Nested case control sampling macro- user guide

Description and citation:

The nCCsampling macro performs incidence density sampling (ie matches cases and controls on person time at risk) and gives the user an option to match on additional confounders (can match on up to 3 categorical or continuous variables) or on a pre-computed confounder summary score (eg disease risk score or propensity score- nearest neighbor matched within a caliper provided by the user) using data from a source cohort. This macro requires a person-level analytic data file containing information on patient identifiers, exposure, confounding variables or a summary confounder score, and timing of the outcome/censoring time. This macro also performs conditional logistic regression analysis that appropriately accounts for incidence density sampling matched design after nested case control sampling within a cohort.

If you use the nCCsampling macro, please cite

Desai RJ, Glynn RJ, Wang S, Gagne JJ. Performance of Disease Risk Score Matching in Nested Case-Control Studies: A Simulation Study. *American journal of epidemiology*. 2016;183(10):949-957.

The core of our macro is based on the following reference:

Richardson DB. An incidence density sampling program for nested case-control analyses. *Occupational and environmental medicine* 2004. 61(12), e59-e59.

Nested case control sampling macro- user guide

Structure:

```
%macro nCCsampling (logfile= "nul:", total_samples=1,
in_data= , n_controls= , outcome_var= , ptime_var= , id_var= ,
out_data= , cat_covariate_list= , full_covariate_list= ,
additional_matching_var=0, match1_var=dummy1, match2_var=dummy2,
match3_var= dummy3,
score_match='no', score_var=dummy4, caliper=0.01, sorting_var=number,
variable_ratio_match='no',
lib= work,
save_n_info= n_info, save_estimates=estimates);
```

Required parameters:

1. **In_data =** Input dataset in a cohort structure (ie one row per patient) with all the covariate, exposure, and outcome information.

2. **n_controls =** Maximum number of controls to be selected for each case (Numeric value).

3. **outcome_var=** Variable that is used to define a case (this variable must have value=1 for cases).

4. **ptime_var =** Person time variable, time axis of the study. Based on this variable, cases will be matched on their outcome date to event-free controls up to that time. It is assumed that follow up for cases stop on the day of the event.

5. **id_var =** Unique identification number variable for everyone in the cohort.

6. **out_data=** Where the dataset containing matched cases and controls should be saved. Remember, if you supply total_samples>1 (see **Optional parameters-2** discussion below), you will get multiple datasets each with a unique numeric suffix. Eg. Out_data1, out_data2 and so on. By default, these datasets will be saved in the work library. If you wish to save it elsewhere, please specify the option ‘lib’, described under **Optional parameters-7** below.

7. **cat_covariate_list:** The list required here is for the class statement of ‘proc logistic’ for performing conditional logistic regression. List all the categorical covariates here.

8. **full_covariate_list:** List all the covariates that you want to adjust for ***in addition to*** your matching here. This list goes into the model statement of ‘proc logistic’. Remember, ***do not*** put the matched covariates in here since matching takes care of that confounding.

Optional parameters:

1. **logfile:** For big datasets where sampling needs to be performed many times, the log window may fill up repeatedly causing the need for users to manually clear it. To prevent that, the default value of this parameter is "nul:", meaning no log will be printed. However, you may want to specify a filepath within quotes to save the log as a separate file (eg logfile= "C:\Users\Desktop\log").

2. **total_samples:** Number of times you want to perform incidence density sampling. Default value is 1.
   NOTE- For score based matching, performing the sampling more than once should always result in the same parameter estimates as this method picks the 'best' matches based on the absolute difference between the scores for cases and controls, which does not vary. However, for traditional matching on a few covariates, sampling more than once will almost always result in different parameter estimates as long as a pool of eligible controls that is larger than user-specified maximum number of controls ('n_controls') is available for each case because this matching scheme will select controls randomly when a larger pool is available. In these circumstances, researchers may want to sample multiple times and present the coefficients based on the distribution of these coefficients across samples to explicitly acknowledge uncertainty in their estimation.

3. **additional_matching_var:** Default is 0, meaning cases and controls will only be matched on person time at risk. User can specify 1, 2 or 3 for up to 3 additional matching variables. If 1, 2 or 3 specified, following parameters ***must*** be supplied,
   a. **match1_var:** Name of the 1st matching variable
   b. **match2_var:** Name of the 2nd matching variable
   c. **match3_var:** Name of the 3rd matching variable

   These variables can be binomial, categorical or continuous; but they need to be provided in a numeric format (eg 0/1 if binomial, 0/1/2/3/4 if categorical). If they are continuous, whole numbers should be provided. This macro performs ***exact match*** on these three covariates when specified. Eg when matching on age, sex and calendar year, a 70 year old female patient beginning treatment in 2005 will only be matched with 70 year old females beginning treatment in 2005. If you want to relax the criteria (eg match within 2 years of age), do so within the source code.

4. **score_match:** 'yes' or 'no'. Default is 'no'. Change this to 'yes' when you want to match on a summary score such as disease risk score and supply the following parameters,
   NOTE: If you specify score_match='yes', it will override the 'additional_matching_var' algorithm. So your cases and controls will be matched on 1) person time at risk and 2) the summary score, but not on the additional variables specified.
   a. **score_var:** Name of the variable on which the match needs to be performed.
   b. **caliper:** Numerical value for the distance within which the cases and controls must be matched.

c. **sorting_var:** Variable name on which all the identified potential controls will be sorted. The default is a variable called ‘number’, which is a randomly generated variable that will be used to pick out controls randomly out of a larger pool of eligible controls for traditional non-score based matching. However, for score based matching, we don’t want this process to be random. Instead, we want to pick the ‘best’ matches out of a pool of available controls. ***This variable should only be changed if score_match=’yes’***, in order to make sure that among all the identified potential controls, those with the least difference in the summary score from cases (ie nearest neighbors) are selected. Specify ***sorting_var=abs_diff*** when score_match=’yes’. The variable abs_diff is generated dynamically in the macro.

5. **variable_ratio_match:** ‘yes’ or ‘no’. Default is ‘no’, meaning if the cases for which pre specified number of controls (see n_controls discussion above) cannot be found after matching on desired variables, they will be dropped. Alternative is ‘yes’, which will include all the cases for whom at least 1 control is found. The ‘out_data’ dataset will have a variable with information on the total number included in each risk set.

6. **lib:** Name of the library where the output dataset(s) will be saved. The default is work.

7. **save_n_info:** Name of the SAS dataset where output containing information on number of successfully matched cases and controls will be saved. The default is a dataset called ‘n_info’, which will be saved in the work library if you do not specify lib in 6 above.

8. **save_estimates**: Name of the SAS dataset where output containing parameter estimates will be saved. The default is a dataset called ‘estimates’, which will be saved in the work library if you do not specify lib in 6 above.

Example dataset and calls:

Following calls can be directly executed in the ‘ncc_example’ dataset that is provided with this guide. It is one of the simulated datasets used in the reference paper (Desai RJ et al. *Am J Epi*. 2016;183(10):949-957.), where you can find the details on how it was generated.

For traditional matching

```
%nCCsampling (logfile= 'C:\Users\rjd48\Desktop\example log',
in_data= ncc_example, n_controls= 5, outcome_var= outcome, ptime_var= survt,
id_var= id, score_match='no', additional_matching_var=2, match1_var=c1,
match2_var=c2,
variable_ratio_match='yes',
out_data= example_matched,
cat_covariate_list= treatment (ref='0'), full_covariate_list= treatment c3 c4
c5 c6);
```

For score based matching

```
%nCCsampling (logfile= 'C:\Users\rjd48\Desktop\example log',
in_data= ncc_example, n_controls= 5, outcome_var= outcome, ptime_var= survt,
id_var= id, score_match='yes', score_var= drs, caliper=0.025,
additional_matching_var=0,
variable_ratio_match='yes',
out_data= example_matched,
cat_covariate_list= treatment (ref='0'), full_covariate_list= treatment);
```