Bios 301: Assignment 2

Due Monday, 15 October, 12:00 PM

50 points total.

Submit a single knitr (either `.rnw` or `.rmd`) file, along with a valid PDF output file. Inside the file, clearly indicate which parts of your responses go with which problems (you may use the original homework document as a template). Raw R code/output or word processor files are not acceptable.

Question 1

20 points

A problem with the Newton-Raphson algorithm is that it needs the derivative f'. If the derivative is hard to compute or does not exist, then we can use the *secant method*, which only requires that the function f is continuous.

Like the Newton-Raphson method, the **secant method** is based on a linear approximation to the function f. Suppose that f has a root at a. For this method we assume that we have *two* current guesses, x_0 and x_1, for the value of a. We will think of x_0 as an older guess and we want to replace the pair x_0, x_1 by the pair x_1, x_2, where x_2 is a new guess.

To find a good new guess x2 we first draw the straight line from $(x_0, f(x_0))$ to $(x_1, f(x_1))$, which is called a secant of the curve $y = f(x)$. Like the tangent, the secant is a linear approximation of the behavior of $y = f(x)$, in the region of the points x_0 and x_1. As the new guess we will use the x-coordinate x_2 of the point at which the secant crosses the x-axis.

The general form of the recurrence equation for the secant method is:

$$x_{i+1} = x_i - f(x_i)\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}$$

Notice that we no longer need to know f' but in return we have to provide *two* initial points, x_0 and x_1.

Write a function that implements the secant algorithm. Validate your program by finding the root of the function $f(x) = \cos(x) - x$. Compare its performance with that of the either the Newton-Raphson or the Fixed-point method – which is faster, and by how much?

Question 2

15 points

Import the HAART dataset (`haart.csv`) from the GitHub repository into R, and perform the following manipulations:

1. Convert date columns into a usable (for analysis) format.
2. Create an indicator variable (one which takes the values 0 or 1 only) to represent death within 1 year of the initial visit.
3. Use the `init.date`, `last visit` and `death.date` to calculate a followup time, which is the difference between the first and either the last visit or a death event (whichever comes first). If these times are longer than 1 year, censor them.
4. Create another indicator variable representing loss to followup; that is, if their status 1 year after the first visit was unknown.
5. Recall our work in class, which separated the `init.reg` field into a set of indicator variables, one for each unique drug. Create these fields and append them to the database as new columns.
6. The dataset `haart2.csv` contains a few additional observations for the same study. Import these and append them to your master dataset (if you were smart about how you coded the previous steps, cleaning the additional observations should be easy!).

Question 3

15 points

The game of craps is played as follows. First, you roll two six-sided dice; let x be the sum of the dice on the first roll. If x = 7 or 11 you win, otherwise you keep rolling until either you get x again, in which case you also win, or until you get a 7 or 11, in which case you lose.

Write a program to simulate a game of craps. You can use the following snippet of code to simulate the roll of two (fair) dice:

```
x <- sum(ceiling(6*runif(2)))
```

The instructor should be able to easily import and run your program (function), and obtain output that clearly shows how the game progressed.