IMPROVING APPLICABILITY OF THE NON-MONOTONE UNIFIED ESTIMATE FOR MISSING DATA

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

in

Statistics

University of Regina

By

David Luke Thiessen

Regina, Saskatchewan

November, 2023

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

David Luke Thiessen, candidate for the degree of **Doctor of Philosophy in Statistics,** has presented a thesis titled, ***Improving applicability of the non-monotone unified estimate for missing data,*** in an oral examination held on **August 23, 2023**. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:	Dr Asokan Mulayath Variyath, Memorial University of Newfoundland
Supervisor:	Dr Yang Zhao, Department of Mathematics and Statistics
Committee Member:	Dr Taehan Bae, Department of Mathematics and Statistics
Committee Member:	Dr Dianliang Deng, Department of Mathematics and Statistics
Committee Member:	Dr Yiyu Yao, Department of Computer Science
Chair of Defense:	Dr Chris Oriet, Faculty of Graduate Studies and Research

Abstract

In applied statistics missing data are a common problem. Performing a "complete case analysis" by removing individuals with missing data causes a loss of statistical power and can cause non-response bias. Inverse probability weighting is one method used to avoid non-response bias. However, when some individuals have partially observed data inverse probability weighting has only a limited ability to use this data. The unified approach (Zhao and Liu, 2021) is a modification of inverse probability weighting that uses "working models" to extract information from individuals with partially observed data. When the probability an individual has missing data can be accurately modeled but the distribution of the data is difficult to model the unified approach is an attractive option. In this thesis we review the theory of the unified estimate and its application to the Cox proportional hazards model for survival data. We present a new R program which can be used to easily fit the unified estimate for generalized linear models or Cox proportional hazards models. Possible hypothesis tests for the fit of the unified estimate and directions for future research are suggested.

Acknowledgements

I would like to start by thanking my parents for their continual love, support, and encouragement. Our regular conversations have helped me keep grounded and connected to the world outside academia and energized me to continue my studies.

Next, I want to deeply thank Dr. Yang Zhao for encouraging me to pursue a PhD and guiding and supporting me through the process. If not for her I probably would not have even begun the degree, let alone completed it.

I want to thank the other faculty and graduate students in the Math and Stats department at the University of Regina. During my studies I got to know many fantastic people studying many interesting topics. This helped me stay curious and open to new ideas. I am particuarly grateful to Dr. Andrei Volodin and Sarah Carnochan Naqvi for their help.

I recieved financial support through scholarships from Dr. Zhao and her NSERC grant, through scholarships, teaching assistantships, travel awards, and university teaching fellow positions from Department of Mathematics and Statistics at the University of Regina, through the Canada Student Financial Assistance Program, and through travel awards from the Statistical Society of Canada. Dr. Hadjistavropoulos and Dr. Tu shared datasets, provided background information on the data, and collaborated on publications.

Post Defense Acknowledgement

I want to thank Dr. Dianliang Deng, Dr. Taehan Bae, and Dr. Yiyu Yao for serving on my thesis committe. I also want to deeply thank Dr. Asokan Mulayath Variyath for serving as the External Examiner. Their comments and feedback have substantially improved the clarity and organization of this thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The unified approach (Zhao & Liu, 2021) is a technique for performing regression modeling with missing data. It is primarily used when there are multiple "missingness patterns" in the data. This occurs often in practice when data is collected from different sources and then combined. The focus of this work is to improve the applicability of the unified approach. We provide an approachable review of the theory of the unified approach, suggest some new methods to assess the fit of the models, provide an R package for calculating the unified approach in generalized linear models and Cox proportional hazards models, and give a detailed example of using the unified approach to estimate a population average.

The layout of this thesis is as follows. In Section 1.1 we give an introduction to the problem of missing data and some common methods to perform statistical analysis when data is missing. In Section 1.2 we describe three motivating datasets with missing data. In Chapter 2 we review some statistical background, including M-estimation, techniques for missing data adjustment, and the Cox model for survival data. In Chapter 3 we define the basic method of the unified approach and examine some of its properties. In this chapter we take a quite general approach by treating the class of M-estimators, which includes many important estimators. In Chapter 4 we review

the unified estimate of the Cox model and provide details on the adjustments that it requires. In Chapter 5 we describe an R package which can be used to fit the unified estimates on generalized linear models and Cox proportional hazard models. The R code itself is given in Appendix B. In Chapter 7 we propose some directions for future research.

Novel contributes in this work appear in Sections 3.5.4, 3.5.5, 4.2, and Chapters 5 and 7.

1.1 The Problem of Missing Data

In nearly any study in applied statistics, some data that were intended to be collected are instead missing. This occurs for a wide variety of reasons. For example, contact information for sampled individuals may be out-of-date, individuals may not be comfortable sharing personal information with interviewers, patients may miss appointments, field equipment may malfunction, etc. As almost all studies have some missing data, we must nevertheless find a way to perform reliable statistical analysis of the data. There is a large literature on missing data which we will now briefly summarise. For further reading and references see Little & Rubin (2019) and van Buuren (2018).

The simplest and most naive attempt to perform statistical analysis is to discard data from any individual with missing data, leaving only the "complete cases" in the data. The resulting dataset has no missing data and can be analysed using the originally chosen method. Complete case analysis has two major disadvantages. The first is the loss of sample size leads to a loss of statistical power. The second is that if individuals who have missing data are systemically different than individuals with observed data there could be significant bias introduced. For example, if women are more likely to answer a survey than men, a complete case analysis of the responses could be skewed

towards women even if the original sample was designed to have an equal distribution of men and women.

An early approach to address non-response bias was to use "Inverse Probability Weighting" (IPW) (Horvitz & Thompson, 1952) to rebalance the sample. By multiplying each individual's contribution to the estimate by the inverse of their probability of responding, the non-response bias is reduced or removed. For example, if men have a 50% chance of responding to the survey, then by multiplying each man's contributed information by 2 we allow each man to represent both himself and another non-responding man. This removes the non-response bias, although the loss of sample size still occurs.

A broad category of methods to deal with missing data is to replace each missing value with some other appropriately chosen value. This type of replacement is called "imputation". Several possible methods in this class are described in van Buuren (2018). One of the most common is "mean value imputation", where any missing values of a variable are replaced with the average of that variable. In some situations this can be better than a complete case analysis, but it places a large mass of observations at the mean. This causes the analysis to be over-confident of any estimates of the mean, as the natural variation that would be observed if there was no missing data has been replaced by observations clustered at the centre. Other methods of single imputation include regression imputation with or without random error added. The situations where imputation with a single value are appropriate are very limited.

An improvement over single imputation and the most popular method for missing data adjustment is Multiple Imputation (Rubin, 1986). In multiple imputation the analyst creates M imputed datasets, each with any missing values replaced by randomly drawn values that are intended to represent the range of values that would be observed if no data were missing. Each of the M datasets are then analysed as if they had no

missing values, then the results are "pooled" into a final conclusion. Because each of the M datasets leads to slightly different results, the variation in the M different results gives some indication of how large of an effect missingness is having on the estimate. Performing multiple imputation requires the analyst to specify either the joint distribution of the data or the distribution of missing variables conditional on the other variables. This is generally more difficult than specifying models for the probability an individual has observed data for inverse probability weighting (Seaman & White, 2013).

A recently proposed method for missing-data adjustment is the unified approach (Zhao & Liu, 2021). This method attempts to improve upon inverse probability weighting by defining additional "working models", which specify relationships between subsets of the variables in the study. By comparing these relationships on individuals with complete data and individuals with missing data, we can extract some information from the individuals with partially observed data and use that to improve the final estimate. Before we begin describing the unified approach in Chapter 3, we now give three examples of how missing data can occur in practice.

1.2 Motivating Data

Our motivating data example is the National Cancer Institute of Canada Clinical Trials Group (NCIC CTG) MA.5 dataset (Levine et al., 2005). This study was carried out to examine whether the combination of cyclophosphamide, epirubicin, and fluorouracil (CEF) was more effective than cyclophosphamide, methotrexate, and fluorouracil (CMF) in preventing relapse in women with early stage breast cancer. The study had 716 participants. 10-year follow-up showed that CEF was significantly more effective than CMF.

Recently there has been increasing interest in evaluating and incorporating patient

quality of life (QoL) in the analysis of treatment, rather than simply survival duration (Basch, 2013). A related topic is assessing the relationship between patient QoL and survival. In the MA5 data QoL was measured using the Breast Cancer Chemotherapy Questionnaire (BCQ) (Levine et al., 1988). QoL measurements in the MA.5 were taken at baseline and approximately once per month, with frequency falling over time. Despite this design, 169 of the 716 patients were missing BCQ measurements at baseline. This is a significant fraction of the data which cannot be used under standard complete-case analysis techniques.

Thiessen et al. (2022) used a Cox proportional hazards model (Cox, 1972) to assess whether the baseline BCQ measurements were predictive of relapse-free survival. To adjust for missing data they compared two methods: the unified approach and multiple imputation (Bartlett et al., 2015). They did not find a significant effect of BCQ on overall survival. However, when BCQ was included in the Cox model both the unified approach and multiple imputation were still able to detect a significant relation between survival and type of chemotherapy used in treatment. The complete case analysis which discarded individuals who were missing the baseline BCQ was unable to detect this significant effect because of the loss of power. For further details see Section 6.2.

Liu & Zhao (2022) used weighted generalized estimating equations (WGEE) with the unified approach to study the association between covariates and longitudinal measurements of the BCQ. They found the unified approach resulted in lower standard errors than the complete case GEE, but the results were extremely sensitive to the choice of correlation structure.

Chapter 2

Background

In this chapter we provide some background and references which will be used in later chapters.

2.1 M-Estimation

To facilitate development on a broad class of estimators we first recall some of the theory of M-estimation. M-estimators are also known as generalized estimating equations. M-estimation is applicable to a wide variety of estimation problems. For an overview of M-estimates see Stefanski & Boos (2002). For detailed mathematical results see Huber (1967), Huber (1973), and Serfling (1980).

Let $\psi(X, \beta)$ be a function. Let the random variable X have a distribution function F. Define a functional T of F, $T(F)$, as the solution to the population expected value equation,

$$0 = E[\psi(X, \beta)] = \int \psi(x, \beta) dF(x) = \int \psi(x, T(F)) dF(x)$$

Denote this solution as β^*. β^* will be the estimand we are trying to estimate. We estimate it by the solution to the sample estimating equations

$$0 = \frac{1}{n}\sum_{i=1}^{n}\psi(X_i, \hat{\beta}) \tag{2.1}$$

We call $\hat{\beta}$ an M-estimate. Many important estimators can be shown to be M-estimates. For example, if $l(X, \beta)$ is the log-likelihood function of X, taking ψ as the derivative of l results in the usual maximum likelihood estimate

$$0 = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\beta}l(X_i, \hat{\beta})$$

If $l(Y, X, \beta)$ is instead the log-likelihood of Y conditional on data X and parameters β, then the maximum likelihood estimate of a generalized linear regression also takes this form. For additional details on generalized linear regression see Dobson & Barnett (2018) or McCullagh (2019). Other estimators that can be viewed as M-estimates include the generalized estimating equations for longitudinal data and the Cox model for survival data.

2.1.1 Asymptotics of M-Estimates

We now review some results from the theory of M-estimators. Serfling (1980) Chapter 7 give theorems for the consistency and asymptotic normality of M-Estimates under two different sets of conditions. The first set assumes that the function ψ is monotone. The second set assumes that ψ is bounded and continuous. All results assume that X is i.i.d.

Define $\lambda_F(\beta) = E[\psi(X, \beta)] = \int \psi(X, \beta)dF(x)$ and $\lambda_{F_n}(\beta) = \int \psi(X, \beta)dF_n(x) = n^{-1}\sum_{i=1}^{n}\psi(X_i, \beta)$.

Lemma 2.1 (Serfling 1980 Lemma 7.2.1 A)**.** *Let β^* be an isolated root of $\lambda_F(\beta) = 0$. Let $\psi(X, \beta)$ be monotone in β. Then β^* is unique and any solution sequence $\{\hat{\beta}_n\}$ of the empirical equation $\lambda_{F_n}(\beta) = 0$ converges to β^* with probability 1. If, further,*

$\psi(X, \beta)$ is continuous in β in a neighborhood of β^*, then there exists such a solution sequence.

Lemma 2.2 (Serfling 1980 Lemma 7.2.1 B)**.** *Let β^* be an isolated root of $\lambda_F(\beta) = 0$. Let $\psi(X, \beta)$ be continuous in β and bounded. Then the empirical equation $\lambda_{F_n}(\beta) = 0$ has a solution sequence $\{\hat{\beta}_n\}$ which converges to β^* with probability 1.*

Theorem 2.1 (Serfling 1980 Theorem 7.2.2 A)**.** *Let β^* be an isolated root of $\lambda_F(\beta) = 0$. Let $\psi(X, \beta)$ be monotone in β. Suppose that $\lambda_F(\beta)$ is differentiable at $\beta = \beta^*$, with $\lambda'_F(\beta^*) \neq 0$. Suppose that $\int \psi^2(X, \beta) dF(x)$ is finite for β in a neighborhood of β^* and is continuous at $\beta = \beta^*$. Then any solution sequence $\hat{\beta}_n$ of the empirical equation $\lambda_{F_n}(\beta) = 0$ satisfies*

$$n^{1/2}(\hat{\beta}_n - \beta^*) \xrightarrow{d} N(0, \frac{\int \psi^2(X, \beta^*) dF(x)}{[\lambda'_F(\beta)]^2}$$

Theorem 2.2 (Serfling 1980 Theorem 7.2.2 B)**.** *Let β^* be an isolated root of $\lambda_F(\beta) = 0$. Let $\partial\psi(X, \beta)/\partial\beta$ be continuous at $\beta = \beta^*$ uniformly in X. Suppose that $\int(\partial\psi(X, \beta)/\partial\beta)|_{\beta^*} dF(x)$ is finite and nonzero, and that $\int \psi^2(X, \beta^*) dF(x) < \infty$. Let $\hat{\beta}_n$ be a solution sequence of $\lambda_{F_n}(\beta) = 0$ satisfying $\hat{\beta}_n \to \beta^*$. Then $\hat{\beta}_n$ satisfies*

$$n^{1/2}(\hat{\beta}_n - \beta^*) \xrightarrow{d} N(0, \frac{\int \psi^2(X, \beta^*) dF(x)}{[\int \partial\psi(X, \beta)/\partial\beta|_{\beta^*} dF(x)]^2}$$

Another version of the theorem is informally described where the condition of uniform continuity is relaxed to only require ψ is continuous but also imposes additional conditions on $\partial\psi(X, \beta)/\partial\beta$.

We will use the following multi-variate generalization of Theorems 2.1 and 2.2,

$$\sqrt{n}(\hat{\beta} - \beta^*) \xrightarrow{d} N(0, V(\beta^*))$$

Where

$$
\begin{aligned}
V(\beta^*) =& A(\beta^*)^{-1}B(\beta^*)A(\beta^*)^{T-1} \\
A(\beta^*) =& E[\frac{-\partial}{\partial\beta^T}\psi(X,\beta)|_{\beta^*}] \\
B(\beta^*) =& E[\psi(X,\beta^*)\psi(X,\beta^*)^T]
\end{aligned}
$$

Note that when $\psi(X,\beta) = \frac{\partial}{\partial\beta}\log(f(X,\beta))$ for a probability density function f and the data follows that distribution, then $\hat{\beta}$ is a MLE, $E[\frac{-\partial}{\partial\beta}\psi(X,\beta^*)] = E[\psi(X,\beta^*)\psi(X,\beta^*)^T]$, and $V(\beta^*)$ reduces to the usual variance for a maximum likelihood estimate.

One way to estimate the variance of $\hat{\beta}$ is to replace the matrices A and B in $V(\beta^*)$ with their sample estimates. This gives the so-called "sandwich estimate" of the variance of $\hat{\beta}$:

$$
\begin{aligned}
\hat{V}(\beta^*) &= \hat{A}(\hat{\beta})^{-1}\hat{B}(\hat{\beta})\hat{A}(\hat{\beta})^{T-1} \\
\hat{A}(\hat{\beta}) &= \frac{1}{n}\sum_{i=1}^{n}\frac{-\partial}{\partial\beta^T}\psi(X_i,\hat{\beta}) \\
\hat{B}(\hat{\beta}) &= \frac{1}{n}\sum_{i=1}^{n}\psi(X_i,\hat{\beta})\psi(X_i,\hat{\beta})^T
\end{aligned}
$$

The sandwich estimate will be one of the ways we estimate the variance of the unified estimate. Note that even if $\hat{A}$ and $\hat{B}$ are unbiased estimators of A and B, $\hat{A}^{-1}$ will not be an unbiased estimator of A^{-1} and $\hat{V}$ will not be an unbiased estimator of V. However, as long as $\hat{A}$ and $\hat{B}$ are consistent estimators of A and B then $\hat{V}$ will be a consistent estimator of V by the continuous mapping theorem (Mann & Wald, 1943).

2.2 Missing Data

Next we present some common notation and definitions for missing data. For more details and references see Little & Rubin (2019).

Typically in the missing data literature R_i is an indicator variable taking the value 1 if the ith individual has completely observed data and 0 otherwise. Sometimes each individual variable is given its own indicator, and we do that here. Define R_{ip} as an indicator variable for if the ith individual has variable p observed. In this work we will also need to consider multiple models having different observation patterns. Further details on notation will be given in Section 3.2.

The probability that an individual has observed or missing data is fundamental to analysis of missing data. Rubin (1976) defined three forms of this probability.

Missing Completely At Random (MCAR) is when the probability an individual has observed data is independent of the true values of the data: $P(R|X, \phi) = P(R|\phi)$.

Missing At Random (MAR) is when the probability an individual has observed data may depend on any observed values of the data, but not on any missing values: $P(R|X, \phi) = P(R|X^{obs}, X^{mis}, \phi) = P(R|X^{obs}, \phi)$

Missing Not At Random (MNAR) is when neither of the previous situations hold. In this case the probability an individual has missing data depends on the unknown values.

Throughout this work we will assume that data is either MCAR or MAR.

Additionally, we may define the overall structure of missing data as either "monotone" or "non-monotone". If it is possible to order the columns of data $X_1, ..., X_p$ in such a way that X_k being observed for an individual implies that X_l is also observed for all $l < k$, then the data is said to have a monotone missingness pattern. This is common

in longitudinal studies when some individuals drop out of the study and have no data after that point. If the columns of data cannot be ordered in such a way the data is said to have a non-monotone pattern.

Some techniques developed for missing data are only applicable when the data has a monotone missing pattern. The unified estimate is able to use data that has either a monotone or a non-monotone missing pattern.

2.3 Inverse Probability Weighting

One of the early methods used to correct for bias in missing data is inverse probability weighting (IPW). This is an extension of the theory from Horvitz & Thompson (1952) developed for unequal sampling. By giving observations with complete data weights inversely proportional to their probability of having complete data, the sample is rebalanced to what it would be if there were no missing data.

To give a brief mathematical justification, recall that in M-estimation the estimand we are looking for is the solution to the population estimating equations $0 = E[\psi(X, \beta^*)]$, estimated as the solution to the sample estimating equations $0 = \frac{1}{n}\sum_{i=1}^{n}\psi(x_i, \hat{\beta})$. If R_i is an indicator for observed data, π_i is the probability of data being observed for the ith individual, and π is bounded away from zero for all i, $\pi_i \geq \epsilon > 0$, then $E[\frac{R}{\pi}\psi(X, \beta)] = E[\psi(X, \beta)]$, and so the solution to the inverse probability estimating equations is the same as the solution to the full data estimating equations, $0 = E[\frac{R}{\pi}\psi(X, \beta^*)]$.

2.3.1 Estimation of Observation Probabilities in IPW

The true probabilities π_i are rarely known. They can be estimated from the same data used to fit the IPW adjusted model. Furthermore, using estimated probability weights is always at least as efficient as using the true weights (Robins et al., 1994). However, the model for the observation probabilities must itself be accurate and not

subject to non-response bias. This can be difficult when data are non-monotone missing. One way to avoid non-response bias is to assume that missingness only depends on fully observed covariates. However, Sun & Tchetgen Tchetgen (2018) show that using only fully observed covariates is a more restrictive assumption than MAR. They propose an estimator for observation probabilities with non-monotone missing data by fitting models for the probability that an individual has missingness pattern L for all missingness patterns that occur in the data. They then calculate the probability an individual has completely observed data by taking the complement of the sum of the missingness pattern probabilities. This method is quite general in theory, but in practice it can fail to converge or give estimated probabilities that are negative. On the other hand, Zhao (2021) propose using the EM algorithm to maximize a pseudo-likelihood with non-parametric modeling of the covariate distribution. This could be used to fit a model for observation probabilities to non-monotone missing data.

In this thesis we will focus on the simpler situation where a parametric model for π is used which only depends on fully observed covariates. If $\pi(x, \alpha)$ is a such a model with a parameter α, then estimating equations $h(x, \alpha)$ can be defined via

$$0 = E[h(x, \alpha^*)] = E\left[\frac{\partial}{\partial \alpha} R \log(\pi(x, \alpha^*)) + (1 - R) \log(1 - \pi(x, \alpha^*)\right]$$

The estimate $\hat{\alpha}$ is found as the solution to the sample estimating equations.

$$0 = \frac{1}{n} \sum_{i=1}^{n} h(x_i, \hat{\alpha})$$

Using $\hat{\alpha}$ to estimate the observation probabilities, $\hat{\pi}_i = \pi(x_i, \hat{\alpha})$, the IPW estimating

equations for the analysis model become

$$0 = E[\frac{R}{\pi(x, \alpha^*)}\psi(x, \beta^*)]$$

And $\hat{\beta}$ is the solution to the sample estimating equations,

$$0 = \frac{1}{n}\sum_{i=1}^{n}\frac{R_i}{\pi(x_i, \hat{\alpha})}\psi(x_i, \hat{\beta})$$

For notational simplicity, define the weighted complete case estimating equations as $f = f(x, \beta, \alpha, R) = \frac{R}{\pi(x,\alpha)}\psi(x, \beta)$.

To derive the theoretical properties of the IPW estimate using estimated probabilities we can stack the parameters and estimating equations from the IPW model and the probability model and treat the combination again as an M-estimator. Let $\theta = (\beta, \alpha)$ and the estimand θ^* be defined via estimating equations $u = (f, h)$,

$$0 = E[u(x, \theta^*)] = E\begin{bmatrix} f(x, \beta^*, \alpha^*, R) \\ h(x, \alpha^*) \end{bmatrix}$$

2.3.2 Variance Estimation in IPW

Whether the observation probabilities are known or estimated, the variance of the estimate $\hat{\beta}$ can be found using the sandwich formula for the variance from M-estimators.

$$V[\sqrt{n}\hat{\theta}] = V(\theta^*) = A(\theta^*)^{-1}B(\theta^*)A(\theta^*)^{T-1}$$

If the observation probabilities are estimated then we have

$$A(\theta^*) = E[\frac{-\partial}{\partial \theta^T} U(x, \theta^*)] = E \begin{bmatrix} \frac{-\partial}{\partial \beta^T} f(x, \beta^*, \alpha^*, R) & \frac{-\partial}{\partial \alpha^T} f(x, \beta^*, \alpha^*, R) \\ \frac{-\partial}{\partial \beta^T} h(x, \alpha^*) & \frac{-\partial}{\partial \alpha^T} h(x, \alpha^*) \end{bmatrix}$$

$$B(\theta^*) = E \begin{bmatrix} f(x, \beta^*, \alpha^*, R) f(x, \beta^*, \alpha^*, R)^T & f(x, \beta^*, \alpha^*, R) h(x, \alpha^*)^T \\ h(x, \alpha^*) f(x, \beta^*, \alpha^*, R)^T & h(x, \alpha^*) h(x, \alpha^*)^T \end{bmatrix}$$

The variance of $\sqrt{n}\hat{\beta}$ is the upper left block of $V(\theta^*)$. Note that in this case the bottom left block of $A(\theta^*)$ is 0 and $A(\theta^*)$ is a block triangular matrix.

Robins et al. (1994) give regularity conditions which guarantee several useful properties. They guarantee the validity of Taylor Series expansions of $u(x, \theta)$, the so-called "generalized information equality", $E[\frac{-\partial}{\partial \alpha^T} f(x, \beta^*, \alpha^*, R)] = E[f(x, \beta^*, \alpha^*, R) h(x, \alpha^*)^T]$, and the usual information equality, $E[\frac{-\partial}{\partial \alpha^T} h(x, \alpha^*)] = E[h(x, \alpha^*) h(x, \alpha^*)^T]$. Following their arguments and simplifying the upper left block of $V(\theta^*)$, we find that $\sqrt{n}(\hat{\beta} - \beta^*)$ with estimated observation probabilities is asymptotically normally distributed with variance,

$$V[\sqrt{n}(\hat{\beta} - \beta^*)] = E\left[\frac{-\partial}{\partial \beta^T} f\right]^{-1} \left[E[ff^T] - E[\frac{-\partial f}{\partial \alpha^T}] E[\frac{-\partial h}{\partial \alpha^T}]^{-1} E[\frac{-\partial f}{\partial \alpha^T}]^T\right] E\left[\frac{-\partial}{\partial \beta^T} f\right]^{T-1}$$

The second part of the middle term is positive semi-definite. Therefore, the variance using estimated observation probabilities is asymptotically at least as efficient as the estimate using the true observation probability probabilities.

The variance can be consistently estimated as,

$$\hat{V}[\sqrt{n}(\hat{\beta}-\beta^*)] = \left[n^{-1}\sum_{i=1}^{n}\frac{-\partial f}{\partial \beta^T}\right]^{-1} \left[(n^{-1}\sum_{i=1}^{n} ff^T) - (n^{-1}\sum_{i=1}^{n}\frac{-\partial f}{\partial \alpha^T})(n^{-1}\sum_{i=1}^{n}\frac{-\partial h}{\partial \alpha^T})^{-1}(n^{-1}\sum_{i=1}^{n}\frac{-\partial f}{\partial \alpha^T})^T\right] \left[n^{-1}\sum_{i=1}^{n}\frac{-\partial f}{\partial \beta^T}\right]^{T-1}$$

Where f and h are estimated using $\hat{\beta}$ and $\hat{\alpha}$.

However, this variance estimate may not be positive definite (Robins et al., 1995). They recommend using the fact that under their regularity conditions, $E[\frac{-\partial}{\partial \alpha^T} f(x,\beta^*,\alpha^*,R)] = E[f(x,\beta^*,\alpha^*,R)h(x,\alpha^*)^T]$ and $E[\frac{-\partial}{\partial \alpha^T} h(x,\alpha^*)] = E[h(x,\alpha^*)h(x,\alpha^*)^T]$. Substituting this into the estimate leads to the alternate variance estimate,

$$\begin{aligned}\hat{V}[\sqrt{n}(\hat{\beta}-\beta^*)] &= \left[n^{-1}\sum_{i=1}^{n}\frac{-\partial f}{\partial \beta^T}\right]^{-1} \left[(n^{-1}\sum_{i=1}^{n} ff^T) - (n^{-1}\sum_{i=1}^{n} fh^T)(n^{-1}\sum_{i=1}^{n} hh^T)^{-1}(n^{-1}\sum_{i=1}^{n} fh^T)^T\right] \left[n^{-1}\sum_{i=1}^{n}\frac{-\partial f}{\partial \beta^T}\right]^{T-1} \\ &= \left[n^{-1}\sum_{i=1}^{n}\frac{-\partial f}{\partial \beta^T}\right]^{-1} \left[n^{-1}\sum_{i=1}^{n}\left(f_i - (\sum_{j=1}^{n} f_j h_j^T)(\sum_{j=1}^{n} h_j h_j^T)^{-1} h_i\right)^{\otimes 2}\right] \left[n^{-1}\sum_{i=1}^{n}\frac{-\partial f}{\partial \beta^T}\right]^{T-1}\end{aligned}$$

Where for a vector x, $x^{\otimes 2} = xx^T$. The middle term of this new variance estimate can be considered as the matrix of crossproducts of residuals from a linear regression of f on h. This variance estimate is consistent and positive definite. Robins et

al. (1994) also define the functions $Resid(A_i, B_i) = A_i - E[A_i B_i^T]E[B_i B_i^T]^{-1} B_i$ and $\hat{R}esid(A_i, B_i) = A_i - (\sum_j A_j B_j^T)(\sum_j B_j B_j^T)^{-1} B_i$, which represent the residuals from a least squares regression of A_i on B_i. This leads to the slightly shorter expression

$$\hat{V}[\sqrt{n}(\hat{\beta} - \beta^*)] = \left[n^{-1} \sum_{i=1}^{n} \frac{-\partial f}{\partial \beta^T} \right]^{-1} \left[n^{-1} \sum_{i=1}^{n} \hat{R}esid(f_i, h_i)^{\otimes 2} \right] \left[n^{-1} \sum_{i=1}^{n} \frac{-\partial f}{\partial \beta^T} \right]^{T-1}$$

Finally, notice that if we make the information equality substitutions directly into $A(\theta^*)$ and use the sample estimate,

$$\tilde{A} = \begin{pmatrix} n^{-1} \sum \frac{-\partial f}{\partial \beta^T} & n^{-1} \sum f h^T \\ 0 & n^{-1} \sum h h^T \end{pmatrix}$$

Then the upper left block of

$$\tilde{V}[\sqrt{n}(\hat{\beta} - \beta^*)] = \tilde{A}^{-1} \hat{B} \tilde{A}^{T-1}$$

gives the desired positive definite variance estimate. Although expressing the calculation in this form involves some unnecessary calculation of the other blocks it will be convenient to use this form later.

2.3.3 Limitation of IPW Estimate

The IPW estimate is limited in its ability to use information from individuals with partially observed data. If the partially observed information can be used to estimate the observation probabilities then some efficiency can be recovered, as seen in the smaller asymptotic variance of the IPW estimate compared to the complete case estimate. But, if the partially observed data does not contribute to the estimation of weights, then it can not be used in the analysis.

2.4 Cox Proportional Hazards

The Cox proportional hazards model (Cox, 1972) is the most widely used model in survival analysis. Counting process notation, introduced by Andersen & Gill (1982), is quite general and commonly used for proofs in survival analysis. Each individual is viewed as having two counting processes. $N_i(t)$ counts the number of events observed for the ith individual up to and including time t. $Y_i(t)$ indicates if the ith individual was under observation just before time t. The covariates may also depend on time and be denoted as $X(t)$. For simplicity we will only consider fixed covariates X. For further details and references see Therneau & Grambsch (2000) and Kalbfleisch & Prentice (2002). Throughout we assume that failure times are continuous and there are no tied event times.

The Cox model specifies that the hazard function $\lambda(t)$ for an individual conditional on covariates x takes the form

$$\lambda(t|x) = \lambda_0(t) \exp(\beta^T x)$$

Where λ_0 is an unspecified baseline hazard and β is a vector of parameters.

The Cox partial likelihood for the proportional hazards model is

$$PL(\beta) = \prod_{i=1}^{n} \prod_{t \geq 0} \left(\frac{Y_i(t) \exp(\beta^T x_i)}{\sum_{j=1}^{n} Y_j(t) \exp(\beta^T x_j)} \right)^{dN_i(t)}$$

To simplify notation, define for $k = 0, 1, 2$

$$S^{(k)}(\beta, t) = \frac{1}{n} \sum_{i=1}^{n} x_i^{\otimes k} Y_i(t) \exp(\beta^T x_i)$$

Where for a vector a, $a^{\otimes 0} = 1$, $a^{\otimes 1} = a$, $a^{\otimes 2} = aa^T$. The weighted average covariate

vector of subjects at risk at time t^-, weighted by their relative risk of failure, is

$$\bar{x}(\beta, t) = \frac{\sum_{j=1}^{n} x_j Y_j(t) \exp(\beta^T x_j)}{\sum_{j=1}^{n} Y_j(t) \exp(\beta^T x_j)} = \frac{S^{(1)}(\beta, t)}{S^{(0)}(\beta, t)}$$

Taking the logarithm of the partial likelihood and differentiating with respect to β, we find the Cox estimating equations,

$$U(\beta) = \sum_{i=1}^{n} \int_{o}^{\infty} \left(x_i - \frac{S^{(1)}(\beta, t)}{S^{(0)}(\beta, t)} \right) dN_i(t) = \sum_{i=1}^{n} \int_{0}^{\infty} \left(x_i - \bar{x}(\beta, t_i) \right) dN_i(t)$$

$\hat{\beta}$ is found as the solution to $U(\beta) = 0$. The solution can be found using numeric methods such as the Newton-Raphson method.

2.4.1 Variance Estimates

Taking negative the second derivative of the log likelihood gives the information matrix.

$$\begin{aligned} I(\beta) &= \frac{-\partial U}{\partial \beta^T} \\ &= \sum_{i=1}^{n} \int_{0}^{\infty} \frac{\sum_{j=1}^{n} Y_j(t) \exp(\beta^T x_j) \left[x_j - \bar{x}(\beta, t) \right] \left[x_j - \bar{x}(\beta, t) \right]^T}{\sum_{j=1}^{n} Y_j(t) \exp(\beta^T x_j)} dN_i(t) \\ &= \sum_{i=1}^{n} \int_{0}^{\infty} \left[\frac{S^{(2)}(\beta, t)}{S^{(0)}(\beta, t)} - \frac{(S^{(1)}(\beta, t))^{\otimes 2}}{(S^{(0)}(\beta, t))^2} \right] dN_i(t) \end{aligned}$$

If the model is correctly specified then the asymptotic variance of $\hat{\beta}$ is given by $E[I(\beta^*)]^{-1}$. However, calculating this expectation is difficult and requires knowledge of the censoring process. Instead the inverse of the observed information is used,

$\hat{V}[\hat{\beta}] = (I(\hat{\beta}))^{-1}$.

$$
\begin{aligned}
I(\hat{\beta}) &= \sum_{i=1}^{n} \int_0^{\infty} \frac{\sum_{j=1}^{n} Y_j(t) \exp(\hat{\beta}^T x_j) \left[x_j - \bar{x}(\hat{\beta}, t)\right] \left[x_j - \bar{x}(\hat{\beta}, t)\right]^T}{\sum_{j=1}^{n} Y_j(t) \exp(\hat{\beta}^T x_j)} dN_i(t) \\
&= \sum_{i=1}^{n} \int_0^{\infty} \left[\frac{S^{(2)}(\hat{\beta}, t)}{S^{(0)}(\hat{\beta}, t)} - \frac{(S^{(1)}(\hat{\beta}, t))^{\otimes 2}}{(S^{(0)}(\hat{\beta}, t))^2} \right] dN_i(t)
\end{aligned}
$$

2.4.2 Inverse Probability Weighting in Cox Models

The use of inverse probability weights for missing data adjustment in the Cox model has been studied by Pugh et al. (1993) and Qi et al. (2005). As before, R_i is an indicator variable indicating if the ith subject has fully observed data. π_i is the probability that the ith individual has fully observed data. With inverse probability weights, $w_i = R_i/\pi_i = R_i/\pi(\alpha, x_i)$, the above Cox estimating equations become,

$$
\begin{aligned}
U_w(\beta) &= \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \bar{x}_w(\beta, t)\right] dN_i(t) \\
\bar{x}_w(\beta, t) &= \frac{\sum_{j=1}^{n} w_j x_j Y_j(t) \exp(\beta^T x_j)}{\sum_{j=1}^{n} w_j Y_j(t) \exp(\beta^T x_j)}
\end{aligned}
$$

Define $\mu(\beta, t) = E[\bar{x}(\beta, t)]$ and $\mu_w(\beta, t) = E[\bar{x}_w(\beta, t)]$. Pugh et al. (1993) show that the following four sets of estimating equations are asymptotically equivalent.

$$
\begin{aligned}
n^{-1} U(\beta) &= n^{-1} \sum_{i=1}^{n} \int_0^{\infty} \left[x_i - \bar{x}(\beta, t)\right] dN_i(t) \\
n^{-1} U^{\mu}(\beta) &= n^{-1} \sum_{i=1}^{n} \int_0^{\infty} \left[x_i - \mu(\beta, t)\right] dN_i(t) \\
n^{-1} U_w(\beta) &= n^{-1} \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \bar{x}_w(\beta, t)\right] dN_i(t) \\
n^{-1} U_w^{\mu}(\beta) &= n^{-1} \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \mu_w(\beta, t)\right] dN_i(t)
\end{aligned}
$$

Where a superscript μ indicates that $\bar{x}$ or $\bar{x}_w$ has been replaced by its expected value. Therefore the solution to the weighted estimating equations is asymptotically

equivalent to the solution from the desired full data estimating equations. Furthermore, define $A_i(t) = \int_0^t Y_i(u) \exp(\beta^T x_i) \lambda_0(u) du$. $A_i(t)$ is the compensator for the counting process $N_i(t)$. Then $M_i(t) = N_i(t) - A_i(t)$ is a mean-zero martingale. It is also easy to show that

$$\sum_{i=1}^{n} \int_0^{\infty} [x_i - \bar{x}(\beta, t)] \, dA_i(t) = 0$$

Therefore we also can replace N_i by M_i in the above estimating equations.

$$\begin{aligned} U_{wM}(\beta) &= \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \bar{x}_w(\beta, t)\right] dM_i(t) \\ &= \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \bar{x}_w(\beta, t)\right] d\left[N_i(t) - A_i(t)\right] \\ &= \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \bar{x}_w(\beta, t)\right] d\left[N_i(t) - \int_0^t Y_i(u) \exp(\beta^T x_i) \lambda_0(u) du\right] \end{aligned}$$

As in Qi et al. (2005) $\lambda_0(t)$ is replaced by a modified Breslow estimate of baseline hazard (Breslow, 1974), which is now weighted by the inverse of the observation probabilities. Note that this differs slightly from the estimate used in Pugh et al. (1993), who do not appear to weight the numerator.

$$\begin{aligned} \hat{\Lambda}_0(t) &= \int_0^t \frac{\sum_{j=1}^n w_j dN_j(u)}{\sum_{j=1}^n w_j Y_j(u) \exp(\beta^T x_j)} \\ \hat{\lambda}_0(t) &= \frac{\sum_{j=1}^n w_j dN_j(t)}{\sum_{j=1}^n w_j Y_j(t) \exp(\beta^T x_j)} \end{aligned}$$

Inserting this estimate in $U_{wM}(\beta)$ leads to

$$\begin{aligned} &U_{wM}(\beta) \\ &= \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \bar{x}_w(\beta, t)\right] d\left[N_i(t) - \int_0^t Y_i(u) \exp(\beta^T x_i) \frac{\sum_{j=1}^n w_j dN_j(u)}{\sum_{j=1}^n w_j Y_j(u) \exp(\beta^T x_j)}\right] \\ &= \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \bar{x}_w(\beta, t)\right] \left[dN_i(t) - \frac{\sum_{j=1}^n w_j Y_i(t) \exp(\beta^T x_i) dN_j(t)}{\sum_{j=1}^n w_j Y_j(t) \exp(\beta^T x_j)}\right] \end{aligned}$$

For notation let $U_i = \int_0^\infty w_i \left[x_i - \bar{x}_w(\beta, t)\right] dM_i(t)$. These U_i will be the summands of the estimating equations we will use for the Cox model.

Pugh et al. (1993) also derive the asymptotic distribution and variance for the solution to the weighted Cox estimating equations when weights are estimated instead of known. Suppose a model for the observation probabilities is given as $\pi(\alpha, x)$ and α is estimated via maximum likelihood estimation as the solution to $T(\alpha) = 0$,

$$T(\alpha) = \sum_{i=1}^{n} T_i = \sum_{i=1}^{n} \frac{\partial}{\partial \alpha} \left[R_i \log \pi(\alpha, x_i) + (1 - R_i) \log(1 - \pi(\alpha, x_i))\right]$$

The limiting distribution of $\hat{\beta}$ using the estimated probabilities, $\hat{\pi}_i = \pi(\hat{\alpha}, x_i)$, is

$$n^{1/2}(\hat{\beta}(\hat{\alpha}) - \beta^*) \xrightarrow{d} N(0, \Sigma^{-1} V \Sigma^{-1})$$

Where

$$\Sigma = E\left[\frac{-\partial}{\partial \beta^T} U_i(\alpha^*, \beta^*)\right]$$
$$V = E[U_i U_i^T] - E[U_i T_i^T] E[T_i T_i^T]^{-1} E[T_i U_i^T]$$

Pugh et al. (1993) show the variance can be estimated by estimating Σ as

$$\hat{\Sigma} = n^{-1} \sum_{i=1}^{n} \int_0^\infty \pi_i(\hat{\alpha})^{-1} R_i \left\{ \frac{S_w^{(2)}(\hat{\beta}(\hat{\alpha}), u)}{S_w^{(0)}(\hat{\beta}(\hat{\alpha}), u)} - \left[\frac{S_w^{(1)}(\hat{\beta}(\hat{\alpha}), u)}{S_w^{(0)}(\hat{\beta}(\hat{\alpha}), u)}\right]^{\otimes 2} \right\} dN_i(u)$$

where for $k = 0, 1, 2$,

$$S_w^{(k)}(\hat{\beta}(\hat{\alpha}), u) = n^{-1} \sum_{i=1}^{n} \pi_i(\hat{\alpha})^{-1} R_i x_i^{\otimes k} Y_i(u) \exp(\hat{\beta}(\hat{\alpha})^T x_i)$$

They suggest estimating V by the empirical covariance matrix of the residuals from a

regression of $\hat{U}_i$ on $\hat{T}_i$,

$$\hat{V} = n^{-1} \sum_{i=1}^{n} \left[\hat{U}_i - \left(n^{-1} \sum_{j=1}^{n} \hat{U}_j \hat{T}_j^T \right) \left(n^{-1} \sum_{j=1}^{n} \hat{T}_j \hat{T}_j^T \right)^{-1} \hat{T}_i \right]^{\otimes 2}$$

Where $\hat{U}_i$ and $\hat{T}_i$ are the sample estimates of U_i and T_i,

$$\hat{U}_i = \int_0^\infty \pi_i(\hat{\alpha})^{-1} R_i \left[X_i - \frac{S_w^{(1)}(\hat{\beta}(\hat{\alpha}), t)}{S_w^{(0)}(\hat{\beta}(\hat{\alpha}), t)} \right] d\hat{M}_i(t)$$

$$\hat{T}_i = \frac{\partial}{\partial \alpha} \left[R_i \log \pi_i(\alpha) + (1 - R_i) \log(1 - \pi_i(\alpha)) \right] |_{\alpha = \hat{\alpha}}$$

As mentioned above, to estimate $dM_i(u) = dN_i(u) - \lambda_0(u) Y_i(u) \exp(\beta_0^T x_i) du$ Pugh et al. (1993) use a modified Breslow estimate for the baseline hazard. They remark that "Although it is tempting to replace $dM_i(u)$ by $dN_i(u)$ in our estimate of U_i, this leads to correlated increments and invalidates the usual sums of squares estimates of covariance."

Although we have closely followed Pugh et al. (1993) above, Qi et al. (2005) also study the inverse probability weighting estimates for the Cox model. They use counting process notation and modern empirical process theory to prove the consistency and asymptotic normality of the estimator. The expressions they find for the asymptotic variance of $\hat{\beta}$ appear slightly different than those in Pugh et al. (1993). Further work will be necessary to determine any differences in the two variance estimates.

2.4.3 Misspecified Cox Models

Struthers & Kalbfleish (1986) show that even if the proportional hazards model is incorrectly specified, the estimator from the Cox partial likelihood is still consistent for some value. This value is defined as the solution to the estimating equations applied

to the true data generating process.

$$0 = E\left[\int_0^\infty X_i - \frac{S^{(1)}(\beta^*, t)}{S^{(0)}(\beta^*, t)} dN_i(t)\right]$$

Importantly, this type of misspecified model includes the situation when a covariate is omitted from the model. In the unified approach we will be deliberately fitting models with some covariates omitted. The conditions they give for the value β^* to exist are similar to conditions that should allow the unified approach to apply to the Cox model.

Chapter 3

Unified Estimate

3.1 Introduction

The Unified Approach (Zhao & Liu, 2021), or the Unified Estimate, is an extension of inverse probability weighting for a regression model which allows the use of information from individuals with partially observed data. The initial idea was proposed by Chen & Chen (2000) in the context of missing data coming from 2-phase sampling in survey sampling. Liu (2013) and Zhao & Liu (2021) extend the idea to data with multiple non-monotone patterns of missingness and apply the result to generalized linear models. Liu (2013) and Liu & Zhao (2022) extended the unified approach to longitudinal data using generalized estimating equations (GEE). Tang (2014) and Thiessen et al. (2022) extend the unified approach to survival data with the Cox proportional hazards model.

In the survey sampling literature the term "model assisted" refers to estimation procedures where an auxiliary or "working" model is also used to described the population distribution (Särndal et al., 1992). An appropriately chosen working model improves the primary estimator, but the basic statistical properties of the estimator

are not dependent on the working model being correctly specified. In this sense the unified estimate is a model assisted estimate of a regression model.

The core idea of the unified approach is to define a primary analysis model of interest and multiple working models. Each working model is a parametric or semi-parametric model which uses a different subset of variables in the data. Each working model is fit twice, once on the complete-case data and once on all available cases for that subset of covariates. This allows the model to use the additional information in the partially observed covariates to improve the estimate of the main analysis model parameters. When data is missing completely at random this procedure requires no other corrections, but when missingness is missing at random we use inverse probability weighting to adjust each of the analysis and working models.

The rest of this Chapter is laid out as follows. In Section 3.2 we introduce the notation for the analysis model, working models, and the models for the observation probabilities of the different subsets of the data. In Section 3.3 we define the unified approach. We discuss variance estimation in Section 3.4. We finish by examining some properties of the unified estimate in Section 3.5.

3.2 Notation

The development of the unified estimate involves some unusual terminology, so we first give an overview of the notation and terminology.

The data X is an n by P matrix, with entries $X_{ip}, i = 1, \ldots, n, p = 1, \ldots, P$, rows $X_i = (X_{i1}, X_{i2}, ..., X_{iP}), i = 1, \ldots, n$, and columns $X_p = (X_{1p}, \ldots, X_{np})^T, p = 1, \ldots, P$.

There is 1 "analysis model" and $J \geq 1$ "working models". The analysis model specifies the primary parametric or semi-parametric relation in the data we want to study. The working models each specify an alternative parametric or semi-parametric relation in

the data. Both the analysis and working models are specified via M-estimators.

The analysis model and each of the J working models all use a (not necessarily proper) subset of the variables of X. Throughout this work we will use a bracketed subscript to indicate that various terms are considered in reference to that subset. Denote the subset of variables used in the analysis model by $X_{(0)}$ and the subsets used in the working models by $X_{(j)}, j = 1, \ldots, J$. That is, $X_{(j)} \subseteq X, j = 0, \ldots, J$. The rows of $X_{(j)}$ are denoted by $X_{i(j)}, i = 1, \ldots, n, j = 0, \ldots, J$. We do not necessarily assume that the full-data analysis model uses all of the variables in X.

The desired full-data analysis model is specified through the population estimating equations $\psi(\beta, X)$.

$$0 = E[\psi(\beta, X)] = E[\psi(\beta, X_{(0)})] \tag{3.1}$$

The J working models are specified through the population estimating equations $\phi_j(\gamma_j, X)$,

$$0 = E[\phi_j(\gamma_j, X)] = E[\phi_j(\gamma_j X_{(j)})], j = 1, \ldots, J \tag{3.2}$$

The analysis model is assumed to be correctly specified. The J working models are not assumed to be correctly specified. However, we do assume that the data satisfies regularity conditions for M-estimators which guarantee a unique solution to the population estimating equations for the working models. Define β^* and γ_j^* to be the solutions to the population estimating equations. That is,

$$0 = E[\psi(X, \beta^*)]$$
$$0 = E[\phi_j(X, \gamma_j^*)], j = 1, \ldots, J$$

As there is missing data, define $R_{ip}, i = 1, \ldots, n, p = 1, \ldots, P$ as indicator variables for the ith individual which take the value 1 if variable X_{ip} is observed and 0 otherwise. Define $R_{i(j)}, i = 1, \ldots, n, j = 0, \ldots, J$ to be an indicator variable for the ith individual

taking the value 1 if all variables in $X_{(j)}$ are observed for that individual, and 0 otherwise. That is,

$$R_{i(j)} = \begin{cases} 1 & \forall X_p \in X_{(j)}, R_{ip} = 1 \\ 0 & \text{otherwise} \end{cases}, j = 0, \ldots, J$$

Individuals with $R_{i(0)} = 1$ are called the "analysis model complete cases", or simply the complete cases. They are indexed by the vector $R_{(0)} = (R_{1(0)}, \ldots, R_{n(0)})$. Individuals with $R_{i(j)} = 1, j = 1, \ldots, J$ are called the "jth working model available cases", or simply the available cases. They are indexed by the vector $R_{(j)} = (R_{1(j)}, \ldots, R_{n(j)}), j = 1, \ldots, J$.

We require that $R_{i(0)} \leq R_{i(j)}, i = 1, \ldots n, j = 1, \ldots, J$. That is, any individual who is a "complete case" for the analysis model must also be an "available case" for all of the working models. In addition, there must be at least one individual where $R_{i(0)} < R_{i(j)}$. That is, we require $\sum R_{i(0)} < \sum R_{i(j)}$.

If the data is MCAR then the above is sufficient. But in general we allow data to be MAR and use inverse probability weighting to adjust for non-response bias. To use IPW we need estimates of the probability the ith individual has observed data for each of the analysis and working models. Define $J + 1$ parametric models for these probabilities, $\pi_{i(j)} = P(R_{i(j)} = 1) = \pi_j(\alpha_j, X_i), j = 0, \ldots, J$ with parameters α_j. It is also possible to use other methods for estimating the observation probabilities, see discussion in Sections 2.3.1 and 3.5.1. Assume that $\pi_{i(j)} \geq \epsilon > 0, i = 1, \ldots, n, j = 0, \ldots, J$. Let $h_j(\alpha, X)$ be the estimating equations for each of these models. For notational simplicity we will not indicate which of the variables in X are used by the different observation probability models. But, assume that the variables used in the observation probability models are fully observed in the sample. We assume that these observation probability models are correctly specified. The true parameters $\alpha_j^*, j = 0, \ldots, J$ solve

the population estimating equations

$$0 = E[h_j(\alpha_j^*, X)], j = 0, \ldots, J$$

Let $\hat{\alpha}_j$ be the sample estimates of the jth observation probability model parameters from solving the sample estimating equations.

$$0 = \frac{1}{n}\sum_{i=1}^{n} h_j(\hat{\alpha}, x_i), j = 0, \ldots, J$$

From these estimates of the observation probability model parameters, create estimated observation probabilities for each of the $J + 1$ subsets.

$$\hat{\pi}_{i(j)} = \pi_j(\hat{\alpha}, x_i), j = 0, \ldots, J$$

Now define the IPW estimating equations for the analysis and working models. Often these IPW estimating equations are simply weighted versions of the full data estimating equations, although some models require additional adjustments. For example, in the Cox proportional hazards model the estimating equations contain an estimate of the average covariate vector of individuals who are at-risk at time t. In a weighted setting this average must also be replaced with a weighted average. For further details on the Cox model see Chapter 4. Denote the terms in the "analysis model IPW estimating equations" by $u_i = u(\beta, \alpha_0, X_{i(0)}, R_{i(0)})$,

$$0 = E\left[\frac{R_{i(0)}}{\pi_0(\alpha_0^*, X_i)}\psi(\beta^*, X_{i(0)})\right] = E[u(\beta^*, \alpha_0^*, X_{i(0)}, R_{i(0)})] \tag{3.3}$$

Each of the J working models will be fit two separate times, so there are $2J$ different sets of IPW estimating equations for the J working models. These equations differ only in the subset of individuals they are fit on and the corresponding observation probability

model. Denote the "complete case working model IPW estimating equations" by $f_j = f_j(\gamma_j, \alpha_0, X_{(j)}, R_{i(0)}), j = 1, \ldots, J$.

$$0 = E\left[\frac{R_{i(0)}}{\pi_0(\alpha_0^*, X_i)}\phi_j(\gamma_j^*, X_{i(j)})\right] = E[f_j(\gamma_j^*, \alpha_0^*, X_{i(j)}, R_{i(0)})], j = 1, \ldots, J \qquad (3.4)$$

Denote the "available case working model IPW estimating equations" by $g_j = g_j(\gamma_j', \alpha_j, x_{i(j)}, R_{i(j)}), j = 1, \ldots, J$,

$$0 = E\left[\frac{R_{i(j)}}{\pi_j(\alpha_j^*, X_i)}\phi_j(\gamma_j'^*, X_{i(j)})\right] = E[g_j(\gamma_j'^*, \alpha_j^*, X_{i(j)}, R_{i(j)})], j = 1, \ldots, J \qquad (3.5)$$

Note that to distinguish the parameters in the complete case and available case working models, we have decorated the parameters from the available case working models with a tick, γ_j'. If the observation probability models are correctly specified, then the solutions to both sets working model population estimating equations will be the same, $\gamma_j^* = \gamma_j'^*$.

Note that because $R_{i(0)} \leq R_{i(j)}$, all of the complete case and available case working models can be fit on the observed data. Furthermore, since $\sum R_{i(0)} < \sum R_{i(j)}$ and $R_{i(0)} \leq R_{i(j)}$, $\hat{\gamma}_j'$ will use strictly more information than $\hat{\gamma}_j$ and be more efficient.

We are now ready to define the unified estimate.

3.3 Definition of the Unified Estimate

So far we have defined the estimating equations and parameters for the analysis model, the two sets of J working models, and the $J + 1$ observation probability models. Let θ be the concatenation of all of the parameters defined so far, $\theta = (\beta, \gamma_1, \ldots, \gamma_J, \gamma_1', \ldots, \gamma_J', \alpha_0, \ldots, \alpha_J)$. We now stack all of the estimating equations

together vertically to get the unified estimating equations $S(\theta, X)$,

$$S_i = S(\theta, X_i) = \begin{pmatrix} u(\beta, \alpha_0, X_{i(0)}, R_{i(0)}) \\ f_1(\gamma_1, \alpha_0, X_{i(1)}, R_{i(0)}) \\ \cdots \\ f_J(\gamma_J, \alpha_0, X_{i(J)}, R_{i(0)}) \\ g_1(\gamma_1', \alpha_1, X_{i(1)}, R_{i(1)}) \\ \cdots \\ g_J(\gamma_J', \alpha_J, X_{i(J)}, R_{i(J)}) \\ h_0(\alpha_0, X_i) \\ h_1(\alpha_1, X_i) \\ \cdots \\ h_J(\alpha_J, X_i) \end{pmatrix}$$

Again for notational simplicity, let f, g, and h denote the vectors $(f_1, \ldots, f_J)$, $(g_1, \ldots, g_J)$, and $(h_0, \ldots, h_J)$. Also let γ, γ', and α denote $(\gamma_1, \ldots, \gamma_J)$, $(\gamma_1', \ldots, \gamma_J')$, and $(\alpha_0, \ldots, \alpha_J)$. Then we can express S_i as

$$S_i = \begin{pmatrix} u(\beta, \alpha) \\ f(\gamma, \alpha) \\ g(\gamma', \alpha) \\ h(\alpha) \end{pmatrix}$$

Let $\hat{\theta}$ be the solution to the sample estimating equations, $0 = \frac{1}{n} \sum S(\hat{\theta}, x_i)$. Although this looks imposing, each piece of the estimating equation can be solved separately and then combined. Under regularity conditions $\hat{\theta}$ should be consistent and have an asymptotic normal distribution. We discuss these properties in Section 3.5.2 below.

$$\sqrt{n}(\hat{\theta} - \theta^*) \sim N(0, V(\theta^*))$$

With

$$V(\theta^*) = A(\theta^*)^{-1}B(\theta^*)A(\theta^*)^{T-1}$$
$$A(\theta^*) = E[\frac{-\partial}{\partial\theta^T}S(\theta, X_i)|_{\theta^*}]$$
$$B(\theta^*) = E[S(\theta^*, X_i)S(\theta^*, X_i)^T]$$

Consider now a transformation of θ. We multiply θ by a constant permutation matrix P,

$$P = \begin{pmatrix} I_\beta & 0 & 0 & 0 \\ 0 & I_\gamma & -I_\gamma & 0 \\ 0 & 0 & 0 & I_\alpha \end{pmatrix}$$

Where I_β is a square identity matrix with the same dimension as β, I_γ is a square identity matrix with the same dimension as γ and γ', and I_α is a square identity matrix with the same dimension as α. This gives

$$P\theta = \begin{pmatrix} \beta \\ \gamma - \gamma' \\ \alpha \end{pmatrix}$$

If the observation probability models are correctly specified, then $\hat{\gamma}$ and $\hat{\gamma}'$ converge to the same value, $\hat{\gamma} - \hat{\gamma}' \to 0$. Therefore we get the asymptotic distribution of $P\hat{\theta}$ as

$$\sqrt{n}\left[\begin{pmatrix} \hat{\beta} \\ \hat{\gamma} - \hat{\gamma}' \\ \hat{\alpha} \end{pmatrix} - \begin{pmatrix} \beta^* \\ 0 \\ \alpha^* \end{pmatrix}\right] \sim N(0, \Sigma)$$

Where the variance Σ is a block matrix defined for notational simplicity.

$$\Sigma := V[\sqrt{n}P\hat{\theta}] = PV(\theta^*)P^T = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{pmatrix}$$

Finally, using properties of the normal distribution, the conditional distribution of $\hat{\beta}$ given $\hat{\gamma} - \hat{\gamma}' = a$ for an observed value a is

$$\sqrt{n}(\hat{\beta} - \beta^*)|(\hat{\gamma} - \hat{\gamma}' = a) \sim N\Big(0 + \Sigma_{12}\Sigma_{22}^{-1}(a - 0), \\ \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\Big)$$

So, conditional on the knowledge that $\hat{\gamma} - \hat{\gamma}' = a$, the estimate $\hat{\beta}$ is biased by a term $\Sigma_{12}\Sigma_{22}^{-1}a$. Subtracting this off leads to the unified estimate,

$$\bar{\beta} := \hat{\beta} - \Sigma_{12}\Sigma_{22}^{-1}(\hat{\gamma} - \hat{\gamma}')$$

With variance

$$V[\bar{\beta}] = (\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})/n$$

As the second term in the variance of the conditional distribution is positive semi-definite, $\bar{\beta}$ is at least as efficient as $\hat{\beta}$.

If any of the available case working models has fully observed data or has MCAR missing data, then we can omit the corresponding observation probability model and use the unweighted estimate for that available case working model.

3.4 Variance Estimation in the Unified Estimate

We now discuss estimating the variance of the unified estimate. The presentation of the variance is slightly different in this work than in previous work on the unified estimate. Here we focus on the variance-covariance matrix of the combined parameters θ or $P\theta$, whereas previous work focused on the variance of $\bar{\beta}$ directly. The results are the same, but we believe this presentation shows the underlying structure more clearly.

From the formula of the unified estimate we see that to calculate $\bar{\beta}$ itself requires estimates of Σ_{12} and Σ_{22}. Calculation of the variance of $\bar{\beta}$ also requires an estimate of Σ_{11}, where

$$\Sigma = V[\sqrt{n}P\hat{\theta}] = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{pmatrix}$$

is the variance-covariance matrix of the vector $\sqrt{n}P\theta = \sqrt{n}(\beta^T, (\gamma - \gamma')^T, \alpha^T)^T$. Rather than estimate Σ directly it is convenient to estimate the variance-covariance matrix of $\sqrt{n}\hat{\theta}$, $V[\sqrt{n}\hat{\theta}] = V(\theta^*)$, make the appropriate transformation to find $\hat{\Sigma}$, and then extract the desired components.

3.4.1 Sandwich Estimate

One method of estimating the variance of $\sqrt{n}\hat{\theta}$ is the sandwich estimate. Because $V[\sqrt{n}\hat{\theta}] = V(\theta^*) = A(\theta^*)^{-1}B(\theta^*)A(\theta^*)^{T-1}$, if we can find consistent estimates of A and B then we can use those to estimate $\hat{V}_{sand}(\theta^*) = \hat{A}^{-1}\hat{B}\hat{A}^{T-1}$. Then $\hat{\Sigma}_{sand} = P\hat{A}^{-1}\hat{B}\hat{A}^{T-1}P^T$

When S_i are independent the matrix $B(\theta^*)$ is simply the expected crossproduct of the S_i terms, and can be estimated by the sample crossproduct. $\hat{B} = n^{-1}\sum \hat{S}_i\hat{S}_i^T$. On the

other hand, the matrix $A(\theta^*)$ is defined as,

$$A(\theta^*) = E\left[\frac{-\partial S(\theta, X_i)}{\partial \theta^T}\right]$$
$$= \begin{pmatrix} E[\frac{-\partial u(\beta,\alpha)}{\partial \beta^T}] & 0 & 0 & E[\frac{-\partial u(\beta,\alpha)}{\partial \alpha^T}] \\ 0 & E[\frac{-\partial f(\gamma,\alpha)}{\partial \gamma^T}] & 0 & E[\frac{-\partial f(\gamma,\alpha)}{\partial \alpha^T}] \\ 0 & 0 & E[\frac{-\partial g(\gamma',\alpha)}{\partial \gamma'^T}] & E[\frac{-\partial g(\gamma',\alpha)}{\partial \alpha^T}] \\ 0 & 0 & 0 & E[\frac{-\partial h(\alpha)}{\partial \alpha^T}] \end{pmatrix}$$

The blocks on the main diagonal of $A(\theta^*)$ can be estimated by standard methods. Estimation of the terms on the upper part of the right-most column is more complicated. We present two proposals for estimation of these terms. Both of them are based on the idea of the "generalized information equality", $E[-\partial u/\partial \alpha] = E[uh^t]$ (Pierce, 1982) (Robins et al., 1995).

Note that in order to use the positive definite variance estimate of Robins et al. (1995) we must substitute $E[\frac{-\partial}{\partial \alpha^T} h(x, \alpha^*)] = E[h(x, \alpha^*)h(x, \alpha^*)^T]$ in addition to using one of the two options below.

3.4.1.1 Option 1

The first option is to replace the entire upper-right block of $A(\theta^*)$ together.

$$\begin{pmatrix} E[\frac{-\partial u(\beta,\alpha)}{\partial \alpha^T}] \\ E[\frac{-\partial f(\gamma,\alpha)}{\partial \alpha^T}] \\ E[\frac{-\partial g(\gamma',\alpha)}{\partial \alpha^T}] \end{pmatrix} = \begin{pmatrix} E[uh^T] \\ E[fh^T] \\ E[gh^T] \end{pmatrix}$$

Although simple to implement, this option does not take full advantage of the number of 0's known to exist in this matrix. Some of these terms should really be equal to 0, but are estimated as crossproducts instead.

3.4.1.2 Option 2

The second option takes full advantage of the knowledge that partial derivatives with respect to parameters not in the estimating equations are 0. We consider the analysis model and each working model separately with respect to each of the observation probability models. Then we have

$$\begin{pmatrix} E[\frac{-\partial u(\beta,\alpha)}{\partial \alpha^T}] \\ E[\frac{-\partial f(\gamma,\alpha)}{\partial \alpha^T}] \\ E[\frac{-\partial g(\gamma',\alpha)}{\partial \alpha^T}] \end{pmatrix} = \begin{pmatrix} E[uh_0^T] & 0 & 0 & \dots & 0 \\ E[f_1 h_0^T] & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E[f_J h_0^T] & 0 & 0 & \dots & 0 \\ 0 & E[g_1 h_1^T] & 0 & \dots & \vdots \\ 0 & 0 & E[g_2 h_2^T] & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & E[g_J h_J^T] \end{pmatrix}$$

Whether Option 1 or Option 2 are used, we can then use the sample estimates of the crossproducts in estimation of A. This allows us to avoid calculating complex derivatives of the estimating equations with respect to the weighting parameters.

3.4.2 Bootstrap Estimate

Another option to estimate the variance is to use the bootstrap (Efron, 1979). With the continuing increase in computational power the bootstrap is convenient to use. To use the bootstrap we resample the original dataset with replacement B times. On each of these bootstrapped datasets we perform all of the calculations in the unified approach, finding B different estimates of $\hat{\theta}$. Then the bootstrap estimate of variance is

$$\hat{V}_{boot}[\sqrt{n}\hat{\theta}] = \frac{n}{B-1}\sum_{b=1}^{B}(\hat{\theta}_b - \bar{\theta})(\hat{\theta}_b - \bar{\theta})^T$$

where $\bar{\theta} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b$. We then make the transformation $\hat{\Sigma}_{boot} = P\hat{V}_{boot}[\sqrt{n}\hat{\theta}]P^T$ and proceed as above.

3.5 Properties of Unified Estimate

We now discuss some of the properties of the unified estimate. In Section 3.5.1 we discuss estimation of probabilities for inverse probability weighting in the unified approach. In Section 3.5.2 we review the consistency and normality of the unified parameter estimates. In Section 3.5.3 we show that the unified estimate is optimal in the sense of having smallest variance in a certain class of estimates. In Section 3.5.4 we demonstrate the efficiency gains from the unified estimate in a simple example with data missing completely at random. In Section 3.5.5 we suggest some approaches for model diagnostics for the unified estimate. Some details of the proofs are deferred to Appendix A.

3.5.1 Estimation of Probabilities

Accurate estimation of the observation probabilities is absolutely essential for the unified estimate. First, because we require that the standard IPW estimate $\hat{\beta}$ is unbiased for β^*. Second, we require that for each working model, $\hat{\gamma} - \hat{\gamma}' \to 0$, which depends on both the complete case observation probability model and the available case observation probability model.

Various methods to estimate these observation probabilities are available. So far the unified approach has only been explored using logistic regression. The R code in Chapter 5 has the potential to include other estimators but it requires more work, see Section 5.3.

When discussing estimation of observation probabilities in inverse probability weighting Robins et al. (1994) write (pg 857):

> In our experience, moderate overparameterization of $\pi(\phi)$ produces significant finite sample bias in our estimated variance of $\hat{\alpha}$ but little bias in $\hat{\alpha}$ itself, suggesting that in this setting, inference (e.g., confidence intervals) should be based on bootstrap estimates of the variability of $\hat{\alpha}$ rather than on the variance estimator of proposition 6.1e.

This recommendation suggests that when we use a flexible estimate of the observation probabilities we should also use the bootstrap estimate of variance for the unified approach. Thiessen et al. (2022) used correctly specified logistic regression models for the observation probabilities and found the sandwich estimate of variance performed worse than expected. Although, that may be related to a problem with the estimating equations they used, see Section 4.2 for discussion on the estimating equations. At present it appears that using the bootstrap variance estimate is the most reliable method when inverse probability weighting is used, regardless of how the observation probabilities are estimated.

3.5.2 Consistency and Normality of Unified Estimate

Chen & Chen (2000) studied an unweighted version of the unified estimate with a single working model. They gave regularity conditions and showed that the vector $(\hat{\beta}, \hat{\gamma})$ is asymptotically jointly normally distributed. They also showed that $\hat{\beta}$, $\hat{\gamma}$, and $\hat{\gamma}'$ are all consistent. Zhao & Liu (2021) applied the same regularity conditions to the entire vector $(\hat{\beta}, \hat{\gamma}, \hat{\gamma}')$ to show joint normality and consistency of all the estimators. Zhao & Liu (2021) also referenced conditions from Robins et al. (1994) to conclude that when IPW weights are estimated by a correctly specified model, the weighted estimate is still consistent and asymptotically jointly normally distributed. From this Zhao and Liu also concluded that $\bar{\beta}$ is consistent and asymptotically normally distributed.

3.5.3 Optimality in a Class of Estimators

We recall a result from Chen & Chen (2000) and show it still holds for the multivariable unified estimate with non-monotone patterns. Consider all estimators of the form

$$\tilde{\beta} = \hat{\beta} - A(\hat{\gamma} - \hat{\gamma}')$$

Where either A or the limit of A as $n \to \infty$ is a fixed matrix. Under the assumptions that $\hat{\gamma}$ and $\hat{\gamma}'$ converge to the same value and that $\hat{\beta}$ is consistent for β^*, then $\tilde{\beta}$ is always an consistent estimate of β^*. Note that if A is a random matrix that is correlated with $(\hat{\gamma} - \hat{\gamma}')$ then $\tilde{\beta}$ may be biased in finite samples. We now show that taking $A = \Sigma_{12}\Sigma_{22}^{-1}$ leads to the smallest variance for $\tilde{\beta}$, where Σ_{12} is the covariance between $\hat{\beta}$ and $(\hat{\gamma} - \hat{\gamma}')$ and Σ_{22} is the variance of $(\hat{\gamma} - \hat{\gamma}')$.

Suppose instead we take $A = \Sigma_{12}\Sigma_{22}^{-1} + B$ for some non-zero matrix B. Then the estimator becomes $\tilde{\beta} = \hat{\beta} - (\Sigma_{12}\Sigma_{22}^{-1} + B)(\hat{\gamma} - \hat{\gamma}')$. As we show in the appendix, the variance of this estimator is

$$\begin{aligned} V[\tilde{\beta}] =& V[\hat{\beta} - (\Sigma_{12}\Sigma_{22}^{-1} + B)(\hat{\gamma} - \hat{\gamma}')] \\ =& \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} + B\Sigma_{22}B^T \end{aligned}$$

The first two terms, $\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$, are the variance of the unified estimate. The third term, $B\Sigma_{22}B^T$, is positive semi-definite. Therefore the variance of $\hat{\beta} - (\Sigma_{12}\Sigma_{22}^{-1} + B)(\hat{\gamma} - \hat{\gamma}')$ is greater than or equal to the variance of $\hat{\beta} - \Sigma_{12}\Sigma_{22}^{-1}(\hat{\gamma} - \hat{\gamma}')$, for any B not equal 0.

Therefore, when Σ is known, $\bar{\beta}$ is the most efficient estimator in the class of estimators $\hat{\beta} - A(\hat{\gamma} - \hat{\gamma}')$.

3.5.4 Efficiency of Unified Estimate

As mentioned in Section 3.1, the primary purpose of the unified approach is to reduce the variance of the analysis model estimate by using information from individuals with partially observed data. From Section 3.3 the variance of the unified estimate is

$$V[\bar{\beta}] = (\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})/n$$

The reduction in variance from the unified estimate is the second term in this expression. Therefore, we can reduce the variance of $\bar{\beta}$ either by increasing the covariance between $\hat{\beta}$ and $\hat{\gamma} - \hat{\gamma}'$ or by decreasing the variance of $\hat{\gamma} - \hat{\gamma}'$. From the former we find that the analysis model parameters should be strongly correlated with the working model parameters. From the latter we find that the working models should be estimated stably, and that the complete case working models and available case working models should not differ too much.

We now take a detailed look at a simple example of the unified estimate of a population average to see the efficiency gains that may occur in practice.

3.5.4.1 Example: Average of joint normal variables, one is MCAR

Suppose that X and Y are jointly normally distributed with a means μ_X and μ_Y, standard deviations $\sigma_X = \sigma_Y = 1$, and correlation $-1 < \rho < 1$.

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right]$$

Suppose X is fully observed and Y is MCAR with observation probability π. R is the observation indicator for Y. Then $R \sim Ber(p = \pi)$, and the observed data is n iid copies of (X, RY, R).

Consider the problem of estimating the population average of Y. We will use an analysis model with the sample average of the observed y values.

$$\bar{Y} = \hat{\beta} = \frac{1}{\sum_{i=1}^{n} R_i} \sum_{i=1}^{n} R_i Y_i$$

The analysis model estimating equations are

$$0 = \sum_{i=1}^{n} u_i(\hat{\beta}, x_{i(0)}) = \sum_{i=1}^{n} R_i(y_i - \hat{\beta})$$

The working models will use the average of the observed x values.

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

The complete case working model estimating equations are

$$0 = \sum_{i=1}^{n} f_i(\hat{\gamma}, x_{i(1)}) = \sum_{i=1}^{n} R_i(x_i - \hat{\gamma})$$

The available case working model estimating equations are

$$0 = \sum_{i=1}^{n} g_i(\hat{\gamma}', x_{i(1)}) = \sum_{i=1}^{n} (x_i - \hat{\gamma}')$$

Stacking the equations, we have that $\hat{\theta} = (\hat{\beta}, \hat{\gamma}, \hat{\gamma}')$ is the solution to

$$0 = \sum_{i=1}^{n} S_i(\hat{\theta}, X_i) = \begin{pmatrix} \sum_{i=1}^{n} R_i(y_i - \hat{\beta}) \\ \sum_{i=1}^{n} R_i(x_i - \hat{\gamma}) \\ \sum_{i=1}^{n} (x_i - \hat{\gamma}') \end{pmatrix}$$

The population estimating equations are

$$0 = E[S(\theta^*, X)] = \begin{pmatrix} \int R(y - \beta^*) dF(y) \\ \int R(x - \gamma^*) dF(x) \\ \int (x - \gamma'^*) dF(x) \end{pmatrix}$$

The expectation is taken with respect to the full distribution of X, Y, R. Because missingness is MCAR, R is independent of X, Y. Therefore we find

$$0 = E[S(\theta^*, X)] = \begin{pmatrix} \int R(y - \beta^*) dF(y) \\ \int R(x - \gamma^*) dF(x) \\ \int (x - \gamma'^*) dF(x) \end{pmatrix} = \begin{pmatrix} \pi \int (y - \beta^*) dF(y) \\ \pi \int (x - \gamma^*) dF(x) \\ \int (x - \gamma'^*) dF(x) \end{pmatrix}$$

The solution to the population estimating equations is (μ_Y, μ_X, μ_X), the population averages of Y and X. Denote the true values $\mu_Y = \beta^*, \mu_X = \gamma^* = \gamma'^*$. The variance of $\hat{\beta}$ requires us to find

$$B(\theta^*) = E[S(\theta^*, X) S(\theta^*, X)^T] = \begin{pmatrix} \pi & \pi\rho & \pi\rho \\ \pi\rho & \pi & \pi \\ \pi\rho & \pi & 1 \end{pmatrix}$$

See Appendix A for detailed calculations. We must also find

$$A(\theta^*) = E[\frac{-\partial}{\partial \theta} S(\theta^*, X)] = \begin{pmatrix} \pi & 0 & 0 \\ 0 & \pi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Then the variance of $\hat{\theta}$ is given by,

$$V[\hat{\theta}] = n^{-1}A^{-1}BA^{T-1} = n^{-1}\begin{pmatrix} 1/\pi & \rho/\pi & \rho \\ \rho/\pi & 1/\pi & 1 \\ \rho & 1 & 1 \end{pmatrix}$$

The permutation matrix in this example is given by

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{pmatrix}$$

So the variance matrix of $P\hat{\theta} = (\hat{\beta}, \hat{\gamma} - \hat{\gamma}')$ is

$$\Sigma := V[P\hat{\theta}] = PV[\hat{\theta}]P^T = n^{-1}\begin{pmatrix} 1/\pi & \rho(1-\pi)/\pi \\ \rho(1-\pi)/\pi & (1-\pi)/\pi \end{pmatrix}$$

The unified estimate then simplifies to

$$\begin{aligned} \bar{\beta} &= \hat{\beta} - \Sigma_{12}\Sigma_{22}^{-1}(\hat{\gamma} - \hat{\gamma}') \\ &= \hat{\beta} - \rho(\hat{\gamma} - \hat{\gamma}') \end{aligned}$$

And the asymptotic variance of $\bar{\beta}$ is

$$V[\bar{\beta}] = n^{-1}(1/\pi - \rho^2(1-\pi)/\pi)$$

Since the variance of $\hat{\beta} = \bar{y}$ is $V[\hat{\beta}] = \pi/n$, the asymptotic relative efficiency is

$$ARE(\bar{\beta}, \hat{\beta}) = V[\hat{\beta}]/V[\bar{\beta}] = 1/(1 - \rho^2(1-\pi))$$

Table 3.1 shows the relative efficiency of the unified estimate compared to the sample average of the Y values for various combinations of correlation, ρ, and probability of

Table 3.1: Relative Efficiency of Unified Estimate for Average

	rho = 0	rho = 0.2	rho = 0.4	rho = 0.6	rho = 0.8
pi = 0.8	1	1.008	1.033	1.078	1.147
pi = 0.6	1	1.016	1.068	1.168	1.344
pi = 0.4	1	1.025	1.106	1.276	1.623
pi = 0.2	1	1.033	1.147	1.404	2.049

Y being observed, π. Figure 3.1 shows a graph of the relative efficiency.

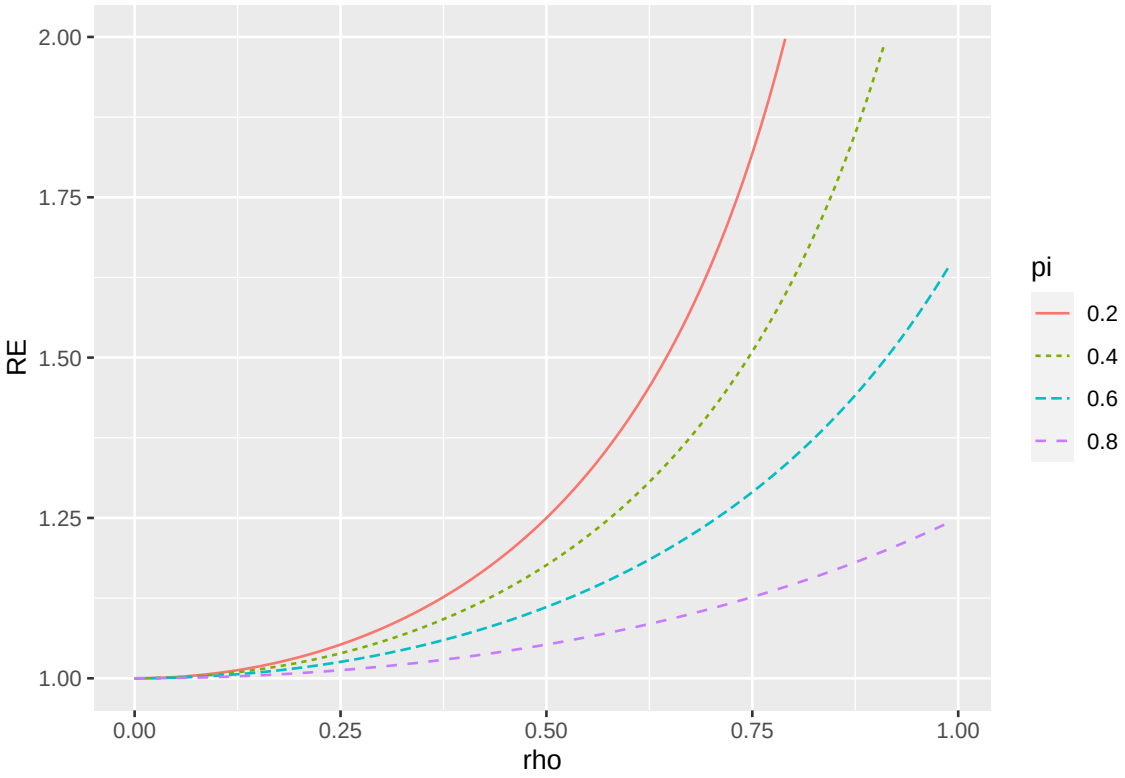

Figure 3.1: Relative Efficiency of Unified Estimate for Average

As we can see, the efficiency gains from the unified approach mostly depend on the strength of the correlation. For moderate situations with approximately 20% missing data and a correlation of 0.6, the unified estimate is roughly 8% more efficient than the naive complete case sample average. But for high correlations the unified approach is more efficient.

In general we will not know the true population correlation. Instead we will have to

estimate it, which will be less efficient and result in some small bias.

3.5.4.2 Simulation of Example

We now will simulate Example 1 from 3.5.4.1 using the `nmuer()` function defined in Chapter 5. For this simulation we set the correlation between X and Y to be 0.8 and the probability of observed data to be 0.8.

```
# Confirm the theoretical results of table 4.1
library(nmuer)
library(MASS)
nsim <- 100
nobs <- 500
cor <- 0.8
pi <- 0.8 # pi is observation probability
gen_dat <- function(nobs,cor,pi) {
  Sigma <- matrix(data = c(1,cor,cor,1),
                  nrow = 2, ncol = 2)
  dat <- mvrnorm(n = nobs, mu = c(0,0), Sigma = Sigma)
  dat[rbinom(nobs,1,pi) == 0,2] <- NA
  return(data.frame(x = dat[,1],
                    y = dat[,2]))
}
naive_var <- function(dat) {
  mod <- glm(y ~ 1, data = dat)
  return(vcov(mod))
}
nmuer_var <- function(dat) {
```

```
  mod <- nmuer(main_model = glm(y ~ 1),
               working_models = list(glm(x ~ 1)),
               data = dat)
  return(mod$beta_bar_var)
}


naive_vars <- rep(0, nsim)
nmuer_vars <- rep(0, nsim)
set.seed(123)
start_time <- proc.time()
for(i in 1:nsim) {
  dat <- gen_dat(nobs = nobs, cor = cor, pi = pi)
  naive_vars[[i]] <- naive_var(dat)
  nmuer_vars[[i]] <- nmuer_var(dat)
}
end_time <- proc.time()
duration <- end_time - start_time
duration
## user system elapsed
## 173.77 1.69 176.25
mean(naive_vars)/mean(nmuer_vars)
## [1] 1.154603
```

With 100 simulations, a sample size of 500, the unified variance estimated with 500 bootstrap samples, a correlation of 0.8, and 80% of the data observed, the simulated result gives a relative efficiency of 1.155. This is close to the calculated asymptotic relative efficiency of 1.147. The code takes about 3 minutes to run on my laptop.

3.5.5 Diagnosing Issues in the Unified Estiamte

The final result of the unified estimate is an estimator in the same class as the original analysis model, but with smaller variance and coefficients that have been slightly adjusted. Therefore it should be possible to apply some standard techniques of model checking to the final fit of the unified estimate. On the other hand, the final fit of the unified estimate is no longer minimizing the original objective function, so some things are different. For example, the sum of the residuals from a unified estimate of a linear regression is no longer 0.

Similarly, the unadjusted analysis model and each of the working models can be inspected individually to assess their fits. Although, recall that the working models are not required to be correctly specified. All that is required is that the parameter estimates from the complete case working models and available case working models converge to the same value.

3.5.5.1 Convergence of Working Models

The unified estimate requires that $\hat{\gamma}$ and $\hat{\gamma}'$ converge to the same value. If the data is MCAR then this will always happen, but if data is MAR and IPW is used to reweight the observations then this convergence depends on the correct specification of the observation probability models.

The most direct way to check convergence is to compare $(\hat{\gamma} - \hat{\gamma}')$ with its variance. Under the null hypothesis that $(\hat{\gamma} - \hat{\gamma}')$ is asymptotically normally distributed with mean 0 and variance $n^{-1}\Sigma_{22}$, we have

$$T_1 = n(\hat{\gamma} - \hat{\gamma}')^T \Sigma_{22}^{-1} (\hat{\gamma} - \hat{\gamma}') \sim \chi^2_{\sum p_j} \tag{3.6}$$

Where $\sum p_j$ is the dimension of γ. Large values of T_1 indicate that $\hat{\gamma}$ and $\hat{\gamma}'$ are

significantly different from each other.

Another way to evaluate this convergence is to check the change in the estimate of β. If $\hat{\beta}$ is significantly different than $\bar{\beta}$ we may suspect that $\hat{\gamma}$ and $\hat{\gamma}'$ are not close enough together to trust. To perform a formal hypothesis test we may either compare this difference to the variance of $\hat{\beta}$ or the variance of $\bar{\beta}$.

$\hat{\beta}$ and $\bar{\beta}$ are asymptotically normally distributed with variances $V[\hat{\beta}] = n^{-1}\Sigma_{11}$ and $V[\bar{\beta}] = n^{-1}(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$. The difference between $\hat{\beta}$ and $\bar{\beta}$ is $\Sigma_{12}\Sigma_{22}^{-1}(\hat{\gamma} - \hat{\gamma}')$. Therefore, we may consider the test statistics,

$$T_2 = n(\hat{\gamma} - \hat{\gamma}')^T \Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}(\hat{\gamma} - \hat{\gamma}') \tag{3.7}$$

$$T_3 = n(\hat{\gamma} - \hat{\gamma}')^T \Sigma_{22}^{-1}\Sigma_{21}\left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)^{-1}\Sigma_{12}\Sigma_{22}^{-1}(\hat{\gamma} - \hat{\gamma}') \tag{3.8}$$

However, T_2 and T_3 do not have central chi-square distributions. Instead they have "generalized chi-square" distributions. Mathai & Provost (1992) discuss properties of similar quadratic forms of random variables. Imhof (1961) provides an algorithm to approximate these distributions. If T_2 or T_3 are larger than the $100(1-\alpha)$th percentile of the distribution we would reject the hypothesis that $\hat{\gamma} - \hat{\gamma}'$ have mean zero and conclude the observation probability models are not adequate.

We may suspect that because the variance of $\bar{\beta}$ is smaller we should compare to that and use T_3. However, the generalized chi-square distributions for the two test statistics are different. Further study is needed to evaluate which test is more powerful.

In the hypothesis test (3.6) we are comparing the difference $(\hat{\gamma} - \hat{\gamma}')$ to its variance. In (3.7) we are comparing the change in $\hat{\beta}$ to its variance and in (3.8) we are comparing the change in $\bar{\beta}$ to its variance. Further work will be needed to determine how useful these hypothesis test are in practice.

Finally, if any of the above tests reject the conclusion that $\hat{\gamma}$ and $\hat{\gamma}'$ are equal, we may want to investigate the pairs of working models individually. By extracting the appropriate diagonal block from Σ_{22} we can perform hypothesis tests on $(\hat{\gamma}_j - \hat{\gamma}'_j)$ for each pair of working models we wish to check. This may give us information that a specific observation probability model is not performing well enough.

Chapter 4

Unified Estimate for Cox Proportional Hazards Model

In this chapter we review the changes to the unified estimate that Tang (2014) and Thiessen et al. (2022) made to adapt the method for the Cox proportional hazards model. We describe how two different sets of estimating equations are used in the A and B matrices. Importantly, in Section 4.2 we discuss a limitation of the sandwich estimate of variance used in Thiessen et al. (2022).

4.1 Estimating Equations to Use in Unified Estimate for Cox Model

As described in section 2.4 two different weighted estimating equations for the Cox Proportional Hazards model with inverse probability weights are:

$$U_{wM}(\beta) = \sum_{i=1}^{n} \int_0^\infty w_i \left[x_i - \bar{x}_w(\beta, t)\right] dM_i(t) \tag{4.1}$$

and

$$U_w(\beta) = \sum_{i=1}^{n} \int_0^{\infty} w_i \left[x_i - \bar{x}_w(\beta, t) \right] dN_i(t) \tag{4.2}$$

Where $\bar{x}_w$ is the weighted average covariate vector of individuals at risk at time t, weighted by both the relative risk and the inverse of the observation probabilities.

$$\bar{x}_w(\beta, t) = \frac{\sum_{j=1}^{n} w_j x_j Y_j(t) \exp(\beta^T x_j)}{\sum_{j=1}^{n} w_j Y_j(t) \exp(\beta^T x_j)}$$

Where $w_i = R_i \pi_i^{-1} = R_i \pi(\alpha, x_i)^{-1}$ is an indicator that the ith individual has completely observed data multiplied by the inverse of the probability the ith individual has completely observed data.

From section 2.4 we also have that $U_w(\beta) = U_{wM}(\beta)$. Therefore, the estimates of $\hat{\beta}$, $\hat{\gamma}$, and $\hat{\gamma}'$ can all use either version of the estimating equations. However, the estimating equations differ in estimates of the crossproduct $E[U_i U_i^T]$. This means the sandwich estimate of variance depends on which equations we use. The summands in (4.2) are simpler to work with than those in (4.1) because the former do not require estimation of the baseline hazard.

Lin & Wei (1989), Pugh et al. (1993), and Qi et al. (2005), all show that the correct estimating equations to use in the sandwich estimate of variance are the summands in (4.1). Therefore we use the summands in (4.1) in the estimation of the crossproducts $E[U_i U_i^T]$. However, for simplicity and because $U_w(\beta) = U_{wM}(\beta)$ we use the summands from (4.2) in estimation of the derivative matrix $E[-\partial u_i / \partial \beta^T]$. We also make the substitutions $E[-\partial u_i / \partial \alpha^T] = E[u_i h_i^T]$ and $E[-\partial h_i / \partial \alpha^T] = E[h_i h_i^T]$ as in Section 3.4.1. Therefore, we define the estimating equations for the analysis model, complete

case working models, and available case working models, to be

$$\hat{u}_i = u_i(\hat{\beta}, \hat{\alpha}_0) = \int_0^\infty \hat{w}_{i(0)} \left[x_{i(0)} - \frac{S^{(1)}_{(0)}(\hat{\beta}, t)}{S^{(0)}_{(0)}(\hat{\beta}, t)} \right] dM_i(t)$$

$$\hat{f}_{ij} = f_{ij}(\hat{\gamma}, \hat{\alpha}_0) = \int_0^\infty \hat{w}_{i(0)} \left[x_{i(j)} - \frac{S^{(1)}_{(j)}(\hat{\gamma}, t)}{S^{(0)}_{(j)}(\hat{\gamma}, t)} \right] dM_i(t), j = 1, \ldots, J$$

$$\hat{g}_{ij} = g_{ij}(\hat{\gamma}', \hat{\alpha}_j) = \int_0^\infty \hat{w}_{i(j)} \left[x_{i(j)} - \frac{\tilde{S}^{(1)}_{(j)}(\hat{\gamma}', t)}{\tilde{S}^{(0)}_{(j)}(\hat{\gamma}', t)} \right] dM_i(t), j = 1, \ldots, J$$

Where

$$\hat{w}_{i(j)} = R_{i(j)} / \pi_j(\hat{\alpha}_j, x_i), j = 0, \ldots, J$$

And for $k = 0, 1, 2$,

$$S^{(k)}_{(0)}(\beta, t) = \sum_{l=1}^{n} \hat{w}_{l(0)} x^{\otimes k}_{l(0)} Y_l(t) \exp(\beta^T x_{l(0)})$$

$$S^{(k)}_{(j)}(\gamma, t) = \sum_{l=1}^{n} \hat{w}_{l(0)} x^{\otimes k}_{l(j)} Y_l(t) \exp(\gamma^T x_{l(j)}), j = 1, \ldots, J$$

$$\tilde{S}^{(k)}_{(j)}(\gamma', t) = \sum_{l=1}^{n} \hat{w}_{l(j)} x^{\otimes k}_{l(j)} Y_l(t) \exp(\gamma'^T x_{l(j)}), j = 1, \ldots, J$$

The estimating equations for the observation probability models are unchanged from Section 3.2,

$$\hat{h}_{ij} = h_j(\hat{\alpha}, x_i), j = 0, \ldots, J$$

We also let $C_i, D_{ij}, E_{ij}, j = 0, \ldots, J$ be negative the partial derivatives of equation (4.2) corresponding to the analysis model, complete case working models, and available

case working models,

$$C_i = \int_0^\infty \frac{S_{(0)}^{(2)}(\beta,t)}{S_{(0)}^{(0)}(\beta,t)} - \frac{(S_{(0)}^{(1)}(\beta,t))^{\otimes 2}}{(S_{(0)}^{(0)}(\beta,t))^2} dN_i(t)$$
$$D_{ij} = \int_0^\infty \frac{S_{(j)}^{(2)}(\gamma,t)}{S_{(j)}^{(0)}(\gamma,t)} - \frac{(S_{(j)}^{(1)}(\gamma,t))^{\otimes 2}}{(S_{(j)}^{(0)}(\gamma,t))^2} dN_i(t), j = 1,\ldots,J$$
$$E_{ij} = \int_0^\infty \frac{\tilde{S}_{(j)}^{(2)}(\gamma',t)}{\tilde{S}_{(j)}^{(0)}(\gamma',t)} - \frac{(\tilde{S}_{(j)}^{(1)}(\gamma',t))^{\otimes 2}}{(\tilde{S}_{(j)}^{(0)}(\gamma',t))^2} dN_i(t), j = 1,\ldots,J$$

Using Option 2 from Section 3.4.1 to deal with the partial derivatives of the score equations with respect to the parameters of the observation probability models, we then estimate A by

$$\hat{A} = n^{-1} \begin{pmatrix} \sum \hat{C}_i & 0 & \ldots & 0 & \sum \hat{u}_i \hat{h}_{i0}^T & \ldots & 0 \\ 0 & \sum \hat{D}_{i1} & \ldots & 0 & \sum \hat{f}_{i1} \hat{h}_{i0}^T & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sum \hat{E}_{iJ} & 0 & \ldots & \sum \hat{g}_{iJ} \hat{h}_{iJ}^T \\ 0 & 0 & \ldots & 0 & \sum \hat{h}_{i0} \hat{h}_{i0}^T & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & \sum \hat{h}_{iJ} \hat{h}_{iJ}^T \end{pmatrix}$$

We estimate B as usual by the sample average of the crossproduct terms $S_i(\hat{\theta})$.

$$\hat{B} = n^{-1} \sum \hat{S}_i \hat{S}_i^T$$
$$= n^{-1} \begin{pmatrix} \sum \hat{u}_i \hat{u}_i{}^T & \sum \hat{u}_i \hat{f}_{i1}^T & \ldots & \sum \hat{u}_i \hat{g}_{iJ}^T & \sum \hat{u}_i \hat{h}_{i0}^T & \ldots & \sum \hat{u}_i \hat{h}_{iJ}^T \\ \sum \hat{f}_{i1} \hat{u}_i{}^T & \sum \hat{f}_{i1} \hat{f}_{i1}^T & \ldots & \sum \hat{f}_{i1} \hat{g}_{iJ}^T & \sum \hat{f}_{i1} \hat{h}_{i0}^T & \ldots & \sum \hat{f}_{i1} \hat{h}_{iJ}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \sum \hat{h}_{iJ} \hat{u}_i{}^T & \sum \hat{h}_{iJ} \hat{f}_{i1}^T & \ldots & \sum \hat{h}_{iJ} \hat{g}_{iJ}^T & \sum \hat{h}_{iJ} \hat{h}_{i0}^T & \ldots & \sum \hat{h}_{iJ} \hat{h}_{iJ}^T \end{pmatrix}$$

4.2 Inadequacy of the Sandwich Variance Estimate with Standard Estimating Equations

We return to the distinction between the Cox estimating equations based on the counting process $N_i(t)$ and on the mean-zero martingale $M_i(t)$. Thiessen et al. (2022) performed two simulation studies for the unified estimate of the Cox model. In those simulations they calculated the sandwich estimate of variance using estimating equations based on $N_i(t)$. That is, they estimated the A matrix of the sandwich variance by

$$\hat{A} = n^{-1} \begin{pmatrix} \sum -\frac{\partial}{\partial \beta^T}\psi_i & 0 & 0 & \sum \psi_i h_i^T \\ 0 & \sum -\frac{\partial}{\partial \gamma^T}\phi_i & 0 & \sum \phi_i h_i^T \\ 0 & 0 & \sum -\frac{\partial}{\partial \gamma'}\tilde{\phi}_i & \sum \tilde{\phi}_i h_i^T \\ 0 & 0 & 0 & \sum -\frac{\partial}{\partial \alpha^T} h_i \end{pmatrix}$$

Where h are the estimating equations for logistic models of the observation probabilities, ψ are inverse probability weighted Cox estimating equations for the analysis model of the form (4.2),

$$\psi_i(\beta, \alpha) = \int_0^\infty R_i \pi_i^{-1}(x_i - \bar{x}_w) Y_i(t) dN_i(t)$$

And ϕ and $\tilde{\phi}$ are similarly Cox estimating equations for the working models based on the counting process $dN_i(t)$. Similarly, the B matrix was estimated as the sample average of the crossproducts of the estimating equations based on the counting process.

$$\hat{B} = n^{-1} \begin{pmatrix} \sum \psi_i \psi_i^T & \sum \psi_i \phi_i^T & \sum \psi_i \tilde{\phi}_i^T & \sum \psi_i h_i^T \\ \sum \phi_i \psi_i^T & \sum \phi_i \phi_i^T & \sum \phi_i \tilde{\phi}_i^T & \sum \phi_i h_i^T \\ \sum \tilde{\phi}_i \psi_i^T & \sum \tilde{\phi}_i \phi_i^T & \sum \tilde{\phi}_i \tilde{\phi}_i^T & \sum \tilde{\phi}_i h_i^T \\ \sum h_i \psi_i^T & \sum h_i \phi_i^T & \sum h_i \tilde{\phi}_i^T & \sum h_i h_i^T \end{pmatrix}$$

In those simulation settings they found that the coverage rate of the 95% confidence intervals was generally less than 95%. They recommended the bootstrap be used for variance estimation as it gave a more reliable result and is computationally feasible for many datasets. An area for future work is to revisit these simulations using instead the martingale based estimating equations and see if they improve the performance of the sandwich estimate of variance.

Chapter 5

R Program for Unified Estimate

In this chapter we describe an R (R Core Team, 2022) function to fit any combination of GLM and Cox Proportional Hazards models through the unified estimate. The function to fit these estimates is named `nmuer()`, for "Non-Monotone Unified Estimate in R".

The R code is available in the appendix and also at https://github.com/DavidLukeThiessen/nmuer

The experimental version of the R package can be installed by running the command:

```
devtools::install_github(repo = "DavidLukeThiessen/nmuer")
```

The package is not yet on CRAN.

After installing the package from github it can be loaded with

```
library(nmuer)
```

5.1 R Meta-Programming

The R programming language supports meta-programming, meaning it is possible to write code which modifies itself as it is running. This is a powerful but complicated feature. For a detailed explanation about meta-programming in R see Whickham (2019) chapters 17-20.

The `nmuer()` function uses meta-programming to allow users to specify working models written as function calls, such as `glm()` or `coxph()`. This allows the estimation code for GLM or Cox models to be taken directly from the 'stats' and 'survival' packages. It also allows users to specify the analysis models in the same method as they normally would while programming in R. We decided this was a simpler interface than providing large lists of potential function arguments.

For example, if a user wants to run a unified estimate with the primary analysis model being a linear regression of y on x_1 and x_2 and one working model regressing y on x_1 only, they could run:

```
nmuer(
  main_model = glm(y ~ x1 + x2),
  working_models = list(
    glm(y ~ x1)
  ),
  data = data)
```

The first model, `glm(y ~ x1 + x2)`, specifies the analysis model and the second, `glm(y ~ x1)`, defines the working model.

`nmuer()` will then capture the main and working models before they are run, modify the code and run both models on appropriate subsets of the data, then combine and

return the final results. The 'rlang' package (Lione & Wickham, 2022) was used to provide additional programming infrastructure for this.

5.2 Function Arguments

In this section we describe the possible arguments used in the function. The overall structure of an `nmuer()` call looks like this:

```
nmuer(
  main_model,
  working_models,
  data,
  main_weight_model = NULL,
  working_weight_models = NULL,
  variance_estimate = c("bootstrap", "sandwich"),
  num_boots = 500,
  DEBUG = FALSE
)
```

Three of these arguments are required: `main_model`, `working_models` and `data`. The others are optional, although `main_weight_model` and `working_weight_model` should usually be used.

`main_model` should be either a single `glm()` or `coxph()` function call. This represents the analysis model in the unified estimate. Whether it is a `glm()` or a `coxph()` model it must include a "formula" specifying the model to be fitted. If it is a `glm()` function it is also possible to specify a "family" argument. This allows `nmuer()` to support any distributions which have already been included in `glm()`. It is not possible to specify "weights", "subset", or "data" arguments inside the `glm()` or `coxph()` functions, as

those will be overwritten when `nmuer()` modifies the code. `nmuer()` automatically calculates which individuals have observed data for each of the analysis and working models and uses those individuals as appropriate. Other standard function arguments to `glm()` or `coxph()` such as the "ties" method in `coxph()` have not been verified, but should in principle work.

`working_models` should be a `list()` of `glm()` or `coxph()` functions calls similar to `main_model`. Each element of the list defines one working model. The `nmuer()` code automatically fits one copy of each working model on the complete cases and one copy on the available cases, calculated from that working model's formula. It is not required that the variables used in `working_models` are a subset of the variables used in `main_model`. But it is required that $R_{(j)} \geq R_{(0)}$ for all working models.

`data` should be a data frame containing all the variables in the analysis model and working models. Any missing values should be coded as the standard `NA` value.

`main_weight_model` and `working_weight_models` are special function arguments created for the `nmuer()` function. We discuss them below in section 5.3.

`variance_estimate` defines whether the `nmuer()` code should estimate the variance using the bootstrap or a sandwich estimate. Currently only bootstrapping is supported, but in the future support for the sandwich estimate of variance may be added.

`num_boots` sets the number of bootstrap replicates to be used when estimating the variance through the bootstrap. By default 500 replicates are used.

`DEBUG` is a flag that makes the function run step by step and print extra output. We use it to ensure the function is performing correctly at each point. Users should not need to use it and we plan to remove this option when we submit the package to CRAN.

5.3 Estimation of Data Observation Probabilities

The estimation of observation probabilities is incredibly important for the unified estimate to be valid. Therefore it would be good if it were possible to estimate these probabilities in different methods, so that a user could pick the most appropriate method.

So far we have programmed the ability to use logistic regression to estimate the observation probabilities. But because that method is quite limited we also allow users to write their own functions to estimate these probabilities. In this way a user who is familiar with another method could estimate observation probabilities with that method instead. For example, someone could estimate the observation probabilities using a random forest. This option has slightly increased the complexity of the `nmuer()` code, but we felt that giving users this ability was very important.

There are two function arguments to specify the observation probabilities: `main_weight_model` and `working_weight_models`. Both work on the same principle. The user specifies a method to use for estimating the observation probabilities and any additional function arguments required for that method to work. Then the `nmuer()` code uses that method on the appropriate subset of the data to estimate the observation probabilities. Finally `nmuer()` uses those probabilities in estimating the analysis and working models.

Currently two options are built-in to the `nmuer` package. The first and simplest is to assume the data is MCAR. This can be specified either by leaving the weight_model arguments empty or by entering `nmuer_weights_constant()`. This does not require any further input from the user.

The second current option is to use a logistic regression model to estimate the observation probabilities. This can be done by entering `nmuer_weights_logistic()`.

This also requires the user to specify which variables to use in the logistic regression. This can be done by supplying the right-hand side of a `formula` object. For example, `missing_formula = ~ X1 + X2 + X1:X2`.

`main_weight_model` should be supplied as a single function, with a formula if needed. `working_weight-models` should be a list of functions. If at least one working weight model is specified then the user must supply one working weight model for each working model. If any of the working models have completely observed data or MCAR missingness then `nmuer_weights_constant()` should be used for that model. For example, if we want to specify that the main observation probability model should be a logistic model based on an auxiliary variable Z_1, the first working observation probability model is a logistic model based on Z_2, and the second working model has either MCAR or completely observed data, we would specify something like:

```
mod <- nmuer(
  main_model = glm(y ~ x1 + x2),
  working_models = list(
    glm(y ~ x1),
    glm(y ~ x2)),
  main_weight_model = nmuer_weights_logistic(missing_formula = ~ z1),
  working_weight_models = list(
    nmuer_weights_logistic(missing_formula = ~ z2),
    nmuer_weights_constant()),
  data = dat)
```

Note that any variables used in the observation probability models must be fully observed.

5.4 Issues with Weights in R

There are a couple issues using R's standard `weight` arguments for non-response adjustment.

The first problem is that in R some modelling functions treat a `weight` argument as supplying either case weights or precision weights. On the other hand, weighting to adjust for missing data is more similar to sampling weights. This means that using the `weight` argument to supply non-response adjustment weights will lead to incorrect estimates of variance from many functions. For example, if we fit 2 logistic models in R using the `glm()` function with `family = binomial` where the first model has weights all equal to 1 and the second has weights all equal to 10, then the estimated variance of the second model will be 1/10th the estimated variance of the first model.

This means that naively using the estimated variances can be incorrect. However, the estimated parameters will still be correct. Some functions and packages will also output variances appropriate for sampling weights. For example, the `svyglm` function from the `survey` package outputs appropriate variances for sampling weights.

Because we use bootstrapping to estimate the variances of the coefficient estimates, the differences between case weights, precision weights, and sampling weights does not cause us problems.

The second problem we encounter is that a logistic regression using non-integer weights will give a warning that the final outcome, $W \times Y$, is not an integer even if Y is an integer. Despite this warning R will still perform the calculations and give the "correct" parameter estimates. These warnings can be avoided by using the quasi-binomial family or suppressing warnings in R. Using the quasi-binomial family leads to a different estimate of the variance compared to the binomial family, but since we use bootstrapping to estimate the variance that is currently not a concern.

5.5 Planned Extension for Coxph()

A major feature of `coxph()` models is to allow start-stop type datasets, where a single individual can have multiple rows in the data and can have time-varying covariates. It would be valuable to maintain this feature, but we have not explored it yet and do not know if it can work with observation probabilities estimated from the same dataframe.

5.6 Example, Unified Logistic Regression with MCAR Covariates

Suppose that the user wants to estimate a logistic model, $logit(P(Y = 1)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$. Suppose that the response Y is fully observed, but both X_1 and X_2 can have missing values. For simplicity assume that missingness is MCAR. In order to use all of the available data, we could specify the analysis model and 3 working models.

$$
\begin{aligned}
logit(P(Y = 1)) &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 \\
logit(P(Y = 1)) &= \gamma_{10} + \gamma_{11} X_1 \\
logit(P(Y = 1)) &= \gamma_{20} + \gamma_{21} X_2 \\
logit(P(Y = 1)) &= \gamma_{30}
\end{aligned}
$$

Individuals who have (Y, X_1, X_2) observed will have $R_0 = 1$, individuals with (Y, X_1) observed will have $R_1 = 1$, individuals with (Y, X_2) observed will have $R_2 = 1$, and all individuals with have $R_3 = 1$ because Y is fully observed.

A simulated dataset for this may be generated as

```r
n <- 500
set.seed(12345)
sample_data <- data.frame(X1 = rnorm(n),
                          X2 = rnorm(n))
sample_data$Y <-
  rbinom(n, 1, prob =
           1/(1 + exp(-(0.5 + (0.2*sample_data$X1) +
                          (0.3*sample_data$X2)))))
sample_data[which(rbinom(n, 1, 0.2) == 1),"X1"] <- NA
sample_data[which(rbinom(n, 1, 0.2) == 1),"X2"] <- NA
```

The average proportion of $Y = 1$ is about 60%.

The unified estimate may be fit as:

```r
mod <- nmuer(
  main_model = glm(Y ~ X1 + X2, family = binomial(link = "logit")),
  working_models = list(
    glm(Y ~ X1, family = binomial(link = "logit")),
    glm(Y ~ X2, family = binomial(link = "logit")),
    glm(Y ~ 1, family = binomial(link = "logit"))
    ),
  data = sample_data)
```

Finally, we can print a summary of the unified estimate:

```r
summary(mod)
# A non-monotone unified estimate with a glm analysis model and 3
↪  working models.
```

```
#
# Call:  nmuer(main_model = glm(Y ~ X1 + X2, family = binomial(link
↪  = "logit")),
#     working_models = list(glm(Y ~ X1, family = binomial(link =
↪  "logit")),
#         glm(Y ~ X2, family = binomial(link = "logit")), glm(Y ~
#             1, family = binomial(link = "logit"))), data =
↪  sample_data)
#
# coefficients:
#                 coef se(coef)      z         p
# (Intercept) 0.499943 0.093577 5.3426 9.162e-08 ***
# X1          0.151794 0.107056 1.4179  0.156219
# X2          0.333383 0.102051 3.2668  0.001088 **
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The original unadjusted output is also saved, so we can compare the unified estimate and standard error with the unadjusted estimates.

```
print(mod$unadjusted_betahat)
# (Intercept)          X1          X2
#   0.5082220   0.1903121   0.3554956
print(sqrt(diag(mod$unadjusted_betahat_var)))
# 0.1133281 0.1222092 0.1181341
```

We can see that although the parameter estimates are not much improved, the

estimated standard errors are notably smaller.

5.7 Example, Unified Logistic Regression with MAR Covariate

This example is similar to 5.6 except we change the probabilities of being missing to depend on Y. $P(R_j = 1|Y = 1) = 0.3$ $P(R_j = 1|Y = 0) = 0.1$, $j = 1, 2$.

```
n <- 500
set.seed(12345)
sample_data <- data.frame(X1 = rnorm(n),
                          X2 = rnorm(n))
sample_data$Y <-
  rbinom(n, 1, prob =
           1/(1 + exp(-(0.5 + (0.2*sample_data$X1) +
                        (0.3*sample_data$X2)))))
sample_data[which(rbinom(
  n, 1,
  prob = ifelse(sample_data$Y == 1, 0.3, 0.1)) == 1),"X1"] <- NA
sample_data[which(rbinom(
  n, 1,
  prob = ifelse(sample_data$Y == 1, 0.3, 0.1)) == 1),"X2"] <- NA
```

To estimate the probability individuals have observed data we specified the `nmuer_weights_logistic()` option to tell `nmuer()` to fit logistic regression models. We use the same logistic regression formula for both the analysis and working models, which is not generally correct but matches common practice. For the final working model which has fully observed data we instead specify `nmuer_weights_constant()`.

Note that there are many warning messages because of non-integer weights as discussed in 5.4, but we have stopped the warnings from being displayed.

```
mod <- nmuer(
  main_model = glm(Y ~ X1 + X2, family = binomial(link = "logit")),
  working_models = list(
    glm(Y ~ X1, family = binomial(link = "logit")),
    glm(Y ~ X2, family = binomial(link = "logit")),
    glm(Y ~ 1, family = binomial(link = "logit"))
    ),
  data = sample_data,
  main_weight_model = nmuer_weights_logistic(missing_formula = Resp ~
  ↪  Y),
  working_weight_models = list(
    nmuer_weights_logistic(missing_formula = Resp ~ Y),
    nmuer_weights_logistic(missing_formula = Resp ~ Y),
    nmuer_weights_constant()))
summary(mod)
# A non-monotone unified estimate with a glm analysis model and 3
↪  working models.
#
# Call:  nmuer(main_model = glm(Y ~ X1 + X2, family = binomial(link
↪  = "logit")),
#     working_models = list(glm(Y ~ X1, family = binomial(link =
↪  "logit")),
#         glm(Y ~ X2, family = binomial(link = "logit")), glm(Y ~
#             1, family = binomial(link = "logit"))), data =
↪  sample_data,
```

```
#     main_weight_model = nmuer_weights_logistic(missing_formula =
↪  Resp ~
#         Y), working_weight_models =
↪  list(nmuer_weights_logistic(missing_formula = Resp ~
#         Y), nmuer_weights_logistic(missing_formula = Resp ~ Y),
#         nmuer_weights_constant()))
#
# coefficients:
#                 coef se(coef)      z         p
# (Intercept) 0.487735 0.092842 5.2534 1.493e-07 ***
# X1          0.115519 0.100412 1.1505 0.2499582
# X2          0.367847 0.098041 3.7520 0.0001754 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The corresponding unadjusted parameter estimates and standard errors are:

```
print(mod$unadjusted_betahat)
# (Intercept)          X1          X2
#   0.4908252   0.1248317   0.3628034
print(sqrt(diag(mod$unadjusted_betahat_var)))
# [1] 0.09316425 0.11987253 0.12356919
```

As before the parameter estimates are not much improved, but the standard errors are smaller. The standard errors for $\hat{\beta}_1$ and $\hat{\beta}_2$ in particular are smaller.

Chapter 6

Data Analysis

In this chapter we recall results from Thiessen et al. (2022) which show the application of the unified estimate for Cox proportional hazards developed in Chapter 4. We first examine a simulation study and then look at a real data analysis based on the MA.5 dataset. Both example analyses show that the proper unified estimate is nearly as efficient as the state-of-the-art multiple imputation method. Meanwhile the unified estimate can be specified without modeling the observation probabilities and it still is able to reduce the bias from non-response. This makes it an attractive choice for analysts who are concerned about missing data but don't have the ability to use complex multiple imputation methods.

6.1 Simulated Data for Proportional Hazards Model

In Thiessen et al. (2022) we performed a simulation study to evaluate the performance of the unified estimate for Cox proportional hazards regression. The covariates in this simulation were normally distributed, the survival time was exponentially distributed, and censoring time was uniformly distributed with

$$Z_1 \sim N(\mu = 0, \sigma = 0.5)$$

$$Z_2 \sim N(\mu = 3 + 0.5Z_1, \sigma = 0.5)$$

$$Z_3 \sim N(\mu = 0.5Z_1 + Z_2, \sigma = 0.5)$$

$$Y \sim Exp(\lambda = \exp(\beta_0 + \beta_1 Z_1 + \beta_2 Z_2 + \beta_3 Z_3))$$

$$C \sim U(min = 0, max = 85)$$

Z_1 was fully observed while Z_2 and Z_3 were missing in a non-monotone pattern. Missingness followed a logistic model depending on the observed failure or censoring time T, Z_1, and missingness for Z_3 depends on whether Z_2 is missing. We considered two simulation settings. In setting A the missingness was independent of the observed time given Z_1 while in setting B the missingness also depended on the observed time. In setting A there were 28% of Z_2 and 35% of Z_3 missing while in setting B there were 12% of Z_2 and 17% of Z_3 missing.

The simulation compared seven different estimators:

- Estimation based on data with no missing values (full)
- Complete Case (CC)
- Weighted Complete Case (WCC)
- SMCFCS Multiple Imputation (MI)
- Unified Estimate with estimated observation probabilities and sandwich estimate of variance (UE)
- Unified Estimate with estimated observation probabilities and bootstrapped variance estimates (UE*)
- Unified Estimate with no observation weights and sandwich estimate of variance (UE^{CC})
- Unified Estimate with no observation weights and bootstrapped variance estimates (UE^{CC*})

The sample size was set at $N = 500$ and $N = 1000$, 1000 simulations were performed for each study. For multiple imputation we used a maximum of 1000 rejection sampling attempts and 5 imputed data sets. For the bootstrapped estimates of variance we used 200 bootstrap samples.

Table 6.1 reports the bias, the empirical standard deviation, the square root of the mean square error, the percent of 95% confidence intervals that contain the true parameter, and the mean of the estimated standard errors for each parameter.

Based on these results we see that the weighted complete case estimate, the multiple imputation estimate, and the unified estimate with observation weights are all able to remove the bias in Setting B. Importantly, even though the unified estimate without observation weights does not completely remove the bias it still reduces the bias. This suggests that the unified estimate may be recommended for general use. Even if researchers do not want to go through the difficult task of estimating the observation probabilities the unified estimate can still remove some of the non-response bias. On the other hand, when the observation probabilities can be modeled successfully the unified estimate is able to match the performance of the SMC-FCS multiple imputation.

6.2 Analysis of MA.5 Data with Proportional Hazards Model

Thiessen et al. (2022) also analysed the MA.5 data introduced in Section 1.2. There we compared the unified estimate with the complete case estimate and the SMCFCS Multiple Imputation estimate (Bartlett et al., 2015). As previously described there were 716 patients and 169 of them were missing the baseline quality of life measurements.

Table 6.2 is reproduced from Thiessen et al. (2022) and shows the estimates and

Table 6.1: Simulation Results

N		$\lvert Bias\rvert$			$s.d.$			RMSE			$95\%CP$			$s.e.$		
		β_1	β_2	β_3	β_1	β_2	β_3	β_1	β_2	β_3	β_1	β_2	β_3	β_1	β_2	β_3
(A) The missingness is independent of the failure time																
500	full	2	5	6	156	192	130	156	192	131	96.5	94.8	94.4	163	186	130
	CC	11	15	14	245	278	191	245	279	191	94.2	94.8	94.5	242	274	191
	WCC	7	15	14	252	280	194	252	280	194	93.2	93.7	94.0	244	275	192
	MI	2	15	13	168	264	185	168	264	186	96.7	93.5	93.0	175	251	180
	UE	16	13	13	174	256	179	175	256	180	95.1	93.0	93.7	174	248	177
	UE*	10	19	15	176	258	182	176	259	182	95.0	94.0	93.8	181	257	184
	UE^{cc}	11	13	13	173	255	177	173	255	178	95.3	93.6	94.1	173	246	176
	UE^{cc*}	12	18	15	174	258	180	174	259	180	96.1	93.3	94.7	181	255	183
1000	full	1	5	4	113	132	92	113	132	92	95.1	95.6	94.9	114	131	92
	CC	6	10	5	159	195	130	159	195	130	95.7	94.4	95.2	168	192	134
	WCC	6	10	4	165	196	132	165	196	132	95.4	93.8	95.1	170	192	134
	MI	4	11	6	121	181	127	121	181	128	95.0	93.6	94.0	122	178	127
	UE	13	10	6	125	182	126	125	182	126	93.5	93.5	94.3	122	174	124
	UE*	9	12	7	126	183	126	126	183	126	94.4	94.5	94.8	125	176	126
	UE^{cc}	10	11	7	125	180	125	125	181	125	92.9	93.3	94.4	122	173	124
	UE^{cc*}	11	13	7	127	180	125	127	181	125	94.2	94.2	94.8	125	175	125
(B) The missingness depends on the failure time																
500	full	2	5	6	156	192	130	156	192	131	96.5	94.8	94.4	163	186	130
	CC	136	65	38	195	233	158	237	242	163	89.9	92.8	93.5	197	225	157
	WCC	7	10	10	187	227	154	187	227	154	96.0	93.4	94.6	194	219	153
	MI	3	9	8	157	215	147	157	215	147	96.6	94.0	94.8	166	210	148
	UE	14	10	9	161	216	149	162	216	149	96.6	94.4	94.8	167	215	153
	UE*	4	8	9	163	219	150	163	219	151	95.9	94.0	94.2	167	211	150
	UE^{cc}	27	50	36	164	223	153	166	228	157	95.9	93.5	93.3	166	214	151
	UE^{cc*}	18	51	37	167	227	156	168	233	160	95.6	93.3	93.6	170	218	154
1000	full	1	5	4	113	132	92	113	132	92	95.1	95.6	94.9	114	131	92
	CC	136	59	30	136	160	111	192	170	115	83.5	94.1	95.1	138	158	110
	WCC	8	5	2	132	155	107	132	155	107	95.9	94.8	95.3	136	154	107
	MI	2	6	5	114	149	105	114	149	105	95.7	95.0	95.1	116	147	104
	UE	7	5	3	116	149	105	116	149	105	95.0	95.9	95.5	117	151	108
	UE*	3	3	2	117	150	105	117	150	105	95.0	95.3	94.5	116	147	104
	UE^{cc}	24	44	30	118	153	108	121	160	112	94.4	94.0	94.2	117	150	107
	UE^{cc*}	15	47	33	118	156	110	119	163	115	95.2	93.1	93.1	118	151	106

Entries of absolute value of bias ($|bias|$), empirical standard deviation ($s.d.$), square root of MSE (RMSE) and standard error ($s.e.$) are multiplied by 1000, and coverage rates of 95% confidence interval ($95\%CP$) are multiplied by 100.

Table 6.2: Results from Analysis of Clinical Trial on Early Breast Cancer

	CC		MI		UE^{cc*}	
	Est.	*s.e.*	*Est.*	*s.e.*	*Est.*	*s.e.*
Total BCQ Score (continuous)	0.205	0.108	0.192	0.104	0.196	0.113
Age (continuous in years)	-0.033	0.010	-0.037	0.009	-0.036	0.010
Size of Tumor: >2 to ≤5 cm vs. ≤2 cm	0.462	0.142	0.363	0.126	0.229	0.106
Size of Tumor: >5 cm vs. ≤2 cm	0.764	0.213	0.647	0.194	0.582	0.186
Number of Lymph Nodes with Cancer: 4-10 vs. 1-3	0.560	0.129	0.482	0.115	0.504	0.117
Number of Lymph Nodes with Cancer: 11+ vs. 1-3	1.183	0.203	1.184	0.180	1.152	0.190
Type of surgery: Total vs. Partial Mastectomy	0.012	0.130	0.163	0.116	0.171	0.117
Type of Chemotherapy: CMF vs. CEF	0.205	0.120	0.259	0.106	0.225	0.104

standard errors for the three estimates. Most of the estimates agree quite well, with both the MI and UE estimates having lower standard errors. In particular, the estimate of the effect of CMF vs CEF chemotherapy is statistically significant in both the MI and UE estimates. The difference is not statistically significant in the complete case estimate.

An additional difference worth noting is that in both the MI and UE estimate the effect of Type of Mastectomy is much larger than the complete case analysis (0.163 and 0.171 vs 0.012, respectively). The effect is still not statistically significant, but both specialized methods for analysing missing data give a result that is more than one standard error away from the complete case estimates.

Chapter 7

Conclusion and Future Work

In this thesis we have reviewed the theory of the unified estimate, gave comments on its implementation and use in practice, and presented an R package which can be used to fit the unified estimate. This supports the work of applied researchers by making tools and advice available to them when they deal with missing data in their work. In the rest of this chapter we give several directions for future research. We briefly recall four issues that were presented in the previous chapters and describe two new areas.

7.1 Estimation of Observation Probabilities

Recall from Section 3.5.1 that the correct specification of the models for the observation probabilities is essential for the unified estimate. Up to now only logistic regression models have been used in to estimate these probabilities for the unified approach. We described in Section 5.3 that the R code for the unified estimate supports extensions to other methods of estimating probabilities, but this feature has not been tested or explored. Researching different ways of estimating these probabilities, how they affect the unified estimate, and implementing some of them in the R package is a very important area for future work.

7.2 Diagnosis of Model Fit for the Unified Estimator

In Section 3.5.5 we discussed methods to evaluate the fit of the unified estimate and proposed two hypothesis tests for the convergence of $\hat{\gamma}$ and $\hat{\gamma}'$. It will be important to evaluate how useful these methods are in practice. Extending the R package with the ability to perform these hypothesis tests would also be very useful.

Additional methods to evaluate the fit of the final estimate would also be very useful. Being able to use the coefficient of determination, adjusted coefficient of determination, AIC, and BIC would allow researchers to compare the results of the unified estimate with other models. It should be possible to extend all of these methods by calculating the measures using the adjusted coefficient estimates from the unified approach. In general we expect the fit to the complete cases to be worse using the unified approach than using the complete case estimates, but it would be interesting to quantify how different they are.

Additional diagnostic checks for influential observations and outliers could also be considered. Deletion statistics such as dfbetas or Cook's distance can be calculated. Because the unified approach is able to use incomplete observations, these measures will be able to be used for all individuals in the data, not just the complete cases. The ability to assess whether a partially observed individual is an outlier may inform decisions about

7.3 Cox Estimating Equations Based on Martingales

As described in Section 4.2, the simulation settings in Thiessen et al. (2022) used score equations based on the counting process $dN_i(t)$, but Lin & Wei (1989), Pugh et al. (1993), Qi et al. (2005), and Tang (2014) recommend using score equations based on the martingale transformation $dM_i(t)$. Using these score equations may improve the performance of the sandwich estimate of variance. For large datasets the bootstrap estimate of variance can become impractical. Therefore improving the performance of the sandwich estimate would be worthwhile.

7.4 Development and Release of an R Package for the Unified Estimate

Although the R code from Chapter 5 and Appendix B is available on the author's github page, it has not thoroughly been tested and is not available on the CRAN network. Further software development and the release of the code on CRAN would be significant milestones in making the unified estimate widely available to applied researchers.

7.5 Regression Estimate of Population Average

We now comment on a relationship between the unified estimate and a regression estimator. The unified estimate for non-monotone missing data is conceptually similar to the regression estimator of a population average from two-stage sampling.

Recall from the survey sampling literature the "Regression Estimator". Suppose two

variables, X and Y, have a linear relationship of the form $Y = a + bX$ for some unknown values a and b, and that the population average of X is known, perhaps from an administrative database or a census. One approach to estimate the average of Y is to take a sample of X and Y, estimate a and b from that sample, and estimate μ_y from $\bar{y}$, the slope b, and the difference between the observed sample average $\bar{x}$ and the known value μ_x. As in Särndal et al. (1992), the formula for the regression estimator is

$$\hat{\mu}_y = \bar{y} + b(\mu_x - \bar{x})$$
$$b = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

A natural development occurs in two-stage sampling. If we do not know the true value of μ_x, but we have a large primary sample giving an estimate of the average, $\tilde{x}$, then we can use that value in place of μ_x,

$$\hat{\mu}_y = \bar{y} + b(\tilde{x} - \bar{x})$$

However, this is the same structure as the unified estimate for the average of Y when a sample (X, Y) is taken and Y has missing values. In particular, if (X, Y) are jointly normally distributed with standard deviation 1, then the situation is exactly as in the example of Section 3.5.4.

There are also other weight-adjustment procedures in survey sampling. For example, calibration (Deville & Särndal, 1992) is used to adjust sampling weights so that weighted sample estimates of population totals match known census values. Further work exploring this relationship may find other interesting relationships between estimators from sampling theory and the unified estimate.

7.6 Using Known Population Values in the Unified Estimate

Occasionally some parameters of the population are known before the study starts. These values can be used to adjust the unified estimator, similar to how the regression estimator from survey sampling uses known population averages or totals. The is can be accomplished by including a complete case working model for that parameter but omitting the corresponding available case working model. Instead of subtracting the available case working model estimator from the complete case working model estimator, we subtract the known parameter values from the complete case working model estimator.

7.6.1 Example, Known Population Average in Linear Regression

Suppose the purpose of a study is to estimate a linear regression of Y on X_1 and X_2; $Y \sim \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$. Y is fully observed, X_1 and X_2 have MCAR missing values, the population average of X_2, μ_{X_2}, is known, and the missingness in X_1 and X_2 is non-monotone.

The analysis model is already specified. We now specify three working models. The first two will be pairs of working models fit on both the complete and available cases, as is usual for the unified approach. The third working model will be a single fit on the complete cases. The first pair of working models fits $Y \sim \gamma_{10} + \gamma_{11} X_1 + \epsilon$ on both the complete cases and available cases, the second fits $Y \sim \gamma_{20} + \gamma_{21} X_2 + \epsilon$ on both the complete cases and available cases, and the third fits $X_2 \sim \gamma_{30} + \epsilon$ on only the complete cases. Denote the estimating equation for the analysis model as $u(\beta, Y, X_1, X_2)$, the estimating equations for the first working model as $f_1(\gamma_1, Y, X_1)$ and $g_1(\gamma_1', Y, X_1)$, the

estimating equations for the second working model as $f_2(\gamma_2, Y, X_2)$ and $g_2(\gamma_2', Y, X_2)$, and the estimating equation for the third working model is given as $f_3(\gamma_3, X_2)$. Because missingness is assumed to be MCAR we do not require observation probability models. Stacking the estimating equations we have

$$0 = \sum S(\theta, Y, X_1, X_2) = \sum \begin{pmatrix} R_0 u(\beta, Y, X_1, X_2) \\ R_0 f_1(\gamma_1, Y, X_1) \\ R_0 f_2(\gamma_2, Y, X_2) \\ R_0 f_3(\gamma_3, X_2) \\ R_1 g_1(\gamma_1', Y, X_1) \\ R_2 g_2(\gamma_2', Y, X_2) \end{pmatrix} \tag{7.1}$$

As before, denote the solution to (7.1) as $\hat{\theta} = (\hat{\beta}, \hat{\gamma}_1, \hat{\gamma}_2, \hat{\gamma}_3, \hat{\gamma}_1', \hat{\gamma}_2')$. Under regularity conditions, the $\hat{\theta}$ is consistent for the vector of true values, $(\beta^*, \gamma_1^*, \gamma_2^*, \mu_{X_2}, \gamma_1^*, \gamma_2^*)$ and $\sqrt{n}(\hat{\theta} - \theta^*)$ has an asymptotic joint normal distribution. Denote the asymptotic variance of $\sqrt{n}\hat{\theta}$ by Σ. We now modify the usual fixed permutation matrix. We instead take two fixed matrices,

$$P = \begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & -I & 0 \\ 0 & 0 & I & 0 & 0 & -I \\ 0 & 0 & 0 & I & 0 & 0 \end{pmatrix}, Q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mu_{X_2} \end{pmatrix}$$

Where by abusing notation I are various appropriately sized identity matrices and 0 are various appropriately sized matrices or vectors of 0s. Multiplying $\hat{\theta}$ by P and the

subtracting Q gives the estimate,

$$\sqrt{n}(P\hat{\theta} - Q) = \left[\begin{pmatrix} \hat{\beta} \\ \hat{\gamma}_1 - \hat{\gamma}_1' \\ \hat{\gamma}_2 - \hat{\gamma}_2' \\ \hat{\gamma}_3 - \mu_{X_2} \end{pmatrix} - \begin{pmatrix} \beta^* \\ 0 \\ 0 \\ 0 \end{pmatrix} \right] \rightarrow N(0, P\Sigma P^T)$$

From here arguments regarding the conditional distribution of $\sqrt{n}(P\hat{\theta} - Q)$ apply as in the usual unified estimate.

In applied work certain properties of the population are often known, such as the average age, gender distribution, or total income. Further development of this extension of the unified approach may be used to incorporate that information in the missing-data adjustment.

References

Andersen, P. K., & Gill, R. D. (1982). Cox's regression model for counting processes: A large sample study. *The Annals of Statistics*, *10*(4), 1100–1120.

Bartlett, J. W., Seaman, S. R., White, I. R., & Carpenter, J. R. (2015). Multiple imputation of covariates by fully conditional specification: Accommodating the substantive model. *Statistical Methods in Medical Research*, *24*(4), 462–487.

Basch, E. (2013). Toward patient-centered drug development in oncology. *The New England Journal of Medicine*, *369*(5), 397–400.

Breslow, N. (1974). Covariance analysis of censored survival data. *Biometrics*, *30*(1), 89–99.

Chen, Y. H., & Chen, H. (2000). A unified approach to regression analysis under double-sampling designs. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *62*(3), 449–460. https://doi.org/10.1111/1467-9868.00243

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, *34*(2), 187–202.

Deville, J.-C., & Särndal, C.-E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association*, *87*(418), 376–382.

Dobson, A. J., & Barnett, A. G. (2018). *An introduction to generalized linear models*. CRC Press.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*(1), 1–26. https://doi.org/10.1214/aos/1176344552

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, *47*(260), 663–685.

Huber, P. J. (1967). The behavior of maximum likelihood estimates under non-standard conditions. *In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. University of California Press,Berkeley, California, Pp. 221–233.*

Huber, P. J. (1973). Robust regression: Asymptotics, conjectures and monte carlo. *The Annals of Statistics*, *1*(5), 799–821.

Imhof, J. P. (1961). Computing the distribution of quadratic forms in normal variables. *Biometrika*, *48*(3-4), 419–426.

Kalbfleisch, J. D., & Prentice, R. L. (2002). *The statistical analysis of failure time data* (2nd ed.). Wiley.

Levine, M. N., Guyatt, G. H., Gent, M., De Pauw, S., Goodyear, M. D., Hryniuk, W. M., Arnold, A., Findlay, B., Skillings, J. R., & Bramwell, V. H. (1988). Quality of life in stage II breast cancer: An instrument for clinical trials. *Journal of Clinical Oncology*, *6*(12), 1798–1810. https://doi.org/10.1200/JCO.1988.6.12.1798

Levine, M. N., Pritchard, K. I., Bramwell, V. H. C., Shepherd, L. E., Tu, D., & Paul, N. (2005). Randomized trial comparing cyclophosphamide, epirubicin, and fluorouracil with cyclophosphamide, methotrexate, and fluorouracil in premenopausal women with node-positive breast cancer: Update of national cancer institute of canada clinical trials group tr. *Journal of Clinical Oncology*, *23*(22), 5166–5170.

Lin, D. Y., & Wei, L. J. (1989). The robust inference for the cox proportional hazards model. *Journal of the American Statistical Association*, *84*(408), 1074–1078.

Lione, H., & Wickham, H. (2022). *rlang: Functions for Base Types and Core R and 'Tidyverse' Features.*

Little, R. J. A., & Rubin, D. B. (2019). *Statistical analysis with missing data., 3rd*

edition. Wiley.

Liu, M. (2013). *Generalized unified approach to regression models with covariates missing in nonmonotone patterns* [PhD Thesis]. University of Regina.

Liu, M., & Zhao, Y. (2022). Weighted generalized estimating equations and unified estimation for longitudinal data with nonmonotone missing data patterns. *Statistics in Medicine*, *41*(7), 1148–1156.

Mann, H. B., & Wald, A. (1943). On stochastic limit and order relationships. *The Annals of Mathematical Statistics*, *14*(3), 217–226. https://doi.org/10.1214/aoms/1177731415

Mathai, A. M., & Provost, S. B. (1992). *Quadratic forms in random variables : Theory and applications*. M. Dekker.

McCullagh, P. (2019). *Generalized linear models*. CRC Press.

Pierce, D. A. (1982). The asymptotic effect of substituting estimators for parameters in certain types of statistics. *The Annals of Statistics*, *10*(2), 475–478.

Pugh, M., Robins, J. M., Lipsitz, S., & Harrington, D. (1993). *Inference in the cox proportional hazards model with missing covariate data* [Technical Report]. Dana-Farber Cancer Institute, Boston, Division of Biostatistical Science.

Qi, L., Wang, C. Y., & Prentice, R. L. (2005). Weighted estimators for proportional hazards regression with missing covariates. *Journal of the American Statistical Association*, *100*(472), 1250–1263. http://www.tandfonline.com/doi/abs/10.1198/016214505000000295

R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, *89*(427), 846–866.

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1995). Analysis of semiparametric

regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association*, *90*(429), 106–121.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Rubin, D. B. (1986). Basic ideas of multiple imputation for nonresponse. *Survey Methodology*, *12*(1), 37–47.

Särndal, C.-E., Swensson, B., & Wretman, J. (1992). *Model assisted survey sampling*. Springer-Verlag.

Seaman, S. R., & White, I. R. (2013). Review of inverse probability weighting for dealing with missing data. *Statistical Methods in Medical Research*, *22*(3), 278–295.

Serfling, R. J. (1980). *Approximation theorems of mathematical statistics*. Wiley.

Stefanski, L. A., & Boos, D. D. (2002). The Calculus of M-Estimation. *The American Statistician*, *56*(1), 29–38.

Struthers, C. A., & Kalbfleish, J. D. (1986). Misspecified proportional hazard models. *Biometrika*, *73*(2), 363–369.

Sun, B., & Tchetgen Tchetgen, E. J. (2018). On inverse probability weighting for nonmonotone missing at random data. *Journal of the American Statistical Association*, *113*(521), 369–379. http://www.tandfonline.com/doi/abs/10.1080/01621459.2016.1256814

Tang, W. (2014). *Unified approach to partially linear model and cox proportional hazards model with missing covariates* [PhD Thesis]. University of Regina.

Therneau, T. M., & Grambsch, P. M. (2000). *Modeling survival data: Extending the Cox model*. Springer.

Thiessen, D. L., Zhao, Y., & Tu, D. (2022). Unified estimation for cox regression model with nonmonotone missing at random covariates. *Statistics in Medicine*.

van Buuren, S. (2018). *Flexible imputation of missing data*. CRC Press, Taylor & Francis Group.

Whickham, H. (2019). *Advanced R* (2nd ed.). Chapman & Hall/CRC. https://doi.or

g/10.1201/9781351201315

Zhao, Y. (2021). Semiparametric model for regression analysis with nonmonotone missing data. *Statistical Methods & Applications*, *30*(2), 461–475. https://doi.org/10.1007/s10260-020-00530-w

Zhao, Y., & Liu, M. (2021). Unified approach for regression models with nonmonotone missing at random data. *AStA Advances in Statistical Analysis*, *105*(1), 87–101. https://doi.org/10.1007/s10182-020-00389-y

Appendix A

Proofs

A.1 Proof of Optimality

In Section 3.5.3, the asymptotic variance of an alternative estimator $\tilde{\beta} = \hat{\beta} - (\Sigma_{12}\Sigma_{22}^{-1} + B)(\hat{\gamma} - \hat{\gamma}')$ with B a conformable matrix, can be found as

$$
\begin{aligned}
V[\tilde{\beta}] =& V[\hat{\beta} + (-\Sigma_{12}\Sigma_{22}^{-1} - B)(\hat{\gamma} - \hat{\gamma}')] \\
=& V[\hat{\beta}] + Cov[\hat{\beta}, (-\Sigma_{12}\Sigma_{22}^{-1} - B)(\hat{\gamma} - \hat{\gamma}')] + \\
& Cov[(-\Sigma_{12}\Sigma_{22}^{-1} - B)(\hat{\gamma} - \hat{\gamma}'), \hat{\beta}] + V[(-\Sigma_{12}\Sigma_{22}^{-1} - B)(\hat{\gamma} - \hat{\gamma}')] \\
=& V[\hat{\beta}] + Cov[\hat{\beta}, (\hat{\gamma} - \hat{\gamma}')](-\Sigma_{12}\Sigma_{22}^{-1} - B)^T + \\
& (-\Sigma_{12}\Sigma_{22}^{-1} - B)Cov[(\hat{\gamma} - \hat{\gamma}'), \hat{\beta}] + \\
& (-\Sigma_{12}\Sigma_{22}^{-1} - B)V[(\hat{\gamma} - \hat{\gamma}')](-\Sigma_{12}\Sigma_{22}^{-1} - B)^T \\
=& \Sigma_{11} + \Sigma_{12}(-\Sigma_{12}\Sigma_{22}^{-1} - B)^T + \\
& (-\Sigma_{12}\Sigma_{22}^{-1} - B)\Sigma_{21} + \\
& (-\Sigma_{12}\Sigma_{22}^{-1} - B)\Sigma_{22}(-\Sigma_{12}\Sigma_{22}^{-1} - B)^T \\
=& \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} + B\Sigma_{22}B^T
\end{aligned}
$$

A.2 Example of Efficiency of Unified Estimate

The remaining details for the example in Section 3.5.4.1 are now given. For the estimating equations

$$S(\theta, X) = \begin{pmatrix} R(y-\beta) \\ R(x-\gamma) \\ (x-\gamma') \end{pmatrix}$$

It was required to find

$$\begin{aligned}
&B(\theta^*) \\
&= E[S(\theta^*, X)S(\theta^*, X)^T] \\
&= \int S(\theta^*, X)S(\theta^*, X)^T dF(X, Y, R) \\
&= \int \begin{pmatrix} R^2(y-\beta^*)^2 & R^2(y-\beta^*)(x-\gamma^*) & R(y-\beta^*)(x-\gamma'^*) \\ R^2(y-\beta^*)(x-\gamma^*) & R^2(x-\gamma^*)^2 & R(x-\gamma^*)(x-\gamma'^*) \\ R(y-\beta^*)(x-\gamma'^*) & R(x-\gamma^*)(x-\gamma'^*) & (x-\gamma'^*)^2 \end{pmatrix} dF(X, Y, R) \\
&= \int \begin{pmatrix} R(y-\beta^*)^2 & R(y-\beta^*)(x-\gamma^*) & R(y-\beta^*)(x-\gamma^*) \\ R(y-\beta^*)(x-\gamma^*) & R(x-\gamma^*)^2 & R(x-\gamma^*)^2 \\ R(y-\beta^*)(x-\gamma^*) & R(x-\gamma^*)^2 & (x-\gamma^*)^2 \end{pmatrix} dF(X, Y, R) \\
&= \int \begin{pmatrix} \pi(y-\beta^*)^2 & \pi(y-\beta^*)(x-\gamma^*) & \pi(y-\beta^*)(x-\gamma^*) \\ \pi(y-\beta^*)(x-\gamma^*) & \pi(x-\gamma^*)^2 & \pi(x-\gamma^*)^2 \\ \pi(y-\beta^*)(x-\gamma^*) & \pi(x-\gamma^*)^2 & (x-\gamma^*)^2 \end{pmatrix} dF(X, Y) \\
&= \begin{pmatrix} \pi & \pi\rho & \pi\rho \\ \pi\rho & \pi & \pi \\ \pi\rho & \pi & 1 \end{pmatrix}
\end{aligned}$$

Note that we used $\gamma^* = \gamma'^*$. We also needed to find

$$\begin{aligned} A(\theta^*) &= E[-\frac{\partial}{\partial\theta}S(\theta^*, X)] \\ &= \int \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & 1 \end{pmatrix} dF(X, Y, R) \\ &= \begin{pmatrix} \pi & 0 & 0 \\ 0 & \pi & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

The variance of $\hat{\theta}$ is

$$\begin{aligned} V[\hat{\theta}] &= A^{-1}BA^{T-1} \\ &= n^{-1} \begin{pmatrix} \pi & 0 & 0 \\ 0 & \pi & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} \pi & \pi\rho & \pi\rho \\ \pi\rho & \pi & \pi \\ \pi\rho & \pi & 1 \end{pmatrix} \begin{pmatrix} \pi & 0 & 0 \\ 0 & \pi & 0 \\ 0 & 0 & 1 \end{pmatrix}^{T-1} \\ &= n^{-1} \begin{pmatrix} 1/\pi & 0 & 0 \\ 0 & 1/\pi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \pi & \pi\rho & \pi\rho \\ \pi\rho & \pi & \pi \\ \pi\rho & \pi & 1 \end{pmatrix} \begin{pmatrix} 1/\pi & 0 & 0 \\ 0 & 1/\pi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= n^{-1} \begin{pmatrix} 1/\pi & \rho/\pi & \rho \\ \rho/\pi & 1/\pi & 1 \\ \rho & 1 & 1 \end{pmatrix} \end{aligned}$$

With permutation matrix P,

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{pmatrix}$$

The variance of $P\hat{\theta} = (\hat{\beta}, \hat{\gamma} - \hat{\gamma}')$ is

$$\begin{aligned}
\Sigma = V[P\hat{\theta}] &= PV[\hat{\theta}]P^T \\
&= n^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1/\pi & \rho/\pi & \rho \\ \rho/\pi & 1/\pi & 1 \\ \rho & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \\
&= n^{-1} \begin{pmatrix} 1/\pi & \rho/\pi - \rho \\ \rho/\pi - \rho & 1/\pi - 1 \end{pmatrix} \\
&= n^{-1} \begin{pmatrix} 1/\pi & \rho(1-\pi)/\pi \\ \rho(1-\pi)/\pi & (1-\pi)/\pi \end{pmatrix}
\end{aligned}$$

The unified estimate is

$$\begin{aligned}
\bar{\beta} &= \hat{\beta} - \Sigma_{12}\Sigma_{22}^{-1}(\hat{\gamma} - \hat{\gamma}') \\
&= \hat{\beta} - (\rho(1-\pi)/\pi)(\pi/(1-\pi))(\hat{\gamma} - \hat{\gamma}') \\
&= \hat{\beta} - \rho(\hat{\gamma} - \hat{\gamma}')
\end{aligned}$$

And the variance of $\bar{\beta}$ is

$$\begin{aligned}
V[\bar{\beta}] &= V[\begin{pmatrix} 1 & -\rho \end{pmatrix} P\hat{\theta}] \\
&= n^{-1} \begin{pmatrix} 1 & -\rho \end{pmatrix} \begin{pmatrix} 1/\pi & \rho(1-\pi)/\pi \\ \rho(1-\pi)/\pi & (1-\pi)/\pi \end{pmatrix} \begin{pmatrix} 1 \\ -\rho \end{pmatrix} \\
&= n^{-1}(1/\pi - \rho^2(1-\pi)/\pi)
\end{aligned}$$

Alternatively, using the conditional variance formula directly gives us the same result

$$\begin{aligned} V[\bar{\beta}] &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\ &= n^{-1}(1/\pi - (\rho(1-\pi)/\pi)((1-\pi)/\pi)^{-1}(\rho(1-\pi)/\pi)) \\ &= n^{-1}(1/\pi - \rho^2(1-\pi)/\pi) \end{aligned}$$

Appendix B

R Code

The code for the unified estimate is given below. It is probably more convenient to download a copy from https://github.com/DavidLukeThiessen/nmuer. The package can be automatically downloaded and installed in R by the command:

```
devtools::install_github(repo = "DavidLukeThiessen/nmuer")
```

```
#' Non-Monotone Unified Estimate
#'
#' @description  This function fits the unified estimate for either
↪  a Cox
#' or a GLM model. It uses bootstrapping to estimate variance.
#'
#' @details   TODO
#'
#' @export
#'
#' @importFrom stats  na.fail update.formula coef complete.cases
↪  cov2cor
```

```
#' fitted.values glm binomial pchisq printCoefmat
#'
#' @param main_model (Required) This is the main analysis model.
↪  This should
#' be either a glm() or coxph() function which includes a formula
↪  but does
#' not include a data argument
#' @param working_models (Required) A list containing the working
↪  models,
#' similar to the main_model specification.
#' @param data (Required) A dataframe with 1 row for each
↪  individual.
#' @param main_weight_model (Optional) A function that takes at
↪  minimum a
#' Data argument and a observation_vector argument, and outputs a
↪  vector of
#' weights. These weights are used in the main_model.
#' @param working_weight_models (Optional) A list of functions that
↪  each
#' take DATA and observation_vector arguments and output a vector of
#' weights.
#' @param variance_estimate (Optional) There are two different ways
↪  of
#' estimating the variance. Only "bootstrap" is currently allowed.
#' "bootstrap" uses bootstrap resampling on rows of the data to
↪  estimate
```

```
#' the variance. "sandwich" will use a robust sandwich estimate of
↪  the form
#' A^-1 * B * A^T-1, where A is the matrix of partial
#' derivatives of the score with respect to the parameters and B is
↪  the
#' matrix of crossproducts of the score vectors.
#' @param num_boots (Optional) The number of bootstrap replications
↪  to use
#' when using the bootstrap method of estimating variance. 200 was
#' recommended as the minimum by Efron, pg 52.
#' @param DEBUG (Optional) Prints extra model fitting and call
↪  information
#' @examples
#' if(require(mice)) {
#'   nmuer_glm_mod <- nmuer(
#'     main_model = glm(hm ~ age*sex),
#'     working_models = list(glm(hr ~ age*sex)),
#'     data = mice::selfreport)
#'   summary(nmuer_glm_mod)
#' }
nmuer <- function(
    main_model,
    working_models,
    data,
    main_weight_model = NULL,
    working_weight_models = NULL,
```

```
    variance_estimate = c("bootstrap","sandwich"),
    num_boots = 500,
    DEBUG = FALSE
) {
  # Start debugger
  ↪ ----------------------------------------------------------
  # Uncomment to enable stepping through the code when DEBUG is TRUE
  # if(DEBUG == TRUE) {
  #   browser()
  # }
  # Read and verify input
  ↪ ------------------------------------------------
  cal <- match.call()
  variance_estimate <- match.arg(variance_estimate)
  if(variance_estimate == "sandwich") {
    stop("Sandwich variance not implemented yet")
  }

  if(any(c("R0","R", "main_weights","working_weights") %in%
         names(data))) {
    stop("Conflict between nmuer internal variables and
         variables defined in data. Please rename any variables named
         R, R0, main_weights, or working_weights")
  }

  # Standardize model specifications
  ↪ -------------------------------------
```

```
  # Need to capture the code with enexpr before proceeding,
  ↪  otherwise R
  # will try to run the code and encounter errors.
  main_model <- rlang::enexpr(main_model)
  working_models <- rlang::enexpr(working_models)
  main_weight_model <- rlang::enexpr(main_weight_model)
  working_weight_models <- rlang::enexpr(working_weight_models)

  # If main_weight_model or working_weight_models are NULL, then use
  # the constant weight functions. IE, assume MCAR.
  if(is.null(main_weight_model)) {
    main_weight_model <- rlang::expr(nmuer_weights_constant())
  }
  if(is.null(working_weight_models)) {
    working_weight_models <- list()
    for(i in 1:length(working_models)) {
      working_weight_models[[i]] <-
↪   rlang::expr(nmuer_weights_constant())
    }
  }

  if(DEBUG == TRUE) {
    print("Before call_standardise()")
    print("main model:")
    print(main_model)
    print("working models:")
```

```
    print(working_models)
    print("main_weight_model:")
    print(main_weight_model)
    print("working_weight_models:")
    print(working_weight_models)
  }
  if(length(working_models) != length(working_weight_models)) {
    stop("working_models and working_weight_models must be the same
    ↪  length")
  }

  main_model <- rlang::call_standardise(main_model)
  main_weight_model <- rlang::call_standardise(main_weight_model)
  for(i in 2:length(working_models)) {
    working_models[[i]] <-
↪   rlang::call_standardise(working_models[[i]])
    working_weight_models[[i]] <-
      rlang::call_standardise(working_weight_models[[i]])
  }
  if(DEBUG == TRUE) {
    print("After call_standa")
    print("main model:")
    print(main_model)
    print("working models:")
    print(working_models)
    print("main_weight_model:")
```

```
  print(main_weight_model)
  print("working_weight_models:")
  print(working_weight_models)
}
# Check if models are adequately specified
 ↪  ------------------------------
nmuer_check_specification(main_model)
for(i in 2:length(working_models)) {
  nmuer_check_specification(working_models[[i]])
}


# Check if there is missing data
 ↪  ---------------------------------------
# There should be some missing data in the main model, and the
# working models should not have exactly the same observed data
# as the main model
R0 <- stats::complete.cases(data[,all.vars(main_model$formula)])
if(all(R0)) {
  stop("the main_model has no missing data")
}
R <- matrix(data = NA, nrow = nrow(data), ncol =
↪  length(working_models)-1)
for(i in 1:(length(working_models)-1)) {
  R[,i] <-
↪  stats::complete.cases(data[,all.vars(working_models[[i+1]])])
  if(all(R0 == R[,i])) {
```

```
    stop(paste0("main_model missingness exactly matches some ",
                "working_model missingness"))
  }
}

# Fit models
 ↪  ------------------------------------------------------------
main_fit <-
  nmuer_fitcoef(data = data,
                main_model = main_model,
                working_models = working_models,
                main_weight_model = main_weight_model,
                working_weight_models = working_weight_models,
                return_fit = TRUE,
                DEBUG = DEBUG)
all_params <- main_fit[c("main_params",
                         "working_params_cc",
                         "working_params_full")]
num_params <- c(length(all_params$main_params),
                length(all_params$working_params_cc),
                length(all_params$working_params_full))
num_working_params <- main_fit$num_working_params
param_vec <- c(all_params$main_params,
               all_params$working_params_cc,
               all_params$working_params_full)
```

```r
# Fit Variance
↪  ------------------------------------------------------------
# Note: If any bootstrap parameters are NA, it tries to bootstrap
↪  again
maxit <- 100
if(variance_estimate == "bootstrap") {
  boot_params <- matrix(data = 0,
                        nrow = num_boots,
                        ncol = sum(num_params))
  for(i in 1:num_boots) {
    continue <- TRUE
    curit <- 1
    while(continue) {
      if(curit > maxit) {
        stop("Bootstrapping failed because of NA parameters")
      }
      boot_data <- data[sample(1:nrow(data),replace=TRUE),]
      boot_params[i,] <- unlist(nmuer_fitcoef(
        data = boot_data,
        main_model = main_model,
        working_models = working_models,
        main_weight_model = main_weight_model,
        working_weight_models = working_weight_models,
        return_fit = FALSE))
      curit <- curit + 1
      if(!(any(is.na(boot_params[i,])))) {
```

```
        continue <- FALSE
      }
    }
  }
  mean_boot_est <- colMeans(boot_params)
  boot_var <- matrix(data = 0,
                     nrow = sum(num_params),
                     ncol = sum(num_params))
  for(i in 1:num_boots) {
    boot_var <- boot_var +
      tcrossprod(boot_params[i,] - mean_boot_est)
  }
  sigma <- boot_var/(num_boots - 1)
} else {
  stop("only bootstrapping variance is currently supported")
}

theta_cor_matrix <- stats::cov2cor(sigma)
if(DEBUG == TRUE) {
  print("the estimated theta covariance matrix is:")
  print(sigma)
  print("the estimated theta correlation matrix is:")
  print(theta_cor_matrix)
}
# calculate the correlations between beta_hat and gamma_hat and
↪  gamma_bar
```

```
beta_gammahat_cors <- list()
beta_gammabar_cors <- list()

for(i in seq_along(num_working_params)) {
  beta_gammahat_cors[[i]] <-
    theta_cor_matrix[1:num_params[1],
                     (num_params[1]+1):
                       (num_params[1]+sum(num_working_params[0:i]))]
  beta_gammabar_cors[[i]] <-
    theta_cor_matrix[1:num_params[1],
                     (num_params[1]+1+num_params[2]):
                       (num_params[1]+num_params[2] +
                          sum(num_working_params[0:i]))]
}

# Calculate Variance
↪  ------------------------------------------------
# Multiply the covariance matrix of theta by a permutation matrix
↪  to
# get the covariance matrix of (beta, gamma_hat - gamma_bar),
↪  called
# omega
permutation_matrix <-
  matrix(data = 0,
         nrow = (num_params[1] + num_params[2]),
         ncol = sum(num_params))
```

```
  permutation_matrix[1:num_params[1],
                     1:num_params[1]] <-
    diag(1, num_params[1])
  permutation_matrix[(num_params[1] + 1):(num_params[1] +
↪   num_params[2]),
                     (num_params[1] + 1):(num_params[1] +
↪   num_params[2])] <-
    diag(1, num_params[2])
  permutation_matrix[(num_params[1] + 1):(num_params[1] +
↪   num_params[2]),
                     (num_params[1] + num_params[2] +
↪   1):sum(num_params)] <-
    diag(-1, num_params[2])
  omega <- permutation_matrix %*% sigma %*% t(permutation_matrix)
  if(DEBUG == TRUE) {
    print("the estimated omega covariance matrix is:")
    print(omega)
    print("the estimated omega correlation matrix is:")
    print(stats::cov2cor(omega))
  }


  # Calculate beta_bar
   ↪  ------------------------------------------------
  beta_hat <- all_params$main_params
  gamma_hat <- all_params$working_params_cc
  gamma_bar <- all_params$working_params_full
```

```r
omega_11 <- omega[
  1:num_params[[1]],
  1:num_params[[1]]]
omega_12 <- omega[
  1:num_params[[1]],
  (num_params[[1]] + 1):(num_params[[1]] + num_params[[2]])]
omega_22 <- omega[
  (num_params[[1]] + 1):(num_params[[1]] + num_params[[2]]),
  (num_params[[1]] + 1):(num_params[[1]] + num_params[[2]])]
beta_bar <- beta_hat - (omega_12 %*% solve(omega_22) %*%
                          (gamma_hat - gamma_bar))

# create output
 ↪  ----------------------------------------------------------
final_fit <- main_fit$main_model_fit
final_fit$coefficients <- beta_bar

# Calculate beta_bar variance
 ↪  ----------------------------------------
beta_var_reduction <- omega_12 %*% solve(omega_22) %*% t(omega_12)
beta_bar_var <- omega_11 - beta_var_reduction
# Below code gives exactly the same variance estimate, as
 ↪  expected.
# lambda_transform_matrix <-
#   matrix(0,
#          nrow = num_params[[1]],
```

```
#           ncol = num_params[[1]] + num_params[[2]])
# lambda_transform_matrix[1:num_params[[1]],
#                         1:num_params[[1]]] <-
#   diag(1,
#        nrow = num_params[[1]],
#        ncol = num_params[[1]])
# lambda_transform_matrix[
#   1:num_params[[1]],
#   (num_params[[1]]+1):(num_params[[1]] + num_params[[2]])] <-
#   -(omega_12 %*% solve(omega_22))
# beta_bar_var <-
#   lambda_transform_matrix %*% omega %*%
 ↪  t(lambda_transform_matrix)

# Add names to output
 ↪  ------------------------------------------------
beta_bar_names <- names(all_params$main_params)
beta_bar <- as.vector(beta_bar)
names(beta_bar) <- beta_bar_names
dimnames(beta_bar_var) <- list(beta_bar_names,beta_bar_names)

# Return Output
 ↪  ------------------------------------------------------------
out <- list(main_model_type = main_model[[1]],
            final_fit = final_fit,
            beta_bar = beta_bar,
```

```
              beta_bar_var = beta_bar_var,
              variance_estimate = variance_estimate,
              num_boots = num_boots,
              unadjusted_betahat = beta_hat,
              call = cal,
              num_working_models = length(working_models)-1,
              beta_gammahat_cors = beta_gammahat_cors,
              beta_gammabar_cors = beta_gammabar_cors,
              unadjusted_betahat_var = omega_11,
              beta_var_reduction = beta_var_reduction)
  class(out) <- "nmuer"
  return(out)
}


#' Weighting functions for unified estimate
#'
#' @description
#' Functions to supply inverse probability weights for unified
#' estimates
#'
#' @details
#' When missingness is MAR, inverse probability weights (ipw) can
#' be used to rebalance the estimating equations to allow for
 ↪  unbiased
#' estimation of parameters. These functions are designed to allow
#' nmuer() to automatically estimate the weights. Users can also
 ↪  write
```

```
#' their own weight functions. The two functions supplied here allow
#' for constant weights and weights estimated by a logistic model.
#'
#' Each weight function MUST take an argument called DATA and an
 ↪  argument
#' called observation_vector. The observation_vector and DATA should
 ↪  be
#' left NULL, the nmuer() code automatically calculates and includes
 ↪  them.
#' Other arguments may be defined depending on the method of
 ↪  estimating
#' weights.
#'
#' Each weight function must output a vector of weights the same
#' length as the number of rows in DATA. These weights are then
#' supplied to the model fitting functions glm() and coxph().
#'
#' nmuer_weights_constant() gives all observations the same weight.
#' This is appropriate for MCAR data. The user does not need to
#' enter any additional information to use this method. This is the
#' default behaviour if no weight functions are supplied, but
 ↪  explicitly
#' calling nmuer_weights_constat() is required if you want to
 ↪  combine
#' some constant weights with some non-constant weights.
#'
```

```
#' nmuer_weights_logistic() uses a GLM model with a binomial family
↪  and
#' a logistic link function to estimate the probability a row has
↪  complete
#' data, then takes the inverse of that probability as the weight. A
#' missing_formula argument is required. The right-hand side of the
#' formula will be used to model observation probability. The
↪  left-hand
#' side of the formula will be automatically replaced by nmuer() to
↪  use
#' the appropriate observation vector. If the working_model has no
#' missing data, then nmuer_weights_logistic() will return a vector
↪  of 1s
#' with a warning.
#'
#' @export
#'
#' @param DATA A dataframe on which the weighting function will
#' be fit to calculate the weights. This is automatically supplied
#' by nmuer()
#' @param observation_vector A logical vector indicating
#' if the ith row has all variables in the analysis or working model
#' observed. This is automatically supplied by nmuer().
#' @param missing_formula A formula, the right-hand side defines
#' the variables to be used in the logistic model. This must be
↪  supplied
```

```
#' by the user.
nmuer_weights_constant <- function(observation_vector = NULL,
                                   DATA = NULL) {
  if(is.null(observation_vector) | is.null(DATA)) {
    stop("Missing observation_vector or DATA")
  }
  return(rep(1,nrow(DATA)))
}


#' @rdname nmuer_weights_constant
#' @export
nmuer_weights_logistic <- function(missing_formula,
                                   observation_vector = NULL,
                                   DATA = NULL) {
  if(is.null(observation_vector) | is.null(DATA)) {
    stop("Missing observation_vector or DATA")
  }
  if(all(observation_vector)) {
    warning(paste0("No missing values in nmuer_weights_logistic ",
                   "detected. Using constant weights."))
    return(rep(1,nrow(DATA)))
  }
  if("R0" %in% names(DATA)) {
    stop("R0 already exists in DATA")
  }
  DATA$R0 <- observation_vector
```

```
  environment(missing_formula) <- environment()
  missing_formula <- stats::update.formula(old = missing_formula,
                                           new = R0 ~.)
  if(any(!stats::complete.cases(DATA[,all.vars(missing_formula)]))) {
    stop("A variable in missing_formula has missing values")
  }
  mod <- stats::glm(formula = missing_formula,
                    family = stats::binomial(link = "logit"),
                    data = DATA,
                    na.action = stats::na.fail)
  weights <- 1/stats::fitted.values(mod)
  if(any(weights > 1000)) {
    warning("Weights > 1000 detected in nmuer_weights_logistic")
  }
  return(weights)
}


# nmuer_check_specification checks if the supplied model has enough
# information to be fit.
# For now I only allow glm() and coxph().
# They must have a formula.
# In the future other models will probably be allowed, and they
# may have other options.
nmuer_check_specification <- function(model) {
  if(!(as.character(model[[1]]) %in% c("glm","coxph"))) {
    stop("only glm and coxph models are currently supported")
```

```
  }
  if(is.null(model$formula)) {
    stop("all analysis and working models must have a formula")
  }
  # if(as.character(model[[1]]) == "geeglm") {
  #   stop("geeglm models are not currently supported")
  #   if(!(id %in% names(model))) {
  #     stop("geeglm models must have an id specified")
  #   }
  # }
}


# nmuer_fitcoef function takes the validated model specifications
↪  and data
# and fits the models. It calculates weights (if they were
↪  specified) and
# fits the models on the appropriate subsets of the data.
# It returns only the coefficients from the analysis
# and working models. It does not calculate variance.
nmuer_fitcoef <- function(
  data,
  main_model,
  working_models,
  main_weight_model,
  working_weight_models,
  return_fit = FALSE,
```

```
  DEBUG = FALSE
) {
  if(DEBUG == TRUE) {
    browser()
  }
  # Calculate weights for the main model and complete case working
  # models. This happens even if missingness is MCAR.
  main_vars <- all.vars(main_model$formula)
  R0 <- stats::complete.cases(data[,main_vars])
  main_weight_model$observation_vector <- R0
  main_weight_model$DATA <- data
  main_weights <- eval(main_weight_model,envir = data)

  # Calculate weights for available case working models
  # Note: i starts at 2, because of strangeness in rlang package.
  # Note: if there is no missingness in working_models, then
  # nmuer_weights_logistic() will return all 1s.
  working_vars <- list()
  R <- matrix(data = FALSE, nrow = nrow(data), ncol =
↪   (length(working_models)-1))
  working_weights <-
    matrix(data = 0, nrow = nrow(data), ncol =
    ↪   (length(working_models)-1))
  for(i in 2:length(working_models)) {
    working_vars[[i-1]] <- all.vars(working_models[[i]]$formula)
    R[,(i-1)] <- stats::complete.cases(data[,working_vars[[i-1]]])
```

```
    working_weight_models[[i]]$observation_vector <- R[,(i-1)]
    working_weight_models[[i]]$DATA <- data
    working_weights[,(i-1)] <- eval(working_weight_models[[i]],
                                    envir = data)
  }
  working_models_cc <- working_models #uses only analysis complete
↪   cases
  working_models_full <- working_models #uses any available cases

  # Modify function calls with the appropriate weights, data, and
  ↪  subset.
  # main_model$weights <- main_weights
  # main_model$data <- data
  # main_model$subset <- R0
  # main_model$na.action <- stats::na.fail
  main_model$weights <- quote(main_weights)
  main_model$data <- quote(data)
  main_model$subset <- quote(R0)
  main_model$na.action <- quote(stats::na.fail)

  for(i in 2:length(working_models)) {
    working_models_cc[[i]]$weights <- main_weights
    working_models_cc[[i]]$data <- data
    working_models_cc[[i]]$subset <- R0
    working_models_cc[[i]]$na.action <- stats::na.fail
    working_models_full[[i]]$weights <- working_weights[,(i-1)]
```

```
  working_models_full[[i]]$data <- data
  working_models_full[[i]]$subset <-  R[,(i-1)]
  working_models_full[[i]]$na.action <- stats::na.fail
}

# If returning the model to use as final_fit, include the x matrix
main_model$x <- return_fit

# Fit the models
main_model_fit <- eval(main_model, envir = data)
working_model_fits_cc <-
  vector(mode = "list", length = length(working_models))
working_model_fits_full <-
  vector(mode = "list", length = length(working_models))
for(i in 2:length(working_models)) {
  working_model_fits_cc[[i]] <-
    eval(working_models_cc[[i]], envir = data)
  working_model_fits_full[[i]] <-
    eval(working_models_full[[i]], envir = data)
}

# Extract parameters and return results
main_params <- stats::coef(main_model_fit)
working_params_cc <- c()
working_params_full <- c()
num_working_params <- c()
```

```
  for(i in 2:length(working_models)) {
    working_params_cc <-
      c(working_params_cc,
        stats::coef(working_model_fits_cc[[i]]))
    working_params_full <-
      c(working_params_full,
        stats::coef(working_model_fits_full[[i]]))
    num_working_params <-
      c(num_working_params,
        length(stats::coef(working_model_fits_cc[[i]])))
  }
  out <- list(main_params = main_params,
              working_params_cc = working_params_cc,
              working_params_full = working_params_full)
  if(return_fit == TRUE) {
    out$main_model_fit <- main_model_fit
    out$num_working_params <- num_working_params
  }
  return(out)
}
# S3 Methods
 ↪  ----------------------------------------------------------------
#' @export
coef.nmuer <- function(object, ...) {
  object$beta_bar
}
```

```r
#' @export
vcov.nmuer <- function(object, ...) {
  object$beta_bar_var
}
#' @export
summary.nmuer <- function(object,
                          digits = max(1, getOption("digits") - 2),
                          ↪  ...) {
  print(object, digits = digits, ...)
}
#' @export
print.nmuer <- function(x, digits = max(1, getOption("digits") - 2),
↪  ...) {
  cat("A non-monotone unified estimate with a ",
      deparse(x$main_model_type), " analysis model and ",
      x$num_working_models,  " working model",
      ifelse(x$num_working_models == 1, ".\n", "s.\n"),
      sep = "")
  cat("\nCall:  ", paste(deparse(x$call), sep = "\n", collapse =
  ↪  "\n"),
      "\n\n", sep = "")
  if(x$main_model_type == "coxph") {
    cat("coefficients:\n")
    coef <- coef(x)
    se <- sqrt(diag(x$beta_bar_var))
    result <- cbind(coef, se, exp(coef), coef/se,
```

```
                    pchisq((coef/se)^2,
                           1, lower.tail = FALSE))
    dimnames(result) <- list(names(coef(x)),
                             c("coef","se(coef)","exp(coef)", "z",
                              ↪  "p"))
    stats::printCoefmat(x = result, digits = digits, signif.stars =
↪   TRUE,
                        P.values = TRUE, has.Pvalue = TRUE)
  } else if(x$main_model_type == "glm") {
    cat("coefficients:\n")
    coef <- coef(x)
    se <- sqrt(diag(x$beta_bar_var))
    result <- cbind(coef, se, coef/se,
                    pchisq((coef/se)^2,
                           1, lower.tail = FALSE))
    dimnames(result) <- list(names(coef(x)),
                             c("coef","se(coef)", "z", "p"))
    stats::printCoefmat(x = result, digits = digits, signif.stars =
↪   TRUE,
                        P.values = TRUE, has.Pvalue = TRUE)
  } else {
    print("unsupported main_model type")
  }
  invisible(x)
}
```