

Lab 5: The pivot

Due 2/17, 3 Questions

In lecture on Monday, we began our move from testing to estimation. Recall that it is rarely the case that we would be satisfied with a statement like “given the data at hand, we can reject the null hypothesis that treatment and control have the same impact on the studied outcome” (that Vioxx and naproxen are not the same in terms of heart attacks, or that visitors don’t respond equally to Tabs versus Lists). Instead we want to know about the magnitude of the difference. We are interested in an estimate and some sense of its accuracy. This lab anticipates our discussion about these topics on Wednesday.

1. Getting started

We will first load up a couple functions that we’ll need for this lab.

```
source("http://www.stat.ucla.edu/~cocteau/stat13/data/pivot.R")  
  
ls()
```

You should see a function called `simtable`. It provides you with a simple tool for conducting a simulated experiment, and storing the results in a two-by-two table. As we mentioned in class, we are returning to tables briefly (only briefly) as we introduce the main ideas behind so-called sampling distributions. This lab will provide you with a simulation-based understanding of what’s going on.

Suppose, for example, that we want to compare a new treatment (an A/B test or perhaps a drug trial) against a known control, measuring a simple yes/no outcome (a patient gets better or not, a visitor clicks on some part of the web site or not, etc.). As we have seen in class, a common null hypothesis is that there is no difference between treatment and control in terms of the outcome variable; subjects are just as likely to exhibit the outcome under either treatment or control.

Recall that there are two kinds of errors we can make when testing a hypothesis; the first is that we reject a null hypothesis that’s true (Type I) and the second is

that we fail to reject a null hypothesis that's false (Type II). We control Type I errors by setting our significance level, α . If we set $\alpha = 0.05$, for example, we will be making Type I errors about 5% of the time. The Type II error cannot be set as easily. Type II errors depend on the magnitude of the difference between treatment and control, as well as on the sample size (the number of subjects we enroll in our study). Broadly, the bigger the difference and the larger the sample size, the less likely we are to make a Type II error (and say that there is no difference between treatment and control when there is).

To study this a bit more closely, we are going to examine how power depends on these two factors (magnitude of the difference and sample size). The function `simtable` will be our computational workhorse here.

2. Simulating an experiment

Suppose we have a large population from which we can sample our study participants. Within this population, a proportion p_c would exhibit the outcome we're looking for if given the control; and a proportion p_t would exhibit the outcome if given the treatment. Our experiment will consist of selecting a total of $n_t + n_c$ subjects at random from the population and assigning n_t to treatment and n_c to control. We then conduct the experiment and create a table of the number of subjects who exhibited the outcome or not, broken down by treatment and control.

(Many of the studies we have considered use this simple random sampling mechanism: the Vioxx trials, A/B web site testing. And this design gives us the clearest mental picture of "repeating" an experiment: We simply select another group of subjects at random from the population. Of course, there are lots of ways to form a study group other than random sampling. Recall that Hill's trial enrolled patients as they walked into the clinic. The mechanism for enrollment is not the most important thing for this lab; instead, no matter how we select patients, we need to be able to come up with a guess p_t for the chance that someone in the treatment group will exhibit the outcome, and p_c the chance that someone in the control group will. Again, it is much easier to imagine repeating an experiment if you're just selecting another random sample of participants, so we'll use that framework for the rest of the lab.)

The function `simtable` will do this for you. Suppose, for example, that we know from our experience with the control, that $p_c = 0.5$; that is, about half of the people who receive the control exhibit the outcome we're looking for. Next, given pilot studies or some other kind of advanced research, we might also expect that $p_t = 0.8$; that is, 80% of people receiving the treatment will exhibit the desired outcome. This, of course, is only a guess and the purpose of our study is to estimate the actual effect of the treatment. Finally, suppose, because of expense or administrative overhead, we can have only 150 people in the control group and 100 in the treatment group.

We would then call `simtable` with the following

```
results = simtable(100,150,0.8,0.5)

results
```

What you should see is a two-by-two table with the rows 1/0 indicating the absence or presence of the desired outcome, respectively; and the columns referring to treatment and control. Repeat the above commands a few times and see how variable the results are. Again, each time we call this command, we recruit another group of subjects from our population; with each group we will get slightly different experimental results.

Recall that if the probability of a Type II error is denoted β , then the power of a test, the chance that we correctly reject a false null hypothesis is $1-\beta$. To see what power our re-randomization test has against this difference ($p_t = 0.8$ versus $p_c = 0.5$), we can perform the test on a number of simulated tables (simulated using the conditions we described) and see how frequently we reject the null hypothesis.

```
test = fisher.test(results)

test$p.value
```

This will give you the P-value for the re-randomization test. (In lecture we mentioned that `rerandomize.test` from Lab 3 was just a call to `fisher.test`, but that we chose to hide the details from you because we hadn't mentioned odds ratios and such yet.).

What P-value did you get? Would you reject the null hypothesis at the 0.05 level? Let's now repeat this process, say, 1,000 times.

```

nsim = 1000
pvalues = rep(0,nsim)

for(i in 1:nsim)
{
  results = simtable(100,150,0.8,0.5)
  test = fisher.test(results)
  pvalues[i] = test$p.value
}

```

This piece of code starts by setting the number of simulations, `nsim`, and then creates a vector of that length (all zeroes) to hold the results (have a look at `pvalues` by typing its name and hitting enter). Then, the line that starts `for(...)` is the beginning of a loop. It says you repeat the commands in the curly braces `{}` for `nsim` times, each time assigning a new value to `i`. The first time through the loop, `i = 1`, the second time `i = 2` and the last time, `i = 1000`. For each iteration, we are generating a new table and conducting a new test and saving the P-value in the corresponding entry of our `pvalues` vector. In the end `pvalues` will have 1,000 P-values, `pvalues[1]` having the number associated with the first iteration, `pvalues[2]` having the number from the second iteration and so on.

We can then see how many trials would have us reject the null hypothesis at the 0.05 level. That is,

```

sum(pvalues<0.05)

mean(pvalues<0.05)

```

The first will give the the number of P-values (out of 1,000) that are less than 0.05, and the second expression will give you the proportion (`pvalues < 0.05` is 1,000 TRUE's and FALSE's, where TRUE becomes 1 and FALSE is 0). The proportion is then an estimate of the probability that we correctly reject the null hypothesis (since we are assuming $p_t = 0.8$ and $p_c = 0.5$ the null is obviously false). In other words, this proportion is the power of the test, $1 - \beta$.

Note that in this case, our power is VERY high. We are virtually certain to detect that the null hypothesis is false.

Question 1: Suppose the treatment effect is 60% and not 80%, but that the control effect is still 50%. Starting with an overall sample size of 200 (100 in treatment and 100 in control), estimate the power to detect this departure from the null. Then, try overall sample sizes of 500, and 800 (again, 500 means 250 in treatment and 250 in control) and see what happens to your power. Does it increase? It is typical to want an experiment with power about 80%. If that's the case, which of these sample sizes would you recommend?

R can perform this repetition for you using a function called `fe.ssize`. (This stands for Fisher's exact test sample size.) It will essentially iterate through the sample size for two values of p_t and p_c . It belongs to an R library called `clinfun`. The functionality of R is extended by people authoring libraries of functions and data and sharing them. In the lab, you can type

```
library(clinfun)
fe.ssize(0.8,0.5)
```

The output returns an estimate (CPS) and an exact value. They should typically be close. For the case of 0.5 and 0.8, the number of subjects you need is quite small (less than 100).

If you are using your own computer, you will need to install the `clinfun` library. This is done by typing

```
install.packages("clinfun")
```

You will be prompted to select a computer near you (scroll down the list and select a machine in California, say) and then let R do its thing. You will then be able to type the two commands above.

Question 2: Evaluate the sample size you need for a treatment probability of 0.6 and a control probability of 0.5. How does this compare to what you found in your simulations? Now, pick another pair of values and see how the sample size changes with the difference between p_t and p_c .

2. Odds Ratios

Recall from lecture that a reasonable measure of the difference in outcomes between treatment and control is the odds ratio. Given a table

	control	treatment
1	a	b
0	c	d

the odds of someone in the control group exhibiting the outcome is a/c , and for the treatment group d/b . The odds ratio is then $(a/c)/(b/d) = ad/bc$.

```
results = simtable(100,150,0.8,0.5)

oddsr = results[1,1]*results[2,2]/(results[2,1]*results[1,2])

oddsr
```

Repeat these lines a few times to get a sense of how the odds ratio varies. Recall, that we can compute the odds ratio for the population; that is, if the odds of seeing the outcome from the treatment group is $0.8/(1-0.8) = 4$, while the odds are even in the control group, $0.5/(1-0.5) = 1$, the odds ratio for the population is $4/1 = 4$.

We are now going to make use of the very same simulation setup used in the previous section, this time computing odds ratios in each iteration.

```
nsim = 1000
oddsratios = rep(0,nsim)

for(i in 1:nsim)
{
  results = simtable(100,150,0.8,0.5)
  oddsratios[i] = results[1,1]*results[2,2]/(results[2,1]*results[1,2])
}
```

We can have a look at the results with a simple histogram.

```
hist(oddsratios)

abline(v=4)
```

The second command adds a vertical line at the point $x = 4$ on the graph. The histogram is an estimate of the so-called sampling distribution for the odds ratio. This distribution captures the variation that comes from repeating our experiment. With each of the 1,000 repetitions, we can imagine selecting a different set of people from our population; some from the treatment group will respond, and some from the control group will respond. With each experiment we will have different numbers of people responding, giving rise to a different odds ratio. The histogram shows how variable our results are.

Question 3: Suppose the treatment effect is 60% and not 80%, but that the control effect is still 50%. Starting with an overall sample size of 200 (100 in treatment and 100 in control), have a look at the histogram of odds ratios. Then, try overall sample sizes of 500, 1,000, 2,000 (again, 500 means 250 in treatment and 250 in control) and see what happens to your histograms. What do you notice? Comment on the shape, the center and the spread of the resulting distributions.