**Working with site x environment data**

Thus far, we have read in and performed basic analyses of the site x species data for Bryce Canyon National Park. In this lesson we will import and analyze the site x environmental data for the same plots. The data are in [brycesite.R](http://ecology.msu.montana.edu/labdsv/R/labs/lab2/brycesite.R), which should already be in your course folder, downloaded from the course website.

Open a new RStudio session and set your working directory with the following libraries:

*labdsv*

*MASS*

*mva*

*optpart*

*stats*

*vegan*

To get the data into R as a data frame, use the following code:

site <- read.table('brycesite.R',header=TRUE,stringsAsFactors=TRUE)

Click on site in the Global Environments window to examine the data. This dataset should look familiar, because we worked with it briefly in the Intro. to R/RStudio/RMarkdown lesson, mostly as a data frame contrast to the bryceveg.R site x species data matrix.

The column names are abbreviations; here is the legend:

plotcode = original plot codes

annrad = annual direct solar radiation in Langleys

asp = slope aspect in degrees

av = aspect value = (1+cosd(asp-30))/2

depth = soil depth = "deep" or "shallow"

east = UTM easting in meters

elev = elevation in feet

grorad = growing season radiation in Langleys

north = UTM northing in meters

pos = topographic position

quad = USGS 7.5 minute quad sheet

slope = percent slope

To work with data frame data, you either have to specify the data frame name and the column you want to work with (denoted with an $) every time you select some variables, or you can do what’s called “attaching” the data frame. Attaching a dataframe allows us to eliminate the dataframe name and field specifier "$" when using a specific field in the data frame.

attach(site)

(I’ll show you an example comparison of using/not using attach() shortly, at the \*\*\* below.)

OK, now that we have the site x environmental data loaded, we can see that there are numerical (continuous) data and categorical data. Some of these variables are likely correlated with each other. Having lots of variables that are correlated with each other is problematic because those “teams” will “drive” patterns in a certain direction that independent data would not.

Let’s make some graphs of the joint distributions of pairs of variables. Such graphs can be used to quickly identify variables that are correlated.

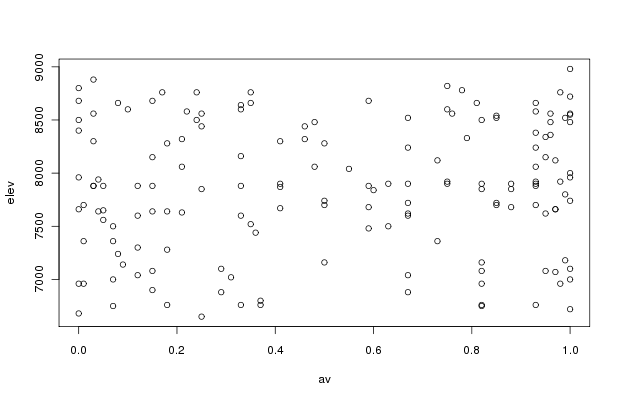
**Joint distributions for continuous variables:**

To look at joint distributions of variables, simply specify one variable as the X axis and the other variable as the Y axis:

plot(av,elev)

#to see the joint distribution of elevation and aspect value; #variable order is X,Y for plot

\*\*\*If we had not used the attach() function, the above code would have to have been written as plot(site$av,site$elev). You can still use that syntax even if you used attach(), it’s just not necessary.



The joint distribution of elevation and aspect values shows no pattern (other than a shotgun blast), meaning that these variables aren’t correlated with each other. (We wouldn’t have expected a relationship between elevation and aspect, anyway.) If you want to test for the actual correlation between them:

cor(av,elev)

[1] 0.09244026

#This is Pearson’s r, the correlation coefficient. This value is #nearly 0, indicative of independence between the two variables.

Sometimes we're interested in how two variables co-occur given a third variable. For example, in Bryce Canyon we might be interested in how the joint distribution of elevation and aspect value varies as a function of soil depth. The simplest approach to this problem is to split the graphic page into two panels as follows:

par(mfcol=c(1,2))

#to get a double plot

The par() function sets numerous graphics parameters (or "par" for short). The specific parameter to be set is passed as an argument to the function. So, mfcol stands for "multi-figure by columns" and the c(1,2) specifies "one row, two columns." (mfrow is for graphs stacked on top of one another; mfcol is for graphs side by side.) Accordingly, we will get a double figure with one panel to the left of the other:

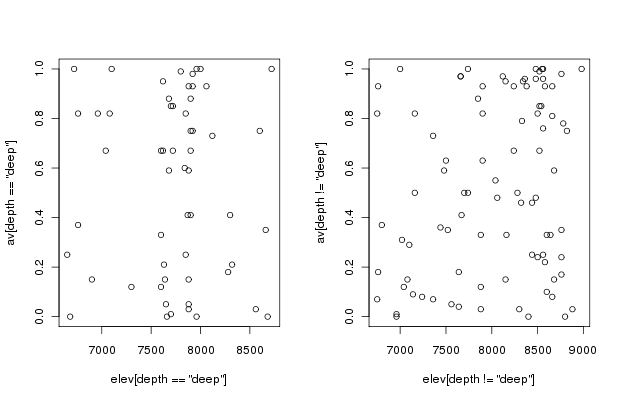
plot(elev[depth=='deep'],av[depth=='deep'])

#to see the joint distribution on deep soil

plot(elev[depth!='deep'],av[depth!='deep'])

# to see the joint distribution on shallow soil

(Go back to the Intro to R/RStudio/RMarkdown lesson for a reminder on R vector and matrix operators, p. 11, to understand what == and != mean.)



You have to tell R when you’re done with multiple plots:

par(mfcol=c(1,1))

# return to single plots

Boxplots are also useful and can be generated for joint distributions:

boxplot(elev~depth)

#visualization of elevation by soil depth

#notice that the Y variable is listed first, different from plot()



The tilde (~) is R-speak for "as a function of" so boxplot(elev~depth) plots elev as a function of depth.

This boxplot shows that there is a lot of overlap in soil depth with elevation; we can formally test this with a t-test.

t.test(elev~depth,var.equal=FALSE)

Welch Modified Two-Sample t-Test

data: elev[depth == "deep"] and elev[depth != "deep"]

t = -3.0848, df = 135.07, p-value = 0.0025

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-482.4255 -105.5039

sample estimates:

mean of x mean of y

7731.429 8025.393

t.test() is the name of the function, and we're using the same order of variables that we used in the boxplot. The var.equal=FALSE argument means not to assume equal variances in the two distributions.

The upper part of the output reports the value of Student's t, the degrees of freedom, and the p-value. The significant p-value indicates that in this case, deep soils do not occur at the same mean elevation as do shallow soils. In fact, as reported a few lines later, the mean for deep soils is 7731 feet and for shallow soils is 8025 feet, and the 95% confidence interval of the difference between them is 105 and 482 feet. The difference is less than we would expect from the boxplots, perhaps, because the distribution of elevations on shallow soils is skewed, making the mean significantly lower than the median, which is the bar indicated on the boxplot. t-tests have lots of assumptions; their use here is simply to reinforce the idea that we should use caution interpreting the meaning of elevation and soil depth without accounting for the other.

So we have plotted joint distributions and boxplots, and done simple, univariate, parametric tests of correlation and a t-test. Those are things that can only be done for continuous variables. For categorical variables, we are pretty much limited to table output.

**Table output for categorical variables:**

We can use table() to look at the number of sampling plots in each class; for example, use table(pos) to see the distribution of topographic positions:

table(pos)

bottom low\_slope mid\_slope ridge up\_slope

20 33 54 18 35

or table(pos,depth) to see the joint distribution of soil depth by topographic position:

table(pos,depth)

deep shallow

bottom 13 6

low\_slope 14 14

mid\_slope 18 31

ridge 5 12

up\_slope 6 26

Similar tables can be produced for other variables.

*Data standardization:*

Add decostand() in *vegan* info from Gardener 2014 pp. 377-380

**Assignment:**  due 0800, Monday, 22 Feb. 2021

Start a fresh RStudio session. Remember to set your working directory to your course folder and use the same package libraries as we used today. From the course website, download the dataset named Ground\_beetles\_habitat.csv to your course folder.

This is the same file you worked with last week, so these first few steps are going to be repeats from what you did last time. First, read in that file into your R session and call it gb. (Header = TRUE since the first row in the file is a list of column names.)

Remember from last time that the data in this data frame are not in a “rows = samples, columns = species and environmental data” format. The columns are:

Species = scientific name

Quantity = number of individuals (i.e., abundance)

Sample = alphanumeric code for each of the 18 sampling locations (i.e., plots)

Abbr = abbreviation of species’ scientific name

Max.Ht = maximum height of the vegetation in the sampling location (plot)

Habitat = three habitat types (edge [ecotone], grass [grassland], wood [woodland]) were sampled, with 6 plots per habitat type

Let’s find out which of the three habitat types had the most individuals, on average. This calls for the tapply() function. We’ll create a new R object called gb.habitat that contains the results of the tapply() function. Put the habitats as rows and species abbreviations as columns:

gb.habitat = tapply(gb$Quantity, INDEX = list(gb$Abbr, gb$Habitat), FUN = mean)

In the above code, FUN = mean indicates that we are taking an average (mean).

Use the head() function to examine the result. Two things immediately stand out about this dataset. First, notice that some entries for Quantity are not round numbers. If Quantity is number of individuals, how can this be? Quantity for gb.habitat is the average number of individuals for each habitat type across all species! Second, you will see lots of entries as NA; in R-speak, **NA stands for missing data**. But if you look at the original data table, you won’t find any missing data entries. In this case, NA stands for 0s (no individuals of a given species in a given habitat). This is something quite different than truly missing data! So let’s replace the NA items with 0s:

gb.habitat[is.na(gb.habitat)] <- 0

Again, use head() to make sure the result looks correct.

Now use either the colSums or rowSums (you must select the correct one!) to answer the following question:

**Q1. Which habitat type had the highest average abundance of beetles?**

Now suppose you wanted to know which habitat type had the most species rather than individuals. Based on what you did last week (used the table() function to create a contingency table), you might think that this would be the way to get that answer:

gbhab.pa = table(gb$Sample, gb$Habitat)

colSums(gbhab.pa)

However, notice that R simply sums species richness by each site, without taking into account that some species were found in more than one site. You can see from the original data that there were only 48 beetle species present, so the numbers returned by colSums(gbhab.pa)are not what we are looking for.

It is shockingly difficult to answer such a simple question in R—how many species are there by a given category in our dataset (in our case, we are interested in species richness by habitat)—with our fairly straightforward data. We will use the specnumber() function from *vegan*. specnumber() requires that the species are the columns and that the categorical groupings that you want (in our case, habitat types) as the rows. Our gb.habitat matrix is in the opposite orientation, but we can transpose rows and columns with the t() function and then apply specnumber():

tgb.habitat <- t(gb.habitat)

beetle.species <- specnumber(tgb.habitat)

So now you can answer this question: **Q2. Which habitat type had the most species?**

Now suppose we’re interested in whether a focal species (*Abax parallelepipedus*, abbreviated as Aba.par) exhibits differences in abundance by habitat. We can use a function called aggregate() with a subset argument that will allow us to display only a subset of the data; in our case, we want all of the samples that contained Aba.par, in what habitat type each of those samples was, and how many individuals were counted in each of those samples:

aggregate(Quantity~Abbr + Sample + Habitat, data=gb, FUN=max, subset = Abbr %in% "Aba.par")

That command aggregates a subset of the data; now you can graph it:

boxplot(Quantity~Habitat, data=gb, subset=Abbr %in% "Aba.par")

Based on the resulting graph, answer the following question: **Q3. Interpret how abundance of *Abax parallelepipedus* is related to habitat type.**

Prior to now, you’ve worked with single datasets. But separate but related datasets can be worked with together. For example, having a site x species dataset and a site x environment dataset isn’t uncommon, especially if you have different teams of researchers collecting different sets of data: one set of researchers collects species info at sites while another set of researchers collects abiotic information at those same sites.

First, we need to install a package that we haven’t used before, *picante*. Go ahead and do so (install and library).

Read in the grassland.community.csv file from the course website:

comm <- read.csv("grassland.community.csv", header = TRUE, row.names = 1)

The row.names = 1 argument indicates that the first column has the name to apply to each row. (It’s like a header statement for rows.)

This is a site x species dataset. Each cell contains the percent cover of a species in a sample. Because many multivariate methods are sensitive to the total abundance in a sample, we must convert these absolute abundance estimates to a relative abundance estimate. First, check total abundance in each sample:

apply(comm, 1, sum)

Then turn percent cover to relative abundance by dividing each value by sample total abundance:

comm <- decostand(comm, method = "total")

Check total abundance now in the converted data matrix:

apply(comm, 1, sum)

Now read in the plot.metadata.csv file from the course website and name it metadata. It also has column names and row names, so use the appropriate syntax.

This is a site x environment dataset with metadata about the samples in grassland.community.csv, including the habitat and site they were collected from, and a few basic environmental variables such as slope and moisture regime.

To relate the two datasets, we must first check whether the data listings in the two files are in the same order:

all.equal(rownames(comm), rownames(metadata))

This returns the value TRUE, indicating that they are in the same order. (If they didn't, we could sort them by this code: metadata <- metadata[rownames(comm), ] .)

OK, so now we have site, species, and environment data, we can ask scientific questions, like what the difference is in species richness by the two different habitat types where sampling occurred (fescue and mixedgrass). Make a boxplot that shows species richness by habitat, and use it to answer this question: **Q4. Does species richness differ by habitat type? Interpret the plot, then do a t-test (assume unequal variances).**

At this point, you may be thinking, “Why didn’t we make a boxplot of ground beetle species richness by habitat type to compare edge vs grass vs wood?” (Go ahead and try making one with similar syntax as what you used to answer question 4, and you’ll see that it won’t work.) Take a look at the Ground\_beetles\_habitat.csv file and think about what a boxplot shows. **Q5. How would you answer your question?**

(For fun, you might try figuring out how to make a boxplot of ground beetle species richness by habitat type!)

Make an RMarkdown Word file of your work and turn that in. Be sure to include your answers to the questions asked! Turn in your assignment as a Word document via email to [iroro.tanshi@ttu.edu](mailto:iroro.tanshi@ttu.edu) no later than 8:00 a.m. on Monday of next week. In your email, please include the following as the Subject line:

Assignment on site x environment