

# assignment\_three\_radcliffe

Don Radcliffe

2/1/2022

## 1

1a), read in data:

```
require(dplyr)
require(tidyr)
require(stringr)
require(tibble)
require(janitor)

require(here)

require(vegan)
require(labdsv)

oak_raw <- read.csv(here('data/Oak_data_47x216.csv'), stringsAsFactors = TRUE) %>%
  rename('stand' = X)
## Just a few rows and columns to prove it's read in:
head(oak_raw[,1:5])
```

```
##      stand Elev.m LatAppx LongAppx Slope.deg
## 1 Stand01      91 44.4831 123.3618         1
## 2 Stand02     106 44.6613 123.2029         7
## 3 Stand03     152 44.7625 123.2641         6
## 4 Stand04      91 45.0081 123.2233         5
## 5 Stand05     152 45.1381 123.3355        10
## 6 Stand06      91 44.3974 123.3215         5
```

1b), I'm subsetting based on DrainageClass, to get stands with good- or well-drained soils only.

```
oak_subset <- oak_raw %>%
  filter(DrainageClass == 'Good' | DrainageClass == 'Well') %>%
  column_to_rownames(var = 'stand')
head(oak_subset[,1:5])
```

```
##      Elev.m LatAppx LongAppx Slope.deg AspClass
## Stand03     152 44.7625 123.2641         6 S.or.W
## Stand05     152 45.1381 123.3355        10 N.or.E
## Stand06      91 44.3974 123.3215         5 N.or.E
## Stand07     213 43.9023 123.3353         6 SE.or.NW
## Stand08     106 45.3837 123.1825        11 SE.or.NW
## Stand10     167 45.3259 123.1417         7      SW
```

1c, I subset species to those present in greater than 5% of the plots, because this is one common method of

reducing the influence of rare species on community level statistics.

```
oak_subset_species <- oak_subset %>%
  select(Abgr.s:Zice) %>%
  vegtab(minval = 0.05 * nrow(oak_subset)) %>%
  rownames_to_column(var = 'stand')
## We can use the str() calls below to show that we've cut the data down,
## from 217 variables to 126 variables, by removing species in less than 5% of plots (and the response)
str(oak_subset_species)
str(oak_subset)
```

1d), I would like to test the effects of logging (cat.) and head load (cont.), two considerations I've been interested in my current and past research.

I expect that community composition will be different between historically logged and unlogged sites, because logging lets light through to the understory in gaps and therefore facilitates a wider range of understory species, and often richer communities.

I expect that community composition will be differ by heat load heat load, because oak forests are often in relatively dry sites, dry site plant species/communities are often determined by moisture, and heat load has a major effect of evaporative pressure, and therefore moisture dynamics.

I expect that the interaction effect of logging and heat load will be an important determinate of plant community composition, because of the interaction of moisture and light limitation; for example, communities and some species may be more limited in their ability to respond to canopy opening on dry sites where moisture is more limiting than light.

To facilitate my analyses, I've formatted my data to be as simple and intuitive as possible, and be in a format that I use in my own work.

```
## A table so we can rejoin to explanatory variables of interest
oak <- oak_subset %>%
  select(NotLogged, HeatLoad) %>%
  rownames_to_column(var = 'stand') %>%
  full_join(oak_subset_species, by = 'stand') %>%
  ## Get the capital letters out of the stand column's values
  mutate(stand = tolower(stand),
    .keep = 'unused',
    .before = NotLogged) %>%
  ## Make logging category easier to interpret
  mutate(logged = case_when(
    NotLogged == 'Yes' ~ 'not_logged',
    NotLogged == 'No' ~ 'logged'),
    .keep = 'unused',
    .after = stand) %>%
  mutate(logged = as.factor(logged)) %>%
  ## Round the heat load and change the name
  mutate(heat_load = round(HeatLoad, 3),
    .keep = 'unused',
    .after = logged) %>%
  ## I'm a snake case guy and would rather deal with species names that way (lowercase with underscores)
  janitor::clean_names(case = 'snake')
head(oak[,1:5])
```

```
##      stand logged heat_load quga_t rhdi_s
## 1 stand03 logged    0.930   265    31.0
## 2 stand05 logged    0.815   108     0.5
## 3 stand06 logged    0.876   134   13.0
```

```
## 4 stand07 logged      0.905    130    24.0
## 5 stand08 logged      0.870    254    22.0
## 6 stand10 logged      0.941    178    27.0
```

1e), I've already applied some adjustments: subsetting by soil drainage and species presence, as well as my extra formatting

I've also decided to remove woody species (those ending in `_s` or `_t`). I've done this for several reasons: the coverage values for the woody species in both tree and shrub layer tend to be *a lot* higher than those of herbaceous species, the tree layer was measured in different units, and I was thinking of understory plants when I formulated my hypotheses.

I didn't want to have to relativize because of different units. I am usually not a fan of relativizing data, because I believe the absolute values of attributes like coverage, density, basal area, etc., contain more interpretable and ecologically useful information than relativized data.

Below I've checked for empty stands, and looked at the distribution of our categorical variable, the heat load index.

```
## Remove trees and shrubs
oak_herb <- oak %>%
  select(-ends_with('_s'), - ends_with('_t'))
  ## we're down to 82 species now.
str(oak_herb)
```

```
## 'data.frame':   39 obs. of  85 variables:
## $ stand      : chr  "stand03" "stand05" "stand06" "stand07" ...
## $ logged      : Factor w/ 2 levels "logged","not_logged": 1 1 1 1 1 1 1 1 1 1 ...
## $ heat_load   : num  0.93 0.815 0.876 0.905 0.87 0.941 0.69 0.809 0.69 0.831 ...
## $ gal         : num  2 0 0.5 0.5 7 0.5 3 0.5 3 2 ...
## $ osnu        : num  0.5 0.5 0.5 0.5 0.5 0.5 0 0.5 0.5 0.5 ...
## $ povu        : num  0.2 0.2 0.2 0.2 0 ...
## $ pomu        : num  0.5 5 11 0.5 0.2 ...
## $ toar        : num  0.5 0.5 0.5 0.5 7 2 0.5 1 1 0.5 ...
## $ brla        : num  0.5 0 0.5 0.5 0.5 0.5 0.5 0 0.5 0.5 ...
## $ hype        : num  0.5 0 0.5 0 0.5 ...
## $ elgl        : num  0.5 0 0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
## $ tegr        : num  0 0.5 0.5 0 0 0.5 0.5 0 4 4 ...
## $ sado        : num  0.5 0 0.5 0.5 0 0 0.5 0.5 0.5 0.5 ...
## $ sacr        : num  0 0 0 0 1 0.5 1 0.5 0.5 0.5 ...
## $ hola        : num  2 0 1 0.5 0 0.5 0 0 0 0 ...
## $ viam        : num  0.5 0.5 0.5 0.5 1 0.5 0 0 0 0 ...
## $ liap        : num  0 0 0 0 0.5 0 28 0.5 5 1 ...
## $ adbi        : num  0.5 0 3 0.5 0.5 ...
## $ popr        : num  0.5 0 0.5 0.5 0.5 0 0 0.5 0 0 ...
## $ agte        : num  1 0 0.5 0.5 0.5 0 0 0.5 0.5 0 ...
## $ frvi        : num  0.5 0 0.5 0.5 0 0 0.5 0.5 0 0.5 ...
## $ ptaq        : num  6 0.5 6 0.5 0 0 0.5 0 0 0 ...
## $ dagl        : num  2 0 1 0 0.2 ...
## $ mebu        : num  0.5 0 0 0 0 ...
## $ feru        : num  0 0 0.5 0 0 0 0 0.5 0.5 0 ...
## $ trla        : num  0.5 0 0 0.5 0 0 0.5 0.5 0.5 0.5 ...
## $ irte        : num  0 0 0.5 0 0 0.5 0 0 0 0 ...
## $ car         : num  0.5 0 0.5 0 0 0.5 0 0 0 0 ...
## $ cipa        : num  0 0 0 0 0 0 0.5 0 5 1 ...
## $ lasa        : num  0 0 0 0 0.5 0.5 0.5 0 0 0 ...
## $ acmi        : num  0 0 0.5 0 0 0 0 0 0 0 ...
```

```

## $ geog      : num  0 0 0.5 0 0.2 ...
## $ brri      : num  0 0 0 0 0 0 0 0.5 0 0 ...
## $ daca      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ nepa      : num  0 0 0 0 0 0 0 0.5 0.5 0.5 ...
## $ erog      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ taof      : num  0 0 0.5 0 0 ...
## $ drar      : num  0 0 0 0 0 0 0 0 10 7 ...
## $ mope      : num  0 0 0 0 2 0 0.5 0 1 0 ...
## $ thoc      : num  0 0 0 0 0 0 1 0.5 0 0 ...
## $ agse      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ dacar     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ cyec      : num  0 0 0.5 0 0 0 0 0 0 0 ...
## $ feoc      : num  0 0 0 0 0 ...
## $ mosi      : num  0 0 0 0 0 ...
## $ fecar     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ gepu      : num  0 0 0 0 0 7 0 0 0 0 ...
## $ agha      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ acar      : num  0 1 0 0 0 0 0 0 0 0 ...
## $ erla      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ kocr      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ plla      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ pogl      : num  0 0 0 0 0 0 0.5 0 0 0.5 ...
## $ vinu      : num  0 0 0 0 0 0.5 0 0 0 0 ...
## $ apan      : num  0 0 0 0.5 0 0 0 0 0 0 ...
## $ brpu      : num  0.5 0 0 0 0.5 ...
## $ phpr      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ poco      : num  0 0 0 0 0 ...
## $ trov      : num  0 0 0 0 0 0 0.5 0 0 0 ...
## $ cato      : num  0 0 0.2 0 0 ...
## $ pogr      : num  0 0 0.5 0 0 0 0 0 0 0 ...
## $ smse      : num  0 0 0 0 0 0 0 0 0 0.5 ...
## $ smra      : num  0 1 0 0 0 0 0 0 0 0 ...
## $ aica      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ all       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ arel      : num  0.5 0 0 0 0 0 0 0 0 0 ...
## $ brse      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ brvu      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ ceum      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ deel      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ frbr      : num  0 0 0 0 0 0.5 0 0 0 0 ...
## $ frcu      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ hyoc      : num  0 0 0 0 0 0 0 0 0.5 0.5 ...
## $ lapo      : num  0 0 0 0 0 0 0 0.5 0 0 ...
## $ lope      : num  0 0 0.5 0 0 0 0 0 0 0 ...
## $ mila      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ trif      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ trpr      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ cost      : num  0 0 0 0 0 ...
## $ crca      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ lico      : num  0 0 0 0 0 ...
## $ prvu      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ ruac      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ aqfo      : num  0 0 0 0 0.2 ...
## $ civu      : num  0 0 0 0 0 0 0 0 0 0 ...

```

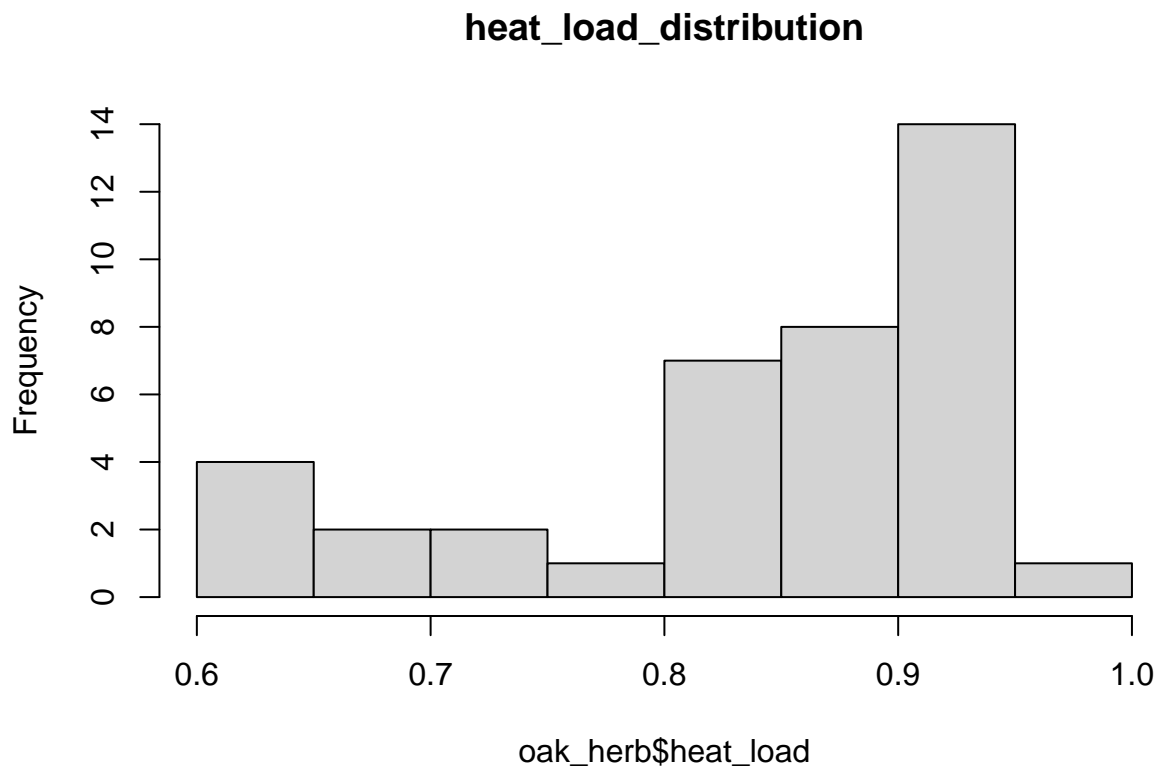
Check for empty stands: there are none!

```
oak_empty <- oak_herb %>%  
  ## Take out categorical variables  
  select(-stand, -heat_load, -logged) %>%  
  ## Add across rows to get total coverage of selected species  
  rowSums()  
oak_empty
```

```
## [1] 21.2  9.7 32.4  8.2 23.8 18.2 57.6 10.1 51.3 30.5 19.1 16.1 13.7 67.2 60.4  
## [16] 21.9  2.2 72.4  6.8 22.6 71.7 19.8 93.1 55.4 17.5 47.2 82.7 31.4  3.6 19.1  
## [31] 70.0 22.9 88.8 25.2 65.1  4.1 18.8  9.2 13.2
```

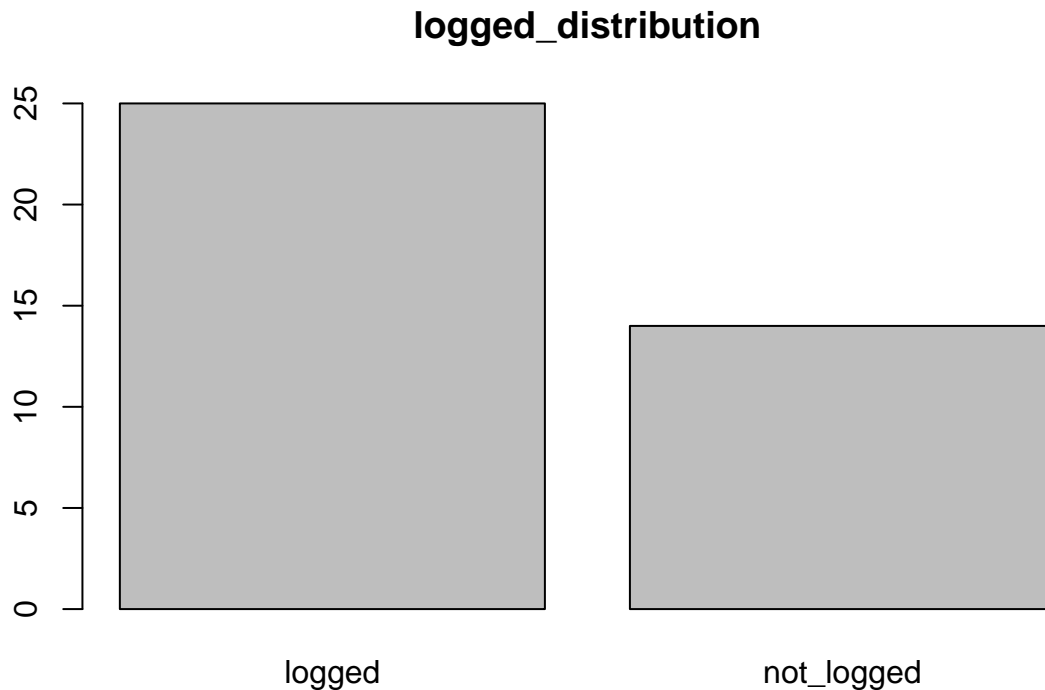
Check the distribution of the heat load data: decidedly not normal, but not transforming because we are running non-parametric permutation tests for this assignment.

```
hist(oak_herb$heat_load, main = 'heat_load_distribution', breaks = 10)
```



Check the distribution of the logged data: more logged stands than unlogged.

```
plot(oak_herb$logged, main = 'logged_distribution')
```



1f) I will choose to use the Bray-Curtis distance, because I am interested in preserving the information on abundance of species, rather than just presence-absence, and because the notes argue that Bray-Curtis distance works best for community data. In our case the ‘abundance’ data of herbaceous species as labeled in the metadata appear to be percent coverage and not actually abundance, and I am unsure of whether this is 100% okay to use with Bray-Curtis but moving ahead.

## 2

2a) The simple models support my hypothesis that both logging and heat load index are important drivers of plant community composition.

*## create separate dataframe for each explanatory variable and for the response variables.*

```
logged <- oak_herb %>%
  select(stand, logged) %>%
  column_to_rownames(var = 'stand')

heat_load <- oak_herb %>%
  select(stand, heat_load) %>%
  column_to_rownames(var = 'stand')

species_coverage <- oak_herb %>%
  select(stand, gal:erla) %>%
  column_to_rownames(var = 'stand')

adonis_logged <- adonis2(formula = species_coverage ~ logged,
  data = oak_herb, method = 'bray')
```

```
adonis_logged
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species_coverage ~ logged, data = oak_herb, method = "bray")
##           Df SumOfSqs      R2      F Pr(>F)
## logged     1   0.6323 0.05652 2.2165  0.03 *
## Residual  37  10.5555 0.94348
## Total     38  11.1878 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis_heat_load <- adonis2(formula = species_coverage ~ heat_load,
                             data = oak_herb, method = 'bray')
adonis_heat_load
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species_coverage ~ heat_load, data = oak_herb, method = "bray")
##           Df SumOfSqs      R2      F Pr(>F)
## heat_load   1   0.9513 0.08503 3.4385 0.004 **
## Residual   37  10.2365 0.91497
## Total      38  11.1878 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2b) I've decided to include an interaction term, because as I stated in my hypothesis I believe there will be an interaction of light availability and moisture dynamics for some plant species/communities.

```
adonis_combined <- adonis2(formula = species_coverage ~ logged + heat_load + logged:heat_load,
                             data = oak_herb, method = 'bray')
adonis_combined
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species_coverage ~ logged + heat_load + logged:heat_load, data = oak_herb, method = "bray")
##           Df SumOfSqs      R2      F Pr(>F)
## logged           1   0.6323 0.05652 2.4263  0.023 *
## heat_load         1   0.8612 0.07698 3.3045  0.006 **
## logged:heat_load   1   0.5729 0.05121 2.1985  0.032 *
## Residual          35   9.1213 0.81529
## Total             38  11.1878 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2c) I've reordered the individual terms, but left the interaction term at the end for both models.

```
adonis_remixed <- adonis2(formula = species_coverage ~ heat_load + logged + logged:heat_load,
                          data = oak_herb, method = 'bray')
adonis_remixed

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species_coverage ~ heat_load + logged + logged:heat_load, data = oak_herb, method = 'bray')
##              Df SumOfSqs      R2      F Pr(>F)
## heat_load      1   0.9513 0.08503 3.6503  0.002 **
## logged          1   0.5422 0.04846 2.0805  0.043 *
## heat_load:logged 1   0.5729 0.05121 2.1985  0.041 *
## Residual       35   9.1213 0.81529
## Total          38  11.1878 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2d) For both logged and heat\_load, more variance in species composition was explained by that variable when it was put first in the model, and the effect can be seen in the sum of squares, r squared, and p value estimates. This tells me that some variance could be explained by either logging or heat load index.

However, the conclusions aren't likely to be affected, all terms were comfortably statistically significant in all models, if judged with a classical p-value threshold of 0.05.

2e) Changing to type III sums of squares changed my possible conclusions drastically, because I only got an estimate for my interaction effect! I'm assuming this means that it is possible to explain all variance with the interaction effect, which makes sense because it includes both variables. When I took the interaction effect out of the type III sums of squares model, I got estimates for both heat load and logging. This changed the conclusion by making logging insignificant, at least with a p-value threshold of 0.05. It also more than doubled the p value for heat load, although it was still fairly low.

Adonis2() also gave me the option to run with by = NULL, which assesses the significance of all the terms together. It comes out at 0.001, which is the lowest possible p value with 999 iterations. I don't find this 'all terms together' result very interesting though.

```
## same model as already run in default, running again to put them all together with all options manual
adonis_type_1 <- adonis2(formula = species_coverage ~ heat_load + logged + logged:heat_load,
                        data = oak_herb, method = 'bray', by = 'terms')
adonis_type_1
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species_coverage ~ heat_load + logged + logged:heat_load, data = oak_herb, method = 'bray', by = 'terms')
##              Df SumOfSqs      R2      F Pr(>F)
## heat_load      1   0.9513 0.08503 3.6503  0.003 **
## logged          1   0.5422 0.04846 2.0805  0.039 *
## heat_load:logged 1   0.5729 0.05121 2.1985  0.037 *
## Residual       35   9.1213 0.81529
## Total          38  11.1878 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
## same model as already run in default, running again to put them all together with all options manual
adonis_type_3 <- adonis2(formula = species_coverage ~ heat_load + logged + logged:heat_load,
                        data = oak_herb, method = 'bray', by = 'margin')
adonis_type_3
```

```
## Permutation test for adonis under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species_coverage ~ heat_load + logged + logged:heat_load, data = oak_herb, method = "bray", by = "margin")
##          Df SumOfSqs      R2      F Pr(>F)
## heat_load:logged  1    0.5729 0.05121 2.1985  0.032 *
## Residual        35    9.1213 0.81529
## Total           38   11.1878 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis_type_3_ni <- adonis2(formula = species_coverage ~ heat_load + logged,
                           data = oak_herb, method = 'bray', by = 'margin')
adonis_type_3_ni
```

```
## Permutation test for adonis under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species_coverage ~ heat_load + logged, data = oak_herb, method = "bray", by = "margin")
##          Df SumOfSqs      R2      F Pr(>F)
## heat_load  1    0.8612 0.07698 3.1981  0.010 **
## logged     1    0.5422 0.04846 2.0135  0.055 .
## Residual   36    9.6943 0.86651
## Total      38   11.1878 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis_type_null <- adonis2(formula = species_coverage ~ heat_load + logged,
                           data = oak_herb, method = 'bray', by = NULL)
adonis_type_null
```

```
## Permutation test for adonis under reduced model
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = species_coverage ~ heat_load + logged, data = oak_herb, method = "bray", by = NULL)
##          Df SumOfSqs      R2      F Pr(>F)
## Model      2    1.4935 0.13349 2.7731  0.002 **
## Residual   36    9.6943 0.86651
## Total      38   11.1878 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2f) I'm going to just compare p-values to keep it simple. Below I made a dataframe of the results for easy visualization.

The more complex models had lower p-values for the same variable as the simple model when that variable

was the first term in the model, but higher p-values when that variable was the second term in the model. The higher p-values when the variable was the second term are easier to explain: they're likely due to shared variance between disproportionately attributed to the first term. The lower p-values in the more complex model may reflect lower residuals in the model when additional terms help improve the fit. It's also possible that some of these patterns occurred due to random chance with the permutations, particularly with `heat_load` which has pretty small p-values/pretty high significance.

I was surprised the interaction term wasn't identical between the two complex models with different ordered terms. I'm wondering if this is due to random chance of permutations?

The type three sums of squares gave more conservative p value estimates, as the notes advertised, when the interaction factor was excluded from the model. This was particularly evident for the logging term, and would likely change the interpretation of the results.

What I learned from comparing these models was a reiteration of the sensitivity of statistical tests to all the decisions we make about implementing them. I'm not super alarmed, however, because the absolute magnitude of the differences doesn't seem catastrophically large. This is especially true if the researcher is evaluating p values as continuous and therefore avoiding the threshold effect around 0.05, and if the researcher is looking at a more holistic set of metrics than p-values alone.

```
model_p_values <- data.frame(
  'model' = c('simple_logging', 'simple_heat', 'logging_heat', 'heat_logging', 'type_three'),
  'logging' = c(0.035, NA, 0.028, 0.038, 0.058),
  'heat' = c(NA, 0.003, 0.007, 0.001, 0.008),
  'interaction' = c(NA, NA, 0.027, 0.030, NA))
model_p_values
```

##	model	logging	heat	interaction
## 1	simple_logging	0.035	NA	NA
## 2	simple_heat	NA	0.003	NA
## 3	logging_heat	0.028	0.007	0.027
## 4	heat_logging	0.038	0.001	0.030
## 5	type_three	0.058	0.008	NA