BIOVEG - Statistical analyses in R

Ecio Souza Diniz and Jan Thiele

### A PROTOCOL TO LEARN AND TEACH STATISTICS IN R WITH VEGETATION DATA

contact: [eciodiniz@gmail.com](mailto:eciodiniz@gmail.com) and [jan.thiele@thuenen.de](mailto:jan.thiele@thuenen.de)

##### RECOMMENDATIONS

1- Complete an introductory R course to understand the basic operations used in the software: e.g. how to import files, attach data, install packages and create objects and graphs.

2- It is important to have a basic understanding of the principles and functions of the statistical analysis you wish to perform.

3- If your analysis does not execute successfully using these syntaxes, it is useful to consult R forums (e.g. Stack Overflow and Stack Exchange - Cross Validated) check if there is something wrong, missing or if this is simply not the appropriate analysis for your data.

4- Be sure about your choice of analysis. Refer to existing literature to help decide which analysis is most appropriate for your data.

5- If possible, it is highly recommended to read the references at the end of this script, especially those which are highlighted.

6- Try not to be overwhelmed by the various methodologies of statistical analysis which might be used to investigate the data. Again, consult with existing literature to help decide which analyses are most suitable.

7- Remember that this script was not created by statisticians. The adaptable nature of the R means that there are often several ways to complete the same analysis. This script describes one way to perform typical statistical analysis of the kind of data commonly generated in biological research.

### Content

NORMALITY TEST  
 - Shapiro Wilk  
DATA TRANSFORMATION   
 - Log, square, inverse and Arcsine square root  
ONE SAMPLE TESTS   
 - T-test and Wilcoxon Signed Rank Test  
TWO SAMPLE TESTS   
 - Independent samples: Wilcoxon Rank-Sum Test (Mann-Whitney U-test) and T-test  
 - Dependent samples: Wilcoxon Rank-Sum Test and Paired T-test  
ANALYSES OF VARIANCE  
 - Anova and Tukey post-hoc  
 - Kruskal-Wallis and Dunn’s test post-hoc  
INDEPENDENCE TEST   
 - Chi-squared  
CORRELATION ANALYSES   
 - Pearson, Spearman and VIF (Variance Inflation Factor)  
DIAGNOSTIC FOR LINEARITY  
 - component+Residual (Partial Residual) Plots  
REGRESSION MODELS   
 - LM (Linear Model)   
 - GLM (Generalized Linear Models   
 - GLMM (Generalized linear mixed models)  
 - GLMM-PQL (Fit GLMM using Penalized Quasi-Likelihood - PQL): spatial and temporal autocorrelation  
 - LOGISTIC REGRESSION  
 - SPATIAL AUTOCORRELATION TEST: Moran’s I  
 - LME (Linear mixed models)  
 - GLS (Generalized Least squares)  
 - QUADRATIC MODEL  
 - GNM (Generalized Nonlinear Models)  
 - GAM (Generalized Aditive Models)  
 - GAMM (Generalized Aditive Mixed Models)  
SUPERVISED LEARNING  
 - Random Forest  
MODELS SELECTION  
 - Akaike Criterion: AIC and QAIC  
MODEL AVERAGING  
 - Average of models selected by AIC or QAIC  
POST-HOC TEST TO REGRESSION MODELS  
 -Tukey’s HSD (honest significant difference)   
 - LmerTest (mixed models)  
MULTIVARIATE ANALYSIS  
 - NMDS, ANOSIM, CCA, PCA, PERMANOVA and INDVAL (Indicator species analysis)

#### Read test datasets

BIOVEG = your source (file) of data. Example:

setwd(“C:/”)

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## 5 Semidecidual p005 AREA1 Aspidosperma\_australe 9 34.31  
## 6 Semidecidual p006 AREA1 Aspidosperma\_olivaceum 7 16.03  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## 5 6.724 0.092 19 1250 24.4  
## 6 7.536 0.020 23 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## 5 2.35 1.02 -40 0 747923 7807727 747883  
## 6 2.90 5.74 0 -10 747923 7807727 747923  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727  
## 5 7807727  
## 6 7807717

Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## 5 s5 VOL ruderal.grass 2002 0.7 75 27  
## 6 s6 VOL woodland 2002 0.2 5 16  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## 5 0.8 20 NONE NONE 292.112 2  
## 6 0.8 10 NONE NONE 298.394 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34  
## 5 W 0.12 2.53 12.34  
## 6 NW 0.16 2.62 10.77

### NORMALITY TEST

One of the most commonly used tests in Botany is Shapiro-Wilk. This test verifies whether one specific variable from your data is normally (gaussian) distributed. If the p-value is below the critical level (e.g. below 0.05) the variable can be described as deviating significantly from the normal (gaussian) distribution.

# usage:  
shapiro.test(BIOVEG$Basal.area)

##   
## Shapiro-Wilk normality test  
##   
## data: BIOVEG$Basal.area  
## W = 0.50624, p-value < 2.2e-16

### DATA TRANSFORMATION

In some cases it is necessary to transform your data, either to achieve normality or to get a better fit for the residuals of your model. Here the mention some of the most commonly applied transformations, but there are many others (e.g. Box cox) and it’s always good to read about other possibilities to maximize the fit and robustness of your analysis.

#### plus:

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4) # reading only the first 4 rows

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

Basal.plus <- BIOVEG$Basal.area + 5  
plus<- cbind(BIOVEG$Basal.area, Basal.plus) # just to have a look at the result  
head(plus, 10) # reading only the first 10 rows of the outcome

## Basal.plus  
## [1,] 0.058 5.058  
## [2,] 0.120 5.120  
## [3,] 0.090 5.090  
## [4,] 0.128 5.128  
## [5,] 0.092 5.092  
## [6,] 0.020 5.020  
## [7,] 0.143 5.143  
## [8,] 0.139 5.139  
## [9,] 0.721 5.721  
## [10,] 0.107 5.107

#### log:

Mainly indicated to try to make right-skewed data normal and by default included in Poisson GLM (i.e. the default link function is log). But, many authors have suggested not to transform count (poisson) data. Read about it if possible.

Note: in case your data contain zeros or negative values, you have to add a constant to all values so that all of them are positive before you can take the logarithm.

Basal.log <- log(BIOVEG$Basal.area)   
c1<- cbind(BIOVEG$Basal.area, Basal.log)  
head(c1, 10) # reading only the first 10 rows of the outcome

## Basal.log  
## [1,] 0.058 -2.8473123  
## [2,] 0.120 -2.1202635  
## [3,] 0.090 -2.4079456  
## [4,] 0.128 -2.0557250  
## [5,] 0.092 -2.3859667  
## [6,] 0.020 -3.9120230  
## [7,] 0.143 -1.9449106  
## [8,] 0.139 -1.9732813  
## [9,] 0.721 -0.3271161  
## [10,] 0.107 -2.2349264

x <- rnorm(10, mean = 2, sd = 2) # example of a variable with values =< 0  
log(x) # does not work

## Warning in log(x): NaNs produced

## [1] NaN 1.2659236 0.4878708 1.1354343 NaN 0.8612742  
## [7] -0.6677834 NaN 1.3378603 1.4309497

c2 <- abs(min(x)) + 0.1 # the constant you have to add to x to make all values > 0  
head(c2, 10) #... 10 rows of the outcome

## [1] 1.567473

x.log <- log(x + c2) # works  
head (x.log, 10) # ...10 rows of the outcome

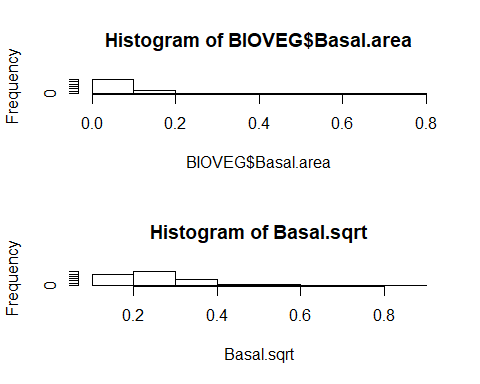
## [1] -2.3025851 1.6319505 1.1619993 1.5432977 0.1195393 1.3695669  
## [7] 0.7325203 -0.1414484 1.6823823 1.7492247

c3<- cbind(x, x.log)  
head(c3, 10) # ... 10 rows of the outcome

## x x.log  
## [1,] -1.4674729 -2.3025851  
## [2,] 3.5463668 1.6319505  
## [3,] 1.6288444 1.1619993  
## [4,] 3.1125251 1.5432977  
## [5,] -0.4404954 0.1195393  
## [6,] 2.3661737 1.3695669  
## [7,] 0.5128441 0.7325203  
## [8,] -0.6993730 -0.1414484  
## [9,] 3.8108808 1.6823823  
## [10,] 4.1826696 1.7492247

#### square:

Basal.sqrt <- sqrt(BIOVEG$Basal.area)  
par(mfrow=c(2,1)) # let's check the transformation using histograms  
hist(BIOVEG$Basal.area)  
hist(Basal.sqrt)



#### Inverse:

Basal.inv <- 1 / BIOVEG$Basal.area  
head(Basal.inv, 10) # reading only the first 10 rows

## [1] 17.241379 8.333333 11.111111 7.812500 10.869565 50.000000 6.993007  
## [8] 7.194245 1.386963 9.345794

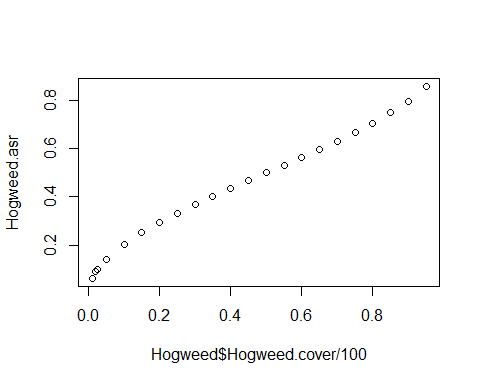
#### Arcsine square root:

Can be used for data that range between zero and 1. Often indicated for binomial data (proportions/ percentages), but it is preferable to use a model with binomial distribution rather than an arcsine square root transformation in this case.

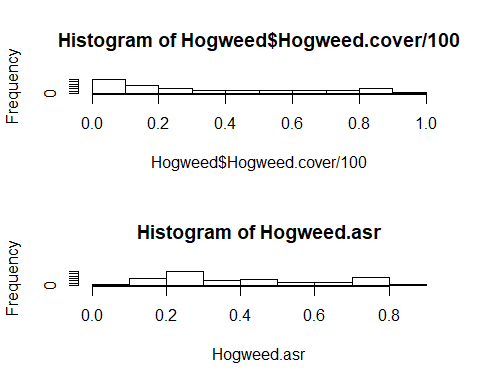
Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed, 4) # reading only the first 4 rows of data

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

Hogweed.asr <- asin(sqrt(Hogweed$Hogweed.cover/100))\*2/pi   
plot(Hogweed$Hogweed.cover/100, Hogweed.asr)



par(mfrow=c(2,1))  
hist(Hogweed$Hogweed.cover/100)  
hist(Hogweed.asr)



### ONE SAMPLE TEST

Indication: to test one sample mean against zero to verify whether there is a significant difference.

#### Parametric

#### T-TEST - One sample

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4) # reading only the first 4 rows

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

# Usage:  
  
t.test(BIOVEG$Basal.area, mu=0) # test against zero

##   
## One Sample t-test  
##   
## data: BIOVEG$Basal.area  
## t = 9.3933, df = 149, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 0.06618719 0.10145281  
## sample estimates:  
## mean of x   
## 0.08382

#### Nonparametric

#### Wilcoxon Signed Rank Test

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4) # reading only the first 4 rows

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

# Usage:  
  
wilcox.test(BIOVEG$Basal.area, mu=0)

##   
## Wilcoxon signed rank test with continuity correction  
##   
## data: BIOVEG$Basal.area  
## V = 11325, p-value < 2.2e-16  
## alternative hypothesis: true location is not equal to 0

### TWO SAMPLE TEST

Indication: to compare two independent samples.

#### Independent Two samples

#### Parametric: T-TEST - Two sample:

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4) # reading only the first 4 rows

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

# Testing whether vegetation type differ in tree basal area average  
  
# usage:   
  
t.test(BIOVEG$Basal.area[BIOVEG$type=="Aluvial"],   
 BIOVEG$Basal.area[BIOVEG$type=="Ombrofila"],alternative="two.sided")

##   
## Welch Two Sample t-test  
##   
## data: BIOVEG$Basal.area[BIOVEG$type == "Aluvial"] and BIOVEG$Basal.area[BIOVEG$type == "Ombrofila"]  
## t = -0.9419, df = 61.174, p-value = 0.3499  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.04934092 0.01774092  
## sample estimates:  
## mean of x mean of y   
## 0.0599 0.0757

# Testing whether a vegetation type has significantly less tree basal  
# than the other  
  
t.test(BIOVEG$Basal.area[BIOVEG$type=="Aluvial"],   
 BIOVEG$Basal.area[BIOVEG$type=="Ombrofila"], alternative="less",   
 var.equal=TRUE)

##   
## Two Sample t-test  
##   
## data: BIOVEG$Basal.area[BIOVEG$type == "Aluvial"] and BIOVEG$Basal.area[BIOVEG$type == "Ombrofila"]  
## t = -0.9419, df = 98, p-value = 0.1743  
## alternative hypothesis: true difference in means is less than 0  
## 95 percent confidence interval:  
## -Inf 0.01205505  
## sample estimates:  
## mean of x mean of y   
## 0.0599 0.0757

#### Dependent samples

Indication: to compare two dependent samples.

#### Parametric: PAIRED T-TEST

t.test(BIOVEG$Basal.area[BIOVEG$type=="Aluvial"],   
 BIOVEG$Basal.area[BIOVEG$type=="Ombrofila"],paired = TRUE)

##   
## Paired t-test  
##   
## data: BIOVEG$Basal.area[BIOVEG$type == "Aluvial"] and BIOVEG$Basal.area[BIOVEG$type == "Ombrofila"]  
## t = -0.92713, df = 49, p-value = 0.3584  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.05004692 0.01844692  
## sample estimates:  
## mean of the differences   
## -0.0158

#### Independent Two samples

#### Nonparametric: Wilcoxon Rank-Sum Test (or Mann-Whitney U-test)

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

Where y is numeric and A is a binary factor (presence or absence, 0 or 1, or similar) we can use the formula notation in wilcox.test

BIOVEG$Aluvial <- ifelse(BIOVEG$type=="Aluvial", 1, 0) # create binary factor  
wilcox.test(BIOVEG$Basal.area ~ BIOVEG$Aluvial)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: BIOVEG$Basal.area by BIOVEG$Aluvial  
## W = 2754, p-value = 0.3121  
## alternative hypothesis: true location shift is not equal to 0

independent 2-group Mann-Whitney U Test; here we specify the two groups to be compared using indexing (square brackets)

wilcox.test(BIOVEG$Basal.area[BIOVEG$type=="Aluvial"],   
 BIOVEG$Basal.area[BIOVEG$type=="Ombrofila"], paired=FALSE)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: BIOVEG$Basal.area[BIOVEG$type == "Aluvial"] and BIOVEG$Basal.area[BIOVEG$type == "Ombrofila"]  
## W = 1385.5, p-value = 0.3519  
## alternative hypothesis: true location shift is not equal to 0

#### Dependent (i.e. Paired) Two-Sample

#### Nonparametric: Wilcoxon Rank-Sum Test

Indication: the same function as the Wilcoxon test and two sample T-test, but addressing paired samples. This applies to data which has been collected on the same plots, for example, either on the same date or at different dates (e.g. comparison between basal area in 2010 and 2014. The basal area of 2014 in general is dependent on how much was accumulated in 2010).

Another test dataset describes the soil nutrient content of a number of different plots. The quantities of each nutrient (N, P, K) were derived from a single soil sample per plot.

Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed, 4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

When you have dependent samples, in “paired” insert TRUE. For independent samples insert FALSE.

# Usage:   
  
wilcox.test(Hogweed$P\_mg\_100g, Hogweed$K\_mg\_100g, paired=TRUE)

##   
## Wilcoxon signed rank test with continuity correction  
##   
## data: Hogweed$P\_mg\_100g and Hogweed$K\_mg\_100g  
## V = 891, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

### ANALYSIS OF VARIANCE

Indication: tests for differences between the means of more than two groups e.g. Diameter in treatment1, Diameter in treatment2, Diameter in treatment3).

Note: All Analyses below relate to variance, one-sample and two-sample. The majority of this analysis can be completed using the basic pre-loaded built-in “stats” package. To perform Dunn’s test (Post hoc), which follows Kruskal-Wallis,the “dunn.test” package must be installed.

#### Parametric tests: One-Way Anova

Note 2: transformations of the dependent variable, e.g. log, can be specified in the syntax of ANOVA and other functions directly.

# usage:  
  
BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

aov(log(Diameter) ~ type, data=BIOVEG)

## Call:  
## aov(formula = log(Diameter) ~ type, data = BIOVEG)  
##   
## Terms:  
## type Residuals  
## Sum of Squares 1.590824 22.039904  
## Deg. of Freedom 2 147  
##   
## Residual standard error: 0.3872097  
## Estimated effects may be unbalanced

results <- aov(log(Diameter) ~ type, data=BIOVEG)  
summary(results)

## Df Sum Sq Mean Sq F value Pr(>F)   
## type 2 1.591 0.7954 5.305 0.00596 \*\*  
## Residuals 147 22.040 0.1499   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

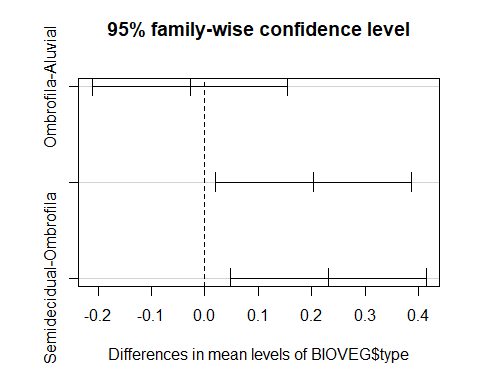
#### Tukey test (post hoc)

Indication: to identify when there is a significant difference between categories previously shown as overall result by parametric ANOVA.

# Usage:  
  
a1 <- aov(log(BIOVEG$Diameter) ~ BIOVEG$type)  
posthoc <- TukeyHSD(x=a1, 'BIOVEG$type', conf.level=0.95)  
print(posthoc)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = log(BIOVEG$Diameter) ~ BIOVEG$type)  
##   
## $`BIOVEG$type`  
## diff lwr upr p adj  
## Ombrofila-Aluvial -0.0280405 -0.21139925 0.1553182 0.9303204  
## Semidecidual-Aluvial 0.2030858 0.01972702 0.3864445 0.0259703  
## Semidecidual-Ombrofila 0.2311263 0.04776753 0.4144850 0.0092752

plot(posthoc)



#### Non-Parametric tests: Kruskal-Wallis

Indication: use following a significant result in the Kruskal-Wallace test (a non-parametric analysis, the underlying distribution is not assumed in advance).

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

#Usage:  
  
r4<- kruskal.test(BIOVEG$Basal.area ~ BIOVEG$type)  
r4

##   
## Kruskal-Wallis rank sum test  
##   
## data: BIOVEG$Basal.area by BIOVEG$type  
## Kruskal-Wallis chi-squared = 10.899, df = 2, p-value = 0.004299

#### Dunn’s Test (post hoc)

It performs the same function as a TukeyHSD test following ANOVA, but here is following Kruskal-Wallis.

install.packages(“dunn.test”)

# Usage:  
  
library(dunn.test)  
  
dunn.test(BIOVEG$Basal.area, g=BIOVEG$type, kw=TRUE, label=TRUE, wrap=TRUE,   
 table=TRUE, list=FALSE, rmc=FALSE, alpha=0.05)

## Kruskal-Wallis rank sum test  
##   
## data: x and group  
## Kruskal-Wallis chi-squared = 10.8989, df = 2, p-value = 0  
##   
##   
## Comparison of x by group   
## (No adjustment)   
## Col Mean-|  
## Row Mean | Aluvial Ombrofil  
## ---------+----------------------  
## Ombrofil | 0.694041  
## | 0.2438  
## |  
## Semideci | -2.448136 -3.142178  
## | 0.0072\* 0.0008\*  
##   
## alpha = 0.05  
## Reject Ho if p <= alpha/2

# when the command line extends beyond more than one row in the text, be sure to select all rows to run the complete command.

### Independence test - Chi-squared

The Chi-squared test is one of the most commonly used tests in biology and biomedicine because it can test if one variable is dependent or not on a certain category. In the example below we use the phenology stages of a species and the abundance of individuals counted at each stage over 4 months (Feb, Mar, Apr, May). Our aim is to check if the abundances at each stage are dependent on the months. In case Chi-squared is significant and, thus, shows dependence, you need to split the table into two in a 2 x 2 format and run the test again. This post-hoc test is called: Partition of Chi-squared. Further Information about this test is widely available online.

install and load the MASS package:

# install.packages("MASS")  
library(MASS)

To build a contingency from columns of a data table you can use. For example:

# tbl <- table(Data$flowering, Data$fruits, Data$Month)

Here, we will just create a contingency table “manually”:

tbl <- rbind(c(7,1,3,9), c(8,5,2,0))  
colnames(tbl) <- c("Feb", "Mar", "Apr", "May")   
rownames(tbl) <- c("Flowering", "Fruits")  
tbl

## Feb Mar Apr May  
## Flowering 7 1 3 9  
## Fruits 8 5 2 0

# Usage:   
  
chisq.test(tbl) # normal p-value of Chi-squared test

## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: tbl  
## X-squared = 11.453, df = 3, p-value = 0.009513

chisq.test(tbl, simulate.p.value=TRUE) # simulated p-value (Monte Carlo test)

##   
## Pearson's Chi-squared test with simulated p-value (based on 2000  
## replicates)  
##   
## data: tbl  
## X-squared = 11.453, df = NA, p-value = 0.004498

The outcome showed a message about the robustness of the Chi-squared appproximation. Thus, first of all be sure about the quality of the dataset you are using for such analyses.

# "Posthoc" test comparing only two month  
  
tbl.1 <- tbl[,c("Mar", "Apr")]  
tbl.1

## Mar Apr  
## Flowering 1 3  
## Fruits 5 2

chisq.test(tbl.1)

## Warning in chisq.test(tbl.1): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: tbl.1  
## X-squared = 0.73661, df = 1, p-value = 0.3907

### CORRELATION ANALYSIS

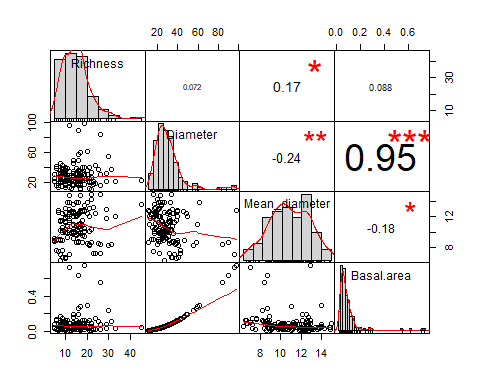
#### Pearson’s correlation test

use PerformanceAnalytics package

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

# Usage:  
  
# install.packages("PerformanceAnalytics")  
library(PerformanceAnalytics)  
chart.Correlation(BIOVEG[,c(5, 6, 7, 8)])



correlacao<- cor(BIOVEG[,c(5, 6, 7, 8)])   
round(correlacao, digits=2)

## Richness Diameter Mean\_diameter Basal.area  
## Richness 1.00 0.07 0.17 0.09  
## Diameter 0.07 1.00 -0.24 0.95  
## Mean\_diameter 0.17 -0.24 1.00 -0.18  
## Basal.area 0.09 0.95 -0.18 1.00

Correlation test between just two variables:

## use the default R function "cor.test"  
  
cor.test(BIOVEG$Richness, BIOVEG$Basal.area)

##   
## Pearson's product-moment correlation  
##   
## data: BIOVEG$Richness and BIOVEG$Basal.area  
## t = 1.0703, df = 148, p-value = 0.2862  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.07365322 0.24447057  
## sample estimates:  
## cor   
## 0.08764283

The number between parentheses are the column numbers of your data file (e.g. Excel file). This is how you select the variables whose correlation you want to check.

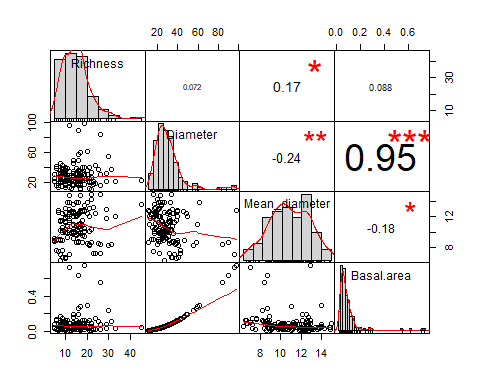
REMEMBER: correlation just shows you if one data value increases/ decreases with the other, but it does not say that they are dependent on each other. To check dependence between variables you need to test their relation and for that there are regression models. However, even regression models do not prove that there is a causal relationships between the variables. Correlation and regression cannot discern between real causal effects and spurious correlations (that are caused by a third variable/ other variables).

#### Spearman’s correlation test

There is a perpetual discussion among statisticians about whetherPearson analysis assumes strictly normality, but a large majority of high impact literature continues to use it, even in non-normal cases. Some researchers recommend Spearman, but, for a number of reasons, its application is more limited than Pearson. Please read about this subject further if it applies to your analysis.

If you want to use Spearman rank correlation just use the ‘method’ argument:

chart.Correlation(BIOVEG[,c(5, 6, 7, 8)], method="spearman")



correlacao<- cor(BIOVEG[,c(5, 6, 7, 8)], method="spearman")   
round(correlacao, digits=2)

## Richness Diameter Mean\_diameter Basal.area  
## Richness 1.00 0.00 0.16 0.00  
## Diameter 0.00 1.00 -0.29 1.00  
## Mean\_diameter 0.16 -0.29 1.00 -0.29  
## Basal.area 0.00 1.00 -0.29 1.00

Correlation test between just two variables:

## use the default R function "cor.test"  
  
cor.test(BIOVEG$Richness, BIOVEG$Basal.area, method="spearman")

## Warning in cor.test.default(BIOVEG$Richness, BIOVEG$Basal.area, method =  
## "spearman"): Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: BIOVEG$Richness and BIOVEG$Basal.area  
## S = 563580, p-value = 0.981  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## -0.00196546

Sometimes Spearman ranking is not able to precisely compute p-values. Thus, an option is also test Pearson and provide the results of both.

#### VIF - Variance Inflation Factor

Among the most robust approaches used nowadays, the variance inflation factor (VIF) is the quotient of the variance in a model with multiple terms (predictors) by the variance of a model with one term alone. This is a great method to quantify the severity of multicollinearity in an ordinary least squares regression analysis (e.g. LM, LME, GLS, etc). VIF provides an index that returns an outcome measuring how much the variance (the square of the estimate’s standard deviation) of an estimated model coefficient is increased because of collinearity.

In Ecology, an usual threshold to consider collinearity among predictors acceptable is VIF < 0.5

For further details and information on the statistical backbone and potential of VIF, please read the great revision and simulations of Dormann et al. 2013 and Zurr et al.2010.

Dormann et al. Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. Ecography, 36(1): 27-46, 2013.

Zuur AF, Ieno EN & Elphick CS. A protocol for data exploration to avoid common statistical problems. Methods in Ecology and Evolution 1: 3-14, 2010.

We use the function vif of the package car to calculate the variance inflation factor:

# install.packages("car")  
library(car)  
  
  
BIODIV.vif<- lm(Species.richness ~ P\_mg\_100g + Hogweed.cover + N\_perc +  
 Habitat.type, data=Hogweed)  
  
vif(BIODIV.vif)

## GVIF Df GVIF^(1/(2\*Df))  
## P\_mg\_100g 1.093411 1 1.045663  
## Hogweed.cover 1.293545 1 1.137341  
## N\_perc 1.122377 1 1.059423  
## Habitat.type 1.434640 4 1.046147

The outcome shows that all predictors have VIF < 5, thus they can be kept together in the same model.

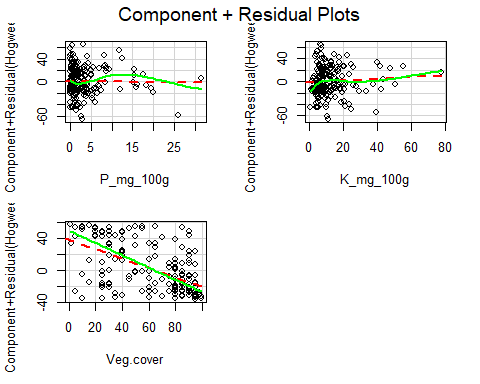
### Diagnostic for linearity

To check for linear relationships among dependent (Y) and predictor variables (X) we can use the function “crPlots” of the package “car”. This function returns a graph that bases the diagnostic on the residuals of the model. The more closely the two lines (solid and dashed) fit each other, the more linear is the relationship among Y and X.

library(car) # graphical diagnostic for linearity  
  
Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

DBH<- glm(Hogweed.cover ~ P\_mg\_100g + K\_mg\_100g + Veg.cover, data=Hogweed,   
 family="gaussian")  
  
crPlots(DBH, col.lines=c("red", "green"))



In the diagnostic we can see that the predictor “Veg.cover” shows the best, and an acceptable, fit for linearity. This is indicated by the close fit between the red line (best fit) and green line. Conversely, the linearity of “P\_mg\_100g” and “K\_mg\_100g” deviates considerably.

### REGRESSION

Indication: Regression analysis is a statistical approach for estimating the relationships between a dependent variable (axis Y and commonly called the outcome variable’) and one or more independent variables (predictors).There are linear and nonlinear regression models, although the linear ones are the most common form of used regression analysis.

#### LM (Linear Model)

This is the most famous type of regression of analysis and relies in the assumptions of the Ordinary least squares: normal distribution, homoscedasticity, linearity between, predictors and dependent variable, weak multicollinearity among predictors, independence of errors and residuals not correlated.

Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

# install.packages("car")  
library(car) # Anova function

# Usage:   
# Model:  
  
BIODIV.pre<- lm(Species.richness ~ P\_mg\_100g + Habitat.type, data=Hogweed)  
  
summary(BIODIV.pre)

##   
## Call:  
## lm(formula = Species.richness ~ P\_mg\_100g + Habitat.type, data = Hogweed)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16.117 -5.058 -1.152 4.912 25.871   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 25.1319 1.2887 19.503 < 2e-16 \*\*\*  
## P\_mg\_100g -0.2768 0.1170 -2.366 0.018946 \*   
## Habitat.typeruderal.grass -2.2314 1.6361 -1.364 0.174184   
## Habitat.typetall.herbs -9.4813 1.5449 -6.137 4.6e-09 \*\*\*  
## Habitat.typewasteland -2.2797 2.2731 -1.003 0.317153   
## Habitat.typewoodland -7.8197 2.1460 -3.644 0.000344 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.564 on 195 degrees of freedom  
## Multiple R-squared: 0.2505, Adjusted R-squared: 0.2313   
## F-statistic: 13.04 on 5 and 195 DF, p-value: 5.965e-11

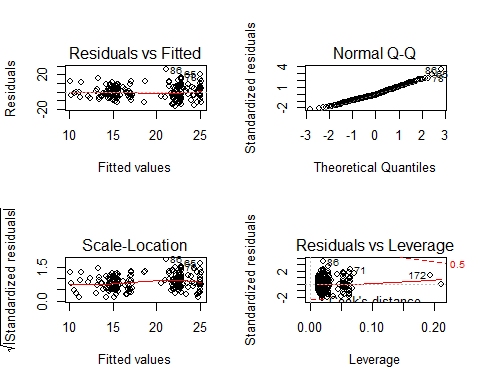
# Anova of the model using the package car:  
Anova(BIODIV.pre)

## Anova Table (Type II tests)  
##   
## Response: Species.richness  
## Sum Sq Df F value Pr(>F)   
## P\_mg\_100g 320.4 1 5.5995 0.01895 \*   
## Habitat.type 3060.2 4 13.3714 1.199e-09 \*\*\*  
## Residuals 11156.9 195   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

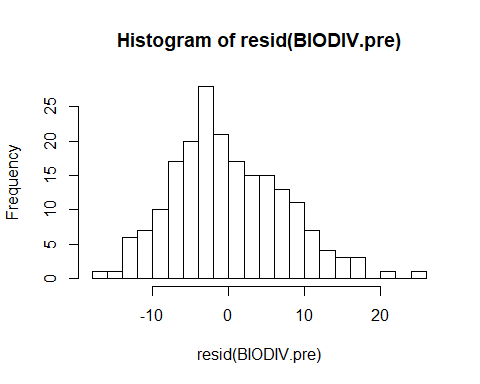
# Anova of the model using the built-in R function:  
BIODIV.lm.aov <- anova(BIODIV.pre, test="F")  
BIODIV.lm.aov

## Analysis of Variance Table  
##   
## Response: Species.richness  
## Df Sum Sq Mean Sq F value Pr(>F)   
## P\_mg\_100g 1 668.9 668.94 11.692 0.0007644 \*\*\*  
## Habitat.type 4 3060.2 765.04 13.371 1.199e-09 \*\*\*  
## Residuals 195 11156.9 57.21   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

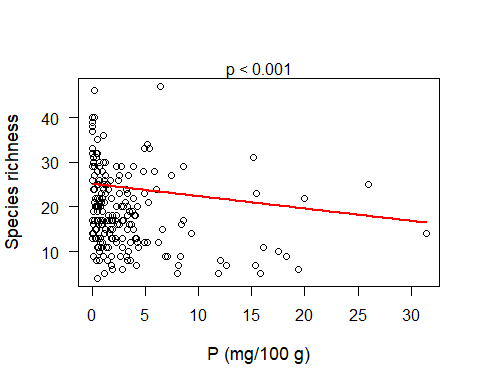
# Residuals  
x11(width=12, height=12)  
par(mfrow=c(2,2))  
plot(BIODIV.pre)



hist(resid(BIODIV.pre),breaks=20)



# Scatterplot graph:   
  
b.0.BIODIV<- BIODIV.pre$coefficients[1]  
b.1.BIODIV<- BIODIV.pre$coefficients[2]  
plot(Hogweed$Species.richness ~ Hogweed$P\_mg\_100g, las=1,   
 ylab="Species richness", xlab="P (mg/100 g)", cex.lab=1.1)  
curve(b.0.BIODIV + b.1.BIODIV\*x, add=TRUE, col="red", lwd=2)  
mtext(ifelse(BIODIV.lm.aov[1,5]<0.001, "p < 0.001",   
 paste("p = ", round(BIODIV.lm.aov[1,5],3))))



To construct the curve (or line) in this LM example, the dependent variable, Species richness, is on the y axis and the metric predictor variable, phosphorous, is on the x axis. The regression line is calculated using the intercept (b.0.BIODIV) plus the estimate of phosphorous (b.1.BIODIV) multiplied by x. The insertion of the P-value of ANOVA is placed according to the location of values in the ANOVA output. That is, the “P\_mg\_100g” is being tested in relation to “Species.richness” and it’s located in the first row of the ANOVA output and its P-value on the 5th column in this row. Thus: BIODIV.lm.aov[1,5]. If you have done any data transformation before (log, square …) you must include it in the syntax of the scatterplot, e.g. log(Hogweed$Species.richness). If the variable include zeros, add a constant, e.g. +1, before taking the log.

#### GLM (Generalized Linear Models)

Note: in GLM you must specify the distribution family of the dependent variable. Thus:

* Gaussian (normal) distribution: Continuous data (e.g. body weight, basal area, height). If the residuals don’t satisfactorily fit the Gaussian distribution, you can try Gamma distribution. BUT, first of all read about both kinds of distribution before you decide which to use . Gamma doesn’t accept negative values.
* Poisson: count data (e.g. species richnees). First check if the Poisson model shows overdispersion. If it does, try “quasipoisson” distribution.
* Binomial: this distribution is often used to model the number of successes

1. or failure (0), but also presence (1) or absence (0), in a sample. Furthermore, data regarding percentage or proportion in a sample.

* Negative BinomiaL: count data in case of overdispersion in “poisson” or “quasipoisson”.

A Poisson or Binomial distribution model might be likely to show under- or overdispersion, indicating that your model fit is not satisfactory. After running the “summary”, check for evidence of overdispersion. You do this by checking residuals and degrees of freedom: residual deviance/degrees of freedom << 1 (e.g. < 0.7) indicate underdispersion, while residual deviance/degrees of freedom >> 1 (e.g. > 1.3) indicate overdispersion. In such cases, you need to use “quasipoisson” or “quasibinomial” distributions instead. The model will then estimate a dispersion parameter that adjusts the variation in the data to the fixed variation of the Poisson or Binomial distribution. If you are analyzing count data (integers), you can also use the Negative Binomial distribution instead of “quasipoisson”. To use Negative Binomial distribution, you use the function “glm.nb”, which is the function for negative binomial glm from the package MASS.

To use Negative Binomial distribution, you just insert like this example: “glm.nb” is the function for negative binomial glm from the package MASS.

# install.packages(MASS)  
library(MASS) # For Negative Binomial GLM  
  
Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

riqueza.glm<- glm.nb(Species.richness ~ P\_mg\_100g + Land.use, data=Hogweed)   
summary(riqueza.glm)

##   
## Call:  
## glm.nb(formula = Species.richness ~ P\_mg\_100g + Land.use, data = Hogweed,   
## init.theta = 7.646689121, link = log)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5831 -0.8338 -0.1544 0.5509 2.9755   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.258909 0.085480 38.125 < 2e-16 \*\*\*  
## P\_mg\_100g -0.018549 0.006899 -2.689 0.007176 \*\*   
## Land.useMAINTENANCE -0.182235 0.111953 -1.628 0.103570   
## Land.useNONE -0.322205 0.092300 -3.491 0.000482 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(7.6467) family taken to be 1)  
##   
## Null deviance: 230.83 on 200 degrees of freedom  
## Residual deviance: 206.28 on 197 degrees of freedom  
## AIC: 1402.9  
##   
## Number of Fisher Scoring iterations: 1  
##   
##   
## Theta: 7.65   
## Std. Err.: 1.08   
##   
## 2 x log-likelihood: -1392.889

Anova(riqueza.glm)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: Species.richness  
## LR Chisq Df Pr(>Chisq)   
## P\_mg\_100g 7.676 1 0.005596 \*\*  
## Land.use 13.723 2 0.001047 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### GLM - Poisson:  
  
riqueza.glm<- glm(Species.richness ~ P\_mg\_100g + Land.use, data=Hogweed,  
 family="poisson")  
summary(riqueza.glm)

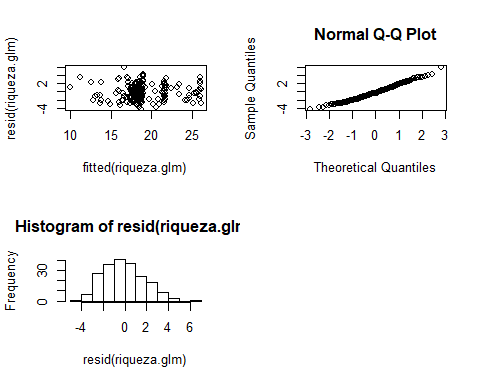
##   
## Call:  
## glm(formula = Species.richness ~ P\_mg\_100g + Land.use, family = "poisson",   
## data = Hogweed)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.1479 -1.5324 -0.3102 1.0464 6.0723   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.263022 0.041535 78.562 < 2e-16 \*\*\*  
## P\_mg\_100g -0.020470 0.004104 -4.988 6.11e-07 \*\*\*  
## Land.useMAINTENANCE -0.181567 0.055790 -3.254 0.00114 \*\*   
## Land.useNONE -0.320027 0.045601 -7.018 2.25e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 775.16 on 200 degrees of freedom  
## Residual deviance: 686.47 on 197 degrees of freedom  
## AIC: 1636.8  
##   
## Number of Fisher Scoring iterations: 4

library(car)   
riqueza.aov <- Anova(riqueza.glm)  
riqueza.aov

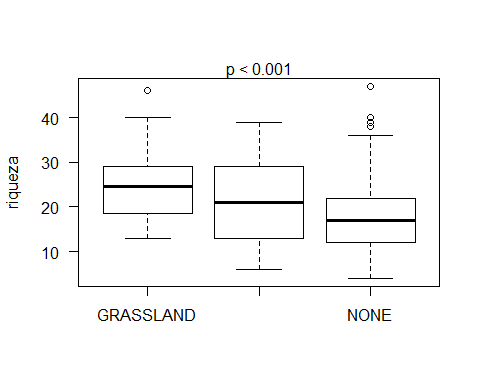
## Analysis of Deviance Table (Type II tests)  
##   
## Response: Species.richness  
## LR Chisq Df Pr(>Chisq)   
## P\_mg\_100g 27.195 1 1.839e-07 \*\*\*  
## Land.use 49.816 2 1.523e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Checking the residuals and see if it does fit good to the model

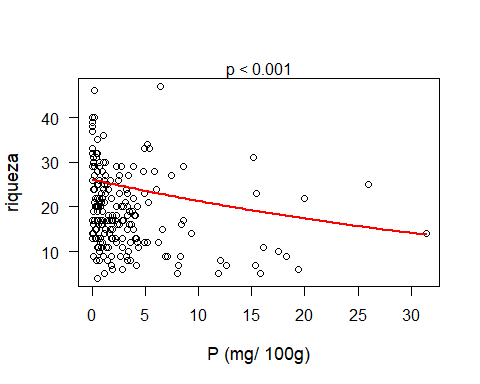
# residuals  
  
par(mfrow=c(2,2))  
plot(fitted(riqueza.glm), resid(riqueza.glm))  
qqnorm(resid(riqueza.glm))  
hist(resid(riqueza.glm))



# Boxplot  
  
par(mfrow=c(1,1))  
boxplot(Hogweed$Species.richness ~ Hogweed$Land.use, las=1, ylab="riqueza")  
mtext(ifelse(riqueza.aov[2,3]<0.001, "p < 0.001",   
paste("p = ", round(riqueza.aov[2,3],3))))



# Scatterplot graph:   
  
b.0.riqueza<- riqueza.glm$coefficients[1]  
b.1.riqueza<- riqueza.glm$coefficients[2]  
plot(Hogweed$Species.richness ~ Hogweed$P\_mg\_100g, las=1, ylab="riqueza",   
 xlab="P (mg/ 100g)", cex.lab=1.1)  
curve(exp(b.0.riqueza + b.1.riqueza\*x), add=TRUE, col="red", lwd=2)  
mtext(ifelse(riqueza.aov[1,3]<0.001, "p < 0.001",   
 paste("p = ", round(riqueza.aov[1,3],3))))



Note 3.: the graph construction follows the principles described with LM previously. If you have used a transformation or a link function (other than “=”) in the model, you can include it in the scatterplot syntax, e.g.log (Species.richness). But if you want to plot the original values of the dependent variable, you can do that, too, you just need to back-transform the model predictions to the original scale. In this case you need to transform from the log-scale (used in the Poisson model) to the untransformed species richness used in the scatterplot. That’s why we need the “exp” function in the example above.

#### GLMM (Generalized Linear Mixed Models)

Note 1: Like GLM, the GLMM also require to specify the distribution family (e.g. Poisson), but they don’t accept “quasipoisson” or “quasibinomial”. If poisson doesn’t fit the data because of overdispersion, try negative binomial distribution as explained in GLM section. The syntax is the same: just insert the argument “nb”. Example: basalarea1<- glmer.nb(…, Do not write family name).

GLMM indication: when you have pseudoreplicates. It’s very common in ecological field research to use subplots within each site and these subplots cannot be considered as real replicates. So, a mixed model is appropriate to treat this situation. The definition of what variable is a fixed effect (e.g. in the example below it is “Hogweed.cover” (cover of giant hogweed) and “Habitat.type”) and what is random effect (e.g. in the example below “Study.area”) depends on your research questions and study design. Usually, design factors, such as “block” or “study area” will be random effects, and the variables that you are really interested in (treatments, soil parameters etc.) will be fixed effects. BUT, read about random and fixed effects before constructing your model.

When the command line doesn’t end in the same row, but continues in the next one (like the first command of the GLMM example below), select and press the command of both two lines together. We broke some command lines here to fit in the R script style.

### GLMM usage:  
  
Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

library(lme4) # glmer

## Loading required package: Matrix

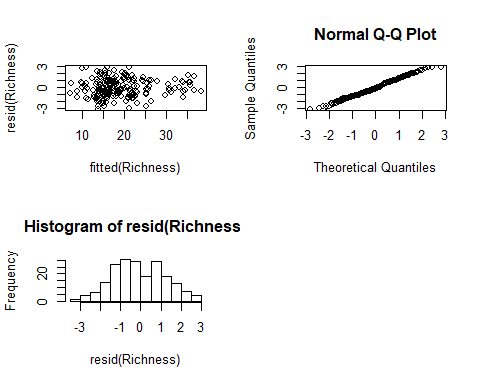
library(car) # Anova  
  
Richness <- glmer(Species.richness ~ scale(Hogweed.cover) + Habitat.type +   
 (1|Study.area), data=Hogweed, family="poisson")  
# Here, the metric predictor variable (Hogweed.cover) was scaled to avoid   
# problems in the calculation of the model  
  
summary(Richness)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: poisson ( log )  
## Formula: Species.richness ~ scale(Hogweed.cover) + Habitat.type + (1 |   
## Study.area)  
## Data: Hogweed  
##   
## AIC BIC logLik deviance df.resid   
## 1352.4 1375.5 -669.2 1338.4 194   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.6389 -0.9970 -0.1876 0.9007 3.3432   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## Study.area (Intercept) 0.08313 0.2883   
## Number of obs: 201, groups: Study.area, 20  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.17346 0.07660 41.429 < 2e-16 \*\*\*  
## scale(Hogweed.cover) -0.07988 0.02052 -3.892 9.94e-05 \*\*\*  
## Habitat.typeruderal.grass -0.13191 0.05530 -2.385 0.0171 \*   
## Habitat.typetall.herbs -0.41489 0.05828 -7.118 1.09e-12 \*\*\*  
## Habitat.typewasteland -0.18083 0.07605 -2.378 0.0174 \*   
## Habitat.typewoodland -0.50020 0.07356 -6.800 1.05e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) sc(H.) Hbtt.typr. Hbtt.typt. Hbtt.typws  
## scl(Hgwd.c) 0.183   
## Hbtt.typrd. -0.424 -0.342   
## Hbtt.typtl. -0.415 -0.394 0.567   
## Hbtt.typwst -0.279 0.057 0.416 0.316   
## Hbtt.typwdl -0.281 -0.042 0.354 0.412 0.269

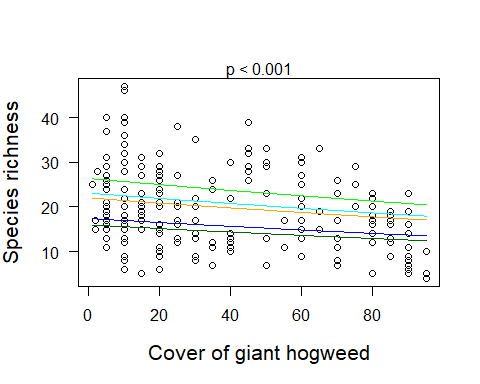
Richness.aov <- Anova(Richness, Type="II")   
Richness.aov

## Analysis of Deviance Table (Type II Wald chisquare tests)  
##   
## Response: Species.richness  
## Chisq Df Pr(>Chisq)   
## scale(Hogweed.cover) 15.149 1 9.937e-05 \*\*\*  
## Habitat.type 76.133 4 1.147e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Residuals  
  
par(mfrow=c(2,2))  
plot(fitted(Richness), resid(Richness))  
qqnorm(resid(Richness))  
qqline(resid(Richness))  
hist(resid(Richness))



### Scatterplots ###   
  
# With the syntax below you can construct graphs of the relation of the depenent   
# variable with a metric predictor variable. This can be done for all levels of   
# the categorical predictor variable (here: Habitat.type). To construct the graph   
# you can just select and run all of the command lines below.   
  
b0<- coef(summary(Richness))[1,1] # intercept (habitat "agricultural grassland")  
b1<- coef(summary(Richness))[2,1] # estimate of hogweed cover  
b2<- coef(summary(Richness))[3,1] # estimate of habitat "ruderal grassland"  
b3<- coef(summary(Richness))[4,1] # estimate of habitat "tall herbs"  
b4<- coef(summary(Richness))[5,1] # estimate of habitat "wasteland"  
b5<- coef(summary(Richness))[6,1] # estimate of habitat "woodland"  
  
x11(width=14, height=12)  
  
plot(Hogweed$Species.richness ~ Hogweed$Hogweed.cover, las=1,   
 ylab="Species richness", xlab="Cover of giant hogweed", cex.lab=1.3)   
  
  
# The model that we apply here used poisson distribution which, by default,   
# includes a log-link (i.e. log transformation of the dependent variable).   
# Thus, we need to use exp() to back-transform the values predicted by the model,   
# to the original values of species richness.   
  
  
mean <- mean(Hogweed$Hogweed.cover)  
sd <- sd(Hogweed$Hogweed.cover)  
  
curve(exp(b0 + b1\*(x-mean)/sd), add=TRUE, col="green") # curve for agricultural grassland  
curve(exp(b0 + b2 + + b1\*(x-mean)/sd), add=TRUE, col="cyan") # curve for ruderal grassland  
curve(exp(b0 + b3 + + b1\*(x-mean)/sd), add=TRUE, col="blue") # curve for tall-herb stands  
curve(exp(b0 + b4 + + b1\*(x-mean)/sd), add=TRUE, col="orange") # curve for wasteland  
curve(exp(b0 + b5 + + b1\*(x-mean)/sd), add=TRUE, col="darkgreen") # curve for woodland  
mtext(ifelse(Richness.aov[1,3]<0.001, "p < 0.001",   
 paste("p = ", round(Richness.aov[1,3],3))))



Note 2: the graph construction follows the principles previously described for LM. For further details type the following into your internet search engine: “how construct scatterplots for Mixed models - GLMM? If you have transformed your data, include it in the scatterplot syntax, e.g. log(Data.h3$area basal+1), OR back-transform the predicted values to the original scale (as in the previous example ).

#### GLMM-PQL (Fit GLMM using Penalized Quasi-Likelihood - PQL)

This type of GLMM is used when you need to compute a mixed model for a non-normal dispersion (e.g. poisson, binomial) , it is necessary to use a generalized mixed model that allows the inclusion of correlation structures to account for spatial or temporal effects in the random effects and residuals of the model. The usual functions lmer and glmer of the package lme4 do not allow you to include such correlation structures. Thus, you can adjust Partial Quasi-likelihoods for glmm with the function glmmPQL of the package MASS. The most used spatial and temporal correlation structures for a wide range of situations are, respectively, corExp (exponential spatial correlation structure) and corAR1 (correlation structure of first order for time series) of the package nlme.

The glmmPQL approximates a quasi-likelihood by iterative fitting of (re)weighted linear mixed effects (lme) models based on the fit of Glm. Thus, it estimates the fixed effects parameters by fitting a Glm with its incorporated correlation (variance-covariance) structure. This constructs an Lme model and refits it to re-estimate the variance-covariance structure, utilizing the variance structure from the previous Glm. The iterations continue until the fit improvement is below a threshold or a defined number of iterations has occurred. Therefore, this approach accommodates heterogeneity and spatial/temporal autocorrelation in the model. However, it worth being aware of the fit of PQL generalization for Poisson models, since it commonly performs poorly. If your performance with glmmPQL is poor using the Poisson model, take a look at glmmTMB and glmmADMB packages and functions.For more information on usage of glmmPQL, other models and correlation structures to account for spatial and temporal autocorrelation of residuals, read Dormann et al. (2007).

#### GLMM - PQL: Temporal autocorrelation

Hogweed <- read.table(“Hogweed.csv”, header=T, sep=“,”, dec=“.”) head(Hogweed)

install.packages(“MASS”) install.packages(“nlme”) install.packages(“lme4”)

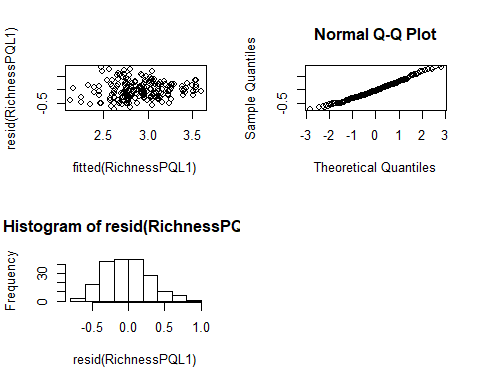
library(MASS) # PQL Estimation Of Generalized Linear Mixed Models (glmmPQL)  
library(nlme) # corAR1 (autocorrelation structure of order 1.   
 # For temporal autocorrelation correction)  
library(lme4) # glmer  
  
RichnessPQL1 <- glmmPQL(Species.richness ~ scale(Hogweed.cover) +   
 Habitat.type, random=~1| Study.area, data=Hogweed,  
 family="poisson", correlation=corAR1())  
  
summary(RichnessPQL1)

## Linear mixed-effects model fit by maximum likelihood  
## Data: Hogweed   
## AIC BIC logLik  
## NA NA NA  
##   
## Random effects:  
## Formula: ~1 | Study.area  
## (Intercept) Residual  
## StdDev: 0.25502 1.360019  
##   
## Correlation Structure: AR(1)  
## Formula: ~1 | Study.area   
## Parameter estimate(s):  
## Phi   
## 0.212907   
## Variance function:  
## Structure: fixed weights  
## Formula: ~invwt   
## Fixed effects: Species.richness ~ scale(Hogweed.cover) + Habitat.type   
## Value Std.Error DF t-value p-value  
## (Intercept) 3.179347 0.08077289 176 39.36156 0.0000  
## scale(Hogweed.cover) -0.084974 0.02702220 176 -3.14458 0.0020  
## Habitat.typeruderal.grass -0.113853 0.07240569 176 -1.57243 0.1176  
## Habitat.typetall.herbs -0.418886 0.07631617 176 -5.48882 0.0000  
## Habitat.typewasteland -0.154868 0.10070796 176 -1.53779 0.1259  
## Habitat.typewoodland -0.471932 0.09683872 176 -4.87338 0.0000  
## Correlation:   
## (Intr) sc(H.) Hbtt.typr. Hbtt.typt. Hbtt.typws  
## scale(Hogweed.cover) 0.240   
## Habitat.typeruderal.grass -0.523 -0.343   
## Habitat.typetall.herbs -0.507 -0.409 0.561   
## Habitat.typewasteland -0.337 0.055 0.408 0.299   
## Habitat.typewoodland -0.357 -0.072 0.368 0.440 0.260   
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.9455147 -0.7636449 -0.1452796 0.6615598 2.3691106   
##   
## Number of Observations: 201  
## Number of Groups: 20

class(RichnessPQL1)="lme"  
anova(object=RichnessPQL1,test="Chisq")

## numDF denDF F-value p-value  
## (Intercept) 1 176 2142.125 <.0001  
## scale(Hogweed.cover) 1 176 29.293 <.0001  
## Habitat.type 4 176 10.733 <.0001

# Residuals  
  
par(mfrow=c(2,2))  
plot(fitted(RichnessPQL1), resid(RichnessPQL1))  
qqnorm(resid(RichnessPQL1))  
qqline(resid(RichnessPQL1))  
hist(resid(RichnessPQL1))



Note: Fitting a “lme” class for the glmmPQL is not recommended as the most appropriate way to acquire p-values from anova of the model above. Here we can compute a lme model for the same variables of the glmmPQL model and compare the outcomes of the anova of both models. Despite lme requiring data to conform to assumptions of normality, we can use this model for a dataset where Y is discrete (Poisson), once the residual fit is as close as possible to normal, which were already complied using the residuals of glmmPQL model.. Let’s see.

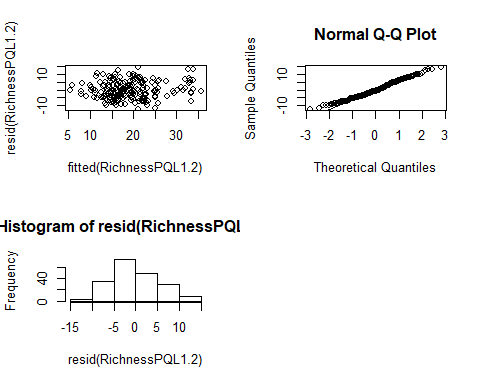
RichnessPQL1.2 <- lme(Species.richness ~ scale(Hogweed.cover) +   
 Habitat.type, random=~1| Study.area,   
 data=Hogweed, correlation = corAR1())  
  
summary(RichnessPQL1.2)

## Linear mixed-effects model fit by REML  
## Data: Hogweed   
## AIC BIC logLik  
## 1316.934 1346.391 -649.4672  
##   
## Random effects:  
## Formula: ~1 | Study.area  
## (Intercept) Residual  
## StdDev: 5.656911 5.910775  
##   
## Correlation Structure: AR(1)  
## Formula: ~1 | Study.area   
## Parameter estimate(s):  
## Phi   
## 0.1828919   
## Fixed effects: Species.richness ~ scale(Hogweed.cover) + Habitat.type   
## Value Std.Error DF t-value p-value  
## (Intercept) 24.741612 1.7117346 176 14.454117 0.0000  
## scale(Hogweed.cover) -1.420397 0.4884303 176 -2.908086 0.0041  
## Habitat.typeruderal.grass -2.853129 1.4641553 176 -1.948652 0.0529  
## Habitat.typetall.herbs -7.851275 1.4603586 176 -5.376265 0.0000  
## Habitat.typewasteland -3.446089 2.0679085 176 -1.666461 0.0974  
## Habitat.typewoodland -9.293588 1.8178651 176 -5.112364 0.0000  
## Correlation:   
## (Intr) sc(H.) Hbtt.typr. Hbtt.typt. Hbtt.typws  
## scale(Hogweed.cover) 0.200   
## Habitat.typeruderal.grass -0.517 -0.332   
## Habitat.typetall.herbs -0.529 -0.417 0.618   
## Habitat.typewasteland -0.345 0.046 0.412 0.352   
## Habitat.typewoodland -0.398 -0.081 0.431 0.506 0.317   
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.0616284 -0.7119363 -0.1222464 0.6971461 2.5063006   
##   
## Number of Observations: 201  
## Number of Groups: 20

anova(RichnessPQL1.2)

## numDF denDF F-value p-value  
## (Intercept) 1 176 212.89541 <.0001  
## scale(Hogweed.cover) 1 176 27.24646 <.0001  
## Habitat.type 4 176 10.43767 <.0001

# Residuals  
  
par(mfrow=c(2,2))  
plot(fitted(RichnessPQL1.2), resid(RichnessPQL1.2))  
qqnorm(resid(RichnessPQL1.2))  
qqline(resid(RichnessPQL1.2))  
hist(resid(RichnessPQL1.2))



Overall, we can see that both the lme and glmmPQL models in this example display well-fitting residuals and the anova outcomes of both provide very similar results.

#### Wald Test for GLMM-PQL

The recommended way to test hypotheses (H0/H1) for glmmPQL, is to compute the Wald between the Wald approach and a typical anova, is that Wald tests encompass marginal tests of parameters, i.e. as they show up in the summary of the model in Wald outcome versus nested model comparisons (either F or Chisq (likedlihood ratio)) in anova() function.

RichnessPQL1 <- glmmPQL(Species.richness ~ scale(Hogweed.cover) +   
 Habitat.type, random=~1| Study.area, data=Hogweed,  
 family="poisson", correlation=corAR1())

## iteration 1

## iteration 2

## iteration 3

summary(RichnessPQL1)

## Linear mixed-effects model fit by maximum likelihood  
## Data: Hogweed   
## AIC BIC logLik  
## NA NA NA  
##   
## Random effects:  
## Formula: ~1 | Study.area  
## (Intercept) Residual  
## StdDev: 0.25502 1.360019  
##   
## Correlation Structure: AR(1)  
## Formula: ~1 | Study.area   
## Parameter estimate(s):  
## Phi   
## 0.212907   
## Variance function:  
## Structure: fixed weights  
## Formula: ~invwt   
## Fixed effects: Species.richness ~ scale(Hogweed.cover) + Habitat.type   
## Value Std.Error DF t-value p-value  
## (Intercept) 3.179347 0.08077289 176 39.36156 0.0000  
## scale(Hogweed.cover) -0.084974 0.02702220 176 -3.14458 0.0020  
## Habitat.typeruderal.grass -0.113853 0.07240569 176 -1.57243 0.1176  
## Habitat.typetall.herbs -0.418886 0.07631617 176 -5.48882 0.0000  
## Habitat.typewasteland -0.154868 0.10070796 176 -1.53779 0.1259  
## Habitat.typewoodland -0.471932 0.09683872 176 -4.87338 0.0000  
## Correlation:   
## (Intr) sc(H.) Hbtt.typr. Hbtt.typt. Hbtt.typws  
## scale(Hogweed.cover) 0.240   
## Habitat.typeruderal.grass -0.523 -0.343   
## Habitat.typetall.herbs -0.507 -0.409 0.561   
## Habitat.typewasteland -0.337 0.055 0.408 0.299   
## Habitat.typewoodland -0.357 -0.072 0.368 0.440 0.260   
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.9455147 -0.7636449 -0.1452796 0.6615598 2.3691106   
##   
## Number of Observations: 201  
## Number of Groups: 20

Since the main statistic of interest is the Wald approach (t for PQL) for the fixed effects parameters, we can firstly fit a crude Wald. The summary of the model gives us crude (Wald) estimates of the p-values for each individual parameter:

coef(summary(RichnessPQL1))

## Value Std.Error DF t-value  
## (Intercept) 3.17934701 0.08077289 176 39.361559  
## scale(Hogweed.cover) -0.08497357 0.02702220 176 -3.144583  
## Habitat.typeruderal.grass -0.11385291 0.07240569 176 -1.572430  
## Habitat.typetall.herbs -0.41888568 0.07631617 176 -5.488820  
## Habitat.typewasteland -0.15486750 0.10070796 176 -1.537788  
## Habitat.typewoodland -0.47193166 0.09683872 176 -4.873378  
## p-value  
## (Intercept) 3.647255e-89  
## scale(Hogweed.cover) 1.952741e-03  
## Habitat.typeruderal.grass 1.176467e-01  
## Habitat.typetall.herbs 1.393556e-07  
## Habitat.typewasteland 1.258963e-01  
## Habitat.typewoodland 2.436558e-06

One could be interested in the investigation of the main effects as a whole. Thus, we use the function “wald.test” of the package aod in order test the influence of the factor as whole using the Wald Z and X² (Wald Chi-squared) test, which is an approach analogous to an ANOVA. In function “wald.test” the following parameters are used:

fixed effects parameter estimates (b) the variance-covariance matrix (Sigma) and (c) a specification of which fixed factor terms (Terms) to combine for the Wald statistic. For more information, please, read the R documentation of the package aod in the wald.test section.

You define the terms by designating the order in which the coefficients of interest appear in the summary of the model. For instance, the predictor scale(Hogweed.cover) is the second coefficient after the intercept and Habitat.typewoodland is sixth and last. If we are interested in the main effects of the groups of parameters as whole, then we write ‘Terms=2:6’. It is important to notice that the Wald X² test is suitable only in the absence of overdispersion in the model. When the model is overdispersed the suitable approach is to use Wald F and X² tests.

# install.packages("aod")  
library(aod)  
  
wald<-wald.test(b=fixef(RichnessPQL1),  
 Sigma=vcov(RichnessPQL1),Terms=2:6)  
  
wald

## Wald test:  
## ----------  
##   
## Chi-squared test:  
## X2 = 74.4, df = 5, P(> X2) = 1.2e-14

The global p-value of Wald confirms the overall significant effect of the numeric (Hogweed.cover) and categorical (Habitat.type) predictors, as found in the other models (glmmPQL and lme), with Species.richness as response variable.

To assess the confidence intervals for model estimates based on Wald statistics.

library(gmodels)  
ci(RichnessPQL1, method="Wald")

## Estimate CI lower CI upper Std. Error  
## (Intercept) 3.17934701 3.0199389 3.33875510 0.08077289  
## scale(Hogweed.cover) -0.08497357 -0.1383028 -0.03164432 0.02702220  
## Habitat.typeruderal.grass -0.11385291 -0.2567480 0.02904221 0.07240569  
## Habitat.typetall.herbs -0.41888568 -0.5694983 -0.26827310 0.07631617  
## Habitat.typewasteland -0.15486750 -0.3536181 0.04388312 0.10070796  
## Habitat.typewoodland -0.47193166 -0.6630462 -0.28081712 0.09683872  
## DF p-value  
## (Intercept) 176 3.647255e-89  
## scale(Hogweed.cover) 176 1.952741e-03  
## Habitat.typeruderal.grass 176 1.176467e-01  
## Habitat.typetall.herbs 176 1.393556e-07  
## Habitat.typewasteland 176 1.258963e-01  
## Habitat.typewoodland 176 2.436558e-06

#### GLMM - PQL: Spatial autocorrelation

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

library(MASS) # PQL Estimation Of Generalized Linear Mixed Models (glmmPQL)  
library(nlme) # corExp (autocorrelation structure of order 1.   
 # For temporal autocorrelation correction)  
library(lme4) # glmer  
  
  
RichnessPQL2 <- glmmPQL(Richness ~ Mortality + Recruitment, random=~1|   
 Site, data=BIOVEG, family=poisson(),   
 correlation = corExp())

## iteration 1

## iteration 2

## iteration 3

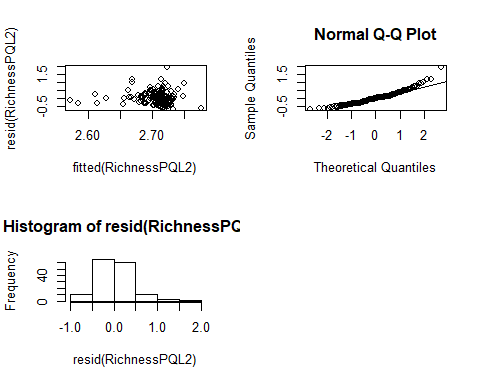
summary(RichnessPQL2)

## Linear mixed-effects model fit by maximum likelihood  
## Data: BIOVEG   
## AIC BIC logLik  
## NA NA NA  
##   
## Random effects:  
## Formula: ~1 | Site  
## (Intercept) Residual  
## StdDev: 2.513287e-05 1.568849  
##   
## Correlation Structure: Exponential spatial correlation  
## Formula: ~1 | Site   
## Parameter estimate(s):  
## range   
## 2.114817   
## Variance function:  
## Structure: fixed weights  
## Formula: ~invwt   
## Fixed effects: Richness ~ Mortality + Recruitment   
## Value Std.Error DF t-value p-value  
## (Intercept) 2.7193377 0.07676437 145 35.42448 0.0000  
## Mortality -0.0076205 0.00877996 145 -0.86795 0.3869  
## Recruitment 0.0044188 0.01109908 145 0.39812 0.6911  
## Correlation:   
## (Intr) Mrtlty  
## Mortality -0.290   
## Recruitment -0.366 -0.075  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.68788217 -0.78368147 -0.02639946 0.48572389 4.85960190   
##   
## Number of Observations: 150  
## Number of Groups: 3

class(RichnessPQL2)="lme"  
anova(RichnessPQL2,test="Chisq")

## numDF denDF F-value p-value  
## (Intercept) 1 145 1629.6022 <.0001  
## Mortality 1 145 0.7061 0.4021  
## Recruitment 1 145 0.1585 0.6911

# Residuals  
  
par(mfrow=c(2,2))  
plot(fitted(RichnessPQL2), resid(RichnessPQL2))  
qqnorm(resid(RichnessPQL2))  
qqline(resid(RichnessPQL2))  
hist(resid(RichnessPQL2))



The model RichnessPQL2 above did not show an acceptable residual fit in the Q.Q Plot and Histogram of residuals. In this case we then need to calculate a glmmPQL negative binomial. Firstly, we compute a glm model for the same set of fixed effect variables to fit the Theta parameter of negative binomial dispersion.

RichnessPQL2.1 <- glm.nb(Richness ~ Mortality + Recruitment, data=BIOVEG)  
theta<-RichnessPQL2.1$theta

RichnessPQL2.1.1 <- glmmPQL(Richness ~ Mortality + Recruitment, random=~1|   
 Site, data=BIOVEG, family=quasipoisson(),   
 correlation = corExp())

## iteration 1

## iteration 2

## iteration 3

summary(RichnessPQL2.1.1)

## Linear mixed-effects model fit by maximum likelihood  
## Data: BIOVEG   
## AIC BIC logLik  
## NA NA NA  
##   
## Random effects:  
## Formula: ~1 | Site  
## (Intercept) Residual  
## StdDev: 2.513287e-05 1.568849  
##   
## Correlation Structure: Exponential spatial correlation  
## Formula: ~1 | Site   
## Parameter estimate(s):  
## range   
## 2.114817   
## Variance function:  
## Structure: fixed weights  
## Formula: ~invwt   
## Fixed effects: Richness ~ Mortality + Recruitment   
## Value Std.Error DF t-value p-value  
## (Intercept) 2.7193377 0.07676437 145 35.42448 0.0000  
## Mortality -0.0076205 0.00877996 145 -0.86795 0.3869  
## Recruitment 0.0044188 0.01109908 145 0.39812 0.6911  
## Correlation:   
## (Intr) Mrtlty  
## Mortality -0.290   
## Recruitment -0.366 -0.075  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.68788217 -0.78368147 -0.02639946 0.48572389 4.85960190   
##   
## Number of Observations: 150  
## Number of Groups: 3

class(RichnessPQL2.1.1)="lme"  
anova(object=RichnessPQL2.1.1,test="Chisq")

## numDF denDF F-value p-value  
## (Intercept) 1 145 1629.6022 <.0001  
## Mortality 1 145 0.7061 0.4021  
## Recruitment 1 145 0.1585 0.6911

The outcomes of both the glmmPQL model with Poisson and Negative binomial dispersion were the same.

Similar to the first example of glmmPQL above, we run a lme model for the same set of variables and compare the residuals and anova outcomes of both models. Despite the fact that lme assumes a normal distribution, we can still consider this model if the residuals reasonably conform with a normal fit.

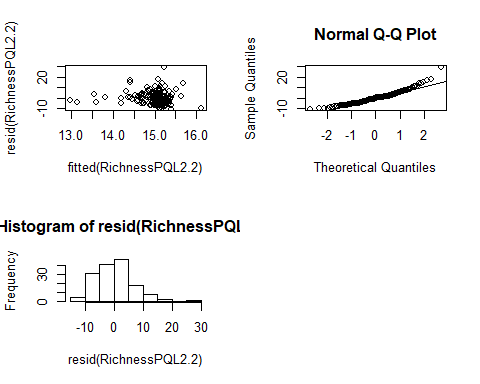
RichnessPQL2.2 <- lme(Richness ~ Mortality + Recruitment, random=~1|   
 Site, data=BIOVEG, correlation   
 = corExp())  
summary(RichnessPQL2.2)

## Linear mixed-effects model fit by REML  
## Data: BIOVEG   
## AIC BIC logLik  
## 909.0265 926.9691 -448.5133  
##   
## Random effects:  
## Formula: ~1 | Site  
## (Intercept) Residual  
## StdDev: 0.003170964 6.205825  
##   
## Correlation Structure: Exponential spatial correlation  
## Formula: ~1 | Site   
## Parameter estimate(s):  
## range   
## 2.19089   
## Fixed effects: Richness ~ Mortality + Recruitment   
## Value Std.Error DF t-value p-value  
## (Intercept) 15.147875 1.1779626 145 12.859385 0.0000  
## Mortality -0.113682 0.1260718 145 -0.901720 0.3687  
## Recruitment 0.072715 0.1686360 145 0.431194 0.6670  
## Correlation:   
## (Intr) Mrtlty  
## Mortality -0.298   
## Recruitment -0.350 -0.076  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.67557693 -0.75056004 -0.02212395 0.47904130 4.79686539   
##   
## Number of Observations: 150  
## Number of Groups: 3

anova(RichnessPQL2.2)

## numDF denDF F-value p-value  
## (Intercept) 1 145 209.95041 <.0001  
## Mortality 1 145 0.75942 0.385  
## Recruitment 1 145 0.18593 0.667

# Residuals  
  
par(mfrow=c(2,2))  
plot(fitted(RichnessPQL2.2), resid(RichnessPQL2.2))  
qqnorm(resid(RichnessPQL2.2))  
qqline(resid(RichnessPQL2.2))  
hist(resid(RichnessPQL2.2))



The anova output of the lme model, similar to those of the glmmPQL with Richness as response variable, shows that none of the predictors are significantly related to this response variable. However, the residual fit of this lme model is also not as good as those of the glmmPQL models for this response variable. Thus, we can try to log-transform Richness, though log transformations of discrete data are largely not advised and their use demands a cautious approach.

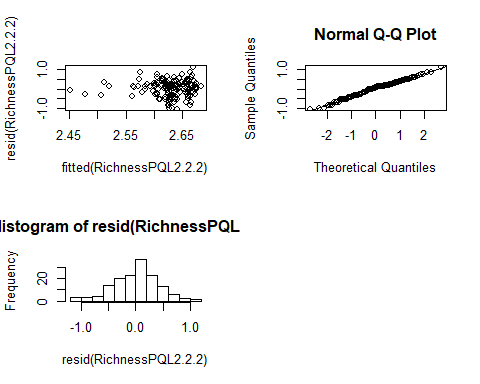
RichnessPQL2.2.2 <- lme(log(Richness) ~ Mortality + Recruitment, random=~1|   
 Site, data=BIOVEG, correlation   
 = corExp())  
summary(RichnessPQL2.2.2)

## Linear mixed-effects model fit by REML  
## Data: BIOVEG   
## AIC BIC logLik  
## 121.274 139.2166 -54.63701  
##   
## Random effects:  
## Formula: ~1 | Site  
## (Intercept) Residual  
## StdDev: 0.04515893 0.4084216  
##   
## Correlation Structure: Exponential spatial correlation  
## Formula: ~1 | Site   
## Parameter estimate(s):  
## range   
## 1.907718   
## Fixed effects: log(Richness) ~ Mortality + Recruitment   
## Value Std.Error DF t-value p-value  
## (Intercept) 2.6515169 0.07939254 145 33.39756 0.0000  
## Mortality -0.0092208 0.00874206 145 -1.05476 0.2933  
## Recruitment 0.0025622 0.01170855 145 0.21883 0.8271  
## Correlation:   
## (Intr) Mrtlty  
## Mortality -0.308   
## Recruitment -0.359 -0.074  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.5263531 -0.7349958 0.1569121 0.6254959 2.7853229   
##   
## Number of Observations: 150  
## Number of Groups: 3

anova(RichnessPQL2.2.2)

## numDF denDF F-value p-value  
## (Intercept) 1 145 1446.6712 <.0001  
## Mortality 1 145 1.0846 0.2994  
## Recruitment 1 145 0.0479 0.8271

# Residuals  
  
par(mfrow=c(2,2))  
plot(fitted(RichnessPQL2.2.2), resid(RichnessPQL2.2.2))  
qqnorm(resid(RichnessPQL2.2.2))  
qqline(resid(RichnessPQL2.2.2))  
hist(resid(RichnessPQL2.2.2))



For this lme model, the log transformation of Richness worked well. The residuals are better fitted and the anova output maintains non-significance for the predictor variables, similar to the previous glmmPQL and lme models. This suggests that there is no great distortion of the empirical condition of the data in the model after the log transformation.

#### Wald Test

Now, let’s compute the Wald test for the model “RichnessPQL2.1.1”

summary(RichnessPQL2.1.1)

## Linear mixed-effects model fit by maximum likelihood  
## Data: BIOVEG   
## AIC BIC logLik  
## NA NA NA  
##   
## Random effects:  
## Formula: ~1 | Site  
## (Intercept) Residual  
## StdDev: 2.513287e-05 1.568849  
##   
## Correlation Structure: Exponential spatial correlation  
## Formula: ~1 | Site   
## Parameter estimate(s):  
## range   
## 2.114817   
## Variance function:  
## Structure: fixed weights  
## Formula: ~invwt   
## Fixed effects: Richness ~ Mortality + Recruitment   
## Value Std.Error DF t-value p-value  
## (Intercept) 2.7193377 0.07676437 145 35.42448 0.0000  
## Mortality -0.0076205 0.00877996 145 -0.86795 0.3869  
## Recruitment 0.0044188 0.01109908 145 0.39812 0.6911  
## Correlation:   
## (Intr) Mrtlty  
## Mortality -0.290   
## Recruitment -0.366 -0.075  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.68788217 -0.78368147 -0.02639946 0.48572389 4.85960190   
##   
## Number of Observations: 150  
## Number of Groups: 3

wald2<-wald.test(b=fixef(RichnessPQL2.1.1),  
 Sigma=vcov(RichnessPQL2.1.1),Terms=2:3)  
  
wald2

## Wald test:  
## ----------  
##   
## Chi-squared test:  
## X2 = 0.88, df = 2, P(> X2) = 0.64

The global p-value of Wald confirms the non-significant effects of the predictors (Mortality and Recruitment) as found in the previous glmmPQL and lme models for this set of variables.

# Confidence Interval under Wald fit.  
  
ci(RichnessPQL2.1.1, method=Wald)

## Estimate CI lower CI upper Std. Error DF  
## (Intercept) 2.719337688 2.56761601 2.871059367 0.076764373 145  
## Mortality -0.007620525 -0.02497375 0.009732702 0.008779955 145  
## Recruitment 0.004418759 -0.01751812 0.026355633 0.011099076 145  
## p-value  
## (Intercept) 2.826205e-73  
## Mortality 3.868582e-01  
## Recruitment 6.911276e-01

##### Useful links and examples:

<https://rpubs.com/bbolker/glmmchapter>

<http://www.flutterbys.com.au/stats/tut/tut11.2a.html>

<https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html>

<https://bbolker.github.io/mixedmodels-misc/ecostats_chap.html>

<https://www.rdocumentation.org/packages/MASS/versions/7.3-47/topics/glmmPQL>

Recommended Reading:

Box, G.E.P., Jenkins, G.M., and Reinsel G.C. (1994) “Time Series Analysis: Forecasting and Control”, 3rd Edition, Holden-Day.

Pinheiro, J.C., and Bates, D.M. (2000) “Mixed-Effects Models in S and S-PLUS”, Springer, esp. pp. 235, 397.

Cressie, N.A.C. (1993), “Statistics for Spatial Data”, J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (2002) “Modern Applied Statistics with S”, 4th Edition, Springer-Verlag.

#### LOGISTIC REGRESSION

Hypothesis test: in a period of 5 years do the continuous variables (mean diameter and mean precipitation) affect the likelihood of tree recruitment, i.e., survival, growth and achieve a minimum diameter criteria (e.g. > 15 cm). That is, we want to know if in a five years-period of forest monitoring, do two continuous variables (Mean\_diameter + Precipitation\_mean) and one categorical (type of vegetation) determine the success of a trees individual likelihood of recruitment (0 indicates the absence of recruitment in a subplot and 1 the presence).

Lreg <- read.table("Logit2.csv", header=T, sep=";", dec=".")  
head(Lreg,4)

## type1 Site Mean\_diameter Precipitation\_mean Rec.binar  
## 1 Ombrophilous AREA1 11.657 1142 1  
## 2 Ombrophilous AREA1 13.793 1142 1  
## 3 Ombrophilous AREA1 17.928 1142 1  
## 4 Ombrophilous AREA1 12.212 1142 1

Recrut.binar<- glm(Rec.binar ~ Mean\_diameter + Precipitation\_mean +   
 type1, data=Lreg,family="binomial")  
Recrut.binar

##   
## Call: glm(formula = Rec.binar ~ Mean\_diameter + Precipitation\_mean +   
## type1, family = "binomial", data = Lreg)  
##   
## Coefficients:  
## (Intercept) Mean\_diameter Precipitation\_mean   
## 4.431746 -0.143722 -0.001657   
## type1Ombrophilous type1Semideciduos   
## -0.825271 -0.120710   
##   
## Degrees of Freedom: 449 Total (i.e. Null); 445 Residual  
## Null Deviance: 586.9   
## Residual Deviance: 553.5 AIC: 563.5

According to the output, the model is logit(pi) = 4.43 + -0.14*Mean\_diameter + 0.001*Precipitation\_mean - -0.82*typeOmbrophilous + 0.12*typeSemideciduos

#### LRT (likelihood ratio test)

To test the overall model fit and hypothesis

LRT compares the full model with a reduced model where the explanatory variables of interest are omitted and provides p-values of the tests calculated using Chi-squared distribution.

Rec.bin.reduced<- glm(Rec.binar ~ 1, data=Lreg, family="binomial")  
anova(Rec.bin.reduced,Recrut.binar, test="Chisq")

## Analysis of Deviance Table  
##   
## Model 1: Rec.binar ~ 1  
## Model 2: Rec.binar ~ Mean\_diameter + Precipitation\_mean + type1  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 449 586.92   
## 2 445 553.52 4 33.4 9.893e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The LRT is 33.399 with a p-value 9.895e-07. Thus, we have strong evidence, based on high significance, that there is influence of some of the predictors in the success or failure of tree recruitment.

Perform tests on the individual regression parameters:

summary(Recrut.binar)

##   
## Call:  
## glm(formula = Rec.binar ~ Mean\_diameter + Precipitation\_mean +   
## type1, family = "binomial", data = Lreg)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8080 -1.2327 0.7476 0.9216 1.9004   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.431746 1.742508 2.543 0.01098 \*   
## Mean\_diameter -0.143722 0.035671 -4.029 5.6e-05 \*\*\*  
## Precipitation\_mean -0.001657 0.001402 -1.182 0.23730   
## type1Ombrophilous -0.825271 0.314016 -2.628 0.00859 \*\*   
## type1Semideciduos -0.120710 0.280880 -0.430 0.66737   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 586.92 on 449 degrees of freedom  
## Residual deviance: 553.52 on 445 degrees of freedom  
## AIC: 563.52  
##   
## Number of Fisher Scoring iterations: 4

We can see that mean diameter (z = -4.029 and p=5.6e-05) appears to have a significant impact on the likelihood of recruitment success or failure of the trees. The same way the vegetation type Ombrophilous (z = -2.628 and p=0.00859) seems to have significant negative impact (Estimate = -0.825271) on the recruitment success. Semideciduous forest (z = -2.628 and p = 0.00859 \*\*)does not seem to have a significant effect on recruitment, once mean diameter and ombrophilous are included in the model. The same can be said about the effect of mean precipitation (z =-1.505 and p = 0.13224) on recruitment, since it’s effect is controlled by the stronger influence of mean diameter and ombrophilous type.

We can check for significance of influence from the predictors by executing a test for the full model.

anova(Recrut.binar, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: Rec.binar  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 449 586.92   
## Mean\_diameter 1 23.8193 448 563.10 1.058e-06 \*\*\*  
## Precipitation\_mean 1 0.1662 447 562.93 0.683539   
## type1 2 9.4142 445 553.52 0.009031 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# To compute the odds of successful tree recruitment as a function of mean diameter   
# and ombrophilous type  
  
exp(coef(Recrut.binar))

## (Intercept) Mean\_diameter Precipitation\_mean   
## 84.0780806 0.8661288 0.9983448   
## type1Ombrophilous type1Semideciduos   
## 0.4381162 0.8862912

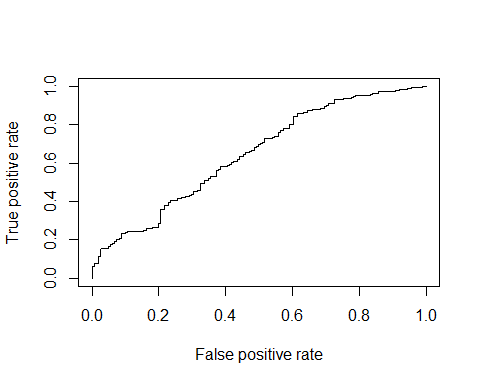
## To create a 95% confidence interval for the estimate, type:  
  
exp(confint.default(Recrut.binar))

## 2.5 % 97.5 %  
## (Intercept) 2.7634844 2558.0472670  
## Mean\_diameter 0.8076423 0.9288507  
## Precipitation\_mean 0.9956056 1.0010915  
## type1Ombrophilous 0.2367547 0.8107368  
## type1Semideciduos 0.5110827 1.5369570

We see that the odds ratio corresponding to mean diameter is 0.87 (95% CI: (0.80, 0.92)) and Ombrophilous is 0.43 (95% CI: (0.23, 0.811))

To use the fitted logistic regression curve to estimate the predictive power and accuracy of the model. The most commonly used way to check the predictive ability of a logistic model is to fit the ROC curve and its AUC (area under the curve), which are typical performance measurements for a binary classifier. The ROC is a curve which plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings, whereas the AUC is the area under the ROC curve. A commonly used rule to assess the robustness of predictive power of the model, is to consider a good predictive ability when AUC is closer to 1 (1 is ideal) than to 0.5.

# install.packages("ROCR")  
library(ROCR)  
  
p <- predict(Recrut.binar, type="response") # predict scorees  
pr <- prediction(p, Lreg$Rec.binar)  
prf <- performance(pr, measure = "tpr", x.measure = "fpr")  
plot(prf)



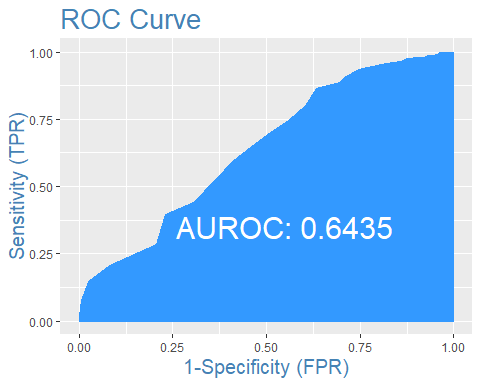
auc <- performance(pr, measure = "auc")  
auc <- auc@y.values[[1]]  
auc

## [1] 0.6462206

Or we can use the package plotROC for a better presentation of the AUC and true and false positive rates.

install.packages(“InformationValue”) # To plot ROC curve

library(InformationValue)  
  
P<- plotROC(Lreg$Rec.binar, p)



Here we got a reasonable predictive accuracy (0.64), but it could be better (e.g. > 0.70).

#### Moran’s I: spatial autocorrelation test

Spatial autocorrelation refers to the correlation between the values of a variable due their proximity in geographical space. This violates the assumption that they are from independent observations.

The function “moransI” of the package “lctools” allows us to estimate the local spatial correlation of the residuals from a linear model.

In order to execute the classical Moran’s I test with this function, we first must calculate and provide metric coordinates, such as UTM coordinates, for each row of data. Firstly you must convert the geographical coordinates belonging to your data to UTM. After this, if you are using subplots (e.g. treatment with 50 subplots of 10 meters per 10 meters) you must add the distance of each subplot from the reference point of the site to the UTM coordinates of the site. This is done separately for X direction (East-West) and Y direction (North-South). If the reference point is the South-West corner of the site, you add the distances of the subplot in W-E and S-N direction to the site coordinates (e.g. 10 m and 30 m). But, if the reference point is the North-East corner, you subtract the distances from the site coordinates.

Each row of your data will contain real coordinates based on the sample units (here represented by subplot) and you will be able to estimate spatial autocorrelation in your linear model.

It is also important to note that Moran’s I ranges from -1 to 1. Values close to -1 imply negative spatial autocorrelation (low values tend to have neighbours with high values and vice versa) and values close to 1 represent positive spatial autocorrelation (spatial clusters of similarly low or high values among neighbour subplots, as in the present example).

The function “moransI” provides the Moran’s I value (named as “Morans.I”) and the two tailed p-value (named “p.value.resampling” in the output) to test the significance of Moran’s I and, consequently, significance of spatial autocorrelation.

In order to graphically demonstrate the results for Moran’s I, you can create a correlogram with the function “correlog” of the package “ncf” as shown in the example below. There, a univariate spatial correlogram based on the spatial distances between the subplots is provided. Additionally, a plot with the corresponding p-values of the Moran’s I statistic is drawn, including a line of the threshold (p=0.05) to facilitate assessment of significance.

# install.packages("lctools")  
# install.packages("ncf")  
  
library(lctools) # Moran's I  
library(ncf) # correlogram  
  
BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG, 4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

LM to base the Moran’s I computation:

Mean.Diameter.pre <- lm(Mean\_diameter ~ Mortality + Recruitment +   
 Site, data=BIOVEG)

##### Moran’s I

Moran.M.diameter <- moransI(Coords=cbind(BIOVEG$local.utm.x, BIOVEG$local.utm.y),   
 Bandwidth=4, resid(Mean.Diameter.pre), WType="Binary")   
  
Moran.M.diameter$Morans.I

## 1   
## 0.6320286

Moran.M.diameter$p.value.resampling # significance of the Moran's I

## 1   
## 9.537331e-37

Moran.M.diameter$p.value.randomization # another variant of p-value

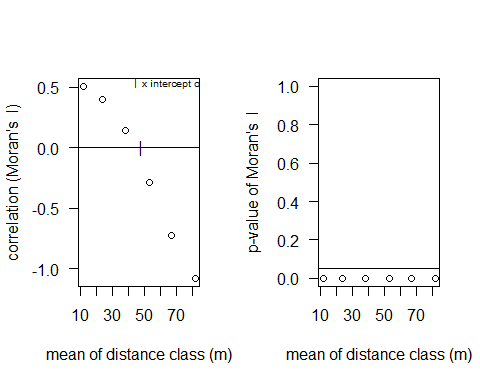
## 1   
## 1.065647e-36

Graph (Correlogram):

Cor.Mean\_diameter <- correlog(x=BIOVEG$local.utm.x, y=BIOVEG$local.utm.y,   
 as.vector(resid(Mean.Diameter.pre)), increment=15, resamp=1000)

## 100 of 1000   
200 of 1000   
300 of 1000   
400 of 1000   
500 of 1000   
600 of 1000   
700 of 1000   
800 of 1000   
900 of 1000   
1000 of 1000

x11(width=12, height=12) ####  
par(mfrow=c(1,2))  
plot(Cor.Mean\_diameter$mean.of.class[1:6], Cor.Mean\_diameter$correlation[1:6],   
 xlab="mean of distance class (m)", ylab="correlation (Moran's I)", las=1)  
abline(0, 0)  
points(Cor.Mean\_diameter$x.intercept, 0, pch="|")  
legend("topright", "x intercept of autocorrelation", pch="|", bty="n", cex=0.6)  
 plot(Cor.Mean\_diameter$mean.of.class[1:6], Cor.Mean\_diameter$p[1:6], ylim=c(0,1),   
 xlab="mean of distance class (m)", ylab="p-value of Moran's I", las=1)  
abline(0.05, 0)



# dev.off() # turn off the graphics device

#### LME (Linear Mixed Effects)

##### With spatial autocorrelation’s correction included

A LME model is a mixed model is implemented to deal with pseudoreplicates by including random effects of, e.g., sites that meet the assumption of normal distribution of residuals. In addition to random effects, it can handle spatial autocorrelation. Therefore, if in a previous test (e.g. Moran’s I) you find significant spatial autocorrelation in a Linear Mixed Model (LMM; with random and fixed effects and normal distribution), you must use LME with a spatial correlation strucure in the model. If you have non-normal data with spatial auto-correlation please check the function glmmPQL of the package MASS to calculate a GLMM with a spatial correlation structure.

There are different variants of spatial correlation structures. One good example is Exponential correlation (function “corExp”). The correlation structure must be included in the syntax of the lme command as in the analyses below. If you don’t include a correlation function (e.g. corExp) in the LME model, it is just like a common LMM. However, like GLMM, this would not be enough to resolve the autocorrelation and provide reliable p-values. In order to test spatial autocorrelation by Moran’s I, you must first construct a LM with the same predictors that will be used in the LME (see LM model used in the section Linearity Tests and Moran’s I). To better understand this approach of handling spatial autocorrelation using the proper regression models, read Dormann et al. (2007).

Dormann, C.F., M. McPherson, J., B. Araújo, M., Bivand, R., Bolliger, J., Carl, G.et al. (2007). Methods to account for spatial autocorrelation in the analysis of species distributional data: A review. Ecography (Cop.)., 30, 609–628

Note 1: LMM can be calculated using either Maximum Likelihood (ML) or Restricted Maximum Likelihood (REML). If we are focusing on estimation and testing of fixed effects, we choose “ML”. In addition to correlation of the residuals, lme can also deal with heterogeneous variance of residuals (heteroscedasticity), e.g. when variances differ among groups, such as sites. For this purpose, we can use the “weights” argument.

# install.packages("nlme")  
# install.packages("car")  
  
library(nlme) # LME models  
library(car) # Anova type II gives the predictors the same chance of   
 # being significant

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG,4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

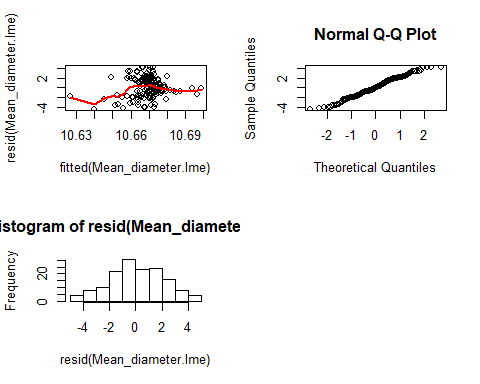
Mean\_diameter.lme<- lme(Mean\_diameter ~ Mortality + Recruitment, data=BIOVEG,   
 method="ML", random= ~1|Site, corr=corExp(form=~local.utm.x+   
 local.utm.y|Site), weights=varIdent(form=~1|Site), na.action=na.omit)  
  
summary(Mean\_diameter.lme)

## Linear mixed-effects model fit by maximum likelihood  
## Data: BIOVEG   
## AIC BIC logLik  
## 97.1106 121.1957 -40.5553  
##   
## Random effects:  
## Formula: ~1 | Site  
## (Intercept) Residual  
## StdDev: 0.0001718405 2.831905  
##   
## Correlation Structure: Exponential spatial correlation  
## Formula: ~local.utm.x + local.utm.y | Site   
## Parameter estimate(s):  
## range   
## 999.8204   
## Variance function:  
## Structure: Different standard deviations per stratum  
## Formula: ~1 | Site   
## Parameter estimates:  
## AREA1 AREA2 AREA3   
## 1.0000000 0.3103472 2.6947936   
## Fixed effects: Mean\_diameter ~ Mortality + Recruitment   
## Value Std.Error DF t-value p-value  
## (Intercept) 10.672167 0.8185581 145 13.037763 0.0000  
## Mortality 0.001605 0.0032656 145 0.491505 0.6238  
## Recruitment -0.003023 0.0091995 145 -0.328623 0.7429  
## Correlation:   
## (Intr) Mrtlty  
## Mortality -0.006   
## Recruitment -0.032 -0.053  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.51239300 -0.68377730 -0.01067181 0.29650175 1.46812947   
##   
## Number of Observations: 150  
## Number of Groups: 3

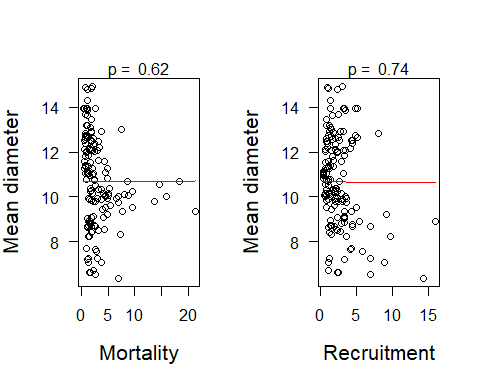
## Anova from car package  
  
An<- Anova(Mean\_diameter.lme, Type = II)  
An

## Analysis of Deviance Table (Type II tests)  
##   
## Response: Mean\_diameter  
## Chisq Df Pr(>Chisq)  
## Mortality 0.2465 1 0.6195  
## Recruitment 0.1102 1 0.7399

## Residuals  
  
x11(width=12, height=12) ####  
par(mfrow=c(2,2))  
plot(fitted(Mean\_diameter.lme), resid(Mean\_diameter.lme))  
lines(smooth.spline(fitted(Mean\_diameter.lme), resid(Mean\_diameter.lme),   
 spar=c(1)), col="red", lwd=2)  
qqnorm(resid(Mean\_diameter.lme))  
qqline(resid(Mean\_diameter.lme))  
hist(resid(Mean\_diameter.lme))  
#dev.off() ## turn off the graphics device



## Scatterplot   
  
b.0.Mean\_diameter<- Mean\_diameter.lme$coefficients$fixed[1]  
b.1.Mean\_diameter<- Mean\_diameter.lme$coefficients$fixed[2]  
b.2.Mean\_diameter<- Mean\_diameter.lme$coefficients$fixed[3]  
  
x11(width=14, height=8)  
par(mfrow=c(1,2))  
plot(BIOVEG$Mean\_diameter ~ BIOVEG$Mortality, las=1, ylab="Mean diameter", xlab="Mortality",   
 cex.lab=1.3)  
curve(b.0.Mean\_diameter + b.1.Mean\_diameter\*x, add=TRUE, col="red")  
mtext(ifelse(An[1,3]<0.001, "p < 0.001", paste("p = ", round(An[1,3],3))))  
plot(BIOVEG$Mean\_diameter ~ BIOVEG$Recruitment, las=1, ylab="Mean diameter", xlab="Recruitment",   
 cex.lab=1.3)  
curve(b.0.Mean\_diameter + b.2.Mean\_diameter\*x, add=TRUE, col="red")  
mtext(ifelse(An[2,3]<0.001, "p < 0.001", paste("p = ", round(An[2,3],3))))



#dev.off() ## turn off the graphics device

#### GLS (Generalized Least Squares)

GLS is another variant of linear regression. It is mostly used to perform analysis when there is significant spatial (or temporal) correlation between the residuals (or if there is heteroscedasticity), but you do not have any random effect in the model. In the absence of random effects, you cannot use lme, but you can use gls. GLS models fit correlation structures of the variance-covariance matrix to model the (spatial) dependence of observations, just as LME models do. Thus, gls is an efficient tool to deal with spatial autocorrelation when you would otherwise use lm (read Dormann et al. 2007).

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

N.B. GLS assumes a normal distribution.

In cases such as in the example below, when you just want to know whether there is variation of a variable among treatments and study sites, for instance, but your data comes from pseudoreplicates (e.g. subplots in our example below), GLS is a good option. Notice that in this case, you use site as a fixed effect. Otherwise, if you want to include site as a random effect, you must use a mixed effect model (GLMM or LME). Before calculating a GLS model, you should use Moran’s I to check the significance of spatial autocorrelation. If significant autocorrelation is found, you must include a correlation structure in the model (e.g. correlation=corExp).

## Pre-test: Spatial autocorrelation (Moran'I)  
  
BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG,4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

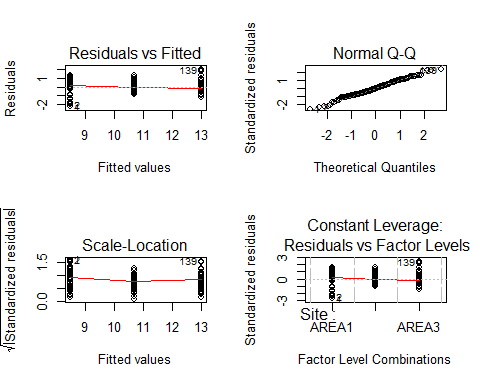
## LM  
  
Mean.Diameter.pre <- lm(Mean\_diameter ~ Site, data=BIOVEG)   
summary(Mean.Diameter.pre)

##   
## Call:  
## lm(formula = Mean\_diameter ~ Site, data = BIOVEG)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.14642 -0.57963 -0.01675 0.58659 1.94452   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.5034 0.1168 72.81 <2e-16 \*\*\*  
## SiteAREA2 2.1837 0.1652 13.22 <2e-16 \*\*\*  
## SiteAREA3 4.4821 0.1652 27.14 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.8258 on 147 degrees of freedom  
## Multiple R-squared: 0.8336, Adjusted R-squared: 0.8314   
## F-statistic: 368.3 on 2 and 147 DF, p-value: < 2.2e-16

shapiro.test(BIOVEG$Mean\_diameter)

##   
## Shapiro-Wilk normality test  
##   
## data: BIOVEG$Mean\_diameter  
## W = 0.98496, p-value = 0.1022

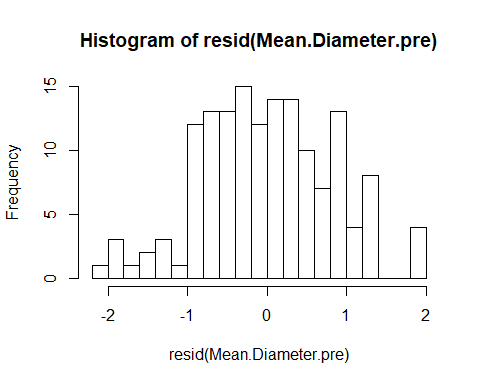
# Residuals  
  
x11(width=12, height=12)  
par(mfrow=c(2,2))  
plot(Mean.Diameter.pre)



dev.off() ## turn off the graphics device

## png   
## 2

x11(width=12, height=12)  
hist(resid(Mean.Diameter.pre),breaks=20)



# dev.off() ## turn off the graphics device

## Moran's I  
  
library(lctools) # Moran's I  
  
Moran.M.diameter <- moransI(Coords=cbind(BIOVEG$local.utm.x, BIOVEG$local.utm.y),   
 Bandwidth=4, resid(Mean.Diameter.pre), WType="Binary")   
  
Moran.M.diameter$Morans.I

## 1   
## 0.6691855

Moran.M.diameter$p.value.resampling # significance of the Moran's I

## 1   
## 6.116947e-41

Moran.M.diameter$p.value.randomization

## 1   
## 6.556257e-41

#### GLS model

Since we found significant spatial autocorrelation in the pre-test with Moran’s I, a correlation (corExp) structure must be added in the syntax of the model

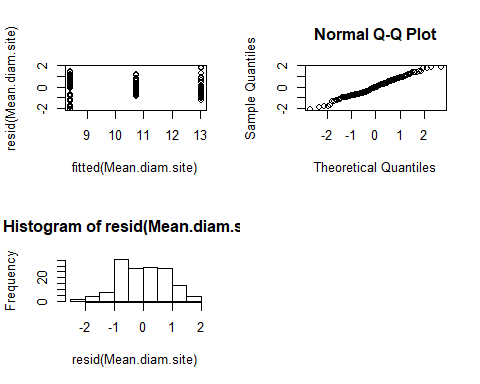
library(nlme)  
library(car)  
  
Mean.diam.site<- gls(Mean\_diameter ~ Site, correlation=corExp(form=~1|Site),   
 data=BIOVEG)  
   
summary(Mean.diam.site)

## Generalized least squares fit by REML  
## Model: Mean\_diameter ~ Site   
## Data: BIOVEG   
## AIC BIC logLik  
## 204.1488 219.1009 -97.07439  
##   
## Correlation Structure: Exponential spatial correlation  
## Formula: ~1 | Site   
## Parameter estimate(s):  
## range   
## 10.55233   
##   
## Coefficients:  
## Value Std.Error t-value p-value  
## (Intercept) 8.391105 0.6110569 13.732118 0.0000  
## SiteAREA2 2.361087 0.8641649 2.732218 0.0071  
## SiteAREA3 4.646891 0.8641649 5.377319 0.0000  
##   
## Correlation:   
## (Intr) SAREA2  
## SiteAREA2 -0.707   
## SiteAREA3 -0.707 0.500  
##   
## Standardized residuals:  
## Min Q1 Med Q3 Max   
## -1.82692630 -0.57058862 -0.03421433 0.53862327 1.69929827   
##   
## Residual standard error: 1.113403   
## Degrees of freedom: 150 total; 147 residual

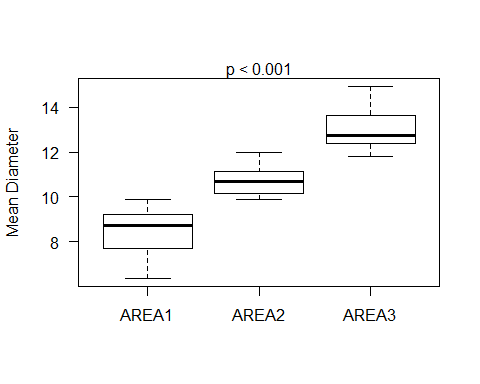
An<- Anova(Mean.diam.site)  
An

## Analysis of Deviance Table (Type II tests)  
##   
## Response: Mean\_diameter  
## Df Chisq Pr(>Chisq)   
## Site 2 28.918 5.254e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Residuals  
  
x11(width=12, height=12)  
par(mfrow=c(2,2))  
plot(fitted(Mean.diam.site), resid(Mean.diam.site))  
qqnorm(resid(Mean.diam.site))  
hist(resid(Mean.diam.site))  
# dev.off()



# Boxplot  
  
x11(width=12, height=12)  
par(mfrow=c(1,1))  
boxplot(BIOVEG$Mean\_diameter ~ BIOVEG$Site, las=1, ylab="Mean Diameter")  
mtext(ifelse(An[1,3]<0.001, "p < 0.001", paste("p = ", round(An[1,3],3))))



# dev.off()

#### QUADRATIC MODEL

A quadratic regression is an extension of simple linear regression aiming to find the best fit equation a set of data (dependent and predictor variables) shaped like a parabola (e.g. U-shape curve). A quadratic fit is useful When the coefficients and, consequently, the curve appear to fit the data better than the linear model does.

Create linear model of species richness vs. Hogweed cover

Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

M.lin <- glm(Species.richness ~ Hogweed.cover + Habitat.type, data=Hogweed,   
 family="quasipoisson")  
summary(M.lin)

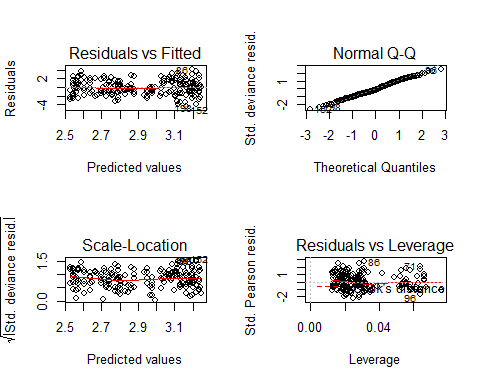
##   
## Call:  
## glm(formula = Species.richness ~ Hogweed.cover + Habitat.type,   
## family = "quasipoisson", data = Hogweed)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.3435 -1.2231 -0.2592 1.2230 4.0514   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.248210 0.059719 54.392 < 2e-16 \*\*\*  
## Hogweed.cover -0.003072 0.001094 -2.807 0.00551 \*\*   
## Habitat.typeruderal.grass -0.022699 0.080921 -0.281 0.77938   
## Habitat.typetall.herbs -0.428306 0.083712 -5.116 7.43e-07 \*\*\*  
## Habitat.typewasteland -0.101591 0.106856 -0.951 0.34292   
## Habitat.typewoodland -0.354923 0.111048 -3.196 0.00163 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 2.884375)  
##   
## Null deviance: 775.16 on 200 degrees of freedom  
## Residual deviance: 571.60 on 195 degrees of freedom  
## AIC: NA  
##   
## Number of Fisher Scoring iterations: 4

anova(M.lin, test = "F")

## Analysis of Deviance Table  
##   
## Model: quasipoisson, link: log  
##   
## Response: Species.richness  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev F Pr(>F)   
## NULL 200 775.16   
## Hogweed.cover 1 70.215 199 704.94 24.343 1.723e-06 \*\*\*  
## Habitat.type 4 133.345 195 571.60 11.557 1.929e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

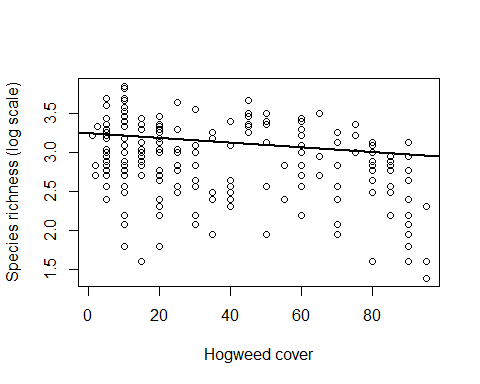
Hogweed cover has a significant negative linear effect on species richness

# Model diangostics  
  
par(mfrow = c(2, 2))  
plot(M.lin)



The diagnostic graphs look ok.

# Plot the data and the regression line  
  
par(mfrow = c(1, 1))  
plot(log(Species.richness) ~ Hogweed.cover, data = Hogweed,   
 xlab = "Hogweed cover", ylab = "Species richness (log scale)")  
abline(glm(Species.richness ~ Hogweed.cover + Habitat.type, data=Hogweed,   
 family="quasipoisson"), lwd = 2)



The plot suggests that a quadartic model mignt fit the data even better

# Create variable of centered and squared Hogweed cover  
  
Mean.cover <- mean(Hogweed$Hogweed.cover)  
Hogweed.cover.sq <- (Hogweed$Hogweed.cover - Mean.cover)^2

Calculate the model with original and, additionally, squared Hogweed cover

M.quad <- glm(Species.richness ~ Hogweed.cover + Hogweed.cover.sq + Habitat.type,   
 data=Hogweed, family="quasipoisson")  
summary(M.quad)

##   
## Call:  
## glm(formula = Species.richness ~ Hogweed.cover + Hogweed.cover.sq +   
## Habitat.type, family = "quasipoisson", data = Hogweed)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.3890 -1.3328 -0.2187 1.0562 4.2712   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.286e+00 6.258e-02 52.502 < 2e-16 \*\*\*  
## Hogweed.cover -1.718e-03 1.303e-03 -1.319 0.188639   
## Hogweed.cover.sq -8.413e-05 4.408e-05 -1.908 0.057825 .   
## Habitat.typeruderal.grass -5.146e-02 8.144e-02 -0.632 0.528221   
## Habitat.typetall.herbs -4.418e-01 8.299e-02 -5.324 2.79e-07 \*\*\*  
## Habitat.typewasteland -1.010e-01 1.060e-01 -0.952 0.342054   
## Habitat.typewoodland -3.748e-01 1.107e-01 -3.387 0.000855 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 2.838361)  
##   
## Null deviance: 775.16 on 200 degrees of freedom  
## Residual deviance: 561.23 on 194 degrees of freedom  
## AIC: NA  
##   
## Number of Fisher Scoring iterations: 4

anova(M.quad, test="F")

## Analysis of Deviance Table  
##   
## Model: quasipoisson, link: log  
##   
## Response: Species.richness  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev F Pr(>F)   
## NULL 200 775.16   
## Hogweed.cover 1 70.215 199 704.94 24.7377 1.442e-06 \*\*\*  
## Hogweed.cover.sq 1 10.471 198 694.47 3.6891 0.05623 .   
## Habitat.type 4 133.242 194 561.23 11.7358 1.479e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Compare linear and quadratic model

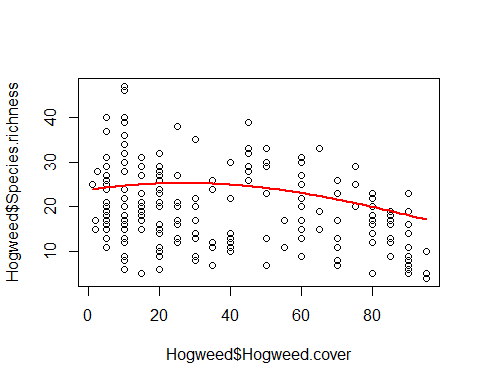
anova(M.lin, M.quad, test = "F")

## Analysis of Deviance Table  
##   
## Model 1: Species.richness ~ Hogweed.cover + Habitat.type  
## Model 2: Species.richness ~ Hogweed.cover + Hogweed.cover.sq + Habitat.type  
## Resid. Df Resid. Dev Df Deviance F Pr(>F)   
## 1 195 571.60   
## 2 194 561.23 1 10.368 3.6527 0.05745 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Squared hogweed cover is (marginally) not significant.

Anyway, let’s create a scatter plot with regression curve.

plot(Hogweed$Species.richness ~ Hogweed$Hogweed.cover)  
  
b0 <- coef(M.quad)[1] # Intercept  
b1 <- coef(M.quad)[2] # Hogweed cover linear  
b2 <- coef(M.quad)[3] # Hogwee cover squared  
# In the curve syntax, we need to center and square "x" before multiplying it with  
# the coefficient of Hogweed.cover.sq.  
curve(exp(b0 + b1\*x + b2\*(x-Mean.cover)^2), add=T, col="red", lwd=2)



#### GNM (Generalized Nonlinear Models)

If you find significant non-linear relationships among variables, you can first try transforming (e.g. log, square) the predictor variable. This will not always be enough to resolve non-linearity issues and it can turn p-values of the linearity tests non-significant. If non-linearity is not resolved using this procedure, models such as LM, GLM and GLS are not the appropriate model for this purpose. Indeed, a better way to deal with non-linearity may be to apply a model that uses a nonlinear function of the predictor variables, i.e. on the right-hand side of the model formula. GNM models still require you to specify the distribution family of the dependent variable (e.g. Gaussian, Poisson, Quasi-poisson). Furthermore, it is possible to use an option to coerce a GLM model to a GNM model, as explained at the end of this section.

If the relationship between some of the predictor and the dependent variables is nonlinear, is necessary to use a proper model that accounts for the nonlinear effects.

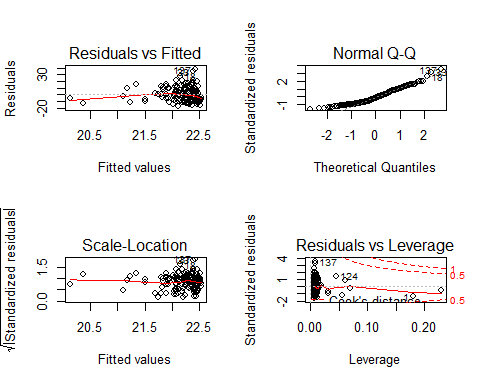
BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG,4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

## Base LM model to apply linearity test  
  
Abundance.pre <- lm(Abundance ~ Recruitment, data=BIOVEG)   
summary(Abundance.pre)

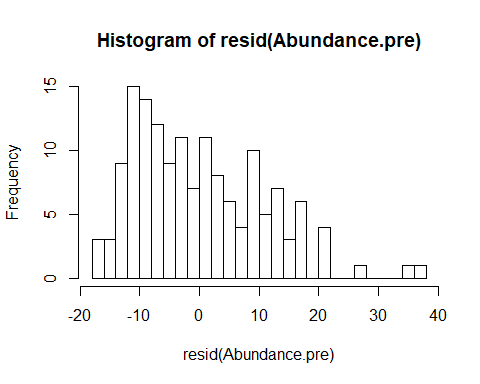
##   
## Call:  
## lm(formula = Abundance ~ Recruitment, data = BIOVEG)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.110 -9.205 -2.062 8.069 37.590   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 22.5720 1.3076 17.262 <2e-16 \*\*\*  
## Recruitment -0.1530 0.3819 -0.401 0.689   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.85 on 148 degrees of freedom  
## Multiple R-squared: 0.001083, Adjusted R-squared: -0.005667   
## F-statistic: 0.1604 on 1 and 148 DF, p-value: 0.6894

## Residuals  
  
x11(width=12, height=12)  
par(mfrow=c(2,2))  
plot(Abundance.pre)

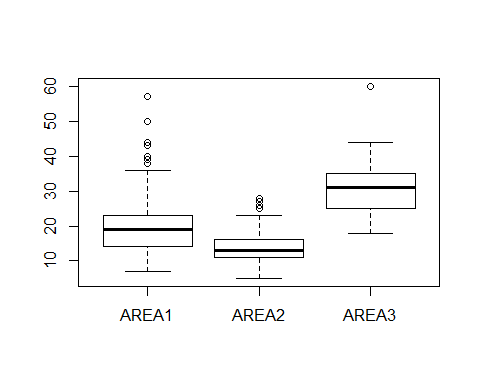


# dev.off() ## turn off the graphics device

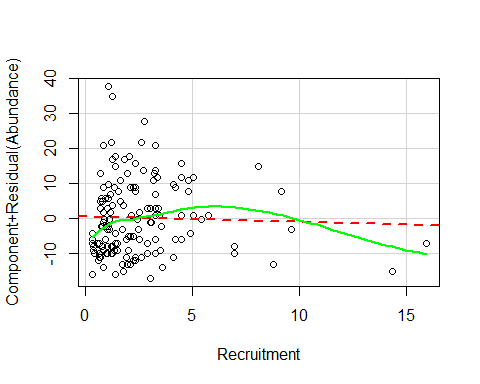
hist(resid(Abundance.pre),breaks=20)



# dev.off() ## turn off the graphics device  
plot(BIOVEG$Site, BIOVEG$Abundance) # variation among sites



### Tests of linearity  
  
library(car)  
  
crPlots(Abundance.pre, col.lines=c("red", "green"))



If one find some nonlinear relationship between predictor and dependent variable using the diagnostic for linearity showed previously here, thus a model that account for nonlinear terms is necessary to carry the statistical prediction.

#### GNM model

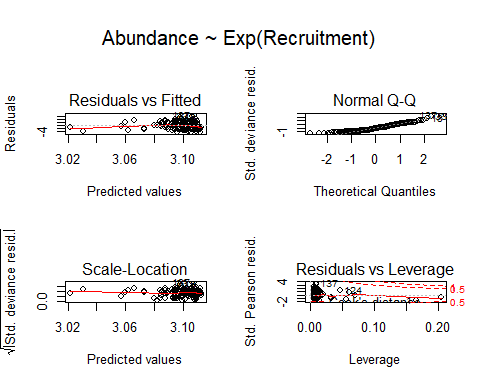
# install.packages("gnm")  
library(gnm)  
  
  
Exp <- function(expression, inst = NULL){list(predictors = list(substitute  
(expression)), term = function(predictors, ...) {paste("exp(", predictors, ")",   
 sep = "")}, call = as.expression(match.call()), match = 1)}   
class(Exp) <- "nonlin"  
  
  
N.gnm<- gnm(Abundance ~ Exp(Recruitment), data=BIOVEG, method = "gnmFit",   
start = NULL, checkLinear = TRUE, verbose = FALSE, family="quasipoisson")  
  
summary(N.gnm)

##   
## Call:  
##   
## gnm(formula = Abundance ~ Exp(Recruitment), family = "quasipoisson",   
## data = BIOVEG, method = "gnmFit", checkLinear = TRUE, start = NULL,   
## verbose = FALSE)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.4003 -2.1178 -0.4718 1.6239 6.5653   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.114609 0.060506 34.949 <2e-16 \*\*\*  
## Exp(.).Recruitment -0.006124 0.018796 -0.326 0.745   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 5.297747)  
##   
## Residual deviance: 758.18 on 148 degrees of freedom  
## AIC: NA  
##   
## Number of iterations: 10

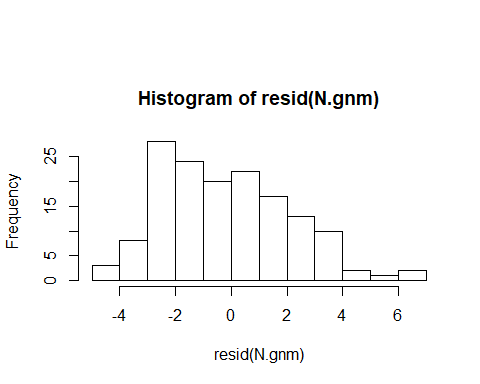
An <- anova(N.gnm, test = "F")  
An

## Analysis of Deviance Table  
##   
## Model: quasipoisson, link: log  
##   
## Response: Abundance  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev F Pr(>F)  
## NULL 149 758.90   
## Exp(Recruitment) 1 0.72765 148 758.18 0.1374 0.7115

## Residuals  
  
x11(width=12, height=12)  
par(mfrow = c(2,2), oma = c(0, 0, 3, 0))  
title <- paste(deparse(N.gnm$formula, width.cutoff = 50), collapse = "\n")  
plot(N.gnm, sub.caption = title)



par(mfrow = c(1,1))  
hist(resid(N.gnm))



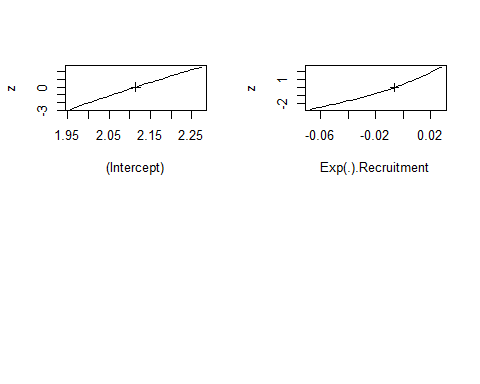
## Deviance profile (optional) - just similar to a summary  
  
prof<- profile(N.gnm, which = ofInterest(N.gnm), alpha = 0.05, trace = TRUE)

##   
## Parameter: (Intercept) down   
##   
## Parameter: (Intercept) up   
##   
## Parameter: Exp(.).Recruitment down   
##   
## Parameter: Exp(.).Recruitment up

prof

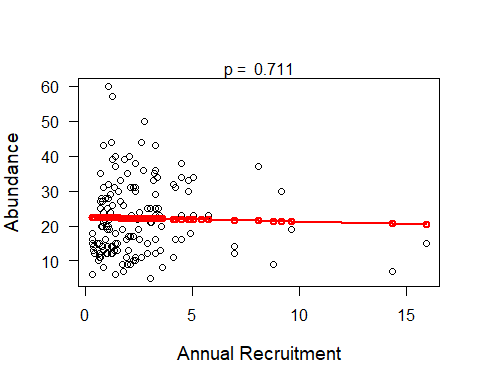
## $`(Intercept)`  
## z par.vals.(Intercept) par.vals.Exp(.).Recruitment  
## 1 -2.8403065 1.956489871 0.020592763  
## 2 -2.1157986 1.996019742 0.015066410  
## 3 -1.3995734 2.035549613 0.008855133  
## 4 -0.6935905 2.075079484 0.001838984  
## 5 0.0000000 2.114609355 -0.006123605  
## 6 0.6788685 2.154139226 -0.015198816  
## 7 1.3405281 2.193669097 -0.025580770  
## 8 1.9823610 2.233198968 -0.037495562  
## 9 2.6016481 2.272728839 -0.051206839  
##   
## $`Exp(.).Recruitment`  
## z par.vals.(Intercept) par.vals.Exp(.).Recruitment  
## 1 -2.6375990 2.241123018 -0.067522297  
## 2 -2.2261319 2.218699511 -0.055242559  
## 3 -1.7670464 2.195032442 -0.042962820  
## 4 -1.2514221 2.169952893 -0.030683082  
## 5 -0.6676060 2.143242561 -0.018403343  
## 6 0.0000000 2.114609355 -0.006123605  
## 7 0.2912927 2.102683614 -0.001272529  
## 8 0.6005749 2.090364248 0.003578546  
## 9 0.9297606 2.077613865 0.008429621  
## 10 1.2810713 2.064388272 0.013280696  
## 11 1.6571012 2.050634724 0.018131772  
## 12 2.0608991 2.036289619 0.022982847  
## 13 2.4960714 2.021275411 0.027833922  
##   
## attr(,"original.fit")  
##   
## Call:  
##   
## gnm(formula = Abundance ~ Exp(Recruitment), family = "quasipoisson",   
## data = BIOVEG, method = "gnmFit", checkLinear = TRUE, start = NULL,   
## verbose = FALSE)  
##   
## Coefficients:  
## (Intercept) Exp(.).Recruitment   
## 2.114609355 -0.006123605   
##   
## Deviance: 758.1769   
## Pearson chi-squared: 784.0666   
## Residual df: 148   
## attr(,"summary")  
##   
## Call:  
##   
## gnm(formula = Abundance ~ Exp(Recruitment), family = "quasipoisson",   
## data = BIOVEG, method = "gnmFit", checkLinear = TRUE, start = NULL,   
## verbose = FALSE)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.4003229 -2.1178163 -0.4718448 1.6238609 6.5653146   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.114609355 0.060506016 34.94875 < 2e-16 \*\*\*  
## Exp(.).Recruitment -0.006123605 0.018795864 -0.32580 0.74504   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 5.297747)  
##   
## Residual deviance: 758.17689 on 148 degrees of freedom  
## AIC: NA  
##   
## Number of iterations: 10  
##   
## attr(,"class")  
## [1] "profile.gnm" "profile.glm" "profile"

x11(width=12, height=8)  
plot(prof)



# dev.off()

## scatterplot   
  
  
x11(width=12, height=12)  
plot(BIOVEG$Abundance ~ BIOVEG$Recruitment, ylab="Abundance",   
xlab= "Annual Recruitment", las = 1, cex.lab=1.2)  
points(BIOVEG$Recruitment, fitted(N.gnm), col = "red", lwd = 2)  
Smooth <- smooth.spline(BIOVEG$Recruitment, fitted(N.gnm))  
lines(Smooth, col = "red", lwd = 2)  
mtext(ifelse(An[2,6]<0.001, "p < 0.001", paste("p = ", round(An[2,6],3))))



# dev.off()

Note 1: If you need to coerce objects of class “glm” to an object of class “gnm” follow the procedure below.

Nall.glm<- glm(Abundance ~ Recruitment, data=BIOVEG, family="quasipoisson")  
N.gnm2<- update(asGnm(Nall.glm))  
N.gnm2

##   
## Call:  
## gnm(formula = Abundance ~ Recruitment, family = "quasipoisson",   
## data = BIOVEG)  
##   
## Coefficients:  
## (Intercept) Recruitment   
## 3.117131 -0.007054   
##   
## Deviance: 758.0411   
## Pearson chi-squared: 783.8135   
## Residual df: 148

anova(N.gnm2, test = "F")

## Analysis of Deviance Table  
##   
## Model: quasipoisson, link: log  
##   
## Response: Abundance  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev F Pr(>F)  
## NULL 149 758.90   
## Recruitment 1 0.86347 148 758.04 0.163 0.687

summary(N.gnm2)

##   
## Call:  
## gnm(formula = Abundance ~ Recruitment, family = "quasipoisson",   
## data = BIOVEG)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.399 -2.119 -0.451 1.632 6.557   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.117131 0.059166 52.684 <2e-16 \*\*\*  
## Recruitment -0.007054 0.017604 -0.401 0.689   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 5.296138)  
##   
## Residual deviance: 758.04 on 148 degrees of freedom  
## AIC: NA  
##   
## Number of iterations: 4

#### GAM (Generalized Aditive Models)

For a GAM model example, let’s assume that we only have fixed effects data(e.g. vegetation type). We will use the same variables as used for the model in a pre- test for linearity ,using the function “crPlots” of the package “car”. Since “P\_mg\_100g” and “K\_mg\_100g” demonstrated the greatest deviation from linearity, we apply a smoothing term (s) for them in the model.

Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

# install.packages("mgcv")  
library(mgcv) # carry out GAM model  
  
spp.gam<- gam(log(Species.richness) ~ s(P\_mg\_100g) + s(K\_mg\_100g) + Veg.cover,   
 data=Hogweed, family="gaussian")  
  
summary(spp.gam)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## log(Species.richness) ~ s(P\_mg\_100g) + s(K\_mg\_100g) + Veg.cover  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.504829 0.075790 33.050 < 2e-16 \*\*\*  
## Veg.cover 0.005250 0.001095 4.795 3.32e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(P\_mg\_100g) 5.987 6.993 3.956 0.000447 \*\*\*  
## s(K\_mg\_100g) 7.544 8.421 2.087 0.050678 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.267 Deviance explained = 32%  
## GCV = 0.19193 Scale est. = 0.1771 n = 201

In the summary, we can see two types of output: parametric coefficients, i.e., the slope of the predictor that indicated a linear relationship and was not subjected to smoothing and the non-linear predictors which were smoothed, showing their F and p-values for significance. The edf refers to the estimated degrees of freedom and basically, the larger the number, the wigglier the fitted model. Values close to 1 indicate a linear term.

anova(spp.gam) # the output is quite similar to the summary

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## log(Species.richness) ~ s(P\_mg\_100g) + s(K\_mg\_100g) + Veg.cover  
##   
## Parametric Terms:  
## df F p-value  
## Veg.cover 1 23 3.32e-06  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value  
## s(P\_mg\_100g) 5.987 6.993 3.956 0.000447  
## s(K\_mg\_100g) 7.544 8.421 2.087 0.050678

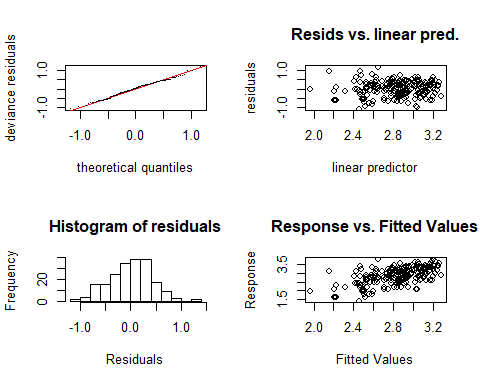
Below we calculate a measure for correlation (analogous to collinearity) known as concurvity, between the predictors in the model, to consider non-linear correlations. The range is from 0 to 1. The closer the value is to 1, the higher the correlation. To know more about concurvity read:

Marra, G. & Wood, S. N. Practical variable selection for generalized additive models. Comput. Stat. Data Anal. 55, 2372-2387 (2011).

concurvity(spp.gam, full=FALSE)

## $worst  
## para s(P\_mg\_100g) s(K\_mg\_100g)  
## para 1.000000e+00 1.496793e-25 4.372779e-26  
## s(P\_mg\_100g) 1.509192e-25 1.000000e+00 9.988476e-01  
## s(K\_mg\_100g) 4.316333e-26 9.988476e-01 1.000000e+00  
##   
## $observed  
## para s(P\_mg\_100g) s(K\_mg\_100g)  
## para 1.000000e+00 5.574905e-28 3.880743e-27  
## s(P\_mg\_100g) 1.509192e-25 1.000000e+00 5.477266e-01  
## s(K\_mg\_100g) 4.316333e-26 3.313584e-01 1.000000e+00  
##   
## $estimate  
## para s(P\_mg\_100g) s(K\_mg\_100g)  
## para 1.000000e+00 2.91690e-28 2.149244e-28  
## s(P\_mg\_100g) 1.509192e-25 1.00000e+00 5.080148e-01  
## s(K\_mg\_100g) 4.316333e-26 4.49389e-01 1.000000e+00

# Residuals  
  
par(mfrow = c(2,2))  
gam.check(spp.gam)



##   
## Method: GCV Optimizer: magic  
## Smoothing parameter selection converged after 8 iterations.  
## The RMS GCV score gradient at convergence was 2.668476e-06 .  
## The Hessian was positive definite.  
## Model rank = 20 / 20   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value  
## s(P\_mg\_100g) 9.00 5.99 1.09 0.83  
## s(K\_mg\_100g) 9.00 7.54 1.14 0.95

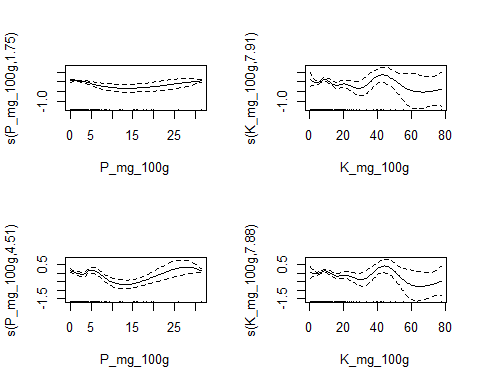
Here the graphical residuals look generally acceptable, since the histogram is not that nice (i.e. symmetrical).

Together with the graphical residuals, the model returns a written output that shows parameters for each predictor with its smoothing term e.g. “k”, “edf”, “K-index” and its p-value. “k-index” is the ratio of neighbor differencing scale estimate to fitted model scale estimate; “edf” is the previously mentioned estimated degrees of freedom and “k’” is the maximum possible EDF for the term. These parameters test whether the basis dimension for smoothing is adequate for the predictor.

For further details regarding this kind of residual diagnostic, please consult the documentation of the function “gam.check” of the package “mgcv”. One way to correctly apply a smoothing term for the predictor is to choose the most appropriate and insert it into the model using the argument “bs”. Look for “smooth.terms” in the mgcv documentation.

In order to test for improvement of the fit of the model, we included the type of smoothing with the argument “bs” and used “K” to choose the number of knots (k) once cubic regression splines have a set number of knots. The default of knots automatically defined by the model “spp.gam” was 9. Let’s test the model fit with 6 and 9 knots. Here we apply “cc” (cyclic cubic regression splines) smoothing to “bs” because of the observed tendency of non-linear fitting in a pre-diangostic graph with the function “crPLots” of package “car”. Here we test of redefine the knots for the predictor P\_mg\_100g because it was diagnosed as being the most non-linear, and improving its fit also may improve the overall fit of the model.

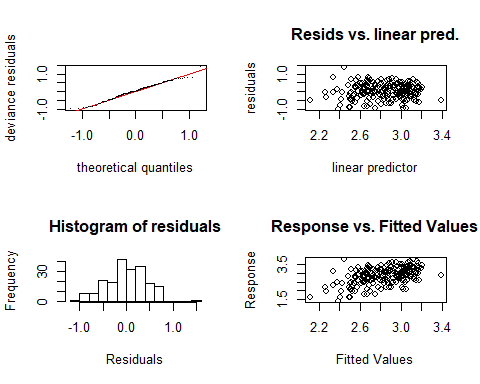
par(mfrow = c(2, 2))  
spp.gam2<- gam(log(Species.richness) ~ s(P\_mg\_100g, bs='cc', k=6) +   
 s(K\_mg\_100g) + Veg.cover, data=Hogweed, family="gaussian")  
plot(spp.gam2)  
spp.gam3<- gam(log(Species.richness) ~ s(P\_mg\_100g, bs='cc', k=9) +   
 s(K\_mg\_100g) + Veg.cover, data=Hogweed, family="gaussian")  
plot(spp.gam3)



We can see that setting models with 6 and 9 knots provides similar model fitting. Thus, the first model (spp.gam) with 9 knots defined automatically by the model is already sufficient for a good model fit.

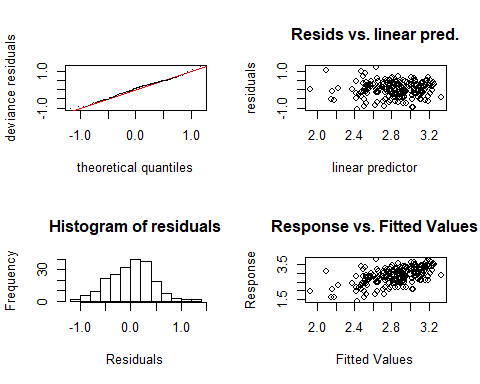
If we still call the residuals for each model test (6 and 9 knots), the degrees of freedom (EDF) of the smoothers are close especially for the predictor “K\_mg\_100g” and they k-index is nonsignificant.

par(mfrow = c(2,2))  
gam.check(spp.gam2)



##   
## Method: GCV Optimizer: magic  
## Smoothing parameter selection converged after 11 iterations.  
## The RMS GCV score gradient at convergence was 1.500108e-07 .  
## The Hessian was positive definite.  
## Model rank = 15 / 15   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value  
## s(P\_mg\_100g) 4.00 1.75 0.99 0.44  
## s(K\_mg\_100g) 9.00 7.91 1.18 0.99

par(mfrow = c(2,2))  
gam.check(spp.gam3)



##   
## Method: GCV Optimizer: magic  
## Smoothing parameter selection converged after 6 iterations.  
## The RMS GCV score gradient at convergence was 1.702051e-07 .  
## The Hessian was positive definite.  
## Model rank = 18 / 18   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value  
## s(P\_mg\_100g) 7.00 4.51 1.08 0.85  
## s(K\_mg\_100g) 9.00 7.88 1.15 0.99

#### GAMM (Generalized Aditive Mixed Models)

To simulate a GAMM model we take the dependent and almost all predictor variables as we used in the previous GAM example. Our intention here is to also include random effects in our modeling, rather than only fixed effects as we did in GAM analysis.

Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

library(mgcv) # carry out GAMM model  
  
spp.gamm<- gamm(log(Species.richness) ~ s(Hogweed.cover) + s(P\_mg\_100g), random =   
 list(Study.area = ~ 1), data=Hogweed, family="gaussian")  
  
summary(spp.gamm)

## Length Class Mode  
## lme 18 lme list  
## gam 31 gam list

Similar to the summary of the GAM model, we have two types of output: parametric coefficients i.e., the slope of the predictor that indicated a linear relationship and was not smoothed and the non-linear predictors with smoothing and their F and p-values for significance.

anova(spp.gamm$gam)

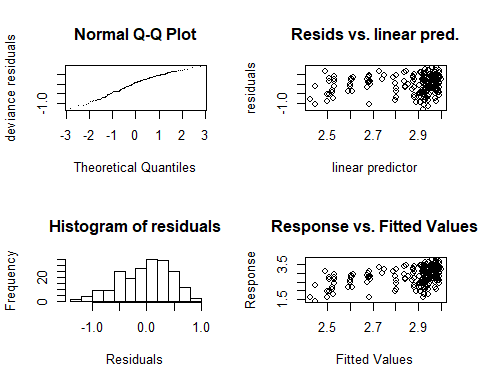
##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## log(Species.richness) ~ s(Hogweed.cover) + s(P\_mg\_100g)  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value  
## s(Hogweed.cover) 2.528 2.528 11.014 4.22e-06  
## s(P\_mg\_100g) 1.000 1.000 0.133 0.715

Below we again conduct a measure for by concurvity between the predictors in the model, to consider non-linear correlations.

concurvity(spp.gamm$gam, full=FALSE)

## $worst  
## para s(Hogweed.cover) s(P\_mg\_100g)  
## para 1.000000e+00 5.664068e-17 1.496793e-25  
## s(Hogweed.cover) 5.664067e-17 1.000000e+00 1.688794e-01  
## s(P\_mg\_100g) 1.479572e-25 1.688794e-01 1.000000e+00  
##   
## $observed  
## para s(Hogweed.cover) s(P\_mg\_100g)  
## para 1.000000e+00 4.617211e-23 4.695046e-33  
## s(Hogweed.cover) 5.664067e-17 1.000000e+00 5.838020e-02  
## s(P\_mg\_100g) 1.479572e-25 1.101885e-01 1.000000e+00  
##   
## $estimate  
## para s(Hogweed.cover) s(P\_mg\_100g)  
## para 1.000000e+00 9.403869e-20 2.916900e-28  
## s(Hogweed.cover) 5.664067e-17 1.000000e+00 4.726491e-02  
## s(P\_mg\_100g) 1.479572e-25 8.170693e-02 1.000000e+00

# Residuals  
par(mfrow = c(2,2))  
gam.check(spp.gamm$gam)



##   
## 'gamm' based fit - care required with interpretation.  
## Checks based on working residuals may be misleading.  
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value   
## s(Hogweed.cover) 9.00 2.53 0.78 0.005 \*\*  
## s(P\_mg\_100g) 9.00 1.00 0.90 0.080 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Here the graphical residuals looks fine in general. However, the predictor “Hogweed.cover” has significant low p-value for k. In order to remember: “k-index” is the ratio of the neighbor differencing scale estimate to fitted model scale estimate; “edf” is the previously mentioned estimated degrees of freedom and “k” is the maximum possible EDF for the term. These parameters test whether the basis dimension for a smoothing is adequate for the predictor.Therefore, here you can try to improve the model by setting different number of knots (k) and maybe choosing another smooth term for this predictor (e.g. “gp” (Gaussian process), “tp” (Thin plate regression splines) and others. Please, check this out searching for “smooth.terms” documentation of the package mgcv.

### SUPERVISED LEARNING

#### Random Forest

The Random Forest model is a supervised learning technique that generates multiple models from a training dataset on the given data and then simply combines their output rules (predictive variables), thus generating a robust high performance model that corrects overfitting and balances variance inequalities. That is, this model improves the predictive power from decision trees, reducing their variances by calculating their averages. The decision tree is a type of modeling that operates with information gain on each node, classifying data points with greater information increment on each node. When all nodes are depleted for their information gain ratings, the model achieves its optimal performance result. Thus, the Random Forest model is considered, in many cases, to be the most robust decision tree approach.

In its results, the Random Forest provides a percentage explanation of the Y variance from the whole set of predictors considered. It also provides the mean square residual error in the predictions from an approach called “Out of Bag Error Estimation (OBB)”, a robust and efficient method of fitting and error estimation.

Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
Hogweed$Plot <- as.integer(Hogweed$Plot) # Transform predictor Plot that is factor  
 # in integer, since random Forest cannot   
 # handle more the 53 categories  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 1 VOL ruderal.grass 2002 0.5 70 18  
## 2 110 VOL ruderal.grass 2002 0.7 65 19  
## 3 126 VOL ruderal.grass 2002 0.4 30 16  
## 4 137 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

# install.packages("randomForest")  
  
library(randomForest)

Below we split our data in train (70% of data) and test (30% of data) sets.

set.seed(100)  
train <- sample(nrow(Hogweed), 0.7\*nrow(Hogweed), replace = FALSE)  
train <- sort(train)  
TrainSet <- Hogweed[train,] # split the data in train set  
TestSet <- Hogweed[-train,] # split the data in test set  
head(TrainSet)# visualize if all variables are present in train set

## Plot Study.area Habitat.type Year Veg.height Veg.cover  
## 2 110 VOL ruderal.grass 2002 0.7 65  
## 3 126 VOL ruderal.grass 2002 0.4 30  
## 4 137 VOL ruderal.grass 2002 0.9 90  
## 5 148 VOL ruderal.grass 2002 0.7 75  
## 8 181 REN ruderal.grass 2002 0.8 90  
## 12 22 REN agr.grassland 2002 0.7 95  
## Species.richness Hogweed.height Hogweed.cover Land.use Disturbance  
## 2 19 1.4 65 NONE TREESHRUB  
## 3 16 1.7 90 NONE TREESHRUB  
## 4 23 1.0 20 NONE NONE  
## 5 27 0.8 20 NONE NONE  
## 8 32 1.1 45 MAINTENANCE NONE  
## 12 27 0.6 5 GRASSLAND NONE  
## Altitude Inclination Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 2 295.653 2 W 0.19 3.56 12.49  
## 3 295.653 2 W 0.19 8.44 39.95  
## 4 292.112 2 W 0.12 2.53 12.34  
## 5 292.112 2 W 0.12 2.53 12.34  
## 8 371.930 2 S 0.21 0.55 6.97  
## 12 377.433 1 S 0.22 7.44 18.27

head(TestSet)# visualize if all variables are present in test set

## Plot Study.area Habitat.type Year Veg.height Veg.cover  
## 1 1 VOL ruderal.grass 2002 0.5 70  
## 6 159 VOL woodland 2002 0.2 5  
## 7 170 REN agr.grassland 2002 0.8 95  
## 9 191 REN ruderal.grass 2002 0.8 90  
## 10 2 REN agr.grassland 2002 0.7 90  
## 11 11 REN tall.herbs 2002 0.5 20  
## Species.richness Hogweed.height Hogweed.cover Land.use Disturbance  
## 1 18 0.9 80 NONE TREESHRUB  
## 6 16 0.8 10 NONE NONE  
## 7 33 1.0 45 GRASSLAND SOILDIS  
## 9 19 1.1 85 MAINTENANCE NONE  
## 10 28 0.8 10 GRASSLAND NONE  
## 11 16 1.8 90 NONE SOILDIS  
## Altitude Inclination Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 295.653 2 W 0.19 3.99 22.27  
## 6 298.394 2 NW 0.16 2.62 10.77  
## 7 394.018 2 S 0.24 5.42 20.46  
## 9 371.930 2 S 0.21 0.55 6.97  
## 10 374.018 0 0 0.22 5.86 15.13  
## 11 379.444 0 0 0.38 0.96 20.34

summary(TrainSet)# visualize if all data for train set is correct, i.e.

## Plot Study.area Habitat.type Year   
## Min. : 4.00 NIE :14 agr.grassland:28 Min. :2002   
## 1st Qu.: 55.75 RUH :14 ruderal.grass:38 1st Qu.:2002   
## Median :103.00 HEL :13 tall.herbs :46 Median :2002   
## Mean :103.11 ENG :11 wasteland :13 Mean :2002   
## 3rd Qu.:150.25 DOM :10 woodland :15 3rd Qu.:2003   
## Max. :200.00 BRE : 9 Max. :2003   
## (Other):69   
## Veg.height Veg.cover Species.richness Hogweed.height   
## Min. :0.1000 Min. : 1.00 Min. : 4.00 Min. :0.2500   
## 1st Qu.:0.3500 1st Qu.:40.00 1st Qu.:13.00 1st Qu.:0.6875   
## Median :0.5000 Median :80.00 Median :18.00 Median :0.9000   
## Mean :0.5539 Mean :64.49 Mean :19.44 Mean :0.9246   
## 3rd Qu.:0.7000 3rd Qu.:90.00 3rd Qu.:25.25 3rd Qu.:1.1000   
## Max. :1.7000 Max. :99.00 Max. :47.00 Max. :1.9000   
##   
## Hogweed.cover Land.use Disturbance Altitude   
## Min. : 1.00 GRASSLAND :18 DEPORG : 6 Min. :127.6   
## 1st Qu.:10.00 MAINTENANCE:26 FLOOD :13 1st Qu.:208.0   
## Median :20.00 NONE :96 NONE :99 Median :322.0   
## Mean :34.55 SOILDIS : 7 Mean :357.9   
## 3rd Qu.:60.00 TREESHRUB:15 3rd Qu.:439.5   
## Max. :95.00 Max. :977.0   
##   
## Inclination Exposition N\_perc P\_mg\_100g   
## Min. : 0.0 0 :95 Min. :0.0400 Min. : 0.000   
## 1st Qu.: 0.0 W :12 1st Qu.:0.2100 1st Qu.: 0.525   
## Median : 0.0 E :10 Median :0.2700 Median : 1.405   
## Mean : 1.4 SW : 8 Mean :0.2853 Mean : 3.128   
## 3rd Qu.: 2.0 S : 5 3rd Qu.:0.3425 3rd Qu.: 3.377   
## Max. :30.0 N : 4 Max. :1.0300 Max. :31.420   
## (Other): 6   
## K\_mg\_100g   
## Min. : 1.440   
## 1st Qu.: 5.728   
## Median : 8.329   
## Mean :11.639   
## 3rd Qu.:13.135   
## Max. :77.769   
##

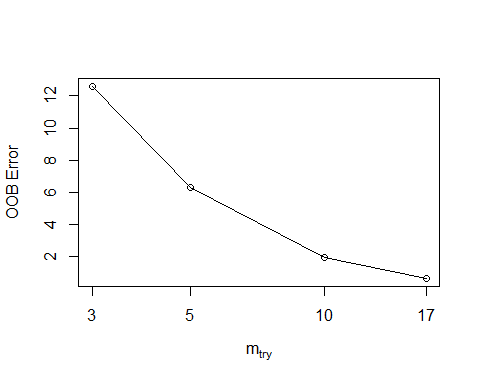
# Min, Median, Mean, Max...  
summary(TestSet) # visualize if all data for test set is correct

## Plot Study.area Habitat.type Year   
## Min. : 1.00 GOE : 6 agr.grassland: 8 Min. :2002   
## 1st Qu.: 46.00 NIE : 6 ruderal.grass:15 1st Qu.:2002   
## Median :100.00 WAT : 6 tall.herbs :31 Median :2002   
## Mean : 96.16 RUH : 5 wasteland : 3 Mean :2002   
## 3rd Qu.:156.00 WEN : 5 woodland : 4 3rd Qu.:2003   
## Max. :201.00 DAH : 4 Max. :2003   
## (Other):29   
## Veg.height Veg.cover Species.richness Hogweed.height   
## Min. :0.2000 Min. : 5.0 Min. : 5.00 Min. :0.300   
## 1st Qu.:0.4000 1st Qu.:40.0 1st Qu.:13.00 1st Qu.:0.800   
## Median :0.5000 Median :70.0 Median :17.00 Median :1.000   
## Mean :0.5893 Mean :61.9 Mean :18.33 Mean :1.023   
## 3rd Qu.:0.7000 3rd Qu.:90.0 3rd Qu.:24.00 3rd Qu.:1.200   
## Max. :1.4000 Max. :98.0 Max. :46.00 Max. :2.400   
##   
## Hogweed.cover Land.use Disturbance Altitude   
## Min. : 5.00 GRASSLAND : 6 DEPORG : 5 Min. :104.5   
## 1st Qu.:15.00 MAINTENANCE: 7 FLOOD : 8 1st Qu.:206.9   
## Median :40.00 NONE :48 NONE :36 Median :316.1   
## Mean :43.11 SOILDIS : 7 Mean :329.7   
## 3rd Qu.:75.00 TREESHRUB: 5 3rd Qu.:355.0   
## Max. :90.00 Max. :978.0   
##   
## Inclination Exposition N\_perc P\_mg\_100g   
## Min. : 0.000 0 :28 Min. :0.0900 Min. : 0.020   
## 1st Qu.: 0.000 S : 8 1st Qu.:0.2000 1st Qu.: 0.820   
## Median : 1.000 W : 8 Median :0.2500 Median : 2.280   
## Mean : 1.918 E : 5 Mean :0.2721 Mean : 3.431   
## 3rd Qu.: 2.000 N : 5 3rd Qu.:0.3400 3rd Qu.: 4.040   
## Max. :25.000 NW : 2 Max. :0.7200 Max. :26.010   
## (Other): 5   
## K\_mg\_100g   
## Min. : 0.75   
## 1st Qu.: 6.32   
## Median : 7.90   
## Mean :12.39   
## 3rd Qu.:14.78   
## Max. :47.74   
##

Verify the best number of predictors (mtry) per node (split) in the branches of the decision trees. The function “mtry” shows from which number of predictors per node the prediction errors are lower, based on OBB error (Out of bag) estimation.

# check how many predictors per node reduce error:  
  
tuneRF(Hogweed, Hogweed$Species.richness)

## mtry = 5 OOB error = 6.304418   
## Searching left ...  
## mtry = 3 OOB error = 12.58651   
## -0.9964583 0.05   
## Searching right ...  
## mtry = 10 OOB error = 1.960619   
## 0.6890087 0.05   
## mtry = 17 OOB error = 0.6492911   
## 0.6688337 0.05



## mtry OOBError  
## 3 3 12.5865073  
## 5 5 6.3044180  
## 10 10 1.9606194  
## 17 17 0.6492911

The outcome shows that increasing the number of predictors per node of the trees decreases the error. Thus, we include all predictors (16) per node in the model.

#### Random Forest Model

set.seed(100)  
RF1 <- randomForest(Species.richness ~ ., data=Hogweed,  
 subset=train, mtry = 16, importance=TRUE, ntree=500)

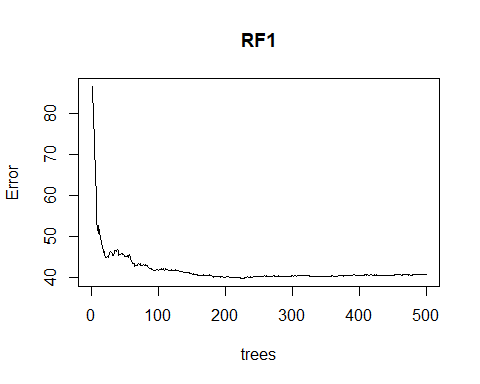
print(RF1)

##   
## Call:  
## randomForest(formula = Species.richness ~ ., data = Hogweed, mtry = 16, importance = TRUE, ntree = 500, subset = train)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 16  
##   
## Mean of squared residuals: 40.6848  
## % Var explained: 47.48

The model with train subset performed poorly regarding the percentage of variance explained in the prediction of species richness by the selected set of predictors.

Let’s see if the number of decision (500) chosen for the model were enough to reduce the errors:

plot(RF1)



The output shows that 500 trees are sufficient to reduce error, since from approximately 100 trees, the error starts to decrease.

#### Importance of predictors

Here we check which predictors are the least and most important when predicting the variation of the dependent variable. The variable importance in Random Forest is computed based on Node Purity and Mean squared error. Node purity measures the total decrease of impurities in the nodes (splits) by computing the average of each predictor over the nodes of all trees, being measured, in regression, by the sum of the squares residuals. Of these two approaches for importance, Mean squared error (%IncMSE) is the most robust and informative measure, where the higher number, the more important the variable. Conversely, IncNodePurity is biased and should only be used if the extra computation time of calculating %IncMSE is challenging. Since the %IncMSE is calculated based on OBB estimation, the expectation is that the MSE will increase, especially if the variable has some importance, thus the label “% IncMSE”.

import<- importance(RF1)  
import

## %IncMSE IncNodePurity  
## Plot 5.5707559 294.35118  
## Study.area 37.1460387 5757.91498  
## Habitat.type 10.9296666 558.60044  
## Year 1.7781753 5.62926  
## Veg.height 4.6763764 250.32433  
## Veg.cover 12.7326538 652.26748  
## Hogweed.height 10.7277820 627.23769  
## Hogweed.cover 2.2307121 341.24214  
## Land.use -0.8118798 27.13485  
## Disturbance 8.2167514 221.78219  
## Altitude 6.3721603 387.67738  
## Inclination 1.0159853 66.17019  
## Exposition 6.1613376 228.69195  
## N\_perc 0.4512169 389.86253  
## P\_mg\_100g 3.2757968 362.76719  
## K\_mg\_100g 3.2115987 378.80948

The outcome states that the most important predictors based on %IncMSE. Remember that the higher number, the more important the variable.

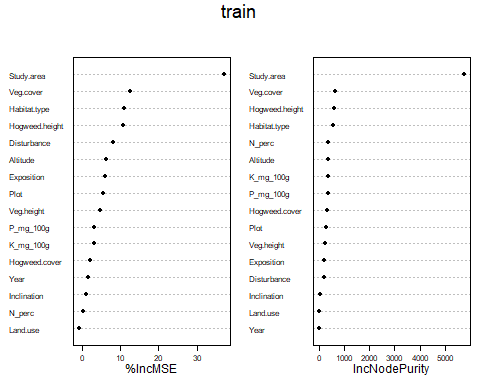
### save output of importance   
  
write.table(import, "Importance.csv", row.names=T, sep=",", dec=".")

Note: Prior to selecting the predictor variables for the Random Forest model, verify whether there is high correlation among them. The presence of highly correlated predictors may bias the computation of importance of each one, for predicting the dependent variable, leading to suboptimal predictor variables being artificially and arbitrarily preferred. For more information about this topic, please read the recommended papers below:

Strobl C, Boulesteix AL, Zeileis A, Hothorn T.Bias in random forest variable importance measures: illustrations, sources and a solution. BMC Bioinformatics, 2007 Jan 25;8:25

Strobl C et al. Conditional variable importance for random forests. BMC Bioinformatics, BMC Bioinformatics. 2008 Jul 11;9:307

# Shows and save importance graphically  
  
# library(lattice) # save the figure:  
# tiff(filename="Fig\_IMPORTANCE2019.jpg", width=180, height=80, units="mm", res=300)  
  
varImpPlot(RF1, sort=TRUE, main=deparse(substitute(train)),   
 cex.lab=1.5, las=1, bty="n", cex=0.5,pch=19, col='black')



# dev.off() ## turn off the graphics device

#### Comparison of the level of error

Here we compare the level of error of the model in relation to the number of predictors per node for the train (70%) and test (30%) set.

The OBB (out of bag error estimation) is computed to the train set and compared to the test set (test.err).

Here we simulate whether the inclusion of all predictors per node of the trees delivers a better or worse performance for the random forest prediction of both train and test set. Then, we reduce the number of trees from 500 to 400 to better fit the whole number of predictors per node.

oob.err=double(16) # 16 regards to total number of predictors  
test.err=double(16)  
  
for(mtry in 1:16)   
{  
 rf=randomForest(Species.richness ~ . , data = Hogweed , subset = train,mtry=mtry,ntree=400)   
 oob.err[mtry] = rf$mse[400] # Error of all trees adjusted  
   
 pred<-predict(rf,Hogweed[-train,]) # Prediction in test set for each tree  
 # Mean Squared Test Error:  
 test.err[mtry]= with(Hogweed[-train,], mean( (Species.richness - pred)^2))  
   
 cat(mtry," ") # show the outcome in R console  
   
}

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

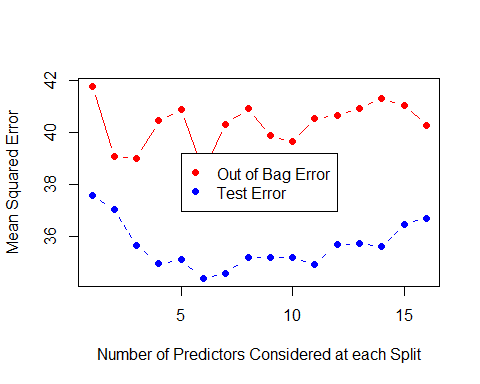
test.err

## [1] 37.56997 37.04643 35.65082 34.96186 35.11245 34.40052 34.58925  
## [8] 35.19916 35.18508 35.20572 34.94613 35.69522 35.74131 35.61271  
## [15] 36.48326 36.70819

oob.err

## [1] 41.77874 39.09036 39.01873 40.45448 40.88128 38.47648 40.33007  
## [8] 40.92160 39.89282 39.65230 40.56144 40.65620 40.93758 41.32343  
## [15] 41.05698 40.28447

matplot(1:mtry , cbind(oob.err,test.err), pch=19 , col=c("red","blue"),type="b",  
 ylab="Mean Squared Error",xlab="Number of Predictors Considered at each Split")  
legend("center",legend=c("Out of Bag Error","Test Error"),pch=19, col=c("red","blue"))



The graph shows that the mean squared error increases slightly in both Out of bag error and test error as the number of predictors increases at each split (node).

In addition, we can check the performance using cross-validation prediction of models with the number of predictors reducing sequentially. In such reduction, the predictors are ranked by variable importance via a nested cross-validation procedure.

rfcv(Hogweed, Hogweed$Species.richness, cv.fold=5)

## $n.var  
## [1] 17 8 4 2 1  
##   
## $error.cv  
## 17 8 4 2 1   
## 8.1906159 7.5375960 2.3589484 2.8383323 0.2002621   
##   
## $predicted  
## $predicted$`17`  
## [1] 17.672433 18.030300 15.337233 22.282800 24.021633 16.897033 28.913267  
## [8] 30.135833 21.520233 26.856300 16.746733 27.260633 15.721633 13.453567  
## [15] 16.507900 19.716600 19.747133 15.485233 23.652200 16.815933 24.503100  
## [22] 22.552900 17.388367 14.350433 20.736033 19.691367 17.359300 22.121633  
## [29] 23.384767 16.301267 18.328810 16.987533 21.580133 28.771967 29.188067  
## [36] 26.065800 26.761333 26.772533 17.291400 14.992867 17.186533 17.125033  
## [43] 15.659167 27.405367 25.042533 18.814633 26.539867 24.301533 23.922633  
## [50] 16.988333 23.194133 18.075133 20.801600 20.664733 13.250767 21.020300  
## [57] 16.391500 23.097700 15.034967 23.521000 19.750467 19.035967 19.168633  
## [64] 12.197600 34.648433 37.296667 34.424167 35.882700 18.934800 18.836767  
## [71] 33.501167 33.183117 33.556867 35.460467 32.355019 31.523833 31.131150  
## [78] 35.963300 34.821252 24.151800 18.554567 19.276467 26.590133 17.334467  
## [85] 23.964333 33.287267 34.879767 30.707700 28.321700 11.301900 27.049000  
## [92] 20.093867 15.456833 21.126567 26.666367 17.503700 11.657867 10.735600  
## [99] 25.840033 12.995600 24.447533 13.827533 17.571000 13.038833 13.499533  
## [106] 25.906633 13.412100 12.218733 12.546467 17.482067 16.671900 11.202300  
## [113] 19.190967 21.911667 16.418367 25.308133 25.034100 18.914567 12.792400  
## [120] 15.181300 16.227733 13.116067 16.928833 13.013033 9.636533 19.637000  
## [127] 20.578267 20.131900 21.181867 14.218033 16.712400 12.863433 12.921967  
## [134] 13.020267 10.116367 23.466033 20.984467 19.057300 10.998900 10.120700  
## [141] 9.287533 18.225333 9.547233 9.134967 8.096767 10.089367 8.693533  
## [148] 10.924900 9.687600 24.000233 14.142300 13.719800 12.156467 12.001210  
## [155] 20.968167 20.483967 15.667233 11.203767 19.519567 12.295833 13.258667  
## [162] 13.845200 14.621967 15.481133 19.361033 12.083700 19.578633 13.317233  
## [169] 14.865433 9.688533 15.565467 22.156700 13.765967 13.456500 18.772900  
## [176] 19.198500 24.825433 24.813933 24.400967 15.700033 15.560467 12.014700  
## [183] 12.644733 16.194233 11.878667 13.769267 24.687300 17.350267 17.200533  
## [190] 15.981167 23.297533 17.945867 16.097367 24.255900 17.967033 11.776433  
## [197] 8.376100 10.382467 9.064367 13.487167 9.122070  
##   
## $predicted$`8`  
## [1] 18.116733 18.126933 15.723933 22.454233 24.285667 16.398400 29.517467  
## [8] 29.963467 22.252033 27.218952 17.309400 27.075233 15.536100 12.620533  
## [15] 16.625233 20.241433 19.972967 15.291467 24.122667 16.771933 24.833400  
## [22] 22.103367 16.993800 14.029033 21.285200 19.938267 17.170667 21.810833  
## [29] 23.199733 16.683500 18.138267 16.906467 21.281517 29.320400 29.480067  
## [36] 26.056467 27.337767 26.539644 17.502833 14.980733 17.348533 17.096700  
## [43] 16.152967 26.927667 24.674417 17.953719 26.677167 24.512167 23.674633  
## [50] 16.476633 23.604467 18.917233 20.689300 21.145200 14.290233 21.287433  
## [57] 16.929605 22.788733 14.713551 23.325417 19.193729 19.671600 19.142833  
## [64] 11.580233 34.646300 37.674133 33.600860 36.438800 18.194833 17.889467  
## [71] 34.581089 33.990619 33.871003 35.782237 31.630133 31.568895 30.632357  
## [78] 36.010570 35.352333 23.402848 17.586800 18.367733 27.925500 17.136033  
## [85] 23.778133 33.513500 35.149200 30.216333 28.386048 11.168900 26.099367  
## [92] 20.632433 15.224500 20.282500 26.349352 16.572314 10.853590 10.018400  
## [99] 26.105267 12.398033 24.829967 13.971211 17.908500 12.965700 12.644400  
## [106] 25.938133 12.814267 11.794100 12.365571 17.487189 16.183822 10.824133  
## [113] 19.566083 21.403833 16.790533 25.751767 25.046300 18.981100 12.734217  
## [120] 15.307400 16.545148 13.253267 17.106744 12.791500 9.175417 19.800333  
## [127] 20.900289 19.650748 20.956702 13.157933 16.599721 12.890300 13.050700  
## [134] 13.447033 10.493267 23.443367 20.432548 18.341200 10.189233 9.109056  
## [141] 8.470224 17.970205 8.876167 8.468467 7.727737 9.864000 7.637333  
## [148] 11.161133 10.037378 23.758833 13.002300 12.467600 13.099267 11.540633  
## [155] 20.825200 20.320633 15.355167 11.915652 20.090500 12.285867 13.461900  
## [162] 14.167667 15.082633 15.264667 19.589133 12.044767 19.478533 12.663900  
## [169] 14.029000 9.381533 15.104767 22.091367 13.700367 12.204567 19.020767  
## [176] 18.804800 26.312500 25.430600 24.673867 15.513867 15.223122 11.019733  
## [183] 11.540633 16.449700 11.563200 13.683000 24.501300 17.617633 17.253200  
## [190] 16.538467 23.473733 18.063133 16.176200 23.753533 18.022900 11.770067  
## [197] 8.699633 11.045514 8.279405 12.382867 8.730467  
##   
## $predicted$`4`  
## [1] 17.750167 18.466400 16.463486 22.696867 25.482767 16.509300 30.831333  
## [8] 31.563824 19.849000 27.984700 17.509233 27.711600 16.321033 12.082710  
## [15] 16.505933 19.929367 19.960767 13.345500 24.779633 16.891033 25.840567  
## [22] 23.781867 18.819767 13.786767 22.185633 19.621900 18.111729 22.974833  
## [29] 23.548717 14.315633 17.749367 17.434067 22.896033 30.233400 30.114533  
## [36] 24.698967 27.248367 28.303600 16.544633 14.148200 13.839567 15.686333  
## [43] 15.941600 28.901067 24.559567 16.581233 28.406767 24.902133 23.747433  
## [50] 15.290767 23.963933 18.996833 20.171867 20.638967 10.823533 22.096967  
## [57] 15.988533 23.364100 12.915667 24.937500 19.775067 19.495833 19.332933  
## [64] 10.041567 38.423719 39.251933 33.473267 37.672843 17.778433 17.875167  
## [71] 36.410476 32.496133 33.923733 37.679667 32.597395 32.953867 29.819300  
## [78] 37.929033 37.975662 27.130800 16.481833 17.497656 27.634400 14.211567  
## [85] 26.671700 36.368010 33.783367 31.540433 30.491233 7.590767 29.423967  
## [92] 20.375567 14.521233 23.098233 27.788167 11.939467 8.732333 7.345867  
## [99] 27.775033 10.179133 24.565400 12.338067 14.354389 13.715000 11.535067  
## [106] 27.672767 12.465609 9.570033 11.614776 17.443833 17.251300 10.133972  
## [113] 19.850408 22.378731 18.067008 28.004567 25.957386 19.995667 13.836188  
## [120] 14.856810 16.913900 13.061010 17.246547 12.163976 10.102440 22.359767  
## [127] 21.681471 20.255424 21.866667 12.123310 17.356333 11.366974 11.571700  
## [134] 12.514600 8.733243 25.397267 22.448857 18.832767 11.164756 9.326433  
## [141] 7.127914 20.512133 7.124732 8.702437 6.914995 9.219770 6.767932  
## [148] 8.740200 10.283953 26.025367 13.127467 8.618733 11.518867 12.318400  
## [155] 21.395200 20.650800 14.775233 11.834176 20.594133 8.933057 11.965867  
## [162] 12.787900 14.311533 15.684467 19.305100 11.397092 20.404967 13.187200  
## [169] 12.799267 7.679048 16.472038 23.254933 14.022300 8.728633 19.046300  
## [176] 19.552300 27.749967 27.810200 24.189333 15.367367 14.879600 10.721067  
## [183] 12.573667 15.732900 9.634867 12.902543 28.063500 17.034067 16.789124  
## [190] 16.160500 24.752733 17.044157 15.420533 26.447033 18.000033 9.768800  
## [197] 6.171900 8.501300 6.115067 9.768800 8.204167  
##   
## $predicted$`2`  
## [1] 17.787040 18.181932 16.632162 21.921698 24.699109 16.639488 30.531218  
## [8] 30.576731 19.284859 28.418904 19.138840 27.493669 16.360752 12.651008  
## [15] 16.861102 19.802243 19.869078 13.436114 24.217259 16.995212 25.390292  
## [22] 24.036283 18.841096 13.961400 21.874063 19.720772 18.089957 22.938552  
## [29] 23.477464 14.321937 17.238112 17.238112 22.868158 29.968024 29.862916  
## [36] 24.681682 27.274289 28.606017 16.974477 14.218342 14.465759 15.687248  
## [43] 16.230523 29.081856 24.696822 16.974477 28.424022 25.165656 23.877056  
## [50] 15.290838 23.792054 18.852658 20.010010 20.522011 10.359489 21.901176  
## [57] 15.802248 23.725885 13.380539 25.235175 19.920152 19.797138 19.091288  
## [64] 10.213443 38.705436 38.274362 33.322843 37.466634 18.894488 19.215907  
## [71] 38.074276 33.121619 33.941686 37.535152 32.143752 33.168057 30.853352  
## [78] 37.720419 37.777322 27.926648 16.635544 17.820417 28.139450 14.936017  
## [85] 25.674017 35.336478 33.607119 31.410933 30.327167 7.934058 28.560792  
## [92] 20.719517 13.810440 23.403379 27.331679 8.948105 8.740511 7.679971  
## [99] 27.913372 10.369589 24.586772 12.220422 12.651138 13.686083 11.401056  
## [106] 27.372650 12.813737 10.025589 11.155855 17.727375 17.727375 9.724002  
## [113] 20.101996 22.572533 17.975740 28.094633 25.906639 19.967648 13.582827  
## [120] 14.807249 16.818869 13.143854 16.775789 12.199573 9.769508 22.605948  
## [127] 21.846125 20.277087 21.746721 12.199573 17.196954 11.310406 11.454073  
## [134] 12.929044 9.541178 24.918511 22.396997 18.344693 11.028118 9.097739  
## [141] 6.729909 20.460728 7.251395 8.802724 7.049811 9.287457 6.709190  
## [148] 7.206243 10.301729 25.372677 12.463781 8.442183 11.521914 12.463781  
## [155] 21.181562 21.002762 14.363596 11.591981 20.918180 8.727583 11.729165  
## [162] 12.387132 13.879798 15.816133 19.492891 11.671132 19.900694 12.813791  
## [169] 13.014660 8.331275 16.447662 22.042219 13.222043 9.309317 18.484167  
## [176] 19.805973 27.394183 27.616381 23.723647 14.920914 14.891754 11.314526  
## [183] 12.995476 16.192886 10.997498 13.652834 27.724741 16.894200 16.743327  
## [190] 16.082700 25.050559 16.743327 15.487371 27.632429 17.422239 8.728471  
## [197] 6.789937 7.802499 6.277764 8.728471 8.423148  
##   
## $predicted$`1`  
## [1] 18.007267 18.912267 15.998000 22.984867 26.946633 16.000000 32.948233  
## [8] 32.009600 18.971067 27.851233 15.992667 26.982400 16.000000 10.992200  
## [15] 17.000000 19.999500 19.999200 13.002000 25.020267 17.000400 26.946633  
## [22] 25.028800 18.966167 12.999200 21.998867 19.999500 18.013467 22.999200  
## [29] 24.012667 13.974700 17.000400 17.000400 22.991733 32.012933 31.018900  
## [36] 24.000933 26.982133 29.001433 16.999400 14.008900 12.999200 14.998700  
## [43] 15.992667 28.984400 23.994333 16.999400 27.915000 25.986633 24.002633  
## [50] 14.998700 24.012667 18.912267 19.999200 20.884667 9.009600 21.993033  
## [57] 15.992667 24.000933 13.002000 25.941600 19.999500 20.000000 18.963867  
## [64] 7.966500 43.760033 39.087567 30.010533 39.282800 15.992667 14.978900  
## [71] 39.685600 30.956600 32.883100 39.025367 31.963233 33.236500 25.941600  
## [78] 39.282067 37.749400 28.998067 16.000000 17.000000 29.956367 11.998000  
## [85] 25.985967 42.349167 33.008267 32.861333 30.988533 6.728400 35.017633  
## [92] 20.884667 14.995767 22.991733 27.851233 8.999900 9.009600 5.289000  
## [99] 29.009233 9.009600 24.980967 11.996400 12.997300 14.001100 10.983067  
## [106] 27.905300 13.000000 7.996067 10.967867 18.008267 18.008267 9.667167  
## [113] 19.993600 21.997967 18.019567 30.014967 25.982000 19.999200 13.991600  
## [120] 15.000000 17.000400 12.999200 16.999400 11.996800 9.667167 22.993633  
## [127] 21.998867 20.884667 22.007867 11.996800 18.008267 10.986467 10.992200  
## [134] 12.997300 5.289000 27.031100 22.996033 19.993600 11.995333 9.009600  
## [141] 5.806033 21.994767 6.728400 9.002433 6.927633 9.892267 5.101500  
## [148] 6.864700 10.986467 29.009333 12.997300 5.725900 10.967867 12.997300  
## [155] 21.994767 20.993000 14.999600 12.002500 20.884667 7.978767 10.983067  
## [162] 11.996400 14.008900 15.992667 19.993600 10.967867 20.993000 12.999200  
## [169] 12.997300 5.756767 16.999400 24.998867 13.974700 7.966500 19.999500  
## [176] 19.993600 28.998067 29.956367 23.994333 14.995767 14.978900 7.960967  
## [183] 13.000000 17.000000 6.728400 12.002500 29.001433 17.000000 16.999400  
## [190] 15.998000 24.990767 16.999400 14.978900 29.009233 19.993600 9.002433  
## [197] 5.101500 7.985400 4.953067 9.002433 8.997300

The section “$error.cv” in the output show us that reducing number of predictors decreases the rates of error in the performance of the model. This outcome make sense and is congruent with large amount of predictors with low importance for the model prediction observed above.

#### Prediction performance

In order to verify the percentage of explanation for the Random Forest model we can adjust a linear model to get a value for R-squared. We do this for the test set created above. If 30% of data from the test produces a high performance, your prediction is good.

predTest <- predict(RF1, TestSet)  
r2\_test <- lm(predTest ~ TestSet$Species.richness)  
summary(r2\_test)

##   
## Call:  
## lm(formula = predTest ~ TestSet$Species.richness)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.3940 -3.0597 -0.1885 2.3862 10.6701   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.04786 1.25847 8.779 2.67e-12 \*\*\*  
## TestSet$Species.richness 0.44538 0.06282 7.090 1.91e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.968 on 59 degrees of freedom  
## Multiple R-squared: 0.46, Adjusted R-squared: 0.4509   
## F-statistic: 50.26 on 1 and 59 DF, p-value: 1.907e-09

The adjusted R-squared shows that the prediction performance of the model is not good, since it predicts less than 50 % of the dependent variable.

Nevertheless, you do not need necessarily to use a linear model to check on prediction performance. You could also calculate the correlation coefficient and, then, square it:

cor(predTest, TestSet$Species.richness) ^ 2

## [1] 0.4600236

#### RMSE

We can adjust a RMSE (Root mean squared error) for the model. The RMSE is particularly useful when it is compared with other models to see if it performs better or worse depending if it has a higher or lower error of prediction.

rfvalpred = predict(RF1,newdata=TestSet)  
rmse = sqrt(mean((TestSet$Species.richness-rfvalpred)^2))  
rmse

## [1] 6.010385

### Model Selection (Akaike)

For both Akaike and GLM / GLMM the predictor variables must be tested for the degree of correlation, because if these variables are highly correlated, it implies collinearity, and the real contributions of each predictor to the dependent variable may be confounded. One conventional way to test correlation, and collinearity, is the Pearson’s correlation (described in this script). Pearson correlation coefficients can be between -1 and 1, and the accepted limit for collinearity is |0.60|. Correlations above 0.60 indicate too much collinearity between predictors. On the other hand, many authors accept correlations up to 0.70 and use variables with this degree of correlation in the model.

Akaike’s function: select the best regression model according to the relationship between the dependent and predictor variables. The best model is normally assumed to be the one with the lowest AICc value. However, the quantity of models considered to be good does not necessarily follow a specific rule, since it depends on your purpose and research approach. It also can depend on the statistical line you follow. Some researchers suggest that a difference between AIC values smaller than 2 (i.e. delta AIC =< 2) indicates that the models are almost equally good. Others use a threshold level of delta AIC =< 4.

#### GLM’s Akaike selection

## Akaike: basic syntax  
  
## install and load "MuMIn" package  
  
# Usage  
  
Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

# install.packages("MuMIn")  
library(MuMIn)  
options(na.action = "na.fail")  
div <- glm(Veg.height ~ P\_mg\_100g + K\_mg\_100g + Land.use, family=gaussian,   
 data=Hogweed)  
tested.div1 <- dredge(div)  
tested.div1

## Global model call: glm(formula = Veg.height ~ P\_mg\_100g + K\_mg\_100g + Land.use,   
## family = gaussian, data = Hogweed)  
## ---  
## Model selection table   
## (Int) K\_mg\_100 Lnd.use P\_mg\_100 df logLik AICc delta weight  
## 7 0.3256 + 0.006875 5 -23.238 56.8 0.00 0.414  
## 3 0.3396 + 4 -24.592 57.4 0.60 0.306  
## 8 0.3330 -0.0011540 + 0.008192 6 -23.093 58.6 1.83 0.165  
## 4 0.3336 0.0006829 + 5 -24.525 59.4 2.57 0.114  
## 5 0.5353 0.009135 3 -32.343 70.8 14.02 0.000  
## 6 0.5399 -0.0005695 0.009789 4 -32.310 72.8 16.04 0.000  
## 1 0.5647 2 -34.565 73.2 16.41 0.000  
## 2 0.5450 0.0016610 3 -34.197 74.5 17.73 0.000  
## Models ranked by AICc(x)

#### LME’s Akaike selection

Firstly you must carry out the pre-tests with LM in the model to be executed as LME model. If normality and linearity are accepted, then, you check for spatial autocorrelation by Moran’s I. If there is significant autocorrelation you must model include a spatial correlation structure (e.g. corExp) within the global LME that will be submited to selection by Akaike such as the example below.

Example of LM model to check for linearity and autocorrelation for the global LME model to execute Akaike’s selection:

Firstly you must conduct pre-tests with LMs in the model to be executed as LME model. If normality and linearity are acceptable, you must then check for spatial autocorrelation by Moran’s I. If there is significant autocorrelation your model must include a spatial correlation structure (e.g. corExp) within the global LME, which will be submitted to selection by Akaike criteria, as described in the example below.

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG,4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

Mean.Diameter.pre <- lm(Mean\_diameter ~ Mortality + Recruitment + Richness + Site,   
 data=BIOVEG)

## LME's Akaike selection  
  
library(nlme) # LME models  
library(MuMIn) # Dredge function to execute the selection  
  
options(na.action = "na.fail")  
div <- lme(Mean\_diameter ~ Mortality + Recruitment + Richness, data=BIOVEG, method="ML", random= ~1|Site, corr=corExp(form=~local.utm.x+local.utm.y|Site),   
 weights=varIdent(form=~1|Site), na.action=na.fail)  
  
  
tested.div1 <- dredge(div)  
tested.div1

## Global model call: lme.formula(fixed = Mean\_diameter ~ Mortality + Recruitment +   
## Richness, data = BIOVEG, random = ~1 | Site, correlation = corExp(form = ~local.utm.x +   
## local.utm.y | Site), weights = varIdent(form = ~1 | Site),   
## method = "ML", na.action = na.fail)  
## ---  
## Model selection table   
## (Intrc) Mrtlt Rcrtm Rchns df logLik AICc delta weight  
## 1 10.67 6 -40.724 94.0 0.00 0.403  
## 2 10.67 0.001550 7 -40.609 96.0 1.97 0.150  
## 3 10.67 -0.002797 7 -40.678 96.1 2.11 0.140  
## 5 10.68 -0.0007914 7 -40.686 96.2 2.13 0.139  
## 4 10.67 0.001605 -0.003023 8 -40.555 98.1 4.10 0.052  
## 6 10.67 0.001428 -0.0005343 8 -40.593 98.2 4.17 0.050  
## 7 10.69 -0.002982 -0.0008561 8 -40.634 98.3 4.25 0.048  
## 8 10.68 0.001472 -0.003134 -0.0005951 9 -40.535 100.4 6.32 0.017  
## Models ranked by AICc(x)   
## Random terms (all models):   
## '1 | Site'

#### QAIC - QUASI Akaike

When the generalized model shows oversdispersion, the typical AIC value cannot be computed using the dredge function. Instead, the QAIC value needs to be computed. With the function “QAIC” (Quasi AIC) of the package MuMIn, calculated a modification of Akaike’s Information Criterion for overdispersed count data (e.g. quasipoisson model) can be calculated, but it also may also be used to compute QAIC for binomial data (e.g. quasipoisson).

#### QAIC - Quasipoisson

Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

# install.packages("MuMIn")  
library(MuMIn)  
options(na.action = "na.fail")

Fist, we execute the function for “x.quasipossion” constructor, which allows for quasipoisson family objects, for ML estimation.

x.quasipoisson <- function(...) {   
 res <- quasipoisson(...)   
 res$aic <- poisson(...)$aic   
 res   
}

riqueza.glm2<- glm(Species.richness ~ P\_mg\_100g + Land.use + K\_mg\_100g   
 + Hogweed.height, data=Hogweed, family="x.quasipoisson")

To compute the variance inflation factor based on residuals of Pearson, where it is necessary to use the “dredge” function to compute the QAIC values and its parameters. There is a discussion among statisticians regarding whether residuals of Pearson are suitable for models (e.g. Poisson and Binomial) based on Chisq, but there are various examples where it provides a robust and reliable fit.

chat<-sum(residuals(riqueza.glm2,"pearson")^2)/riqueza.glm2$df.residual   
chat

## [1] 3.231642

riqueza.QAIC <- dredge(riqueza.glm2, rank = "QAIC", chat = chat)

## Fixed term is "(Intercept)"

riqueza.QAIC

## Global model call: glm(formula = Species.richness ~ P\_mg\_100g + Land.use + K\_mg\_100g +   
## Hogweed.height, family = "x.quasipoisson", data = Hogweed)  
## ---  
## Model selection table   
## (Int) Hgw.hgh K\_mg\_100 Lnd.use P\_mg\_100 df logLik QAIC delta weight  
## 10 3.412 -0.4344 -0.02074 3 -788.820 496.2 0.00 0.478  
## 12 3.432 -0.4319 -0.002688 -0.01770 4 -787.878 497.6 1.42 0.235  
## 14 3.433 -0.3805 + -0.01978 5 -786.527 498.8 2.58 0.131  
## 16 3.449 -0.3805 -0.002554 + -0.01689 6 -785.687 500.2 4.06 0.063  
## 4 3.433 -0.4422 -0.006557 3 -795.857 500.5 4.35 0.054  
## 8 3.450 -0.3837 -0.006229 + 5 -792.800 502.6 6.46 0.019  
## 2 3.368 -0.4539 2 -803.606 503.3 7.15 0.013  
## 6 3.396 -0.3839 + 4 -799.668 504.9 8.71 0.006  
## 13 3.263 + -0.02047 4 -814.401 514.0 17.83 0.000  
## 15 3.280 -0.002651 + -0.01747 5 -813.488 515.5 19.27 0.000  
## 7 3.277 -0.006323 + 4 -820.933 518.1 21.87 0.000  
## 5 3.222 + 3 -827.998 520.4 24.25 0.000  
## 9 3.022 -0.02408 2 -839.309 525.4 29.25 0.000  
## 11 3.050 -0.003526 -0.02014 3 -837.648 526.4 30.22 0.000  
## 3 3.038 -0.007717 2 -847.930 530.8 34.58 0.000  
## 1 2.950 1 -858.742 535.5 39.27 0.000  
## Models ranked by QAIC(x, chat = 3.23164202774229)

We can compare the QAIC selection by using Pearson’s residuals to fit variance inflation factor, by dividing the deviance by residual degrees of freedom.

chat2=(deviance(riqueza.glm2) / df.residual(riqueza.glm2))  
  
riqueza2.QAIC <- dredge(riqueza.glm2, rank = "QAIC", chat = chat2)

## Fixed term is "(Intercept)"

riqueza2.QAIC

## Global model call: glm(formula = Species.richness ~ P\_mg\_100g + Land.use + K\_mg\_100g +   
## Hogweed.height, family = "x.quasipoisson", data = Hogweed)  
## ---  
## Model selection table   
## (Int) Hgw.hgh K\_mg\_100 Lnd.use P\_mg\_100 df logLik QAIC delta weight  
## 10 3.412 -0.4344 -0.02074 3 -788.820 497.1 0.00 0.478  
## 12 3.432 -0.4319 -0.002688 -0.01770 4 -787.878 498.5 1.42 0.235  
## 14 3.433 -0.3805 + -0.01978 5 -786.527 499.6 2.58 0.132  
## 16 3.449 -0.3805 -0.002554 + -0.01689 6 -785.687 501.1 4.06 0.063  
## 4 3.433 -0.4422 -0.006557 3 -795.857 501.4 4.36 0.054  
## 8 3.450 -0.3837 -0.006229 + 5 -792.800 503.5 6.47 0.019  
## 2 3.368 -0.4539 2 -803.606 504.2 7.17 0.013  
## 6 3.396 -0.3839 + 4 -799.668 505.8 8.73 0.006  
## 13 3.263 + -0.02047 4 -814.401 514.9 17.86 0.000  
## 15 3.280 -0.002651 + -0.01747 5 -813.488 516.3 19.29 0.000  
## 7 3.277 -0.006323 + 4 -820.933 519.0 21.91 0.000  
## 5 3.222 + 3 -827.998 521.3 24.29 0.000  
## 9 3.022 -0.02408 2 -839.309 526.4 29.30 0.000  
## 11 3.050 -0.003526 -0.02014 3 -837.648 527.3 30.27 0.000  
## 3 3.038 -0.007717 2 -847.930 531.7 34.65 0.000  
## 1 2.950 1 -858.742 536.4 39.35 0.000  
## Models ranked by QAIC(x, chat = 3.2258850123267)

We see that the outcomes of QAIC selection using the two approaches above, to compute ‘chat’ as the variance inflation factors, results in virtually the same outcomes.

#### QAIC - Quasibinomial

# install.packages("MuMIn")  
library(MuMIn)  
  
  
Hogweed <- read.table("Hogweed.csv", header=T, sep=",", dec=".")  
head(Hogweed,4)

## Plot Study.area Habitat.type Year Veg.height Veg.cover Species.richness  
## 1 s1 VOL ruderal.grass 2002 0.5 70 18  
## 2 s2 VOL ruderal.grass 2002 0.7 65 19  
## 3 s3 VOL ruderal.grass 2002 0.4 30 16  
## 4 s4 VOL ruderal.grass 2002 0.9 90 23  
## Hogweed.height Hogweed.cover Land.use Disturbance Altitude Inclination  
## 1 0.9 80 NONE TREESHRUB 295.653 2  
## 2 1.4 65 NONE TREESHRUB 295.653 2  
## 3 1.7 90 NONE TREESHRUB 295.653 2  
## 4 1.0 20 NONE NONE 292.112 2  
## Exposition N\_perc P\_mg\_100g K\_mg\_100g  
## 1 W 0.19 3.99 22.27  
## 2 W 0.19 3.56 12.49  
## 3 W 0.19 8.44 39.95  
## 4 W 0.12 2.53 12.34

First, let’s execute a ‘hacked’ constructor for quasibinomial family object for ML estimation

x.quasibinomial <- function(...) {  
 res <- quasibinomial(...)  
 res$aic <- binomial(...)$aic  
 res  
}

coverHog.glm1 <- glm(Hogweed.cover/100 ~ Veg.height + N\_perc +   
 Habitat.type + Land.use + P\_mg\_100g,   
 data=Hogweed,family="x.quasibinomial")  
  
chat3=(deviance(coverHog.glm1) / df.residual(coverHog.glm1))  
  
options(na.action = "na.fail")  
AICselectCover <- dredge(coverHog.glm1)

## Fixed term is "(Intercept)"

AICselectCover

## Global model call: glm(formula = Hogweed.cover/100 ~ Veg.height + N\_perc + Habitat.type +   
## Land.use + P\_mg\_100g, family = "x.quasibinomial", data = Hogweed)  
## ---  
## Model selection table   
## (Int) Hbt.typ Lnd.use N\_prc P\_mg\_100 Veg.hgh df logLik AICc  
## 18 -1.0670 + -1.4490 6 -103.960 220.4  
## 22 -0.9835 + -0.26910 -1.4480 7 -103.885 222.3  
## 26 -1.0640 + -0.0021020 -1.4450 7 -103.923 222.4  
## 30 -0.9841 + -0.26550 -0.0003552 -1.4470 8 -103.880 224.5  
## 20 -1.2730 + + -1.5110 8 -104.183 225.1  
## 28 -1.2680 + + -0.0035190 -1.5050 9 -104.119 227.2  
## 24 -1.2030 + + -0.22390 -1.5090 9 -104.125 227.2  
## 32 -1.2070 + + -0.20140 -0.0021360 -1.5050 10 -104.093 229.3  
## 2 -1.6180 + 5 -111.991 234.3  
## 10 -1.5930 + -0.0112400 6 -111.663 235.8  
## 6 -1.5310 + -0.28200 6 -111.911 236.3  
## 14 -1.5390 + -0.18180 -0.0099980 7 -111.649 237.9  
## 4 -1.7550 + + 7 -112.295 239.2  
## 12 -1.7290 + + -0.0121200 8 -111.946 240.6  
## 8 -1.6720 + + -0.27060 8 -112.220 241.2  
## 16 -1.6840 + + -0.15350 -0.0110100 9 -111.937 242.8  
## 19 -1.3360 + -1.0910 4 -118.099 244.4  
## 23 -1.5340 + 0.66320 -1.1130 5 -117.915 246.1  
## 27 -1.3480 + 0.0093530 -1.1100 5 -118.169 246.6  
## 31 -1.5250 + 0.61040 0.0053720 -1.1220 6 -117.969 248.4  
## 17 -0.2031 -0.5782 2 -123.782 251.6  
## 3 -1.6940 + 3 -122.882 251.9  
## 21 -0.2390 0.12860 -0.5786 3 -123.786 253.7  
## 25 -0.2347 0.0172700 -0.6221 3 -123.797 253.7  
## 7 -1.8480 + 0.50420 4 -122.812 253.8  
## 1 -0.5257 1 -125.948 253.9  
## 11 -1.6990 + 0.0023620 4 -122.920 254.0  
## 29 -0.2312 -0.01305 0.0173400 -0.6222 4 -123.796 255.8  
## 15 -1.8490 + 0.51560 -0.0011110 5 -122.792 255.9  
## 5 -0.5598 0.12110 2 -125.954 256.0  
## 9 -0.5636 0.0116700 2 -126.068 256.2  
## 13 -0.5699 0.02369 0.0115400 3 -126.068 258.3  
## delta weight  
## 18 0.00 0.495  
## 22 2.00 0.182  
## 26 2.07 0.176  
## 30 4.16 0.062  
## 20 4.76 0.046  
## 28 6.83 0.016  
## 24 6.84 0.016  
## 32 8.99 0.006  
## 2 13.94 0.000  
## 10 15.41 0.000  
## 6 15.90 0.000  
## 14 17.53 0.000  
## 4 18.82 0.000  
## 12 20.29 0.000  
## 8 20.84 0.000  
## 16 22.46 0.000  
## 19 24.05 0.000  
## 23 25.79 0.000  
## 27 26.29 0.000  
## 31 28.02 0.000  
## 17 31.27 0.000  
## 3 31.53 0.000  
## 21 33.34 0.000  
## 25 33.36 0.000  
## 7 33.48 0.000  
## 1 33.56 0.000  
## 11 33.69 0.000  
## 29 35.44 0.000  
## 15 35.54 0.000  
## 5 35.62 0.000  
## 9 35.85 0.000  
## 13 37.91 0.000  
## Models ranked by AICc(x)

### MODEL AVERAGING

With the function “model.avg” of the package MuMIn we can compute model averaging based on an information criterion (e.g. AIC (Akaike) or BIC (Baysean)).

BIOVEG <- read.table("BIOVEG3.csv", header=TRUE, sep=",", dec=".")  
head(BIOVEG, 4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

library(MuMIn)  
options(na.action = "na.fail")  
riqueza <- glm(Richness ~ Mortality + Recruitment + Diameter   
 + Temperature\_mean, family=gaussian, data=BIOVEG)  
riqueza.AICc <- dredge(riqueza)

## Fixed term is "(Intercept)"

riqueza.AICc

## Global model call: glm(formula = Richness ~ Mortality + Recruitment + Diameter +   
## Temperature\_mean, family = gaussian, data = BIOVEG)  
## ---  
## Model selection table   
## (Int) Dmt Mrt Rcr Tmp\_men df logLik AICc delta weight  
## 5 15.920 -0.3395 3 -483.219 972.6 0.00 0.140  
## 1 15.060 2 -484.472 973.0 0.42 0.113  
## 6 14.730 0.04540 -0.3990 4 -482.416 973.1 0.50 0.108  
## 13 5.449 -0.4582 0.47360 4 -482.436 973.1 0.55 0.106  
## 7 16.350 -0.1435 -0.3391 4 -482.774 973.8 1.22 0.076  
## 14 5.844 0.03940 -0.4936 0.40910 5 -481.840 974.1 1.49 0.066  
## 3 15.500 -0.1440 3 -484.031 974.2 1.62 0.062  
## 2 14.140 0.03131 3 -484.079 974.3 1.72 0.059  
## 8 15.180 0.04361 -0.1330 -0.3963 5 -482.031 974.5 1.88 0.055  
## 15 6.374 -0.1302 -0.4518 0.44970 5 -482.068 974.6 1.95 0.053  
## 9 11.170 0.17120 3 -484.352 974.9 2.27 0.045  
## 16 6.704 0.03805 -0.1229 -0.4864 0.38880 6 -481.511 975.6 3.01 0.031  
## 4 14.610 0.02957 -0.1369 4 -483.680 975.6 3.03 0.031  
## 11 12.080 -0.1396 0.15010 4 -483.939 976.2 3.55 0.024  
## 10 11.780 0.02887 0.10680 4 -484.035 976.3 3.74 0.021  
## 12 12.630 0.02755 -0.1348 0.08937 5 -483.649 977.7 5.11 0.011  
## Models ranked by AICc(x)

#### Average of the best modes under delta < 4

summary(model.avg(riqueza.AICc, subset = delta < 4))

##   
## Call:  
## model.avg(object = riqueza.AICc, subset = delta < 4)  
##   
## Component model call:   
## glm(formula = Richness ~ <15 unique rhs>, family = gaussian, data   
## = BIOVEG)  
##   
## Component models:   
## df logLik AICc delta weight  
## 3 3 -483.22 972.60 0.00 0.14  
## (Null) 2 -484.47 973.03 0.42 0.11  
## 13 4 -482.42 973.11 0.50 0.11  
## 34 4 -482.44 973.15 0.55 0.11  
## 23 4 -482.77 973.82 1.22 0.08  
## 134 5 -481.84 974.10 1.49 0.07  
## 2 3 -484.03 974.23 1.62 0.06  
## 1 3 -484.08 974.32 1.72 0.06  
## 123 5 -482.03 974.48 1.88 0.06  
## 234 5 -482.07 974.55 1.95 0.05  
## 4 3 -484.35 974.87 2.27 0.05  
## 1234 6 -481.51 975.61 3.01 0.03  
## 12 4 -483.68 975.64 3.03 0.03  
## 24 4 -483.94 976.15 3.55 0.02  
## 14 4 -484.03 976.35 3.74 0.02  
##   
## Term codes:   
## Diameter Mortality Recruitment Temperature\_mean   
## 1 2 3 4   
##   
## Model-averaged coefficients:   
## (full average)   
## Estimate Std. Error Adjusted SE z value Pr(>|z|)   
## (Intercept) 12.51626 6.47186 6.50420 1.924 0.0543 .  
## Recruitment -0.26106 0.26949 0.27049 0.965 0.3345   
## Diameter 0.01462 0.02932 0.02946 0.496 0.6196   
## Temperature\_mean 0.12805 0.29325 0.29466 0.435 0.6639   
## Mortality -0.04573 0.10993 0.11053 0.414 0.6791   
##   
## (conditional average)   
## Estimate Std. Error Adjusted SE z value Pr(>|z|)   
## (Intercept) 12.51626 6.47186 6.50420 1.924 0.0543 .  
## Recruitment -0.40696 0.23202 0.23383 1.740 0.0818 .  
## Diameter 0.03895 0.03664 0.03693 1.055 0.2916   
## Temperature\_mean 0.36591 0.39838 0.40134 0.912 0.3619   
## Mortality -0.13691 0.15394 0.15522 0.882 0.3777   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### POST-HOC TEST TO REGRESSION MODELS

A common necessity when using GLM, LM or GLMM using categorical (factor variable) predictors, is to conduct multiple comparison of means among the pairs of categories. Such comparison using regression models is still being developed and improved. For instance, there are currently no 100% robust, standard unbiased post-hoc tests which can be applied in Mixed Models (e.g. GLMM, LME). Below we provide examples of two of the most commonly used post-hoc methods applied in Ecology but we highly advise you to read about their application and carefully consider whether their use is appropriate for your analysis.

If after you run your model, you find a significant difference of the means among categories (as shown by fitting ANOVA to count data in this GLM model example with species richness as target variable), then you need a post-hoc (posteriori) test to show you between which categories the difference occurred. Post-hoc tests on Linear Models become even more important the more predictor variables you use. If you want to run an a posteriori comparison of the model, knowing that there are significant differences among categories as there are in this example, this comparison will consider the weight of other predictor variables (e.g. k, ca, light).

#### TUKEY’S POST HOC TEST (Tukey’s HSD (honest significant difference) test)

BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG,4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

# Usage  
  
# install.packages("multcomp")  
library(multcomp)  
  
  
richtype <- glm(Richness ~ type, data=BIOVEG, family=poisson)  
summary(richtype)

##   
## Call:  
## glm(formula = Richness ~ type, family = poisson, data = BIOVEG)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.8720 -1.2326 -0.1111 0.8646 5.5989   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.61593 0.03824 68.416 < 2e-16 \*\*\*  
## typeOmbrofila 0.22197 0.05131 4.326 1.52e-05 \*\*\*  
## typeSemidecidual 0.05268 0.05338 0.987 0.324   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 350.25 on 149 degrees of freedom  
## Residual deviance: 329.36 on 147 degrees of freedom  
## AIC: 1007.9  
##   
## Number of Fisher Scoring iterations: 4

summary(glht(richtype, linfct=mcp(type = "Tukey")))

##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Multiple Comparisons of Means: Tukey Contrasts  
##   
##   
## Fit: glm(formula = Richness ~ type, family = poisson, data = BIOVEG)  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)   
## Ombrofila - Aluvial == 0 0.22197 0.05131 4.326 < 0.001 \*\*\*  
## Semidecidual - Aluvial == 0 0.05268 0.05338 0.987 0.58493   
## Semidecidual - Ombrofila == 0 -0.16929 0.05058 -3.347 0.00247 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## (Adjusted p values reported -- single-step method)

cldm0 <- cld(glht(richtype, linfct=mcp(type = "Tukey")))  
cldm0 # here show between each pairs of type there is the significant difference

## Aluvial Ombrofila Semidecidual   
## "a" "b" "a"

#### LMERTEST

This test is used for mixed models (e.g. GLMM, LME) which have Pseudoreplication (when your data is distributed in subplots within each site. E.g. 3 sites with 50 10 m x 10 m plots) and you need to consider this effect (normally called random effect, but be careful when reading about this subject since there remains some disagreement between statisticians). For instance, if you need to not only consider data values per site or fragment, but also data values per subplot within each site. Suppose you need to know the phylogenetic diversity per subplot within each site, you need to recognize that the subplots are not real replicates and so you need to use a mixed model.

Atention: The “lmer test” from the package of the same name is one of the most commonly used post-hoc tests in this case, but there remain disagreements among statisticians about its robustness when homogeneity of variance and normality are violated. Read about this method and carefully consider whether it is appropriate for your analysis. The “lmerTest” package calculate p-values of differences of means among categories.

# install.packages("lmerTest")  
library(lmerTest)   
  
BIOVEG <- read.table("BIOVEG3.csv", header=T, sep=",", dec=".")  
head(BIOVEG,4)

## type Plot Site Species Richness Diameter  
## 1 Semidecidual p001 AREA1 Alchornea\_triplinervia 8 27.12  
## 2 Semidecidual p002 AREA1 Amaioua\_intermedia 10 39.15  
## 3 Semidecidual p003 AREA1 Aniba\_firmula 10 33.93  
## 4 Semidecidual p004 AREA1 Annona\_cacans 10 40.43  
## Mean\_diameter Basal.area Abundance Precipitation\_mean Temperature\_mean  
## 1 6.357 0.058 7 1250 24.4  
## 2 6.524 0.120 12 1250 24.4  
## 3 6.632 0.090 10 1250 24.4  
## 4 6.626 0.128 19 1250 24.4  
## Mortality Recruitment local.x local.y utm.x utm.y local.utm.x  
## 1 6.94 14.34 0 0 747923 7807727 747923  
## 2 2.74 6.94 -10 0 747923 7807727 747913  
## 3 1.71 2.35 -20 0 747923 7807727 747903  
## 4 1.60 2.47 -30 0 747923 7807727 747893  
## local.utm.y  
## 1 7807727  
## 2 7807727  
## 3 7807727  
## 4 7807727

## use to Continuos data  
  
# call the function "glmer" from package "lme4"   
DIAM<-lmer(Diameter ~ type + (1|Site), REML=FALSE, data=BIOVEG)  
  
difflsmeans(DIAM, test.effs="type")

## Least Squares Means table:  
##   
## Estimate Std. Error df t value lower  
## typeAluvial - typeOmbrofila -0.8800 2.7343 150 -0.3218 -6.2828  
## typeAluvial - typeSemidecidual -7.8790 2.7343 150 -2.8815 -13.2818  
## typeOmbrofila - typeSemidecidual -6.9990 2.7343 150 -2.5597 -12.4018  
## upper Pr(>|t|)   
## typeAluvial - typeOmbrofila 4.5228 0.748026   
## typeAluvial - typeSemidecidual -2.4762 0.004539 \*\*  
## typeOmbrofila - typeSemidecidual -1.5962 0.011466 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Confidence level: 95%  
## Degrees of freedom method: Satterthwaite

### MULTIVARIATE ANALYSIS

This approach is indicated to analyze more than one statistical outcome variable at a time. Usually, multivariate analysis to establish relationships among variables, where they were sampled by multiple measurements in sample or experimental units, and their sampling design.

#### NMDS - Non-metric multidimensional scaling

NMDS is used to robustly demonstrate the original position of data in multidimensional space using a reduced number of dimensions that can be easily plotted and visualized.

# use "vegan" package:  
  
# install.packages("vegan")  
library("vegan")

# Adapted NMDS template option:  
  
matrix<- read.table("matrixspp.txt", header=T, sep="\t")  
matrix[1:5,1:4] # shows only part of the data

## abarema\_brachystachya abarema\_villosa acanthocladus\_pulcherrimus  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## alchornea\_glandulosa  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0

# NMDS  
  
NMDS\_bray<- metaMDS(matrix, k=2, distance="bray")

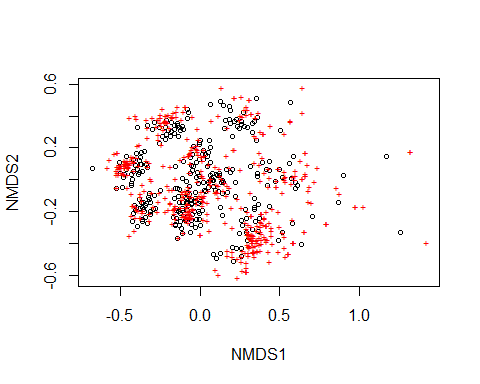
## Wisconsin double standardization  
## Run 0 stress 0.2198643   
## Run 1 stress 0.2216841   
## Run 2 stress 0.2198769   
## ... Procrustes: rmse 0.00288897 max resid 0.0325339   
## Run 3 stress 0.2217402   
## Run 4 stress 0.2215109   
## Run 5 stress 0.2187916   
## ... New best solution  
## ... Procrustes: rmse 0.04319214 max resid 0.1200068   
## Run 6 stress 0.2216311   
## Run 7 stress 0.2194974   
## Run 8 stress 0.2198929   
## Run 9 stress 0.2205943   
## Run 10 stress 0.2183727   
## ... New best solution  
## ... Procrustes: rmse 0.0179176 max resid 0.07948068   
## Run 11 stress 0.2194878   
## Run 12 stress 0.21851   
## ... Procrustes: rmse 0.005008619 max resid 0.06515672   
## Run 13 stress 0.221705   
## Run 14 stress 0.2210081   
## Run 15 stress 0.2198895   
## Run 16 stress 0.2188199   
## ... Procrustes: rmse 0.01624568 max resid 0.07908814   
## Run 17 stress 0.2187936   
## ... Procrustes: rmse 0.01789406 max resid 0.08045457   
## Run 18 stress 0.2188017   
## ... Procrustes: rmse 0.01777192 max resid 0.08036169   
## Run 19 stress 0.2265645   
## Run 20 stress 0.2312948   
## \*\*\* No convergence -- monoMDS stopping criteria:  
## 1: no. of iterations >= maxit  
## 19: stress ratio > sratmax

NMDS\_bray$stress ## < 0.3: check if it was lower like this in your analysis

## [1] 0.2183727

If stress < 0.3 the program returns “TRUE” which indicates acceptable levels of stress within your model. It implies how robust your analysis’ result was.

plot(NMDS\_bray) ### ordinary graph, just distinction in 2 colors and symbols



# Save scores  
  
write.table(NMDS\_bray$points, "sitesR.txt", row.names=T, sep="\t")  
write.table(NMDS\_bray$species, "species.txt", row.names=F, sep="\t")

WARNING: The data sets used here are for training purposes only. The two matrices from test dataset don’t represent a realistic species composition of the vegetation types described here. We collated species # data from our personal datasets and shuffled them into the columns of the species matrix. For example, you mind find that species common in Semideciduous forest (SEMI) might instead be listed amongst Rain Forest (OMB) species.

Better NMDS graph: distinction between vegetation types (as in the following example: ssm, as, fo). You must to use 2 matrix files, one with species and another one with classification in vegetation types (e.g. if you just want to check species distribution between types) or environmental variables (e.g. soil: ph, Al…). In this example we just want to check the species distribution among vegetation types. Thus, you need the species matrix file (here it is the object “matrix”) and another one with category (here it is “matrix4”). Each row number of “matrix4” corresponds to the same row number in “matrix”. Therefore both matrices must have the same sequence and number of rows.

Note 1: again all of the these 3 matrices must have the same sequence and number of rows. Within each row you classify the soil variable value and element, or in case of vegetation the category.

Better Graph (separation among types: OMB, GAL, SEMI):

matrix<- read.table("matrixspp.txt", header=T, sep="\t")  
matrix[1:5,1:4] # shows only part of the data

## abarema\_brachystachya abarema\_villosa acanthocladus\_pulcherrimus  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## alchornea\_glandulosa  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0

matrix5<- read.table("matrixtype.txt", header=T, sep="\t")  
head(matrix5) # shows by default only the first 6 rows of data

## type codigos  
## 1 OMBR OMB  
## 2 OMBR OMB  
## 3 OMBR OMB  
## 4 OMBR OMB  
## 5 OMBR OMB  
## 6 OMBR OMB

forest.nmds <- metaMDS(matrix, distance="bray")

## Wisconsin double standardization  
## Run 0 stress 0.2198643   
## Run 1 stress 0.2192034   
## ... New best solution  
## ... Procrustes: rmse 0.04259014 max resid 0.1203069   
## Run 2 stress 0.2190073   
## ... New best solution  
## ... Procrustes: rmse 0.005225512 max resid 0.05012064   
## Run 3 stress 0.2214929   
## Run 4 stress 0.2188155   
## ... New best solution  
## ... Procrustes: rmse 0.003405314 max resid 0.04596126   
## Run 5 stress 0.2198851   
## Run 6 stress 0.2317977   
## Run 7 stress 0.2196541   
## Run 8 stress 0.2205652   
## Run 9 stress 0.2199014   
## Run 10 stress 0.2194214   
## Run 11 stress 0.219893   
## Run 12 stress 0.2193951   
## Run 13 stress 0.2198767   
## Run 14 stress 0.2185807   
## ... New best solution  
## ... Procrustes: rmse 0.01736558 max resid 0.07833819   
## Run 15 stress 0.2214877   
## Run 16 stress 0.2206233   
## Run 17 stress 0.2282496   
## Run 18 stress 0.2189244   
## ... Procrustes: rmse 0.01710657 max resid 0.07930599   
## Run 19 stress 0.2199584   
## Run 20 stress 0.2196082   
## \*\*\* No convergence -- monoMDS stopping criteria:  
## 1: no. of iterations >= maxit  
## 19: stress ratio > sratmax

grupo<- matrix5$codigos  
grupo.nivel <- levels(grupo)  
head(grupo) # shows by default only the first 6 rows of data

## [1] OMB OMB OMB OMB OMB OMB  
## Levels: GAL OMB SEMI

head(grupo.nivel) # shows by default only the first 6 rows of data

## [1] "GAL" "OMB" "SEMI"

escores.nmds <- scores(forest.nmds)  
head(escores.nmds, 10) # shows only the scores of the first 10 rows

## NMDS1 NMDS2  
## 1 -0.43338491 0.06702318  
## 2 0.16152959 -0.07929433  
## 3 -0.14601269 0.30380524  
## 4 0.12447163 0.45757042  
## 5 -0.05114155 0.13083438  
## 6 0.34695833 -0.14847638  
## 7 -0.32401926 -0.13951549  
## 8 0.41282420 0.05414265  
## 9 -0.47627775 0.10440796  
## 10 0.03482718 -0.07052191

forest.nmds$stress

## [1] 0.2185807

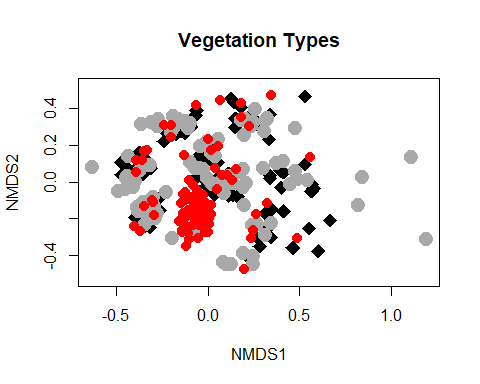
p <- ordiplot(escores.nmds, type="n", main="Vegetation Types")

## species scores not available

grupo.nivel

## [1] "GAL" "OMB" "SEMI"

points(escores.nmds[grupo=="OMB",], pch=(18), cex=2, col="black")  
points(escores.nmds[grupo=="GAL",], pch=(19), cex=2, col="darkgrey")   
points(escores.nmds[grupo=="SEMI",], pch=(20), cex=2, col="red")



#### Analysis of similarities (ANOSIM)

Basic template by “vegan”

The two matrices used here are the same as those used previously in NMDS analysis

# usage:   
  
library(vegan)  
  
# Data  
  
matrix<- read.table("matrixspp.txt", header=T, sep="\t")  
matrix[1:5,1:4] # shows only part of the data

## abarema\_brachystachya abarema\_villosa acanthocladus\_pulcherrimus  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## alchornea\_glandulosa  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0

matrix5<- read.table("matrixtype.txt", header=T, sep="\t")  
head(matrix5) # shows by default only the first 6 rows of data

## type codigos  
## 1 OMBR OMB  
## 2 OMBR OMB  
## 3 OMBR OMB  
## 4 OMBR OMB  
## 5 OMBR OMB  
## 6 OMBR OMB

#### Analysis-ANOSIM

matrix[1:5,1:4] # shows only part of the data

## abarema\_brachystachya abarema\_villosa acanthocladus\_pulcherrimus  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## alchornea\_glandulosa  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0

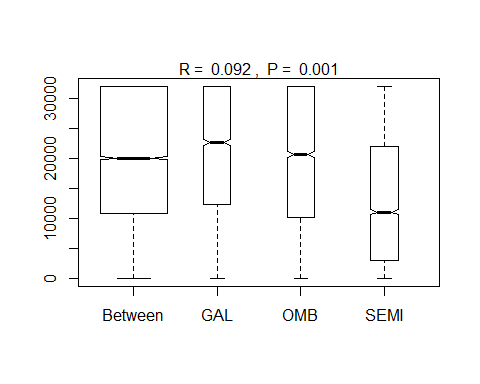
head(matrix5) # shows by default only the first 6 rows of data

## type codigos  
## 1 OMBR OMB  
## 2 OMBR OMB  
## 3 OMBR OMB  
## 4 OMBR OMB  
## 5 OMBR OMB  
## 6 OMBR OMB

attach(matrix5)  
forest.dist <- vegdist(matrix, "bray")  
forest.ano <- anosim(matrix, matrix5$codigo)  
summary(forest.ano)

##   
## Call:  
## anosim(x = matrix, grouping = matrix5$codigo)   
## Dissimilarity: bray   
##   
## ANOSIM statistic R: 0.09155   
## Significance: 0.001   
##   
## Permutation: free  
## Number of permutations: 999  
##   
## Upper quantiles of permutations (null model):  
## 90% 95% 97.5% 99%   
## 0.00741 0.01019 0.01246 0.01541   
##   
## Dissimilarity ranks between and within classes:  
## 0% 25% 50% 75% 100% N  
## Between 3 10782.0 20070 32024.0 32024 26133  
## GAL 4 12251.5 22714 32024.0 32024 4278  
## OMB 2 10176.5 20662 32024.0 32024 4278  
## SEMI 1 3031.5 11048 22024.5 32024 4371

plot(forest.ano)



forest.nmds$stress

## [1] 0.2185807

The graph described in NMDS analysis can also be used to illustrate the result of ANOSIM.

Important: This ANOSIM using R just provides the Global R statistics and p-value, i.e. whether there are differences or not. If you need to know between which pairs there are significant differences, the most pratical and acessible way is to do ANOSIM in the “PAST” software. To do this you insert the names of the categories in the first column (grey), which corresponds to the rows , and the species’ names in the horizontal grey row, which corresponds to the columns. Your columns will contain the species’ names. In the rest of the cells are filled with the species abundance data per plot. The last step is to mark the first column with type names in different colors. Just go in edit and you’ll find how to insert different colors to specific lines. “Past” requires at least 2 different color groups to be defined. Lastly, just click on “Multivariatve” and select “ANOSIM-One way”. The output will return the Global and the pairwise values.

#### CCA (Canonical Correspondence Analysis)

CCA, also refered as constrained correspondence analysis, is one of the most commonly used multivariate analyses to model and to test effects of environmental variables (e.g. soil properties (ph, N, P, K etc) on species composition of plant communities.

# install.packages("vegan")   
library(vegan)  
  
# Usage:  
  
# Species matrix  
  
matrix<- read.table("matrixspp.txt", header=T, sep="\t")  
matrix[1:5,1:4] # shows only part of the data

## abarema\_brachystachya abarema\_villosa acanthocladus\_pulcherrimus  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## alchornea\_glandulosa  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0

# Environmental matrix  
  
env<- read.table("matrixenv.txt", header=T, sep="\t")  
head(env)

## pH P K Ca Mg Al  
## 1 4.5 4 39 0.1 0.1 2.6  
## 2 4.5 4 45 0.1 0.1 3.3  
## 3 4.4 3 39 0.2 0.1 3.2  
## 4 4.2 4 44 0.2 0.1 2.2  
## 5 4.0 5 36 0.2 0.1 4.1  
## 6 4.3 4 28 0.1 0.1 2.1

forest.cca <- cca(matrix ~ pH + P + K + Ca + Mg + Al, env, na.action = na.fail)  
forest.cca

## Call: cca(formula = matrix ~ pH + P + K + Ca + Mg + Al, data =  
## env, na.action = na.fail)  
##   
## Inertia Proportion Rank  
## Total 26.84694 1.00000   
## Constrained 0.57015 0.02124 6  
## Unconstrained 26.27679 0.97876 273  
## Inertia is scaled Chi-square   
## 73 species (variables) deleted due to missingness  
##   
## Eigenvalues for constrained axes:  
## CCA1 CCA2 CCA3 CCA4 CCA5 CCA6   
## 0.13345 0.10707 0.09487 0.08878 0.07677 0.06921   
##   
## Eigenvalues for unconstrained axes:  
## CA1 CA2 CA3 CA4 CA5 CA6 CA7 CA8   
## 0.7949 0.7109 0.6902 0.6704 0.6100 0.5536 0.5123 0.5007   
## (Showing 8 of 273 unconstrained eigenvalues)

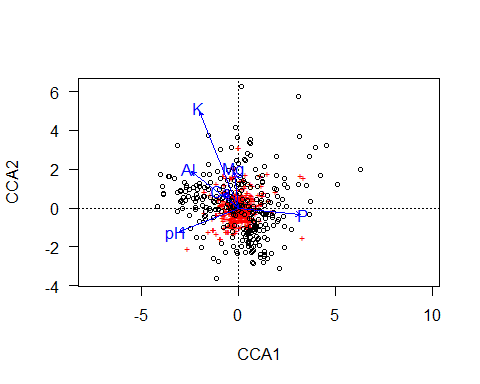
anova(forest.cca, by="margin")

## Permutation test for cca under reduced model  
## Marginal effects of terms  
## Permutation: free  
## Number of permutations: 999  
##   
## Model: cca(formula = matrix ~ pH + P + K + Ca + Mg + Al, data = env, na.action = na.fail)  
## Df ChiSquare F Pr(>F)  
## pH 1 0.1070 1.1121 0.178  
## P 1 0.1046 1.0862 0.262  
## K 1 0.1015 1.0546 0.325  
## Ca 1 0.0974 1.0120 0.432  
## Mg 1 0.0815 0.8467 0.799  
## Al 1 0.1020 1.0597 0.316  
## Residual 273 26.2768

summary<- summary(forest.cca) # scores of CCA  
head(summary) # shows by default only the first 6 rows of the scores

##   
## Call:  
## cca(formula = matrix ~ pH + P + K + Ca + Mg + Al, data = env, na.action = na.fail)   
##   
## Partitioning of scaled Chi-square:  
## Inertia Proportion  
## Total 26.8469 1.00000  
## Constrained 0.5702 0.02124  
## Unconstrained 26.2768 0.97876  
##   
## Eigenvalues, and their contribution to the scaled Chi-square   
##   
## Importance of components:  
## CCA1 CCA2 CCA3 CCA4 CCA5  
## Eigenvalue 0.133454 0.107067 0.094870 0.088783 0.076768  
## Proportion Explained 0.004971 0.003988 0.003534 0.003307 0.002859  
## Cumulative Proportion 0.004971 0.008959 0.012493 0.015800 0.018659  
## CCA6 CA1 CA2 CA3 CA4 CA5  
## Eigenvalue 0.069211 0.79491 0.71093 0.69017 0.67039 0.61000  
## Proportion Explained 0.002578 0.02961 0.02648 0.02571 0.02497 0.02272  
## Cumulative Proportion 0.021237 0.05085 0.07733 0.10303 0.12801 0.15073  
## CA6 CA7 CA8 CA9 CA10 CA11  
## Eigenvalue 0.55365 0.51231 0.50066 0.43055 0.37484 0.35749  
## Proportion Explained 0.02062 0.01908 0.01865 0.01604 0.01396 0.01332  
## Cumulative Proportion 0.17135 0.19043 0.20908 0.22512 0.23908 0.25240  
## CA12 CA13 CA14 CA15 CA16 CA17  
## Eigenvalue 0.33137 0.31731 0.31476 0.29647 0.29313 0.28059  
## Proportion Explained 0.01234 0.01182 0.01172 0.01104 0.01092 0.01045  
## Cumulative Proportion 0.26474 0.27656 0.28828 0.29932 0.31024 0.32069  
## CA18 CA19 CA20 CA21 CA22 CA23  
## Eigenvalue 0.27204 0.264399 0.257427 0.253720 0.247108 0.243790  
## Proportion Explained 0.01013 0.009848 0.009589 0.009451 0.009204 0.009081  
## Cumulative Proportion 0.33083 0.340676 0.350265 0.359715 0.368920 0.378000  
## CA24 CA25 CA26 CA27 CA28 CA29  
## Eigenvalue 0.243745 0.240429 0.229699 0.22041 0.216962 0.215906  
## Proportion Explained 0.009079 0.008956 0.008556 0.00821 0.008081 0.008042  
## Cumulative Proportion 0.387079 0.396035 0.404591 0.41280 0.420882 0.428924  
## CA30 CA31 CA32 CA33 CA34 CA35  
## Eigenvalue 0.212313 0.211601 0.21100 0.20539 0.202015 0.200763  
## Proportion Explained 0.007908 0.007882 0.00786 0.00765 0.007525 0.007478  
## Cumulative Proportion 0.436833 0.444714 0.45257 0.46022 0.467749 0.475227  
## CA36 CA37 CA38 CA39 CA40 CA41  
## Eigenvalue 0.197002 0.193546 0.191625 0.19034 0.183262 0.180381  
## Proportion Explained 0.007338 0.007209 0.007138 0.00709 0.006826 0.006719  
## Cumulative Proportion 0.482565 0.489774 0.496912 0.50400 0.510828 0.517547  
## CA42 CA43 CA44 CA45 CA46  
## Eigenvalue 0.179151 0.178841 0.172691 0.171183 0.168806  
## Proportion Explained 0.006673 0.006661 0.006432 0.006376 0.006288  
## Cumulative Proportion 0.524220 0.530881 0.537314 0.543690 0.549978  
## CA47 CA48 CA49 CA50 CA51  
## Eigenvalue 0.166302 0.164254 0.163920 0.162202 0.159697  
## Proportion Explained 0.006194 0.006118 0.006106 0.006042 0.005948  
## Cumulative Proportion 0.556172 0.562290 0.568396 0.574438 0.580386  
## CA52 CA53 CA54 CA55 CA56  
## Eigenvalue 0.158880 0.156539 0.152281 0.151755 0.150853  
## Proportion Explained 0.005918 0.005831 0.005672 0.005653 0.005619  
## Cumulative Proportion 0.586304 0.592135 0.597807 0.603460 0.609079  
## CA57 CA58 CA59 CA60 CA61  
## Eigenvalue 0.148587 0.145735 0.144786 0.144191 0.142750  
## Proportion Explained 0.005535 0.005428 0.005393 0.005371 0.005317  
## Cumulative Proportion 0.614613 0.620042 0.625435 0.630806 0.636123  
## CA62 CA63 CA64 CA65 CA66  
## Eigenvalue 0.141302 0.140067 0.139000 0.136403 0.135129  
## Proportion Explained 0.005263 0.005217 0.005177 0.005081 0.005033  
## Cumulative Proportion 0.641386 0.646603 0.651781 0.656862 0.661895  
## CA67 CA68 CA69 CA70 CA71  
## Eigenvalue 0.131766 0.130343 0.129344 0.127076 0.126107  
## Proportion Explained 0.004908 0.004855 0.004818 0.004733 0.004697  
## Cumulative Proportion 0.666803 0.671658 0.676476 0.681209 0.685906  
## CA72 CA73 CA74 CA75 CA76 CA77  
## Eigenvalue 0.124247 0.123811 0.122370 0.119772 0.11758 0.11705  
## Proportion Explained 0.004628 0.004612 0.004558 0.004461 0.00438 0.00436  
## Cumulative Proportion 0.690534 0.695146 0.699704 0.704165 0.70854 0.71290  
## CA78 CA79 CA80 CA81 CA82  
## Eigenvalue 0.115508 0.114218 0.112989 0.110408 0.109197  
## Proportion Explained 0.004302 0.004254 0.004209 0.004113 0.004067  
## Cumulative Proportion 0.717207 0.721462 0.725670 0.729783 0.733850  
## CA83 CA84 CA85 CA86 CA87  
## Eigenvalue 0.108409 0.108173 0.105441 0.104595 0.102905  
## Proportion Explained 0.004038 0.004029 0.003927 0.003896 0.003833  
## Cumulative Proportion 0.737888 0.741918 0.745845 0.749741 0.753574  
## CA88 CA89 CA90 CA91 CA92  
## Eigenvalue 0.101358 0.100752 0.100439 0.100086 0.097841  
## Proportion Explained 0.003775 0.003753 0.003741 0.003728 0.003644  
## Cumulative Proportion 0.757349 0.761102 0.764843 0.768571 0.772216  
## CA93 CA94 CA95 CA96 CA97 CA98  
## Eigenvalue 0.096959 0.096093 0.094704 0.09368 0.092814 0.092428  
## Proportion Explained 0.003612 0.003579 0.003528 0.00349 0.003457 0.003443  
## Cumulative Proportion 0.775827 0.779407 0.782934 0.78642 0.789881 0.793324  
## CA99 CA100 CA101 CA102 CA103  
## Eigenvalue 0.091093 0.089654 0.089457 0.087943 0.086575  
## Proportion Explained 0.003393 0.003339 0.003332 0.003276 0.003225  
## Cumulative Proportion 0.796717 0.800056 0.803388 0.806664 0.809889  
## CA104 CA105 CA106 CA107 CA108  
## Eigenvalue 0.085801 0.085266 0.084390 0.083684 0.083050  
## Proportion Explained 0.003196 0.003176 0.003143 0.003117 0.003093  
## Cumulative Proportion 0.813085 0.816261 0.819404 0.822521 0.825615  
## CA109 CA110 CA111 CA112 CA113  
## Eigenvalue 0.082650 0.081223 0.080599 0.079815 0.078284  
## Proportion Explained 0.003079 0.003025 0.003002 0.002973 0.002916  
## Cumulative Proportion 0.828693 0.831719 0.834721 0.837694 0.840610  
## CA114 CA115 CA116 CA117 CA118 CA119  
## Eigenvalue 0.077683 0.077200 0.076070 0.075753 0.07330 0.072731  
## Proportion Explained 0.002894 0.002876 0.002833 0.002822 0.00273 0.002709  
## Cumulative Proportion 0.843503 0.846379 0.849212 0.852034 0.85476 0.857473  
## CA120 CA121 CA122 CA123 CA124 CA125  
## Eigenvalue 0.071921 0.071254 0.069834 0.06928 0.068163 0.067691  
## Proportion Explained 0.002679 0.002654 0.002601 0.00258 0.002539 0.002521  
## Cumulative Proportion 0.860152 0.862806 0.865407 0.86799 0.870527 0.873048  
## CA126 CA127 CA128 CA129 CA130  
## Eigenvalue 0.066292 0.065208 0.064862 0.064022 0.063521  
## Proportion Explained 0.002469 0.002429 0.002416 0.002385 0.002366  
## Cumulative Proportion 0.875517 0.877946 0.880362 0.882747 0.885113  
## CA131 CA132 CA133 CA134 CA135  
## Eigenvalue 0.062631 0.061629 0.060607 0.059574 0.058172  
## Proportion Explained 0.002333 0.002296 0.002258 0.002219 0.002167  
## Cumulative Proportion 0.887446 0.889742 0.891999 0.894218 0.896385  
## CA136 CA137 CA138 CA139 CA140  
## Eigenvalue 0.057939 0.057631 0.056849 0.056001 0.054700  
## Proportion Explained 0.002158 0.002147 0.002118 0.002086 0.002037  
## Cumulative Proportion 0.898543 0.900690 0.902807 0.904893 0.906931  
## CA141 CA142 CA143 CA144 CA145  
## Eigenvalue 0.053797 0.052781 0.052687 0.052129 0.050532  
## Proportion Explained 0.002004 0.001966 0.001962 0.001942 0.001882  
## Cumulative Proportion 0.908934 0.910900 0.912863 0.914805 0.916687  
## CA146 CA147 CA148 CA149 CA150  
## Eigenvalue 0.050143 0.049893 0.048909 0.047714 0.046742  
## Proportion Explained 0.001868 0.001858 0.001822 0.001777 0.001741  
## Cumulative Proportion 0.918555 0.920413 0.922235 0.924012 0.925753  
## CA151 CA152 CA153 CA154 CA155 CA156  
## Eigenvalue 0.046119 0.04590 0.044665 0.044152 0.043621 0.043195  
## Proportion Explained 0.001718 0.00171 0.001664 0.001645 0.001625 0.001609  
## Cumulative Proportion 0.927471 0.92918 0.930844 0.932489 0.934114 0.935722  
## CA157 CA158 CA159 CA160 CA161  
## Eigenvalue 0.042589 0.040889 0.040483 0.039867 0.039142  
## Proportion Explained 0.001586 0.001523 0.001508 0.001485 0.001458  
## Cumulative Proportion 0.937309 0.938832 0.940340 0.941825 0.943283  
## CA162 CA163 CA164 CA165 CA166  
## Eigenvalue 0.038772 0.038209 0.036714 0.036433 0.035424  
## Proportion Explained 0.001444 0.001423 0.001368 0.001357 0.001319  
## Cumulative Proportion 0.944727 0.946150 0.947518 0.948875 0.950194  
## CA167 CA168 CA169 CA170 CA171 CA172  
## Eigenvalue 0.034841 0.034034 0.03357 0.032810 0.032202 0.031659  
## Proportion Explained 0.001298 0.001268 0.00125 0.001222 0.001199 0.001179  
## Cumulative Proportion 0.951492 0.952760 0.95401 0.955232 0.956432 0.957611  
## CA173 CA174 CA175 CA176 CA177 CA178  
## Eigenvalue 0.031358 0.030700 0.030271 0.03008 0.028924 0.028624  
## Proportion Explained 0.001168 0.001144 0.001128 0.00112 0.001077 0.001066  
## Cumulative Proportion 0.958779 0.959923 0.961050 0.96217 0.963248 0.964314  
## CA179 CA180 CA181 CA182 CA183  
## Eigenvalue 0.027961 0.027319 0.02710 0.0262734 0.026229  
## Proportion Explained 0.001041 0.001018 0.00101 0.0009786 0.000977  
## Cumulative Proportion 0.965355 0.966373 0.96738 0.9683612 0.969338  
## CA184 CA185 CA186 CA187 CA188  
## Eigenvalue 0.0255123 0.0247811 0.02443 0.0240143 0.0238019  
## Proportion Explained 0.0009503 0.0009231 0.00091 0.0008945 0.0008866  
## Cumulative Proportion 0.9702885 0.9712116 0.97212 0.9730160 0.9739026  
## CA189 CA190 CA191 CA192 CA193  
## Eigenvalue 0.0231740 0.0222444 0.0218978 0.0213761 0.0208438  
## Proportion Explained 0.0008632 0.0008286 0.0008157 0.0007962 0.0007764  
## Cumulative Proportion 0.9747658 0.9755944 0.9764100 0.9772062 0.9779826  
## CA194 CA195 CA196 CA197 CA198  
## Eigenvalue 0.0204801 0.0203209 0.0195663 0.0193642 0.0192158  
## Proportion Explained 0.0007628 0.0007569 0.0007288 0.0007213 0.0007158  
## Cumulative Proportion 0.9787455 0.9795024 0.9802312 0.9809525 0.9816682  
## CA199 CA200 CA201 CA202 CA203  
## Eigenvalue 0.0182656 0.0180880 0.0170441 0.0167410 0.0161192  
## Proportion Explained 0.0006804 0.0006737 0.0006349 0.0006236 0.0006004  
## Cumulative Proportion 0.9823486 0.9830223 0.9836572 0.9842808 0.9848812  
## CA204 CA205 CA206 CA207 CA208  
## Eigenvalue 0.0160310 0.0155517 0.0149094 0.0142466 0.0141906  
## Proportion Explained 0.0005971 0.0005793 0.0005553 0.0005307 0.0005286  
## Cumulative Proportion 0.9854783 0.9860576 0.9866129 0.9871436 0.9876722  
## CA209 CA210 CA211 CA212 CA213  
## Eigenvalue 0.0137521 0.0135397 0.0131839 0.0125312 0.0123308  
## Proportion Explained 0.0005122 0.0005043 0.0004911 0.0004668 0.0004593  
## Cumulative Proportion 0.9881844 0.9886888 0.9891798 0.9896466 0.9901059  
## CA214 CA215 CA216 CA217 CA218  
## Eigenvalue 0.0119986 0.0118160 0.0112397 0.0110831 0.0109782  
## Proportion Explained 0.0004469 0.0004401 0.0004187 0.0004128 0.0004089  
## Cumulative Proportion 0.9905528 0.9909929 0.9914116 0.9918244 0.9922333  
## CA219 CA220 CA221 CA222 CA223  
## Eigenvalue 0.0105488 0.0100077 0.0096448 0.0092001 0.0089624  
## Proportion Explained 0.0003929 0.0003728 0.0003593 0.0003427 0.0003338  
## Cumulative Proportion 0.9926263 0.9929990 0.9933583 0.9937010 0.9940348  
## CA224 CA225 CA226 CA227 CA228  
## Eigenvalue 0.0089161 0.0086083 0.008053 0.0077037 0.0074697  
## Proportion Explained 0.0003321 0.0003206 0.000300 0.0002869 0.0002782  
## Cumulative Proportion 0.9943669 0.9946876 0.994988 0.9952745 0.9955527  
## CA229 CA230 CA231 CA232 CA233  
## Eigenvalue 0.0072798 0.0071658 0.0066610 0.0062860 0.0058629  
## Proportion Explained 0.0002712 0.0002669 0.0002481 0.0002341 0.0002184  
## Cumulative Proportion 0.9958239 0.9960908 0.9963389 0.9965730 0.9967914  
## CA234 CA235 CA236 CA237 CA238  
## Eigenvalue 0.005692 0.0056442 0.0052689 0.0050255 0.004779  
## Proportion Explained 0.000212 0.0002102 0.0001963 0.0001872 0.000178  
## Cumulative Proportion 0.997003 0.9972137 0.9974099 0.9975971 0.997775  
## CA239 CA240 CA241 CA242 CA243  
## Eigenvalue 0.0046528 0.004536 0.004027 0.0035879 0.0034546  
## Proportion Explained 0.0001733 0.000169 0.000150 0.0001336 0.0001287  
## Cumulative Proportion 0.9979484 0.998117 0.998267 0.9984010 0.9985297  
## CA244 CA245 CA246 CA247 CA248  
## Eigenvalue 0.0032627 0.003034 0.0028833 0.0027852 2.587e-03  
## Proportion Explained 0.0001215 0.000113 0.0001074 0.0001037 9.638e-05  
## Cumulative Proportion 0.9986512 0.998764 0.9988716 0.9989754 9.991e-01  
## CA249 CA250 CA251 CA252 CA253  
## Eigenvalue 2.432e-03 2.309e-03 2.199e-03 2.057e-03 1.994e-03  
## Proportion Explained 9.057e-05 8.602e-05 8.192e-05 7.662e-05 7.426e-05  
## Cumulative Proportion 9.992e-01 9.992e-01 9.993e-01 9.994e-01 9.995e-01  
## CA254 CA255 CA256 CA257 CA258  
## Eigenvalue 0.0017532 1.531e-03 1.333e-03 1.243e-03 0.0010927  
## Proportion Explained 0.0000653 5.702e-05 4.964e-05 4.629e-05 0.0000407  
## Cumulative Proportion 0.9995465 9.996e-01 9.997e-01 9.997e-01 0.9997401  
## CA259 CA260 CA261 CA262 CA263  
## Eigenvalue 1.016e-03 9.462e-04 8.598e-04 8.044e-04 6.902e-04  
## Proportion Explained 3.784e-05 3.524e-05 3.203e-05 2.996e-05 2.571e-05  
## Cumulative Proportion 9.998e-01 9.998e-01 9.998e-01 9.999e-01 9.999e-01  
## CA264 CA265 CA266 CA267 CA268  
## Eigenvalue 6.058e-04 4.842e-04 3.749e-04 2.771e-04 2.547e-04  
## Proportion Explained 2.257e-05 1.804e-05 1.396e-05 1.032e-05 9.485e-06  
## Cumulative Proportion 9.999e-01 9.999e-01 1.000e+00 1.000e+00 1.000e+00  
## CA269 CA270 CA271 CA272 CA273  
## Eigenvalue 2.293e-04 1.765e-04 1.570e-04 7.871e-05 2.199e-05  
## Proportion Explained 8.541e-06 6.574e-06 5.849e-06 2.932e-06 8.190e-07  
## Cumulative Proportion 1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00  
##   
## Accumulated constrained eigenvalues  
## Importance of components:  
## CCA1 CCA2 CCA3 CCA4 CCA5 CCA6  
## Eigenvalue 0.1335 0.1071 0.09487 0.08878 0.07677 0.06921  
## Proportion Explained 0.2341 0.1878 0.16639 0.15572 0.13464 0.12139  
## Cumulative Proportion 0.2341 0.4219 0.58825 0.74397 0.87861 1.00000  
##   
## Scaling 2 for species and site scores  
## \* Species are scaled proportional to eigenvalues  
## \* Sites are unscaled: weighted dispersion equal on all dimensions  
##   
##   
## Species scores  
##   
## CCA1 CCA2 CCA3 CCA4 CCA5  
## abarema\_brachystachya -0.08600 0.74307 -0.05159 0.91193 0.05159  
## abarema\_villosa -0.15080 1.70220 -0.98401 -1.92075 -1.03712  
## acanthocladus\_pulcherrimus 1.21131 0.22509 -0.50765 0.64935 0.01608  
## alchornea\_glandulosa -0.91166 -0.24970 -0.07451 0.07332 -0.26944  
## alibertia\_concolor 0.32728 -0.45372 -0.11631 -0.28745 0.42483  
## alibertia\_edulis 0.01383 0.05318 0.17868 0.25923 0.04599  
## ....   
## CCA6  
## abarema\_brachystachya -0.113733  
## abarema\_villosa -0.675995  
## acanthocladus\_pulcherrimus 1.273453  
## alchornea\_glandulosa -0.001909  
## alibertia\_concolor -0.449355  
## alibertia\_edulis -0.012734  
## ....   
##   
##   
## Site scores (weighted averages of species scores)  
##   
## CCA1 CCA2 CCA3 CCA4 CCA5 CCA6  
## row1 1.48951 -1.6420 -1.8624 -2.1549 -0.3991 -1.0164  
## row2 -0.07990 1.1061 -1.2860 -3.2203 2.5605 0.9580  
## row3 2.02275 -0.7063 0.4961 -1.3433 0.8617 0.3517  
## row4 6.29102 2.0096 -1.4887 -0.5053 3.8836 3.0659  
## row5 0.00129 -0.1397 1.8836 -1.8595 1.4453 3.1196  
## row6 1.38952 -0.8245 -1.4034 1.5555 0.6720 -1.1118  
## ....   
##   
##   
## Site constraints (linear combinations of constraining variables)  
##   
## CCA1 CCA2 CCA3 CCA4 CCA5 CCA6  
## row1 0.5722 -0.12906 -0.64977 -1.18507 0.6791 -0.021241  
## row2 -0.1000 -0.02029 -0.23062 -1.51626 0.7934 0.512563  
## row3 -0.5512 -0.54413 -0.03378 -0.23777 0.2450 0.631624  
## row4 1.5712 0.61372 -0.46802 0.08159 0.5547 -0.227517  
## row5 0.5161 -0.62360 1.33765 -0.91375 0.2887 1.930741  
## row6 1.4922 -0.40025 -0.81149 -0.48078 0.3170 0.009357  
## ....   
##   
##   
## Biplot scores for constraining variables  
##   
## CCA1 CCA2 CCA3 CCA4 CCA5 CCA6  
## pH -0.50007 -0.19882 -0.19379 -0.41096 0.06805 -0.7066  
## P 0.52640 -0.05314 0.65806 -0.49648 0.10292 -0.1730  
## K -0.32880 0.82533 0.37257 -0.07746 0.04001 -0.2536  
## Ca -0.11852 0.11749 0.72741 0.22325 -0.32501 -0.5362  
## Mg -0.03656 0.30274 0.07982 -0.27226 -0.89713 -0.1472  
## Al -0.39741 0.31070 0.40749 -0.22859 0.04560 0.7247

# ordinary plot  
x11(width=12, height=12)  
plot(forest.cca, las=1)



# dev.off()

#### PCA (Principal Component Analysis

PCA is a method commonly used to separate variables (e.g environmental data, such as soil conditions) in ordination before submitting them to further analysis.

To execute the PCA in the example below, we use the function rda, though other functions such as prcomp or princomp can also be used to calculate PCA. It’s important to correctly interpret the PCA output before trying to understand values returned by the function “PCAsignificance” and the command “PC1.exp”, which are explained below. The command PC1.exp shows the percentage of explanation of each axis. In the function “PCAsignificance” the result named “percentage of variance” shows the percentage of variation explained by axis 1. You can access further information about this, and other values, by requesting R help for this function by writing: ?PCAsignificance.

The output of rda, v, contains the scores of the “species” (here: soil variables) and sites on the ordination axes. These are not the same as the correlations of variables with the axes.

# install.packages("vegan")   
  
library(vegan) # rda function for PCA analysis  
  
# Environmental matrix  
env<- read.table("matrixenv.txt", header=T, sep="\t")  
head(env)

## pH P K Ca Mg Al  
## 1 4.5 4 39 0.1 0.1 2.6  
## 2 4.5 4 45 0.1 0.1 3.3  
## 3 4.4 3 39 0.2 0.1 3.2  
## 4 4.2 4 44 0.2 0.1 2.2  
## 5 4.0 5 36 0.2 0.1 4.1  
## 6 4.3 4 28 0.1 0.1 2.1

# Usage:  
  
pca.env <- rda(env, scale=TRUE) # scale = TRUE - to standardize the data  
pca.env

## Call: rda(X = env, scale = TRUE)  
##   
## Inertia Rank  
## Total 6   
## Unconstrained 6 6  
## Inertia is correlations   
##   
## Eigenvalues for unconstrained axes:  
## PC1 PC2 PC3 PC4 PC5 PC6   
## 1.9853 1.4433 1.0506 0.7259 0.5413 0.2536

summary<- summary(pca.env)   
head(summary) # shows by default only the first 6 rows of the scores

##   
## Call:  
## rda(X = env, scale = TRUE)   
##   
## Partitioning of correlations:  
## Inertia Proportion  
## Total 6 1  
## Unconstrained 6 1  
##   
## Eigenvalues, and their contribution to the correlations   
##   
## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6  
## Eigenvalue 1.9853 1.4433 1.0506 0.7259 0.54130 0.25355  
## Proportion Explained 0.3309 0.2406 0.1751 0.1210 0.09022 0.04226  
## Cumulative Proportion 0.3309 0.5714 0.7465 0.8675 0.95774 1.00000  
##   
## Scaling 2 for species and site scores  
## \* Species are scaled proportional to eigenvalues  
## \* Sites are unscaled: weighted dispersion equal on all dimensions  
## \* General scaling constant of scores: 6.396448   
##   
##   
## Species scores  
##   
## PC1 PC2 PC3 PC4 PC5 PC6  
## pH -0.7012 2.0436 -0.3792 0.883221 1.0642 0.30809  
## P -0.9181 -0.4401 2.2376 -0.225144 0.8270 -0.20217  
## K -2.1470 -0.4973 -0.5432 0.972407 -0.2224 -0.81988  
## Ca -2.1582 0.4553 0.6081 -0.001057 -1.0408 0.70784  
## Mg -1.6190 0.1906 -0.9777 -1.712833 0.5125 -0.09719  
## Al -0.5612 -2.2319 -0.6268 0.489924 0.6923 0.64071  
##   
##   
## Site scores (weighted sums of species scores)  
##   
## PC1 PC2 PC3 PC4 PC5 PC6  
## sit1 0.2332 0.089598 0.17463 0.08095 0.4550 -0.2855  
## sit2 0.1368 -0.145176 0.06386 0.24593 0.6126 -0.1015  
## sit3 0.1862 -0.107166 -0.08787 0.13652 0.1706 0.2740  
## sit4 0.1762 0.005016 0.30162 -0.10225 -0.1222 -0.5942  
## sit5 0.1254 -0.741146 0.40371 -0.10187 0.3741 0.3728  
## sit6 0.4021 0.131917 0.32129 -0.24087 0.1643 -0.3517  
## ....

If you want to test for the signifance of PCA, you can either use the function PCAsignificance of the package BiodiversityR (e.g. PCAsignificance(pca.env, axes=6)) or to compute an anova under a rda.

(av <- pca.env$CA$eig) # av - inspects the eigenvalues. The sum of eigenvalues

## PC1 PC2 PC3 PC4 PC5 PC6   
## 1.9852662 1.4433273 1.0506311 0.7259281 0.5412958 0.2535515

# represent the total percentage of explanation assigned   
 # to each axis.  
 # pca.env$CA$":Relationship of each environmental variable  
 # with the axes.The closer the result is to 1, the larger   
 # the association with the positive or negative axis, respectively.

sum.av <- sum(av)  
PC1.exp <- 100\*(av[1]/sum(av)) # calculate the percentage of explanation of   
PC1.exp # each axis. Here it shows the percentage of the

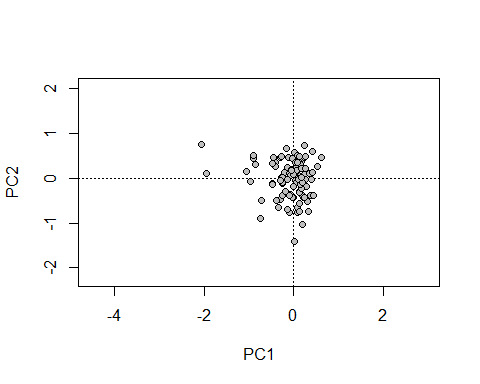
## PC1   
## 33.08777

round(PC1.exp, 2) # You can choose how many decimals after the point you would

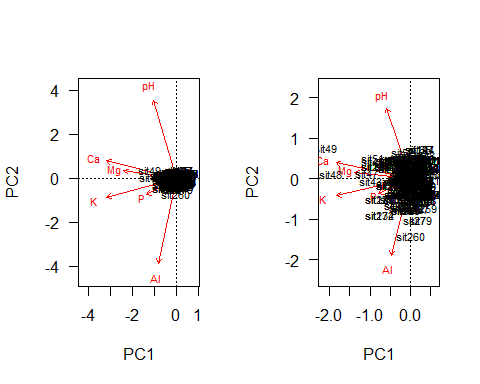
## PC1   
## 33.09

# like to keep. Here it is 2.

# Simple gragh  
  
x11(width=12, height=12)  
plot(pca.env, type="n")  
points(pca.env, display = "sites", col=1, pch=21, cex=1, bg="grey")

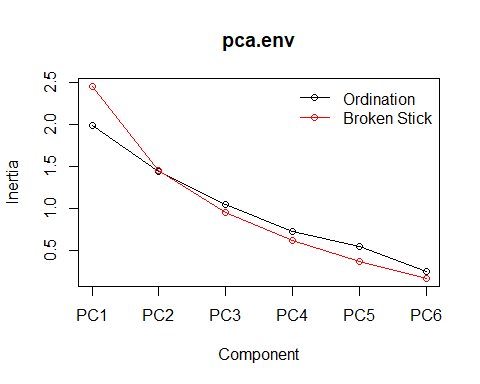


# Biplot graph  
  
par(mfrow=c(1,2))  
plot1 <- biplot(pca.env, scaling=1, type="text", las=1) # sites scaled by   
 # eigenvalues  
plot2 <- biplot(pca.env, scaling=2, type="text", las=1) # species scaled by



# eigenvalues  
# dev.off()

# Screeplots - select axes with values from the ordination that are larger   
# than the expected value under the Broken stick criteria.  
  
screeplot(pca.env, bstick = TRUE, type = "lines")



#### Analysis of Indicator Species (INDVAL)

The association index named IndVal computed with the function “multipatt” of the package “indicspecies” is generated by combinations of the input clusters and comparison of each combination with the matrix species. Thus, each species receives the combination with the highest association value and the best matching patterns are tested for statistical significance of the associations. Thus this analysis is useful to indicate species that can be used to gather information about specific characteristics of habitats or environment (e.g. forest types; soil types; drought or wetter sites).

The name “type” within the command below refers to the categorical variable in the forest type matrix (matrixtype.txt). “codigos” are labels given to abbreviate forest type names: OMB = Rain (Ombrophilous) Forest; GAL = Gallery Forest and SEMI = Semideciduos forest.

# install.packages("indicspecies")  
  
library(indicspecies)  
  
# Species matrix  
matrixspp<- read.table("matrixspp.txt", header=T, sep="\t")  
matrixspp[1:5,1:4] # shows only part of the data

## abarema\_brachystachya abarema\_villosa acanthocladus\_pulcherrimus  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## alchornea\_glandulosa  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0

# Vegetation type  
type <- read.table("matrixtype.txt", header=T)  
head(type)

## type codigos  
## 1 OMBR OMB  
## 2 OMBR OMB  
## 3 OMBR OMB  
## 4 OMBR OMB  
## 5 OMBR OMB  
## 6 OMBR OMB

wetpt = multipatt(matrixspp, type$codigos) # Runs the combination analysis   
 # using IndVal.g as statistic  
head(wetpt, 3) # shows by only the first 3 rows of the scores

## $call  
## multipatt(x = matrixspp, cluster = type$codigos)  
##   
## $func  
## [1] "IndVal.g"  
##   
## $cluster  
## [1] OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB   
## [15] OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB   
## [29] OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB   
## [43] OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB   
## [57] OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB   
## [71] OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB OMB   
## [85] OMB OMB OMB OMB OMB OMB OMB OMB OMB GAL GAL GAL GAL GAL   
## [99] GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL   
## [113] GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL   
## [127] GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL   
## [141] GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL   
## [155] GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL   
## [169] GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL GAL   
## [183] GAL GAL GAL GAL SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI  
## [197] SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI  
## [211] SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI  
## [225] SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI  
## [239] SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI  
## [253] SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI  
## [267] SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI SEMI  
## Levels: GAL OMB SEMI

summary(wetpt) # Lists those species with significant association to one combination

##   
## Multilevel pattern analysis  
## ---------------------------  
##   
## Association function: IndVal.g  
## Significance level (alpha): 0.05  
##   
## Total number of species: 532  
## Selected number of species: 62   
## Number of species associated to 1 group: 49   
## Number of species associated to 2 groups: 13   
##   
## List of species associated to each combination:   
##   
## Group GAL #sps. 12   
## stat p.value   
## callisthene\_major 0.360 0.005 \*\*  
## allagoptera\_caudescens 0.302 0.005 \*\*  
## guazuma\_ulmifolia 0.292 0.005 \*\*  
## alibertia\_edulis 0.280 0.010 \*\*  
## miconia\_cabucu 0.260 0.045 \*   
## dilodendron\_bipinnatum 0.245 0.010 \*\*  
## protium\_warmingiana 0.239 0.015 \*   
## diospyros\_brasiliensis 0.239 0.005 \*\*  
## miconia\_latecrenata 0.238 0.010 \*\*  
## parinari\_parvifolia 0.232 0.005 \*\*  
## callisthene\_minor 0.224 0.025 \*   
## psidium\_oblongatum 0.207 0.015 \*   
##   
## Group OMB #sps. 8   
## stat p.value   
## licania\_kunthiana 0.417 0.005 \*\*  
## phyllostemonodaphne\_geminiflora 0.260 0.005 \*\*  
## styrax\_ferrugineus 0.254 0.010 \*\*  
## ixora\_gardneriana 0.232 0.025 \*   
## deguelia\_hatschbachii 0.224 0.030 \*   
## virola\_officinalis 0.223 0.045 \*   
## myrciaria\_floribunda 0.217 0.015 \*   
## licania\_spicata 0.180 0.050 \*   
##   
## Group SEMI #sps. 29   
## stat p.value   
## casearia\_arborea 0.716 0.005 \*\*  
## lacistema\_pubescens 0.632 0.005 \*\*  
## matayba\_elaeagnoides 0.630 0.005 \*\*  
## xylopia\_sericea 0.484 0.005 \*\*  
## erythroxylum\_pelleterianum 0.469 0.005 \*\*  
## xylopia\_aromatica 0.467 0.005 \*\*  
## miconia\_calvescens 0.461 0.005 \*\*  
## myrcia\_guianensis 0.461 0.005 \*\*  
## miconia\_sellowiana 0.442 0.005 \*\*  
## pera\_heteranthera 0.427 0.005 \*\*  
## himatanthus\_phagedaenicus 0.425 0.005 \*\*  
## sacoglottis\_mattogrossensis 0.396 0.005 \*\*  
## stryphnodendron\_guianense 0.395 0.005 \*\*  
## siparuna\_reginae 0.387 0.005 \*\*  
## jacaranda\_macrantha 0.384 0.005 \*\*  
## gomidesia\_tijucensis 0.372 0.005 \*\*  
## dalbergia\_nigra 0.365 0.005 \*\*  
## guarea\_pendula 0.358 0.005 \*\*  
## cryptocarya\_aschersoniana 0.326 0.005 \*\*  
## vismia\_martiana 0.315 0.005 \*\*  
## guatteria\_gomeziana 0.309 0.010 \*\*  
## ocotea\_diospyrifolia 0.273 0.010 \*\*  
## handroanthus\_riodocensis 0.253 0.010 \*\*  
## duguetia\_lanceolata 0.234 0.005 \*\*  
## hirtella\_racemosa 0.231 0.030 \*   
## swartzia\_polyphylla 0.231 0.010 \*\*  
## sloanea\_guianensis 0.217 0.035 \*   
## alchornea\_glandulosa 0.210 0.050 \*   
## vitex\_megapotamica 0.206 0.030 \*   
##   
## Group GAL+OMB #sps. 10   
## stat p.value   
## myrcia\_sp 0.502 0.005 \*\*  
## aspidosperma\_discolor 0.415 0.030 \*   
## copaifera\_langsdorffii 0.409 0.005 \*\*  
## senefeldera\_verticillata 0.376 0.040 \*   
## protium\_heptaphyllum 0.323 0.005 \*\*  
## clarisia\_ilicifolia 0.293 0.005 \*\*  
## eschweilera\_ovata 0.292 0.030 \*   
## terminalia\_glabrescens 0.264 0.015 \*   
## ocotea\_divaricata 0.220 0.050 \*   
## tabebuia\_serratifolia 0.220 0.030 \*   
##   
## Group GAL+SEMI #sps. 1   
## stat p.value   
## miconia\_albicans 0.243 0.04 \*  
##   
## Group OMB+SEMI #sps. 2   
## stat p.value   
## aniba\_firmula 0.472 0.010 \*\*  
## ocotea\_dispersa 0.317 0.025 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### Permutational Analysis of variance (PERMANOVA)

PERMANOVA is a non-parametric multivariate statistical test used to compare groups of objects based on any measure of distance. Commonly this analysis is used to compare taxon in samples where the groups of samples are compared.

library(vegan)  
library(permute)  
library(lattice)  
  
spp <- read.table("matrixspp.txt", header=T)  
spp[1:5,10:13]

## allophylus\_racemosus alseis\_floribunda amaioua\_guianensis  
## 1 0 0 1  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## amaioua\_intermedia  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0

log(spp+1)->spp2  
spp2[1:5,10:13]

## allophylus\_racemosus alseis\_floribunda amaioua\_guianensis  
## 1 0 0 0.6931472  
## 2 0 0 0.0000000  
## 3 0 0 0.0000000  
## 4 0 0 0.0000000  
## 5 0 0 0.0000000  
## amaioua\_intermedia  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0

Usage:

Testing for species composition over different forest types: the name “type” within the command below is the categorical variable in the forest type matrix (matrixtype.txt). “codigos” are just labels given to abbreviate forest typenames: OMB = Rain (Ombrophilous) Forest; GAL = Gallery Forest and SEMI = Semideciduos forest.

type<- read.table("matrixtype.txt", header=T, sep="\t")  
head(type)

## type codigos  
## 1 OMBR OMB  
## 2 OMBR OMB  
## 3 OMBR OMB  
## 4 OMBR OMB  
## 5 OMBR OMB  
## 6 OMBR OMB

adonis(spp2 ~ codigos, data=type, permutations=999) -> perma.result1  
perma.result1

##   
## Call:  
## adonis(formula = spp2 ~ codigos, data = type, permutations = 999)   
##   
## Permutation: free  
## Number of permutations: 999  
##   
## Terms added sequentially (first to last)  
##   
## Df SumsOfSqs MeanSqs F.Model R2 Pr(>F)   
## codigos 2 5.283 2.64162 6.5743 0.04532 0.001 \*\*\*  
## Residuals 277 111.302 0.40181 0.95468   
## Total 279 116.585 1.00000   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Testing for soil influence in the tree species abundance: the names in the right side of the formula regard the soil variables contained in the environmental matrix (“matrixenv.txt”). The category “types” regard the forest type (OMB, GAL and SEMI).

env <- read.table("matrixenv.txt", header=T)  
head(env)

## pH P K Ca Mg Al  
## 1 4.5 4 39 0.1 0.1 2.6  
## 2 4.5 4 45 0.1 0.1 3.3  
## 3 4.4 3 39 0.2 0.1 3.2  
## 4 4.2 4 44 0.2 0.1 2.2  
## 5 4.0 5 36 0.2 0.1 4.1  
## 6 4.3 4 28 0.1 0.1 2.1

adonis(spp2 ~ pH + P + K + Ca + Mg + Al, data=env, permutations=999)-> perm.result  
perm.result # Result to permanova pseudo-F, R2 e p.

##   
## Call:  
## adonis(formula = spp2 ~ pH + P + K + Ca + Mg + Al, data = env, permutations = 999)   
##   
## Permutation: free  
## Number of permutations: 999  
##   
## Terms added sequentially (first to last)  
##   
## Df SumsOfSqs MeanSqs F.Model R2 Pr(>F)   
## pH 1 0.345 0.34542 0.82775 0.00296 0.703   
## P 1 0.540 0.54033 1.29483 0.00463 0.152   
## K 1 0.506 0.50551 1.21140 0.00434 0.193   
## Ca 1 0.377 0.37685 0.90306 0.00323 0.587   
## Mg 1 0.294 0.29425 0.70513 0.00252 0.874   
## Al 1 0.600 0.60038 1.43874 0.00515 0.076 .  
## Residuals 273 113.923 0.41730 0.97716   
## Total 279 116.585 1.00000   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### RECOMMENDED READING

#### Highlight:

GOTELLI, N.J.; ELLISON, A.M. A Primer of Ecological Statistics. Sinauer Associates Publishers, 2004. Version in Portuguese: GOTELLI, N.J.; ELLISON, A.M. Princípios de Estatística em Ecologia. Porto Alegre: Artmed, 2011. 527 p.

CRAWLEY, M.J. The R Book: Second Edtion. Chichester: Imperial College London at Silwood Park, John Wiley & Sons Ltd., 2012. 1051p.

BORCARD, D.; GILLET, F.; LEGENDRE, P. Numerical ecology with R. New York: Springer, 2011. 306 p.

FORTIN, M. J.; DALE, M. Spatial analysis: a guide for ecologists. Cambridge: Cambridge University, 2005. 365 p.

HINKLE, D.E.; WIERSMA, W. JURS, S.G. Applied Statistics for the Behavioral Sciences. 5th ed. Boston: Houghton Mifflin, 2003. 756 p.

LEGENDRE, P.; LEGENDRE, L. Numerical ecology. Third English edition. Amsterdam: Elsevier Science, 2012. 1006 p.

#### Complementary:

BEGON’S text book is highly recommendable reading on sampling design and statistical analysis in ecological studies (e.g. distribution, population growth, behaviour) of plants and animals.

BEGON, M.; TOWNSEND, C.R.; HARPER, J.L. Ecology: From Individuals to Ecosystems, 4th Edition. Wiley-Blackwell, 2005. 750 pages. Version in Portuguese: BEGON, M.; TOWNSEND, C.R.; HARPER, J.L. Ecologia: de indivíduos a ecossistemas. 4 ed. Porto Alegre: Artmed, 2007. 752 p.