Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Emily Mitchell Date

Regression Models for a Continuous Outcome Subject to Pooling

By

Emily Mitchell
Doctor of Philosophy

Biostatistics

Robert H. Lyles, Ph.D.
Advisor

Qi Long, Ph.D.
Committee Member

Amita Manatunga, Ph.D.
Committee Member

Enrique Schisterman, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Regression Models for a Continuous Outcome Subject to Pooling

By

Emily Mitchell
B.A., University of South Carolina, 2008
M.S., Emory University, 2012

Advisor: Robert H. Lyles, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Biostatistics
2013

Abstract

Regression Models for a Continuous Outcome Subject to Pooling

By Emily Mitchell

The potential for research involving biospecimens can be hindered by the high cost of laboratory assays. To reduce cost, strategies such as randomly selecting a portion of specimens for analysis or randomly pooling specimens prior to performing laboratory assays may be employed, yet are often accompanied by a considerable loss of statistical efficiency. Intuitively, forming pools from specimens with similar covariate values will help maintain high precision levels among regression coefficient estimates by preserving the relationship between the outcome and predictor variables. To implement this strategy, we propose a novel pooling method based on the k-means clustering algorithm. This method is tested in a linear regression setting, then applied in subsequent studies to promote efficiency.

Linear regression provides a convenient avenue to test potential efficiency gains from k-means pooling. Many biomarkers measured in epidemiological studies, however, exhibit a positive, right-skewed distribution, for which linear regression may not be appropriate. Regression models suitable to this type of outcome data are explored, including a modification of multiple linear regression on a log-transformation of pool-wise data and a novel parameterization of the gamma distribution.

If pools are formed from specimens with identical covariate values, regression analyses on a right-skewed, pooled outcome are greatly simplified. When these x-homogeneous pools cannot be formed, we propose a quasi-likelihood model for pooled specimens as well as a Monte Carlo Expectation Maximization (MCEM) algorithm. We then develop an extension of Akaike's Information Criterion to help select the best model. Simulations demonstrate that these analytical methods provide essentially unbiased estimates of coefficient parameters as well as their standard errors when appropriate assumptions are met.

In conclusion, when the number of laboratory tests is limited by budget, pooling specimens prior to performing lab assays can be an effective way to save money. High levels of precision can be maintained by exploiting covariate information to form pools, as in k-means pooling, then selecting the best-fitting model using an AIC-type criterion. When pools are formed strategically and analyzed under the appropriate models, pooling can considerably reduce costs with minimal information loss.

Regression Models for a Continuous Outcome Subject to Pooling

By

Emily Mitchell
B.A., University of South Carolina, 2008
M.S., Emory University, 2012

Advisor: Robert H. Lyles, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Biostatistics
2013

Acknowledgements

To my advisor, Bob Lyles, for his constant intellectual support and contagious confidence.

To my mom, Peggy, for raising me with the freedom and courage to build my own story.

And to my dad, Toby, without whose memory and inspiration I might never have chosen to follow in his footsteps.

Contents

List of Figures

List of Tables

Chapter 1

Background

1.1 Origins and Applications of Pooling

The introduction, or rather, popularization of pooling is often attributed to Robert Dorfman (1943), who applied the technique to blood samples when testing soldiers for syphilis. The idea behind this type of pooling is that, when a disease has low prevalence, it is more cost effective to test pooled groups of specimens, then retest each of the specimens from any positive pools individually, rather than to simply perform lab tests on each individual specimen. Since then, pooling has become a popular strategy for reducing cost, for instance, in testing donated blood for HIV or determining regional prevalence of blood-bourne diseases (Brookmeyer, 1999; Emmanuel et al., 1988; Lan, Hsieh, and Yen, 1993; Vansteelandt, Goetghebeur, and Verstraeten, 2000).

Weinberg and Umbach (1999) note that pooling can help preserve irreplaceable specimens, by requiring only a portion of the stored sample for analysis. Along the same lines, pooling can make use of samples that may lack enough volume to be analyzed individually, thus extracting information from specimens that may otherwise have been presumed unusable.

Pooling can also be helpful from the standpoint of reducing the number of assay nondetects when a laboratory limit of detection is present. Schisterman and Vexler (2008) discuss the advantages to pooling when estimating mean and variance of biospecimens subject to a limit of detection, where the utility of pooling depends on the value of the

detection threshold relative to the mean of the biomarker value.

While pooling may be performed for various reasons, in this paper we assume that the primary motivating factor is to reduce laboratory costs. Study designs recommended in this paper are made under this assumption, and additional consideration of alternate designs may be necessary if pooling is done for a different reason, such as to reduce the number of non-detects when a limit of detection is present.

Specifically, we focus our investigation on efficiently estimating regression coefficients when a continuous outcome is subject to pooling. Let Y_{ij} denote the j^{th} subject in the i^{th} pool, and let $\mathbf{x}_{ij}$ be the vector of covariates corresponding to this outcome. Furthermore, suppose that we can model the relationship between the predictor and outcome with one of the following models:

1. $h(Y_{ij}) = \alpha + \mathbf{x}_{ij}\boldsymbol{\beta} + \epsilon_{ij}$

2. $g(\mu_{ij}) = \alpha + \mathbf{x}_{ij}\boldsymbol{\beta}$

The first model is a linear regression model where h denotes a transformation on Y_{ij} (which could be the identity) and ϵ_{ij} is the error term such that $E(\epsilon_{ij}) = 0$ and $Var(\epsilon_{ij}) = \sigma^2$ for all $j = 1, \ldots, k_i$, $i = 1, \ldots, n$. The second model is a generalized linear regression model, where the link function g represents a transformation of $\mu_{ij} = E(Y_{ij})$, the mean of Y_{ij}. Our main objective in this study is to effectively and efficiently estimate the vector of regression coefficients $\boldsymbol{\beta}$, when the outcome is only known by its pooled measurements. In doing so, we propose and evaluate regression models for pooled data, as well as pooling designs that promote estimate efficiency.

1.2 Efficient Pooling Designs

A common aversion to pooling is the fear of losing information as a consequence of reducing the total sample size. For this reason, many efforts have been made to identify efficient pooling designs, in order to maintain a high level of statistical precision while reducing costs. The power of pooling as a potential cost-saving tool is particularly compelling when strategically pooled samples outperform analyses on the same number of randomly sampled

individual specimens.

Vexler, Liu, and Schisterman (2006) demonstrate the advantage of pooling under a two-stage sampling design when a limit of detection is present. They consider this scenario and discuss methods to optimize efficiency when a biomarker is distributed under a normal or gamma distribution. Schisterman et al. (2010) also consider optimal pooling strategies when estimating marginal means and variances under normally-distributed data, paying special attention to proper estimation of pooling and measurement error. They recommend a hybrid pooled-unpooled design, where the unpooled specimens permit effective estimation of these error components. Malinovsky, Albert, and Schisterman (2012) extend this hybrid design to a Gaussian random effects model, with an emphasis on using strategic pooling designs to efficiently estimate the intraclass correlation coefficient.

In a linear regression setting, an optimal pooling strategy for estimate precision is based on a D-optimal design, which seeks to maximize $|\mathbf{X}^T\mathbf{X}|$, where $\mathbf{X}$ is the design matrix. Ma et al. (2011) propose the use of this D-optimal design using information on a known or inexpensively-assessed biomarker to optimize pools on a correlated biomarker with an expensive laboratory assay.

1.2.1 Clustering

In Chapter 2, we propose applying a k-means clustering algorithm as an efficient pooling design and use simulations to assess this strategy under linear regression. In later sections, we apply this clustering method to generalized regression models to promote estimate precision.

A wide variety of clustering algorithms are available, the growth of which has been promoted by increasingly powerful computing capacity. Hartigan (1975) gives a nice introduction to various clustering procedures, although the methods he discusses may now be somewhat outdated. For a more recent summary, Jain, Murty, and Flynn (1999) and Abbas (2008) provide an overview of clustering techniques, the former containing some helpful illustrations comparing the various procedures.

While the objective of most clustering strategies is for classification purposes, our use of clustering, specifically the k-means clustering algorithm, has a slightly different motivation.

Instead of correct identification of membership to underlying groups, our goal is to exploit existing clustering techniques that exhibit the secondary effect of producing efficient estimates in a regression setting. The k-means clustering algorithm is particularly suited to our purposes since it seeks to maximize the between-cluster sum of squares, which corresponds directly with minimizing the variance of the regression coefficient estimate in simple linear regression (SLR). This connection to SLR will be discussed in more detail in Chapter 2. For now, we will focus on implementation and optimization of this algorithm.

Both SAS and R have built-in functions that perform k-means clustering. Both functions accept arguments specifying the data to be clustered (e.g. the fully-known set of covariate values), and the desired number of clusters, k. If our goal were for classification purposes, it would be beneficial to identify an optimal value for k. For our purposes, however, we can choose k to be the maximum allowable number of lab tests commensurate with available funds.

The *kmeans* function in R seeks to minimize the within-cluster sum of squares. By default, this function applies the k-means algorithm described in Hartigan and Wong (1979). Due to the complexity of the clustering algorithm, a local minimum is identified. Thus, it may be desirable to run the algorithm multiple times and choose the clustering that is optimal based on a predetermined criterion (e.g. maximizing $|\mathbf{X}^T\mathbf{X}|$) in order to improve efficiency.

Efforts have been made to provide a more globally optimal k-means algorithm. One strategy is to apply a leader algorithm that forces a minimum distance between initial cluster centers. Doing this creates separation between clusters at the start of the algorithm, especially when compared with randomly choosing initial cluster seeds (the default in R's *kmeans* function). The FASTCLUS procedure in SAS applies Hartigan's leader algorithm (Hartigan, 1975), and performs the k-means algorithm detailed by MacQueen (1967).

Another attempt at improving the k-means algorithm is to use a stepwise approach. Likas, Vlassis, and Verbeek (2003) propose this global version, which, instead of randomly selecting k initial cluster centers, "proceeds in an incremental way attempting to optimally add one new cluster center at each stage". To do this, Likas et al. recommend beginning by performing k-means with $k = 1$. Then, the resulting cluster center is combined with each of

the N observations as a candidate pair of initial seeds for $k = 2$. The pair that maximizes the algorithm is then chosen for the next iteration, and again combined and tested with each of the N observations at $k = 3$. This process is repeated until the desired number of clusters has been achieved.

1.3 Pooling in Logistic Regression

Much of the research concerning pooling has focused on the logistic regression setting, perhaps as a natural extension of pooling's origins in identifying disease presence or absence. In the following sections, we briefly summarize regression models for logistic regression; first, when pooling is performed on a binary outcome, then when the binary outcome is known for each individual, and pooling is performed on an exposure of interest.

1.3.1 When a Binary Outcome is Pooled

Consider the scenario explored by Vansteelandt et al. (2000), where logistic regression is performed on a binary outcome that is subject to pooling. In their paper, Vansteelandt et al. define a case pool as a pool that tests positive, indicating that at least one of the specimens in that pool is a case. A control pool is then a pool that tests negative, meaning that all specimens comprising that pool are controls. They propose direct maximization of the observed likelihood, where the log-likelihood of the pooled measurements is:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} Y_i^p \log f(k_i, \mathbf{x}_i) + (1 - Y_i^p) \log[1 - f(k_i, \mathbf{x}_i)] \tag{1.1}$$

where k_i represents the pool size, $\mathbf{x}_i$ and Y_i^p the covariate vector and measured outcome, respectively, for pool i, and $f(k_i, \mathbf{x}_i) = Pr(Y_i^p = 1|\mathbf{x}_i)$. Depending on the desired regression model, $f(k_i, \mathbf{x}_i)$ could have various forms. In logistic regression, for instance,

$$Pr(Y_i^p = 0|\mathbf{x}_i) = \prod_{j=1}^{k_i} Pr(Y_{ij} = 0|\mathbf{x}_{ij}) = \prod_{j=1}^{k_i} [1 + \exp(\mathbf{x}_{ij}\boldsymbol{\beta})]^{-1}$$

since the probability that pool i is a control pool is equal to the product of the probabilities that each of the specimens is a control, assuming that all specimens are independent. Then

$$Pr(Y_i^p = 1|\mathbf{x}_i) = 1 - Pr(Y_i^p = 0|\mathbf{x}_i) = 1 - \prod_{j=1}^{k_i}[1 + \exp(\mathbf{x}_{ij}\boldsymbol{\beta})]^{-1}.$$

Vansteelandt et al. also consider the effect of test sensitivity and specificity on the resulting estimates, but this topic is beyond the scope of this paper, so for our purposes, we assume a sensitivity and specificity of 1 for all tests. Vansteelandt et al. demonstrate that optimal pooling strategies can reduce cost up to 44% with virtually no precision loss in the calculation of disease prevalence, when a binary outcome is subject to pooling. Specifically, they recommend an $\mathbf{x}$-homogeneous pooling design to maximize precision. When $\mathbf{x}$-homogeneous pools cannot be formed, we recommend k-means clustering as a means to preserve the relationship between the outcome and covariates. In Section 2.6 we use simulations to demonstrate the benefit of k-means clustering in this logistic regression setting.

1.3.2 When an Exposure is Pooled

Weinberg and Umbach (1999) also consider pooling in a logistic regression setting, but focus on the situation when each individual's case status is known, and pooling is performed on a continuous exposure. To improve estimate precision, they recommend pooling conditional on case status. Using a known outcome to inform pools may induce bias in the resulting estimates if appropriate measures are not taken.

Let $\mathbf{E}$ represent the continuous exposure that is subject to pooling, and let $S = \sum_{j=1}^{g} E_j$ denote the sum of the exposures in a pool (i.e. pool size $\times$ measured value of pool), where g is the pool size. To develop appropriate, consistent regression estimates, Weinberg and Umbach take advantage of the multiplicative structure of the risk model in order to propose a set based logistic regression model:

$$\frac{Pr(\text{case set}|S)}{Pr(\text{control set}|S)} = \exp(\mu^* g + \beta S + \log r_g),$$

where β is the regression coefficient of interest and $\log r_g$ represents "the number of case

sets of size g divided by the number of control sets of size g" (Weinberg and Umbach, 1999). Note that this last component necessitates a bit more attention to pooling design, since it forces at least one case pool of size g if there are any control pools of size g, and vice versa. This model permits valid estimation of the coefficient of interest, β, using only pooled values, and suffers a surprisingly small loss of efficiency relative to the individual-level logistic model. The model is also flexible in accommodating additional covariates in the same manner, as well as interaction terms involving the exposure.

Zhang and Albert (2011), Zhang et al. (2012), and Lyles et al. (2012) also consider pooling on an exposure in a logistic regression setting. Zhang and Albert (2011) apply a regression calibration approach when a continuous exposure is pooled, while Zhang et al. (2012) and Lyles et al. (2012) develop maximum likelihood methods to estimate regression coefficients when a pooled exposure is binary.

1.4 Pooling on a Right-Skewed, Continuous Variable

Pooled measurements are often assumed to represent the arithmetic mean of the individuals comprising that pool. This property facilitates analysis of pooled data under certain distributions, such as Gaussian or gamma, due to summation properties of these distributions, so that pools retain the assumed distribution of the individual specimens. For distributions that do not share this summation property, such as the lognormal, alternate methods must be taken to analyze pooled values.

1.4.1 Convolution

A sum of random variables can be characterized exactly by a convolution formula. Let $Y_1, \ldots, Y_n$ be independent random variables and let $f_i(y_i)$ denote the density of Y_i for $i = 1, \ldots, n$. Then for $S = \sum_{i=1}^{n} Y_i$, the density of S can be written as the convolution:

$$f_S(S) = \int_{Y_2} \cdots \int_{Y_n} \left[f_1 \left(S - \sum_{i=2}^{n} Y_i \right) f_2(Y_2) \ldots f_n(Y_n) \right] dY_2 \ldots dY_n. \tag{1.2}$$

Vexler, Liu, and Schisterman (2010) apply a deconvolution method to estimate empirical characteristic functions of a sum of random variables without applying any distributional assumptions on the individual specimens. When pool size is small (e.g. 2 or 3), (1.2) can be evaluated for each pool using numerical integration. These values can then be inserted into an observed likelihood function to obtain MLEs of the desired regression coefficients $\boldsymbol{\beta}$. When pool size exceeds 3, however, this technique tends to become computationally intractable, and initial efforts even for pools with only two specimens revealed some convergence issues.

1.4.2 Approximating the Density of a Sum of Random Variables

A natural inclination when approximating the sum of random variables may be to apply the Central Limit Theorem (CLT), which states that, if $Y_1, \ldots, Y_n$ are independent and identically-distributed with mean μ and finite variance σ^2, then

$$\frac{1}{\sqrt{n}\sigma}\left(\sum_{i=1}^{n} Y_i - n\mu\right) \to N(0,1)$$

as $n \to \infty$ (Bain and Engelhardt, 1992). Caudill (2010, 2011) has dedicated several papers to producing estimates of the mean of pooled, lognormally distributed data based on an extension of the CLT. He explores a moment matching technique with bias-correction methods based on characteristics of the lognormal distribution, as well as an application of the CLT for larger pool sizes (Caudill, Turner, and Patterson, 2007; Caudill, 2010, 2011). However, since the CLT requires large pool sizes in order to accurately approximate the distribution of a sum (or mean) of right-skewed random variables, we seek other methods that can also accommodate moderate to small pool sizes.

The field of engineering has produced an abundance of literature concerning the approximation of the density of a sum of lognormal random variables. In electrical engineering, the sum of lognormal variables is often used to characterize applications in wireless communications, such as co-channel interference and large-scale signal fading (Beaulieu, Abu-Dayya, and McLane, 1995; Beaulieu and Xie, 2004; Santos Filho, Yacoub, and Cardieri, 2006; Li, 2007; Li et al., 2011; Liu et al., 2007; Szyszkowicz and Yanikomeroglu, 2009; Tellambura

and Senaratne, 2010; Zhang and Song, 2006). Initially, we considered these methods as a potential solution to the problem of estimating regression coefficients for a pooled, lognormal outcome. In particular, we focused on the modified-power-lognormal (MPLN) function proposed by Szyszkowicz and Yanikomeroglu (2009). Let $f(y_i; \mu_i, \sigma_i)$ denote the lognormal density, such that

$$f(y_i; \mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi} y_i \sigma_i} e^{-\frac{1}{2}\left(\frac{\ln y_i - \mu_i}{\sigma_i}\right)^2},$$

and let $X = \sum_{i=1}^{n} Y_i$ denote the sum of n lognormally distributed random variables. Szyszkowicz and Yanikomeroglu propose that the distribution of X can be approximated by the MPLN function:

$$f_{MPLN}(x) = \frac{t}{\sqrt{2\pi} x s} e^{-\frac{1}{2}\left(\frac{\ln x - m}{s}\right)^2} \Phi^{t-1}\left(\frac{\ln x - m}{s}\right),$$

where m, s, and t are functions of σ_i and μ_i, $i = 1, \ldots, n$. While they demonstrate the success of this proposed function to approximate the density of a sum of lognormal variables, we were unable to effectively apply this strategy to estimate the regression coefficients of interest.

1.4.3 Monte Carlo Expectation Maximization (MCEM) Algorithm

Since the sum of lognormal random variables is not so easily approximated in a manner that also permits proper estimation of the regression coefficients of interest, we turn instead to missing data mechanisms to calculate MLEs.

The Expectation Maximization algorithm was popularized by Dempster, Laird, and Rubin (1977). The algorithm maximizes the observed data log-likelihood by exploiting the more convenient structure of the complete data log-likelihood. This concept works well with pooled data, since the mean of each group of specimens is observed, while the individual measurements are the unknown (i.e., missing) data.

The EM algorithm is composed of two steps. In the Expectation (E) Step, the conditional expectation is evaluated at the current iteration of the parameter estimates. Let $\mathbf{Y}^p = (Y_1^p, \ldots, Y_n^p)$ denote the vector of observed, pooled outcomes, and let Y_{ij} denote the

value of the unknown, individual outcome for subject j in pool i. Then the E step evaluates:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = E\left[\log L_c(\boldsymbol{\theta})|\mathbf{Y}^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}\right] = \sum_{i=1}^{n} E\left[\sum_{j=1}^{k_i} \log f(Y_{ij}|\mathbf{X}, \boldsymbol{\theta})|Y_i^p, \boldsymbol{\theta}^{(t)}\right] \tag{1.3}$$

in terms of the parameter vector $\boldsymbol{\theta}$, where L_c denotes the complete likelihood, $\mathbf{X}$ is the fully-known individual-level covariate data, and $\boldsymbol{\theta}^{(t)}$ is the estimate of the parameter vector at the t^{th} (i.e. current) iteration.

The Maximization (M) step then maximizes (1.3) with respect to $\boldsymbol{\theta}$ to get a new estimate, $\boldsymbol{\theta}^{(t+1)}$. This step is often straightforward, particularly when the Y_{ij}'s are assumed to follow a distribution from the exponential family. The E step, on the other hand, can be quite difficult to evaluate, such as when a lognormal distribution is assumed. Due to this complexity, we apply Monte Carlo methods, which are founded on the Law of Large Numbers, to approximate (1.3). For any function of the complete data h,

$$E\left[h(\mathbf{Y}_i)|Y_i^p, \boldsymbol{\theta}^{(t)}\right] \approx \frac{1}{M}\sum_{m=1}^{M} h(\mathbf{Y}_{i,m}),$$

where $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{ik_i})$, and $\mathbf{Y}_{i,m} = (Y_{i1,m}, \ldots, Y_{ik_i,m})$ is generated from the conditional distribution $g(\mathbf{Y}_i|Y_i^p, \boldsymbol{\theta}^{(t)})$. The Monte Carlo size M is chosen to be large enough so that the properties of the Law of Large Numbers holds. Several papers provide good descriptions of the MCEM algorithm (Levine and Casella, 2001; Wei and Tanner, 1990; Booth and Hobert, 1999). In particular, Levine and Casella (2001) propose a dynamic updating formula for determining the optimal Monte Carlo size based on the proximity of the estimated parameters to the MLEs at each iteration. Although this computationally streamlined strategy was deemed unnecessary for the simulations in this paper, it may prove quite helpful for more complex situations when the conservation of computing time is imperative.

In some situations, $g(Y_{ij}|Y_i^p, \boldsymbol{\theta}^{(t)})$ may not be a known distribution or have closed form. In such cases, generating Monte Carlo samples from this conditional distribution may require additional techniques. In this paper, we consider rejection sampling and importance sampling to overcome this obstacle (Levine and Casella, 2001). Additional details concerning the application of these sampling methods and the MCEM algorithm to pooled,

right-skewed data are outlined in Section 3.7.

One of the disadvantages of the EM algorithm is that standard errors (SE) are not directly produced. Several methods have been proposed to calculate these SEs, some of which can be found in Jamshidian and Jennrich (2000), Oakes (1999), and Louis (1982). For this study, we found Louis's method to be theoretically defensible, compatible with MC methods, and successful in practice. Thus, we apply an extension of this method to calculate SE's when the MCEM algorithm is employed, details of which can be found in Section 3.7.4.

1.5 Quasi-Likelihood

When dealing with skewed data, whether pooled or individual, it may be helpful to fit a quasi-likelihood (QL) model. The concept of quasi-likelihood was introduced by Wedderburn (1974) as an alternative to maximum likelihood estimation when the underlying distribution is unknown. Quasi-likelihood requires specification of only the first and second moments; instead of maximizing a fully-specified log-likelihood, the quasi-likelihood is maximized, where the contribution of observation z_i is:

$$K(z_i, \mu_i) = \int_{z_i}^{\mu_i} \frac{z_i - t}{\phi V(t)} dt + c(z_i),$$

where $\mu_i = E(z_i; \boldsymbol{\beta})$ is a known function of some parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$, often assumed to be the coefficients in a regression setting. $V(\bullet)$ is a known function of the mean, ϕ is a dispersion parameter, and $c(\bullet)$ is some function of z_i not dependent on μ_i. Wedderburn argues that K has properties similar to a log-likelihood, namely:

1. $E\left(\frac{dK}{d\mu_i}\right) = E\left(\frac{dK}{d\beta_j}\right) = 0$
2. $E\left(\frac{dK}{d\beta_j}\frac{dK}{d\beta_{j'}}\right) = -E\left(\frac{d^2K}{d\beta_j d\beta_{j'}}\right) = \frac{1}{V(\mu)}\frac{d\mu_i}{d\beta_j}\frac{d\mu_i}{d\beta_{j'}}$

Maximum quasi-likelihood estimates are then found by solving the estimating equations with respect to $\boldsymbol{\beta}$:

$$\sum_{i=1}^{n} \frac{dK(z_i, \mu_i)}{d\boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{z_i - \mu_i}{\phi V(\mu_i)} \frac{d\mu_i}{d\boldsymbol{\beta}} = 0$$

The inclusion of the dispersion parameter ϕ in the QL formulation permits specification of the mean-variance relationship only up to a constant. Note that inclusion of this parameter does not affect estimation of the quasi-likelihood estimates $\hat{\boldsymbol{\beta}}_{QL}$. Wedderburn (1974) recommends estimating ϕ with the method of moments:

$$\tilde{\phi} = \frac{1}{n-p} \sum_{i=1}^{n} \frac{(z_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}$$

where $\hat{\mu}_i$ is evaluated at $\hat{\boldsymbol{\beta}}_{QL}$. Furthermore, the standard errors of $\hat{\boldsymbol{\beta}}_{QL}$ can be approximated by:

$$Var(\hat{\boldsymbol{\beta}}_{QL}) \approx \left[E\left(\sum_{i=1}^{n} \frac{d^2 K_i}{d\boldsymbol{\beta} d\boldsymbol{\beta}^T} \right) \right]^{-1},$$

evaluated at $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}_{QL}$. Additional details and examples on the use of the quasi-likelihood method can be found in Heyde (1997), Huber (1964), McCullagh (1983), and Wedderburn (1974). In Chapter 4, we apply QL models as an alternative to maximum likelihood methods when performing regression on a pooled outcome. These methods are closely associated with gamma regression models and provide a convenient outlet for analyzing $\mathbf{x}$-heterogeneous pools.

Chapter 2

Linear Regression on a Pooled Outcome

2.1 Introduction

In this chapter, we explore some of the challenges and concerns that arise when pooling is considered, specifically when pooling is used to assess a continuous outcome variable that is to be modeled via linear regression. First, we consider the statistical theory underlying several scenarios that a researcher might encounter when working with data from pooled specimens. We then use this theory to determine efficient strategies for assigning pools, with an emphasis on maintaining high precision levels when estimating regression coefficients, while saving resources by reducing the required number of lab assays. Specifically, we propose a novel pooling strategy based on the k-means clustering algorithm as a means to reduce laboratory costs while maintaining a high level of statistical efficiency when predictor variables are measured on all subjects, but the outcome of interest is assessed in pools. We perform simulation studies to compare k-means pooling with existing pooling and selection strategies under simple and multiple linear regression models.

The linear regression scenario is particularly instructive, as it permits both a natural framework for analysis (via weighted least squares) and a clear roadmap for efficient pooling design considerations. Simulation results suggest that while all of the pooling methods

considered maintain unbiased estimates and appropriate confidence interval coverage of the coefficient parameters, pooling under k-means clustering provides the most precise estimates, closely approximating the results from the full data and losing minimal precision as the total number of pools decreases. We then apply these methods to a regression analysis of 2005 – 2007 data on HDL cholesterol, serum estradiol, and other variables examined in the BioCycle Study (Schliep et al., 2012). In conclusion, when the number of lab tests is limited by budget, pooling specimens based on k-means clustering prior to performing lab assays can be an effective way to save money with minimal information loss in a regression setting.

2.2 Regression Formulation

For this study, we assume that the number of feasible lab assays (n) is limited by budget, so that physically combining individual biological specimens into pools is an attractive option. We first discuss the statistical theory underlying several scenarios that may be encountered when working with data from pooled specimens.

2.2.1 Equal Aliquot Volumes

Consider the MLR model:

$$Y_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \epsilon_{ij},\ j = 1 \dots k_i,\ i = 1 \dots n,$$

where Y_{ij} is the outcome and $\mathbf{x}_{ij} = (1, x_{ij1}, \dots, x_{ijP})$ the row vector of covariates for the j^{th} subject in the i^{th} pool, $\boldsymbol{\beta}$ is the column vector of coefficients, k_i is the number of specimens in pool i (i.e. pool size), and ϵ_{ij} is the error term with mean 0 and variance σ^2. Furthermore, let $N = \sum_{i=1}^{n} k_i$ denote the total sample size.

In practice, each specimen might have a different volume, depending on the amount initially collected, or the remaining volume after portions were taken for use in other studies. These differing aliquot volumes form a pooled measurement that is a weighted average of the value of each specimen included in the pool, requiring a slight variation in the regression

formulation. We will first describe the simpler situation, when all aliquot volumes are assumed to be equal. Except where otherwise noted, we assume that the same aliquot volume is contributed by each member of a pool for the remainder of this study.

When all specimens contribute equal aliquot volumes to the pool, we assume that the lab assay applied to a pooled sample yields the mean concentration from the individual specimens comprising the pool. The MLR model for the i^{th} pool then becomes:

$$Y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i,\ i = 1 \ldots n,$$

where $Y_i = \frac{1}{k_i}\sum_{j=1}^{k_i} Y_{ij}$ is the measured value of the i^{th} pool, and $\epsilon_i = \frac{1}{k_i}\sum_{j=1}^{k_i} \epsilon_{ij}$ is the error term for pool i, such that $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2/k_i$. $\mathbf{x}_i = (1, \bar{x}_{i\bullet 1}, \ldots, \bar{x}_{i\bullet P})$ represents the vector of predictors for pool i, where $\bar{x}_{i\bullet p} = \frac{1}{k_i}\sum_{j=1}^{k_i} x_{ijp}$ is the arithmetic mean of the p^{th} predictor across all specimens in pool i.

The MLR model for the entire dataset is then:

$$\mathbf{Y}^* = \mathbf{X}^*\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{Y}^*_{n\times 1} = \{Y_i : i = 1 \ldots n\}$ and $\mathbf{X}^*_{n\times(P+1)} = \{\mathbf{x}_i : i = 1 \ldots n\}$. Furthermore, we assume that $E(\boldsymbol{\epsilon}) = \mathbf{0}_n$ and $Var(\boldsymbol{\epsilon}) = \sigma^2\mathbf{V}$, where $\mathbf{V}_{n\times n} = diag(1/k_i)$ is the diagonal matrix with (i,i) element equal to $1/k_i$ $(i = 1 \ldots n)$. This setting permits a classical application of weighted least squares (WLS) with weight matrix $\mathbf{V}^{-1} = diag(k_i)$; the WLS estimators of $\boldsymbol{\beta}$ and σ^2 are:

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{Y}^* \\
\hat{\sigma}^2 &= \frac{\mathbf{Y}^{*T}\left[\mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}^*(\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{V}^{-1}\right]\mathbf{Y}^*}{v_E},
\end{aligned}$$

where $v_E = n - rank(\mathbf{X}^*)$. Note that if $k_i = k$ for all pools, i.e. all pool sizes are equal,

then $\mathbf{V} = (1/k)\mathbf{I}_n$, which reduces to an ordinary least squares with estimators:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\mathbf{X}^{*T}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\mathbf{Y}^{*} \\ \hat{\sigma}^2 &= \frac{k\mathbf{Y}^{*T}\left[\mathbf{I}_n - \mathbf{X}^{*}(\mathbf{X}^{*T}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\right]\mathbf{Y}^{*}}{v_E},\end{aligned}$$

The above weighted and unweighted estimators for the vector $\boldsymbol{\beta}$, along with the corresponding estimated variance-covariance matrices of $\hat{\boldsymbol{\beta}}$, in each case, are strictly unbiased following WLS theory since:

$$E(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\mathbf{V}^{-1}E(\mathbf{Y}^{*}) = (\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*}\boldsymbol{\beta} = \boldsymbol{\beta}$$

and, letting $\mathbf{A} = [\mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}^{*}(\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\mathbf{V}^{-1}]$,

$$\begin{aligned}E(\hat{\sigma}^2) &= \frac{E(\mathbf{Y}^{*T}\mathbf{A}\mathbf{Y}^{*})}{v_E} \\ &= \frac{1}{v_E}\{tr[\mathbf{A}Var(\mathbf{Y}^{*})] + E(\mathbf{Y}^{*})^T\mathbf{A}E(\mathbf{Y}^{*})\} \\ &= \frac{1}{v_E}[\sigma^2 tr(\mathbf{A}\mathbf{V}) + \boldsymbol{\beta}^T\mathbf{X}^{*T}\mathbf{A}\mathbf{X}^{*}\boldsymbol{\beta}] \\ &= \frac{\sigma^2}{v_E}tr[\mathbf{I}_n - \mathbf{V}^{-1}\mathbf{X}^{*}(\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}] \\ &= \frac{\sigma^2}{v_E}[n - rank(\mathbf{X}^{*})] \\ &= \sigma^2,\end{aligned}$$

since $\mathbf{X}^{*T}\mathbf{A}\mathbf{X}^{*} = \mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X} - \mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*}(\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X} = \mathbf{0}$.

2.2.2 Unequal Aliquot Volumes

When aliquot volumes are not uniform across specimens, we assume these differing aliquot volumes are known and yield a pooled measurement that is a weighted average of the specimens constituting that pool. In these situations, appropriate adjustments must be made to the weight matrix used in the WLS analysis. Let a_{ij} be the number of units (e.g., mL) contributed by the j^{th} member of pool i. It is then reasonable to assume that the measurement for pool i is the weighted average $Y_i = \left(\sum_{j=1}^{k_i} a_{ij}\right)^{-1}\sum_{j=1}^{k_i} a_{ij}Y_{ij}$, and the

MLR model for the i^{th} pool becomes

$$Y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i,\ i = 1 \ldots n,$$

where $\mathbf{x}_i = \left(\sum_{j=1}^{k_i} a_{ij}\right)^{-1} \sum_{j=1}^{k_i} a_{ij}\mathbf{x}_{ij}$ is the vector of predictors for the i^{th} pool consisting of the weighted averages of each predictor across all specimens in pool i. The random error is denoted by $\epsilon_i = \left(\sum_{j=1}^{k_i} a_{ij}\right)^{-1} \sum_{j=1}^{k_i} a_{ij}\epsilon_{ij}$, such that $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2 v_i$, with $v_i = \left(\sum_{j=1}^{k_i} a_{ij}\right)^{-2} \sum_{j=1}^{k_i} a_{ij}^2$. Standard WLS regression can still be performed, this time with weight matrix $\mathbf{V}^{-1} = diag(1/v_i)$. Failure to include the appropriate weights in the regression analysis would be expected to result in a loss of efficiency. Flawed inference due to invalid estimation of regression coefficient standard errors could also occur, unless robust standard errors were applied. Suppose $\mathbf{V}^{-1}$ is the true weight matrix, such that $Var(Y) = \sigma^2\mathbf{V}$, but that the weight matrix is misspecified as $\mathbf{W}^{-1}$. Then $\hat{\boldsymbol{\beta}}$ will remain unbiased, since

$$E(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^{*T}\mathbf{W}^{-1}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{W}^{-1}E(\mathbf{Y}^*) = (\mathbf{X}^{*T}\mathbf{W}^{-1}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{W}^{-1}\mathbf{X}^*\boldsymbol{\beta} = \boldsymbol{\beta}$$

but $\hat{\sigma}^2$ will be biased, since, letting $\mathbf{B} = [\mathbf{W}^{-1} - \mathbf{W}^{-1}\mathbf{X}^*(\mathbf{X}^{*T}\mathbf{W}^{-1}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{W}^{-1}]$,

$$\begin{aligned}
E(\hat{\sigma}^2) &= \frac{E(\mathbf{Y}^{*T}\mathbf{B}\mathbf{Y}^*)}{v_E} \\
&= \frac{1}{v_E}\{tr[\mathbf{B}Var(\mathbf{Y}^*)] + E(\mathbf{Y}^*)^T\mathbf{B}E(\mathbf{Y}^*)\} \\
&= \frac{\sigma^2}{v_E}tr(\mathbf{B}\mathbf{V}) \\
&= \frac{\sigma^2}{v_E}tr\{[\mathbf{W}^{-1} - \mathbf{W}^{-1}\mathbf{X}^*(\mathbf{X}^{*T}\mathbf{W}^{-1}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{W}^{-1}]\mathbf{V}\} \\
&\neq \sigma^2
\end{aligned}$$

Furthermore, if aliquot information is excluded from calculation of each pool's covariate vector (i.e. if the unweighted means are used), coefficient estimates may also be biased, since, now, $\hat{\boldsymbol{\beta}}_u = (\mathbf{X_u}^{*T}\mathbf{V}^{-1}\mathbf{X_u}^*)^{-1}\mathbf{X_u}^{*T}\mathbf{V}^{-1}\mathbf{Y}^*$, where $\mathbf{X}_u^*$ is the pooled design matrix of

unweighted covariate means, and

$$E(\hat{\boldsymbol{\beta}}_u) = (\mathbf{X_u}^{*T}\mathbf{V}^{-1}\mathbf{X_u}^{*})^{-1}\mathbf{X_u}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*}\boldsymbol{\beta} \neq \boldsymbol{\beta}.$$

2.3 Pooling and Selection Methods

Suppose data on the vector of predictors ($\mathbf{X}$) has been collected on N subjects, but that the budget permits only n lab assays ($n < N$) for assessment of the response Y. The simplest way to reduce the number of assays is to randomly select n specimens for inclusion in the analysis. Another strategy is to randomly allocate each of the N specimens into one of n equal-sized pools, so that all (or essentially all) specimens are included in the analysis. Given that the predictor ($\mathbf{X}$) data are available on each subject prior to pooling, however, substantial gains in efficiency relative to these random strategies are possible when this information is applied to the selection or pooling process.

Based on the WLS models presented in Section 2.2, an optimal pooling or selection strategy with respect to efficient estimates of the coefficient vector $\boldsymbol{\beta}$ would minimize $Var(\hat{\beta}_p)$ for all $p = (1, \ldots, P)$, where P is the total number of predictors. In the case of SLR, $P = 1$, so this simplifies to minimizing

$$Var(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n} w_i(x_i - \bar{x}_w)^2}, \tag{2.1}$$

where w_i is the weight corresponding to observation i and $\bar{x}_w = (\sum w_i)^{-1}(\sum w_i x_i)$ is the weighted mean.

2.3.1 "Smart" Selection

For any strict selection strategy applied to the data (i.e. only non-pooled data), $w_i = 1$ for all i in (2.1), assuming no weight contributions unrelated to pooling (e.g. from sampling design). For the most efficient selection strategy, minimizing (2.1) equates to choosing the n observations with covariate (x_i) values farthest from their mean ($\bar{x}$). When $\mathbf{x} = (x_1, \ldots, x_N)$ is symmetric, this can often be achieved by ordering the data by $\mathbf{x}$, then selecting half of the desired number of samples from each of the top and bottom of the ordered data. We

refer to this strategy as "smart" selection.

2.3.2 "Smart" Pooling

Although "smart" selection is the most efficient selection strategy for SLR, one major disadvantage of this method is the complete omission of some biospecimens (generally, those closest to the overall mean) from the analysis. A potential improvement is based on a similar idea, but utilizes pooling instead of selection to limit the total number of lab assays performed.

For pools with equal aliquot volumes, $w_i = k_i$ is the number of assays in pool i (pool size), and the components in the denominator of (2.1) are defined as $x_i = \frac{1}{k_i}\sum_{j=1}^{k_i} x_{ij}$ and $\bar{x}_w = \frac{1}{N}\sum_{i=1}^{n} k_i x_i = \frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{k_i} x_{ij}$. Minimizing the variance of $\hat{\beta}_1$ is then synonymous with maximizing the between-pool sum of squares. When pool sizes are equal, this can be achieved by ordering the data by $\mathbf{x}$ and forming pools sequentially, so that pool i contains the set of observations $\left\{(y, x_{(j)}) : (i-1)k < j \leq ik\right\}$ where $(y, x_{(j)})$ is the observation associated with the j^{th} order statistic of $\mathbf{x}$. We call this strategy "smart" pooling; similar arguments have been made using a D-optimality design for pooling to assess X (rather than Y) under an SLR formulation (Ma et al., 2011).

When pools sizes are equal, the "smart" pooling strategy will minimize the variance of $\hat{\beta}_1$. To see this, consider two pools of size k, denoted P_r and P_s, $r < s$, such that $P_r = \{(y, x_{(j)}) : (r-1)k < j \leq rk\}$ and $P_s = \{(y, x_{(j)}) : (s-1)k < j \leq sk\}$, where $x_{(j)}$ is the j^{th} order statistic. Let $\bar{x}_r$ and $\bar{x}_s$ denote the measured values for these pools, respectively. Now suppose that two elements in these pools are switched, say $x_{r'}$ from pool P_r and $x_{s'}$ from pool P_s.

Then the measured value for pool P_r will increase, since every element in the original pool P_s is greater than every element in the original pool P_r. Similarly, the measured value for pool P_s will decrease. Let δ_r denote this change in the measured value of P_r, and let δ_s denote the absolute value of the change in the measured value of P_s, such that

$$\delta = \delta_r = \delta_s = \frac{1}{k}(x_{s'} - x_{r'})$$

The sum of squares components for the original pools is:

$$(\bar{x}_r - \bar{x})^2 + (\bar{x}_s - \bar{x})^2$$

And for the new pools is

$$(\bar{x}_r + \delta_r - \bar{x})^2 + (\bar{x}_s - \delta_s - \bar{x})^2$$

Then

$$\begin{aligned}
&(\bar{x}_r + \delta - \bar{x})^2 + (\bar{x}_s - \delta - \bar{x})^2 \\
&\quad = (\bar{x}_r - \bar{x})^2 + (\bar{x}_s - \bar{x})^2 + 2\delta(\bar{x}_r - \bar{x}) - 2\delta(\bar{x}_s - \bar{x}) + 2\delta^2 \\
&\quad = (\bar{x}_r - \bar{x})^2 + (\bar{x}_s - \bar{x})^2 + 2\delta\left[\bar{x}_r - \bar{x}_s + \delta\right] \\
&\quad = (\bar{x}_r - \bar{x})^2 + (\bar{x}_s - \bar{x})^2 + \frac{2}{k^2}(x_{s'} - x_{r'})\left[(k\bar{x}_r - x_{r'}) - (k\bar{x}_s - x_{s'})\right] \\
&\quad = (\bar{x}_r - \bar{x})^2 + (\bar{x}_s - \bar{x})^2 - \frac{2}{k^2}(x_{s'} - x_{r'})\left(\sum_{s \neq s'} x_s - \sum_{r \neq r'} x_r\right) \\
&\quad < (\bar{x}_r - \bar{x})^2 + (\bar{x}_s - \bar{x})^2
\end{aligned}$$

since $\frac{2}{k^2} > 0$, $(x_{s'} - x_{r'}) > 0$, and $\sum_{s \neq s'} x_s > \sum_{r \neq r'} x_r$. Thus, deviating from the "smart" pooling strategy results in a decrease of the total between-pool sum of squares, corresponding to an increase in the variance of $\hat{\beta}_1$. This result generalizes to any change in the "smart" pooling strategy, since any alteration of the pools can be broken down into pairwise switches.

2.3.3 k-means Clustering

Further improvements in efficiency can be achieved over "smart" pooling when pool sizes are permitted to vary. An optimal solution can be targeted through a k-means clustering algorithm, which is designed to distribute experimental units into groups, or clusters, such that the between-cluster sum of squares is maximized (Hartigan, 1975). In the SLR case, the clusters so identified comprise the optimal pools by virtue of minimizing (2.1), since the between-cluster sum of squares (now the between-pool sum of squares) has been maximized in the k-means algorithm.

Although the k-means algorithm identifies the optimal pooling strategy for a fixed number of pools in an SLR scenario, it provides only small efficiency gains over "smart" pooling, which differs from k-means clustering only in that it requires each pool to contain the same number of specimens. The main advantage of k-means pooling emerges from its flexibility as a tool for efficient allocation of subjects to pools based on an arbitrary number of predictor variables, such as in multiple linear regression (MLR). In an MLR setting, we seek to minimize $Var(\hat{\beta}_p)$ for $p = 1, \ldots, P$, where

$$\begin{aligned} Var(\hat{\beta}_p) &= \left[p^{th}\text{element of } diag\left(\sigma^2(\mathbf{X}^{*T}\mathbf{V}^{-1}\mathbf{X}^{*})^{-1}\right)\right] \\ &= \sigma^2\left[(1 - r^2_{\mathbf{x}_p|\mathbf{X}_{(-p)}})\sum_{i=1}^{n} w_i(x_{i,p} - \bar{x}_p)^2\right]^{-1}. \end{aligned} \tag{2.2}$$

$\bar{x}_p$ is the weighted mean of $\mathbf{x}_p = (x_{1,p}, \ldots, x_{n,p})$, the vector of the p^{th} covariate values for each observation, and $r^2_{\mathbf{x}_p|\mathbf{X}_{(-p)}}$ is the squared coefficient of multiple determination from the weighted regression of $\mathbf{x}_p$ on the other covariates. Of course, simultaneously maximizing efficiency for all regression coefficients is challenging, since a near-optimal pooling strategy for one can be far from optimal for others. The k-means clustering algorithm is particularly helpful in this case, since it aims to maximize $\sum_{i=1}^{n} w_i(x_{i,p} - \bar{x}_p)^2$, the between-cluster sum of squares, for all p (Hartigan, 1975). In concept, this is a generalization of the D-optimal design for SLR, which seeks to maximize the determinant of the $\mathbf{X^T X}$ matrix.

Reducing the total within-cluster sum of squares under the k-means clustering algorithm can also help reduce $r^2_{x_p|\mathbf{X}_{(-p)}}$, thus improving the efficiency of the coefficient estimate for β_p. To see this, consider an example with two independent covariates and 100 observations, where we seek to create 4 distinct pools based on the observed covariate data. Figure 2.1 illustrates k-means clustering on X_1 versus clustering on both predictor variables (X_1 and X_2). In general, the r^2 value will be smaller when all variables are used to form clusters. This reduction in r^2, however, may be accompanied by a decrease in the between sum of squares value for one of the variables. In the Figure 2.1 example, although the between sum of squares for X_1 is smaller when both variables are included in the clustering procedure (right panel), the corresponding reduction in r^2 is large enough to produce an

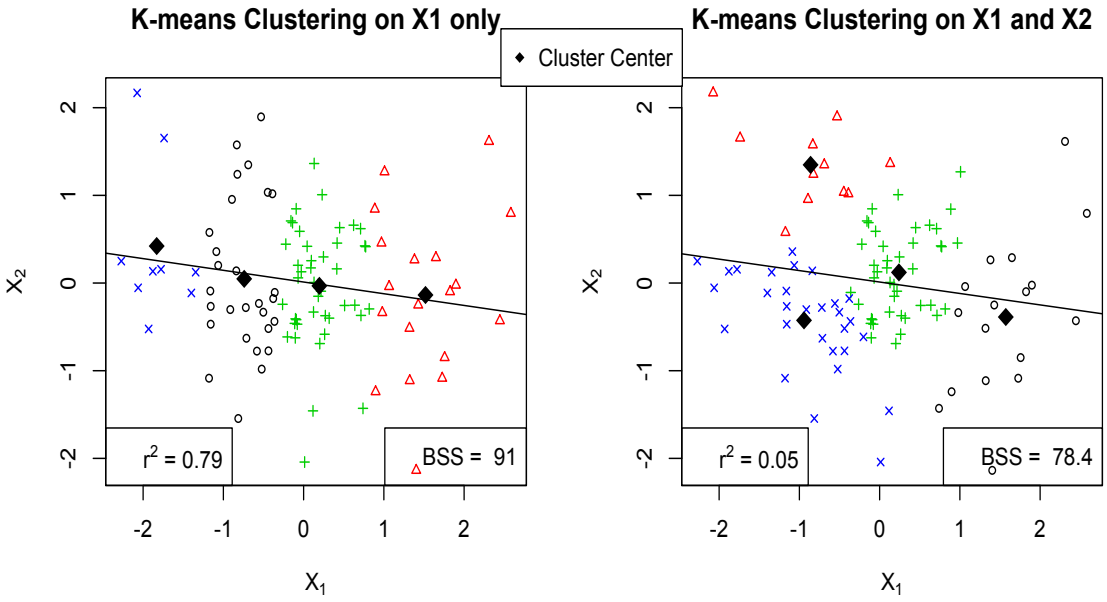

Figure 2.1: k-means clustering on independent variables. 100 observations were grouped into four clusters, indicated by different symbols. The left panel displays clusters based only on X_1. The right panel shows clusters based on both variables. Between SS values for X_1 and r^2 values corresponding to the resulting 4 pools are displayed for each clustering strategy.

overall reduction in variance for the coefficient of X_1. Thus, even when covariates are independent and particularly when they are correlated, k-means clustering applied to all covariates tends to improve overall efficiency, especially when all are viewed as equally important.

The *kmeans* function in R version 2.15.0 was used to define k-means clusters in this report. The desired number of clusters (pools) can be input into the function, making the approach a natural fit for the purpose of study planning based on a fixed number of budgeted lab assays. Specifically, we recommend the application of k-means clustering to the complete data on the full set of predictors $(x_1, \ldots, x_P)$, while specifying a number of clusters equal to the number of laboratory assays supported by the study budget. Given the requested number of clusters, the function randomly chooses a distinct set of observations from the input dataset as initial cluster centers. It then searches for the optimal clustering strategy in a neighborhood of these initial centers (R Development Core Team, 2012). Like most k-means clustering algorithms, the *kmeans* function in R finds a locally optimal solution

Table 2.1: Empirical standard deviations of regression coefficient estimates from 1000 simulations comparing k-means clustering in SAS and R. $N = 240$ and $n = 30$ for each simulation, with an average pool size of 8.

Method	β_1	β_2	β_3	β_4
SAS FASTCLUS	0.1112	0.0195	0.0130	0.0088
R local *kmeans*	0.1092	0.0193	0.0128	0.0088
R global *kmeans*	0.1108	0.0189	0.0129	0.0086

due to the computational complexity of the problem. Efforts have been made to find more globally optimal solutions through the prudent choice of initial cluster centers (e.g. leader algorithms) or by repeating the procedure at a number of random starts and choosing the best clustering (Jain et al., 1999). The SAS FASTCLUS procedure, for instance, uses Hartigan's leader algorithm to choose initial cluster centers prior to performing the k-means clustering algorithm. Examples of the application of the k-means algorithm in R and SAS are provided in Appendix A.1.

Table 2.1 illustrates a simulation study comparing the efficiency of three implementations of k-means clustering. 1000 simulations were performed where data was generated with $N = 240$ and $n = 30$ to mimic the simulations performed in Section 2.4.3. Clusters were formed using SAS's PROC FASTCLUS, R's *kmeans* function, and a global version applying a stepwise strategy to R's *kmeans* function (Likas et al., 2003). Empirical standard deviations are provided to compare precision of regression estimates from pools formed under each of the clustering tools.

These results suggest that, on average, each method yields very comparable estimate precision in the linear regression setting studied here. Since local k-means is more accessible and computationally efficient than global k-means, all subsequent k-means clustering results were produced using R's local *kmeans* function, unless otherwise noted. Even though this simulation study suggests that global and local k-means perform similarly, the global version gives the same clustering every time, whereas clusters formed from the local version can vary, due to the random selection of initial cluster seeds. FASTCLUS will also generate the same clusters each time, so long as the order of the observations in the dataset is not changed. Since R was used for all simulations in this report, we apply the global version

when performing data analysis in Section 2.5, to avoid the complication of choosing from multiple clustering options.

When a particular coefficient is of primary interest, variations on the standard k-means procedure, such as weighted k-means clustering, can be performed to further improve precision. It is common practice to standardize variables prior to performing k-means clustering, in order to prevent any one variable from exerting too much or too little influence on the algorithm (SAS Institute Inc., 2010). A particular variable can be multiplied by a weight constant after standardization, in order to exert more influence on the clustering procedure and, subsequently, to reduce the variance of the weighted variable's coefficient estimate. This improved efficiency through weighting one variable will nearly always result in a decrease in efficiency for all other variables, so it is important to carefully assess the goals of the regression analysis prior to performing weighted k-means clustering.

It is worth noting that, when the correct weights are applied, the WLS regression coefficient estimates and their estimated variances remain unbiased for all of the pooling and selection methods considered here, including "smart" selection and pooling as well as pooling based on k-means clustering. This follows from well-known missing data theory, since these strategies depend only on the fully observed covariate values and not on the outcome (Y) conditional on the covariates (Glynn and Laird, 1986; Little, 1992; Little and Rubin, 2002).

In the next section, we use simulations mimicking the BioCycle dataset to compare the k-means clustering strategy to the previously described "smart" and random pooling and selection strategies in SLR, considering both equal and unequal aliquot scenarios. We then simulate a multiple linear regression setting to compare regression on pools formed from standardized k-means to those formed from clustering on a single variable, as well as to a weighted k-means approach. Finally, we use artificial pooling to test these methods on the BioCycle Study dataset under both SLR and MLR.

Table 2.2: Mean and standard deviation for covariates in the BioCycle dataset, as well as estimated regression coefficients, standard errors, and residual error (σ) for SLR and MLR on outcome HDL from the complete BioCycle study dataset ($N = 240$).

		Estimate (SE)	
Variable	Mean (SD)	SLR	MLR
Intercept		21.282 (2.394)	13.801 (2.925)
log(estradiol)	4.730 (0.639)	-0.832 (0.502)	-0.938 (0.471)
BMI	3.174 (0.158)		0.079 (0.078)
Vitamin E	2.147 (0.519)		0.041 (0.037)
Age	27.30 (8.164)		0.207 (0.037)
σ		4.954	4.638

2.4 Simulation Study

The BioCycle Study, conducted from 2005 to 2007, followed premenopausal women from Western New York State for one or two complete menstrual cycles. Regularly menstruating women not currently taking oral contraceptives were eligible for participation. 259 women between the ages of 18 and 44 completed the study. Data collected during the study included age (years) and BMI (kg/m^2), as well as serum estrodial, vitamin E, and HDL levels, which were measured on the 22nd day of a participant's menstrual cycle. Participant BMI was right-skewed, with values ranging from 16.1 to 35.0, with an average body mass index of 24.2. Vitamin E and estradiol levels were also right-skewed, with average values of 10.1 and 136.4, respectively. Age appeared to be approximately normal, with an average participant age of 27 and standard deviation of 8.2. To facilitate analysis, observations containing missing data were removed, and the first 240 of the remaining 242 complete cases were included in the final dataset. In our study, we treat HDL level as the outcome and perform artificial pooling on this variable to test the various pooling strategies. The remaining variables were treated as fully known and their values were used to facilitate the k-means and "smart" pooling and selection processes.

The distributions of the covariates BMI, vitamin E level, and estradiol level were right-skewed. Table 2.2 gives the mean and standard deviation for each of these variables after a log transformation, along with the mean and standard deviation for patient age. The sample covariances among these covariates ranged from -0.023 to 0.428. Table 2.2 also

provides the results (based on complete data) from two regression models of interest: the SLR of HDL on log(estradiol), and the MLR of HDL on log(estradiol), BMI, vitamin E, and age.

The results from the regression analysis on the complete BioCycle study dataset motivated the following simulations to test the pooling and selection strategies discussed in Section 2.3. To mimic the predictor variables from this dataset, a multivariate normal distribution was generated with mean vector and covariance matrix matching the observed sample means and covariances of the predictor variables for each simulation. The simulated version of log(BMI) and log(vitamin E) were exponentiated to match their format in the original dataset, while log(estradiol) was not transformed. The outcome (HDL) was then generated via SLR or MLR, based on the estimated parameters summarized in Table 2.2.

2.4.1 SLR: Equal Aliquots

10,000 simulations were performed in R for each scenario. For the first simulation study, we simulate a simple linear regression with equal aliquot volumes. For this simulation, the predictor and outcome variables were generated to mimic the results from regressing HDL levels on log(estradiol) from the BioCycle Study (Table 2.2), such that $Y \sim N(21.3 - 0.83X_1, 4.95^2)$. Simple linear regression was performed under each of the pooling and selection methods discussed in Section 2.3, as well as random pooling and random selection. The *kmeans* function in R version 2.15.0 was used to define k-means clusters. For random selection, only the first n observations from the simulated dataset were retained, while random pooling assigned each group of $k = N/n$ sequential observations to the same pool. "Smart" pooling was performed similarly to random pooling, except that the simulated data was ordered by X_1 prior to assigning pools. "Smart" selection was conducted by calculating the squared distance $(x_{i1} - \bar{x}_1)^2$ for all i, then choosing the n observations with the largest squared distance values to be included in the analysis.

For all simulations (both SLR and MLR), performance of each method is assessed through bias, relative efficiency, and 95% confidence interval coverage for coefficient estimates, where relative efficiency is defined as the ratio of $\text{SD}(\hat{\beta}_p^f)$, the empirical standard deviation (SD) of $\hat{\beta}_p$ from the full data regression, to $\text{SD}(\hat{\beta}_p)$, the SD of $\hat{\beta}_p$ under the spec-

Table 2.3: Relative efficiency and 95% confidence interval (CI) coverage of $\hat{\beta}_1$ from SLR simulation on $N = 240$ observations with equal volume aliquots. n = number of pools formed or observations selected for analysis, and relative efficiency = $SD(\hat{\beta}_1^f)/SD(\hat{\beta}_1)$, where $\hat{\beta}_1^f$ is the parameter estimate under the full data regression. Regression parameters were simulated to mimic the SLR results of HDL levels on log(estradiol) from the BioCycle study.

	Relative Efficiency (95% CI Coverage)			
Method	$n = 120$	$n = 60$	$n = 30$	$n = 16$
k-means	1.000 (95.1)	0.999 (95.0)	0.996 (95.0)	0.993 (95.1)
Smart Pooling	0.999 (95.2)	0.998 (95.1)	0.995 (95.1)	0.988 (95.3)
Smart Selection	0.963 (94.9)	0.837 (94.9)	0.691 (94.8)	0.562 (95.2)
Random Pooling	0.703 (95.2)	0.491 (95.0)	0.337 (95.0)	0.231 (95.0)
Random Selection	0.704 (95.2)	0.489 (95.0)	0.341 (95.2)	0.237 (95.3)

ified method. To calculate confidence intervals, the additional assumption of normality of errors is applied, so that confidence intervals are calculated as: $\hat{\beta}_p \pm t_{0.975,df}\hat{SE}$, where $\hat{SE}$ is the standard error estimate of $\hat{\beta}_p$, and $t_{0.975,df}$ is the critical value of the t-distribution with $df = n - (P + 1)$ degrees of freedom.

Since all methods provide unbiased estimates of β_1 as well as its standard error, these values were omitted in order to streamline the results. Table 2.3 displays the relative efficiency and confidence interval coverage for the SLR results. While all methods also provide appropriate confidence interval coverage ($\sim 95\%$) for $\hat{\beta}_1$, estimates calculated under k-means pooling are the most precise for every sample size, closely approximating the results from the full data and losing only a trivial amount of precision as the total number of pools decreases. The "smart" pooling method performs similarly to k-means, although with slightly less precision in all situations, likely due to its additional restriction of equal-sized pools. As expected, the random selection and pooling strategies are the least efficient methods, displaying considerable precision loss with decreased sample size. While both random strategies appear to perform similarly with respect to precision, "smart" pooling noticeably outperforms "smart" selection, providing estimates from the smallest sample size simulation ($n = 16$) that are more precise than estimates from the largest simulated sample size under "smart" selection ($n = 120$).

Another advantage of using "smart" pooling over "smart" selection is its facility in

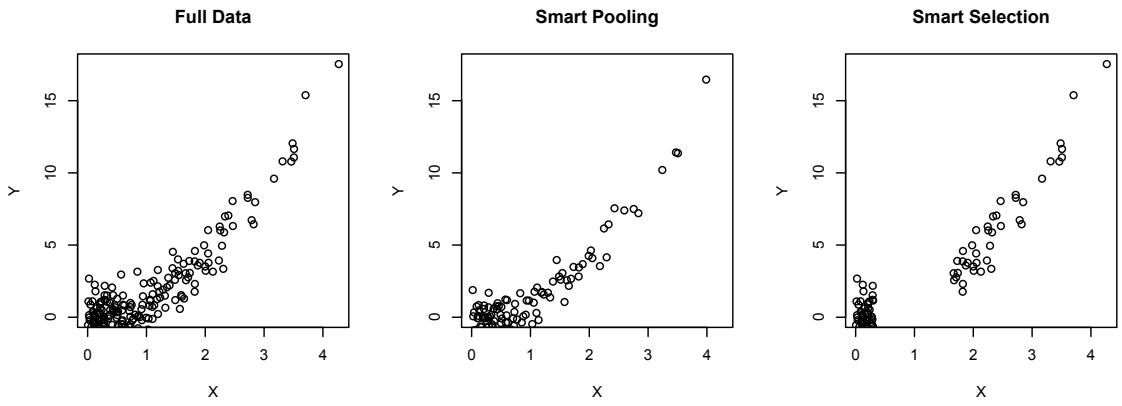

Figure 2.2: Scatterplots of X and Y for the full data, "smart" pooling, and "smart" selection methods.

regression diagnostics. It is often beneficial to check the assumption of a linear association between the outcome and the predictor variable to determine whether a transformation on X is required. For example, if a quadratic instead of a linear relationship exists between X and Y, this relationship is often preserved in "smart" pooling, but may not be as apparent in "smart" selection. Figure 2.2 demonstrates a situation where "smart" pooling gives a clearer picture of the true relationship between X and Y than "smart" selection. For this illustration, 200 observations were generated, with $X \sim \text{Exp}(1)$ and $Y \sim N(X^2, 1)$. For "smart" pooling, 100 pools of size two were formed, and for "smart" selection, 100 observations were selected.

The quadratic relationship between X and Y, which can be identified in the scatterplot from the full data, is preserved in the "smart" pooling scatterplot. This relationship is not as apparent in the "smart" selection scatterplot due to elimination of observations near the center of the X distribution, and is less likely to be identified under this method.

2.4.2 SLR: Unequal Aliquots

When pools consist of aliquots with unequal volumes, it is important to apply the correct weights in the regression analysis (specified in Section 2.2.2). Failure to do so would likely result in a loss of precision for coefficient estimates, inappropriate standard error estimates, or both.

Table 2.4: Relative efficiency with respect to the full data and 95% confidence interval (CI) coverage of $\hat{\beta}_1$ from SLR simulations with unequal aliquot volumes.

	Relative Efficiency (95% CI Coverage)			
Method	$n = 120$	$n = 60$	$n = 30$	$n = 16$
	Selection Strategies			
Smart Selection	0.963 (94.9)	0.837 (94.9)	0.691 (94.8)	0.562 (95.2)
Random Selection	0.704 (95.2)	0.489 (95.0)	0.341 (95.2)	0.237 (95.3)
	Correctly-Specified Aliquots[†]			
k-means	0.999 (94.9)	0.995 (95.1)	0.986 (95.1)	0.950 (95.1)
Smart Pooling	0.957 (95.2)	0.930 (95.3)	0.919 (95.2)	0.908 (95.1)
Random Pooling	0.704 (95.2)	0.490 (94.8)	0.338 (95.0)	0.232 (95.0)
	Ignoring Aliquots			
k-means	0.950 (94.9)	0.929 (95.1)	0.918 (95.2)	0.911 (94.9)
Smart Pooling	0.950 (95.2)	0.926 (95.2)	0.917 (95.2)	0.906 (95.3)
Random Pooling	0.669 (95.1)	0.453 (95.0)	0.311 (95.1)	0.210 (94.7)

[†]Aliquot volumes randomly selected from the set $(1/4, 1/2, 3/4, 1)$ informed the k-means clustering strategy as well as the appropriate weights for all pooling strategies under the heading "Correctly-Specified Aliquots." "Ignoring Aliquots" presents regression results when aliquot volumes are ignored in both the pool allocation and analysis steps.

Assuming aliquot volumes are known a priori, estimate precision can also be improved by using these values to inform the clustering procedure, particularly in simple linear regression. By including aliquot volume as if it were another subject-specific covariate in the clustering algorithm, pools are formed from specimens with similar aliquot sizes. This strategy can increase precision since w_i, the weight contribution for pool i in (2.1), tends to deflate when pool i is formed from specimens with different aliquot sizes, thus increasing the overall variance for $\hat{\beta}_1$. By forming pools from specimens with similar aliquot sizes, this potential precision loss is mitigated. This improvement in efficiency is most noticeable in simple linear regression with a binary predictor, but becomes less apparent with the inclusion of additional predictor variables in the regression formulation. Fortunately, the relative efficiency of each pooling method can be assessed prior to actually forming the pools by evaluating the denominators of equations (2.1) or (2.2), depending on which type of regression is being performed.

Table 2.4 illustrates the potential efficiency gains when aliquot volumes are included in

the clustering algorithm, as well as the consequences of ignoring differing aliquot volumes in the regression analysis. For this simulation, aliquot volumes of $1/4, 1/2, 3/4$, and 1 were randomly assigned to observations. These volumes were used to inform the k-means clustering procedure only, but were included as weights in the regression analysis for all pooling strategies. While aliquot volumes were assumed to have no effect on the selection strategies, the SLR results under these methods are reiterated in this table for comparison purposes.

A common mistake when dealing with unequal-sized aliquots might be to ignore the aliquot sizes completely, omitting this information from the clustering procedure, the calculation of the weighted mean of the predictor variables, and the application of the weights in the regression procedure. If the covariate means are calculated correctly (i.e. as weighted means), but the weights in the regression procedure exclude aliquot information, the coefficient estimates are expected to remain unbiased, but the estimate of their variance is expected to be incorrect. Furthermore, if aliquot information is excluded from the calculation of the pooled covariates, the coefficient estimates themselves may be biased. Results from this erroneous method are also included in Table 2.4 as an illustration of the potential consequence of failing to adequately account for differing aliquot volumes. In this simulation, the bias associated with $\hat{\beta}_1$ (not shown here) under models which completely ignore aliquot volumes was minor, although this will not always be the case.

Incorporating aliquot volumes into the k-means clustering procedure provides estimates of β_1 that are nearly fully efficient for the larger sample sizes, while omitting this information from the pooling strategy, as in "smart" pooling, results in relative efficiency levels that are outperformed by "smart" selection in the largest sample size simulated. This reduced efficiency under "smart" pooling, however, is much less sensitive to decreasing sample sizes and easily beats "smart" selection in each of the remaining sample size cases. In all cases, correctly incorporating aliquot information into the calculation of pooled covariates and weights improves estimate precision.

While aliquot volumes in this simulation were assumed to be random, in practice it is often the case that the aliquot volume is correlated with a predictor variable. This could occur, for instance, if specimens from certain demographic or exposure groups are of particular interest, and, subsequently, a greater portion of these specimens were used in

other studies. In such cases, using aliquot sizes to inform pools is less helpful because the k-means clustering procedure will form similar pools regardless of aliquot volume due to the correlation between aliquot size and the predictor variable.

2.4.3 Multiple Linear Regression

Since k-means is clearly the most efficient pooling method out of those tested in an SLR setting, we now assess its performance in MLR. 10,000 simulations were conducted to mimic the BioCycle dataset, with $N = 240$ and $(X_1, \log(X_2), \log(X_3), X_4) \sim N_4(\mu_X, \Sigma)$ where $\mu_X = (4.730, 3.174, 2.147, 27.296)$ (see Table 2.2), and (to match the sample covariance matrix),

$$\Sigma = \begin{pmatrix} 0.408 & 0.008 & \text{-}0.023 & 0.140 \\ 0.008 & 0.025 & \text{-}0.006 & 0.167 \\ \text{-}0.023 & \text{-}0.006 & 0.269 & 0.428 \\ 0.140 & 0.167 & 0.428 & 66.64 \end{pmatrix}.$$

The outcome (Y) was then generated, conditional on the simulated covariate values, such that $Y \sim N(\mu_Y, 4.64^2)$, where $\mu_Y = 13.8 - 0.94X_1 + 0.08X_2 + 0.04X_3 + 0.21X_4$.

In this simulation study, standard k-means is compared to weighted k-means, "smart" pooling, and "smart" selection on X_1, where X_1 may be considered the main variable of interest. For "smart" pooling and selection, only X_1 was included in the pooling or selection procedure. For standard k-means, all covariates were standardized prior to clustering, to ensure that each would contribute a similar impact on the final clusters. Covariates were also standardized for weighted k-means, but X_1 was then multiplied by a weight of 2. The value of two was chosen for the weights in this analysis as it provides a reasonable balance toward slightly improving the precision of $\hat{\beta}_1$ without considerably penalizing the precision of the other coefficient estimates.

Figure 2.3 gives an illustration of weighting the k-means clustering procedure, where weights with various magnitudes are applied to X_1 prior to clustering, with $N = 240$ and $n = 60$. Points represent the relative efficiency of the regression estimates, compared with efficiency of estimates from the full data.

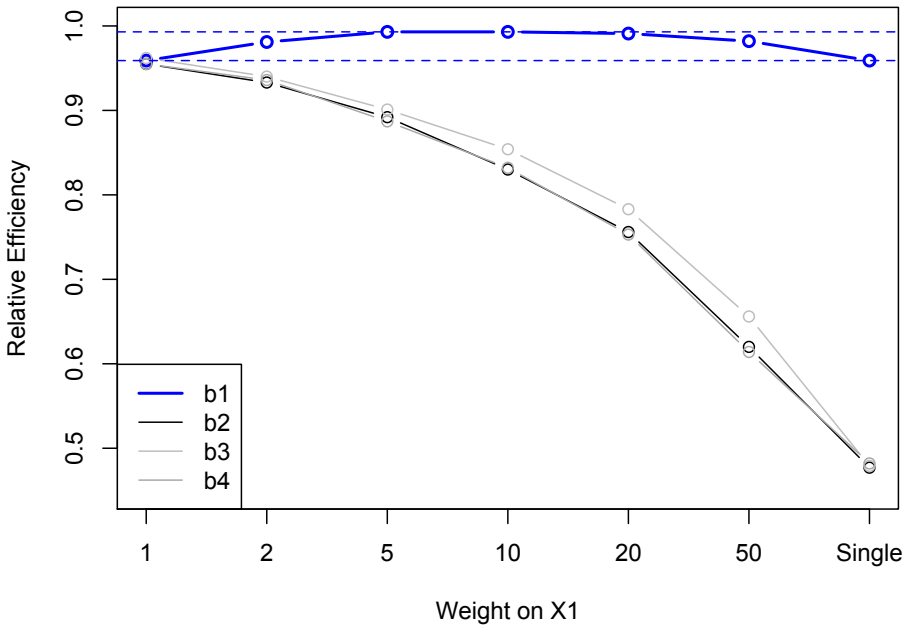

Figure 2.3: Weighted k-means, with weights applied to X_1 for 60 pools. "Single" refers to k-means applied only to X_1. Horizontal dotted lines provide visual comparison of the minimum and maximum relative efficiency values.

As evidenced in this illustration, β_1 gains a small amount of precision, with maximal precision obtained between weights of 5 and 10. The corresponding drop in precision for the remaining coefficients, however, is severe. Fortunately, analysis of the potential precision change based on the chosen weight is available prior to pooling, since we are working under the assumption that the complete individual-level covariate matrix is known. In general, small to moderate weights are preferred since weighted k-means becomes indistinguishable from "smart" pooling when weights become too large; the influence of the remaining variables on weighted k-means clustering is essentially nullified.

Relative efficiency and 95% confidence interval coverage for these MLR simulations are displayed in Table 2.5. Results from random pooling are also supplied for comparison purposes, where observations were randomly combined into n equal-sized pools. Random selection was also performed, but these results were omitted from Table 2.5 since they proved indistinguishable from random pooling. Again, bias is omitted from the results display since all coefficient estimates are unbiased. All methods at all sample sizes also provide close to nominal 95% confidence interval coverage, confirming the validity of the estimates as well as their estimated standard errors under each pooling method. While the precision of each coefficient estimate is proportional to the total number of pools, the relative efficiency of these estimates varies between methods.

For β_2, β_3, and β_4, standard k-means provides the most precise estimates at all sample sizes, maintaining over 98% efficiency at half the number of lab tests required, and continues to maintain over 84% efficiency at only 16 lab tests. Weighted k-means performs similarly, but with slightly less precision for these unweighted variables. Weighted k-means provides the most precise estimates of β_1, our coefficient of interest, maintaining over 93% efficiency at the smallest sample size.

Regardless of the correlation between the covariates, further simulations (not shown) indicate that weighted k-means regularly outperforms both single and standard k-means for efficiently estimating β_1, and standard k-means performs best for the remaining coefficients. It is important to keep in mind that the performance of weighted k-means is dependent on the magnitude of the chosen weights, with larger weights for one variable often corresponding to worse precision for the unweighted variables. Although it is possible to improve the

Table 2.5: Relative efficiency and 95% confidence interval (CI) coverage for estimated regression coefficients after pooling based on different versions of k-means at various sample sizes. "Standard" refers to standard k-means pooling, "Single" to pooling based on k-means clustering on X_1 only, "Weighted" to pooling based on weighted k-means clustering on X_1, and "Random" to random pooling into equal-sized pools.

	Relative Efficiency (95% CI Coverage)			
Method	β_1	β_2	β_3	β_4
		$n = 120$		
Standard	0.983 (95.2)	0.985 (95.0)	0.984 (95.0)	0.981 (94.9)
Weighted	0.993 (95.2)	0.978 (94.9)	0.980 (95.1)	0.977 (95.0)
Single	0.988 (95.2)	0.697 (95.0)	0.691 (94.9)	0.698 (95.2)
Random	0.696 (94.7)	0.706 (95.1)	0.697 (95.0)	0.705 (95.3)
		$n = 60$		
Standard	0.955 (94.9)	0.957 (95.1)	0.957 (94.9)	0.951 (95.0)
Weighted	0.982 (94.8)	0.936 (95.0)	0.939 (95.2)	0.939 (95.1)
Single	0.959 (94.8)	0.479 (95.0)	0.479 (95.0)	0.478 (94.7)
Random	0.475 (94.7)	0.479 (94.7)	0.483 (95.5)	0.483 (95.2)
		$n = 30$		
Standard	0.911 (95.0)	0.901 (94.8)	0.915 (95.3)	0.905 (95.1)
Weighted	0.965 (94.8)	0.866 (94.7)	0.870 (95.0)	0.857 (94.7)
Single	0.905 (95.0)	0.324 (95.2)	0.317 (95.0)	0.320 (94.6)
Random	0.319 (95.1)	0.321 (95.3)	0.327 (95.7)	0.324 (95.2)
		$n = 16$		
Standard	0.852 (94.8)	0.842 (94.7)	0.855 (95.2)	0.845 (95.2)
Weighted	0.937 (94.8)	0.764 (94.6)	0.795 (95.2)	0.764 (94.9)
Single	0.810 (94.7)	0.208 (94.8)	0.207 (94.8)	0.206 (95.1)
Random	0.207 (95.1)	0.208 (95.2)	0.209 (95.5)	0.207 (95.1)

precision of a particular coefficient by weighting its corresponding variable, the magnitude of the cumulative precision loss in the other variables is often much more considerable than the precision gained, as illustrated in Figure 2.3. Furthermore, if the weights for a particular variable become too large, the results begin to resemble those of single k-means on that variable, which can result in a loss of precision for the variable of interest, depending on the covariate correlation structure. Thus, potential weights must be carefully considered before clustering with weighted k-means. As an efficient global strategy, we generally recommend standard k-means pooling. All of our empirical studies suggest that this method yields outstanding efficiency gains over random pooling and provides the best overall solution, particularly when all variables are considered equally important.

Table 2.6: Coefficient estimates and standard errors for SLR on BioCycle dataset. For k-means pooling, average pool size was $240/n$, with maximum pool sizes ranging from 5 ($n = 120$) to 29 ($n = 16$). Each sample size had at least one observation comprising a single pool, with the exception of $n = 16$, which had a minimum pool size of 2. For "smart" and random pooling, all pools were of size $240/n$. For the random strategies, the data was randomly ordered prior to implementing any pooling or selection.

log(estradiol) (SE)	Full Data ($n = 240$): -0.832 (0.502)			
Method	$n = 120$	$n = 60$	$n = 30$	$n = 16$
k-means	-0.834 (0.521)	-0.827 (0.448)	-0.837 (0.358)	-0.839 (0.353)
Smart Pooling	-0.825 (0.506)	-0.839 (0.449)	-0.823 (0.415)	-0.889 (0.434)
Smart Selection	-0.791 (0.512)	-0.599 (0.586)	-0.843 (0.715)	-0.865 (0.987)
Random Pooling	-0.944 (0.627)	1.430 (0.936)	0.805 (1.443)	-1.421 (1.988)
Random Selection	-1.816 (0.748)	-2.229 (1.053)	-0.698 (1.201)	-3.283 (2.838)

2.5 Data Analysis

Based on the simulations testing the various pooling and selection strategies, k-means clustering, on average, provides the most precise estimates in both simple and multiple linear regression. Now, we test these various strategies on the BioCycle study dataset, to determine whether k-means provides the most similar results to those from regression on the full dataset.

For this data analysis, random pooling and selection strategies were performed by pooling or selecting the first n observations after re-sorting the dataset by a randomly-generated variable. For all of the analyses involving k-means pooling, a global version of k-means was used (Likas et al., 2003). To compare the various pooling and selection methods, we first consider simple linear regression. Table 2.6 provides SLR results for regressing HDL level on log(estradiol). As expected from our simulations, the k-means and "smart" pooling perform the best out of all the methods in terms of approximating the coefficient estimates and standard errors from the full data regression. In general, k-means pooling provides estimates that are most similar to those from the full sample, with "smart" pooling only slightly further away. For this particular dataset, the standard errors for k-means and "smart" pooling tend to decrease as the number of pools decreases. This trait appeared in approximately 10% of the simulations performed in Section 2.4.1, indicating that this trend is uncommon but not implausible.

Table 2.7: Results from MLR on BioCycle dataset after k-means pooling under various sample sizes. The first row provides the regression estimates for the full dataset (unpooled).

	Estimate (SE)			
n	log(estradiol)	BMI	Vitamin E	Age
240	-0.938 (0.471)	0.079 (0.078)	0.041 (0.037)	0.207 (0.037)
120	-0.879 (0.472)	0.071 (0.078)	0.045 (0.037)	0.209 (0.037)
60	-0.868 (0.494)	0.057 (0.081)	0.042 (0.038)	0.205 (0.039)
30	-0.729 (0.558)	0.121 (0.092)	0.031 (0.042)	0.194 (0.043)
16	-1.297 (0.540)	0.149 (0.085)	0.042 (0.039)	0.183 (0.040)

In our next analysis, MLR was applied to clusters formed from a global k-means clustering algorithm on all four predictor variables, at various numbers of pools. The results are displayed in Table 2.7. Coefficient estimates and standard errors from halving the total number of pools (assays) approximate the results from the full sample extremely closely for all coefficients, both in terms of point estimates and standard errors. Although more discrepancy is seen as expected for smaller numbers of pools, remarkably little efficiency is lost even based on as few as 16 pools. By using the fully observed covariate data to strategically inform the pooling procedure, we can closely approximate the MLR results from the full data while drastically reducing laboratory costs.

2.6 Applying k-means to Logistic Regression

As evidenced by the simulation studies and data analysis, k-means clustering can provide a powerful and accessible tool to assign pools when performing linear regression on a pooled outcome, and is particularly helpful at promoting estimate precision when one or more covariates is continuous. Furthermore, the potential for efficiency gains available from pooling under k-means is not limited to a linear regression setting.

For instance, k-means pooling can contribute to efficient pooling in the logistic regression setting considered by Vansteelandt et al. (2000), described in Section 1.3.1, where a binary outcome is pooled. In their paper, Vansteelandt et al. advocate pooling specimens with similar or identical covariate values in order to achieve more precision. For datasets with only categorical or binary covariates, $\mathbf{x}$-homogeneous pools can often be formed, so long as the number of covariates is small relative to the sample size. For datasets in which it is not

Table 2.8: k-means pooling vs. smart and random pooling in a logistic regression setting. Mean estimate and empirical standard deviation (SD) are provided.

		Mean (SD)	
Pooling Method	$\beta_1 = 1$	$\beta_2 = -1$	$\beta_3 = 0.8$
		$N = 2000$	
Full Data	1.003 (0.065)	-1.001 (0.064)	0.802 (0.060)
		$n = 1000$	
k-means Pooling	1.006 (0.085)	-1.004 (0.083)	0.803 (0.077)
Random Sample	1.009 (0.091)	-1.005 (0.093)	0.807 (0.087)
Random Pooling	1.007 (0.099)	-1.008 (0.099)	0.805 (0.095)
Smart Pooling on X_1	1.005 (0.087)	-1.004 (0.101)	0.807 (0.097)
Smart Pooling on X_2	1.007 (0.100)	-1.006 (0.085)	0.804 (0.094)
Smart Pooling on X_3	1.009 (0.100)	-1.005 (0.100)	0.804 (0.078)
		$n = 500$	
k-means Pooling	1.020 (0.124)	-1.016 (0.121)	0.811 (0.112)
Random Sample	1.017 (0.134)	-1.010 (0.130)	0.812 (0.123)
Random Pooling	1.040 (0.203)	-1.037 (0.203)	0.819 (0.194)
Smart Pooling on X_1	1.021 (0.137)	-1.016 (0.193)	0.818 (0.185)
Smart Pooling on X_2	1.022 (0.189)	-1.021 (0.139)	0.811 (0.183)
Smart Pooling on X_3	1.026 (0.191)	-1.023 (0.197)	0.821 (0.126)

possible to create $\mathbf{x}$-homogeneous pools, k-means clustering can be implemented to form pools with similar covariate values.

Table 2.8 demonstrates the potential efficiency gains in performing k-means clustering in a logistic regression setting. For this simulation, 2500 replications with sample size $N = 2000$ were performed in R, and the *optim* function was used to maximize the log-likelihood (1.1). Three covariates, (X_1, X_2, X_3) were generated independently from standard normal distributions, and the outcome (Y) was generated from a Bernoulli distribution with $Pr(Y = 1) = -0.5 + (1)X1 - (1)X2 + (0.8)X_3$. Several pooling strategies were then applied to the dataset, testing $n = 1000$ pools and $n = 500$ pools. Pools formed by k-means clustering were compared to pools formed randomly, and by smart pooling on each of the covariates. Mean estimates and empirical standard deviation are provided.

In this logistic regression setting, k-means pooling continues to provide precise, essentially unbiased estimates of the regression coefficients, outperforming both random selection and pooling, as well as all of the smart pooling methods, even for those coefficients whose

corresponding covariates have been pooled under smart pooling. An additional note is that as the number of pools is reduced to one-fourth the original sample size, random sampling outperforms all smart pooling methods for all variables. This is likely due to the loss of information for mixed pools (those containing a mix of case and control specimens), which are more likely to occur when pool sizes are larger. This characteristic emphasizes the importance of analyzing potential efficiency loss prior to pooling, a task that is possible so long as the entire set of individual-level covariate information is known.

2.7 Discussion

When the number of lab tests that can be performed is limited by budget, pooling specimens based on k-means clustering prior to performing lab assays can be an effective way to save money with minimal information loss in a linear regression setting. For simple linear regression in particular, k-means clustering provides an optimal clustering strategy for the precision of $\hat{\beta}_1$ for a fixed number of pools (n), losing only a minimal amount of precision even for small n (or equivalently, large pool sizes). In addition, incorporating aliquot volumes, when applicable, into the k-means clustering procedure can help reduce the precision loss that may accompany pooling unequal-sized aliquots.

In multiple linear regression settings, k-means clustering provides an accessible method to identify a specified number of pools commensurate with available resources for performing lab assays. By utilizing all of the covariate data to inform pooling, it provides an excellent overall solution aimed at favorable precision for each coefficient estimate, far outperforming random pooling. Weighted k-means can be useful if the precision of a particular coefficient is deemed more important than that of the others, but should be used only after careful consideration of the potential precision reduction of the remaining coefficient estimates.

Not only does k-means clustering outperform more ad hoc pooling and selection methods with respect to maintaining coefficient precision, but also in its flexibility and straightforward application. Clearly, both random pooling and random selection are far from optimal strategies, as they lose a considerable amount of efficiency, even at the largest sample size tested. "Smart" pooling, while maintaining good efficiency in the SLR setting or for a par-

ticular variable in MLR, is not readily generalizable to incorporate all covariates in the MLR setting. "Smart" selection, even if generalized to multiple variables, provides only marginal improvements over random strategies and suffers from the exclusion of data points close to the mean. This can be problematic when assessing regression diagnostics (Figure 2.2). In addition to these disadvantages, none of these current cost-saving methods for analyzing biomarkers comes close to maintaining the high level of efficiency for all variables provided by standard k-means clustering.

The goal of this analysis and simulation study was to illustrate the benefits of using k-means clustering to inform pools when performing linear regression on a pooled outcome. The simulations were designed to mimic the BioCycle study dataset, but many alterations on these assumptions are likely to occur in real-life pooling scenarios. For instance, precision loss attendant to pooling is expected to become more sensitive to decreasing sample sizes with the inclusion of more covariates. Other considerations when pooling include the potential influence of measurement or pooling error on the measured values of the pools, which has been explored by Schisterman et al. (2010), as well as possible limitations on pooling strategies due to instrument sensitivity (e.g. minimum specimen volume requirement).

Fortunately, since the potential precision of each estimated coefficient depends only on the covariates, exploration of the best pooling or selection strategy can be investigated prior to performing any physical pooling. Thus, not only can strategic pooling of biospecimens considerably reduce laboratory costs, but the subsequent potential precision loss can be assessed prior to any actual pooling, so that the advantages and disadvantages of pooling on a specific dataset can be thoroughly evaluated beforehand. This characteristic may prove particularly useful in a cost-benefit analysis, when determining the optimal number of pools to balance statistical precision and lab expense. Furthermore, the proposed efficient pooling strategy based on k-means clustering applied to individual covariate values is expected to be efficient for any outcome that might be measured on the pooled samples via linear regression. Thus, samples pooled based on this strategy retain their potential statistical efficiency advantages for the analysis of multiple outcomes, so long as the same covariates are to be considered.

In the next two chapters, we consider extensions of linear regression on a right-skewed

outcome, which may require methods such as applying a transformation to the outcome or performing generalized linear regression under distributional assumptions appropriate for right-skewed data.

Chapter 3

Lognormal Regression Models for a Skewed, Pooled Outcome

3.1 Introduction

In the previous chapter, we highlighted the benefits of using covariate data to inform pooling on an outcome in a linear regression setting. Many biomarkers measured in laboratory analyses, however, are positive, right-skewed variables. When these biomarkers are treated as the dependent variable in regression settings, a log transformation of the individual-level outcome is often applied in order to validate standard linear regression analyses. Analysis of pooled specimens, however, may not be straightforward. In such cases, we will see that a slight modification of the usual regression method can still provide valid and precise coefficient estimates when pools are formed with identical covariate values.

When these $\mathbf{x}$-homogeneous pools cannot be formed, we recommend applying a Monte Carlo Expectation Maximization (MCEM) algorithm to identify maximum likelihood estimates (MLEs). Simulation studies demonstrate that these analytical methods provide essentially unbiased estimates of coefficient parameters as well as their standard errors when appropriate assumptions are met. Furthermore, if the fully observed covariate data is used to inform the pooling strategy, a high level of efficiency can be maintained at a fraction of the total lab cost. Utilizing these informative pooling strategies in conjunction with the

appropriate analytical techniques allows researchers to meet budgetary constraints without sacrificing precision.

In the following section we introduce the Collaborative Perinatal Project (CPP), the motivating dataset for this study. In Section 3.3 we describe the analytical methods considered, along with the conditions required for their validity. We discuss various pooling methods in Section 3.8, with particular focus on the k-means clustering technique introduced in Chapter 2 to promote estimate efficiency. Section 3.9 provides simulation studies that demonstrate the validity of the proposed analytical strategies and illustrate the benefits of informative pooling techniques. Finally, we apply these methods to a substudy from the CPP.

3.2 A Motivating Example: Cytokines in the CPP

The Collaborative Perinatal Project (CPP) was conducted between 1959 and 1974 to examine associations between various exposures and pregnancy outcomes (Hardy, 2003). In a nested case-control study of stored serum samples from the CPP, several cytokines were measured in participants that experienced a spontaneous abortion (SA), along with controls matched to cases by gestational age (GA) at sample collection (Whitcomb et al., 2007). Accompanying covariates include participant demographics such as age, race, and smoking status. While the cytokines in relation to SA from this CPP study have previously been analyzed via logistic regression with SA status as the dependent variable (Whitcomb et al., 2007, 2008, 2012), our study treats monocyte chemotactic protein 1 (MCP1) as the outcome, and SA status, age, race, and smoking status as predictors. The positive, right-skewed nature of MCP1, as well as nearly all the cytokines measured in this study, motivates the development of methods to analyze pooled, skewed outcomes in a regression setting. Specifically, we seek to estimate the parameters of an underlying individual-level lognormal regression model for the dependent variable MCP1, when measurements of MCP1 are obtained on pooled samples.

This dataset is particularly compelling since it contains both individual-level as well as pooled measurements of the cytokines, where pools were formed randomly within SA status

(maximum pool size = 2) as part of a study design incorporating a methods component to assess measurement error (Whitcomb et al., 2012). This unique characteristic enables analysis of both observed pooled measurements as well as expected pooled values, the latter calculated as the actual arithmetic mean of measurements on individual specimens. We use this dataset collected from the CPP study to illustrate analytical methods and demonstrate the advantages of informative pooling techniques, so that future studies of this type can benefit from the increased precision provided by strategic designs.

3.3 Regression Model for Individual Subjects

Performing linear regression on a right-skewed biomarker often invites a log transformation. Suppose that the log of the outcome is linearly associated with the predictor variables, so that the true model can be represented by:

$$\log(Y_{ij}) = \alpha + \mathbf{x}_{ij}\boldsymbol{\beta} + \epsilon_{ij},\ j = 1, \ldots, k_i,\ i = 1, \ldots, n, \tag{3.1}$$

where α is the intercept, $\boldsymbol{\beta}$ is the $P \times 1$ column vector of coefficients, and Y_{ij}, ϵ_{ij}, and $\mathbf{x}_{ij} = (x_{ij1}, \ldots, x_{ijP})$ are the outcome, error, and row vector of covariates for the j^{th} subject in the i^{th} pool, respectively. Furthermore, let $N = \sum_{i=1}^{n} k_i$ denote the total number of subjects, where k_i represents the number of specimens in pool i (i.e., pool size). The ϵ_{ij}'s are assumed independent and identically distributed with $E(\epsilon_{ij}) = 0$ and $Var(\epsilon_{ij}) = \sigma^2$. If the value of each individual's outcome were known, a straightforward application of multiple linear regression (MLR) on the log-transformed outcome would yield the desired parameter estimates. Similarly, if n individual specimens are selected for analysis, the same MLR estimation procedure could be applied to this subset of the full data. When specimens are pooled, however, only the measured value of the pool is known, while each specimen's outcome (Y_{ij}) remains unobserved, so that a simple application of (3.1) to the pooled measurements might not be appropriate. Details on the various pooling strategies considered are reviewed in Section 3.8. For now, we consider methods for analyzing data based on specimens that have already been pooled, where our objective is valid and efficient

estimation of $\boldsymbol{\beta}$.

3.4 Naive Model for Pooled Data

A natural inclination when faced with analyzing pooled, right-skewed data may be to perform linear regression on a log-transformation of the measured values of each pool:

$$\textbf{Naive Model}: \quad \log(Y_i^p) = \alpha + \mathbf{x}_i\boldsymbol{\beta} + \delta_i^{(1)},$$

where Y_i^p and $\mathbf{x}_i = (\bar{x}_{i\bullet 1}, \ldots, \bar{x}_{i\bullet P})$ are the measured outcome and vector of predictors for pool i, respectively, such that $\bar{x}_{i\bullet p} = \frac{1}{k_i}\sum_{j=1}^{k_i} x_{ijp}$ is the arithmetic mean of the p^{th} predictor across all specimens in pool i, and we assume that each $Y_i^p = \frac{1}{k_i}\sum_{j=1}^{k_i} Y_{ij}$ reflects the average of the individual concentrations among specimens constituting that pool as determined by laboratory assay.

In order to apply the method of least squares, let us initially assume that the expectation and variance of $\delta_i^{(1)}$, the error term under this pooled model, is preserved from the unpooled, true model (3.1), so that $E(\delta_i^{(1)}) = 0$ and $Var(\delta_i^{(1)}) = \sigma^2$. Our parameter estimate for $\boldsymbol{\theta}_1 = (\alpha, \boldsymbol{\beta})$ under the naive model is then:

$$\hat{\boldsymbol{\theta}}_1 = (\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T(\log \mathbf{Y}^p),$$

where $\mathbf{Y}^p = (Y_1^p, \ldots, Y_n^p)'$ is the vector of observed, pooled outcomes, and $\mathbf{X}_1 = (\mathbf{1} \quad \mathbf{X})$, where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_P)$ is the $n \times P$ matrix of pool-wise covariate vectors and $\mathbf{1}$ is the $n \times 1$ column vector of ones. To determine the expectation of $\hat{\boldsymbol{\theta}}_1$, we need to evaluate $E(\log \mathbf{Y}^p)$, where this expectation is conditional on $\mathbf{X}$. This expectation is not defined by the model assumptions, due to the non-linearity of the log function:

$$E(\log Y_i^p) = E\left[\log\left(\frac{1}{k_i}\sum_{j=1}^{k_i} Y_{ij}\right)\right] \neq \frac{1}{k_i}\sum_{j=1}^{k_i} E(\log Y_{ij}).$$

We can, however, approximate this value with a second-order Taylor series expansion:

$$\begin{aligned} E(\log Y_i^p) &\approx E\left\{\log[E(Y_i^p)] + \frac{Y_i^p - E(Y_i^p)}{E(Y_i^p)} - \frac{[Y_i^p - E(Y_i^p)]^2}{2E(Y_i^p)^2}\right\} \\ &= \log[E(Y_i^p)] - \frac{Var(Y_i^p)}{2E(Y_i^p)^2} \\ &= \log\left[\frac{1}{k_i}\sum_{j=1}^{k_i} E(Y_{ij})\right] - \frac{k_i^{-2}\left[\sum_{j=1}^{k_i} Var(Y_{ij})\right]}{2k_i^{-2}\left[\sum_{j=1}^{k_i} E(Y_{ij})\right]^2}, \end{aligned} \tag{3.2}$$

where $Y_{ij} = e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}+\epsilon_{ij}}$ from (3.1) and $Var(Y_{ij})$ is implicitly conditional on $\mathbf{x}_{ij}$. Now, if pools are $\mathbf{x}$-homogeneous, such that $\mathbf{x}_{ij} = \mathbf{x}_i$ for all $j = 1, \ldots, k_i$, then (3.2) reduces to:

$$\begin{aligned} E(\log Y_i^p) &\approx \alpha + \mathbf{x}_i\boldsymbol{\beta} + \log\left[\frac{1}{k_i}\sum_{j=1}^{k_i} E(e^{\epsilon_{ij}})\right] - \frac{\sum_{j=1}^{k_i} Var(e^{\epsilon_{ij}})}{2\left[\sum_{j=1}^{k_i} E(e^{\epsilon_{ij}})\right]^2} \\ &= \alpha + \mathbf{x}_i\boldsymbol{\beta} + \log(a) - \frac{c}{2k_i}, \end{aligned} \tag{3.3}$$

where $a = E(e^{\epsilon_{11}})$ and $c = Var(e^{\epsilon_{11}})/E(e^{\epsilon_{11}})^2$. This last step is based on the assumption that the ϵ_{ij}'s are independent and identically distributed. Let $\mathbf{K} = diag(k_1, \ldots, k_n)$ denote the diagonal matrix with (i,i) element equal to k_i $(i = 1, \ldots, n)$. Then (3.3) can be written in matrix form as:

$$E(\log \mathbf{Y}^p) \approx \mathbf{1}(\alpha + \log a) + \mathbf{X}\boldsymbol{\beta} - (c/2)\mathbf{K}^{-1}\mathbf{1} \tag{3.4}$$

where $\mathbf{K}^{-1}\mathbf{1} = (k_1^{-1}, \ldots, k_n^{-1})^T$ is the column vector of inverted pool sizes. Then the expectation of $\hat{\boldsymbol{\theta}}_1$ is approximately:

$$\begin{aligned} E(\hat{\boldsymbol{\theta}}_1) = E\begin{pmatrix} \hat{\alpha} \\ \hat{\boldsymbol{\beta}} \end{pmatrix} &\approx (\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T E(\log \mathbf{Y}^p) \\ &= (\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T\left[\mathbf{X}_1\begin{pmatrix} \alpha + \log(a) \\ \boldsymbol{\beta} \end{pmatrix} - (c/2)\mathbf{K}^{-1}\mathbf{1}\right] \\ &= \begin{pmatrix} \alpha + \log(a) \\ \boldsymbol{\beta} \end{pmatrix} - (c/2)(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T\mathbf{K}^{-1}\mathbf{1}, \end{aligned}$$

Thus, both $\hat{\alpha}$ and $\hat{\boldsymbol{\beta}}$ are expected to be biased under the Naive Model.

3.5 Approximate Model for Pooled Data

To mitigate the potential bias induced by the vector of inverted pool sizes, $\mathbf{K}^{-1}\mathbf{1}$, we can incorporate pool size into the regression model:

$$\textbf{Approximate Model}: \quad \log(Y_i^p) = \alpha + \gamma k_i^{-1} + \mathbf{x}_i\boldsymbol{\beta} + \delta_i^{(2)},$$

where γ is the regression coefficient corresponding to k_i^{-1} and $\delta_i^{(2)}$ represents the error term for pool i under this Approximate Model, where we are still working under the assumption of $\mathbf{x}$-homogeneous pools, i.e. $\mathbf{x}_{ij} = \mathbf{x}_i$ for all $j = 1, \ldots, k_i$. Note that when all pool sizes are equal, this model essentially reduces to the Naive Model.

Before performing least squares regression based on this new model, we can approximate the variance of the log-transformed pooled outcomes to determine whether a weighted least squares approach could improve efficiency. Using a first-order Taylor series expansion,

$$Var(\log Y_i^p) \approx Var\left\{\log[E(Y_i^p)] + \frac{Y_i^p - E(Y_i^p)}{E(Y_i^p)}\right\} = \frac{Var(Y_i^p)}{E(Y_i^p)^2} = \frac{c}{k_i}. \tag{3.5}$$

Since the variance of each pooled outcome is a function of pool size, efficiency could potentially be improved by applying a weighted least squares (WLS) regression with weight matrix $\mathbf{K} = diag(k_1, \ldots, k_n)$. The WLS parameter estimate for $\boldsymbol{\theta}_2 = (\alpha, \gamma, \boldsymbol{\beta})$ under the Approximate Model is then:

$$\hat{\boldsymbol{\theta}}_2 = (\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}(\log \mathbf{Y}^p),$$

where $\mathbf{X}_2 = (\mathbf{1} \quad \mathbf{K}^{-1}\mathbf{1} \quad \mathbf{X})$. Applying the Taylor series approximation from (3.4), the

expectation of $\hat{\boldsymbol{\theta}}_2$ is approximately:

$$\begin{aligned} E(\hat{\boldsymbol{\theta}}_2) &= E\begin{pmatrix} \hat{\alpha} \\ \hat{\gamma} \\ \hat{\boldsymbol{\beta}} \end{pmatrix} = (\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}[E(\log \mathbf{Y}^p)] \\ &\approx (\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}\left[\mathbf{1}(\alpha + \log a) - (c/2)\mathbf{K}^{-1}\mathbf{1} + \mathbf{X}\boldsymbol{\beta}\right] \\ &= (\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2\begin{pmatrix} \alpha + \log(a) \\ -c/2 \\ \boldsymbol{\beta} \end{pmatrix} \\ &= \begin{pmatrix} \alpha + \log(a) \\ -c/2 \\ \boldsymbol{\beta} \end{pmatrix}. \end{aligned}$$

Under this model, $\hat{\alpha}$ remains biased by a factor of $\log(a)$, $\hat{\gamma}$ is an approximately unbiased estimator of $-c/2$, and $\hat{\boldsymbol{\beta}}$ will be an approximately unbiased estimator of the original coefficient vector $\boldsymbol{\beta}$. Furthermore, $\hat{Var}(\hat{\boldsymbol{\theta}}_2) = \hat{c}(\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}$, the estimated variance of $\hat{\boldsymbol{\theta}}_2$, is approximately unbiased as well, where $\hat{c}$ is the usual WLS estimate of the variance of the outcome. To see this, let $\mathbf{B} = (\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}$. Following standard WLS theory, the variance of $\hat{\boldsymbol{\theta}}_2 = (\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}(\log \mathbf{Y}^p)$ is:

$$Var(\hat{\boldsymbol{\theta}}_2) = Var(\mathbf{B}\log \mathbf{Y}^p) = \mathbf{B}Var(\log \mathbf{Y}^p)\mathbf{B}^{\mathbf{T}} \approx c\mathbf{B}\mathbf{K}^{-1}\mathbf{B}^{\mathbf{T}} = c(\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}$$

where the approximation $Var(\log \mathbf{Y}^p) \approx c\mathbf{K}^{-1}$ is from (3.5). Now, let $v_E = n - rank(\mathbf{X}_2)$ and $\mathbf{A} = \mathbf{K}\mathbf{X}_2(\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}$. Again following standard WLS procedures, the estimate of this variance is $\hat{Var}(\hat{\boldsymbol{\theta}}_2) = \hat{c}(\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}$, where

$$\hat{c} = \frac{(\log \mathbf{Y}^p)^T(\mathbf{K} - \mathbf{A})(\log \mathbf{Y}^p)}{v_E},$$

which will be approximately unbiased if $E(\hat{c}) \approx c$. Let $\mathbf{X}_2\boldsymbol{\theta}^*$ denote the true value of

$E(\log \mathbf{Y}^p)$. Then

$$\begin{aligned} E(\hat{c}) &= E\left[\frac{(\log \mathbf{Y}^p)^T(\mathbf{K}-\mathbf{A})(\log \mathbf{Y}^p)}{v_E}\right] \\ &= \frac{1}{v_E}\left\{tr[(\mathbf{K}-\mathbf{A})Var(\log \mathbf{Y}^p)] + E(\log \mathbf{Y}^p)^T(\mathbf{K}-\mathbf{A})E(\log \mathbf{Y}^p)\right\} \\ &\approx \frac{1}{v_E}\left\{tr[(\mathbf{K}-\mathbf{A})c\mathbf{K}^{-1}] + \boldsymbol{\theta}^{*T}\mathbf{X}_2^T(\mathbf{K}-\mathbf{A})\mathbf{X}_2\boldsymbol{\theta}^*\right\} \\ &= \frac{c}{v_E}\{tr(\mathbf{K}\mathbf{K}^{-1}) - tr[\mathbf{K}\mathbf{X}_2(\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}\mathbf{K}^{-1}]\} \\ &= \frac{c}{v_E}[n - rank(X_2)] \\ &= c, \end{aligned}$$

since $\mathbf{X}_2^T(\mathbf{K}-\mathbf{A})\mathbf{X}_2 = \mathbf{X}_2^T[\mathbf{K}-\mathbf{K}\mathbf{X}_2(\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}]\mathbf{X}_2 = \mathbf{0}$. Thus, the estimated variance of $\hat{\boldsymbol{\theta}}_2$ will be approximately unbiased. Furthermore, when the total number of pools (n) is large, $\hat{\boldsymbol{\theta}}_2$ will be approximately normally distributed due to asymptotic properties under the central limit theorem, since $\hat{\boldsymbol{\theta}}_2$ can be written as a sample mean:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_2 &= (\mathbf{X}_2^T\mathbf{K}\mathbf{X}_2)^{-1}\mathbf{X}_2^T\mathbf{K}(\log \mathbf{Y}^p) \\ &= \left(\sum_{i=1}^n \mathbf{x}_i^T k_i \mathbf{x}_i\right)^{-1}\left(\sum_{i=1}^n \mathbf{x}_i^T k_i \log Y_i^p\right) \\ &= \left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^T k_i \mathbf{x}_i\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^T k_i \log Y_i^p\right). \end{aligned}$$

where $\left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^T k_i \mathbf{x}_i\right)^{-1}$ is fixed. Thus, the usual 95% confidence intervals based on the normal distribution should provide nominal 95% coverage in large samples. Since this property only applies when n is large, confidence intervals may be too liberal in small samples. Thus, although the additional assumption of normality on the original errors (ϵ_{ij}) does not necessarily dictate a t-distribution for the elements of $\hat{\boldsymbol{\beta}}$, applying the standard t reference distribution with $n - P - 1$ degrees of freedom is a reasonable measure to help alleviate overly liberal confidence intervals when sample size is small.

One advantage of the Approximate Model is that specific distributional assumptions about the errors are not required, since the asymptotic normality and consistency of the WLS estimators are based on the central limit theorem. Instead, the validity of this method

depends on the independence of the individual specimens, the convergence of the Taylor series approximation, and the accuracy of the assumed linear relationship between the covariates and the log of the outcome. In Section 3.9, we demonstrate the potential repercussions of assuming the Naive Model, as well as the advantages of applying the Approximate Model to analyze **x**-homogeneous pools. The simplicity of the Approximate Model and its flexibility in not requiring any distributional assumptions are further bolstered by simulation results.

3.6 Calculating MLEs

It is not always possible to form **x**-homogeneous pools, especially if one or more of the covariates are continuous. In such cases, the Taylor series approximations from Section 3.5 are no longer justified. Instead, parametric approaches to identify MLEs of the $\boldsymbol{\beta}$ vector may be the best option. While these methods do require distributional assumptions, they provide theoretically sound alternatives to the Approximate Model when pools are heterogeneous.

A natural method to calculate MLEs is to maximize the observed data likelihood. As noted in Section 1.4.1, the density for each pool, say $f_p(Y_i^p)$, consists of a $(k_i - 1)$-fold integral

$$f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta}) = \int_{Y_{ik_i}} \cdots \int_{Y_{i2}} k_i \left[f_1 \left(k_i Y_i^p - \sum_{j=2}^{k_i} Y_{ij} \right) f_2(Y_2) \ldots f_{k_i}(Y_{k_i}) \right] dY_{i2} \ldots dY_{ik_i},$$

where $f_j(y) = f(y|\mathbf{x}_{ij}, \boldsymbol{\theta})$ is the assumed density of the individual level data that depends on the parameter vector $\boldsymbol{\theta}$ as well as the covariate vector $\mathbf{x}_{ij}$. When there are at most two specimens in each pool (i.e. $k_i \leq 2$ for all i), the observed log-likelihood can often be maximized through existing numerical integration and optimization functions such as the *optim* function in R or the NLPQN function in SAS IML (R Development Core Team, 2012; SAS Institute Inc., 2010). For larger pool sizes, however, numerical optimization of the likelihood can quickly become computationally intractable. The integrand characterizing the density of a sum of lognormal random variables, in particular, has a reputation for being especially poorly-behaved (Beaulieu and Xie, 2004; Barakat, 1976; Santos Filho et al., 2006). In sub-

sequent simulations and analyses, we apply direct optimization of the convolution formula when possible. For larger pools sizes we propose a Monte Carlo Expectation Maximization (MCEM) algorithm as a more dependable tool to optimize the observed likelihood.

3.7 MLEs via MCEM

The EM algorithm has a natural application to pooled data, since the complete data (i.e. all individual outcomes) presumably follows a distribution from which MLE calculation is simple. Similar to the traditional EM algorithm, the Monte Carlo EM algorithm seeks to maximize the expected value of the conditional log-likelihood in lieu of the observed likelihood (Dempster et al., 1977; Wei and Tanner, 1990).

Let $L_c(\boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{k_i} f(Y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\theta})$ denote the complete likelihood, where $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta}, \sigma)$ and f is the density of the unpooled Y_{ij}'s. In this scenario, we view the vector of measured, pooled outcomes, $\mathbf{Y}^p = (Y_1^p, \ldots, Y_n^p)$, as the observed data and the vector of individual outcomes, $\mathbf{Y} = (\{Y_{ij}\} : j = 1, \ldots, k_i, i = 1, \ldots, n)$, as the missing data, with the restriction $\sum_{j=1}^{k_i} Y_{ij} = k_i Y_i^p$. Applying this restriction, the missing data in pool i is essentially reduced to $(Y_{i2}, \ldots, Y_{ik_i})$, since, given Y_i^p, $Y_{i1} = k_i Y_i^p - \sum_{j=2}^{k_i} Y_{ij}$.

3.7.1 E step

The Expectation step of the algorithm requires calculation of the expected value of the complete log-likelihood given the observed data. Let $g(Y_{i2}, \ldots, Y_{ik_i}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ denote the density of the missing data given the observed (pooled) data under the parameter vector $\boldsymbol{\theta}^{(t)}$ and fully observed covariate data $\mathbf{X}$ for each $i = 1, \ldots, n$. Then the expected conditional log-likelihood is:

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= E[\log L_c(\boldsymbol{\theta})|\mathbf{Y}^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}] \\ &= \sum_{i=1}^{n} E_g \left\{ \log f \left[\left(k_i Y_i^p - \sum_{j=2}^{k_i} Y_{ij} \right) |\mathbf{x}_{i1}, \boldsymbol{\theta} \right] + \sum_{j=2}^{k_i} \log f(Y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\theta}) \right\} \quad (3.6) \end{aligned}$$

where $\boldsymbol{\theta}^{(t)} = (\alpha^{(t)}, \boldsymbol{\beta}^{(t)}, \sigma^{(t)})$ is the estimate of the parameter vector at the t^{th} iteration of the algorithm. Let $h(\mathbf{Y}_{i[-1]})$ represent any of the continuous functions of the missing

data contained in (3.6), where $\mathbf{Y}_{i[\text{-}1]} = (Y_{i2}, \ldots, Y_{ik_i})$. For data that follows a right-skewed distribution (e.g. lognormal), finding a closed form expression for $E_g[h(\mathbf{Y}_{i[\text{-}1]})]$ can be difficult. In such cases, we recommend using Monte Carlo methods to approximate this value.

3.7.2 Monte Carlo Estimation

By the weak law of large numbers (WLLN), the conditional expectation $E_g[h(\mathbf{Y}_{i[\text{-}1]})]$ can be estimated by the Monte Carlo approximation:

$$E_g\left[h(\mathbf{Y}_{i[\text{-}1]})\right] = \int_{Y_{ik_i}} \cdots \int_{Y_{i2}} h\left(\mathbf{Y}_{i[\text{-}1]}\right) g\left(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}\right) d\mathbf{Y}_{i[\text{-}1]} \approx \frac{1}{M} \sum_{m=1}^{M} h\left(\mathbf{Y}_{i[\text{-}1],m}\right),$$

where $\mathbf{Y}_{i[\text{-}1],m} = (Y_{i2,m}, \ldots, Y_{ik_i,m})$ is generated under the joint conditional distribution $g(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ for each m, and M is a number large enough for the asymptotic properties of the WLLN to hold. Now, we can re-write $g(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ as:

$$g(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) = \frac{f_c(\mathbf{Y}_{i[\text{-}1]}, Y_i^p | \mathbf{X}, \boldsymbol{\theta}^{(t)})}{f_p(Y_i^p | \mathbf{X}, \boldsymbol{\theta}^{(t)})}$$

where f_p is the marginal density for Y_i^p and $f_c(\mathbf{Y}_{i[\text{-}1]}, Y_i^p | \mathbf{X}, \boldsymbol{\theta}^{(t)})$ denotes the density of the complete data. Using the joint transformation approach, let $U_j = Y_{ij}$ for $j = 2, \ldots, k_i$ and let $V = Y_i^p = \frac{1}{k_i} \sum_{j=1}^{k_i} Y_{ij}$, so that $Y_{i1} = k_i V - \sum_{j=2}^{k_i} U_j$. Then

$$\begin{aligned} f(V, U_2, \ldots, U_{k_i}) &= |J| \times f[Y_{i1}(v), Y_{i2}(u_2) \ldots, Y_{ik_i}(u_{k_i})] \\ &= k_i f\left(k_i V - \sum_{j=2}^{k_i} U_j | \mathbf{x}_{i1}, \boldsymbol{\theta}^{(t)}\right) \prod_{j=2}^{k_i} f(U_j | \mathbf{x}_{ij}, \boldsymbol{\theta}^{(t)}) \end{aligned}$$

where the Jacobian J, is calculated as:

$$|J| = \left|\frac{d\mathbf{Y}}{d(V, \mathbf{U})}\right| = \begin{vmatrix} k_i & -1 & -1 & \ldots & -1 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{vmatrix} = k_i$$

Then

$$g(\mathbf{Y}_{i[\text{-}1]}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) \tag{3.7}$$
$$= \frac{k_i I(\sum_{j=2}^{k_i} Y_{ij} < k_i Y_i^p) f(k_i Y_i^p - \sum_{j=2}^{k_i} Y_{ij}|\mathbf{x}_{i1}, \boldsymbol{\theta}^{(t)}) \prod_{j=2}^{k_i} f(Y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\theta}^{(t)})}{f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta}^{(t)})},$$

where we have incorporated the linear restriction on the Y_{ij}'s into the density expression as an indicator function. The main difficulty in generating data from (3.7) is meeting the inequality constraint contained in the indicator function.

Rejection Sampling A straightforward method toward generating data from the conditional distribution $g(\mathbf{Y}_{i[\text{-}1]}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ is to first generate each $Y_{ij,m}$ from $f(Y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\theta}^{(t)})$ for $j = 2, \ldots, k_i$. If these simulated data fail to meet the restriction (i.e. if $\sum_{j=2}^{k_i} Y_{ij,m} \geq k_i Y_i^p$), then the sample is rejected and a new sample is generated. This process continues until the desired number of samples, M, have been produced. As expected, this method becomes increasingly slow with larger values of M, particularly when the distributional variance or pool size are large.

Importance Sampling A more computationally efficient method than rejection sampling can be achieved through importance sampling (Lange, 2010). The basic idea behind this strategy is to identify a distribution that is similar to $g(\mathbf{Y}_{i[\text{-}1]}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ but from which samples are easier to obtain. Importance weights are then applied to the Monte Carlo estimate of $E_g[h(\mathbf{Y}_{i[\text{-}1]})]$ in order to account for generating data under the alternate distribution. Let $g^*(\mathbf{Y}_{i[\text{-}1]}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ represent this alternate generating distribution.

Then:

$$
\begin{aligned}
E_g[h(\mathbf{Y}_{i[\text{-}1]})] &= \int_{Y_{ik_i}} \cdots \int_{Y_{i2}} h(\mathbf{Y}_{i[\text{-}1]}) g(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) d\mathbf{Y}_{i[\text{-}1]} \\
&= \frac{\int_{Y_{ik_i}} \cdots \int_{Y_{i2}} h(\mathbf{Y}_{i[\text{-}1]}) g(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) d\mathbf{Y}_{i[\text{-}1]}}{\int_{Y_{ik_i}} \cdots \int_{Y_{i2}} g(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) d\mathbf{Y}_{i[\text{-}1]}} \\
&= \frac{\int_{Y_{ik_i}} \cdots \int_{Y_{i2}} h(\mathbf{Y}_{i[\text{-}1]}) w(\mathbf{Y}_{i[\text{-}1]}) g^*(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) d\mathbf{Y}_{i[\text{-}1]}}{\int_{Y_{ik_i}} \cdots \int_{Y_{i2}} w(\mathbf{Y}_{i[\text{-}1]}) g^*(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) d\mathbf{Y}_{i[\text{-}1]}} \\
&\approx \frac{\sum_{m=1}^{M} h(\mathbf{Y}_{i[\text{-}1],m}) w(\mathbf{Y}_{i[\text{-}1],m})}{\sum_{m=1}^{M} w(\mathbf{Y}_{i[\text{-}1],m})}
\end{aligned}
$$

where $w(\mathbf{Y}_{i[\text{-}1]}) = g(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) / g^*(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ are the importance weights, and each $\mathbf{Y}_{i[\text{-}1],m} = (Y_{i2,m}, \ldots, Y_{ik_i,m})$ is now generated under the alternate distribution, $g^*(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$. The first step in this derivation was based on the property that the integral of a density function over its entire domain is 1. The advantage of including this denominator is that any constants (e.g., functions of Y_i^p not dependent on m) will cancel in the final step.

Since the linear restriction $\sum_{j=2}^{k_i} Y_{ij,m} < k_i Y_i^p$ poses the main difficulty, we can choose a distribution $g^*(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ that satisfies this restriction first, then identify the appropriate weights to obtain a good approximation of the desired expectation. While there may be multiple candidates for $g^*(\mathbf{Y}_{i[\text{-}1]} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$, one straightforward way to guarantee the linear restriction is to first generate each $Y_{ij,m}$ from $f(Y_{ij} | \mathbf{x}_{ij}, \boldsymbol{\theta}^{(t)})$ for $j = 1, \ldots, k_i$. This step is similar to the first step of the rejection sampling method, except that this time we are also generating Y_{i1}. Then, instead of rejecting the sample if it does not meet the linear constraint, we alter the sample so that it automatically does. Let $\mathbf{Z}_{i,m} = (Z_{i1,m}, \ldots, Z_{ik_i,m})$ denote this new sample, where $Z_{ij,m} = Y_{ij,m}(k_i Y_i^p) / (\sum_{j=1}^{k_i} Y_{ij,m})$ for all $j = 1, \ldots, k_i$. This strategy guarantees that the new sample will sum to $k_i Y_i^p$, so that exactly M samples need to be generated at each iteration. Rejection sampling, on the other hand, requires a minimum of M samples, and often many more, depending on how often the linear constraint is met. Table 3.1 illustrates the potential computational savings of performing importance

sampling over rejection sampling under a lognormal distribution. The results are based on a simulation study of 30,000 repetitions, where an outcome (Y) was simulated under a lognormal distribution such that $E[\log(Y)] = 0$ and $Var[\log(Y)] = 1$, then were grouped in pairs to mimic a random pooling strategy with pool size = 2. Given these simulated pooled measurements, both rejection and importance sampling methods were employed at various values of M.

Table 3.1: Computational efficiency of importance sampling vs. rejection sampling. M_{RS} refers to the average number of samples generated under rejection sampling before M are accepted, M_{IS} denotes the number of samples generated under importance sampling. t_{RS} and t_{IS} refers to the average time in seconds of the computational time required to perform rejection and importance sampling, respectively, with a Monte Carlo size of M.

	Monte Carlo Size (M)			
	10	50	100	500
M_{IS}	10	50	100	500
M_{RS}	15	74	147	740
t_{RS}/t_{IS}	3.8	14.7	24.3	52.8

As evident by this study, the average amount of additional time required to perform rejection sampling instead of importance sampling increases noticeably with larger values of M, since this method must generate around 50% more samples in order to obtain the desired Monte Carlo size, whereas importance sampling generates exactly M samples for each pool. When parameters such as variance and pool size are increased, this discrepancy is even more pronounced. Although importance sampling requires a fair amount of initial effort in calculating appropriate weights, enormous computational savings can be accrued by performing importance sampling over rejection sampling, thus facilitating more rapid convergence.

To calculate the importance weights, we must first determine the appropriate expression for $g^*(\mathbf{Z}_{i[\text{-}1],m}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$. Following the derivation outlined in Frigyik, Kapila, and Gupta (2010), the joint density of $(Z_{i1,m}, \ldots, Z_{ik_i,m})$ can be found by applying the change of variable:

$$(Y_{i1,m}, \ldots, Y_{ik_i,m}) = \left[S\left(k_i Y_i^p - \sum_{j=2}^{k_i} Z_{ij,m} \right), SZ_{i2,m}, \ldots, SZ_{ik_i,m} \right],$$

where S is defined as $S = (\sum_{j=1}^{k_i} Y_{ij,m})/k_i Y_i^p$. Then the joint distribution of $(S, \mathbf{Z}_{i[\text{-}1],m})$ is:

$$g_s(S, \mathbf{Z}_{i[\text{-}1],m} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}) = \\ |J| \times f\left[S\left(k_i Y_i^p - \sum_{j=2}^{k_i} Z_{ij,m} \right) | Y_i^p, \mathbf{x}_{i1}, \boldsymbol{\theta}^{(t)} \right] \prod_{j=2}^{k_i} f(S Z_{ij,m} | Y_i^p, \mathbf{x}_{ij}, \boldsymbol{\theta}^{(t)}), \qquad (3.8)$$

where the determinant of the Jacobian, $|J|$, is calculated as:

$$\begin{aligned} |J| &= \left| \frac{d\mathbf{Y}}{d(S, \mathbf{Z})} \right| \\ &= \begin{vmatrix} k_i Y_i^p - \sum_{j=2}^{k_i} Z_{ij,m} & -S & -S & \dots & -S \\ Z_{i2,m} & S & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ Z_{ik_i,m} & 0 & 0 & \dots & S \end{vmatrix} \\ &= \left(k_i Y_i^p - \sum_{j=2}^{k_i} Z_{ij,m} \right) S^{k_i - 1} + S^{k_i - 1} \sum_{j=2}^{k_i} Z_{i2,m} \\ &= k_i Y_i^p S^{k_i - 1} \end{aligned}$$

Integrating over the domain of S will give the joint density of the alternate generating function $g^*(\mathbf{Z}_{i[\text{-}1],m} | Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$. If this expression has a closed form, calculation of the importance weights is straightforward. Examples of this process are given in Section 3.7.5 for a lognormally distributed outcome, and in Section 4.2.2 for a gamma-distributed outcome.

3.7.3 M step

Maximizing Q with respect to $\boldsymbol{\theta}$ is generally a straightforward task once the conditional expectations have been approximated. This is particularly true when the assumed distribution of the outcome is a member of the exponential family, for which there are numerous maximization functions available in software packages (e.g. SAS, R).

3.7.4 Standard Error Estimation

One of the drawbacks of using an EM type algorithm is that calculating standard error estimates can prove difficult. While various methods have been proposed to estimate the observed information matrix (Jamshidian and Jennrich, 2000; Oakes, 1999; Louis, 1982), for this study, since Monte Carlo techniques are required to approximate the conditional expectations of the MLEs, we apply MC approximations to strategies similar to Louis's method to estimate the observed information matrix (Louis, 1982).

Let $l_{obs} = \sum_{i=1}^{n} \log f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta})$ denote the observed log-likelihood, $f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}|\mathbf{X}, \boldsymbol{\theta})$ the density of the complete data for pool i, and $Q_i = E_g[\log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}|\mathbf{X}, \boldsymbol{\theta})]$, the i^{th} component of Q. Then the Hessian can be written as:

$$\frac{d^2 l_{obs}}{d\boldsymbol{\theta}^2} = \sum_{i=1}^{n} \left\{ \frac{1}{f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta})} \left[\frac{d^2}{d\boldsymbol{\theta}^2} f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta}) \right] - \left[\frac{d}{d\boldsymbol{\theta}} \log f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta}) \right] \left[\frac{d}{d\boldsymbol{\theta}} \log f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta}) \right]^T \right\}, \tag{3.9}$$

where

$$\begin{aligned}
\frac{d}{d\boldsymbol{\theta}} \log f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta}) &= [f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta})]^{-1} \left[\frac{d}{d\boldsymbol{\theta}} f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta}) \right] \\
&= [f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta})]^{-1} \int_{Y_{ik_i}} \cdots \int_{Y_{i2}} \frac{d}{d\boldsymbol{\theta}} f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}|\mathbf{X}, \boldsymbol{\theta}) d\mathbf{Y}_{i[\text{-}1]} \\
&= \int_{Y_{ik_i}} \cdots \int_{Y_{i2}} \frac{\frac{d}{d\boldsymbol{\theta}} f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}|\mathbf{X}, \boldsymbol{\theta})}{f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}|\mathbf{X}, \boldsymbol{\theta})} \frac{f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}|\mathbf{X}, \boldsymbol{\theta})}{f_p(Y_i^p|\mathbf{X}, \boldsymbol{\theta})} d\mathbf{Y}_{i[\text{-}1]} \\
&= \int_{Y_{ik_i}} \cdots \int_{Y_{i2}} \left[\frac{d}{d\boldsymbol{\theta}} \log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}|\mathbf{X}, \boldsymbol{\theta}) \right] g(\mathbf{Y}_{i[\text{-}1]}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}) d\mathbf{Y}_{i[\text{-}1]} \\
&= E_g \left[\frac{d}{d\boldsymbol{\theta}} \log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}|\mathbf{X}, \boldsymbol{\theta}) \right] \\
&= \frac{dQ_i}{d\boldsymbol{\theta}},
\end{aligned}$$

and $[f_p(Y_i^p|\mathbf{X},\boldsymbol{\theta})]^{-1}\frac{d^2}{d\boldsymbol{\theta}^2}f_p(Y_i^p|\mathbf{X},\boldsymbol{\theta})$ can be written as:

$$\begin{aligned}
&= \int_{Y_{ik_i}}\cdots\int_{Y_{i2}}\left[\frac{d^2}{d\boldsymbol{\theta}^2}f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]}|\mathbf{X},\boldsymbol{\theta})\right][f_p(Y_i^p|\mathbf{X},\boldsymbol{\theta})]^{-1}d\mathbf{Y}_{i[\text{-}1]}\\
&= \int_{Y_{ik_i}}\cdots\int_{Y_{i2}}\frac{\frac{d^2}{d\boldsymbol{\theta}^2}f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]}|\mathbf{X},\boldsymbol{\theta})}{f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]}|\mathbf{X},\boldsymbol{\theta})}g(\mathbf{Y}_{i[\text{-}1]}|Y_i^p,\mathbf{X},\boldsymbol{\theta})d\mathbf{Y}_{i[\text{-}1]}\\
&= E_g\left[\frac{\frac{d^2}{d\boldsymbol{\theta}^2}f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]}|\mathbf{X},\boldsymbol{\theta})}{f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]}|\mathbf{X},\boldsymbol{\theta})}\right]\\
&= E_g\left\{\frac{d^2}{d\boldsymbol{\theta}^2}\log f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]})+\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]})\right]\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]})\right]^T\right\}\\
&= \frac{d^2Q_i}{d\boldsymbol{\theta}^2}+E_g\left\{\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]})\right]\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p,\mathbf{Y}_{i[\text{-}1]})\right]^T\right\}.
\end{aligned}$$

Monte Carlo methods can be used to approximate (3.9), which can then be inverted to give the negative of the variance-covariance matrix of the MLEs (Tan et al., 2007).

3.7.5 Example: Lognormal Distribution

While the MCEM methods previously outlined could theoretically be applied to any parametric model, we provide explicit calculations of these steps for a lognormal model for pooled data, since right-skewed outcomes are often assumed to be lognormally distributed in regression settings.

E step Applying the lognormal density to (3.6), the conditional expectation of the complete log-likelihood becomes:

$$\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= c(\mathbf{Y})-N\log\sigma-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{k_i}E_g[(\log Y_{ij}-\mu_{ij})^2]\\
&= c(\mathbf{Y})-N\log\sigma-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{k_i}\left[h_2(Y_{ij})-2\mu_{ij}h_1(Y_{ij})+\mu_{ij}^2\right] \qquad (3.10)
\end{aligned}$$

where $Y_{i1}=k_iY_i^p-\sum_{j=2}^{k_i}Y_{ij}$, $\mu_{ij}=\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}$, $c(\mathbf{Y})=-E_g[\sum\sum\log(Y_{ij}\sqrt{2\pi})]$,and $h_b(Y_{ij})=E_g[(\log Y_{ij})^b]$ for $b=1,2$. To estimate h_1 and h_2 we apply the importance sampling method outlined in Section 3.7.2. For a lognormally distributed outcome, integrating over S in (3.8)

gives a closed from for the alternate sampling distribution $g^*(\mathbf{Z}_{i[\text{-}1],m}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$:

$$\begin{aligned} g^* &= \int_0^\infty k_i Y_i^p s^{k_i-1} \frac{(2\pi\sigma^2)^{-k_i/2}}{\prod_{j=1}^{k_i}(s z_{ij,m})} \exp\left[-\frac{\sum_{j=1}^{k_i}(\log s + \log z_{ij,m} - \mu_{ij})^2}{2\sigma^2}\right] ds \\ &= k_i Y_i^p \left(\prod_{j=1}^{k_i} f_{ij}\right) \int_0^\infty s^{-1} \exp\left[-\frac{(\log s)^2 + 2(\log s)k_i^{-1}\sum_{j=1}^{k_i}(\log z_{ij,m} - \mu_{ij})}{2\sigma^2 k_i^{-1}}\right] ds \\ &= k_i Y_i^p \left(\prod_{j=1}^{k_i} f_{ij}\right) \exp\left\{\frac{1}{2\sigma^2 k_i}\left[\sum_{j=1}^{k_i}(\log z_{ij,m} - \mu_{ij})\right]^2\right\} \sqrt{\frac{2\pi\sigma^2}{k_i}} \end{aligned}$$

where $f_{ij} = f(z_{ij,m}|\mathbf{x}_{ij}, \boldsymbol{\theta})$ is the lognormal density and $z_{i1,m} = k_i Y_i^p - \sum_{j=2}^{k_i} z_{ij,m}$. The importance weights are then:

$$\begin{aligned} w(\mathbf{Z}_{i[\text{-}1],m}) &= \frac{g(\mathbf{Z}_{i[\text{-}1],m}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})}{g^*(\mathbf{Z}_{i[\text{-}1],m}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})} \\ &= c(Y_i^p) \frac{I(\sum_{j=2}^{k_i} z_{ij,m} < k_i Y_i^p)\left(\prod_{j=1}^{k_i} f_{ij}\right)}{\left(\prod_{j=1}^{k_i} f_{ij}\right)} \exp\left\{-\frac{\left[\sum_{j=1}^{k_i}(\log z_{ij,m} - \mu_{ij})\right]^2}{2\sigma^2 k_i}\right\} \\ &= c(Y_i^p) \exp\left\{-\frac{\left[\sum_{j=1}^{k_i}(\log z_{ij,m} - \mu_{ij})\right]^2}{2\sigma^2 k_i}\right\}, \end{aligned} \tag{3.11}$$

where $c(Y_i^p)$ is a function of the observed data that does not depend on m. Note that $I(\sum_{j=2}^{k_i} z_{ij,m} < k_i Y_i^p) = 1$ since the $\mathbf{Z}_{i[\text{-}1],m}$ are designed specifically to fulfill this criterion. Then each $h_b(Y_{ij}) = E_g[(\log Y_{ij})^b]$ for $b = 1, 2$ in (3.10) can be approximated by:

$$\begin{aligned} h_b(Y_{ij}) &\approx \frac{\sum_{m=1}^M (\log z_{ij,m})^b w(\mathbf{Z}_{i[\text{-}1],m})}{\sum_{m=1}^M w(\mathbf{Z}_{i[\text{-}1],m})} \\ &= \frac{\sum_{m=1}^M (\log z_{ij,m})^b \exp\left\{-\left[\sum_{j=1}^{k_i}(\log z_{ij,m} - \mu_{ij})\right]^2 (2\sigma^2 k_i)^{-1}\right\}}{\sum_{m=1}^M \exp\left\{-\left[\sum_{j=1}^{k_i}(\log z_{ij,m} - \mu_{ij})\right]^2 (2\sigma^2 k_i)^{-1}\right\}}, \end{aligned}$$

where $\mathbf{Z}_{i[\text{-}1],m}$ is generated under g^*. Note that $c(Y_i^p)$ from (3.11) cancels from this approximation since this expression does not depend on m.

Several strategies for choosing the best values of M at each iteration have been explored (Booth and Hobert, 1999; Caffo, Jank, and Jones, 2005; Levine and Casella, 2001; Tan,

Tian, and Fang, 2007; Wei and Tanner, 1990). These strategies consist of starting with a small value for M at early iterations, then increasing at higher iterations. This technique, referred to as "ascent-based MCEM", serves to quickly move the algorithm to the appropriate neighborhood of the MLEs, then gradually reduces the error associated with the Monte Carlo estimation as the algorithm stabilizes. Theoretically, Monte Carlo error could be virtually eliminated as M approaches infinity. Often, however, moderately large values of M will suffice (e.g. $M \approx 10{,}000$). For the MCEM algorithm applied to the simulations in this paper, we speed convergence by calculating starting values under the Approximate Model, which will often give good approximations of the neighborhood of the coefficient estimates, even when the assumptions required for the validity of the Approximate Model are not met. After obtaining these starting values, we set $M = 50$, and after every 20 iterations, M is increased by 25%. For simulations presented in this paper, the algorithm was run for 500 iterations, since additional iterations produced only negligible changes in the parameter estimates.

M step Closed form solutions for the update formulas of α, $\boldsymbol{\beta}$ and σ can be found by solving for the roots of the gradient vector. The update formulas for these parameters are given by:

$$\begin{aligned}
\begin{pmatrix} \hat{\alpha}^{(t+1)} \\ \hat{\boldsymbol{\beta}}^{(t+1)} \end{pmatrix} &= (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{X}_1^T \mathbf{h}_1(\mathbf{Y}) \\
\hat{\sigma}^{(t+1)} &= \left\{ \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{k_i} \left[h_2(Y_{ij}) - 2\hat{\mu}_{ij}^{(t)} h_1(Y_{ij}) + (\hat{\mu}_{ij}^{(t)})^2 \right] \right\}^{1/2}
\end{aligned}$$

where $\mathbf{X}_1 = (\mathbf{1} \;\; \mathbf{X})$ is the design matrix, $\mathbf{X}$ is the matrix of fully-observed covariate data, $\mathbf{h}_1(\mathbf{Y}) = E_g(\log \mathbf{Y})$, and $\hat{\mu}_{ij}^{(t)} = \hat{\alpha}^{(t)} + \mathbf{x}_{ij} \hat{\boldsymbol{\beta}}^{(t)}$. Note that these parameter estimates from the t^{th} iteration are also embedded in $h_1(Y_{ij})$ and $h_2(Y_{ij})$, since these values were approximated based on data generated from densities conditional on $\boldsymbol{\theta}^{(t)}$.

Standard Error Estimation After calculating an expression for Q, estimating the information matrix is fairly straightforward. Referring back to Section 3.7.4, the observed

information can be written as

$$\frac{d^2 l_{obs}}{d\boldsymbol{\theta}^2} = \frac{d^2 Q}{d\boldsymbol{\theta}^2} - \left(\frac{dQ}{d\boldsymbol{\theta}}\right)\left(\frac{dQ}{d\boldsymbol{\theta}}\right)^T + \sum_{i=1}^{n} E_g\left\{\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]})\right]\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]})\right]^T\right\}. \tag{3.12}$$

Existing software functions, such as the *hessian* and *grad* functions in the R package "numDeriv", can be employed to numerically calculate the first two terms in (3.12) once Q_i has been evaluated (via MC methods) for each i at the final iteration of the MCEM algorithm. An expression for the last component, however, must be developed under the assumed distribution. The gradient of the complete log-likelihood under the lognormal distribution is:

$$\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}) = \begin{pmatrix} \sigma^{-2}\sum_{j=1}^{k_i}(\log Y_{ij} - \mu_{ij}) \\ \sigma^{-2}\sum_{j=1}^{k_i}\mathbf{x}_{ij}^T(\log Y_{ij} - \mu_{ij}) \\ -k_i\sigma^{-1} + \sigma^{-3}\sum_{j=1}^{k_i}(\log Y_{ij} - \mu_{ij})^2 \end{pmatrix},$$

where $\mu_{ij} = \alpha + \mathbf{x}_{ij}\boldsymbol{\beta}$. $E_g\left\{\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]})\right]\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]})\right]^T\right\}$ can then be estimated for each $i = 1, \ldots, n$ through importance sampling.

3.8 Pooling Methods

Recall that, for this study, we assume that N specimens have been collected, but only n ($< N$) lab tests can be afforded. Perhaps the simplest way to meet this requirement is to randomly select n of the available specimens. While this selection strategy allows for straightforward analysis of this subset, it omits many of the specimens from the study altogether, often resulting in a considerable loss of efficiency. Randomly pooling specimens into equal-sized groups would allow all of the specimens to be included in the analysis, but this method is often accompanied by a similar efficiency loss. An alternative approach to reduce the number of lab tests while maintaining a high level of efficiency incorporates covariate data into the pooling process. So long as the pools are based only on the fully observed covariate values, appropriately defined regression coefficient estimates as well as their estimated variances remain valid as a consequence of viewing pooled outcomes as

partially missing data (Little and Rubin, 2002).

3.8.1 x-homogeneous Pools

As demonstrated in Section 3.5, when pools are formed from subjects with identical covariates (for binary or categorical covariates), valid coefficient estimates can easily be obtained by applying a linear regression to the log-transformed pooled values (Approximate Model). In order to form **x**-homogeneous pools, the number of desired pools (n) must be larger than the number of unique groups with identical covariate values, say G. For this study, homogeneous pools were formed under the following conditions:

1. Each group of unique covariates is required to supply at least one pool.
2. Groups containing only one member contribute that individual specimen for analysis.
3. When possible, pools are formed to have similar sizes.

3.8.2 k-means Clustering

When it is not possible to form **x**-homogeneous pools (e.g. when $n < G$ or at least one covariate is continuous), the k-means clustering algorithm discussed in Chapter 2 can be applied to the design matrix to maintain similar covariate values in each pool. In subsequent simulations, we standardize each variable prior to performing k-means clustering to ensure that each variable contributes similar influence on the resulting clusters, since we consider all predictors to be equally important. This technique will often, but not always, identify pools that are homogeneous with respect to any binary or categorical variables included in the clustering procedure. Alternatively, if we were interested in a particular variable, we could use the weighted k-means technique described in Section 2.3.3 to improve the efficiency of its corresponding coefficient estimate. As illustrated in Figure 2.3, however, this increase in precision can be accompanied by a reduction in precision for the remaining estimates, so careful consideration must be taken before implementing such weighting strategies.

3.9 Simulation Study

For each of the simulation scenarios, 5000 simulations were performed in R. Datasets from the first two simulation studies were simulated to resemble actual motivating data described in Section 3.2, with sample size $N = 672$. Independent predictor variables were generated to mimic age (years), smoking status (yes/no), race (1 = white / 2 = black), and SA status (yes/no), and the outcome variable was generated to resemble the cytokine MCP1 (μg/mL) based on a lognormal regression against those predictors. Age was simulated as a normal random variable with mean 26.6 and standard deviation 6.4, then rounded to the nearest whole number (this permits the formation of **x**-homogeneous pools when average pool size is small). Smoking status, race, and SA status were simulated as Bernoulli random variables with probabilities 0.47, 0.28, and 0.46, respectively. The outcome, MCP1, was generated under a lognormal distribution such that $E[\log(\text{MCP1})|\mathbf{X}] = -2.48 + 0.017(\text{Age}) + 0.007(\text{Smoking Status}) - 0.388(\text{Race}) + 0.132(\text{SA})$ and $Var[\log(\text{MCP1})|\mathbf{X}] = 1.19$.

In the first study, we assess each of the proposed analytical strategies when applied to **x**-homogeneous pools ($n = 336$) mimicking data from the CPP substudy, and in the next study we compare estimate precision from the various pooling strategies applied to the same generated datasets, comparing k-means clustering to random pooling and selection when **x**-homogeneous pools cannot be formed ($n = 112$).

The last two simulation studies were developed to assess performance of the analytical methods in additional scenarios. First, we generate a dataset such that application of all proposed methods (excluding the Naive Model) is feasible and theoretically justified. Specifically, pools were formed **x**-homogeneously on the covariates (to justify analysis under the Approximate Model) with a maximum pool size of 2 (to enable application of the Convolution Method). In the first two simulation studies, the nature of the simulated data precluded formation of pools with both of these characteristics. The final simulation demonstrates a scenario in which the Approximate Model fails and the Convolution Method falters, to caution against analysis via the former when pools are not **x**-homogeneous, and via the latter (even for pools of maximum size 2) when the convolution integral may be poorly behaved.

3.9.1 Comparing Analytical Strategies

The goal of this first simulation study is to assess each of the discussed analytical strategies for **x**-homogeneous pools applied to data resembling the CPP substudy, where an analytical method is deemed appropriate if it provides accurate estimates of the regression coefficients as well as their standard errors. Pools were formed based on the **x**-homogeneous clustering strategy described in Section 3.8.1, where pool sizes ranged from 1 to 6. Analytical strategies under consideration included standard least squares regression on log-transformed pooled outcomes (Naive Model), WLS on the log-transformed pools with inverted pool size as a predictor variable (Approximate Model), and the likelihood-based MCEM strategy under lognormal regression (MCEM Model). We also provide regression results from the full data as well as a random sample of size $n = 336$ for comparison purposes. Since many of the pools in this simulation consisted of more than 2 specimens, direct optimization of the likelihood under the Convolution approach was not viable for this first simulation.

Table 3.2 displays the mean bias and empirical standard deviation (SD) of the regression coefficient estimates. The ratio of mean estimated standard error to empirical standard deviation ($\widehat{\text{SE}}/\text{SD}$) is also provided, where a value of 1 is ideal. 95% confidence interval (CI) coverage is based on the estimated standard errors and a t-reference distribution with $n - 5$ degrees of freedom.

Based on these simulation results, the Naive Model provides biased estimates, which can result in severe CI undercoverage. This characteristic is particularly noticeable for $\hat{\beta}_3$, which has only 81% CI coverage. The remaining methods provide approximately unbiased estimates of the regression coefficients (Mean Bias ≈ 0) as well as their estimated standard errors ($\widehat{\text{SE}}/\text{SD} \approx 1$) and close to 95% CI coverage. Thus, both the Approximate as well as the MCEM Models provide valid results when pools are **x**-homogeneous.

Although the main purpose of this simulation is to test the validity of the proposed analytical methods, it is also worth noting that estimates from these **x**-homogeneous pools analyzed under the Approximate and MCEM Models are noticeably more precise than those from a random sample, and are only slightly less efficient than estimates from the full dataset. The MCEM method appears to provide marginally more precise estimates

Table 3.2: Simulation results comparing various analytical models for lognormal regression on **x**-homogeneous pools. "SD" refers to empirical standard deviation of regression coefficient estimates and "$\hat{\text{SE}}$" is the mean estimated standard error.

Mean Bias (SD) $\hat{\text{SE}}$/SD (95% CI Coverage)					
Method	n	$\beta_1 = 0.017$	$\beta_2 = 0.007$	β_3 = -0.388	$\beta_4 = 0.132$
Full Data	672	0.000 (0.007) 0.99 (94.8)	-0.002 (0.085) 1.00 (95.1)	-0.001 (0.092) 1.02 (95.4)	0.001 (0.085) 1.00 (94.7)
Random Sample	336	0.000 (0.009) 1.00 (95.1)	-0.003 (0.120) 1.00 (94.6)	-0.003 (0.130) 1.03 (95.7)	0.000 (0.118) 1.02 (95.3)
Naive Model	336	0.000 (0.007) 0.89 (92.0)	-0.016 (0.099) 1.00 (94.8)	-0.111 (0.106) 0.98 (80.9)	-0.016 (0.099) 1.01 (94.6)
Approx. Model	336	0.000 (0.007) 0.99 (94.6)	-0.002 (0.092) 0.98 (94.7)	-0.002 (0.105) 1.00 (95.3)	0.001 (0.092) 0.98 (94.3)
MCEM Model	336	0.000 (0.007) 0.98 (94.3)	-0.002 (0.091) 0.99 (94.9)	-0.003 (0.098) 1.01 (95.0)	0.001 (0.090) 1.00 (94.8)

than those under the Approximate Model, most likely due to the fact that this method identifies MLEs, which are well-known to be the most efficient estimates when the assumed underlying distribution is correctly specified, as in this simulation. This slight improvement in efficiency, however, is unlikely to motivate the additional computational time and effort required to implement the MCEM method. Thus, the Approximate Model may be the most desirable analytical method when pools are **x**-homogeneous, due to the simplicity of its application as well as its flexibility in not requiring specific distributional assumptions on the errors in the underlying model (3.1).

3.9.2 Comparing Pooling Strategies

In the next simulation, we compare regression results based on pooling by k-means clustering versus random pooling, when **x**-homogeneous pools cannot be formed due to a small number of pools ($n = 112$). k-means clustering was performed using the *kmeans* function in R, where pool sizes ranged from 1 to 49, with an average size of 6. Each of the pooled strategies was analyzed under the MCEM algorithm, since the heterogeneity of the pools and large

pool sizes precluded defensible application of the Approximate Model and Convolution Method.

Table 3.3: Simulation results for regression analysis on various pooling methods using the MCEM algorithm. "SD" refers to empirical standard deviation of regression coefficient estimates and "$\hat{\text{SE}}$" is the mean estimated standard error.

Mean Bias (SE) $\hat{\text{SE}}$/SD (95% CI Coverage)					
Method	n	$\beta_1 = 0.017$	$\beta_2 = 0.007$	β_3 = -0.388	$\beta_4 = 0.132$
Full Data	672	0.000 (0.007) 0.99 (94.8)	-0.002 (0.085) 1.00 (95.1)	-0.001 (0.092) 1.02 (95.4)	0.001 (0.085) 1.00 (94.7)
Random Sample	112	0.000 (0.017) 0.99 (95.2)	-0.002 (0.210) 1.00 (95.1)	0.002 (0.231) 1.01 (95.6)	0.004 (0.212) 0.99 (95.1)
Random Pools	112	0.000 (0.019) 0.96 (93.8)	-0.009 (0.247) 0.97 (94.6)	-0.018 (0.318) 0.97 (95.4)	-0.001 (0.251) 0.95 (94.2)
k-means Pools	112	0.000 (0.008) 0.96 (93.8)	-0.003 (0.101) 0.97 (94.5)	-0.005 (0.108) 0.99 (95.1)	0.000 (0.099) 0.98 (94.5)

Based on the results in Table 3.3, all pooling and selection strategies provide approximately unbiased estimates with close to nominal 95% CI coverage. While random pooling seems to give slightly biased estimates of β_3, further examination suggests that this apparent bias is a consequence of the estimate exhibiting a slightly left-skewed distribution, likely due to the sample size being too small for asymptotic normality of this MLE to apply. Estimates of β_3 under k-means pooling, on the other hand, do not exhibit this characteristic, suggesting that asymptotic properties may be applicable at a smaller number of pools when k-means clustering is performed. k-means pooling also provides coefficient estimates that are considerably more precise than both random strategies, more than doubling the efficiency for each of the estimates, and losing surprisingly little efficiency relative to the full data analysis even at $1/6^{\text{th}}$ the original sample size. Thus, a considerable amount of information can be retained from the full data when k-means pooling is performed and appropriate analytical techniques are applied, at just a fraction of the total laboratory cost.

3.9.3 Convolution Method

In the previous simulations, the Convolution Method was not applied because pool sizes often exceeded 2, causing the numerical integration required for optimization under this method to become computationally intractable. In this simulation study, datasets and pools are formed specifically to permit application of each of the methods proposed in this chapter. The Naive Model was not included in this analysis as it has been shown, both theoretically and in simulations, to be an invalid estimation procedure. In fact, the Naive Model would only be expected to provide a valid estimation procedure if pools were formed **x**-homogeneously and with equal-sized pools, as the Naive Model would be then equivalent to the Approximate Model. Actual datasets with the possibility of forming such pools, however, are unlikely to be encountered in most real-world research settings.

For this simulation, datasets with a total sample size of $N = 1800$ were generated to contain three Bernoulli-distributed covariates (X_1, X_2, X_3) with probabilities 0.3, 0.5, and 0.8, respectively. The outcome Y was generated as a lognormal random variable such that $E(\log Y|\mathbf{X}) = 0 + 0.5X_1 + 0.3X_2 - 0.1X_3$ and $Var(\log Y|\mathbf{X}) = 0.64$. 1000 **x**-homogeneous pools were then formed so that every pool contained either 1 or 2 specimens. Results for this study can be found in Table 3.4.

These results serve to validate each of the proposed analytical strategies. All methods provide essentially unbiased estimates of both the regression coefficients as well as their standard errors, and exhibit close to 95% confidence interval coverage. Estimates from the MCEM and Convolution Methods are nearly identical, supporting the validity of the MCEM method as an alternative to direct optimization of the observed likelihood (i.e., the Convolution Method) for calculating MLEs. Results from the Approximate Model are also extremely similar to those from both the MCEM and Convolution Method, further endorsing this model as a more accessible estimation procedure when pools are **x**-homogeneous. Similar to the simulation results from Section 3.9.1, the **x**-homogeneous pooling strategy provides more precise estimates (i.e., lower SD) than a random sample of the same size.

Table 3.4: Additional simulation results for regression analysis on the proposed analytical methods, where pools are **x**-homogeneous and pool size does not exceed 2. "SD" refers to empirical standard deviation of regression coefficient estimates and "$\hat{\text{SE}}$" is the mean estimated standard error.

Mean Bias (SE) $\hat{\text{SE}}$/SD (95% CI Coverage)				
Method	n	$\beta_1 = 0.5$	$\beta_2 = 0.3$	β_3 = -0.1
Full Data	1800	0.000 (0.041) 1.01 (95.1)	0.000 (0.038) 0.99 (94.8)	0.000 (0.048) 0.99 (94.6)
Random Sample	1000	0.000 (0.055) 1.01 (95.2)	0.000 (0.051) 1.00 (95.0)	-0.001 (0.064) 0.99 (94.5)
Approximate Model	1000	0.000 (0.043) 1.01 (94.9)	0.000 (0.040) 0.99 (94.6)	0.000 (0.050) 0.99 (94.5)
MCEM Model	1000	0.000 (0.042) 1.01 (95.4)	0.000 (0.039) 0.99 (94.6)	0.000 (0.050) 0.99 (94.7)
Convolution Method	1000	0.000 (0.042) 1.01 (95.4)	0.000 (0.039) 0.99 (94.7)	0.000 (0.050) 0.99 (94.8)

3.9.4 A Cautionary Tale

As evident in previous simulations, the Approximate Model can be a valuable tool for analyzing **x**-homogeneous pools. When pooled covariate values are similar, although not identical, within pools, this model may appear to provide valid estimates of the regression coefficients; care must be taken, however, since fitting the Approximate Model when pools are not entirely homogeneous can result in flawed inference. To illustrate the potential repercussions of applying the Approximate Model to heterogeneous pools, we performed a simulation study in R with 5000 repetitions, for $N = 400$, $X_1 \sim Exp(0.3)$, $X_2 \sim Bernoulli(0.15)$, $X_3 \sim Bernoulli(0.8)$, and $\log(Y) \sim N(\mu, 0.6^2)$, such that $\mu = 3 - 0.5(X_1) + 0.7(X_2) + 0.2(X_3)$. Pools were formed randomly in groups of 2 ($n = 200$), then fit under the Approximate Model, the MCEM algorithm, and the Convolution Method. Note that in this situation, the Approximate Model is equivalent to the Naive Model, since all pools have the same size.

Numerically, the Convolution Method proved rather unreliable in analyzing many of these datasets. Performing numerical integration with the *integrate* function in R suffered

from extremely low rates of convergence. To mitigate this issue, the *quadinf* function from the "pracma" package was applied as an alternate numerical integration procedure. Convergence rates under this revised convolution function were close to 62%.

Table 3.5: Simulation results for 200 randomly formed pools of size 2 fit under the Approximate Model, MCEM, and Convolution Method, where the Approximate Model is expected to perform poorly due to **x**-heterogeneity of pools. "SD" refers to empirical standard deviation of regression coefficient estimates and "$\widehat{\text{SE}}$" is the mean estimated standard error.

Mean Bias (SD) $\widehat{\text{SE}}$/SD (95% CI Coverage)				
Method	n	$\beta_1 = -0.5$	$\beta_2 = 0.7$	$\beta_3 = 0.2$
Full Data	400	0.000 (0.009) 0.99 (95.4)	0.000 (0.085) 0.99 (94.9)	0.001 (0.076) 0.99 (94.7)
Approximate Model	200	0.188 (0.036) 0.58 (0.0)	0.054 (0.201) 0.97 (93.3)	-0.007 (0.174) 1.00 (95.0)
MCEM Model	200	0.000 (0.021) 0.97 (94.7)	0.000 (0.112) 0.98 (94.3)	-0.001 (0.111) 0.99 (95.2)
Convolution Method	200	0.015 (0.040) 0.51 (82.9)	-0.002 (0.114) 0.99 (94.6)	-0.002 (0.113) 1.00 (95.1)

As evident in Table 3.5, the Approximate Model suffers noticeable bias for both β_1 and β_2, accentuated by a 0% CI coverage for β_1. Even when the Convolution Method appeared to be converging properly, simulation results suggest otherwise, as this method fails to produce optimal estimates, particularly with respect to β_1. The MCEM model, on the other hand, produces essentially unbiased estimates of the coefficient parameters and standard errors, with CI coverage close to 95%.

This simulation emphasizes the importance of choosing analytical techniques that are appropriate for the pooling method. Furthermore, even analytical methods that may be theoretically justified, such as the Convolution Method in this example, can still fall victim to sub-optimal convergence rates. More importantly, apparent convergence of the Convolution Method for an individual dataset, particularly when additional effort is required in order to achieve that convergence, does not necessarily imply true convergence. The MCEM method, on the other hand, while requiring more computational effort, can provide valid estimates where these other analytical procedures fail.

3.10 Data Analysis

Analysis of the CPP substudy example was conducted on data from 672 participants who provided complete information on MCP1 as well as each of the 4 covariates. The single MCP1 measurement that fell below the detection limit was assigned a value of 0.00001, and race values were restricted to include values 1 (white) and 2 (black), as only a small number (40 observations) were of other races. In addition, specimens from 508 of the participants had been combined to form 254 actual pools, each containing 2 specimens (Whitcomb et al., 2012). MCP1 values were measured again on these pools. Thus, we have access to MCP1 measurements from the complete dataset (672 lab assays), as well as from a dataset of 254 pools and 164 individual specimens (418 lab assays).

For this analysis, we first perform linear regression on a log-transformation of MCP1 on the 672 individual measurements. These results serve as the "gold standard" for subsequent analyses, since they represent the maximum information available from the dataset. Next, we analyze the set of 418 observed pooled measurements and individual specimens. As an additional comparison, we then perform regression on the same set of 418 pools and individuals, but this time, we use artificial pooling to determine the expected value of each pool, calculated as the arithmetic mean of the measurements from individual specimens. These observed and expected pooled datasets are analyzed under both the MCEM algorithm and the Convolution Method.

For our final analysis, $\mathbf{x}$-homogeneous pools are created artificially, in order to illustrate the results that would have been available had the entire set of observed covariate information been used to inform the pooling process. At the desired sample size ($n = 418$) pool sizes ranged from 1 to 5 in order to maintain homogeneity among the pooled covariates. The Convolution approach was not available due to these larger pool sizes. Instead, we analyze these $\mathbf{x}$-homogeneous pools using the MCEM algorithm and the Approximate Model.

Results of these data analyses are provided in Table 3.6. Estimates from the MCEM and Convolution approaches are almost identical, validating the performance of these methods as appropriate algorithms for estimating MLEs. The observed and expected pools provide similar conclusions with respect to estimated standard errors and significance levels, al-

Table 3.6: Results from regression analyses on the individual and pooled dataset. A "*" indicates predictors that were found to be significantly associated with log(MCP1) at the 0.05 level. "Full Data" was analyzed via linear regression on log(MCP1), "Observed Pools" and "Expected Pools" under MCEM and Convolution Method, and "Homogeneous Pools" under both MCEM and Approximate Model.

Estimate (SE)					
Data	Age (years)	Smoking Status (yes/no)	Race (black/white)	SA Status (yes/no)	$\hat{\sigma}$
Full Data	0.017 (0.007)*	0.007 (0.085)	-0.388 (0.095)*	0.132 (0.086)	1.09
Observed Pools					
MCEM	0.020 (0.009)*	0.083 (0.127)	-0.159 (0.146)	0.135 (0.104)	1.23
Convolution	0.020 (0.010)*	0.083 (0.127)	-0.159 (0.148)	0.136 (0.104)	1.23
Expected Pools					
MCEM	0.021 (0.009)*	0.045 (0.125)	-0.214 (0.146)	0.122 (0.101)	1.21
Convolution	0.021 (0.009)*	0.045 (0.124)	-0.214 (0.146)	0.122 (0.101)	1.21
Homogeneous Pools					
MCEM	0.017 (0.007)*	0.026 (0.092)	-0.306 (0.102)*	0.143 (0.093)	1.12
Approx.	0.016 (0.007)*	0.022 (0.092)	-0.308 (0.103)*	0.141 (0.092)	1.17

though the actual estimates tend to vary. This discrepancy is likely due to measurement error or pooling error, a topic explored in depth by Schisterman et al. (2010) when measuring an exposure of interest on pooled samples. While detailed evaluation of these potential sources of error is beyond the scope of the current study, this issue highlights the importance of addressing the potential effects of additional error components when analyzing biomarkers.

For $\mathbf{x}$-homogeneous pools, the MCEM algorithm and the Approximate Model provide almost identical results, emphasizing the advantages of the more accessible Approximate Model. Estimates from these artificially-formed pools are generally most similar to those from the full data analysis, and all are more precise than estimates obtained from the actual pools, which were formed homogeneously only with respect to SA status (Whitcomb et al., 2012). In addition, these $\mathbf{x}$-homogeneous pools concur with the full data results that race is significantly associated with levels of MCP1, a relationship that is lost when pools are randomly formed with respect to this covariate. As demonstrated by this data analysis and corroborated by the simulation studies, utilizing the entire covariate information to create

pools can preserve associations present in the full dataset, maintain a high level of efficiency, and even simplify the analytical process when pools are **x**-homogeneous.

3.11 Discussion

Our goal for this chapter was to develop methods for analyzing pooled, right-skewed data, specifically when a log-transformation is needed on the individual-level outcome and budgetary constraints limit the number of assays that can be performed. When covariate data is available prior to any physical pooling, this information can be utilized to form pools that will produce precise regression estimates, often losing a minimal amount of information at a fraction of the original sample size. When possible, forming **x**-homogeneous pools will not only tend to maximize efficiency for a particular sample size, but can also allow exploitation of a Taylor series approximation, so that a suitably specified linear regression model applied to the log of the observed, pooled values will yield appropriate and precise estimates.

If it is not possible to form **x**-homogeneous pools, k-means clustering can provide an efficient pooling strategy with respect to estimation of regression coefficients, but subsequent pools will likely require application of an MCEM algorithm, since at least one pool will almost certainly contain more than two specimens. If specimens have already been combined into non-homogeneous pools, as in the motivating dataset from the CPP study, MLEs of regression estimates can be calculated via an MCEM algorithm, or through a Convolution Method if pool size does not exceed 2. These estimation procedures, however, require additional distributional assumptions on the outcome and can prove computationally demanding. Assessment of the validity of these assumptions is complicated by the fact that many of the available measurements are pooled. While it is possible to use only the unpooled data for common regression diagnostics, much of the original information may be lost. In the next chapter, we explore the consequences of distributional misspecification on the analytical methods presented here.

Chapter 4

Comparing Parametric and Semi-Parametric Models for a Skewed, Pooled Outcome

4.1 Introduction

When faced with the task of performing regression on a positive, right-skewed continuous outcome, a common approach is to assume a lognormal distribution, so that applying a log transformation to the outcome will simplify to a linear regression model. In Chapter 3, this approach was extended to accommodate pooled data from a lognormal distribution, and appropriate estimation procedures were developed and tested. These strategies, however, may not be ideal for all right-skewed distributions.

Another model that may prove useful for analyzing such data is the gamma distribution. Similar to the lognormal, the gamma distribution is also appropriate for modeling positive and continuous right-skewed outcomes, and is likewise often used in conjunction with a log link in generalized linear regression models. In fact, the lognormal and gamma regression models are often interchangeable (Firth, 1988). The gamma distribution can be particularly beneficial for modeling data on pooled specimens, since, if the individual-level measurements follow a gamma distribution, the pooled measurements (assumed to be the

mean of individual specimens) also follow a gamma distribution when pools are formed from individuals with identical covariate values. Recent epidemiological studies involving pooled specimens have successfully utilized this summation property in special cases of gamma regression, e.g., two-group comparisons of mean biomarker levels (Whitcomb et al., 2012; Perkins et al., 2011).

When pools are not homogeneous with respect to covariates, however, the summation property does not apply under the usual parameterization of the gamma regression model. When at most two covariates groups are represented in a pool, the observed, pooled, likelihood can be maximized where the density of each pool is characterized by a convolution integral (Perkins et al., 2011). For pools containing specimens with more than two unique covariate values, however, numerically evaluating the likelihood quickly becomes computationally intractable, due to high-dimensional integrals in the pooled density.

We propose several methods for dealing with this issue, which we then assess in simulation studies. The first applies an alternate parameterization for gamma regression, which can take full advantage of the gamma summation property for all types of pooled data. The second method calculates MLEs based on the standard gamma regression parameterization, applying a specific version of the Monte Carlo Expectation Maximization (MCEM) method described in Section 3.7. The third approach applies a strategy based on quasi-likelihood methods, where only the mean and variance of the pooled measurements are specified, as opposed to the entire distribution. This semi-parametric model permits straightforward analysis and provides a more flexible framework for modeling skewed data, as the full specification of the outcome distribution is not required. As a consequence of these weaker assumptions, however, estimate precision can deteriorate when compared with correctly specified fully parametric models.

Akaike's Information Criterion (AIC) provides a convenient and useful guide for helping select the best parametric model in order to help ensure validity and optimize estimate precision, when full specification of the outcome distribution is deemed appropriate. Previous studies have demonstrated the effectiveness of AIC in differentiating between the lognormal and gamma models (Burnham and Anderson, 2002; Dick, 2004). For standard likelihood-based regression methods, calculation of AIC is straightforward. The proposed

MCEM algorithms for pooled outcome data, however, do not directly provide an observed data likelihood. Instead, we propose and demonstrate the approximation of AIC using additional Monte Carlo methods.

In the next section, we introduce each of the parametric models to be considered, for individual-level as well as pooled data. We then describe the semi-parametric models that require specification of only the first two conditional moments of the outcome, and determine appropriate application of these models specific to pool type. Next, we perform simulation studies under each pooling strategy to illustrate the potential consequences of model misspecification. Finally, we apply these methods to regression performed on the substudy of data collected from the CPP.

4.2 Parametric Regression Models for Skewed Outcomes

Suppose we have N subjects separated into n groups (later to be defined as pools), where group i contains k_i subjects, so that $N = \sum_{i=1}^{n} k_i$. Let the 'ij' subscript denote the j^{th} subject in the i^{th} group. In this section we present three potentially appealing parametric regression models for a positive, right-skewed outcome.

4.2.1 Lognormal Model

As mentioned in Chapter 3, positive, right-skewed data are often assumed to be lognormally distributed, in order to take advantage of the convenient properties of the normal distribution applied to a log-transformation on the outcome. While the details of this model can be found in Chapter 3, we reiterate here for convenience:

$$\log(Y_{ij}) = \alpha + \mathbf{x}_{ij}\boldsymbol{\beta} + \epsilon_{ij}, \quad j = 1, \ldots, k_i, \ i = 1, \ldots, n$$

where α is the intercept and $\boldsymbol{\beta}$ the $P \times 1$ vector of regression coefficients. Y_{ij}, $\mathbf{x}_{ij}$, and ϵ_{ij} represent the outcome, covariate vector, and error for the j^{th} subject in the i^{th} group, respectively, and we assume independent errors with $E(\epsilon_{ij}) = 0$ and $Var(\epsilon_{ij}) = \sigma^2$. Recall that, while no further distributional assumptions are required in order to calculate least

squares regression coefficients and standard errors, the additional assumption of normality on the ϵ_{ij}'s is necessary to apply exact 95% confidence intervals based on the t-distribution, as well as to compare model fit using likelihood-based model selection criteria such as AIC.

As discussed in Section 3.5, a semi-parametric weighted least squares estimation approach can provide appropriate estimates when pools are formed with identical covariate values. We will discuss this model more in Section 4.3.1. For now, we will revisit the likelihood-based MCEM method for calculating MLEs under **x**-heterogeneously pooled data.

As detailed in Section 3.7, the MCEM algorithm calculates MLEs by maximizing the expectation of the complete log-likelihood, given the observed data. Let the complete likelihood be denoted by $L_c(\boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{k_i} f(Y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\theta})$, where f is the density of the individual (unpooled) outcomes. At each iteration of the EM algorithm, parameter estimates are updated by maximizing $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = E[\log L_c(\boldsymbol{\theta})|\mathbf{Y}^p, \mathbf{X}, \boldsymbol{\theta}^{(t)}]$ with respect to $\boldsymbol{\theta}$, where $\mathbf{Y}^p$ represents the observed vector of pooled outcome measurements. We apply a Monte Carlo approximation using importance sampling to estimate Q since a closed form is difficult to achieve when a lognormal assumption is applied to heterogeneous pools.

Let $h(\mathbf{Y}_{i[\text{-}1]})$ represent any of the functions of the missing data contained in Q. Then, under importance sampling, $E_g[h(\mathbf{Y}_{i[\text{-}1]})]$ can be approximated by

$$E_g[h(\mathbf{Y}_{i[\text{-}1]})] \approx \frac{\sum_{m=1}^{M} h(\mathbf{Y}_{i[\text{-}1],m}) w(\mathbf{Y}_{i[\text{-}1],m})}{\sum_{m=1}^{M} w(\mathbf{Y}_{i[\text{-}1],m})},$$

where each $Y_{ij,m}$ is generated under g^*, the alternate distribution proposed in Section 3.7.2 that facilities generation of samples conforming to the linear inequality constraint. The expression for $w(\mathbf{Y}_{i[\text{-}1]}) = g(\mathbf{Y}_{i[\text{-}1]}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})/g^*(\mathbf{Y}_{i[\text{-}1]}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})$ depends on the assumed distribution of the individual-level data. When the data are assumed lognormal, these weights are:

$$w(\mathbf{Y}_{i[\text{-}1],m}) = \exp\left\{ -\frac{1}{2\sigma^2 k_i} \left[\sum_{j=1}^{k_i} (\log y_{ij,m} - \mu_{ij}) \right]^2 \right\}$$

where $y_{i1,m} = k_i Y_i^p - \sum_{j=2}^{k_i} y_{ij,m}$ and $\mu_{ij} = \alpha + \mathbf{x}_{ij}\boldsymbol{\beta}$ (derivation in Section 3.7.5). To calculate standard errors when using the MCEM algorithm, we apply similar MC techniques to those based on Louis's method (Louis, 1982), where the information matrix can be written

as:

$$\frac{d^2Q}{d\boldsymbol{\theta}^2} - \left(\frac{dQ}{d\boldsymbol{\theta}}\right)\left(\frac{dQ}{d\boldsymbol{\theta}}\right)^T + \sum_{i=1}^{n} E_g\left\{\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]})\right]\left[\frac{d}{d\boldsymbol{\theta}}\log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]})\right]^T\right\}. \quad (4.1)$$

In Section 3.7.5, we provide an explicit expression for the gradient of the complete log-likelihood, which can be estimated using additional Monte Carlo methods based on importance sampling. As noted in Section 3.7.4, numerical approximation functions in existing software packages (e.g. *hessian* and *grad* functions from "numDeriv" package in R) can be used to calculate the Hessian and gradient of the conditional expectation (Q), denoted $\frac{d^2Q}{d\boldsymbol{\theta}^2}$ and $\frac{dQ}{d\boldsymbol{\theta}}$, respectively, in (4.1).

4.2.2 Gamma[1] Model

As mentioned previously, the gamma distribution provides another popular model for a right-skewed outcome. Let $Y_{ij} \sim Gamma(a_{ij}, b_{ij})$, where a_{ij} and b_{ij} are the shape and scale parameters, respectively, and let $f(Y_{ij})$ denote the gamma density for the observation from the j^{th} subject in the i^{th} group, such that:

$$f(Y_{ij}) = \frac{1}{\Gamma(a_{ij})b_{ij}^{a_{ij}}} e^{-Y_{ij}/b_{ij}} Y_{ij}^{a_{ij}-1}, \quad a_{ij}, b_{ij} > 0,$$

where $\mu_{ij} = E(Y_{ij}) = a_{ij}b_{ij}$ and $Var(Y_{ij}) = a_{ij}b_{ij}^2$. The log link is most commonly used in conjunction with gamma regression, giving the model:

$$\log(\mu_{ij}) = \eta_{ij} = \alpha + \mathbf{x}_{ij}\boldsymbol{\beta}$$

where η_{ij} represents the linear predictor component from generalized linear model theory. In most standard GLM procedures (e.g., PROC GENMOD in SAS and *glm* in R), the default parameterization is to assume a constant shape parameter (a) and allow the scale parameter (b_{ij}) to model the expectation, so that $b_{ij} = a^{-1}\mu_{ij} = a^{-1}\exp(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})$. We will refer to this parameterization as the gamma[1] model. This parameterization maintains a constant coefficient of variation (CV), where $CV = \sqrt{Var(y)}/E(Y)$, a property shared by the lognormal distribution.

The gamma distribution enjoys a convenient summation property that renders it a natural analytical tool for pooled data (Whitcomb et al., 2012; Perkins et al., 2011). Namely, if the Y_{ij}'s are independent and share a common within-pool scale parameter (i.e., $Y_{ij} \sim Gamma(a_{ij}, b_i)$ for $j = 1, \ldots, k_i$), then Y_i^p also follows a gamma distribution, such that $Y_i^p \sim Gamma(\sum_{j=1}^{k_i} a_{ij}, b_i k_i^{-1})$. To see this, consider the moment-generating function (mgf) for Y_{ij}:

$$m_{Y_{ij}}(t) = E\left(e^{tY_{ij}}\right) = [1 - (b_i t)]^{-a_{ij}}.$$

Then the mgf for $Y_i^p = k_i^{-1} \sum_{j=1}^{k_i} Y_{ij}$ is:

$$\begin{aligned} m_{Y_i^p}(t) &= E\left(e^{tY_i}\right) = E\left(e^{tk_i^{-1}\sum_{j=1}^{k_i} Y_{ij}}\right) = \prod_{j=1}^{k_i} E\left[e^{(t/k_i)Y_{ij}}\right] \\ &= \prod_{j=1}^{k_i} [1 - b_i(t/k_i)]^{-a_{ij}} = [1 - (b_i/k_i)t]^{-\sum_{j=1}^{k_i} a_{ij}}, \end{aligned}$$

which is just the mgf for a gamma random variable with shape $\sum_{j=1}^{k_i} a_{ij}$ and scale b_i/k_i.

Although the gamma[1] model permits the scale parameter to vary across all specimens, an $\mathbf{x}$-homogeneous pooling strategy maintains a constant within-pool scale since all $\mathbf{x}_{ij} = \mathbf{x}_i$ and thus $b_{ij} = a^{-1}\exp(\alpha + \mathbf{x}_i\boldsymbol{\beta})$ for all $j = 1, \ldots, k_i$. So even though the scale parameter varies across pools, the constant scale within each pool permits application of the gamma summation property, so that $Y_i^p \sim Gamma(k_i a, b_i k_i^{-1})$. This characteristic results in the following mean model based on pooled specimens:

$$\begin{aligned} \log(\mu_i) &= \log[E(Y_i^p)] \\ &= \log\left[E\left(\frac{1}{k_i}\sum_{j=1}^{k_i} Y_{ij}\right)\right] \\ &= \log\left[\left(\frac{1}{k_i}\sum_{j=1}^{k_i} \exp(\alpha + \mathbf{x}_i\boldsymbol{\beta})\right)\right] \\ &= \alpha + \mathbf{x}_i\boldsymbol{\beta}, \end{aligned}$$

Furthermore, since $Var(Y_i^p) = k_i^{-1}(\mu_i^2/a)$, a weighted regression with weights $\{k_i : i = 1, \ldots, n\}$ can easily be applied using standard glm software. Alternatively, the observed data

likelihood can be specified and maximized directly using optimization routines available in standard statistical software.

A big advantage of fitting the gamma[1] model to **x**-homogeneous pools is that it produces identical coefficient estimates as under the full data. Suppose that all pools are formed homogeneously with respect to covariate values such that $Y_{ij} \sim Gamma(a, b_i)$ and $b_i = a^{-1}\mu_i = a^{-1}\exp(\alpha + \mathbf{x}_i\boldsymbol{\beta})$ for all $j = 1, \ldots, k_i$. Then the MLEs for $\boldsymbol{\beta}^* = (\alpha, \boldsymbol{\beta})'$ under the full data maximize the unpooled log-likelihood:

$$\begin{aligned} l_f(\boldsymbol{\beta}^*) &= \sum_{i=1}^{n}\sum_{j=1}^{k_i}\left[-\log\Gamma(a) - a\log(b_i) - Y_{ij}/b_i + (a-1)\log Y_{ij}\right] \\ &= -a\sum_{i=1}^{n}\sum_{j=1}^{k_i}\left[\alpha + \mathbf{x}_i\boldsymbol{\beta} + Y_{ij}e^{-(\alpha+\mathbf{x}_i\boldsymbol{\beta})}\right] \\ &= -a\sum_{i=1}^{n}\left[k_i(\alpha + \mathbf{x}_i\boldsymbol{\beta}) + e^{-(\alpha+\mathbf{x}_i\boldsymbol{\beta})}k_iY_i^p\right] \end{aligned}$$

where any expressions not containing $\boldsymbol{\beta}^*$ were removed since they have no impact on the estimation procedure for this parameter. For **x**-homogeneous pools, $Y_i^p \sim Gamma(k_ia, b_ik_i^{-1})$ for each pool, so that the MLEs for $\boldsymbol{\beta}^*$ based on the observed, pooled data maximize:

$$\begin{aligned} l_p(\boldsymbol{\beta}^*) &= \sum_{i=1}^{n}\left[-\log\Gamma(k_ia) - k_ia\log(b_i/k_i) - k_iY_i^p/b_i + (k_ia-1)\log Y_i^p\right] \\ &= -a\sum_{i=1}^{n}\left[k_i(\alpha + \mathbf{x}_i\boldsymbol{\beta}) + e^{-(\alpha+\mathbf{x}_i\boldsymbol{\beta})}k_iY_i^p\right] \end{aligned}$$

where, again, we have removed any expression not containing $\boldsymbol{\beta}^*$. Thus, since the log-likelihoods are identical, the MLEs for $\boldsymbol{\beta}^*$ calculated from both the full and pooled log-likelihood will also be identical. While it may be tempting to pool all specimens with identical covariate values due to this preservation of precision, doing so will almost certainly result in poor standard error estimates (Schisterman et al., 2010). Thus, it is recommended to form enough pools so that resulting inference based on both estimates and standard errors can be trusted.

When pools are **x**-heterogeneous, the summation property of the gamma distribution no longer applies under the gamma[1] model. Instead, we can use MCEM methods (or

Convolution when pool size ≤ 2) similar to those applied to the lognormal model under heterogeneous pools. The main difference is the calculation of the importance weights. When the gamma[1] model is assumed, g^* (referring back to Section 3.7.2) is calculated as:

$$\begin{aligned}
g^* &= \int_0^\infty k_i Y_i^p s^{k_i-1} \prod_{j=1}^{k_i} f(s y_{ij,m} | Y_i^p, \mathbf{x}_{ij}, \boldsymbol{\theta}^{(t)}) ds, \\
&= k_i Y_i^p \int_0^\infty s^{k_i-1} \prod_{j=1}^{k_i} \left\{ \Gamma(a)^{-1} \left(\frac{\mu_{ij}}{a}\right)^{-a} (s y_{ij,m})^{a-1} \exp\left[-\frac{a(s y_{ij,m})}{\mu_{ij}}\right] \right\} ds \\
&= k_i Y_i^p \exp\left(\sum_{j=1}^{k_i} \frac{a y_{ij,m}}{\mu_{ij}}\right) \prod_{j=1}^{k_i} f(y_{ij,m}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \int_0^\infty s^{a k_i - 1} \exp\left(-s \sum_{j=1}^{k_i} \frac{a y_{ij,m}}{\mu_{ij}}\right) ds \\
&= k_i Y_i^p \exp\left(\sum_{j=1}^{k_i} \frac{a y_{ij,m}}{\mu_{ij}}\right) \prod_{j=1}^{k_i} f(y_{ij,m}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \left[\Gamma(a k_i) \left(\sum_{j=1}^{k_i} \frac{a y_{ij,m}}{\mu ij}\right)^{-a k_i}\right]
\end{aligned}$$

where $y_{i1,m} = k_i Y_i^p - \sum_{j=2}^{k_i} y_{ij,m}$, $\mu_{ij} = \exp(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})$, and f represents the density under a gamma[1] distribution. The weights are then:

$$\begin{aligned}
w(\mathbf{Y}_{i[\text{-}1],m}) &= \frac{g(\mathbf{Y}_{i[\text{-}1],m}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})}{g^*(\mathbf{Y}_{i[\text{-}1],m}|Y_i^p, \mathbf{X}, \boldsymbol{\theta}^{(t)})} \\
&= c(Y_i^p) \left(\sum_{j=1}^{k_i} \frac{a y_{ij,m}}{\mu_{ij}}\right)^{a k_i} \exp\left(-\sum_{j=1}^{k_i} \frac{a y_{ij,m}}{\mu_{ij}}\right)
\end{aligned}$$

To calculate the appropriate standard errors from (4.1), the gradient of the gamma[1] model must be derived. To simplify notation, let $\boldsymbol{\beta}^* = (\alpha, \boldsymbol{\beta})$ and let $\mathbf{x}_{ij}^* = (1 \;\; x_{ij1} \ldots x_{ijp})$. Then, under a gamma[1] model, the gradient for pool i can be written as:

$$\begin{aligned}
\frac{d}{d\boldsymbol{\theta}} \log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]}) &= \frac{d}{d\boldsymbol{\theta}} \sum_{j=1}^{k_i} \left[-\log\Gamma(a) - a\log(\mu_{ij}/a) - a Y_{ij}/\mu_{ij} + (a-1)\log(Y_{ij})\right] \\
&= \begin{pmatrix} a \sum_{j=1}^{k_i} \mathbf{x}_{ij}^{*\,T} (Y_{ij}/\mu_{ij} - 1) \\ k_i[1 + \log a - \psi(a)] + \sum_{j=1}^{k_i} \left[\log(Y_{ij}/\mu_{ij}) - Y_{ij}/\mu_{ij}\right] \end{pmatrix},
\end{aligned}$$

where $\boldsymbol{\theta} = (\boldsymbol{\beta}^*, a)$, $\mu_{ij} = \exp(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})$ and $\psi(t) = \frac{d}{dt}\log\Gamma(t)$ is the digamma function. $E_g\left\{\left[\frac{d}{d\boldsymbol{\theta}} \log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]})\right]\left[\frac{d}{d\boldsymbol{\theta}} \log f_c(Y_i^p, \mathbf{Y}_{i[\text{-}1]})\right]^T\right\}$ can then be estimated for each pool using

importance sampling techniques.

4.2.3 Gamma^2 Model

In addition to the gamma^1 model, we consider an alternate parameterization of the gamma distribution, which we refer to as the gamma^2 model. This parameterization assumes a constant scale parameter (b) and models the mean as a function of the shape parameter (a_{ij}). Here, we set $a_{ij} = b^{-1}\mu_{ij} = b^{-1}\exp(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})$, and the relationship between the expectation and variance is now linear, such that $Var(Y_{ij}) = b\mu_{ij}$. This model can prove particularly appealing when data consists of pooled outcome measurements, as the constant scale parameter allows exploitation of the gamma sum property regardless of the homogeneity of pools. Calculating MLEs under this parameterization can be achieved by numerically optimizing a user-defined log-likelihood function in packages such as PROC NLMIXED in SAS or *optim* in R. Sample code is provided in Appendix A.2.

For individual level data, the log-likelihood is maximized with respect to $(\boldsymbol{\beta}^*, b) = (\alpha, \boldsymbol{\beta}, b)$, where

$$\begin{aligned}\log L(\boldsymbol{\beta}^*, b) &= \sum_{i=1}^{n}\sum_{j=1}^{k_i}\log f(Y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\beta}^*, b) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{k_i}[-\log\Gamma(a_{ij}) - a_{ij}\log b - Y_{ij}/b + (a_{ij}-1)\log Y_{ij}], \qquad (4.2)\end{aligned}$$

where $a_{ij} = b^{-1}\exp(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})$.

For pooled data, since the gamma^2 model assumes a constant scale parameter, the sum property applies to pooled outcomes regardless of pooling strategy, so that the pooled values maintain a gamma distribution, such that $Y_i^p \sim Gamma(\sum_{j=1}^{k_i} a_{ij}, bk_i^{-1})$. Maximizing the log-likelihood for pooled data is straightforward, requiring only a slight variation from (4.2), where we now maximize:

$$\begin{aligned}\log L(\boldsymbol{\beta}^*, b) &= \sum_{i=1}^{n}\log f(Y_i^p|\mathbf{X}, \boldsymbol{\beta}^*, b) \\ &= \sum_{i=1}^{n}[-\log\Gamma(a_i) - a_i\log(bk_i^{-1}) - k_iY_i^p/b + (a_i-1)\log Y_i^p)] \qquad (4.3)\end{aligned}$$

where $a_i = \sum_{j=1}^{k_i} a_{ij} = b^{-1} \sum_{j=1}^{k_i} \exp(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})$. Under $\mathbf{x}$-homogeneous pools, a_i reduces to $a_i = k_i b^{-1} \exp(\alpha + \mathbf{x}_i\boldsymbol{\beta})$. Again, numerical optimization methods can be used to maximize (4.3) with respect to b and $\boldsymbol{\beta}^*$.

While each of the aforementioned models may be well suited to regression analysis on a positive, right-skewed outcome, they are unlikely to fit equally well to a particular set of data. A constant mean-variance relationship, for instance, should be best fit by the gamma^2 model, whereas an outcome exhibiting a constant CV would be better modeled by the lognormal or gamma^1 models. Figure 4.1 illustrates how individual-level data generated under each of these models may differ. Each of the models are generated with linear predictor $\eta_i = -1 + x_i$, where $X \sim Bernoulli(0.5)$ and $i = 1, \ldots, 10000$. The first row shows histograms of data generated under a lognormal distribution, such that $\log(Y_i) \sim N(\eta_i, 0.5^2)$. The second row is from data generated under a gamma^1 model, with $Y_i \sim Gamma(a = 2, b_i = e^{\eta_i}/2)$, and the final row has $Y_i \sim Gamma(a_i = 2e^{\eta_i}, b = 1/2)$. The right column provides the histograms of the log of the outcome, separated by the levels of x. As expected, a log transformation on lognormal data provides approximately normal distributions, which readily invites a least squares fit. The log of the gamma^1 outcome, while slightly left-skewed, suggests that a log-transformation may fit this data well. Those from gamma^2, however, are highly left-skewed, indicating that applying a log-transformation to data generated under this distribution may overcorrect the original skewness. In such cases, a model based directly on the gamma^2 distribution may be more appropriate.

4.3 Semi-parametric Regression Models for Skewed Data

While the parametric models described in Section 4.2 will provide the most precise coefficient estimates when correctly specified, semi-parametric models may be preferred when full distributional specification on the outcome could be unreliable. As we will demonstrate in the following sections, these semi-parametric models can greatly simplify analytical procedures, particularly for pooled outcome measurements.

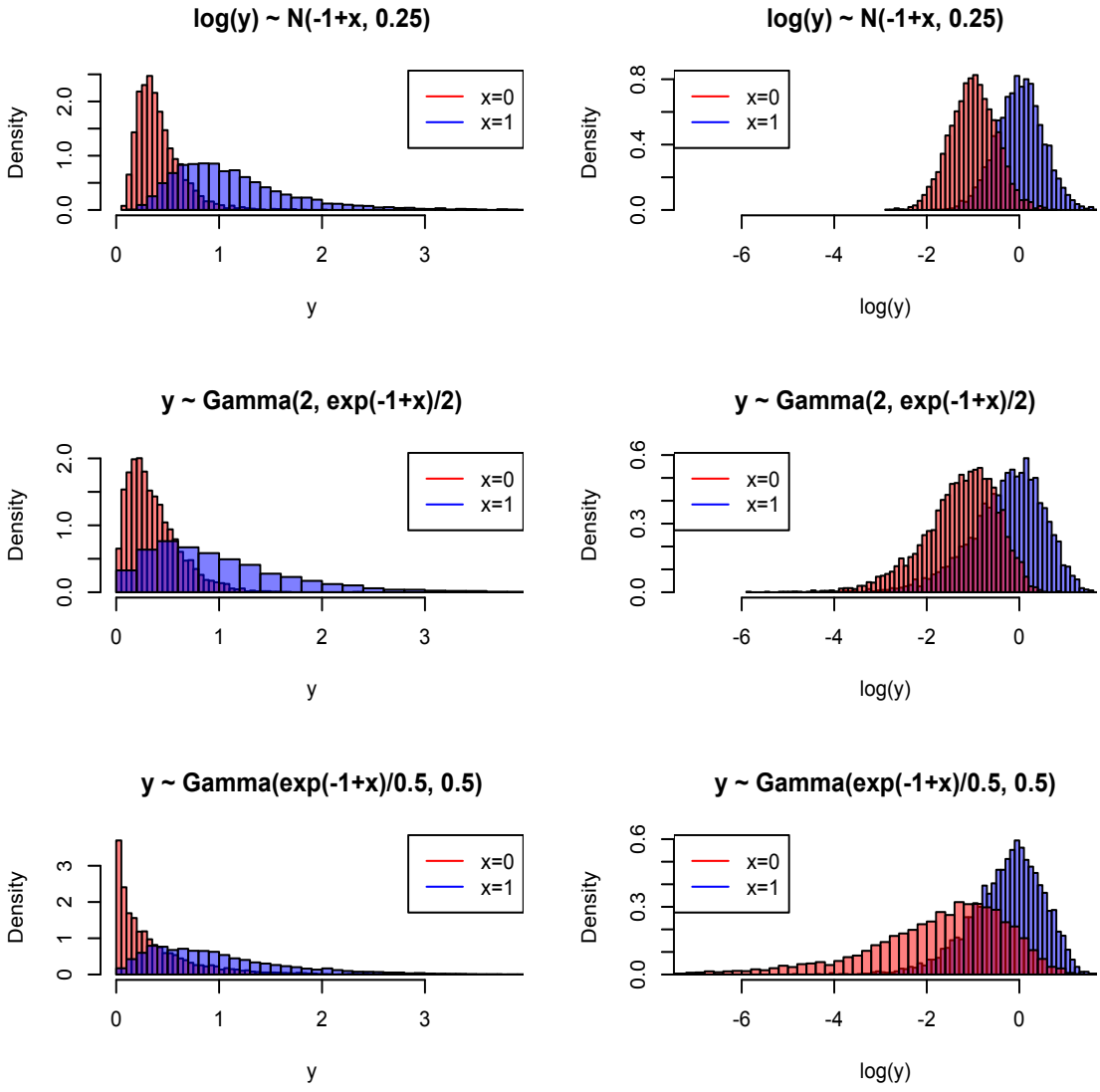

Figure 4.1: Histograms of data generated under lognormal, gamma[1], and gamma[2] distributions.

4.3.1 Approximate Model Revisited

As discussed in Section 3.5, when pools are formed with identical covariate values, a weighted least squares estimation applied to the Approximate Model can provide approximately unbiased estimates of $\boldsymbol{\beta}$ as well as standard errors:

$$\textbf{Approximate Model:} \qquad \log(Y_i^p) \approx \alpha + \gamma k_i^{-1} + \mathbf{x}_i\boldsymbol{\beta} + \delta_i,$$

where δ_i is the error term, k_i is the pool size for pool i, and $Y_i^p = \frac{1}{k_i}\sum_{j=1}^{k_i} Y_{ij}$ denotes the measured value for the i^{th} pool, assumed to be the arithmetic mean response for the individuals comprising that pool. This approximation is based on a Taylor Series expansion, which reduces the effect of pool size on the expectation of $\log(Y_i^p)$ by incorporating it as a predictor variable. The validity of the Approximate Model follows from the assumptions $E(\log Y_{ij}) = \alpha + \mathbf{x}_i\boldsymbol{\beta}$ and $Var(\log Y_{ij}) = \sigma^2$ for all $j = 1, \ldots, k_i$ and $i = 1 \ldots n$. While this model does not require an explicit distributional specification on the outcome, we will demonstrate that this model provides appropriate estimates when data is generated from either a lognormal or gamma[1] model.

In Section 3.5, we showed that for **x**-homogeneous pools, the Taylor Series approximation gives

$$E(\log Y_i^p) \approx \log\left[E(Y_{i1})\right] - \frac{Var(Y_{i1})}{2k_i E(Y_{i1})^2}$$

and

$$Var(\log Y_i^p) \approx \frac{Var(Y_{i1})}{k_i E(Y_{i1})^2}$$

since the homogeneity of pools results in equality of the first two moments for each member of that pool (e.g., $E(Y_{i1}) = E(Y_{i2}) = \cdots = E(Y_{ik_i})$). When the individual specimens are lognormally distributed, $E(Y_{i1}) = e^{(\alpha + \mathbf{x}_i\boldsymbol{\beta} + \sigma^2/2)}$ and $Var(Y_{i1}) = E(Y_{i1})^2(e^{\sigma^2} - 1)$ for all $j = 1, \ldots, k_i$. It follows that:

$$E(\log Y_i^p) \approx (\alpha + \sigma^2/2) - (2k_i)^{-1}(e^{\sigma^2} - 1) + \mathbf{x}_i\boldsymbol{\beta}$$

and

$$Var(\log Y_i^p) \approx k_i^{-1}(e^{\sigma^2} - 1)$$

which is consistent with the requirements for the Approximate Model, such that the original $\boldsymbol{\beta}$ vector is preserved once the inverted pool sizes are incorporated as a covariate, and the variance for each pool is a function of that pool's size.

Now suppose that the Y_{ij}'s are instead generated from a gamma[1] model. Then $E(Y_{i1}) = e^{(\alpha+\mathbf{x}_i\boldsymbol{\beta})}$ and $Var(Y_{i1}) = a^{-1}E(Y_{i1})^2$. Thus,

$$E(\log Y_i^p) \approx \alpha - (2ak_i)^{-1} + \mathbf{x}_i\boldsymbol{\beta}$$

and

$$Var(\log Y_i^p) \approx (ak_i)^{-1},$$

once again preserving the $\boldsymbol{\beta}$ vector by including the k_i^{-1}'s in the mean model and motivating a weighted least squares, as the variance is a function of pool size.

On the other hand, this model is not appropriate when data are generated under the gamma[2] model. Under this alternate parameterizion of gamma regression, $E(Y_{i1}) = e^{(\alpha+\mathbf{x}_i\boldsymbol{\beta})}$ and $Var(Y_{i1}) = bE(Y_{i1})$, resulting in the following approximate mean for the pooled outcomes:

$$E(\log Y_i^p) \approx \alpha + \mathbf{x}_i\boldsymbol{\beta} - \frac{b}{2k_i}e^{-(\alpha+\mathbf{x}_i\boldsymbol{\beta})}$$

Clearly, a least squares approach will not provide appropriate estimates of the regression coefficients under the gamma[2] model, since the approximate expectation of the log-transformed pooled values is a nonlinear function of $\boldsymbol{\beta}$. Thus, we see that although the Approximate Model may not be universally applicable, it does provide a more flexible and computationally accessible alternative to fully parametric models such as the MCEM algorithm based on the lognormal distribution. The semi-parametric nature of the Approximate Model, while bolstering versatility, unfortunately precludes the application of likelihood-based model selection tools such as AIC.

Recall that the Approximate Model is only applicable to $\mathbf{x}$-homogeneous pools and can perform poorly when applied to heterogeneous pools (see Table 3.5). Thus far, we have

only discussed parametric models for analyzing heterogeneous pools; specifically, direct maximum likelihood under the gamma^2 model or the MCEM algorithm applied to lognormal or gamma^1 data. In the next section, we propose a semi-parametric model based on quasi-likelihood methods that provides a straightforward and flexible framework for analyzing **x**-heterogeneous pools.

4.3.2 Quasi-Likelihood Models

The application of quasi-likelihood methods in this context is particularly compelling due to a close connection to the gamma distribution under certain conditions (Wedderburn, 1974). To estimate QL regression coefficient estimates on the individual-level data, the following QL score equations are solved with respect to $\boldsymbol{\beta}^* = (\alpha, \boldsymbol{\beta})$:

$$\sum_{i=1}^{n}\sum_{j=1}^{k_i} \frac{Y_{ij}-\mu_{ij}}{\phi V(\mu_{ij})}\left(\frac{d\mu_{ij}}{d\boldsymbol{\beta}^*}\right) = 0 \tag{4.4}$$

where μ_{ij} is some known function of $\boldsymbol{\beta}^*$ and $V(\mu_{ij})$ is a known function of μ_{ij}. Assuming a constant coefficient of variation, i.e. $V(\mu_{ij}) = \mu_{ij}^2$, these QL score equations are equivalent to the ML score equations under the gamma^1 model, where the ML score equations are defined as:

$$\begin{aligned}
S(\boldsymbol{\beta}^*) &= \sum_{i=1}^{n}\sum_{j=1}^{k_i} \frac{d}{d\boldsymbol{\beta}^*}\log f(Y_{ij}|\mathbf{x}_{ij},\boldsymbol{\beta}^*,a) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k_i} \frac{d}{d\boldsymbol{\beta}^*}\left[-\log\Gamma(a) - a\log(a^{-1}\mu_{ij}) - aY_{ij}/\mu_{ij} + (a-1)\log Y_{ij}\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k_i} a\left(\frac{Y_{ij}}{\mu_{ij}^2} - \frac{1}{\mu_{ij}}\right)\left(\frac{d\mu_{ij}}{d\boldsymbol{\beta}^*}\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k_i} \frac{Y_{ij}-\mu_{ij}}{a^{-1}\mu_{ij}^2}\left(\frac{d\mu_{ij}}{d\boldsymbol{\beta}^*}\right) = 0
\end{aligned} \tag{4.5}$$

Thus the $\hat{\boldsymbol{\beta}}^* = (\hat{\alpha}, \hat{\boldsymbol{\beta}})$ estimates calculated under (4.4) and (4.5) will be identical, since ϕ and a have no effect on the solution of these equations with respect to $\boldsymbol{\beta}^*$. We will refer to the QL model with log link and $V(\mu) = \mu^2$ as the QL^1 model.

While the ML gamma[1] model and the QL[1] model provide identical coefficient estimates, the main difference between these two approaches is the estimation of the standard errors. Under the gamma[1] model, $\hat{a}$ is the maximum likelihood estimate (MLE), whereas, under QL[1], the estimate for ϕ is moment-based:

$$\tilde{\phi} = \frac{1}{n-(P+1)} \sum_{i=1}^{n} \sum_{j=1}^{k_i} \frac{(Y_{ij} - \hat{\mu}_{ij})^2}{\hat{\mu}_{ij}^2}$$

where P is the number of covariates in the model (not including the intercept) and $\hat{\mu}_{ij} = \exp(\hat{\alpha} + \boldsymbol{x}_{ij}\hat{\boldsymbol{\beta}})$. One practical note is that the default standard errors calculated under R's *glm* function with the gamma distribution and log link are based on the QL estimate of the dispersion parameter, whereas SAS PROC GENMOD will calculate standard errors based on ML estimates of a.

The QL[1] model is also convenient in that its mean and variance assumptions also apply to a lognormally distributed outcome. Let Y_{ij} be lognormally distributed such that $E(\log Y_{ij}) = \alpha + \mathbf{x}_{ij}\boldsymbol{\beta}$ and $Var(\log Y_{ij}) = \sigma^2$. Then, from the properties of the lognormal distribution,

$$E(Y_{ij}) = e^{\alpha + \mathbf{x}_{ij}\boldsymbol{\beta} + \sigma^2/2}$$

and

$$Var(Y_{ij}) = (e^{\sigma^2} - 1)E(Y_{ij})^2.$$

Note that since $\log(\mu_{ij}) = \log E(Y_{ij}) = (\alpha + \sigma^2/2) + \mathbf{x}_{ij}\boldsymbol{\beta}$, applying a log link will preserve $\boldsymbol{\beta}$ since the additional $\sigma^2/2$ term will be absorbed into the intercept estimate. Similarly, the $(e^{\sigma^2} - 1)$ expression in the variance of Y_{ij} will be absorbed into the dispersion parameter, so that the mean-variance relationship from the QL[1] model ($V(\mu) = \mu^2$) is appropriate. Thus, estimates of $\boldsymbol{\beta}$ as well as their standard errors should remain valid when the QL[1] model is fit to a lognormally distributed outcome.

This quasi-likelihood method is easily extended to **x**-homogeneous pools. When pools

are $\mathbf{x}$-homogeneous, such that $\mathbf{x}_{ij} = \mathbf{x}_i$ for all $j = 1, \ldots, k_i$, then

$$\mu_i = E(Y_i^p) = \frac{1}{k_i} \sum_{j=1}^{k_i} E(Y_{ij}) = e^{\alpha + \mathbf{x}_i \boldsymbol{\beta}}$$

and

$$Var(Y_i^p) = \frac{1}{k_i^2} \sum_{j=1}^{k_i} Var(Y_{ij}) = \frac{1}{k_i^2} \sum_{j=1}^{k_i} \mu_{ij}^2 = k_i^{-1} e^{2(\alpha + \mathbf{x}_i \boldsymbol{\beta})} = \mu_i^2 / k_i$$

Estimates for $\boldsymbol{\beta}^* = (\alpha, \boldsymbol{\beta})$ are then calculated by solving the QL score equations:

$$\sum_{i=1}^{n} \frac{k_i (Y_i^p - \mu_i)}{\phi \mu_i^2} \left(\frac{d\mu_i}{d\boldsymbol{\beta}^*} \right) = 0, \tag{4.6}$$

Just as $\mathbf{x}$-homogeneous pools evaluated under the gamma^1 model will give identical estimates to those calculated under the full gamma^1 model, regression estimates calculated from (4.6) will also be identical to those calculated under the QL^1 model applied to the full data. To see this, consider the individual-level QL score equations. Under $\mathbf{x}$-homogeneous pools, $\mu_{ij} = \mu_i$ for all $j = 1, \ldots, k_i$, so that (4.4) reduces to:

$$\sum_{i=1}^{n} \sum_{j=1}^{k_i} \frac{Y_{ij} - \mu_i}{\phi V(\mu_i)} \left(\frac{d\mu_i}{d\boldsymbol{\beta}^*} \right) = \sum_{i=1}^{n} \frac{k_i (Y_i^p - \mu_i)}{\phi \mu_i^2} \left(\frac{d\mu_i}{d\boldsymbol{\beta}^*} \right) = 0,$$

which is identical to the pool-wise QL score equations in (4.6). This property means that, so long as pools are formed homogeneously on the covariates values, and one can validly assume that pooled assay measurements equal the arithmetic mean of their constituents, the precision of the QL $\boldsymbol{\beta}^*$ estimates will be maintained regardless of the total number of pools. Again, we do not advise pooling all individuals with identical covariate values, since the validity of the standard error estimates depends on asymptotic properties that are lost when the total number of pools is small.

Perhaps the most valuable aspect of this semi-parametric approach is the accessibility of analyzing heterogeneous pools when an outcome is right-skewed. For $\mathbf{x}$-heterogeneous pools, we extend the traditional QL framework to solve the following score equations with

respect to $\boldsymbol{\beta}^* = (\alpha, \boldsymbol{\beta})$:

$$U(\boldsymbol{\beta}^*) = \sum_{i=1}^{n} \frac{(Y_i^p - \mu_i)}{Var(Y_i^p)} \frac{d\mu_i}{d\boldsymbol{\beta}^*} = \sum_{i=1}^{n} \frac{(Y_i^p - \mu_i)}{\phi V(\boldsymbol{\beta}^*)} \frac{d\mu_i}{d\boldsymbol{\beta}^*} = 0 \tag{4.7}$$

where

$$\begin{aligned} \mu_i = E(Y_i^p) &= k_i^{-1} \sum_{j=1}^{k_i} E(Y_{ij}) = k_i^{-1} \sum_{j=1}^{k_i} e^{\alpha + \mathbf{x}_{ij}\boldsymbol{\beta}} \\ V(\boldsymbol{\beta}^*) &\propto k_i^{-2} \sum_{j=1}^{k_i} Var(Y_{ij}) = (k_i^{-2}/a) \sum_{j=1}^{k_i} e^{2(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})} \propto k_i^{-2} \sum_{j=1}^{k_i} e^{2(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})} \end{aligned}$$

and

$$\frac{d\mu_i}{d\boldsymbol{\beta}^*} = k_i^{-1} \sum_{j=1}^{k_i} (\mathbf{x}_{ij}^*)^T e^{\alpha + \mathbf{x}_{ij}\boldsymbol{\beta}}$$

Note that the format of (4.7) is slightly different from classical QL estimation, since $Var(Y_i^p)$ is a function of $(\alpha, \boldsymbol{\beta})$ rather than a function of μ_i. Under suitable regularity conditions, estimates of $(\alpha, \boldsymbol{\beta})$ will remain consistent and asymptotically normal so long as the assumptions on the individual level data hold, namely, that $E(Y_{ij}) = \mu_{ij} = \exp(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})$ and $Var(Y_{ij}) \propto \mu_{ij}^2$ (Huber, 1964; McCullagh, 1983; Wedderburn, 1974).

Computationally, (4.7) can be solved numerically using nonlinear equation solvers such as the *nleqslv* function in R (R Development Core Team, 2012). An example of this code is provided in Appendix A.3. Since the shape parameter in the variance function is a constant, it will be absorbed by ϕ as an estimate of the dispersion, so it can be omitted from the specification of the QL score equations. Estimates of the standard errors of the resulting QL estimates $\hat{\boldsymbol{\beta}}_{QL}$ are found by taking the square root of the diagonal of the dispersion matrix D, where:

$$D = \left[-\sum_{i=1}^{n} U'(\boldsymbol{\beta}^*) \right]^{-1} = \left[-\frac{1}{\phi} \sum_{i=1}^{n} \frac{d}{d\boldsymbol{\beta}^*} \left(\frac{Y_i^p - \mu_i}{V(\boldsymbol{\beta}^*)} \frac{d\mu_i}{d\boldsymbol{\beta}^*} \right) \right]^{-1}$$

where D is evaluated at the QL estimates $\hat{\boldsymbol{\beta}}_{QL}^*$ and the moment estimate of ϕ (Wedderburn, 1974). D/ϕ can be readily obtained by specifying the *jacobian* option in the *nleqslv* R function. Since the dispersion parameter is not included in the estimation procedure for

$\hat{\boldsymbol{\beta}}^*_{QL}$, D/ϕ must be multiplied by the estimate $\tilde{\phi}$ in order to get accurate standard error estimates. While appropriate estimates of the dispersion matrix will be obtained so long as the variance function is correctly specified, a more robust estimator of the standard errors can be obtained using a sandwich estimator if the specified mean-variance relationship is uncertain (White, 1982).

Similar QL methods could be applied to pooled data under different assumptions for the first two moments. In this paper, we will limit our study to the QL model with constant CV, as well as one with a linear mean-variance relationship. The latter is considered in order to match the mean-variance relationship under the gamma^2 model. Similar QL equations are required for this model under heterogeneous pools, but now

$$Var(Y_i^p) = k_i^{-2} \sum_{j=1}^{k_i} Var(Y_{ij}) = (k_i^{-2}/a) \sum_{j=1}^{k_i} e^{(\alpha + \mathbf{x}_{ij}\boldsymbol{\beta})}$$

We will refer to this QL model with linear mean-variance structure as QL^2.

The QL^2 model does not have the same relationship with the gamma^2 model as the QL^1 enjoys with the ML gamma^1 model. The QL^1 extension of the gamma^1 model is special, since coefficient estimates for this model are identical to the MLEs for full and $\mathbf{x}$-homogeneous pools. That is, for the gamma^1 model, no precision is lost by specifying only the first two moments of Y_i^p as opposed to the entire distribution. While this relationship does not hold between the QL^2 and gamma^2 models, the QL^2 model may still provide a nice alternative to the ML-based gamma^2 model when a linear mean-variance relationship may yield a better model for the data.

These quasi-likelihood based methods are quite useful in providing more flexible models than their ML-based counterparts, due to fewer required assumptions. They are also helpful in offering a convenient and straightforward analytical procedure for heterogeneous pools, particularly when compared with the MCEM methods used to calculate MLEs under certain distributions. The QL^1 model is especially compelling since its coefficient estimates match the precision of the MLEs under the gamma^1 model exactly. Aside from this anomaly, the semi-parametric models described in this section will generally suffer a loss of efficiency relative to the efficiency potential under a correctly-specified maximum-likelihood based

model. From classical theory, we know that MLEs calculated from a correctly-specified parametric model will maximize precision of the coefficient estimates. In the next section, we consider a criterion for choosing the best parametric model, when maximizing precision is of utmost importance.

4.4 Model Selection

As illustrated by Figure 4.1, a lognormal distribution may provide a reasonable fit for data generated under a gamma^1 distribution, but not under the gamma^2 model. Similarly, the gamma^2 model may not be appropriate to model data from the lognormal or gamma^1 distributions.

In order to choose the best model among these three alternatives, we apply an AIC-based information criterion to both full and pooled data to compare models under different distributional assumptions. Details concerning the proper use of AIC to select between models with differing probability distributions can be found in Burnham and Anderson (2002). For full data or a random sample of individual specimens, AIC is defined as:

$$AIC = -2\log L(\boldsymbol{\theta}) + 2K = -2\sum_{i=1}^{n}\sum_{j=1}^{k_i}\log f(Y_{ij}|\mathbf{x}_{ij},\boldsymbol{\theta}) + 2K$$

where K refers to the total number of estimated parameters in the model, and a lower value of AIC indicates better fit (Akaike, 1974). An important note is that in order to compare these values between the gamma and lognormal models, the data must be on the same scale. For instance, the AIC under a normal model for $\log(\mathbf{Y})$ will be different from that under the lognormal model on the untransformed $\mathbf{Y}$.

Under both gamma models, we can insert the observed likelihood into the AIC equation when pools are $\mathbf{x}$-homogeneous, since these pools retain a fully-specified gamma distribution. A closed form likelihood is available for the gamma^2 model under heterogeneous pools as well. One practical note is that when estimates are obtained under a weighted regression function, as in the gamma^1 model, the *glm* function in R multiplies the given weight by the log-density contribution for each observation, whereas the GENMOD procedure in SAS

standardizes the dispersion parameter for each density by the given weight. While these techniques provide identical coefficient parameters, the given value of the AIC varies. Care must be taken to maintain a similar treatment of all AIC values in order to ensure proper comparison. Furthermore, since the default estimate for the dispersion parameter in R's glm function is based on the moment estimate, the MLE of this parameter must be calculated separately in order to create a true log-likelihood to be used in the AIC function under the gamma^1 model for both individual-level data as well as $\mathbf{x}$-homogeneous pools.

For models that employ the MCEM algorithm to calculate MLEs, such as the lognormal model applied to pooled outcomes or the gamma^1 model fit to heterogeneous pools, we recommend a Monte Carlo estimation of the observed log-likelihood. For these models, $f(Y_i^p)$ can be approximated by:

$$\begin{aligned} f(Y_i^p) &= \int_{Y_{ik_i}} \cdots \int_{Y_{i2}} f(Y_i^p, Y_{i2}, \ldots, Y_{ik_i}) d\mathbf{Y}_{i[\text{-}1]} \\ &= \int_{Y_{ik_i}} \cdots \int_{Y_{i2}} k_i f\left(k_i Y_i^p - \sum_{j=2}^{k_i} Y_{ij} | \mathbf{x}_{i1}, \boldsymbol{\theta}\right) I\left(\sum_{j=2}^{k_i} Y_{ij} < k_i Y_i^p\right) \prod_{j=2}^{k_i} f(Y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\theta}) d\mathbf{Y}_{i[\text{-}1]} \\ &= E_{\mathbf{Y}_{i[\text{-}1]}}\left[k_i f\left(k_i Y_i^p - \sum_{j=2}^{k_i} Y_{ij} | \mathbf{x}_{i1}, \boldsymbol{\theta}\right) I\left(\sum_{j=2}^{k_i} Y_{ij} < k_i Y_i^p\right)\right], \end{aligned} \tag{4.8}$$

Estimating this value with Monte Carlo methods is straightforward once the MLEs have been calculated, as we can approximate (4.8) with

$$f(Y_i^p) \approx \frac{k_i}{M} \sum_{m=1}^{M} \left[f\left(k_i Y_i^p - \sum_{j=2}^{k_i} Y_{ij,m} | \mathbf{x}_{i1}, \hat{\boldsymbol{\theta}}\right) I\left(\sum_{j=2}^{k_i} Y_{ij,m} < k_i Y_i^p\right)\right] \tag{4.9}$$

where each $Y_{ij,m}$ is generated from $f(y_{ij}|\mathbf{x}_{ij}, \hat{\boldsymbol{\theta}})$ for $m = 1, \ldots, M$, and $\hat{\boldsymbol{\theta}}$ denotes the MLE. A similar concept concerning this type of MC estimation is applied in Dupuis et al. (2007). By the law of large numbers, this approximation will converge to $f(Y_i^p)$ for large M. In our simulations, we set M = 10,000, as this number proved sufficiently large.

Table 4.1 shows simulation results for the frequency of model selection when data was generated and fit under each of the three distributions: (1) lognormal (2) gamma^1 and

(3) gamma^2, for various pooling types. Simulation conditions are described in detail in Section 4.5. For all pooling types fit under the gamma^2 model as well as non-heterogeneous pools fit to the gamma^1 model, the observed log-likelihood was used to calculate AIC, since all pooled outcomes retain gamma distributions in these scenarios. All pools under the lognormal model and heterogeneous pools under the gamma^1 model are based on the MCEM algorithm, where the MC approximation technique from (4.9) was used to estimate the log-likelihood. For the lognormal model based on the full dataset or a random sample, the additional assumption of normality is applied to the errors in order to obtain a log-likelihood. Recall that, although the Approximate Model provides valid estimates of $\boldsymbol{\beta}$ when pools are $\mathbf{x}$-homogeneous, this model is not eligible to produce AIC values, since it is based on weighted least squares instead of maximum likelihood analysis. Thus, the MCEM method must be used to calculate MLEs and the Monte-Carlo approximate AIC when a lognormal model is fit to any pooled data. In general, if pool size does not exceed 2, the Convolution Method can also be applied to calculate MLEs and AIC value based on the observed log-likelihood. For this simulation study, however, all simulated datasets contained pool sizes greater than 2, so the Convolution-based MLEs and AIC values were unavailable.

As indicated by this simulation study, AIC tends to provide a fairly effective measure for selecting the best distribution, correctly choosing the lognormal distribution over 95% of the time, and accurately identifying the best gamma model over 60% of the time. Furthermore, our MC approximation techniques tend to perform as well as the closed-form AIC under a random sample of the same number of pools. This result suggests that the ability of AIC to select the best model is not hindered by the additional MC methods applied to approximate this value under models that require MCEM methods to calculate MLEs. Needless to say, this study represents only one instance of AIC's ability to select the best model. In other situations, the lognormal and gamma^1 models may fit certain datasets similarly well; while AIC may not be as effective in selecting the true underlying distribution in such cases, it is more likely that both of these models may provide equally appropriate fit.

The benefit of applying a model-selection technique like AIC is to optimize precision of the regression coefficient estimates by selecting the best distribution for modeling the

Table 4.1: Frequency of model selection based on AIC.

			Fit Model		
True Model	n	Sampling Method	lognormal	gamma^1	gamma^2
lognormal	651	Full Dataset	100.0	0.0	0.0
	404	Random Sample	100.0	0.0	0.0
		x-homogeneous Pools	99.9	0.1	0.0
	150	Random Sample	97.0	2.6	0.4
		x-heterogeneous Pools	95.7	3.0	1.2
gamma^1	651	Full Dataset	0.0	80.4	19.6
	404	Random Sample	0.0	73.5	26.5
		x-homogeneous Pools	0.0	76.8	23.2
	150	Random Sample	1.2	60.3	38.5
		x-heterogeneous Pools	3.9	66.6	29.5
gamma^2	651	Full Dataset	0.0	12.5	87.5
	404	Random Sample	0.0	19.5	80.5
		x-homogeneous Pools	0.0	17.8	82.2
	150	Random Sample	0.7	33.6	65.7
		x-heterogeneous Pools	1.9	27.4	70.7

data. In the next section, we use simulation studies to assess the effects of using AIC as the only tool to select a model, and compare these results to those fit under the true underlying distribution. We also compare these ML-based methods to those from the semi-parameteric methods discussed in Section 4.3. Based on well-known characteristics of maximum likelihood, we expect the ML models to provide the most precise estimates when correctly specified. Yet we are also curious as to the potential consequences of distributional misspecification, as well as the effectiveness of relying on AIC to choose the best distribution.

While, in general, semi-parametric models are expected to have less potential precision than the fully parametric models, they provide a more flexible framework when full specification of the outcome distribution is dubious. Additionally, the typical lapse in precision as a result of weaker assumptions is overcome in part by the close relationship between the QL^1 and gamma^1 models. Thus, we will use the following simulation studies to more closely examine the trade-offs between potential precision gains under parametric models and the increased flexibility of the semi-parametric models.

4.5 Simulation Study

For each of the simulation studies, 5000 simulations were performed in R. Datasets were generated to resemble the data from the CPP study discussed in Section 3.2, but this time treating the interferon gamma inducible protein (IP) as the outcome. Predictor variables were simulated to mimic covariates from the CPP data, namely, X_1 = age (years), X_2 = smoking status (yes/no), X_3 = race (1 = white / 2 = black), and X_4 = SA status (yes/no). X_1 was generated from a normal distribution with a mean of 27 and standard deviation of 6.5, then rounded to the nearest whole number. X_2, X_3, and X_4 were each generated from Bernoulli distributions with probabilities 0.47, 0.30, and 0.45. The simulated outcome (Y) was based on estimates from a generalized linear model fit to the individual-level IP data, such that

$$\eta = -4.26 + 0.012(\text{Age}) + 0.022(\text{Smoking Status}) + 0.169(\text{Race}) - 0.091(\text{SA status}).$$

For all simulations, three different sets of outcomes with sample size $N = 651$ were generated under the following distributions:

1. $\log(Y) \sim N(\eta, 1)$
2. $Y \sim gamma^1(1.24, \exp(\eta)/1.24)$
3. $Y \sim gamma^2(\exp(\eta)/0.02, 0.02)$

For each of the following simulations, datasets were fit under the full data, **x**-homogeneous pools, and heterogeneous pools. **x**-homogeneous pools were formed based on the description in Section 3.8.1, and **x**-heterogeneous pools were formed via k-means clustering on the standardized versions of all the covariates. For the full data, the *glm* function in R was applied to the lognormal and gamma^1 models. Since the default gamma regression in R provides standard errors based on the QL estimated dispersion parameter, the *gamma.dispersion* function was used to calculate the MLE for the shape parameter, then applied to the information matrix to obtain the ML-based standard error estimates for the gamma^1 model. For the quasi-likelihood methods under full data and **x**-homogeneous pools, the "quasi"

family under the *glm* function was specified, with variance function appropriate to the desired model. For heterogeneous pools from the QL^1 and QL^2 models, R's nonlinear solver *nleqslv* was applied to (4.7) to calculate regression coefficient estimates.

To calculate MLEs under the gamma^2 model, the *optim* function with box constraints in R was used. To speed convergence, starting values for the regression coefficients were set as the weighted least squares estimates, and the analytical gradient function was passed directly to the optimization procedure. Let $a_i = b^{-1}\sum_{j=1}^{k_i}\exp(\alpha+\mathbf{x}_{ij}\boldsymbol{\beta})$ so that $a_i'(\boldsymbol{\beta}^*) = b^{-1}\sum_{j=1}^{k_i}\mathbf{x}_{ij}^{*\,T}\exp(\alpha+\mathbf{x}_{ij}\boldsymbol{\beta})$ and $a_i'(b) = -b^{-2}\sum_{j=1}^{k_i}\exp(\alpha+\mathbf{x}_{ij}\boldsymbol{\beta})$ are the derivatives of a_i with respect to $\boldsymbol{\beta}^* = (\alpha, \boldsymbol{\beta})$ and b, respectively. Then the gradient is:

$$\frac{dl}{d\boldsymbol{\theta}} = \begin{pmatrix} l'(\boldsymbol{\beta}^*) \\ l'(b) \end{pmatrix} = \begin{pmatrix} a_i'(\boldsymbol{\beta}^*)[\log(k_i Y_i^p/b) - \psi(a_i)] \\ a_i'(b)[\log(k_i Y_i^p/b) - \psi(a_i)] - a_i/b + b^{-2}k_i Y_i^p \end{pmatrix}$$

For the first simulation, the outcome is generated from the lognormal distribution specified above. The next simulation treats data generated from the gamma^1 distribution, and the final simulation generates outcomes from the gamma^2 distribution. Each of the models discussed in Sections 4.2 and 4.3 were applied under each scenario, with the exception of the Approximate Model, which is only applicable under $\mathbf{x}$-homogeneous pools. At each simulation, we also calculate AIC values, applying the proposed Monte Carlo approximation when necessary, and calculate coefficient estimates under the selected best-fitting distribution. For each simulation, AIC values were calculated according to Section 4.4, and the model with lowest AIC was selected as the best model. Results from these models chosen by AIC are also provided in order to assess performance when AIC is the only measure of fit considered. Table 4.2 gives a summary of each of the simulations performed and the estimation procedures applied under each scenario. Note that although a semi-parametric least squares estimation is available under the lognormal model applied to the full data, we assume normality of the errors in order to calculate AIC values, resulting in the ML-based parametric model.

Table 4.2: Summary of simulations, including analytical methods performed under each assumed model. (W)LS = (Weighted) Least Squares, ML = Maximum Likelihood, QL = Quasi-Likelihood, MCEM = Monte Carlo Expectation Maximization

	Model					
Pooling Method	lognormal	gamma^1	gamma^2	Approx.	QL^1	QL^2
Full Data	LS†/ML*	ML*	ML*	–	QL†	QL†
x-homogeneous Pools	MCEM*	ML*	ML*	WLS†	QL†	QL†
x-heterogeneous Pools	MCEM*	MCEM*	ML*	–	QL†	QL†

* Parametric (Maximum-Likelihood) Methods

† Semi-Parameteric (Quasi-Likelihood) Methods

4.5.1 Lognormal

Table 4.3 provides mean bias, ratio of estimated standard error ($\hat{\text{SE}}$) to empirical standard deviation (SD), and 95% CI coverage, to assess the validity of each of the models when the underlying distribution is lognormal. A valid estimation procedure will exhibit a mean bias close to 0, nominal 95% confidence interval (CI) coverage, and $\hat{\text{SE}}$/SD close to 1.

As expected, since the lognormal models correctly specify the underlying distribution, they provide ideal estimates, with bias close to 0, mean estimated standard errors close to empirical standard deviation, and close to nominal 95% CI coverage. Results from choosing an ML model based on the AIC values performs almost identically to these lognormal models, due to the AIC's ability to accurately select the lognormal distribution, as evidenced in Table 4.1.

Table 4.3 shows that the Approximate Model (AP) performs well when applied to **x**-homogeneous pools, validating this semi-parametric model as an alternate analytical strategy to MCEM when pools are formed with identical covariates. Both QL^1 and QL^2 also provide approximately unbiased estimates, since these models assume the correct link function corresponding to lognormal regression. QL^1 slightly outperforms QL^2 with respect to estimating standard errors, since the QL^1 model assumes a constant CV, which is characteristic of the lognormal distribution, while the QL^2 model misspecifies the mean-variance relationship as linear.

While the misspecified gamma^2 model tends to produce biased estimates, gamma^1 performs identically to the QL^1 model with respect to bias, as a consequence of the relationship

Table 4.3: Simulation results comparing regression models applied to a lognormal outcome. "SD" refers to empirical standard deviation and "$\widehat{\text{SE}}$" represents the mean estimated standard errors. Model fit codes are LN = lognormal, GA^1 = gamma^1, GA^2 = gamma^2, AIC = AIC-selected model, QL^1 = Quasi-likelihood with constant CV, QL^2 = Quasi-likelihood with linear mean-variance structure, AP = Approximate Model.

	Mean Bias ($\widehat{\text{SE}}$/SD), 95% CI Coverage			
	β_1	β_2	β_3	β_4
		Full Data ($n = 651$)		
LN	0.000 (1.00), 94.8	0.000 (1.00), 95.3	0.000 (1.00), 95.1	0.001 (0.99), 94.8
GA^1	0.000 (0.73), 84.5	0.000 (0.72), 85.0	-0.003 (0.73), 84.4	0.000 (0.72), 83.8
GA^2	-0.004 (1.16), 88.5	-0.008 (1.16), 97.7	-0.059 (1.13), 87.9	0.033 (1.15), 94.6
AIC	0.000 (1.00), 94.8	0.000 (1.00), 95.3	0.000 (1.00), 95.1	0.001 (0.99), 94.8
QL^1	0.000 (0.99), 94.6	0.000 (0.98), 94.5	-0.003 (0.98), 94.4	0.000 (0.97), 94.3
QL^2	0.000 (0.98), 94.7	0.000 (0.97), 94.5	-0.002 (0.97), 95.0	0.001 (0.96), 94.6
		x-homogeneous Pools ($n = 404$)		
LN	0.000 (1.00), 94.7	0.000 (1.00), 95.0	0.000 (0.99), 95.3	0.000 (0.99), 94.4
GA^1	0.000 (0.77), 86.5	0.000 (0.76), 86.8	-0.003 (0.76), 86.6	0.000 (0.75), 86.0
GA^2	-0.004 (1.13), 92.0	-0.004 (1.09), 96.5	-0.034 (1.07), 94.4	0.027 (1.08), 94.8
AIC	0.000 (1.00), 94.7	0.000 (1.00), 95.0	0.000 (0.99), 95.3	0.000 (0.99), 94.4
AP	0.000 (1.01), 94.7	0.000 (0.98), 94.7	0.001 (0.98), 95.0	0.001 (0.97), 94.2
QL^1	0.000 (0.98), 94.5	0.000 (0.97), 94.2	-0.003 (0.98), 94.2	0.000 (0.96), 94.1
QL^2	0.000 (0.96), 94.5	0.000 (0.96), 94.4	-0.003 (0.95), 93.7	0.000 (0.96), 94.3
		x-heterogeneous Pools ($n = 150$)		
LN	0.000 (0.99), 94.5	-0.001 (0.98), 94.3	-0.003 (0.98), 94.7	0.000 (0.96), 94.0
GA^1	0.000 (0.83), 89.0	0.000 (0.82), 89.3	-0.003 (0.82), 89.0	0.000 (0.81), 88.6
GA^2	-0.002 (1.07), 94.3	-0.002 (0.98), 94.3	-0.013 (0.97), 94.0	0.015 (0.97), 93.9
AIC	0.000 (0.98), 94.2	-0.001 (0.98), 94.0	-0.004 (0.97), 94.5	0.000 (0.96), 93.8
QL^1	0.000 (0.98), 94.0	0.000 (0.96), 93.6	-0.003 (0.97), 93.6	0.000 (0.96), 93.7
QL^2	0.000 (0.96), 94.2	0.000 (0.96), 93.7	-0.003 (0.94), 93.3	0.000 (0.96), 94.0

between these two models detailed in Section 4.3.2. Note, however, that the gamma^1 model noticeably underestimates standard errors, resulting in confidence interval undercoverage. This difference between the performance of the gamma^1 and QL^1 models highlights the flexibility of the QL-based models in providing valid inference for data generated under various distributions, so long as the specification of the first two moments is correct.

4.5.2 Gamma^1

For an outcome generated under the gamma^1 distribution, the parametric lognormal and gamma^1 models tend to perform well (Table 4.4). The semi-parametric models (Approximate Model, QL^1, and QL^2) perform quite well in estimating the $\boldsymbol{\beta}$ vector as well as its standard errors, even though the mean-variance relationship under QL^2 is misspecified. Once again, the gamma^2 model, which requires full distributional specification under the alternate parameterization of the gamma distribution, fails to provide adequate analysis, resulting in biased estimates and sub-optimal confidence interval coverage.

Results from the model chosen from AIC values provide approximately unbiased estimates of the regression coefficients, but tend to slightly underestimate standard errors, likely due to the occasional misspecification of the true distribution. Thus, it may be preferable to fit one of the semi-parametric models, or perhaps to invoke a more robust method for estimating standard errors (such as bootstrap or a sandwich estimator) in order to guard against this flaw (Efron, 1979; White, 1982).

One interesting point is that when the outcome is lognormally distributed, as in Table 4.3, the gamma^1 model noticeably underestimates the SE estimates, while SE's estimated under the QL^1 model are close to the empirical standard deviation. When the underlying distribution is in fact gamma^1, however, the gamma^1 and QL^1 models provide, on average, nearly identical SE estimates. This characteristic suggests that when analyzing a single dataset, considerable differences between the standard error estimates under the gamma^1 vs. QL^1 models may indicate that the gamma^1 model does not provide an ideal fit.

Table 4.4: Simulation results comparing regression models applied to an outcome generated from a gamma[1] distribution. "SD" refers to empirical standard deviation and "$\hat{\text{SE}}$" the mean estimated standard errors. Model fit codes are LN = lognormal, GA[1] = gamma[1], GA[2] = gamma[2], AIC = AIC-selected model, QL[1] = Quasi-likelihood with constant CV, QL[2] = Quasi-likelihood with linear mean-variance structure, AP = Approximate Model.

	Mean Bias ($\hat{\text{SE}}$/SD), 95% CI Coverage			
	β_1	β_2	β_3	β_4
	Full Data ($n = 651$)			
LN	0.000 (0.99), 94.9	-0.001 (0.99), 94.8	-0.003 (1.01), 95.3	0.000 (0.99), 94.9
GA[1]	0.000 (0.99), 94.9	0.000 (0.98), 94.7	-0.003 (1.00), 95.0	-0.001 (0.99), 94.8
GA[2]	-0.004 (0.98), 85.0	-0.008 (0.98), 94.4	-0.057 (0.98), 84.5	0.030 (0.99), 91.7
AIC	0.000 (0.94), 92.8	-0.001 (0.94), 93.6	-0.007 (0.94), 92.9	0.002 (0.94), 93.5
QL[1]	0.000 (0.99), 94.8	0.000 (0.98), 94.6	-0.003 (1.00), 94.8	-0.001 (0.99), 94.5
QL[2]	0.000 (1.00), 94.9	0.000 (0.98), 94.4	-0.002 (0.99), 94.8	-0.001 (0.99), 94.8
	x-homogeneous Pools ($n = 404$)			
LN	0.000 (0.97), 94.4	-0.002 (1.01), 95.4	-0.008 (1.01), 95.0	-0.002 (1.01), 95.2
GA[1]	0.000 (0.99), 94.8	0.000 (0.98), 94.7	-0.003 (1.00), 94.9	-0.001 (0.98), 94.6
GA[2]	-0.003 (0.98), 89.4	-0.005 (0.98), 94.8	-0.041 (0.97), 90.2	0.020 (0.99), 93.4
AIC	0.000 (0.95), 93.4	-0.001 (0.95), 93.6	-0.006 (0.95), 93.4	0.001 (0.95), 93.5
AP	0.000 (0.99), 94.7	-0.001 (1.01), 95.6	-0.002 (1.02), 95.1	-0.001 (1.02), 95.5
QL[1]	0.000 (0.99), 94.6	0.000 (0.98), 94.6	-0.003 (1.00), 94.7	-0.001 (0.99), 94.5
QL[2]	0.000 (0.99), 94.6	-0.001 (0.98), 94.5	-0.003 (0.97), 94.2	-0.001 (0.99), 94.8
	x-heterogeneous Pools ($n = 150$)			
LN	0.000 (0.96), 94.1	0.000 (1.02), 95.5	-0.005 (1.02), 95.5	-0.001 (1.03), 95.5
GA[1]	0.000 (0.98), 94.2	0.000 (0.97), 94.4	-0.003 (0.98), 94.5	-0.001 (0.97), 94.4
GA[2]	-0.002 (0.98), 92.6	-0.002 (0.97), 94.4	-0.020 (0.96), 93.4	0.008 (0.99), 94.4
AIC	0.000 (0.95), 93.2	-0.001 (0.95), 93.7	-0.004 (0.96), 93.6	0.000 (0.96), 93.9
QL[1]	0.000 (0.99), 94.5	0.000 (0.98), 94.6	-0.003 (1.00), 94.6	-0.001 (0.98), 94.5
QL[2]	0.000 (0.99), 94.6	-0.001 (0.98), 94.6	-0.003 (0.97), 94.0	-0.001 (1.00), 94.9

Table 4.5: Simulation results comparing regression models applied to an outcome generated from a gamma^2 distribution. "SD" refers to empirical standard deviation and "$\widehat{\text{SE}}$" represents the mean estimated standard errors. Model fit codes are LN = lognormal, GA^1 = gamma^1, GA^2 = gamma^2, AIC = AIC-selected model, QL^1 = Quasi-likelihood with constant CV, QL^2 = Quasi-likelihood with linear mean-variance structure, AP = Approximate Model.

	Mean Bias ($\widehat{\text{SE}}$/SD), 95% CI Coverage			
	β_1	β_2	β_3	β_4
	Full Data ($n = 651$)			
LN	0.008 (1.00), 82.4	0.014 (0.99), 94.4	0.104 (1.06), 86.2	-0.059 (0.98), 90.7
GA^1	0.000 (0.99), 94.6	0.000 (0.99), 94.6	-0.001 (1.04), 96.0	0.000 (0.98), 94.4
GA^2	0.000 (1.00), 95.0	0.000 (0.99), 94.6	-0.001 (1.00), 95.4	0.001 (0.98), 94.8
AIC	0.001 (0.97), 94.3	0.001 (0.94), 93.8	0.005 (0.99), 95.2	-0.003 (0.94), 93.7
QL^1	0.000 (0.99), 94.6	0.000 (0.99), 94.7	-0.001 (1.03), 96.0	0.000 (0.97), 94.4
QL^2	0.000 (1.00), 94.8	0.000 (0.99), 94.6	-0.002 (1.00), 95.3	0.000 (0.98), 94.8
	x-homogeneous Pools ($n = 404$)			
LN	0.006 (0.97), 86.5	0.008 (1.01), 95.2	0.063 (1.06), 92.2	-0.039 (1.01), 93.8
GA^1	0.000 (0.98), 94.6	0.000 (0.99), 94.5	-0.001 (1.03), 95.9	0.000 (0.97), 94.2
GA^2	0.000 (0.99), 94.9	0.000 (0.98), 94.3	-0.001 (1.01), 95.6	0.001 (0.98), 94.6
AIC	0.001 (0.95), 94.0	0.001 (0.94), 93.3	0.005 (0.99), 95.2	-0.003 (0.94), 93.6
AP	0.006 (0.99), 88.0	0.009 (1.02), 95.3	0.065 (1.08), 92.3	-0.034 (1.01), 94.2
QL^1	0.000 (0.98), 94.4	0.000 (0.99), 94.5	-0.001 (1.03), 95.9	0.000 (0.97), 94.3
QL^2	0.000 (1.00), 94.8	0.000 (0.99), 94.6	-0.002 (1.00), 95.3	0.000 (0.98), 94.6
	x-heterogeneous Pools ($n = 150$)			
LN	0.004 (0.94), 90.6	0.004 (1.04), 95.3	0.027 (1.06), 95.6	-0.015 (1.03), 95.4
GA^1	0.000 (0.97), 94.3	0.000 (0.97), 94.0	-0.001 (1.01), 95.5	0.000 (0.96), 93.8
GA^2	0.000 (0.98), 94.8	0.000 (0.97), 94.2	-0.002 (0.99), 95.1	-0.001 (0.97), 94.3
AIC	0.001 (0.95), 93.9	0.001 (0.94), 93.4	0.003 (0.98), 94.6	-0.003 (0.94), 93.6
QL^1	0.000 (0.98), 94.4	0.000 (0.98), 94.1	-0.001 (1.02), 95.4	0.000 (0.96), 93.8
QL^2	0.000 (0.99), 94.6	0.000 (0.99), 94.2	-0.002 (1.00), 95.0	0.000 (0.98), 94.2

4.5.3 Gamma^2

Results from the final simulation are provided in Table 4.5, where data is generated under the gamma^2 model. When correctly specified, the gamma^2 model performs as expected, providing essentially unbiased regression estimates and appropriate CI coverage. This final simulation verifies the incompatibility between the lognormal and gamma^2 distributions, as neither perform well under reciprocal misspecification. The gamma^1 model, on the other hand, tends to perform quite well with respect to estimate validity even when the underlying distribution is gamma^2.

Both of the QL models also provide valid estimates, with the QL^2 model slightly outperforming QL^1 due to its correct specification of the variance structure. Estimates calculated under the Approximate Model, on the other hand, are noticeably biased, with bias very similar to that inherent in the lognormal model calculated under the MCEM algorithm. This result confirms our derivation in Section 4.3.1, where we showed that the Approximate Model might not be appropriate to fit data from the gamma^2 distribution.

Here we notice a similar trend from the AIC-based model as under the gamma^1 distribution. Namely, that while estimates tend to remain approximately unbiased, standard error estimates and confidence intervals tend to suffer slightly, suggesting application of a robust standard error estimation procedure.

4.5.4 Precision

Now that we have assessed the validity of each of the models under misspecification, we consider the potential efficiency gains from choosing the best model. Table 4.6 provides results on the estimate precision of each model under the three different types of pooling, where lower empirical standard deviation (SD) indicates a more precise estimation procedure. In this table, SD's that are crossed out are from models that produce invalid (i.e., biased) coefficient estimates. Since these models are not valid, their precision values are irrelevant. Although the gamma^1 model underestimates the SE's when the true model is lognormal, these precision values were not crossed out, since it is possible to calculate robust variance estimators. Thus, these poorly estimated SE's do not necessarily disqualify the gamma^1

model as a viable option in this situation.

With respect to precision, the correctly specified distribution, as expected, produces the most precise estimates (lowest SD) among the class of unbiased estimators, for all types of pooling. Here, the QL models perform extremely similarly to each other, with only minuscule differences in precision, suggesting that both variance functions may provide useful models for these simulated datasets.

When the underlying distribution is gamma^1, the relationship between the parametric gamma^1 model and the semi-parametric QL^1 model is particularly impressive. Corroborating our derivations in Section 4.3.2, precision values are identical under these two models for full data and $\mathbf{x}$-homogeneous pools, and extremely similar under heterogeneous pools, likely due to the informative pooling strategy applied via k-means clustering. These simulation results underscore the advantage of the QL^1 model, which requires fewer assumptions than its likelihood-based counterpart, yet enjoys the same precision. In addition, analyzing $\mathbf{x}$-heterogeneous pools under the QL^1 model is much simpler than calculating MLEs under the MCEM algorithm.

Using AIC to choose the model tends to perform well, and can help maintain a high level of precision, closely approximating the precision levels under the true model. Recall that models chosen from AIC can suffer from underestimation of SE's, so it may be desirable to apply robust standard error estimates, particularly when either of the gamma models are chosen.

4.5.5 Naive QL Models

In the previous simulations, both quasi-likelihood methods performed extremely similarly, indicating that for the simulation setting mimicking the motivating dataset, the choice between a linear mean-variance structure or a constant CV may not noticeably impact the results. This similarity in performance, however, will not always hold. As mentioned previously, the QL^1 model constructed under heterogeneous pools is not equivalent to applying a QL with constant CV to the pooled data. In some cases, these two different models will produce very similar results, especially when within-pool covariates are similar, such as when pools are formed via k-means clustering on all covariates. Consider a quasi-likelihood

Table 4.6: Empirical standard deviation (SD) of regression estimates under lognormal, gamma, and quasi-likelihood regression models. SD's under models that produce invalid (i.e. biased) results are crossed out, since precision under these models is irrelevant.

	Empirical Standard Deviation $SD(\hat{\beta}_1)/SD(\hat{\beta}_2)/SD(\hat{\beta}_3)/SD(\hat{\beta}_4)$		
Model	Full Data ($N = 651$)	**x**-homogeneous Pools ($n = 404$)	**x**-heterogeneous Pools ($n = 150$)
	True Model: Lognormal		
LN	0.006/0.079/0.086/0.080	0.006/0.082/0.090/0.083	0.007/0.090/0.097/0.092
GA^1	0.008/0.101/0.110/0.103	0.008/0.101/0.110/0.103	0.008/0.101/0.110/0.103
GA^2	~~0.004/0.051/0.057/0.052~~	~~0.004/0.061/0.066/0.062~~	~~0.005/0.080/0.085/0.081~~
AIC	0.006/0.079/0.086/0.080	0.006/0.082/0.090/0.083	0.007/0.090/0.097/0.092
AP	–	0.006/0.083/0.092/0.084	–
QL^1	0.008/0.101/0.110/0.103	0.008/0.101/0.110/0.103	0.008/0.101/0.110/0.103
QL^2	0.007/0.090/0.095/0.092	0.008/0.104/0.112/0.105	0.008/0.104/0.112/0.105
	True Model: Gamma^1		
LN	0.007/0.088/0.093/0.087	0.007/0.083/0.090/0.083	0.006/0.078/0.084/0.077
GA^1	0.005/0.072/0.077/0.072	0.005/0.072/0.077/0.072	0.005/0.072/0.077/0.072
GA^2	~~0.004/0.059/0.064/0.058~~	~~0.005/0.063/0.068/0.063~~	~~0.005/0.069/0.074/0.068~~
AIC	0.006/0.072/0.078/0.072	0.005/0.072/0.078/0.072	0.006/0.073/0.078/0.072
AP	–	0.006/0.081/0.090/0.082	–
QL^1	0.005/0.072/0.077/0.072	0.005/0.072/0.077/0.072	0.005/0.072/0.077/0.072
QL^2	0.006/0.076/0.080/0.076	0.005/0.072/0.077/0.072	0.005/0.072/0.077/0.072
	True Model: Gamma^2		
LN	~~0.008/0.103/0.105/0.104~~	~~0.008/0.097/0.099/0.097~~	~~0.007/0.088/0.092/0.089~~
GA^1	0.006/0.079/0.083/0.081	0.006/0.079/0.083/0.081	0.006/0.080/0.083/0.081
GA^2	0.005/0.062/0.066/0.063	0.005/0.068/0.070/0.068	0.005/0.075/0.077/0.076
AIC	0.005/0.067/0.069/0.068	0.005/0.073/0.074/0.073	0.006/0.078/0.081/0.079
AP	–	~~0.007/0.094/0.099/0.095~~	–
QL^1	0.006/0.079/0.083/0.081	0.006/0.079/0.083/0.081	0.006/0.079/0.083/0.081
QL^2	0.006/0.079/0.083/0.081	0.006/0.079/0.083/0.081	0.006/0.079/0.083/0.081

model with constant CV applied to pooled data, so that the first and second moments of the pooled outcomes are specified as:

$$E(Y_i^p) = \mu_i = e^{\alpha + \mathbf{x}_i \boldsymbol{\beta}} \quad \text{and} \quad Var(Y_i^p) = V(\mu_i) \propto e^{2(\alpha + \mathbf{x}_i \boldsymbol{\beta})} \tag{4.10}$$

where $\mathbf{x}_i = k_i^{-1} \sum_{j=1}^{k_i} \mathbf{x}_{ij}$. Let us refer to this model as the "naive" QL model. Under the QL^1 model applied to heterogeneous pools the mean and variance are adapted to incorporate the fully-known covariate information:

$$E(Y_i^p) = k_i^{-1} \sum_{j=1}^{k_i} e^{\alpha + \mathbf{x}_{ij} \boldsymbol{\beta}} \quad \text{and} \quad Var(Y_i^p) \propto k_i^{-2} \sum_{j=1}^{k_i} e^{2(\alpha + \mathbf{x}_{ij} \boldsymbol{\beta})}. \tag{4.11}$$

When within-pool covariate values are similar, such that $\mathbf{x}_{ij} \approx \mathbf{x}_i$ for all $j = 1, \ldots k_i$, then the two models are approximately equivalent. When covariates are not similar, however, these two models can produce very different results. The same characteristic also applies to the QL^2 model and the "naive" QL with linear mean-variance relationship applied to pooled data, with the exception that the variance term no longer has a '2' in the exponent.

In this next simulation, we demonstrate the potential repercussions of fitting a quasi-likelihood with moment-specifications from (4.10) when covariate values are dissimilar. We use the same simulations generated in Section 3.9.4, where $N = 400$, $X_1 \sim Exp(0.3)$, $X_2 \sim Bernoulli(0.15)$, $X_3 \sim Bernoulli(0.8)$, and $\log(Y) \sim N(\mu, 0.6^2)$, with $\mu = 3 - 0.5(X_1) + 0.7(X_2) + 0.2(X_3)$. Since the outcome is lognormally distributed, a quasi-likelihood with log link and constant CV should provide valid results on the coefficient estimates as well as their standard errors. Pools were then formed randomly in groups of 2, such that the pooled sample size is $n = 200$.

Tables 4.7a and 4.7b provide results from this simulation on the full data and randomly pooled data, for QL models under both a linear and quadratic (i.e. constant CV) mean-variance relationship. For the pooled data, QL models are first applied "naively" directly to the pooled data, then are redefined under the proper mean and variance functions (4.11).

As expected, both QL models on the full data provide valid estimates of the coefficient parameters, since the link function was correctly specified. Since the outcome is lognormally

Table 4.7a: Mean bias and 95% CI coverage comparing QL models on individual-level and randomly pooled specimens. "SD" refers to empirical standard deviation and "$\hat{\text{SE}}$" represents the mean estimated standard errors. $V(\mu)$ represents the variance function in the QL model.

			Mean Bias ($\hat{\text{SE}}$/SD) 95% CI Coverage		
Method	μ	$V(\mu)$	β_1	β_2	β_3
		Full Data ($N = 400$)			
QL^1	$e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}}$	μ_{ij}^2	0.000 (0.98) 94.7	-0.002 (0.98) 94.1	0.002 (0.98) 94.4
QL^2	$e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}}$	μ_{ij}	0.000 (1.00) 95.3	-0.002 (0.63) 79.1	0.003 (0.79) 87.6
		Randomly Pooled Data ($n = 200$)			
QL$_{naive}$	$e^{\alpha+\mathbf{x}_{i}\boldsymbol{\beta}}$	μ_{i}^2	0.265 (0.57) 0.0	0.083 (0.94) 91.7	-0.003 (0.97) 94.1
QL$_{naive}$	$e^{\alpha+\mathbf{x}_{i}\boldsymbol{\beta}}$	μ_{i}	0.227 (0.81) 0.0	0.072 (0.91) 90.1	-0.005 (1.04) 95.7
QL^1	$\frac{1}{k_i}\sum_{j=1}^{k_i} e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}}$	$\frac{1}{k_i}\sum_{j=1}^{k_i} \mu_{ij}^2$	-0.001 (0.97) 94.3	-0.003 (0.98) 94.1	0.002 (0.99) 94.5
QL^2	$\frac{1}{k_i}\sum_{j=1}^{k_i} e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}}$	$\frac{1}{k_i}\sum_{j=1}^{k_i} \mu_{ij}$	-0.001 (1.25) 98.3	-0.004 (0.74) 85.2	0.005 (0.92) 91.6

Table 4.7b: Empirical standard deviation for various QL models on individual-level and randomly pooled specimens. $V(\mu)$ represents the variance function in the QL model. SD's under models that produce invalid (i.e. biased) results are crossed out, since precision under these models are irrelevant.

			Empirical standard deviation		
Method	μ	$V(\mu)$	β_1	β_2	β_3
		Full Data ($N = 400$)			
QL^1	$e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}}$	μ_{ij}^2	0.010	0.094	0.083
QL^2	$e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}}$	μ_{ij}	0.026	0.118	0.110
		Randomly Pooled Data ($n = 200$)			
QL$_{naive}$	$e^{\alpha+\mathbf{x}_{i}\boldsymbol{\beta}}$	μ_{i}^2	~~0.043~~	~~0.242~~	~~0.209~~
QL$_{naive}$	$e^{\alpha+\mathbf{x}_{i}\boldsymbol{\beta}}$	μ_{i}	~~0.037~~	~~0.192~~	~~0.174~~
QL^1	$k_i^{-1}\sum_{j=1}^{k_i} e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}}$	$k_i^{-1}\sum_{j=1}^{k_i} \mu_{ij}^2$	0.023	0.121	0.120
QL^2	$k_i^{-1}\sum_{j=1}^{k_i} e^{\alpha+\mathbf{x}_{ij}\boldsymbol{\beta}}$	$k_i^{-1}\sum_{j=1}^{k_i} \mu_{ij}$	0.037	0.136	0.140

distributed, the assumption of constant CV (i.e. $V(\mu) = \mu^2$) is appropriate, thus the QL^1 model on the full data also provides valid SE estimates and nominal 95% CI coverage, whereas QL^2 underestimates the true SE's, due to misspecification of the variance structure under this model. Likewise, if data were generated such that the variance structure were a linear function of the mean, the QL^2 model would be expected to outperform the QL^1 model.

For pooled data, both naive models provide biased coefficient estimates and display poor CI coverage, emphasized by a 0% CI coverage for $\hat{\boldsymbol{\beta}}_1$. The QL^1 and QL^2 models applied to these heterogeneous pools perform similarly to their full data counterparts. The results from the QL^1 model are particularly impressive, as this model provides both valid standard error estimates as well as precise coefficient estimates. Its performance is especially motivating in light of its relatively straightforward implementation. While performing MCEM under a lognormal assumption would provide more precise estimates, this QL^1 model can provide a more accessible 'first look' at data containing heterogeneous pools, and a convenient alternative to ML estimation when distributional specification is dubious.

4.6 Data Analysis

Among the 651 individual-level measurements for the outcome, cytokine IP, the four values that fell below the detection limit were set to 0.0005, half the lowest observed limit of detection. In addition to the dataset containing IP values on individual specimens, 508 of those specimens were then physically combined and measured in pools of 2, formed homogeneously with respect to participant SA status (Whitcomb et al., 2012). This subsequent pooling, which was conducted as part of a study design including a methods component, formed a secondary, hybrid dataset consisting of the 208 pooled values as well as the initial measurements on 164 of the individual specimens that were not pooled. In addition to these observed pooled values, we also artificially recreate these pools based on the individual measurements, in order to obtain the regression estimates on the expected values of the pools, calculated as the mean of the IP values from the specimens comprising each pool. Each of the resulting datasets, (full, observed pools, and expected pools) were then fit to each of

the available models. For both pooled datasets, since pools are not entirely homogeneous, the MCEM method was used to fit the lognormal and ML-based gamma[1] models. Since all pool sizes were less than or equal to 2, the Convolution approach of maximizing the observed log-likelihood is also available. This method was applied to the pooled specimens, along with its corresponding AIC value, in order to compare the Monte Carlo based approximations to estimates from direct maximization of the observed log-likelihood. AIC values, where applicable, were generated based on the log-likelihood of the pooled data. For models fit under the MCEM method, the Monte Carlo approximation of the pooled log-likelihood was estimated with an MC size of 10,000. Since the QL models are semi-parametric, AIC values are not available for these methods. Instead, simulations results must be relied upon to assess the performance of these models.

As illustrated in Table 4.8, different conclusions can result from different assumed models. Although not identical, estimates for regression coefficients and standard errors between the MCEM and Convolution methods are extremely similar. In particular, MC AIC values are nearly identical to the AICs calculated from direct optimization of the likelihood via the Convolution Method, thus validating the MC-based AIC as an appropriate technique for approximating this criterion. All AIC values in this analysis tend to favor the lognormal distribution regardless of pooling type, although additional information is recommended to help select the final model. If preventing a Type I Error is a priority, then perhaps one of the QL models may be the most favorable, with application of a variance sandwich estimator for additional assurance (White, 1982). These models are also perhaps the easiest to fit under heterogeneous pools, since the MCEM method is not required. If power is more important, then the potentially more precise lognormal model may be the top contender, due to its favorable AIC values. As suggested by the simulations, the gamma[2] model would perform well assuming the data are actually distributed according to this model, but tends to perform poorly otherwise.

Several issues arose concerning analysis of this pooled dataset. One complication evident in this data analysis is the difference in parameter estimates between the observed and expected pools, suggesting that further information concerning the impact of pooling and measurement error may influence not only the decision on whether to pool data, but also

Table 4.8: Regression estimates and estimated standard errors on data from CPP substudy, with cytokine IP as the outcome. A '*' indicates significant association at the 0.05 level. Gamma[1] refers to the ML gamma regression model with constant shape parameter, gamma[2] the ML gamma regression model with constant scale parameter, and QL to quasi-likelihood methods, where a '1' superscript assumes a constant CV on the individual data and the '2' superscript a linear mean-variance structure.

Fit	AIC	Age (yrs)	Smoking Status (yes/no)	Race (black/white)	SA Status (yes/no)
Full Data ($n = 671$)					
Lognormal	-3894	0.009 (0.006)	0.017 (0.074)	0.185 (0.081)*	-0.017 (0.075)
Gamma[1]	-3808	0.012 (0.005)*	0.022 (0.069)	0.169 (0.076)*	-0.091 (0.070)
Gamma[2]	-3804	0.006 (0.004)	0.008 (0.057)	0.118 (0.062)	-0.008 (0.058)
QL[1]	NA	0.012 (0.008)	0.022 (0.095)	0.169 (0.104)	-0.091 (0.096)
QL[2]	NA	0.011 (0.007)	0.017 (0.095)	0.159 (0.101)	-0.095 (0.096)
Observed Pools ($n = 404$)					
Lognormal					
MCEM	-2443	0.018 (0.009)*	-0.004 (0.116)	0.315 (0.126)*	-0.028 (0.093)
Conv.	-2443	0.018 (0.007)*	-0.001 (0.105)	0.312 (0.124)*	-0.029 (0.094)
Gamma[1]					
MCEM	-2390	0.025 (0.009)*	-0.022 (0.130)	0.353 (0.122)*	-0.093 (0.088)
Conv.	-2390	0.025 (0.008)*	-0.019 (0.129)	0.350 (0.119)*	-0.093 (0.088)
Gamma[2]	-2383	0.011 (0.006)	-0.006 (0.089)	0.194 (0.095)*	-0.018 (0.073)
QL[1]	NA	0.022 (0.010)*	0.010 (0.143)	0.320 (0.151)*	-0.094 (0.110)
QL[2]	NA	0.019 (0.008)*	0.005 (0.119)	0.300 (0.125)*	-0.100 (0.095)
Expected Pools ($n = 404$)					
Lognormal					
MCEM	-2435	0.014 (0.008)	0.012 (0.108)	0.333 (0.116)*	-0.004 (0.086)
Conv.	-2436	0.014 (0.008)	0.013 (0.108)	0.329 (0.116)*	-0.002 (0.086)
Gamma[1]					
MCEM	-2364	0.014 (0.009)	0.067 (0.163)	0.365 (0.119)*	-0.112 (0.084)
Conv.	-2366	0.014 (0.009)	0.072 (0.157)	0.364 (0.116)*	-0.111 (0.084)
Gamma[2]	-2361	0.009 (0.006)	0.008 (0.086)	0.214 (0.091)*	-0.005 (0.071)
QL[1]	NA	0.013 (0.010)	0.084 (0.147)	0.337 (0.151)*	-0.111 (0.108)
QL[2]	NA	0.012 (0.009)	0.071 (0.119)	0.309 (0.123)*	-0.113 (0.093)

the treatment of that pooled data. While these pools were formed homogeneously only with respect to SA, more precision may have been preserved had they been formed homogeneously with respect to all covariates included in the model. In addition, the simpler estimation procedures would be available under this pooling strategy, making model comparison much more accessible.

4.7 Discussion

In summary, the best model for a particular dataset depends on the underlying distribution of the outcome and the type of pooling. Misspecification will almost certainly lead to reduced precision, and can even cause bias in estimates of both the regression coefficients as well as their corresponding standard errors, leading to flawed inference. Semi-parametric models provide an alternative to full specification on the outcome distribution, helping to guard against these potential errors by requiring specification on only the first two moments. The benefit of these quasi-likelihood based models is particularly compelling when analyzing heterogeneous pools, as implementation of these analytical procedures is straightforward, especially when compared with the MCEM algorithm.

On the other hand, when maximizing precision is of utmost importance, these maximum-likelihood based models may be preferable. AIC can help improve precision by identifying the best model, but applies only to likelihood-based methods. In any case, AIC should mainly be used as a guidance tool, in conjunction with other information, such as prior experience and prioritizing between fewer assumptions vs. less precision, in order to choose the best model for each specific study.

As illustrated in Table 4.7a, incorrect specification of the mean or variance structure in a quasi-likelihood framework could result in flawed inference. A quasi-likelihood information criterion may be available to help choose the best mean and variance structures under these semi-parametric models (Pan, 2001), and additional research is recommended to verify the efficacy of these selection methods applied to pooled data. An additional guard against misspecification of the variance structure is the application of sandwich estimators. While we did not explicitly apply these types of estimates in our study, previous research has

demonstrated their ability to provide robust standard error estimates even under incorrect specification of the mean-variance relationship (White, 1982).

The goal of this chapter was to provide and test various analytical strategies available to model right-skewed outcomes in a regression setting that may be subject to pooling. The methods presented here provide a base of available models to analyze datasets similar to the CPP data. Exhausting all possible models for right-skewed data subject to pooling is well beyond the scope of this project. Techniques similar to those discussed here, however, may be applied to additional likelihood-based or semi-parametric methods, depending on the qualities of each particular dataset.

The simulation studies illustrated characteristics of each of the considered models under misspecification, in order to help select the most desirable model. As evidenced in the data analysis, however, real datasets are rarely immune to complications such as a limit of detection or pooling and measurement error. While recent studies have researched these topics with respect to pooling (Schisterman and Vexler, 2008; Schisterman et al., 2010), additional investigation extending these ideas specifically to right-skewed, pooled outcomes, may prove helpful in order to maximize the accessibility and benefit of pooling biospecimens.

Chapter 5

Summary and Future Research

This study focused on analysis and design considerations when a regression outcome is assayed on pooled samples. Simulation results demonstrated the advantages of strategic pooling designs; specifically, when pools are formed from specimens with similar covariate values, high levels of statistical efficiency can be maintained. Furthermore, when pools are $\mathbf{x}$-homogeneous, analytical methods can be greatly simplified. For a right-skewed regression outcome, we developed and tested analytical methods appropriate for this type of data based on parametric and semi-parametric models. These methods were applied to several datasets from epidemiological studies. In particular, the CPP substudy provided important insights, as this dataset contained both individual as well as pooled measurements.

As is often the case, the application of these methods to a real dataset raised several additional considerations. In Chapter 2, we examined an SLR scenario when specimens may contribute unequal aliquot sizes to pools. While we anticipate that similar treatment of the aliquots applied to additional pooling scenarios (e.g. generalized linear regression) would result in similar consequences, it may be useful to verify this assumption through additional studies.

One of the more obvious issues evident from the data analyses, perhaps, is the difference between the observed measurement from an actual pool and the expected value based on the average of the measurements from the individual specimens comprising that pool. This discrepancy may be a result of measurement or pooling error, or both, and is potentially important because most analytical methods for handling pooled specimens (including ours)

assume that the lab assay returns the average biomarker concentration across members of the pool. Schisterman et al. (2010) explored this topic with respect to estimating the marginal mean and variance for a particular biomarker, recommending a hybrid pooled-unpooled design as an effective strategy in evaluating pooling and measurement error for a gamma or normal distribution. We anticipate that a similar design could prove helpful in evaluating error components in a regression setting, and in future research, we intend to extend these methods to the regression settings considered here, both when pooling is applied to the outcome and when it is applied to an exposure of interest. The latter scenario raises the more challenging questions regarding processing and measurement error adjustment.

Another complication evident during data analysis is the effect of the limit of detection on pooled data. Schisterman and Vexler (2008) demonstrated the potential advantages of pooling when estimating the mean and variance of biospecimens subject to a limit of detection, where the decision concerning whether or not to pool depends on the detection threshold. Furthermore, the treatment of non-detects can affect analytical results. Additional exploration of these topics is recommended in order to identify an ideal strategy for dealing with this complication when regression is performed on a pooled outcome.

In this study, we limited our focus to pooling on an outcome, where pools are formed based only on fully-known covariate data. Another area of interest, however, is pooling an exposure, where pools are potentially informed by a fully known outcome and other covariates. In general, informative pooling strategies that incorporate the outcome into the pooling method tend to produce biased estimates of the regression coefficients. In a logistic regression setting, Prentice and Pyke (1979) demonstrated that sampling on an outcome does not induce bias into the regression coefficient estimates, which simplifies analyses under a case-control sampling design. Weinberg and Umbach (1999) were able to take advantage of this result to assess a pooled exposure in logistic regression, where pools are formed homogeneously with respect to a binary outcome. In Chapter 1 we briefly summarized their method, which exploits the multiplicative properties of the risk model to maintain the benefit of the Prentice and Pyke result when pooling on exposure.

For linear regression or other generalized linear regression models, a 'safe' way to avoid

this bias is to limit pooling strategies to only incorporating other covariate values, so at least the remaining coefficient estimates will be precise, and, under appropriate analytical models, all estimates will be consistent. Additional efficiency may be gained by including the outcome information into the pooling strategy, however, and we intend to explore the possibility of correcting any potential bias to maintain estimate accuracy and precision. One potential mitigation technique is based on bias-correction methods incorporated into an outcome-dependent sampling design. Weaver and Zhou (2005) recommend an alternate likelihood that utilizes a semi-parametric approximation of the distribution of the predictor variables in order to reduce this potential bias. Just as Weinberg and Umbach (1999) extended the results from Prentice and Pyke (1979) to a pooled exposure, we intend to explore the possibility of extending these outcome-dependent sampling methods to a pooling scenario, in order to optimize efficiency from pooling based on a continuous outcome while maintaining valid estimates of the regression coefficients.

A critical aspect of any regression analysis is assumption validation. Currently, we are not aware of any diagnostic tests developed specifically for pooled data. Examination of any potential assumption violations is complicated by the nature of pooled data, which can mask the underlying distribution. When assumptions apply directly to the pooled measurements, such as in the linear regression scenario in Chapter 2, the Approximate Model in Chapter 3, or some of the models in Chapter 4, we expect extension of the usual diagnostic techniques to be relatively straightforward. For those models that make assumptions on the individual specimens that do not carry directly over to the pools, such as any of the models that apply the MCEM method, diagnostic procedures may not be straightforward. In such cases, we recommend application of a hybrid pooling design, where pools as well as individual specimens are measured. In this way, diagnostics can be performed on the individual data, so that any violations of the assumptions might be identified using this subset. While the k-means algorithm will often single out individual specimens naturally, additional samples may be desired in order to obtain enough individual specimens for reliable evaluation of diagnostic tests. Future research will focus on extending the Monte Carlo methods used to approximate the AIC for these models to calculation of deviance or other measures of fit. While AIC can help choose the better of two models, deviance can aid in determining

whether a particular model might have poor fit.

Similar methods could also be developed for the quasi-likelihood models from Chapter 4. In this study, the main motivation in applying quasi-likelihood models to pooled data was the relative ease of analysis. The robustness of these QL methods, however, could be further exploited by developing a pooled version of the sandwich estimator (White, 1982). A quasi-likelihood information criterion (QIC) for pooled regression might also prove useful in testing between candidate mean and variance structures (Pan, 2001).

A further extension of these quasi-likelihood methods might also apply to longitudinal data, where generalized estimating equations (GEE) can be used to evaluate correlated data in a regression setting. Chen et al. (2009) and Lyles et al. (2012) considered nonlinear mixed models for a binary outcome and exposure, respectively, when within-subject pooling is applied to evaluate these variables in a longitudinal study, and Malinovsky et al. (2012) considered pooling designs for estimating the intraclass correlation coefficient under a Gaussian random effects model when the repeated outcome is pooled. Future work will focus on developing similar methods to handle longitudinal or otherwise correlated outcomes in the skewed regression settings considered in Chapters 3 and 4.

A primary focus of this study was to consider pooling strategies when budgetary constraints limit the number of feasible lab tests to be performed. The strategic pooling designs, such as $\mathbf{x}$-homogeneous pools and k-means pooling, are advantageous under this assumption. When pooling is performed for other reasons, such as to reduce the number of non-detects subject to a limit of detection, additional designs will likely need to be considered for optimal performance. Ultimately, the best pooling design will depend on additional considerations specific to each study. A thorough evaluation of cost savings vs. potential precision reduction, feasibility and practicality of implementation, and inclusion of individual specimens to facilitate error assessment and model diagnostics is recommended. While pooling can be a valuable tool in reducing the cost of lab assays, the advantages and disadvantages of any pooling strategies as well as availability of appropriate analytical methods, whenever possible, should be carefully considered prior to implementation.

Bibliography

Abbas, O.A. (2008). Comparisons Between Data Clustering Algorithms. *The International Arab Journal of Information Technology* **5,** (3) 320–325.

Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control* **19,** (6) 716–723.

Bain, L.J, and Engelhardt, M. (1992). *Introduction to Probability and Mathematical Statistics,* Second Edition. Belmont: Brooks/Cole Cengage Learning.

Barakat, R.B. (1976). Sums of independent lognormally distributed random variables. *Journal of the Optical Society of America* **66,** (3) 211–216.

Beaulieu, N.C., Abu-Dayya, A.A., and McLane, P.J. (1995). Estimating the Distribution of a Sum of Independent Lognormal Random Variables. *IEEE Transactions on Communications* **43,** (12) 2869–2873.

Beaulieu, N.C., and Xie, Q. (2004). An Optimal Lognormal Approximation to Lognormal Sum Distributions. *IEEE Transactions on Vehicular Technology.* **53,** (2) 479–489.

Booth, J.G., and Hobert, J.P. (1999). Maximizing Generalized Linear Mixed Model Likelihoods with an Automated Monte Carlo EM Algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **61,** (1) 265–285.

Brookmeyer, R. (1999). Analysis of Multistage Pooling Studies of Biological Specimens for Estimating Disease Incidence and Prevalence. *Biometrics* **55,** 608–612.

Burnham, K.P., and Anderson, D.R. (2002). Selection When Probability Distributions Differ

by Model. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd edition. (pp. 317–323). New York: Springer.

Caffo, B.S., Jank, W., and Jones, G. L. (2005). Ascent-Based Monte Carlo expectation-maximization. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **67,** (2) 235–251.

Caudill, S.P., Turner, W.E., and Patterson, D.G. Jr. (2007) Geometric mean estimation from pooled samples. *Chemosphere* **69,** 371–380.

Caudill, S.P. (2010) Characterizing populations of individuals using pooled samples. *Journal of Exposure Science and Environmental Epidemiology* **20,** 29–37.

Caudill, S.P. (2011) Important issues related to using pooled samples for environmental chemical biomonitoring. *Statistics in Medicine* **30,** 515–521.

Chen P., Tebbs J.M., and Bilder C.R. (2009). Group testing regression models with fixed and random effects. *Biometrics* **65,** 1270–1278.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **39,** (1) 1–38.

Dick, E.J. (2004). Beyond 'lognormal versus gamma': discrimination among error distributions for generalized linear models. *Fisheries Research* **70**, 351–366.

Dorfman, R. (1943). The detection of defective members of large populations. *Annals of Mathematical Statistics* **14,** 436–440.

Dupuis, P., Leder, K., and Wang, H. (2007). Importance sampling for sums of random variables with regularly varying tails. *ACM Transactions on Modeling and Computer Simulation* **17,** (3) 1–21.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *Annals of Statistics* **7,** (1) 1–26.

Emmanuel, J.C., Bassett, M.T., Smith, H.J., and Jacobs, J.A. (1988). Pooling of sera for human immunodeficiency virus (HIV) testing: an economical method for use in developing countries. *Journal of Clinical Pathology* **41,** 582–585.

Firth, D. (1988). Multiplicative Errors: Log-Normal or Gamma? *Journal of the Royal Statistical Society. Series B (Methodological)* **50,** (2) 266–268.

Frigyik, B.A., Kapila, A., and Gupta, M.R. (2010). *Introduction to the Dirichlet Distribution and Related Processes*. UWEE Technical Report. Department of Electrical Engineering, University of Washington.

Glynn, R. J., and Laird, N. M., (1986). Regression estimates and missing data: complete-case analysis. Technical Report, Harvard School of Public Health, Dept. of Biostatistics.

Hardy, J.B. (2003). The Collaborative Perinatal Project: lessons and legacy. *Annals of Epidemiology* **5,** 303–311.

Hartigan, J.A. (1975). *Clustering Algorithms.* New York: Wiley.

Hartigan, J.A. and Wong, M.A. (1979). A K-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28,** (1) 100–108.

Heyde, C.C. (1997). *Quasi-Likelihood and Its Application: A General Approach to Optimal Parameter Estimation*. New York: Springer.

Huber, P.J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics* **35,** (1) 73–101.

Jain, A.K., Murty, M.N., and Flynn, P.J. (1999). Data Clustering: A Review. *ACM Computing Surveys* **31,** (3) 264–323.

Jamshidian, M., and Jennrich, R.I. (2000). Standard Errors for EM Estimation. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **62,** (2) 257–270.

Lan, S., Hsieh, C., and Yen, Y. (1993) Pooling Strategies for Screening Blood in Areas with Low Prevalence of HIV. *Biometrical Journal* **35,** (5) 553–565.

Lange, K. (2010). *Numerical Analysis for Statisticians.* New York: Springer. 2nd edition.

Levine, R.A., and Casella, G. (2001). Implementations of the Monte Carlo EM Algorithm. *Journal of Computational and Graphical Statistics* **10,** (3) 422–439.

Li, X. (2007). *A Novel Accurate Approximation Method of Lognormal Sum Random Variables.* (Master's Thesis). B.E. Electronic Engineering Department, Tsinghua University.

Li, X., Wu, Z., Chakravarthy, V.D., and Wu, Z.W. (2011). A Low-Complexity Approximation to Lognormal Sum Distributions via Transformed Log Skew Normal Distribution. *IEEE Transactions on Vehicular Technology* **60,** (8) 4040–4045.

Likas, A., Vlassis, N., and Verbeek, J.J. (2003) The global k-means clustering algorithm. *Pattern Recognition* **36,** 451–461.

Little, R.J.A. (1992). Regression with Missing X's: A Review. *Journal of the American Statistical Association* **87,** (420) 1227–1237.

Little, R.J.A. and Rubin, D.B. (2002). *Statistical Analysis with Missing Data, Second Edition.* Wiley: New York.

Liu, Z., Almhana, J., Wang, F., and McGorman, R. (2007). Mixture Lognormal Approximations to Lognormal Sum Distributions. *IEEE Communications Letters* **11,** (9) 711–713.

Louis, T.A. (1982). Finding the Observed Information Matrix when Using the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **44,** (2) 226 - 233.

Lyles, R.H., Tang, L., Lin, J., Zhang, Z., and Mukherjee, B. (2012). Likelihood-based methods for regression analysis with binary exposure status assessed by pooling, *Statistics in Medicine* **31,** 2485–2497.

Ma, C.-X., Vexler A., Schisterman E.F., Tian L. (2011). Cost-efficient designs based on linearly associated biomarkers. *Journal of Applied Statistics* **38,** (12) 2739–2750.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* **1,** 281–297.

Malinovsky, Y., Albert, P.S., and Schisterman, E.F. (2012). Pooling Designs for Outcomes under a Gaussian Random Effects Model. *Biometrics* **68,** 45 –52.

McCullagh, P. (1983). Quasi-Likelihood Functions. *The Annals of Statistics* **11,** (1) 59–67.

Oakes, D. (1999). Direct Calculation of the Information Matrix via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **61,** (2) 479–482.

Pan, W. (2001). Akaike's Information Criterion in Generalized Estimating Equations. *Biometrics* **57,** (1) 120–125.

Perkins, N.J., Whitcomb, B.W., Lyles, R., and Schisterman, E.F. (2011). Analysis of randomly pooled case-control data. Poster session presented at: Joint Statistical Meetings. July 30 - August 4, Miami, FL.

Prentice, R.L., and Pyke, R. (1979). Logistic disease incidence models and case-control studies. *Biometrika* **66,** (3) 403–411.

R Development Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Santos Filho, J.C.S., Yacoub, M.D., and Cardieri, P. (2006). Highly accurate range-adaptive lognormal approximation to lognormal sum distributions. *Electronics Letters* **42,** (6).

SAS Institute Inc. (2010). SAS/STAT 9.2 Users Guide, Second Edition. Cary, NC: SAS Institute Inc.

Schisterman, E.F., and Vexler, A. (2008). To pool or not to pool, from whether to when: applications of pooling to biospecimens subject to a limit of detection. *Pediatric and Perinatal Epidemiology* **22,** 486–496.

Schisterman, E.F., Vexler, A., Mumford, S.L., and Perkins, N.J. (2010). Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers. *Statistics in Medicine* **29,** 597–613.

Schliep, K.C., Schisterman, E.F., Mumford, S.L., Pollack, A.Z., Zhang, C., Ye, A., Stanford, J.B., Hammoud, A.O., Porucznik, C.A., and Wactawski-Wende, J. (2012). Caffeinated beverage intake and reproductive hormones among premenopausal women in the BioCycle Study. *American Journal of Clinical Nutrition* **95,** (2) 488–497.

Szyszkowicz, S.S., and Yanikomeroglu, H. (2009). Fitting the Modified-Power-Lognormal to the Sum of Independent Lognormals Distribution. *IEEE Global Telecommunications Conference Proceedings.* Nov. 30 - Dec. 4, Honolulu, HI.

Tan, M., Tian, G.-L, and Fang, H.-B. (2007). An efficient MCEM algorithm for fitting generalized linear mixed models for correlated binary data. *Journal of Statistical Computation and Simulation* **77,** (11) 929–943.

Tellambura, C. and Senaratne, D. (2010). Accurate Computation of the MGF of the Lognormal Distribution and its Application to Sum of Lognormals. *IEEE Transactions on Communications* **58,** (5) 1568–1577.

Vansteelandt, S., Goetghebeur, E., and Verstraeten, T. (2000). Regression models for disease prevalence with diagnostic tests on pools of serum samples. *Biometrics* **56,** (4) 1126–1133.

Vexler, A., Liu, A., and Schisterman, E.F. (2006). Efficient Design and Analysis of Biospecimens with Measurements Subject to Detection Limit. *Biometrical Journal* **48,** (5) 780–791.

Vexler, A., Liu, A., and Schisterman, E.F. (2010). Nonparametric deconvolution of density estimation based on observed sums. *Journal of Nonparametric Statistics* **22,** (1) 23–29.

Weaver, M.A., and Zhou, H. (2005). An Estimated Likelihood Method for Continuous Outcome Regression Models with Outcome-Dependent Sampling. *Journal of the American Statistical Association* **100,** (470) 459–469.

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

Wei, G.C.G., and Tanner, M.A. (1990). A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association* **85,** (411) 699–704.

Weinberg, C.R., and Umbach, D.M. (1999). Using pooled exposure assessment to improve efficiency in case-control studies. *Biometrics* **55,** (3) 718–726.

Whitcomb, B.W., Schisterman, E.F., Klebanoff, M.A., Baumgarten, M., Rhoton-Vlasak, A., Luo, X., and Chegini, N. (2007) Circulating chemokine levels and miscarriage. *American Journal of Epidemiology* **166,** (3) 323–331.

Whitcomb, B.W., Schisterman, E.F., Klebanoff, M.A, Baumgarten, M., Luo, X., and Chegini, N. (2008) Circulating levels of cytokines during pregnancy: thrombopoietin is elevated in miscarriage. *Fertility and Sterility* **89,** (6) 1795–1802.

Whitcomb, B.W., Perkins, N.J., Zhang, Z., Ye, A., and Lyles, R.H. (2012) Assessment of skewed exposure in case-control studies with pooling. *Statistics in Medicine* **31,** 2461–2472.

White, H. (1982) Maximum Likelihood Estimation of Misspecified Models. *Econometrica* **50,** (1) 1–25.

Zhang, Q.T., and Song, S.H. (2006). Model Selection and Estimation for Lognormal Sums in Pearson's Framework. *IEEE Vehicular Technology Conference.* **6,** 2823–2827.

Zhang Z, and Albert P.S. (2011). Binary regression analysis with pooled exposure measurements: a regression calibration approach. *Biometrics* **67,** 636–645.

Zhang, Z., Liu, A., Lyles, R.H., and Mukherjee, B. (2012). Logistic regression analysis of biomarker data subject to pooling and dichotomization. *Statistics in Medicine* **31,** 2473–2484.

Appendix A

R and SAS Code Examples

A.1 k-means Clustering

R Code

X is standardized using the defined function 'stdize' so that each covariate has similar influence on clustering procedure. Then *kmeans* is applied to form 50 clusters from the original 100 observations.

```
X = data.frame(x1 = rnorm(100),x2 = rnorm(100))
stdize = function(t) (t-mean(t))/sd(t)
stdX = apply(X,2,stdize)
kmeans(stdX,50)
```

SAS Code

X is standardized using the STANDARD procedure, and 50 clusters are formed using the FASTCLUS procedure.

```
proc standard data=one out=stand mean=0 std=1;
  var x1 x2; run;
proc fastclus data=stand out=clust maxclusters=50 converge=0 maxiter=100;
  var x1 x2; run;
```

A.2 gamma^2 Model

R Code

R code to calculate MLEs from gamma^2. Code as shown applies to individual-level data as well as both **x**-homogeneous and heterogeneous pools. The defined AVE function just alters the built-in R *ave* function so that it can be used in conjunction with *apply* without duplicating the function argument. The *Poolit* function was written to simplify the pooling process, by averaging (or summing) the data by cluster, then ordering by cluster number and dropping duplicates (i.e. keeping only one observation per pool). When optimizing the *gamma*2 function, the 'y' vector should be a vector of unique pooled measurements, in the order of cluster number. In the *optim* statement, 'par' is a vector of starting values for $\theta = (\boldsymbol{\beta}, \phi)$. In this code, the intercept (α) is included in the coefficient vector ($\boldsymbol{\beta}$).

```
AVE = function(x,...,fun = mean) ave(x,...,FUN=fun)
Poolit = function(data,cluster,sums=F){
     if(sums) fun = sum else fun = mean
     unique(apply(cbind(cluster,data),2,
          FUN=AVE,cluster,fun=fun)[order(cluster),])[,-1]
     }


gamma2 = function(theta,y,X,cluster=1:length(y)){
     b = theta[length(theta)]
     Beta = theta[-length(theta)]
     eta = as.matrix(cbind(1,X))%*%Beta
     mu.ij = exp(eta)
     mu.i  = Poolit(mu.ij,cluster,sum=T)
     a.i   = mu.i/b
     ki    = tabulate(cluster)[tabulate(cluster)>0]
     b.i   = b/ki
```

```
     ll = sum(dgamma(y,shape=a.i,scale=b.i,log=T))
     return(-ll)
     }


optim(par,fn=gamma2,y=y,X=X,cluster=cluster,
     hessian=T,method="L-BFGS-B",lower=c(rep(-Inf,length(par)-1),1E-7))
```

SAS Code

SAS code to calculate MLEs from gamma^2 model using NLMIXED procedure. Shown here for full data, but can be altered to accommodate **x**-homogeneous or heterogeneous pools.

```
PROC NLMIXED data=one;
     *starting values;
        parms  alpha = 0   beta1 = 0   beta2 = 0    scale = 1;
     ai   = exp(alpha + beta1*x1 + beta2*x2);
     trm1 = -lgamma(ai) - ai*log(scale);
     trm2 = (ai-1)*log(y) - y/scale;
     LL   = trm1 + trm2;
     model y ~ general(LL);
     ods output parameterestimates = est   convergencestatus = rc;
run;
```

A.3 QL Model under heterogeneous pools.

R Code

R code using *nleqslv* to calculate QL estimates for $\boldsymbol{\beta}$ under heterogeneous pools ('nleqslv' package required). The variance function for individual data (V.ij) can be altered to specify a constant CV or a linear mean-variance relationship. Again, 'y' should be a vector of unique pooled measurements in the order of cluster number, and 'par' is a vector of starting values for θ. Note that this code only calculates regression coefficients; the dispersion parameter

ϕ must be calculated separately.

```
library(nleqslv)


QL.htro = function(Beta,y,X,cluster=1:length(y)){
     X1 = as.matrix(cbind(1,X))
     eta = X1%*%Beta
     mu.ij = exp(eta)


     V.ij  = mu.ij^2   # QL1: constant CV
     #V.ij  = mu.ij   # QL2: linear mean-var


     ki   = tabulate(cluster)[tabulate(cluster)>0]
     mu.i = Poolit(mu.ij,cluster,sum=F)
     V.i  = Poolit(V.ij,cluster,sum=T)/ki^2
     dmu  = Poolit(X1*c(mu.ij),cluster)


     Q = t(dmu)%*%as.matrix((y-mu.i)/V.i)
     return(Q)
}


nleqslv(par,QL.htro,y=y,X=X,cluster=cluster,jacobian=T)
```