	30.2	110200
145	36.2	18.6	58.7	35.1	110300
0	30.4	16.7	69.1	49.8	110400
162	54.5	11	71.1	24.6	110500
105	48.5	13.8	69.7	18.6	110600

Now we must merge the data frame to the tract polygons. This can be trick in R, since you need to preserve the order of the polygons. First we make a copy of the shapefile's attribute table, then we merge the attribute table to the data frame then put the merged data back on the polygons.

```
mdat<-geodat@data
mdat<-merge(x=mdat, y=dat,
by.x="TRACTCE00", by.y="tract", all.x=T,
sort=F)
```

This line of code is a left inner join, where we keep all of the polygons, and we do not sort the new data be the tract ID.

```
geodat@data<-mdat
```

In Bexar county, there are some tracts which are military bases, and don't have estimates, so we get rid of them.

```
keep<-!is.na(geodat$fertrate)
geodat<-geodat[keep,]
```

Part 4) Making a simple thematic map

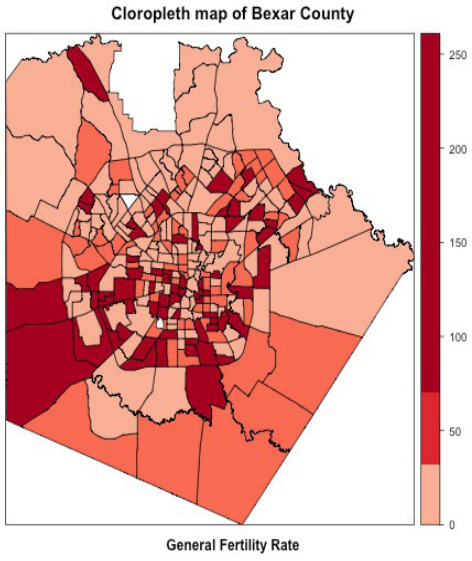
Making a choropleth map in R is very simple, and the RColorBrewer library makes the friendly color schemes of Color Brewer (http://colorbrewer2.org/) available in R, but we need to install the library.

```
install.packages("RColorBrewer",
repos="http://cran.us.r-project.org"))
library(RColorBrewer)
```

The next code maps the general fertility rate for the tracts using *spplot()*

```
spplot(geodat, "fertrate",
    at=quantile(geodat$fertrate, p=c(0,.25, .5, .
75, 1), na.rm=T),
            col.regions=brewer.pal(5, "Reds"),
main="Choropleth map of Bexar County",
sub="General Fertility Rate")
```

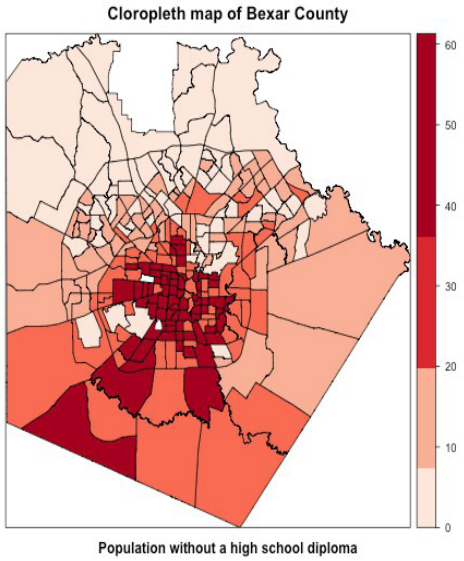
This code makes a quartile map of the fertility rate, the plot produced should look like this:



That is the simple method for plotting variables, here is the map of the percentage of the population without a high school diploma.

```
spplot(geodat, "pnohs",
    at=quantile(geodat$pnohs, p=c(0,.25, .5, .75,
1), na.rm=T),
    col.regions=brewer.pal(5, "Reds"),
```

main="Cloropleth map of Bexar County", sub= "Population without a high school diploma")



Part 5) Exploratory spatial data analysis (ESDA)

A common first step in any spatial statistical analysis is to use exploratory methods to learn about the nature of your data. Tools such as GeoDa (Anselin et al. 2006) have revolutionized the ease by which ESDA is done, allowing for brushing of data, descriptive graphical summaries and standard Univariate and multivariate ESDA statistics. One problem with GeoDa is lack of flexibility. what it does, it does very well, but if users want ESDA functionality within a larger programming environment, R provides a good solution for many of the common ESDA tasks.

Most ESDA methods work using some distance or contiguity-based spatial weighting scheme to measure which observations are close or contiguous to one another. To create such weights in R is a two-step process for contiguity weights, and a three step process for distance-based weights.

Contiguity based weights
First we will make a simple Queen-based contiguity binary weight matrix to represent our geographies. Polygons are considered to be Queen adjacent if they share a vertex or a line segment.

sa.nb<-poly2nb(geodat, queen=T)
summary(sa.nb)

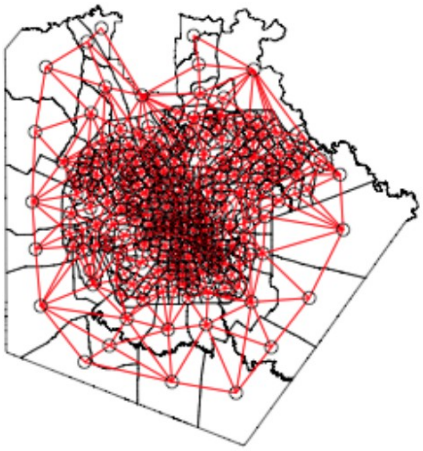
Neighbour list object:
Number of regions: 276
Number of nonzero links: 1762
Percentage nonzero weights: 2.313064
Average number of links: 6.384058
Link number distribution:

2 3 4 5 6 7 8 9 10 11 12 13
2 4 27 50 65 67 37 13 8 1 1 1
2 least connected regions:
197 235 with 2 links
1 most connected region:
236 with 13 links

sa.wt<-nb2listw(neighbours=sa.nb, style="B")

now the object *sa.wt* is a representation of the binary weights. If we want to see what the network of points looks like, we can plot it.

plot(geodat)



plot(sa.nb, coordinates(geodat), add=T, col="red")

Distance based weights area also easily accomplished. For example, a threshold

distance-based contiguity weight system is slightly more complicated. Since the Tiger files are in a lat/long geographic coordinate system, we should really project them into a projected coordinate system to measure distances meaningfully. The rgdal package (Keitt et al. 2012) provides projection methods, but it can be challenging to install for users of Mac OS or Linux. I would refer you to instructions here for Mac users. Once rgdal is installed, load the library

library(rgdal)

Then project the data into a projected coordinate system, I chose UTM Zone 14 North, which is the UTM zone for Bexar County

geodat.utm<-
spTransform(geodat,CRS=CRS("+proj=utm +zone=14 +datum=NAD83"))

Then do the distance based contiguity, using a 10km distance cut off.

sa.dw1<-
dnearneigh(x=coordinates(geodat.utm), d1=0, d2=10000)
sa.wtd<-nb2listw(sa.dw1, style="B")

Now we can examine our data for spatial autocorrelation using the Moran I statistics. Here I use a Monte-Carlo testing method similar to that used by GeoDa.

moran.mc(x=geodat.utm$fertrate,listw=sa.wt, nsim=99)

Monte-Carlo simulation of Moran's I

data: geodat$fertrate
weights: sa.wt
number of simulations + 1: 100

statistic = 0.0933, observed rank = 99, p-value = 0.01
alternative hypothesis: greater

moran.mc(x=geodat.utm$pnohs, listw=sa.wt, nsim=99)

Monte-Carlo simulation of Moran's I
data: geodat$pnohs
weights: sa.wt
number of simulations + 1: 100

statistic = 0.7506, observed rank = 100, p-value = 0.01
alternative hypothesis: greater

And both variables show significant spatial autocorrelation using the Moran statistic, although the fertility variable's value is much lower (.09) than the high school education (.75).

Next we could use Geary's C as an alternative measure of autocorrelation.

geary.mc(x=geodat.utm$fertrate, listw=sa.wt, nsim=99)

Monte-Carlo simulation of Geary's C

data: geodat$fertrate
weights: sa.wt
number of simulations + 1: 100

statistic = 0.9049, observed rank = 4, p-value = 0.04
alternative hypothesis: greater

geary.mc(x=geodat.utm$pnohs, listw=sa.wt, nsim=99)

Monte-Carlo simulation of Geary's C

data: geodat$pnohs
weights: sa.wt
number of simulations + 1: 100

statistic = 0.2349, observed rank = 1, p-value = 0.01
alternative hypothesis: greater

And again, we see significant autocorrelation for both variables, although the fertility variable is close to being insignificant using alpha = .05.

Using the distance-based contiguity weights, we still see significant autocorrelation, although the

values of I have decreased

moran.mc(x=geodat.utm$fertrate, listw=sa.wtd, nsim=99)

Monte-Carlo simulation of Moran's I

data: geodat$fertrate
weights: sa.wtd
number of simulations + 1: 100

statistic = 0.0442, observed rank = 100, p-value = 0.01
alternative hypothesis: greater

moran.mc(x=geodat$pnohs, listw=sa.wtd, nsim=99)

Monte-Carlo simulation of Moran's I

data: geodat$pnohs
weights: sa.wtd
number of simulations + 1: 100

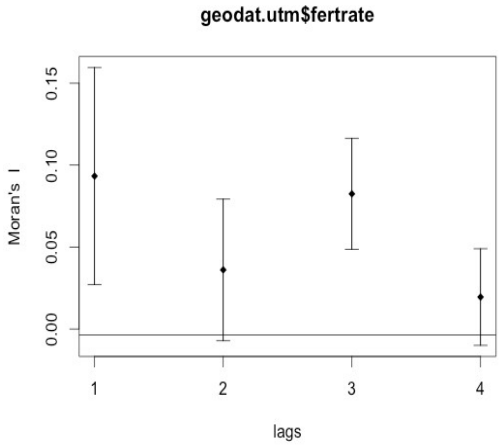
statistic = 0.5204, observed rank = 100, p-value = 0.01
alternative hypothesis: greater

Spatial correllograms

The spatial correlogram is also a useful tool for examining if autocorrelation is a local pheonomena (only present among close neighbors) or if it extends over the entire study area (higher order neighbors). It is easily implemented and plotted in R using *sp.correlogram()*

plot(sp.correlogram(neighbours=sa.nb, var=geodat.utm$fertrate,order=4, method="I",style="B"))

The plot below shows significant autocorrelation using the Moran I statistic among first order neighbors and third order neighbors, but not second or fourth order neighbors.



Local Indicators of Spatial Autocorrelation

A very popular methodology in ESDA is using local versions of autocorrelation statistics (Anselin 1995) to visualize if autocorrelation is present among a subset of observations within the study region. This is often referred to as local clustering, where groups of observations close to one another show highly correlated attributes. Here, we implement the local Moran statistic and use the false discovery rate correction (Benjamini and Yekutieli 2001) to correct the p-values for the local statistical tests. This correction adjusts the p-value of the local tests for the large number of tests under consideration in the analysis (i.e. each observation has a test statistic, which increases the type 1 error rate substantially).

First, we calculate the local Moran statistic and store it in an object.

local.mi.fert<-
localmoran(x=geodat.utm$fertrate, listw=sa.wt, alternative="two.sided", p.adjust.method="fdr")

Then we put the value of the I statistic into the shapefile, it is the first column of *local.mi.fert*
geodat.utm$lmi<-local.mi.fert[,1]

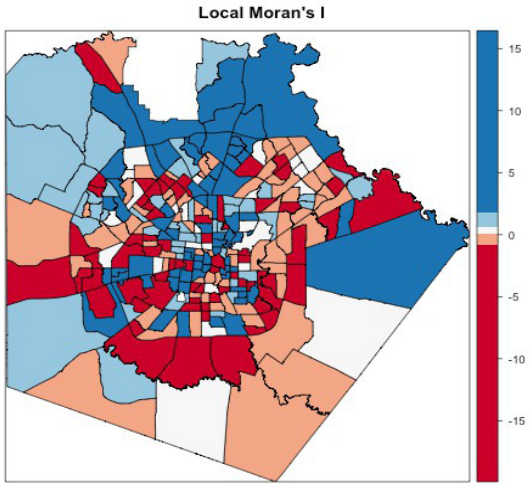
Then we put the p-value for each local test in the shapefile as well, it is the 5th column.
geodat.utm$lmi.p<-local.mi.fert[,5]

Then we code the tracts by what level of statistical significance they exhibit using an if-then-else statement, and convert this to a factor variable so it will plot nicely later.

```
geodat.utm$lmi.p.sig<-
as.factor(ifelse(local.mi.fert[,5]<.001,"Sig
p<.001",  ifelse(local.mi.fert[,5]<.05,"Sig p<.05",
"NS" )))
```

Then we plot the values of the local I statistics for each tract.

```
spplot(geodat.utm,"lmi",
at=summary(geodat.utm$lmi),
col.regions=brewer.pal(5, "RdBu"),
main="Local Moran's I")
```



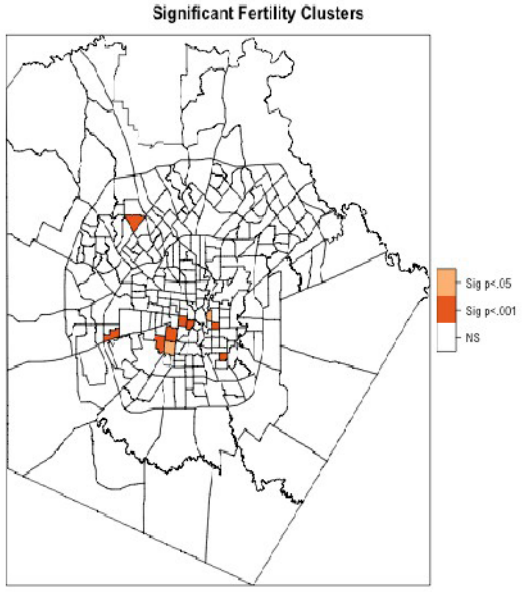
Then we plot the significant local clusters of the fertility rate.

```
spplot(geodat.utm,"lmi.p.sig",
col.regions=c("white", "#E6550D","#FDAE6B"))
```

We see that there are several tracts which show significant local autocorrelation in the fertility rate, most of which occur around the central part of San Antonio, on the east and west sides of downtown.

6) Summary

In this column, we have seen how to obtain current data from the American Community Survey for Census Tracts in Bexar County, Texas and how to merge this data to Census geographies. We also saw how to carry out several forms of exploratory spatial data analysis. This was all done in the free R programming environment.



Over the next few issues of *Spatial Demography*, I will illustrate other spatial statistical methods available in R on this data set described above. While initially R can be imposing to learn, I hope that these examples will popularize its use for spatial analysis in demography, and show users of other software packages the utility and flexibility of the R programming language.

References

Anselin, L., *Local indicators of spatial autocorrelation – LISA*. Geographical Analysis, 1995. **27**: p. 93-115.

Anselin, L., I. Syabri, and Y. Kho, *GeoDa: An Introduction to Spatial Data Analysis*. Geographical Analysis, 2006. **38**(1): p. 5-22.

Benjamini, Y. and D. Yekutieli, *The control of the false discovery rate in multiple testing under dependency*. Annals of Statistics, 2001. **29**: p. 1165-1188.

Bivand, R., et al., *spdep: Spatial dependence: weighting schemes, statistics and models*, 2011.

Glenn, E.H., *acs: Download and manipulate data from the US Census American Community Survey*, 2012.

U.S. Census Bureau. 2010. *State & county Quickfacts: Bexar County, Texas*. Retrieved March 18, 2013. http://quickfacts.census.gov.

Keitt, T.H., et al., *rgdal: Bindings for the Geospatial Data Abstraction Library*, 2012.