

R_Activity_11

Dahee Ahn

2024-11-21

```
rm(list=ls())  
library(permute)  
library(vegan)
```

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

```
## This is vegan 2.6-8
```

```
library(reshape2)  
library(lattice)  
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
## filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
## intersect, setdiff, setequal, union
```

1. We will use another open source data set from the NSF Harvard Forest Long-term Ecological Research (LTER) site. These data are spiders collected in the Hemlock Removal Experiment. Remember, this experiment includes four treatments (Hemlock girdled, Hemlock logged, Hemlock control, and Hardwood control) each replicated across two ($n = 2$) 90 x 90 m plots. Load the `HarvardForest_spiders` data into R. We will characterize spider communities

among these four treatments. Load the HarvardForest_HerbLayer data into R. We will also assess the relationship among spiders and under story plants.

```
spiders <- read.csv(file="C:/Users/chemk/Desktop/Classes/ENT6707_DataAnalysis/week14/Data/HarvardForest_spiders.csv", header=T, na.strings=c("", ".", "NA"))
herb <- read.csv(file="C:/Users/chemk/Desktop/Classes/ENT6707_DataAnalysis/week14/Data/HarvardForest_HerbLayer.csv", header=T, na.strings=c("", ".", "NA"))
```

```
View(spiders)
str(spiders)
```

```
## 'data.frame':    1852 obs. of  13 variables:
## $ start.date      : chr  "7/18/2008" "5/19/2008" "9/22/2008" "7/18/2008" ...
## $ end.date        : chr  "7/28/2008" "5/29/2008" "10/2/2008" "7/28/2008" ...
## $ block           : chr  "Valley" "Valley" "Valley" "Valley" ...
## $ treatment       : chr  "Girdled" "Girdled" "Girdled" "Girdled" ...
## $ plot            : int   1 1 1 1 1 1 1 1 1 ...
## $ replicate       : int   5 3 1 5 5 5 1 5 1 ...
## $ sampling.method : chr  "Litter_sifting" "Litter_sifting" "Litter_sifting" "Litter_sifting"
## ...
## $ family           : chr  "LINYPHIIDAE" "SALTICIDAE" "LINYPHIIDAE" "AMAUROBIIDAE" ...
## $ genus            : chr  "Ceraticelus" "immSalt" "immLiny" "immAmau" ...
## $ species          : chr  "laetabilis" "sp." "sp." "sp." ...
## $ males            : int   0 0 0 0 0 1 0 1 0 ...
## $ females          : int   2 0 0 0 0 0 0 1 0 ...
## $ immature         : int   0 2 16 4 1 0 2 0 1 ...
```

```
View(herb)
str(herb)
```

```
## 'data.frame':    5825 obs. of  8 variables:
## $ X               : int   1 2 3 4 5 6 7 8 9 10 ...
## $ plot            : int   1 1 1 1 1 1 1 1 1 1 ...
## $ subplot         : int   1 2 3 4 5 6 6 6 7 7 ...
## $ trt             : chr  "girdled" "girdled" "girdled" "girdled" ...
## $ block           : chr  "valley" "valley" "valley" "valley" ...
## $ year            : int  2003 2003 2003 2003 2003 2003 2003 2003 2003 2003 ...
## $ species         : chr  "ztot.cover" "ztot.cover" "ztot.cover" "ztot.cover" ...
## $ cover           : num   0 0 0 0 0 0.5 5 5.5 2 15 ...
```

2. Before running any analyses, you will have to do some data wrangling. First, create a new variable abundance by summing the counts of adult male and female spiders. Next, change the data set from long format to wide format using spider genus as the taxonomic resolution (i.e., each column should be a spider genus). Be sure that the new data frame includes the predictor variables block, plot, replicate, treatment, and sampling method. Then, remove any columns that do not have count data (some genera are indicated as immatures with imm), as well as three columns with unidentified spiders (Linytold, LinyTold, and unk_told). Change the variables treatment and sampling method to factors. Calculate a dissimilarity matrix for the spider data using the bray-curtis method. Provide the distance matrix output.

```
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':  
##  
##    date, intersect, setdiff, union
```

```
spiders$abundance <- spiders$males + spiders$females # create a new variable  
spiders$start.date <- as.Date(spiders$start.date, format = "%m/%d/%Y")  
spiders$year <- year(spiders$start.date)  
View(spiders)
```

```
library(reshape2) # change from long format to wide format  
spider.matrix <- dcast(spiders, year + block + plot + treatment + sampling.method ~ genus, sum,  
value.var = "abundance", na.rm =TRUE)  
View(spider.matrix)
```

```
spider.matrix$year<-as.factor(spider.matrix$year)
spider.matrix$block<-as.factor(spider.matrix$block)
spider.matrix$plot<-as.factor(spider.matrix$plot)
spider.matrix$treatment<-as.factor(spider.matrix$treatment)
spider.matrix$sampling.method<-as.factor(spider.matrix$sampling.method)
```

```
library(dplyr)
spider_cleaned <- spider.matrix[colSums(spider.matrix[, 6:74] != 0) > 0, ]
spider_cleaned_1 <- spider_cleaned %>% select(-contains("imm"), -c(LinytoID, LinyToID, unk_toI
D))
spider_cleaned_2 <- spider_cleaned_1[complete.cases(spider_cleaned_1), ]
View(spider_cleaned_2)
```

```
spider_cleaned_2$treatment <- as.factor(spider_cleaned_2$treatment)
spider_cleaned_2$sampling.method <- as.factor(spider_cleaned_2$sampling.method)
View(spider_cleaned_2)
```

```
spider1 <- spider_cleaned_2[, 6:56]
spider1$treatment <- spider_cleaned_2$treatment
View(spider1)
```

```
library(vegan)
dis.matrix.pa <- vegdist(spider1[, 1:51], method="bray")
dis.matrix.pa
```

##	1	2	3	4	5	6	7
## 2	0.8000000						
## 3	0.5142119	0.8005115					
## 4	0.8235294	0.6153846	0.7479452				
## 5	0.6306818	0.8314607	0.2982456	0.6666667			
## 6	0.8860759	0.7407407	0.8369906	0.4705882	0.7676056		
## 7	0.4700855	0.7690141	0.4023438	0.5683891	0.4716981	0.7597173	
## 8	0.9033457	0.7216117	0.8093023	0.4331984	0.7468354	0.6616915	0.6395939
## 9	0.4326241	0.8531469	0.3182844	0.7923077	0.4264706	0.8971963	0.5085995
## 10	0.7812500	0.5714286	0.7337110	0.3294118	0.6792453	0.4032258	0.6845426
## 11	0.5314286	0.7877095	0.7142857	0.8169935	0.7342193	0.9439252	0.7400000
## 12	0.7741935	0.2698413	0.7897311	0.4955752	0.7754011	0.7000000	0.7104558
## 13	0.6163522	0.8447205	0.2818372	0.7837838	0.2972973	0.8720000	0.5304740
## 14	0.8232044	0.6216216	0.7485380	0.3962264	0.7003257	0.2566372	0.6928105
## 15	0.5943396	0.7943925	0.4256410	0.7213930	0.4945455	0.8483146	0.4316940
## 16	0.8300000	0.6568627	0.7673130	0.3483146	0.6748466	0.4090909	0.6307692
##	8	9	10	11	12	13	14
## 2							
## 3							
## 4							
## 5							
## 6							
## 7							
## 8							
## 9	0.9076923						
## 10	0.6085106	0.7741935					
## 11	0.8990826	0.5844156	0.7588652				
## 12	0.6769759	0.8026316	0.4672897	0.7360406			
## 13	0.8725762	0.2727273	0.7394366	0.6853933	0.7941176		
## 14	0.6071429	0.7974684	0.2244898	0.7692308	0.5665025	0.7435897	
## 15	0.7516060	0.6666667	0.7692308	0.8391421	0.8116592	0.5465116	0.7678100
## 16	0.4979424	0.8593750	0.3975904	0.8255034	0.6036036	0.7602740	0.3419355
##	15						
## 2							
## 3							
## 4							
## 5							
## 6							
## 7							
## 8							
## 9							
## 10							
## 11							
## 12							
## 13							
## 14							
## 15							
## 16	0.6582915						

3. Run a nonmetric multidimensional scaling (NMDS) model using the metaMDS() function on the spider genera data. Provide a figure showing the treatment groups with 95% confidence interval ellipses. Report the model stress.

Stress: 0.08807097

```
nmds.spider.pa <- metaMDS(dis.matrix.pa, trymax=500, autotransform = TRUE, k=2)
```

```
## Run 0 stress 0.08807097
## Run 1 stress 0.1016486
## Run 2 stress 0.1532871
## Run 3 stress 0.1016486
## Run 4 stress 0.1366764
## Run 5 stress 0.1016486
## Run 6 stress 0.1490606
## Run 7 stress 0.1479332
## Run 8 stress 0.08807097
## ... New best solution
## ... Procrustes: rmse 1.596747e-06 max resid 4.592532e-06
## ... Similar to previous best
## Run 9 stress 0.08807097
## ... Procrustes: rmse 1.769705e-06 max resid 4.898762e-06
## ... Similar to previous best
## Run 10 stress 0.08807097
## ... Procrustes: rmse 6.339017e-07 max resid 1.504007e-06
## ... Similar to previous best
## Run 11 stress 0.1016486
## Run 12 stress 0.1485113
## Run 13 stress 0.08807097
## ... Procrustes: rmse 2.100096e-06 max resid 5.9269e-06
## ... Similar to previous best
## Run 14 stress 0.1016486
## Run 15 stress 0.1022463
## Run 16 stress 0.1016486
## Run 17 stress 0.08807097
## ... Procrustes: rmse 3.370256e-06 max resid 8.385953e-06
## ... Similar to previous best
## Run 18 stress 0.137284
## Run 19 stress 0.08807097
## ... Procrustes: rmse 9.662875e-07 max resid 2.343791e-06
## ... Similar to previous best
## Run 20 stress 0.1016486
## *** Best solution repeated 6 times
```

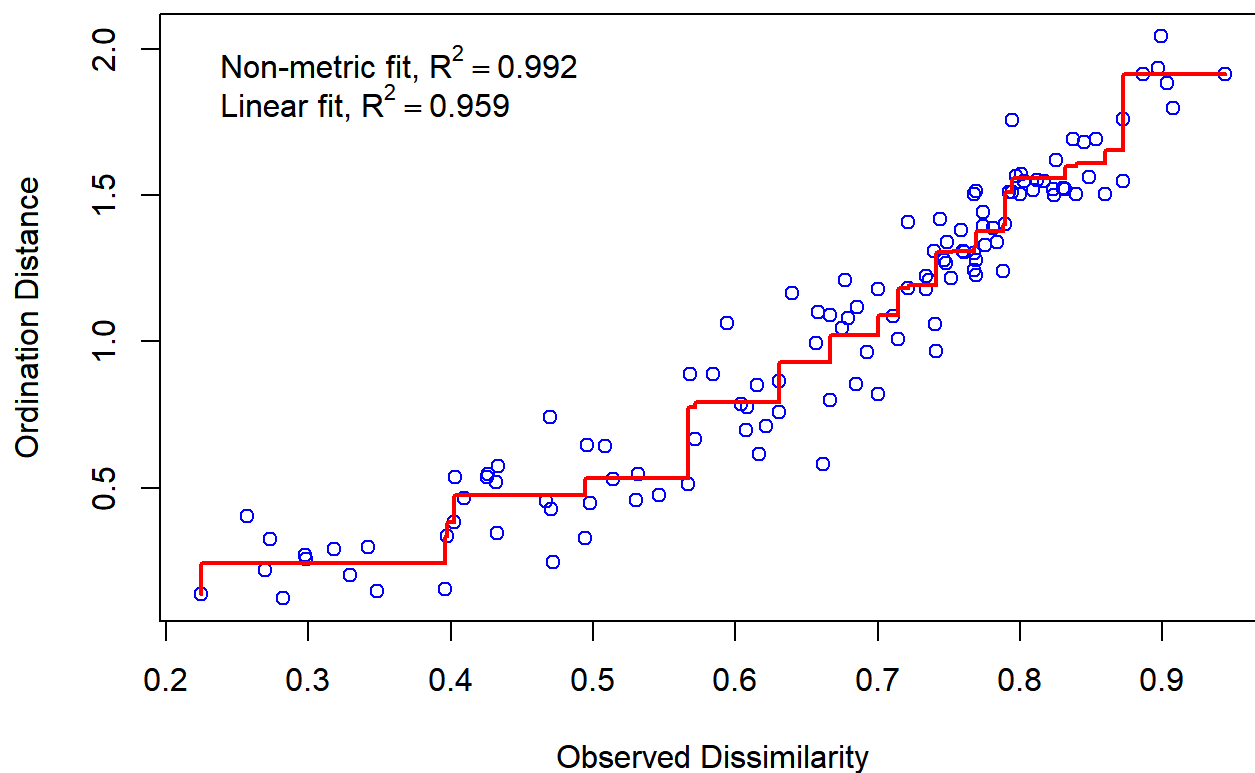
```
nmds.spider.pa
```

```
##
## Call:
## metaMDS(comm = dis.matrix.pa, k = 2, trymax = 500, autotransform = TRUE)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      dis.matrix.pa
## Distance: bray
##
## Dimensions: 2
## Stress:    0.08807097
## Stress type 1, weak ties
## Best solution was repeated 6 times in 20 tries
## The best solution was from try 8 (random start)
## Scaling: centring, PC rotation, halfchange scaling
## Species: scores missing
```

```
any(is.na(dis.matrix.pa))
```

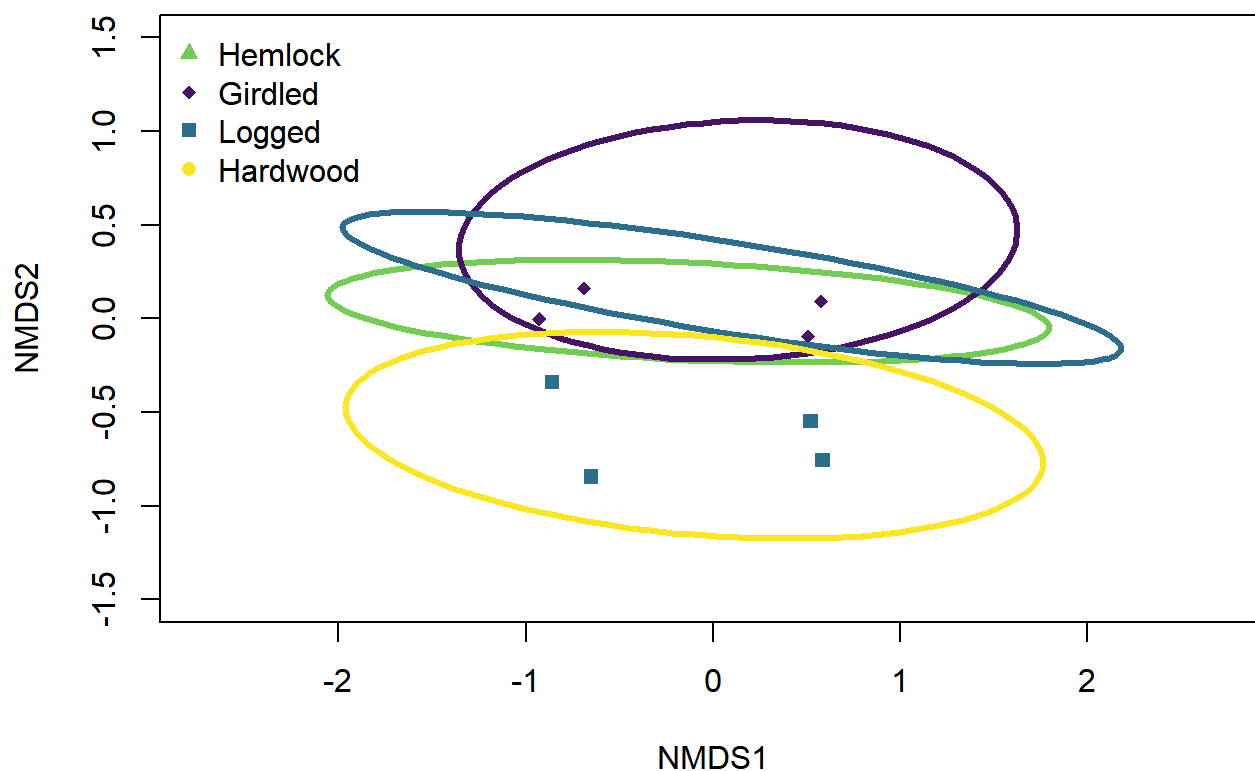
```
## [1] FALSE
```

```
stressplot(nmds.spider.pa)
```



```
library(vegan)
```

```
ordiplot(nmds.spider.pa, display = "sites", type="n", xlim = c(-1.5, 1.5), ylim=c(-1.5, 1.5))
points(nmds.spider.pa, display = "sites",select= which(spider_cleaned_2$treatment=="HemlockContr
ol"),pch=17,cex=1, col="#73D055FF")
points(nmds.spider.pa, display = "sites",select= which(spider_cleaned_2$treatment=="Girdled"),pc
h = 18,cex=1,col="#481567FF")
points(nmds.spider.pa, display = "sites",select= which(spider_cleaned_2$treatment=="Logged"),pch
=15,cex=1,col="#2D708EFF")
points(nmds.spider.pa, disp = "sites",select= which(spider_cleaned_2$treatment=="HardwoodContro
l"),pch=16,cex=1, col="#FDE725FF")
ordiellipse(nmds.spider.pa,spider_cleaned_2$treatment,draw="lines",col=c("#73D055FF","#481567F
F","#2D708EFF","#FDE725FF"), lwd=3,kind="sd",conf=0.95,label=FALSE)
legend("topleft",legend= c("Hemlock","Girdled","Logged","Hardwood"), pch=c(17,18,15,16),cex=1,bt
y="n",col=c("#73D055FF","#481567FF","#2D708EFF","#FDE725FF"))
```



4. Run a permutational multivariate analysis of variance (PERMANOVA) using `adonis2()`. Model the bray-curtis distance matrix as a function of treatment.

Provide the PERMANOVA output.

```
adonis2(dis.matrix.pa~spider1$treatment, permutations=999)
```

```
## Permutation test for adonis under reduced model
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dis.matrix.pa ~ spider1$treatment, permutations = 999)
##           Df SumOfSqs      R2      F Pr(>F)
## Model      3   0.9211 0.26771 1.4623  0.173
## Residual  12   2.5196 0.73229
## Total     15   3.4407 1.00000
```

```
library(pairwiseAdonis)
```

```
## Loading required package: cluster
```

```
pairwise.adonis(dis.matrix.pa, spider1$treatment)
```

##		pairs	Df	SumsOfSqs	F.Model	R2	p.value	p.adjusted	sig
## 1	Logged vs Girdled	1	0.29367973	1.3319950	0.18166884	0.341	1.000		
## 2	Logged vs Hemlock	1	0.47633776	2.1553416	0.26428588	0.066	0.396		
## 3	Logged vs Hardwood	1	0.49814502	2.3928361	0.28510459	0.045	0.270		
## 4	Girdled vs Hemlock	1	0.08785326	0.4148971	0.06467712	0.571	1.000		
## 5	Girdled vs Hardwood	1	0.22740760	1.1431793	0.16003788	0.337	1.000		
## 6	Hemlock vs Hardwood	1	0.25875632	1.2973633	0.17778522	0.348	1.000		

5. Interpret the output (using the figure and PERMANOVA results).

According to PERMANOVA, there is no significant differences among treatment groups ($p_{\text{adjusted}} = 0.187$). Treatment groups explain approximately 26.77% of the total variance, suggesting that the remaining 73.23% is explained by factors other than the treatment groups.

When looking at pairwise PERMANOVA, there are no significant differences between groups ($p_{\text{adjusted}} > 0.28$). Also, the differences between Logged and Hemlock, and Logged and Hardwood treatments suggest relatively strong differences compared to other paired groups.

Overall, however, all the treatment groups contribute small portion of total data variation, and there are no statistically clear results.

6. Now, some data wrangling for the understory herb data. Change the data set from long format to wide format using species as the taxonomic resolution (i.e., each column should be a plant species). Be sure that the new data frame includes the predictor variables year, block, trt, plot, and subplot. Then, subset the data by year 2008, so it aligns with the spider data. Lastly, double check that the number of samples per plot aligns between the spider and plant data sets. Each plot should have 10 samples (indicated as replicates or subplots)

```
library(dplyr)
herb.matrix <- dcast(herb, year+block+plot+trt ~ species, mean, value.var="cover", fill=0)
herb.matrix <- herb.matrix %>% rename(treatment = trt)
View(herb.matrix)
```

```
herb.matrix$year<-as.factor(herb.matrix$year)
herb.matrix$block<-as.factor(herb.matrix$block)
herb.matrix$plot<-as.factor(herb.matrix$plot)
herb.matrix$treatment<-as.factor(herb.matrix$treatment)
```

```
levels(herb.matrix$year)
```

```
## [1] "2003" "2004" "2005" "2006" "2007" "2008" "2009" "2010" "2011" "2012"
## [11] "2013" "2014" "2015" "2016" "2017" "2018" "2019" "2020" "2021" "2022"
## [21] "2023"
```

```
herb.matrix <- herb.matrix %>% filter(year=="2008") %>% droplevels()
herb.matrix <- herb.matrix[, colSums(herb.matrix !=0)>0]
View(herb.matrix)
```

7. Run a detrended correspondence analysis (DCA) to determine the most appropriate constrained model for these data. Provide the output. Use the original

data set with abundances.

```
DCA <- decorana(spider_cleaned_2[,6:56])
DCA
```

```
##
## Call:
## decorana(veg = spider_cleaned_2[, 6:56])
##
## Detrended correspondence analysis with 26 segments.
## Rescaling of axes with 4 iterations.
## Total inertia (scaled Chi-square): 1.7996
##
##              DCA1   DCA2   DCA3   DCA4
## Eigenvalues      0.5277 0.3071 0.14496 0.10451
## Additive Eigenvalues 0.5277 0.3024 0.14426 0.07513
## Decorana values    0.5472 0.1858 0.07301 0.03454
## Axis lengths      2.6433 2.5389 1.39815 1.27517
```

8. Run the appropriate model (RDA or CCA). Provide a figure.

```
spider.matrix.rda <- dcast(spiders, year+ block + plot + treatment ~ genus, sum,value.var="abundance",na.rm=TRUE)
View(spider.matrix.rda)

spider.matrix.rda$year<-as.factor(spider.matrix.rda$year)
spider.matrix.rda$block<-as.factor(spider.matrix.rda$block)
spider.matrix.rda$plot<-as.factor(spider.matrix.rda$plot)
spider.matrix.rda$treatment<-as.factor(spider.matrix.rda$treatment)

spider.matrix.rda <- spider.matrix.rda %>% select(-contains("imm"), -c(LinytoID, LinyToID, unk_toID))

View(spider.matrix.rda)
```

```
library(vegan)
spider.rda <- rda(spider.matrix.rda[,5:55] ~ ., herb.matrix[,5:41])
```

```
##
## Some constraints or conditions were aliased because they were redundant. This
## can happen if terms are linearly dependent (collinear): 'corcor', 'denobs',
## 'epirep', 'erehie', 'eurdiv', 'gootes', 'hamvir', 'hupluc', 'ilever', 'lysbor',
## 'lysqua', 'maican', 'medvir', 'mitrep', 'monuni', 'oclacu', 'osmcin', 'parnov',
## 'parqui', 'polacr', 'pyrmin', 'ruball', 'rubhis', 'rubida', 'uvuses', 'vacang',
## 'vaccor', 'vacsp', 'vibnud', 'ztot.cover'
```

```
##
## The model is overfitted with no unconstrained (residual) component
```

```
summary(spider.rda)
```

```
##
## Call:
## rda(formula = spider.matrix.rda[, 5:55] ~ arahis + aranud + berthu +      carpen + carsp + co
rcor + denobs + denpun + drysp + epirep +      erehie + eurdiv + gootes + hamvir + hupluc + ilev
er + lysbor +      lysqua + maican + medvir + mitrep + monuni + oclacu + osmcin +      parnov +
parqui + polacr + pyrmin + ruball + rubhis + rubida +      uvuses + vacang + vaccor + vacsp + vi
bnud + ztot.cover, data = herb.matrix[,      5:41])
##
## Partitioning of variance:
##              Inertia Proportion
## Total          4249          1
## Constrained     4249          1
## Unconstrained      0          0
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##              RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## Eigenvalue    1705.6156 1170.9117 833.2808 247.16656 196.69548 71.61410
## Proportion Explained  0.4014  0.2756  0.1961  0.05817  0.04629  0.01685
## Cumulative Proportion 0.4014  0.6769  0.8730  0.93121  0.97750  0.99436
##              RDA7
## Eigenvalue      23.983549
## Proportion Explained 0.005644
## Cumulative Proportion 1.000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##              RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## Eigenvalue    1705.6156 1170.9117 833.2808 247.16656 196.69548 71.61410
## Proportion Explained  0.4014  0.2756  0.1961  0.05817  0.04629  0.01685
## Cumulative Proportion 0.4014  0.6769  0.8730  0.93121  0.97750  0.99436
##              RDA7
## Eigenvalue      23.983549
## Proportion Explained 0.005644
## Cumulative Proportion 1.000000
```

```
mod0.rda <- rda(spider.matrix.rda[,5:55] ~ 1, herb.matrix[,5:41])
spider.rda.red <- ordistep(mod0.rda, scope = formula(spider.rda), direction = "both", permutatio
ns = how(nperm = 199))
```

```
##
## Start: spider.matrix.rda[, 5:55] ~ 1
##
##           Df      AIC      F Pr(>F)
## + aranud    1 66.816 2.6779 0.015 *
## + vaccor    1 66.345 3.2039 0.020 *
## + monuni    1 66.236 3.3303 0.025 *
## + maican    1 66.308 3.2458 0.025 *
## + carpen    1 66.699 2.8054 0.035 *
## + denobs    1 66.807 2.6870 0.040 *
## + ztot.cover 1 67.399 2.0672 0.055 .
## + denpun    1 67.260 2.2087 0.060 .
## + arahis    1 67.148 2.3250 0.105
## + uvuses    1 66.509 3.0170 0.115
## + medvir    1 66.509 3.0170 0.130
## + hupluc    1 67.781 1.6911 0.150
## + lysbor    1 67.893 1.5849 0.155
## + ruball    1 67.816 1.6574 0.210
## + hamvir    1 67.949 1.5320 0.235
## + eurdiv    1 67.949 1.5320 0.235
## + parqui    1 67.949 1.5320 0.255
## + corcor    1 67.949 1.5320 0.260
## + vacang    1 68.102 1.3890 0.265
## + rubhis    1 68.419 1.1014 0.300
## + oclacu    1 67.949 1.5320 0.315
## + carsp     1 68.431 1.0913 0.325
## + vibnud    1 67.949 1.5320 0.330
## + rubida    1 68.434 1.0883 0.370
## + vacsp     1 68.553 0.9840 0.535
## + lysqua    1 68.553 0.9840 0.540
## + mitrep    1 68.784 0.7848 0.590
## + erehie    1 68.932 0.6609 0.600
## + gootes    1 68.932 0.6609 0.615
## + epiresp   1 69.131 0.4971 0.730
## + pyrmin    1 69.131 0.4971 0.755
## + osmcin    1 69.131 0.4971 0.770
## + ilever    1 69.131 0.4971 0.780
## + berthu    1 69.131 0.4971 0.800
## + parnov    1 69.164 0.4704 0.905
## + drysp     1 69.366 0.3089 0.965
## + polacr    1 69.269 0.3860 1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: spider.matrix.rda[, 5:55] ~ aranud
##
##           Df      AIC      F Pr(>F)
## - aranud    1 67.768 2.6779 0.025 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           Df      AIC      F Pr(>F)
```

```

## + ztot.cover 1 64.930 3.1267 0.010 **
## + vaccor 1 65.285 2.7741 0.010 **
## + hamvir 1 65.306 2.7536 0.010 **
## + oclacu 1 65.306 2.7536 0.010 **
## + carpen 1 65.306 2.7536 0.015 *
## + eurdiv 1 65.306 2.7536 0.015 *
## + uvuses 1 65.306 2.7536 0.015 *
## + medvir 1 65.306 2.7536 0.020 *
## + vibnud 1 65.306 2.7536 0.020 *
## + hupluc 1 65.388 2.6747 0.020 *
## + monuni 1 65.306 2.7536 0.025 *
## + parqui 1 65.306 2.7536 0.025 *
## + vacang 1 65.527 2.5422 0.025 *
## + arahis 1 65.244 2.8139 0.030 *
## + denobs 1 65.306 2.7536 0.035 *
## + corcor 1 65.306 2.7536 0.035 *
## + maican 1 65.306 2.7536 0.040 *
## + denpun 1 65.832 2.2605 0.085 .
## + ruball 1 66.206 1.9281 0.105
## + rubida 1 67.028 1.2522 0.275
## + rubhis 1 67.003 1.2717 0.280
## + carsp 1 67.256 1.0760 0.430
## + erehie 1 67.389 0.9764 0.475
## + lysqua 1 67.409 0.9614 0.495
## + mitrep 1 67.611 0.8127 0.520
## + gootes 1 67.389 0.9764 0.530
## + vacsp 1 67.409 0.9614 0.550
## + parnov 1 67.785 0.6878 0.705
## + lysbor 1 67.717 0.7361 0.715
## + drysp 1 68.277 0.3482 0.890
## + epirep 1 68.289 0.3400 0.925
## + pyrmin 1 68.289 0.3400 0.930
## + ilever 1 68.289 0.3400 0.935
## + berthu 1 68.289 0.3400 0.940
## + osmcin 1 68.289 0.3400 0.940
## + polacr 1 68.289 0.3403 0.975
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: spider.matrix.rda[, 5:55] ~ aranud + ztot.cover
##
##           Df    AIC      F Pr(>F)
## - ztot.cover 1 66.816 3.1267 0.005 **
## - aranud     1 67.399 3.7419 0.005 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           Df    AIC      F Pr(>F)
## + eurdiv 1 60.149 5.3357 0.005 **
## + monuni 1 60.149 5.3357 0.005 **
## + uvuses 1 60.149 5.3357 0.005 **
## + vaccor 1 60.121 5.3686 0.010 **

```

```

## + denobs 1 60.149 5.3357 0.010 **
## + oclacu 1 60.149 5.3357 0.010 **
## + parqui 1 60.149 5.3357 0.010 **
## + corcor 1 60.149 5.3357 0.010 **
## + carpen 1 60.149 5.3357 0.015 *
## + maican 1 60.149 5.3357 0.015 *
## + medvir 1 60.149 5.3357 0.015 *
## + vibnud 1 60.149 5.3357 0.015 *
## + hamvir 1 60.149 5.3357 0.015 *
## + hupluc 1 60.553 4.8764 0.015 *
## + vacang 1 60.724 4.6883 0.020 *
## + arahis 1 63.676 2.0077 0.205
## + ruball 1 64.866 1.1772 0.305
## + denpun 1 64.845 1.1907 0.345
## + mitrep 1 65.059 1.0535 0.375
## + rubhis 1 65.430 0.8249 0.440
## + rubida 1 65.427 0.8263 0.490
## + erehie 1 65.769 0.6249 0.620
## + parnov 1 65.823 0.5937 0.665
## + gootes 1 65.769 0.6249 0.675
## + carsp 1 65.916 0.5406 0.715
## + lysqua 1 65.999 0.4936 0.715
## + lysbor 1 65.870 0.5666 0.720
## + vacsp 1 65.999 0.4936 0.745
## + pyrmin 1 66.005 0.4901 0.750
## + osmcin 1 66.005 0.4901 0.755
## + epirep 1 66.005 0.4901 0.795
## + drysp 1 66.035 0.4735 0.805
## + ilever 1 66.005 0.4901 0.810
## + berthu 1 66.005 0.4901 0.835
## + polacr 1 66.377 0.2863 0.930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: spider.matrix.rda[, 5:55] ~ aranud + ztot.cover + eurdiv
##
##           Df    AIC      F Pr(>F)
## - eurdiv    1 64.930 5.3357 0.005 **
## - ztot.cover 1 65.306 5.7850 0.005 **
## - aranud     1 66.691 7.6350 0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           Df    AIC      F Pr(>F)
## + ruball    1 59.736 1.0564 0.425
## + erehie    1 59.448 1.2053 0.430
## + gootes    1 59.448 1.2053 0.455
## + lysbor    1 59.396 1.2325 0.460
## + hupluc    1 59.396 1.2325 0.470
## + denpun    1 59.695 1.0770 0.470
## + drysp     1 60.011 0.9193 0.485
## + rubhis    1 59.856 0.9957 0.490

```

```
## + rubida 1 59.823 1.0123 0.495
## + lysqua 1 59.975 0.9372 0.500
## + carsp 1 59.990 0.9297 0.500
## + parnov 1 59.396 1.2325 0.505
## + berthu 1 59.828 1.0098 0.505
## + pyrmin 1 59.828 1.0098 0.505
## + arahis 1 59.890 0.9791 0.505
## + ilever 1 59.828 1.0098 0.515
## + vaccor 1 59.828 1.0098 0.520
## + vacsp 1 59.975 0.9372 0.520
## + osmcin 1 59.828 1.0098 0.540
## + vacang 1 59.975 0.9372 0.550
## + epirep 1 59.828 1.0098 0.585
## + polacr 1 60.639 0.6232 0.650
## + mitrep 1 60.909 0.5032 0.690
## + carpen 0 60.149
## + corcor 0 60.149
## + denobs 0 60.149
## + hamvir 0 60.149
## + maican 0 60.149
## + medvir 0 60.149
## + monuni 0 60.149
## + oclacu 0 60.149
## + parqui 0 60.149
## + uvuses 0 60.149
## + vibnud 0 60.149
```

```
summary(spider.rda.red)
```

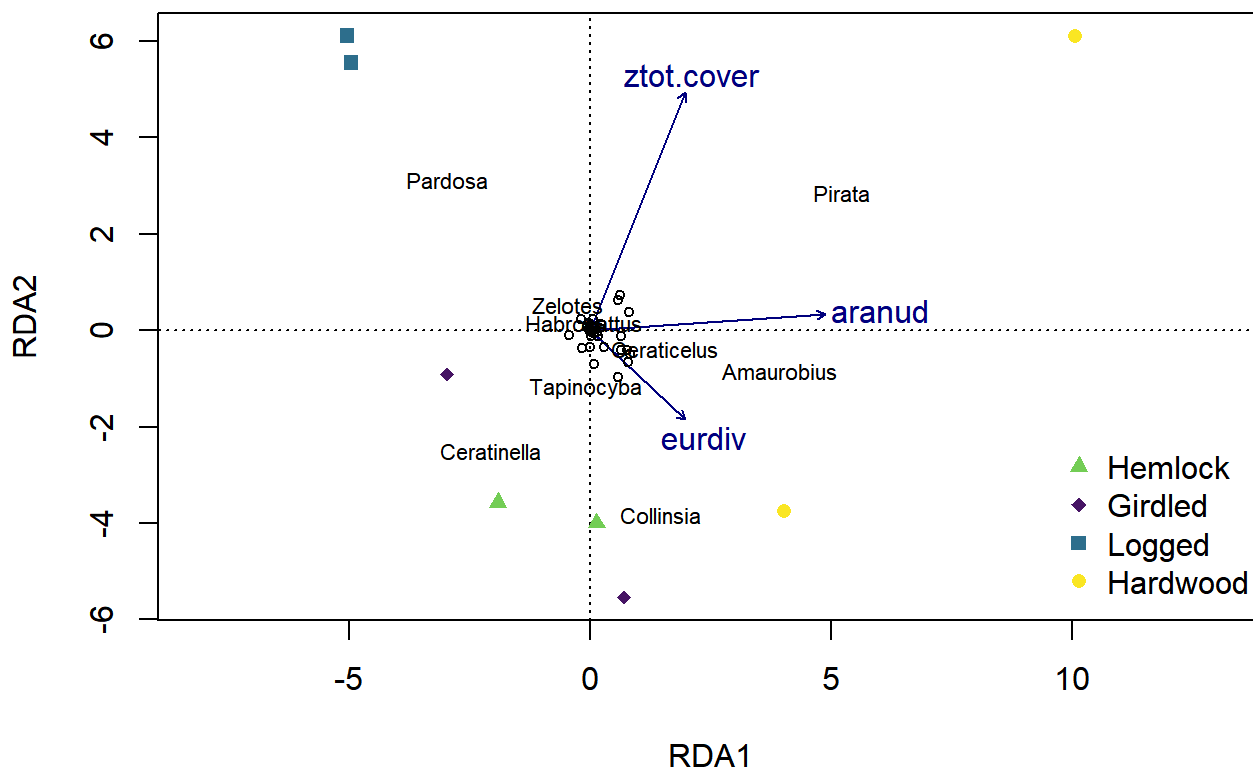


```
##
## Call:
## rda(formula = spider.matrix.rda[, 5:55] ~ aranud + ztot.cover +      eurdiv, data = herb.matr
ix[, 5:41])
##
## Partitioning of variance:
##           Inertia Proportion
## Total      4249.3      1.0000
## Constrained 3474.8      0.8177
## Unconstrained 774.5      0.1823
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##           RDA1      RDA2      RDA3      PC1      PC2
## Eigenvalue    1617.1570 1123.8860 733.7342 322.57753 262.13231
## Proportion Explained    0.3806    0.2645    0.1727    0.07591    0.06169
## Cumulative Proportion    0.3806    0.6451    0.8177    0.89365    0.95534
##           PC3      PC4
## Eigenvalue    128.58928 61.1915
## Proportion Explained    0.03026    0.0144
## Cumulative Proportion    0.98560    1.0000
##
## Accumulated constrained eigenvalues
## Importance of components:
##           RDA1      RDA2      RDA3
## Eigenvalue    1617.1570 1123.8860 733.7342
## Proportion Explained    0.4654    0.3234    0.2112
## Cumulative Proportion    0.4654    0.7888    1.0000
```

```
anova(spider.rda.red, by = 'axis')
```

```
## Permutation test for rda under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = spider.matrix.rda[, 5:55] ~ aranud + ztot.cover + eurdiv, data = herb.ma
trix[, 5:41])
##           Df Variance      F Pr(>F)
## RDA1       1 1617.16 8.3521 0.002 **
## RDA2       1 1123.89 5.8045 0.004 **
## RDA3       1  733.73 3.7895 0.046 *
## Residual   4   774.49
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ordiplot(spider.rda.red, display = c('si', 'cn'), type = 'n')
points(spider.rda.red, dis = "sites", select = which(herb.matrix$treatment=="hemlock"), pch = 17, cex = 1, col = "#73D055FF")
points(spider.rda.red, dis = "sites", select = which(herb.matrix$treatment=="girdled"), pch = 18, cex = 1, col = "#481567FF")
points(spider.rda.red, dis = "sites", select = which(herb.matrix$treatment=="logged"), pch = 15, cex = 1, col = "#2D708EFF")
points(spider.rda.red, dis = "sites", select = which(herb.matrix$treatment=="hardwood"), pch = 16, cex = 1, col = "#FDE725FF")
text(spider.rda.red, display = 'cn', col = 'navy', cex = 1)
orditorp(spider.rda.red, display = 'sp')
legend("bottomright", legend = c("Hemlock", "Girdled", "Logged", "Hardwood"), pch = c(17, 18, 15, 16), cex = 1, bty = "n", col = c("#73D055FF", "#481567FF", "#2D708EFF", "#FDE725FF"))
```



9. Interpret the results.

According to the Redundancy Analysis, the constrained model accounts for 61.49% of the overall data variability (RDA1: $p = 0.005$, RDA2: $p = 0.005$). Environmental variables also appear to have a relatively strong influence on the herb genera arahis, monuni, and carpen.

Specifically, monuni and carpen are likely to grow better in Hardwood environments, relatively. Additionally, the spider genus Pirate seems to occur more frequently in areas where monuni and carpen grow.

In logged hemlock plots, the herb genus arahis is more likely to appear.