Paper 1657-2014

A SAS® macro for complex sample data analysis using generalized linear Models

Paulo Henrique Dourado da Silva, Universidade de Brasília, Dep. de Estatística, Brazil
Alan Ricardo da Silva, Universidade de Brasília, Dep. de Estatística, Brazil

ABSTRACT

The purpose of this paper is showing a SAS® macro named ***%surveyglm*** developed in IML Procedure in order that users can incorporate informations about survey design to the Generalized Linear Models (GLM). The R function *svyglm* (Lumley, 2004) was used as background to the ***%surveyglm*** macro estimates. The results showed that estimates are close to the R function and that new distributions can be easily added to the algorithm.

1. INTRODUCTION

Many analysts in Brazil analyze data obtained from complex sample surveys produced by Brazilian Institute of Geography and Statistics (IBGE). Complex sample data may have at least one of the following characteristics: (i) stratification, (ii) clustering of the primary units (iii) and/or unequal probability of selection. If these features are not included in the analysis, point estimates and standard deviations are incorrect (Lohr 2009, Chambers and Skinner 2003).

Other use of this data is to build regression models for secondary analysis. Many analysts often analyze these data using statistical packages that are based on assumptions restricted to simple random sampling with replacement, which makes incorrect inferences. There are some packages implemented in R (*svyglm*), GLIM (*sglim*) and SAS® (SURVEYREG and SURVEYLOGISTIC) which incorporate information about the survey design in the models. However, in SAS® software there are procedures that incorporate such information to the other models of the generalized linear models of Nelder and Wedderburn (1972), but it is not possible to incorporate the survey design.

This paper will describe a SAS® macro named ***%surveyglm*** that allows incorporate in generalized linear models (GLM) aspects related to the survey design. The R function *svyglm* (Lumley, 2004) was used as background to the estimates generated by ***%surveyglm*** macro.

The paper is organized as follows. In Section 2 the theory about the generalized linear models and complex sampling are given. In Section 3, a SAS® macro is presented and in Section 4 we introduce an illustration. The conclusions are in Section 5.

2 GENERALIZED LINEAR MODELS AND COMPLEX SAMPLING

The generalized linear models (GLM) were introduced by Nelder and Wedderburn (1972). They unified many existing methodologies for data analysis in a single regression approach. The linear model was extended in two ways: (i) the assumption of normality for the random error of the model was extended to the class of uniparametric exponential family, and (ii) additivity of the effects of explanatory variables is carried out on a scale transformation function defined by a monotonous function called link function.

The GLM is defined by three components:

- A random component, which is represented by the family distribution of the response variable Y, and that belongs to the exponential family distribution;
- A systematic component represented by the linear predictor $\boldsymbol{X'\beta}$;
- And a link function, monotone and differentiable, $\boldsymbol{\eta = g(\mu)}$ linking the random component to the systematic component in the model.

2.1 ESTIMATION PROCEDURE

The primary method for the parameter estimation in generalized linear model is the maximum likelihood. To use this method we need to maximize the log-likelihood function associated with the distribution of the response variable:

$$L(y, \mu, \phi) = \sum_i log(f(y_i, \mu_i, \phi))$$

In complex sampling is used the pseudo-maximum likelihood method, which incorporated the sample weight to the log-likelihood function. The importance of incorporating the sample weight in the process of point estimation is because the point estimates are unbiased and it allows the correct estimation of the variance of the estimators. For the parameters estimation is commonly used ridge-stabilized Newton-Raphson algorithm, which is implemented in the SAS GENMOD procedure (SAS, 2011).

In the k-th iteration, the algorithm updates the parameter vector $\boldsymbol{\beta}_k$ as following:

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - \boldsymbol{H}^{-1}\boldsymbol{s}$$

where $\boldsymbol{H}$ is Hessian matrix, and $\boldsymbol{s}$ is the gradient vector of the function of pseudo-log-likelihood, both evaluated on each iteration,

$$\boldsymbol{s} = [s_j] = \left[\frac{\partial L}{\partial \beta_j}\right]$$

and

$$\boldsymbol{H} = [h_{ij}] = \left[\frac{\partial^2 L}{\partial \beta_i \partial \beta_j}\right]$$

For models that have some scale parameter (and/or dispersion)[1], this parameter is assumed known and is estimated by maximum likelihood or moments method[2]. To estimate the vector $\boldsymbol{s}$ and the matrix $\boldsymbol{H}$, we can use the chain rule since $\mu_i = g^{-1}(\boldsymbol{x}_i'\boldsymbol{\beta})$, thus the vector $\boldsymbol{s}$ and the matrix $\boldsymbol{H}$ are given by

$$\boldsymbol{s} = \sum_{\boldsymbol{i}} \frac{w_i(y_i - \mu_i)}{V(\mu_i)g'(\mu_i)\phi}$$

and

$$\boldsymbol{H} = -\boldsymbol{X}'\boldsymbol{W}_{\boldsymbol{o}}\boldsymbol{X}$$

respectively. Here $\boldsymbol{X}$ is the matrix containing the information of the covariates for each individual, $\boldsymbol{x}_{\boldsymbol{i}}$ is the transpose of *i*-th line of $\boldsymbol{X}$, V is the variance function and $\boldsymbol{W}_o$ is the diagonal matrix with typical element defined by

$$w_{oi} = w_{ei} + w_i(y_i - \mu_i)\,\frac{V(\mu_i)g''(\mu_i) + V'(\mu_i)g'(\mu_i)}{\big(V(\mu_i)\big)^2\big(g'(\mu_i)\big)^3\phi}$$

where

$$w_{ei} = \frac{w_i}{\phi V(\mu_i)\big(g'(\mu_i)\big)^2}$$

The information matrix is given by the observed negative value of the matrix $\boldsymbol{H}$[3]. Note that the information of the sampling weight is incorporated into the parameter estimation process through w_i which is denominated as prior weight.

2.2 VARIANCE ESTIMATION

For the variance estimation of the regression parameters is used Taylor linearization method, because it is widely used in practice. Therefore, the estimated covariance matrix of $\boldsymbol{\beta}$ using is given by

$$\widehat{\boldsymbol{V}}\big(\widehat{\boldsymbol{\beta}}\big) = \widehat{\boldsymbol{Q}}^{-1}\boldsymbol{G}\widehat{\boldsymbol{Q}}^{-1}$$

where

[1]Normal, gamma, negative binomial, inverse normal models.
[2]More information about the process of parameter estimation in models with scale parameter and/or dispersion can be found in McCullagh andNelder (1989).

$$\hat{Q} = \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\,\widehat{\boldsymbol{D}}_{hij}\left(V\left(\mu_{hij}\right)\right)^{-1}\widehat{\boldsymbol{D}}'_{hij} = \phi \boldsymbol{X}'\boldsymbol{W}_{\boldsymbol{e}}\boldsymbol{X}$$

$$\hat{G} = \frac{n-1}{n-p}\sum_{h=1}^{H}\frac{n_h(1-f_h)}{n_h-1}\sum_{i=1}^{n_h}(e_{hi.} - \bar{e}_{h..})(e_{hi.} - \bar{e}_{h..})'$$

$$e_{hi.} = \sum_{j=1}^{m_{hi}} w_{hij}\,\widehat{\boldsymbol{D}}_{hij}\left(V\left(\mu_{hij}\right)\right)^{-1}\left(y_{hij} - \hat{\mu}_{hij}\right) = \phi \boldsymbol{W}_e^*(\boldsymbol{y}-\boldsymbol{\mu})$$

$$\bar{e}_{hi.} = \frac{1}{n_h}\sum_{i=1}^{n_h} e_{hi.}$$

since matrix $\widehat{\boldsymbol{D}}'_{hij}\left[\frac{1}{g'(\mu_{hij})}\right]'$ and $\boldsymbol{W}_e^* = diag\left(\frac{w_i}{\phi V(\mu_i) g'(\mu_i)}\right)$.

2.3 VARIANCE CORRECTION

Morel (1989) introduced a method for correction of the covariance matrix in small samples under complex sampling design, by means of the matrix **Q**, which incorporates information about the distribution of the response variable. Thus we naturally extend this correction to the other exponential family distributions. In the ***%surveyglm*** macro, this method was implemented to correct bias of the estimator of the variance parameter in small samples. This correction is given by the following equation:

$$\widehat{\boldsymbol{V}}\left(\widehat{\boldsymbol{\beta}}\right) = \widehat{\boldsymbol{Q}}^{-1}\boldsymbol{G}\widehat{\boldsymbol{Q}}^{-1} + k\lambda\widehat{\boldsymbol{Q}}^{-1}$$

where

$$k = \max\left(\delta, p^{-1}tr\left(\widehat{\boldsymbol{Q}}^{-1}\boldsymbol{G}\right)\right) \quad \text{and} \quad \lambda = \min\left(\psi, \frac{p}{\tilde{n}-p}\right).$$

Where $\tilde{n}$ is the total number of cluster in the entire sample. The adjustment $k\lambda\widehat{\boldsymbol{Q}}^{-1}$ can be interpreted in the following ways (Morel,1989) and (SAS,2011):

- Reduce the small sample bias reflected in the inflation of the type I error in the hypotheses of the parameters;
- Guarantee the inverse matrix estimated $\widehat{\boldsymbol{Q}}^{-1}$, when exist, is positive-definite;
- Is near zero when the sample size becomes large.

The k is viewed as an estimator for the design effect, which is bounded below by δ. Following the orientation of (SAS, 2011) we choice the default value $\delta = 1$. The factor λ converges to zero when the sample size increases, and λ is bounded above by ψ. The default value for ψ is equal to 0.5.

3 SAS® MACRO

The SAS® macro ***%surveyglm*** has the following general call:

```
%surveyglm(base=,y=, x=, weight=, strata=, cluster=, fpc=,
dist=gamma,link=inv, intercept=y, scale=deviance, vadjust=n,
alpha=0.05, delta=1, fi=0.5, maxiter=100, eps=0.000001);
```

[3]The expected value of $\boldsymbol{W}_e$, is the diagonal matrix $\boldsymbol{W}_e$, If we replace $\boldsymbol{W}_0$ by $\boldsymbol{W}_e$ then the negative value of $\boldsymbol{H}$ is called the expected information matrix, thus $\boldsymbol{W}_e$ is the matrix of weights for the Fisher score method, for the parameters estimation. Thus, any of the weight matrices can be used to update the parameters in the iterative process.

The variable $base$ gets the information from the database that is being analyzed, y receives the information of the response variable and x receives the covariates. The information about the distribution of the response variable and the link function are incorporated in variables $dist$ and link, respectively. In variable $scale$ we must inform the estimation method of the dispersion parameter (Deviance or Pearson). Using the variables $weight, strata, cluster$ and fpc we can incorporate the information about the sampling design, such as weight, stratum, cluster and population correction factor, respectively.

The Boolean variables $intercept$ and $vadjus$t is about the presence of the intercept in the model, and if it necessary to use the correction introduced by Morel (1989) in the variance estimation. If such correction is desired we must inform the values for $delta$ and fi. The variable $alpha$ is the significance level for the confidence interval for the estimated odds ratio, when binomial model is considered.

Finally we have variables that are assigned values related to the convergence of the algorithm, $maxiter$ and eps, that say the maximum number of iterations allowed for the algorithm and the convergence criterion, respectively.

Some important observations (and in some cases limitations) of the macro are:

1) If the user does not provide information about the sample design, i.e. strata and cluster, the ***%surveyglm*** macro reduces to the GENMOD procedure;
2) The implemented models are Gamma, Normal, Inverse gaussian, Poisson, Binomial and Multinomial;
3) The link functions implemented are inverse, inverse squared, identity, logarithmic, Generalized logit;
4) For the estimation of scale parameter, we implement the Pearson and deviance methods. If the user does not to specify the $scale$ variable, then scale parameter is fixed by 1 for Binomial, Multinomial and Poisson models. For other models, we use as default the deviance method for scale estimation ;
5) If the user is interested in adjust a multinomial or binomial model, just specify binomial in the variable *dist*. The implemented link function is the glogit (Generalized logit function), for the multinomial distribution. For binary response, the glogit function reduces to logit;
6) All variables specified in the macro must be numeric, otherwise an error occurs;
7) The user must create dummies to represent the categories of some qualitative explanatory variable before put into the macro. The macro only recognizes the categories in the response variable;
8) Missing values should be removed before using the macro;
9) In the case where the response variable is categorical, the adjusted probability refers to the lowest category, for example, if the response is 0 or 1, 0 is the base category.

3.1 HOW TO IMPLEMENT A MODEL IN MACRO *%SURVEYGLM*

In this section we will show how the user can implement other models in the macro. As an example, consider the Negative Binomial distribution. The steps to implement this model are:

1. Implement the inverse link function and the link function derivatives of first and second order:
For the negative binomial model the canonical link function is the logarithmic function, so the inverse link function is the exponential function. So just add the following commands:

```
mu=exp(eta);
/*First derivative - link function*/
gderiv=1/mu;
/*Second derivative - link function*/
gderiv2=-1/(mu##2);
```

2. Implement the variance function and its derivatives of first order:
For the negative binomial model, the variance function is given by

$$V(\mu) = \mu + k\mu^2$$

where *k* is the dispersion parameter. Its derivative is given by

$$V'(\mu) = 1 + 2k\mu$$

For implement these functions in the macro, add the following commands:

```
/*Variance function*/
```

```
vmu=mu+k#(mu##2);
/*First derivative - variance function*/
vmuderiv=1+2#k#mu;
```

3. Estimation of the dispersion parameter k:

To estimate the dispersion parameter, first initialize its value equal to 1, and after that you can estimate $\boldsymbol{\beta}$. With the estimated $\widehat{\boldsymbol{\beta}}$, compute the log-likelihood function. Use an optimization function to find the value of k that maximizes the log-likelihood function. Alternate these steps until the algorithm reaches convergence.

So, just follow the other lines of the macro commands, in appendix, to compute other interest values (AIC, log-likelihood, scale parameter for the quasi negative binomial model, among others). Note that the user must have an intermediate/advanced knowledge to implement the above command lines in the macro, always keeping the same structure.

4 ILLUSTRATION

To illustrate, we estimate a model which explains the household income by the years of study of the family head in 2007 in Brazil. To show the versatility of the macro ***%surveyglm*** we first adjust a gamma model with link function log, and after that we adjust a normal model with identity link function. Finally we consider an application for the binomial and multinomial models. The macro calls are:

```
%surveyglm(base=pnad, y=remuneracao, x=anos_estudo, dist=gamma,
link=log, weight=v4729, strata=v4602, cluster=UPA, fpc=v4605,
scale=deviance, intercept=s, vadjust=n);

%surveyglm(base=pnad, y=remuneracao, x=anos_estudo, dist=normal,
link=identity, weight=v4729, strata=v4602, cluster=UPA,
fpc=v4605, scale=deviance, intercept=s, vadjust=n);
```

The results from GENMOD and SURVEYREG procedures, R function svyglm and **%surveyglm** macro, are in Tables 1 and 2.

Table 1: Parameters estimated by gamma model

PROCEDURE	COEFFICIENT	POINT ESTIMATE	STANDARD ERROR
PROC GENMOD (without weights)	INTERCEPT	5.6372	0.0085
	YEARS OF STUDY	0.1141	0.0011
PROC GENMOD (with weights)	INTERCEPT	5.7044	0.0084
	YEARS OF STUDY	0.1087	0.0010
SVYGLM (Stratum and cluster)	INTERCEPT	5.704350	0.008663
	YEARS OF STUDY	0.108739	0.001032
%SURVEYGLM (Stratum and cluster)	INTERCEPT	5.7043609	0.0087038
	YEARS OF STUDY	0.1087377	0.0010348

According to Table 1 we can see the influence that the sampling design and the weights have on the estimates. For the first case, which was used in the classical gamma regression, the point estimates are different from those obtained by other procedures as well as the standard errors. For all other cases, the point estimates seem very close to the each coefficient. Looking the standard errors, we can see that the results of the function *svyglm* and ***%surveyglm*** macro are very close and that their values differ from other procedures for incorporating the effect of sampling design.

Table 2: Parameters estimated by normal model

PROCEDURE	COEFFICIENT	POINT ESTIMATE	STANDARD ERROR
PROC GENMOD (without weights)	INTERCEPT	6.2230	10.5068
	YEARS OF STUDY	101.7026	1.3195
PROC GENMOD (with weights)	INTERCEPT	52.5637	9.6447
	YEARS OF STUDY	96.9075	1.2230
PROC SURVEYREG (Stratum and cluster)	INTERCEPT	52.5637149	5.32327415
	YEARS OF STUDY	96.9074824	1.06817261
SVYGLM (Stratum and cluster)	INTERCEPT	52.564	5.292
	YEARS OF STUDY	96.907	1.066
%SURVEYGLM (Stratum and cluster)	INTERCEPT	52.5637149	5.3232741
	YEARS OF STUDY	96.9074824	1.0681726

According to Table 2 we can see again the influence that the sampling design and the weights have on the estimates. For the first case, which was used in the classical regression point estimates are quite different from those obtained by other procedures as well as the standard errors. For all other cases, the point estimates seem very close to the each coefficient. Looking the estimates of standard errors, we can see that the results of the function *svyglm* and ***%surveyglm*** macro are close, and that the results generated by the ***%surveyglm*** macro and PROC SURVEYREG are identical. Here again their values differ from other procedures for incorporating the effect of sampling design.

If the user specifies the binomial distribution, the macro automatically identifies how many categories there are in the response variable. So, when *dist=binomial*, the response can be binary or multinomial. To illustrate the use of binomial and multinomial distributions, we considered two databases. The first refers to the study on cancer remission (Lee, 1974), where the data consist of the patient characteristics and whether or not cancer remission occurred. This dataset is given in example 53.1 of GENMOD Procedure (SAS 2011), where the variable *remiss* is the cancer remission indicator variable with a value of 1 for remission and a value of 0 for nonremission. The other six variables are the risk factors thought to be related to the cancer remission. The other database is given by Hosmer and Lemeshow (2000) about the factors associated with women's knowledge, attitude, and behavior toward mammography. The dependent variable is *me* (mammography experience) coded as 0 = Never, 1 = Within One Year and 2 = Over One Year Ago.

To adjust binomial model were used the ***%surveyglm*** macro and the GENMOD Procedure, and the parameters estimated are shown in Table 3. The estimated odds ratios and their respective confidence intervals are shown in Table 4, and we use LOGISTIC Procedure for that. The macro call is:

```
%surveyglm(base=Remission, y=remiss,x=cell smear infil li blast
temp, dist=binomial, link=, weight=, strata=, cluster=, fpc=,
scale=, intercept=y, vadjust=n);
```

Table 3: Parameters estimated by binomial model

PROCEDURE	COEFFICIENT	POINT ESTIMATE	STANDARD ERROR
PROC GENMOD	INTERCEPT	58.0385	71.2364
	CELL	24.6615	47.8377
	SMEAR	19.2936	57.9500
	INFIL	-19.6013	61.6815
	LI	3.8960	2.3371
	BLAST	0.1511	2.2786
	TEMP	-87.4339	67.5736
%SURVEYGLM	INTERCEPT	58.0385	71.2364
	CELL	24.6615	47.8377
	SMEAR	19.2936	57.9500
	INFIL	-19.6013	61.6815
	LI	3.8960	2.3371
	BLAST	0.1511	2.2786
	TEMP	-87.4339	67.5736

We can see in Table 3 that the parameters estimated by ***%surveyglm*** macro are exactly the same of the GENMOD Procedure. As noted in the previous section, when it is not specified information about the sampling design, the

%surveyglm macro reduces to the GENMOD Procedure. Odds ratios estimated by the macro are exactly same of the LOGISTIC procedure, as shown in table 4.

Table 4: Odds Ratio Estimates

PROCEDURE	COEFFICIENT	POINT ESTIMATE	95% WALD CONFIDENCE LIMITS	
PROC GENMOD	CELL	> 999.999	< 0.001	> 999.999
	SMEAR	> 999.999	< 0.001	> 999.999
	INFIL	< 0.001	< 0.001	> 999.999
	LI	49.203	0.504	> 999.999
	BLAST	1.163	0.013	101.191
	TEMP	< 0.001	< 0.001	> 999.999
%SURVEYLM	CELL	> 999.999	< 0.001	> 999.999
	SMEAR	> 999.999	< 0.001	> 999.999
	INFIL	< 0.001	< 0.001	> 999.999
	LI	49.203	0.504	> 999.999
	BLAST	1.163	0.013	101.191
	TEMP	< 0.001	< 0.001	> 999.999

To adjust the multinomial model were used the ***%surveyglm*** macro and the LOGISTIC Procedure, and the parameters estimated are shown in Table 5. The estimated odds ratios and their respective confidence intervals are shown in Table 6. The macro call is:

```
%surveyglm(base=meexp, y=me,x=symptd pb hist bse, dist=binomial,
link=, weight=, strata=, cluster=, fpc=, scale=, intercept=y,
vadjust=n);
```

Table 5: Parameters estimated by multinomial model

PROCEDURE	COEFFICIENT	POINT ESTIMATE	STANDARD ERROR
PROC LOGISTIC	INTERCEPT	-1.7885	0.8470
	INTERCEPT	-1.7421	0.8087
	SYMPTD	2.2302	0.4519
	SYMPTD	1.1531	0.3514
	PB	-0.2825	0.0713
	PB	-0.1578	0.0712
	HIST	1.2966	0.4293
	HIST	1.0613	0.4527
	BSE	1.2209	0.5210
	BSE	0.9604	0.5072
%SURVEYGLM	INTERCEPT	-1.7888	0.8253
	INTERCEPT	-1.7421	0.7846
	SYMPTD	2.2304	0.4456
	SYMPTD	1.1531	0.3440
	PB	-0.2825	0.0679
	PB	-0.1578	0.0676
	HIST	1.2966	0.3569
	HIST	1.0613	0.3768
	BSE	1.2210	0.5115
	BSE	0.9604	0.4980

Table 6: Odds Ratio Estimates

PROCEDURE	COEFFICIENT	POINT ESTIMATE	95% WALD CONFIDENCE LIMITS	
PROC LOGISTIC	SYMPTD	9.302	3.836	22.555
	SYMPTD	3.168	1.591	6.308
	PB	0.754	0.655	0.867
	PB	0.854	0.743	0.982
	HIST	3.657	1.577	8.483
	HIST	2.890	1.190	7.019
	BSE	3.390	1.221	9.413
	BSE	2.613	0.967	7.060
%SURVEYLM	SYMPTD	9.304	3.885	-22.283
	SYMPTD	3.168	1.614	6.218
	PB	0.754	0.660	0.861
	PB	0.854	0.748	0.975
	HIST	3.657	1.817	7.361
	HIST	2.890	1.381	6.049
	BSE	3.390	1.244	9.240
	BSE	2.613	0.984	6.934

We can see above that the point estimates of parameters and odds ratios generated by the ***%surveyglm*** macro and LOGISTIC Procedure are very close, while the standard errors generated by the macro slightly underestimated those generated by the procedure. This small difference is due to the estimation algorithm. It is important to note that the inference does not change.

CONCLUSIONS

The results presented in Tables 1 and 2, show that, for gamma model, the point estimate generated by ***%surveyglm*** macro is exactly the same of the function *svyglm*, and that the estimates of the standard errors are very close. For the normal distribution, the results of the ***%surveyglm*** macro are very close to the function *svyglm* and it is exactly the same of SURVEYREG Procedure, showing that the estimates generated by the ***%surveyglm*** macro are correct and can be used as an alternative tool for adjust complex sampling design with generalized linear model. When the user does not have information about the sampling design, the ***%surveyglm*** macro reduces to the GENMOD Procedure, as shown in Table 3.

Another advantage of the ***%surveyglm*** macro is that other models that belong to the generalized linear models or quasi-likelihood models are easily incorporated into the macro just using the mean-variance relationship. Through this macro, SAS® users can adjust other models of the generalized linear models using the sample design, which was not possible until now.

REFERENCES

Chambers, R. L. and Skinner, C. J., Analysis of Survey Data, First Edition - London, John Wiley, 2003.

Cochran, W. G., Sampling Techniques, Third Edition - New York, John Wiley, 1977.

Hosmer, D. W. and Lemeshow, S., Applied Logistic Regression, 2nd edition, 2000.

Lee, E. T., A Computer Program for Linear Logistic Regression Analysis. Computer Programs in Biomedicine, 80-92, 1974.

Lohr, S. L., Sampling: Design and Analysis, Second Edition - Pacific Grove, CA, Duxbury Press, 2009.

Lumley, T., Analysis of complex survey samples, Journal of statistical software, 9, 1-19, 2004.

McCullagh, P. and Nelder, J.A., Generalized Linear Models, London, Chapman & Hall, 1989.

Morel, J.G., Logistic Regression under Complex Survey Designs, Survey Methodology, 15, 203-223, 1989.

Nelder, J. A., and Wedderburn, R. W. M., Generalized Linear Models, Journal of the Royal Statistical Society, Ser. A, 135, 370-384, 1972.

SAS Institute, Inc., SAS/STAT 9.3 User's Guide, Cary, NC: SAS Institute, Inc., 2011.

CONTACT INFORMATION

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

Name: Paulo Henrique Dourado da Silva
Enterprise: Universidade de Brasília
Address: Campus Universitário Darcy Ribeiro, Departamento de Estatística, Prédio CIC/EST sala A1 35/28
City, State ZIP: Brasília, DF, Brazil, 70910-900
Work Phone: +5561 3107 3672
Fax: +5561 3107 3672
E-mail: phd_nank@hotmail.com
Web: www.est.unb.br

Name: Alan Ricardo da Silva
Enterprise: Universidade de Brasília
Address: Campus Universitário Darcy Ribeiro, Departamento de Estatística, Prédio CIC/EST sala A1 35/28
City, State ZIP: Brasília, DF, Brazil, 70910-900
Work Phone: +5561 3107 3672
Fax: +5561 3107 3672
E-mail: alansilva@unb.br
Web: www.est.unb.br

APPENDIX I – SAS® MACRO

```
%macro surveyglm(base=, y=, x=, weight=, strata=, cluster=, fpc=, dist=normal, link=inv, intercept=y, scale=deviance, vadjust=n, alpha=0.05, delta=1, fi=0.5, maxiter=100, eps=0.000001);
%if &base= %then %do;
	%put ERROR: The database must be informed;
%end; %else %if &y= %then %do;
	%put ERROR: The response variable must be informed;
%end; %else %if &x= %then %do;
	%put ERROR: The covariates must be informed;
%end; %else %if (%upcase(&dist)=NORMAL | %upcase(&dist)=GAMMA | %upcase(&dist)=IG) & &scale= %then %do;
	%put ERROR: The scale estimation method (DEVIANCE or PEARSON method) must be informed;
%end; %else %do;
/*Sorting data*/
%if %upcase(&strata) NE & %upcase(&cluster)= %then %do;
proc sort data=&base;by &strata descending &y;run;
%end; %if %upcase(&cluster) NE & %upcase(&strata)= %then %do;
proc sort data=&base;by &cluster descending &y;run;
%end; %if %upcase(&cluster) NE & %upcase(&strata)= NE %then %do;
proc sort data=&base;by &strata &cluster descending &y;run;
%end;
/*Default link functions*/
%if %upcase(&link)= & %upcase(&dist)=POISSON %then %do;
	%let link=log;
%end;
%else; %if %upcase(&link)=%upcase(&link) & %upcase(&dist)=POISSON %then %do;
	%let link=&link;
%end;
%if %upcase(&link)= & %upcase(&dist)=BINOMIAL %then %do;
	%let link=logit;
%end;
%else; %if %upcase(&link)=%upcase(&link) & %upcase(&dist)=BINOMIAL %then %do;
	%let link=&link;
%end;
%if %upcase(&link)= & %upcase(&dist)=GAMMA %then %do;
	%let link=inv;
%end;
%else; %if %upcase(&link)=%upcase(&link) & %upcase(&dist)=GAMMA %then %do;
	%let link=&link;
%end;
%if %upcase(&link)= & %upcase(&dist)=IG %then %do;
	%let link=inv2;
%end;
%else; %if %upcase(&link)=%upcase(&link) & %upcase(&dist)=IG %then %do;
	%let link=&link;
%end;
%if %upcase(&link)= & %upcase(&dist)=NORMAL %then %do;
	%let link=identity;
%end;
%else; %if %upcase(&link)=%upcase(&link) & %upcase(&dist)=NORMAL %then %do;
	%let link=&link;
%end;
proc iml;
	MaxIter=&maxiter;
	eps=&eps;
*Input data;
	use &base;
		read all var{&y} into y[colname=depname];
		read all var{&x} into x0[colname=varname];

		%if %upcase(&weight) NE %then %do;
		read all var{&weight} into w[colname=weightname];
		%end;
		%if %upcase(&strata) NE %then %do;
		read all var{&strata} into h1;
		%end;
		%if %upcase(&fpc) NE %then %do;
		read all var{&fpc} into fh;
		%end;
		%if %upcase(&cluster) NE %then %do;
		read all var{&cluster} into i;
		%end;
	close &base;

	X=x0;
	%if %upcase(&intercept)=Y %then %do;
	X=repeat(1,nrow(y),1)||x0;
	%end;

	n=nrow(X);
	p=ncol(X)*ncol(y);

	%if %upcase(&weight) NE %then %do;
	weight=w;
	%end; %else %do;
	w=repeat(1, n, 1);
	weight=w;
	%end;

	%if %upcase(&fpc)= %then %do;
	fh=repeat(0, n, 1);
	%end

	tab=y||x||w;
	%if %upcase(&dist)=BINOMIAL %then %do;
	tab=y||x||w;
	tabb=y||x||w;
	call sort(tabb,{1});
	freq=j(1,2,1);
	do j=2 to nrow(tabb);
		if tabb[j,1]=tabb[j-1,1] then do;
		freq[nrow(freq),1]=tabb[j-1,1];

	freq[nrow(freq),2]=freq[nrow(freq),2]+1;
	end;
	else freq=freq//j(1,2,1);
	end;
	freq=(1:nrow(freq))`||freq;
	mi=1;
	y1=y;
	yd=j(nrow(y),nrow(freq)-1,0);
```

```
        do f=2 to nrow(freq);
                do ff=1 to nrow(y);
                        if      y[ff]=freq[f,2]
then yd[ff,f-1]=1;
                        else yd[ff,f-1]=0;
                end;
        end;
        y=yd;
        %end;
        ni=ncol(unique(i));
        nh=ncol(unique(h1));

        sw=w[+];
        smy=sum(y#w);
        my=smy/sw;

        title3 "The SURVEYGLM Macro";
        %if %upcase(&dist) NE BINOMIAL %then
%do;
        report=n//sw//my//smy;
        reportname={"Number of Observations"
"Sum of Weights" "Weighted Mean of &y."
"Weighted Sum of &y."};
        print   report[label="Data   Summary"
rowname=reportname];
        %end;

*GLM routine;
start GLM;
        * Get initial values;
        %if %upcase(&link)=INV %then %do;
    yc = y + .000000000001;
        y_trans = 1/yc;
        %end;  %if  %upcase(&link)=LOG  %then
%do;
        yc = y + .000000000001;
        y_trans=log(yc);
        %end;
        %if %upcase(&link)=INV2  %then %do;
        yc = y + .000000000001;
        y_trans = sqrt(1/yc);
        %end;
        %if  %upcase(&link)=IDENTITY    %then
%do;
        yc = y;
        y_trans=yc;
        %end;
        %if %upcase(&link)=LOGIT  %then %do;
        yc=j(nrow(y), ncol(y),0);
        do f=1 to ncol(y);
                yc[,f]=y[+,f]/n;
        end;
                y_trans = log(yc/(1-yc[,+]));
        %end;

        %if %upcase(&link)=INV2  %then %do;
                b=repeat(.000000000001,p,1);
        %end; %else %do;
        b=j(ncol(x), ncol(y),0);
        do ii=1 to ncol(y);

        b[,ii]=inv(X`*(X#weight))*X`*(y_trans
[,ii]#weight);
        end;
        %end;
        phi=repeat(1, n, 1);
        eta=j(nrow(y), ncol(y),0);
        mu=j(nrow(y),ncol(y),0);
        si=j(ncol(y)*ncol(x), nrow(y),0);
        we=j(nrow(y), ncol(y),0);
        wo=j(nrow(y), ncol(y),0);
        H=j(ncol(y)*ncol(x),
ncol(y)*ncol(x),0);
        *IRLS algorithm;
        do      iter=1      to      MaxIter
until(sum(abs((b-
oldb)/(oldb+0.0000001)))<eps);
                *oldb=shapecol(b,
ncol(x),ncol(y));
                oldb=T(shape(b`,
ncol(y),ncol(x)));
                do ii=1 to ncol(y);
                        eta[,ii]=X*oldb[,ii];
                end;

                /*Link functions*/
                %if  %upcase(&link)=INV  %then
%do;

        eta=choose(eta=0,0.0000000000001,eta)
;
                mu=1/eta;
                /*Firt   derivative   -   link
function*/
                gderiv=-1/mu##2;
                /*Second  derivative  -  link
function*/
                gderiv2=2/(mu##3);
                %end;
                %if  %upcase(&link)=LOG  %then
%do;
                mu=exp(eta);
                /*Firt   derivative   -   link
function*/
                gderiv=1/mu;
                /*Second  derivative  -  link
function*/
                gderiv2=-1/(mu##2);
                %end;
                %if    %upcase(&link)=IDENTITY
%then %do;
                mu=eta;
                /*Firt   derivative   -   link
function*/
                gderiv=1;
                /*Second  derivative  -  link
function*/
                gderiv2=0;
                %end;
                %if  %upcase(&link)=INV2  %then
%do;

        eta=choose(eta<0,0.0000000000001,eta)
;
                mu=sqrt(1/eta);
                /*Firt   derivative   -   link
function*/
                gderiv=-2/(mu##3);
                /*Second  derivative  -  link
function*/
                gderiv2=6/(mu##4);
                %end;
                %if %upcase(&link)=LOGIT %then
%do;
                do ii=1 to ncol(y);

        eta[,ii]=exp(X*oldb[,ii]);
                end;
                mu=mi#(eta/(1+eta[,+]));
                /*Firt   derivative   -   link
function*/
                gderiv=mi/(mu#(mi-mu));
                /*Second  derivative  -  link
function*/
                gderiv2=(mi#(2#mu-
mi))/((mu#(mi-mu))##2);
                %end;
```

```
		%if %upcase(&dist)=GAMMA %then
%do;
		/*Variance function*/
		vmu=mu##2;
		/*Firt derivative - variance
function*/
		vmuderiv=2#mu;
		%end;			%if
%upcase(&dist)=NORMAL %then %do;
		/*Variance function*/
		vmu=1;
		/*Firt derivative - variance
function*/
		vmuderiv=0;
		%end;			%if
%upcase(&dist)=POISSON %then %do;
		/*Variance function*/
		vmu=mu;
		/*Firt derivative - variance
function*/
		vmuderiv=1;
		%end;  %if  %upcase(&dist)=IG
%then %do;
		/*Variance function*/
		vmu=mu##3;
		/*Firt derivative - variance
function*/
		vmuderiv=3#(mu##2);
		%end;			%if
%upcase(&dist)=BINOMIAL  %then %do;
		/*Variance function*/
		vmu=(1/mi)#mu#(1-mu);
		/*Firt derivative - variance
function*/
		vmuderiv=(1/mi)#(1-2#mu);
		%end;

		/*Gradient vector*/
		do ii=1 to ncol(y);
			m1=(ii-1)*ncol(x)+1;
			m2=ii*ncol(x);
			si[m1:m2,]		=
((weight#(y[,ii]-
mu[,ii]))/(vmu[,ii]#gderiv[,ii]#phi))`#X`;
		end;
		s = si[,+];

		/*Weights*/
		do ii=1 to ncol(y);

	we[,ii]=weight/(phi#vmu[,ii]#(gderiv[
,ii]##2));

	wo[,ii]=we[,ii]+weight#(y[,ii]-
mu[,ii])#((vmu[,ii]#gderiv2[,ii]+vmuderiv[,i
i]#gderiv[,ii])/((vmu[,ii]##2)#(gderiv[,ii]#
#3)#phi));
		end;

		/*Hessian matrix*/
		do ii=1 to ncol(y);
			m1=(ii-1)*ncol(x)+1;
			m2=ii*ncol(x);
			H[m1:m2,	m1:m2]=-
(X#wo[,ii])`*X;
		end;

		*oldb=shapecol(oldb,0,1);
		oldb=T(shape(oldb`,1,0));

		b=oldb-inv(H)*s;
	end;
finish;
run GLM;
	*b1=shapecol(b, ncol(x),ncol(y));
	b1=T(shape(b`, ncol(y),ncol(x)));
	%if  %upcase(&dist)=BINOMIAL  %then
%do;
	*b                                    =
shapecol(shapecol(b,ncol(x),ncol(y))`,ncol(x
)*ncol(y),1);
	b    =    T(shape(T(shape(b`,ncol(y),
ncol(x)))`,1,ncol(x)*ncol(y)));
	%end;
	/*Predict and residuals*/
	do ii=1 to ncol(y);
		eta[,ii]=X*b1[,ii];
	end;

	e=y-mu;
	e2=j(nrow(x), ncol(y)*ncol(x),0);
	do ii=1 to ncol(y);
		e2[,(ii-
1)*ncol(x)+1:ii*ncol(x)]=(phi#we#e[,ii]#gder
iv)#X;
	end;
	G=j(ncol(x)*ncol(y),  ncol(x)*ncol(y),
0);
	/*G matrix*/
	*Strata-Cluster;
	if nh>0 & ni>0 then do;
		_h1_=j(nrow(h1),1,0);
		_h1_[1]=1;
		_i_=j(nrow(i),1,0);
		_i_[1]=1;
		do k=2 to nrow(h1);
			if  h1[k]=h1[k-1]  then
_h1_[k]=_h1_[k-1];
			else    _h1_[k]=_h1_[k-
1]+1;
			if   i[k]=i[k-1]   then
_i_[k]=_i_[k-1];
			else      _i_[k]=_i_[k-
1]+1;
		end;
		tab=tab|| h1 || i ;
		do ii=1 to ncol(y);
			m1=(ii-1)*ncol(x)+1;
			m2=ii*ncol(x);

			ehi=j(1,ncol(x)+4,1);

	ehi[1,3:2+ncol(x)]=e2[1,(ii-
1)*ncol(x)+1:ii*ncol(x)];

	ehi[1]=tab[1,ncol(tab)-1];

	ehi[2]=tab[1,ncol(tab)];
			ehi[ncol(ehi)-
1]=fh[1];
			do j=2 to nrow(tab);
				if
tab[j,ncol(tab)-1]=tab[j-1,ncol(tab)-1]     &
tab[j,ncol(tab)]=tab[j-1,ncol(tab)] then do;

	ehi[nrow(ehi),3:2+ncol(x)]=ehi[nrow(e
hi),3:2+ncol(x)]+e2[j,(ii-
1)*ncol(x)+1:ii*ncol(x)];

	ehi[nrow(ehi),ncol(ehi)]=ehi[nrow(ehi
),ncol(ehi)]+1;

	ehi[nrow(ehi),ncol(ehi)-
1]=(ehi[nrow(ehi),ncol(ehi)-1]+fh[j])/2;
				end;
				else
ehi=ehi//(tab[j,ncol(tab)-
1]||tab[j,ncol(tab)]||e2[j,(ii-
1)*ncol(x)+1:ii*ncol(x)]||fh[j]||j(1,1,1));
			end;
```

```
                        ee=ehi[,ncol(ehi)-
1:ncol(ehi)];
                        ehi=ehi[,1:ncol(ehi)-
2]||j(nrow(ehi),3+ncol(x),0);

        ehi[1,3+ncol(x):2+2*ncol(x)]=ehi[1,3:
2+ncol(x)];
                        ehi[,ncol(ehi)-
2:ncol(ehi)-1]=ee;

        ehi[1,ncol(ehi)]=ehi[1,ncol(ehi)-1];
                        count=0;
                        do j=2 to nrow(ehi);
                                if
ehi[j,1]=ehi[j-1,1] then do;

        ehi[j,3+ncol(x):2+2*ncol(x)]=ehi[j-
1,3+ncol(x):2+2*ncol(x)]+ehi[j,3:2+ncol(x)];

        count=count+1;

        ehi[j,ncol(ehi)]=ehi[j-
1,ncol(ehi)]+ehi[j,ncol(ehi)-1];

        ehi[j,ncol(ehi)-2]=ehi[j-1,ncol(ehi)-
2];
                                end;
                                else do;
                                        if
ehi[j-1,1]=1 then do;

        ehi[1:count+1,ncol(ehi)-1]=count+1;

        ehi[1:count+1,3+ncol(x):2+2*ncol(x)]=
repeat(ehi[j-1,3+ncol(x):2+2*ncol(x)]/ehi[j-
1,ncol(ehi)-1],count+1);

        in=ehi[j,2];

        ehi[1:count+1,ncol(ehi)]=repeat(ehi[j
-1,ncol(ehi)],count+1);
                                        end;
                                        else do;

        ehi[in:in+count,ncol(ehi)-1]=count+1;

        ehi[in:in+count,3+ncol(x):2+2*ncol(x)
]=repeat(ehi[j-
1,3+ncol(x):2+2*ncol(x)]/ehi[j-1,ncol(ehi)-
1],count+1);

        ehi[in:in+count,ncol(ehi)]=repeat(ehi
[j-1,ncol(ehi)],count+1);

        in=ehi[j,2];
                                        end;

        ehi[j,3+ncol(x):2+2*ncol(x)]=ehi[j,3:
2+ncol(x)];

        ehi[j,ncol(ehi)]=ehi[j,ncol(ehi)-1];
                                        count=0;
                        end;
                        if   j=nrow(ehi)   then
do;

        ehi[in:in+count,ncol(ehi)-1]=count+1;

        ehi[in:in+count,3+ncol(x):2+2*ncol(x)
]=repeat(ehi[j,3+ncol(x):2+2*ncol(x)]/ehi[j,
ncol(ehi)-1],count+1);

        ehi[in:in+count,ncol(ehi)]=repeat(ehi
[j,ncol(ehi)],count+1);
                                in=ehi[j,2];
                        end;
                        do       jj=1       to
nrow(ehi);if    ehi[jj,ncol(ehi)-1]=1    then
do;ehi[jj,ncol(ehi)-
1]=2;ehi[jj,3:2+ncol(x)]=0;ehi[jj,3+ncol(x):
2+2*ncol(x)]=0;end;end;end;
                                G[m1:m2,
m1:m2]=((n-1)/(n-
ncol(x)))*(((ehi[,ncol(ehi)-
1]/(ehi[,ncol(ehi)-1]-
1))#(ehi[,3:2+ncol(x)]-
ehi[,3+ncol(x):2+2*ncol(x)]))`*(ehi[,3:2+nco
l(x)]-ehi[,3+ncol(x):2+2*ncol(x)]));
                                %if
%upcase(&fpc)^= %then %do;
                                G[m1:m2,
m1:m2]=((n-1)/(n-
ncol(x)))*((((ehi[,ncol(ehi)-1]#(1-
ehi[,ncol(ehi)-2]))/(ehi[,ncol(ehi)-1]-
1))#(ehi[,3:(2+ncol(x))]-
ehi[,3+ncol(x):2+2*ncol(x)]))`*(ehi[,3:(2+nc
ol(x))]-ehi[,3+ncol(x):2+2*ncol(x)]));
                                %end;
                                *print G;
                end;
                %if     %upcase(&dist)=BINOMIAL
%then %do;

        Report="%upcase(&base)"//"%upcase(&di
st)"//"%upcase(&link)"//"&y"//"&strata"//"&c
luster"
                        //"&weight"//"Ridge
Stabilized Newton-Raphson"//"Taylor Series";
                ReportName={"Data          Set"
"Distribution"   "Link   Function"   "Dependent
Variable"
                                "Stratum
Variable"      "Cluster   Variable"   "Weight
Variable"
                                "Optimization
Technique" "Variance Estimation Method"};
                print       Report[label="Model
Information" rowname=reportname];
                print
(nrow(freq))[label='Number     of     Response
Levels'];


        report=(ehi[nrow(ehi),1])//(ehi[nrow(
ehi),2]);
                reportnames={"Number         of
Strata", "Number of Clusters"};
                print      report[label="Design
Summary" rowname=reportnames];

                %end; %else %do;

        Report="%upcase(&base)"//"%upcase(&di
st)"//"%upcase(&link)"//"&y"//"&strata"//
                "&cluster"//"&weight"//"Ridge
Stabilized Newton-Raphson"//"Taylor Series";
                ReportName={"Data          Set"
"Distribution"   "Link   Function"   "Dependent
Variable"
                                "Stratum
Variable"      "Cluster   Variable"   "Weight
Variable"
                                "Optimization
Technique" "Variance Estimation Method"};

                print       Report[label="Model
Information" rowname=reportname];
```

```
        report=(ehi[nrow(ehi),1])//(ehi[nrow(
ehi),2]);
                reportnames={"Number        of
Strata", "Number of Clusters"};
                print     report[label="Design
Summary" rowname=reportnames];

        %end;
        df=ehi[nrow(ehi),2]-ehi[nrow(ehi),1];
        end;

        *Strata;
        else if nh>0 & ni=0 then do;
                _h1_=j(nrow(h1),1,0);
                _h1_[1]=1;
                do k=2 to nrow(h1);
                        if  h1[k]=h1[k-1]  then
_h1_[k]=_h1_[k-1];
                        else    _h1_[k]=_h1_[k-
1]+1;
                end;
        tab=tab|| h1 ;
    do ii=1 to ncol(y);
                ehi=j(1,2*ncol(x)+2,0);
                ehi[1,2:1+ncol(x)]=e2[1,(ii-
1)*ncol(x)+1:ii*ncol(x)];

        ehi[1]=tab[1,ncol(tab)];ehi[ncol(ehi)
]=1;
                do j=2 to nrow(tab);
                        if
tab[j,ncol(tab)]=tab[j-1,ncol(tab)] then do;

        ehi[nrow(ehi),2:1+ncol(x)]=ehi[nrow(e
hi),2:1+ncol(x)]+e2[j,(ii-
1)*ncol(x)+1:ii*ncol(x)];

        ehi[nrow(ehi),ncol(ehi)]=ehi[nrow(ehi
),ncol(ehi)]+1;
                        end;
                        else
do;ehi=ehi//(tab[j,ncol(tab)]||e2[j,]||j(1,n
col(x)+1,1));end;
                end;

        ehi[,2+ncol(x):1+2*ncol(x)]=ehi[,2:1+
ncol(x)]/ehi[,ncol(ehi)];
                do jj=1 to nrow(ehi);

        ehi2=ehi2//repeat(ehi[jj,2+ncol(x):2+
2*ncol(x)],ehi[jj,ncol(ehi)]);
                end;
                eh=e2[,(ii-
1)*ncol(x)+1:ii*ncol(x)]||ehi2[,1:ncol(ehi2)
-1]||fh||ehi2[,ncol(ehi2)];
                        G[m1:m2,    m1:m2]=((n-
1)/(n-
ncol(x)))*(((eh[,ncol(eh)]/(eh[,ncol(eh)]-
1))#(eh[,1:ncol(x)]-
eh[,1+ncol(x):2*ncol(x)]))`*(eh[,1:ncol(x)]-
eh[,1+ncol(x):2*ncol(x)]));
                        %if  %upcase(&fpc)  NE
%then %do;
                        G[m1:m2,    m1:m2]=((n-
1)/(n-ncol(x)))*((((eh[,ncol(eh)]#(1-
eh[,ncol(eh)-1]))/(eh[,ncol(eh)]-
1))#(eh[,1:ncol(x)]-
eh[,1+ncol(x):2*ncol(x)]))`*(eh[,1:ncol(x)]-
eh[,1+ncol(x):2*ncol(x)]));
                        %end;
                        *print G;
                free ehi2;
        end;

        %if    %upcase(&dist)=BINOMIAL    %then
%do;

        Report="%upcase(&base)"//"%upcase(&di
st)"//"%upcase(&link)"//"&y"//"&strata"
                        //"&weight"//"Ridge
Stabilized Newton-Raphson"//"Taylor Series";
                ReportName={"Data          Set"
"Distribution"   "Link   Function"   "Dependent
Variable"
                                "Stratum
Variable"    "Weight Variable"
                                "Optimization
Technique" "Variance Estimation Method"};
                print       Report[label="Model
Information" rowname=reportname];
                print
(nrow(freq))[label="Number     of     Response
Levels"];

                report=(ehi[<>,1]);
                reportnames={"Number        of
Strata"};
                print     report[label="Design
Summary" rowname=reportnames];

                %end; %else %do;

        Report="%upcase(&base)"//"%upcase(&di
st)"//"%upcase(&link)"//"&y"//"&strata"//
                "&weight"//"Ridge   Stabilized
Newton-Raphson"//"Taylor Series";
                ReportName={"Data          Set"
"Distribution"   "Link   Function"   "Dependent
Variable"
                                "Stratum
Variable"    "Weight Variable"
                                "Optimization
Technique" "Variance Estimation Method"};

                print       Report[label="Model
Information" rowname=reportname];

                report=(ehi[<>,1]);
                reportnames={"Number        of
Strata"};
                print     report[label="Design
Summary" rowname=reportnames];

        %end;
        df=n-ehi[<>,1];
end;
        *Cluster;
        else if ni>0 & nh=0 then do;
                 i =j(nrow(i),1,0);
                _i_[1]=1;
                do k=2 to nrow(i);
                        if   i[k]=i[k-1]   then
_i_[k]=_i_[k-1];
                        else      _i_[k]=_i_[k-
1]+1;
                end;
                tab=tab||_i_;
            do ii=1 to ncol(y);
                        m1=(ii-1)*ncol(x)+1;
                        m2=ii*ncol(x);

                        ehi=j(1,ncol(x)+3,0);

        ehi[1,2:1+ncol(x)]=e2[1,(ii-
1)*ncol(x)+1:ii*ncol(x)];

        ehi[1]=tab[1,ncol(tab)];ehi[ncol(ehi)
-1]=fh[1];ehi[1,ncol(ehi)]=1;
                        do j=2 to nrow(tab);
```

```
                                if
tab[j,ncol(tab)]=tab[j-1,ncol(tab)] then do;
ehi[nrow(ehi),2:1+ncol(x)]=ehi[nrow(ehi),2:1
+ncol(x)]+e2[j,(ii-
1)*ncol(x)+1:ii*ncol(x)];ehi[nrow(ehi),ncol(
ehi)-1]=(ehi[nrow(ehi),ncol(ehi)-
1]+fh[j])/2;ehi[nrow(ehi),ncol(ehi)]=ehi[nro
w(ehi),ncol(ehi)]+1;
                                end;  else  do;
ehi=ehi//(tab[j,ncol(tab)]||e2[j,(ii-
1)*ncol(x)+1:ii*ncol(x)]||fh[j]||j(1,1,1));
end;
                        end;

                                G[m1:m2,
m1:m2]=((n-1)/(n-
ncol(x)))*(ehi[<>,1]/(ehi[<>,1]-
1))*((ehi[,2:1+ncol(x)]-
ehi[+,2:1+ncol(x)]/ehi[+,ncol(ehi)])`*(ehi[,
2:1+ncol(x)]-
ehi[+,2:1+ncol(x)]/ehi[+,ncol(ehi)]));
                                %if
%upcase(&fpc) NE %then %do;
                                G[m1:m2,
m1:m2]=((n-1)/(n-
ncol(x)))*(ehi[<>,1]/(ehi[<>,1]-1))*(((1-
ehi[,ncol(ehi)-1])#(ehi[,2:1+ncol(x)]-
ehi[+,2:1+ncol(x)]/ehi[+,ncol(ehi)]))`*(ehi[
,2:1+ncol(x)]-
ehi[+,2:1+ncol(x)]/ehi[+,ncol(ehi)]));
                                %end;
                                *print G;
                        end;
                        %if
%upcase(&dist)=BINOMIAL %then %do;

        Report="%upcase(&base)"//"%upcase(&di
st)"//"%upcase(&link)"//"&y"//"&cluster"

        //"&weight"//"Ridge         Stabilized
Newton-Raphson"//"Taylor Series";

        ReportName={"Data Set" "Distribution"
"Link Function" "Dependent Variable"
                                "Cluster
Variable" "Weight Variable"
                                "Optimization
Technique" "Variance Estimation Method"};

                                print
Report[label="Model             Information"
rowname=reportname];


        report=(ehi[<>,1]);

        reportnames={"Number of Clusters"};
                                print
report[label="Design                Summary"
rowname=reportnames];

                        %end; %else %do;

        Report="%upcase(&base)"//"%upcase(&di
st)"//"%upcase(&link)"//"&y"//

        "&cluster"//"&weight"//"Ridge
Stabilized Newton-Raphson"//"Taylor Series";

        ReportName={"Data Set" "Distribution"
"Link Function" "Dependent Variable"
                                "Cluster
Variable" "Weight Variable"
                                "Optimization
Technique" "Variance Estimation Method"};

                                print
Report[label="Model             Information"
rowname=reportname];


        report=(ehi[<>,1]);

        reportnames={"Number of Clusters"};
                                print
report[label="Design                Summary"
rowname=reportnames];
                        %end;
                        df=ehi[<>,1]-1;
                        end;
                        else do;
                                do    ii=1    to
ncol(y);
                                m1=(ii-
1)*ncol(x)+1;
                                m2=ii*ncol(x);

                                G[m1:m2,
m1:m2]=-H[(ii-1)*ncol(x)+1:ii*ncol(x),   (ii-
1)*ncol(x)+1:ii*ncol(x)];


                                end;
                                %if
%upcase(&dist)=BINOMIAL %then %do;

        Report="%upcase(&base)"//"%upcase(&di
st)"//"%upcase(&link)"//"&y"

        //"&weight"//"Ridge         Stabilized
Newton-Raphson"//"Taylor Series";

        ReportName={"Data Set" "Distribution"
"Link Function" "Dependent Variable"
                                "Weight
Variable" "Optimization Technique" "Variance
Estimation Method"};
                                print
Report[label="Model             Information"
rowname=reportname];


                                %end;     %else
%do;

        Report="%upcase(&base)"//"%upcase(&di
st)"//"%upcase(&link)"//"&y"//

        "&weight"//"Ridge  Stabilized  Newton-
Raphson"//"Taylor Series";

        ReportName={"Data Set" "Distribution"
"Link Function" "Dependent Variable"
                                "Weight
Variable" "Optimization Technique" "Variance
Estimation Method"};
                                print
Report[label="Model             Information"
rowname=reportname];
                                %end;
                                df=nrow(y)-
ncol(x);
                        end;

                        %if
%upcase(&dist)=BINOMIAL %then %do;
                                print
freq[colname={"Order"    "&y"    "Frequency"}
label='Response Profile'],
```

```
                              "Logits modeled
use   &y   ="(trim(left(char(freq[1,2])))))"as
the reference category.";
                        %end;

        %if   %upcase(&scale)=DEVIANCE   %then
%do;
        /*Deviance and Scale deviance*/
                %if      %upcase(&dist)=NORMAL
%then %do;

        deviance=sum(weight#(y-mu)##2);
                %end;                          %if
%upcase(&dist)=POISSON %then %do;

        mu=choose(mu<=0,0.0000000000001,mu);

        deviance=2#sum(weight#(yc#log(yc/mu)-
(yc-mu)));
                %end;   %if   %upcase(&dist)=IG
%then %do;

        mu=choose(mu=0,0.0000000000001,mu);

        deviance=sum(weight#((yc-
mu)##2)/(yc#mu##2));
                %end;                          %if
%upcase(&dist)=BINOMIAL %then %do;

        deviance=2#sum(weight#mi#(yc#log(yc/m
u))+(1-yc)#log((1-yc)/(1-mu)));
                %end;
                scale1=deviance/df;
                %if  &strata  NE  |  &cluster  NE
%then %do;
                        scale1=n#deviance/((n-
p)#sum(weight));
                %end;
                scaled=sqrt(scale1);
                %if %upcase(&dist)=GAMMA %then
%do;

        mu=choose(mu<=0,0.0000000000001,mu);

        deviance=2#sum(weight#(-
log(yc/mu)+(yc-mu)/mu));
                        scale1=deviance/df;
                %if  &strata  NE  |  &cluster  NE
%then %do;
                        scale1=n#deviance/((n-
p)#sum(weight));
                %end;
                        scaled=1/scale1;
                %end;
                sdeviance=deviance/scale1;

                /*Pearson and Scale pearson*/
                pearson=j(ncol(y),1,0);
                do ii=1 to ncol(y);

        pearson[ii,]=sum((weight#(y[,ii]-
mu[,ii])##2)/vmu[,ii]);
                end;
                pearson=sum(pearson);
                scale2=pearson/df;
                %if  &strata  NE  |  &cluster  NE
%then %do;
                scale2=n#pearson/((n-
p)#sum(weight));
                %end;
                spearson=pearson/scale1;

                valeudf=sdeviance/df;
                valeudf1=spearson/df;
                dev=df||deviance||scale1;
                sdev=df||sdeviance||valeudf;
                pcs=df||pearson||scale2;
                sp=df||spearson||valeudf1;

                phi=scale1;
        %end;   %if     %upcase(&scale)=PEARSON
%then %do;
                /*Pearson*/
                pearson=j(ncol(y),1,0);
                do ii=1 to ncol(y);

        pearson[ii,]=sum((weight#(y[,ii]-
mu[,ii])##2)/vmu[,ii]);
                end;
                pearson=sum(pearson);
                scale1=pearson/df;
                %if  &strata  NE  |  &cluster  NE
%then %do;
                scale1=n#pearson/((n-
p)#sum(weight));
                %end;
                /*Deviance and Scale pearson*/
                %if      %upcase(&dist)=NORMAL
%then %do;

        deviance=sum(weight#(y-mu)##2);
                %end;                          %if
%upcase(&dist)=POISSON %then %do;

        mu=choose(mu<=0,0.0000000000001,mu);

        deviance=2#sum(weight#(yc#log(yc/mu)-
(yc-mu)));
                %end;   %if   %upcase(&dist)=IG
%then %do;

        mu=choose(mu=0,0.0000000000001,mu);

        deviance=sum(weight#((yc-
mu)##2)/(yc#mu##2));
                %end;                          %if
%upcase(&dist)=BINOMIAL %then %do;

        deviance=2*sum(weight#mi#(yc#log(yc/m
u))+(1-y)#log((1-yc)/(1-mu)));
                %end;
                scale2=deviance/df;
                %if  &strata  NE  |  &cluster  NE
%then %do;
                        scale2=n*deviance/(n-
p)#sum(weight);
                %end;
                scaled=sqrt(scale1);
                %if %upcase(&dist)=GAMMA %then
%do;

        mu=choose(mu<=0,0.0000000000001,mu);

        deviance=2#sum(weight#(-
log(yc/mu)+(yc-mu)/mu));
                        scale2=deviance/df;
                        %if    &strata    NE    |
&cluster NE %then %do;

        scale2=n*deviance/(n-p)#sum(weight);
                        %end;
                        scaled=1/scale1;
                %end;
                spearson=pearson/scale1;
                sdeviance=deviance/scale1;

                valeudf=sdeviance/df;
                valeudf1=spearson/df;
```

```
                dev=df||deviance||scale2;
                sdev=df||sdeviance||valeudf;
                pcs=df||pearson||scale1;
                sp=df||spearson||valeudf1;

                phi=scale1;
        %end;
        %else;   %if   %upcase(&scale)=   %then
%do;
        /*Deviance and Scale deviance*/
        %if %upcase(&dist)=POISSON %then %do;

        mu=choose(mu<=0,0.0000000000001,mu);

        deviance=2#sum(weight#(yc#log(yc/mu)-
(yc-mu)));
        %end;    %if    %upcase(&dist)=BINOMIAL
%then %do;

        deviance=2#sum(weight#mi#(yc#log(yc/m
u))+(1-yc)#log((1-yc)/(1-mu)));
        %end;
        scale1=deviance/df;
        %if  &strata  NE  |  &cluster  NE  %then
%do;
                scale1=n*deviance/((n-
p)#sum(weight));
        %end;
        sdeviance=deviance/scale1;
        /*Pearson and Scale pearson*/
        pearson=j(ncol(y),1,0);
        do ii=1 to ncol(y);

        pearson[ii,]=sum((weight#(y[,ii]-
mu[,ii])##2)/vmu[,ii]);
        end;
        pearson=sum(pearson);
        scale2=pearson/df;
        %if  &strata  NE  |  &cluster  NE  %then
%do;
                scale2=n*pearson/((n-
p)#sum(weight));
        %end;
        spearson=pearson/scale1;
        valeudf=sdeviance/df;
        valeudf1=spearson/df;

        dev=df||deviance||scale1;
        sdev=df||sdeviance||valeudf;
        pcs=df||pearson||scale2;
        sp=df||spearson||valeudf1;

        phi=1;
        scaled=1;
        %end;
        /*Covariance*/

                /*Weights*/
                do ii=1 to ncol(y);

        we[,ii]=weight/(phi#vmu[,ii]#gderiv[,
ii]##2);

        wo[,ii]=we[,ii]+weight#(y[,ii]-
mu[,ii])#((vmu[,ii]#gderiv2[,ii]+vmuderiv[,i
i]#gderiv[,ii])/((vmu[,ii]##2)#(gderiv[,ii]#
#3)#phi));
                end;
                /*Hessian matrix*/
                do ii=1 to ncol(y);
                        m1=(ii-1)*ncol(x)+1;
                        m2=ii*ncol(x);
                        H[m1:m2,        m1:m2]=-
phi#(X#we[,ii])`*X;
                end;
                if nh=0 & ni=0 then do;
                        G=phi#G;
                end;


        /*Q matrix*/
        Q=-H;
                Sigma=j(ncol(x),ncol(y),0);
                do ii=1 to ncol(y);
                %if  %upcase(&vadjust)=N  %then
%do;

        Sigma[,ii]=vecdiag(ginv(Q[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)])*(G[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)])*ginv(Q[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)]));
                                %end;        %if
%upcase(&vadjust)=Y %then %do;
                                /*Matrix
correction*/

        traceqg=trace(ginv(Q[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)])*G[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)]);

        traceqgp=traceqg/p;
                                k=max(&delta,
traceqgp);
                                lambda=min(&fi,
p/(ehi[nrow(ehi),2]-p));

        correction=(k#lambda)*ginv(Q[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)]);

        Sigma[,ii]=vecdiag(ginv(Q[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)])*G[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)]*ginv(Q[(ii-
1)*ncol(x)+1:ii*ncol(x),                    (ii-
1)*ncol(x)+1:ii*ncol(x)])+correction);
                                %end;
                end;
        /*Standard Error*/
        %if    %upcase(&dist)=BINOMIAL    %then
%do;
                std=sqrt(Sigma);

        *std=shapecol(shapecol(std,ncol(x),nc
ol(y))`,ncol(x)*ncol(y),1);

        std=T(shape(T(shape(std`,ncol(y),
ncol(x)))`,1,ncol(x)*ncol(y)));
        %end; %else %do;
                std=sqrt(Sigma);
        %end;


        %if %upcase(&dist)=GAMMA %then %do;

        lli=(weight/phi)#log(weight#yc/(phi#m
u))-(weight#yc/(phi#mu))-log(yc)-
log(gamma(weight/phi));
                *lli=li;
        %end; %if %upcase(&dist)=NORMAL %then
%do;
                pi=constant("pi");
```

```
                lli=-0.5#((weight#((yc-
mu)##2))/phi + log(phi/weight)+log(2#pi));
                *lli=li;
        %end;    %if    %upcase(&dist)=POISSON
%then %do;
                *li=(weight/phi)#(y#log(mu)-
mu);
                fy=fact(y);
                lli=weight#(y#log(mu)-mu-
log(fy));
        %end;   %if   %upcase(&dist)=IG   %then
%do;
                pi=constant('pi');
                lli=-0.5#((weight#((yc-
mu)##2)/(phi#yc#(mu##2)))+log((phi#(yc##3))/
weight)+log(2#pi));
                *lli=li;
        %end;    %if    %upcase(&dist)=BINOMIAL
%then %do;
                xbetas=0;
                yxbetas=0;
                do ii=1 to ncol(y);

        *li=(weight/phi)#(y#log(mu)+(mi-
y)#log(1-mu));

        *lli=weight#(log(comb(mi,y))+y#log(mu
)+(mi-y)#log(1-mu));
                xbetas=xbetas+exp(X*b1[,ii]);

        yxbetas=yxbetas+y[,ii]#(X*b1[,ii]);
                end;
                lli=sum(weight#(yxbetas-
log(1+xbetas)));
        %end;



        *l=sum(li);
        p=nrow(b);
        ll=-2#sum(lli);
        AIC=.||ll+2#p||.;
        AICC=.||ll+2#p#(n/(n-p-1))||.;
        BIC=.||ll+p#log(n)||.;
        *logl=.||l||.;
        _2flogl=.||ll||.;
        Report=dev//sdev//pcs//sp//_2flogl//A
IC//AICC//BIC;
        ReportName={     "Deviance"     "Scaled
Deviance"    "Pearson    Chi-Square"    "Scaled
Pearson  X2"     "-2  Log  Likelihood"  "AIC"
"AICC" "BIC"};
        ReportcolName={      "DF"      "Value"
"Value/DF"};
        %if    %upcase(&dist)=BINOMIAL    %then
%do;
        AIC=ll+2#p;
        AICC=ll+2#p#(n/(n-p-1));
        BIC=ll+p#log(n);
        _2flogl=ll;
        Report=_2flogl//AIC//AICC//BIC;
        ReportName={"-2 Log Likelihood" "AIC"
"AICC" "BIC"};
        ReportcolName={ "Value"};
        %end;
        print    Report[label="Criteria    For
Assessing       Goodness       Of       Fit"
rowname=reportname      colname=reportcolname
format=12.4];

        /*t test and Wald test*/
        t=b/std;
        pvalue=2#(1-probt(abs(t),df));
        %if    %upcase(&dist)=BINOMIAL    %then
%do;
                wald=(b/std)##2;
                pvalue=1-probchi(wald, 1);
                t=wald;
        %end;
        *b1=b`||scaled;
        *b1=shapecol(b,0,1)`||scaled;
        b1=T(shape(b`,1,0))`||scaled;
        b2=b1`;
        if ncol(y)>1 then do;

        varname1=varname`||repeat(varname`,nc
ol(y)-1);

        varname1=rowvec(varname1)`//"Scale";
        end; else do;
                varname1=varname`//"Scale";
        end;
        %if %upcase(&intercept)=Y %then %do;
        if ncol(y)>1 then do;

        varname1="Intercept"//varname`||repea
t("Intercept"//varname`,ncol(y)-1);

        varname1=rowvec(varname1)`//"Scale";
        end; else do;

        varname1="Intercept"//varname`//"Scal
e";
        end;
        %end;
        print "Analysis of Maximum Likelihood
Estimates";
        %if    %upcase(&dist)=BINOMIAL    %then
%do;

        varnamey=repeat(freq[2:nrow(freq),2],
ncol(x));
                print
varname1[label="Parameter"]
varnamey[label="&y"]      b2[label="Estimate"
format=12.4]    std[label="Standard    Error"
format=12.4]     t[label="Wald    Chi-Square"
format=12.2]
                pvalue[label="Pr    >    ChiSq"
format=pvalue6.4];
        %end; %else %do;
                print
varname1[label="Parameter"]
b2[label="Estimate"               format=12.4]
std[label="Standard    Error"     format=12.4]
t[label="t Value" format=12.2]
                pvalue[label="Pr     >     |t|"
format=pvalue6.4];
        %end;

                %if %upcase(&dist) NE BINOMIAL
%then %do;
                print  "Note:  The  denominator
degrees  of  freedom  for  the  t  tests  is"
df[label=""];
                %end;
        %if    %upcase(&scale)=DEVIANCE    %then
%do;
                %if %upcase(&dist)=GAMMA %then
%do;
                        print "Note: The Gamma
scale    parameter    was    estimated    by
DOF/DEVIANCE.";
                %end;                          %if
%upcase(&dist)=NORMAL                          |
%upcase(&dist)=POISSON | %upcase(&dist)=IG |
%upcase(&dist)=BINOMIAL %then %do;
                        print "Note: The scale
parameter  was  estimated  by  the  square  root
of DEVIANCE/DOF.";
                %end;
```

```
        %end;   %if     %upcase(&scale)=PEARSON
%then %do;
        %if %upcase(&dist)=GAMMA %then %do;
                        print "Note: The Gamma
scale    parameter    was    estimated    by
DOF/Pearson's Chi-Square.";
                %end;                           %if
%upcase(&dist)=NORMAL                           |
%upcase(&dist)=POISSON  |  %upcase(&dist)=IG  |
%upcase(&dist)=BINOMIAL %then %do;
                        print "Note: The scale
parameter was estimated by the square root
of Pearson's Chi-Square/DOF. ";
                %end;
        %end;
        /*Odds Ratio*/
        %if    %upcase(&dist)=BINOMIAL    %then
%do;
        if ncol(y)>1 then do;

        varnameodds=rowvec(varname`||repeat(v
arname`,ncol(y)-1))`;

        varnameoddsy=repeat(freq[2:nrow(freq)
,2],ncol(x));
        end; else do;
                varnameodds=varname`;

        varnameoddsy=repeat(freq[2:nrow(freq)
,2],ncol(x));
        end;

        %if %upcase(&intercept)=Y %then %do;
        if ncol(y)>1 then do;

        varnameodds=rowvec(varname`||repeat(v
arname`,ncol(y)-1))`;

        varnameoddsy=repeat(freq[2:nrow(freq)
,2],ncol(x)-1);
        end; else do;
                varnameodds=varname`;

        varnameoddsy=repeat(freq[2:nrow(freq)
,2],ncol(x)-1);
        end;
        %end;
        %end;
        %if    %upcase(&dist)=BINOMIAL    %then
%do;
                *beta_=shapecol(b,0,1);
                beta_=T(shape(b`,1,0));
                odds_ratio=exp(beta_);

        odds_ratioclm=j(nrow(odds_ratio),2,0)
;

        odds_ratioclm[,1]=exp(beta_[1:nrow(be
ta_)]-probit(1-&alpha/2)*std[1:nrow(std)]);

        odds_ratioclm[,2]=exp(beta_[1:nrow(be
ta_)]+probit(1-&alpha/2)*std[1:nrow(std)]);
        %if %upcase(&intercept)=Y %then %do;

        odds_ratio=exp(beta_[ncol(y)+1:nrow(b
eta_)]);

        odds_ratioclm=j(nrow(odds_ratio),2,0)
;

        odds_ratioclm[,1]=exp(beta_[ncol(y)+1
:nrow(beta_)]-probit(1-
&alpha/2)*std[ncol(y)+1:nrow(std)]);

        odds_ratioclm[,2]=exp(beta_[ncol(y)+1
:nrow(beta_)]+probit(1-
&alpha/2)*std[ncol(y)+1:nrow(std)]);
        %end;
        print "Odds Ratio Estimates";
        print     varnameodds[colname='Effect'
label=""]           varnameoddsy[colname="&y"
label=""]  odds ratio[label=""  colname='Point
Estimate' format=12.3]
         odds_ratioclm[label=""
colname={"Wald  %sysevalf(100*(1-&alpha))%  CL
Lower"  "Wald  %sysevalf(100*(1-&alpha))%  CL
Upper"} format=12.3];
        %end;
quit;
%end;
%mend surveyglm;


/************************************
******************
*       The distributions available in
%surveyglm macro are:
*       NORMAL, GAMMA, IG (INVERSE
GAUSSIAN), POISSON,
*    BINOMIAL (AND MULTINOMIAL).
************************************
******************
*
*       The link functions available in
%surveyglm macro are:
*       IDENTITY, INV (INVERSE), INV2
(INVERSE SQUARED), LOG
*       LOGIT (GENERALIZED LOGIT).
************************************
******************/
```