Homework 4- Edward Trout

Building a Generative Model for Species Distribution

Step 1: Choose a Study Organism

Organism: Tardigrades!

Step 2: Decide on an environmental predictor variable that determines abundance for your organism.

Environmental Predictor: Silicate concentration.

a) Explain why your predictor variable is likely to represent the distribution of your organism.

As the amount of silicate in an area increases, so does habitat heterogeneity and therefore greater protection from predators and reduced chance of negative intraspecies interactions. On the otherhand, silicate concentration makes it harder for tardigrades to disperse. There will be a small negative effect of silicate on presence and a positivie effect of silicate on abundance.

Step 3: Following the R code, "simulating a landscape.R" generate a map of your environmental variable using the raster function.

This is the main function that this model will be built on- a random draw from a parameterized multivariate normal distribution.

```
rmvn <- function(n, mu = 0, V = matrix(1)) {
  p <- length(mu)
  if (any(is.na(match(dim(V), p))))
    stop("Dimension problem!")
  D <- chol(V)
  t(matrix(rnorm(n * p), ncol = p) %*% D + rep(mu, rep(n, p)))
}
```

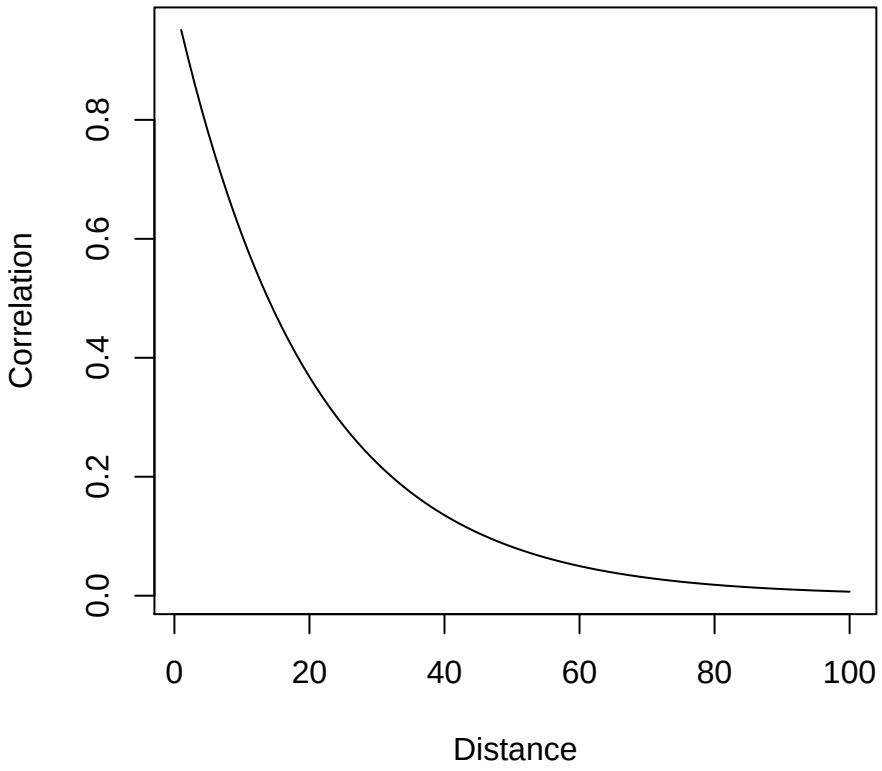
This function will simulate these draws. The mutlivariate distribution is used here to model autocorrelation in the spatial distribution of an environmental covairiate- here silicate concentration. The theory goes that the presence of something in the environment will be correlated with the presence of other like things. Concentrations of silicate particles follow this phonomenon to some degree. Now we can decide how much.

```
# Set up a square lattice region
simgrid <- expand.grid(1:50, 1:50)        # every possible combination of these numbers
n <- nrow(simgrid)

# Set up distance matrix
distance <- as.matrix(dist(simgrid))
```

This creates the spatially explicit matrix for our environmental data to go into. The grid will be 50 x 50 pixels. Each pixel represents a potential sampling point, in this scenario we will say that a pixel represents a $1cm^2$, giving the entire sampling area a size of $2500cm^2$ or $.25m^2$.

```
phi = .05 #phi determines scale of distance variation
#how does changing phi change the spatial aggregation in the plotted raster?
plot(1:100, exp(-phi * 1:100), type = "l", xlab = "Distance", ylab = "Correlation")
```



The object phi represents ϕ, the autocorrelation factor, controlling the variation of the environmental covariate over distance.

b) What is your phi value and why? (why is your environmental variable more or less spatially-autocorrelated)

I chose a ϕ of .05 as there is a healthy amount of autocorrelation with silicate. Imagining a naive pool of water, there would be an initial deposition event with high concentration followed by a slow distriution of silicate particles through the pool. Tardigrades don't usually occur in areas of high current, thus silicate dispersal would be quite passive. As we can see from the plot, there is a high degree of correlation up until a distance of 30 pixels (the inflection point).

```
Xo <-rmvn(1, rep(1, n),exp(-phi * distance))

#X is predictor variable. Does it make more sense to have a discrete or
#continuous predictor?
```

If our predictor variable was best described by the normal distribution, then the object Xo would generate the appropriate environmental covariate values for the space. As we can see, the rmvn is called by the arguments of 1 (a default mean), n (the number of spaces/pixels), and the spatial autocorrelation- combining phi (our autocorrelation factor ϕ) and the distance (space between any two pixels).

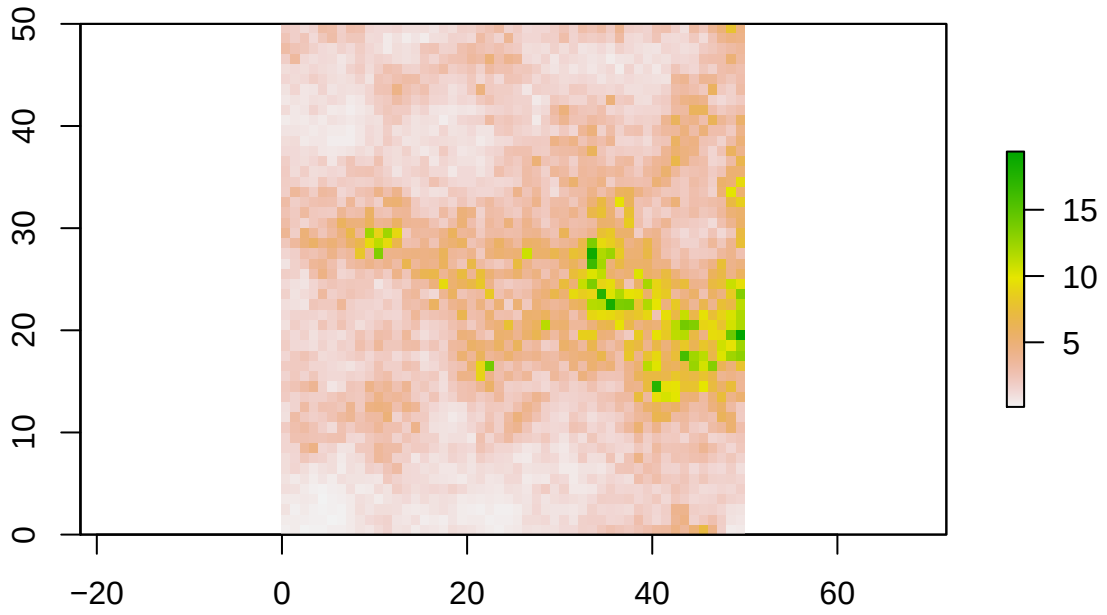
However, silicate concentration is not best described by a normal distribution, but in fact by a gamma distribution. It is continuous, but there are no non-zero values. Therefore, we will nest the multi-variate normal distribution as the mean inside of a call of the gamma distribution, creating a hierchical model.

```
X <- rgamma(n,
            rate = (20 / exp(rmvn(1, rep(1, n), exp(-phi * distance)))),
            shape = 20)    #hierarchical model!
```

The object X is now a vector of our environmental covariate values.

```
# Visualize results
Xraster <- rasterFromXYZ(cbind(simgrid[, 1:2] - 0.5, X))

plot(Xraster)   # this shows distribution of predictor variable - silicate concentration
```



The raster function attaches our x (1) and y (2) values from the grid (simgrid) to the environmental covariates (X), and shades it appropriately.

Step 4: Choose a number of points to sample for your study organism

Study Points: 500. This would be a rigorous but feasible rate of 500 water samples each from 1cm^2 points.

c) What constrains the number of points you can sample?

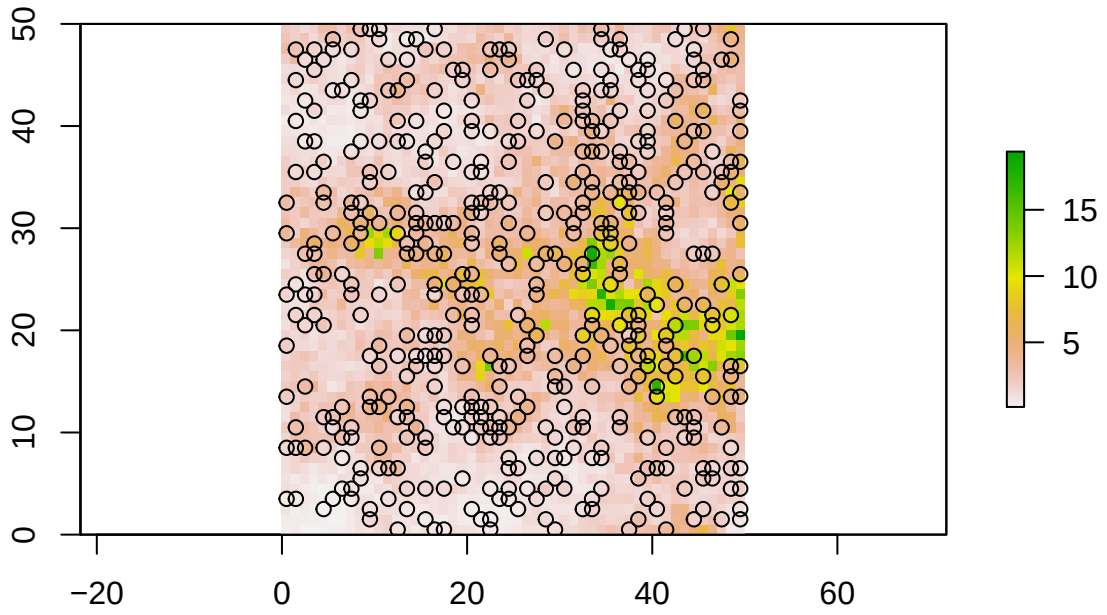
The constraints on the sample size here is really only how long an observer can crouch over a pool of water for. 500 samples would most likely take like six hours or so if two observers were at the pool.

d) Include a map of your environmental variable + sampling points as output

```
#Converting raster to a dataframe
spat_dat=rasterToPoints(Xraster)
```

```
#how many points can you sample?
GO=sample(x=c(1:nrow(spat_dat)),size=500)

plot(Xraster)
points(spat_dat[GO,c(1:2)])
```



Step 5: We will be modeling organism abundance as a hurdle model, or zero-inflated negative binomial. This type of stochastic model has two components: a presence/absence component that represents dispersal and a count component that represents abundance. For each of these components, decide on an intercept and a slope (for your environmental covariate).

e) For your organism, what is a biological reason you might observe a hurdle model?

Tardigrades are likely to observe a hurdel model because they do not disperse much in general on the observable large scales we would be measuring them on. Therefore it is pertinent to consider that there may be low establishment at the given study site, and to account for presence/absence first before abundance.

For the Presence portion of the Hurdle Model:

```
# does organism reach any given pixel at all?  effect of environmental covariate on dispersal

presence_intercept= 0.65
presence_slope= -.0475

PA=rbinom(500,plogis(presence_intercept+spat_dat[GO,3]*presence_slope),
```

```
          size=1)
```

Now, the object PA is a list of 0/1 for every point in our area of the presence (1) or absence (0) of tardigrades there based on the influence of silicate concentration in each point on dispersal.

For the Abundance portion of the Hurdle Model:

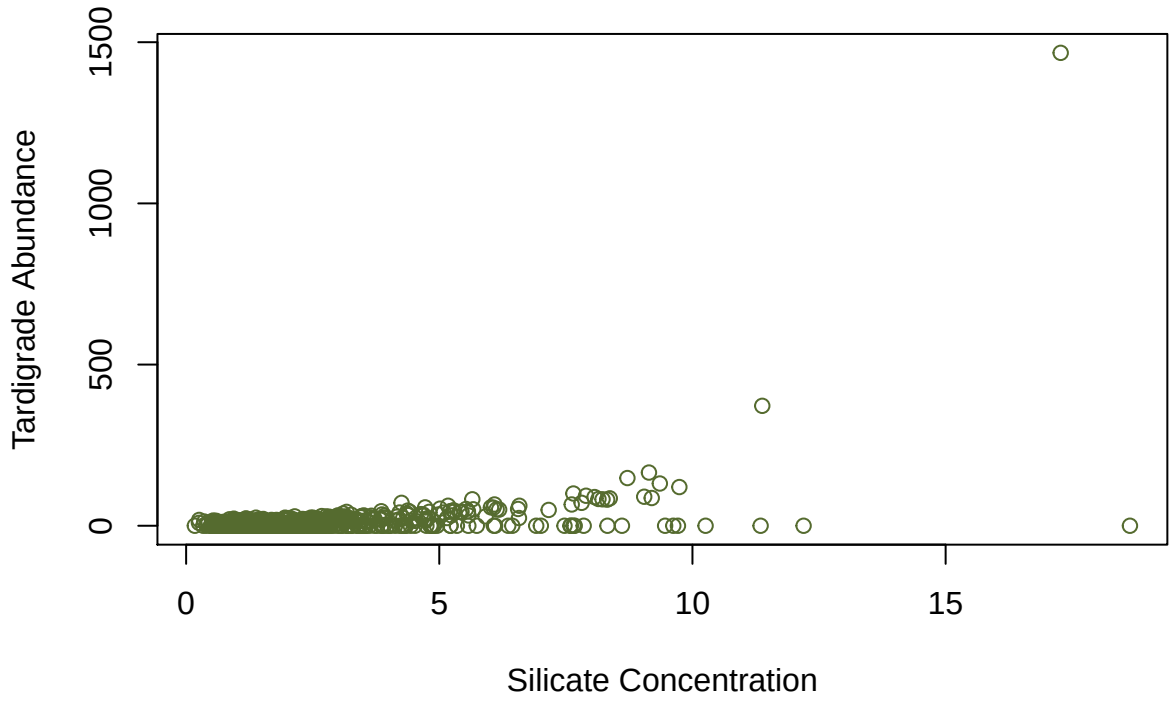
```
# does organism do well at given pixel? effect of environmental covariate on abundance
count_intercept=2
count_slope= .321
over_dispersion= 10

abundance = PA*rnbinom(500,mu=
                       exp(count_intercept+count_slope*spat_dat[GO,3]),
                     size=over_dispersion)
```

f) By multiplying the binomial and negative binomial variable to get a total abundance, what are we assuming about these distributions?

The underlying assumption for one is that both distributions are in the same units- which is not true. The binomial distribution is describing the units in terms of probability space (0 or 1) whereas the negative binomial describes units in integer counts of indididuals. It works here because the presence/absence data is being used as a logical whether to consider or not consider each point for applying the effect of the environmental covariate on abundance. i.e. when there is no tardigrades at a point, the effects of silicate concentration on tardigrade abundance should not be considered at that point.

g) Create a plot of your abundance vs. your environmental covariate



As we can see here, there is an increase in tardigrade abundance as silicate concentration increases, with the included zeroes across the board due to the independent presence/absence component to the hurdle model.

Step 6: Give a dataframe with your environmental covariate and abundance to your neighbor.

```
write.csv(data.frame(abundance, spat_dat[GO,3]), file = "tardigrades.csv")
```

h) Analyze your neighbor's data. Can you recapture the true slope and intercept parameters they used to simulate the data?

The data is from Louis, investigating the effect of water depth on the abundance of an invasive plant species.

```
louis <- read.csv("fake_nbdata.csv")

head(louis,3)
```

```
##   X abundance.nb water_depth
## 1 1            0    3.955565
## 2 2            4    7.400343
## 3 3            0   10.673958
```

First, we will make an object that reflects the presence and absence of the species, and then run a binomial glm to find the presence intercept and slope parameters.

```
louis_presence <- ifelse(louis$abundance > 0, 1, 0)

louis.bn <- glm(louis_presence ~ louis$water_depth, family = "binomial")

coef(louis.bn)
```

```
##       (Intercept) louis$water_depth
##        -0.4024688        -0.1548959
```

```
confint(louis.bn)
```

```
##                         2.5 %     97.5 %
## (Intercept)       -0.9176834  0.1137544
## louis$water_depth -0.2032622 -0.1140519
```

The parameters as decided by Louis were Intercept: -.0005, Slope: -.15. The intercept was not well estimated but the confidence interval does contain in. The slope was very well recaptured.

Next, the abundances that are not zero will be analyzed with a negative binomial glm. We will subset out the zero-abundances because we have already analyzed their importance to some degree in the presence-absence binomial model. Technically there can be (and often are) zeroes in a negative binomial glm to when analyzing count data, however because we cannot tell outright which are due to the dispersal presence-absence and which are due to the abundance, we will assume that we have effectively modelled zero abundance with the binomial glm.

```
abundance_louis <- subset(louis, louis$abundance > 0)

louis.nb <- glm.nb(abundance_louis$abundance ~ abundance_louis$water_depth)

coef(louis.nb)
```

```
##                  (Intercept) abundance_louis$water_depth
##                   1.04762237                  0.01543773
```

```
confint(louis.nb)
```

```
##                                   2.5 %     97.5 %
## (Intercept)                   0.6271657 1.46397491
## abundance_louis$water_depth -0.0252724 0.05624336
```

The parameters as decided by Louis were Intercept: .9, Slope: -.0001, Overdispersion: .95. The intercept was recaptured well, however the slope was not. The slope's confidence intervals also overlap zero, which the true slope was close to, so perhaps this is a good recapture? It is hard to tell when the slope is close to zero as it is here.

i) Given the multivariate normal we used to generate spatial patterns in our covariate, how might you analyze spatial autocorrelation for your response variable? (organism counts)

In order to efficiently account for spatial autocorrelation in our count models, we would have to add an explicit argument into the linear equation. Where we usually have a variable described by a slope and an intercept, it must now be described by a slope, an intercept, and an autocorrelation factor. $Y = a + b * x$ now becomes $Y = a + b * x + A$, where A is an autocorrelation factor. From *Dormann* et al. (2007)[1], this autocorrelation factor is controlled by the autocorrelation coefficient ρ - similar to our ϕ. Usually the autocorrelation factor is a sum of the values of Y from surrounding points all multiplied by a weighted factor of their distance.

[1] Dormann et al. (2007), *Methods to account for spatial autocorrelation in the analysis of species distributional data: a review.* Ecography 30: 609-628