Paper 032-2007

A SAS® Macro for Automating the 'Change-In-Estimate' Strategy for Assessing Confounding

Julius Atashili and Myduc Ta
Department of Epidemiology, University of North Carolina at Chapel Hill,
Chapel Hill, NC

ABSTRACT

Although the "change-in-estimate" (CIE) strategy is conceptually more accurate than the "significance-test-of-the covariate" (STC) strategy for assessing confounding in the causal modeling of observational epidemiological data, the use of the CIE strategy is hampered by the cumbersome need for analysts to run an enormous amount of models. Indeed, assessing n covariates for potential confounding using the CIE strategy requires ((n)*(n+1)/2)+1) models. With 10 potential confounders, this translates to as many as 56 models. Automating the CIE strategy in SAS® will allow for a better assessment of confounding using this method.

This paper introduces %backconf, a SAS® macro that automatically implements the CIE strategy using a backward elimination of covariates resulting in the least change of the estimate of the exposure variable. %Backconf incorporates the GENMOD procedure and easily accommodates different values allowed for by the *class*, *where*, *link*, *dist*, *offset* and *weight* statements. It can thus be used for Linear, Binomial, Logistic, or Poisson regression modeling amongst others. Furthermore, unlike existing software for implementing this approach, %backconf accommodates groups of variables to be considered as one entity (for example a variable and its higher quadratic or cubic terms) and covariates (such as matching variables) to be continually adjusted for in the model. Results of each step of the elimination of variables are summarized in a user-friendly and easily interpretable table at the end of the output. After reading this paper, SAS® users should be able to readily apply this macro to facilitate their analysis.

INTRODUCTION

Despite the multitude of definitions of confounding in epidemiology, most authors concur on the prerequisite for a covariate to be associated with both the outcome (dependent variable) and the main exposure variable under study for that variable to be a confounder. One of the commonly used method of assessing confounders, the "significance-test-of-the covariate" (STC) strategy [1] is readily available in most software. However, this method may be inaccurate because the p-value of the covariate is solely a reflection of the association between the covariate and the outcome (or disease) [2,3]. The "change-in-estimate" (CIE) strategy compares the estimate of exposure-outcome association in a model adjusting for a covariate to that same estimate in a model that excludes that specific covariate [1]. This strategy has been advocated as being more accurate as it accounts for both covariate-outcome and covariate-exposure association [2,3]. Despite this recognized accuracy, the use of the CIE strategy is hampered by the cumbersome need for analysts to run an enormous amount of models. Indeed, assessing n covariates for potential confounding using the CIE strategy requires ((n)*(n+1)/2)+1) models. With 10 potential confounders, this translates to as many as 56 models. Furthermore, the CIE strategy is not currently automated in most software commonly used for statistical analysis of epidemiological data. Automating the CIE strategy in SAS® will allow for a better assessment of confounding using this method.

THE %BACKCONF MACRO

The detailed code for the macro is included in the Appendix. This macro uses PROC GENMOD to compute change in the estimate for the exposure-outcome relationship to assess confounding using a backward elimination approach. PROC GENMOD can estimate the epidemiologic measures: risk ratio (RR), incidence rate ratio (IRR), risk difference (RD), and odds ratio (OR). The user specifies the appropriate link and distribution functions to estimate the epidemiologic parameter of interest.

PROC GENMOD estimates the exposure-outcome relationship from the full model containing all covariates. The macro then automates a model building process using backward deletion, dropping the covariate that resulted in the smallest change in estimate at each of the PROC GENMOD iterations. Output includes results from all the models ran and for each of the iterations, the variable that was dropped, the adjusted exposure-outcome estimates and their corresponding confidence intervals (CI), and the change in the estimate for the exposure-outcome relationship.

USER INPUT

The syntax to invoke the macro is as follows

%backconf(data= , yvar= , exp= , covars= , catvars= , grpcovars1= , grpcovars2= , matchvars= , link= , dist= , where= , offset= , weight=);

Where the arguments are as follows:

User Specified Parameter	Parameter Type	Description
data	Required	Specifies the input data set. Can be temporary or permanent data set. If permanent data set library reference should be included along with the data set name. For example: inlib.dataset_name
yvar	Required	Specifies the dependent (outcome) variable for the model.
exp	Required	Specifies the primary exposure variable for the model.
covars	Required	Specifies covariates for the model. Include only non-grouped, and non-categorical variables (exclude exposure variable)
catvar	Optional	Used for specification of non-binary categorical variables.
grpcovars1	Optional	Specifies a first set of grouped covariates for the model. Example: a group of indicators listed as "ind1 ind2 ind3". Leave blank if none.
grpcovars2	Optional	Specifies a second set of grouped covariates for the model. Example: a group of covariate and its higher orders listed as " x xsq xcub ". Leave blank if none.
matchvars	Optional	Variables to always be adjusted for, yet not assessed as confounders. Example: a matching variable.
link	Required	Specifies the link function for PROC GENMOD. Default is logit. Options include: logit, log, identity, cumlogit, probit, etc. Refer to the SAS documentation for complete list.
dist	Required	Specifies the distribution of the errors for PROC GENMOD. Default is binomial. Options include: normal, Poisson, negbin, etc. Refer to SAS documentation for complete list.
where	Optional	Used to subset input data into strata of effect measure modifier to assess confounding within each strata of the effect measure modifier. THE MACRO DOES NOT INCORPORATE INTERACTION TERMS. User inputs the subsetting condition. For example: varname=1. Leave blank if none.
offset	Optional	Specifies offset for PROC GENMOD. Leave blank if none. Default is value of zero. Useful for Poisson models.
weight	Optional	Specifies weight for PROC GENMOD. Leave blank if none. Default is value of 1.

HOW IT WORKS

The macro successively:

1. Defines variables and parameters.
2. Defines pseudocovariates that are surrogates for grouped covariates and considered as one entity. These pseudocovariates will be used when the grouped variables need to be considered as one, while the original covariates will be considered ungrouped in executing models.
3. Counts covariates entered by user, using the SAS® Word macro.
4. Runs a full model with all covariates. This is labeled Cycle 0. The estimate for exposure variable is output to a data set. This is the fully adjusted estimate.
5. Runs a first cycle of reduced models.
    a. Drops one potential confounder, runs a model, outputs the beta estimate for the exposure variable.
    b. Drops another potential confounder (after replacing the previous one), runs another model and outputs the estimate for the exposure variable.
    c. Repeats until all potential confounders are successively dropped.
    d. Assesses which potential confounder (when dropped) resulted in the least change in estimate.
    e. Permanently drops that potential confounder and outputs a data set that identifies it and the corresponding estimate for the exposure variable (when the potential confounder was dropped).
    f. Proceeds with the remaining potential confounders to the next cycle

6. Runs a second cycle of reduced models. Repeats steps as for first cycle above.
7. Runs cycles until all potential confounders have been dropped from the model. Repeats steps as for first cycle above.
8. Regroups (concatenates) data sets from each cycle and reports the variable dropped at each cycle as well as the estimate for the exposure variable.

SPECIAL CONSIDERATIONS

Variable names: The names of covariates specified for "covars" must be unique: no variable name should include the name of another variable. For example listing "matage age" in "covars" will result in a bug in the macro as "age" is included "matage". Simply rename "age" in your data set to circumvent this bug. The variable names can share the same letters without any problem, so long as none of them is completely included in the other. For example "matage dage" will work just fine.

Interaction: This macro does not incorporate interaction terms. To assess confounding within strata of a categorical effect modifier, use the "where" option to subset the data set to each strata of the effect modifier.

Grouped Covariates: A maximum of 2 sets of grouped covariates can be accommodated.
As with every other modeling process, verify the log and model outputs to make sure the macro is modeling the desired outcome (example 1 versus 0) and that convergence was achieved for each of the models.

MACRO OUTPUT

In addition to generating output for each of the models, the macro also generates a summary of the variables dropped, the adjusted estimates and corresponding confidence intervals at each step, and the change in the estimate for the exposure variable.
A description of the summary output generated for each modeling cycle is as follows:

Output Parameter	Description
CYCLE	Iteration cycle; cycle zero is the fully adjusted model
DROPPED	Name of variable that was dropped from model
[epidmeasure]	Model estimated epidemiologic measure of association for exposure-outcome relationship (e.g. risk difference [RD], risk ratio[RR], incidence rate ratio[IRR], odds ratio [OR])
L95CL[epidmeasure]	Lower 95% confidence interval for corresponding epidemiologic measure of association.
U95CL[epidmeasure]	Upper 95% confidence interval for corresponding epidemiologic measure of association.
CLR ; CLD	These are measures of precision. Confidence Limit Ratio: ratio of upper to lower confidence limits for ratio measure of association or Confidence Limit Difference: difference between upper and lower confidence limits for a difference measure.
PCTDIFF[epidmeasure]	Percentage change in the model estimated epidemiologic measure of association for the exposure-outcome relationship. Computed as 100* (indexmodel[epidmeasure] – fully adjusted[epidmeasure])/fully adjusted[epidmeasure].
LNCO[epidmeasure]; CORD	The log of the confounding [epidmeasure]. This is a relatively novel measure that accounts for the exponential scale of ratio epidemiological effect measures (RR, OR, IRR). This measure linearizes the assessment of the percentage change in estimate. Lnco[epidmeasure] is computed as |ln(indexmodel[epidmeasure] / fully adjusted[epidmeasure])|. The confounding risk difference (CoRD) is the equivalent measure for the difference measures of effect. These are already on the linear scale and thus do not need any logarithmic transformation.

The remainder of the output includes detailed results from each model generated by PROC GENMOD and a listing of the following parameters:

Output Parameter	Description
COVARSINMODEL	Covariates left in each model
CYCLE	Iteration cycle; cycle zero is fully adjusted model
DROPPED	Name of variable that was dropped from model
ESTIMATE	Model estimate for exposure-outcome relationship
FULADJ	Estimate for exposure-outcome relationship from fully adjusted model
PARAMETER	Identifies the exposure variable
PDIF	Change in model estimate for exposure-outcome relationship
PDIFABS	Absolute value of the change in estimate
STDERR	Standard error of model estimate for exposure-outcome relationship

CONCLUSION

Assessment of confounding is an important aspect of data analysis. The %Backconf macro described in this paper automates the iterative process of confounding assessment based on a change in estimate strategy using a backwards elimination approach. By employing the GENMOD procedure the macro easily accommodates a variety of common epidemiologic measures of association. The %Backconf macro provides an efficient means by which this method of assessing confounding can be implemented. SAS® users should be able to readily apply this macro to facilitate their analysis.

ACKNOWLEDGEMENTS

We thank Dr Jay Kaufman, Elizabeth Torrone and Caroline Hoffman for their helpful suggestions.

REFERENCES

1. Maldonado G, Greenland S. Simulation study of confounder-selection strategies. Am *J Epidemiol.* 1993;138(11):923-36.

2. Rothman K J, Greenland S. Modern Epidemiology 1998;1:256-257.

3. Sonis J. A closer look at confounding. *Fam Med. 1998*;30(8):584-8.

APPENDIX

```
%macro backconf(data= , yvar= , exp= , covars=, grpcovars1=, grpcovars2=, matchvars=,
catvars=, link=, dist= , where= , offset=, weight= );
 options  nodate;

/*creating pseudogrpcovars to represent the grouped vars  and be counted and dropped as
one entity*/

data &data; set &data; pseudogrpcovars1=.; pseudogrpcovars2=.; run;

%let pseudogrpcovars1=pseudogrpcovars1 ; %if &grpcovars1= %then %let pseudogrpcovars1= ;
%let pseudogrpcovars2=pseudogrpcovars2;  %if &grpcovars2= %then %let pseudogrpcovars2= ;


/*redefining covars to include grouped and categorical covars*/
%let covars = &covars &pseudogrpcovars1 &pseudogrpcovars2 &catvars;

/*count the number of covariates*/
%let numcovars= %words(&covars);

/*set defaults for required parameters*/
%if &link=  %then %let link= logit;
%if &dist=  %then %let dist= binomial;

/*including a variable zero, used to set a blank offset to a default value of 0, a
variable one, used to set a default weight of 1*/
data &data; set &data; zero=0; one=1; run;
%if &offset=  %then %let offset=zero;
%if &weight=  %then %let weight=one;


/*to change name of pseudogrpcovar to grpcovar*/

/* converting macro variable list to a value of a variable in a data set*/
       Data A;
              length covarname $500;
              length aftermodel $500;
              covarname=symget('covars');
              covarname=tranwrd(covarname, "pseudogrpcovars1", "&grpcovars1");
              covarname=tranwrd(covarname, "pseudogrpcovars2", "&grpcovars2");
              aftermodel=symget('covars');
              run;

       /*convert variable value to macrovariable*/
       Data A;
              set A;
              call symput('covars', covarname);
              call symput('aftermodel', aftermodel);
              run;


/* runs full model and outputs parameters to data set*/

proc genmod data=&data descending;
       ods output ParameterEstimates=parm (keep=parameter Estimate Stderr
where=(parameter="&exp"));
       class &catvars;
       model &yvar = &exp &covars &matchvars /link=&link dist=&dist offset=&offset;
              title2 "Full Model: Covars = &covars";
              where &where;
              weight &weight;
              run;

/*fully adjusted data set - for use later in creating final listing*/
data dropped0;
```

```
	set parm;
	covarsinmodel="&covars";
	fuladj=estimate;
	cycle=0;
	run;


/*switch back to covar with pseudgrpcovar rather than grpcovar*/
	%let covars=&aftermodel;

/*to iterate 1st level drops*/
%let cycle=1;
%do %until (&cycle=(&numcovars+1));

%if &cycle ne 1 %then %let covars=&nextcovars;

%let count=1;

%do %until (%qscan(&covars, &count)=);

%let x=%qscan(&covars, &count);


%let dropped=%scan(&covars, &count);

/* converting macro var list to a value of a variable in a data set*/
/* and deleting the dropped variable from 'list'*/
Data todrop&count;
	length covarname $500;
	length covarlist $500;
	covarname=symget('covars');
	covarlist = tranwrd(covarname, "&dropped", " ");
	run;

/*convert variable value to macrovariable*/
Data todrop&count;
	set todrop&count;
	call symput('rcovars', covarlist);
	run;


/*to change name of pseudogrpcovar to grpcovar*/
/* converting macro variable list to a value of a variable in a data set*/

	Data B;
		length covarname $500;
		length aftermodel $500;
		covarname=symget('rcovars');
		covarname=tranwrd(covarname, "pseudogrpcovars1", "&grpcovars1");
		covarname=tranwrd(covarname, "pseudogrpcovars2", "&grpcovars2");
		aftermodel=symget('rcovars');
		run;

	Data B;
		set B;
		call symput('rcovars', covarname);
		call symput('aftermodel', aftermodel);
		run;


proc genmod data=&data descending;
		ods output ParameterEstimates=parm&count
		(keep=parameter Estimate Stderr where=(parameter="&exp"));
		class &catvars;
model &yvar = &exp &rcovars &matchvars/link=&link dist=&dist offset=&offset;
		title2 "cycle &cycle :reduced model, dropped &dropped";
		title3 "covars in model = &rcovars";
```

```
              where &where;
              weight &weight;
              run;

/*switch back to rcovar with pseudgrpcovar rather than grpcovar*/
       %let rcovars=&aftermodel;



/* adds variable to identify which variable was dropped from model */
/* keeping track of which variable was dropped and which remained in model*/
data parm&count;
       set parm&count;
       length dropped $20.;
       dropped="&dropped";
       length covarsinmodel $500.;
       covarsinmodel="&rcovars";
       run;


/** concatenates data sets with parameters of interest from each iteration of GENMOD **/

%if &count=1 %then %do;

/*concatenate estimate from full model and 1st var dropped*/

data parms;
       length dropped $20.;
       set parm parm1;
       run;
       %end;

%else %if &count >1 %then %do;

/*concatenate estimates from data with each of other vars dropped*/

data parms;
       length dropped $20.;
       set parms parm&count;
       run;
       %end;
%let count=%eval(&count+1);

%end;

/* keeps fuladj (fully adjusted estmate) down for all observations*/
data parmsadj;
       set parms;
       retain fuladj;
       if _N_ =1 then fuladj=estimate;
       run;

/*computing change in estimate*/
data parmsdif;
       set parmsadj;
       pdif=(estimate-fuladj);
       pdifabs=abs(pdif);
       run;

/* sorts by ascending order of value of absolute difference between value Full and &
Adjusted Estimate */

proc sort data=parmsdif;
       by pdifabs;
       run;

/* creates 2 data sets: DROPPED and KEPT */
```

```
/* outputs variable with least difference to dropped*/

data dropped&cycle kept&cycle;
       set parmsdif;
       cycle=&cycle;
       if _N_ = 2 then output dropped&cycle;
       else output kept&cycle;
       run;


/** to create a new macrolist of covariates (for next cycle of models) with variable
names which are the value of 'covarsinmodel' variable */
/*convert variable value to macrovariable*/

Data _Null_;
       set dropped&cycle;
       call symput('nextcovars', covarsinmodel);
       run;


/*to be specific on relative risk measure ie to distinguish OR from risk ratios RR,
Incidence rate ratio (IRR) and RD*/

%let rr=OR; /*rr is generic term: relative risk*/
%if (%upcase(&link))=LOG & (%upcase(&dist))= BINOMIAL %then %let rr= RR;
%else %if (%upcase(&link))=LOG & (%upcase(&dist))= POISSON %then %let rr= IRR;
%else %if (%upcase(&link))=IDENTITY & (%upcase(&dist))= BINOMIAL %then %let rr= RD;


*concatenating dropped data sets;

%if &cycle=1 %then %do;
/*concatenate estimate from full model and 1st lowest conf dropped*/

data alldropped;
       set dropped0 dropped1;
       run;
       %end;

%else %if &cycle >1 %then %do;

/*concatenate estimate from subsequent dropped data sets*/

%if (%upcase(&link))=IDENTITY & (%upcase(&dist))= BINOMIAL %then %do;
       data alldropped;
              set alldropped dropped&cycle;
              &rr= (estimate);
              l95cl&rr=(estimate-(1.96*stderr));
              u95cl&rr=(estimate+(1.96*stderr));
              pctdiff&rr=100*(&rr-((fuladj)))/(abs(fuladj));
              co&rr = abs(&rr -fuladj);
              cld=u95cl&rr - l95cl&rr;
              if compress(dropped)="pseudogrpcovars1" then dropped="&grpcovars1";
              if compress(dropped)="pseudogrpcovars2" then dropped="&grpcovars2";
              run;
              %end;
       %else %do;
       data alldropped;
              set alldropped dropped&cycle;
              &rr=exp(estimate);
              l95cl&rr=exp(estimate-(1.96*stderr));
              u95cl&rr=exp(estimate+(1.96*stderr));
              pctdiff&rr=100*(&rr-(exp(fuladj)))/(exp(fuladj));
              lnco&rr = abs(log(&rr/(exp(fuladj))));
              clr = u95cl&rr / l95cl&rr;
              if compress(dropped)="pseudogrpcovars1" then dropped="&grpcovars1";
              if compress(dropped)="pseudogrpcovars2" then dropped="&grpcovars2";
```

```
              run;
              %end;
       %end;


%let cycle=%eval(&cycle+1);

%end;

proc print data=alldropped noobs;
       title; title1;title2;
       run;

       %if (%upcase(&link))=IDENTITY & (%upcase(&dist))= BINOMIAL %then %do;
       proc print data=alldropped noobs;
       var cycle dropped estimate stderr pdif pdifabs;
       title ; title1 "Summary results"; title2;
       format _NUMERIC_ 8.4 cycle 8. cld 8.2;
       run;

       proc print data=alldropped noobs;
       var cycle dropped &rr l95cl&rr u95cl&rr cld pctdiff&rr co&rr;
       title ; title1 "Summary results &rr scale"; title2;
       format _NUMERIC_ 8.4 cycle 8. cld 8.2;
       run;
       %end;

       %else %do;
       proc print data=alldropped noobs;
       var cycle dropped estimate stderr pdif pdifabs;
       title ; title1 "Summary results "; title2;
       format _NUMERIC_ 8.4 cycle 8. cld 8.2;
       run;

       proc print data=alldropped noobs;
       var cycle dropped &rr l95cl&rr u95cl&rr clr pctdiff&rr lnco&rr;
       title ; title1 "Summary results -&rr scale"; title2;
       format _NUMERIC_ 8.4 cycle 8. cld 8.2;
       run;

       %end;

%mend backconf;



/*The macro 'WORD' below came from the SAS technical support website
       at the address http://support.sas.com/faq/016/FAQ01617.html*/

%MACRO WORDS(string); /*to count number of covars*/

        %Local Count Word;
       %let Count=1;
       %let Word=%qscan(&string,&Count,%str( ));
       %do %while(&Word ne);
    %let Count=%eval(&Count+1);
    %let Word=%qscan(&string,&Count,%str(  ));
       %end;
       %eval(&Count-1)

%mend words;
```

CONTACT INFORMATION

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

Julius Atashili
Department of Epidemiology
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-7435
atashili@email.unc.edu