SIMULTANEOUS INFERENCE IN GENERALIZED LINEAR MODEL SETTINGS

By

AMY WAGLER

Bachelor of Science in Mathematics
University of Texas of the Permian Basin
Odessa, Texas, USA
1995

Master of Science in Statistics
Oklahoma State University
Stillwater, Oklahoma, USA
2003

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2007

COPYRIGHT ©

By

AMY WAGLER

July, 2007

SIMULTANEOUS INFERENCE IN GENERALIZED LINEAR MODEL SETTINGS

Dissertation Approved:

Dr. Melinda McCann

Dissertation Advisor

Dr. Stephanie Monks

Dr. Mark Payton

Dr. Wade Brorsen

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

There are many I would like to thank for assistance, reassurance and inspiration during the time I worked on this dissertation. First, I would like to thank the entire Department of Statistics at Oklahoma State University including all of the faculty members, staff, and students. I appreciate your kindness and concern for me. Of the faculty members, I would specifically like to thank my committee members, Dr. Stephanie Monks, Dr. Mark Payton, and Dr. Wade Brorsen, for the time and effort they put into this research project. Most importantly, I would like to thank my advisor, Dr. Melinda McCann, for her support and encouragment, perseverance in research, and friendship. I was fortunate to have an advisor such as you. I would also like to thank Trecia Kippola, Tracy Morris, and Janae Nicholson who listened to my many complaints and were just good friends.

The most emphatic thanks I reserve for my family, who are always loving and supportive. My Mom, Dad, and sisters for believing I could do this even when I didn't believe. Most of all, I would like to thank Ron, my husband, and Olive, my daughter. When I needed help, encouragement, or a baby-sitter, my husband was always there. When I was frustrated and feeling down, a smile from Olive was all I needed to put things in perspective. Thank you and I love you both.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Modern epidemiological and medical research routinely employs generalized linear modeling. These models can be helpful in understanding what behaviors or traits can influence the incidence of a particular disease or characteristic. For example, logistic regression models provide a means of relating the incidence of some trait or disease to a set of possible predictor variables, while loglinear models help us understand associations between a trait and predictor variables.

After building a generalized linear model(GLM), one typically wishes to estimate particular quantities of interest such as response probabilities, odds ratios, or relative risks. Customarily, these are reported via confidence intervals or confidence bounds using some pre-specified level of significance for each inference. For example, using a loglinear model, one could report $100(1-\alpha)\%$ confidence intervals for each relative risk resulting from the model. Using one-at-a-time intervals is appropriate when the investigators are not making overall conclusions about the quantities of interest. For example, if the aforementioned loglinear model was estimated and $100(1-\alpha)\%$ confidence intervals for the relative risks were reported, conclusions about each individual relative risk could be made, but any statements comparing these relative risks would inflate the assumed α error rate. Research on simultaneous estimation procedures for quantities from generalized linear models has received little attention beyond very routine treatments, such as making Bonferroni adjustments to the usual confidence interval or constructing Scheffé intervals. However, recent advances made in simultaneous inference for linear models may be applied in the generalized linear model setting. Additionally, further improvements may be made by utilizing some unique properties of the estimated parameters from generalized linear models. I plan to present the justification for employing these simultaneous inference methods in the generalized linear model setting and, via simulation, compare their performance. Thus, the objective of this study is to develop simultaneous interval based procedures that will estimate functions of linear combinations of the parameters of a generalized linear model. Specifically, this includes simultaneously estimating the expected response function, odds ratios, and relative risks from generalized linear models.

Overall, attention is focused on quantities that are estimated from a GLM, not the estimation of the GLM itself. Obviously, the performance of any of these procedures will be influenced by how well the model is estimated, but this dissertation will assume the model is well estimated. Additionally, all of the procedures developed in this paper involve constructing interval estimates. Often, procedures that account for multiplicity employ hypothesis tests to make overall conclusions about a set of data. It is more appropriate in the applications I will discuss to use simultaneous intervals instead of stepwise procedures, since I wish to not only detect statistical differences between quantities, but also to assess the practical significance of these differences. Thus, all methods discussed are interval-based procedures.

Before presenting the details of generalized linear models, a practical example of the implementation of a GLM may provide a frame of reference. A 2003 study from the *American Journal of Epidemiology* explored the relationship between maternal stress and preterm birth [1]. Several previously identified sources of maternal stress, such as high incidence of life events, increased anxiety, living in a dangerous neighborhood, and increased perception of stress, were explored for any association with preterm births. Specifically, the study focused on predicting the prevalence of preterm birth among pregnant women aged 16 or older from two prenatal clinics in central North Carolina. Upon admission to the study, women were asked to complete

Table 1.1: Maternal Stress Relative Risks and 95% Confidence Intervals

Life Events Stress	RR	95% CI
No Stress	1.00	
Med-Low Stress	1.5	(1.0, 2.2)
Med-High Stress	1.4	(0.9, 2.1)
High Stress	1.8	(1.2, 2.7)

questionnaires in addition to completing a psychological instrument. Also, several blood, urine, and genital tract tests were conducted in order to assess the physical health of the candidates. In all, 2,029 women were eligible, recruited, and completed the preliminary tests in order to participate in the study. Of these participants, 231 delivered preterm, less than 37 weeks gestation. A loglinear model was employed to assess the relationship between the sources and levels of maternal stress and preterm birth. As a result, the authors considered the resulting model relative risks for each individual stress factor or level of a stress factor and its association with preterm birth. Additionally, 95% confidence intervals were computed for each relative risk. The relative risk will be discussed in detail later, but note that for this study each relative risk is the risk of preterm birth for an individual with one particular maternal stress factor relative to the risk of preterm birth for an individual with none of the other identified sources of maternal stress present. Thus a large relative risk for a particular source of maternal stress indicates a strong association between that stress factor and preterm birth. In general, other quantities derived from the generalized linear model may also be of interest. Table 1.1 contains results from the preterm birth study discussed, though these results have been simplified from the actual implemented model for ease of presentation. In particular, the relative risks for different levels of stress due to general life events is presented. As previously discussed, note that the 95% confidence intervals for each relative risk presented in Table 1.1 estimate the risk of preterm birth for each individual subject to a particular level of a maternal stress factor (life event stress) with reference to the control (the case with no identified source of maternal stress). For this scenario, one-at-a-time inferences are reasonable if the researcher wishes to answer questions such as, "how does the presence of one source of maternal stress affect the risk of preterm birth?" Note that this question is only concerned with the presence of a particular stress factor and how it affects the incidence of preterm births. If one wishes to make any overall conclusions comparing how the multiple sources of maternal stress affect the risk of preterm birth, then another estimation procedure that accounts for multiplicity needs to be implemented. For instance, in the preterm birth study, the researchers reported the above relative risks and confidence intervals, and then remarked that "(w)omen in the highest negative life events impact quartile had the highest risk (RR=1.8, 95% CI: 1.2, 2.7); however, the middle categories did not show increasing risk with increasing measures of stress." This kind of conclusion is inappropriate given that the researchers only computed one-at-a-time 95% confidence intervals for the relative risks. Therefore, this is a case that would benefit from simultaneous inference on the relative risks.

As another example where simultaneous inference would be appropriate, consider the case where the researcher wishes to identify the set of the sources and levels of maternal stress that are significantly associated with preterm birth. In this case the one-at-a-time intervals are again inappropriate. In order to make this kind of conclusion, the researcher needs to determine which groups of relative risks are significantly different from 1. If the researcher additionally wants to determine the practical significance of the differences between the varying sources and levels of maternal stress and the control, then confidence intervals with a multiplicity adjustment are required. Stepwise procedures are not adequate as they only determine where statistically significant differences exist, but do not provide a way to estimate the scale of these differences. Additionally, it is often desirable to make conclusions such as “if the subject has one level of a predictor variable, then he is twice as likely to have the disease than if he has any other level of that predictor variable”. Many other examples of similar conclusions could be given, but generally, these conclusions are comparing one parameter to another and the desired outcome is to somehow relate these parameters. Thus, if one wishes to make any comparisons of these parameters, it is desirable to control the overall type I error rate by accounting for the multiplicity of inferences.

In the following chapters, I will present the motivation for simultaneous inference of certain parameters and outline both the current methodologies and my proposed methodologies. Additionally, the simulation results of the proposed methods are presented and analyzed. Specifically, in chapter two, I present the generalized linear model and the typical quantities that are estimated from the model, and discuss why simultaneous inference of these quantities is essential in some situations. Additionally, I review some methods for computing one-at-a-time confidence intervals on various quantities resulting from generalized linear models. In chapter three, I outline the current methodologies used for simultaneously estimating various functions resulting from generalized linear models, and I propose four new methods to estimate these parameters from GLMs. In chapter four, I summarize how I evaluated these new methods using simulation and present the simulation results. Finally, I propose some future research questions regarding simultaneous estimation of a GLM and present some applications of the new methods in the concluding chapter.

CHAPTER 2

The GLM and Estimated Quantities

There are several generalized linear models (GLM) that permit estimates, such as the odds ratio or relative risk, where multiplicity adjustments often seem warranted. Some of these models include the logistic regression model, loglinear model, Poisson regression model, and the probit or complementary log-log model. In general, a GLM can be expressed as

$$Y_i = g^{-1}(\boldsymbol{x}_i'\boldsymbol{\beta} + \boldsymbol{\epsilon}_i), \quad i = 1, \ldots, n \tag{2.1}$$

or alternatively,

$$\phi_i = g(E(Y_i|\boldsymbol{x}_i)) = \boldsymbol{x}_i'\boldsymbol{\beta}, \quad i = 1, \ldots, n \tag{2.2}$$

where g links the expected response, $E(Y_i|\boldsymbol{x}_i)$, to ϕ_i, with $\boldsymbol{x}_i$ the vector of covariates corresponding to Y_i, $\boldsymbol{\beta}$ the $k \times 1$ vector of regression parameters, and ϵ_i independently and identically distributed random variables. In the later sections, $\boldsymbol{Y} = (Y_1, \ldots, Y_n)'$ is the vector of responses and $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$ is the full rank matrix of predictor variables. Each GLM corresponds to a particular link function g, typically called the canonical link when it transforms the mean to the natural parameter. In general, the maximum likelihood estimate (MLE) of the regression parameters is denoted $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_k)$. The MLE is asymptotically multivariate normal with mean $\boldsymbol{\beta}$ and covariance matrix

$$\boldsymbol{V} = (\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1} \tag{2.3}$$

where $\boldsymbol{W}$ is a diagonal matrix with diagonal elements $w_i = (\partial\mu_i/\partial\phi_i)^2/var(Y_i)$ for $\mu_i = E(Y_i|\boldsymbol{x}_i)$ and ϕ_i in (2.2). Thus,

$$\hat{\boldsymbol{\beta}} \dot{\sim} N_k(\boldsymbol{\beta}, \boldsymbol{V}). \tag{2.4}$$

We can estimate the covariance matrix, $\boldsymbol{V}$, by

$$\hat{\boldsymbol{V}} = c\hat{o}v(\hat{\boldsymbol{\beta}}) = (\boldsymbol{X'}\hat{\boldsymbol{W}}\boldsymbol{X})^{-1} \tag{2.5}$$

with $\hat{\boldsymbol{W}} = \boldsymbol{W}|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}$. Further results will require the estimated covariance of $\boldsymbol{x}_i'\boldsymbol{\beta}$ for a given $\boldsymbol{x}_i$ vector with k known elements. This is given by

$$\hat{\sigma}^2_{GLM}(\boldsymbol{x}_i) = \sqrt{\boldsymbol{x}_i'\hat{\boldsymbol{V}}\boldsymbol{x}_i}, \quad i = 1, \ldots, n. \tag{2.6}$$

At times I will need to refer to a linear model in this proposal. A linear model is generally given by

$$Y_i = \boldsymbol{x}_i'\theta + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, n \tag{2.7}$$

or alternatively,

$$E(\boldsymbol{Y}) = \boldsymbol{X\theta} \tag{2.8}$$

where $\boldsymbol{\theta}$ is the vector of regression parameters, $\boldsymbol{Y}$ and $\boldsymbol{X}$ are as previously defined, and $\epsilon_i \sim \ iid\ N(0, \sigma^2_{LM})$. The MLEs for $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$ are denoted $\hat{\boldsymbol{\theta}}$ and

$$\hat{\boldsymbol{\theta}} \sim N_k(\boldsymbol{\theta}, \sigma^2_{LM}\boldsymbol{F}) \tag{2.9}$$

where σ^2_{LM} is the variance of the model residuals for a linear model and $\boldsymbol{F} = (\boldsymbol{X'X})^{-1}$. Whenever a linear model is referenced in this paper, the notation presented above will be utilized.

As discussed previously, the objective of this research is not to merely estimate a GLM, but to simultaneously estimate quantities derived from a GLM. There are many natural quantities that can be of interest when modeling data with a GLM. These include measures such as the expected mean response, the odds ratio, the relative risk, and possibly others. While the focus of this paper is on the estimation of these quantities from GLMs, it should be mentioned that they may be estimated directly from the data. I will first present these measures in general, and then discuss them specifically in the context of generalized linear models.

2.1 The Expected Mean Response

The expected mean response is a generic term for either the response probability or the mean response. Depending on the sampling distribution of the data, either one or the other is of interest. For example, if we assume binomial sampling, the expected mean response function is the probability of a success for a given level of the predictor variables, or the response probability. When a Poisson sampling scheme is assumed, the expected mean response is the average for a particular cell in the contingency table, or the mean response.

A response probability is the proper quantity of interest if one wishes to understand the probability of developing a disease or another characteristic for a given set of predictor variables that are believed to be associated with the disease. For example, a response probability could be used to inform a particular patient of their probability of developing a particular disease given their history and profile. With respect to the preterm birth example, if a doctor has a patient known to be experiencing a major life event, such as a death in the family, then she could ascertain that patient's risk of preterm birth and take appropriate measures.

Conversely, the mean response might be used in a situation where a clinician has recorded a host of risk factors for a particular disease and wishes to predict how many of the subjects will develop the disease. This communicates how many patients on average will or will not develop a certain characteristic. In the context of the preterm birth example, the Poisson mean response could be used to estimate how many subjects out of the total sample size will experience a preterm birth. The

Poisson mean response is simply estimated directly from the frequencies given in a contingency table when it is not estimated from a model.

In general, a one-at-a-time $100(1-\alpha)\%$ confidence interval for the response probability is given by

$$\hat{\pi}_i \pm z_{\alpha/2}(v\hat{a}r(\hat{\pi}_i))^{1/2} \tag{2.10}$$

where $var(\hat{\pi}_i) = \frac{\pi_i(1-\pi_i)}{m_i}$ and $v\hat{a}r(\hat{\pi}_i) = \frac{\hat{\pi}_i(1-\hat{\pi}_i)}{m_i}$. (Note that confidence intervals on the mean response for Poisson sampling distribution models are not typically computed.) This one-at-a-time confidence interval for π_i is often employed to estimate an expected mean response for binomial or multinomial sampling scenarios. If the researcher simply wants to know how a particular level of the predictor variables affects the incidence of disease, this is all that needs to be calculated. First, consider the case where the predictor variable is categorical, as in the preterm birth example. Suppose a clinician wishes to estimate the risk of preterm birth for a particular patient in her clinic. Then the one-at-a-time interval would be adequate. Alternatively, consider a scenario where a researcher wants to make some kind of overall conclusion about the relationship between all the sources and levels of maternal stress and preterm birth. For example, suppose the researcher wishes to compare the risk of preterm birth for all the maternal stress factors and their levels. In order to simultaneously estimate these differences, the researcher would need to employ some kind of procedure that accounts for the multiple inferences being made. Additionally, it would be of practical interest to identify a group of maternal stress factors and levels that are "most associated" with preterm birth and another group of maternal stress factors and levels that are "least associated" with preterm births. This too would necessitate a procedure that adjusts for multiplicity while also providing interval estimates of the response probabilities for each stress factor. Finally, consider the case where the researcher would want to compare the probability of preterm birth for each maternal stress factor or level with the probability of preterm birth for a control or reference level. In this example, the reasonable reference level would be subjects that have no identified sources of maternal stress. Again the appropriate procedure would adjust for multiplicity.

Now consider the case where the predictor variable is continuous. Often, with a continuous predictor variable, a specific range of the domain is of particular interest. For example, if we added a continuous measure of each patient's prepregnancy body mass index (BMI) in the preterm birth study, we might have particular interest in BMI's less than 19.8 (underweight), 19.8 to 26.0 (normal weight), 26.0 to 29.0 (overweight), and over 29.0 (obese). It may be of interest to compare the expected number of preterm birth cases for subjects within these BMI groups within a particular maternal stress factor group. In order to make conclusions such as the obese patients have the largest number of preterm birth cases, interval estimates need to be used that account for the multiple inferences being made. The methods I propose will adjust for this kind of multiplicity.

2.2 The Odds Ratio

The odds ratio is a widely used measure in epidemiological and medical applications. The odds ratio is generally defined as the ratio of the odds of a characteristic (or disease) occurring in one group to the odds of it occurring in another group. With reference to the preterm birth study, odds ratios could have been computed that would estimate the relative odds of preterm birth for a particular level of a maternal stress factor to the odds for those mothers with no identifiable stress factors. Thus, an odds ratio of 2.11 for mothers who live in a neighborhood perceived to be dangerous, would be interpreted as: the odds of delivering a preterm infant when living in a neighborhood that is perceived to be dangerous is 2.11 times greater than the odds of having a preterm infant when a subject is not exposed to any identifiable sources of maternal stress.

Table 2.1: Sample Contingency Table

	$X = x_1$	$X = x_2$	
$Y = 1$	a	b	m_1
$Y = 0$	c	d	m_2
	n_1	n_2	n

The sample odds ratio can easily be computed from the raw data and is given by $\hat{\eta} = \frac{ad}{bc}$ for counts as given in Table 2.1 irrespective of which sampling model (binomial, multinomial, or Poisson) is assumed for the cell counts. For large samples, again under all sampling models, the log odds ratio, $log(\hat{\eta})$ is asymptotically normal with mean $log(\eta)$ and estimated standard error $\hat{\sigma}_{log\hat{\eta}} = (\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d})^{1/2}$. Thus, a $100(1 - \alpha)\%$ one-at-a-time large sample confidence interval for the log odds ratio is given by

$$log(\hat{\eta}) \pm z_{\alpha/2}\hat{\sigma}_{log(\hat{\eta})}. \tag{2.11}$$

Exponentiating the lower and upper bounds of this interval yields confidence bounds for the odds ratio.

It is common to see one-at-a-time confidence intervals for odds ratios reported along with their point estimates. Suppose a researcher wants to report the estimated odds ratio for a particular patient profile with confidence limits. For example, in the preterm birth example, she may want to report the odds ratio of preterm birth for those exposed to a particular maternal stress factor compared to those with no identifiable maternal stress factors. In this case, the one-at-a-time intervals are appropriate. Alternatively, consider the case where the researcher wants to identify which, if any, of the maternal stress factors or levels of a stress factor are statistically associated with a preterm birth or to identify a set of stress factors or level of a stress factor whose association with preterm birth is larger than that for no stress factors present. The one-at-a-time intervals will not suffice for these kinds of questions as there are multiple inferences being made. In order to control the type I error rate, a simultaneous estimation procedure should be utilized. Additionally, the researcher may wish to compare the odds of preterm birth for any maternal stress factor to the odds of preterm birth for all other sources of maternal stress. In order to do this, a multiple comparison procedure must also be utilized as the researcher actually wants to compare the probabilities of preterm birth across all the possible sources of maternal stress. It is tempting to make overall conclusions about the odds ratios when reporting estimated odds ratios via one-at-a-time confidence intervals. However, the error rate associated with these overall conclusions based on multiple one-at-a-time intervals is not controlled, or even known. In this case, a method that simultaneously estimates the parameters is warranted.

2.3 Other Quantities

Another quantity frequently reported is the relative risk. The relative risk communicates the risk of developing a disease at one level of the predictor variable relative to another level of the predictor variable. An example of a study employing relative risks is the preterm birth study. In Table 2, the estimated relative risk would be given by $\hat{\gamma} = \frac{a/n_1}{b/n_2}$ where the counts are as in Table 2.1. Suppose a researcher reports that the estimated relative risk of preterm birth is 1.75, given the subject lives in a neighborhood perceived to be dangerous. Thus the proportion of those that experience a preterm birth among those that live in the dangerous neighborhood is estimated to be 1.75 times the proportion of those who experience a preterm birth among those with no identified sources of stress. Again, the log scale is often utilized and large sample derivations show that the log of the sample relative risk, $log(\hat{\gamma})$ is asymptotically normal with mean $log(\gamma)$ and estimated standard error $\hat{\sigma}_{log(\gamma)} = (\frac{1-\hat{\pi}_1}{\hat{\pi}_1 n_1} + \frac{1-\hat{\pi}_2}{\hat{\pi}_2 n_2})^{1/2}$

where $\hat{\pi}_i$ for $i = 1, 2$ is the estimated probability of disease among those in group i and n_i is the sample size for group $i = 1, 2$. Thus, the $100(1-\alpha)\%$ confidence interval for the log relative risk is

$$log(\hat{\gamma}) \pm z_{\alpha/2}\hat{\sigma}_{log(\hat{\gamma})}. \tag{2.12}$$

These resulting bounds may be exponentiated in order to obtain confidence limits on the relative risk.

It is sufficient to report an estimated relative risk via a one-at-a-time confidence interval when a researcher only needs to understand how one level of the predictor variable affects the incidence of disease. The preterm birth study reported relative risks and the associated confidence intervals, thus only individual inferences about each source of maternal stress or level of a maternal stress factor relative to the case with no source of stress can be made. However, suppose a researcher wishes to pick out which risk factor or level of a risk factor contributes most to a disease or condition, or obtain ranking information for the sources and levels of maternal stress with respect to risk of disease or condition. As estimation is still also of interest, multiplicity adjustments need to be made to the confidence intervals.

In addition to the relative risk, other quantities should be considered as well. For example, many epidemiological researchers find the attributable proportion a useful measure. Suppose we have a disease and several risk factors for that disease. Then the attributable proportion would be the probability that a diseased individual in the given risk factor has the disease because of that risk factor [2]. This is of interest when there are multiple risk factors for a disease. Thus, this measure is of particular interest in case-control studies where the incidence of disease is related to several risk factors as it allows the researcher to understand how much the disease could be reduced by eliminating a particular risk factor.

One-at-a-time confidence intervals can also be utilized to estimate the attributable proportions. Model-based confidence interval formulas can be computed on the usual attributable proportion. Often transformations of the relative risk are utilized to compute bounds on the attributable proportion since they can be more efficient asymptotically. However, the MLE-based interval performs adequately [3] and is more easily adjusted for simultaneous inference in the sequel. Thus, a one-at-a-time confidence interval for the attributable proportion, denoted κ, is given by,

$$\hat{\kappa} \pm z_{\alpha/2} \times v\hat{a}r(\hat{\kappa})^{1/2} \tag{2.13}$$

where $z_{\alpha/2}$ is the z critical value that gives $100(1-\alpha)\%$ confidence. (Details for computing $v\hat{a}r(\hat{\kappa})$ are given in [4].)

Again, this interval is all that is required in many applications. However, if the researcher wishes to compare the attributable proportions for a group of risk factors, an adjustment for multiplicity would be necessary. This might be necessary if, for example, one wished to understand which risk factor should be focused on most for prevention of the disease. Here we would want to identify the largest attributable proportion and focus on disease prevention via reducing the effect of that risk factor.

2.4 Interval Estimation from GLMs

Though we have introduced notation for both linear models and GLMs, the rest of this section focuses on the particular GLMs utilized to illustrate the results in this paper. While the methods derived apply to any GLM, particular attention will be devoted to the logistic and Poisson models due to their applicability and popularity.

2.4.1 Logistic Regression Model

The logistic regression model is widely used in epidemiological and health science applications. The predictor variable in a logistic regression model can be either a single variable or a vector of variables. Thus, let $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ be a vector of predictor variables for $i = 1, \ldots, n$ where n is the total number of observations, and k

is the number of predictor variables. Thus, $\boldsymbol{x}_i$ is the i^{th} vector of predictor variables. Recall that for qualitative covariates, the $\boldsymbol{x}_i$'s would be defined as appropriate indicator variables. For example, in the preterm birth study $\boldsymbol{x}_i$ could be the maternal stress vector of predictor variables with binary elements indicating the presence of a particular source or level of a source of maternal stress. Thus, if the i^{th} case is a patient exposed only to dangerous neighborhoods as a source of maternal stress, we would code $\boldsymbol{x}_i = (1, 1, 0, 0, 0, \ldots, 0)$ where the first element has a 1 for the intercept term, the second place has a 1 to indicate the presence of stress in the form of a dangerous neighborhood, and the other elements of the vector are 0 indicating the patient was not exposed to the other sources of maternal stress. A logistic regression model assumes that the probability of a success for the i^{th} observation is $\pi(\boldsymbol{x}_i)$ where

$$\pi(\boldsymbol{x}_i) = P[Y_i = 1] = \frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}}} = \frac{e^{\beta_1 x_{i1} + \ldots + \beta_k x_{ik}}}{1 + e^{\beta_1 x_{i1} + \ldots + \beta_k x_{ik}}}, \quad i = 1, \ldots, n. \tag{2.14}$$

The matrix $\boldsymbol{X}$, as previously defined, contains information relating to the predicted value, $\boldsymbol{Y}$, for the model. Alternatively, we can express this model as

$$\phi(\boldsymbol{x}_i) = logit[\pi(\boldsymbol{x}_i)] = \ln[\frac{\pi(\boldsymbol{x}_i)}{1 - \pi(\boldsymbol{x}_i)}] = \boldsymbol{x}_i'\boldsymbol{\beta}, \quad i = 1, \ldots, n. \tag{2.15}$$

Now let the MLE of $\pi(\boldsymbol{x}_i)$ be denoted $\hat{\pi}(\boldsymbol{x}_i)$. We will make the usual assumption that the Y_i random variables are independent and binomially distributed with parameters m_i (assumed known) and $\pi(\boldsymbol{x}_i)$ given by (2.14), $i = 1, \ldots, n$. Thus, $\boldsymbol{W} = diag[m_i \pi(\boldsymbol{x}_i)(1 - \pi(\boldsymbol{x}_i))]$, $i = 1 \ldots, n$ and the asymptotic distribution of the MLE of $\boldsymbol{\beta}$ is given by (2.4). For $\hat{\boldsymbol{W}} = diag[m_i \hat{\pi}(\boldsymbol{x}_i)(1 - \hat{\pi}(\boldsymbol{x}_i))]$, $i = 1, \ldots, n$, the estimated covariance matrix of $\hat{\boldsymbol{\beta}}$ is given by (2.5).

When it is assumed that a logistic regression model is appropriate, the typical quantities of interest are the coefficients of the regression model, or the log odds ratios, $\beta_i, i = 1, \ldots, k$, and the response probabilities, $\pi(\boldsymbol{x}_i)$. These quantities relate to what was generally referred to as the expected response function. In particular, the expected response function for a logistic regression model is the response probability since this model assumes binomial sampling.

When considering response probabilities for single experimental units, one-at-a-time confidence intervals seem appropriate. (Confidence intervals provide additional information about the precision of the estimated response probability, so are often preferable to point estimates.) An appropriate confidence interval on the $logit(\pi(\boldsymbol{x}_i)) = \boldsymbol{x}_i'\boldsymbol{\beta}$ is computed by

$$\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} \pm z_{1-\alpha/2}\hat{\sigma}_{GLM}(\boldsymbol{x}_i) \tag{2.16}$$

where $z_{1-\alpha/2}$ is a z-percentile and $\hat{\sigma}_{GLM}(\boldsymbol{x}_i)$ is given by (2.6) with $\hat{\boldsymbol{W}}$ as previously defined. Let the upper and lower limits of (2.16) be denoted by U_{LOGIT} and L_{LOGIT}, respectively. Then we can apply the anti-logit and obtain bounds on the response probability. Thus, a $100(1-\alpha)\%$ confidence interval for the response probability is given by

$$\left(\frac{exp(L_{LOGIT})}{1+exp(L_{LOGIT})}, \frac{exp(U_{LOGIT})}{1+exp(U_{LOGIT})}\right). \tag{2.17}$$

Another quantity of interest from a logistic regression model is the odds ratio. One-at-a-time large sample confidence intervals can easily be constructed on the log odds ratios, as they are linear functions of the k logistic regression coefficients, $\boldsymbol{\beta}$. Thus we may utilize the asymptotic multivariate normal distribution of the maximum likelihood estimates of these k logistic regression coefficients, given by (2.4), to obtain large sample confidence intervals for the appropriate odds ratios. For illustration, suppose a particular odds ratio is given by $exp(\boldsymbol{c}_i\boldsymbol{\beta})$ for $\boldsymbol{c}_i = (c_{i1}, \ldots, c_{ik})$, a vector of appropriate constants. Then a one-at-a-time large sample confidence interval for this particular odds ratio is given by

$$\left(exp\{\boldsymbol{c}_i\hat{\boldsymbol{\beta}} - z_{\alpha/2}\hat{\sigma}_{GLM}(\boldsymbol{c}_i)\}, exp\{\boldsymbol{c}_i\hat{\boldsymbol{\beta}} + z_{\alpha/2}\hat{\sigma}_{GLM}(\boldsymbol{c}_i)\}\right) \tag{2.18}$$

where $\hat{\sigma}_{GLM}(\boldsymbol{c}_i)$ is given by (2.6). Typically, in epidemiological applications, the logistic regression model employed for computing the model-based odds ratios utilizes reference coding. When reference coding, as explained below, is utilized, special care must be taken in interpreting the model-based odds ratios.

2.4.2 Reference Coding for Logistic Regression

If a logistic regression model is employed for a categorical predictor variable, the design coding typically used necessitates that one of the levels of $\boldsymbol{x}$ be a reference level. Then odds ratios that result from the model coefficients are observed and compared to that reference level. Most often the reference level is a true control, but at times the reference level is arbitrary. When the reference level is informative, we may wish to: (1) estimate all the odds ratios relative to the reference level simultaneously, thereby allowing the researcher to assess the practical significance of any observed difference from the reference level while also providing the ability to evaluate which non-reference levels are significantly greater than or less than the reference level and (2) make comparisons for a pre-specified set of contrasts of the odds ratios. If the reference level is arbitrary, it seems reasonable to simultaneously compute all odds ratios or all odds ratio differences and then emulate one of the two scenarios described above. Again, if we wish to assess the practical significance of any estimates we need to estimate these quantities simultaneously rather than utilize a stepwise procedure. Note that for both above cases, when there is only one categorical predictor variable $\boldsymbol{x}$, then all inference procedures performed on the odds ratios resulting from the logistic regression model are equivalent to any similar analysis performed on the crude data in contingency table format. Differences will occur in models with multiple covariates.

The standard method for computing the odds ratios resulting from a logistic regression model using reference coding for the design matrix is to exponentiate the appropriate linear combinations of the estimated regression coefficients. For example, if we have a logit model such as (2.15), where there are k levels for our single predictor variable, then we can utilize the explanatory variables $x_1, \ldots, x_{k-1}$ with the covariate

Table 2.2: Examples of Estimated Odds Ratios

$e^{\hat{\beta}_1}$	The odds comparing the first non-reference level to the reference level
$e^{\hat{\beta}_2}$	The odds comparing the second non-reference level to the reference level
$\vdots$	$\vdots$
$e^{\hat{\beta}_2-\hat{\beta}_1}$	The odds comparing the second non-reference level to the first non-reference level
$\vdots$	$\vdots$
$e^{\hat{\beta}_k-\hat{\beta}_{k-1}}$	The odds comparing the k^{th} non-reference level to the $(k-1)^{th}$ non-reference level

vector at the reference level of our predictor variable equal to $\mathbf{0}$, that is, $x_1 = \ldots = x_{k-1} = 0$. Thus, $x_1, \ldots, x_{k-1}$ would be defined as indicator variables for the $k-1$ non-reference levels of our predictor variable. When this model is assumed, then we can interpret $e^{\hat{\beta}_1}$ as follows: the odds that $Y = 1$ for the first non-reference level is $e^{\hat{\beta}_1}$ times greater than that for the reference level. Table 2.2 illustrates other estimated odds ratios and their corresponding interpretations. The estimated odds ratios defined in Table 2.2 could then be utilized to construct confidence intervals that would aid in interpreting the model.

2.4.3 Loglinear or Poisson Model

Another model often employed in epidemiological studies is the loglinear model. The loglinear model relates the counts of a Poisson or multinomial distribution to a set of covariates. It may assume the total sample size is random or fixed, depending on whether the model assumes Poisson or multinomial sampling, respectively. For an

$I \times J$ contingency table let $N = I \times J$. Note that the number of cells in a contingency table, N, is distinct from the sample size or number of observations, denoted n, although they can be equal. Whenever the number of observations, n, is fixed, we have multinomial sampling for Y_i, $i = 1, \ldots, N-1$. However, when the sample size n is not fixed, we usually assume Poisson sampling for Y_i, $i = 1, \ldots, N$. For ease in notation let $n^* = N - 1$ for multinomial sampling and N for Poisson. Then the loglinear model is

$$log(\mu(\boldsymbol{x}_i)) = \boldsymbol{x}_i'\boldsymbol{\beta}, \quad i = 1, \ldots, n^* \tag{2.19}$$

where $E(\boldsymbol{Y}) = \boldsymbol{\mu} = (\mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_{n^*}))'$ is the vector of expected counts of the respective cells of the contingency table, $\boldsymbol{x}_i$ is a $1 \times k$ vector of covariates as described in (2.2), and $\boldsymbol{\beta}$ is a k-dimensional vector of model parameters. A loglinear model may also be expressed as

$$log(\mu(\boldsymbol{x}_i)) = \sum_{j=1}^{k} \beta_j x_{ij}, \quad i = 1, \ldots, n^*, \tag{2.20}$$

where each x_{ij} is the covariate value corresponding to β_i for the i^{th} level of $\boldsymbol{Y}$, $i = 1, \ldots, n^*$, and $j = 1, \ldots, k$. Recall the assumption that Y_i is a Poisson or multinomial random variable. Thus, the expectation of any Y_i is a positive value, $\mu(\boldsymbol{x}_i)$ for $i = 1, \ldots, n^*$.

The derivation of the large sample distribution of the model parameters depends on the sampling assumptions. When n is not fixed, we assume Poisson sampling. Then the MLE of $\hat{\boldsymbol{\beta}}$ is asymptotically normal with mean $\boldsymbol{\beta}$ and covariance matrix $\boldsymbol{V} = (\boldsymbol{X'}diag(\boldsymbol{\mu})\boldsymbol{X})^{-1}$. Notice that $\boldsymbol{W} = diag[\boldsymbol{\mu}]$. Thus, the estimated covariance matrix of $\hat{\boldsymbol{\beta}}$ is given by $\hat{\boldsymbol{V}} = c\hat{o}v(\hat{\boldsymbol{\beta}}) = [\boldsymbol{X'}diag(\hat{\mu})\boldsymbol{X}]^{-1}$. For Poisson sampling, we have,

$$\hat{\boldsymbol{\beta}} \dot{\sim} N(\boldsymbol{\beta}, (\boldsymbol{X'}diag(\boldsymbol{\mu})\boldsymbol{X})^{-1}). \tag{2.21}$$

Alternatively, when n, the overall sample size, is fixed we assume multinomial sampling. Typically, under multinomial sampling, we have interest in cell probabilities,

$\hat{\boldsymbol{\pi}} = \hat{\boldsymbol{\mu}}/n$. Here the $\hat{\boldsymbol{\pi}}$ are multivariate normal with mean $\boldsymbol{\pi}$ and covariance matrix $\boldsymbol{V} = cov(\hat{\boldsymbol{\beta}}) = \{\boldsymbol{X'}[diag\boldsymbol{(\mu)} - \boldsymbol{(\mu\mu'}/n)]\boldsymbol{X}\}^{-1} = \{n\boldsymbol{X'}[diag\boldsymbol{(\pi)} - \boldsymbol{\pi\pi'}]\boldsymbol{X}\}^{-1}$. Notice that $\boldsymbol{W} = diag\boldsymbol{(\mu)} - \boldsymbol{(\mu\mu'}/n)$. Additionally, the estimated covariance matrix for the regression parameters is given by $\hat{\boldsymbol{V}} = c\hat{o}v(\hat{\boldsymbol{\beta}}) = \{\boldsymbol{X'}[diag(\hat{\boldsymbol{\mu}}) - (\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}'/n)]\boldsymbol{X}\}^{-1} = \{\boldsymbol{X'}[diag(\hat{\boldsymbol{\pi}}) - \hat{\boldsymbol{\pi}}\hat{\boldsymbol{\pi}}']\boldsymbol{X}/n\}^{-1}$ when we have one multinomial sample. Thus for multinomial sampling,

$$\hat{\boldsymbol{\beta}} \dot{\sim} N(\boldsymbol{\beta}, (\boldsymbol{X'}[diag\boldsymbol{(\mu)} - (\boldsymbol{\mu\mu'}/n)]\boldsymbol{X})^{-1}). \tag{2.22}$$

Notice that the asymptotic normality of the parameters holds for both Poisson and multinomial sampling. When Poisson sampling is assumed, the expected response function is the mean cell count, $\boldsymbol{\mu}$. Alternatively, when multinomial sampling is assumed, the expected response function is $\boldsymbol{\pi}$. All inferences on the model parameters or any functions of the model parameters can be made via the asymptotic distributions previously stated. I will focus on the case where Poisson sampling is assumed as it is the customary assumption. Additionally, when Poisson sampling is assumed, the loglinear model is often referred to as a Poisson model. Intervals for response probabilities from multinomial loglinear models could be formed in a manner similar to that described for logistic regression, but this is rarely done with loglinear models. Instead, focus is usually on the estimated relative risks.

When utilizing a loglinear model the relative risk yields point estimates that are often more applicable to clinical situations than the odds ratio; thus we consider relative risk here. The use of the relative risk is very common in epidemiological applications, thus discussion of the relative risk will focus on these types of scenarios. Estimating the relative risk from a loglinear model is a particularly easy implementation since it may be shown that the estimated relative risk is simply $exp(\hat{\beta}_1)$ where $\hat{\beta}_1$ is the slope coefficient for x_1, the predictor variable indicating presence of the intervention. Other models, such as the logistic model, could be used similarly to estimate the relative risk, although other models do not always yield simple formulas.

Consider for instance, the case where we are estimating the relative risk from a loglinear model. The estimated relative risks would be of the form $exp(\hat{\beta}_j)$ where $\hat{\beta}_j$ is the estimated slope coefficient for the covariate x_j, $j = 1, \ldots, k$. Let $\boldsymbol{c}_i$ be a vector with the j^{th} element equal to 1 and all other elements equal to 0. A confidence band is formed by

$$\Big(exp\{\hat{\beta}_j - z_{\alpha/2} \times \hat{\sigma}_{GLM}(\boldsymbol{c}_j)\}, exp\{\hat{\beta}_j + z_{\alpha/2} \times \hat{\sigma}_{GLM}(\boldsymbol{c}_j)\}\Big)$$

where $\hat{\sigma}_{GLM}(\boldsymbol{c}_j)$ is given by (2.6) and $\boldsymbol{W}$ for Poisson sampling is given previously in by equation 2.21.

Other models using alternative canonical links could also be considered. For example, other GLMs are formed by utilizing the probit link, where $g = \Phi^{-1}(\pi(x))$, and the complementary log-log link, where $g = log(-log(1 - \pi(x)))$. These both assume a binomial sampling scenario and the usual focus is on the resulting probability of success, $\pi(x)$.

CHAPTER 3

Inferences on Quantities Estimated from a GLM

When utilizing GLMs, several quantities may be of interest. For example, the expected response, odds ratio, or relative risk may be estimated via the GLM. This section focuses on the case where the expected response function is of primary concern. All the methods discussed utilize the fact that GLMs may be expressed as

$$g(E(Y_i|\boldsymbol{x}_i)) = \boldsymbol{x}_i'\boldsymbol{\beta} \tag{3.1}$$

where Y_i is the response for the i^{th} observation, $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ik})$ is the vector of appropriate covariate values for the i^{th} observation, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$ is the vector of parameters, and g is the canonical link. (Specific assumptions and details on this model are given in equations (2.1) and (2.2).)

3.1 Inference on the Mean Response

This section will focus on the response probability or estimated mean response, $\pi(\boldsymbol{x}_i) = E(Y_i|\boldsymbol{x}_i)$, $i = 1, \ldots, n$ in a GLM assuming binomial or multinomial sampling. Alternatively, if we assume Poisson sampling, we would have interest in the expected cell counts, $\mu(\boldsymbol{x}_i) = E(Y_i|\boldsymbol{x}_i)$, $i = 1, \ldots, n$. This general methodology can be extended to the Poisson sampling scheme provided our inferences are on $\mu(\boldsymbol{x}_i)$, rather than $\pi(\boldsymbol{x}_i)$. Suppose we have a covariate $\boldsymbol{X}$ with k-dimensional domain $\mathcal{I}$ in a GLM of the form $g(E(Y|\boldsymbol{x}_i)) = \boldsymbol{x}_i'\boldsymbol{\beta}$, $i = 1, \ldots, n$. Then let $\mathcal{X} \subset \mathcal{I}$ be a compact subset of the domain which is of special interest. The intention of this section is to bound the expected response function, $E(Y|\boldsymbol{x}_i)$, for all $\boldsymbol{x}_i \in \mathcal{X}$ using confidence bounds on a

GLM. The subset $\mathcal{X}$ can be a set of the domain that is of special interest or it may be selected to answer a particular question. Discussion is restricted to the case where there is one covariate, but the methodologies may be extended to cases with many covariates. Even in the single covariate case, $\boldsymbol{X}$ may be a matrix if, for example, the model employs reference coding.

3.1.1 Previous Methods

Two primary approaches for simultaneously estimating the mean response function are discussed. The first is a conventional approach utilizing bounds similar to the well-known Scheffé bounds. The second is a modern approach utilizing solutions referred to as tube-formulas for constructing simultaneous intervals.

Scheffé bounds are a well-known methodology in simultaneous inferences, and are widely applied in linear models and generalized linear models. Some of the regularity conditions necessary for applying Scheffé bounds include that the sample size is sufficiently large and that the domain for the predictor variable is fixed [5]. Under these suitable regularity conditions, the maximum likelihood estimates (MLEs) of a linear model are multivariate normal with mean vector $\boldsymbol{\beta}$ and covariance matrix equal to the inverse of the Fisher information matrix $\sigma^2_{LM}\boldsymbol{F}$ where $\boldsymbol{F} = (\boldsymbol{X'X})^{-1}$. (See (2.8) for details on model assumptions.) In a standard regression model, Scheffé bounds are often utilized to obtain simultaneous intervals. These bounds are simultaneous for all $\boldsymbol{x}_i \in \mathbb{R}^k$ and thus are conservative for any finite set of such comparisons. Alternatively, Casella and Strawderman [6] derived Scheffé-type bounds for a regression model with restrictions assumed on the domain. These intervals are exact for this restricted domain. The advantage of assuming these restrictions is that the usual Scheffé bounds are conservative when the entire domain is not used. Piegorsch and Casella [7] utilized the Casella and Strawderman (CS) method to obtain simultaneous bounds on a logistic regression model. Specifically, they obtained Scheffé-type bounds on the $\boldsymbol{x}_i'\boldsymbol{\beta}$ in a logistic regression utilizing a restricted predictor variable domain of rectangular form. The method originally developed by Casella and Strawderman, and later extended by Piegorsch and Casella, is less conservative than the usual Scheffé bounds as it restricts the predictor variable space.

It is desirable at this point to reparameterize the model so that it is in the so-called diagonalized form (Casella and Strawderman [6]). This will simplify the calculations used hereafter. By the Spectral Theorem for symmetric matrices [8], the matrix $\boldsymbol{F}$ may be decomposed, given that $\boldsymbol{F}$ is symmetric. Thus, a linear model may be diagonalized by noting that $\boldsymbol{F} = \boldsymbol{UDU'}$ where $\boldsymbol{D} = diag(\lambda_i)$, a diagonal $k \times k$ matrix of the ordered eigenvalues of $\boldsymbol{F}$, and $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$ is the $k \times k$ matrix of corresponding orthonormal eigenvectors. Now define $\boldsymbol{Z}_{n\times k} = \boldsymbol{XUD^{-1/2}}$ and $\boldsymbol{\eta}_{k\times 1} = \boldsymbol{D^{1/2}U'\theta}$ where $\boldsymbol{UU'} = \boldsymbol{I}$ since each row,$\boldsymbol{u}_i$, is orthonormal. Thus,

$$\boldsymbol{Z\eta} = [\boldsymbol{XUD^{-1/2}}][\boldsymbol{D^{1/2}U'\theta}] = \boldsymbol{XUU'\theta} = \boldsymbol{XI\theta} = \boldsymbol{X\theta}$$

where the model may be written as $\boldsymbol{Y} = \boldsymbol{Z\eta} + \boldsymbol{\epsilon}$. Note that $\hat{\boldsymbol{\eta}} = \boldsymbol{D^{1/2}U'\hat{\theta}}$ is distributed $N_k(\boldsymbol{\eta}, \sigma^2_{LM}\boldsymbol{I})$, given that (2.4) holds.

The authors Casella and Strawderman [6] consider bounding linear models of the form $Y_i = \boldsymbol{x}_i\boldsymbol{\theta} + \epsilon_i$ with the usual restrictions on $\boldsymbol{\epsilon}$ (see (2.7)) and with a domain for $\boldsymbol{x}_i$ of the form $\Omega_{\boldsymbol{x}_i} = \{\boldsymbol{x}_i : \sum_{j=1}^{r} x_{ij}^2 \geq q^2 \sum_{j=r+1}^{k} x_{ij}^2\}$ where q is a fixed constant. When $r = 1$ these regions are cone-shaped regions, and if $r > 1$ there is no easy visualization of the space. Casella and Strawderman achieve exact results for bounding linear models for domains of this general form. Alternatively, both Casella and Strawderman [6] and Piegorsch and Casella [7] consider a more defined set of interval constraints on $\boldsymbol{X}$ which are of the form

$$R_{\boldsymbol{x}_i} = \{a_{11} < x_{i1} < a_{12}, a_{21} < x_{i2} < a_{22}, \ldots, a_{k1} < x_{ik} < a_{k2}\} \subset \Omega_{\boldsymbol{x}_i}$$

for a specified q. These intervals would be of particular interest in many experimental settings and thus are assumed for the remainder of this section.

The goal of the restricted-Scheffé procedure developed by Casella and Strawderman is to bound the regression function for all $\boldsymbol{x}_i \in \Omega_{\boldsymbol{x}_i}$. Thus, keeping in mind the objective of inference on $E(Y_i|\boldsymbol{x}_i) = \boldsymbol{x}_i'\boldsymbol{\theta}$, consider

$$S(\Omega_{\boldsymbol{x}_i}) = \{\boldsymbol{\theta} : (\boldsymbol{x}_i'\hat{\boldsymbol{\theta}} - \boldsymbol{x}_i'\boldsymbol{\theta})^2 \leq d^2\sigma_{LM}^2\boldsymbol{x}_i'\boldsymbol{F}^{-1}\boldsymbol{x}_i \; \forall \boldsymbol{x}_i \in \Omega_{\boldsymbol{x}_i}\} \tag{3.2}$$

where d is an arbitrary constant. Casella and Strawderman derive a procedure to calculate the value of d that yields,

$$P[S(\Omega_{\boldsymbol{x}_i})] = 1 - \alpha. \tag{3.3}$$

Their derivation involves considering a domain for $\boldsymbol{Z}$ similar to $\Omega_{\boldsymbol{x}_i}$,

$$\Omega_{\boldsymbol{z}_i} = \{\boldsymbol{z}_i : \sum_{j=1}^{r} z_{ij}^2 \geq q^2 \sum_{j=r+1}^{k} z_{ij}^2\}$$

where q is a fixed constant. Thus, Casella and Strawderman prove that for a specified d

$$P[S(\Omega_{\boldsymbol{z}_i})] = P[\{\boldsymbol{\eta} : (\boldsymbol{z}_i'\hat{\boldsymbol{\eta}} - \boldsymbol{z}_i'\boldsymbol{\eta})^2 \leq d^2\sigma_{LM}^2\boldsymbol{z}_i'\boldsymbol{z}_i \; \forall \boldsymbol{z}_i \in \Omega_{\boldsymbol{z}_i}\}] = 1 - \alpha \tag{3.4}$$

where $\hat{\boldsymbol{\eta}}$ is the MLE under spectral decomposition. Notice that the only difference between the sets $S(\Omega_{\boldsymbol{x}_i})$ and $S(\Omega_{\boldsymbol{z}_i})$ is the space we are operating in. Recall the form assumed about the domain of interest, $R_{\boldsymbol{x}_i}$. This is a convex set in $\mathbb{R}^k$. Thus, the image of this set, $R_{\boldsymbol{z}_i}$, will also be convex, since a linear map preserves convexity. Note that for $\boldsymbol{\gamma} = \frac{\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}}{\sigma_{LM}}$, the quantity

$$S(\Omega_{\boldsymbol{z}_i}) = \{\boldsymbol{\gamma} : (\boldsymbol{\gamma}\boldsymbol{z}_i)^2 \leq d^2\boldsymbol{z}_i'\boldsymbol{z}_i \quad \forall \boldsymbol{z}_i \in \Omega_{\boldsymbol{z}_i}\}. \tag{3.5}$$

Assume this form of $S(\Omega_{\boldsymbol{z}_i})$ henceforth. Since we have a domain of the form $\Omega_{\boldsymbol{z}_i}$ and wish to obtain a Scheffé-type probability band, then via Theorem 1 in [6] we have

$$P(S(\Omega_{\boldsymbol{z}_i})) = P(\chi_k^2 \leq d^2) + P(E_{r,s}(b, d^2)) \tag{3.6}$$

where $E_{r,s}(b, d^2) = \{(\chi_r^2, \chi_s^2) : \chi_r^2 + \chi_s^2 \geq d^2, (a\chi_r + b\chi_s)^2 \leq d^2, \chi_r^2 \leq q^2\chi_s^2\}$, $a^2 + b^2 = 1$, and χ_r^2 and χ_s^2 are independent chi-square random variables. Also note that q is a fixed constant determined by $\Omega_{\boldsymbol{z}_i}$ and b, r, and s are determined by the value of d and the parameters of the problem. (For specific details see Casella and Strawderman [6].) A solution for the quantity d which yields appropriate simultaneous intervals may be found by setting the right hand side of (3.6) equal to $1 - \alpha$ and solving for d. Casella and Strawderman applied this theorem to a linear model achieving exact results for all $\boldsymbol{x}_i \in \Omega_{\boldsymbol{x}_i}$, thus yielding a conservative solution for all $\boldsymbol{x}_i \in R_{\boldsymbol{x}_i} \subset \Omega_{\boldsymbol{x}_i}$. The details of the derivation of the appropriate $\Omega_{\boldsymbol{z}_i}$ (and hence $\Omega_{\boldsymbol{x}_i}$) are given by Casella and Strawderman [6]. The resulting restricted-Scheffé intervals are of the form $\hat{Y} \pm d\hat{\sigma}(\boldsymbol{x}_i)$ where d is a critical value determined by an algorithm which is described in Appendix C.

Piegorsch and Casella applied this procedure specifically to logistic regression. However, it has not been applied for use in a generic GLM, and it is unclear how these bounds will perform for other generalized linear models. Note that although this method is still conservative, it is less conservative than the conventional Scheffé bounds, as it is not applicable for the entire predictor variable space.

As another alternative to the Scheffé-type bounds, Sun, Loader, and McCormick (2000) [9] (SLM) proposed a solution for simultaneously estimating the mean response for the general class of GLMs with all $\boldsymbol{x}_i$ in a compact set. This general method of bounding a regression function, called simultaneous confidence regions (SCR), can account for a variety of linear and nonlinear models. Specifically, the SCR bounds can be applied when there are heteroscedastic and non-additive error terms, as is the case for many GLMs. The SCR bounds utilize error expansions to approximate the non-coverage probability for a GLM. They are far less conservative than Scheffé solutions and perform exceptionally well for moderate sample sizes.

The SCR bounds are based on applying the so-called tube formula due to Naiman [10] with various possible adjustments. The tube formula provides a lower bound for the coverage probability of a confidence band of a regression function over a specified closed set. However, the tube-formula assumes the error distribution of the model is normal. Clearly, this is not a valid assumption if we have a GLM, although it does provide a starting point for constructing confidence bounds, as the large sample error distribution is approximately normal. Obviously, this assumption will be problematic for smaller sample sizes.

The basic tube-formula methodology is described by Naiman in his 1986 paper (Naiman [10]). This paper outlines a solution for constructing simultaneous confidence bands on polynomial regression models of the form

$$Y_i = \sum_{j=1}^{k} \theta_j f_j(x_i) + e_i,\ i = 1, \ldots, n,\ x_i \in I. \tag{3.7}$$

Here it is assumed that I is a closed interval in $\mathbb{R}$, that $e_i \sim iidN(0, \sigma^2_{LM})$ with σ^2_{LM} unknown, and that θ_j $(j = 1, \ldots, k)$ are unknown constants. The vector

$$\boldsymbol{f}(x_i) = (f_1(x_i), \ldots, f_k(x_i))'$$

maps from I to $\mathbb{R}^k$. Naiman's intent is to provide simultaneous confidence bounds on $E(Y_i|x_i) = \boldsymbol{\theta' f}(x_i)$ for all $\boldsymbol{x}_i \in \boldsymbol{I}$ where an estimate $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_k)'$ is available such that $\hat{\boldsymbol{\theta}}$ is distributed $N(\boldsymbol{\theta}, \sigma^2_{LM}\boldsymbol{F})$ with σ^2_{LM} unknown, $\boldsymbol{F}$ known and s^2_{LM} an independent estimator of σ^2_{LM} such that $\frac{\nu s^2_{LM}}{\sigma^2_{LM}} \sim \chi^2_\nu$ $(\nu = n - k)$.

In order to understand how Naiman derives these bounds, consider alternatively another mapping, γ, from I to the unit sphere S^{k-1} centered at the origin of $\mathbb{R}^k$, such that γ is piecewise differentiable and

$$\Lambda(\gamma) = \int_I ||\gamma'(x)||\partial x \tag{3.8}$$

is finite. Since γ maps $\boldsymbol{I}$ to the unit sphere, S^{k-1}, it will be considered in place of the primary mapping $\boldsymbol{f}(x_i)$. Specifically, it is the projection of $\boldsymbol{f}(x_i)$ on S^{k-1}. Note that $\gamma(x) = \|\boldsymbol{Pf}(x_i)\|^{-1}\boldsymbol{Pf}(x_i)$ for $x_i \in I$ where $\boldsymbol{P}$ is a $k \times k$ matrix such that $\boldsymbol{F} = \boldsymbol{P'P}$. The quantity $\Lambda(\gamma)$ is called the path length, as it measures the length of the path, γ, a continuous mapping essentially connecting the points in the image of $\boldsymbol{f}$. The image of the path in S^{k-1} is then denoted by $\Gamma(\gamma) = \{\boldsymbol{\gamma}(x) : x \in I\}$. The goal is to bound Γ with a tube via bounding the $\Lambda(\gamma)$, which equivalently bounds the regression function, $\boldsymbol{f}(x_i)$, on I as they share the same domain.

Regarding the image of the path function, $\Gamma(\gamma)$, Naiman demonstrates that

$$\mu(\Gamma(\gamma)_{(g)}) \leq min(F_{k-2,2}[\frac{2(g^{-2}-1)}{k-2}] \times \frac{\Lambda}{2\pi} + \frac{1}{2}F_{k-1,1}[\frac{g^{-2}-1}{k-1}], 1) \qquad (3.9)$$

where $g \in [0,1]$ such that $\Gamma_g = \{\boldsymbol{u} \in S^{k-1} : c_\Gamma(\boldsymbol{u}) \geq g\}$ (a set of points in S^{k-1} that surround Γ) with $c_\Gamma(\boldsymbol{u}) = sup\{\boldsymbol{u'v} : \boldsymbol{v} \in \Gamma\}$ for any $\boldsymbol{u} \in S^{k-1}$ and μ is the uniform measure. Naiman then applies these results to obtain confidence bands of the form $\hat{\boldsymbol{\theta}}'\boldsymbol{f}(x_i) \pm d(\hat{\sigma}_{LM})(\boldsymbol{f}(x_i)'\boldsymbol{F}\boldsymbol{f}(x_i))$. The intervals are formed utilizing the critical value d which is determined by setting

$$1 - \int_0^{1/d} min(F_{k-2,2}[\frac{2(dt^{-2}-1)}{k-2} \times \frac{\Lambda}{\pi} + \frac{1}{2}F_{k-1,1}[\frac{dt^{-2}-1}{k-1}], 1)f_T(t)\partial t \qquad (3.10)$$

equal to $1-\alpha$ and solving for d. Here $f_T(t)$ is the density of a random variable T where $rT^2 \sim F_{\nu,r}$.

Utilizing these tube-formula bounds, SLM form simultaneous bounds on the expected response function for a GLM. Recall that Naiman derived these bounds assuming normally distributed residuals. Clearly, generalized linear models only have normally distributed residuals asymptotically. Thus, the tube-formulas were originally applied directly via the asymptotic normality of the residuals to obtain asymptotic simultaneous confidence bands. Modifications were then made to the usual tube-based bounds to improve them for small to moderate sample sizes.

The following description outlines how to apply the tube-formula bounds to GLMs. Let the maximum likelihood estimate (MLE) of a predicted response for a GLM at $\boldsymbol{x}_i$ be denoted by $\hat{Y}_i$. Ultimately, the interval desired is of the form

$$I_d(\boldsymbol{x}_i) = (g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} - d\hat{\sigma}_{GLM}(\boldsymbol{x}_i)), g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} + d\hat{\sigma}_{GLM}(\boldsymbol{x}_i))) \;\; \forall \boldsymbol{x}_i \in \mathcal{X}$$

where $[\hat{\sigma}_{GLM}(\boldsymbol{x}_i)]^2$ is the asymptotic variance of $\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}$ and $\mathcal{X}$ is a particular compact subset of the domain. The tube formulas will enable us to find a value d such that

$$P[g(E(Y_i|\boldsymbol{x}_i)) \in I_d(\boldsymbol{x}_i), \quad for\ all\ \boldsymbol{x}_i \in \mathcal{X}] \geq 1 - \alpha. \tag{3.11}$$

Applying the tube formula directly to solve for d, yields what SLM term a naive SCR. We will utilize the notation d_{TUBE} to indicate a critical value calculated in this manner. This solution performs adequately when the asymptotic distribution of the residuals is nearly normal. However, this method will not attain the desired confidence level when the sample size is relatively small, as typically the residuals are nonnormal discrete random variables for GLMs. In order to improve the small sample performance, the authors consider some modifications to the tube-formula.

They begin by approximating the sampling distribution of the residual via construction of expansions on the estimated model. This approximation of the sampling distribution will be utilized to obtain a critical point for the confidence interval formula that is adjusted with respect to the bias introduced by the MLEs. Consider the random process $W_n(\boldsymbol{x}_i) = \frac{g(\hat{Y}_i) - g(E(Y_i|\boldsymbol{x}_i))}{\hat{\sigma}_G(\boldsymbol{x}_i)}$ where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ik})$ is the i^{th} vector of $\boldsymbol{X} = (\boldsymbol{x}_i, \ldots, \boldsymbol{x}_n)'$ and $[\hat{\sigma}_G(\boldsymbol{x}_i)]^2$ is the asymptotic variance of $g(\hat{Y_i})$. This converges in distribution to a Gaussian random field. Let $W(\boldsymbol{x}_i)$ be a random variable with the same distribution as the limiting distribution of $W_n(\boldsymbol{x}_i)$. Then the bias behaves like $|W_n(\boldsymbol{x}_i) - W(\boldsymbol{x}_i)|$. This equivalent expression of the bias may be bounded via inverse Edgeworth expansions. SLM propose three corrections that can aid in correcting the bias introduced from estimating the regression parameters in a GLM with MLEs. These three solutions are based on the inverse Edgeworth expansion of the random process given by,

$$|W_n(\boldsymbol{x}_i)| = |W(\boldsymbol{x}_i)| - p_2(\boldsymbol{x}_i, W_n(\boldsymbol{x}_i)) \tag{3.12}$$

where the term subtracted can be thought of as a correction for the bias of the process. It is based on the centered moments of the process $W_n(\boldsymbol{x}_i)$ denoted κ_i for $i = 1, 2, 3, 4$

and is given by,

$$\begin{aligned} p_2(\boldsymbol{x}_i, Z) = -Z\{\frac{1}{2}[\kappa_2(\boldsymbol{x}_i) - 1 + \kappa_1^2(\boldsymbol{x}_i)] \\ +\frac{1}{24}[\kappa_4(\boldsymbol{x}_i) + 4\kappa_1(\boldsymbol{x}_i)\kappa_3(\boldsymbol{x}_i)](Z^2 - 3) \\ +\frac{1}{72}\kappa_3^2(\boldsymbol{x}_i)(Z^4 - 10Z^2 + 15)\} = O(n^{-1}). \end{aligned} \qquad (3.13)$$

The κ_i's may be computed as detailed by Hall(1992, [11]). Details are provided in Appendix D.

The inverse Edgeworth expansions (3.13) are then also utilized to account for the bias typically observed in the MLEs of generalized linear models. A first version of a corrected SCR, denoted SCR1, is a solution where the bias term, $p_2(\boldsymbol{x}_i, W_n(\boldsymbol{x}_i))$, is bounded. First consider the supremum of the bias term, $R_p^{'} = \begin{array}{c}\sup \\ \boldsymbol{x}_i \in \mathcal{X}\end{array}\{p_2(\boldsymbol{x}_i, W_n(\boldsymbol{x}_i))\} = O_p(1/n)$. We want to find a positive constant $R_p^{'}$ such that $P[R_p^{'} \leq r_p^{'}] = 1 - \alpha$ as $n \longrightarrow \infty$. Details are given in Sun, Loader, and McCormick [9] and Hall [11]. Additionally, specific calculation procedures are described in Appendix D.

These $r_p^{'}$ values are then used to correct the bias in the choice of d via the tube-formula method. Namely, our new interval is given by

$$\left(g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} - d_{SCR1}\hat{\sigma}_{GLM}(\boldsymbol{x}_i)), g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} + d_{SCR1}\hat{\sigma}_{GLM}(\boldsymbol{x}_i))\right) \qquad (3.14)$$

where the new critical point, d_{SCR1} is equal to $d_{TUBE} - |r_p'|$ where d_{TUBE} is the aforementioned solution.

Another version of the corrected SCR, SCR2, considers the modified process, $W_n^0(\boldsymbol{x}_i) = \frac{W_n(\boldsymbol{x}_i) - \kappa_1(\boldsymbol{x}_i)}{\sqrt{\kappa_2(\boldsymbol{x}_i)}}$, such that $|W_n^0(\boldsymbol{x}_i)| = |W(\boldsymbol{x}_i)| - q_2(\boldsymbol{x}_i, W_n^0(\boldsymbol{x}_i))$ with q_2 similar to p_2. The tube formula is then applied to $W_n^0(\boldsymbol{x}_i)$. Bounding this normalized process, $W_n^0(\boldsymbol{x}_i)$, further corrects the bias. Doing this results in confidence bounds on the $E(Y_i|\boldsymbol{x}_i)$ which are an improvement of the tube method applied directly to $W_n(\boldsymbol{x}_i)$. It is of interest to note that this method corrects the bias via a first level approximation. The resulting confidence region is of the form,

$$\left(g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} - d_{SCR2}\hat{\sigma}_{GLM}(\boldsymbol{x}_i)), g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} + d_{SCR2}\hat{\sigma}_{GLM}(\boldsymbol{x}_i))\right) \qquad (3.15)$$

where d_{SCR2} is this bias-corrected solution for the critical value. Note that this is just a critical value, like d, that is corrected for the bias. SLM term this a two-sided corrected SCR. Recall that it utilizes the modified Gaussian process that corrects the bias inherit in the MLE estimates and finds a critical value that adjusts for that bias.

A last solution, called the centered SCR (SCR3), begins by estimating the mean and variance of the Gaussian process $W_n(\boldsymbol{x}_i)$. These are the centered moments and are given by $\hat{\kappa}_1(\boldsymbol{x}_i)$ and $\hat{\kappa}_2(\boldsymbol{x}_i)$, respectively. These essentially move and rescale the confidence region so it is no longer biased. The tube-based critical value d_{TUBE} is again involved, so the final interval is

$$\Big(g^{-1}((\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^* - d_{TUBE}\hat{\sigma}_i^*), g^{-1}((\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^* + d_{TUBE}\hat{\sigma}_i^*)\Big) \tag{3.16}$$

where $(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^* = \boldsymbol{x}_i'\hat{\boldsymbol{\beta}} - \hat{\kappa}_1(\boldsymbol{x}_i)\hat{\sigma}_{GLM}(\boldsymbol{x}_i)$ and $\hat{\sigma}_i^* = \hat{\sigma}_{GLM}(\boldsymbol{x}_i)\sqrt{\hat{\kappa}_2(\boldsymbol{x}_i)})$. These formulas are given in Appendix D.

3.1.2 Proposed Methods

Expanding on the methodologies presented in the previous section, I have developed two new approaches for estimating a mean response function over a specified compact set via confidence regions.

The first proposed method is based on the restricted-Scheffé bounds developed by Casella and Strawderman and further refined by Piegorsch and Casella. In Piegorsch and Casella [7], the authors derive and implement conservative simultaneous bounds on the response probabilities of logistic regression models for rectangular domains. I have generalized these bounds on the expected response function for any GLM. Outlined below is the method by which these bounds may be computed.

For any GLM, let the anti link function be g^{-1} so that

$$E(Y_i|\boldsymbol{x}_i) = g^{-1}(\boldsymbol{x}_i'\boldsymbol{\beta}), \quad i = 1, \ldots, n \tag{3.17}$$

where $E(Y_i|\boldsymbol{x}_i)$ denotes the response probability (logistic regression) or the mean response (loglinear models or Poisson regression) for the specified covariate levels given by $\boldsymbol{x}_i$ (Complete model specifications are given in (2.2)). Recall that the MLE of $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}$, is asymptotically normal with mean $\boldsymbol{\beta}$ and $k \times k$ covariance matrix $\boldsymbol{V}$ where $\boldsymbol{V} = (\boldsymbol{X'WX})^{-1}$ with $\boldsymbol{W}$ defined in (2.3).

Applying the restricted-Scheffé procedure of Casella and Strawderman to GLMs yields appropriate conservative simultaneous confidence intervals for the mean response. Casella and Strawderman assumed that the MLEs of the regression parameters were normally distributed with a specified mean vector and covariance matrix. For our case we only have asymptotic normality of the MLEs and therefore, the probability in (3.6) is not exactly $1 - \alpha$ but instead converges to $1 - \alpha$ as $n \to \infty$. We will also require a slightly different definition for $S(\Omega_{\boldsymbol{x}_i})$ and $S(\Omega_{\boldsymbol{z}_i})$. Recall $S(\Omega_{\boldsymbol{x}_i}) = \{\boldsymbol{\theta} : (\boldsymbol{x}_i'\hat{\boldsymbol{\theta}} - \boldsymbol{x}_i'\boldsymbol{\theta})^2 \leq d^2\sigma_{LM}^2\boldsymbol{x}_i'\boldsymbol{F}^{-1}\boldsymbol{x}_i \ \forall \boldsymbol{x}_i \in \Omega_{\boldsymbol{x}_i}\}$ for linear models. Here however $S(\Omega_{\boldsymbol{x}_i}) = \{\boldsymbol{\beta} : (\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} - \boldsymbol{x}_i'\boldsymbol{\beta})^2 \leq d^2\boldsymbol{x}_i'\boldsymbol{V}^{-1}\boldsymbol{x}_i \ \forall \boldsymbol{x}_i \in \Omega_{\boldsymbol{x}_i}\}$. Notice that the inequalities in both sets have an upper bound given by the variance of $\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}$ or $\boldsymbol{x}_i'\hat{\boldsymbol{\theta}}$, respectively. $S(\Omega_{\boldsymbol{z}_i})$ will have a similar definition for GLMs and the diagonalization described in section 4.1 applies with $\boldsymbol{F} = \boldsymbol{V}$. Thus, for any $S(\Omega_{\boldsymbol{z}_i})$ of the form (3.5) generalized appropriately for a GLM,

$$P(S(\Omega_{\boldsymbol{z}_i})) \to P(\chi_k^2 \leq d^2) + P(E_{r,s}(b, d^2)) \tag{3.18}$$

as $n \to \infty$ where

$$E_{r,s}(b, d^2) = \{(\chi_r^2, \chi_s^2) : \chi_r^2 + \chi_s^2 \geq d^2, (a\chi_r + b\chi_s)^2 \leq d^2, \chi_r^2 \leq q^2\chi_s^2\}, \tag{3.19}$$

where $a^2 + b^2 = 1$ and χ_r^2 and χ_s^2 are independent chi-square random variables. Note that q is a fixed constant determined by the particular $\Omega_{\boldsymbol{x}_i}$. (Recall we will choose the smallest $\Omega_{\boldsymbol{x}_i} \supset R_{\boldsymbol{x}_i}$.) Additionally, the constants b, r, and s are determined by the value of d and the parameters of the problem. (Details of the computation of b, r, and s specifically for GLMs are given in Appendix C.) Notice that the coverage probability of the set $S(\Omega_{\boldsymbol{z}_i})$ is the sum of the usual coverage probability of the Scheffe set ($P(\chi_p^2 \leq d^2)$) and a probability that adjusts for the restricted domain. Recall that these bounds are derived assuming a domain of the form $\Omega_{\boldsymbol{x}_i}$. If a set of the form $R_{\boldsymbol{x}_i}$ is of interest, an approximate answer may still be found as in Piegorsch and Casella. In order to find a bound for a domain of the form $R_{\boldsymbol{x}_i}$, the smallest set of the form $\Omega_{\boldsymbol{x}_i}$ is established such that $\Omega_{\boldsymbol{x}_i}$ contains $R_{\boldsymbol{x}_i}$. (This procedure is detailed in Appendix C.) We may consider sets on either the $\boldsymbol{X}$ domain, $\Omega_{\boldsymbol{x}_i}$ and $R_{\boldsymbol{x}_i}$, or their equivalent sets on the $\boldsymbol{Z}$ domain, $\Omega_{\boldsymbol{z}_i}$ and $R_{\boldsymbol{z}_i}$.

Recall that the method of Casella and Strawderman provides simultaneous bounds on a linear model for a transformed domain of the form $\Omega_{\boldsymbol{z}_i}$, and thus equivalently for $\Omega_{\boldsymbol{x}_i}$. The adapted method of Piegorsch and Casella computes these bounds for domains of the form $R_{\boldsymbol{x}_i}$ in logistic regression models. I propose extending these bounds for use in any generalized linear model with a canonical link. The simultaneous bounds may be transformed from $\boldsymbol{x}_i'\boldsymbol{\beta}$, $\boldsymbol{x}_i \in R_{\boldsymbol{x}_i}$, to the expected response function via the anti-link function.

In order to apply the Casella-Strawderman results to GLMs, we must first show that the probability of the set $S(\Omega_{\boldsymbol{x}_i})$ converges to $1-\alpha$.

Corollary 3.1 *If $\hat{\boldsymbol{\beta}}$ is asymptotically normal with mean vector $\boldsymbol{\beta}$ and covariance matrix $\boldsymbol{V}$, then*

$$P(S(\Omega_{\boldsymbol{z}_i})) \to P(\chi_k^2 \leq d^2) + P(E_{r,s}(b,d^2)) \qquad \text{as} \quad n \to \infty. \tag{3.20}$$

with $E_{r,s}(b,d^2)$ given by (3.19) where q is determined by the particular $\Omega_{\boldsymbol{z}_i}$ considered and appropriate constants b, r, and s.

Proof: Recall $\Omega_{\boldsymbol{z}_i} = \{\boldsymbol{z}_i : \sum_{j=1}^{r} z_{ij}^2 \geq q^2 \sum_{j=r+1}^{k} z_{ij}^2\}$ for a specified constant q. Here $\boldsymbol{z}_i$ is the diagonalized form of $\boldsymbol{x}_i$. Theorem 1 from Casella and Strawderman [6] gave exact equality of the same probabilities in (3.20) under exact normality for $\hat{\boldsymbol{\beta}}$. Consequently, under asymptotic normality we have (3.20). □

Unfortunately (3.20) requires $\boldsymbol{V}$ known. Consider the following corollary to the Casella and Strawderman theorem, that holds for GLMs.

Corollary 3.2 *If $\hat{\boldsymbol{\beta}}$ is asymptotically normal with mean vector $\boldsymbol{\beta}$ and covariance matrix $\boldsymbol{V}$, estimated by $\hat{\boldsymbol{V}}$, then for $S(\Omega_{\boldsymbol{z}_i})$ utilizing $\hat{\boldsymbol{V}}$ instead of $\boldsymbol{V}$ (3.20) still holds.*

Proof: Let $\hat{\boldsymbol{V}} = (\boldsymbol{X'}\hat{\boldsymbol{W}}\boldsymbol{X})^{-1}$ then since $\hat{\boldsymbol{\beta}} \xrightarrow{p} \boldsymbol{\beta}$ we have $\hat{\boldsymbol{W}} = \boldsymbol{W}|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} \xrightarrow{p} \boldsymbol{W}$. Thus $\hat{\boldsymbol{V}} = (\boldsymbol{X'}\hat{\boldsymbol{W}}\boldsymbol{X})^{-1} \xrightarrow{p} \boldsymbol{V} = (\boldsymbol{X'}\boldsymbol{W}\boldsymbol{X})^{-1}$. Consequently the convergence in (3.20) also holds when $\boldsymbol{V}$ is estimated by $\hat{\boldsymbol{V}}$. □

Now let d_{CS} be the value of d such that the probability on the right-hand side of (3.20) is $1-\alpha$. Then

$$P(g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} - |d_{CS}|\hat{\sigma}_i^*) \leq E(Y_i|\boldsymbol{x}_i) \leq g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} + |d_{CS}|\hat{\sigma}_i^*)) \rightarrow 1-\alpha \quad \text{as} \quad n \rightarrow \infty \tag{3.21}$$

where $\hat{\sigma}_i^* = (\boldsymbol{x}_i'\hat{\boldsymbol{V}}^{-1}\boldsymbol{x}_i)^{1/2}$.

We have theoretically demonstrated that the restricted-Scheffé bounds of Casella and Strawderman may be applied to GLMs, but the computational details remain unclear. A detailed computational algorithm to compute the restricted-Scheffé bounds for any GLM is given in Appendix C.

As an alternative to the restricted-Scheffé bounds, we can also apply an estimate of $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}^*$ to the simultaneous confidence regions (SCRs) developed by SLM [9] and the restricted-Scheffé bounds for GLMs. Recall that SLM derived four SCR bounds. First, the tube-based bounds were applied to the maximum likelihood estimators by appealing to their asymptotic normal distribution (d_{TUBE}). Second, the bias was bounded and then the tube-formula solution was applied (d_{SCR1}). Third, the SCR bounds were derived for a modified process which accounts for the bias in the distribution of the MLE (d_{SCR2}). And fourth, the random process was centered and rescaled to correct the bias before applying the tube-based bounds (centered SCR). The various solutions all attempt to correct the bias of the maximum likelihood estimates for GLMs. In particular, when the sample size is small the MLEs are highly nonnormal, and the adjustments to the tube formulas are especially helpful. I propose estimating simultaneous SCR bounds, and restricted-Scheffé bounds, that are not based on the MLE, but on an alternative to the MLE, the penalized maximum likelihood estimator (pMLE). The pMLE is a bias-corrected estimate that is closely related to the usual MLE. The tube-formula bounds can then be applied utilizing the pMLE estimates rather than the MLEs. In particular, the naive SCR and centered SCR (SCR3) bounds can easily be applied. The difference between the proposed method and the methods of Sun et al. (2000) [9] is that utilizing the pMLE estimates doesn't merely "correct" the bias, but prevents the bias from occurring (in the first order) a priori. No additional bias correcting procedures will be necessary since the pMLE estimate has little bias from the start. Additionally, this method eliminates inestimable model parameters (0 or ∞) due to zero cell counts in GLMs. This difficulty was not resolved by the methodologies presented by SLM, and can be particularly troublesome with small to moderate sample sizes. Also, recall that the bias corrections employed in some of the SCR bounds were only bias-reducing asymptotically. This procedure is bias-reducing for any sample size.

The penalized maximum likelihood estimate (pMLE) was developed by David Firth [12] as an alternative to the MLE. Firth developed these estimators for use in models, such as GLMs, where the typical MLE is known to be a biased estimate. Specifically, all of the derivations depend on the model belonging to the general exponential class. These distributions have the general form

$$f(t, \theta) = exp\{[(t\theta - K(\theta))]/a(\phi) + c(t, \phi)\}, \tag{3.22}$$

where when ϕ, the dispersion parameter, is known, this simplifies to

$$f(t, \theta) = exp\{(t\theta - K(\theta))\}$$

[13]. Although this is an unusual form for an exponential class model, it lends itself quite well to a derivation later in the section, and this form can be shown to be equivalent to the standard form under the correct reparameterization. The general pMLE procedure involves penalizing the score function via the Jeffreys invariant prior for the particular parameter of interest. This penalty yields estimates of the regression parameters that are unbiased in the first-order. Typically, when computing an MLE, the first derivative of the log likelihood, often called the score function, is computed and set equal to 0, yielding the MLE as the solution. For estimating a set of parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$, let the usual vector of score functions be denoted $\boldsymbol{U(\theta)} = (U_1(\boldsymbol{\theta}), \ldots, U_k(\boldsymbol{\theta}))'$. Note that $U_r(\boldsymbol{\theta})$ is the derivative of the log-likelihood (score function) with respect to the r^{th} parameter. Firth proposes shifting the score function to correct the bias present in most MLE estimates for GLMs. The shift is determined by the estimated bias, $b(\boldsymbol{\theta})$ and information matrix, $i(\boldsymbol{\theta})$. Then the shifted or penalized score function, $\boldsymbol{U^*(\theta)} = U(\boldsymbol{\theta}) - i(\boldsymbol{\theta})b(\boldsymbol{\theta}) = (U_1^*(\boldsymbol{\theta}), \ldots, U_k^*(\boldsymbol{\theta}))'$ is set equal to $\mathbf{0}$ and a penalized MLE is the solution. Thus, the pMLE, $\boldsymbol{\theta}^*$, is the solution to $\boldsymbol{U^*(\theta)} = \mathbf{0}$. The details of estimating the bias are given in Firth [12], but as previously noted, the calculations are based on utilizing a Jeffreys prior for the parameters of interest, $\boldsymbol{\theta}$. Firth applies this estimator to the logistic regression model, but the methodology can be applied to any exponential class model, specifically any GLM. Firth states that for a GLM with a canonical link the modified score function is given by

$$U_r^*(\boldsymbol{\theta}) = U_r(\boldsymbol{\theta}) + \frac{1}{2\phi}\sum_{i=1}^{n}\left(\frac{\kappa_{3i}}{\kappa_{2i}}\right) h_i x_{ir}, \quad (r = 1, \ldots, k), \tag{3.23}$$

where $U_r(\boldsymbol{\theta})$ is the derivative of the log likelihood function for the r^{th} parameter, n is the number of observations Y_i, ϕ is the dispersion parameter in (3.22), and h_i is the i^{th} diagonal of the hat matrix (the leverage). The leverage, or the i^{th} diagonal of the hat matrix, is a measure of the distance between each observation, x_{ir}, and the mean, $\overline{x}$. (See McCullough and Nelder [13] for an explicit definition.) Additionally, κ_{ti} is the

t^{th} $(t = 2, 3)$ cumulant, or t^{th} central moment, of Y_i, $i = 1, \ldots, n$. The cumulants may be calculated using the cumulant generating function $K_i(s) = log(M_i(s))$ where $M_i(s)$ is the moment generating function for the distribution of the dependent variable, Y_i. Each cumulant is found by taking the derivative of $K_i(s)$ with respect to s and then letting $s = 0$. For example, $\kappa_{2i} = K_i^{(2)}(0)$ where $K_i^{(2)}(s)$ is the second derivative of the cumulant generating function. Note that the third cumulant, κ_{3i}, estimates the bias. Justification for this formula is outlined by McCullough and Nelder (1989) [13] in section 15 of *Generalized Linear Models*. Specifically, assume a binomial logit model has a response variable $Y_i \sim BIN(m_i, \pi_i)$, $i = 1, \ldots, n$ where m_i, $i = 1, \ldots, n$ are assumed known and the Y_i variables are independent. Notice that in this scenario $k = n$. Thus, the moment generating function is given by $M_i(s) = (\pi_i e^s + (1 - \pi_i))^{m_i}$. Moreover, the cumulant generating function would be $K_i(s) = m_i log(\pi_i e^s + (1 - \pi_i))$. Thus, the first, second, and third derivatives of the cumulant generating function are given by

$$K_i'(s) = \frac{m_i \pi_i e^s}{\pi_i e^s + (1 - \pi_i)},$$

$$K_i^{(2)}(s) = \frac{m_i \pi_i e^s}{\pi_i e^s + (1 - \pi_i)} - \frac{m_i \pi_i^2 e^{2s}}{(\pi_i e^s + (1 - \pi_i))^2},$$

and

$$K_i^{(3)}(s) = \frac{m_i \pi_i e^s}{\pi_i e^s + (1 - \pi_i)} - \frac{3 m_i \pi_i^2 e^{2s}}{(\pi_i e^s + (1 - \pi_i))^2} + \frac{m_i \pi_i^3 e^{3s}}{(\pi_i e^s + (1 - \pi_i))^3},$$

respectively. Consequently, the second and third cumulants are $\kappa_{2i} = m_i \pi_i (1 - \pi_i)$ and $\kappa_{3i} = m_i \pi_i (1 - \pi_i)(1 - 2\pi_i)$. Additionally, the likelihood function is given by $L(\pi_i) = \pi_i^{y_i} (1 - \pi_i)^{m_i - y_i}$, $i = 1, \ldots, n$. Thus, the usual score function, the first derivative of the log likelihood is given by $U_r(\boldsymbol{\pi}) = \sum_{i=1}^{n} (y_i - m_i \pi_i) x_{ir}$, $r = 1, \ldots, n$.

Therefore, the penalized score function is

$$U_r^*(\boldsymbol{\pi}) = \sum_{i=1}^{n}(y_i - m_i\pi_i)x_{ir} + \frac{1}{2\phi}\sum_{i=1}^{n}(1-2\pi_i)h_i x_{ir}, \quad r = 1, \ldots, n. \tag{3.24}$$

In the case of a single binomial trial, the dispersion parameter is $\phi = 1/1 = 1$ (McCullough and Nelder (1989) [13]). Thus, (3.24) simplifies to,

$$U_r^*(\boldsymbol{\pi}) = \sum_{i=1}^{n}\{(y_i + \frac{h_i}{2}) - (h_i + m_i)\pi_i\}x_{ir}$$

for $r = 1, \ldots, n$.

Note that in models with no bias present in the MLE, such as a simple linear regression model, the score function will not be penalized as the third cumulant will be zero.

Now we may apply the tube-formula confidence bounds to the penalized MLEs, rather than the MLEs. All calculations will follow the previous SCR interval descriptions since the information matrix of the pMLEs is the usual information matrix of the MLEs (See Firth (1993) [12]). Consequently no alteration of the methodology is necessary. We are simply replacing $\hat{\boldsymbol{\beta}}$ with the pMLE of $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\beta}}^*$. We do not need to consider the first two SCR methods as they were adjusting for the bias. Rather, we will simply consider the critical value d_{TUBE} applied to the pMLE $\hat{\boldsymbol{\beta}}^*$. Additionally, the centered SCR could be applied with possible improvement as this interval is re-centered and re-scaled. The CS bounds can also be calculated utilizing $\hat{\boldsymbol{\beta}}^*$ instead of $\hat{\boldsymbol{\beta}}$. We refer to this as the pCS bounds in the sequel. No other modifications will be necessary.

We will refer to our SCR bounds utilizing the pMLEs as bias prevented SCRs, or pSCRs. The first of these is given by

$$(g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}^* - d_{TUBE}\hat{\sigma}(\boldsymbol{x}_i)), g^{-1}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}^* + d_{TUBE}\hat{\sigma}(\boldsymbol{x}_i))), \quad i = 1, \ldots, n \tag{3.25}$$

where β^* is a penalized maximum likelihood estimate, $\hat{\sigma}(\boldsymbol{x}_i)$ is given by (2.6), and d_{TUBE} is obtained as described in section 3.1.1. The Casella-Strawderman results could also be applied to an interval of the form (3.25) with d_{TUBE} replaced by d_{CS}, where d_{CS} is obtained as described in section 3.1.1. Finally, the centered SCR may be utilized to yield the second pSCR. This interval is of the form

$$\left(g^{-1}((\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^* - d_{pSCR2}), g^{-1}((\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^* + d_{pSCR2})\right), \quad i = 1, \ldots, n \tag{3.26}$$

for $(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^* = \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}^* - \hat{\kappa}_1^*(\boldsymbol{x}_i)\sqrt{\boldsymbol{x}_i'\hat{\boldsymbol{V}}\boldsymbol{x}_i}$ and $d_{pSCR2} = d_{TUBE}\sqrt{\boldsymbol{x}_i'\hat{\boldsymbol{V}}\boldsymbol{x}_i\hat{\kappa}_2^*(\boldsymbol{x}_i)}$ where $\hat{\kappa}_1^*(\boldsymbol{x}_i)$ and $\hat{\kappa}_2^*(\boldsymbol{x}_i)$ are now based on the estimator $\hat{\boldsymbol{\beta}}^*$. The formulas for these moments of the Gaussian field are given in Appendix D.

Since $\hat{\boldsymbol{\beta}}^*$ attempts to eliminate the bias, it is reasonable to expect that the confidence regions based on this estimator will attain the desired level of confidence for smaller sample sizes than the SCR bounds of SLM. The pSCR bounds for moderate to large samples should be very similar to SLM's corrected and centered SCR bounds.

3.2 Bounds for Simultaneously Estimating Functions of Model Parameters

Often an estimate of something other than the expected response, whether a probability or a mean, is desired. In previous sections, quantities such as the odds ratio, relative risk, and attributable proportion were discussed. Odds ratios or relative risks have immediate clinical application and are often easier for non-statisticians to understand than expected responses. As discussed previously, all of these quantities may be estimated directly from the data. However, it is preferable at times to estimate these quantities via generalized linear models. Specifically, these quantities may all be expressed as functions of the parameters of GLMs. Consequently, it is possible to utilize all of the methodologies discussed in section 3.1 to simultaneously estimate quantities such as the odds ratio or relative risk. In this section I propose methods that utilize these simultaneous bounds for GLMs to estimate quantities such as odds ratios, thereby accounting for multiplicity of inference. Since the odds ratio, relative risk, and attributable proportion were described previously, I will not review these quantities. Rather we begin by reviewing some relevant methods for simultaneous estimation.

3.2.1 Previous Methods

When quantities such as the odds ratios or relative risks are of interest, we are often utilizing discrete random variables to predict a binary response variable. While it is possible to have continuous predictors and compute quantities such as the odds ratio, the use and application of the odds ratio in these situations is less obvious and the need for multiplicity is less apparent. Consequently, attention will first focus on the case of categorical predictors.

Researchers have previously explored simultaneous estimation of various sets of the parameters. For example, in 1996, McCann and Edwards [14] proposed a procedure to simultaneously estimate p contrasts of k unknown parameters. This method utilizes Naiman's Inequality, discussed previously, to obtain conservative simultaneous confidence regions for the p contrasts of interest. These new bounds outperform the existing competing conservative bounds for many scenarios. However, the McCann-Edwards (ME) method applied only to linear models in general. I propose adapting the SCR and related bounds from section 3.1 to simultaneously bound p contrasts of k unknown parameters from generalized linear models in an analogous manner. First, I will review the ME method for linear models, and then detail the proposed method for GLMs.

Assume we have a regression model of the form,

$$\boldsymbol{Y} = \boldsymbol{X}'\boldsymbol{\theta}$$

where $\boldsymbol{Y}$ is a $n \times 1$ vector of responses, $\boldsymbol{X}$ is a $k \times n$ matrix of predictor variables, and $\boldsymbol{\theta}$ is a $k \times 1$ vector of regression parameters as described by (2.7). Assume that the MLE for $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}$, is asymptotically multivariate normal with mean $\boldsymbol{\theta}$ and covariance matrix $\sigma^2_{LM}\boldsymbol{F}$ with $\boldsymbol{F}$ assumed known and full rank. Also assume that an estimate for σ^2_{LM} exists and is given by $\hat{\sigma}^2_{LM}$ where $\hat{\sigma}^2_{LM}$ is independent of $\hat{\boldsymbol{\theta}}$ and is such that $\frac{\nu\hat{\sigma}^2_{LM}}{\sigma^2_{LM}} \sim \chi^2_\nu$. Thus, we have k unknown parameters to estimate via the usual MLE, $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_k)$. Now suppose p linear combinations of the regression parameters are of interest. Let $\boldsymbol{C}$ be a $p \times k$ matrix of constants such that $\boldsymbol{C\theta}$ is a vector of these linear combinations of $\boldsymbol{\theta}$. Now, given the distributional assumptions on $\hat{\boldsymbol{\theta}}$, $\boldsymbol{C}\hat{\boldsymbol{\theta}}$ is multivariate normal with mean $\boldsymbol{C\theta}$ and covariance matrix $\sigma^2_{LM}\boldsymbol{CFC'}$. Thus, if a single contrast is given by $\boldsymbol{c}'_j\boldsymbol{\theta}$ where $\boldsymbol{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_p)'$ and each $\boldsymbol{c}'_j = (c_{j1}, \ldots, c_{jk})$, then we can form exact simultaneous interval estimates for each contrast via the following formula:

$$\boldsymbol{c}'_j\hat{\boldsymbol{\theta}} \pm d\hat{\sigma}_j,\ j = 1, \ldots, p \tag{3.27}$$

where $\hat{\sigma}_j = \hat{\sigma}_{LM}(\boldsymbol{c}'_j\boldsymbol{F}\boldsymbol{c}_j)^{1/2}$ and d is a p-dimensional multivariate t quantile with ν degrees of freedom and correlation matrix $\boldsymbol{R}$, with $\boldsymbol{R}$ the correlation matrix corresponding to $\boldsymbol{CFC'}$. The path length inequality proposed by McCann and Edwards provides a conservative solution for d that outperforms the existing conservative solutions in many cases. Note that their solution only provides conservative intervals, as obtaining the multivariate-t quantile providing an exact interval is generally an intractable problem. The following theorem given by McCann and Edwards (1996) in [14] details this solution.

Theorem 3.1 *Let $\boldsymbol{T}$ have a p-dimensional multivariate-t distribution with degrees of freedom ν and underlying correlation matrix $\boldsymbol{R}$ of rank r. The probability*

$$P(|T_j| \leq d,\ j = 1, \ldots, p)$$

is bounded below by the expression

$$1 - \int_0^{1/d} min(F_{r-2,2}[(s((dt)^{-2} - 1))/(r-2)] \times (\Lambda/\pi) + F_{r-1,1}[((dt)^{-2} - 1)/(r-1)], 1) f_T(t)dt,$$

with

$$\Lambda = \sum_{j=1}^{p-1} cos^{-1}(|r_{j,j+1}|),$$

where $F_{m,n}$ is the distribution function of an F random variable with m and n degrees of freedom and f_T is the density function of a random variable T such that $rT^2 \sim F_{\nu,r}$. If d is such that the foregoing expression is at least $1-\alpha$, then the intervals (3.27) will be conservative simultaneous $(1-\alpha)100\%$ confidence intervals.

This inequality determines a value of d that depends on the correlation structure $\boldsymbol{R}$ and the path length Λ. The path length also depends on the ordering of the indices $1, 2, \ldots, p$ and yields the smallest value of d for the optimum ordering. ME recommend estimating the optimum ordering via the nearest neighbor algorithm (Townsend, 1987) as no exact solution exists and this method often provides the optimum ordering. It is noted that as the path length function approaches either infinity or zero the value of d becomes $(rF_{\alpha,r,\nu})^{1/2}$ (Scheffé's critical value) or $t_{\alpha/2,\nu}$ (the one-at-a-time critical value), respectively. Simulations show that when the degrees of freedom are low and the number of comparisons are high, the ME method outperforms other existing conservative solutions. Note that the ME method has simply applied Naiman's inequality to an interval and parameterization carefully chosen to contain the quantities of interest.

However, the ME method is only strictly valid for linear models. I propose adapting the SCR bounds and their counterparts for generalized linear models in an analogous manner in order to make simultaneous inference on linear combinations of the parameters from GLMs, such as odds ratios or relative risks.

3.2.2 Proposed Methods

In order to estimate the linear combinations of the GLM parameters, I propose applying the SCR and PC methodologies in a fashion similar to the ME bounds. Recall that the only requirement for applying the SCR-type bounds is normality of the regression parameter estimates, which is true asymptotically for GLMs. Application of these bounds to various quantities of interest are outlined in detail below.

3.2.2.1 A General Set of Comparisons of the Model Parameters

Consider the general setup and estimators for GLMs presented in Chapter 2. In section 3.1 the SCR bands were applied to GLMs to estimate the expected response function simultaneously over a closed set χ. Now these bounds, along with the pSCR, CS, and pCS bounds, may be applied to simultaneously estimate a set of p linear combinations of the regression parameters from a GLM. These linear combinations will each be of the form $\boldsymbol{c}_j'\boldsymbol{\beta}$ where $\boldsymbol{c}_j$ is a $k \times 1$ vector for $j = 1, \dots, p$. Let $\boldsymbol{C} = (\boldsymbol{c}_1, \dots, \boldsymbol{c}_p)'$ be a $p \times k$ matrix. Thus $\boldsymbol{C\beta}$ is a vector of the p linear combinations of interest. Utilizing the SCR methodologies, we have that the expected response function is simultaneously estimated by bounds with at least $100(1-\alpha)\%$ coverage $\forall \boldsymbol{x}_i \in \mathcal{X}$. As the results detailed in section 3.1 can be applied to simultaneously estimate the expected response function, they can be applied to $\mathcal{X}$ to obtain simultaneous intervals on $g^{-1}(\boldsymbol{c}_j'\boldsymbol{\beta})$ provided $\boldsymbol{c}_j \in \mathcal{X}$ for $j = 1, \dots, p$. This interval may then be transformed to obtain simultaneous bounds on the $\boldsymbol{c}_j'\boldsymbol{\beta}$ via the link g. Generally, the bounds will be of the form

$$\boldsymbol{c}_j'\hat{\boldsymbol{\beta}} \pm d \times \hat{\sigma}_{GLM}(\boldsymbol{c}_j), \quad j = 1, \dots, p \tag{3.28}$$

where $\hat{\sigma}_{GLM}(\boldsymbol{c}_j)$ is given by (2.6). The following theorems detail the use of the aforementioned critical values to obtain $100(1-\alpha)\%$ coverage for a fixed set of p linear combinations of the parameters. Note that we will have eight possible solutions for confidence bands on a fixed set of p linear combinations of the parameters since the usual CS, the pCS, the four SCR, and the two pSCR intervals may all be applied.

Recall the domains, denoted $R_{\boldsymbol{x}_i}$, presented in section 3.1 pertaining to the CS method for linear models. Let $R^*_{\boldsymbol{x}_i}$ be the smallest hyper-rectangle of the CS form that contains the $\boldsymbol{c}_j, \quad \forall \quad j = 1, \ldots, p$. A set of this form will be utilized in the following theorem.

Theorem 3.2 *Under the GLM setting described in (2.1), the asymptotic simultaneous coverage probability of the bands (3.28) has a lower bound of* $1-\alpha$ *for* $d = d_{CS}$ *when* d_{CS} *is computed for* $R_{\boldsymbol{x}_i} = R^*_{\boldsymbol{x}_i}$*. The same holds for the bands (3.28) when* $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^*$*, the pMLE of* $\boldsymbol{\beta}$*.*

Proof: Note that this result holds for any $\boldsymbol{x}_i \in \Omega_{\boldsymbol{x}_i}$ and that $\Omega_{\boldsymbol{x}_i} \supset R_{\boldsymbol{x}_i} = R^*_{\boldsymbol{x}_i}$ where the vectors $\boldsymbol{c}_j = (c_{j1}, \ldots, c_{jp})$, $j = 1, \ldots, p$, are embedded in the hyper-rectangle $R^*_{\boldsymbol{x}_i}$. Thus $\boldsymbol{c}_j, \quad j = 1, \ldots, p$ is contained in $\Omega_{\boldsymbol{x}_i}$ and consequently, utilizing $d = d_{CS}$ in (3.28) guarantees at least $100(1-\alpha)\%$ simultaneous coverage asymptotically for the p intervals of interest. Also note that the limiting distributions of $\hat{\boldsymbol{\beta}}^*$ and $\hat{\boldsymbol{\beta}}$ are identical. Thus the asymptotic coverage of the bands (3.28) based on $\hat{\boldsymbol{\beta}}^*$ is the same as those based on $\hat{\boldsymbol{\beta}}$. $\square$

Now let $\mathcal{X}^*$ be the smallest compact subset of the domain where $\boldsymbol{c}_j \in \mathcal{X}^*, \quad \forall \quad j = 1, \ldots, p$.

Theorem 3.3 *Under the GLM setting described in (2.1), the asymptotic simultaneous coverage probability of the bands (3.28) has a lower bound of* $1-\alpha$ *for* $d = d_{TUBE}$*,* $d = d_{SCR1}$ *and* $d = d_{SCR2}$ *where these critical values are computed for* $\mathcal{X} = \mathcal{X}^*$*. For* $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^*$*, the pMLE of* $\boldsymbol{\beta}$*, the same holds for* $d = d_{TUBE}$*.*

Proof: Note that this result holds for any $\boldsymbol{x}_i \in \mathcal{X} = \mathcal{X}^*$. The vectors $\boldsymbol{c}_j = (c_{j1}, \ldots, c_{jk})$, $j = 1, \ldots, p$, are embedded in the set $\mathcal{X}^*$. Thus, $\boldsymbol{c}_j \in \mathcal{X} = \mathcal{X}^* \quad \forall j$ and consequently, utilizing $d = d_{TUBE}$, $d = d_{SCR1}$ or d_{SCR2} in (3.28) guarantees at least $100(1-\alpha)\%$ simultaneous coverage asymptotically for the intervals. Moreover, since the limiting distributions of $\hat{\boldsymbol{\beta}}^*$ and $\hat{\boldsymbol{\beta}}$ are identical, then the asymptotic coverage of the bands (3.28) based on $\hat{\boldsymbol{\beta}}^*$ is the same as those based on $\hat{\boldsymbol{\beta}}$. $\square$

Theorem 3.4 *Under the GLM setting described in (2.1), the asymptotic simultaneous coverage probability of the band*

$$\boldsymbol{c}_j'\hat{\boldsymbol{\beta}} - \hat{\kappa}_1(\boldsymbol{c}_i)\hat{\sigma}_{GLM}(\boldsymbol{c}_i) \pm d_{TUBE}\hat{\sigma}_{GLM}(\boldsymbol{c}_i)\sqrt{\hat{\kappa}_2(\boldsymbol{c}_i)} \tag{3.29}$$

where $\hat{\sigma}_{GLM}(\boldsymbol{c}_j')$ is given by (2.6), has a lower bound of $1-\alpha$ with $\hat{\kappa}_1(\boldsymbol{c}_j)$ and $\hat{\kappa}_2(\boldsymbol{c}_j)$ defined as stated in section 3.1 and $\mathcal{X} = \mathcal{X}^*$. The same holds for $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^*$, the pMLE of $\boldsymbol{\beta}$ with $\hat{\kappa}_1(\boldsymbol{c}_j)$ and $\hat{\kappa}_2(\boldsymbol{c}_j)$ defined appropriately for $\hat{\boldsymbol{\beta}}^*$ and $\mathcal{X}$ again equal to $\mathcal{X}^*$.

Proof: Note that this result holds for any $\boldsymbol{x}_i \in \mathcal{X}^*$. The vectors $\boldsymbol{c}_j = (c_{j1}, \ldots, c_{jk})$, $j = 1, \ldots, p$, are embedded in the set $\mathcal{X}^*$. Thus, $\boldsymbol{c}_j \in \mathcal{X} = \mathcal{X}^* \quad \forall j$ and consequently the intervals in (3.29) utilizing $\hat{\boldsymbol{\beta}}$, the usual MLE for $\boldsymbol{\beta}$, guarantee at least $100(1-\alpha)\%$ simultaneous coverage asymptotically for the intervals. Since the limiting distributions of $\hat{\boldsymbol{\beta}}^*$ and $\hat{\boldsymbol{\beta}}$ are identical, then the asymptotic coverage of the bands (3.28) based on $\hat{\boldsymbol{\beta}}^*$ is the same as those based on $\hat{\boldsymbol{\beta}}$. □

3.2.2.2 Illustrations of Simultaneous Procedures for Particular Scenarios

The simultaneous procedures for estimating any specified combination of the model parameters from a GLM include the following: SCR (all four forms), PC (restricted-Scheffé), pPC (pMLE restricted-Scheffé), and both pSCR bounds. The relevant bounding procedures will be demonstrated for the odds ratio, relative risk, and attributable proportion, but note that other quantities of interest could also be estimated.

Since we have assumed the model estimated is a GLM, it is possible to apply any of the proposed eight confidence bounds to the general expected response function. Following the procedure described in section 3.1, one could find the bounds for any GLM on a restricted domain and then apply the link function. For example, a GLM is generally given by

$$g(E(Y_i|\boldsymbol{x}_i)) = \boldsymbol{x}_i'\boldsymbol{\beta}, \quad i = 1, \ldots, n.$$

We will outline the procedure for some common link functions, *log* and *logit*. Note that it is possible to simultaneously estimate any specified set of linear combination of the regression parameters for a specified GLM using any of the eight methods described in previously in this chapter.

As an example, recall the preterm birth study discussed in Chapter 1. This study employed a loglinear model where the reference level was the case with no apparent source of maternal stress. Recall the model was given by (2.14), thus let $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)$ be the $(k+1) \times 1$ parameter vector, for this example $k = 3$. In this study, the relative risks for each level of the source of maternal stress compared back to the cases with no identified maternal stress factor were of primary interest and, as discussed previously, the overall conclusions made in the study merits simultaneous estimation of these relative risks. We focused specifically on the variable indicating "Life Event" stress (see Table 1.1). This factor had four overall levels for the independent variable indicating the presence of any "Life Event" maternal stress. To utilize the procedure outlined in section 3.2.2.1, we need to define the matrix $\boldsymbol{C}_{3\times 4} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_3)'$. Specifically, let $\boldsymbol{c}_j = (c_{j1}, c_{j2}, \ldots, c_{j4})$, where $c_{j,i} = 0$ for $i \neq j+1$ and $c_{j,j+1} = 1$, $i = 1, \ldots, 4$; $j = 1, \ldots, 3$. Since β_j is the log of the relative risk for the j^{th} level of the independent variable, then $\boldsymbol{C\beta} = (\beta_1, \beta_2, \beta_3)'$, is the vector of log relative risks for "Life Event" stress in the preterm birth study. The $\boldsymbol{C}$ matrix that yields $\boldsymbol{C\beta}$ equal to the log relative risks for other reference-coded Poisson models would be defined similarly. Notice that in our example $\mathcal{X}$ will be the three dimensional subspace where $x_1 = 0$ and $\sum_{i=2}^{4} x_i \leq 1$. (Note that $\boldsymbol{c}_j$, $j = 1, \ldots, 3$, is contained in this $\mathcal{X}$.)

Asymptotic simultaneous $100(1-\alpha)\%$ confidence bands for the log relative risks of interest in the preterm birth study can now be formed by utilizing (3.28) or (3.29) with an appropriate critical value and $\hat{\boldsymbol{\beta}}$ or $\hat{\boldsymbol{\beta}}^*$ as warranted. To obtain asymptotic simultaneous confidence bands on the relative risks, the bands on the log relative risks would be exponentiated. These intervals will allow us to make overall conclusions with a specified asymptotic error rate.

Alternatively, now suppose relative risks are of interest using a slightly different model. Specifically, model-based relative risks can be estimated utilizing a Poisson regression on a proportion. Thus, assume a simple model of the form,

$$log(\pi(\boldsymbol{x}_i)) = \beta_0 + \boldsymbol{x}_i'\boldsymbol{\beta} = \boldsymbol{x}_i^{*'}\boldsymbol{\beta}^*$$

for $i = 1, \ldots, n$ where $\boldsymbol{\beta}^*_{(k+1)\times 1} = (\beta_0, \beta_1, \ldots, \beta_k)'$ and $\boldsymbol{x}_i^* = (1, x_{i1}, \ldots, x_{ik})'$ is a predictor vector of dimension $k+1$. Then the relative risk is easily estimated by exponentiating any one of the regression parameters, e^{β_i} $(i = 1, \ldots, k)$. Thus, in order to estimate the i^{th} relative risk, let the matrix $\boldsymbol{C}$ be defined as described in 5.2.1 with each $c_{ij} = 0$ when $i \neq j+1$ and $c_{j,j+1} = 1$, $i = 1, \ldots, k+1$; $j = 1, \ldots, p$. Again we can obtain asymptotic simultaneous $100(1-\alpha)\%$ confidence bands for the relative risks by utilizing (2.12) or (3.29) with an appropriate critical value and $\hat{\boldsymbol{\beta}}$ or $\hat{\boldsymbol{\beta}}^*$ as warranted. Here $\mathcal{X}$ would be similar to that for the skin cancer study. To complete the calculations, the resulting bounds would be exponentiated, as they were for the odds ratio calculations.

Once relative risks are estimated from a Poisson regression model, it may be helpful to additionally estimate the attributable proportions. Recall that the point estimate of a relative risk is given by $\gamma_i = exp(\boldsymbol{c}_j'\boldsymbol{\beta})$ for $j = 1, \ldots, p$. As the attributable proportion is a one-to-one increasing function of the relative risk, the following holds, $\kappa_i = 1 - \frac{1}{\gamma_i} = \frac{\gamma_i - 1}{\gamma_i}$, $i = 1, \ldots, n$ where κ is the attributable proportion and γ is the relative risk. Suppose the relative risk interval has lower and upper limits denoted L_{RR} and U_{RR}, respectively. Then the limits on the attributable proportion are given by

$$\left(\frac{L_{RR} - 1}{L_{RR}}, \frac{U_{RR} - 1}{U_{RR}} \right). \tag{3.30}$$

If (L_{RR}, U_{RR}) were obtained with simultaneous coverage for a specified set, then the same simultaneous coverage properties will hold for the equivalent set of attributable proportions.

Note that all the previous examples assumed the estimated quantities should refer back to the reference level. However, this general methodology can be extended to make other kinds of comparisons between the estimated quantities. For example, if we consider a logistic model with reference coding, we could consider an alternative set of odds ratios for joint estimation. Recall that every e^{β_i} is the odds ratio for the i^{th} level compared to the reference level (or control). Alternatively, suppose it was of interest to estimate the odds ratio comparing the first nonreference level of the covariate to every other nonreference level. Then these odds ratios could be estimated via $e^{\beta_1 - \beta_j}$ for $j = 2, \ldots, k$. Thus, the contrast matrix, $\boldsymbol{C}$, would have rows that appear something like,

$$\boldsymbol{c}_j = (0, 1, 0, \ldots, 0, -1, 0, \ldots, 0).$$

Then via (3.28) or (3.29) we could estimate the log odds ratios simultaneously for an appropriate d and $\hat{\boldsymbol{\beta}}$ or $\hat{\boldsymbol{\beta}}^*$ as warranted. Simply exponentiating the results would give simultaneous bands for this particular set of odds ratios.

CHAPTER 4

Simulations

In order to evaluate the performance of the proposed restricted Scheffé bounds, pMLE Scheffé bounds, and the pMLE SCR bounds, I conducted Monte Carlo simulations. I ran two main simulation studies; simulations for the simultaneous estimation of the expected response function and simulations for the simultaneous estimation of a linear function of the regression parameters which easily gives simultaneous intervals on the odds ratios, relative risks, or attributable proportions via transformation. Some recommendations are provided on the choice of intervals for various scenarios. Additionally, an example of this kind of transformation is given in section 5.2.

4.1 Expected Response Function Simulations

These simulations assume only one predictor variable so that the vector of parameter estimates is of the general form $\boldsymbol{\beta} = (\beta_0, \beta_1)'$. The estimated coverage, or alternatively, the estimated error was recorded for each scenario simulated. I simulated scenarios for various values of the parameter $\boldsymbol{\beta}$, the sample size, n, and the predictor variable, x. Simulations focused on the most commonly utilized GLMs, logistic regression and Poisson models, although other models could be investigated. Simulations included the following scenarios. Regression parameters as follows: (1) $\boldsymbol{\beta} = (-1, -0.5)$, (2) $\boldsymbol{\beta} = (0, 1)$, (3) $\boldsymbol{\beta} = (2, 4)$, and (4) $\boldsymbol{\beta} = (-0.5, -1)$. The sample sizes were n=10, 25, and 50. The domain, $\mathcal{X}$, for both logit and Poisson models is continuous. For the logit model, I generated points equally spaced over wide and narrow intervals. First appropriate values of π were chosen that would determine either a wide or narrow range for the domain. For a wide domain π_L=0.1 and π_U=0.9 and for a narrow domain π_L=0.25 and π_U=0.75. The $\boldsymbol{X}$ interval endpoints were then determined by inverting the GLM so that $x = (logit(\pi) - \beta_0)/\beta_1$ where β_0 and β_1 are the parameters. Thus, for the logit model, $x_L = (logit(\pi_L) - \beta_0)/\beta_1$ and $x_U = (logit(\pi_U) - \beta_0)/\beta_1$ are the endpoints of the domain. Then let $x_1 = x_L$ and $x_n = x_U$ and let the remaining points be equally spaced between x_1 and x_n. Thus, $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$ is a vector of a countable set of points contained in $\mathcal{X} = \{x : x_L \leq x \leq x_U\}$. In order to simulate the response, a set of Y_i, $i = 1, \ldots, n$ response variables was generated in the following manner. First, we generated Uniform(0,1) random variables, U_i, $i = 1, \ldots, n$. Then we let the response variable Y_i be 1 if $U_i < \pi(\boldsymbol{x}_i)$ and 0 otherwise when generating data for a logit model. Note that $\pi(\boldsymbol{x}_i) = \frac{1}{1+e^{\beta_0+\beta_1\boldsymbol{x}_i}}$ for $i = 1, \ldots, n$. For the Poisson model, I generated a set of uniform random variables for the $\boldsymbol{X}$ values. The wide domain was distributed Uniform(0,1), while the narrow domain was distributed Uniform(0,0.5). To generate the response variable for a Poisson regression model, Y_i is a Poisson random variable generated to have mean $\mu(\boldsymbol{x}_i) = e^{\beta_0+\beta_1\boldsymbol{x}_i}$ for each $i = 1, \ldots, n$. For both models, I then used this set of Y_i values and the $\boldsymbol{X} \in \mathcal{X}$ data set to compute the estimated parameter values and estimated covariance matrix. These were also utilized to obtain the equations for the bounds over $\mathcal{X}$ for each method evaluated. Note that we are not generating binomial, multinomial or Poisson random variables and fitting the model to the generated observations. Rather, we are assuming the model holds exactly. For situations where the logistic and Poisson models do not work well, these methods could perform quite poorly. For each sample I will note whether the estimated bounds cover the true response function $\boldsymbol{X}$, a finite set contained in $\mathcal{X}$. I will then estimate the simultaneous coverage with the empirical coverage of my simulated samples.

Additionally, in order to determine the number of simulated samples required to estimate the error to within ±0.005 we calculated a lower bound on the number of simulations. If we are willing to tolerate 5% type I error ($\alpha = 0.05$), a lower bound on the number of simulations may be determined via $[(0.05)(0.95)/\rho]^{1/2} < 0.005$ where ρ is the number of simulations [7]. This yields $\rho > 1900$. Thus we ran 5000 simulations to reduce error.

4.1.1 Expected Response Function Simulation Results

The most significant distinction among all the competing intervals with regard to the empirical confidence level, was the estimator used, MLE or pMLE. See Figure 4.1 for a plot comparing two MLE intervals to two pMLE intervals. Generally, the pMLE based intervals achieved the desired level of confidence at all sample sizes. In contrast, the MLE based intervals as a group only reached the desired level of confidence for some cases of n=25 or 50. Clearly, the ability of the pMLE intervals to achieve the desired level of α at any sample size is an improvement over any of the usual MLE intervals. Though we see this improvement in the reliability of the pMLE

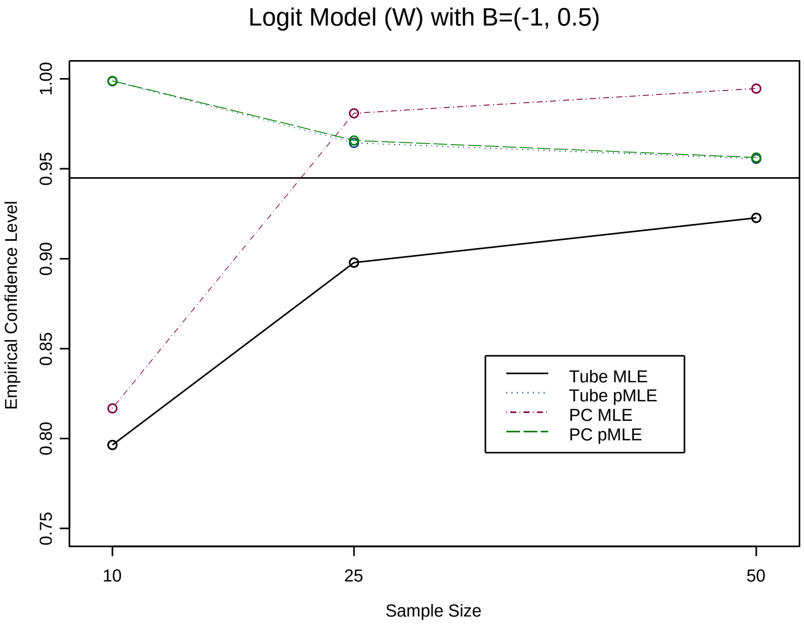

Figure 4.1: MLE and PMLE SCR Intervals with wide range

based intervals, the pMLE based intervals are extremely conservative, particularly at small sample sizes. Again, see Figure 4.1 for an example. This could be due to the computation of the biasing constant used to shift the likelihood equations for solving for the pMLE. At the larger sample sizes the pMLE bias adjustment is more precise while at small sample sizes this adjustment is more conservative.

Recall that we applied the pMLE estimator to the restricted-Scheffé, naive SCR, and SCR3 intervals. Interestingly, the reshifted and rescaled SCR interval (SCR3) does not perform as well as the naive tube-based SCR interval or any of the other intervals. See Figure 4.2 for an example. Similar behavior was observed to Figure 4.2 for other parameter sets. Clearly, when using a bias-preventing estimator, trying to correct the remaining bias is not helpful, and in fact is often detrimental. Though the pMLE SCR3 intervals are not usually a good idea as an alternative to any MLE based intervals, the other pMLE intervals (restricted-Scheffé and naive SCR) perform far better than the MLE intervals in all cases. See both Figure 4.1 and Figure 4.2 for

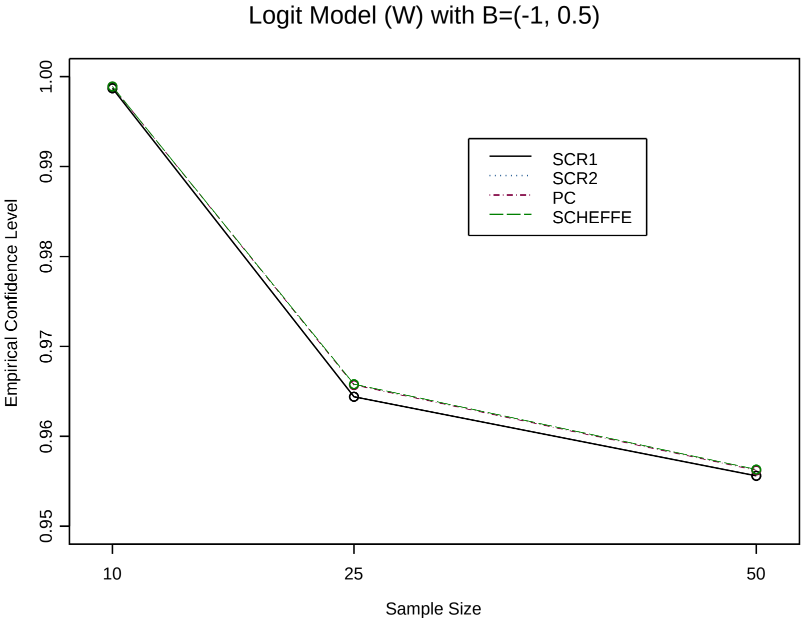

Figure 4.2: pMLE Intervals Logit-Wide Range B=1

examples.

As the sample size increases to moderately large sizes (n=25, 50 or 100), the MLE based intervals as a whole do reach the desired level of confidence and the pMLE based estimators's empirical confidence level decreases to a level much closer to the intended level of confidence (see Figure 4.1), and in many cases the pMLE based intervals (either PC or naive SCR) are actually less conservative than the usual MLE based intervals. While the pMLE based intervals were intended to address poor coverage at small sample sizes, these intervals also appear to improve the conservative nature of the usual MLE based intervals at the moderate sample sizes. Thus, at these moderate to large sample sizes the pMLE based intervals still attain the desired level of confidence but, in general, do not over-reach the desired confidence level as the usual MLE based intervals often do.

Note that so far I have presented only one parameter case for the logit model. Recall that I investigated four sets of parameters for the logit model and three sets of parameters for the Poisson model. The plots comparing the two superior pMLE intervals to the corresponding MLE intervals are displayed in Figures 4.3 to 4.15 in this section. Generally, for the logit model simulations, we see very similar behavior to the plots analyzed previously. However, for some choices of the regression parameters, a sample size of 100 was required to achieve the desired confidence level with the naive tube MLE interval. See Figures 4.3 to 4.5.

In regard to the domain used in the estimation of the interval, when a wide domain was assumed for estimation of the GLM, the MLE based intervals needed larger sample sizes to attain the desired level of confidence than when the domain width was narrow. This held for the logit model in particular (see Figure 4.3). However, this trend did not translate to the pMLE based intervals in general, where the domain of interest really only affected how conservative the intervals were.

The choice of the four parameter sets did not change the overall trends observed

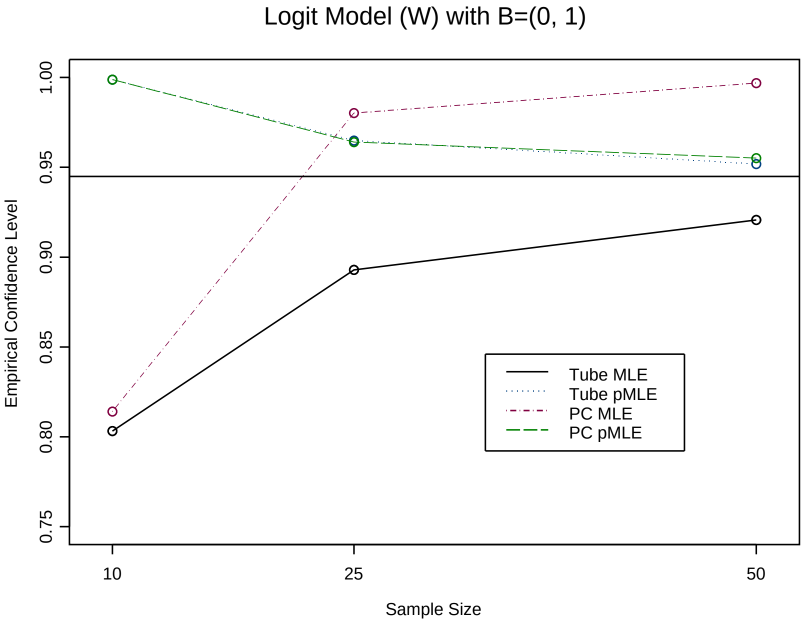

Figure 4.3: MLE and PMLE SCR Intervals with wide range

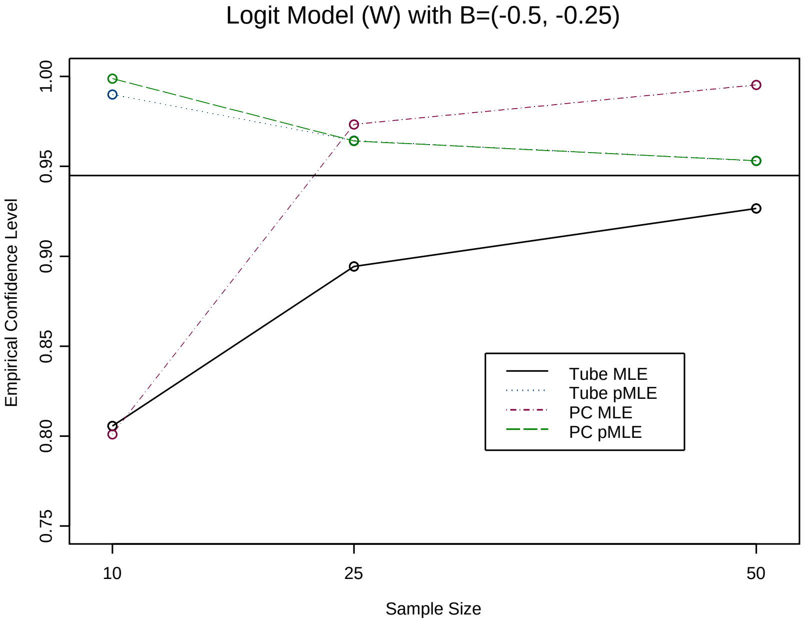

Figure 4.4: MLE and PMLE SCR Intervals with wide range

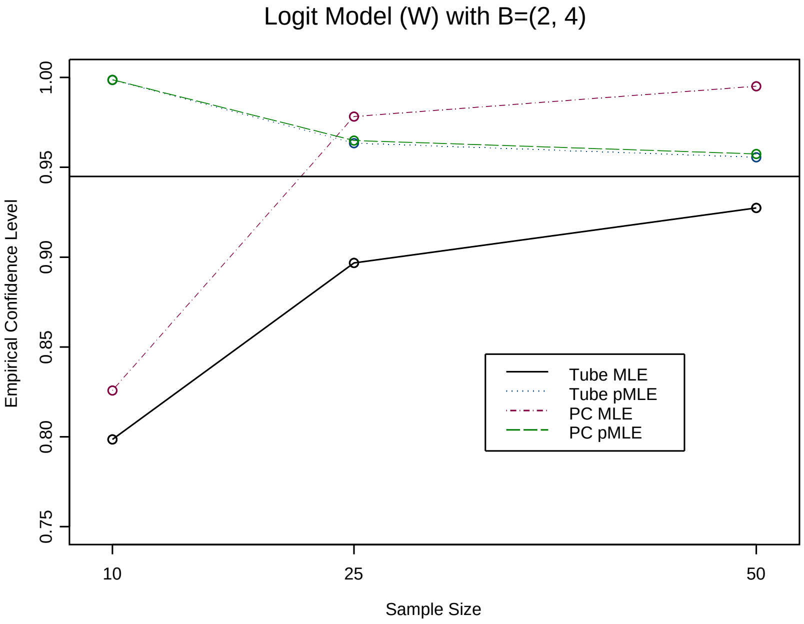

Figure 4.5: MLE and PMLE SCR Intervals with wide range

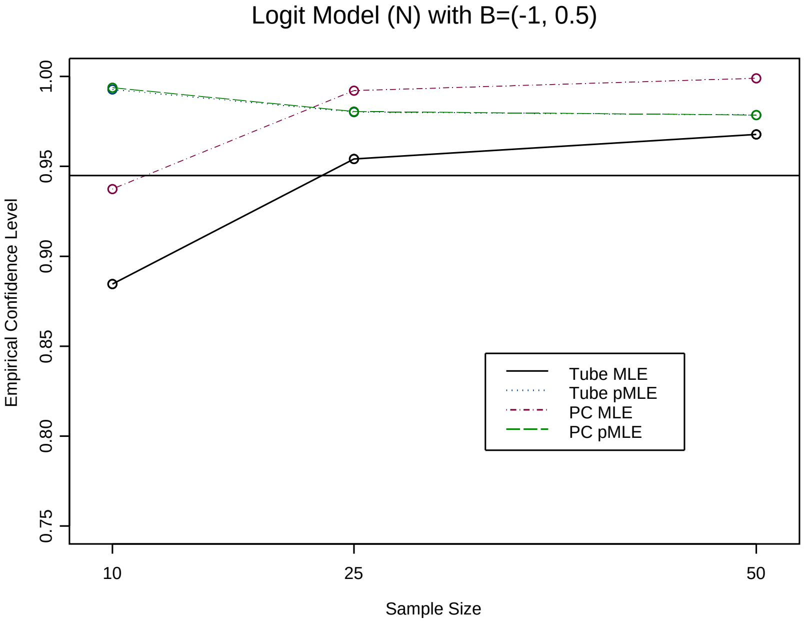

Figure 4.6: MLE and PMLE SCR Intervals with narrow range

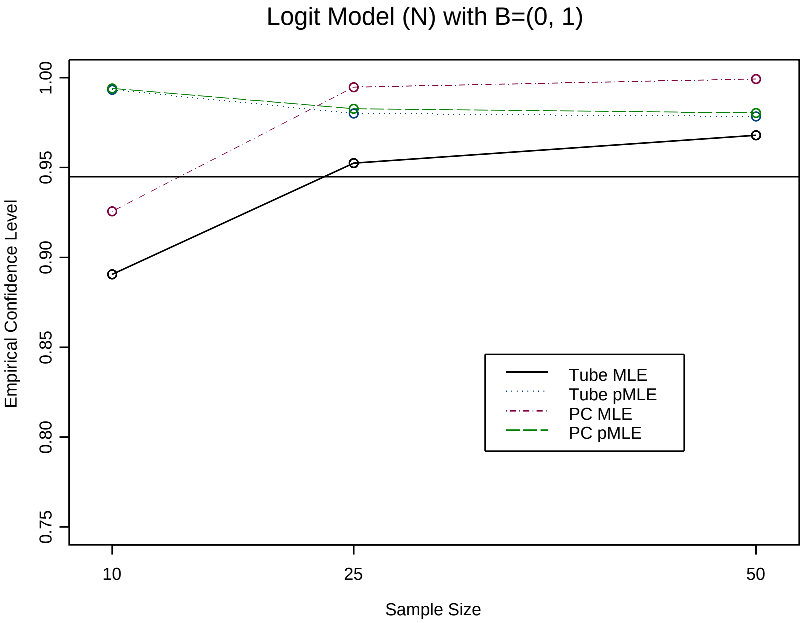

Figure 4.7: MLE and PMLE SCR Intervals with narrow range

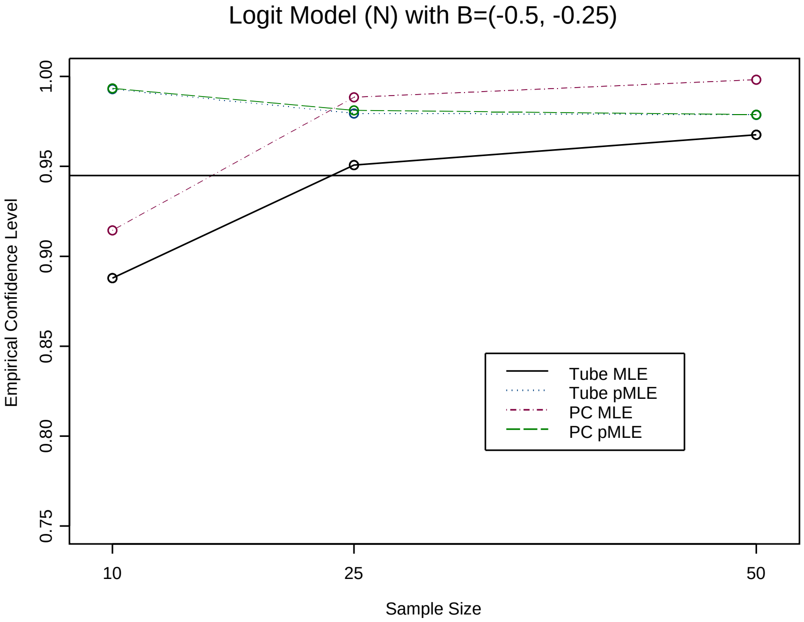

Figure 4.8: MLE and PMLE SCR Intervals with narrow range

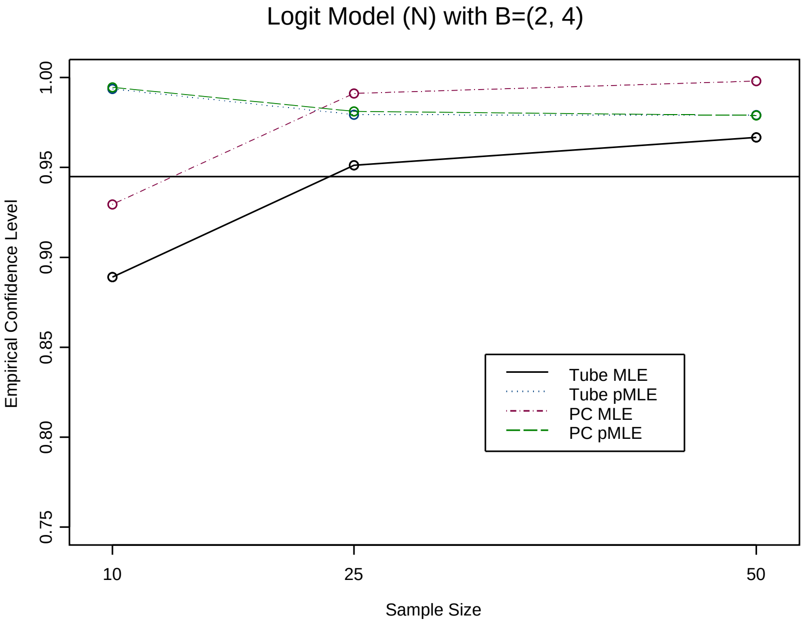

Figure 4.9: MLE and PMLE SCR Intervals with narrow range

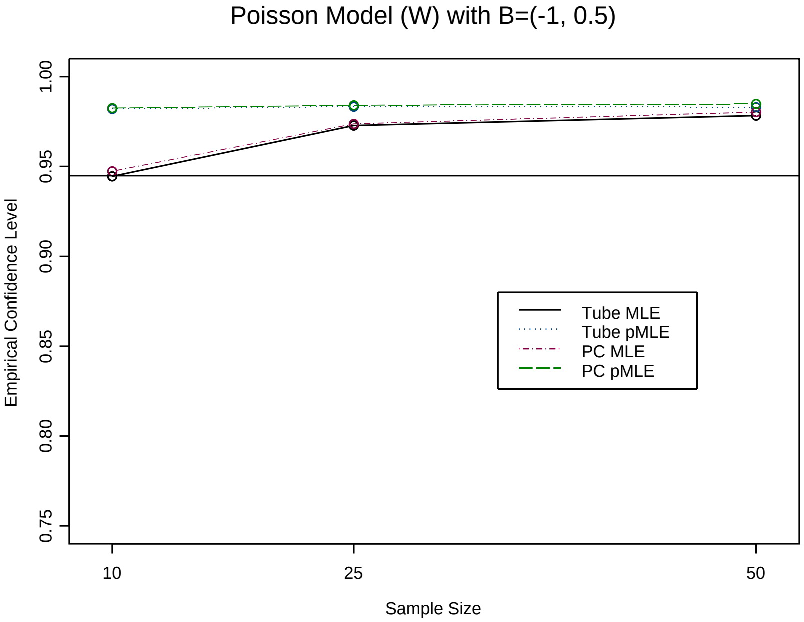

Figure 4.10: MLE and PMLE SCR Intervals with narrow range

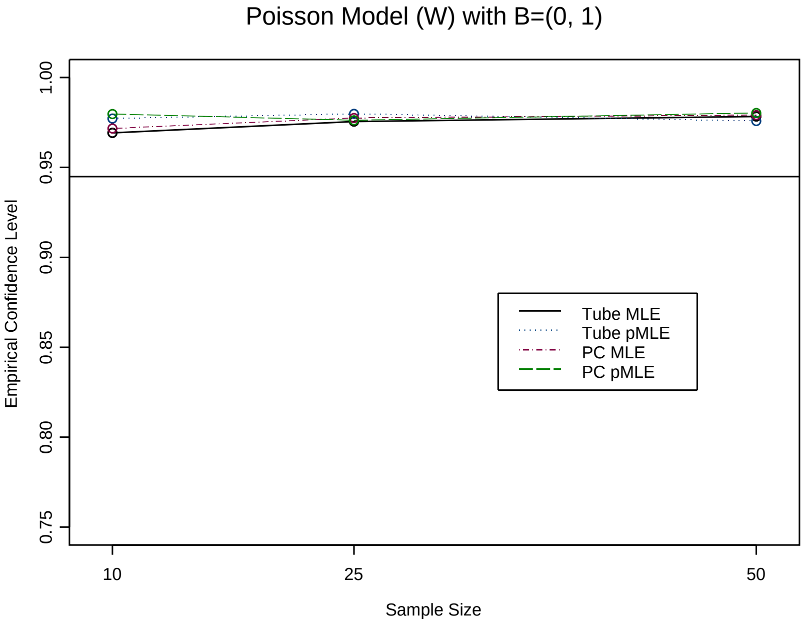

Figure 4.11: MLE and PMLE SCR Intervals with narrow range

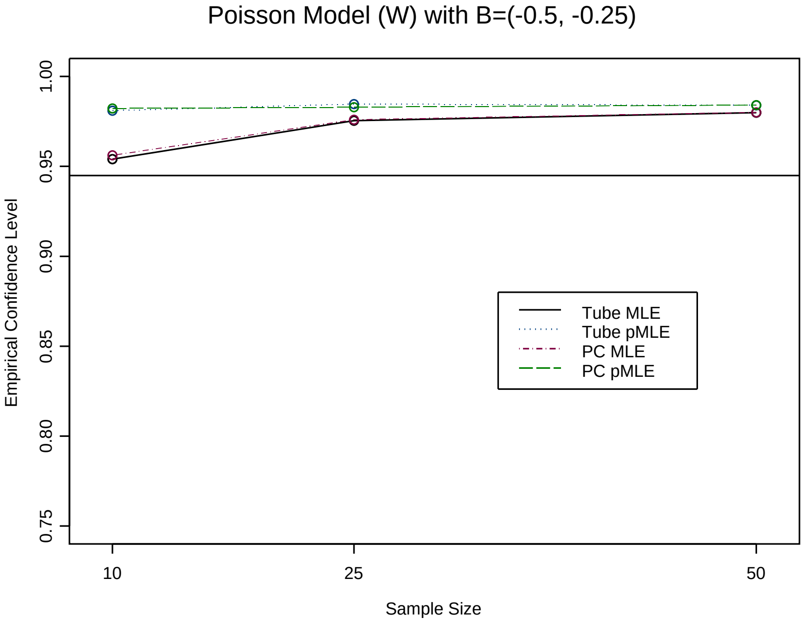

Figure 4.12: MLE and PMLE SCR Intervals with narrow range

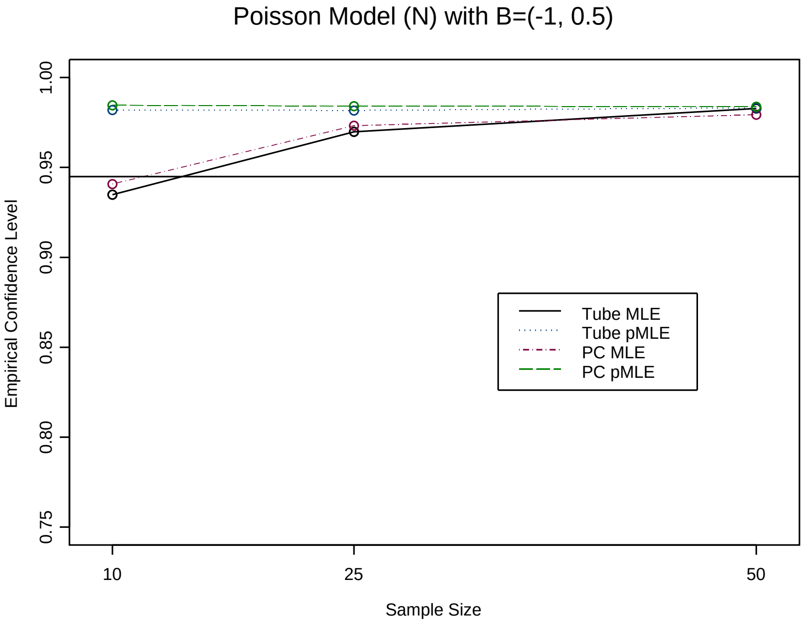

Figure 4.13: MLE and PMLE SCR Intervals with narrow range

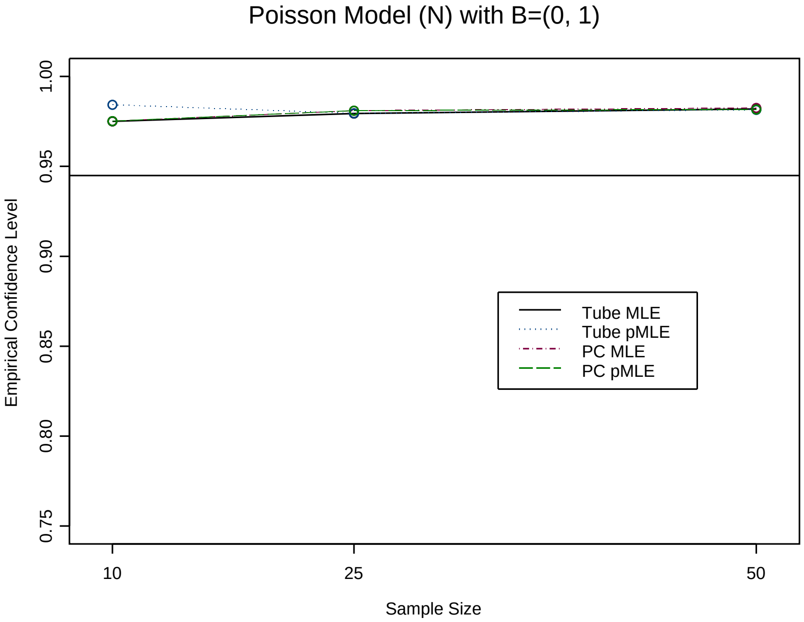

Figure 4.14: MLE and PMLE SCR Intervals with narrow range

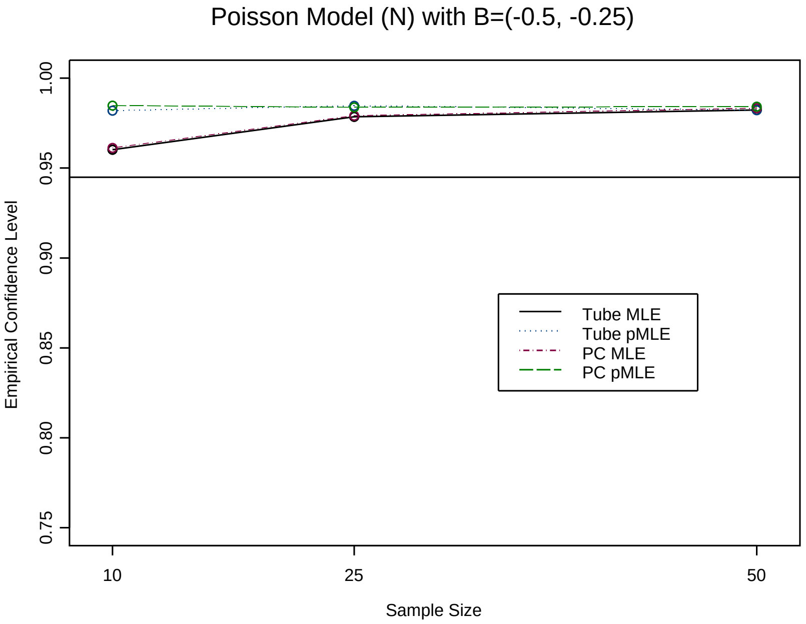

Figure 4.15: MLE and PMLE SCR Intervals with narrow range

for the competing intervals significantly. In contrast, the choice of the link function did. The logistic models were in general less conservative than the Poisson models (compare Figures 4.1 and 4.10). The performance of the MLE and competing pMLE intervals was most distinct when investigating the logistic regression model. Conversely, the Poisson model empirical confidence levels did not vary greatly across the sample sizes when the pMLE based intervals were utilized. Even the MLE based intervals were conservative at times for this link function and attained the desired level of confidence even at the smallest sample sizes for many cases (see Figure 4.10 as an example). Thus, many of the comments made about the logistic regression models do not apply when considering Poisson regression models. In general, little distinction can be made among the competing intervals. At times, the MLE based intervals appear to be a little less conservative than the pMLE based intervals, but often that difference in negligable. I believe that the conservative nature of all the methods for estimating the mean response of a Poisson model needs to be addressed.

This conservative behavior may be due to the tendency of Poisson regression models to overestimate the variance of the regression parameters (see [15] and [16]). A sandwich estimator of the variance should be studied to potentially correct this problem [17]. A sandwich variance estimator, along with generalized estimating equations (GEE's) for estimating the model parameters, can correct a misspecified variance function for Poisson models. The sandwich estimator is a robust variance estimator that consistently estimates the true variance when the parametric model fails to hold. Since the parameters estimated from GEE's have asymptotic normality when utilizing the sandwiched covariance matrix, we may directly apply the usual MLE based interval methods.

Though the MLE intervals are not of primary concern, it should be noted that among the MLE methods, as expected, the Scheffé is the most conservative of all. Yet little distinction can be made among the other methods except that the SCR3 intervals tend to not reach the desired level of confidence as quickly as the other SCR intervals. See Appendix A for examples. However, these trends do not translate to the intervals utilizing the pMLE estimator since: 1) not all SCR intervals are employed using this estimator and thus cannot be compared, and 2) the bias-preventing estimate does drastically change the behavior of each interval overall.

4.2 Functions of the Parameters Simulations

In order to estimate the error associated with the confidence regions for estimating the set of regression parameters, and hence, odds ratios, relative risks or attributable proportions, I will again assume only one predictor variable. In general, the single predictor variable is assumed to be categorical. However, using reference cell coding for one categorical predictor entails utilizing several binary predictor variables. This will be taken into account in the simulations. When evaluating the odds ratio we also need to consider what type of multiple comparisons could be of interest. I will consider only contrasts corresponding to comparisons with a control. For example, comparisons with a control would require simultaneously estimating the odds ratio for each nonreference level with the reference level from a logistic regression model when reference cell coding is utilized. This entails simultaneously estimating all slope parameters. Recall that the control is the reference level, thus if we have $k+1$ regression parameters, there will be k comparisons or k odds ratios to be simultaneously estimated. Thus we are simultaneously estimating e^{β_i} for every $i = 1, \dots, k$. Similarly, the relative risks for each level of a predictor variable with reference to the control could be investigated for a Poisson regression model. The attributable proportion also could be observed for both a Poisson regression model and a logistic regression model. These too are one-to-one functions of the k slope parameters.

We investigated both logistic and Poisson regression models for evaluating the limits on the estimated odds ratios, relative risks, or attributable proportions. In this scenario the generation of the $\mathcal{X}$ data set and Y_i, $i = 1, \dots, n$ was identical to that described in section 4.1. However, in this case we evaluated the estimated coverage of the simultaneous confidence bounds for the discrete set of interest only. This coverage was estimated in an analogous manner to that for the expected response. Namely, the data were generated, the model was estimated, and finally the intervals for the slope parameters were constructed. Each time the interval captured the true parameter value, a success was recorded and the number of captures out of all k comparisons was recorded. This was repeated 5000 times and the average empirical confidence level was recorded. For the purpose of the simulations, I considered $k = 4$, where k is the number of estimated parameters (slope parameters in this special case). The sample sizes considered are n=50, 100, 200, and 300 and $\alpha = 0.05$. Also, as in 4.1, I recorded the empirical confidence level for each method considered.

At times, the estimated covariance matrix was near-singular. This means that the covariance matrix was estimated to be a quantity such that when the calculations for computing the inverse of the matrix were begun, an error occurs in the LU factorization. An error is returned in this case by the Fortran compiler. Additionally, the model is ill-fitting if the response vector is either all 1's or all 0's. Thus, cases where the response vector is all 0 or 1 or the covariance matrix is singular or near-singular were recorded and data were regenerated. When n=50, there were 252 cases that were thrown out and the data regenerated. When n=100, 200, or 300, no cases of near-singular matrices or all 0's or 1's response vectors occurred.

4.2.1 Functions of the Parameters Simulation Results

As with the intervals on the mean response, performance of the intervals on the parameters was most affected by the estimator utilized, MLE or pMLE. In general, when the pMLE estimator was used for any interval method, the desired level of confidence was reached at any sample size (see Figure 4.16 or Figure 4.17).

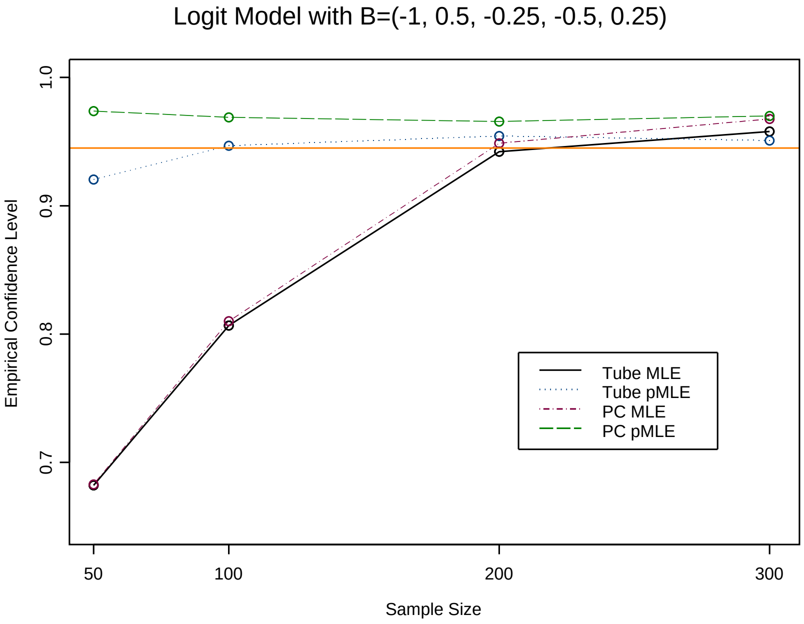

Figure 4.16: MLE and PMLE SCR Intervals for the Parameters

In contrast, the MLE based intervals did not in general attain the desired confi-

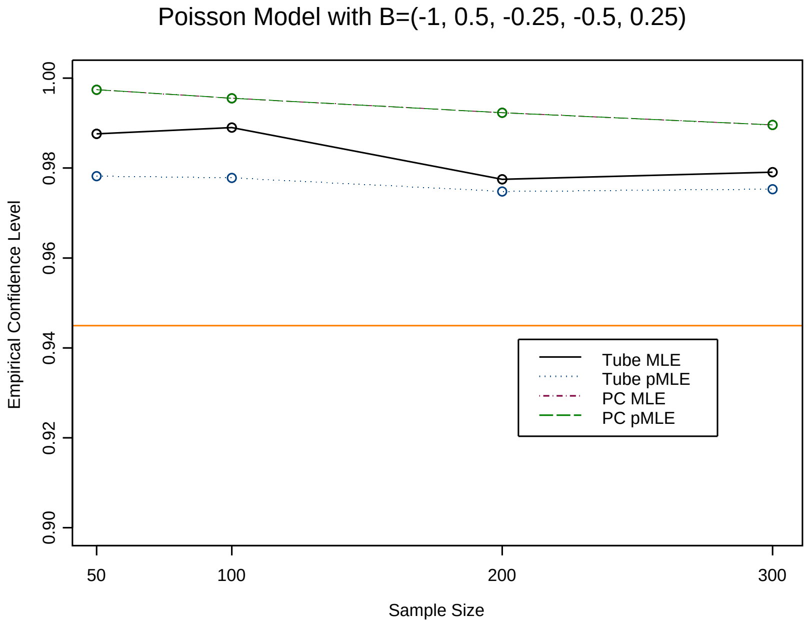

Figure 4.17: MLE and PMLE SCR Intervals for the Parameters

dence level until either $n = 200$ or $n = 300$ for most cases studied. At the smallest sample size ($n = 50$), the pMLE based interval attained the desired level of confidence and was not overly conservative (see Figure 4.16 or Figure 4.17). As the sample size increased, the pMLE based interval's empirical confidence level slowly approaches the desired confidence level, just as we saw with the mean response simulations. At the largest sample size simulated ($n = 300$), the pMLE based interval had an empirical confidence level between the various MLE intervals. In general, only the MLE Scheffé and PC intervals are more conservative than the two pMLE intervals, while all the MLE SCR intervals are a little less conservative. Thus, whether a practitioner utilizes the MLE or pMLE based intervals makes little difference at these larger sample sizes.

CHAPTER 5

CONCLUSIONS

5.1 Application of Proposed Methods

Using practical examples with emphasis placed on implementation of the procedures and the interpretation of the results, I will now illustrate how to utilize the proposed intervals. One example focuses on estimating the mean response of a GLM while the other utilizes a GLM with reference coded binary predictors to estimate a set of odds ratios.

5.1.1 Diabetes among Pima Women

Diabetes is a common disease among females of the Pima culture in Arizona. A study was conducted to better understand the incidence of diabetes in the population of Pima women [18]. One explanatory variable that is believed to be associated with diabetes in young Pima women, age 24 and younger, is the plasma glucose concentration. The glucose concentration was measured with an oral glucose tolerance test on the 51 Pima women aged 24 or younger. A binary variable indicates presence of diabetes in the young women (0=no diabetes and 1=diabetes), thus a simple logistic regression model is reasonable. Table 5.1 contains the estimated probability of diabetes for women with high glucose readings (140 and above). Individual 95% confidence intervals were calculated for each proportion as well as naive tube or SCR1 intervals with the restriction that the glucose was greater than 140. Note that each proportion in Table 5.1 uses the notation $\pi_{glucoselevel}$ with the estimated proportion giving the probability of diabetes for an individual Pima women with that glucose

Table 5.1: Intervals for Proportion of Pima Women with Diabetes

Point Estimate for π	95% MLE Individual Confidence Intervals for π	95% pMLE SCR1 Simultaneous Confidence Intervals for π
$\hat{\pi}_{140}$=0.268	(0.109,0.523)	(0.101,0.561)
$\hat{\pi}_{142}$=0.297	(0.120,0.567)	(0.108,0.598)
$\hat{\pi}_{143}$=0.312	(0.125,0.589)	(0.111,0.616)
$\hat{\pi}_{148}$=0.391	(0.155,0.693)	(0.128,0.703)
$\hat{\pi}_{151}$=0.443	(0.175,0.748)	(0.140,0.750)
$\hat{\pi}_{154}$=0.495	(0.197,0.796)	(0.151,0.792)
$\hat{\pi}_{177}$=0.831	(0.417,0.971)	(0.259,0.962)
$\hat{\pi}_{188}$=0.914	(0.540,0.990)	(0.322,0.984)
$\hat{\pi}_{199}$=0.958	(0.658,0.996)	(0.391,0.994)

level. A discussion of the results follows the presented interval estimates in Table 5.1. First note that, as expected, the individual confidence intervals are narrower than the simultaneous intervals. Thus, using the individual confidence intervals, it will be easier to reject certain proportions as the true value. Note that the proportion estimates given in Table 5.1 are from the pMLE estimated model. The MLE model was $logit(\hat{\pi}) = -10.840 + 0.0703X$ while the pMLE model was $logit(\hat{\pi}) = -10.826 + 0.0701X$. The change in the length and center of the intervals could lead to very different conclusions given certain research questions.

5.1.2 Depression in Adolescents

A study was conducted on the classification of depression (high or low) among adolescents between the ages of 12 to 18 with either learning disabilities (LD) or with serious emotional disturbances (SED) [19]. Six risk factors for high levels of depression were considered (as a combination of the age (12-14,15-16,17-18) factor and group (LD and SED) factor). Thus, there were a total of 6 levels for a single categorical predictor variable. In the text, Epidemiological Research Methods [19], it is suggested that a logistic regression model is appropriate for these data. The estimated odds ratios are provided to ascertain any differences in the level of depression for the different groups. Simultaneous estimation of the odds ratio from the logistic regression model should be considered here since it is reasonable to make conclusions about the group with the highest or lowest odds of high levels of depression. As suggested in the text [19], the group with the lowest risk of high levels of depression (17-18,SED) was the referent or control category. The reference coding takes this into account so that every log odds ratio refers back to that baseline category. The estimated log odds ratios for the 5 estimated slope coefficients are given in table 5.2. Note that $\beta_{age,condition}$ and $\theta_{age,condition}$ refer to the parameter or odds ratio respectively for an adolescent of a particular age group and condition. Additionally, Table 5.2 contains the estimated model odds ratios comparing the odds of high levels of depression for each risk category with reference to the 17-18,SED category and the individual and restricted-Scheffé or PC simultaneous intervals. All point estimates utilize the pMLE estimated model. Note that the 95% individual confidence intervals demonstrated an odds ratio significantly different than 1 for the case where the adolescent was age 12-14 and learning disabled. Once simultaneous adjustments are made, the significant association no longer exists. This would be an example of a case where the one-at-a-time intervals and simultaneous intervals would contradict and care should be taken about which methodology is appropriate. For instance, in order to say that the odds of high levels of depression is highest among the group aged 12 to 14 and with serious emotional disorders, simultaneous intervals would need to be computed. Clearly, that conclusion should not be made for this study since that is not supported via the simultaneous intervals.

Table 5.2: Intervals for Odds Ratio for Depression in Adolescents

Point Estimate for Odds Ratio	95% MLE Individual Confidence Interval for θ	95% pMLE PC Simultaneous Confidence Interval for θ
$\hat{\theta}_{12-14,LD}$=1.939	(0.812,4.627)	(0.384,8.926)
$\hat{\theta}_{12-14,SED}$=4.683	(1.642,13.383)	(0.671,29.874)
$\hat{\theta}_{15-16,LD}$=1.616	(0.651,4.102)	(0.300,8.053)
$\hat{\theta}_{15-16,SED}$=1.458	(0.520,4.088)	(0.222,9.189)
$\hat{\theta}_{17-18,LD}$=1.844	(0.696,4.884)	(0.307,10.381)
$\hat{\beta}_{17-18,SED}$=1.00		

5.2 Overall Conclusions

The proposed pMLE based intervals, including the pScheffé, pMLE restricted-Scheffé, and pSCR1 intervals, did improve small sample estimation over the usual MLE based intervals. As demonstrated, the usual MLE based intervals often could not attain the desired level of confidence for what was considered a small sample size for the varying models. In contrast, the pMLE did attain the desired level of confidence. However, the penalty with using the pMLE based intervals is that these intervals are very conservative at these small sample sizes. As the sample size increases to moderate levels, the distinction between the MLE and pMLE based intervals lessens, but the pMLE intervals, as a whole, tend to be less conservative than the MLE based intervals.

5.2.1 Recommendations for Estimating the Mean Response

When selecting an interval method for a set of comparisons, attention should be paid to what the set of predictor variables are. For instance, when there are many comparisons and interest is focused on the entire estimation space, the Scheffé intervals would ensure the desired level of confidence. However, if a restricted interval of the predictor variable is of interest or a finite number of discrete points embedded in a continuous domain is of interest, then the restricted-Scheffé or one of the SCR methods, respectively, is most advantageous.

In summary, recommendations are to use pMLE intervals for all small to moderate sample sizes. Specifically, the pMLE SCR1 intervals were the least conservative of all the pMLE based intervals and are thus the best choice. However, for ease of computation, the restricted-Scheffé pMLE intervals are a good second choice with far more computational ease. Then, for moderate and larger sample sizes, the pMLE based intervals are again recommendeded, though the behavior of these intervals have not been studied for any sample size greater than 50. Again, the SCR1 pMLE interval is the overall best choice. Though even less distinction may be made between the pMLE intervals at the larger sample sizes.

5.2.2 Recommendations for Estimating the Parameters

Overall, for small to moderate sample sizes, the SCR1 pMLE attains the desired level of confidence while not being as conservative as the PC pMLE interval. Thus, for most cases where the sample size is greater than 50, I recommend utilizing a naive tube critical point with the pMLE estimators. For sample sizes smaller than 50, the cautious choice would be the PC pMLE interval. This is a slightly conservative interval, yet it attains the desired level of confidence, unlike any other competing interval. I would not recommend using the MLE based intervals at smaller sample sizes. If the sample size is moderate to large (n=200 or 300) there is little observed difference between the pMLE and MLE intervals and thus, for convenience, the MLE intervals may justifiably be utilized.

5.3 Future Research

There are many avenues to explore in the future that are related to this research topic. Some of these include performing in-depth simulations of these methods for more complex GLMs. Namely, if a GLM has a predictor matrix, given by $\boldsymbol{X}$, that is a mix of categorical and continuous predictors, then nothing is known about how the aforementioned simultaneous procedures would behave. There are a host of other issues to explore as well when we have a predictor matrix such as $\boldsymbol{X}$. For instance, what multiple comparison techniques are applicable, what should we compare, and how do we adjust for the other variables in the model? Additionally, for the single predictor variable case, other configurations of the contrast matrix $\boldsymbol{C}$ should be considered. For example, these other forms of $\boldsymbol{C}$ could be utilized to assess how the methods perform for all-pairwise types of comparisons. Also, GLMs with interaction and quadratic terms need to be explored. This entails describing what the odds ratios and relative risks are for the interaction and quadratic terms as well as evaluating and determining the appropriate simultaneous estimation techniques for these more complex models. Other estimation methods for the Poisson models should also be explored. As demonstrated in the simulation studies, the Poisson models may have over-estimated the standard errors associated with the parameters. Thus, sandwich variance estimators via generalized estimating equations as described in section 4.1.1 would be a reasonable solution to this problem. Finally, pMLE based parameter estimates for improving GLM estimation should be explored, particularly in cases where the sample size is typically quite small. For example, a natural application is dose-response models. These models typically have 10 or less replications per dose and there are frequent problems with bias in the parameters estimates. Additionally, when at least one dose has either all success's or all failure's for the response variable, the usual MLE estimates fail. The pMLE estimates would be a reasonable solution to this problem and some simulation studies to assess the accuracy of these estimators would be beneficial.

BIBLIOGRAPHY

[1] Dole, Savitz, Hertz-Picciotto, Siega-Riz, McMahon, and Buekens, "Maternal stress and preterm birth," *American Journal of Epidemiology*, vol. 157, no. 1, pp. 14–24, 2003.

[2] K. Rothman, *Modern Epidemiology.* Little, Brown and Company, 1986.

[3] H. Leung and L. Kupper, "Comparisons of confidence intervals for attributable risk," *Biometrics*, vol. 37, pp. 293–302, 1981.

[4] J. Benichou and M. Gail, "Variance calculations and confidence intervals for estimates of the attributable risk based on logistic models," *Biometrics*, vol. 46, pp. 991–1003, 1990.

[5] W. W. Hauck, "A note on confidence bands for the logistic response curve," *American Statistician*, vol. 37, no. 2, pp. 158–160, 1983.

[6] G. Casella and W. Strawderman, "Confidence bands for linear regression with restricted predictor variables," *Journal of the American Statistical Association*, vol. 75, pp. 862–868, 1980.

[7] D. Piegorsch and G. Casella, "Confidence bands for logistic regression with restricted predictor variables," *Biometrics*, vol. 4, pp. 739–750, 1988.

[8] S. Axler, *Linear Algebra Done Right.* Springer Verlag, 1997.

[9] J. Sun, S. Loader, and D. McCormick, "Confidence bands in generalized linear models," *The Annals of Statistics*, vol. 28, no. 2, pp. 429–460, 2000.

[10] D. Naiman, "Conservative confidence bands in curvilinear regression," *The Annals of Statistics*, vol. 14, no. 3, pp. 896–906, 1986.

[11] P. Hall, *The bootstrap and Edgeworth expansion.* New York: Springer-Verlag, 1 ed., 1992.

[12] D. Firth, "Bias reduction of maximum likelihood estimates," *Biometrika*, vol. 80, no. 1, pp. 27–38, 1993.

[13] P. McCullagh and J. Nelder, *Generalized Linear Models 2nd ed.* Chapman and Hall, 1989.

[14] M. McCann and D. Edwards, "A path length inequality for the multivariate-t distribution, with applications to multiple comparisons," *Journal of the American Statistical Association*, vol. 91, pp. 211–216, March 1996.

[15] S. Greenland, "Model-based estimation of relative risks and other epidemiologic measures in studies of common outcomes and in case-control studies," *American Journal of Epidemiology*, vol. 160, no. 4, pp. 301–305, 2004.

[16] L. McNutt, W. Chuntao, X. Xiaonan, and J. Hafner, "Estimating the relative risk in cohort studies and in studies with common outcomes," *American Journal of Epidemiology*, vol. 157, no. 10, pp. 940–943, 2003.

[17] A. Agresti, *Categorical Data Analysis.* Hoboken, New Jersey: Wiley-Interscience, 2 ed., 2000.

[18] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," *Proceedings of the Symposium on Computer Applications in Medical Care (Washington, 1988*, p. 261265, 1988.

[19] D. McNeil, *Epidemiological Research Methods.* New York: John Wiley and Sons, 1 ed., 1996.

[20] S. Walter, “Estimation and interpretation of the attributable risk in health research,” *Biometrics*, vol. 32, pp. 829–849, 1976.

[21] Dwyer, Stankovich, Blizzard, FitzGerald, Dickinson, Reilly, Williamson, Ashbolt, Berwick, and Sale, “Does the addition of information on genotype improve prediction of the risk of melanoma and nonmelanoma skin cancer beyond that obtained from skin phenotype?,” *American Journal of Epidemiology*, vol. 159, no. 9, pp. 826–833, 2004.

[22] Liou, Adler, FitzSimmons, Cahill, Hibbs, and Marshall, “Predictive 5-year survivorship model of cystic fibrosis,” *American Journal of Epidemiology*, vol. 153, no. 4, pp. 345–352, 2001.

[23] Abrahamowicz, MacKenzie, and Esdaile, “Time-dependent hazard ratio: Modeling and hypothesis testing with application in lupus nephritis,” *Journal of the American Statistical Association*, vol. 91, no. 436, pp. 1432–1439, 1996.

[24] Fitzgerald, Robinson, and Pester, “Application of benzo(a)pyrene and coal tar tumor dose-response data to a modified benchmark dose method of guideline development,” *Environmental Health Perspectives*, vol. 112, no. 14, pp. 1341–1347, 2004.

[25] S. Greenland, “Interval estimation by simulation as an alternative to and extension of confidence intervals,” *International Journal of Epidemiology*, vol. 33, no. 6, pp. 1389–1397, 2004.

APPENDIX A

FIGURES

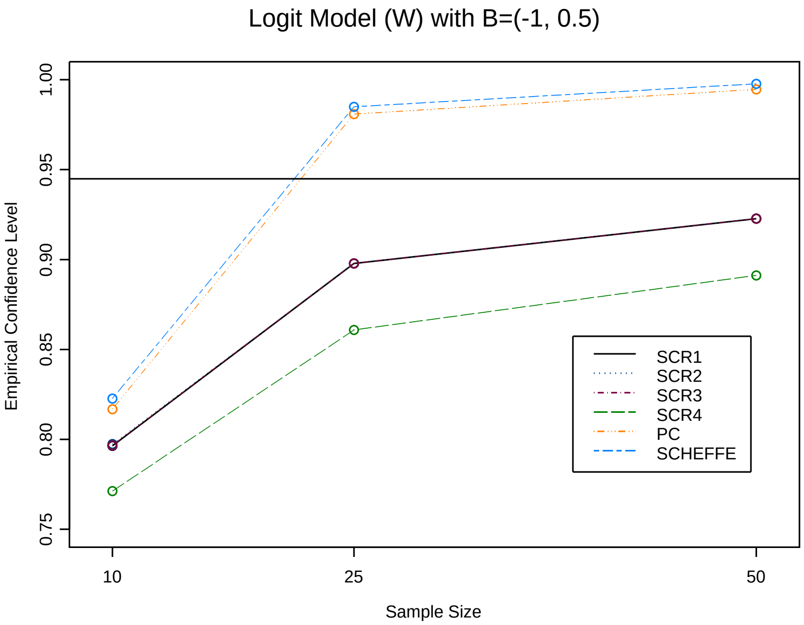

Figure A.1: MLE Intervals Logit-Wide Range B=1

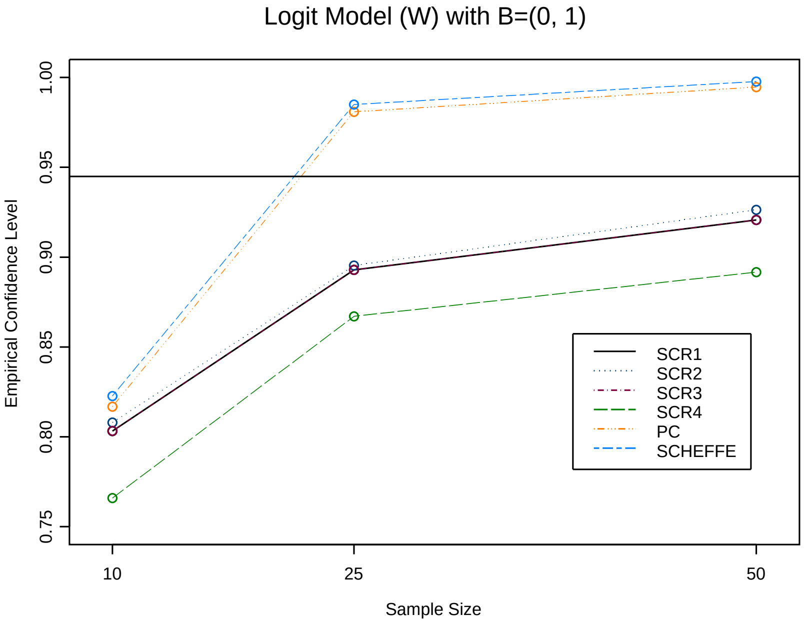

Figure A.2: MLE Intervals Logit-Wide Range B=2

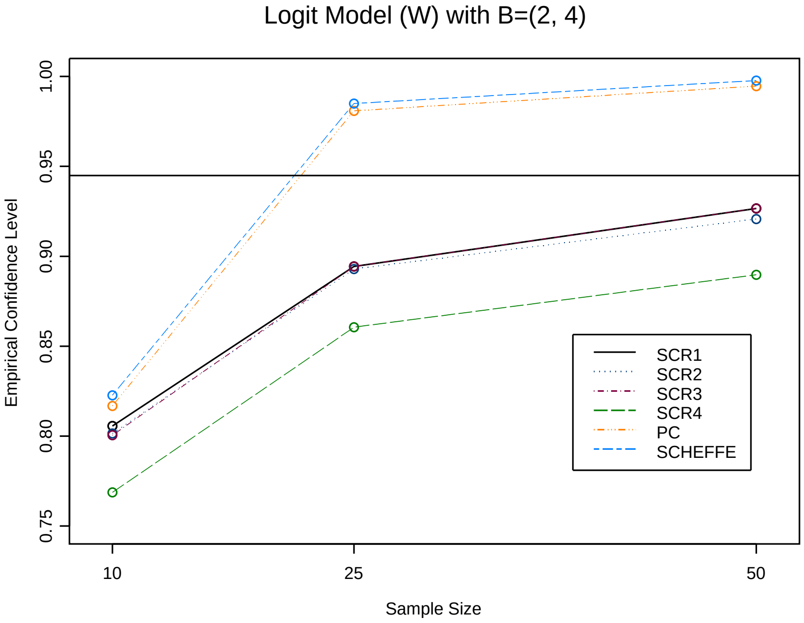

Figure A.3: MLE Intervals Logit-Wide Range B=3

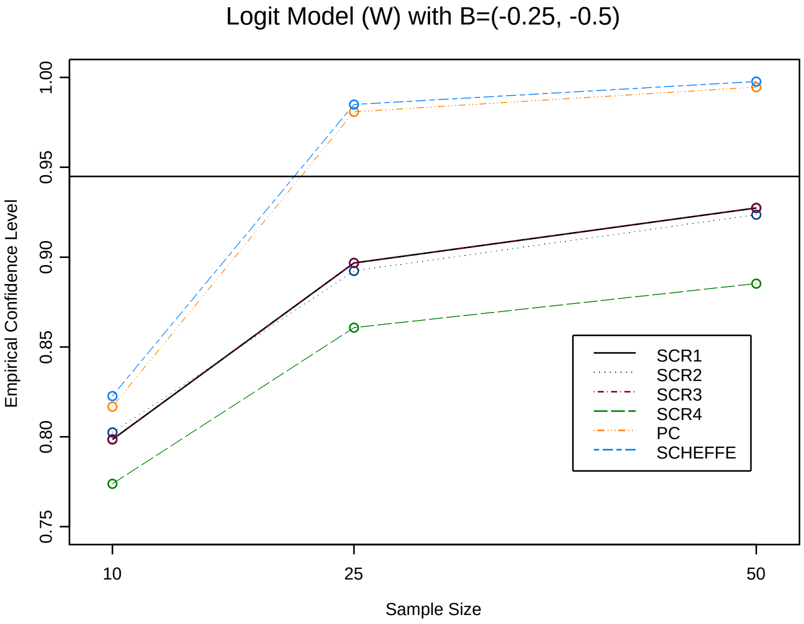

Figure A.4: MLE Intervals Logit-Wide Range B=4

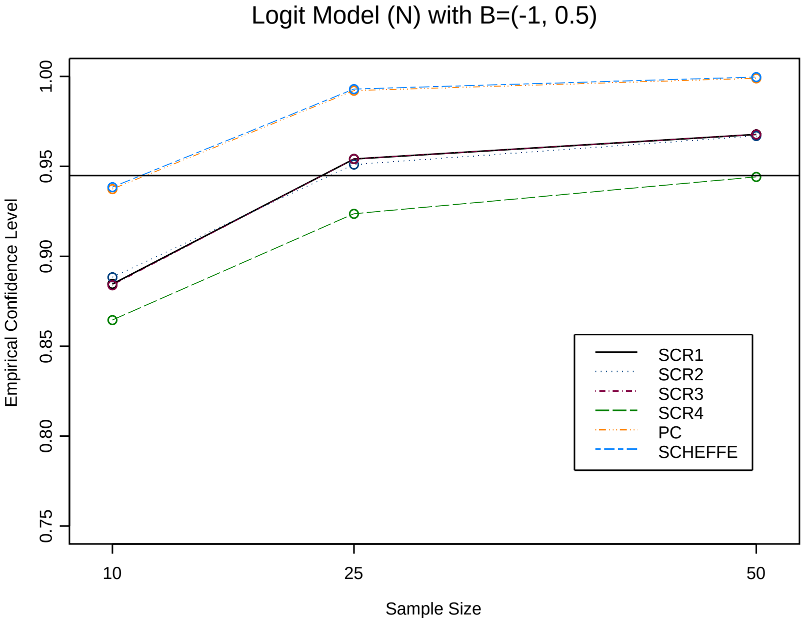

Figure A.5: MLE Intervals Logit-Narrow Range B=1

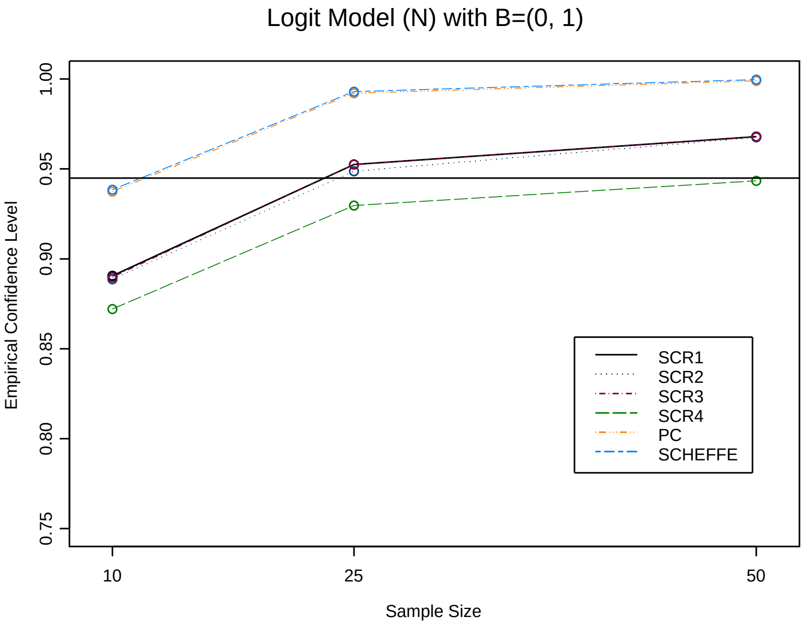

Figure A.6: MLE Intervals Logit-Narrow Range B=2

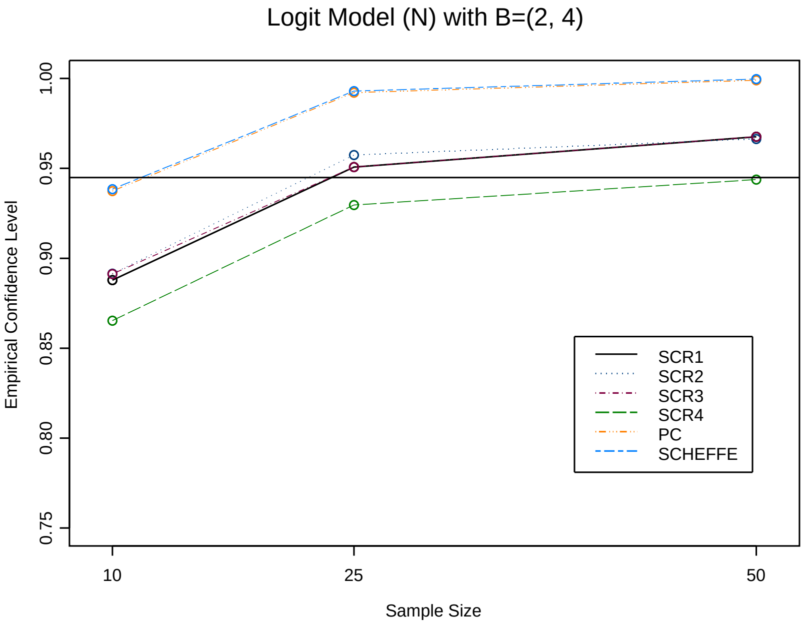

Figure A.7: MLE Intervals Logit-Narrow Range B=3

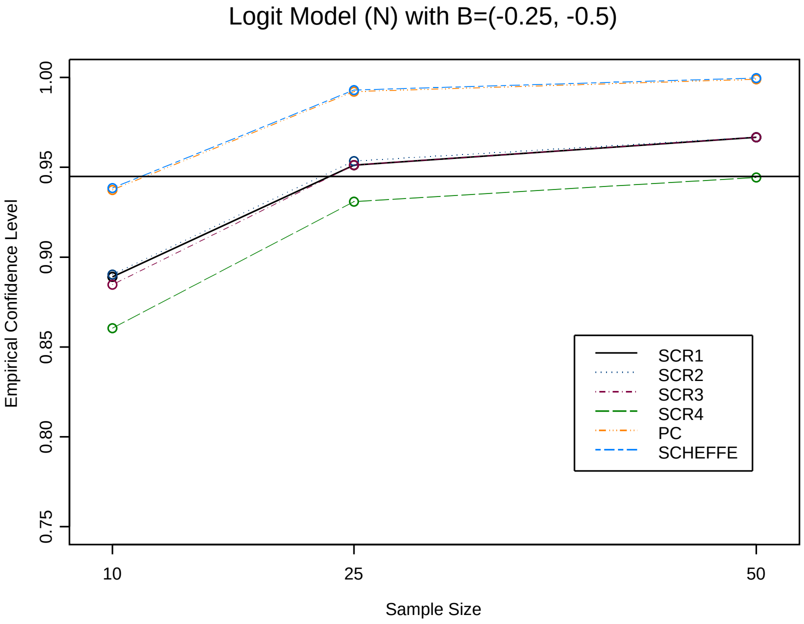

Figure A.8: MLE Intervals Logit-Narrow Range B=4

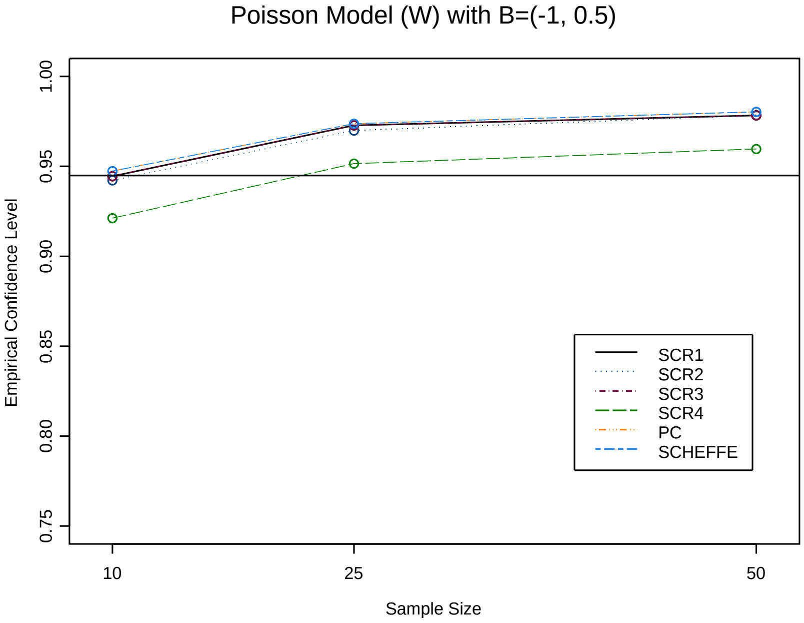

Figure A.9: MLE Intervals Poisson-Wide Range B=1

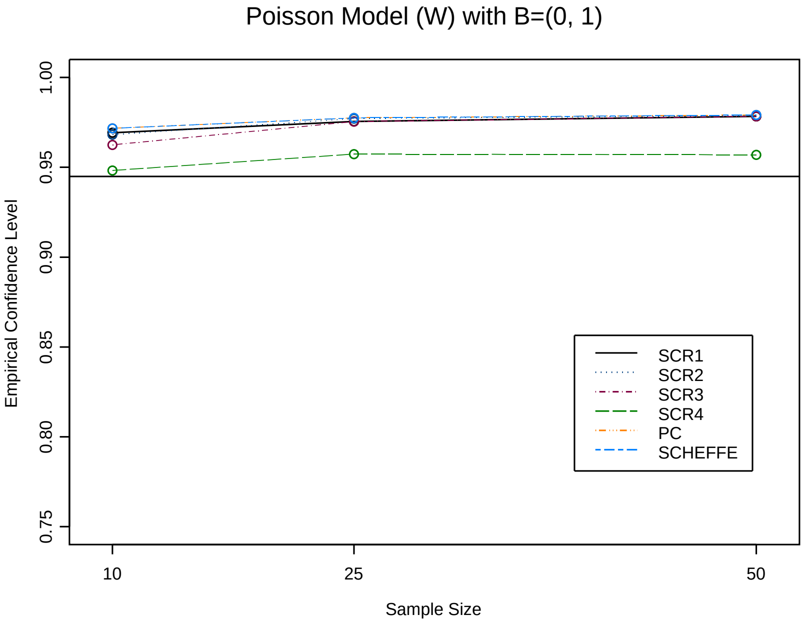

Figure A.10: MLE Intervals Poisson-Wide Range B=2

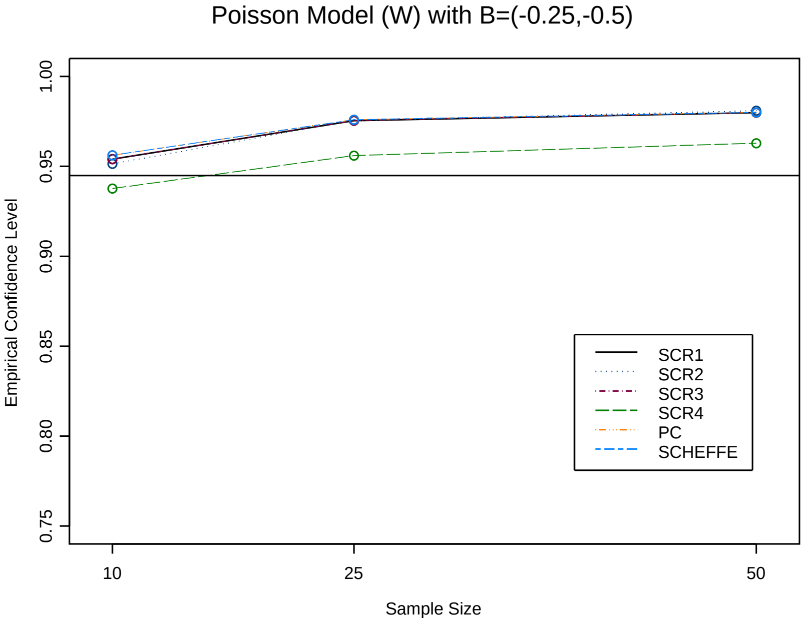

Figure A.11: MLE Intervals Poisson-Wide Range B=3

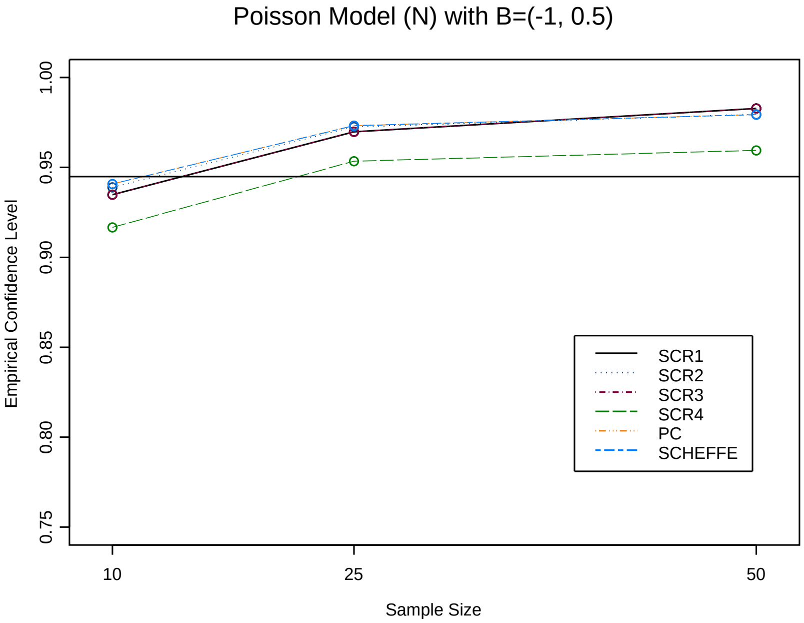

Figure A.12: MLE Intervals Poisson-Narrow Range B=1

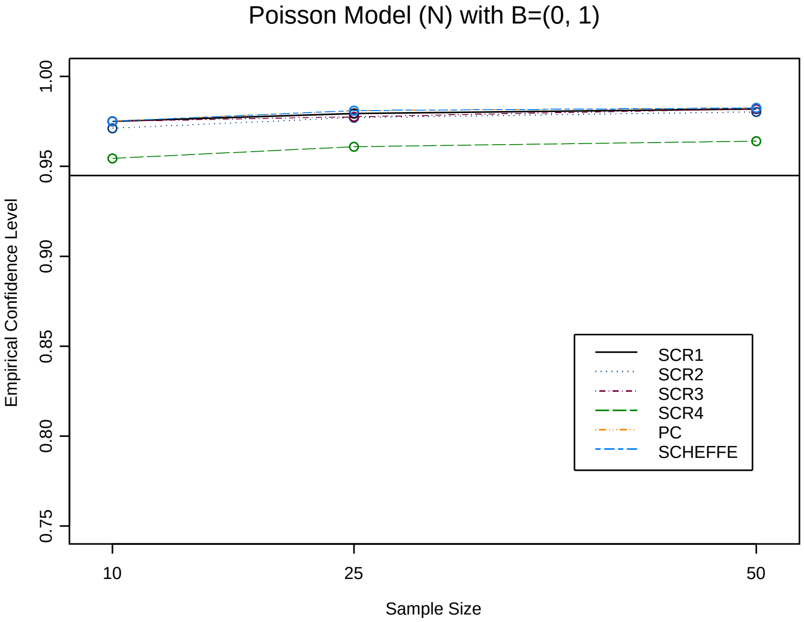

Figure A.13: MLE Intervals Poisson-Narrow Range B=2

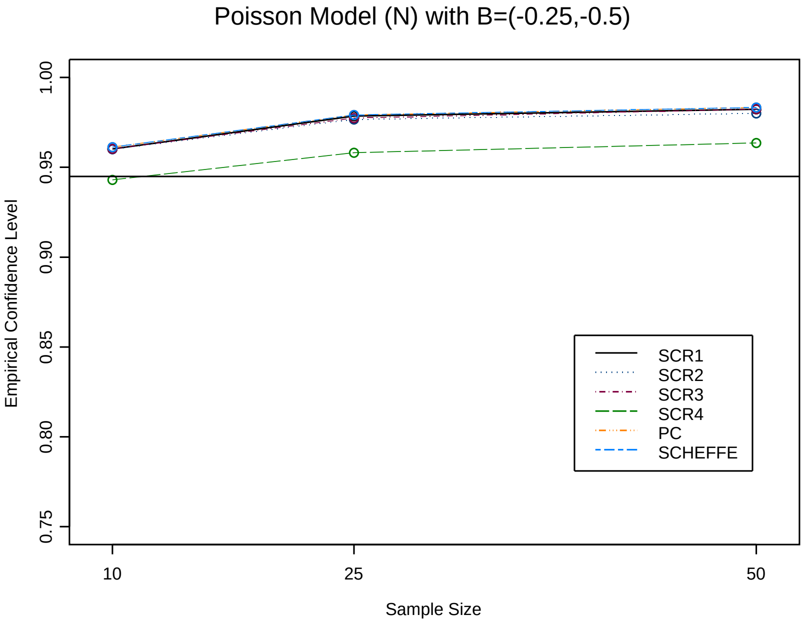

Figure A.14: MLE Intervals Poisson-Narrow Range B=3

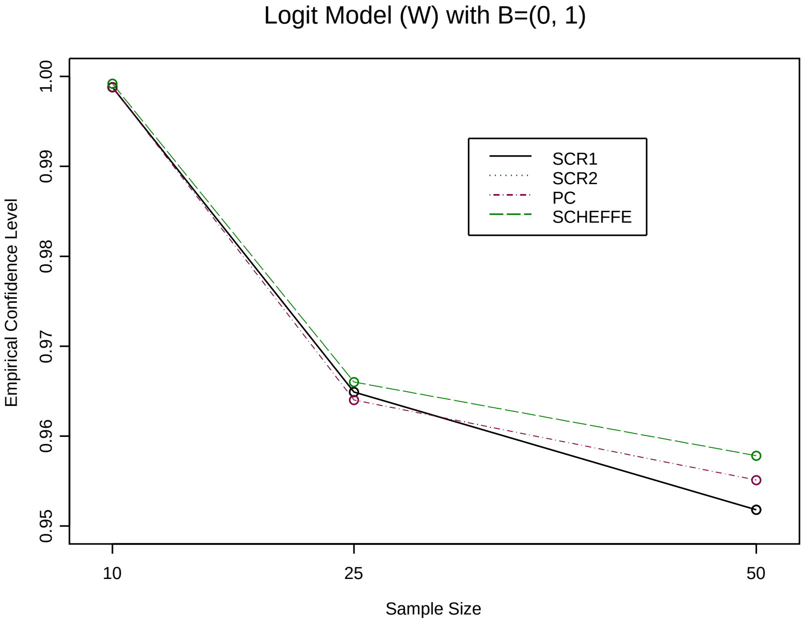

Figure A.15: pMLE Intervals Logit-Wide Range B=2

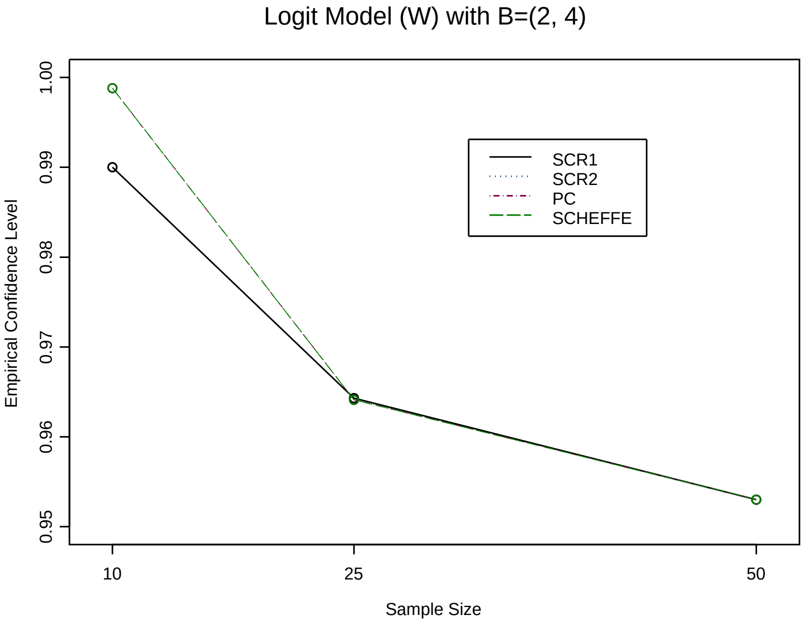

Figure A.16: pMLE Intervals Logit-Wide Range B=3

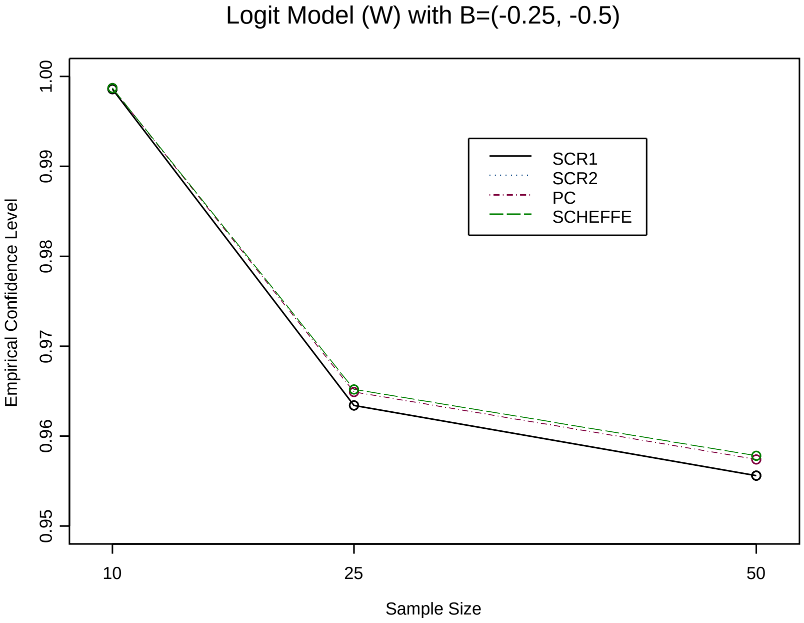

Figure A.17: pMLE Intervals Logit-Wide Range B=4

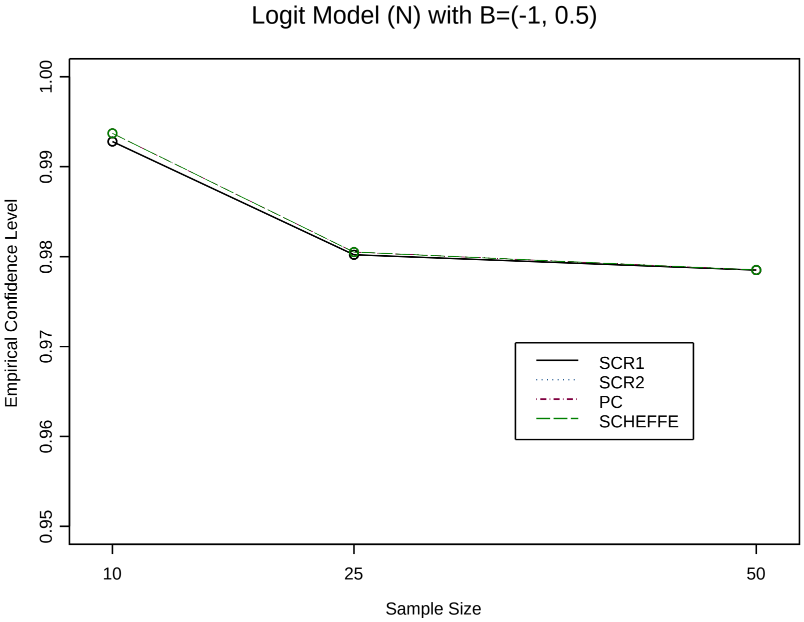

Figure A.18: pMLE Intervals Logit-Narrow Range B=1

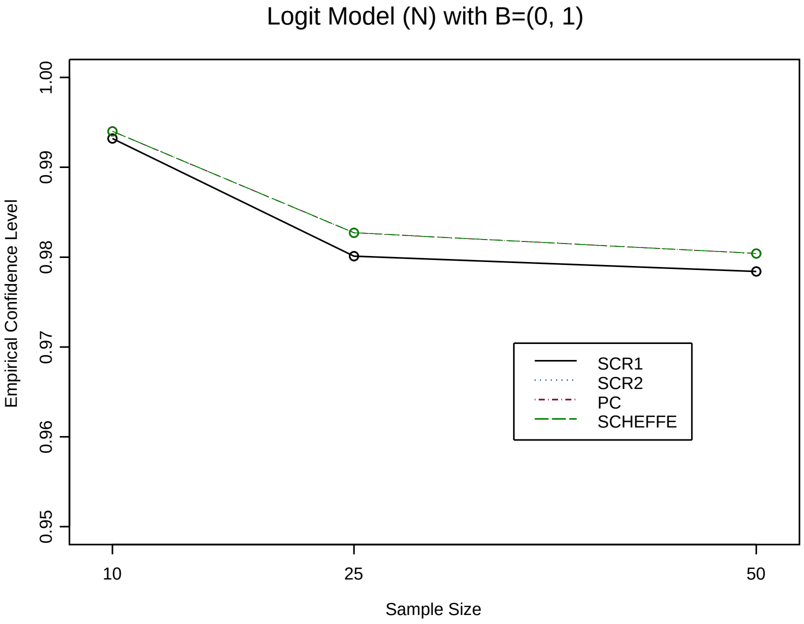

Figure A.19: pMLE Intervals Logit-Narrow Range B=2

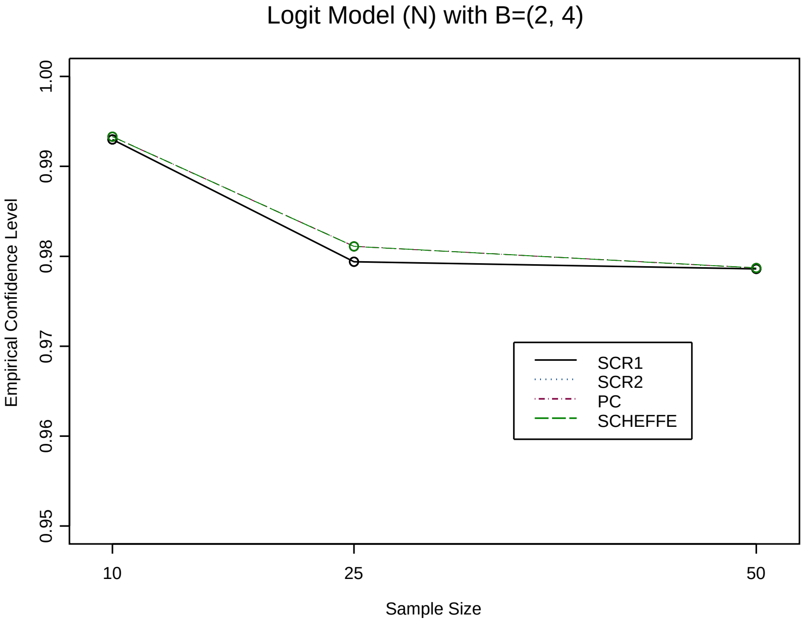

Figure A.20: pMLE Intervals Logit-Narrow Range B=3

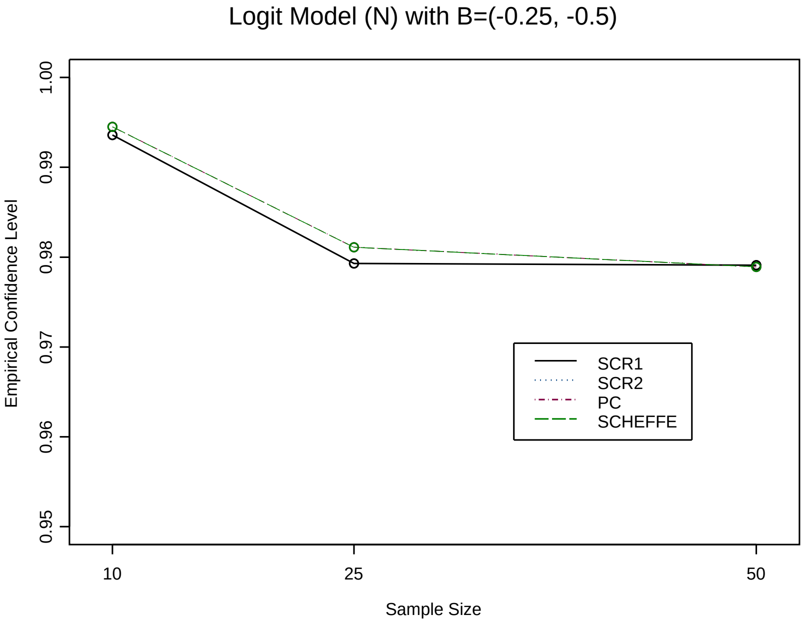

Figure A.21: pMLE Intervals Logit-Narrow Range B=4

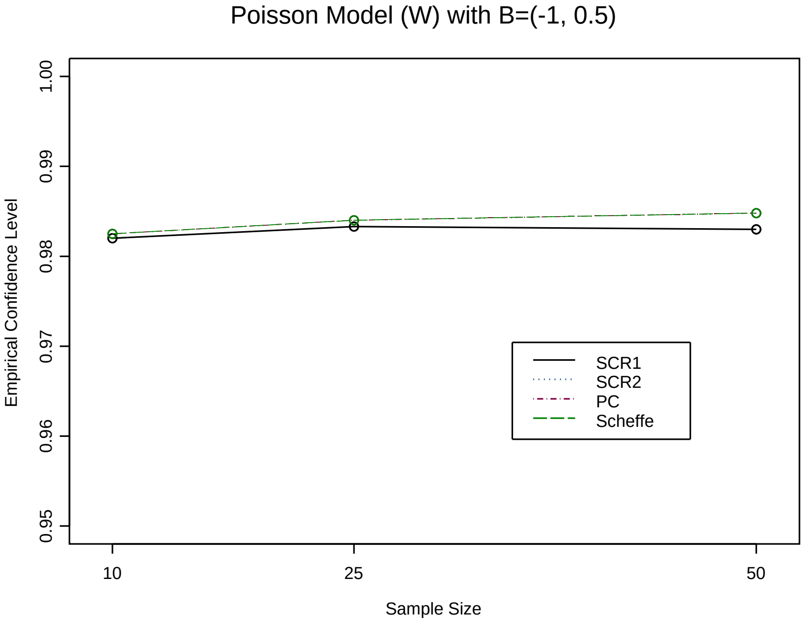

Figure A.22: pMLE Intervals Poisson-Wide Range B=1

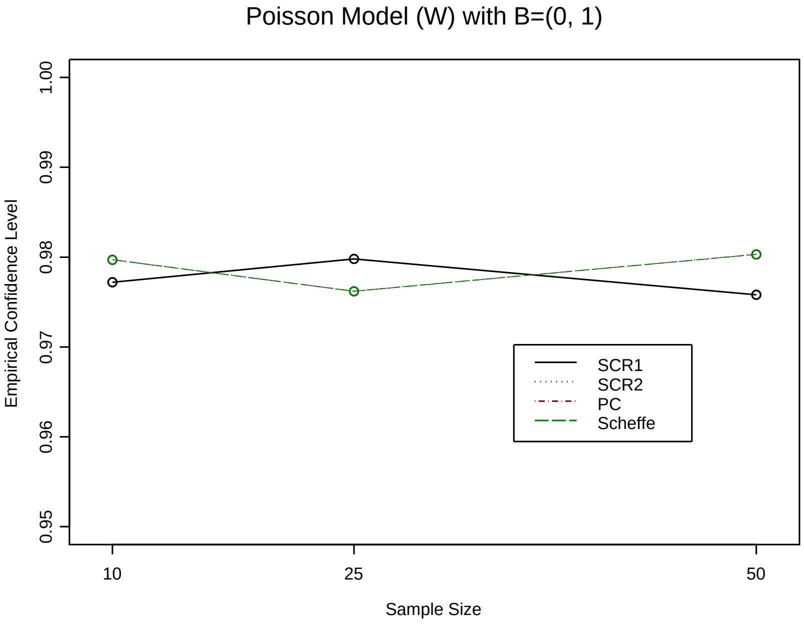

Figure A.23: pMLE Intervals Poisson-Wide Range B=2

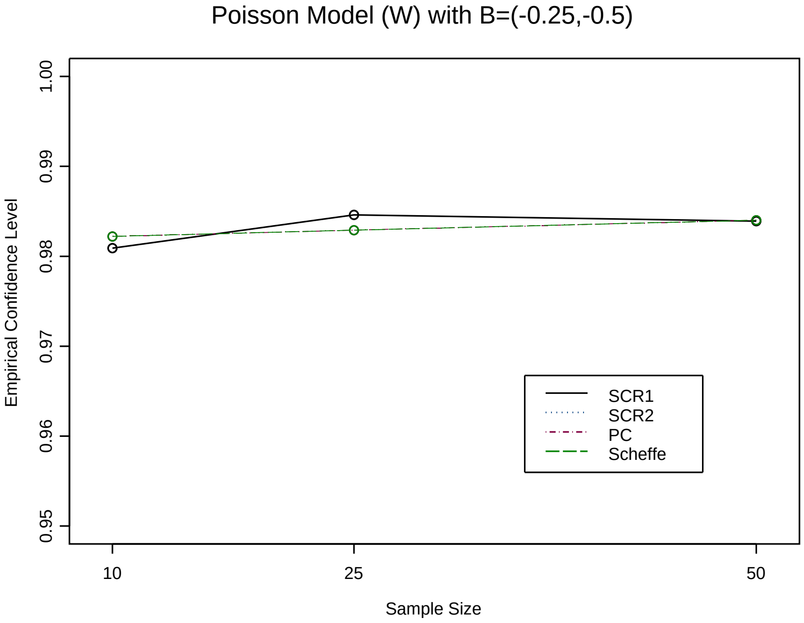

Figure A.24: pMLE Intervals Poisson-Wide Range B=3

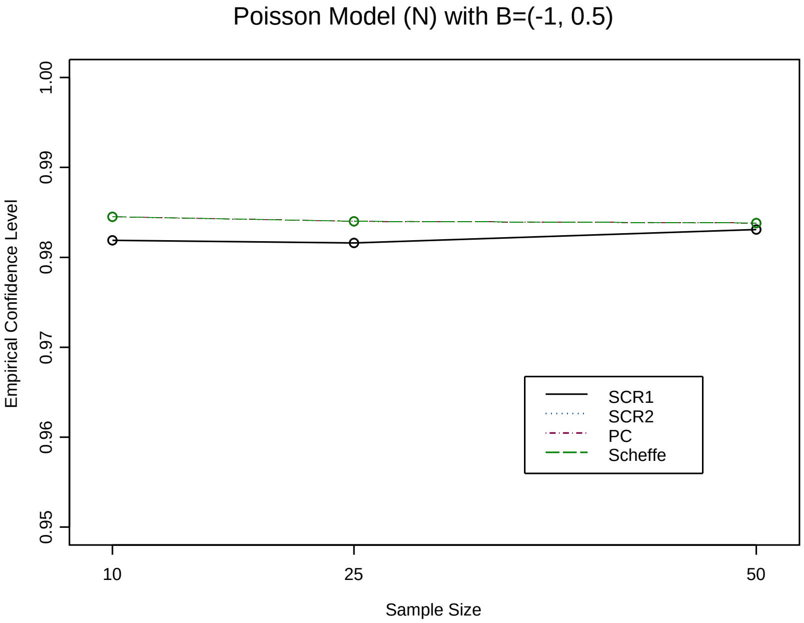

Figure A.25: pMLE Intervals Poisson-Narrow Range B=1

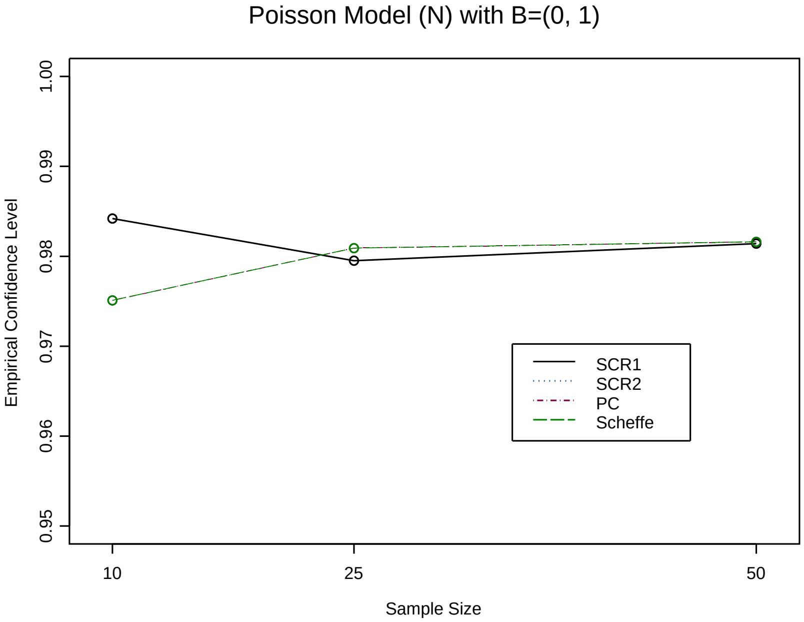

Figure A.26: pMLE Intervals Poisson-Narrow Range B=2

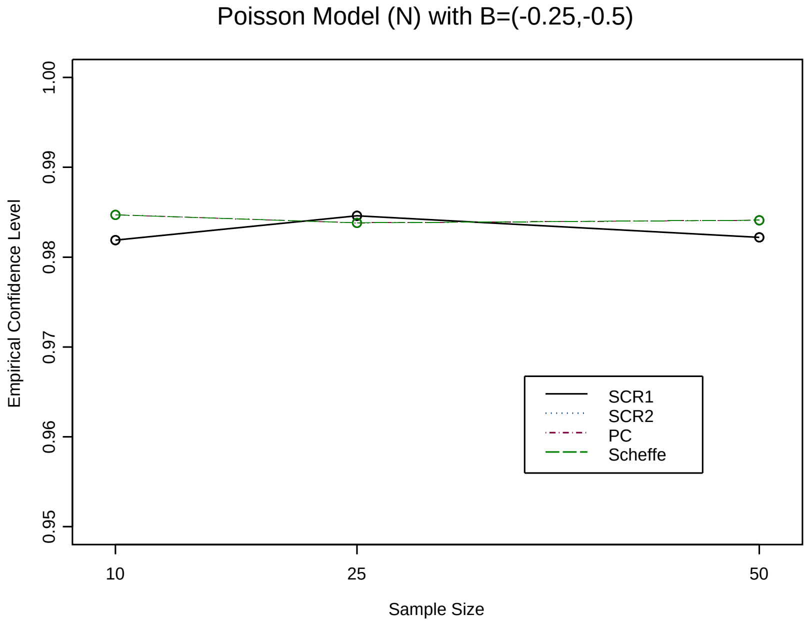

Figure A.27: pMLE Intervals Poisson-Narrow Range B=3

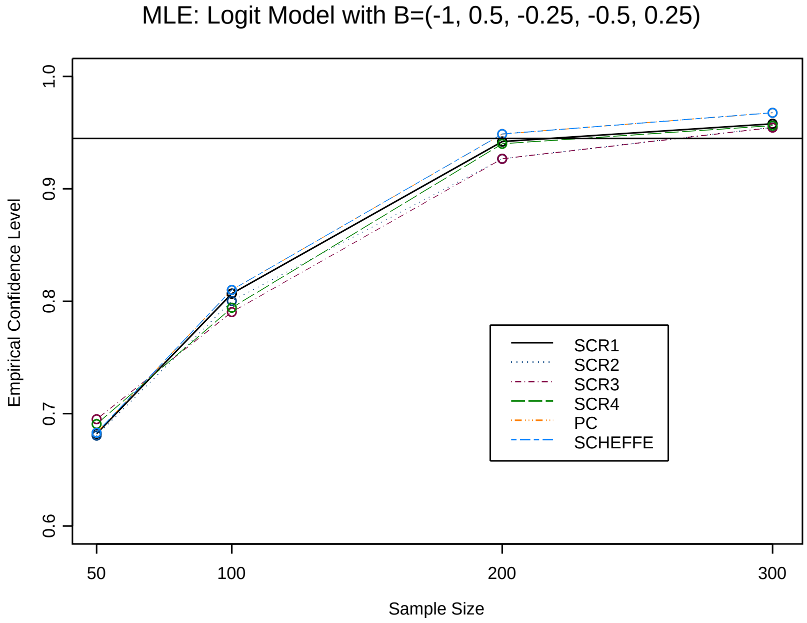

Figure A.28: MLE Intervals for the Parameters - Logit

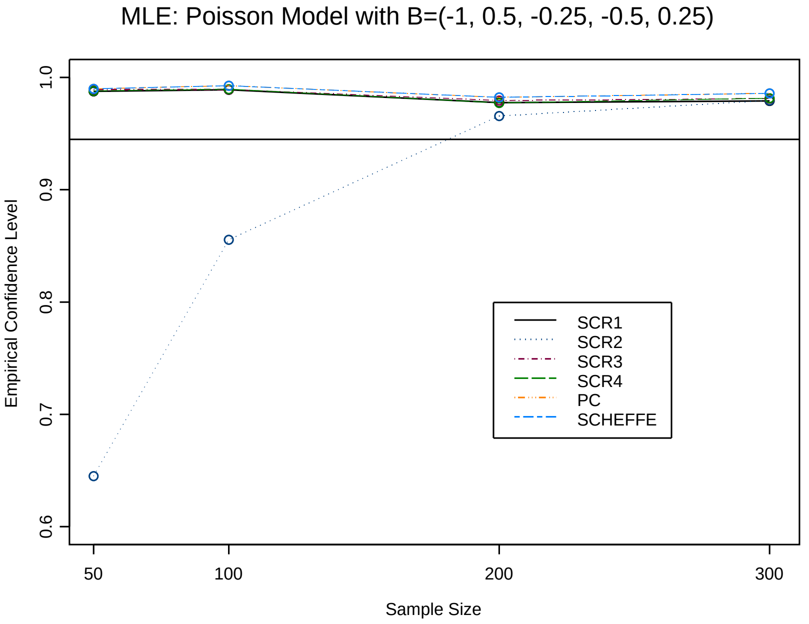

Figure A.29: MLE Intervals for the Parameters - Poisson

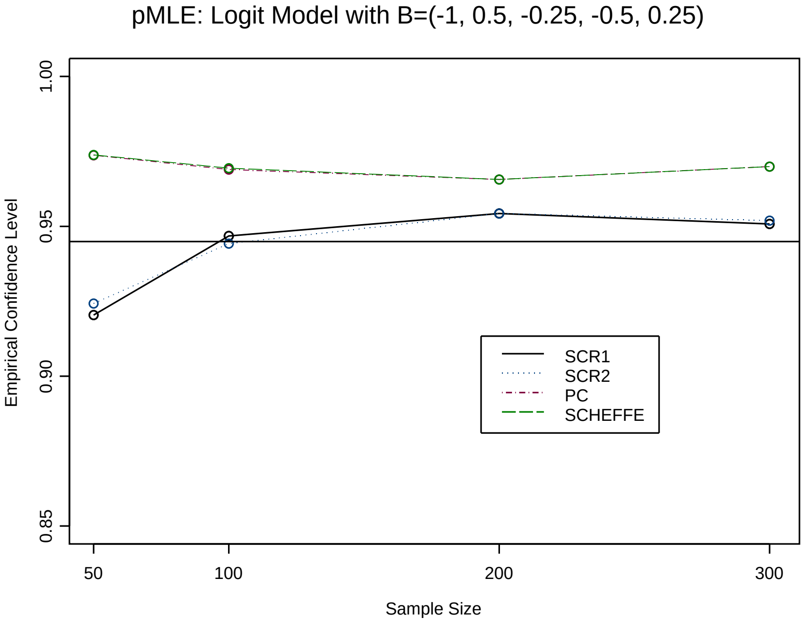

Figure A.30: pMLE Intervals for the Parameters - Logit

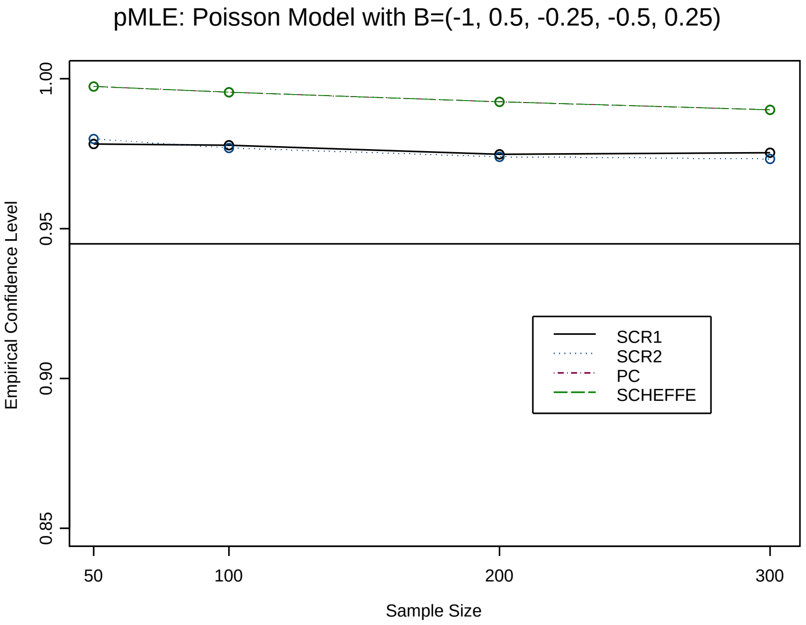

Figure A.31: pMLE Intervals for the Parameters - Poisson

APPENDIX B

Moments of Binomial and Poisson Random Variables

The moments of the binomial and poisson random variables are utilized in the SCR methodology. The following lays out the derivation of these as used in the coding for calculating the various SCR critical values. Both derivations calculate the generalized linear model moment, $b(\theta)$ based on the exponential class form $exp\{y\theta - b(\theta) + a(y)\}$ as in [13].

B.1 The Binomial Random Variable

If $Y \sim$ BIN(n,π), then the likelihood is given by,

$$\binom{n}{\pi}\pi^{\sum y}(1-\pi)^{n-\sum y} = exp\{\sum y(log\pi - log(1-\pi)) + nlog(1-\pi) + log\binom{n}{\pi}\}$$

. Thus, $b(\pi) = -nlog(1-\pi)$ where $\pi = \frac{e^{\eta}}{1+e^{eta}}$ for $\eta = \theta$ given that the logit or log is a canonical link. Then, if we reparameterize $b(\pi)$ in terms of θ, then $b(\theta) = nlog(1+e^{\theta})$. Then the moments of the binomial random variable are given by the derivatives of $b(\theta)$. The first moment is:

$$\mu_\theta = b'(\theta) = \frac{ne^\theta}{1+e^\theta} = \frac{n\frac{\pi}{1-\pi}}{1+\frac{\pi}{1-\pi}} = n\pi.$$

The second moment is

$$\sigma^2_\theta = b''(\theta) = \frac{ne^\theta}{(1+e^\theta)^2}.$$

The third moment is

$$b^3(\theta) = \frac{ne^\theta(1-e^\theta)}{(1+e^\theta)^3}$$

and the fourth moment is

$$b^4(\theta) = ne^\theta(1-e^\theta)\frac{-3e^\theta}{(1+e^\theta)^4} + \frac{e^\theta - 2e^\theta}{(1+e^\theta)^3} == \frac{e^\theta - 4e^{2\theta} + e^{3\theta}}{(1+e^\theta)^4}.$$

B.2 The Poisson Random Variable

When $Y \sim$ POI(μ), then the likelihood is given by,

$$\frac{e^{n\mu}\mu^{\sum y}}{y!} = exp\{-log(y!) + \sum ylog\mu - \mu\}.$$

Thus, $b(\mu) = \mu$ and given that $\theta = log\mu$, then $b(\theta) = e^\theta$. Then the derivatives are all given by

$$b'(\theta) = b''(\theta) = b^3(\theta) = b^4(\theta) = e^\theta.$$

APPENDIX C

Restricted-Scheffé Methodology

The following algorithm is adapted from both Casella and Strawderman (1980) and Piegorsch and Casella (1988) for any GLM. Suppose we have interest in constrained regions of the form $R_{\boldsymbol{x}_i} = \{a_{11} \leq x_1 \leq a_{12}, a_{21} \leq x_2 \leq a_{22}, \ldots, a_{k1} \leq x_k \leq a_{k2}\}$ where the a_{mi} ($m = 1, \ldots, K$ and $i = 1, 2$) are specified. Generally, this algorithm finds a set of vertices that are of the form $\Omega_{\boldsymbol{x}_i} = \{\boldsymbol{x} : \sum_{m=1}^{r} x_m^2 \geq q^2 \sum_{m=r+1}^{K+1} z_m^2\}$ that contain the true set of vertices for the estimation space of interest. This subset, $\Omega_{\boldsymbol{x}_i}^R$, is the $\Omega_{\boldsymbol{x}_i}$ that most closely matches $\boldsymbol{R}_x$ and on which we will base the critical point for the Scheffé-based intervals. The algorithm, closely following the one outlined by Piegorsch and Casella [7], is as follows:

1) Find the 2^k vertices of the hyper-rectangle defined by $R_{\boldsymbol{x}_i}$. These are denoted $v_j = \{v_{mj}\}_{m=1}^{k}$ where each $v_{mj} = (a_{ml}, a_{m'l})$ for $j = 1, \ldots, 2^k$, $m = 1, \ldots, k$, $m' = 1, \ldots, k$, and $l = 1, \ldots, K$. Note that when estimating the mean response all $v_{0j} = 1$ while estimating the vector of the regression coefficients all $v_{0j} = 0$.

2) Compute the diagonalized vertices, $\boldsymbol{\psi}_j = \boldsymbol{D^{-1/2}U'v_j} = \{\psi_{Mj}\}_{M=1}^{k+1}$ where $j = 1, \ldots, 2^k$ ranges the dimension of the k-dimensional real set, $\mathbb{R}^k$. The M^{th} element of this vector can be written as $\psi_{Mj} = \lambda_M^{-1/2} \sum_{L=1}^{K+1} u_{LM} v_{L-1,j}$ where λ_M is the M^{th} diagonal of the eigenvalue matrix $\boldsymbol{D}$, u_{LM} is in the L^{th} and M^{th} row and column of the eigenvector matrix $\boldsymbol{U}$, and $M = 1, \ldots, k$. At this step, we have the diagonalized vertices that need to be matched as closely as possible to vertices of sets of the form of $\Omega_{\boldsymbol{z}_i}$.

3) In order to find the set of vertices that are the closest match to R_z, compute the minimum and maximum among the diagonalized vertices $z_M^{max} = max|\boldsymbol{\psi}_{Mj}|$ and

$$z_M^{min} = \begin{cases} 0 & \text{if } min_j\{\boldsymbol{\psi}_{Mj}\} < 0 < max_j\{\boldsymbol{\psi}_{Mj}\} \\ min_j\{|\psi_{Mj}|\} & \text{otherwise} \end{cases}$$

This summarizes the vertices by recovering the maximum and minimum.

4) Calculate the following quantity,

$$Q_r^2 = \sum_{M=1}^{r} (z_M^{min})^2 / \sum_{M=r+1}^{k+1} (z_M^{max})^2$$

for each $r = 1, \ldots, k$. These values are informative about $R_{\boldsymbol{x}_i}$. For example, if $Q_r^2 = 0$, the set $\Omega_{\boldsymbol{x}_i}$ does not contain the image of $R_{\boldsymbol{x}_i}$. Whenever, $Q_r^2 > 0$, the set $\Omega_{\boldsymbol{x}_i}$ does contains the image of $R_{\boldsymbol{x}_i}$ and $Q_r^2 \geq d^2$ for any $r = 1, \ldots, k$. Thus, this step finds the sets which contain the R_z set so now all that needs to be done is to pick out the set most resembling R_z.

5) Now each d_r^2 is estimated. If $Q_r^2 = 0$, then $d_r^2 = \chi^2_{k+1,\alpha}$ which is the typical value using the Scheffé bounds. This reflects that fact that if a region of $R_{\boldsymbol{x}_i}$ is not restricted then the typical Scheffé bounds are the most appropriate. Whenever $Q_r^2 > 0$, then Q_r^2 is the largest d^2 that allows $\Omega_{\boldsymbol{z}_i}$ to contain R_z. In this case, let $B^2 = (1+Q_r^2)^{-1}$. This may be considered a measure of the size of the constraint region. Then the value of d_r^2 can be given by a table [6] or found by computer. Specifically, this value is found by finding the solution to the following equality:

$$P(E_{r,s}(b, d^2) + P(E_{s,r}(a, d^2) = P(a\chi_r + b\chi_s)^2 \leq d^2) - P(\chi_p^2 \leq d^2)$$

where $E_{r,s}(b, d^2)$ and $E_{s,r}(a, d^2)$ are as defined previously. This is a very tedious calculation and requires a computer to provide a solution.

6) Finally, let $d_S^2 =^{min}_{\ r} \{d_r^2\}$. This is the value that bounds the GLM when applied to the formula. Alternatively, one may choose, in step 5) the small of all Q_r^2

which also yields the smallest d^2.

$$\boldsymbol{x'}\hat{\boldsymbol{\beta}} \pm |d_S|(\boldsymbol{x'F^{-1}x})^{1/2}\ \forall \boldsymbol{x} \in R_{\boldsymbol{x}_i}$$

When the domain of interest for the predictor variable is continuous this algorithm may be followed explicitely for any set of restrictions given by R_x. However, when estimating linear combinations of the parameters, a slight change must be made. Now the limits on the domain are not given by R_x but are rectangular regions defined by the c_i's. For example, when considering the simultaneous estimation of the vector of parameters $(\beta_1, \ldots, \beta_k)$, our contrast matrix is a sequence of 0's and 1's as described in chapter 3. Since we are trying to estimate the actual βi's (i=1,...,k) instead of $\boldsymbol{X}'\beta$, we need to capture the rectangular region defined by all the 0's and 1's. The easiest way to do this is to shift that rectangular region. For example, in two dimensions, we are trying to capture a rectangular region with the vertices (0,0), (1,0), (0,1), and (1,1). Simply shift this rectangle to have the vertices (1,1), (1,2), (2,1), and (2,2) so that the quadric form $\Omega_{\boldsymbol{x}_i}$ contains the rectangle. This concept may be extended to higher dimensions so that a hyper-rectangle is defined in k dimensions and the calculation of the quantity Q_r^2 is given by (C).

APPENDIX D

SCR Methodology

The SCR1 solution utilized the "tube" formula proposed recently by Naiman [10]. Assume there are d points in n-dimensional Euclidean space of interest. If these points are connected they form a curve. Tube formulas compute the volume of a tube of radius r about this curve. Tube-based intervals utilize the following solution for d, the critical value, when $k = 1$ (k=dimension)

$$\alpha \approx \frac{\kappa_0}{\pi} exp(-d^2/2) + \delta * (1 - \Phi(d))$$

where κ_0 is the volume of the region defined by the points and δ is the Euler-Poincare characteristic of the region.

Then for $k > 1$

$$\begin{aligned} \alpha \approx & \frac{\kappa_0 \delta(\frac{k+1}{2})}{\pi^{\frac{k+1}{2}}}(1 - F_{k+1,\nu}(\frac{d^2}{k+1})) \\ & + \frac{\zeta_0 \delta(\frac{k}{2})}{2\pi^{\frac{k+1}{2}}}(1 - F_{k,\nu}(\frac{d^2}{k})) \\ & + \frac{(\kappa_2 + \zeta_1 + m_0)\delta(\frac{k-1}{2})}{2\pi^{\frac{k+1}{2}}} \\ & (1 - F_{k-1,\nu}(\frac{d^2}{k-1})) \end{aligned}$$

where κ_0 is the volume of the region defined by the points, ζ_0 is the surface area of this region, κ_2 is the curvature of the region, ζ_1 is the curvature of the boundary of this region, and m_0 is the rotation angle. The naive or SCR1 solution directly utilizes these formulas.

The SCR3 solution utilized the bias correction of the Gaussian random field,

$W_n(\mathbf{x})$, via the equality

$$|W_n(\mathbf{x})| = |\mathbf{W}(\mathbf{x})| - \mathbf{p_2}(\mathbf{x}, \mathbf{W_n}(\mathbf{x})). \tag{D.1}$$

The biasing constant is given by

$$\begin{aligned} p_2(x, z) = &- z\{\frac{1}{2}[\kappa_2(\mathbf{x}) - \mathbf{1} + \boldsymbol{\kappa}_{\mathbf{1}}^{\mathbf{2}}(\mathbf{x})] \\ &+ \frac{1}{24}[\kappa_4(\mathbf{x}) + \mathbf{4}\boldsymbol{\kappa}_{\mathbf{1}}(\mathbf{x})\boldsymbol{\kappa}_{\mathbf{3}}(\mathbf{x})](\mathbf{z^2} - \mathbf{3}) \\ &\frac{1}{72}\kappa_3^2(\mathbf{x})(\mathbf{z^4} - \mathbf{10z^2} + \mathbf{15}) = \mathbf{O}(\frac{\mathbf{1}}{\mathbf{n}}) \end{aligned}$$

The moments of $W_n(\mathbf{x})$ are given by the following,

$$\begin{aligned} \kappa_1(\mathbf{x}) &= E[W_n(\mathbf{x})] = \mu_{\mathbf{n}}^{'}(\mathbf{x}) \\ \kappa_2(\mathbf{x}) &= E[W_n(\mathbf{x}) - \mathbf{E}[\mathbf{W_n}(\mathbf{x})]]^{\mathbf{2}} \\ &= 1 + \frac{1}{2}C_1 - \frac{1}{2}C_2 - 3C_3 + \frac{1}{2}C_4 \\ &+ C_6 - \frac{1}{2}C_7 + \frac{7}{4}C_8 \\ \kappa_3(\mathbf{x}) &= E[W_n(\mathbf{x}) - \mathbf{E}[\mathbf{W_n}(\mathbf{x})]]^{\mathbf{3}} \\ \kappa_4(\mathbf{x}) &= E[W_n(\mathbf{x}) - \mathbf{E}[\mathbf{W_n}(\mathbf{x})]]^{\mathbf{4}} - \mathbf{3}\boldsymbol{\kappa}_{\mathbf{2}}^{\mathbf{2}}(\mathbf{x}) \\ &= -9C_3 = 3C_6 + 6C_8 + 3C_9. \end{aligned}$$

The C_i's $(i = 1, \ldots, 9)$ utilized in the above calculations are given in the following equations. Note that the quantities, b^j's, are based on the distribution of the response variable and the estimated parameters. Thus, they depend on whether the usual MLE estimators or alternative pMLE estimators are used. See Appendix B for the description of the b^j's.

$$C_1 = \frac{1}{n^3}\sum_i\sum_j b_i^{(3)} b_j^{(3)} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_i}\rangle \langle \mathbf{s}(\mathbf{x}), \mathbf{u_j}\rangle \langle \mathbf{u_i}, \mathbf{u_j}\rangle^{\mathbf{2}} \tag{D.2}$$

$$C_2 = \frac{1}{n^2}\sum_i b_i^{(4)} b_j^{(3)} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_i}\rangle^{\mathbf{2}} \langle \mathbf{u_i}, \mathbf{u_j}\rangle \tag{D.3}$$

$$C_3 = \frac{1}{n^3}\sum_i\sum_j b_i^{(3)} b_j^{(3)} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_i}\rangle^{\mathbf{2}} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_j}\rangle^{\mathbf{2}} \langle \mathbf{u_i}, \mathbf{u_j}\rangle \tag{D.4}$$

$$C_4 = \frac{1}{n^3}\sum_i\sum_j b_i^{(3)} b_j^{(3)} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_i}\rangle^{\mathbf{2}} \langle \mathbf{u_j}, \mathbf{u_j}\rangle \langle \mathbf{u_i}, \mathbf{u_j}\rangle \tag{D.5}$$

$$C_5 = C_1 \tag{D.6}$$

$$C_6 = \frac{1}{n^2}\sum_i b_i^{(4)} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_i}\rangle^{\mathbf{4}} \tag{D.7}$$

$$C_7 = \frac{1}{n^3}\sum_i\sum_j b_i^{(3)} b_j^{(3)} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_i}\rangle \langle \mathbf{s}(\mathbf{x}), \mathbf{u_j}\rangle^{\mathbf{3}} \langle \mathbf{u_i}, \mathbf{u_i}\rangle \tag{D.8}$$

$$C_8 = \frac{1}{n^3}\sum_i\sum_j b_i^{(3)} b_j^{(3)} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_i}\rangle^{\mathbf{3}} \langle \mathbf{s}(\mathbf{x}), \mathbf{u_j}\rangle^{\mathbf{3}} \tag{D.9}$$

$$C_9 = \frac{1}{n^2}\sum_i [b_i^{(2)}]^2 \langle \mathbf{s}(\mathbf{x}), \mathbf{u_i}\rangle^{\mathbf{4}}. \tag{D.10}$$

APPENDIX E

Fortran Code: Piegorsch-Casella PMLE for the Mean Response

```
USE MSIMSL

INTEGER PINT,REPS

DOUBLE PRECISION P,R

PARAMETER(PINT=2,REPS=5000, LDA=PINT, LDEVEC=PINT,LDAINV=PINT)

DOUBLE PRECISION RANGEX(2),VER(PINT,2**(PINT-1)),

DVER(PINT,2**(PINT-1)),ZMAX(2**(PINT-1)),ZMIN(2**(PINT-1))

DOUBLE PRECISION S,A2, B2 ,Q2,C, Dp,Z(2),ECL(REPS),

ECL2(REPS),N,MEANECL,MEANECL2,LEFT,RIGHT,MUTEMP

DOUBLE PRECISION BETAP(2),XMAT(:,:),LINEAR(:),LINEARP,U(:),

MU(:),MUHAT(:),FACT

REAL MEANTEMP

INTEGER PTEMP(:),DONE,ITER,Y2(:)

ALLOCATABLE XMAT,LINEAR,U,MU,MUHAT,PTEMP,Y2

INTEGER L,RT,ISEED,STEP,NINT,CONF,CONF2,ITER2

DOUBLE PRECISION ERS

DOUBLE PRECISION CHIP

DOUBLE PRECISION BETAF(PINT),DELTAF(PINT),LOGLIK,LOGLIKOLD,

DET1,DET2,FISH(PINT,PINT)

DOUBLE PRECISION USTAR(PINT),LOGLIK1,LOGLIK2,FACU(PINT,PINT),

MX,BIAS(PINT),MEANBIAS(PINT)

COMMON S, R, P, A2, B2, C
```

```
      INTEGER IRULE,K,I,J,RINT,IPVT(PINT)
      DOUBLE PRECISION LOW(2),UP(2),ERRABS,ERREL,TEMP,COVLIN,TEMP2(2),XSUM
      DOUBLE PRECISION ERSRESULT1,ERSRESULT2,ERSTEMP,CHIPTEMP,FX,FX1,C0,C1,C2
      DOUBLE PRECISION ERREST,CHIPRESULT1
      DOUBLE PRECISION CHIPRESULT2
      EXTERNAL ERS, CHIP
      PARAMETER(Delta=1.0D-3,Epsilon=1.0D-6,Max=1000,Small=1.0D-6)
      DOUBLE PRECISION EPS,MUL(2)
      PARAMETER  (EPS=1.0D-2, ICEN=0, IFIX=0, IFRQ=0, ILT=0, INIT=0,INTCEP=1)
      DOUBLE PRECISION CASE(:,:), COEF(:,:),COV(:,:),H2(:,:)
      DOUBLE PRECISION H3(:,:),HAT(:,:),WGT(:,:)
      DOUBLE PRECISION X(:,:), RESULTS(7),INTER,VTEMP(PINT,PINT),X2(:,:)
      DOUBLE PRECISION F(PINT,PINT),D(LDA),UMAT(LDEVEC,PINT),
      D2(PINT,PINT),D3(PINT,PINT)
      ALLOCATABLE CASE,COEF,COV,X,X2,H2,H3,HAT,WGT

      OPEN (UNIT=8, FILE=’C:/Results/pc results/Dferror.txt’)
      OPEN (UNIT=9, FILE=’C:/Results/pc results/RESULTS.txt’)
      OPEN (UNIT=10, FILE=’C:/Results/pc results/RELERR.txt’)
      OPEN (UNIT=11, FILE=’C:/Results/pc results/ECL.txt’)
      OPEN (UNIT=12, FILE=’C:/Results/pc results/DATA.txt’)

  !   DEFINE R, S, AND P AND INTEGER VERSIONS FOR LATER
      P=2.0D0
      R=1.0D0
      S=P-R
      RINT=INT(R)
```

```
    ISEED=34271
    CALL RNSET(ISEED)
    C10=P*DFIN(9.5D-1,P,1.0D6)
!   LOOP FOR MODEL TYPE 1=LOGIT, 2=POISSON, 3=PROBIT
    DO 2 B=1,2
    IF (B.EQ.1) THEN
    MODEL=3
    ELSEIF (B.EQ.2) THEN
    MODEL=0
    ELSEIF (B.EQ.3) THEN
    MODEL=4
    ENDIF
!   LOOP FOR BETA PARAMETERS 1=(-1,.5), 2=(0,1), 3=(2,4),
4=(-.25,-.5)
    DO 3 H=1,4
    IF (H.EQ.1) THEN
    BETAP(1)=-1.0D0
    BETAP(2)=5.0D-1
    ELSEIF (H.EQ.2) THEN
    BETAP(1)=0.0D0
    BETAP(2)=1.0D0
    ELSEIF (H.EQ.3) THEN
    BETAP(1)=2.0D0
    BETAP(2)=4.0D0
    ELSEIF (H.EQ.4) THEN
    BETAP(1)=-2.5D-1
    BETAP(2)=-5.0D-1
```

```
      ENDIF
!     LOOP FOR N (N=10,25,50,100,200)
      DO 4 W=1,3
      IF (W.EQ.1) THEN
      N=1.0D1
      ELSEIF (W.EQ.2) THEN
      N=2.5D1
      ELSEIF (W.EQ.3) THEN
      N=5.0D1
      ELSEIF (W.EQ.4) THEN
      N=1.0D2
      ELSEIF (W.EQ.5) THEN
      N=2.0D2
      ENDIF
      NINT=INT(N)
      LDCASE=NINT
      LDCOEF=PINT
      LDCOV=PINT
      LDX=NINT
      NCOL=PINT+1
      NOBS=NINT
      ALLOCATE (Y2(NINT),CASE(LDCASE,5),COEF(LDCOEF,4),
      COV(LDCOV,4),X(LDX,NCOL))
      ALLOCATE (XMAT(NINT,2),LINEAR(NINT),PTEMP(NINT),
      U(NINT),MU(NINT),MUHAT(NINT))
      ALLOCATE (X2(LDX,NCOL),H2(PINT,NINT),H3(NINT,PINT),
      HAT(NINT,NINT),WGT(NINT,NINT))
```

```
!   LOOP FOR RESTRICTED DOMAIN TYPES 1=UNRESTRICTED, 2=WIDE,
3=NARROW
    DO 5 G=2,3
    IF (G.EQ.1) THEN
    MUL(1)=1.0D-4
    MUL(2)=1.0D0-1.0D-4
    ELSEIF (G.EQ.2) THEN
    MUL(1)=1.0D-1
    MUL(2)=9.0D-1
    ELSEIF (G.EQ.3) THEN
    MUL(1)=2.5D-1
    MUL(2)=7.5D-1
    ENDIF
    IF (MODEL.EQ.3) THEN
    RANGEX=(DLOG(MUL/(1-MUL))-BETAP(1))/BETAP(2)
    ELSEIF (MODEL.EQ.0) THEN
    RANGEX(2)=0.0D0
    RANGEX(1)=1.0D0
    ELSEIF (MODEL.EQ.4) THEN
    K=1
    DO WHILE (K.LE.2)
    RANGEX(K)=(DNORDF(MUL(K))-BETAP(1))/BETAP(2)
    K=K+1
    ENDDO
    ENDIF
!   LOOP FOR DOMAIN TYPES 1=EQUALLY SPACED, 2=ONE CLUSTER
    DO 6 E=1,1
    ECL=0.0D0
    ECL2=0.0D0
    MEANBIAS=0.0D0
    DO 7 A=1,REPS
    BETAF=0.0D0
    IF (E.EQ.1) THEN
    IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
    K=1
    EQSPACE:DO WHILE (K.LT.N)
    INTER=(RANGEX(2)-RANGEX(1))/(N-1)
    X(K,1)=RANGEX(1)+INTER*(K-1)+1.0D-8
    X(K+1,1)=RANGEX(1)+INTER*K+1.0D-8
    K=K+1
    ENDDO EQSPACE
    ELSEIF (MODEL.EQ.0) THEN
    IF (G.EQ.2) THEN
    CALL DRNUN(NINT,X(:,1))
    ELSEIF (G.EQ.3) THEN
    CALL DRNUN(NINT,U)
    X(:,1)=5.0D-1*U
    ENDIF
    ENDIF
    ELSEIF (E.EQ.2) THEN
    NCLUS=NINT*0.2
!   GENERATE N-NUM CLUSTER EQUALLY SPACED PTS
    CALL DRNUN(NINT-NCLUS,U(1:NINT-NCLUS))
    MUHAT(1:NINT-NCLUS)=MUL(1)+U(1:NINT-NCLUS)*
    (MUL(2)-MUL(1))
!   GENEATE A POINT IN THE RANGE TO CLUSTER PTS ABOUT
    UTEMP=DRNUNF()
    MUTEMP=MUL(1)+UTEMP*(MUL(2)-MUL(1))
    CALL DRNUN(NCLUS,U(1:NCLUS))
!   CLUSTER ABOUT POINT WITH WIDTH=0.2
    MUHAT(NINT-NCLUS+1:NINT)=(MUTEMP-1.0D-1)+
    U(1:NCLUS)*2.0D-1
!   FIX IF CLUSTERED POINTS FALL OUTSIDE RANGE OF X
    DO 8 Y=1,NCLUS
    IF ((MUHAT(NINT-NCLUS+Y).LT.MUL(1))) THEN
    MUHAT(NINT-NCLUS+Y)=MUL(1)
    ELSEIF (MUHAT(NINT-NCLUS+Y).GT.MUL(2)) THEN
    MUHAT(NINT-NCLUS+Y)=MUL(2)
    ENDIF
    8 CONTINUE
    IF (MODEL.EQ.3) THEN
    X(:,1)=(DLOG(MUHAT/(1.0D0-MUHAT)+Small)-
    BETAP(1))/BETAP(2)
    ELSEIF (MODEL.EQ.0) THEN
    X(:,1)=(DLOG(MUHAT+Small)-BETAP(1))/BETAP(2)
    ELSEIF (MODEL.EQ.4) THEN
    K=1
    DO WHILE (K.LE.2)
    RANGEX(K)=(DNORDF(MUL(K))-BETAP(1))
    /BETAP(2)
    K=K+1
    ENDDO
    ENDIF
    ENDIF
    XMAT(:,1)=1.0D0
    XMAT(:,2)=X(:,1)
 !NOTE: TEMP VALUE FOR LINEAR - LATER WILL BE
    USING EST BETAS!!!
    LINEAR=MATMUL(XMAT,BETAP)
!   SIMULATE Y NOW !   SIMULATE UNIFORM(0,1) RV'S
    CALL DRNUN(NINT,U)
!   SIMULATE RESPONSE Y
    DO 10 I=1,NINT
    IF (MODEL.EQ.3) THEN
    MU(I)=DEXP(LINEAR(I))/(1+DEXP(LINEAR(I)))
    ELSEIF (MODEL.EQ.0) THEN
    MU(I)=DEXP(LINEAR(I))
    ELSEIF (MODEL.EQ.4) THEN
    MU(I)=DNORDF(LINEAR(I))
    ENDIF
!   THE 3RD COLUMN OF X IS Y
    IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
    IF (U(I).LT.MU(I)) THEN
    X(I,3)=1.0D0
    ELSE
    X(I,3)=0.0D0
    ENDIF
    ELSE
    RANGEX(1)=DMIN1(X(I,1),RANGEX(1))

    RANGEX(2)=DMAX1(X(I,1),RANGEX(2))

    MEANTEMP=REAL(MU(I))

    CALL RNPOI(NINT,MEANTEMP,PTEMP)

    X(I,3)=DBLE(PTEMP(I))

    ENDIF

    10 CONTINUE

!PMLE CALCULATIONS START HERE

    IF (MODEL.EQ.3) THEN

    X(:,2)=1.0D0

    ELSEIF (MODEL.EQ.0) THEN

    X(:,2)=MU

    ENDIF

    WGT=0.0D0

    LOGLIK1=0.0D0

    LOGLIK2=0.0D0

    MUSUM1=0.0D0

    MUSUM2=0.0D0

    XSUM=SUM(X(:,3))/N

    Y2=INT(X(:,3))

    IF (MODEL.EQ.3) THEN

    BETAF(1)=DLOG((XSUM)/((1.0D0-XSUM+Small))+Small)

    ELSEIF (MODEL.EQ.0) THEN

    BETAF(1)=DLOG(XSUM+Small)

    ENDIF

    BETAF(2)=0.0D0

    LINEAR=MATMUL(XMAT,BETAF)
```

```
DO 55 I=1,NINT
IF (MODEL.EQ.3) THEN
MUHAT(I)=DEXP(LINEAR(I))/(1+DEXP(LINEAR(I)))
WGT(I,I)=DSQRT(MUHAT(I)*(1.0D0-MUHAT(I)))
IF (X(I,3).EQ.1.0D0) THEN
LOGLIK1=LOGLIK1+(DLOG(MUHAT(I)+Small))
ELSE
LOGLIK2=LOGLIK2+(DLOG(1-MUHAT(I)+Small))
ENDIF
ELSEIF (MODEL.EQ.0) THEN
MUHAT(I)=DEXP(LINEAR(I))
WGT(I,I)=DSQRT(MUHAT(I))
IF (Y2(I).GT.169) THEN
Y2(I)=169
ENDIF
FACT=DFAC(Y2(I))
LOGLIK1=LOGLIK1-MUHAT(I)+X(I,3)*
DLOG(MUHAT(I)+Small)-DLOG(FACT)
LOGLIK2=0.0D0
ELSEIF (MODEL.EQ.4) THEN
MUHAT(I)=DNORDF(LINEAR(I))
WGT(I,I)=DSQRT(MUHAT(I)*(1.0D0-MUHAT(I)))
ENDIF
55 CONTINUE
LOGLIK=LOGLIK1+LOGLIK2
IPRINT=0
TEMP3=0
```

```
    H2=MATMUL(TRANSPOSE(XMAT),WGT)

    FISH=MATMUL(H2,TRANSPOSE(H2))

    CALL DLFTSF (PINT,FISH , LDA, FACU, LDA, IPVT)

!   Compute the determinant

    CALL DLFDSF (PINT, FACU, LDA, IPVT, DET1, DET2)

    LOGLIK=LOGLIK+5.0D-1*(DET1*1.0D1**DET2)

    ITER=0

    LOGLIKOLD=0.0D0

    ITER=0

    DO WHILE (ITER.LT.25)

    ITER=ITER+1

    H2=MATMUL(TRANSPOSE(XMAT),WGT)

    FISH=MATMUL(H2,TRANSPOSE(H2))

    CALL DLINRG(PINT,FISH,LDA,F,LDAINV)

    H3=MATMUL(TRANSPOSE(H2),F)

    HAT=MATMUL(H3,H2)

    DO 66 O=1,NINT

    IF (MODEL.EQ.3) THEN

    X2(O,3)=X(O,3)-MUHAT(O)+HAT(O,O)*

   (5.0D-1-MUHAT(O))

    ELSEIF (MODEL.EQ.0) THEN

    X2(O,3)=X(O,3)-MUHAT(O)+HAT(O,O)/2.0D0

    ENDIF

    66 CONTINUE

   USTAR=MATMUL(TRANSPOSE(XMAT),X2(:,3))

    DELTAF=MATMUL(F,USTAR)

    MX=DMAX1(DABS(DELTAF(1)),DABS(DELTAF(2)))/10
    IF (MX.GT.1.0D0) THEN
    DELTAF=DELTAF/MX
    ENDIF
    BETAF=BETAF+DELTAF
    LINEAR=MATMUL(XMAT,BETAF)
    LOGLIKOLD=LOGLIK
! DO HALF-STEPS
    DONE=0
    ITER2=0
    DO WHILE (DONE.EQ.0)
    ITER2=ITER2+1
    IF (MODEL.EQ.3) THEN
    DO 65 I=1,NINT
    MUHAT(I)=1.0D0/(1.0D0+DEXP(-LINEAR(I)))
    IF (X(I,3).EQ.1.0D0) THEN
    LOGLIK1=LOGLIK1+(DLOG(MUHAT(I)+Small))
    ELSE
    LOGLIK2=LOGLIK2+(DLOG(1-MUHAT(I)+Small))
    ENDIF
    65 CONTINUE
    ELSEIF (MODEL.EQ.0) THEN
   !FACT=DFAC(Y2(I))
    DO 67 I=1,NINT
    MUHAT(I)=DEXP(LINEAR(I))
    LOGLIK1=LOGLIK1-MUHAT(I)+X(I,3)*
    DLOG(MUHAT(I)+Small)-DLOG(FACT)
    LOGLIK2=0.0D0
    67 CONTINUE
    ENDIF
    LOGLIK=LOGLIK1+LOGLIK2
    H2=MATMUL(TRANSPOSE(XMAT),WGT)
   FISH=MATMUL(H2,XMAT)
    CALL DLFTSF (PINT,FISH , LDA, FACU, LDA, IPVT)
!                                         Compute the determinant
   CALL DLFDSF (PINT, FACU, LDA, IPVT, DET1, DET2)
    LOGLIK=LOGLIK+5.0D-1*(DET1*1.0D1**DET2)
    IF ((LOGLIK.GT.LOGLIKOLD).OR.(ITER2.EQ.5)) THEN
    DONE=1
   !ITER=30
    ELSE
    BETAF=BETAF-DELTAF*2.0D0**(-I)
    ENDIF
    ENDDO
    IF (SUM(DABS(DELTAF)).LT.1.0D-4) THEN
    ITER=250
    ENDIF
    ENDDO
    LINEAR=MATMUL(XMAT,BETAF)
    BIAS=BETAF-BETAP
   !ENDDO
!   RETRIVE VERTICES FROM RECTANGULAR RESTRICTIONS ON X
    VER(1,:)=1.0D0
    L=1
    DO 11 I=2,PINT
    STEP=2**(PINT-I)

    DO 12 J=1,L,2*STEP

    VER(I,J:STEP+J-1)=RANGEX(1)

    VER(I,STEP+J:2*STEP+J-1)=RANGEX(2)

    12 CONTINUE

    L=L+STEP

!   CALCULATE Q2 FROM P.867-868 OF CS !   FIRST CALCULATE

DIAGONALIZED VERTICES

    CALL DEVCSF (PINT, FISH, LDA, D, UMAT, LDEVEC)

    DO 16 K=1,PINT

    DO 17 L=1,PINT

    IF (K.EQ.L) THEN

    D2(K,L)=D(K)**5.0D-1

    ELSE

    D2(K,L)=0.0D0

    ENDIF

    17 CONTINUE

    16 CONTINUE

    CALL DLINRG(PINT,D2,LDA,D3,LDAINV)

    VTEMP=MATMUL(D3,TRANSPOSE(UMAT))

    DVER=MATMUL(VTEMP,VER)

    IF (PINT.EQ.2) THEN

    ZMAX(2)=DVER(2,2)

    ZMIN(1)=DVER(2,1)

    IF ((ZMAX(2).GT.0.0D0).AND.(ZMIN(1).LT.0.0D0))

    THEN

    ZMIN(1)=0.0D0
```

```
  ELSE
  ZMIN(1)=DMIN1(DABS(DVER(1,1)),
  DABS(DVER(1,2)),DABS(DVER(2,1)),
  DABS(DVER(2,2)))
  ENDIF
  ZMAX(2)=DMAX1(DABS(DVER(1,1)),
 DABS(DVER(1,2)),DABS(DVER(2,1)),
 DABS(DVER(2,2)))
  ELSE
  DO 13 K=1,(2**(PINT-1)-1)
  Z(I-1)=DVER(I-1,K)
  Z(I)=DVER(I,K)
  ZMAX(K)=(DMAX1(Z(I-1),Z(I)))
  TEMP=(DMIN1(Z(I-1),Z(I)))
IF ((TEMP.LT.0.0D0).AND.
  (ZMAX(K).GT.0.0D0)) THEN
  ZMIN(K)=0.0D0
  ELSE
 ZMIN(K)=DABS(TEMP)
  ENDIF
  ZMAX(K)=DMAX1(DABS(ZMAX(K)),
  DABS(Z(I)))
  13 CONTINUE
  ENDIF
  11 CONTINUE
  RT=1
  Q2=0.0D0
```

```
    DO WHILE (RT.LT.(2**(PINT-1)))

    Q2TEMP=SUM(ZMIN(1:RT)**2.0D0)/

    SUM(ZMAX(RT+1:2**(PINT-1))**2.0D0)

    IF (Q2TEMP.EQ.0.0D0) THEN

    RT=RT+1

    Q2=Q2TEMP

    RT=2**(PINT-1)

    ENDIF

    ENDDO

    B2=(1.0D0+Q2)**(-1.0D0)

    A2=1-B2

!   STARTING VALUES FOR THE SECANT METHOD

    C0=1.96D0**2.0D0

    C1=C10

!   DEFINE UPPER AND LOWER LIMITS FOR ERS INTEGRAL

    LOW(1)=C0

    LOW(2)=C1

!   BE CAREFUL WHEN B2=0

    IF (B2.GT.1.0D-6) THEN

    UP(1)=(C0)/(B2)

    UP(2)=(C1)/(B2)

    ELSE

    UP(1)=9.99D9

    UP(2)=9.99D9

    END IF

    C=DSQRT(C0)

    ERRABS=1.0d-6
    ERREL=1.0d-6
    IRULE=2
!   Call 1st ERS integral
    IF (B2.EQ.0.0D0) THEN
    CALL DQDAGI(ERS,LOW(1),1,ERRABS,ERREL,
    ERSRESULT1,ERREST)
    ELSE IF (B2.EQ.1.0D0) THEN
    ERSRESULT1=0.0D0
    ELSE
    CALL DQDAG(ERS,LOW(1),UP(1),ERRABS,ERREL,
    IRULE,ERSRESULT1,ERREST)
    END IF
!  Call 2nd ERS integral
   C=DSQRT(C1)
    IF (B2.EQ.0.0D0) THEN
    CALL DQDAGI(ERS,LOW(2),1,ERRABS,ERREL,
    ERSRESULT2,ERREST)
    ELSE IF (B2.EQ.1.0D0) THEN
    ERSRESULT2=0.0D0
    ELSE
    CALL DQDAG(ERS,LOW(2),UP(2),ERRABS,ERREL,
   IRULE,ERSRESULT2,ERREST)
    END IF
!   SET LIMITS FOR CHIP INTEGRAL
    LOW(1)=0.0D0
    LOW(2)=0.0D0
    UP(1)=C0
    UP(2)=C1
!   Call the 1st chi square (p) integral
    CALL DQDAG(CHIP,LOW(1),UP(1),ERRABS,ERREL,
    IRULE,CHIPRESULT1,ERREST)
!   Call the 2nd chi square (p) integral
    CALL DQDAG(CHIP,LOW(2),UP(2),ERRABS,ERREL,
    IRULE,CHIPRESULT2,ERREST)
!   COMPUTE FIRST TWO VALUES OF THE FUNCTION F - SHOULD HAVE 0.95
BETWEEN THESE
    FX=ERSRESULT1+CHIPRESULT1
    FX1=ERSRESULT2+CHIPRESULT2
!   K COUNTS HOW MANY ITERATIONS
    K=0
   AbsErr=1.0d0
!   BEGIN LOOP TO OPTIMIZE F FOR C2 UNTIL ABSERR < EPS
    SECANTLOOP: DO WHILE ((K.LT.Max).AND.
             (AbsErr.GT.Epsilon))
!   CALCULATES NEW ITERATION OF C
    Df=(FX1-FX)/(C1-C0)
    IF (Df.EQ.0) THEN
     WRITE (8,*) Df
   ELSE
!   CALCULATES NEW ITERATION OF C
   Dp=(FX1-9.5D-1)/Df
    C2=(C1-Dp)**1.0D0
    ENDIF
!   CALCULATE NEW FX FOR C2
```

```
    LOW(1)=C2
    IF (B2.NE.0.0D0) THEN
    UP(1)=(C2)/(B2)
    ENDIF
    ERRABS=1.0d-6
    ERREL=1.0d-6
    IRULE=2
!   Call ERS integral
    C=DSQRT(C2)
   IF (B2.EQ.0.0D0) THEN
    CALL DQDAGI(ERS,LOW(1),1,ERRABS,
      ERREL,ERSTEMP,ERREST)
    ELSEIF (B2.EQ.1.0D0) THEN
    ERSTEMP=0.0D0
    ELSE
    CALL DQDAG(ERS,LOW(1),UP(1),ERRABS,
               ERREL,IRULE,ERSTEMP,ERREST)
    ENDIF
    LOW(1)=0.0D0
    UP(1)=C2
!   Call the chi square (p) integral
    CALL DQDAG(CHIP,LOW(1),UP(1),ERRABS,
            ERREL,IRULE,CHIPTEMP,ERREST)
!   CALCULATES THE NEW F VALUE (SOMEWHERE BETWEEN FX AND FX1)
    TEMP=ERSTEMP+CHIPTEMP
!   CALCULATE THE ERRORS
   AbsErr=DABS(TEMP-9.5D-1)
```

```
    RelErr=DABS(Dp)/(DABS(C2)+Small)
!   RECORDS SMALL RELERR
    IF (RelErr.GT.Delta) THEN
     WRITE(10,*) K,RelErr
    ENDIF
!   IF TEMP < 0.95 THEN OVERESTIMATED C2
    IF (TEMP.LT.9.5D-1) THEN
     C0=C2
     FX=TEMP
    ELSE
!   IF TEMP >= 0.95 THEN C2 UNDERESTIMATED
     C1=C2
     FX1=TEMP
    ENDIF
    K=K+1
    ENDDO SECANTLOOP
    CONF=0
    CONF2=0
    DO 60 I=1,NINT
   LINEARP=DDOT(PINT,XMAT(I,:),1,BETAP,1)
    TEMP2=MATMUL(XMAT(I,:),F)
    COVLIN=DDOT(PINT,TEMP2,1,XMAT(I,:),1)
            +F(2,2)*XMAT(I,2)**2.0D0)
!   RECORD WHEN BOUNDS ARE ESTIMATED CORRECTLY
    LEFT=(LINEAR(I)-LINEARP)**2.0D0
    RIGHT=C2*((COVLIN))
    IF (LEFT.LE.RIGHT) THEN
```

```
CONF=CONF+1
 ENDIF
 LEFT=(LINEAR(I)-LINEARP)**2.0D0
 RIGHT=C10*((COVLIN))
 IF (LEFT.LE.RIGHT) THEN
  CONF2=CONF2+1
 ENDIF
 60 CONTINUE
 ECL(A)=DBLE(CONF)/N
 ECL2(A)=DBLE(CONF2)/N
 MEANECL=SUM(ECL)/REPS
 MEANECL2=SUM(ECL2)/REPS
 MEANBIAS=(MEANBIAS+BIAS)/REPS
 RESULTS(1)=MODEL
 RESULTS(2)=H
 RESULTS(3)=NINT
 RESULTS(4)=G
 RESULTS(5)=E
 RESULTS(6)=MEANECL
 RESULTS(7)=MEANECL2

 7 CONTINUE
 WRITE(11,*) MEANECL,MEANECL2,MEANBIAS
 WRITE(9,*) RESULTS
 6 CONTINUE
 5 CONTINUE
 DEALLOCATE (Y2,CASE,COEF,COV,X)
```

```
   DEALLOCATE (XMAT,LINEAR,U,MU,MUHAT,PTEMP)

    DEALLOCATE (X2,H2,H3,HAT,WGT)

    4 CONTINUE

    3 CONTINUE

    2 CONTINUE


    STOP

    END


!   Now start defining the functions !   used in the above main

program !   Define the ERS function

    DOUBLE PRECISION FUNCTION ERS(T)

    DOUBLE PRECISION PART1, PART2, PART3, DFD, DFN

    COMMON S,R,P,A2,B2,C

    DOUBLE PRECISION T,S,R,P,A2,B2,C, TEMP, TEMP2, TEMP3, TEMP4,TEMP5,TEMP6

    DOUBLE PRECISION DGAMMA, DFDF

    !

    DFD=S

    DFN=R

    TEMP4=C*(T-C**2.0D0)**5.0D-1

    TEMP5=DSQRT(A2)*DSQRT(B2)*T

    TEMP6=A2*T-C**2.0D0

    PART1=DFDF((S/R)*((TEMP4-TEMP5)/TEMP6)**2.d0,DFN,DFD)

    PART2=T**((P/2.d0)-1.d0)

    TEMP=P/2.0D0

    TEMP2=2.0D0**TEMP

    TEMP3=DGAMMA(TEMP)*TEMP2
    PART3=DEXP(-T/2.d0)/TEMP3

    ERS=PART1*(PART2*PART3)


    RETURN

    END



!   Define the chi-square function

    DOUBLE PRECISION FUNCTION CHIP(T)

    DOUBLE PRECISION PART1, PART2, PART3

    COMMON S,R,P,A2,B2,C

    DOUBLE PRECISION T,S,R,P

    DOUBLE PRECISION DGAMMA

    PART1=T**((P/2.d0)-1.d0)

    PART2=DEXP(-T/2.d0)

    PART3=DGAMMA(P/2.d0)*2.d0**(P/2.d0)

    CHIP=(PART1*PART2)/(PART3)

    RETURN

    END
```

APPENDIX F

Fortran Code: SCR PMLE for the Mean Response

```
USE MSIMSL
INTEGER N,P,M,INCX,LDA,LDR,LDAINV,VER,REPS
DOUBLE PRECISION Small
PARAMETER(N=100,P=2,M=41,REPS=5000,LDA=P,LDR=P,LDFAC=P,
LDAINV=P,Small=1.0D-6)
REAL MEANTEMP
DOUBLE PRECISION B2(N),B3(N),B4(N),B(P,P),A(P,P)
DOUBLE PRECISION AINV(P,P),LO(N),HI(N)
DOUBLE PRECISION SU(N,N),S(P,N),UI(P,N),U(N,N),K1,UITEMP(P)
DOUBLE PRECISION CRIT,C(9),KAP(5,N),KAQ(5,N),MAX_P2(N)
DOUBLE PRECISION BT(P,P),TEMP2(P),TEMP3,TEMP4,BTINV(P,P)
DOUBLE PRECISION P2,Q2,F,P3,N2,XMAT(N,P),INFO(P,P)
DOUBLE PRECISION UCRIT(N),K0,CRITLO,CRITHI,FX,FX1,ALPHA,MAX,ABSERR
DOUBLE PRECISION DF,DP,TEMPFX,LINEAR(N),LINEARP(N),VAR(N),MU(N),K3,TOL
DOUBLE PRECISION XLO,XHI,YLO,YHI,PI,X1,X2,Y1,Y2,XT,YT,CRITHI0
DOUBLE PRECISION FIRST(P),SECOND(P),DIF(P),NORMDIF,T(P,M),INTER,RESULTS(6)
DOUBLE PRECISION SUM1,SUM2,MEAN1,MEAN2,NORMC,UNIF(N),RANGEX(2),
ECL(REPS),MEANECL
EXTERNAL P2,Q2,F
INTEGER I,J,K,V,IRANK,CONF,PTEMP(N)
LOGICAL PIVOT
```

```
    DOUBLE PRECISION H2(P,N),H3(N,P),HAT(N,N)
    INTEGER ITER,ITER2,YF(N)
    DOUBLE PRECISION X(N,P),XF(N,P),WGT(N,N)
    DOUBLE PRECISION BETAP(P),MUL(2),MUHAT(N),UTEMP,MUTEMP,FACT
    DOUBLE PRECISION BETAF(P),LOGLIK1,LOGLIK2,MUSUM1,MUSUM2,XSUM
    DOUBLE PRECISION FISH(P,P),USTAR(P),DELTAF(P),FCOV(P,P)
    DOUBLE PRECISION MX,LOGLIK,LOGLIKOLD,LEFT,RIGHT
    INTEGER COUNT,NCLUS

    COMMON KAP,KAQ,P3,N2,C,C2
    OPEN (UNIT=9, FILE='C:/Results/RESULTS_SCR100_bothmean.txt')
    OPEN (UNIT=10, FILE='C:/Results/ECL100_bothmean.txt')

    MAX=500
    EPSILON=1.0D-6
    ALPHA=5.0D-2
    PI=3.1415926535897932D0
    P3=DBLE(P)
    N2=DBLE(N)
    K0=0.0D0
    RINT=INT(R)
    CALL RNSET(34271)
    COUNT=0
!   THIS IS SCHEFFE SOLUTION
    CRITHI0=(P3*DFIN(9.5D-1,P3,1.0D6))
!   LOOP FOR MODEL TYPE 1=LOGIT, 2=POISSON, 3=PROBIT
    DO 2 Z=1,2
```

```
        IF (Z.EQ.1) THEN
            MODEL=3
        ELSEIF (Z.EQ.2) THEN
            MODEL=0
        ELSEIF (Z.EQ.3) THEN
            MODEL=4
        ENDIF
!   LOOP FOR BETA PARAMETERS 1=(-1,.5), 2=(0,1), 3=(2,4),
4=(-.25,-.5)
        DO 3 H=1,4
            IF (H.EQ.1) THEN
                BETAP(1)=-1.0D0
                BETAP(2)=5.0D-1
            ELSEIF (H.EQ.2) THEN
                BETAP(1)=0.0D0
                BETAP(2)=1.0D0
            ELSEIF (H.EQ.3) THEN
                BETAP(1)=2.0D0
                BETAP(2)=4.0D0
            ELSEIF (H.EQ.4) THEN
                BETAP(1)=-2.5D-1
                BETAP(2)=-5.0D-1
            ENDIF
!   LOOP FOR RESTRICTED DOMAIN TYPES 1=UNRESTRICTED, 2=WIDE,
3=NARROW
            DO 5 G=2,3
                IF (G.EQ.1) THEN
```

```
                    MUL(1)=1.0D-8
                    MUL(2)=1.0D0-1.0D-8
                ELSEIF (G.EQ.2) THEN
                    MUL(1)=1.0D-1
                    MUL(2)=9.0D-1
                ELSEIF (G.EQ.3) THEN
                    MUL(1)=2.5D-1
                    MUL(2)=7.5D-1
                ENDIF
                IF (MODEL.EQ.3) THEN
                    RANGEX=(DLOG(MUL/(1-MUL))
                    -BETAP(1))/BETAP(2)
                ELSEIF (MODEL.EQ.0) THEN
                    RANGEX(2)=0.0D0
                    RANGEX(1)=1.0D0
                ELSEIF (MODEL.EQ.4) THEN
                    K=1
                    DO WHILE (K.LE.2)
                        RANGEX(K)=(DNORDF(MUL(K))
                        -BETAP(1))/BETAP(2)
                        K=K+1
                    ENDDO
                ENDIF

!   LOOP FOR DOMAIN TYPES 1=EQUALLY SPACED, 2=ONE CLUSTER
                DO 6 D=1,1
                ECL=0.0D0
                DO 7 R=1,4
                    VER=R
!   COUNTS EACH COMBINATION OF SIMULATION SPECS (THERE ARE
3*4*2*3=72 COMBINATIONS)
                DO 8 L=1,REPS
                    DONE=0
                    CONF=0
                    DO WHILE (DONE.EQ.0)
                    CALL RNSET(0)
                    IF (D.EQ.1) THEN
                !   GENERATE EQUALLY SPACED X'S
                    IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
                     K=1
                    EQSPACE:DO WHILE (K.LT.N)
                     INTER=(RANGEX(2)-RANGEX(1))/(N-1)
                     X(K,1)=RANGEX(1)+INTER*(K-1)+1.0D-8
                     X(K+1,1)=RANGEX(1)+INTER*K+1.0D-8
                     K=K+1
                     ENDDO EQSPACE
                    ELSEIF (MODEL.EQ.0) THEN
                     IF (G.EQ.2) THEN
                    CALL DRNUN(N,X(:,1))
                    ELSEIF (G.EQ.3) THEN
                    CALL DRNUN(N,UNIF)
                    X(:,1)=5.0D-1*UNIF
                    ENDIF
                    ENDIF
```

```
  ELSEIF (D.EQ.2) THEN
    NCLUS=N*2.0D-1
  !   GENERATE N-NUM CLUSTER EQUALLY SPACED PTS
    CALL DRNUN(N-NCLUS,UNIF(1:N-NCLUS))
    MUHAT(1:N-NCLUS)=MUL(1)+UNIF(1:N-NCLUS)
    *(MUL(2)-MUL(1))
 !   GENEATE A POINT IN THE RANGE TO CLUSTER PTS ABOUT
    UTEMP=DRNUNF()
    MUTEMP=MUL(1)+UTEMP*(MUL(2)-MUL(1))
    CALL DRNUN(NCLUS,UNIF(1:NCLUS))
!   CLUSTER ABOUT POINT WITH WIDTH=0.2
    MUHAT(N-NCLUS+1:N)=(MUTEMP-1.0D-1)+
    UNIF(1:NCLUS)*
          2.0D-1
      !   FIX IF CLUSTERED POINTS FALL OUTSIDE RANGE OF X
   DO 9 J=1,NCLUS
     IF ((MUHAT(N-NCLUS+J).LT.MUL(1))) THEN
        MUHAT(N-NCLUS+J)=MUL(1)
       ELSEIF (MUHAT(N-NCLUS+J).GT.MUL(2)) THEN
        MUHAT(N-NCLUS+J)=MUL(2)
     ENDIF
    9 CONTINUE
    IF (MODEL.EQ.3) THEN
     X(:,1)=(DLOG(MUHAT/(1.0D0-MUHAT)+Small)-
      BETAP(1))/BETAP(2)
   ELSEIF (MODEL.EQ.0) THEN
     X(:,1)=(DLOG(MUHAT+Small)-BETAP(1))/BETAP(2)
```

```
                      ELSEIF (MODEL.EQ.4) THEN
                       K=1
                       DO WHILE (K.LE.N)
                        X(K,1)=(DNORDF(MUL(K))-BETAP(1))/BETAP(2)
                        K=K+1
                       ENDDO
                       ENDIF
                      ENDIF
                      XMAT(:,1)=1.0D0
                      XMAT(:,2)=X(:,1)
                     LINEARP=MATMUL(XMAT,BETAP)
    !   SIMULATE Y NOW
!   SIMULATE UNIFORM(0,1) RV'S
                        CALL DRNUN(N,UNIF)
!   SIMULATE RESPONSE Y
                    DO 10 I=1,N
                    IF (MODEL.EQ.3) THEN
                        MU(I)=DEXP(LINEARP(I))/
                        (1+DEXP(LINEARP(I)))
                     ELSEIF (MODEL.EQ.0) THEN
                        MU(I)=DEXP(LINEARP(I))
                     ELSEIF (MODEL.EQ.4) THEN
                        MU(K)=DNORDF(LINEARP(I))
                     ENDIF
                     IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
                       IF (UNIF(I).LT.MU(I)) THEN
                         X(I,3)=1.0D0
                       ELSE
                         X(I,3)=0.0D0
                       ENDIF
                     ELSE
                      RANGEX(1)=DMIN1(X(I,1),RANGEX(1))
                      RANGEX(2)=DMAX1(X(I,1),RANGEX(2))
!                     IF (UNIF(I).LT.MU(I)) THEN
                       MEANTEMP=REAL(MU(I))
                       CALL RNPOI(N,MEANTEMP,PTEMP)
                        X(I,3)=DBLE(PTEMP(I))
                      ENDIF
                     10 CONTINUE

!PMLE CALCULATIONS START HERE
                    IF (MODEL.EQ.3) THEN
                        X(:,2)=1.0D0
                    ELSEIF (MODEL.EQ.0) THEN
                        X(:,2)=MU
                    ENDIF
                    WGT=0.0D0
                    LOGLIK1=0.0D0
                    LOGLIK2=0.0D0
                    MUSUM1=0.0D0
                    MUSUM2=0.0D0
                    XSUM=SUM(X(:,3))/N
                    YF=INT(X(:,3))
                    IF (MODEL.EQ.3) THEN
```

```
    BETAF(1)=DLOG((XSUM)/
    ((1.0D0-XSUM+Small))+Small)
ELSEIF (MODEL.EQ.0) THEN
    BETAF(1)=DLOG(XSUM+Small)
ENDIF
BETAF(2)=0.0D0
LINEAR=MATMUL(XMAT,BETAF)
DO 55 I=1,N
  IF (MODEL.EQ.3) THEN
    MUHAT(I)=DEXP(LINEAR(I))/
    (1+DEXP(LINEAR(I)))
    IF (MUHAT(I).LT.1.0D-5) THEN
      MUHAT(I)=1.0D-5
     ENDIF
     WGT(I,I)=DSQRT(MUHAT(I)*
      (1.0D0-MUHAT(I)))
     IF (X(I,3).EQ.1.0D0) THEN
      LOGLIK1=LOGLIK1+(DLOG
      (MUHAT(I)+Small))
      ELSE
       LOGLIK2=LOGLIK2+
      (DLOG(1-MUHAT(I)+Small))
      ENDIF
    ELSEIF (MODEL.EQ.0) THEN
      MUHAT(I)=DEXP(LINEAR(I))
       WGT(I,I)=DSQRT(MUHAT(I))
       IF (YF(I).GT.169) THEN
         YF(I)=169
       ENDIF
       FACT=DFAC(YF(I))
       LOGLIK1=LOGLIK1-MUHAT(I)+X(I,3)*
       DLOG(MUHAT(I)+Small)-DLOG(FACT)
       LOGLIK2=0.0D0
       ELSEIF (MODEL.EQ.4) THEN
       MUHAT(I)=DNORDF(LINEAR(I))
       WGT(I,I)=DSQRT(MUHAT(I)*
       (1.0D0-MUHAT(I)))
       ENDIF
       55 CONTINUE
       LOGLIK=LOGLIK1+LOGLIK2
       IPRINT=0
       TEMP3=0
       ITER=0
       LOGLIKOLD=0.0D0
       ITER=0
       DO WHILE (ITER.LT.25)
         ITER=ITER+1
         H2=MATMUL(TRANSPOSE(XMAT),WGT)
         FISH=MATMUL(H2,TRANSPOSE(H2))
         CALL DLINRG(P,FISH,LDA,FCOV,LDAINV)
         H3=MATMUL(TRANSPOSE(2),FCOV)
         HAT=MATMUL(H3,H2)
         DO 66 O=1,N
         IF (MODEL.EQ.3) THEN
```

```
   XF(0,3)=X(0,3)-MUHAT(0)+HAT(0,0)*(5.0D-1-MUHAT(0))
ELSEIF (MODEL.EQ.0) THEN
   XF(0,3)=X(0,3)-MUHAT(0)+HAT(0,0)/2.0D0
ENDIF
66 CONTINUE
USTAR=MATMUL(TRANSPOSE(XMAT),XF(:,3))
DELTAF=MATMUL(FCOV,USTAR)
MX=DMAX1(DABS(DELTAF(1)),DABS(DELTAF(2)))/10
IF (MX.GT.1.0D0) THEN
   DELTAF=DELTAF/MX
ENDIF
BETAF=BETAF+DELTAF
LINEAR=MATMUL(XMAT,BETAF)
LOGLIKOLD=LOGLIK
! DO HALF-STEPS
  DONE=0
  ITER2=0
  DO WHILE (DONE.EQ.0)
   ITER2=ITER2+1
   IF (MODEL.EQ.3) THEN
    DO 65 I=1,N
     MUHAT(I)=1.0D0/(1.0D0+
     DEXP(-LINEAR(I)))
     IF (X(I,3).EQ.1.0D0) THEN
       LOGLIK1=LOGLIK1+
       (DLOG(MUHAT(I)+Small))
     ELSE
```

```
               LOGLIK2=LOGLIK2+
               (DLOG(1-MUHAT(I)+Small))
             ENDIF
             65 CONTINUE
             ELSEIF (MODEL.EQ.0) THEN
             !FACT=DFAC(Y2(I))
             DO 67 I=1,N
             MUHAT(I)=DEXP(LINEAR(I))
            LOGLIK1=LOGLIK1-MUHAT(I)+
              X(I,3)*DLOG(MUHAT(I)+Small)-
              DLOG(FACT)
               LOGLIK2=0.0D0
             67 CONTINUE
         ENDIF
    LOGLIK=LOGLIK1+LOGLIK2
   IF ((LOGLIK.GT.LOGLIKOLD).OR.
    (ITER2.EQ.5)) THEN
    DONE=1
    !ITER=30
   ELSE
  BETAF=BETAF-DELTAF*2.0D0**(-I)
  ENDIF
  ENDDO
 IF (SUM(DABS(DELTAF)).LT.1.0D-4) THEN
  ITER=250
ENDIF
ENDDO
```

```
                      LINEAR=MATMUL(XMAT,BETAF)
                      CALL DLINRG(P,FCOV,LDA,INFO,LDAINV)
!   COMPUTE SCALED INFO P.433
                      A=INFO/N2
!   COMPUTE A INVERSE
                   CALL DLINRG(P,A,LDA,AINV,LDAINV)
                   IF (IERCD().EQ.0) THEN
                     DONE=1
                    ENDIF
                     ENDIF
                        ENDDO
                        TOL=100*DMACH(4)

                        PIVOT=.FALSE.
!   COMPUTE CHOLESKY DECOMP OF INFORMATION SCALED !   THIS SOLUTION
GIVES B'B=A
                    CALL DCHFAC(P,A,LDA,TOL,IRANK,B,LDR)
                BT=TRANSPOSE(B)
!   COMPUTES BT_INV
                  CALL DLINRG(P,BT,LDA,BTINV,LDAINV)



!   START CALCULATION OF S(X) FOR ALL V=1,N !   NOTE THAT THERE IS A
UNIQUE S(X) FOR EACH UNIQUE X !   CALCULATE S(X) FROM P.435 CALL IT
S !   COMPUTE UI FROM P.437
             IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
               B2=N2*(DEXP(LINEAR))/((1.0D0+
                DEXP(LINEAR))**2.0D0)
              B3=N2*(DEXP(LINEAR)*(1.0D0-DEXP(LINEAR)))/
                  (1.0D0+DEXP(LINEAR))**3.0D0
             B4=N2*((DEXP(LINEAR)-4.0D0*
               DEXP(2.0D0*LINEAR)+DEXP(3.0D0*LINEAR)))/
                  ((1.0D0+DEXP(LINEAR))**4.0D0)
             ELSEIF (MODEL.EQ.0) THEN
               B2=DEXP(LINEAR)
               B3=B2
               B4=B2
             ENDIF
           DO 13 V=1,N
!   MUST DIVIDE UI BY N GIVEN HOW DATA IS ENTERED FOR CTGLM
SUBROUTINE!!!!!!
           TEMP2=MATMUL(BTINV,XMAT(V,:))
           CALL DVCAL (P,1/N2,TEMP2,1,UI(:,V),1)
           TEMP2=MATMUL(AINV,XMAT(V,:))
          TEMP3=DDOT(P,XMAT(V,:), 1, TEMP2, 1)
           TEMP4=1/DSQRT(TEMP3)
!  THIS MULTIPLIES A SCALAR (TEMP4) BY VECTOR (UI) TO GET S(X) FOR I
           CALL DVCAL (P, TEMP4, UI(:,V), 1, S(:,V), 1)
           13 CONTINUE


              DO 30 I=1,N
             DO 31 J=1,N
             U(I,J)=DDOT(P,UI(:,I),1,UI(:,J),1)
!   DOT PRODUCT OF S(X) AND UI
```

```
              SU(I,J)=DDOT(P,S(:,I),1,UI(:,J),1)
             31 CONTINUE
             30 CONTINUE

             DO 14 V=1,N
!   NOTE THAT VAR(X'BLINEAR)=X*VAR(BETA)*X' !   FORMULA FROM AGRESTI
P. 172
             TEMP2=MATMUL(XMAT(V,:),FCOV)
            VAR(V)=DDOT(P,TEMP2,1,XMAT(V,:),1)
!   SET K1 AND K3 TO 0 (SO IT DOESN'T BUILD ON PAST VALUES)
             K1=0.0D0
            K3=0.0D0
!   INITIALIZE C VECTOR
           C=0.0D0
           DO 20 I=1,N
!   C CALCS P.437
           C(2)=C(2)+(B4(I)*(SU(V,I)**2.0D0)
               *U(I,I))
           C(6)=C(6)+(B4(I)*(SU(V,I)**4.0D0))
           C(9)=C(9)+((B2(I)**2.0D0)*SU(V,I)**4.0D0)
!   SEE P 436
          K1=K1+(B3(I)*((SU(V,I)**3.0D0)-
             SU(V,I)*U(I,I)))
         K3=K3+(B3(I)*(SU(V,I)**3.0D0))
            DO 32 J=1,N
         C(1)=C(1)+(B3(I)*B3(J)*SU(V,I)*
           SU(V,J)*U(I,J)**2.0D0)
          C(3)=C(3)+(B3(I)*B3(J)*(SU(V,I)
           **2.0D0)*(SU(V,J)**2.0D0)*U(I,J))
          C(4)=C(4)+(B3(I)*B3(J)*U(I,J)*
           (SU(V,I)**2.0D0)*U(I,J)*U(J,J))
          C(7)=C(7)+(B3(I)*B3(J)*SU(V,I)*
           (SU(V,J)**3.0D0)*U(I,I))
         C(8)=C(8)+(B3(I)*B3(J)*(SU(V,I)
             **3.0D0)*(SU(V,J)**3.0D0))
          32 CONTINUE
           20  CONTINUE
          C(1)=C(1)/(N2**3.0D0)
          C(2)=C(2)/(N2**2.0D0)
          C(3)=C(3)/(N2**3.0D0)
          C(4)=C(4)/(N2**3.0D0)
          C(5)=C(1)
          C(6)=C(6)/(N2**2.0D0)
         C(7)=C(7)/(N2**3.0D0)
          C(8)=C(8)/(N2**3.0D0)
         C(9)=C(9)/(N2**2.0D0)
          K1=K1/(2.0D0*(N2**1.5D0))
          K3=K3/(N2**1.5D0)
!   CONSTANTS USED IN P(X,Z) COMPUTATION P.437
          KAP(1,V)=K1
          TEMP3=C(1)-C(2)+C(4)-C(7)
           KAP(2,V)=1.0D0+5.0D-1*TEMP3-3.0D0*C(3)
              +C(6)+1.75D0*C(8)
           KAP(3,V)=K3
           KAP(4,V)=-9.0D0*C(3)+3.0D0*C(6)+6.0D0*
           C(8)+3.0D0*C(9)
!   CONSTANTS USED IN Q(X,U) COMPUTATIONS P.438
           KAQ(2,V)=C(3)-1.5D0*C(8)-C(5)-C(4)+
                            5.0D-1*C(7)+C(6)-C(2)
           KAQ(3,V)=KAP(3,V)
           KAQ(4,V)=-3.0D0*C(3)-6.0D0*C(4)-6.0D0*
                    C(5)+3.0D0*C(6)+3.0D0*C(7)-3.0D0*C(8)+3.0D0*C(9)
          14 CONTINUE
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !
THIS SECTION NOW STARTS NAIMAN CRITICAL VALUE CALCS !   CREATE
MATRIX T THAT CONTAINS DOMAIN PARTITIONED LIKE ON P.441
            T(1,:)=1.0D0
            DO 50 I=2,P
            K=1
            GETT:DO WHILE ((K.LE.M).AND.(I.GT.1))
            INTER=(RANGEX(2)-RANGEX(1))/(M-1)
            T(I,K)=RANGEX(1)+INTER*(K-1)
            T(I,K+1)=RANGEX(1)+INTER*K
           K=K+1
           ENDDO GETT
            50 CONTINUE
           K0=0.0D0
          DO 40 K=1,(M-1)
!   APPROXIMATE THE MANIFOLD VOLUME (K0) LIKE ON P.441 !   COMPUTE
S(T(K))
           UITEMP=MATMUL(BTINV,T(:,K))
```

```
!       CALL DLINRG(P,A,LDA,AINV,LDAINV)
          TEMP2=MATMUL(AINV,T(:,K))
          TEMP3=DDOT(P,T(:,K), 1, TEMP2, 1)
          TEMP4=1/DSQRT(TEMP3)
         CALL DVCAL (P, TEMP4, UITEMP, 1, SECOND, 1)
!   COMPUTE S(T(K+1))
          UITEMP=MATMUL(BTINV,T(:,K+1))
!       CALL DLINRG(P,A,LDA,AINV,LDAINV)
          TEMP2=MATMUL(AINV,T(:,K+1))
         TEMP3=DDOT(P,T(:,K+1), 1, TEMP2, 1)
          TEMP4=1/DSQRT(TEMP3)
         CALL DVCAL (P, TEMP4, UITEMP, 1, FIRST, 1)
!   COMPUTE DIFFERENCE OF S(T(K))-S(T(K+1))
         DIF=FIRST-SECOND
         INCX=1
!   GET THE NORM OF THE DIFF
        NORMDIF=DNRM2(P,DIF,INCX)
!   ADD TO PREVIOUS NORMS TO APPROXIMATE K0
       K0=K0+NORMDIF
     40  CONTINUE

!   CONTINUE CALC CRIT HERE !   GET S(A) TO SEE IF S(A)=S(B) FOR
INTERVAL [A,B] !   T(:,1) CONTAINS POINT A
          CRITHI=CRITHI0
          CRITLO=1.96D0**2.0D0
          UITEMP=MATMUL(BTINV,T(:,1))
          TEMP2=MATMUL(AINV,T(:,1))
          TEMP3=DDOT(P,T(:,1), 1, TEMP2, 1)

          TEMP4=1/DSQRT(TEMP3)

          CALL DVCAL (P, TEMP4, UITEMP, 1, FIRST, 1)

!   GET S(B) !   T(:,M) CONTAINS POINT B

         UITEMP=MATMUL(BTINV,T(:,M))

         TEMP2=MATMUL(AINV,T(:,M))

         TEMP3=DDOT(P,T(:,M), 1, TEMP2, 1)

         TEMP4=1/DSQRT(TEMP3)

         CALL DVCAL (P, TEMP4, UITEMP, 1, SECOND, 1)

!   COMPUTE DIFFERENCE OF S(A)-S(B) AND THEN TAKE NORM !   IF NORM=0

THEN S(A)=S(B)

        DIF=FIRST-SECOND

        INCX=1

        NORMDIF=DNRM2(P,DIF,INCX)

        IF (NORMDIF.EQ.0) THEN

         E=0.0D0

        ELSE

        E=1.0D0

       ENDIF

!   STARTING VALUES FOR SECANT METHOD !   SEE P.438 FOR FORMULA FOR

FX


      NORMC=1-DNORDF(CRITLO)

      FX=K0/(PI)*DEXP(-(CRITLO**2.0D0)/2.0D0)+

      2.0D0*E*NORMC-ALPHA

      NORMC=1-DNORDF(CRITHI)

      FX1=K0/(PI)*DEXP(-(CRITHI**2.0D0)/2.0D0)+
          2.0D0*E*NORMC-ALPHA
      K=1
      ABSERR=1.0D0
     SECANT: DO WHILE ((K.LT.MAX).AND.
         (ABSERR.GT.EPSILON))
       DF=(FX1-FX)/(CRITHI-CRITLO+Small)
       DP=(FX1)/(DF+Small)
       CRIT=CRITHI-DP
       NORMC=1-DNORDF(CRIT)
       TEMPFX=K0/(PI)*DEXP(-(CRIT**2.0D0)/2.0D0)+
       2.0D0*E*NORMC-ALPHA
      ABSERR=DABS(TEMPFX)
       IF (TEMPFX.LT.0) THEN
        CRITHI=CRIT
       FX1=TEMPFX
        ELSE
        CRITLO=CRIT
        FX=TEMPFX
        ENDIF
       K=K+1
       ENDDO SECANT
      IF (CRIT.GT.CRITHI0) THEN
        CRIT=CRITHI0
      ENDIF
       SUM1=0.0D0
       SUM2=0.0D0
       MEAN1=0.0D0
```

```
       MEAN2=0.0D0
       DO 60 V=1,N
!   CALCULATE CI FOR ETA=X'BETA THEN TRANSFORM TO GET CI ON MEAN
RESP !   CONSTRUCT CI'S HERE (RECALL 4 SCB CI'S)
      IF (VER.EQ.1) THEN
!   BASIC SCR (N IS NUMBER OF POINTS IN X TO MAKE PREDICTIONS
FOR-NEVER MIND FOR NOW!)
     LO(V)=LINEAR(V)-DSQRT(CRIT)*DSQRT(VAR(V))
     HI(V)=LINEAR(V)+DSQRT(CRIT)*DSQRT(VAR(V))
     LEFT=(LINEAR(V)-LINEARP(V))**2.0D0
     RIGHT=(CRIT**2.0D0)*VAR(V)
     !IF ((LINEARP(V).GE.LO(V)).AND.(LINEARP(V).
             LE.HI(V))) THEN
     IF (LEFT.LE.RIGHT) THEN
       CONF=CONF+1
     ENDIF
     !  CENTERED SCR SEE P.637 OF SUN(2001)
     ELSEIF (VER.EQ.2) THEN
     LO(V)=LINEAR(V)-KAP(1,V)*DSQRT(VAR(V))-(
         CRIT)*DSQRT(VAR(V))*DSQRT(KAP(2,V))
     HI(V)=LINEAR(V)-KAP(1,V)*DSQRT(VAR(V))+
        (CRIT)*DSQRT(VAR(V))*DSQRT(KAP(2,V))
     SUM1=SUM1+KAP(1,V)
     SUM2=SUM2+KAP(2,V)
     IF ((LINEARP(V).GE.LO(V)).AND.(LINEARP(V)
             .LE.HI(V))) THEN
     CONF=CONF+1
```

```
ENDIF

ELSEIF (VER.EQ.3) THEN

!   CORRECTED 2 SCR

   !U=SOLVE Q2 FOR U: |u|+q2=c

  XLO=1.0D-2

  XHI=2.0D0*CRIT

  YLO=Q2(XLO)-CRIT

  YHI=Q2(XHI)-CRIT

  X1=XLO

  X2=XHI

  Y1=YLO

  Y2=YHI

 ABSERR=1.0D0

  COUNT=1

  SECANTLOOP: DO WHILE ((ABSERR.GT.EPSILON)

     .AND.(COUNT.LT.25))

  XT = X2 - ((X2-X1)*Y2)/(Y2-Y1+Small)

 YT = Q2(XT) - CRIT

  IF (YT*YLO>0) THEN

  XLO = XT

 YLO = YT

 ELSE

  XHI = XT

 YHI = YT

 ENDIF

X1=XLO

X2=XHI
```

```
     Y1=YLO

     Y2=YHI

     ABSERR=DABS(YT)

     COUNT=COUNT+1

      ENDDO SECANTLOOP

      UCRIT(V)=Q2(XT)

      LO(V)=LINEAR(V)-UCRIT(V)*DSQRT(VAR(V))

      HI(V)=LINEAR(V)+UCRIT(V)*DSQRT(VAR(V))

      SUM1=SUM1+UCRIT(V)

      IF ((LINEARP(V).GE.LO(V)).AND.

     (LINEARP(V).LE.HI(V))) THEN

      CONF=CONF+1

      ENDIF

      ELSEIF (VER.EQ.4) THEN

!   CORRECTED SCR !   GET MAX OF P2 BY SECANT METHOD

     XLO=1.0D-2

     XHI=2.0D0*CRIT

     YLO=P2(XLO)

     YHI=P2(XHI)

     X1=XLO

     X2=XHI

     Y1=YLO

     Y2=YHI

     ABSERR=1.0D0

      K=1

     SECANTLOOP2: DO WHILE ((K.LT.MAX).AND.

      (ABSERR.GT.EPSILON))
     XT = X2 - ((X2-X1)*Y2)/(Y2-Y1+Small)

      YT = P2(XT)

     IF (YT*YLO>0) THEN

       XLO = XT

     XLO = YT

       ELSE

     XHI = XT

     YHI = YT

     ENDIF

     X1=XLO

     X2=XHI

     Y1=YLO

     Y2=YHI

     ABSERR=DABS(YT)

     K=K+1

     ENDDO SECANTLOOP2

     MAX_P2(V) = DABS(P2(XT))

     UCRIT(V)=CRIT-MAX_P2(V)

!   NOW USE SECANT METHOD AGAIN TO FIND CRIT FOR EQ.41 P. 440

     CRITLO=1.96D0

!   THIS IS SCHEFFE SOLUTION

     CRITHI=CRITHI0

     NORMC=1-DNORDF(CRITLO-MAX_P2(V))

     FX=K0/(2.0D0*PI)*DEXP(-((CRITLO-MAX_P2(V))

         **2.0D0)/2.0D0)+E*NORMC-ALPHA

     NORMC=1-DNORDF(CRITHI-MAX_P2(V))

     FX1=K0/(2.0D0*PI)*DEXP(-((CRITHI-MAX_P2(V))
          **2.0D0)/2.0D0)+E*NORMC-ALPHA
    K=1
      ABSERR=1.0D0
      SECANT2: DO WHILE ((K.LT.MAX).AND.
             (ABSERR.GT.EPSILON))
      DF=(FX1-FX)/(CRITHI-CRITLO+Small)
      DP=(FX1)/(DF+Small)
    !NOTE: THIS IS AUCRIT REALLY
       CRIT=CRITHI-DP
       NORMC=1-DNORDF(CRIT-MAX_P2(V))
       TEMPFX=K0/(2.0D0*PI)*DEXP(-((CRIT-
       MAX_P2(V))**2.0D0)/2.0D0)+E*NORMC-ALPHA
             ABSERR=DABS(TEMPFX)
      IF (TEMPFX.LT.0) THEN
       CRITHI=CRIT
        FX1=TEMPFX
       ELSE
        CRITLO=CRIT
      FX=TEMPFX
       ENDIF
       K=K+1
       ENDDO SECANT2
    LO(V)=LINEAR(V)-UCRIT(V)*DSQRT(VAR(V))
     HI(V)=LINEAR(V)+UCRIT(V)*DSQRT(VAR(V))
       SUM2=SUM2+MAX_P2(V)
       IF ((LINEARP(V).GE.LO(V)).AND.
       (LINEARP(V).LE.HI(V))) THEN
```

```
     CONF=CONF+1
     ENDIF
    ENDIF
   60 CONTINUE
    ECL(L)=CONF/N2
   8 CONTINUE
  MEANECL=SUM(ECL)/DBLE(REPS)
    RESULTS(1)=MODEL
    RESULTS(2)=H
    RESULTS(3)=G
    RESULTS(4)=N
    RESULTS(5)=MEANECL
    RESULTS(6)=VER
   WRITE(9,*) RESULTS
    WRITE(10,*) MEANECL
    7 CONTINUE
   6 CONTINUE
 5 CONTINUE
  3 CONTINUE
2 CONTINUE
 END
```

```
DOUBLE PRECISION FUNCTION P2(U)

DOUBLE PRECISION U,KAP(5),KAQ(5),C(9),C2,N2,P3,

PART1,PART2,PART3

COMMON KAP,KAQ,P3,N2,C,C2

PART1=KAP(2)-1.0D0+KAP(1)**2.0D0

PART2=(KAP(4)+4.0D0*KAP(1)*KAP(3))*(U**2.0D0-3.0D0)

PART3=KAP(3)*(U**4.0D0-1.0D1*U**2.0D0+1.5D1)

P2=U*(3.6D1*PART1+3.0D0*PART2+PART3)/7.2D1

RETURN

END


DOUBLE PRECISION FUNCTION Q2(U)

DOUBLE PRECISION U,KAQ(5),KAP(5),P3,N2,C2,C(9),

PART1,PART2

COMMON KAP,KAQ,P3,N2,C,C2

PART1=U*U-3.0D0

PART2=(U*U-1.0D1)*U*U+1.5D1

Q2=-U*(3.6D1*KAQ(2) + 3.0D0*KAQ(4)*PART1 + KAP(3)

**2.0D0*PART2)/7.2D1

RETURN

END
```

APPENDIX G

Fortran Code: Piegorsch-Casella PMLE for the Parameters

```
USE MSIMSL

INTEGER PINT,REPS,KINT,LDA,LDEVEC,LDAINV

DOUBLE PRECISION P,R

PARAMETER(PINT=5,NINT=300,REPS=5000, KINT=PINT-1,

LDA=PINT, LDEVEC=PINT,LDAINV=PINT)

DOUBLE PRECISION RANGEX(2),RX(2),VER(2**(PINT-1),PINT),

DVER(PINT,2**(PINT-1)),ZMAX(PINT),ZMIN(PINT)

DOUBLE PRECISION S,A2, B2 ,Q2,C, Dp,Z(2),ECL(REPS),

ECL2(REPS),N,MEANECL,MEANECL2,LEFT,RIGHT

DOUBLE PRECISION BETAP(PINT),XMAT(NINT,PINT),CMAT(KINT,PINT),

LINEAR(NINT),LINEARP(NINT),MU(NINT)

DOUBLE PRECISION VTEMP(PINT,PINT),LINEARCP(KINT),

LINEARC(KINT),UNIF(NINT)

DOUBLE PRECISION RESULTS(6),D(PINT),D2(PINT,PINT),D3(PINT,PINT),

F(PINT,PINT),UMAT(PINT,PINT)

REAL MEANTEMP,PS

INTEGER PTEMP(KINT),DONE,TEMPX(NINT)

INTEGER L,ISEED,NINT,CONF,CONF2

DOUBLE PRECISION ERS

DOUBLE PRECISION CHIP

COMMON S, R, P, A2, B2, C
```

```
INTEGER IRULE,K,I,J,RINT

DOUBLE PRECISION LOW(2),UP(2),ERRABS,ERREL,TEMP,COVLIN,TEMP2(PINT)

DOUBLE PRECISION ERSRESULT1,ERSRESULT2,ERSTEMP,CHIPTEMP,FX,FX1,C0,C1,C2

DOUBLE PRECISION ERREST,CHIPRESULT1,ZMAXTEMP,CRIT

DOUBLE PRECISION CHIPRESULT2

EXTERNAL ERS, CHIP

PARAMETER(Delta=1.0D-3,Epsilon=1.0D-6,Max=1000,Small=1.0D-10)

INTEGER IPVT(PINT),COUNT2

DOUBLE PRECISION H2(PINT,NINT),H3(NINT,PINT),HAT(NINT,NINT)

INTEGER COUNT,ITER,ITER2,YF(NINT),ICODE

DOUBLE PRECISION X(NINT,PINT+1),X2(NINT,PINT+1),WGT(NINT,NINT)

DOUBLE PRECISION MUHAT(NINT),FACT

DOUBLE PRECISION BETAF(PINT),LOGLIK1,LOGLIK2,MUSUM1,MUSUM2,XSUM

DOUBLE PRECISION FISH(PINT,PINT),FACU(PINT,PINT),DET1,DET2,

USTAR(PINT),DELTAF(PINT)

DOUBLE PRECISION MX,LOGLIK,LOGLIKOLD,C10





OPEN (UNIT=8, FILE='C:/Results/CMAT RESULTS/DferrorPMLE300.txt')

OPEN (UNIT=9, FILE='C:/Results/CMAT RESULTS/RESULTS_PCPMLE300.txt')

OPEN (UNIT=10, FILE='C:/Results/CMAT RESULTS/RELERR.txt')

OPEN (UNIT=11, FILE='C:/Results/CMAT RESULTS/ECL_PCPMLE300.txt')

OPEN (UNIT=12, FILE='C:/Results/CMAT RESULTS/DATA.txt')
```

```
!   DEFINE R, S, AND P AND INTEGER VERSIONS FOR LATER

    P=DBLE(PINT)

    R=DBLE(KINT)

    N=DBLE(NINT)

    S=P-R

    RINT=INT(R)

    ISEED=3427

    CALL RNSET(ISEED)

    C10=P*DFIN(9.5D-1,P,1.0D6)



!   LOOP FOR MODEL TYPE 1=LOGIT, 2=POISSON, 3=PROBIT

        DO 2 B=1,2

            IF (B.EQ.1) THEN

                MODEL=3

            ELSEIF (B.EQ.2) THEN

                MODEL=0

            ELSEIF (B.EQ.3) THEN

                MODEL=4

            ENDIF

!   LOOP FOR BETA PARAMETERS 1=(-1,.5), 2=(0,1), 3=(2,4),

4=(-.25,-.5)

            DO 3 H=1,2

                IF (H.EQ.1) THEN

                    BETAP(1)=-1.0D0

                    BETAP(2)=5.0D-1

                    BETAP(3)=-2.5D-1
                    BETAP(4)=-5.0D-1
                    BETAP(5)=2.5D-1
                ELSEIF (H.EQ.2) THEN
                    BETAP(1)=1.0D0
                    BETAP(2)=2.0D0
                    BETAP(3)=4.0D0
                    BETAP(4)=2.0D0
                    BETAP(5)=4.0D0
                ELSEIF (H.EQ.3) THEN
                    BETAP(1)=-1.0D0
                    BETAP(2)=2.0D0
                    BETAP(3)=4.0D0
                    BETAP(4)=2.0D0
                    BETAP(5)=4.0D0
                ENDIF
!   LOOP FOR DOMAIN TYPES 1=EQUALLY SPACED, 2=ONE CLUSTER
DO 6 E=1,1
DO 7 A=1,REPS
   DONE=0
   CONF=0
   CONF2=0
   88 CONTINUE
!   GENERATE X'S - THESE ARE BINOMIAL RV'S
   DONE=1
   RANGEX(1)=DRNUNF()
   RANGEX(2)=DRNUNF()
   X=0.0D0
   XMAT=0.0D0

   PS=5.0D-1

   CALL RNBIN (NINT, KINT, PS, TEMPX)

   DO 33 Q=1,NINT

      IF (TEMPX(Q).EQ.0) THEN

          X(Q,1)=1.0D0

          XMAT(Q,2)=1.0D0

     ELSEIF (TEMPX(Q).EQ.1) THEN

          X(Q,2)=1.0D0

          XMAT(Q,3)=1.0D0

     ELSEIF (TEMPX(Q).EQ.2) THEN

          X(Q,3)=1.0D0

          XMAT(Q,4)=1.0D0

     ELSEIF (TEMPX(Q).EQ.3) THEN

          X(Q,4)=1.0D0

          XMAT(Q,5)=1.0D0

     ENDIF

   33 CONTINUE

   DO 34 Q=2,PINT

     IF (SUM(XMAT(:,Q)).EQ.0) THEN

         GOTO 88

     ENDIF

   34 CONTINUE


     XMAT(:,1)=1.0D0

! CRATE C MATRIX

    DO 11 K=1,PINT-1
```

```
       DO 12 W=1,KINT
         IF (W.EQ.K) THEN
               CMAT(W,K+1)=1.0D0
         ELSE
               CMAT(W,K+1)=0.0D0
         ENDIF
       12 CONTINUE
    11 CONTINUE
    CMAT(:,1)=0.0D0
    LINEARP=MATMUL(XMAT,BETAP)
    LINEARCP=MATMUL(CMAT,BETAP)
    !   SIMULATE     Y NOW
!   SIMULATE UNIFORM    (0,1) RV'S
CALL DRNUN(NINT,UNIF)
!   SIMULATE RESPONSE Y
DO 10 I=1,NINT
IF (MODEL.EQ.3) THEN
MU(I)=DEXP(LINEARP(I))/
(1+DEXP(LINEARP(I)))
ELSEIF (MODEL.EQ.0) THEN
MU(I)=DEXP(LINEARP(I))
ELSEIF (MODEL.EQ.4) THEN
MU(I)=DNORDF(LINEARP(I))
ENDIF
IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
IF (UNIF(I).LT.MU(I)) THEN
X(I,PINT+1)=1.0D0
```

```
ELSE
 X(I,PINT+1)=0.0D0
 ENDIF
 ELSE
 RANGEX(1)=DMIN1(X(I,1),RANGEX(1))
  RANGEX(2)=DMAX1(X(I,1),RANGEX(2))
  !IF (UNIF(I).LT.MU(I)) THEN
  MEANTEMP=REAL(MU(I))
  CALL RNPOI(NINT,MEANTEMP,PTEMP)
  X(I,PINT+1)=DBLE(PTEMP(I))

  ENDIF
  10 CONTINUE

!PMLE CALCULATIONS START HERE
IF (MODEL.EQ.3) THEN
X(:,PINT)=1.0D0
ELSEIF (MODEL.EQ.0) THEN
X(:,PINT)=MU
 ENDIF
 WGT=0.0D0
 LOGLIK1=0.0D0
 LOGLIK2=0.0D0
 MUSUM1=0.0D0
 MUSUM2=0.0D0
 XSUM=SUM(X(:,PINT+1))/N
 YF=INT(X(:,PINT+1))
```

```
BETAF=0.0D0
IF (MODEL.EQ.3) THEN
BETAF(1)=DLOG((XSUM)/((1.0D0-XSUM+Small))+Small)
ELSEIF (MODEL.EQ.0) THEN
BETAF(1)=DLOG(XSUM+Small)
ENDIF
BETAF(2)=0.0D0
LINEAR=MATMUL(XMAT,BETAF)
DO 55 I=1,NINT
IF (MODEL.EQ.3) THEN
MUHAT(I)=DEXP(LINEAR(I))/(1+DEXP(LINEAR(I)))
IF (MUHAT(I).LT.1.0D-4) THEN
MUHAT(I)=1.0D-4
ENDIF
WGT(I,I)=DSQRT(MUHAT(I)*(1.0D0-MUHAT(I)))
IF (X(I,PINT+1).EQ.1.0D0) THEN
LOGLIK1=LOGLIK1+(DLOG(MUHAT(I)+Small))
ELSE
LOGLIK2=LOGLIK2+(DLOG(1-MUHAT(I)+Small))
ENDIF
 ELSEIF (MODEL.EQ.0) THEN
 MUHAT(I)=DEXP(LINEAR(I))
 WGT(I,I)=DSQRT(MUHAT(I))
 IF (YF(I).GT.169) THEN
 YF(I)=169
 ENDIF
 FACT=DFAC(YF(I))
```

```
  LOGLIK1=LOGLIK1-MUHAT(I)+X(I,PINT+1)*
  DLOG(MUHAT(I)+Small)-DLOG(FACT)
  LOGLIK2=0.0D0
  ELSEIF (MODEL.EQ.4) THEN
  MUHAT(I)=DNORDF(LINEAR(I))
  WGT(I,I)=DSQRT(MUHAT(I)*(1.0D0-MUHAT(I)))
  ENDIF
  55 CONTINUE

  LOGLIK=LOGLIK1+LOGLIK2
  TEMP3=0
  H2=MATMUL(TRANSPOSE(XMAT),WGT)
  FISH=MATMUL(H2,TRANSPOSE(H2))
  CALL ERSET(0,0,0)
  CALL DLFTSF (PINT,FISH , LDA, FACU, LDA, IPVT)
!                                 Compute the determinant

CALL DLFDSF (PINT, FACU, LDA, IPVT, DET1, DET2)

LOGLIK=LOGLIK+5.0D-1*(DET1*1.0D1**DET2)
ITER=0
LOGLIKOLD=0.0D0
ITER=0
DO WHILE (ITER.LT.25)
ITER=ITER+1
H2=MATMUL(TRANSPOSE(XMAT),WGT)
FISH=MATMUL(H2,TRANSPOSE(H2))
```

```
CALL ERSET(0,0,0)

CALL DLINRG(PINT,FISH,LDA,F,LDAINV)

ICODE=IERCD()

IF ((ICODE.EQ.1).OR.(ICODE.EQ.2)) THEN

COUNT2=COUNT2+1

WRITE(8,*) ICODE,L,H,R,COUNT2

GOTO 88

ENDIF


H3=MATMUL(TRANSPOSE(H2),F)

HAT=MATMUL(H3,H2)

LINEAR=MATMUL(XMAT,BETAF)

DO 66 O=1,NINT

IF (MODEL.EQ.3) THEN

MUHAT(O)=1.0D0/(1.0D0+DEXP(-LINEAR(O)))

X2(O,PINT+1)=X(O,PINT+1)-MUHAT(O)+HAT(O,O)*(5.0D-1-MUHAT(O))

ELSEIF (MODEL.EQ.0) THEN

MUHAT(O)=DEXP(LINEAR(O))

X2(O,PINT+1)=X(O,PINT+1)-MUHAT(O)+HAT(O,O)/2.0D0

ENDIF

66 CONTINUE

USTAR=MATMUL(TRANSPOSE(XMAT),X2(:,PINT+1))

DELTAF=MATMUL(F,USTAR)

MX=DMAX1(DABS(DELTAF(1)),DABS(DELTAF(2)))/10

IF (MX.GT.1.0D0) THEN

DELTAF=DELTAF/MX

ENDIF
```

```
BETAF=BETAF+DELTAF
LOGLIKOLD=LOGLIK
!  DO HALF-STEPS
DONE=0
ITER2=0
LINEAR=MATMUL(XMAT,BETAF)
DO WHILE (DONE.EQ.0)
ITER2=ITER2+1
IF (MODEL.EQ.3) THEN
DO 65 I=1,NINT
MUHAT(I)=1.0D0/(1.0D0+DEXP(-LINEAR(I)))
IF (X(I,3).EQ.1.0D0) THEN
 LOGLIK1=LOGLIK1+(DLOG(MUHAT(I)+Small))
 ELSE
  LOGLIK2=LOGLIK2+(DLOG(1-MUHAT(I)+Small))
  ENDIF
  65 CONTINUE
  ELSEIF (MODEL.EQ.0) THEN

  DO 67 I=1,NINT
     MUHAT(I)=DEXP(LINEAR(I))
        LOGLIK1=LOGLIK1-MUHAT(I)+
          X(I,PINT+1)*DLOG(MUHAT(I)+Small)-
            DLOG(FACT)
              LOGLIK2=0.0D0
              67 CONTINUE
 ENDIF
 LOGLIK=LOGLIK1+LOGLIK2
 H2=MATMUL(TRANSPOSE(XMAT),WGT)
 FISH=MATMUL(H2,XMAT)
 CALL DLFTSF (PINT,FISH , LDA, FACU,LDA, IPVT)
!                                 Compute the determinant

CALL DLFDSF (PINT, FACU, LDA, IPVT,DET1, DET2)
LOGLIK=LOGLIK+5.0D-1*(DET1*1.0D1**DET2)
IF ((LOGLIK.GT.LOGLIKOLD).OR.
(ITER2.EQ.5)) THEN
DONE=1
 ELSE
 BETAF=BETAF-DELTAF*2.0D0**(-I)
 ENDIF
 ENDDO
 IF (SUM(DABS(DELTAF)).LT.1.0D-2) THEN
 ITER=2.5D5
 ENDIF
 ENDDO
 LINEAR=MATMUL(XMAT,BETAF)
 LINEARC=MATMUL(CMAT,BETAF)



 iF (DONE.EQ.1) THEN
 CALL DLINRG(PINT,FISH,LDA,F,LDAINV)
 ENDIF
 IF (IERCD().NE.0) THEN
 DONE=1
 ENDIF
!   CALCULATE Q2 FROM P.867-868 OF CS !   FIRST CALCULATE
DIAGONALIZED VERTICES
CALL DEVCSF (PINT, F, LDA, D, UMAT, LDEVEC)
DO 16 K=1,PINT
DO 17 L=1,PINT
IF (K.EQ.L) THEN
D2(K,L)=D(K)
ELSE
D2(K,L)=0.0D0
ENDIF
17 CONTINUE
16 CONTINUE


RX(1)=1.0D0
RX(2)=2.0D0
COUNT=1
DO 98 I=1,2
DO 97 J=1,2
DO 96 K=1,2
DO 95 L=1,2
VER(COUNT,2)=RX(I)
VER(COUNT,3)=RX(J)
VER(COUNT,4)=RX(K)
VER(COUNT,5)=RX(L)

COUNT=COUNT+1

95 CONTINUE

96 CONTINUE

97 CONTINUE

98 CONTINUE


CALL DLINRG(PINT,D2**5.0-1,LDA,D3,LDAINV)

VTEMP=MATMUL(D3,TRANSPOSE(UMAT))

DVER=MATMUL(VTEMP,TRANSPOSE(VER))


DO 25 K=1,PINT

ZMAXTEMP=-1.0D10

DO 13 I=1,2**KINT-1

Z(1)=DVER(K,I)

Z(2)=DVER(K,I+1)

ZMAXTEMP=(DMAX1(Z(1),Z(2),ZMAXTEMP))

TEMP=(DMIN1(Z(1),Z(2),ZMAXTEMP))

IF ((TEMP.LT.0.0D0).AND.(ZMAXTEMP.GT.0.0D0)) THEN

ZMIN(K)=0.0D0

ELSE

ZMIN(K)=DMIN1(DABS(Z(1)),DABS(Z(2)),DABS(ZMAXTEMP))

ENDIF

ZMAX(K)=DMAX1(DABS(Z(1)),DABS(Z(2)),DABS(ZMAXTEMP))

13 CONTINUE

25 CONTINUE

CRIT=1.0D10
```

```
DO 26 K=1,KINT


Q2=SUM((ZMIN(1:K)+1.0D0)**2.0D0)/SUM((ZMAX(K+1:PINT)-5.0D-1)**2.0D0)


B2=(1.0D0+Q2)**(-1.0D0)

A2=1-B2

!   STARTING VALUES FOR THE SECANT METHOD

C0=1.96D0**2.0D0

C1=C10

!   DEFINE UPPER AND LOWER LIMITS FOR ERS INTEGRAL

LOW(1)=C0

LOW(2)=C1

!   BE CAREFUL WHEN B2=0

IF (B2.GT.1.0D-6) THEN

UP(1)=(C0)/(B2)

UP(2)=(C1)/(B2)

ELSE

UP(1)=9.99D9

UP(2)=9.99D10

END IF

C=DSQRT(C0)

ERRABS=1.0d-6

ERREL=1.0d-6

IRULE=2

!   Call 1st ERS integral

IF (B2.EQ.0.0D0) THEN

CALL DQDAGI(ERS,LOW(1),1,ERRABS,ERREL,ERSRESULT1,ERREST)
```

```
ELSE IF (B2.EQ.1.0D0) THEN

ERSRESULT1=0.0D0

ELSE

CALL DQDAG(ERS,LOW(1),UP(1)+Small,ERRABS,ERREL,IRULE,ERSRESULT1,ERREST)

END IF

!   Call 2nd ERS integral

C=DSQRT(C1)

IF (B2.EQ.0.0D0) THEN

CALL DQDAGI(ERS,LOW(2),1,

ERRABS,ERREL,ERSRESULT2,ERREST)

ELSE IF (B2.EQ.1.0D0) THEN

ERSRESULT2=0.0D0

ELSE

CALL DQDAG(ERS,LOW(2),UP(2)+Small,ERRABS,ERREL,IRULE,ERSRESULT2,ERREST)

END IF

!   SET LIMITS FOR CHIP INTEGRAL

LOW(1)=0.0D0

LOW(2)=0.0D0

UP(1)=C0

UP(2)=C1

!   Call the 1st chi square (p) integral

CALL DQDAG(CHIP,LOW(1),UP(1),ERRABS,

ERREL,IRULE,CHIPRESULT1,ERREST)

!   Call the 2nd chi square (p) integral

CALL DQDAG(CHIP,LOW(2),UP(2),ERRABS,

ERREL,IRULE,CHIPRESULT2,ERREST)

!   COMPUTE FIRST TWO VALUES OF THE FUNCTION F - SHOULD HAVE 0.95
```

```
BETWEEN THESE
FX=ERSRESULT1+CHIPRESULT1
FX1=ERSRESULT2+CHIPRESULT2
!   K COUNTS HOW MANY ITERATIONS
 L=0
 AbsErr=1.0d0
!   BEGIN LOOP TO OPTIMIZE F FOR C2 UNTIL ABSERR < EPS
SECANTLOOP: DO WHILE ((L.LT.Max).AND.(AbsErr.GT.Epsilon))
!   CALCULATES NEW ITERATION OF C
Df=(FX1-FX)/(C1-C0)
IF (Df.EQ.0) THEN
WRITE (8,*) Df
ELSE
!   CALCULATES NEW ITERATION OF C
Dp=(FX1-9.5D-1)/Df
C2=(C1-Dp)**1.0D0
ENDIF

!   CALCULATE NEW FX FOR C2
LOW(1)=C2
IF (B2.NE.0.0D0) THEN
UP(1)=(C2)/(B2)
 ENDIF
 ERRABS=1.0d-6
 ERREL=1.0d-6
 IRULE=2
!   Call ERS integral
```

```
C=DSQRT(C2)
IF (B2.EQ.0.0D0) THEN
CALL DQDAGI(ERS,LOW(1),1,ERRABS,
ERREL,ERSTEMP,ERREST)
ELSEIF (B2.EQ.1.0D0) THEN
ERSTEMP=0.0D0
ELSE
CALL DQDAG(ERS,LOW(1),UP(1),ERRABS,ERREL,IRULE,ERSTEMP,ERREST)
ENDIF
LOW(1)=0.0D0
UP(1)=C2
!   Call the chi square (p) integral
CALL DQDAG(CHIP,LOW(1),UP(1),ERRABS,ERREL,IRULE,CHIPTEMP,ERREST)
!   CALCULATES THE NEW F VALUE (SOMEWHERE BETWEEN FX AND FX1)
TEMP=ERSTEMP+CHIPTEMP
!   CALCULATE THE ERRORS
AbsErr=DABS(TEMP-9.5D-1)
RelErr=DABS(Dp)/(DABS(C2)+Small)
!   RECORDS SMALL RELERR
IF (RelErr.GT.Delta) THEN
WRITE(10,*) L,RelErr
ENDIF
!   IF TEMP < 0.95 THEN OVERESTIMATED C2
IF (TEMP.LT.9.5D-1) THEN
C0=C2
FX=TEMP
ELSE
```

```
!   IF TEMP >= 0.95 THEN C2 UNDERESTIMATED

C1=C2

FX1=TEMP

ENDIF

L=L+1


ENDDO SECANTLOOP

26 CONTINUE

CONF=0

CONF2=0

DO 60 I=1,KINT

TEMP2=MATMUL(CMAT(I,:),F)

COVLIN=DDOT(PINT,TEMP2,1,CMAT(I,:),1)

!   RECORD WHEN BOUNDS ARE ESTIMATED CORRECTLY

LEFT=(LINEARC(I)-LINEARCP(I))**2.0D0

RIGHT=CRIT*COVLIN

IF (LEFT.LE.RIGHT) THEN

CONF=CONF+1

ENDIF

LEFT=(LINEARC(I)-LINEARCP(I))**2.0D0

RIGHT=C10*COVLIN

IF (LEFT.LE.RIGHT) THEN

CONF2=CONF2+1

ENDIF

60 CONTINUE

K=DBLE(KINT)

ECL(A)=DBLE(CONF)/K
```

```
ECL2(A)=DBLE(CONF2)/K

MEANECL=SUM(ECL)/REPS

MEANECL2=SUM(ECL2)/REPS

RESULTS(1)=MODEL

RESULTS(2)=H

RESULTS(3)=NINT

RESULTS(4)=E

RESULTS(5)=MEANECL

RESULTS(6)=MEANECL2

7 CONTINUE

WRITE(11,*) MEANECL,MEANECL2

WRITE(9,*) C2,RESULTS

6 CONTINUE

5 CONTINUE

4 CONTINUE

3 CONTINUE

2 CONTINUE


    STOP

    END


!   Now start defining the functions !   used in the above main

program !   Define the ERS function

    DOUBLE PRECISION FUNCTION ERS(T)

    DOUBLE PRECISION PART1, PART2, PART3, DFD, DFN

    COMMON S,R,P,A2,B2,C

    DOUBLE PRECISION T,S,R,P,A2,B2,C, TEMP, TEMP2, TEMP3, TEMP4,TEMP5,TEMP6
    DOUBLE PRECISION DGAMMA, DFDF
    !
    DFD=S
    DFN=R
    TEMP4=C*(T-C**2.0D0)**5.0D-1
    TEMP5=DSQRT(A2)*DSQRT(B2)*T
    TEMP6=A2*T-C**2.0D0
    PART1=DFDF((S/R)*((TEMP4-TEMP5)/TEMP6)**2.d0,DFN,DFD)
    PART2=T**((P/2.d0)-1.d0)
    TEMP=P/2.0D0
    TEMP2=2.0D0**TEMP
    TEMP3=DGAMMA(TEMP)*TEMP2
    PART3=DEXP(-T/2.d0)/TEMP3
    ERS=PART1*(PART2*PART3)

    RETURN
    END



!   Define the chi-square function
    DOUBLE PRECISION FUNCTION CHIP(T)
    DOUBLE PRECISION PART1, PART2, PART3
    COMMON S,R,P,A2,B2,C
    DOUBLE PRECISION T,S,R,P
    DOUBLE PRECISION DGAMMA
```

```
PART1=T**((P/2.d0)-1.d0)

PART2=DEXP(-T/2.d0)

PART3=DGAMMA(P/2.d0)*2.d0**(P/2.d0)

CHIP=(PART1*PART2)/(PART3)

RETURN

END
```

APPENDIX H

Fortran Code: SCR PMLE for the Parameters

```
USE MSIMSL
INTEGER N,P,M,INCX,LDA,LDR,LDAINV,VER,REPS,NCOMP,MAXK
DOUBLE PRECISION Small
PARAMETER(N=200,P=5,REPS=5000,NCOMP=P-1,M=NCOMP*(NCOMP+1)-1,
Small=1.0D-10,MAXK=100,LDA=P,LDR=P,LDFAC=P,LDAINV=P)
REAL MEANTEMP,PS
DOUBLE PRECISION B2(NCOMP),B3(NCOMP),B4(NCOMP),B(NCOMP,NCOMP),
A(NCOMP,NCOMP)
DOUBLE PRECISION AINV(NCOMP,NCOMP),LO(NCOMP),HI(NCOMP),
CMAT(NCOMP,P),XMAT(N,P)
DOUBLE PRECISION SU(NCOMP,NCOMP),S(P,NCOMP),UI(P,NCOMP),
U(NCOMP,NCOMP),K1,UITEMP(NCOMP)
DOUBLE PRECISION CRIT,C(9),KAP(5,NCOMP),KAQ(5,NCOMP),MAX_P2(NCOMP)
DOUBLE PRECISION BT(NCOMP,NCOMP),TEMP2(NCOMP),TEMP3,TEMP4,
TEMP5(NCOMP)
DOUBLE PRECISION TEMP6(NCOMP),K2VEC(NCOMP),BTINV(NCOMP,NCOMP)
DOUBLE PRECISION P2,Q2,F,P3,N2
DOUBLE PRECISION UCRIT(NCOMP),K0,CRITLO,CRITHI,FX,FX1,ALPHA,
MAX,ABSERR,KSI2VEC(NCOMP)
DOUBLE PRECISION KSI,UIT(NCOMP,M**(NCOMP-2)),ST(NCOMP,M**(NCOMP-2)),
K2(10),K2MEAN
```

```
DOUBLE PRECISION STEMP1(NCOMP),STEMP2(NCOMP),STEMP3(NCOMP),KSI2,
KSI2MEAN
DOUBLE PRECISION DF,DP,TEMPFX,LINEAR(N),LINEARP(N),VAR(NCOMP),K3,TOL,C10
DOUBLE PRECISION XLO,XHI,YLO,YHI,PI,X1,X2,Y1,Y2,XT,YT,LINEARC(NCOMP),
TEMPT(NCOMP,M**NCOMP)
DOUBLE PRECISION FIRST(NCOMP),SECOND(NCOMP),DIF(NCOMP),NORMDIF,T(NCOMP,
M**NCOMP),INTER,RESULTS(6)
DOUBLE PRECISION SUM1,SUM2,MEAN1,MEAN2,UNIF(N),RANGEX(2),ECL(REPS),MEANECL
DOUBLE PRECISION PART1,PART2,PART3,PART4,PART5,PART6,GRAD(NCOMP,M**NCOMP),M0
DOUBLE PRECISION CROSS(2,3),CROSSDOT,NORM1,NORM2,COSTH,THETA
DOUBLE PRECISION GRADIENT1(NCOMP),GRADIENT2(NCOMP),LINEARCP(NCOMP)
EXTERNAL P2,Q2,F
INTEGER I,J,K,IRANK,CONF,PTEMP(NCOMP),IPVT(P),COUNT2,LOOP
DOUBLE PRECISION H2(P,N),H3(N,P),HAT(N,N)
DOUBLE PRECISION MU(N),V
INTEGER COUNT,TEMPX(N),ITER,ITER2,YF(N),ICODE
DOUBLE PRECISION COV2(NCOMP,NCOMP),X(N,P+1),XF(N,P+1),WGT(N,N)
DOUBLE PRECISION BETAP(P),MUHAT(N),FACT
DOUBLE PRECISION BETAF(P),LOGLIK1,LOGLIK2,MUSUM1,MUSUM2,XSUM
DOUBLE PRECISION FISH(P,P),FACU(P,P),DET1,DET2,USTAR(P),DELTAF(P),FCOV(P,P)
DOUBLE PRECISION MX,LOGLIK,LOGLIKOLD,GG,HH

COMMON KAP,KAQ,P3,N2,C,C2,LOOP


OPEN (UNIT=8, FILE='C:/Results/ERRORS.txt')
OPEN (UNIT=9, FILE='C:/Results/RESULTS.txt')
```

```
    OPEN (UNIT=10, FILE='C:/Results/ECL.txt')


    MAX=500

    EPSILON=1.0D-6

    ALPHA=5.0D-2

    PI=3.1415926535897932D0

    P3=DBLE(P)

    N2=DBLE(N)

    K0=0.0D0


    RINT=INT(R)

    CALL RNSET(34271)

    COUNT=0

!   THIS IS SCHEFFE SOLUTION


    C10=(P3*DFIN(9.5D-1,P3,1.0D6))

!   LOOP FOR MODEL TYPE 1=LOGIT, 2=POISSON, 3=PROBIT

    DO 2 Z=1,2

        IF (Z.EQ.1) THEN

            MODEL=3

        ELSEIF (Z.EQ.2) THEN

            MODEL=0

        ELSEIF (Z.EQ.3) THEN

            MODEL=4

        ENDIF

!   LOOP FOR BETA PARAMETERS

        DO 3 H=1,2
```

```
            IF (H.EQ.1) THEN
                BETAP(1)=-1.0D0
                BETAP(2)=5.0D-1
                BETAP(3)=-2.5D-1
                BETAP(4)=-5.0D-1
                BETAP(5)=2.5D-1
            ELSEIF (H.EQ.2) THEN
                BETAP(1)=1.0D0
                BETAP(2)=2.0D0
                BETAP(3)=4.0D0
                BETAP(4)=2.0D0
                BETAP(5)=4.0D0
            ELSEIF (H.EQ.3) THEN
                BETAP(1)=-1.0D0
                BETAP(2)=2.0D0
                BETAP(3)=4.0D0
                BETAP(4)=2.0D0
                BETAP(5)=4.0D0
            ENDIF
!   LOOP FOR RESTRICTED DOMAIN TYPES 1=UNRESTRICTED, 2=WIDE,
3=NARROW

!   LOOP FOR DOMAIN TYPES 1=EQUALLY SPACED, 2=ONE CLUSTER

                DO 7 R=1,4
                    VER=R
                    COUNT2=0
```

```
!   COUNTS EACH COMBINATION OF SIMULATION SPECS

                    DO 8 L=1,REPS

                        DONE=0

                        CONF=0


                        88 CONTINUE

! !   GENERATE X'S

                        DONE=1

                        RANGEX(1)=DRNUNF()

                        RANGEX(2)=DRNUNF()

                        X=0.0D0

                        XMAT=0.0D0

                        PS=5.0D-1

                        CALL RNBIN (N, NCOMP, PS, TEMPX)

                        DO 33 Q=1,N

                            IF (TEMPX(Q).EQ.0) THEN

                                X(Q,1)=1.0D0

                                XMAT(Q,2)=1.0D0

                            ELSEIF (TEMPX(Q).EQ.1) THEN

                                X(Q,2)=1.0D0

                                XMAT(Q,3)=1.0D0

                            ELSEIF (TEMPX(Q).EQ.2) THEN

                                X(Q,3)=1.0D0

                                XMAT(Q,4)=1.0D0

                            ELSEIF (TEMPX(Q).EQ.3) THEN

                                X(Q,4)=1.0D0

                                XMAT(Q,5)=1.0D0
                            ENDIF
                        33 CONTINUE
                        DO 34 Q=2,P
                            IF (SUM(XMAT(:,Q)).EQ.0) THEN
                                GOTO 88
                            ENDIF
                        34 CONTINUE

                        XMAT(:,1)=1.0D0
! CREATE C MATRIX
                        DO 11 K=1,P-1
                            DO 12 W=1,NCOMP
                                IF (W.EQ.K) THEN
                                    CMAT(W,K+1)=1.0D0
                                ELSE
                                    CMAT(W,K+1)=0.0D0
                                ENDIF
                            12 CONTINUE
                        11 CONTINUE
                        CMAT(:,1)=0.0D0
                        LINEARP=MATMUL(XMAT,BETAP)
                        LINEARCP=MATMUL(CMAT,BETAP)
    !   SIMULATE Y NOW
!   SIMULATE UNIFORM(0,1) RV'S
                        CALL DRNUN(N,UNIF)
!   SIMULATE RESPONSE Y
                        DO 10 I=1,N
```

```
                            IF (MODEL.EQ.3) THEN
                                MU(I)=DEXP(LINEARP(I))/(1+DEXP(LINEARP(I)))
                            ELSEIF (MODEL.EQ.0) THEN
                                MU(I)=DEXP(LINEARP(I))
                            ELSEIF (MODEL.EQ.4) THEN
                                MU(I)=DNORDF(LINEARP(I))
                            ENDIF
                            IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
                                IF (UNIF(I).LT.MU(I)) THEN
                                    X(I,P+1)=1.0D0
                                ELSE
                                    X(I,P+1)=0.0D0
                                ENDIF
                            ELSE
                                RANGEX(1)=DMIN1(X(I,1),RANGEX(1))
                                RANGEX(2)=DMAX1(X(I,1),RANGEX(2))
                                MEANTEMP=REAL(MU(I))
                                CALL RNPOI(N,MEANTEMP,PTEMP)
                                X(I,P+1)=DBLE(PTEMP(I))
                            ENDIF
                        10 CONTINUE


!PMLE CALCULATIONS START HERE
IF (MODEL.EQ.3) THEN
X(:,P)=1.0D0
ELSEIF (MODEL.EQ.0) THEN
X(:,P)=MU
```

```
ENDIF

WGT=0.0D0

LOGLIK1=0.0D0

LOGLIK2=0.0D0

MUSUM1=0.0D0

MUSUM2=0.0D0

XSUM=SUM(X(:,P+1))/N

YF=INT(X(:,P+1))

BETAF=0.0D0

IF (MODEL.EQ.3) THEN

BETAF(1)=DLOG((XSUM)/((1.0D0-XSUM+Small))+Small)

ELSEIF (MODEL.EQ.0) THEN

BETAF(1)=DLOG(XSUM+Small)

ENDIF

BETAF(2)=0.0D0

LINEAR=MATMUL(XMAT,BETAF)

DO 55 I=1,N

IF (MODEL.EQ.3) THEN

MUHAT(I)=DEXP(LINEAR(I))/

(1+DEXP(LINEAR(I)))

WGT(I,I)=DSQRT(MUHAT(I)*(1.0D0-MUHAT(I)))

IF (X(I,3).EQ.1.0D0) THEN

LOGLIK1=LOGLIK1+(DLOG(MUHAT(I)+Small))

ELSE

LOGLIK2=LOGLIK2+(DLOG(1-MUHAT(I)+Small))

ENDIF
```

```
ELSEIF (MODEL.EQ.0) THEN
MUHAT(I)=DEXP(LINEAR(I))
WGT(I,I)=DSQRT(MUHAT(I))
IF (YF(I).GT.169) THEN
YF(I)=169
 ENDIF
 FACT=DFAC(YF(I))
 LOGLIK1=LOGLIK1-MUHAT(I)+X(I,P+1)*
 DLOG(MUHAT(I)+Small)-DLOG(FACT)
 LOGLIK2=0.0D0
 ELSEIF (MODEL.EQ.4) THEN
 MUHAT(I)=DNORDF(LINEAR(I))
 WGT(I,I)=DSQRT(MUHAT(I)*(1.0D0-MUHAT(I)))
 ENDIF
 55 CONTINUE

 LOGLIK=LOGLIK1+LOGLIK2
 TEMP3=0
 H2=MATMUL(TRANSPOSE(XMAT),WGT)
 FISH=MATMUL(H2,TRANSPOSE(H2))
 CALL ERSET(0,0,0)
 CALL DLFTSF (P,FISH , LDA, FACU, LDA, IPVT)
!                Compute the determinant

CALL DLFDSF (P, FACU, LDA, IPVT, DET1, DET2)

LOGLIK=LOGLIK+5.0D-1*(DET1*1.0D1**DET2)
```

```
ITER=0

LOGLIKOLD=0.0D0

ITER=0

DO WHILE (ITER.LT.25)

ITER=ITER+1

H2=MATMUL(TRANSPOSE(XMAT),WGT)

FISH=MATMUL(H2,TRANSPOSE(H2))

CALL DLINRG(P,FISH,LDA,FCOV,LDAINV)

ICODE=IERCD()

IF ((ICODE.EQ.1).OR.(ICODE.EQ.2)) THEN

COUNT2=COUNT2+1

WRITE(8,*) ICODE,L,H,R,COUNT2

GOTO 88

ENDIF

H3=MATMUL(TRANSPOSE(H2),FCOV)

HAT=MATMUL(H3,H2)

LINEAR=MATMUL(XMAT,BETAF)

DO 66 O=1,N

IF (MODEL.EQ.3) THEN

MUHAT(O)=1.0D0/(1.0D0+DEXP(-LINEAR(O)))

XF(O,P+1)=X(O,P+1)-MUHAT(O)+HAT(O,O)*(5.0D-1+(1.0D0-MUHAT(O)))

ELSEIF (MODEL.EQ.0) THEN

MUHAT(O)=DEXP(LINEAR(O))

XF(O,P+1)=X(O,P+1)-MUHAT(O)+HAT(O,O)*5.0D-1

ENDIF

66 CONTINUE

USTAR=MATMUL(TRANSPOSE(XMAT),XF(:,P+1))
```

```
DELTAF=MATMUL(FCOV,USTAR)

MX=DMAX1(DABS(DELTAF(1)),DABS(DELTAF(2)))/10.0D0

IF (MX.GT.1.0D0) THEN

DELTAF=DELTAF/MX

ENDIF

BETAF=BETAF+DELTAF

LOGLIKOLD=LOGLIK

! DO HALF-STEPS

DONE=0

ITER2=0

LINEAR=MATMUL(XMAT,BETAF)

DO WHILE (DONE.EQ.0)

ITER2=ITER2+1

IF (MODEL.EQ.3) THEN

DO 65 I=1,N

MUHAT(I)=1.0D0/(1.0D0+DEXP(-LINEAR(I)))


WGT(I,I)=DSQRT(MUHAT(I)*(1.0D0-MUHAT(I)))

IF (X(I,P+1).EQ.1.0D0) THEN

LOGLIK1=LOGLIK1+(DLOG(MUHAT(I)+Small))

ELSE

LOGLIK2=LOGLIK2+(DLOG(1-MUHAT(I)+Small))

ENDIF

65 CONTINUE

ELSEIF (MODEL.EQ.0) THEN

DO 67 I=1,N

MUHAT(I)=DEXP(LINEAR(I))
```

```
WGT(I,I)=DSQRT(MUHAT(I))

LOGLIK1=LOGLIK1-MUHAT(I)+X(I,P+1)*

DLOG(MUHAT(I)+Small)-DLOG(FACT)

LOGLIK2=0.0D0

67 CONTINUE

ENDIF

LOGLIK=LOGLIK1+LOGLIK2

H2=MATMUL(TRANSPOSE(XMAT),WGT)

FISH=MATMUL(H2,XMAT)

CALL DLFTSF (P,FISH , LDA, FACU, LDA, IPVT)

!                             Compute the determinant


CALL DLFDSF (P, FACU, LDA, IPVT, DET1, DET2)

LOGLIK=LOGLIK+5.0D-1*(DET1*1.0D1**DET2)

IF ((LOGLIK.GT.LOGLIKOLD).OR.(ITER2.EQ.5)) THEN

DONE=1

ELSE

BETAF=BETAF-DELTAF*2.0D0**(-I)

ENDIF

ENDDO

IF (SUM(DABS(DELTAF)).LT.1.0D-2) THEN

ITER=2.5D5

ENDIF

ENDDO

LINEAR=MATMUL(XMAT,BETAF)
```

```
CALL DLINRG(P,FISH,P,FCOV,P)
!   THIS IS COV MAT OF BETAS (COV2)
COV2=FCOV(2:P,2:P)
!   COMPUTE INFORMATION MATRIX P.432 !   FIRST COMPUTE
LINEAR=X'BETAHAT
LINEARC=MATMUL(CMAT,BETAF)


!   COMPUTE SCALED INFO P.433


!   COMPUTE A INVERSE


AINV=COV2*N2
CALL DLINRG(NCOMP,AINV,NCOMP,A,NCOMP)


!   COMPUTE CHOLESKY DECOMP OF INFORMATION SCALED !   THIS SOLUTION
GIVES B'B=A
TOL=100*DMACH(4)
CALL DCHFAC(NCOMP,A,NCOMP,TOL,IRANK,B,NCOMP)
BT=TRANSPOSE(B)
!   COMPUTES BT_INV
CALL DLINRG(NCOMP,BT,NCOMP,BTINV,NCOMP)



!   START CALCULATION OF S(X) FOR ALL V=1,N !   NOTE THAT THERE IS A
UNIQUE S(X) FOR EACH UNIQUE X !   CALCULATE S(X) FROM P.435 CALL IT
S !   COMPUTE UI FROM P.437
```

```
IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN

B2=N2*(DEXP(LINEARC))/((1.0D0+DEXP(LINEARC))**2.0D0)

B3=N2*(DEXP(LINEARC)*(1.0D0-DEXP(LINEARC)))/

(1.0D0+DEXP(LINEARC))**3.0D0

B4=N2*((DEXP(LINEARC)-4.0D0*DEXP(2.0D0*LINEARC)

+DEXP(3.0D0*LINEARC)))/((1.0D0+DEXP(LINEARC))**4.0D0)

ELSEIF (MODEL.EQ.0) THEN

B2=DEXP(LINEARC)

B3=B2

B4=B2

ENDIF


DO 13 V=1,NCOMP


!   MUST DIVIDE UI BY N GIVEN HOW DATA IS ENTERED FOR CALCULATION OF

BETAS!!!!!

TEMP2=MATMUL(BTINV,CMAT(V,2:P))

CALL DVCAL (P,1/N2,TEMP2,1,UI(:,V),1)

TEMP2=MATMUL(AINV,CMAT(V,2:P))

TEMP3=DDOT(P,CMAT(V,2:P), 1, TEMP2, 1)

IF (TEMP3.LT.1.0D-5) THEN

TEMP3=1.0D-5

ENDIF

TEMP4=1/DSQRT(TEMP3)

!   THIS MULTIPLIES A SCALAR (TEMP4) BY VECTOR (UI) TO GET S(X) FOR

I

CALL DVCAL (P, TEMP4, UI(:,V), 1, S(:,V), 1)
```

```
13 CONTINUE


DO 30 I=1,NCOMP
DO 31 J=1,NCOMP
U(I,J)=DDOT(P,UI(:,I),1,UI(:,J),1)
!   DOT PRODUCT OF S(X) AND UI
SU(I,J)=DDOT(P,S(:,I),1,UI(:,J),1)
31 CONTINUE
30 CONTINUE


DO 14 V=1,NCOMP
!   NOTE THAT VAR(X’BLINEAR)=X*VAR(BETA)*X’ !   FORMULA FROM AGRESTI
P. 172
VAR(V)=FCOV(V,V)
!   SET K1 AND K3 TO 0 (SO IT DOESN’T BUILD ON PAST VALUES)
K1=0.0D0
K3=0.0D0
!   INITIALIZE C VECTOR
C=0.0D0


DO 20 I=1,NCOMP
!   C CALCS P.437
C(2)=C(2)+(B4(I)*(SU(V,I)**2.0D0)*U(I,I))
C(6)=C(6)+(B4(I)*(SU(V,I)**4.0D0))
C(9)=C(9)+((B2(I)**2.0D0)*SU(V,I)**4.0D0)
!   SEE P 436
K1=K1+(B3(I)*((SU(V,I)**3.0D0)-SU(V,I)*U(I,I)))
```

```
K3=K3+(B3(I)*(SU(V,I)**3.0D0))

DO 32 J=1,N

C(1)=C(1)+(B3(I)*B3(J)*SU(V,I)*SU(V,J)*

U(I,J)**2.0D0)

C(3)=C(3)+(B3(I)*B3(J)*(SU(V,I)**2.0D0)*

(SU(V,J)**2.0D0)*U(I,J))

C(4)=C(4)+(B3(I)*B3(J)*U(I,J)*(SU(V,I)**2.0D0)

*U(I,J)*U(J,J))

C(7)=C(7)+(B3(I)*B3(J)*SU(V,I)*(SU(V,J)

**3.0D0)*U(I,I))

C(8)=C(8)+(B3(I)*B3(J)*(SU(V,I)**3.0D0)*

(SU(V,J)**3.0D0))

32 CONTINUE

20  CONTINUE



!   FINISH CALCULATION OF C CONSTANTS FOR EDGEWORTH EXPANSIONS

C(1)=C(1)/(N2**3.0D0)

C(2)=C(2)/(N2**2.0D0)

C(3)=C(3)/(N2**3.0D0)

C(4)=C(4)/(N2**3.0D0)

C(5)=C(1)

C(6)=C(6)/(N2**2.0D0)

C(7)=C(7)/(N2**3.0D0)

C(8)=C(8)/(N2**3.0D0)

C(9)=C(9)/(N2**2.0D0)

K1=K1/(2.0D0*(N2**1.5D0))
```

```
K3=K3/(N2**1.5D0)


!   CONSTANTS USED IN P(X,Z) COMPUTATION P.437

KAP(1,V)=K1

TEMP3=C(1)-C(2)+C(4)-C(7)

KAP(2,V)=1.0D0+5.0D-1*TEMP3-3.0D0*C(3)+C(6)

 +1.75D0*C(8)

 KAP(3,V)=K3

 KAP(4,V)=-9.0D0*C(3)+3.0D0*C(6)+6.0D0*C(8)

 +3.0D0*C(9)


!   CONSTANTS USED IN Q(X,U) COMPUTATIONS P.438

KAQ(2,V)=C(3)-1.5D0*C(8)-C(5)-C(4)+5.0D-1*

C(7)+C(6)-C(2)

KAQ(3,V)=KAP(3,V)

KAQ(4,V)=-3.0D0*C(3)-6.0D0*C(4)-6.0D0*C(5)

+3.0D0*C(6)+3.0D0*C(7)-3.0D0*C(8)+3.0D0*C(9)

14 CONTINUE

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !

THIS SECTION NOW STARTS NAIMAN CRITICAL VALUE CALCS !   CREATE

MATRIX T THAT CONTAINS DOMAIN PARTITIONED LIKE ON P.441

!T(1,:)=0.0D0

K=1

INTER=1.0D0/(M-1.0D0)


COUNT=1

IF (P.EQ.5) THEN
```

```
DO 51 J=1,M
DO 53 K=1,M
DO 54 Q=1,M
DO 56 O=1,M
TEMPT(1,COUNT)=INTER*(J-1)
TEMPT(2,COUNT)=INTER*(K-1)
TEMPT(3,COUNT)=INTER*(Q-1)
TEMPT(4,COUNT)=INTER*(O-1)
COUNT=COUNT+1
56 CONTINUE
54 CONTINUE
53 CONTINUE
51 CONTINUE
 ENDIF

 COUNT=0
 DO 50 I=1,M**NCOMP
 SUMCOL=(SUM(TEMPT(:,I)))
 IF (SUMCOL.EQ.1) THEN
 COUNT=COUNT+1
 T(:,COUNT)=TEMPT(:,I)
  ENDIF
  SUMCOL=0.0D0
  50 CONTINUE
  KO=0.0D0
  KSI=0.0D0
  DO 40 K=1,(COUNT-1)
!   APPROXIMATE THE MANIFOLD VOLUME (K0) LIKE ON P.441 !   COMPUTE
S(T(K))
UITEMP=MATMUL(BTINV,T(:,K))
TEMP2=MATMUL(AINV,T(:,K))
TEMP3=DDOT(NCOMP,T(:,K), 1, TEMP2, 1)
IF (TEMP3.LT.1.0D-5) THEN
TEMP3=1.0D-5
ENDIF
TEMP4=1/DSQRT(TEMP3)
CALL DVCAL (NCOMP, TEMP4, UITEMP, 1, SECOND, 1)
!   COMPUTE S(T(K+1))
UITEMP=MATMUL(BTINV,T(:,K+1))
TEMP2=MATMUL(AINV,T(:,K+1))
TEMP3=DDOT(NCOMP,T(:,K+1), 1, TEMP2, 1)
IF (TEMP3.LT.1.0D-5) THEN
TEMP3=1.0D-5
ENDIF
TEMP4=1/DSQRT(TEMP3)
CALL DVCAL (NCOMP, TEMP4, UITEMP, 1, FIRST, 1)
!   COMPUTE DIFFERENCE OF S(T(K))-S(T(K+1))
DIF=FIRST-SECOND
INCX=1
!   GET THE NORM OF THE DIFF
NORMDIF=DNRM2(NCOMP,DIF,INCX)
!   ADD TO PREVIOUS NORMS TO APPROXIMATE K0
K0=K0+NORMDIF
```

```
40  CONTINUE

!   APPROXIMATE THE MANIFOLD SURFACE AREA (KSI)
!   TO DO THIS WE CALC VOLUME ON D-1 DIMENSIONS
DO 42 Q=1,NCOMP
DO 41 J=1,(COUNT-1)
TEMPT(:,J)=T(:,J)
TEMPT(:,J+1)=T(:,J+1)
TEMPT(Q,J)=0.0D0
TEMPT(Q,J+1)=0.0D0
UITEMP=MATMUL(BTINV,TEMPT(:,J))
!       CALL DLINRG(P,A,LDA,AINV,LDAINV)
TEMP2=MATMUL(AINV,TEMPT(:,J))
TEMP3=DDOT(NCOMP,TEMPT(:,J),1, TEMP2, 1)
IF (TEMP3.LT.1.0D-5) THEN
TEMP3=1.0D-5
ENDIF
TEMP4=1/DSQRT(TEMP3)
CALL DVCAL (NCOMP, TEMP4, UITEMP,1, SECOND, 1)
!   COMPUTE S(T(K+1))
UITEMP=MATMUL(BTINV,TEMPT(:,J+1))
!       CALL DLINRG(P,A,LDA,AINV,LDAINV)
TEMP2=MATMUL(AINV,TEMPT(:,J+1))
TEMP3=DDOT(NCOMP,TEMPT(:,J+1), 1,TEMP2, 1)
IF (TEMP3.LT.1.0D-5) THEN
TEMP3=1.0D-5
ENDIF
```

```
TEMP4=1/DSQRT(TEMP3)

CALL DVCAL (NCOMP, TEMP4, UITEMP,1, FIRST, 1)

!   COMPUTE DIFFERENCE OF S(T(K))-S(T(K+1))

DIF=FIRST-SECOND

INCX=1

!   GET THE NORM OF THE DIFF

NORMDIF=DNRM2(NCOMP,DIF,INCX)

!   ADD TO PREVIOUS NORMS TO APPROXIMATE K0

KSI=KSI+NORMDIF

41 CONTINUE

42 CONTINUE


!   CALCULATE CURVATURE FOR (NCOMP) CHOOSE (NCOMP-3) 3 TUPLES AND

TAKE THE MEAN

DO 76 V=1,(COUNT-1)

!   MUST DIVIDE UI BY N GIVEN HOW DATA IS ENTERED FOR CTGLM

SUBROUTINE!!!!!!

DO 99 J=1,P-2

DO 98 K=J+1,P

DO 97 I=0,P-J

GG=J

HH=K

TEMPT(:,V)=T(:,V)

TEMPT(GG,V)=0.0D0

TEMPT(HH,V)=0.0D0

TEMPT(:,V+1)=T(:,V+1)

TEMPT(GG+I,V+1)=0.0D0
TEMPT(HH+I,V+1)=0.0D0

TEMP2=MATMUL(BTINV,TEMPT(:,V))

CALL DVCAL (NCOMP,1/N2,TEMP2,1,UIT(:,V),1)

TEMP2=MATMUL(AINV,TEMPT(:,V))

TEMP3=DDOT(NCOMP,TEMPT(:,V),1, TEMP2, 1)

IF (TEMP3.LT.1.0D-5) THEN

TEMP3=1.0D-5

ENDIF

TEMP4=1/DSQRT(TEMP3)

!  THIS MULTIPLIES A SCALAR (TEMP4) BY VECTOR (UI) TO GET S(X) FOR

I

CALL DVCAL (NCOMP, TEMP4, UIT(:,V),1, ST(:,V), 1)

TEMP2=MATMUL(BTINV,TEMPT(:,V+1))

CALL DVCAL (NCOMP,1/N2,TEMP2,1,UIT(:,V+1),1)

TEMP2=MATMUL(AINV,TEMPT(:,V+1))

TEMP3=DDOT(NCOMP,TEMPT(:,V+1), 1,TEMP2, 1)

IF (TEMP3.LT.1.0D-5) THEN

TEMP3=1.0D-5

 ENDIF

 TEMP4=1/DSQRT(TEMP3)

!   HIS MULTIPLIES A SCALAR (TEMP4) BY VECTOR (UI) TO GET S(X) FOR

I

CALL DVCAL (NCOMP, TEMP4,UIT(:,V+1), 1, ST(:,V+1), 1)

CALL DVCAL(NCOMP,2.0D0,ST(:,V),1,TEMP5,1)

TEMP6=ST(:,V+1)-TEMP5+ST(:,V-1)

CALL DVCAL(NCOMP,INTER**2.0D0,TEMP6,1,K2VEC,1)

K2(V)=DNRM2(NCOMP,K2VEC,INCX)
```

```
97 CONTINUE

98 CONTINUE

99 CONTINUE

76 CONTINUE

K2MEAN=SUM(K2)


!   CALCULATE CURVATURE FOR (NCOMP) CHOOSE(NCOMP-3) 3 TUPLES AND TAKE THE MEAN

IF (P.EQ.5) THEN

DO 47 I=2,P

DO 49 J=2,P

DO 46 K=2,P

DO 48 V=1,(COUNT-1)

STEMP1=ST(:,V+1)

STEMP1(I)=0.0D0

 STEMP2=ST(:,V)

  STEMP2(J)=0.0D0

  STEMP3=ST(:,V-1)

  STEMP3(K)=0.0D0

  CALL DVCAL(NCOMP,2.0D0,STEMP2,1,TEMP5,1)

  TEMP6=STEMP1-TEMP5+STEMP3

  CALL DVCAL(NCOMP,INTER**2.0D0,TEMP6,1,KSI2VEC,1)

  KSI2=KSI2+DNRM2(NCOMP,KSI2VEC,INCX)

  48 CONTINUE

  46 CONTINUE

  49 CONTINUE

  47 CONTINUE

  ENDIF
  KSI2MEAN=KSI2/4.0D0


!   CALC ROTATION ANGLE HERE !   ASSUMES ANGLE BETWEEN ALL POSSIBLE
TWO SIDES MEETING
M0=0.0D0
INTER=1.0D0/(DBLE(P)-1.0D0)
DO 81 J=1,NCOMP-1
DO 82 K=1,2
DO 80 V=1,100,99
IF (K.EQ.1) THEN
TEMPT(:,V)=T(:,V)-INTER
TEMPT(J,V)=0.0D0
ELSE
TEMPT(:,V)=T(:,V)-INTER
TEMPT(J+1,V)=0.0D0
ENDIF
TEMP2=MATMUL(BTINV,TEMPT(:,V))
CALL DVCAL (NCOMP,1/N2,TEMP2,1,UIT(:,V),1)
TEMP2=MATMUL(AINV,TEMPT(:,V))
TEMP3=DDOT(NCOMP,TEMPT(:,V), 1, TEMP2, 1)
IF (TEMP3.LT.1.0D-5) THEN
TEMP3=1.0D-5
ENDIF
TEMP4=1/DSQRT(TEMP3)
!   THIS MULTIPLIES A SCALAR (TEMP4) BY VECTOR (UI) TO GET S(X) FOR
I
CALL DVCAL (NCOMP, TEMP4, UIT(:,V), 1, ST(:,V), 1)
```

```
IF (K.EQ.1) THEN

TEMPT(:,V+1)=T(:,V+1)+INTER

TEMPT(J,V+1)=0.0D0

ELSE

TEMPT(:,V+1)=T(:,V+1)+INTER

TEMPT(J+1,V+1)=0.0D0

ENDIF


TEMP2=MATMUL(BTINV,TEMPT(:,V+1))

CALL DVCAL (NCOMP,1/N2,TEMP2,1,UIT(:,V+1),1)

TEMP2=MATMUL(AINV,TEMPT(:,V+1))

TEMP3=DDOT(NCOMP,TEMPT(:,V+1), 1, TEMP2, 1)

IF (TEMP3.LT.1.0D-5) THEN

TEMP3=1.0D-5

ENDIF

TEMP4=1/DSQRT(TEMP3)

!   THIS MULTIPLIES A SCALAR (TEMP4) BY VECTOR (UI) TO GET S(X) FOR

I

CALL DVCAL (NCOMP, TEMP4, UIT(:,V+1), 1, ST(:,V+1), 1)


GRAD(:,V)=ST(:,V+1)-ST(:,V)


IF (V.EQ.1) THEN

CALL DVCAL(NCOMP,INTER,GRAD(:,V),1,GRADIENT1,1)

ELSE

CALL DVCAL(NCOMP,INTER,GRAD(:,V),1,GRADIENT2,1)
```

```
ENDIF



80 CONTINUE

!   NOW CHOOSE THREE POINTS WITHIN EACH SURFACE DEFINED BY THE

GRADIENT SO THAT !   WE CAN COMPUTE THE NORMAL VECTORS FOR EACH

SURFACE !   THEN THE CROSS PRODUCT OF (B-A)X(C-A) GIVES ME THE

NORMAL VECTOR (COMPUTE A NORMAL !   VECTOR FOR EACH SURFACE) THEN

USE COS(THETA) TO GET THETA

                            ! COMPUTE NORMAL VECTORWS TO EACH SURFACE

                            GRAD1 AND GRAD2


!   NOW CALCULATE THE ROTATION ANGLE BETWEEN EACH OF THE 4 CHOOSE 2

=6 FACES\


IF (J.EQ.1) THEN

CROSS(K,1)=GRADIENT1(4)*GRADIENT2(3)-

GRADIENT1(3)*GRADIENT2(4)

CROSS(K,2)=GRADIENT1(4)*GRADIENT2(2)-

GRADIENT1(2)*GRADIENT2(4)

CROSS(K,3)=GRADIENT1(3)*GRADIENT2(2)-

GRADIENT1(2)*GRADIENT2(3)

ELSEIF (J.EQ.2) THEN

CROSS(K,1)=GRADIENT1(4)*GRADIENT2(3)-

GRADIENT1(3)*GRADIENT2(4)

CROSS(K,2)=GRADIENT1(4)*GRADIENT2(1)-

GRADIENT1(1)*GRADIENT2(4)
```

```
CROSS(K,3)=GRADIENT1(3)*GRADIENT2(1)-
GRADIENT1(1)*GRADIENT2(3)
ELSEIF (J.EQ.3) THEN
CROSS(K,1)=GRADIENT1(4)*GRADIENT2(2)-
GRADIENT1(2)*GRADIENT2(4)
CROSS(K,2)=GRADIENT1(4)*GRADIENT2(1)-
GRADIENT1(1)*GRADIENT2(4)
CROSS(K,3)=GRADIENT1(2)*GRADIENT2(1)-
GRADIENT1(1)*GRADIENT2(2)
ELSEIF (J.EQ.4) THEN
CROSS(K,1)=GRADIENT1(3)*GRADIENT2(2)-
GRADIENT1(2)*GRADIENT2(3)
CROSS(K,2)=GRADIENT1(3)*GRADIENT2(1)-
GRADIENT1(1)*GRADIENT2(3)
CROSS(K,3)=GRADIENT1(2)*GRADIENT2(1)-
GRADIENT1(1)*GRADIENT2(2)
ENDIF
82 CONTINUE
CROSSDOT=DDOT(3,CROSS(1,:),1,CROSS(2,:),1)
NORM1=DNRM2(3,CROSS(1,:),INCX)
NORM2=DNRM2(3,CROSS(2,:),INCX)
COSTH=CROSSDOT/(NORM1*NORM2+Small)
THETA=DACOS(COSTH)
M0=M0+THETA
81 CONTINUE
```

```
!   NOW GET CONSTANT USING IDENDITY FROM SU(2000)




CRITLO=1.96D0

CRITHI=DSQRT(C10)

!   STARTING VALUES FOR SECANT METHOD

PART1=(K0*DGAMMA((DBLE(NCOMP)+1.0D0)/2.0D0))/

(PI**((DBLE(NCOMP)+1.0D0)/2.0D0))

PART2=1.0D0-DFDF(CRITLO**2.0D0/(DBLE(NCOMP)

+1.0D0),DBLE(NCOMP+1),DBLE(N-P))

PART3=(KSI*DGAMMA(DBLE(NCOMP)/2.0D0))/

(2.0D0*PI**(DBLE(NCOMP)/2.0D0))

PART4=1.0D0-DFDF(CRITLO**2.0D0/

DBLE(NCOMP),DBLE(NCOMP),DBLE(N-P))

PART5=((K2MEAN+KSI2MEAN+M0)*DGAMMA(

(DBLE(NCOMP)-1.0D0)/2.0D0))/

(2.0D0*PI*PI**((DBLE(NCOMP)-1.0D0)/2.0D0))

PART6=1.0D0-DFDF(CRITLO**2.0D0/(DBLE

(NCOMP)-1.0D0),DBLE(NCOMP-1),DBLE(N-P))

FX=PART1*PART2+PART3*PART4+PART5*PART6-ALPHA

PART1=(K0*DGAMMA((DBLE(NCOMP)+1.0D0)/2.0D0))

/(PI**((DBLE(NCOMP)+1.0D0)/2.0D0))

PART2=1.0D0-DFDF(CRITHI**2.0D0/(DBLE(NCOMP)

+1.0D0),DBLE(NCOMP+1),DBLE(N-P))

PART3=(KSI*DGAMMA(DBLE(NCOMP)/2.0D0))/

(2.0D0*PI**(DBLE(NCOMP)/2.0D0))
```

```
PART4=1.0D0-DFDF(CRITHI**2.0D0/DBLE(NCOMP),
DBLE(NCOMP),DBLE(N-P))
PART5=((K2MEAN+KSI2MEAN+M0)*DGAMMA(
(DBLE(NCOMP)-1.0D0)/2.0D0))/(2.0D0*PI*PI**
((DBLE(NCOMP)-1.0D0)/2.0D0))
PART6=1.0D0-DFDF(CRITHI**2.0D0/(DBLE(NCOMP)
-1.0D0),DBLE(NCOMP-1),DBLE(N-P))
FX1=PART1*PART2+PART3*PART4+PART5*PART6-ALPHA

K=1
ABSERR=1.0D0
SECANT: DO WHILE ((K.LT.MAXK).AND.(ABSERR.GT.EPSILON))
DF=(FX1-FX)/(CRITHI-CRITLO+Small)

DP=(FX1)/(DF+Small)
CRIT=CRITHI-DP
 PART1=(K0*DGAMMA((DBLE(NCOMP)+1.0D0)/2.0D0))
  /(PI**((DBLE(NCOMP)+1.0D0)/2.0D0))
  PART2=1.0D0-DFDF(CRIT**2.0D0/(DBLE(NCOMP)
  +1.0D0),DBLE(NCOMP+1),DBLE(N-P))
  PART3=(KSI*DGAMMA(DBLE(NCOMP)/2.0D0))/
  (2.0D0*PI**(DBLE(NCOMP)/2.0D0))
  PART4=1.0D0-DFDF(CRIT**2.0D0/DBLE(NCOMP),
  DBLE(NCOMP),DBLE(N-P))
  PART5=((K2MEAN+KSI2MEAN+M0)*DGAMMA((
  DBLE(NCOMP)-1.0D0)/2.0D0))/(2.0D0*PI*PI**
  ((DBLE(NCOMP)-1.0D0)/2.0D0))
  PART6=1.0D0-DFDF(CRIT**2.0D0/(DBLE(NCOMP)

  -1.0D0),DBLE(NCOMP-1),DBLE(N-P))

  TEMPFX=PART1*PART2+PART3*PART4+PART5*PART6-ALPHA

  ABSERR=DABS(TEMPFX)

  IF (TEMPFX.LT.0) THEN

  CRITHI=CRIT

  FX1=TEMPFX

  ELSE

  CRITLO=CRIT

  FX=TEMPFX

  ENDIF

  K=K+1

  ENDDO SECANT



  SUM1=0.0D0

  SUM2=0.0D0

  MEAN1=0.0D0

  MEAN2=0.0D0


  DO 60 V=1,NCOMP




!   CALCULATE CI FOR ETA=X'BETA THEN TRANSFORM TO GET CI ON MEAN

RESP
```

```
!   CONSTRUCT CI'S HERE (RECALL 4 SCB CI'S)
IF (VER.EQ.1) THEN
!   BASIC SCR (N IS NUMBER OF POINTS IN X TO MAKE PREDICTIONS
FOR-NEVER MIND FOR NOW!)

LO(V)=LINEARC(V)-(CRIT)*DSQRT(VAR(V))
HI(V)=LINEARC(V)+(CRIT)*DSQRT(VAR(V))
IF ((LINEARCP(V).GE.LO(V)).AND.
(LINEARCP(V).LE.HI(V))) THEN
CONF=CONF+1
ENDIF
!   CENTERED SCR SEE P.637 OF SUN(2001)
ELSEIF (VER.EQ.2) THEN
IF (KAP(2,V).LT.1.0D-5) THEN
KAP(2,V)=1.0D-5
ENDIF
LO(V)=LINEARC(V)-KAP(1,V)*DSQRT(VAR(V))
-(CRIT)*DSQRT(VAR(V))*DSQRT(KAP(2,V))
HI(V)=LINEARC(V)-KAP(1,V)*DSQRT(VAR(V))
+(CRIT)*DSQRT(VAR(V))*DSQRT(KAP(2,V))
SUM1=SUM1+KAP(1,V)
SUM2=SUM2+KAP(2,V)
IF ((LINEARCP(V).GE.LO(V)).AND.
(LINEARCP(V).LE.HI(V))) THEN
CONF=CONF+1
ENDIF
```

```
ELSEIF (VER.EQ.3) THEN

!   CORRECTED 2 SCR

        !U=SOLVE Q2 FOR U: |u|+q2=c

         XLO=0.0D0

         XHI=1.4D0

         !LOOP=V

         YLO=Q2(XLO,V)

         YHI=Q2(XHI,V)

         X1=XLO

         X2=XHI

         Y1=YLO

         Y2=YHI

         ABSERR=1.0D0

         COUNT=1

         SECANTLOOP: DO WHILE ((ABSERR.GT.EPSILON)

         .AND.(COUNT.LT.250))

         XT = X2 - ((X2-X1)*Y2)/(Y2-Y1+Small)

         YT = Q2(XT,V) - CRIT

         IF (YT*YLO>0) THEN

         XLO = XT

         YLO = YT

         ELSE

         XHI = XT

         YHI = YT

         ENDIF

         X1=XLO

         X2=XHI
         Y1=YLO
         Y2=YHI
         ABSERR=DABS(YT)
         COUNT=COUNT+1
         ENDDO SECANTLOOP
         UCRIT(V)=Q2(XT,V)-CRIT
         IF (DABS(UCRIT(V)).LT.1.0D-4) THEN
         UCRIT(V)=1.0D-4
         ENDIF
         LO(V)=LINEARC(V)-(DABS(UCRIT(V)))*DSQRT(VAR(V))
         HI(V)=LINEARC(V)+(DABS(UCRIT(V)))*DSQRT(VAR(V))
         SUM1=SUM1+UCRIT(V)
         IF ((LINEARCP(V).GE.LO(V)).AND.
         (LINEARCP(V).LE.HI(V))) THEN
         CONF=CONF+1
         ENDIF
         ELSEIF (VER.EQ.4) THEN
!   CORRECTED SCR !   GET MAX OF P2 BY SECANT METHOD
XLO=1.0D-2
XHI=2.0D0*CRIT
YLO=P2(XLO,V)
YHI=P2(XHI,V)
X1=XLO
X2=XHI
Y1=YLO
Y2=YHI
ABSERR=1.0D0
```

```
K=1
LOOP=V
SECANTLOOP2: DO WHILE ((K.LT.MAX).
AND.(ABSERR.GT.EPSILON))
XT = X2 - ((X2-X1)*Y2)/(Y2-Y1+Small)
YT = P2(XT,V)
IF (YT*YLO>0) THEN
XLO = XT
YLO = YT
ELSE
XHI = XT
YHI = YT
ENDIF
X1=XLO
X2=XHI
Y1=YLO
Y2=YHI
ABSERR=DABS(YT)
K=K+1
ENDDO SECANTLOOP2
MAX_P2(V) = DABS(P2(XT,V))
UCRIT(V)=CRIT-MAX_P2(V)
!   NOW USE SECANT METHOD AGAIN TO FIND CRIT FOR EQ.41 P. 440
CRITLO=1.96D0
!   THIS IS SCHEFFE SOLUTION
CRITHI=DSQRT(C10)
!   STARTING VALUES FOR SECANT METHOD
```

```
PART1=(K0*DGAMMA((DBLE(NCOMP)+1.0D0)/2.0D0))/
(PI**((DBLE(NCOMP)+1.0D0)/2.0D0))
PART2=1.0D0-DFDF((CRITLO-MAX_P2(V))**2.0D0/
(DBLE(NCOMP)+1.0D0),DBLE(NCOMP+1),DBLE(N-P))
PART3=(KSI*DGAMMA(DBLE(NCOMP)/2.0D0))/
(2.0D0*PI**(DBLE(NCOMP)/2.0D0))
PART4=1.0D0-DFDF((CRITLO-MAX_P2(V))**2.0D0/
DBLE(NCOMP),DBLE(NCOMP),DBLE(N-P))
PART5=((K2MEAN+KSI2MEAN+M0)*DGAMMA(
(DBLE(NCOMP)-1.0D0)/2.0D0))/(2.0D0*PI*
PI**((DBLE(NCOMP)-1.0D0)/2.0D0))
PART6=1.0D0-DFDF((CRITLO-MAX_P2(V))**2.0D0/
(DBLE(NCOMP)-1.0D0),DBLE(NCOMP-1),DBLE(N-P))
FX=PART1*PART2+PART3*PART4+PART5*PART6-ALPHA
PART1=(K0*DGAMMA((DBLE(NCOMP)+1.0D0)/2.0D0))/
(PI**((DBLE(NCOMP)+1.0D0)/2.0D0))
PART2=1.0D0-DFDF((CRITHI-MAX_P2(V))**2.0D0/
(DBLE(NCOMP)+1.0D0),DBLE(NCOMP+1),DBLE(N-P))
PART3=(KSI*DGAMMA(DBLE(NCOMP)/2.0D0))/
(2.0D0*PI**(DBLE(NCOMP)/2.0D0))
PART4=1.0D0-DFDF((CRITHI-MAX_P2(V))**2.0D0/
DBLE(NCOMP),DBLE(NCOMP),DBLE(N-P))
PART5=((K2MEAN+KSI2MEAN+M0)*DGAMMA(
(DBLE(NCOMP)-1.0D0)/2.0D0))/(2.0D0*PI*
PI**((DBLE(NCOMP)-1.0D0)/2.0D0))
PART6=1.0D0-DFDF((CRITHI-MAX_P2(V))**2.0D0/
(DBLE(NCOMP)-1.0D0),DBLE(NCOMP-1),DBLE(N-P))
```

```
FX1=PART1*PART2+PART3*PART4+PART5*PART6-ALPHA

K=1
ABSERR=1.0D0
 SECANT2: DO WHILE ((K.LT.MAXK).AND.(ABSERR.GT.EPSILON))
 DF=(FX1-FX)/(CRITHI-CRITLO+Small)
 DP=(FX1)/(DF+Small)
  CRIT=CRITHI-DP
  PART1=(K0*DGAMMA((DBLE(NCOMP)+1.0D0)/2.0D0))/(
  PI**((DBLE(NCOMP)+1.0D0)/2.0D0))
  PART2=1.0D0-DFDF((CRIT-MAX_P2(V))**2.0D0/
  (DBLE(NCOMP)+1.0D0),DBLE(NCOMP+1),DBLE(N-P))
  PART3=(KSI*DGAMMA(DBLE(NCOMP)/2.0D0))/
  (2.0D0*PI**(DBLE(NCOMP)/2.0D0))
  PART4=1.0D0-DFDF((CRIT-MAX_P2(V))**2.0D0/
  DBLE(NCOMP),DBLE(NCOMP),DBLE(N-P))
  PART5=((K2MEAN+KSI2MEAN+M0)*DGAMMA(
  (DBLE(NCOMP)-1.0D0)/2.0D0))/(2.0D0*PI*
  PI**((DBLE(NCOMP)-1.0D0)/2.0D0))
  PART6=1.0D0-DFDF((CRIT-MAX_P2(V))**2.0D0/
  (DBLE(NCOMP)-1.0D0),DBLE(NCOMP-1),DBLE(N-P))
  TEMPFX=PART1*PART2+PART3*PART4+PART5*PART6-ALPHA
  ABSERR=DABS(TEMPFX)
  IF (TEMPFX.LT.0) THEN
  CRITHI=CRIT
  FX1=TEMPFX
  ELSE
```

```
CRITLO=CRIT
FX=TEMPFX
ENDIF
K=K+1
ENDDO SECANT2
IF (CRIT.LT.1.0D-5) THEN
CRIT=1.0D-5
ENDIF

LO(V)=LINEARC(V)-(CRIT)*DSQRT(VAR(V))
HI(V)=LINEARC(V)+(CRIT)*DSQRT(VAR(V))
SUM2=SUM2+MAX_P2(V)
IF ((LINEARCP(V).GE.LO(V)).AND.(LINEARCP(V)
.LE.HI(V))) THEN
CONF=CONF+1
ENDIF
ENDIF
 60 CONTINUE
 ECL(L)=DBLE(CONF)/DBLE(NCOMP)
 8 CONTINUE
 MEANECL=SUM(ECL)/DBLE(REPS)
 RESULTS(1)=MODEL
 RESULTS(2)=H
 RESULTS(4)=N
 RESULTS(5)=MEANECL
 RESULTS(6)=VER
 WRITE(9,*) RESULTS
 WRITE(10,*) MEANECL
 CALL ERSET(0,2,2)
 7 CONTINUE
 3 CONTINUE
  2 CONTINUE
  END
```

```
 DOUBLE PRECISION FUNCTION P2(U,V)
 DOUBLE PRECISION U,KAP(5,4),KAQ(5,4),C(9),C2,N2,P3,
 PART1,PART2,PART3,V
 INTEGER LOOP
 COMMON KAP,KAQ,P3,N2,C,C2,LOOP
 PART1=KAP(2,V)-1.0D0+KAP(1,V)**2.0D0
 PART2=(KAP(4,V)+4.0D0*KAP(1,V)*KAP(V,3))*(U**2.0D0-3.0D0)
 PART3=KAP(3,V)*(U**4.0D0-1.0D1*U**2.0D0+1.5D1)
 P2=U*(3.6D1*PART1+3.0D0*PART2+PART3)/7.2D1
 RETURN
 END

 DOUBLE PRECISION FUNCTION Q2(U,V)
 DOUBLE PRECISION U,KAQ(5,4),KAP(5,4),P3,N2,C2,C(9),
 PART1,PART2,V

 INTEGER LOOP

 COMMON KAP,KAQ,P3,N2,C,C2,LOOP

 PART1=U*U-3.0D0

 PART2=(U*U-1.0D1)*U*U+1.5D1

 Q2=-U*(3.6D1*KAQ(2,V) + 3.0D0*KAQ(4,V)*PART1 +

  KAP(3,V)**2.0D0*PART2)/7.2D1

 RETURN

 END
```

VITA

Amy Wagler

Candidate for the Degree of

Doctor of Philosophy

Dissertation: SIMULTANEOUS INFERENCE IN GENERALIZED LINEAR MODEL SETTINGS

Major Field: Statistics

Biographical:

Personal Data: Born in Midwest City, Oklahoma, USA on March 29, 1974.

Education:

Received the B.S. degree from University of Texas of the Permian Basin, Odessa, Texas, USA, 1995, in Mathematics
Received the M.S. degree from Oklahoma State University, Stillwater, Oklahoma, USA, 2003, in Statistics
Completed the requirements for the degree of Doctor of Philosophy with a major in Statistics Oklahoma State University in July, 2007.

Research Interests: Simultaneous Inference, Categorical Data Analysis, Generalized Linear Models

Experience:

Graduate Teaching Associate,Oklahoma State University, 2003-2007. Regression Analysis, Engineering Statistics, and Statistical Methods II.
Graduate Research Associate, Oklahoma State University, 2002-2003. Worked under Dr. Christopher Bilder on NSF funded project.
Graduate Teaching Assistant, Oklahoma State University, 2000-2002. Engineering Statistics, Business Statistics, Elementary Statistics.

Name: Amy Wagler Date of Degree: July, 2007

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: SIMULTANEOUS INFERENCE IN GENERALIZED LINEAR MODEL SETTINGS

Pages in Study: 210 Candidate for the Degree of Doctor of Philosophy

Major Field: Statistics

Generalized Linear Models (GLM's) are utilized in a variety of statistical applications. Many times the estimated quantities from the models are of primary interest. These estimated quantities may include the mean response, odds ratio, relative risk, or attributable proportion. In these cases overall conclusions about these quantities may be desirable. Currently few sophisticated methods exist to simultaneously estimate these quantities from a GLM. I propose several methods of estimating these quantities simultaneously and compare them to the existing methods. Intervals for the expected response of the GLM and any set of linear combinations of the GLM are explored. Most existing methods emphasis the simultaneous estimation of the expected response; few consider estimation of the sets of regression parameters, and hence quantities such as the odds ratio or relative risk. Additionally, almost all intervals employ maximum likelihood estimators (MLEs) for the model parameters. MLEs are often biased estimators for GLMs, particularly at small sample sizes. Thus, another set of intervals is proposed that utilize an alternative estimator for the parameters, the penalized maximum likelihood estimator (pMLE). This estimator is very similar to the usual MLE, but it is shifted in order to account for the bias typically present in the MLE for GLMs. Various critical values of the simultaneous intervals are explored for both the MLE and pMLE based intervals. Emphasis is placed on scenarios where the sample size is small relative to the number of parameters being estimated. Simulation studies compare the various intervals and suggest general recommendations. The pMLE based intervals proposed exhibit superior performance, particularly at small and moderate sample sizes. While usual MLE based intervals typically do not attain the desired level of confidence at the small sample sizes, the pMLE based intervals do. Additionally, at moderate to large sample sizes the pMLE based intervals are, in many cases, less conservative than the usual MLE based intervals.

ADVISOR'S APPROVAL: ______________________________