

## Course Logistics

How to organize a statistical computing class?

## Stat 133

- 80 students
- Sophomores and juniors
- Stat, Math, CS, and other science and eng majors
- No prerequisites for the course
- 3 hours of "lecture"
- 1 hour of lab
- Undergraduate student assistants – lab time and online office hours

## Lecture Formats

- Annotated transcripts - prepared before class and posted after class
- Worksheets – focus on main concepts; students write up solutions on board or instructor types them into computer and we discuss
- Time dedicated to discussion of projects – first amongst groups, then whole class
- Powerpoint - limited use, post after class
- Interweave vignettes, stat topics, computing, project background - may be not all in one lecture but definitely all in one week.
- Demo – still looking for more active learning techniques

## Lab Format

- Required attendance
- Worksheets – early on students worked on worksheets that focused on concepts needed for assignments
- Free-form – most recently students work on assignments and get assistance from TA and Undergrad TAs
- Undergrad TAs – hire 2-3 students who took the class the previous semester to act as helpers in lab and in chat room
- New labroom, shared laptops, tables seat 2-3, projector can display student work: can move in and out of whole class discussion and group work

### Electronic discussions

- Asynchronous forums –
  - Thread for each HW/project and for general questions
  - Progress from instructor answering all questions to students answering each others questions
  - Avoid posting code
  - Expect everyone to post a question or answer within first couple of weeks of the semester
  - Any email questions or OH questions that are pertinent to all, instructor posts the Q and the A or asks the student to post the Q in order to get the A.

### Electronic Discussions

- Chat room
  - Converted instructor OH into chat room time
  - Later in the evening when students are working on assignments
  - It can get quite confusing when answering many questions at once
  - Many students hang out in the chat room observing; they gain courage to ask questions as semester progresses

### Other Electronic forms of communication

- Wiki
  - Used to create groups, and post their assignments
  - Potential to create course materials
- Google groups
  - Archive of messages sent to the group
  - Expect all students to participate in asking and/or answering questions

### Homework and Projects

- Balance 6 HW and 2 projects
- HWs
  - Smaller in scope
  - Mix of computational tasks and EDA
  - Feed into projects
- Projects
  - Break into stages – quality control, timing
  - Done in groups
  - Combines multiple topics

## Group work

- Most HWs are done individually
- Projects are done in groups
  - take photos of the group
  - Group name
  - Spend time in class creating groups
- When groups work on different projects, they meet me outside of class
- Mix up group members between first and second project
- Same grade for all group members, unless it is clear that someone has not done their share

## Alternative

- 10 week quarter, 30 students
- 5 Assignments
- Done individually
- Assignments mirror job assignment
  - Need to find out more details
  - Open ended in nature

## Kernighan & Pike - TPOp

- ☞ Style  
(Write code to be read by a human. It may be you.)
  - ☞ Use descriptive names for variables & functions, short names for local variables
  - ☞ Indent code appropriately
  - ☞ Parenthesize to resolve ambiguity
  - ☞ Define “variables” for constants that might change!
  - ☞ Using existing functions  
Spend time searching.

## Style

- ☞ Comment code & expressions.  
Don't belabor the obvious
- ☞ Don't comment bad code, rewrite it.
- ☞ Don't contradict the code in comments
- ☞ Clarify, don't confuse.

## Roger Peng's Coding Standards

- Program files should *always* be ASCII text files.
- Program files should always be immediately source-able into R.
- Always use a mono-space font to write code. Variable space fonts like Times New Roman or Charter or Georgia are not appropriate.
- Always indent your code. If you use an editor like GNU Emacs, then there is support for automatic indentation of code. Indentations should be 8 spaces wide.
- Comments should be indented to the same level of indentation of the code to which the comment pertains.
- Your code should not extend past 80 columns. Break long lines if you have to. Exceptions can be made only for hard-coded constants (such as path names or URLs) which cannot easily be wrapped or shortened.

## Grading Policy

- Participation – 4%
- HW Assignments – 6 for a total of 48%
- Projects – 2 for a total of 36%
- Final exam – 12%

## Exams

- Exam is there to make sure students have learned the basics
- Provide return values for code
- Apply reasoning to a particular setting
- Demonstrate understanding of basic concepts
- Tried oral exam once – with a set of provided questions

## Grading Homeworks

- Shell scripts to launch R and run code, time when profiling code
- Focus on one aspect of the HW
- TAs do HW grading

### Grading Projects

- Rubric for projects – holistic grading
- Grading projects – emphasis on written report and plots.
- Peer grading – I tried it last spring for another course
- Have each student denote contributions to project of each team member (split 100% of effort, excluding self) – ask on final

### Code of Student Conduct

- Want to encourage discussion, but the code and text must be your own
- Writing code is like writing a paper, you must acknowledge the contribution of others
- Syllabus includes a reference to the campus code
- If you are unsure what constitutes a violation of academic honesty, see the instructor