1 Introduction

This seminar is an introduction to the R programming language. The main focus is on using R for processing, analyzing, and visualizing large data sets. We’ll also cover the fundamentals of R and its data structures, using R for calculating statistics, connecting R to a database, and other useful features. The 6-week class meets twice a week.

2 Office Hours

Brian Levine (brian@cs.umass.edu): Fridays 2pm–3pm Room 274 of the Computer Science building.

There is no TA or grader for this class.

3 Grading

Your overall grade for the course will be the weighted sum of these components:

- 4 programming assignments (55% total).
- One short presentation (10%).
- One tutorial writeup (same topic as the presentation) (20%).
- The short proposal for your presentation and write up (5%).
- Participation (10%). I will provide a grade for participation once at 3 weeks and then again at the end of class, each worth half of this component. I will assign grades of only full, half, or no credit for participation.

- Extra Credit: You can receive extra credit towards your programming grade by publicly responding to questions about R posted to Stack Overflow, as long as your responses are helpful and demonstrate knowledge (and don’t repeat an existing answer, etc.). Be sure to send me your username at the start of the semester.

- Attendance: You can miss only one class unexcused over the 6-week period. If you miss two unexcused, then your participation grade can be no higher than 0%. If you miss three, your overall grade will be lowered by a full letter (after figuring in the 0%). You will receive a failing grade for the course if you miss 4 or more classes unexcused. Allowable excuses are defined by university policy (e.g., medical or religious).

There are no exams.

3.1 Assignments

I will use github exclusively to accept assignments. Each student will have their own private repository that only the student and I can access. I will not accept assignments late. Only the last check in before the due date will be graded.

You cannot use an available R library or function to complete an assignment that asks you to code up the solution yourself. For example, if I ask you to code up something that creates a naïve Bayes classifier, you can’t use a library that has it coded up already.

Assignments will be graded based on the rubric that appears in Table 1 (modified from https://www.csulb.edu/colleges/coe/cecs/views/programs/undergrad/grade_prog.shtml). Since the specification trait is worth twice as much as the other categories, assignments have a maximum grade of $2 \times 7 + 7 + 7 = 28$ points each. I won’t be debugging your code in order to determine how to give you more points. If the document doesn’t compile to a PDF via knitr/Latex and compute the assignment as embedded R code, I’m not going to read the native Rnw file to give you points.

4 Polices

All official material for the class can be found at http://forensics.umass.edu/classes/cs691BL/.

Cell phones etc are not allowed during class lecture. Please bring a laptop for using R-studio only. Do not surf the web, chat, etc.
Table 1: Programming assignment rubric.

<table>
<thead>
<tr>
<th>Specification (weighted twice)</th>
<th>Nicely Done (7 pts)</th>
<th>Flawed (4 pts)</th>
<th>Amateur (1 pt)</th>
<th>Zilch (0 pts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The program works and meets all of the specifications.</td>
<td>The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.</td>
<td>The program produces correct results but does not display them correctly.</td>
<td>The program produces incorrect results.</td>
<td></td>
</tr>
<tr>
<td>Efficiency and Readability</td>
<td>The code is exceptionally well organized and very easy to follow. The code is extremely efficient without sacrificing readability and understanding.</td>
<td>The code is fairly easy to read. The code is fairly efficient without sacrificing readability and understanding.</td>
<td>The code is readable only by someone who knows what it is supposed to be doing. The code is brute force and unnecessarily long.</td>
<td>The code is poorly organized and very difficult to read. The code is huge and appears to be patched together.</td>
</tr>
<tr>
<td>Documentation and Style</td>
<td>The documentation is well written and clearly explains what the code is accomplishing and how. The style guide is followed.</td>
<td>The documentation consists of embedded comment and some simple header documentation that is somewhat useful in understanding the code. They style guide is used somewhat.</td>
<td>The documentation is simply comments embedded in the code with some simple header comments separating routines. Some other style guide is used, I guess.</td>
<td>The documentation is simply comments embedded in the code and does not help the reader understand the code. What's a style guide?</td>
</tr>
</tbody>
</table>

4.1 Collaboration and Plagiarism

Your programming and answers to written assignments must be your own. When in doubt, contact me about whether a potential action would be considered plagiarism. Here are the guidelines. When asking others for help, you can show them small portions of your source code (the part with the bug). When providing help to others, you cannot show them any portion of your own code. Do not type out any solution for the receiver of help or dictate a solution. Describe the solution or explain what is wrong. When asking others for help, do not take notes about the solution other than to jot down publicly available references. Use only verbal communication. If you use paper during a discussion with others, throw it out before returning to your own assignment. If you use a whiteboard or blackboard, erase it before returning to your assignment, and so on.

If you do discuss material with anyone besides the instructors, acknowledge your collaborators in each write-up. If you obtain a key insight with help (e.g., through library work or a friend), acknowledge your source, briefly state the insight, and write up the solution on your own. You cannot post to public or private forums asking for help with any aspect of the assignments.

Never misrepresent someone’s work as your own. It must be absolutely clear what material is your original work. You MUST cite all your sources properly. You must remove any possibility of someone else’s work from being misconstrued as yours. I consider the facilitation of plagiarism (giving your work to someone else) to be plagiarism as well.

Plagiarism and other anti-intellectual behavior will be dealt with severely. If you plagiarize, I will work to remove you from this university, in addition to giving you a failing grade. Investigating plagiarism is a pleasant experience for neither instructor nor student. Please help us by avoiding any questionable behavior.

As a condition of continued enrollment in this course, you agree to submit all assignments to an automated service that checks your code against others (both in the class and outside the class). All submitted assignments will be included in the UMass Amherst dedicated database of assignments and will be used solely for the purpose of checking for possible plagiarism during the grading process and during this term and in the future.

If you discover that another student in the course is cheating, and you do not tell us about, you will be sanctioned as well.

You can read the University’s policies on cheating as well at [http://www.umass.edu/dean_students/codeofconduct/acadhonesty/#B](http://www.umass.edu/dean_students/codeofconduct/acadhonesty/#B).