Geographical Ecology

Z620: Quantitative Biodiversity, Indiana University

February 13, 2015

OVERVIEW

In this exercise, we will add a geographical context to alpha (α) and beta (β) diversity. We will introduce Geographical Information Systems (GIS) to map and spatially examine environmental and biodiversity data. This will allow us to explore core concepts like spatial autocorrelation, aggregation, and scale dependence.

After completing this exercise you will be able to:
1. Identify primary concepts and patterns of geographical ecology
2. Examine effects of geographic distance on environmental and ecological similarity.
3. Characterize aggregation of abundance across space.
4. Examine the extent to which patterns of diversity depend on spatial scale.
5. Use geospatial data and packages to conduct GIS operations in R.
6. Use control structures such as `loops` to control how R operates on variables.

1.) SETUP

A. Retrieve and Set Your Working Directory

```
rm(list=ls())
getwd()
setwd("~/GitHub/QuantitativeBiodiversity/Assignments/GeographicalEcology")
```

B. Load Packages

We will use the `vegan` package for biodiversity estimators and related functions.

```
require("vegan")
```

We will also use a suite of six packages developed in R for geographical information systems (GIS). Be sure to run **install.packages(PackageName)**, if not previously installed, where PackageName is the name of the package you want installed. Or, run **install.packages(PackageName, type="source", dependencies=TRUE)**, if the previous command doesn't work.

```
require("sp")          # Classes and methods for handling spatial data
require("geoR")        # Methods for geostatistical analyses
require("rgdal")       # Geospatial Data Abstraction Library
require("raster")      # Methods to create a RasterLayer object
require("RgoogleMaps") # For querying the Google server for static maps.
require("maptools")    # Tools for manipulating and reading geospatial data
```

C. Load and Compile a Large Dataset

We will analyze environmental and bacterial community data from a survey of shallow ponds found east of Bloomington, IN. These ponds are scattered throughout Brown County State Park, Yellowood State Forest, and Hoosier National Forest. In the 1940s, Maury Reeves of the Indiana Department of Natural Resources began constructing refuge ponds for wildlife. In the summer of 2013, we visited approximately 50 of these ponds and recorded their geographic locations using a GPS unit; 'GPS' is the acronym for Global Positioning System. We sampled aspects of water chemistry, physical properties, and bacterial community composition. Let's load the environmental and site-by-species data for the refuge ponds.

```r
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=",")
```

Take a look at the `Environment` tab in the upper right console of RStudio. You should see there are 16,384 operational taxonomic units (OTUs) distributed across 51 sites, for which, we have 19 environmental and geographic variables recorded. These variables include elevation (m), geographical coordinates (lat-long), temperature (C), Diameter(m), Depth(m), redox potential (ORP), specific conductivity or SpC (uS/cm), dissolved Oxygen (mg/L), total dissolved solids (g/L), salinity (p.s.u.=ppm), color - measured at absorbance = 660; an estimate of carbon in the water sample, chlorophyll a (ug/ml), dissolved organic carbon (mg/L), dissolved organic nitrogen (mg/L), and total phosphorus (ug/L).

In addition to this wealth of environmental data, let's add four diversity-related columns of data to `Ponds` data set. These will provide basic diversity-related variables to explore with respect to geograpy and environmental conditions. There will be a column for richness (S), total abundance (N), Shannon's Diversity (H), and Simpson's evenness (De).

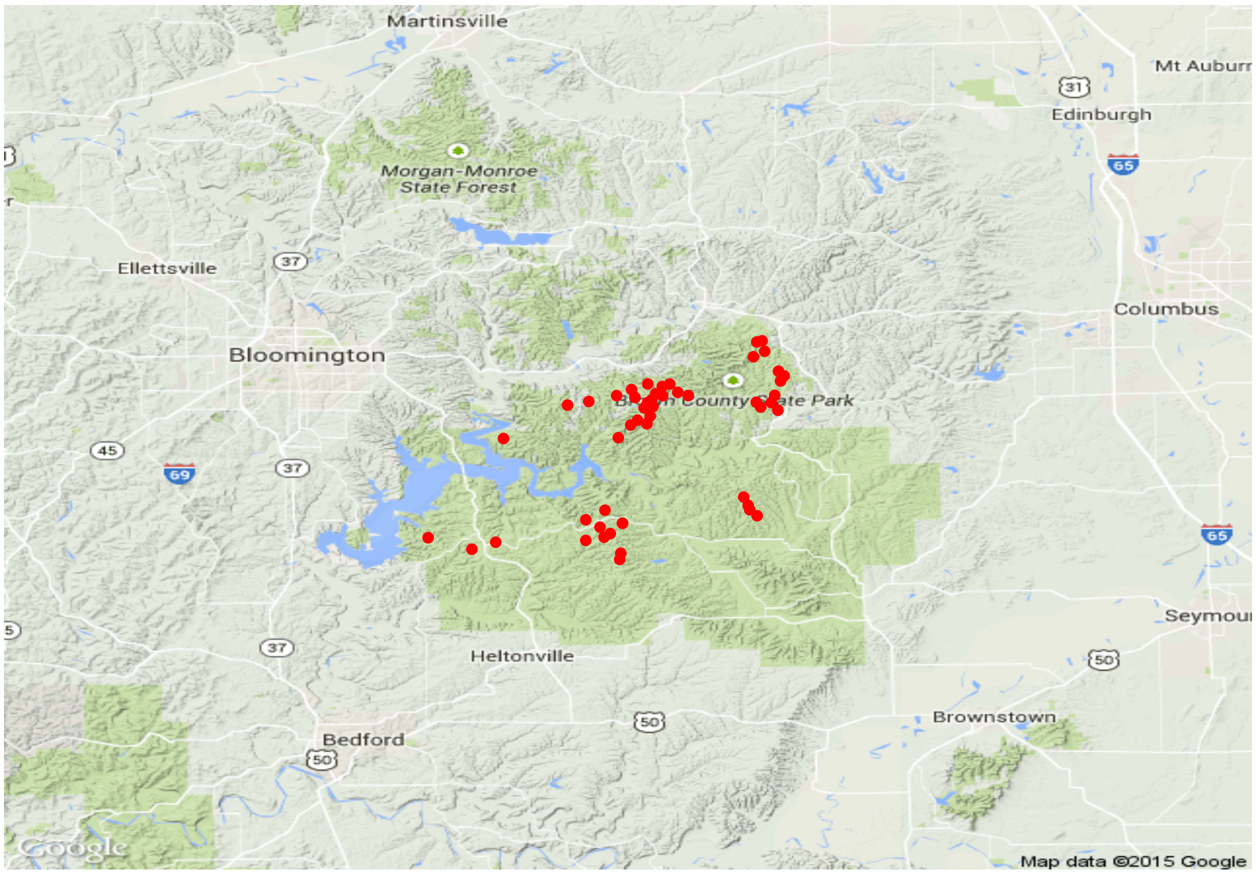
```r
otu.names <- names(OTUs) # Get the names of the OTUs
OTUs <- as.data.frame(OTUs[-1]) # remove first column (site names)

Ponds$N <- as.vector(rowSums(OTUs)) # numbers of reads
Ponds$S <- as.vector(rowSums(OTUs > 0) * 1)
# For each site: S equals the number of non-zero abundances

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

Now we have a large compilation of geographical, environmental, and biodiversity data. Let's do some Geographical Ecology!

2.) MAP SAMPLES AND DATA

Let's visualize the spatial distribution of our samples with a basic map in RStudio. Let's generate a map of the refuge ponds using the `GetMap` function in the package `RgoogleMaps`. This map will be centered on Brown County, Indiana (39.1 latitude, -86.3 longitude).

```r
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)
```

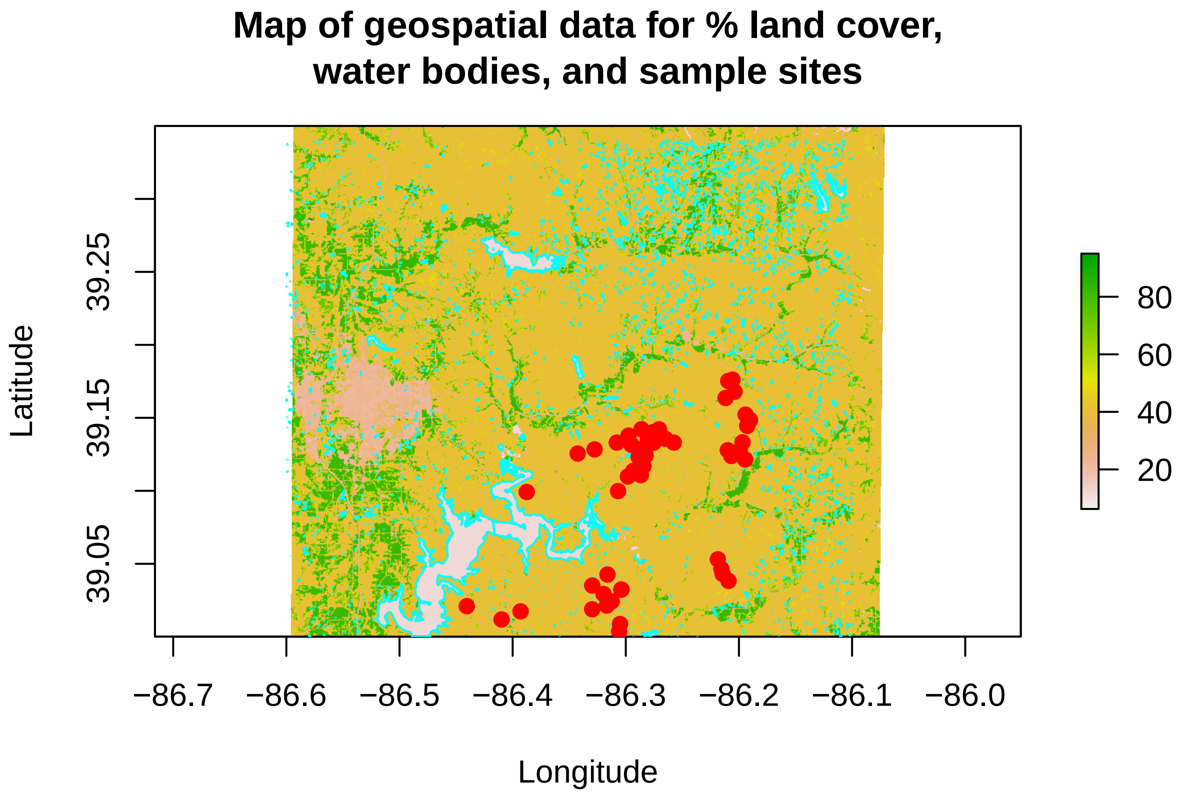


This map displays a lot of useful information that we otherwise, would not have been aware of. For example, all points are on State or National Forest land. Likewise, the sample ponds appear to be aggregated in four or five small groups and distributed across a topographically complex area.

Despite being a fast way to contextualize our sample ponds within the broader landscape, the Google map misses a lot of information that would otherwise help us to understand the environmental and geographical factors that may coincide with our observations on diversity. Likewise, because the Google map is only an image, it doesn't contain any extractable environmental or geographic data.

For spatially explicit data on environmental and geographic features, i.e. geospatial data, we can turn to one of the many freely accessible online GIS databases and warehouses. Here, we will use the high quality geospatial data on Indiana water bodies and percent landcover. We obtained these data 'layers' from the **IndianaMap** geographical layer gallery: http://maps.indiana.edu/layerGallery.html.

```
Land.Cover <- raster("LandCover/LandCover.tif")
plot(Land.Cover, xlab='Longitude', ylab='Latitude',
  main='Map of geospatial data for % land cover,\nwater bodies, and sample sites')

Water.Bodies <- readShapeSpatial("water/water.shp")
plot(Water.Bodies, border='cyan', axes=TRUE, add = TRUE)

Refuge.Ponds <- SpatialPoints(cbind(lons, lats))
plot(Refuge.Ponds, line='r', col='red', pch = 20, cex=1.5, add=TRUE)
```



Note, that the percent land cover, water bodies, and points for refuge ponds are in spatial agreement, i.e., there is no obvious mis-alignment. That is because we have previously modified each layer to have the same *datum*, *projection*, and nearly the same *extent*. **See the glossary .Rmd file for basic definitions of these and other GIS terms.**

Working with geospatial data can be challenging because there is so much information involved with correctly identifying where on Earth something occurs and because there are many ways to represent points on a globe with 2-dimensional surface, i.e., a map. But, whether it's data on temperature, elevation, soils, geology, human demographics, ecoregions, etc., diverse data can be found among the many GIS warehouses. Here are a few: 1.) USGS: http://viewer.nationalmap.gov/viewer/ 2.) State organizations: http://www.indianamap.org/resources.php 3.) USDA: http://datagateway.nrcs.usda.gov/

3. PRIMARY CONCEPTS AND PATTERNS

Having imported our primary community and environmental data from the refuge ponds, as well as having obtained a wealth of geospatial data from online sources, we are now ready to make a data intensive exploration into primary concepts and patterns of geographical ecology.

A. Spatial Autocorrelation

Tobler's first law of geography states that "Everything is related to everything else, but near things are more related than distant things" (Tobler 1970). This law is a formulation of the concept of spatial autocorrelation. In short, spatial autocorrelation is the degree to which spatial variables are either clustered in space (positive autocorrelation) or over-dispersed (negative autocorrelation).

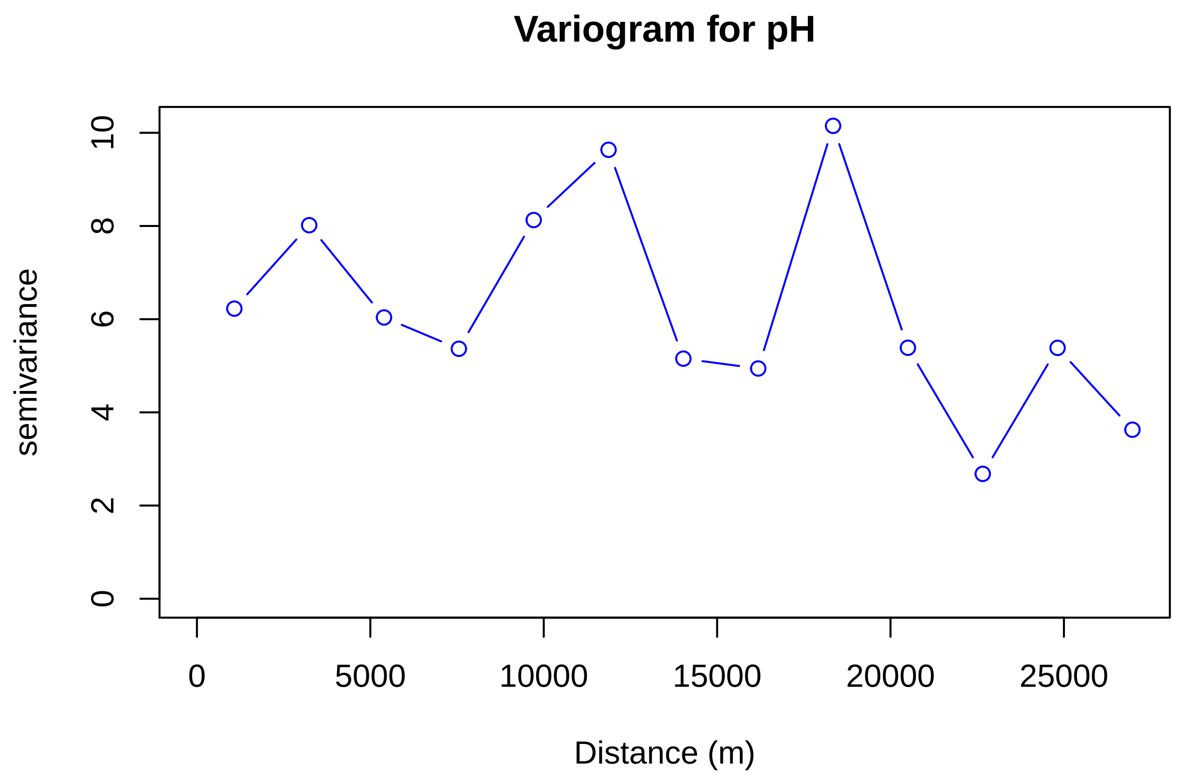
When examing spatial data, it is important to check for autocorrelation not just among variables but across distance. We want to know more than whether variables are generally auto-correlated, but how greatly that autocorrelation changes (increases or decreases) with distance. Variables that are highly autocorrelated at the meter scale may or may not be less autocorrelated at the kilometer scale. If we want to use these variables in our analyses, we should not use the scales over which they are autocorrelated.

Here, we reveal a way of detecting autocorrelation with respect to scale, that is, by using a **variogram**. Variograms are frequently used in spatial analyses and reveal the degree of spatial autocorrelation in sample data and how the autocorrelation (measured as the **semivariance**) changes over distance.

The semivariance is a measure of the dispersion of all observations that fall below the mean. In fact, if you were to find the differences between all possible points spaced a constant distance apart and then find the variance among the differences, and then divide that variance in half, you would have the semivariance. While the semivariance is similar to the variance, it only considers observations below the mean. In this case, higher semivariance means implies greater dispersion (lower autocorrelation) of values at a given spatial scale.

Let's plot the variogram for one of our environmental variables.

```r
xy <- data.frame(pond.name = Ponds$Sample_ID, lats = Ponds$lat, lons = Ponds$long)
coordinates(xy) <- c("lats", "lons")

proj4string(xy) <- CRS("+proj=longlat +datum=NAD83")
# Identifying the current projection and datum
UTM <- spTransform(xy, CRS("+proj=utm +zone=51 ellps=WGS84"))
# Transforming the projection and data, so we can get meaningful distances

UTM<-as.data.frame(UTM)
coords<-UTM[,1:2]

var.trend.geoR <- variog(coords=coords, data=Ponds$DOC, option="bin") # try TDS and pH!!
plot(var.trend.geoR, type = "b", main = "Variogram for pH", xlab='Distance (m)',
     col='blue')
```



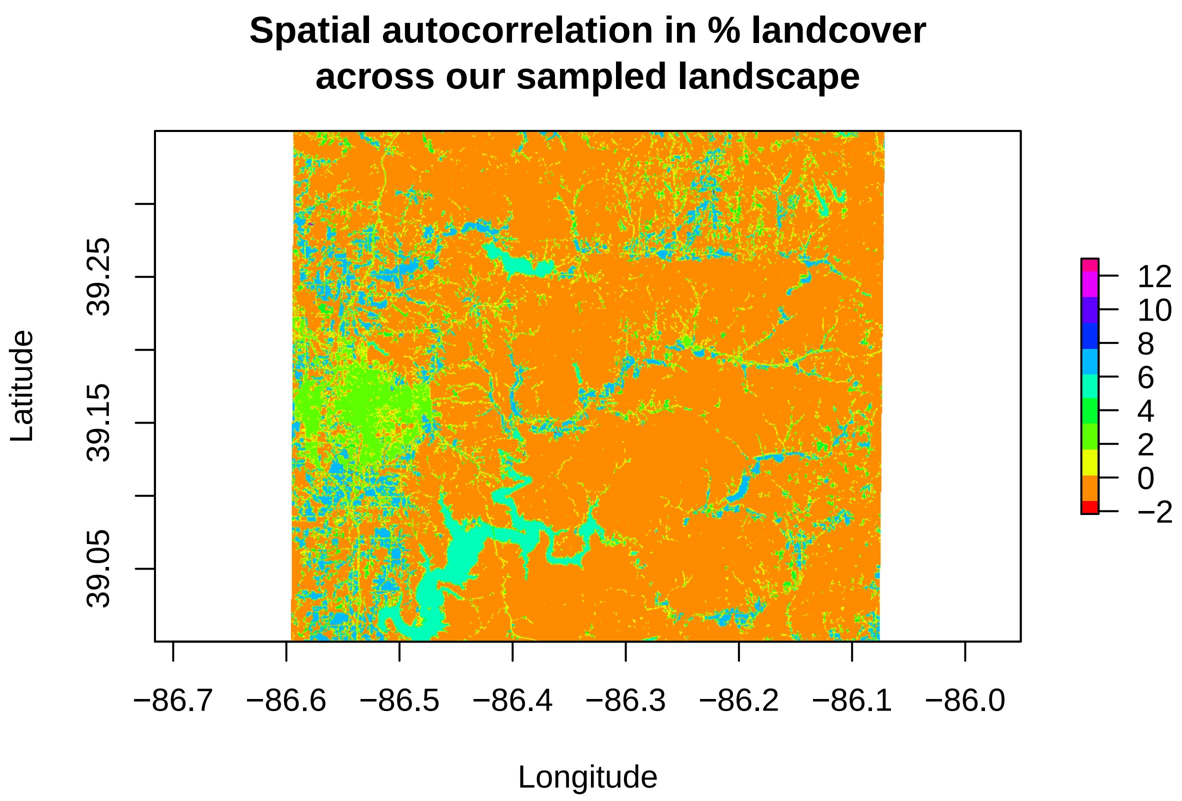
Question 1: How does the **semivariance** change with distance? Is there a distance (i.e. "spatial lag") where semivariance is very high or low?

> ***Answer 1***:

For a more visually informative picture, we can visualize autocorrelation across the landscape, by calculating **Moran's I**, a correlational statistic that measures autocorrelation based on feature locations and feature values. Moran's I evaluates whether the pattern expressed is clustered, dispersed, or random. The function we will use also assigns a global Moran index value, a z-score and p-value.

Take for example, our land cover data represented in a raster file. Raster files are one of two types of digital graphic files, the other being vector files. Images in vector files are made of many lines and curves (or paths) that create the image. In contrast, raster images are composed of pixels and for our purposes, encode the values of the environment at a precise point. Using R's `raster` package, we can calculate **global** (across the landscape) and **local** (in comparison to neighbors) measures of **Moran's I**.

```r
Moran(Land.Cover)
LC.Moran <- MoranLocal(Land.Cover)
plot(LC.Moran, xlab="Longitude", ylab="Latitude",
     main="Spatial autocorrelation in % landcover\nacross our sampled landscape",
     col=rainbow(11, alpha=1))
```



Question 2: Moran's global I can range from 0 to 1. What was Moran's global I?

> ***Answer 2***:

Question : Looking at the 'landscape' of Moran's *local* I for percent landcover, make some observation about spatial autocorrelation. For example, where is it high or low?

Pattern 1: Distance-decay relationship

The distance-decay relationship is the primary pattern of spatial autocorrelation, and captures the rate of decreasing similarity with increasing distance. This pattern addresses whether communities close to one another are more similar than communities that are farther away. The distance-decay pattern can also be used to address whether near environments have greater similarity than far ones.

Let's load the `simba` package and generate distance decay relationships for bacterial communities of our refuge ponds and for some of the environmental variables we measured. Note, this analysis will span the next few code blocks.

```
require("simba")

struc.dist <- 1 - vegdist(OTUs) # Bray-Curtis similarity between the plots
coord.dist <- dist(as.matrix(lats, lons)) # geographical distance between plots

# transform environmental data to numeric types
temp <- as.numeric(Ponds$"Salinity")
```

```r
elev <- as.numeric(Ponds$"ORP")
depth <- as.numeric(Ponds$"Depth")
doc <- as.numeric(Ponds$"DOC")

# calculate the distance (Euclidean) between the plots regarding environmental variables
env.dist <- 1 - vegdist(cbind(temp, elev, depth, doc), "euclidean")

# transform all distance matrices into list format:
struc.dist.ls <- liste(struc.dist, entry="struc")
env.dist.ls <- liste(env.dist, entry="env")
coord.dist.ls <- liste(coord.dist, entry="dist")
```

Now, create a data frame containing similarity of the environment and similarity of community.

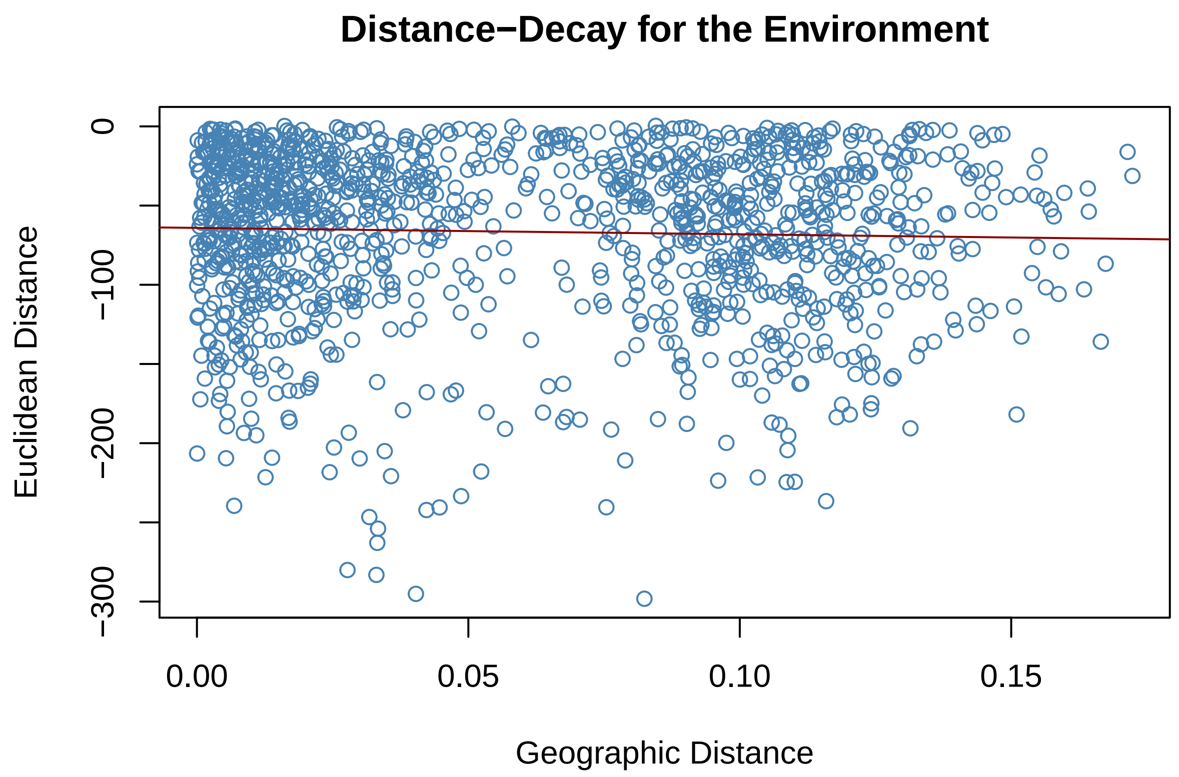
```r
df <- data.frame(coord.dist.ls, env.dist.ls[,3], struc.dist.ls[,3])
names(df)[4:5] <- c("env", "struc")
attach(df) #df <- subset(df, struc != 0)
```

Finally, let's plot the Distance-decay relationships, with regression lines in red.

```r
par(mfrow=c(1, 1))
plot(dist, env, xlab="Geographic Distance", ylab="Euclidean Distance",
     main = "Distance-Decay for the Environment", col='SteelBlue')

OLS <- lm(env ~ dist)
OLS # print regression results to the screen
```

```
##
## Call:
## lm(formula = env ~ dist)
##
## Coefficients:
## (Intercept)         dist
##      -64.11       -40.30
```

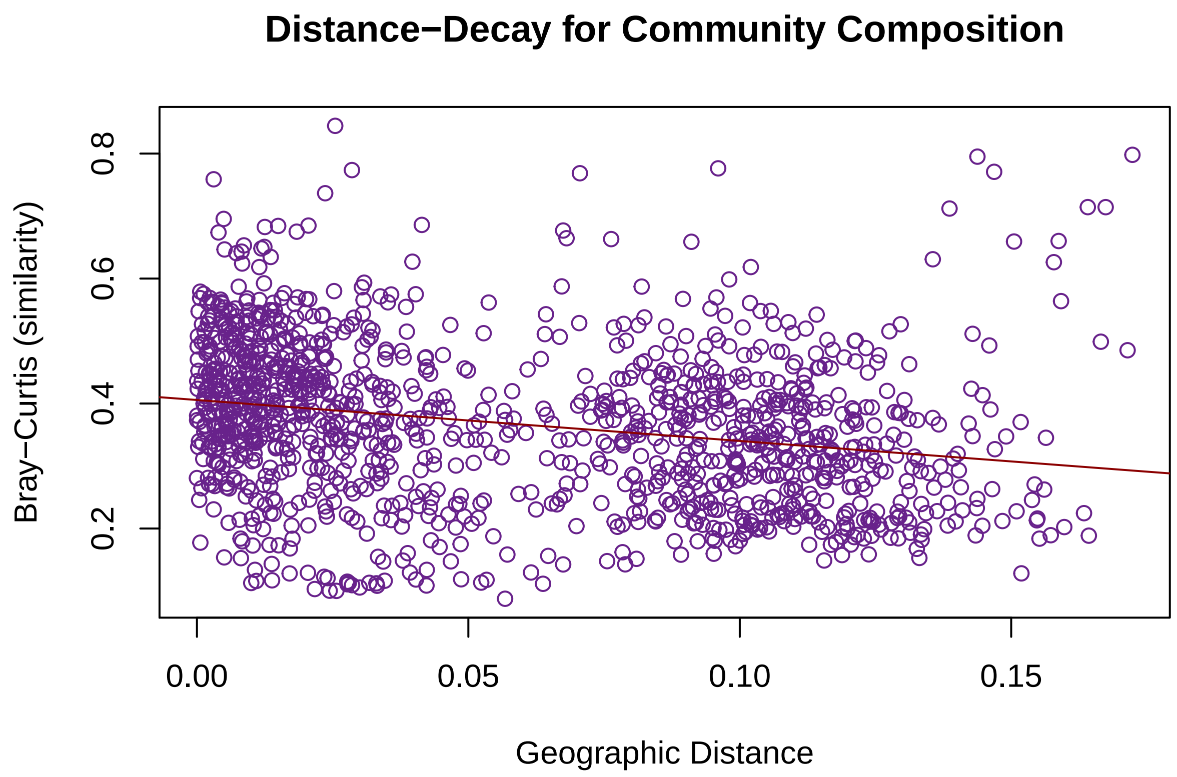
```r
abline(OLS, col="red4")
```



```r
par(mfrow=c(1, 1))
plot(dist, struc, xlab="Geographic Distance", ylab="Bray-Curtis (similarity)",
     main="Distance-Decay for Community Composition", col='darkorchid4')

OLS <- lm(struc ~ dist)
OLS # print regression results to the screen
```

```
##
## Call:
## lm(formula = struc ~ dist)
##
## Coefficients:
## (Intercept)         dist
##      0.4056      -0.6552
```

```r
abline(OLS, col="red4")
```



Let's, examine the slope of the regression lines, asking whether they are significantly different from one another.

```
diffslope(dist, env, dist, struc)
```

```
##
## Is difference in slope significant?
## Significance is based on 1000 permutations
##
## Call:
## diffslope(x1 = dist, y1 = env, x2 = dist, y2 = struc)
##
## Difference in Slope: -39.64
## Significance: 0.094
##
## Empirical upper confidence limits of r:
##   90%   95% 97.5%   99%
##  38.5  48.4  57.1  65.7
```

Question 3: Are microbial communities that are closer in geographic distance also closer in compositional similarity? How about for environmental condition?

> ***Answer 3***:

Concept 2: Spatial Aggregation

Tobler made a general observation that occurs in nearly all systems, i.e., spatial autocorrelation. A related observation is that natural phenomena are generally clustered, i.e., spatially aggregated. That is, individuals, conditions, and events often occur in patches, clusters, pulses, etc. Take for example, the ponds in our sample area. A highly level of aggregation would suggest that if we encounter one individual, then we are likely to encounter others of the same species nearby.

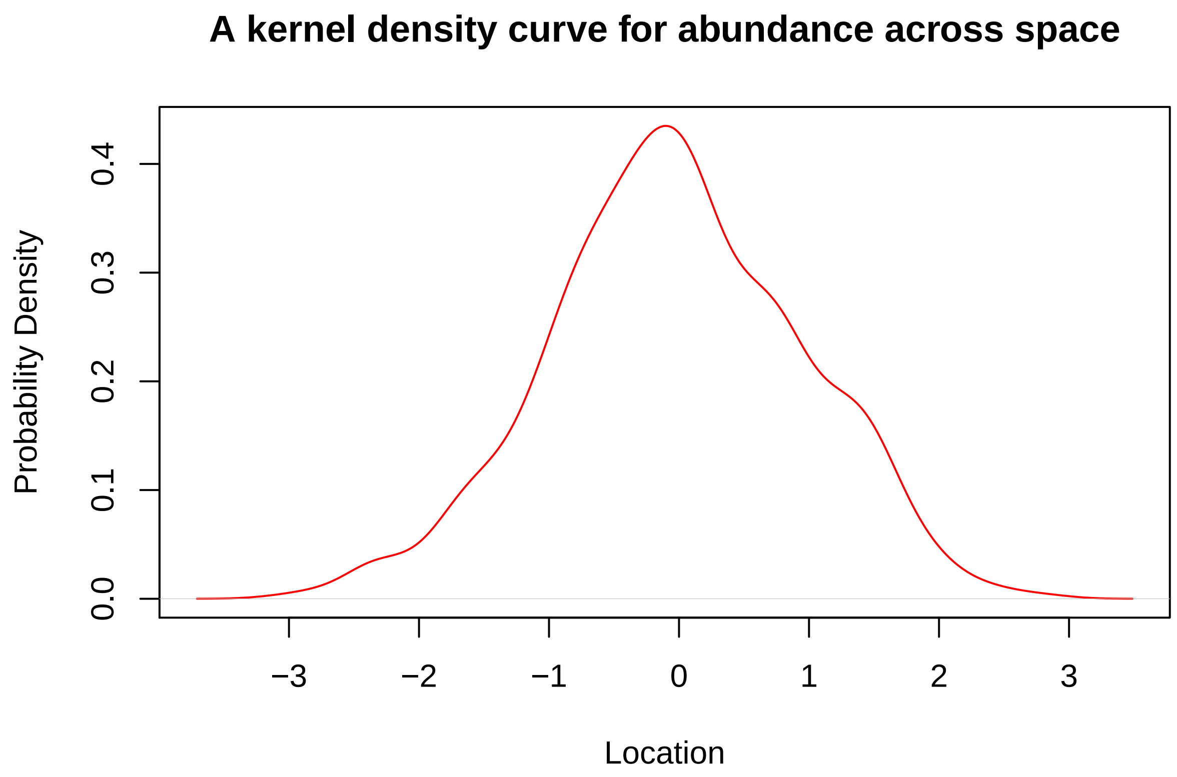
Pattern 2: Spatial abundance distribution

One of the primary patterns of spatial aggregation in ecology is the distribution of a species abundance within a landscape, also referred to as the **species spatial abundance distribution (SSAD)**. The SSAD reveals the frequency at which we find a species at a particular abundance. In this way, the SSAD is similar to a histogram.

Here, we will examine SSADs for OTU's in the refuge pond dataset by constructing **kernel-density curves**. Kernel density curves are analogous to histograms, but avoid the arbitrary creation of bins or discrete classes. In constructing kernel density curves, we attempt to account for uncertainty and sampling error by focusing on the probability that a randomly drawn data point will take a value within a particular range, instead of the exact frequencies we observed.

For example, suppose we were interested in how the location of individuals varied across sites or samples. Let's simulate this by drawing values from a normal distribution, at random.

```r
S.list <- rnorm(1000) # 1000 randomly drawn values from a normal distribution
plot(density(S.list), col = 'red', xlab='Location',
     ylab='Probability Density',  main = 'A kernel density curve for abundance across space')
```



Below, we will examine the SSADs of OTUs that are randomly drawn from the refuge ponds dataset. But first, let's begin by defining a function that will generate the SSAD for a randomly drawn OTU.

```
ssad <- function(x){
  ad <- c(2, 2)
  ad <- OTUs[, otu]
  ad = as.vector(t(x = ad))
  ad = ad[ad > 0]
}
```

Next, we will draw 4 OTUs at random and plot their SSADs. But first, we will need to introduce **while loops**.

While loops

If you have ever heard anything to the effect of, "While you're at it, do this. . . ", then you are familiar with the concept of a while loop. The while loop is a type of **control flow** structure that allows us to tell a program to perform an operation *while* some condition is satisfied.

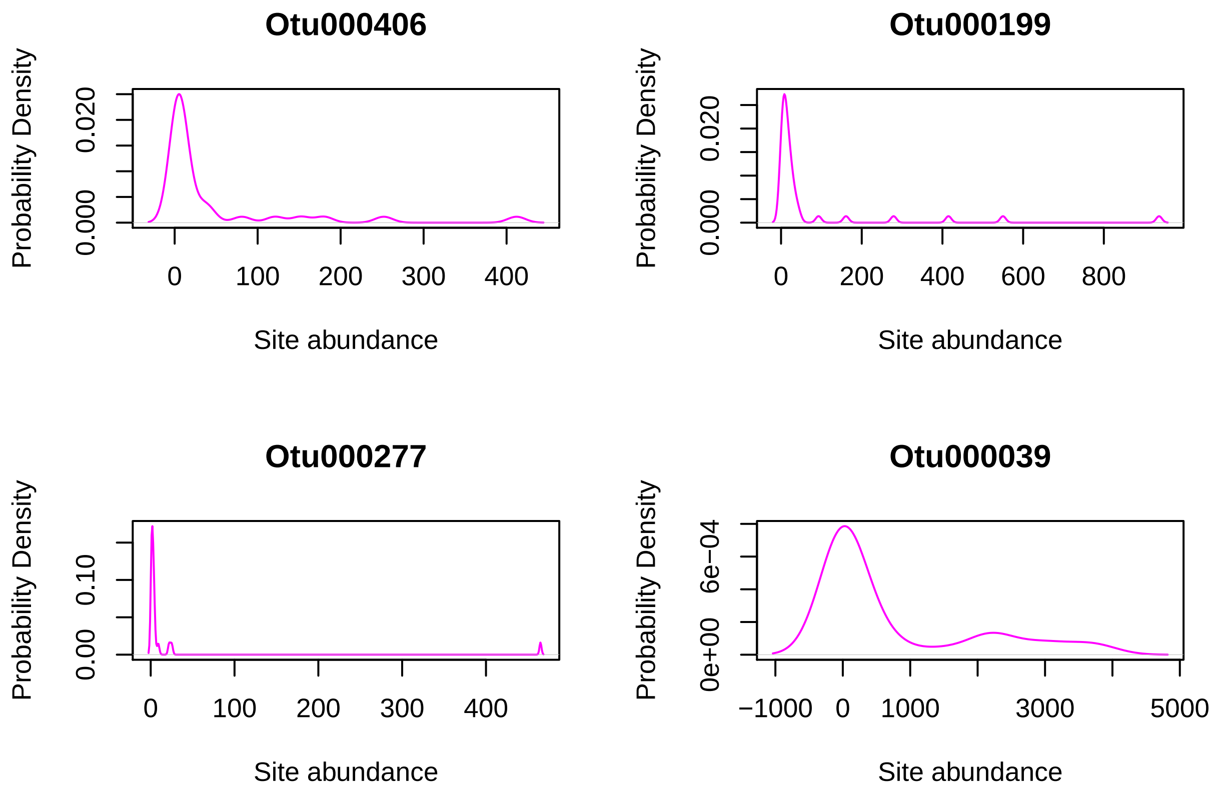
For example, we might want R to draw numbers at random until 4 numbers less than 50 have been drawn.

```
numbers = c()
while (length(numbers) < 4){ # while the counter is less than 4
  x <- runif(1, 1, 100) # draw a number at random from 1 to 100
  if (x < 50){     # if the number is less than 50...
    numbers <- c(numbers, x)
  }
}
numbers # check our numbers, each should be less than 50
```

```
## [1] 14.406948 32.912888  2.449078  2.372687
```

Having very briefly intodroduced while loops, and inaverdently, an `if` statement, lets write a chunk of code that will draw OTUs at random from our refuge ponds dataset, and then generate their spatial abundance distributions (i.e. SSADs).

```
par(mfrow=c(2, 2))

ct <- 0          # a counter variable
while (ct < 4){ # the while statement
  otu <- sample(1:length(OTUs), 1) # choose a random OTU
  ad <- ssad(otu)                  # find the OTU's SSAD
  if (length(ad) > 10 & sum(ad > 100)){
    ct <- ct + 1
    plot(density(ad), col = 'magenta', xlab='Site abundance',
    ylab='Probability Density',  main = otu.names[otu])
    }
  }
```



Feel free to run this chunk as many time as you like.

Question 4: Is the sampled abundance for a given OTU often aggregated? If so, how do you know, that is, how do you interpret the pattern in the kernel density curve? Are there many sites with low abundance and few sites with high abundance?

> ***Answer 4***:

Question : The SSAD is a statistical distribution and like all statistical distributions, the SSAD has a mean, median, mode, variance, skewness, kurtosis, etc. What aspects of the SSAD would you choose to quantify and report, and why?

Question 5: Each row in the site-by-species matrix represents a site. Each column represents an OTU. If the SSAD is generated by considering all rows for a single column (i.e. OTU), then what do we obtain when we consider all columns for a given row (i.e. site)? Have we examined this sort of data structure before?

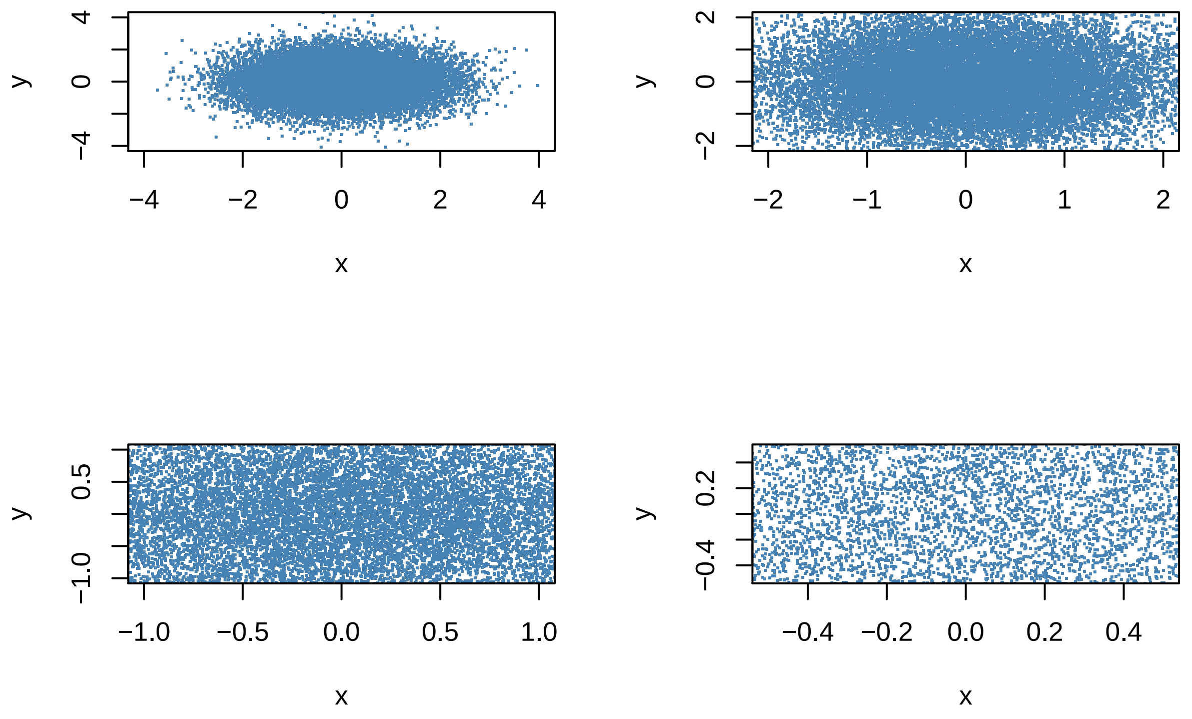
> ***Answer 5***:

Concept 3: Scale-Dependence

Our idea of whether variables are spatially autocorrelated and whether the abundances of OTUs are spatially aggregated can change with aspects of spatial scale, i.e. extent and grain. **Extent** is the greatest distance considered in an observation or study. **Grain** is the smallest or primary unit by which the extent is measured.

Let's generate two random samples from a normal distribution, one sample for x-coordinates and one for y-coordinates. We'll let each x-y pair represent the location of a single individual, where all individuals belong to the same species. Then, we'll plot the spatial distribution of our randomly distributed individuals at different extents.

```r
par(mfrow=c(2, 2))

x <- rnorm(20000)
y <- rnorm(20000)

plot(x,y, xlim=c(-4, 4), ylim=c(-4, 4), pch=".", col='Steelblue')
plot(x,y, xlim=c(-2, 2), ylim=c(-2, 2), pch=".", col='Steelblue')
plot(x,y, xlim=c(-1, 1), ylim=c(-1, 1), pch=".", col='Steelblue')
plot(x,y, xlim=c(-0.5, 0.5), ylim=c(-0.5, 0.5), pch=".", col='Steelblue')
```



Question 6: What effect does changing the extent have on aggregation? Do you find this important or interesting given that 1.) all points were drawn from the same distribution and 2.) each plot contains the same points as all other plots with smaller extent?

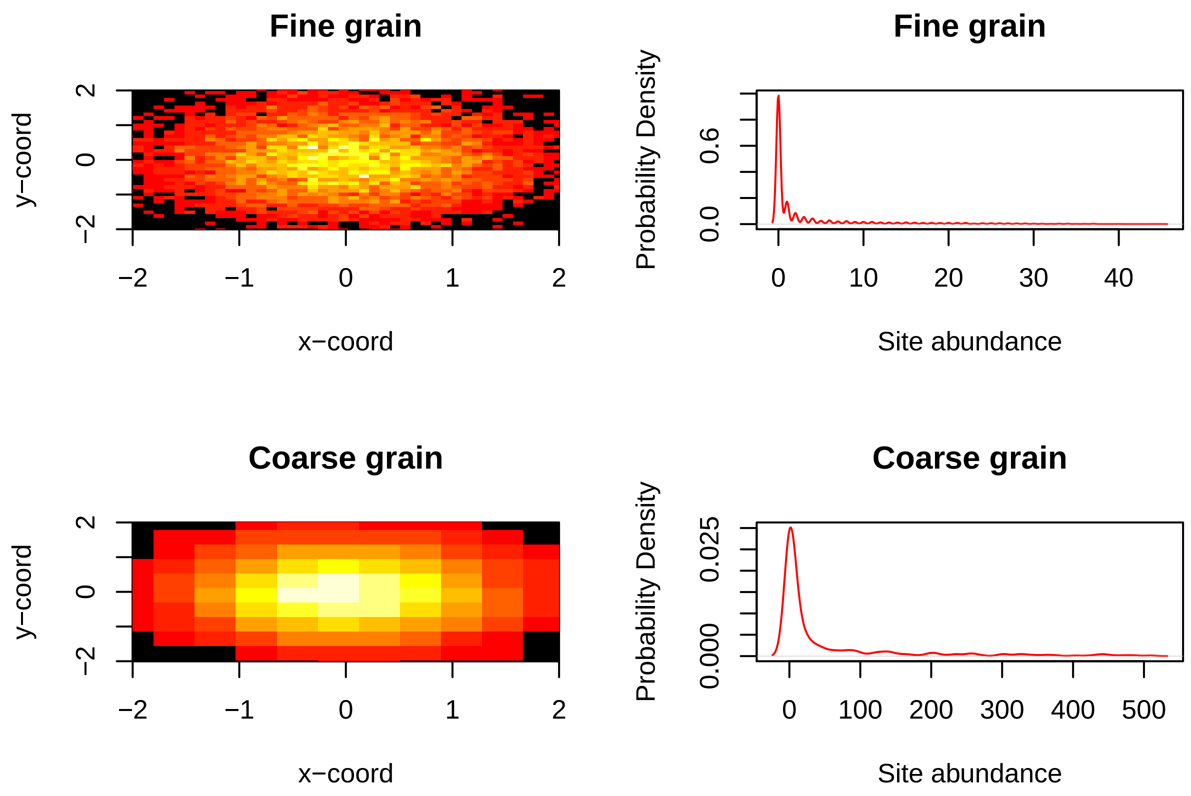
Answer 6:

It should be clear from above, that 'random' does not mean absent of aggregation. In fact, most statistical distributions from which random samples can be drawn are very aggregated. That is, they have obvious modes around which most values occur.

Moving on, let's explore the effect of changing spatial `grain`, from a fine grain to a coarse grain. We will do this while holding extent constant and will plot heat maps (i.e. 2D histogram) revealing the density of individuals in the landscape. We will then plot kernel density curves to reveal the probability that an individual chosen at random from the landscape will have come from a site with a particular abundance.

```r
require("gplots")
par(mfrow=c(2, 2))

df <- data.frame(x,y)

h1 <- hist2d(df, nbins=80, show=TRUE, xlim=c(-2,2), ylim=c(-2,2),
             xlab='x-coord', ylab='y-coord', main = "Fine grain" )
ad <- h1$counts
plot(density(ad), col = 'red', xlab='Site abundance',
     ylab='Probability Density',  main = "Fine grain")
```

```
h4 <- hist2d(df, nbins=20, show=TRUE, xlim=c(-2,2), ylim=c(-2,2),
             xlab='x-coord', ylab='y-coord', main = "Coarse grain" )
ad <- h4$counts
plot(density(ad), col = 'red', xlab='Site abundance',
     ylab='Probability Density',  main = "Coarse grain")
```



Question 7: Beyond changing the pixilated appearance of the plots, what does changing the spatial grain mean for interpreting aggregation? Consider the kernel density plots.

> ***Answer 7***:

Question 8: How are the kernel density curves we just generated for our randomly drawn points related to the species spatial abundance distributions (SSAD) that we generated for OTUs in our refuge plots?

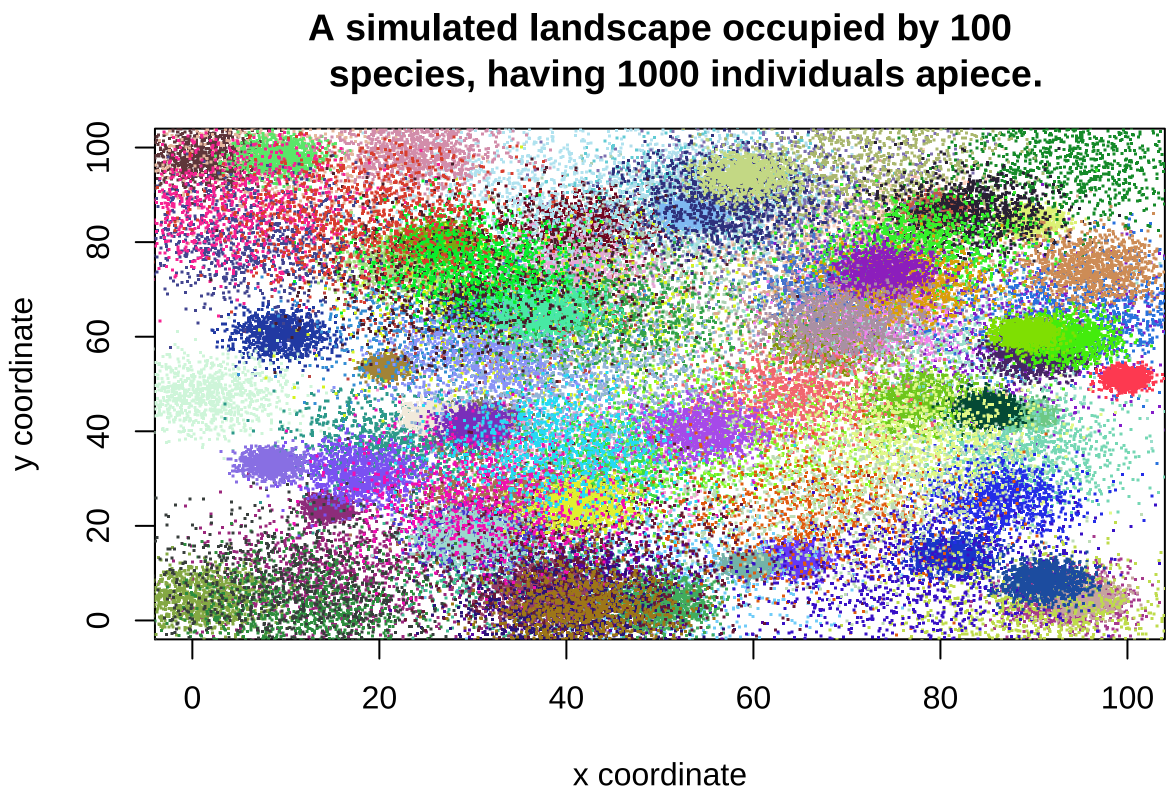
> ***Answer 8***:

Primary Concept 3: Spatial Accumulation

So far, we have discussed spatial autocorrelation and aggregation as core concepts of geographical ecology. Likewise, we have introduced and examined primary patterns for both of those concepts. Here, we introduce another core concept, accumulation across space. It may seem self-evident that, if starting from the scale of a single individual and increasing our sample area, that we will inevitably encounter more species, OTUs, or other taxa.

For example, suppose we replicate our above random sampling strategy of drawing x-y coordinates from a normal distribution. But, instead of drawing just one sample representing one species, we will draw 50 samples, with each representing a species with 1000 individuals.

```
community <- c()
species <- c()

# initiate the plot
plot(0, 0, col='white', xlim = c(0, 100), ylim = c(0, 100),
     xlab='x coordinate', ylab='y coordinate',
     main='A simulated landscape occupied by 100
     species, having 1000 individuals apiece.')

while (length(community) < 100){ # while our community has less than 100 species
  # choosing the mean, standard deviation, and colors at random
  std <- runif(1, 1, 10)
  x <- rnorm(1000, mean = runif(1, 0, 100), sd = std)
  y <- rnorm(1000, 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
  }
```

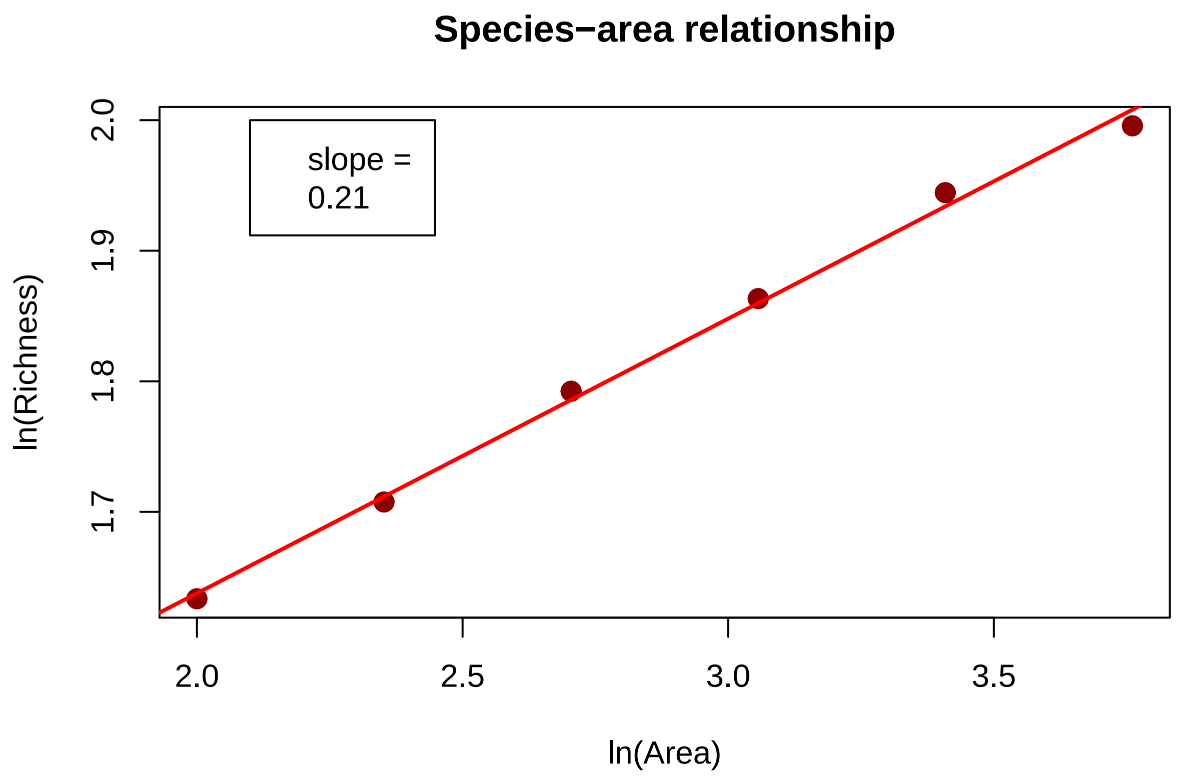


Having generated a simulated landscape occupied by 50 species having 1000 individuals apiece, we can examine how richness can accumulate with area. Let's begin by picking a corner at random and then accumulating area.

```
lim <- 10

S.list <- c()
A.list <- c()

while (lim <= 100){ # while the spatial extent on the x and y is less than or equal to 100
  S <- 0            # Set richness to be zero
  for (sp in community){ # for each species in the community
    xs <- sp[[1]] # assign the x coordinates
    ys <- sp[[2]] # assing the y coordinates
    sp.name <- sp[[3]]  # assign the species name
    xy.coords <- cbind(xs, ys)
    for (xy in xy.coords){ # for each pair of xy coordinates in xy.coords
      if (max(xy) <= lim){ # if the individual is within our current spatial extent...
        S <- S + 1         # then the species occurs there
          break  # break out of the last for loop because we're only considering
                 # incidence, and not abundance.
                 # In other words, if the species occurs once, that's good enough
        }
      }
    }
  S.list <- c(S.list, log10(S))
  A.list <- c(A.list, log10(lim^2))
  lim <- lim * 1.5  # increase the extent multiplicately, but slowly
}
```

Having generated our primary vectors, S.list and A.list, we can analyze how species richness scales with area. In short, we can analyze one of ecology's oldest and most intensively studied patterns, the **Species-Area Relationship**.

```
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=2.1, y=2, c('slope = ', z))
```



Pattern 3: Species-area relationship (SAR)

The fact that we accumulate species, and likewise increase richness, with increasing area is far from interesting. In fact, we just showed that we can expect this as a result of random sampling. What is interesting is the rate at which the accumulation of taxa occurs. Arrhenius (1921) first described the general form of the *species-area relationship (SAR)* as a power-law: $S = cA^z$ where S is species richnness and A is area.

Power-laws reveal how one quantity scales with another, most often across orders of magnitude. Arrhenius's formula predicts a rate of increase in richness that is approximately linear in log-log space. That is, $log(S) = c + zlog(A)$, where z is the scaling exponent.

Question 9: The authors of your assigned reading revealed that the exponent of the SAR may be influenced by geographic, ecological, and evolutionary factors. But, what in general, is the value of z?

Answer 9:

Question 10: What was the slope of the species-area relationship for our randomly assembled community? Is this similar to the slopes you encountered in the reading?

Answer 10:

Question 11: We could use this 'random placement' approach to model how many ecological phenomena might occur via random sampling. What other spatial aspects of alpha and beta diversity could we address? Suggest at least 3.

Answer 11:

7) HOMEWORK

1.) Complete the in-class exercise, Knit to a pdf, and submit a pull request.

2.) Each refuge pond has an associated diameter. Build the species-area relationship for the refuge pond dataset using the following recipe:

1. Using the formula for the area of a circle, calculate the area for each pond.
2. Randomly choose one pond and obtain its area and richness.
3. Choose two ponds at random and obtain their combined richness and their summed area. Do not simply sum the richnesses of the two sites, as this will result in double-counting species.
4. Choose three, four, five, etc. ponds at random, repeating the above steps each time. You will eventually work you way up to 51 ponds. **You will need to use loops for this**.
5. At this point you should have two vectors, one for richness (S) and one for area (A).
6. Plot the SAR and estimate its slope.
7. REPEAT steps 2 through 6 one thousand times, adding each new SAR to the same plot. Once again, you will need to use loops, as above.
8. In a second plot, generate a kernel density curve for the slopes of the SARs.

3.) Draw several general conclusions from your analyses in question #2.

4.) Which environmental and diversity variables reveal positive spatial autocorrelation?

5.) A. How many OTUs are present at more than 10 sites? How many OTUs only occur at one site?

6.) In considering total abundances (N) among the refuge ponds, we are really only considering the number of detected 16S rRNA reads for any given OTU. Find the mode of the SSAD for each OTU that is present at 10 or more sites; it's difficult to generate an informative histogram for less than 10 sites. Then, generate a kernel density curve for these modes, revealing the pattern of modal abundance across these more common OTUs. Draw general conclusions about trends across the refuge pond OTUs from your results.