

Backward Selection Method: Immunity Test of covariates

Prathamesh Athavale, inventiv international pharma services pvt. Ltd., Dombivli, India

ABSTRACT

Backward Selection Method: Elimination round of variables using Mixed model repeated measure. This method is used to eliminate covariates in the model based on their significance. It follows step by step approach to eliminate variables one by one.

INTRODUCTION

Purpose of this paper is to provide **SAS® code** which can be used for **backward selection** method. This code is useful in case where backward selection of variable is performed on covariates. Mixed model repeated measure is the analysis method being used to perform selection of the variables.

SAS provides Type III Tests of Fixed Effects when we run **PROC MIXED** on data which determines covariate being significant or non-significant based on their prob-F value (**Pr > F** column value). If the prob-F value is > 0.05 (here 0.05 is determined by alpha or can be specified) the variable will be dropped from the model and this process is repeated till we get all prob-F values < 0.05. This is very useful when we are not sure which all variables are contributing to the model. With the help of Backward Selection we can find optimum model for the analysis.

ILLUSTRATION

Let's start with an example; At the beginning we might not be sure which all factors are causing change in certain Lab test from its baseline. The cause might be some of the variables among baseline height, baseline weight, baseline age, baseline vitals, etc. along with treatment and baseline of that Lab Test itself.

This particular code also determines Covariance Structure required for the model on the basis of minimum AIC value (One of the approach to find covariance structure) and uses that structure to eliminate insignificant covariates from the model. Covariance structure is very crucial in this case because analysis method is Mixed Model Repeated measures collected over different time points on the same subject (patient).

It is always good to list all possible variables that we think might be required to find appropriate model. This code will eliminate insignificant covariates one after the other instead of dropping all insignificant variables at once. Dropping criteria is Maximum **prob-F value**. Variable with maximum **prob-F value** greater than 0.15 will be dropped in first iteration and model results will be obtained post dropping the variable and then next variable will be dropped till all the variables with **prob-F value** less than or equal to 0.15 are left in the model along with fixed effects such as treatment, time points etc.

Type 3 Tests of Fixed Effects

	Num Den			
Effect	DF	DF	F Value	Pr > F
.....				
var05	1	254	14.78	0.0002
var11	3	254	3.53	0.0156
var25	1	254	3.79	0.6526

THINGS TO REMEMBER

Before you use following macro and piece of code for the purpose of finding right model few things you must keep in mind and those are:

1. Use sufficient and realistic data to perform analysis otherwise results would not be reliable.
2. Correctly determine required variables that you want to pass in this code.
3. Recognize **CLASS statement variables** correctly i.e. those variables which are categorical in nature (And in case you are passing any character variables).
4. Make sure you don't have **duplicate** records at time point for any subject.
5. Be careful while copy pasting code in **SAS** as alignment may differ from what appears in paper.

```
/*Description of the code
```

```
*****
```

Purpose of the code: Eliminate variables one by one based on their prob-f value

Instruction: 1. Use realistic and sufficient data

2. Read comments before executing the program

3. Code has a provision of assessing 26 variables but list can be amended based on requirement

4. bsln trtname and vstnm are must variables if the values are captured then set to 1.

5. In case information is not captured on different time points then this code works on single time point information in that case set value of vstnm as 1.

6. Edit class statement as per requirement.

```
**end*****
```

```
*****/
```

```
options mprint mlogic nocenter nodate symbolgen mautosource ;
```

```
data anadata1;
```

```
  set anadata_1;
```

```
  bsln = baselinevalue;***This is the baseline value of the parameter we are assessing;
```

```
  trtname = treatment;***Name of the treatment|| This may be a different control variable depending upon requirement;
```

```
  vstnm = timepoint;***Variable which contains time point information like visit numbers on which parameter values have been collected;
```

```
*****
```

```
  Store independent variable names (variables which are on right side of |=| in the model
```

```
  ie variables that we assess significant as covariates
```

```
  in input dataset in serial order as given below
```

```
  Note: alter/expand this list as per your requirement
```

```
*****;
```

```
  var01 = invnum;
```

```
  var02 = age;
```

```
  var03 = gender;
```

```
  var04 = hgtatbsln;
```

```
  var05 = bmiatbsln;
```

```
  var06 = hba1cbl;
```

```
  var07 = lbblvaltr;
```

```
  var08 = vitd_avg;
```

```
  var09 = blmyostn;
```

```
  var10 = srghtr;
```

```
  var11 = xyz1;
```

```
  var12 = xyz2;
```

```
  var13 = xyz3;
```

PhUSE 2014

```
var14 = xyz4;
var15 = xyz5;
var16 = xyz6;
var17 = xyz7;
var18 = xyz8;
var19 = xyz9;
var20 = xyz10;
var21 = xyz11;
var22 = xyz12;
var23 = xyz13;
var24 = xyz14;
var25 = xyz15;
run;

*****
*****
input: It is an input dataset name
*****
*****;
%macro model(input=anadata1);

*****
Declare macro variables for their global existence;
*****;
%global var01 var02 var03 var04 var05 var06 var07 var08 var09 var10 var11 var12
var13 var14 var15 var16 var17 var18 var19
      var20 var21 var22 var23 var24 var25 var26;

*****
Assign macro variable names with their actual variable names
alter/expand this list as per your requirement
*****;
%let var01 = ;*in case you don't want assign value to var01. Assign it missing this
way;
%let var02 = var02;
%let var03 = var03;
%let var04 = var04;
%let var05 = var05;
%let var06 = var06;
%let var07 = var07;
%let var08 = var08;
%let var09 = var09;
%let var10 = var10;
%let var11 = var11;
%let var12 = var12;
%let var13 = var13;
%let var14 = var14;
%let var15 = var15;
%let var16 = var16;
%let var17 = var17;
%let var18 = var18;
%let var19 = var19;
%let var20 = var20;
%let var21 = var21;
%let var22 = var22;
%let var23 = var23;
%let var24 = var24;
```

PhUSE 2014

```
%let var25 = var25;
%let var26 = ;*Another example of not assigning any value to the variable;

proc sql noprint;
  select count(distinct subno) into: totrec
  from &input;
quit;

data model;
  set &input;
run;

*****
Declare macro variables for their global existence;
*****;

%global nrobs dsid chk cov cov1 covar typ  covariancestructurefinalaccepted;

%let typ = ;
%let cov = ;

*****
Macro to generate AIC value for Converging covariance structure
*****;

%macro fin_aicmin_cov();

  %if %sysfunc(exist(convergence)) %then %do;
    proc datasets nolist nodetails lib = work;
      delete convergence;
    run;
  %end;

  %if %sysfunc(exist(aic_val)) %then %do;
    proc datasets nolist nodetails lib = work;
      delete aic_val;
    run;
  %end;

  %if %sysfunc(exist(min_aicval)) %then %do;
    proc datasets nolist nodetails lib = work;
      delete min_aicval;
    run;
  %end;

  %macro get_aic_val(covar=,typ = );

  %if %sysfunc(exist(convrg_critra)) %then %do;
    proc datasets nolist nodetails lib = work;
      delete convrg_critra;
    run;
  %end;

  %if %sysfunc(exist(fit)) %then %do;
    proc datasets nolist nodetails lib = work;
      delete fit;
    run;
  %end;

  %let cov1=&covar;
```

PhUSE 2014

```

ods listing close;
proc mixed data = model;
    class trtname vstnm subno &var01 &var03 &var10 &var11; *Class variables
requires update as per requirement;
    model anavar = bsln &var01 trtname vstnm trtname*vstnm &var02 &var03
&var04 &var05 &var06 &var07 &var08 &var09 &var10 &var11 &var12 &var13 &var14 &var15
&var16 &var17 &var18 &var19 &var20 &var21 &var22 &var23 &var24 &var25 &var26 /htype
= 3 alpha = .10
        ddfm = kr;
    %if &typ = arh or &typ=ar %then %do ;
        repeated vstnm / type = &cov1. sub = subno;
    %end ;
    %else %do;
        repeated vstnm / type = &cov1. sub = subno;
    %end ;
    ods output convergencestatus=convrgr_critra FitStatistics=fit;
run;
ods listing ;

%if %sysfunc(exist(convrgr_critra)) %then %do;

    %if %sysfunc(exist(convergence)) %then %do;
        data convergence (keep= reason cov);
        set convergence convrgr_critra(where = ( lowercase(reason) =
"convergence criteria met.") in =a );
        if a then do; cov="&cov1"; type ="&typ."; end;
    run;
    %end;
    %else %do;
        data convergence (keep= reason cov);
        set convrgr_critra(where = ( lowercase(reason) = "convergence
criteria met.") in =a );
        if a then do; cov="&cov1"; type ="&typ."; end;
    run;
    %end;

%end;

%if %sysfunc(exist(fit)) %then %do;

    %if %sysfunc(exist(aic_val)) %then %do;
        data aic_val ;
            length cov type $10.;
        set aic_val fit(where = ( compress(lowercase(descr)) =
"aic(smallerisbetter)") in =a );
        if a then do; cov="&cov1"; type ="&typ."; end;
    run;
    %end;
    %else %do;
        data aic_val ;
            length cov type $10.;
        set fit(where = (compress(lowercase(descr)) =
"aic(smallerisbetter)") in =a );
        if a then do; cov="&cov1"; type ="&typ."; end;
    run;
    %end;

%end;

%mend get_aic_val;

```

PhUSE 2014

```

%let nrobs = 0 ;
%get_aic_val(covar=un,typ =un);
%get_aic_val(covar=toeph,typ =toeph);
%get_aic_val(covar=arh(1),typ=arh);
%get_aic_val(covar=csh,typ =csh);
%get_aic_val(covar=toep,typ =toep);
%get_aic_val(covar=ar(1),typ=ar);
%get_aic_val(covar=cs,typ =cs);

%if %sysfunc(exist(aic_val)) %then %do;

    proc sql noprint;
        create table min_aicval as
        select * from aic_val where value = (select min(value) from
aic_val);
        select cov, type into :cov, :typ from min_aicval;
    run;

    %let cov=&cov.;
    %let typ=&typ.;
%end;

%let cov2 = %sysfunc(compress(&cov,"()1")) ;

data cov;
    cov = "&cov2";
    length covlnm $50;
    select(cov);
        when("un") covlnm = "Unstructured";
        when("toeph") covlnm = "Heterogeneous toeplitz";
        when("arh") covlnm = "Heterogeneous first-order autoregressive";
            when("csh") covlnm = "Heterogeneous Compound Symmetric";
        when("toep") covlnm = "Toeplitz";
        when("ar") covlnm = "Autoregressive";
        when("cs") covlnm = "Compound Symmetric";
        otherwise covlnm = " ";
    end;
run;

data cov;
    length cov $ 10.;
    set cov;
    if compress(lowercase(cov)) in ("ar", "arh") then cov = compress(cov) ||
"(1)";
run;

%mend fin_aicmin_cov;

%fin_aicmin_cov;

%macro backwrdsselection();

ods listing close;
proc mixed data = model;
    class trtname vstnm subno &var01 &var03 &var10 &var11;*Class variables
requires update as per requirement;
    model anavar = bsln &var01 trtname vstnm trtname*vstnm &var02 &var03 &var04
&var05 &var06 &var07 &var08 &var09 &var10 &var11 &var12 &var13 &var14 &var15 &var16
&var17 &var18 &var19 &var20 &var21 &var22 &var23 &var24 &var25 &var26 /htype = 3
alpha = .10
    ddfm = kr;

```

PhUSE 2014

```
%if &typ = arh or &typ=ar %then %do ;
    repeated vstnm / type = &cov. sub = subno;
%end ;
%else %do;
    repeated vstnm / type = &cov. sub = subno;
%end ;
ods output tests3 = tst_temp;
run;
ods listing ;

data missing_prob;
    set tst_temp;
    where probf = .;
run;

%if %sysfunc(exist(tst_temp)) %then %do;
proc sort data = tst_temp;
    where lowercase(effect) not in ('trtname*vstnm' 'trtname' 'bsln' 'vstnm') and probf
> .15;
    by descending probf;
run;
%end;
proc sql noprint; select count(*) into :obscount
from tst_temp;
quit;

%global execflg;

%if &obscount > 0 %then %do;

data tst_temp;
    set tst_temp;
    if _n_ = 1 then call symput('clearvar', effect);
    if _n_ = 1 then call symput('valueftbl', compress(put(probf,8.5)));
run;

%put dropped variable is &clearvar with probf as &valueftbl;

%let &clearvar = ;

%let execflg = 1;

        proc datasets nolist nodetails lib = work;
            delete tst_temp;
        run;

%end;

%else %do;

%let execflg = 0;

%if %sysfunc(exist(missing_prob)) %then %do;
proc sort data = missing_prob;
    where lowercase(effect) not in ('trtname*vstnm' 'trtname' 'bsln' 'vstnm');
    by probf;
run;

data missing_prob;
    set missing_prob end = end1;
    if end1 then call symput('tot_mis_var_cnt', compress(put(_n_,8.0)));
    call symput('clearvar' || compress(put(_n_,8.0)),effect);
```

PhUSE 2014

```

run;

%do i = 1 %to &tot_mis_var_cnt;

%let &&clearvar&i = ;

%end;

%end;

%end;

%mend backwrselection;

%backwrselection;

*****
This part does iterations till only required variables remain in the model
*****;

%macro exec();

%do %while(&execflg = 1);

%fin_aicmin_cov;

%backwrselection;

%end;

%if &execflg = 0 %then %do;

%fin_aicmin_cov;

%let varavlbl = &var02 &var03 &var04 &var05 &var06 &var07 &var08 &var09 &var10
&var11 &var12 &var13 &var14 &var15 &var16 &var17 &var18 &var19
                &var20 &var21 &var22 &var23 &var24 &var25 &var26;

%put &varavlbl;

%end;

%mend exec;

%exec;

*****
*****;
*Checking levels of the data on fixed effects like time points and treatment
;
*****
*****;

proc sort data = model nodupkey out = chk (keep=trtname vstnm) ;
  by TRTNAME vstnm ;
run ;

data _null_ ;
  set chk ;
  level= compbl(''|compress(put(TRTNAME,best.))||'|'
'|'|'|'|compress(put(vstnm,6.2))||'|') ;

```

PhUSE 2014

```
call symput ('level' || left(put(_n_,3.)),level) ;
call symput ('tot',put(_n_,5.)) ;
run ;

*****
Final run of proc mixed
*****;

ods listing;
proc mixed data = model;
    class trtname vstnm subno &var01 &var03 &var10 &var11;*Class variables
requires update as per requirement;
    model anavar = bsln &var01 trtname vstnm trtname*vstnm &var02 &var03
&var04 &var05 &var06 &var07 &var08 &var09 &var10 &var11 &var12 &var13 &var14 &var15
&var16 &var17 &var18 &var19 &var20 &var21 &var22 &var23 &var24 &var25 &var26 /htype
= 3 alpha = .10
        ddfm = kr;
    %if &typ = arh or &typ=ar %then %do ;
        repeated vstnm / type = &cov. sub = subno;
    %end ;
    %else %do;
        repeated vstnm / type = &cov. sub = subno;
    %end ;
    lsmeans TRTNAME*vstnm /diff pdiff alpha=0.1 ;
    ods output   diffs=dif lsmeans = ls tests3 = tst;
run;

%mend model;

%model(input=anadata1);
.
```

CONCLUSION

Make sure that data for analysis is accurate, sufficient and realistic. Double check the data provided to this code as code will execute even on wrong data as long as required variables and proper syntax is available but the outcome will not be appropriate.

ACKNOWLEDGMENTS

I would like to thank my family, friends and colleagues for their constant support.

RECOMMENDED READING

Kindly read “**Common Statistical Methods for Clinical Research with SAS Examples**” by Glenn Walker for understanding concepts of **PROC MIXED**.

CONTACT INFORMATION

Author Name: Prathamesh Prakash Athavale
Company: inventiv international pharma services pvt. Ltd.
Address: 304, Om Sadguru, Pendse Nagar
City / Postcode: Dombivli(India) 421201
Work Phone: 02030569230
Fax:
Email: Prathamesh.athavale@gmail.com
Web:

Brand and product names are trademarks of their respective companies.