

Module “4”: Lab 1

Ellen Bledsoe

2023-05-01

Species Richness and Diversity

Today, we will be working with community data to calculate species richness, plot species accumulation curves, and calculate two different of species diversity indices.

Set-Up

For today’s lab and assignment, we will need to load a new package called **vegan**. I’ve already installed the package on Posit Cloud, so we only need to load it into our work space. We are also going to load the **tidyverse**.

```
library(vegan)
library(tidyverse)
```

The **vegan** package is one of the most popular packages available for working with community-level data. It includes functions for many multivariate analyses commonly used in the analysis of ecological communities. We will be using it today for its species accumulation curve functions.

Species Accumulation Curves

For this example, we are going to use a dataset of fish abundances from the Doubs River in France. Each column represents a species, and each row is a survey.

```
fish <- read_csv("../data_raw/Doubs_fish.csv")
```

```
## Rows: 30 Columns: 27
## -- Column specification -----
## Delimiter: ","
## dbf (27): CHA, TRU, VAI, LOC, OMB, BLA, HOT, TOX, VAN, CHE, BAR, SPI, GOU, B...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
head(fish)
```

```
## # A tibble: 6 x 27
##   CHA  TRU  VAI  LOC  OMB  BLA  HOT  TOX  VAN  CHE  BAR  SPI  GOU
```

```
##      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1      0      3      0      0      0      0      0      0      0      0      0      0
## 2      0      5      4      3      0      0      0      0      0      0      0      0
## 3      0      5      5      5      0      0      0      0      0      0      0      0
## 4      0      4      5      5      0      0      0      0      0      1      0      1
## 5      0      2      3      2      0      0      0      0      5      2      0      2
## 6      0      3      4      5      0      0      0      0      1      2      0      1
## # ... with 14 more variables: BRO <dbl>, PER <dbl>, BOU <dbl>, PSO <dbl>,
## #   ROT <dbl>, CAR <dbl>, TAN <dbl>, BCO <dbl>, PCH <dbl>, GRE <dbl>,
## #   GAR <dbl>, BBO <dbl>, ABL <dbl>, ANG <dbl>
```

First, we need to run our survey by species matrix through the `specaccum()` function to prepare the data to be put into a species accumulation curve model.

```
fish_sac <- specaccum(fish)
```

Now, we want to put the object we just created, `fish_sac`, into the `fitspecaccum()` function. We need to specify the model we want as “asympt”.

```
fish_fit_sac <- fitspecaccum(fish_sac, model = "asympt")
```

Now that we’ve fit the model, we can get out the estimated asymptote of the species accumulation curve and plot the predicted curve.

`coef()` gives us the estimated asymptote of the species accumulation curve.

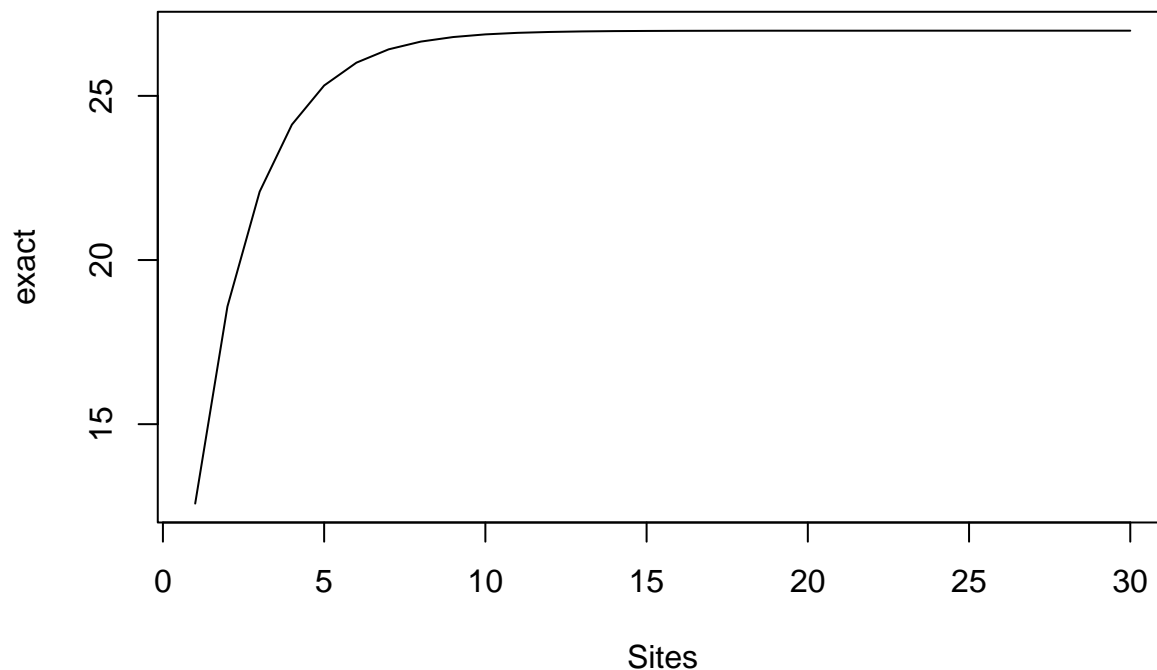
```
coef(fish_fit_sac)
```

```
##      Asym      R0      lrc
## 26.9858415 2.2980754 -0.6184861
```

The asymptote is coming out right around 27, meaning that there are likely about 27 species in our community. If we look at the original data, we see that the data frame has 27 columns (AKA species), so we’ve probably caught all the species in the community.

Let’s plot the data to see what it looks like.

```
plot(fish_fit_sac)
```



Looks like we hit the asymptote well before we stopped sampling, which is a great sign.

Species Richness and Species Diversity

Let's run through how we can calculate the richness, and Shannon and Simpson Diversity Indices for a community.

```
# Don't worry about understanding this code!
# I'm making the frequencies of each fish species and making them into a dataframe
# The rownames are turned into a new column called "species"
fish_abund <- as_tibble(fish_sac$freq, rownames = "species")
head(fish_abund)
```

```
## # A tibble: 6 x 2
##   species value
##   <chr>   <dbl>
## 1 CHA      8
## 2 TRU     17
## 3 VAI     20
## 4 LOC     24
## 5 OMB      8
## 6 BLA      8
```

Species Richness

Fortunately, calculating species richness is fairly straightforward. We already know that the richness of the fish community is 27, but let's calculate that with code. The `nrow()` function will count how many rows we have for us, which, given the way our data is being presented, will be the species richness of the community.

```
nrow(fish_abund)
```

```
## [1] 27
```

Shannon Index

We'll start with the Shannon Index for the Doubs fish community.

Our first step is to sum up the total number of individuals in the community. We need this value to create the proportion for each species.

We will use the sum function and specify that we want to add up the numbers in the “value” column by using the dollar sign operator.

```
N <- sum(fish_abund$value)
N
```

```
## [1] 375
```

Now we want to calculate the proportion of the population made up by each species. We will store these values by adding them into a new column in the **abund** data frame.

We do this by using the dollar sign operator again. On the left side of the assignment arrow, we create a new column. On the right side, we tell R what the new column should contain. In this case, we want to create the proportion for each species—the number of individuals per species divided by the total number of individuals in the population. We will call this new column **prop**.

```
fish_abund$prop <- fish_abund$value / N
head(fish_abund)
```

```
## # A tibble: 6 x 3
##   species value  prop
##   <chr>   <dbl> <dbl>
## 1 CHA      8 0.0213
## 2 TRU     17 0.0453
## 3 VAI     20 0.0533
## 4 LOC     24 0.064
## 5 OMB      8 0.0213
## 6 BLA      8 0.0213
```

Note that we now have a new column with all the proportions calculated. We want to continue this by taking the natural log of that proportion. We use the **log()** function to do that.

```
fish_abund$ln_prop <- log(fish_abund$prop)
head(fish_abund)
```

```
## # A tibble: 6 x 4
##   species value  prop ln_prop
##   <chr>   <dbl> <dbl> <dbl>
## 1 CHA      8 0.0213 -3.85
## 2 TRU     17 0.0453 -3.09
## 3 VAI     20 0.0533 -2.93
## 4 LOC     24 0.064 -2.75
## 5 OMB      8 0.0213 -3.85
## 6 BLA      8 0.0213 -3.85
```

Now we need to multiply the values in the two columns we just created.

```
fish_abund$prop_x_lnprop <- fish_abund$prop * fish_abund$ln_prop
head(fish_abund)
```

```
## # A tibble: 6 x 5
##   species value   prop ln_prop prop_x_lnprop
##   <chr>   <dbl> <dbl>   <dbl>         <dbl>
## 1 CHA      8 0.0213 -3.85      -0.0821
## 2 TRU     17 0.0453 -3.09      -0.140
## 3 VAI     20 0.0533 -2.93      -0.156
## 4 LOC     24 0.064  -2.75      -0.176
## 5 OMB      8 0.0213 -3.85      -0.0821
## 6 BLA      8 0.0213 -3.85      -0.0821
```

Finally, we can sum these values and multiple by -1 to calculate the Shannon Index (H) for this community.

```
H = sum(fish_abund$prop_x_lnprop) * -1
H
```

```
## [1] 3.239048
```

You will want to use a similar strategy to calculate the Simpson Index, as well.

tidyverse

What would this look like in the tidyverse? If you're curious, here is the code below.

```
H_tidy <- as_tibble(fish_sac$freq, rownames = "species") %>%
  mutate(prop = value / N, # using the N already created
         ln_prop = log(prop),
         prop_x_lnprop = prop * ln_prop) %>%
  summarise(H = sum(prop_x_lnprop) * -1)
H_tidy
```

```
## # A tibble: 1 x 1
##       H
##   <dbl>
## 1  3.24
```

The power of the tidyverse really shines when you have multiple communities for which you want to calculate values, such as the Shannon Index.

Take a look at this example:

```
algar <- read.csv("../data_raw/algar_data.csv")
head(algar)
```

```
##   Deployment.Location.ID Alces.alces Canis.latrans Canis.lupus
## 1                      Algar08      5             0           1
## 2                      Algar27      1             3           4
```

```
## 3      Algar29      7      0      1
## 4      Algar31      3      0      6
## 5      Algar32      1      0      0
## 6      Algar34      1      0      0
##   Cervus.canadensis Grus.canadensis Lepus.americanus Lynx.canadensis
## 1      0      0      1      0
## 2      1      0      0      4
## 3      0      13     0      1
## 4      0      13     1      0
## 5      0      3      0      1
## 6      0      0      0      4
##   Martes.americana Odocoileus.virginianus Rangifer.tarandus
## 1      0      2      1
## 2      0      1      0
## 3      0      24     0
## 4      4      1      15
## 5      0      1      2
## 6      0      3      0
##   Tamiasciurus.hudsonicus Ursus.amERICANUS
## 1      0      0
## 2      0      0
## 3      0      7
## 4      0      2
## 5      0      0
## 6      0      0
```

This dataset has 14 different species observed over 20 different communities. It would be very annoying to have to calculate 20 Shannon Indices using base R, the way we did above!

With `tidyverse`, we can calculate all of them at the same time.

First, we convert the data from a wide format (site x species) to a long format and remove any zeros (they cause NAs).

```
algar_long <- algar %>%
  rownames_to_column("site") %>%
  pivot_longer(Alces.alces:Ursus.amERICANUS,
    names_to = "species", values_to = "abundance") %>%
  filter(abundance > 0)
head(algar_long)
```

```
## # A tibble: 6 x 4
##   site Deployment.Location.ID species      abundance
##   <chr> <chr>                <chr>          <int>
## 1 1     Algar08              Alces.alces      5
## 2 1     Algar08              Canis.lupus      1
## 3 1     Algar08              Lepus.amERICANUS 1
## 4 1     Algar08              Odocoileus.virginianus 2
## 5 1     Algar08              Rangifer.tarandus 1
## 6 2     Algar27              Alces.alces      1
```

From there, we create groups for each site then proceed with our calculations.

```
H_indices <- algar_long %>%
  group_by(Deployment.Location.ID) %>%           # group the data by site
  mutate(N = sum(abundance),                    # calculate the total abundance for each site
         prop = abundance / N,                  # calculate the proportion of each species per site
         ln_prop = log(prop),
         prop_ln_prop = prop * ln_prop) %>%
  summarise(H = sum(prop_ln_prop) * -1)
H_indices
```

```
## # A tibble: 20 x 2
##   Deployment.Location.ID      H
##   <chr>                  <dbl>
## 1 Algar08                1.36
## 2 Algar27                1.61
## 3 Algar29                1.39
## 4 Algar31                1.70
## 5 Algar32                1.49
## 6 Algar34                0.974
## 7 Algar35                1.85
## 8 Algar37                0.404
## 9 Algar38                0.712
## 10 Algar39              1.49
## 11 Algar61              0.333
## 12 Algar62              0.806
## 13 Algar63              0.726
## 14 Algar64              0.760
## 15 Algar65              0.325
## 16 Algar66              1.05
## 17 Algar67              0.914
## 18 Algar68              1.01
## 19 Algar69              0.958
## 20 Algar70              1.63
```

Fun! And much faster :)