The Stata Journal () , Number , pp. 1–11

The Regression Calibration Method for Fitting Generalized Linear Models with Additive Measurement Error

James W. Hardin
Norman J. Arnold School of Public Health
University of South Carolina
Columbia, SC 29208

Henrik Schmiediche
Department of Statistics MS-3143
Texas A&M University
College Station, TX 77843-3143

Raymond J. Carroll
Department of Statistics MS-3143
Texas A&M University
College Station, TX 77843-3143

Abstract. This paper discusses and illustrates the method of regression calibration. This is a straightforward technique for fitting models with additive measurement error. We present this discussion in terms of generalized linear models (GLMs) following the notation defined in Hardin and Carroll (2003). Discussion will include specified measurement error, measurement error estimated by replicate error-prone proxies, and measurement error estimated by instrumental variables. The discussion focuses on software developed as part of a small business innovation research (SBIR) grant from the National Institutes of Health (NIH). The authors of this paper served as consultants to Stata Corporation on that grant.

Keywords: regression calibration, measurement error, instrumental variables, replicate measures, generalized linear models

1 Introduction

This paper describes software for analyzing measurement error models. This software production was funded by a National Institutes of Health (NIH) Small Business Innovation Research Grant (SBIR) (2R44RR12435-02) which was managed by Stata Corporation. The goal of the work described in the grant is the production of software to analyze statistical models where one or more covariates are measured with error. The software development included two major features. The first development feature is the development of the Stata program to support communication to dynamically linked user-written computer code. Stata Corporation was responsible for this development and support for user-written code in the C language was added to Stata version 8. Stata refers to compiled user-written code as a plugins and maintains documentation on their website at `http://www.stata.com/support/plugins`. See Hardin and Carroll (2003) for an introduction.

The project described was supported by Grant Number R44 RR12435 from the National Institutes of Health, National Center for Research Resources. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the National Center for Research Resources.

Following the presentation in Carroll et al. (1995), we discuss the regression calibration method for correcting bias in fitting generalized linear measurement error models.

The basis of the regression calibration algorithm for measurement error analysis is the construction of the calibration model for generation of estimated covariate values $\widehat{\mathbf{X}}$ for the unknown covariates $\mathbf{X_U}$. After all, were we privy to $\mathbf{X_U}$ there would be no issue. Somehow, we must estimate the missing information in such a way as to eliminate bias in the estimated coefficients and we must take into account this estimation when constructing variance estimates.

Given the resulting bias discussed in Hardin and Carroll (2003), we must first build the calibration function so that the bias can be removed from the results of the standard analysis. The following section provides the formula and explanations for how this is done.

This article focuses on one possible approach to addressing measurement error. Our discussion is meant to provide the reader with an understanding of the statistical concepts and theory behind the regression calibration method. Just as important, we also wish to communicate the capabilities of the software developed to include regression calibration functionality.

2 Regression calibration

The regression calibration method is a simple approach wherein we need only develop and fit the calibration model for the regression of the unknown covariates $\mathbf{X_U}$ on $(\mathbf{X_Z}, \mathbf{X_W})$. This is accomplished using replication, validation, or instrumental data in place of the unknown $\mathbf{X_U}$. This first stage regression results in a calibration function for estimating $\mathbf{X_U}$. The unobserved covariates are then replaced by their predicted values from the calibration model in a standard analysis. Finally, the standard errors are adjusted to account for the estimation of the unknown covariates. The typical approach is to calculate standard errors using bootstrap or sandwich methods, but asymptotic standard errors have been derived by one of the authors (Carroll) and are included in the associated software.

With replicate data, the measurement error variance may be estimated by

$$\widehat{\mathbf{\Sigma}}_{uu} = \frac{1}{\sum_{i=1}^{n}(k_i - 1)} \sum_{i=1}^{n} \sum_{j=1}^{k_i} (\mathbf{W}_{ij} - \overline{\mathbf{W}}_{i\cdot})(\mathbf{W}_{ij} - \overline{\mathbf{W}}_{i\cdot})^{\mathrm{T}}$$

Alternatively, the user may specify this variance matrix if it is known or estimated externally (to the dataset in use).

We need information to substitute for $\mathbf{X_U}$ and we know that the best linear approximant to $\mathbf{X_U}$ given $(\mathbf{Z}, \overline{\mathbf{W}})$ is

$$\widehat{\mathbf{X}}_{\mathbf{U}} \approx \mu_{\mathbf{X_U}} + \begin{pmatrix} \mathbf{\Sigma}_{xx} \\ \mathbf{\Sigma}_{zx} \end{pmatrix} \begin{bmatrix} \mathbf{\Sigma}_{xx} + \mathbf{\Sigma}_{uu}/k & \mathbf{\Sigma}_{xz} \\ \mathbf{\Sigma}_{zx} & \mathbf{\Sigma}_{zz} \end{bmatrix}^{-1} \begin{pmatrix} \overline{\mathbf{W}} - \mu_{\mathbf{W}} \\ \mathbf{Z} - \mu_{\mathbf{Z}} \end{pmatrix} \tag{1}$$

In order to operationalize this linear approximant, we make the usual substitutions for unknown quantities. First, we use $\widehat{\mu}_{\mathbf{W}} = \mu_{\mathbf{X_U}}$; substituting the mean of the replicate values for the mean of the unknown covariates. In addition,

$$\widehat{\mathbf{\Sigma}}_{zz} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{Z}_i - \overline{\mathbf{Z}}_\cdot)(\mathbf{Z}_i - \overline{\mathbf{Z}}_\cdot)^{\mathrm{T}}$$

as the usual analysis of variance estimate. We use

$$\begin{aligned}
\widehat{\mathbf{\Sigma}}_{xz} &= \frac{1}{\nu}\sum_{i=1}^{n} k_i(\overline{\mathbf{W}}_{i\cdot} - \widehat{\mu}_{\mathbf{W}})(\mathbf{Z}_i - \overline{\mathbf{Z}}_\cdot)^{\mathrm{T}} \\
\widehat{\mathbf{\Sigma}}_{xx} &= \frac{1}{\nu}\left\{\sum_{i=1}^{n} k_i(\overline{\mathbf{W}}_{i\cdot} - \widehat{\mu}_{\mathbf{W}})(\overline{\mathbf{W}}_{i\cdot} - \widehat{\mu}_{\mathbf{W}})^{\mathrm{T}}\right\} - \frac{n-1}{\nu}\widehat{\mathbf{\Sigma}}_{uu}
\end{aligned}$$

where $\nu = \sum_i k_i - \sum_i k_i^2 / \sum_i k_i$. The estimated variance matrix for the unknown $\mathbf{X_U}$ is seen in two components due to the variance of $\mathbf{X_U}$ and the measurement error variance.

Armed with these estimates, it is straightforward to mechanically derive the estimated values for the unknown $\mathbf{X_U}$, produce estimates $\widehat{\mathbf{X}}_{\mathbf{U}}$ using equation 1, and proceed with a standard analysis. Obviously, calculation of a variance matrix for the estimated coefficients will have to address the additional parameters from this substitution.

3 Comparison to current Stata commands

Hardin and Carroll (2003) illustrate the effect of measurement error in linear regression. Clearly, if we had an estimate of the unknown measurement error variance, we could adjust the results from an analysis using the error-prone covariates. In fact, the [R] **eivreg** command does this by requiring a specification of the attenuation factor (reliability ratio). To see the relationship, let's follow the example from [R] **eivreg**.

```
. eivreg price foreign weight, r(weight .85)
                assumed                          errors-in-variables regression
variable     reliability
------------------------                                Number of obs =      74
  weight          0.8500                                F(  2,    71) =   50.37
       *          1.0000                                Prob > F      =  0.0000
                                                        R-squared     =  0.6483
                                                        Root MSE      = 1773.54

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     foreign |     4637.32   624.5362     7.43   0.000      3392.03    5882.609
      weight |     4.31985    .431431    10.01   0.000     3.459601    5.180099
       _cons |    -8257.017   1452.086    -5.69   0.000    -11152.39    -5361.64
------------------------------------------------------------------------------
```

Since the `weight` variable is 85% reliable, we know that 15% of the variance for that error-prone variable is due to the measurement error variance. It is easy to calculate this:

```
. summ weight
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      weight |        74    3019.459    777.1936       1760       4840

. local suu = .15*r(Var)

. mat uinit = (‘suu’)
```

We put the estimated variance into a matrix because the `rcal` command requires this. We do not have any replicate measurements, so we will specify the same measurement error variance as was specified as a reliability to the `eivreg` command.

```
. rcal (price=foreign) (www: weight), suuinit(uinit) suuignore
Regression calibration                          No. of obs         =        74
Link         = Identity
Residual df  =          71                      Wald F(2,71)       =     25.15
                                                Prob > F           =    0.0000
               (IRLS EIM)                       Scale param        =   4481776
Variance Function: V(u) = 1                     [Gaussian]
Link Function    : g(u) = u                     [Identity]

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     foreign |    4637.32    873.833     5.31   0.000     2894.946    6379.693
         www |    4.31985   .6103749     7.08   0.000     3.102797    5.536903
       _cons |  -8257.017   1345.591    -6.14   0.000    -10940.05   -5573.985
------------------------------------------------------------------------------
```

In this example, it is assumed that the `weight` variable is 85% reliable. Alternatively, we say that `weight` is an error prone covariate for the unobserved true covariate. We assume that `weight` is equal to the unobserved covariate plus error. The total variance for the `weight` variable is equal to the variance of the unobserved covariate plus the variance of the measurement error.

Aside from the peculiarities of the syntax for the new `rcal` command, the resulting adjustment made to remove the bias from the estimated coefficients is the same for these commands. The scale parameter is equal to the MSE from the `eivreg` command. The difference in algorithms for calculating the variance matrix is obvious, but the standard errors are approximately of the same scale. We can investigate this further by requesting bootstrap standard errors from the `rcal` command.

```
. rcal (price=foreign) (www: weight), suuinit(uinit) brep(299)
Regression calibration                          No. of obs         =        74
                                                Bootstrap reps     =       299
Residual df  =          71                      Wald F(2,71)       =     17.92
                                                Prob > F           =    0.0000
               (IRLS EIM)                       Scale param        =   4481776
Variance Function: V(u) = 1                     [Gaussian]
Link Function    : g(u) = u                     [Identity]

------------------------------------------------------------------------------
             |                 Bootstrap
```

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
foreign	4637.32	901.7399	5.14	0.000	2839.301	6435.338
www	4.31985	.7216277	5.99	0.000	2.880965	5.758735
_cons	-8257.017	2436.941	-3.39	0.001	-13116.14	-3397.894

4 Syntax and options

The previous section illustrated a very basic use of the `rcal` command. Generally, there are two or more parenthesized equations specified to the command. The first parenthesized equation is the dependent variable $\mathbf{Y}$, followed by an equal sign, followed by the list of covariates measured without error $\mathbf{X_Z}$.

Subsequent parenthesized equations list information on the measurement error variables. For each unobserved variable, an equation is specified with a label, followed by a colon, followed by the list of replicate variables for that unobserved variable. Hardin and Carroll (2003) explain that these lists must all be the same length and the order matters when specifying the replicate variables for the unobserved variable.

If the measurement error equations all have a single replicate, the user must specify the variance for the measurement error variance. This is specified with the `suuinit()` option which names a matrix communicating the measurement error variance. This matrix should be square and of dimension the number of measurement error equations.

For example,

```
rcal (y=z1 z2 z3) (x1: w11 w12 w13) (x2: w21 w22 w23)
```

specifies a model with dependent variable `y`, $\mathbf{X_Z}$ given by [`z1`, `z2`, `z3`], and includes information for two unobserved variables. The labels `x1` and `x2` are restricted to not coincide with any variable names that exist in the dataset. The unobserved variable represented by the `x1` label will be estimated by the 3 replicates stored in the `w11`, `w12`, and `w13` variables. Likewise, we have a similar description for the unobserved variable represented by the `x2` label. Utilizing the notation laid out in Hardin and Carroll (2003), we have $\mathbf{X_U}$ represented by [`x1`, `x2`]. The replicate observed variables are $\mathbf{X_W}$ specified by [(`w11`, `w12`, `w13`) , (`w21`, `w22`, `w23`)]. Writing the $\mathbf{X_W}$ matrix in this manner makes it clear that there are two unknown covariates so that the dimension of the measurement error variance is (2×2).

Since there are three possible replicates for each observation for the two different unobserved variables in this particular model, the user is not required to specify the measurement error variance. However, this does not preclude the user from providing this estimate if it is available.

An additional consideration of user-specified measurement error variances is the confidence of the estimate. If the measurement error variance is specified, then the variability of obtaining this estimate is ignored in the calculation of the standard errors

for the model of interest. One can also ignore this variability while still estimating the measurement error variance via replicates using the `uuignore` option.

4.1 Computational note on calculating standard errors

The calculation of the asymptotic and sandwich standard errors are very computationally intensive. Since much time was spend optimizing the algorithm, the results will generally be available within seconds for small to moderate size data sets. For large data sets, however, the time required to compute the standard errors can be very long. Should `rcal` require more than 30 seconds to provide an answer, it will display an estimate of the time it will take to complete the calculation.

The `rcal` command implements a fast internal bootstrap which for large data sets is *significantly* faster than the default asymptotic standard errors (when using the default or a reasonable number of bootstrap replicates). The larger the data set the more of a speed advantage the bootstrap will have over asymptotic standard errors. In one case a hundred fold speed differential was observed for a data with of 100,000 observations.

It is also possible to calculate the naive variance regression calibration estimate using the `naive` option. Naive standard errors are calculated by generating the missing covariates and assuming them to be true. In general this option is for pedagogical and diagnostic purposes only since the standard errors it calculates are incorrect. One advantage naive estimates do have is that they require no special computational effort. That makes them useful when testing the `rcal` command and options on a large data set. Another use would be if no standard errors are desired, say if one wanted to use the `rcal` command to calculate the jackknife standard errors.

5 Building the missing covariate

The `rcal` command does not directly support generation of the missing covariates. Often, researchers think that the missing covariate is simply replaced by the mean of the replicates, but this is not true. See equation 1 for clarification.

In the do-file below, we generate some data to illustrate how you can build the covariate values that are used in fitting a GLM via the regression calibration method.

```
set seed 139423
set obs 100
gen z1 = uniform()
gen z2 = invnorm(uniform())
gen x1 = uniform()
gen w1 = x1 + invnorm(uniform())*.5
gen w2 = x1 + invnorm(uniform())*.5
gen w3 = x1 + invnorm(uniform())*.5
gen y  = z1 + 2*z2 + 3*x1 + 4 + invnorm(uniform())

gen wbar = (w1+w2+w3)/3

reg y z1 z2 wbar
```

```
rcal (y=z1 z2) (what: w1 w2 w3)
```

Note how simply replacing the missing covariate with the mean of the replicates does not generate the same point estimates estimated using regression calibration. The correct formula given in equation 1 can be calculated using the saved results of the `rcal` command.

```
mat szz = e(szz)
mat sxx = e(sxx)
mat sxz = e(sxz)
mat suu = e(suu)

mat a  = sxx  sxz'
mat b1 = sxx + suu/3 , sxz
mat b2 = sxz', szz
mat b  = b1  b2
mat c  = a' * syminv(b)

summ z1
gen z1b = z1 - r(mean)
summ z2
gen z2b = z2 - r(mean)
summ wbar
gen wbarb = wbar - r(mean)

gen xh = c[1,1] * wbarb + c[1,2] * z1b + c[1,3] * z2b
```

Once we build the missing covariate, we can verify our calculations using `regress`.

```
. rcal (y=z1 z2) (xhat: w1 w2 w3)
Regression calibration                          No. of obs         =       100
Link         = Identity
Residual df  =          96                      Wald F(3,96)       =    104.32
                                                Prob > F           =    0.0000
               (IRLS EIM)                       Scale param        =  1.437038
Variance Function: V(u) = 1                     [Gaussian]
Link Function    : g(u) = u                     [Identity]

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          z1 |   .4172764   .4620955     0.90   0.369    -.499976    1.334529
          z2 |   2.235349   .1274325    17.54   0.000    1.982397      2.4883
        xhat |   2.252182   .6978701     3.23   0.002    .8669208    3.637443
       _cons |   4.828447    .258867    18.65   0.000      4.3146    5.342294
------------------------------------------------------------------------------

. reg y z1 z2 xh
      Source |       SS       df       MS              Number of obs =     100
-------------+------------------------------           F(  3,    96) =  109.72
       Model |  472.993992     3  157.664664            Prob > F      =  0.0000
    Residual |  137.955609    96   1.4370376           R-squared     =  0.7742
-------------+------------------------------           Adj R-squared =  0.7671
       Total |  610.949602    99   6.1712081           Root MSE      =  1.1988

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          z1 |   .4172764    .457687     0.91   0.364    -.4912252    1.325778
          z2 |   2.235349   .1265596    17.66   0.000     1.98413    2.486567
```

```
------------------------------------------------------------------------------
          xh |   2.252182   .5015322     4.49   0.000     1.256648    3.247715
       _cons |   5.795312   .2456567    23.59   0.000     5.307688    6.282937
------------------------------------------------------------------------------
```

This do-file illustrates how the missing covariate is estimated. The generation of the missing variable in the Stata dataset is specific to the size of the problem that we simulated, but the mechanics of the operation should be apparent.

6 NHANES example

Using data from the National Health And Examination Survey (NHANES), we investigate the presence of breast cancer `qbc` as a function of other covariates; `qage` is the age in years of the patient, `pir` is the poverty index ratio, `qbmi` is the body mass index, `alcohol` is indicator for whether the individual uses alcohol, `famhist` is an indicator of whether there is a family history of breast cancer, and `agemen` is the age at menarche. Two additional covariates `qcalorie` and `qsatfat` are measurements recorded as recalls for the individual on their saturated caloric fat intake.

We fit a regression calibration model calculating bootstrap standard errors. In this model, we specify that `qcalorie` and `satfat` are replicate measures for an unknown covariate measure of the saturated fat intake of the individual. This model is fit in much less time than an equivalent SIMEX model—see Hardin et al. (2003) for comparison.

```
. local y "qbc"
. local z "qage pir qbmi alcohol famhist agemen"
. local x "qcalorie qsatfat"
. rcal (‘y’ = ‘z’) (w: ‘x’), fam(bin) brep(199) seed(12394)
Regression calibration                          No. of obs         =      3145
                                                Bootstrap reps     =       199
Residual df   =        3137                     Wald F(7,3137)     =      4.35
                                                Prob > F           =    0.0001
               (IRLS EIM)                       Scale param        =  1.068394
Variance Function: V(u) = u(1-u)                [Bernoulli]
Link Function    : g(u) = log(u/(1-u))          [Logit]

------------------------------------------------------------------------------
             |               Bootstrap
         qbc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        qage |   .0648976   .0169298     3.83   0.000      .031703    .0980921
         pir |   .1283659   .0756971     1.70   0.090     -.020055    .2767867
        qbmi |  -1.484298   2.451664    -0.61   0.545    -6.291326    3.322731
     alcohol |    .446928   .2893229     1.54   0.123    -.1203533    1.014209
    famhist |   .6662337   .4817085     1.38   0.167    -.2782621    1.610729
     agemen |  -.1504096    .270934    -0.56   0.579    -.6816354    .3808162
          w |  -.0463208   .0224759    -2.06   0.039    -.0903898   -.0022519
      _cons |   -5.28918   1.066968    -4.96   0.000    -7.381205   -3.197154
------------------------------------------------------------------------------
```

7 Formal Stata syntax

`rcal (` *depvar* `=` [*varlist*] `)` [([*label* `:`]*varlist*) ... ([*label*`:`]*varlist*)] [*weight*]
[`if` *exp*] [`in` *range*] [`,` `message(`#`)` `family(`*string*`)` `link(`*string*`)`
`ltolerance(`#`)` `iterate(`#`)`

`level(`#`)` `robust` `naive` `suuinit(`*matrixname*`)` `suuignore`

`bstrap` `brep(`#`)` `btrim(`#`)` `seed(`#`)` `saving(`*string*`)` `replace`]

General Options

`message(`#`)` specifies the desired (observed) level of printed messages of the plugin module. Users can use this option to suppress or request warning and informational messages.

0) Display nothing, not even fatal error messages.

1) Display print fatal error messages only.

2) Display warning messages (default).

3) Display informational messages.

4) Display more informational messages.

Note that the ADO code that handles the I/O to the plugin may still print error messages regardless of the message level setting.

The message command can also be used to see intermediate details of the internal calculations of the code. These where used by the authors to debug the code. The notation and mnemonics used are not documented and may not correspond to anything in the printed documentation. Furthermore the numbers may be in a raw and unadjusted format that are difficult to interpret.

5, 6 & 7) Display details with increasing verbosity.

Messages levels are cumulative.

`family(`*string*`)` specifies the distribution of the dependent variable. The `gaussian` family is the default. The choices and valid family and link combinations are the same as for Stata's `glm` command.

`link(`*string*`)` specifies the link function; the default is the canonical link for the cmd:family() specified. The choices and valid family and link combinations are the same as for Stata's `glm` command.

`ltolerance(`#`)` specifies the convergence criterion for the change in deviance between iterations. The default is $10^{-}6$.

`iterate(#)` specifies the maximum number of iterations allowed in fitting the model; The default is 100. It is rare that one needs to increase this.

Standard Error Options

`level(#)` specifies the confidence level, in percent, for confidence intervals of the coefficients.

`robust` specifies that the Huber/White/sandwich estimator of variance is to be used in place of the traditional calculation. We do not support the `cluster` option in `rcal`.

`naive` specifies that the naive variance estimate should be used. This means that the missing covariates measured with error are generated, but then assumed to be true. This option is for pedagogical and diagnostic purposes and should not be otherwise used.

`suuinit(`*matrixname*`)` specifies the measurement error covariance matrix. This is calculated from the replications in the measurement error variables if it is not specified.

`suuignore` specifies that the error associated with estimating the measurement error should be ignored. This option could be specified if the user provides an estimate of the measurement error from an external source believed to be true. This may be relevant if the covariance comes from a large, careful independent study, for which only summary statistics are available. Note that this does not mean that the measurement error is ignored, but rather the error associated with estimating it is ignored.

Bootstrap Options

`bstrap` specifies that bootstrap standard errors should be calculated. The bootstrap is internal to the code for the regression calibration command. The estimated time to perform the bootstrap will be displayed should the bootstrap require more than 30 seconds.

`brep(#)` specifies the number of bootstrap samples generated to calculate the bootstrap standard errors of the fitted coefficients. The default is `199`.

`btrim(#)` specifies the amount of trimming applied to the collection of bootstrap samples prior to calculation of the bootstrap standard errors. The default is `.02` meaning that 1% of the samples (rounded) will be trimmed at each end.

When the bootstrap is run with `mess(3)` an informational message similar to this one will display:

```
Average number of iterations per   GLM call: 3.0
Maximum number of iterations for a GLM call: 3
Minimum number of iterations for a GLM call: 3
Trimming total of 4 bootstrap replications (2.0%).
Maximum change in standard errors due to trimming: 2.4%
```

indicating that 4 samples (2 on each end) where trimmed and that this trimming resulted in a 2.4% change in magnitude of one of the standard errors. All other

standard errors changed less than 2.4%. This simple diagnostic gives an indication on how trimming influenced the bootstrap standard errors.

`seed(#)` specifies a random number seed used in generating random samples for the bootstrap calculations. This option has no effect if bootstrapping is not specified. Its main purpose is to allow repeatability of bootstrap results. The default is `0` which will seed the random number generator using the system clock.

`saving(`*string*`)` specifies the file to which the bootstrap results will be saved.

`replace` the existing 'bootstrap results' file if it exists.

8 References

Carroll, R. J., D. Ruppert, and L. A. Stefanski. 1995. *Measurement Error in Nonlinear Models*. London: Chapman & Hall.

Hardin, J. W. and R. J. Carroll. 2003. Measurement Error, GLMs, and Notational Conventions. *Stata Journal* ??(??): ???–???

Hardin, J. W., H. Schmiediche, and R. J. Carroll. 2003. The Simulation Extrapolation Method for Fitting Generalized Linear Models with Additive Measurement Error. *Stata Journal* ??(??): ???–???

About the Authors

James W. Hardin (jhardin@stat.tamu.edu), is a Research Scientist in the Center for Health Services and Policy Research and an Associate Research Professor in the Department of Epidemiology and Biostatistics at the Norman J. Arnold School of Public Health, University of South Carolina, Columbia, SC 29208, USA.

Henrik Schmiediche (henrik@stat.tamu.edu), is a Senior Lecturer and Senior Systems Analyst, Department of Statistics, MS 3143, Texas A&M University, College Station, TX 77843-3143, USA.

Raymond J. Carroll (carroll@stat.tamu.edu) is a Distinguished Professor, Department of Statistics and Department of Biostatistics & Epidemiology, MS 3143, Texas A&M University, College Station, TX 77843-3143, USA.

The authors' research was supported by the National Institutes of Health (NIH) Small Business Innovation Research Grant (SBIR) (2R44RR12435-02) with Stata Corporation, 4905 Lakeway Drive, College Station, TX 77845, USA.