EART 125: Resampling (two-sample tests) in R    Feb 22

Last class you learned how to use the `sample()` function to randomly resample a dataset and how to repeat that procedure many times using `replicate()`.

The approach for performing two-sample comparisons involves many of the same steps, and you actually have used all of the necessary R functions already, so the main challenge is deciding how to break the task into small steps and put those steps in an appropriate order. Programming, no matter what type, is fundamentally about breaking a problem into its component tasks and arranging them in a step-wise fashion to produce the desired result.

**Part 1: Bootstrap two-sample tests**

Because bootstrapping involves sampling with replacement, it’s actually easier to implement than randomization for two-sample tests. You already know all of the functions required, but there are two new ones that can be helpful:

- `abs(x)` #returns the absolute values of a numeric vector
- `rbind(a, b, ...)` #combines two+ matrices/data frames by rows

These two-sample tests mimic or simulate the procedure for standard hypothesis testing, so you must:

1) Combine the two samples into a single pool
2) Randomly sample the pooled data, with replacement, to create two new samples, with the same sample sizes as the two original data
3) Calculate the difference between the parameter estimate (mean, median, etc.) of the two random samples
4) Repeat steps 1-3 many times (e.g., 10000 times) to generate a sampling distribution
5) The p value is the probability of observing a difference in parameter estimates at least as extreme as the one observed between the actual samples

I would recommend first writing the steps out in words (as R comments, which are preceded by the # symbol), and then trying to convert each step into programming commands. It’s probably easiest to write steps 3 as a separate function, which you can then include within the `replicate()` command.

One warning: beware of NA values and how you treat them!

**Section 2: Randomization**

Randomization is set up in nearly the exact same way as bootstrapping. However, sampling without replacement does introduce additional complexity.

In sampling with replacement (bootstrapping), you can just run the `sample()` function twice, once to generate random group 1 and once to generate random group 2, because it’s OK for values to be sampled more than once. In sampling without replacement (randomization), each value can only occur once, so you can only use the `sample()` function once to produce a single vector with the values in randomized order.
However, remember that you can extract particular items from a vector by specifying their positions by a number in square brackets. For example:

`Vector[1]` #extracts just the first item from the vector  
`Vector[c(1,4)]` #extracts the first and fourth items from the vector  
`Vector[seq(10,15)]` #extracts items 10 through 15 from the vector  
`Vector[10:15]` #also extracts items 10 through 15 from the vector  

Again, it’s probably easiest to write your own function to do the randomization itself, and then you can put that function within `replicate()`.