Applied Spatial Statistics in R, Section 1

Introduction

Yuri M. Zhukov

IQSS, Harvard University

January 16, 2010

Overview

1. Introduction
   - Why use spatial methods?
   - The spatial autoregressive data generating process
2. Spatial Data and Basic Visualization in R
   - Points
   - Polygons
   - Grids
3. Spatial Autocorrelation
4. Spatial Weights
5. Point Processes
6. Geostatistics
7. Spatial Regression
   - Models for continuous dependent variables
   - Models for categorical dependent variables
   - Spatiotemporal models

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:

$$y_i = X_i\beta + \epsilon_i$$
$$\epsilon_i \sim N(0, \sigma^2), \quad i = 1, \ldots, n$$

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:

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

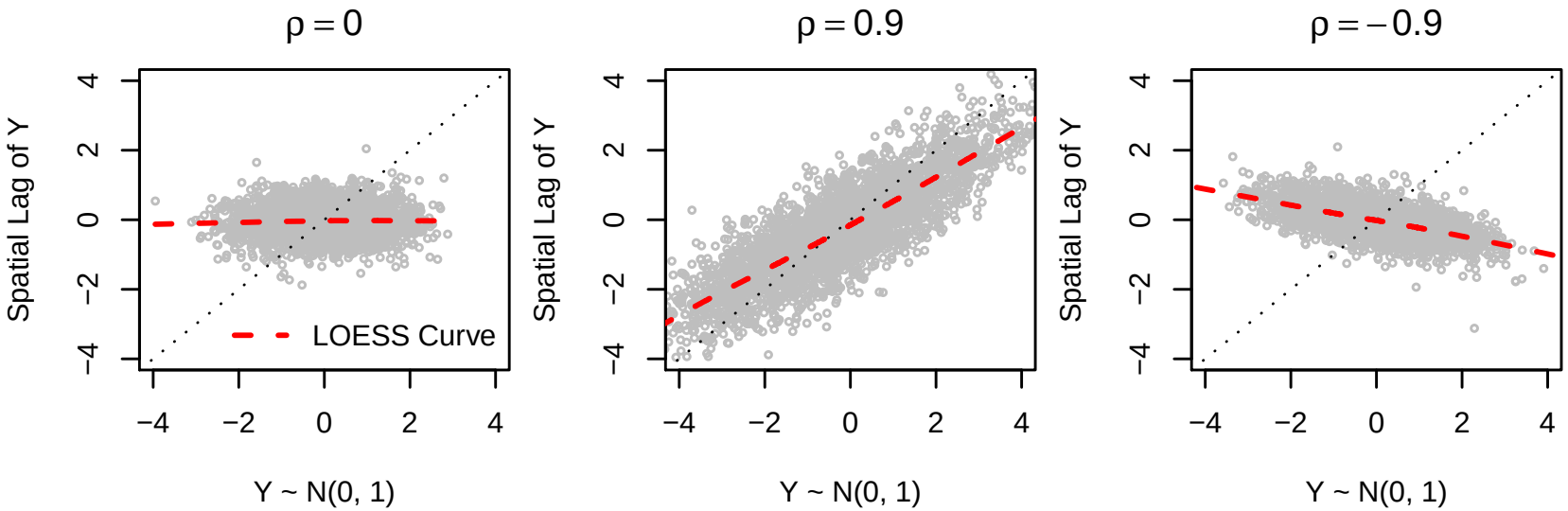
In matrix notation:

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

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, $\mathbf{W}(\mathbf{W}\iota_n) = \dots = \mathbf{W}^q\iota_n = \iota_n$ $\forall \quad q \geq 0$), 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)

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
R	Free	High	Weights, spatial econometrics, geostatistics
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`
Point pattern analysis	`spatstat`, `splancs`, `spatialkernel`
Geostatistics	`gstat`, `geoR`, `geoRglm`, `spBayes`
Disease mapping	`DCluster`, `spgwr`, `glmmBUGS`, `diseasemapping`
Spatial regression	`spdep`, `spatcounts`

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 `crime`, which is just a table of geographic coordinates (decimal degrees) for crime locations in Baltimore, MD.

```
head(crime)
```

```
  ID   LONG        LAT
1  1 -76.65159 39.23941
2  2 -76.47434 39.35274
3  3 -76.51726 39.25874
4  4 -76.52607 39.40707
5  5 -76.51001 39.33571
6  6 -76.70375 39.26605
```

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

Points

Create matrix of coordinates

```
 sp_point <- cbind(crime$LONG, crime$LAT)
colnames(sp_point) <- c("LONG","LAT")
```

Define Projection: UTM Zone 17

```
 proj <- CRS("+proj=utm +zone=17
+datum=WGS84")
```

Create spatial object

```
 data.sp <- SpatialPointsDataFrame(
coords=sp_point, data=crime,
proj4string=proj)
```

Plot the data

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

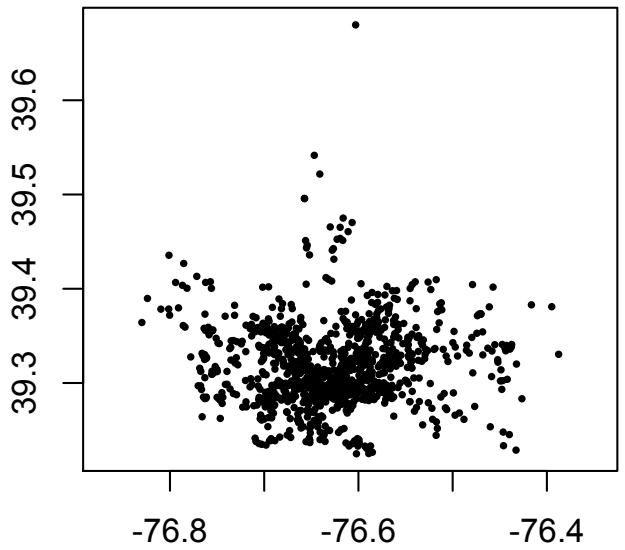


Figure: Baltimore Crime Locations

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

Let's take a look at the `election` dataset.

```
summary(election)
```

```
 Object of class SpatialPolygonsDataFrame
                           min         max
Coordinates:    r1   -124.73142   -66.96985
                r2     24.95597    49.37173
Is projected:  TRUE
proj4string :  [+proj=lcc+lon_0=90w +lat_1=20n +lat_2=60n]
```

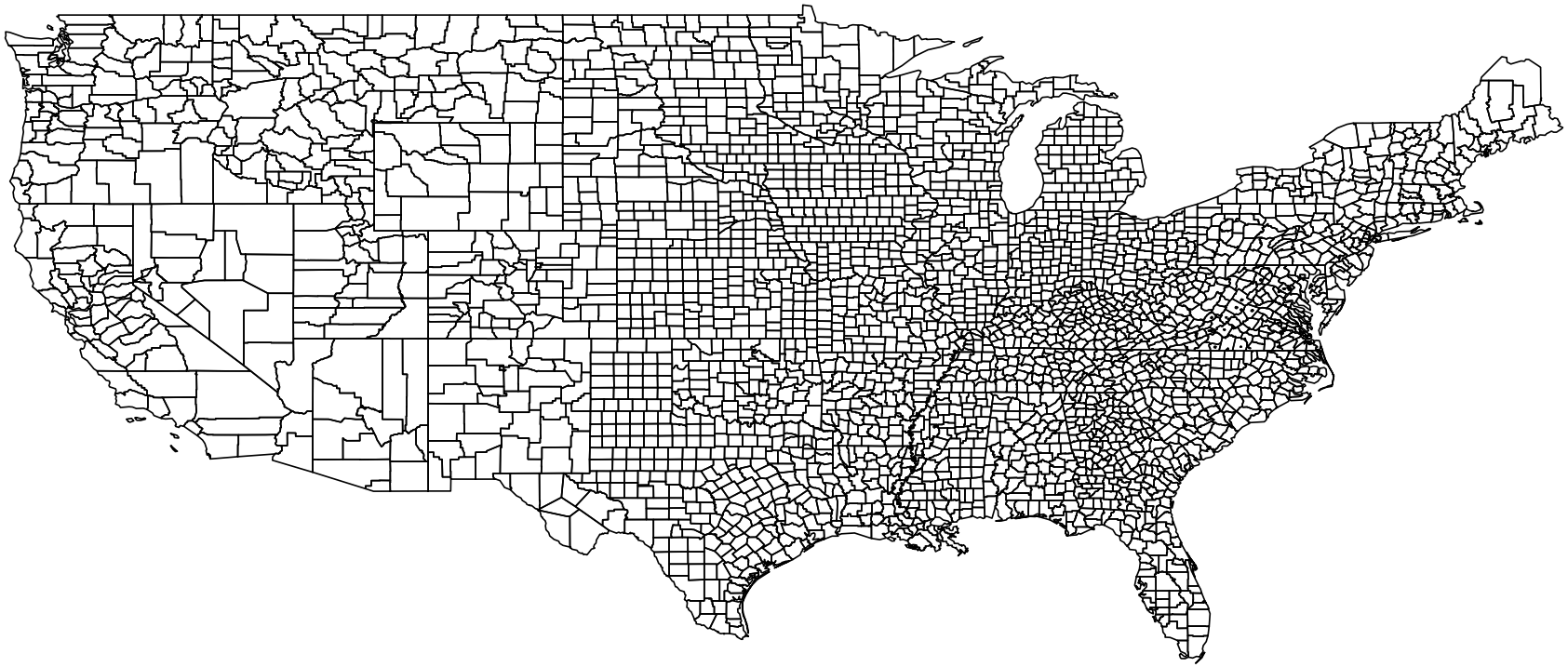
- The data are stored as a `SpatialPolygonsDataFrame`, which is a subclass of `SpatialPolygons` containing a data.frame of attributes.
- In this case, the polygons represent U.S. counties and attributes include results from the 2004 Presidential Election.

```
names(election)
```

```
 [1] "NAME" "STATE_NAME" "STATE_FIPS" "CNTY_FIPS" "FIPS" "AREA" "FIPS_num" "Bush"
[9] "Kerry" "County_F" "Nader" "Total" "Bush_pct" "Kerry_pct" "Nader_pct"
```

Polygons and Lines: Visualization

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



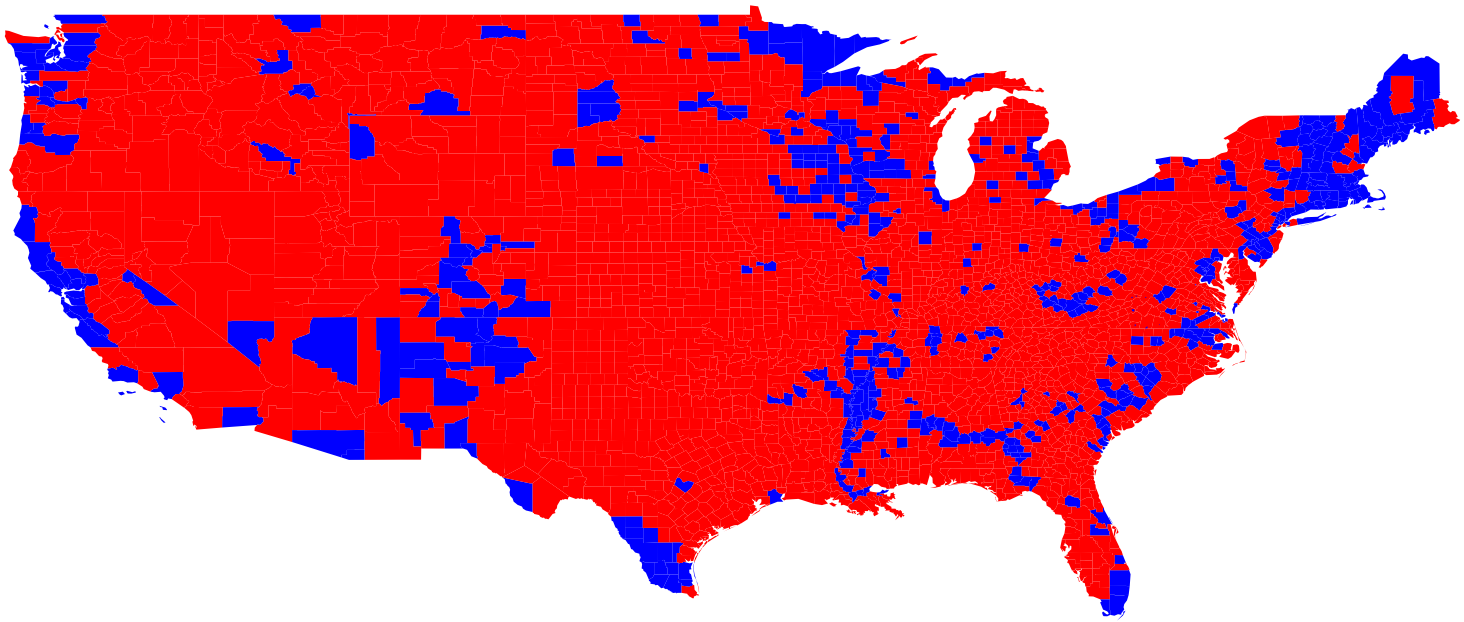
Polygons and Lines: Visualization

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

     ```
     cols <- ifelse(election$Bush > election$Kerry,"red","blue")
     ```

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

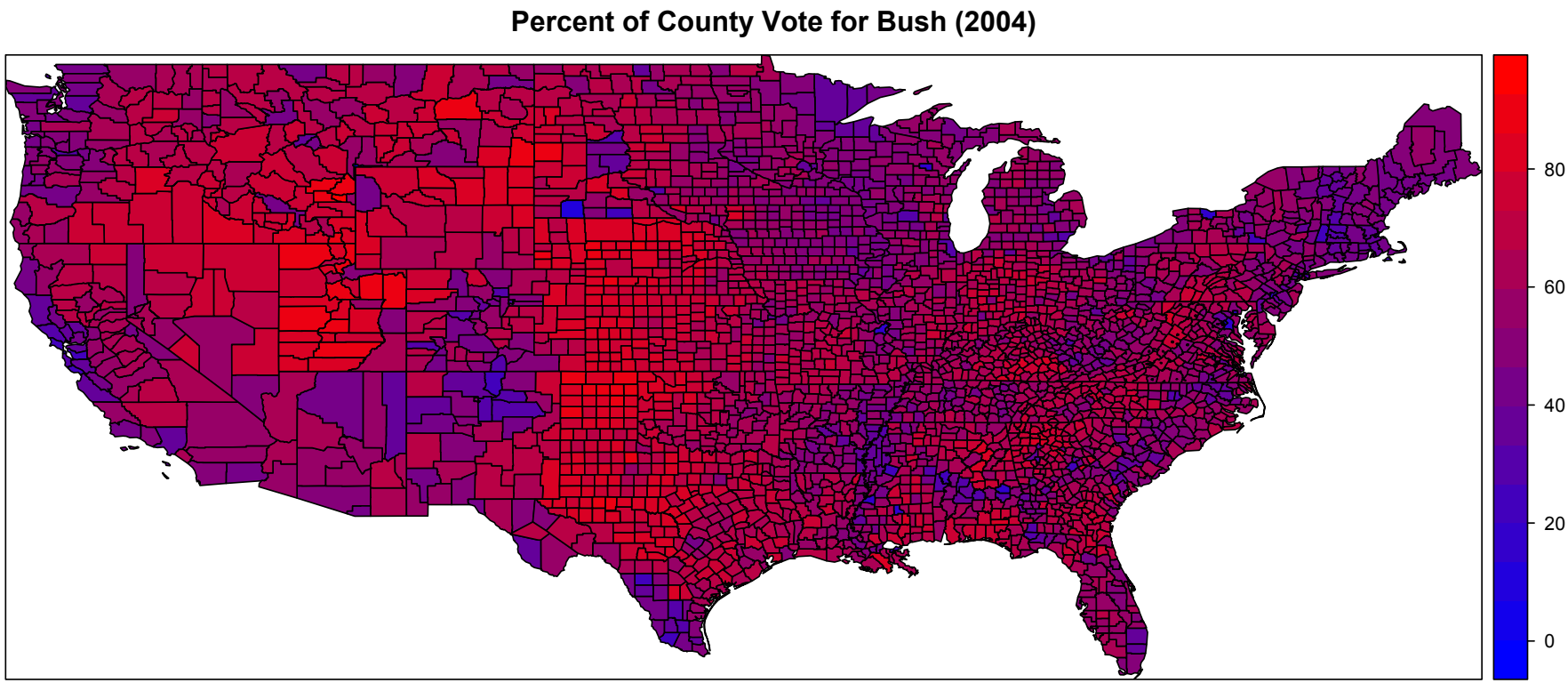
     ```
     plot(election,col=cols,border=NA)
     ```



Polygons and Lines: Visualization

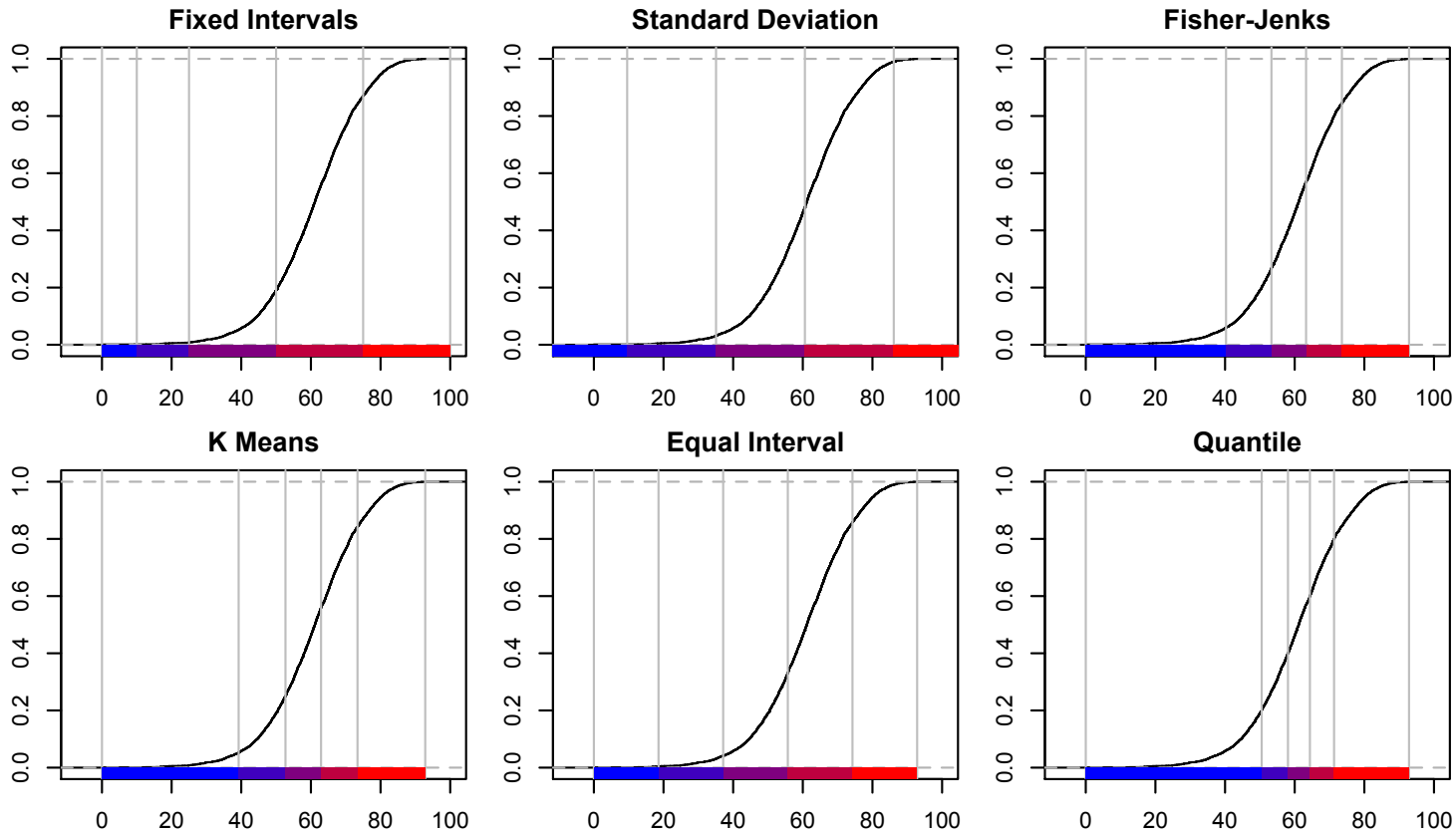
With a continuous variable, the same logic applies. A relatively simple approach is to create a custom color palette and use `spplot()`.

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



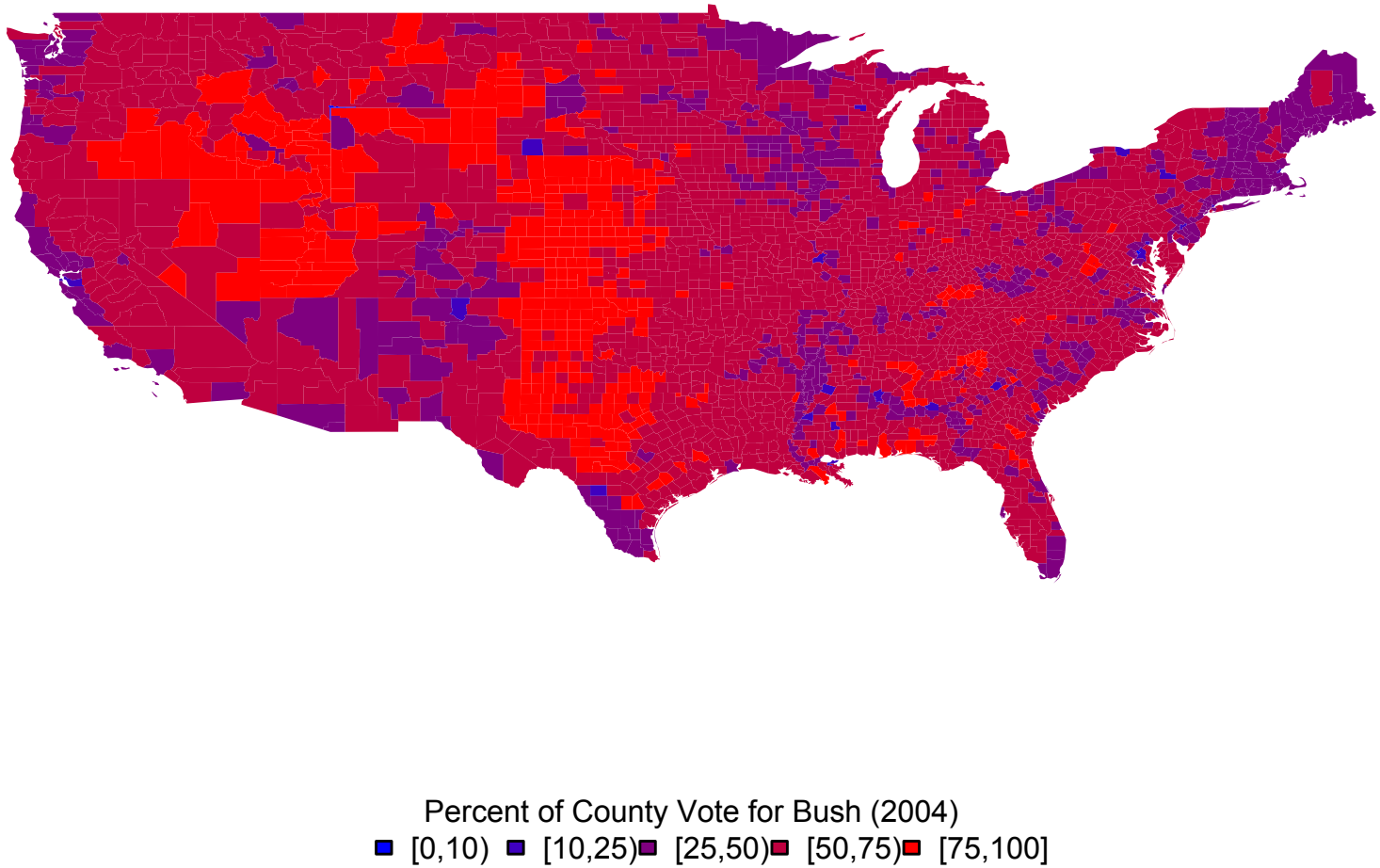
Polygons and Lines: Visualization

- 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 `Bush_pct`, or percentage of popular vote won by George W. Bush.



Polygons and Lines: Visualization

- Here is a plot of county results using the fixed intervals:



Grids

A raster grid divides the study region into a set of identical, regularly-spaced, discrete elements (pixels), each of which records the value or presence/absence of a quantity of interest.

- Rasters originated in image processing, and are used to record properties varying continuously with space.
- Common uses include remote sensing data, elevation models and spatial prediction (weather forecasts, disease risk, etc.).

Take a look at the data structure of the `volcano` dataset, a grid of elevation measures for the Maunga Whau Volcano in New Zealand:

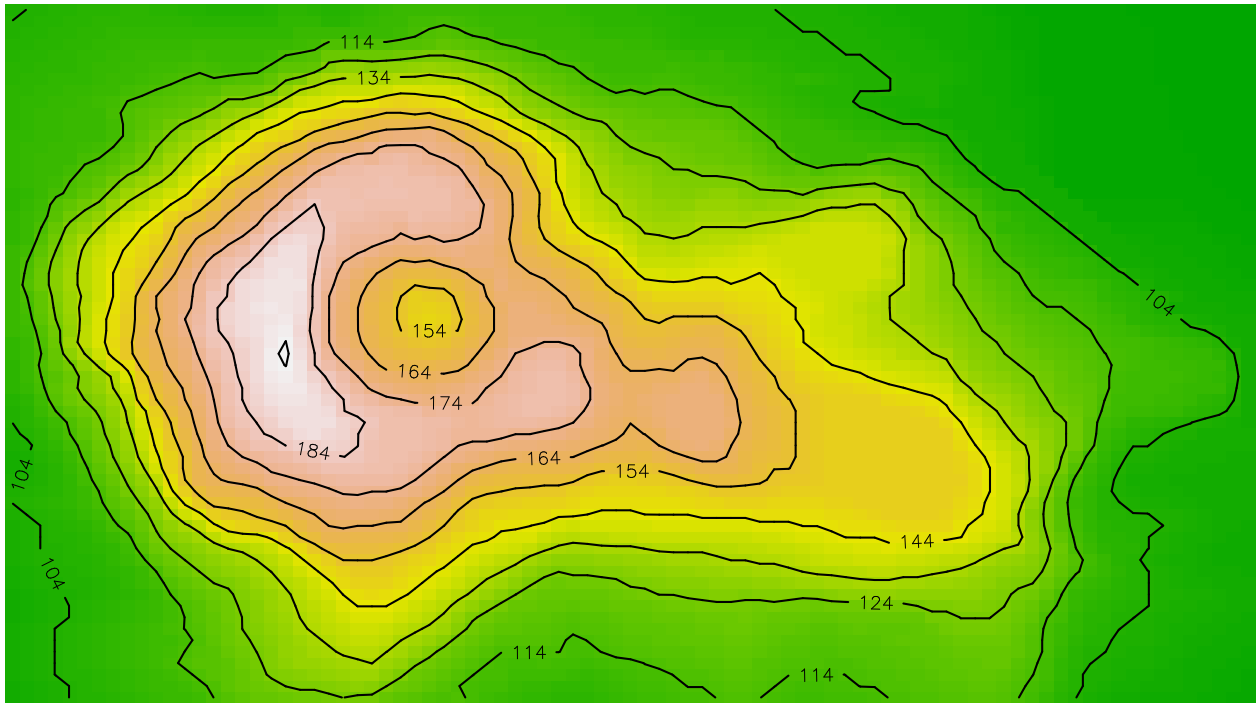
```
head(volcano)[,1:6]
```

	1	2	3	4	5	6
1	100.00	100.00	101.00	101.00	101.00	101.00
2	101.00	101.00	102.00	102.00	102.00	102.00
3	102.00	102.00	103.00	103.00	103.00	103.00
4	103.00	103.00	104.00	104.00	104.00	104.00
5	104.00	104.00	105.00	105.00	105.00	105.00
6	105.00	105.00	105.00	106.00	106.00	106.00

Grids: Visualization

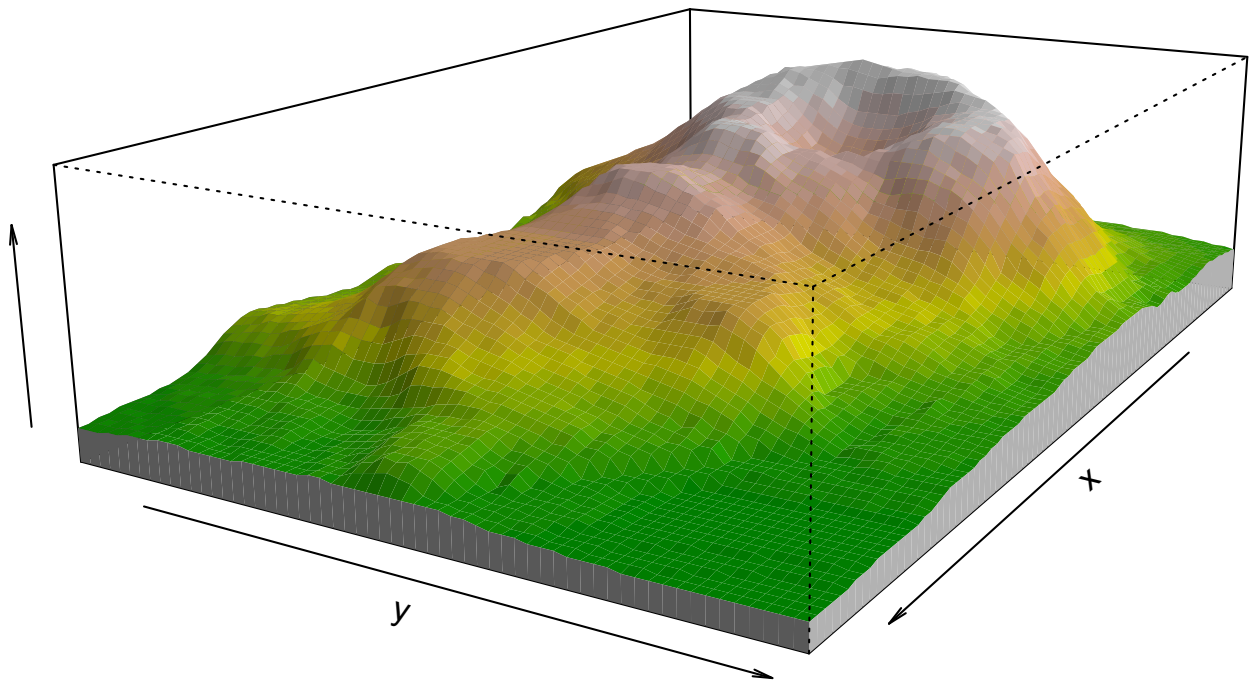
Grids can be visualized in two dimensions as contour plots, as images with a color gradient, or both.

```
image(x=10*(1:nrow(z)), y=10*(1:ncol(z)), z=volcano,
col=terrain.colors(100), axes=F)
contour(x=10*(1:nrow(z)), y=10*(1:ncol(z)), z=volcano,
levels=seq(from=min(z), to=max(z), by=10),axes=F, add=T)
```



Grids: Visualization

Grids can also be visualized in three dimensions with the `persp()` command and a grid of palette colors (similar to vector of colors from previous example).



Examples in R

Switch to R tutorial script. Section 1.