Paper SA-02

SAS® Model Selection Macros for Complex Survey Data Using PROC SURVEYLOGISTIC/SURVEYREG

Fang Wang, NORC at the University of Chicago, Chicago, IL

Hee-Choon Shin, NORC at the University of Chicago, Chicago, IL

ABSTRACT

Regression model selection is widely used in analyzing survey data. In SAS 9.1, Proc Surveylogistic and Proc Surveyreg were developed for analyzing data from complex surveys (stratified and/or clustered surveys). But neither of them has the capacity of automated model selection. “Manual” coding for each specific task would be time-consuming and very labor intensive. This paper details two major macros to perform automated forward, backward and stepwise model selection for complex survey data: %StepSvylog for logistic regression using Proc Surveylogistic, and %StepSvyreg for linear regression using Proc Surveyreg.

KEY WORDS: forward selection, backward selection, stepwise selection, complex survey, proc surveylogistic, proc surveyreg

1. INTRODUCTION

A large number of variables and observations may be collected for large-scale complex surveys. In the absence of prior knowledge, data analysts may include very many variables in their models at the initial stage of modeling in order to reduce possible model bias. In general, a complicated model including many insignificant variables may result in less predictive power, and it may often be difficult to interpret the results. In these cases, a more parsimonious model becomes desirable in practice. Automated model selection is a powerful tool for researchers to choose explanatory variables with strongest correlation with the response variable.

In SAS 9.1, Proc Surveylogistic and Proc Surveyreg are developed for modeling samples from complex surveys. But neither of them has the function of automated model selection. Existed procedures Proc Logistic, Proc Reg and Proc Glmselect with automated model selection features do not allow users to incorporate survey designs in the regressions. In Proc Logistic, Proc Reg and Proc Glmselect, models are fitted and selected based on the assumption that input samples are collected through simple random sampling. Therefore, standard errors and corresponding p-values could be underestimated or overestimated depending on different survey designs.

Analyzing a stratified sample as if it were a simple random sample would overestimate the standard errors. Analyzing a cluster sample as if it were a simple random sample would usually underestimate the standard errors.

2. METHOD

The SAS macros %StepSvylog and %StepSvyreg presented here can implement forward, backward and stepwise variable selection based on the p-values computed through proc surveylogistic and proc surveyreg. Each of the %StepSvylog macro and %StepSvyreg macro calls four nested macros which 1) scan candidate independent variables, 2) fit models, 3) forward selection macro, and 4) backward elimination macro. The criterions for effects to enter or depart a model are p-values as below:

- &SLENTRY: if the p-value of a candidate effect is less than &SLENTRY, this new effect enters the model.
- &SLSTAY: if the p-value of an existing effect is less than &SLSTAY, the effect is kept in the model, otherwise the effect departs the model.

The four sub-macros called in %StepSvylog are:

1. %ScanVar: read in the explanatory variables from the candidate list
2. %SvyLog: fit the logistic regression models using SAS proc surveylogistic
3. %ForwardLog: implement the forward model selection for logistic models
4. %BackwardLog: the backward model selection for logistic models

The four sub-macros called in %StepSvyreg are:

1. %ScanVar: read in the explanatory variables, the same macro used in %StepSvylog
2. %SvyReg: fit the logistic regression models using SAS proc surveylogistic
3. %ForwardReg: implement the forward model selection for logistic models
4. %BackwardReg: the backward model selection for logistic models

The forward and backward selection macros use ODS procedure to retrieve the data containing the p-values from logistic model or linear model, and make decisions on adding or deleting effects.

The forward selection macro, backward selection macro and main stepwise macro flow charts are presented in figure 1 to 3.

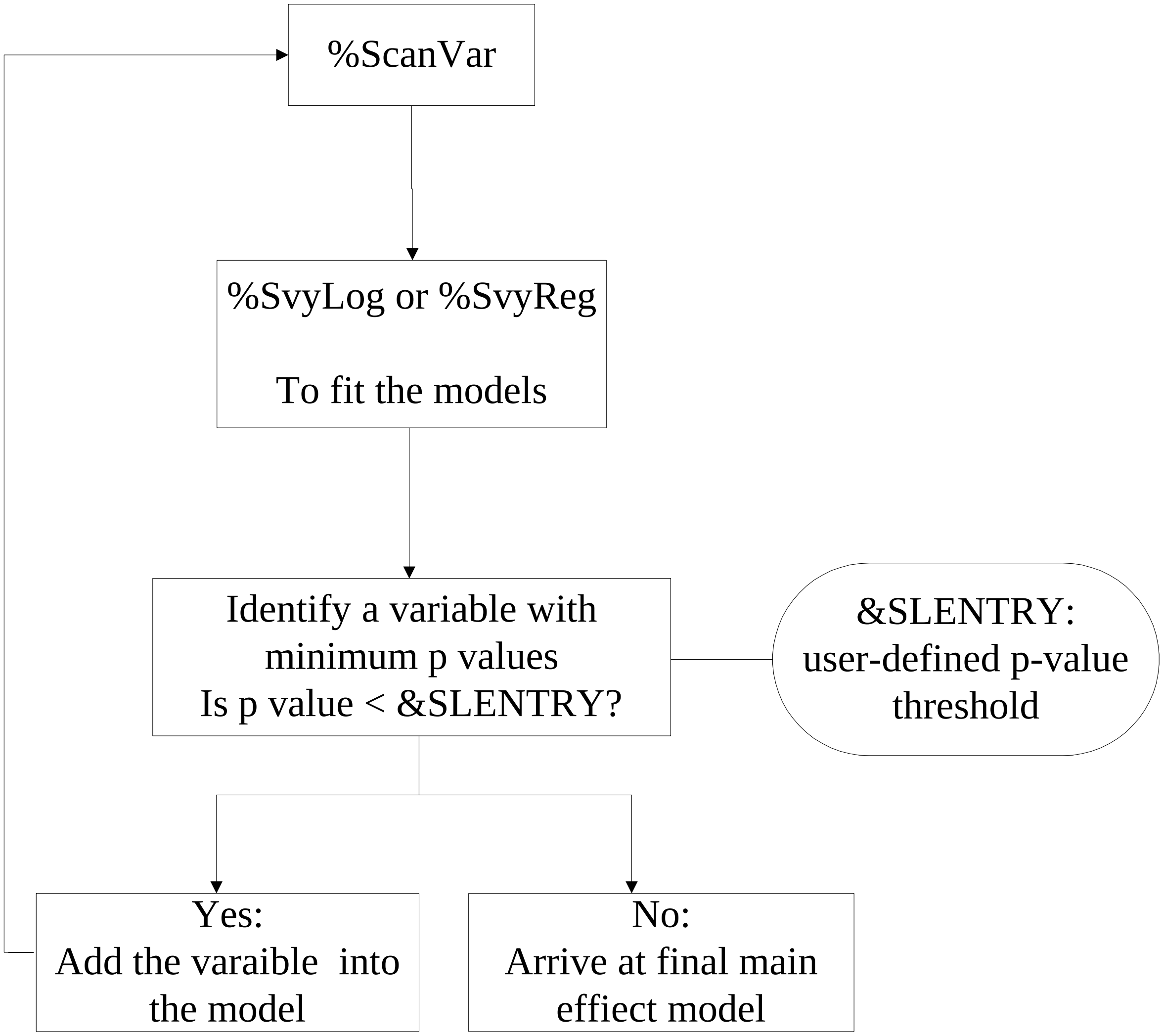
%ScanVar
%SvyLog or %SvyReg
To fit the models
Identify a variable with minimum p values
Is p value < &SLENTRY?
&SLENTRY: user-defined p-value threshold
Yes: Add the varaible into the model
No: Arrive at final main effiect model

Figure 1 %ForwardLog and %ForwardReg macro flow chart

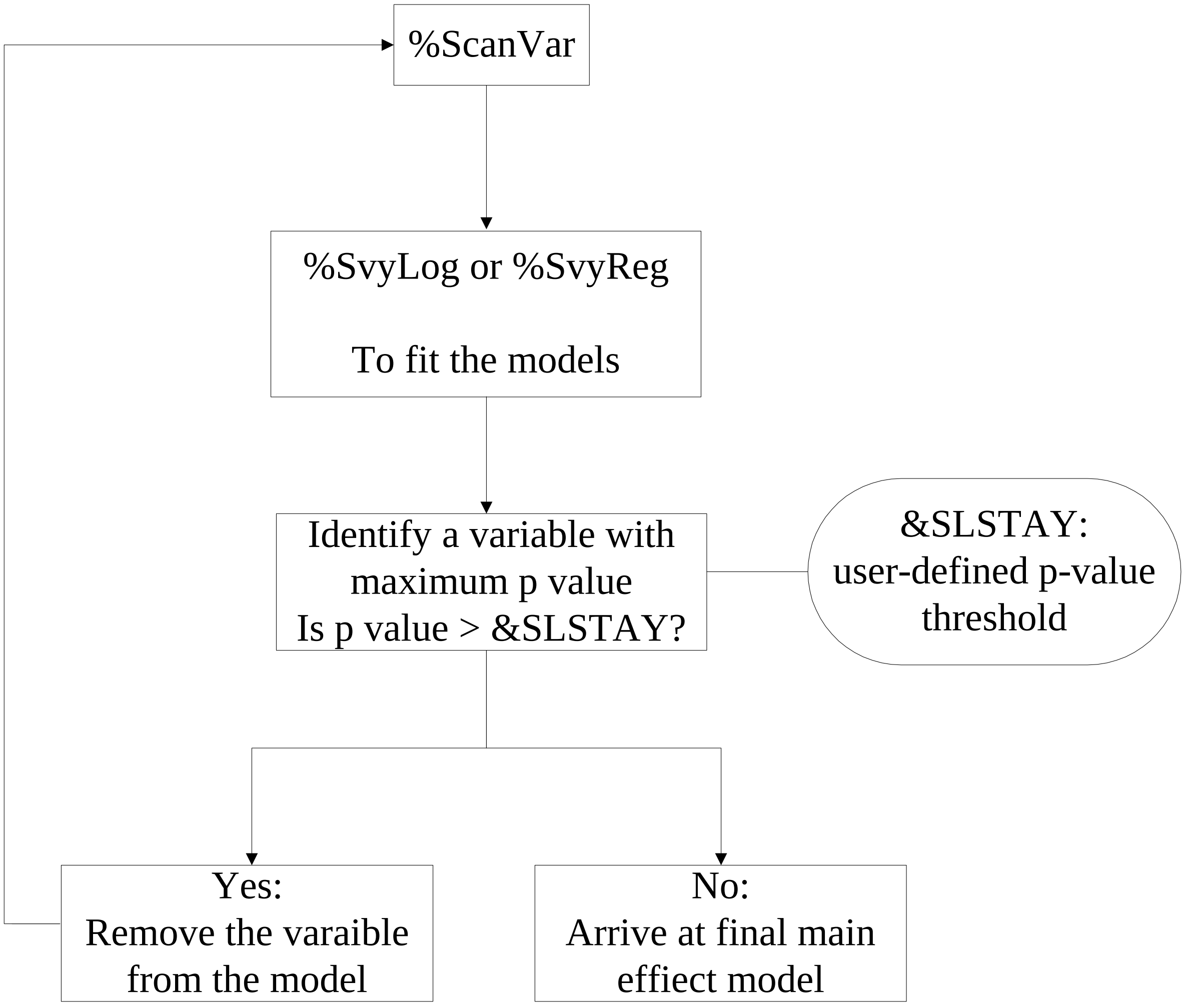
%ScanVar
%SvyLog or %SvyReg
To fit the models
Identify a variable with maximum p value
Is p value > &SLSTAY?
&SLSTAY:
user-defined p-value threshold
Yes:
Remove the varaible from the model
No:
Arrive at final main effiect model

Figure 2 %BackwardLog and %BackwardReg macro flow chart

The main stepwise selection macros can implement user specified forward, backward or stepwise selection calling the forward or/and backward macros. The main macro flow chart is in figure 3

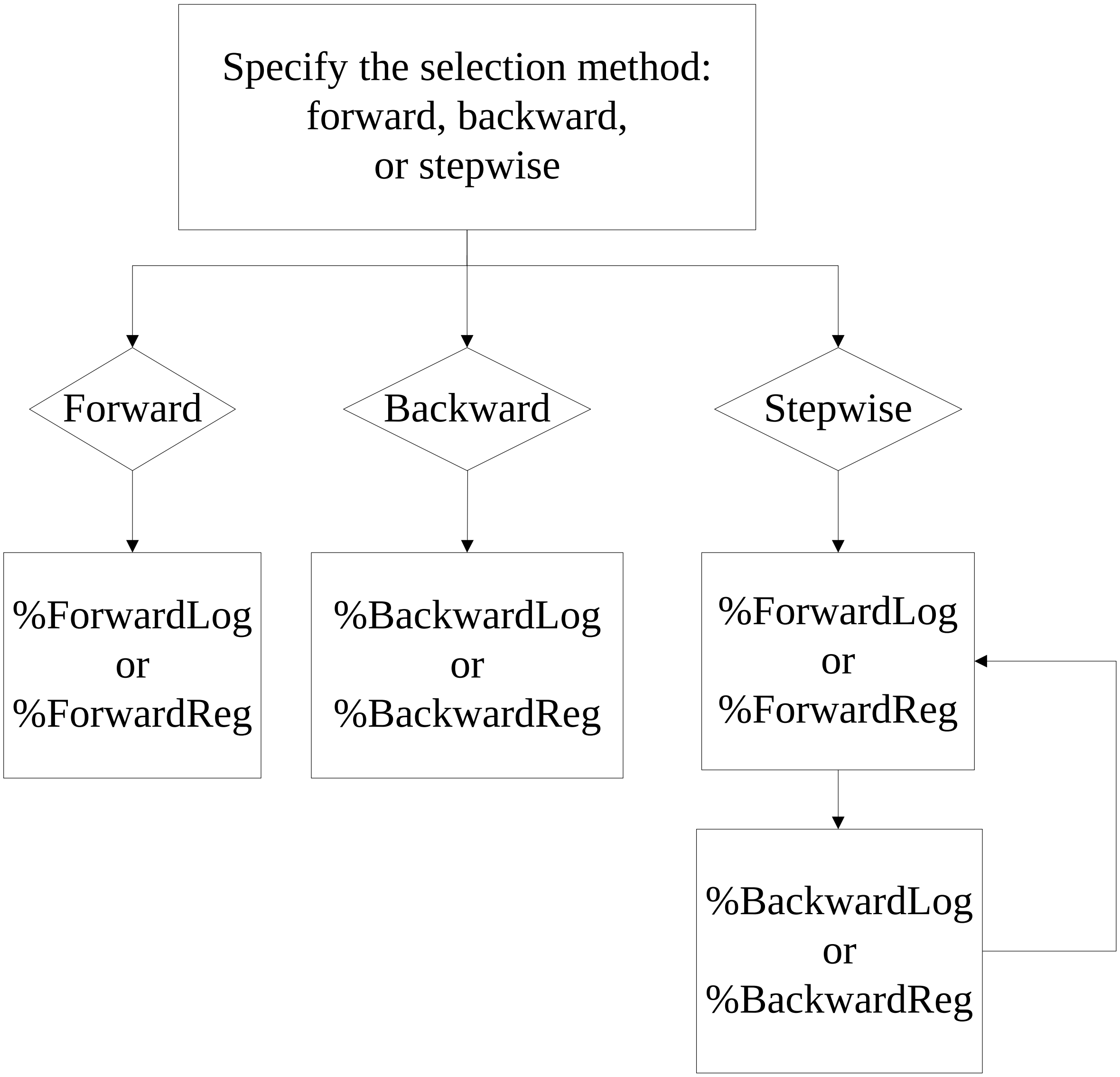

Figure 3 Main macro %StepSvyLog and %StepSvyReg macro flow chart

3. MACRO ARGUMENTS

The layouts of the two macros are:

```
%StepSvyreg(DEP,INTVAR,INTCLASS,DATASET,SLENTRY,SLSTAY,WEIGHT,STRATA,CLUSTER,
METHOD);
```

```
%StepSvylog(DEP,INTVAR,INTCLASS,DATASET,SLENTRY,SLSTAY,WEIGHT,STRATA,CLUSTER,
METHOD);
```

More specific information about the arguments to the macros is given in Table 1. Output tables displaying the final selection results are described in Table 2.

Table 1 Arguments for the macro StepSvyreg and StepSvylog.

Argument	Description	Usage
DEP	dependent variable	Required
INTVAR	all the candidate independent variables	Required
INTCLASS	categorical independent variables among	Optional
DATASET	input dataset	Required
SLENTRY	entry p value	Required
SLSTAY	stay p value	Required
WEIGHT	weight for weighted regression models	Optional
STRATA	complex survey strata variable	Optional
CLUSTER	complex survey cluster variable	Optional
METHOD	selection method:1= forward, 2=backward,	Required

The scripts of the two macros are in the appendix.

Table 2 Description of output datasets

Macro	Selection	METHOD	Output data	Variables in the output data
%StepSvylog	Forward	1	Step_1	Parm: selected covariate name, ProbChisq: p-value
	Backward	2	Step_2	Parm: selected covariate name, ProbChisq: p-value
	Stepwise	3	Step_3	Parm: selected covariate name, ProbChisq: p-value
%StepSvyreg	Forward	1	Step_1	Parm: selected covariate name, ProbF: p-value
	Backward	2	Step_2	Parm: selected covariate name, ProbF: p-value
	Stepwise	3	Step_3	Parm: selected covariate name, ProbF: p-value

4. EXAMPLE

We will work through the use of the two macros with two examples. Implementation of the macros is demonstrated using a dataset from the 2008 National Immunization Survey – Teen (NIS-Teen) Public Use File. The NIS-Teen is the largest survey ever to assess vaccination levels of adolescents 13-17 years of age in the U.S conducted by NORC. The NIS-Teen collects data by interviewing households randomly selected from 56 areas: all 50 States, the District of Columbia, and 5 designated areas for oversampling. The 56 areas are strata of the 2008 NIS-Teen survey. There are no clusters within each stratum. The 2008 NIS-Teen public use file and corresponding codebook can be downloaded here http://www.cdc.gov/nchs/nis/data_files.htm.

The same fourteen demographic variables are used as candidate independent variables in both examples:

1. RACE_K: race of teen with multi-race category.
2. I_HISP_K: is teen Hispanic or Latino?
3. SEX: gender of teen.
4. AGE: age in years of selected teen
5. AGEGRP_M_I: mother's age categories
6. MARITAL: marital status of mother
7. CHILDNM: number of children under 18 years of age in household
8. C1R: number of people in household
9. EDUC1: education level of mother with 4 categories
10. EDUC_TR: teen's current grade in school
11. INCQ298A: family income categories
12. CEN_REG: Census region based on true state of residence
13. LANGUAGE: language in which interview was conducted
14. NUM_PROVR: number of valid and unique providers indentified by respondent

For each macro, we present the model selection results from forward, backward and stepwise selection methods. We also compare the weighted and un-weighted model selection results from %StepSvylog vs. Proc Logistic, and %StepSvyreg vs. Proc Glmselect under three scenarios: forward, backward, stepwise. The results of the two examples are shown in Table 3 to Table 6 in below. In the examples, both entry model (&SLENTRY) and depart model (&SLSTAY) significant level are 0.05. The selection method is specified in the macro variable METHOD: METHOD=1 is forward selection; METHOD=2 is backward selection; METHOD=3 is stepwise selection.

Data Preparation for modeling

For household variables in the example dataset, the missing value codes are 77 for DON'T KNOW and 99 for REFUSED. To serve the modeling purposes, 77 and 99 of the 16 model variables (2 dependent variables and 14 independent variables) are coded as missing in the SAS data step below:

```
data teen;
set puf.nisteenpuf08;
keep rddwt estiapt08 race_k i_hisp_k sex age agegrp_m_i marital childnm c1r
educ1 educ_tr incq298a cen_reg language num_provr flu_any_rec ckup_age;
array vars {16} race_k i_hisp_k sex age agegrp_m_i marital childnm c1r educ1
educ_tr incq298a cen_reg language num_provr flu_any_rec ckup_age;
do i=1 to 16;
  if vars[i] in (77 99) then vars[i]=.;
end;
run;
```

Several macro variables: input dataset, covariates, strata and weight are generated to make the SAS code of using the two macros shorter and cleaner:

```
%let input=teen;
%let ind=race_k i_hisp_k sex age agegrp_m_i marital childnm c1r educ1 educ_tr
incq298a cen_reg language num_provr;
%let classv=race_k i_hisp_k sex agegrp_m_i marital educ1 educ_tr cen_reg
language;
%let strata=estiapt08;
%let weight=rddwt;
```

Example 1: %StepSvylog

Both weighted models and un-weighted models with or without survey design are compared.

- Weight variable is *RDDWT: final RDD phase weight.*
- Dependent variable is *FLU_ANY_REC: has teen received any influenza vaccinations in past 12 months?*
- Strata variable *is ESTIAP08: 2008 NIS-Teen estimation area*

1. Weighted logistic models:

The SAS code to fit Proc Logistic model with normalized RDD weight is as below:

```
%let dep=FLU_ANY_REC;

proc logistic DATA=&input descending;
class &classv;
model &dep = &ind /selection=forward slentry=0.05 slstay=0.05 rsquare
NODUMMYPRINT NODESIGNPRINT NODP;
weight &weight./normalize;
title "&dep";
run;
```

For backward or stepwise selection, specify selection=backward or selection=stepwise in the model option.

The SAS codes of using the macro %StepSvylog with origianl RDD weight are as below:

```
/*forward*/
%StepSvylog(FLU_ANY_REC,&ind.,&classv.,&input.,0.05,0.05,&weight.,&strata.,,1);

/*backward*/
%StepSvylog(FLU_ANY_REC,&ind.,&classv.,&input.,0.05,0.05,&weight.,&strata.,,2);

/*stepwise*/
%StepSvylog(FLU_ANY_REC,&ind.,&classv.,&input.,0.05,0.05,&weight.,&strata.,,3);
```

The only differences between forward, backward and stepwise codes are specifying Method=1, 2, or 3 in the last macro variable. The macro variable CLUSTER is missing since there is no clusters for NIS-Teen.

The selected covariates and their p-values for weighted models are displayed in Table 3.

Table 3 Compare weighted model selection results from %SteSvylog and Proc Logistic using FLU_ANY_REC as dependent variable and RDDWT as weight

Forward			Backward			Stepwise		
%StepSvylog								
	Selected Covariates	P-vlue		Selected Covariates	P-vlue		Selected Covariates	P-vlue
1	RACE_K	<.0001	1	RACE_K	<.0001	1	RACE_K	<.0001
2	CEN_REG	0.0014	2	CEN_REG	0.0011	2	CEN_REG	0.0011
3	EDUC1	0.0036	3	EDUC1	0.0032	3	EDUC1	0.0032
4	INCQ298A	0.017	4	INCQ298A	0.0201	4	INCQ298A	0.0201
5	AGE	0.0515	5	EDUC_TR	0.0321	5	EDUC_TR	0.0321
Proc Logistic								
	Selected Covariates	P-vlue		Selected Covariates	P-vlue		Selected Covariates	P-vlue
1	RACE_K	<.0001	1	RACE_K	<.0001	1	RACE_K	<.0001
2	EDUC1	<.0001	2	CEN_REG	<.0001	2	EDUC1	<.0001
3	CEN_REG	<.0001	3	EDUC1	<.0001	3	CEN_REG	<.0001
4	EDUC_TR	<.0001	4	EDUC_TR	<.0001	4	EDUC_TR	<.0001
5	INCQ298A	<.0001	5	INCQ298A	<.0001	5	INCQ298A	<.0001
6	NUM_PROVR	0.0165	6	CHILDNM	0.0017	6	NUM_PROVR	0.0165
7	SEX	0.0220	7	C1R	0.0061	7	SEX	0.0220
			8	NUM_PROVR	0.0145			
			9	SEX	0.0278			

*insignificant variable included according to 0.05 significant level

2. Un-weighted logistic Models:

The SAS code to fit un-weighted Proc Logistic model is as below:

```
%let dep=FLU_ANY_REC;

proc logistic DATA=&input descending;
class &classv;
model &dep = &ind /selection=forward slentry=0.05 slstay=0.05 rsquare
NODUMMYPRINT NODESIGNPRINT NODP;
title "&dep";
run;
```

Same as weighted models, for backward or stepwise selection, specify selection=backward or selection=stepwise in the model option.

The SAS codes of using the macro %StepSvylog with WEIGHT missing are as below:

```
/*forward*/
%StepSvylog(FLU_ANY_REC,&ind.,&classv.,&input.,0.05,0.05,,&strata.,,1);

/*backward*/
%StepSvylog(FLU_ANY_REC,&ind.,&classv.,&input.,0.05,0.05,,&strata.,,2);

/*stepwise*/
%StepSvylog(FLU_ANY_REC,&ind.,&classv.,&input.,0.05,0.05,,&strata.,,3);
```

The selected covariates and their p-values for weighted models are displayed in Table 4.

Table 4 Compare un-weighted model selection results from %SteSvylog and Proc Logistic using FLU_ANY_REC as dependent variable

Forward			Backward			Stepwise		
%StepSvylog								
	Selected Covariates	**P-vlue**		**Selected Covariates**	**P-vlue**		**Selected Covariates**	**P-vlue**
1	RACE_K	<.0001	**1**	RACE_K	<.0001	**1**	RACE_K	<.0001
2	AGE	<.0001	**2**	AGE	<.0001	**2**	AGE	<.0001
3	EDUC1	<.0001	**3**	EDUC1	<.0001	**3**	EDUC1	<.0001
4	SEX	0.0002	**4**	I_HISP_K	<.0001	**4**	SEX	0.0003
5	NUM_PROVR	0.0027	**5**	CHILDNM	0.0004	**5**	NUM_PROVR	0.0027
6	I_HISP_K	0.0027	**6**	SEX	0.0008	**6**	I_HISP_K	0.0027
7	AGEGRP_M_I	0.0127	**7**	NUM_PROVR	0.0012	**7**	AGEGRP_M_I	0.0127
8	INCQ298A	0.0273	**8**	C1R	0.0015	**8**	INCQ298A	0.0273
9	CEN_REG	0.0292	**9**	CEN_REG	0.0033	**9**	CEN_REG	0.0292
			10	AGEGRP_M_I	0.0124			
proc logistic								
	Selected Covariates	**P-vlue**		**Selected Covariates**	**P-vlue**		**Selected Covariates**	**P-vlue**
1	RACE_K	<.0001	**1**	RACE_K	<.0001	**1**	RACE_K	<.0001
2	AGE	<.0001	**2**	AGE	<.0001	**2**	AGE	<.0001
3	EDUC1	<.0001	**3**	SEX	0.0004	**3**	EDUC1	<.0001
4	SEX	0.0003	**4**	EDUC1	0.0004	**4**	SEX	0.0003
5	I_HISP_K	0.0027	**5**	LANGUAGE	0.0005	**5**	I_HISP_K	0.0027
6	NUM_PROVR	0.0027	**6**	CHILDNM	0.0013	**6**	NUM_PROVR	0.0027
7	AGEGRP_M_I	0.0119	**7**	C1R	0.0020	**7**	AGEGRP_M_I	0.0119
8	INCQ298A	0.0291	**8**	NUM_PROVR	0.0023	**8**	INCQ298A	0.0291
9	CEN_REG	0.0321	**9**	AGEGRP_M_I	0.0056	**9**	CEN_REG	0.0321
			10	CEN_REG	0.0250			

Summary of Example 1:

1. For weighted models, the selection results are very different. Proc Logistic selected more variables than %StepSvylog in forward, backward and stepwise.
2. For un-weighted models, Proc Logistic and %StepSvylog selected same variables in forward, backward and stepwise; but the p-values are different. And the orders of the variables sorted by p-values (numbers in the first column in each cell) from smallest to largest are different.
3. One insignificant variables AGE is kept in the forward %StepSvylog model, which indicates the advantage of using stepwise over forward.

Example 2 %StepSvyreg

For linear models, both weighted and un-weighted models with or without survey design are compared.

- Weight variable is *RDDWT: final RDD phase weight.*
- Dependent variable is *CKUP_AGE: age in years at last check-up*.
- Strata variable *is ESTIAP08: 2008 NIS-Teen estimation area*

3. Weighted linear models

The SAS code to fit PROC GLMSELECT model with RDD weight is as below:

```
%let dep=CKUP_AGE;

proc GLMSELECT DATA=&input;
class &classv;
model &dep = &ind /orderselect selection=forward(SELECT=SL STOP=SL SLE=0.05)
details=all;
weight &weight.;
title "&dep";
run;
```

The SAS codes of using the macro %StepSvyreg with RDD weight are as below:

```
/*forward*/
%StepSvyreg(CKUP_AGE,&ind.,&classv.,&input.,0.05,0.05,&weight.,&strata.,,1);

/*backward*/
%StepSvyreg(CKUP_AGE,&ind.,&classv.,&input.,0.05,0.05,&weight.,&strata.,,2);

/*stepwise*/
%StepSvyreg(CKUP_AGE,&ind.,&classv.,&input.,0.05,0.05,&weight.,&strata.,,3);
```

The selected covariates and their p-values are displayed in Table 5.

Table 5 Compare weighted model selection results from %SteSvyreg and Proc glmselect using CKUP_AGE as dependent variable, RDDWT as weight

Forward			Backward			Stepwise		
%StepSvyreg								
	Selected Covariates	P-vlue		Selected Covariates	P-vlue		Selected Covariates	P-vlue
1	AGE	<.0001	1	AGE	<.0001	1	AGE	<.0001
2	CEN_REG	<.0001	2	CEN_REG	<.0001	2	CEN_REG	<.0001
3	RACE_K	<.0001	3	RACE_K	<.0001	3	RACE_K	<.0001
4	EDUC_TR	<.0001	4	EDUC_TR	<.0001	4	EDUC_TR	<.0001
5	AGEGRP_M_I	0.0013	5	AGEGRP_M_I	0.0013	5	AGEGRP_M_I	0.0013
6	C1R	0.0022	6	C1R	0.0022	6	C1R	0.0022
7	INCQ298A	0.0032	7	INCQ298A	0.0032	7	INCQ298A	0.0032
8	EDUC1	0.0094	8	EDUC1	0.0094	8	EDUC1	0.0094
proc glmselect								
	Selected Covariates	P-vlue		Selected Covariates	P-vlue		Selected Covariates	P-vlue
1	AGE	<.0001	1	AGE	<.0001	1	AGE	<.0001
2	CEN_REG	<.0001	2	CEN_REG	<.0001	2	CEN_REG	<.0001
3	EDUC1	<.0001	3	EDUC1	<.0001	3	EDUC1	<.0001
4	EDUC_TR	<.0001	4	EDUC_TR	<.0001	4	EDUC_TR	<.0001
5	RACE_K	<.0001	5	RACE_K	<.0001	5	RACE_K	<.0001
6	C1R	<.0001	6	C1R	<.0001	6	C1R	<.0001
7	INCQ298A	<.0001	7	INCQ298A	<.0001	7	INCQ298A	<.0001
8	AGEGRP_M_I	<.0001	8	AGEGRP_M_I	<.0001	8	AGEGRP_M_I	<.0001
9	LANGUAGE	0.0002	9	LANGUAGE	0.0002	9	LANGUAGE	0.0002
10	I_HISP_K	0.0014	10	I_HISP_K	0.0014	10	I_HISP_K	0.0014
11	NUM_PROVR	0.0229	11	NUM_PROVR	0.0229	11	NUM_PROVR	0.0229

4. Un-weighted linear models

The SAS code of using PROC GLMSELECT without weight is as below:

```
%let dep=CKUP_AGE;

proc GLMSELECT DATA=&input;
class &classv;
model &dep = &ind /orderselect selection=forward(SELECT=SL STOP=SL SLE=0.05)
details=all;
title "&dep";
run;
```

The SAS codes of using the macro %StepSvyreg without weight are as below:

```
/*forward*/
%StepSvyreg(CKUP_AGE,&ind.,&classv.,&input.,0.05,0.05,,&strata.,,1);

/*backward*/
%StepSvyreg(CKUP_AGE,&ind.,&classv.,&input.,0.05,0.05,,&strata.,,2);

/*stepwise*/
%StepSvyreg(CKUP_AGE,&ind.,&classv.,&input.,0.05,0.05,,&strata.,,3);
```

The selected covariates and their p-values are displayed in Table 6

Table 6 Compare un-weighted model selection results from %SteSvyreg and Proc glmselect using CKUP_AGE as dependent variable

Forward			**Backward**			**Stepwise**		
%StepSvyreg								
Selected Covariates		**P-vlue**	**Selected Covariates**		**P-vlue**	**Selected Covariates**		**P-vlue**
1	AGE	<.0001	**1**	AGE	<.0001	**1**	AGE	<.0001
2	CEN_REG	<.0001	**2**	CEN_REG	<.0001	**2**	CEN_REG	<.0001
3	EDUC1	<.0001	**3**	EDUC1	<.0001	**3**	EDUC1	<.0001
4	RACE_K	<.0001	**4**	RACE_K	<.0001	**4**	RACE_K	<.0001
5	EDUC_TR	<.0001	**5**	EDUC_TR	<.0001	**5**	EDUC_TR	<.0001
6	I_HISP_K	<.0001	**6**	I_HISP_K	<.0001	**6**	I_HISP_K	<.0001
7	C1R	<.0001	**7**	C1R	<.0001	**7**	C1R	<.0001
8	AGEGRP_M_I	0.0001	**8**	AGEGRP_M_I	0.0001	**8**	AGEGRP_M_I	0.0001
9	LANGUAGE	0.0016	**9**	LANGUAGE	0.0016	**9**	LANGUAGE	0.0016
10	SEX	0.004	**10**	SEX	0.004	**10**	SEX	0.004
11	INCQ298A	0.034	**11**	INCQ298A	0.034	**11**	INCQ298A	0.034
proc glmselect								
Selected Covariates		**P-vlue**	**Selected Covariates**		**P-vlue**	**Selected Covariates**		**P-vlue**
1	AGE	<.0001	**1**	AGE	<.0001	**1**	AGE	<.0001
2	CEN_REG	<.0001	**2**	CEN_REG	<.0001	**2**	CEN_REG	<.0001
3	EDUC1	<.0001	**3**	EDUC1	<.0001	**3**	EDUC1	<.0001
4	RACE_K	<.0001	**4**	RACE_K	<.0001	**4**	RACE_K	<.0001
5	EDUC_TR	<.0001	**5**	EDUC_TR	<.0001	**5**	EDUC_TR	<.0001
6	AGEGRP_M_I	<.0001	**6**	AGEGRP_M_I	<.0001	**6**	AGEGRP_M_I	<.0001
7	C1R	0.0003	**7**	C1R	0.0003	**7**	C1R	0.0003
8	I_HISP_K	0.0004	**8**	I_HISP_K	0.0004	**8**	I_HISP_K	0.0004
9	SEX	0.0037	**9**	SEX	0.0037	**9**	SEX	0.0037
10	LANGUAGE	0.0038	**10**	LANGUAGE	0.0038	**10**	LANGUAGE	0.0038
11	INCQ298A	0.0275	**11**	INCQ298A	0.0275	**11**	INCQ298A	0.0275

Summary of Example 2:

1. For weighted models, the selection results are very different. Proc Glmselect selected more variables than %StepSvyreg in forward, backward and stepwise.
2. For un-weighted models, Proc Glmselect and %StepSvyreg selected same variables in forward, backward and stepwise; but the p-values are different. But the orders of the variables sorted by p-values (numbers in the first column in each cell) from smallest to largest are different.

Conclusion:

For logistic and linear models using complex survey data, incorporating survey design in the weighted models has more significant effects than in the un-weighted models. But even if un-weighted models selected same variables under the criterion of p-values, the orders of the variables sorted by p-values from smallest to largest are different.

REFERENCE

SAS OnlineDoc® 9.1.3, Copyright © 2002-2005, SAS Institute Inc., Cary, NC, USA; All rights reserved. Produced in the United States of America.

Zoran Bursac, C. Heath Gauss, D. Keith Williams, and David Hosmer, 2007. “A Purposeful Selection of Variables Macro for Logistic Regression”. *Proceedings of SAS Global Forum 2007, Paper 173-2007*.

5. ACKNOWLEDGMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

The authors would like to offer their sincere appreciation to Rebecca Wang, Ken Copeland, Nada Ganesh and Ben Duffey. Special thanks also go to for their statistical advice

6. CONTACT INFORMATION

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

Fang Wang
NORC at the University of Chicago
55 East Monroe Street, 20th Floor, Chicago IL 60603
Work Phone: (312) 325-2558
Email: wang-fang@norc.org

Hee-Choon Shin
NORC at the University of Chicago
1155 E. 60th Street, Chicago IL 60637
Work Phone: (773)256-6150
Email: shinh@uchicago.edu

APPENDIX: SAS CODE

```
/********************************************
* Programmer: Fang Wang wang-fang@norc.org  *
* Sep 2011                                  *
*********************************************/
%macro ScanVar (c,class);

%if &class = %then %do;

%put No class covariates given. ;

%end;

%global nclass;
%global classnames;

%global n;
%global covnames;

%let classcnt=1;
%let classnames=;

%do %while (%scan(&class,&classcnt,%STR( )) ne  );

        %let class&classcnt="%scan(&class,&classcnt,%STR( ))";
        %if &classcnt=1 %then %let classnames=%upcase(&&class&classcnt);
        %else %let classnames=&classnames %upcase(&&class&classcnt);
        %let classcnt=%eval(&classcnt+1);

%end;

    %GLOBAL nclass;
    %let nclass=%eval(&classcnt-1);
    %put Number of class covariates = &nclass ;


    data classv;
      length covname $40;
        keep covname;
        array word $40 w1-w&nclass (&classnames);
        do i=1 to &nclass ;
          covname = compress(word(i));
          output ;
        end;
        run;


%if &c = %then %do;

%put ERROR: No covariates given. Expecting at least one covariate;

%end;

%let covcnt=1;
%let covnames=;

%do %while (%scan(&c,&covcnt,%STR( )) ne  );

        %let c&covcnt="%scan(&c,&covcnt,%STR( ))";
        %if &covcnt=1 %then %let covnames=%upcase(&&c&covcnt);
        %else %let covnames=&covnames %upcase(&&c&covcnt);
        %let covcnt=%eval(&covcnt+1);

%end;


    %let n=%eval(&covcnt-1);


    %put Number of covariates = &n ;



    data indv;
            length covname $ 40;
        keep covname;
        array word $40  w1-w&n (&covnames);
        do i=1 to &n ;
          covname = compress(word(i));
          output ;
        end;
        run;

proc sql;
create table covariates as
select indv.covname as covname, classv.covname as classname
from indv left join classv on indv.covname=classv.covname;
quit;
```

```
%mend ScanVar;


/********* surveyreg *************/

%macro svyreg(dep,ind,class,input,out,weight,strata,cluster);

PROC surveyreg DATA=&input;
ods output Effects=&out(rename=EFFECT=parm);
STRATA &STRATA;
CLUSTER &CLUSTER;
WEIGHT &WEIGHT;
CLASS &CLASS;
MODEL &DEP = &IND;
RUN;

data &out;
set &out;
length parm $ 40;
if _N_ in (1 2) then delete;
keep parm probf fvalue;
run;

%mend svyreg;


%macro Forwardreg (SLENTRY);

%global new;
%global new_class;
%global old;
%global old_class;
%global c;
%global class;
%global nold;
%global ntotal;

data covariates ; set covariates nobs=numobs ;
    order = _N_;
    call symput ('startn', left(put(numobs,8.))) ;
run ;

%if &startn < &n %then %do;
    %put ERROR: NO VARIABLES ;
%end ;

%if &startn >= &n %then %do;

          data covariates ; set covariates ;
        temp = compress('c'||order) ;
            temp_class = compress('class'||order) ;
        call symput (temp , covname) ;
            call symput (temp_class , classname) ;
        drop temp temp_class;
        run ;
    %put MODEL VARIABLES: ;

  %do i = 1 %to &startn ;

   %put %upcase(&&c&i);

  %end;

    %do i=1 %to %eval(&&startn) ;
      title "Forward round=&round";
      %svyreg(dep=&DEP,ind=&old &&c&i,CLASS=&old_class
&&class&i,input=&regdata,out=uf&i,weight=&weight,strata=&strata,cluster=&clus
ter);
      data uf&i;
        set uf&i end=last;
        if last then output;
        run;
    %end;


%end ;

data candidates_temp;
length parm $40;
set uf1;
run;

%do i=2 %to  %eval(&&startn) ;
proc datasets nolist force;
append base=candidates_temp data=uf&i;
run;
%end;

%do i=1 %to %eval(&&startn);
proc datasets nolist;
delete uf&i;
run;
%end;

proc sql;
create table candidates as
select candidates_temp.*, classv.covname as class
from candidates_temp left join classv on
classv.covname=upcase(candidates_temp.parm) order by fvalue desc;
quit;

data candidates;
set candidates;
format probf 30.29;
tn=_n_;
run;

%let new=;
%let new_class=;
%let class=;
%let c=;
```

```
proc sql /*noprint*/;
select Probf into :pmin from candidates having tn=1;

select distinct parm into :new separated by ' '
from candidates having probf<=&SLENTRY and tn=1;

select distinct class into :new_class separated by ' '
from candidates having parm="&new.";

select distinct parm into :c separated by ' '
from candidates having parm^="&new.";

select distinct class into :class separated by ' '
from candidates having parm^="&new_class." and class^='';

quit;

%let old=&old &new;
%let old_class=&old_class &new_class;

title "Summary model forward round&round";
%svyreg(dep=&DEP,ind=&old,CLASS=&old_class,input=&regdata,out=step_&METHOD,we
ight=&weight,strata=&strata,cluster=&cluster);

proc sort data=step_&METHOD;
by probf;
run;

%mend Forwardreg;


%macro Backwardreg(SLSTAY);
%local i;

%let DEP=%upcase(&DEP);
%let old=%upcase(&old);
%let dataset=%upcase(&dataset);

data step_&METHOD;
length parm $ 40;
if parm='' then delete;
run;

title "Backward";
%let i=1;
%do %until (&pmax<=&slstay or &varlist='');
%if &i = 1 %then
%svyreg(DEP=&DEP ,IND=&old, CLASS=&old_class,
input=&regdata,out=pval,weight=&weight,strata=&strata,cluster=&cluster); %*in
itial model;
%else %do;
%svyreg(DEP=&DEP ,IND=&varlist,CLASS=&old_class,input=&regdata,out=pval,weigh
t=&weight,strata=&strata,cluster=&cluster); %*reduced model;
%end;

proc sort data=step_&METHOD; by parm;
proc sort data=pval; by parm;
data step_&METHOD;
format p&i 30.29;
merge step_&METHOD pval;
by parm;
p&i=probf;
drop probf;
run;

%let varlist='';
proc sql noprint;
select round(max(probf),0.001) into :pmax from pval;

select distinct parm into :varlist separated by ' '
from pval
having probf^=max(probf);
quit;
%let j=%eval(&i);
%let i=%eval(&i+1);

%end;

proc sort data=step_&METHOD;
by p&j;
run;

%let removed=;

proc sql /*noprint*/;

select distinct parm into :removed separated by ' ' from step_&METHOD having
p&j=. ;

select distinct parm into :old separated by ' ' from step_&METHOD having
p&j^=.;

quit;

data step_&METHOD;
retain parm p&j;
set step_&METHOD;
rename p&j=ProbF;
keep parm p&j;
if p&j^=.;
run;

%let c=&c &removed;

%mend Backwardreg;

%macro StepSvyreg(
/********************************************
* Programmer: Fang Wang wang-fang@norc.org  *
* Sep 2011                                  *
*********************************************/
DEP /*dependent variable*/
,INTVAR /*all the candidate independent variavles */
,INTCLASS /*candidate categorical independent varaibels*/
,DATASET/*input dataset*/
,SLENTRY/*entry p value*/
,SLSTAY/*stay p value*/
,WEIGHT/*weight for wegihted regression models*/
,STRATA/*complex survye strata variable*/
,CLUSTER/*complex survey cluster variable*/
,METHOD/*selection method: forward, backward or stepwise*/
);

%let round=1;
%let old=;
%let old_class=;
%let pmin=0;


%let method=%upcase(&method);

%if &method=1 %then %do;
 %do %until (&new= or &pmin>&SLENTRY or (&n=1));
  %if &round = 1 %then %do;
   %ScanVar (&intvar,&intclass);
   %let regdata=&dataset;
   %Forwardreg(&SLENTRY);
   %let round=%eval(&round+1);
  %end;
  %else %do;
   %ScanVar (&c,&class);
   %Forwardreg(&SLENTRY);
   %let round=%eval(&round+1);
  %end;
 %end;
%end;

%if &method=2 %then %do;
 %ScanVar (&intvar,&intclass);
  %let old=&intvar;
  %let old_class=&intclass;
  %let regdata=&dataset;
 %Backwardreg(&SLSTAY);
%end;

%if &method=3 %then %do;
 %do %until (&new= or &pmin>&SLENTRY or (&n=1));
  %if &round = 1 %then %do;
   %ScanVar (&intvar,&intclass);
   %let regdata=&dataset;
   %Forwardreg(&SLENTRY);
   %Backwardreg(&SLSTAY);
   %let round=%eval(&round+1);
  %end;
  %else %do;
   %ScanVar (&c,&class);
   %Forwardreg(&SLENTRY);
   %Backwardreg(&SLSTAY);
   %let round=%eval(&round+1);
  %end;
 %end;
%end;

proc print data=step_&METHOD;
format ProbF 7.6;
title "&DEP";
run;

%mend StepSvyreg;

/************** surveylogistic *****************/

%macro Svylog(dep,ind,class,input,out,weight,strata,cluster);

%if &weight^= %then %do;
PROC surveylogistic DATA=&input;
ods output Type3=type3(rename=EFFECT=parm)
ParameterEstimates=parameter(rename=Variable=parm);
STRATA &STRATA;
CLUSTER &CLUSTER;
WEIGHT &WEIGHT;
CLASS &CLASS;
MODEL &DEP(EVENT='1')= &IND;
RUN;
%end;

%else %do;
PROC surveylogistic DATA=&input;
ods output Type3=type3(rename=EFFECT=parm)
ParameterEstimates=parameter(rename=Variable=parm);
STRATA &STRATA;
CLUSTER &CLUSTER;
CLASS &CLASS;
MODEL &DEP(EVENT='1')= &IND;
RUN;
%end;

%if &class= %then %do;
data &out;
length parm $20.;
set parameter;
if parm='' or parm='Intercept' then delete;
keep parm waldchisq probchisq;
run;
%end;

%else %do;
data &out;
length parm $20.;
set type3;
if parm='' or parm='Intercept' then delete;
keep parm waldchisq probchisq;
run;
%end;

%mend Svylog;
```

```
%macro Forwardlog (SLENTRY);

%global new;
%global new_class;
%global c;
%global class;
%global nold;
%global ntotal;

data covariates ; set covariates nobs=numobs ;
    order = _N_;
    call symput ('startn', left(put(numobs,8.))) ;
run ;

%if &startn < &n %then %do;
    %put ERROR: NO VARIABLES ;
%end ;

%if &startn >= &n %then %do;

          data covariates ; set covariates ;
        temp = compress('c'||order) ;
            temp_class = compress('class'||order) ;
        call symput (temp , covname) ;
            call symput (temp_class , classname) ;
        drop temp temp_class;
        run ;

    %put MODEL VARIABLES: ;

  %do i = 1 %to &startn ;

   %put %upcase(&&c&i);
   %put %upcase(&&class&i);

  %end;

    %do i=1 %to %eval(&&startn) ;
      title "Forward round=&round";
      %Svylog(dep=&DEP,ind=&old &&c&i,CLASS=&old_class
&&class&i,input=&regdata,out=uf&i,weight=&weight,strata=&strata,cluster=&clus
ter);
      data uf&i;
        set uf&i end=last;
        if last then output;
        run;
    %end;


%end ; /* END LOOP */

data candidates_temp;set uf1;
run;

%do i=2 %to  %eval(&&startn) ;
proc datasets nolist force;
```

```
append base=candidates_temp data=uf&i;
run;
%end;

%do i=1 %to %eval(&&startn);
proc datasets nolist;
delete uf&i;
run;
%end;

proc sql;
create table candidates as
select candidates_temp.*, classv.covname as class
from candidates_temp left join classv on
classv.covname=upcase(candidates_temp.parm) order by ProbChiSq;
quit;

data candidates;
set candidates;
format ProbChiSq 30.29;
tn=_n_;
run;

%let new=;
%let new_class=;
%let class=;
%let c=;


proc sql /*noprint*/;
select ProbChiSq into :pmin from candidates having tn=1;

select distinct parm into :new
from candidates having tn=1 and ProbChiSq<=&SLENTRY;

select distinct class into :new_class
from candidates having parm="&new.";

select distinct parm into :c separated by ' '
from candidates having parm^="&new.";

select distinct class into :class separated by ' '
from candidates having parm^="&new_class." and class^='';

quit;

%let old=&old &new;
%let old_class=&old_class &new_class;

title "Summary model forward round&round";
%Svylog(dep=&DEP,ind=&old,CLASS=&old_class,input=&regdata,out=STEP_&METHOD,we
ight=&weight,strata=&strata,cluster=&cluster);

proc sort data=STEP_&METHOD;
by ProbChiSq;
run;
```

```
%mend Forwardlog;


%macro Backwardlog(SLSTAY);
%local i;

%let DEP=%upcase(&DEP);
%let old=%upcase(&old);
%let dataset=%upcase(&dataset);

data STEP_&METHOD;
length parm $ 20;
if parm='' then delete;
run;

title "Backward";
%let i=1;
%do %until (&pmax<=&slstay or &varlist='');
%if &i = 1 %then
%Svylog(DEP=&DEP ,IND=&old, CLASS=&old_class,
input=&regdata,out=pval,weight=&weight,strata=&strata,cluster=&cluster); %*in
itial model;
%else %do;
%Svylog(DEP=&DEP ,IND=&varlist,CLASS=&old_class,input=&regdata,out=pval,weigh
t=&weight,strata=&strata,cluster=&cluster); %*reduced model;
%end;
proc sort data=STEP_&METHOD; by parm;
proc sort data=pval; by parm;
data STEP_&METHOD;
merge STEP_&METHOD pval;
by parm;
p&i=ProbChiSq;
format p&i 6.5;
drop ProbChiSq;
run;
%let varlist='';
proc sql noprint;
select round(max(ProbChiSq),0.001) into :pmax
from pval;
select distinct parm into :varlist separated by ' '
from pval
having ProbChiSq^=max(ProbChiSq);
quit;
%let j=%eval(&i);
%let i=%eval(&i+1);
%end;

proc sort data=STEP_&METHOD;
by p&j;
run;

%let removed=;

proc sql noprint;

select distinct parm into :removed separated by ' ' from STEP_&METHOD having
p&j=. ;
```

```
select distinct parm into :old separated by ' ' from STEP_&METHOD having
p&j^=. order by p&j;

quit;

data step_&METHOD;
set step_&METHOD;
rename p&j=ProbChiSq;
keep parm p&j;
if p&j^=.;
run;

%let c=&c &removed;

%mend Backwardlog;

%macro StepSvylog(
/********************************************
* Programmer: Fang Wang wang-fang@norc.org  *
* Sep 2011                                  *
*********************************************/
DEP /*dependent variable*/
,INTVAR /*all the candidate independent variavles */
,INTCLASS /*candidate categorical independent varaibels*/
,DATASET/*input dataset*/
,SLENTRY/*entry p value*/
,SLSTAY/*stay p value*/
,WEIGHT/*weight for wegihted regression models*/
,STRATA/*complex survye strata variable*/
,CLUSTER/*complex survey cluster variable*/
,METHOD/*selection method: forward, backward or stepwise*/
);
%let round=1;
%global old;
%global old_class;
%let old=;
%let old_class=;
%let pmin=0;

%let method=%upcase(&method);

%if &method=1 %then %do;
 %do %until (&new= or &pmin>&SLENTRY or (&n=1));
  %if &round = 1 %then %do;
   %ScanVar (&intvar,&intclass);
   %let regdata=&dataset;
   %Forwardlog(&SLENTRY);
   %let round=%eval(&round+1);
  %end;
  %else %do;
   %ScanVar (&c,&class);
   %Forwardlog(&SLENTRY);
   %let round=%eval(&round+1);
  %end;
 %end;
%end;
```

```
%if &method=2 %then %do;
   %let old=&intvar;
   %let old_class=&intclass;
   %let regdata=&dataset;
 %Backwardlog(&SLSTAY);
%end;

%if &method=3 %then %do;
 %do %until (&new= or &pmin>&SLENTRY or (&n=1));
  %if &round = 1 %then %do;
   %ScanVar (&intvar,&intclass);
   %let regdata=&dataset;
   %Forwardlog(&SLENTRY);
   %Backwardlog(&SLSTAY);
   %let round=%eval(&round+1);
  %end;
   %else %do;
   %ScanVar (&c,&class);
   %Forwardlog(&SLENTRY);
   %Backwardlog(&SLSTAY);
   %let round=%eval(&round+1);
  %end;
 %end;
%end;

proc print data=STEP_&METHOD;
title "&DEP: backward deletion";
run;

%mend StepSvylog;
```