

# assignment\_three\_radcliffe

Don Radcliffe

2/3/2022

## 1

1a, read in data:

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

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, subset to 70-90% of plots. Doing so based on DrainageClass:

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

```
##      stand Elev.m LatAppx LongAppx Slope.deg
## 1 Stand03     152 44.7625 123.2641         6
## 2 Stand05     152 45.1381 123.3355        10
## 3 Stand06      91 44.3974 123.3215         5
## 4 Stand07     213 43.9023 123.3353         6
## 5 Stand08     106 45.3837 123.1825        11
## 6 Stand10     167 45.3259 123.1417         7
```

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. I did this manually before seeing in the notes that the `vegtab()` function would do this for me, woops! But, figuring out how to do it manually was also an instructive process:

```
oak_species_subset <- oak_subset %>%
  ## Select just the species columns
  select(c(Abgr.s:Zice)) %>%
```

```

## Turn it into presence absence data
mutate_all(function(x) {
  case_when(
    x > 0 ~ 1,
    x == 0 ~ 0)
})

## colsums to add up how many plots each species is present in.
## can't seem to do this in dataframe format without breaking the pipe:
oak_species_less_than_five_percent <- data.frame(colSums(oak_species_subset)) %>%
  ## Filter to those species present in 2 plots or less, the 5% threshold is between 2 and 3.
  filter(colSums.oak_species_subset. < 3) %>%
  ## Make it a vector of the row names, which are the species in less than 5% of plots
  row.names()

## Now select those rownames out of our oak_subset data,
## to get our final dataframe ready for analysis
oak1 <- select(oak_subset, -oak_species_less_than_five_percent)

## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(oak_species_less_than_five_percent)` instead of `oak_species_less_than_five_percent` t
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.

## Let's give it a more wieldly name, I d

## We can use the str() calls below to show that we've cut the data down,
## from 217 variables to 123 variables, by removing speices in less than 5% of plots
str(oak1)
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 species richness is greater on sites with a history of logging, because it let light through to the understory in gaps and therefore facilitates a wider range of understory species.

I expect that species richness is greater on sites with less heat loading, because oaks grow on dry sites to begin with, and I've filtered to high levels of soil drainage, so that the driest sites (with the highest heat loads) may be too dry for many species. It's also possible that species richness is greater on sites with more heat loading because there tends to be more open canopy space at drier sites.

I expect that species richness will be greatest with the interaction of logging with low heat loading, because of the combination of opening canopy space with relatively more moisture that may be tolerable for more species.

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.

```

## Don friendly data formatting:
oak <- oak1 %>%
  ## Subset to only the data I will use.
  select(stand, NotLogged, HeatLoad, Abgr.t:Vinu) %>%
  ## 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) %>%
## 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')
str(oak)

```

```

## 'data.frame':   39 obs. of  98 variables:
## $ stand      : chr  "stand03" "stand05" "stand06" "stand07" ...
## $ logged      : chr  "logged" "logged" "logged" "logged" ...
## $ heat_load   : num  0.93 0.815 0.876 0.905 0.87 0.941 0.69 0.809 0.69 0.831 ...
## $ abgr_t      : num  0 0.5 0 0 0 0 0 0 0 0 ...
## $ acar        : num  0 1 0 0 0 0 0 0 0 0 ...
## $ acma_s      : num  0.2 0.5 5 1 0 ...
## $ acma_t      : num  0 27 2 0.5 0 0 0 0.5 0 3 ...
## $ acmi        : num  0 0 0.5 0 0 0 0 0 0 0 ...
## $ adbi        : num  0.5 0 3 0.5 0.5 ...
## $ agha        : num  0 0 0 0 0 0 0 0 0 0 ...
## $ agse        : num  0 0 0 0 0 0 0 0 0 0 ...
## $ agte        : num  1 0 0.5 0.5 0.5 0 0 0.5 0.5 0 ...
## $ amal_s      : num  3 3 12 0.5 3 3 0.5 2 0 0.5 ...
## $ apan        : num  0 0 0 0.5 0 0 0 0 0 0 ...
## $ arme_s      : num  0 0 0 3 0 0 0 0 0 0.5 ...
## $ beaq_s      : num  0 0.5 0 0 0 0 0 0 0.5 1 ...
## $ brla        : num  0.5 0 0.5 0.5 0.5 0.5 0.5 0 0.5 0.5 ...
## $ brpu        : num  0.5 0 0 0 0.5 ...
## $ brri        : num  0 0 0 0 0 0 0 0.5 0 0 ...
## $ car         : num  0.5 0 0.5 0 0 0.5 0 0 0 0 ...
## $ cato        : num  0 0 0.2 0 0 ...
## $ cipa        : num  0 0 0 0 0 0 0.5 0 5 1 ...
## $ coco_s      : num  15 22 0 2 0.2 ...
## $ conu_t      : num  0 0 1 0 0 0 0 0 0 0 ...
## $ crdo_s      : num  0.2 0.5 1 0.2 0 ...
## $ crox_s      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ cyec        : num  0 0 0.5 0 0 0 0 0 0 0 ...
## $ daca        : num  0 0 0 0 0 0 0 0 0 0 ...
## $ dacar       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ dagl        : num  2 0 1 0 0.2 ...
## $ drar        : num  0 0 0 0 0 0 0 0 10 7 ...
## $ elgl        : num  0.5 0 0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
## $ erla        : num  0 0 0 0 0 0 0 0 0 0 ...
## $ erog        : num  0 0 0 0 0 0 0 0 0 0 ...
## $ feca        : num  0 0 0 0 0 0 0 0 0 0 ...
## $ feoc        : num  0 0 0 0 0 ...
## $ feru        : num  0 0 0.5 0 0 0 0 0.5 0.5 0 ...
## $ frla_s      : num  0 0 0 0 0 0 0 0 0 0 ...

```

```

## $ frla_t : num 0 0 0.5 0 0 0 0 0 0 0 ...
## $ frvi : num 0.5 0 0.5 0.5 0 0 0.5 0.5 0 0.5 ...
## $ gal : num 2 0 0.5 0.5 7 0.5 3 0.5 3 2 ...
## $ geog : num 0 0 0.5 0 0.2 ...
## $ gepu : num 0 0 0 0 0 7 0 0 0 0 ...
## $ hodi_s : num 1 7 0 0 0 0 0 0 7 13 ...
## $ hola : num 2 0 1 0.5 0 0.5 0 0 0 0 ...
## $ hype : num 0.5 0 0.5 0 0.5 ...
## $ irte : num 0 0 0.5 0 0 0.5 0 0 0 0 ...
## $ kocr : num 0 0 0 0 0 0 0 0 0 0 ...
## $ lasa : num 0 0 0 0 0.5 0.5 0.5 0 0 0 ...
## $ liap : num 0 0 0 0 0.5 0 28 0.5 5 1 ...
## $ loci_s : num 0 0 0.5 0 0 0 0 0 0 0 ...
## $ mal_s : num 0 0 2 0 0 0 0 0 0 0 ...
## $ mebu : num 0.5 0 0 0 0 ...
## $ mope : num 0 0 0 0 2 0 0.5 0 1 0 ...
## $ mosi : num 0 0 0 0 0 ...
## $ nepa : num 0 0 0 0 0 0 0 0.5 0.5 0.5 ...
## $ osce_s : num 0 4 1 0 0.2 ...
## $ osnu : num 0.5 0.5 0.5 0.5 0.5 0.5 0 0.5 0.5 0.5 ...
## $ phle_s : num 0 2 0 0 0 0 0 0 2 0.5 ...
## $ phpr : num 0 0 0 0 0 0 0 0 0 0 ...
## $ plla : num 0 0 0 0 0 0 0 0 0 0 ...
## $ poco : num 0 0 0 0 0 ...
## $ pogl : num 0 0 0 0 0 0 0.5 0 0 0.5 ...
## $ pogr : num 0 0 0.5 0 0 0 0 0 0 0 ...
## $ pomu : num 0.5 5 11 0.5 0.2 ...
## $ popr : num 0.5 0 0.5 0.5 0.5 0 0 0.5 0 0 ...
## $ povu : num 0.2 0.2 0.2 0.2 0 ...
## $ prav_s : num 0 1 0.5 0.5 1 4 0 0 0 0 ...
## $ prav_t : num 0 0 2 0 0 0 0 0 0 0 ...
## $ psme_s : num 0 7 0 0.5 0 ...
## $ psme_t : num 0 5 0 0.5 0 0 0 0 0 0.5 ...
## $ ptaq : num 6 0.5 6 0.5 0 0 0.5 0 0 0 ...
## $ pyco_t : num 0 0 0 0 0 2 0 0 0 0 ...
## $ pyfu_s : num 0 0 0.2 0.5 0 ...
## $ quga_s : num 0.5 0.5 8 3 0.5 0 0.5 0.5 0.5 0 ...
## $ quga_t : int 265 108 134 130 254 178 161 108 170 182 ...
## $ rhdi_s : num 31 0.5 13 24 22 27 27 78 48 39 ...
## $ rhpu_s : num 0.5 3 0 0.5 0 0.5 0.5 0 0 0.5 ...
## $ rhpu_t : num 0 1 0 0 0 0 0 0 0 0 ...
## $ roeg_s : num 2 3 7 0.2 0 ...
## $ rogy_s : num 0 0 0.5 0 2 3 0.5 0 2 4 ...
## $ ronu_s : num 0.5 0.5 1 0 0 0 0 0 0 0 ...
## $ ropi_s : num 0 0 0 0 0 0 0 0 0 0 ...
## $ rula_s : num 0 0 0 0 0 0.5 0 0.5 0 0.5 ...
## $ rupa_s : num 0 32 0 0 0 ...
## $ rupr_s : num 0 0 0 0 0.2 ...
## $ ruur_s : num 49 0.5 3 2 0 7 4 0 7 23 ...
## $ sacr : num 0 0 0 0 1 0.5 1 0.5 0.5 0.5 ...
## $ sado : num 0.5 0 0.5 0.5 0 0 0.5 0.5 0.5 0.5 ...
## $ syal_s : num 4 21 18 0.2 50 ...
## $ taof : num 0 0 0.5 0 0 ...
## $ tegr : num 0 0.5 0.5 0 0 0.5 0.5 0 4 4 ...

```

```
## $ thoc      : num  0 0 0 0 0 0 1 0.5 0 0 ...
## $ toar      : num  0.5 0.5 0.5 0.5 7 2 0.5 1 1 0.5 ...
## $ trla      : num  0.5 0 0 0.5 0 0 0.5 0.5 0.5 0.5 ...
## $ trov      : num  0 0 0 0 0 0 0.5 0 0 0 ...
## $ viam      : num  0.5 0.5 0.5 0.5 1 0.5 0 0 0 0 ...
## $ vinu      : num  0 0 0 0 0 0.5 0 0 0 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 58 species now.
str(oak_herb)
```

```
## 'data.frame': 39 obs. of 61 variables:
## $ stand      : chr  "stand03" "stand05" "stand06" "stand07" ...
## $ logged     : chr  "logged" "logged" "logged" "logged" ...
## $ heat_load  : num  0.93 0.815 0.876 0.905 0.87 0.941 0.69 0.809 0.69 0.831 ...
## $ acar       : num  0 1 0 0 0 0 0 0 0 0 ...
## $ acmi       : num  0 0 0.5 0 0 0 0 0 0 0 ...
## $ adbi       : num  0.5 0 3 0.5 0.5 ...
## $ agha       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ agse       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ agte       : num  1 0 0.5 0.5 0.5 0 0 0.5 0.5 0 ...
## $ apan       : num  0 0 0 0.5 0 0 0 0 0 0 ...
## $ brla       : num  0.5 0 0.5 0.5 0.5 0.5 0.5 0 0.5 0.5 ...
## $ brpu       : num  0.5 0 0 0 0.5 ...
## $ brri       : num  0 0 0 0 0 0 0 0.5 0 0 ...
## $ car        : num  0.5 0 0.5 0 0 0.5 0 0 0 0 ...
## $ cato       : num  0 0 0.2 0 0 ...
## $ cipa       : num  0 0 0 0 0 0 0.5 0 5 1 ...
## $ cyec       : num  0 0 0.5 0 0 0 0 0 0 0 ...
## $ daca       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ dacar      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ dagl       : num  2 0 1 0 0.2 ...
## $ drar       : num  0 0 0 0 0 0 0 0 10 7 ...
## $ elgl       : num  0.5 0 0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
## $ erla       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ erog       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ feca       : num  0 0 0 0 0 0 0 0 0 0 ...
## $ feoc       : num  0 0 0 0 0 ...
## $ feru       : num  0 0 0.5 0 0 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 ...
```

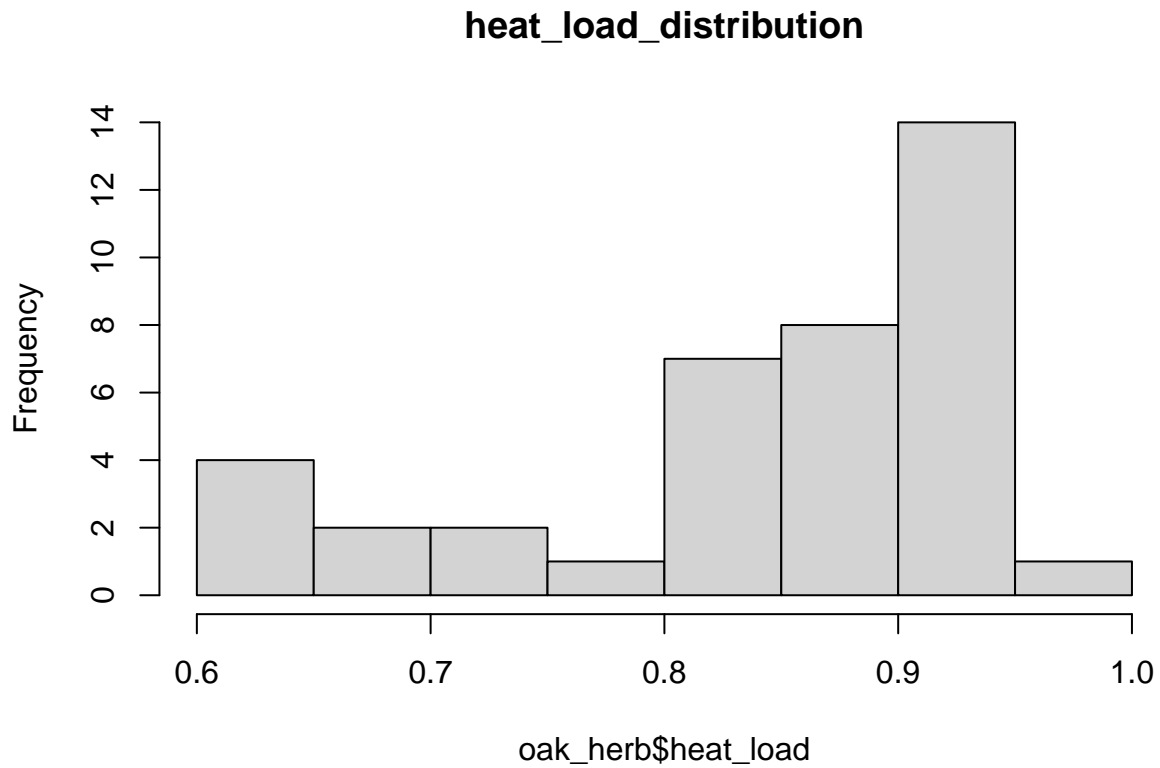
```
## $ gal      : num  2 0 0.5 0.5 7 0.5 3 0.5 3 2 ...
## $ geog     : num  0 0 0.5 0 0.2 ...
## $ gepu     : num  0 0 0 0 0 7 0 0 0 0 ...
## $ hola     : num  2 0 1 0.5 0 0.5 0 0 0 0 ...
## $ hype     : num  0.5 0 0.5 0 0.5 ...
## $ irte     : num  0 0 0.5 0 0 0.5 0 0 0 0 ...
## $ kocr     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ lasa     : num  0 0 0 0 0.5 0.5 0.5 0 0 0 ...
## $ liap     : num  0 0 0 0 0.5 0 28 0.5 5 1 ...
## $ mebu     : num  0.5 0 0 0 0 ...
## $ mope     : num  0 0 0 0 2 0 0.5 0 1 0 ...
## $ mosi     : num  0 0 0 0 0 ...
## $ nepa     : num  0 0 0 0 0 0 0 0.5 0.5 0.5 ...
## $ osnu     : num  0.5 0.5 0.5 0.5 0.5 0.5 0 0.5 0.5 0.5 ...
## $ phpr     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ plla     : num  0 0 0 0 0 0 0 0 0 0 ...
## $ poco     : num  0 0 0 0 0 ...
## $ pogl     : num  0 0 0 0 0 0 0.5 0 0 0.5 ...
## $ pogr     : num  0 0 0.5 0 0 0 0 0 0 0 ...
## $ pomu     : num  0.5 5 11 0.5 0.2 ...
## $ popr     : num  0.5 0 0.5 0.5 0.5 0 0 0.5 0 0 ...
## $ povu     : num  0.2 0.2 0.2 0.2 0 ...
## $ ptaq     : num  6 0.5 6 0.5 0 0 0.5 0 0 0 ...
## $ sacr     : num  0 0 0 0 1 0.5 1 0.5 0.5 0.5 ...
## $ sado     : num  0.5 0 0.5 0.5 0 0 0.5 0.5 0.5 0.5 ...
## $ taof     : num  0 0 0.5 0 0 ...
## $ tegr     : num  0 0.5 0.5 0 0 0.5 0.5 0 4 4 ...
## $ thoc     : num  0 0 0 0 0 0 1 0.5 0 0 ...
## $ toar     : num  0.5 0.5 0.5 0.5 7 2 0.5 1 1 0.5 ...
## $ trla     : num  0.5 0 0 0.5 0 0 0.5 0.5 0.5 0.5 ...
## $ trov     : num  0 0 0 0 0 0 0.5 0 0 0 ...
## $ viam     : num  0.5 0.5 0.5 0.5 1 0.5 0 0 0 0 ...
## $ vinu     : num  0 0 0 0 0 0.5 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] 20.7  8.7 31.9  8.2 23.6 17.2 57.6  9.6 50.8 29.1 19.1 15.4 13.7 67.2 60.4
## [16] 21.9  2.2 67.7  6.8 19.6 71.7 18.8 93.1 55.2 17.5 47.2 82.7 29.4  3.6 18.6
## [31] 66.3 22.9 88.8 24.7 65.1  4.1 17.1  8.8 13.2
```

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



**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.

**2**