PharmaSUG 2014 - Paper SP07

A SAS® Macro to Evaluate Balance after Propensity Score Matching

Erin Hulbert, Optum Life Sciences, Eden Prairie, MN
Lee Brekke, Optum Life Sciences, Eden Prairie, MN

ABSTRACT

Propensity score matching is a method used to reduce bias in observational studies by creating two populations that are similar (i.e., balanced) across a number of covariates using a match on only a single scalar, the propensity score. The matched samples are then treated as a quasi-experimental population, allowing for simplified analysis of study outcomes. Whenever a propensity match is performed, the balance between the two samples should be evaluated. Balance checking may be used to compare matches from multiple iterations of the propensity score model or from different matching algorithms and to provide information for any trade-offs between the closeness of the match and final sample size. Additionally, imbalances in the final matched sample should be kept in mind and possibly adjusted for when analyzing study outcomes.

The published literature encourages using a variety of methods to evaluate balance. We have developed a SAS macro that will run a series of tests on the pre- and post-matched samples, following published guidelines. This macro evaluates balance independent of the methods used to create the propensity score or perform matching. Macro output includes information about the pre- and post-matched sample sizes, distribution of propensity scores, and the results of tests to compare covariates of interest, including calculating standard differences. The output is in RTF format so that it can easily reviewed by non-programmers. This SAS macro to evaluate balance with easy-to-read output allows for thorough testing of balance and contributes to a better final matched sample.

INTRODUCTION

Propensity score matching is a statistical method used in observational studies that matches cases and controls on many covariates using one scalar, the propensity score (Rosenbaum and Rubin, 1983; D'Agostino, 1998). This mimics a randomized control trial, creating a sample with similar (i.e., balanced) distributions of the covariates in both the case and control subgroups as well as similar distributions of probability to receive treatment. This quasi-experimental design reduces bias and allows for more direct and simplified analysis of outcomes. Typically, a scalar propensity score is created from a logit or probit modal as the predicted probability that a subject received treatment given a set of baseline characteristics. One or more controls are matched to each case on this propensity score as closely as possible while maintaining an adequate sample size.

Evaluating the balance after matching provides information about how well the propensity score and matching algorithms created case and control populations with similar baseline covariate distributions. Both the propensity score model and the matching parameters can be run, evaluated, tweaked and run again until an acceptable match is created (Ho, 2007; Austin, 2008). An acceptable match is study-dependent and is often the result of a trade-off between the closeness of the match and maintaining sample size. Testing the balance after each change to the propensity score or matching algorithms provides helpful information to inform that trade-off. The final balance after matching gives information about how successful the match was overall. The literature calls for a variety of tests to check the balance of a matched sample (Ho, 2007) and there are no hard thresholds where imbalance is considered insignificant (Imai, 2008). Instead, each covariate should be evaluated in the context of the current study, with consideration to how it may affect the outcomes of interest. Covariates that were not included in the propensity score model should also be checked (Ho, 2007). It is more important to show balance among covariates that are thought to have a significant direct effect on the outcomes. Even a balanced sample can have random differences, so imbalance in covariates that are not thought to affect the outcome may be acceptable (Austin, 2009).

SAS MACRO TO EVALUATE BALANCE

We have developed the SAS macro PMDIAG (Appendix 1) to run a series of tests suggested by the literature to check the balance of a matched sample. Testing of balance is independent of propensity score calculation and matching method. A diagnostic report in RTF format is created, which is easy to read for non-programmers and can be shared with clients. A typical matching program will create the propensity score using PROC LOGISTIC, call a matching macro and then call the balance testing macro PMDIAG.

PMDIAG requires the user to provide the name of the pre-match data set with all patients and the post-match data set that includes only matched patients. Both data sets must contain variables for patient id, case, the propensity score and all covariates included in the balance tests. The post-match dataset must also contain a variable uniquely identifying the matched records (i.e., a match id). The names of the required variables and the values of the case variable are provided to the macro as parameters so renaming or re-coding variables is not required. Categorical variables should be included as dummy (binary) variables for each level. The macro call must also include the name of the RTF file to create. Optional parameters include a flag to skip creation of the propensity score distribution graphs, the ODS style to use while creating the RTF file and column width values for the covariate summary table. In a one-to-many match, the macro optionally allows one to specify a weight variable to adjust for the number of controls in the post-match distribution. The weight variable itself is typically creating during the matching process where various weights may be used depending on the matching process used. All titles created by the macro use TITLE3, so TITLE1 and TITLE2 can be defined before the macro is called to create program-specific headers. This macro was written and tested using SAS version 9.2.

MACRO OUTPUT

SAMPLE SIZE

The first section of the output file provides sample sizes before and after the match in order to assess how many cases and controls were matched and if the post-match sample sizes are adequate for analysis. The FREQ procedure is used to display the overall frequency of the case and cohort patients in the pre-match dataset (Table 1), the frequency of cases and controls with a valid propensity score in the pre-match dataset (not shown) and the frequency of cases and controls in the matched dataset (Table 2). If more than one control was matched for each case, a frequency of the number of matches per case will be printed (not shown).

Table 1. Pre-match cohort counts

Case				
case	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Control	10000	97.09	10000	97.09
Case	300	2.91	10300	100.00

Table 2. Post-match cohort counts

Case				
case	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Control	296	50.00	296	50.00
Case	296	50.00	592	100.00

PROPENSITY SCORE DISTRIBUTIONS

The next section displays the distribution of propensity score as a histogram among cases and controls before and after the match. The TTEST procedure creates the graph using ODS Graphics and selecting the SummaryPanel:

```
ods rtf select SummaryPanel;
ods listing exclude SummaryPanel;
ods graphics on;
proc ttest data=&predat.(rename=(&prob_graph.=Propensity)) plot;
  title3 "Pre-match propensity scores by cohort";
  class &case_control.;
  var Propensity;
run;
ods graphics off;
```

For the post-match distribution graphs, if the weight parameter is defined, an integer weight variable is created in the post-match data set to use with the FREQ statement in PROC TTEST.

Pre-match distributions can be examined for reasonableness and to evaluate the overlap between the two cohorts. If the distributions are similar, a close match can likely be created without much loss of sample size. However, if there is little or no overlap between the distributions a large portion of the cases may not be matched. Modifying the propensity score model in this situation may lead to larger overlap but likely at the expense of balance for some variables. Other statistical techniques (e.g., regression) might be considered as an alternative in these instances or they may be used after the match to adjust for the remaining imbalance.

The pre- and post-match propensity score distributions among cases can be compared to evaluate how the cases that were retained are representative of the pre-match sample. Ideally, these distributions will be very similar. Post-match propensity score distribution between the cases and controls should also be similar, although this alone is not enough to show balance (Austin, 2009).

The distribution of difference in propensity score between matched pairs (not shown) is created as a visual check that it centers on zero and is within the calipers specified during the matching process (if calipers were used). The distribution of propensity score in patients that were excluded from the matched sample (not shown) provides a visual check of the remaining overlap between the unmatched records and whether modification of the matching criteria might result in additional matches.

Figure 1. Pre-match propensity scores by cohort

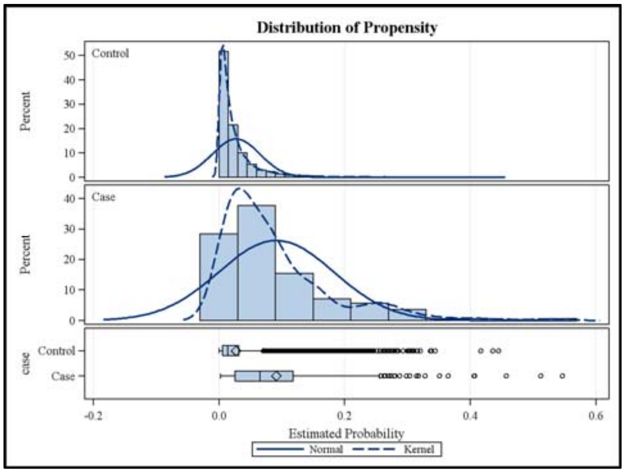

Figure 2. Weighted post-match propensity scores by cohort

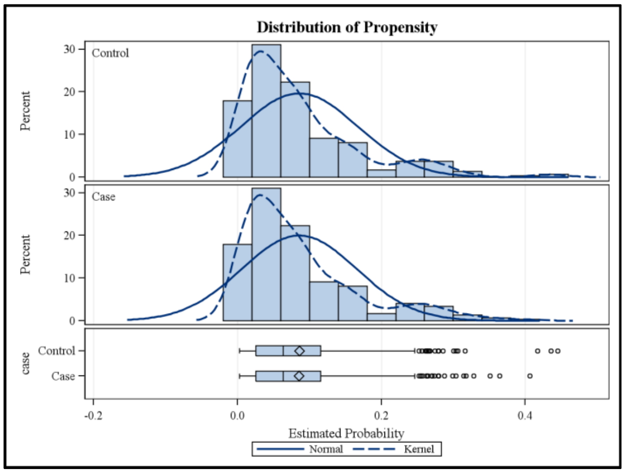

TABLE OF COVARIATES TO CHECK BALANCE

The final summary table created by the macro lists all the propensity score model inputs and other additional baseline covariates, with descriptive statistics pre- and post-match (Table 3). If more than one control has been matched to each case, these statistics weight the control values as needed. Binary variables are treated as continuous variables for all calculations, including standard difference.

PROC TTEST is used within macro PMTESTS to calculate means, standard deviations and the p-value of the difference in means between cases and controls for each covariate. Summary data sets are created for the pre-match and post-match samples that contain one observation per covariate, with variables for case mean, control mean, case standard deviation, control standard deviation and the p-value of the t-test. An overall summary data set is created by merging these two data sets and the standard differences in the pre- and post-match samples are calculated using the following formula:

```
stddiff = 100*(case_mean - cont_mean) / (sqrt((case_stddev**2 + cont_stddev**2)/2));
```

The summary table is printed using PROC REPORT.

The means can be visually inspected for balance (Austin, 2008). Additionally, residual differences in means can be evaluated for practical and/or clinical meaningfulness, since they are provided in the units of the baseline measure (Ho, 2007). Standard differences are used to compare the difference in the covariate between the pre- and post-matched sample, since the calculation is not dependent on sample size, which will generally be smaller after matching (Ho, 2007; Imai, 2008). There is no rule as to how small the standard difference value should be, although 10% is often used as a rough guideline (Rosenbaum and Rubin, 1985).

The p-value of a t-test comparison of the means pre- and post-match can also be included. The t-test is generally not considered to be an appropriate test of balance post-match and its use after propensity score matching has been widely criticized (Austin, 2008) with the standardized difference often suggested as a more appropriate alternative.

Table 3. Pre-match cohort counts

Variable	Variable Description	Pre Match Case Mean	Pre Match Control Mean	Pre Match Diff	Pre Match Stand Diff (%)	Post Match Case Mean	Post Match Control Mean	Post Match Diff	Post Match Stand Diff (%)	Pre Match T-test p-value	Post Match T-test p-value
male	Male	0.5233	0.5565	-0.0332	-6.65	0.5203	0.5203	0.0000	0.00	0.2547	1.0000
age	Age	59.7149	51.9488	7.7660	98.21	59.4892	59.8245	-0.3353	-4.37	<.0001	0.5953
b_hyperten	Baseline Hypertension	0.7667	0.7651	0.0016	0.37	0.7635	0.7669	-0.0034	-0.80	0.9497	0.9229
b_hyperlip	Baseline Hyperlipidemia	0.7400	0.8100	-0.0700	-16.81	0.7466	0.7196	0.0270	6.10	0.0067	0.4581
b_rxcost	Baseline Pharmacy Costs	1719.7	2120.2	-400.476	-22.22	1728.9	1798.6	-69.6710	-4.22	0.0003	0.6076
b_medcost	Baseline Medical Costs	3985.7	3038.4	947.2411	9.19	4038.9	2693.0	1345.911	13.30	0.1782	0.1063
region_mw	Region Midwest	0.1567	0.4053	-0.2486	-57.53	0.1588	0.1757	-0.0169	-4.52	<.0001	0.5826
region_so	Region South	0.2933	0.1982	0.0951	22.21	0.2939	0.3074	-0.0135	-2.94	0.0004	0.7205
region_we	Region West	0.1833	0.1957	-0.0124	-3.15	0.1858	0.1791	0.0068	1.75	0.5945	0.8318
region_ne	Region Northeast	0.3667	0.2008	0.1659	37.39	0.3615	0.3378	0.0236	4.95	<.0001	0.5471
b_ambcnt	Count of Baseline Ambulatory Visits	12.4033	12.4438	-0.0405	-1.17	12.4122	12.6791	-0.2669	-7.56	0.8435	0.3579

CONCLUSION

A standard macro to evaluate balance after propensity score matching with easy-to read output allows for thorough testing of balance. This macro runs multiple tests of balance quickly and can be run as many times as needed to compare different models of propensity score in order to create a better final matched sample.

REFERENCES

Austin, P.C. (2008). A Critical Appraisal of Propensity-Score Matching in the Medical Literature Between 1996 and 2003. *Statistics in Medicine*, 27(12), 2037-2049.

Austin, P.C. (2009). Balance Diagnostics for Comparing the Distribution of Baseline Covariates Between Treatment Groups in Propensity-Score Matched Samples. *Statistics in Medicine*, 28(25), 3083-107.

D'Agostino, R.B. (1998). Propensity Score Methods for Bias Reduction in the Comparison of a Treatment to a Non-Randomized Control Group. *Statistics in Medicine*, 17(19), 2265-2281.

Ho, D.E. (2007). Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference. *Political Analysis*, 15(3), 199-236.

Imai, K., King, G., Stuart, E.A. (2008). Misunderstandings Between Experimentalists and Observationalists About Causal Inference. *Journal of the Royal Statistical Society, Series A*, 171, Part 2, 481-502.

Rosenbaum, P. R., Rubin, D.B. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika,* 70 (1), 41-55.

Rosenbaum, P. R., Rubin, D.B. (1985). Constructing a Control Group Using Multivariate Matched Sampling Methods That Incorporate the Propensity score. *The American Statistician*, 39(1), 33-38.

CONTACT INFORMATION

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

Erin Hulbert, MS
Optum Life Sciences, Health Economics and Outcomes Research
12125 Technology Drive
Eden Prairie, MN 55344, USA
erin.hulbert@optum.com
www.optum.com/life-sciences

APPENDIX 1

```
/*******************************************************************************
Program Name:         pmdiag.sas
Brief Description:    Create output to evaluate propensity score match
Macro arguments:
 predat       = Input dataset name of all patients, before match including libname
                if not temporary (e.g., mainlib.analysis).
                Must include both the cases and control populations.
 postdat      = Input dataset name of matched cases and controls.
                including libname if not temporary (e.g., mainlib.analysis).
                Must include both the cases and control populations.
 idvar        = Name of unique patient variable (e.g. patient_id).
 prob_graph   = Name of propensity score variable to use in report graphs.
 case_control = Name of binary variable distinguishing case vs control records.
 caseval      = Value of &case_control. variable indicating a case record.
 contval      = Value of &case_control. variable indicating a control record.
 outpath      = Path for diagnotic report including filename excluding ".rtf".
 style        = SAS ODS style to use when creating RTF file (defaults to rtf).
 varlst       = Variable list used to create propensity score.
 varlst_test  = Variable list to compare pre and post matched standardized
                differences.  (Should include all variables in your propensity
                model (varlst) and may include additional variables).
 labwidth     = Width of label column in inches (defaults to 1.0).
 cohwidth     = Width of statistics columns in inches (defaults to 0.7).
 pwidth       = Width of T-test p-value columns in inches (defaults to 0.7).
 dographs     = Y or N to include propensity score distribution graphs.
                Setting to N may improve program speed with large datasets.
 weight       = Name of variable to use for weights if number of controls >1.
 matchid      = Name of variable that contains match ID.
*******************************************************************************/

/* Macro for calculating statistics and independent t-tests */
%macro pmtests(insas,mpre,case_control,caseval,contval,varlst_test,weight);
  ods output Statistics=pm_stats(index=(variable))
           TTests=pm_ttest(index=(variable))
           Equality=pm_var_test(index=(variable));
  proc ttest data=&insas. nobyvar;
    title4 "Variable differences (&insas.)";
    class &case_control.;
    var &varlst_test.;
    format _all_;
    %if &weight. ne %then %do;
      weight &weight.;
    %end;
  run; quit;
  ods output close;

  data pm_ttest(index=(variable));
    merge pm_ttest pm_var_test(drop=method);
    by variable;
    variable = lowcase(variable);
    length choose $100;
    retain choose;
    if first.variable then do;
      if (probf > 0.05) then choose = 'Pooled';
      else choose = 'Satterthwaite';
    end;
    if method = choose then output;
    keep variable tvalue probt choose;
    rename
      tvalue = &mpre.ttst_tvalue
      probt  = &mpre.ttst_probt
      choose = &mpre.ttst_choose;
  run;

  data _case(index=(variable) rename=(n=&mpre.case_n
             mean=&mpre.case_mean  stddev=&mpre.case_stddev))
       _cont(index=(variable) rename=(n=&mpre.cont_n
             mean=&mpre.cont_mean  stddev=&mpre.cont_stddev))
       _diff(index=(variable) rename=(n=&mpre.diff_n
             mean=&mpre.diff_mean  stddev=&mpre.diff_stddev));
    set pm_stats;
    variable = lowcase(variable);
    /* For weighted case replace stddev estimate with n* stderr
       (this would leave stddev unchanged in unweighted case) */
    %if &weight. ne %then %do;
      if n gt 0 then stddev = stderr*sqrt(n);
      else stddev = .;
    %end;
    if class = "&caseval." then output _case;
    else if class = "&contval." then output _cont;
    else output _diff;
    keep variable n mean stddev;
  run;

  data &mpre.stats_one(index=(variable));
    length variable $32;
    merge _case _cont _diff pm_ttest;
    by variable;
  run;
%mend;

/* Macro to run diagnostics on the match and produce a report */
%macro pmdiag(predat,postdat,idvar,prob_graph,case_control,caseval,contval,
              outpath,style,varlst,varlst_test,labwidth,cohwidth,pwidth,
              dographs,weight,matchid);
  options orientation=landscape missing=' ' center nodate;
  %local int_weight;
  /* Set default parameters */
  %if &style. = %then %let style = rtf;
  %if &labwidth. = %then %let labwidth = 1.0;
  %if &cohwidth. = %then %let cohwidth = 0.7;
  %if &pwidth. = %then %let pwidth = 0.7;
  %if %upcase(&dographs.) = N %then %let dographs = N;
  %else %let dographs = Y;

  /* calculate approximate integer weight for post-histogram
     (which doesn't work with fractional weights in SAS 9.2) */
  %if (&dographs. = Y) and (&weight. ne ) %then %do;
    proc sql noprint;
      select min(&weight.) into :minweight
      from &postdat.
      where (&weight. gt 0);
    quit;

    data _null_;
      if &minweight. le 10 then
        call symput('multweight',trim(left(fact(ceil(1/&minweight.)))));
      else call symput('mult_weight',trim(left(1/&minweight.)));
      stop;
    run;

    data _postdatmod;
      set &postdat.;
      int_weightmod = ceil(&multweight.*&weight.);
    run;

    %let postdatmod = _postdatmod;
    %let int_weight = int_weightmod;
  %end;
  %else %let postdatmod = &postdat.;

  %if &matchid. ne %then %do;
    proc sql;
      /* Calculate case-control propensity score differences */
      create table outt as
      select (case.&prob_graph. - cntrl.&prob_graph.) as Case_Control_Diff,
             case.&matchid. as matchid
      from &postdat.(where=(&case_control. = &caseval.)) as case,
           &postdat.(where=(&case_control. = &contval.)) as cntrl
      where case.&matchid. = cntrl.&matchid.;
      /* calculate control counts */
      create table control_counts as
      select count(*) as control_count
      from outt
      group by matchid;
    quit;
  %end;

  ods rtf file = "&outpath..rtf" bodytitle style=&style. notoc_data;
  ods escapechar = '^';
  /*print a table of variable names and lables used to create PS*/
  %if (%symexist(varlst) = 1) %then %do;
    %if %quote(&varlst.) ne %then %do;
      data varnames;
        if _n_ = 1 then set &predat.;
        length Variable $32 Label $260;
        array varnama &varlst.;
        do over varnama;
          Variable = lowcase(vname(varnama));
          Label = vlabel(varnama);
          output;
        end;
        keep variable label;
        stop;
      run;
      proc print noobs; title3 'Variables in propensity score model'; run;
    %end;
  %end;

  /*print frequencies of sample size pre- and post-match*/
  proc freq data = &predat.;
    title3 "Pre-match cohort counts";
    tables &case_control.;
  run;
  proc freq data = &predat.;
    where &prob_graph. gt .Z;
    title3 "Pre-match cohort counts with non-missing propensity score";
    tables &case_control.;
  run;
  proc freq data = &postdat.;
    title3 "Post-match cohort counts";
    tables &case_control.;
  run;
  %if &matchid. ne %then %do;
    proc freq data = control_counts;
      title3 "Post-match number of control matches per case";
      tables control_count;
    run;
  %end;

  /*create propensity score distribution graphs*/
  %if &dographs. = Y %then %do;
    ods rtf select SummaryPanel;
    ods listing exclude SummaryPanel QQPlot;
    ods graphics on;
    proc ttest data=&predat.(rename=(&prob_graph.=Propensity)) plot;
      title3 "Pre-match propensity scores by cohort";
      class &case_control.;
      var Propensity;
    run;
    ods graphics off;

    ods rtf select SummaryPanel;
    ods graphics on;
    ods listing exclude SummaryPanel QQPlot;
```

```
proc ttest data=&postdatmod.(rename=(&prob_graph.=Propensity)) plot;
  %if &int_weight. ne %then %do;
    title3 "Weighted post-match propensity scores by cohort";
  %end;
  %else %do;
    title3 "Post-match propensity scores by cohort";
  %end;
  class &case_control.;
  var Propensity;
  %if &int_weight. ne %then %do; freq &int_weight.; %end;
run;
ods graphics off;
ods rtf exclude all;

%if &matchid. ne %then %do;
  ods rtf select SummaryPanel;
  ods listing exclude SummaryPanel QQPlot;
  ods graphics on;
  proc ttest data=outt plot;
    title3 "Post-matched paired propensity score differences";
    var Case_Control_Diff;
    label Case_Control_Diff = 'Case-Control Propensity Difference';
  run;
  ods graphics off;
  ods rtf exclude all;
%end;

   /*if there is at least one un-matched case and one-unmatched control,
     create output for the un-matched patients*/
proc sort data=&predat.(keep=&idvar. &prob_graph. &case_control.) out=insort;
  by &idvar.;
run;
proc sort data=&postdat.(keep=&idvar.) out=outsort;
  by &idvar.;
run;
data nomatch;
  merge insort(in=inin) outsort(in=inout);
  by &idvar.;
  if inout then delete;
run;

proc freq data = nomatch;
 tables &case_control. /out = nomatch_freq;
run;

data _null_;
 set nomatch_freq nobs =  nomatch_cnt;
 call symput('nomatch_cnt', nomatch_cnt);
run;

%if &nomatch_cnt. = 2 %then %do;
  ods rtf select SummaryPanel;
  ods listing exclude SummaryPanel QQPlot;
  ods graphics on;
  proc ttest data=nomatch(rename=(&prob_graph.=Propensity)) plot;
    title3 "Propensity scores for dropped records by cohort";
```

```
      class &case_control.;
      var Propensity;
    run;
  %end;
  ods graphics off;
%end;
ods rtf exclude all;

/*create summary table of covariates*/
%pmtests(&predat., pre_, &case_control.,&caseval.,&contval.,&varlst_test.)
%pmtests(&postdat.,post_,&case_control.,&caseval.,&contval.,&varlst_test.,&weight.)

ods listing close;
data varnames_test;
  if _n_ = 1 then set &predat.;
  length variable $32 label $260;
  array varnama &varlst_test.;
  do over varnama;
    variable = lowcase(vname(varnama));
    label = vlabel(varnama);
    vorder = _i_;
    output;
  end;
  keep variable label vorder;
  stop;
run;
proc sort nodupkey; by variable; run;

data stats_one;
  merge pre_stats_one post_stats_one varnames_test;
  by variable;
  length pre_stddiff post_stddiff post_stddiff2 post_diffred pre_diff post_diff 8;
  format
    pre_stddiff post_stddiff post_stddiff2 post_diffred 6.2
    pre_diff post_diff 8.4;
  label
    pre_stddiff     = 'Pre-Match Standardized Difference                         '
    post_stddiff    = 'Post-Match Standardized Difference, (Pre-match variances) '
    post_stddiff2   = 'Post-Match Standardized Difference, (Post-match variances)'
    post_diffred    = 'Post-Match Difference Percent Reduction                   '
    pre_diff        = 'Pre-Match Difference                                      '
    post_diff       = 'Post-Match Difference                                     '
    pre_ttst_probt  = 'Pre-Match T-Test P-Value                                  '
    post_ttst_probt = 'Post-Match T-Test P-Value                                 ';

  pre_denom  = sqrt((pre_case_stddev**2 + pre_cont_stddev**2)/2);
  post_denom = sqrt((post_case_stddev**2 + post_cont_stddev**2)/2);
  pre_diff  = pre_case_mean - pre_cont_mean;
  post_diff = post_case_mean - post_cont_mean;
  if pre_denom gt 0 then do;
    pre_stddiff  = 100*(pre_case_mean - pre_cont_mean)/pre_denom;
    post_stddiff = 100*(post_case_mean - post_cont_mean)/pre_denom;
  end;
  else do;
    pre_stddiff  = .;
    post_stddiff = .;
    end;
    if post_denom gt 0
      then post_stddiff2 = 100*(post_case_mean - post_cont_mean)/post_denom;
    else post_stddiff2 = .;
    if abs(pre_case_mean - pre_cont_mean) gt 0
      then post_diffred = 100*(1 - abs((post_case_mean - post_cont_mean)
                          /(pre_case_mean - pre_cont_mean)));
    else if ((pre_case_mean - pre_cont_mean) = 0) and
            ((post_case_mean - post_cont_mean) = 0)
      then post_diffred = 0;
    else post_diffred = .;
    drop pre_denom post_denom;
  run;
  proc sort;
    by vorder;
  run;
  ods rtf select all;

  title3 'Variable Balance Checks';
  proc report data=stats_one nowd split = "\" style=[protectspecialchars=off];
    column variable label pre_case_mean pre_cont_mean pre_diff pre_stddiff
           post_case_mean post_cont_mean post_diff post_stddiff2 pre_ttst_probt
           post_ttst_probt;
    define variable       / display "Variable"
                            left  style={cellwidth=&labwidth. in};
    define label          / display "Variable\Description"
                            left  style={cellwidth=1.5 in};
    define pre_case_mean  / display "Pre\Match\Case\Mean"
                            right style={cellwidth=&cohwidth. in};
    define pre_cont_mean  / display "Pre\Match\Control\Mean"
                            right style={cellwidth=&cohwidth. in};
    define pre_diff       / display "Pre\Match\Diff"
                            right style={cellwidth=&cohwidth. in};
    define pre_stddiff    / display "Pre\Match\Stand\Diff (%)"
                            right style={cellwidth=&cohwidth. in};
    define post_case_mean / display "Post\Match\Case\Mean"
                            right style={cellwidth=&cohwidth. in};
    define post_cont_mean / display "Post\Match\Control\Mean"
                            right style={cellwidth=&cohwidth. in};
    define post_diff      / display "Post\Match\Diff"
                            right style={cellwidth=&cohwidth. in};
    define post_stddiff2  / display "Post\Match\Stand\Diff (%)"
                            right style={cellwidth=&cohwidth. in};
    define pre_ttst_probt / display "Pre\Match\T-test\p-value"
                            right style={cellwidth = &pwidth. in};
    define post_ttst_probt/ display "Post\Match\T-test\p-value"
                            right style={cellwidth = &pwidth. in};
  run;
  ods rtf exclude all;
  ods rtf close;
  ods listing;
%mend;
```