Paper 102-2019

Using convergence status in a macro to iterate through a series of models

Kristi Busico Metzger, Children's Hospital of Philadelphia, Philadelphia, Pennsylvania

ABSTRACT

Some analyses require running a series of models with the same specifications but with different outcomes. In epidemiologic analyses, we often want to record the results of a crude model (i.e., one with no covariates) and an adjusted model (i.e., one that includes covariates). In these situations, it can be helpful to construct a macro to loop through both the crude and adjusted models and then output the results into data sets using ODS OUTPUT statements. For analyses in which there is potential for an iterative model to not converge (such as for rarer outcomes), it is helpful to also output the convergence status. The convergence status can be used within a macro to automatically direct the analytic process through the desired analyses if convergence is attained or to alternative options if it is not. In this example, I will use a series of repeated-measures generalized estimating equations models in PROC GENMOD to model the rate ratios of specific types of motor vehicle crashes by driver age. For each type of motor vehicle crash, the macro loops through a crude model, then to a partially adjusted model (controlling for a limited number of covariates) then to a fully adjusted model (controlling for all covariates) if the more parsimonious model converges or exits the macro loop if it does not converge. The macro concatenates the results of each model for all outcomes into a single dataset with indicators for the type of model (crude, partially adjusted, or fully adjusted).

INTRODUCTION

Epidemiologists conduct regression modeling to describe associations between an outcome of interest and an exposure of interest, adjusting for covariates or other confounding factors. Often, an epidemiological analysis plan will involve multiple models with different sets of covariates, different outcomes of interest, or both. In this situation, macros can be useful tools to loop through the series of models and save the results to more easily output into a manuscript table.

For this paper, I will use a simplified example from a series of recent epidemiological studies of driver characteristics and motor vehicle crashes (Curry 2019; Curry 2017). In this example, we want to compare crash rates among drivers of different age groups (younger vs older), adjusting for demographic characteristics including sex, race/ethnicity, and household income. We followed thousands of drivers over several years and recorded the number of crashes each driver was involved in for each month of follow up. The analysis uses a generalized estimating equation model with a Poisson process to estimate the rate ratio. In addition to adjusting for demographics, we account for within-driver correlation using a repeated-measures framework. Our goal is to not only compare overall crash rates, but also specific types of crashes, including crashes in which the driver was at fault, crashes that occurred at night, and crashes which involved alcohol use by the driver.

THE DESIRED MODELS

This analysis uses PROC GENMOD to conduct repeated-measures generalized estimating equations models. ODS OUTPUT is used to output the results (rate ratio and 95% confidence interval) from the ESTIMATE statement into a SAS dataset. By saving the results in a dataset, we can generate a table that aggregates results from multiple models in SAS and reduce transcription errors made from manually retyping results from the output window to another document. The code for the adjusted model is as follows:.

```
ods output estimates=_est;

proc genmod data=crashdata;
class uniqueid agegroup(ref='Older' param=ref) sex raceethn income;
model CRASH_ALL = agegroup sex raceethn income
   / dist = poisson link = log offset = monthoffset type3 pscale;
```

```
repeated subject=uniqueid / corr=IND;
estimate "CRASH_ALL" agegroup 1;
run;
```

The code for the crude model examines just the association of age on all crashes and removes the covariates sex, race/ethnicity, and household income. The model statement is reduced to the following (the other statements remain the same):

```
model CRASH_ALL = agegroup
   / dist = poisson link = log offset = monthoffset type3 pscale;
```

THE CONVERGENCE PROBLEM

Because we have hundreds of thousands of months of follow-up for thousands of drivers, plus the models use a REPEATED statement, we know that some fully adjusted models with the more rare outcome (such as crashes involving alcohol use) may not converge. To account for this possibility, we want to do two things. First, we want to use ODS OUTPUT to save the part of the results that states the model's convergence status so that we don't have to go searching through the output or log windows. We can add the following code to the ODS OUTPUT statement prior to our model:

```
ods output convergencestatus=_converge;
```

We can also run a partially adjusted model with just sex as a covariate and modify the MODEL statement as follows:

```
model CRASH_ALL = agegroup sex
   / dist = poisson link = log offset = monthoffset type3 pscale;
```

THE MACRO

Because we're repeating these analyses for the 4 outcomes of interest, we want to use a macro to loop through the crude and adjusted models and save the results of each model into a single output dataset that includes all the results. We also want the macro to run the code for the adjusted model only if the more parsimonious model converges or exit the macro loop if it does not converge. By using a macro to go through this process, we can reduce the potential for errors due to rewriting repetitive coding. We can also increase efficiency by allowing all the models to run sequentially without stopping to check for convergence manually. The full code for the macro crashloop is included in the Appendix.

THE MACRO PARAMETERS

When invoking the macro, we will submit the following macro parameters:

- dsin = input dataset that will be used for the models
- crashlist = list of crash outcome variables, separated by spaces
- dsout = name of output dataset that will including modeling results

Thus, for this example, we'll submit the following code to invoke the macro:

```
%crashloop (dsin = CRASHDATA,
            crashlist = CRASH_ALL CRASH_ATFAULT CRASH_NIGHT CRASH_ALCOHOL,
            dsout = RESULTS)
```

THE LOOP

Within the body of the macro, we'll use a %DO process and CALL SYMPUT to loop through the list of crash outcome variables. For each variable in the crashlist, the macro will potentially move through a series of models---first the crude model, then the partially adjusted model, then the fully adjusted model---and output the convergence status and results of the ESTIMATE statement of each model into separate

SAS data sets. The data sets produced by each loop will include the suffix &i to differentiate between the outcome variables numerically. The first part of the macro code, through the crude model, is listed below:

```
%do i=1 %to %sysfunc(countw("&crashlist "));

   data _NULL_;
   call symputx('crashvar', "%qscan(&crashlist, &i, ' ')");
   run;

   /** CRUDE model **/

   ods output convergencestatus=_converge_crude_&i
              estimates=_estimates_crude_&i;

   proc genmod data=&dsin;
   class uniqueid agegroup(ref='Older' param=ref);
   model &crashvar = agegroup
      / dist = poisson link = log offset = monthoffset type3 pscale;
   repeated subject=uniqueid / corr=IND;
   estimate "&crashvar" agegroup 1;
   run;

*** ADDITIONAL CODE ***;

%end;
```

THE CONDITIONAL SEQUENCE

Next, we use a PROC SQL statement to assess the maximum value of the status variable in the convergence status data set for the crude model and create a macro variable with that value:

```
proc sql;
select max(status)
into :c_status
from _converge_crude_&i;
quit;
```

If status has a maximum value of zero, then the model converged; if status is greater than zero, then the model did not converge. Note that this data set contains additional information about the reason the model may not have converged, although we don't use it in this paper.

We will use the convergence status to conditionally move through the rest of the macro. If the crude model does not converge, there is no reason to try to run the adjusted model. We use %IF - %THEN/%ELSE processing combined with a %DO loop to dynamically direct the macro:

```
%if &c_status > 0 %then %do;

   data results_&i;
   length crash convergence $20;
   crash="&crashvar";
   convergence='No convergence';
   run;

%end;

%else %if &c_status = 0 %then %do;

   /** Partially adjusted model controlling for sex only **/
```

```
   ods output convergencestatus=_converge_partial_&i
              estimates=_estimates_partial_&i;

   proc genmod data=&dsin;
   class uniqueid agegroup(ref='Older' param=ref) sex;
   model &crashvar = agegroup sex
      / dist = poisson link = log offset = monthoffset type3 pscale;
   repeated subject=uniqueid / corr=IND;
   estimate "&crashvar" agegroup 1;
   run;

*** ADDITIONAL CODE ***;

%end;
```

If the convergence status indicates the model did not converge, the macro is directed to create a results data set with a value of “no convergence” inserted into an indicator variable called convergence and then skip to the end of the macro. If the convergence status, indicates the model did converge, then macro is directed to run the partially adjusted model.

Similarly, the macro then extracts status from the convergence status data set for the partially adjusted model in a PROC SQL statement and create another macro parameter. Another %IF - %THEN/%ELSE processing code is combined with a %DO loop to either create a results dataset or move on to the fully adjusted model. In the situation in which the partially adjusted model does not converge, we do have results from the crude model, so the results data set will include the rate ratio and 95% confidence intervals from the crude model only. The indicator variable convergence now has a value of “Crude only.” The macro will then skip to the end.

Finally, if the partially adjusted model converged, the macro will dynamically move to the code for the fully adjusted model. This time we’ll use the convergence status database for the fully adjusted model to conditionally direct the macro to either save the results of the crude and fully adjusted model (or the results of the crude and partially adjusted model) and add the variable convergence with the appropriate indicator value.

THE RESULTS

After all the %IF - %THEN/%ELSE processing with %do loops are executed, the final section of the macro concatenates the results dataset for each crash type:

```
data &dsout.;
set results_:;
run;
```

Because we conditionally created the results_&i data sets with the same variables and variable attributes, the final data step succeeds in creating our new results data base &dsout. Table 1 below is an example of how this results data base could look.

crash	convergence	cRR	cRR_L	cRR_U	aRR	aRR_L	aRR_U
CRASH_ALL	FULL MODEL	1.25	1.14	1.36	1.23	1.13	1.33
CRASH_ATFAULT	FULL MODEL	1.30	1.18	1.41	1.28	1.15	1.40
CRASH_NIGHT	PARTIAL MODEL	1.15	0.99	1.30	1.14	0.95	1.35
CRASH_ALCOHOL	CRUDE MODEL	2.30	4.81	0.62	.	.	.

Table 1. Sample data base of results from macro.

CONCLUSION

We can take advantage of outputted results to dynamically progress through code within a macro. By using ODS OUTPUT to save the convergence status from of the model output, then PROC SQL to assess the value of the convergence status, we can conditionally move through a series of models using %IF - %THEN/%ELSE processing and %DO LOOPS to save the final results into a single data set. This process has several advantages over hard coding all the possible models, including 1) reducing coding errors associated with rewriting similar code again and again, 2) recording information about convergence status in a data set for easy reference, and 3) saving time by reducing the need to submit code and review it model by model.

REFERENCES

Curry AE, Yerys BE, Metzger KB, Carey ME, Power TJ. 2019. “Traffic Crashes, Violations, and Suspensions Among Young Drivers With ADHD.” *Pediatrics* 143 (6) e20182305; DOI: 10.1542/peds.2018-2305

Curry AE, Metzger KB, Williams AF, Tefft BC. 2017. “Comparison of older and younger novice driver crash rates: Informing the need for extended Graduated Driver Licensing restrictions.” Accident Analysis & Prevention 108(66-73).

ACKNOWLEDGEMENTS

I’d like to thank Alli Curry for the opportunity to work on these interesting epidemiological studies. I’d also like to thank Melissa Pfeiffer for all of her insight, particularly in relation to data and SAS.

CONTACT INFORMATION

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

Kristi Busico Metzger, PhD, MPH
Children’s Hospital of Philadelphia
512-825-5543
krisbusmetz@gmail.com

APPENDIX

```
%macro crashloop (dsin=, crashlist=, dsout=);

%do i=1 %to %sysfunc(countw("&crashlist "));

   data _NULL_;
   call symputx('crashvar', "%qscan(&crashlist, &i, ' ')");
   run;

   /** CRUDE model **/

   ods output convergencestatus=_converge_crude_&i
              estimates=_estimates_crude_&i;

   proc genmod data=&dsin;
   class uniqueid agegroup(ref='Older' param=ref);
   model &crashvar = agegroup
      / dist = poisson link = log offset = monthoffset type3 pscale;
   repeated subject=uniqueid / corr=IND;
   estimate "&crashvar" agegroup 1;
   run;

   proc sql;
   select max(status)
   into :c_status
   from _conv_crude_&i;
   quit;

   %if &c_status > 0 %then %do;

      data results_&i;
      length crash convergence $20;
      crash="&crashvar";
      convergence='No convergence';
      run;

   %end;

   %else %if &c_status = 0 %then %do;

      /** Partially adjusted model controlling for sex only **/

      ods output convergencestatus=_converge_partial_&i
                 estimates=_estimates_partial_&i;

      proc genmod data=&dsin;
      class uniqueid agegroup(ref='Older' param=ref) sex;
      model &crashvar = agegroup sex
         / dist = poisson link = log offset = monthoffset type3 pscale;
      repeated subject=uniqueid / corr=IND;
      estimate "&crashvar" agegroup 1;
      run;

      proc sql;
      select max(status)
      into :p_status
```

```
from _conv_partial_&i;
quit;

%if &p_status > 0 %then %do;

   data results_&I (keep=crash convergence crr:);
   length crash convergence $20;
   set _est_crude_&i (rename=(label=crash meanestimate=cRR
                              meanlowercl=cRR_L meanuppercl=cRR_U));
   convergence='Crude only';
   run;

%end;

%else %if &p_status = 0 %then %do;

   /** Fully adjusted model controlling for sex, race/ethnicity,
       household income **/

   ods output convergencestatus=_converge_full_&i
              estimates=_estimates_full_&i;

   proc genmod data=&dsin;
   class uniqueid agegroup(ref='Older' param=ref) sex raceethn income;
   model &crashvar = agegroup sex raceethn income
      / dist = poisson link = log offset = monthoffset type3 pscale;
   repeated subject=uniqueid / corr=IND;
   estimate "&crashvar" agegroup 1;
   run;

   proc sql;
   select max(status)
   into :f_status
   from _conv_full_&i;
   quit;

   %if &f_status > 0 %then %do;

      data results_&i (keep=crash convergence crr: arr:);
      length crash convergence $20;
      merge _est_crude_&i (rename=(label=crash meanestimate=cRR
                                   meanlowercl=cRR_L meanuppercl=cRR_U))
            _est_partial_&I (rename=(meanestimate=aRR
                                  meanlowercl=aRR_L meanuppercl=aRR_U));
      convergence='Partial model';
      run;

   %end;

   %else %if &f_status = 0 %then %do;

      data results_&i (keep=crash crr: convergence);
      length crash convergence $20;
      merge _est_crude_&i (rename=(label=crash meanestimate=cRR
                                   meanlowercl=cRR_L meanuppercl=cRR_U))
      _est_full_&I (rename=(meanestimate=aRR meanlowercl=aRR_L
                              meanuppercl=aRR_U));
```

```
            convergence='Full model';
            run;

         %end;

      %end;

   %end;

%end;

data &dsout.;
set results_:;
run;

proc datasets library=work;
delete _:;
quit;

%mend crashloop;
```