

Analysis_birdsdata

Cheyenne, Kathina & Lina

4/7/2021

This document have been created to include all relevant parts in the analysis of our animal diversity project.

##Data preparation and data observation

The first code's chunk will be used to load all libraries needed to proceed.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5      v purrr   0.3.4
## v tibble  3.1.2      v dplyr  1.0.7
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   2.0.1      v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 4.0.4
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.5-7
```

```
library(nlme)
```

```
## Warning: package 'nlme' was built under R version 4.0.5
```

```
##
```

```
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## collapse
```

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.0.5
```

```
##
```

```
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## select
```

```
library(MuMIn)
```

```
## Warning: package 'MuMIn' was built under R version 4.0.5
```

In this second chunk, the data to be used will be loaded into a data frame and transferred into adequate data types.

```
dat <- read.csv("data/birds_dataset.csv", sep=";")
parks <- read.csv("data/parks.csv", sep=";")
forest <- read.csv("data/forest.csv", sep=";")
dat$category <- as.factor(dat$category)
dat$site <- as.factor(dat$site)
```

Following, species richness, species abundance and rarefied richness of species will be calculated.

```
dat$species_richness <- specnumber(dat[,4:31]) #species richness
dat$species_abund <- rowSums(dat[,4:31]) #abundances
dat$rarefied_richness <- rarefy(dat[,4:31],min(dat$species_abund)) #rarefied richness based on the subs

dat$species_richnessparks <- specnumber(parks[,4:31]) #species richness
dat$species_richnessforest <- specnumber(forest[,4:31]) #species richness
```

subsets for forest and park + median, mean, min max...

```
forest <- subset(dat, category=="forest")
park <- subset(dat, category=="park")
median(forest$species_abund)
```

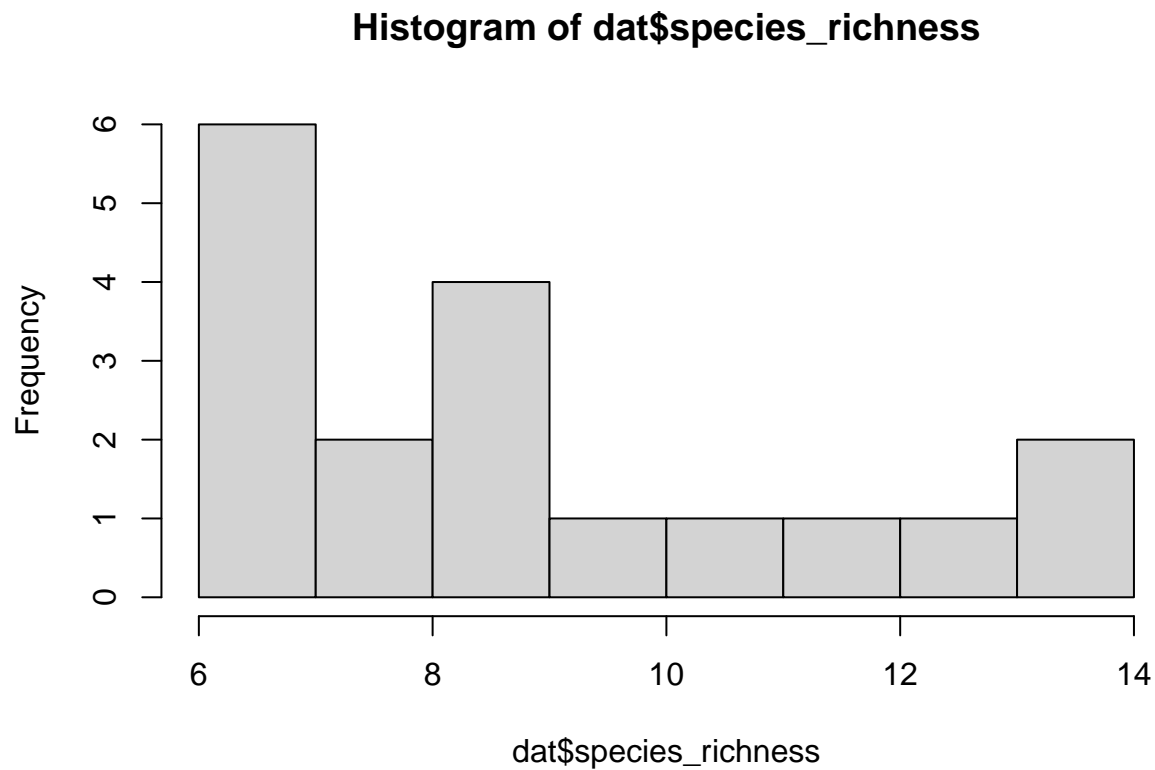
```
## [1] 16
```

```
median(park$species_abund)
```

```
## [1] 24
```

After, data is observed comparing the plots from parks and forests.

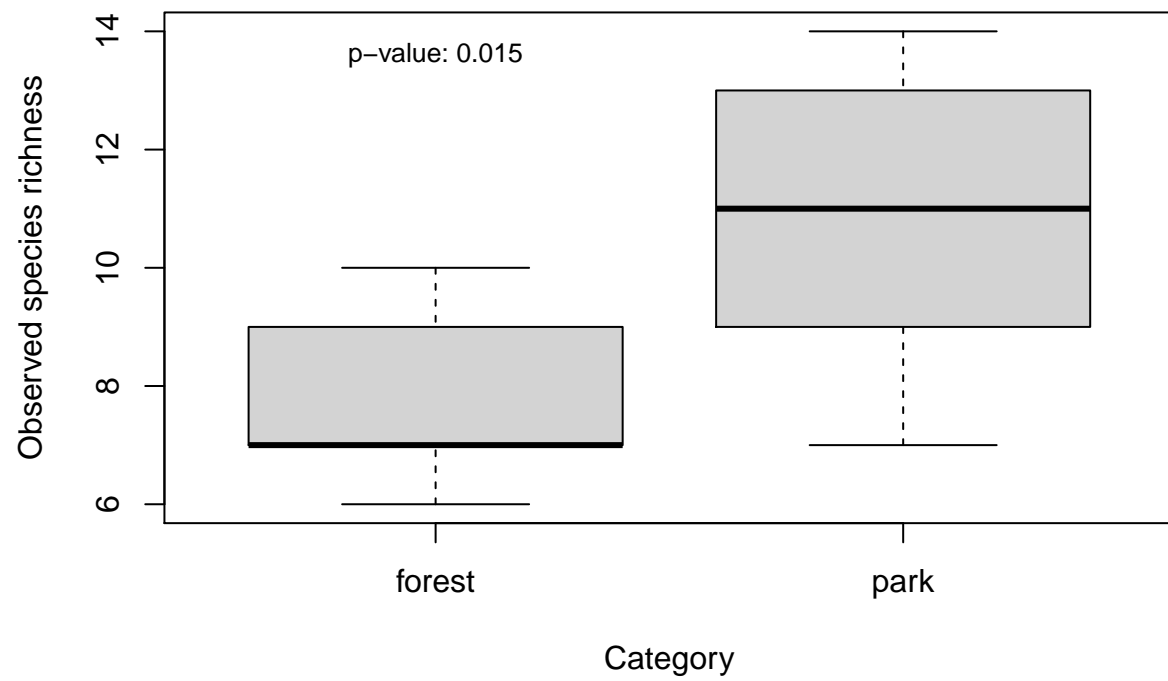
```
hist(dat$species_richness) # not normally distributed
```



```
kruskal.test(dat$species_richness, dat$category) #p-value = 0.01533
```

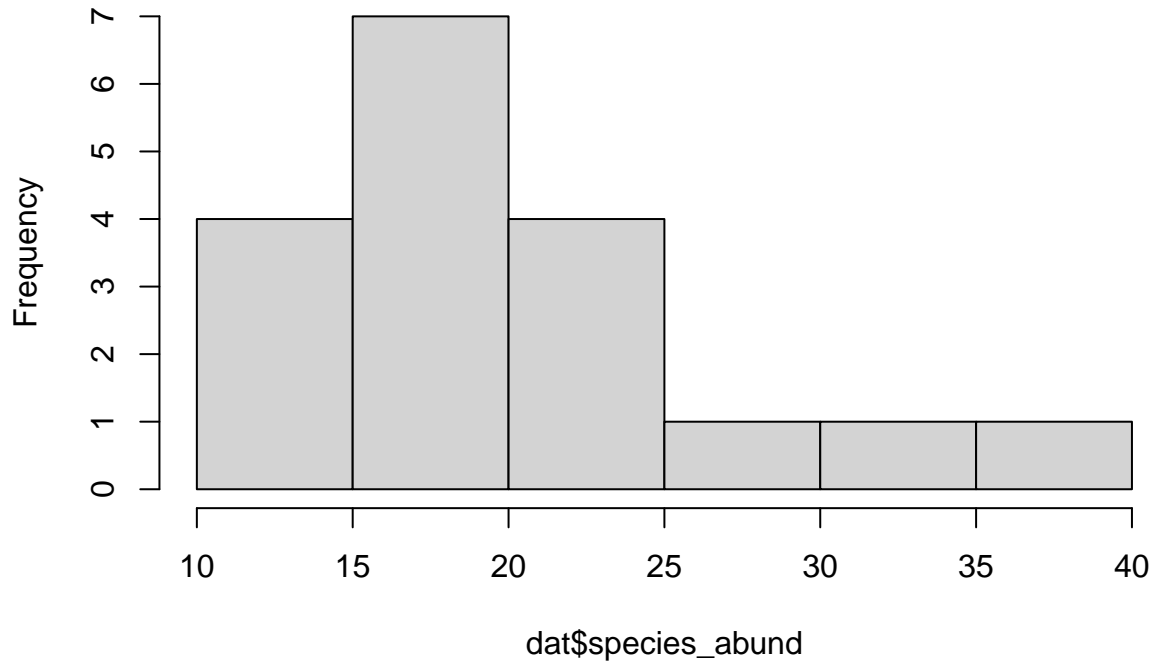
```
##  
## Kruskal-Wallis rank sum test  
##  
## data: dat$species_richness and dat$category  
## Kruskal-Wallis chi-squared = 5.8783, df = 1, p-value = 0.01533
```

```
boxplot(species_richness~category, data=dat, xlab= "Category", ylab= "Observed species richness")  
text(1,13.6,labels = "p-value: 0.015", cex = 0.8)
```



```
hist(dat$species_abund) # not normally distributed
```

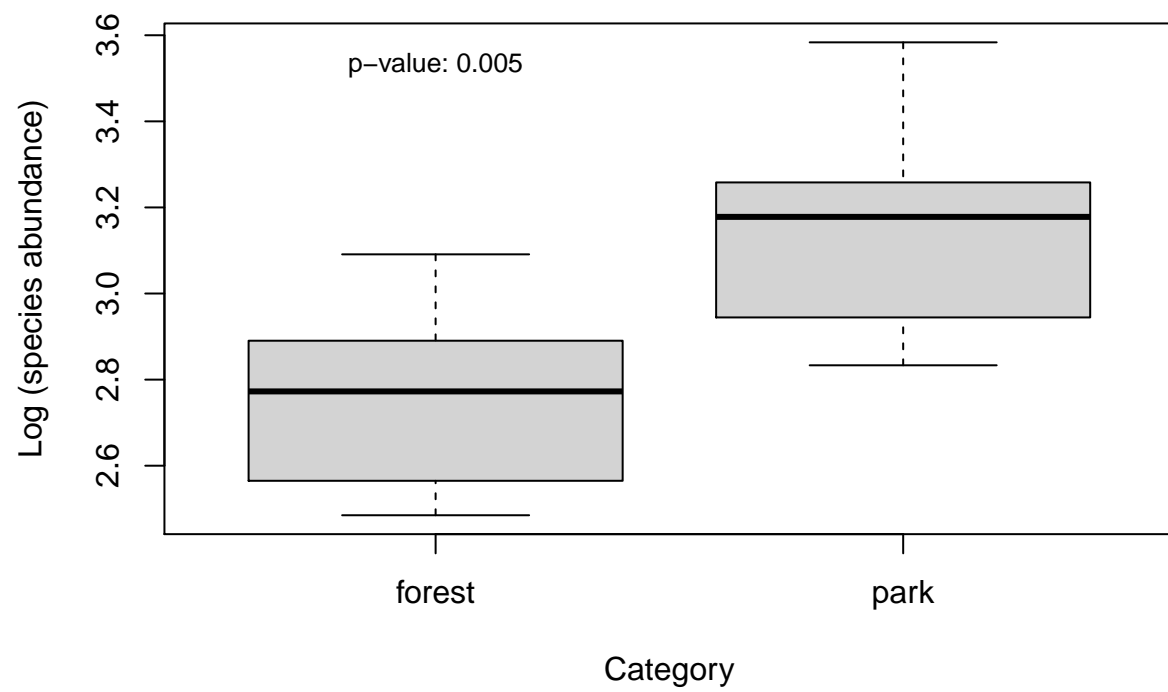
Histogram of dat\$species_abund



```
kruskal.test(dat$species_abund, dat$category) #p-value = 0.0046
```

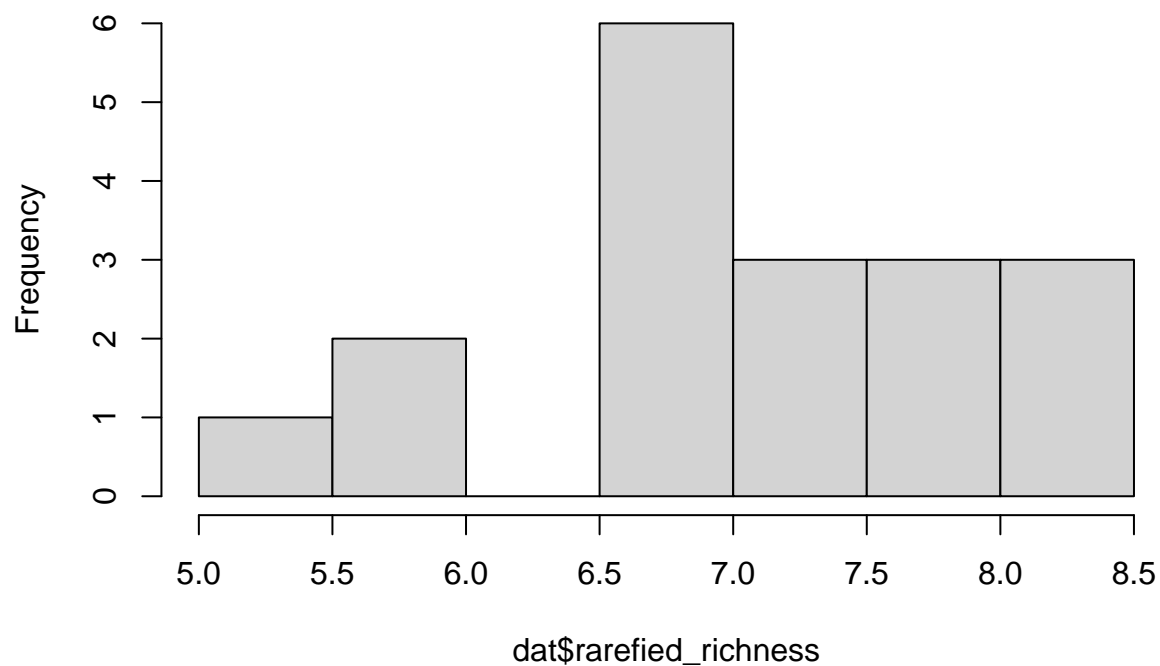
```
##  
## Kruskal-Wallis rank sum test  
##  
## data: dat$species_abund and dat$category  
## Kruskal-Wallis chi-squared = 8.0175, df = 1, p-value = 0.004633
```

```
boxplot(log(species_abund)~category, data=dat, xlab= "Category", ylab= "Log (species abundance)")  
text(1,3.53,labels = "p-value: 0.005", cex = 0.8)
```



```
hist(dat$rarefied_richness) # normally distributed
```

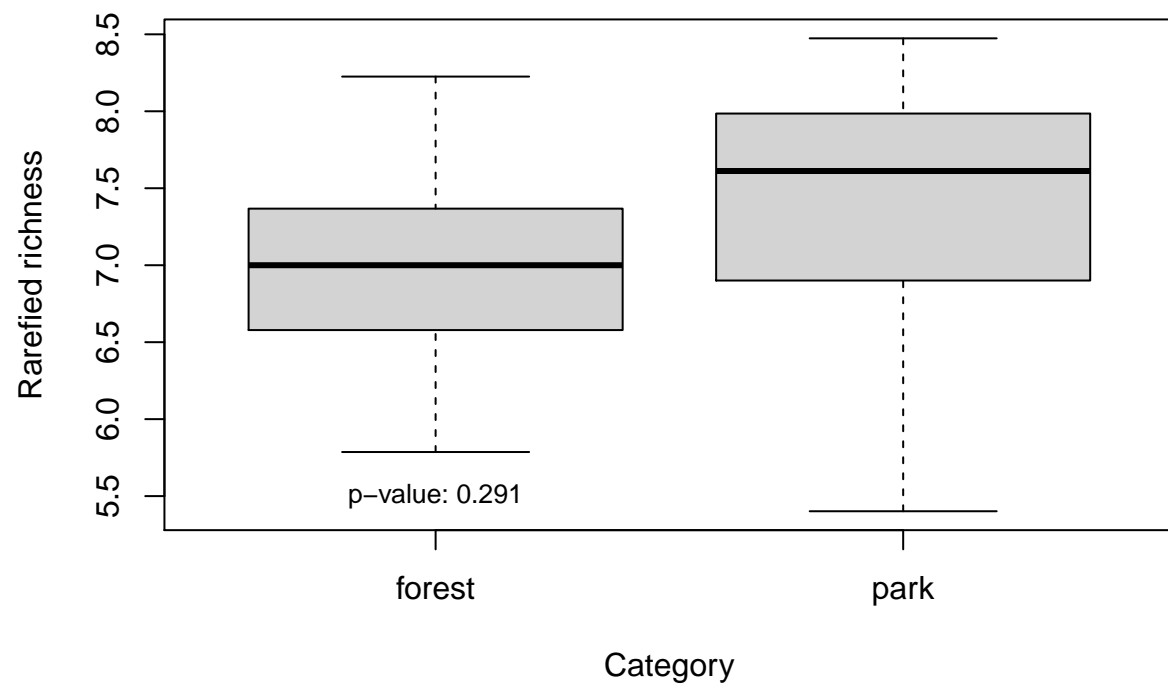
Histogram of dat\$rarefied_richness



```
mod<-aov(dat$rarefied_richness ~ dat$category)
summary(mod) #p-value: 0.291
```

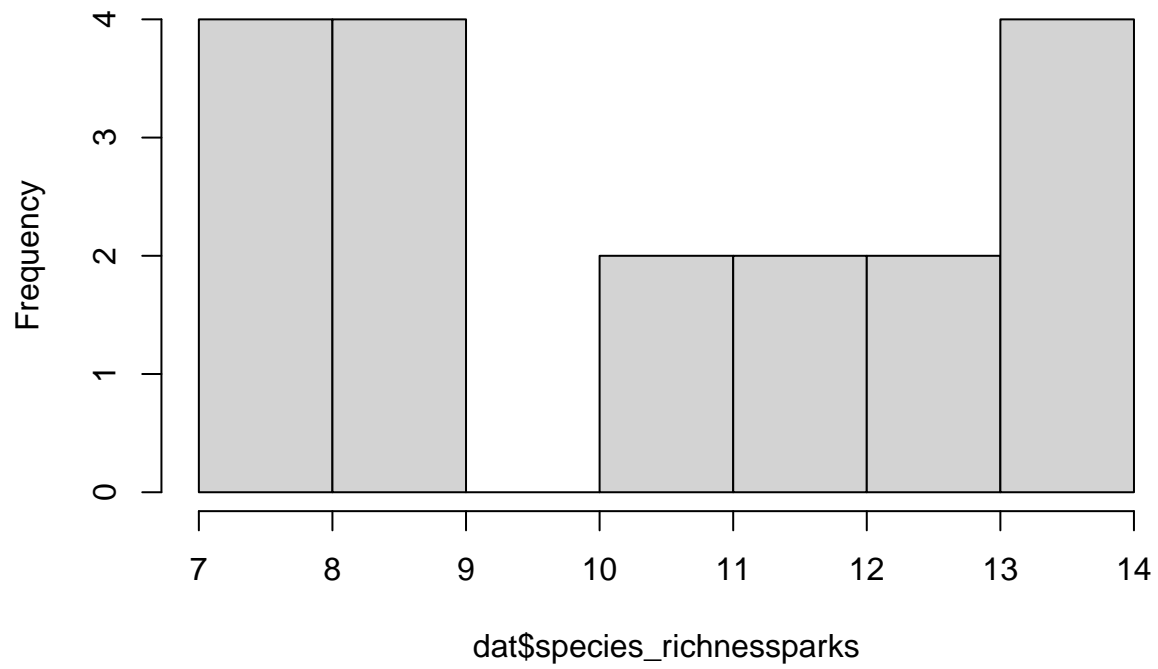
```
##           Df Sum Sq Mean Sq F value Pr(>F)
## dat$category  1  0.941  0.9405  1.194  0.291
## Residuals   16 12.605  0.7878
```

```
boxplot(rarefied_richness~category, data=dat, xlab= "Category", ylab= "Rarefied richness")
text(1,5.5,labels = "p-value: 0.291", cex = 0.8)
```



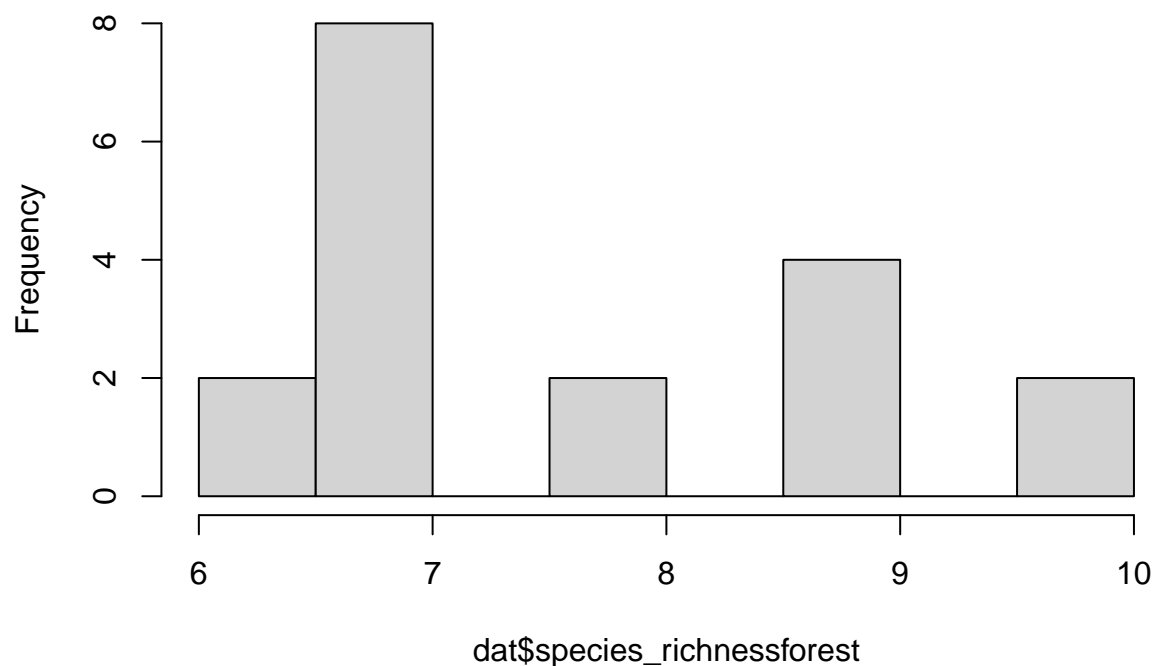
```
hist(dat$species_richnessparks)
```


Histogram of dat\$species_richnessparks



```
hist(dat$species_richnessforest)
```

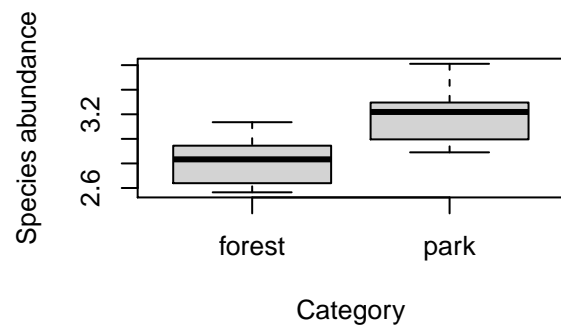
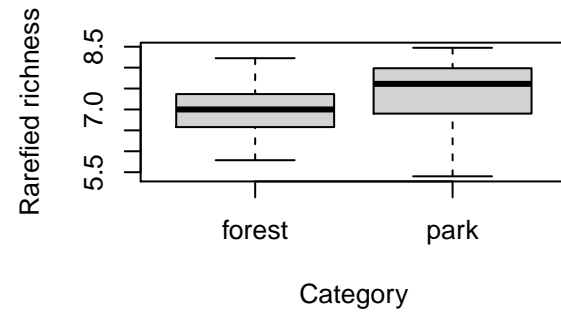
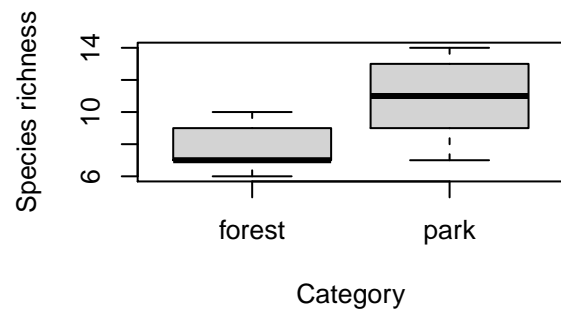
Histogram of dat\$species_richnessforest



```
layout(matrix(c(1:4), nrow=2, byrow=FALSE))
boxplot(species_richness~category, data=dat, xlab= "Category", ylab= "Species richness")
boxplot(log(species_abund+1)~category, data=dat, xlab= "Category", ylab= "Species abundance")
?boxplot
```

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

```
boxplot(rarefied_richness~category, data=dat, xlab= "Category", ylab= "Rarefied richness")
layout(matrix(c(1:4), nrow=2, byrow=FALSE))
```

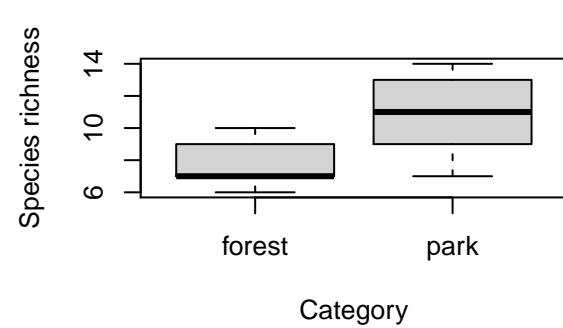
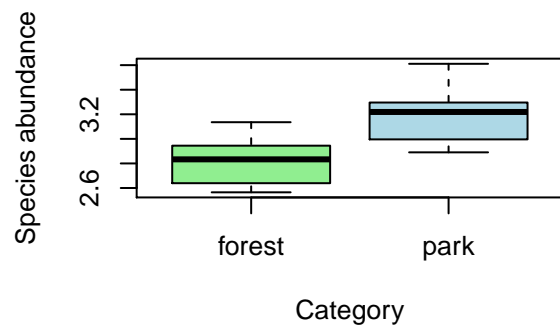
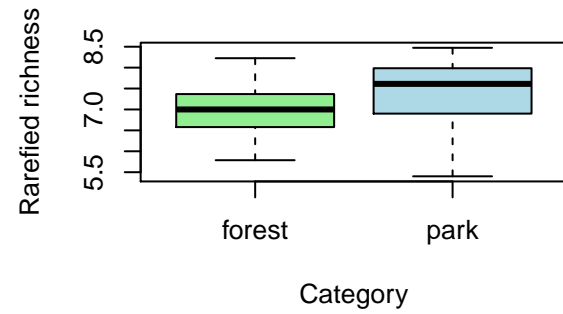
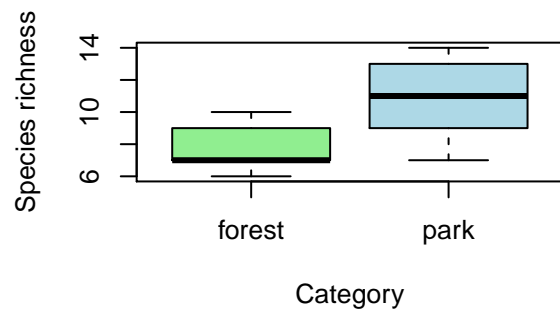


```

boxplot(species_richness~category, data=dat, xlab= "Category", ylab= "Species richness", col= c("lightgreen", "lightblue"))
boxplot(log(species_abund+1)~category, data=dat, xlab= "Category", ylab= "Species abundance", col= c("lightgreen", "lightblue"))
?boxplot
boxplot(rarefied_richness~category, data=dat, xlab= "Category", ylab= "Rarefied richness", col= c("lightgreen", "lightblue"))

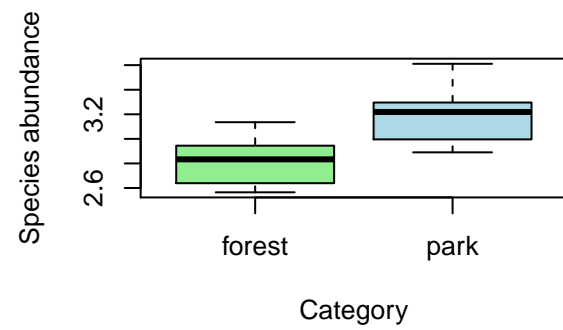
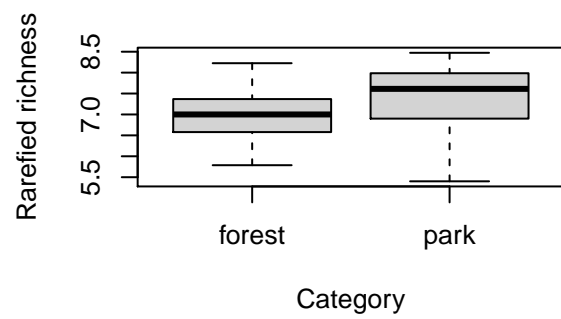
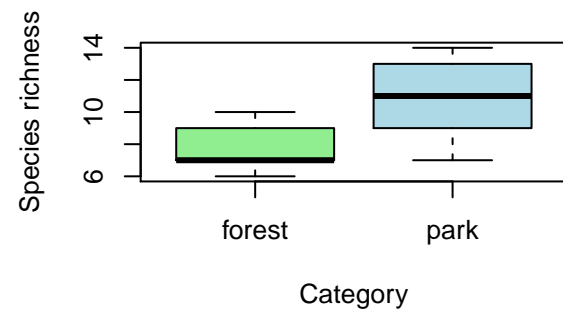
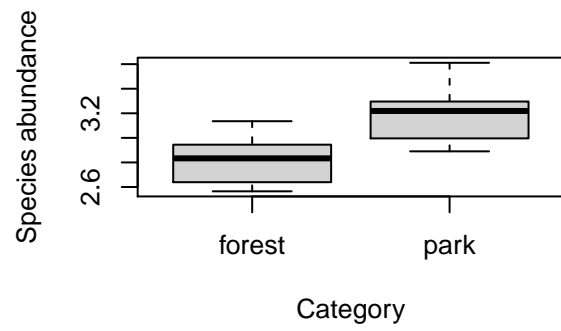
boxplot(species_richness~category, data=dat, xlab= "Category", ylab= "Species richness")

```

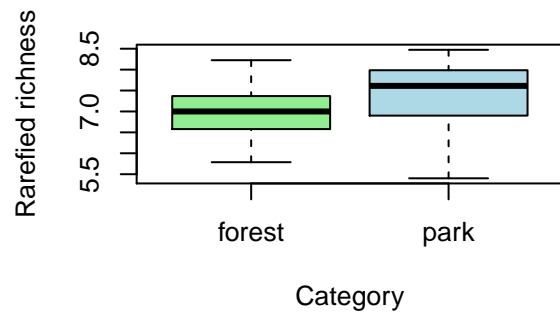


```
boxplot(log(species_abund+1)~category, data=dat, xlab= "Category", ylab= "Species abundance")
?boxplot
boxplot(rarefied_richness~category, data=dat, xlab= "Category", ylab= "Rarefied richness")

boxplot(species_richness~category, data=dat, xlab= "Category", ylab= "Species richness", col= c("lightgreen", "lightblue"))
boxplot(log(species_abund+1)~category, data=dat, xlab= "Category", ylab= "Species abundance", col= c("lightgreen", "lightblue"))
```



```
?boxplot
boxplot(rarefied_richness~category, data=dat, xlab= "Category", ylab= "Rarefied richness", col= c("lightgreen", "lightblue"))
```



Now let's begin with the data analysis

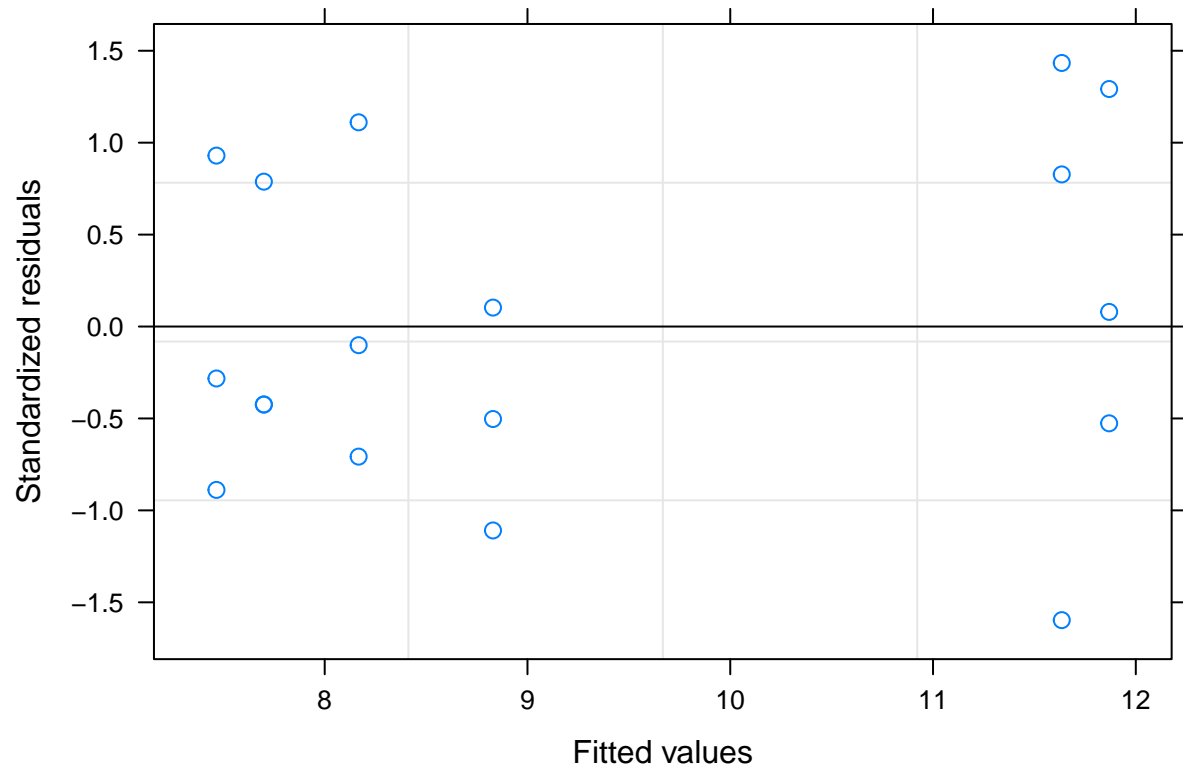
#2a linear models: to check for difference between habitat types with the implementation of mixed effect models.

```
mod1 <- lme(species_richness~category, random = (~1|site), data=dat) #model structure, random=... speci
summary(mod1) #model output - important is the "fixed effects" part. Here "forest" is hiding in the "In
```

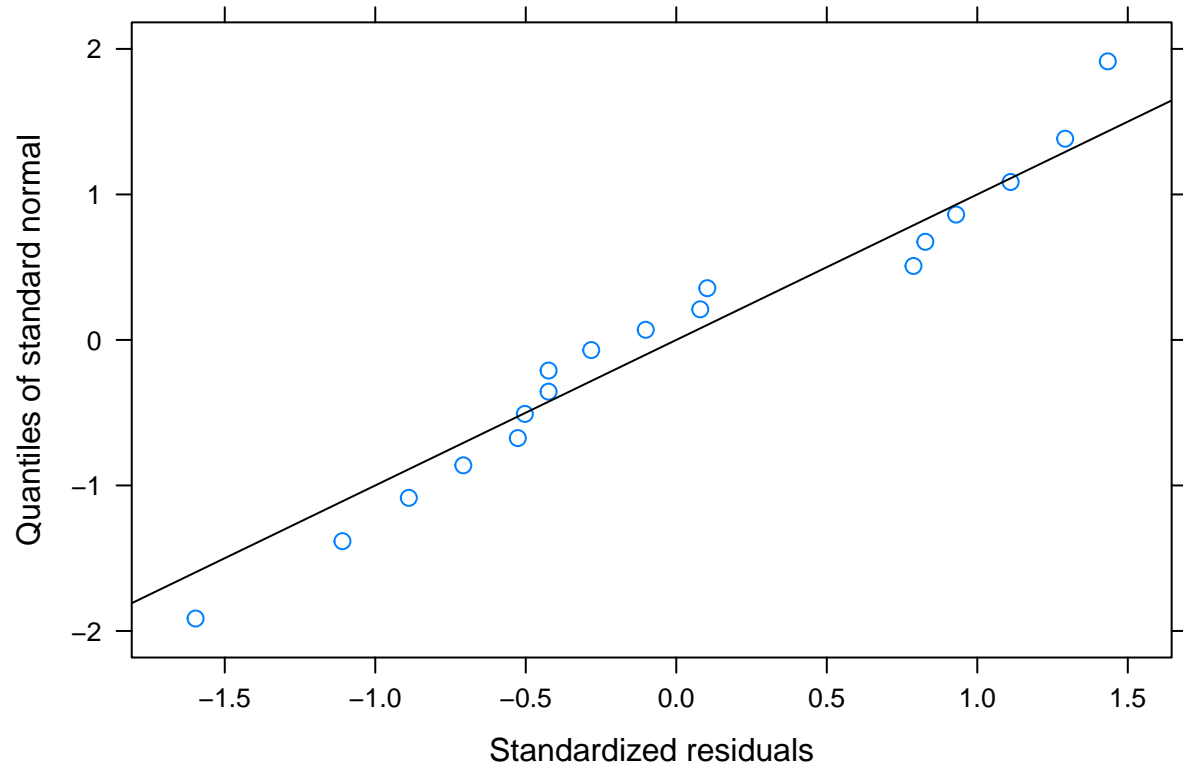
```
## Linear mixed-effects model fit by REML
##   Data: dat
##       AIC      BIC    logLik
##  78.65584 81.7462 -35.32792
##
## Random effects:
## Formula: ~1 | site
##      (Intercept) Residual
## StdDev:      1.459325 1.649916
##
## Fixed effects: species_richness ~ category
##              Value Std.Error DF   t-value p-value
## (Intercept)  7.777778  1.006154 12  7.730207  0.0000
## categorypark  3.000000  1.422916  4  2.108346  0.1027
## Correlation:
##              (Intr)
## categorypark -0.707
##
```

```
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.5969444 -0.5205776 -0.1919700  0.8175666  1.4335132
##
## Number of Observations: 18
## Number of Groups: 6
```

```
plot(mod1) #check for homogeneity of variances (data points should have similar vertical spread along t
```



```
qqnorm(mod1, ~resid(.,type="p"), abline = c(0,1)) #check for normality of residuals (should not be comp
```



#2b linear models to include environmental variables and see the variation of the data with each of them.

MIXED EFFECTS MODEL WITH SPECIES RICHNESS

`round(cor(dat[,32:45]),2)` *#check which predictor variables are strongly correlated (below -0.7 or above*

```
##          canopy_cover n_tree_spec n_tree_ind dbh_min dbh_min5 dbh_mean
## canopy_cover          1.00      -0.68       0.22  -0.43  -0.38  -0.42
## n_tree_spec          -0.68       1.00      -0.42   0.30   0.18   0.31
## n_tree_ind           0.22      -0.42       1.00  -0.12  -0.51  -0.49
## dbh_min            -0.43       0.30      -0.12   1.00   0.37   0.68
## dbh_min5           -0.38       0.18      -0.51   0.37   1.00   0.71
## dbh_mean           -0.42       0.31      -0.49   0.68   0.71   1.00
## dbh_max            -0.33       0.23      -0.52   0.26   0.57   0.83
## dbh_median         -0.29       0.29      -0.56   0.30   0.89   0.74
## dbh_sd             -0.69       0.78      -0.63   0.09   0.39   0.36
## n_microhabitats     0.30      -0.42      -0.12  -0.06   0.44   0.27
## latitude            0.30      -0.60       0.22  -0.23   0.04  -0.08
## longitude           0.71      -0.64       0.32  -0.57  -0.50  -0.42
## size                0.64      -0.52       0.35  -0.33  -0.55  -0.52
## temperature        -0.02       0.03       0.44  -0.29  -0.49  -0.26
##          dbh_max dbh_median dbh_sd n_microhabitats latitude longitude
## canopy_cover   -0.33   -0.29  -0.69           0.30    0.30    0.71
## n_tree_spec     0.23    0.29   0.78          -0.42   -0.60   -0.64
## n_tree_ind     -0.52   -0.56  -0.63          -0.12    0.22    0.32
## dbh_min         0.26    0.30   0.09          -0.06   -0.23   -0.57
```


## dbh_min5	0.57	0.89	0.39	0.44	0.04	-0.50
## dbh_mean	0.83	0.74	0.36	0.27	-0.08	-0.42
## dbh_max	1.00	0.57	0.47	0.14	0.03	-0.19
## dbh_median	0.57	1.00	0.44	0.49	-0.06	-0.39
## dbh_sd	0.47	0.44	1.00	-0.32	-0.40	-0.58
## n_microhabitats	0.14	0.49	-0.32	1.00	0.30	0.21
## latitude	0.03	-0.06	-0.40	0.30	1.00	0.14
## longitude	-0.19	-0.39	-0.58	0.21	0.14	1.00
## size	-0.52	-0.48	-0.61	0.22	-0.09	0.80
## temperature	-0.06	-0.38	-0.02	-0.45	0.05	0.23
##	size temperature					
## canopy_cover	0.64	-0.02				
## n_tree_spec	-0.52	0.03				
## n_tree_ind	0.35	0.44				
## dbh_min	-0.33	-0.29				
## dbh_min5	-0.55	-0.49				
## dbh_mean	-0.52	-0.26				
## dbh_max	-0.52	-0.06				
## dbh_median	-0.48	-0.38				
## dbh_sd	-0.61	-0.02				
## n_microhabitats	0.22	-0.45				
## latitude	-0.09	0.05				
## longitude	0.80	0.23				
## size	1.00	-0.07				
## temperature	-0.07	1.00				

```
mod1 <- lme(species_richness ~ category*size + canopy_cover + n_tree_spec + n_tree_ind + log(dbh_min) +
summary(mod1)
```

```
## Linear mixed-effects model fit by maximum likelihood
```

```
## Data: dat
```

```
## AIC BIC logLik
```

```
## 79.34161 90.91644 -26.6708
```

```
##
```

```
## Random effects:
```

```
## Formula: ~1 | site
```

```
## (Intercept) Residual
```

```
## StdDev: 4.101158e-05 1.064785
```

```
##
```

```
## Fixed effects: species_richness ~ category * size + canopy_cover + n_tree_spec + n_tree_ind +
```

```
## Value Std.Error DF t-value p-value
```

```
## (Intercept) 13.449991 5.833096 6 2.3058067 0.0606
```

```
## categorypark 0.925977 3.442663 1 0.2689712 0.8327
```

```
## size -0.004229 0.011728 1 -0.3606218 0.7797
```

```
## canopy_cover -0.042049 0.027271 6 -1.5418946 0.1740
```

```
## n_tree_spec -0.372451 0.390288 6 -0.9542969 0.3768
```

```
## n_tree_ind -0.015531 0.161447 6 -0.0961977 0.9265
```

```
## log(dbh_min) 1.622178 0.851637 6 1.9047752 0.1055
```

```
## dbh_min5 -0.051502 0.070771 6 -0.7277286 0.4942
```

```
## n_microhabitats 0.029756 0.114604 6 0.2596375 0.8038
```

```
## temperature -0.201717 0.171316 1 -1.1774589 0.4482
```

```
## categorypark:size 0.004324 0.062229 1 0.0694820 0.9558
```

```
## Correlation:
```

```
## (Intr) ctgryp size cnpy_c n_tr_s n_tr_n lg(d_) dbh_m5
```

```

## categorypark      -0.139
## size              -0.306  0.593
## canopy_cover      -0.552  0.312  0.025
## n_tree_spec       -0.447 -0.589 -0.291  0.096
## n_tree_ind        -0.481  0.067 -0.027  0.134  0.430
## log(dbh_min)      -0.401 -0.217 -0.125  0.033  0.261  0.049
## dbh_min5          -0.169 -0.578  0.013 -0.050  0.412  0.135 -0.207
## n_microhabitats   -0.099  0.425  0.029  0.093 -0.257 -0.176  0.020 -0.663
## temperature       -0.335  0.142  0.333  0.151 -0.282 -0.499  0.074  0.045
## categorypark:size  0.200  0.045  0.019  0.163 -0.424 -0.503 -0.438  0.032
##                  n_mcrh tmprtr
## categorypark
## size
## canopy_cover
## n_tree_spec
## n_tree_ind
## log(dbh_min)
## dbh_min5
## n_microhabitats
## temperature      0.328
## categorypark:size 0.299  0.468
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.4774299 -0.8641612 -0.1156606  0.4119055  2.3217607
##
## Number of Observations: 18
## Number of Groups: 6

```

```
mod1.1 <- stepAIC(mod1) #model simplification based on AIC-value of the model
```

```

## Start:  AIC=79.34
## species_richness ~ category * size + canopy_cover + n_tree_spec +
##      n_tree_ind + log(dbh_min) + dbh_min5 + n_microhabitats +
##      temperature
##
##              Df    AIC
## - category:size    1 77.354
## - n_tree_ind       1 77.365
## - n_microhabitats  1 77.514
## - dbh_min5         1 78.654
## <none>              79.342
## - n_tree_spec      1 79.543
## - temperature      1 80.594
## - canopy_cover     1 82.605
## - log(dbh_min)     1 84.858
##
## Step:  AIC=77.35
## species_richness ~ category + size + canopy_cover + n_tree_spec +
##      n_tree_ind + log(dbh_min) + dbh_min5 + n_microhabitats +
##      temperature
##
##              Df    AIC
## - n_tree_ind      1 75.367

```

```

## - n_microhabitats 1 75.514
## - category        1 75.535
## - size            1 75.688
## - dbh_min5        1 76.675
## <none>            77.354
## - n_tree_spec     1 77.852
## - temperature     1 79.622
## - canopy_cover    1 80.806
## - log(dbh_min)    1 84.496
##
## Step: AIC=75.37
## species_richness ~ category + size + canopy_cover + n_tree_spec +
##   log(dbh_min) + dbh_min5 + n_microhabitats + temperature
##
##           Df    AIC
## - n_microhabitats 1 73.524
## - category        1 73.560
## - size            1 73.703
## - dbh_min5        1 74.683
## <none>            75.367
## - n_tree_spec     1 75.961
## - temperature     1 78.287
## - canopy_cover    1 79.003
## - log(dbh_min)    1 82.760
##
## Step: AIC=73.52
## species_richness ~ category + size + canopy_cover + n_tree_spec +
##   log(dbh_min) + dbh_min5 + temperature
##
##           Df    AIC
## - category        1 71.611
## - size            1 71.869
## - dbh_min5        1 73.033
## <none>            73.524
## - n_tree_spec     1 73.980
## - temperature     1 76.975
## - canopy_cover    1 77.237
## - log(dbh_min)    1 80.787
##
## Step: AIC=71.61
## species_richness ~ size + canopy_cover + n_tree_spec + log(dbh_min) +
##   dbh_min5 + temperature
##
##           Df    AIC
## - size            1 70.645
## - dbh_min5        1 71.139
## <none>            71.611
## - n_tree_spec     1 72.982
## - temperature     1 75.195
## - canopy_cover    1 76.107
## - log(dbh_min)    1 80.025
##
## Step: AIC=70.65
## species_richness ~ canopy_cover + n_tree_spec + log(dbh_min) +

```

```
##      dbh_min5 + temperature
##
##              Df      AIC
## - dbh_min5      1 69.299
## <none>           70.645
## - n_tree_spec   1 71.260
## - temperature   1 73.198
## - canopy_cover   1 76.119
## - log(dbh_min)   1 79.032
##
## Step: AIC=69.3
## species_richness ~ canopy_cover + n_tree_spec + log(dbh_min) +
##      temperature
##
##              Df      AIC
## <none>           69.299
## - n_tree_spec   1 69.688
## - temperature   1 71.200
## - canopy_cover   1 74.131
## - log(dbh_min)   1 77.308
```

```
summary(mod1.1) #final model which includes only the most important predictors
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: dat
##       AIC      BIC    logLik
## 69.29947 75.53207 -27.64974
##
## Random effects:
## Formula: ~1 | site
##      (Intercept) Residual
## StdDev: 2.762129e-05 1.124297
##
## Fixed effects: species_richness ~ canopy_cover + n_tree_spec + log(dbh_min) +      temperature
##              Value Std.Error DF   t-value p-value
## (Intercept) 11.366118  3.314519  9  3.429190  0.0075
## canopy_cover -0.044574  0.018196  9 -2.449668  0.0368
## n_tree_spec  -0.241219  0.177235  9 -1.361014  0.2066
## log(dbh_min)  1.542315  0.496008  9  3.109455  0.0125
## temperature -0.160682  0.090601  4 -1.773506  0.1508
## Correlation:
##      (Intr) cnpy_c n_tr_s lg(d_)
## canopy_cover -0.791
## n_tree_spec  -0.495  0.595
## log(dbh_min) -0.759  0.467 -0.013
## temperature -0.681  0.241 -0.024  0.519
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.67276372 -0.80102592 -0.08660587  0.68432847  2.15093491
##
## Number of Observations: 18
## Number of Groups: 6
```

```
mod1.2 <- update(mod1.1, ~.-n_tree_spec)
summary(mod1.2)
```

```
## Linear mixed-effects model fit by maximum likelihood
## Data: dat
##      AIC      BIC    logLik
## 69.68834 75.03057 -28.84417
##
## Random effects:
## Formula: ~1 | site
##      (Intercept) Residual
## StdDev:    0.1799486 1.188182
##
## Fixed effects: species_richness ~ canopy_cover + log(dbh_min) + temperature
##              Value Std.Error DF   t-value p-value
## (Intercept)  9.140053 2.9508347 10   3.097447  0.0113
## canopy_cover -0.029890 0.0150141 10  -1.990770  0.0745
## log(dbh_min)  1.532576 0.5067240 10   3.024479  0.0128
## temperature -0.163764 0.0945977  4  -1.731167  0.1585
## Correlation:
##      (Intr) cnpy_c lg(d_)
## canopy_cover -0.700
## log(dbh_min) -0.875  0.574
## temperature -0.797  0.303  0.507
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.47877950 -0.81708368  0.01400286  0.42067808  2.51962252
##
## Number of Observations: 18
## Number of Groups: 6
```

```
anova(mod1.1, mod1.2)
```

```
##      Model df      AIC      BIC    logLik  Test L.Ratio p-value
## mod1.1    1  7 69.29947 75.53207 -27.64973
## mod1.2    2  6 69.68834 75.03057 -28.84417 1 vs 2 2.38887 0.1222
```

```
mod1.3 <- update(mod1.2, ~.-temperature)
summary(mod1.3)
```

```
## Linear mixed-effects model fit by maximum likelihood
## Data: dat
##      AIC      BIC    logLik
## 70.90548 75.35733 -30.45274
##
## Random effects:
## Formula: ~1 | site
##      (Intercept) Residual
## StdDev:    0.550221 1.212528
##
## Fixed effects: species_richness ~ canopy_cover + log(dbh_min)
```

```
##               Value Std.Error DF   t-value p-value
## (Intercept)    5.562682 1.8467347 10   3.012172  0.0131
## canopy_cover -0.025139 0.0152642 10  -1.646896  0.1306
## log(dbh_min)  1.862918 0.4588731 10   4.059768  0.0023
## Correlation:
##               (Intr) cnpy_c
## canopy_cover -0.756
## log(dbh_min) -0.882  0.427
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.6480211 -0.6031499 -0.2136438  0.7787394  1.8252090
##
## Number of Observations: 18
## Number of Groups: 6
```

```
anova(mod1.2, mod1.3) #close to signifcance effect with these variables (species increasing with incr
```

```
##           Model df          AIC          BIC    logLik    Test  L.Ratio p-value
## mod1.2         1  6 69.68834 75.03057 -28.84417
## mod1.3         2  5 70.90548 75.35733 -30.45274 1 vs 2 3.217136  0.0729
```

```
mod1.4 <- update(mod1.3, ~.-canopy_cover)
summary(mod1.4)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: dat
##           AIC          BIC    logLik
##   71.50383 75.06532 -31.75191
##
## Random effects:
## Formula: ~1 | site
##           (Intercept) Residual
## StdDev: 0.0001335345  1.41207
##
## Fixed effects: species_richness ~ log(dbh_min)
##               Value Std.Error DF   t-value p-value
## (Intercept)  2.833014 1.1783306 11  2.404261  0.0350
## log(dbh_min) 2.342352 0.4085936 11  5.732719  0.0001
## Correlation:
##               (Intr)
## log(dbh_min) -0.954
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -2.06423425 -0.69965226  0.06165372  0.63232038  1.83834164
##
## Number of Observations: 18
## Number of Groups: 6
```

```
anova(mod1.3,mod1.4)
```

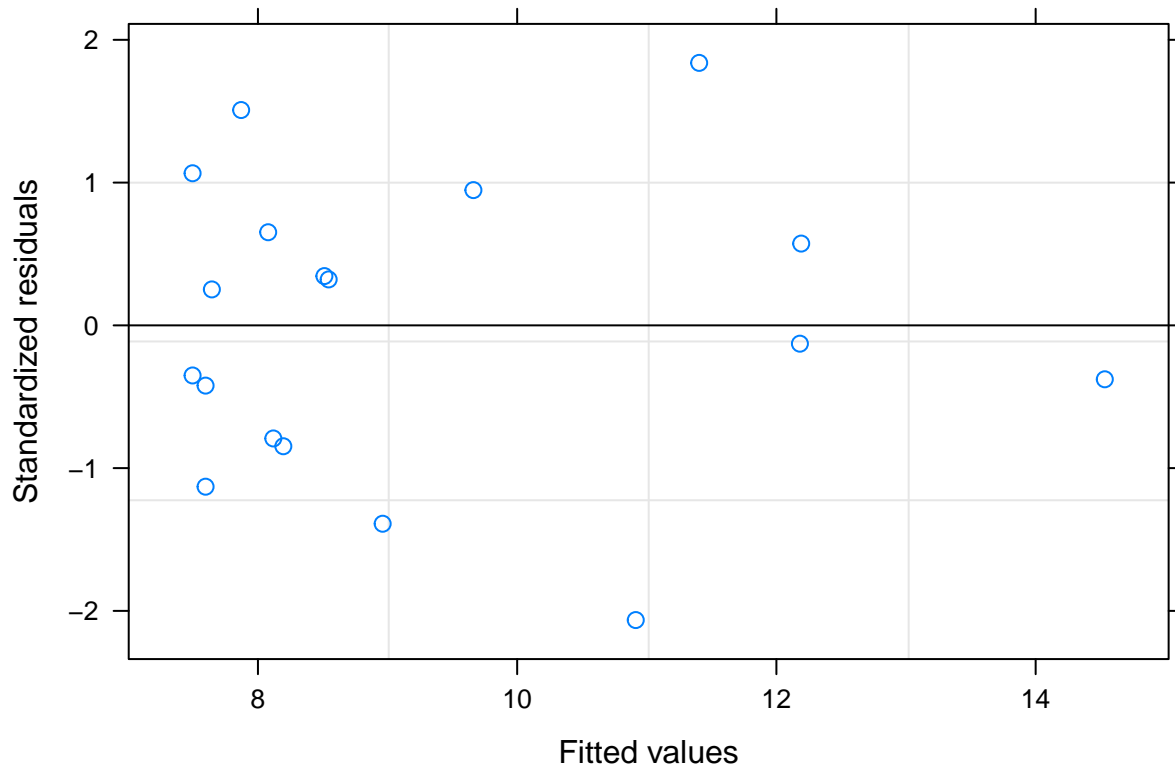
```
##           Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## mod1.3      1   5 70.90548 75.35733 -30.45274
## mod1.4      2   4 71.50383 75.06532 -31.75191 1 vs 2 2.598353 0.107
```

```
#final model:
#lme(species_richness ~ log(dbh_min), random = (~1/site), data=dat, method="ML")
#https://jonlefccheck.net/2013/03/13/r2-for-linear-mixed-effects-models/
r.squaredGLMM(mod1.4)
```

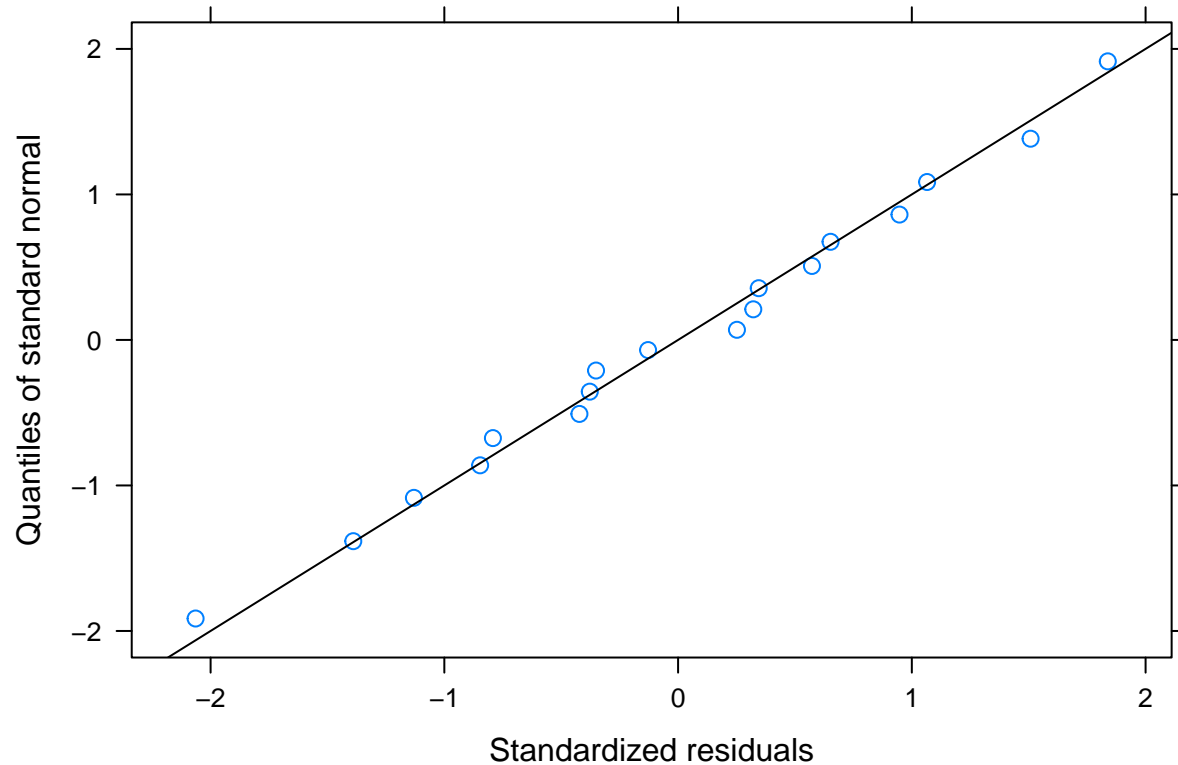
```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.
```

```
##           R2m      R2c
## [1,] 0.6850223 0.6850223
```

```
plot(mod1.4) #check for homogeneity of variances (data points should have similar vertical spread along
```



```
qqnorm(mod1.4, ~resid(.,type="p"), abline=c(0,1)) #check for normality of residuals (should not be comp
```



#standard deviation, coefficient of variation (sd/mean) to make variation independent from out mean val

```
#dat$sddbh_mean <- sd() we can try as alternative
library(car)
```

```
## Warning: package 'car' was built under R version 4.0.5
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      recode
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
##      some
```

```
vif(mod1.2)
```

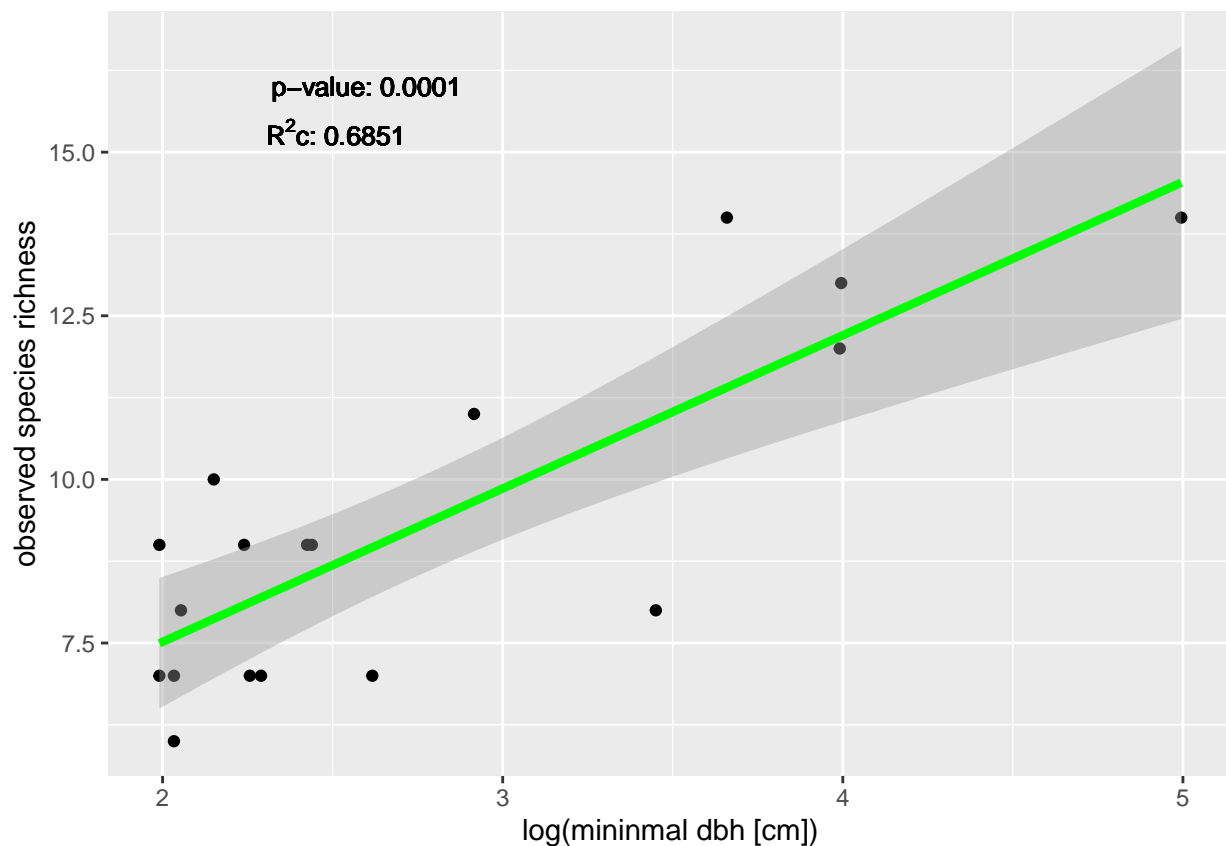
```
## canopy_cover log(dbh_min) temperature
##      1.492654      1.824734      1.346892
```



```
ggplot(dat, aes(x = log(dbh_min), y = species_richness))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=2.6, y=16, label="p-value: 0.0001", size=3.5) +
  geom_text(x=2.51, y=15.3, label=expression(paste("R"2, "c: 0.6851")), size=3.5) +
  labs(x="log(minimal dbh [cm])", y="observed species richness")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

```
## Warning in is.na(x): is.na() aplicado a un objeto que no es (lista o vector) de
## tipo 'expression'
```



```
#2c Linear Mixed Effect models
```

```
### MIXED EFFECTS MODEL WITH BIRD ABUNDANCE
```

```
mod2.1 <- lme(species_abund ~ category*size + canopy_cover + n_tree_spec + n_tree_ind + log(dbh_min) + c
summary(mod2.1)
```

```
## Linear mixed-effects model fit by maximum likelihood
```

```
## Data: dat
```

```
## AIC BIC logLik
```

```
## 110.9826 122.5575 -42.49131
```

```
##
```

```
## Random effects:
```

```

## Formula: ~1 | site
## (Intercept) Residual
## StdDev: 5.575945e-05 2.564311
##
## Fixed effects: species_abund ~ category * size + canopy_cover + n_tree_spec + n_tree_ind + log
## Value Std.Error DF t-value p-value
## (Intercept) 18.087181 17.052295 6 1.0606890 0.3296
## categorypark 1.639039 8.730317 1 0.1877410 0.8819
## size -0.018921 0.028943 1 -0.6537125 0.6314
## canopy_cover -0.092920 0.069269 6 -1.3414462 0.2283
## n_tree_spec -0.246951 0.918143 6 -0.2689682 0.7970
## n_tree_ind 0.022661 0.386631 6 0.0586127 0.9552
## log(dbh_min) 2.407254 2.053033 6 1.1725356 0.2854
## dbh_min5 0.047614 0.138542 6 0.3436810 0.7428
## log(n_microhabitats) 2.742905 2.485440 6 1.1035893 0.3121
## temperature -0.208719 0.485090 1 -0.4302687 0.7413
## categorypark:size -0.005155 0.182488 1 -0.0282467 0.9820
## Correlation:
## (Intr) ctgryp size cnpy_c n_tr_s n_tr_n lg(d_) dbh_m5
## categorypark -0.369
## size -0.369 0.650
## canopy_cover -0.613 0.414 0.094
## n_tree_spec -0.314 -0.541 -0.315 0.068
## n_tree_ind -0.332 0.063 -0.053 0.097 0.417
## log(dbh_min) -0.301 -0.239 -0.133 0.014 0.279 0.059
## dbh_min5 -0.014 -0.545 -0.047 -0.115 0.361 0.077 -0.219
## log(n_microhabitats) -0.573 0.511 0.220 0.330 -0.146 -0.141 -0.049 -0.390
## temperature -0.553 0.306 0.399 0.293 -0.259 -0.461 0.028 0.042
## categorypark:size -0.200 0.254 0.145 0.310 -0.383 -0.460 -0.395 -0.010
## lg(n_) tmprtr
## categorypark
## size
## canopy_cover
## n_tree_spec
## n_tree_ind
## log(dbh_min)
## dbh_min5
## log(n_microhabitats)
## temperature 0.595
## categorypark:size 0.621 0.628
##
## Standardized Within-Group Residuals:
## Min Q1 Med Q3 Max
## -1.6063561 -0.7198577 -0.2194506 0.9435601 1.8810180
##
## Number of Observations: 18
## Number of Groups: 6

mod2.2 <- stepAIC(mod2.1) #model simplification based on AIC-value of the model

## Start: AIC=110.98
## species_abund ~ category * size + canopy_cover + n_tree_spec +
## n_tree_ind + log(dbh_min) + dbh_min5 + log(n_microhabitats) +
## temperature

```

```

##
##               Df      AIC
## - category:size      1 108.98
## - n_tree_ind         1 108.99
## - n_tree_spec        1 109.17
## - dbh_min5           1 109.28
## - temperature        1 109.45
## <none>                110.98
## - log(n_microhabitats) 1 111.87
## - log(dbh_min)        1 112.21
## - canopy_cover        1 113.10
##
## Step:  AIC=108.98
## species_abund ~ category + size + canopy_cover + n_tree_spec +
##       n_tree_ind + log(dbh_min) + dbh_min5 + log(n_microhabitats) +
##       temperature
##
##               Df      AIC
## - n_tree_ind         1 106.99
## - category           1 107.09
## - n_tree_spec        1 107.22
## - dbh_min5           1 107.28
## - temperature        1 107.69
## - size               1 108.06
## <none>                108.98
## - log(dbh_min)       1 110.69
## - canopy_cover       1 111.44
## - log(n_microhabitats) 1 111.60
##
## Step:  AIC=106.99
## species_abund ~ category + size + canopy_cover + n_tree_spec +
##       log(dbh_min) + dbh_min5 + log(n_microhabitats) + temperature
##
##               Df      AIC
## - category           1 105.09
## - n_tree_spec        1 105.27
## - dbh_min5           1 105.29
## - temperature        1 105.71
## - size               1 106.07
## <none>                106.99
## - log(dbh_min)       1 108.81
## - log(n_microhabitats) 1 109.73
## - canopy_cover       1 109.87
##
## Step:  AIC=105.09
## species_abund ~ size + canopy_cover + n_tree_spec + log(dbh_min) +
##       dbh_min5 + log(n_microhabitats) + temperature
##
##               Df      AIC
## - n_tree_spec        1 103.27
## - dbh_min5           1 103.89
## - temperature        1 104.00
## <none>                105.09
## - size               1 105.64

```

```

## - log(dbh_min)          1 107.08
## - log(n_microhabitats) 1 108.21
## - canopy_cover         1 108.85
##
## Step: AIC=103.27
## species_abund ~ size + canopy_cover + log(dbh_min) + dbh_min5 +
##       log(n_microhabitats) + temperature
##
##               Df    AIC
## - temperature      1 102.03
## - dbh_min5         1 102.10
## <none>              103.27
## - size             1 103.67
## - log(dbh_min)     1 105.57
## - canopy_cover     1 107.32
## - log(n_microhabitats) 1 108.41
##
## Step: AIC=102.04
## species_abund ~ size + canopy_cover + log(dbh_min) + dbh_min5 +
##       log(n_microhabitats)
##
##               Df    AIC
## - dbh_min5         1 101.49
## - size             1 101.81
## <none>              102.03
## - canopy_cover     1 105.76
## - log(dbh_min)     1 106.20
## - log(n_microhabitats) 1 108.22
##
## Step: AIC=101.49
## species_abund ~ size + canopy_cover + log(dbh_min) + log(n_microhabitats)
##
##               Df    AIC
## <none>              101.49
## - size             1 103.86
## - canopy_cover     1 105.00
## - log(dbh_min)     1 111.47
## - log(n_microhabitats) 1 112.79

```

```
summary(mod2.2) #final model which includes only the most important predictors
```

```

## Linear mixed-effects model fit by maximum likelihood
##   Data: dat
##       AIC      BIC    logLik
## 101.4876 107.7202 -43.74381
##
## Random effects:
## Formula: ~1 | site
##      (Intercept) Residual
## StdDev: 6.353355e-05 2.749099
##
## Fixed effects: species_abund ~ size + canopy_cover + log(dbh_min) + log(n_microhabitats)
##               Value Std.Error DF   t-value p-value
## (Intercept)    11.127950  4.317066  9   2.577665  0.0298

```

```
## size                -0.026102  0.013803  4 -1.891107  0.1316
## canopy_cover        -0.090785  0.042043  9 -2.159361  0.0591
## log(dbh_min)         3.715107  1.059513  9  3.506431  0.0067
## log(n_microhabitats) 3.660144  0.964060  9  3.796592  0.0042
## Correlation:
##                    (Intr) size  cnpy_c lg(d_)
## size                -0.087
## canopy_cover        -0.533 -0.439
## log(dbh_min)        -0.896  0.203  0.311
## log(n_microhabitats) -0.143 -0.152 -0.326  0.003
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.66990039 -0.84208140  0.03715311  0.85684838  1.71424488
##
## Number of Observations: 18
## Number of Groups: 6
```

```
anova(mod2.1, mod2.2)
```

```
##           Model df          AIC          BIC      logLik    Test  L.Ratio p-value
## mod2.1         1 13 110.9826 122.5575 -42.49131
## mod2.2         2  7 101.4876 107.7202 -43.74381 1 vs 2 2.505011 0.8679
```

```
#mod2.3 <- update(mod2.2, ~.-size)
#summary(mod2.3)
#anova(mod2.2, mod2.3)

#mod2.4<-update(mod2.3, ~.-canopy_cover)
#summary(mod2.4)
#anova(mod2.2,mod2.4)
```

```
#2d Plotting
```

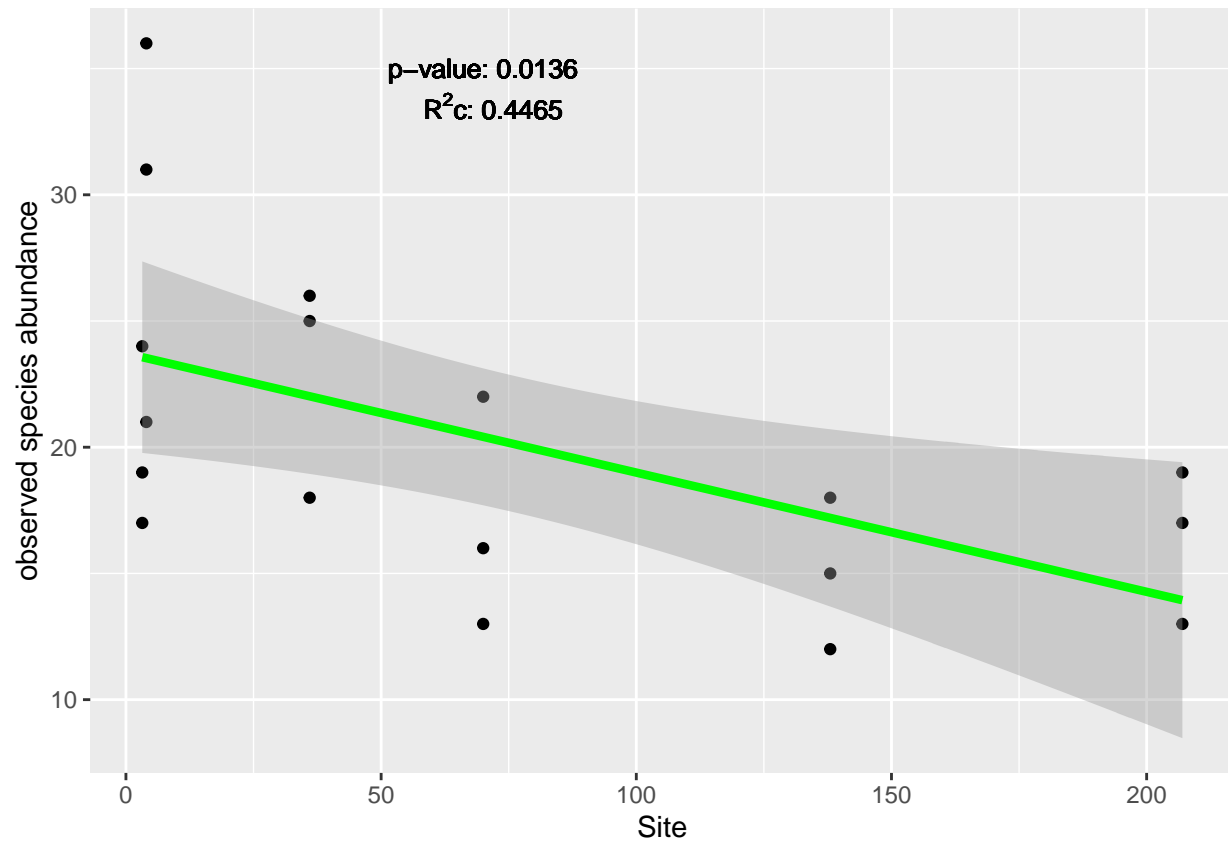
```
# plot model for abundance vs. size
mod_size<-lme(species_abund ~ size, random = (~1|site), data=dat, method="ML")
r.squaredGLMM(mod_size)
```

```
##           R2m          R2c
## [1,] 0.3340045 0.4465148
```

```
ggplot(dat, aes(x = size, y = species_abund))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=70, y=35, label="p-value: 0.0136", size=3.5) +
  geom_text(x=72,y=33.5, label=expression(paste("R"2,"c: 0.4465")), size=3.5) +
  labs(x="Site", y="observed species abundance")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

```
## Warning in is.na(x): is.na() aplicado a un objeto que no es (lista o vector) de
## tipo 'expression'
```



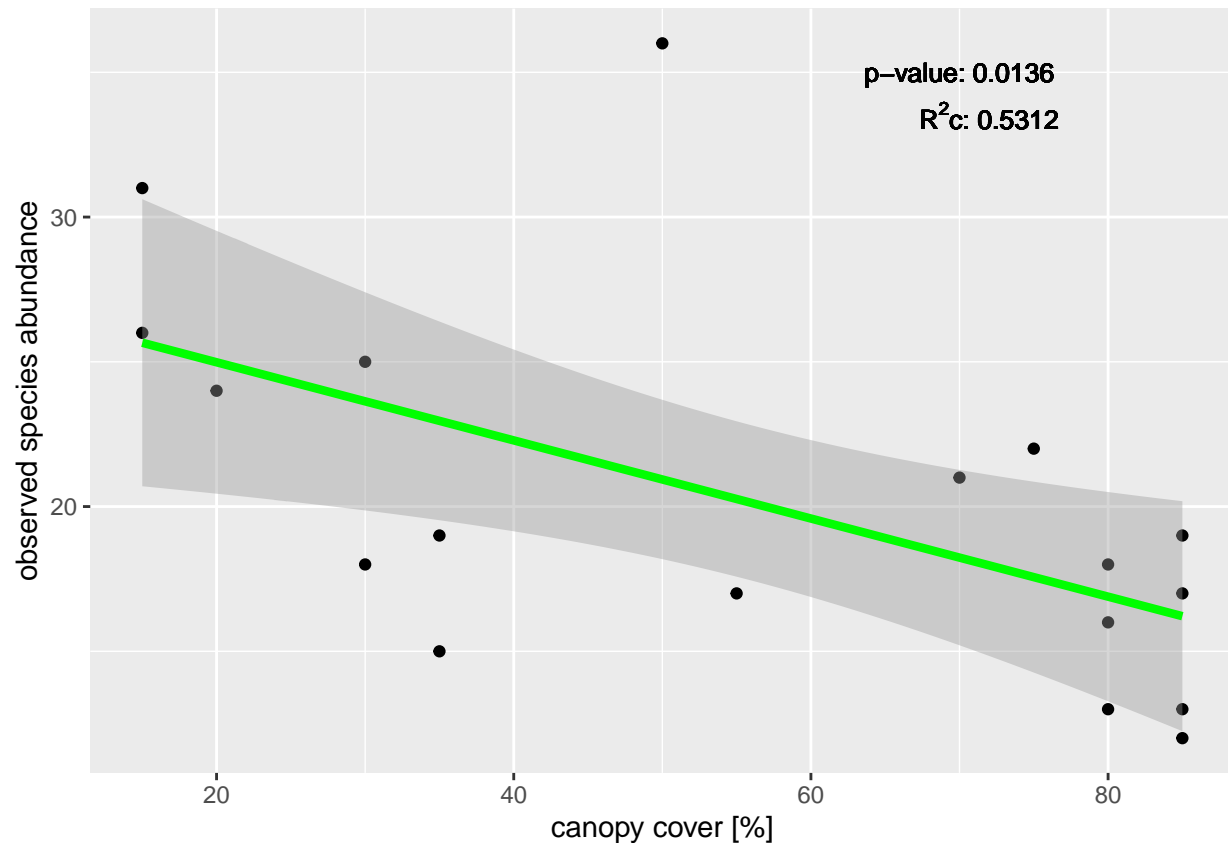
```
# plot model for abundance vs. canopy cover
mod_can<-lme(species_abund ~ canopy_cover, random = (~1|site), data=dat, method="ML")
r.squaredGLMM(mod_can)
```

```
##           R2m      R2c
## [1,] 0.3208035 0.5312365
```

```
ggplot(dat, aes(x = canopy_cover, y = species_abund))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=70, y=35, label="p-value: 0.0136", size=3.5) +
  geom_text(x=72,y=33.5, label=expression(paste("R"2,"c: 0.5312")), size=3.5) +
  labs(x="canopy cover [%]", y="observed species abundance")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

```
## Warning in is.na(x): is.na() aplicado a un objeto que no es (lista o vector) de
## tipo 'expression'
```



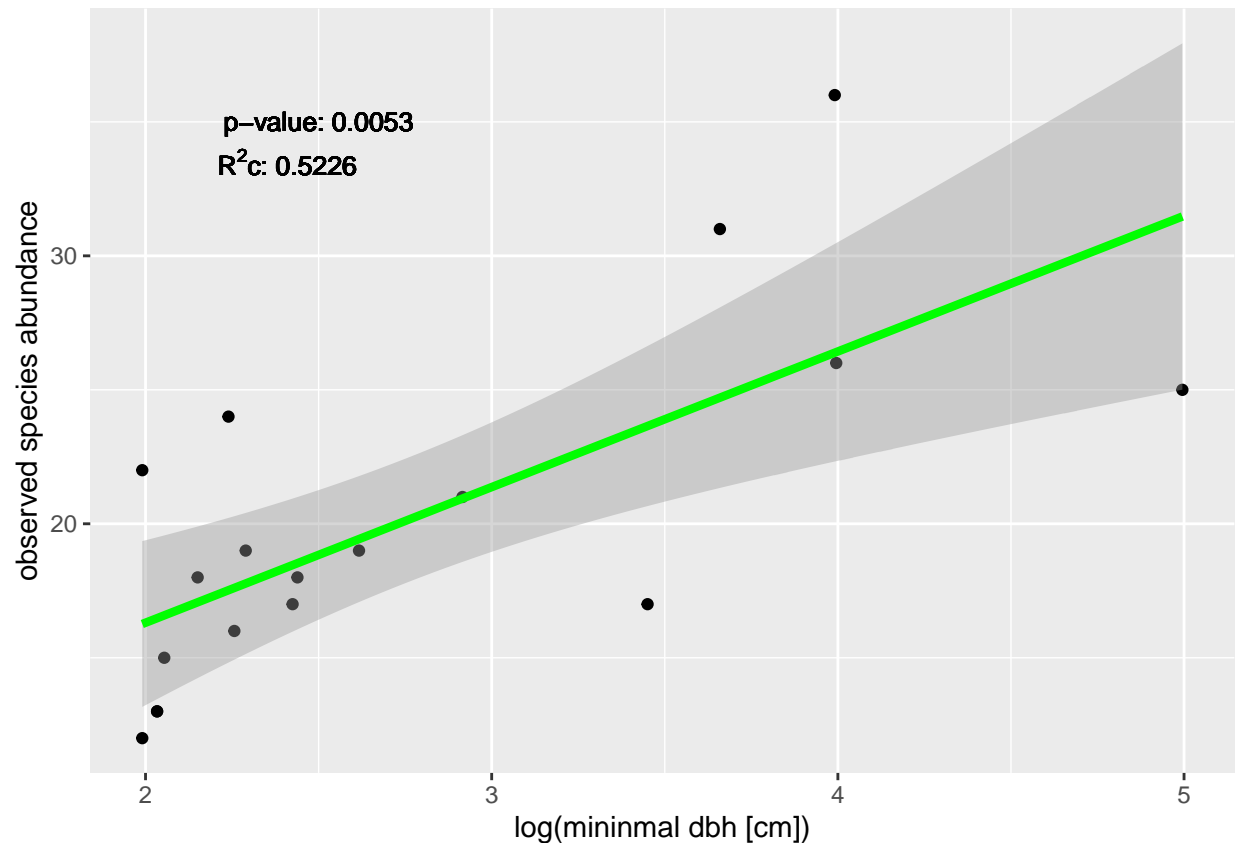
```
#plot model for abundance vs. minimal dbh
mod_dbh<-lme(species_abund ~ log(dbh_min), random = (~1|site), data=dat, method="ML")
r.squaredGLMM(mod_dbh)
```

```
##           R2m           R2c
## [1,] 0.506479 0.5225862
```

```
ggplot(dat, aes(x = log(dbh_min), y = species_abund))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=2.5, y=35, label="p-value: 0.0053", size=3.5) +
  geom_text(x=2.41,y=33.5, label=expression(paste("R"^2,"c: 0.5226")), size=3.5) +
  labs(x="log(mininmal dbh [cm])", y="observed species abundance")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

```
## Warning in is.na(x): is.na() aplicado a un objeto que no es (lista o vector) de
## tipo 'expression'
```



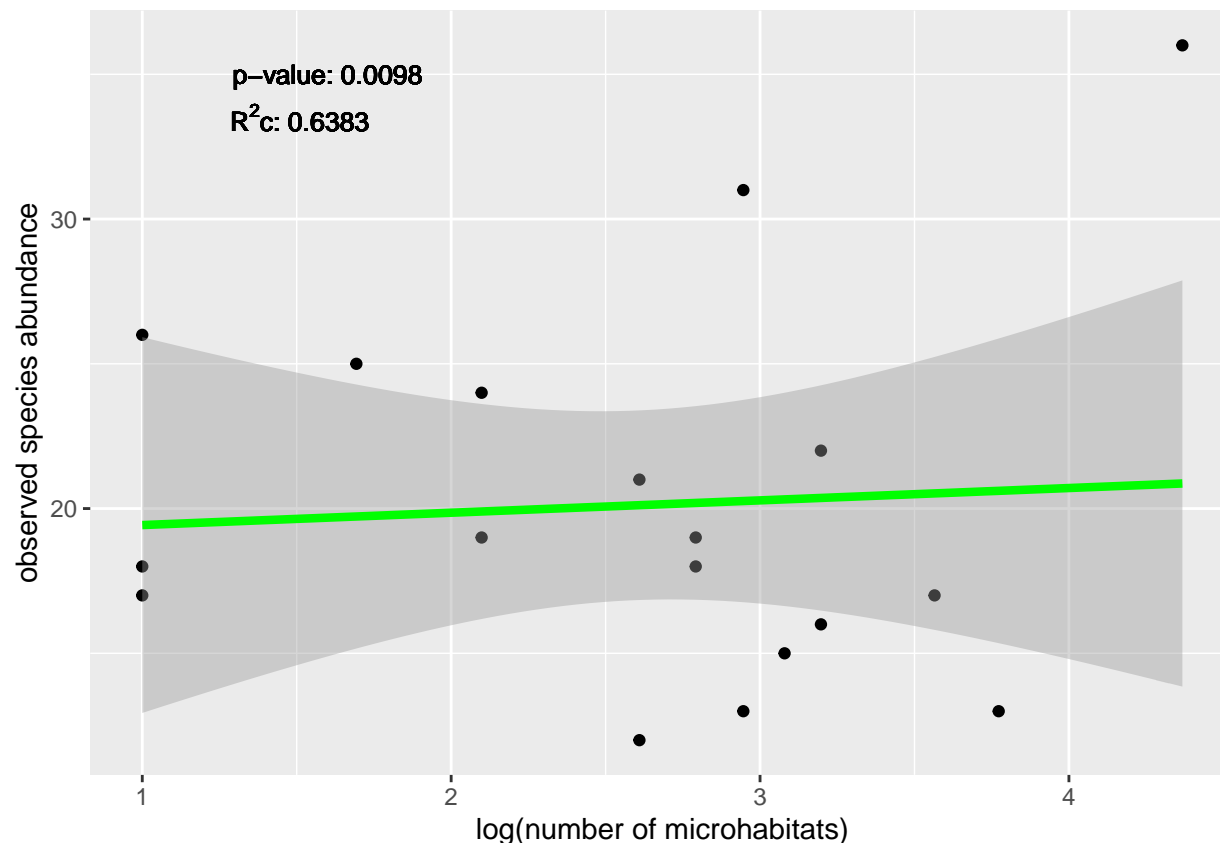
```
# plot model for abundance vs. microhabitats
mod_mic<-lme(species_abund ~ log(n_microhabitats), random = (~1|site), data=dat, method="ML")
r.squaredGLMM(mod_mic)
```

```
##           R2m       R2c
## [1,] 0.1144617 0.6382438
```

```
ggplot(dat, aes(x = log(n_microhabitats)+1, y = species_abund))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=1.6, y=35, label="p-value: 0.0098", size=3.5) +
  geom_text(x=1.51,y=33.5, label=expression(paste("R"^2,"c: 0.6383")), size=3.5) +
  labs(x="log(number of microhabitats)", y="observed species abundance")
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

```
## Warning in is.na(x): is.na() aplicado a un objeto que no es (lista o vector) de
## tipo 'expression'
```

MIXED EFFECTS MODEL WITH RAREFIED RICHNESS

```
mod3 <- lme(rarefied_richness ~ category*size + canopy_cover + n_tree_spec + n_tree_ind + log(dbh_min) +
summary(mod3)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: dat
##       AIC      BIC    logLik
##  58.56492 69.24938 -17.28246
##
## Random effects:
## Formula: ~1 | site
##      (Intercept)  Residual
## StdDev: 2.442777e-05 0.6320405
##
## Fixed effects: rarefied_richness ~ category * size + canopy_cover + n_tree_spec +      n_tree_ind +
##
##              Value Std.Error DF   t-value p-value
## (Intercept)    8.419457  3.192485  7  2.6372736  0.0336
## categorypark   -0.269752  1.559754  1 -0.1729449  0.8910
## size           -0.000680  0.006511  1 -0.1044220  0.9338
## canopy_cover   -0.009324  0.015123  7 -0.6165055  0.5571
## n_tree_spec    -0.193305  0.197460  7 -0.9789587  0.3602
## n_tree_ind     -0.026853  0.088828  7 -0.3022983  0.7712
## log(dbh_min)    0.459643  0.462676  7  0.9934439  0.3536
## n_microhabitats -0.007390  0.047624  7 -0.1551677  0.8811
## temperature    -0.019742  0.095026  1 -0.2077477  0.8696
## categorypark:size 0.017530  0.034535  1  0.5076063  0.7010
```

```
## Correlation:
##          (Intr) ctgryp size   cnpy_c n_tr_s n_tr_n lg(d_) n_mcrh
## categorypark -0.294
## size         -0.308  0.736
## canopy_cover -0.569  0.348  0.026
## n_tree_spec  -0.420 -0.472 -0.325  0.128
## n_tree_ind   -0.469  0.179 -0.029  0.142  0.415
## log(dbh_min) -0.452 -0.421 -0.125  0.023  0.388  0.079
## n_microhabitats -0.285  0.068  0.051  0.079  0.023 -0.117 -0.160
## temperature  -0.333  0.206  0.332  0.154 -0.330 -0.510  0.085  0.478
## categorypark:size 0.209  0.078  0.018  0.165 -0.480 -0.512 -0.441  0.428
##          tmprtr
## categorypark
## size
## canopy_cover
## n_tree_spec
## n_tree_ind
## log(dbh_min)
## n_microhabitats
## temperature
## categorypark:size 0.467
##
## Standardized Within-Group Residuals:
##          Min          Q1          Med          Q3          Max
## -2.1905486 -0.3771982  0.1406606  0.6747755  1.9225911
##
## Number of Observations: 18
## Number of Groups: 6
```

```
mod3.1 <- stepAIC(mod3) #model simplification based on AIC-value of the model
```

```
## Start: AIC=58.56
## rarefied_richness ~ category * size + canopy_cover + n_tree_spec +
##       n_tree_ind + log(dbh_min) + n_microhabitats + temperature
##
##              Df      AIC
## - n_microhabitats 1 56.619
## - temperature     1 56.662
## - n_tree_ind       1 56.769
## - category:size    1 57.136
## - canopy_cover     1 57.400
## - n_tree_spec      1 58.543
## <none>              58.565
## - log(dbh_min)     1 58.659
##
## Step: AIC=56.62
## rarefied_richness ~ category + size + canopy_cover + n_tree_spec +
##       n_tree_ind + log(dbh_min) + temperature + category:size
##
##              Df      AIC
## - temperature     1 54.671
## - n_tree_ind       1 54.851
## - canopy_cover     1 55.425
## - category:size    1 55.502
```

```

## - n_tree_spec      1 56.591
## <none>              56.619
## - log(dbh_min)     1 56.659
##
## Step: AIC=54.67
## rarefied_richness ~ category + size + canopy_cover + n_tree_spec +
##      n_tree_ind + log(dbh_min) + category:size
##
##              Df      AIC
## - n_tree_ind      1 53.159
## - canopy_cover     1 53.436
## - category:size    1 53.814
## <none>              54.671
## - log(dbh_min)     1 54.890
## - n_tree_spec      1 55.119
##
## Step: AIC=53.16
## rarefied_richness ~ category + size + canopy_cover + n_tree_spec +
##      log(dbh_min) + category:size
##
##              Df      AIC
## - canopy_cover     1 51.664
## - category:size    1 51.880
## <none>              53.159
## - n_tree_spec      1 53.213
## - log(dbh_min)     1 53.774
##
## Step: AIC=51.66
## rarefied_richness ~ category + size + n_tree_spec + log(dbh_min) +
##      category:size
##
##              Df      AIC
## - category:size    1 50.738
## - n_tree_spec      1 51.381
## <none>              51.664
## - log(dbh_min)     1 52.136
##
## Step: AIC=50.74
## rarefied_richness ~ category + size + n_tree_spec + log(dbh_min)
##
##              Df      AIC
## - size              1 48.740
## - category          1 48.753
## - n_tree_spec       1 49.576
## <none>              50.738
## - log(dbh_min)     1 53.379
##
## Step: AIC=48.74
## rarefied_richness ~ category + n_tree_spec + log(dbh_min)
##
##              Df      AIC
## - category          1 46.758
## - n_tree_spec       1 47.633
## <none>              48.740

```

```
## - log(dbh_min) 1 51.712
##
## Step: AIC=46.76
## rarefied_richness ~ n_tree_spec + log(dbh_min)
##
##           Df    AIC
## - n_tree_spec 1 46.091
## <none>          46.758
## - log(dbh_min) 1 52.240
##
## Step: AIC=46.09
## rarefied_richness ~ log(dbh_min)
##
##           Df    AIC
## <none>          46.091
## - log(dbh_min) 1 50.433
```

```
summary(mod3.1) #final model which includes only the most important predictors
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: dat
##       AIC      BIC    logLik
## 46.09053 49.65202 -19.04527
##
## Random effects:
## Formula: ~1 | site
##      (Intercept) Residual
## StdDev:  0.3015021 0.6402719
##
## Fixed effects: rarefied_richness ~ log(dbh_min)
##              Value Std.Error DF   t-value p-value
## (Intercept)  5.551859 0.6230627 11  8.910594  0.0000
## log(dbh_min) 0.567502 0.2136461 11  2.656270  0.0223
## Correlation:
##              (Intr)
## log(dbh_min) -0.943
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.78828290 -0.39209673  0.05314066  0.66857616  1.81864161
##
## Number of Observations: 18
## Number of Groups: 6
```

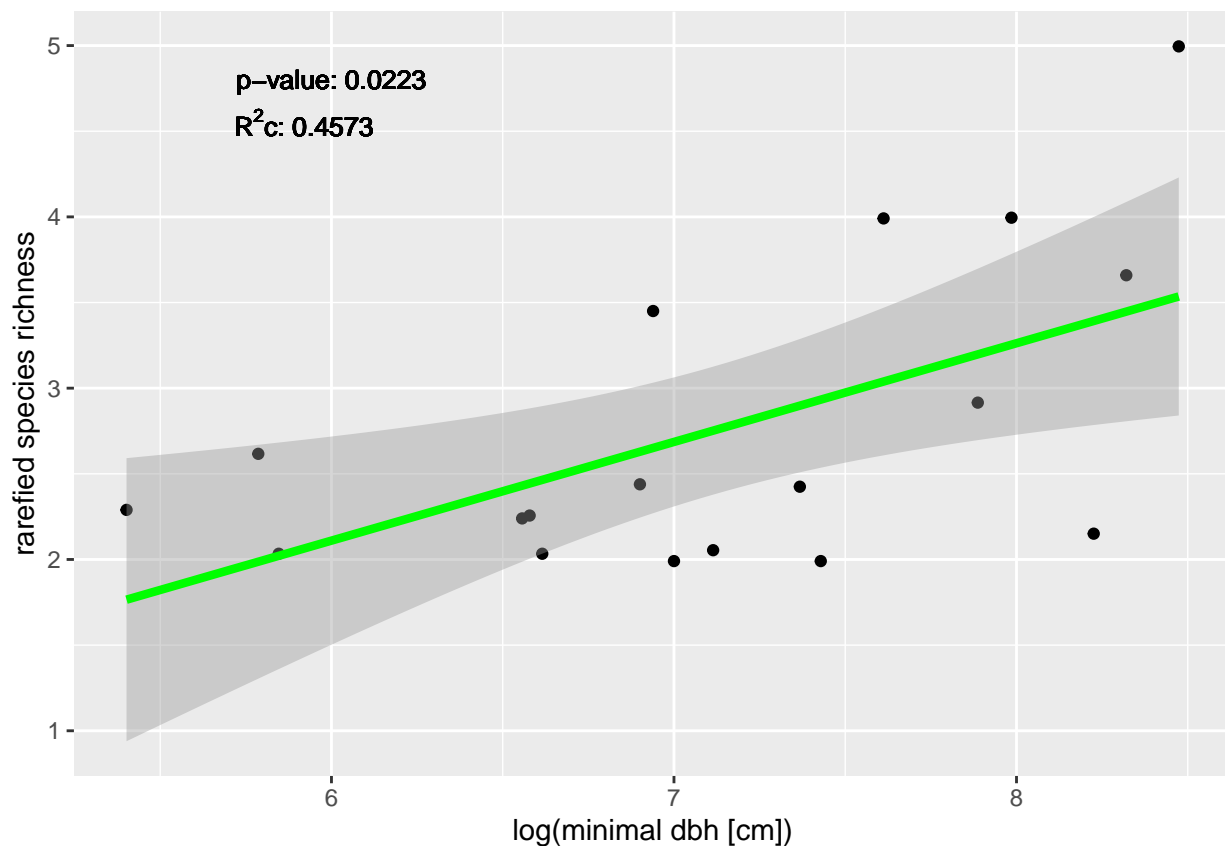
```
# plot model for rarefied richness vs. min dbh
mod_dbh<-lme(rarefied_richness ~ log(dbh_min), random = (~1|site), data=dat, method="ML")
r.squaredGLMM(mod3.1)
```

```
##           R2m           R2c
## [1,] 0.33697 0.4573085
```

```
ggplot(dat, aes(x = rarefied_richness, y = log(dbh_min)))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=6, y=4.8, label="p-value: 0.0223", size=3.5) +
  geom_text(x=5.92,y=4.55, label=expression(paste("R"2, "c: 0.4573")), size=3.5) +
  labs(x="log(minimal dbh [cm])", y="rarefied species richness")
```

'geom_smooth()' using formula 'y ~ x'

Warning in is.na(x): is.na() aplicado a un objeto que no es (lista o vector) de
tipo 'expression'



```
#plot(log(dbh_min)~rarefied_richness, data=dat, ylim=c(0,5))
#mod<-lm(rarefied_richness ~ log(dbh_min), data=dat, poly(degree = 2))
```

plot graphs

```
a <- ggplot(dat, aes(x = log(dbh_min), y = species_richness))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=2.6, y=16, label="p-value: 0.0001", size=3.5) +
  geom_text(x=2.51,y=15.3,label=expression(paste("R"2, "c: 0.6851")), size=3.5) +
  labs(x="log(mininmal dbh [cm])", y="observed species richness")
```

```

b <- ggplot(dat, aes(x = size, y = species_abund))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=70, y=35, label="p-value: 0.0136", size=3.5) +
  geom_text(x=72,y=33.5, label=expression(paste("R"^2,"c: 0.4465")), size=3.5) +
  labs(x="Site", y="observed species abundance")

c <-ggplot(dat, aes(x = canopy_cover, y = species_abund))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=70, y=35, label="p-value: 0.0136", size=3.5) +
  geom_text(x=72,y=33.5, label=expression(paste("R"^2,"c: 0.5312")), size=3.5) +
  labs(x="canopy cover [%]", y="observed species abundance")

d <- ggplot(dat, aes(x = log(dbh_min), y = species_abund))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=2.5, y=35, label="p-value: 0.0053", size=3.5) +
  geom_text(x=2.41,y=33.5, label=expression(paste("R"^2,"c: 0.5226")), size=3.5) +
  labs(x="log(minimal dbh [cm])", y="observed species abundance")

e <- ggplot(dat, aes(x = log(n_microhabitats)+1, y = species_abund))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=1.6, y=35, label="p-value: 0.0098", size=3.5) +
  geom_text(x=1.51,y=33.5, label=expression(paste("R"^2,"c: 0.6383")), size=3.5) +
  labs(x="log(number of microhabitats)", y="observed species abundance")

f <- ggplot(dat, aes(x = rarefied_richness, y = log(dbh_min)))+
  geom_point() +
  geom_smooth(method = lm, se = TRUE, colour = 'green', size = 1.5) +
  geom_text(x=6, y=4.8, label="p-value: 0.0223", size=3.5) +
  geom_text(x=5.92,y=4.55, label=expression(paste("R"^2,"c: 0.4573")), size=3.5) +
  labs(x="log(minimal dbh [cm])", y="rarefied species richness")

```

##2c Ordination with NMDS (to look for differences in species composition)

```
nmd1 <- metaMDS(dat[,4:31], distance="horn", k=2) #NMDS analysis based on Morisita-Horn-Index as a diss
```

```

## Run 0 stress 0.1663514
## Run 1 stress 0.1707767
## Run 2 stress 0.1777238
## Run 3 stress 0.1663514
## ... New best solution
## ... Procrustes: rmse 3.436212e-05  max resid 9.07174e-05
## ... Similar to previous best
## Run 4 stress 0.2268103
## Run 5 stress 0.1663514
## ... Procrustes: rmse 4.36103e-05  max resid 0.0001420161
## ... Similar to previous best

```

```
## Run 6 stress 0.1663514
## ... Procrustes: rmse 1.355284e-05  max resid 4.541948e-05
## ... Similar to previous best
## Run 7 stress 0.1663514
## ... Procrustes: rmse 5.124492e-05  max resid 0.0001690546
## ... Similar to previous best
## Run 8 stress 0.1777238
## Run 9 stress 0.1663514
## ... Procrustes: rmse 6.80829e-06  max resid 2.233673e-05
## ... Similar to previous best
## Run 10 stress 0.1663514
## ... Procrustes: rmse 4.018989e-05  max resid 0.0001083132
## ... Similar to previous best
## Run 11 stress 0.1663514
## ... Procrustes: rmse 4.451713e-06  max resid 1.46174e-05
## ... Similar to previous best
## Run 12 stress 0.1663514
## ... Procrustes: rmse 2.663593e-05  max resid 8.737447e-05
## ... Similar to previous best
## Run 13 stress 0.1828469
## Run 14 stress 0.1777238
## Run 15 stress 0.1663514
## ... Procrustes: rmse 5.383646e-05  max resid 0.0001762232
## ... Similar to previous best
## Run 16 stress 0.182847
## Run 17 stress 0.1663514
## ... Procrustes: rmse 2.159601e-05  max resid 7.101394e-05
## ... Similar to previous best
## Run 18 stress 0.1707767
## Run 19 stress 0.1663514
## ... Procrustes: rmse 6.347427e-06  max resid 1.934522e-05
## ... Similar to previous best
## Run 20 stress 0.1777238
## *** Solution reached
```

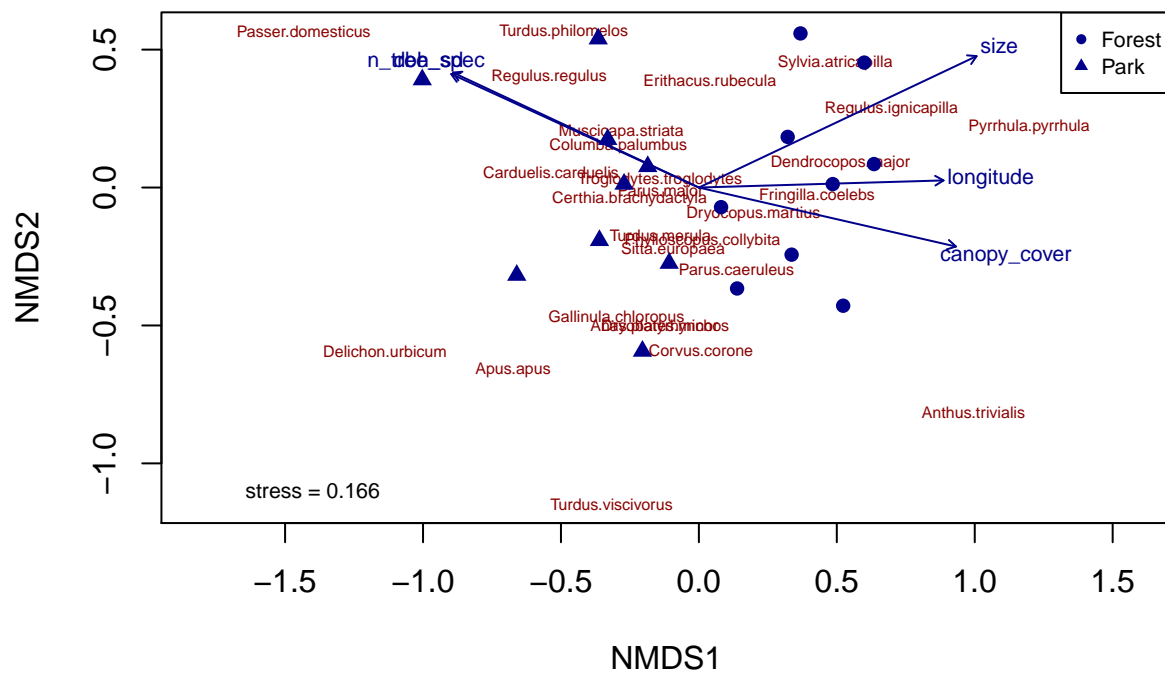
```
# orditkplot(nmd1, display = "species", col = "darkred", fill = NA, border = NA, cex = 0.6)
ordipLOT(nmd1, choices = c(1, 2), type = "n") # ylim = c(-0.75, 0.5), xlim = c(-1.25, 1.3))
ordilabel(nmd1, display = "species", col = "darkred", fill = NA, border = NA, cex = 0.5)
points(nmd1, pch=c(16, 17)[as.numeric(as.factor(dat$category))], col = "darkblue") #add sampling points
legend("topright", pch = c(16, 17), c("Forest","Park"), col = "darkblue", cex = 0.7) #add legend
text(-1.4, -1.1, labels = "stress = 0.166", cex = 0.7)

ef <- envfit(nmd1, dat[,32:44]) #check for correlation of dissimilarity gradients with environmental va
ef #results
```

```
##
## ***VECTORS
##
##          NMDS1    NMDS2    r2 Pr(>r)
## canopy_cover    0.97464 -0.22377 0.5449 0.004 **
## n_tree_spec     -0.90908  0.41661 0.5815 0.002 **
## n_tree_ind       0.98934 -0.14563 0.1254 0.374
## dbh_min         -0.95999 -0.28004 0.0958 0.485
## dbh_min5        -0.66028 -0.75102 0.2847 0.076 .
```

```
## dbh_mean      -0.71934 -0.69466 0.3251  0.058 .
## dbh_max       -0.75683 -0.65361 0.3293  0.055 .
## dbh_median    -0.64468 -0.76445 0.2933  0.062 .
## dbh_sd        -0.90588  0.42354 0.5773  0.002 **
## n_microhabitats 0.48726 -0.87326 0.2378  0.128
## latitude       0.66161 -0.74985 0.2294  0.139
## longitude      0.99958  0.02910 0.4694  0.008 **
## size           0.90394  0.42766 0.7395  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```
plot(ef, p.max=0.05, col = "darkblue", cex = 0.7) #add significant environmental variables to the NMDS
```



Now the species accumulation curve is shown.

```
SAC_park <- specaccum(subset(dat[,4:31], dat$category == "park"))
```

```
## Warning in cor(x > 0): the standard deviation is zero
```

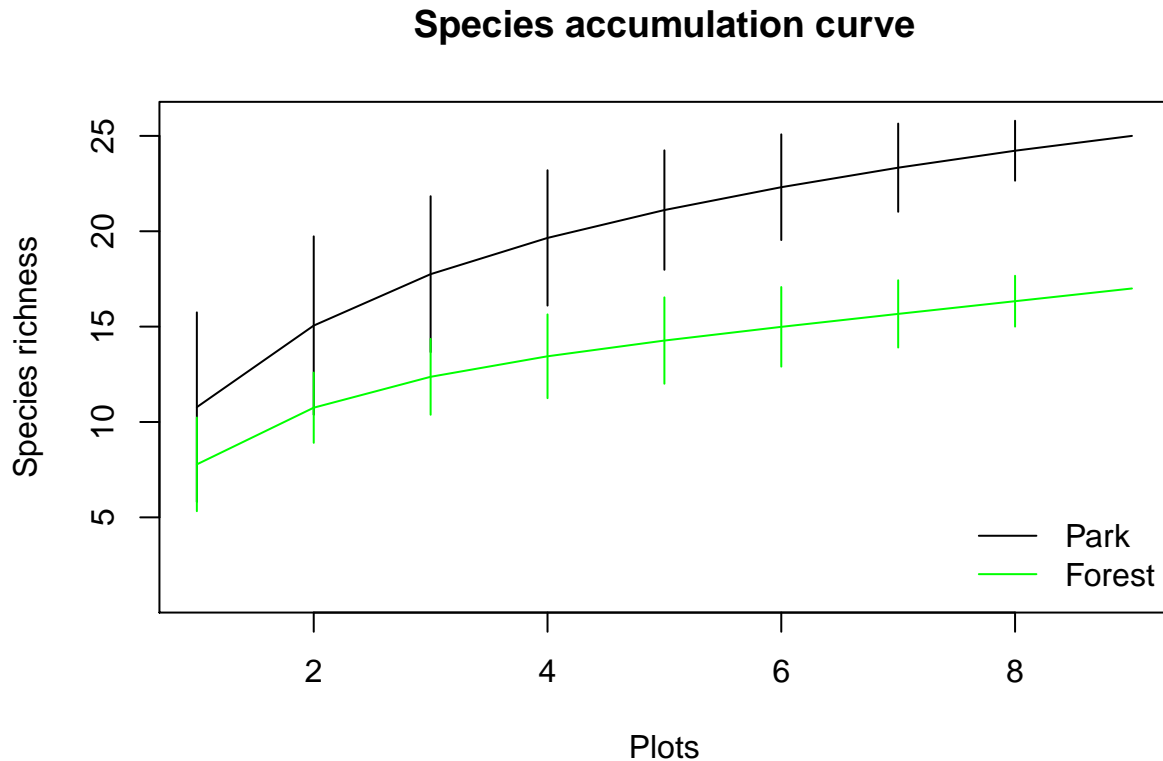
```
SAC_fore <- specaccum(subset(dat[,4:31], dat$category == "forest"))
```

```
## Warning in cor(x > 0): the standard deviation is zero
```



```
plot(SAC_park, xlab = "Plots", ylab = "Species richness", main="Species accumulation curve")

plot(SAC_fore, xlab = "Plots", ylab = "Species richness", main="Species accumulation curve", col="green",
legend("bottomright", legend = c("Park","Forest"), col = c("black","green"), lwd=1, bty = "n")
```



Analysis of bird's diversity and the variables measured from these two types of ecosystems.

```
# parks <- read.csv("data/parks.csv", sep=";")
# forest <- read.csv("data/forest.csv", sep=";")
alpha <- specnumber(dat[,4:31]) # or use the binary site-species matrix
gamma <- ncol(dat[,colSums(dat[,4:31])>0])
```

```
##Lande's index (beta) diversity
gamma - mean(alpha)
```

```
## [1] 40.72222
```

```
##Whittaker's index
```

```
#gamma/mean(alpha)
```

```
#For parks
```

```
alphap <- specnumber(parks[,4:31])
```

```
gammap <- ncol(parks[,colSums(parks[,4:31]) > 0])

##Lande's index
gammap - mean(alphap)
```

```
## [1] 28.22222
```

```
##Whittaker's index
```

```
#gammap/mean(alphap)
```

```
#For forest
```

```
alphaf <- specnumber(forest[,4:31])
```

```
gammaf <- ncol(parks[,colSums(parks[,4:31]) > 0])
```

```
##Lande's index
```

```
gammaf - mean(alphaf)
```

```
## [1] 31.22222
```

```
##Whittaker's index
```

```
#gammaf/mean(alphaf)
```

The number of shared and unique species for a given for the two plots combine and separated.

```
beta_virt <- betadiver(dat[,4:31], method = NA)
# a
beta_virt$a
```

```
##      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
## 2      7
## 3      7  9
## 4      4  5  4
## 5      6  6  6  5
## 6      5  5  7  4  5
## 7      4  5  5  3  3  2
## 8      5  6  7  4  4  6  4
## 9      5  6  6  3  4  4  4  6
## 10     4  5  5  4  5  5  2  5  4
## 11     7  8  8  6  7  6  5  7  6  7
## 12     5  6  7  4  6  6  3  6  5  6  7
## 13     4  6  7  3  4  4  3  5  5  4  5  5
## 14     5  7  7  4  4  5  4  7  6  5  7  5  6
## 15     5  5  6  3  4  4  2  4  4  4  5  5  5  5
## 16     7  6  8  4  6  7  3  5  4  4  6  6  4  4  4
## 17     9  8  9  5  6  7  5  7  7  6  9  6  5  7  6  8
## 18     7  7  8  6  5  7  4  7  6  5  7  6  5  6  6  8  11
```

```
beta_virt$b
```

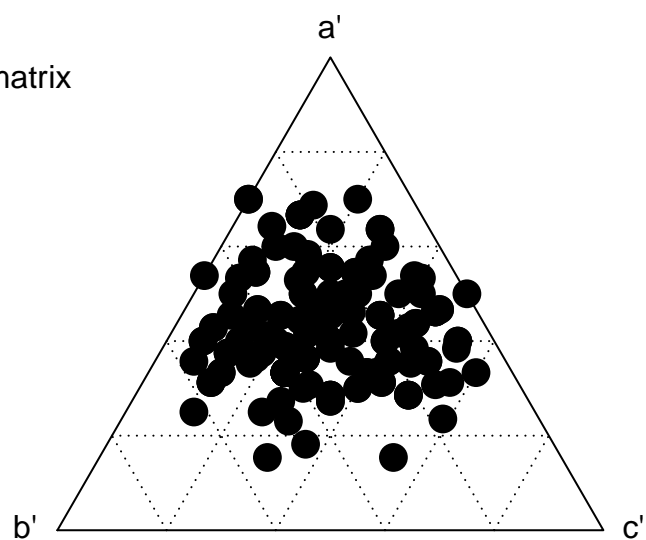
```
##      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
## 2      5
## 3      7  5
## 4      3  2  3
## 5      1  1  1  2
## 6      4  4  2  5  4
## 7      2  1  1  3  3  4
## 8      4  3  2  5  5  3  5
## 9      2  1  1  4  3  3  3  1
## 10     3  2  2  3  2  2  5  2  3
## 11     3  2  2  4  3  4  5  3  4  3
## 12     3  2  1  4  2  2  5  2  3  2  1
## 13     3  1  0  4  3  3  4  2  2  3  2  2
## 14     3  1  1  4  4  3  4  1  2  3  1  3  2
## 15     4  4  3  6  5  5  7  5  5  5  4  4  4  4
## 16     2  3  1  5  3  2  6  4  5  5  3  3  5  5  5
## 17     4  5  4  8  7  6  8  6  6  7  4  7  8  6  7  5
## 18     7  7  6  8  9  7 10  7  8  9  7  8  9  8  8  6  3
```

```
beta_virt$c
```

```
##      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
## 2      4
## 3      4  3
## 4      7  7 10
## 5      5  6  8  2
## 6      6  7  7  3  2
## 7      7  7  9  4  4  7
## 8      6  6  7  3  3  3  2
## 9      6  6  8  4  3  5  2  3
## 10     7  7  9  3  2  4  4  4  3
## 11     4  4  6  1  0  3  1  2  1  0
## 12     6  6  7  3  1  3  3  3  2  1  3
## 13     7  6  7  4  3  5  3  4  2  3  5  3
## 14     6  5  7  3  3  4  2  2  1  2  3  3  1
## 15     6  7  8  4  3  5  4  5  3  3  5  3  2  3
## 16     4  6  6  3  1  2  3  4  3  3  4  2  3  4  5
## 17     2  4  5  2  1  2  1  2  0  1  1  2  2  1  3  1
## 18     4  5  6  1  2  2  2  2  1  2  3  2  2  2  3  1  2
```

```
plot(betadiver(dat[,4:31], method=NA), pch = 16, cex = 2,
legend("topleft", legend = "Virtual matrix", bty = "n")
```

Virtual matrix



```
beta_virtp <- betadiver(parks, method = NA)
# a
beta_virtp$a
```

```
##      1  2  3  4  5  6  7  8
## 2 23
## 3 23 25
## 4 20 22 23
## 5 21 23 23 22
## 6 21 21 22 21 21
## 7 23 22 24 20 20 20
## 8 25 24 25 21 23 22 24
## 9 23 23 24 21 22 22 24 27
```

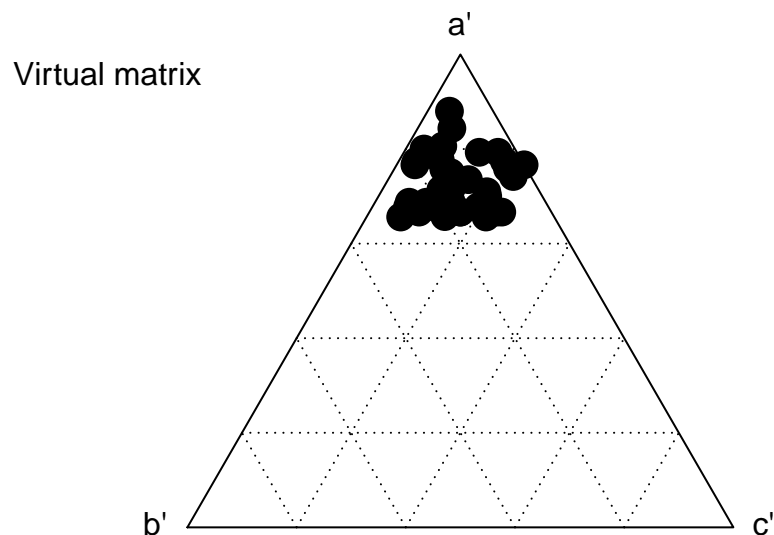
```
beta_virtp$b
```

```
##      1  2  3  4  5  6  7  8
## 2 5
## 3 7 5
## 4 3 1 0
## 5 3 1 1 2
## 6 4 4 3 4 4
## 7 2 3 1 5 5 5
## 8 4 5 4 8 6 7 5
## 9 7 7 6 9 8 8 6 3
```

```
beta_virtp$c
```

```
##  1 2 3 4 5 6 7 8
## 2 4
## 3 4 3
## 4 7 6 7
## 5 6 5 7 1
## 6 6 7 8 2 3
## 7 4 6 6 3 4 5
## 8 2 4 5 2 1 3 1
## 9 4 5 6 2 2 3 1 2
```

```
plot(betadiver(parks, method=NA), pch = 16, cex = 2,)  
legend("topleft", legend = "Virtual matrix", bty = "n")
```

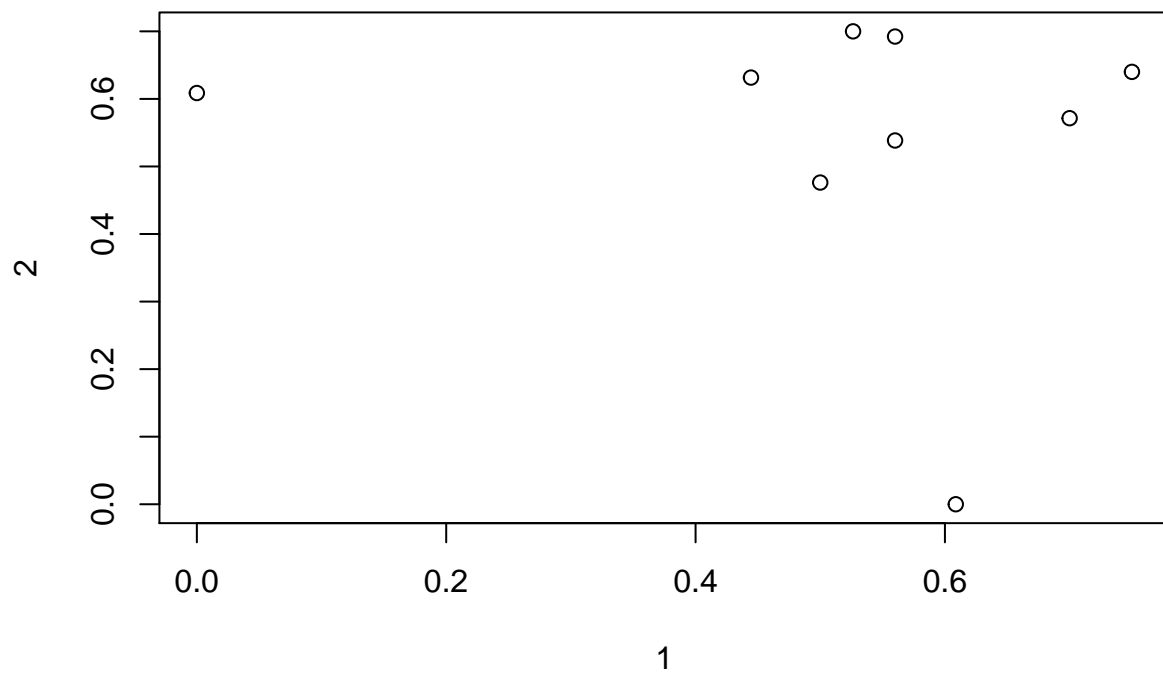


```
#vennd iagram <- as alternative to show the overlap of shared species
```

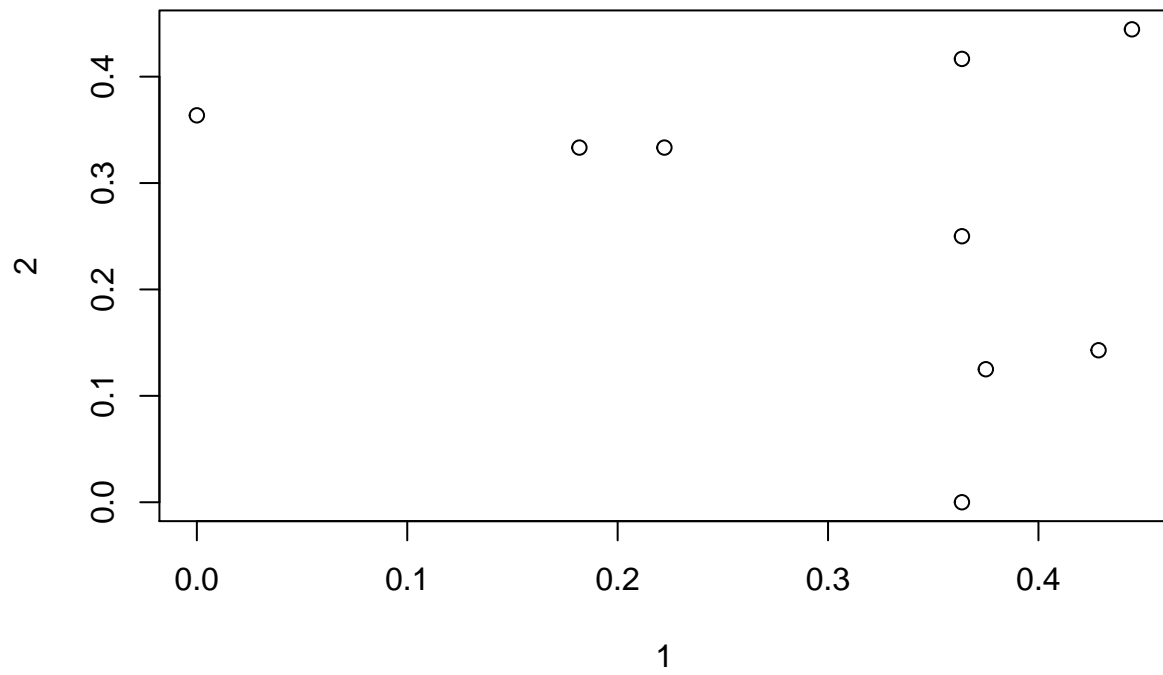
We can see how nice heterogeneity is present between the two ecosystems

Now the similarity between plots by Sorensen, Simpson and Jaccard

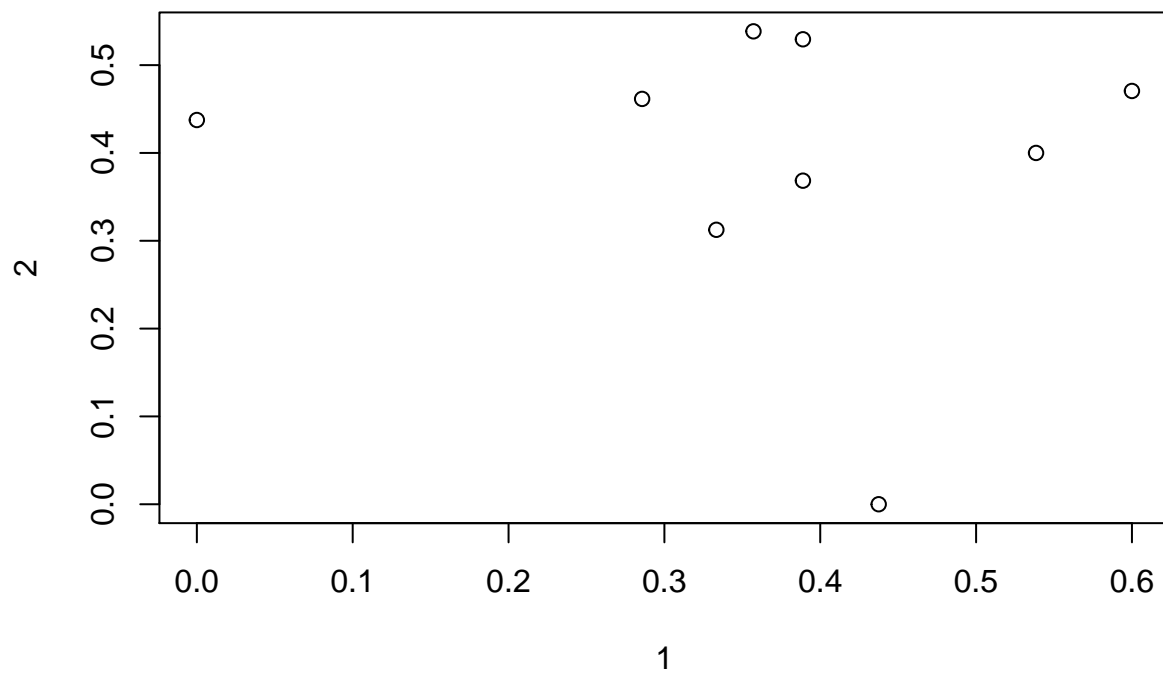
```
#Sorensen similarity  
sorparks <- betadiver(parks[,4:31], method = "sor")  
sorensen <- as.matrix(sorparks)[,]  
plot(sorensen)
```



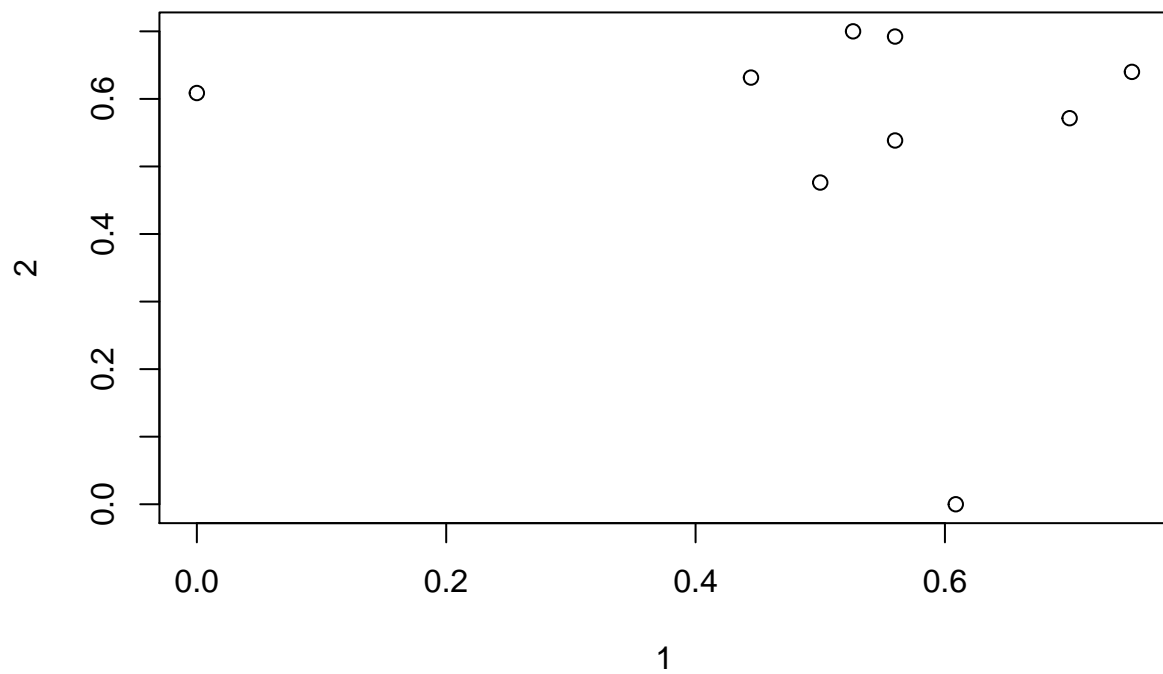
```
#Simpson similarity  
simpark<- betadiver(parks[,4:31], method = "sim")  
simpsonpark <- as.matrix(simpark)[,]  
plot(simpsonpark)
```



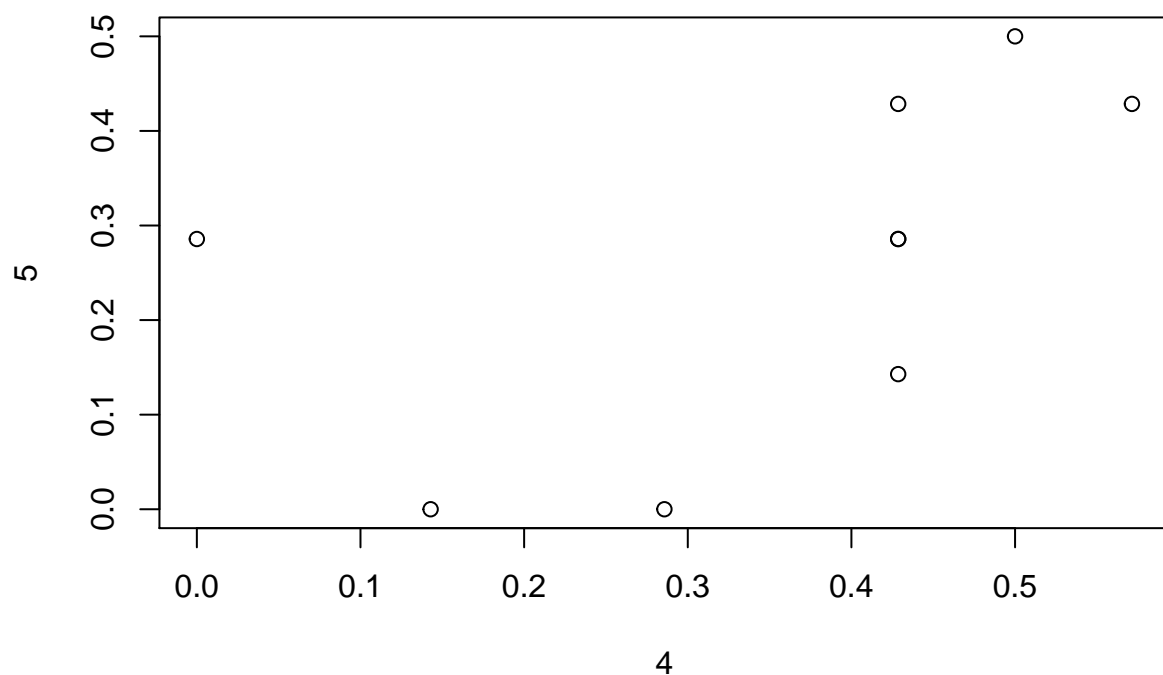
```
# Jaccard similarity  
jparks <- betadiver(parks[,4:31], method = "j")  
plot(as.matrix(jparks)[,])
```



```
#Sorensen similarity  
sorforest <- betadiver(forest[,4:31], method = "sor")  
sorensenforest <- as.matrix(sorforest)[,]  
plot(sorensen)
```

```
#Simpson similarity  
simf <- betadiver(forest[,4:31], method = "sim")  
simpson <- as.matrix(simf)[,]  
plot(simpson)
```

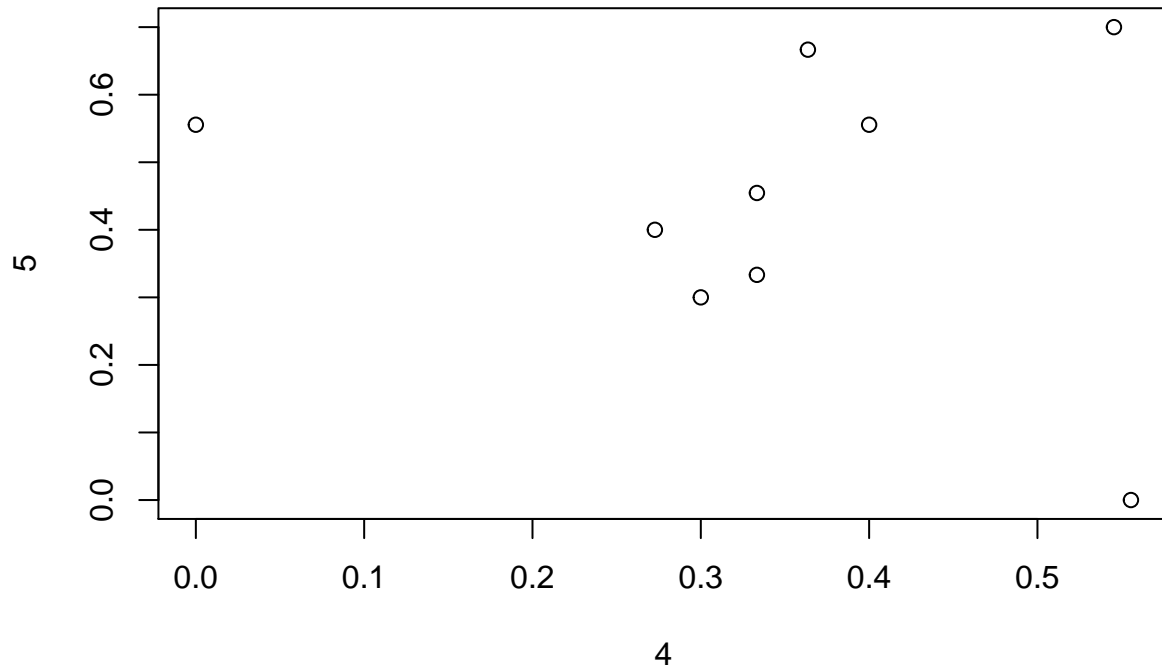


```
# Jaccard similarity
```

```
jforest <- betadiver(forest[,4:31], method = "j")
as.matrix(jforest)[,]
```

```
##           4           5           6           7           8           9           10
## 4  0.0000000  0.5555556  0.3333333  0.3000000  0.3333333  0.2727273  0.4000000
## 5  0.5555556  0.0000000  0.4545455  0.3000000  0.3333333  0.4000000  0.5555556
## 6  0.3333333  0.4545455  0.0000000  0.1538462  0.5000000  0.3333333  0.4545455
## 7  0.3000000  0.3000000  0.1538462  0.0000000  0.3636364  0.4444444  0.1818182
## 8  0.3333333  0.3333333  0.5000000  0.3636364  0.0000000  0.6000000  0.4545455
## 9  0.2727273  0.4000000  0.3333333  0.4444444  0.6000000  0.0000000  0.4000000
## 10 0.4000000  0.5555556  0.4545455  0.1818182  0.4545455  0.4000000  0.0000000
## 11 0.5454545  0.7000000  0.4615385  0.4545455  0.5833333  0.5454545  0.7000000
## 12 0.3636364  0.6666667  0.5454545  0.2727273  0.5454545  0.5000000  0.6666667
##           11           12
## 4  0.5454545  0.3636364
## 5  0.7000000  0.6666667
## 6  0.4615385  0.5454545
## 7  0.4545455  0.2727273
## 8  0.5833333  0.5454545
## 9  0.5454545  0.5000000
## 10 0.7000000  0.6666667
## 11 0.0000000  0.6363636
## 12 0.6363636  0.0000000
```

```
plot(as.matrix(jforest)[,])
```



```
#calculate mean of these similarities indices and compare between sites
```

RANK abundance curve <- takes abundant species per site and plot it against relative species abundance rank

```
# Klosterpark
plot1_1 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[1,4:31]))
plot1_2 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[2,4:31]))
plot1_3 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[3,4:31]))

plot1_12 <- merge(plot1_1, plot1_2, by = "sp")
plot1 <- merge(plot1_12, plot1_3, by = "sp")
plot1$abun <- plot1$ab.x + plot1$ab.y + plot1$ab
plot1 <- plot1[which(plot1$abun!=0),]
dim(plot1)
```

```
## [1] 20 5
```

```
plot1$relabun <- plot1$abun * 100 / sum(plot1$abun)
plot1$rank <- rank(-plot1$relabun, ties.method = "random")
plot1 <- plot1[order(plot1$rank),]
```

```

# City Forest
plot2_1 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[4,4:31]))
plot2_2 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[5,4:31]))
plot2_3 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[6,4:31]))

plot2_12 <- merge(plot2_1, plot2_2, by = "sp")
plot2 <- merge(plot2_12, plot2_3, by = "sp")
plot2$abun <- plot2$ab.x + plot2$ab.y + plot2$ab
plot2 <- plot2[which(plot2$abun!=0),]
dim(plot2)

```

```
## [1] 12 5
```

```

plot2$relabun <- plot2$abun * 100 / sum(plot2$abun)
plot2$rank <- rank(-plot2$abun, ties.method = "random")
plot2 <- plot2[order(plot2$rank),]

```

```

# Forest Weende
plot3_1 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[7,4:31]))
plot3_2 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[8,4:31]))
plot3_3 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[9,4:31]))

plot3_12 <- merge(plot3_1, plot3_2, by = "sp")
plot3 <- merge(plot3_12, plot3_3, by = "sp")
plot3$abun <- plot3$ab.x + plot3$ab.y + plot3$ab
plot3 <- plot3[which(plot3$abun!=0),]
dim(plot3)

```

```
## [1] 12 5
```

```

plot3$relabun <- plot3$abun * 100 / sum(plot3$abun)
plot3$rank <- rank(-plot3$abun, ties.method = "random")
plot3 <- plot3[order(plot3$rank),]

```

```

# Forest Billingshäuser Schlucht
plot4_1 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[10,4:31]))
plot4_2 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[11,4:31]))
plot4_3 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[12,4:31]))

plot4_12 <- merge(plot4_1, plot4_2, by = "sp")
plot4 <- merge(plot4_12, plot4_3, by = "sp")
plot4$abun <- plot4$ab.x + plot4$ab.y + plot4$ab
plot4 <- plot4[which(plot4$abun!=0),]
dim(plot4)

```

```
## [1] 11 5
```

```

plot4$relabun <- plot4$abun * 100 / sum(plot4$abun)
plot4$rank <- rank(-plot4$abun, ties.method = "random")
plot4 <- plot4[order(plot4$rank),]

```

```

# Cheltenham Park
plot5_1 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[13,4:31]))
plot5_2 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[14,4:31]))
plot5_3 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[15,4:31]))

plot5_12 <- merge(plot5_1, plot5_2, by = "sp")
plot5 <- merge(plot5_12, plot5_3, by = "sp")
plot5$abun <- plot5$ab.x + plot5$ab.y + plot5$ab
plot5 <- plot5[which(plot5$abun!=0),]
dim(plot5)

```

```
## [1] 13 5
```

```

plot5$relabun <- plot5$abun * 100 / sum(plot5$abun)
plot5$rank <- rank(-plot5$abun, ties.method = "random")
plot5 <- plot5[order(plot5$rank),]

```

```

# City Cemetery
plot6_1 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[16,4:31]))
plot6_2 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[17,4:31]))
plot6_3 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[18,4:31]))

plot6_12 <- merge(plot6_1, plot6_2, by = "sp")
plot6 <- merge(plot6_12, plot6_3, by = "sp")
plot6$abun <- plot6$ab.x + plot6$ab.y + plot6$ab
plot6 <- plot6[which(plot6$abun!=0),]
dim(plot6)

```

```
## [1] 16 5
```

```

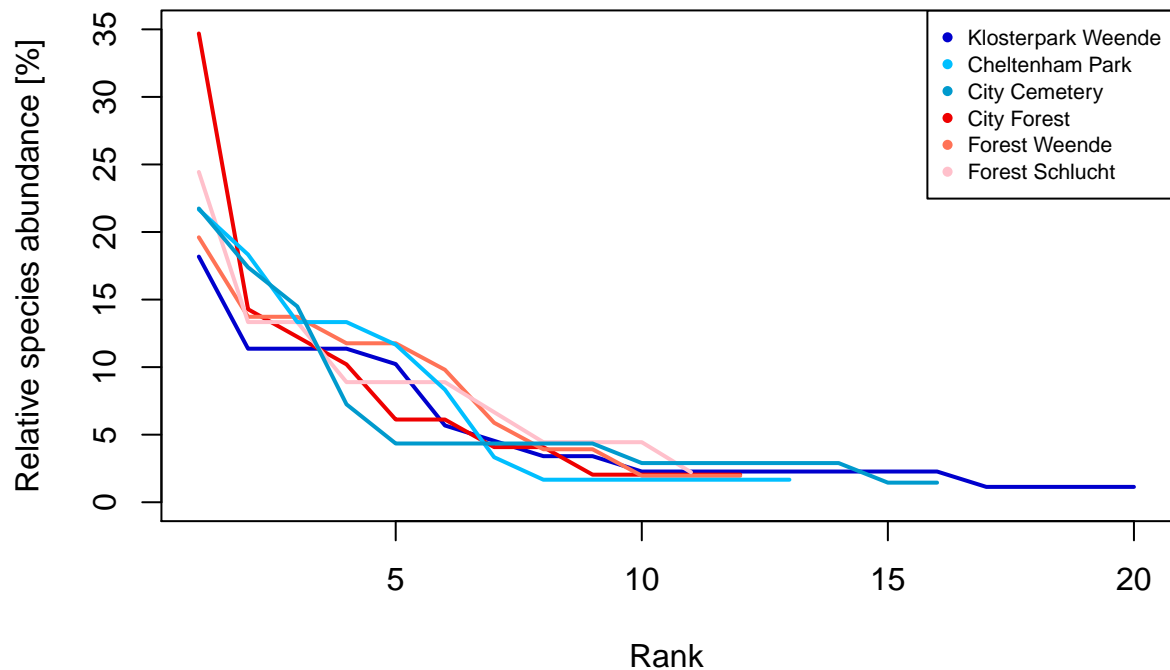
plot6$relabun <- plot6$abun * 100 / sum(plot6$abun)
plot6$rank <- rank(-plot6$abun, ties.method = "random")
plot6 <- plot6[order(plot6$rank),]

```

```

# Plotting
plot(plot1$rank, plot1$relabun, type = "l",
     col = "blue3", pch = 16, lwd = 2,
     ylim = c(0, 35),
     xlab = "Rank", ylab = "Relative species abundance [%]")
points(plot2$rank, plot2$relabun, type = "l",
       col = "red2", pch = 16, lwd = 2)
points(plot3$rank, plot3$relabun, type = "l",
       col = "coral1", pch = 16, lwd = 2)
points(plot4$rank, plot4$relabun, type = "l",
       col = "pink", pch = 16, lwd = 2)
points(plot5$rank, plot5$relabun, type = "l",
       col = "deepskyblue", pch = 16, lwd = 2)
points(plot6$rank, plot6$relabun, type = "l",
       col = "deepskyblue3", pch = 16, lwd = 2)
legend("topright", legend = c("Klosterpark Weende", "Cheltenham Park", "City Cemetery", "City Forest", "

```



```
#library(wesanderson)
#library("RColorBrewer")

#par(mfrow = c(1, 1))
#plot1_1 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[1,4:31]))

#plot1_1 <- plot1_1[which(plot1_1$ab!=0),]
#dim(plot1_1)

# Add rank of species in the first community
#plot1_1$rank <- rank(-plot1_1$ab, ties.method = "random")
# Ordering data before plotting
#plot1_1 <- plot1_1[order(plot1_1$rank), ]
# Plot
#plot(plot1_1$rank, plot1_1$ab, type = "b",
#     col = "coral", pch = 16, lwd = 1,
#     main = "RAD",
#     xlab = "Rank", ylab = "Abundances")

#plot1_2 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[2,4:31]))
#plot1_2 <- plot1_2[which(plot1_2$ab!=0),]
#dim(plot1_2)

#plot1_2$rank <- rank(-plot1_2$ab, ties.method = "random")
```

```

#plot1_2 <- plot1_2[order(plot1_2$rank), ]
#points(plot1_2$rank, plot1_2$ab, type = "both", col = "coral2", pch = 16)

#plot1_3 <- data.frame(sp = colnames(dat[,4:31]), ab = as.numeric(dat[3,4:31]))

#plot1_3 <- plot1_3[which(plot1_3$ab!=0),]
#dim(plot1_3)
#plot1_3$rank <- rank(-plot1_3$ab, ties.method = "random")
#plot1_3 <- plot1_3[order(plot1_3$rank), ]
#points(plot1_3$rank, plot1_3$ab, type = "b",
# col = "coral3", pch = 16, lwd = 1)

#plot2_1 <- data.frame(sp = colnames(dat[,4:31]),
# ab = as.numeric(dat[4,4:31]))
#plot2_1 <- plot2_1[which(plot2_1$ab!=0),]
#dim(plot2_1)
#$rank <- rank(-plot2_1$ab, ties.method = "random")
#plot2_1 <- plot2_1[order(plot2_1$rank), ]
#points(plot2_1$rank, plot2_1$ab, type = "b",
# col = "aquamarine", pch = 16, lwd = 1)

#plot2_2 <- data.frame(sp = colnames(dat[,4:31]),
# ab = as.numeric(dat[5,4:31]))
#plot2_2 <- plot2_2[which(plot2_2$ab!=0),]
#dim(plot2_2)
#plot2_2$rank <- rank(-plot2_2$ab, ties.method = "random")
#plot2_2 <- plot2_2[order(plot2_2$rank), ]
#points(plot2_2$rank, plot2_2$ab, type = "b",
# col = "aquamarine3", pch = 16, lwd = 1)

#plot2_3 <- data.frame(sp = colnames(dat[,4:31]),
# ab = as.numeric(dat[6,4:31]))
#plot2_3 <- plot2_3[which(plot2_3$ab!=0),]
#dim(plot2_3)
#plot2_3$rank <- rank(-plot2_3$ab, ties.method = "random")
#plot2_3 <- plot2_3[order(plot2_3$rank), ]
#points(plot2_3$rank, plot2_3$ab, type = "b",
# col = "aquamarine4", pch = 16, lwd = 1)

#plot3_1 <- data.frame(sp = colnames(dat[,4:31]),
# ab = as.numeric(dat[7,4:31]))
#plot3_1 <- plot3_1[which(plot3_1$ab!=0),]
#dim(plot3_1)
#plot3_1$rank <- rank(-plot3_1$ab, ties.method = "random")
#plot3_1 <- plot3_1[order(plot3_1$rank), ]
#points(plot3_1$rank, plot3_1$ab, type = "b",
# col = "antiquewhite", pch = 16, lwd = 1)

#plot3_2 <- data.frame(sp = colnames(dat[,4:31]),
# ab = as.numeric(dat[8,4:31]))

```

```

#plot3_2 <- plot3_2[which(plot3_1$ab!=0),]
#(plot3_2)
#plot3_2$rank <- rank(-plot3_2$ab, ties.method = "random")
#plot3_2 <- plot3_2[order(plot3_2$rank), ]
#points(plot3_2$rank, plot3_2$ab, type = "b",
#   col = "antiquewhite3", pch = 16, lwd = 1)

#plot3_3 <- data.frame(sp = colnames(dat[,4:31]),
#   ab = as.numeric(dat[9,4:31]))
#plot3_3 <- plot3_3[which(plot3_1$ab!=0),]
#dim(plot3_3)
#plot3_3$rank <- rank(-plot3_3$ab, ties.method = "random")
#plot3_3 <- plot3_3[order(plot3_3$rank), ]
#points(plot3_3$rank, plot3_3$ab, type = "b",
#   col = "antiquewhite4", pch = 16, lwd = 1)

#plot4_1 <- data.frame(sp = colnames(dat[,4:31]),
#   ab = as.numeric(dat[10,4:31]))
#plot4_1 <- plot4_1[which(plot4_1$ab!=0),]
#dim(plot4_1)
#plot4_1$rank <- rank(-plot4_1$ab, ties.method = "random")
#plot4_1 <- plot4_1[order(plot4_1$rank), ]
#points(plot4_1$rank, plot4_1$ab, type = "b",
#   col = "brown1", pch = 16, lwd = 1)

#plot4_2 <- data.frame(sp = colnames(dat[,4:31]),
#   ab = as.numeric(dat[11,4:31]))
#plot4_2 <- plot4_2[which(plot4_2$ab!=0),]
#dim(plot4_2)
#plot4_2$rank <- rank(-plot4_2$ab, ties.method = "random")
#plot4_2 <- plot4_2[order(plot4_2$rank), ]
#points(plot4_2$rank, plot4_2$ab, type = "b",
#   col = "brown4", pch = 16, lwd = 1)

#plot4_3 <- data.frame(sp = colnames(dat[,4:31]),
#   ab = as.numeric(dat[12,4:31]))
#plot4_3 <- plot4_3[which(plot4_3$ab!=0),]
#dim(plot4_3)
#plot4_3$rank <- rank(-plot4_3$ab, ties.method = "random")
#plot4_3 <- plot4_3[order(plot4_3$rank), ]
#points(plot4_3$rank, plot4_3$ab, type = "b",
#   col = "brown2", pch = 16, lwd = 1)

#plot5_1 <- data.frame(sp = colnames(dat[,4:31]),
#   ab = as.numeric(dat[13,4:31]))
#plot5_1 <- plot5_1[which(plot5_1$ab!=0),]
#dim(plot5_1)
#plot5_1$rank <- rank(-plot5_1$ab, ties.method = "random")
#plot5_1 <- plot5_1[order(plot5_1$rank), ]
#points(plot5_1$rank, plot5_1$ab, type = "b",

```



```

#   col = "chartreuse", pch = 16, lwd = 1)

#plot5_2 <- data.frame(sp = colnames(dat[,4:31]),
#                      ab = as.numeric(dat[14,4:31]))
#plot5_2 <- plot5_2[which(plot5_2$ab!=0),]
#dim(plot5_2)
#plot5_2$rank <- rank(-plot5_2$ab, ties.method = "random")
#plot5_2 <- plot5_2[order(plot5_2$rank), ]
#points(plot5_2$rank, plot5_2$ab, type = "b",
#       col = "chartreuse3", pch = 16, lwd = 1)

#plot5_3 <- data.frame(sp = colnames(dat[,4:31]),
#                      ab = as.numeric(dat[15,4:31]))
#plot5_3 <- plot5_3[which(plot5_3$ab!=0),]
#dim(plot5_3)
#plot5_3$rank <- rank(-plot5_3$ab, ties.method = "random")
#plot5_3 <- plot5_3[order(plot5_3$rank), ]
#points(plot5_3$rank, plot5_3$ab, type = "b",
#       col = "chartreuse4", pch = 16, lwd = 1)

#plot6_1 <- data.frame(sp = colnames(dat[,4:31]),
#                      ab = as.numeric(dat[16,4:31]))
#plot6_1 <- plot6_1[which(plot6_1$ab!=0),]
#dim(plot6_1)
#plot6_1$rank <- rank(-plot6_1$ab, ties.method = "random")
#plot6_1 <- plot6_1[order(plot6_1$rank), ]
#points(plot6_1$rank, plot6_1$ab, type = "b",
#       col = "cadetblue", pch = 16, lwd = 1)

#plot6_2 <- data.frame(sp = colnames(dat[,4:31]),
#                      ab = as.numeric(dat[17,4:31]))
#plot6_2 <- plot6_2[which(plot6_2$ab!=0),]
#dim(plot6_2)
#plot6_2$rank <- rank(-plot6_2$ab, ties.method = "random")
#plot6_2 <- plot6_2[order(plot6_2$rank), ]
#points(plot6_2$rank, plot6_2$ab, type = "b",
#       col = "cadetblue3", pch = 16, lwd = 1)

#plot6_3 <- data.frame(sp = colnames(dat[,4:31]),
#                      ab = as.numeric(dat[18,4:31]))
#plot6_3 <- plot6_3[which(plot6_3$ab!=0),]
#dim(plot6_3)
#plot6_3$rank <- rank(-plot6_3$ab, ties.method = "random")
#plot6_3 <- plot6_3[order(plot6_3$rank), ]
#points(plot6_3$rank, plot6_3$ab, type = "b",
#       col = "cadetblue4", pch = 16, lwd = 1)

#legend( "topright",legend=c("1_1 (11 species)", "1_2 (12 species)", "1_3 (14 species)", "2_1 (7 specie

```

```

#datplot <- bind_rows(plot1_1,plot1_2,plot1_3)

#plot6_2$plot <- "plot6_2"

#plot1 <- cbind(plot1_1[,1:3], plot1_2[,1:3], plot1_3[,1:3])

#ggplot()+
# geom_line(aes(rank, ab, colour="plot1_1"), data=plot1_1)+
#geom_line(aes(rank, ab, colour="plot1_2"), data=plot1_2)+
#geom_line(aes(rank, ab, colour="plot1_3"), data=plot1_3)+
#geom_line(aes(rank, ab, colour="plot2_1"), data=plot2_1)+
#geom_line(aes(rank, ab, colour="plot2_2"), data=plot2_2)+
#geom_line(aes(rank, ab, colour="plot2_3"), data=plot2_3)+
#geom_line(aes(rank, ab, colour="plot3_1"), data=plot3_1)+
#geom_line(aes(rank, ab, colour="plot3_2"), data=plot3_2)+
#geom_line(aes(rank, ab, colour="plot3_3"), data=plot3_3)+
#geom_line(aes(rank, ab, colour="plot4_1"), data=plot4_1)+
#geom_line(aes(rank, ab, colour="plot4_2"), data=plot4_2)+
#geom_line(aes(rank, ab, colour="plot4_3"), data=plot4_3)+
#geom_line(aes(rank, ab, colour="plot5_1"), data=plot5_1)+
#geom_line(aes(rank, ab, colour="plot5_2"), data=plot5_2)+
#geom_line(aes(rank, ab, colour="plot5_3"), data=plot5_3)+
#geom_line(aes(rank, ab, colour="plot6_1"), data=plot6_1)+
#geom_line(aes(rank, ab, colour="plot6_2"), data=plot6_2)+
#geom_line(aes(rank, ab, colour="plot6_3"), data=plot6_3)+
# labs(y = "Abundance", x = "Rank", color = "")

#rank abundance curve per site
#dat$site<-aggregate(data=dat[,1:31],.~site, sum)

```

```

datp_NMDS <- metaMDS(jparks, k = 2)

```

```

## Run 0 stress 0.1891579
## Run 1 stress 0.2114898
## Run 2 stress 0.2278299
## Run 3 stress 0.1958146
## Run 4 stress 0.2275349
## Run 5 stress 0.2125581
## Run 6 stress 0.2288515
## Run 7 stress 0.2084861
## Run 8 stress 0.1958146
## Run 9 stress 0.1958146
## Run 10 stress 0.1891579
## ... New best solution
## ... Procrustes: rmse 0.0001650848  max resid 0.0003568055
## ... Similar to previous best
## Run 11 stress 0.218472
## Run 12 stress 0.2566182
## Run 13 stress 0.2097903
## Run 14 stress 0.1881943
## ... New best solution

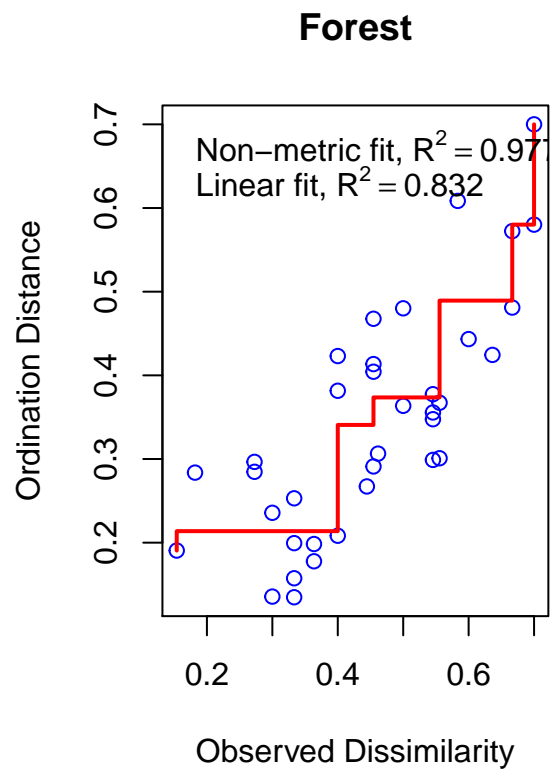
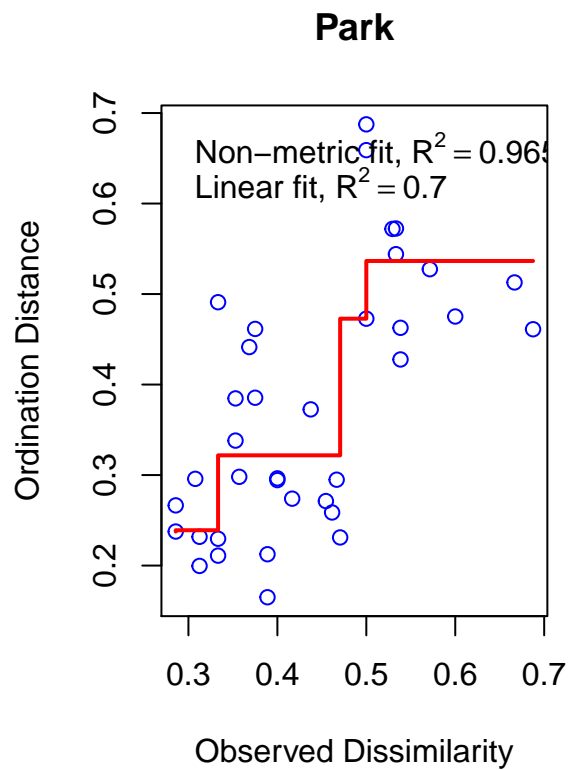
```

```
## ... Procrustes: rmse 0.1177804  max resid 0.2142411
## Run 15 stress 0.1888744
## Run 16 stress 0.221843
## Run 17 stress 0.2084861
## Run 18 stress 0.1958146
## Run 19 stress 0.2252626
## Run 20 stress 0.2811944
## *** No convergence -- monoMDS stopping criteria:
##      18: stress ratio > sratmax
##      2: scale factor of the gradient < sfgrmin
```

```
datf_NMDS <- metaMDS(jforest, k=2)
```

```
## Run 0 stress 0.174371
## Run 1 stress 0.1506224
## ... New best solution
## ... Procrustes: rmse 0.2244702  max resid 0.4691289
## Run 2 stress 0.1853256
## Run 3 stress 0.1794955
## Run 4 stress 0.2197233
## Run 5 stress 0.1794955
## Run 6 stress 0.1634615
## Run 7 stress 0.292948
## Run 8 stress 0.1587959
## Run 9 stress 0.1964856
## Run 10 stress 0.1506224
## ... New best solution
## ... Procrustes: rmse 0.0001961169  max resid 0.0003892772
## ... Similar to previous best
## Run 11 stress 0.1506224
## ... New best solution
## ... Procrustes: rmse 2.273825e-05  max resid 4.505792e-05
## ... Similar to previous best
## Run 12 stress 0.1676012
## Run 13 stress 0.1644232
## Run 14 stress 0.173818
## Run 15 stress 0.1865512
## Run 16 stress 0.1634615
## Run 17 stress 0.1759371
## Run 18 stress 0.1804467
## Run 19 stress 0.1634615
## Run 20 stress 0.1634615
## *** Solution reached
```

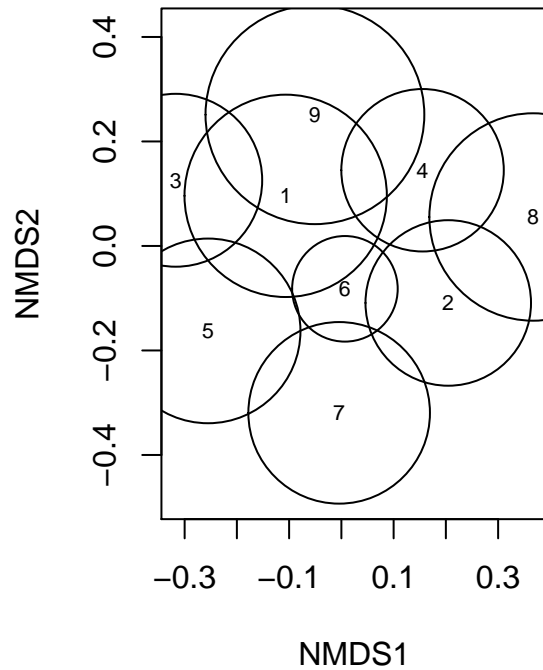
```
par(mfrow = c(1, 2))
stressplot(datp_NMDS, main = "Park")
stressplot(datf_NMDS, main = "Forest")
```



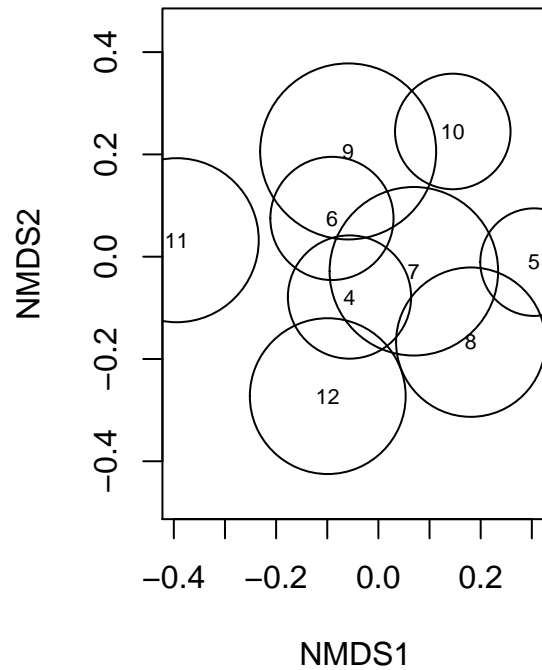
```
par(mfrow = c(1, 2))
plot(datp_NMDS, display = "sites", type = "t", main = "Goodness of fit")
points(datp_NMDS, display = "sites", cex = goodness(datp_NMDS)*200)
title(main = "Goodness of fit")

plot(datf_NMDS, display = "sites", type = "t", main = "Goodness of fit")
points(datf_NMDS, display = "sites", cex = goodness(datf_NMDS)*200)
title(main = "Goodness of fit")
```

Goodness of fit

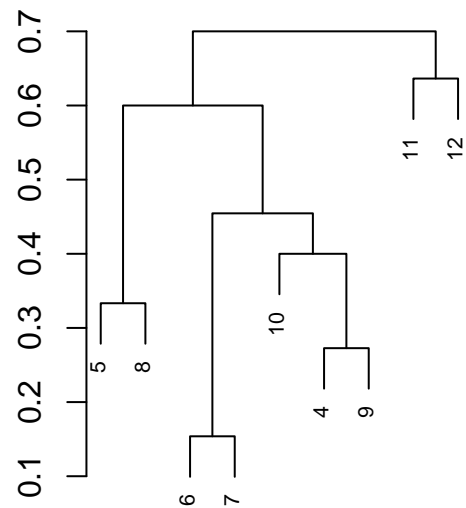
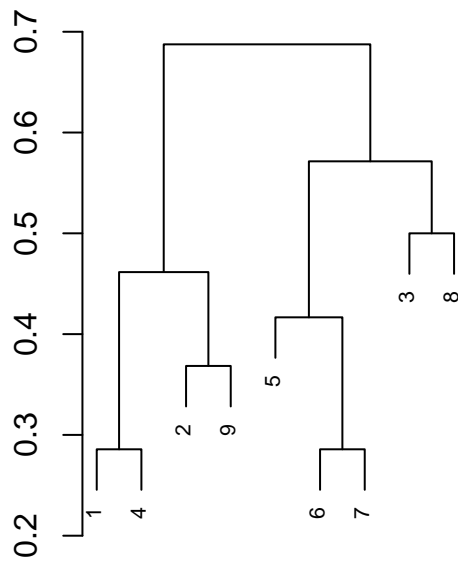


Goodness of fit



```
datpclust <- hclust(jparks, method = "complete")
plot(datpclust, xlab = "", cex = 0.7, ann = FALSE)
abline(h = 0.78, lwd = 2, col = "black", lty = 2)

datfclust <- hclust(jforest, method = "complete")
plot(datfclust, xlab = "", cex = 0.7, ann = FALSE)
abline(h = 0.78, lwd = 2, col = "black", lty = 2)
```



```

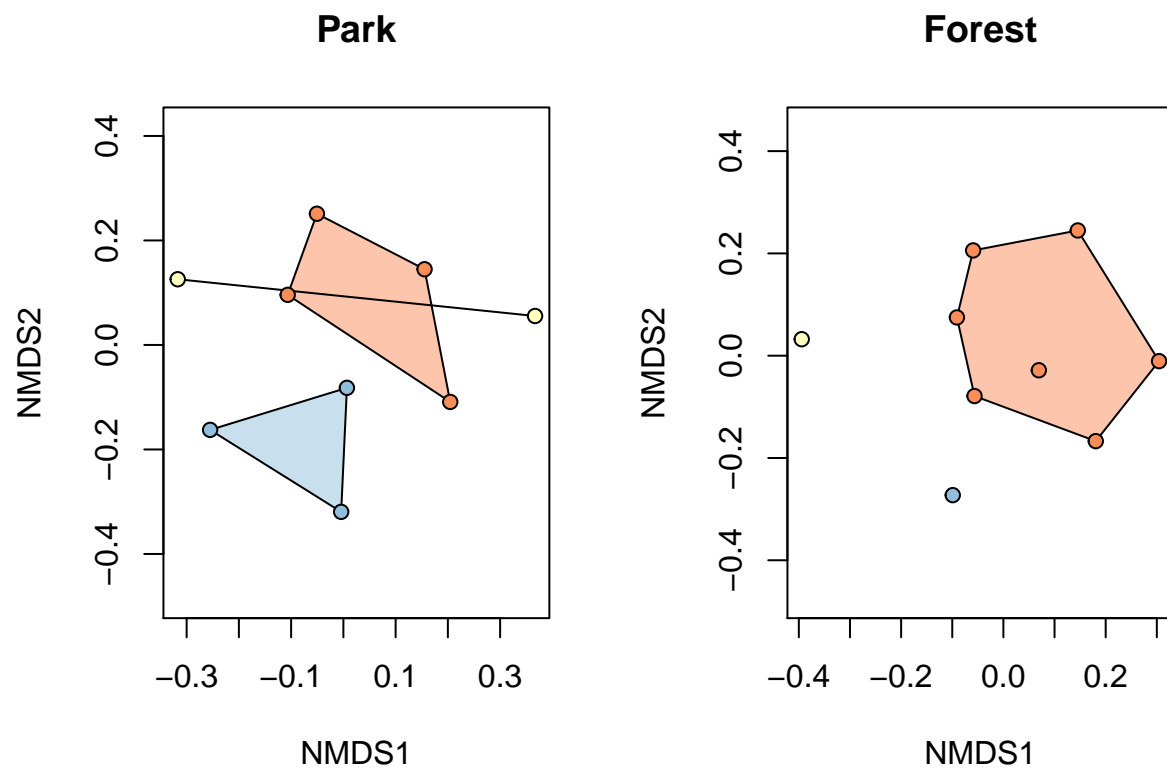
datpclust_3 <- cutree(datpclust, k = 3)

plot(datp_NMDS, display = "sites", type = "n", main = "Park")
ordihull(datp_NMDS, groups = datpclust_3,
         draw = "polygon", col = c("#fc8d59", "#ffffbf", "#91bfdb"),
         label = FALSE)
points(datp_NMDS, display = "sites", pch = 21, col = "black",
       bg = c("#fc8d59", "#ffffbf", "#91bfdb")[datpclust_3])
legend(x = 0.4, y = -0.2, legend = c(paste0("Cl. ", seq(1:3))),
      col = c("#fc8d59", "#ffffbf", "#91bfdb"), pch = 16)

datfclust_3 <- cutree(datfclust, k = 3)

plot(datf_NMDS, display = "sites", type = "n", main = "Forest")
ordihull(datf_NMDS, groups = datfclust_3,
         draw = "polygon", col = c("#fc8d59", "#ffffbf", "#91bfdb"),
         label = FALSE)
points(datf_NMDS, display = "sites", pch = 21, col = "black",
       bg = c("#fc8d59", "#ffffbf", "#91bfdb")[datfclust_3])
legend(x = 0.4, y = -0.2, legend = c(paste0("Cl. ", seq(1:3))),
      col = c("#fc8d59", "#ffffbf", "#91bfdb"), pch = 16)

```



Shannon diversity index

```
H <- diversity(dat[,4:31], index = "shannon")
plot(H~dat$site, xlab="Sampling plots", xaxt="n", ylab = "Shannon index")
axis(1, at=1:6, labels=c("KL", "CI", "WE", "SC", "CH", "CE"))
```

