PharmaSUG 2019 - Paper AD-064

%LOGBINAUTO: A SAS 9.4 Macro for Automated Forward Selection of Log-Binomial Models

Matthew Finnemeyer, Vertex, Pharmaceuticals, Inc.;

ABSTRACT

In a study, relying on log-binomial models to generate estimates of relative risk for clinical outcomes, there was a need to select from a plethora of clinically-relevant covariates to produce parsimonious explanatory models.

While SAS 9.4 allows for automated selection (forward, back, stepwise) for some non-linear models, it does not extend this functionality to all non-linear models, including log-binomial models. As it would be very time-consuming to code this for each model, the %LOGBINAUTO macro was created to automate forward selection for log-binomial models and implemented for the study.

This macro utilizes nested %DO loops and non-sequentially manages a list of user-supplied covariates, using measures of statistical significance and improved model-fit, to select for their inclusion into a log-binomial model.

This paper will outline the code and implementation of the %LOGBINAUTO macro for SAS 9.4, designed for an audience familiar with SAS MACRO language and the PROC GENMOD procedure..

INTRODUCTION

This paper will outline and explain the use of SAS 9.4 macro code to extend the functionality of automated forward selection to log-binomial models. This macro, %LOGBINAUTO, uses PROC GENMOD (binomial distribution, log link). As of SAS 9.4/M5, PROC LOGISTIC and HPGENSELECT allow for the use of automated selection (backward, forward, stepwise) for logistic models (binomial distribution, identity link) but do not extend to log-binomial models.

In order to extend this functionality, we must replicate the key elements of SAS's automated selection process:

1) The ability to produce the selected model with individual covariates and identify their potential inclusion, based on pre-established statistical significance and model fit criteria.

2) Retain model fit criterion for each tested covariate to identify the best candidate for inclusion

3) Upon retaining a candidate covariate, update the model for each of the remaining covariates and repeat the ranking/selection process

4) Allow for a "non-terminal" stopping point for automated selection, to avoid unnecessary screening iterations.

In this paper, we will use a scenario based on published public health research[1], containing a log-binomial model with a binary outcome, a fixed categorical primary covariate, and a mix of continuous and categorical secondary covariates (clinical or demographic in nature) to be tested for inclusion. The code will be presented in linear order, separated into functional groups to explain the purpose and mechanism of the code in contributing to the four key elements established above.

USER INPUTS: MACRO PARAMETERS

The LOGBINAUTO macro call consists of nine keyword parameters; two must be defined by the user for the macro to execute, with the remaining seven defaulting to values that will permit normal macro functioning {Appendix Section A}:

The nine macro parameters are defined as follows:

1. DATA = Source SAS dataset
2. OUTCOME= Binomial Outcome Variable {SAS variable in &DATA dataset}
3. PRIMCOV= Covariate(s) to be forced into model (if any) {SAS variable(s) in &DATA dataset}
4. SECCOVLIST= All secondary covariate(s) to be tested for inclusion, {SAS variable(s) in &DATA dataset; separated by commas}
5. PERR= Type of p-value error to be used– (e.g. Type1 or Type3)
6. PTHRESH= $\chi 2$ p-value threshold for inclusion of a secondary covariate to the model
7. CLASSLIST= All ordinal/categorical secondary covariates in &SECCOVLIST {space delimited}
8. ID= Suffix for names of interim and final models' output datasets
9. PREDLEVEL= Default numeric value of 2 indicates primary covariate is binary, otherwise designed to handle comparisons of up to 3 levels of primary covariate {e.g. High—Med—Low}

%LOGBINAUTO: CODE IN ACTION

INITIALIZING LOCAL VARIABLES AND CONSTRUCTING THE &J : &I LOOPS:

Prior to the initialization of the nested &J: &I loops, %LOGBINAUTO creates five local macro variables with preset values; four of these variables {&KEPTVAR, &KEPTCLASSVAR, &CURRCLASSVAR, &LASTAIC} will be re-defined during subsequent loops. &COVLIST_NO1 is left unmodified, to allow for user QC {Appendix Section B}:

```
%local keptvar keptclassvar currclassvar;
%let lastAIC = 99999.99;
%let covlist_no1 = %qupcase(&seccovlist.);
```

The &J loop is constructed to iterate up to once per user-specified secondary covariate (e.g. Six covariates will result in &J having a maximum value of 6). Each &I loop is constructed to iterate once per covariate in the current covariate list (i.e. &&COVLIST_NO&J), with the screened covariate, per &I iteration, identified as &&_&I.VAR. If &&_&I.VAR was identified by the user as a categorical/ordinal variable {i.e. &CLASSLIST), &CURRCLASSVAR is populated with the same covariate name {Appendix Section C}:

```
%do j = 1 %to   %sysfunc(countw(&covlist_no1., %str( ) ));
%do i = 1 %to sysfunc(countw(%str(&&covlist_no&j.), %str(, ) ));
%let _&i.var = %scan(&&covlist_no&j., &i.);
%if %index(%qupcase(&classlist.), &&_&i.var) %then %let currclassvar =
  &&_&i.var;
%else %let currclassvar = %str();
```

COVARIATE SCREENING:

PROC GENMOD Log-Binomial Screening Models

In each &I iteration, %LOGBINAUTO produces a log-binomial model of the outcome variable {&OUTCOME}, "contingent" on the primary covariate {&PRIMCOV} and any previously-selected secondary covariates (&KEPTVAR). The Model Fit and Error Type results from the model are output as

individual datasets {Fit&&_&I.VAR, &PERR&&_&I.VAR}, which allow for the ranking and selection of the "best" covariate for the model {Appendix Section D}:

```
title "Step &j.";
title2 "Covariate #&i.: &&_&i.var.";
proc genmod data = &data.  ;
  class &outcome.  &primcov. (Descending)  &keptclassvar. &currclassvar.;
  model &outcome. (Event = "Yes") = &primcov. &keptvar. &&_&i.var.
                                                / dist = bin link = log perr.;
  ods output genmod.ModelFit = Fit&&_&i.var.
             genmod.&perr = &Perr.&&_&i.var.;
run;
```

Display 1 is an example PROC GENMOD statement generated by %LOGBINAUTO

```
title "Step 5";
title2 "Covariate #1: IVCATH_COL";
proc genmod data = renal_PI_compr_unkno_allcrbl ;
class AKIMR BMD_Cut_n (Descending) URINTCT_COL HFAIL RACE SST_COL IVCATH_COL;
model AKIMR (Event = "Yes") = BMD_Cut_n URINTCT_COL HFAIL RACE SST_COL IVCATH_COL / dist = bin link = log type1 ;
ods output genmod.ModelFit = FitIVCATH_COL genmod.type1 = Type1IVCATH_COL;
run;
```

Display 1. Sample PROC GENMOD Statement

Error-checking In Output Datasets

As failure of a log-binomial model to converge can occur, especially in later &J iterations, %LOGBINAUTO uses PROC SQL to confirm the creation of a covariate's model fit dataset {Fit&&_&I.VAR} before it can be processed for covariate screening {Appendix Section E}:

```
PROC SQL noprint;
  select * from Fit&&_&i.var.;
QUIT;
```

If the model fit dataset was properly output, the pre-selected model fit criterion information is extracted {e.g. AIC} as the variable VALUE and set to merge with final row of the model's Error dataset, resulting in a one observation dataset {Sel&&_&I.VAR} containing the:

1. covariate name
2. log-binomial model's AIC
3. covariate's $\chi 2$ p-value
4. covariate's deviance

```
%if &SQLObs ne 0 %then %do;
  data Fit&&_&i.var.;
  set Fit&&_&i.var;
    if substr (criterion, 1, 4) ne "AIC " then delete;
    mergefl = 1;
    keep mergefl value;
  run;

  data &Perr.&&_&i.var.;
  Length Source $15;
  set &Perr.&&_&i.var. end=eof;
    mergefl = 1;
    if eof then output;
    keep source deviance mergefl probchisq;
```

```
    run;

    data Sel&&_&i.var;
    merge &Perr.&&_&i.var. Fit&&_&i.var;
      by mergefl;
      drop mergefl;
    run;
```

Then, the covariate's selection dataset is appended into a &J iteration screening dataset {allSel&J.&ID.}, which will be used to select the "best" covariate:

```
    proc append base = allSel&j.&id. data = Sel&&_&i.var;
    run;
  %end;
```

Otherwise, if PROC SQL finds the model fit dataset to be empty, a QC %PUT statement notifies the user and SAS's internal error flags are reset to allow continued function of the macro. The &J loop continues with the next &I iteration:

```
  %else                                                                %do;
    %put WARNING: Model Estimates not created, &&_&i.var. not included in
    current iteration of evaluable variables;

    %put WARNING: SAS syntax check mode turned off for remainder of this SAS
    run;

    %put WARNING: You may want to remove &&_&i.var from your covariate list;

    options nosyntaxcheck nodmssynchk;
    options obs=max replace;
  %end;
```

COVARIATE SELECTION:

Once all covariates have been screened in a &J iteration, the cumulative covariate criteria dataset is sorted by the model fit criterion (VALUE, here representing AIC) to prioritize the covariate with the "best" fit.

%LOGBINAUTO checks 4 criteria to determine if a screened covariate is the "best fit":

1. p-value (PROBCHISQ) is less than or equal to the user-established p-value threshold (&PTHRESH).
2. A secondary covariate has not already been marked for selection (KEPTFLAG) in this iteration.
3. AIC value (VALUE) is lower than the AIC of the last model iteration (&LASTAIC from the prior &J loop).
4. Deviance is greater than 0 (deviance of 0.000 indicates the model failed to converge for the given covariate).

If all criteria are met, the secondary covariate is marked for selection (KEEP) and output to a dataset (keepvar&J.). All secondary covariates are output to a dataset (Modelsel&J). {Appendix Section F}:

```
  proc sort data = allSel&j.&id.;
    by value;
  run;

  data Modesel&j. keepvar&j.;
  set allSel&j.&id.;
    retain keptflag;
    if probchisq le &pthresh. and keptflag ne 1 and value lt &lastAIC.
```

```
  and deviance gt 0 then do;
    keep =1;
    keptflag =1;
  end;
  source = upcase(Source);
  output Modesel&j.;
  if keep then output keepvar&j.;
run;
```

Display 2 is a sample output listing of secondary covariates, ranked and selected for inclusion into a log-binomial model by %LOGBINAUTO:

Loop 2: Selecting variables

Deviance	Prob ChiSq	Value	keptflag	keep	Source
288.5129	0.0271	296.5129	1	1	HFAIL
288.6996	0.0426	296.6996	1	.	URINTCT_COL
288.7234	0.0306	296.7234	1	.	CRCL_BL
289.2184	0.0580	297.2184	1	.	IVCATH_COL
289.9254	0.0624	297.9254	1	.	RECVMYC_COL
290.0346	0.0667	298.0346	1	.	QUINOLONES
290.5398	0.1318	298.5398	1	.	SST_COL

Display 2. Sample Output of Ranked & Selected Covariates

RETAINING THE SELECTED COVARIATE:

PROC SQL is used to determine if the keepvar&J dataset was populated with a selected covariate; if no observations exist, %LOGBINAUTO will use a %GOTO statement to immediately terminate covariate screening and “transition into the final stage of the macro” {Appendix Section G}:

```
proc sql noprint;
  select count(keep) into :nobs from keepvar&j.;
quit;

%if &nobs. = 0 %then %goto leave;
```

Otherwise, keepvar&J is appended to the keptvar&ID dataset, which serves as a repository of all covariates selected during the current “call” of %LOGBINAUTO. The keptclassvar&ID dataset is then derived from keptvar&J, retaining any observations where the covariate was identified by the user, in &CLASSLIST, as a categorical/ordinal variable:

```
proc append base = keptvar&id. data = keepvar&j.;
run;

data keptclassvar&id.;
set keptvar&id.;
  do i= 1 to countw("%qupcase(&classlist.)");
    if Source = scan("%qupcase(&classlist.)", i) then output;
  end;
run;
```

PROC SQL is used to re-define the macro variables, &KEPTVAR and &KEPTCLASSVAR, as space-delimited lists of all covariates (SOURCE) found in keptvar&ID and keptclassvar&ID, respectively; you may recall that these macro variables are used within the covariate-screening PROC GENMOD statement as previously-selected covariates to be included in the model.

```
proc sql noprint;
  select source into :keptvar separated by ' ' from keptvar&id.;
```

```
quit;

proc sql noprint;
  select source  into :keptclassvar separated by ' ' from keptclassvar&id.;
quit;
```

PREPARING THE NEXT &J ITERATION:

The model fit criterion of the last-selected covariate is retained by re-defining &LASTAIC as the VALUE (e.g. AIC) of keepvar&J; this will be used in the next &J iteration's screening of covariates. The covariate selected in the last &J iteration is retained as both &REMVAR and &&REMVAR&J, the latter being reserved for user QC and including a trailing comma {Appendix Section H}:

```
proc sql noprint;
  select Value into :lastAIC separated by ' ' from keepvar&j.;
quit;

data _null_;
set keepvar&j.;
  call symput('remvar', upcase(Source));
run;

%let remvar&j. = %str(&remvar.,);
```

&K is defined as the next iteration of the &J loop, allowing for the "end-of-iteration" creation of the list of covariates remaining to be screened (&&COVLIST_NO&K) , by removing the last selected covariate (&&REMVAR&J) from the previous &J iteration's covariate list (&&COVLIST_NO&J); nested TRANWRD functions remove the covariate name and remaining comma.

```
%let k = %eval(&j. + 1);
%let covlist_no&k. = qsysfunc(tranwrd(&&covlist_no&j., &&remvar&j.,));
```

At the end of the &J iteration, a QC %PUT statement informs the user of the &J iteration, all presently-selected covariates, the remaining covariates to be screened, and the model fit criterion of the last-selected model.

```
%put Iteration &J., Kept Secondary Covariates are &keptvar. || Remaining
Secondary Covariates are &&covlist_no&k. ||Model AIC is &lastAIC.;
```

The next &J iteration is then initiated until all &J iterations have been completed, or %LOGBINAUTO terminates early when no covariate is selected during a &J iteration.

PRODUCING THE FINAL LOG-BINOMIAL MODEL:

Once the final &J iteration has executed, %LOGBINAUTO generates both a primary covariate-only log-binomial model and a model incorporating all selected covariates; the estimates of each model are output as &ID.unadjust_est and &ID.fin_est for the potential creation of "Relative Risk" datasets and/or tables {Appendix Section I}:

```
title "Unadjusted Model: &Outcome by &primcov.";
proc genmod data = &data.;
  class &outcome.  &primcov. (Descending);
  model &outcome. (Event = "Yes") = &primcov.
                                          / dist = bin link = log &perr.;
  %if &predlevel. le 2 %then %do;
    estimate "&outcome. &Primcov." &primcov. 1 -1 ;
  %end;
```

```
  %else %do;
    estimate "&outcome. &Primcov. Med vs. Low" &primcov. 0 1 -1;
    estimate "&outcome. &Primcov. High vs. Med" &primcov. 1 -1;
    estimate "&outcome. &Primcov. High vs. Low" &primcov. 1 0 -1;
  %end;
  ods output genmod.estimates = &id.unadj_est;
run;

title "Final Model for &Outcome. and &Primcov.";
proc genmod data = &data.;
  class &outcome.  &primcov. (Descending) &keptclassvar.;
  model &outcome. (Event = "Yes") = &primcov. &keptvar.
                                          / dist = bin link = log &perr.;
  %if &predlevel. le 2 %then %do;
    estimate "&outcome. &Primcov." &primcov. 1 -1;
  %end;

  %else %do;
    estimate "&outcome. &Primcov. Med vs. Low" &primcov. 0 1 -1;
    estimate "&outcome. &Primcov. High vs. Med" &primcov. 1 -1;
    estimate "&outcome. &Primcov. High vs. Low" &primcov. 1 0 -1;
  %end;
    ods output genmod.estimates = &id.fin_est;
run;
```

Display 3 is a sample PROC GENMOD statement with all covariates selected for inclusion by %LOGBINAUTO:

```
title "Final Model for AKIMR and BMD_Cut_n";


proc genmod data = renal_PI_compr_unkno_allcrbl ;
class AKIMR BMD_Cut_n (Descending) URINTCT_COL HFAIL RACE SST_COL IVCATH_COL;
model AKIMR (Event = "Yes") = BMD_Cut_n URINTCT_COL HFAIL RACE SST_COL IVCATH_COL / dist = bin link = log type1 ;
estimate "AKIMR BMD_Cut_n" BMD_Cut_n 1 -1 ;
ods output genmod.estimates = AKIbmd_allcrblfin_est;
run;
```

Display 3. Sample Final Log-Binomial model with macro-selected covariates

CONCLUSION

The %LOGBINAUTO macro offers users access to automated forward-selection of both continuous and categorical variables for a log-binomial model. Users can customize their selection process to utilize either Type I or Type III error, set $\chi 2$ p-value thresholds for covariate inclusion, provide a list of testable covariates in any order they choose, and can model outcome variables with two or three levels.

At present, %LOGBINAUTO only allows for AIC as the model fit criterion for log-binomial covariate selection, but the addition of new macro variable(s) could allow for alternative model fit criterion. Backward-selection is beyond the scope of this paper, but could be developed using %LOGBINAUTO as a foundation for the requisite nested &J : &I loops and use of the &COVLIST_NO, &KEPTVAR, and &REMVAR variables.

REFERENCES

1. Lodise JTP, Rosenkranz SL, Finnemeyer M, Huvane J, Pereira A, Sims M, et al. The emperor's new clothes: prospective observational evaluation of the association between the day 2 vancomycin exposure and failure rates among adult hospitalized patients with mrsa bloodstream infections (provide). *Open Forum Infect Dis* 2017;4:S30–1. Available at https://academic.oup.com/ofid/article/4/suppl_1/S30/4293954

ACKNOWLEDGMENTS

Thank you to Dr. Sue Rosencranz and Dr. Thomas Lodise for supporting my development and testing of the %LOGBINAUTO macro for the PROVIDE study.

Thank you as well to Todd Case for encouraging me to develop a presentation of this work and to present at PharmaSUG.

And lastly, thank you to my wife, Dr. Valerie Finnemeyer, for her ever-boundless support and editorial skill.

CONTACT INFORMATION

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

Matthew Finnemeyer
Vertex Pharmaceuticals, Inc.
mfinnemeyer@gmail.com

APPENDIX (FULL CODE)

SECTION A

```
%macro logbinauto (data = final_PI_compr, outcome=Failure, primcov =
BMD_Cut_n, seccovlist = %str(),perr = type1, pthresh = 0.10, classlist =
%str(), id = , predlevel = 2);

%if %sysevalf(%superq(data)= , boolean) %then %do
  %put ERROR: Source dataset not specified;
  %return;
%end;

%if %sysevalf(%superq(outcome)= , boolean) %then %do
  %put ERROR: Binomial outcome variable dataset not specified;
  %return;
%end;
```

SECTION B

```
%local keptvar keptclassvar currclassvar;
%let lastAIC = 99999.99;
%let covlist_no1 = %qupcase(&seccovlist.);
```

SECTION C

```
%do j = 1 %to   %sysfunc(countw(&covlist_no1., %str( ) ));
  %do i = 1 %to sysfunc(countw(%str(&&covlist_no&j.), %str(, ) ));
    %let _&i.var = %scan(&&covlist_no&j., &i.);

    %if %index(%qupcase(&classlist.), &&_&i.var) %then %let currclassvar =
    &&_&i.var;
    %else %let currclassvar = %str();
```

SECTION D

```
    title "Step &j.";
    title2 "Covariate #&i.: &&_&i.var.";
    proc genmod data = &data.  ;
      class &outcome.  &primcov. (Descending)  &keptclassvar. &currclassvar.;
      model &outcome. (Event = "Yes") = &primcov. &keptvar. &&_&i.var.
                                              / dist = bin link = log &perr. ;
      ods output genmod.ModelFit = Fit&&_&i.var.
                 genmod.&perr = &Perr.&&_&i.var.;
    run;
```

SECTION E

```
    PROC SQL noprint;
      select * from Fit&&_&i.var.;
    QUIT;

    %if &SQLObs ne 0 %then %do;
      data Fit&&_&i.var.;
      set Fit&&_&i.var;
```

```
      if substr (criterion, 1, 4) ne "AIC " then delete;
      mergefl = 1;
      keep mergefl value;
    run;

    data &Perr.&&_&i.var.;
    Length Source $15;
    set &Perr.&&_&i.var. end=eof;
      mergefl = 1;
      if eof then output;
      keep source deviance mergefl probchisq;
    run;

    data Sel&&_&i.var;
    merge &Perr.&&_&i.var. Fit&&_&i.var;
      by mergefl;
      drop mergefl;
    run;

    proc append base = allSel&j.&id. data = Sel&&_&i.var;
    run;

  %end;

  %else %do;
    %put WARNING: Model Estimates not created, &&_&i.var. not included in
    current iteration of evaluable variables;
    %put WARNING: SAS syntax check mode turned off for remainder of this
    SAS run;
    %put WARNING: You may want to remove &&_&i.var from your covariate
    list;
    options nosyntaxcheck nodmssynchk;
    options obs=max replace;
   %end;

%end; /*&I iteration end*/
```

SECTION F

```
proc sort data = allSel&j.&id.;
  by value;
run;

data Modesel&j. keepvar&j.;
set allSel&j.&id.;
  retain keptflag;
  if deviance gt 0 and probchisq le &pthresh. and value lt &lastAIC.
  and keptflag ne 1 then do;
    keep =1;
    keptflag =1;
  end;

  Source= upcase(Source);
```

```
    output Modesel&j.;
    if keep then output keepvar&j.;
   run;

  title "Loop &j.: Ranking / selecting variables";
  proc print data = Modesel&j.;
  run;
```

SECTION G

```
  proc sql noprint;
    select count(keep) into :nobs from keepvar&j.;
  quit;

  %if &nobs. = 0 %then %goto leave;

  proc append base = keptvar&id. data = keepvar&j.;
  run;

  data keptclassvar&id.;
  set keptvar&id.;
    do i= 1 to countw("%qupcase(&classlist.)");
      if Source = scan("%qupcase(&classlist.)", i) then output;
    end;
  run;


  proc sql noprint;
     select source into :keptvar separated by ' ' from keptvar&id.;
   quit;

   proc sql noprint;
     select source into :keptclassvar separated by ' ' from keptclassvar&id.;
   quit;
```

SECTION H

```
   proc sql noprint;
     select Value into :lastAIC separated by ' ' from keepvar&j.;
   quit;

   data _null_;
   set keepvar&j.;
     call symput('remvar', upcase(Source));
   run;

   %let remvar&j. = %str(&remvar.,);
   %let k = %eval(&j. + 1);
   %let covlist_no&k. = qsysfunc(tranwrd(&&covlist_no&j., &&remvar&j.,));

   %put Iteration &J., Kept Secondary Covariates are &keptvar. || Remaining
   Secondary Covariates are &&covlist_no&k. ||Model AIC is &lastAIC.;

%end; /*&J iteration end*/

%leave:
```

```
%put No More Selectable Covariates: J = &J.;
```

SECTION I

```
title "Unadjusted Model: &Outcome by &primcov.";
proc genmod data = &data.;
  class &outcome.  &primcov. (Descending);
  model &outcome. (Event = "Yes") = &primcov./ dist = bin link = log &perr. ;

  %if &predlevel. le 2 %then %do;
    estimate "&outcome. &Primcov." &primcov. 1 -1;
  %end;

  %else %do;
    estimate "&outcome. &Primcov. Med vs. Low" &primcov. 0 1 -1;
    estimate "&outcome. &Primcov. High vs. Med" &primcov. 1 -1;
    estimate "&outcome. &Primcov. High vs. Low" &primcov. 1 0 -1;
  %end;

  ods output genmod.estimates = &id.unadj_est;
run;

title "Final Model for &Outcome. and &Primcov.";
proc genmod data = &data.;
  class &outcome.  &primcov. (Descending) &keptclassvar.;
  model &outcome. (Event = "Yes") = &primcov. &keptvar.
                                          / dist = bin link = log &perr.;
  %if &predlevel. le 2 %then %do;
    estimate "&outcome. &Primcov." &primcov. 1 -1;
  %end;

  %else %do;
    estimate "&outcome. &Primcov. Med vs. Low" &primcov. 0 1 -1;
    estimate "&outcome. &Primcov. High vs. Med" &primcov. 1 -1;
    estimate "&outcome. &Primcov. High vs. Low" &primcov. 1 0 -1 ;
  %end;

  ods output genmod.estimates = &id.fin_est;
run;

%put Final Kept Secondary Covariates are &keptvar. ||Final Model AIC is
&lastAIC.;

%mend logbinauto;
```