

A SAS MACRO TO GENERATE NORMAL AND HALF-NORMAL PLOTS WITH SIMULATED ENVELOPE

Paulo Henrique Dourado da SILVA¹

- **ABSTRACT:** *This work will provide a SAS macro to generate normal and half-normal plots using the simulated envelope technique proposed by Atkinson (1985). The macro name is %envelope. This macro is based on the GENMOD procedure, and the main motivation for your creating is that such a diagnostic tool is not available in software SAS. It will be presented two illustrations of the use of this macro, one for simulated database and one using a database available in GENMOD procedure help.*
- **KEYWORDS:** *Generalized linear model; diagnostics; normal plots; half-normal plots; SAS.*

1 Introduction

Since it's origin, the generalized linear models (GLM) (Nelder and Wedderburn 1972) have been widely used in various areas of knowledge such as medicine, agronomy, economics, quality control and other. The steps for setting a GLM data can be summarized in the following steps:

1. Specification of the random component of the model (Distribution of the response variable);
2. Specification of the systematic part of the model (Linear predictor);
3. Choice of the link function;
4. Estimation of model parameters;
5. Generation of Model diagnostics.

¹University of Brasilia - UnB, Department of Statistics, CEP 70910-900, Brasilia, DF, Brazil.
E-mail: paulodourado.unb@gmail.com

The model diagnostic is an essential part of the modeling process, this step allows the analysis of suitability of the specified model and thus make the inference more consistent, accurate and reliable. There are several methods of model diagnostic (see McCullagh and Nelder 1989, Atkinson 1985, Atkinson et. al. 1989, Dobson 2001), among them there are the graphical methods which allow, in a simple and visual manner, identify potential problems in steps 1, 2 or 3. Among the graphical methods there are normal and half-normal plots, which allow you to check whether the choice of the random component is appropriate and identify possible outliers in the data (Weisberg, 2005). In SAS the setting of a GLM is done through GENMOD procedure (SAS Institute, 2011), this procedure provides several statistical diagnostics and plots important for model diagnostic. But, in SAS, there is no command or procedure that creates normal or half-normal plots with simulated envelope, which are objects of study in this work. To fill this gap was created a macro named *%envelope* which allows to generate normal and half-normal plots with simulated envelope. This paper is divided as follows: Section 2 will be introduced briefly the generalized linear models and the estimation process. Section 3 will be discussed briefly the model diagnostic in generalized linear models and the methodology proposed by Atkinson (1985) for the generation of normal and half normal plots with simulated envelope yet. The macro is described in section 4, section 5 will show some illustrations of the use of macro *%envelope*. Finally in section 6 the final considerations will appear.

2 Generalized linear models

The generalized linear models (GLM) were introduced by Nelder and Wedderburn (1972). They unified many existing analysis methodologies 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 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 , which belongs to exponential family distribution;
- A systematic component represented by the linear predictor $\mathbf{X}'\boldsymbol{\beta}$;
- And a link function, monotone and differentiable, $\boldsymbol{\eta} = g(\boldsymbol{\mu})$ linking the random component to the systematic component in the model.

2.1 Estimation procedure

The primary method for parameter estimation in generalized linear model is the maximum likelihood method. To use this method, one needs to maximize the

log-likelihood function associated with the distribution of the response variable:

$$L(y, \mu, \phi) = \sum_i \log(f(y_i, \mu_i, \phi)). \quad (1)$$

The ridge-stabilized Newton-Raphson algorithm is commonly used in parameter estimation, which is implemented in the SAS GENMOD procedure (SAS Institute, 2011). In the k -th iteration, the algorithm updates the parameter vector β_k as following:

$$\beta_{k+1} = \beta_k - \mathbf{H}^{-1} \mathbf{s}, \quad (2)$$

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

$$\mathbf{s} = [s_j] = \left[\frac{\partial L}{\partial \beta_j} \right], \quad (3)$$

and

$$\mathbf{H} = [h_{ij}] = \left[\frac{\partial^2 L}{\partial \beta_i \partial \beta_j} \right]. \quad (4)$$

For models that have some scale parameter (and/or dispersion)¹, this parameter is assumed known and is estimated by maximum likelihood or moments method². To estimate the vector s and the matrix H , we can use the chain rule since $\mu_i = g^{-1}(\mathbf{x}'_i \beta)$, thus the vector \mathbf{s} and the matrix \mathbf{H} are given by

$$\mathbf{s} = \sum_i \frac{w_i (y_i - \mu_i)}{V(\mu_i) g'(\mu_i) \phi}, \quad (5)$$

and

$$\mathbf{H} = -\mathbf{X}' \mathbf{W}_o \mathbf{X}, \quad (6)$$

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

$$w_{oi} = w_{ei} + w_i (y_i - \mu_i) \frac{V(\mu_i) g''(\mu_i) + V'(\mu_i) g'(\mu_i)}{(V(\mu_i))^2 (g'(\mu_i))^3 \phi}, \quad (7)$$

where

$$w_{ei} = \frac{w_i}{\phi V(\mu_i) (g'(\mu_i))^2}. \quad (8)$$

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

¹Normal, gamma, negative binomial, inverse normal model

²More information about the process of parameter estimation in models with scale parameter and/or dispersion can be found in McCullagh and Nelder (1989)

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

3 Model diagnostic

Setting an GLM involves three main parts: i) definition of the distribution of the response variable (random component), ii) choice of the link function which connects the random component to the systematic component, and iii) the definition of the matrix model (specification the systematic part). In practice, even a careful choice of model may result in an unsatisfactory fit.

Poor model fitting may be caused by several factors, all of which could produce systematic deviations from the model, among these factors it can be mentioned: Inappropriate choice of the link function, omission of important explanatory variables, inappropriate choice of the distribution of the response variable, outliers and influential points among other factors.

The previous statements make clear the need and importance of the diagnostic stage of model construction, to verify whether the model is a good fit to the data. In literature there are several techniques to make the diagnostics for generalized linear models such as residual analysis and techniques to identify outliers and influential values (Cook's distance, leverage, DFFITS, DFBETAS), for more information about these techniques to under viewpoint of generalized linear models can be consulted McCullagh and Nelder (1989), Atkinson (1985) and Dobson (2001).

Among these diagnostic techniques are the graphical methods. Such methods are of fundamental importance because they can identify, visually and simply, factors causing systematic deviations from the model and how these factors influence the different components that are part of the GLM .

Recently, the normal and half-normal plots with simulated envelope gained importance as diagnostics tools. They are because they are powerful and simple tools for diagnostics, especially in models with complex distributions such as those studied by Vieira et. al. (2000). The normal and half-normal plots with simulated envelope are the main products of this work and will be detailed in the next section.

3.1 Normal and Half-normal probability plots with simulated envelope

The normal probability plot stands for two aspects (Weisberg 2005):

- identifying the distribution of the data originating and
- identifying outliers in the dataset.

The construction of this plot demands following the steps above:

1. Fit a model in the data set and get $d_{(i)}$, the values sorted in a certain statistical diagnostic (residual, leverage, Cook distance, and others);
2. from the order statistic in position (i) , calculate the corresponding cumulative probability p_i and their quantile. This probability p_i is generally approximated by

$$p_i = \frac{i - c}{n - 2c + 1},$$

where $0 < c < 1$. Several authors recommend the use of $c = 3/8$. For this value of c , the quantile z_i may be given by

$$z_i = \Phi^{-1} \left(\frac{i - 0.375}{n + 0.25} \right);$$

3. Plot $d_{(i)}$ versus z_i .

To construct the half-normal probability plot, plot the values $|d_{(i)}|$ versus z_i wherein $z_i = \Phi^{-1} \left(\frac{i+n-0.125}{2n+0.5} \right)$. To assist in interpreting half-normal probability plot, Atkinson (1985) proposed the addition of a simulated envelope. This plot is obtained through the following steps:

1. Fit a model in the data set and get $d_{(i)}$, the values sorted in a certain statistical diagnostic (residual, leverage, Cook distance, and others);
2. Simulate 19 samples of response variable, using the estimates obtained after a particular model to be fitted to the data and the same values for the explanatory variables;
3. Setting the same model each of the 19 samples and calculate the absolute values of the ordered statistics of diagnostic interest, $d_{j(i)}^*$, $j = 1, \dots, 19$ and $i = 1, \dots, n$;
4. For each i , calculate the mean (or median), minimum (or quantile $\alpha/2$) and the maximum (or quantile $1 - \alpha/2$) of the $d_{j(i)}^*$;
5. Plot the quantities calculated in the previous item and $d_{(i)}$ versus z_i .

The interpretation of the generated graph should be as follow: Once the the correct model is chosen, the statistics $d_{(i)}$ obtained from observations are inside the envelope (between the borders of the plotted statistics), note that this method can be applied also to normal probability plot. Hinde and Demétrio (1997) present several macros that allow you to make these plots for various models using the software GLIM.

4 Macro *%envelope*

In this section the parameters of the macro *%envelope* will be briefly described, full description of the macro is in the **Appendix B**.

The general call of the macro is given by

```
%macro envelope(data=, predict=, resid=, m=1, pzero=, plinear=,
zlinear=, class_v=,offset=,dispersion=, scale=, family=Normal,
sim=10, link=identity, zlink=logit, type=HN, quasi=F, off=F);
```

Where:

1. *data* - Enter the database analyzed;⁴
2. *predict* - Informs the predicted values of the final model;
3. *resid* - Informs the residuals obtained from the final model;
4. *m* - Number of cases for the binomial model;⁵
5. *pzero* - Informs the zeroes probability for the final model with an excess of zero;
6. *plinear* - Linear predictor;
7. *zlinear* - Linear predictor for an excess of zero;
8. *class_v* - Categorical variables of the model;
9. *offset* - Offset;
10. *dispersion* - Dispersion parameter;
11. *scale* - Scale parameter for the quasi-likelihood, gamma, normal, inverse normal, and others models;
12. *family* - Family of distributions of the response variable;
13. *sim* - Simulation Number;
14. *link* - Linking functions for mean model;
15. *zlink* - Linking functions for the zero model;
16. *type* - Plot type - HN=Half-Normal ou N=Normal;
17. *quasi* - Is it quasi-likelihood model? (Ex: Quasi-poisson model) T or F;
18. *off* - The model has offset? (T or F).

5 Results

To illustrate the macro *%envelope* described in the previous section, it was first used a simulated database. In this data set, 100 observations were generated from a gamma model with scale parameter $\phi = 3$, with inverse link function $g(\mu_i) = 1/x_i\beta$, and with $x_i \sim N(0, 1)$. The SAS commands for the generation of the database are given by:

⁴The database should contain the reported variables in the model and two columns, one informing the predicted values and other informing residuals.

⁵The parameter *m* should be used when specifying the response variable of the binomial model in command *MODEL* in the following form *model r/m = covariates / dist = binomial*;

```
data teste;
  do i=1 to 100; x=rannor(435); mu=1/(0.2*x+1); output; end;
run;
```

```
data teste; set teste; y=(mu/3)*rangam(231, 3);
run;
```

After, Normal model with identity link function was adjusted via GENMOD procedure

```
proc genmod data=teste; model y=x/ dist=normal;
  output out=teste1 p=predito stdresdev=residuo;
run;
```

Done that, the following commands were executed to generate the plots with simulated envelope

```
%envelope(data=teste1, predict=predito, resid=residuo, plinear=x,
  scale=0.6706, family=normal, link=identity, type=HN);
%envelope(data=teste1, predict=predito, resid=residuo, plinear=x,
  scale=0.6706, family=normal, link=identity, type=N);
```

Which generated the simulated envelopes illustrated in Figure 1.

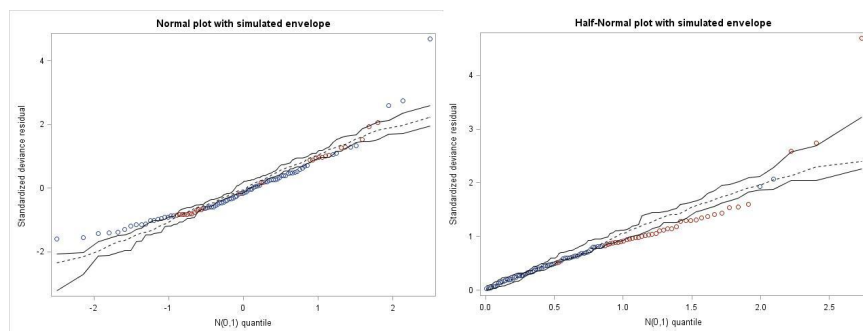


Figure 1 - Normal and Half-Normal plot for Normal model with identity link function

As might be expected, most of the points are outside the envelope, showing that the fitted model is not suitable for the data. The failure on adjusting was captured by the graphics.

A gamma model with inverse link function was also adjusted, the commands used were:

```
proc genmod data=teste; model y=x/ dist=gamma;
  output out=teste1 p=predito stdresdev=residuo;
run;
```

Done that, were executed the following commands:

```
%envelope(data=teste1, predict=predito, resid=residuo, plinear=x,
scale=2.9810, family=Gamma, link=identity, type=HN);
%envelope(data=teste1, predict=predito, resid=residuo, plinear=x,
scale=2.9810, family=Gamma, link=identity, type=N);
```

Which generated the simulated envelopes illustrated in Figure 2.

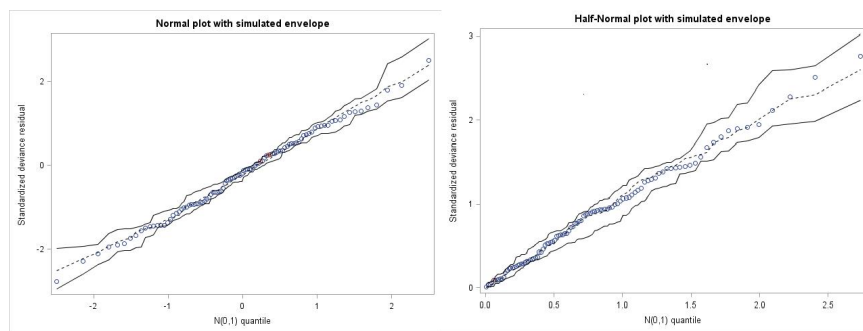


Figure 2 - Normal and Half-Normal plot for Gamma model with inverse link function

The next analyzed database is related to the application of gamma distribution in real data (Example 39.3 GENMOD procedure, SAS Institute 2011). The database is illustrated in **Appendix A**. The gamma model was adjusted using the following commands:

```
proc genmod data = lifdat; class mfg;
model lifetime = mfg / dist=gamma link=power(-1) type3;
output out = lifedat_out pred = predito stdresdev = residuo;
run;
```

After the adjustment, the following commands were executed to verify the suitability of the chosen distribution for the response variable, as well as the presence of outliers

```
%envelope(data=lifedat_out, predict=predito, resid=residuo, class_v=mfg,
plinear=mfg, scale=0.8275, family=Gamma, link=power(-1), type=HN);
%envelope(data=lifedat_out, predict=predito, resid=residuo, class_v=mfg,
plinear=mfg, scale=0.8275, family=Gamma, link=log, type=N);
```

Which generated the simulated envelopes illustrated in Figure 3.

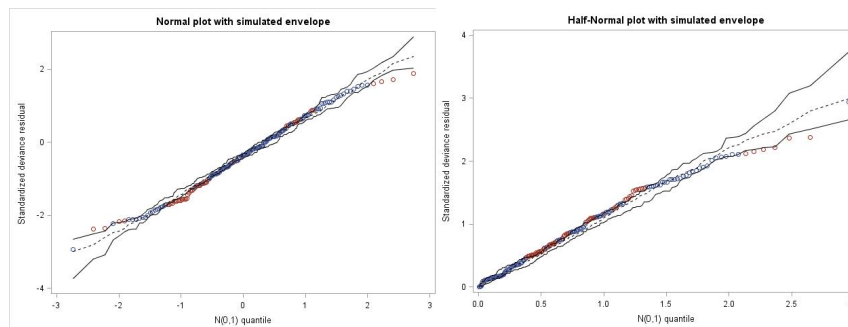


Figure 3 - Normal and Half-Normal plot for life data

Conclusions

The present work introduced a SAS macro named `%envelope`. The macro allows the analyst to generate normal and half-normal plots with simulated envelope, as proposed by Atkinson (1985). This macro fills a gap in existing diagnostic methods implemented in SAS, it was built simply using the support provided by the GENMOD procedure. Through this macro, analysts will have another tool, simple and efficient, to verify the suitability of the distribution chosen to model the response variable, as well as detect possible outliers and influential points in the data. As seen from the above results, in case of a misspecification of the distribution of a response variable in the interest model, this fact is immediately reported in normal or half-normal plot through the appearance of significant amount of points outside the envelope. Other diagnostic statistics can be easily added to the macro through the output options of the GENMOD procedure.

Acknowledgments

The author is grateful for the comments of the professor Alan Ricardo da Silva in order to improve the quality of the article.

SILVA, P. H. D. Uma macro SAS para gerar gráficos normal e meio-normal com envelope simulado. *Rev. Bras. Biom.*, São Paulo, v.32, n.3, p.460-473, 2014.

- RESUMO: Esse trabalho irá mostrar uma macro SAS para gerar gráficos normal e meio-normal usando a técnica de envelope simulado proposto por Atkinson (1985). O nome da macro é `%envelope`. Essa macro é baseada no procedimento GENMOD, e a principal motivação para sua criação é que tal ferramenta de diagnóstico não está disponível no *software* SAS. Serão apresentados duas ilustrações do uso dessa macro, uma para uma base de dados simulada e outra usando uma base de dados disponível na ajuda do procedimento GENMOD.

- PALAVRAS-CHAVE: Modelos lineares generalizados; diagnóstico; gráfico normal; gráfico meio-normal; SAS.

References

- ATKINSON, A. C. *Plots, Transformations, and Regression*. Oxford: Oxford University Press, 1985. 296p.
- DEMÉTRIO, C. G. B.; HINDE, J. Half normal plots and overdispersion. *GLIM Newsletter*, v.27, p.19-26, 1997.
- DOBSON, A. J. *An Introduction to Generalized Linear Models*. 3.ed. London: Chapman & Hall/CRC, 2008. 320p.
- McCULLACH, P.; NEDER, J. A. *Generalized linear models*. 2.ed. London: Chapman and Hall, 1989. 511p.
- NELDER, J. A., and WEDDERBURN, R. W. M., Generalized Linear Models, *Journal of the Royal Statistical Society Ser. A*, v.135, p.370-384, 1972.
- SAS INSTITUTE, INC., SAS/STAT 9.3 User's Guide, Cary, NC: SAS Institute, Inc., 2011.
- VIEIRA, A.; HINDE, J.; DEMÉTRIO, C., Zero-inflated proportion data models applied to a biological control assay. *Journal of Applied Statistics*, v.27, p.373-389, 2000.
- WEISBERG, S. *Applied linear regression*. 3.ed. New York: John Wiley, 2005.

Received in 30.05.2014.

Approved after revised in 14.08.2014.

APPENDIX A - Life data

```
data A;
  input lifetime@@ ;
  mfg = 'A';
  datalines;
620 470 260 89 388 242 103 100 39 460 284 1285
218 393 106 158 152 477 403 103 69 158 818 947
399 1274 32 12 134 660 548 381 203 871 193 531
317 85 1410 250 41 1101 32 421 32 343 376 1512
1792 47 95 76 515 72 1585 253 6 860 89 1055
537 101 385 176 11 565 164 16 1267 352 160 195
1279 356 751 500 803 560 151 24 689 1119 1733 2194
763 555 14 45 776 1
;
run;

data B;
  input lifetime@@ ;
  mfg = 'B';
  datalines;
1747 945 12 1453 14 150 20 41 35 69 195 89
1090 1868 294 96 618 44 142 892 1307 310 230 30
403 860 23 406 1054 1935 561 348 130 13 230 250
317 304 79 1793 536 12 9 256 201 733 510 660
122 27 273 1231 182 289 667 761 1096 43 44 87
405 998 1409 61 278 407 113 25 940 28 848 41
646 575 219 303 304 38 195 1061 174 377 388 10
246 323 198 234 39 30 55 729 813 1216 1618 539
6 1566 459 946 764 794 35 181 147 116 141 19
380 609 546
;
run;

data lifdat;
  set A B;
run;
```

APPENDIX B *%envelope* macro

```

%macro envelope(data=, predict=,
resid=, m=1, pzero=, plinear=,
zlinear=, class_v=, offset=,
dispersion=, scale=,
family=Normal, sim=10,
link=identity,
zlink=logit, type=HN, quasi=F,
off=F);
%if %upcase(&family)=BINOMIAL
%then %do;
/*Binomial*/
%do i=1 %to &sim;
/*simulating the response
variable*/
data &data; set &data;
resp=ranbin(&i., &m., &predict);
run;
/*Residual generating*/
ods select none;
%if %upcase(&quasi)=F %then %do;
ods select none;
proc genmod data=&data descending;
class &class_v;
%if &m=1 %then %do;
model resp= &plinear /
dist=binomial link=&link;
%end; %else %do;
model resp/&m.= &plinear /
dist=binomial link=&link; %end;
output out=&data
STDRESDEV=deviance&i;
run;%end;
%else %if %upcase(&quasi)=T %then
%do; ods select none;
proc genmod data=&data;
class &class_v; %if &m=1 %then %do;
model resp= &plinear/dist=binomial
link=&link noscale scale=&scale;
%end; %else %do;
model resp/&m.= &plinear /
dist=binomial
link=&link noscale scale=&scale;
%end;
output out=&data
STDRESDEV=deviance&i;
run; %end; %end;%end;
%if %upcase(&family)=GAMMA %then
%do;
/*Gamma*/
%do i=1 %to &sim;
/*simulating the response variable*/
data &data; set &data;
resp=(&predict./&scale.)*rangam(&i, &scale.);
run;
/*Residual generating*/
ods select none;
proc genmod data=&data;
class &class_v; model resp= &plinear / dist=gamma
link=&link; output out=&data STDRESDEV=deviance&i;
run; %end; %end;
%else %if %upcase(&family)=GEOM %then %do;
/*Geometric*/
%do i=1 %to &sim;
/*simulating the response variable*/
data &data; set &data;
gama=(&predict.*rangam(&i, 1)); resp=ranpoi(&i, gama);
run;
/*Residual generating*/
ods select none;
proc genmod data=&data;
class &class_v; model resp= &plinear / dist=geom
link=&link; output out=&data STDRESDEV=deviance&i;
run; %end; %end;
%else %if %upcase(&family)=NB %then %do;
/*Negative Binomial*/
%do i=1 %to &sim;
/*simulating the response variable*/
data &data; set &data;
theta=1/&dispersion.;
gama=(&predict.*rangam(&i, theta))/theta;
resp=ranpoi(&i, gama);
run;
/*Residual generating*/
%if %upcase(&quasi)=F %then %do;
ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear / dist=nb
link=&link;
output out=&data
STDRESDEV=deviance&i;
run;
%end;
%else %if %upcase(&quasi)=T
%then %do;
ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear / dist=nb
link=&link
noscale scale=&scale;
output out=&data
STDRESDEV=deviance&i;
run;%end;%end;%end;
%else %if %upcase(&family)=IG
%then; %do;
/*Inverse Gaussian*/
proc sql;
select COUNT(*) into :nobs
from &data;
quit;
%do i=1 %to &sim;
/*simulating the response
variable*/
data ig;
do i=1 to &nobs;
v=rannor(&i); z=ranuni(&i);
output;
end;run;
data &data; set &data; set ig;
y=v**2;
mu2=&predict.**2;
scale=1/(&scale.);
x=&predict.+(y*(mu2)/(2*scale))-
((&predict.)/(2*scale))*
sqrt(4*&predict.*scale*y+(mu2)*
(y**2));

```

```

if z <= &predict./(x+&predict.) then
rig=x; else rig=(mu2)/x;
run;
data &data; set &data; set ig;
run;
/*Residual generating*/
ods select none;
proc genmod data=&data;
class &class_v;
model rig= &plinear / dist=igaussian
link=&link;
output out=&data STDRESDEV=deviance&i;
run;

%end;
%else %if %upcase(&family)=NORMAL %then %do;
/*Normal*/
proc sql;
select COUNT(*) into :nobs from &data;
quit;
%do i=1 %to &sim;
/*simulating the response variable*/
data rnorm;
do j=1 to &nobs;
norm=rannorm(&i);
output;
end;
drop j;
run;

data &data; set &data; set rnorm;
resp=&predict.*&scale.*norm;
run;

/*Residual generating*/
ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear / dist=normal
link=&link;
output out=&data STDRESDEV=deviance&i;
run;%end;%end;
%else %if %upcase(&family)=POISSON
%then %do; /*Poisson*/
%do i=1 %to &sim;
/*simulating the response variable*/
data &data; set &data;
resp=ranpoi(&i, &predict);
run;

/*Residual generating*/
%if %upcase(&quasi)=F
%then %do;
%if %upcase(&off)=F
%then %do;
ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear /
dist=poissonlink=&link;
output out=&data
STDRESDEV=deviance&i;
run;
%end; %else %if %upcase(&off)=T
%then %do;
ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear / dist=poisson
link=&link offset=&offset;
output out=&data
STDRESDEV=deviance&i;
run;%end;%end;
%else %if %upcase(&quasi)=T
%then %do;
%if %upcase(&off)=F
%then %do;ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear / dist=poisson
link=&link noscale
scale=&scale;
output out=&data
STDRESDEV=deviance&i;
run;%end;
%else %if %upcase(&off)=T
%then %do;ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear / dist=poisson
link=&link
offset=&offset noscale
scale=&scale;
output out=&data
STDRESDEV=deviance&i;
run;%end;%end;%end;%end;
%else
%if %upcase(&family)=ZINB %then %do;
/*Zero Inflated Negative Binomial*/
%do i=1 %to &sim;
/*simulating the response variable*/
data &data; set &data;
theta=1/&scale.;
gama=(&predict.*rangam(&i, theta))/theta;
negbin=ranpoi(&i, gama);
ben=ranbin(&i, 1, &pzero);
if ben=1 then resp=0; else
resp=negbin;
run;

/*Residual generating*/
ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear / dist=zinb
link=&link;
zeromodel &zlinear / link=&zlink;
output out=&data reschi=deviance&i;
run;
%end;
%end;
%else %if %upcase(&family)=ZIP %then %do;
/*Zero Inflated Poisson*/;
%do i=1 %to &sim;
/*simulating the response variable*/
data &data; set &data;
pois=ranpoi(&i, &predict.);
ben=ranbin(&i, 1, &pzero);
if ben=1 then resp=0; else
resp=pois;
run;

```

```

/*Residual generating*/
ods select none;
proc genmod data=&data;
class &class_v;
model resp= &plinear / dist=zip
link=&link;
zeromodel &zlinear / link=&zlink;
output out=&data reschi=deviance&i;
run;

%end;
%end;
data dev; set &data;
keep deviance1 -- deviance&sim;
run;

proc iml;
use dev;
read all into e;
close;

use &data;
read all var{ &resid. } into r;
close;

n=nrow(e);

%if &type=HN %then %do;

r=abs(r);
do i=1 to &sim;
m=abs(e[,i]);
call sort(m);
e[,i]=m;
end;
p=(n+(1:n)-0.125)/(2*n+0.5);

%end; %else %if &type=N %then %do;

do i=1 to &sim;
m=e[,i];
call sort(m);
e[,i]=m;
end;

p=((1:n)-0.375)/(n+0.25);
%end;

q=quantile("Normal", p);
te=t(e);
k=j(n,3,0);
do i=1 to n;
h=te[,i];
k[i,1]=min(h);
k[i,2]=median(h);
k[i,3]=max(h);
end;
call sort(r);
base=r||t(q)||k;
*print base;
create envelope from base;
append from base;
quit;

data envelope1; set envelope;

if col1 < col3 or col1 > col5 then color=1;
else color=2;
run;

%if &type=HN %then %do;

ods select all;
proc sgplot data=envelope1 noautolegend;
title "Half-Normal plot with simulated envelope";
scatter y=COL1 x=COL2 / group=color;
series y=COL3 x=COL2 / lineattrs=(color=black);
series y=COL4 x=COL2 /
lineattrs=(color=black pattern=2);
series y=COL5 x=COL2 / lineattrs=(color=black);
xaxis label="N(0,1) quantile";
yaxis label="Standardized deviance residual";
run;
quit;

%end; %else %if &type=N %then %do;

ods select all;
proc sgplot data=envelope1 noautolegend;
title "Normal plot with simulated envelope";
scatter y=COL1 x=COL2 / group=color;
series y=COL3 x=COL2 / lineattrs=(color=black);
series y=COL4 x=COL2 /
lineattrs=(color=black pattern=2);
series y=COL5 x=COL2 / lineattrs=(color=black);
xaxis label="N(0,1) quantile";
yaxis label="Standardized deviance residual ";
run;quit;%end;%mend envelope;

```