Constructing Inverse Probability Weights for Continuous Exposures

A Comparison of Methods

Ashley I. Naimi,[a] *Erica E.M. Moodie,*[a] *Nathalie Auger,*[b] *and Jay S. Kaufman*[a]

Abstract: Inverse probability–weighted marginal structural models with binary exposures are common in epidemiology. Constructing inverse probability weights for a continuous exposure can be complicated by the presence of outliers, and the need to identify a parametric form for the exposure and account for nonconstant exposure variance. We explored the performance of various methods to construct inverse probability weights for continuous exposures using Monte Carlo simulation. We generated two continuous exposures and binary outcomes using data sampled from a large empirical cohort. The first exposure followed a normal distribution with homoscedastic variance. The second exposure followed a contaminated Poisson distribution, with heteroscedastic variance equal to the conditional mean. We assessed six methods to construct inverse probability weights using: a normal distribution, a normal distribution with heteroscedastic variance, a truncated normal distribution with heteroscedastic variance, a gamma distribution, a *t* distribution (1, 3, and 5 degrees of freedom), and a quantile binning approach (based on 10, 15, and 20 exposure categories). We estimated the marginal odds ratio for a single-unit increase in each simulated exposure in a regression model weighted by the inverse probability weights constructed using each approach, and then computed the bias and mean squared error for each method. For the homoscedastic exposure, the standard normal, gamma, and quantile binning approaches performed best. For the heteroscedastic exposure, the quantile binning, gamma, and heteroscedastic normal approaches performed best. Our results suggest that the quantile binning approach is a simple and versatile way to construct inverse probability weights for continuous exposures.

(*Epidemiology* 2014;25: 292–299)

Inverse probability weighting (IPW) estimators for marginal structural models are common in epidemiology. Such estimators can be used to estimate a standardized measure of effect for time-fixed exposures[1] and account for confounding and selection bias due to measured time-varying confounders affected by prior exposure.[2] The use of IPW has also been extended to address a number of other methodological issues, including time-modified confounding,[3] interaction and effect measure modification,[4,5] spillover effects,[6] and effect decomposition.[7]

Informally, weighted regression models estimate exposure effects in a “pseudo-population” obtained by weighting the observed data by the inverse of the conditional probability of the observed exposure, rendering exposure independent of measured confounders.[8,9] Cole and Hernán[10] have provided guidelines on how to construct inverse probability weights to estimate exposure effects. For a binary exposure, individual weights are constructed by estimating the predicted probability of the observed exposure, often using logistic regression. Weights are constructed by taking the inverse of these predicted probabilities. To obtain asymptotically unbiased exposure effect estimates, correct specification of the model for the weights is required.[8,9,11] To avoid excessively upweighting individuals, weights are usually stabilized by replacing the numerator with the marginal probability of the observed exposure.

Inverse probability weights for a continuous exposure are constructed in a similar fashion.[10–13] However, constructing inverse probability weights for a continuous exposure can be complicated by a number of issues not encountered in the binary exposure setting, including the need to identify a correct distributional form for the exposure, the need to account for nonconstant exposure variance (heteroscedasticity), and the need to deal with outliers that can make highly variable weights more likely.

Submitted 5 April 2013; accepted 6 September 2013.

From the [a]Department of Epidemiology, Biostatistics, and Occupational Health, McGill University, Montreal, QC, Canada; and [b]Institut national de santé publique du Québec, and Research Centre of the University of Montreal Hospital Centre, Montreal, QC, Canada.

The authors report no conflicts of interest.

Supported by Post-Doctoral Research Award from the Fonds de recherche du Québec-Santé (A.I.N.), Natural Sciences and Engineering Research Council University Faculty Award (E.E.M.M.), Career Award from the Fonds de recherche du Québec-Santé (N.A.), and supported by the Canada Research Chairs Program (J.S.K.).

SDC Supplemental digital content is available through direct URL citations in the HTML and PDF versions of this article (www.epidem.com). This content is not peer-reviewed or copy-edited; it is the sole responsibility of the author.

Correspondence: Ashley I. Naimi, Department of Epidemiology, Biostatistics and Occupational Health, McGill University, Purvis Hall, 1020 Pine Ave West, Montreal, QC, Canada H3A 1A2. E-mail: ashley.naimi@mcgill.ca.

ISSN: 1044-3983/14/2502-0292
DOI: 10.1097/EDE.0000000000000053

We explore the performance of six methods to construct inverse probability weights for continuous exposures: a standard approach (based on a normal distribution), an approach based on the gamma distribution, a heteroscedastic normal approach, a heteroscedastic truncated normal approach, a heavy-tailed approach (based on the *t* distribution), and a quantile binning approach (based on exposure ranks). The gamma and heteroscedastic normal models may be more appropriate for exposure distributions in which the variance changes as a function of measured covariates. The quantile binning and *t* distribution approach may help mitigate issues with highly variable weights because of the way in which they handle outliers in the exposure's distribution. Finally, quantile binning may be more appropriate for markedly nonnormal or heteroscedastic exposure distributions because it does not rely on distributional assumptions. We assess the performance of each method in a simulation study using two exposures generated from data in the Québec birth file.

METHODS

Simulated Data

We used Monte Carlo simulation[14] to assess the performance of each method to construct IPWs for continuous exposures. We used data from the Québec birth file[15] to generate our simulated data. We extracted information on 1.3 million live singleton births that occurred between 1989 and 2008 (inclusive), including gestational age (in completed weeks), maternal and paternal date of birth, and parity (total number of previous deliveries). Gestational age at birth was used to create a preterm birth indicator defined as less than 37 completed weeks. We used the preterm birth indicator to set the marginal probability of simulated outcome.

We generated 3000 Monte Carlo samples of 1500 observations each, using covariate information sampled without replacement from the Québec birth file. Covariates were maternal age, paternal age, and parity, which we collectively denote as C. We simulated two exposures, each with a mean defined as:

$$\begin{aligned}\mu = {} & 0.025 \times mage + 0.0025 \times page + 0.00125 \times mage \times page \\ & - 0.21 \times I(parity = 2) - 0.22 \times I(parity = 3) \\ & - 0.45 \times I(parity = 4) - 0.45 \times I(parity \geq 5),\end{aligned}$$

where $I(\bullet)$ is an indicator function taking on a value of 1 when the argument $(\bullet)$ is true (0 otherwise), and *mage* and *page* are maternal and paternal age in years. The first exposure (homoscedastic) followed a normal distribution, defined as $X_1 = 15 + \mu + \varepsilon_1$, with $\varepsilon_1 \sim N(0,2)$. The second exposure (heteroscedastic) followed a Poisson distribution with mean = variance = μ. To make this exposure continuous, we added to it a standard normal random variable. We set negative exposure values equal to zero, yielding a zero-inflated distribution.

For each exposure, we generated a binary outcome following a Bernoulli distribution with

$$\begin{aligned}P(Y_j = 1 \mid X_j, C) = {} & \{1 + \exp(-[\alpha_{0j} + \log(1.25) X_j \\ & + \log(1.7) mage^{1/2} + \log(1.5) page^{1/2} \\ & + \log(0.75) I(parity = 2) \\ & + \log(0.8) I(parity = 3) \\ & + \log(0.85) I(parity = 4) \\ & + \log(0.9) I(parity = 5)])\}^{-1}\end{aligned}$$

where $\alpha_{01} = -11.5$ for exposure X_1 and $\alpha_{02} = -8.05$ for exposure X_2, chosen to render the marginal probability of Y approximately 0.08, which was the proportion of preterm births observed in the birth file. To determine the true marginal value for a single-unit increase in each exposure, we estimated the marginal odds ratio for each scenario using the method of marginal standardization[16] in a simulated data set with 4.5 million observations.

For each exposure scenario in each of the 3000 Monte Carlo samples, we estimated the marginal log-odds ratio for a single-unit increase in the exposure using generalized estimating equations to obtain robust standard error estimates,[17] with IPW to account for confounding.[9]

Statistical Analysis

Stabilized inverse probability weights (*sw*) were defined as:

$$sw = \frac{f_X(X; \mu_1, \sigma_1^2)}{f_{X|C}(X \mid C = c; \mu_2, \sigma_2^2)}, \qquad (1)$$

where $f_{\bullet}(\bullet)$ denotes the probability density function with mean μ and variance σ^2, X a continuous exposure, and C the set of confounders. When the exposure is binary, the density functions in the numerator and denominator of the stabilized weights reduce to the probability mass functions for a binary random variable: $P(X = 1 \mid C = c)$ for exposed persons, and $P(X = 0 \mid C = c)$ for unexposed persons.[9] Thus, the stabilized weight equations for a binary exposure are a special case of (1).

The numerator of (1) is used to control the variability of the estimator.[11(p 584)] The marginal exposure density (ie, not conditional on C) is a common choice and yields a marginal estimate of the exposure-outcome effect.[10] When a model for the denominator of the stabilized weights is correctly specified, weighting by *sw* results in an asymptotically unbiased estimator for the true causal effect.[11,12] We constructed weights defined by *sw* using six approaches.

Normal IPWs

First, we assumed that $f_{\bullet}(\bullet)$ was a normal density and used two generalized linear models with an identity link and normal density function to estimate μ and σ^2 for the numerator and denominator of the stabilized weights:

$$\text{Numerator model}: X_i = \hat{\mu}_1 + \hat{\varepsilon}_{i1} \sim N(0,\hat{\sigma}_1^2)$$

$$\text{Denominator model}: X_i = \hat{\mu}_2 + \hat{\varepsilon}_{i2} \sim N(0,\hat{\sigma}_2^2),$$

where $\hat{\mu}_1 = \hat{\mu}_2 = \hat{\beta}_0 + \hat{\beta}C$, and where $\hat{\varepsilon}_{i1}$ and $\hat{\varepsilon}_{i2}$ are the residuals for individual i from the numerator and denominator model, respectively.

Standard IPWs assuming homoscedastic variance were created by using the predicted means $\hat{\mu}$ for each person. We used the deviance statistic divided by its degrees of freedom from a generalized linear model with a normal error distribution and identity link function as an estimate of $\hat{\sigma}$ for all persons. These estimates were used as parameter estimate values for the normal distribution, defined as:

$$\hat{f}_k(x_i) = \frac{1}{\hat{\sigma}_k\sqrt{2\pi}}\exp\left(\hat{\varepsilon}_{ik}^2 \big/ 2\hat{\sigma}_k^2\right) \tag{2}$$

where (henceforth) $k = 1,2$ denotes the numerator or denominator model.

Inverse probability weights with heteroscedastic variance were implemented by further specifying the variance in (2) as a function of confounders C. Specifically, we defined a variance function[18] as $\hat{\sigma}_{ik} = \hat{\sigma}_k \exp(\hat{\theta}_k C_i)$ for the denominator models, and we generated a stabilized weight for each person using the predicted mean and variance conditional on covariates. We implemented two techniques to obtain heteroscedastic normal IPWs: a two-stage approach as in previous work[13] and a single-stage approach. We also used a heteroscedastic truncated normal model to construct weights by bounding (2) between a_k and b_k , and dividing (2) by $\Phi\{(b_k - \mu_k)/\sigma_k\} - \Phi\{(a_k - \mu_k)/\sigma_k\}$, where $\Phi\{\bullet\}$ is the normal cumulative distribution function. For each exposure, we chose $\min(X_k)-1$ and $\max(X_k)+1$ as the lower (a_k) and upper (b_k) bound, respectively.

Heavy-tailed IPWs

Heavy-tailed IPWs assuming homoscedastic variance were created by standardizing residuals from the normal generalized linear models for the numerator and denominator (defined above) and used to generate a numerator and denominator density value for each person from the t distribution, defined as:

$$\hat{f}_k^{(\eta)}(x_i) = \frac{\Gamma\left\{\frac{\eta+1}{2}\right\}}{\sqrt{\eta\pi}\,\Gamma\left\{\frac{\eta}{2}\right\}}\left(1+\frac{\hat{e}_{ik}^2}{\eta}\right)^{-\frac{\eta+1}{2}} \tag{3}$$

where $\hat{e}_{ik}$ is the standardized residual, η represents the degrees of freedom for the t distribution, and $\Gamma\{\bullet\} = (\bullet-1)!$ represents the gamma function for argument $\{\bullet\}$. For our analysis, we used $\eta = 1$, 3, and 5. Relative to the normal distribution, the t distribution's heavy tails should mitigate the large variation in the weights due to outliers in the far tails of the distribution.[19,20(p 465)] For example, as observed in the birth file, a person whose standardized residual value from the numerator and denominator model is 5.4 and 6.8, respectively, will have a weight of approximately 5,115 using the standard normal distribution, but a weight of 1.7 using the t distribution with 1 degree of freedom.

Gamma IPWs

Gamma IPWs were obtained by fitting a generalized linear model with log-link and gamma distribution to estimate a gamma scale parameter and each person's predicted exposure value given they were exposed. The density function for the gamma IPWs was defined as:

$$\hat{f}_k(x_i) = \frac{1}{x_i\Gamma\left(1/\hat{\kappa}_k^2\right)}\left(\frac{x_i}{\hat{\kappa}_k^2\hat{\mu}_k}\right)^{1/\hat{\kappa}_k^2}\exp\left(-\frac{x_i}{\hat{\kappa}_k^2\hat{\mu}_k}\right)\text{if } x > 0, \tag{4}$$

where κ is the gamma scale parameter. Importantly, the variance of a random variable following the gamma distribution is defined as a function of the mean ($\sigma^2 = \kappa^2\mu^2$), and the gamma model may thus better account for heteroscedastic exposure distributions.

Quantile Binning IPWs

Finally, we used a quantile binning approach to approximate $f_{\cdot}(\bullet)$ by ranking X into j quantiles, and fitting a conditional cumulative logistic model to estimate the predicted probability of being in a given quantile.[21] Specifically, we estimated the denominator value using a model defined as

$$\log\left\{\frac{\alpha_{m+1}+\cdots+\alpha_j}{\alpha_1+\cdots+\alpha_m}\right\} = \gamma_{0m} + \gamma C,$$

where α_m is the probability of being in category m. Predicted probabilities for being in category m were calculated as $\text{expit}\{\gamma_{0m} + \gamma C\} - \text{expit}\{\gamma_{0(m-1)} + \gamma C\}$, where $\text{expit}\{\bullet\} = 1/\{1+\exp(-\bullet)\}$. In our study, we ranked X using $j = 10$, 15, and 20 quantiles. Therefore, the numerator for the weights (the marginal probability of falling into category j) is calculated as $1/j$.

Following standard practice,[10] we explored the impact of truncating the normal, normal heteroscedastic, and gamma weights at three levels: 1st and 99th, 5th and 95th, and 10th and 90th percentiles. In total, 26 weighting schemes were considered, based on six modeling approaches: a normal distribution (with no truncation and truncation at the 1st and 99th, 5th and 95th, and 10th and 90th percentiles), a t distribution (with 1, 3, and 5 degrees of freedom), a gamma model (untruncated and truncated at previously specified percentiles), a normal model with heteroscedastic variance (untruncated and truncated at previously specified percentiles), a truncated normal model with heteroscedastic variance (untruncated and truncated at previously specified percentiles), and a cumulative logistic model for discretized outcomes (based on 10, 15, and 20 categories). Because the heteroscedastic exposure was characterized by an excess of zero values, we modified the above density functions to include a zero-inflated portion.[22]

We assessed the performance of each method using bias (defined as the average of the difference between the point estimate and the true value across Monte Carlo samples) and mean squared error (defined as squared bias plus the Monte Carlo variance of the estimate). Furthermore, we compare the distributions of the weights across all Monte Carlo samples. SAS version 9.3 (SAS Institute, Cary, NC) was used for all analyses. In the eAppendix (http://links.lww.com/EDE/A756), we provide SAS code illustrating how to construct inverse probability weights for a continuous exposure using all six approaches, as well as the SAS code used to conduct all simulations.

RESULTS

Table 1 shows descriptive statistics for the simulated data, including the four covariates sampled from the Québec birth file. These covariate statistic values can be used to replicate our simulation study.[23] Figure 1 shows histograms of the simulated exposures following a normal (panel A) and contaminated Poisson (panel B) distribution, with normal and nonparametric kernel density curves overlaid. In panel A, the kernel density curve and the normal density curve lay directly on top of one another. In panel B, the zero-inflated density portion is clearly seen at the *x*-axis origin. Figure 2 shows scatterplots of the residuals obtained from two confounder-adjusted ordinary least squares regression models for each simulated exposure. As expected, Figure 2A shows no pattern in the residuals, confirming a homoscedastic exposure distribution. Figure 2B depicts a marked fan shape opening to the right, confirming a heteroscedastic exposure distribution. Moreover, the set of data points below the fan-shaped portion of the plot reveals the zero-inflated nature of the heteroscedastic exposure points.

TABLE 1. Descriptive Statistics for the Simulation Data

Variable (Distribution)	Mean	Variance
Maternal age (normal)	29.84	21.60
Paternal age (normal)	32.52	30.45
Parity (Poisson)		
2	0.24	0.18
3	0.07	0.07
4	0.02	0.02
5+	0.02	0.02
X_1 (normal)	16.98	2.47
X_2 (Poisson)	2.06	2.76
Y_1 (Bernoulli)	0.08	0.07
Y_2 (Bernoulli)	0.09	0.08

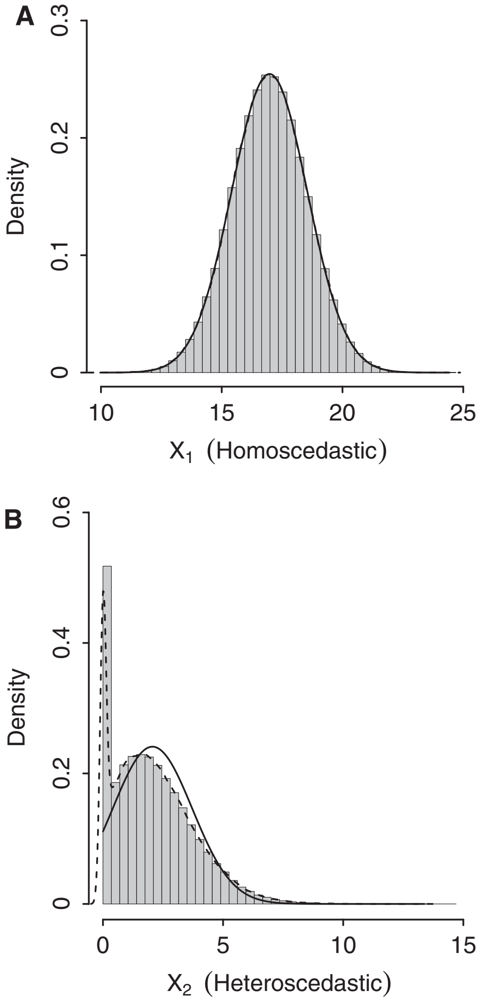

FIGURE 1. Histograms for both simulated exposures. A, normally distributed exposure (homoscedastic); B, contaminated Poisson exposure (heteroscedastic). Solid curve indicates normal density estimate; dashed curve, Kernel density estimate.

Table 2 describes the inverse probability weights for all 3000 Monte Carlo samples in our simulation study from each weighting scheme. With the exception of the weights from the heteroscedastic normal model for the heteroscedastic exposure, the mean of the weights obtained from all methods and both exposures was close to one. The mean of the weights from the heteroscedastic normal model for the heteroscedastic exposure was 1.56 (minimum = 0.00, maximum = 1.2×10^6). These extreme values were due to the influence of fewer than 1% of the data points in each Monte Carlo sample. Truncating the distribution of these heteroscedastic normal weights at the 1st and 99th percentile resulted in weights with a mean of 1.00 (minimum = 0.21, maximum = 4.01). For the heteroscedastic exposure, over all 4.5 million observations, 288 data points had heteroscedastic normal weights greater than the 25% of the Monte Carlo sample size ($1500\times0.25 = 375$). Of all other methods assessed, only the two-stage heteroscedastic normal

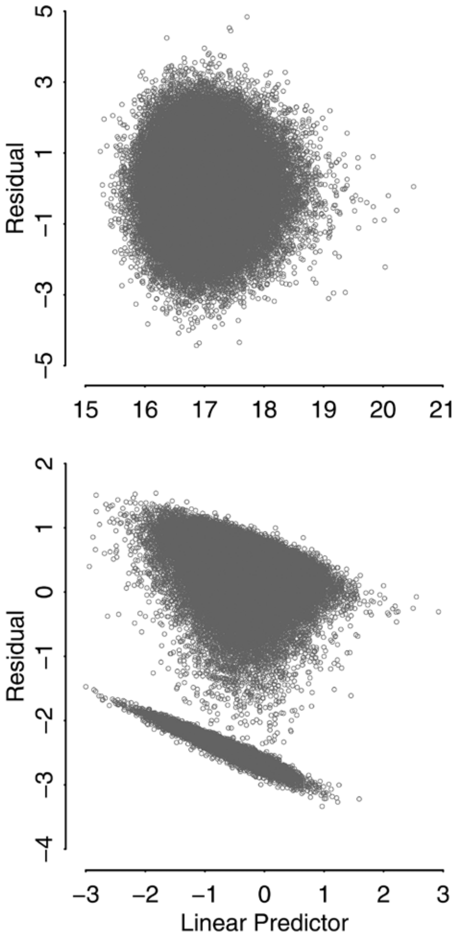

FIGURE 2. Standardized residual plots from ordinary least squares regression models conditional on all confounders for both simulated exposures. A, Homoscedastic exposure; B, heteroscedastic exposure.

approach yielded two observations (of all 4.5 million) with weights greater than 25% of the sample size, with values of 1,306.98 and 1,838.08.

Figure 3 shows the bias of various IPW methods from two weighted regression models for both simulated scenarios. For the heteroscedastic normal weights, we present results from the single-stage approach with weights truncated at the 1st and 99th percentile of the distribution. The eFigure (http://links.lww.com/EDE/A756) reproduces Figure 3 with all assessed methods included. For the normally distributed homoscedastic exposure, the standard normal and the heteroscedastic normal approaches were unbiased. All other methods were slightly biased, with the gamma and quantile binning approaches least so, as indicated by means (denoted by ×) that were closest to zero. For the heteroscedastic exposure, all methods were slightly biased, with the heteroscedastic normal (untruncated) and quantile binning approach (with 10, 15, and 20 categories) least so: the magnitude of bias for these approaches was 0.022, 0.011, 0.013, and 0.012, respectively.

TABLE 2. Distribution of Inverse Probability Weights for Both Simulated Exposures

	Homoscedastic	Heteroscedastic
Method	Mean (min, max)	Mean (min, max)
Normal	1.00 (0.00, 131.70)	1.01 (0.34, 10.13)
Weight truncation		
(1,99)	0.99 (0.24, 3.41)	1.00 (0.61, 2.22)
(5,95)	0.98 (0.49, 1.77)	1.00 (0.73, 1.50)
(10,90)	0.97 (0.61, 1.40)	1.00 (0.76, 1.30)
Heteroscedastic normal	1.00 (0.00, 64.94)	1.56 (0.00, 1.2 ×10^6)
t Distribution		
Degrees of freedom		
1	1.03 (0.14, 7.20)	1.01 (0.35, 3.26)
3	1.03 (0.07, 11.04)	1.01 (0.25, 3.82)
5	1.04 (0.04, 16.15)	1.01 (0.20, 4.14)
Gamma distribution	1.00 (0.00, 97.65)	0.99 (0.01, 41.20)
Quantile binning categories		
10	1.00 (0.11, 28.65)	1.00 (0.11, 16.41)
15	1.00 (0.08, 30.29)	0.98 (0.08, 264.83)
20	1.00 (0.06, 28.91)	0.95 (0.06, 74.58)

As expected, truncating the distribution of the weights at higher percentiles resulted in an increase in bias and a decrease in variance.[10] However, a decrease in the overall mean squared error was not observed with increasing truncation. As observed in Table 3, truncating the normal weights beyond the 1st and 99th percentile yielded no further decrease in mean squared error for either the homoscedastic or heteroscedastic exposures. The same was observed for the other heteroscedastic normal weights and gamma distribution weights.

DISCUSSION

We compared six methods for constructing inverse probability weights for a continuous exposure with simulated data generated from an empirical cohort. The first method assumed that the exposure followed a normal distribution with constant variance. This approach has been used in prior research to estimate the effect of epoetin dose on hematocrit levels,[24] and mortality in hemodialysis patients,[25] or the relation between loneliness and symptoms of depression.[26] We found that when the exposure followed a normal distribution with constant variance, this method was unbiased for the true exposure effect. When the exposure was heteroscedastic, using the homoscedastic normal approach resulted in a small bias.

We also used a heteroscedastic normal model to construct weights for both exposures. We assessed the performance of several techniques to implement this approach. The first was a two-stage approach previously implemented to assess the relation between the proportion of residents in a given neighborhood living in poverty and individual alcohol use.[13] This approach can easily be implemented with two regression

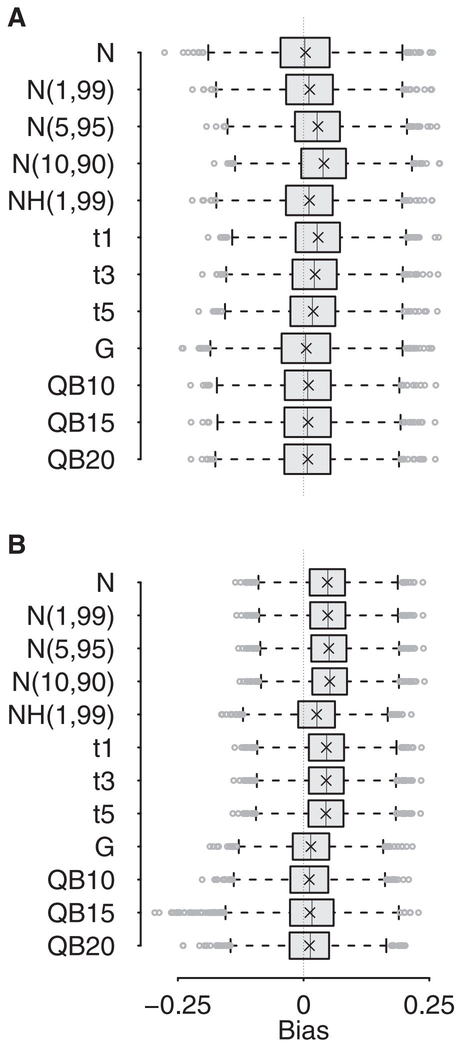

FIGURE 3. Bias of IPW methods from two marginal structural models in two simulated scenarios. A, Homoscedastic exposure; B, heteroscedastic exposure. N, standard normal distribution; N(1,99), truncated at 1st and 99th percentile; N(5,95), truncated at 5th and 95th percentile; N(10,90), truncated at 10th and 90th percentile; NH, normal distribution with heteroscedasticity; t1, *t* distribution with 1df; t3, *t* distribution with 3df; t5, *t* distribution with 5df; G, gamma distribution; QB10, quantile binning 10 categories; QB15, quantile binning 15 categories; QB20, quantile binning 20 categories; ×, mean value.

models: one for the conditional mean of the exposure, and a second for the mean of the squared residuals from the first model. The second was a single-stage approach based on joint maximization of the likelihood for the mean and variance models. This approach can be easily implemented using standard software (eg, SAS procedure) and is equivalent to the two-stage approach when the variance is independent of the mean function.[18] The third technique is based on constructing weights using a truncated normal distribution and joint maximization of the likelihood for the mean and variance models.[27(§12.4)] In our study, each of these methods yielded similar bias, and the major differences in mean squared error were due to very few influential observations with large weights.

TABLE 3. Mean Squared Error for Weighted Regression Model Parameters Obtained Using Different Inverse Probability–weighted Estimators for Two Simulated Exposures

	Homoscedastic	Heteroscedastic
Normal	0.0054	0.0049
Truncated		
(1,99)	0.0048	0.0050
(5,95)	0.0051	0.0052
(10,90)	0.0058	0.0053
Heteroscedastic normal	0.0054	0.0434
t Distribution		
Degrees of freedom		
1	0.0051	0.0047
3	0.0049	0.0047
5	0.0048	0.0047
Gamma distribution	0.0053	0.0033
Quantile binning categories		
10	0.0048	0.0032
15	0.0048	0.0052
20	0.0048	0.0036

Because the variance of a gamma random variable is defined as a function of the mean, the gamma distribution may, in principle, accommodate heteroscedasticity in the exposure. However, while the bias observed using the gamma distribution for the homoscedastic exposure was negligible, bias was more pronounced for the heteroscedastic distribution. One possible explanation lies in the fact that the gamma distribution approach requires a constant coefficient of variation (defined as the standard deviation of the exposure divided by the mean).[28 (p 285)] Indeed, further analyses revealed that the constant coefficient of variation assumption was met for the homoscedastic exposure but not for the heteroscedastic exposure. Thus, while the gamma distribution may not be generally useful to account for heteroscedasticity, it may be useful when the constant coefficient of variation assumption is met.

In principle, constructing inverse probability weights using the *t* distribution should mitigate the impact of outliers on the variation in the weights.[19,20(p 465)]Still, we found little difference between the approach using the *t* distribution and the standard normal approach after truncating the weights. This may be due to the fact that our analyses focused on the performance of methods under homoscedastic and heteroscedastic exposures. Thus, while outliers were present in the heteroscedastic exposure, they may not have been dramatic enough to observe important differences with the *t* distribution. Second, when dealing with outliers, it is possible that the benefit gained from truncating the distribution of the weights obtained from a standard or heteroscedastic normal approach is equal to that gained from using the *t* distribution. Future research could compare the performance of weight truncation, the *t* distribution, and other robust methods or mixture models that may better account for the presence of outliers.

Finally, quantile binning is a relatively simple approach that makes no assumptions about the distribution of the exposure. This approach first ranks the exposure into quantiles. For zero-inflated exposures, one can also create a separate category for the zero portion and rank the continuous portion into quantiles (in our study, both approaches yielded similar results). One then fits a multinomial or polytomous logistic regression model to these categories to obtain the predicted probabilities of falling into the observed exposure category. We used a cumulative logistic regression model, which compares the probability of being above a certain threshold to the probability of being below.[29(p 301)] We also used a baseline-category logistic regression model, which compares the probability of being in a certain category to being in a chosen referent category.[29(p 293)] Both methods yielded similar results.

Marginal structural models with a binary exposure are common in epidemiologic research. Westreich et al[30] explored the finite-sample properties of marginal structural Cox proportional hazards regression models with a binary exposure under various confounding scenarios and found their performance followed expectations based on asymptotic theory. For a binary exposure, Lee et al[31,32] compared the performance of machine learning algorithms (including standard, pruned, bagged, and boosted classification and regression trees, and random forests) to standard logistic regression (with and without truncating) to construct inverse probability weights in a simulation study. They found that boosted classification and regression trees were superior to other methods in a majority of the scenarios they explored. While continuous exposures are common in epidemiology, no studies have explored the performance of various methods to construct inverse probability weights for a continuous exposure.

Continuous exposures follow a wide variety of parametric forms, including complex zero-inflated and other mixture distributions. Examples include asbestos exposure in occupational epidemiology,[33] epoetin dose among elderly hemodialysis patients,[25] and occlusion therapy for the treatment of childhood amblyopia.[34] To obtain an asymptotically unbiased and fully efficient inverse probability–weighted estimator with such exposures would require the use of complex mixture models that are not readily fit using most standard software packages. Moreover, the presence of heteroscedasticity would require accounting for the variance of the exposure as a function of relevant covariates. The absence of a relatively straightforward approach to construct inverse probability weights for complex, possibly heteroscedastic, exposure distributions may preclude the use of marginal structural models or lead to biased estimation in such settings. We show that a simple quantile binning approach provides relatively unbiased effect estimates with an acceptable mean squared error under various conditions compared with other methods.

It is important to note that, for the heteroscedastic exposure, all methods were biased for the true causal effect. However, the magnitude of the observed bias is comparable to the known finite-sample bias that characterizes the Cox proportional hazards regression model. Johnson et al[35] found that the Cox partial likelihood estimator was biased up to 6% in finite samples. This is comparable with the maximum bias observe in our study (5.5% for the truncated normal approach with weights truncated at the 10th and 90th percentile, eFigure, http://links.lww.com/EDE/A756). For the quantile binning approach, the largest bias we observed was 1.3% for the heteroscedastic exposure with 15 categories.

Our estimand of interest is the marginal effect of a single-unit increase in the simulated exposures. To estimate this parameter, we stabilized the weights using only the marginal probability density (distribution) function of the exposure. We could have achieved additional stability in the weights by conditioning the numerator on a selection of confounders, and then adjusting for these confounders in our regression model for the outcome. However, this would have changed our estimand to a covariate-conditional effect.[36] For purposes of interpretation, unconfounded marginal effect estimates are arguably of greater public health relevance.[37] Moreover, they may also be used to plot weighted dose-response functions that avoid complications that can arise in more standard practices.

Our simulation analyses were limited, in that we did not explore the performance of each method under a range of causal effects, confounding patterns, and sample sizes. Moreover, we restricted our attention to the time-fixed exposure setting. As with all simulation research, our results apply to the scenarios explored.[38] We did, however, use empirical data sampled from the Québec birth file to generate our simulation data, and we assessed a wide range of approaches to construct inverse probability weights for a continuous exposure. We show how sensitive these estimates can be when the true exposure distribution is heteroscedastic, but when methods that assume homoscedasticity are used. We found that quantile binning is a simple way to obtain inverse probability weights for marginal structural models and performs well whether the exposure distribution is normal with constant variance or markedly nonnormal with nonconstant variance.

ACKNOWLEDGMENTS

We thank Richard MacLehose and two anonymous reviewers for helpful comments on previous versions of the manuscript. We thank Miguel Hernán for providing SAS code to fit the heteroscedastic truncated normal model.

REFERENCES

1. Sato T, Matsuyama Y. Marginal structural models as a tool for standardization. *Epidemiology*. 2003;14:680–686.
2. Robins JM. Structural nested failure time models. In: Andersen P, Keiding N, eds. *The Encyclopedia of Biostatistics*. Chichester: John Wiley and Sons; 1998;4372–4389.
3. Platt RW, Schisterman EF, Cole SR. Time-modified confounding. *Am J Epidemiol*. 2009;170:687–694.

4. Chiba Y, Azuma K, Okumura J. Marginal structural models for estimating effect modification. *Ann Epidemiol.* 2009;19:298–303.
5. Vanderweele TJ, Vansteelandt S, Robins JM. Marginal structural models for sufficient cause interactions. *Am J Epidemiol*. 2010;171:506–514.
6. Tchetgen Tchetgen EJ, VanderWeele TJ. On causal inference in the presence of interference. *Stat Methods Med Res*. 2012;21:55–75.
7. VanderWeele TJ. Marginal structural models for direct and indirect effects (erratum in Epidemiology 2009; 20(4):629). *Epidemiol*. 2009;20:18–26.
8. Hernán MA, Brumback B, Robins JM. Marginal structural models to estimate the causal effect of zidovudine on the survival of HIV-positive men. *Epidemiol*. 2000;11:561–570.
9. Robins JM, Hernán MA, Brumback B. Marginal structural models and causal inference in epidemiology. *Epidemiology*. 2000;11:550–560.
10. Cole SR, Hernán MA. Constructing inverse probability weights for marginal structural models. *Am J Epidemiol*. 2008;168:656–664.
11. Robins JM. Marginal structural models versus structural nested models as tools for causal inference. In: Halloran M, Berry D, eds. *Statistical Models in Epidemiology: The Environment and Clinical Trials*. Vol. 16. New York, NY: Springer-Verlag; 1999;95–134.
12. Hernán MA, Robins JM. Estimating causal effects from epidemiological data. *J Epidemiol Community Health*. 2006;60:578–586.
13. Cerdá M, Diez-Roux AV, Tchetgen ET, Gordon-Larsen P, Kiefe C. The relationship between neighborhood poverty and alcohol use: estimation by marginal structural models. *Epidemiology*. 2010;21:482–489.
14. Metropolis N, Ulam S. The Monte Carlo method. *J Am Stat Assoc*. 1949;44:335–341.
15. Auger N, Abrahamowicz M, Park AL, Wynant W. Extreme maternal education and preterm birth: time-to-event analysis of age and nativity-dependent risks. *Ann Epidemiol.* 2013;23:1–6.
16. Localio AR, Margolis DJ, Berlin JA. Relative risks and confidence intervals were easily computed indirectly from multivariable logistic regression. *J Clin Epidemiol*. 2007;60:874–882.
17. Liang KYL, Zeger SL. Longitudinal data analysis using generalized linear models. *Biometrika*. 1986;73:13–22.
18. Davidian M, Carroll RJ. Variance function estimation. *J Am Stat Assoc*. 1987;82:1079–1091.
19. Lange KL, Little RJA, Taylor JMG. Robust statistical modeling using the *t* distribution. *J Am Stat Assoc*. 1989;84:881–896.
20. Gelman A, Carlin J, Stern H, Rubin D. *Bayesian Data Analysis. Texts in Statistical Science*. Boca Raton, Fla: Chapman & Hall/CRC; 2004.
21. McCullagh P. Regression models for ordinal data. *J R Stat Soc B*. 1980;42:109–142.
22. Lambert D. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*. 1992;34:1–14.
23. Peng RD, Dominici F, Zeger SL. Reproducible epidemiologic research. *Am J Epidemiol*. 2006;163:783–789.
24. Cotter D, Zhang Y, Thamer M, Kaufman J, Hernán MA. The effect of epoetin dose on hematocrit. *Kidney Int*. 2008;73:347–353.
25. Zhang Y, Thamer M, Cotter D, Kaufman J, Hernán MA. Estimated effect of epoetin dosage on survival among elderly hemodialysis patients in the United States. *Clin J Am Soc Nephrol*. 2009;4:638–644.
26. VanderWeele TJ, Hawkley LC, Thisted RA, Cacioppo JT. A marginal structural model analysis for loneliness: implications for intervention trials and clinical practice. *J Consult Clin Psychol*. 2011;79:225–235.
27. Hernán MA, Robins J. *Causal Inference. Forthcoming*. Chapman/Hall, Available at: http://www.hsph.harvard.edu/miguel-hernan/causal-inference-book/. Accessed 25 June 2013.
28. McCullagh P, Nelder JA. *Generalized Linear Models*. 2nd ed. London; New York: Chapman and Hall; 2000.
29. Agresti A. *Categorical Data Analysis. Wiley Series in Probability and Statistics*. 3rd ed. Hoboken, NJ: Wiley; 2013.
30. Westreich D, Cole SR, Schisterman EF, Platt RW. A simulation study of finite-sample properties of marginal structural Cox proportional hazards models. *Stat Med*. 2012;31:2098–2109.
31. Lee BK, Lessler J, Stuart EA. Improving propensity score weighting using machine learning. *Stat Med*. 2010;29:337–346.
32. Lee BK, Lessler J, Stuart EA. Weight trimming and propensity score weighting. *PLoS One*. 2011;6:e18174.
33. Naimi AI, Cole SR, Hudgens MG, Brookhart MA, Richardson DB. Assessing the component associations of the healthy worker survivor bias: occupational asbestos exposure and lung cancer mortality. *Ann Epidemiol.* 2013;23:334–341.
34. Moodie EE, Stephens DA. Estimation of dose–response functions for longitudinal data using the generalised propensity score. *Stat Methods in Med Res*. 2012;21:149–166.
35. Johnson ME, Tolley HD, Bryson MC, Goldman AS. Covariate analysis of survival data: a small-sample study of Cox's model. *Biometrics*. 1982;38:685–698.
36. Kaufman JS. Marginalia: comparing adjusted effect measures. *Epidemiol*. 2010;21:490–493.
37. Robins JM, Greenland S, Hu FC. Estimation of the causal effect of a time-varying exposure on the marginal mean of a repeated binary outcome. *J Am Stat Assoc*. 1999;94:687–700.
38. Maldonado G, Greenland S. The importance of critically interpreting simulation studies. *Epidemiol*. 1997;8:453–456.

```
*IN THIS CODE, X = EXPOSURE, {C1-C3} ARE CONFOUNDERS;
*NORMAL, CONSTANT VARIANCE;
proc genmod data=a;
   model x = c1 c2 c3 / maxiter=100;
   *variance of x is sse/df = valuedf;
   ods output Modelfit=ss(where=(criterion="Deviance"));
   output out=x xbeta=xb;*predicted mean is xb;
data ss;set ss;merg=1;
data x;set x;merg=1;
data a1;
   merge x ss;
   by merg;
   sd_x=sqrt(valuedf);*standard deviation;
   pdf_den = pdf("normal",x,xb,sd_x);*denominator density value;
   drop criterion df value valuedf;
run;
*t, CONSTANT VARIANCE;
proc genmod data=a;
   model  x = c1 c2 c3 / maxiter=100;
   output out=x stdresdev=e;
data a1;
   set x;
   pdf_den = pdf("t",e,1);*denominator density value with 1 df;
run;
*NORMAL, HETEROSCEDASTIC VARIANCE, TRUNCATED NORMAL, HETEROSCEDASTIC VARIANCE;
*QLIM CODE PROVIDED BY DR MIGUEL HERNAN;
proc qlim data=a;
   *TO FIT TRUNCATED NORMAL, UNCOMMENT BELOW;
    model x =  c1 c2 c3 ;* / truncated(lb=5 ub=30) ;
    hetero x ~  c1 c2 c3  / link = exp noconst;
    nloptions  tech=newrap maxiter=1000 maxfunc = 2000;
    output out=x predicted  errstd;
data a1;
   set x;
   pdf_den = pdf("normal",x,p_x,errstd_x);
   *above gives normal heteroscedastic;
   p_n = pdf("normal",x,p_x,errstd_x) ;
   p_lb = cdf("normal",5,p_x,errstd_x);
   p_ub = cdf("normal",30,p_x,errstd_x);
   pdf_den_trunc = p_n / (p_lb - p_ub);
   *above gives truncated normal heteroscedastic;
run;
*GAMMA;
proc genmod data=a;
   model  x = c1 c2 c3 / dist=gamma link=log maxiter=100;
   ods output ParameterEstimates=s(where=(Parameter="Scale"));
   output out=x p=xb;
data s;set s;merg=1;rename estimate=scale;
drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;
data x; set x;merg=1;
data a1;
   merge x s;
   by merg;
   *determine Gamma shape parameter;
   lambda = xb/scale;
   *estimate denominator density;
   pdf_den = pdf("gamma",x,scale,lambda);
run;
*to obtain numerator density values for all of the above,
fit models without c1, c2, or c3 and repeat data step a1;
*QUANTILE BINNING;
proc rank data=a out=x ties=low groups=10;
var x;ranks x_r;
```

```
proc logistic data=x desc;
    model x_r = c1 c2 c3 / maxiter=100;
   output out=a1 predprobs=i;
run;quit;run;
data a1;
   set a1;
   array ii(*) ip_1-ip_10;
   *ip_1-ip_10 are predicted probabilities of being in category 1 to 10;
   do j=1 to 10;
      *set density to predicted probability of observed exposure category;
      if x_r = j then pdf_den=ii(j);
   end;
   drop j ip_1-ip_10;
run;quit;run;
*set numerator density value to 1/j;
*END;
```

N
N(1,99)
N(5,95)
N(10,90)
NHc
NHc(1,99)
NHc(5,95)
NHc(10,90)
NHh
NHh(1,99)
NHh(5,95)
NHh(10,90)
NHu
NHu(1,99)
NHu(5,95)
NHu(10,90)
t1
t3
t5
G
G(1,99)
G(5,95)
G(10,90)
QB10
QB15
QB20
(A)
N
N(1,99)
N(5,95)
N(10,90)
NHc
NHc(1,99)
NHc(5,95)
NHc(10,90)
NHh
NHh(1,99)
NHh(5,95)
NHh(10,90)
NHu
NHu(1,99)
NHu(5,95)
NHu(10,90)
t1
t3
t5
G
G(1,99)
G(5,95)
G(10,90)
QB10
QB15
QB20
(B)
−0.5
0
0.25
0.5
Bias

```
*******************************************************************************;
"CONSTRUCTING INVERSE PROBABILITY WEIGHTS FOR A CONTINUOUS EXPOUSRE"
**AUTHOR: ASHLEY ISAAC NAIMI (ashley.naimi@mcgill.ca)
**VERSION: 3.3
**PURPOSE: GENERATE SIMULATED DATA AND ASSESS DIFFERENT METHODS TO CONSTRUCT IPW
*******************************************************************************;
%inc "c:\ain\header.txt";
%inc "c:\ain\spline.txt";
proc printto log="Y:\Documents\Research\Papers\ConstructingWeights\20Jun13\ipw.sim2.log";run;
proc printto
print="Y:\Documents\Research\Papers\ConstructingWeights\20Jun13\ipw.sim2.lst";run;
*IMPORT QUEBEC BIRTH FILE DATA TO GENERATE SIMULATED DATA. KEEP ONLY MATERNAL AND PATERNAL
AGE AND PARITY;
proc import datafile="Y:\Documents\Research\Papers\ConstructingWeights\20Jun13\ipimoms.txt"
     out=moms
     dbms=dlm
     replace;
     delimiter='09'x;
run;
data moms;set moms;id = _N_;keep mage page parity id;run;
*SELECT 3000 IID RANDOM SAMPLES OF QUEBEC BIRTH FILE DATA TO GENERATE SIMULATED EXPOSURE AND
OUTCOME DATA;
proc surveyselect data=moms
                            out=a noprint
                            method=srs
                            reps=50
                            seed=675847
                            sampsize=1500;
run;
proc print data=a (obs=5);run;
*CREATE DISJOINT INDICATOR VARIALBES FOR PARITY;
data a;
       set a;
       array par(*) parity1-parity5;
       do pp = 1 to 5;
             par(pp) = (parity=pp);
       end;
run;
*GENERATE TWO SIMULATED EXPOSURES (x2a and x2c) AND OUTCOMES AS A FUNCTION OF QUEBEC BIRTH
FILE DATA COVARIATES;
data a;
       set a;
       call streaminit(789654);
                    r = rand("bernoulli",.5);
                    mu = .025*mage + .0025*page + .00125*mage*page
                           - .21*parity2 - .22*parity3 - .45*parity4 - .45*parity5;
                    x2a = rand("normal",15 + mu, 1.5);*r + exp(.95*rannor(987))*(1-r);*exp(mu
+ rand("normal",0,.25));*CONTAMINATED-NORMAL DISTRIBUTION: HOMOSCEDASTIC;
                    if x2a < 0 then x2a = 0;
                    jitter = rand("normal");
                    x2c = (rand("Poisson",mu) + jitter);*POISSON DISTRIBUTION:
HETEROSCEDASTIC WITH MEAN = VARIANCE;
                    if x2c < 0 then x2c = 0;
                    p2a = 1/(1+exp(-(-11.5 + log(1.25)*x2a + log(1.7)*sqrt(mage) +
log(1.5)*sqrt(page)
                                                                                +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5)));
                    y2a = rand("bernoulli",p2a);
                    p2c = 1/(1+exp(-(-8.05 + log(1.25)*x2c + log(1.7)*sqrt(mage) +
log(1.5)*sqrt(page)
                                                                                +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5)));
                    y2c = rand("bernoulli",p2c);
                    smage = sqrt(mage);
                    spage = sqrt(page);
                    logx2a = x2a;
                    llx2a = log(x2a);
                    logx2c = log(x2c+.001);
                    x2cind = (x2c>0);
                    pind = sum(of parity2-parity5);
                    cens=1;
                    magepage = mage*page;
rename replicate=h ;
label replicate = ' ' ;
run;
proc means data=a min max mean median q1 q3 var nmiss maxdec=3;
var x2a x2c logx2c y2a y2c mage page parity2-parity5;run;
*DETERMINE MARGINAL TRUE VALUE USING METHOD OF MARGINAL STANDARDIZATION: Localio AR, ET AL.
Journal of Clinical Epidemiology. 2007;
data check;
       set a;
       linear2anum = -11.5 + log(1.25)*17.984 + log(1.7)*sqrt(mage) + log(1.5)*sqrt(page) +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5;
       p2anum = 1/(1+exp(-(linear2anum)));
       linear2aden = -11.5 + log(1.25)*16.984 + log(1.7)*sqrt(mage) + log(1.5)*sqrt(page) +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5;
       p2aden = 1/(1+exp(-(linear2aden)));

       linear2cnum = -8.05 + log(1.25)*2.067 + log(1.7)*sqrt(mage) + log(1.5)*sqrt(page) +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5;
       p2cnum = 1/(1+exp(-(linear2cnum)));
       linear2cden = -8.05 + log(1.25)*1.067 + log(1.7)*sqrt(mage) + log(1.5)*sqrt(page) +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5;
       p2cden = 1/(1+exp(-(linear2cden)));
run;
proc means data=check noprint mean;
var p2anum p2aden;
var p2cnum p2cden;
output out=meancheck mean=p2anum p2aden p2cnum p2cden;
data meancheck;set meancheck;
truex2a = log((p2anum/(1-p2anum))/(p2aden/(1-p2aden)));truex2c = log((p2cnum/(1-
p2cnum))/(p2cden/(1-p2cden)));
merg = 1;
keep merg truex2a truex2c;
run;

*CONSTRUCTING IPWS;
*MODEL MARGINAL AND CONDITIONAL MEAN OF EXPOSURE TO CONSTRUCT IPW;

*FOR X2C MODEL ZERO-INFLATED PORTION;
ods select none;
proc logistic data=a;
       by h;
       title "Continuous Exposure Model: Numerator Zero Portion";
       model x2cind = / maxiter=100;
       output out=nx2cbin p=nx2cbin;
proc logistic data=a;
       by h;
       title "Continuous Exposure Model: Denominator Binary Portion";
       model x2cind = mage page mage*page parity2-parity5 / maxiter=100;
       output out=dx2cbin p=dx2cbin;

*NORMAL MODEL;
proc genmod data=a;
       by h;
       title "Continuous Exposure Model: Numerator";
       model logx2a = / maxiter=100;
       ods output Modelfit=ssnx2a(where=(criterion="Deviance"));
       output out=nx2aa xbeta=nxba stdresdev=nx2a;run;
data nx2a;
       merge nx2aa ssnx2a;
       by h;
       varnx2a=sqrt(valuedf);
       drop criterion df value valuedf;
run;
proc genmod data=a;
       by h;*where x2c>0;
       title "Continuous Exposure Model: Numerator";
       model logx2c = / maxiter=100;
       ods output Modelfit=ssnx2c(where=(criterion="Deviance"));
       output out=nx2cc xbeta=nxbc stdreschi=nx2c;
data nx2c;
       merge nx2cc ssnx2c;
       by h;
       varnx2c=sqrt(valuedf);
       drop criterion df value valuedf;
run;
proc genmod data=a;
       by h;
       title "Continuous Exposure Model: Denominator";
       model x2a = mage page mage*page parity2-parity5 / maxiter=100;
       ods output Modelfit=ssdx2a(where=(criterion="Deviance"));
       output out=dx2aa xbeta=dxba stdresdev=dx2a;
data dx2a;
       merge dx2aa ssdx2a;
       by h;
       vardx2a=sqrt(valuedf);
       drop criterion df value valuedf;
run;
proc genmod data=a;
       by h;*where x2c>0;
       title "Continuous Exposure Model: Denominator";
       model logx2c = mage page mage*page parity2-parity5 /  maxiter=100;
       ods output Modelfit=ssdx2c(where = (criterion = "Deviance"));
       output out=dx2cc xbeta=dxbc stdreschi=dx2c;run;
data dx2c;
       merge dx2cc ssnx2c;
       by h;
       vardx2c=sqrt(valuedf);
       drop criterion df value valuedf;
run;
ods select all;
proc plot data=dx2a;
title "Residual Plot A";
plot dx2a*dxba ;
proc plot data=dx2c;
title "Residual Plot C";
plot dx2c*dxbc ;
run;quit;run;

*NORMAL MODELS ACCOUNTING FOR HETEROSCEDASTICITY;
*METHOD IMPLEMENTED BY CERDA ET AL AS CITED IN MAIN TEXT;
ods select none;
proc reg data=a;by h;
       model x2a = mage page magepage parity2 parity3 parity4 parity5;
       output out=resid r=rr;
data resid;
       set resid;
       r2 = rr**2;
```

```
proc genmod data=resid;by h;
       model r2 = mage pind / dist=normal link=log noint;
       output out=x2avar p=x2avar;
proc reg data=a;by h;
       model logx2c = mage page magepage parity2 parity3 parity4 parity5;
       output out=resid r=rr;
data resid;
       set resid;
       r2 = rr**2;
proc genmod data=resid;by h;
       model r2 = mage pind / dist=normal link=log noint;
       output out=x2cvar p=x2cvar;*variance;
run;quit;run;

*METHOD IMPLEMENTED BY HERNAN AND ROBINS CAUSAL INFERENCE BOOK, CHAPTER 12 SECTION 4
(TRUNCATED NORMAL WITH HETEROSCEDASTIC VARIANCE);
proc qlim data=a ;
       by h;
    model x2a =   / truncated(lb=5  ub=30);
    hetero x2a ~    / link = exp noconst;
    nloptions  tech=newrap maxiter=1000 maxfunc = 2000;
    output out = numx2a_th predicted  errstd ;
data numx2a_th;set numx2a_th;rename p_x2a=Pn_x2a errstd_x2a=Errstdn_x2a;run;
proc qlim data=a ;
       by h;
    model x2a =  mage page magepage parity2 parity3 parity4 parity5 / truncated(lb=5  ub=30);
    hetero x2a ~  mage pind  / link = exp noconst;
    nloptions  tech=newrap maxiter=1000 maxfunc = 2000;
    output out = denx2a_th predicted  errstd ;
data denx2a_th;set denx2a_th;rename p_x2a=Pd_x2a errstd_x2a=Errstdd_x2a;
proc qlim data=a ;
       by h;
    model logx2c =   / truncated(lb=-7.5 ub=3.5);
    hetero logx2c ~    / link = exp noconst;
    nloptions  tech=newrap maxiter=1000 maxfunc = 2000;
    output out = numx2c_th predicted  errstd ;
data numx2c_th;set numx2c_th;rename p_logx2c=Pn_x2c errstd_logx2c=Errstdn_x2c;run;
proc qlim data=a ;
       by h;
    model logx2c =  mage page magepage parity2 parity3 parity4 parity5 / truncated(lb=-7.5
ub=3.5);
    hetero logx2c ~  mage pind  / link = exp noconst;
    nloptions tech=newrap maxiter=1000 maxfunc = 2000;
    output out = denx2c_th predicted  errstd ;
data denx2c_th;set denx2c_th;rename p_logx2c=Pd_x2c errstd_logx2c=Errstdd_x2c;
run;quit;run;

*USER WRITTEN CODE FOR HETERO NORMAL;
proc nlmixed data=a tech=NRRIDG maxiter=1000 maxfunc=2000;
       title "Numerator Model";
       by h;
       parms b0=0 sd0=1;
       bounds sd0 > 0;
       *MEAN;
       mu = b0;
       *CONSTANT VARIANCE;
       sigma_sq = sd0;
       sigma = sqrt(sigma_sq);
       model x2a ~ normal(mu,sigma_sq);
       predict mu  out=bbnx2amu;
       predict sigma out=bbnx2asig;
proc nlmixed data=a tech=NRRIDG maxiter=1000 maxfunc=2000;
       title "Denominator Model";
       by h;
       parms b0=0 b1=0 b2=0 b3=0 b4=0 b5=0 b6=0 b7=0 sigma_sq=1 sd1=1 sd2=1;
       bounds sigma_sq > 0, sd1>0, sd2>0;
       *MEAN;
       mu = b0 + b1*mage + b2*page + b3*mage*page + b4*parity2 + b5*parity3 + b6*parity4 +
b7*parity5;
       *HETEROSCEDASTIC VARIANCE FUNCTION;
       *See, e.g., Davidian and Carroll (1987) JASA Col 82 Issue 400 for justification of
equation;
       sigma_i_sq = log(sigma_sq) + sd1*mage + sd2*pind;
       sigma_i = sqrt(sigma_i_sq);
       model x2a ~ normal(mu,sigma_i_sq);
       predict mu  out=bbdx2amu;
       predict sigma_i out=bbdx2asig;
run;quit;run;
data bbnx2amu;set bbnx2amu;rename pred=nx2amu;run;
data bbnx2asig;set bbnx2asig;rename pred=nx2asig;run;
data bbdx2amu;set bbdx2amu;rename pred=dx2amu;run;
data bbdx2asig;set bbdx2asig;rename pred=dx2asig;run;
proc nlmixed data=a tech=nrridg maxiter=1000 maxfunc=2000;
       by h;
       parms b0=0 g0=0 sd0=1;
       bounds sd0 > 0;
       *CONTINUOUS MEAN PORTION;
       mu = b0;
       *ZERO INFLATED MEAN PORTION;
       pi = 1/(1+exp(-(g0)));
       *CONSTANT VARIANCE;
       sigma_sq = sd0;
       sigma = sqrt(sigma_sq);
       ll = (x2c=0)*(log(pi)) + (x2c>0)*((log(1-pi))+log(pdf('normal',logx2c,mu,sigma)));
       model logx2c ~ general(ll);
       predict mu  out=bbnx2cmu;
       predict pi out=bbnx2cpi;
       predict sigma out=bbnx2csig;run;
proc nlmixed data=a tech=nrridg maxiter=1000 maxfunc=2000;
       by h;
       parms b0=0 b1=0 b2=0 b3=0 b4=0 b5=0 b6=0 b7=0
                g0=0 g1=0  g2=0
                sigma_sq=1 sd1=0 sd2=0;
       bounds sigma_sq > 0;
       *CONTINUOUS MEAN PORTION;
       mu = b0 + b1*mage + b2*page + b3*mage*page + b4*parity2 + b5*parity3 + b6*parity4 +
b7*parity5;
       *ZERO INFLATED MEAN PORTION;
       pi = 1/(1+exp(-(g0 + g1*mage + g2*pind)));
       *HETEROSCEDASTIC VARIANCE;
       log_sigma_i_sq = log(sigma_sq) + sd1*mage + sd2*pind;
       sigma_i = sqrt(exp(log_sigma_i_sq));
       ll = (x2c=0)*log(pi) + (x2c>0)*((log(1-pi))+log(pdf('normal',logx2c,mu,sigma_i)));
       model logx2c ~ general(ll);
       predict mu  out=bbdx2cmu;
       predict pi out=bbdx2cpi;
       predict sigma_i out=bbdx2csig;
run;quit;run;
data bbnx2cmu;set bbnx2cmu;rename pred=nx2cmu;keep h id pred;run;
data bbnx2cpi;set bbnx2cpi;rename pred=nx2cpi;keep h id pred;run;
data bbnx2csig;set bbnx2csig;rename pred=nx2csig;keep h id pred;run;
data bbdx2cmu;set bbdx2cmu;rename pred=dx2cmu;keep h id pred;run;
data bbdx2cpi;set bbdx2cpi;rename pred=dx2cpi;keep h id pred;run;
data bbdx2csig;set bbdx2csig;rename pred=dx2csig;keep h id pred;run;

*GAMMA MODEL;
```

```
proc genmod data=a ;
       by h;
       title "Continuous Exposure Model: Numerator";
       model x2a = / link=log dist=gamma maxiter=100;
       ods output ParameterEstimates=ngx2aa1(where=(Parameter="Scale"));
       output out=ngx2aa2 p=ngx2a;run;
data ngx2a;
       merge ngx2aa1 ngx2aa2;
       by h;
       ngx2ascale = estimate;
       drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;

proc genmod data=a ;
       where x2c > 0;
       by h;
       title "Continuous Exposure Model: Numerator Continuous Portion";
       model x2c = / link=log dist=gamma maxiter=100;
       ods output ParameterEstimates=ngx2cc1(where=(Parameter="Scale"));
       output out=ngx2cc2 p=ngx2c;
data ngx2c;
       merge ngx2cc1 ngx2cc2;
       by h;
       ngx2cscale = estimate;
       drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;
proc genmod data=a ;
       by h;
       title "Continuous Exposure Model: Denominator";
       model x2a = mage page mage*page parity2-parity5 / link=log dist=gamma maxiter=100;
       ods output ParameterEstimates=dgx2aa1(where=(Parameter="Scale"));
       output out=dgx2aa2 p=dgx2a;
data dgx2a;
       merge dgx2aa1 dgx2aa2;
       by h;
       dgx2ascale = estimate;
       drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;

proc genmod data=a ;
       where x2c > 0;
       by h;
       title "Continuous Exposure Model: Denominator Continuous Portion";
       model x2c = mage page mage*page parity2-parity5 / link=log dist=gamma maxiter=100;
       ods output ParameterEstimates=dgx2cc1(where=(Parameter="Scale"));
       output out=dgx2cc2 p=dgx2c;run;
data dgx2c;
       merge dgx2cc1 dgx2cc2;
       by h;
       dgx2cscale = estimate;
       drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;
run;quit;run;

*CHECK IF CV IS CONSTANT ACROSS MU (GAMMA ASSUMPTION);
data dx2a;
       set dx2a;
       rr = dx2a**2;
proc genmod data=dx2a;
model rr = dxba / link=log dist=normal;
output out=cv p=sigma;
run;quit;run;
data cv;
       set cv;
       sd = sigma**(1/2);
       cv = sd/dxba;
ods select all;
```

```
proc plot data=cv;
plot cv*dxba;
run;quit;run;
ods select none;
proc reg data=cv;
model cv = dxba;
run;quit;run;

proc genmod data=a;
       where x2c>0;
       model x2c = mage page mage*page parity2-parity5 /  maxiter=100;
       output out=cv xbeta=dxbc stdreschi=dx2c;run;
data cv;
       set cv;
       rr = dx2c**2;
proc genmod data=cv;
where x2c>0;
model rr = dxbc / link=log dist=normal;
output out=cv p=sigma;
run;quit;run;
data cv;
       set cv;
       sd = sigma**(1/2);
       cv = sd/(dxbc);
run;
ods select all;
proc plot data=cv;
plot cv*dxbc;
run;quit;run;
proc reg data=cv;
model cv = dxbc;
run;quit;run;

*QUANTILE BINNING APPROACH;
*RANK CONTINUOUS EXPOSURES INTO CATEGORIES;
proc rank data=a out=cc(keep= h id x2a x2c x2arank10 x2crank10) ties=low groups=10;
by h;var x2a x2c;ranks x2arank10 x2crank10;run;
proc rank data=cc out=cc ties=low groups=15;
by h;var x2a x2c;ranks x2arank15 x2crank15;run;
proc rank data=cc out=cc ties=low groups=20;
by h;var x2a x2c;ranks x2arank20 x2crank20;run;
proc sort data=a;by h id;
proc sort data=cc;by h id;
data bb;merge a cc;by h id;
data bb;set bb;
array catrank(6) x2arank10 x2crank10 x2arank15 x2crank15 x2arank20 x2crank20;
do jj = 1 to 6;
       catrank(jj) = catrank(jj)+1;
end;
run;
*FIT CUMULATIVE LOGISTIC MODELS TO RANKED EXPOSURE VALUES;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 10";
       model x2arank10 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa1 predprobs=i;
run;quit;run;
data dcatx2a10;
       set aa1;
       array ii(*) ip_1-ip_10;
       do kappa=1 to 10;
              if x2arank10 = kappa then dencata10=ii(kappa);
       end;
       drop kappa ip_1-ip_10;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 10";
       model x2crank10 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa3 predprobs=i;
run;quit;run;
data dcatx2c10;
       set aa3;
       array ii(*) ip_1-ip_10;
       do kappa=1 to 10;
              if x2crank10 = kappa then dencatc10=ii(kappa);
       end;
       drop kappa ip_1-ip_10;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 15";
       model x2arank15 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa1 predprobs=i;
run;quit;run;
data dcatx2a15;
       set aa1;
       array ii(*) ip_1-ip_15;
       do kappa=1 to 15;
              if x2arank15 = kappa then dencata15=ii(kappa);
       end;
       drop kappa ip_1-ip_15;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 15";
       model x2crank15 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa3 predprobs=i;
run;quit;run;
data dcatx2c15;
       set aa3;
       array ii(*) ip_1-ip_15;
       do kappa=1 to 15;
              if x2crank15 = kappa then dencatc15=ii(kappa);
       end;
       drop kappa ip_1-ip_15;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 20";
       model x2arank20 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa1 predprobs=i;
run;quit;run;
data dcatx2a20;
       set aa1;
       array ii(*) ip_1-ip_20;
       do kappa=1 to 20;
              if x2arank20 = kappa then dencata20=ii(kappa);
       end;
       drop kappa ip_1-ip_20;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 20";
       model x2crank20 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa3 predprobs=i;
```

```
run;quit;run;
data dcatx2c20;
       set aa3;
       array ii(*) ip_1-ip_20;
       do kappa=1 to 20;
              if x2crank20 = kappa then dencatc20=ii(kappa);
       end;
       drop kappa ip_1-ip_20;
run;
*CREATE STABILIZED IPWs;
data b;
       merge a nx2cbin dx2cbin nx2a nx2c dx2a dx2c ngx2a ngx2c dgx2a dgx2c x2avar x2cvar
                numx2a_th denx2a_th numx2c_th denx2c_th dcatx2a10 dcatx2a15 dcatx2a20
dcatx2c10 dcatx2c15 dcatx2c20
                bbnx2amu bbnx2asig bbdx2amu bbdx2asig
                bbnx2cmu bbnx2cpi bbnx2csig bbdx2cmu bbdx2cpi bbdx2csig;
       by h id;

       *NORMAL;
       *x2a;
       numx2a=pdf("normal",x2a,nxba,varnx2a);
       denx2a=pdf("normal",x2a,dxba,vardx2a);

       *x2c;
       *ZERO INFLATED PORTION;
       if x2cind = 0 then do;
              numx2c=nx2cbin + (1-nx2cbin)*pdf("normal",0,nxbc,varnx2c);
              denx2c=dx2cbin + (1-dx2cbin)*pdf("normal",0,dxbc,vardx2c);
       end;
       *CONTINUOUS PORTION;
       if x2cind = 1 then do;
              numx2c=(1-nx2cbin)*pdf("normal",logx2c,nxbc,varnx2c);
              denx2c=(1-dx2cbin)*pdf("normal",logx2c,dxbc,vardx2c);
       end;

       *GAMMA;
       *x2a;
       *DETERMINE GAMMA SHAPE PARAMETER  (LAMBDA);
       ngx2alambda = ngx2a/ngx2ascale;
       dgx2alambda = dgx2a/dgx2ascale;

       numgx2a = pdf("gamma",x2a,ngx2ascale,ngx2alambda);
       dengx2a = pdf("gamma",x2a,dgx2ascale,dgx2alambda);

       *x2c;
       *DETERMINE GAMMA SHAPE PARAMETER (LAMBDA);
       *ZERO INFLATED PORTION;
       if x2cind = 0 then do;
              numgx2c = nx2cbin;*b/c gamma pdf defined as zero when x2c = 0, no continuous
portion;
              dengx2c = dx2cbin;
       end;
       *CONTINUOUS PORTION;
       if x2cind = 1 then do;
              ngx2clambda = ngx2c/ngx2cscale;
              dgx2clambda = dgx2c/dgx2cscale;
              numgx2c = (1-nx2cbin)*pdf("gamma",x2c,ngx2cscale,ngx2clambda);
              dengx2c = (1-dx2cbin)*pdf("gamma",x2c,dgx2cscale,dgx2clambda);
       end;

       *NORMAL HETEROSCEDASTIC: USING CERDA ET AL'S METHOD;
       *x2a;
       numx2ah=pdf("normal",x2a,nxba,varnx2a);
       denx2ah=pdf("normal",x2a,dxba,sqrt(x2avar));

       *x2c;
       *ZERO INFLATED PORTION;
       if x2cind = 0 then do;
              numx2ch=nx2cbin;* + (1-nx2cbin)*pdf("normal",0,nxbc,varnx2c);
              denx2ch=dx2cbin;* + (1-dx2cbin)*pdf("normal",0,dxbc,x2cvar);
       end;
       *CONTINUOUS PORTION;
       if x2cind = 1 then do;
              numx2ch=(1-nx2cbin)*pdf("normal",logx2c,nxbc,varnx2c);
              denx2ch=(1-dx2cbin)*pdf("normal",logx2c,dxbc,sqrt(x2cvar));
       end;

       *NORMAL HETEROSCEDASTIC: USING HERNAN & ROBINS' METHOD;
       *x2a;
       nx2a_th = pdf("normal",x2a,Pn_x2a,Errstdn_x2a);
       ubnx2a_th = cdf("normal",30, Pn_x2a, Errstdn_x2a);
       lbnx2a_th = cdf("normal",5, Pn_x2a, Errstdn_x2a) ;

       numx2a_th = nx2a_th / (ubnx2a_th - lbnx2a_th);

       dx2a_th = pdf("normal",x2a,Pd_x2a,Errstdd_x2a);
       ubdx2a_th = cdf("normal",30, Pd_x2a, Errstdd_x2a);
       lbdx2a_th = cdf("normal",5, Pd_x2a, Errstdd_x2a) ;

       denx2a_th = dx2a_th / (ubdx2a_th - lbdx2a_th);

       *x2c;
       *ZERO INFLATED PORTION;
       if x2cind = 0 then do;
              numx2c_th=nx2cbin + (1-
nx2cbin)*(pdf("normal",logx2c,pn_x2c,errstdn_x2c)/(cdf("normal",3.5,pd_x2c,errstdd_x2c)-
cdf("normal",-7.5,pn_x2c,errstdn_x2c)));
              denx2c_th=dx2cbin + (1-
dx2cbin)*(pdf("normal",logx2c,pd_x2c,errstdd_x2c)/(cdf("normal",3.5,pd_x2c,errstdd_x2c)-
cdf("normal",-7.5,pd_x2c,errstdd_x2c)));
       end;
       *CONTINUOUS PORTION;
       if x2cind = 1 then do;
              numx2c_th=(1-
nx2cbin)*(pdf("normal",logx2c,pn_x2c,errstdn_x2c)/(cdf("normal",3.5,pd_x2c,errstdd_x2c)-
cdf("normal",-7.5,pn_x2c,errstdn_x2c)));
              denx2c_th=(1-
dx2cbin)*(pdf("normal",logx2c,pd_x2c,errstdd_x2c)/(cdf("normal",3.5,pd_x2c,errstdd_x2c)-
cdf("normal",-7.5,pd_x2c,errstdd_x2c)));
       end;

       *NORMAL HETEROSCEDASTIC: USER WRITTEN;
       *x2a;
       numx2ahu=pdf("normal",x2a,nx2amu,nx2asig);
       denx2ahu=pdf("normal",x2a,dx2amu,dx2asig);

       *x2c;
       *ZERO PORTION;
       if x2c = 0 then do;
              numx2chu = nx2cpi;
              denx2chu = dx2cpi;
       end;
       *CONTINUOUS PORTION;
       if x2c > 0 then do;
              numx2chu = (1-nx2cpi)*pdf("normal",logx2c,nx2cmu,nx2csig);
              denx2chu = (1-dx2cpi)*pdf("normal",logx2c,dx2cmu,dx2csig);
       end;

       *T DISTRIBUTED;
       numt1x2a = pdf("T",nx2a,1);
       numt1x2c = pdf("T",nx2c,1);
       dent1x2a = pdf("T",dx2a,1);
       dent1x2c = pdf("T",dx2c,1);

       numt3x2a = pdf("T",nx2a,3);
       numt3x2c = pdf("T",nx2c,3);
       dent3x2a = pdf("T",dx2a,3);
       dent3x2c = pdf("T",dx2c,3);

       numt5x2a = pdf("T",nx2a,5);
       numt5x2c = pdf("T",nx2c,5);
       dent5x2a = pdf("T",dx2a,5);
       dent5x2c = pdf("T",dx2c,5);

       *STABILIZED WEIGHTS;
       *NORMAL;
       swx2a=numx2a/denx2a;
       swx2c=numx2c/denx2c;

       *HETEROSCEDASTIC NORMAL;
       swx2ah=numx2ah/denx2ah;
       swx2ch=numx2ch/denx2ch;

       *TRUNCATED HETEROSCEDASTIC NORMAL;
       swx2a_th = numx2a_th/denx2a_th;
       swx2c_th = numx2c_th/denx2c_th;

       *USER WRITTEN HETERO NORMAL;
       swx2ahu = numx2ahu/denx2ahu;
       swx2chu = numx2chu/denx2chu;

       *GAMMA;
       swgx2a=numgx2a/dengx2a;
       swgx2c=numgx2c/dengx2c;

       *T;
       swt1x2a=numt1x2a/dent1x2a;
       swt1x2c=numt1x2c/dent1x2c;

       swt3x2a=numt3x2a/dent3x2a;
       swt3x2c=numt3x2c/dent3x2c;

       swt5x2a=numt5x2a/dent5x2a;
       swt5x2c=numt5x2c/dent5x2c;

       *QUANTILE BINNING APPROACH;
       *BECAUSE QUANTILES ARE USED, NUMERATOR PROBABILITIES ARE SIMPLY 1/#QUANTILES;
       swcatx2a10 = (1/10)/dencata10;
       swcatx2a15 = (1/15)/dencata15;
       swcatx2a20 = (1/20)/dencata20;

       swcatx2c10 = (1/10)/dencatc10;
       swcatx2c15 = (1/15)/dencatc15;
       swcatx2c20 = (1/20)/dencatc20;

       merg=1;
run;
*FIND TRUNCATION VALUES;
proc univariate data=b noprint;by h;var swx2a swx2c swx2ah swx2ch swx2ahu swx2chu swx2a_th
```

```
swx2c_th swgx2a swgx2c;
output out=ptile p1 =swx2a1 swx2c1 swx2ah1 swx2ch1 swx2ahu1 swx2chu1 swx2a_th1 swx2c_th1
swgx2a1 swgx2c1
                         p99=swx2a99 swx2c99 swx2ah99 swx2ch99 swx2ahu99 swx2chu99 swx2a_th99
swx2c_th99 swgx2a99 swgx2c99
                            p5 =swx2a5 swx2c5 swx2ah5 swx2ch5 swx2ahu5 swx2chu5 swx2a_th5
swx2c_th5 swgx2a5 swgx2c5
                            p95=swx2a95 swx2c95 swx2ah95 swx2ch95 swx2ahu95 swx2chu95
swx2a_th95 swx2c_th95 swgx2a95 swgx2c95
                            p10=swx2a10 swx2c10 swx2ah10 swx2ch10 swx2ahu10 swx2chu10
swx2a_th10 swx2c_th10 swgx2a10 swgx2c10
                            p90=swx2a90 swx2c90 swx2ah90 swx2ch90 swx2ahu90 swx2chu90
swx2a_th90 swx2c_th90 swgx2a90 swgx2c90;
run;
data ptile;set ptile;merg=1;
*TRUNCATE STABILIZED WEIGHTS FROM NORMAL AND GAMMA MODELS;
data cc;merge b ptile;by h merg;
*NORMAL X2A AND X2C;
swx2a199 = swx2a;
if swx2a < swx2a1 then swx2a199 = swx2a1;
if swx2a > swx2a99 then swx2a199 = swx2a99;

swx2c199 = swx2c;
if swx2c < swx2c1 then swx2c199 = swx2c1;
if swx2c > swx2c99 then swx2c199 = swx2c99;

swx2a595 = swx2a;
if swx2a < swx2a5 then swx2a595 = swx2a5;
if swx2a > swx2a95 then swx2a595 = swx2a95;

swx2c595 = swx2c;
if swx2c < swx2c5 then swx2c595 = swx2c5;
if swx2c > swx2c95 then swx2c595 = swx2c95;

swx2a1090 = swx2a;
if swx2a < swx2a10 then swx2a1090 = swx2a10;
if swx2a > swx2a90 then swx2a1090 = swx2a90;

swx2c1090 = swx2c;
if swx2c < swx2c10 then swx2c1090 = swx2c10;
if swx2c > swx2c90 then swx2c1090 = swx2c90;

*NORMAL HETEROSCEDASTIC X2A AND X2C;
swx2ah199 = swx2ah;
if swx2ah < swx2ah1 then swx2ah199 = swx2ah1;
if swx2ah > swx2ah99 then swx2ah199 = swx2ah99;

swx2ch199 = swx2ch;
if swx2ch < swx2ch1 then swx2ch199 = swx2ch1;
if swx2ch > swx2ch99 then swx2ch199 = swx2ch99;

swx2ah595 = swx2ah;
if swx2ah < swx2ah5 then swx2ah595 = swx2ah5;
if swx2ah > swx2ah95 then swx2ah595 = swx2ah95;

swx2ch595 = swx2ch;
if swx2ch < swx2ch5 then swx2ch595 = swx2ch5;
if swx2ch > swx2ch95 then swx2ch595 = swx2ch95;

swx2ah1090 = swx2ah;
if swx2ah < swx2ah10 then swx2ah1090 = swx2ah10;
if swx2ah > swx2ah90 then swx2ah1090 = swx2ah90;
```

```
swx2ch1090 = swx2ch;
if swx2ch < swx2ch10 then swx2ch1090 = swx2ch10;
if swx2ch > swx2ch90 then swx2ch1090 = swx2ch90;

*TRUNCATED NORMAL HETEROSCEDASTIC X2A AND X2C;
swx2a_th199 = swx2a_th;
if swx2a_th < swx2a_th1 then swx2a_th199 = swx2a_th1;
if swx2a_th > swx2a_th99 then swx2a_th199 = swx2a_th99;

swx2c_th199 = swx2c_th;
if swx2c_th < swx2c_th1 then swx2c_th199 = swx2c_th1;
if swx2c_th > swx2c_th99 then swx2c_th199 = swx2c_th99;

swx2a_th595 = swx2a_th;
if swx2a_th < swx2a_th5 then swx2a_th595 = swx2a_th5;
if swx2a_th > swx2a_th95 then swx2a_th595 = swx2a_th95;

swx2c_th595 = swx2c_th;
if swx2c_th < swx2c_th5 then swx2c_th595 = swx2c_th5;
if swx2c_th > swx2c_th95 then swx2c_th595 = swx2c_th95;

swx2a_th1090 = swx2a_th;
if swx2a_th < swx2a_th10 then swx2a_th1090 = swx2a_th10;
if swx2a_th > swx2a_th90 then swx2a_th1090 = swx2a_th90;

swx2c_th1090 = swx2c_th;
if swx2c_th < swx2c_th10 then swx2c_th1090 = swx2c_th10;
if swx2c_th > swx2c_th90 then swx2c_th1090 = swx2c_th90;

*USER WRITTEN NORMAL HETEROSCEDASTIC X2A AND X2C;
swx2ahu199 = swx2ahu;
if swx2ahu < swx2ahu1 thuen swx2ahu199 = swx2ahu1;
if swx2ahu > swx2ahu99 thuen swx2ahu199 = swx2ahu99;

swx2chu199 = swx2chu;
if swx2chu < swx2chu1 then swx2chu199 = swx2chu1;
if swx2chu > swx2chu99 then swx2chu199 = swx2chu99;

swx2ahu595 = swx2ahu;
if swx2ahu < swx2ahu5 then swx2ahu595 = swx2ahu5;
if swx2ahu > swx2ahu95 then swx2ahu595 = swx2ahu95;

swx2chu595 = swx2chu;
if swx2chu < swx2chu5 then swx2chu595 = swx2chu5;
if swx2chu > swx2chu95 then swx2chu595 = swx2chu95;

swx2ahu1090 = swx2ahu;
if swx2ahu < swx2ahu10 then swx2ahu1090 = swx2ahu10;
if swx2ahu > swx2ahu90 then swx2ahu1090 = swx2ahu90;

swx2chu1090 = swx2chu;
if swx2chu < swx2chu10 then swx2chu1090 = swx2chu10;
if swx2chu > swx2chu90 then swx2chu1090 = swx2chu90;

*GAMMA X2A AND X2C;

swgx2a199 = swgx2a;
if swgx2a < swgx2a1 then swgx2a199 = swgx2a1;
if swgx2a > swgx2a99 then swgx2a199 = swgx2a99;

swgx2a595 = swgx2a;
if swgx2a < swgx2a5 then swgx2a595 = swgx2a5;
if swgx2a > swgx2a95 then swgx2a595 = swgx2a95;
```

```
swgx2a1090 = swgx2a;
if swgx2a < swgx2a10 then swgx2a1090 = swgx2a10;
if swgx2a > swgx2a90 then swgx2a1090 = swgx2a90;

swgx2c199 = swgx2c;
if swgx2c < swgx2c1 then swgx2c199 = swgx2c1;
if swgx2c > swgx2c99 then swgx2c199 = swgx2c99;

swgx2c595 = swgx2c;
if swgx2c < swgx2c5 then swgx2c595 = swgx2c5;
if swgx2c > swgx2c95 then swgx2c595 = swgx2c95;

swgx2c1090 = swgx2c;
if swgx2c < swgx2c10 then swgx2c1090 = swgx2c10;
if swgx2c > swgx2c90 then swgx2c1090 = swgx2c90;

run;

data c;set cc;keep
       swx2a swx2c
       swx2ah swx2ch
       swx2ahu swx2chu
       swx2a_th swx2c_th

       swx2a199 swx2a595 swx2a1090
       swx2c199 swx2c595 swx2c1090
       swx2ah199 swx2ah595 swx2ah1090
       swx2ch199 swx2ch595 swx2ch1090
       swx2ahu199 swx2ahu595 swx2ahu1090
       swx2chu199 swx2chu595 swx2chu1090
       swx2a_th199 swx2a_th595 swx2a_th1090
       swx2c_th199 swx2c_th595 swx2c_th1090

       swgx2a swgx2c
       swgx2a199 swgx2a595 swgx2a1090
       swgx2c199 swgx2c595 swgx2c1090

       swt1x2a swt1x2c
       swt3x2a swt3x2c
       swt5x2a swt5x2c

       swcatx2a10 swcatx2c10
       swcatx2a15 swcatx2c15
       swcatx2a20 swcatx2c20

       x2a y2a x2c y2c h id ;run;
ods select all;
proc means data=c min max mean nmiss maxdec=2;
title "Distribution of Weights";
var
       swx2a swx2c
       swx2ah swx2ch
       swx2ahu swx2chu
       swx2a_th swx2c_th

       swx2a199 swx2a595 swx2a1090
       swx2c199 swx2c595 swx2c1090
       swx2ah199 swx2ah595 swx2ah1090
       swx2ch199 swx2ch595 swx2ch1090
       swx2ahu199 swx2ahu595 swx2ahu1090
       swx2chu199 swx2chu595 swx2chu1090
       swx2a_th199 swx2a_th595 swx2a_th1090
       swx2c_th199 swx2c_th595 swx2c_th1090

       swgx2a swgx2c
       swgx2a199 swgx2a595 swgx2a1090
       swgx2c199 swgx2c595 swgx2c1090

       swt1x2a swt1x2c
       swt3x2a swt3x2c
       swt5x2a swt5x2c

       swcatx2a10 swcatx2c10
       swcatx2a15 swcatx2c15
       swcatx2a20 swcatx2c20 ;
run;quit;run;
*FINAL MODELS;
ods select none;
title "x2a";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta1(where=(parm = "x2a"));run;
data beta1;set beta1;length type $ 10;type = "N";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (1,99)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta2(where=(parm = "x2a"));run;
data beta2;set beta2;length type $ 10;type = "N(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (5,95)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta3(where=(parm = "x2a"));run;
data beta3;set beta3;length type $ 10;type = "N(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (10,90)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4(where=(parm = "x2a"));run;
data beta4;set beta4;length type $ 10;type = "N(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ah;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a(where=(parm = "x2a"));run;
data beta4a;set beta4a;length type $ 10;type = "NHc";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ah199;
```

```
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a1(where=(parm = "x2a"));run;
data beta4a1;set beta4a1;length type $ 10;type = "NHc(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ah595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a2(where=(parm = "x2a"));run;
data beta4a2;set beta4a2;length type $ 10;type = "NHc(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ah1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a3(where=(parm = "x2a"));run;
data beta4a3;set beta4a3;length type $ 10;type = "NHc(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a_th;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a4(where=(parm = "x2a"));run;
data beta4a4;set beta4a4;length type $ 10;type = "NHh";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a_th199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a5(where=(parm = "x2a"));run;
data beta4a5;set beta4a5;length type $ 10;type = "NHh(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a_th595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a6(where=(parm = "x2a"));run;
data beta4a6;set beta4a6;length type $ 10;type = "NHh(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a_th1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a7(where=(parm = "x2a"));run;
data beta4a7;set beta4a7;length type $ 10;type = "NHh(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ahu;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a8(where=(parm = "x2a"));run;
data beta4a8;set beta4a8;length type $ 10;type = "NHu";
proc genmod data=c desc;
by h;class id;
```

```
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ahu199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a9(where=(parm = "x2a"));run;
data beta4a9;set beta4a9;length type $ 10;type = "NHu(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ahu595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a10(where=(parm = "x2a"));run;
data beta4a10;set beta4a10;length type $ 10;type = "NHu(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ahu1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a11(where=(parm = "x2a"));run;
data beta4a11;set beta4a11;length type $ 10;type = "NHu(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2a = x2a / dist=bin maxiter=100;
weight swgx2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4b(where=(parm = "x2a"));run;
data beta4b;set beta4b;length type $ 10;type = "G";
proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2a = x2a / dist=bin maxiter=100;
weight swgx2a199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4b1(where=(parm = "x2a"));run;
data beta4b1;set beta4b1;length type $ 10;type = "G(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2a = x2a / dist=bin maxiter=100;
weight swgx2a595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4b2(where=(parm = "x2a"));run;
data beta4b2;set beta4b2;length type $ 10;type = "G(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2a = x2a / dist=bin maxiter=100;
weight swgx2a1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4b3(where=(parm = "x2a"));run;
data beta4b3;set beta4b3;length type $ 10;type = "G(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (1df)";
model y2a = x2a / dist=bin maxiter=100;
weight swt1x2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta5(where=(parm = "x2a"));run;
```

```
data beta5;set beta5;length type $ 10;type = "t1";
proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (3df)";
model y2a = x2a / dist=bin maxiter=100;
weight swt3x2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta6(where=(parm = "x2a"));run;
data beta6;set beta6;length type $ 10;type = "t3";
proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (5df)";
model y2a = x2a / dist=bin maxiter=100;
weight swt5x2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta7(where=(parm = "x2a"));run;
data beta7;set beta7;length type $ 10;type = "t5";

proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (10)";
model y2a = x2a / dist=bin maxiter=100;
weight swcatx2a10;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta8(where=(parm = "x2a"));run;
data beta8;set beta8;length type $ 10;type = "QB10";
proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (15)";
model y2a = x2a / dist=bin maxiter=100;
weight swcatx2a15;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta9(where=(parm = "x2a"));run;
data beta9;set beta9;length type $ 10;type = "QB15";
proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (20)";
model y2a = x2a / dist=bin maxiter=100;
weight swcatx2a20;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta10(where=(parm = "x2a"));run;
data beta10;set beta10;length type $ 10;type = "QB20";
run;

title "x2c";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta11(where=(parm = "x2c"));run;
data beta11;set beta11;length type $ 10;type = "N";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (1,99)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta12(where=(parm = "x2c"));run;
data beta12;set beta12;length type $ 10;type = "N(1,99)";
proc genmod data=c desc;
by h;class id;
```

```
title2 "Standard Normal (5,95)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta13(where=(parm = "x2c"));run;
data beta13;set beta13;length type $ 10;type = "N(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (10,90)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14(where=(parm = "x2c"));run;
data beta14;set beta14;length type $ 10;type = "N(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2ch;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a(where=(parm = "x2c"));run;
data beta14a;set beta14a;length type $ 10;type = "NHc";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2ch199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a1(where=(parm = "x2c"));run;
data beta14a1;set beta14a1;length type $ 10;type = "NHc(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2ch595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a2(where=(parm = "x2c"));run;
data beta14a2;set beta14a2;length type $ 10;type = "NHc(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2ch1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a3(where=(parm = "x2c"));run;
data beta14a3;set beta14a3;length type $ 10;type = "NHc(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c_th;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a4(where=(parm = "x2c"));run;
data beta14a4;set beta14a4;length type $ 10;type = "NHh";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c_th199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a5(where=(parm = "x2c"));run;
```

```
data beta14a5;set beta14a5;length type $ 10;type = "NHh(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c_th595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a6(where=(parm = "x2c"));run;
data beta14a6;set beta14a6;length type $ 10;type = "NHh(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c_th1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a7(where=(parm = "x2c"));run;
data beta14a7;set beta14a7;length type $ 10;type = "NHh(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2chu;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a8(where=(parm = "x2c"));run;
data beta14a8;set beta14a8;length type $ 10;type = "NHu";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2chu199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a9(where=(parm = "x2c"));run;
data beta14a9;set beta14a9;length type $ 10;type = "NHu(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2chu595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a10(where=(parm = "x2c"));run;
data beta14a10;set beta14a10;length type $ 10;type = "NHu(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2chu1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a11(where=(parm = "x2c"));run;
data beta14a11;set beta14a11;length type $ 10;type = "NHu(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2c = x2c / dist=bin maxiter=100;
weight swgx2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14b(where=(parm = "x2c"));run;
data beta14b;set beta14b;length type $ 10;type = "G";
proc genmod data=c desc;
by h;class id;
title2 "Gamma(1,99)";
model y2c = x2c / dist=bin maxiter=100;
```

```
weight swgx2c199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14c(where=(parm = "x2c"));run;
data beta14c;set beta14c;length type $ 10;type = "G(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Gamma(5,95)";
model y2c = x2c / dist=bin maxiter=100;
weight swgx2c595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14d(where=(parm = "x2c"));run;
data beta14d;set beta14d;length type $ 10;type = "G(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Gamma(10,90)";
model y2c = x2c / dist=bin maxiter=100;
weight swgx2c1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14e(where=(parm = "x2c"));run;
data beta14e;set beta14e;length type $ 10;type = "G(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (1df)";
model y2c = x2c / dist=bin maxiter=100;
weight swt1x2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta15(where=(parm = "x2c"));run;
data beta15;set beta15;length type $ 10;type = "t1";
proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (3df)";
model y2c = x2c / dist=bin maxiter=100;
weight swt3x2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta16(where=(parm = "x2c"));run;
data beta16;set beta16;length type $ 10;type = "t3";
proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (5df)";
model y2c = x2c / dist=bin maxiter=100;
weight swt5x2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta17(where=(parm = "x2c"));run;
data beta17;set beta17;length type $ 10;type = "t5";
proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (10)";
model y2c = x2c / dist=bin maxiter=100;
weight swcatx2c10;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta18(where=(parm = "x2c"));run;
data beta18;set beta18;length type $ 10;type = "QB10";
proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (15)";
model y2c = x2c / dist=bin maxiter=100;
weight swcatx2c15;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta19(where=(parm = "x2c"));run;
data beta19;set beta19;length type $ 10;type = "QB15";
proc genmod data=c desc;
by h;class id;
```

```
title2 "Quantile Binning (20)";
model y2c = x2c / dist=bin maxiter=100;
weight swcatx2c20;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta20(where=(parm = "x2c"));run;
data beta20;set beta20;length type $ 10;type = "QB20";
run;
title;title2;
ods select all;
*COMPUTE BIAS AND MSE;
data beta;
       set beta1-beta10 beta4a beta4a1-beta4a11 beta4b beta4b1-beta4b3 beta11-beta20 beta14a
beta14a1-beta14a11 beta14b beta14c beta14d beta14e;
       merg=1;
data bias;
       merge meancheck beta;
       by merg;
       if parm = "x2a" then do; bias = estimate - truex2a;  mse = (estimate -
truex2a)**2;end;
       if parm = "x2c" then do; bias = estimate - truex2c;  mse = (estimate -
truex2c)**2;end;

       if type="QB20" then typen=1;
       if type="QB15" then typen=2;
       if type="QB10" then typen=3;

       if type="G(10,90)" then typen=4;
       if type="G(5,95)" then typen=5;
       if type="G(1,99)" then typen=6;
       if type="G" then typen=7;

       if type="t5" then typen=8;
       if type="t3" then typen=9;
       if type="t1" then typen=10;

       if type="NHu(10,90)" then typen=11;
       if type="NHu(5,95)" then typen=12;
       if type="NHu(1,99)" then typen=13;
       if type="NHu" then typen=14;

       if type="NHh(10,90)" then typen=15;
       if type="NHh(5,95)" then typen=16;
       if type="NHh(1,99)" then typen=17;
       if type="NHh" then typen=18;

       if type="NHc(10,90)" then typen=19;
       if type="NHc(5,95)" then typen=20;
       if type="NHc(1,99)" then typen=21;
       if type="NHc" then typen=22;

       if type="N(10,90)" then typen=23;
       if type="N(5,95)" then typen=24;
       if type="N(1,99)" then typen=25;
       if type="N" then typen=26;

run;
proc format;
       value tpn 1 = "QB20"
                      2 = "QB15"
                      3 = "QB10"

                      4 = "G(10,90)"
                      5 = "G(5,95)"
```

```
                           6 = "G(1,99)"
                           7 = "G"

                           8 = "t5"
                           9 = "t3"
                           10 = "t1"

                           11 = "NHu(10,90)"
                           12 = "NHu(5,95)"
                           13 = "NHu(1,99)"
                           14 = "NHu"

                           15 = "NHh(10,90)"
                           16 = "NHh(5,95)"
                           17 = "NHh(1,99)"
                           18 = "NHh"

                           19 = "NHc(10,90)"
                           20 = "NHc(5,95)"
                           21 = "NHc(1,99)"
                           22 = "NHc"

                           23 = "N(10,90)"
                           24 = "N(5,95)"
                           25 = "N(1,99)"
                           26 = "N"
;
*RESULTS;
proc sort data=bias;by parm typen;run;
proc means data=bias mean maxdec=4;
class parm typen;
var bias mse;format typen tpn.;run;
proc univariate data=bias noprint;
class parm type;
var bias;histogram;
run;quit;run;
*END;
```