HW4

Louis Jochems

March 4, 2019

library("betareg")

## Warning: package 'betareg' was built under R version 3.4.4

library("raster")

## Warning: package 'raster' was built under R version 3.4.4

## Loading required package: sp

library("rgeos")

## Warning: package 'rgeos' was built under R version 3.4.4

## rgeos version: 0.4-2, (SVN revision 581)  
## GEOS runtime version: 3.6.1-CAPI-1.10.1   
## Linking to sp version: 1.3-1   
## Polygon checking: TRUE

library("rgdal")

## Warning: package 'rgdal' was built under R version 3.4.4

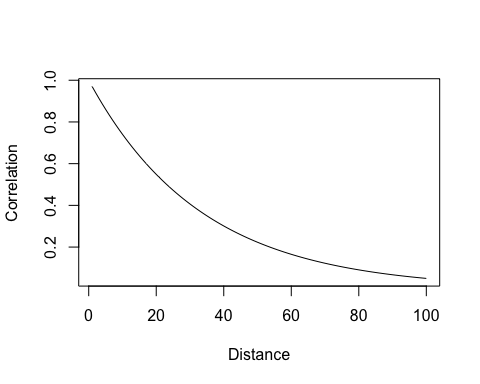
## rgdal: version: 1.3-9, (SVN revision 794)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 2.1.3, released 2017/20/01  
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/3.4/Resources/library/rgdal/gdal  
## GDAL binary built with GEOS: FALSE   
## Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ\_VERSION: 493]  
## Path to PROJ.4 shared files: /Library/Frameworks/R.framework/Versions/3.4/Resources/library/rgdal/proj  
## Linking to sp version: 1.3-1

library("MASS")

##   
## Attaching package: 'MASS'

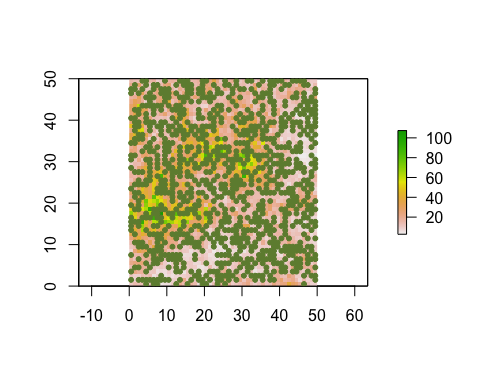
## The following objects are masked from 'package:raster':  
##   
## area, select

library("betareg")  
  
# Define function to draw random samples from a multivariate normal  
# distribution  
  
#multivariate normal, represents normal   
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, sd=1), ncol = p) %\*% D + rep(mu, rep(n, p)))  
}  
  
# Set up a square lattice region, all possible combination of those numbers (x,y coordinates)  
simgrid <- expand.grid(1:50, 1:50)  
n <- nrow(simgrid)  
  
# Set up distance matrix  
distance <- as.matrix(dist(simgrid))  
# Generate random variable  
  
phi = 0.03 #phi determines scale of distance variation, bigger more noise, smaller more autocorrelation   
#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")

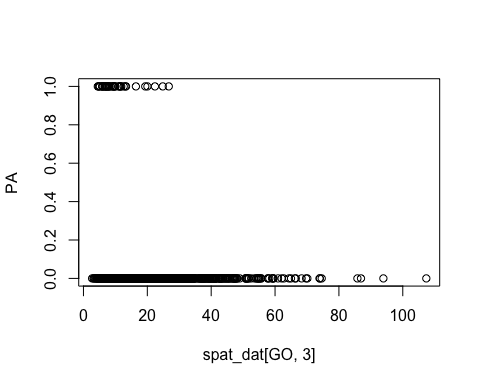


#simulating gamma for predictor, 0.03 because water depth is highly spatially autocorellated   
phi = 0.03  
  
X <-rmvn(1, rep(2.5, n),exp(-phi \* distance)) #n=50\*50 from nrow   
  
X <- rgamma(n,rate=20/exp(rmvn(1, rep(2.5, n),exp(-phi \* distance))),shape=20)  
  
  
#a) This predictor is likely to respresent the distribution of my organism because frogbit likely thrives at shallower water depth. It may not establish in too deep of waters, due to wave action and currents. It may like shallower waters because it is protected from such disturbances. Also since water\_depth is continuous and positive I chose rgamma() to get random draws for values between 0 and 150 cm depth.   
#b) My phi value is 0.03 because water depth is highly spatially autocorrelated since at any give depth, the adjacent pixel (in this case) is likely to exhibit a very similar value in depth.

#c) The main constraints on the number of points I can sample are resolution (pixel size) and extent of my raster map. I chose 1000 pixels as n because I would be realistic/robust to sample for frogbit presence/abundance in at least this many pixels within a high resolution (let's say, 10cm) image of a wetland landscape.   
  
# Visualize results d)  
Xraster <- rasterFromXYZ(cbind(simgrid[, 1:2] - 0.5, X))  
  
  
plot(Xraster)  
  
  
#Converting raster to a dataframe   
spat\_dat=rasterToPoints(Xraster)  
  
  
#how many points can you sample? choose number appropriate for study organism, represents sample points Go subset of spatdat give to neighbor  
GO=sample(x=c(1:nrow(spat\_dat)),size=1000)  
  
points(spat\_dat[GO,c(1:2)],col="darkolivegreen4", pch=20)

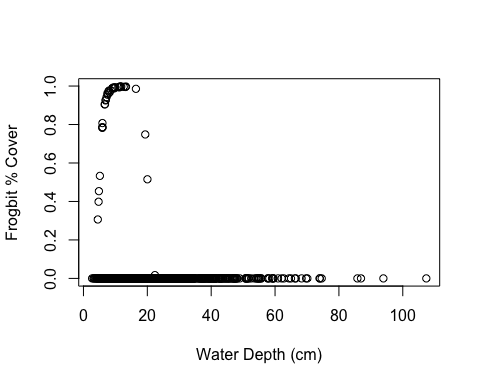


#how would you deal with spatial autocorrelation here?  
#based on biology of organism  
presence\_intercept=-1.2  
presence\_slope=-0.15  
  
  
PA=rbinom(1000,plogis(presence\_intercept+spat\_dat[GO,3]\*presence\_slope),  
 size=1)  
  
plot(PA~spat\_dat[GO,3])

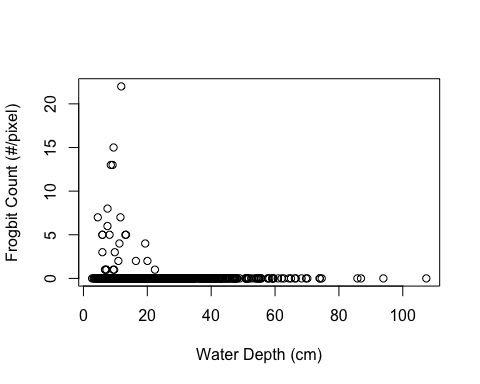


#generating abundance, values depend on FB  
count\_intercept=-10  
count\_slope=2.5  
quadslope=-0.0998  
phi=1000  
  
mean\_relationship=plogis(-10+2.5\*spat\_dat[GO,3]+-0.0998\*spat\_dat[GO,3]^2)

#hurdle model with Presence/Abundance (0, then abundance doesn't matter), error with this   
abundance=PA\*rbeta(1000,shape1=mean\_relationship\*1000,shape2=(1-mean\_relationship)\*1000)  
  
#for percent cover of pixels in remotely sensed image   
plot(abundance~spat\_dat[GO,3], xlab= "Water Depth (cm)",ylab="Frogbit % Cover")



#or nb if we were sampling quadrats in field   
count\_intercept=0.9  
count\_slope=-0.001  
over\_dispersion=0.95  
  
abundance.nb=PA\*rnbinom(1000,mu=exp(count\_intercept+count\_slope\*spat\_dat[GO,3]),size=over\_dispersion)  
  
plot(abundance.nb~spat\_dat[GO,3], xlab="Water Depth (cm)", ylab="Frogbit Count (#/pixel)")



#e) A biological reason for why I might observe a hurdle model with frogbit is because water depth may not only influence whether frogbit is present in a given pixel but also influence its abundance once it establishes in a site. Therefore, we need a hurdle model to assess counts/percent cov in pixels after the threshold of presence (ie. 1) is crossed.   
#f) We are assuming that these distributions are independent from one another.   
#g) see both % cov and count plots

tardi=read.csv("tardigrades.csv")  
  
PA1=ifelse(tardi$abundance>0,1,0)  
  
m1=glm(PA1~tardi$silica\_content, family="binomial")  
coef(m1)

## (Intercept) tardi$silica\_content   
## 0.54651115 0.04727023

plogis(0.54651115)

## [1] 0.6333258

confint(m1)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 0.31126681 0.7806142  
## tardi$silica\_content -0.05660658 0.1668128

summary(m1)

##   
## Call:  
## glm(formula = PA1 ~ tardi$silica\_content, family = "binomial")  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6205 -1.4262 0.9230 0.9442 0.9551   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.54651 0.11948 4.574 4.78e-06 \*\*\*  
## tardi$silica\_content 0.04727 0.05612 0.842 0.4   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 648.68 on 499 degrees of freedom  
## Residual deviance: 647.92 on 498 degrees of freedom  
## AIC: 651.92  
##   
## Number of Fisher Scoring iterations: 4

#for count data

abundance2 <- tardi$abundance[which(tardi$abundance>0)]  
envcov2 <- tardi$abundance[which(tardi$abundance>0)]  
m2=glm.nb(abundance2~envcov2) #env cov 3 column of spatdat, nb because high dispersion parameter of 30.45

coef(m2)

## (Intercept) envcov2   
## 2.437255825 0.005615242

confint(m2)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 2.365587873 2.509586  
## envcov2 0.004682797 NA

summary(m2)

##   
## Call:  
## glm.nb(formula = abundance2 ~ envcov2, init.theta = 3.388694346,   
## link = log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -6.0729 -0.7743 -0.2987 0.3162 3.8224   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.4372558 0.0346081 70.42 <2e-16 \*\*\*  
## envcov2 0.0056152 0.0002156 26.04 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(3.3887) family taken to be 1)  
##   
## Null deviance: 1621.50 on 323 degrees of freedom  
## Residual deviance: 312.31 on 322 degrees of freedom  
## AIC: 2187.9  
##   
## Number of Fisher Scoring iterations: 1  
##   
##   
## Theta: 3.389   
## Std. Err.: 0.306   
## Warning while fitting theta: alternation limit reached   
##   
## 2 x log-likelihood: -2181.9

#h)  
  
Presence/Absence binomial

|  |  |
| --- | --- |
| True Parameters | Estimated |
| Slope =-0.0475 | 0.0473 |
| Intercept= 0.65 | 0.633 |

Negative Binomial Count

|  |  |
| --- | --- |
| True Parameters | Estimated |
| Slope =0.321 | 0.0056 |
| Intercept= 2 | 2.43 |
| Overdispersion=10 | 3.3887 |

#i) I would likely analyze the SA for counts of my data with a correlogram and/or semivariogram. I could also check the phi parameter in the model output (at least in the negative binomial model)