Basics of Geographic Analysis in R

Introduction

Yuri M. Zhukov

GOV 2525: Political Geography

February 25, 2013

Overview

1. Introduction
2. Spatial Data and Basic Visualization in R
3. Spatial Autocorrelation
4. Spatial Weights
5. Spatial Regression

Motivations for going spatial

Independence assumption not valid

The attributes of observation i may influence the attributes of j.

Spatial heterogeneity

The magnitude and direction of a treatment effect may vary across space.

Omitted variable bias

There may be some unobserved or latent influences shared by geographical or network "neighbors".

Illustrative examples

Epidemiology

How to model the spread of a contagious disease?

Criminology

How to identify crime hot spots?

Real estate

How to predict housing prices?

Counterinsurgency

"Oil spot" modeling and clear-hold-build

Organizational learning and network diffusion

How to model the adoption of an innovation?

Non-spatial DGP

In the linear case:

$$
\begin{aligned}
y_i &= X_i\beta + \epsilon_i \\
\epsilon_i &\sim N(0, \sigma^2), \quad i = 1, \ldots, n
\end{aligned}
$$

Assumptions

- Observed values at location i independent of those at location j
- Residuals are independent ($E[\epsilon_i \epsilon_j] = E[\epsilon_i]E[\epsilon_j] = 0$)

The independence assumption greatly simplifies the model, but may be difficult to justify in some contexts...

Spatial DGP

With two neighbors i and j:

$$
\begin{aligned}
y_i &= \alpha_j y_j + X_i \beta + \epsilon_i \\
y_j &= \alpha_i y_i + X_j \beta + \epsilon_j \\
\epsilon_i &\sim N(0, \sigma^2), \quad i = 1 \\
\epsilon_j &\sim N(0, \sigma^2), \quad j = 2
\end{aligned}
$$

Assumptions

- Observed values at location i depend on those at location j, and vice versa
- Data generating process is "simultaneous" (more on this later)

Spatial DGP

With n observations, we can generalize:

$$\begin{aligned} y_i =& \rho \sum_{j=1}^{n} W_{ij} y_j + X_i \beta + \epsilon_i \\ \epsilon_i \sim& N(0, \sigma^2), \quad i = 1, \ldots, n \end{aligned}$$

In matrix notation:

$$\begin{aligned} \mathbf{y} =& \rho \mathbf{W} \mathbf{y} + \mathbf{X} \beta + \epsilon \\ \epsilon \sim& N(0, \sigma^2 \mathbf{I}_n) \end{aligned}$$

where $\mathbf{W}$ is the spatial weights matrix, ρ is a spatial autoregressive scalar parameter, and $\mathbf{I}_n$ is an $n \times n$ identity matrix

Spatial DGP

- When $\rho = 0$, the variable in not spatially autocorrelated. Information about a measurement in one location gives us no information about the value in neighboring locations (spatial independence).
- When $\rho > 0$, the variable in positively spatially autocorrelated. Neighboring values tend to be similar to each other (clustering).
- When $\rho < 0$, the variable in negatively spatially autocorrelated. Neighboring values tend to be different to each other (segregation).

Spatial DGP

Let's develop this further, for the moment dropping $\mathbf{X}\beta$ and introducing constant term vector of ones ι_n:

$$\begin{aligned}
\mathbf{y} &= \rho\mathbf{W}\mathbf{y} + \iota_n\alpha + \epsilon \\
(\mathbf{I}_n - \rho\mathbf{W})y &= \iota_n\alpha + \epsilon \\
y &= (\mathbf{I}_n - \rho\mathbf{W})^{-1}\iota_n\alpha + (\mathbf{I}_n - \rho\mathbf{W})^{-1}\epsilon \\
\epsilon &\sim N(0, \sigma^2\mathbf{I}_n)
\end{aligned}$$

Spatial DGP

Assuming $|\rho| < 1$, the inverse can be expressed as an infinite series

$$(\mathbf{I}_n - \rho\mathbf{W})^{-1} = \mathbf{I}_n + \rho\mathbf{W} + \rho^2\mathbf{W}^2 + \rho^3\mathbf{W}^3 + \dots$$

implying that

$$\begin{aligned} y = {} & \iota_n\alpha + \rho\mathbf{W}\iota_n\alpha + \rho^2\mathbf{W}^2\iota_n\alpha + \dots \\ & + \epsilon + \rho\mathbf{W}\epsilon + \rho^2\mathbf{W}^2\epsilon + \dots \end{aligned}$$

Since α is a scalar and $\mathbf{W}\iota_n = \iota_n$ (similarly, $\mathbf{W}(\mathbf{W}\iota_n) = \dots = \mathbf{W}^q\iota_n = \iota_n \quad \forall \quad q \geq 0$), this expression simplifies to:

$$y = (1 - \rho)^{-1}\iota_n\alpha + \epsilon + \rho\mathbf{W}\epsilon + \rho^2\mathbf{W}^2\epsilon + \dots$$

Spatial DGP

- Let's say that the rows of the weights matrix $\mathbf{W}$ represent first-order neighbors.
- Then by matrix multiplication, the rows of $\mathbf{W}^2$ would represent second-order neighbors (neighbors of one's neighbors), $\mathbf{W}^3$ third-order neighbors, and so on.
- But wait a minute... isn't i is a second-order neighbor of itself?
- This introduces simultaneous feedback into the model, where each observation y_i depends on the disturbances associated with both first- and higher-order neighbors.
- The influence of higher order neighbors declines when ρ is small (ρ can be interpreted as a discount factor reflecting a decay of influence for more distant observations)
- ...but we still have a mean and VCov structure for observations in the vector $\mathbf{y}$ that depends in a complicated way on other observations.

Spatial DGP

- Simultaneous feedback is not necessarily a bad thing...
- It can be useful if we're modeling spatial spillover effects from neighboring observations to an origin location *i* where the initial impact occurred.
- This approach effectively treats all observations as potential origins of an impact.
- But we also have to be very careful in how we treat spatial data, and how we conceive of the feedback process with regard to time.
- With cross-sectional data, observations are often taken to represent an equilibrium outcome of the spatial process we are modeling.
- But if spatial feedback is modeled as a dynamic process, the measured spatial dependence may vary with the time scale of data collection.

Further Reading

- A.D. Cliff and J.K. Ord (1973), *Spatial Autocorrelation* (London: Pion)
- B.D. Ripley(1981), *Spatial Statistics* (New York: Wiley)
- L. Anselin (1988), *Spatial Econometrics: Methods and Models* (Dordrecht, The Netherlands: Kluwer Academic Publishers)
- P.J. Diggle (2003), *Statistical Analysis of Spatial Point Patterns* (London: Arnold)
- R.S. Bivand, E.J. Pebesma and V. Gomez-Rubio (2008), *Applied Spatial Data Analysis with R* (New York: Springer)
- J. Le Sage and R.K. Pace (2009), *Introduction to Spatial Econometrics* (CRC Press)
- N. Cressie and C.K. Wikle (2011), *Statistics for Spatio-Temporal Data* (Wiley)

Outline

Software options

Application	Availability	Learning Curve	Key Functionality
ArcGIS	License	Medium	Geoprocessing, visualization
GeoBUGS	Free	High	Bayesian analysis
GeoDa	Free	Low	ESDA, ML spatial regression
GRASS	Free	High	Image processing, spatial modeling
Matlab	License	High	Spatial econometrics
QGIS	Free	Medium	Visualization (mostly)
R	Free	High	Weights, spatial econometrics, geostatistics, point processes
STARS	Free	Low	Space-time analysis

Spatial Analysis in R

Task	Packages
Data management	`sp`, `rgdal`, `maptools`
Integration with other GIS	`rgdal`, `RArcInfo`, `SQLiteMap`, `RgoogleMaps`, `spgrass6`, `RPyGeo`, `R2WinBUGS`, `geonames`, `OpenStreetMap`
Point pattern analysis	`spatstat`, `splancs`, `spatialkernel`
Geostatistics	`gstat`, `geoR`, `geoRglm`, `spBayes`
Disease mapping	`DCluster`, `spgwr`, `glmmBUGS`, `diseasemapping`
Spatial regression	`spdep`, `spatcounts`, `McSpatial`, `splm`, `spatialprobit`, `mgcv`

Task	Packages
Data management	sp, rgdal, maptools
Integration with other GIS	rgdal, RArcInfo, SQLiteMap, RgoogleMaps, spgrass6, RPyGeo, R2WinBUGS, geonames, OpenStreetMap
Point pattern analysis	spatstat, splancs, spatialkernel
Geostatistics	gstat, geoR, geoRglm, spBayes
Disease mapping	DCluster, spgwr, glmmBUGS, diseasemapping
Spatial regression	spdep, spatcounts, McSpatial, splm, spatialprobit, mgcv

Where to Find Spatial Data?

Coordinates and Basemaps:

Geographical Place Names `http://www.geonames.org/`

Global Administrative Areas `http://gadm.org/country`

Land Cover and Elevation
`http://eros.usgs.gov/#/Find_Data`

Geo-referenced Data:

2000 U.S. Census Data `http:`
`//disasternets.calit2.uci.edu/census2000/`

Natural Resources `http://www.prio.no/CSCW/Datasets/`
`Geographical-and-Resource/`

International Conflict Data `http://www.acleddata.com/`

A large number of links is also available at
`http://gis.harvard.edu/`

Points

Points are the most basic form of spatial data

- Points are pairs of coordinates (x, y), representing events, observation posts, individuals, cities or any other discrete object defined in space.
- Let's take a look at the dataset `voters`, which includes a set of geographic coordinates (decimal degrees) corresponding to addresses of NC voters listed as requiring ID prior to 2012 election

```
head(voters)
```

LONG	LAT	X_NC	Y_NC	STREET	ZIP	...
-79.53	36.12	561968.95	262516.57	336 SLATE DR	27249	
-79.48	36.05	566746.67	255535.70	3311 HOSKINS CT	27215	
-79.41	36.10	572742.74	260736.26	1718 HILTON RD	27217	
-79.28	36.10	584556.66	260316.05	106 CLEVELAND ST	27302	
-79.34	36.14	578550.34	264813.73	2038 N NC HWY 49 #2	27217	
-79.45	36.08	569221.12	258562.05	1515 S MEBANE ST #62	27215	...

- To work with these data in R, we will need to create a `spatial` object from this table.

Points

Create matrix of coordinates

```
xy_voters <- voters[,c("X_NC","Y_NC")]
```

Define projection: NC State Plane

```
proj<-CRS("+proj=lcc +lat_1=34.333
+lat_2=36.167 +lat_0=33.75 +lon_0=-79
+x_0=609601.22 +y_0=0 +ellps=GRS80
+datum=NAD83 +units=m +no_defs")
```

Create spatial object

```
sp.points<-SpatialPointsDataFrame(coords
= xy_voters, data = voters, proj4string =
proj)
```

Plot the data

```
plot(sp.points, pch=16, cex=.5, axes=T)
```

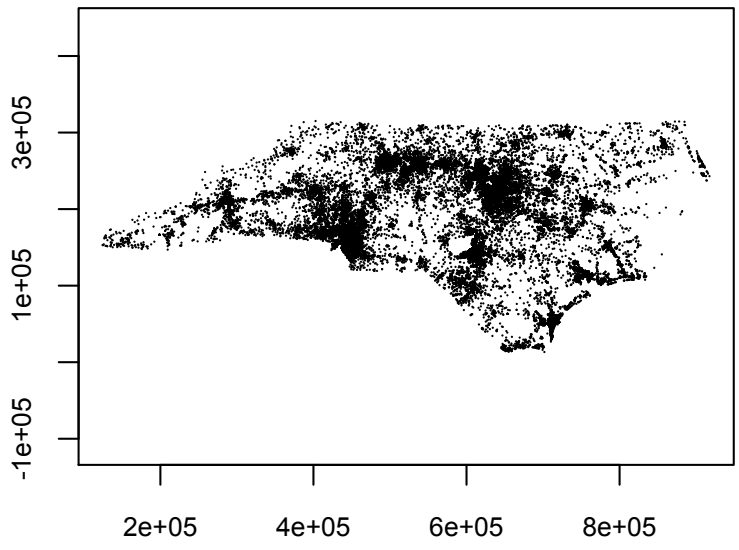


Figure: NC voter ID challenges

Polygons and Lines

Polygons can be thought of as sequences of connected points, where the first point is the same as the last.

- An open polygon, where the sequence of points does not result in a closed shape with a defined area, is called a line.
- In the R environment, line and polygon data are stored in objects of classes `SpatialPolygons` and `SpatialLines`:

```
getClass("Polygon")
```

```
Class Polygon [package "sp"]
 Name:      labpt      area      hole   ringDir   coords
 Class:   numeric   numeric   logical   integer   matrix
```

```
getClass("SpatialPolygons")
```

```
Class SpatialPolygons [package "sp"]
 Name:    polygons   plotOrder     bbox   proj4string
 Class:       list     integer   matrix           CRS
```

Polygons and Lines

Vector data, including polygons, are often stored as ESRI Shapefiles.

- Includes: shapes (`.shp`), index (`.shx`), attribute table (`.dbf`)

These can be loaded into R using the `maptools` library.

Opening Shapefiles

- `data <- readShapePoints("Directory/MyShapefile")`
- `data <- readShapeLines("Directory/MyShapefile")`
- `data <- readShapePoly("Directory/MyShapefile")`

Saving Shapefiles

- `writePointsShape(data, file="Directory/MyShapefile")`
- `writeLinesShape(data, file="Directory/MyShapefile")`
- `writePolyShape(data, file="Directory/MyShapefile")`

Polygons and Lines

Let's take a look at the NC_Counties dataset.

```
sp.poly <- readShapePoly("Data/NC_Counties"); summary(sp.poly)
```

```
 Object of class SpatialPolygonsDataFrame Coordinates:
             min        max
 x   123998.523   935338.9
 y     1018.252   317904.1
Is projected:  TRUE
proj4string :  [+proj=lcc +lat_1=34.333 +lat_2=36.167 +lat_0=33.75
+lon_0=-79 +x_0=609601.22 +y_0=0 +ellps=GRS80 +datum=NAD83 +units=m
+no_defs]
```

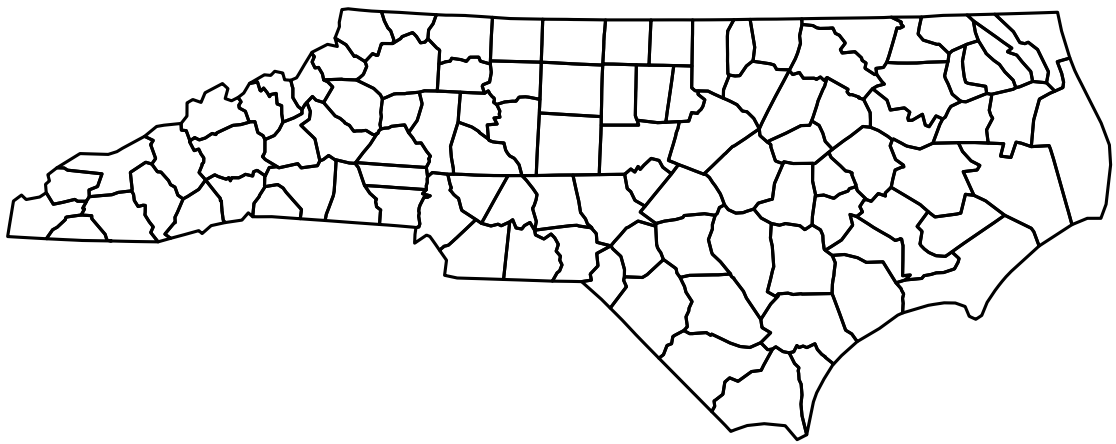
- The data are stored as a SpatialPolygonsDataFrame, a subclass of SpatialPolygons containing a data.frame of attributes.
- In this case, the polygons represent North Carolina counties and attributes include 2008 election results and demographics.

```
names(sp.poly)
```

```
"SP_ID" "ID" "OBAMA2008" "MCCAIN2008" "COUNTY_NAM" "POP2008" "POP2000"
"GROWTH" "MIG_AMOUNT" "MIG_RATE" "NONWHT_PCT" "FRNBORN" "HIGHSCH"
"COLLEGE" "VETERAN" "TVLTIME" "HSDSIZE" "INCPPC" "INCMED" "POVERTY"
"MANFCTR"
```

Polygons and Lines: Visualization

Let's visualize the study region with `plot(sp.poly)`.



Now let's plot some attributes...

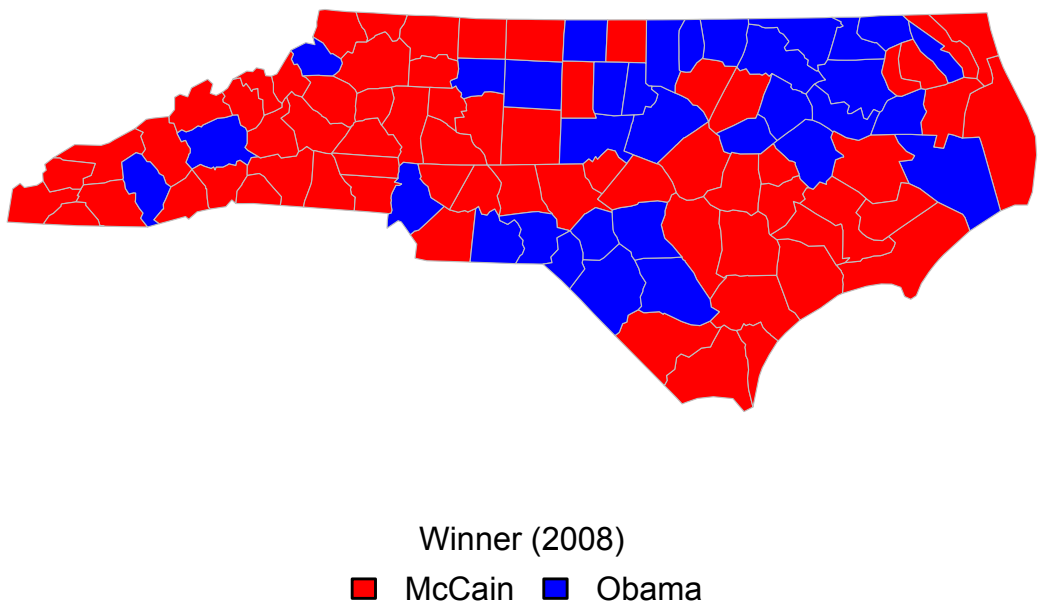
Polygons and Lines: Visualization

- For a `categorical` variable (win/lose), visualization is simple...
  1. Create a vector of colors, where each county won by McCain is coded `"red"` and every each county won by Obama is `"blue"`.

```
cols <- ifelse(sp.poly$OBAMA2008 >
sp.poly$MCCAIN2008,"blue","red")
```

  2. Use the resulting color vector with the `plot()` command.

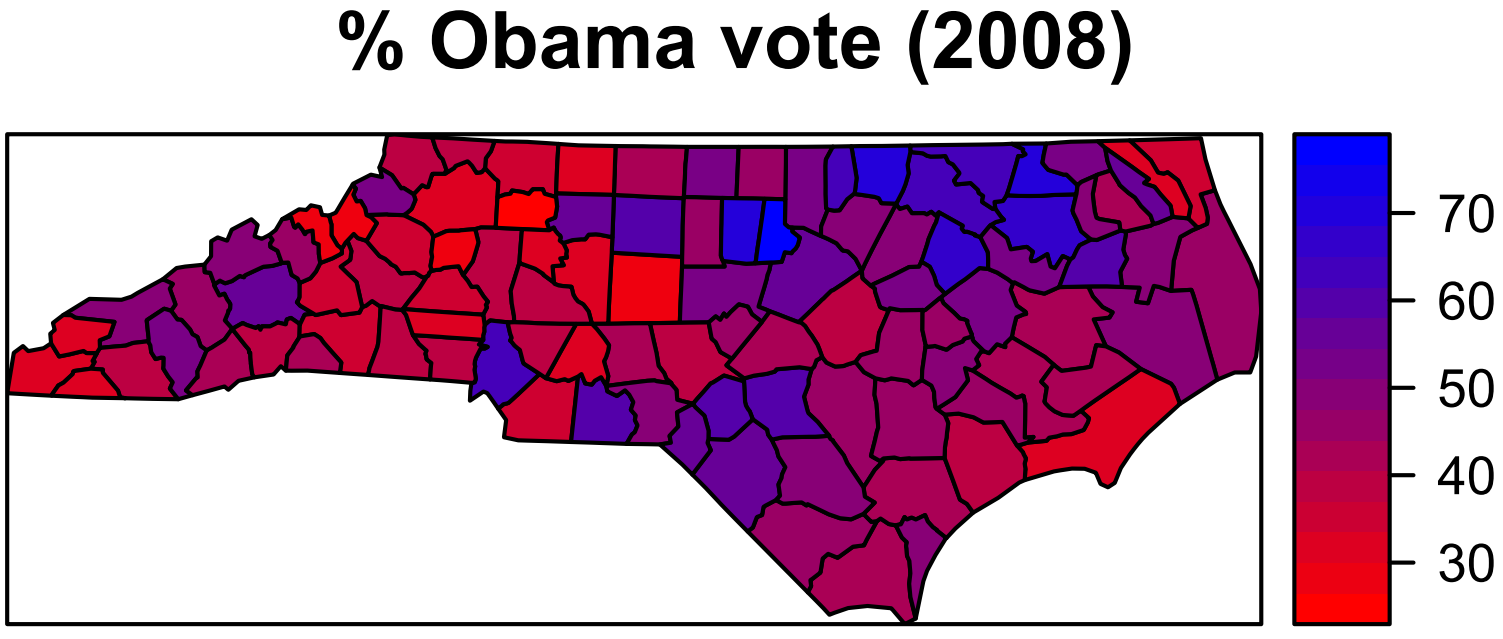
```
plot(sp.poly,col=cols,border=NA)
```



Polygons and Lines: Visualization

With a continuous variable, the same logic applies. A relatively simple approach is to create a custom color palette and use `spplot()`.

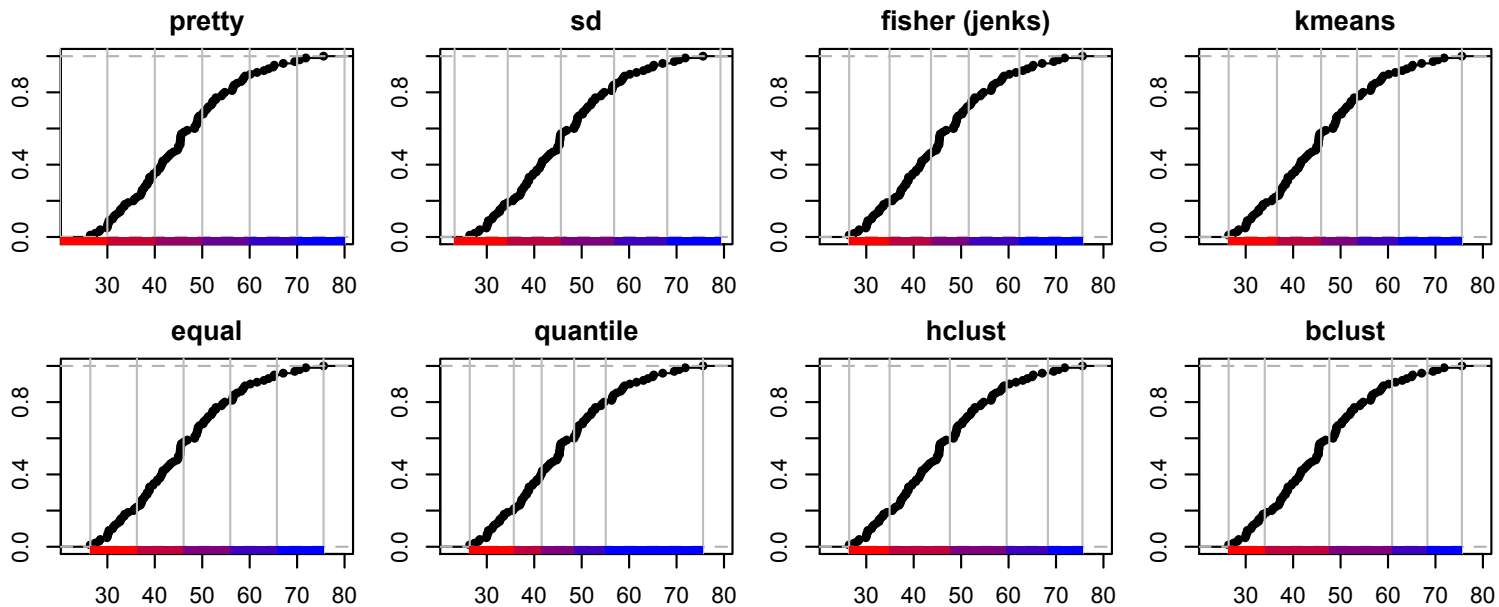
```
br.palette <- colorRampPalette(c("red","blue"), space = "rgb")
spplot(sp.poly,zcol="OBAMA2008",col.regions=br.palette(100))
```



Polygons and Lines: Visualization

- We can also create a color palette for custom classification intervals with the `classInt` package.
- Here is a comparison of six such palettes for the variable `OBAMA2008`, or percentage of popular vote won by Barack Obama.

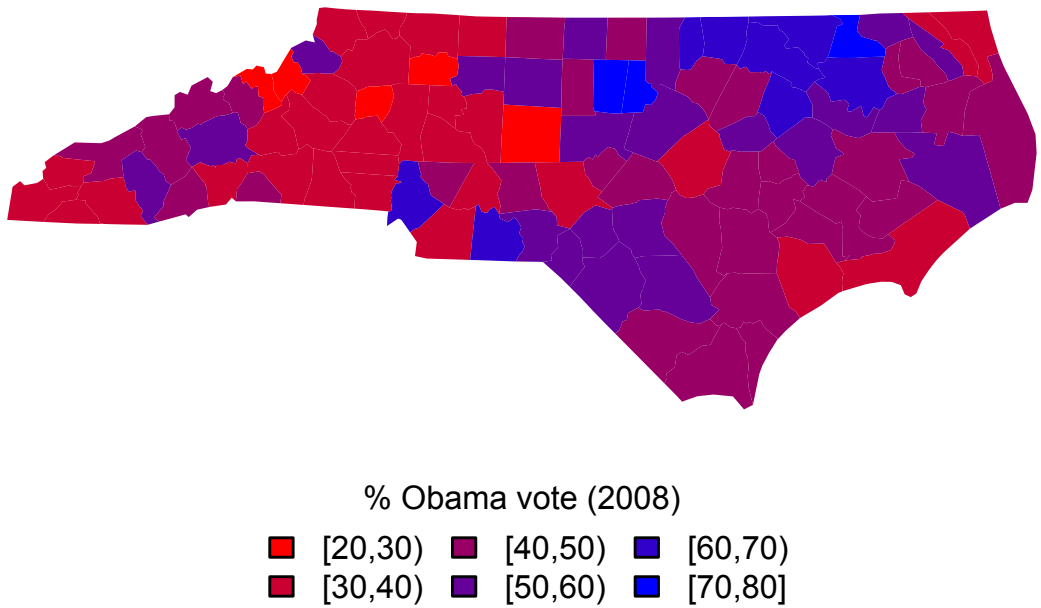
```
var <- sp.poly$OBAMA2008
custom.palette(var=var,col.vec=c("red","blue"),n.color=5,choose=T)
```



Polygons and Lines: Visualization

- Here is a plot of county results using `"pretty"` (default) intervals:

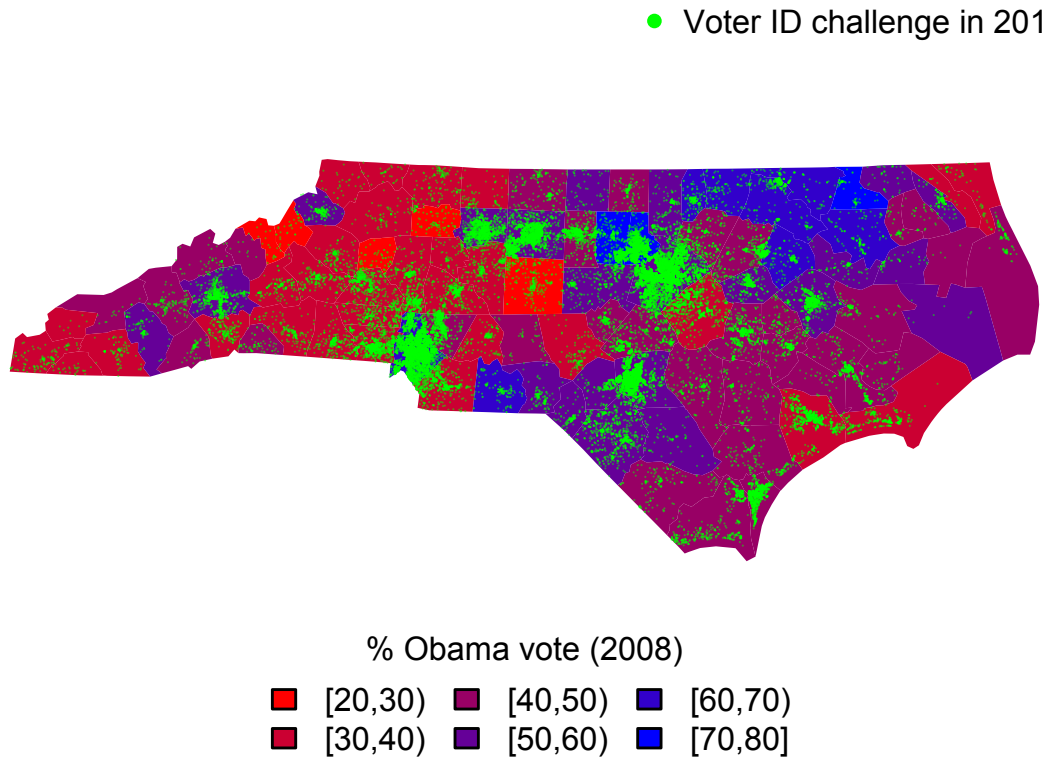
```
custom.palette(var=var,col.vec=c("red","blue"),n.color=5,style="pretty")
plot(sp.poly,col=cols,border=NA)
```



Polygons and Lines: Overlays

- Let's see how the voter ID challenges are distributed across the state

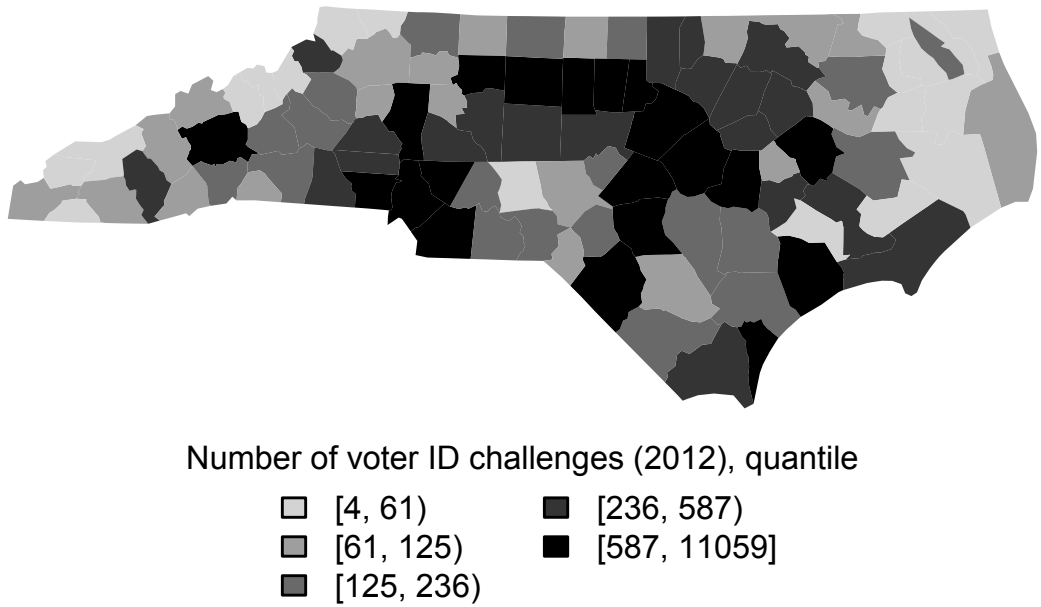
```
plot(sp.points, pch=16, cex=.1, col=rgb(0,1,0,alpha=.5), add=T)
```



Polygons and Lines: Overlays

- It is also possible to join the two layers

```
ovr <- overlay(x=sp.poly2,y=sp.points)
x2 <- cbind(sp.points, ovr)
mat <- aggregate(rep(1,nrow(x2)),by=list(ID=x2$ID),FUN=sum)
sp.poly2 <- sp.merge(sp_data=sp.poly,tab_data=mat,id="ID")
```



Examples in R

Switch to R tutorial script.