STAT 230/530: Introductory Data Analysis

Syllabus, Spring 2011 (updated 1/6/2011)

John W. Emerson
Yale University

1 Practical Information

- Although my name is John, I would prefer that you call me Jay, Mr. Emerson, Dr. Emerson, or Professor Emerson. No extra credit points are based on this choice.
- 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: Friday 1:00-2:30 with others TBA.
- Lectures: Monday and Wednesday, 2:30-3:45, tentatively in Mason Labs 211; if interest is lower than expected, we'll move to the the Statlab, 140 Prospect Street (above the Social Science Library). Stay tuned for more information and check the **ClassesV2** site for updates.
- Lab/Review/Help Sessions (strongly recommended but not required): tentatively Monday evenings, 7-9 or 8-10 in the Statlab, 140 Prospect Street. Take advantage of them.
- TF: Taylor Arnold (taylor.arnold@yale.com).

2 Objectives

This course reviews and expands upon core topics in probability and statistics through the study and practice of data analysis. Topics include numerical and graphical summaries of data, hypothesis testing, confidence intervals, counts and tables, analysis of variance, regression, principal components, and cluster analysis. Upon completion of this course, students should be able to think critically about data and apply standard statistical inference procedures to draw conclusions from such analyses. This course will be computationally, not mathematically, intensive and will use the **R** language and environment for statistical computing and graphics.

3 Prerequisites

I'll assume a familiarity with topics covered in a typical introductory statistics course. Students with no prior coursework in statistics should consider taking STAT 100 (MWF 10:30-11:20). Taking STAT 230 as a first course in statistics is not generally recommended; permission from the instructor is required.

4 Philosophy

This course can be 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 a bit 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. In some cases, you might see opportunities 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.

5 Homework and shopping this class

Homework will be assigned and collected every Wednesday. While I understand and appreciate the role of shopping period, the first assignment will be due on Wednesday, January 19 and is required of anyone who plans to take the class. This will not be an easy course to slip into after the first week (with three class meetings instead of the usual two) and will be more fun and productive if you stay on top of the material, which builds on itself week after week. I will keep the **ClassesV2** site (`http://classesv2.yale.edu`)

completely updated with respect to the assigned homework. Problem sets will be collected at the beginning of class (printed copies are required and a printer will not be available in the Mason Labs); exceptions will require a note from your College Dean to avoid a late penalty (see below).

6 Exams

There are no exams.

7 Projects and real data

A final project is required. Students may work individually or in small groups. Keep your eyes open for real data opportunities; drop me an email if you have questions or if you find something you think I might be interested in. Some of the end-of-semester homework assignments will require demonstrating progress in data collection and analysis. The final project is due during the scheduled exam slot (the afternoon of May 5) and must be printed out and submitted to 24 Hillhouse Avenue before 4 PM.

8 Grades

The weekly homework assignments will count for 80% of the final grade. 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). This is particularly true in the first month of the course, when you will be building your statistical computing skills. The final project will constitute the remaining 20% of the grade, although progress on the project will also count as part of the later homework assignments.

Note that failing to turn in one homework assignments will make a Yale College "A/A-" (or a graduate school "H") grade unlikely; I encourage submission of late homework (which is preferable to no homework), with a penalty of 10% for each day late.

9 Shopping period meetings

I'll be available for short meetings during the first week of shopping period if you have questions or concerns. Please see the **ClassesV2** course page **Sign-up** starting January 8.

10 Proofs

If you want rigorous mathematical proofs I can recommend another class (STAT 242).

11 Computing

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

This is a computationally intensive course, and you will be required to experiment and follow examples given in class. Unedited computer output will not be accepted; you must carefully document and explain your work. You will not be required to do original "computer programming," so to speak; my examples in class will serve as a basis for any required analysis.

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

12 Books

There is no required text. I will provide fairly extensive notes. You might want to have access to your introductory statistics textbook or something comparable. I particularly like **Stats: Data and Models** by De Veaux, Velleman, and Bock.

12.1 Free!

An Introduction to R, by Venables and Smith and the **R** Development Core Team. This may be downloaded for free from the **R** Project website (`http://www.r-project.org/`, see **Manuals**). 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.

12.2 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. I've used it in the past, but it's terribly expensive.

- For a light (and fun) reference on introductory statistical concepts (not data analysis), you might consider a book by Gonick and Smith (1993), **The Cartoon Guide to Statistics**.
- Yes, of course there are others. I may note a few of them as the course proceeds.

13 For graduate students

Graduate students will have some additional requirements, including occasional extra sessions (perhaps weekly) to be announced.

14 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.

15 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. Introductory Data Analysis is not a course about "learning to use **R**." It's a course about data analysis and statistics and will include the study of a few wonderful real-world problems. 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, Packages: CRAN** link. The mirror sites for the United States appear near the bottom of the next page: choose one (I tend to use the one hosted at Carnegie Mellon). Next, you'll download and install **R** (the first section at the top of the page, not surprisingly called **Download and Install R**; you don't need the "source code"). See below for the different operating systems. The current version of **R** is 2.12.1.

15.1 R for Windows

On the Windows download page, click on **base**. Download the installer, **Download R 2.12.1 for Windows**. 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.

The only "gotcha" I've run into lately under Windows is with Windows Vista. If you have Vista, please see the note under **Frequently asked questions** before downloading the installer.

15.2 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. Taylor recently bought a Mac, fortunately!

15.3 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.

15.4 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**!

15.5 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 8).

I realize the image may be difficult to read, but if you look closely there are two commands in the **R 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) that 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
```

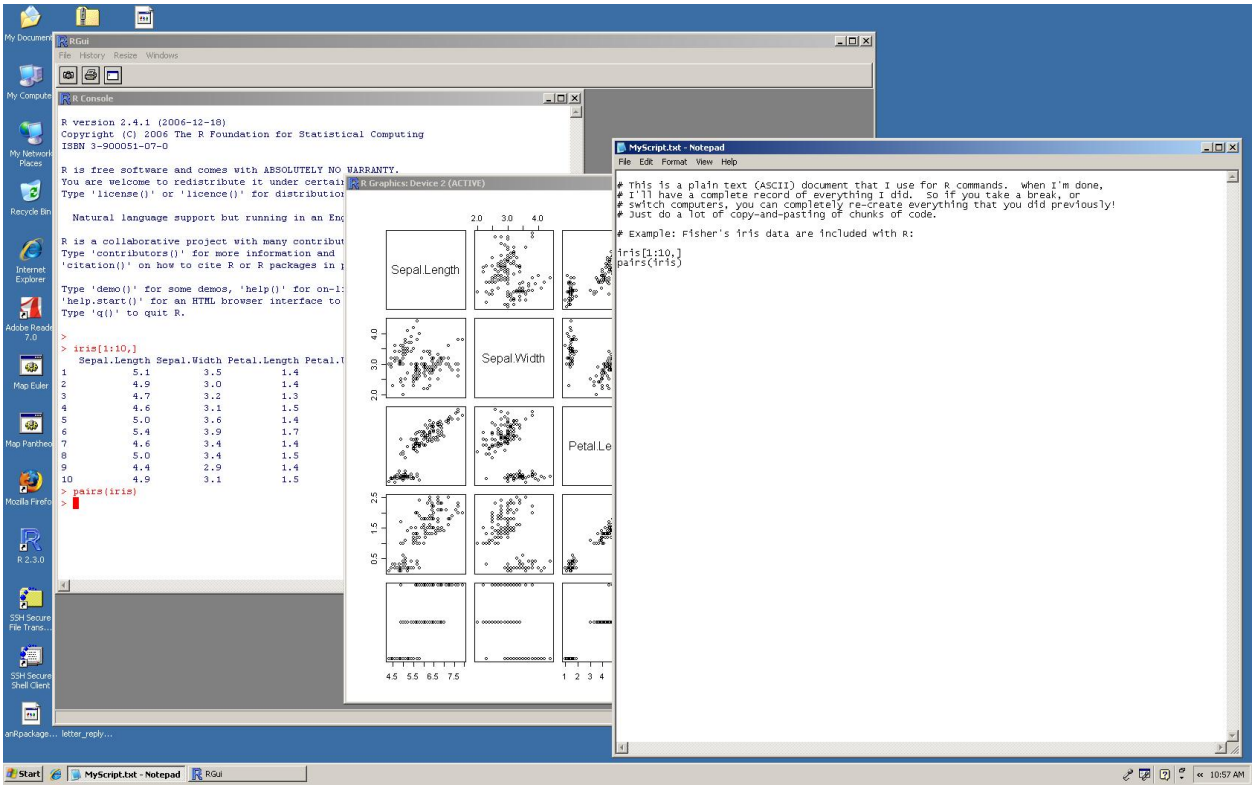

Figure 1: A screenshot showing the **R** window and the **R** console upon startup. I often use **Windows Notepad** to keep a record of my work. I cut and paste commands from the notepad into the **R** console. When there are mistakes, I fix the script in the notepad, and recopy the commands until I'm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

Try it yourself. Don't be shy.

The second command in Figure 1 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.

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 **R Console** and hit return. Then answer "no" – you don't want to save the workspace!

15.6 Caps matter

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

```
> pairs(iris)
```

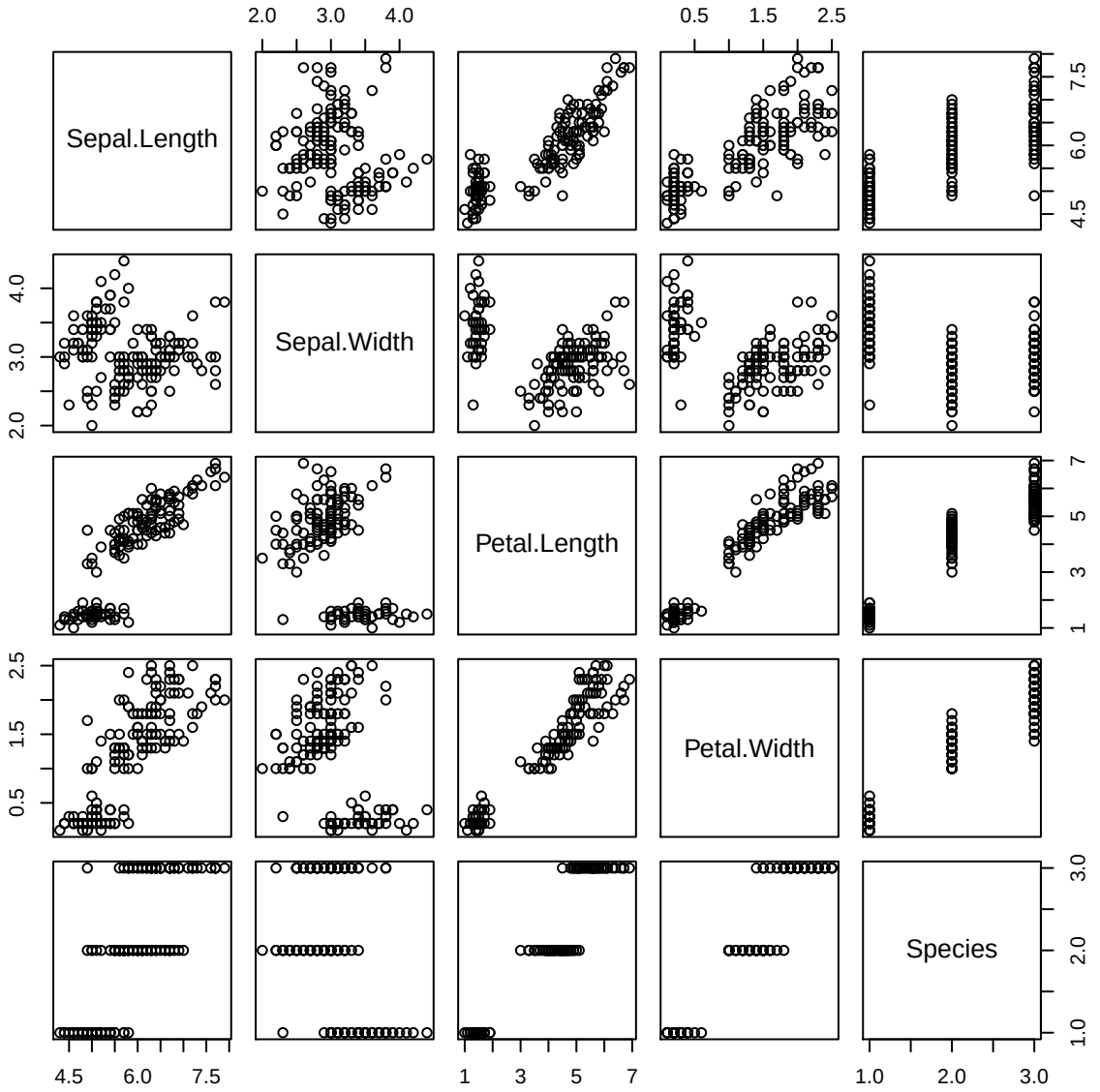

Figure 2: A pairs plot of Fisher's iris data.