Problem Set 5

R Programming (Due Mar. 21)

Instructions

1. The following questions should each be answered within an R script. Be sure to provide many comments in the script to facilitate grading. Undocumented code will not be graded.
2. Work on git. Fork the repository found at `https://github.com/carlson9/PS6`. As you add your code, commit and push frequently. Use meaningful commit messages – these may affect your grade.
3. You may work in teams, but each student should develop their own R script. To be clear, there should be no copy and paste. Each keystroke in the assignment should be your own.
4. If you have any questions regarding the Problem Set, contact the TAs or use their office hours.
5. For students new to programming, this may take a while. Get started.

Primary Race

1. To prepare you for your midterm, your goal here will be to use `devtools` to create an S4 R package.
2. The package should include appropriate functions and appropriate documentation.
3. The package should have a class named Candidate that has slots named name, delegatesWon, party, and delegatesNeeded.
4. When creating an instance of the class, only candidate name, delegates won, and party should be provided by the user, and the slot for delegates needed should be calculated within the class definition through a call to a function. As most users are unfamiliar with S4, create a function named createCandidate that takes as arguments the necessary information and returns an instance of class Candidate.
5. The package must include a `show` and `print` method for the main class of interest.
6. Create a function named propNeeded that takes as arguments a candidate and the number of delegates remaining, and returns the proportion of delegates remaining needed for the candidate to win.
7. You only need the one class, but using more than one might be useful. Subclasses may also be helpful.

8. GRADS ONLY: create a class named Race that has all the candidates, a slot for party, and a slot for delegates remaining (which should be calculated automatically). Include a `plot` method that can be called on an instance of Race that produces a meaningful plot of the candidates side by side, maybe a bar graph that shows total won and total needed. If you want to get creative, use different slots for pledged super delegates and use them in your display.