Plant Ecology **Spring 2006**

Syllabus

Instructor: Dr. Hank Stevens *Course No.: BOT 434/534, Plant Community Ecology Methods*

Contact Information

Dr. Hank Stevens

Office 338 Pearson Hall

Phone 529-4206

Class Meeting Times 320 Pearson Hall, Tuesdays, 1 - 5 PM, Jan. 11 - Feb. 8.

Office Hours 1-3 Thursdays.

E-mail HStevens@muohio.edu. We encourage you to email us, and when you do, please put 'BOT401' in the subject line. Otherwise we may delete your email by accident.

1 Description of the Course

This one credit laboratory course focuses on the processes of data collection, management, screening, and analysis of multivariate data from plant communities.

I emphasize vascular plant communities because many students must grapple with vegetation data, whether their research focuses primarily on animals, plants, or soil microbes. In spite of the plant-centered focus, the statistical and programming methods we cover apply to any biological community. We spend one week sampling plant communities, and 4 weeks learning statistical programming and data management, and analyzing multivariate data sets.

Plant communities are biologically important. Plant communities cover the vast majority of the land surface of the planet. This vegetation drives or underlies the dynamics of animal populations and communities, and ecosystem functioning, such as energy budgets or nitrogen cycling. Plant communities respond to both long and short term variation in climate, and to variation in disturbance regimes. Because of its prevalence, its responsiveness, and its importance to other ecological dynamics, a great deal of research in ecology and environmental science requires a description of plant communities.

This course is a sprint course to introduce methods of collecting, analyzing, and interpreting ecological data on plant communities. It will thereby provide background for research, as well as for careers with organizations such as The Nature Conservancy and agencies such as the U.S. Forest Service.

This course (BOT 434/534) is a work in progress. Its content changes yearly, reflecting course development, current topics and methods in ecology, the needs and pre-existing expertise of the students, and student feedback on the course. Because this course is not rigidly defined, the syllabus is subject to change. We will update the syllabus as we go, reflecting what we manage to cover in each class period.

The course is a one credit lab course, done in a sprint format, over five weeks. Therefore, the amount of work a student should be ready to do is equivalent to a three credit upper division (BOT 434) or graduate course (BOT 534), but for only 5 weeks. Toward this end, I will be offering office hours during which we can *all meet and work together* on the computationally intensive parts of the course.

2 Software

This course will make extensive use of the R programming language. This language is very similar to the S language that is used in the S-Plus statistical sofware package. Nearly all code that runs in R will run in S-Plus as well. Among the advantage's of R over the commercial package S-Plus are that R tends to be updated more frequently, it has a truly helpful and nearly instantaneous list-serve, R is free (GNU licensed, like Linux), and it runs on every computer platform I have ever heard of including, of course, Macintosh and Windows, OS/2, Unix, and Linux.

The following is a short excerpt from R's Homepage (http://www.r-project.org/):

The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

3 Course Readings

We will rely primarily on:

- Stevens, M.H.H. 2005. Introduction to Community Analysis in R. *This is a lab manual I will provide in class.*
- McCune, B. and J. B. Grace. 2002. Analysis of Ecological Communities, MjM Software Design. 307 pp.

Order McCune and Grace (2002) from http://www.ptinet.net/~mjm/index.htm.

We may also read from other sources including both the peer-reviewed literature, book chapters, and on-line references. These will be distributed in class.

4 Data sets

You will generate one data set from data collected the first day of class, and will be provided with several others, including the following:

veg.csv A plot × species matrix, with column headers; comma delimited. Use for Introductory section for ordination.

env.csv A plot × environmental variable matrix used in the Introductory section and for ordination.

veg1.dat A small three column matrix with plot, spp, dbh; tab delimited. Use in Exploratory data analysis for learning about data manipulation.

veg2.dat A moderate sized three column matrix with plot, spp, dbh; tab delimited. Use at the end of Exploratory data analysis to calculate importance values.

5 Course Outline

- Aside from the first day in the field, all other class meetings will be in hands-on computer programming and data analysis.
- Each class period will cover a specified part of the lab manual.
- The precise amount of material we cover will be guided by our progress through the material.
- The assigned reading should be read by the next class period, and reviewed following the class.

Week 1

Tuesday

In-class Manual (Chapter 1). Fieldwork! Sampling woody vegetation with plot-based methods.

Assignments 1. Enter data. 2. Read McCune and Grace (2002) 1-24; Manual, Chapter 2.

Thursday

Office hours Manual (Chapter 2). Introduction to the R language.

Assignments Read McCune and Grace (2002) 35-44, 58-79; Manual, Chapter 3.

Week 2

Tuesday

In-class Manual (Chapter 2). Simple Exploratory Data Analysis.

Assignments Read McCune and Grace (2002) 58-79.

Thursday

Office hours Finish Chapter 2. Simple EDA.

Assignments Read McCune and Grace (2002) 102-116; Manual, Chapter 4.

Week 3

Tuesday

In-class Manual (Chapter 4). Principal Components Analysis.

Assignments Read McCune and Grace (2002) 125-142.

Thursday

In-class Finish Chapter 4. Principal Components Analysis.

Assignments Read McCune and Grace (2002) 125-142.

Week 4

Tuesday

In-class Manual (Chapter 5). Non-metric Multidimensional Scaling.

Assignments Read Minchin 1987

Thursday

Office hours Finish Chapter 5, Non-metric Multidimensional Scaling.

Assignments TBA

Week 5

Content TBA. Possible topics include: monte-carlo simulation and testing; Graduate Student Projects, Cluster Analysis.

6 Evaluation of Student Performance

- Programing Journal - 50%
- Exercises - 40%
- Participation - 10%

6.1 Programming Journal

Students will be evaluated on three aspects of the course: their programming journal, explicit independent exercises, and collegiality. The programming journal is the cornerstone of this course, providing you with a powerful learning tool. The explicit exercises are more like standard problem sets or homework assignments, in that I ask you to solve particular, defined, problems. Collegiality is simply participation in a semi-group, semi-autonomous enterprise.

The programming journal is a day-by-day, minute-by-minute journal of what you do on the computer. It is really a cleverly disguised writing exercise that documents your process of discover. As you go through the tutorial I have provided, you will write computer code to manage and analyze your data, and interspersed with the code, you will write notes to yourself about what you are doing. You will "comment" the code. Any and all comments are good to put in.

As you enter code, you will also include comments that fall into several categories, and in essence, you should strive to write your own tutorial. *First*, you should include comments at the beginning of each section that remind you of the purpose of the following material. This purpose may may code oriented, such as learning about a new function, or it may be data oriented, trying to extract particular information from the data, or it may be both. Thus you should approach the code having read through it once, and systematically organize your work. *Second*, you should include detailed explanations for each step, with some comments more detailed than others. *Third*, you should include diary-like comments that are spontaneous responses to what you are doing. You need to take notes on what you are doing, what you discover about the code, and what you have to tell yourself in order to make sense of the code. It will be like a diary, in that you should feel free to write anything, and it is, in general, temporally sequential. It will also be like a private tutorial to yourself about items that you want to elaborate on. You will also want to reflect on what you have down in the past, so that you bring to bear new knowledge on old problems, so feel free to edit your comments. It will give you an opportunity to interact with your best friend - yourself!

I will evaluate the journal by assessing its completeness. In one sense, chapters 3-5 are such a programming journal, but it is my journal and its intended audience is you. In another sense, the code will speak for itself - it needs no comment, because it is exactly what it is. A boneheaded instructor like me, however, needs to by led by the nose through each step, so I will look an organizing structure or outline, and for comments explaining each step. I will also look for reflection (e.g., "Wow, that was so cool that I could do linear regression in a single step!") and more thoughtful analysis (e.g., "Frequency seems to differ from density in that frequency emphasizes spatially disjunct occurrences, whereas density emphasizes local abundance."). I will rate the programming journal A, B, C, D, or F.

6.2 Exercises

The exercises are distributed throughout the lab manual, and I intend or you to complete them *in class or during office hours*. They must be completed successfully. Period.

If you completely the exercises independently, that's wonderful; if you help your peers as well, that is even better. If you complete the exercises in true collaboration with someone else, that is also good. If you complete the exercises with a lot of one-sided help from your peers, or lots of guided questions from me, that is, ..., well, better than not accomplishing them at all.

All together, the exercises total 100 points. The point values assigned to each exercise is indicated along with the exercise in the manual. In general, the point values of the exercises increase as you work your way through the lab manual.

6.3 Participation

Collegiality in a small group setting is important, and a facet of research that many people enjoy. I will assume everyone is collegial (A) until someone proves me wrong (B, C, D, F).

6.4 Grades

I will assign each student's final grades on the basis of the following formula, using the example of John Doe's scores:

```
A=4;B=3;C=2;D=1;F=0;

John.Doe.Grades <- c(B,A,A)

J.Doe.Final.Grade <- John.Doe.Grades * c(0.5,0.4,0.1)~\#~Weighting

\#~Thus~the~programming~journal~is~worth~50\%~of~the~final~grade,

\#~the~exercises~together~are~worth~40\%,~and~participation

\#~is~worth~10\%.
```

Upon completion of the course, you will know exactly what this means!

I have no problem with the *concept* of giving everyone A's, and students have typically gotten good grades in this course. I also find, however, that some students struggle with some of this material, and so in practice, not all students get A's.

7 Evaluation of Instructor Performance and Course Content

You will have an opportunity to evaluate me in this course. Please note that I take these evaluations very seriously, and they are one of the reasons the content of this course is in continual flux. Rest assured, also, that my bosses take your evaluations very seriously; they hear what students have to say.