# Statistics 13, Lab 3 (Part 1)

## Hypothesis testing

### 1. Getting started

On the last page of this document you will find a comparison between the "treatment" and "control" for a simple A/B test carried out by the New York Times. Recall from lecture that A/B testing is a kind of randomized trial in which visitors are divided into two groups. As members of the control group browse the web site, they are shown the site's typical design, the pages are unchanged. The treatment group, on the other hand, sees an experimental version of the site, with pages typically containing one or more changes. Our data for this lab come from one such test. If you look carefully at the two images on the last page, you will see that the treatment for this experiment involved removing a navigation bar from the middle of the page heading. The experiment was restricted to pages in the Movies Section (and not the other sections like Science or Health or Opinion) and took place between February 7 and March 10 in 2008.

In broad strokes, A/B testing is similar to the randomized controlled trials we spent so much time on in lecture. People are assigned to receive the treatment or control at random. For A/B testing, however, the stakes are a lot lower: no one dies in an A/B test, no one gets sick, no one has a heart attack. Instead, the experimenters record various facts about the visit, looking to see if the change has induced certain behaviors; perhaps causing visitors to spend more time on the site, to read more pages, or to spend more money. This test had an added behavior of interest: The experimenters had reason to believe that under the treatment, the page without the navigation bar, visitors will be forced to make greater use the search box. To summarize, the experiment randomized visitors into two groups, the treatment and a control. The question of interest: Do visitors receiving the treatment use the search box more often than visitors reveiving the control? To answer this, you will look at 1,000 visitors who were new to the site, visitors that had never been to `nytimes.com` before. Remember from Lecture 6 that this determination is based on so-called cookie data stored by the visitor's browser. The details are not particularly important here; what is important is that these 1,000 visitors had never seen `nytimes.com` much less the Movies Section before. Once they arrived, they were randomized to receive either the treatment or the control.

As with your previous four labs, we begin this one by loading data into your R session. After starting RStudio, enter the following command.

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

You should see objects in your workspace that include `simtrials` and `movies`. We'll focus on `movies` for now; it contains the results from the A/B test, each row corresponding to a different visitor.

```
dim(movies)

names(movies)
```

There are 1,000 rows in this data set, each being a different visit (by a different visitor). The variables are `id`, the visitor's ID (set via a cookie – see Lecture 6); `treatment`, whether the visitor recieved pages that were unchanged or were missing a navigation bar; `count`, the number of times this visitor clicked on the "go" button in the search box, initiating a search; `outcome` a 0/1 variable that is 0 if the visitor did not use the search box and 1 if they did at least once; `day` indicating the day the visit took place, an integer where 0 means the first day of the study and 31 means the last; and `hour`, the hour of the day the visit took place, with 0 being midnight and 12 being noon.

You can have a look at these variables with commands like

```
head(movies$treatment)

table(movies$treatment)
```

(the first displaying the first six of the `treatment` values, and the second tabulating all 1,000 of them for you). Repeat this for `outcome` and make sure your data match what we reported in our introductory paragraphs above.

## 2. Re-randomization and hypothesis testing

The study was designed to examine if removing the navigation bar at the top of the page led to increased use of the search box. Let's cast this into the hypothesis testing framework we have developed in lecture.

*Question 1: Given the description of the experiment and the aims of the experimenters (and keeping in mind the work we did on clinical trials) state the null hypothesis that they should employ when analyzing their results.*

Previous experimentation with `nytimes.com` and other sites strongly suggested that removing the navigation bar should not decrease usage of the search box and so the researchers wanted an alternative hypothesis that the number of visitors using the search box is larger under the new design that is missing the navigation bar. Following our discussion in class, our test statistic will be the number of visitors in the "no bar" group that used the search bar. Given our alternative hypothesis, we will be looking to see if the value of this test statistic observed in our experimental data is larger than what we would expect under the null hypothesis. Our significance level will be 0.05 because that is the standard for A/B tests and web site designers will be expecting this choice.

With that setup, let's have a look at our experimental results. You can make our standard table from lecture with the command

```
table(movies$outcome,movies$treatment)
```

*Question 2: Compute the conditional proportion of visitors that used the search box with the new design missing the search bar. Do the same for the control group seeing the usual Movies page. Does this study suggest people receiving the new design are more or less likely to use the search box?*

As we did in class, we will now consider re-randomizing the trial to see whether this result could have occurred by chance. Again, our test statistic is the number of people who were shown the “no bar” option and used the search box. Under the null hypothesis that the two versions of the web site have the same effect on visitors’ use of the search box, the 77 visitors who did in fact perform a search would have done so no matter which design they were shown. The fact that 52 visitors shown the “no bar” version conducted a search was merely because randomization had arranged for 52 of the 77 searchers to be shown this version. This is an expression of the null hypothesis. We will use re-randomization to see if 52 is a typical value under the null.

Toward this end, suppose we assign our table of outcomes and treatments to a new variable called `results.` We can then extract this number using the subsetting rules we learned last time.

```
results <- table(movies$outcome,movies$treatment)
```

```
results[2,2]
```

Here we asked for the element in the second row and second column of the table; the number of visitors shown the “no bar” design who searched the site. Check that this number agrees with our previous discussion (that it’s 52).

To create a new assignment of people into treatment groups, we can use the command `sample`. As its name suggests, it takes data you give it and reorders the values randomly.

```
simtreatment <- sample(movies$treatment)
```

```
head(simtreatment)
```

This should give you a random re-ordering of the treatment assignments. You can imagine R placing the data in `movies$treatment` into a hat (496 tickets with the word “unchanged” on them, and 504 with the words “no bar” on them), mixing them up and drawing them one at a time. The first draw being the first value in `simtreatment`, the second draw being its second value and so on. The variable `simtreatment` should again contain 496+504=1,000 values. Technically, the object `simtreatment` is a vector, an ordered collection of values (in this case 1,000 assignments into `unchanged` or `no bar`). Whereas `movies` is a data frame and has two dimensions (both rows and columns), `simtreatment` has only length.

```
length(simtreatment)
```

Notice that we have seen vectors before. When we extract data from a data frame (as with `movies$outcome`), the data are returned to us in this way.

Finally, you can see that `simtreatment` has the right numbers with

```
table(simtreatment)
```

With this as a new division into treatment and control, you can then see how many visitors who searched the site were re-randomized into each group

```
results <- table(movies$outcome,simtreatment)
```

```
results[2,2]
```

Do this a few times, meaning use `sample` to create a new `simtreatment` and then form a new `results` table and look at `results[2,2]`. (Remember the "up arrow" key is your friend when you want to repeat previous commands in R.) What do you see? The other object you loaded in your workspace is called `simtrials`. It will basically repeat this process as many times as you like. Here we call the function for 10,000 re-randomizations (this will take a few seconds depending on your computer – in the next lecture we'll see a formula that will make actually re-randomizing unnecessary for smallish-sized tables).

```
out <- simtrials(movies$outcome,movies$treatment,10000)
```

```
out
```

The function or command `simtrials` takes three arguments (and the order you provide them is important): the first is the outcome variable from your trial, the second is the original randomization into two groups, and the last is number of times you'd like to have R re-randomize. The result which we stored in a new variable `out`. Like `simtreatment`, the object `out` is just an ordered collection of numbers, a vector. In this case it contains 10,000 numbers, each one corresponding to the count of visitors in a (re-randomized) "no bar" group that used the search bar.

```
length(out)
```

```
head(out)
```

(As an aside, you can have a look at the command `simtrials` by typing its name and hitting enter.

```
simtrials
```

In this case you will see some R code that looks like the code we typed above. The only real difference is that there is a `for(){ ... }` loop that takes our couple of commands and repeats them n times. One of the interesting things about R is that many of its commands are written in R itself. Here, we took the lines that we typed out as we worked with our data and pulled them into a new command, extending R. We will have an extra – and voluntary – lab session devoted to more advanced work with R later in the quarter.)

We can "see" the 10,000 values in the variable `out` by making a table.

```
table(out)/10000
```

or a barplot

```
barplot(table(out)/10000)
```

Unlike in the last lab, here we have divided the table output by 10,000 to get the proportion (our simulation re-randomized the tables 10,000 times and so `out` has 10,000 entries).

*Question 3: In the re-randomization process, what range of values are possible for the number of visitors in* *the "no bar" group that searched? In other words, how small could that number be? How big? What values did you see in your simulation? Is there a difference and why? Finally, what shape does the display of the re-randomized values in* `out` *have? Is it symmetric? Nearly so? Where is it centered (use the median)? How does our observed value of the test statistic (52) compare to this distribution? Informally, what can you say about the reasonableness of the null hypothesis?*

*Question 4: To compute Fisher's P-value associated with this experiment, we look at the proportion of tables that place 52 or more visitors who searched into the "no bar" group. Using the output from*

```
    table(out)/10000
```

*estimate the P-value using your re-randomizations. Recall that we initially selected a significance level of 0.05. Can we reject the null hypothesis of no difference between the two designs? Repeat the simulation with another 10,000 re-randomizations. Do your conclusions change?*

*Bonus: We only looked at two variables in the data set* `movies`. *Use what you learned in Lab 2 to report on some of the other variables. Are they qualitative or quantitative? What do they "look" like? Compute some summaries (numerical or graphical) and report on what you see.*

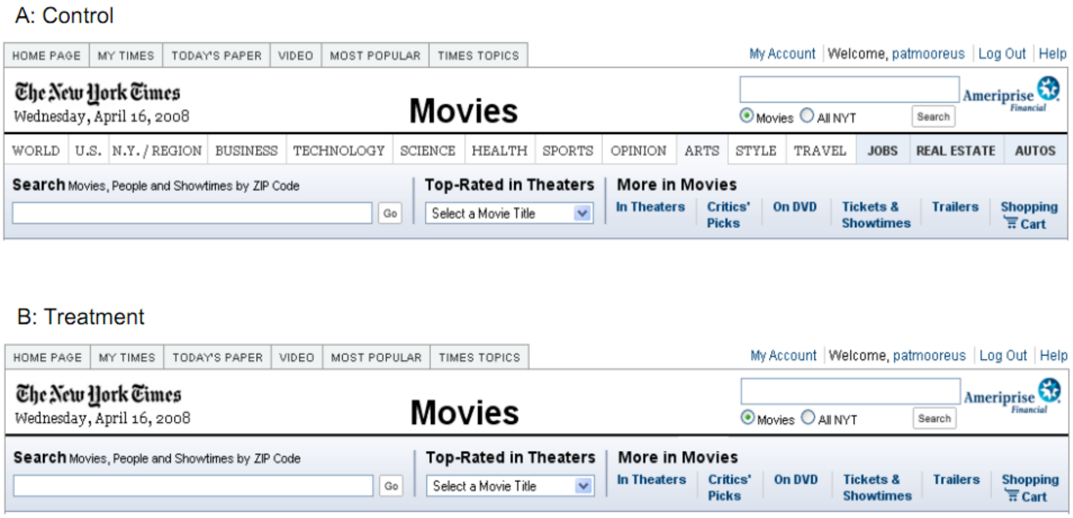

Figure 1: The control `unchanged` and treatment `no bar`.