

Assignment: Spatial Diversity

Roy Z Moger-Reischer; Z620: Quantitative Biodiversity, Indiana University

03 February, 2017

OVERVIEW

This assignment will emphasize primary concepts and patterns associated with spatial diversity, while using R as a Geographic Information Systems (GIS) environment. Complete the assignment by referring to examples in the handout.

After completing this assignment you will be able to:

1. Begin using R as a geographical information systems (GIS) environment.
2. Identify primary concepts and patterns of spatial diversity.
3. Examine effects of geographic distance on community similarity.
4. Generate simulated spatial data.

Directions:

1. Change “Student Name” on line 3 (above) with your name.
2. Complete as much of the assignment as possible during class; what you do not complete in class will need to be done on your own outside of class.
3. Use the handout as a guide; it contains a more complete description of data sets along with the proper scripting needed to carry out the assignment.
4. Be sure to **answer the questions** in this assignment document. Space for your answer is provided in this document and indicated by the “>” character. If you need a second paragraph be sure to start the first line with “>”.
5. Before you leave the classroom, **push** this file to your GitHub repo.
6. When you are done with the assignment, **Knit** the text and code into an html file.
7. After Knitting, please submit the completed assignment by creating a **pull request** via GitHub. Your pull request should include this file *spatial_assignment.Rmd* and the html output of Knitr (*spatial_assignment.html*).

1) R SETUP

In the R code chunk below, provide the code to:

1. Clear your R environment
2. Print your current working directory,
3. Set your working directory to your “/Week4-Spatial” folder, and

```
rm(list=ls())  
getwd()  
#setwd
```

2) LOADING R PACKAGES

In the R code chunk below, do the following:

1. Install and/or load the following packages: **vegan**, **sp**, **gstat**, **raster**, **RgoogleMaps**, **maptools**, **rgdal**, **simba**, **gplots**, **rgeos**

```

#install.packages("vegan")
require(vegan)
#install.packages('sp') # Classes and methods for handling spatial data
require(sp)
#install.packages('gstat') # geostatistical analyses
require(gstat)
#install.packages('raster') # create a RasterLayer object
require(raster)
#install.packages('RgoogleMaps') #find Google server for static maps.
require(RgoogleMaps)
#install.packages('maptools') # Tools for geospatial data
require(maptools)
#install.packages('rgdal') # Geospatial Data Abstraction Library
require(rgdal)
#install.packages('simba')
require("simba")
#install.packages('gplots')
require("gplots")

? simba
? rgdal
? sp
#install.packages("ggplot2")
require(ggplot2)

```

Question 1: What are the packages `simba`, `sp`, and `rgdal` used for?

Answer 1:

`simba`: “Calculate similarity measures for binary data.” That is, if we have species data in binary format, we can calculate (dis)similarity metrics.

`sp`: This package helps turn R into a GIS software. It defines classes that represent point data, line data, shape data, coordinates data. It looks like this could help R interpret GIS shapefiles and the like.

`rgdal`: there was no automatic documentation for this package. It looks like this a package that interfaces with `sp` to be able to read shapefiles and interpret a coordinate reference system and projections.

3) LOADING DATA

In the R code chunk below, use the example in the handout to do the following:

1. Load the Site-by-Species matrix for the Indiana ponds datasets: `BrownCoData/SiteBySpecies.csv`
2. Load the Environmental data matrix: `BrownCoData/20130801_PondDataMod.csv`
3. Assign the operational taxonomic units (OTUs) to a variable ‘`otu.names`’
4. Remove the first column (i.e., site names) from the OTU matrix.

```

Ponds <- read.table(file = "BrownCoData/20130801_PondDataMod.csv", head = TRUE, sep = ",")
lats <- as.numeric(Ponds[, 3]) # latitudes (north and south)
lons <- as.numeric(Ponds[, 4]) # longitudes (east and west)
OTUs <- read.csv(file = "BrownCoData/SiteBySpecies.csv", head = TRUE, sep = ",")
otu.names = names(OTUs)
OTUs<-as.data.frame(OTUs[-1])

```

```
str(OTUs)
S.obs <- function(x="") {
  rowSums(x>0) *1
}#for each row x, take the sum of columns for which x > 0
max(S.obs(OTUs))
```

Question 2a: How many sites and OTUs are in the SiteBySpecies matrix?

Answer 2a: There are 51 sites. There are 16384-1=16383 distinct OTUs.

Question 2b: What is the greatest species richness found among sites?

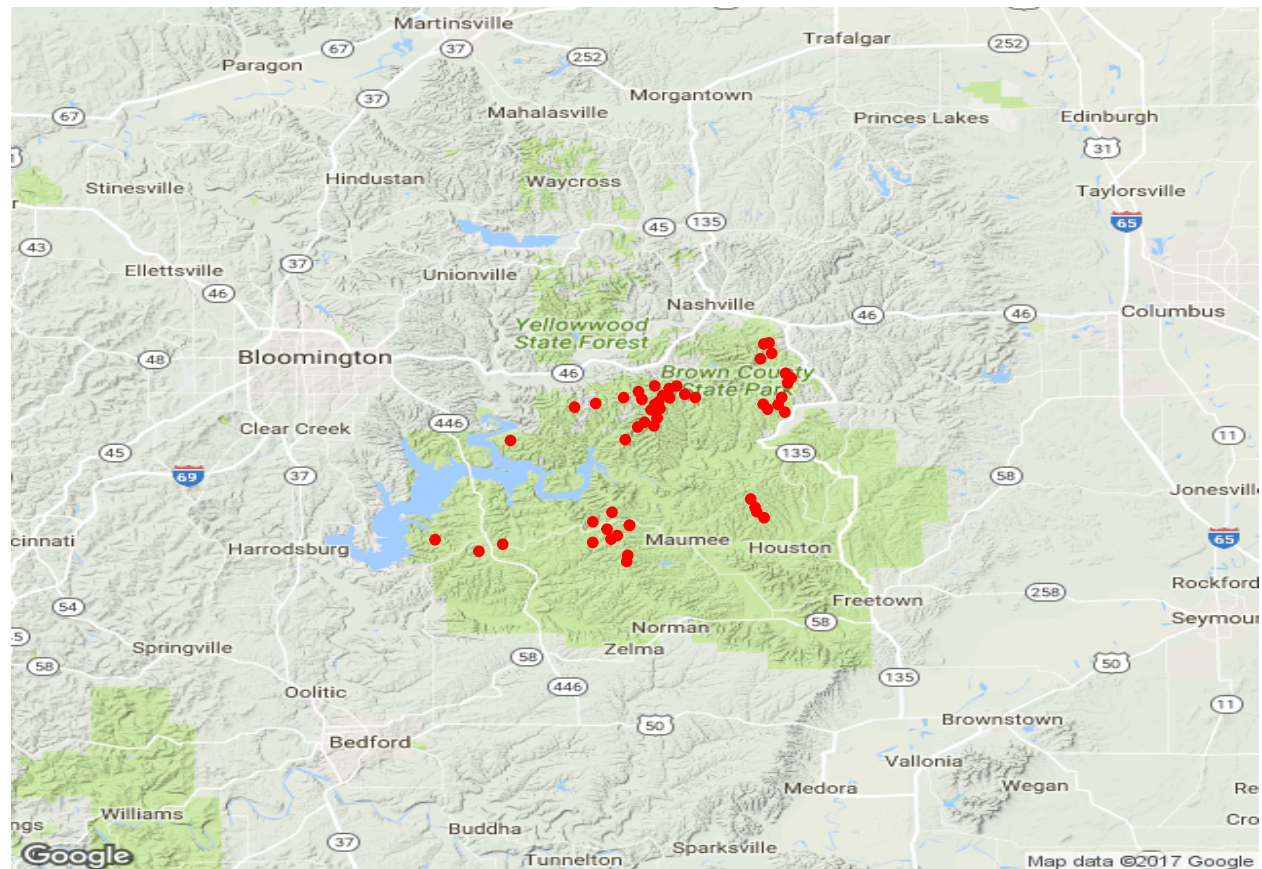
Answer 2b: 3659

4) GENERATE MAPS

In the R code chunk below, do the following:

1. Using the example in the handout, visualize the spatial distribution of our samples with a basic map in RStudio using the `GetMap` function in the package `RgoogleMaps`. This map will be centered on Brown County, Indiana (39.1 latitude, -86.3 longitude).

```
newmap <- GetMap(center = c(39.1,-86.3), zoom = 10, destfile = "PondsMap.png", maptype="terrain")
PlotOnStaticMap(newmap, zoom = 10, cex = 2, col = 'blue') # Plot map in RStudio
PlotOnStaticMap(newmap, lats, lons, cex = 1, pch = 20, col = 'red', add = TRUE)
```



Question 3: Briefly describe the geographical layout of our sites.

Answer 3: There are 3 main spatial clusters of sites (north central, south central, northeast). There are also small clusters of sites to the southeast and southwest, with one single site located in the northwest. Some are nearer to water than others.

In the R code chunk below, do the following:

1. Using the example in the handout, build a map by combining lat-long data from our ponds with land cover data and data on the locations and shapes of surrounding water bodies.

```
# 1. Import TreeCover.tif as a raster file.
Tree.Cover <- raster("TreeCover/TreeCover.tif")

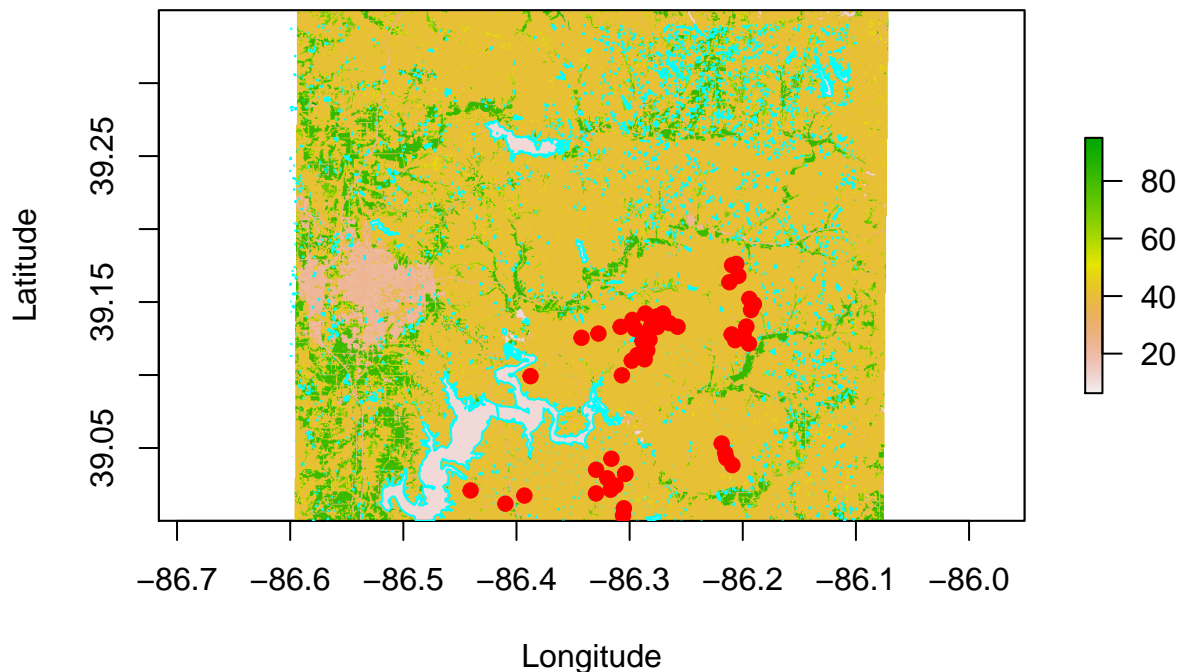
## NOTE: rgdal::checkCRSArgs: no proj_defs.dat in PROJ.4 shared files

# 2. Plot the % tree cover data
plot(Tree.Cover, xlab = 'Longitude', ylab = 'Latitude',
main = 'Map of geospatial data for % tree cover, \nwater bodies, and sample sites')

## NOTE: rgdal::checkCRSArgs: no proj_defs.dat in PROJ.4 shared files
## NOTE: rgdal::checkCRSArgs: no proj_defs.dat in PROJ.4 shared files

# 3. Import water bodies as a shapefile.
Water.Bodies <- readShapeSpatial("water/water.shp")
# 4. Plot the water bodies around our study area, i.e., Monroe County.
plot(Water.Bodies, border='cyan', axes = TRUE, add = TRUE)
# 5. Convert lat-long data for ponds to georeferenced points.
Refuge.Ponds <- SpatialPoints(cbind(lons, lats))
# 6. Plot the refuge pond locations
plot(Refuge.Ponds, line='r', col='red', pch = 20, cex = 1.5, add = TRUE)
```

**Map of geospatial data for % tree cover,
water bodies, and sample sites**



Question 4a: What are datums and projections?

Answer 4a: A datum defines the shape of Earth's surface. Then a projection maps that 3D

shape onto a 2D map.

5) UNDERSTANDING SPATIAL AUTOCORRELATION

Question 5: In your own words, explain the concept of spatial autocorrelation.

Answer 5: At its simplest, spatial autocorrelation means that the values for some phenotype for each of two points in space are not independent of the distance between the two points.

6) EXAMINING DISTANCE-DECAY

Question 6: In your own words, explain what a distance decay pattern is and what it reveals.

Answer 6: Two points that are near one another in space are more likely to share similar characteristics than are two points which are not near one another. If you compare, pairwise, the entire population of points, you should observe this pattern for all pairs of close points having a higher similarity score than the pairs of distant points. The rate (slope) of change of extent of autocorrelation could be displayed as a curve; this curve represents the “decay.”

In the R code chunk below, do the following:

1. Generate the distance decay relationship for bacterial communities of our refuge ponds and for some of the environmental variables that were measured. Note: You will need to use some of the data transformations within the *semivariogram* section of the handout.

```
# 1) Calculate Bray-Curtis similarity between plots using the `vegdist()` function
comm.dist <- 1 - vegdist(OTUs) # this is Bray-Curtis dissimilarity
# 2) Assign UTM latitude and longitude data to 'lats' and 'lons' variables
xy <- data.frame(pond.name = Ponds$Sample_ID, lats = Ponds$lat, lons = Ponds$long)
coordinates(xy) <- c("lats", "lons") # lat-lon dataframe
# Identify the current projection
proj4string(xy) <- CRS("+proj=longlat + datum=NAD83")

## Error in CRS("+proj=longlat + datum=NAD83"): major axis or radius = 0 or not given
# Transform the projection and data to UTM
UTM <- spTransform(xy, CRS("+proj=utm + zone=51 ellps=WGS84"))

## Error in CRS("+proj=utm + zone=51 ellps=WGS84"): major axis or radius = 0 or not given
UTM <- as.data.frame(UTM)

## Error in as.data.frame(UTM): object 'UTM' not found
Ponds$lats_utm <- UTM[,2] # latitude data

## Error in eval(expr, envir, enclos): object 'UTM' not found
Ponds$lons_utm <- UTM[,3] # longitude data

## Error in eval(expr, envir, enclos): object 'UTM' not found
coordinates(Ponds) = ~lats_utm+lons_utm

## Error in eval(expr, envir, enclos): object 'lats_utm' not found
vgm <- variogram(TDS~1, data=Ponds)

## Error in .local(obj, ...): cannot derive coordinates from non-numeric matrix
```



```

vgm.fit = fit.variogram(vgm, model = vgm(1, "Sph", 900, 1, fit.sill=F, fit.range=F))

## Error in fit.variogram(vgm, model = vgm(1, "Sph", 900, 1, fit.sill = F, : object should be of class
plot(vgm, vgm.fit)

## Error in plot(vgm, vgm.fit): object 'vgm.fit' not found

lats <- as.numeric(Ponds$lats_utm) # latitude data
lons <- as.numeric(Ponds$lons_utm) # longitude data

# 3) Calculate geographic distance between plots and assign to the variable 'coord.dist'
coord.dist <- dist(as.matrix(lats, lons)) # for each lat & its corresponding lon, calculate the orthodrom
# 4) Transform environmental data to numeric type, and assign to variable 'x1'
x1 <- as.numeric(Ponds$"SpC")
# 5) Using the `vegdist()` function in `simba`, calculate the Euclidean distance between the plots for
env.dist <- vegdist(x1, "euclidean")
# 6) Transform all distance matrices into database format using the `liste()` function in `simba`:
comm.dist.ls <- liste(comm.dist, entry="comm") # must learn more abt "entry"
env.dist.ls <- liste(env.dist, entry="env")
coord.dist.ls <- liste(coord.dist, entry="dist")

## Error in data.frame(NBX, NBY, obj.v): arguments imply differing number of rows: 1, 0
# 7) Create a data frame containing similarity of the environment and similarity of community.
df <- data.frame(coord.dist.ls, env.dist.ls[,3], comm.dist.ls[,3])

## Error in data.frame(coord.dist.ls, env.dist.ls[, 3], comm.dist.ls[, 3]): object 'coord.dist.ls' not
# 8) Attach the columns labels 'env' and 'struc' to the dataframe you just made.
names(df)[4:5] <- c("env", "struc")

## Error in names(df)[4:5] <- c("env", "struc"): names() applied to a non-vector
attach(df)

## Error in attach(df): 'attach' only works for lists, data frames and environments
# 9) After setting the plot parameters, plot the distance-decay relationships, with regression lines in
par(mfrow=c(1, 2), pty="s")
plot(env, struc, xlab="Environmental Distance", ylab="1 - Bray-Curtis",
main = "Environment", col='SteelBlue')

## Error in plot(env, struc, xlab = "Environmental Distance", ylab = "1 - Bray-Curtis", : object 'env'
OLS <- lm(struc ~ env)

## Error in eval(expr, envir, enclos): object 'struc' not found
OLS # print regression results to the screen

## Error in eval(expr, envir, enclos): object 'OLS' not found
abline(OLS, col="red4")

## Error in abline(OLS, col = "red4"): object 'OLS' not found
plot(dist, struc, xlab="Geographic Distance", ylab="1 - Bray-Curtis",
main="Community\nComposition", col='darkorchid4')

## Error in plot(dist, struc, xlab = "Geographic Distance", ylab = "1 - Bray-Curtis", : object 'struc'

```

```

OLS <- lm(struc ~ dist)

## Error in eval(expr, envir, enclos): object 'struc' not found
OLS # print regression results to the screen

## Error in eval(expr, envir, enclos): object 'OLS' not found
abline(OLS, col="red4")

## Error in abline(OLS, col = "red4"): object 'OLS' not found
# 10) Use `simba` to calculate the difference in slope or intercept of two regression lines
diffslope(env, struc, dist, struc)

## Error in mean(y1): object 'struc' not found

```

Question 7: What can you conclude about community similarity with regards to environmental distance and geographic distance?

Answer 7:

7) EXAMINING SPECIES SPATIAL ABUNDANCE DISTRIBUTIONS

Question 8: In your own words, explain the species spatial abundance distribution and what it reveals.

Answer 8:

In the R code chunk below, do the following:

1. Define a function that will generate the SSAD for a given OTU.
2. Draw six OTUs at random from the IN ponds dataset and plot their SSADs as kernel density curves. Use **while loops** and **if** statements to accomplish this.

```

# 1. Define an SSAD function

# 2. Set plot parameters

# 3. Declare a counter variable

# 4. Write a while loop to plot the SSADs of six species chosen at random

```

8) UNDERSTANDING SPATIAL SCALE

Many patterns of biodiversity relate to spatial scale.

Question 9: List, describe, and give examples of the two main aspects of spatial scale

Answer 9:

9) CONSTRUCTING THE SPECIES-AREA RELATIONSHIP

Question 10: In your own words, describe the species-area relationship.

Answer 10:

In the R code chunk below, provide the code to:

1. Simulate the spatial distribution of a community with 100 species, letting each species have between 1 and 1,000 individuals.

```
# 1. Declare variables to hold simulated community and species information
```

```
# 2. Populate the simulated landscape
```

While consult the handout for assistance, in the R chunk below, provide the code to:

1. Use a nested design to examine the SAR of our simulated community.
2. Plot the SAR and regression line.

```
# 1. Declare the spatial extent and lists to hold species richness and area data
```

```
# 2. Construct a 'while' loop and 'for' loop combination to quantify the numbers of species for progress
```

```
# 3. Be sure to log10-transform the richness and area data
```

In the R code chunk below, provide the code to:

1. Plot the richness and area data as a scatter plot.
2. Calculate and plot the regression line
3. Add a legend for the z-value (i.e., slope of the SAR)

Question 10a: Describe how richness relates to area in our simulated data by interpreting the slope of the SAR.

Answer 10a:

Question 10b: What does the y-intercept of the SAR represent?

Answer 10b:

SYNTHESIS

Load the dataset you are using for your project. Plot and discuss either the geographic Distance-Decay relationship, the SSADs for at least four species, or any variant of the SAR (e.g., random accumulation of plots or areas, accumulation of contiguous plots or areas, nested design).