Computerized Matching of Cases to Controls

Erik J. Bergstralh
Jon L. Kosanke

Technical Report Number 56

April 1995

We wish to thank Mr. Kenneth P. Offord and Dr. Steven J. Jacobsen for valuable comments and also Ms. Sharon Wellik for secretarial assistance.

Computerized Matching of Cases to Controls

Erik J. Bergstralh
Jon L. Kosanke

Abstract

The purpose of this report is to describe a new SAS® macro, %match, written to facilitate the matching of cases to controls, where one case is matched to one or more controls. The term "case" is used rather generically and might be defined in any of the following ways depending on the study design: 1) a group of subjects exposed to a certain risk factor, 2) patients with a particular disease, or 3) patients who received a certain type of treatment. Controls would in a sense be the opposite of the cases in that they are subjects *not* exposed to the risk factor, subjects *without* the disease or patients who did *not* receive the treatment of interest. The goal is to compare the cases and controls regarding some endpoint of interest. To minimize confounding bias in these retrospective comparisons, controls are often matched to cases on one or more factors felt to be related to both exposure and the endpoint. This macro defines a distance measure between cases and potential controls which is based on the matching factors. The control chosen for a particular case is the one that is closest to the case in terms of the distance measure. Optimal matching, as described by Rosenbaum in 1989, produces the matched set with the smallest total distance between cases and matched controls. The macro has options for both greedy and optimal matching algorithms.

Table of Contents

1. Matching Problem:

In classic case-control studies we may wish to assess whether those with a disease (cases) have a higher exposure rate to some potential risk factors than those without the disease (controls). In cohort observational studies, we often wish to make an inference about the effects of a certain exposure or treatment on some endpoint of interest. In the cohort studies, we may think of the cases as those exposed to a risk factor (or those who received the treatment) and define controls as patients *not* exposed to the risk factor (or those who did *not* receive the treatment). In either setting, one may wish to match cases to controls on factors felt to be related to both exposure and the endpoint of interest.

Matching can be done for several reasons as noted by Rosenbaum and Rubin [1]. A common reason is to remove bias in assessing the effect of the exposure or treatment. For example, it may be important that the controls be of the same gender, age and have a similar smoking history as that of the cases. A second reason may be that the endpoint to be measured is very costly and it is not feasible to measure it on all N cases and M potential controls. If N is much less than M, then a matched set of controls may result in a considerable savings. A third possible reason for matching, as opposed to adjustment for case-control differences using multivariate modeling, is that it may be viewed as a more convincing method of adjustment for audiences uncomfortable with multivariate techniques.

The mechanics of how we typically select our controls can be illustrated from a typical Rochester Epidemiology Project case-control study design [2]. In this setting, our *cases* are the incident cohort of patients who develop the disease of interest during a specified time period. *Controls* are patients who were seen by a physician for any reason during the same period, who do *not* have the disease of interest. (A variation of this design might be to view the entire population as a cohort and use risk set case-control sampling as discussed in Section 6.)

We usually match our cases to the controls on gender (exact), age (± 2 years), year of birth (± 5 years) and duration of residence in the community as approximated by length of medical record (defined as the closest Mayo Clinic number). Separate lists of cases and controls are then generated for each gender and sorted by age, year of birth within age and Mayo Clinic number within year. The list of cases is scanned sequentially and the "best" available control(s) selected. The Mayo Clinic numbers of these case-control sets would then be hard-coded back into the computer. This process is very time consuming and prone to human error. In addition, it does not guarantee that one will end up with the "best" possible matched set of controls.

In the Sections that follow, we will review the concept of distance between cases and potential controls (Section 2), review the greedy matching algorithm (Section 3), and discuss the use of calipers to attain better matches (Section 4). Next, we will review the work of Rosenbaum [3] on optimal matching algorithms (Section 5), discuss the use of risk set case-control sampling (Section 6), and present a new SAS macro to implement both our past matching method (greedy algorithm) and the optimal technique (Section 7). Finally, in Section 8 we present examples on how to use the new matching macro, %match.

2. Distance Measures:

Before we can decide which control is "best" for a particular case we need to set up some notation and a definition of distance between cases and potential controls.

Let

$$\underline{X}^1 = \left\{X_1^1, X_2^1, \ldots, X_p^1\right\} \quad \text{and}$$

$$\underline{X}^0 = \left\{X_1^0, X_2^0, \ldots, X_p^0\right\}$$

be the vector of matching variables for the N cases and M ($\geq$N) potential controls respectively. Let

D_{ij} ="distance" between the i^{th} case and the j^{th} potential control.

Choices for D_{ij} might be:

(1) $D_{ij} = \sum_{k=1}^{p} \left(X_{ik}^1 - X_{jk}^0\right)^2 * W_k$, W_k is a non-negative weight associated with matching variable (k).

(2) $D_{ij} = \left(\underline{X}_i^1 - \underline{X}_j^0\right) \underline{S}^{-1} \left(\underline{X}_i^1 - \underline{X}_j^0\right)'$, where $\underline{S}$ is a pooled within group estimate of the variance-covariance matrix of the X's. This is the Mahalanobis distance.

(3) $D_{ij} = \left| P\left(X_i^1\right) - P\left(X_j^0\right) \right|$, where P( ) is the propensity score (the likelihood of being a "case"), expressed as a function of the matching variables.

(4) $D_{ij} = \sum_{k=1}^{p} \left| \text{rank}\left(X_{ik}^1\right) - \text{rank}\left(X_{jk}^0\right) \right| * W_k$, where the ranks for each X are defined using the combined group of N cases and M potential controls.

(5) $D_{ij} = \sum_{k=1}^{p} \left| X_{ik}^1 - X_{jk}^0 \right| * W_k$

The macro we have written utilizes definition 5 (the weighted sum of the absolute differences in the X's). This is not very restrictive as instead of the actual X's, one could input standardized X's, a propensity score (definition 3), or the ranks of the X's (definition 4). In addition, the weights allow the user considerable flexibility to modify the distance definition.

It is important to consider a measure of how well the entire group of cases and their controls are matched. One natural choice is the total distance (T) for a set of N matched cases.

$$T = \sum_{i=1}^{N} D_{ij}$$

The definition of T is easily extended to the situation where there are multiple controls for each case. Following definition 5, another measure might be defined using a weighted sum of the absolute differences in covariate means.

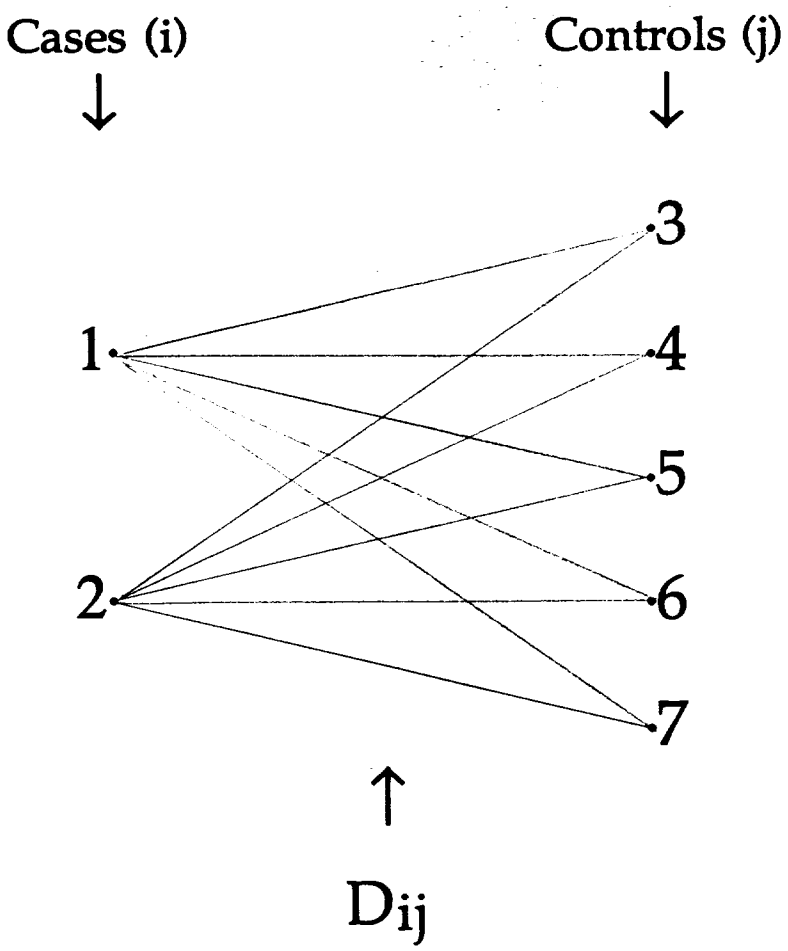
$$M = \sum_{k=1}^{p} \left| \bar{X}_k^1 - \bar{X}_k^0 \right| * W_k$$

M is a weaker measure than T as it can be small, even if some individual differences are large [3]. We have chosen T as our measure of matching efficacy and will emphasize matched sets that minimize T.

3. Greedy Matching Algorithm:

The "greedy" algorithm is essentially what is often done in matching cases to controls. Figure 1 is a graphic of how the algorithm would proceed for 2 cases and 5 potential controls. We see that D_{ij} needs to be calculated between every case and control for a total of MxN distances. The algorithm as implemented in %match is as follows:

Figure 1. Greedy algorithm: N=2 , M=5



i) Randomly sort the N cases and M controls.

ii) Match the first case in the list to the closest control, i.e. the one with the smallest D_{ij}.

iii) Move on to the second case and match it to the closest control among those remaining and repeat the process until all N cases have been matched.

With multiple controls per case, we match all cases to one control first and then make another pass through the list to match second controls. This means that case-control matches formed on the first pass are the closest.

The greedy algorithm has several properties, all of which are nicely described by Rosenbaum [3].

a. Once a match is made it is never broken.

b. Each decision made is the best among the *currently* available choices. This is why it is called "greedy," in that it makes the best decision for now, without any consideration of its future impact on the total distance, T.

c. The algorithm often runs into bottlenecks toward the end, as there are fewer controls to choose from. This implies that one needs to be careful how the data are sorted. It is recommended that both the case and control lists be randomly sorted.

d. The algorithm usually produces good matches, but it is not guaranteed to minimize the total distance, T.

4. Calipers:

One modification commonly used is to only consider controls that are within certain caliper limits of the case. These "calipers" are the same for all cases and might be based on either D_{ij} (caliper = c) or the individual X's (caliper = c_k). Formally, this means that control j is a candidate to be matched to case i only if

$$D_{ij} \leq c$$

and

$$\left| X^1_{ik} - X^0_{jk} \right| \leq c_k \quad \text{for } k = 1, 2, \ldots, p.$$

In the Rochester Epidemiology Project example given above, we often use calipers on the individual X's of 0, 2 and 5 for gender (coded 1,2), age and year of birth, respectively. Caliper matching usually produces very good matches and in fact Rosenbaum recommends its use with the propensity score [3]. However, use of calipers may result in incomplete matching (not being able to find a match for all N cases). It has been demonstrated that incomplete matching can lead to serious bias [4].

Operationally, calipers are easily incorporated into matching algorithms. One simply eliminates up-front any case-control combinations that violate the caliper restrictions. This may greatly decrease computer time as not all the MxN distances need to be evaluated.

5. Optimal Algorithm:

While the greedy algorithm works reasonably well, it is important to have an algorithm that produces the optimal set of matches, i.e., that with the smallest possible T. In 1989, Paul Rosenbaum wrote an article on optimal matching [3]. This article tied together parts of the statistics and operations research literature by using network flow theory to find the optimal solution for the case-control matching problem.

Network flow problems may involve finding the route associated with the least cost when shipping goods from a point of origin (source) to their final destination (sink). The problem is complicated by the fact that the goods must be shipped through a network of multiple intermediate destinations (nodes) before reaching the final destination. The nodes are connected by arcs (shipping routes), each of which may have a specific capacity and cost per unit shipped. Because of the capacity restrictions, the total number of units shipped may have to be divided into several lots and shipped over differing routes. This type of network is depicted in Figure 2. Many algorithms and software (PROC NETFLOW in SAS/OR®) exist for solving these types of problems.

Rosenbaum demonstrated that the matching problem could be set-up as a network flow problem [3]. The network required for two cases and five potential controls is given in Figure 3. Source and sink nodes do not have an obvious interpretation but are necessary to make the software work. The key is that each case has an arc to all possible controls and there are no case-to-case or control-to-control arcs. This means that the optimal routes identified must provide links (matches) from cases to controls. The costs and capacities for each type of arc assuming m-controls matched to each of the N cases are given below:

Table 1: Network flow set-up for each type of arc with 1-m matching.

Arc-Type	Cost	Capacity
source-to-case	0	m
case-to-control	D_{ij}	1
control-to-sink	0	1

® SAS/OR is a registered trademark of SAS Institute Inc., Cary, NC, USA.

Figure 2. Network flow problems.

Ship 1000 units of material from source to sink and minimize total cost subject to arc-specific capacity restrictions.

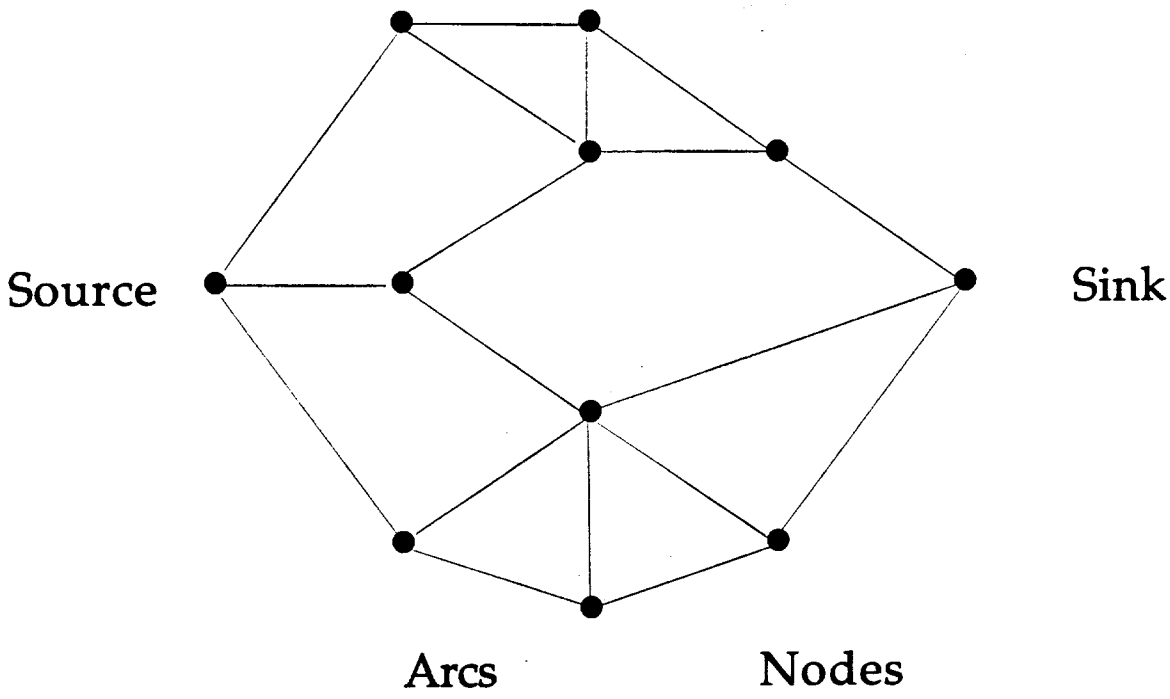
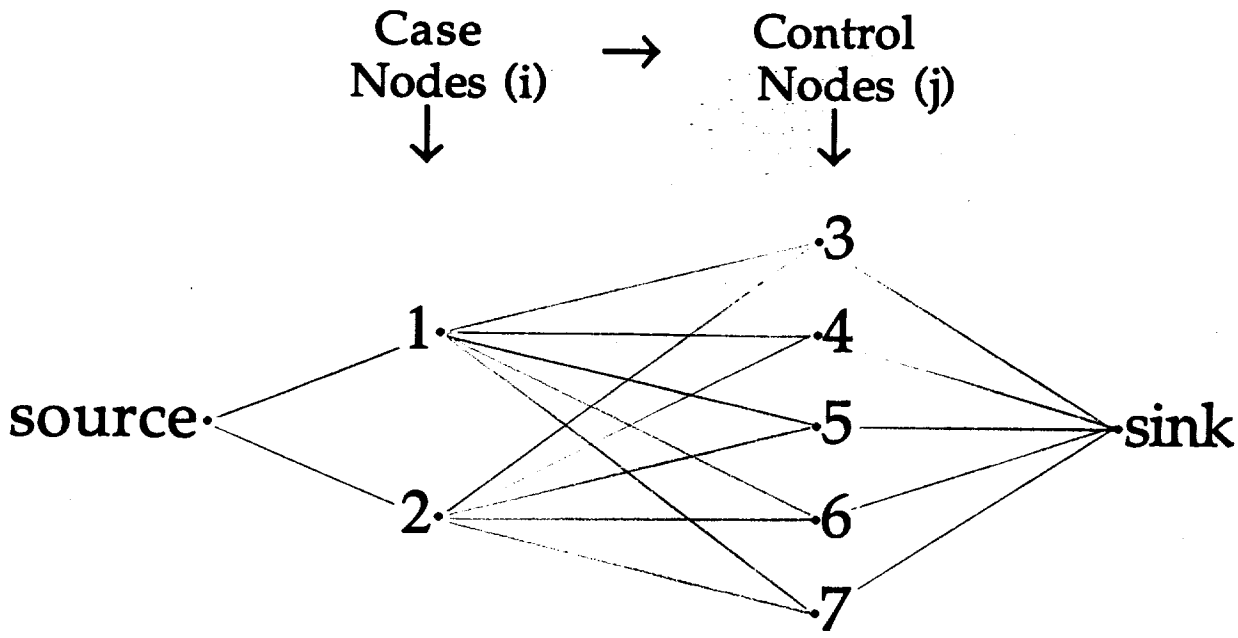


Figure 3. Matching problem as a network flow problem: N=2 , M=5



arcs

- Flow ——————→
- No within case (within control) arcs
- Each arc: cost
  capacity

One must also specify the total demand (number of units to be shipped) for the network. For 1-1 matching, this is simply the number of cases (N). For 1-m matching, the total demand is mN.

Rosenbaum also provided a solution to the problem where one wishes to perform a matched analysis and use *all* M controls and match a variable number ($\geq$1) of controls to each case [3]. This was accomplished by adding an "extra" node to the network with costs and capacities as demonstrated in Figure 4. From the user's perspective, the main difference between this problem and 1-m matching is that one needs to specify the minimum (α) and maximum (β) number of controls per case.

6. Risk Set Sampling:

Often times case-control studies are conducted within a large pre-defined *fixed* cohort of patients with some common condition. An example might be patients who received prostatectomy for prostate cancer. While the cohort may number in the thousands, perhaps fewer than 100 patients have experienced the event of interest (say death due to prostate cancer) during the follow-up period. As new prognostic markers are developed to predict outcome following prostatectomy, we have the choice of evaluating the marker on the entire cohort or doing a nested case-control study. It is often the case that such markers are expensive and/or time consuming to measure. In this setting, one might consider evaluating the marker only for the those who have died of prostate cancer (cases) and a matched set of patients who have not died of prostate cancer (controls). Selecting controls from the risk set(the patients who have not yet experienced the endpoint at the time of the case's death) is the recommended approach in these designs [5]. This design allows one to estimate an odds ratio which is an unbiased estimate of the relative risk one would have estimated using the Cox proportional hazards model with the marker evaluated for the entire cohort. A graphic of this type of sampling is displayed in Figure 5 for a pseudo cohort of size 8 (=N+M) with 3 events (N) and age as the matching variable. Note that a control at one time may be a case at a later time and that a control is matched to only one case. Using the

Figure 4. 1-m_i matching as a network flow problem: N=2 , M=5

- Use all M controls
- α = min (m_j)
- β = max (m_i)
- X = "extra" node
- (,) = (arc cost, arc capacity)

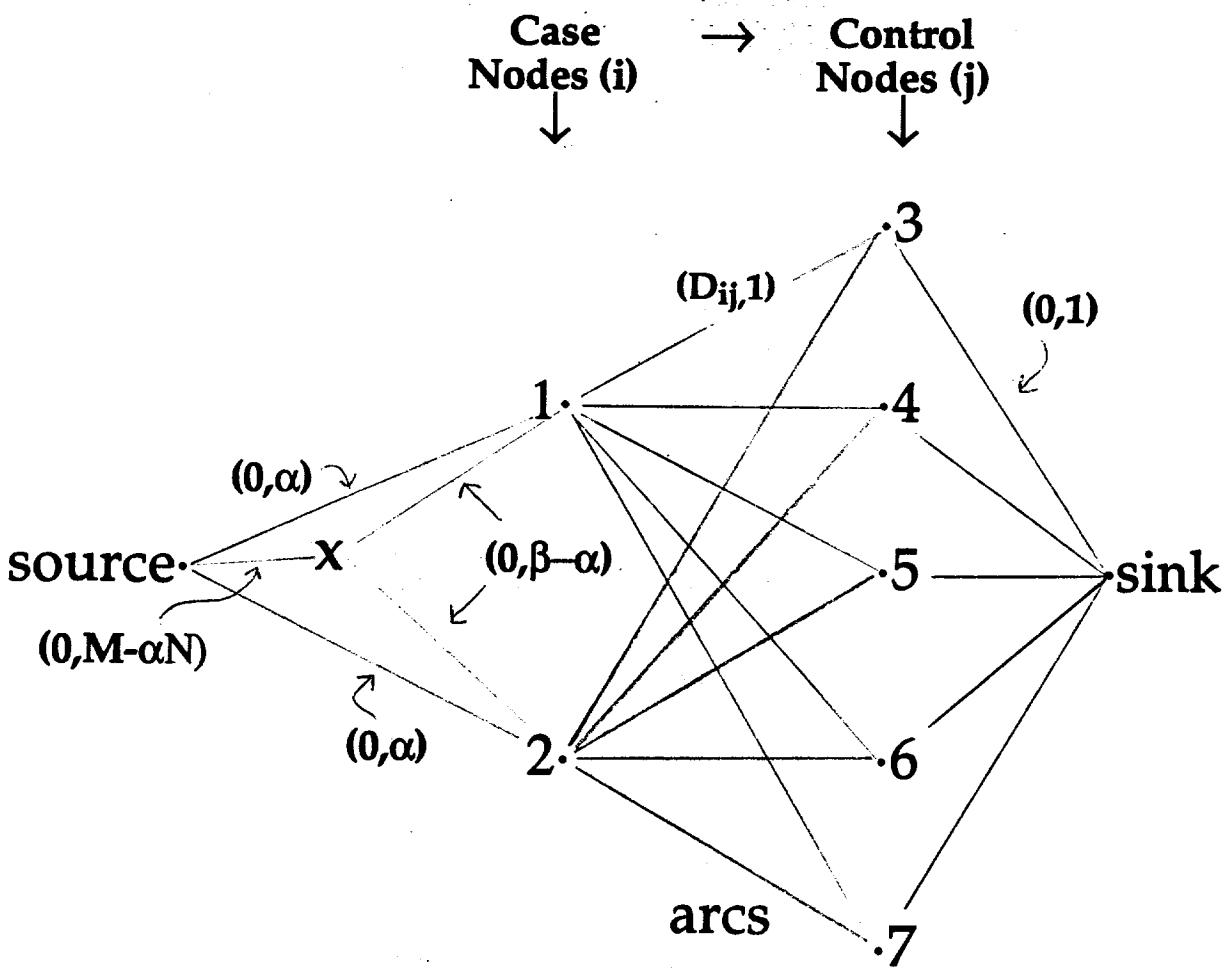
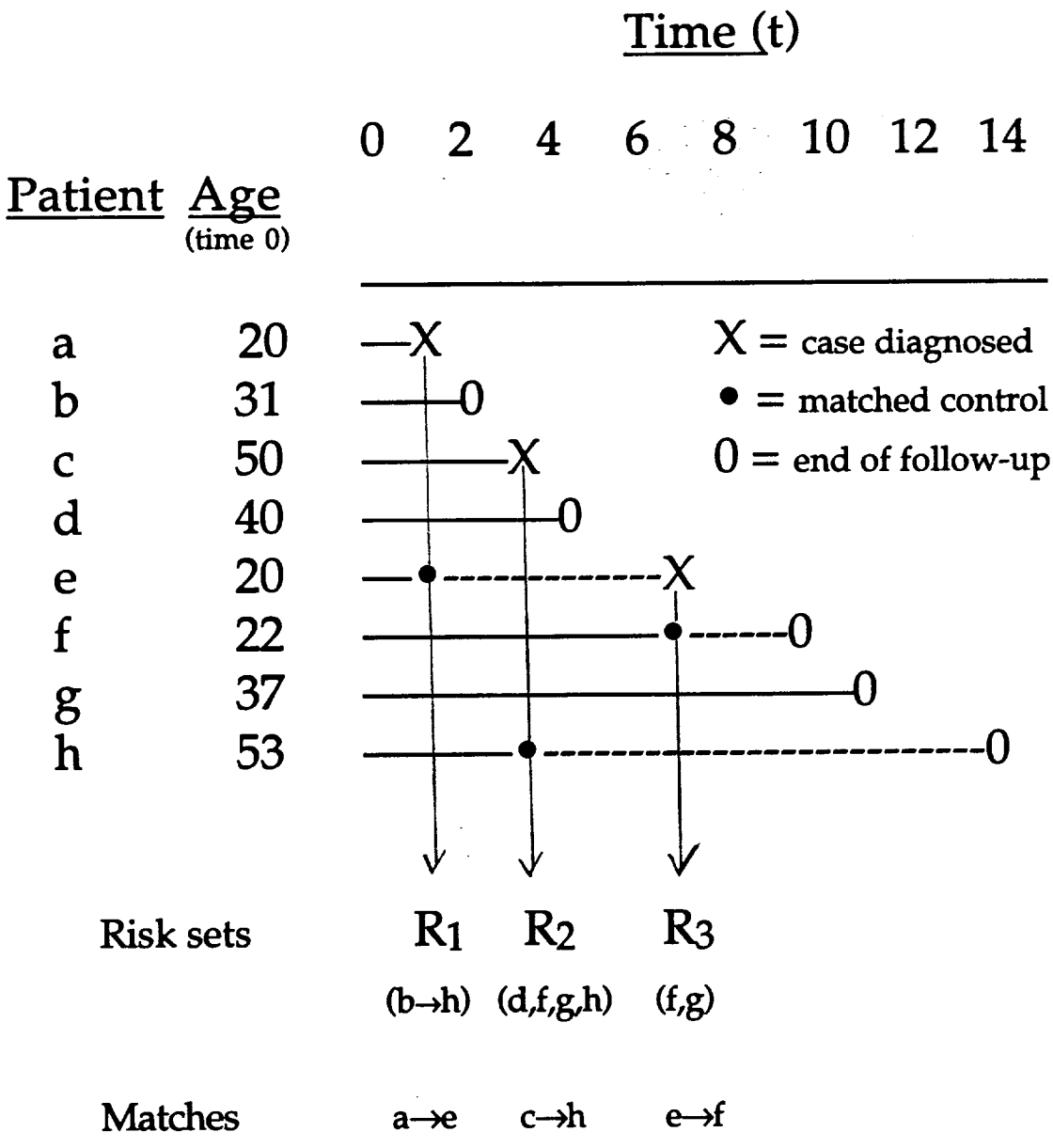


Figure 5. Selecting controls from the risk set in cohort studies.

Example: N=3 cases of prostate cancer diagnosed over a 14-year period in a cohort of 8 (N+M) subjects.



time option in %match, one can select controls from the risk set for fixed cohorts. Matching for *dynamic* cohorts (where patients may migrate into the population after the beginning of the study period) is not possible with the time option. However, one might run %match sequentially for each appropriate risk set eliminating selected controls from subsequent risk sets.

7. SAS Macro: %match:

The SAS macro %match was written to provide users with an accurate and optimal method of matching cases to controls. It utilizes the weighted sum of the absolute differences in the matching variables (definition 5 above) as its distance measure and produces an output data set containing the results of the matching. The macro has two sub-macros (%greedy, %optimal) for the greedy and optimal matching algorithms and an option to perform matching within risk sets. The optimal matching option requires SAS/OR on the system as it uses the NETFLOW procedure. Computer time required for the optimal method is proportional to MxN.

The macro assumes that you have two SAS data sets, one that includes the N cases, an ID variable, and the matching variables; and the second that includes the M potential controls, an ID variable, and the matching variables. The matching variables must have the same names in each data set.

The macro call statement is as follows:

```
%match(case=,control=,idca=,idco=,
       mvars=,wts=,dmaxk=,dmax=,
       method=,
       ncontls=,seedca=,seedco=,
       mincont=,maxcont=,maxiter=,
       out=,outnmca=,outnmco=,print=);
```

A brief definition of the parameters is given below. The entire macro with complete definitions is given in the Appendix (R=required parameter):

(R) case = SAS data set for cases.

(R) control = SAS data set of possible controls.

(R) idca = ID variable for the cases.

(R) idco = ID variable for the controls.

(R) mvars = list of numeric matching variables common and identically named in both the case and control data sets. For example, **mvars**=male age birthyr .

(R) wts = list of non-negative weights corresponding to each matching variable. For example wts=10 2 1 might correspond to male, age and birthyr, respectively.

dmaxk = calipers for individual x's. Using the above example and specifying dmaxk = 0 2 5 results in exact matches on gender, matches $\leq$2 years on age, and $\leq$5 years on birth year.

dmax = caliper for the D_{ij}, the weighted distance summed over all matching variables.

time = time variable used to define risk sets for risk-set sampling with fixed cohorts. Matches are valid only if control time > case time.

(R) method = **GREEDY** or **OPTIMAL**. Defines which matching algorithm to use.

ncontls = fixed number of controls to match to each case. This option is ignored if the mincont and maxcont options are used.

**** Options specific to **GREEDY** method **********

(R) seedca = positive integer seed value used to randomly sort the cases prior to matching.

(R) seedco = positive integer seed value used to randomly sort the controls prior to matching. Use different values for seedca and seedco.

**** Options specific to **OPTIMAL** method **********

mincont = minimum number of controls (α) per case. Used only for matching *all* controls as described in Section 5.

maxcont = maximum number of controls (β) per case. Used only for matching *all* controls as described in Section 5.

maxiter = maximum number of iterations for PROC NETFLOW to use under the optimal method. Default value is 10000.

**** OUTPUT options applicable to either method **********

print = Option to print data for matched cases. Use **print=y** to print data and **print=n** for no printout.

out = name of SAS data set containing the results of the matching process. Unmatched cases are not included.

outnmca = name of SAS data set containing *non-matched cases.*

outnmco = name of SAS data set containing *non-matched controls.*

8. Examples:

A. Graves' Ophthalmopathy and Sinusitis.

Patients treated with orbital decompression for Graves' ophthalmopathy may experience sinusitis at some time following surgery. Among a cohort of 428 such patients, 86 of the 383 survivors were identified as having had subsequent sinusitis [6]. The investigators wished to identify risk factors for sinusitis by sending a questionnaire to the 86 cases and a set of matched controls selected from the 297 patients without known sinusitis. The matching factors were to be gender, age at time of orbital decompression and calendar year of orbital decompression. We wished to match exactly on gender and felt it to be at least twice as important to match on age as compared to calendar year. With this in mind, arbitrary weights of 20, 2, and 1 were assigned to gender, age, and calendar year, respectively.

Our cases are in a data set named "case", the potential controls in a data set named "cont", and the matching variables are named male, age_od and yr_od for male gender (coded 0=female, 1=male), age and year of orbital decompression, respectively. With calipers of 0, 2 and 5 and weights of 20, 2 and 1 for gender, age and year, respectively, the following SAS code would perform the matching using the **optimal** algorithm:

```
%match(case=case, control=cont, idca=clinic, idco=clinic,
    mvars= male age_od yr_od,
    wts= 20 2 1,
    dmaxk=0 2 5,
    method=optimal,
    maxiter=10000);
```

Note that weight for male gender (20) is irrelevant as its caliper is 0. This means that the contribution of gender to D_{ij} and the total distance T will always be "$|0|*w_k$", or 0. Using the default print=Y option the printed output for this example is displayed in Figures 6 (partial list of matches) and 7 (summary data for matched cases and controls). The data listing provides the I.D. number of the case and its matched control(s), the weighted distance (D_{ij}), the absolute difference for each of the matching variables (male,age_od,yr_od) and the actual values of the matching variables for both the case and its control(s). This listing is a PRINT of the output data set (_out). The listing is sorted by the id number of the cases. Sorting the listing by descending D_{ij} may also be useful as a means to quickly scan the poorest matches.

The summary data (Figure 7) is a MEANS on D_{ij} , the absolute differences in each of the matching variables and the actual value of the matching variables for cases and controls. The "N" column indicates how many cases were matched. The total distance, T, is displayed under the "Sum" for D_{ij}. In this example, the total distance was 197 and 83 of 86 cases were matched. The code to run the same example using the **greedy** algorithm would be:

Figure 6. Optimal Matching with calipers.

Data listing for matched cases and controls.

OBS	clinic CASE	clinic CONTROL	CONTROL NUMBER	WEIGHTED DIFFERENCE	MALE ABS. DIFF	age_od ABS. DIFF	yr_od ABS. DIFF	male CASE	male CONTROL	age_od CASE	age_od CONTROL	yr_od CASE	yr_od CONTROL
1	11	12	1	4	0	0	4	0	0	55	55	1971	1975
2	13	22	1	0	0	0	0	0	0	53	53	1974	1974
3	15	6	1	7	0	1	5	1	1	56	55	1970	1975
4	16	4	1	0	0	0	0	0	0	53	53	1977	1977
5	17	29	1	2	0	1	0	0	0	51	50	1972	1972
	.	.											
	.	.											
	.	.											
	.	.											
	.	.											
81	19	2	1	3	0	1	1	0	0	52	51	1987	1988
82	21	8	1	2	0	0	2	0	0	56	56	1988	1986
83	23	14	1	0	0	0	0	0	0	47	47	1988	1988
				197									

```
match macro: case=case   control=cont   idca=clinic  idco=clinic
      mvars=male age_od yr_od   wts=20 2 1 dmaxk=0 2 5 dmax=   ncontls=1
      method=optimal   seedca=   seedco=
      out=mtch    outnmca=_nmca    outnmco=_nmco
```

Figure 7. Optimal matching with calipers.

Summary data for matched cases and controls.

__CONT_N	N Obs	Variable	Label	N	Mean	Sum	Minimum	Maximum
1	83	__DIJ	WEIGHTED/DIFFERENCE	83	2.37	197.00	0	9.00
		__DIF1	male/ABS. DIFF	83	0	0	0	0
		__DIF2	age_od/ABS. DIFF	83	0.52	43.00	0	2.00
		__DIF3	yr_od/ABS. DIFF	83	1.34	111.00	0	5.00
		__CA1	male/CASE	83	0.17	14.00	0	1.00
		__CA2	age_od/CASE	83	47.72	3961.00	22.00	77.00
		__CA3	yr_od/CASE	83	1978.87	164246.00	1969.00	1988.00
		__CO1	male/CONTROL	83	0.17	14.00	0	1.00
		__CO2	age_od/CONTROL	83	47.86	3972.00	23.00	77.00
		__CO3	yr_od/CONTROL	83	1979.02	164259.00	1971.00	1988.00


```
match macro: case=case   control=cont   idca=clinic  idco=clinic
      mvars=male age_od yr_od   wts=20 2 1 dmaxk=0 2 5 dmax=   ncontls=1
      method=optimal   seedca=   seedco=
      out=mtch    outnmca=_nmca    outnmco=_nmco
```

```
%match(case=case, control=cont, idca=clinic, idco=clinic,
        mvars= male age_od yr_od,
        wts= 20 2 1,
        dmaxk=0 2 5,
        method=greedy,
        seedca=87877,
        seedco=987973).
```

The total distance, T, for the greedy method with calipers was 209, about 6% higher than the optimal method. However, the greedy algorithm matched one less case (82) as compared to optimal matching (83). The results of these two examples are displayed in Table 2 (right side), along with another set of examples where calipers were not used (left side). The SAS code for *not* using calipers is identical to that given above, except that the "dmaxk" option is not used. We see that not using calipers results in all cases being matched and a total distance of 251 for the greedy method and 228 for optimal (a 10% improvement). In the power plant example presented by Rosenbaum [3] an 11% improvement was noted for optimal matching. The greedy method was repeated 10 times for the current example (using different case and control sorting seeds) with a mean D_{ij} of 242 (6% higher than optimal) and a range from 235 to 250. The examples in Table 2 without caliper had 97% of the matches within the age caliper and 98% within the year caliper. Note that the weight of 20 for gender (compared to 2 and 1 for age and year) produced exact matching on gender for both algorithms without using calipers.

Table 2: Matching results for subsets of Graves' ophthalmopathy patients (86 cases with sinusitis, 297 controls without) illustrating use of calipers and type of algorithm.*

	No Calipers		Calipers (gender = 0, age ≤ 2, year ≤ 5)	
	Algorithm		Algorithm	
Measure	Greedy	Optimal	Greedy	Optimal
Number matched	86	86	82	83
Distance (D_{ij}):				
total	251**	228	209	197
mean	2.92	2.65	2.52	2.37
maximum	17	14	9	9
Gender:				
% matched exactly	100	100	100	100
Age, years:				
cases, mean	46.7	46.8	48.0	47.7
controls, mean	46.8	46.9	48.0	47.8
mean |difference|***	0.7	0.6	0.6	0.5
maximum |difference|	7	7	2	2
% |difference| ≤2	97	97	100	100
Calendar year:				
cases, mean	1978.8	1978.8	1978.8	1978.9
controls, mean	1979.2	1979.0	1979.1	1979.0
mean |difference|	1.5	1.4	1.4	1.3
maximum |difference|	8	8	5	5
% |difference| ≤5	98	97	100	100

* Weights = 20,2,1 for gender, age and year respectively.

** A mini-simulation study was done by running the greedy algorithm 10 times with different seeds for sorting cases and controls. Total distance averaged 6% higher(mean=242, range 235 to 250) than the optimal method.

*** |difference| = Absolute value of the difference between case age minus control age with similar definitions for calendar year.

B. Risk-Set Sampling Example.

Using the made-up data presented in Section 6 and Figure 5, the SAS code for setting up risk set sampling in a fixed cohort and the output data listing are displayed in Figure 8. Note that the "control" data set must contain the entire cohort (both cases and non-cases) and that the case data set is defined in the usual manner (includes only the cases). The "time" option must be used with the time variable being the follow-up time for non-cases and the time of the event for the cases. On the output listing we see that case *a* is matched to control *e* (who later became a case), case *c* to control *h* and case *e* to control *f*.

9. Conclusion:

%match has proven to be a useful tool in the design of our controlled retrospective studies. It eliminates the need to manually select controls. In doing so it gives the user the flexibility to try different distance measures, weighting factors, matching variables and matching algorithms. Our experience and that of Rosenbaum [3] would suggest that optimal matching produces matched sets that are 5-10% "closer" than those defined using the greedy algorithm. Using calipers produces good matches, however, one needs to be sure that the level of incomplete matching remains low in order to keep bias to a minimum.

Figure 8. Risk set matching.

a. Code:

```
data a;        * all data (cases+non-cases);
               * t = time of event (evt=1),
input id $1. t evt age;
cards;
a    1  1   20
b    2  0   31
c    3  1   50
d    5  0   40
e    7  1   20
f   11  0   22
g   12  0   37
h   13  0   53
proc print;

data b; set a; ** cases;
    if evt=1;
proc print;

%match (case=b, control=a, idca=id, idco=id,
        method=optimal, time=t,
    mvars=age, wts=1);
```

b. Output:

Data listing for matched cases and controls -

OBS	id CASE	id CONTROL	CONTROL NUMBER	t CASE	t CONTROL	DISTANCE D_IJ	age ABS.DIFF	age CASE	age CONTROL
1	a	e	1	1	7	0	0	20	20
2	c	h	1	3	13	3	3	50	53
3	e	f	1	7	11	2	2	20	22
						5			

10. References:

1. Rosenbaum PR and Rubin D: Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. The American Statistician, 39:33-38, 1985.

2. O'Fallon WM, Beard CM, Weigel KM, Kurland LT: The clinical record as an epidemiological data source: The Rochester Project experience. Proc. of the Fourteenth Hawaii International Conference on System Sciences. Vol. II, pp. 123-143, 1981.

3. Rosenbaum PR: Optimal matching for observational studies. Journal American Statistical Association, 84(408):1024-1032, 1989.

4. Rosenbaum PR: The bias due to incomplete matching. Biometrics, 41:106-116, 1985.

5. Lubin JH and Gail MH: Biased selection of controls for case-control analyses of cohort studies. Biometrics, 40:63-75, 1984.

6. Garrity JA, Fatourechi V, Bergstralh EJ, et al: Results of orbital decompression in 428 patients with severe Graves' ophthalmopathy. American Journal Ophthalmopathy, 116: 533-547, 1993.

APPENDIX

```
/*
Macro name: %match

Authors: Jon Kosanke and Erik Bergstralh

Date: April 25, 1995

Macro function:

   The purpose of this macro is to match 1 or more controls(from a total
   of M) for each of N cases.  The controls may be matched to the cases by
   one or more factors(X's).  The control selected for a particular
   case(i) will be the control(j) closest to the case in terms of Dij.
   Dij is just the weighted sum of the absolute differences between the
   case and control matching factors.  I.e.,

       Dij= SUM ( W.k*ABS(X.ik-X.jk) ), where the sum is over the number
                                        of matching factors X(with index
                                        k) and W.k = the weight assigned
                                        to matching factor k and X.ik =
                                        the value of variable X(k) for
                                        subject i.

   The control(j) selected for a case(i) is that with the smallest Dij
   which is less than or equal DMAX(and which is compatible with the DMAXK
   option below), where DMAX is defined by the user.  In the case of ties,
   the first one encountered will be used.  The higher the user-defined
   weight, the more likely it is that the case and control will be matched
   on the factor.  Assign large weights (relative to the other weights) to
   obtain exact matches for two-level factors such as gender.

   Using the GREEDY method, once a match is made it is never broken.  This
   may result in inefficiencies if a previously matched control would be a
   better match for the current case than those controls currently
   available.

   The OPTIMAL method uses PROC NETFLOW from SAS/OR to find the set of
   matches that minimizes the sum of Dij over all possible sets of
   matches.  The OPTIMAL method also has an option for a variable number
   of controls per case.

Call statement:
%match(case=,control=,idca=,idco=,
       mvars=,wts=,dmaxk=,dmax=,
       time=,
       method=,
       ncontls=,seedca=,seedco=,
       mincont=,maxcont=,maxiter=,
       out=,outnmca=,outnmco=,print=);

Parameter definitions(R=required parameter):

  R     case=SAS data set of cases.  Must contain the IDCA variable and
        the matching variables.

  R     control=SAS data set of possible controls.  Must contain the IDCO
        variable and the matching variables.  Note the macro assumes
        that the cases and controls are in different data sets.
        FOR RISK SET MATCHING THIS DATA SET SHOULD INCLUDE BOTH CASES
        AND CONTROLS.

  R     idca=ID variable for the cases.

  R     idco=ID variable for the controls.

  R     time=time variable used to define risk sets.  Matches are only
        valid if the control time > case time.

  R     mvars=list of numeric matching variables common to both case and
        control data sets.  For example, mvars=male age birthyr.

  R     wts=list of non-negative weights corresponding to each matching
        variable.  For example wts=10 2 1 corresponding to male, age
        and birthyr as in the above example.

        dmaxk=list of non-negative values corresponding to each matching
        variable.  These numbers are the largest possible absolute
        differences compatible with a valid match.  Cases will NOT be
        matched to a control if ANY of the INDIVIDUAL matching factor
        differences are >DMAXK.  This optional parameter allows
        one to form matches of the type male+/-0, age+/-2, birth
        year+/-5 by specifying DMAXK=0 2 5.  Given that
        a possible control meets the DMAXK criteria, the macro
        selects the control with the smallest Dij.  If this list
        is shorter that the MVARS list, 1-1 matching will be done
        until the DMAXK list is exhausted.  If this list is
        longer, the extra DMAXK values are ignored.


        dmax=largest value of Dij considered to be a valid match.  If you
        want to match exactly on a two-level factor(such as gender
        coded as 0 or 1) then assign DMAX to be less than the weight
        for the factor.  In the example above, one could use wt=10 for
        male and dmax=9.  Leave DMAX blank if any Dij is a valid match.
        One would typically NOT use both DMAXK and DMAX.  The only
        advantage to using both, would be to further restrict potential
        matches that meet the DMAXK criteria.

  R     method= GREEDY or OPTIMAL.  See reference below.

        ncontls=fixed number of controls to match to each case.  The
        default is 1.  Using the GREEDY method with multiple
        controls per case, the algorithm will first match every
        case to one control and then again match each case to a
        second control, etc.  Controls selected on the first pass
        will be stronger matches than those selected in later
        rounds.  The output data set contains a variable(cont_n)
        which indicates on which round the control was selected.
        This option is ignored if a variable number of controls is
        to be used with the OPTIMAL method(see MINCONT and MAXCONT
        parameters below).

        **** Options specific to GREEDY method ****************************
  R     seedca=seed value used to randomly sort the cases prior to matching
        using the GREEDY method.  This positive integer must be less
        than (2**31)-1 and will be used as input to the RANUNI
        function.  The greedy matching algorithm is order dependent
        which, among other things means that cases matched first
        will be on average more similar to their controls than those
        matched last(as the number of control choices will be
        limited).  If the matching order is related to confounding
        factors (possibly age or calendar time) then biases may
        result.  Therefore it is generally considered good practice
        when using the GREEDY method to randomly sort both the cases
        and controls before beginning the matching process.

  R     seedco=seed value used to randomly sort the controls prior to
        matching using the GREEDY method.  This seed value must
        also be an integer less than (2**31)-1.
```

```
      **** Options specific to OPTIMAL method ***************************
      mincont=minimum number of controls per case using the OPTIMAL method
            with a variable number of controls(see Section 3.3 of
            Rosenbaum).  MINCONT must be >=1.

      maxcont=maximum number of controls per case using the OPTIMAL method
           with a variable number of controls(see Section 3.3 of Rosenbaum).
           MAXCONT must be >= MINCONT and <= M-N+1.

      maxiter=maximum number of iterations for PROC NETFLOW to use under
              the OPTIMAL method.  Default value is 10000.

      **** OUTPUT options applicable to either method ******************
      print= Option to print data for matched cases. Use PRINT=y to
            print data and PRINT=n or blank to not print.  Default is y.

      out=name of SAS data set containing the results of the matching
          process.  Unmatched cases are not included.  See outnm below.
          The default name is __out.  This data set will have the
          following layout:

          Case_id   Cont_id  Cont_n   Dij  Delta_caco MVARS_ca  MVARS_co
             1        67       1       5.2 (Differences & actual values for
             1        78       2       6.1  matching factors for cases &
             2        52       1       2.9  controls)
             2        92       2       3.1
             .        .        .       .
             .        .        .       .

      outnmca=name of SAS data set containing NON-matched cases.
              Default name is __nmca .

      outnmco=name of SAS data set containing NON-matched controls.
              Default name is __nmco .


References: Bergstralh, EJ and Kosanke JL(1995).  Computerized matching
            of cases to controls.  Section of Biostatistics Technical
            Report 56.  Mayo Foundation.

            Paul R.  Rosenbaum.  Optimal matching for observational
            studies.  JASA, 84(408), pp. 1024-1032, 1989.

Example: 1-1 matching by male(exact), age(+-2) and year(+-5).
         The wt for male is not relevant, as only exact matches
         on male will be considered.  The weight for age(2) is
         double that for year(1).

    A. Optimal method.

     %match(case=case,control=cont,idca=clinic,idco=clinic,
           mvars=male age_od yr_od,maxiter=10000,
           wts=2 2 1, dmaxk=0 2 5,out=mtch,
           method=optimal);

    B. Greedy method.

     %match(case=case,control=cont,idca=clinic,idco=clinic,
           mvars=male age_od yr_od,
           wts=2 2 1, dmaxk=0 2 5,out=mtch,
           method=greedy,seedca=87877,seedco=987973);

  ************************************************************** */

%MACRO MATCH(CASE=,CONTROL=,IDCA=,IDCO=,MVARS=,WTS=,DMAXK=,DMAX=,NCONTLS=1,
           TIME=,
           METHOD=,SEEDCA=,SEEDCO=,MAXITER=10000,PRINT=y,
           OUT=__out,OUTNMCA=__nmca,OUTNMCO=__nmco,MINCONT=,MAXCONT=);

 %LET BAD=0;
 %IF %LENGTH(&CASE)=0 %THEN %DO;
    %PUT 'ERROR: NO CASE DATASET SUPPLIED';
    %LET BAD=1;
 %END;
 %IF %LENGTH(&CONTROL)=0 %THEN %DO;
    %PUT 'ERROR: NO CONTROL DATASET SUPPLIED';
    %LET BAD=1;
 %END;
 %IF %LENGTH(&IDCA)=0 %THEN %DO;
    %PUT 'ERROR: NO IDCA VARIABLE SUPPLIED';
    %LET BAD=1;
 %END;
 %IF %LENGTH(&IDCO)=0 %THEN %DO;
    %PUT 'ERROR: NO IDCO VARIABLE SUPPLIED';
    %LET BAD=1;
 %END;
 %IF %LENGTH(&MVARS)=0 %THEN %DO;
    %PUT 'ERROR: NO MATCHING VARIABLES SUPPLIED';
    %LET BAD=1;
 %END;
 %IF %LENGTH(&WTS)=0 %THEN %DO;
    %PUT 'ERROR: NO WEIGHTS SUPPLIED';
    %LET BAD=1;
 %END;
 %IF %UPCASE(&METHOD)=GREEDY %THEN %DO;
    %IF %LENGTH(&SEEDCA)=0 %THEN %DO;
       %PUT 'ERROR: NO SEEDCA VALUE SUPPLIED';
       %LET BAD=1;
    %END;
    %IF %LENGTH(&SEEDCO)=0 %THEN %DO;
       %PUT 'ERROR: NO SEEDCO VALUE SUPPLIED';
       %LET BAD=1;
    %END;
 %END;
 %IF %LENGTH(&OUT)=0 %THEN %DO;
    %PUT 'ERROR: NO OUTPUT DATASET SUPPLIED';
    %LET BAD=1;
 %END;
 %IF %UPCASE(&METHOD)^=GREEDY & %UPCASE(&METHOD)^=OPTIMAL %THEN %DO;
    %PUT 'ERROR: METHOD MUST BE GREEDY OR OPTIMAL';
    %LET BAD=1;
 %END;
 %if (&mincont=  and &maxcont^= ) or (&mincont^=  and &maxcont= ) %then %do;
    %put 'ERROR: MINCONT AND MAXCONT MUST BOTH BE SPECIFIED';
    %let bad=1;
 %end;
 %LET NVAR=0;
 %DO %UNTIL(%SCAN(&MVARS,&NVAR+1,' ')= );
    %LET NVAR=%EVAL(&NVAR+1);
 %END;
 %LET NWTS=0;
 %DO %UNTIL(%SCAN(&WTS,&NWTS+1,' ')= );
    %LET NWTS=%EVAL(&NWTS+1);
 %END;
 %IF &NVAR^= &NWTS %THEN %DO;
    %PUT 'ERROR: #VARS MUST EQUAL #WTS';
    %LET BAD=1;
 %END;
 %LET NK=0;
 %IF &DMAXK^=  %THEN %DO %UNTIL(%SCAN(&DMAXK,&NK+1,' ')= );
```

```
   %LET NK=%EVAL(&NK+1);
%END;
%IF &NK>&NVAR %THEN %LET NK=&NVAR;
%DO I=1 %TO &NVAR;
   %LET V&I=%SCAN(&MVARS,&I,' ');
%END;
%DO I=1 %TO &NWTS;
   %LET W&I=%SCAN(&WTS,&I,' ');
   %IF &&W&I<0 %THEN %DO;
      %PUT 'ERROR: WEIGHTS MUST BE NON-NEGATIVE';
      %LET BAD=1;
   %END;
%END;
%DO I=1 %TO &NK;
   %LET K&I=%SCAN(&DMAXK,&I,' ');
   %IF &&K&I<0 %THEN %DO;
      %PUT 'ERROR: DMAXK VALUES MUST BE NON-NEGATIVE';
      %LET BAD=1;
   %END;
%END;
%MACRO DIJ;
   %DO I=1 %TO &NVAR-1;
      &&W&I*ABS(__CA&I-__CO&I) +
   %END;
   &&W&NVAR*ABS(__CA&NVAR-__CO&NVAR);
%MEND DIJ;
%MACRO MAX1;
   %IF &DMAX^= %THEN %DO;
      & __D<=&DMAX
   %END;
   %DO I=1 %TO &NK;
      & ABS(__CA&I-__CO&I)<=&&K&I
   %END;
%MEND MAX1;
%MACRO MAX2;
   %IF &DMAX= & &NK=0 %THEN %DO;
      %IF &time^= %then %do;
         if __cotime>__catime then
      %end;
      output;
   %end;
   %IF &DMAX^= & &NK=0 %THEN %DO;
      IF _COST_<=&DMAX
      %if &time^= %then %do;
         & __cotime>__catime
      %end;
      THEN OUTPUT;
   %END;
   %IF &DMAX= & &NK>0 %THEN %DO;
      IF ABS(__CA1-__CO1)<=&K1
      %DO I=2 %TO &NK;
         & ABS(__CA&I-__CO&I)<=&&K&I
      %END;
      %if &time^= %then %do;
         & __cotime>__catime
      %end;
      THEN OUTPUT;
   %END;
   %IF &DMAX^= & &NK>0 %THEN %DO;
      IF _COST_<=&DMAX
      %DO I=1 %TO &NK;
         & ABS(__CA&I-__CO&I)<=&&K&I
      %END;
      %if &time^= %then %do;
         & __cotime>__catime
      %end;
      THEN OUTPUT;
   %END;
%MEND MAX2;
%MACRO LBLS;
   %DO I=1 %TO &NVAR;
      __CA&I="&&V&I/CASE"
      __CO&I="&&V&I/CONTROL"
      __DIF&I="&&V&I/ABS. DIFF "
      __WT&I="&&V&I/WEIGHT"
   %END;
%MEND LBLS;
%MACRO VBLES;
   %DO I=1 %TO &NVAR;
      __DIF&I
   %END;
   %DO I=1 %TO &NVAR;
      __CA&I __CO&I
   %END;
%MEND VBLES;
%MACRO GREEDY;
 %GLOBAL BAD2;
   DATA __CASE; SET &CASE END=EOF;
    KEEP __IDCA __CA1-__CA&NVAR __R &mvars
       %if &time^= %then %do;
          __catime
       %end;
       ;
      __IDCA=&IDCA;
      %if &time^= %then %do;
         __catime=&time;
      %end;
      %DO I=1 %TO &NVAR;
         __CA&I=&&V&I;
      %END;
      SEED=&SEEDCA;
      __R=RANUNI( SEED  );
      IF EOF THEN CALL SYMPUT('NCA',_N_);
   PROC SORT; BY __R __IDCA;
   DATA __CONT; SET &CONTROL END=EOF;
    KEEP __IDCO __CO1-__CO&NVAR __R &mvars
     %if &time^= %then %do;
        __cotime
     %end;
     ;
      __IDCO=&IDCO;
      %if &time^= %then %do;
         __cotime=&time;
      %end;
      %DO I=1 %TO &NVAR;
         __CO&I=&&V&I;
      %END;
      SEED=&SEEDCO;
      __R=RANUNI( SEED  );
      IF EOF THEN CALL SYMPUT('NCO',_N_);
   RUN;
   %LET BAD2=0;
   %IF &NCO < %EVAL(&NCA*&NCONTLS) %THEN %DO;
      %PUT 'ERROR: NOT ENOUGH CONTROLS TO MAKE REQUESTED MATCHES';
      %LET BAD2=1;
   %END;
   %IF &BAD2=0 %THEN %DO;
      PROC SORT; BY __R __IDCO;
      DATA __MATCH;
       KEEP __IDCA __CA1-__CA&NVAR __DIJ __MATCH __CONT_N;
```

```
      ARRAY __USED(&NCO) $ 1 _TEMPORARY_;
        DO __I=1 TO &NCO;
          __USED(__I)='0';
        END;
        DO __I=1 TO &NCONTLS;
          DO __J=1 TO &NCA;
            SET __CASE POINT=__J;
            __SMALL=.;
            __MATCH=.;
            DO __K=1 TO &NCO;
              IF __USED(__K)='0' THEN DO;
                SET __CONT POINT=__K;
                __D=%DIJ
                IF __d^=. & (__SMALL=. | __D<__SMALL) %MAX1
                %if &time^= %then %do;
                  & __cotime > __catime
                %end;
                THEN DO;
                  __SMALL=__D;
                  __MATCH=__K;
                  __DIJ=__D;
                  __CONT_N=__I;
                END;
              END;
            END;
            IF __MATCH^=. THEN DO;
              __USED(__MATCH)='1';
              OUTPUT;
            END;
          END;
        END;
        STOP;
      DATA &OUT;
       SET __MATCH;
       SET __CONT POINT=__MATCH;
       KEEP __IDCA __IDCO __CONT_N __DIJ __CA1-__CA&NVAR
            __CO1-__CO&NVAR __DIF1-__DIF&NVAR __WT1-__WT&NVAR
            %if &time^= %then %do;
              __catime __cotime
            %end;
            ;
       LABEL __IDCA="&IDCA/CASE"
             __IDCO="&IDCO/CONTROL"
             %if &time^= %then %do;
               __catime="&time/CASE"
               __cotime="&time/CONTROL"
             %end;
             __CONT_N='CONTROL/NUMBER'
             __DIJ='DISTANCE/D_IJ'
             %LBLS;
        %DO I=1 %TO &NVAR;
          __DIF&I=abs(__CA&I-__CO&I);
          __WT&I=&&W&I;
        %END;
    %END;
%MEND GREEDY;
%MACRO OPTIMAL;
 %GLOBAL BAD2;
   DATA __CASE; SET &CASE END=EOF;
    KEEP __IDCA __CA1-__CA&NVAR &mvars
      %if &time^= %then %do;
        __catime
      %end;
      ;
      __IDCA=&IDCA;
      %if &time^= %then %do;
        __catime=&time;
      %end;
      %DO I=1 %TO &NVAR;
        __CA&I=&&V&I;
      %END;
      IF EOF THEN CALL SYMPUT('NCA',_N_);
   DATA __CONT; SET &CONTROL END=EOF;
    KEEP __IDCO __CO1-__CO&NVAR &mvars
      %if &time^= %then %do;
        __cotime
      %end;
      ;
      __IDCO=&IDCO;
      %if &time^= %then %do;
        __cotime=&time;
      %end;
      %DO I=1 %TO &NVAR;
        __CO&I=&&V&I;
      %END;
      IF EOF THEN CALL SYMPUT('NCO',_N_);
   RUN;
   %LET BAD2=0;
   %IF &NCO < %EVAL(&NCA*&NCONTLS) %THEN %DO;
      %PUT 'ERROR: NOT ENOUGH CONTROLS TO MAKE REQUESTED MATCHES';
      %LET BAD2=1;
   %END;
   %IF &BAD2=0 %THEN %DO;
      DATA __DIST1;
       SET __CASE;
       LENGTH __FROM __TO $ 80;
         DO I=1 TO &NCO;
           SET __CONT POINT=I;
           _COST_=%DIJ;
           __FROM=left(__IDCA);
           __TO=left(trim(__IDCO) || '_co');
           _CAPAC_=1;
           IF _COST_^=. THEN DO;
             %MAX2
           END;
         END;
         DATA __GOODCO;
          SET __DIST1;
          KEEP __IDCO;
         PROC SORT; BY __IDCO;
         DATA __GOODCO;
          SET __GOODCO; BY __IDCO;
            IF FIRST.__IDCO;
         data _null_;
           i=1;
           set __goodco point=i nobs=n;
           call symput('newcont',n);
           stop;
         DATA __DIST2;
          LENGTH __FROM __TO $ 80;
            DO I=1 TO N;
              SET __GOODCO POINT=I NOBS=N;
              __FROM=left(trim(__IDCO) || '_co');
              __TO='SK';
              _COST_=0;
              _CAPAC_=1;
              OUTPUT;
            END;
         STOP;
         DATA __GOODCA;
```

```
      SET __DIST1;
      KEEP __IDCA;
    PROC SORT; BY __IDCA;
    DATA __GOODCA;
     SET __GOODCA; BY __IDCA;
       IF FIRST.__IDCA;
    DATA __DIST3;
     LENGTH __FROM __TO $ 80;
       DO I=1 TO N;
          SET __GOODCA POINT=I NOBS=N;
          __FROM='SC';
          __TO=left(__idca);
          _COST_=0;
          %if &mincont= %then %do;
             _CAPAC_=&NCONTLS;
          %end;
          %else %do;
             _capac_=&mincont;
          %end;
          OUTPUT;
       END;
       %if &mincont^= %then %do;
          __from='SC';
          __to='EXTRA';
          _capac_=&newcont-&mincont*n;
          _cost_=0;
          output;
          do i=1 to n;
             set __goodca point=i;
             __from='EXTRA';
             __to=left(__idca);
             _cost_=0;
             _capac_=&maxcont-&mincont;
             output;
          end;
       %end;
       CALL SYMPUT('NEWCASE',N);
    STOP;
    DATA __DIST;
     SET __DIST1 __DIST2 __DIST3;
%LET DEM=%EVAL(&NEWCASE*&NCONTLS);
PROC NETFLOW
   MAXIT1=&MAXITER
   %if &mincont= %then %do;
      DEMAND=&DEM
   %end;
   %else %do;
      demand=&newcont
   %end;
   SOURCENODE='SC'
   SINKNODE='SK'
   ARCDATA=__DIST
   ARCOUT=__MATCH;
 TAIL __FROM;
 HEAD __TO;
DATA __OUT;
 SET __MATCH;
   IF _FLOW_>0 & __FROM^in ('SC' 'EXTRA') & __TO^='SK';
   __DIJ=_FCOST_;
   %DO I=1 %TO &NVAR;
      __DIF&I=abs(__CA&I-__CO&I);
      __WT&I=&&W&I;
   %END;
PROC SORT; BY __IDCA __DIJ;
DATA &OUT;
      SET __OUT; BY __IDCA;
        drop __from -- _status_;
        IF FIRST.__IDCA THEN __CONT_N=0;
        __CONT_N+1;
      LABEL __IDCA="&IDCA/CASE"
            __IDCO="&IDCO/CONTROL"
            %if &time^= %then %do;
               __catime="&time/CASE"
               __cotime="&time/CONTROL"
            %end;
            __CONT_N='CONTROL/NUMBER'
            __DIJ='DISTANCE/D_IJ'
            %LBLS;
   %END;
%MEND OPTIMAL;
%IF &BAD=0 %THEN %DO;
   %IF %UPCASE(&METHOD)=GREEDY %THEN %DO;
      %GREEDY
   %END;
   %ELSE %DO;
      %OPTIMAL
   %END;
   %IF &BAD2=0 %THEN %DO;
      PROC SORT DATA=&OUT; BY __IDCA __CONT_N;
      proc sort data=__case; by __IDCA;
      data &outnmca; merge __case
           &out(in=__inout where=(__cont_n=1)); by __idca;
           if __inout=0; **non-matches;

      proc sort data=__cont; by __IDCO;
      proc sort data=&out; by __IDCO;
      data &outnmco; merge __cont
           &out(in=__inout); by __idco;
           if __inout=0; **non-matched controls;
      proc sort data=&out; by __IDCA; **re-sort match data set by case id;

      %if %upcase(&print)=Y %then %do;
      PROC PRINT data=&out LABEL SPLIT='/';
       VAR __IDCA __IDCO __CONT_N
       %if &time^= %then %do;
          __catime __cotime
       %end;
       __DIJ %VBLES;
       sum __dij;
      title9'Data listing for matched cases and controls';
      footnote"match macro: case=&case control=&control idca=&idca idco=&idco";
      footnote2"   mvars=&mvars  wts=&wts dmaxk=&dmaxk dmax=&dmax
      ncontls=&ncontls";
      %if &time^= %then %do;
         footnote3"   time=&time  method=&method  seedca=&seedca  seedco=&seedco  ";
      %end;
      %else %do;
         footnote3"   method=&method  seedca=&seedca  seedco=&seedco";
      %end;
      footnote4"   out=&out   outnmca=&outnmca  outnmco=&outnmco";
      run;
      title9'Summary data for matched cases and controls';
      proc means data=&out n mean sum min max; class __cont_n;var __dij
        %if &nvar >=2 %then %do; __dif1-__dif&nvar  __ca1-__ca&nvar
                          %if &time^= %then %do;
                             __catime
                          %end;
                          __co1-__co&nvar
                          %if &time^= %then %do;
                             __cotime
```

```
                        %end;
                        ;
        %end;
        %else %do;
                        __difl __cal
                        %if &time^= %then %do;
                          __catime
                        %end;
                        __col
                        %if &time^= %then %do;
                          __cotime
                        %end;
                        ;
        %end;
      run;
      proc means data=&outnmca n mean sum min max; var &mvars;
      title9'Summary data for NON-matched cases';
      run;
      proc means data=&outnmco n mean sum min max; var &mvars;
      title9'Summary data for NON-matched controls';
      run;
      %end;
    %END;
  %END;
   title9; footnote;
   run;
%MEND MATCH;
```