

Grand Kasai Randomization

Sergiy Radyakin

04 April 2022

Problem setting

The UNHCR survey in the Grand Kasai region of the DRC (2022) uses a particular experimental section, which relies on randomization to acquire respondents' valuation of different factors impacting the relationships between refugees/displaced persons and hosting communities.

The experiment is based on the comparison of two randomly created profiles A and B. The rules for which are the following:

1. *Each profile consists of a number of dimensions N (in this implementation $N=3$), which we will denote with Latin letters X , Y , and Z . For example, the three dimensions X , Y , and Z may be "Distance", "Complexity", and "Considerations for displaced people".*
2. *The dimensions are randomly ordered, and the order is the same for each pair of profiles A and B.*
3. *For each dimension there is a certain number of images M (in this implementation $M=2$), which are selected at random. The same image may be included in both profiles A and B.*
4. *The two randomly generated profiles A and B should never be identical (must differ by at least one image).*

It is convenient to present the resulting profiles that we are seeking to generate with formulas like: $L\#-L\#-L\#$, where L stands for a letter denoting the dimension, and $\#$ stands for the image number selected for that dimension.

For example, here are some valid profiles (the code should generate such profiles):

A: X1-Y1-Z1

B: X2-Y2-Z2

A: Z1-X2-Y1

B: Z2-X1-Y1

A: Y1-X2-Z1

B: Y2-X1-Z1

And here are some invalid profiles (the code should not generate such pairs):

A: X1-X1-Z1

B: X2-X2-Z2

(violates rule #1 because only two dimensions are mentioned in each profile)

A: X1-Y1-Z1

B: Z2-Y2-X2

(violates rule #2 because the order of dimensions differs between the profiles).

A: X1-Y5-Z1

B: Z2-Y2-X2

(violates rule #3 because the index of selected image (5) for dimension Y of profile A is higher than the possible number of images (2)).

A: X1-Y1-Z1

B: X1-Y1-Z1

(violates rule #4 because the two profiles are identical).

Implementation in Survey Solutions

Since the two profiles must exhibit the same order of dimensions, we will randomize their order once and reuse them for both profiles A and B. Given three dimensions X, Y, and Z there will be six possible combinations of their order: XYZ, XZY, YXZ, YZX, ZXY, and ZYX.

We can randomize a particular order of dimensions by randomly picking one of these six combinations, like so:

```
L=new[] {"XYZ", "XZY", "YXZ", "YZX", "ZXY",  
        "ZYX"}[(int)Math.Floor(r.Value*6)];
```

where r is a random value between 0 and 1.

We now need to populate the dimensions with selected images. We will do this by picking six (three dimensions times two profiles) random numbers $d_1, d_2, d_3, d_4, d_5, d_6$. It is convenient to do this by extracting individual digits from a single random number, like so:

```
S = r.Value.ToString("0.000000");  
d1 = int.Parse(S[2].ToString())>=5 ? "2":"1";  
d2 = int.Parse(S[3].ToString())>=5 ? "2":"1";  
d3 = int.Parse(S[4].ToString())>=5 ? "2":"1";  
d4 = int.Parse(S[5].ToString())>=5 ? "2":"1";  
d5 = int.Parse(S[6].ToString())>=5 ? "2":"1";
```

```
d6b = int.Parse(S[7].ToString())>=5 ? "2":"1";
```

where r is a random number between 0 and 1. When converting the random number to a string, we apply a mask to ensure that the number of digits after the comma is six (noting that the integer part is always zero, because the random number is between 0 and 1). When extracting individual digits from the string, we use the fact that indexing in arrays and strings is zero-based in C#. If a random digit is 0, 1, 2, 3, or 4 then image #1 will be selected for that dimension, otherwise, if a random digit is 5, 6, 7, 8, or 9 then image #2 will be selected for that dimension.

Note that we are not done yet, because we need to make sure the two profiles are not identical. We verify that to determine the last coordinate of the second profile:

```
d6 = (d1.ToString() + d2.ToString() + d3.ToString()) != (d4.ToString() + d5.ToString() + d6b.ToString()) ? d6b : (3 - int.Parse(d6b)).ToString();
```

Where the very last part of the expression is responsible for selecting another image for the sixth coordinate if the two profiles are identical. This decision was arbitrary. One could have opted rather to change the first coordinate of the first profile, or any other coordinate of any profile. Given that the dimensions X, Y, and Z are also randomized between the positions 1, 2, and 3 that should not affect which dimension will be altered to make the profiles different.

We can now construct the two profiles A and B by concatenating the necessary letters and digits:

```
ProfileA = "A: " + L[0] + d1 + "-" + L[1] + d2 + "-" + L[2] + d3  
ProfileB = "B: " + L[0] + d4 + "-" + L[1] + d5 + "-" + L[2] + d6
```

All intermediate and final results of the above calculations we will store in the calculated variables of our Survey Solutions questionnaire. We can further mark the intermediate results as not intended for export, while the final results we will retain in the export data for having access to them after the end of the experiment.

One could also agree that the random number used for randomization of the order of the dimensions is better be different from the random number used for selection of images for the profiles, to avoid systematic selection bias (such as if X is the first dimension then the first image is always selected for it in profile B, for example). We do this by generating two pseudo-random numbers:

```
r1 = Quest.IRNd();  
r2 = new Random((int)Math.Floor(r1*100000)).NextDouble();
```

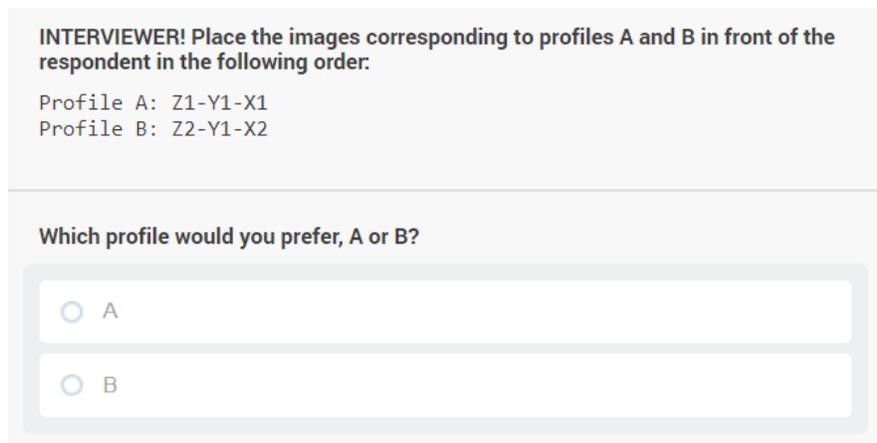
which we then use instead of r in the above calculations. Note that we initialize the random number with a random seed to make sure that our second random number also varies with realizations. If we were to do instead:

```
r2 = new Random().NextDouble();
```

we'd always get the same number in any realization. We've used a multiplier of 100000, but we could have used any other large value, which would be sufficient to make sure the second random sequence is initialized with a random seed.

Appearance in Survey Solutions

The calculations discussed above do not have any visual presence in the Survey Solutions questionnaire, but we can display the resulting profiles in the interview by including the profile variables into a static text via text substitution:



INTERVIEWER! Place the images corresponding to profiles A and B in front of the respondent in the following order:

Profile A: Z1-Y1-X1
Profile B: Z2-Y1-X2

Which profile would you prefer, A or B?

A

B

Considerations for extension

The implementation above is done for $N=3$ and $M=2$. But it can be generalized for other values of N and M .

As N grows, we will have more combinations of the order of the dimensions, and their total number will grow as $N!$, thus for 4 dimensions we will have 24 combinations and for 5 already 120 combinations. For values of N above 5 it will not be feasible to include all possible combinations of order of dimensions into a Survey Solutions' expression. But it is also not necessary. One could use a string-shuffling algorithm of a kind shown here:

<https://stackoverflow.com/questions/4739903/shuffle-string-c-sharp>

for this purpose.

Also, as N grows, we will eventually run out of the decimal grid width for the numbers stored in Survey Solutions. So, rather than relying on individual digits of a single random number, one should generate more random numbers. This can be done using the same mechanism shown above for the $r1$ and $r2$.

Likewise, as M grows operating on a single digit will not be convenient. For example, when $M=3$ we will need to divide 10 digits into 3 groups, which may result in an unequal probability of selection between the images of the same dimension (for example, $\{0,1,2\}$ for #1, $\{3,4,5\}$ for #2, and $\{6,7,8,9\}$ for #3). Having a new random number for each coordinate will avoid having these compromises.