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 explore environmental, spatial, 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. Examine spatial aggregation of abundance.
4. Examine the extent to which patterns of diversity depend on spatial scale.
5. Find and use geospatial data 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.

Let's begin by loading the environmental data and site-by-species matrix for the refuge ponds.

```
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 21 environmental and geographic variables recorded. These variables include elevation (in meters) and geographical coordinates (lat-long), temperature (Celcius), dissolved oxygen (DO), ...

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

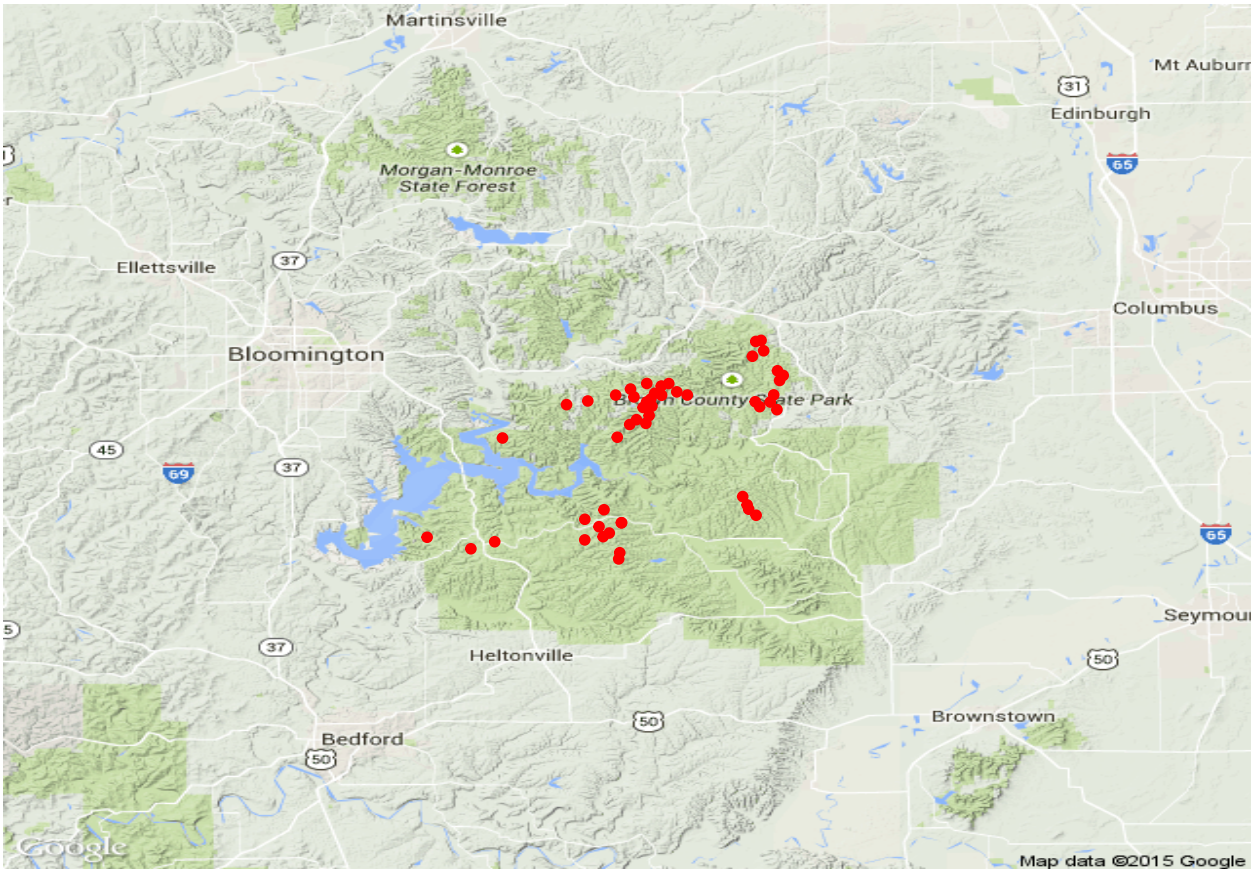
```
otu.names <- names(OTUs) # Get the names of the OTUs
OTUs <- as.data.frame(OTUs[-1]) # remove first column (site names)

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, 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).

```
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)
```

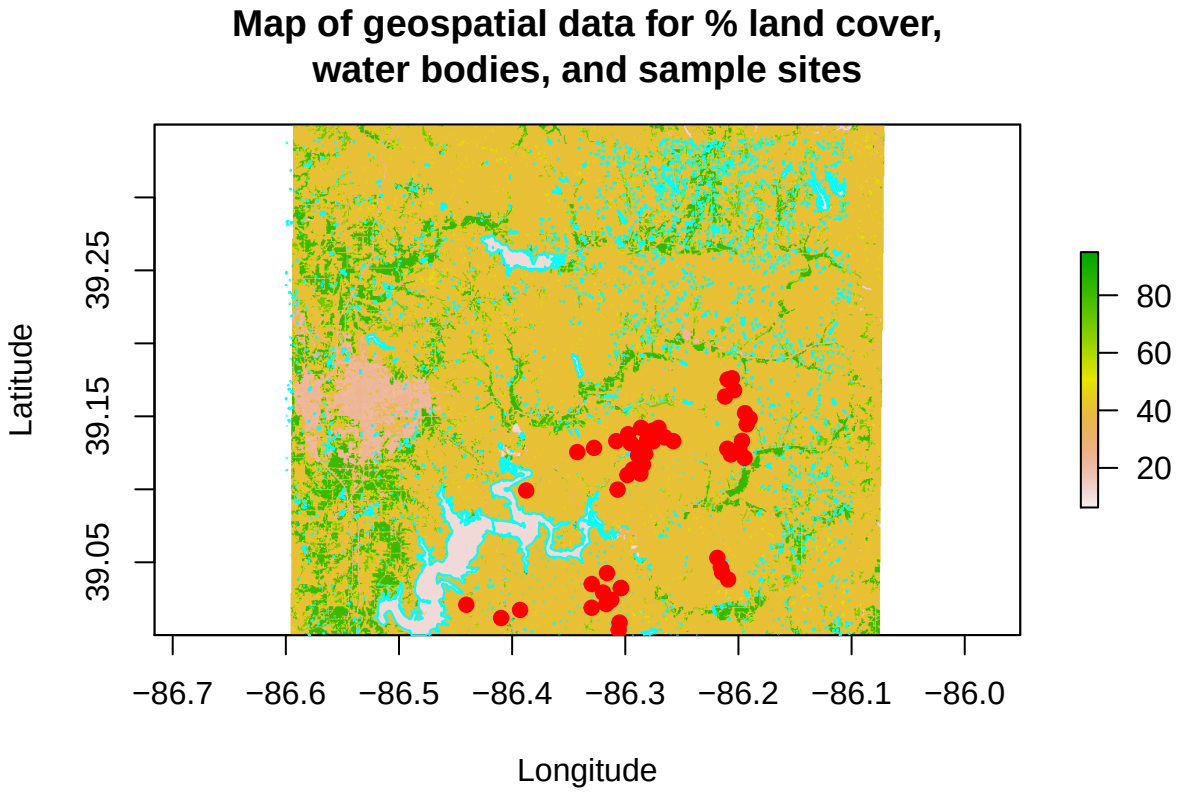


This map displays a lot of useful information that we otherwise, would not have been aware of. For example, all points are on National Forest land except for one that is north of all others. Likewise, the sample ponds appear to be aggregated in four or five small groups and distributed across a topographically 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 actually contains nothing but visual 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**.

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 some: 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 OF GEOGRAPHICAL ECOLOGY

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. Concept 1: Spatial Autocorrelation

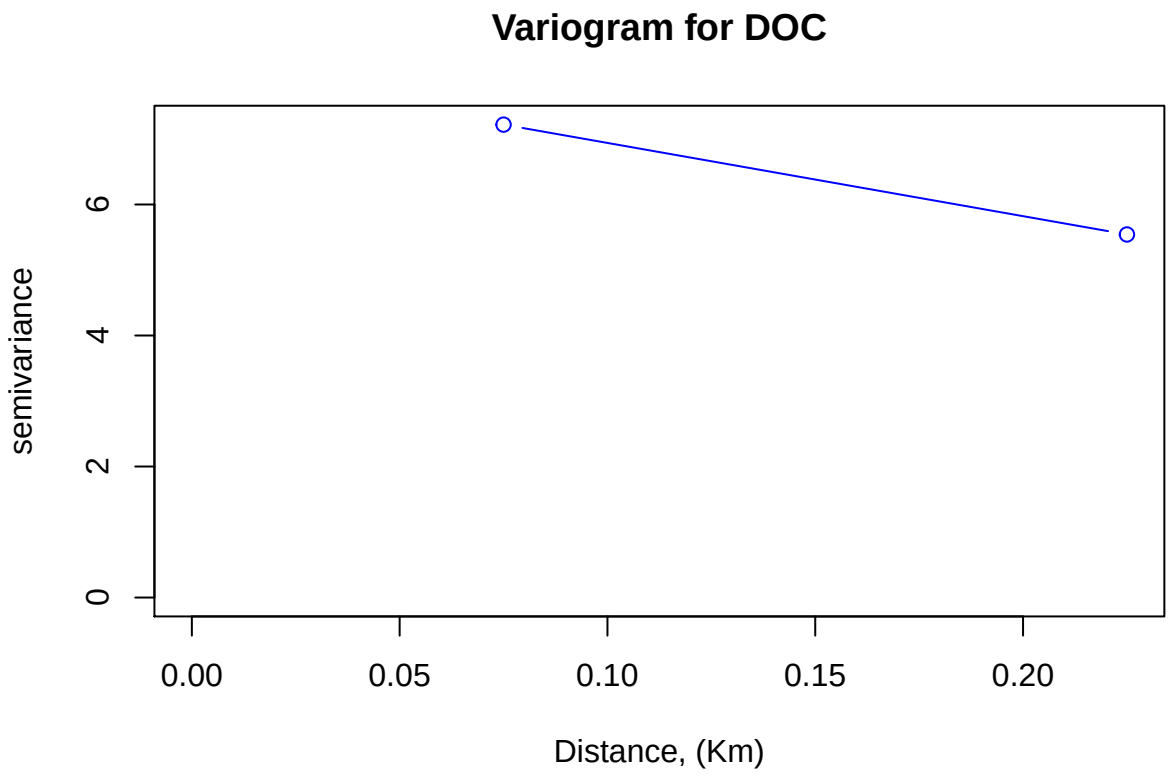
Tobler's first law of geography states that "Everything is related to everything else, but near things are more related than distant things" (Tobler 1970). This law is a formulation of the concept of spatial autocorrelation. In short, spatial autocorrelation is the degree to which spatial variables are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

This week, we reveal another 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 changes over scales of distance.

```
dists <- dist(cbind(lats, lons))
breaks = seq(0, 1.5, l = 11)

v1 <- variog(coords = cbind(lats,lons), data = Ponds$DOC, breaks = breaks)
v1.summary <- cbind(c(1:10), v1$v, v1$n)
colnames(v1.summary) <- c("lag", "semi-variance", "# of pairs")
plot(v1, type = "b", main = "Variogram for DOC", xlab='Distance, (Km)', col='blue')
```



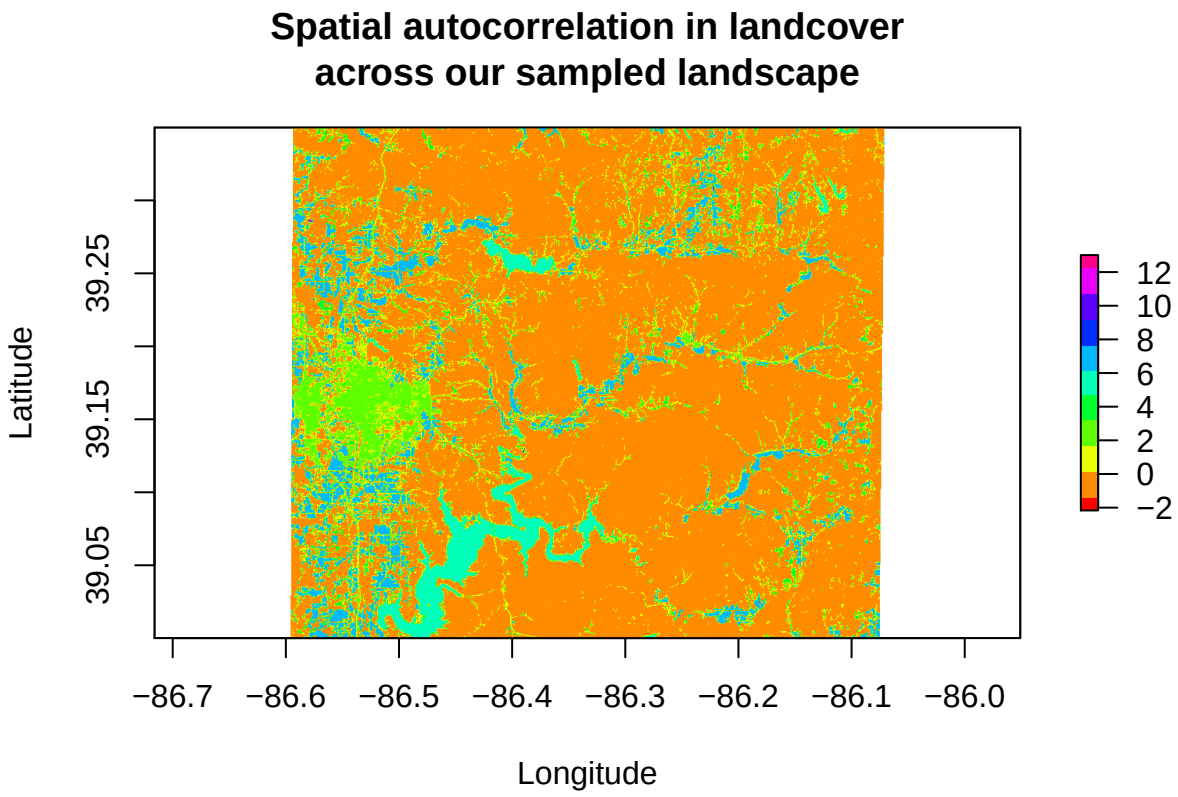
As you can see, the **semivariance** decreases with distances, where at a distance near 0.4 kilometers the semi-variance approaches 2. But, what's the semivariance? If you were to find the differences between all possible points spaced a constant distance apart and then find the variance among the differences, and then divide that variance in half, you would have the semivariance! Consequently, a variance of 4 or semivariance of 2 might be considerd small.

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 and assigns an index value, a z-score and p-value. Take for example, our land cover data represented in a raster file. Using R's `raster` package, we can calculate **global** (across the landscape) and **local** (in comparison to neighbors) measures of **Moran's I**.

For a more 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 ###SPATIAL PATTERNS### ARE clustered, dispersed, or random, with the null hypothesis that the data are not autocorrelated. Take for example, our land cover data represented in a raster file. Using R's `raster` package, we can calculate **global** (across the landscape) and **local** (in comparison to neighbors) measures of **Moran's I**.

```
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))
```



Looking at the 'landscape' of Moran's I, we can see that spatial autocorrelation, at least in land cover, is generally low.

Primary 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 patterns address the question of whether communities closer to one another are more similar than communities that are farther away from each other. Likewise, the distance-decay pattern can 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.

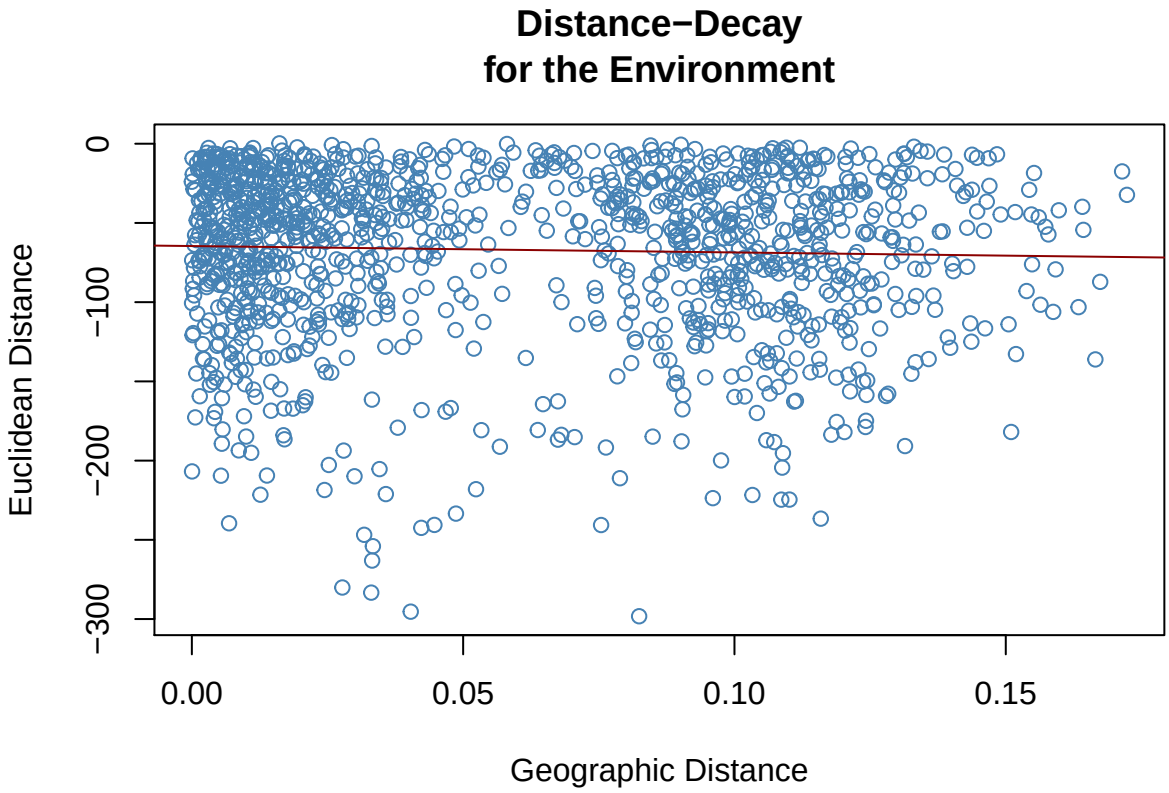
```
require("simba")

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$"Salinity")
elev <- as.numeric(Ponds$"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.

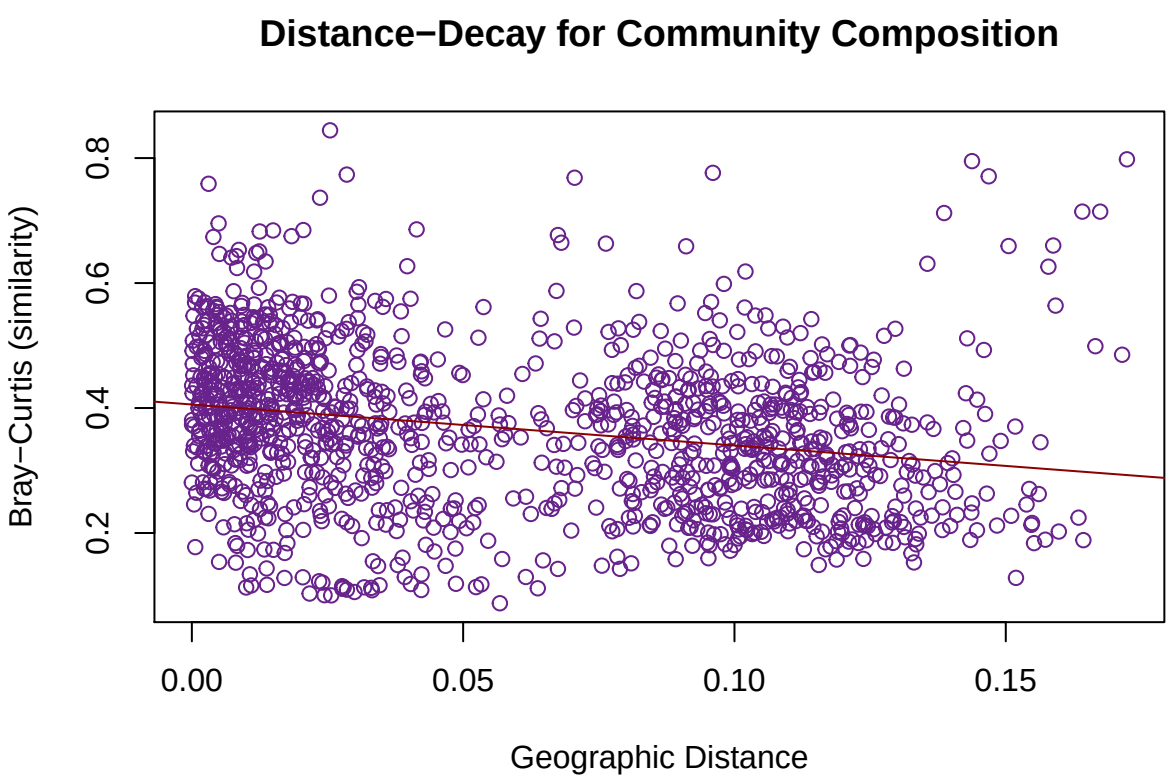
```
df <- data.frame(coord.dist.ls, env.dist.ls[,3], struc.dist.ls[,3])
names(df)[4:5] <- c("env", "struc")
attach(df) #df <- subset(df, struc != 0)
```

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

```
par(mfrow=c(1, 1))
plot(dist, env, xlab="Geographic Distance", ylab="Euclidean Distance",
     main = "Distance-Decay\nfor the Environment", col='SteelBlue')
abline(lm(env ~ dist), col="red4")
```




```
par(mfrow=c(1, 1))
plot(dist, struc, xlab="Geographic Distance", ylab="Bray-Curtis (similarity)",
     main="Distance-Decay for Community Composition", col='darkorchid4')
abline(lm(struc ~ dist), 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: -39.67
## Significance: 0.096
##
## Empirical upper confidence limits of r:
##   90%   95% 97.5%   99%
##  36.4  44.9  52.6  62.9
```

Question : It seems that microbial communities that are closer in geographic distance are also closer in compositional similarity. However, this does not seem to be the case for environmental similarity, at least, in the variables we either measured.

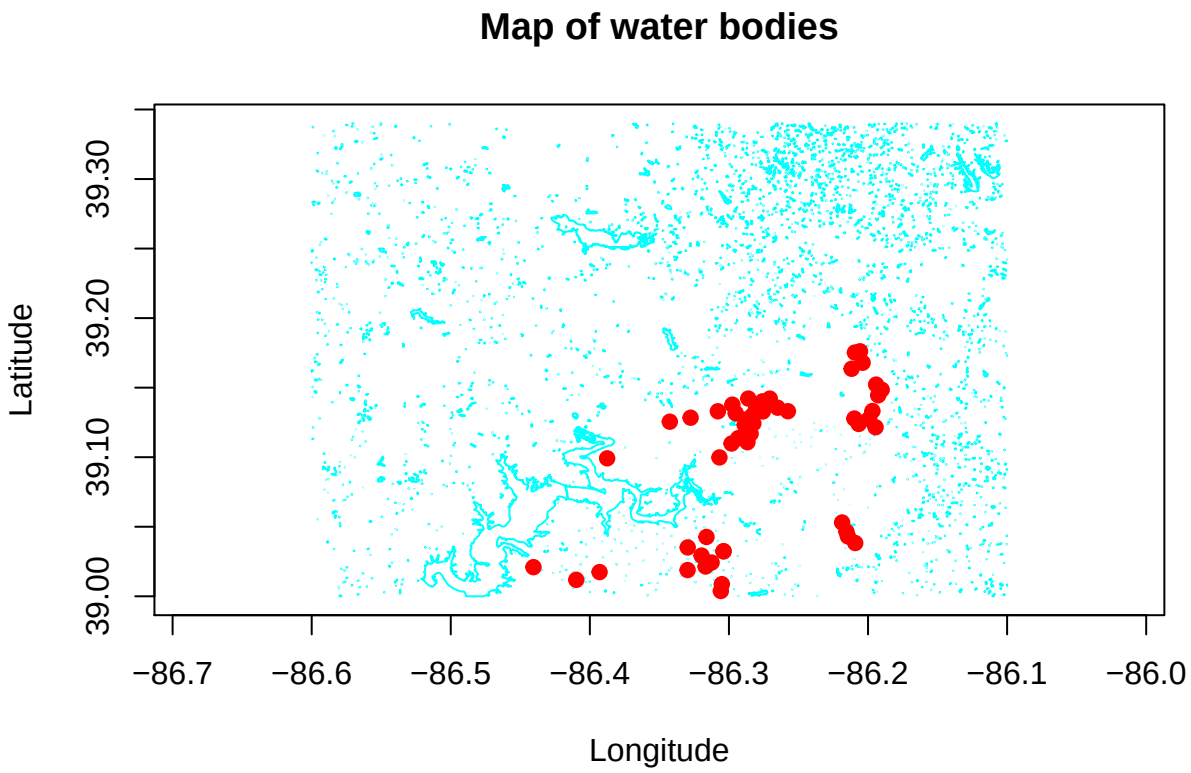
Answer :

So, rather than being spatially autocorrelated perhaps, environmental conditions are extremely patchy. This might make natural sense considering that we're examining aquatic habitat patches, i.e., refuge ponds. But then why would closer communities be more similar than far one? Perhaps, an examination of spatial aggregation in can help.

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. Take for example, the ponds in our sample area. A highly level of aggregation would suggest that if we encounter one pond, then we are likely to encounter others nearby.

```
Water.Bodies <- readShapeSpatial("water/water.shp")
plot(Water.Bodies, xlab='Longitude', ylab='Latitude',
main='Map of water bodies', border='cyan', axes=TRUE)

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



Question : What can you say about the spatial aggregation of ponds and refuge ponds?

Answer :

Likewise, a high level of aggregation would suggest that, if we find one individual of a species then we will likely find others nearby. Here, we will examine aggregation with respect to the sampled abundances (i.e. detection of gene reads) of bacterial taxa sampled among the refuge ponds.

Pattern 2: Spatial abundance distribution

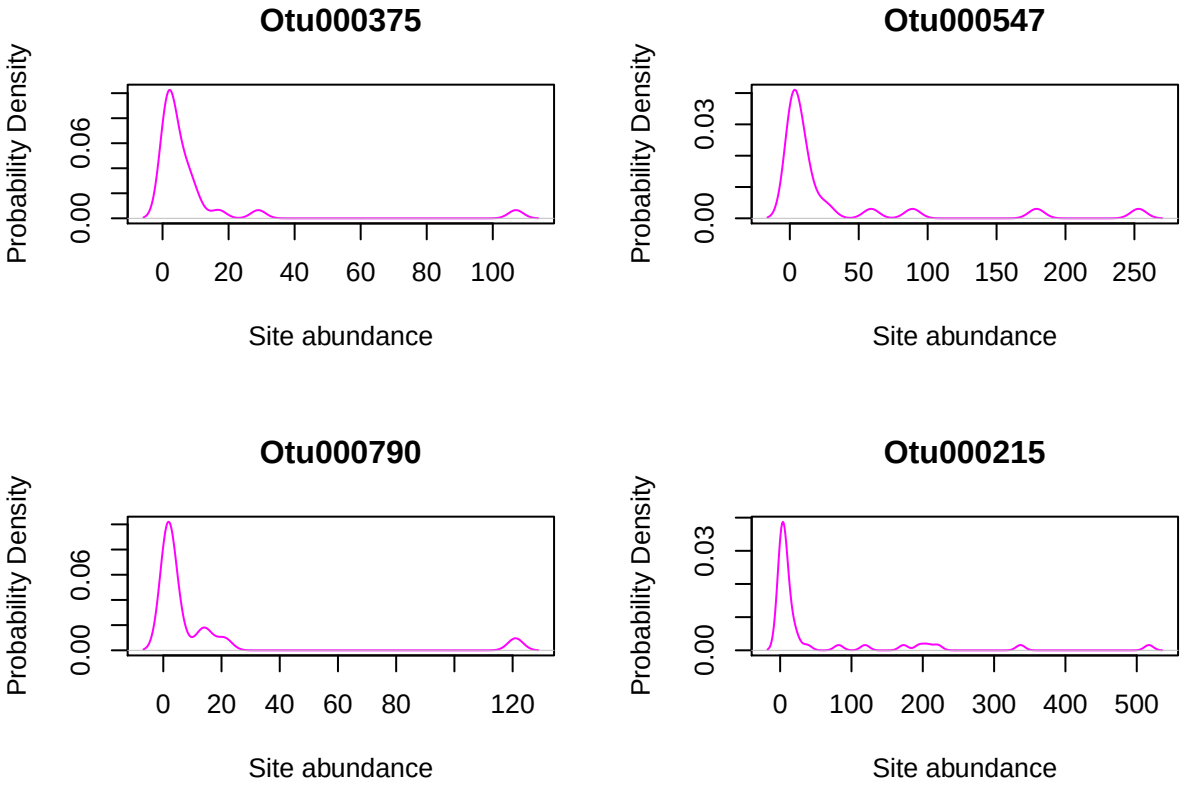
One of the primary patterns of spatial aggregation in ecology is the distribution of a species across a landscape, also referred to as the **species spatial abundance distribution (SSAD)**. The SSAD reveals the frequency at which we find a species at a particular abundance, and is similar to a histogram.

Here, we will examine SSADs for OTU's in refuge pond dataset by constructing **kernel-density curves**. Kernel density curves are analogous to histograms, but avoid the arbitrary creation of bins. 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.

Let's draw a bacterial OTU at random from the collection of refuge ponds and plot its SSAD.

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

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

ct <- 0
while (ct < 4){
  otu <- sample(1:length(OTUs), 1) # a random OTU
  ad <- ssad(otu)
  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 : Is the sampled abundance for a given OTU often aggregated? If so, how do you know, i.e., how do you interpret the patterns in the kernel density curve?

Answer 3:

Question : 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? If so, then elaborate.

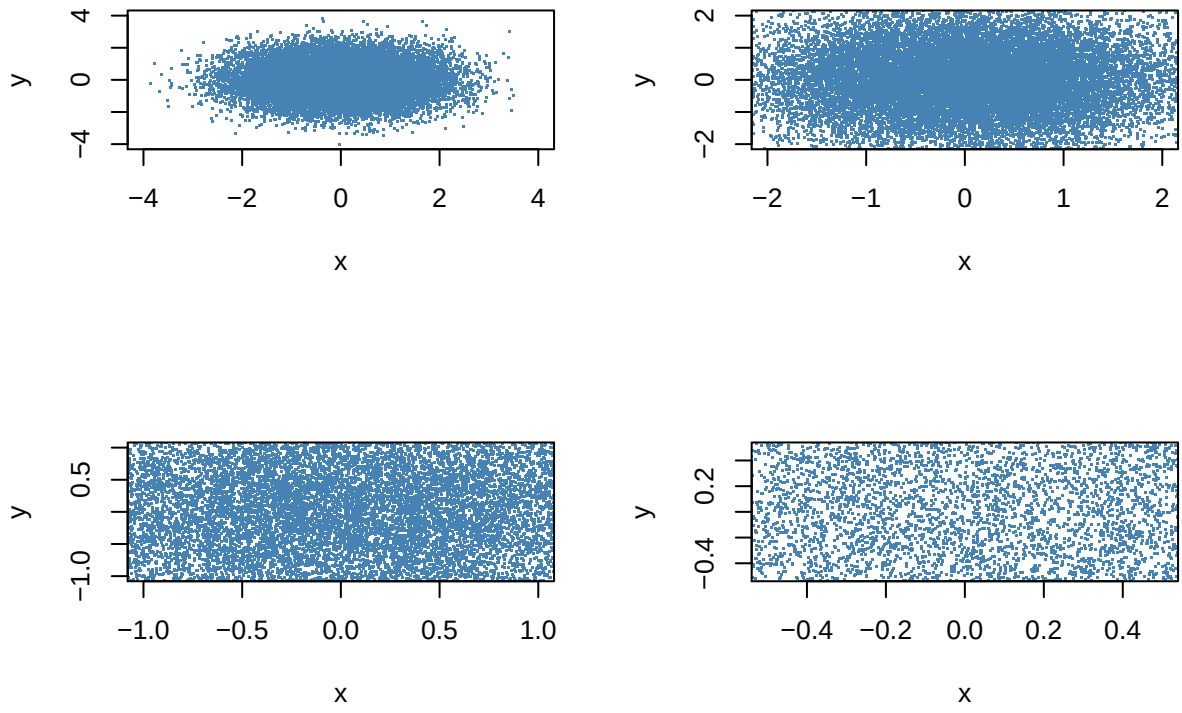
Answers :

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 extent the is measured.

Let's, we generate a random draw from a normal distribution (a.k.a., Gaussian distribution, bell curve), letting each point represent the location of a single individual, where all individuals belong to the same species.

```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: 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:

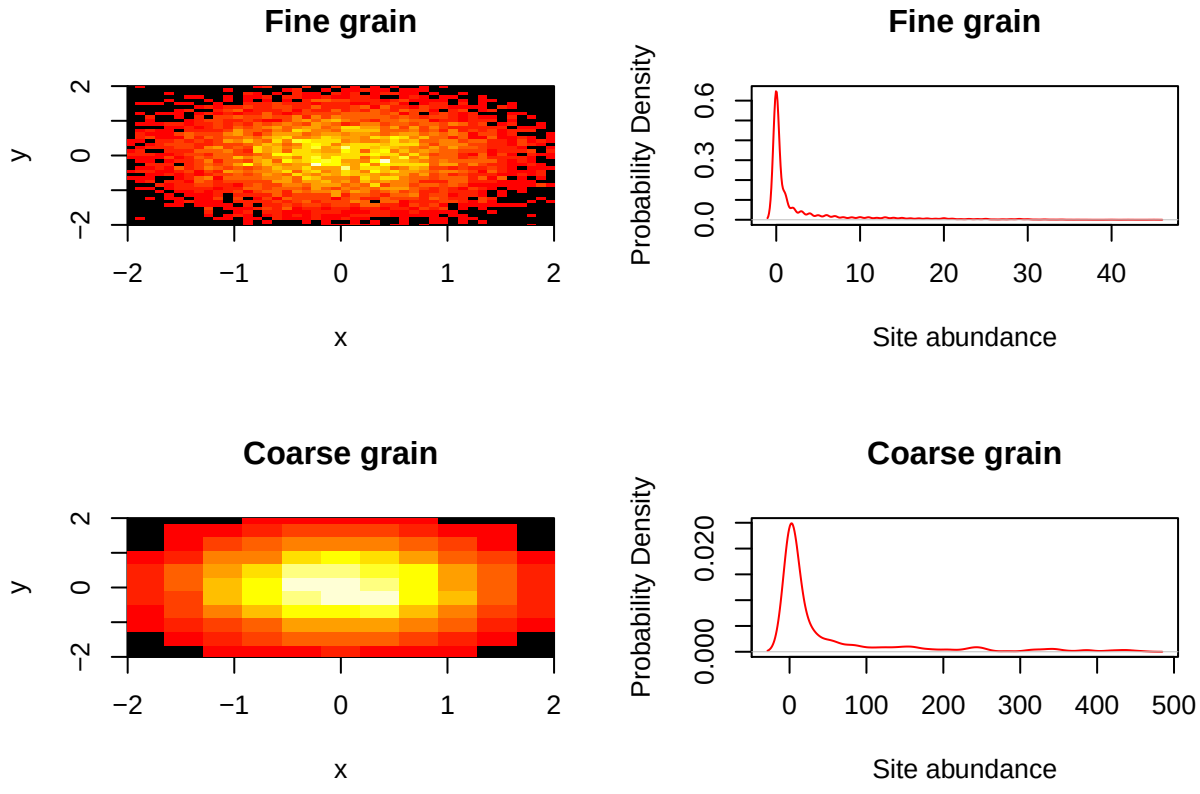
Now, let's explore the effect of changing spatial `grain`, from a fine grain to a coarse grain. We will do this while holding extent constant and will plot heat maps (i.e. 2D histogram) revealing the density of individuals in the landscape. We will then plot kernel density curves to reveal the probability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

```r
require("gplots")
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:stats':
##
##     lowess
```

```
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', ylab='y', 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', ylab='y', main = "Coarse grain" )
ad <- h4$counts
plot(density(ad), col = 'red', xlab='Site abundance',
     ylab='Probability Density',  main = "Coarse grain")
```



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

Answer :

Question : How are the kernel density curves we just generated for our randomly drawn points analogous to the species spatial abundance distributions (SSAD) that we generated for OTUs in our refuge plots? Do they basically represent the same thing? If so, how?

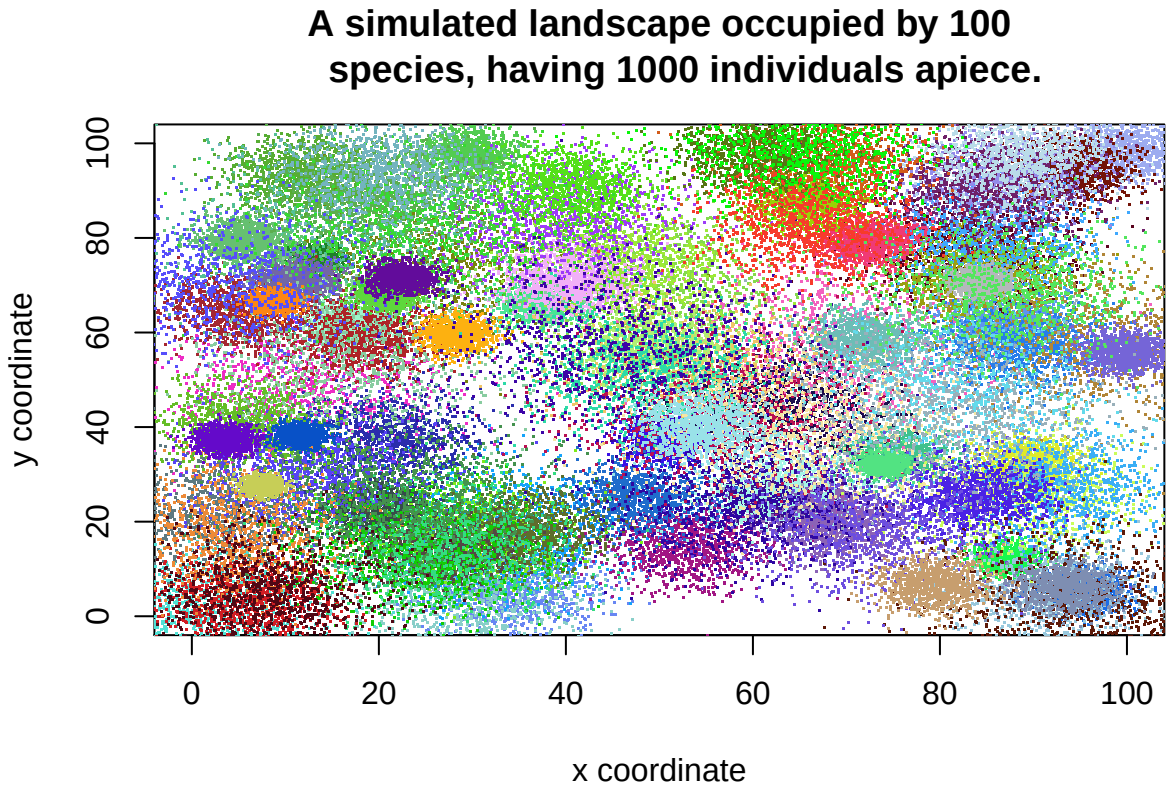
Answer :

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 as we encounter more individuals.

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 representing the spatial distribution of 50 species, each with 1000 individuals.

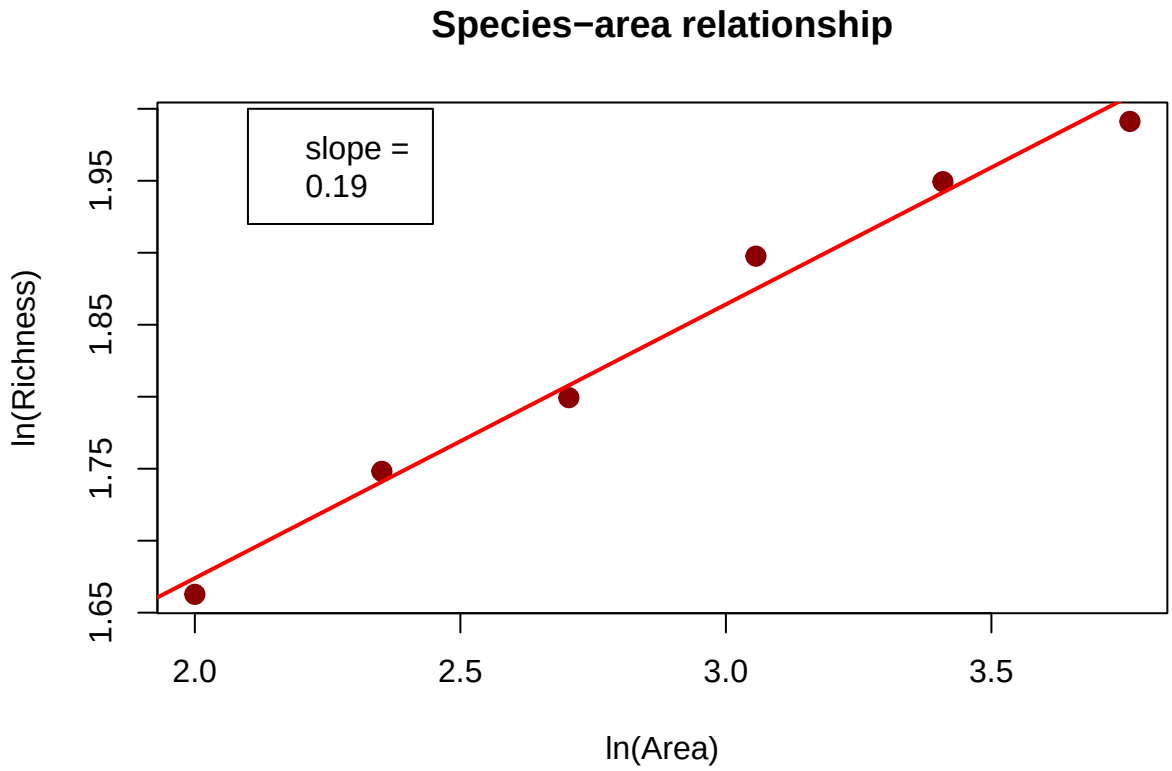
```
community <- c()
species <- c()

# 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){
  # 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 of occupied by 50 species with 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 (i.e. quadrats) by adding on additional rows and columns.

```
lim <- 10

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

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*1.5
```

```
}

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))
```




```
results
```

```
##
## Call:
## lm(formula = S.list ~ A.list)
##
## Coefficients:
## (Intercept)       A.list
##      1.2936       0.1902
```

*** Question***: What was the slop of the species-area relationship? Is this similar to the slopes you encountered in the reading?

*** Question***: 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:

Species-area relationship (SAR)

That we accumulate species, and likewise increase richness, with increasing area is not very interesting. 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 (i.e. orders of magnitude) to the increase in area, i.e., $log(S) = c + zlog(A)$

Question : Power-laws are widespread in ecology. We have seen power-laws used in this class before, that is, in our alpha-diversity exercise. Can you recall where and for what the power-law was used?

Answer :

In Arrhenius's equation, the power-law exponent represents the rate at which log(S) in increases with log(A), i.e., the scaling relationship.

Question : 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 :