

# Brown County Ponds

*Ken Locey*

*January 27, 2015*

The purpose of this document is to explore patterns and relationships of biodiversity and biogeography for a set of 52 pond samples from central Indiana, mainly Brown Country.

## 1.) SETUP

### Retrieve and Set Your Working Directory

```
rm(list=ls())
getwd()
setwd("~/Desktop/RGIS")
```

### Basic packages

```
#install.packages("sp") # classes for spatial data
#install.packages("raster") # grids, rasters
#install.packages("rasterVis") # raster visualisation
#install.packages("maptools")
#install.packages("rgeos", type="source") # and their dependencies# and their dependencies
#install.packages("RgoogleMaps")
#install.packages("googleVis")

#install.packages("BiodiversityR")
require("BiodiversityR")
```

```
## Loading required package: BiodiversityR
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'BiodiversityR'
```

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

```
## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.0-10
```

```
require("sp") # classes for spatial data
```

```
## Loading required package: sp
```

```

require("raster")  # grids, rasters

## Loading required package: raster

require("rasterVis") # raster visualisation

## Loading required package: rasterVis
## Loading required package: latticeExtra
## Loading required package: RColorBrewer
## Loading required package: hexbin

require("maptools")

## Loading required package: maptools
## Checking rgeos availability: TRUE

require("rgeos") # and their dependencies

## Loading required package: rgeos
## rgeos version: 0.3-8, (SVN revision 460)
## GEOS runtime version: 3.4.2-CAPI-1.8.2 r3921
## Polygon checking: TRUE

require("dismo")

## Loading required package: dismo

require("RgoogleMaps") # overlays of data onto a Google map.

## Loading required package: RgoogleMaps

require("googleVis")

## Loading required package: googleVis
##
## Welcome to googleVis version 0.5.8
##
## Please read the Google API Terms of Use
## before you start using the package:
## https://developers.google.com/terms/
##
## Note, the plot method of googleVis will by default use
## the standard browser to display its output.
##
## See the googleVis package vignettes for more details,
## or visit http://github.com/mages/googleVis.
##
## To suppress this message use:
## suppressPackageStartupMessages(library(googleVis))

```

## 2.) Mapping

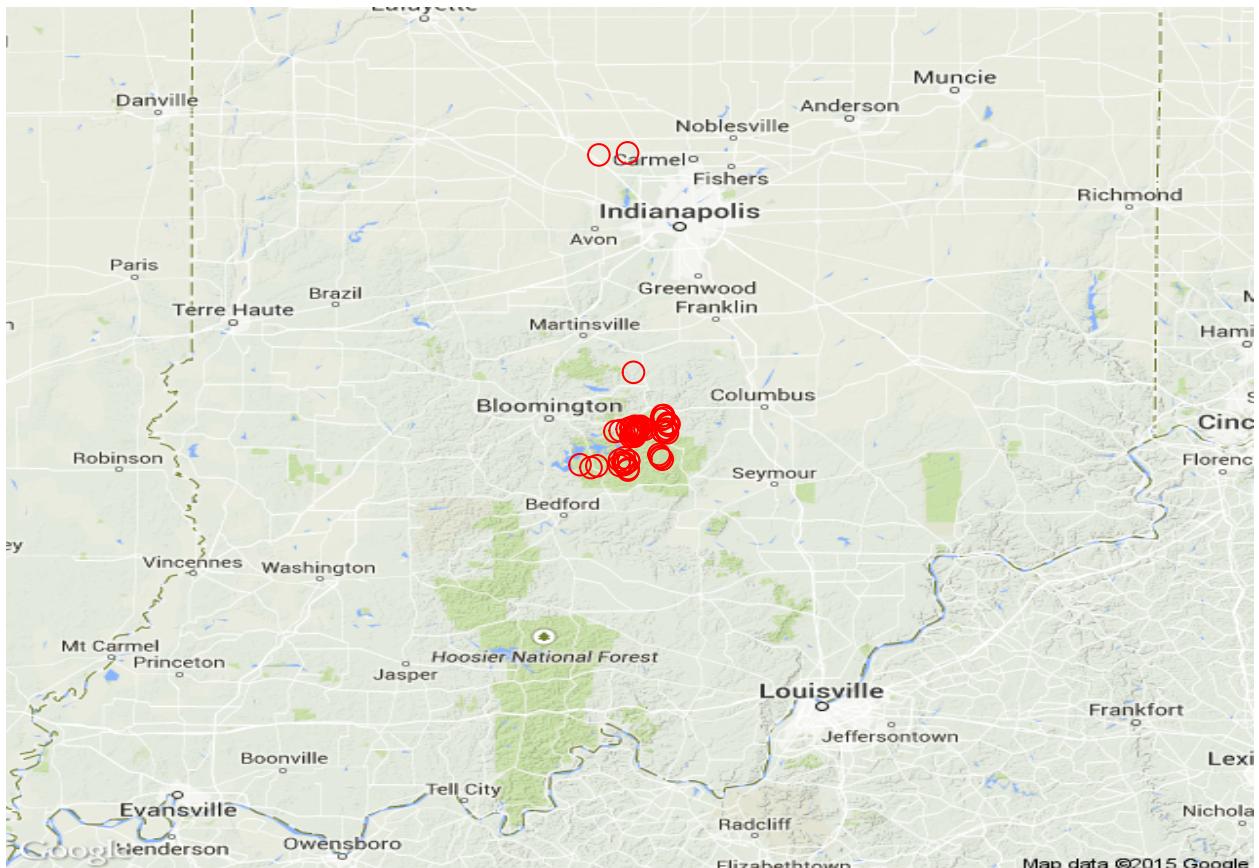
### Map sample ponds onto Google map

Retrieve base maps from Google with the `gmap` function in Species distribution modeling package `dismo`.

```
# Map centered on Brown County
newmap <- GetMap(center = c(39.1,-86.3), zoom = 8, destfile = "BrownComap.png", maptype="terrain")

PlotOnStaticMap(newmap, zoom = 8, cex = 2, col='blue')
Ponds <- read.table(file="BrownCoData/20130801_PondData.csv", head=TRUE, sep=",") 

lats <- Ponds[, 3]
lons <- Ponds[, 4]
PlotOnStaticMap(newmap, lats, lons, zoom=8, cex=1.5, col='red', add = FALSE)
```



### Plot sample ponds on web-page to search, scroll, magnify

```
names <- Ponds[, 1]
latlongs <- transform(names, latlongs = interaction(lats, lons, sep=':'))

M1 <- gvisMap(latlongs, "latlongs", "X_data", options = list(showTip=TRUE, showLine=TRUE, enableScrollW
```

```
## starting httpd help server ... done
```

### 3.) Simple relationships of environment to demographics

#### Environmental parameters

```
EnvData <- Ponds[5:21]
Temp <- EnvData[6]
```

#### A function for calculating OTU richness

```
OTU.obs <- function(x = ""){
  rowSums(x > 0) * 1
}
```

#### Generate data object for S and N of dormant and active across samples

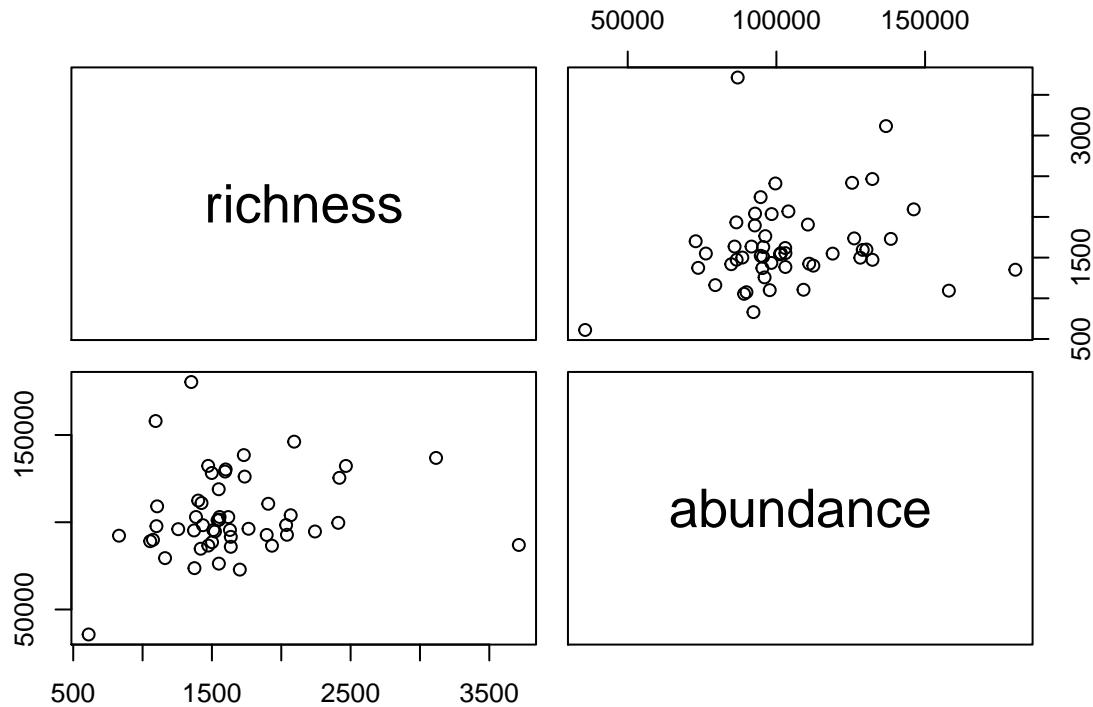
```
# Active
Active <- read.csv(file="BrownCoData/active.csv", head=TRUE, sep=",")
Active <- Active[-1] # remove first column (names)
Active.S <- as.matrix(OTU.obs(Active))
Active.N <- as.matrix(rowSums(Active)) # no. reads

# Dormant
Dormant <- read.csv(file="BrownCoData/dormant.csv", head=TRUE, sep=",")
Dormant <- Dormant[-1] # remove first column (names)
Dormant.S <- as.matrix(OTU.obs(Dormant))
Dormant.N <- as.matrix(rowSums(Dormant)) # no. reads
```

#### Scatter plots and correlation of sample-based N and S

Because regardless of what they mean, Ken is always thinking about N and S.

```
A.dem <- cbind(Active.S, Active.N)
labels <- c('richness', 'abundance')
pairs(A.dem, labels)
```



```

cor <- cor(A.dem)
cor

##          [,1]      [,2]
## [1,] 1.0000000 0.1958592
## [2,] 0.1958592 1.0000000

mydata <- as.data.frame(Active.N)
fit <- lm(Active.S ~ Active.N, data=mydata)
summary(fit)

##
## Call:
## lm(formula = Active.S ~ Active.N, data = mydata)
##
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -786.05 -254.62  -88.06  197.51 2138.08 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.201e+03  3.246e+02   3.700 0.000537 ***
## Active.N    4.297e-03  3.042e-03   1.412 0.164059    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

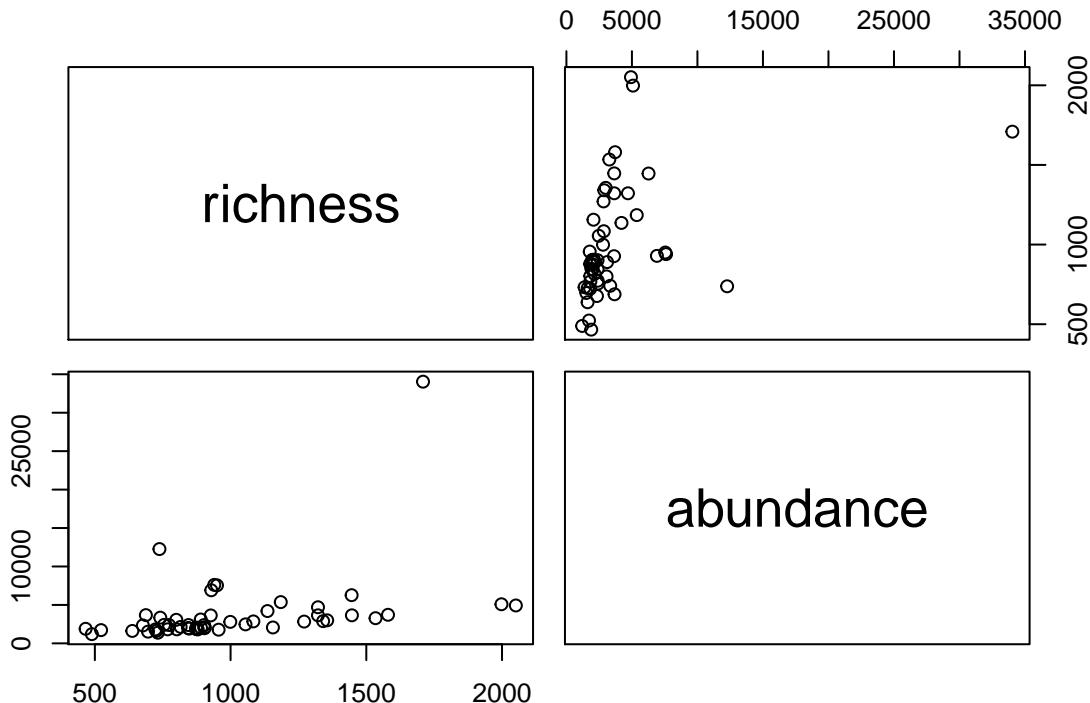
## 
## Residual standard error: 518.4 on 50 degrees of freedom
## Multiple R-squared:  0.03836,   Adjusted R-squared:  0.01913
## F-statistic: 1.995 on 1 and 50 DF,  p-value: 0.1641

```

```

Ddem <- cbind(Dormant.S, Dormant.N)
labels <- c('richness', 'abundance')
pairs(Ddem, labels)

```



```

cor <- cor(Ddem)
cor

```

```

##           [,1]      [,2]
## [1,] 1.0000000 0.3862878
## [2,] 0.3862878 1.0000000

```

```

mydata <- as.data.frame(Dormant.N)
fit <- lm(Dormant.S ~ Dormant.N, data=mydata)
summary(fit)

```

```

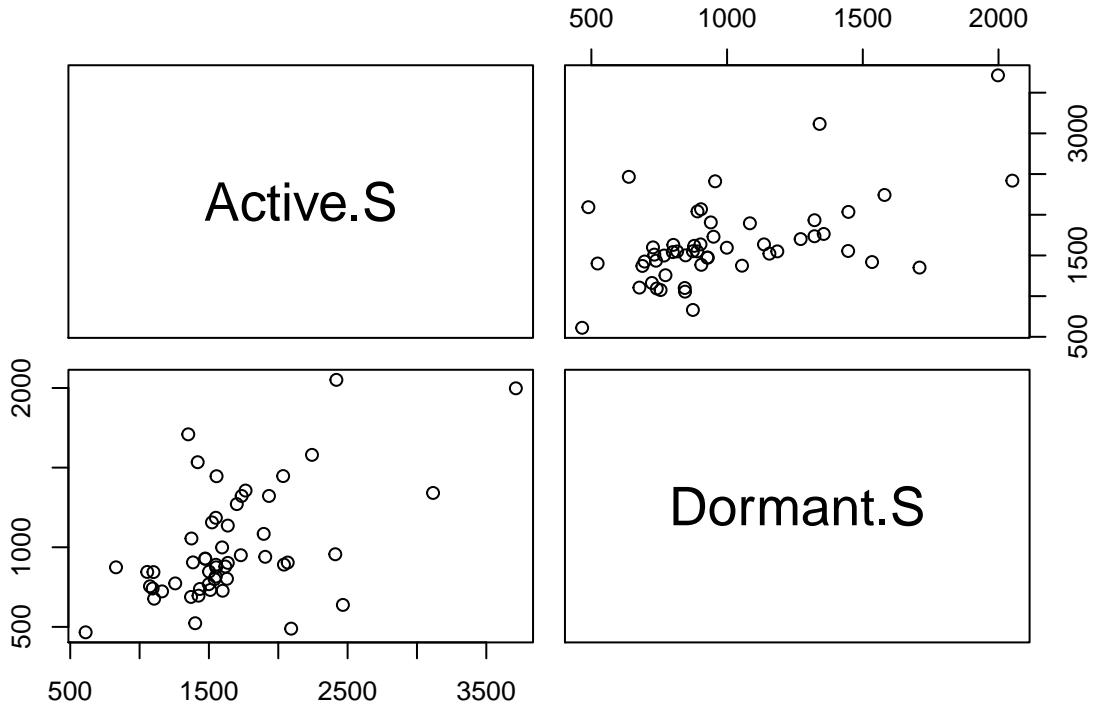
##
## Call:
## lm(formula = Dormant.S ~ Dormant.N, data = mydata)
## 
```

```

## Residuals:
##      Min       1Q   Median      3Q     Max
## -503.00 -188.07  -81.53  129.87 1020.22
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.896e+02  5.823e+01 15.278 < 2e-16 ***
## Dormant.N   2.865e-02  9.674e-03  2.961  0.00468 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 326.3 on 50 degrees of freedom
## Multiple R-squared:  0.1492, Adjusted R-squared:  0.1322
## F-statistic: 8.769 on 1 and 50 DF,  p-value: 0.004676

AvD.S <- cbind(Active.S, Dormant.S)
labels <- c('Active.S', 'Dormant.S')
pairs(AvD.S, labels)

```



```

cor <- cor(AvD.S)
cor

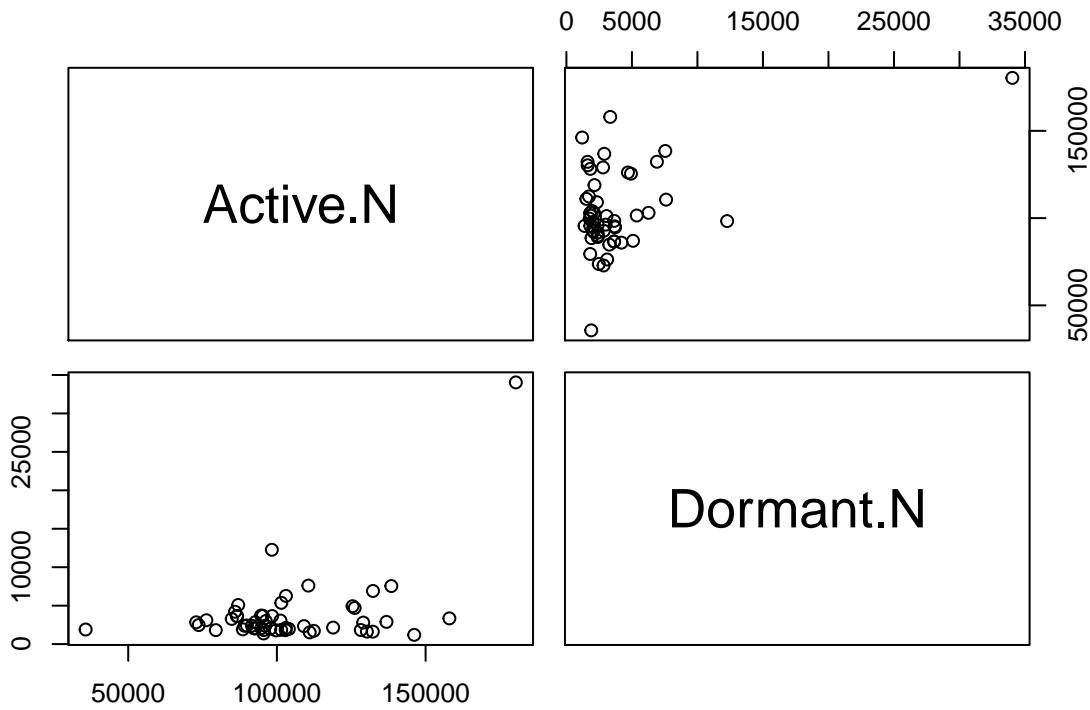
##          [,1]      [,2]
## [1,] 1.0000000 0.5421979
## [2,] 0.5421979 1.0000000

```

```

AvD.N <- cbind(Active.N, Dormant.N)
labels <- c('Active.N', 'Dormant.N')
pairs(AvD.N, labels)

```



```

cor <- cor(AvD.N)
cor

```

```

##          [,1]      [,2]
## [1,] 1.0000000 0.4551134
## [2,] 0.4551134 1.0000000

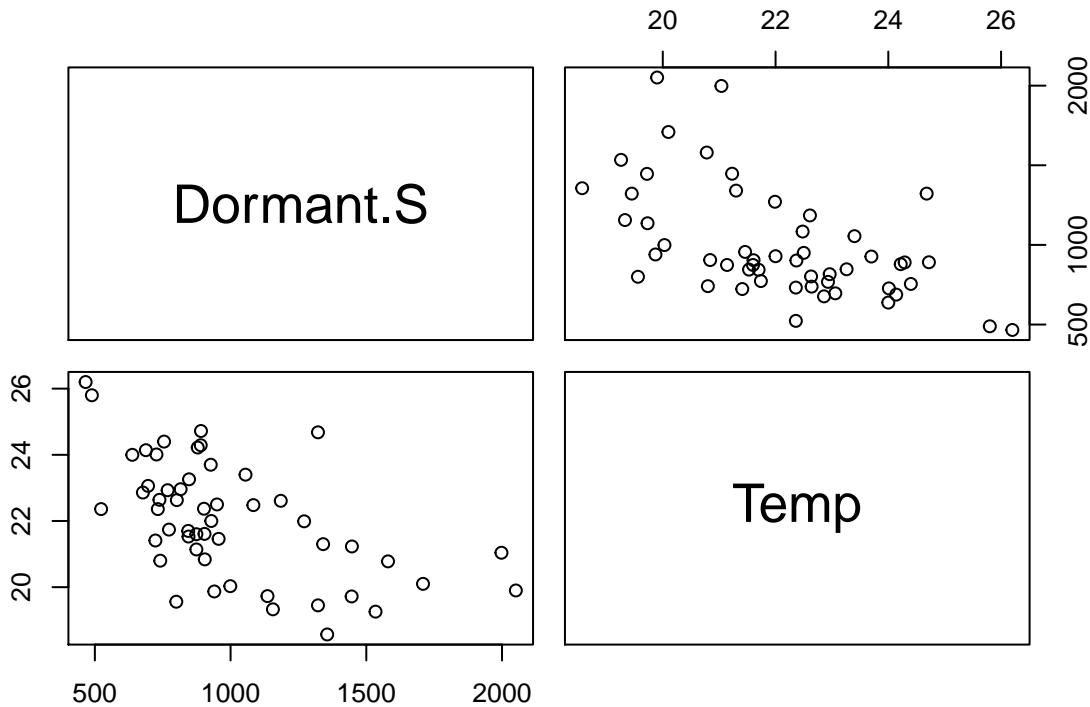
```

Simple sample demographics vs. Environmental variables

```

# Dormant richness vs. Temp
SvTemp <- cbind(Dormant.S, Temp)
labels <- c('Dormant.S', 'Temp')
pairs(SvTemp, labels)

```



```
cor <- cor(SvTemp)
cor
```

```
##           Dormant.S      Temp
## Dormant.S  1.0000000 -0.5607961
## Temp       -0.5607961  1.0000000
```

```
mydata <- as.data.frame(Temp)
fit <- lm(Dormant.S ~ Temp, data=mydata)
summary(fit) # kinda sorta
```

```
##
## Call:
## lm(formula = Dormant.S ~ Temp, data = mydata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -472.5 -192.6  -76.2  152.2  889.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3433.59     510.13   6.731 1.58e-08 ***
## Temp        -110.48      23.07  -4.789 1.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

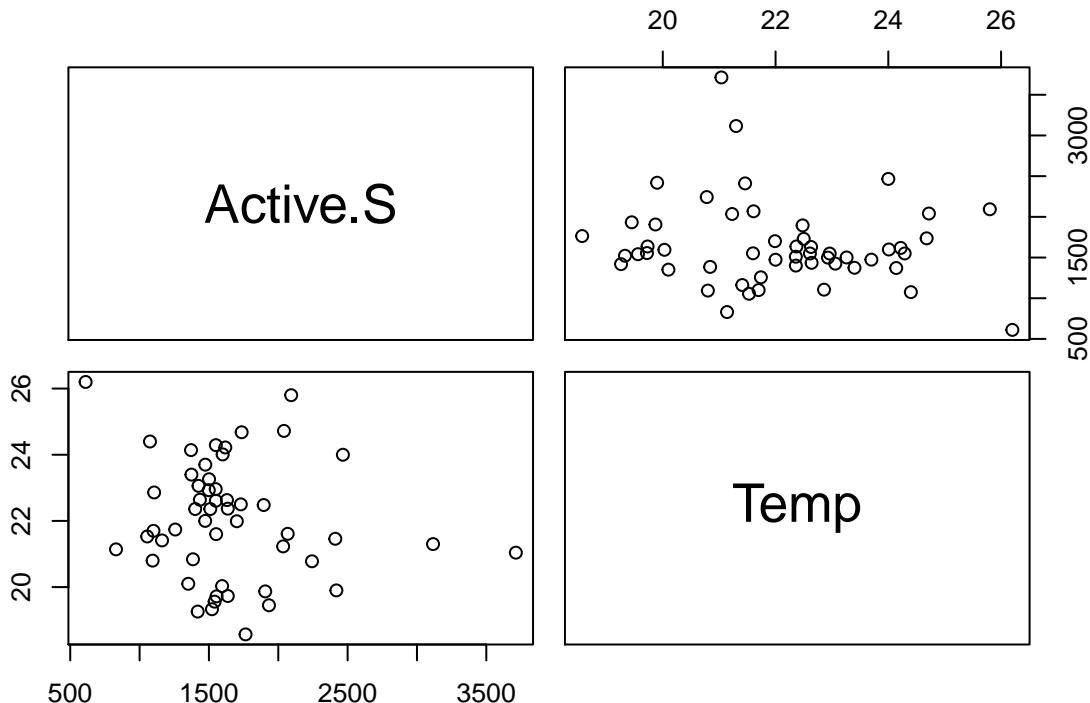
## 
## Residual standard error: 292.9 on 50 degrees of freedom
## Multiple R-squared:  0.3145, Adjusted R-squared:  0.3008
## F-statistic: 22.94 on 1 and 50 DF,  p-value: 1.532e-05

```

```

# Active richness vs. Temp
SvTemp <- cbind(Active.S, Temp)
labels <- c('Active.S', 'Temp')
pairs(SvTemp, labels)

```



```

cor <- cor(SvTemp)
cor

```

```

##          Active.S      Temp
## Active.S  1.0000000 -0.1538089
## Temp     -0.1538089  1.0000000

```

```

mydata <- as.data.frame(Temp)
fit <- lm(Active.S ~ Temp, data=mydata)
summary(fit) # wonk wonk

```

```

## 
## Call:
## lm(formula = Active.S ~ Temp, data = mydata)

```

```

## 
## Residuals:
##   Min     1Q Median     3Q    Max
## -859.2 -237.0 -104.2  161.3 2019.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2646.41    909.72   2.909   0.0054 **
## Temp        -45.28     41.14  -1.101   0.2763
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 522.4 on 50 degrees of freedom
## Multiple R-squared:  0.02366, Adjusted R-squared:  0.00413
## F-statistic: 1.212 on 1 and 50 DF, p-value: 0.2763

```

### Messing around with Temperature, Salinity, and Temporo-mandibular Syndrome (TDS)

```

fit <- lm(Dormant.S ~ Temp, data= EnvData)
summary(fit)

```

```

## 
## Call:
## lm(formula = Dormant.S ~ Temp, data = EnvData)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -472.5 -192.6  -76.2  152.2  889.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3433.59     510.13   6.731 1.58e-08 ***
## Temp        -110.48     23.07  -4.789 1.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 292.9 on 50 degrees of freedom
## Multiple R-squared:  0.3145, Adjusted R-squared:  0.3008
## F-statistic: 22.94 on 1 and 50 DF, p-value: 1.532e-05

```

```

fit <- lm(Active.S ~ Salinity + TDS, data= EnvData)
summary(fit)

```

```

## 
## Call:
## lm(formula = Active.S ~ Salinity + TDS, data = EnvData)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -726.87 -268.06  -82.55  173.49 2081.02
## 
```

```

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1928.63     158.25 12.188 < 2e-16 ***
## Salinity    -15249.73    6910.12 -2.207 0.032039 *
## TDS         362.82      97.91   3.705 0.000537 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 468.6 on 49 degrees of freedom
## Multiple R-squared:  0.23, Adjusted R-squared:  0.1986
## F-statistic: 7.318 on 2 and 49 DF,  p-value: 0.001656

```

## 4.) alpha-Diversity

Evenness

Diversity

Local richness Estimation

A function using vegan's estimateR function

```

S.est <- function(x = ""){
  est <- estimateR(x)
  chao1 <- est[2]
  ace <- est[4]
  return(chao1)
}

```

Plotting kernel density curves for estimated Richness

```

par(mfrow=c(1, 2))

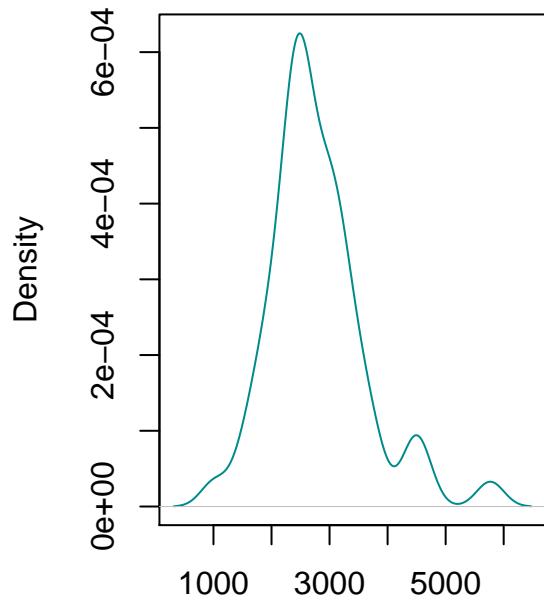
Active.S.est <- apply(Active, 1, S.est)
Dormant.S.est <- apply(Dormant, 1, S.est)

ASx <- density(Active.S.est)
DSx <- density(Dormant.S.est)

plot(ASx, main = "Est. Richness Active, Chao1", col = 'DarkCyan')
plot(DSx, main = "Est. Richness Dormant, Chao1", col = 'magenta')

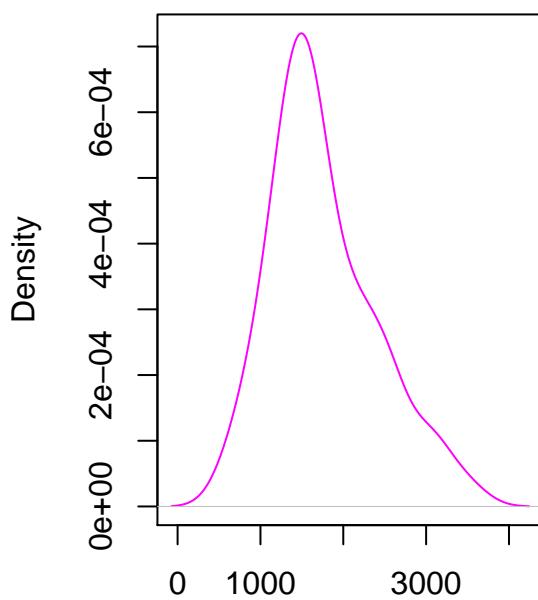
```

**Est. Richness Active, Chao1**



N = 52 Bandwidth = 234.2

**Est. Richness Dormant, Chao1**



N = 52 Bandwidth = 249.4

## Coverage

A functions for coverage (C)

```
# Good's Coverage == Bad Logic
Goods <- function(x = ""){
  1.0 - (sum(x == 1) / sum(x))
}
```

Plotting kernel density curves for coverage

```
par(mfrow=c(1, 2))

Active.C <- apply(Active, 1, Goods)
summary(Active.C)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 0.9814  0.9911  0.9924  0.9923  0.9940  0.9967

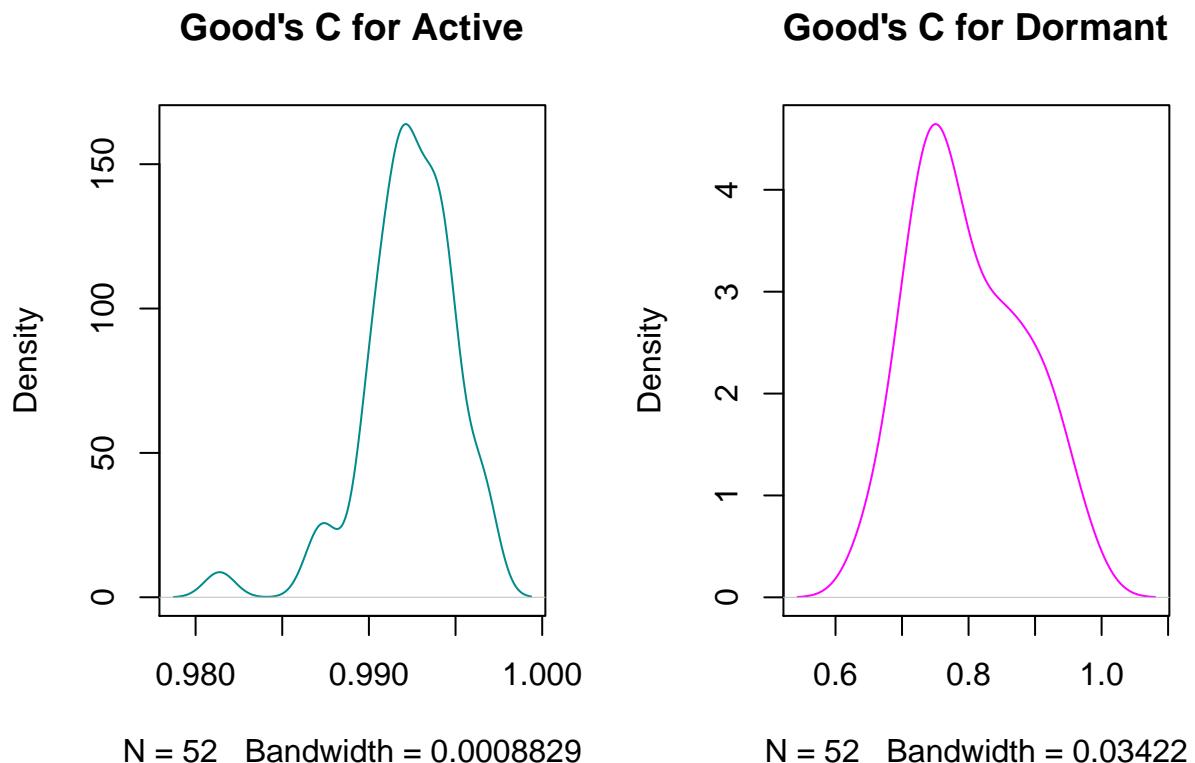
Ax <- density(Active.C)

Dormant.C <- apply(Dormant, 1, Goods)
summary(Dormant.C)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##  0.6459  0.7354  0.7760  0.7991  0.8600  0.9777
```

```
Dx <- density(Dormant.C)

plot(Ax, main = "Good's C for Active", col = 'DarkCyan')
plot(Dx, main = "Good's C for Dormant", col = 'magenta')
```

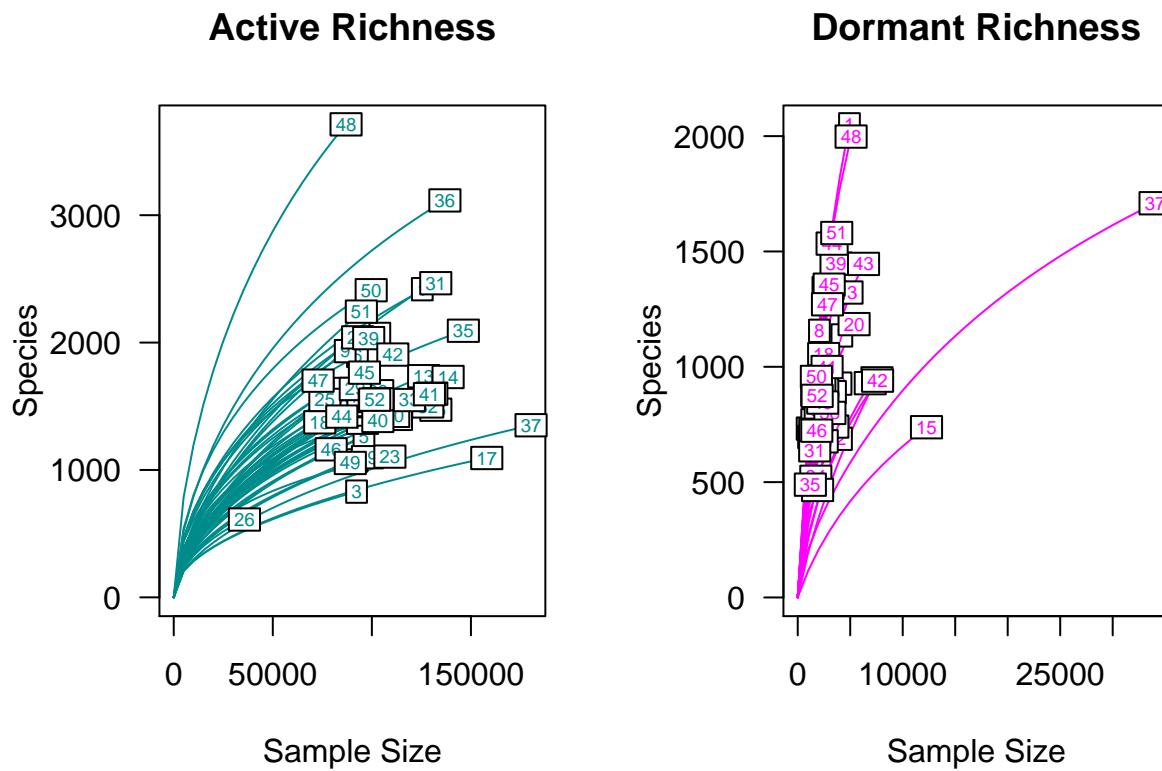


### Rarefaction

Plotting rarefaction curves for each site's Active and Dormant

```
par(mfrow=c(1,2))
min.N <- min(rowSums(Active))
S.rarefy <- rarefy(x = Active, sample = min.N, se = TRUE)
rarecurve(x = Active, main = "Active Richness", step = 5000, col = "DarkCyan", cex = 0.6, las=1)

min.N <- min(rowSums(Dormant))
S.rarefy <- rarefy(x = Dormant, sample = min.N, se = TRUE)
rarecurve(x = Dormant, main = "Dormant Richness", step = 1000, col = "magenta", cex = 0.6, las=1)
```



## 5.) beta-Diversity

```
# INSERT NIFTY ANALYSES OF UNIVARIATE, DISTANCE, AND DISSIMILARITY METRICS
```

## 6.) Gamma Gamma Gamma-Diversity

### Richness Estimation (gamma)

```
Active.gamma <- specpool(Active)
Dormant.gamma <- specpool(Dormant)
```

### OTU-accumulation curves (gamma)

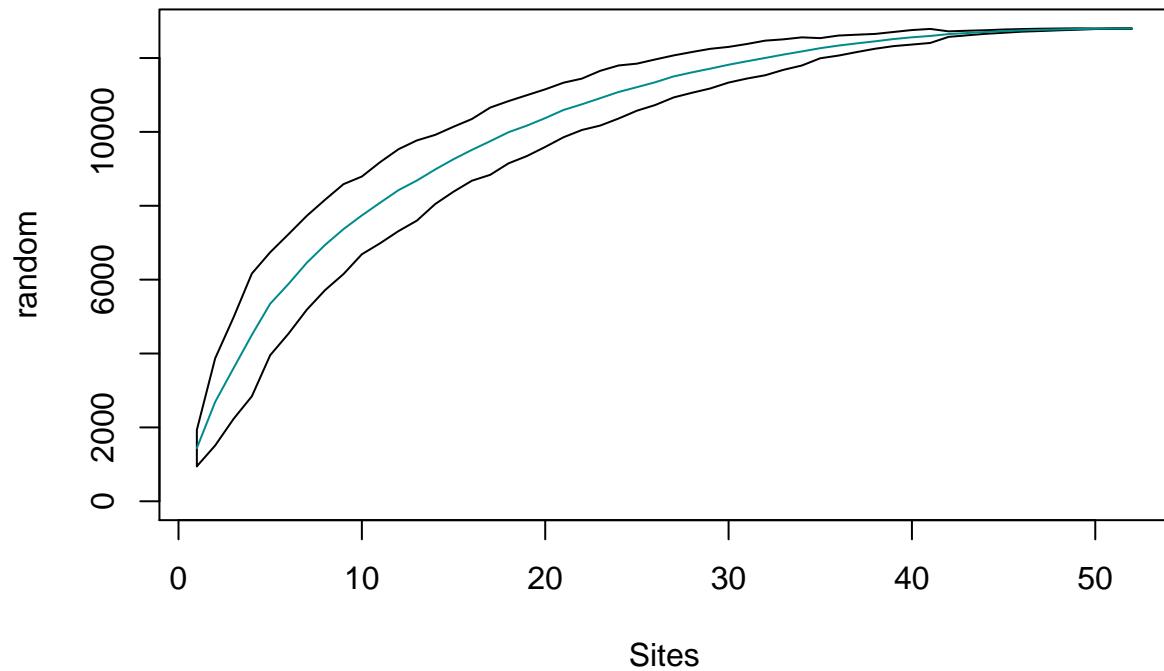
For Active

```
x <- specaccum(Active, method = "random", permutations = 20, ci.type = "polygon", conditioned = TRUE, g

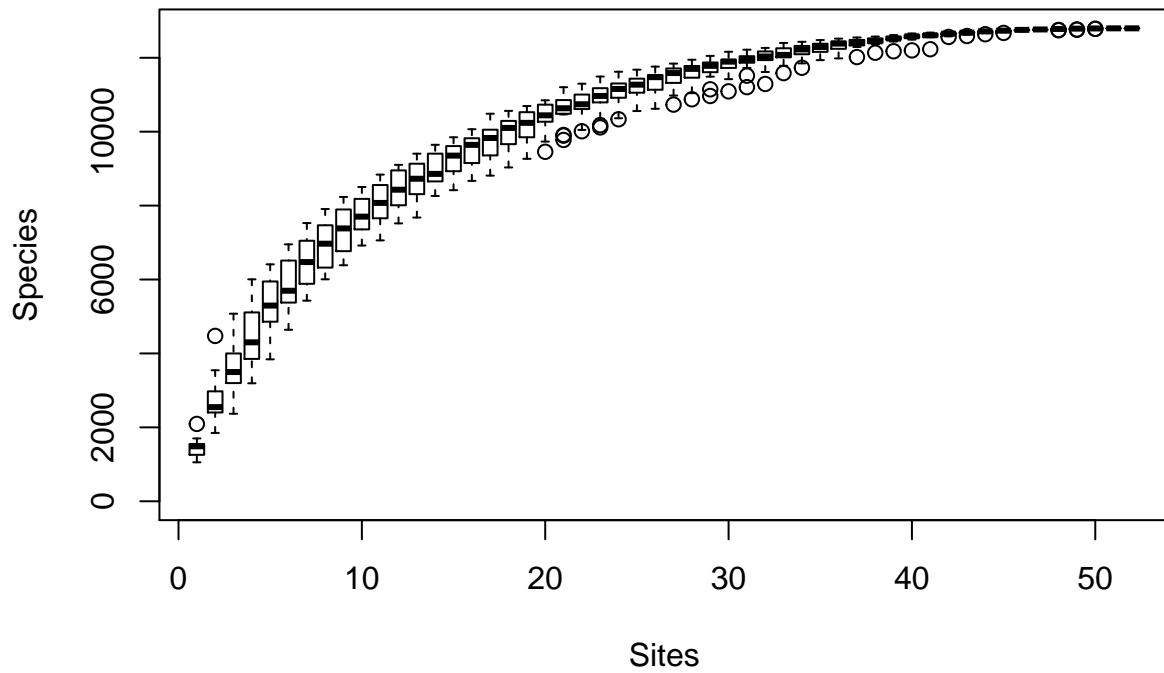
plot(x, add = FALSE, ci = 2, ci.type = "polygon", col = "DarkCyan", ci.col = col, ci.lty = 1, xlab = "S

## Warning in polygon(c(xaxvar, rev(xaxvar)), c(x$richness - ci * x$sd,
## rev(x$richness + : supplied color is neither numeric nor character
```

## Active gamma–diversity



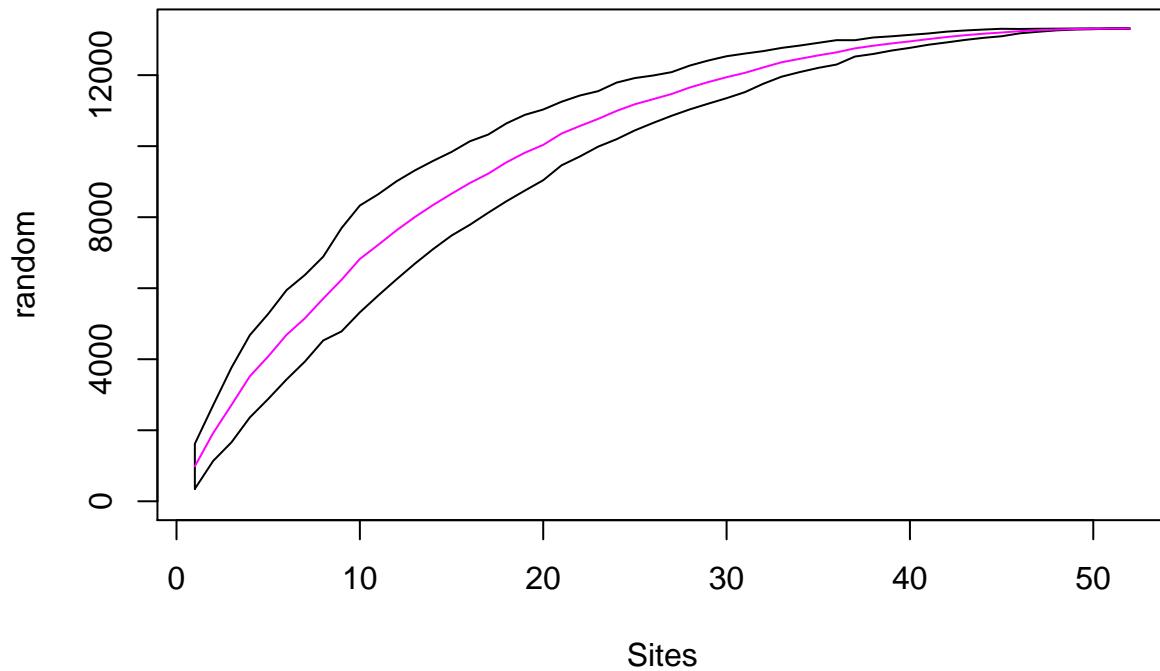
```
boxplot(x, add = FALSE)
```



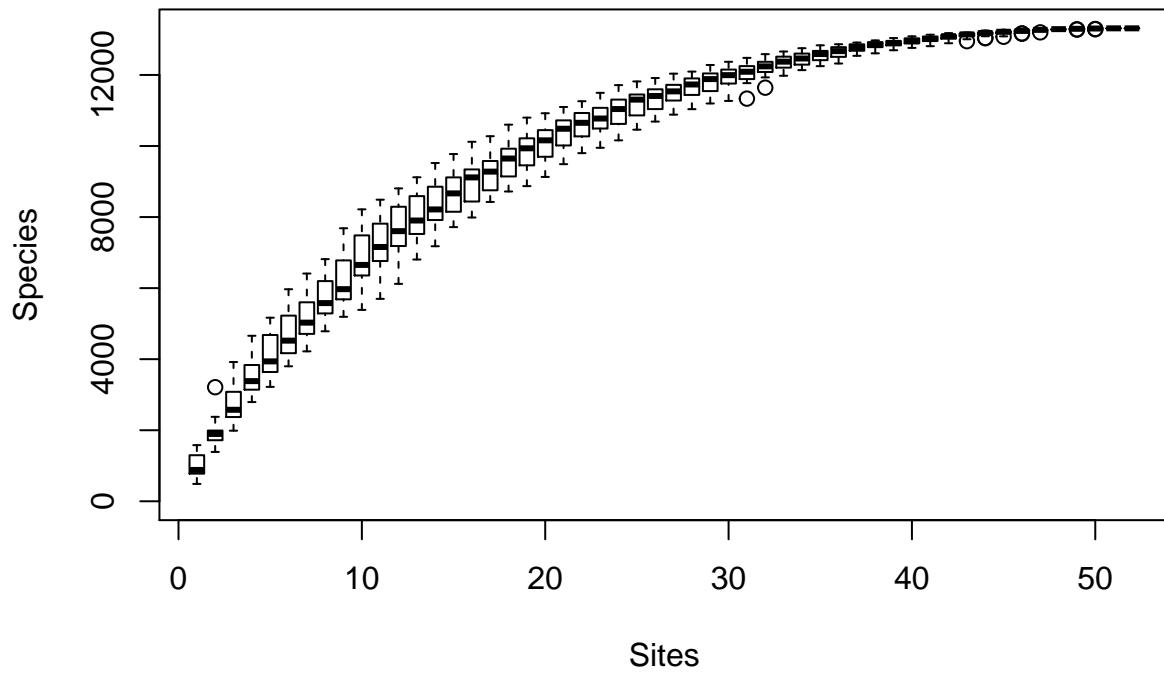
For Dormant

```
x <- specaccum(Dormant, method = "random", permutations = 20, ci.type = "polygon", conditioned = TRUE, ci.lty = 1)
plot(x, add = FALSE, ci = 2, ci.type = "polygon", col = "magenta", ci.col = col, ci.lty = 1, xlab = "Si"
## Warning in polygon(c(xaxvar, rev(xaxvar)), c(x$richness - ci * x$sd,
## rev(x$richness + : supplied color is neither numeric nor character
```

## Dormant gamma–diversity



```
boxplot(x, add = FALSE)
```



## 7.) Macroecological Patterns and Relationships

### Species-abundance distributions

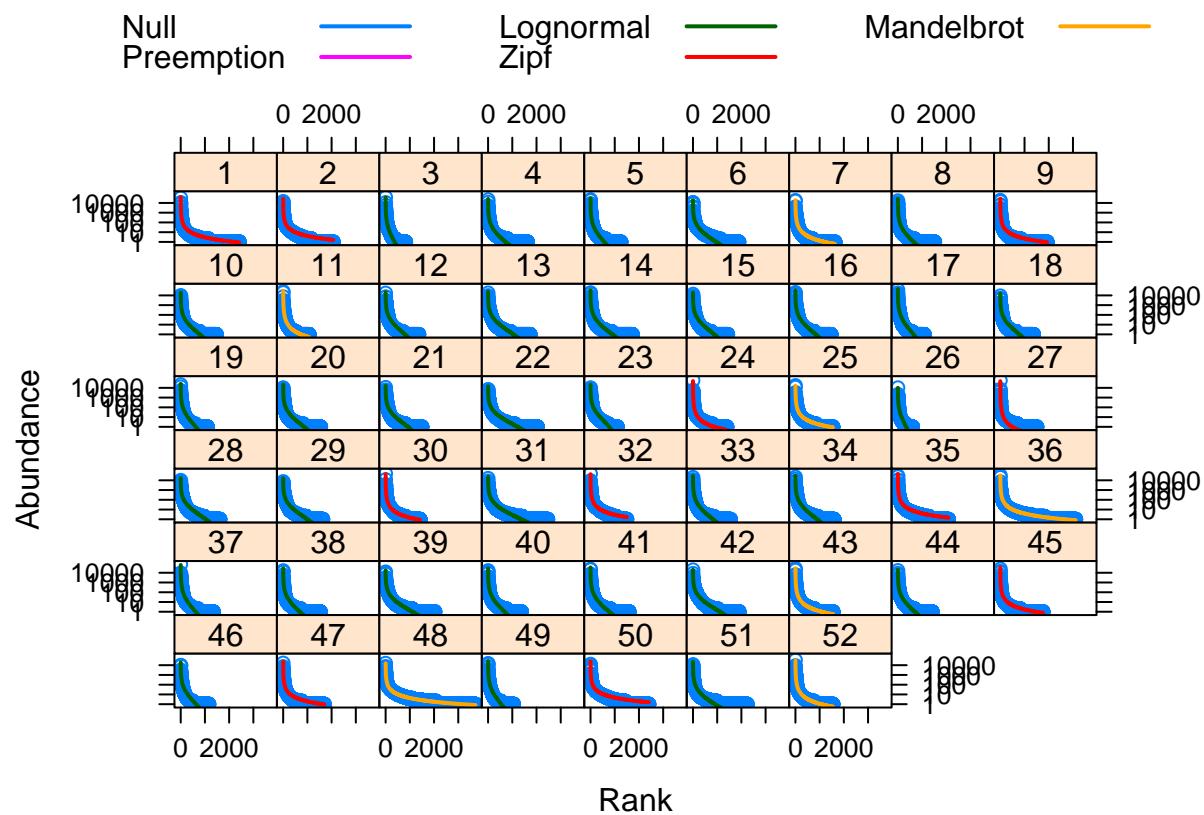
Using function in vegan

```
Active.RACresults <- radfit(Active)

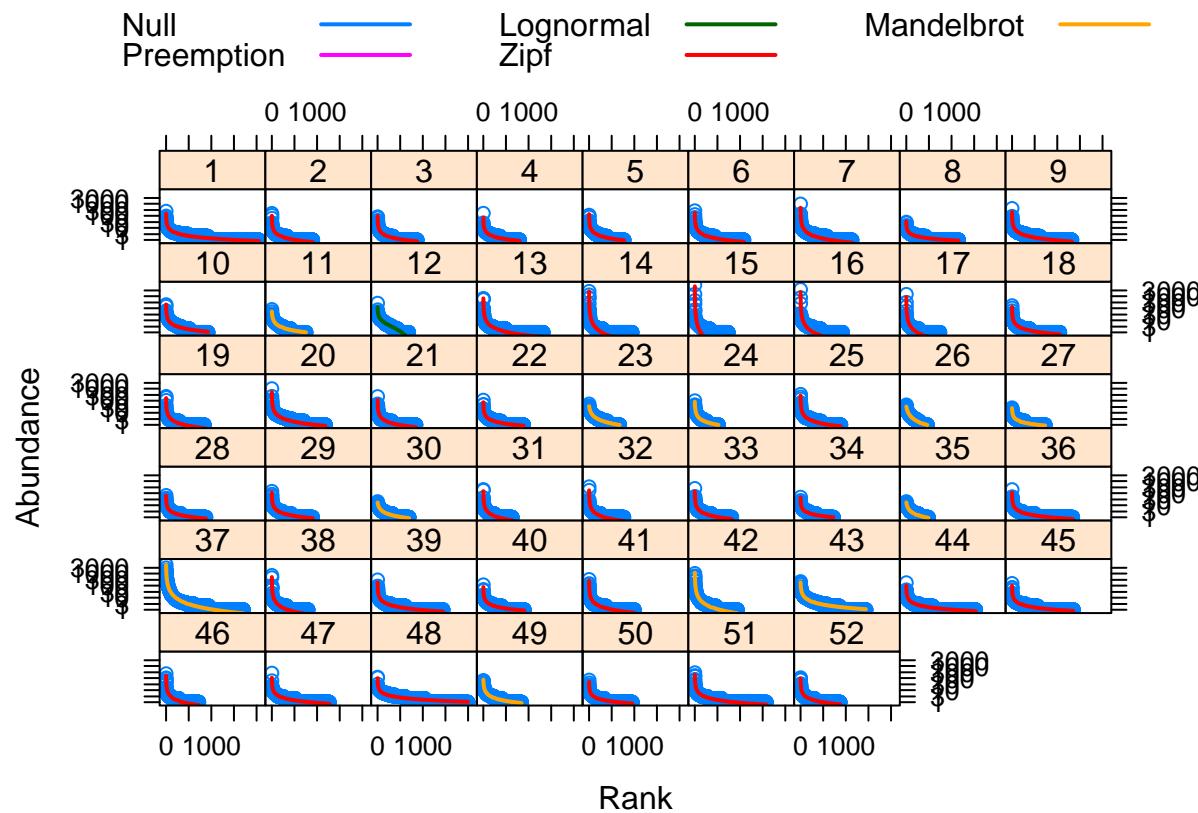
## Warning: glm.fit: fitted rates numerically 0 occurred

## Warning: glm.fit: algorithm did not converge

#Active.RACresults
plot(Active.RACresults)
```



```
Dormant.RACResults <- radfit(Dormant)
#Dormant.RACresults
plot(Dormant.RACResults, las=1, cex.lab = 1.4, cex.axis = 1.25)
```



Species-area relationships

Spatial-abundance distributions

Rarity, Dominance, and Evenness vs. N