5-8-2015

Diagnostics and Model Selection for Generalized Linear Models and Generalized Estimating Equations

Chelsea Boquet Deroche
University of South Carolina - Columbia

DIAGNOSTICS AND MODEL SELECTION FOR GENERALIZED LINEAR MODELS AND GENERALIZED ESTIMATING EQUATIONS

by

Chelsea Boquet Deroche

Bachelor of Science
Nicholls State University, 2009

Master of Science
Louisiana State University, 2011

Submitted in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy in

Biostatistics

The Norman J. Arnold School of Public Health

University of South Carolina

2015

Accepted by:

James W. Hardin, Major Professor

Suzanne McDermott, Committee Member

Bo Cai, Committee Member

Kevin Bennett, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

DEDICATION

I dedicate this dissertation to my loving and understanding husband Joshua Deroche, my encouraging and compassionate parents Kevin and Martina (Tina) Boquet and sister Lindsey, and my supportive and forever proud grandparents William (Bill) and Katherine (Kat) Smith. Your unlimited support, encouragement, constant love and patience has sustained me throughout this process. I appreciate everything you have done for me, and I am blessed to have you all in my life. I love you all more than you all will ever know.

ACKNOWLEDGEMENTS

I would like to thank my major advisor, Dr. James W. Hardin for agreeing to mentor me and challenging me in the process. You have been extremely supportive, very patient, and encouraging throughout the entire dissertation process. Thank you for being there for the many weekly meetings, long coding sessions, manuscript and dissertation edits, and random life advice. You mentorship has strengthened my statistical writing skills and statistical consulting ability. Thank you for taking the time to sit down with me and teach me to code in Stata. Without your creative suggestions, I would not have been able to discover so many areas of research in biostatistics.

I would also like to extend my sincere gratitude to Dr. Suzanne McDermott for taking me on as her graduate research assistant and mentoring me during the past three years. You have been motivational and encouraging. Working with you has strengthened my research skills and has opened my eyes to the statistical needs for researchers. Without your generous financial support and advice during my PhD studies, I would not have been able to be as successful. Thank you for being there for me not only in my work, but also in my life and future endeavors.

Thank you, Dr. Bo Cai for instructing me in your expertise of longitudinal data analysis and Bayesian analysis and for being patient with me in the process. Your keen eye has kept me from making careless errors in my work, and I appreciate all the work you have put into proofreading my dissertation.

I would also like to thank Dr. Kevin Bennett for agreeing to be part of this dissertation process. Your insightful advice for examples and quick feedback on my work has helped me succeed. I appreciate your time and effort towards helping me throughout this process.

ABSTRACT

The use of generalized linear models and generalized estimating equations in the public health and medical fields are important tools for research, specifically for modeling clinical trials, evaluating preventive measures, and secondary data analysis. It is important for these researchers to have the necessary tools to analyze and model their data correctly. This dissertation focuses on a penalized maximum likelihood estimation method for generalized linear models, measures of association such as the coefficient of determination and R^2 for generalized estimating equations, and a modified quasi-likelihood information criterion for generalized estimation equations.

Common problems that arise during estimation of generalized linear models are bias of the estimates, small sample size, or complete or quasi-complete separation of data points. To address these problems, the first part of this dissertation introduces a penalized maximum likelihood approach that includes a penalty term directly in the score function prior to maximization of the likelihood, and then implements this method into statistical software.

Generalized estimating equations are also an innovative way to model the within group correlation for longitudinal, clustered, or panel data. Currently, not many diagnostic statistics are available for these models. In the second part of this dissertation, we propose an R^2 and several pseudo-R^2 measures that help researchers with variable selection and provide a goodness of fit measure for the selected model. These calculations are also made accessible to researchers in statistical software.

Generalized estimating equations are an extension to the generalized linear model specifically designed to address the within group correlation. To model the within group correlation in generalized estimating equations, the researcher must select the working correlation structure. However, the current quasi-likelihood information criterion for selecting the working correlation structure is not efficient in that it tends to favor the independent structure which assumes there is no within group correlation. In the last part of this dissertation, we propose a modified quasi-likelihood information criterion that outperforms the current quasi-likelihood information criterion in that this criterion favors the correct structure a large majority of the time. The efficiency of the estimates are improved when using the modified quasi-likelihood information criterion.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the public health field, generalized linear models (GLM) and generalized estimating equations (GEE) are widely used for analysis of clinical studies and secondary data analysis. It is important for these researchers to have the necessary tools to analyze and model their data correctly. This dissertation focuses on a penalized maximum likelihood estimation method for generalized linear models, measures of association such as the coefficient of determination and R^2 for generalized estimating equations, and a modified quasi-likelihood information criterion for generalized estimating equations.

Occasionally, problems with convergence of the maximum likelihood arise in generalized linear models (GLMs). Non-convergence of the maximum likelihood estimates can result from reasons such as complete separation in the data, extremely large values that create a difficult situation for convergence, bias, and small sample size. When one or more of these phenomenon occur during model estimation, researchers are limited in the ways to deal with this situation. Researchers are even more limited in software if the response variable is not a binary outcome. Firth's penalized maximum likelihood estimation approach is currently only available for binary response models in the most widely used statistical software programs SAS, R, and Stata.

One of the first steps in estimation of the parameters in a generalized estimating equation model is to specify a working correlation matrix to be used in the estimating

equation before maximization. If this matrix is incorrectly specified, efficiency is lost in the generalized estimating equations estimates. The current criteria for selecting a working correlation matrix is flawed as in it favors a more simple correlation structure such as independent correlation matrix. The choice of the independent correlation structure assumes that the clusters within the data are not correlated. In other words, it assumes that there is no within group correlation, which we know is normally not the case in panel, cluster, or longitudinal data.

Often researchers want a measure to show how much variance is explained in the chosen model. In multiple linear regression, the R^2 measure helps researchers with variable selection and provides a goodness of fit measure for the selected model. Currently this type of measure is not readily available for models with clustered, longitudinal or panel data. Non-linear regression models also have pseudo-R^2 measures that are not available for generalized estimating equations models. One difference between generalized linear models and generalized estimating equations is the availability of the maximum likelihood. For generalized estimating equation models, the maximum likelihood is not available, and the quasi-likelihood is used. For pseudo- R^2 measures that include the maximum likelihood within the calculation, a different approach will have to be used.

In this dissertation work, I will accomplish three tasks. First, I will extend a penalized maximum likelihood estimation method to generalized linear models and implement the penalized maximum likelihood estimation method in Stata, a statistical software. Second, I will propose an R^2 and some pseudo-R^2 measurements for generalized estimating equations and create a post-estimation command available for use

in Stata. Third, I will propose a modified quasi-likelihood information criterion that identifies the true underlying covariance structure better than the currently available quasi-likelihood information criterion.

CHAPTER 2

GENERALIZED LINEAR MODELS

This chapter presents an introduction to generalized linear models with emphasis on model building. Common link functions and variance functions are presented and discussed.

2.1 INTRODUCTION

Generalized Linear Model theory was introduced by Nelder and Wedderburn [1972]. This theory provided a unity for an entire class of regression models. The basis of this unity is a focus on the single-parameter exponential family of probability distributions. Member distributions of the exponential family include the normal, Poisson, binomial, gamma, inverse Gaussian, negative binomial, and geometric distributions. The exponential family notation which includes a location (mean) parameter and a variance which is written as a function of the mean times a scalar parameter allows the specification of models for all exponential family member distributions including those which are continuous, count, binary, discrete, and proportional outcomes.

The standard linear regression model can be derived from several assumptions. The first assumption is that each observation of the response variable originates from the normal distribution: $y_i \sim N(\mu_i, \sigma_i^2)$. The second assumption is that the distributions for all observations have a common variance: $\sigma_i^2 = \sigma^2$ for all i. The third assumption is that there is a direct relationship between the linear predictor and the expected value of the model: $x_i\beta = g(\mu_i)$ where $g()$ is the identity function linking the linear predictor to the

mean, x_i is a vector of covariates for the ith observation, and $E(y) = \mu = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$. The goal of the generalized linear model is to specify the relationship between the response variable and its' predictors. Note that the properties of the estimators do not depend on the assumption of normality.

Generalized linear models are developed by relaxing the assumptions of the standard linear regression model. An initially nonlinear relationship can be restructured into a linear relationship through the linear predictor and the mean. Generalized linear models are defined by the specified distribution (variance function) and the link function. The assumptions of the generalized linear model as stated by Breslow [1996] are that the observations are independent, and that the variance function $v(\mu)$, the scale factor $a(\phi)$, and the link function are correctly specified, the explanatory variables are in the correct form, and the residuals have the correct distribution.

2.2 Model Building

The components of a generalized linear model are similar to the components of the standard linear regression model. The first component needed is the response variable, y, for which the conditional variance follows that of a distribution belonging to the exponential family. The second component needed is a linear systematic component (the linear predictor), $\eta = \boldsymbol{X}\beta$, the product of the parameters β and the design matrix $\boldsymbol{X}$. The third component is the link function that relates the linear predictor to the mean. The fourth component is the variance function $v(\mu)$ defining the variance of the response variable in terms of its mean μ, $V(y) = a(\phi)v(\mu)$, where $a(\phi)$ is the scale factor and the variance is allowed to change with the covariates as a function of the mean.

Generalized linear models are formulated within the framework of the exponential family of distributions written as

$$f_y(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

where θ is the canonical parameter of location and ϕ is the parameter of scale. The canonical parameter relates to the mean and the scalar parameter relates to the variance for the exponential family members.

Since the observations, y_i, are independent, the joint probability density function of the sample of n observations, given the parameters ϕ and θ, is defined by the product of the densities of the individual observations. Combining these densities, the joint probability density function expressed as a function of ϕ and θ given the observations, y_i into what is called the likelihood, L, is written as

$$L(\theta,\phi;y_1,y_2,\dots y_n) = \prod_{i=1}^{n} \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right\}$$

To obtain estimates of ϕ and θ that maximize the likelihood function, it is easier to work with the log likelihood,

$$\mathcal{L}(\theta,\phi;y_1,y_2,\dots y_n) = \sum_{i=1}^{n} \left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right\}$$

since the values that maximize the likelihood are the same values that maximize the log likelihood. The canonical parameter is represented by θ, $b(\theta)$ is the cumulant, ϕ is the dispersion parameter, and $c()$ is the normalization parameter. This notation also provides simple calculations of the first and second derivatives for maximum likelihood estimation so that $E(y) = b'(\theta)$ and $V(y) = b''(\theta)a(\phi)$.

Each distribution that is part of the exponential family has a unique canonical link, cumulant, and expectation of the canonical link. Table 2.1 shows θ, $b(\theta)$, and $b'(\theta)$ for members of the single parameter exponential family.

To obtain maximum likelihood estimates, substitute the link function of the linear predictor for the expected value of the outcome μ. The estimating equation can be written as

$$\left[\frac{\partial L}{\partial \beta}\right] = X^T \left(\frac{y - E(y)}{a(\phi)}\right) \frac{1}{v(E(y))} \frac{\partial g^{-1}(\eta)}{\partial \eta} = [\mathbf{0}_{px1}]$$

Here the linear predictor η is equated to the canonical link θ. Any monotonic link function that maps the linear predictor to the range implied to the variance function can be chosen.

2.3 Link Functions

Each distribution that is a member of the exponential family has compatible link functions meant to be used under different situations. The Gaussian family model assumes a normally distributed response variable and generally uses the identity link. The identity link assumes a continuous response and can take on negative or positive values. The log-normal model is also based on the Gaussian distribution but uses the log link. The log link is used for response data that only takes on positive values on the continuous scale.

The gamma family model is used for modelling outcomes for which the response can take on only values greater than or equal to zero. This model is generally used with continuous response data but can also be used with count data where the count data take the shape of a gamma distribution. The gamma model is compatible with the reciprocal link for modelling the rate or the log link for modelling the log-rate. The gamma model

can also be used with the identity link to model duration data and assumes there is a one-to-one relationship between η and μ.

The inverse Gaussian distribution is most appropriate to use when modeling a nonnegative response that has a high initial peak, quick drop, and long right tail or when modeling discrete data. The log and identity links are commonly used with such outcomes and are similar to the gamma model.

The binomial-logit family consisting of the Bernoulli/binomial distributions are used to model discrete or proportional responses. This family can be used to model number of successes out of a number of trials. The links that are commonly used with this family are logit, probit, log-log, complementary log-log, identity, log, inverse, and log-complement. The logit link is equivalent to logistic regression where log-odds are modeled while the probit link is used to model data in terms of normal-based probabilities. The complementary log-log defines a sigmoid curve where the upper part is more stretched out than the logit or probit, and the log-log defines a sigmoid curve where the lower part is more stretched out than the logit or probit. The log link produces estimates of the log risk ratio, the log-complement estimates log health ratio, and the identity link yields estimates of the risk difference.

The Poisson family is used to model response variables that are counts or rates. The identity link measures the rate difference while the log link is used to measure the difference in the log of the expected incidence rate ratio.

The negative-binomial distribution can also be used to model count outcomes. This model is useful with overdispersed (relative to the Poisson) count data. It can be derived as a Poisson-gamma mixture. The log link here also estimates log incidence rate-ratios

like the Poisson model. The geometric family is the negative-binomial with the scale parameter ϕ equal to 1. The log link for the geometric family also measures incidence rate-ratios.

Table 2.1 Corresponding canonical link, cumulant, and expected value of y

Distribution	θ	$b(\theta)$	$b'(\theta)$
Binomial	$log\left(\frac{\theta}{1-\theta}\right)$	$\log(1+\exp(\theta))$	$\frac{\exp(\theta)}{1+\exp(\theta)}$
Normal	θ	$\frac{\theta^2}{2}$	θ
Poisson	$\log(\theta)$	$\exp(\theta)$	$\exp(\theta)$
Inverse Gaussian	$\frac{1}{2}\theta^2$	$\sqrt{2\theta}$	$\frac{1}{\sqrt{2\theta}}$
Gamma	$\frac{1}{\theta}$	$-\log\left(\frac{1}{\theta}\right)$	$\frac{1}{\theta}$
Negative Binomial	$\log\left(\frac{\alpha\theta}{1+\alpha\theta}\right)$	$\frac{\log(1-\exp(\theta))}{\alpha}$	$\frac{1}{\alpha}\left(\frac{\exp(\theta)}{1-\exp(\theta)}\right)$

Table 2.2: Common link and variance function combinations

Density	Link Function	Variance Function
Gaussian	Identity	Gaussian
Bernoulli	Logit	Bernoulli
Bernoulli	Probit	Bernoulli
Poisson	Log	Poisson
Negative Binomial	Log	Poisson
Negative Binomial	Negative Binomial	Negative Binomial
Gamma	Reciprocal	Gamma

Table 2.3: Link functions

Link function	$\eta = g(\mu)$
Identity	μ
Logit	$\log\left(\frac{\mu}{1-\mu}\right)$
Log	$\log(\mu)$
Negative Binomial	$\log\left(\frac{\alpha\mu}{1+\alpha\mu}\right)$
Log-complement	$\log(1-\mu)$
Log-log	$-\log(-\log(\mu))$
Probit	$\Phi^{-1}(\mu)$
Reciprocal	$\frac{1}{\mu}$

Table 2.4: Variance functions

Variance Function	$v(\mu)$
Gaussian	1
Bernoulli	$\mu(1-\mu)$
Binomial(k)	$\mu\left(1-\frac{\mu}{k}\right)$
Poisson	μ
Gamma	μ^2
Inverse Gaussian	μ^3
Negative Binomial	$\mu+\alpha\mu$
Power(k)	μ^k

CHAPTER 3

PENALIZED MAXIMUM LIKELIHOOD APPROACH FOR GENERALIZED LINEAR MODELS

This chapter discusses a penalized maximum likelihood method for generalized linear models. The derivation is described and Stata software and examples are displayed.

3.1 INTRODUCTION

This section focuses on the development of a method and its implementation into statistical software Stata. A new Stata command for estimating generalized linear models via penalized maximum likelihood is presented. In the past, only a subset of such models have been available to Stata users through the user-written `firthlogit` (Coveney [2008]) command for binomial models (using on the logit link function). The new `firthglm` command estimates any generalized linear model supported by the `glm` command using penalized log-likelihood.

3.2 CURRENT ISSUES

Firth's penalized maximum likelihood (Firth [1993]) approach was originally developed to reduce the bias of maximum likelihood estimates. The asymptotic bias of the maximum likelihood estimate $\hat{\theta}$ can be written as

$$b(\theta) = \frac{b_1(\theta)}{n} + \frac{b_2(\theta)}{n^2} + \cdots$$

Previously two approaches were used to correct for this bias. The Jackknife method which requires no theoretical calculation but loses precision in the estimate and

the substitution method. The substitution method substitutes $\hat{\theta}$ for the unknown θ in $b(\theta)$ and gives the second order efficient, bias-corrected estimate as $\hat{\theta}_{BC} = \hat{\theta} - \frac{b_1(\hat{\theta})}{n}$. The jackknife and $\hat{\theta}_{BC}$ are bias-reducing only in an asymptotic sense when $\hat{\theta}$ is infinite. Both of these methods use a corrective approach rather than a preventive approach.

This bias arises from a combination of the unbiasedness of the score function at the true value of θ and the curved nature of the score function. To remove bias from the maximum likelihood estimator, Firth's method adds a bias correcting term to the score function. For generalized linear models (GLMs), our target is the canonical parameter of an exponential family, and in this case, the bias term is simply the Jeffreys invariant prior. Jeffreys prior removes the bias, and the end result is a penalized log-(pseudo) likelihood function.

Firth showed that for a random sample from a normal distribution, the bias-reducing penalty function produces an exactly unbiased estimate for θ for sample sizes larger than three. For logisitic regression, the maximum likelihood estimate of β is found to be biased away from the point $\beta = 0$ which requires bias correction with some degree of shrinkage of β towards this point. When the target parameter is the canonical parameter of an exponential family, the estimate is second-order efficient, which means Jeffreys prior is sufficient in removing the bias from the maximum likelihood estimate.

In 2002, Heinze and Schemper claimed that this method developed by Firth was also useful in solving the problem of separation (Heinze and Schemper [2002]). With regard to the relationship of a covariate to an outcome variable in a data set, there are three configurations of n observations we can observe: complete separation, quasi-complete separation, and overlap. In complete separation, the outcome variable separates

one or more predictor variables completely. For example, consider a binary outcome variable and a continuous predictor. If all outcomes with value 1 have corresponding predictor values less than 4 while all outcomes with value 0 have corresponding predictor values greater than 6, we have complete separation. In this situation, the maximum likelihood estimate of the regression parameter on the predictor variable does not exist or tends to infinity.

Quasi-complete separation exists when the outcome variable separates one or more predictor variables to a certain point. Consider the previous example of complete separation where the corresponding predictor values are separated similarly, but now the predictor values for both outcomes (0 and 1) include the value 5. Here the only probability to estimate is the probability the predictor value equals 5. All the other predictor values are separated by the outcome variable. In this situation, the maximum likelihood estimate of the regression parameter on the predictor variable also does not exist.

Overlap exists when there is no separation in the data, and this situation is generally not a problem for parameter estimation. Since there is no separation, the maximum likelihood estimate exists.

When separation arises, there are a number of options to consider. One can omit the variable from the model, allowing estimates to be obtained for the other parameters. By omitting the variable, the information about the effect of the possible risk factor is lost. One can manipulate the data by using an ad hoc adjustment such as changing cell frequencies or forcing the largest or smallest observation to have the opposite effect. This option could be misleading and also has undesirable properties. One option in Stata is to

use exact regression (such as `exlogit` or `expoisson`) which replaces the maximum likelihood estimate by a median unbiased estimate where the estimate of a parameter and inference are based on the exact null distribution of the sufficient statistic conditional on the observed values of the other sufficient statistics. This method is useful with one variable but cannot be used when two or more variables lead to degenerate distributions of all sufficient statistics.

This penalized maximum likelihood method is currently available for logistic regression, but these situations are not limited to binary outcomes and can occur for any specified generalized linear model. In this manuscript we introduce Firth's penalized maximum likelihood estimation in Section 3.3. In Section 3.4, the Stata syntax is shown for the new `firthglm` command, and the examples are contained in Section 3.5.

3.3 METHODS

A bias reduction of maximum likelihood estimates in generalized linear models was introduced by Firth [1993]. Instead of taking a corrective approach by estimating the maximum likelihood and then adding a penalty term, Firth modifies the score function and then produces the maximum likelihood estimate. This is particularly useful when the maximum likelihood estimate does not exist or is infinite.

The penalized likelihood equation written within the framework of the exponential family of distributions is defined as

$$L(\theta, \phi; y_1, y_2, \dots, y_n) = \prod_{i=1}^{n} \exp\left\{\frac{y_i\theta_i - b\theta_i}{a(\phi)} + c(y_i, \phi)\right\} |i(\theta)|^{1/2}$$

where $y_1, y_2, \ldots, y_n$ are the sample of independent observations, θ_i is the canonical location parameter for the ith observation, ϕ is the scale parameter, and $|i(\theta)|^{1/2}$ is the Jeffreys [1946] invariant prior.

We can take the log of equation 2 to obtain the penalized log-likelihood

$$\mathcal{L}(\theta, \phi; y_1, y_2, \ldots, y_n) = \sum_{i=1}^{n} \left\{ \frac{y_i \theta_i - b\theta_i}{a(\phi)} + c(y_i, \phi) \right\} + \frac{1}{2} \log |i(\theta)|$$

since the values that maximize the likelihood also maximize the log-likelihood. The estimates can then be computed using Stata's ml optimization commands.

Because the likelihood is written in exponential family notation (Hardin and Hilbe [2011]), we can specify penalized models for not only binary outcomes, but also count, proportional, discrete, and continuous outcomes.

Firth notes that bias reduction can be affected by the number of factors, especially the skewness of the maximum likelihood estimate. In this case, one might sacrifice precision in the estimates. However, in his paper, Firth states that when employing logistic regression, the maximum likelihood estimate is unbiased and reduces the variance of the parameter estimates. We are to expect smaller standard errors when using `firthglm`. Confidence intervals will be affected since in reality, the estimate's lower bound should be negative infinity when the maximum likelihood estimate tends to negative infinity, and the upper bound should be positive infinity when the maximum likelihood estimate tends to positive infinity.

3.4 Stata Syntax

Software accompanying this section includes the command files as well as supporting files for prediction and help. In all of the following syntax diagrams, unspecified options

include the usual collection of maximization and display options available to all estimation commands.

Equivalent in syntax to the to the `glm` command, the basic syntax for the penalized generalized linear model is given by

`firthglm` [*depvar* [*indepvars*]] [*if*] [*weight*] [, *]

It should be noted, that the penalized log-likelihood maximization method is implemented using Stata's `ml` commands specifying the `d0` optimization method. As such, the `firthglm` command does not support some of the `vce()` options that are available in the `glm` command specifically, the `firthglm` command does not support `opg`, `unbiased`, `robust`, or `cluster`. Similarly, the `firthglm` command does not support the `pweight` option.

Help files are included for the estimation and post-estimation specifications of these models. The help files include example specifications.

3.5 Real Data Analysis

All examples were analyzed using the 12.1 version of Stata (Stata Corp, College Station, TX). We show two examples in this section. The first example demonstrates bias reduction when using a Poisson regression. The second example shows how the penalized maximum likelihood method is useful when there is separation in the data and the maximum likelihood does not converge. This example uses logistic regression, and we also compare the `firthglm` method with bootstrapping.

The first example uses a ship accident dataset from McCullagh and Nelder [1989], listed on page 205 of the text. This dataset includes the number of reported damage incidents, `accident`, the collective months of service by ship type, `service`,

the period of operation, `op_00_00`, the construction year, `co_00_00`, and the type of ship, `ship`. The exposure in the model is the collective months of service. To better define some variables, we use the indicator variables `op_00_00` to show the starting and ending years of operation and the indicator variables `co_00_00` to show the starting and ending years of construction. For example, `op_70_74` shows whether the ship was in operation from 1970 to 1974, and `co_60_64` shows whether the ship was in construction from 1960 to 1964. There is a total of 34 full observations in this dataset.

We run a Poisson model of `accident` on `op_75_79`, `co_65_69`, `co_70_74`, `co_75_79`, and `ship` with a log link. To obtain risk ratios, we use the `eform` option.

The results of the model are in Table 3.1. The results show that whether the ship was in service between 1975 and 1979 is a significant predictor of number of accidents. Also, whether the ship was constructed between 1965 and 1969 or between 1970 and 1974 are also significant predictors of number of accidents. The fourth indicator variable `co_75_79` is not significant with a p-value of 0.052. Ship types 2 and 3 are significantly different from ship type 1 while ship type 4 and 5 are not significantly different from ship type 1.

To examine bias reduction, we can run `firthglm` using the same model options. From the output in Table 3.2, we can see that the penalized log likelihood of -51.4 is a good bit smaller than the log likelihood of -68.3. The Aikaike Information Criteria (AIC) for the penalized maximum likelihood method is smaller than the non-penalized method (3.55 compared to 4.55). The deviance for the penalized method is slightly larger than the deviance for the non-penalized method (38.8 compared to 38.7). We have similar results

for all variables in the model except `co_75_79`. In the non-penalized GLM model, this variable was not significant, but is now significant with a p-value of 0.047.

The command `firthglm` can be applied to any generalized linear model and canonical link supported by Stata's `glm` command using penalized log-likelihood. We illustrate another model using data provided by Dr. José Villa at the USDA-ARS Honey Bee Breeding, Genetics and Physiology Laboratory (Deroche et al. [2011]) where convergence is not achieved in a binary response regression model with a log-log link. Convergence is not achieved due to the issue of complete separation in the data.

These data contain the levels of mite infestation (`mites`) in a longitudinal study on bee colonies from nine different genetic `origins`. Measurements of mite infestation were recorded every `season` over a seven year period as well as the `status` (dead or alive) of each colony. Here `origin 9` and `season 4` are the referent groups.

Two genetic origins did not experience any deaths due to the level of mite infestation (`origin 1` and `origin 4`). This is an example of separation. The effect of this separation on the model can be seen when using Stata's `glm` command. We illustrate a binomial model of `status` on `mites`, `origin1-origin8`, and `season1-season3` with a log-log link. We can see in Table 3.3 that the estimates for `origin1` and `origin4` are -2.45 and -2.17 with associated standard error estimates which are approximately 104 and 88. The maximum likelihood estimates of these terms tend toward negative infinity which implies the odds ratio is tending toward zero. From this model, we can see that the only significant predictor of status is mites.

This inability to reach convergence is fixed by estimating a penalized maximum likelihood model with the `firthglm` command and `family(binomial)` with

`link(loglog)`. The parameter estimates without convergence problems have estimates and standard errors slightly smaller than those given by `glm`, but the significant predictors of status have not changed. These results can be seen in Table 3.4.

Another alternative to the penalized maximum likelihood method is to use `vce(bootstrap)` within the `glm` model. The results of this method can be seen in Table 3.5. The standard errors obtained are larger than those obtained through `firthglm` but still much smaller than those from `glm`. The parameter estimates for `origin1` and `origin4` are still large, but now they are significant with p-values of 0.011 and 0.014 respectively. Bootstrapping is efficient in giving smaller standard errors, but in this case, gives misleading results. The option `vce(bootstrap)` is also available within the `firthglm` command.

To explore other options for dealing with separation, we ran a regular logistic model using `family(binomial)` and `link(logit)`. We tried to compare this to using `exlogistic`, but this method failed to estimate the model and combat the issue of separation.

3.5 Conclusions and Discussions

This penalized maximum likelihood method for generalized linear models has been proven to be useful in bias reduction and solving the problem of separation in data. In the past, this method was only available in software for a binary outcome using logistic models. The `firthglm` command broadens this penalized maximum likelihood method to all generalized linear models regardless of the structure of the response variable or the canonical link used in modeling.

Table 3.1 GLM Poisson model with log link

```
gen double exposure = ln(service)
6 missing values generated)
glm accident op_75_79 co_65_69 co_70_74 co_75_79 i.ship,
family(poiss) link(log) offset(exposure) eform nolog
```

Generalized linear models	No. of obs = 34
Optimization : ML	Residual df = 25
	Scale parameter = 1
Deviance = 38.69505154	(1/df) Deviance = 1.547802
Pearson = 42.27525312	(1/df) Pearson = 1.69101
Variance function: V(u) = u	[Poisson]
Link function : g(u) = ln(u)	[Log]
	AIC = 4.545928
Log likelihood = -68.28077143	BIC = -49.46396

accident	IRR	OIM Std. Err.	z	P>|z|	[95%Conf.	Interval]
op_75_79	1.468831	.1737218	3.25	0.001	1.164926	1.852019
co_65_69	2.008002	.3004803	4.66	0.000	1.497577	2.692398
co_70_74	2.26693	.384865	4.82	0.000	1.625274	3.161912
co_75_79	1.573695	.3669393	1.94	0.052	.9964273	2.485397
ship						
2	.5808026	.1031447	-3.06	0.002	.4100754	. 8226088
3	.502881	.1654716	-2.09	0.037	.2638638	.9584087
4	.926852	.2693234	-0.26	0.794	.5244081	1.638141
5	1.384833	.3266535	1.38	0.168	.8722007	2.198762
_cons	.0016518	.0003592	-29.46	0.000	.0010786	.0025295
exposure	1	(offset)				

Table 3.2 Penalized GLM Poisson model with log link

```
firthglm accident op_75_79 co_65_69 co_70_74 co_75_79
i.ship, family(poiss) link(log) offset(exposure) eform
nolog
```

Generalized linear models	No. of obs = 34
Optimization : ML	Residual df = 25
	Scale parameter = 1
Deviance = 38.78425338	(1/df) Deviance = 1.55137
Pearson = 41.00930919	(1/df) Pearson = 1.640372
Variance function: V(u) = u	[Poisson]
Link function : g(u) = ln(u)	[Log]
	AIC = 3.554387
Log likelihood = -51.42457974	BIC = -49.37476

accident	IRR	OIM Std. Err.	z	P>|z|	[95%Conf.	Interval]
op_75_79	1.467798	.1733692	3.25	0.001	1.164465	1.850146
co_65_69	2.003015	.2988503	4.66	0.000	1.49515	2.683389
co_70_74	2.262953	.3833049	4.82	0.000	1.623666	3.153946
co_75_79	1.584269	.3677294	1.98	0.047	1.005204	2.496916
ship						
2	.5757154	.1017826	-3.12	0.002	.4071188	.8141315
3	.5179367	.1675552	-2.03	0.042	.2747316	.9764384
4	.9416689	.270467	-0.21	0.834	.5363093	1.653412
5	1.389521	.3255163	1.40	0.160	.8779272	2.199237
_cons	.0016818	.0003642	-29.46	0.000	.0011002	.0025708
exposure	1	(offset)				

Table 3.3 GLM binomial model with log-log link

```
glm status mites b4.seasons b9.origins, fam(binomial)
link(loglog) nolog
```

Generalized linear models						No. of obs = 331
Optimization : ML						Residual df = 318
						Scale parameter = 1
Deviance = 206.9444987						(1/df) Deviance = .6507689
Pearson = 315.5780133						(1/df) Pearson = .9923837
Variance function: V(u) = u*(1-u)						[Binomial]
Link function : g(u) = -ln(-ln(u))						[Log-Log]
						AIC = .7037598
Log likelihood = -103.4722493						BIC = -1638.129
status	Coef.	OIM Std. Err.	z	P>|z|	[95%Conf.	Interval]
mites	.5884078	.172608	3.41	0.001	.2501024	.9267133
seasons						
1	.0925229	.2275317	0.41	0.684	-.353431	.5384769
2	-.2882072	.2621791	-1.10	0.272	-.8020689	.2256544
3	-.1231914	.2426097	-0.51	0.612	-.5986977	.352315
origins						
1	-2.446689	103.9552	-0.02	0.981	-206.1952	201.3018
2	.3107892	.3712186	0.84	0.402	-.4167859	1.038364
3	-.0229646	.3548255	-0.06	0.948	-.7184097	.6724805
4	-2.166469	88.2183	-0.02	0.980	-175.0712	170.7382
5	.6283637	.5669993	1.11	0.268	-.4829344	1.739662
6	.9753662	.6890866	1.42	0.157	-.3752188	2.325951
7	-.002432	.3758121	-0.01	0.995	-.7390102	.7341462
8	.276403	.6358697	0.43	0.664	-.9698788	1.522685
_cons	-1.096796	.3952132	-2.78	0.006	-1.8714	-.3221925

Table 3.4 Penalized GLM binomial model with log-log link

```
firthglm status mites b4.seasons b9.origins, fam(binomial)
link(loglog) nolog
```

Penalized generalized linear models						No. of obs = 331
Optimization : PML						Residual df = 318
						Scale parameter = 1
Deviance = 206.9444987						(1/df) Deviance = .6507689
Pearson = 315.5780133						(1/df) Pearson = .9923837
Variance function: V(u) = u*(1-u)						[Binomial]
Link function : g(u) = -ln(-ln(u))						[Log-Log]
						AIC = .7037598
Log likelihood = -103.4722493						BIC = -1638.129
status	Coef.	OIM Std. Err.	z	P>|z|	[95%Conf.	Interval]
mites	.5547919	.1654294	3.35	0.001	.2305562	.8790277
seasons						
1	.0737509	.219216	0.34	0.737	-.3559045	.5034062
2	-.2744021	.2493442	-1.10	0.271	-.7631077	.2143036
3	-.1188671	.2327333	-0.51	0.610	-.5750159	.3372817
origins						
1	-.6938187	.5873771	-1.18	0.238	-1.845057	.4574192
2	.2788829	.3545535	0.79	0.432	-.4160292	.973795
3	-.0426837	.3372096	-0.13	0.899	-.7036024	.6182349
4	-.6047809	.5269037	-1.15	0.251	-1.637493	.4279315
5	.5826037	.5338476	1.09	0.275	-.4637183	1.628926
6	.8637302	.6364989	1.36	0.175	-.3837847	2.111245
7	-.0243892	.3586332	-0.07	0.946	-.7272974	.678519
8	.2827737	.5829796	0.49	0.628	-.8598453	1.425393
_cons	-1.042334	.3732566	-2.79	0.005	-1.773903	-.3107642

Table 3.5 Bootstrap GLM binomial model with log-log link

```
glm status mites b4.seasons b9.origins, fam(binomial)
link(loglog) vce(bootstrap) nolog
(running glm on estimation sample)
Bootstrap replications (50)
```

Generalized linear models					No. of obs = 331	
Optimization : ML					Residual df = 318	
					Scale parameter = 1	
Deviance = 206.9444987					(1/df) Deviance = .6507689	
Pearson = 315.5780133					(1/df) Pearson = .9923837	
Variance function: V(u) = u*(1-u)					[Binomial]	
Link function : g(u) = -ln(-ln(u))					[Log-Log]	
					AIC = .7037598	
Log likelihood = -103.4722493					BIC = -1638.129	
status	Observed Coef.	Bootstrap Std. Err.	z	P>|z|	Normal [95%Conf.	-based Interval]
mites	.5884078	.1770143	3.32	0.001	.2414662	.9353494
seasons						
1	.0925229	.2458477	0.38	0.707	-.3893298	.5743756
2	-.2882072	2.07542	-0.14	0.890	-4.355955	3.779541
3	-.1231914	.1877112	-0.66	0.512	-.4910986	.2447159
origins						
1	-2.446689	.9572005	-2.56	0.011	-4.322767	-.5706107
2	.3107892	.8979404	0.35	0.729	-1.449142	2.07072
3	-.0229646	.8877846	-0.03	0.979	-1.76299	1.717061
4	-2.166469	.8853953	-2.45	0.014	-3.901812	-.4311264
5	.6283637	.9077334	0.69	0.489	-1.150761	2.407489
6	.9753662	1.708581	0.57	0.568	-2.373391	4.324123
7	-.002432	.9998806	-0.00	0.998	-1.962162	1.957298
8	.276403	2.27747	0.12	0.903	-4.187357	4.740163
_cons	-1.096796	.8921839	-1.23	0.219	-2.845444	.6518522

CHAPTER 4

GENERALIZED ESTIMATING EQUATIONS

This chapter introduces generalized estimating equations and its link to generalized linear models. The extension of the working correlation matrix is discussed and the quasi-likelihood is introduced.

4.1 INTRODUCTION

One vital assumptions of generalized linear models is independence of the observations. This assumption is violated when the data may be grouped in some manner such as patients from the same hospital or when multiple observations are made on the same subject over time. There are multiple ways to address clustered, panel or longitudinal data, and each method has its own advantages and limitations. The naïve way to address this type of data is to ignore the panel structure of the data yielding a pooled (independence) estimator. This method results in a consistent estimator but one that is not efficient leading to (possibly) unreliable standard error estimates. Another way to address panel data is to include an effect for each panel in the estimating equation. This method allows fixed or random effects and conditional or unconditional effects. When the data include a finite number of panels in a population where each panel is represented in the sample, it is more reasonable to consider an unconditional fixed effects estimator. However, if there exists an infinite number of panels in the population, it is more reasonable to consider a conditional fixed effects estimator. Here one can include a fixed incremental change per group. A conditional fixed effects estimator is one for which the

model conditions out the fixed effects from the estimation leading to a log-likelihood which does not depend on the fixed effects. Here one can make inferences about population averages and where the mean response is conditional only on covariates. Also known as subject-specific models, the random effects model allows regression coefficients (intercept and slope) to vary from person to person according to a random effects distribution. The transitional Markov model represents the probability distribution at each time point as conditional on the previous time point and is usually estimated using Gibbs sampling. An increasingly popular alternative introduced by Liang and Zeger [1986] is known as Generalized Estimating Equations.

4.2 Extension of Generalized Linear Models

In their manuscript, Liang and Zeger [1986] provide an extension to generalized linear models which they refer to as population-averaged generalized estimating equations. This method induces an interpretation of the coefficients as population averages and introduces the dependency (non-independence) of the observations directly into the estimating equation of the pooled estimator. The estimating equation in a generalized linear model which assumes independence can be written as

$$\left[\frac{\partial L}{\partial \beta}\right] = \sum_{i=1}^{n} {X_i}^T D \left(\frac{\partial g^{-1}(\eta_i)}{\partial \eta_i}\right) (v(E(y_i)))^{-1} \left(\frac{y_i - E(y_i)}{a(\phi)}\right)$$

$$= \sum_{i=1}^{n} {X_i}^T D \left(\frac{\partial g^{-1}(\eta_i)}{\partial \eta_i}\right) \left(v\left(E(y_i)\right)^{-\frac{1}{2}}\right)^T I(n_i) v\left(E(y_i)\right)^{-\frac{1}{2}} \left(\frac{y_i - E(y_i)}{a(\phi)}\right)$$

where $I(n_i)$ is the identity matrix representing the within-group correlation (assumed to be independent). One can parameterize an alternative correlation matrix to model the within-group correlation structure by replacing the identity matrix with a working correlation $R(\alpha)$ so that the estimating equation is now written as

$$\left[\frac{\partial L}{\partial \beta}\right] = \sum_{i=1}^{n} {X_i}^T D \left(\frac{\partial g^{-1}(\eta_i)}{\partial \eta_i}\right)\left(v\left(E(y_i)\right)^{-\frac{1}{2}}\right)^T R(\alpha) v\left(E(y_i)\right)^{-\frac{1}{2}} \left(\frac{y_i - E(y_i)}{a(\phi)}\right)$$

where α is a vector of parameters through which the matrix R is structurally constrained to represent the working or within-panel correlation. Here it is shown that the focus of the generalized estimating equation is on the marginal distribution and the estimator which sums the panel-level contribution to the estimating equations after accounting for the within-panel correlation. Thus, the estimating equation for the regression parameters β are formed for the average (sum) of the panels.

4.3 Working Correlation Structure

The researcher or analyst is charged with making the correct structural choice of the working correlation matrix for models estimated using generalized estimating equations. There are several correlation structure choices available in software. The most commonly used correlation structures are the independent, exchangeable, autoregressive(1), and unstructured. These structures are illustrated in Table 4.1.

Consider independent observations from n individuals. For each individual i, a response y_{it} and a $p \times 1$ covariate vector $x_{it} = (x_{it1}, x_{it2}, \dots, x_{itp})^T$ are gathered at times $t = 1, 2, \dots, m_i$. Let $Y_i = (y_{i1}, y_{i2}, \dots, y_{im_i})^T$ be the $m_i \times 1$ vector of responses for the i^{th} individual and $X_i = (x_{i1}^T, x_{i2}^T, \dots, x_{im_i}^T)^T$ be the $m_i \times p$ corresponding covariate matrix. The working correlation structure is chosen for the full model prior to the model selection of the number of covariates.

4.4 Quasi-Likelihood

The quasi-likelihood is constructed for the mean parameter $\mu = E(y)$ and the dispersion parameter ϕ where y is the scalar response variable rather than by specifying a

probability distribution. McCullagh and Nelder (1989) give the log quasi-likelihood based on the model specification $E(y) = \mu$ and $var(y) = \phi v(\mu)$ as

$$Q(\mu, \phi; y) = \int_{y}^{\mu} \frac{y - t}{\phi v(t)} dt$$

The quasi-likelihood can be written as a function of the regression coefficients β, for example $Q(\beta, \phi; (y, x)) = Q(g^{-1}(x\beta), \phi; y)$. If it is assumed that the working independence model $R = I$ is selected, then the paired observations (Y_{ij}, X_{ij}) in the data D are independent. Then the quasi-likelihood based on D is

$$Q(\beta, \phi; I, D) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} Q\left(\beta, \phi; \left(Y_{ij}, X_{ij}\right)\right).$$

Then the quasi-deviance can be defined as

$$Deviance = 2 \int_{\hat{\mu}}^{y} \frac{y - \mu}{v(\mu)} d\mu.$$

The quasi-likelihood can also be written in terms of the quasi-deviance as

$$Q(M) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} Q\left(\beta, \phi; \left(Y_{ij}, X_{ij}\right)\right) = -\frac{Dev(M)}{2}$$

where $Dev(M) = 2 \int_{\hat{\mu}}^{y} \frac{y-\mu}{v(\mu)} d\mu$.

Table 4.1 Working Correlation Structures

Working correlation structure	Example 3 x 3 matrix
Independent: $Corr(y_{ij}, y_{ik}) = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Exchangeable: $Corr(y_{ij}, y_{ik}) = \begin{cases} 1 & j = k \\ \alpha & j \neq k \end{cases}$	$\begin{bmatrix} 1 & \alpha & \alpha \\ \alpha & 1 & \alpha \\ \alpha & \alpha & 1 \end{bmatrix}$
AR-1: $Corr(y_{ij}, y_{ik}) = \alpha^{\lvert j-k \rvert}$	$\begin{bmatrix} 1 & \alpha & \alpha^2 \\ \alpha & 1 & \alpha \\ \alpha^2 & \alpha & 1 \end{bmatrix}$
Unstructured: $Corr(y_{ij}, y_{ik}) = \begin{cases} 1 & j = k \\ \alpha_{min(j,k),\max(j,k)} & j \neq k \end{cases}$	$\begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} \\ \alpha_{12} & 1 & \alpha_{23} \\ \alpha_{13} & \alpha_{23} & 1 \end{bmatrix}$

CHAPTER 5

R^2 AND PSEUDO-R^2 FOR GENERALIZED ESTIMATING EQUATIONS

This chapter introduces the R^2 and pseudo-R^2 statistics currently available for generalized linear models. The likelihood is replaced with the quasi-likelihood and a post estimation command for Stata is introduced.

5.1 INTRODUCTION

For generalized linear models, there are many model measures (diagnostic criteria) that are not available for generalized estimating equation models. One of these measures is the coefficient of determination R^2. Natarajan, et.al [2007] proposed a measure of partial association for GEE and a coefficient of determination to measure the strength of association between the outcome variable and the fitted values based on the estimated coefficients. The psuedo-R^2 statistics to be explored are Efron's psuedo-R^2 (for continuous and binary outcomes) [1978], McFadden's likelihood ratio index (for any outcome) [1974], Ben-Akiva and Lerman's adjusted likelihood ratio index (for any outcome) [1985], Cox and Snell [1968] and Maddala [1983] combined transformation of likelihood ratio (for any outcome), and Cragg and Uhler's normed measure (for any outcome) [1970]. Where the likelihood is used for a calculation, GEE's quasi-likelihood calculation will be inserted.

5.2 CURRENT AVAILABILITY OF R^2 IN GENERALIZED LINEAR MODELS

The most commonly used R^2 is the one that involves the calculation of residual sum of squares (RSS) and total sum of squares (TSS). This R^2 can be interpreted as the percent

variance explained and is written as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$

It can also be interpreted as the squared correlation and the ratio of variances. The numerator is in terms of the differences of the observed and fitted values, while the denominator is in terms of the differences of the observed and mean values.

The above measure is used for linear regression. When these types of statistics are applied to generalized linear models, they are called pseudo-R^2 statistics. Other measures are available for models other than linear regression. The ones discussed herein are Efron's pseudo-R^2, McFadden's likelihood ratio index, Ben-Akiva and Lerman adjusted likelihood ratio index, Cragg and Uhler normed measure, and the Cox-Snell or transformation of likelihood ratio.

Efron [1978] defines a measure as an extension to the regression model's "percent variance explained" interpretation and is given by

$$R^2_{Effron} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\hat{\mu}$ is the model predicted probabilities. This measure was originally directed at binary outcome models, but can also be used for continuous models by replacing the $\hat{\mu}_i$ with $\hat{y}_i$.

McFadden [1974] defines a measure, sometimes called the likelihood-ratio index, as another extension to the "percent variance explained interpretation" given by

$$R^2_{\text{McFadden}} = 1 - \frac{\mathcal{L}(M_\beta)}{\mathcal{L}(M_\alpha)}$$

where $\mathcal{L}$ is the log-likelihood, M_α is the model with only an intercept, and M_β is the model with intercept and covariates.

Ben-Akiva and Lerman [1985] extended McFadden's pseudo-R^2 to include an adjustment for the number of parameters in the model. This adjustment is similar to the adjusted R^2 in linear regression and is given by the formula

$$R^2_{\text{Ben-Akiva\&Lerman}} = 1 - \frac{\mathcal{L}(M_\beta) - p}{\mathcal{L}(M_\alpha)}$$

where p is the number of parameters in the model. The intention behind the adjustment is to decrease the likelihood so that non-significant variables included in the model do not cause a significant increase in the criterion measure.

By combining the work of Cox and Snell [1968] and Maddala [1983], a maximum likelihood pseudo-R^2 is described in the formula

$$R^2_{\text{ML}} = 1 - \left\{\frac{L(M_\alpha)}{L(M_\beta)}\right\}^{\frac{2}{n}} = 1 - \exp\left(-\frac{G^2}{n}\right)$$

where $G^2 = -2\ln\left\{\frac{L(M_\alpha)}{L(M_\beta)}\right\}$. This measure is an extension to the transformation of the likelihood ratio.

The last measure examined here is the Cragg and Uhler [1970] normed measure. Cragg and Uhler introduced a transformation of the likelihood ratio pseudo-R^2 because the R^2_{ML} does not approach 1 as the fit of the two comparison models converge. The normed version of the R^2_{ML} is given by

$$R^2_{\text{Cragg\&Uhler}} = \frac{R^2_{\text{ML}}}{\max R^2_{\text{ML}}} = \frac{1 - \left\{\frac{L(M_\alpha)}{L(M_\beta)}\right\}^{\frac{2}{n}}}{1 - L(M_\alpha)^{2/n}}.$$

These pseudo-R^2 measures are available in a user-written Stata command named `fitstat`. The command `fitstat` is a post estimation command that calculates the McFadden, Ben-Akiva and Lerman (adjusted McFadden), Cox-Snell, Cragg-Uhler, and Efron pseudo-R^2 measures after computing the `clogit`, `cloglog`, `intreg`, `logisitic`, `logit`, `mlogit`, `nbreg`, `ocratio`, `ologit`, `oprobit`, `poisson`, `probit`, `regress`, `tnbreg`, `tpoisson`, `zinb`, `zip`, or `ztb` regression models. The `fitstat` command was developed by Long and Freese [2014].

5.3 Generalized Linear Model R^2 Extension to Generalized Estimating Equations

Natarajan, et.al [2007] proposed a measure of partial association for GEE and a coefficient of determination to measure the strength of association between the outcome variable and all of the coefficients. Using ordinary least squares (OLS) to estimate the regression parameters allows the estimate of the partial correlation coefficient to be a monotone function of the Z-statistic that is used to test whether a single regression coefficient is equal to zero. Natarajan, et. al. propose to use the transformation of the GEE Z-statistic as a measure of partial association. Following this same thought, they propose to use a function of the Wald statistic that tests whether all parameters (except the intercept) are equal to zero to generate the coefficient of determination.

For clustered data, each individual $i(i = 1, \dots, N)$ has an n_ix1 response vector $Y_i = [Y_{i1}, \dots, Y_{in_i}]^T$ and a Kx1 covariate vector $x_{ij} = [x_{ij1}, \dots, x_{ijK}]^T$. To calculate β estimates through a GEE approach, the equation below is solved iteratively.

$$S(\beta; R, D) \equiv \sum_{i=1}^{n} D_i' V_i^{-1} (Y_i - \mu_i) = 0,$$

where $D_i = D_i(\beta) = \frac{\partial \mu_i(\beta)}{\partial \beta'}$ and V_i is a working covariance matrix of Y_i. The working correlation matrix $R = R(\alpha)$ can be expressed in terms of $V_i = A_i^{1/2} R(\alpha) A_i^{1/2}$, where A_i is a diagonal matrix with elements $var(Y_{ij}) = \phi v(\mu_{ij})$, which is specified as a function of the mean $\mu_{ij} = E(Y_{ij}|x_{ij}) = g(x_{ij}'\beta)$. The parameter α represents a vector of some unknown parameters involved in estimating the working correlation structure.

The GEE Wald Z-statistic to test $H_0: \beta_K = 0$ is calculated using the estimate of the $\hat{\beta}_K$ and dividing it by the model-based standard error estimate of $\hat{\beta}_K$; $\sqrt{\widehat{Var}(\hat{\beta}_K)}$. Since the Wald statistic has been shown to have poor properties as $|\hat{\beta}_K|$ gets large, they propose to use the Wald statistic with the variance of $\hat{\beta}_K$ estimated under the null $H_0: \beta_K = 0$ by replacing $\widehat{Var}(\hat{\beta}_K)$ with the GEE robust variance estimate $\widetilde{Var}(\hat{\beta}_K)$. This gives a measure of partial association which under the null is approximately chi-square with 1 degree of freedom.

$$\widetilde{Z_K} = \frac{\hat{\beta}_K}{\sqrt{\widetilde{Var}(\hat{\beta}_K)}}$$

We then can define the measure of partial association between Y_{ij} and x_{ijK} to be

$$\widetilde{\rho_K} = \frac{\tilde{z}_K/\sqrt{N}}{\sqrt{1 + \tilde{z}_K^2/N}}$$

which ranges from -1 to 1.

For the GEE model, $E(Y_{ij}|x_{ij}) = g(x_{ij}'\beta_K)$, the Wald test to test $H_0: \beta_1 = \cdots = \beta_K = 0$ can be used to form an R^2 statistic. Following the above derivation of the measure of partial association, it is proposed that the coefficient of determination be written as

$$\widetilde{R^2} = \frac{\tilde{Q}/N}{1 + \tilde{Q}/N}$$

where

$$\tilde{Q} = [C\hat{\beta}]'[C\,\widetilde{Var}(\hat{\beta}_K)C']^{-1}[C\hat{\beta}],$$

is the Wald statistic with the GEE robust covariance matrix estimated under the null. This statistic will range from 0 to 1 but does not guarantee that a model with more covariates would have a larger $\widetilde{R^2}$. This is shown by Natarajan et al. when an additional covariate adds very little information.

In order to generalize the pseudo-R^2 measures discussed in section 5.3 to generalized estimating equations, the maximum likelihood calculations must be replaced with the quasi-likelihood calculations. The extended measures for R^2_{McFadden} is now

$$gR^2_{\text{McFadden}} = 1 - \frac{Q(M_\beta)}{Q(M_\alpha)}$$

where $Q(M_\alpha)$ is the quasi-likelihood for the model with only an intercept and $Q(M_\beta)$ is the model with intercept and predictors. The quasi-likelihood can also be written in terms of the quasi-deviance as

$$Q(M) = \sum_{i=1}^{n}\sum_{j=1}^{n_i} Q\left(\beta, \phi; \left(Y_{ij}, X_{ij}\right)\right) = -\frac{Dev(M)}{2}$$

where $Dev(M) = 2\int_{\hat{\mu}}^{y} \frac{y-\mu}{v(\mu)} d\mu$. The extended measure for $R^2_{\text{Ben-Akiva\&Lerman}}$ is then given by

$$gR^2_{\text{Ben-Akiva\&Lerman}} = 1 - \frac{Q(M_\beta) - p}{Q(M_\alpha)}.$$

Similar replacements are made for R^2_{ML} and $R^2_{\text{Cragg\&Uhler}}$ so that the formulas are

$$gR^2_{\text{ML}} = 1 - \left\{\frac{Q(M_\alpha)}{Q(M_\beta)}\right\}^{\frac{2}{n}}$$

and

$$gR^2_{\text{Cragg\&Uhler}} = \frac{gR^2_{\text{ML}}}{\max gR^2_{\text{ML}}} = \frac{1 - \left\{\frac{Q(M_\alpha)}{Q(M_\beta)}\right\}^{\frac{2}{n}}}{1 - Q(M_\alpha)^{2/n}}.$$

In Efron's pseudo-R^2, we replace the single summand for observations with a double summand to account for the panel observations and within a panel. The generalized Efron pseudo-R^2 can be written as

$$gR^2_{\text{Efron}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(y_{ij} - \hat{y}_{ij}\right)^2}{\sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(y_{ij} - \bar{y}\right)^2}$$

These calculations were made available in Stata in a user-written, post-estimation command named `estatg`. This command is available after any GEE model is estimated in Stata. The post-estimation command works for any link and variance function.

5.4 Real Data Analysis

To test the extensions of the pseudo-R^2 measures in GEE models, one can compare the quasi-likelihood pseudo-R^2 measures to the likelihood based pseudo- pseudo-R^2 measures with a sample dataset. We used a dataset on low birthweight from Homer and Lemeshow (2013) in Stata. There are a total of 189 observations in this dataset. This dataset includes an identification code for each mother, an indicator variable of low birth weight (`low`), the age of the mother (`age`), the categorical variable race (`race`), an indicator variable of whether or not the mother smoked during pregnancy (`smoke`), the mother's pre-pregnancy weight (`lwt`), an indicator variable of whether or not the mother had a history of premature labor (`ptl`), a history of hypertension (`ht`), or at the time of

birth had uterine irritability (`ui`). We run a logistic model (link function as binomial and variance function as independent) on `age`, `lwt`, `race`, `smoke`, `ptl`, `ht`, and `ui` on `low`. The `xtgee` model fit is shown in Table 5.1 where it is shown that `lwt`, `race`, `smoke`, and `ht` are significant predictors of `low`. The output from `estatg` is given in Table 5.2 for the GEE R^2 and the five pseudo-R^2.

We can compare this to the output of `fitstat` after running a `logit` model and see that the replacement of the maximum likelihood with the quasi-likelihood works well in this situation. When fitting the same model under GLM and independent GEE, we have the same results. The results of the `logit` model in Table 5.3 show that `lwt`, `race`, `smoke`, and `ht` are significant predictors of `low`. The results of `fitstat` in Table 5.4 match the results of `estatg` found in Table 5.2. Note that the GEE -R^2 measure is not available in the output of `fitstat` since this measure is specifically designed for GEE models.

5.5 Conclusion and Discussion

The R^2 measure is a popular goodness of fit statistic that was not made available for GEE models. By building on other pseudo-R^2 measures and writing them in terms of the quasi-likelihood instead of the maximum likelihood, we have made an important statistic that will be available in Stata for longitudinal, clustered, or panel data. Researchers can now get a measure of the variance explained in a GEE model. The GEE R^2 measure will be an important tool in model selection for finding the balance of significant predictors.

Table 5.1 Results of xtgee model with binomial link and independent working correlation

```
xtset id
xtgee low age lwt i.race smoke ptl ht ui, family(binomial)
robust corr(ind)nolog
```

GEE population-averaged model						No. of obs = 189
Group variable: id						Number of groups = 189
Link: logit						Obs per group: min = 1
Family: binomial						avg = 1.0
Correlation: independent						Max = 1
						Wald chi2(8) = 29.02
Scale parameter: 1						Prob > chi2 = 0.0003
Pearson chi2(189): 182.02						Deviance = 201.45
Dispersion (Pearson): .9630865						Dispersion = 1.065862
						(Std. Err. Adjusted for clustering on id)
low	Coef.	Robust Std. Err.	z	P>|z|	[95%Conf.	Interval]
age	-0.0271003	0.033843	-0.8	0.423	-0.09343	0.03923
lwt	-0.01515	0.007128	-2.13	0.034	-0.02912	-0.00118
race						
2	1.262647	0.507421	2.49	0.013	0.26812	2.257175
3	0.862079	0.4335	1.99	0.047	0.012435	1.711724
smoke	0.923345	0.386719	2.39	0.017	0.165389	1.6813
ptl	0.541837	0.411416	1.32	0.188	-0.26452	1.348197
ht	1.832518	0.656366	2.79	0.005	0.546064	3.118971
ui	0.758514	0.488396	1.55	0.12	-0.19873	1.715752
_cons	0.461224	1.222866	0.38	0.706	-1.93555	2.857997

Table 5.2 Results of `estatg`

Pseudo-R2 measures for GEE models					
GEE	Efron	McFadden	Ben-Akiva Lerman	Cox Snell	Cragg Uhler
0.1331	0.1642	0.1416	0.0649	0.1612	0.2267

Table 5.3 Results of logit model

```
logit low age lwt i.race smoke ptl ht ui nolog
```

Logistic regression					Number of obs = 189	
					LR chi2(8) = 33.22	
					Prob > chi2 = 0.0001	
Log likelihood = -100.724					Pseudo R2 = 0.1416	
low	Coef.	Std. Err.	z	P>|z|	[95%Conf.	Interval]
age	-0.0271	0.03645	-0.74	0.457	-0.09854	0.044341
lwt	-0.01515	0.006926	-2.19	0.029	-0.02873	-0.00158
race						
2	1.262647	0.52641	2.4	0.016	0.230902	2.294392
3	0.862079	0.439153	1.96	0.05	0.001355	1.722804
smoke	0.923345	0.400827	2.3	0.021	0.137739	1.708951
ptl	0.541837	0.346249	1.56	0.118	-0.1368	1.220472
ht	1.832518	0.691629	2.65	0.008	0.476949	3.188086
ui	0.758514	0.459377	1.65	0.099	-0.14185	1.658875
_cons	0.461224	1.20459	0.38	0.702	-1.89973	2.822176

Table 5.4 Results of `fitstat`

	logit
Log-likelihood	
Model	-100.724
Intercept	-117.336
Chi-square	
Deviance	201.448
LR	33.224
p-value	0
R2	
McFadden	0.142
McFadden (adjusted)	0.065
McKelvey & Zavoina	0.246
Cox-Snell/ML	0.161
Cragg-Uhler/Nagelkerke	0.227
Efron	0.164
Tjur’s D	0.167
Count	0.735
Count (adjusted)	0.153
IC	
AIC	219.448
AIC divided by n	1.161
BIC	248.624
Variance of	
e	3.29
y-star	4.363

CHAPTER 6

MODIFIED QUASI-LIKELIHOOD INFORMATION CRITERION

This chapter introduces the current quasi-likelihood information criterion for selecting the working correlation structure for generalized estimating equations. The modified quasi-likelihood information criterion is proposed and simulation results are shown to evaluate the modified QIC.

6.1 INTRODUCTION

In generalized linear models, there is a statistic that measures the relative quality of a model for a given dataset referred to as the Akaike information criterion (AIC). This criterion measure is used to balance finding the best model in terms of maximizing the likelihood with model simplification in terms of including only those terms that substantially contribute to the model. The AIC gives an estimate of the information lost when a given model is compared to the expectation of the true model; it is also a measure of separation between these two models. This theory is based on the Kullback-Leibler (1951) information divergence which is a non-symmetric measure of the difference between two probability distributions. The Kullback-Leibler information between a candidate model and the true model is written as

$$\Delta_0(\beta, \beta_*) = E_{M_*}[-2\mathcal{L}(\beta; D)]$$

where $\mathcal{L}$ is the log-likelihood and the expectation E_{M_*} is taken with respect to the true distribution of D.

The AIC is defined as a function of the log likelihood along with a penalty term based on the number of parameters in the model. It is an unbiased estimator of $E_{M_*}[-2\mathcal{L}(\beta; D)]$ where $\hat{\beta}$ is the maximum likelihood estimator under any candidate model and the expectation is taken over the random $\hat{\beta}$. The goal is to find the model with the lowest loss of information which implies that the lowest AIC is preferred. The AIC should only be used to compare GLMs of the same link and variance function. The AIC measure is written as

$$AIC = -2\mathcal{L}(\theta, \phi; y_1, y_2, \ldots y_n) + 2p$$

where p is the number of parameters estimated in the model. This criterion measure is extremely useful in model selection for GLMs, but cannot be used for GEEs due to the fact that GEEs are non-likelihood based.

6.2 Current Quasi-Likelihood Information Criterion

Pan (2001) developed the quasi-likelihood information criterion (QIC) based on the AIC. He proposed replacing the likelihood by the quasi-likelihood under the working independence model to define a new measure similar to Kullback-Leibler (1951) as

$$\Delta_0(\beta, \beta_*, I) = E_{M_*}[-2Q(\beta; I, D)].$$

Here it is assumed that any quasi-likelihood model can be indexed by the parameter vector β, and that β_* is the corresponding parameter for the quasi-likelihood model introduced by the true data-generating model M_*.

Pan assumes that the GEE estimator $\hat{\beta} = \hat{\beta}(R)$ is obtained using any general working correlation structure R. Then $E_{M_*}[\Delta_0(\beta, \beta_*, I)]$ can then be approximated as

$$E_{M_*}[\Delta_0(\beta, \beta_*, I)] \approx -2E_{M_*}\left[Q\left(\hat{\beta}; I, D\right)\right] + 2E_{M_*}\left[\left(\hat{\beta} - \beta_*\right)' S\left(\hat{\beta}; I, D\right)\right] + 2\,\text{trace}(\Omega_I, J),$$

where $J = cov(\hat{\beta})$, which can be consistently estimated by the sandwich or robust covariance estimator, $\hat{V}_r$ and Ω_I can also be consistently estimated by its empirical estimator $\widehat{\Omega}_I = -\frac{\partial^2 Q(\beta;I,D)}{\partial\beta\partial\beta'}|_{\beta=\widehat{\beta}} = \sum_{i=1}^{n}(D_i A_i^{-1} D_i)^{-1} D_i V_i^{-1} var(y_i)\, V_i^{-1} D_i (D_i A_i^{-1} D_i)^{-1}$. The estimating equation in the second term is $S(\beta; R, D) \equiv \sum_{i=1}^{n} D_i' V_i^{-1}(Y_i - \mu_i) = 0$, where $D_i = D_i(\beta) = \frac{\partial \mu_i(\beta)}{\partial \beta'}$ and V_i is a working covariance matrix of Y_i. The working covariance matrix of Y_i can be expressed as $R = R(\alpha), V_i = A_i^{\frac{1}{2}} R(\alpha) A_i^{\frac{1}{2}}$, where A_i is a diagonal matrix with elements $var(Y_{ij}) = \phi v(\mu_{ij})$, which is a specified function of the mean. Pan ignores the second term which is difficult to estimate, and proposes the estimator

$$QIC(R) \equiv -2Q(\beta(R); I, D) + 2\ \text{trace}(\widehat{\Omega}_I, \hat{V}_r).$$

The $QIC(R)$ measure is Pan's proposed quasi-likelihood under the independence model criterion for GEE.

Pan notes that ignoring the second term does somewhat influence the performance of $QIC(R)$, but not dramatically. In his simulations, he shows that the AIC is more efficient than the QIC for possibly two reasons. The first reason is that the maximum likelihood estimator of β is more efficient than the GEE estimator, and the second reason is that the information of the true correlation structure is within the likelihood function in the AIC, but it is not embedded in the quasi-likelihood in the QIC. In Pan's simulation, he only examines the independent, exchangeable, and AR(1) working correlation matrices and does not include the unstructured matrix as a choice. He found that the QIC favored the correct correlation structure 67.8% to 72.1% when the sample size was 50 and 100 respectively.

Other researchers have examined the performance of Pan's QIC. Barnett et. al. [2010] noted that the overall success of the QIC was 29.4% and favored the simpler covariance structure in when examining ecological data. Hin, Carey, and Wang [2007] showed that QIC had a detection rate of 60-70% for most of the simulated scenarios using clustered data. When the incorrect structure is favored such as the independence structure, Fitzmaurice [1995] notes that assuming independence can lead to a considerable loss of efficiency in estimating the regression parameters.

6.3 Modified Quasi-Likelihood Information Criterion

Since the current information criterion is not as efficient as it could be, a modified QIC is proposed. This modified QIC is built on the current QIC as it still uses the calculated quasi-likelihood measure and takes into account the number of parameters in the model. However, this modified QIC also takes into account the number of correlation coefficients estimated in the model, denoted as m. The modified QIC can be written as

$$mQIC = -2Q\left(\hat{\beta}(R); I, D\right) + 2\text{trace}\left(\hat{\Omega}_I, \hat{V}_r\right) * 2p - m(\text{trace}\left(\hat{\Omega}_I, \hat{V}_I\right),$$

where $\hat{V}_I = \sum_{i=1}^{n} \frac{1}{n} (D_i A_i^{-1} D_i)$. The modified QIC provides a balance between the independent structure that has no correlation estimates and the unstructured covariance matrix that estimates the most correlation parameters.

The modified QIC favors the unstructured covariance matrix as in the modified QIC under the unstructured working correlation matrix always calculates the smallest value for the modified QIC. The modified QIC's third term is zero when the independent working correlation matrix, and the modified QIC reduces to $mQIC(I) = -2Q\left(\hat{\beta}(R); I, D\right) + 2\text{trace}\left(\hat{\Omega}_I, \hat{V}_r\right) * 2p$.

6.4 Simulation Studies

Simulations were used to demonstrate and asses the performance of the proposed modified QIC. In total, 44 different combinations of correct covariance structure, number of measurements on each subject, and correlation value ρ were examined. The independent, exchangeable, autoregressive(1), and unstructured covariance matrices were examined. Possible number of measurements t were 3, 5, 7, and 9. The possible values of ρ ranged from slightly correlated to heavily correlated: 0.1, 0.3, 0.5, 0.7, and 0.9.

The first step in the simulation was to generate panel data with the specified covariance structure. The model chosen for simulation was the same as in Pan [2001] and Fitzmaurice [1995]. The response variable Y_{it} is a binary outcome and its marginal mean is μ_{it}, with

$$logit(\mu_{it}) = \beta_0 + \beta_1 x_{1,it} + \beta_2 (t-1)$$

where the $x_{1,it}$ are identically and independently distributed Bernoulli ($x_{1,it} = 0$ or $x_{1,it} = 1$ with probability ½) and $\beta_0 = 0.25 = -\beta_1 = -\beta_2$ and where $t = 1, 2, 3$ and $i = 1, \ldots, n$. The Y_i joint distribution was simulated from Bahadur's [1961] representation from Fitzmaurice [1995]. A sample size of 1000 was generated under a specified working covariance structure. The models were fit using Stata's xtgee command, and then the AIC, Pan's QIC, and the modified QIC were calculated and recorded into a separate dataset. Each simulation run was ranked and the chosen correlation structure with smallest QIC was recorded. Tables showing the percentage of the working correlation matrix selected by Pan's QIC versus the modified QIC for the marginal logistic model from 1000 independent replications for each structure is shown

below. Since Pan's original simulation did not include the unstructured covariance matrix as a choice, the choices of covariance structure are independent, exchangeable, or autoregressive(1).

Table 6.1 shows that Pan's QIC favors the correct correlation structure less than 50% of the time under all combinations of t and ρ. When the number of measurements on a single person gets large, Pan's QIC has a more difficult time selecting the correct correlation structure. The same is true when the measurements within an individual become more correlated. When comparing Pan's QIC to the modified QIC percentages from Table 6.2, it is clear that the modified QIC outperforms Pan's QIC in selection of the correct covariance structure. When the number of measurements on a single person gets large, the modified QIC still performs well. The same is true when the measurements within an individual become more correlated.

Table 6.3 shows that Pan's QIC favors the correct correlation structure less than 50% of the time under all combinations of t and ρ. When the number of measurements on a single person gets large, Pan's QIC has a more difficult time selecting the correct correlation structure. The same is true when the measurements within an individual become more correlated. When comparing Pan's QIC to the modified QIC percentages from Table 6.4, it is clear that the modified QIC outperforms Pan's QIC in selection of the correct covariance structure. When the number of measurements on a single person gets large, the modified QIC still performs well. The same is true when the measurements within an individual become more correlated.

When the true correlation structure is independent, Pan's QIC performs better than the modified QIC. Table 6.5 shows that the modified QIC always chooses

unstructured first and independent last. Pan's QIC only chooses the independent working correlation structure less than 32% of the time.

6.5 Conclusion and Discussion

Under all combinations of the simulation, the modified QIC outperforms the currently available Pan's QIC. The best percentage that Pan's QIC achieves is 54% while the best percentage the modified QIC achieves is 99.2%. Pan's simulation did not include the unstructured covariance matrix as a choice. In our simulation, when including the unstructured covariance structure as a choice, the modified QIC favors the unstructured matrix all the time while Pan's QIC continues to favor the more simplified independent structure. It is important to use the correct correlation structure in modeling because the correct correlation structure improves estimation efficiency.

Table 6.1: Proportion of time Pan's QIC identifies correct correlation structure when the true correlation structure is exchangeable

t	$\rho = 0.1$	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$
3	48.0%	31.4%	11.8%	4.9%	1.7%
5	42.9%	13.4%	2.5%	0.5%	0.0%
7	33.1%	5.6%	0.8%	0.1%	0.0%
9	26.6%	2.5%	0.4%	0.0%	0.0%

Table 6.2: Proportion of time modified QIC identifies correct correlation structure when the true correlation structure is exchangeable

t	$\rho = 0.1$	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$
3	92.0%	99.2%	96.2%	91.4%	78.1%
5	98.4%	96.1%	92.0%	83.8%	63.6%
7	99.2%	91.0%	80.1%	67.1%	37.7%
9	98.8%	82.0%	66.3%	46.6%	20.6%

Table 6.3: Proportion of time Pan's QIC identifies correct correlation structure when the true correlation structure is AR(1)

t	$\rho = 0.1$	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$
3	54.3%	33.0%	12.0%	4.1%	1.4%
5	54.3%	20.1%	4.2%	0.3%	0.2%
7	50.4%	13.5%	1.6%	0.0%	0.0%
9	47.3%	6.8%	0.5%	0.0%	0.0%

Table 6.4 Proportion of time modified QIC identifies correct correlation structure when the true correlation structure is AR(1)

t	$\rho = 0.1$	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$
3	89.8%	97.3%	89.6%	69.0%	51.2%
5	97.3%	98.5%	85.8%	55.5%	37.0%
7	98.2%	96.9%	71.6%	36.6%	21.1%
9	98.8%	94.7%	59.2%	20.5%	12.3%

Table 6.5 Comparison when true correlation structure is independent

	$t = 3$	$t = 5$	$t = 7$	$t = 9$
Pan's QIC	31.2%	29.1%	26.6%	28.0%
Modified QIC	0.0%	0.0%	0.0%	0.0%

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

This dissertation has presented three significant contributions to generalized linear models and generalized estimating equations. These contributions included a penalized maximum likelihood estimation method for generalized linear models, an R^2 and several pseudo-R^2 measures for generalized estimating equations, and a modified quasi-likelihood information criterion for generalized estimating equations.

The new estimation method presented within this dissertation helps fix problems encountered in real data analysis such as bias, small sample size, and separation of data points. The penalized maximum likelihood estimation method is available as a Stata command `firthglm`. The new statistics presented for generalized estimating equation models further extend the usefulness and interpretation of these widely used models and provide diagnostic and model selection tools not previously available to researchers. The R^2 and pseuso-R^2 calculations are available in a post estimation command, `estatg`, in Stata.

7.2 FUTURE WORK

In order to expand this work and make these statistics and methods available to more researchers, packages for other statistical software will be developed. The penalized maximum likelihood estimation method, R^2 and pseuso-R^2 calculations, and modified

QIC calculations will be made available in SAS, statistical analysis system, and R, a free statistical software environment.

In generalized estimating equations, there is also another criterion for selecting the covariates, or independent variables, to include within the model. This criterion is known as the QIC_u. I will investigate a modified QIC_u measure and implement it within the same software packages. In the future, I will also investigate a similar modification to the Bayesian information criterion (BIC) and evaluate its efficiency compared to the Akaike information criterion, Quasi-likelihood information criterion, and the modified Quasi-likelihood information criterion.

REFERENCES

Adrian G. Barnett, Nicola Koper, Annette J. Dobson, Fiona Schmielgelow, and Micheline Manseau. Using information criteria to select the correct variance-covariance structure for longitudinal data in ecology. *Methods in Ecology & Evolution*, 1(1): 15-24, 2010.

Moshe Ben-Akiva and Steven R. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge, MA: MIT Press, 1985.

Raghu R. Bahadur. A representation of the joint distribution of responses to n dichotomous items. *Studies in Item Analysis and Prediction,* Volume VI, *Stanford Mathematical Studies in the Social Sciences,* ed. H. Solomon, 158-168, Stanford, CA: Stanford University Press , 1961.

Joseph Coveney. FIRTHLOGIT: Stata module to calculate bias reduction in logistic Regression. http://EconPapers.repec.org/RePEc:boc:bocode:s456948, 2008.

John G. Cragg and Russell S. Uhler. The demand for automobiles. *Canadian Journal of Economics*, 12(3): 386–406, 1970.

Chelsea B. Deroche, Jose D. Villa, and Luis A. Escobar. Statistical methods to quantify the effect of mite parasitism on the probability of death in honey bee colonies. *Estadistica*, 63(181): 95-112, 2011.

Bradley Efron. Regression and ANOVA with zero-one data: measures of residual variation. *Journal of the American Statistical Association*, 73(161): 113–121, 1978.

David Firth. Bias reduction of maximum likelihood estimates, *Biometrika*, 80(1): 27-38, 1993.

Garrett M. Fitzmaurice. A caveat concerning independence estimating equations with multiple multivariate binary data. *Biometrics*, 51(1): 309–317, 1995.

James W. Hardin and Joseph M. Hilbe. *Generalized Linear Models and Extensions (Third Edition)*. College Station, TX: Stata Press, 2011.

Walter W. Hauck, Jr. and Allan Donner. Wald's test as applied to hypotheses in logit analysis. *Journal of the American Statistical Association*, 72(360): 851-853, 1977.

Georg Heinze and Michael Schemper. A soulution to the problem of separation in logistic regression. *Statistics in Medicine*, 2: 2409-2419. 2002.

Lin-Yee Hin, Vincent J. Carey, and You-Gan Wang. Criteria for working-correlation-structure selection in GEE: assessment via simulation. *The American Statistician*, 61(4): 360–364, 2007.

David W. Hosmer, Jr. Stanley Lemeshow, Rodney X. Sturdivant. *Applied Logistic Regression (Third Edition*). New York: Wiley, 2013.

Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Royal Society Publishing: Proceedings of the Royal Society of London. Series A (Mathematical and Physical Sciences)*, 186(1007): 456-461, 1946.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1): 79-86, 1951.

Kung-Yee Liang and Scott L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika* 73(1): 13-22, 1986.

Gangadharrao Soundalyarao Maddala. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press, 1986.

Lonnie Magee. R^2 measures based in Wald and likelihood ratio joint significant tests. *The American Statistician*, 44(3): 250-253, 1990.

Peter McCullagh and John A. Nelder. *Generalized Linear Models (Second Edition).* London: Chapman & Hall, 1989.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. *Frontiers of Econometrics*, ed. P. Zarembka, 105–142, New York: Academic Press, 1974.

Richard D. McKelvey and William Zavoina. A statistical model for the analysis of ordinal level dependent variables. *Journal of Mathematical Sociology*, 4(1): 103–120, 1975.

Sundar Natarajan, Stuart Lipsitz, Michael Parzen, Stephen Lipshultz. A measure of partial association for generalized estimating equations. *Statistical Modelling*, 7(2): 175-190, 2007.

Wei Pan. Akaike's Information criterion in generalized estimating equations. *Biometrics*, 57(1): 120-125, 2001.

Scott L. Zeger, Kung-Yee Liang, and Paul S. Albert. Models for longitudinal data: A generalized estimating equation approach. *Biometrics*, 42(4): 1049-1060, 1988.

APPENDIX A – STATA CODE FOR R^2 AND PSEUDO-R^2

```
*! version 1.0.0
program define estatg, rclass
        syntax

        quietly {
                if "`e(cmd)'" != "xtgee" {
                        noi di as err "this command must follow -xtgee-"
                        exit 199
                }


                local y "`e(depvar)'"

                tempname b Vr C Q
                local N     = e(N)                  // Number of observations
                local Ng    = e(Ng)                 // Number of groups
                matrix `b'  = e(b)                  // Estimated coefficient vector
                matrix `Vr' = e(V)                  // Variance estimate

                FixMat `b' `Vr'
                local p = colsof(`Vr')-1

                if `p' == 0 {
                        matrix `C'  = (1)
                }
                else {
                        matrix `C'  = I(`p') , J(`p',1,0)
                }
                matrix `Q'  = (`C'*`b')' * syminv(`C'*`Vr'*`C') * (`C'*`b')
                local Qv    = `Q'[1,1]
                local r2    = (`Qv'/`N') / (1 + (`Qv'/`N'))

                GetM
                local m "`r(m)'"

                tempname xb mu
                predict double `xb', xb   // linear predictor
                predict double `mu'       // default is the mean (scale of outcome)    mu is y-
hat
```

```
		replace `xb' = . if ! e(sample)


		tempvar ef1 ef2
		gen double `ef1' = (`y'-`mu')^2		// Sum(y - yhat)
		summ `y' if e(sample), meanonly
		gen double `ef2' = (`y'-`r(mean)')^2	// Sum(y - ybar)
		summ `ef1', meanonly
		local num = r(sum)
		summ `ef2', meanonly
		local den = r(sum)

		local efron = 1 - `num'/`den'

		GetVar
		local var = r(var)

		global SGLM_y "`y'"  // Set this global variable so we can use the
glim_v## commands

		global SGLM_m "`m'"		// Set this global variable so we can use the
glim_v## commands

		global SGLM_s1 = 1 // `e(phi)'


		local scale = e(phi)
		tempvar QR
		glim_v`var' 3 `xb' `mu' `QR'
		summ `QR' if e(sample), meanonly
		local QbetaR = r(sum)/`scale'

		tempvar QI

		preserve			// preserve - we are going to run another model
		tempname hold
		estimates store `hold'

		capture {
			local cmd "`e(cmdline)'"
			local i   = index("`cmd'",",")
			local ip1 = `i'+1
			local opt = substr("`cmd'",`ip1',.)
```

```
if "`e(wtype)'" != "" {
                local wexp "[`e(wtype)' `e(wexp)']"
        }
        xtgee `y' `wexp' if e(sample), `opt'

        tempvar mu2
        predict `mu2'

        glim_v`var' 3 `xb' `mu2' `QI'
        summ `QI' if e(sample), meanonly
        local QbetaI = r(sum) / `e(phi)'

}
estimates restore `hold'
restore

if "`QbetaI'" == "" {
        local QbetaI = .
}


// local QbetaR = 2*abs(`QbetaR')
// local QbetaI = 2*abs(`QbetaI')

local mf    = 1-(`QbetaR'/`QbetaI')
local bal   = 1-((`QbetaR'+2*(`p'+1))/(`QbetaI'))
local power = 2/`N'
local ml    = 1 - exp((`QbetaR'-`QbetaI')/`N')
local cu    = `ml'/(1-exp(-`QbetaI'/`N'))

local rvals "N r2 efron mf bal ml cu"
if "`e(family)'" == "Gaussian" {
        local rvals "N r2 efron"
}

noi di
noi di "Pseudo-R2 measures for GEE models"

if "`e(family)'" == "Gaussian" {
        noi di as txt ///
                _col(10) "GEE" ///
                _col(20) "Efron"

        noi di _col(10) _c
        foreach val in r2 efron {
                noi di as res %6.4f ``val'' "    " _c
```

```
                }
        }
        else {
                noi di as txt ///
                        _col(10) "" ///
                        _col(20) "" ///
                        _col(30) "" ///
                        _col(40) "Ben-Akiva" ///
                        _col(50) "Cox" ///
                        _col(60) "Cragg"
                noi di as txt  ///
                        _col(10) "GEE" ///
                        _col(20) "Efron" ///
                        _col(30) "McFadden" ///
                        _col(40) "Lerman" ///
                        _col(50) "Snell" ///
                        _col(60) "Uhler"

                noi di _col(10) _c
                foreach val in r2 efron mf bal ml cu {
                        noi di as res %6.4f ``val" "    " _c
                }
        }
        noi di

        foreach val in `rvals' {
                ret scalar `val' = ``val"
        }

        ret scalar QbetaI = `QbetaI'
        ret scalar QbetaR = `QbetaR'
    }
end
```

```
capture program drop GetVar
program define GetVar, rclass
	quietly {
		local f = substr(lower("`e(family)'"),1,3)   // Grab first 3 letters of e(family)

		local v = .                        // Map each family to its glim_v# command
		if "`f'" == "bin" {
			local v = 2
		}
		else if "`f'" == "gau" {
			local v = 1
		}
		else if "`f'" == "gam" {
			local v = 4
		}
		else if "`f'" == "poi" {
			local v = 3
		}
		else if "`f'" == "inv" {
			local v = 5
		}
		else if "`f'" == "neg" {
			local v = 6
		}
		ret scalar var = `v'
	}
end

capture program drop GetM
program define GetM, rclass
	quietly {
		local cmd = lower("`e(cmdline)'")
		local i   = index("`cmd'","family(binomial")
			local j   = `i'+15
		local cmd = substr("`cmd'",`j',.)
		local i   = index("`cmd'",")")
		local i   = `i'-1
		local m   = substr("`cmd'",1,`i')      // Get argument to "family(binomial arg)"

		if "`m'" == "" {
			local m "1"
		}
		ret local m "`m'"
	}
end
```

```
capture program drop FixMat
program define FixMat
        args b v
        quietly {
                tempname bc vc
                mat `bc' = `b'
                mat `vc' = `v'
                local ind ""
                local nn = 0
                local nc = colsof(`bc')
                forvalues k=1/`nc' {
                        if `vc'[`k',`k'] != 0 {
                                local ind = "`ind' `k'"
                                local nn = `nn'+1
                        }
                }
                mat `b' = J(1,`nn',0)
                mat `v' = J(`nn',`nn',0)
                local k 1
                foreach j in `ind' {
                        mat `b'[1,`k'] = `bc'[1,`j']
                        local h 1
                        foreach i in `ind' {
                                mat `v'[`k',`h'] = `vc'[`j',`i']
                                mat `v'[`h',`k'] = `vc'[`i',`j']
                                local h = `h'+1
                        }
                        local k = `k'+1
                }
        }
end
```

```
program drop _all
clear

webuse lbw
xtset id
xtgee low age lwt i.race smoke ptl ht ui, family(binomial) robust corr(ind)
estatg
ret list
logit low age lwt i.race smoke ptl ht ui
fitstat
exit
xtgee bwt age lwt i.race smoke ptl ht ui, fam(gauss) robust corr(ind)
estatg
ret list
exit
```

APPENDIX B – STATA CODE FOR MODIFIED QIC SIMULATION

```
capture program drop MySim
program define MySim, rclass
        args n t rho
        drop _all
        MakePanelData `n' `t', gee(exch `rho') logistic clear
        tempvar eta
        gen double `eta' = 0

        local nglim = 1

        global SGLM_m "1"
        local fam "bin"
        local lnk "logit"
        local nglim = 2


        // Model A
        xtgee y a b, i(id) t(t) fam(`fam') link(`lnk') corr(ind)
        tempname betaI SigmaI muI
        matrix `betaI'   = e(b)    // A(1) from notes
        matrix `SigmaI'  = e(V)    // A(2) from notes
        predict double `muI', mu   // A(3) from notes

        tempname traceR traceI

        foreach corr in ind exch ar1 unst {

                // Model B
                qui xtgee y a b, i(id) t(t) fam(`fam') link(`lnk') corr(`corr') robust
                tempname betaR VR muR
                matrix `betaR' = e(b)     // B(1) from notes
                matrix `VR'    = e(V)     // B(2) from notes
                predict double `muR', mu  // B(3) from notes
        local p = rowsof(`VR')    // B(4) from notes
                local scale = e(phi)      // B(5) from notes
                local rank = e(rank)
```

```
// Model c
qui capture noi glm y a b, fam(`fam') link(`lnk') from(`betaR') iter(0)

tempname SigmaR
matrix `SigmaR' = e(V)    // C(1) from notes


/*************************************************************
    Need to define global macro SGLM_m so I can use -glm-
    helper programs to calculate the quasilikelihood.

    Helper functions work like this:

    glim_v# TODO ETA MU QQ
    #={1,2,3,4,5} for {gauss, binomial, poisson, gamma, inv gauss}
    TODO = 3 if you want quasilikelihood defined in variable QQ
    ETA  = whatever (ignored for TODO=3)
    MU   = values to use for fitted values in calculation of QQ
    QQ   = place to store quaslikelihood values

*************************************************************/

if `nglim' == 2 {
        global SGLM_m "1"
}
tempvar eta QR QI

qui gen `eta' = .

qui glim_v`nglim' 3 `eta' `muR' `QR'

qui glim_v`nglim' 3 `eta' `muI' `QI'

qui summ `QR', meanonly
local QbetaR = r(sum)/`scale'   // C(2) from notes

qui summ `QI', meanonly
local QbetaI = r(sum)/`scale'   // C(3) from notes

matrix `traceR' = trace(invsym(`SigmaR')*`VR')
matrix `traceI' = trace(invsym(`SigmaI')*`VR')

local offR = `traceR'[1,1]
local offI = `traceI'[1,1]
```

```
		local ncorr = 0
		if "`corr'" == "exch" | "`corr'" == "ar1" {
			local ncorr = 1
		}
		else if "`corr'" == "unst" {
			local ncorr = `t' * (`t'-1) / 2
		}

		ret scalar AIC`corr'	=  `QbetaR' + 2*`p' + 2*`ncorr'
		ret scalar PanQIC`corr'	=  `QbetaR' + 2*`offR'
		ret scalar HHQIC`corr'	=  `QbetaR' + 2*`offI'
		ret scalar QICu`corr'	=  `QbetaR' + 2*`p'
		ret scalar traceR`corr'	=  `offR'
		ret scalar traceI`corr'	=  `offI'
		ret scalar QbetaR`corr'	=  `QbetaR'
		ret scalar QbetaI`corr'	=  `QbetaI'

		ret scalar scale`corr'	=  `scale'

		ret scalar New1`corr'	=  `QbetaR' + `offR' + `offI'
		ret scalar New2`corr'	=  `QbetaR' + 2*`offR'*2*`p'
		ret scalar New3`corr'	=  `QbetaR' + 2*`offI'*2*`p'
		ret scalar New4`corr'	=  `QbetaR' + ((2*(`offR'+1)*(`offR'+2))/(`n'-
`offR'-2))
		ret scalar New5`corr'	=  `QbetaR' + ((2*(`offI'+1)*(`offI'+2))/(`n'-`offI'-
2))
		ret scalar New6`corr'	=  `QbetaR' +
(((`offR'+1)*(`offI'+1)*(`offR'+2)*(`offI'+2))/(`n'-(0.5*`offI'+0.5*`offR')-2))
		ret scalar New7`corr'	=  `QbetaR' + (`n'/(`n'-`p'-`ncorr'-
2))*(2*(`p'+`ncorr'+2))
		ret scalar New8`corr'	=  `QbetaR' + (2*(`m'+1)*(`m'+2))/(`n'-`m'-2)
		ret scalar New9`corr'	=  `QbetaR' + (2*(`m'+1)*(`m'+2))/(`p'-`m'-2)
		ret scalar New16`corr'	=  `QbetaR' + 2*`traceR'*2*`p' - `ncor'*`traceI'
		ret scalar New17`corr'	=  `QbetaR' + 2*`offR'*2*`p' - `ncor'*`offI'

		ret scalar p`corr'	= `p'
		ret scalar t`corr'	= `t'
		ret scalar ncor`corr'	= `ncorr'
		ret scalar offR`corr'	= `offR'
		ret scalar offI`corr'	= `offI'
		ret scalar ncor`corr'	= `ncorr'
		ret scalar traceR`corr'	=  `traceR'
		ret scalar traceI`corr'	=  `traceI'
		}
end
```

```
local types "ar1 exch ind unst"
local args ""
foreach corr in `types' {

        local args "`args' AIC`corr'     = r(AIC`corr')"
        local args "`args' PanQIC`corr'  = r(PanQIC`corr')"
        local args "`args' HHQIC`corr'   = r(HHQIC`corr')"
        local args "`args' QICu`corr'    = r(QICu`corr')"
        local args "`args' traceR`corr'  = r(traceR`corr')"
        local args "`args' traceI`corr'  = r(traceI`corr')"
        local args "`args' QbetaR`corr'  = r(QbetaR`corr')"
        local args "`args' QbetaI`corr'  = r(QbetaI`corr')"
        local args "`args' scale`corr'   = r(scale`corr')"

        local args "`args' New1`corr'    = r(New1`corr')"
        local args "`args' New2`corr'    = r(New2`corr')"
        local args "`args' New3`corr'    = r(New3`corr')"
        local args "`args' New4`corr'    = r(New4`corr')"
        local args "`args' New5`corr'    = r(New5`corr')"
        local args "`args' New6`corr'    = r(New6`corr')"
        local args "`args' New7`corr'    = r(New7`corr')"
        local args "`args' New8`corr'    = r(New8`corr')"
        local args "`args' New9`corr'    = r(New9`corr')"
        local args "`args' New16`corr'    = r(New16`corr')"
        local args "`args' New17`corr'    = r(New17`corr')"

        local args "`args' p`corr'       = r(p`corr')"
        local args "`args' t`corr'       = r(t`corr')"
        local args "`args' ncor`corr'    = r(ncor`corr')"

}

// exch 0.1
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch01n100t3a, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch01n100t5a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch01n100t7a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch01n100t9a, replace

//exch 0.3
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch03n100t3a, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch03n100t5a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch03n100t7a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch03n100t9a, replace

//exch 0.5
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch05n100t3a, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch05n100t5a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch05n100t7a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch05n100t9a, replace

//exch 0.7
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch07n100t3a, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch07n100t5a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch07n100t7a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch07n100t9a, replace

// exch 0.9
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch09n100t3a, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch09n100t5a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch09n100t7a, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save exch09n100t9a, replace

// ar1 0.1
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar101n1000t3, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar101n1000t5, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar101n1000t7, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar101n1000t9, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 20 .1
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar101n1000t20, replace
```

```
//ar1 0.3
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar103n1000t3, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar103n1000t5, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar103n1000t7, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar103n1000t9, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 20 .3
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar103n1000t20, replace

//ar1 0.5
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar105n1000t3, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar105n1000t5, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar105n1000t7, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar105n1000t9, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 20 .5
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar105n1000t20, replace

//ar1 0.7
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar107n1000t3, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar107n1000t5, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar107n1000t7, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar107n1000t9, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 20 .7
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar107n1000t20, replace

// ar1 0.9
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar109n1000t3, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar109n1000t5, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar109n1000t7, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar109n1000t9, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 20 .9
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save ar109n1000t20, replace
```

```
// ind 0.1
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 3 0
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save indn1000t3, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 5 0
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save indn1000t5, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 7 0
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save indn1000t7, replace

set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 9 0
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save indn1000t9, replace
```

```
set seed 9262014
set more off
simulate `args', reps(1000) : MySim 1000 20 0
gen iteration = _n
reshape i iteration
reshape j type ar1 exch ind unst, string
reshape xij AIC PanQIC HHQIC QICu traceR traceI QbetaR QbetaI scale New1 New2
New3 New4 New5 New6 New7 New8 New9 New16 New17 p t ncor
reshape long
save indn1000t20, replace

exit
```

```
program define MakePanelData
        version 11
        syntax anything(name=numlist) [, GEE(string) REGression GAMma POIsson
LOGistic PRObit RE(string) CLEAR]
        quietly {
                describe
                if r(N)+r(k)>0 & "`clear'"=="" {
                        noi di as err "You must specify -clear- if there are data in memory"
                        exit 199
                }

                local n : word 1 of `numlist'
                local t : word 2 of `numlist'

                capture confirm integer number `n'
                if _rc {
                        noi di as err "First argument is incorrect: must be a positive
integer"
                        exit 199
                }
                capture confirm integer number `t'
                if _rc {
                        noi di as err "Second argument is incorrect: must be a positive
integer"
                        exit 199
                }

                local nargs : word count `logistic' `poisson' `regression' `probit' `gamma'

                if `nargs' == 0 {
                        local regression "regression"
                }

                if `nargs' > 1 {
                        noi di as err "You cannot specify more than one of {logistic,
poisson, regression, probit}"
                        exit 199
                }

                if `n' < `t' | `t' < 1 | `t' > 100 {
                        noi di as err "You must specify: at least as many groups as
periods"
                        noi di as err "                number of periods in [1,99]"
                        exit 199
                }
                drop _all
```

```
                set obs `n'

                if "`gee'" != "" & "`re'" != "" {
                        noi di as err "You cannot specify both gee() and re()"
                        exit 199
                }

                if "`gee'" == "ind" {
                        local gee "exch 0"
                }

                if "`gee'" != "" {
                        tempname R
                        local type : word 1 of `gee'
                        local val  : word 2 of `gee'

                        if "`type'" == "user" {
                                mat `R' = `val'
                                local typearg "user `val'"
                        }
                        else {
                                if index("exchar1ind","`type'") == 0 {
                                        noi di as err "Unknown GEE() type"
                                        exit 199
                                }
                                capture confirm num `val'
                                if _rc {
                                        noi di as err "Argument for GEE is not a number"
                                }
                                MakeR `R' "`type'" `val' `t'
                                local typearg "`type'"
                        }

                        noi CheckR `R'
                        MakeGEE`regression'`gamma'`poisson'`logistic'`probit' `R'
"`typearg'"
                }
                else {
                        MakeRE`regression'`gamma'`poisson'`logistic'`probit'
                }
        }
        describe
        notes
end
```

```
program define CheckR
        args R
        capture confirm matrix `R'
        if _rc {
                noi di as err "Correlation matrix does not exist"
                exit 199
        }
        local r = rowsof(`R')
        local c = colsof(`R')
        if `r' != `c' {
                noi di as err "Correlation matrix is not square"
                exit 199
        }
        forvalues i=1/`r' {
                forvalues j=1/`c' {
                        if `i'==`j' {
                                if `R'[`i',`j'] != 1 {
                                        noi di as err "Correlation matrix does not have 1 on
all diagonals"
                                        exit 199
                                }
                        }
                        else {
                                if `R'[`i',`j'] < -.9999 | `R'[`i',`j'] > .9999 {
                                        noi di as err "Correlation matrix has off diagonal
elements > 0.9999 in absolute value"
                                        exit 199
                                }
                                if `R'[`i',`j'] != `R'[`j',`i'] {
                                        noi di as err "Correlation matrix is not symmetric"
                                        exit 199
                                }
                        }
                }
        }
end
```

```
program define MakeR
        args R typ val t

        if "`typ'" == "exch" {
                mat `R' = (1-`val')*I(`t') + J(`t',`t',`val')
        }
        if "`typ'" == "ar1" {
                mat `R' = J(`t',`t',0)
                forvalues row=1/`t' {
                        forvalues col=1/`t' {
                                mat `R'[`row',`col'] = (`val')^(abs(`row'-`col'))
                        }
                }
        }
        if "`typ'" == "ind" {
                mat `R' = I(`t')
        }
end


program define MakeBinaryR
        args R
        local t = colsof(`R')
        tempname S
        matrix `S' = I(`t')

        forvalues i=1/`t' {
                local p`i' = 0.5
        }
        local tm1 = `t'-1
        forvalues i=1/`tm1' {
                local ip1 = `i'+1
                forvalues j=`ip1'/`t' {
                        local rij = `R'[`i',`j']
                        local pij = `rij'*sqrt(`p`i''*(1-`p`i'')*`p`j''*(1-`p`j'')) + `p`i''*`p`j''
                        local rho = sin(2*_pi*(`pij'-0.25))
                        matrix `S'[`i',`j'] = `rho'
                        matrix `S'[`j',`i'] = `rho'
                }
        }
        matrix `R' = `S'
end
```

```
program define MakePoissonR
        args R mu
        local t = colsof(`R')
        tempname S
        matrix `S' = I(`t')

        preserve
        tempvar n1 n2 p1 p2 p3 sd1 sd2

        local tm1 = `t'-1
        forvalues i=1/`tm1' {
                local ip1 = `i'+1
                replace `sd1' = sqrt(`mu'`i')
                replace `n1' = rnormal(`mu'`i',`sd1')
                replace `n3' = rnormal(`mu'`i',`sd1')
                replace `p1' = invpoisson(`mu'`i',normprob(`mu'`i'))
                replace `p3' = invpoisson(`mu'`i',1-normprob(`mu'`i'))
                forvalues j=`ip1'/`t' {
                        replace `sd2' = sqrt(`mu'`j')
                        replace `n2'  = rnormal(`mu'`j',`sd2')
                        replace `n2' =
                        corr
        gen double `n1' = rnormal(`mu',1

        forvalues i=1/`t' {
                local p`i' = 0.5
        }
        local tm1 = `t'-1
        forvalues i=1/`tm1' {
                local ip1 = `i'+1
                forvalues j=`ip1'/`t' {
                        local rij = `R'[`i',`j']
                        local pij = `rij'*sqrt(`p`i''*(1-`p`i'')*`p`j''*(1-`p`j'')) + `p`i''*`p`j''
                        local rho = sin(2*_pi*(`pij'-0.25))
                        matrix `S'[`i',`j'] = `rho'
                        matrix `S'[`j',`i'] = `rho'
                }
        }
        matrix `R' = `S'
end
```

```
program define Finalize
        quietly {
                keep y* a* b*
                gen id  = _n
                reshape i id
                reshape j t
                reshape xij y a b
                noi di
                noi corr y*
                noi di
                reshape long
                compress
                label var a      "binary(p=.5) predictor variable"
                label var b      "uniform(0,1) predictor variable"
                label var id     "panel/group identifier number"
                label var t      "within-group order number"
                label var y      "outcome variable with specified within-group corr"
        }
end
```

```
program define MakeCorrNorm
        args R
        local t = colsof(`R')

        * Create a series of N(0,1) vars
        forvalues i=1/`t' {
                gen double a`i' = rnormal(0,1)
                summ a`i'
                replace a`i' = (a`i'-r(mean))/r(sd)
        }
        * The sample correlation of the a1,...,at variables is not exactly equal to
        * the specified values, so we create n1,...,nt variables with the desired property.
        corr `alist', cov

        tempname R1 R2 R1R2
        matrix `R1' = cholesky(syminv(r(C)))
        matrix `R2' = cholesky(`R')
        matrix `R1R2' = `R1'*`R2'

        forvalues i=1/`t' {
                gen double n`i' = 0
                forvalues j=1/`t' {
                        replace n`i' = n`i' + `R1R2'[`j',`i']*a`j'
                }
        }
        forvalues i=1/`t' {
                drop a`i'
        }
end
```

```
program define MakeGEEregression

        args R type
        local t = colsof(`R')

        MakeCorrNorm `R'
        forvalues i=1/`t' {
                gen double a`i' = uniform() < .5
                gen double b`i' = uniform()
                gen double y`i' = n`i' + .25 - .25*a`i' - .25*b`i'

        }

        noi Finalize
        label data  "Regression GEE with linear predictor = -.25*a - .25*b  +.25"
        note: GENERATED FOR:  xtgee y a b, i(id) t(t) fam(gauss) corr(`type')

end



program define MakeGEEgamma

        args R type
        local t = colsof(`R')
        MakeCorrNorm `R'

        noi di as err "No support for gamma yet"
        exit 199

        noi Finalize
        label data  "Gamma GEE with linear predictor = 0.1x + 0.4"
        note: GENERATED FOR:  xtgee y a b, i(id) t(t) fam(gamma) corr(`type')

end
```

```
program define MakeGEEpoisson

        args R type
        local t = colsof(`R')

        MakePoissonR `R'
        MakeCorrNorm `R'


        forvalues i=1/`t' {
                gen double a`i' = uniform() < .5
                gen double b`i' = uniform()
                gen double y`i' = n`i' + .25 - .25*a`i' - .25*b`i'

        }

        noi di as err "No support for poisson yet"
        exit 199

        noi Finalize
        label data  "Poisson GEE with linear predictor = 2*x + 3"
        note: GENERATED FOR:  xtgee y a b, i(id) t(t) fam(poisson) corr(`type')

end



program define MakeGEElogistic
        args R type

        MakeBinaryR `R'
        MakeCorrNorm `R'


        * We have y1, ..., yn which are binary and have the desired correlation
        * Now, we have to generate the predictors for the given outcomes.  This is
        * backwards from the usual approach, but there is no way around it since we
        * want to specify the correlation of the outcomes.
        *
        * WARNING:  Do not change the definition of the covariates to depend on the
value of t
        *       unless you change the manner in which observations are defined.  The
code
        *       below is not robust to defining predictor variables that are correlated with
        *       the value of time.
```

```
        local t = colsof(`R')

        forvalues i=1/`t' {
                gen byte y`i' = (n`i' > 0)
                gen double a`i' = .
                gen double b`i' = .

        }

        d, short
        local N = r(N)
        forvalues i=1/`N' {
                forvalues j=1/`t' {
                        local y = y`j'[`i']
                        local flag 1
                        while `flag' {
                                local x1 = uniform() < .5
                                local x2 = uniform()
                                local eta = .25 - .25*`x1' -.25*`x2'
                                local mu = 1/(1+exp(-`eta'))
                                if (rbinomial(1,`mu') == `y') {
                                        replace a`j' = `x1' in `i'
                                        replace b`j' = `x2' in `i'
                                        local flag = 0
                                }
                        }
                }
        }

        noi Finalize

        label data  "Logistic GEE with linear predictor = -.25a - .25b + .25"

        note: GENERATED FOR:  xtgee y a b, i(id) t(t) fam(binomial) corr(`type')

end
```

```
program define MakeGEEprobit
        args R type
        MakeBinaryR `R'
        MakeCorrNorm `R'

        * We have y1, ..., yn which are binary and have the desired correlation
        * Now, we have to generate the predictors for the given outcomes.  This is
        * backwards from the usual approach, but there is no way around it since we
        * want to specify the correlation of the outcomes.
        *
        * WARNING:  Do not change the definition of the covariates to depend on the
value of t
        *       unless you change the manner in which observations are defined.  The
code
        *       below is not robust to defining predictor variables that are correlated with
        *       the value of time.

        local t = colsof(`R')
        forvalues i=1/`t' {
                gen byte y`i' = (n`i' > 0)
                gen double a`i' = .
                gen double b`i' = .
        }

        d, short
        local N = r(N)
        forvalues i=1/`N' {
                forvalues j=1/`t' {
                        local y = y`j'[`i']
                        local flag 1
                        while `flag' {
                                local x1 = uniform() < .5
                                local x2 = uniform()
                                local eta = .25 - .25*`x1' -.25*`x2'
                                local mu = invnorm(`eta')
                                if (rbinomial(1,`mu') == `y') {
                                        replace a`j' = `x1' in `i'
                                        replace b`j' = `x2' in `i'
                                        local flag = 0
                                }
                        }
                }
        }
        noi Finalize
        label data  "Probit GEE with linear predictor = -.25*a -.25b + .25"
```

```
        note: GENERATED FOR:  xtgee y a b, i(id) t(t)  fam(binomial) link(probit)
corr(`type')
end

exit
```

This program creates a dataset useful for panel data modeling.