



Paper AS09

Incidence Density Sampling for Nested Case-Control Study Designs

Quratul Ann, IQVIA, London, United Kingdom
Rachel Tham, IQVIA, London, United Kingdom

ABSTRACT

The nested case-control design is commonly used in safety studies, as it is an efficient approach to an epidemiological investigation. When the design is implemented appropriately, it can yield unbiased estimates of relative risk; and in exchange for a small loss in precision, considerable improved efficiency and reduction in costs can be achieved. There are several ways of sampling controls to cases in a nested case-control design. The most sophisticated way is via incidence density sampling. Incidence density sampling matches cases to controls based on the dynamic risk set at the time of case occurrence, where the probability of control selection is proportional to the total person-time at risk. Incidence density sampling alleviates the rare disease assumption; however, it is rarely used due to its computational complexity. This paper presents a novel and simple Statistical Analysis System (SAS) program for incidence density sampling, which could minimise bias and introduce a more appropriate way of optimising the matching of controls to cases.

INTRODUCTION

Retrospective studies gather available previous exposure information for a clearly defined source population and the outcome event is determined for all members of the cohort⁴. Retrospective studies facilitate an efficient approach to draw inferences and determine the association between exposure and disease prevalence particularly for rare outcomes and with long-term disease conditions. Real-world data (RWD) are a valuable source of information for investigating health-related outcomes in human populations. RWD includes information on routine health care data captured in electronic health records (EHR). There is an increasing interest in generating real-world evidence in studies that use electronic medical record data to investigate the effectiveness and safety of treatments. RWD reflects the real-life utility of treatments in the clinical setting and includes the extensive amount of information on subjects with varying characteristics, comorbidities, diagnosis, and treatments. The use of RWD in post-authorisation safety studies (PASS) has become the standard practice.

Most retrospective database studies can be susceptible to bias such as observer bias, selection bias and recall bias. Observer bias occurs when there are differences in the recording of exposure information between cases and control based on their outcome and the investigator's knowledge of an individual's disease status. Selection bias occurs when cases (or controls) are included (or excluded) from a study because of some characteristics they exhibit, which are related to exposure to the risk factor under evaluation¹⁴. Confounding variables may go unrecognised due to inadequate knowledge of how these interrelate with the outcome of interest. Therefore, such studies require the application of strong principles of epidemiologic study design and sophisticated analytical methods to reduce bias and adjust for confounding.

The case-control study simply compares a case, subjects with the target disease or outcome of interest, with a matched control, who do not have the target disease or the outcome of interest, in the source eligible cohort. The directionality of the effect of the case-control study is from the outcome to the past exposure variable of interest to determine the association between exposure and the targeted disease. The risk in case-control studies is measured using odds ratios (OR) between cases and controls. If the target disease is rare OR provides a good estimate of both risk ratio and rate ratio in the analysis. However, the case-control study design has a number of limitations, including selection bias - largely the control group should be representative of the population of interest and it's prone to the methodological errors in the analysis.

Steps to follow for case-control selection:

1. Define the source population for the study.
2. Define exposure(s) and the outcome of interest.
3. Identify cases and determine their exposure status.
4. Identify controls and determine their exposure status
5. Calculate and interpret the exposure odds ratio.



Controls:

- Sample from the source eligible cohort that would give rise to cases that are still at risk until the end of the study time period.
- Match controls with cases on various demographic characteristics such as age, gender and year of initiation of disease
- Sampling with a replacement or without replacement
- Representative of the source population and controls are allowed to become the cases depending on the time at risk
- An individual who developed the disease during the study time period may also be selected as a control for another case and contrariwise

The paper presents a novel and simple Statistical Analysis System (SAS) program for incidence density sampling that could minimise selection bias and introduce a more appropriate way of optimising the selection of controls to cases in the study. This paper will briefly cover and explain the following topics:

- Nested case-control studies
- Incidence density sampling
- SAS program for selecting controls using incidence density sampling – Example study

NESTED CASE-CONTROL STUDIES

Case-control studies performed within the context of cohort studies are commonly called “nested,” “synthetic,” or “case-control within cohort” studies¹. The nested case-control design is frequently applied to the study of the disease exposure-outcome relationships. In a nested case-control study, cases of a disease are identified and, for each case, a number of healthy matched controls (one or more) are selected from among those in the source cohort who are disease-free and are still under the observation at the time when the case developed the disease in the analysis. The nested case-control study design is an extremely powerful approach that provides an enormous reduction in costs, data collection efforts, and analysis compared to a full study cohort approach. The nested case-control study achieves all this with a relatively minor loss in statistical efficiency. The nested case-control study minimises selection bias and recall bias (cases and controls may recall past exposure differently) in the study.

In clinical research, when information on essential epidemiological parameters such as the outcome event, person-time to the outcome event, and the time-dependent covariates are available in the source cohort, then it's particularly practical to create incidence density sampled nested case-control data.

INCIDENCE DENSITY SAMPLING

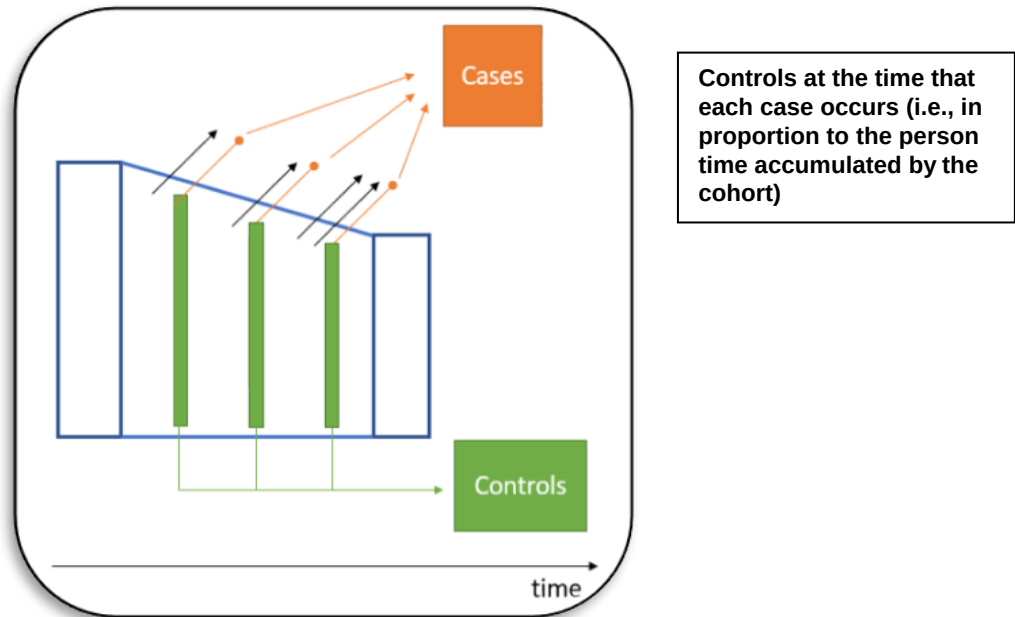
Incidence density sampling is the least biased method for control sampling in nested case-control studies¹³. This allows obtaining a representative sample of person-time at risk of eligible cohort members within a case-control study. The controls are sampled from the risk population at the time of incidence of each case. The probability of selecting the control from the source population at risk is proportional to the total person-time at risk in the source population. The main purpose of this method is to reduce the bias from the changes observed in the prevalence of the exposure in the analysis. This technique is recommended for producing unbiased results in nested case-control analyses⁸.

Index date is defined as the first occurrence of the case in the study period. Each individual case is assessed and then corresponding one (or more) controls are selected at random for each respected case from the source cohort at risk at the time of each case identification, as highlighted in the SAS program below.

The density case-control cohort selection:

The selection of controls is conditional on those who have survived at least as long as the index case, see figure 1. This procedure is consistently repeated for all cases present in the cohort.

Figure 1: Incidence Density Case-Control Sampling – Methods of selecting controls



A standard 3 × 2 table, as shown in table 1, can be used to evaluate the data obtained from the case-control study design. Each cell in the table below presents the count for cases and for the control group categorised by exposure status (presence vs absence) and shown as:

Table 1: Illustration of a standard 3 × 2 table Contingency Table

	Exposed	Unexposed
Cases	A	B
Controls	C	D
Person time	T1	T0
Source population	A + C	B + D

In case-control studies, neither exposed nor unexposed person-time (T1, T0) can be directly evaluated from the available data and the control groups are used to reflect the person-time distribution of the exposure in the source population. The exposure OR from the case-control study directly estimates the relative risk (incidence rate ratio) in the source population as shown below:

$$\begin{aligned}
 \text{Rate ratio (relative risk)} &= \text{Incidence exposed} / \text{Incidence unexposed} \\
 \text{Rate ratio} &= (A / T1) / (B / T0) \\
 &\text{can be rearranged to obtain} \\
 \text{Rate ratio} &= (A / B) / (T1 / T0) \\
 &\text{Which is same as} \\
 &\text{Odds of exposure in diseased} / \text{ratio of exposed to unexposed person-time}
 \end{aligned}$$

SAS PROGRAM FOR SELECTING CONTROLS USING INCIDENCE DENSITY SAMPLING

The provided SAS program for incidence density sampling has been kept simple for ease of understanding and illustration purposes. In the presented program, we have matched the patients on age at risk variable; however, this program can be readily expanded further to match on other potential confounder variables including fixed and time-varying component variables. Generally, in most studies, it's sufficient to match the patients on the age variable. The SAS program can be instantly adapted for fixed component variables, such as gender, by restricting the dataset to males and females separately and running the SAS program separately for both subcohorts (males vs females) to produce the real-world evidence in safety studies. The variables used for incidence density sampling from the data file are listed below in detail.



Required variables for the macro:

Required variable name	Definition
Pat_ID	The unique identifier for the patient/participant included in the study
Index_Date	The start date for the beginning of the follow-up period for the Pat_ID
Index_Age	The individual (Pat_ID) age (in years) at their Index_Date
Censor_Date	The last date of follow up for the Pat_ID
Censor_Age	The individual (Pat_ID) age (in years) at their Censor_Date
Censor	Study disease flag (0 = no experience/non-case, 1 = experienced/case)

Potential additional variables for the macro:

Variable Name	Definition/When is it required
DOB	Date of birth, required if AgeRange parameter is specified
[other matching vars]	Additional matching variables, required if specified in ExactVar or RangeVar

Macro generated variables include:

Variable Name	Definition
Match_ID	The unique identifier for a case and its matched controls (the do loop count)
CaseCont	A flag indicating if the row is a control (0) or case (1)

Important notes for the macro:

1. This macro deletes tables in the work library: tmp_, ToAppend_, Cases_, Match_;
2. This macro creates or replaces tables in the work library starting with: tmp_, ToAppend_, Cases_, Match_, _IDSmacro, MatchFmt, or MatchQC_sum
3. This macro generates (and will replace any) formats called: casecont (0 = control, 1 = case), MatchFmt (for the values of 0 to nControl, if nControl parameter > 0)
4. Spaces (instead of commas) should be placed between the desired ExactVar and RangeVar parameter lists (alternatively these parameters can be "missing" or blank)
5. This macro will replace any global title and footnote statements

```
%macro IDSmatch (
  InTodata= /* Input dataset that contains required variables for matching */
  AgeRange= /* Range +/- x years from the cases start age, can be missing */
  ExactVar= /* Variable names for matching on exact same value as the case */
  RangeVar= /* Variable name and acceptable range for matching to case */
  nControl= /* Max number of controls per case. For all matches, specify 0 */
  RandSeed= /* Specify the seed for generating same random numbers */
);

* ===== *;
* ===== MACRO SET-UP ===== *;
* ===== *;
/* STEP 1: set up quotations macro for use later */
%let quote = %str(%" %");

/* STEP 2: set up case control format for outputs */
proc format;
  value casecont (default = 10)
    0 = "Control"
    1 = "Case";
run;
```




```

/* STEP 3: find number of rows in input dataset for reporting section */
data _IDSmacro;
    set &InTodata. end=last;
    if last then call symputx('nTotRow',_n_);
run;

/* STEP 4: preserve formats from InTodata dataset for the final output */
proc sql noprint;
    select strip(name) || " " || strip(format)
           , strip(name) || "=" || ifc(type = "num","0;","&quote.;")
    into :macroform separated by " "
           , :macrovars separated by " "
    from dictionary.columns
    where libname="WORK"
           and memname="_IDSMACRO"
           and name in ('pat_id' 'index_date' 'censor_date' 'censor');
quit;

/* STEP 5: design an empty dataset for QC of matched controls to cases */
data MatchQC;
    format MatchID NumMatch 8.;
    MatchID = 0; NumMatch = 0;
    stop;
run;

/* STEP 6: design an empty dataset for all matched cases and controls */
data Matched;
    format &macroform. MatchID 8. CaseCont casecont.;
    &macrovars. MatchID = 0; CaseCont = 0;
    stop;
run;

* ===== *;
* ===== Phase I: Quality Control (QC) Checks ===== *;
* ===== *;

/* STEP 1: Check for commas in ExactVar and RangeVar */
%if %sysfunc(indexc(&ExactVar.,%str(,))) > 0 %then %do;
    %put ERROR: Commas exist in the ExactVar parameter, please replace commas
        with spaces. ExactVar = &ExactVar.;
    %abort;
%end;

/* STEP 2: Check for missing or implausible Censor/ExactVar/RangeVar variables */
%do x=1 %to %sysfunc(countw(Censor &ExactVar.));
    %let QC_var = %scan(Censor &ExactVar.,&x.);
    proc means data=_IDSmacro nmiss nway noprint;
        var &QC_var.;
        output N=n Min=min Max=max out=tmp_&QC_var. (drop=_TYPE_);
    run;

    proc sql noprint;
        select _FREQ_, n, min, max

```



```

into :freq, :n, :min, :max
from tmp_&QC_var.;
quit;

/* QC checks if variable = censor */
%if &x. = 1 %then %do;
  %if %eval(&freq.) ne %eval(&n.) %then %do;
    %put ERROR: Variable &QC_var. has missing values, please investigate;
    %abort;
  %end; /* end freq censor QC check loop */

  %if &min ne 0 %then %do;
    %put ERROR: Censor variable exceeds the minimum value of 0 (non-case),
      please investigate;
    %abort;
  %end; /* end min censor QC check loop */

  %if &max ne 1 %then %do;
    %put ERROR: Censor variable exceeds the maximum value of 1 (case),
      please investigate;
    %abort;
  %end; /* end max censor QC check loop */
%end; /* end censor QC check loop */

/* QC checks if variable is from ExactVar */
%else %if &x. > 1 %then %do;
  %if %eval(&freq.) ne %eval(&n.) %then
    %put WARNING: Variable &QC_var. has missing values, please confirm;
  %if &min. < 0 %then
    %put WARNING: Variable &QC_var. has values that exceed 0, please confirm;
  %if &max. > 10 %then
    %put WARNING: High amount of levels %sysfunc(strip(&max.)) for exact
      matching variable &QC_var.;
  %end; /* end ExactVar QC check loop */
%end; /* end censor/ExactVar QC check loop */

/* QC checks if variable is from RangeVar */
%if &RangeVar. ne %then %do;
  %do x=1 %to %sysfunc(countw(&RangeVar.)) %by 2;
    %let QC_var = %scan(&RangeVar., &x.);
    proc means data=_IDSmacro nmiss nway noprint;
      var &QC_var.;
      output N=n Min=min Max=max out=tmp_&QC_var. (drop=_TYPE_);
    run;

    proc sql noprint;
      select _FREQ_, n, min, max
      into :freq, :n, :min, :max
      from tmp_&QC_var.;
    quit;

```

```

%if %eval(&freq.) ne %eval(&n.) %then
  %put WARNING: Variable &QC_var. has missing values, please confirm;
%if &min. < 0 %then
  %put WARNING: Variable &QC_var. has values that exceed 0, please confirm;
%if &max. > 10 %then
  %put WARNING: High amount of levels (%sysfunc(strip(&max.))) for range
    matching variable &QC_var.;
%end; /* end count RangeVar */
%end; /* end RangeVar loop */

/* STEP 3: report duplicate participant identifiers */
proc sql noprint;
  select count(distinct pat_id), count(*) into :pat, :row from _IDSmacro;
quit;

%if %eval(&pat.) ne %eval(&row.) %then %do;
  %put "ERROR: Duplicates exist in &InTodata.. dataset &pat. patients out of
    &row. rows";
  %abort;
%end;

* ===== *;
* ===== Phase II: Matching loop through all cases ===== *;
* ===== *;
/* STEP 1: create unique matchID in cases dataset and calculate number of cases */
data Cases;
  set _IDSmacro (where=(censor = 1)) end=last;
  format MatchID 8.;
  MatchID = _n_;
  if last then call symputx('ncases',MatchID);
run;
%put NOTE: Number of cases in &InTodata. dataset = &ncases.;

/* STEP 2: Matching loop assigns local macros of case values to match controls */
%do i=1 %to &ncases.;
  proc transpose data=Cases (where=(MatchID = &i.))
    out=Cases_t&i. (rename=COL1=Value)
    name=Variable;
  run;

  /* STEP 2a: Exact matching criteria for local macros */
  %let eVars = %sysfunc(countw(pat_id index_age index_date censor_age
    censor_date &ExactVar.));

  %let Exact = ;
  %do vars = 1 %to &eVars.;
    %let variable = %sysfunc(scan(pat_id index_age index_date censor_age
      censor_date &ExactVar.,&vars));

    data _null_;
      set Cases_t&i. (where=(variable = "&variable."));
      call symputx("&variable.",Value);
    run;
  
```

```
%if &vars. > %sysfunc(countw(pat_id index_age index_date censor_age
                                censor_date)) %then %do;
    %let Exact = &Exact. if &variable. eq &&&variable. %str( );
%end; /* end exact if statement appending */
%end; /* end ExactVar matching loop */

/* STEP 2b: Ranges matching criteria for local macros */
%if &RangeVar. ne %then %do;
    %let rVars = %sysfunc(countw(&RangeVar.));
    %let Range = ;
    %do rans = 1 %to &rVars. %by 2;
        %let variable = %sysfunc(scan(&RangeVar.,&rans.));
        %let VarRange = %sysfunc(scan(&RangeVar.,&rans. + 1));
        data _null_;
            set Cases_t&i. (where=(variable = "&variable."));
            call symputx("&variable.",Value);
            call symputx("Ran",&VarRange.);
        run;
        %let Range = &Range. if (&&&variable. - &Ran.) < &variable. <
                                (&&&variable. + &Ran.) %str( );
    %end; /* end range if statement appending */
%end; /* end RangeVar matching loop */

/* STEP 2c: find all matching controls and assign a random seed number */
data Match_&i.;
    set _IDSmacro (where=(pat_id ne &pat_id. /* case cannot match to itself */
                        and index_date <= &index_date. /* control must follow up for */
                        and censor_date >= &censor_date. /* = or greater than case*/));
    format MatchID 8.;

/* controls assigned MatchID becomes the same as the case */
MatchID = &i.;

/* Any additional exact variables to match on */
if "&ExactVar." ne "" then do;
    &Exact.;
end;

/* Matching criteria that may have ranges relative to the case */
%if &AgeRange. ne %then %do;
    if (&index_age. - &AgeRange.) < intck('YEAR',DOB,&index_date.) <
        (&index_age. + &AgeRange.);
    if (&censor_age. - &AgeRange.) < intck('YEAR',DOB,&censor_date.) <
        (&censor_age. + &AgeRange.);
%end;

/* any additional range variables to match on */
%if &RangeVar. ne %then %do;
    &Range.;
%end;
```



```

/* Set random number for randomised 1:n matching */
RandNumb = uniform(&RandSeed.);
run;

/* STEP 2d: sort controls that match by RandNumb */
proc sort data=Match_&i. out=Match_&i.s;
  by RandNumb;
run;

/* STEP 2e: append case and matched controls into dataset and match count */
data Match_All&i. %if &ncontrol. > 0 %then %do;
  (where=(OrderRow <= &ncontrol.)); %end;
  set Cases      (in=Cas keep=pat_id index_date censor_date censor MatchID
                  where=(MatchID = &i.))
              Match_&i.s (in=Con keep=pat_id Index_Date Censor_Date Censor MatchID
                          RandNumb) end=last;
  format CaseCont casecont.;

  if Cas then do;
    OrderRow = 0;
    CaseCont = 1;
  end;
  if Con then do;
    OrderRow = _n_ - 1;
    CaseCont = 0;
  end;
  if last then call symputx('nMatches',OrderRow);
run;

data ToAppend_&i.;
  format MatchID NumMatch 8.;
  MatchID   = &i.;
  NumMatch  = &nMatches.;
run;

/* STEP 2f: Append to master tables */
proc append base=MatchQC data=ToAppend_&i.; run;
%if &nMatches. > 0 %then %do;
  proc append base=Matched data=Match_All&i. (drop=RandNumb OrderRow); run;
%end; /* end match append */
%end; /* end cases loop */

* ===== *;
* ===== Phase III: Reporting ===== *;
* ===== *;
/* STEP 1: create MatchFmt if nControl is specified for the levels of nControl */
%if &nControl. > 0 %then %do;
  title "Incidence density sampling macro report: frequency of 1:n matches";

  /* STEP 1a: create table of all possible categories */

```

```

data MatchFmt;
  format Label $15.;
  FmtName = "MatchFmt";
  do Start = 0 to (&nControl.-1);
    End = Start;
    Label = strip(Start);
    output;
  end;
  Start = &nControl.;
  End = &nTotRow.;
  Label = strip("&nControl. +");
  output;
run;

/* STEP 1b: put categories into a format */
proc format cntlin=MatchFmt; run;

/* STEP 1c: frequency of all possible combinations (using preloadfmt) */
proc tabulate data=MatchQC;
  class NumMatch / preloadfmt;
  format NumMatch MatchFmt.;
  table NumMatch="Number of Matches per Case" ALL="TOTAL", N="n" PctN="%" /
    printmiss misstext="0";
run;
%end; /* end if nControl is > 0 */

/* STEP 2: local macro for number of cases without matched controls */
proc sql noprint;
  select count(*) into :MisMatch from MatchQC where NumMatch = 0;
quit;

title "Incidence density sampling macro report: summary statistics for the
      number of matches per case";
footnote1 "Total number of cases in &InToData.: &ncases.";
footnote2 "Total number of rows in &InToData.: &nTotRow.";
footnote3 "Number of cases with no matches: %sysfunc(strip(&MisMatch.))";

/* STEP 3: summary statistics for reporting the matching of controls to cases */
proc means data=MatchQC noprint;
  var NumMatch;
  output N=n Mean=mean STD=SD Min=min p25=p25 median=medi p75=p75 Max=max
    out=MatchQC_sum;
run;

/* STEP 4: output */
proc print data=MatchQC_sum (drop=_TYPE_ _FREQ_) noobs label;
  label n="Number of Cases"
  mean = "Average number of matches per case"
  SD = "Standard deviation for the number of matches per case"
  Min = "Minimum number of matches per case"
  p25 = "25th percentile for number of matches per case"

```



```

medi    ="Median number of matches per case"
p75     ="75th percentile for number of matches per case"
max      ="Maximum number of matches per case";
run;

/* STEP 5: clean up titles, footnotes and helper tables */
title;
footnote;
proc datasets noprint;
    delete _IDSmacro MatchFmt MatchQC_sum ToAppend_ Cases_ Match_ tmp_;;
run;
%mend IDSmatch;

```

CONCLUSION

Incidence Density Sampling refers to sampling controls from among the set of disease-free individuals matched with time at the time of case incidence. Time-matched analysis of case-control data with density sampling yields estimates of the relative risk (incidence rate ratio) in the source population. In this paper, we have clearly outlined the main rationale for using incidence density sampling for nested case-control studies, which is the recommended approach for safety studies. We have also provided a novel and simple SAS program for incidence density sampling, which could minimise selection bias in the study and applies a more appropriate way of optimising the selection of controls to cases. The paper summaries how the data might be analysed using SAS in a nested case-control study and the results obtained from the provided SAS program suggests that the incidence density sampling is the most suitable method and has significant strengths associated with this to fulfill study aims with respect to the dataset selected while accounting for study outcomes and endpoints. The incidence density sampling method for nested case-control analyses is a useful tool for epidemiological research⁸. The presented SAS program provides unbiased and objective results, together with a flexible approach, which can be easily adapted to suit a wide variety of study settings.

REFERENCES

1. Beaumont, J. J., Steenland, K., Minton, A., & Meyer, S. (1989). A Computer Program for Incidence Density Sampling of Controls in Case-Control Studies Nested within Occupational Cohort Studies. *American Journal of epidemiology*, Vol. 129, No. 1.
2. Gately, R., & Turnbull, B. (2009). Incidence-Density Sampling: Alternative Programming Approaches to Achieve the Same End. *NESUG 2009 - Pharmaceuticals, Health Care, and Life Sciences*.
3. Kiri, V. A. (2012). Programming Challenges of Sampling Controls to Cases from the Dynamic Risk Sets in Nested Case Control Studies. *PhUSE*.
4. Matthews, R., & Brill, I. (n.d.). SAS ® Programs to Select Controls for Matched Case-Control Studies. *SUGI 30*.
5. Pearce, N. (1989). Incidence Density Matching with a Simple SAS Computer Program. *International Journal of Epidemiology*, 981-984.
6. Pearce, N. (2016). Analysis of matched case-control studies. *British Medical Journal (BMJ)*. doi: <https://doi.org/10.1136/bmj.i969>
7. Prentice, R. I. (June 1986). On the Design of Synthetic Case-Control Studies. *Biometrics*, 42, 301 - 310 .
8. Richardson, D. B. (2004). An incidence density sampling program for nested case-control analyses. *Occup Environ Med*. doi:10.1136/oem.2004.014472
9. Robins, J. M., Gail, M. H., & Lubin, J. H. (Jun., 1986). More on "Biased Selection of Controls for Case-Control Analyses of Cohort Studies". *International Biometric Society*, 42, No. 2, 293 - 299. Retrieved from <http://links.jstor.org/sici?sici=0006-341X%28198606%2942%3A2%3C293%3AMO%22SOC%3E2.0.CO%3B2-C>
10. Steyerberg, E. W. (2019). *Clinical Prediction Models*. Leiden University Medical Center, Leiden, The Netherlands: Springer Nature Switzerland AG 2019.
11. Vandenbroucke, J. P., & Pearce, N. (2012). Case-control studies: basic concepts. *International Journal of Epidemiology*, 1480-1489.
12. Vandenbroucke, J. P., & Pearce, N. (2012). Incidence rates in dynamic populations. *International Journal of Epidemiology*, 1472-1479.
13. Wang, M.-H., Shugart, Y. Y., Cole, S. R., & Platz, E. A. (July 24, 2019). A Simulation Study of Control Sampling Methods for Nested Case-Control Studies of Genetic and Molecular Biomarkers and Prostate



- Cancer Progressio. *American Association for Cancer Research.*, Cancer Epidemiol Biomarkers Prev 2009;18:706–11. Retrieved from <https://cebp.aacrjournals.org/>
14. Hennekens CH, Buring JE. *Epidemiology in Medicine*, Lippincott Williams & Wilkins, 1987.

ACKNOWLEDGMENTS

For their support and help with funding, the authors would like to thank IQVIA company.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact us at:

Author:	Quratul Ann & Rachel Tham
Company:	IQVIA
Address:	210 Pentonville Road London, N1 9JY, UK
Work Phone:	+44 7833 402 763
Email:	Quratul.Ann@iqvia.com & rachel.tham@iqvia.com
Web:	linkedin.com/in/quratulann & linkedin.com/in/rachel-tham-76046118