Assignment: Spatial Diversity

Evgeniya Polezhaeva; Z620: Quantitative Biodiversity, Indiana University

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

```
rm(list=ls())
getwd()
setwd("D:/Jane/GitHub/QB2017_Polezhaeva/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`

```
package.list <- c('vegan', 'sp', 'gstat', 'raster', 'RgoogleMaps', 'maptools', 'rgdal', 'simba', 'gplots
for (package in package.list) {
  if (!require(package, character.only=T, quietly=T)) {
    install.packages(package)
    library(package, character.only=T)
  } }
require("rgdal")
```

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

> ***Answer 1***: simba: Besides a function for the calculation of similarity measures with binary data (for instance presence/absence species data) the package contains some simple wrapper functions for reshaping species lists into matrices and vice versa and some other functions for further processing of similarity data. sp : This package provides S4 classes for importing, manipulating and exporting spatial data in R, and for methods including print/show, plot, subset, [, [[, $, names, dim, summary, and a number of methods specific to spatial data handling. rgdal : Provides bindings to Frank Warmerdam's Geospatial Data Abstraction Library (GDAL) (>= 1.6.3) and access to projection/transformation operations from the PROJ.4 library. The GDAL and PROJ.4 libraries are external to the package, and, when installing the package from source, must be correctly installed first. Both GDAL raster and OGR vector map data can be imported into R, and GDAL raster data and OGR vector data exported. Use is made of classes defined in the sp package. Windows and Mac Intel OS X binaries (including GDAL, PROJ.4 and Expat) are provided on CRAN.

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 = ",")
OTUs <- read.table( file = "BrownCoData/SiteBySpecies.csv", head = TRUE, sep = ",")
otu.names <- names(OTUs)
OTUs <- as.data.frame(OTUs[-1])
```

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

> ***Answer 2a***: 51 sites, 16383 species

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

> ***Answer 2b***: 3259

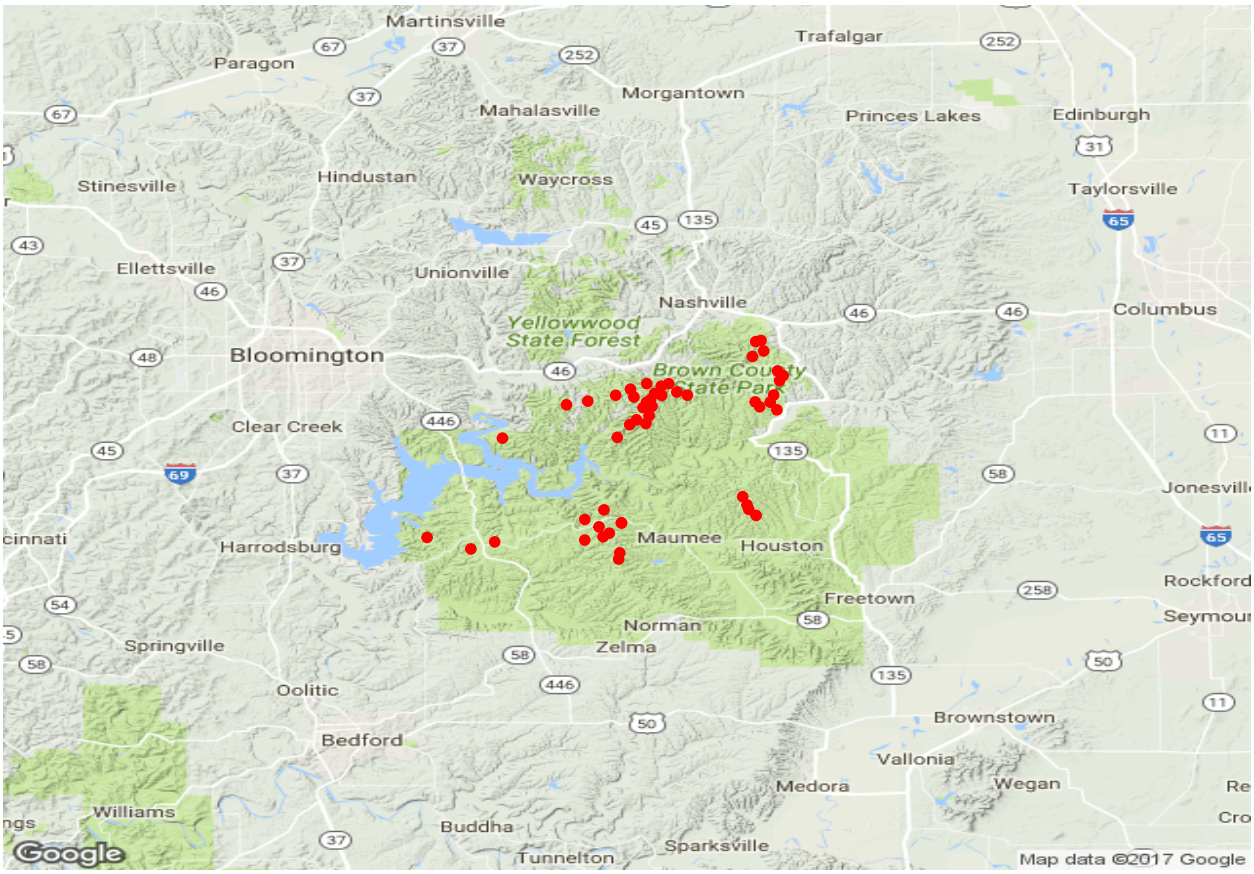
4) GENERATE MAPS

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).

```
require("RgoogleMaps")
lats <- as.numeric(Ponds[,3])
```

```
lons <- as.numeric(Ponds[,4])
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)
```



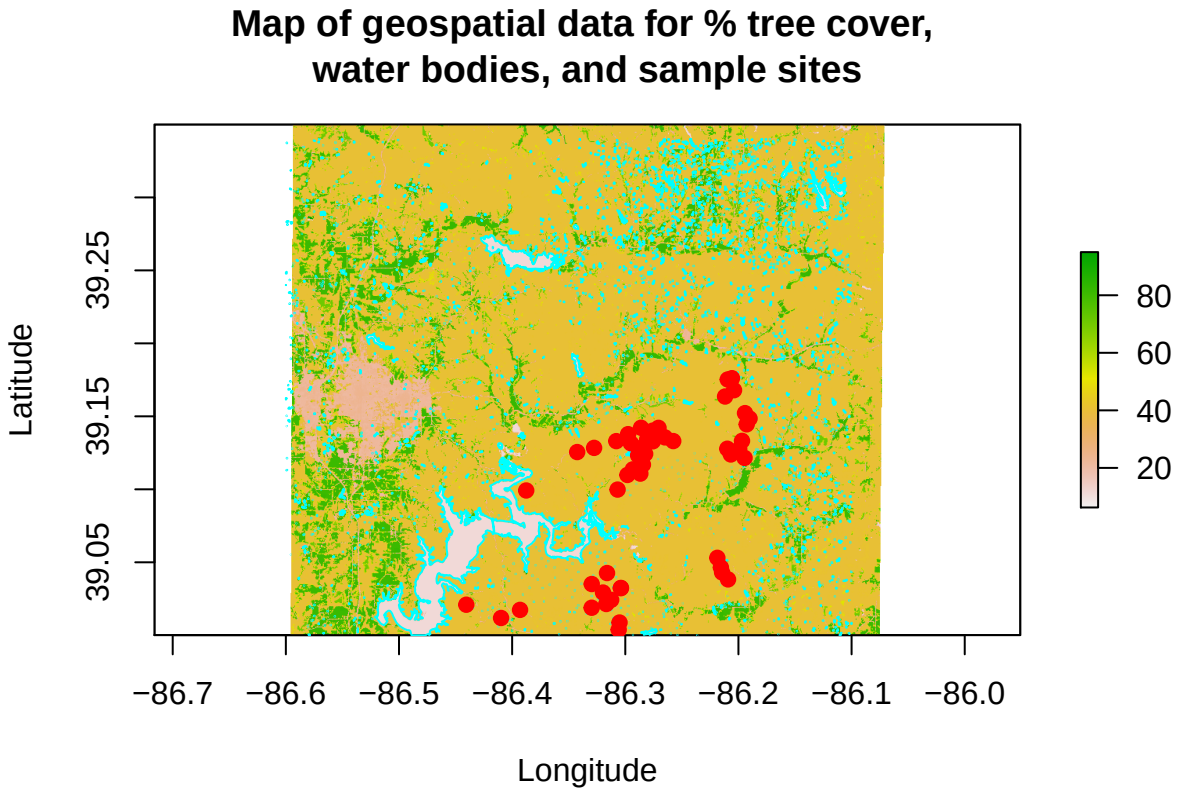
Question 3: Briefly describe the geographical layout of our sites.

> ***Answer 3***: All points are located near Bloomington, IN on State land, and forms 5 groups that are distributed across topolograficaly complex area.

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")
# 2. Plot the % tree cover data
plot(Tree.Cover, xlab = "Longitude", ylab = "Latitude", main = "Map of geospatial data for % tree cover
# 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 = T, add =T)
# 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 = T)
```



Question 4a: What are datums and projections?

> ***Answer 4a***: Datums are models for Earth's shape. Projections are the ways in which coordinates on a sphere are projected onto a 2-D surface

5) UNDERSTANDING SPATIAL AUTOCORRELATION

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

> ***Answer 5***: Spatially close sites tend to be related stronger than spatially farther ones.

6) EXAMINING DISTANCE-DECAY

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

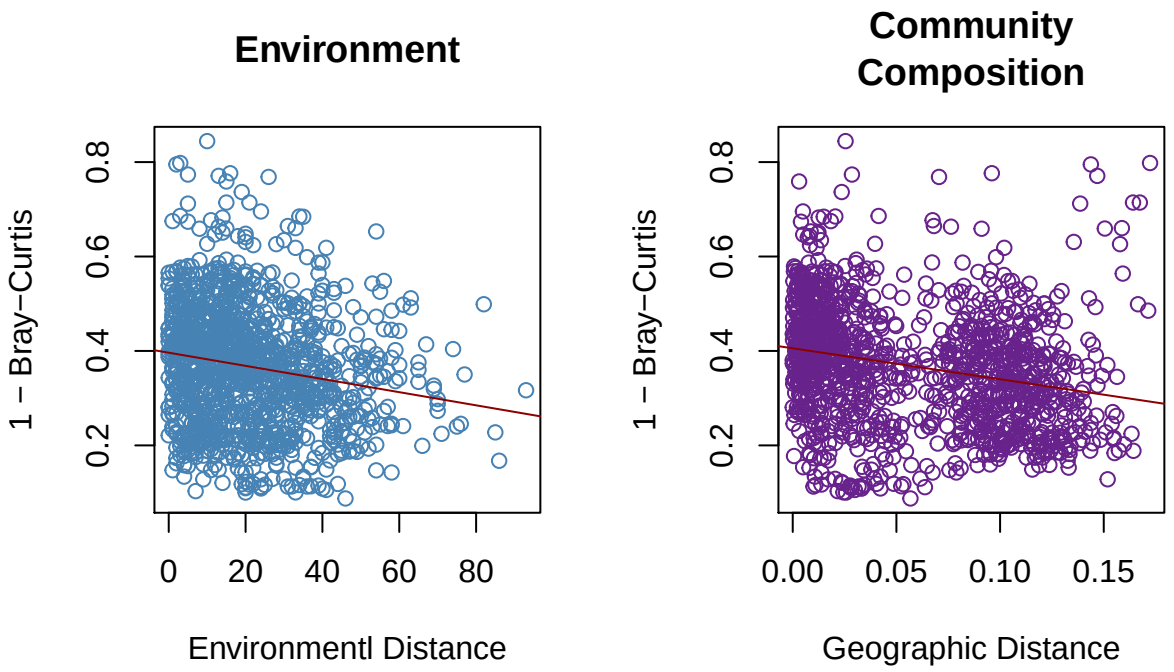
> ***Answer 6***: The geograhpic distance-decay relationship is a pattern of degreasing similarities between communities, environment, etc. with distance.

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.

```
xy <- data.frame(env = Ponds$TDS, pond.name = Ponds$Sample_ID, lats = Ponds$lat, lons = Ponds$long)

# 1) Calculate Bray-Curtis similarity between plots using the `vegdist()` function
```

```r
comm.dist <- 1 - vegdist(OTUs)
# 2) Assign UTM lattitude and longitude data to 'lats' and 'lons' variables
lats <- as.numeric(xy$lats)
lons <- as.numeric(xy$lons)
# 3) Calculate geographic distance between plots and assign to the variable 'coord.dist'
coord.dist <- dist(as.matrix(lats, lons))
# 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")
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 = "Environmentl Distance", ylab = "1 - Bray-Curtis", main = "Environment", col =
OLS <- lm(struc ~ env)
OLS
abline(OLS, col = "red4")
plot(dist, struc, xlab = "Geographic Distance", ylab = "1 - Bray-Curtis", main = "Community\nCompositio
OLS <- lm(struc ~ dist)
OLS
abline(OLS, col = "red4")
```




```
# 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 regars to environmental distance and geographic distance?

> ***Answer 7***: Community Bray-Curtis similarity decreasing with both environmental and geographic distance.

7) EXAMINING SPECIES SPATIAL ABUNDANCE DISTRIBUTIONS

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

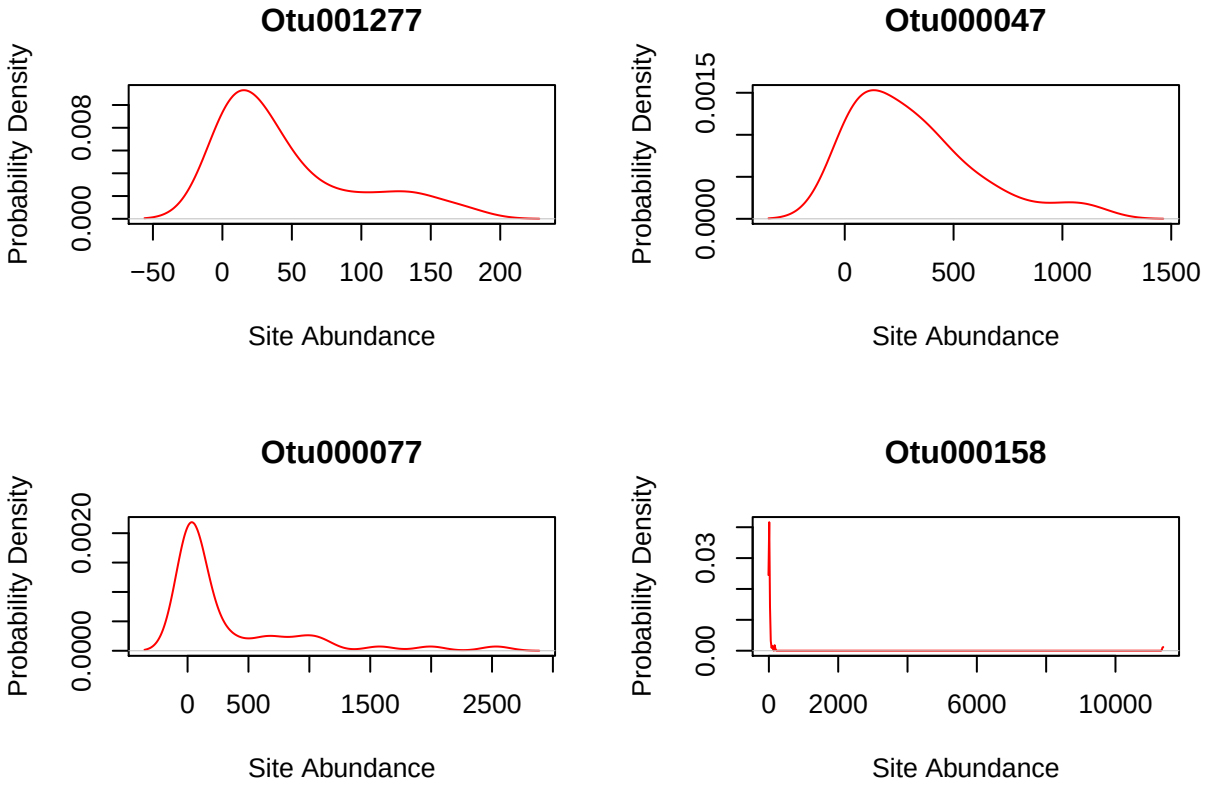
> ***Answer 8***: Species Spatial abundance distribution is graphic illustration of frequences (y-axis) at which individuals are found at given abundances (x-axis)

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.

```
# 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.nar
}}
```



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 components of spatial scale are extent and grain. Extent is the greatest distance considered in an observation or study (50 ha). Grain is the smallest or primary unit by which the extent is measured (1 ha).

9) CONSTRUCTING THE SPECIES-AREA RELATIONSHIP

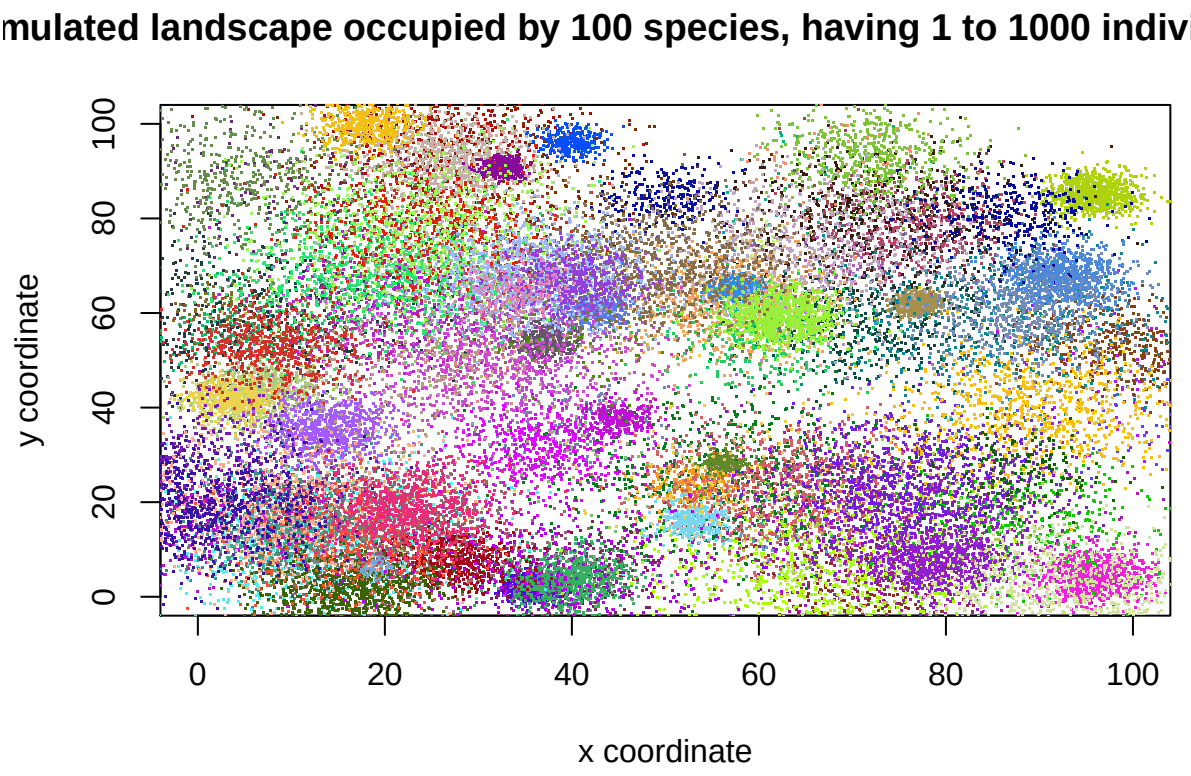
Question 10: In your own words, describe the species-area relationship.

> ***Answer 10***: SAR is a rate at which new species are discovered with increasing sampling area. Coefficients(c, z) change with scale, but species richness is proportional (c) to the power z of 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()
species <- c()
plot(0, 0, col='white', xlim = c(0, 100), ylim = c(0, 100), xlab='x coordinate', ylab='y coordinate', ma
# 2. Populate the simulated landscape
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:

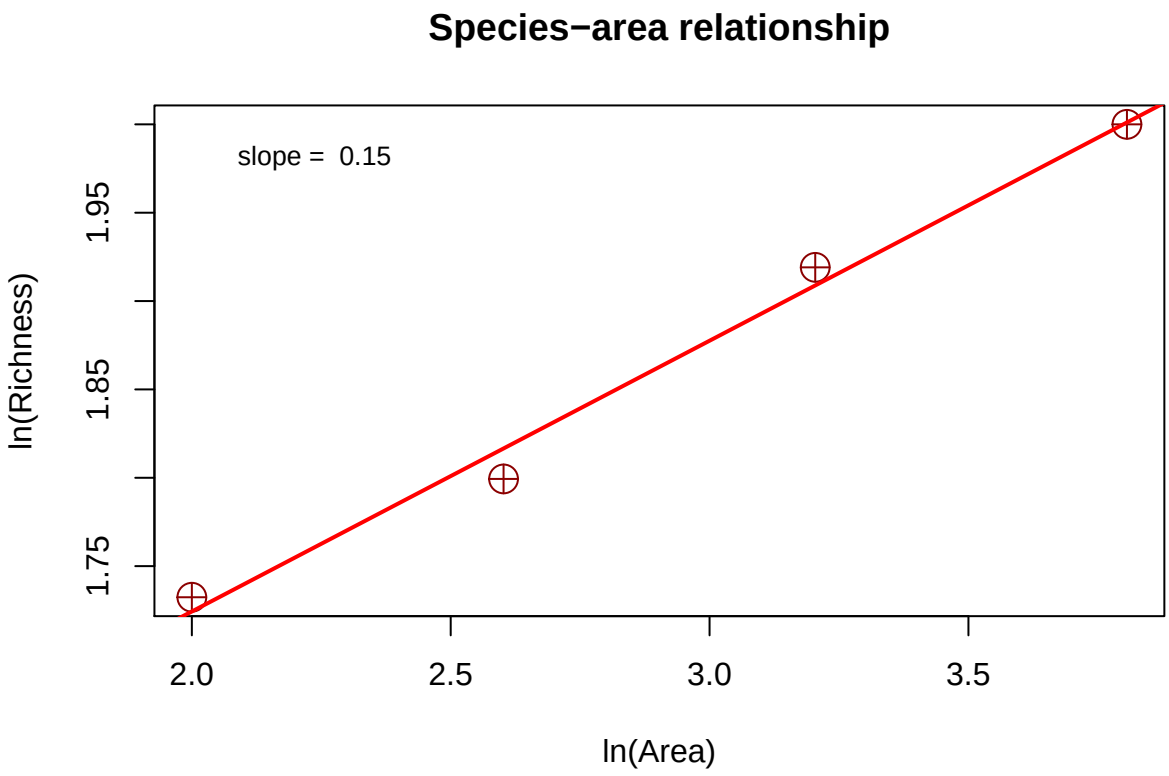
1. Use a nested design to examine the SAR of our simulated community.
2. Plot the SAR and regression line.

```
# 1. Declare the spatial extent and lists to hold species richness and area data
lim <- 10
S.list <- c()
A.list <- c()
# 2. Construct a 'while' loop and 'for' loop combination to quantify the numbers of species for progres
while (lim <= 100){
  S <- 0
  for (sp in community){
    xs <- sp[[1]]
    ys <- sp[[2]]
    sp.name <- sp[[3]]
    xy.coords <- cbind(xs, ys)
    for (xy in xy.coords){
      if (max(xy) <= lim){
        S <- S + 1
          break
      }
    }
  }
  S.list <- c(S.list, log10(S))
  A.list <- c(A.list, log10(lim^2))
  lim <- lim *2
}
# 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)

```
results <- lm(S.list ~ A.list)
plot(A.list, S.list, col = "dark red", pch = 10, cex = 2, main = "Species-area relationship", xlab = "l
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 logarithmic scale richness is proportional to sampling area (according to Arrhenius's power-law prediction)

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

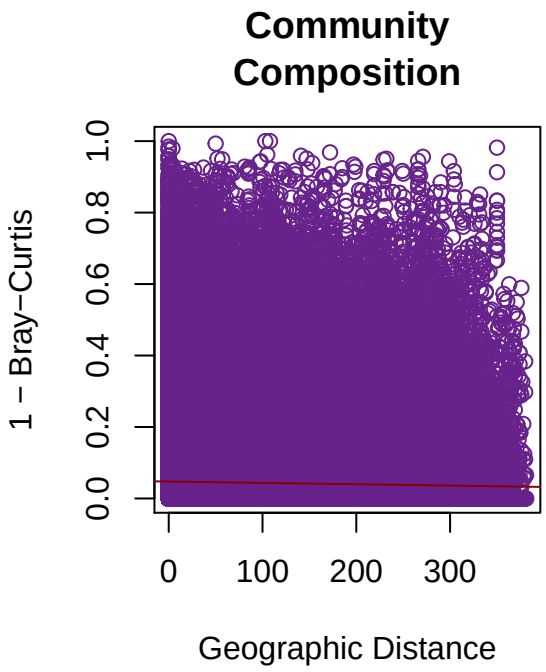
> ***Answer 10b***: The y-intercept of the SAR represent the natural logarithm of coefficient c from equetion log(S)= log(c)+zlog(A) (Amount of species on area equal to 1)

SYNTHESIS

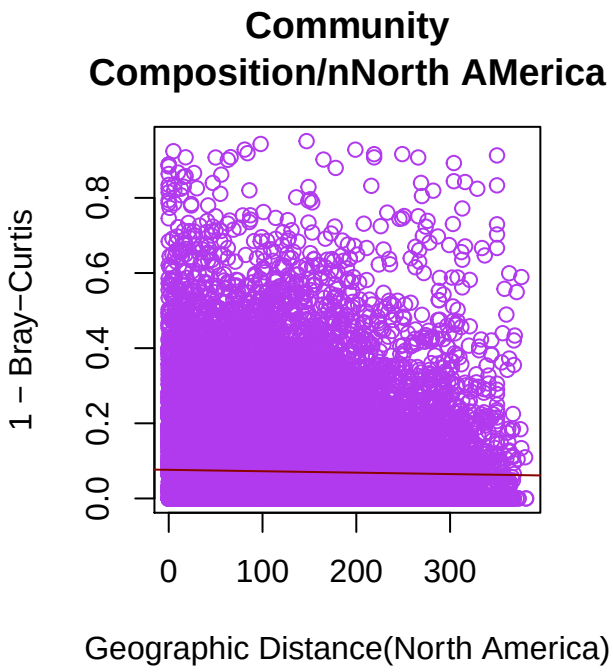
Load the dataset you are using for your project. Plot and discuss either the geogrpahic Distance-Decay relationship, the SSADs for at least four species, or any variant of the SAR (e.g., random accumulation of plots or areas, accumulation of contiguous plots or areas, nested design).

```
SbyS_mammals <- read.table("SbyS_mammals.txt", quote = , sep = "\t", header = TRUE, fill = TRUE)
env_mammals <- read.table("MCDB_sites.csv", quote = "", sep = ",", header = TRUE, fill = TRUE)
xy_mammals <- data.frame(siteID = env_mammals$Site_ID, lats = env_mammals$Latitude, lons = env_mammals$

# 1) Calculate Bray-Curtis similarity between plots using the `vegdist()` function
comm.dist_m <- 1 - vegdist(SbyS_mammals)
# 2) Assign UTM lattitude and longitude data to 'lats' and 'lons' variables
lats <- as.numeric(xy_mammals$lats)
lons <- as.numeric(xy_mammals$lons)
# 3) Calculate geographic distance between plots and assign to the variable 'coord.dist'
```

```r
coord.dist_m <- dist(as.matrix(lats, lons))
# 6) Transform all distance matrices into database format using the `liste()` function in `simba`:
comm.dist.ls_m <- liste(comm.dist_m, entry = "comm")
coord.dist.ls_m <- liste(coord.dist_m, entry = "dist")
# 7) Create a data frame containing similarity of community.
df <- data.frame(coord.dist.ls_m, comm.dist.ls_m[,3])
# 8) Attach the columns labels 'struc' to the dataframe you just made.
names(df)[4] <- c("struc")
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(dist, struc, xlab = "Geographic Distance", ylab = "1 - Bray-Curtis", main = "Community\nCompositio
OLS <- lm(struc ~ dist)
OLS
abline(OLS, col = "red4")


#For the North America
America <- c("USA", "Canada", "Mexico")
SbyS_mammals_TNA <- data.frame()
sites <- read.table("D:/Jane/IU/QB/Mammals/MCDB_sites.csv", quote = "", sep = ",", header = TRUE, fill =
sites_TNA <- subset.data.frame(sites, sites$Country == America)
SbyS_mammals_TNA <- subset(SbyS_mammals, row.names(SbyS_mammals) %in% sites_TNA$Site_ID)

env_mammals_TNA <- subset(env_mammals, row.names(SbyS_mammals) %in% sites_TNA$Site_ID)
xy_mammals_TNA <- data.frame(siteID = env_mammals_TNA$Site_ID, lats = env_mammals_TNA$Latitude, lons = e

# 1) Calculate Bray-Curtis similarity between plots using the `vegdist()` function
comm.dist_m_TNA <- 1 - vegdist(SbyS_mammals_TNA)
# 2) Assign UTM lattitude and longitude data to 'lats' and 'lons' variables
lats <- as.numeric(xy_mammals_TNA$lats)
lons <- as.numeric(xy_mammals_TNA$lons)
# 3) Calculate geographic distance between plots and assign to the variable 'coord.dist'
coord.dist_m_TNA <- dist(as.matrix(lats, lons))
# 6) Transform all distance matrices into database format using the `liste()` function in `simba`:
comm.dist.ls_m_TNA <- liste(comm.dist_m_TNA, entry = "comm")
coord.dist.ls_m_TNA <- liste(coord.dist_m_TNA, entry = "dist")
# 7) Create a data frame containing similarity of community.
df_TNA <- data.frame(coord.dist.ls_m_TNA, comm.dist.ls_m_TNA[,3])
# 8) Attach the columns labels 'struc' to the dataframe you just made.
names(df_TNA)[4] <- c("struc")
attach(df_TNA)
# 9) After setting the plot parameters, plot the distance-decay relationships, with regression lines in
par(mfrow = c(1,2), pty = "s")
```




```
plot(dist, struc, xlab = "Geographic Distance(North America)", ylab = "1 - Bray-Curtis", main = "Commun
OLS <- lm(struc ~ dist)
OLS
abline(OLS, col = "red4")
```



The geographic distance-decay graph represent expected trend of decreasing of comnunity similarity with geographic distance. For North America subset of data the distance decay reationship has the same slop (-3.76e-5) but slightly different interception coefficient. It can mean that it this slop coefficient is general for mammallian communitie. The fact that the trend is not so obvious could be caused by enormous area of sampling. This hypothesis has to be evaluated on the rest continents.)