Paper SA16-2012

Model Selection Using Recursive Macro

Enhancements to R^2 Selection in PROC REG

Anca M Tilea, University of Michigan, Ann Arbor MI
Philip L Francis III, Eastern Michigan University, Ypsilanti, MI
Brenda W Gillespie, PhD, University of Michigan, Ann Arbor, MI
Rajiv Saran, MD, University of Michigan, Ann Arbor, MI

ABSTRACT

“Beauty is in the eye of the beholder”, the saying goes. Similarly goes with model selection methods, “Model selection is in the eye [hand] of the statistician”. There are as many methods as statisticians. Some are preferred, and some are rather tedious but lead to the “best” model. I recently had to perform model selection for a regression analysis, using the R^2 statistic to get the BEST SUBSETS models. In the first step, starting with all possible covariates as potential candidates, I was going to use a PROC REG, with selection = RSQUARE. In the second step, I was patiently going to run through all the BEST SUBSETS models and flag the ones that had all covariates significant in the model (say, using $\alpha = 0.05$). Thirdly, I would “compile” all the covariates that contributed to all-significant models, and re-run the PROC REG, since we know that starting with all possible covariates in the model may shrink the sample size significantly. Re-run step 2, if the number of covariates is still large, otherwise select the covariates with significant p-values, and finalize the model. Running the model using selection = RSQUARE is manageable, but the prospect of manually running each model, after checking the output file for significant p-values and repeating the process, was a bit disheartening. Fortunately, there are programmatic tools waiting to be employed for an extremely quick result. This is when a couple of ODS OUTPUT datasets and PROC SQL came in very handy.
This paper aims to add to existing SAS® methods, specifically the R^2 selection method with the BESTSUBSETS option of PROC REG, with the help of SAS® macro language. The SQL procedure, CALL SYMPUT, SCAN function and various Output Delivery System (ODS) statements are used to create the macro that will help the statistician make a [more] informed modeling decision.

INTRODUCTION

The REG procedure in SAS® offers nine options for model selection [1]. There is a vast literature on the pros and cons of each of these selection methods [2]; with complete data, the consensus is that, the best method is that of best subset selection, with the R^2 as a popular selection criterion. Briefly, R^2 is the proportion of the variability of the dependent variable that is explained by the independent variables. R^2 ranges from 0 to 1 and is a function of the total sum of squares (SST), and the error sum of square (SSE). The equation for the R^2 is

$$R^2 = 1 - {}^{SSE}/_{SST}$$

When the SELECTION = RSQUARE option is used in SAS® PROC REG, R^2 is calculated for all possible subset models. . A higher R^2 is considered better, taking into account the magnitude of R^2 increase and the significance of each variable. However, even this method has one major drawback: in the presence of missing data, the “complete case” sample size is used, which may be much smaller that the “full” sample size available with the optimal covariates. This loss of information in the model selection phase may result in choosing a less optimal model.

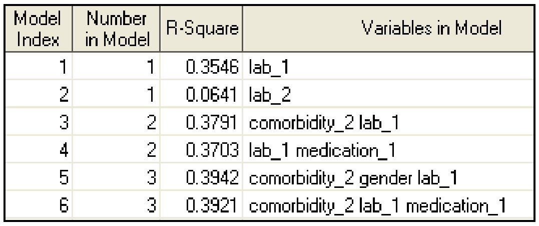
OVERVIEW

The purpose of this paper is to provide a SAS® macro to help the analyst use the R^2 selection method in the presence of missing data. Using the ODS statement, a BEST SUBSETS SAS® data set can be created. An example data set produced by the ODS statement is shown in Figure 1, with the following components:

1. *Model Index* provides the number of models produced by the procedure
2. *Number in Model* provides the number of covariates in each model
3. *R-Square* provides the R^2 for each model
4. *Variables in Model* is the list of the covariates that were selected for each model

The estimates and the associated p-values for the BEST SUBSETS models are not provided in this option.

```
PROC REG DATA = &my_data.;
    MODEL &outcome. = &covariates./SELECTION = rsquare BEST = 2;
    ODS OUTPUT SubsetSelSummary = work.Best_subsets;
RUN; QUIT;
```



Model Index	Number in Model	R-Square	Variables in Model
1	1	0.3546	lab_1
2	1	0.0641	lab_2
3	2	0.3791	comorbidity_2 lab_1
4	2	0.3703	lab_1 medication_1
5	3	0.3942	comorbidity_2 gender lab_1
6	3	0.3921	comorbidity_2 lab_1 medication_1

Figure 1. Best Subsets sample data set

The SELECTION = RSQUARE in PROC REG, for a given set of candidates covariates, will only use the cases with data available on ALL covariates, or the complete-case data. Thus, the sample size for the models saved in the BEST SUBSETS data set may be significantly smaller than the original sample size, depending on the missingness of the data.

The proposed macro, ***%model_select****,* will enhance the R^2 model selection with BEST SUBSETS option by adding:

- model specific sample size information
- model specific estimates & p-values
- list of ALL significant models

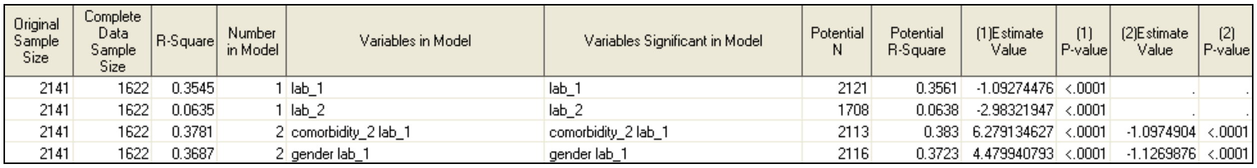
As a result of the macro, three SAS® data sets will be outputted. All three data sets start from the BEST SUBSETS data set (Figure 1) and various components are appended, for three scenarios.
The three scenarios are:

1. Complete-case data are used to calculate the estimates and associated p-values for each model in the BEST SUBSETS list, and appended to the original BEST SUBSETS data set. I.e., each model in the BEST SUBSETS list will be restricted to no missing data for ALL candidate covariates.
2. Full data are used to calculate the estimates and associated p-values for each model in the BEST SUBSETS list. I.e., for the one-covariate model PROC REG will restrict the data to be non-missing for that one covariate. For the 3-covariate model, PROC REG will restrict the data to be non-missing for those specific three covariates. Thus, the sample size may change for each individual model in the BEST SUBSETS list. The “potential” sample size and the “potential R^2” will be appended to the BEST SUBSETS data set, as well as the estimates and the p-values (Figure 2).
3. In the full-data models from step 2, each variable is assessed for significance. Eliminate the least significant covariate one at the time, until all covariates are significant (i.e., manual backwards selection), the “final”/significant model is appended to the original BEST SUBSETS data set, with the sample size information, R^2, and the estimates and p-values

Sample results from the ***%model_select*** macro are shown in Figure 2 with the following key components:

1. *Original Sample Size*: the original size of the data set, with missing data included
2. *Complete-case N*: sample size for the BEST SUBSETS models (complete-case data including all covariates of interest)
3. *R-Square*: Model-based R^2 for the BEST SUBSETS models
4. *Number in Model*: total number of covariates in each model
5. *Variables in Model*: list of the covariates used in each model
6. *Variables Significant in the Model*: significant covariates from each model
7. *Potential N*: complete-case sample size for a model using a subset of variables. Also called the full-data sample size. Potential N ≥ Complete-case N.
8. *Potential R-square*: Model-based R^2 for models based on Potential N.

The rest of the columns represent the individual *estimates* and *p-values*



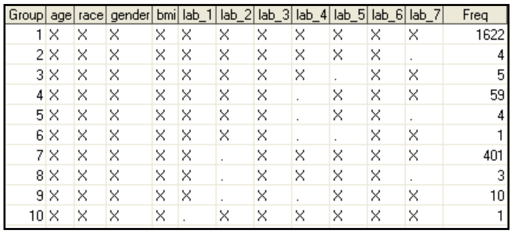
Original Sample Size	Complete Data Sample Size	R-Square	Number in Model	Variables in Model	Variables Significant in Model	Potential N	Potential R-Square	(1)Estimate Value	(1) P-value	(2)Estimate Value	(2) P-value
2141	1622	0.3545	1	lab_1	lab_1	2121	0.3561	-1.09274476	<.0001	.	.
2141	1622	0.0635	1	lab_2	lab_2	1708	0.0638	-2.98321947	<.0001	.	.
2141	1622	0.3781	2	comorbidity_2 lab_1	comorbidity_2 lab_1	2113	0.383	6.279134627	<.0001	-1.0974904	<.0001
2141	1622	0.3687	2	gender lab_1	gender lab_1	2116	0.3723	4.479940793	<.0001	-1.1269876	<.0001

Figure 2. Sample results from the % model_select macro

METHODS

There are many possible models with various complete-case subsets in the presence of missing data; therefore, we start by exploring the missing data patterns. A useful tool for this purpose is the MI procedure, which displays the missing data patterns for a given set of variables. An example of the SAS® code and output from PROC MI is given in Figure 3. For the given data, there are 1,622 records with non-missing values for all variables (*complete cases*), indicated by an X. For example, variable *lab_2* has the most missing data, with 401 + 3 + 10 = 414 missing values. Thus, the sample size will shrink by 414 for all models including *lab_2*. Note that this procedure will not run if there is no missing data for all records, or if there are records that have ALL missing variables.

```
PROC MI DATA = &my_data.;
    VAR age race gender bmi lab_1 lab_2 lab_3 lab_4 lab_5 lab_6 lab_7;
ODS OUTPUT MissPattern = missing;
RUN;
```



Group	age	race	gender	bmi	lab_1	lab_2	lab_3	lab_4	lab_5	lab_6	lab_7	Freq
1	X	X	X	X	X	X	X	X	X	X	X	1622
2	X	X	X	X	X	X	X	X	X	X	.	4
3	X	X	X	X	X	X	X	X	.	X	X	5
4	X	X	X	X	X	X	X	.	X	X	X	59
5	X	X	X	X	X	X	X	.	X	X	.	4
6	X	X	X	X	X	X	X	.	.	X	X	1
7	X	X	X	X	X	.	X	X	X	X	X	401
8	X	X	X	X	X	.	X	X	X	X	.	3
9	X	X	X	X	X	.	X	.	X	X	X	10
10	X	X	X	X	.	X	X	X	X	X	X	1

Figure 3. Missing patterns data set

To overcome the problem of modeling in the presence of missing data, we propose the macro, ***%model_select*** – based on the BEST SUBSETS algorithm – which sequentially increases the sample size by excluding non-significant variables. The proposed macro is based on linear regression (PROC REG, SELECTION = RSQUARE), but it is easily modified for logistic (PROC LOGISTIC, SELECTION = SCORE) and Cox regression (PROC PHREG, SELECTION = SCORE).

Two modeling paths are possible when using this macro: BEST SUBSETS with elimination of never-significant variables to possibly increase sample size, and BEST SUBSETS followed by a step-down of each candidate model. It is difficult to test every variable in a model with all possible sample sizes. Thus, neither strategy provides an exhaustive search of all possible models; the "best" model may not be found by either algorithm. However, the two modeling paths increase the chances of finding the best model candidates.

The steps of the macro are listed below, as illustrated in Figure 4:

METHOD #1: BEST SUBSETS

1. Perform a BEST SUBSETS selection to get the list of candidate models with complete-case data.
2. For each model in the BEST SUBSETS list, calculate the estimates and associated p-values based on the complete-case sample size, as well as on the full (potential) sample size.
3. Examine the macro output for variables that are never significant, and consider the difference between complete-case and full sample sizes for the optimal models.
    - Eliminate the covariates that are never significant in the complete-case analysis and/or have substantial missing data. This requires investigator judgment.
    - A limitation to this approach is that the user will "flag" the significant covariates based on a specific sample. If one were to eliminate any given covariate and re-run the entire process, potentially different covariates may be flagged as significant or not.
4. Re-run from step 1 if desired.

METHOD #2: BEST SUBSETS FOLLOWED BY STEP-DOWN

1. Perform a BEST SUBSETS selection to get the list of candidate models with complete-case data.
2. For each model in the BEST SUBSETS list, calculate the estimates and associated p-values based on the full sample size.
3. Eliminate the non-significant covariates one at the time, until all covariates are significant (i.e., manual backwards selection).
   - The full potential sample size is used for each model.
   - This set of step-down models is saved in a SAS dataset for visual examination.

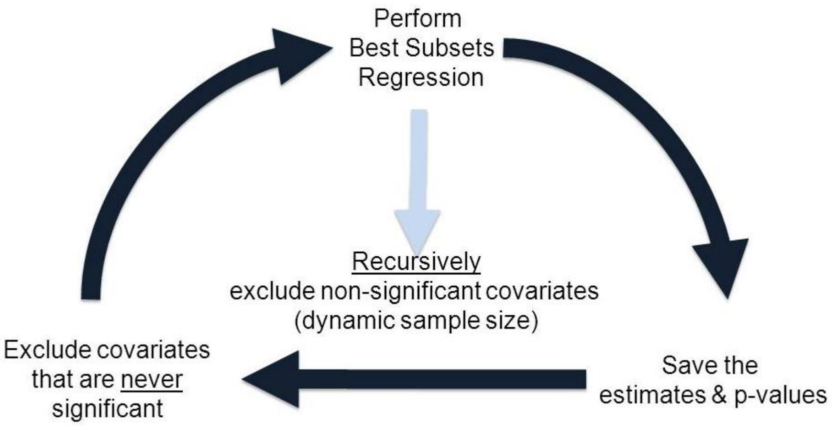


Figure 4. Steps of the %model_select macro

SAS® CODE TO RUN %MODEL_SELECT MACRO

The complete SAS® code and sample data are available at this website
http://www-personal.umich.edu/~atilea/MWSUG and www.sasCommunity.org

Define the ***%model_select*** macro using SAS® `INCLUDE` statement.

```
%INCLUDE "\\...\MWSUG\model_select_macro.sas";
```

Use the macro `%LET` statement to define the potential covariates list. For example,

```
%LET var_list = age race gender bmi comorbidity_1 comorbidity_2 comorbidity_3;
```

The ***%model_select*** macro takes four parameters:

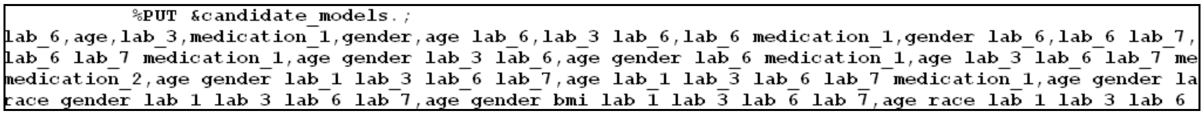
❶ Data set name: user will provide the name of the data set.
❷ Outcome name: user will provide the name of the outcome variable of interest.
❸ Covariates list: user provides the covariate list.
❹ Best option: user will provide the number of models in each subset selection.

```
                       ❶               ❷                  ❸          ❹
%model_select(my_data=my.mwsug,outcome=my_outcome,covariates=var_list.,best=2);
```

HIGHLIGHTS FROM %MODEL_SELECT MACRO

Save the models from the BEST SUBSETS output data set in a string:

```
PROC SQL NOPRINT;
    SELECT VarsInModel
    INTO: candidate_models SEPARATED BY ","
    FROM work.Best_subsets;
QUIT;
%PUT &candidate_models.; /* print the result */
```



```
          %PUT &candidate_models.;
lab_6,age,lab_3,medication_1,gender,age lab_6,lab_3 lab_6,lab_6 medication_1,gender lab_6,lab_6 lab_7,
lab_6 lab_7 medication_1,age gender lab_3 lab_6,age gender lab_6 medication_1,age lab_3 lab_6 lab_7 me
medication_2,age gender lab_1 lab_3 lab_6 lab_7,age lab_1 lab_3 lab_6 lab_7 medication_1,age gender la
race gender lab 1 lab 3 lab 6 lab 7,age gender bmi lab 1 lab 3 lab 6 lab 7,age race lab 1 lab 3 lab 6
```

Figure 5. Candidate models, separated by commas, as a macro variable

Iteratively, DO-loop through each of the models in the *&candidate_models* variable, and perform the following steps:

Step 1: Run the ith regression on the complete-case data and save the estimates, p-values, and the model R^2 using several ODS statements. This will be a "long", or stacked, data set with a row for each covariate. The *%reg* macro will be called.

Step 2: For the ith model, arrange the estimates and p-values from long-to-wide. Create an indicator of significance for each p-value. The *%wide* macro will be called.

Step 3: Combine sample size information, model fit (given by the R^2) and the saved information from Step 2 into one data set for the ith model. The *%combine* macro will be called.

Step 4: Repeat Steps 1, 2 and 3 on the full data set to calculate and save the "potential" sample size, "potential" model R^2, the estimates and the associated p-values. The *%reg, %wide,* and *%combine* macros will be called.

Step 5: Recursively remove the least significant variable from each of the models from Step 4 until all remaining variables are significant. The *%trim_down* macro will be called.

If there is a large difference between the complete-case and full sample size among the optimal models, we may re-run Steps 1 through 5, this time excluding the covariates that are never significant.

To perform the steps listed in the DO-loop, we use several macros, which are described below.

The ***%reg*** macro will run a PROC REG on a specific data set. It takes three arguments:

- where = gives the option to condition on no missing data or, if left blank, no condition applies
- data = the data set name used for the ODS data set output. (If "data = complete" then a complete_n_&model. data set will be created. Similarly, complete_fit_&model., and complete_estim_&model. data sets.)
- var_list = the list of covariates to be used, i.e., when used for the complete or full data, the var_list takes value of the initial list of covariates. But when the *%trim_down* macro is used, the var_list will take the value of the variables that are significant in the final model of the recursion.

```
%macro reg(where = , data = , var_list = );
       PROC REG DATA = &my_data.;
                     WHERE &where.;
                     MODEL outcome = &var_list.;
                     ODS OUTPUT NObs = &data._n_&model. /*-- sample size --*/
                            (WHERE = (label = "Number of Observations Used")
                             KEEP = N label model RENAME = (n = potential_n));
                     ODS OUTPUT FitStatistics = &data._fit_&model.
                            (WHERE = (Label2 = "R-Square")
                             KEEP = Label2 cvalue2 model);
                     ODS OUTPUT ParameterEstimates = &data._estim_&model.
                            (KEEP = Variable estimate Probt model);
                     /* this second estimates output statement will be */
                     /* used in the %trim_down macro */
                     ODS OUTPUT ParameterEstimates = estim_&model.
                            (KEEP = Variable Probt);
              RUN;
%mend reg;
```

The ***%wide*** macro will transform the data from long to wide using ARRAY and DO statements. It takes two arguments:

- data_in = the estimates data set(s): from the complete, full or trimmed-down model. This is the long (or stacked) data.

- data_out = the wide output data set.

```
%macro wide(data_in = , data_out = );
       DATA &data_out._pval_&model.
              (DROP = i covar: sig_name: probt variable estimate);
             /* max_covar is the total number of covariates */
             ARRAY pvals {&max_covar.};
             ARRAY not_sign {&max_covar.};
             ARRAY estim {&max_covar.};
             ARRAY covar {&max_covar.} $20;
             ARRAY sig_name {&max_covar.} $20;
             DO i = 1 TO &max_covar. UNTIL(last.model);
             SET &data_in.(where = (variable ne "Intercept"));
                           BY model;
                           covar(i) = variable;
                           estim(i) = estimate;
                           pvals(i) = probt;
                           not_sign(i) = (probt > 0.05);
                           /* if a covariate is significant, save its name */
                           IF not_sign(i) = 0 THEN sig_name(i) = covar(i);
             END;
             /* create the Modelindex variable to merge with BEST SUBSETS */
                    modelIndex = &model.;
                    /* list the covariates that are   significant in the model*/
             VariablesInModel = catx("", of sig_name1-sig_name&max_covar.);
             LABEL VariablesInModel = "Variables Significant in Model";
       RUN;
%mend wide;
```

The ***%combine*** macro combines the estimates, p-values, sample size, and R^2 for each model. It takes two arguments:

- data_out = the output data set
- merge_set = the list of the data sets to be merged (i.e., the N, the fit, the estimates and P-values)

```
%macro combine(data_out = , merge_set =   );
       DATA &data_out.(DROP = label: cValue2 model);
             MERGE &merge_set.;
             BY model;
             modelIndex = &model.;
             potential_r = cvalue2 * 1;/*-- change to numeric --*/
             LABEL potential_r = "Potential R" potential_N = "Potential N";
             model_sign = (SUM ( OF not_sign1 -- not_sign&max_covar.) = 0);
             LABEL model_sign = "Model significant overall ? (Yes = 1,No = 0)";
             complete_n = &complete_n.; /*-- information from BEST SUBSETS --*/
             LABEL complete_n = "Complete Data Sample Size";
             total_n = &total_n.; /*-- information from BEST SUBSETS --*/
             LABEL total_n = "Original Sample Size";
       RUN;
%mend combine;
```

The ***%trim_down*** macro recursively trims-down each model until a significant model is reached. It only takes one argument, the model number (which ranges from 1 to (p-1)*5 + 1), where *p* is the number of candidate covariates. Several PROC SQL statements are used. The data set outputted from the *%reg* macro "estim_&model." is used.

```
PROC SQL NOPRINT
    SELECT Variable INTO: all_var_list SEPARATED BY " "
FROM work.estim_&model.
/* this data set comes from the %reg macro */
WHERE variable ne "Intercept"

HAVING probt < MAX (probt) | SUM((probt > 0.05)) = 0
```

A	B	A OR B
False	False	False
True	False	True
False	True	True
True	True	True

Table 1. Logical OR

The condition above executes a logical OR (inclusive-OR) as shown in Table 1:

1. If all covariates are not significant, it selects all but the covariate that has the highest (max) p-value. Thus, probt < MAX (probt) resolves to 1 (TRUE), and SUM ((probt > 0.05)) = 0 resolves to 0 (FALSE).
2. If some covariates are significant and some are not, it will select all covariates but the one with the max p-value. Similar to the scenario above, probt < MAX (probt) resolves to 1 (TRUE) and

   SUM ((probt > 0.05)) = 0 resolves to 0 (FALSE).
3. When all covariates are significant it will select all but the covariate that has the max p-value. Thus, probt < MAX (probt) resolves to 1 (TRUE) and SUM ((probt > 0.05)) = 0 resolves to 1 (TRUE). Because the second condition is true it will not run the model (as you shall see further down).
4. The case where both conditions are FALSE doesn't apply.

```
SELECT Variable INTO: one_var_list SEPARATED BY " "
FROM work.estim_&model.
WHERE variable ne "Intercept"
HAVING probt > 0.05
```

The *one_var_list* macro variable will only be used when there are single-covariate models that are not significant.

```
SELECT SUM((probt > 0.05)) INTO: sum_not_sign
FROM work.estim_&model.
```

We SUM over all the indicators that the probability is > 0.05 and save the information in a macro variable.
The macro calls itself if there are still variables to be removed. For single-covariate models, no trimming is necessary, so one regression is executed and the estimates are saved.

```
%macro trim_down(model);
/* initialize variables to be used */
%LET all_var_list = ;
%LET one_var_list = ;
%LET sum_not_sign = ;
              /* using the estimates saved from the model on the FULL data */
              /* select the covariates and save them in a macro variable */
              /* called all_var_list. We do not include the Intercept*/
              PROC SQL NOPRINT;
                     SELECT Variable INTO: all_var_list SEPARATED BY " "
                     FROM work.estim_&model.
                     WHERE variable ne "Intercept"
                     HAVING probt < MAX(probt) | SUM((probt > 0.05)) = 0;

                     SELECT Variable INTO: one_var_list SEPARATED BY " "
                     FROM work.estim_&model.
                     WHERE variable ne "Intercept"
                     HAVING probt > 0.05;

                     SELECT SUM((probt > 0.05)) INTO: sum_not_sign
                     FROM work.estim_&model.;
              QUIT;
              /* if the list is not empty, execute the regression
                     WHILE there is at least one not significant covariate
                     (information given by sum_not_sign macro) */
              %IF &all_var_list. ne  %THEN %DO;
                     %IF (%SCAN("&sum_not_sign",1," ") ne 0 ) %THEN %DO;
                     PROC REG DATA = &my_data.;
                            MODEL outcome = %scan("&all_var_list.", 1, ",");
                     ODS OUTPUT ParameterEstimates = estim_&model.;
                     RUN;QUIT;
                     %trim_down(&model.); /* recursively call the macro */
                     %END;
              %END;
              /* if we have a one-covariate model, execute the regression */
              %ELSE %DO;
```

```
                    PROC REG DATA = &my_data.;
                           MODEL outcome = %scan("&one_var_list.", 1, ",");
                    ODS OUTPUT ParameterEstimates = estim_&model.;
                    RUN;QUIT;
                    %END;
%mend trim_down;
```

The ***%clean*** macro uses two arguments:

- data = data set name to specify if this is from full, complete or trimmed data models
- selection = the name of the final data set

The final data sets are created in this macro. The macro starts by setting all of the (p-1)*5 + 1 data sets that were outputted in the DO-loop. Each observation represents a model. Using PROC SORT we sort the data by Modelindex to merge with the original BEST SUBSETS data. In a DATA step, we merge the BEST SUBSETS data set with the data that contain the estimates, p-values, sample size and R^2 for complete-case, full data and trimmed models. We label the variables and finally sort by Modelindex.

```
%macro clean(data = , selection = );
       DATA &data.;
              SET &data.:; /* use : to simplify the SET statement */
       RUN;
       PROC SORT DATA = &data.;BY modelIndex;RUN;
       DATA &selection.(DROP = dependent model control);
       /* use FORMAT statement to re-arrange some of the variables order */
       FORMAT modelIndex total_n complete_N rsquare model_sign varsInModel
       VariablesInModel potential_n potential_r ;
       MERGE best_subsets &data.;
              BY modelIndex;
          %macro label();
              %DO i = 1 %TO &max_covar.;
              LABEL pvals&i. = "(&i.) P-value"
             not_sign&i. = "(&i.)Not-Significant in the model (Yes = 0, No = 1)"
              estim&i. = "(&i.)Estimate Value";
              %END;
          %mend;
          %label();
       RUN;
       PROC SORT DATA = &selection. OUT = my.&selection.;
              BY ModelIndex;
       RUN;
%mend clean;
```

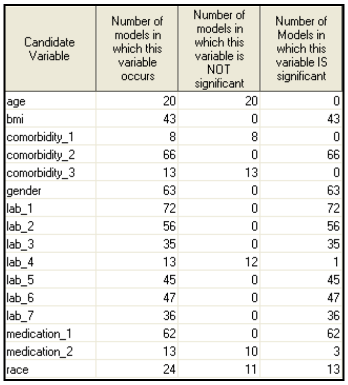
The ***%flag*** macro uses two arguments:

- data_in = input data set
- data_out = output data set

A data step creates a data set that will “stack” all the estimates and p-value data sets that were outputted from each regression.

Using the SELECT statement in PROC SQL on the “stack”-ed data set, we create a data set containing the variables shown below in Figure 6. We count how many times each of the potential covariates occurs in any of the BEST SUBSETS models. We save this information in a macro variable, called *n_var_model*. We save in a macro variable, *n_not_sig,* how many times a variable is not significant in a model. Finally, we take the difference of the two macro variables to determine the number of models in which a given covariate is significant.

```
%macro flag(data_In = , data_out = );
PROC SQL NOPRINT;
   /* create a permanent data set */
   CREATE TABLE &data_out. AS
   SELECT variable LABEL = "Candidate Variable",
   /* count how many times each covariate occurs */
   COUNT( variable) AS n_var_model
   LABEL = "Number of models in which this variable occurs",
   /* sum over the indicators for p-value > 0.05 */
   SUM((probt > 0.05)) AS n_not_sig
   LABEL =  "Number of models in which this variable is NOT significant",
   /* to find out how many times a variable is significant */
   /* take the difference between how many times a */
   /* variable occurs and how many times is not significant */
   (COUNT( variable) - SUM((probt > 0.05))) AS sig_in_model
    LABEL = "Number of Models in which this variable IS significant"
FROM &data_in.
GROUP BY variable; /* execute all commands above within each variable */
QUIT;
%mend flag;
```



Candidate Variable	Number of models in which this variable occurs	Number of models in which this variable is NOT significant	Number of Models in which this variable IS significant
age	20	20	0
bmi	43	0	43
comorbidity_1	8	8	0
comorbidity_2	66	0	66
comorbidity_3	13	13	0
gender	63	0	63
lab_1	72	0	72
lab_2	56	0	56
lab_3	35	0	35
lab_4	13	12	1
lab_5	45	0	45
lab_6	47	0	47
lab_7	36	0	36
medication_1	62	0	62
medication_2	13	10	3
race	24	11	13

Figure 6. Significant covariates

If the analyst decides to eliminate some of the covariates and re-run the ***%model_select*** macro, she can do so by using the following SAS® code:

- Save a new &var_list macro variable
- Call the *%model_select* macro

```
PROC SQL NOPRINT;
       SELECT Variable INTO: sig_vars_complete SEPARATED BY " "
       FROM sig_vars_complete
       /* select only the significant covariates */
       WHERE sig_in_model ne 0;
QUIT;
%PUT &sig_vars_complete.;
```

Re-run the macro %model_select with var_list = &sig_vars_complete.

CONCLUSION

With the %model_select macro, the task of model selection is made much easier, especially when there are a large number of candidate covariates to choose from. We describe a macro that will present the statistician with a multitude of information before selecting a "best" model. We provide insights on sample size changes as different covariates sets are considered. This macro has been modified for logistic regression and Cox regression and is available on http://www-personal.umich.edu/~atilea/MWSUG and www.sasCommunity.org

REFERENCES

- SAS® 9.22 User's Guide
  http://support.sas.com/documentation/cdl/en/statug/63347/HTML/default/viewer.htm#statug_reg_sect030.htm

- R. R. Hocking. *A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear. Regression*, Biometrics, Vol. 32, No. 1. (Mar., 1976), pp. 1-49.
- SAS® Character Functions http://support.sas.com/publishing/pubcat/chaps/59343.pdf

ACKNOWLEDGMENTS

I am very grateful to MWSUG 2012 for allowing me to share my work with other SAS users. I would also like to thank very much Flannery Campbell, Deanna Chyn, and Sarah Casino for their time and positive feedback. I am very grateful to Brenda Gillespie for her statistical expertise, positive feedback, and time she spent helping me with this paper.

CONTACT INFORMATION

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

Anca M Tilea
University of Michigan, School of Public Health
1415 Washington Heights, Suite 3645 A SPH I
Ann Arbor, MI 48109
734.763.6611
734.763.4004
atilea@med.umich.edu
http://www-personal.umich.edu/~atilea