Lecture 2

Intuitive p-values

After you have designed and appropriate experiment and collected data, you report a p-value that is suppose to convey how significant your results are. The interpretation of the p-value depends on what you are testing. For now, let us consider a simple example:

Example 1 Suppose that in psychology self-esteem experiment of the last lecture, fourteen students were randomly assigned to two groups, a treatment group that provided positive self esteem reinforcement and a control group that provided some study hints without the self-esteem component. The results of a quiz after the treatment are recorded as follows:

treatment	control
10	5
15	9
16	10
16	10
17	14
19	18
20	20

Is the self-esteem treatment effective? Can we quantify how sure we are that it is effective? The way you might have seen this question posed in a previous course is:

$$H_0 : \mu_{treatment} = \mu_{control} \quad \text{versus} \quad H_a : \mu_{treatment} > \mu_{control}$$

Such symbolic mumbo-jumbo will NOT be used in your reports or in most of this text. The equivalent question/statement in plain English is: Did the self-esteem treatment improve quiz scores? The null hypothesis that the self-esteem treatment had no effect is implied and the one-sided alternative hypothesis that the self-esteem treatment improved quiz scores is made explicit.

If there is no improvement in the sample we have collected, then the answer to the research question is obvious: the self-esteem treatment does not appear to improve quiz scores. But there is an improvement in the sample we have collected. The mean score in the self-esteem treatment group is 3.85 points higher than the mean score of the control group. We would like to quantify this improvement and say something about the probability of seeing such an improvement or a larger one IF there were actually no difference between the two groups. If the probability we would get a sample showing as large or a larger impact of the self esteem treatment given no real impact is low, then we are compelled to conclude that the self esteem treatment actually did have an impact. If the probability of seeing as large of an impact as we see in our data simply due to random chance is high, then we have no real evidence that the self esteem treatment worked - the improvement we saw might simply be due to chance. ■

Chance is relative. At some point, Sir Ronald Fisher decided that one out of twenty (5%) was rare and that has been adopted as the standard cut-off for significance for most purposes ever since. However, would you be comfortable with a 5% chance of dying today? Would you consider that rare enough to be acceptable? What about a 5% error rate on a drug test used to screen you for employment? Is a 5% chance of error in the trajectory range of a comet or an airplane acceptable? Perhaps a 5% chance of error in the prediction of the salary range for boys based on personal factors is a reasonable error for a sociology journal, a 5% chance of error in the prediction of fig leaf size based on fertilizer treatment is a reasonable error in agriculture, etc... But certain problems call for a more stringent criterion than 5% and certain problems call for a less stringent criterion. As a purely hypothetical example of the latter, say that you are looking for an association between mercury in flu vaccines and autism. The probability of getting the association you have in your data or a stronger association is 8%. That isn't statistically significant according to Fisher's cut-off but do you think it constitutes proof that mercury in flu vaccines is safe? Is reporting "no statistically significant association" a responsible thing to do in this circumstance? Does your answer depend on whether the study involved 10, 100, 1000, or 10,000 children? Why and how?

Bootstrapping

We are going to assess the significance of the self-esteem experiment using simulation, also called bootstrapping. To do this, we put all the scores from both the treatment and control groups into a single pot or urn. Mathematicians, especially probabilists, are often drawing balls from urns and bootstrapping is a practical application of this idea. We sample from the urn with replacement (bootstrapping) or without replacement (called a permutation test if we, in fact, go through all possible allocations of scores to treatment and control groups once) in order to create a new set of treatment and control scores. We then re-evaluate our chosen test statistic for this new data and we repeat this process of sampling and re-evaluating many times (usually, at least 1000). We consider the proportion of test statistics from the (1000) re-sampled data sets that were more extreme that the original test statistic from the original data set. This proportion is an estimate of how likely it was that we would have gotten our original test statistic or something more extreme by chance alone.

Software

Statistics for the masses is made feasible by the existence of good software packages. Below is a small subset of the possible choices for software for this course:

- **SAS**: SAS is a well-maintained, established, software program with a team of people maintaining it and writing new algorithms for it. It often requires writing short pieces of code; it is expensive to rent; one is not allowed to own a copy ever. Further, it can provide so much information about an analysis that it is hard to tell what information is relevant to your situation. It is used by the government agencies, some businesses, and some academics. It is not completely platform independent although now MACs allow directly running PC programs.
- **R**: R is free, platform independent, well-maintained but by volunteers, have many donated algorithms but these are vetted mainly by the academic community rather than by a dedicated company team.

- **SPSS**: Parts of SPSS can be owned rather than rented and additional modules for advanced techniques can also be bought or rented. It has a nice user-interface but I personally found its programming language to be completely non-intuitive.
- **Minitab**: Minitab is a solid, reliable, relatively inexpensive, easy-to-use statistical software package that you can rent or own. It has a reasonable programming language, good graphics, and a reasonable array of statistical tests and graphical techniques. It is ideal for students who have little to no programming experience and desire user-friendly, fill-in menus. This is the software that I will use for this class.

Example 2 Bootstrapping the self-esteem data using Minitab:

For this example, we will use the naive test statistic that is the raw difference in means of the two groups. The outline for our code is the following:

1. Pool the data
2. Randomly assign the data to treatment and control (1000 times)
3. Calculate the difference for each random assignment
4. Determining the proportion of times the difference obtained by random assignment exceeds the actual difference obtained by the data (3.85)

To do any procedure 1000 times, it is useful to be able to write a little code. You just need to be able to repeat what you can do once. To see the written commands, click on the session window, go to Editor and click on Enable Commands. To do this procedure once using the pre-programmed menu items, we:

1. Stack the data and the labels (Data > Stack > Columns)
2. Randomize the data (Calc > Random Data > Sample from Columns)
3. Calculate the difference of the randomized data (Calc > Calculator)

The Minitab code is the following:

```
MTB > Stack 'Treatment' 'Control' c3;
SUBC>   Subscripts c4;
SUBC>   UseNames.
MTB > Sample 7 C3 c5;
SUBC > Replace.
MTB > Sample 7 C3 c6;
SUBC > Replace.
MTB > let c8(1) = mean(c5) - mean(c6)
```

Now we have the basis for writing a Macro (saved as Randomize.MAC in the folder that contains the Minitab file with the data):

```
GMACRO
Randomize

Do k1 = 1:10

Sample 7 c3 c5;
Replace.
Sample 7 c3 c6;
Replace.
let c8(k1) = mean(c5) - mean(c6)

ENDDO

ENDMACRO
```

Advice: Until you debug your macros, keep your DO loops small. You must save your Minitab project and your Minitab macro in the same folder.

Your run the macro by issuing the command:

```
MTB >  %Randomize
```

Let's do it with 1000 (if it works with 100). Then we sort the difference we get by randomizing the data, count the number greater than or equal to 3.85, and figure out the proportion of times randomizing the results would give us a difference greater than the one we got with the actual data. Sorting in Minitab is done with the Data > Sort menu. This is one way of obtaining an intuitive p-value. ■

Exercises for Lecture 2

1. Go through the above exercise in Minitab yourself so that you learn the menu commands and verify you can do the programmed bootstrap.
2. Modify the above code so that instead of randomizing the subscripts (Treatment and Control) you randomize the list of numbers instead.
3. Modify the above code so that the code itself does the sorting.
4. Modify the above code so that not only does it do the sorting but it also provide the p-value. (This is harder and might require you searching the HELP menu to find the proper code and commands).