Problem Set 2

R Programming (Due Feb. 9)

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/jeonghkim/PS2` and add your code, committing and pushing 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.

Benford's law

Recent work in political science has proposed Benford's law as a method for identifying electoral fraud. The idea is that specific integer totals should appear in the *first significant digit* a known number of times if the data is being generated "naturally."

1) Calculating violations

Two ways of testing violations of Benford's law are proposed below. Let X_i represent the observed proportional frequency of the integer i in observed vote totals. So, for example, X_1 would represent the proportion vote totals where the integer 1 appears in the first significant digit.

- Leemis' m statistic

$$m = max_{i=1}^{9}\Big\{(X_i) - log_{10}(1 + 1/i)\Big\}$$

- Cho-Gains' d

$$d = \sqrt{\sum_{i=1}^{9}\Big((X_i) - log_{10}(1 + 1/i)\Big)^2}$$

Write a function to calculate these statistics. The function should take as an input (i) a matrix or vector of election returns and (ii) an option (or options) that controls whether the m statistic should be calculated, the d statistic should be calculated, or both. The output should be a list containing the results, *including the full digit distribution*.

2) Critical values

For each statistic, we can reject the null hypothesis of *no fraud* if the statistic reaches the critical values in the table below.

	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
Leemis' m	0.851	0.967	1.212
Cho-Gains' d	1.212	1.330	1.569

Create a new function called `print.benfords()` that will output a table containing:

- The name of each statistic
- The statistic as it was calculated
- The relevant number of asterisk's (e.g., one star for significance at the $\alpha = .10$ level, etc.)
- A legend at the bottom explaining the asterisk's (similar to what you see when you print an `lm` object.).

You can provide this output in any way you like, but it must be clearly organized and easy to understand. Don't forget to document your code.

Create another function that uses `print.benfords()` to create a csv containing the table in a directory provided as an argument to the function (hint: `sink()`).