

Paper PO07

Application of an Exact Covariate Matching Macro in Outcome Analysis

Guiping Yang, Premier, Inc., Charlotte, NC
Stephen Stemkovski, Premier Inc., Charlotte, NC

ABSTRACT

In terms of comparable sample selection and treatment effect evaluation, propensity score matching is not always the most appropriate solution for observational studies. This paper presents a SAS[®] MACRO, %ExactMatch, to assist the investigators in such circumstances. This macro addresses the challenges of matching treated subjects to the most similar controls based on specific observables. The implementations of this macro in healthcare outcome analyses have proved it as an invaluable method of selecting the comparable samples easily and conveniently. %ExactMatch performs most efficiently with small treatment groups and considerably large pools of control subjects. Categorical factors are matched exactly at the specification level; however, continuous measurements are compared within certain calipers predetermined by users. The maximum acceptable distance for individual numerical factors between treatment and control subjects should be chosen carefully. No replacement of controls is allowed, once it is selected, it will not be put back into the pool for further matching. Sample size adjustment is optional; users prejudge whether or not to adjust the ratio of controls to treated samples for matching. All study subjects are involved in weighting, as a result, the variance from matching shrinks significantly. One treatment subject may be matched to one or more control partners as desired. The nearest one is picked up first, the second closest is next and so on. Influence on treatment evaluation from noise of covariates declines significantly and thus selection bias, which is investigated in the last step, balance checking, any unlevelled covariates are reported to user automatically.

KEYWORDS: Exact matching, multiple covariates, shortest distance, weighting, balance, observational study

INTRODUCTION

The casual effect evaluation in clinical trial, healthcare outcome or epidemiology usually is of crucial concern for many researchers. With the presence of experimental data, in theory, both observable and unobservable covariates are expected to be balanced by randomization, therefore, assessment of the treatment effect will not be biased unexpectedly. Unfortunately, this is not largely the case in retrospective, observational studies.

Observational studies, in name itself, mean that all measurements in the study are observable and measurable. Those immeasurable hidden factors or measures where collection is difficult due to cost, time or even ethical issues find no way to be balanced. Based on these limitations inherent in observational studies, the matching algorithm is capable of working only with observables; the hidden bias in the study can not be handled by statistical estimation or feasible assumptions. Therefore, the assumption of independence for observational studies is needed. Alternative wordings for this are 'Unconfoundedness' (Imbens 2004), 'Ignorable Treatment Assignment' (Rosenbaum and Rubin 1983), 'Exogeneity' (Zhao 2006) and 'Conditional Independence Assumption' (Lechner 1999).

Matching is a favored technique for seeking comparable sample and assessing treatment effects in observational studies due to its optimal feature of mimicking randomized experimental design. The main effort of matching is to capture samples with very similar distribution as in the original population, which makes the selected sample more representative in effect evaluation. By picking up those subjects with exact or close values in the covariate matrix, subjects look more 'randomly' assigned to treatment and control groups in the analysis.

In recent years, statistical techniques for matching have attracted widespread attention of researchers and have become an 'official' tool to remove selection bias in non-experimental studies. Of them, propensity scores matching harvests most concern since the initial fundamental work of Cochran in 1965 and continuous efforts from many scientists and experts (Rubin 1974, 1977, Rosenbaum and Rubin 1983, 1984, Rosenbaum 1995, Rubin and Thomas 1996, Ming and Rosenbaum 2000). It replaces many background covariates with a simple function of them to adjust the selection error, which overcomes computation difficulty and practical infeasibility in other strategies. Large overt bias and high dimensionality from many covariates are not a problem at all (Rosenbaum, 1983).

Like two edges of a sword, the limitations and risks in applying propensity score matching still leave some space for covariate matching. When examining the efficiency of covariate matching and propensity score matching, Angrist and Hahn (2004) determined that exact covariate matching performs more efficiently in finite samples than propensity score method. They also concluded that there were implementation limits for propensity score matching, small cell size, low explanatory power of covariates or treatment probabilities close to boundary, namely 0 or 1. Monte Carlo experiment was conducted by Zhao in 2004 for data requirement comparisons between covariate matching estimators and propensity score approaches. Zhao concluded that there is no obvious advantage between these two methods at all. In some stances, covariate matching is preferred. In other studies, propensity score might be more appropriate. Above all, Mahalanobis distance is relatively robust under different settings (Cochan and Rubin 1973, Taguchi and Jugaulum 2002, Rubin 2006, Yang, Stemkowski and Saunders 2007 and Jaramillo 2007), which is the basis for macro %ExactMatch proposed in current paper. Requirement on exact control of some specific covariates can be achieved more efficiently by covariate matching.

Over the years, the literatures described exact covariate matching from various perspectives such as theoretical foundations, application approaches, advantages and disadvantages as well as comparisons with propensity score matching methods. A number of discussions on its evaluation value have confirmed its shoulder-to-shoulder status with propensity score matching, sometimes even overperformed, which can be found directly or indirectly in statistical, economical, pharmaceutical or other application fields. Unfortunately, there is no a practical guideline or easy-to-use software assisting users, especially to those far away from statistical logic, to conduct their studies without difficulty. Here, we fill the gap by means of detailed explanations in several facets.

The paper is structured as follows. The methodological framework of macro %ExactMatch is detailed in next section. After that, the design of the macro and its explanation are detailed. The syntax and parameters of macro are outlined in the fourth part. Prior to the conclusion, a practical example of the macro call demonstrates the macro's characteristics and excellent performance.

ALGORITHM

The basic idea of exact matching of subject i in treatment group to subject j in control group on covariate X was proposed by Zhao in 2004 as,

$$X_i = X_j \Rightarrow f_t(X_i) = f_t(X_j) \quad (1)$$

And

$$d(X_i, X_j) < \varepsilon \Rightarrow d'[f_t(X_i), f_t(X_j)] < \delta \quad (2)$$

Where, $t=0$ or 1 , d and d' are some mathematical functions in density predetermined by users, ε and δ denote the standard error or variance. Equation (1) applies to cases where exact fitting is available; otherwise, approximate matching in equation (2) will be adopted.

We adopt this theoretical logic and extend it to fit our criteria for macro %ExactMatch, expressed in equation (3). Exact comparison is employed for categorical variables, where the same characteristic of the counterpart is required. The second part is descriptive of continuous factors; potential partner in the control group with closest distance to a given treated subject will be chosen. We refer to the first part as real exact matching and the second part as approximate matching. These two parts are packaged into the whole covariate matching algorithm. In %ExactMatch proposed by us, we also modified the second part to Minimum-Maximum-Distance Rule, instead of directly-predefined distances. The measurement of matching is expressed as

$$D = \left\{ \sum_{m=1}^c (X_{mi} = X_{mj}) + \min \left[\sum_{n=1}^k \left(\frac{|Y_{ni} - Y_{nj}|}{w_n} \leq \xi_n \right) \right] \right\} \quad (3)$$

Where, D is the total comparison measurement, The first part $\sum_{m=1}^c (X_{mi} = X_{mj})$ is same as above, exactly matching

on specific levels of m^{th} categorical variables. Within the second term, ξ is the maximum tolerable distance between treatment and control subjects in raw value for the n^{th} continuous factor. Totally, there are C categorical variables and

k numerical variables involved in matching. Element w_n plays a weighing role before Minimum-Maximum-Distance is finally inspected, compared and concluded for best suitable pair, then the second best and so on in descending order.

Basically, Minimum-Maximum-Distance algorithm is a two-step procedure for bias reduction. These two stages were named as Approximate Filtering and Precise Measuring.

(1). Approximate Filtering.

This step is characterized as a big filter; the size of hole is the acceptable maximum distance between a given treatment subject and all control subjects, regardless of the distance direction, only absolute value matters. In equation (3), $|Y_{ni} - Y_{nj}| \leq \xi_n$ carries out this function on continuous variables. For each continuous measurement, all control subjects with distance to a certain treatment subject equal to or less than radius ξ on this specific variable go through for further examination. ξ need be predefined by users based on practical experience, logic feasibility or empirical suggestions. All continuous variables have their own thresholds.

On the other hand, one or more categorical variables are specified for filtering. Only those controls with exactly same categorical variable level as treatment subjects are released ($X_{mi} = X_{mj}$). This part is really 'exact matching'. All categorical variables and continuous variables combined act as a powerful tool in filtering process, nearly 50-85% of controls should be held as the candidates for further examination.

(2). Precise Measuring.

Through filtering in the first step, some potential candidates in control group will undergo further scrutiny. These control subjects are involved in the weighted summary of distances on all continuous variables. Weighting factor fortifies the power of gauge significantly. Here, categorical variables will not be used any more, as they are already exactly compared in step one. However, continuous variables keep on catching more competent controls. By picking up the nearest one by $\min[\sum_{n=1}^k (\frac{|Y_{ni} - Y_{nj}| \leq \xi_n}{w_n})]$, the best match is identified. On demand of more matches in control, control subjects with further and further distances from treatment subject may be included at the expense of matching quality due to the increasing variance.

These two steps work sequentially and repeat for each treatment subject. The formula of Minimum-Maximum-Distance reflects 'Two filters and one catch' structure indeed.

This core part of algorithm employed in macro %ExactMatch does not leave the reasonable assumptions behind. As described in the above discussion, we assume the independence between treated units and control units, between treatment selection and treatment effect and among individual units. Background noise from balancing covariates should have pronounced effects on both treatment choice and effect evaluation; and the hidden unobservable or immeasurable elements should not exist, or have equivalent or negligible effects on both treatment participation and evaluation.

Besides matching functions, %ExactMatch provides some statistical tests to validate balancing effects. These tests will give users assurance about matching quality for both categorical and continuous variables. Needless to say, sensitivity analysis would be another added benefit of %ExactMatch if used.

DESIGN COMPONENTS

Lock log and listing window

Due to large amount of calculations involved, listing and log information is suppressed during matching, instead, those messages useful for debug or related to final results are listed after the window is unlocked. This feature frees users from clearing output and log windows repeatedly but still keep important messages.

Variable resolution

Comparison factors are quoted by single quotation mark on purpose for differentiating them from other macro parameters. They must be dequoted by a small macro before matching. The resolution of these parameters also serves

for logic checking. Both the wrong number of parameters and missing of these items will be investigated and updated to users.

Missing values imputation

Generally speaking, missed values of comparison variables will lead to poorer matched control subjects to a treatment subject and even threat to comparison feasibility. Missing values are treated as a specific value; therefore, they are considered being matched only if both treatment and control subjects have missing values of the same covariates. In %ExactMatch, records with missed values were checked in advance. For a large sample, removing such missing records will have minimal influence in overall distribution given that they only account for a small percentage of the sample and missing mechanism is not related to treatment evaluation. In this case, for simplicity, those records with missing values will be thrown away. Nevertheless, if such missing values occupy a large proportion, for example, over 30%, or missing information is of critical importance to overall treatment effect assessment, a simple imputation approach, which entails replacing missing categorical variable with the level of highest frequency and continuous variable with sample mean, is introduced to insure sample data integrity. In this way, sample distribution will be close to its population distribution.

Matching time

Matching time of %ExactMatch is closely related to sample size in control group. If matching time is a concern, some strategies are recommended to reduce sample size of the control group, which is key determinant of computing speed. The threshold values determine the scope for first filtering, a lower threshold might be a good choice for more precise selection on highly qualified control subjects, so that, in the subsequent computation step, less time will be spent on sorting, comparing and measuring distance. However, this method is at the expense of less available controls and hence rising variance. If specifications of some measuring factor are allowed to be redefined for more rapid running, the running time will be shortened significantly. For some categorical factors, if classification is relatively granular, to replace them with some coarse classification will definitely reduce computing time. The number of comparison variables, particularly continuous variables, is proportional to the computing and comparing time to a certain extent. As part of conditions for first filtering, more categorical factors means the shrinkage of prequalified candidates for further filtering, and vice versa.

Sample size adjustment

Sample size adjustment is optional in this macro. If users prefer to no adjustment to sample size, all treatment and control subjects are involved in matching. Once the user decides to adopt adjustment to shorten matching time and at same time, is sure that cutting down sample will not cause inexact or incomplete matching and damage bias minimization, certain positive integer should be predefined to serve as ratio of control to treatment groups for matching after sample adjustment.

By examination of sample size effect on matching result in terms of running time and balanced covariates between treatment group and control group, some cutting point for sample adjustment can be predetermined before matching when sample size, especially control group, is relatively large. In this way, control sample size reduction will desirably decrease matching time but not at sacrifice of matching quality. When the treatment subjects are over 1,000 and the ratio of control subject number to treatment subject number is greater than user specified ratio, only a proportion of control group subjects will be randomly selected by %ExactMatch for matching.

Matching numbers

In practice, usually it needs to compare treatment and control groups with one-to-one algorithm according to one to multiple risk factors. However, in some cases, in order to reduce variance from control selection, more than two control subjects might be matched to one treatment subject based on matching numbers specified by users or numbers of available controls in dataset. The most optimal case is a full matching, where all possible combinations of treatment subjects and control subjects are compared, one treatment subject may be matched to one or more control partners, and visa versa. In terms of some practical concerns in pharmaceutical and outcome researches, macro presented here exploits one-to-one and extended one-to-many algorithms.

All subjects in treatment are considered and examined one by one. In case of paired matching, the matched pairs are flagged in final result for future possible identification in analysis. If two or more controls are required to match each individual treatment subject, flags are labeled in an intermediate SAS table named MATCH, which is temporally saved in work library and will be deleted in next run of matching.

Logic checking

Both data files and user provided parameters will be validated in macro whenever it is necessary to guarantee the appropriate environment for macro's running. The error message will be provided if parameter miss-specification, logic error or data integrity error are detected.

Weighting and variance suppression

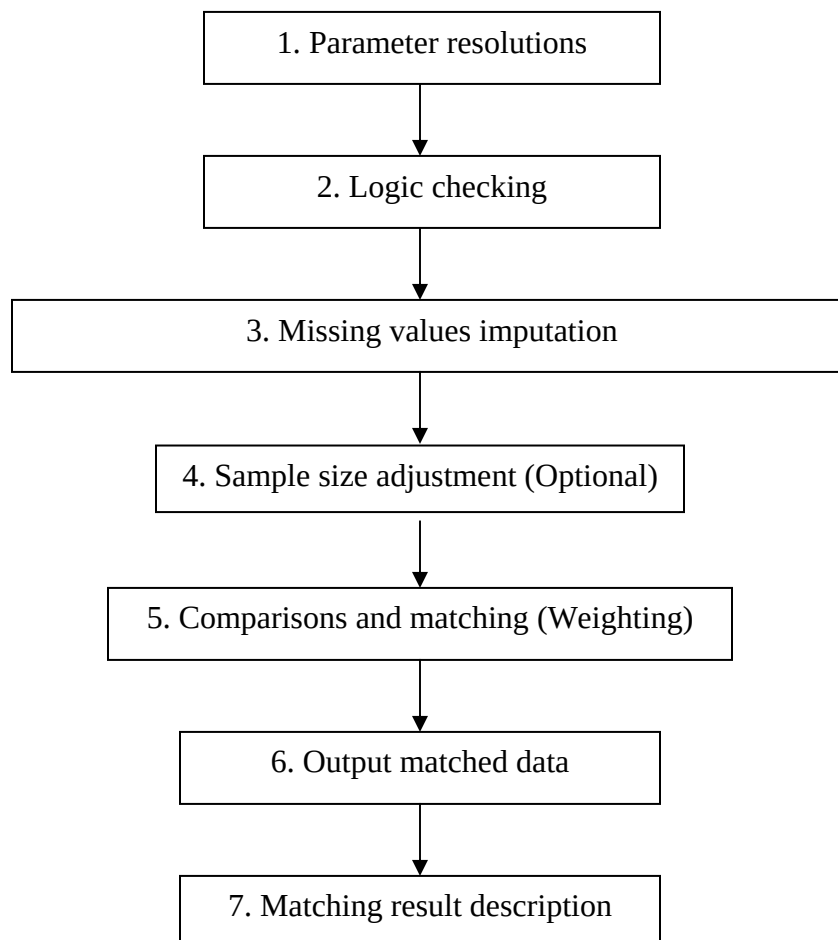
To reduce variance during matching, many channels are accessible. Of them, weighting is a commonly used approach and adopted in %ExactMatch. Weights of individual numerical comparison variables are the reverses of its corresponding standard deviation. Because all subjects in input data are involved, variance will greatly drop down. Further, variance reduction can be realized by upgrading one-to-one matching to one-to-many matching. More stringent matching factors and lower matching threshold is conducive to better matched samples and less running time, but often at expense of rising variance due to the broken distribution assumption. A balance between selection bias reduction and matching variance reduction must be stricken.

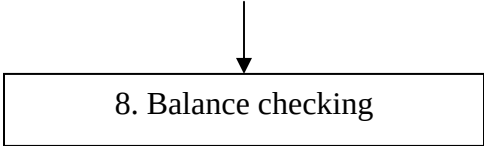
Balancing Check

Categorical covariates are exactly matched. Therefore, only balance test on continuous factors was conducted in use of exact paired t-test. The default test significance level is 0.05, which is commonly used in healthcare studies and epidemiology. Tests are conducted immediately after the matching step and the test results are outputted to the same location as that of input and output data files. As long as the significant difference between treated and control groups are found for any comparison covariates, a message will be written to the log window. Therefore, the user may make a decision in time on how to deal with it, adjusting matching measurements or including this factor as an influential element into subsequent regression analysis. The ideal consequence is that all factors concerned tend to have a very similar distribution between two groups, thus, their effects on treatment evaluation are as close to zero as possible or are equivalent, which is the wanted target of covariate matching.

SYNTAX AND PARAMETERS

%ExactMatch does exact covariate matching based on specified parameters in following steps.





8. Balance checking

Before working into any example illustration, let us detail the macro syntax first, including the meanings and usages of the macro parameters.

The syntax of macro is:

%ExactMatch (Path, ID, DataIn, DataOut, Trt, classVar, numVar, diffVar, Match_N, Adjust);

%ExactMatch contains ten parameters, relating to both data information and comparison requirements. Measurement factors are user-provided classical, numerical variables and corresponsive caliper values.

Path:	The location saving data before matching and after matching as well as balance checking results. Libref is not appropriate here;
ID:	To identify the unique subjects in both treatment and control groups, for instance, Patient_id;
DataIn:	The input SAS dataset ready for matching, saved in the above location. Subject ID, treatment factor and all measurements must be included;
DataOut:	The out SAS dataset saving matched result with same structure and location as those of input data;
Trt	The binary treatment factor with value 1 for treatment group and 0 for control group;
classVar:	A list of categorical variables for comparisons, quoted in single quotation mark and separated by blank;
numVar:	A list of numerical variables for comparisons, quoted in single quotation mark and separated by blank;
diffVar:	A list of numbers for corresponsive numerical variables, indicating maximal acceptable distances between treatment and control subjects, predefined by user, quoted in single quotation mark and separated by blank;
Match_N:	Number of control subjects to each treatment subject after matching, prespecified by user. Its value is positive integer.
Adjust:	The determination of sample size adjustment, the value is 'No' if no adjustment is needed, or any positive integers, for instance, 100, expressing the proportion of control sample to treatment sample for matching after adjustment procedure. No quote is required. This integer value is distinct from matching number above.

It is noted that all parameters must be specified, even some are not suitable to specify, use empty space to locate. Variables and threshold values used in examination need be quoted by quotation mark. This macro is appropriate for binary treatment or alike, the generalization is likely to be exploited. Due to the application of some newest functions, SAS version 9 or above is required.

EXAMPLE

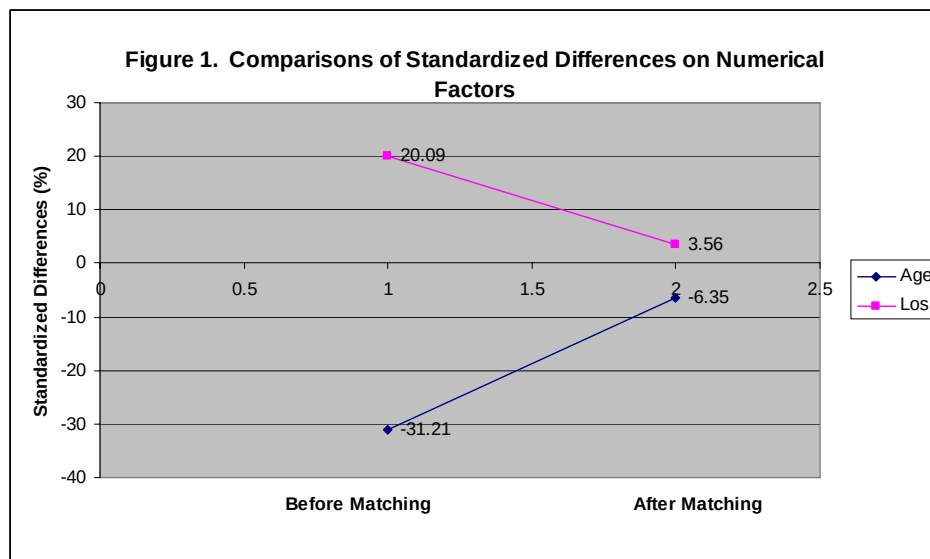
This section demonstrates how %ExactMatch was called for task. The illustration was based on a sample pulled from a big warehouse of Premier healthcare Informatics. The total subjects in study are 122,325, of which, 1,557 patients fall in treatment group, where patients obtained the treatment of medicine 'Dotrecogin alfa (activated)'. Exact covariate matching performed on factors including categorical variables, hospital, gender, race, discharge status, bed-size as well as numerical variables, patient age and length of stay (LOS). Thresholds for age and length of stay were frozen as 15 years and 10 days respectively. No sample adjustment is requested. After matching, totally 1,271 treated subjects

found the counterparts in control by one-to-one algorithm, matching rate is about 81.6%. The descriptive statistics and balance tests on these covariates are exhibited in tables.

As expected, all categorical factors become balanced on the button with p value at 1.0000. Both patient age and length of stay, in Table 1, are much over the critical point 0.05 ($P=0.1093$ and $P=0.3698$). However, before matching, both display the significant differences ($P<0.0001$) between two groups. The plots of standardized difference offer us more visual evidence for this excellent matching fruit by displaying the apparent dissimilarity before matching and great closeness after matching. 0% of absolute standardized difference means the exact same distribution, but, 10% of absolute standardized difference is the critical point of questionable unlikeness. Figure 1 indicates the overall standardized difference for the whole sample. The calculations for overall standardized difference on whole sample and standardized difference distribution for individual subjects were borrowed from Love (2003). Before matching, both patient age and hospital stay have absolute values pretty far from 10%, 20% and -31.21% respectively, which indicates the obvious unbalance between treatment and control groups. The rapid drop to zero in absolute values after matching, 3.56% for patient age and -6.35% for hospitalization days, tells us that a very satisfying balance was achieved. In addition, the distributions of standardized differences in Figure 2 exhibits the severe skewness and more scattered on unbalanced factors and relatively much symmetric shape for balanced factors, where most points piled up around zero.

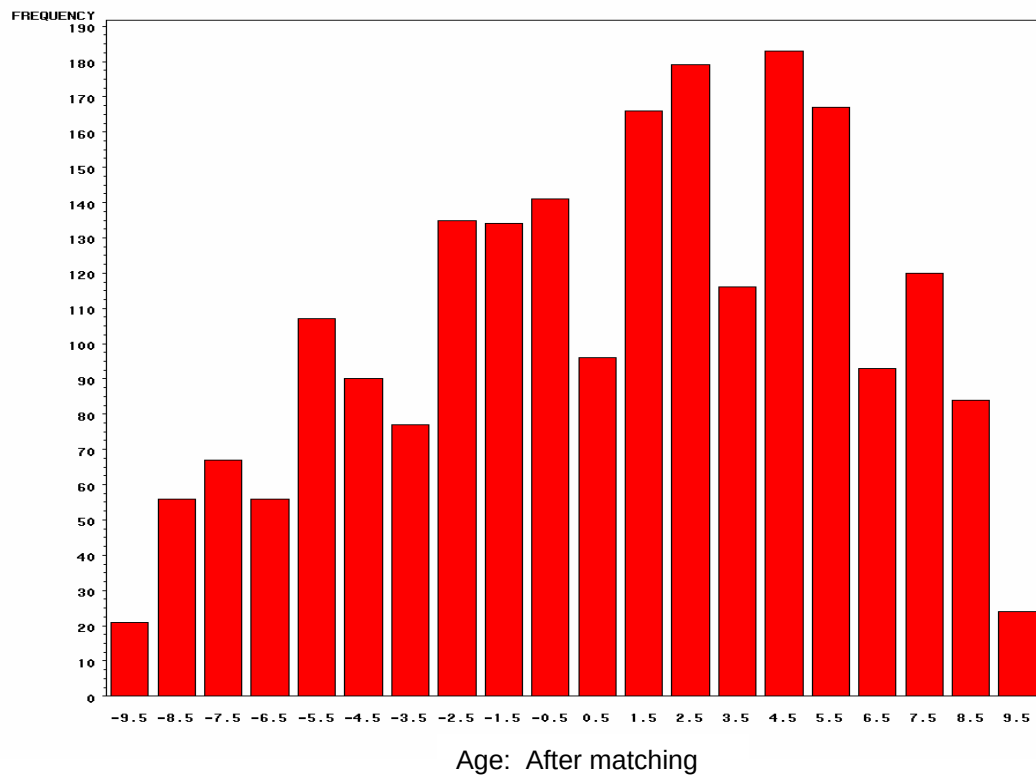
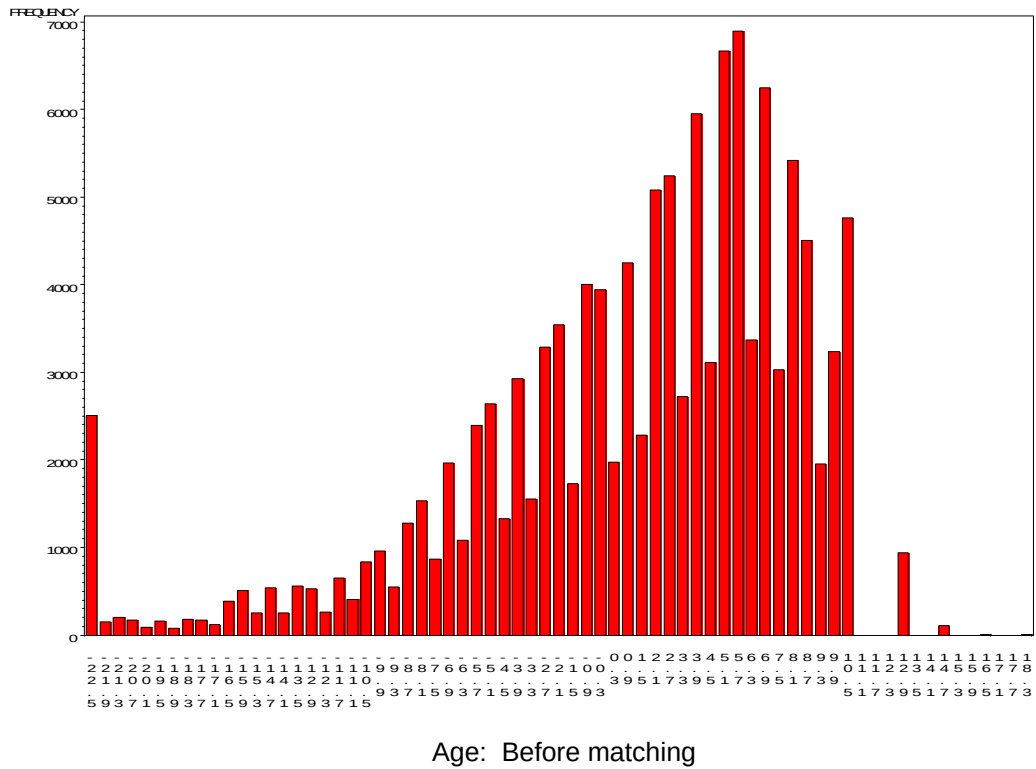
Table 1. Balance Checking on Continuous Measurements by Mean and Standard Deviation

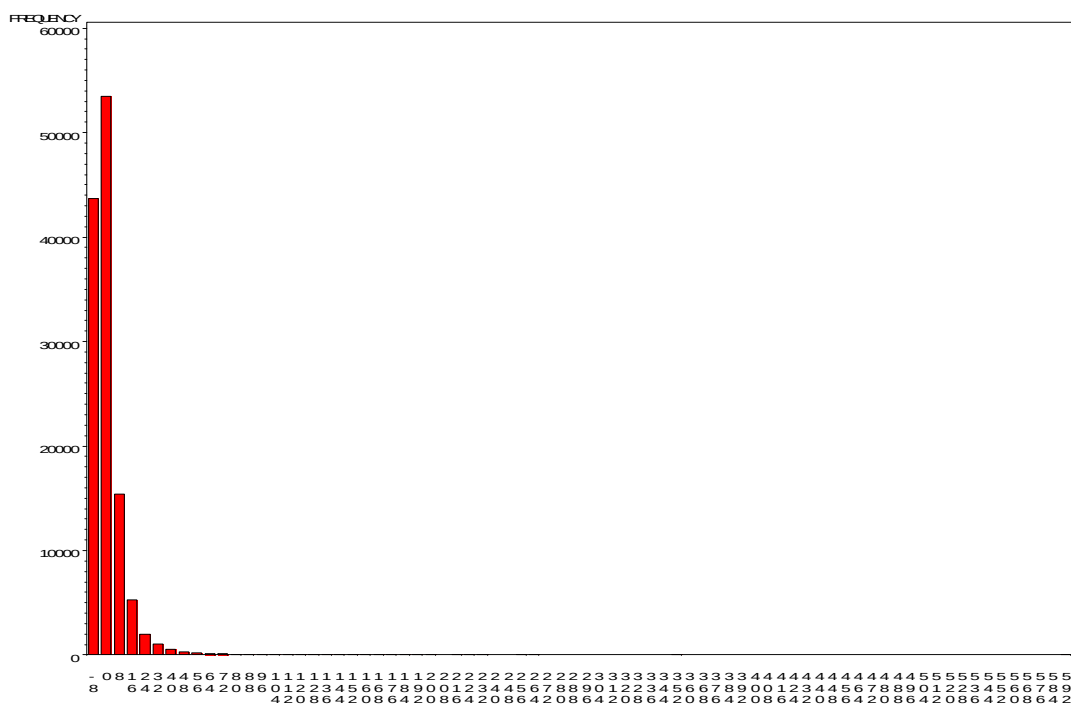
Comparison	Factor	Control	Treatment	T value	P value	Significant Difference
Before Matching	Age (Years)	64.3 (19.6)	58.5 (17.5)	12.67	<0.0001	Yes
	Length of Stay (Days)	16.8 (18.8)	20.7 (19.9)	-7.67	<0.0001	Yes
After Matching	Age (Years)	61.5 (16.2)	60.5 (16.6)	1.60	0.1093	No
	Length of Stay (Days)	16.1 (13.4)	16.6 (13.7)	-0.90	0.3698	No



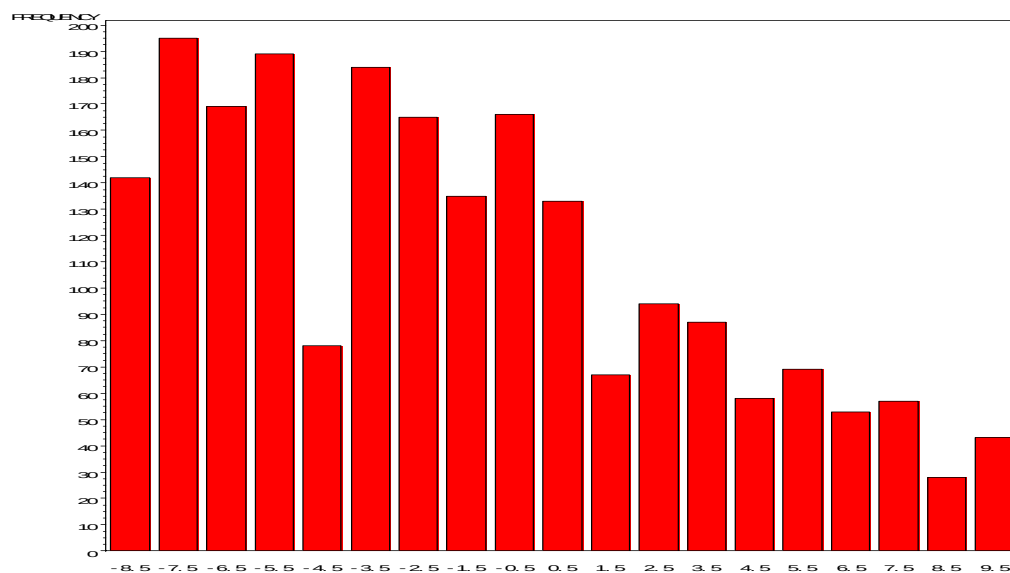
*Los=Length of stay(days).

Figure 2. Distribution of Standardized Differences for Numerical Factors





Length of Stay: Before matching



Length of Stay: Before matching

FINAL COMMENTS

This macro was developed and tested in SAS 9.1.2., on which some new functions are supported. The algorithm and results were checked thoroughly by varying sample size, covariate selection or missing parameters. However, no guarantees of the correctness of macros are provided in all sorts of environments. The necessary modifications for specific cases are encouraged if needed.

CONCLUSION

%ExactMatch has been proven as a convenient way to choose comparable sample in observational studies where propensity score is not applicable to remove selection bias. Minimum-Maximum-Distance algorithm catches the closest pairs efficiently. If sample size adjustment is adopted, by randomly selecting samples in control group to keep the optimal matching speed and preciseness, it relieves heavy burden on computation and shortens the extra running time without any drop in matching efficiency and accuracy. The maximum distance of numerical factor between treatment and control groups is adjustable with respect of study goal, sample requirement and empirical knowledge from statisticians or researchers. A weighting methodology cuts down variance by means of involutions of all available study subjects. No replacement is in need in this macro. Balance checking for measured covariates is automatically conducted following matching. In one word, %ExactMatch offers a straightforward and easy-to-use tool for any levels of users.

REFERENCES:

- Angrist, J.D. and Hahn, J. (2004) When to control for covariates? Panel asymptotic results for estimates of treatment effects, *The Review of Economics and Statistics*, 86, 58-72.
- Cochan, W.G. (1965) The planning of observational studies of human populations (with discussion), *Journal of the Royal Statistical Society Series A*, 128, 234-230.
- Cochan, W.G. and Rubbin D.B. (1973) Controlling bias in observational studies: a review. *The Indian journal of Statistics, Sankya, Series A*.
- Imbens, G.W. (2004) Nonparametric estimation of average treatment effects under exogeneity: A review, *Review of Economic and Statistics*, 86, 4-29.
- Jaramillo, A. (2007) Promotion response analysis in pharmaceutical industry –lessons to be learned from clinical trials methodologies. *PharmaSUG Conference*, Denver, CO, June 3-6.
- Lechner, M. (1999) Earnings and employment effects of continuous off-the-job training in East Germany after unification, *Journal of Business and Economic Statistics*, 17, 74-90.
- Love, T.E. (2003) Propensity scores: what do they do, how should I use them and why should I care? *ASA Cleveland Chapter*, December. <http://www.chrp.org/love/ASACleveland2003Propensity.pdf>
- Ming, K. and Rosenbaum, P.R. (2000) Substantial gains in bias reduction from matching with a variable number of controls, *Biometrics*, 56, 118-124.
- Rosenbaum, P.R. and Rubin, D.B. (1983) The central role of the propensity score in observational studies for causal effects, *Biometrika*, 70, 41-55.
- Rosenbaum, P.R. and Rubin, D.B. (1984) Reducing bias in observational studies using subclassification on the propensity score, *Journal of the American Statistical Association*, 79, 516-524.
- Rosenbaum, P.R. (1995) *Observational studies*, Springer Series in Statistics, New York.
- Rubin, D.B. (1974) Estimating causal effects of treatments in randomized and nonrandomized studies, *Journal of Educational Psychology*, 66, 688-701.
- Rubin, D.B. (1977) Assignment to treatment group on the basis of a covariate, *Journal of Educational Statistics*, 2, 1-26.

- Rubin, D.B. and Thomas, N. (1996) Matching using estimated propensity scores: relating theory to practice, *Biometrics*, 52, 249-264.
- Rubin D.B. (2006). *Matched sampling for causal effects*, p170, Cambridge University Press.
- Taguchi G. and Jugulum R. (2002) *The Mahalanobis-Taguchi Strategy: A Pattern Technology System*, New York, NY: Wiley.
- Yang G., Stemkowski and Saunder W. (2007) A review of propensity score application in healthcare outcome and epidemiology. *PharmaSUG Conference*, Denver, CO, June 3-6.
- Zhao, Z. (2004) Using matching to estimate treatment effects: Data requirements, matching metrics and Monte Carlo evidence, *The Review of Economics and Statistics*, 86, 91-107.

ACKNOWLEDGMENTS

The author would like to thank Rechun He for his encouragement and reviewing this manuscript with the valuable comments.

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Guiping Yang
Premier Inc.
2320 Cascade Pointe Blvd. Suite 100
Charlotte, NC 28208
Work Phone: (704) 733-5668
Fax: (704)357-3383
E- mail: Guiping_yang@Premierinc.com
Web: <http://www.premierinc.com>

APPENDIX:

```
%macro ExactMatch(Path,Id,DataIn,DataOut,Trt,classVar,numVar,diffVar,Match_N,Adjust);

options formdlm='-' mlogic mprint symbolgen nodate nonumber;

libname lib "&path.";

/*Check if the input data is provided*/
%if %sysfunc(fileexist(&path.\&datain..sas7bdat))=0 %then %do;
    %put Error: No input data exists !;
    %goto exit;
%end;

/*Record the time to start matching*/
%let start=left(put(datetime(),dateampm22.));
data dd; Hello="This matching starts running at "&start.; run;
proc print noobs;run;

/*Close output window*/
ods listing close;

/*Close log window*/
filename junk dummy;
proc printto log=junk; run;

/*****
/*****factor resolutions*****/
/*****/

/*Dequote parameters with quotation marks*/
%macro DeQuote (classVar);
    %LET length = %length(&classVar);
    %IF %index(%STR(%"),%qsubstr(&classVar,1,1)) and &length>1 and
        %qsubstr(&classVar,&length,1)=%qsubstr(&classVar,1,1) %then
        %substr(&classVar,2,%eval(&Length-2));
    %ELSE &classVar;
%mend;

/*Identify class or numerical variables to compare*/
%macro cal(var);

    %global &var.N ;

    /*count variable number to compare*/
    data _NULL_;
        &var.N=countc(strip(compbl(&&&var.Var.)),")+(lengthn(&&&var.Var.)>0);
        call symput("&var.N",&var.N);
    run;

    %put &&&var.N.;

    /*Assign individual variable into macro variable*/
    data _NULL_;
        %do n=1 %to &&&var.N.;
            %global &var.&n.;
            %let &var.&n.=%scan(%str(%dequote(&&&var.Var.)), &n.);
        %end;

end;
```

```

run;

/*Show values of assigned macro variables*/
data _NULL_;
    %do n=1 %to &&var.N.;
        %put &&var.&n.;
    %end;
run;

%mend;
%cal(class);
%cal(num);
%cal(diff);

/*calculate standard deviation for each numerical variables as weighting parameters*/
%do n=1 %to &numN.;
    %global std&n.;
    proc sql; select std(&&num&n.) into :std&n. from lib.&datain.; %put &&std&n.;
%end;

%let lastN=&&num%trim(%left(&numN.)); %put &lastN.;
%let lastC=&&class%trim(%left(&classN.)); %put &lastC.;
%let lastD=&&diff%trim(%left(&diffN.)); %put &lastD.;
%let lastS=&&std%trim(%left(&numN.)); %put &lastS.;

/*****
/*****Logic check*****/
/*****

/*Check if no comparison factor is defined*/
%if &classN.=0 and &numN.=0 %then %do;
    %put Error: No comparison requirements were defined for matching !;
    %goto exit;
%end;

/*Check if the corresponsive limitation values are offered for continuous varilables*/
%if &numN.^=&diffN. %then %do;
    %put Error: Number of continuous varilable does not matching number of limitation points !;
    %goto exit;
%end;

/*Delete postmatched table in work library*/
proc datasets library=work nolist; delete match; run;

/*****
/*****Missing imputations in all comparison variables*****/
/*****

proc sql;
    select count(*) into :total_obs from lib.&datain.;

    %if &classN.>0 %then %do;
        %do n1=1 %to &classN.;
            %global missC&n1.;
            select count(*) into :missC&n1. from lib.&datain. where &&class&n1.=";
        %end;
    %end;
    %if &numN.>0 %then %do;
        %do n2=1 %to &numN.;
            %global missN&n2.;

```

```

select count(*) into :missN&n2. from lib.&datain. where &&num&n2.=.;
%end;
%end;

%let lastM=&&missN%trim(%left(&numN.)); %put &lastM.;

data _Null_;
    miss_ratio=sum(%do n1=1 %to &classN.; &&missC&n1.,%end; %do n2=1 %to &numN.-1;
        &&missN&n2.,%end; &lastM.)/&total_obs.;
    call symput ('Miss_ratio',miss_ratio);
run;

%put &miss_ratio.;

%if &miss_ratio.>0.3 %then %do;
    /*Categorical variables: Highest ratio method*/
    %do n=1 %to &classN.;
        proc sql;
            create table &&class&n. as
            select  a.&&class&n.,a.cnt/(a.cnt+b.cnt) as ratio
            from    (select &&class&n.,count(*) as cnt
                    from    lib.&datain.
                    where   &trt.=1
                    and     &&class&n. is not null
                    group by 1) a,
                    (select &&class&n.,count(*) as cnt
                    from    lib.&datain.
                    where   &trt.=0
                    and     &&class&n. is not null
                    group by 1) b
            where   a.&&class&n.=b.&&class&n.;

            select &&class&n. into :max from &&class&n. having ratio=max(ratio);

            data lib.&datain.;
                set lib.&datain.;
                if &&class&n.=" then &&class&n.="&max.";
            run;

        %end;

    /*Continuous variables: Average value method*/
    %do n=1 %to &numN.;
        proc sql;
            select avg(&&num&n.) into :mn&n. from lib.&datain.
            where &&num&n. is not missing;

            data lib.&datain.;
                set lib.&datain.;
                if &&num&N.=. then &&num&N.=&mn.;
            run;

        %end;
    %end;

%else %if &miss_ratio.>0 %then %do;
    %do n=1 %to &classN.;
        data lib.&datain.; set lib.&datain.; where &&class&n.^=" ; run;
    %end;

    %do n=1 %to &numN.;
        data lib.&datain.; set lib.&datain.; where &&num&n.^=" ; run;
    %end;

```

```

%end;

/*Devide records into control and case groups*/
data case ctrl;
  set lib.&datain.
    (keep=&id. &trt.
      %if &classN.>0 %then %do;
        %do v1=1 %to &classN.; &&class&v1. %end;
      %end;
      %if &numN.>0 %then %do;
        %do v2=1 %to &numN.; &&num&v2. %end;
      %end;
    );
  if &trt.=1 then output case; else output ctrl;
run;

/*****Sample size adjustment*****/
%global case_n ctrl_n ideal_ctrl left_ctrl;
proc sql; select count(distinct &id.) into :case_n from case ;
proc sql; select count(distinct &id.) into :ctrl_n from ctrl ;

%if %upcase(&Adjust.)=NO %then %do;

  %put Sample size will not be adjusted per request.;

%end;

%else %do;

  %put sample size will be adjusted to ratio of case to control 1 : &Adjust.;

  %let ideal_ctrl=%eval(&case_n.*&adjust.);
  %let left_ctrl=%eval(&ctrl_n.-&ideal_ctrl.);
  %put &case_n. &ctrl_n. &ideal_ctrl. &left_ctrl.;

  %if &case_n.>1000 and &ctrl_n.>&ideal_ctrl. %then %do;

    data x(keep=remove);
      do i=1 to &left_ctrl.*2;
        remove=round(ranuni(123)*&ctrl_n.);
        output;
      end;
    run;

    proc sort data=x nodupkey;by remove; where remove>0;run;

    data x; set x; if _N_<=&left_ctrl.;run;

    proc sql;select remove into :remove separated by ',' from x;

    data ctrl; set ctrl; if _N_ in (&remove.) then delete; run;

  %end;

%end;

proc sort data=case; by &id.;run;

/*****Exact Matching*****/

```

```

/*****
/*Match records in case group with all available records in control group based on given
comparison requirements, selecting the best match with minimum distance and output to a
seperate sas file, then, delete the matched case and controls from pool. This procedure
proceeds until all records in case group use up. The prematched information will be carried
out to postmatched sample.*/

%do i=1 %to &case_n.;

    data case&i.; set case; if _N_=&i.;run;

    proc sql;
        create table match&i. as
        select distinct
            a.&id. as case_&id.,
            b.&id. as ctrl_&id.,
            %if &numN.>0 %then %do;
                sum( %do k=1 %to &numN.-1;
                    abs(a.&&num&k.-b.&&num&k.)/&&std&k.,
                    %end;
                    abs(a.&lastN.-b.&lastN.)/&lastS.)
            %end;
            %else 0 as diff
        from case&i. a,
             ctrl b
        where
            /*Classical variable comparison conditions*/

            /*&classN.>0 and &numN=0*/
            %if &classN.>0 and &numN.=0 %then %do;
                %do j=1 %to &classN.-1;
                    a.&&class&j.=b.&&class&j. and
                %end;
                a.&lastC.=b.&lastC.
            %end;

            /*&classN.>0 and &numN>0*/
            %else %if &classN.>0 and &numN.>0 %then %do;
                %do j=1 %to &classN.;
                    a.&&class&j.=b.&&class&j. and
                %end;
            %end;

            /*Numerical variable comparison conditions*/
            %if &numN.>0 %then %do;
                %do m=1 %to &numN.-1;
                    abs(a.&&num&m.-b.&&num&m.)<=&&diff&m. and
                %end;
                abs(a.&lastN.-b.&lastN.)<=&lastD.
            %end;

        order by 1,3;
    quit;

    proc sort data=match&i.; by case_&id. diff;run;

    data match&i.; set match&i.; by case_&id. diff; if _N_<=&Match_N.;run;

    proc sql; select count( *) into :obs from match&i.;

    %if &obs.>0 %then %do;

```

```

proc append data=match&i. base=match ;run;

%end;

proc sql; delete from ctrl where &id. in (select distinct ctrl_&id. from match&i.);

%end;

/*Label matched number for each pair of case and ctrl subjects*/
data match; set match; pair_flag=_N_;run;

/*Retrieve information in original input data for those matched pairs*/
proc sql;

create table lib.&dataout. As

%if &match_N.=1 %then %do;
select *
from (select distinct a.*,b.pair_flag
from lib.&datain. a,
(select distinct case_&id. as &id., pair_flag from match) b
where a.&id.=b.&id.)

union
select *
from (select distinct a.*,c.pair_flag
from lib.&datain. a,
(select distinct ctrl_&id. as &id., pair_flag from match) c
where a.&id.=c.&id.);

%end;

%if &match_N.>=2 %then %do;
select *
from (select distinct a.*
from lib.&datain. a,
(select distinct case_&id. as &id. from match) b
where a.&id.=b.&id.)

union
select *
from (select distinct a.*
from lib.&datain. a,
(select distinct ctrl_&id. as &id. from match) c
where a.&id.=c.&id.);

%end;

%global final_obs;
select count(*) into :final_obs from match; %put &final_obs.;

quit;

/*Reopen log window*/
proc printto; run;

/*Reopen output window*/
ods listing;

/*Recording matching ending time*/
%let end=left(put(datetime(),dateampm22.));
data dd; Hello="This matching ends running at "&end. ;run;
proc print noobs;run;

/*****
/*****Check matching rate and distribution*****/
/*****/

```

```

title 'The frequency of original treatment and control subjects are:';
proc freq data=lib.&datain.; tables &trt.;run;

%if %sysfunc(fileexist(&path.\&dataout..sas7bdat)) and &final_obs.>0 %then %do;

%put Data set &path.\&dataout. already exists.;

title 'The frequency of matched treatment and control subjects are:';
proc freq data=lib.&dataout.; tables &trt.;run;
title;

proc sql noprint; select count(distinct &id.) into :matched from lib.&dataout. where &trt.=1;

title "Original subjects in treatment group are &case_n.";
title2 "After matching, Matched subjects are &matched.";
title3 "Overall, matching rate is %sysevalf(&matched./&case_n.*100) %.";
data y;xxx='Matching is done successfully !';run;
proc print noobs;run;

/*****
/*****Test of balancing factors*****/
/*****/

proc datasets library=lib nolist; delete class_statistics class_Pvalue;run;

ods listing close;

/*Tests on continuous variables*/
proc datasets library=lib nolist; delete num_statistics num_Pvalue;run;

proc ttest data=lib.&dataout.;
var %do g=1 %to &numN.; &&num&g. %end;;
class &trt.;
ods output Statistics=lib.num_statistics(keep=variable class N mean stddev
                                         where=(class^='Diff (1-2)'))
            ttests=lib.num_pvalue(keep=variable tvalue probt
                                   rename=(tvalue=value probt=probt));
run;

ods output close;

proc transpose data=lib.num_statistics
out=lib.num_statistics( drop=_LABEL_ rename=(_NAME_=Statistics variable=factor))
prefix=&trt.;
by variable;
id class;
run;

proc sort data=lib.num_pvalue nodupkey;by variable;run;

data lib.num_Pvalue;
length Diff $3;
set lib.num_Pvalue;
if prob<0.05 then Diff='Yes';else Diff='No';
run;

ods listing;

title;footnote;

```

```

/*Output the continuous variables not balanced*/
proc sql;
    %global num_no Num_test;

    select count(distinct Variable) into :num_test from lib.num_pvalue where diff='Yes';

    %if &Num_test.>=1 %then %do;
        select distinct variable into :Num_No separated by ','
        from lib.num_Pvalue
        where diff='Yes';

        %put Sorry, numerical factors: "&num_no." are not balanced !;
    %end;
    %else %do;
        data x; Congratulations='All numerical factors are balanced.';run;

        proc print noobs;run;
    %end;

    data y;xxx='Balance checking is done successfully !';run;
    proc print noobs; run;

%end;

%else %do;
    data y;xxx='Matching failed, no pair was selected, sorry !';run;
    proc print noobs;run;
%end;

title; footnote;

proc datasets library=work kill;run;

%exit:

%mend;

/*****Program is over here*****/

```