Nonparametric Lab
Data Science Team
June 1st, 2018

Introduction

In this lab, we will focus on the two-sample permutation test. We will look at two cloud seeding examples; let’s start from a simple example and then look a larger dataset.

Scientists are interested in whether humans can use chemicals to let clouds produce more rainfall. In one study, researchers in Florida explored whether injecting silver iodide into cumulus clouds would lead to increased rainfall. On each day that was judged to be suitable for cloud seeding, a target cloud was identified and a plane flew through the target cloud in order to seed it.

Randomization was used to determine whether or not to load a seeding mechanism and seed the target cloud with silver iodide on that day. Radar was used to measure the volume of rainfall from the selected cloud during the next 24 hours. The volume of rain was measured in volume units of acre-feet, the “height” of rain across 1 acre.” (Rossman and Chance, 2006)

Let us start with a simple example.

Suppose scientists have conducted the experiment for only three days. With such a small sample size, we don’t necessarily need to use R. Among the three days, in two of them, the clouds were seeded and on the other day, the cloud was unseeded.

Below are the recorded rainfalls.

Day1 unseeded 147.8
Day2 seeded 489.1
Day3 seeded 119.0

The null hypothesis $H_0$ is that: injecting silver iodide into cumulus clouds would have no effect on the amount of rainfall we get.

Let us use the difference in means between the two groups as our testing statistic and think about the following questions:

- what is the difference that we observe in means between the two groups? Do your calculation as seeded - unseeded.
- under $H_0$, is a permutation test appropriate? In other words, can labels/groups be randomly assigned to different research subjects?
- how many possible cloud assignments would result in two seeded clouds and one unseeded cloud?
- calculate the difference in means between the two groups for each assignment you listed in the previous step. make sure your calculation is performed as seeded - unseeded.

Remember the p-value represents the probability of “something as or more extreme than what we observe” under the assumption that $H_0$ is true.

- based on all the mean differences you calculated in the permutations, what is the p-value? after you get the p-value, what is your conclusion to our testing problem?
- how many different p-values we can possibly get in this scenario? that’s how much a dataset of size three can do.
The above example is a simple permutation test example and the sample size is so small that we can carry it out by hand. What if we have 10 days of observations?

## A larger dataset

Let us load the data from `cloudseeding`, which contains 10 days data. We create vectors `rainfall` and `treatment` separately to ease our work later.

```r
pacman::p_load(gtools, dplyr)

cloudseeding <- read.csv("http://web.stanford.edu/class/stats101/nonparametric/cloudseeding.txt", sep="", header = TRUE)

cloudseeding

<table>
<thead>
<tr>
<th>rainfall</th>
<th>treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>147.8</td>
<td>unseeded</td>
</tr>
<tr>
<td>26.1</td>
<td>unseeded</td>
</tr>
<tr>
<td>95.0</td>
<td>unseeded</td>
</tr>
<tr>
<td>1.0</td>
<td>unseeded</td>
</tr>
<tr>
<td>4.1</td>
<td>seeded</td>
</tr>
<tr>
<td>119.0</td>
<td>seeded</td>
</tr>
<tr>
<td>489.1</td>
<td>seeded</td>
</tr>
<tr>
<td>978.0</td>
<td>seeded</td>
</tr>
<tr>
<td>255.0</td>
<td>seeded</td>
</tr>
<tr>
<td>2745.6</td>
<td>seeded</td>
</tr>
</tbody>
</table>
```

Here is an approach.

```r
rainfall <- cloudseeding$rainfall
trtmt <- cloudseeding$treatment
teststat.obs <- mean(rainfall[trtmt == "seeded"] - mean(rainfall[trtmt != "seeded"])

[1] 697.6583
```

- how many different assignments of seeded/unseeded are possible? Let us list all possible assignments of the rainfalls to the two groups, with four unseeded clouds and six seeded clouds.

The library `gtools` has a handy function for enumerating combinations. Combinations are permutations where the ordering does not matter; this can be seen by using `apply` to sort each row (i.e., by setting `MARGIN = 1`) and retaining only the unique entries.

```r
require(gtools)
combinations(5, 4)

[1,] 1  2  3  4
[2,] 1  2  3  5
[3,] 1  2  4  5
[4,] 1  3  4  5
[5,] 2  3  4  5
dim(permutations(5, 4))

[1] 120  4
unique(t(apply(permutations(5, 4), MARGIN = 1, sort)))
```
Now we can calculate the difference in means between the groups for each possible assignment of seeded/unseeded clouds. Recall

```
table(trtmt)
```

```
trtmt
  seeded unseeded
  6      4
```

```
choose(10, 4)
```

```
[1] 210
```

Let’s call the number of treated units \( K \) and generate all possible combinations.

```
N <- nrow(cloudseeding)
K <- sum(trtmt == "seeded")
combos <- combinations(N, K)
```

We can then use those combinations as indices that counterfactually assign each observation to either treated or untreated. Recall negative subscripts are efficient way to drop elements from vectors.

```
letters[-c(4:26)]
```

```
[1] "a" "b" "c"
```

```
Ncombos <- choose(N, K)
teststats.sim <- matrix(nrow=Ncombos)
for(i in 1:Ncombos){
  teststats.sim[i] <- mean(rainfall[combos[i,]]) - mean(rainfall[-combos[i,]])
}
head(teststats.sim)
```

```
[1,] 1 2 3 4
[2,] 1 2 3 5
[3,] 1 2 4 5
[4,] 1 3 4 5
[5,] 2 3 4 5
```

To calculate p-value, we calculate the proportion of possible assignments that have a difference in means that is greater than or equal to the observed difference in means.

```
head(teststats.sim >= teststat.obs)
```

```
[,1]
[1,] FALSE
[2,] FALSE
[3,] FALSE
[4,] FALSE
[5,] FALSE
[6,] FALSE
mean(teststats.sim >= teststat.obs)

[1] 0.03809524

what's your conclusion?

**Load the Data**

```r
movies <- read.csv("http://s3.amazonaws.com/dcwoods2717/movies.csv", stringsAsFactors = FALSE)
glimpse(movies)
```

Observations: 2,961  
Variables: 11  

- `$title` <chr> "Over the Hill to the Poorhouse", "The Bro...  
- `$genre` <chr> "Crime", "Musical", "Comedy", "Comedy", "C...  
- `$director` <chr> "Harry F. Millarde", "Harry Beaumont", "Ll...  
- `$year` <int> 1920, 1929, 1933, 1935, 1936, 1937, 1939, 14...  
- `$duration` <int> 110, 100, 89, 81, 83, 102, 226, 88, 14...  
- `$gross` <int> 3000000, 2808000, 2300000, 3000000, 163245...  
- `$budget` <int> 100000, 379000, 439000, 609000, 1500000, 2...  
- `$cast_facebook_likes` <int> 4, 109, 995, 824, 352, 229, 2509, 1862, 11...  
- `$votes` <int> 5, 4546, 7921, 13269, 143086, 133348, 2918...  
- `$reviews` <int> 2, 107, 162, 164, 331, 349, 746, 863, 252,...  
- `$rating` <dbl> 4.8, 6.3, 7.7, 7.8, 8.6, 7.7, 8.1, 8.2, 7....

Choose Two Similar Directors with < 20 films total.

Here is some code for viewing the most popular directors.

```r
sort(table(movies$director), decreasing = TRUE)[1:10]
```

<table>
<thead>
<tr>
<th>Director</th>
<th>Films</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steven Spielberg</td>
<td>23</td>
</tr>
<tr>
<td>Clint Eastwood</td>
<td>19</td>
</tr>
<tr>
<td>Martin Scorsese</td>
<td>16</td>
</tr>
<tr>
<td>Tim Burton</td>
<td>16</td>
</tr>
<tr>
<td>Spike Lee</td>
<td>15</td>
</tr>
<tr>
<td>Steven Soderbergh</td>
<td>15</td>
</tr>
<tr>
<td>Woody Allen</td>
<td>15</td>
</tr>
<tr>
<td>Renny Harlin</td>
<td>14</td>
</tr>
<tr>
<td>Ridley Scott</td>
<td>14</td>
</tr>
<tr>
<td>Barry Levinson</td>
<td>13</td>
</tr>
</tbody>
</table>

Pick two directors that make similar kinds of movies with less than 20 movies between them.

**Careful:** Suppose you pick Steven Spielberg and Clint Eastwood, who made 23 and 19 movies, respectively, in the dataset. `choose(42, 23)` is 446,775,310,800–that will take too long!

1. Make a Hypothesis

Make a hypothesis about one of the quantitative variables (such as `$gross` or `$rating`...). Who is more popular? Profitable? Note you may want to avoid `$rating` or other variables that may have repeat values...

Create a new dataset with just those directors.

```r
d <- filter(movies, movies$director %in% c("Francis Ford Coppola", "Oliver Stone"))
glimpse(d)
```
Observations: 19
Variables: 11

$ genre <chr> "Crime", "Crime", "Drama", "Crime", "Crime...  
$ director <chr> "Francis Ford Coppola", "Francis Ford Copp...  
$ duration <int> 175, 220, 289, 114, 123, 103, 126, 110, 14...
$ gross <int> 134821952, 57300000, 78800000, 25600000, 2...
$ budget <int> 6000000, 13000000, 31500000, 10000000, 580...
$ cast_facebook_likes <int> 28122, 39960, 25313, 12097, 18793, 14672, ...
$ votes <int> 1155770, 790926, 450676, 57363, 12771, 271...
$ reviews <int> 2446, 799, 1244, 368, 120, 140, 333, 118, ...
$ rating <dbl> 9.2, 9.0, 8.5, 7.2, 6.5, 6.3, 7.4, 7.3, 7....

Following “Fisher’s Argument” in nonparametric-lecture02...

Suppose \( W \) is your statistic. Your code may look something like

\[
\begin{align*}
N & \leftarrow \text{nrow}(d) \\
K & \leftarrow \text{sum}(d$director == "Francis Ford Coppola") \\
\text{combos} & \leftarrow \text{combinations}(N, K) \\
\text{likes} & \leftarrow d$cast\_facebook\_likes \\
W.\text{obs} & \leftarrow \text{mean}(\text{likes}[d$director == "Francis Ford Coppola"] \- \text{mean}(\text{likes}[d$director != "Francis Ford Coppola"])) \\
W.\text{sims} & \leftarrow \text{matrix}(\text{nrow}=N) \\
\text{for}(i \text{ in } 1:N) \{ \\
& \hspace{1cm} W.\text{sims}[i] \leftarrow \text{mean}(\text{likes}[\text{combos}[i,]]) \- \text{mean}(\text{likes}[-\text{combos}[i,]]) \\
\} \\
p & \leftarrow \text{mean}(W.\text{sims} >= W.\text{obs}) \\
\end{align*}
\]

[1] 0

**Questions**: What is the observed sample differences in mean for your research question? Should we accept that observed difference as statistically significant? Consider \text{summary}(\text{likes}[d$director == "Francis Ford Coppola"]\) and \text{summary}(\text{mean}(\text{likes}[d$director != "Francis Ford Coppola"]))—under what conditions (in terms of min and max) would we expect a \( p \) value of 0 or 1?

2. Wilcoxon Rank-Sum Test: two-sample test

Repeat your analysis using Wilcoxon Rank-Sum test.

The Wilcoxon rank-sum test is a permutation test that uses the sum of the ranks as a test statistic. Let us first review the definition of rank. Let \( X_1, \cdots, X_n \) the dataset of \( n \) observations. The rank of \( X_i \) among the \( n \) observations is given by

\[
\text{rank} (X_i) = \text{the number of } X_j \text{'s } \leq X_i.
\]

\[
\text{rank(c(1, 0, 4, 3, 4, 5, 11, 6, 7, 9))}
\]

[1] 2.0 1.0 4.5 3.0 4.5 6.0 10.0 7.0 8.0 9.0

The test statistic for the Wilcoxon rank-sum test is calculated as the following: we will rank all values with the two groups combined, then calculate \( W_{\text{obs}} \), our test statistic, as the sum of the ranks in group 1.
• Is it the same (or at least statistically indistinguishable) if we decide to use the sum of the ranks in group 2 as our test statistic?

• Use permutation test to calculate p-value and state your conclusion. In other words, find all the permutations and calculate $W$ for each assignment. Do your findings change?

3. Non Parametric Association

Construct a new data set. For example...

```r
crimeMovies <- filter(movies, genre == "Crime", year > 2000)
```

Use the remaining time to compute non-parametric measures of association (Spearman’s, Kendall’s...). What do these associations show? How do they compare and contrast with parametric measures?