Consequences of Early Stopping

Alex

10/9/2018

What happens if we stop early?

Write a small simulation to learn about what happens if we write a test that stops **as soon** as there is as "stat sig" treatment effect.

Here's the catch, make it so that there is *exactly* zero treatment effect.

```
whole_experiment <- function(experiment_size, number_of_experiments, crit_val) {
  ## What happens if we run this the right way?
  ##  ---
  ##  In this simulation, both y1 and y0 are drawn from the
  ## same distribution. That is, there is *no treatment effect
  ## at all*.

  p_val <- rep(NA, number_of_experiments)

  for(sim in 1:number_of_experiments) {
    y1 <- rnorm(experiment_size)
    y0 <- rnorm(experiment_size)

    p_val[sim] <- t.test(y1, y0)$p.value
  }

  return(p_val < crit_val)
}
```

If we've got this built, then we can run it in the next cell, and save the results of the simulation into an object called `res`.

```
res <- whole_experiment(experiment_size = 100, number_of_experiments = 1000, 0.05)
```

And so, presuming that we set our rejection criteria to be 0.05, we had about 0.064 of these experiments come back with a false positive result. That isn't too bad at all!

Let's see what happens if we work through this the *wrong* way.

```
cheat_experiment <- function(experiment_size, number_of_experiments, crit_val) {
    ##
    ## What happens if we run it the wrong way?
    ## ---
    ## In this experiment, we're going to simulate looking after *every*
    ## new person filters through your experiment. While there is still
    ## no treatment effect at all, the longer we look at the experiment,
    ## the more likely we are to call a winner.
    ## ---
    ## - The outter loop works through each of the *simulations* which is
    ##   an experiment.
    ## - The inner loop checks each of the rows in that particular
    ##   experiment.
```

```
    ##
    ## note that we're being just a little lazy in the coding this
    ## for legibility. we need at least four observations for the
    ## t-test to work. and we're indexing so that there are four
    ## leading NAs in our p-value results object that we're cutting
    ## out when we compute the test.

  p_val  <- rep(NA, experiment_size)
  reject <- rep(NA, number_of_experiments)

  for(sim in 1:number_of_experiments) {

    y1 <- rnorm(experiment_size)
    y0 <- rnorm(experiment_size)

    for(i in  4:experiment_size) {
      ## we need to build this inner loop so that we can check
      ## for the p-value after every new piece of data comes in
      ## this is additional compared ot the last function, but
      ## doesn't change the core nature of the function
      p_val[i] <- t.test(y1[1:i], y0[1:i])$p.value
      }

    ## in the first function we returned whether the p-value at the
    ## end of the test was smaller than the critical value. here
    ## wer're going to check to see if *any* of the p-values are
    ## smaller than the p-value. after all, that would be our trigger
    ## to stop the test.
    reject[sim] <- any(p_val < crit_val, na.rm = TRUE)
  }

  ## return whether we would have rejected the null hypothesis.
  return(reject)
}
```

With that function made, we can ask the question, what is our false rejection rate?

```
false_rejections <- cheat_experiment(100, 100, 0.05)
mean(false_rejections)
```

```
## [1] 0.39
```

Oh goodness.

If we have 100 observations, and we're willing to stop the experiment as soon as we observe *any* type of "stat sig" result, then, rather than having a 5% chance of a result being just noise, instead we're going to have a 39% chance of this result being just noise.

Not to worry though

This is the kind of problem that goes away if we have more data...

... right?

Task to do together:

Individually, or as a group: pick a *few* values of an experiment size, some smaller than 100, and some larger than 100.

- Before you run the simulation: Make a guess in your group about whether there will be a relationship between the size of the experiment group and the false discovery rate.
- Will a larger experiment falsely reject with higher, lower, or the same probability?
- Once you "experiment" (aka: goof around with this code) and see the pattern, can you explain what is happening that is leading to the pattern that you observe?

–>