BIO 423 - Lab 5

Benjamin Blonder

Spring 2019

Learning outcomes

R goals:

- Make and visualize linear regressions
- Transform data
- Join dataframes on primary keys
- Filter, group, and arrange dataframes using `dplyr`

Content goals:

- Computer metrics and latitudinal drivers of alpha diversity
- Compute major metrics of macroecology: species-abundance distributions, species-area relationships

The Gentry dataset

We will begin by exploring biodiversity trends in Alwyn Gentry's forest transect dataset:

`http://www.mobot.org/MOBOT/Research/gentry/transect.shtml`

Gentry was interested in the drivers of plant biodiversity at regional and continental scales. Over more than two decades he personally set up and inventoried all plant species in more than 200 forested sites on six continents, often in very remote areas. Each site includes collections of all plants with stem diameters equal to or exceeding 2.5 cm diameter at breast height (dbh) along ten 2 x 50 m transects. Each site is 0.1 hectares in size. Gentry's foundational work was cut short when his small plane crashed in western Ecuador in 1993.

Gentry's data have been curated by the Missouri Botanical Garden and made available through the BIEN database to us. They need a bit of pre-processing to be useful.

```
library(data.table) # install this package if needed

data_gentry = read.csv("gentry_transects.csv")

str(data_gentry)
```

```
## 'data.frame':    32638 obs. of  18 variables:
##  $ X                          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ plot_name                  : Factor w/ 165 levels "ACHUPALL","ALLPAHUA",..: 115 115 115 115 115 1
##  $ subplot                    : Factor w/ 13 levels "`","0","1","10",..: 3 3 3 3 3 3 3 4 4 4 ...
##  $ elevation_m                : int  1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...
##  $ plot_area_ha               : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ sampling_protocol          : Factor w/ 1 level "0.1 ha  transect, stems >= 2.5 cm dbh": 1 1 1 1 1
##  $ recorded_by                : Factor w/ 4 levels "Alwyn H. Gentry",..: NA 1 1 1 1 1 1 NA NA 1 ...
##  $ scrubbed_species_binomial  : Factor w/ 3674 levels "Abarema barbouriana",..: 740 854 1246 2218 278
##  $ individual_count           : int  1 2 4 1 1 1 3 3 1 10 ...
##  $ latitude                   : num  -24.6 -24.6 NA -24.6 -24.6 ...
##  $ longitude                  : num  -64.7 -64.7 NA -64.7 -64.7 -64.7 -64.7 -64.7 -64.7 NA ...
##  $ date_collected             : logi  NA NA NA NA NA NA ...
```

```
##  $ datasource                 : Factor w/ 1 level "SALVIAS": 1 1 1 1 1 1 1 1 1 ...
##  $ dataset                    : Factor w/ 1 level "Gentry Transect Dataset": 1 1 1 1 1 1 1 1 1 ...
##  $ dataowner                  : Factor w/ 1 level "James S. MIller": 1 1 1 1 1 1 1 1 1 ...
##  $ custodial_institution_codes: logi  NA NA NA NA NA NA ...
##  $ collection_code            : logi  NA NA NA NA NA NA ...
##  $ datasource_id              : int  14 14 14 14 14 14 14 14 14 14 ...
```

```
# how many plots do we have to work with?
length(unique(data_gentry$plot_name))
```

```
## [1] 165
```

```
# note that this is a subset of the full Gentry dataset most useful for teaching purposes.
# The MOBOT link above includes the entire dataset.
```

Some of the species are missing names due to being un-identifiable in the field:

```
table(is.na(data_gentry$scrubbed_species_binomial))
```

```
##
## FALSE  TRUE
## 18679 13959
```

To fix this, we need to give these species names. However each species needs its own unique name, as we are not sure if the missing species name in one case is the same name as in other cases.

```
# first, convert the names from factors to characters so that we can add new levels
data_gentry$scrubbed_species_binomial = as.character(data_gentry$scrubbed_species_binomial)

# which is a useful command you can use to find the indices of items in a vector
index_sp_no_name = which(is.na(data_gentry$scrubbed_species_binomial))
data_gentry$scrubbed_species_binomial[index_sp_no_name] =
    paste("Species",1:length(index_sp_no_name))
```

Some observations also have missing counts. We will fill these assuming they have a count of one.

```
# how many rows have missing counts?
table(is.na(data_gentry$individual_count))
```

```
##
## FALSE  TRUE
## 32637     1
```

```
# set observations with missing counts to have a count of 1
data_gentry$individual_count[which(is.na(data_gentry$individual_count))] = 1
```

Note as well that while each row in the dataframe represents an observation of one to several individuals of a species, a species may occur in more than one row of the dataframe. This situation corresponds to Gentry re-encountering a cluster of the same species in multiple locations in the plot. We will rewrite these values to include only a single row for each species, and to include a total count for all individuals. To do so, we first repeat each species names as many times as the number of individual counts recorded, then create a `table` to summarize the total counts across each species. This code is in turn wrapped in a `by` call that repeats this analysis within each data subset corresponding to a unique `plot_name`. To do this we use an anonymous function passed to the `FUN` argument of `by`. This is a bit complicated - don't worry if you are not interested in the details.

```
gentry_counts_list = by(data_gentry, data_gentry$plot_name, function(x) {

  # repeat the data, then convert to factors for use in the table command
  # then count up all the entries
```

```
  result_table = table(factor(rep(x$scrubbed_species_binomial, x$individual_count)))

  # create a data frame summarizing this subset
  df_table = data.frame(plot_name=x$plot_name[1], # keep the plot_name as primary key
                        latitude=x$latitude[1],
                        longitude=x$longitude[1],
                        species=names(result_table), # pull table names
                        abundance=as.numeric(result_table)) # pull table counts
  return(df_table)
})
# convert results to a data frame
gentry_counts = rbindlist(gentry_counts_list)

str(gentry_counts)
```

```
## Classes 'data.table' and 'data.frame':   22609 obs. of  5 variables:
##  $ plot_name: Factor w/ 165 levels "ACHUPALL","ALLPAHUA",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ latitude : num  -3.45 -3.45 -3.45 -3.45 -3.45 -3.45 -3.45 -3.45 -3.45 -3.45 ...
##  $ longitude: num  -78.4 -78.4 -78.4 -78.4 -78.4 ...
##  $ species  : Factor w/ 17633 levels "Alchornea pearcei",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ abundance: num  11 4 2 1 2 1 2 4 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

We now have a dataframe, `gentry_counts`, which is ready to use for analysis. The data includes information on the plot name (`plot_name`), the species (`species`), and the number of individuals of that species (`abundance`).

We can also separately extract the unique table of metadata for each plot:

```
# pick site-level columns, and retain only unique combinations
# the na.omit is needed to avoid some blank entries from the database structuring
gentry_metadata = na.omit(unique(data_gentry[,c("plot_name","latitude","longitude","elevation_m")]))

str(gentry_metadata)
```

```
## 'data.frame':    165 obs. of  4 variables:
##  $ plot_name  : Factor w/ 165 levels "ACHUPALL","ALLPAHUA",..: 115 134 22 45 56 57 64 74 94 95 ...
##  $ latitude   : num  -24.6 -24.7 -15 -14.6 -18.8 ...
##  $ longitude  : num  -64.7 -64.5 -68.5 -68.5 -62.3 ...
##  $ elevation_m: int  1000 1300 1540 1000 350 350 280 1540 280 370 ...
##  - attr(*, "na.action")= 'omit' Named int  2 5 7 9 11 13 15 17 19 21 ...
##   ..- attr(*, "names")= chr  "3" "182" "364" "589" ...
```

The metadata includes the `latitude`, `longitude`, and `elevation_m` for each plot.

We are ready to start some real analyses after this data cleaning!

Metrics of alpha diversity

We can calculate several metrics of alpha diversity within each site. Richness, is the unique number of species present. Abundance is the total number of individuals; Shannon diversity quantifies how the individuals are distributed among species.

A first key question we will address focuses on how alpha diversity varies across environments. To answer this question we first need to summarize the data.

```
summary_richness_list = by(gentry_counts, gentry_counts$plot_name, function(x) {
  data.frame(plot_name=x$plot_name[1],
             species_richness=length(x$species))
  })

summary_richness = rbindlist(summary_richness_list)

# count the mean richness within each plot
mean(summary_richness$species_richness)
```

```
## [1] 137.0242
```

```
# alternatively, using the tidyr functionality...
library(dplyr) # install if needed
```

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

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

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

```
# the below functionality uses code from the R 'tidyverse' -
# a set of advanced functions to manipulate data.
# these functions use 'tibbles' instead of dataframes
# and allow output to be 'piped' between functions sequentially
# they also allow easy grouping and summarizing, as you can see below.
summary_richness_tidyr = gentry_counts %>%
                           group_by(plot_name) %>%
                           summarize(species_richness=n_distinct(species))
summary_abundance_tidyr = gentry_counts %>%
                           group_by(plot_name) %>%
                           summarize(total_abundance=sum(abundance))

print(summary_richness_tidyr)
```

```
## # A tibble: 165 x 2
##    plot_name species_richness
##    <fct>                <int>
##  1 ACHUPALL               198
##  2 ALLPAHUA               283
##  3 ALTERDOC                87
##  4 ALTODEMI               150
##  5 ALTOSAPA               175
##  6 AMOTAPE                 75
##  7 ANCHICAY               238
##  8 ANTADO                 273
##  9 ARARACUA               346
## 10 ARCATING               117
```

```
## # ... with 155 more rows
```

```
print(summary_richness_tidyr)
```

```
## # A tibble: 165 x 2
##    plot_name species_richness
##    <fct>                <int>
##  1 ACHUPALL               198
##  2 ALLPAHUA               283
##  3 ALTERDOC                87
##  4 ALTODEMI               150
##  5 ALTOSAPA               175
##  6 AMOTAPE                 75
##  7 ANCHICAY               238
##  8 ANTADO                 273
##  9 ARARACUA               346
## 10 ARCATING               117
## # ... with 155 more rows
```

We can also calculate more advanced metrics of diversity like the Shannon richness,

$$H = -\sum_{i}^{n} p_i \log p_i$$

where

$$p_i = A_i / \sum_{i}^{n} A_i$$

and A_i is the abundance of species i. Unlike richness it accounts for abundance and evenness. The index is maximized if all species have the same abundance.

```
shannons_H = function(A)
{
  p = A / sum(A)

  H = -1*sum(p * log(p))

  return(H)
}

summary_H_tidyr = gentry_counts %>%
                            group_by(plot_name) %>%
                            summarize(H=shannons_H(abundance))

print(summary_H_tidyr)
```

```
## # A tibble: 165 x 2
##    plot_name     H
##    <fct>     <dbl>
##  1 ACHUPALL   5.00
##  2 ALLPAHUA   5.45
##  3 ALTERDOC   4.26
##  4 ALTODEMI   4.84
##  5 ALTOSAPA   4.90
##  6 AMOTAPE    3.70
##  7 ANCHICAY   5.21
##  8 ANTADO     5.46
```

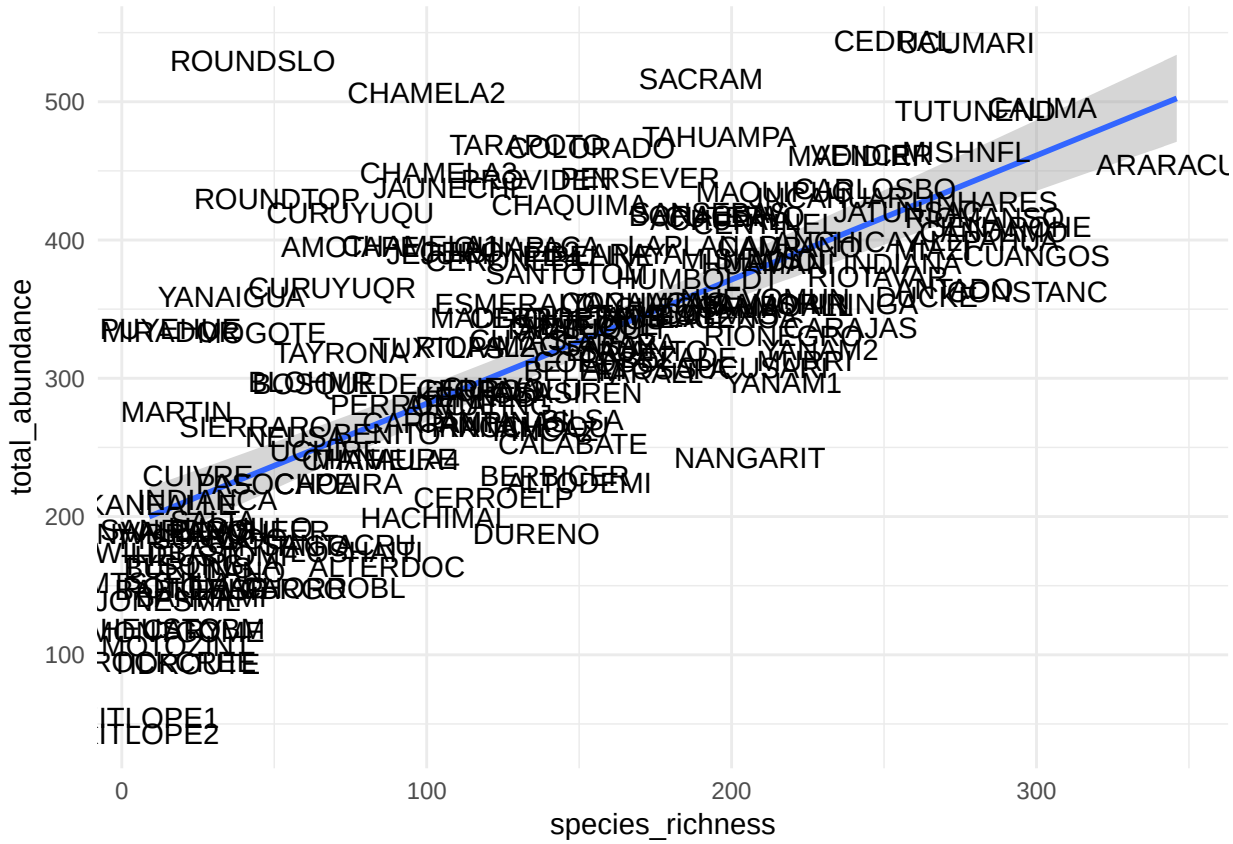
```
##  9 ARARACUA   5.69
## 10 ARCATING   4.45
## # ... with 155 more rows
```

Our next challenge is to merge these different summaries to each other and to the metadata. To do so we need to do a database 'inner-join' where a primary key is used to cross-reference values in one data frame against those in another. An inner join means that only items found in both tables will be retained. You can read about other types of joins (left, right, outer) in the help for the `join` function. Here we will join the richness and the abundance datasets using `plot_name` as a primary key.

```
df_joined = inner_join(summary_richness_tidyr, summary_abundance_tidyr, by="plot_name")

# note that we now have paired values of richness and abundance for each plot!
print(df_joined)
```

```
## # A tibble: 165 x 3
##    plot_name species_richness total_abundance
##    <fct>                <int>           <dbl>
##  1 ACHUPALL               198             415
##  2 ALLPAHUA               283             400
##  3 ALTERDOC                87             164
##  4 ALTODEMI               150             225
##  5 ALTOSAPA               175             307
##  6 AMOTAPE                 75             395
##  7 ANCHICAY               238             399
##  8 ANTADO                 273             365
##  9 ARARACUA               346             455
## 10 ARCATING               117             281
## # ... with 155 more rows
```

```
library(ggplot2)

# now we can plot the data - note we are also adding a regression line to the plot, with geom_smooth.
ggplot(df_joined,aes(x=species_richness,y=total_abundance)) +
  geom_smooth(method="lm") + # the new line of code!
  theme_minimal() +
  geom_text(aes(label=plot_name))
```




```
# we can also find the equation for the smoothing line
model_linear = lm(total_abundance~species_richness,data=df_joined)
summary(model_linear)
```

```
##
## Call:
## lm(formula = total_abundance ~ species_richness, data = df_joined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -157.00  -48.66  -12.31   42.59  299.49
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      191.91757   11.35856   16.90   <2e-16 ***
## species_richness   0.89746    0.07059   12.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76.5 on 163 degrees of freedom
## Multiple R-squared:  0.4979, Adjusted R-squared:  0.4948
## F-statistic: 161.7 on 1 and 163 DF,  p-value: < 2.2e-16
```

```
# the relationship is positive and is significant (p<2.2e-16).
# with an R2 value of 0.49 and with a slope of 0.89.

# we can also predict values from the model
# here is a prediction of abundance in a site with species richness of 100
```

```r
predict(model_linear,data.frame(species_richness=100))
```

```
##        1
## 281.6633
```

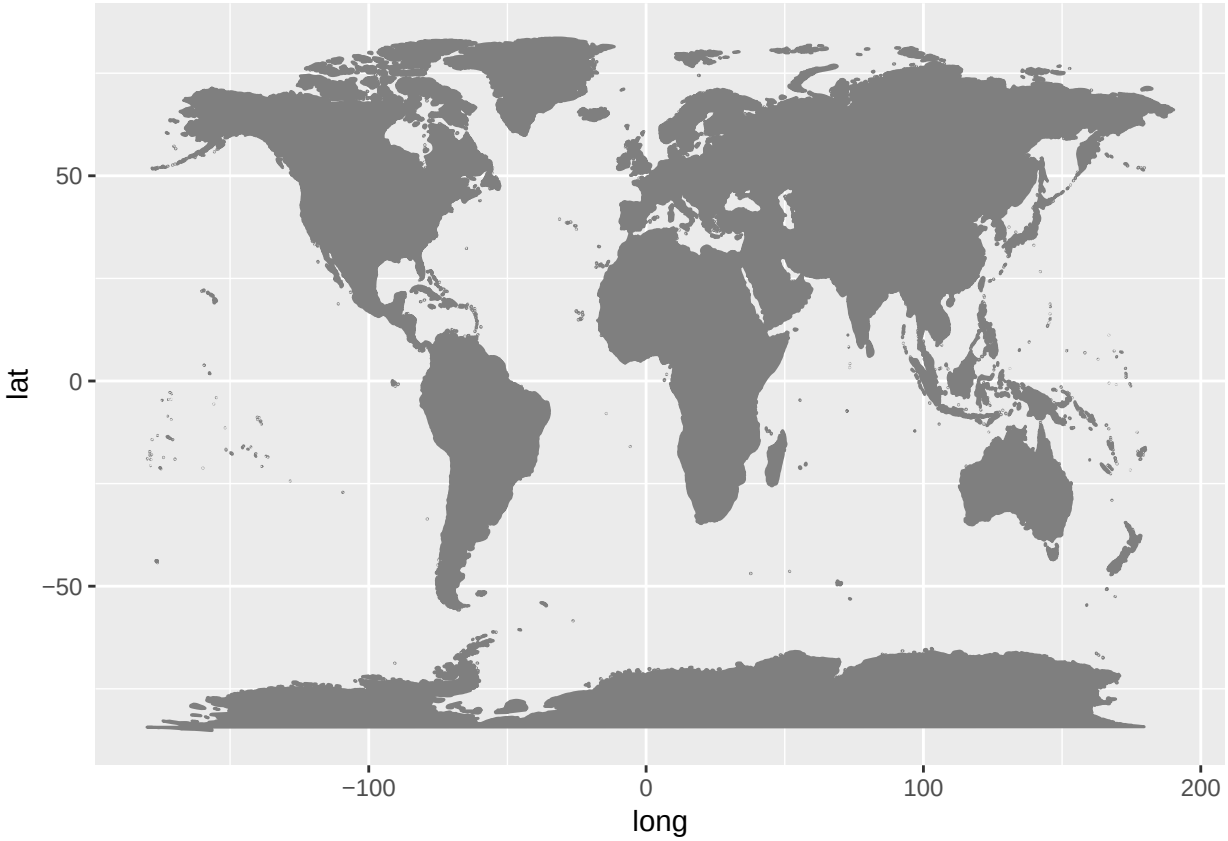
Questions

1. Join in the `gentry_metadata`. What is the minimum and maximum richness in the Gentry dataset? (Hint: you can use `which.min` to get the index of a row satisfying a criterion).
2. For the plots listed above, where are they located? Look up their latitude/longitude in the metadata, then determine the country/region using Google Maps.
3. How are values of species richness related to values of Shannon's H? Make a `ggplot` graph showing the relation across all plots. Do you think the H index is useful?
4. Make a plot of how alpha diversity varies with absolute latitude (hint: use `abs(latitude)` as the x-axis variable). How many more or fewer species do we see for each degree of latitude? (hint: this is the slope of the regression). Report the regression statistics.

Optional question for graduate students

- Does the latitudinal gradient of alpha diversity vary between the Northern and Southern hemispheres? You can get at this by creating a hemisphere column in the data based on whether latitude is greater than zero or not. Then plot, adding a `col=` or `group=` to the plot `aes`. You can run an ANCOVA or a linear model to test for a significant hemisphere effect.
- Make a map of species richness, coloring plots by their richness and locating each according to their latitude and longitude. You can draw a base map using

```r
mapWorld <- borders("world", colour="gray50", fill="gray50")
ggplot() + mapWorld
```

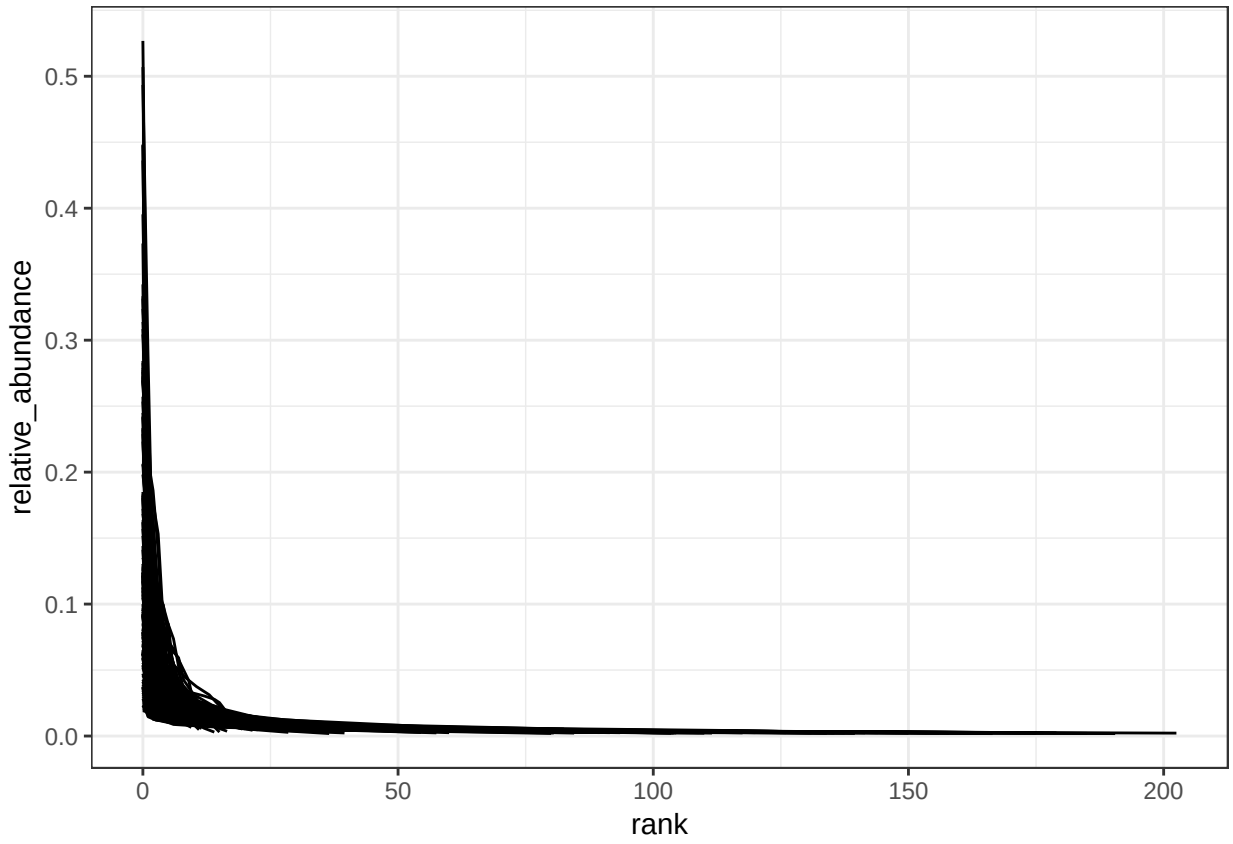


Then add extra layers to show the plots. You may need to install a few extra packages to get this to work. Where was Gentry most active in sampling? How might this bias the latitudinal trends seen above? * We have climate data from WorldClim from a previous week lab. Use the Gentry metadata to extract values of precipitation for each plot. Find out how alpha diversity changes with precipitation. Fit a regression model to summarize the results.

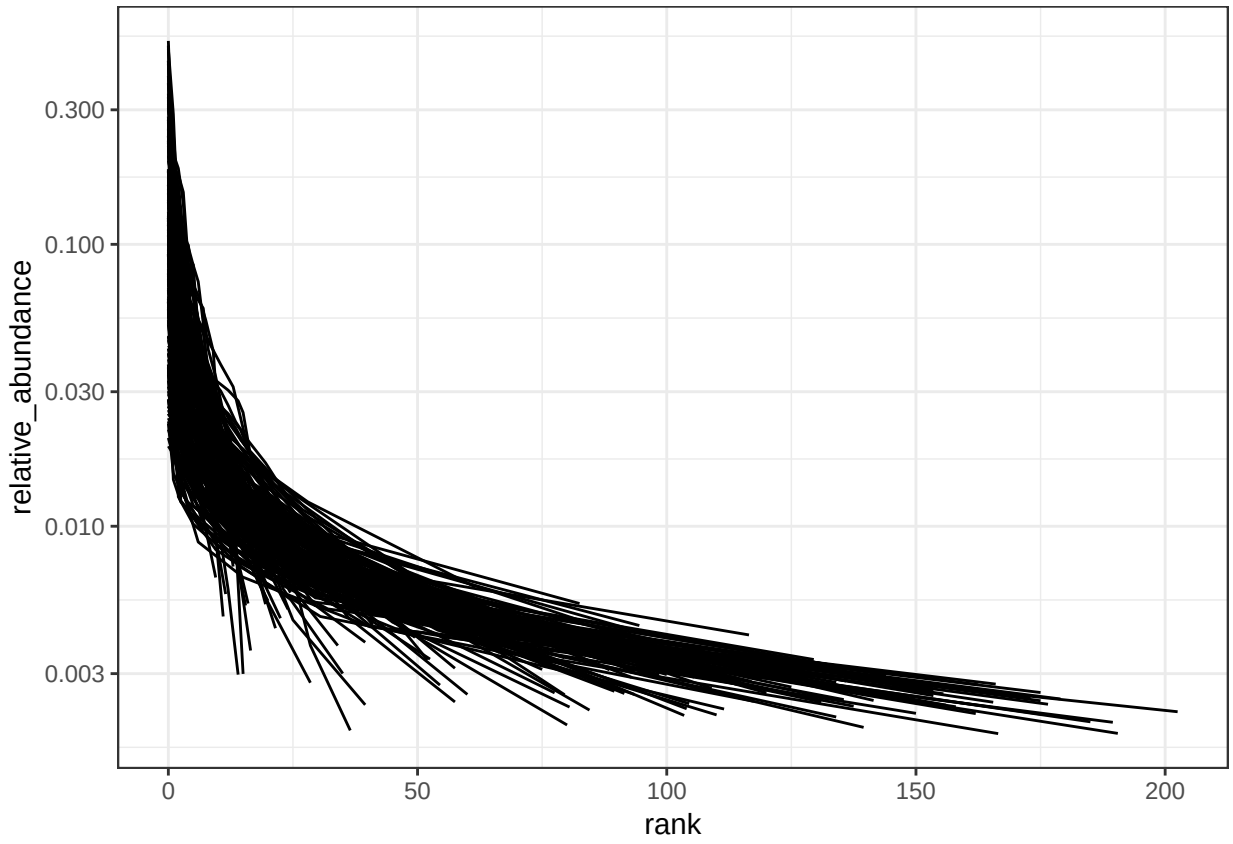
Species abundance distributions (SADs)

A second key pattern in macroecology is the distribution of individuals into species. Let's see what the Gentry data show us. First we will create a rank-abundance diagram. This is a plot of relative abundance vs. a species' rank (i.e. a numeric ordering of whether it is the 1st, 2nd, ... nth most common species). This is easy to do with the `dplyr` package - we simply group data by `plot_name`, then define ranks and relative abundances, and re-order within each group by rank:

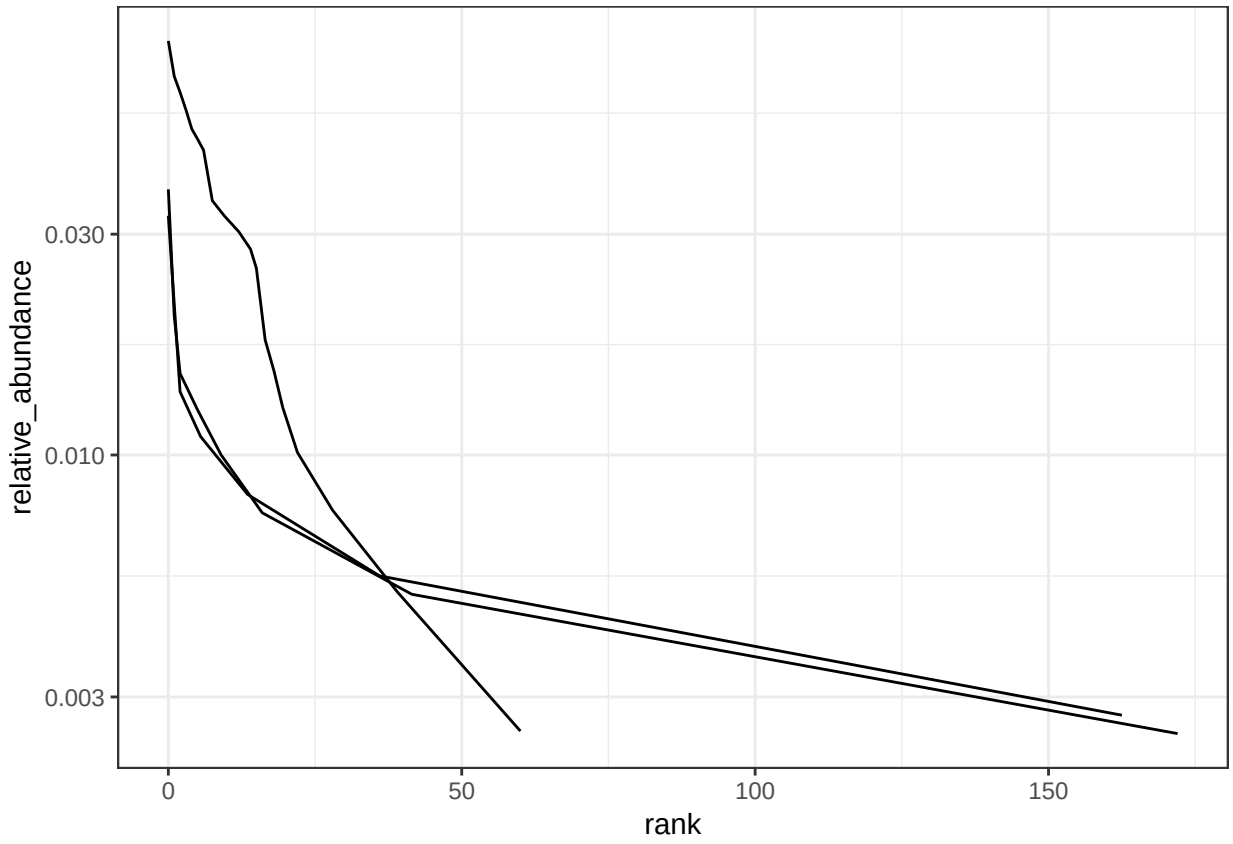
```
gentry_rank = gentry_counts %>%
                        group_by(plot_name) %>%
                        mutate(relative_abundance=abundance/sum(abundance),
                                rank=rank(abundance)) %>% # new columns
                        mutate(rank=max(rank) - rank) %>% # reverse order ranks for easy plotting
                        arrange(rank) # reorder by rank within each dataframe for easy plotting

# visualize the data, one line per plot
ggplot(gentry_rank,aes(x=rank,y=relative_abundance,group=plot_name)) +
  geom_line() +
  theme_bw()
```




```
# the above graph was not so easy to see, so let's visualize with the y-axis log-transformed
ggplot(gentry_rank,aes(x=rank,y=relative_abundance,group=plot_name)) +
  geom_line() +
  theme_bw() +
  scale_y_log10()
```




```
# plot only a subset of diagrams using 'filter' and the %in% operator
gentry_rank_ss = gentry_rank %>% filter(plot_name %in% c("AMOTAPE","ANTADO","ALLPAHUA"))
ggplot(gentry_rank_ss,aes(x=rank,y=relative_abundance,group=plot_name)) +
  geom_line() +
  theme_bw() +
  scale_y_log10()
```



Questions

5. Subset the `gentry_rank` data for just the `POTOMAC` and the `YANAM2` plots. (hint, try using row indexing as `gentry_rank$plot_name %in% c("YANAM2","POTOMAC")` or `%>% filter(plot_name %in% c("YANAM2","POTOMAC")) %>% arrange(plot_name,rank)`) Report the most common species in each plot. (hint, you can either use `filter(rank==min(rank)` after another `group_by(plot_name` or you can visually inspect the data subset to look for the minimum-rank species.
6. Make a plot of the rank-abundance diagram for these two sites. Show it in your results, and explain how to biologically interpret the x- and the y-intercept of each line.
7. Imagine that the `POTOMAC` site was invaded by a large number of species, each of which currently has low abundance. Explain in 1-2 sentences how would this qualitatively change the shape of the rank abundance diagram.

Optional questions for graduate students

- What statistical distribution best fits these species abundance distributions? A few leading candidates are the log-series, the log-normal, and the zero-sum multinomial (from neutral theory) (see McGill et al (2007) (PDF in this folder). You can determine which of these is the best fit to the data using a range of approaches. The `sads` package has a range of `fit*` functions you can use. Compare the most likely models using `logLik` or (better) `AIC`, which penalizes the likelihood based on the number of free parameters in each model. Which distribution is most commonly the best fit across all the Gentry plots? You can also fit a smaller set of distributions using the `fitdistr` function in the `MASS` package.
- Another classic macroecological pattern is the decay of community similarity with geographic distance. Construct a geographic distance matrix from the metadata using `dist`; construct a community matrix

(sites x species) from the Gentry data using the `gather` and `spread` functions in `dplyr`. Entries of this matrix should be 0 if the species is absent and equal the abundance at the site if the species is present. Then use the `vegdist` function in the `vegan` package to get a community dissimilarity matrix; then subtract it from one to get a similarity matrix instead. Plot values of the geographic distance matrix against the values of the community similarity matrix (hint: you may need to use `arrange` to ensure entries in both matrices correspond to the same site combinations). Estimate the rate of decay of community similarity with distance. Does the model get best fit by a linear, exponential, or other type of fit?

Species-area relationships

We will last investigate how species richness (S) changes with area A. The classic expectation is

$$S = kA^z$$

where k is a constant and z is an exponent that typically takes value $z \approx 0.25$. This relationship can be rewritten by log-transforming both sides as

$$log(S) = log(k) + z \times log(A)$$

. That is, log-species richness should increase linearly with log-area with a slope of z

We will examine evidence for this pattern using a different dataset, because the Gentry plots all have the same area (0.1 hectares). We will use data for long-horned beetles of south Florida, as published in Browne & Peck (1996). They surveyed beetle diversity on Florida and several outlying islands, including the Keys and the Dry Tortugas.

```
data_brownepeck = read.csv('browne_peck_1996.csv')

str(data_brownepeck)
```

```
## 'data.frame':    13 obs. of  4 variables:
##  $ Land.type: Factor w/ 5 levels "Island","Lower Keys",..: 3 4 5 5 5 5 2 2 2 2 ...
##  $ Name     : Factor w/ 13 levels "Big Pine Key",..: 6 12 7 10 5 8 11 1 2 3 ...
##  $ A.km2    : num  149913 5080 55.1 4.3 3.7 ...
##  $ S.num    : int  213 91 44 16 12 15 16 24 126 8 ...
```

```
# calculate log-axes
data_brownepeck$logS = log(data_brownepeck$S.num)
data_brownepeck$logA = log(data_brownepeck$A.km2)
```

Questions

8. Make a plot of the relationship between y=`logS` and x=`logA`. Label each point by its island name. (hint: `geom_text(aes(label=Name))`). Also draw a regression line.
9. Estimate the slope of the line using a linear model (hint: run `lm` as in the previous example, and report the coefficient value. Do your data support the $z = 0.25$ prediction? (You can use `confint` to be precise)
10. Which island has the most species relative to its size based on the regression?

What to hand in

- A single Word Document including:

– written answers (1-2 sentences) and figures for each question above
– A copy of your R script (the contents of your `.R` file pasted into the Word document)
– Author contribution statement