sg*yyy*	rglm - Robust variance estimates for generalized linear models

Roger Newson
Imperial College School of Medicine, London, UK
r.newson@ic.ac.uk

Syntax

`rglm` [*varlist*] [*weight*] [`if` *exp*] [`in` *range*] [`,` `cluster(`*varname*`)` `mspec` `tdist` `minus(`#`)` *glm-options*]

where *glm-options* are any of the options available for `glm`; see [R] **glm**.

`fweights`, `iweights` and `aweights` are allowed; see [U] **weight**.

Description

`rglm` fits generalized linear models and calculates a Huber (sandwich) estimate of the variance-covariance matrix of estimates. It can be used alone or called without arguments after a previous call to `glm`. As with other "robust" commands, the units may be considered to fall into clusters.

Options

`cluster(`*varname*`)` specifies the variable which defines sampling clusters.

`mspec` specifies that full Huber variances be used. These are robust to mis-specification of conditional means. If `mspec` is absent, semi-Huber variances are calculated, robust to variance mis-specification caused by overdispersion, underdispersion, heteroscedasticity and clustering, but assuming that conditional means are specified correctly by the model. (Except in the case of canonical link functions, where the semi-Huber variance is the full Huber variance. See Section 2.5 of McCullagh and Nelder (1989).)

`tdist` specifies that P-values and confidence intervals are to be calculated assuming estimates to have a t-distribution with $M - p$ degrees of freedom, where p is the number of model parameters, and M is the number of clusters if `cluster` is specified, or the number of observations (or sum of frequency weights) if `cluster` is not specified.

`minus(`# `)` specifies the `minus` parameter to pass to `_robust`, to apply a finite-sample adjustment to the Huber covariance matrix. If absent (or negative), it is reset to p (the number of model parameters).

If a *varlist* is supplied, then all `glm` options are allowed. If not, then the only `glm` options allowed are `level` and `eform`, and `cluster`, `mspec`, `tdist` and `minus` are ignored.

Methods and Formulas

In a generalized linear model (GLM), we attempt to predict a $n \times 1$ outcome variate Y using a $n \times p$ matrix X of predictor variates, where n is the number of observations and p is the number of parameters. These parameters form a $p \times 1$ vector β. The $n \times 1$ vector $\eta = X\beta$, known as the linear predictor, is used to predict Y, which is assumed in the model to have a conditional expectation equal to the $n \times 1$ vector μ. The vectors η and μ are assumed in the model to have a relation of the form $\eta_i = g(\mu_i)$, where $g(\cdot)$ is a monotonically increasing function, referred to as the link function. The conditional variance of y_i, given X, is assumed in the model to be proportional to a variance function $V(\mu_i)$. The choice of link function $g(\cdot)$ and variance function $V(\cdot)$ distinguishes one GLM from another. We will assume frequency weights (`fweights`) f_i and non-frequency weights (`iweights`) w_i, both defaulting to ones if not specified. (In fact, only one kind of weight may be specified, but that is a very minor defect of Stata, not a mathematical requirement.)

The fitting of a GLM involves finding values of β which give a zero value simultaneously for the p sums of scores $\sum_{i=1}^{n} f_i \psi_{ij}$, for j from 1 to p. The $n \times p$ matrix Ψ is defined such that $\psi_{ij} = w_i x_{ij} S_i$, where the S_i are in turn defined by

$$S_i = S(y_i, \eta_i) = \frac{d\mu_i}{d\eta_i} \left[V(\mu_i)\right]^{-1} (\mu_i - y_i). \tag{1}$$

In the model, S_i is interpreted as the derivative, with respect to η_i, of the i'th squared deviance residual, which is proportional to the conditional log likelihood of y_i given the row matrix X_i. ψ_{ij} is the corresponding derivative with respect to β_j. (See [R] **glm** or McCullagh and Nelder, 1989.)

The derivative of the j'th sum of scores with respect to the k'th parameter β_k is equal to $\sum_{i=1}^{n} f_i w_i x_{ij} x_{ik} H_i$, where H_i is the i'th Hessian function

$$H_i = \frac{dS_i}{d\eta_i}$$

$$\begin{aligned}
&= [V(\mu_i)]^{-1}\left(\frac{d\mu_i}{d\eta_i}\right)^2 + (\mu_i - y_i)\frac{d}{d\eta_i}\left\{[V(\mu_i)]^{-1}\frac{d\mu_i}{d\eta_i}\right\}\\
&= [V(\mu_i)]^{-1}\left(\frac{d\mu_i}{d\eta_i}\right)^2 + [V(\mu_i)]^{-1}\frac{d^2\mu_i}{d\eta_i^2}(\mu_i - y_i) - [V(\mu_i)]^{-2}\frac{dV(\mu_i)}{d\mu_i}\left(\frac{d\mu_i}{d\eta_i}\right)^2(\mu_i - y_i)\\
&= [V(\mu_i)]^{-1}\left[\left(\frac{d\mu_i}{d\eta_i}\right)^2 + \frac{d^2\mu_i}{d\eta_i^2}(\mu_i - y_i) - \frac{dV(\mu_i)}{d\mu_i}\frac{d\mu_i}{d\eta_i}S_i\right]. \qquad (2)
\end{aligned}$$

To estimate the dispersion matrix of the parameters β_j, we proceed as follows, using the principles of Huber (1967). Define the $p \times p$ matrix $D = \sum_{i=1}^n f_i w_i H_i X_i^T X_i$. The variance expression depends on whether or not clusters are specified. If there are no clusters, then the estimated dispersion matrix is

$$\frac{\sum_{i=1}^n f_i}{\sum_{i=1}^n f_i - k_{\text{minus}}}\left(\Psi D^{-1}\right)^T F\left(\Psi D^{-1}\right), \qquad (3)$$

where k_{minus} is the value given by the `minus` option, and F is the $n \times n$ diagonal matrix of the frequency weights f_i. If there are clusters, we denote by M the number of these clusters and define Ψ^* as the $M \times p$ matrix with one row per cluster, equal to the sum of the rows of Ψ corresponding to observations in that cluster, and estimate the dispersion matrix as

$$\frac{n}{n - k_{\text{minus}}}\left(\Psi^* D^{-1}\right)^T\left(\Psi^* D^{-1}\right). \qquad (4)$$

(It does not make sense to have both clusters and `fweights`, because $f_i > 1$ implies that the i'th observation represents multiple clusters.)

To calculate the Hessian in the general case by (2), we must know the variance function with its first derivative, and the inverse link fuction with its first two derivatives. The available variance functions have names corresponding to distributional families, whose variances are proportional to the respective functions, and their formulae and derivatives are as follows:

Family name	$V(\mu)$	$dV(\mu)/d\mu$
Gaussian (normal)	1	0
Gamma	μ^2	2μ
Inverse Gaussian	μ^3	$3\mu^2$
Bernoulli	$\mu(1-\mu)$	$1-2\mu$
Poisson	μ	1
Negative Binomial (shape parameter$=k$)	$\mu + k\mu^2$	$1 + 2k\mu$

The case of fitting a binomial model with totals m_i to the y_i is handled by `rglm` as equivalent to fitting a Bernoulli model to the proportions y_i/m_i and multiplying the `iweights` by the m_i. (That is to say, we substitute y_i/m_i for y_i, and $w_i m_i$ for w_i, in the formulae above.) In the case of the negative binomial distribution, the shape parameter k is defined according to the conventions of the Stata manuals and the innards of `glm.ado`, in which k is the reciprocal of the parameter of the same name defined in McCullagh and Nelder, 1989. I do not know how this confusing state of affairs came about.

The available forms for a link function $\eta = g(\mu)$ also have names. The following table gives their formulae and inverses, with their first and second derivatives. The derivatives are expressed in a computationally convenient form. In the case of the probit link, $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function, and $\phi(\cdot)$ is its derivative, the standard Gaussian probability density function.

Link function	$g(\mu)$	$g^{-1}(\eta)$	$d\mu/d\eta$	$d^2\mu/d\eta^2$
Identity	μ	η	1	0
Log	$\ln\mu$	e^η	μ	μ
Logit	$\ln[\mu/(1-\mu)]$	$e^\eta/(1+e^\eta)$	$\mu(1-\mu)$	$\mu(1-\mu)(1-2\mu)$
Probit	$\Phi^{-1}(\mu)$	$\Phi(\eta)$	$\phi(\eta)$	$-\eta\phi(\eta)$
Complementary log-log	$\log[-\log(1-\mu)]$	$1-e^{-e^\eta}$	$(\mu-1)\log(1-\mu)$	$[1+\log(1-\mu)]d\mu/d\eta$
Odds power q	$[\mu/(1-\mu)]^q$	$\eta^{1/q}/(1+\eta^{1/q})$	$1/q\mu^{1-q}(1-\mu)^{1+q}$	$\mu^{-q}(1-\mu)^q(1-2\mu-q)d\mu/d\eta$
Power q	μ^q	$\eta^{1/q}$	$q^{-1}\mu^{1-q}$	$(1-q)q^{-1}\mu^{-q}d\mu/d\eta$
Negative binomial (shape parameter$=k$)	$\ln[k\mu/(k\mu+1)]$	$k^{-1}e^\eta/(1-e^\eta)$	$\mu+k\mu^2$	$(1+2k\mu)d\mu/d\eta$

The negative binomial link function defined here is the correct version, consistent with the notation of the Stata manuals and with `glm.ado`. (The definition in [R] **glm** is a misprint.)

The calculation of the Hessian is greatly simplified if we can ignore the second term of the second line of (2), in which case we have

$$H_i = [V(\mu_i)]^{-1} \left(\frac{d\mu_i}{d\eta_i} \right)^2 , \tag{5}$$

and we need only know the variance function and the first derivative of the inverse link. This equality holds, in the expectation, if the model is indeed a correct specification of the conditional mean of Y given X, so that $E(\mu_i - y_i) = 0$ for each individual i. It also holds if the link function is the canonical link for the variance function. In this case, the variance function is proportional to the first derivative of the inverse link, and their ratio is a constant function of η, so the second term of the second line in (2) is zero. (See Section 2.5 of McCullagh and Nelder, 1989.) The variances calculated using the formula (5) are known as semi-Huber variances, whereas the variances calculated using formula (2) are known as full Huber variances. The semi-Huber variances (given by default) are robust to heteroscedasticity, overdispersion, underdispersion and clustering. The full Huber variances (obtained by the `mspec` option) are robust to all of these, and also to mis-specification of the conditional expectation. So, for instance, if we are fitting the parameters of a straight line, using a link function non-canonical for the chosen variance function, and the true relationship is slightly curved, then the parameters are estimates of the straight line giving the best fit to that curve, and the full Huber variances are consistent estimators of the true variance, in the population from which the rows of X and Y are jointly sampled. (I have not had time to do much research on how important the difference between semi-Huber and full Huber variances is in practice.)

Example 1

I often use `rglm` for carrying out unequal-variance t-tests on logs, using the `eform` option to get confidence intervals (CIs) for the two group geometric means and their ratio. For instance, in the case of the auto data, we might decide (after looking at stem-and-leaf plots) that `mpg` (miles per gallon) was distributed lognormally rather than normally. The calculation of the geometric means and their ratio is carried out by Stata as follows:

```
. * Geometric averages and their ratio *;
. gene logmpg=log(mpg);
. gene byte us=!foreign;
. * Stem and leaf plots *;
. stem mpg;
Stem-and-leaf plot for mpg (Mileage (mpg))
  1t | 22
  1f | 44444455
  1s | 66667777
  1. | 88888888899999999
  2* | 00011111
  2t | 22222333
  2f | 444455555
  2s | 666
  2. | 8889
  3* | 001
  3t |
  3f | 455
  3s |
  3. |
  4* | 1
. stem logmpg;
Stem-and-leaf plot for logmpg
logmpg rounded to nearest multiple of .01
plot in units of .01
  24* | 88
  25* |
  26* | 444444
  27* | 117777
  28* | 3333999999999
  29* | 44444444
  30* | 0004444499999
  31* | 4448888
  32* | 22222666
  33* | 3337
  34* | 003
  35* | 366
  36* |
  37* | 1
. * Geometric averages *;
. rglm logmpg foreign us,tdist eform noconst;
GLM with semi-Huber standard errors
Gaussian (normal) distribution, identity link
Number of observations: 74
------------------------------------------------------------------------------
         |              Semi-Huber
  logmpg |     e^coef   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 foreign |   23.96499   1.333711     57.079   0.000       21.44846    26.77678
      us |   19.30189   .6250385     91.414   0.000       18.09527    20.58898
------------------------------------------------------------------------------
. * Ratio between geometric averages *;
. rglm logmpg foreign,tdist eform;
GLM with semi-Huber standard errors
Gaussian (normal) distribution, identity link
Number of observations: 74
------------------------------------------------------------------------------
         |              Semi-Huber
  logmpg |     e^coef   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 foreign |   1.241587   .0799433      3.361   0.001       1.092027    1.411631
------------------------------------------------------------------------------
```

We find that foreign cars travelled at a geometric average of 23.96 mpg, whereas US cars travelled at a geometric average of 19.30 mpg. The foreign/US ratio was 1.24 (95% CI, 1.09 to 1.41), so foreign cars, on average, were 9% to 41% more efficient than US cars.

Example 2

We might also do a probit analysis to find a way of guessing whether a car is foreign, based on knowledge of its fuel efficiency and weight. The probit link is non-canonical for the Bernoulli variance function, so the full Huber variance will in general be different from the semi-Huber variance. Here, the analysis is carried out in three ways: using `glm`, using `rglm` with semi-Huber variances, and using `rglm` with full Huber variances. The results are as follows:

```
. * Non-robust using glm *;
. glm foreign mpg weight,family(bernoulli) link(probit);
Iteration 1 : deviance =   58.5137
Iteration 2 : deviance =   54.3546
Iteration 3 : deviance =   53.7194
Iteration 4 : deviance =   53.6887
Iteration 5 : deviance =   53.6884
Iteration 6 : deviance =   53.6884
Iteration 7 : deviance =   53.6884
Residual df  =        71                                No. of obs =        74
Pearson X2   =  51.28325                                Deviance   =  53.68838
Dispersion   =  .7222992                                Dispersion =  .7561743
Bernoulli distribution, probit link
------------------------------------------------------------------------------
 foreign |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     mpg |  -.1039505    .054209     -1.918   0.055      -.2101981    .0022972
  weight |  -.0023355    .000557     -4.193   0.000      -.0034273   -.0012438
   _cons |   8.275465   2.578791      3.209   0.001       3.221128     13.3298
------------------------------------------------------------------------------
. * Robust using rglm *;
. rglm foreign mpg weight,family(bernoulli) link(probit);
GLM with semi-Huber standard errors
Bernoulli distribution, probit link
Number of observations: 74
------------------------------------------------------------------------------
         |              Semi-Huber
 foreign |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     mpg |  -.1039505   .0690653     -1.505   0.132       -.239316     .031415
  weight |  -.0023355    .000497     -4.699   0.000      -.0033097   -.0013614
   _cons |   8.275465   2.751861      3.007   0.003       2.881916    13.66901
------------------------------------------------------------------------------
. * Robust using rglm with mis-specification correction *;
. rglm foreign mpg weight,family(bernoulli) link(probit) mspec;
GLM with full Huber standard errors
Bernoulli distribution, probit link
Number of observations: 74
------------------------------------------------------------------------------
         |               Huber
 foreign |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     mpg |  -.1039505    .060185     -1.727   0.084       -.2219109     .01401
  weight |  -.0023355   .0005003     -4.669   0.000        -.003316  -.0013551
   _cons |   8.275465   2.574692      3.214   0.001        3.229161   13.32177
------------------------------------------------------------------------------
```

Note that the parameter estimates are the same with all three methods, but the confidence limits are slightly different. All three methods find that the data are (just) compatible with the hypothesis that the coefficient of `mpg` is zero. (That is to say, the hypothesis that, once you know the weight of a car, you can hazard a guess as to whether or not it is American, and be as likely to be right as you would have been if you also knew its fuel efficiency.)

Example 3

This example is based on the housing data. Here, the data points are states of the USA, and we want to predict median rent from `pcturban` (percent urban) and `hsngval` (median housing value). This example compares the output from `rglm, tdist` with those from `regress` and `regress, robust`. Note that the two robust methods produce the same result (as they should), but the non-robust method gives the same estimates and very different CIs.

```
. * Regression analysis *;
. * Non-robust *;
. regr rent hsngval pcturban;
  Source |       SS       df       MS                  Number of obs =      50
---------+------------------------------               F(  2,    47) =   47.54
   Model |  40983.5269     2  20491.7635               Prob > F      =  0.0000
Residual |  20259.5931    47  431.055172               R-squared     =  0.6692
---------+------------------------------               Adj R-squared =  0.6551
   Total |    61243.12    49  1249.85959               Root MSE      =  20.762
------------------------------------------------------------------------------
    rent |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 hsngval |   .0015205   .0002276      6.681   0.000        .0010627    .0019784
pcturban |   .5248216   .2490782      2.107   0.040        .0237408    1.025902
   _cons |   125.9033   14.18537      8.876   0.000        97.36603    154.4406
------------------------------------------------------------------------------
. * Robust using regress *;
. regr rent hsngval pcturban,robust;
Regression with robust standard errors                 Number of obs =      50
                                                       F(  2,    47) =   34.47
                                                       Prob > F      =  0.0000
                                                       R-squared     =  0.6692
                                                       Root MSE      =  20.762
------------------------------------------------------------------------------
         |              Robust
    rent |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 hsngval |   .0015205   .0004654      3.267   0.002        .0005842    .0024568
pcturban |   .5248216    .309813      1.694   0.097       -.0984417    1.148085
   _cons |   125.9033   12.60741      9.986   0.000        100.5405    151.2662
------------------------------------------------------------------------------
. * Robust using rglm *;
. rglm rent hsngval pcturban,tdist mspec;
GLM with full Huber standard errors
Gaussian (normal) distribution, identity link
Number of observations: 50
------------------------------------------------------------------------------
         |               Huber
    rent |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 hsngval |   .0015205   .0004654      3.267   0.002        .0005842    .0024568
pcturban |   .5248216    .309813      1.694   0.097       -.0984417    1.148085
   _cons |   125.9033   12.60741      9.986   0.000        100.5405    151.2662
------------------------------------------------------------------------------
```

Validation

A program as comprehensive as `rglm` requires more validation than three examples. Accordingly, an intensive validation was carried out, using the auto data. `rglm` was tested using all six available variance functions, with one y-variate for each (`rep78` for the three discrete families, `mpg` for the three continuous families). Each family was tested with one canonical and one non-canonical link, except the binomial family, which was tested with its canonical logit link and all the non-canonical links for which the Binomial family is obligatory. (So every family and link was tested, and every family was tested with a canonical and a non-canonical link.) For each combination of family and link, three models were fitted. These had parameters as follows:

Model 1. One parameter, corresponding to the grand mean.

Model 2. Two groups (US and foreign cars), with parameters corresponding to two group means.

Model 3. Two parameters (an intercept and the slope of a quantitative covariate).

The quantitative covariate in Model 3 was always `gratio` for identity links and `weight` for non-identity links. (This was done because, when `mpg` and `rep78` were plotted against `weight` and `gratio`, the relationships involving `gratio` looked more linear.) The models fitted for each distributional family are summarized below.

Family	Y-variate	Canonical link	Covariate for canonical link	Non-canonical link(s)	Covariate for non-canonical link(s)
`gaussian`	`mpg`	`identity`	`gratio`	`log`	`weight`
`gamma`	`mpg`	`power -1`	`weight`	`identity`	`gratio`
`igaussian`	`mpg`	`power -2`	`weight`	`identity`	`gratio`
`binomial 6`	`rep78`	`logit`	`weight`	`probit,cloglog,opower 2`	`weight`
`poisson`	`rep78`	`log`	`weight`	`identity`	`gratio`
`nbinomial`	`rep78`	`nbinomial`	`weight`	`identity`	`gratio`

For each of the 42 models fitted, the dispersion was estimated in five different ways. These were the orthodox (Nelder) method given as default by `glm`, semi-Huber and full Huber variances without clustering, and semi-Huber and full Huber variances with clustering by `manuf`. Each parameter of each model therefore had five alternative standard errors (SEs).

In theory, some of these distinct SEs were expected to be equal. In the case of Model 1, there was no possibility for heteroskedasticity, overdispersion, underdispersion or mis-specification (as there is a single constant X-variate of ones), so all three unclustered SEs were expected to be equal, and both the clustered SEs were expected to be equal. In Model 2, there was a possibility of heteroskedasticity (because of unequal group variances), and sometimes overdispersion and underdispersion, but no possibility of mis-specification (because the predicted value of each individual is its group mean). The semi-Huber SE was therefore expected to be equal to the corresponding full Huber SE in each clustering class, although the orthodox, unclustered Huber and clustered Huber SEs were expected to be different. In Model 3, there was a possibility of heteroskedasticity, overdispersion, underdispersion and mis-specification, so all five SEs were expected to be different. Therefore, if `rglm` is working correctly, then we expect the SEs of Model 1 parameters to fall into two pre-defined equivalence groups (clustered and unclustered), the SEs of Model 2 parameters to fall into three pre-defined equivalence groups (orthodox, clustered Huber and unclustered Huber), and the SEs of Model 3 parameters to fall into five pre-defined equivalence groups of one each. SEs in the same equivalence group should be equal (or different only to the extent compatible with floating point calculation error), whereas SEs for the same model in different equivalence groups should be different.

As it happened, no two SEs in the same equivalence class were different by a ratio of more than 1.0001 (that is to say, the largest SE in an equivalence class was never more than 0.01% greater than the smallest SE in the same equivalence class). There was a lot more variation between equivalence classes for the same parameter of the same model. No two SEs in different equivalence classes for the same parameter of the same model differed by a ratio of less than 1.0020. That is to say, for any two SEs in different equivalence classes for the same parameter of the same model, the larger was always greater than the smaller by more than 0.2%, and usually the variation was much greater.

Figure 1 shows standard errors plotted on a binary log scale for all parameters of all models fitted. In the left-hand plot, the data points are SE equivalence classes (more than one for each parameter of each model), and the largest SE in the equivalence class is plotted against the smallest SE in the equivalence class. Note that all points are on the line of equality. In the right-hand plot, the data points are model parameters (one for each parameter of each model), and the largest SE for the parameter is plotted against the smallest SE calculated for that parameter. Note that the data points are visibly above the line of equality, although usually not so far above it as to indicate that the different SE calculation formulae give results in different binary orders of magnitude. So, unlike Example 3, these *ad hoc* examples do not truly demonstrate the advantages of Huber variances, although they do demonstrate that the SEs calculated by `rglm` using different methods are equal when they are supposed to be.

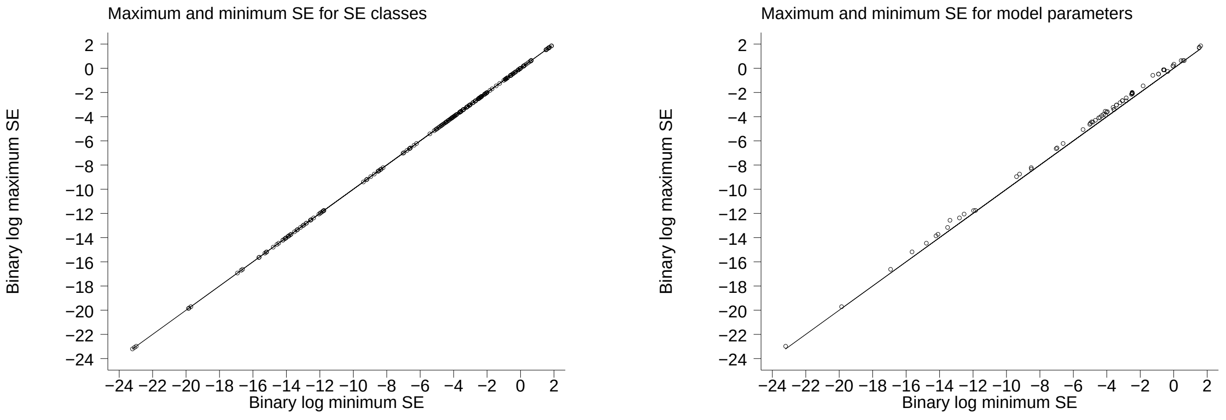

Figure 1. The results of the validation study for `rglm`.

Acknowledgment

This program was based on a previous version called `hglm`, which calculated only semi-Huber variances, and was kindly supplied to the author by David Clayton of MRC in Cambridge, England. The present author cleaned out some bugs, and added the options `mspec`, `tdist` and `minus`.

Also see

Manual: [R] **glm**, [R] **_robust**

References

Huber, P. J. 1967. The behaviour of maximum likelihood estimates under non-standard conditions. In *Proceedings of the Fifth Berkeley Symposium in Mathematical Statistics and Probability*. Berkeley, CA: University of California Press, 221-233.

McCullagh, P. and J. A. Nelder. 1989. *Generalized Linear Models. 2nd edition.* London: Chapman and Hall.