CollegeBoard

Creating Matched Samples Using Exact Matching

Kelly E. Godfrey

March 2016

Executive Summary

Creating matched samples prior to conducting analyses using treatment and control groups is one way to control for observable group differences non-parametrically and reduce dependence on model assumptions, thereby producing results that are easier to understand and interpret. This paper presents an optimization approach to create two matched one-to-one samples that simultaneously maximizes the number of matches while minimizing the differences between matches. Example SAS code for matching two samples is provided, as well as guidance for expanding the match to three or more groups. It is assumed the reader has a working knowledge of basic research terminology and basic SAS coding, and some minor familiarity with SAS macro functions.

Introduction

By creating and analyzing matched samples, researchers can simplify their analyses to include fewer covariate variables, relying less on model assumptions, and thus generating results that may be easier to report and interpret. When two groups essentially "look" the same, it is easier to explore their differences and make comparisons based on group membership. However, it should be noted that matched samples do not replace the ideal experimental design in which participants are randomly assigned to groups as opposed to self-selection, and causal inferences may not be fully supported by the analyses.

One of the most popular matching approaches in recent literature is propensity score matching, or PSM (Rosenbaum & Rubin, 1983). This approach uses logistic regression to predict group membership, assigning propensity scores to individual participants, then matching those participants across membership groups using those propensity scores. While this approach ideally removes selection bias of observable characteristics, it does not always produce the desired balance (Austin, 2008). For a more thorough critique of propensity score matching, see King and Nielsen (2016).

The optimization approach presented here was born out of a need to produce matched samples of students from two groups: one who had participated in a particular advanced high school course or set of courses, and one who had not. Propensity score matching produced matched samples of students who not only were unbalanced in terms of student sex, race/ethnicity, and parental education levels but also were significantly different on average test scores. Therefore, an approach to match students one-to-one where sex, race/ethnicity, and parental education levels were identical and test scores were very close (if not equal) was developed in an attempt to create, by force, a balanced sample of students. This report describes the exact matching approach developed, related data considerations, preparation and requirements, and expansion beyond two samples. SAS code with annotations to be used in future research is also provided.

Requirements

In order to create matched samples, there are several requirements:

1. *Two or more groups of study participants to be matched.* Group membership must be exclusive, in that participants must belong to one group and one group only.
2. *At least one variable, discrete or continuous, to be used for matching between groups.* There is no limit on the number of variables used in matching, but as more variables

are considered, the fewer matches will likely be returned. A combination of discrete and continuous matching variables is presented here.

3. *Some overlap in matching variables between the groups.* The more the groups overlap on the variables used in the matching, the better the match. For example, Figure 1 demonstrates a situation where two groups do not have a lot of overlap in the distributions of a variable, *score*. Figure 2, on the other hand, depicts good overlap between the two groups and is a more desirable situation for producing as many matches as possible.

Figure 1: Poor overlap between two groups on matching variable.

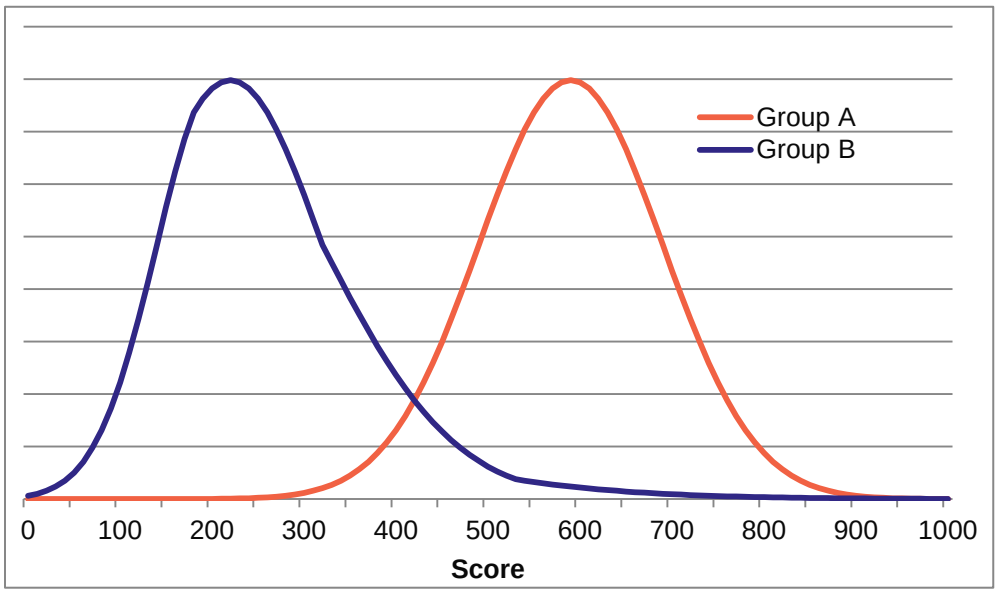
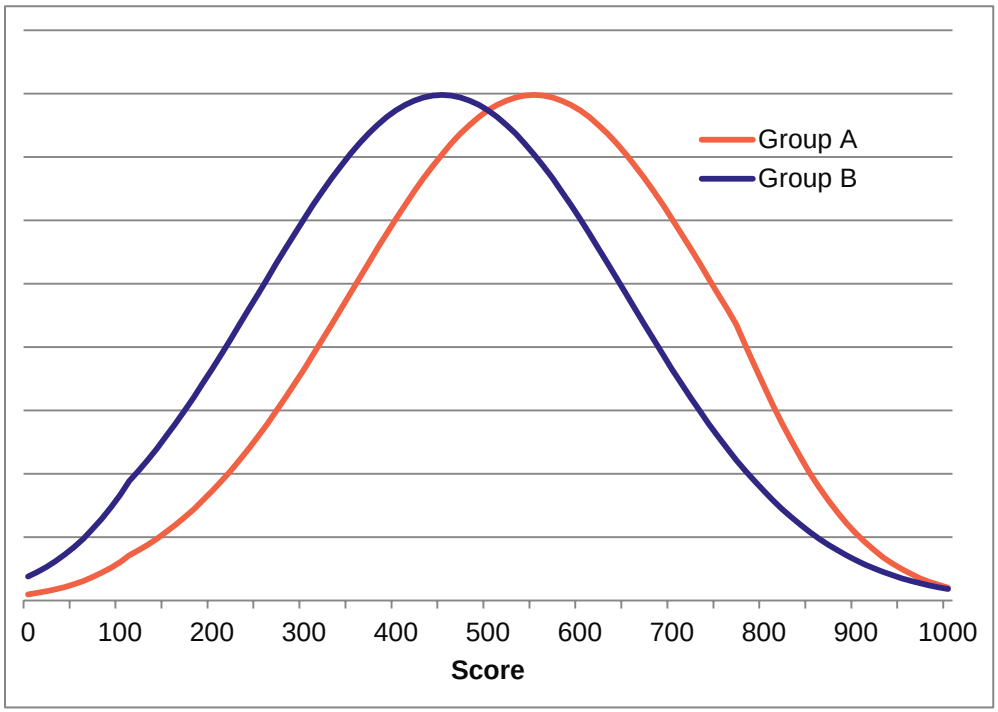


Figure 2: Good overlap between two groups on matching variable.



Exact Matching: Two Groups

Preparation

Before matching, you must determine the variables to be used for the match. Variables may be discrete (such as sex of the participant) or continuous (such as test score), and there should be reasonable overlap between the groups on each of the variables. Variables are considered simultaneously during the match, so multivariate overlap is also necessary for matching a proportion of individuals. While more variables may create a stronger match, keep in mind that including more constraints will likely decrease the number of matches identified, and researchers should select only those that are meaningful and well-populated to be included in the matching process.

Once matching variables are identified, the researcher then creates rules around those variables. What must be exactly the same between matches? For example, do matches need to be the same sex and race/ethnicity? What must be close or adjacent? For example, do matches need to have identical test scores, or just close scores? How close is acceptable? The level of acceptable closeness is at the researcher's discretion, and variables can be matched at varying degrees of similarity.

When matching two groups of participants, it is highly recommended that the smaller group be selected as "group A" and the larger as "group B." In this approach, group A receives some preference in maintaining as many members as possible in the final matched sample. Assign each participant a unique ID if they do not already have one.

Matching Steps

1. Once rules have been determined during your preparation steps and your two groups are ready to be matched, the first step coded in this approach creates a list of *all possible matches* of participants from the larger group (group B) to each participant in the smaller group (group A). Any participant from group A without potential matches in group B is removed from consideration at this point. This particular step is conducted *with replacement*, so participants from group B can be matched to more than one participant from group A. The list of potential matches is then put into one file until all participants from group A are considered.
2. A proximity score is then calculated for all possible matches using the variables that are allowed to be inexact matches. One approach to calculating this score involves calculating the average absolute value of the difference between the variables. For example, if your match were to use Reading, Math, and Writing scores along with other demographic variables to create pairs of similar students, the proximity score would be calculated as follows:

$$prox = \frac{|Reading_A - Reading_B| + |Math_A - Math_B| + |Writing_A - Writing_B|}{3}$$

   This score represents an overall *closeness* of scores of the two possible matches and will help optimize the matches. It is important to note that lower proximity values here indicate closer scores between two potential matches, and larger proximity indicates larger distance. The smaller this value, the closer the two sets of scores are, with zero indicating that all three test scores were identical between the two students.
3. This matching approach attempts to find closer matches while simultaneously maximizing the number of matches. To help do this, the number of possible matches in group B is calculated for each participant in group A, followed by the number of matches in group A for each member of group B. A participant is considered "lonely" if he or she has very few possible matches from the other group.
4. All possible matches are then ordered by the following:
   - 4.1 Number of matches for the A participant
   - 4.2 Proximity scores
   - 4.3 Number of matches for the B participant

5. This first returns the loneliest participant from group A with the closest match to the loneliest group B participant. This match is then moved to the matched sample list, and the two participants (A and B) are removed from the pools for further consideration.
6. At this point, the number of possible matches for each member of groups A and B are recalculated, and the files are reordered using the same specifications listed above. The first match listed is moved to the matched sample list, and the process (steps 4–6) continues repeating until there are no more participants from group A remaining.

SAS code for this approach, which can be adapted to suit a wide variety of matching needs, is located in Appendix A.

Expanding to Three Samples

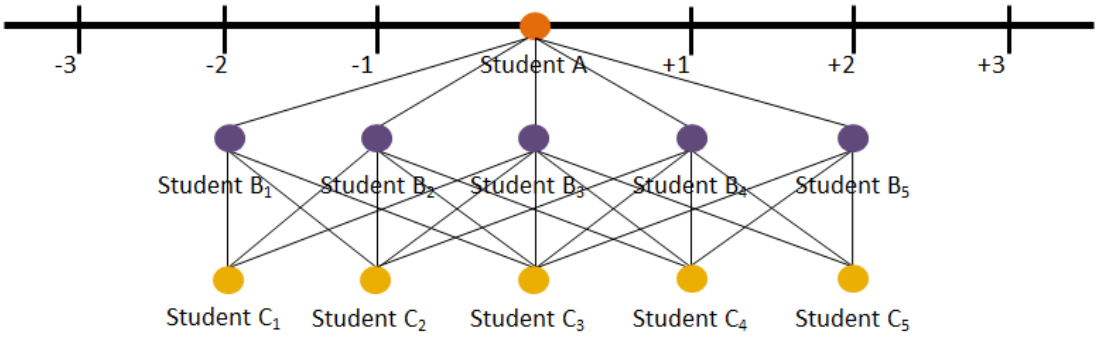
In some cases, it may be necessary to match three or more samples for analysis. This can be achieved by altering the approach presented previously.

Just as with two groups, when matching three groups of participants, group membership must be mutually exclusive; participants can belong to only one group. Any number of matching variables can be used, in any combination of discrete and continuous. It is recommended that group A be the smallest group, followed by group B, and then group C is the largest. The steps are similar to those outlined for the two-group match, but with the following adaptations:

1. In developing the list of all possible matches for every participant in group A, the combinations are determined in two steps instead of one:
    a. Every participant from group B that meets the criteria for match group participant A is listed, just as a two-group match.
    b. Every participant from group C is compared to the combination of A and B participants to determine if the rules are met. If participant meets the rules for group A participant and group B participant, the match possibility is maintained.

    Figure 3 depicts an example of these two steps. In this example, the matching rules for one continuous variable require participants (in this case, students) to be within two test score points of each other in order to be considered for possible match. There are a total of 19 possible match combinations depicted, each of which would be considered further for possible inclusion in the final list of matches.

Figure 3: Triads of possible matches within two score points.



2. The proximity score calculation includes the additional group, such that the distances from A to B, B to C, and A to C are now all considered. To use the example presented previously in the two-group match (three scores: Reading, Math, and Writing), proximity would be calculated as the average absolute difference of every combination of student from one group to another:

$$prox = \frac{|Reading_A - Reading_B| + |Reading_B - Reading_C| + |Reading_A - Reading_C| + |Math_A - Math_B| \ldots}{9}$$

3. Instead of simply the number of matches for each participant from group A and group B, in matching three groups, the number of possible matched triads are counted for each participant in group A and each in group B.
4. Similar to the two-group approach, the next step is to then sort on the number of possible matches for each participant A, proximity scores, and the number of triad matches for participant B. The first triads listed are for the loneliest participant A's tightest matches, in order by loneliest participant B.
5. The top match listed is selected for matched sample and then steps 2 through 5 are repeated until no group A participants remain for consideration.

Recommendations for Checking the Match

Once your matches have been selected, it is strongly recommended that you check the final sample to ensure that your rules have been met.

Every pair should be identical on the variables where you have specified this constraint. For continuous variables that have been allowed to vary within specified bounds, group mean differences should not be meaningful and/or statistically significant. This can be checked by conducting a simple t-test to compare means. It should be noted that p-values are influenced

by sample size, so if a large number of pairs is produced by your match, a smaller mean difference may be considered statistically significant. For this reason, it is recommended to use your best judgment and practical reasoning for ultimately determining trustworthiness of the match, including considering higher moments such as standard deviation, skewness, and so on. If group differences remain too large for your preference, it is recommended that you reduce the degree to which continuous variables are allowed to vary. Of course, this may reduce the number of matches produced in total.

One additional check that is recommended is to calculate the percentage of participants from the original group A sample maintained by the match and to check the level of representativeness of the matched sample. If one or more groups are systematically removed by the matching approach, this will impact the generalizability of the subsequent analyses you conduct.

Challenges and Limitations

Matched samples do not replace the ideal approach of experimental design, where subjects are randomly assigned to groups or treatments. Causal inferences, therefore, are not recommended. However, group differences are better understood and explained when some inherent differences are accounted for and essentially removed in the match. Matched subjects may differ on constructs that were not measured, and therefore some unobserved group differences may remain after matching. This limitation should be acknowledged when reporting and interpreting results.

As described in the requirements for matching and demonstrated in Figures 1 and 2, the match is limited by the amount of overlap between the two groups. If the groups differ greatly on the variables with which you wish to match, the number of matches will be limited. If some groups are insufficiently represented in the resulting matched sample, this matching approach may need to be reconsidered. Future research comparing the outcome of this approach to the matches created by other methods such as propensity score matching using simulated data in a variety of realistic designs would be useful in further developing our understanding of and confidence in creating matched samples.

References

Austin, P. C. (2008). A critical appraisal of propensity-score matching in the medical literature between 1996 and 2003. *Statistics in medicine*, *27(12)*: 2037–2049.

King, G., & Nielsen, R. (2016). "Why propensity scores should not be used for matching." Retrieved from http://gking.harvard.edu/files/gking/files/psnot.pdf?m=1439838506

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1): 41–55.

Appendix A: SAS Code for Two Sample Match

Prior to this code being executed, two files were created: GroupA contains all students who participated in a specific advanced math course. GroupB lists students who did not. All students had a unique ID and had values for the following matching variables:

- Exactly matched variables:
  - Sex
  - Race/ethnic code
  - Parental education
- Similar variables (within two score points):
  - Verbal score
  - Writing score
  - Math score

It is recommended that the SAS log be written to an external file in order to allow the match algorithm to complete without user assistance clearing the log window. With an increased number of participants, log output may become extensive and computing time increased. The code for this command is included below.

```
/*************************************************************************/
/*                            Exact Match*/
/*                          Kelly E Godfrey*/
/*************************************************************************/

*Change log to output to file in folder on hard-drive;
filename myfile 'C:\Matching\mylog.log'; /*Change the directory listing*/
proc printto log=myfile;
run;

/*Sequentially number the students in each group*/
proc sort data = groupA;
      by unique_id;
run;
data groupA(keep=studenta unique_id reading math writing sex ethnic
parent_ed group); /*StudentA is the sequential number assigned here*/
      retain studenta unique_id reading math writing sex ethnic parent_ed
group;
      set groupA;
      studentA + 1;
run;
proc sort data = groupB;
      by unique_id;
run;
```

```
data groupB(keep=studentB unique_id reading math writing sex ethnic
parent_ed group); /*StudentB is the sequential number assigned here*/
      retain studentB unique_id reading math writing sex ethnic parent_ed
group;
      set groupB;
      studentB + 1;
run;

/*How many students are in Group A?*/
proc sql noprint;
      select count(*) into: num_students
      from groupA;
quit;

/*Start the macro to go through the process, one student A at a time*/
%MACRO pods(num);
      %DO i = 1 %TO &num;
/*Choose the first student to be matched from group A, get scores and
relevant variables*/
data _null_;
      set groupA;
      if studentA = &i then do;
            call symput('ethnic_a',ethnic); /*Get ethnic code*/
            call symput('sex_a',sex); /*Get sex code*/
            call symput('parented_a',parent_ed); /*Get parent ed code*/
            call symput('reading_a',reading); /*Get reading test score*/
            call symput('writing_a',writing); /*Get writing test score*/
            call symput('math_a',math); /*Get math test score*/
      end;
run;
/*Find students in group B who match exactly on sex, ethnicity & parent
ed, and test scores within 2 points*/
data temp_matches1;
      set groupB;
      where (ethnic = &ethnic_a) and (sex = &sex_a) and (parent_ed =
&parented_a)
            and (abs(reading - &reading_a) le 2)
            and (abs(writing - &writing_a) le 2)
            and (abs(math - &math_a) le 2);
run;
/*Spit out all possible matches into one file - each student A with all
possible B matches*/
data temp1(keep=studentA studentB);
      retain studentA studentB;
      set temp_matches1;
      studentA = &i;
run;
/*Build onto a big list of potential matches to be narrowed down later*/
proc datasets nolist;
      append base = matches1 data = temp1 force;
      delete temp1 temp_matches1;
run;
quit;
%END;
%MEND pods;
```

```
%pods(&num_students)
proc sort data = matches1;
      by studenta studentb;
run;

*Changing variable names for comparisons later;
data studentsA(keep=studentA unique_ida readinga matha writinga parent_eda
ethnica sexa);
      set groupA;
      unique_ida = unique_id;
      readinga = reading; /*Change this to your variable*/
      matha = math; /*Change this to your variable*/
      writinga = writing; /*Change this to your variable*/
      parent_eda = parent_ed; /*Change this to your variable*/
      ethnica = ethnic; /*Change this to your variable*/
      sexa = sex; /*Change this to your variable*/
run;
data studentsB(keep=studentB unique_idB readingB mathB writingB parent_edB
ethnicb sexb);
      set groupB;
      unique_idB = unique_id;
      readingB = reading; /*Change this to your variable*/
      mathB = math; /*Change this to your variable*/
      writingB = writing; /*Change this to your variable*/
      parent_edB = parent_ed; /*Change this to your variable*/
      ethnicb = ethnic; /*Change this to your variable*/
      sexb = sex; /*Change this to your variable*/
run;
proc sort data = matches1;
      by studenta;
run;
data matches1;
      merge matches1(in=matches) studentsA;
      by studentA;
      if matches = 1;
run;
proc sort data = matches1;
      by studentb;
run;
data matches1;
      merge matches1(in=matches) studentsB(in=b);
      by studentb;
      if matches = 1;
run;
proc sort data = matches1;
      by studenta;
run;

/*Matches1 is a file with ALL studentA-studentB possible pairings.
Students from each group may appear more than once. They are
ordered by studentA sequential ID assigned in earlier code, then studentB
sequential ID. Proximity has not yet been taken into
account.*/
```

```
*************************************************************************;
*           Calculate a proximity index for the possible matches  ;
*************************************************************************;
data proximities;
      retain studenta studentb prox;
      set matches1;
      prox = (((abs(readingA - readingB)) + (abs(mathA - mathB)) +
(abs(writingA - writingB))) / 3);
run;

/*Proximities is a file with all of the possible studentA-studentB
possible matches as well as a proximity score calculated
for each possible match. This will allow us to choose closer matches when
we can.*/


*************************************************************************;
*           Matching macro;
*************************************************************************;
%MACRO MATCH();
*Order the proximities file by studentA;
proc sort data = proximities;
      by studentA;
run;
*Count the number of matches each studentA has;
data num_matchesA(keep=studentA num_matchesA);
      merge proximities;
      by studentA;
      if first.studentA then num_matchesA = 0;
            num_matchesA + 1;
      if last.studentA then output;
run;
*Merge the counts back into the proximities file;
data proximities;
      merge num_matchesA proximities;
      by studentA;
run;
*Order the proximities file by studentB;
proc sort data = proximities;
      by studentB;
run;
*Count the number of matches each studentB has;
data num_matchesB(keep=studentB num_matchesB);
      merge proximities;
      by studentB;
      if first.studentB then num_matchesB = 0;
            num_matchesB + 1;
      if last.studentB then output;
run;
*Merge the counts back into the proximities file;
data proximities;
```

```
      retain studentA num_matchesA studentB num_matchesB prox;
      merge num_matchesB proximities;
      by studentB;
run;
*****************************************************************;

*                 Work with loneliest studentAs first             ;

*****************************************************************;
proc sort data = proximities;
      by num_matchesA prox num_matchesB;
run;
*Pull top observation for consideration;
data temp;
      merge proximities (firstobs = 1 obs = 1);
run;
*Keep this match - loneliest A with closest loneliest B;
data _null_;
      set temp;
      call symput('student_A',studentA);
      call symput('student_B',studentB);
run;
*Remove that match (student A and student B) from further consideration;
proc sort data = proximities;
      by num_matchesA prox num_matchesB;
      where (studentA ne &student_A) and (studentB ne &student_B);
run;
*Write that match to the matched sample file, Matches;
proc datasets nolist;
      append base = matches data = temp force;
      delete temp;
run;
quit;
proc sql noprint;
      select count(*) into: num_matches_left
      from proximities;
quit;
%MEND MATCH;
*Count the number of possible matches left - is it more than 0?;
*Call the above macro, as long as the number of possible matches is
greater than 0;
%MACRO REPEAT();
proc sql noprint;
      select count(*) into: num_matches_left
      from proximities;
quit;
%put &num_matches_left;
%DO %WHILE ((&num_matches_left * 1) > 0);
      %MATCH();
%END;
%MEND REPEAT;
%REPEAT()
*Reset log output;
proc printto;
run;
```

```
*Create a matched file for checking group differences;
data matched_sample;
      set matches;
      /*Your matching variables go here*/
      keep unique_id group reading math writing parent_ed ethnic sex;
      unique_id = unique_ida;
      group = 'A';
      reading = readinga; /*matching variable*/
      math = matha; /*matching variable*/
      writing = writinga; /*matching variable*/
      parent_ed = parent_eda; /*matching variable*/
      ethnic = ethnica; /*matching variable*/
      sex = sexa; /*matching variable*/
run;
data temp;
      set matches;
      /*Your matching variables go here*/
      keep unique_id group reading math writing parent_ed ethnic sex;
      unique_id = unique_idb;
      group = 'B';
      reading = readingb; /*matching variable*/
      math = mathb; /*matching variable*/
      writing = writingb; /*matching variable*/
      parent_ed = parent_edb; /*matching variable*/
      ethnic = ethnicb; /*matching variable*/
      sex = sexb; /*matching variable*/
run;
proc datasets noprint;
      append base = matched_sample data = temp force;
      delete proximities studentsa studentsb matches1 num_matchesa
num_matchesb temp;
run;
quit;
```

About the College Board

The College Board is a mission-driven not-for-profit organization that connects students to college success and opportunity. Founded in 1900, the College Board was created to expand access to higher education. Today, the membership association is made up of over 6,000 of the world's leading educational institutions and is dedicated to promoting excellence and equity in education. Each year, the College Board helps more than seven million students prepare for a successful transition to college through programs and services in college readiness and college success — including the SAT® and the Advanced Placement Program®. The organization also serves the education community through research and advocacy on behalf of students, educators and schools. For further information, visit www.collegeboard.org.

00402-002