Assignment: Spatial Diversity

Katie Beidler; Z620: Quantitative Biodiversity, Indiana University

06 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 refering 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 wit 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

```r
rm(list=ls())
getwd()
setwd("/Users/bhbeidler/GitHub/QB2017_Beidler/Week4-Spatial")
```

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`

```
clr = function() {
  ENV = globalenv()
  ll = ls(envir = ENV)
  ll = ll[ll != "clr"]
  rm(list = ll, envir = ENV)
}
require(vegan)
require(sp)
require(gstat)
require(raster)
require(RgoogleMaps)
require(maptools)
require(rgdal)
require(rgeos)
require(simba)
require(gplots)
```

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

> ***Answer 1***: The Simba package uses a collection of functions for similarity analysis of vegetation data, as well as, functions for reshaping species lists into matrices. The sp function creates classes for spatial data and provides utility functions, e.g. for plotting data as maps, spatial selection, as well as methods for retrieving coordinates, for subsetting, print, summary, etc. Rgdal is a package that helps you read in vector-based spatial data.

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 = ",")
# remove first column (site names)
OTUs = as.data.frame(OTUs[-1])

# To get S, we sum the number of non-zero abundances for a site
S.obs = function(x = ""){ rowSums(x > 0) * 1}
otu.names = names(OTUs) # Get the names of the OTUs
Ponds$N = as.vector(rowSums(OTUs)) # numbers of reads

Ponds$S = S.obs(OTUs) # number of species OTUs
```

```
max(Ponds$S)
Ponds$H = as.vector(diversity(OTUs, index = "shannon"))
# To get De at each site, we divide Simpson's Diversity by OTU site richness
Ponds$D = as.vector(diversity(OTUs, index = "invsimpson")/Ponds$S)
```

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

> ***Answer 2a***: There are 51 sites and 16,383 OTUs.

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

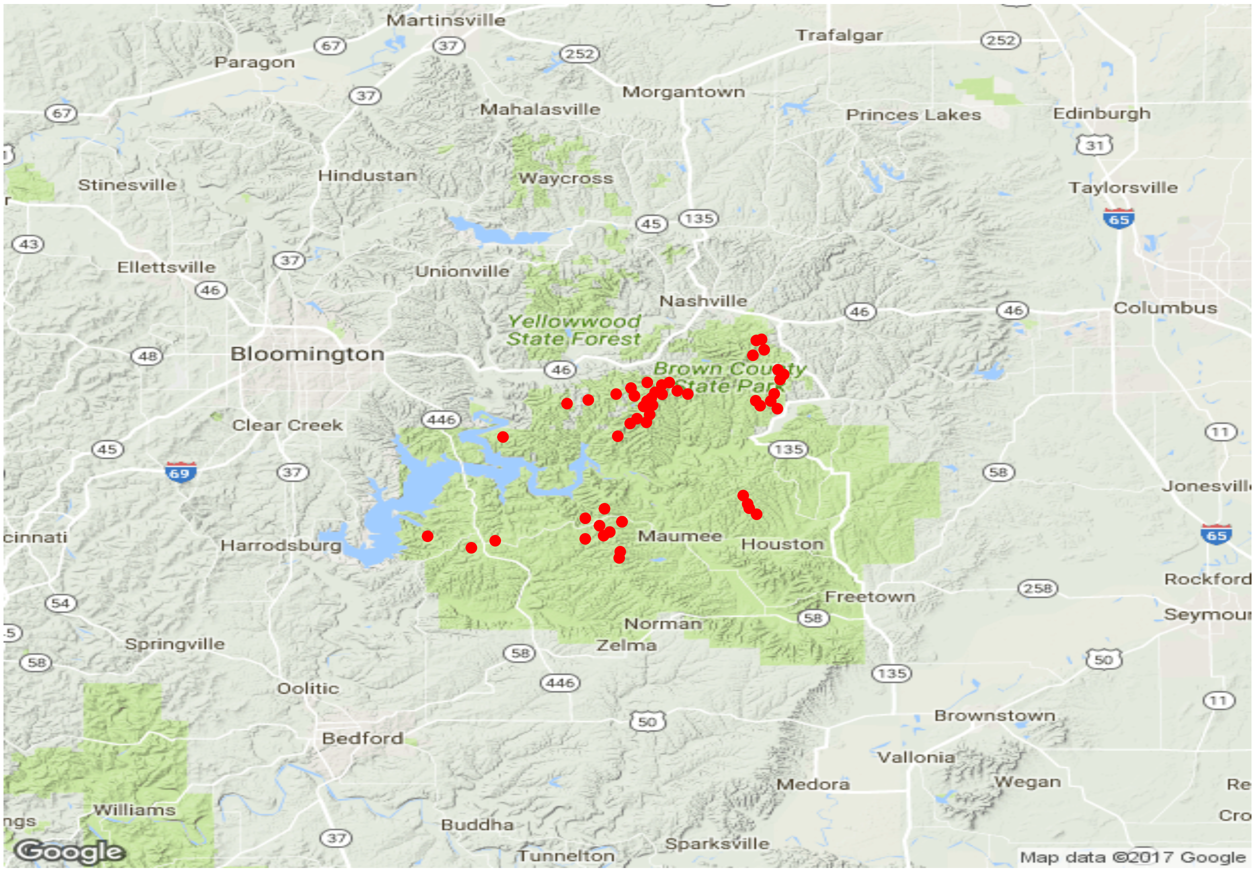
> ***Answer 2b***: 3,659 is the greatest species richness found at site 47.

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')
PlotOnStaticMap(newmap, lats, lons, cex = 1, pch = 20, col =  'red' , add = TRUE)
```



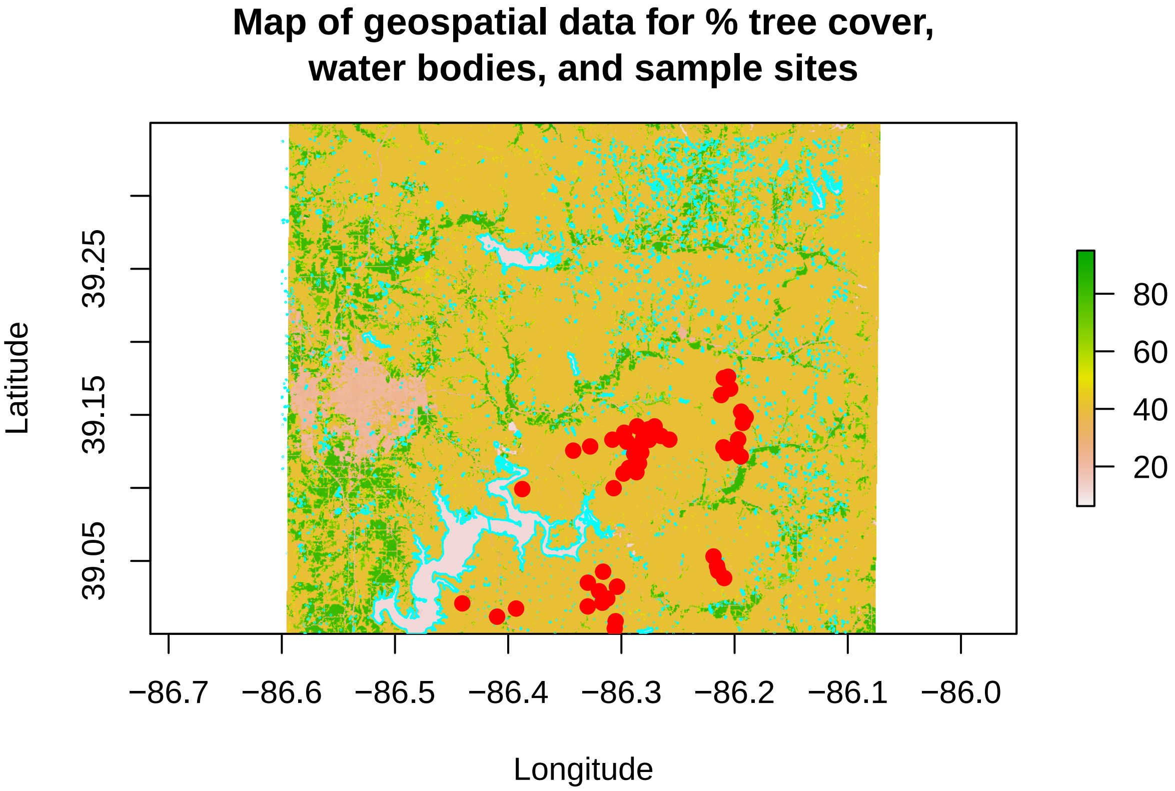
Question 3: Briefly describe the geographical layout of our sites.

> ***Answer 3***: There are five clusters of sites located close to Lake Monroe in the Brown County State park region.

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



Question 4a: What are datums and projections?

> ***Answer 4a***: A datum is a model for Earth's shape and a projection is the way in which coordinates on a sphere are projected onto a 2-D surface. A certain amount of distortian is associated with each projection- so it is important to choose the projection that is most appropriate for the spatial features you are trying to represent.

5) UNDERSTANDING SPATIAL AUTOCORRELATION

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

> ***Answer 5***: Spatial autocorrelation is the tendency of two individuals that are closer in space to be more similar to one another than individuals that are further apart- resulting in clustering or gradient patterns.

6) EXAMINING DISTANCE-DECAY

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

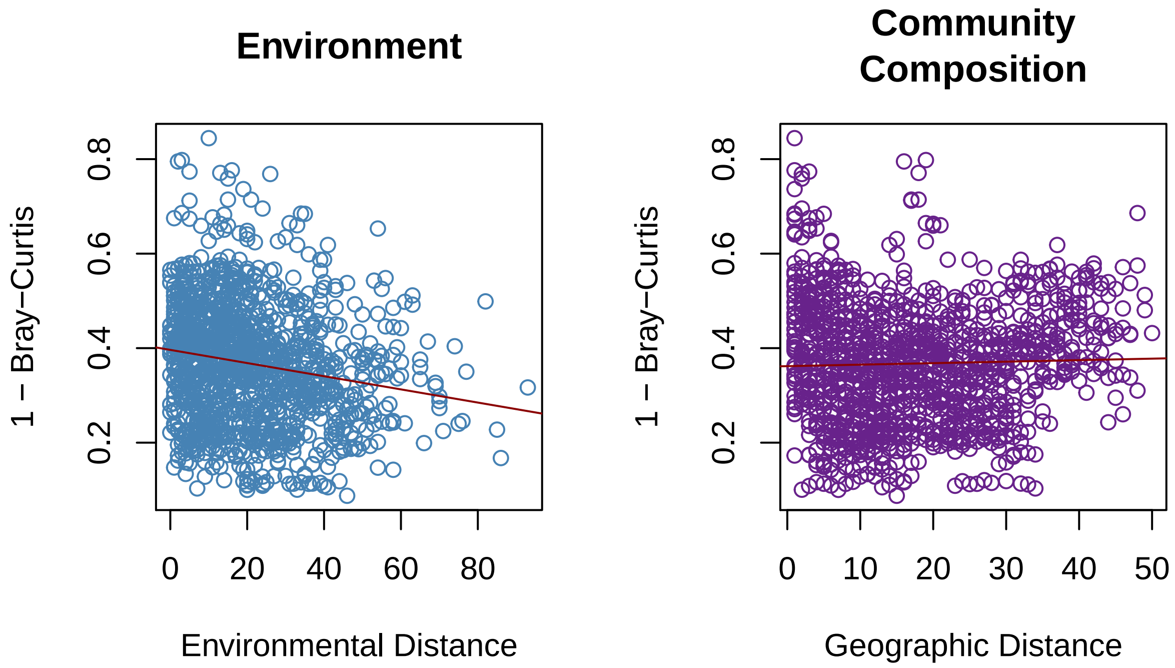
> ***Answer 6***: The distance-decay relationship shows the pattern of autocorrelation or how Bray-curtis dissimilarity decreases as geographic distance increases. If there is no distance decay relationship, sites are not spatially autocorrelated and there is no distance at which you can't compare species.

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.

```r
# Construct a new dataframe for coordinates
xy = data.frame(env = Ponds$TDS, pond.name = Ponds$Sample_ID, lats = Ponds$lat, lons = Ponds$long)
coordinates(xy) = ~lats+lons # Transform 'xy' into a spatial points dataframe
# Identify the current projection (i.e., lat-long) and datum (NAD83)
proj4string(xy) = CRS("+proj=longlat +datum=NAD83") # coordinate reference system (CRS)
UTM = spTransform(xy, CRS("+proj=utm +zone=51 +ellps=WGS84"))
UTM = as.data.frame(UTM)
xy$lats_utm = UTM[,2] # lattitude data according to UTM
xy$lons_utm = UTM[,3] # longitude data according to UTM

# 1) Calculate Bray-Curtis similarity between plots using the `vegdist()` function
comm.dist = 1 - vegdist(OTUs) # Bray-Curtis similarity between the plots
# 2) Assign UTM lattitude and longitude data to 'lats' and 'lons' variables
lats = as.numeric(xy$lats_utm) # lattitude data
lons = as.numeric(xy$lons_utm) # longitude data
# 3) Calculate geographic distance between plots and assign to the variable 'coord.dist'
coord.dist = dist(as.matrix(lats, lons)) # geographical distance between plots
# 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")
env.dist.ls = liste(env.dist, entry="env")
coord.dist.ls = liste(coord.dist, entry="dist")
# 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])
# 8) Attach the columns labels 'env' and 'struc' to the dataframe you just made.
names(df)[4:5] = c("env", "struc")
```

```r
attach(df)
# 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')

OLS = lm(struc ~ env)
OLS # print regression results to the screen
abline(OLS, col="red4")
plot(dist, struc, xlab="Geographic Distance", ylab="1 - Bray-Curtis", main="Community\nComposition", co
OLS = lm(struc ~ dist)
abline(OLS, col="red4")
```



```r
# 10) Use `simba` to calculates the difference in slope or intercept of two regression lines
diffslope(env, struc, dist, struc)
```

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

> ***Answer 7***: Bacterial community similarity decreases with increasing environmental distance and does not differ with respect to geographic distance. In other words near environments (refuge ponds) are more similar with respect to species of bacteria than far ones. However, bacterial communities close to one another do not differ from communities that are farther away.

7) EXAMINING SPECIES SPATIAL ABUNDANCE DISTRIBUTIONS

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

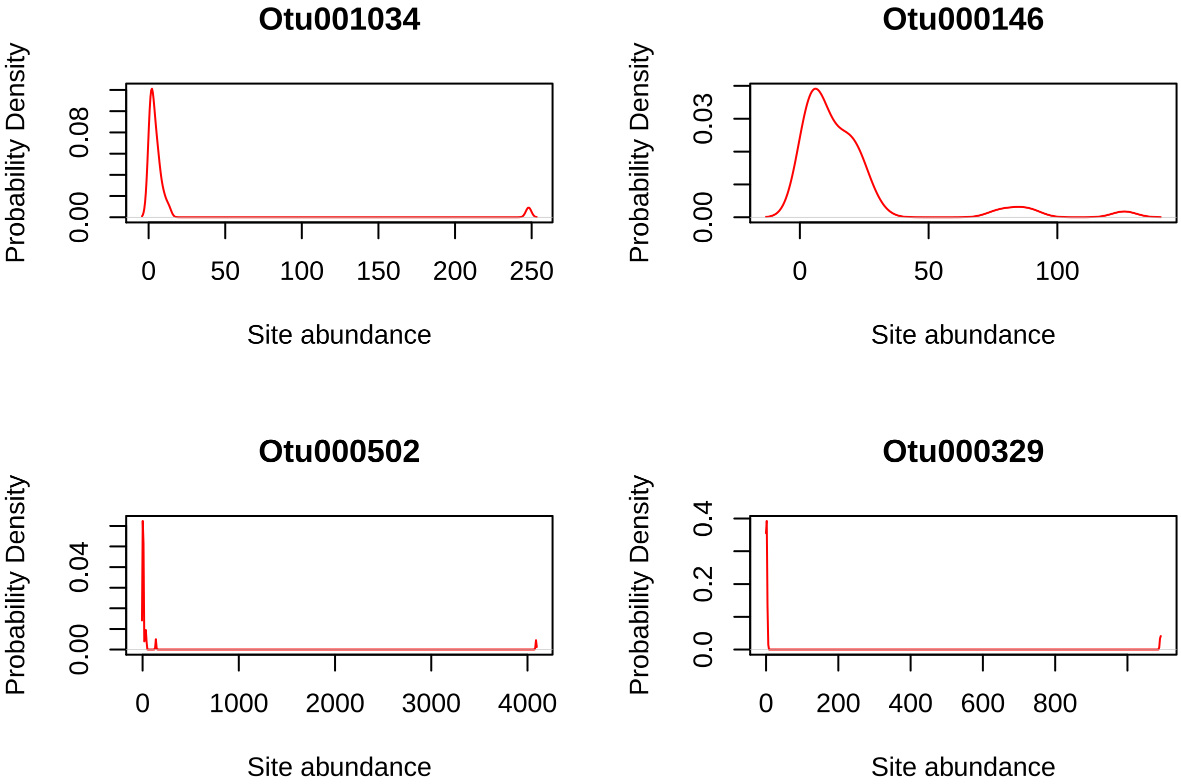
> ***Answer 8***: The spatial abundance distribution (SSAD) shows how individuals are distributed across a given area. The SSAD reveals the probability of finding individuals at a particular

abundance. The SSAD reflects commonness and rarity and demonstrates that a species is only abundant in a few of the sites where it is found.

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 and plot their SSADs as kernel density curves. Use **while loops** and **if** statements to accomplish this.

```r
# 1. Define an SSAD function
ssad = function(x){
  ad = c(2, 2)
  ad = OTUs[, otu]
  ad = as.vector(t(x = ad))
  ad = ad[ad > 0]
}
# 2. Set plot parameters
par(mfrow=c(2, 2))
# 3. Declare a counter variable
ct = 0
# 4. Write a while loop to plot the SSADs of six species chosen at random
while (ct < 4){
otu = sample(1:length(OTUs), 1)
ad = ssad(otu)
if (length(ad) > 10 & sum(ad > 100)){
ct = ct + 1
plot(density(ad), col = 'red', xlab='Site abundance',
ylab='Probability Density', main = otu.names[otu])
}
}
```



```
dev.off()
```

```
## null device
##           1
```

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***: The two main aspects of spatial scale are extent and grain. Extent is the overall sampling area or the greatest distance considered in an investigation. For a 25 ha forest research plot made up of 20 x 20 m subplots, the extent would be 25 ha and the sampling resolution or grain would be the 20 x 20 m unit.

9) CONSTRUCTING THE SPECIES-AREA RELATIONSHIP

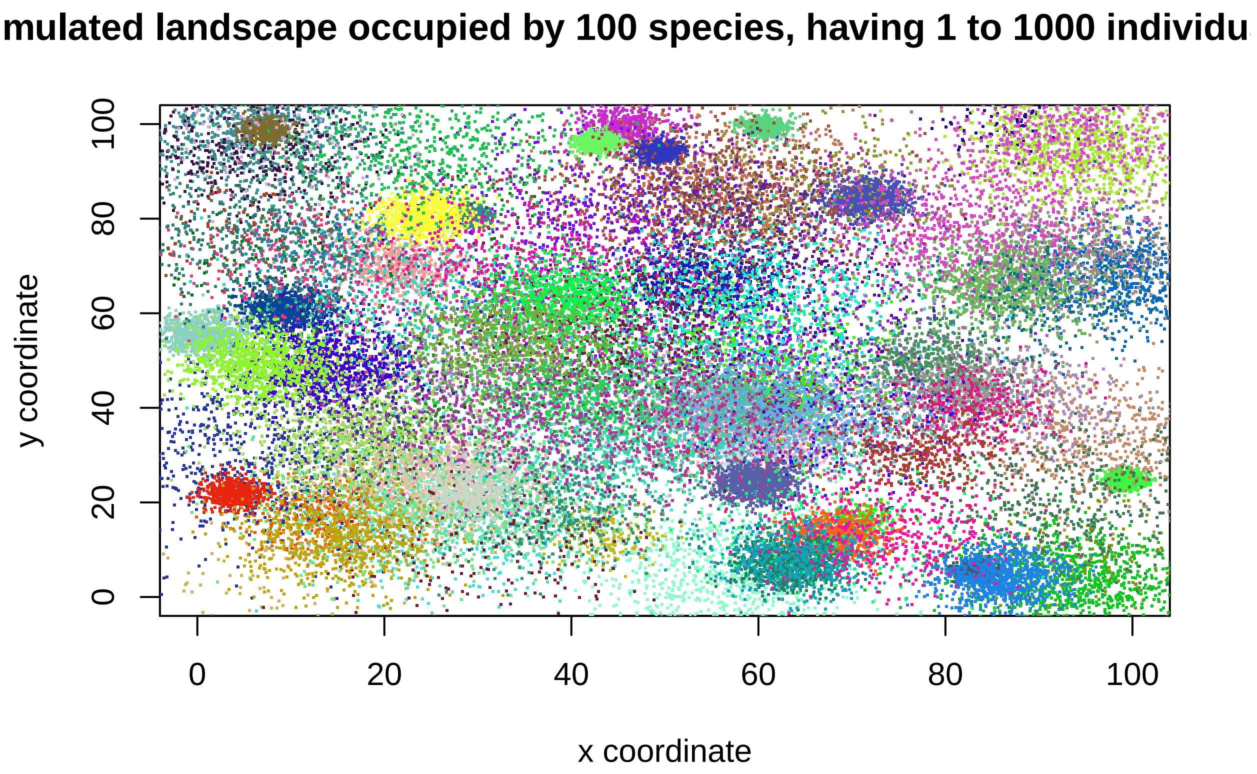
Question 10: In your own words, describe the species-area relationship.

> ***Answer 10***: The species-area relationship describes the general pattern of increase in richness with increasing sampling area.

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
community = c()
# an initial empty community
species = c() # with zero species

# 2. Populate the simulated landscape
plot(0, 0, col= "white" , xlim = c(0, 100), ylim = c(0, 100), xlab= 'x coordinate' , ylab= 'y coordinat
while (length(community) < 100){
  std = runif(1, 1, 10)
  ab = sample(1000, 1)
  x = rnorm(ab, mean = runif(1, 0, 100), sd = std)
  y = rnorm(ab, mean = runif(1, 0, 100), sd = std)
  color = c(rgb(runif(1),runif(1),runif(1)))
  points(x, y, pch=".", col=color)
  species = list(x, y, color)
  community[[length(community)+1]] = species
}
```



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.

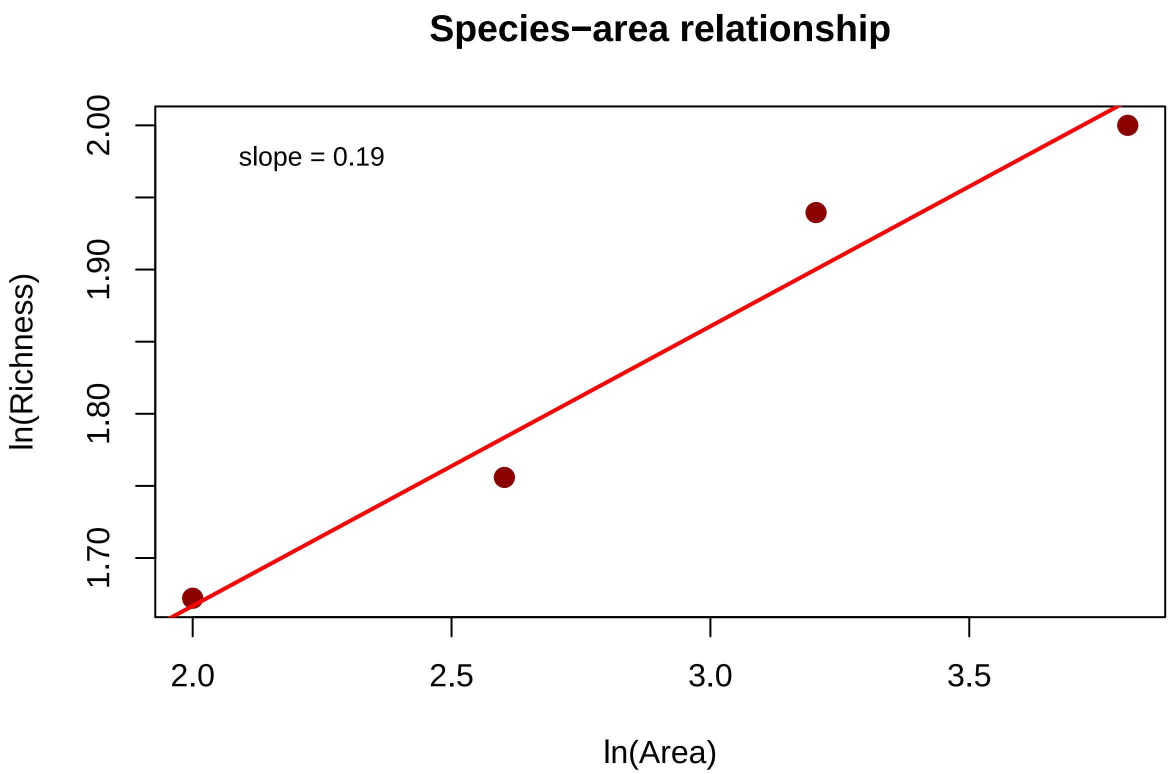
```
# 1. Declare the spatial extent and lists to hold species richness and area data
lim = 10 # smallest spatial extent. This also equals the spatial grain.
S.list = c() # holds the number of species
A.list = c() # holds the spatial scales

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

```
while (lim <= 100) {  # while the spatial extent is less than or equal to 100
  S = 0 # initiate richness at zero
  for (sp in community){
    xs = sp[[1]] # assign the x coordinates
    ys = sp[[2]] # assign the y coordinates
    sp.name = sp[[3]] # assign the species name
    xy.coords = cbind(xs, ys)
    for (xy in xy.coords){
      if (max(xy) <= lim){
        S = S + 1
          break
    }
  }
}
# 3. Be sure to log10-transform the richness and area data
S.list = c(S.list, log10(S))
A.list = c(A.list, log10(lim^2))
lim = lim * 2
}
```

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)

```
results = lm(S.list ~ A.list)
plot(A.list, S.list, col="dark red", pch=20, cex=2,
     main="Species-area relationship",
     xlab= 'ln(Area)' ,ylab= 'ln(Richness)')

abline(results, col="red", lwd=2)

int = round(results[[1]][[1]],2)
z = round(results[[1]][[2]],2)
legend(x=2, y=2, paste(c( 'slope =', z), collapse = " "), cex=0.8,
       box.lty=0)
```



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

> ***Answer 10a***: In our simulation, richness increased with increasing area. The slope for the the simulated data is z= 0.21, so as area increases from ~ 5 to 50, richness increases from ~ 3 to 9.

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

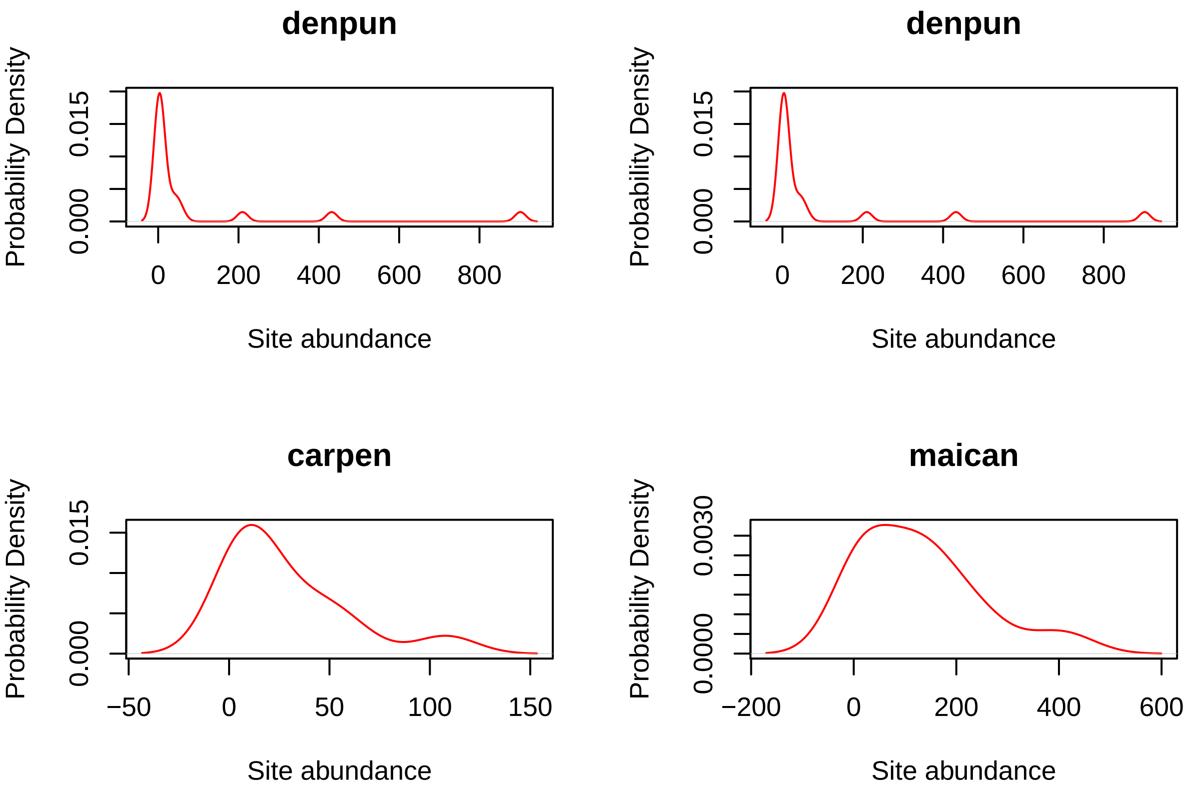
> ***Answer 10b***: The y-intercept of the SAR (c = 3.28 for the simulated data) represents a constant which equals the number of species that would exist if the habitat area was confined to one square unit. c can vary with respect to taxonomic group.

SYNTHESIS

Load the dataset you are using for your project. Plot and discuss either the geogrpahic 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).

```r
# Loading required packages
require("dplyr")

# Reading in plant diversity data
plant = read.csv("/Users/bhbeidler/GitHub/QB2017_DivPro/Data/HF_plants.csv")

# Subsetting the data for 2009
plant_09 = filter(plant, year == 2009)

# Separating out the treatments from the site by species matricies
```

```
plant_09_sbys = plant_09[ ,4:43]

# Getting the names of plant species
sp.names = names(plant_09_sbys)

# SSAD function for tree species
ssad = function(x){
  ad = c(2, 2)
  ad = plant_09_sbys[, sp]
  ad = as.vector(t(x = ad))
  ad = ad[ad > 0]
}
# 2. Set plot parameters
par(mfrow=c(2, 2))
# 3. Declare a counter variable
ct = 0

# 4. Write a while loop to plot the SSADs of 4 species chosen at random
while (ct < 4){
sp = sample(1:length(plant_09_sbys), 1)
ad = ssad(sp)
if (length(ad) > 10 & sum(ad > 100)){
ct = ct + 1
plot(density(ad), col = 'red', xlab='Site abundance',
ylab='Probability Density', main = sp.names[sp])
}
}
```



```
dev.off()
```

Synthesis: The purpose of this experiment was to test for the interactive effects of warming and nitrogen additions on plant diversity. To sample plants, investigators counted all stems in each 3m x 3m plot. There are a total of 24 plots (6 replicates for each treatment). Because the plots were located close to one another, examining the effects of geographic distance on community similarity may not be particularly interesting in this instance. Furthermore, we were unable to find geographic coordinates for each plot. Plotting the SSADS for 4 species (*Maianthemum canadense, Mitchella repens, Gaultheria procumbens, and Dennstaedtia punctilobula*) did provide some insight into whether or not the species were rare or abundant at the sites. (*Dennstaedtia punctilobula*) seemed to be the rarest of the 4 species- with lower probabilities of finding the species at higher abundances.