

/* Automatic selection macro of fixed effects with PROC GLIMMIX
DECEMBER 2007

Programmer: Adrien FRANCAIS
Statistical Enginner

Albert Bonniot Institute
Team 11-Outcome of cancer and critical illnesses
UJF-INSERM U823
38706 LA TRONCHE CEDEX
FRANCE
04 76 54 94 00

MACRO VARIABLES

TABLE : work table to select variables in the model.

SELECTION : you have to choose the type of variables selection : BACKWARD or FORWARD

By default, the BACKWARD procedure is specified (faster).

I thought to compute the STEPWISE procedure but it doesn't matter to do it because with the FORWARD procedure, when a variable enter in the model, it can't be excluded after.

N : maximum level in the hierarchical model. It must be contained between 2 and 5. For example, one random effect means a 2-level model(classes for example) and 2 two randoms effects means a 3-level model (school and classes).

LEVEL5 LEVEL4 LEVEL3 NIVEAU2 : random effect to specify (for example : school and classes).

EFFECT5 EFFECT4 EFFECT3 EFFECT2 : you have to precise which kind of random effect you want to take into account

By default, the INTERCEPT random effect is specified. You can use `_RESIDUAL_` option

METHOD : method used to comute the maximum likelihood, you have to choose between MSPL (default), RSPL, MMPL and RMPL. Look at the GLIMMIX.PDF online for some explanations.

DDFM : degrees of freedom used to compute tests and confidence intervals. You have to choose between

BETWITHIN (BW), CONTAIN (CON), NONE, RESIDUAL (RES), KENWARDROGER (KR) and SATTERTHWAITE (SATTERTH).

Values in brackets indicates values to indicate in the macro variable.

By default, we use the SATTERTH ddfm.

Look at the GLIMMIX.PDF online for some explanations.

VARLIST : variables proposed to enter in the model : binary variables(0 or 1) or continuous variables.

Becareful : you have to take into account capital letters included in the variable

VARCIBLE : it is the binary variable outcome to predict, coded as 0 or 1.

THRESHOLD : inclusion criteria for covariates into the multivariable model.

This p-value is given by the F test when we add this variable in the precedent model (FORWARD method) or when

it is the worst p value in the model.

We fixed by default pvalue at 0.05 .

TYPE : type of covariance matrix. By default, we use VC for unstructured covariance, CHOLESKY (CHOL) for Cholesky parametrization, UN, AR(1) for an autoregressive correlation.

INCLUDE : list of variables which you want to force in the model

VARCLASS : liste of variables to treat in class

Look at the GLIMMIX.PDF online for some explanations.

Before using this macro, you have to :

- Prepare your database with correct variables (without missing values)
- Compile the macro.
- Execute the macro :

```
%glimmixstep(table=appli,selection=backward,n=3,level3=pays2,level2= N__du_centre,
effect3=intercept,effect2=intercept,method=MSPL,ddfm=satterth,
varlist=phys1000 nurses totexp govexphealth urbanpop internet private,
varcible=conflicus,threshold=0.05,type=vc,include=,varclass=);
```

You can see LOG screen for information of algorithm, number of STEP and which variable was retained (or removed) at each step.

In the OUTPUT screen, you have all GLIMMIX procedures used by the macro. It's interesting to see this part to understand each Step of selection.

The last OUTPUT is the final model. Becareful with ths model, it is an informative model. You have to think about logical of the model, possible collinearity betwven variables. It is just an help for the statistician when there are lots of variables.

I finally recommend you to test two types of variables selection and also to read the PDF related to GLIMMIX macro.

If you use this macro for work to be published please use the

following citation:

Francais Adrien (2007).

Automatic selection macro of fixed effects with PROC GLIMMIX.

*/

/*CODE*/

```
%macro glimmixstep(table=,selection=backward,n=,level5=,level4=,level3=,level2=,
effect5=intercept,effect4=intercept,effect3=intercept,effect2=intercept,method=MSPL,
ddfm=satterth,varlist=,varcible=,threshold=0.05,type=vc,include=,varclass=);
```

```
%let error=0;
```

```
%let errormodal=0;
```

```
/*some errors messages*/
```

```
%if &table = %then %do;
```

```
    %put "ERROR: You didn't give the work table : Please enter the name of work
table";%let error=1;
```

```
%end;
```

```
%if &varlist = %then %do;
```

```
    %put "ERROR: You didn't give potentially variables to select : Please enter
variables";%let error=1;
```

```
%end;
```

```
%if &varcible = %then %do;
```

```
    %put "ERROR: You didn't give the outcome variable : Please enter binary variables (0
or 1)";%let error=1;
```

```
%end;
```

```
%if &n = or &n<2 or &n>5 %then %do;
```

```
    %put "ERROR: You didn't give the maximal level of study : Please enter number
between 2 and 5";%let error=1;
```

```
%end;
```

```
%if &threshold<=0 or &threshold>0.25 %then %do;
```

```
    %put "ERROR: You didn't give a correct threshold : Please enter number between 0
and 0.25, preferably 0.05";
```

```
    %let error=1;
```

```
%end;
```

```
/*to test if the outcome variable is binary*/
```

```
%put "We test if the outcome variable is binary";
```

```
%if &table ne %then %do;
```

```
    ods output onewayfreqs=aaa(keep=&varcible);
```

```
    title "Test of variable &varcible";
```

```
    proc freq data=&table;tables &varcible;run;
```

```
    data aaaa;set aaa;id=1;run;
```

```
    data aaaaa;set aaaa;by id;retain errormodal 0;
```

```
    if first.id and &varcible ne 0 then errormodal=1;
```

```
    if last.id and &varcible ne 1 then errormodal=1;
```

```
    call symput ('errormodal',errormodal);
```

```
    run;
```

```
%end;
```

```

%if &errormodal = 1 %then %do;
    %put "ERROR: You didn't give a correct outcome variable : Please enter a binary
variable";%let error=1;
%end;

%if &error = 1 %then %do;
    %put "PLEASE CORRECT ERROR(S) DETAILED ABOVE";
%end;

/*if there is no error then we start the macro*/
%if &error ne 1 %then %do;

/*we declare some local variables*/
%let varselec=;%let notconv=0;%let stop=0; %let j=0;

/*we precise random effects in function of the number of levels in the model*/
%if &n=5 %then %do;
    %let random=%str(random &effect2 / subject=&level2 solution type=&type;
random &effect3 / subject=&level3 solution type=&type;
random &effect4 / subject=&level4 solution type=&type;
random &effect5 / subject=&level5 solution type=&type;);
    %let class=%str(class &level5;class &level4;class &level3;class &level2 &varclass;);
%end;
%if &n=4 %then %do;
    %let random=%str(random &effect2 / subject=&level2 solution type=&type;
random &effect3 / subject=&level3 solution type=&type;
random &effect4 / subject=&level4 solution type=&type;);
    %let class=%str(class &level4;class &level3;class &level2 &varclass;);
%end;
%if &n=3 %then %do;
    %let random=%str(random &effect2 / subject=&level2 solution type=&type;
random &effect3 / subject=&level3 solution type=&type;);
    %let class=%str(class &level3;class &level2 &varclass;);
%end;
%if &n=2 %then %do;
    %let random=%str(random &effect2 / subject=&level2 solution type=&type;);
    %let class=%str(class &level2 &varclass;);
%end;

%put "WE SELECT VARIABLES WITH THE &selection PROCEDURE";
%put "GLIMMIX model with a &n.-level model random effects";

/*FIRST APPROCH FORWARD
WE ADD ONE VARIABLE BY STEP*/
%if &selection=forward %then %do;
/*We do a loop corresponding at each step of variables selection*/
    %do %while (&stop=0);
        %let j = %eval(&j+1);
        %put "STEP &j";

```

```

data tested_var;format effect $50. probT 4.2;run; /*we
create an empty table*/
%local i var;
%let i=0;
/*we keep one by one variables which remain to test*/
%do %while(%scan(&varlist,&i+1,%str( )) ne %str( ));
    %let i = %eval(&i+1);
    %let var = %scan(&varlist,&i,%str( ));
/*we run GLIMMIX procedure on variables selected
&varelec : variables already selected in the model
&var : variables which remain to test */
ods output
parameterestimates=parameterestimates(keep=effect probT);
title "Model with the variable &i";
proc glimmix data=&table

method=&method;

&class;
model &varcible (event='1')=

&varelec &var /

solution dist=binary link=logit

ddfm=&ddfm oddsratio ;

&random;
nloptions tech=nrridg;
ods exclude solutionr;
run;
/*we keep p values on each new variable to test*/
data parameterestimates;set

parameterestimates;nobis=1;run;

data parameterestimates;set

parameterestimates;if last.nobis;by nobis;run;

data tested_var;set tested_var

parameterestimates;run;

/*we add at the table all precedent
results*/

%end;

/*we keep the highest p value and then we compare it to the threshold
given in the macro option*/
data gtgt;set tested_var;if probT=. then delete;run;
proc sort data=gtgt;by probT;run;
data apport;set gtgt;by nobis;if first.nobis;run;

data fsdfds;set apport;retain stopper 0;
if probT>&threshold then stopper=1;else stopper=0;
call symput('stop',stopper);
call symput('var',effect);
call symput('thres',probT);
run;

```

```

/*if the variable is not significant, we stop the macro*/
%if &stop=1 %then %do;
    %put "Algorithm tell us to stop looking for new variables in the
model";
    %put "Last variable proposed to be add was &var with a p-value
of &thres";
%end;
%else %do;
    %put "Algorithm tell us to continue looking for new variables in
the model";

    %let newvar=&var;
    %put "The new variable is &newvar";
/*we obtain the list of variables kept certain in the model*/
    %let varselec=&varselec &newvar;
    %put "Variables selected at step &j are &varselec";

/*part to give us the list of variables which remain to test at the
next step j+1*/
    %let varenplus=;
    %let i=0;
    %do %while(%scan(&varlist,&i+1,%str( )) ne %str( ));
        %let i = %eval(&i+1);
        %let variable= %scan(&varlist,&i,%str( ));
        %if &variable ne &newvar %then %do;
            %let varselection=1;
        %end;
        %else %do;
            %let varselection=0;
        %end;
        %if &varselection=1 %then %do;
            %let varenplus= &varenplus &variable ;
        %end;
    %end;
    %end;
/*New list of variables to test at the next step*/
    %let varlist=&varenplus;
    %put "Variables to test at the next step are &varlist";
%end;
%end;
/* FINAL MODEL*/
%if &varselec = %then %do;
    %put "There is no significant covariates at threshold &threshold to explain the
variable &varcible";
    %put "Propose another variables in the model or increase the threshold";
%end;
%else %do;
    title "Final model selected with &selection procedure";
    title2 'Informative model : run also with the other type of selection';
    title3 'Keep a critical sense about the final model';
    ods output solutionR=solutionR
    oddsratios=oddsratios (keep=estimate df lower upper)

```

```

parameterestimates=parameterestimates(keep=estimate df effect probT);
proc glimmix data=&table method=&method;
&class;
model &varcible (event='1')=&include &varselec/
solution dist=binary link=logit ddfm=&ddfm oddsratio ;
&random;nloptions tech=nrridg;run;
%put "Final variables selected at the final step &j with the &selection selection
are &varselec";
%put "The final model is available in the Results section";

/*fixed effects*/
data oddsratios2;set oddsratios;rename estimate=oddsratios;format lower upper 5.2;run;
data parameterestimates2;set parameterestimates;
if effect="Intercept" then delete;oddsratios=round(exp(estimate),.001);run;

/*we merge oddsratio results and parameters estimates*/
data effects;
merge parameterestimates2 oddsratios2;
run;

/*we add the value of Intercept*/
data intercept;set parameterestimates;if effect="Intercept" then output;run;

/*we obtain the final table with all fixed effects*/
data fixedeffects;
set intercept effects;
if oddsratios ne . then do;
confidence_interval=['!!compress(round(lower,.01))!!' ;
'!!compress(round(upper,.01))!!'];
end;
drop lower upper;
run;

title "Summary of fixed effects";
proc print data=fixedeffects;run;
%put "You can see the Results section to consult fixed effects";
%put "Summary of fixed effects are resumed in the table FIXEFFECTS";

/*random effects*/
data randomeffects;set solutionr;if probT<=&threshold;run;
title "Summary of random effects";
proc print data=randomeffects;run;
%put "You can see the Results section to consult random effects";
%put "Summary of significant random effects are resumed in the table RANDEFFECTS";
%put "All random effects are available in the table SOLUTIONR and in the final model
results";

%end;

```

```

/*we delete all work tables*/
title ";
proc datasets ;
delete apport fsdfds gtgt fsdf tested_var
intercept effects oddsratios oddsratios2 parameterestimates2 parameterestimates;
run;
quit;
%end;

/*SECOND APPROCH BACKWARD
WE REMOVE ONE VARIABLE BY STEP*/
%if &selection=backward %then %do;
/*at the first step, we run model wih all variables proposed and at each step, we remove the
worst variable*/
                %do %while (&stop=0);
                        %let j = %eval(&j+1);
                                %put "STEP &j";
                                        title "Model at step &j";
                                                ods output
parameterestimates=parameterestimates(keep=effect probT);
proc glimmix data=&table
method=&method;
                                &class;
                                model &varcible (event='1')=
&varlist /
                                solution dist=binary link=logit
ddfm=&ddfm oddsratio ;
                                &random;
                                nloptions tech=nrridg;
                                ods exclude solutionr;
                                run;
/*we remove the worst variable thanks to the p value*/
                data parameterestimates;set parameterestimates;nobis=1;call
symput('num',nobis);run;
/*if algorithm did not converge*/
                %if &num= %then %do;
                        %let stop=1;
                        %let notconv=1;
                        %put "Algorithm did not converge";
                        %put "Look at variables proposed in the model";
                %end;
/*else*/
                %if &stop=0 %then %do;
                        proc sort data=parameterestimates;by descending probT;run;
                        data apport;set parameterestimates;by nobis;if first.nobis;run;

                        data fsdfds;set apport;retain stopper 0;

```

```

        if probT<&threshold then stopper=1;else stopper=0;
        call symput('stop',stopper);
        call symput('var',effect);
        call symput('thres',probT);
        run;
    %end;

    %if &stop=1 and &notconv=0 %then %do;
        %put "Algorithm tell us to stop removing for new variables in
the model";
        %put "Last variable proposed to be remove was &var with a p-
value of &thres";
    %end;
    %if &stop=0 and &notconv=0 %then %do;
        %put "Algorithm tell us to continue removing for variables in
the model";

        %let newvar=&var;
        %put "The variable dropped is &newvar";

    /*part to give us the list of variables which remain to test at the next step j+1*/
        %let varenplus=;
        %let i=0;
        %do %while(%scan(&varlist,&i+1,%str( )) ne %str( ));
            %let i = %eval(&i+1);
            %let variable= %scan(&varlist,&i,%str( ));
            %if &variable ne &newvar %then %do;
                %let varselection=1;
            %end;
            %else %do;
                %let varselection=0;
            %end;
            %if &varselection=1 %then %do;
                %let varenplus= &varenplus &variable ;
            %end;
        %end;
        %end;
        /*New list of variables to test at the next step*/
        %let varlist=&varenplus;
        /*we stop algorithm if there is no variable significant*/
        %if &varlist= %then %do;%let stop=1;%end;
        %put "Variables to test at the next step are &varlist";
    %end;

    %end;

    /* FINAL MODEL*/
    /*if the model converged*/
    %if &notconv=0 %then %do;
        %if &varlist = %then %do;
            %put "There is no significant covariates at threshold &threshold to explain the
variable &varcible";

```

```

        %put "Propose another variables in the model or increase the threshold";
    %end;
    %else %do;
        title "Final model selected with &selection procedure";
        title2 'Informative model : run also with the other type of selection';
        title3 'Keep a critical sense about the final model';
        ods output solutionR=solutionR
        oddsratios=oddsratios (keep=estimate df lower upper)
        parameterestimates=parameterestimates(keep=estimate df effect probT);
        proc glimmix data=&table method=&method;
            &class;
            model &varcible (event='1')=&include &varlist/
            solution dist=binary link=logit ddfm=&ddfm oddsratio ;
            &random;
            nloptions tech=nrridg;
        run;
        %put "Final variables selected at the final step &j with the &selection selection
are &varlist";
        %put "The final model is available in the Results section";

ods output close;

/*fixed effects*/
data oddsratios2;set oddsratios;rename estimate=oddsratios;format lower upper 5.2;run;
data parameterestimates2;set parameterestimates;
if effect="Intercept" then delete;oddsratios=round(exp(estimate),.001);run;

/*we merge oddsratio results and parameters estimates*/
data effects;merge parameterestimates2 oddsratios2;run;

/*we add the value of Intercept*/
data intercept;set parameterestimates;if effect="Intercept" then output;run;

/*we obtain the final table with all fixed effects*/
data fixedeffects;
set intercept effects;
if oddsratios ne . then do;
    confidence_interval='[!!compress(round(lower,.01))!!' ;
    '!!compress(round(upper,.01))!!]';
end;
drop lower upper;
run;

title "Summary of fixed effects";
proc print data=fixedeffects;run;
%put "You can see the Results section to consult fixed effects";
%put "Summary of fixed effects are resumed in the table FIXEDEFECTS";

/*random effects*/

```

```
data randomeffects;set solutionr;if probT<=&threshold;run;
title "Summary of random effects";
proc print data=randomeffects;run;
%put "You can see the Results section to consult random effects";
%put "Summary of significant random effects are resumed in the table RANOMEFFECTS";
%put "All random effects are available in the table SOLUTIONR and in the final model
results";
%end;

/*we delete all work tables*/
%end;
title ";
proc datasets ;
delete aaaaa aaaa aaa apport fsdfds parameterestimates tgtg
intercept effects oddsratios oddsratios2 parameterestimates2 ;
run;

%end;

%end;
%mend;
```