

Random and Systematic Randomization Tests

Joseph Free

Department of Mathematics, University of North Florida

March 24, 2020

Abstract

Randomization tests are nonparametric methods used to test hypotheses regarding experimental treatments in the absence of common assumptions such as random sampling or homogeneity of variances. In this report we consider a randomization test in a one-way ANOVA setting with three treatments on an artificial data set with the purpose of demonstrating the general idea underlying randomization test.

1 Introduction

Randomization tests are type of nonparametric statistical test based on the random assignment of experimental units to treatment groups. R.A. Fisher is credited with their creation in the 1930s, and used the idea as a theoretical defense of Student's t-test rather than a statistical tool [ET93]. The idea behind randomization tests is incredibly simple, and naturally void of most assumptions, such as random sampling or equal variance. The only requirement of randomization tests is the random assignment of subjects to treatments[Kut05].

The crux of the randomization test is the realization that permutations of observed data between treatments does not alter its distribution under the null hypothesis. That is, given two treatments A and B, a value observed under treatment A could have equally resulted under treatment B under the null hypothesis of equal treatment effects. With this observation one may consider all the possible permutations of data values across treatments. The set of all such permutations is called a reference set. For each permutation in the reference set, a test statistic may computed, and the proportion of permutations with test statistics at least as large as the statistic based on the original data determines the achieved significance level of the test. This approach is often referred to as *systematic permutation*, as it requires the construction of every possible permutation in the reference set. For large amounts of data, systematic permutation is often unfeasible. So one may alternatively employ *random permutation*. In this case, a sample of permutations are randomly generated, the test statistic is calculated for each element of the same, and the achieved level of significance is again the proportion of statistics greater than the original, relative to the sample.

One potential shortcoming of randomization tests is that statistical inference is confined to the subjects used in the experiment, and generalization must be supported by reasoned argument. Additionally, while the general idea behind randomization tests doesn't change, the procedure one must follow to do the data permutation is generally dependent on the experimental design of the situation being studied [BS13].

2 Problem

In this report we consider artificial data arising in a one-way ANOVA setting with three treatments: A, B, and C. The data are arranged in the table below.

Table 1: One-way ANOVA Data

A	B	C
6, 8	9, 11, 9	17, 15, 16, 16

We wish to evaluate the hypothesis $H_0 : \mu_A = \mu_B = \mu_C = 0$ using both random and systematic permutation.

3 Method

3.1 Systematic Permutation

Systematic permutation involves obtaining all possible permutations of the observed data. In the one-way ANOVA case given above in Table 1, we can determine that there are $\frac{9!}{2!3!4!} = 1260$ distinct arrangements of the data points across treatments A, B, and C. To generate them, we first assign unique indices to each of the data points

Table 2: Data with indices assigned

	A	B	C
Data:	6, 8	9, 11, 9	17, 15, 16, 16
Indices:	1, 2	3, 4, 5	6, 7, 8, 9

Then, starting with treatment A we find all distinct combinations of indices 1-9 taken two at a time. There are $\binom{9}{2}$ such combinations of indices. For each one of those combinations, all of the $\binom{7}{3}$ remaining combinations of treatment B are then generated. The remaining indices are then default allocated to Treatment C. See Table 3 for an example.

For each of the permutations generated, the F-test statistic is recalculated. The p-value, or achieved significance level, of the randomization test is then calculated to be

$$p = \frac{\#\{F^* > \text{original } F\}}{\text{total number of permutations}} = \frac{\#\{F^* > \text{original } F\}}{1260}$$

where F^* represents the F statistic for the permuted data.

Table 3: Systematic Permutation Example

A	B	C
(1,2)	(3,4,5)	(6,7,8,9)
\vdots	(3,4,6)	(5,7,8,9)
\vdots	(3,4,7)	(5,6,8,9)
\vdots	\vdots	\vdots
(3,5)	(1,2,4)	(6,7,8,9)
\vdots	(1,2,6)	(4,7,8,9)
\vdots	(1,2,7)	(4,6,8,9)
\vdots	\vdots	\vdots
(8,9)	(1,2,3)	(4,5,6,7)
\vdots	\vdots	\vdots

It should be noted that the procedure can be optimized slightly by modifying the test statistic. The F-statistic is calculated as the ratio of the Mean Square Treatment and the Mean Square Error values. That is, $F = \frac{MSTR}{MSE}$. These values can be equivalently expressed as

$$MSTR = \frac{1}{k-1} \left(\sum \left(\frac{T_i^2}{n_i} \right) - \frac{T^2}{n} \right)$$

$$MSE = \frac{1}{n-k} \left(\sum x_i^2 - \sum \frac{T_i^2}{n_i} \right)$$

where T_i and n_i represent the total and number of subjects, respectively, for a given treatment, and T and n are the overall treatment total and number of subjects. With this formulation, it can be shown that the quantity $\sum T_i^2/n_i$ is equivalent to the F statistic. This is done by making the following observations [Edg95]:

- The F-statistic contains the ratio $\frac{n-k}{k-1}$, which is constant over all data permutations, so removing it doesn't effect how permutations may be ranked according to F value. This implies that the ratio of the sum of treatment squares, $SSTR$, to the sum of squared errors, SSE is equivalent to the F value. That is, $\frac{SSTR}{SSE} \sim F$
- The total sum of squares is constant over all data permutations, and can be partitioned into the sum of treatment squares and sum of squared errors. Furthermore, as the sum of treatment squares increases, the sum of squared errors must decrease. Thus, the ratio $\frac{SSTR}{SSE}$ and $SSTR$ vary in the same direction across all data permutations. Consequently, $SSTR$ is equivalent to $\frac{SSTR}{SSE}$, and hence F.
- Finally, note that $SSTR$ is precisely equal to $\sum \left(\frac{T_i^2}{n_i} \right) - \frac{T^2}{n}$, and the quantity $\frac{T^2}{n}$ is constant

across all permutations. So it's removal does not effect how permutations may be ranked by equivalent statistics. Consequently, $\sum T_i^2/n_i \sim SSTR \sim F$.

With this observation, we can restate the criteria for calculating the achieved significance level:

$$p = \frac{\#\{\sum T_i^{*2}/n_i > \text{original } \sum T_i^2/n_i\}}{\text{total number of permutations}} = \frac{\#\{\sum T_i^{*2}/n_i > \text{original } \sum T_i^2/n_i\}}{1260}$$

3.2 Random Permutation

Random permutation can be employed when systematic permutation is unfeasible or impossible due to computational limitations. In the one-way ANOVA case, this amounts to randomly generating R permutations and calculating the achieved significance level using the formula above. The only difference being that the denominator is now R instead of the total number of possible permutations. In this report, we take $R = 800$:

$$p = \frac{\#\{\sum T_i^{*2}/n_i > \text{original } \sum T_i^2/n_i\}}{800}$$

To generate the collection of random permutations, labels are assigned to the original data as in Table 2, then 2 indices are randomly selected for allocation to treatment A, 3 are randomly selected from the remaining 7 indices for treatment B, and the remaining indices are allocated to treatment C. This process is then repeated R times, and the equivalent statistic calculated for each permutation.

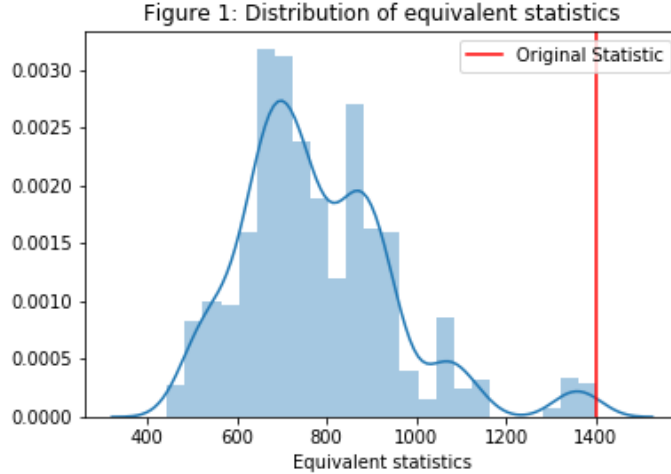


Figure 1: Distribution of the equivalent statistics for systematic permutation. The position of the original statistic marked by the vertical red line on the right of the figure.

4 Results

4.1 Systematic Permutation

Running the randomization test using systematic permutation yielded an equivalent statistic $\sum T_i^2/n_i = 1402.33$ for the original data, with resulting p-value of $p = 0.00079$, well below standard levels of significance. Thus we may reject the null hypothesis that all treatment means are all equal. The equivalent statistics for each permutation were plotted along with the original statistic for perspective in Figure 1.

4.2 Random Permutation

The randomization test using 800 random permutations yielded a p-value of 0. Thereby agreeing with the conclusion drawn from the systematic permutation. Figure 2 shows the distribution of equivalent statistics for the 800 permutations generated, along with a reference line for the original test statistic. Additionally, to further investigate the random permutation, the test was replicated 1000 times and p-values for each run tabulated. Their distribution is given in Figure 3, in which it is evident that for all cases the null hypothesis is rejected.

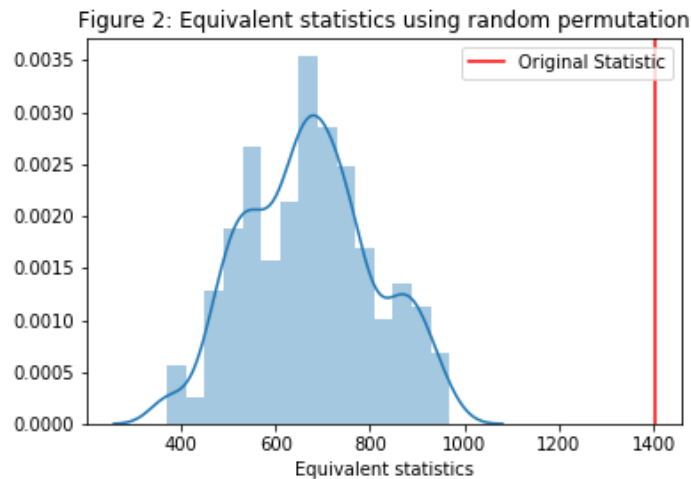


Figure 2: Distribution of equivalent statistics for random permutation. Original statistic is given by the solid red line.

5 Conclusion

In this report, we have only briefly looked at randomization tests in a one-way ANOVA context, using both systematic and random permutation. While the example may have been contrived, the importance of the method cannot be understated. In experimental research, it is typically very uncommon to have random selection of experimental units – especially so in social science and epidemiology.

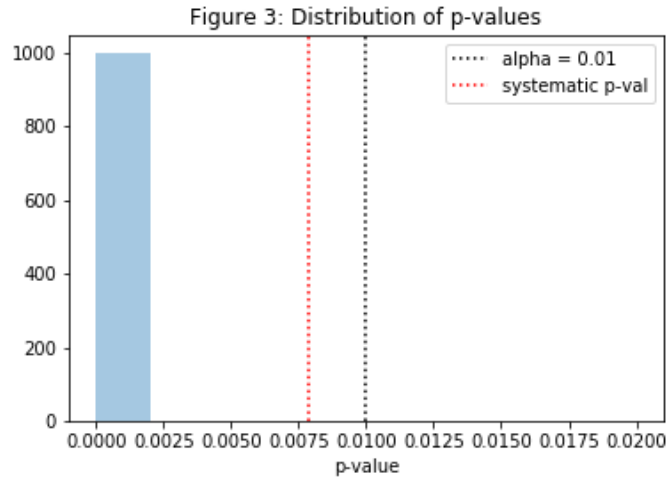


Figure 3: p-values for 1000 runs of randomization tests using 800 random permutations.

Thus randomization tests are both widely applicable and beneficial in designed experiments in that the tests are only contingent on random assignment to treatment groups. Furthermore, because of the non-parametric nature of the test, the opportunity for letting the data speak for itself. That said, a point of caution must be made: because randomization tests can typically only be applied in experimental designs, the validity of the results are naturally tied to the data collection mechanism and how well the experimenter can control for non-randomized variables. Consequently, we may end this report by observing that the randomization test, like any tool, is best wielded by one who understands the proper context and circumstances for its use.

References

- [BS13] D.D. Boos and L.A. Stefanski. *Essential Statistical Inference: Theory and Methods*. Springer Texts in Statistics. Springer New York, 2013.
- [Edg95] E. Edgington. *Randomization Tests, Fourth Edition*. Statistics: A Series of Textbooks and Monographs. Taylor & Francis, 1995.
- [ET93] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA, 1993.
- [Kut05] M.H. Kutner. *Applied Linear Statistical Models*. McGraw-Hill international edition. McGraw-Hill Irwin, 2005.

Appendix

Joseph Free

Project 3

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from itertools import combinations
```

1. Systematic Permutation

To perform systematic permutation on the data

A	B	C
6,8	9, 11, 9	17, 15, 16, 16

we implement the following algorithm:

algorithm

- Assign to each observation a unique index ranging from 1 to 9.
- Generate all partitions of the indices into three groups of size $n_A = 2, n_B = 3, n_C = 4$ by
 - Finding all combinations of 2 indices taken from the 9 for treatment A, and then
 - For each combination of indices for treatment A, finding all possible combinations of 3 indices from the remaining 9-2 = 7 indices for treatment B.
 - Treatment C is then relegated the remaining indices.
- For the resulting partitions, calculate the equivalent statistic

$$T^2 = \frac{T_A^2}{n_a} + \frac{T_B^2}{n_B} + \frac{T_C^2}{n_C}$$

- Calculate the p-value for the randomization test using

$$p = \frac{\{\#T^2 \geq \text{original } T^2\}}{1260}$$

```

In [2]: # Assign labels to data points.
data_points = {1:6, 2:8,          # Treatment A
               3:9, 4:11, 5:9,     # Treatment B
               6:17, 7:15, 8:16, 9:16} # Treatment C

# Initialize array of equivalent statistics.
equivalent_statistics = np.array([])
perms = []

# Initialize list of indices
l = [1,2,3,4,5,6,7,8,9]
# Begin permutations.
# For each combination of values for treatment A....
for permA in list(combinations(l, 2)):
    A_prime = list(permA)

    # ....determine which indices remain to be permuted....
    remaining = [value for value in l if value not in A_prime]

    # ....From remaining indices, find all combinations of 3 taken from 7 for treatment B
    for permB in list(combinations(remaining, 3)):
        B_prime = list(permB)

        # Again determine remaining indices. These are the remaining values relegated to Treatment C.
        remaining = [value for value in remaining if value not in B_prime]

        # Calculate statistic for a completed permutation.
        TA = sum([data_points[i] for i in permA])**2/2
        TB = sum([data_points[i] for i in permB])**2/3
        TC = sum([data_points[i] for i in remaining])**2/4
        equivalent_statistics = np.append(equivalent_statistics,TA+TB+TC)
        perms.append(A_prime + B_prime + remaining)

# Calculate p-value for the randomization test.
originalT = equivalent_statistics[0]
numerator = sum( equivalent_statistics >= originalT )
pval = numerator/1260
print("Randomization p-value: " + str(pval))
print("Original T:" + str(originalT))
print("Number of equivalent statistics: " + str(len(equivalent_statistics)))

```

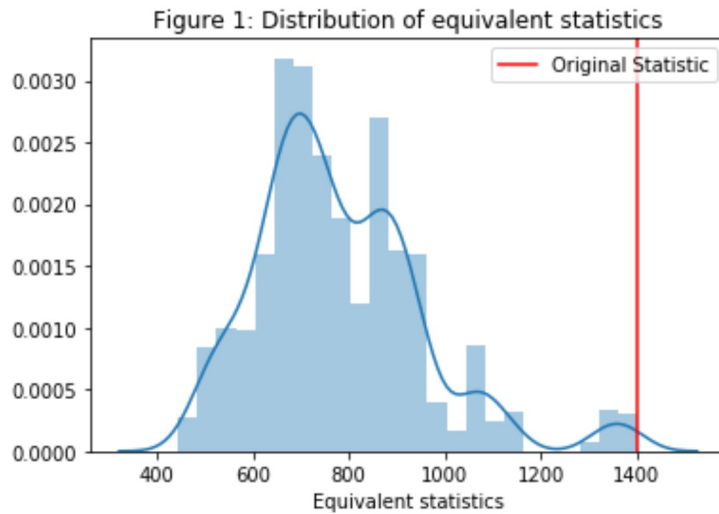
```

Randomization p-value: 0.0007936507936507937
Original T:1402.3333333333333
Number of equivalent statistics: 1260

```



```
In [3]: # Distribution of equivalent statistics with original stat. overlaid.
ax = sns.distplot(equivalent_statistics)
ax.set_xlabel("Equivalent statistics")
ax.set_title("Figure 1: Distribution of equivalent statistics")
ax.axvline(originalT, color='red')
ax.legend(["Original Statistic"])
plt.savefig('project 3/fig1.png', format = 'png');
```



2. Random Permutation

To implement random permutation, the following algorithm is used.

algorithm

- Assign to each observation a unique index ranging from 1 to 9.
- For R = 800 times, do the following:
- Randomly allocate indices to the first two treatments groups of size $n_A = 2, n_B = 3$ by
 - Randomly selecting two elements of index values from 1 through 9 for treatment A, removing them from the list of indices.
 - For the remaining seven elements, randomly select three elements for treatment B, removing these too from the list.
 - Treatment C is then relegated the remaining indices.
- For the resulting permutations, calculate the equivalent statistic

$$T^2 = \frac{T_A^2}{n_a} + \frac{T_B^2}{n_B} + \frac{T_C^2}{n_C}$$

- Calculate the p-value for the randomization test using

$$p = \frac{\{\#T^2 \geq \text{original } T^2\}}{800}$$

```

In [4]: def random_perm(n=800):
    # Assign labels to data points.
    data_points = {1:6, 2:8,          # Treatment A
                   3:9, 4:11, 5:9,     # Treatment B
                   6:17, 7:15, 8:16, 9:16} # Treatment C
    l = [1,2,3,4,5,6,7,8,9]

    # Calculate original statistic for comparison.
    TA = sum([data_points[i] for i in l[0:2]])**2/2
    TB = sum([data_points[i] for i in l[2:5]])**2/3
    TC = sum([data_points[i] for i in l[5:9]])**2/4
    originalT = TA+TB+TC

    # Initialize array to save equivalent stats.
    equivalent_statistics = np.array([])
    for i in range(0,n):
        # Generate indices for random allocation.
        perm = []
        for j in range(8,3,-1):
            perm.append(l.pop(np.random.randint(0,j)))

        # Calculate an equivalent statistic.
        TA = sum([data_points[i] for i in perm[0:2]])**2/2
        TB = sum([data_points[i] for i in perm[2:5]])**2/3
        TC = sum([data_points[i] for i in l[5:9]])**2/4
        equivalent_statistics = np.append(equivalent_statistics,TA+TB+TC)

    # Reset l
    l = [1,2,3,4,5,6,7,8,9]

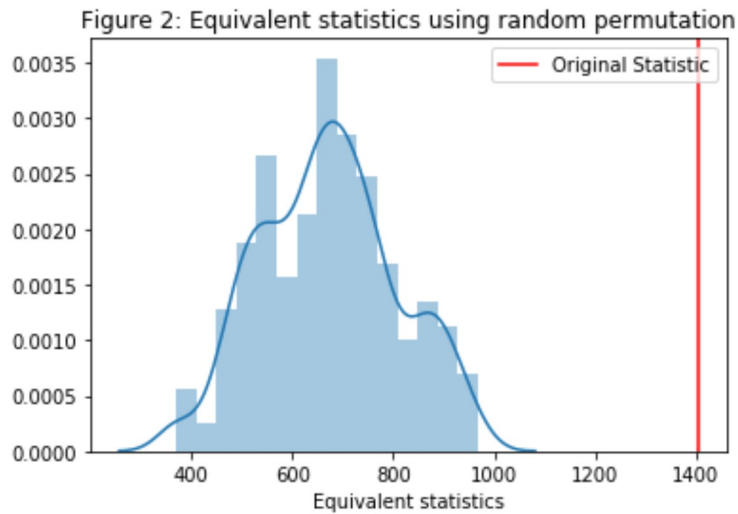
    # Calculate and return p-value.
    numerator = sum( equivalent_statistics >= originalT )
    pval = numerator/n
    return (pval, equivalent_statistics)

np.random.seed(829)
results = random_perm(800)
print("Randomization p-value: " + str(results[0]))

```

Randomization p-value: 0.0

```
In [5]: # Plot distribution of random equivalent statistics.
ax = sns.distplot(results[1])
ax.set_xlabel("Equivalent statistics")
ax.set_title("Figure 2: Equivalent statistics using random permutation")
ax.axvline(originalT, color='red')
ax.legend(["Original Statistic"])
plt.savefig('project 3/fig2.png', format = 'png');
```



```
In [6]: # Run 1000 trials of randomized permutations using size 800.
p = []
for i in range(0,1000):
    p.append(random_perm(800)[0])
ax = sns.distplot(p, bins=10, kde=False, hist_kws={'range':(0,.02)})
ax.set_xlabel("p-value")
ax.axvline(0.01, ls = 'dotted', color='black')
ax.axvline(0.0079, ls = 'dotted', color='red')
ax.set_title("Figure 3: Distribution of p-values");
ax.legend(["alpha = 0.01", "systematic p-val"])
plt.savefig('project 3/fig3.png', format = 'png');
```

