STAT 361/661: Data Analysis

Syllabus, Fall 2009

John W. Emerson
Yale University

1 Objectives, Prerequisites, and Philosophy

This course is intended for first year graduate students in statistics, graduate students in other disciplines, and advanced undergraduate students. This course assumes students are familiar with intermediate probability and statistics on the level of our STAT 241/242 course sequence.

I'll be honest: I find this course difficult to describe. Will you learn about linear regression? Of course, although you shouldn't be starting from ground zero. Will you learn how to use **R** for linear regression? Of course, but that in itself isn't very ambitious; after all, dozens of software packages provide tools for linear regression. I don't want you to leave the course feeling that you have learned about a limited set of tools, allowing you to do only certain types of analyses. I want you to feel prepared to face the unexpected, equipped with a set of skills enabling you to adapt to the inevitable surprises of data analysis. When faced with a fresh challenge, I want you to think, "I may not know the answer, but I bet I can figure it out." Someday, I want you to think, "that was one of the most practically useful courses I had at Yale."

Along the way, we'll learn about computer programming, algorithms, data structures, probability, statistics, simulation, numerical techniques, optimization, graphical methods, and computational efficiency. You should be able to think critically about data, use graphical and numerical summaries, apply standard statistical inference procedures (when appropriate) and draw conclusions from such analyses. But most importantly, you should be willing to break out of the box and conduct new, innovative analyses of problems when standard analyses may not be appropriate.

This course will be computationally intensive, and there is no substitute for getting your hands dirty. I expect to make my share of mistakes this semester (some intentional, some not), and we'll learn from them together. In data analysis, I believe you learn as much (and sometimes more) when things "don't work" than when they go as planned. You have succeeded when you can figure out why something doesn't work (or why some analysis isn't appropriate) and deduce an appropriate course of action as a result. You must be willing to try out new things and to make mistakes – you can't break the computer (at least, it won't go up in smoke), and the sky won't fall. Seek to understand the mistakes, and move onward.

2 Topics and Schedule (more or less)

Week 1-2. An informal introduction to **R**. Note that graphical methods and simulation will be used throughout the course, so much of the early material is not just about the data analysis, but is about gaining confidence using **R**. Topics to include a comparison of t-procedures with non-parametric alternatives (including bootstrap and permutation methods).

Weeks 3 and 4. Simple linear regression, diagnostics, and one-way analysis of variance; an introduction to contrasts.

Weeks 5 and 6. Multiple regression and variable selection; more on contrasts.

Weeks 7 and 8. Generalized linear models, maximum likelihood and optimization.

Weeks 9 and 10. Survival analysis, cluster analysis, and an introduction to time series analysis.

Weeks 11 and 12. Topics in Bayesian data analysis; the Gibbs sampler; missing data; the EM algorithm.

3 Practical Information

- E-mail is the best way to reach me: john.emerson@yale.edu.
- Office: I'm in 24 Hillhouse Room B06, phone 432-0638.
- Office hours (tentatively) Friday 1:00-2:30, and by appointment.
- Lectures: Monday and Wednesday, 2:30-3:45 in the StatLab, 140 Prospect St..
- Lab/Review/Help Sessions (strongly recommended but not required): TBA. I expect that these sessions could save you several hours of homework time. Take advantage of them.
- TF: Peisi Yan (peisi.yan@gmail.com). Office hours TBA.

4 Books

4.1 Free!

An Introduction to R, by Venables and Smith and the R Development Core Team. I can get you an official bound softcover version of this book, as well as **The R Reference Card**, for $15, printed by REvolution Computing here in New Haven. However, both may be downloaded for free from the R Project website (`http://www.r-project.org`).

There are plenty of other free references available from the R Project website. If you particularly like (or dislike) any of them, please let me know.

4.2 Other references, in no particular order

- **Data Analysis and Graphics Using R; An Example-based Approach**, by John Maindonald and John Braun. Cambridge Series in Statistical and Probabilitistic Mathematics, 2003. Has some nice things, but not quite right as the sole textbook for this course.
- **Modern Applied Statistics with S**, by Venables and Ripley. Don't be misled by the title, **R** is an implementation of **S**, and this book contains notes specific to **R** when appropriate.
- Sanford Weisberg: **Applied Linear Regression**.
- Ramsey and Schafer (2002), **The Statistical Sleuth: A Course in Methods of Data Analysis**, 2nd edition. There are some nice examples in this book, and I like the organization.
- Chambers and Hastie: **Statistical Models in S**. Again, the **S** language is what you're using in **R**.
- Yes, of course there are others. I may note a few of them as the course proceeds.

5 Exams

There are no exams.

6 Computing

Computing facilities are available at the StatLab (`http://statlab.stat.yale.edu`) and **R** is installed on Yale's cluster PCs, but I expect many of you will use your own laptops.

This is a computationally intensive course, and you will be required to experiment beyond the examples given in class. Unedited computer output will not be accepted; you must carefully document and explain your work. Some of what you will learn is "computer programming," so to speak; my examples in class should serve as a basis for any required analysis, but you will be expected to use other sources for help: online help, web searches, and some of the non-required references listed above.

R may be freely downloaded from the web (`http://www.r-project.org`), runs on all platforms (Mac/PC/Linux), and is very similar (though not identical) to S-Plus. Consultants at the Statlab are familiar with **R**, and may be able to help you get started with **R**; please see the Statlab web site for more information: `http://statlab.stat.yale.edu`. Note, however, the consultants are not Teaching Fellows for the course; please be considerate.

7 Grades and Homework

There will be regular homework assignments which will count for 75% of the final grade, 5% of which will be tied to an in-class presentation of part of your homework solution. This course will require a substantial time commitment each week (I've discussed the course philosophy elsewhere and won't defend the importance of this here), and to succeed in this course, you must stay on top of the material. This is particularly true in the first month of the course, when you will be building your statistical computing skills. A final project and class participation will constitute the remaining 25% (20% and 5%, respectively). The final project is due at 5 PM on the date of the scheduled final exam in December (there will NOT be any exams).

Note that failing to turn in one homework assignment will make a Yale College "A/A-" or graduate school "H" grade unlikely; I encourage submission of late homework (preferable to no homework), with a penalty of 10% for each day late. I will keep the Classes server (`http://classesv2.yale.edu`) completely updated with respect to the assigned homework.

8 Shopping this class and homework

I want to avoid any confusion. All homework assignments (including the first one) are required of everyone, submitted on time. This will not be an easy course to slip into after the first class or two.

9 Citation and thanks

I would like to thank my father/colleague, John Emerson of Middlebury College, for his teaching, support, and guidance over the years. I also owe a huge debt of gratitude to my advisor and colleague, John Hartigan of Yale University, for influencing my growth as a statistician and data analyst.

10 Getting Started with R

R is called a statistical programming environment. However, this is not a computer programming course, and I don't expect you to have any prior computing programming experience. Data Analysis is not a course about "learning to use **R**." It's a course about a few real-world problems, data analysis and statistics. Having said that, the practice of doing data analysis is necessarily computational, and you learn (and gain experience) by *doing*, not by listening to someone else talk about it. I have found that by engaging data using the **R** environment (instead of using a traditional menu-driven statistical software package), students refine their analytical skills and think more creatively.

I encourage you to start using **R** as soon as possible: it is freely available from http://www.r-project.org. It's installed on all the Statlab computers as well as Yale's cluster computers. But you'll probably want it on your computer, too.

From the main R Project web page, follow the "Download > CRAN" link. The mirror sites for the United States appear near the bottom of the next page: choose one (I usually use the one hosted at Carnegie Mellon). Next, you'll download and install **R** (you don't need the "source code"). See below.

10.1 R for Windows

On the Windows download page, click on "base." The only thing you need is the setup program, R-2.9.2-win32.exe. It will take a few minutes to download, and I recommend you save it to your desktop (instead of running it directly from the web). Once the download is complete, double-click the icon to start the installation. Simple questions are posed during the installation – accept the default settings.

10.2 R for Macs

I have less experience here, but fortunately the R Project page provides some helpful advice depending on your version of the Mac OS. Follow the instructions, and if there is a problem we can help you figure it out.

10.3 R for Linux

Wow. I'm guessing if you fall into this category you can figure it out on your own. But if you have trouble, just send me an email.

10.4 Done?

When the installation is complete, a big blue **R** icon should appear on your desktop (at least, in Windows). Congratulations – you're ready to start using **R**!

10.5 Try it!

You can't break **R**. You can't break your computer by using **R**. So, just try it out! Double-click the blue **R** that is probably on the desktop. If you can't find it, then maybe **R** hasn't been installed. Go back to the previous section, or ask for help.

When **R** starts, there will be a few menus and small buttons at the top, and a single inner window called the "R Console." By default, the **R** window will be maximized, filling the screen. I recommend you make it smaller (so it doesn't cover the entire desktop). Use a basic text editor (NOT Microsoft Word, but something like Windows Notepad) to accumulate commands as you work, building a script. I will do this regularly in class. If you have experience programming in another language such as **C/C++** or **Java**, you will be comfortable with this approach – just pretend you are writing a program, or parts of a program. See Figure 1 (page 7).

I realize the image may be difficult to read, but if you look closely there are two commands in the Console. The command prompt is `>`, and the first command is `iris[1:10,]`. I can even reproduce this for you in the notes, formatted nicely:

```
> iris[1:10, ]

   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
```

R includes more than a few data sets (often classical data sets such as this one from the Grandfather of Statistics, Sir R.A. Fisher), and they are available for immediate use. This one is contained in an **R** object called `iris`, and if you just typed `iris` at the prompt and hit return, you would see the entire data set (all 150 lines). Our first command, above, says "just show me the first 10 lines, or rows, of the data set." In fact, there are two commands, here, and you might guess the result of `1:10` (yes, among other things, **R** can count):

```
> 1:10

 [1]  1  2  3  4  5  6  7  8  9 10
```

Try it yourself! Don't be shy!

The second command produces a plot, called a "pairs plot" or a "scatterplot matrix." We can talk more about this later, but for the moment I'm satisfied if you just follow the directions and make sure you get the same plot as I did, reproduced in Figure 2.

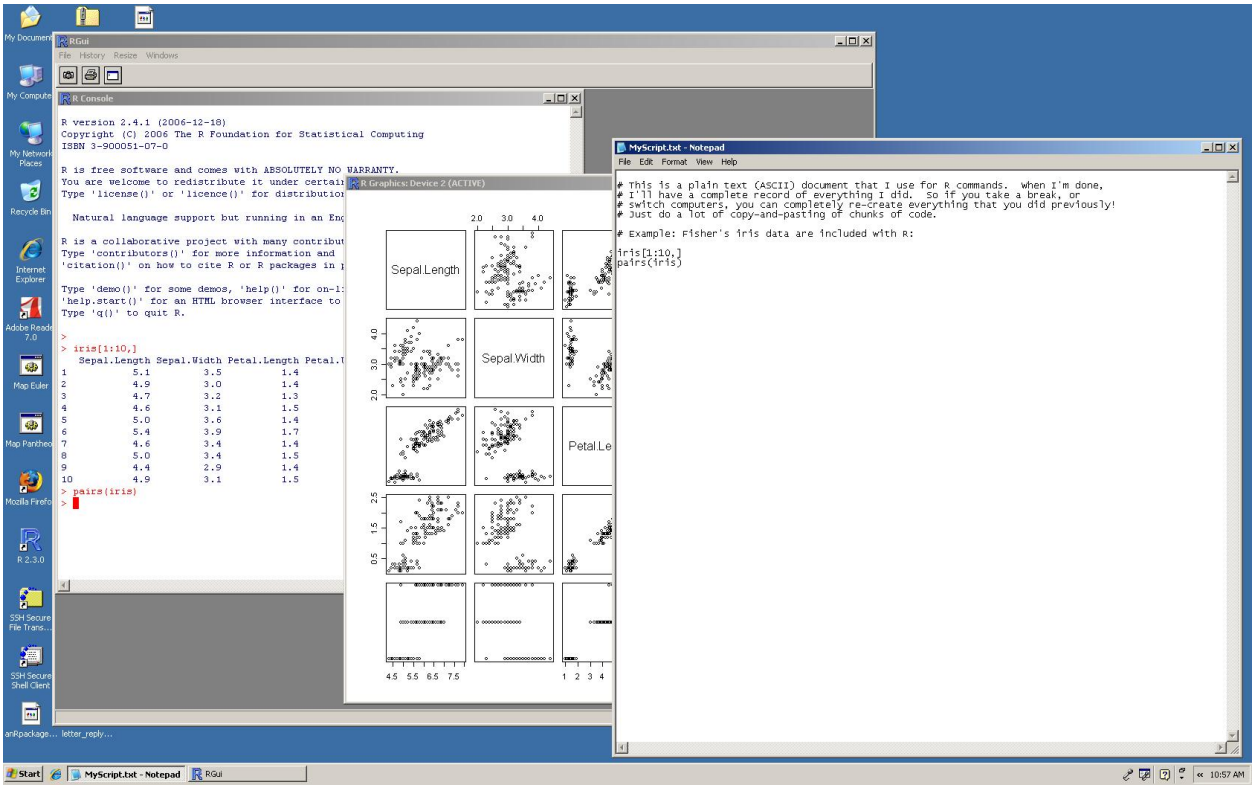

Figure 1: A screenshot showing the **R** window and the **R** console upon startup. I also use Windows "Notepad" to keep a record of my work. I cut and paste commands from the notepad into the **R** console. When there are mistakes, I fix the script in the notepad, and recopy the commands until I'm satisfied.

Finally, when you exit **R**, you will be asked about saving the workspace. I recommend *not* saving the workspace. Doing so can be risky, as new **R** sessions may then include objects used previously, a potential source of confusion. We will talk about saving results in class. To exit **R**, type `q()` in the Console and hit return. Then answer "no" – you don't want to save the workspace!

10.6 Caps matter

R does care about the difference between upper and lower case letters. Just be aware of that.

```
> pairs(iris)
```

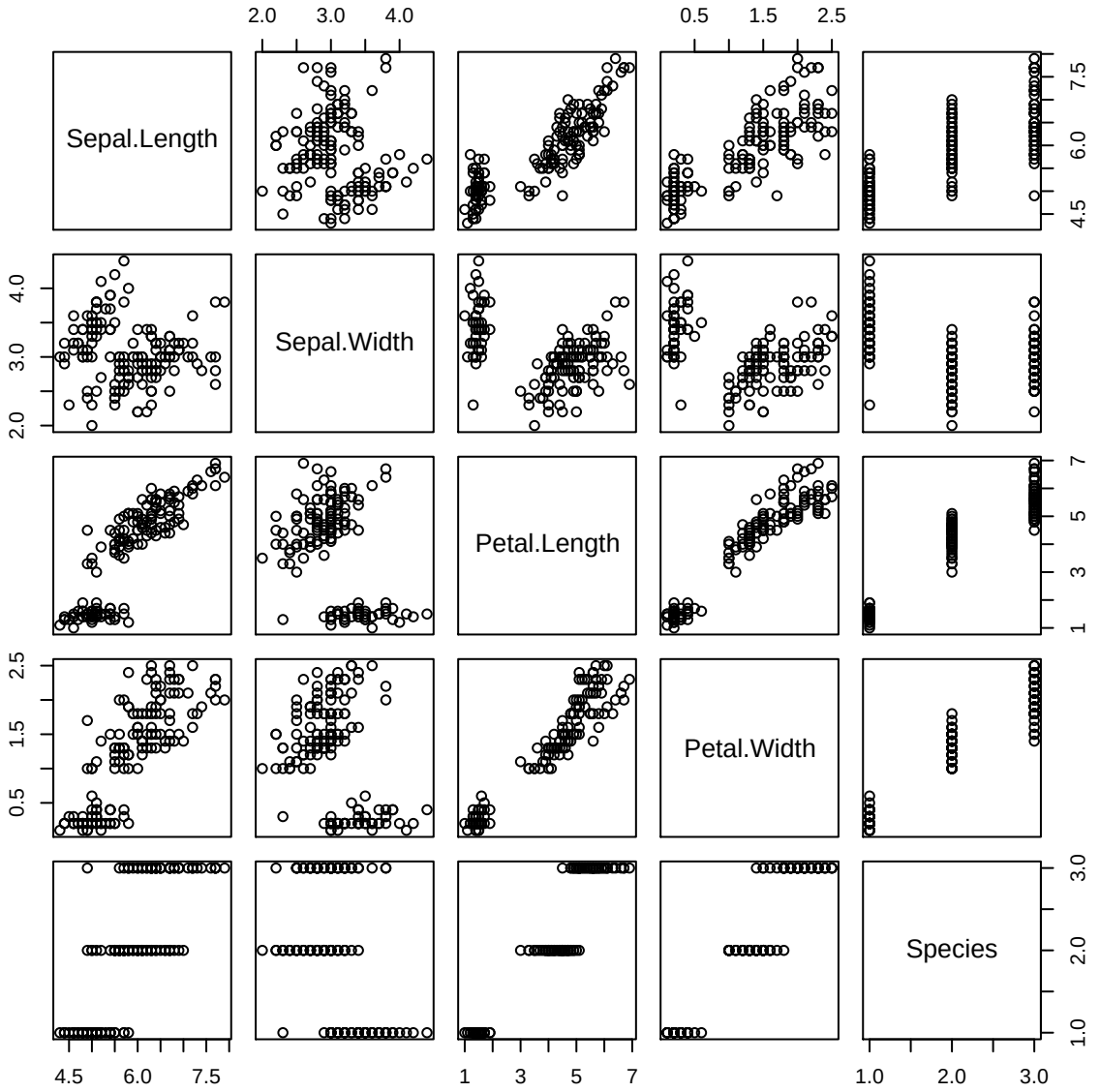

Figure 2: A pairs plot of Fisher's iris data.