

ESTIMATING MEANS AND DISTRIBUTIONS OF COUNT DATA FROM SURVEYS  
INVOKING COMPLEX SAMPLING DESIGNS: APPLICATION TO MEASURING  
CROSS-SECTIONAL AND LIFETIME ALCOHOL INTAKE FROM NHANES AND  
NLSY

by

ELYSIA A. GARCIA, MPH, BS

APPROVED:



---

STACIA M. DESANTIS, PHD, MS



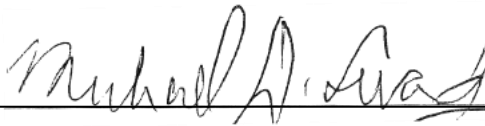
---

STACIA M. DESANTIS, PHD, MS



---

ALANNA C. MORRISON, PHD, FAHA



---

MICHAEL D. SWARTZ, PHD, MA

---

DEAN, THE UNIVERSITY OF TEXAS  
SCHOOL OF PUBLIC HEALTH

Copyright  
by  
Elysia A. Garcia, PhD, MPH, BS  
2020

## DEDICATION

To Ricardo, Patricia, and Monica Garcia, Ronald Paranal, Manuel and Brigida Blanco,  
Emma Garcia, Dr. Rose-Mary Rodriguez, and Dr. Jeanne Hill

ESTIMATING MEANS AND DISTRIBUTIONS OF COUNT DATA FROM SURVEYS  
INVOKING COMPLEX SAMPLING DESIGNS: APPLICATION TO MEASURING  
CROSS-SECTIONAL AND LIFETIME ALCOHOL INTAKE FROM NHANES AND  
NLSY

by

ELYSIA A. GARCIA  
MPH, University of North Texas Health Science Center, 2016  
BS, Baylor University, 2014

Presented to the Faculty of The University of Texas

School of Public Health

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS  
SCHOOL OF PUBLIC HEALTH  
Houston, Texas  
May, 2020

## PREFACE

I had a joint fascination for mathematics and diseases from a young age which led me down the road of public health and biostatistics early on in my education career. However, it was the culture of leadership and service inspired by my family and my childhood raised in the military that led me to pursuing a PhD. Thus, when deciding on a dissertation research topic, I wanted a project that would join these loves and values in order to have a direct application to public health efforts.

## ACKNOWLEDGEMENTS

I first and foremost would like to acknowledge and thank my family for their continuous love and support. There have been many roadblocks along this path towards a PhD, but you have been beside me every step of the way offering advice, comfort, laughs, and the occasional frozen homemade meal sent express. To my dad, Ricardo, my mom, Patricia, and my sister, Monica; thank you and I love you dearly.

I would also like to acknowledge the contribution of my dissertation panel; Dr. Stacia DeSantis, my dissertation and academic advisor, Dr. Alanna Morrision, my epidemiology advisor, Dr. Michael Swartz, my bioinformatics advisor, and my external reviewers, Dr. Luis Leon Novelo and Dr. Wenyaw Chan. Thank you for your time and guidance during this project.

Finally, I would like to acknowledge the National Institute of General Medical Sciences Predoctoral Training grant (T32 GM074902) which supported this research.

ESTIMATING MEANS AND DISTRIBUTIONS OF COUNT DATA FROM SURVEYS  
INVOKING COMPLEX SAMPLING DESIGNS: APPLICATION TO MEASURING  
CROSS-SECTIONAL AND LIFETIME ALCOHOL INTAKE FROM NHANES AND  
NLSY

Elysia A. Garcia, PhD, MPH, BS  
The University of Texas  
School of Public Health, 2020

Dissertation Chair: Stacia M. DeSantis, PhD, MS

It is often of interest to measure the distribution (i.e. mean and percentiles) of count outcomes from national survey data to assess population consumption and guide public health efforts for substances such as alcohol, cigarettes, marijuana, and other illicit or licit drugs. Currently available methods for estimating the distribution of dietary intakes do not immediately lend themselves to estimating the consumption of substances measured as counts, nor do they accommodate the complex design elements – strata, cluster, and weight – characteristic of national surveys. We introduce an accurate methodology, called the Survey-Adjusted Count (SAC) method, and an associated SAS macro for estimating population-level distribution statistics (means, percentiles, and standard errors) for cross-sectional and longitudinal count data that arise from complex national surveys. First, a negative binomial hurdle is used to estimate the product of the probability of consuming in a given time period (e.g. day or year) and the amount consumed in the time period, over the number of longitudinal observations available in the study (two or more, depending on the study). These parts are then linked, thus allowing for correlation between the two model parts. Using these model-based

parameter estimates, the distribution of consumption is then simulated to calculate population mean consumption and percentiles. Standard errors are then estimated using Balanced Repeated Replication method which accounts for stratification, clustering, and weighting. We validated the methodology via a simulation study comparing its performance versus currently available methods, and illustrated the utility of the method by estimating alcohol intake from a cross-sectional and a longitudinal survey – the National Health and Nutrition Examination Survey and the National Longitudinal Survey of Youth, respectively. Application of the method to these data allowed us to estimate mean (cross-sectional) and lifetime mean (longitudinal) consumption of alcohol, as well as percentiles and standard errors. Furthermore, using the SAS macro we provide, these distributions can be estimated by subgroups/demographics of interest. Therefore, utilizing the SAC method presented, we can attain accurate estimates of short-term and/or lifetime consumption for counts by subgroups of interest and facilitate accurate public health recommendations.



## TABLE OF CONTENTS

List of Tables .....	i
List of Figures .....	ii
Background .....	1
Literature Review.....	1
The National Cancer Institute Method: The current gold standard method for estimating the distribution of usual nutrient intake from survey data .....	6
Public Health Significance.....	8
Aims and Objectives .....	10
Methods.....	11
Model Assumptions .....	12
Model Building .....	13
Model Fitting .....	15
Estimation of the Distribution of Survey Count Data.....	17
Estimation of Standard Errors of the Distribution of Survey Count Data .....	17
Results.....	20
Simulation Study .....	20
Results .....	21
Data Application: Estimating mean and distribution of alcohol consumption from NHANES recall.....	28
Results .....	30
Data Application: Estimating mean and distribution of alcohol consumption from NLSY 1997 recall .....	39
Results .....	42
Discussion .....	49
Conclusion .....	51
Appendices.....	54
References.....	140

## LIST OF TABLES

Table 1: Comparison of mean percentiles and their standard errors based on the simulated count data with 20% zero-inflation. ....	23
Table 2: Comparison of mean percentiles and their bias based on the simulated count data with 20% zero-inflation. ....	23
Table 3: Comparison of means, standard errors, and bias based on the simulated count data with 20% zero-inflation. ....	24
Table 4: Comparison of mean percentiles and their standard errors based on the simulated count data with 50% zero-inflation. ....	26
Table 5: Comparison of mean percentiles and their bias based on the simulated count data with 50% zero-inflation. ....	26
Table 6: Comparison of means, standard errors, and bias based on the simulated count data with 50% zero-inflation. ....	27
Table 7: Descriptive statistics for NHANES 2003-2004 and 2015-2016. ....	31
Table 8: SAS model estimates, odds ratios, and rate ratios for the negative binomial hurdle model for population average alcohol consumption of NHANES 2003-2004. ....	34
Table 9: SAS model estimates, odds ratios, and rate ratios for the negative binomial hurdle model for population average alcohol consumption of NHANES 2015-2016. ....	35
Table 10: SAS distribution estimates and standard errors for population average alcohol consumption of NHANES 2003-2004. ....	37
Table 12: Estimated percentile of alcohol consumption per day for adults $\geq 19$ years, estimated using five methods. Data are from the National Health and Nutritional Examination Survey 2003-2016. ....	38
Table 13: Descriptive statistics for NLSY 1997 waves 2013-2015. ....	44
Table 14: SAS model estimates, odds ratios, and rate ratios for the negative binomial hurdle model for population average alcohol consumption of NLSY 1997 waves 2013-2015. ....	47
Table 15: SAS distribution estimates and standard errors for population average alcohol consumption of NLSY 1997 waves 2013-2015. ....	48
Table 16: SAS distribution estimates categorized by risky drinking status for population average alcohol consumption of NLSY 1997 waves 2013-2015. ....	49

## LIST OF FIGURES

Figure 1: Comparison of selected mean percentiles and 95% confidence intervals for two-day mean, the NCI Method, and the SAC Method from simulated data with 20% zero-inflation. ....	25
Figure 2: Comparison of selected mean percentiles and 95% confidence intervals for two-day mean, the NCI Method, and the SAC Method from simulated data with 50% zero-inflation. ....	28
Figure 3: Estimated percentile of alcohol consumption per day for adults $\geq 19$ years, estimated using five methods. Data are from the National Health and Nutritional Examination Survey 2003-2016. ....	39
Figure 4: Diagram of NLSY 1997 measurement of alcohol as a count outcome. ....	42

## List of Appendices

Appendix A: SAC Method MIXTRAN SAS Macro for model estimation .....	54
Appendix B: SAC Method DISTRIB SAS Macro for distribution estimation .....	103
Appendix C: SAS code to make replicate BRR weights for NLSY 2013-2015 .....	126
Appendix D: BRR SAS Macro for estimating distribution standard errors .....	131
Appendix E: Simulation Study .....	138

## **BACKGROUND**

### **Literature Review**

Illicit drug, legal drug, and alcohol use in the United States (US) has seen an upsurge in recent years; however, how much of an illicit or licit substance people in the US are consuming remains unknown. This thesis introduces a statistical methodology that can more accurately measure population consumption of such substances compared to current methods. This research would contribute to the 1.) implementation of risk analyses to determine associations between drug and alcohol consumption and prevalent health outcomes, 2.) application to longitudinal surveys to determine association between lifetime consumption and incident health outcomes, and 3.) characterization of consumption in the US by important demographic characteristics, for which public health measures could be targeted. The focus of this dissertation will be on alcohol use, but the same methods apply to any licit or illicit substance.

The proposed statistical methodology accommodates the four-stage, complex sampling design while summarizing longitudinal counts of reported alcohol consumption, subsequently enabling estimation of population consumption in the form of means, distributions, and percentiles, potentially by subgroups of interest (e.g., demographics), that will help inform public health decisions. By taking into account the stratification, clustering, and weighting characteristics of complex surveys, we are can estimate correct standard errors for the aforementioned parameters. The methodology takes into account not only the unique nature of consumption data, but also accommodates the fact that such data has specific qualities and behaviors that require tailored methodology and modeling schemes to analyze

data effectively and output correct population estimates of consumption (for example, “weekend drinking”).

One motivating data example is the National Health and Nutritional Examination Survey (NHANES), a national epidemiologic cross-sectional survey implementing complex multi-stage sampling strategies. From NHANES, variables such as substance consumption are collected using methods from retrospective and prospective calendar recall data. The first challenge presented by such data is accommodating the survey sampling procedure utilized to create a representative sample of the US population. This sampling design results in three components that must be utilized in any subsequent statistical analysis: strata, clusters, and weights. Strata are defined by geography and proportions of minority populations and represent the sampling units used to estimate sampling error (“NHANES Web Tutorial Frequently Asked Questions (FAQs),” 2014). Clustering refers to the homogeneity of individuals within a given cluster. Finally, weights are assigned to each sampled person in the survey, and account for oversampling, survey non-response, and post-stratification. The result is a sample that is representative of the US Census civilian non-institutionalized population. This type of design is employed across most US epidemiologic surveys performed by the Centers for Disease Control and Prevention (CDC) and the National Institutes of Health (NIH) and is, therefore, a mainstay.

One goal of the design is to decrease the amount of correlation between sample persons within a cluster to create a more nationally representative sample. This is achieved by sampling fewer individuals within each cluster and sampling more clusters (CDC, 2011). Furthermore, such surveys also incorporate multi-stage sampling designs, which allows the

surveyed sample to be reflective of the US population during the year sampled due to how each stage was selected with probability proportional to the population. For example, the NHANES is characterized by the following 4-stages:

1. Primary sampling units (PSUs) consisting of mostly single counties
2. Segments divided from within PSUs (usually city blocks or their equivalent)
3. Households within each segment from which a sample is randomly drawn
4. Individuals randomly selected from all persons in selected households.

To increase the reliability and precision of estimates of health outcomes, larger samples are collected for certain subgroups of particular public health interest. Thus, to estimate representative parameters of drug and alcohol consumption for a population, the incorporation of the unique sample design and the components it utilizes is crucial.

Current strategies to summarize drug and alcohol consumption from epidemiological survey data also present the challenge of how to quantify the consumption of such substances. At this time there is no single way or universally agreed-upon measure (Anton et al., 2006; Shirley, Small, Lynch, Maisto, & Oslin, 2010). For example, alcohol consumption calendar-based methods typically report the number of drinks each day surveyed, for which statistical summaries for analysis have been derived for typical clinical trials (Anton et al., 2006; Shirley, Small, Lynch, Maisto, & Oslin, 2010; DeSantis & Bandyopadhyay, 2011). Typically many different summaries are reported such as number of drinks per day, number of drinks per drinking day, and percentage of binge drinking days; however, as counts are most commonly recorded by calendar recall, these should be directly analyzed (Shirley, Small, Lynch, Maisto, & Oslin, 2010; DeSantis & Bandyopadhyay, 2011).

In addition to the lack of consensus for the measurement of consumption of substances, there is also a lack of consensus on methodologies that can accurately estimate average consumption from survey data. Current methods exist to estimate averages and distributions for continuous longitudinally reported survey data (e.g., grams (g)/day of a food, nutrient, or supplement such as rice, calcium, or a protein shake); however, these have a couple of limitations. First, they do not directly handle longitudinal count data (with likely excess zeros). Secondly, they do not fully accommodate the complex multi-stage sampling design used in many epidemiologic surveys, such as NHANES.

Existing methods for assessing “usual intake” (e.g. g/day) of foods and supplements vary. The most naive way to calculate “usual dietary intake” (average intake over two or more dietary recall days of the survey) is to measure several single-day intakes and then estimate the mean. This results in increased within-person variability leading to a larger variance and, consequentially, biased estimates of usual intake (Dodd et al., 2006). More advanced methods use a similar framework as outlined in Dodd et al. (2006), which can be summarized in three steps:

1. Describe the assumed relationship between multiple observations of individual 24-hour recall (24 HR) and individual usual intake
2. Divide the total variation in 24 HR into within-person and between-person variation elements
3. Estimate the usual intake distribution accounting for the within-person variation.



The Institute of Medicine (IOM) method utilizes this framework to estimate the average intake of dietary components, but it is only suitable for independent sampling surveys. The Best-Power (BP) method builds upon the IOM method to accommodate survey data; however, neither accommodate excess zero observations observed with both episodically-consumed food and supplement intake, and substance intake, which is the focus of this paper. The Iowa State University Food (ISUF) and National Cancer Institute (NCI) methods both attempt to address these inadequacies. Nusser et.al.'s (1990) ISUF method incorporates a two-part model for continuous variables (g/day) in which zero observations are treated separately from other observations (Nusser, Carriquiry, Jensen, & Fuller, 1990). The NCI method moves a step further by incorporating a link between errors generated by the two-part model; specifically, it allows a correlation structure to account for the fact that the amount (g/day) consumed on a consumption day is correlated with the probability of consumption. The NCI method is also unique in allowing for analysis of usual intake for subgroups of the population, for example, important demographics. Based on its advancements compared to the above methods, its proven superior performance (Laureano, Torman, Crispim, Dekkers, & Camey, 2016; Souverein et al., 2011; Tooze et al., 2010), and its excellent packaging into downloadable SAS macros, the NCI method is currently the gold standard for analyzing usual dietary intake (National Cancer Institute, 2018; Usual Dietary Intakes: SAS Macros for the NCI Method, 2018).

For estimating the distribution of population intake for count outcomes, the NCI method has some limitations: 1.) it is only applicable to continuous outcomes (versus counts) and 2.) it does not account for the stratification component of the survey design in variance

estimation. In a statistical sense, this limits its applicability or immediate adaptability to the current setting and would potentially lead to incorrect inference. Here, we derive a novel statistical methodology, named the Survey-Adjusted Count (SAC) method, and associated macros in the SAS 9.4 language to estimate population-level statistics (means, variances, and distributions) for cross-sectional and longitudinal count data that arise from complex national surveys that allow for accurate estimates of intake, intake by subgroups, lifetime intake, and facilitate accurate public health recommendations.

***The National Cancer Institute Method: The current gold standard method for estimating the distribution of usual nutrient intake from survey data***

The NCI method estimates the usual intake distributions for episodically-consumed dietary components, which exhibit a large proportion of zero intakes on any given day. Further details on the methodology can be found in the primary document by Tooze et.al. (2010) and we elaborate below.

Let the observed daily consumption (in g/day) for a food component be  $R_{ij}$  where  $i$  is the individual ( $i = 1, \dots, N$ ) and  $j$  is the day ( $j = 1, 2$  for NHANES but without the loss of generality the maximum  $j$  could clearly be larger). The two-part model consists of the following components: the probability of consuming a food on a particular day, and the amount eaten on the consumption day given that food was consumed. The probability of consumption is modeled using a mixed-effects logistic regression; subsequently, the amount consumed is then modeled utilizing a mixed-effects linear regression after Box-Cox transformation of the 24HR data,  $R_{ij}^* = g(R_{ij}, \lambda)$ , with  $g(r, \lambda) = (r^\lambda - 1)\lambda^{-1}$  being the

transformation factor. When  $\lambda = 0$  the natural log transformation is utilized. These two models are then linked via the individual random person-specific effects, producing a variance-covariance structure which is then included in the final joint model (e.g., Farewell, Long, Tom, Yiu, & Su, 2017; Liu, Cowen, Strawderman, & Shih, 2010; Min & Agresti, 2005). This variance-covariance structure accommodates for the correlation between the probability of consumption and the amount consumed during the consumption day. The sampling weight component of complex survey design is incorporated into two parts within the model framework. First, the weight is incorporated by weighting each observation by the inverse weight. This is accomplished in SAS PROC NLMIXED by including a “frequency” statement in the generalized linear model of starting-point estimates for the logistic regression and the normal linear regression. The frequency statement for the generalized linear model treats each observation in the input data set as though it occurred the number of times indicated by the value of the corresponding weight (SAS Institute Inc., 2008). Second, the mixed-effects models treat the weight as the number of subjects that have data identical to the subjects presented in the given data. This is done by multiplying the log-likelihood contribution of each subject by its weight, which consists of one or more observations. In the NCI SAS Macro (Usual Dietary Intakes: SAS Macros for the NCI Method, 2018), the weight variable is included in the “replicate” statement within the PROC NLMIXED procedure (SAS Institute Inc., 2015). This estimates the mixed-effects logistic and linear regression, which utilizes the starting-point estimates from those two generalized linear models, respectively.

Estimates from the NCI method's model estimation are then used along with the Monte Carlo (MC) method to estimate the distribution of consumption for the population by evaluating these statistics in random samples from known populations of simulated data (Mooney, 1997). Tooze et al. utilize a nine-point quadrature approximation method (similarly presented in the ISUF method of Nusser, Carriquiry, Dodd, & Fuller (1996), to approximate usual total intake for an individual,  $i$ , denoted as  $T_i$  (Tooze et al., 2010; Nusser, Carriquiry, Dodd, & Fuller, 1996). To achieve this, one first defines the Box-Cox transformation of the 24HR data,  $g(r, \lambda) = (r^\lambda - 1)\lambda^{-1}$ . The simulated mean is  $\mu_l^* = \mathbf{X}_l' \hat{\boldsymbol{\beta}} + \mu_l$  where  $l = 1, \dots, kN$  and  $\mathbf{X}_l = \mathbf{X}_i$  for  $l = i, \dots, ki$  where  $k$  is the simulation realizations,  $i$  is the sampled individual, and  $N$  is the sample size. The nine-point approximation is then as follows:

$$T_l \cong \sum_{k=1}^9 w_k g^{-1}(\mu_l^* + c_k \sigma_e^2) = \sum_{k=1}^9 w_k ((\mu_l^* + c_k \sigma_e^2) \lambda + 1)^{\frac{1}{\lambda}}$$

where estimates of  $\lambda$  and  $\sigma_e^2$  are utilized from the prior two-part model parameter estimates to obtain  $T_i$ . Furthermore, the nine points,  $c_k$ , and nine weights,  $w_k$ , where  $\sum w_k = 1$ , are constructed so that the first five moments of the discrete distribution are the same for the first nine estimated moments of the conditional distribution of  $\mu_l^*$  conditional on  $\mathbf{X}_l' \hat{\boldsymbol{\beta}}$  (Nusser, Carriquiry, Dodd, & Fuller, 1996; Tooze et al., 2010).

### **Public Health Significance**

The SAC method applies to the measurement of substance intakes such as alcohol, marijuana, or illicit drug consumption from complex survey data. This is of particular interest due to the unique nature and challenges inherent in substance-use data (e.g., these are often

measured as counts) as well as behaviors (random fluctuation in use, increased weekend use, etc.). Most importantly, the SAC method accommodates the challenges of complex survey data - namely clustering, stratification, and weights.

For substance use outcomes, typically calendar-based methods of recall are used which report the number of consumption (e.g. standard drinks, cigarettes, marijuana joints) in a specified time period, for which statistical summaries for analysis have been derived for standard clinical trials (Anton et al., 2006; Shirley, Small, Lynch, Maisto, & Oslin, 2010; DeSantis & Bandyopadhyay, 2011). Substance-use outcome data are also characterized by zero-inflation, thus the SAC method's two-part count model allows for the probability of consumption of the substance in question and the amount consumed to follow unique distributions (DeSantis & Bandyopadhyay, 2011; DeSantis et al., 2013; Gueorguieva et al., 2010; Heilbron, 2007; Mullahy, 1986; Shirley, Small, Lynch, Maisto, & Oslin, 2010; Ver Hoef & Boveng, 2007; Zhu, Luo, & DeSantis, 2017). The SAC method also allows for the analysis of subgroup effects, which is pinnacle in substance use research where consumption trends based on such characteristics as location, race/ethnicity, or gender can differ greatly (Becker, McClellan, & Reed, 2017; Center for Behavioral Health Statistics and Quality, 2017; McCabe et al., 2007; Substance Abuse and Mental Health Services Administration, Center for Behavioral Health Statistics and Quality, 2016; Witbrodt, Mulia, Zeng, & Kerr, 2014). Finally, the ability of the method to accurately define the distribution (i.e. mean and percentiles) and its associated standard errors of substance-use consumption allows for the characterization of risky and/or abusive consumption behaviors in the population based on categorizations and covariates of interest to researchers. For example, the Substance Abuse

and Mental Health Services Administration (SAMHSA) and the NIAAA define binge drinking as more than five alcoholic drinks for males or more than four alcoholic drinks for females on the same occasion (i.e., at the same time or within a couple of hours of each other) on at least one day in the past month, and defines heavy alcohol use as binge drinking on five or more days in the past month (“Drinking Levels Defined,” 2011). Therefore, having details for population consumption, particularly for the upper percentiles of consumption, enables researchers to characterize risky behaviors, enact interventions, and monitor consumption over time in order to contribute to further public health efforts.

### **Aims and Objectives**

Our main objective for this project is to disseminate a methodology – i.e. the SAC method – to accurately estimate the distribution (i.e. mean and percentiles) and its standard errors for substance consumption measured as counts from national survey data.

Furthermore, we defined three specific aims to accompany this objective:

1. Develop a statistical method to estimate the mean, distribution, and standard errors of count variables from cross-sectional national survey data with an application to NHANES,
2. Develop a statistical method to estimate the mean, distribution, and standard errors of count variables from longitudinal national survey data with an application to NLSY 1997,
3. Develop a SAS macro to implement the new method for population average intake of a count outcome.

## METHODS

We present a method to estimate population mean and distribution (percentiles) of consumption for a count outcome by subgroups, from longitudinal zero-inflated count data. A longitudinal, two-part negative binomial hurdle model (HNB) is applied to accommodate count zero-inflation. This model was chosen based on the obvious zero-inflated nature of alcohol use in the general population. It was also chosen based on the fact that all individuals are at risk for exposure to consumption but have the choice of consuming or not, and subsequently, the choice of how much to consume. The model also allows repeated measures data with correlated random effects. The stratification, cluster, and weight components of complex survey design are incorporated into the model. We then simulate the distribution of population consumption to give results of population consumption which incorporate the three complex survey design components in order to estimate correct standard errors of the distribution statistics (i.e. mean and percentiles). Our SAC method requires two or more longitudinal observations of the outcome of interest on at least a random subset of the population. With these adaptations, the SAC method can be extended to measure population distribution of consumption for a count outcome by user-specified subgroups in order to advise public health recommendations. For this paper, we focused on alcohol consumption as our outcome of interest, however, the SAC method is applicable to a wide range of consumption outcomes measured as a count.

## Model Assumptions

Let  $T_{ij}$  be the true alcohol consumption for an individual  $i$  ( $i = 1, \dots, n$ ) on day  $j$  ( $j = 1, \dots, m$ ) on the original scale. The average consumption for an individual is  $T_i = E(T_{ij}|i)$ , i.e. the conditional expectation of the true single-day consumption. Also, let  $p_i$  be the true probability of consuming alcohol. The probability that the true consumption is greater than zero for a given person  $i$ , is  $p_i = \Pr(T_{ij} > 0|i)$ . Finally, let  $A_i$  be the true average consumption amount; thus, the average true consumption given that the individual consumed on the day is  $A_i = E[T_{ij}|i, T_{ij} > 0]$ . Based on these properties, it can be shown that true average consumption is the product of the probability and the amount with  $T_i = E[T_{ij}|i] = p_i A_i$ . However, without the true values, we rely instead on the recall reported consumption.

Let  $Y_{ij}$  be the observed number of drinks consumed from recall data for person  $i$  on day  $j$ . It is an assumption that  $Y_{ij}$  is an unbiased measure of the consumption day amount and that a count is reported if and only if the individual consumed alcohol that day. Therefore, the probability of consumption on recall is the same as the probability of true consumption with  $\Pr(Y_{ij} > 0|i) = \Pr(T_{ij} > 0|i) = p_i$ . The assumption that the recall measure is an unbiased instrument for usual amount consumed on a consumption day gives the property  $E[Y_{ij}|i; Y_{ij} > 0] = A_i$ , thus that the recall exhibits random error. Finally, it is assumed that, on average, the recall measure is unbiased for true usual consumption  $E[Y_{ij}|i] = p_i A_i = T_i$ .



## Model Building

The SAC method relies on two model components; the probability of consuming alcohol and the amount of consumption in that time frame (Bandyopadhyay, DeSantis, Korte, & Brady, 2011; Farewell, Long, Tom, Yiu, & Su, 2017; Liu, Cowen, Strawderman, & Shih, 2010; Min & Agresti, 2005; Tooze et al., 2010). The probability component of the two-part model is reflected in the first part of the hierarchical Negative Binomial Hurdle (HNB) model. To determine whether a person consumed alcohol, a mixed-effects logistic regression model is proposed,

$$\text{logit} \left( P(Y_{ij} > 0 | \mathbf{X}_{1i}, u_{1i}) \right) = \beta_{10} + \boldsymbol{\beta}'_{X1} \mathbf{X}_{1i} + \boldsymbol{\beta}'_{Time_{1j}} \mathbf{Time}_{1j} + u_{1i}, \quad (1)$$

where  $\mathbf{X}_{1i}$  is a vector of covariates,  $\mathbf{Time}_{1j}$  is a vector of  $j$  time points, and  $u_{1i}$  is a person-specific random effect with distribution  $N(0, \sigma_{u_1}^2)$ . These random effects allow correlation within an individual for multiple time points of response when estimating mean consumption over the time period measured.

The second part of the two-part model models the amount consumed (if alcohol consumption occurred). Utilizing a mixed-effects negative binomial regression, the linear function is,

$$\log(E[Y_{ij} | Y_{ij} > 0; \mathbf{X}_{2i}, u_{2i}]) = \beta_{20} + \boldsymbol{\beta}'_{X2} \mathbf{X}_{2i} + \boldsymbol{\beta}'_{Time_{2j}} \mathbf{Time}_{2j} + u_{2i}, \quad (2)$$

where  $\mathbf{X}_{2i}$  is a vector of covariates,  $\mathbf{Time}_{2j}$  is a vector of  $j$  time points, and  $u_{2i} \sim N(0, \sigma_{u_2}^2)$  is a person-specific random effect. The random effects are assumed to be additive and independent.

Finally, it is assumed that the person-specific random effects,  $u_{1i}$  and  $u_{2i}$ , have a bivariate normal distribution  $(u_{1i}, u_{2i})' \sim BVN(\mathbf{0}, \mathbf{\Sigma})$  and variance-covariance matrix is

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{u_1}^2 & \rho\sigma_{\mu 1}\sigma_{\mu 2} \\ \rho\sigma_{\mu 1}\sigma_{\mu 2} & \sigma_{u_2}^2 \end{pmatrix}. \quad (3)$$

A hurdle model is applied using the two components alliterated above that are linked via the variance person-specific random effects and the variance-covariance matrix (Min & Agresti, 2005; Tooze et al., 2010). The hurdle model is written as a mixture of a point mass model at zero and a truncated negative binomial distribution for positive counts. Let the random variable  $Y_{ij}$  denote the count responses for substance consumption for the person  $i$  on time point (day, year, etc.),  $j$ . The hurdle distribution for the observed number of substance consumption is expressed as

$$P(Y_{ij} = y_{ij}) = \begin{cases} p_{ij} & \text{if } y_{ij} = 0 \\ (1 - p_{ij}) \frac{f(y_{ij})}{1 - f(0)} & \text{if } y_{ij} > 0 \end{cases} \quad (4)$$

where  $1 - p_{ij}$  is the binomial probability of "crossing the hurdle" into substance consumption and  $f(y_{ij})$  is the standard Negative Binomial distribution with rate parameter

$\lambda_{ij}$  and dispersion parameter  $k$ ,  $f(y_{ij}) = \frac{\Gamma(y_{ij} + \frac{1}{k})}{\Gamma(y_{ij} + 1)\Gamma(\frac{1}{k})} \left( \frac{1}{1 + k\lambda_{ij}} \right)^{\frac{1}{k}} \left( \frac{k\lambda_{ij}}{1 + k\lambda_{ij}} \right)^{y_{ij}}$ . When  $k \rightarrow 0$ , the

HNB model converges to the Poisson Hurdle model.

We note the proposed HNB model is not novel; however, its adaptation to survey data that will be described below, is novel. Refer to Appendix A for SAS macro of this model.

## Model Fitting

The model is fit by first estimating start points via a generalized linear model (GLM) for the binomial logit,

$$\text{logit}\left(P(Y_{ij} > 0 | \mathbf{X}_{1i}, u_{1i})\right) = \boldsymbol{\omega}_{ij}(\beta_{10} + \boldsymbol{\beta}'_{X1}\mathbf{X}_{1i} + \boldsymbol{\beta}'_{Time_1j}\mathbf{Time}_{1j}), \quad (5)$$

and negative binomial distribution,

$$\log(E[Y_{ij} | Y_{ij} > 0; \mathbf{X}_{2i}, u_{2i}]) = \boldsymbol{\omega}_{ij}(\beta_{20} + \boldsymbol{\beta}'_{X2}\mathbf{X}_{2i} + \boldsymbol{\beta}'_{Time_2j}\mathbf{Time}_{2j}), \quad (6)$$

without random effects. Survey weights,  $\boldsymbol{\omega}_{ij}$ , are incorporated in this parameter estimation stage by multiplying the generalized linear model by the vector of weights for each individual,  $i$ , at time point  $j$  (SAS Institute Inc., 2015). The estimated starting points are then used in estimating the mixed-effects regression model for the logistic (Formula 1) and negative binomial (Formula 2) distributions. The mixed-effects binomial log-likelihood is

$$\mathcal{L} = \sum_{i=1}^n \omega_i [y_{ij} \log(p_{ij}) + (1 - y_{ij}) \log(1 - p_{ij})] \quad (7)$$

and negative binomial log-likelihood is,

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^n \omega_i & \left[ \log \left[ \Gamma \left( y_{ij} + \frac{1}{k} \right) \right] - \log \left[ \Gamma \left( \frac{1}{k} \right) \right] - \frac{1}{k} \log(1 + k\lambda_{ij}) \right. \\ & \left. - y_{ij} \log(1 + k\lambda_{ij}) + y_{ij} \log(k) + y_{ij} \log(\lambda_{ij}) \right], \end{aligned} \quad (8)$$

where  $\omega_i$  is the survey-weighted replicate variable that multiplies the log-likelihood contribution of each subject. In essence, this treats the weight component of the survey design as the number of subjects in the population that have data identical to the data for the

last time observation,  $j$ , of a given subject,  $i$ , in the data (Chou & Steenhard, 2009; SAS Institute Inc., 2015).

The log-likelihood for the HNB model is expressed as

$$\mathcal{L}(\beta, \Theta) = \sum_{i=1}^n \omega_i \left[ I(y_{ij} = 0) \log(p_{ij}) + I(y_{ij} > 0) \left[ \log(1 - p_{ij}) + \log(f(y_{ij}) - (1 - f(0))) \right] \right] \quad (9)$$

where  $\beta$  is the parameter estimates for the explanatory variables in the truncated HNB model,  $\Theta$  is any additional parameter estimates, and  $\omega_i$  is the survey-weighted replicate variable which multiplies the log-likelihood contribution of each subject by the weight integer. The full likelihood of the model is maximized utilizing Nelder-Mead simplex optimization. The Nelder-Mead simplex optimization finds the minimum of a function consisting of  $n$  variables while not requiring the derivative of the function, as is necessary with Newton and quasi-Newton optimization methods (Nelder & Mead, 1965). This quality allows greater applicability to a wider array of optimization problems (Chong & Zak, 2013).

In applying the model fitting specifications to SAS, the binomial and negative binomial GLM are estimated utilizing PROC GLM with the survey weight,  $\omega_{ij}$ , incorporated via the "frequency" statement in the procedure. The mixed models are fitted utilizing PROC NLMIXED including random effects for within-person variability and survey weight,  $\omega_i$ , by the "replicate" statement. The NLMIXED procedure is then used to estimate the HNB model with survey weight,  $\omega_i$ , as a "replicate" statement. The optimization specifications such as Nelder-Mead simplex optimization are incorporated in the NLMIXED procedure.

Refer to Appendix A for SAS coding.

### **Estimation of the Distribution of Survey Count Data**

The approximation and estimation of the distribution in the form of population percentiles and average consumption are done by utilizing the output from model fitting (refer to “Model fitting”) as the foundation of the MC approach. Using this approach, a bivariate distribution of random effects that is proportional to that found in the sampled individuals used to model the HNB is generated by first computing  $\mathbf{X}_i' \hat{\boldsymbol{\beta}}$  for each sampled individual,  $i$ . Next,  $k = 100$  realizations of  $u_l$  are simulated for each person in the data set, with  $u_l$  being a  $N(0, \hat{\sigma}_u^2)$  random variable. These estimated random effects are then combined with the empirical distribution of the fixed effects that come from the data and form a simulated  $\mu_l^* = \mathbf{X}_l' \hat{\boldsymbol{\beta}} + u_l$  where  $l = 1, \dots, kN$  and  $X_l = X_i$  for  $l = i, \dots, ki$ . This will result in  $100N$  values of  $\mu_l^*$ , which are assumed to reflect a representative sample of count average consumption for the population.

Next, the distribution of consumption,  $T_l$ , is approximated by

$$T_l = E(Y_l | \mathbf{X}_l, u_l; \boldsymbol{\beta}) \cong \exp(\mathbf{X}_l' \boldsymbol{\beta}). \quad (10)$$

Finally, linear interpolation is used to obtain distribution percentiles.

Refer to Appendix B for SAS macro coding.

### **Estimation of Standard Errors of the Distribution of Survey Count Data**

To incorporate the survey stratification component to obtain correct standard error estimations of the distribution, a replication-based method is utilized. Replication methodology requires drawing multiple replicates, i.e. subsamples, from the full sample by

following a specific resampling scheme. From each replicate, the parameter of interest is estimated, and then the variability among the replicated estimates is used to compute the variance and, subsequently, the standard errors of the full estimate (Barbosa, Sichieri, & Junger, 2013; SAS Institute Inc., 2010).

The approach we use is the Balanced Repeated Replication (BRR) method, specifically Fay's method (Korn & Graubard, 2011). The BRR method utilizes half the sample at a time, including one of the two PSUs from each stratum. From this half sample, a parameter estimate is calculated and the original weights for the remaining PSUs are adjusted; these are called the replicate weights (SAS Institute Inc., 2010). Each replicate is calculated by deleting one PSU per stratum according to the corresponding Hadamard matrix which determines the combination of PSUs that will appear in a given replicate (2010). Estimates are then combined for each replicate weight to compute the variance.

Fay's extension of the BRR replaces half of the sample that is weighted as zero to instead be weighted by a factor weight,  $F$ . The factor weight, which is a proportion ranging from zero to one, weights half of the sample by  $F$  and the other half by  $2 - F$ . The variance estimate for Fay's BRR method is then computed as follows:

$$Var(\hat{\theta}) = \frac{1}{H} \frac{1}{(1 - F)^2} \sum_{h=1}^H (\hat{\theta}_h - \hat{\theta})^2 \quad (11)$$

where  $\hat{\theta}$  is the variance of the parameter of interest,  $\hat{\theta}_h$  is the variance of the parameter of interest for the half sample, and  $H$  is the number of half "balanced" samples as determined by the Hadamard matrix. Standard errors are then calculated as follows:

$$SE(\hat{\theta}) = \frac{\sqrt{Var(\hat{\theta})}}{\sqrt{N}}. \quad (12)$$

Fay's BRR method was developed specifically for two PSUs per stratum. Thus, those strata with more than two PSUs have the PSUs in the respective stratum randomly grouped into two groups for analysis (Barbosa, Sichieri, & Junger, 2013). A weighted factor of  $F = 0.3$  is chosen for variance estimation based on literary recommendation and to be consistent with previous studies (Barbosa, Sichieri, & Junger, 2013; Buckman, Parsons, & Kahle, 2016; Graubard & Korn, 1999; “NHANES Dietary Web Tutorial: Modeling Usual Intake Using Dietary Recall Data: Task 4,” n.d.; Wolter, 1985). This process can result in  $2^S$  replicates, where  $S$  is the strata. The number of replicates necessary for Fay's BRR is the smallest integer that is divisible by four and is greater than or equal to  $S$  (“NHANES Dietary Web Tutorial: Modeling Usual Intake Using Dietary Recall Data: Task 4,” n.d.).

Fay's BRR method is deemed appropriate for the standard error estimation due to the ease of implementation and provision of consistency in variance estimation methods across literature (Barbosa, Sichieri, & Junger, 2013; Buckman, Parsons, & Kahle, 2016). A potential limitation of utilizing Fay's BRR method is its tendency to underestimate the standard error for small sample sizes (Paben, 1999); however, for large national surveys, this concern will not apply.

Refer to Appendices C and D for SAS coding of Fay's method and replicate generation.

## RESULTS

### Simulation Study

A simulation study was performed to calculate and compare mean and percentile estimates to the user-generated truth, in addition to the accuracy of standard error estimates among existing methods: two-day means, NCI method (for continuous data), and our novel SAC method outlined in the “Methods” section. The design of the simulation follows closely to those previously performed for estimating usual dietary intakes from survey data (Tooze, 2010). Count data are simulated from the negative binomial hurdle with zero-inflation of approximately 20%. Furthermore, a case with very high zero-inflation of approximately 50% is simulated, which closely reflects the zero-inflation levels commonly seen in complex survey designs such as NHANES. The two parts of the model are linked via correlated person-specific errors which are bivariate-normally distributed. Thirty data sets are simulated from a simple random sampling scheme with 100 individuals in each sample and 10 days of observation for each individual. Truth was obtained from averaging the percentiles and means of 10 days of data for all 30 data sets combined. Simple two-day means were calculated at the individual level for two days of consumption and then averaging across the 30 simulations. The NCI method and SAC method were fit to the two days of data, and the mean of the 5th, 10th, 25th, 50th, 75th, 90th, and 95th percentiles were estimated and compared with truth and with two-day means. The NCI method within this simulation follows the methodology outlined in “The current gold standard method for estimating the distribution of usual nutrient intake from survey data: The NCI method” by treating the count outcome as continuous and normally distributed. The SAC method produces statistics via the



methodology outlined in the “Methods” section. While 30 simulations is small, the computational burden for the NCI and SAC methods is high.

Refer to Appendix E for SAS coding.

## ***Results***

Results of the simulation study with 20% zero-inflation are displayed in Tables 1, 2, and 3. Tables 1 and 2 summarize the average percentiles for each simulated sample based on the four estimation methods described previously; 1.) true percentiles based on the overall simulated sample, 2.) the two-day mean percentiles, 3.) the NCI method applied to count outcomes, and 4.) the SAC method. Figure 1 provides a graphical representation of the data presented in Table 1 and the mean percentiles' 95% confidence intervals based on the standard errors given in Table 1. Comparing the performance of the methods via simulation, the two-day mean percentiles reflect the true distribution more closely than other estimation methods utilized, which is expected due to the simple sampling design simulated. However, when comparing the SAC and NCI method, Tables 1 and 2 show the SAC method performed better than the NCI method by estimating the true distribution of the simulated count data more closely (less bias) (Table 2). The greatest accuracy was seen in the upper and lower percentiles of the count outcomes for the SAC method versus the NCI method (Table 1, Table 2, Figure 1).

Table 3 summarizes the average mean for each sample for the three methods compared to the truth. While the two-day distribution displayed in Table 1 more closely estimated the true distribution of the simulated data, the two-day mean overestimated the true average and had the greatest bias compared to other methods analyzed (Table 3). Overall, the

SAC method resulted in a mean estimate closer to the truth (least bias) than both the 2-day mean and NCI method (Table 3). A caveat to this is that the SAC method produced a slightly larger standard error for the mean estimate than the NCI method (Table 3).

Although the performance of the SAC is not perfect, it is better than the current gold standard NCI method for two reasons – estimates are less biased for the distribution of percentiles and mean (though the significance of the latter is called into question due to its small value) and SEs are more accurate due to the incorporation of BRR replication for complex survey designs. Also, when we considered the SAC method's performance under greater zero-inflation of 50% as shown in Tables 4, 5, and 6, we saw that the SAC method continued to perform better than the NCI method and a simple two-day estimate for distribution percentiles and mean with smaller SEs and/or less bias. Again, this was particularly pertinent for the lower and upper percentiles of the data distribution (Table 4 and Figure 2). We also note that with this simulation's unusually high proportion of zeros (50%) – a percentage reflective of what is seen in complex surveys measuring alcohol consumption – it is notoriously quite difficult to accurately model counts in settings where 50% zeros are expected (as data with 50% zeros are basically dichotomous data). For count data that are so highly skewed with a very large proportion of zero counts, one would likely present percentiles as opposed to model-based means. Finally, Figures 1 and 2 show that the SAC method did excellent in accurately estimating the mean alcohol consumption from survey data in the upper percentiles, which would be the cohort of interest to target for risk assessments, public health interventions, etc.

Table 1: Comparison of mean percentiles and their standard errors based on the simulated count data with 20% zero-inflation.

Estimation Method	Percentiles (SE)						
	5th	10th	25th	50th	75th	90th	95th
Truth <sup>a</sup>	0	0	1	1.967	3.567	6.333	8.733
Two-Day Mean <sup>b</sup>	0 (0)	0 (0)	0.9667 (0.0333)	1.933 (0.04632)	3.883 (0.07836)	7.200 (0.1688)	10.05 (0.2449)
NCI Method <sup>c</sup>	1.394 (0.06993)	1.625 (0.06856)	2.098 (0.06366)	2.748 (0.05410)	3.593 (0.07279)	4.534 (0.1429)	5.214 (0.2123)
SAC Method <sup>d</sup>	0.7468 (0.05833)	0.9345 (0.05908)	1.352 (0.05623)	2.055 (0.0501)	3.152 (0.07902)	4.607 (0.1769)	5.816 (0.2860)

<sup>a</sup>average of percentiles of 10 days of data for all 30 samples.

<sup>b</sup>average of percentiles of the first 2 days of data for all 30 samples.

<sup>c</sup>average of percentiles for 2 days of data with the outcome measured as a count.

<sup>d</sup>average of percentiles for 2 days of data with the outcome measured as a count.

Table 2: Comparison of mean percentiles and their bias based on the simulated count data with 20% zero-inflation.

Estimation Method	Percentiles (Bias)						
	5th	10th	25th	50th	75th	90th	95th
Truth <sup>a</sup>	0	0	1	1.967	3.567	6.333	8.733
Two-Day Mean <sup>b</sup>	0 (0)	0 (0)	0.9667 (-0.0333)	1.933 (-0.0340)	3.883 (0.3160)	7.200 (0.8670)	10.05 (1.317)
NCI Method <sup>c</sup>	1.394 (1.394)	1.625 (1.625)	2.098 (1.098)	2.748 (0.7810)	3.593 (0.02600)	4.534 (-1.796)	5.214 (-3.519)
SAC Method <sup>d</sup>	0.7468 (0.7468)	0.9345 (0.9345)	1.352 (0.3520)	2.055 (0.08800)	3.152 (-0.4150)	4.607 (-1.726)	5.816 (-2.917)

<sup>a</sup>average of percentiles of 10 days of data for all 30 samples.

<sup>b</sup>average of percentiles of the first 2 days of data for all 30 samples.

<sup>c</sup>average of percentiles for 2 days of data with the outcome measured as a count.

<sup>d</sup>average of percentiles for 2 days of data with the outcome measured as a count.

Table 3: Comparison of means, standard errors, and bias based on the simulated count data with 20% zero-inflation.

Estimation Method	Mean (SE)	Bias
Truth	2.736	
Two-Day Mean	3.002 (0.05410)	0.2660
NCI Method	2.958 (0.05470)	0.2220
SAC Method	2.516 (0.05690)	-0.2200

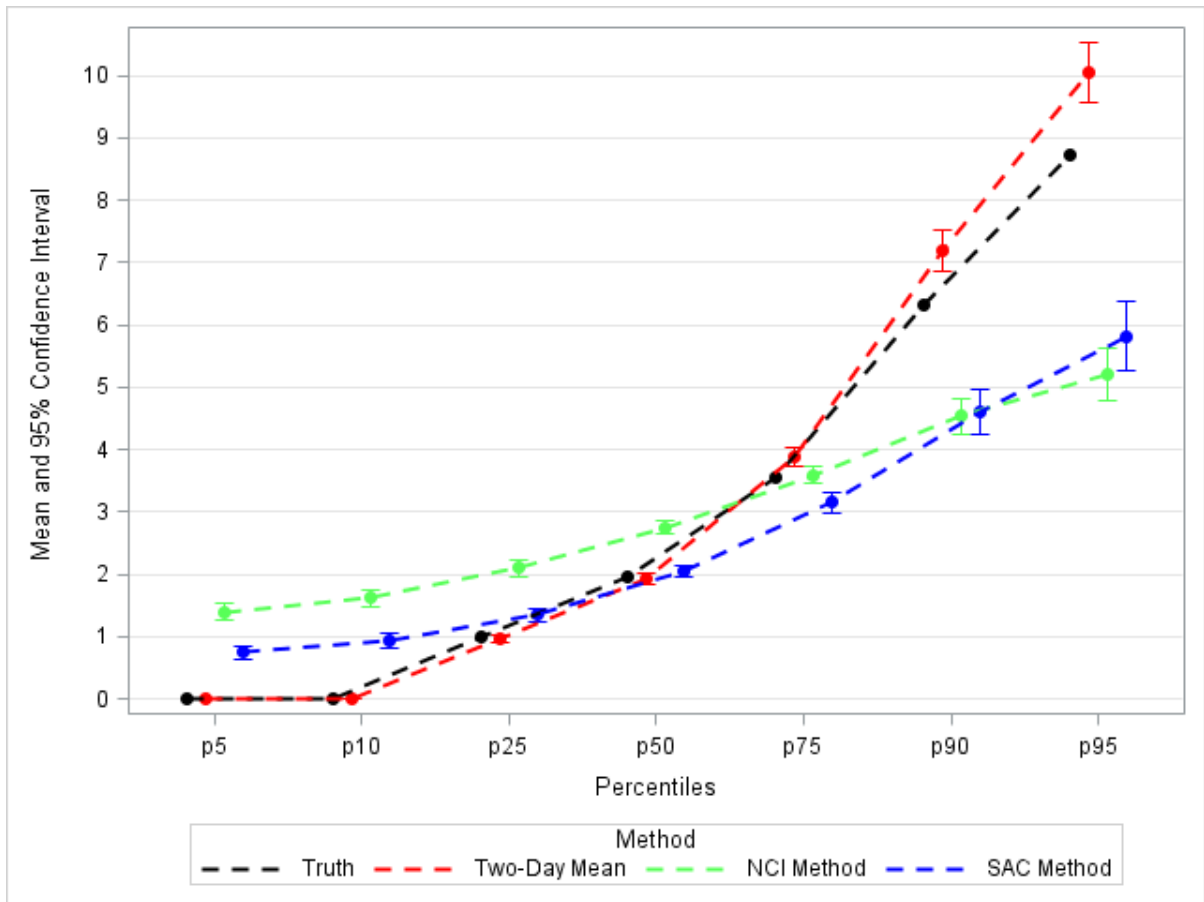


Figure 1: Comparison of selected mean percentiles and 95% confidence intervals for two-day mean, the NCI Method, and the SAC Method from simulated data with 20% zero-inflation.

Table 4: Comparison of mean percentiles and their standard errors based on the simulated count data with 50% zero-inflation.

Estimation Method	Percentiles (SE)						
	5th	10th	25th	50th	75th	90th	95th
Truth <sup>a</sup>	0	0	0	0	1	3.2	5.25
Two-Day Mean <sup>b</sup>	0 (0)	0 (0)	0 (0)	1.283 (0.2030)	3.683 (0.5681)	7 (1.129)	10.35 (1.721)
NCI Method <sup>c</sup>	1.0872 (0.07356)	1.295 (0.07265)	1.721 (0.06904)	2.319 (0.06026)	3.099 (0.07432)	4.013 (0.1338)	4.664 (0.1932)
SAC Method <sup>d</sup>	0.5625 (0.04932)	0.7213 (0.05459)	1.085 (0.06180)	1.697 (0.06800)	2.648 (0.1009)	3.977 (0.1933)	5.078 (0.2901)

<sup>a</sup>average of percentiles of 10 days of data for all 30 samples.

<sup>b</sup>average of percentiles of the first 2 days of data for all 30 samples.

<sup>c</sup>average of percentiles for 2 days of data with the outcome measured as a count.

<sup>d</sup>average of percentiles for 2 days of data with the outcome measured as a count.

Table 5: Comparison of mean percentiles and their bias based on the simulated count data with 50% zero-inflation.

Estimation Method	Percentiles (Bias)						
	5th	10th	25th	50th	75th	90th	95th
Truth <sup>a</sup>	0	0	0	0	1	3.2	5.25
Two-Day Mean <sup>b</sup>	0 (0)	0 (0)	0 (0)	1.283 (1.283)	3.683 (2.683)	7 (3.800)	10.35 (5.100)
NCI Method <sup>c</sup>	1.0872 (1.0872)	1.295 (1.295)	1.721 (1.721)	2.319 (2.319)	3.099 (2.099)	4.013 (0.8130)	4.664 (-0.5860)
SAC Method <sup>d</sup>	0.5625 (0.5625)	0.7213 (0.7213)	1.085 (1.085)	1.697 (1.697)	2.648 (1.648)	3.977 (0.7770)	5.078 (-0.1720)

<sup>a</sup>average of percentiles of 10 days of data for all 30 samples.

<sup>b</sup>average of percentiles of the first 2 days of data for all 30 samples.

<sup>c</sup>average of percentiles for 2 days of data with the outcome measured as a count.

<sup>d</sup>average of percentiles for 2 days of data with the outcome measured as a count.

Table 6: Comparison of means, standard errors, and bias based on the simulated count data with 50% zero-inflation.

Estimation Method	Mean (SE)	Bias
Truth	0.9953	
Two-Day Mean	2.839 (0.4739)	1.844
NCI Method	2.532 (0.05513)	1.537
SAC Method	2.120 (0.07842)	1.125

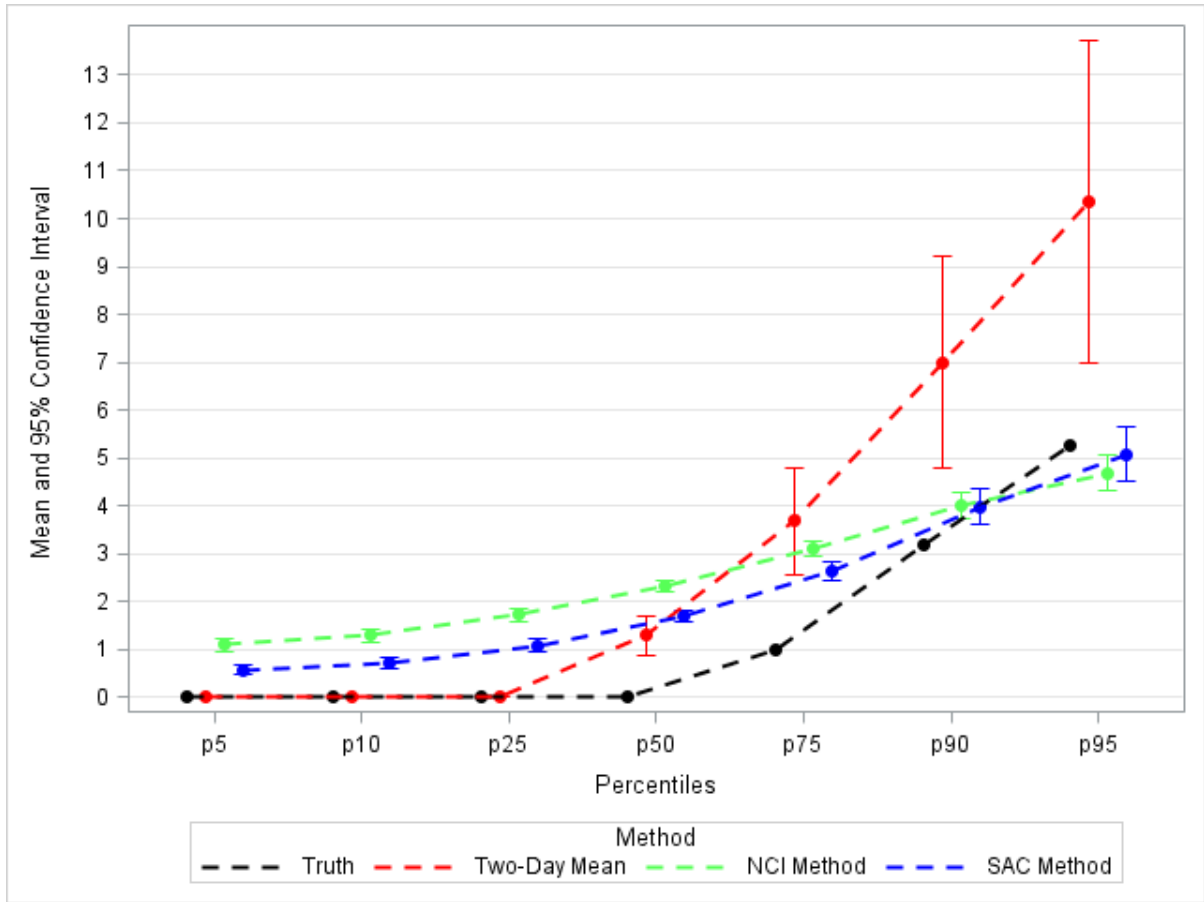


Figure 2: Comparison of selected mean percentiles and 95% confidence intervals for two-day mean, the NCI Method, and the SAC Method from simulated data with 50% zero-inflation.

### Data Application: Estimating the mean and distribution of alcohol consumption from NHANES recall

We analyzed the previously described National Health and Nutritional Examination Survey (NHANES) in order to present the first summarized population average daily consumption of alcohol. Specifically for illustration, we compared population average alcoholic consumption in consumers of  $\geq 19$  years of age for NHANES years 2003-2004 to 2015-2016 to observe any changes over time. The measure of interest was a surveyed



measure of alcohol consumption in grams over two 24 hour recall (24HR) time frames. All NHANES's questionnaires are interviewer-administered using the US Department of Agriculture (USDA) Automated Multiple-Pass Method (Butler, Poti, & Popkin, 2016).

Alcohol consumption rates within the 24HR measurements of NHANES are calculated using USDA-assigned food codes specific to each beverage reported. These reported food codes are then matched to food composition tables on the USDA Nutrient Database for Standard Reference from which the alcohol content in grams is collected based on the amount and type of alcoholic drink consumed, which is categorized into one of three categories: beer, wine, liquor/mixed drinks. A standard drink is defined as approximately 14 grams of pure alcohol (i.e. ethanol) as is cited by the NIAAA which can be found in approximately 12 fl oz of beer, 5 fl oz of wine, and 1.5 fl oz of distilled spirits (Guenther, Bowman, & Goldman, 2010; “What Is A Standard Drink? | NIAAA,” n.d). Alcohol content for mixed alcoholic beverages that contained multiple alcohol food codes is calculated by summing the grams of alcohol presented by each food code. Alcoholic beverages used as ingredients of flavorings in food preparation are excluded from the analyses (Butler, Poti, & Popkin, 2016; Guenther, P.M., Bowman, S.A., & Goldman, J.D., 2010).

For this analysis, alcohol consumption presented in the NHANES 24HR measurements was converted to count (its original form collected) by dividing each participant's 24HR measure by 14 grams to obtain one standard US drink. If required, these converted measurements were then rounded down to obtain integers. Covariates within the analysis include gender, age, if consumption of alcohol occurred during the weekend (i.e., the “weekend effect” discussed by NCI), and the longitudinal measure of time (in days).

Fay's BRR weights were calculated for the NHANES data utilizing the methods outlined in the section titled, “Estimation of standard errors of the distribution of survey count data”. Within the analysis, there were two unique time points from 2003-2004 and a total of 16 stratum; therefore, 16 replicate BRR weights were generated for analysis. The same numbers apply to the years 2015-2016. Following the model outlined in the section titled, “Model Building”, results consisted of parameter estimates for daily consumption in relation to covariates within the model, distribution estimates (mean and percentiles), and their adjusted standard errors as calculated using the components of the complex sampling design and Fay's BRR method.

Finally, further validation of the novel SAC method was performed by comparing our methodology's performance to those compared in “Simulation Study”. The baseline for comparison of estimation performance was the NCI method applied in its intended setting, where continuous alcohol consumption was converted to grams. The output of the distribution and mean was then converted to count from continuous grams using the NIAAA criteria of a standard drink (14 grams of pure alcohol = 1 standard drink).

## ***Results***

Descriptive statistics for NHANES cohorts 2003-2004 and 2015-2016 are displayed in Table 7. Covariates of interest in the analysis include gender, age in years, weekend (Friday, Saturday, or Sunday) versus weekday (Monday through Thursday) consumption of alcohol, and the longitudinal measure of time in days (i.e. 24 hours).

Of the 9,260 subjects analyzed between the two cohort years, the majority were female (52.08%) and had an approximate average age of 49 years. Overall, the average

alcohol consumption for the surveyed sample was approximately half a standard drink a day between the two days of observed drinking, with a higher average consumption of alcohol being observed during day 1 24HR (0.6923 standard drinks) compared to day 2 (0.4078 standard drinks) and with average weekday consumption of alcohol (52%) being slightly higher than average weekend consumption (48%) for all observation days over the two cohort years.

Table 7: Descriptive statistics for NHANES 2003-2004 and 2015-2016.

Variable	2003-2004	2015-2016
	% (N)	
Gender		
Male	47.73% (2,167)	48.09% (2,270)
Female	52.27% (2,373)	51.91% (2,450)
Day		
Weekend	48.16% (4,373)	47.89% (4,521)
Weekday	51.84% (4,707)	52.11% (4,919)
	Mean (SD)	
Age (years)	48.87 (20.23)	49.25 (18.06)
Alcohol Consumed (# standard drinks)	0.5873 (1.980)	0.5315 (1.680)
Day 1	0.7281 (2.176)	0.6573 (1.931)
Day 2	0.4345 (1.732)	0.3811 (1.304)

Application to NHANES of the model outlined in the “Model Building” section produced model parameter estimates summarized in Table 8 and 9 for the years 2003-2004 and 2015-2016 respectively. Tables 8 and 9 show output exactly as would be produced by

NCI's publicly available SAS Macro with the inclusion of odds ratios and rate ratios (Usual Dietary Intakes: SAS Macros for the NCI Method, 2018). Interpretation of parameters of the HNB requires two parts; the probability of consuming and the amount consumed. The probability of consuming (shown as “pr” under the name of the parameter estimate in Tables 8 and 9) follows a binomial distribution; therefore, interpretation of probability parameter estimates are in terms of the log-odds of alcohol consumption holding all other variables constant and their converted odds ratios (Table 8 and 9). The amount consumed (i.e. “am” shown under the name of the parameter estimate in Tables 8 and 9) by the HNB model, does not have an interpretation dependent on the performance of the previous probability component. Therefore, the amount consumed is interpreted for a negative binomial distribution and is the difference in logs of expected counts of standard drinks of alcohol consumed, holding all other variables at zero. These parameter estimates can also be converted to rate ratios and interpreted in terms of the mean counts of standard drinks in a day from the untruncated (negative binomial) distribution holding all other variables at zero (Table 8 and 9).

Referring to Table 8, the probability of consumption shows that, when holding all other variables as zero, females had a decreased odds of alcohol consumption on a given day compared to males (OR = 0.1899). Similarly, an increase in age by a year (OR = 0.9879) as well as the effect over time between measured consumption days 1 and 2 (OR = 0.5813) also decreased the odds of alcohol consumption when holding all other variables as zero respectively. However, the status of the recall day being a weekend more than doubled the odds of alcohol consumption compared to the recall day status being a weekday (OR =

2.384) holding all other variables constant. Regarding the amount consumed, and holding all other variables at zero for all interpretations, females have a decreased rate of consumption compared to males ( $RR = 0.5500$ ). For each year increase in age, a respondent's rate for standard drinks of alcohol consumption also decreased ( $RR = 0.9859$ ). It being the weekend increased the rate of alcohol consumption compared to it being a weekday ( $RR = 1.100$ ). Finally, longitudinally the difference in the rate of consumption for a standard alcoholic drink also decreased over the two days of observations ( $RR = 0.8118$ ). This last observation could be associated with or slightly collinear with the weekend effect.

The interpretation of Table 9 is directionally the same as that for Table 8 for both the probability and amount components of the model parameter estimates. However, the magnitude of the effect on parameter estimates shows a difference between 2003-2004 and 2015-2016. For example, the protective effects of gender (female) and the longitudinal time measurement in days was greater for the 2003-2004 sample than for the 2015-2016 sample of NHANES in decreasing the probability of alcohol consumption. Also, though weekend status did increase the probability of consumption for both cohort years, there was a smaller magnitude of effect for the 2003-2004 sample versus the 2015-2016 sample.

Table 8: SAS model estimates, odds ratios, and rate ratios for the negative binomial hurdle model for population average alcohol consumption of NHANES 2003-2004.

Parameter	Name	Estimate	OR	RR
$\beta_{10}$	Intercept–pr	-1.6614	0.1899	
$\beta_{GENDER1}$	Gender (female) –pr	-1.9103	0.1480	
$\beta_{RIDAGEYR1}$	Age (year)–pr	-0.01219	0.9879	
$\beta_{WEEKEND1}$	Weekend Status (weekend)–pr	0.8689	2.384	
$\beta_{DAY12}$	Day (day2) –pr	-0.5425	0.5813	
$\sigma_{\mu_1}^2$ <sup>a</sup>	Reparam Var(u1)–pr	1.0628		
$\beta_{20}$	Intercept–am	1.2106		3.355
$\beta_{GENDER2}$	Gender (female)–am	-0.5978		0.5500
$\beta_{RIDAGEYR2}$	Age (year) –am	-0.01421		0.9859
$\beta_{WEEKEND2}$	Weekend Status (weekend) –am	0.09557		1.100
$\beta_{DAY22}$	Day (day2) –am	-0.2085		0.8118
$\sigma_{\mu_2}^2$ <sup>b</sup>	Reparam Var(u2)–am	-0.2939		
$k$	Inverse Dispersion–am	0.1218		
$Z_{\rho}$	Z-trans of Correlation	0.6944		

<sup>a</sup>Random effect variance parameter estimate for probability.

<sup>b</sup>Random effect variance parameter estimate for amount.

Table 9: SAS model estimates, odds ratios, and rate ratios for the negative binomial hurdle model for population average alcohol consumption of NHANES 2015-2016.

Parameter	Name	Estimate	OR	RR
$\beta_{10}$	Intercept–pr	-1.8727	0.1537	
$\beta_{GENDER1}$	Gender (female)–pr	-1.2143	0.2969	
$\beta_{RIDAGEYR1}$	Age (year)–pr	-0.0090	0.9910	
$\beta_{WEEKEND1}$	Weekend Status (weekend)–pr	0.9153	2.497	
$\beta_{DAY12}$	Day (day2)–pr	-0.2249	0.7986	
$\sigma_{\mu_1}^2$ <sup>a</sup>	Reparam Var(u1)–pr	1.0217		
$\beta_{20}$	Intercept–am	1.0785		2.940
$\beta_{GENDER2}$	Gender (female)–am	-0.6105		0.5431
$\beta_{RIDAGEYR2}$	Age (year)–am	-0.0110		0.9891
$\beta_{WEEKEND2}$	Weekend Status (weekend)–am	0.0920		1.096
$\beta_{DAY22}$	Day (day2)–am	-0.1777		0.8372
$\sigma_{\mu_2}^2$ <sup>b</sup>	Reparam Var(u2)–am	-0.3454		
$k$	Inverse Dispersion–am	0.0832		
$Z_{\rho}$	Z-trans of Correlation	0.5545		

<sup>a</sup>Random effect variance parameter estimate for probability.

<sup>b</sup>Random effect variance parameter estimate for amount.

The NHANES's data outputs and parameter estimates from Table 8's model are used to estimate the distribution and percentiles of alcohol consumption in participants aged 19 years and older from the years 2003-2004. Results in Table 10 mirror exactly the output produced from the publicly available NCI SAS macro (Usual Dietary Intakes: SAS Macros

for the NCI Method, 2018). Table 10 shows the average consumption of alcohol, distribution (percentiles) of consumption, and associated standard errors using the BRR methodology for our SAC approach (“Estimation of standard errors of the distribution of survey count data”). Also analyzed is the cut point probability (i.e. "cutprob1") for half of one standard serving of alcohol (Table 10). Table 7's HNB model parameter estimates are utilized to estimate the distribution of average alcoholic consumption in the same subset of 19 years of age and older from the years 2015-2016. The distribution estimates in the form of mean alcohol consumption, percentiles of consumption, and cut point probability (i.e. "cutprob1") for 0.5 standard serving of alcoholic drink is displayed in Table 9 alongside their associated standard error estimates (Table 11).

As is shown in Table 10, in 2003-2004, the US population average consumption of alcohol was 0.715 standard drinks daily with a standard error (SE) of 0.03956. The median daily consumption of alcohol was 0.1090 standard drinks (SE = 0.03291), or approximately zero standard drinks for the population. One sees a dramatic increase in the number of standard drinks consumed at the 90th percentile of the US population and greater, with the average daily number of standard drinks greater than two for the 90th percentile and approximately four for the 95th percentile. Using the cut point of 0.5 standard serving of alcohol daily, approximately 70% (SE = 0.01949) of the US population drinks this amount of alcohol or less daily (Table 8).

In comparison, US population average consumption of alcohol for the years 2015-2016 (Table 11) decreased compared to 2003-2004, with the population drinking 0.6398 standard drinks in a day (SE = 0.04467). There was also a decrease in median daily



consumption of alcohol in 2015-2016 compared to 2003-2004 with 50% of the population drinking 0.1048 standard drinks or less daily, and with smaller standard error (SE = 0.02732). Table 11 shows that the percentiles of drinking in the US population in 2015-2016 improved from NHANES years 2003-2004. Finally, there was also an increase in the percent of the 2015-2016 population who drink the cut point of 0.5 standard serving of alcohol daily (72%, SE = 0.02122) (Table 11).

Table 10: SAS distribution estimates and standard errors for population average alcohol consumption of NHANES 2003-2004.

	Mean	N	Percentiles							cutprob1
			5th	10th	25th	50th	75th	90th	95th	
Estimate	0.7510	4719	0.0003888	0.001357	0.01110	0.1090	0.7373	2.246	3.660	0.6967
SE	0.03956		0.0003847	0.001098	0.006214	0.03291	0.08049	0.1282	0.2122	0.01949

Table 11: SAS distribution estimates and standard errors for population average alcohol consumption of NHANES 2015-2016.

	Mean	N	Percentiles							cutprob1
			5th	10th	25th	50th	75th	90th	95th	
Estimate	0.63984	5139	0.0004912	0.001625	0.01214	0.1048	0.6358	1.880	3.070	0.7158
SE	0.04467		0.0004030	0.001109	0.005589	0.02732	0.08808	0.1561	0.2218	0.02122

For completeness, Table 12 and Figure 3 show alcohol consumption for NHANES 2003-2016 across five estimation approaches; 1.) the NCI method applied to continuous alcohol consumption available via NHANES (grams of alcohol), 2.) two-day standard mean, 3.) survey mean, 4.) the NCI method applied to counts (# of standard drinks), and 5.) the

SAC method. As we do not know the truth, we cannot state which method is correct.

However, due to the validation of the NCI method in application to continuous outcomes, we could apply these results as a baseline of measure for performance. Based on this, it is clear our novel SAC and the NCI method for continuous standard drinks provided fairly similar results, with some differences noted at the upper end of the distribution. The NCI method innapropriately applied to count data likely incorrectly estimated the distribution and, therefore, is not recommended for use with count data.

Table 12: Estimated percentile of alcohol consumption per day for adults  $\geq 19$  years, estimated using five methods. Data are from the National Health and Nutritional Examination Survey 2003-2016.

Estimation Method	Mean	5th	10th	25th	50th	75th	90th	95th
NCI Method (cont. standardized) <sup>a</sup>	1.085	0.004820	0.01336	0.07330	0.4359	1.642	3.227	4.155
Standard Mean	0.5873	0	0	0	0	0	2	4
Survey Mean	0.6875	0	0	0	0	0	1.967	3.611
NCI Method <sup>b</sup>	1.161	0.0003743	0.001193	0.01153	0.1924	1.570	4.031	5.393
SAC Method	1.061	0.001246	0.004081	0.02963	0.2449	1.234	3.111	4.737

<sup>a</sup>run continuous alcohol (g) with NCI method and then percentiles and mean divided by #g of alcohol in standard drink (14 g = 1 standard drink)

<sup>b</sup>run converted alcohol count (14 g = 1 standard drink) with NCI method

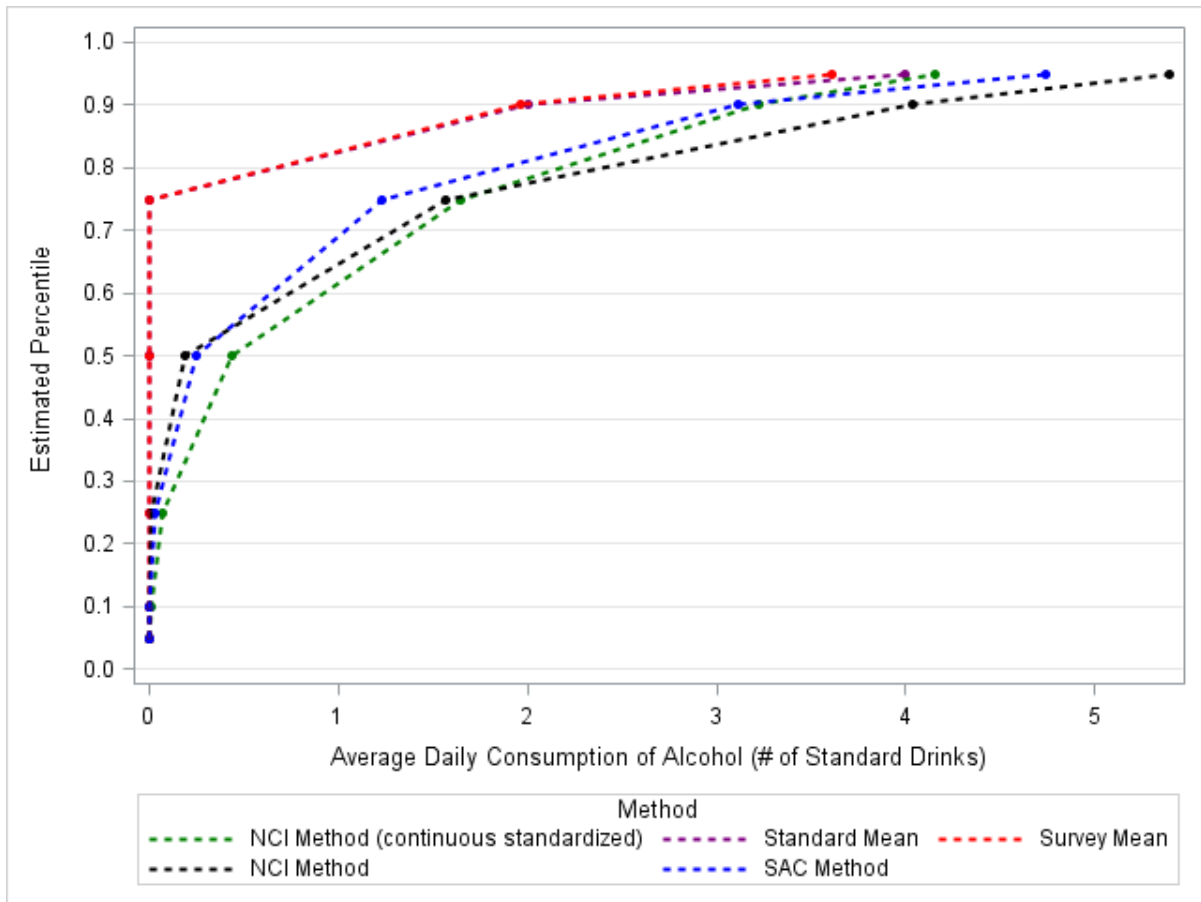


Figure 3: Estimated percentile of alcohol consumption per day for adults  $\geq 19$  years, estimated using five methods. Data are from the National Health and Nutritional Examination Survey 2003-2016.

### Data Application: Estimating the mean and distribution of alcohol consumption from NLSY 1997 recall

We conducted an analysis applying the SAC method to the National Longitudinal Survey of Youth (NLSY) 1997 in order to present summarized population average daily consumption of alcohol over a lifespan.

The NLSY, like NHANES, utilizes a complex multi-stage sampling design with survey weights, clusters, and strata. The NLSY 1997 is collected by the Bureau of Labor Statistics, under the National Longitudinal Surveys program, and follows the lives of a sample cohort of US youth born between 1980 and 1984. All NLSY surveys are interviewer-administered utilizing a computer-assisted personal interview (CAPI) instrument. For the current analysis, we used NLSY 1997 data from the years 2013 to 2015. The measure of interest for each year was the number of standard alcoholic beverages consumed over the last 30 days.

Specifically, the measure is assessed by the following questions:

1. Have you had a drink of an alcoholic beverage in the past twelve months? (By a drink we mean a can or bottle of beer, a glass of wine, a mixed drink, or a shot of liquor.)
2. During the last 30 days, on how many days did you have one or more drinks of an alcoholic beverage?
3. In the past 30 days, on the days you drank alcohol, about how many drinks did you usually have?

Based on answers to question 1, CAPI then leads the respondent to question 2 if answered "yes". Any response to question 2 other than zero then leads the respondent to question 3 which is measured as a count. If respondents answered "no" to question 1 or "0" or "Valid Skip" to question 2, then they are recorded as having zero standard drinks of alcohol for question 3 (refer to Figure 4). A standard drink is defined as approximately 14 grams of pure alcohol (i.e. ethanol) as is cited by the NIAAA which can be found in approximately 12 fl oz of beer, 5 fl oz of wine, and 1.5 fl oz of distilled spirits (Guenther, Bowman, & Goldman, 2010; "What Is A Standard Drink? | NIAAA," n.d). The variable derived for analysis was the

number of standard drinks consumed over the 30-day period prior to interview as recalled in question 3. Covariates for our analysis included gender, race/ethnicity (Black, Hispanic, Mixed Race Non-Hispanic, Non-Black/Non-Hispanic), dichotomous risky drinking status as defined by the NIAAA and SAMHSA, and longitudinal time in years. Risky drinking was defined as more than four and five drinks in a day for females and males, respectively. (“Drinking Levels Defined,” 2011).

Fay's BRR weights were calculated for the NLSY 1997 utilizing the methods outlined in the section titled, “Estimation of standard errors of the distribution of survey count data”. Within the analysis, there were two unique time points (2013 and 2015) and a total of 117 stratum; therefore, 120 replicate BRR weights were generated for analysis. Results are presented as parameter estimates for monthly alcohol consumption by the above-mentioned covariates, mean and distribution estimates (percentiles), and adjusted standard errors calculated using weight, cluster, and strata components of the sampling design and Fay's BRR method. Furthermore, subgroup analysis was performed based on risky drinking status, resulting in mean and distribution estimates for each subgroup categorization.

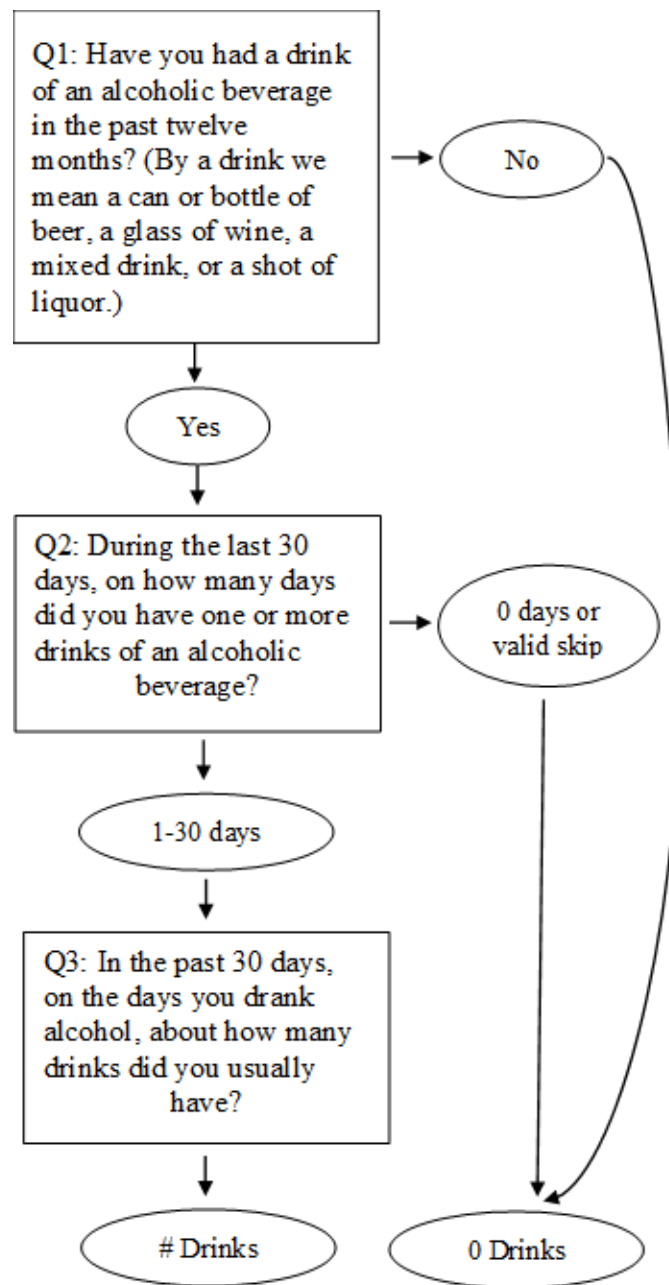


Figure 4: Diagram of NLSY 1997 measurement of alcohol as a count outcome.

### **Results**

Descriptive statistics for NLSY 1997 waves 2013-2015 are shown in Table 13. Of the 4,456 subjects analyzed for the two waves of survey years, the majority were female

(55.41%), Non-Black/Non-Hispanic (52.94%), and did not exhibit risky drinking as defined by the NIAAA and SAMHSA. The average alcohol consumption for the population was approximately 1.7 standard drinks a day on drinking days and 13 standard drinks in a month. Slightly higher alcohol consumption in both drinking days and in a month period was observed in 2013 versus 2015, as well as a greater number of drinking days (Table 13).

Table 13: Descriptive statistics for NLSY 1997 waves 2013-2015.

Variable	% (N)
Gender	
Male	44.59% (1,987)
Female	55.41% (2,469)
Race/Ethnicity	
Black	25.65% (1,143)
Hispanic	20.38% (908)
Mixed Race (Non-Hispanic)	1.03% (46)
Non-Black/ Non-Hispanic	52.94% (2,359)
Risky Drinking Status	
Risky Drinking	5.80% (517)
Non-Risky Drinking	94.20% (8,395)
	Mean (SD)
#Drinking Days in Month	4.675 (6.504)
Year 2013	4.720 (6.427)
Year 2015	4.630 (6.580)
#Drinks on Drinking Days	1.781 (1.973)
Year 2013	1.831 (2.039)
Year 2015	1.730 (1.903)
# Drinks in Month	13.09 (20.88)
Year 2013	13.35 (20.89)
Year 2015	12.83 (20.87)

The application of the model outlined in the “Model Building” section produced model parameter estimates summarized in Table 14. Table 14 shows output exactly as would



be produced by the NCI's publicly available SAS Macro with the inclusion of odds and rate ratios (Usual Dietary Intakes: SAS Macros for the NCI Method, 2018). Interpretation of parameters of the HNB has two parts; the probability of consuming and the amount consumed which is not dependent on the results of the probability component. Interpretation follows that described in “Data Application: Estimating mean and distribution of alcohol consumption from NHANES recall”.

Referring to Table 14, all the following interpretations are under the condition of holding all other variables as zero. Females had decreased odds of alcohol consumption in a 30-day period compared to males (OR = 0.5378). The effect over time of measured consumption 30-day periods between waves 2013 and 2015 also decreased the odds of alcohol consumption (OR = 0.9101). However, the effect of race/ethnicity on the probability of alcohol consumption was dependent on the groups compared. Hispanic respondents had a decreased odds of alcohol consumption in the period of interest compared to Black respondents (OR = 0.5378), while Mixed Race (Non-Hispanic) and Non-Black/Non-Hispanic had increased odds of alcohol consumption compared to Black respondents (OR = 1.228 and OR = 3.414 respectively). Finally, being characterized with risky drinking status dramatically increased the odds of alcohol consumption compared to being characterized with non-risky drinking status (OR = 872.53). Regarding the amount consumed, females had a decreased rate of alcohol consumption compared to males (RR = 0.5175). Longitudinally the difference in the rate of alcohol consumption in a 30-day period for a standard alcohol drink decreased over the two year waves of observation (RR = 0.8894). When comparing race/ethnicity, Hispanic and Mixed Race (Non-Hispanic) respondents had a decreased rate of consumption

compared to Black respondents ( $RR = 0.8850$  and  $RR = 0.8547$  respectively). However, Non-Black/Non-Hispanic respondents had an increased rate of consumption compared to Black respondents ( $RR = 3.414$ ). Finally, those respondents with risky drinking status had an increased rate of alcohol consumption compared to respondents with non-risky drinking status ( $RR = 2.806$ ).

Table 14: SAS model estimates, odds ratios, and rate ratios for the negative binomial hurdle model for population average alcohol consumption of NLSY 1997 waves 2013-2015.

Parameter	Name	Estimate	OR	RR
$\beta_{10}$	Intercept–pr	1.9366	6.935	
$\beta_{GENDER1}$	Gender (female)–pr	-0.6203	0.5378	
$\beta_{RETH11}$	Race/Ethnicity (Hispanic)–pr	-0.7706	0.4627	
$\beta_{RETH12}$	Race/Ethnicity (Mixed Race (Non-Hispanic))–pr	0.2053	1.228	
$\beta_{RETH13}$	Race/Ethnicity (Non-Black / Non-Hispanic)–pr	0.3280	3.414	
$\beta_{RISKY1}$	Risky Drinking (yes)–pr	6.7714	872.53	
$\beta_{YEAR12}$	Year (year2)–pr	-0.0942	0.9101	
$\sigma_{\mu_1}^2$ <sup>a</sup>	Reparam Var(u1)–pr	1.0722		
$\beta_{20}$	Intercept–am	2.8367		17.06
$\beta_{GENDER2}$	Gender (female)–am	-0.6588		0.5175
$\beta_{RETH21}$	Race/Ethnicity (Hispanic)–am	-0.1222		0.8850
$\beta_{RETH22}$	Race/Ethnicity (Mixed Race (Non-Hispanic))–am	-0.1570		0.8547
$\beta_{RETH23}$	Race/Ethnicity (Non-Black / Non-Hispanic)–am	0.5669		1.763
$\beta_{RISKY2}$	Risky Drinking (yes)–am	1.0318		2.806
$\beta_{YEAR22}$	Year (year2)–am	-0.1172		0.8894
$\sigma_{\mu_2}^2$ <sup>b</sup>	Reparam Var(u2)–am	-0.1296		
$k$	Inverse Dispersion–am	0.3242		
$Z_\rho$	Z-trans of Correlation	0.4710		

<sup>a</sup>Random effect variance parameter estimate for probability.

<sup>b</sup>Random effect variance parameter estimate for amount.

The NLSY 1997's data outputs and parameter estimates from the model presented in Table 14 are used to estimate the distribution and percentiles of consumption for alcohol in survey participants for the waves 2013-2015. Results in Table 15 mirror exactly the output produced from the publicly available NCI SAS macro (Usual Dietary Intakes: SAS Macros for the NCI Method, 2018). Table 15 shows the average consumption of alcohol in standard drinks for a 30-day period, distribution (percentiles) of consumption, and associated standard errors using the BRR methodology for our SAC approach (refer to “Estimation of Standard Errors of the Distribution of Survey Count Data”).

As is shown in Table 15, from waves 2013-2015, the population average consumption of alcohol was 17.24 standard drinks in a 30-day period with a SE of 3.128. The median 30-day consumption of alcohol was 8.711 standard drinks (SE = 1.687), or approximately 9 standard drinks for the population. One sees a dramatic increase in the number of standard drinks consumed in a 30-day period at the 90th percentile of the population and greater, with the average consumption being approximately 20 standard drinks for the 90th percentile and approximately 61 standard drinks for the 95th percentile.

Table 15: SAS distribution estimates and standard errors for population average alcohol consumption of NLSY 1997 waves 2013-2015.

	Mean	N	Percentiles							cutprob1
			5th	10th	25th	50th	75th	90th	95th	
Estimate	17.24	4456	0.1939	0.6229	2.909	8.711	20.46	40.96	61.45	
SE	3.128		0.06680	0.1787	0.6589	1.687	3.751	7.410	11.29	

We further expanded on results presented formerly in Tables 14 and 15 by showing the distribution of alcoholic consumption for NLSY 1997 waves 2013-2015 categorized by NIAAA's definition of risky drinking. As shown in Table 16, the average US consumption of alcohol for those who do not exhibit risky drinking was 15.28 standard drinks in a 30-day period with a median consumption of 7.764 standard drinks. In comparison, those who exhibit risky drinking had a dramatically higher average consumption of alcohol: 51.94 standard drinks in a 30-day period with a median consumption of 31.57 standard drinks. This trend continued for the upper percentiles of consumption for risky drinkers (Table 16).

Table 16: SAS distribution estimates categorized by risky drinking status for population average alcohol consumption of NLSY 1997 waves 2013-2015.

Risky Drinking	Mean	N	Percentiles							cutprob1
			5th	10th	25th	50th	75th	90th	95th	
No	15.28	4179	0.2225	0.6433	2.628	7.764	18.42	36.76	54.64	
Yes	51.94	277	6.080	8.825	15.98	31.57	62.17	114.51	164.16	

## DISCUSSION

Although there are several available methods to determine the mean consumption of foods (Dodd et al., 2006; Nusser, Carriquiry, Jensen, & Fuller, 1990; Tooze et al., 2010), these methods do not immediately lend themselves to estimating consumption of substances measured as counts, nor do they accommodate all four-stages of the survey sampling design. The NCI method is the gold-standard for estimating continuous food consumption; however, our novel approach is more appropriate for estimating substance consumption, as the model

applied is uniquely suited to zero-inflated count data captured via complex survey designs, and, like the NCI method, accounts for measurement error inherent in self-report via the use of a two-part model with random subject-specific effects.

Our simulation study showed that the SAC method is comparable to the NCI method but these may provide different results for items measured as counts. In simulating a simple random survey with varying levels of zero-inflation, the SAC method overall provided distribution and mean estimates with less bias from the truth. Notably, the NCI method produced similar results to the SAC method when we used a continuous measure of drinking (grams). However, the NCI method overestimated average consumption compared to the SAC approach, both cross-sectionally (NHANES) and longitudinally (NLSY 1997).

The utility and performance of the SAC method were demonstrated on multiple cross-sectional waves of NHANES data and on the longitudinal survey observations of NLSY 1997. When analyzing NHANES data, the SAC method had a similar computational burden as the NCI method. Additionally, the SAC method produced more meaningful distributions of daily alcohol consumption, with near zero for most of the population and high values at the upper percentiles. We also showed that over a decade (from 2003 to 2016), the mean and distributions of alcohol consumption in the US population has decreased. For NLSY 1997, we presented an innovative way to accurately estimate lifetime alcohol consumption, the summary of which can be used for risk assessments (e.g., to measure the association between lifetime alcohol intake and prospectively collected outcomes), and trends over time can be assessed. Furthermore, the SAC method lends itself to analysis based on categorizations of risky drinking, showing the extreme differences in average and percentiles of alcohol

consumption in the risky and non-risky drinking subgroups. Application of this approach is useful to substance use researchers for risk assessments, or to provide markers of intervention effectiveness. The method also accommodates covariates, which allowed us to demonstrate that being female and time (in both days and years) are protective factors for both the probability of alcohol consumption and the amount consumed. Future analyses using our software operationalized as SAS macros are limitless - the software may be used to determine subgroup effects, facilitate risk assessments in cross-sectional or prospective epidemiologic studies, and use this information for public health recommendations. As NHANES geospatial data are available under certain conditions, our method could straightforwardly be adapted in a spatio-temporal context.

## **CONCLUSION**

The novel SAC method utilizes a two-part model for zero-inflated counts that is applicable multi-stage, complex survey designs, enabling investigators to measure the consumption of licit and illicit substances in the US population - e.g., alcohol, cigarettes, marijuana, which are typically recorded in epidemiologic surveys as count variables. This dissertation investigated the SAC method's performance versus currently available methods that do not handle counts and do not account for the full survey design. Via simulation and application, it is clear the SAC method performs better than the current gold standard for continuous data (the NCI method) which, unlike the SAC, does not account for the full survey design. The utility of the SAC approach was demonstrated in estimating alcohol consumption in the US from both cross-sectional and longitudinal epidemiologic surveys.

There are some limitations to the current study. Estimates within the simulation study were not directly comparable to the distribution of the simulated truth. However, the simulation scenarios we performed closely follow those presented to validate the NCI method, and while the SAC method did not closely reflect the true simulated estimates of consumption, the method performed better than existing gold-standard methodologies or simple two-day means. Another limitation of the SAC method is its reliance on the assumption that the measure of count outcome utilized within survey methodology is a true measurement of the consumption for an individual. This is partly mitigated by the inclusion of random effects in both parts of the count model. While survey methods can be validated externally, such as the survey methodology for NHANES, there is still the risk of bias such as recall bias, social-desirability bias, and performance bias, especially in the case of alcohol consumption. Studies have dealt with such bias by including latent variables within the modeling approach to quantify the extent of bias (Cagnone & Viroli, 2018; Duncan, Duncan, & Hops, 1998). It is also of note that though the outcome of interest for the SAC method is measured as a count, the output of the methodology is on a continuous spectrum. This characteristic of the method's distribution outputs allows for a larger breadth of detail to be gleaned from the methodology where, if the distribution were to be outputted as a count, we would see a large portion of percentiles being valued as zero due to high zero-inflation in the data. Instead, with the final measure of standard drinks measured as continuous, the nuances of changes in alcohol consumption for the population can be recorded over time and with the introduction of interventions. Finally, a limitation concerning the application of the methodology to NHANES is that participants' daily consumption showed a higher proportion



of weekend drinking, thus leaving as an open question the interpretability of daily consumption (as drinking primarily occurs on weekends while survey questionnaires are administered on weekdays). However, the inclusion of the weekend effect as a covariate in analyses, as well as separate calculation of weekend versus weekday effect in the estimation of the distribution consumption does mitigate the issue. This ability of our model to adjust for weekend effects is, in fact, a relevant strength since many studies have established that consumption is higher on weekends (Hughes, Fingar, Budney, Naud, & Helzer, 2014; Lau-Barraco, Braitman, Linden-Carmichael, & Stamates, 2016).

While we introduced a novel methodology and SAS macro, there is still an opportunity for further research. Future goals are to develop a methodology and corresponding statistical package that accurately predicts individual substance consumption for use in disease models, and to extend the above work to accommodate spatio-temporal information available upon request from survey data.

## APPENDICES

### Appendix A: SAC Method MIXTRAN SAS Macro for model estimation

```

/*****
/*****
/*
/*   THE MIXTRAN MACRO
/*
/*****
/*
/*           VERSION 4           01/22/2020
/*
/*
/* This MIXTRAN macro is used for the analysis of substance intakes
/* measured as a count outcome and specifically characterized by an
/* excess of zero observations. Output from the MIXTRAN macro is used
/* by the DISTRIB macro for estimation of the distribution of usual
/* intake.
/*
/*
/* For substance intakes measured as a count, the MIXTRAN macro fits a
/* two-part Negative Binomial Hurdle Model where the first part
/* considers the probability of consumption and the second part
/* considers the consumption-day amount. The model allows for
/* covariates and includes a random effect in both parts and allows
/* for correlation between the random effects.
/*
/*
/* To fit this two-part mixed model with correlated random effects
/* (i.e. the correlated model), starting values for the two parts of
/* the model are obtained by first using the GENMOD procedure to fit
/* a probability model and an amount model. Then a negative binomial
/* mixed model with uncorrelated random effects (i.e. the uncorrelated
/* model) is fit using two calls to the NLMIXED procedure, and the
/* parameter estimates from this model are used as starting values
/* for the correlated model.
/*
/*
/* The syntax for calling the macro is:
/*
/* %MIXTRAN
/* (data=, response=, foodtype=, subject=, repeat=,
/* covars_prob=, covars_amt=, outlib=, modeltype=,
/* lambda=, replicate_var=, seq=, dist=,
/* weekend=, vargroup=, numvargroups=,
/* start val1=, start val2=, start val3=, vcontrol=,
/* nloptions=, titles=, printlevel=, subgroup=)
/*
/* where
/*
/*   "data"           * Specifies the dataset to be used.
/*
/*   "response"       * Specifies the longitudinal recall variable.
/*
/*   "foodtype"       * Specifies a name for the analysis, used to

```

```

/*          identify the output data sets.  This value can          */
/*          be the same as the response variable.                  */
/*          */
/* "subject"      * Specifies the variable that uniquely          */
/*                  identifies each subject.                      */
/*          */
/* "repeat"      * Specifies the variable that indexes repeated    */
/*                  observations for each subject.                */
/*          */
/* "covars_prob"  Specifies a list of covariates for the first     */
/*                  part of the model that models the probability  */
/*                  of consumption.  Covariates must be separated */
/*                  by spaces.  Interactions must be in the order  */
/*                  specified by PROC GENMOD.  If the model type    */
/*                  is "amount" then covars_prob should be left as */
/*                  a null string.                                */
/*          */
/* "covars_amt"   * Specifies a list of covariates for the second  */
/*                  part of the model that models the consumption- */
/*                  day amount.  Covariates must be separated by  */
/*                  spaces.  Interactions must be in the order    */
/*                  specified by PROC GENMOD.                      */
/*          */
/* "outlib"      * Specifies a directory where output data sets   */
/*                  are stored.  Outlib can not be null.          */
/*          */
/* "modeltype"    * Specifies the model.  The possible values are: */
/*                  "null string" = fit correlated model,          */
/*                  "corr"       = fit correlated model,          */
/*                  "nocorr"     = fit uncorrelated model,         */
/*                  "amount"     = fit amount-only model.          */
/*          */
/* "lambda"      * Specifies a user supplied value for the        */
/*                  Box-Cox transformation parameter, lambda.  If */
/*                  a value is not supplied, the macro will        */
/*                  calculate a value for lambda.  For HNB or ZINB  */
/*                  model, it is required that lambda=1 to not     */
/*                  require any transformation.                    */
/*          */
/* "replicate_var" Specifies the variable to be used in the        */
/*                  replicate statement of PROC NLMIXED or the      */
/*                  freq statement of PROC UNIVARIATE.  The        */
/*                  specified variable must be integer valued.     */
/*          */
/* "seq"         Specifies one or more sequence indicator          */
/*                  variables to account for effects due to the    */
/*                  sequence number of a subject's records.  This */
/*                  variable can NOT also appear in covars_prob    */
/*                  or covars_amt.                                */
/*          */
/* "dist"        * Specifies the distribution.  Use the same       */
/*                  distribution as specified in MIXTRAN.           */

```

```

/*          The possible values are:          */
/*          "HNB"          = fit Negative Binomial Hurdle      */
/*          distribution,   */
/*          "ZINB"         = fit Zero-Inflated Negative        */
/*          Binomial distribution.          */
/*          */
/* "weekend"              Specifies the weekend (Fri.-Sun.) indicator */
/*                          variable to account for a weekend effect.  A */
/*                          value of 1 represents a Fri.-Sun. record, and */
/*                          a value of 0 represents a Mon.-Thurs. record. */
/*                          This variable can NOT also appear in */
/*                          covars_prob or covars_amt.          */
/*          */
/* "vargroup"              Specifies a variable that groups observations */
/*                          to allow the model to incorporate a separate */
/*                          residual variance parameter for each of these */
/*                          groups of observations.  If the output from */
/*                          this macro is to be used in the DISTRIB macro, */
/*                          then only the weekend variable can be used. */
/*          */
/* "numvargroups"          Specifies the number of groups defined by the */
/*                          vargroup variable.  If the output from this */
/*                          macro is to be used in the DISTRIB macro and */
/*                          weekend is the "vargroup" variable, then the */
/*                          number of groups is 2.              */
/*          */
/* "start_val1"            Starting values for probability model (nocorr). */
/*                          Use only when vcontrol is called and parameter */
/*                          estimates (i.e. _parmsf1_"foodtype") from a */
/*                          previous execution of an analogous model are */
/*                          desired. */
/*                          Specifies the starting values data set for the */
/*                          1st PROC NLMIXED (i.e. NLMIXED for probability */
/*                          model). */
/*          */
/* "start_val2"            Starting values for the amount model. */
/*                          Use only when vcontrol is called and parameter */
/*                          estimates (i.e. _parmsf2_"foodtype") from a */
/*                          previous execution of an analogous model are */
/*                          desired. */
/*                          Specifies the starting values data set for the */
/*                          2nd PROC NLMIXED (i.e. NLMIXED for amount */
/*                          model). */
/*          */
/* "start_val3"            Starting values for correlated model (corr). */
/*                          Use only when vcontrol and parameter */
/*                          estimates (i.e. _parmsf3_"foodtype") from a */
/*                          previous execution of an analogous model are */
/*                          desired. */
/*                          Specifies the starting values data set for the */
/*                          3rd PROC NLMIXED (i.e. NLMIXED for correlated */
/*                          model).

```

```

/*
/* "vcontrol"      Use only when starting values from a previous
/*                  execution of the same model are also used.
/*                  Specifies a 1 to 6 character name to
/*                  differentiate output data sets for runs using
/*                  the same food. See the parameters start_val1,
/*                  start_val2, and start_val3. The default is
/*                  null.
/*
/* "nloptions"      Specifies a list of options to be added to all
/*                  calls to PROC NL MIXED, for example:
/*                  nloptions=qpoints=1 gconv=1e-12 itdetails.
/*                  If you wish to use a HNB or ZINB model,
/*                  tech=NMSIMP is required to be specified for
/*                  optimization.
/*
/* "titles"         Specifies the number of title lines (0-4) to
/*                  be reserved for the user's titles. Up to 4
/*                  title lines may be reserved for the user's
/*                  titles. The remaining title lines are used by
/*                  the macro. The default value is 0.
/*
/* "printlevel"     Specifies 1, 2, or 3 to control the amount of
/*                  information printed in the list file.
/*                  Printlevel=1 prints only the summary reports.
/*                  Printlevel=2 prints summary reports and output
/*                  from the NL MIXED procedures. Printlevel=2 is
/*                  the default value. Printlevel=3 prints
/*                  summary reports and output from all of the
/*                  statistical procedures.
/*
/* "subgroup"       Specifies one categorical variable used for
/*                  the calculation of a separate usual intake
/*                  distribution for each subgroup. This variable
/*                  can be created from a combination of other
/*                  variables (e.g. age and sex) but all variables
/*                  used to define the subgroup variable must also
/*                  be among the covariates in the model. The
/*                  subgroup variable is used in the DISTRIB
/*                  macro; however it can optionally be included
/*                  in the call to the MIXTRAN macro to achieve
/*                  backward compatibility with version 1.1
/*                  Calling a subgroup variable in MIXTRAN does not
/*                  users to only the named subgroup in DISTRIB.
/*                  A different subgroup variable can be called in
/*                  DISTRIB but see documentation for DISTRIB on how
/*                  to do this.
/*                  The subgroup parameter can now be called in
/*                  DISTRIB.
/*
/* Note:  * Parameters marked with an asterisk are mandatory, so a
/*          value must be supplied in the macro call.

```

```

/*                                                    */
/* Caution:  variable name "YN" is reserved for this macro.  */
/*                                                    */
/* Caution:  data set names "data" and "data0" and "misc_info" are  */
/* reserved for this macro.  */
/*                                                    */
/*****
;

**** Global macro variables are declared. ****
%global foodtype vcontrol ;

%macro MIXTRAN
(data=, response=, foodtype=, subject=, repeat=, covars_prob=,
covars_amt=,
outlib=, modeltype=, dist = , lambda=, replicate_var=, seq=, weekend=,
vargroup=,
numvargroups=, start_val1=, start_val2=, start_val3=, vcontrol=,
nloptions=, titles=, printlevel=, subgroup=);

*****;

%let var_ulstart=20;

*****;
**      setup      ;
*****;
%let success = 0 ;          /* successful execution flag */
%let Convflag = 0 ;        /* flags convergence errors from Proc NLMIXED */
*/

** IMS Specific ;
%let failed = 0 ;
** END OF IMS SPECIFIC;

%let modeltype=%upcase(&modeltype);

%let numvgminus1 = %eval(&numvargroups - 1);
%if &numvargroups = %str() %then %let numvargroups=%str(0);

%put ## in the MIXTRAN macro: ;
%put ## the response variable is &response ;
%put ## the data set being analysed is &data ;

%if %length(&vcontrol) >6 %then %do;          /* capture version control */
    %let vcontrol=%substr(&vcontrol,1,6);
    %put ## vcontrol reduced to 6 characters ;
%end;

```

```

%if &vcontrol ne %str() %then %do;                                /* version control in effect
*/
  %put ## vcontrol is &vcontrol ;
  %put ## start_val1:&start_val1;
  %put ## start_val2:&start_val2;
  %put ## start_val3:&start_val3;

  %if (&modeltype=%str(AMOUNT) and &start_val2=%str())
    or (&modeltype=%str(NOCORR) and (&start_val1=%str() or
&start_val2=%str()))
    or (&modeltype=%str(CORR) and &start_val3=%str())
    %then %do ;
    %put ## WARNING: if the parameter vcontrol is not blank, the user
must provide starting values;
    %put          in the parameters start_val1-start_val3 as appropriate ;
    %put          MIXTRAN will not execute properly;
  %end;                                                         /* of missing start values */
%end ;                                                         /* of version control in
effect */

%else %if &vcontrol = %str() %then %do ;                          /*version control not in
effect */
  %if (&start_val1 ne %str() or &start_val2 ne %str() or &start_val3 ne
%str())
    %then %do ;
    %put ## WARNING: if the parameter vcontrol is null the user can NOT
provide starting values;
    %put          in any of the parameters start_val1-start_val3;
    %put          MIXTRAN will not execute properly;
  %end;                                                         /* of unwanted start values
*/
%end;                                                         /*of version control not in
effect */
                                                         /*of version control not in
effect */

%if &start_val1 =%str() %then %let start_val1=start1vargrp;
%if &start_val2 =%str() %then %let start_val2=start2vargrp;
%if &start_val3=%str() %then %let start_val3=startm;

/* set up code for printing output */
%let print_on = %str() ;
%let print_off = %str(ods exclude all ;) ;
%if &printlevel=%str(2) %then %let print_on = %str(ods select all ;) ;
%else %if &printlevel=%str(3) %then %let print_off = %str() ;

/*---turn off printing if printlevel ne 3---*/
&print_off ; ** ods exclude all;

/*---read in macro variables---*/

/* the replicate variable */

```

```

%if &replicate_var = %str() %then %do ;          /* if no replicate variable
*/
  %let Freqing = %str(**unreplicated);
  %let replicate = %str(**unreplicated); ;
  %let replicate_var = %str(dummywt) ;          /* assign dummy weight if
user */
                                                /* did not supply a
replicate */
                                                /* variable. A value of 1
will*/
                                                /* be supplied */
%end; /* no replicate variable */
%else %do ;                                     /* replicate variable
supplied */
  %let Freqing = FREQ &replicate_var ;          /* for the univariate */
  %let Replicate = replicate &replicate_var; /* for the nlmixed */
  %put ## &replicate;
%end; /* of replicate variable supplied */

/* If no Title lines reserved set titles to 0 */
%if &titles = %str() %then %let titles=0;
%else %if %eval(&titles) gt 4 %then %do ; /* too many titles reserved */
  %let titles = 4;
  %put ## Number of title lines reserved for user changed to the maximum
of 4 ;
%end ; /* of too many title lines reserved */
%put ## number of titles reserved is &titles ;

/* Note whether or not correlations will be run */

%if &modeltype=%str() %then %let modeltype=%str(CORR);
%if &modeltype eq %str(NOCORR) %then %put ## Code for uncorrelated model
will be executed;
%else %if &modeltype eq %str(AMOUNT) %then %put ## Code for amount-only
model will be executed;
%else %if &modeltype eq %str(CORR) %then %put ## Code for correlated model
will be executed;
%else %do ;
  %put ## Error -- invalid modeltype -- this execution of MIXTRAN will
STOP -- ;
  %goto convexit;
%end;

/* Note what kind of distribution will be run */

%if &dist=%str() %then %let dist=%str(HNB);
%if &dist eq %str(ZINB) %then %put ## Code for Zero-Inflated Negative
Binomial distribution will be executed;
%else %if &dist eq %str(HNB) %then %put ## Code for Negative Binomial
Hurdle distribution will be executed;
%else %do ;

```



```

    %put ## Error -- invalid dist -- this execution of MIXTRAN will STOP --
;
    %goto convexit;
%end;

/* If user supplies lambda value set the bounds statement to */
/* null in the nlmixed for amount model. Otherwise set bounds */
/* for amtlambda. */
if &lambda eq %str() %then %let
    lambdabounds = %str(bounds A_LAMBDA>0.01;) ;
else %let lambdabounds = %str() ;
%put ## lambdabounds is: &lambdabounds ;

/* Note if subgroup requested */
if &subgroup ne %str() %then %put ## the subgroup is: &subgroup ;

/* list the options for nlmixed if any */
if &nloptions ne %str() %then %put ## options for NLMIXED are:
&nloptions;

/* the covariates */
***** Probability covars *****;
if &modeltype ne AMOUNT %then %do; /* models with prob portion*/
    %let vars_prob=%upcase(%quote(&covars_prob &weekend &seq));
    %put ## The Probability Covariates Are: ;
    %let I=1;
    %do %until(%qscan(&vars_prob,&I,%str( ))=%str());
        %let varb&I=%qscan(&vars_prob,&I,%str( ));
        %put ## &&varb&i;
        %let I=%eval(&I+1);
    %end; /* do %until(%qscan(&vars_prob... */
    %let cnt_prob=%eval(&I-1);
    %if &vars_prob=%str() %then %let cnt_prob=0;
%end; /* models with prob portion */

***** Amount covars *****;
%let vars_amt=%upcase(%quote(&covars_amt &weekend &seq));
%put ## The Amount Covariates Are: ;
%let I=1;
%do %until(%qscan(&vars_amt,&I,%str( ))=%str());
    %let varl&I=%qscan(&vars_amt,&I,%str( ));
    %put ## &&varl&i;
    %let I=%eval(&I+1);
%end; /* do %until(%qscan(&vars_amt... */

%let cnt_amt=%eval(&I-1);
%if &vars_amt=%str() %then %let cnt_amt=0;

%let weekend=%upcase(&weekend);
%let seq=%upcase(&seq) ;

```

```

%if &weekend ne %STR() %then %let predxb = %str(x1b1_0 x2b2_0 x1b1_1
x2b2 1 ) ;
%else %let predxb = %str(x1b1 x2b2 );

*****
/*---make datasets---*/

data data (drop=wtflag) ;
  set &data;

  if &response=0 then YN=0;
  else if &response>0 then YN=1;

  *** check that the replicate variable, if supplied, is an integer;
  *** If none supplied, then assign the value of 1 to the dummy weight
variable;
  wtflag = '0';
  %let wtch = 0 ;
  %If &replicate_var ne %str(dummywt) %then %do;
    If &replicate_var ne int(&replicate_var) then do ;
      put;
      put '*****';
      put "*** ERROR ***" ;
      put "*** The replicate variable &replicate_var is not an integer.";
      put "*** Processing of the macro MIXTRAN will be stopped.";
      put '*****';
      put;
      wtflag= '1';
      call symput('wtch',wtflag);
      stop;
    end;
  %end ;
  %else %do ;
    dummywt=1;
  %end;
run;

  %if &wtch = 1 %then %goto convexit;

/* Sort data by subject, so NLMIXED detects repeated records for the same
subject.
For clarity, the data will be sorted by both subject and sequence
number.*/
proc sort data=data;
  by &subject &repeat;
run;

/*---changed to response=. in order to get predictions for everyone---*/
data data0;
  set data;
  by &subject &repeat;
  if &response=0 then do; &response=.; end;

```

```

run;

/* for amount models change 0 intake to half the smallest amount actually
eaten */
/*%if &modeltype = %str(AMOUNT) %then %do;
proc sql;
    UPDATE data0
        Set &response=(select min(b.&response)*.5 as minFOOD from data as b
where 0<&response)
        Where &response=. ;
quit;

%end;    */ /* of changing 0 intake to half the minimum eaten for amount
models */

/* check the number of repeat observations, and the number with postive
values of the response variable */
/*Issue an error message if fewer than 2 subjects have more than 1
record (lack repeat observations) */
%let tempda = %str() ;

%if &modeltype = %str(AMOUNT) %then %do;

%let tempda = repeat ;
proc freq data=data0 noprint ;
    where &response gt 0 ;
    tables &subject / noprint out=&tempda;
run;

proc sql noprint;                                /* this works because amount
response never = 0 and only 1 or 2 recalls are allowed */
    select distinct count(count)
        into :RecallCount
        from &tempda
        where count > 1
        ;
quit;
%put ## the number of repeat observations = &RecallCount.;

%if %eval(&RecallCount) <2 %then %do ;
    %put ## Error: There are fewer than two subjects with repeat
observations. This execution of MIXTRAN will STOP ;
    %goto convexit ;
%end ;    /* recalls <2 for amount */
%else %if %eval(&RecallCount) <11 %then %do;
    #put ## Warning: There are only &RecallCount subjects with repeat
observations. Estimates might be unstable ;
%end;    /* recalls 2-10 for amount */
%end ;    /* recalls generally for amount */
run;

```

```

/* For episodically consumed foods, provide the number of observed 24-
hour recalls
/* by the number of their 24-hour recalls greater than 0.
/* The count is reported unweighted, the percentage is reported weighted
if a replicate variable is used.
/* There must be at least 2 positive recalls for episodic foods */

%if &modeltype ne %str(AMOUNT) %then %do ;

    data _mxt_recalls (keep=NumberRecalls NumberRecallsGT0 &repeat
&response &replicate var);
        set data0 (keep=&subject &repeat &response &replicate_var
&subgroup);
        by &subject &repeat
        ;
        label
            NumberRecalls          = "Number of Recalls"
            NumberRecallsGT0 = "Number of Recalls with response greater than
0 (i.e. eaten)"
        ;
        if first.&subject then do;
            NumberRecalls = 0;
            NumberRecallsGT0 = 0;
        end;
        NumberRecalls+1 ;
        if &response>0 then NumberRecallsGT0+1;
        if last.&subject then output;
run;

proc freq data= mxt recalls noprint;
    tables NumberRecalls*NumberRecallsGT0/
out=_mxt_Recalls_count(drop=percent) ;
    title%eval(&titles+1) "Number of Observed 24 Hour Recalls by
Number of 24 Hour Recalls Greater Than 0";
run;
proc freq data= mxt recalls noprint;
    tables NumberRecalls*NumberRecallsGT0/
out=_mxt_Recalls_percent(drop=count) ;
    weight &replicate var ;
    title%eval(&titles+1) "Weighted percents of Observed 24 Hour Recalls
by Number of 24 Hour Recalls Greater Than 0";
run;

/* combine the unweighted counts and the weighted percents*/
proc sort data= mxt recalls count ;
    by NumberRecalls NumberRecallsGT0 ;
proc sort data=_mxt_recalls_percent ;
    by NumberRecalls NumberRecallsGT0 ;
run;
data _mxt_recalls ;
    merge _mxt_recalls_count

```

```

                                _mxt_recalls_percent;
by NumberRecalls NumberRecallsGT0 ;

ods select all ;    /* temporarily turn the printing back on */
proc print data=_mxt_recalls noobs label ;
  ID NumberRecalls ;
  title%eval(&titles+1) "Subjects Grouped Into Number of Observed
24-Hour Recalls by The Number of Recalls With Positive Value";
  title%eval(&titles+2) "Percents Are Weighted If A Weight Is In
Use. Counts Are Not Weighted" ;
run;

  ** There should be at least one subject with two or more positive
recalls to continue. **
  ** If between 2 and 10 warn the user that estimates might be
unstable          ** ;
  proc sql noprint;
    select sum(count)
    into :Positive
    from   mxt recalls
    where  NumberRecalls>1 and NumberRecallsGT0>1 ;
  quit;
run;
/*

  %if %eval(&positive) <11 %then %do;
    %if %eval(&positive) <2 %then %do;
      %put ## Error: There are &positive subjects with
positive repeat observations. MIXTRAN will STOP ;
      %put ## Error: There must be at least one subject with
two or more positive recalls to continue. ;
      %put ## Error: This execution of MIXTRAN will STOP ;
      %goto convexit ;
    %end;          /* <2 positive recalls */
/*
    %else %do ;
      %put ## Warning: the number of subjects with two or more
positive recalls is small (&positive). Estimates might be unstable.;
      %put ## Warning: There should be at least 10 subjects with
two or more positive recalls. ;
      %end;          /* 2-10 positive recalls*/
/*
    %end ;          /* positive recalls */
/*
    %put The number of subjects with two or more positive responses is
&positive ;
  */
  Proc Datasets nolist ;
    delete
    _mxt_recalls _mxt_recalls_count _mxt_recalls_percent ;
run;

```

```

        &print_off ;    /* reset the printing level */

    %end ;              /* of checking for recalls for episodic foods */
/* %end ;              /* of version control not in effect to check recalls
( i.e. base run) */

*****;
** Find the smallest non-zero value of the response var
** over all subjects (min_amt).
** also find
** the model type, the name of the replicate (weight) variable
** and if applicable the number of var groups.
** These will be passed to the DISTRIB macro;
*****;
/* minimum amount of intake */
proc univariate data = data0 noprint;
    var &response ;
    output out=misc_info min=min_amt ;
run;

/* other data to pass to DISTRIB macro */
data misc_info ;
    set misc_info(keep=min_amt);
    FreqName=symget('replicate var'); /* the name of the weight variable */
    numvargrps=&numvargroups;          /* number of var groups */
    %if &weekend eq %str() %then %do ; /* the weekend flag */
        weekendflag=0;
    %end;
    %else %do ;
        weekendflag=1;
    %end;

/* reduce the data to one record per person */
proc sort data=data out=_persons nodupkey;
    by &subject;
    data persons ;
    set _persons (keep=&subject &replicate_var &response &subgroup ) ;
    by &subject ;
    indwts=&replicate_var;

***** end of general set up
*****;

*****
**;
/*      start of code for re-runs using only correlated nlmixed
*/;

**RERUNS ;

```

```

%if &vcontrol ne %str() %then %do ;

    ** reference the correct variable parameter for the correlated nlmixed
    **;
    %let amtlambda = %str(A_LAMBDA) ;
    %let vu1 = %str(P_VAR_U1) ;
    %let vu2 = %str(A_VAR_U2) ;
    %put this is a &modeltype rerun (&vcontrol);
    %put the starting value data set for the probability model is
&start_val1 ;
    %put the starting value data set for the amount model is &start_val2 ;
    %put the amtlambda is &amtlambda ;
    %put the nb_k is &nb_k ;
    %put the vu1 is &vu1 ;
    %put the vu2 is &vu2 ;

    ** read in the strings for etal etc. Used in the correlated nlmixed
    **;
    %if &modeltype ne AMOUNT %then %do;
        data etas ;
            set &outlib..etas_&foodtype ;
            call symput('eta1',eta 1);
            call symput('eta2',eta 2);
            call symput('nonu1seq1',shorteta1);
            call symput('nonu2seq2',shorteta2);
        run;
        %put eta1 is &eta1;
        %put eta2 is &eta2;
        run;
    %end;
    %else %do ;
        data etas ;
            set &outlib..etas_&foodtype ;
            call symput('eta2',eta 2);
            call symput('nonu2seq2',shorteta2);
        run;
        %put eta2 is &eta2;
        run;
    %end;

%end;    /* of code for all reruns */

*****
*****;
/*    If this a re-run and the model is correlated, then skip the initial
*/
/*    probability and amount parts of the macro.
*/
/*    If this is a base run, then proceed with calculation of starting
values */
/*    and setting up names.
*/

```

```

/*      If this a re-run and the model is amount or nocorr, then skip the
*/
/*      starting values and setting up names and go to the nlmixed
procedures      */

%if (&vcontrol = %str())
or (&vcontrol ne %str() and &modeltype ne %str(CORR))
%then %do ;

    %if &modeltype ne AMOUNT %then %do; /* models with prob portion*/

        %if &vcontrol = %str() %then %do ; /* base runs with corr or nocorr
*/

            /*---find starting estimates for probability model---*/

            ods output ParameterEstimates=parmsg1(rename=Parameter=Name)
            modelfit=modelfitb;

            title%eval(&titles+1) "Starting Estimates for Probability Model";

            proc genmod data=data descending namelen=30;
                model yn=&vars_prob/dist=binomial;
                &freqing;
                run;

            data random1;
                Format Parameter $30.;
                Parameter= Compress('P' || '_VAR_U1');
                call symput('vu1',parameter);
                Name='Var(Rndm Effect)';
                Estimate=&var_ulstart;
                run;
                %put VU1 = &vu1 ;

            data newnames1;
                format var name $70. Parameter $30.;
                %let crd1 = P01_INTERCEPT ;

                %If &cnt_prob > 0 %then %do;
                    %let J = 1 ;
                    %do %until (&j=(&cnt_prob+1));
                        %if %eval(&j) lt 9 %then %let znum = 0;
                        %else %let znum=%str() ;
                        %let crd1 = &&crd1 P&znum.%eval(&j+1)_&&varb&j ;
                        %let j=%eval(&j+1);
                    %end;
                %end; /* cnt_prob>0 */

            set parmsg1;

            if Name='Scale' then delete;

```



```

%if &cnt_prob>0 %then %do;

    %let int=%scan(&crd1,1,%str( ));
    if Name='Intercept' then Parameter="&int";

    %let j=1;
    %do %until(&j=(&cnt_prob+1));
        %let up=%upcase(&&varb&j);
        %let lngth=%length(&&varb&j);
        %put &up &lngth;
        %let varname=%scan(&crd1,(&j+1),%str( ));
        if UPCASE(Name)="&up" then do;
            Parameter="&varname";
            var_name="&up";
        end; /* else if UPCASE(Name)="&up"... */
        %let j=%eval(&j+1);
    %end; /* do %until(&j=(&cn... */
%end; /* if &cnt_prob>0... */

%if &cnt_prob=0 %then %do;
    var_name = ' ' ; ** initialise if no covariates **;
    %let int=%scan(&crd1,1,%str( ));
    if Name='Intercept' then Parameter="&int";
%end; /* if &cnt_prob=0... */
run;

data start1;
format Name $30. Parameter $30.;
set newnames1 random1;
if Parameter not in ('P01_INTERCEPT",&vu1") then
    list=trim(Parameter)||'*'||trim(Var Name);
else if Parameter in ('P01_INTERCEPT",&vu1") then
    list=trim(Parameter);
keep Parameter Name Estimate list;
run;

data trans1;
set start1;
if Parameter in ("&vu1") then delete;
run;

proc transpose data=trans1 out=out1;
var list;
run;

data eqn1 ;
format etal eta_1 $2000.;
set out1;

%if &cnt_prob>0 %then %do;
    %let k=1;
    etal=trim(col&k);

```

```

    %if (&k ne (&cnt_prob+1)) %then %do %until(&k=(&cnt_prob+1));
        %let k=%eval(&k+1);
        etal=trim(et1)||'+'||trim(col&k);
    %end; /* do %until(&k...*/
    call symput('nonu1',etal);
    eta_1=trim(et1)||' + u1';

    /* ** delete the sequence variables from etal and create a */
    /* ** string to calculate xlb1 for weekend runs */
    /* ** u1 will not be added to etal either. */
    %if &seq ne %str() %then %do ;
        %let seqname=%qscan(&seq,1,%str( ));
        %put seqname= &seqname;
        nseq=index(et1,"&seqname");
        %put nseq= ;
        shortetal=substr(et1,1,(nseq-6));
        call symput('nonulseq1',shortetal);
        drop nseq shortetal;
    %end; /* of seq ne () */
    %else %do;
        call symput('nonulseq1',etal);
    %end;
%end; /* %if &cnt_prob>0...*/

%if &cnt_prob=0 %then %do;
    call symput('nonulseq1',col1);
    eta_1=trim(col1)||' + u1';
%end; /* %if &cnt_prob=0...*/
call symput('etal',eta_1);
run;

%put "response is          " &response;
%put "etal is              " &etal;

data start1vargrp;
    length parameter $30.;
    set start1;

    if parameter = "&vu1" then do;
        parameter = 'P_LOGSDU1';
        name = 'Reparam Var(u1)';
        estimate = log(sqrt(estimate));
        list = 'P_LOGSDU1';
        output;
    end;
    else output;
run;
%end ;          /* of vcontrol = '' (base) for corr or nocorr */

/* want all runs with nocorr to do the nlmixed, base or re-run */
/* and the base corr runs ; */

```

```

*****;
** Run nlmixed for probability model;
*****;

/*---turn the printing back on for level gt 1 ---*/

&print_on ;      ** ods select all;

ods output ParameterEstimates=parmsf1 AdditionalEstimates=adprmsf1
FitStatistics=fitf1
ConvergenceStatus=convf1;

title%eval(&titles+1) 'Probability Model';

proc nlmixed data=data &nloptions;
  parms /data=&start vall;
  &replicate ;                      /* if replicate variable provided
*/

  &vu1 = EXP(2*P_LOGSDU1);

  x1b1u1=&etal;
  p=exp(x1b1u1)/(1+exp(x1b1u1));
  x1b1=x1b1u1-u1;
  model YN ~ binomial(1,p);
  random u1 ~ normal(0,&vu1) subject=&subject out=predulu;
  estimate "P_VAR_U1" EXP(2*P_LOGSDU1);
  predict p out=predpu;
  predict x1b1 out=predx1b1u;
  run;

data null ;
  set convf1;
  call symput('convbi',Reason);
  run;

/* check for convergence error. End processing if found */
Data _null_ ;
  %let ccsbi=%index(&convbi,convergence criterion satisfied) ;
  %if &ccsbi eq %str(0) %then %do ;

    %put *****;
    %put ### Convergence Problem from PROC NLMIXED ;
    %put ### Macro MIXTRAN Stopped Due to Convergence Error in
Probability Model;
    %put ### Response Variable is &response ;
    %put ### data is &data;
    %put ### weight variable is &replicate_var;
    %put ** ERROR** MSG: &convbi ;
    %put *****;

```

```

        %let convflag = 1 ;

        ** IMS SPECIFIC: set flags to note which strata has failed.
        ** these will be used in selecting data for Distrib in the calling
program;
        %let fail_count = %eval(&fail_count+1) ;
        %let failed = 1;

        data fail&fail_count ;
            fcount=&fail_count;
            food="&eaten";
            subpop="&data";
            run_num="&replicate_var";
            reason="&convbi";
            converr=&convflag;
            output;
        ** end of IMS specific code;

        Proc Datasets nolist ;
            delete
                CONV1 DATA DATA0 EQN1 FIT1 MODELFITB NEWNAMES1 OUT1
PARMSF1
                PARMSG1 PREDP1 PREDU1U PREDX1B1U RANDOM1 START1 TRANS1
            ;
        %GOTO convexit;
    %end; /* of convergence error check in convbi */
run ;

proc transpose data=adprmsf1 out=adprmsf1(drop=_name_);
    id label;
    var estimate;
run;

%end; /* models with prob portion */

%if &vcontrol = %str() %then %do; /* base run for all models */

        *****;
        ** starting estimates for amount ;
        *****;

        /*---turn off printing if printlevel lt 3---*/
        &print_off ; ** ods exclude all ;

        /*---find starting estimates for amount model---*/

        title&eval(&titles+1) "Starting Estimates for Amount Model" ;

        /* if lambda not supplied by user the macro assigns a value */
        %if &lambda eq %str() %then %do ;

```

```

%if &cnt_amt>0 %then %do;

    proc transreg data=data0 details pbo;
        ods output boxcox=boxcox;
        where &response>0;
        model BoxCox(&response / lambda=0.05 to 1 by 0.05) =
identity(&vars_amt);
        run;

    data _null_;
        set boxcox;
        if ci='<';
        call symput ('lambda',lambda);
        run;

%end; /* %if cnt_amt>0 ... */
%else %let lambda=0.4;
%end ; /* of macro assigning lambda if none supplied by user */

data data0;
    set data0;
    %if &lambda=%str(0) %then %do;
        boxcoxy=log(&response) ;
    %end; /* deals with a lambda of 0 */
    %else %do;
        boxcoxy=(&response**&lambda-0)/&lambda;
    %end;
    run;
option nomprint;
*****genmod for starting values *****;
ods output ParameterEstimates=parmsg2(rename=Parameter=Name);

proc genmod data=data0 namelen=30;
    model boxcoxy =&vars_amt/dist=nb;
    &freqing ;
    run;

***** Calculate variance values
*****;

/*Isolate response variable*/
data something (keep = &response);
    set data;
run;

data something (rename = (&response = y));
    set something;
    if &response ne .;
run;

```

```

/* Transpose Data. */
proc transpose data=parmsg2 out=other;
    var estimate;
    id Name;
run;

/* get linear estimate for each subject based on model parameter estimates
from genmod starting values*/
data other_again (drop = _NAME_ Intercept Dispersion) ;
    set other;
run;

proc iml;
    start linear;

        varNames = {"y"}; /* these vars have the same type */
        use something; /* open the data set */
        read all var varNames into Y; /* read y var into matrix
*/
        close something; /* close the data set */

        use other again;
        read all var _NUM_ into X;
        close other_again;

        yhat= Y*X;
        lin = yhat[,+];

        create MyData from lin [colname=varNames];/*output linear estimates
to MyData dataset*/
        append from lin;
        close MyData;

/*Print lin;*/

finish linear;
run linear;
quit;

/* Transform 0's in MyData to missing*/
data MyData2;
    set MyData;
    if y = 0 then y = .;
    rename y = vars;
run;

proc iml;
    start linear2;

        varNames = {"vars"}; /* these vars have the same type */
        use MyData2; /* open the data set */

```

```

        read all var varNames into Y;          /* read y var into matrix
*/
        close MyData2; /* close the data set */

        varNames = {"intercept"};
        use other;
        read all var varNames into int;
        close other;

        varNames = {"Dispersion"};
        use other;
        read all var varNames into disp;
        close other;

        yhat= int+Y;
        EXPyhat = exp(yhat);
        mult = EXPyhat*disp;
        nbp = 1/(1+mult);

        create MyData3 from nbp [colname=varNames];
        append from nbp;
        close MyData3;

finish linear2;
run linear2;
quit;

data MyData4;
    set MyData3;
    rename Dispersion = nbp;
run;

proc iml;
    start variance;

        varNames = {"nbp"}; /* these vars have the same type */
        use MyData4;          /* open the data set */
        read all var varNames into nbp; /* read y var into matrix
*/
        close MyData4; /* close the data set */

        varNames = {"Dispersion"};
        use other;
        read all var varNames into disp;
        close other;

        nb_k = 1/disp;
        nbp_diag = nbp # nbp;

        nb_mean=( (1-nbp) * (nb_k) ) /nbp;

```

```

nb_var1 = ((1-nbp)*(nb_k))/nbp_diag;

nb_var = mean(nb_var1);

create MyData7 from nb_var1 [colname=varNames];
append from nb_var1;
close MyData7;

create MyData5 from nb_var [colname=varNames];
append from nb_var;
close MyData5;

finish variance;
run variance;
quit;

data MyData6 ;
    set MyData5;
    rename Dispersion = nb_var;
run;

data other2;
    merge other MyData6;
run;

/* Calculate k, p, mean and variance from intercept and dispersion
parameters. */
data other3;
    set other2;
    nb_k = 1/dispersion;
run;

/* Save k and p in macro variables. */
data null ;
    set other3;
    call symput('nb_k', nb_k);
    call symput('nb_var', nb_var);
run;
    %let nb_k = &nb_k;
    %put the nb k is &nb k;
/*%let a_nb_k=&nb_k;*/
/* update parmsg2 dataset with new variables */

proc transpose data=other3 out=other3;
run;

data other3;
    set other3;
    rename _NAME_ = Name;

```



```

run;

proc sort data = parmsg2;
    by Name;
run;

proc sort data = other3;
    by Name;
run;

data parmsg2;
    update parmsg2 other3;
    by Name;
run;

***** End Calculations for
variance values *****;

data random2;
    Format Parameter $30.;
    Parameter= Compress('A' || '_VAR_U2');
    call symput('vu2',parameter);
    Name='Var(Rndm Effect)';
    Estimate=1;
run;
%put VU2= &vu2;

data newnames2;
    format var_name $70. parameter $30.;
    %let crd2 =A01_INTERCEPT ;
    %If &cnt_amt > 0 %then %do;
        %let J = 1 ;
        %do %until (&j=(&cnt_amt+1));
            %if %eval(&j) lt 9 %then %let znum = 0;
            %else %let znum=%str() ;
            %let crd2 = &&crd2 A&znum.%eval(&j+1)_&&var1&j ;
            %let j=%eval(&j+1);
        %end;
        %put crd2 is &crd2;
    %end; /* cnt_amt>0 */
    set parmsg2;
    %if &cnt_amt>0 %then %do;
        %let int=%qscan(&crd2,1,%str( ));
        if Name='Intercept' then Parameter="&int";
        if Name = 'nb_k' then do;
            Parameter = 'A NB K';
            Estimate = Estimate;
            Name = 'Inverse Dispersion';
        end;
        else if Name = 'nb_var' then do;
            Parameter='A_VAR_E';
            Estimate= Estimate;
    %end;

```

```

        Name='Residual';
    end; /* else if Name=... */ /* of name = 'dispersion' */
%let j=1;
%do %until(&j=(&cnt_amt+1));
    %let up=%qupcase(&&varl&j);
    %let lngth=%length(&&varl&j);
    %let varname=%qscan(&crd2,(&j+1),%str( ));
    if UPCASE(Name)="%&up" then do;
        Parameter="%&varname";
        var_name="%&up";
    end; /* else if UPCASE ... */
    %let j=%eval(&j+1);
%end; /* do %until(&j... */
%end; /* %if &cnt_amt>0... */
%if &cnt_amt=0 %then %do;
    var_name = ' ' ; ** initialise if no covariates **;
    %let int=%qscan(&crd2,1,%str( ));
    if Name='Intercept' then Parameter="%&int";
        if Name = 'nb_k' then do;
            Parameter = 'A_NB_K';
            Estimate = Estimate;
            Name = 'Inverse Dispersion';
        end;
    else if Name = 'nb_var' then do;
        Parameter='A VAR E';
        Estimate = Estimate;
        Name='Residual';
    end; /* else if Name=... */
%end; /* %if &cnt_amt=0... */
run;

data lambda;
    Format Parameter $30.;
    Parameter= Compress('A' || ' LAMBDA');
    call symput('AMTLAMBDA',parameter);
    Estimate=&lambda;
    Name='lambda';
run;
%put amtlambda= &amtlambda ;

data start2;
    format Name $30. Parameter $30.;
    set newnames2 random2 lambda(in=inlam);
    /* delete the lambda record if user supplied lambda value */
    %if &lambdabounds eq %str() %then %do ;
        if inlam then delete;
    %end;
    if Parameter not in ('A01_INTERCEPT',"&vu2",'A_NB_K','') then list=
        trim(Parameter)||'*'||trim(Var_Name);
    else if Parameter in ('A01_INTERCEPT',"&vu2",'A_NB_K','') then
list=trim(Parameter);

```

```

keep Parameter Name Estimate list;
run;

data trans2;
  set start2;
  if Parameter in ("&vu2", 'A_VAR_E', '', 'A_NB_K') then delete;
run;

proc transpose data=trans2 out=out2;
  var list;
run;

data eqn2 ;
  format eta2 eta_2 $2000.;
  %if &seq ne %str() %then %do ;
    format shorteta2 $2000.;
  %end;
  set out2;
  %if &cnt_amt>0 %then %do;
    %let k=1;
    eta2=trim(col&k);
    %if (&k ne (&cnt_amt+1)) %then %do %until (&k=(&cnt_amt+1));
      %let k=%eval(&k+1);
      eta2=trim(eta2)||'+'||trim(col&k);
    %end; /* %if (&k ne... */
    call symput('nonu2',eta2);
    eta_2=trim(eta2)||' + u2';
    /* ** delete the sequence variables from eta2 and create a */
    /* ** string to calculate x2b2 for weekend runs */
    /* ** u2 will not be added to eta2 either. */
    %if &seq ne %str() %then %do ;
      %let seqname=%qscan(&seq,1,%str( ));
      %put seqname= &seqname;
      nseq2=index(eta2,"&seqname");
      put nseq2= ;
      shorteta2=substr(eta2,1,(nseq2-6));
      call symput('nonu2seq2',shorteta2);
      drop nseq2 shorteta2;
    %end; /* of seq ne () */
    %else %do;
      call symput('nonu2seq2',eta2);
    %end;
  %end; /* %if cnt_amt>0 ... */
  %if &cnt_amt=0 %then %do;
    call symput('nonu2seq2',col1);
    eta_2=trim(col1)||' + u2';
  %end; /* %if &cnt_amt=0... */

  call symput('eta2',eta_2);
run;

%put "eta2 is " &eta2;

```

```

data start2vargrp;
  length parameter $30.;
  set start2;

  if Parameter in (') then delete;

  %if &vargroup ne %STR() %then %do;
    if parameter='A_VAR_E' then do;
      list=' ';
      parameter = 'A LOGSDE';
      name = 'Resid, Reparam Var Grp1';
      estimate = log(sqrt(estimate));
      output;
      do vg = 2 to &numvargroups;
        parameter = 'A_DELTAVG' || left(put(vg,2.));
        name = 'Resid, Delta for Var Grp' || left(put(vg,2.));
        estimate=0;
        output;
      end;
    end;
  %end; /* vargroup present */
  %else %do;
    if parameter='A_VAR_E' then do;
      list=' ';
      parameter = 'A_LOGSDE';
      name = 'Resid, Reparam ';
      estimate = log(sqrt(estimate));
      output;
    end;
  %end;
  else if parameter = "&vu2" then do;
    parameter = 'A_LOGSDU2';
    name = 'Reparam Var(u2)';
    estimate = log(sqrt(estimate));
    list = 'A_LOGSDU2';
    output;
  end;
  else output;

  %end; /* of using this code if a base run- vcontrol = ' '
  */
  /* all base runs or re-runs with amount or nocorr use this
  nlmixed */

  ****
  ** nlmixed for amount
  ****
  /*---turn the printing back on for level GT 1 ---*/

  &print_on ;    ** ods select all;

```

```

ods output ParameterEstimates=parmsf2 AdditionalEstimates=adprmsf2
FitStatistics=fitf2
ConvergenceStatus=convf2;

title%eval(&titles+1) 'Amount Model';

proc nlmixed data=data0 &nloptions;
  parms / data=&start val2;
  &replicate ;                               /* if replicate variable provided */
*/
  &lambdabounds ;

  %if &vargroup ne %STR() %then %do;
    ARRAY VARGRP[&NUMVGMINUS1] A_VARGRP2-A_VARGRP&NUMVARGROUPS;
    ARRAY A_DELTAVG[&numvgminus1] A_DELTAVG2-A_DELTAVG&numvargroups;

    if &vargroup=1 then do;
      A_VARGRP1 = EXP(2*A_LOGSDE);
      A_VAR_E = A_VARGRP1;
    end;
    else do vg = 2 to &numvargroups;
      if &vargroup=vg then do;
        indx=vg-1;
        vargrp[indx] = exp(2*(A_LOGSDE + A_DELTAVG[indx]));
        A_VAR_E = VARGRP[INDX];
      end;
    end;
  %end;
  %else %do;
    A_VAR_E = exp(2*A_LOGSDE);
  %end;
  &vu2 = exp(2*A_LOGSDU2);

  x2b2u2=&eta2;
  lambda_nb = exp(x2b2u2);
  p=exp(1)/(1+exp(-x2b2u2));
  alpha = 1/A_NB_K;

  %if (&lambdabounds eq %str()) %then %let amtlambda = &lambda; /*
user supplied lambda */
  %if &lambda ne %str(0) %then %do;
    boxcoxy=(&response**&amtlambda-0)/&amtlambda;
  %end;                               /* lambda ne 0 */
  %else %do;
    boxcoxy=log(&response) ;
  %end;                               /* lambda=0 */

  pdf=(gamma(&response+alpha)/(gamma(&response+1)*gamma(alpha)))

  *((1/(1+A_NB_K*lambda_nb))**(alpha)*(A_NB_K*lambda_nb/(1+A_NB_K*lambda_nb))**&response);
  ll=log(pdf);

```

```

    model &response ~ general(11);
    random u2 ~ normal(0,&vu2) subject=&subject out=predu2u;
    estimate "A_VAR_U2" exp(2*A_LOGSDU2); /* common approach is to
reparameterize the variance estimates or use a Choleskyroot
reparameterization*/
    %if &vargroup ne %STR() %then %do;
        estimate "A_VARGRP1" exp(2*A_LOGSDE);
        %let kvg = 2;
        %do kvg = 2 %to &numvargroups;
            estimate "A_VARGRP&kvg" exp(2*(A_LOGSDE + A_DELTAVG&kvg));
        %end;
    %end;
    %else %do;
        estimate "A_VAR_E" exp(2*A_LOGSDE);
    %end;
    e_int=x2b2u2;
    x2b2=x2b2u2-u2;
    predict e_int out=predeintu;
    predict x2b2 out=predx2b2u;
    run;

*****;
data null ;
    set convf2;
    call symput('convin',Reason);
    run;

/* check for convergence error. End processing if found */
Data null ;
    %let ccsin=%index(&convin,convergence criterion satisfied) ;
    %if &ccsin eq %str(0) %then %do ;

        %put *****;
        %put ##! Convergence Problem from PROC NLMIXED ;
        %put ##! Macro MIXTRAN Stopped Due to Convergence Error in Amount
Model;
        %put ##! Response Variable is &response ;
        %put ##! data is &data;
        %put ##! weight variable is &replicate_var;
        %put ##! ERROR MSG: &convin ;
        %put *****;
        %let convflag = 2 ;

        ** IMS SPECIFIC: set flags to note which strata has failed.
        ** these will be used in selecting data for Distrib in the calling
program;
        %let fail_count = %eval(&fail_count+1) ;
        %let failed = 1;

        data fail&fail_count ;

```

```

        fcount=&fail_count;
        food="&eaten";
        subpop="&data";
        run_num="&replicate_var";
        reason="&convin";
        converr=&convflag;
        output;
    ** end of IMS specific code;

Proc Datasets nolist ;
    delete
        BOXCOX CONV1 CONV2 DATA DATA0 EQN1 EQN2 FIT1 FIT2 LAMBDA
        MODELFITB NEWNAMES1 NEWNAMES2 OUT1 OUT2
        PARMSF1 PARMSF2 PARMSG1 PARMSG2 PREDEINTU PREDPU PREDU1U
        PREDU2U PREDX1B1U PREDX2B2U RANDOM1 RANDOM2 START1 START2
        TRANS1 TRANS2
    ;
    %GOTO convexit;
%end; /* of convergence error check in convin */
run ;

proc transpose data=adprmsf2 out=adprmsf2(drop=_name_);
id label;
var estimate;
run;

*****;
** start of covariance ;
*****;

/*---turn off printing if print level lt 3---*/
&print_off ; **ods exclude all;

/*---compute starting value for covariance---*/
/*---and prepare data sets for summary reports---*/
%if &modeltype ne AMOUNT %then %do; /* prob part available */
    data cov;
        Format Parameter $30.;
        set parmsf1 parmsf2;
        if Parameter in ("&vu1", "&vu2", 'P_LOGSDU1', 'A_LOGSDU2');
        keep Parameter Estimate;
        run;
%end; /* prob part available*/

%else %do; /* no prob part */
    data cov;
        Format Parameter $30.;
        set parmsf2;
        if Parameter in ("&vu2", 'A_LOGSDU2');
        keep Parameter Estimate;

```

```

run;
%end; /* no prob part */

proc transpose data=cov out=outcov;
var Estimate;
id Parameter;
run;

%if &modeltype ne AMOUNT %then %do; /* prob part available */
data cov2;
set outcov;
Format Parameter $30.;
Rho=0.5;
Parameter='Z_U';
Name='Z-trans of Correlation';
Estimate=0.5*(log(1 + Rho) - log(1-Rho));
keep Parameter Estimate Name;
run;

data fit1;
format Parameter $30. Name $30.;
set fitf1;
if Descr in ('-2 Log Likelihood','AIC (smaller is better)');
if Descr='-2 Log Likelihood' then do;
Parameter='ll';
Name='-2 Log Likelihood--bi';
end; /* if Descr='-2 Log.. */
if Descr='AIC (smaller is better)' then do;
Parameter='aic';
Name='AIC--bi';
end; /* if Descr='AIC... */

proc sort data = fit1;
by Parameter;
run;

%end; /* prob part available*/

/* base run only to set up files for printing (all models) */
%if &vcontrol = %str() %then %do ;
data fit2;
format Parameter $30. Name $30.;
set fitf2;
if Descr in ('-2 Log Likelihood','AIC (smaller is better)');
if Descr='-2 Log Likelihood' then do;
Parameter='ll';
Name="-2 Log Likelihood--"||"amount";
end; /* if Descr='-2 Log.. */
if Descr='AIC (smaller is better)' then do;
Parameter='aic';
Name="AIC--"||"amount";
end; /* if Descr='AIC... */

```



```

proc sort data = fit2;
  by Parameter;
run;

  %if &modeltype ne AMOUNT and &modeltype ne NOCORR %then %do;    /*
correlated part only */
  data fitboth;
    merge fit1 (rename=Value=Valueb) fit2 (rename=Value=Valuel);
    by Parameter;
    m2llo=Valueb + Valuel;
    run;
  %end; /* correlated part only */

  %if &modeltype ne AMOUNT %then %do;    /* prob part available */
  data names;
    format Parameter $30. type $6. Name $30. ;
    set &start_val1 (in=a) &start_val2 (in=b) cov2 (in=c) fit1 (in=d)
fit2 (in=e);
    if a=1 then type='1.pr';
    else if b=1 then type='2.am';
    else if c=1 then type='3.cov';
    else if d=1 or e=1 then type='4.fit';
    If type in ('1.pr', '2.am') then Name2=trim(Name)||'--
'||substr(type,3,2);
    else Name2=Name;
    keep Parameter Name type Name2 Value;
  %end;    /* prob part available */

  %else %do; /* no prob part */
  data names;
    format Parameter $30. type $6. Name $30. ;
    set &start_val2 (in=b) fit2 (in=e);
    if b=1 then type='2.am';
    else if e=1 then type='4.fit';
    If type in ('2.am') then Name2=trim(Name)||'--'||substr(type,3,2);
    else Name2=Name;
    keep Parameter Name type Name2 Value;
  %end; /* no prob part */

proc sort data=names;
  by Parameter;
run;

data names2;
  set names;
  if type='4.fit' then delete;
run;

  %end ;    /* of vcontrol = ' ' - base run  setting up files for
printing*/

```

```

%if &modeltype ne AMOUNT %then %do;    /* prob part available */
  data start3 ;
    format Parameter $30.;
    set parmsf1 parmsf2 cov2 ;
  run;

  data misc_info;
    merge misc_info adprmsf1 adprmsf2;
    cov_u1u2=0;
  run;
%end;    /* prob part available */

%else %do; /* no prob part */
  data start3 ;
    format Parameter $30.;
    set parmsf2 ;
  run;

  data misc_info;
    merge misc_info adprmsf2;
  run;
%end; /* no prob part */

proc sort data=start3;
  by Parameter;
run;

/* starting data set for 3rd nlmixed in base runs */
%if &vcontrol = %str() %then %do ;    /* base run */
  data startm;
    merge names2 start3;
    by Parameter;
  run;

  data estprint;
    merge names start3;
    by Parameter;
  run;

  proc sort data=estprint;
    by type;
  run;
  &print on ;    ** ods select all;
%end ;    /* of vcontrol = ' ' base run */

*****
**;
  ***** Save parameter estimates and predicted values from nlmixed
****;
  ***** procs for probability model and amount model
****;

```

```

*****
**;
```

```

/* save parmsf1 and parmsf2 if this is a base run */
%if &vcontrol = %str() %then %do ; /* base run - save parmsf1 and 2 */

data &outlib..etas_&foodtype;
  set eqn2 (keep=eta_2);
  format shorteta2 $2000. ;
  shorteta2=symget('nonu2seq2');
run;

%if &modeltype ne %str(AMOUNT) %then %do ;
  data &outlib..etas_&foodtype;
    merge &outlib..etas_&foodtype eqn1 (keep=eta_1);
    format shorteta1 $2000. ;
    shorteta1=symget('nonu1seq1');
  run;
  data &outlib.._parmsf1_&foodtype ;
    set parmsf1(keep=parameter estimate) ;
  run;
%end ; /* no probability part */
data &outlib.._parmsf2_&foodtype ;
  set parmsf2(keep=parameter estimate);
run;
%end ; /* vcontrol = ' ' - base run, saving parmsf1 and parmsf2 */

/* save parameter estimates */

data param unc; /* param data set for uncorrelated part, with version
control giving date and time */
  Format Parameter $30.;
  set start3 ;
run;

proc transpose data= param unc
out=&outlib.._param_unc_&foodtype&vcontrol(drop=_name_);
  id parameter;
  var estimate;
run;

data &outlib.._param_unc_&foodtype&vcontrol;
  merge &outlib.._param_unc_&foodtype&vcontrol
  misc_info; /* adding descriptive
info */

/* if user supplied lambda, restore lambda to output data set for use
in DISTRIB */
%if &lambdabounds eq %str() %then %do ;
  a_lambda=&amtlambda;
%end;
```

```

/* save predicted data */

/* add sum of weights and number of subjects to predx2b2u */

proc sort data = predx2b2u ; by &subject ;
data predx2b2u ;
    merge predx2b2u
        _persons ;
        by &subject ;

/* _predicted data set for uncorrelated or amount */
%if &modeltype ne AMOUNT %then %do;
    %if &weekend ne %STR() %then %do;
        data &outlib._pred_unc_&foodtype&vcontrol (keep=&subject &predxb
&replicate_var &weekend
                                &subgroup );
        if (_n_=1) then set &outlib._param_unc_&foodtype&vcontrol;
        set predx2b2u(drop=&weekend);
        by &subject &repeat;
        if (first.&subject);

        /* create x1b1, x2b2 for weekend/seasonal or other temporal
records*/
        /* weekend=0 indicates time0 */
        /* weekend=1 indicates time1 */
        &weekend=0;
        x1b1_0=(&nonulseq1);
        x2b2_0=(&nonu2seq2);

        &weekend=1;
        x1b1_1=(&nonulseq1);
        x2b2_1=(&nonu2seq2);

        run;
    %end; /* of &weekend ne %str() (i.e. have a weekend variable) */

    %else %do;
        data &outlib._pred_unc_&foodtype&vcontrol (keep=&subject &predxb
&replicate_var
                                &subgroup );
        merge predx1b1u (keep=&subject &repeat pred &replicate_var
rename=(pred=x1b1))
            predx2b2u (keep=&subject &repeat pred &subgroup
rename=(pred=x2b2));
        by &subject &repeat;
        if (first.&subject);

        run;
    %end; /* &weekend is blank, i.e. no weekend variable */
%end; /* prob part available */

```

```

%else %do;
  %if &weekend ne %STR() %then %do;
    data &outlib.._pred_unc_&foodtype&vcontrol (keep=&subject x2b2_0
x2b2_1 &replicate_var
&weekend
&subgroup);
    if (_n_=1) then set &outlib.._param_unc_&foodtype&vcontrol;
    set predx2b2u(drop=&weekend);
    by &subject &repeat;
    if (first.&subject);
    /* create x2b2 for weekday, weekend records*/
    /* weekend=0 indicates a weekday, Monday-Thursday */
    /* weekend=1 indicates a weekend, Friday-Sunday */
    &weekend=0;
    x2b2_0=(&nonu2seq2);

    &weekend=1;
    x2b2_1=(&nonu2seq2);

    run;
  %end; /* &weekend is available */
%else %do;
  data &outlib.._pred_unc_&foodtype&vcontrol (keep=&subject x2b2
&replicate_var
&subgroup );
  set predx2b2u (keep=&subject &repeat &replicate_var &subgroup
pred rename=(pred=x2b2));
  by &subject &repeat;
  if (first.&subject);

  run;
%end; /* &weekend is not available */
%end; /* prob part unavailable */

%put ## the data sets &outlib.._pred_unc_&foodtype&vcontrol and
&outlib.._param_unc_&foodtype&vcontrol ;
%put ## have been output for use in the DISTRIB macro or other software
using AMOUNT or NOCORR models;

*****
*****;

&print on ; ** ods select all;
%END ; /* END OF SKIPPING FIRST PART FOR CORRELATED RE_RUNS */

*****;
/* **** run correlated part unless modeltype is 'amount' or 'nocorr'
***** */
%if &modeltype eq CORR %then %do ;

/*---turn the printing back on--- for level GT 1 */

```

```

&print_on ;  ** ods select all;

*****;
** nlmixed for correlated ;
*****;
data &start_val3;
  set &start_val3;
  if Parameter in ('numvargroups','min_amt') then delete;
run;

/*---Fit Correlated Model---*/

title%eval(&titles+1) 'Correlated Model';

ods output ParameterEstimates=parmsf3 AdditionalEstimates=adprmsf3
FitStatistics=fitf3 ConvergenceStatus=convf3;

proc nlmixed data=data &nloptions;
  parms / data=&start_val3 ;
  &replicate ; /* If replicate variable supplied */
  &lambda bounds ;

  %if &vargroup ne %STR() %then %do;
    ARRAY VARGRP[&NUMVGMINUS1] A_VARGRP2-A_VARGRP&NUMVARGROUPS;
    ARRAY A_DELTAVG[&numvgminus1] A_DELTAVG2-A_DELTAVG&numvargroups;

    if &vargroup=1 then do;
      A_VARGRP1 = EXP(2*A_LOGSDE);
      A_VAR_E = A_VARGRP1;
    end;
    else do vg = 2 to &numvargroups;
      if &vargroup=vg then do;
        indx=vg-1;
        VARGRP[INDX] = EXP(2*(A_LOGSDE + A_DELTAVG[INDX]));
        A_VAR_E = VARGRP[INDX];
      end;
    end;
  %end;
  %else %do;
    A_VAR_E = EXP(2*A_LOGSDE);
  %end;

  &vu1= EXP(2*P_LOGSDU1);
  &vu2= EXP(2*A_LOGSDU2);
  RHO=(EXP(2*Z_U) - 1) / (EXP(2*Z_U) + 1);
  COV_U1U2 = RHO*EXP(P_LOGSDU1+A_LOGSDU2);

  %if &dist = %str(HNB) %then %do;

  x1b1u1=&etal;
  p=exp(x1b1u1)/(1+exp(x1b1u1));

```

```

ll_b = 1-p;

x2b2u2=&eta2;
lambda_nb = exp(x2b2u2);
e int=x2b2u2;
alpha=1/A_NB_K;

pdf=(gamma(&response+alpha)/(gamma(&response+1)*gamma(alpha)))*((1/(
1+A_NB_K*lambda_nb))**(alpha)*(A_NB_K*lambda_nb/(1+A_NB_K*lambda_nb))**
&response);

ll_n = (p) * (pdf / (1-(1+A_NB_K*lambda_nb)**(-alpha))));

if &response = 0 then ll = log(ll_b);
else if &response > 0 then ll = log(ll_n);

%end;

%else %if &dist = %str(ZINB) %then %do;

x1b1u1=&eta1;
x2b2u2=&eta2;
e int=x2b2u2;
lambda nb = exp(x2b2u2);
p=exp(x1b1u1)/(1+exp(x1b1u1));
alpha=1/A_NB_K;

ll_b = p+((1-p)*((alpha/(lambda_nb+alpha))**alpha));

pdf=(gamma(&response+alpha)/(gamma(&response+1)*gamma(alpha)))*((1/(
1+A_NB_K*lambda_nb))**(alpha)*(A_NB_K*lambda_nb/(1+A_NB_K*lambda_nb))**
&response);
ll_n = (1-p) * pdf;

if &response = 0 then ll = log(ll_b);
else if &response > 0 then ll = log(ll_n);

%end;

model &response ~ general(ll);
random u1 u2 ~ normal([0,0],[&vu1,COV_U1U2,&vu2]) subject=&subject
out=rdm;
estimate "P_VAR_U1" EXP(2*P_LOGSDU1);
estimate "A_VAR_U2" EXP(2*A_LOGSDU2);
%if &vargroup ne %STR() %THEN %DO;
estimate "A_VARGRP1" EXP(2*A_LOGSDE);
%let kvg = 2;
%do kvg = 2 %to &NUMVARGROUPS;

```

```

        estimate "A_VARGRP&KVG" EXP(2*(A_LOGSDE + A_DELTAVG&KVG));
    %end;
%end;
%else %do;
    estimate "A_VAR_E" EXP(2*A_LOGSDE);
%end;
estimate "RHO" (EXP(2*Z_U) - 1) / (EXP(2*Z_U) + 1);
estimate "COV U1U2" RHO*EXP(P_LOGSDU1+A_LOGSDU2);
x2b2=x2b2u2-u2;
x1b1=x1b1u1-u1;
predict p out=predp;
predict e_int out=predeint;
predict x2b2 out=predx2b2;
predict x1b1 out=predx1b1;
predict u1 out=_predu1;
predict u2 out=_predu2;
run;
** end of NLMIXED for correlations;

data _null_;
    set convf3;
    call symput('convcr',Reason);
run;

/* check for convergence error. End processing if found */
Data _null_ ;
%let ccscr=%index(&convcr,convergence criterion satisfied) ;
%if &ccscr=%str(0) %then %do ;

    %put *****;
    %put ### Convergence Problem from PROC NLMIXED ;
    %put ### Macro MIXTRAN Stopped Due to Convergence Error in Correlated
Model ;
    %put ### Response Variable is &response ;
    %put ### data is &data;
    %put ### weight variable is &replicate_var;
    %put **ERROR** MSG: &convcr ;
    %put *****;
    %let convflag = 3 ;
    ** IMS SPECIFIC: set flags to note which strata has failed. ;
    %let fail_count = %eval(&fail_count+1) ;
    %let failed = 1;

    data fail&fail_count ;
        fcount=&fail_count;
        food="&eaten";
        subpop="&data";
        run_num="&replicate_var";
        reason="&convcr";
        converr=&convflag;
        output;

```



```

** end of IMS specific code;

** End of IMS SPECIFIC ;
Proc Datasets nolist ;
  delete
    BOXCOX CONV1 CONV2 CONV3 COV COV2 DATA DATA0 EQN1 EQN2 ESTPRINT
    FIT1 FIT2 FITBOTH FITF1 FITF2 FITF3 LAMBDA MODELFITB
    NAMES NAMES2 NEWNAMES1 NEWNAMES2 OUT1 OUT2 OUTCOV PARMSF1
    PARMSF2 PARMSF3 PARMSG1 PARMSG2 PREDEINT PREDEINTU PREDP PREDPU
    PREDU1 PREDU1U PREDU2 PREDU2U PREDX1B1 PREDX1B1U PREDX2B2
    PREDX2B2U RANDOM1 RANDOM2 RNDM START1 START2 START3 STARTM
    TRANS1 TRANS2 _PARAM _PREDICTED _PREDU1 _PREDU2
  ;
  %GOTO convexit;
%end; /* of convergence error check in convcr */
run ;

proc transpose data=adprmsf3 out=adprmsf3(drop=_name_);
  id label;
  var estimate;
run;

data misc_info;
  merge misc_info adprmsf3;
run;

data pred1;
  set rndm;
  if effect='u1';
run;

data pred2;
  set rndm;
  if effect='u2';
run;

data _null_;
  set convf3;
  call symput('convcr',Reason);
run;

** IMS SPECIFIC ;
data _null_;

  %let origvcontrol = &vcontrol;

  %if &vcontrol = %str(0) %then %let vcontrol = %str(); /* special
case, for CORR base runs per Kevin Dodd */

run;

```

```

** END IMS SPECIFIC ;

*****;
** save parmsf3 ;
*****;

    %if &vcontrol = %str() %then %do ;
        data &outlib._parmsf3_&foodtype&vcontrol;
        set parmsf3(keep=parameter estimate);
        run;
    %end;
    *****;
    **** Save parameter estimates and predicted values ****;
    **** from correlated procs ****;
    *****;
    data _param; /* _param data set for correlated part, with version
control giving date and time */
        Format Parameter $30.;
        set parmsf3 ;
        run;

    proc transpose data= param
out=&outlib._param_&foodtype&vcontrol(drop=_name_);
        id parameter;
        var estimate;
        run;

    data &outlib._param_&foodtype&vcontrol ;
        merge &outlib._param_&foodtype&vcontrol
            misc_info ; /* add minamt and other information
*/
        /* if user supplied lambda, restore lambda to output data set
for use in DISTRIB */
        %if &lambdabounds eq %str() %then %do ;
            a_lambda=&amtlambda;
        %end;

    /* save predicted data */

    /* add sum of weights and number of subjects to predx2b2 */

    proc sort data = predx2b2 ; by &subject ;
    data predx2b2 ;
        merge predx2b2
            persons ;
        by &subject ;
    /* _predicted data set for correlated part */
    %if &weekend ne %STR() %then %do; /* weekend variable exists */
        data &outlib._pred_&foodtype&vcontrol (keep=&subject &predxb
&replicate var &weekend &subgroup);
            if (_n_=1) then set &outlib._param_&foodtype&vcontrol;

```

```

set predx2b2(drop=&weekend);
by &subject &repeat;
if (first.&subject);

/* create x1b1, x2b2 for weekday, weekend records*/
&weekend=0;
x1b1_0=(&nonulseq1);
x2b2_0=(&nonu2seq2);

&weekend=1;
x1b1_1=(&nonulseq1);
x2b2_1=(&nonu2seq2);

run;
%end; /* of &weekend ne %str() */

%else %do;
data &outlib.._pred_&foodtype&vcontrol (keep=&subject &predxb
&replicate_var &subgroup);
merge predx1b1 (keep=&subject &repeat pred &replicate_var
rename=(pred=x1b1))
predx2b2 (keep=&subject &repeat &replicate_var pred &subgroup
rename=(pred=x2b2));
by &subject &repeat;
if (first.&subject);

run;
%end;

%put ## the data sets &outlib.._pred_&foodtype&vcontrol and
&outlib.._param_&foodtype&vcontrol ;
%put ## have been output for use in the DISTRIB macro or other software
using CORR models;

** IMS SPECIFIC;
data null ;
%if &origvcontrol = %str(0) %then %let vcontrol = %str(0); /*
reinstate value of vcontrol,special case, for CORR base runs per Kevin
Dodd */
run;

** END IMS SPECIFIC;

*****;
** prepare data for printing summary reports;
*****;
***Skip reports if a re-run, only write reports for a base run ****;
%if &vcontrol=%str() %then %do;
data fit3;
format Parameter $30.;
set fitf3;

```

```

if Descr in ('-2 Log Likelihood','AIC (smaller is better)');
if Descr='-2 Log Likelihood' then do;
    Parameter='ll';
    Name2='-2 Log Likelihood';
    type='4.fit';
end; /* if Descr='-2 Log... */
if Descr='AIC (smaller is better)' then do;
    Parameter='aic';
    Name2='AIC';
    type='4.fit';
end; /* if Descr='AIC ... */
run;

data namesf;
    set &start_val3 fit3;
    keep Parameter Name2 type Value;
run;

proc sort data=namesf ;
    by Parameter;
run;

proc sort data=parmsf3;
    by Parameter;
run;

proc sort data=fitboth;
    by Parameter;
run;

data final;
    Format Parameter $30.;
    merge parmsf3 namesf fitboth;
    by Parameter;
    if Parameter='ll' then do;
        lrtest=m2llo-Value;
        probl1=1-probchi(lrtest,1);
    end; /* if parameter=... */

proc sort data=final;
    by type;
run;

data ll(keep= ll);
    set final;
    if parameter = 'll';
    ll = value;

data aic(keep=aic);
    set final;
    if parameter = 'aic';
    aic = value;

```

```

data numparms;
  Format Parameter $30.;
  merge ll aic;
  pcorr = (aic - ll)/2;
  pgenmod = pcorr - 3;
  porig = pcorr - 1;
  parameter = 'aic';
  run;
%end ;          /* of if vcontrol is blank (base run) for reports */
%end ;          /* of if modeltype eq CORR */
ods select all;
** end of correlated calculations ;
*****;
%if &vcontrol=%str() %then %do ; /* write reports if a base run */
  /*---turn the printing back on---*/
ods select all;

  /*---Report results for Uncorrelated Model---*/

data _null_;
  title%eval(&titles+1) "Results from Fitting Uncorrelated (amount or
nocorr) Model";
  title%eval(&titles+2) "Response Variable: &response";
  title%eval(&titles+3);
  title%eval(&titles+4) "Convergence Status:";
  %if &modeltype ne AMOUNT %then %do ;
    title%eval(&titles+5) "  Probability Model -- &convbi";
    title%eval(&titles+6) "  Amount Model -- &convin";
  %end;
  %else %do;
    title%eval(&titles+5) "  Amount Model -- &convin";
  %end;
  retain aaa 1 ab 2 bb 27 c 52 d 62 e 72;
  file print header=header ll=lines ls=126 ps=80;
  set estprint end=eof; by type;
  if type in ('1.pr', '2.am') then
    put @ab Parameter
        @bb Name2
        @c Estimate 8.4-r
        @;
    ** print SE and probt if weight variable is not used **;
    if "&replicate var" eq "dummywt" then put
        @d StandardError 8.4-r
        @e Probt 6.4-r;
    else put;

  if lines=1 then do;
    put @aaa 80*'_';
  end; /* if lines=1 ... */
  if eof then do;
    put @aaa 80*'_';
  end;

```

```

        end; /* if eof ... */
    return;
header:
    put // @aaa 80*'_';
    put @ab 'Parameter' @bb 'Name' @c 'Estimate'
    @;
    if "&replicate_var" eq "dummywt" then put @d ' Std Err' @e
'Prob>|t|';
    else put;
    put @aaa 80*'_';
    return;
run;

data estprint2;
    set estprint;
    by type Parameter;
    prevval=lag(Value);
    if last.Parameter then sum=Value+prevval;
    run;

data _null_;
    title%eval(&titles+1) "Results from Fitting Uncorrelated (amount or
nocorr) Model";
    title%eval(&titles+2) "Response Variable: &response";
    retain aaa 3 ab 5 bb 40 cc 56;
    file print header=header ll=lines ls=126 ps=70;
    set estprint2 end=eof;
    by type;
    if type in ('4.fit') then put @ab Name2 @bb Value 8.2-r @cc sum 8.2-
r;

    if lines=1 then do;
        put @aaa 70*'_';
    end; /* if lines =1... */
    if eof then do;
        put @aaa 70*'_';
    end; /* if eof ... */
    return;
header:
    put // @aaa 70*'_';
    put @ab 'Name' @bb 'Value' @cc ' Sum';
    put @aaa 70*'_';
    return;
run;

    title%eval(&titles+1);

*****;

/*---Report results for Correlated Model---*/

%if &modeltype eq CORR %then %do ;

```

```

data null ;
title%eval(&titles+1) "Results from Fitting Correlated Model";
title%eval(&titles+2) "Response Variable: &response";
title%eval(&titles+3);
title%eval(&titles+4) "Convergence Status:";
title%eval(&titles+5) "    &convcr";
retain aaa 1 ab 2 bb 27 c 52 d 62 e 72;
file print header=header ll=lines ls=126 ps=80;
set final end=eof;
by type;
if type in ('1.pr','2.am','3.cov') then do;
    put @ab Parameter
        @bb Name2
        @c Estimate 8.4-r
        @;
    ** print SE and probt if weight variable is not used **;
    if "&replicate var" eq "dummywt" then put
        @d StandardError 8.4-r
        @e Probt 6.4-r;
    else put;
end; /* if type 1.pr 2.am 3.co */
if lines=1 then do;
    put @aaa 80*'_';
end; /* if lines=1 then... */
if eof then do;
    if "&replicate var" ne "dummywt" then
        put @1 "Standard errors not printed due to use of replicate
variable &replicate var" ;
    put @aaa 80*'_';
end; /* if eof then... */
return;
header:
put // @aaa 80*'_';
put @ab 'Parameter' @bb 'Name' @c 'Estimate'
    @;
if "&replicate_var" eq "dummywt" then put@d ' Std Err' @e
'Prob>|t|';
else put;
put @aaa 80*'_';
return;
run;

data null ;
title%eval(&titles+1) "Results from Fitting Correlated Model";
title%eval(&titles+2) "Response Variable: &response";
retain aaa 3 ab 5 bb 25 cc 35 dd 50;
file print header=header ll=lines ls=126 ps=70;
set final end=eof;
by type;
if type in ('4.fit') then
    put @ab Name2 @bb Value 8.2-r @cc lrtest 8.2-r @dd probll 6.4-r;
if lines=1 then do;

```

```

        put @aaa 70*' _';
    end; /* if lines = 1 ... */
    if eof then do;
        put @aaa 70*' _';
    end; /* if eof ... */
    return;
header:
    put // @aaa 70*' _';
    put @ab 'Name' @bb 'Value' @cc 'Diff in -211' @dd 'p-value';
    put @aaa 70*' _';
    return;
run;

proc sort data=final;
    by Parameter;
run;

proc sort data=numparms;
    by Parameter;
run;

data finall1;
    merge final(rename=Value=m211_corr) numparms;
    by Parameter;
    if Parameter in ('11','aic');
    convcr = "&convcr";
run;
%end;          /* of if modeltype eq CORR */
%end ;         /* of reports etc. if a base run */
run;

** end of printing correlated reports;

*****;
***** Clean up remaining data sets created/used in macro MIXTRAN only
***** ;

proc datasets nolist;
    delete
        MISC_INFO _PERSONS data data0 &tempda
    ;
    %if &modeltype=%str(CORR) %then %do ;
        delete
            FITF3 PARMSF3 CONVF3
            PREDEINT PREDP PREDU1 PREDU2 PREDX1B1 PREDX2B2 _PREDU1 _PREDU2
        ;
        %if &vcontrol=%str() %then %do ;
            delete
                BOXCOX CONVF1 CONVF2 COV COV2 EQN1 EQN2 ESTPRINT
                ESTPRINT2 FIT1 FIT2 FITBOTH FITF1 FITF2 LAMBDA
                MODELFITB NAMES NAMES2 NEWNAMES1 NEWNAMES2
                OUT1 OUT2 OUTCOV PARMSF1 PARMSF2 PARMSG1 PARMSG2

```



```

PREDEINTU PREDPU PREDU1U PREDU2U PREDX1B1U
PREDX2B2U RANDOM1 RANDOM2 START1 START2 START3
TRANS1 TRANS2
ADPRMSF1 ADPRMSF2 ADPRMSF3 START1VARGRP START2VARGRP
AIC FINAL FINALLL FIT3 LL NAMESF NUMPARMS RNDM
PARAM UNC STARTM PARAM

;
%end ; /* vcontrol = ' ' - base run corr */
%else %do ;
delete
ADPRMSF3 PARAM
;
%end ; /* re-run in corr of clean up of data sets */
%end ; /* CORR */

%else %if &modeltype=%str(NOCORR) %then %do ;
%if &vcontrol = %str() %then %do ; /* base run */
delete
BOXCOX CONV1 CONV2 COV COV2 EQN1 EQN2 ESTPRINT
ESTPRINT2 FIT1 FIT2 FITF1 FITF2 LAMBDA
MODELFITB NAMES NAMES2 NEWNAMES1 NEWNAMES2
OUT1 OUT2 OUTCOV PARMSF1 PARMSF2 PARMSG1 PARMSG2
PREDEINTU PREDPU PREDU1U PREDU2U PREDX1B1U
PREDX2B2U RANDOM1 RANDOM2 START1 START2 START3
STARTM TRANS1 TRANS2
ADPRMSF1 ADPRMSF2 START1VARGRP START2VARGRP
;
%end ; /* vcontrol = ' ' base run in nocorr in clean up of data
sets */
%else %do ; /* if a re-run */
delete
COV COV2 OUTCOV START3 param_unc
PARMSF1 PARMSF2
PREDEINTU PREDPU PREDU1U PREDU2U PREDX1B1U PREDX2B2U
ADPRMSF1 ADPRMSF2 FITF1 FITF2
convf1 convf2
;
%end ; /* re-run of nocorr in clean up of data sets. */

%end; /* NOCORR */
%else %if &modeltype=%str(AMOUNT) %then %do;
%if &vcontrol = %str() %then %do ;
delete
BOXCOX CONV2 COV EQN2 ESTPRINT
ESTPRINT2 FIT2 FITF2 LAMBDA
NAMES NAMES2 NEWNAMES2
OUT2 OUTCOV PARMSF2 PARMSG2
PREDEINTU PREDU2U
PREDX2B2U RANDOM2 START2 START3
STARTM TRANS2 ADPRMSF2 START2VARGRP PARAM UNC
;

```

```

%end ; /* vcontrol = ' ' - base run amount in clean up */
%else %do ; /* re-run */
    delete
    cov cov2 outcov start3 _param_unc
    parmsf2 predu2u predx2b2u
    adprmsf2 fitf2 convf2
    ;
%end ; /* re-run in amount clean up */

%end ; /* AMOUNT */
run;

** succesful conclusion message **;
%let Success = 1; **Reach here only if did not exit early ;

%convexit:

run;

*****Documentation *****;
** clear titles generated inside the macro **;
title%eval(&titles+1) ;

***** draw a line under the end of the macro output in the list file***;
data null ;
    file print;
    put @1 80*' ' ;
    put @1 " End of MIXTRAN Macro Call for &foodtype &modeltype &vcontrol"
    @78 '***'
    ;
    if &success = 1 then put " Execution of MIXTRAN was successful ";
    else put " Execution of MIXTRAN was NOT successful check the log ";
    put @1 80*' ' ;
    run ;

** message to the log **
    %if &success = 1 %then %put ## Execution of MIXTRAN was successful for
&data &replicate_var ;
    %else %put ## Execution of MIXTRAN was NOT successful for &data
&replicate_var - check the log ;
    %put ##



---


    %put; _____ ;
%mend MIXTRAN;

/* END OF THE MIXTRAN MACRO */
/*****

```

## Appendix B: SAC Method DISTRIB SAS Macro for distribution estimation

```
/* *****  
/*  
/* THE DISTRIB MACRO  
/*  
/* *****  
/*          VERSION 3          01/22/2020  
/*  
/*  
/*  
/* The DISTRIB macro uses results from the MIXTRAN macro and  
/* estimates the distribution of usual intake for substance intakes  
/* measured as a count. The data can then be used to calculate percentiles  
/* and, optionally, the percent meeting the recommended daily  
/* intake for a population.  
/*  
/* The DISTRIB macro contains two main functions.  
/*  
/* First, the DISTRIB macro reads data sets of parameter estimates  
/* and predicted values output by the MIXTRAN macro. Monte Carlo  
/* simulation of the random effect(s) is used to estimate the  
/* distribution of usual intake. This data set can be saved.  
/*  
/* Second, once the data containing the estimated usual intake are  
/* available, percentiles and cutpoints can be calculated. The addition  
/* of a sub group variable is accommodated, so that statistics can be  
/* calculated by subgroup and for the overall data set. Optionally the  
/* percent who meet recommended daily intake values can be calculated.  
/*  
/* To accomplish this and allow flexibility, the DISTRIB macro  
/* contains two sub-macros and some general code to set up  
/* and call the macros as requested.  
/*  
/* The macro MC uses monte carlo simulation of the random effect(s) to  
/* estimate the distribution of usual intake.  
/* The output data set can be saved for future use.  
/*  
/* The macro PC reads in the usual intake values calculated in the macro  
/* MC, normalizes the weights, calculates the percentiles of usual intake,  
/* cutpoints if requested, and optionally the percent meeting  
/* recommended intake. A single subgroup variable can be accommodated  
/* in the macro PC. The resulting data set can be saved for future use.  
/* *****;  
/*  
/*  
/* The syntax for calling the DISTRIB macro is:  
/*  
/*      %DISTRIB (call type=, seed=, nsim mc=, modeltype=, dist=,  
/*                pred=, param=, outlib=, cutpoints=, ncutpnt=,  
/*                byvar=, subgroup=, add_da=, subject=,  
/*                titles=, food=, mcsimda=,  
/*                recamt=, recamt_co=, recamt_hi=,
```

```

/*          wkend_prop=);
/*
/* where:
/*
/* "call_type" * Specifies which parts of the DISTRIB macro should
/*               be invoked. (FULL, MC, PC). FULL is the default.
/*               A null string implies FULL.
/*               FULL = invoke both the calculation
/*                     of the estimated intake amount (using monte carlo
/*                     simulation) and of the percentiles (and optionally
/*                     the percent not meeting the recommended amount of
/*                     intake).
/*               MC = restrict the macro to calculating
/*                     the intake amount (using monte carlo simulation).
/*               PC = use the intake estimates - calculated
/*                     in the MC macro in DISTRIB -
/*                     to calculate percentiles and, optionally, the
/*                     percent meeting the recommended amount of
/*                     intake, and/or cutpoints.
/*
/* "seed"      * Specifies the seed for the random number
/*               generator used for the Monte Carlo simulation of
/*               the random effects u1 and u2.
/*               Required if the call_type is FULL or MC.
/*               Not used if call_type is PC.
/*
/* "nsim_mc"   * Specifies the number of repetitions to be used in
/*               the Monte Carlo simulation. For each subject,
/*               one record will be output for each repetition.
/*               Required if the call_type is FULL or MC.
/*               Not used if call_type is PC.
/*
/* "modeltype" * Specifies the model that was used by the MIXTRAN
/*               macro to prepare the data for the DISTRIB macro.
/*               The value must be the same as the model declared
/*               for the MIXTRAN macro. The default is correlated.
/*               The possible values are:
/*               null string = fit correlated model,
/*               corr       = fit correlated model,
/*               nocorr      = fit uncorrelated model,
/*               amount      = fit amount-only model.
/*
/* "dist"      * Specifies the distribution. Use the same distribution as
/*               specified in MIXTRAN.
/*               The possible values are:
/*
/*               "HNB"       = fit Negative Binomial Hurdle
/*                           distribution,
/*               "ZINB"      = fit Zero-Inflated Negative
/*                           Binomial distribution.
/*
/* "pred"      * Specifies the name of the data set containing

```

```

/*          predicted values calculated in the MIXTRAN macro.
/*          Required.
/*
/* "param"      * Specifies the name of the data set containing the
/*                parameter estimates calculated in the MIXTRAN
/*                macro.
/*                Required if the call_type is FULL or MC.
/*                Not used if call_type is PC.
/*
/* "outlib"     * Specifies the library reference to which the
/*                output data set of distributions will be written.
/*
/* "cutpoints"  Specifies a list of cutoff points separated by a
/*                space.
/*                Not used if call_type is MC.
/*
/* "ncutpnt"    Specifies the number of cutoff points.  If cutoff
/*                points are given, ncutpnt must also be given.
/*                Not used if call_type is MC.
/*
/* "byvar"      Specifies a list of by-variables that are in the
/*                data sets "pred" and "param" to indicate that the
/*                MIXTRAN model was fit separately for each
/*                by-group. The estimates used in the calculation will
/*                differ based on the by group, however The DISTRIB
/*                macro will ultimately produce estimates of the entire
/*                population (not distributions within each by group).
/*                To obtain distributions for subpopulations, use the
/*                "subgroup" parameter.
/*
/* "subgroup"   Specifies one categorical variable used for the
/*                calculation of a separate usual intake
/*                distribution for each subgroup. The distribution
/*                of usual intake will also be calculated for the
/*                overall data set. Requires that the parameter
/*                add_da be supplied.
/*                Not used if call_type is MC.
/*
/* "add_da"     The name of the data set containing the subgroup
/*                variable by which the percentiles are to be
/*                calculated, and/or the recommended amount variable.
/*                This data set must include:
/*                the ID variable declared in the parameter SUBJECT,
/*                and one or both of the following variables:
/*                the variable named in the parameter SUBGROUP
/*                the variable(s) named in the parameter(s) recamt and/or
/*                recamt_hi.
/*                This parameter is required if either of the parameters
/*                subgroup or recamt are called.
/*                Not used if call_type is MC.
/*
/* "subject" *   Specifies the variable that uniquely identifies

```

```

/*          each subject. (The ID.)
/*
/*
/* "titles"      Specifies the number of title lines to be
/*                reserved for the user's titles.
/*                The default value is 0.
/*
/* "food"        * Specifies a name for the analysis, used to name
/*                the output data set.
/*
/* "mcsimda"     * Specifies the name of the data set containing the
/*                intake amount derived from the monte carlo
/*                simulations. To read or write the data file from disk
/*                include a libname. Note: due to simulations (see
/*                parameter nsim_mc) the data set can grow quite large.
/*                Default value is work.mcsim in which case the data set
/*                is not saved for later use.
/*                Required if call_type is PC.
/*
/* "recamt"      The name of the variable containing the cut off level for
/*                the recommended amount of consumption for this food.
/*                If the value of the "recamt co" parameter is R then this
/*                variable is used as the lower limit of the range.
/*                If this parameter is used the name of the data set
/*                containing this variable must be supplied via the
/*                parameter "add_da".
/*                Not used if the call_type is MC.
/*
/* "recamt_co"   the Comparison Operator between individual intake and
/*                the recommended amount described in the "recamt"
/*                parameter.
/*                Options are:
/*                LT - less than the "recamt" value,
/*                LE - less than or equal to the "recamt" value,
/*                GE - greater than or equal to the "recamt" value,
/*                GT - greater than the "recamt" value,
/*                R  - a range of values between two proportions,
/*                inclusive.
/*                If the R option is used then the lower value will be the
/*                value of the variable in the "recamt" parameter, and the
/*                upper value must be provided via the parameter
/*                "recamt_hi".
/*                The "recamt co" parameter is required if the "recamt"
/*                parameter is supplied.
/*                Not used if the call type is MC.
/*
/* "recamt_hi"   The name of the variable containing the upper limit of
/*                of a range of inclusive values used to compare intake
/*                to a recommended amount. This parameter is required if
/*                the value of the "recamt_co" parameter is R.
/*                Not used if the call type is MC.
/*

```

```

/* wkend_prop      A value between 0 and 1 (not inclusive).
/*                This parameter specifies the proportional weight for the
/*                weekend days if "weekend" was used in the MIXTRAN macro.
/*                Either a fraction or decimal number is acceptable.
/*                The remaining (e.g. weekday) proportion is calculated
/*                within the macro. The default weights for weekdays
/*                and weekend days are 4/7 and 3/7 respectively.
/*                Note: it is possible to use the "weekend" and
/*                "wkend_prop" parameters with a binary variable
/*                other than a weekend indicator.
/*                Not used if the call_type is PC.
/*
/* Note:  * Parameters marked with an asterisk are mandatory, a
/*        value must be supplied in the macro call.
/*
/*****
;

%macro DISTRIB (call_type=,seed=, nsim_mc=, response = , modeltype=,
dist=, pred=, param=,
               outlib=, cutpoints=, ncutpnt=, byvar=,
               subgroup=,add da=,
               subject=, titles=, food=, mcsimda=,
               recamt=, recamt_co=, recamt_hi=,
               wkend_prop=);

*****
;

%macro MC;
/*****
/* Macro to define the intake using monte carlo simulations      */
/* This macro is invoked if the call type is FULL, MC or null    */
/*****

/* merge parameter and predicted data sets */
data _predicted2;
    %if (&byvar = %str()) %then %do;    /* allow different estimates by by-
group. */
        set &pred;
        if (_n_ = 1) then set _param;
    %end;
    %else %do;
        merge &pred _param;
        by &byvar;
    %end;

/* set up variables for one part models  in monte carlo repetitions */

%if &modeltype ne CORR %then %do;
    rho = 0;

```

```

%if &modeltype eq AMOUNT %then %do;
  cov_ulu2 = 0 ;
  z_u      = 0;
  p_var_u1 = 0;
  %if &wkendflag eq %str(1) %then %do;
    x1b1_0 = . ;
    x1b1_1 = . ;
  %end ;          /* of if weekend */
  %else %do;
    x1b1 = . ;
  %end ;          /* of not weekend */
%end;             /* of model = AMOUNT */
%end;             /* of setup for model ne CORR */

  if p_var_u1 > 0 then stddev_u1 = sqrt(p_var_u1);
  if p_var_u1 = 0 then stddev_u1 = 0; /* changes for amount strata to
process all models in simulations */
  stddev_u2 = sqrt(a var u2);
  if cov_ulu2 = 0 then corr_ulu2 = 0;
  else corr_ulu2 = COV_U1U2 / (stddev_u1 * stddev_u2);

run;

*****;

/* monte carlo simulations of intake. */

data &mcsimda ;
;
  retain seed &seed;
  set _predicted2 end=eof;

  /* Keep only variables needed. Which ones depend whether this is a
weekend run */
  %if &wkendflag eq %str(1) %then %do ;
    %if &modeltype ne %str(AMOUNT) %then %let mcp = %str(mc_p mc_p_0
mc_p_1);
    %else %let mcp = ;
    keep
      &subject mcsim_wt numsim
      mc_t
      &mcp
      mc_a
    ;
  %end ; /* of keep for weekends */
  %else %do ;
    %if &modeltype ne %str(AMOUNT) %then %let mcp = %str( mc_p);
    %else %let mcp = ;
    keep

```



```

        &subject mcsim_wt numsims
        mc_t &mcp mc_a
    ;
%end ;    /* of keep for not weekends */

/* arrays for 9 point approximation */
    array fbt (9) fbt1-fbt9;    /* new code from Kevin & Janet
Tooze 03/16 and 3/11 2010 */
    array xbt (9) xbt1-xbt9;
    array bt (9) bt1-bt9;
    array cj (9) cj1-cj9 (-2.1 -1.3 -0.8 -0.5 0 0.5 0.8 1.3 2.1);
    array wj (9) wj1-wj9 (0.063345 0.080255 0.070458 0.159698
0.252489 0.159698 0.070458 0.080255 0.063345);

/* divide the individual weight by the number of repetitions */
mcsim_wt=&freq_var/&nsim_mc ;

/* keep the number of simulations to divide into the number of
subjects in the pc macro */
numsims=%eval(&nsim_mc);

/* assign variance of e here */
%if &numvargrps =%str(2) and &wkendflag = %str(1) %then %do;
    a_var_e_0 = a_vargrp2;
    a_var_e_1 = a_vargrp1;
%end;
%else %if &numvargrps gt %str(2) %then %do;
    put "User ERROR: The DISTRIB macro does not process more than two
variance groups at this time. ";
%end;

/* generate monte carlo random effects. */

do mcsim = 1 to &nsim_mc;
    /* calculate u1 and u2 */

    call rannor(seed,u1);
    call rannor(seed,u2);
    u2 = corr_u1u2 * u1 + sqrt(1 - corr_u1u2**2) * u2;
    u1 = stddev_u1 * u1;
    u2 = stddev_u2 * u2;

    /* for weekend runs calculate intake from x1b1_0, x1b1_1 x2b2_0 &
x2b2_1 */
    /* for runs with no weekend use x1b1 and x2b2
*/
    %if &wkendflag eq %str(1) %then %do;
        %let wk = _0 ;
        %let wkgr = 2 ;
        %if &numvargrps ne %str(0) %then %let ve = _0 ;
        %else %let ve = ;    /* for variance of e
*/

```

```

%end; /* wkendflag = 1 */
%else %do ;
    %let wk = ;
    %let wkgr=1;
    %let ve = ;
%end; /* wkendflag ne 1 */
%do i = 1 %to &wkgr;

    ** UPDATED 02/10. In Amount models x1b1_0 and x1b1_1 have been set
    to missing. **;

    %if &dist=%str() %then %let dist=%str(HNB); **Default will run HNB
    distribution ;

    %if &dist = %str(HNB) %then %do;

        if x1b1&wk ^= . then mc_logit_p&wk = x1b1&wk + u1; /* u1 will
        always be nonmissing, but may be zero */

        /*%if mc_logit_p&wk ^= . %then %do;

            if &response = 0 then mc_logit_p1_p&wk = mc_logit_p&wk;
            if &response > 0 then mc_logit_p2_p&wk = log(1+
exp(mc_logit_p&wk));

        %end;

            if mc_logit_p&wk ^= . then mc_p&wk =( mc_logit_p1_p&wk -
mc_logit_p2_p&wk);*/
            if mc_logit_p&wk ^= . then mc_p&wk = 1 / (1 + exp(-
mc_logit_p&wk));
            else if stddev_u1 = 0 then mc_p&wk = 1; /* make sure the
missing value was on purpose for an AMOUNT run */

            if x2b2&wk ^= . then mc_bca&wk = x2b2&wk + u2;
            /*if &response > 0 then mc_a&wk = ((&response*mc_bca&wk) -
exp(mc_bca&wk)-log(1-exp(-exp(mc_bca&wk)))) - log(fact(&response));*/
            if mc_bca&wk ^= . then mc_a&wk = exp(mc_bca&wk);/*use GLM
function
http://dept.stat.lsa.umich.edu/~kshedden/Courses/Stat600/Notes/glm.pdf*/
            else if stddev_u2 = 0 then mc_a&wk = 1;

            mc_t&wk = mc_p&wk*mc_a&wk;

        %end;

    %else %if &dist = %str(ZINB) %then %do;
        if x1b1&wk ^= . then mc_logit_p&wk = x1b1&wk + u1; /* u1 will
        always be nonmissing, but may be zero */
        if mc_logit_p&wk ^= . then mc_p&wk = 1 / (1 + exp(-
mc_logit_p&wk));

```

```

        else if stddev_u1 = 0 then mc_p&wk = 1;          /* make sure the
missing value was on purpose for an AMOUNT run */

        if x2b2&wk ^= . then mc_bca&wk = x2b2&wk + u2;
        if mc_bca&wk ^= . then mc_a&wk = (1-mc_p&wk)
*exp(mc_bca&wk);/*use https://stats.idre.ucla.edu/stata/faq/how-can-i-
manually-generate-the-predicted-counts-from-a-zip-or-zinb-model-based-on-
the-parameter-estimates/ */
        /*
http://www.karlin.mff.cuni.cz/~pesta/NMFM404/zinb.html*/
        else if stddev_u2 = 0 then mc_a&wk = 1;

        *****;

        mc_t&wk = mc_p&wk*mc_a&wk;

        %end;

        %let wk= 1;
        %if &numvargrps ne %str(0) %then %let ve= _1 ;
        %else %let ve = ;          /* for variance of e
*/

        %end;    /* of i = 1 to wkgr */

        %if &wkendflag = %str(1) %then %do ;
        mc_t=((&wkday_prop*mc_t_0)+(&wkend_prop*mc_t_1));
        mc_a=((&wkday_prop*mc_a_0)+(&wkend_prop*mc_a_1));
        %if &modeltype ne %str(AMOUNT) %then %do;
        mc_p=((&wkday_prop*mc_p_0)+(&wkend_prop*mc_p_1));
        %end;    /* wkendflag = 1 (not Amount) */          %end;    /* wkendflag =
1 */

        output ;

        end;          /* of monte carlo simulations through _mcsim */

        if (eof) then call symput("seed",trim(left(put(seed,12.))));

        run;

        %mend ;          /*end of monte carlo simulations macro mc ; */
        *****
        *****;
        *****
        *****;

        /*****
        ***** */
        /* start of the percentiles macro */
        /*****
        ***** */

```

```

%macro PC ;

/*****
*****/
/* the percentiles macro (PC) is invoked if the call_type is FULL, PC or
null *****/
/*
/* this macro will merge subgroup data if any to the estimated intake data
/* saved by the macro MC. The intake data will be used to:
/* calculate the normalised weight (by subgroup if any and for all
groups combined);
/* figure the percentiles, and cutpoints if requested;
/* create the percentage meeting the recommended amount of intake if
requested ;
/* (all by subgroup if any and including the overall group);
/* and write out a data set containing the percentiles and other
descriptive information */
*****/
*****/

/* general setup for macro PC */

/* if there is a subgroup or recommended amount then read in the
additional data file */

** for backward compatibility the subgroup could be on the predicted
data set or in add_da **;
** Compatibility required by Janet Tooze ** ;
** if a subgroup variable is specified then first check the predicted
data set.
** if it is not in the predicted data set look for it in the
additional data set. **;

%let check = %str(0) ;
%let addcheck = %str(0) ;
%if &subgroup ne %str() %then %do ;
    %let dssub=%sysfunc(open(&pred,is));
    %if &dssub = 1 %then %do;
        /*data _null_;
        dset=open("&pred");
        call symput ('check',varnum(dssub,"&subgroup")); */
        %let check=%qsysfunc(varnum(&dssub,&subgroup));
        %let rcsb=%sysfunc(close(&dssub));
    %end ; /* of checking for the subgroup variable in the predicted
data set */
run;

%if &check ne 0 %then %do ;
    %let psub=%str(_predsub) ;
    proc sort data=&pred out=&psub(keep=&subject &subgroup) nodupkey
;

```

```

        by &subject ;
    %end;    /* of there is a subgroup variable in the predicted data set
*/
    %end ;    /* of subgroup ne blank */

    /* if necessary read in the additional data */
    %if &recamt ne %str() or (&subgroup ne %str() and &check = %str(0))
%then %do;
        %let addcheck = %str(1) ;
        %let altsubgroup = &subgroup;                ** more backward
compatibility **;
        %if &check ne %str(0) %then %let altsubgroup=%str() ; ** case where
the subgroup is in predicted and recamt is being called;

        proc sort data=&add_da out = _subs(keep = &subject &altsubgroup
&recamt &recamt_hi) nodupkey;
            by &subject ;
        run;
    %end ;    /* of reading in additional data (add_da) if needed */

    %if &check ne %str(0) %then %do;

        %if &addcheck = 1 %then %do ;
            data _subs ;
            merge   predsub
                    _subs ;
            by &subject;
        %end;    /* of both add_da and _predSubs exist */
        %else %do ;
            proc datasets nolist library=work ;
                change   predsub =   subs ;
            %end;    /* _predsubs exists but not add_da */
        %end;    /* of _predsub exists */
        run;

    /* set up the subgroup code for use in the percentile calculations
*/

    %if &subgroup = %str( ) %then %do ;
        %let first_subgroup = %str( ) ;
        %let last_subgroup  = %str( ) ;
    %end;
    %if &subgroup ne %str( ) %then %do ;
        %let first_subgroup = %str(if first.&subgroup) ;
        %let last_subgroup  = %str(or last.&subgroup)   ;

    /* create a value for the overall subgroup (all records) if subgroup
is used */

    *****;
    /* ** if a subgroup variable is in use, the PC macro will create an

```

```

    ** overall category. This will be coded either _overall or
    ** -255 depending on the type of the variable in the subgroup.
    ** If subgroup has been identified a second record will be output
    ** each time with the overall subgroup code and an adjusted
weight
    ** and number of subjects.
    ** Note that the intake values must have already been calculated
    ** in the macro MC, a part of the macro DISTRIB.
*/

    /* find the variable type of the subgroup, if any, for the overall
category */

    %let dsid=%sysfunc(open(work._subs,is));
    %if &dsid = 1 %then %do;
        %let subnum=%qsysfunc(varnum(&dsid,&subgroup));
        %let subtype=%qsysfunc(vartype(&dsid,&subnum));
        %let rc=%sysfunc(close(&dsid));
    %end; /* of reading in variable type */
    %else %put " user error: unable to assign subgroup variable type
";

run;

    /* end of code to find if subgroups is character or numeric */
%end ; /* of if a subgroup */
*****;

/* read in the mcsim data file */

Proc sort data=&mcsimda out=_mcsim1;
by &subject;

/* subset to records in the predicted data set */

proc sort data=&pred out=_subset(keep=&subject) nodupkey ;
by &subject;

data _mcsim1 _notinmc(keep=&subject);
merge _mcsim1 (in=inmc)
      _subset (in=insub);
by &subject ;
if inmc and insub then output mcsim1 ;
else if insub then output _notinmc;
run;
proc print data= notinmc ;
title%eval(&titles+1) "records in the subset but not in the mc_sim
data";
run;

/* If subgroup or recamt merge with supplemental file */
/* and create the recommended amount flag if requested */

```

```

    %if &recamt ne %str() or &subgroup ne %str() %then %do;
data _mcsim1 _nomatch;
    merge _mcsim1 (in=inmc)
           _subs (in=insub);
    by &subject ;
    if inmc and insub then do;

        /* recommended amount: - flag success */

        %if &recamt ne %str() %then %do;
            if mc_t ne . then &recamtflag = 0 ;
            %if &recamt_co ne %str(R) %then %do; /*
not a range. values lt, le, ge, gt) */
                if mc_t &recamt_co &recamt then &recamtflag = 1; /*
assign 1 if true*/
            %end;
            %else %do ;
                if &recamt le mc_t le &recamt_hi then &recamtflag = 1 ; /*
inclusive range */
            %end;
        %end; /* of assigning recommended amount flag */

        output _mcsim1 ;

    end; /* of inmc and insub */

    else output _nomatch ;

run;

proc print data=_nomatch (obs=1) ;
    title%eval(&titles+1) 'First observation of data not matching in the
intake data and the supplemental data';
run;
%end ; /* of merging the supplemental data onto mcsim */

/* If subgroup is used output one new record for every observation for
an overall level */

%if &subgroup ne %str( ) %then %do ;
data _mcsim1 ;
    set _mcsim1 ;

output;

    %if &subtype = %str(N) %then %do ; /* numeric variable */
        &subgroup = -255 ;
    %end;
    %else %if &subtype=%str(C) %then %do ; /* character variable */

```

```

        &subgroup = "_Overall" ;
    %end;
    %else %do ;
        %put ## Error in recode of subgroups, vartype = &subtype ;
    %end;
    output _mcsim1;
%end ;          /* of adding the overall record for the subgroup */

*****;
/* calculate group weights and total weights. */
proc means data= mcsim1 nway noprint;
    %if &subgroup ne %str() %then class &subgroup ; ;
    var mcsim_wt ;
    output out=_grpinfo(keep=grpwts numsubjects &subgroup) sum=grpwts
n=numsubjects ;
run;

/* merge the weights onto the mcsim data */
/* and calculate the adjusted weights by subgroup and overall */

%if &subgroup ne %str() %then %do ;
    proc sort data=_mcsim1 ;
        by &subgroup;
    %end;

data _mcsim1 ;
    %if &subgroup ne %str() %then %do ;
        merge _mcsim1 end=eof
            _grpinfo
            ;
        by &subgroup;
    %end ;
    %else %do ;
        set mcsim1 end=eof ;
        if _n_ =1 then set _grpinfo ;
    %end;

    /* calculate the adjusted weight */

    adjwt=mcsim_wt/grpwts;
    output ;
    /* output the adjusted weight and the intake */
    totsum+adjwt ;          /* sum of adjwts = 1 if no subgroups, or # of
subgroups +1 */
    if eof then put "the sum of the adjusted weights = " totsum ;

*****
;

***** sort by mc_t for calculation of percentiles *****;

```



```

proc sort data= _mcsim1; ;
  by &subgroup mc_t;
run;

*****
*
** calculate the percentiles by linear interpolation.
*****
*;

data outpercentiles;

  set _mcsim1 end=finalrec;
  by &subgroup mc_t;

  retain jperc 1
         adjwt_ties wsum calcflag 0
         last_wsumplush last_mc_t
         tpercentile0-tpercentile100 horizline0-horizline100 ;
  ;
  %if &ncutpnt ne %str() %then %do ;
    retain kcut 1
           cutcalcflag 0
           &cutprobs
           &cutverts
    ;
    %end ;      /* of retain if cutpoints used */

  array horizline(101) horizline0-horizline100;
  array tpercentile(101) tpercentile0-tpercentile100;
  %if &ncutpnt ne %str() %then %do ;
    array cutvertline(&ncutpnt) &cutverts;
    array cutprob(&ncutpnt) &cutprobs;
  %end ;

  ***** read in cutpoints and generate values for the horizontal lines
  ***** ;
  if _n_ = 1 then do ;
    do j = 0 to 100 ;
      horizline(j+1) = j/100 ;
    end;
    %if &ncutpnt ne %str() %then %do ;
      %do c = 1 %to &ncutpnt ;
        cutvertline&c = %scan(&cutpoints,&c,%str( )) ;
      %end ;
    %end ;      /* of reading in the cutpoints */
  end ;      /* of _n_ = 1 on percentiles calculation data */

  %if &subgroup ne %str() %then %do ; /* initialise if subgroup used */
    if first.&subgroup then do;
      do j = 0 to 100;
        tpercentile(j+1) = . ;
      end;
    end;
  end;

```

```

end;
jperc = 1;
adjwt_ties = 0;
wsum = 0 ;
calcflag = 0 ;
last_wsumplush = . ;
last_mc_t = . ;
sortorder=0;
%if &ncutpnt ne %str() %then %do ;
    kcut = 1 ;
    cutcalcflag = 0;
    %do c = 1 %to &ncutpnt ;
        cutprob&c = . ;
    %end;
%end ;          /* of resetting cutprob values & flags to missing if
subgroups */
end;
%end ;          /* initialising if subgroup */

adjwt_ties = adjwt_ties + adjwt; **** Calculate adjusted weights for
tied t values;

if last.mc_t then do;

    halfw = adjwt_ties/2;
    wsumplush = wsum + halfw;

    *** Need 2 points to calculate slope and intercept. Therefore, skip
calculations for first t value ****;
    do while ( calcflag=1 and ((wsumplush >= horizline(jperc)) or
(finalrec=1) &last_subgroup) );

        **** use linear interpolation to obtain percentiles;

        slope = ( wsumplush - last_wsumplush ) / ( mc_t - last_mc_t );
        interceptb = wsumplush - slope * mc_t;

        tpercentile(jperc) = max(0, ( horizline(jperc) - interceptb ) /
slope);

        if jperc = 101 then calcflag = 0; **** All percentiles calculated.
Set flag to end loop;
        else jperc = jperc + 1;

    end; /* of do while for calculating percentiles*/

*****;
    *** do while for cut points if requested */

```

```

    %if &ncutpnt ne %str() %then %do ;

        *** Need 2 points to calculate slope and intercept.  Therefore, skip
calculations for first t value ;
        do while ( cutcalcflag=1 and ((mc_t >= cutvertline(kcut)) or
(finalrec=1) &last_subgroup) );

            **** use linear interpolation to obtain cutpoint probabilities;

            slope = ( wsumplush - last_wsumplush ) / ( mc_t - last_mc_t );
            interceptb = wsumplush - slope * mc_t;

            tempcutprob = max(0, ( slope * cutvertline(kcut) + interceptb ));
            cutprob(kcut) = min(1,tempcutprob);

            if kcut = &ncutpnt then cutcalcflag = 0;  **** All cutpoint
probabilities calculated.  Set flag to end loop;
            else kcut = kcut + 1;

        end;    /* of do while for cutpoint probabilities */

    %end ;    /* of making cutpoint probabilities if requested */

    *****;

    if finalrec &last_subgroup then output outpercentiles;
    else do;    *** prepare variable values for calculations involving
next t value;
        adjwt_ties = 0;
        wsum = wsumplush + halfw;
        last wsumplush = wsumplush;
        last_mc_t = mc_t;
        calcflag=1;
        %if &ncutpnt ne %str() %then %do ;
            if kcut < &ncutpnt then cutcalcflag=1;
        %end ;
    end;

    end;  /* of last.mc_t */

run;

*****
*
** end of percentiles and cutpoints code
*****
**;

/* combine percentiles, cutpoint probabilities, weighted mean of mc_t and
number of subjects
/* and propotion meeting recommended consumption if desired

```

```

/* into one data file and save it. The data set will be saved to
"&outlib.".
/* the name will be 'descript' followed by the food name in &food and if a
/* weight variable was used then also by the name of the weight variable
*/

proc means data=_mcsim1 noprint nway;
  %if &subgroup ne %str() %then %do ;
    class &subgroup ;
  %end;
  var mc t &recamtflag;
  weight adjwt ;
  output out=_mean mean=mean_mc_t &proprec;

  /* sort groupinfo if subgroup, and subset means to subgroup level data
  */
  %if &subgroup ne %str() %then %do ;
    proc sort data=_grpinfo ; by &subgroup ;
  %end;

  /* if the weight variable is named 'dummyst' then do not add the value
  to the output names */
  /* because it means there was no weight variable in MIXTRAN so a dummy
  weight of 1 was substituted */
  %if &freq_var = %str(dummywt) %then %let freq_var= %str();

  data &outlib..descript_&food._&freq_var (keep= &subgroup numsubjects
  mean_mc_t &proprec
          tpercentile0-tpercentile100 &cutprobs);
  merge _mean (keep=&subgroup mean_mc_t &proprec _type_ )
        outpercentiles
        _grpinfo
        ;
  %if &subgroup ne %str() %then %do ;
    by &subgroup ;
    if first.&subgroup ;
  %end ;
  %else %do ;
    if _n_ =1 ; /* no subgroups, only need the number of subjects
overall */
  %end ;
  %if &recamt ne %str() %then &proprec=&proprec*100; ; /* change
proportion failing rec amt to percent */
  numsubjects=numsubjects/numsims; /* numsubjects
calculated from simulated data */

run;

** CHECK;
Proc print data=&outlib..descript_&food._&freq_var noobs uniform;

```

```

    %if (&subgroup ^= %str()) %then id &subgroup%str(;;);
    var numsubjects mean mc t &proprec
        tpercentile1 tpercentile5 tpercentile10 tpercentile15
        tpercentile25 tpercentile50 tpercentile75
        tpercentile85 tpercentile90 tpercentile95 tpercentile99
        &cutprobs ;
    title%eval(&titles+1) "Selected Percentiles and Cutpoint Probabilities
from the Distribution &food &freq_var";
run;

%if %eval(&syscc) = 0 %then %do;
    %put ## the data set output by DISTRIB will be
&outlib..descript_&food._&freq_var ;
%end; /* of note to user re output data set */

proc datasets lib=work nolist ;
    delete
        _mcsim1 /* to keep _mcsim1 for further analysis in this
program comment it out */
        outpercentiles
        _mean
        grpinfo
        _param
        _subset
        notinmc
        _subs
        ;
    %if &call_type = %str(FULL) %then %do ;
        delete _predicted2 ;
    %end;
    %if &subgroup ne %str() %then %do;
        delete _nomatch ;
    %end;

run;
%mend ; /* end of percentiles macro pc */

*****
*****;
*****
*****;

*****
/* general set up for all calls to the DISTRIB macro*/
*****;

%let success = 0 ; /* successful execution flag */

%put ## In the DISTRIB macro: ;

/* for backward compatibility with the tutorials, make sure there is an
mcsim data set */

```

```

    %if &mcsimda=%str() %then %let mcsimda=work.mcsim ;

/* read in the parameter data set and create macro variables needed from
_param */
data _param ;
    set &param ;
        if _n_ =1 then do; /* get the weight variable, the number of variance
                            /* groups and the flag indicating a weekend variable
                            /* if any of these were used in the MIXTRAN macro
*/
        call symput('freq var',FreqName);
        %if %upcase(&call_type) ne %str(PC) %then %do ;
            call symput('Numvargrps',left(Numvargrps));
            call symput('wkendflag',weekendflag);
        %end;

    end;
run;

%let call_type=%upcase(&call_type);
%if &call_type = %str() %then %let call_type=%str(FULL);
%let modeltype=%upcase(&modeltype);
%let dist=%upcase(&dist);
%let response = %upcase(&response);
%let freq var=%left(%trim(&freq var));
/* If no Title lines reserved set titles to 0 */
%if &titles = %str() %then %let titles=0;
/*%if &freq_var = %str(dummywt) %then %let freq_var= %str(); */

/* set up a macro variable to keep the cut point probabilities if declared
*/
%if call_type ne %str(MC) %then %do ;

    %if (&ncutpnt = %str() and &cutpoints ne %str() ) or (&cutpoints =
%str() and &ncutpnt ne %str() ) %then %do ;
        %let ncutpnt = %str() ;
        %let cutpoints = %str() ;
        %put ## user warning -both cutpoints and number of cutpoints must be
given. ;
        %put ##                Cutpoints will be ignored;
    %end; /* need both cutpoints and number of cutpoints */

    %if &ncutpnt ne %str() %then %do ;
        %if &ncutpnt = %eval(1) %then %do ;
            %let cutprobs=%str(cutprobl) ;
            %let cutverts=%str(cutvertline1) ;
        %end; /* only one cutpoint */
        %else %do ;
            %let cutprobs=%str(cutprobl-cutprob&ncutpnt) ;
            %let cutverts=%str(cutvertline1-cutvertline&ncutpnt) ;

```

```

    %end;      /* more than one cutpoint */
%end;        /* have cutpoints */

%else %do ;   /* no cutpoints */
    %let cutprobs=%str() ;
    %let cutverts=%str() ;
%end ;        /* no cutpoints */

/* if the proportion meeting the recommended amount of consumption is
to be calculated */
%if &recamt ne %str() %then %do ;
    /* check for supplemental data set with the recamt variable name */
    %if &add_da =%str() %then %do;
        %put ## USER ERROR: the data set name containing the variable
&recamt must be supplied in the parameter add_da;
        %goto DISTEXIT;
    %end ;    /* of check for supplemental data for the recamt variable */
    %let recamt_co=%upcase(&recamt_co);
    %if &recamt_co=%str() %then %do ;    /* need a value for recamt_co */
        %put ## USER ERROR: a comparison value is required in the recamt_co
parameter for the recommend amount.;
        %put ##                Either add a value to the recamt_co parameter
or remove the value from the recamt parameter.;
        %goto DISTEXIT;
    %end;    /* of recamt co is blank */
    %else %do ;                                /* check that the value for
recamt_co is valid*/
        %let validth = (LT LE GE GT R);
        %let t=%index(&validth,&recamt_co);
        %if %eval(&t) = 0 %then %do;
            %put ## USER ERROR. The value "&recamt_co" for the parameter
recamt_co is not valid. ;
            %goto DISTEXIT;
        %end;    /* of invalid value for recamt co (other than blank) */
        %else %if &recamt_co = %str(R) and &recamt_hi = %str() %then %do ;
/* check for the upper range value */
            %put ## USER ERROR: The parameter recamt_co specified a range
(value=R). ;
            %put ##                No upper value for the range is given ;
            %put ##                Please supply a variable name for the
parameter recamt_hi ;
            %put ##                or change the value of the parameter
recamt co ;
            %goto DISTEXIT;
        %end;    /* upper value for range if recamt_co = R */
    %end ;    /* recamt_co not blank */

    %let recamtflag = &recamt._flag ;
    %let proprec = percent_rec_amt;
%end; /* there is a recamt - set up */
%else %do ;
    %let recamtflag = ;

```

```

        %let proprec = ;
        %end; /* there is no recamt */

%end ; /* of if call_type ne MC t*/
run;

/* determine the proportions to be used in the case of "weekend" runs */
%if &call type ne %str(PC) %then %do ;
    %if &wkendflag eq %str(1) %then %do;
        %if &wkend_prop ne %str() %then %do ;
            %if %sysevalf(&wkend_prop) ge 1 %then %do ;
                %put ## USER ERROR: the weekend proportion must be less than 1.
The proportion given is &wkend_prop ;
                %goto DISTEXIT ;
            %end ; /* of wkend_prop ge 1 */
            %let wkday_prop = %sysevalf(1-&wkend_prop);
        %end; /* of wkend_prop not blank */
        %else %do ;
            %let wkday_prop = 4/7 ;
            %let wkend_prop = 3/7 ;
        %end; /* of default use for weekend proportions */
    %end ; /* of wkendflag 1 */
%end; /* of determining the weekend proportions if not PC */

/* notes to the log to help the user */

%put ## the call_type is &call_type ;
%put ## the model type is &modeltype ;
%put ## the response variable is &response ;
%put ## the distribution is &dist ;
%put ## parameter data set = &param ;
%put ## predicted data set = &pred ;
%put ## the food variable is &food ;
%if &freq var ne %str(dummywt) %then %put ## the weight variable name is
&freq_var ;
%if &byvar ne %str() %then %put ## the by-variable is &byvar ;
%if &call type ne %str(PC) %then %do ;
    %if &wkendflag = %str(1) %then %do ;
        %put ## coded for a weekend variable ;
        %put ## the proportional weights for the "weekend" are: &wkday_prop
and &wkend_prop ;
    %end;
    %if &numvargrps ne %str(0) %then %put ## the number of variance groups
is &numvargrps ;
%end;
%if &call type ne %str(MC) %then %do;
    %if &subgroup ne %str() %then %put ## the subgroup variable is
&subgroup ;
    %if &add_da ne %str() %then %put ## the data set containing additional
information is &add_da;
    %if &ncutpnt ne %str() %then %do;
        %put ## the number of cutpoints is &ncutpnt;
    %end;
%end;

```



```

    %put ## the cutpoints are &cutpoints ;
    %end ;
%end;
%if &recamt ne %str() %then %put ## recommended amount consumed will be
tested using &recamt_co  &recamt &recamt_hi;

*****;
/* determine which parts of the DISTRIB macro to invoke */

%if &call_type = %str(FULL) %then %do;
    %mc ;
    %pc ;
%end;
%else %if &call_type = %str(MC) %then %mc ;
%else %if &call_type = %str(PC) %then %pc ;
%else %do ;
    %put ## USER WARNING:  INVALID call_type in the call to the DISTRIB
macro ;
    %goto DISTEXIT ;
%end;

    %let success=1;      /* Flagging for succesful run */

%DISTEXIT:
;
%if &success ne 1 %then %do ;
    %put ## THE DISTRIB MACRO FAILED.  PLEASE CHECK THE LOG. ;
%end;
%else %put ## The DISTRIB macro completed succesfully ;

** clean up titles ** ;
title%eval(&titles+1);

*****
/* end of general set up for all calls to the DISTRIB macro*/
*****;
%mend ;  /* end of DISTRIB macro; */

```

## Appendix C: SAS code to make replicate BRR weights for NLSY 2013-2015

```
title 'Create BRR weights for NLSY 2013-2015';

*make brr weights ;
*call libraries;

LIBNAME NH "C:\...\NH";
LIBNAME OUT "C:\...\OUT";

DATA data (keep= PUBID SAMPLING_PANEL_WEIGHT13 /*DR1DRSTZ*/ SEX RACE
RACE_ETHNICITY VSTRAT VPUS);
    SET nh.alcohol; /*dataset for SAC method analysis with weight,
cluster, and strata variables*/
    run;

proc sort data=data;
    by SEQN;
run;

data INDIV;
    * Combine demographic info with dietary weights for people in the
    dietary dataset only;
set data;
    * Make poststratification variable PSCCELL (post-stratification cell);
    * use six age groups as directed ;
select;

    when (RACE_ETHNICITY=1 and SEX=1) PSCCELL=1;
    when (RACE_ETHNICITY=2 and SEX=1) PSCCELL=2;
    when (RACE_ETHNICITY=3 and SEX=1) PSCCELL=3;
    when (RACE_ETHNICITY=4 and SEX=1) PSCCELL=4;

    when (RACE_ETHNICITY=1 and SEX=2) PSCCELL=5;
    when (RACE_ETHNICITY=2 and SEX=2) PSCCELL=6;
    when (RACE_ETHNICITY=3 and SEX=2) PSCCELL=7;
    when (RACE_ETHNICITY=4 and SEX=2) PSCCELL=8;

end;
run;
*-----
;
* Identify any strata with more than 2 PSUs
;
*-----
;
proc freq data=INDIV noprint;
```

```

        tables VSTRAT*VPUS/out=chk1 (keep=VSTRAT VPUS);
run;
proc freq data=chk1 order=FREQ;
    tables VSTRAT/nopercent nocum;
    title2 "Check for strata with more than 2 PSUs";
    title3 "Any problem strata are listed first";
run;
proc datasets nolist;
    delete chk1;
run;
*-----
;
* Collapse PSUs in 3-PSU strata
;
*-----
;

/*
data INDIV;
    set INDIV;
    if (SDMVSTRA=86 and SDMVPSU=2) then SDMVPSU=3;
    if (SDMVSTRA=90 and SDMVPSU=2) then SDMVPSU=3;
    if (SDMVSTRA=91 and SDMVPSU=2) then SDMVPSU=3;
    if (SDMVSTRA=92 and SDMVPSU=2) then SDMVPSU=3;
run;

*/
*-----
;
* Get Control Totals by PS Cells
;
*-----
;
proc freq data=INDIV noprint;
    tables PSCELL/out=CTRL (rename=(COUNT=_PSTOTAL_) drop=PERCENT);
    weight SAMPLING_PANEL_WEIGHT13;
run;
proc print data=CTRL;
    title2 "Control Totals for Poststratification";
    title3;
run;
*-----
;
* Get BRR weights using SURVEYMEANS
;
*-----
;

proc surveymeans data=INDIV varmethod=brr (outweights=BRR(drop=_PSwt_)
fay=.3) noprint;
    var PUBID;

```

```

strata VSTRAT;
cluster VPUS;
weight SAMPLING_PANEL_WEIGHT13;
poststrata PSCELL/PSTOTAL=CTRL;
run;

*-----;
* Macro for integerizing weights ;
*-----;
%macro make_brr_wts(indata,cellvar,basewt,outdata);
*-----;
* Macro parameters: ;
* ;
* indata: Data set with base and replicate weights from SURVEYMEANS ;
* cellvar: Variable name used on the POSTSTRATA statement of the ;
* SURVEYMEANS procedure ;
* basewt: Variable name used on the WEIGHT statement of the ;
* SURVEYMEANS procedure ;
* outdata: Name of output data set containing integer weights. Can be ;
* either one- or two-level name. ;
*-----;

* Automatically determine the number of replicate weights;
proc contents data=&indata out=chk2(keep=NAME TYPE) noprint;
run;
proc summary data=chk2(where=(substr(NAME,1,6)='RepWt_'));
var TYPE;
output out=chk3 n=numwts;
run;
* Export replicate weight count to a macro variable;
data null ;
set chk3;
if (_n_=1) then call symput('numwts',compress(input(numwts,12.)));
run;
proc datasets nolist;
delete chk2 chk3;
run;
* Create variable to identify original sort order;
data &indata;
set &indata;
_orig_order=_n_;
run;
* Sort by poststratification cell variable;
proc sort data=&indata;
by &cellvar;
run;
* Cumulate-and-Round the base weights and all replicate weights;
data &indata;
set &indata;
retain _cum0 _cum1-_cum&numwts _rcum0 _rcum1-_rcum&numwts -1;
by &cellvar;
array _AA_ (%eval(&numwts+1)) &basewt RepWt_1-RepWt_&numwts;

```

```

array _BB_ (%eval(&numwts+1)) _cum0 _cum1-_cum&numwts;
array _CC_ (%eval(&numwts+1)) _rcum0 _rcum1-_rcum&numwts;
do _i = 1 to %eval(&numwts+1);
    if (first.&cellvar)
        then do;
            _BB_(_i)=0;
            _CC_(_i)=0;
        end;
    _BB_(_i)=_BB_(_i) + _AA_(_i);
    _AA_(_i)=round(_BB_(_i)) - _CC_(_i);
    _CC_(_i)=_CC_(_i)+_AA_(_i);
end;
drop _cum0 _cum1-_cum&numwts _rcum0 _rcum1-_rcum&numwts _i;
run;
* Restore original sort order;
proc sort data=&indata out=&outdata(drop=_orig_order_);
    by _orig_order_;
run;
* Check control totals for base weight, and first/last BRR weight;
proc freq data=&outdata noprint;
    tables PSCCELL/out=chk1(rename=(COUNT=_TOTAL_BASE_) drop=PERCENT);
    weight &basewt;
run;
proc freq data=&outdata noprint;
    tables PSCCELL/out=chk2(rename=(COUNT=_TOTAL_REPWT1_)
drop=PERCENT);
    weight RepWt_1;
run;
proc freq data=&outdata noprint;
    tables PSCCELL/out=chk3(rename=(COUNT=_TOTAL_REPWT&numwts._)
drop=PERCENT);
    weight RepWt_&numwts;
run;

data chk;
    merge CTRL chk1 chk2 chk3;
    by PSCCELL;
run;
proc print noobs data=chk;
    title2 "Poststratification Totals for Integerized Weights";
run;
* Cleanup;
proc datasets nolist;
    delete chk1 chk2 chk3;
run;
%mend c_and_r;

%make_brr_wts(BRR,PSCCELL,SAMPLING_PANEL_WEIGHT13,OUT.DBRR0304);

```

```
*117 strata so replicates is 120;
proc freq data = data;
table VSTRAT;
run;

data nh.BRR_alcohol;
    * Combine demographic info with dietary weights for people in the
    dietary dataset only;
    merge nh.alcohol out.dbrr0304;
    by PUBID;

run;
```

## Appendix D: BRR SAS Macro for estimating distribution standard errors

```
LIBNAME NH "C:\...\NH";
LIBNAME mylib "C:\...\mylib";

*-----;
* Create a dataset from the permanent dataset libname.demoadv ;
*-----;
data alcohol;
set nh.brr_alcohol_clean;

RepWt_0 = rndw1; /*rename primary survey weight*/

run;

*-----;
* Remove missing values in the replicate weight variables ;
*-----;

data alcoh;
  set alcohol;
  if RepWt_0 ne .;
  if RepWT_0 ne 0;
  if RepWt_120 ne .; /* keeps only those observations with dietary data */
run;

*-----;
* Create a dataset with one line per observation (two lines per ;
* respondent) ;
*-----;

/* append the datasets and create dummy variables */

proc sort; by seqn day; run;

data adult2;
Set alcoh;
if d_alcohol ne .;
if year = 1 then year2 = 0;
else if year = 2 then year2 = 1;

if d_day < 0 then d_day = 0;

if d_day = . then d_day_cat = .;
if d_day = 0 then d_day_cat = 0;
if 0 < d_day <=7 then d_day_cat = 1;
if 7 < d_day <=14 then d_day_cat = 2;
if 14 < d_day <=21 then d_day_cat = 3;
```

```

else if 21 < d_day then d_day_cat = 4;

if race_ethnicity =1 then RETH1 = 1;
else RETH1 = 0;
if race_ethnicity =2 then RETH2= 1;
else RETH2 = 0;
if race_ethnicity =3 then RETH3 = 1;
else RETH3 = 0;

m_alcohol = d_alcohol * d_day;

if (gender = 0) and (d_alcohol >= 6) then risky = 1;
if (gender = 0) and (d_alcohol < 6) then risky = 0;
if (gender = 1) and (d_alcohol >= 5) then risky = 1;
if (gender = 1) and (d_alcohol <5) then risky = 0;

run;

*-----;
* Sort the data by respondent and year ;
*-----;

proc sort; by PUBID year; run;

proc datasets library=work;
    delete alcoh ;
run;

*-----;
* Identify extreme values and delete if necessary ;
*-----;

data want;
    set adult2;

        by PUBID;
        if m_alcohol > 125 then deleteflag=1;
        else deleteflag = 0;
        retain deleteflag 0;

run;
proc sql;
    create table adult2 as
    select *
    from want (drop = deleteflag)
    where PUBID not in (select distinct PUBID from want where
deleteflag=1);
quit;

```



```

/*****/

*-----;
* Include the MIXTRAN and DISTRIB macro ;
*-----;

%include 'C:\...\MIXTRAN.sas';
%include 'C:\...\DISTRIB.sas';

title1 "Task 3 - Distribution of Alcohol";

*-----;
* Use the mixtran macro to fit the model and the distrib macro to ;
* estimate the distribution of consumption. ;
* This is done within a macro that runs the BRR runs also and calculates ;
* the BRR standard errors. ;
*-----;

%macro
BRR(data,response,foodtype,subject,repeat,covars_prob,covars_amt,outlib,pr
ed,param,modeltype,
seq,dist,weekend,vargroup,numvargroups,subgroup,start_val1,start_val2,star
t_val3,vcontrol, nloptions,titles,printlevel,
cutpts,ncutpts,nsim_mc,byvar,final, call_type );

/*call the MIXTRAN macro;*/
%MIXTRAN (data=&data, response=&response, foodtype=&foodtype,
subject=&subject, repeat=&repeat,
covars_prob=&covars_prob, covars_amt=&covars_amt, outlib=&outlib,
modeltype=&modeltype,replicate var= RepWt 0 /*put primary weight here*/,
seq=&seq, dist = &dist, lambda=1 /* required value for HNB or ZINB*/,
weekend=&weekend, vargroup=&vargroup, numvargroups=&numvargroups,
subgroup=&subgroup,start_val1=&start_val1, start_val2=&start_val2,
start_val3=&start_val3, vcontrol=&vcontrol,
nloptions=&nloptions, titles=&titles, printlevel=&printlevel) ;

/*call the DISTRIB macro;*/
%DISTRIB (call_type=&call_type,seed=5454768, nsim_mc=&nsim_mc,
modeltype=&modeltype, dist=&dist, pred=&pred, param=&param,
outlib=&outlib, cutpoints=&cutpts, ncutpnt=&ncutpts, byvar=&byvar,
subgroup=&subgroup,add_da=&data,subject=&subject, titles=&titles,
food=&foodtype, mcsimda=mylib. mcsim f m alcohol RepWt 0 /*input
manually*/, recamt=, recamt_co=, recamt_hi=,wkend_prop=3/7);

```

```

*the dataset &outlib..descript_&foodtype._RepWt_0 is created in the
distrib macro;
*mergeby is used later to merge data;
data dist;
set &outlib..descript_&foodtype._RepWt_0 /*input primary weight
manually*/;
mergeby=1;
keep &subgroup mergeby numsubjects mean_mc_t tpercentile1-tpercentile99
cutprobl-cutprob&&ncutpts. mergeby;
run;

*start BRR runs;
%do run= 1 %to 120 /*input number of replicates generated in Appendix C*/;

*turn off notes to save room;
options nonotes;
*label run number in log;
%put ~~~~~~ Run &run ~~~~~~;

*call MIXTRAN macro;
%MIXTRAN (data=&data, response=&response, foodtype=&foodtype,
subject=&subject, repeat=&repeat, covars_prob=&covars_prob,
covars amt=&covars amt, outlib=&outlib, modeltype=&modeltype,
replicate var=RepWt &run /*will run for each RepWt replicate generated
based on &run*/, seq=&seq, dist = &dist, lambda=1 /*preset for HNB or ZINB
model*/, weekend=&weekend, vargroup=&vargroup, numvargroups=&numvargroups,
subgroup=&subgroup, start_val1=&start_val1, start_val2=&start_val2,
start_val3=&start_val3, vcontrol=&vcontrol, nloptions=&nloptions,
titles=&titles, printlevel=&printlevel) ;

*call DISTRIB macro;
%DISTRIB (call_type=&call_type, seed=5454768, nsim_mc=&nsim_mc,
modeltype=&modeltype, dist=&dist, pred=&pred, param=&param,
outlib=&outlib, cutpoints=&cutpts, ncutpnt=&ncutpts, byvar=&byvar,
subgroup=&subgroup, add da=&data, subject=&subject, titles=&titles,
food=&foodtype, mcsimda=mylib. mcsim f m alcohol RepWt &run /*input
manually*/, recamt=, recamt_co=, recamt_hi=, wkend_prop=3/7);

*the dataset outlib.descript_&foodtype._RepWt_&run comes from the distrib
macro;
*renames the variables;
*mergeby is used later;
data distbrr;
set &outlib..descript_&foodtype._RepWt_&run /*input manually*/;
rename numsubjects=bnumsubjects mean_mc_t=bmean_mc_t tpercentile1-
tpercentile99=btpercentile1-btpercentile99
cutprobl-cutprob&&ncutpts.=bcutprobl-bcutprob&&ncutpts.;
run=&run;
mergeby=1;
data distbrr;

```

```

set distbrr;
keep &subgroup bnumsubjects bmean mc t btpercentile1-btpercentile99
bcutprob1-bcutprob&&ncutpts. mergeby;
run;

*appends brr datasets;
proc append base=brr_runs data=distbrr;
run;

*deletes distbrr dataset;
/*proc datasets nolist; delete distbrr; run;*/

%end; *end of brr runs;

*sort the dist and brr_runs datasets before merging;
proc sort data=dist; by &subgroup mergeby;
proc sort data=brr_runs; by &subgroup mergeby;

*merge dist and brr_runs and calculate the squared difference between the
brr estimate and the parameter from the first run;
data distall;
merge dist brr_runs; by &subgroup mergeby;
array bvar (*) bmean_mc_t btpercentile1-btpercentile99
bcutprob1-bcutprob&&ncutpts.;
array varo (*) mean_mc_t tpercentile1-tpercentile99
cutprob1-cutprob&&ncutpts.;
array dsqr (*) dbmean_mc_t dbtpercentile1-dbtpercentile99
dbcutprob1-dbcutprob&&ncutpts.;
do i=1 to dim(bvar);
dsqr[i]=(bvar[i]-varo[i])**2;
end;
run;

*compute the sums of the squares;
proc means data=distall sum noprint; by &subgroup mergeby;
var dbmean_mc_t dbtpercentile1-dbtpercentile99 dbcutprob1-
dbcutprob&&ncutpts.;
output out=sums sum= sum_dbmean_mc_t sum_dbtpercentile1-
sum_dbtpercentile99 sum_dbcutprob1-sum_dbcutprob&&ncutpts.;
run;

*calculate the standard errors;
data brr;
set sums;
array sumt (*) sum_dbmean_mc_t sum_dbtpercentile1-sum_dbtpercentile99
sum_dbcutprob1-sum_dbcutprob&&ncutpts.;
array se (*) mean_mc_t tpercentile1-tpercentile99 cutprob1-
cutprob&&ncutpts.;
do j=1 to dim(sumt);
se[j]=sqrt((sumt[j])/ (120 /*edit to equal number of replicate weights
generated in Appendix C*/ * .49));
end;

```

```

    keep mean_mc_t  tpercentile1-tpercentile99  cutprob1-cutprob&&ncutpts.
    &subgroup mergeby;
run;

*create the final dataset;
data toprint1;
    set dist;
    line=1;* These are the point estimates;
    keep &subgroup numsubjects mean_mc_t  tpercentile1-tpercentile99
    cutprob1-cutprob&&ncutpts. line;
run;

data toprint2;
    set brr;
    line=2;* These are standard errors, but rename them so they stack
under the point estimates;
    keep &subgroup mean mc t  tpercentile1-tpercentile99
    cutprob1-cutprob&&ncutpts. line;
run;

* Stack point ests, ses, and blank lines;
data &final;
set toprint1 toprint2;
run;

* Sort them appropriately;
proc sort data=&final;
    by &subgroup line;
run;

* Print statistics. Note negparen10.1 format statement to put parens on
the std errs;
proc print data=&final split=' ' noobs;
    var &subgroup line  mean_mc_t  tpercentile5 tpercentile10
tpercentile25 tpercentile50
    tpercentile75 tpercentile90 tpercentile95 cutprob1-cutprob&&ncutpts.;
/*    format line line.  mean_mc_t  tpercentile1-tpercentile99
negparen10.1
    cutprob1-cutprob&&ncutpts. negparen6.2 ;*/
    title 'Usual Intake of Alcohol';
    title2 'NLSY 2013-2015';
run;

%mend BRR;

*-----;
*-----;
*-----          End of the BRR macro          -----;
*-----;
*-----;

```

```
%BRR(data=adult2,response=m_alcohol,foodtype=m_alcohol,subject=PUBID,
dist=HNB, repeat=year2, covars prob=GENDER RETH1 RETH2 RETH3 risky,
covars_amt=GENDER RETH1 RETH2 RETH3 risky, outlib=mylib,
pred=mylib._pred_m_alcohol, param=mylib._param_m_alcohol, modeltype=corr,
seq=year2, weekend=,vargroup= , numvargroups= , subgroup= , start_val1= ,
start_val2= , start_val3= , vcontrol= , nloptions=qmax=300 tech=NMSIMP,
titles=4,printlevel=2,cutpts=0.5, ncutpts=1, nsim_mc=100, byvar= ,
final=mylib.final, call_type=Full);
```

## Appendix E: Simulation Study

### R CODE TO CREATE SIMULATED DATA (RIZOPOULOS, 2018)

```
setwd("~/Dissertation/Simulation/")
rm(list=ls())
library(MASS)
maindir <- "~/Dissertation/Simulation/"
datadir <- "simdata/"
data.output.file <- paste (maindir, datadir, "simdata_", sep="")

nsim <- 30 # number of simulated datasets
mySeeds <- round(runif(nsim,min=10,max=10000))
n <- 100 # number of subjects
K <- 1000 # number of measurements per subject (i.e. timepoints)
#t_max <- 5 # maximum follow-up time

for (myloop in 1:nsim) {

  set.seed(mySeeds[myloop])

  # we constuct a data frame with the design:
  # everyone has a baseline measurment, and then measurements
  # at random follow-up times
  DF <- data.frame(id = rep(seq_len(n), each = K),
    #time = c(replicate(n, c(0, sort(runif(K - 1, 0, t_max))))),
    time = c(1:K)#,
    # sex = rep(gl(2, n/2, labels = c("male", "female")), each = K)
  )

  # design matrices for the fixed and random effects non-zero part
  X <- model.matrix(~ #sex *
    time, data = DF)
  Z <- model.matrix(~ 1, data = DF)
  # design matrices for the fixed and random effects zero part
  X_zi <- model.matrix(~ #sex
    1, data = DF)
  Z_zi <- model.matrix(~ 1, data = DF)

  betas <- c(2, #0.05, 0.05,
```

```

-0.000005) # fixed effects coefficients non-zero part
shape <- 2
gammas <- c(-1.5, 0.5) # fixed effects coefficients zero part
D11 <- 0.5 # variance of random intercepts non-zero part
D22 <- 0.4 # variance of random intercepts zero part

# we simulate random effects
b <- cbind(rnorm(n, sd = sqrt(D11)), rnorm(n, sd = sqrt(D22)))
# linear predictor non-zero part
eta_y <- as.vector(X %*% betas + rowSums(Z * b[DF$id, 1, drop = FALSE]))
# linear predictor zero part
eta_zi <- as.vector(X_zi %*% gammas + rowSums(Z_zi * b[DF$id, 2, drop = FALSE]))
# we simulate negative binomial longitudinal data
DF$y <- rbinom(n * K, size = shape, mu = exp(eta_y))
# we set the extra zeros
DF$y[as.logical(rbinom(n * K, size = 1, prob = plogis(eta_zi)))] <- 0

data <- DF

mydata <- paste(data.output.file, myloop, ".csv", sep = "")
write.table(data, file = mydata, sep = ",", row.names = F, col.names = T)
}

lapply(data, function(y){ length(which(y==0))/length(y)})

```

## REFERENCES

- Anton, R. F., O'Malley, S. S., Ciraulo, D. A., Cisler, R. A., Couper, D., Donovan, D. M., . . . Zweben, A. (2006). Combined pharmacotherapies and behavioral interventions for alcohol dependence: the COMBINE study: a randomized controlled trial. *Jama*, 295(17), 2003-2017. doi:10.1001/jama.295.17.2003
- Barbosa, F. d. S., Sichieri, R., & Junger, W. L. (2013). Assessing usual dietary intake in complex sample design surveys: the National Dietary Survey. *Revista de Saúde Pública*, 47, 171s-176s.
- Becker, J. B., McClellan, M. L., & Reed, B. G. (2017). Sex differences, gender and addiction. *J Neurosci Res*, 95(1-2), 136-147. doi:10.1002/jnr.23963
- Buckman, D. W., Parsons, R., & Kahle, L. (2016). NCI Method Estimates of Usual Intake Distributions for Fish Consumption in Idaho.
- Butler, L., Poti, J. M., & Popkin, B. M. (2016). Trends in Energy Intake from Alcoholic Beverages among US Adults by Sociodemographic Characteristics, 1989-2012. *J Acad Nutr Diet*, 116(7), 1087-1100.e1086. doi:10.1016/j.jand.2016.03.008
- Cagnone, S., & Viroli, C. (2018). Multivariate latent variable transition models of longitudinal mixed data: an analysis on alcohol use disorder. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(5), 1399-1418. doi:10.1111/rssc.12285



Center for Behavioral Health Statistics and Quality. (2017) Results from the 2016 National Survey on Drug Use and Health: Detailed Tables. Rockville, MD: Substance Abuse and Mental Health Services Administration.

Center for Disease Control and Prevention. (2011) Continuous NHANES Web Tutorial: Variance Estimation: Variance Estimation in NHANES. (2011). Retrieved from <https://www.cdc.gov/nchs/tutorials/NHANES/SurveyDesign/Info1.htm>

Chong, E. K. P., & Zak, S. H. (2013). An Introduction to Optimization. Somerset, UNITED STATES: John Wiley & Sons, Incorporated.

Chou, N., & Steenhard, K. D. (2009). A Flexible Count Data Regression Model Using SAS®PROC NLMIXED. SAS Global Forum, Statistics and Data Analysis, 250.

DeSantis, S. M., & Bandyopadhyay, D. (2011). Hidden Markov models for zero-inflated Poisson counts with an application to substance use. *Stat Med*, 30(14), 1678-1694. doi:10.1002/sim.4207

Dodd, K. W., Guenther, P. M., Freedman, L. S., Subar, A. F., Kipnis, V., Midthune, D., . . . Krebs-Smith, S. M. (2006). Statistical methods for estimating usual intake of nutrients and foods: a review of the theory. *J Am Diet Assoc*, 106(10), 1640-1650. doi:10.1016/j.jada.2006.07.011

- Duncan, T. E., Duncan, S. C., & Hops, H. (1998). Latent variable modeling of longitudinal and multilevel alcohol use data. *J Stud Alcohol*, 59(4), 399-408.  
doi:10.15288/jsa.1998.59.399
- Farewell, V. T., Long, D. L., Tom, B. D. M., Yiu, S., & Su, L. (2017). Two-Part and Related Regression Models for Longitudinal Data. *Annual Review of Statistics and Its Application*, 4(1), 283-315. doi:10.1146/annurev-statistics-060116-054131
- Graubard, B. I., & Korn, E. L. (1999). Analyzing health surveys for cancer-related objectives. *J Natl Cancer Inst*, 91(12), 1005-1016. doi:10.1093/jnci/91.12.1005
- Guenther, P. M., Bowman, S. A., & Goldman, J. D. (2010). Alcoholic Beverage Consumption by Adults 21 Years and Over in the United States: Results From the National Health and Nutrition Examination Survey, 2003-2006: Technical Report. Retrieved from  
[https://www.cnpp.usda.gov/sites/default/files/dietary\\_guidelines\\_for\\_americans/AlcoholicBeveragesConsumption.pdf](https://www.cnpp.usda.gov/sites/default/files/dietary_guidelines_for_americans/AlcoholicBeveragesConsumption.pdf)
- Hughes, J. R., Fingar, J. R., Budney, A. J., Naud, S., Helzer, J. E., & Callas, P. W. (2014). Marijuana use and intoxication among daily users: an intensive longitudinal study. *Addict Behav*, 39(10), 1464-1470. doi:10.1016/j.addbeh.2014.05.024
- Korn, E. L., & Graubard, B. I. (2011). *Analysis of health surveys* (Vol. 323). John Wiley & Sons.

- Lau-Barraco, C., Braitman, A. L., Linden-Carmichael, A. N., & Stamates, A. L. (2016). Differences in weekday versus weekend drinking among nonstudent emerging adults. *Exp Clin Psychopharmacol*, 24(2), 100-109. doi:10.1037/pha0000068
- Laureano, G. H., Torman, V. B., Crispim, S. P., Dekkers, A. L., & Camey, S. A. (2016). Comparison of the ISU, NCI, MSM, and SPADE Methods for Estimating Usual Intake: A Simulation Study of Nutrients Consumed Daily. *Nutrients*, 8(3), 166. doi:10.3390/nu8030166
- Liu, L., Strawderman, R. L., Cowen, M. E., & Shih, Y. C. T. (2010). A flexible two-part random effects model for correlated medical costs. *Journal of Health Economic*, 29(1), 110--123. doi:10.1016/j.jhealeco.2009.11.010
- McCabe, S. E., Morales, M., Cranford, J. A., Delva, J., McPherson, M. D., & Boyd, C. J. (2007). Race/ethnicity and gender differences in drug use and abuse among college students. *J Ethn Subst Abuse*, 6(2), 75-95. doi:10.1300/J233v06n02\_06
- Min, Y., & Agresti, A. (2005). Random effect models for repeated measures of zero-inflated count data. *Statistical Modelling*, 5(1), 1-19. doi:10.1191/1471082X05st084oa
- Mooney, C. Z. (1997). Monte Carlo simulation. Thousand Oaks, CA, US: Sage Publications, Inc.

National Institute on Alcohol Abuse and Alcoholism. (n.d.-a). NHANES dietary web tutorial:

Modeling usual intake using dietary recall data: Task 4. Retrieved 2018-08-22, from

<https://www.cdc.gov/nchs/tutorials/dietary/Advanced/ModelUsualIntake/Info4.htm>

National Institute on Alcohol Abuse and Alcoholism. (n.d.-b). What is a standard drink? |

National Institute on Alcohol Abuse and Alcoholism (NIAAA). Retrieved 2018-06-

18, from [https://www.niaaa.nih.gov/alcohol-health/overview-alcohol-](https://www.niaaa.nih.gov/alcohol-health/overview-alcohol-consumption/what-standard-drink)

[consumption/what-standard-drink](https://www.niaaa.nih.gov/alcohol-health/overview-alcohol-consumption/what-standard-drink)

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. The

Computer Journal, 7(4), 308-313. doi:10.1093/comjnl/7.4.308

NHANES Web Tutorial Frequently Asked Questions (FAQs). (2014). Retrieved 2018-08-24,

from <https://www.cdc.gov/nchs/tutorials/NHANES/FAQs.htm>

Nusser, S. M., Carriquiry, A. L., Dodd, K. W., & Fuller, W. A. (1996). A Semiparametric

Transformation Approach to Estimating Usual Daily Intake Distributions. Journal of the American Statistical Association, 91(436), 1440-1449.

doi:10.1080/01621459.1996.10476712

Nusser, S. M., Carriquiry, A. L., Jensen, H. H., & Fuller, W. A. (1990). A Transformation

Approach to Estimating Usual Intake Distributions. CARD Working Papers, 95.

Paben, S. P. (1999). Comparison of Variance Estimation Methods for the National

Compensation Survey. Washington, DC: U.S. BUREAU OF LABOR STATISTICS.

- Rizopoulos, D. (2018). Zero-Inflated Poisson and Negative Binomial Models with GLMMadaptive. Retrieved from <https://www.r-bloggers.com/zero-inflated-poisson-and-negative-binomial-models-with-glmmadaptive/>
- Rotgers, F. (1997). Assessing Alcohol Problems: A Guide for Clinicians and Researchers. *J Stud Alcohol*, 58(1), 106.
- SAS Institute Inc. (2008). SAS/STAT® 9.2 User's Guide. Cary, NC: SAS Institute Inc. Retrieved 2019-10-29, from <https://support.sas.com/documentation/cdl/en/statuggenmod/61787/PDF/default/statuggenmod.pdf>
- SAS Institute Inc. (2010). SAS/STAT® 9.2 User's Guide. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. (2015). SAS/STAT® 14.1 User's Guide. Cary, NC: SAS Institute Inc. Retrieved 2019-10-29, from <https://support.sas.com/documentation/onlinedoc/stat/141/nlmixed.pdf>
- Shirley, K. E., Small, D. S., Lynch, K. G., Maisto, S. A., & Oslin, D. W. (2010). Hidden Markov models for alcoholism treatment trial data. *Ann. Appl. Stat.*, 4(1), 366-395. doi:10.1214/09-AOAS282
- Sobell, L. C., & Sobell, M. B. (2003). Assessing Alcohol Problems: Assessment of Drinking Behavior (2<sup>nd</sup> ed.) (No. 03-3745). Bethesda, Maryland: National Institute of Health.

Souverein, O. W., Dekkers, A. L., Geelen, A., Haubrock, J., de Vries, J. H., Ocke, M. C., . . .  
van 't Veer, P. (2011). Comparing four methods to estimate usual intake distributions.

Eur J Clin Nutr, 65 Suppl 1, S92-101. doi:10.1038/ejcn.2011.93

Substance Abuse and Mental Health Services Administration, Center for Behavioral Health  
Statistics and Quality. (2016). Treatment Episode Data Set (TEDS): 2004-2014  
(Tech. Rep.). Rockville, MD: Substance Abuse and Mental Health Services  
Administration.

Tooze, J. A., Kipnis, V., Buckman, D. W., Carroll, R. J., Freedman, L. S., Guenther, P. M., . . .

. Dodd, K. W. (2010). A mixed-effects model approach for estimating the distribution  
of usual intake of nutrients: the NCI method. Stat Med, 29(27), 2857-2868.

doi:10.1002/sim.4063

Usual Dietary Intakes: SAS Macros for NCI Method. (2018). Retrieved from

<https://epi.grants.cancer.gov/diet/usualintakes/macros.html>

Witbrodt, J., Mulia, N., Zeng, S. E., & Kerr, W. C. (2014). Racial/ethnic disparities in  
alcohol-related problems: differences by gender and level of heavy drinking.

Alcoholism, clinical and experimental research, 38(6), 1662-1670.

doi:10.1111/acer.12398

Wolter, K. M. (1985). Introduction to variance estimation. New York: Springer-Verlag.