The Stata Journal (2002)
2, Number 4, pp. 378–390

Programmable GLM: Two user-defined links

Weihua Guan
Stata Corporation

Roberto G. Gutierrez
Stata Corporation

Abstract. With the release of Stata 7, the `glm` command for fitting generalized linear models underwent a substantial overhaul. Stata 7 `glm` contains an expanded array of variance estimators, regression diagnostics, and other enhancements. The overhaul took place to coincide with the release of Hardin and Hilbe (2001). With the new `glm` came a modular design that enables users to program customized link functions, variance functions, and weight functions to be used if Newey–West covariance estimates are desired. Because cases requiring customized link functions are the more prevalent in the literature, only those are considered here. We give two examples where a nonstandard link function is required: the relative survival model of Hakulinen and Tenkanen (1987) and a logistic model that accounts for natural response as described in Collett (2003). The relative ease (over previous versions of Stata) with which these alternate links can be programmed into `glm` is demonstrated.

Keywords: st0027, GLM, survival analysis, Cox regression, programming

1 Introduction

The theory of Generalized Linear Models (GLM) can be traced back to Nelder and Wedderburn (1972), who noted that generalizing some of the assumptions of linear regression resulted in a rich class of models. This class contained many existing models, such as logistic regression, which were already being used by researchers.

The basic premise of GLM is best understood by the following chronology, summarized here and exquisitely detailed in Hardin and Hilbe (2001). Many regression models, such as standard linear, logistic, probit, and Poisson, can be expressed in canonical exponential form. By expressing these models in canonical exponential form, it is noted that the relevant portion of the log likelihood differs over these models only by the specification of a *link function* and a *variance function*. The link function is so named because it expresses the conditional mean of the response as a function of the linear predictor, and thus "links" the response to the linear predictor, the standard ingredient in a linear regression model. The variance function expresses the variance of the response as a function of the mean.

Estimates of regression parameters may be obtained by maximizing the log likelihood. Two algorithms by which this may be achieved are Newton–Raphson and the method of Fisher scoring. Fisher scoring differs from Newton–Raphson in that the *expectation* of the matrix of second derivatives (the expected Hessian) is utilized rather than the observed Hessian. Regardless of the algorithm used, the calculations take a general form obtained from the exponential form of the likelihood, the link and variance functions, and the derivatives thereof. Using Fisher scoring as opposed to Newton–

Raphson simplifies the required calculations in two ways. First, Fisher scoring requires fewer analytical derivatives of the link and variance functions. Second, the method of Fisher scoring amounts to a series of repeated weighted linear regressions, known as *iterated reweighted least squares* (IRLS).

Furthermore, the distributional assumption of the response may be dropped altogether, in which case, the GLM model consists of only a link and variance function specification. In this case, the likelihood based on an exponential family is still utilized, but is referred to as a *quasi-likelihood*. Maximum quasi-likelihood estimators possess many of the same large sample properties as their distribution-based counterparts.

With Stata 7, the `glm` command for fitting generalized linear models underwent a substantial overhaul. Among the additions to the command were additional variance estimators based on the observed Hessian, the jackknife, the bootstrap, and additional diagnostic measures such as Anscombe residuals and Cook's distances. These enhancements were designed to make the most of the "many models under one roof" philosophy of `glm`. With these changes also came a more modular design, which allows the user to program his own link functions, variance functions, and weight functions for use with the Newey–West estimator of variance; see Hardin and Hilbe (2001) for a full treatment.

By far, the most useful of these programmer's features is the ability to program one's own link functions. There exist several examples in the literature of models that consist of a standard GLM variance function specification, such as the binomial or Poisson, coupled with a nonstandard link function. In this paper, we demonstrate two of these models: the Hakulinen and Tenkanen (1987) relative survival model, and a logistic model that accounts for natural response (Collett 2003).

Section 2 of this paper gives some details of the calculations involved in GLM, which serves to motivate the ingredients one needs to program a link function. Section 3 demonstrates the link for the relative survival model, Section 4 the link for the logistic model with natural response, and Section 5 some concluding remarks.

2 Calculations

Assume that the distribution of response, y, given a linear predictor, $\mathbf{x}\boldsymbol{\beta}$, is a member of the canonical exponential family. For a random sample of n observations $(y_i, \mathbf{x}_i)$, $i = 1, \ldots, n$, define $\eta_i = \mathbf{x}_i\boldsymbol{\beta}$. Also define

$$\mu_i \equiv E(y_i) = g^{-1}(\eta_i); \qquad \mathrm{Var}(y_i) = V(\mu_i)a(\phi)$$

for some $g()$, known as the *link function*, $V()$, known as the *variance function*, and some scale parameter $a(\phi)$. The fact that the mean and variance of y_i can be expressed in this manner follows by properties of the canonical exponential family; see Hardin and Hilbe (2001) for details.

Additionally, the maximum likelihood estimate of $\boldsymbol{\beta}$ may be obtained as the solution to the system of equations

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{a(\phi)V(\mu_i)} \left(\frac{\partial g^{-1}(\eta)}{\partial \eta} \right) \Bigg|_{\eta=\eta_i} \mathbf{x}_i^t = 0 \tag{1}$$

where L is the log likelihood. For example, in the case of OLS regression, $g()$ is the identity function, $V(\mu) = 1$, and $a(\phi)$ is the error variance. In the case of canonical Poisson regression, $g(\mu) = \ln(\mu)$, $V(\mu) = \mu$, and $a(\phi) = 1$.

Part of the appeal behind GLM is that the exponential family assumption may be dropped altogether, in favor of merely specifying $g()$ and $V()$. In this case, L is not a true likelihood, but a quasi-likelihood, and the solution to (1) would yield the maximum quasi-likelihood estimate of $\boldsymbol{\beta}$. This is how `glm` works: you specify $g()$ using the `link()` option and you specify $V()$ using the `family()` option, and there are many standard links and families to choose from; see [R] **glm**. Whether you want to think of the resulting $\widehat{\boldsymbol{\beta}}$ as maximum likelihood or maximum quasi-likelihood depends on whether you accept the distributional assumption.

Given the form of (1), the calculations involved in its solution can be separated into those involving $g()$ and those involving $V()$, and Stata takes full advantage of this. For example, suppose you want to fit a GLM with the Poisson variance function and log link. You would then use

```
. glm depvar indep_vars, family(poisson) link(log) ...
```

and when you do so, you are actually specifying two ado-files. `poisson` in the above points to an ado-file in which the calculations pertaining to $V()$ are contained. `log` points to an ado-file containing the calculations pertaining to $g()$. This not only allows links and families to be mixed and matched with ease, but also allows the definition of new links and families by the creation of new ado-files.

In general, the solution to (1) requires some sort of iterative method, such as the method of Newton–Raphson or the method of Fisher scoring. The method of Newton–Raphson (the default in `glm`) involves the matrix of second derivatives of L with respect to $\boldsymbol{\beta}$ (the Hessian), and examination of (1) reveals that this would involve the additional evaluation of $\partial^2 g^{-1}(\eta)/\partial\eta^2$ and $\partial V(\mu)/\partial\mu$. The calculation of the former is contained in the ado-file pertaining to $g()$, and the latter in the ado-file pertaining to $V()$.

The method of Fisher scoring (obtained by specifying `irls` to `glm`) involves using the expected value of the Hessian rather than the observed Hessian, and from (1) it can be shown that this would only involve the already necessary evaluation of $\partial g^{-1}(\eta)/\partial\eta$. This calculation is contained in the ado-file pertaining to $g()$.

In order to program a customized link function, one would thus need only the following information:

1. The calculation of $\eta = g(\mu)$.
2. The calculation of $\mu = g^{-1}(\eta)$.
3. The calculation of $\partial g^{-1}(\eta)/\partial\eta = \partial\mu/\partial\eta$.
4. If estimation via Newton–Raphson is desired, the calculation of $\partial^2 g^{-1}(\eta)/\partial\eta^2 = \partial^2\mu/\partial\eta^2$.

Items 1 and 2 are necessary so that `glm` is able to communicate between those results given in μ and those given in η. Some additional consideration is required concerning the processing of information associated with the binomial variance function, since in `glm`, Bernoulli and binomial count responses are treated jointly.

The implementation of the above is demonstrated by example in the following.

3 Example: Relative survival

3.1 The Hakulinen & Tenkanen model

Consider the relative survival model of Hakulinen and Tenkanen (1987), who considered relative survival rates of those patients with a certain disease to those from the disease-free population. Patient follow-up time is divided into g fixed subintervals $[t_j, t_{j+1})$, for $j = 1, \ldots, g$. The total hazard rate for the ith patient in the jth follow-up is

$$\lambda_{ij}(t) = \exp\{\alpha_j(t) + \mathbf{x}_{ij}\boldsymbol{\beta}\} + \lambda^*_{ij}(t) \tag{2}$$

where $\lambda^*_{ij}(t)$ is the hazard rate for the disease-free population, $\mathbf{x}_{ij}$ is a row–vector of covariates, and $v_{ij}(t) = \lambda_{ij}(t) - \lambda^*_{ij}(t)$ is the hazard rate for the population with the disease of interest. $\alpha_j(t)$ is a constant unique to each follow-up; i.e. a "baseline hazard rate". This model can be seen as a Cox proportional hazards model with an additional additive hazard component.

Given the hazard rates, the survival rates are then calculated as

$$\begin{aligned} r_{ij} &= \exp\left\{-\int_{t_j}^{t_{j+1}} v_{ij}(t)dt\right\} \\ p_{ij} &= \exp\left\{-\int_{t_j}^{t_{j+1}} \lambda_{ij}(t)dt\right\} \\ p^*_{ij} &= \exp\left\{-\int_{t_j}^{t_{j+1}} \lambda^*_{ij}(t)dt\right\} \end{aligned}$$

and from (2), we get

$$\ln\{-\ln(p_{ij}/p^*_{ij})\} = \gamma_j + \mathbf{x}_{ij}\boldsymbol{\beta} \tag{3}$$

where p_{ij} is the survival rate for those in the study, p^*_{ij} is the survival rate for the disease-free population, and $r_{ij} = p_{ij}/p^*_{ij}$ is the relative survival rate, or, equivalently, the survival rate when death is due only to the disease of interest. γ_j is an intercept term unique to each follow-up time.

The survival rate for the disease-free population, p^*_{ij}, is taken to be fixed, as it is usually available from life tables. One of the advantages of this model is that should p^*_{ij} be measured with bias, it may be reasonable to assume that the same (multiplicative) bias applies to p_{ij}, and thus r_{ij} would be free of this bias.

Given fixed p^*, (3) suggests (for the observable data) a binomial model with success probability p_{ij}, with a modified complimentary log-log link function

$$\eta = g(\mu) = \ln\{-\ln(\mu/p^*)\} \tag{4}$$

3.2 Programming the link function

In the calculations that follow, we replace μ by μ/m in the right-hand side of (4), where m is the "binomial denominator", the number of trials for which the number of successes is the response. This is to ensure that our link program can handle both Bernoulli and aggregate count responses. In cases where we have a Bernoulli or other non-binomial response, m is simply set to one.

The required calculations are then given by

$$\begin{aligned}
\mu &= g^{-1}(\eta) = mp^* \exp\{-\exp(\eta)\} \\
\frac{\partial \mu}{\partial \eta} &= -mp^* \exp\{-\exp(\eta)\} \exp(-\eta) = -\mu \exp(\eta) \\
\frac{\partial^2 \mu}{\partial \eta^2} &= -\mu \exp(\eta) \{\exp(\eta) + 1\}
\end{aligned}$$

Given these calculation, we can now define the new link program, `relsurv`, contained in the file `relsurv.ado`, the contents of which are listed below:

(*Continued on next page*)

```
program define relsurv
        version 7
        args todo eta mu return

        if `todo' == -1 {                                       /* Title */
                global SGLM_lt "Hakulinen-Tenkanen"
                if "$SGLM_m" == "1" {
                        global SGLM_lf "ln(-ln(u/$SGLM_p))"
                }
                else {
                        global SGLM_lf "ln(-ln(u/($SGLM_m*$SGLM_p)))"
                }
                exit
        }
        if `todo' == 0 {                                  /* eta = g(mu) */
                gen double `eta' = ln(-ln(`mu'/($SGLM_m*$SGLM_p)))
                exit
        }
        if `todo' == 1 {                                /* mu = g^-1(eta) */
                gen double `mu' = $SGLM_m*$SGLM_p*(exp(-exp(`eta')))
                exit
        }
        if `todo' == 2 {                                /* (d mu)/(d eta) */
                gen double `return' = -`mu'*exp(`eta')
                exit
        }
        if `todo' == 3 {                              /* (d^2 mu)(d eta^2) */
                gen double `return' = -`mu'*exp(`eta')*(exp(`eta')+1)
                exit
        }
        noi di as err "Unknown call to glm link function"
        exit 198
end
```

Some notes:

1. Link programs contain four arguments: `todo`, `eta`, `mu`, and `return`, although it is not critical that they be named as so. `todo` controls the action of the code, with `todo == -1` used as an initialization stage where titles for `glm` output are set. `eta` and `mu` are self-explanatory, and the `return` argument is used for derivatives.

2. Aside from the four arguments, `glm` communicates with the link program via global macros that begin with `SGLM`. `SGLM_m` holds the binomial denominator, or `1` if not used. `SGLM_lt` and `SGLM_lf` hold titles for display.

3. The global macro `SGLM_p` is used to contain an optional argument to the link function. For our purposes, we use `SGLM_p` (coincidentally) to hold the name of the variable containing p^*, the survival rate for the disease-free population.

   Because it is a global macro, however, `SGLM_p` could also be set to contain a constant or matrix name, depending on what is needed for that particular link. To use this link, one specifies the option `link(relsurv` *arg*`)` to `glm`, and `SGLM_p` is set to contain *arg*, whatever that may be. `relsurv` is the name of the program.

4. `todo==3` dictates the calculation of the second derivative of the inverse link, required only if Newton–Raphson estimation is desired. If only IRLS is wanted, then you can simply set `'return'` to missing. In that case, you would then need to specify the `irls` option to `glm`. Otherwise, the program will issue an error.

5. The derivatives may be expressed in terms of `mu`, `eta`, or both, whichever is most convenient.

3.3 Melanoma data

The following data were obtained from Dickman (1998), and represent a subset of skin melanoma cases diagnosed from 1975–1994 and followed up through 1995. The data were originally obtained from the Finnish Cancer Registry (Dickman et al. 1999). Since the data presented below are only a subset of the full study, it is noted in Dickman (1998) that the following is to be used only for illustrative purposes:

```
. describe
Contains data from skin2.dta
  obs:            80
 vars:             8                          28 Oct 2002 11:17
 size:         1,680 (87.2% of memory free)
```

variable name	storage type	display format	value label	variable label
fu	byte	%4.0g		Follow-up period, 1-5
nd	byte	%4.0g		Number died
ld	float	%9.0g		Number in group
ps	float	%9.0g		survival rate (non-diseased)
age	byte	%9.0g	age	Age group
ns	float	%9.0g		Number survived
female	byte	%8.0g	sex	1 if female
dgnyear	byte	%9.0g	dgnyear	year of diagnosis

```
Sorted by:
. list in 1/10, noobs
```

fu	nd	ld	ps	age	ns	female	dgnyear
3	29	321	.97772	60-74	292	female	1975-1984
4	12	242	.98523	45-59	230	male	1975-1984
3	30	205	.90208	75+	175	female	1985-94
5	13	177	.94929	60-74	164	male	1975-1984
3	16	323	.99537	45-59	307	female	1975-1984
4	5	354	.99887	0-44	349	female	1985-94
5	7	31	.86427	75+	24	male	1975-1984
3	16	53	.8748	75+	37	male	1975-1984
2	6	379	.99907	0-44	373	female	1975-1984
3	6	360	.99607	45-59	354	female	1985-94

We fit the relative survival model using the number survived (`ns`) as the response, cohort size (`ld`) as the binomial denominator, and indicator variables for sex, age group, year of diagnosis, and follow-up period as covariates. The name of the variable containing disease-free survival rates, `ps`, is passed as an argument to our link function.

```
. xi: glm ns i.fu female i.age dgnyear, family(binomial ld) link(relsurv ps)
i.fu              _Ifu_1-5            (naturally coded; _Ifu_1 omitted)
i.age             _Iage_1-4           (naturally coded; _Iage_1 omitted)
note: ld has non-integer values
note: ns has non-integer values

Iteration 0:   log likelihood = -213.68868
Iteration 1:   log likelihood = -205.31208
Iteration 2:   log likelihood = -205.15948
Iteration 3:   log likelihood = -205.15886
Iteration 4:   log likelihood = -205.15886

Generalized linear models                         No. of obs      =        80
Optimization     : ML: Newton-Raphson             Residual df     =        70
                                                  Scale param     =         1
Deviance         =  75.71093802                   (1/df) Deviance =  1.081585
Pearson          =  74.96041422                   (1/df) Pearson  =  1.070863

Variance function: V(u) = u*(1-u/ld)              [Binomial]
Link function    : g(u) = ln(-ln(u/(ld*ps)))      [Hakulinen-Tenkanen]
Standard errors  : OIM

Log likelihood   = -205.1588641                   AIC             =  5.378972
BIC              = -231.0309264
```

ns	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_Ifu_2	1.899964	.2984151	6.37	0.000	1.315081	2.484847
_Ifu_3	1.960777	.2990444	6.56	0.000	1.37466	2.546893
_Ifu_4	1.672874	.3071967	5.45	0.000	1.07078	2.274969
_Ifu_5	1.523127	.3145164	4.84	0.000	.9066862	2.139568
female	-.5711185	.0971435	-5.88	0.000	-.7615164	-.3807207
_Iage_2	.3248696	.1251922	2.59	0.009	.0794974	.5702419
_Iage_3	.6390242	.1282907	4.98	0.000	.3875791	.8904693
_Iage_4	1.161009	.1723661	6.74	0.000	.8231777	1.498841
dgnyear	-.4640116	.0977171	-4.75	0.000	-.6555337	-.2724895
_cons	-4.998121	.3036738	-16.46	0.000	-5.593311	-4.402932

and we note that the results match those of Dickman (1998).

Besides ease, an advantage of programming this model into `glm` rather than writing our own `ml` program is that all the diagnostic tools contained in `glm` are now available to us. For example, a common diagnostic is a plot of Pearson residuals versus the linear predictor, easily obtained by typing

```
. predict xbeta, xb
. predict pearson, pearson
. graph pearson xbeta
```

which produces Figure 1. In this case, the plot reveals no visible problems with the model fit.

(*Continued on next page*)

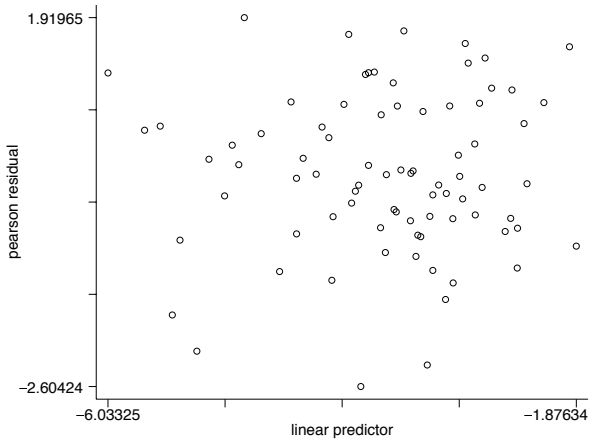

Figure 1: Pearson residuals versus linear predictor

4 Example: Natural response

4.1 A modified logit link

Collett (2003) describes a logistic regression model that adjusts for response due to factors other than those of interest, known as *natural response* (or *natural mortality* when appropriate). For a count response y_i, assume that $y_i \sim \text{Bin}(n_i, \mu_i)$ for $i = 1, \ldots, n$. The observable response probability, μ_i, is taken to be

$$\mu_i = \pi + (1 - \pi)\mu_i^* \tag{5}$$

where π is taken to be some fixed (or well-estimated) probability of response to factors outside those of interest, and μ_i^* is the probability of a response due to the factors of interest. What one observes is either the natural response or, given no natural response, a response due to the factors of interest. Equation (5) is known as Abbott's formula.

If one assumes a logit model for the true response probability,

$$\log\left(\frac{\mu_i^*}{1 - \mu_i^*}\right) = \mathbf{x}_i\boldsymbol{\beta}$$

where $\mathbf{x}_i$ are the factors of interest, then combining this with (5) results in the following model for the observable response probability:

$$\log\left(\frac{\mu_i - \pi}{1 - \mu_i}\right) = \mathbf{x}_i\boldsymbol{\beta} = \eta_i \tag{6}$$

Of course, one could estimate π jointly with $\boldsymbol{\beta}$ via maximum likelihood, but in the context of GLM, one must treat π as fixed since it does not pertain to the linear predictor.

4.2 The link program

From (6), including the binomial denominator (m) and suppressing the subscripts, we obtain

$$\mu = m\{\exp(\eta) + \pi\}\{1 + \exp(\eta)\}^{-1}$$

Calculating first and second derivatives yields the following code for the logit link function with natural response:

```
program define logit_nr
        version 7
        args todo eta mu return

        if `todo' == -1 {                                       /* Title */
                global SGLM_lt "Logit - natural response"
                if "$SGLM_m" == "1" {
                        global SGLM_lf "ln((u-$SGLM_p)/(1-u))"
                }
                else {
                        global SGLM_lf "ln((u-$SGLM_m*$SGLM_p)/($SGLM_m-u))"
                }
                exit
        }
        if `todo' == 0 {                                  /* eta = g(mu) */
                gen double `eta' = ln((`mu'-$SGLM_m*$SGLM_p)/($SGLM_m-`mu'))
                exit
        }
        if `todo' == 1 {                                /* mu = g^-1(eta) */
                gen double `mu' = $SGLM_m*(exp(`eta') + $SGLM_p) / /*
                        */ (1 + exp(`eta'))
                exit
        }
        if `todo' == 2 {                                /* (d mu)/(d eta) */
                gen double `return' = $SGLM_m*exp(`eta')*(1-$SGLM_p) / /*
                        */ ((1 + exp(`eta'))^2)
                exit
        }
        if `todo' == 3 {                              /* (d^2 mu)(d eta^2) */
                gen double `return' = $SGLM_m*exp(`eta')*(1-$SGLM_p)* /*
                        */ (1-exp(`eta')) / ((1 + exp(`eta'))^3)
                exit
        }
        noi di as err "Unknown call to glm link function"
        exit 198
end
```

Note that this time the global macro `SGLM_p` holds the value of π, which we have been considering constant. However, given the way `SGLM_p` is used in the program, our code would apply even if π were instead π_i and varied over the data, in which case, `SGLM_p` would hold the name of variable containing the values of π_i.

4.3 Analysis of flour beetle data

As was done in Collett (2003), we apply the logit model with natural response to data on flour beetle response to insecticide treatment.

```
. use flour
```

```
. list
     insecti~e    deposit          y          n
  1.       DDT          2          3         50
  2.       DDT       2.64          5         49
  3.       DDT       3.48         19         47
  4.       DDT       4.59         19         50
  5.       DDT       6.06         24         49
  6.       DDT          8         35         50
  7.       BHC          2          2         50
  8.       BHC       2.64         14         49
  9.       BHC       3.48         20         50
 10.       BHC       4.59         27         50
 11.       BHC       6.06         41         50
 12.       BHC          8         40         50
 13. DDT & BHC          2         28         50
 14. DDT & BHC       2.64         37         50
 15. DDT & BHC       3.48         46         50
 16. DDT & BHC       4.59         48         50
 17. DDT & BHC       6.06         48         50
 18. DDT & BHC          8         50         50
```

Data were collected on $n = 18$ batches of flour beetles, each approximately 50 beetles in size. Three insecticide treatments were applied (`insecticide`) to six levels of spray deposit (`deposit`). The response `y` is the number killed, whether by the insecticide or otherwise. A control group of 200 beetles were also examined, of which 20 died, yielding an estimated natural mortality rate of 0.10.

A binomial model with the natural response link was then fit to the data, using the insecticide type and the natural logarithm of deposit level as covariates.

```
. generate log_dep = log(deposit)
. xi: glm y i.insecticide log_dep, family(binomial n) link(logit_nr 0.10)
i.insecticide     _Iinsectici_1-3     (naturally coded; _Iinsectici_1 omitted)
Iteration 0:   log likelihood = -47.552937
Iteration 1:   log likelihood = -44.947778
Iteration 2:   log likelihood = -44.930099
Iteration 3:   log likelihood = -44.930092
Iteration 4:   log likelihood = -44.930092
Generalized linear models                         No. of obs      =         18
Optimization     : ML: Newton-Raphson             Residual df     =         14
                                                  Scale param     =          1
Deviance         =  26.38923716                   (1/df) Deviance =  1.884946
Pearson          =  24.29630348                   (1/df) Pearson  =   1.73545
Variance function: V(u) = u*(1-u/n)               [Binomial]
Link function    : g(u) = ln((u-n*0.10)/(n-u))    [Logit - natural response]
Standard errors  : OIM
Log likelihood   =  -44.9300923                   AIC             =  5.436677
BIC              = -14.07596745
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_Iinsectic~2	.9098902	.247889	3.67	0.000	.4240367	1.395744
_Iinsectic~3	3.637506	.3221903	11.29	0.000	3.006024	4.268987
log_dep	3.113487	.2763729	11.27	0.000	2.571806	3.655168
_cons	-5.634301	.5048525	-11.16	0.000	-6.623794	-4.644809

In this case, ignoring the effect of natural mortality and simply fitting a standard binomial/logit model yielded somewhat different, but not substantially different estimates.

5 Concluding remarks

The purpose of this paper was to demonstrate the relative ease with which link functions could be programmed into `glm` with Stata 7. The code required was shown to be compact, general, and isolated to the creation of one new ado-file.

Although one may also program their own variance functions, the need to do so is not well-demonstrated in the literature. The most popular example of a nonstandard variance function is the squared-binomial variance function of Wedderburn (1974), as done using Stata in Hardin and Hilbe (2001). One obstacle to the creation of other variance functions is the limited class of such functions yielding an analytical form for the quasi-deviance. Another is the absence of need for new variance functions, given the flexibility of the already standard power family, where the variance is taken to be some general power of the mean.

6 Acknowledgments

We thank Paul Dickman of the Karolinska Institutet in Stockholm and Andrew Sloggett of the London School of Hygiene and Tropical Medicine, Centre for Population Studies, who worked with us to develop a Stata 6 version of `glm` with the Hakulinen & Tenkanen link function. We also thank Timo Hakulinen of the Finnish Cancer Registry, for granting us access to the melanoma data.

7 References

Collett, D. 2003. *Modelling Binary Data*. 2d ed. Boca Raton: Chapman & Hall/CRC.

Dickman, P. 1998. *Fitting the Hakulinen and Tenkanen relative survival model to the localised skin melanoma.*
http://www.pauldickman.com/book/melanoma/hakulinen

Dickman, P., T. Hakulinen, T. Luostarinen, E. Pukkala, R. Sankila, B. Söderman, and L. Teppo. 1999. Survival of Cancer Patients in Finland 1955-1994. *Acta Oncologica* 38 (Suppl. 12): 1–103.

Hakulinen, T. and L. Tenkanen. 1987. Regression analysis of relative survival rates. *Applied Statistics* 3: 309–317.

Hardin, J. and J. Hilbe. 2001. *Generalized Linear Models and Extensions*. College Station, TX: Stata Press.

Nelder, J. A. and R. W. M. Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society: Series A* 135(3): 370–384.

Wedderburn, R. W. M. 1974. Quasi–likelihood functions, generalized linear models, and the Gauss–Newton method. *Biometrika* 61(3): 439–447.

About the Authors

Weihua Guan is a Statistician at Stata Corporation.
Roberto G. Gutierrez is Director of Statistics at Stata Corporation.